Data-Efficient Knowledge Distillation with Teacher Assistant-Based Dynamic Objective Alignment

Yangyan Xu[1,2], Cong Cao[1], Fangfang Yuan[1(✉)], Rongxin Mi[3(✉)], Dakui Wang[1], Yanbing Liu[1,2], and Majing Su[4]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{xuyangyan,caocong,yuanfangfang,wangdakui,liuyanbing}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
mirongxin@cert.org.cn
[4] The 6th Research Institute of China Electronic Corporations, Beijing, China
sumj@ncse.com.cn

Abstract. Pre-trained language models encounter a bottleneck in production due to their high computational cost. Model compression methods have emerged as critical technologies for overcoming this bottleneck. As a popular compression method, knowledge distillation transfers knowledge from a large (teacher) model to a small (student) one. However, existing methods perform distillation on the entire data, which easily leads to repetitive learning for the student. Furthermore, the capacity gap between the teacher and student hinders knowledge transfer. To address these issues, we propose the Data-efficient Knowledge Distillation (DeKD) with teacher assistant-based dynamic objective alignment, which empowers the student to dynamically adjust the learning process. Specifically, we first design an entropy-based strategy to select informative instances at the data level, which can reduce the learning from the mastered instances for the student. Next, we introduce the teacher assistant as an auxiliary model for the student at the model level to mitigate the degradation of distillation performance. Finally, we further develop the mechanism of dynamically aligning intermediate representations of the teacher to ensure effective knowledge transfer at the objective level. Extensive experiments on the benchmark datasets show that our method outperforms the state-of-the-art methods.

Keywords: Pre-trained language model · Model compression · Knowledge distillation

1 Introduction

Large-scale pre-trained language models (PLMs), such as BERT [1], XLNet [24], RoBERTa [5], T5 [9], and GPT-4 [8], have reached very competitive performance

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

and simply require fine-tuning of downstream natural language processing (NLP) tasks [4, 23]. However, PLMs require large computational resources for huge amounts of model parameters, which leads to overloaded GPU usage and slow inference speeds in real-world production. To reduce computation and carbon footprint, knowledge distillation (KD) [2] has emerged as an effective method to compress large models into small ones and has gradually become the most popular choice among various compression methods.

Table 1. The performance comparison between BERT-base and BERT-large under different numbers of layers and varying data conditions.

Teacher	**Layer**	**SST-2**	**QNLI**
		Acc	**Acc**
BERT-base	3-layer (50%)	85.44	82.24
	3-layer	85.55	82.46
	12-layer	92.55	91.32
BERT-large	3-layer (50%)	83.25	78.94
	3-layer	83.71	79.22
	24-layer	93.00	92.66

The core concept of KD is based on the teacher-student learning framework, in which the teacher transfers knowledge to the student via soft targets. Existing KD methods [3, 6, 12, 13, 15, 21] mainly focus on transferring knowledge from the teacher model to the student model in the form of single or multiple teacher models. However, these methods have two major drawbacks: (1) They do not take into account the student's mastery of knowledge during the distillation process, so the student continues to learn instances that contain repeated information. (2) They ignore the capability gap between the small student and the large teacher, which degrades the distillation performance. For example, as shown in Table 1, part of the data can produce the similar distillation performance as all the data. This is because the student model can gain important knowledge from a portion of the informative data. Furthermore, the 3-layer student model distilled from the stronger teacher model is weaker than the same student model distilled from the weaker teacher model on the same tasks. Generally speaking, BERT-large performs better than BERT-base on the SST-2 and QNLI tasks, but a stronger teacher model does not always lead to a better student model. The reason is that the competency of the small student model cannot match that of the large teacher model, which weakens distillation performance.

Based on the above insights, in this paper, we propose the Data-efficient Knowledge Distillation (DeKD) with teacher assistant-based dynamic objective alignment, which promotes knowledge transfer from the teacher model to the student model and improves the distillation performance as the competency of the student evolves. On the one hand, as distillation progresses, the student's learning on a downstream task is gradually deepened, and examples that the stu-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

dent has already learned should be eliminated, which inspires us to investigate which data is more important to distillation. On the other hand, the capacity gap between the weak student and the strong teacher motivates us to overcome the limitations of the student. We strive to answer the following research questions: (RQ1) Which data is actually useful for the student model during the distillation process? (RQ2) How to consider the evolution of the student model to realize efficient distillation? Specifically, we first choose representative instances to learn based on entropy to maximize data efficiency and prevent the student from repeating learning at the data level. Then, we introduce a teacher assistant model at the model level, which allows the student to decide whether to query the teacher or the teacher assistant for enhancing the performance of KD. Moreover, at the objective level, we further design a dynamic objective alignment strategy that aligns the informative layers to alleviate the objective supervision problem between the large teacher and the small student. We conduct extensive experiments on several benchmark datasets to validate the effectiveness of our method. Experimental results clearly show that our DeKD significantly boosts the performance of the student model.

