Adaptive Contrastive Knowledge Distillation for BERT Compression

Jinyang Guo[1,2]*, **Jiaheng Liu**[2]*, **Zining Wang**[2], **Yuqing Ma**[2],
Ruihao Gong[2,3], **Ke Xu**[2] **and Xianglong Liu**[2]†
[1]Institute of Artificial Intelligence, Beihang University
[2]State Key Lab of Software Development Environment, Beihang University
[3]SenseTime Group Limited
{jinyangguo,liujiaheng}@buaa.edu.cn, xlliu@buaa.edu.cn

Abstract

In this paper, we propose a new knowledge distillation approach called adaptive contrastive knowledge distillation (ACKD) for BERT compression. Different from existing knowledge distillation methods for BERT that implicitly learn discriminative student features by mimicking the teacher features, we first introduce a novel contrastive distillation loss (CDL) based on hidden state features in BERT as the explicit supervision to learn discriminative student features. We further observe sentences with similar features may have completely different meanings, which makes them hard to distinguish. Existing methods do not pay sufficient attention to these hard samples with less discriminative features. Therefore, we propose a new strategy called sample adaptive reweighting (SAR) to adaptively pay more attention to these hard samples and strengthen their discrimination abilities. We incorporate our SAR strategy into our CDL and form the adaptive contrastive distillation loss, based on which we construct our ACKD framework. Comprehensive experiments on multiple natural language processing tasks demonstrate the effectiveness of our ACKD framework.

1 Introduction

Recently, deep learning (Liu et al., 2023; Guo et al., 2023; Liu et al., 2021; Guo et al., 2022a) has achieved success in many natural language processing tasks. However, due to limited computation and storage resources, current deep learning approaches are hard to be deployed on mobile devices. Knowledge distillation is an effective approach to compress the model for mobile deployment, which aims to use a pretrained teacher network to help the training of a lightweight student network. To achieve this, the student needs to learn discriminative features. Namely, we need to push the features of the sample from different classes (negative pairs) far away from each other and keep the features of the samples from the same classes (positive pairs) close.

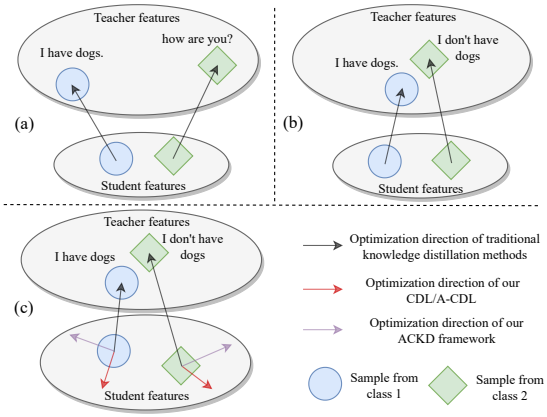


Figure 1: (a) and (b): Traditional knowledge distillation method. Student fails to learn discriminative features when sentences from different classes have similar features in the teacher. (c): Our ACKD framework, which uses explicit supervision to push student features from different classes far away from each other.

Current knowledge distillation methods for BERT implicitly learn discriminative student features. They assume the teacher is well-learned (i.e., features of negative pairs are far away from each other in the teacher). Then, they minimize the feature distance of each sample between the teacher and student to make the student feature discriminative, as shown in Fig. 1(a). In this way, the features of negative pairs in the student can be pulled far away from each other. However, the aforementioned assumption is not always held. Commonly used words will appear in the sentences with different meanings, causing the features of negative pairs in the teacher to be close to each other, as shown in Fig. 1(b). In this case, training the student using the current knowledge distillation paradigm will result in the features of negative pairs in the student being close to each other as well. So, it is desirable to in-

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 8941–8953
July 9-14, 2023

Table 1: Examples of hard samples from GLUE.