As a summary, the contributions of this paper are threefold:

- We are the first to consider efficient KD from the data, model, and objective levels, which is critical but overlooked by existing KD methods.
- We choose informative instances based on the prediction entropy of the student to achieve a competitive performance on part of the data. Meanwhile, we introduce the teacher assistant model and dynamic supervision alignment to improve the performance of the student as it evolves.
- We conduct extensive experiments on the benchmark datasets, and the results demonstrate that our method outperforms the state-of-the-art distillation methods.

2 Related Work

Knowledge distillation [2,22] aims to compress the knowledge of a large and computationally complex model into a simple and computationally efficient model. The KD approach has been widely used in the compression of pre-trained language models. Existing KD methods for compressing large-scale language models are divided into general distillation and task-specific distillation.

General distillation refers to conducting KD on the universal text corpus. For example, DistilBERT [10] presents a method for pre-training a smaller general-purpose language representation model, which can subsequently be fine-tuned to perform well on a variety of tasks. PD [14] demonstrates that pre-training is still crucial in the setting of smaller architectures, and that fine-tuning pre-trained compact models may compete with more complicated strategies suggested in concurrent work. In addition, the large Transformer-based pre-trained models can be compressed using a straightforward method called deep self-attention distillation, which is presented in MiniLM [19]. The self-attention module of the large model (teacher), which is crucial to Transformer networks, is deeply

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

imitated to train the small model (student). In the subsequent work, MiniLMv2 [18] uses the self-attention relation distillation to generalize and streamline the deep self-attention distillation of MiniLM [19]. The above studies of general distillation require extra training time and computational resources, and they are not applicable in resource-limited scenarios.

Instead, task-specific distillation trains the student model on specific downstream tasks. In particular, BERT-PKD [13] encourages the student model to extract knowledge from the intermediate layers of the teacher model, rather than just learning parameters from the last layer of the teacher model. Recently, the idea of combining the knowledge from models with different capacities has been explored [6, 12, 15]. Besides, MUKI [3] broadens the concept of KD from mimicking teachers to integrating teacher knowledge, and proposes Knowledge Integration (KI) for PLMs. KI attempts to train a flexible student capable of making predictions over the union of teacher label sets given multiple fine-tuned teacher-PLMs, each of which is capable of conducting classification over a unique label set. Nevertheless, the effectiveness of distilling the knowledge from a large language model into a small one has not yet been well studied. The phenomenon of the student repeating learning and the gap in capacity between the large-scale teacher and the compact student still exist.

In this study, we focus on the task-specific distillation, which is widely used in practice. Compared to previous KD approaches, we further investigate efficient KD and consider the competency evolution of the student model to comprehensively improve the distillation effect.

3 Methodology

We propose the DeKD framework, and Fig. 1 depicts its general design. Firstly, we choose informative data using an entropy-based approach to prevent repeated learning of the student model. Then, we introduce the teacher assistant to make the student model match the competency of the teacher model, thus alleviating the ability gap to boost the distillation performance. In addition, objective alignment can also bring additional performance improvement. The specific implementation strategy is to select the informative layers from the teacher and then let our student align with the teacher's hidden representations of these layers.

3.1 Preliminary

The goal of knowledge distillation is to train the student model S not just using the information supplied by true labels but also by studying how the teacher model T represents and interacts with the data.

KD [2] uses the teacher's model outputs, for instance, as a soft learning target for the student. We represent $S(x)$ and $T(x)$ as the output logit vectors of the student and the teacher for input x, respectively. Then, the Kullback-Leibler

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

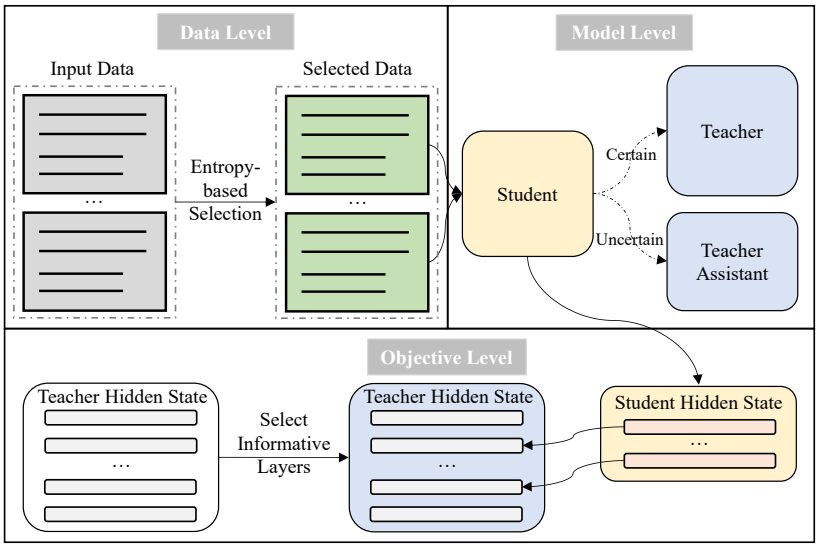


Fig. 1. Our proposed DeKD framework.

(KL) divergence loss between the student and teacher output is calculated as follows:

$$\mathcal{L}_{KL} = \mathrm{KL}(\phi(S(x)/\tau)\|\phi(T(x)/\tau)), \tag{1}$$

where $\phi(\cdot)$ refers to the softmax function, and τ is included as a temperature hyperparameter to provide additional control over signal softening from the output of the teacher model.

The distillation loss and the original classification loss (i.e., the cross-entropy loss) over the ground-truth label y are used to update the parameters of the student:

$$\mathcal{L}_{CE} = -y \log \phi(S(x)), \tag{2}$$

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{CE} + \lambda\mathcal{L}_{KL}, \tag{3}$$

where λ is the hyperparameter that regulates the trade-off between the two losses. It should be noted that training of the original distillation is performed indiscriminately on all instances based on the given objectives and the weights corresponding to different objectives. However, it is unreasonable to ignore the evolution of the student model during the training process. This motivates us to explore an efficient distillation framework from three aspects: data, model, and objective, which will improve the learning efficiency of the student model.

3.2 Data Selection (DS)

In response to the first research question, we explore which data is more beneficial to the performance of the student model. The student becomes stronger as

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

the distillation progresses, which easily leads to repeated learning for those instances that the student has mastered. Therefore, selecting informative instances is important to avoid repeated learning by the student.

Formally, given N instances in a batch, $P(y|x) = \phi(S(x))$ represents the output class probability distribution of the student across the class label y for each instance x. The scaled entropy of the probability distribution is used to compute the uncertainty score U_x for x, and U_x is calculated as:

$$U_x = \frac{u_x}{\log |Y|}, \tag{4}$$

$$u_x = -\sum_{y=1}^{|Y|} P(y \mid x) \log P(y \mid x), \tag{5}$$

where Y is the number of labeled classes. We rank the instances in a batch based on their prediction uncertainty and pick just the top $N \times r$ instances to query the teacher model. Here, $r \in (0, 1]$ refers to the selection ratio that controls the number of instances to query. The selected instances have high uncertainty scores, indicating that they are informative instances that the student should learn from.

3.3 Teacher Assistant (TA)

In this part, we respond to the second research question from the model level. The capacity gap between the teacher and the student is an inherent issue. Our solution is to introduce the teacher assistant model and dynamically query the teacher or the teacher assistant according to the evolution of the student's competency during the training process. The core idea behind this is to empower the student to adjust the learning process based on its current state.

We assume that the student can rely on the teacher assistant during the initial training stage and turn to the teacher for more accurate supervision signals as the student becomes stronger. More specifically, we sort the instances in a batch in accordance with the prediction confidence of the student model. The confidence C_x is measured by entropy, as:

$$C_x = Entropy(\phi(S(x))). \tag{6}$$

Here, the higher the confidence C_x is, the more uncertain the student is. Therefore, we can evenly divide instances into the certain and uncertain ones for the student. For the certain part, the student learns the supervision signals from the teacher, while the teacher assistant provides soft labels for the instances about which the student is uncertain. The loss function is determined as:

$$\mathcal{L}_{TA} = \mathcal{L}_{KL}^{T} + \mathcal{L}_{KL}^{A}, \tag{7}$$

where $\mathcal{L}_{KL}^{T}$ refers to the KL divergence distance between the student and the teacher, and $\mathcal{L}_{KL}^{A}$ denotes the KL divergence distance between the student and the teacher assistant.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