Linguistic acceptable	Linguistic unacceptable
Harry coughed himself into a fit.	Harry coughed us into a fit.
This building got taller and taller.	This building is taller and taller.
Bill cried himself to sleep.	Bill cried Sue to sleep.

troduce explicit supervision (e.g., a well-designed loss) to push the features of negative pairs in the student far away from each other.

Another issue in the existing knowledge distillation methods is that they do not pay sufficient attention to hard samples in the distillation process. Similar sentences may have completely different meanings. For example, for the linguistic acceptability task, although the sentences "We yelled ourselves hoarse" and "We yelled Harry hoarse" are similar as they only have one different word, the first sentence is linguistically acceptable while the latter one is not, making them fall into different categories. This makes these sentences hard to distinguish because their features are similar and thus less discriminative. This phenomenon often occurs in other natural language processing tasks, and we provide more examples from GLUE benchmark (Wang et al., 2019) in Table 1. Therefore, it is also desirable to pay more attention to hard samples to strengthen their discrimination abilities.

To solve the aforementioned problems, we propose a new knowledge distillation framework called adaptive contrastive knowledge distillation (ACKD). Specifically, to tackle the first issue (i.e., lack of explicit supervision), we introduce the concept of contrastive learning (Gutmann and Hyvärinen, 2010; Oord et al., 2018; Saunshi et al., 2019; Hjelm et al., 2018) to knowledge distillation and design a contrastive distillation loss (CDL) as the explicit supervision to maximize the distance of the features from negative pairs. In particular, for each sample s, our CDL aims to maximize the similarity between the features of s in the student and that in the teacher, and minimize the similarity between the features of s in student and the features from the negative pairs of s in teacher. As shown in Fig. 1(c), our CDL can effectively push the features from negative pairs far away from each other.

To tackle the second issue (i.e., learning of hard samples), we propose a new strategy called sample adaptive reweighting (SAR) in our ACKD framework to adaptively pay more attention to hard samples to strengthen their discrimination abilities. Specifically, we utilize a neural network as a predictor to predict the discrimination ability of the feature for each sample based on its learned feature. Then, we reweight the loss from different samples according to the predicted discrimination ability. As all operations in this process are differentiable, the parameters of the predictor can be jointly learned with the student. We seamlessly incorporate our SAR strategy into the newly proposed CDL and construct the adaptive contrastive distillation loss (A-CDL).

We combine our A-CDL with the existing knowledge distillation methods and construct our Adaptive Contrastive Knowledge Distillation (ACKD) framework. It is also a non-trivial task to construct our ACKD framework as our A-CDL is calculated based on the features, which can only be calculated inside one mini-batch due to the property of current deep learning frameworks (i.e., features will be released after the calculation of current batch). So, the diversity of negative paired samples is limited by the batch size, causing an inaccurate optimization direction. To overcome this issue, inspired by (He et al., 2020), we construct a dynamic feature storage that can store the features from a large number of samples, based on which we calculate our A-CDL to increase the sample diversity.

In summary, the main contribution of this paper can be summarized as follows:

- We propose a novel contrastive distillation loss (CDL) to introduce explicit supervision for learning discriminative student features.
- We propose a new strategy called sample adaptive reweighting (SAR) strategy to adaptively pay more attention to hard samples and strengthen their discrimination abilities. We seamlessly incorporate our SAR strategy into our CDL and form the adaptive contrastive distillation loss (A-CDL). Based on A-CDL, we construct our new adaptive contrastive knowledge distillation (ACKD) framework for BERT compression, in which dynamic feature storage is used to increase the diversity of samples.
- Comprehensive experiments on multiple natural language processing tasks demonstrate the effectiveness of our ACKD framework.