Algorithm 1 Training of the Student Model

Input: Training data $\mathcal{D}$, number of epochs E, set of parameters Ω needed to train the student model

Output: An optimized student

for *epoch* $e = 1$ to E **do**
  for *each batch* $\mathcal{D}_b \in \mathcal{D}$ **do**
    Select informative instances via the uncertainty score: $U_x = \frac{u_x}{\log |Y|}$.
    Divide the instances into two parts: the certainty part and the uncertainty part by confidence: $C_x = Entropy(\phi(S(x)))$.
    Loss $\mathcal{L}_{TA}$ becomes: $\mathcal{L}_{TA} = \mathcal{L}_{KL}^{T} + \mathcal{L}_{KL}^{A}$.
    Compute the entropy of the hidden representations H: $R_x = \frac{Entropy(\phi(H(x)))}{\log |Y|}$.
    Select the M layers via the R_x value to align the teacher.
    Update parameters Ω by: $\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{TA} + \lambda_3 \mathcal{L}_{DA}$.
  end for
end for

3.4 Dynamic Objective Alignment (DOA)

We finally deal with the second research question from the objective level. Inspired by the previous studies [13] on the alignment of the intermediate layers between the teacher and the student, we further investigate dynamic objective alignment to boost the distillation performance. According to the previous studies, the corresponding aligned objective weights are determined by hyperparametric search and remain constant during the training. To address the aforementioned issue, we choose the informative layers based on the entropy calculation of the corresponding layer representations to dynamically align the teacher, thus preventing unnecessary alignment.

We first compute the entropy R_x of the hidden representations H of the teacher:

$$R_x = \frac{Entropy(\phi(H(x)))}{\log |Y|}, \tag{8}$$

the greater the value of R_x is, the more informative the hidden representations of this layer are. We sort the R_x of each layer from large to small and select M layers with higher R_x values. Then, the loss of dynamic objective alignment becomes

$$\mathcal{L}_{DA} = \sum_{i=1}^{M} \left\| \frac{\mathbf{h}_i^s}{\|\mathbf{h}_i^s\|_2} - \frac{\mathbf{h}_{I(j)}^t}{\left\|\mathbf{h}_{I(j)}^t\right\|_2} \right\|_2^2, \tag{9}$$

where M represents the number of layers in the student model, $I(j)$ denotes that the i-th layer of the student is aligned with the j-th layer of the teacher; $\mathbf{h}^s$ and $\mathbf{h}^t$ are hidden representations of the student and the teacher, respectively.

3.5 Total Loss

Finally, the total loss is determined as follows:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{TA} + \lambda_3 \mathcal{L}_{DA}, \tag{10}$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

where λ_1, λ_2 and λ_3 are hyperparameters for adjusting the loss weight. The overall training process of our DeKD is divided into three steps. At the beginning, we fine-tune BERT on the corresponding downstream task to get the teacher model. We then distill the teacher model to obtain the teacher assistant model. After that, we run Algorithm 1 to produce the final student model.

4 Experiments

4.1 Datasets

We conduct evaluations on eight representative text classification benchmarks. (1) We choose three different NLP tasks: Paraphrase Similarity Matching (PSM), Sentiment Classification (SC), and Natural Language Inference (NLI). For the PSM tasks, we select MRPC and QQP [16]. For the SC tasks, we test on SST-2 [16] and Emotion [11]. For the NLI tasks, we evaluate on QNLI and MNLI [16]. (2) We also add two additional text classification tasks: AG News [26] and IMDb [7]. The statistics of the datasets are shown in Table 2.

Table 2. Statistics of the datasets.

Dataset	**#Train**	**#Dev**	**#Test**
MRPC	3,668	408	1,725
SST-2	67,349	872	1,821
QNLI	104,743	5,463	5,463
MNLI	392,702	9,832	9,847
AG News	120,000	-	7,600
QQP	363,849	40,430	390,965
Emotion	16,000	2,000	2,000
IMDb	20,000	5,000	25,000

4.2 Baselines

We choose seven representative KD methods as baselines. Moreover, we consider these methods with 3 and 6 layers of transformers as the student models. The baselines we compare are as follows:

Vanilla KD [2]: By minimizing the original KL divergence loss, the student model is trained to emulate the soft targets created by the logits of the teacher model.