2 Related Work

Knowledge distillation. Recently, model compression methods (Guo et al., 2020b,a,c, 2021, 2023,

2022b; Wei et al., 2023; Qin et al., 2022, 2023a,c,b; Liu et al., 2022c, 2020, 2022a; Peng et al., 2019) attracts many attentions, among which knowledge distillation approaches (Liu et al., 2022b) were proposed to accelerate deep neural networks (Ma et al., 2022, 2021; Hu et al., 2021). For example, (Hinton et al., 2015) first proposed to use the so-called dark knowledge as the additional supervision for training the student. After this work, many methods (Romero et al., 2015; Zagoruyko and Komodakis, 2017) were proposed to utilize the intermediate feature as the supervision in the distillation process. Another line of work finds knowledge distillation cannot achieve promising performance if there is a large capacity gap between teacher and student. Therefore, this line of works aims to use a sequence of teacher models to better transfer the knowledge to the student, including RCO (Jin et al., 2019) and TAKD (Mirzadeh et al., 2020). However, all of these works do not consider the relationship between different samples (e.g., the correlation between negative pairs), while our ACKD uses the relationship among samples as the explicit supervision to learn more discriminative features.

There are also knowledge distillation approaches (Tian et al., 2019) that utilize the relation between different samples when learning the student, which is more related to our ACKD framework. For example, (Tung and Mori, 2019) proposed to use the similarity of the features from different samples as the knowledge to train the student. (Park et al., 2019) and (Yim et al., 2017) use the mutual relation of different samples as the knowledge for distillation. However, these methods only use the student to mimic the sample relation in the teacher, which also lacks explicit supervision for the student to learn discriminative features. In contrast, our ACKD framework uses the newly proposed A-CDL to explicitly push the features of negative pairs far away from each other. Moreover, these methods do not consider the learning of hard sample problem for natural language processing tasks. In our ACKD, we use the SAR strategy to pay more attention to hard samples.

Knowledge distillation for BERT. Many methods were also proposed for compressing BERT (Devlin et al., 2018; Sanh et al., 2019; Zhou et al., 2022; Haidar et al., 2022; Jafari et al., 2021; Passban et al., 2021). For example, patient knowledge distillation (Sun et al., 2019) proposed to use intermediate features as the supervision to train a small student BERT. TinyBERT (Jiao et al., 2019) uses a two-stage distillation strategy for BERT compression. Although these methods can compress BERT for efficient inference, explicit supervision for learning discriminative student features is not used in these methods. While (Fu et al., 2021) also uses contrastive loss for BERT distillation, they do not use SAR strategy and ignores the sample difficulties. (Sun et al., 2020) proposed CoDIR method to capture structural knowledge in the intermediate layers. Unlike our ACKD framework, these approaches do not consider paying more attention to hard samples.

3 Adaptive Contrastive Knowledge Distillation

In this section, we will introduce our adaptive contrastive distillation (ACKD) framework. The goal of our ACKD framework is to use a pre-trained teacher model with a large capacity to help the training of a lightweight student model, and its overview is shown in Fig. 2. The loss of our ACKD framework when training the student comes from four parts: cross-entropy loss (CEL), knowledge distillation loss (KDL), patient loss (PTL), and our adaptive contrastive distillation loss (A-CDL).

3.1 Preliminary

Patient distillation (Sun et al., 2019) was proposed to compress BERT. Given the training dataset with N samples $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, the student network can be trained by using the loss function as follows:

$$\begin{aligned}\mathcal{L}_{pre} &= \alpha\mathcal{L}_{ce} + (1-\alpha)\mathcal{L}_{kd} + \beta\mathcal{L}_{pt} \\ &= \frac{1}{N}\sum_{i=1}^{N}[\alpha \cdot CE(\mathcal{T}(x_i;\theta^T), y_i) \\ &+ (1-\alpha)\cdot ST(\mathcal{T}(x_i;\theta^T), \mathcal{S}(x_i;\theta^S)) \\ &+ \beta \cdot \sum_{m=1}^{M} MSE(z_i^{T,m}, z_i^{S,m})].\end{aligned} \tag{1}$$

$\mathcal{L}_{ce}$ is the task-specific loss and $CE(\cdot,\cdot)$ is the corresponding loss function, in which cross-entropy is commonly adopted for the classification task. $\mathcal{L}_{kd}$ is the knowledge distillation loss and $ST(\cdot,\cdot)$ denotes the corresponding loss function, in which the Kullback–Leibler divergence of the output probability distribution between the teacher and student is commonly adopted. $\mathcal{L}_{pt}$ is the patient loss introduced in (Sun et al., 2019) and $MSE(\cdot,\cdot)$ is

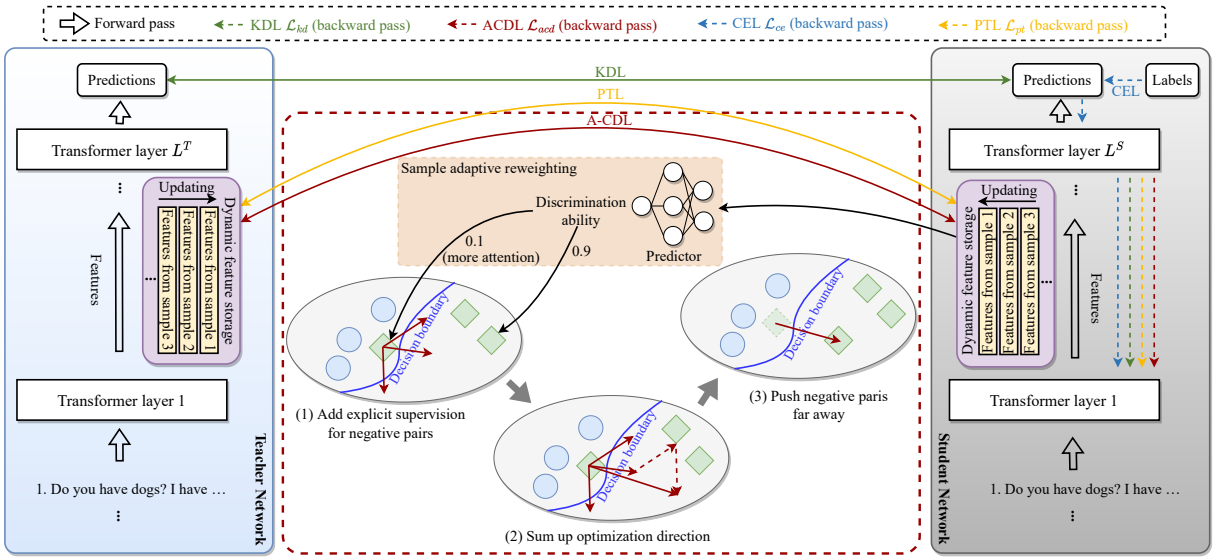


Figure 2: Overview of our ACKD framework for BERT compression, which consists of the losses from four parts: cross-entropy loss (CEL), knowledge distillation loss (KDL), patient loss (PTL), and adaptive contrastive distillation loss (A-CDL). In this figure, the teacher and student have L^T and L^S layers, respectively. Our A-CDL aims to push negative pairs far away from each other, which is calculated based on the hidden state features and the predicted discrimination abilities of different samples. Dynamic feature storage is used to increase the diversity of samples.

the mean square error function. $\mathcal{T}$ and $\mathcal{S}$ are the teacher and student networks, and their parameters are denoted as θ^T and θ^S, respectively. $z_i^{T,m}$ and $z_i^{S,m}$ denote the hidden state feature from the teacher and the student for the i-th sample at the m-th paired layers when calculating the patient loss, respectively. M is the number of layers that the patient loss is inserted. α and β are the hyperparameters to control the trade-off of different terms. The loss $\mathcal{L}_{ce}$, $\mathcal{L}_{kd}$, and $\mathcal{L}_{pt}$ correspond to the CEL, KDL, and PTL in Fig. 2, respectively.

3.2 Contrastive Distillation Loss

Although the loss in Eq. (1) can transfer the knowledge from teacher to student, it lacks explicit supervision to learn discriminative student features. Namely, it only provides the supervision to pull the features from the same sample in teacher and student close to each other, while lacking the supervision to push the features from different classes far away from each other for more discriminative feature learning (Harwood et al., 2017; Wu et al., 2017; Suh et al., 2019). To this end, we first design our contrastive distillation loss (CDL) in the ACKD framework.