BERT-PKD [13]: To fully exploit the rich knowledge contained in the deep structure of the teacher model, the patient-KD method enables the student model to patiently learn from the teacher through a multi-layer distillation process.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

DFA [12]: DFA tries to learn a compact student model capable of handling the comprehensive classification issue from multiple trained teacher models, each of which specializes in a different classification problem.

CFL [6]: CFL maps the hidden representations of the teachers into a common space. The student is trained by matching the mapped features to those of the teachers, with supplemental supervision from the logits combination.

UHC [15]: The class sets of teacher models are used by UHC to divide the student logits into subsets. Each subset is trained to mimic the output of the teacher model that corresponds to it.

MUKI [3]: Based on the estimated model uncertainty, MUKI designs a model uncertainty-aware knowledge integration framework. The golden supervision is approximated by either taking the outputs of the most confident teacher or softly integrating different teacher predictions according to their relative importance.

KSM [17]: KSM proposes an actor-critic method for selecting appropriate knowledge transfer at different training steps. This optimization considers the impact of knowledge selection on future training steps.

Table 3. The performance of the student model on the test set of the benchmark datasets. The best results from each group of student models are in bold. We also report the average performance for each task in the "AVG." column.

Method	Student	MRPC	SST-2	QNLI	MNLI	AG News	QQP	Emotion	IMDb	AVG.
BERT-base	-	88.48	92.55	91.32	83.87	94.71	90.96	93.55	89.24	90.59
Vanilla KD	3-layer	75.90	79.66	78.15	71.22	60.11	81.32	53.91	79.25	72.44
BERT-PKD	3-layer	76.55	83.53	80.71	72.31	62.17	83.81	56.56	80.13	74.47
DFA	3-layer	75.47	82.61	79.33	71.01	63.23	80.57	54.63	79.36	73.28
CFL	3-layer	76.57	83.78	81.88	73.13	60.21	84.26	59.38	80.87	75.01
UHC	3-layer	78.57	84.86	82.35	73.76	75.67	84.68	64.72	81.35	78.25
MUKI	3-layer	80.99	85.67	83.05	74.37	88.19	85.81	72.91	82.93	81.74
KSM	3-layer	81.90	85.83	83.62	74.63	90.76	86.12	89.31	83.95	84.52
DeKD (Ours)	3-layer	**83.39**	**86.01**	**84.11**	**74.82**	**93.68**	**86.57**	**92.35**	**84.24**	**85.65**
Vanilla KD	6-layer	80.21	81.37	79.86	72.33	62.58	82.37	55.71	80.33	74.35
BERT-PKD	6-layer	81.42	84.16	81.57	73.67	63.57	84.28	59.23	81.27	76.15
DFA	6-layer	81.12	83.36	80.76	72.81	65.46	82.24	58.62	80.22	75.57
CFL	6-layer	82.53	84.67	82.39	74.63	64.74	85.92	62.17	82.34	77.42
UHC	6-layer	83.72	85.74	83.62	75.67	78.19	86.71	68.16	83.66	80.68
MUKI	6-layer	84.97	86.29	85.16	77.54	90.63	87.35	79.64	84.51	84.51
KSM	6-layer	87.19	89.21	86.09	79.26	92.37	88.93	91.82	85.31	87.52
DeKD (Ours)	6-layer	**87.48**	**89.91**	**87.00**	**80.17**	**94.28**	**89.17**	**93.15**	**86.12**	**88.41**

4.3 Implementation Details

We implement our method based on the HuggingFace transformers library [20]. The results of all experiments are obtained from a single NVIDIA V100 GPU. We first fine-tune the 12-layer BERT-base as the teacher model and distill it to

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

obtain the 6-layer and 9-layer teacher assistant models, respectively. The 6-layer teacher assistant is configured for the 3-layer student, and the 9-layer teacher assistant is configured for the 6-layer student. Then, we fine-tune the 3-layer and 6-layer students via our method to get the final student models, respectively. In our setting, we set the number of fine-tuning epochs to 3, the batch size to 32, the learning rate to 2e-5, and the distillation temperature to 5. Meanwhile, we set the loss equilibrium coefficients λ_1 as 0.5, λ_2 as 0.3, and λ_3 as 0.2. For the data selection rate r, we set it to 0.5. For the objective alignment layers M, we set M to 3 and 6 for 3-layer and 6-layer students, respectively. All experiments are repeated five times, and we report the average results over five runs with different seeds.