As our CDL can be introduced at different layers, below, we only focus on the m-th paired layer and omit the layer index for better presentation. For example, we use z_i^T and z_i^S to denote the hidden state features for the i-th sample at this layer in teacher and student, respectively. The CDL can be written as follows:

$$\begin{aligned} \mathcal{L}_{cd} &= -log \sum_{i=1}^{N} \frac{POS}{POS + NEG}, \\ \text{where } POS &= exp(h(z_i^S, z_i^T)), \\ NEG &= \sum\nolimits_{z_j^T \in \mathcal{N}_i} exp(h(z_i^S, z_j^T)), \\ h(z_i^S, z_j^T) &= cosine(z_i^S, z_j^T). \end{aligned} \tag{2}$$

Here, $cosine(\cdot, \cdot)$ denotes the cosine similarity. $\mathcal{N}_i$ denotes the set containing the hidden state features of the samples from different classes with the i-th sample (i.e., negative pair).

3.3 Sample Adaptive Reweighting

As mentioned in Sec. 1, similar sentences may have completely different meanings, which makes these samples hard to distinguish. To this end, we propose our sample adaptive reweighting (SAR) strategy to adaptively pay more attention to these hard samples. Specifically, we use a predictor network to predict the discrimination ability of each sample based on its learned features, and incorporate this predicted discrimination into our CDL to form adaptive contrastive distillation loss (A-CDL). For-

mally, the A-CDL can be written as follows:

$$\mathcal{L}_{acd} = -\,log \sum_{i=1}^{N} \frac{POS}{POS + \overline{NEG}},$$
$$\text{where } \overline{NEG} = \frac{1}{w_i} \sum\nolimits_{z_j^T \in \mathcal{N}_i} exp(h(z_i^S, z_j^T)),$$
$$w_i = Sigmoid(\mathcal{P}(z_i^S; \theta_p)). \tag{3}$$

Here, w_i is the predicted discrimination ability of the i-th sample. $\mathcal{P}(\cdot,\cdot)$ is the function of the predictor, which is implemented by a neural network. θ_p is the learnable parameter of the predictor. $Sigmoid(\cdot)$ is the sigmoid function, which is used to ensure the predicted discrimination abilities are positive. The other notations are the same as before. As all operations are differentiable in this process, we can jointly train this predictor with the student network in distillation. In this way, we can adaptively assign higher weight $\frac{1}{w_i}$ on the samples with less discriminative features and finally form the adaptive contrastive distillation loss, which corresponds to A-CDL in Fig. 2. Note that our predictor is implemented by a simple neural network. Therefore, the extra computation caused by the predictor can be neglected compared with that required by the gradient calculation.

3.4 Overall Loss Function

As our A-CDL can be introduced to different paired layers of the teacher and student networks, for better presentation, below, we additionally use the superscript $\cdot^m$ to denote the corresponding symbols for the m-th paired layers that A-CDL is inserted. So, the loss function when training the student network in our ACKD framework can be written as:

$$\begin{aligned}\mathcal{L}_{total} &= \alpha\mathcal{L}_{ce} + (1-\alpha)\mathcal{L}_{kd} + \beta\mathcal{L}_{pt} + \gamma\mathcal{L}_{acd} \\ &= \frac{1}{N}\sum\nolimits_{i=1}^{N}[\alpha \cdot CE(\mathcal{T}(x_i;\theta^T), y_i) \\ &\quad + (1-\alpha)\cdot ST(\mathcal{T}(x_i;\theta^T), \mathcal{S}(x_i;\theta^S)) \\ &\quad + \beta \cdot \sum\nolimits_{m=1}^{M} MSE(z_i^{T,m}, z_i^{S,m}) \\ &\quad + \gamma \cdot \sum\nolimits_{m=1}^{M} -log\frac{POS}{POS + \overline{NEG}}], \\ \text{where } POS &= exp(h(z_i^{S,m}, z_i^{T,m})), \\ \overline{NEG} &= \frac{1}{w_i^m} \sum_{z_j^{T,m} \in \mathcal{N}_i} exp(h(z_i^{S,m}, z_j^{T,m})).\end{aligned} \tag{4}$$

α, β, and γ are the hyperparameters to control the importance of different terms. $\mathcal{L}_{ce}$, $\mathcal{L}_{kd}$, and $\mathcal{L}_{pt}$ are the cross-entropy loss, the knowledge distillation loss, and the patient loss, respectively, which are introduced in Eq. (1). $\mathcal{L}_{acd}$ is our newly proposed adaptive contrastive distillation loss introduced in Eq. (3). Other notations are the same as before. By using the loss introduced in Eq. (4), we can use explicit supervision to push the features of negative pairs in the student far away from each other, with the consideration of the sample discrimination abilities. In this way, we construct our ACKD framework for BERT compression.

3.5 Dynamic Feature Storage

When introducing the A-CDL to the existing knowledge distillation methods and constructing our ACKD framework, another issue is that A-CDL requires large sample diversity, which is not required in the existing knowledge distillation approaches, making the construction of our ACKD framework a non-trivial task. Specifically, the term $\overline{NEG}$ is calculated based on the features of different samples. Due to the property of the current deep learning framework, features will be released after the calculation of each mini-batch. Therefore, we can only calculate $\overline{NEG}$ based on the samples in one mini-batch. So, the feature of the i-th sample can be only pushed far away from those of a small portion of negative pairs, which causes inaccurate optimization direction. Inspired by (He et al., 2020), we construct dynamic feature storage to increase the sample diversity. Specifically, after the calculation of each batch, we store the features of this batch in the storage for $\overline{NEG}$ calculation. At the same time, labels of these samples will be also stored in the storage for identifying the samples in $\mathcal{N}_i$. As the BERT model processes a sequence of tokens in parallel, the feature dimension is relatively large, which causes more memory burden to GPU. Therefore, to further save memory usage, we only store the features of the layer that A-CDL is inserted. After the storage is full, we update storage based on the first in first out strategy. In our implementation, we set the storage size as 1000. In this way, we increase sample diversity when calculating $\overline{NEG}$.

3.6 Discussion

The design concept of our A-CDL is as follows. In the distillation process, the loss $\mathcal{L}_{acd}$ will be minimized. To achieve this, we will maximize the value inside the $-log(\cdot)$ function. So in the training process, the numerator POS will be increased, which pulls the feature from the same sample in

teacher and student close to each other. At the same time, the denominator term $\overline{NEG}$ will be decreased, which pushes the feature of the j-th sample from different classes in the student far away from that of the i-th sample in the teacher. Moreover, by using discrimination ability $\frac{1}{w_i}$, we assign higher weights to the samples with less discriminative features. In this way, we introduce explicit supervision with the consideration of sample discrimination abilities to learn more discriminative student features.

From another point of view, our A-CDL can also be viewed as the loss to "eliminate" the influence of incorrect predictions from the teacher when learning the student. Specifically, as in Fig. 1(b), if the green sample is close to the blue one and is misclassified by the teacher, traditional knowledge distillation methods will not be aware of this misclassification. So the green sample in the student will be "attracted" by that in the teacher (black arrow), causing misclassification in the student as well. In contrast, from Eq. (3), the negative pair set $\mathcal{N}_i$ when calculating $\overline{NEG}$ is obtained based on the ground truth labels. Therefore, as in Fig. 1(c), despite the green sample being misclassified by the teacher, the green sample in the student will be "repelled" by the blue sample in the teacher (red arrow). Although the cross-entropy loss for student is also based on the ground truth labels, the optimization direction will be affected by the incorrect teacher prediction. So our A-CDL can "eliminate" the influence of incorrect predictions from teacher to some extent.