4.4 Evaluation Metrics

Following prior work [1], we report the F1 score for MRPC, and we use accuracy as the evaluation metric for other tasks.

4.5 Performance Comparison

Table 3 shows the performance comparison with baselines on the text classification tasks. We draw the following observations from the table:

(1) DeKD performs the best on all the text classification tasks. The reason is that DeKD not only considers efficient learning of the student at the data level, but also further improves the distillation performance from the model and objective levels. In general, the classification accuracy of DeKD is from 0.34% to 19.44% greater than the results of the best competitor. None of these baselines, with the exception of DeKD, can simultaneously stand out across all tasks.

(2) The results of traditional KD (Vanilla KD), intermediate representations-based KD (BERT-PKD), and the distillation methods considering multiple teachers (DFA, CFL, UHC, MUKI, and KSM) are all consistent with our expectations. As the conventional distillation method does not employ any additional supervision signals or intermediate representations, it performs poorly on most tasks. However, the method of combining intermediate layer information exceeds the original distillation method. The reason is that the method based on intermediate representations encourages the student model to extract knowledge from previous layers of the teacher model rather than learning parameters from the last layer of the teacher model. In fact, this demonstrates that the student can gain incremental knowledge by learning multiple intermediate layers.

(3) Compared with KSM, the performances of DFA, CFL, UHC, and MUKI are not satisfactory. Although DFA uses additional features to align objectives, it cannot achieve better results. This is due to the instability of feature-aligned supervision, and teacher features are fine-tuned to specifically target different semantic classes. CFL can learn a multitalented and lightweight student model capable of mastering the integration knowledge of heterogeneous teachers, but it suffers from the same issue as DFA in that the supervision based on feature alignments is unstable. The performance of UHC exceeds that of DFA and CFL,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

but is inferior to MUKI and KSM. This demonstrates a potential supervision conflict as UHC matches the student output independently to that of teachers, thus limiting its generalizability across datasets. To summarize, the above results demonstrate that simply combining different supervision signals is ineffective. However, our DeKD designs a more efficient framework that significantly improves the distillation performance of the student model by selecting informative data, assisting the student's evolution with the teacher assistant, and dynamically selecting alignment signals.

Table 4. Ablation study on MRPC, SST-2, and IMDb tasks.

Method	MRPC	SST-2	IMDb
DeKD	87.48	89.91	86.12
w/o DS	86.84	87.84	85.52
w/o TA	86.74	87.61	85.50
w/o DOA	86.33	87.38	85.18

4.6 Ablation Study

In order to evaluate the contributions of different parts of DeKD, we design the ablation experiments. Experimental results are shown in Table 4. Due to space constraints, we show only the results on the MRPC, SST-2, and IMDb datasets. Other results are similar, so we omit them. We design three different configurations: w/o DS, w/o TA, and w/o DOA.

w/o DS. This configuration removes the entropy-based selection strategy. Compared with DeKD, the overall performance drops without DS, implying that selecting informative instances can reduce repetitive learning caused by data redundancy. This entropy-based selection strategy is capable of making better use of limited queries.

w/o TA. Without TA, the model performance declines on the three tasks. This is because the auxiliary model, i.e., the teacher assistant, becomes a boosting factor in the evolution of the student model. Therefore, the performance of this configuration is inferior to that of DeKD, demonstrating that the addition of the teacher assistant can bridge the capacity gap between the student and the teacher.

w/o DOA. When dynamic objective alignment is not taken into account, this configuration performs worse than DeKD. This implies that learning via dynamically aligning middle representations can help the student quickly understand tasks, thus improving prediction confidence.

To summarize, DeKD is superior to the first three configurations, which shows that all the components together can improve the performance of the student model.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

Table 5. The mean, standard deviation, and statistically significant T-test (p-Value) of five different runs on SST-2 and IMDb. The superscripts 1, 2, and 3 respectively denote statistically significant improvements over UHC, MUKI, and KSM.

	SST-2	**Mean**	**Stdev**
UHC	85.74, 83.61, 84.22, 82.37, 85.69	84.33	1.281
MUKI	86.29, 82.16, 85.34, 84.68, 86.12	84.92	1.495
KSM	89.21, 89.01, 88.36, 89.18, 88.27	88.81	0.408
Ours^{1,2,3}	89.91, 89.88, 89.65, 89.17, 88.96	89.51	0.383
	IMDb	**Mean**	**Stdev**
UHC	83.66, 81.21, 83.38, 82.93, 82.07	82.65	0.899
MUKI	84.51, 83.23, 84.47, 83.13, 82.98	83.66	0.679
KSM	85.31, 85.26, 84.93, 85.28, 84.97	85.15	0.165
Ours^{1,2,3}	86.12, 86.08, 86.02, 85.85, 85.93	86.00	0.099

4.7 Model Analysis

Variance Analysis. Taking the 6-layer student setup as an example, we carry out five experiments with different seeds and calculate their mean and standard deviation (Stdev). Moreover, we also conduct a two-sided statistically significant t-test (p-value) with a threshold of 0.05 and compare the baseline methods with our DeKD method. We report the experimental results in Table 5. As shown, our method is statistically significant compared to baselines.

Data Selection Strategies. In addition to the scaled entropy-based (SE) selection strategy, we also implement three other common strategies to compute the uncertainty score U_x for each instance x:

Random, which randomly selects $N \times r$ instances as the baseline to evaluate the effectiveness of selection strategies.

Least-Confidence (LC), which indicates the uncertainty of the model to the predicted class $\hat{y} = \arg\max_y P(y \mid x)$:

$$U_x = 1 - P(\hat{y} \mid x). \tag{11}$$

Margin, which is calculated as the margin between the first and second most probable classes, y_1^* and y_2^*:

$$U_x = P\left(y_1^* \mid x\right) - P\left(y_2^* \mid x\right). \tag{12}$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

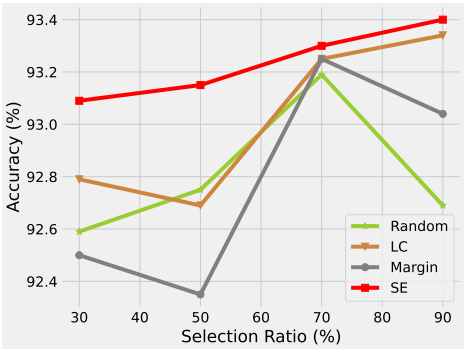
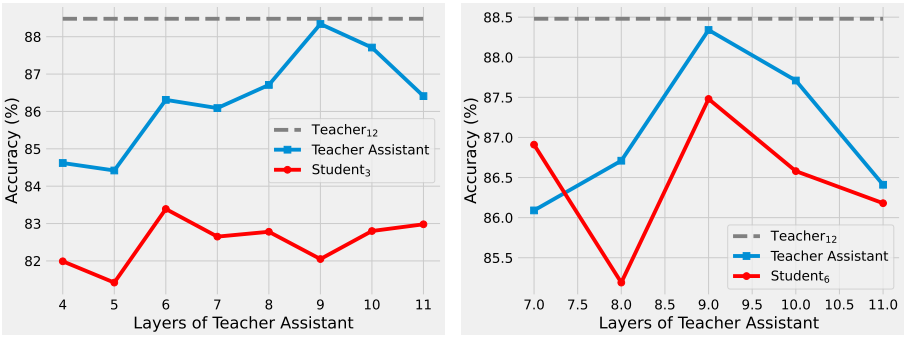


Fig. 2. The average accuracy of five experiments with four strategies under different selection ratios.



(a) *Student*$_3$ (b) *Student*$_6$

Fig. 3. The performance of the student models with 3 and 6 layers under different configurations of teacher assistants.

As shown in Fig. 2, we change the selection ratio r to check the results of different strategies on the Emotion dataset. The results on all datasets show a consistent trend, so one of them is selected for analysis.

From Fig. 2, we can observe that: (1) The selection strategy based on the entropy of student prediction can make better use of limited queries, which is better than other strategies. (2) We can use approximately 50% of the training data to achieve satisfactory performance. This shows that about 50% of the data can cover the training data well, so learning from these instances can sufficiently train the student model. It is helpful to choose informative instances for reducing repetitive learning caused by data redundancy. (3) There is a trade-off between performance and training cost, i.e., increasing the selection ratio usually improves the performance of the student model but leads to a greater training cost.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

Layers of Teacher Assistant. According to the previous work [25], we make two configurations for the student model: 3-layer and 6-layer. We explore the influence of different layers of teacher assistants on the student model under these two configurations. For the MRPC task, the number of layers of teacher assistants ranges from 4 to 11 for the 3-layer student, while the number of layers of teacher assistants ranges from 7 to 11 for the 6-layer student. The teacher assistants are distilled from the 12-layer teacher. In Fig. 3 (a), when the number of teacher assistant layers is 6, the student performs best. In Fig. 3 (b), when the number of teacher assistant layers is 9, the performance of the student is at its peak. It demonstrates that the teacher assistant model, which sits in the middle of the number of layers between the teacher and student models, can solve the problem of the small student not being able to match the capacity of the large teacher.

5 Conclusion

In this paper, we address the issues of repeated learning of instances and the gap between the student and teacher models in knowledge distillation. We put forward an entropy-based selection strategy, and then, through the teacher assistant and dynamic supervision alignment, we can improve the learning efficiency and distillation performance as the student model evolves. Extensive experimental results on the benchmark datasets demonstrate that our proposed method achieves consistent improvements over the state-of-the-art approaches. In the future, we will explore deploying our method on mobile devices for efficient inference.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (No. 2023YFC3303800).

References

1. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (2019)
2. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
3. Li, L., Lin, Y., Ren, X., Zhao, G., Li, P., Zhou, J., Sun, X.: From mimicking to integrating: Knowledge integration for pre-trained language models. In: EMNLP. pp. 6391–6402 (2022)
4. Li, Z., Xu, X., Shen, T., Xu, C., Gu, J.C., Tao, C.: Leveraging large language models for nlg evaluation: A survey. arXiv preprint arXiv:2401.07103 (2024)
5. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
6. Luo, S., Wang, X., Fang, G., Hu, Y., Tao, D., Song, M.: Knowledge amalgamation from heterogeneous networks by common feature learning. In: IJCAI. pp. 3087–3093 (2019)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13

7. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: ACL. pp. 142–150 (2011)
8. OpenAI: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
9. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
10. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
11. Saravia, E., Liu, H.T., Huang, Y., Wu, J., Chen, Y.: CARER: contextualized affect representations for emotion recognition. In: EMNLP. pp. 3687–3697 (2018)
12. Shen, C., Wang, X., Song, J., Sun, L., Song, M.: Amalgamating knowledge towards comprehensive classification. In: AAAI. pp. 3068–3075 (2019)
13. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression. In: ACL/IJCNLP. pp. 4322–4331 (2019)
14. Turc, I., Chang, M., Lee, K., Toutanova, K.: Well-read students learn better: the impact of student initialization on knowledge distillation. arXiv preprint arXiv:1908.08962 (2019)
15. Vongkulbhisal, J., Vinayavekhin, P., Scarzanella, M.V.: Unifying heterogeneous classifiers with distillation. In: CVPR. pp. 3175–3184 (2019)
16. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: ICLR (2019)
17. Wang, C., Lu, Y., Mu, Y., Hu, Y., Xiao, T., Zhu, J.: Improved knowledge distillation for pre-trained language models via knowledge selection. arXiv preprint arXiv:2302.00444 (2023)
18. Wang, W., Bao, H., Huang, S., Dong, L., Wei, F.: Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In: ACL/IJCNLP. pp. 2140–2151 (2021)
19. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In: NeurIPS (2020)
20. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: EMNLP. pp. 38–45 (2020)
21. Xu, G., Liu, Z., Loy, C.C.: Computation-efficient knowledge distillation via uncertainty-aware mixup. Pattern Recognit. **138**, 109338 (2023)
22. Xu, Y., Yuan, F., Cao, C., Su, M., Lu, Y., Liu, Y.: A contrastive self-distillation bert with kernel alignment-based inference. In: ICCS. pp. 553–565. Springer (2023)
23. Xu, Y., Yuan, F., Cao, C., Zhang, X., Su, M., Wang, D., Liu, Y.: Metabert: Collaborative meta-learning for accelerating bert inference. In: CSCWD. pp. 119–124. IEEE (2023)
24. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: NeurIPS. pp. 5754–5764 (2019)
25. Yuan, F., Shou, L., Pei, J., Lin, W., Gong, M., Fu, Y., Jiang, D.: Reinforced multi-teacher selection for knowledge distillation. In: AAAI. pp. 14284–14291 (2021)
26. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NeurIPS. pp. 649–657 (2015)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_13