4 Experiments

In this section, we perform comprehensive experiments and extensive ablation studies.

4.1 Datasets

We follow many works (Sun et al., 2019; Zhou et al., 2022) to evaluate our ACKD framework on the GLUE benchmark (Wang et al., 2019). Specifically, we use the development set of the GLUE benchmark and use four tasks for evaluation: Paraphrase Similarity Matching, Sentiment Classification, Natural Language Inference, and Linguistic Acceptability. For Paraphrase Similarity Matching, we use MRPC (Dolan and Brockett, 2005), QQP, and STS-B (Conneau and Kiela, 2018) for evaluation. For Sentiment Classification, we use SST-2 (Socher et al., 2013) for evaluation. For Natural Language Inference, we use MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), and RTE (Wang et al., 2019) for evaluation. For Linguistic Acceptability, we use CoLA (Warstadt et al., 2019) for evaluation.

Following many works (Sun et al., 2019; Zhou et al., 2022), we report the results on MNLI-m and MNLI-mm on MNLI. For MRPC and QQP, we report both F1 and accuracy. For STS-B, we report Pearson and Spearman correlation. For CoLA, we report Matthew's correlation. We use accuracy as the metric for other datasets.

4.2 Implementation Details

We implement our ACKD framework based on the PyTorch framework. We follow previous works (Sun et al., 2019; Zhou et al., 2022) to evaluate our ACKD under the task-specific setting, in which the teacher network is firstly fine-tuned on downstream tasks and the student network is also trained based on the downstream tasks in the distillation process. Following (Sun et al., 2019), we use the BERT-Base model as the teacher network, and use BERT with 3 and 6 layers as the student models (denoted as BERT_3 and BERT_6), respectively. The number of hidden states is set as 768 in both teacher and student networks. We follow (Sun et al., 2019) to assume the lower layers of the teacher network also contain important information and should be passed to the student. Therefore, we choose the "skip" strategy in (Sun et al., 2019) to insert our A-CDL, which can bring stronger supervision.

We first finetune the pre-trained BERT-Base model on downstream tasks as the corresponding teacher models. The maximum sequence length is set as 128, and AdamW (Loshchilov and Hutter, 2018) optimizer is adopted. We set the initial learning rate and batch size as $2e^{-5}$ and 8, respectively. The training epoch ranges from 2 to 4 for different downstream tasks. Then, we train our student network by using our ACKD framework. The discrimination predictor for generating w_i in Eq. (3) is implemented by a two-layer neural network. The size of dynamic feature storage is set as 1000. We follow (Sun et al., 2019; Zhou et al., 2022) to perform hyperparameter search over student learning rate from $\{1e^{-5}, 2e^{-5}, 5e^{-5}\}$, the batch size from {8, 16, 32}, the hyperparameter α from {0.1, 0.3, 0.5}, β from {20, 40, 60}, and γ from $\{5e^{-4}, 5e^{-3}, 5e^{-2}\}$. The other hyperparameters are the same as those when training the teacher network.

Table 2: Performance comparison of different methods on the dev set of the GLUE benchmark. CoDIR uses RoBERTa-base as the teacher, and we report the median performance of this method copied from published paper. The results in **bold** indicate the best result, while the results underlined indicate the second-best result.

Method	#Param	Speed-up	GLUE							
			CoLA (Matt.)	MNLI (Acc -m/-mm)	MRPC (F1/Acc)	QNLI (Acc)	QQP (F1/Acc)	RTE (Acc.)	SST-2 (Acc.)	STS-B (Pear./Spear.)
Teacher Network: BERT-Base										
BERT-Base (Devlin et al., 2018)	110M	1.0×	60.8	84.6/84.4	91.6/87.6	91.6	88.5/91.4	71.4	93.0	90.2/89.8
Student Network: $BERT_3$										
PKD (Sun et al., 2019)	46M	4.0×	39.8	75.9/76.6	84.1/75.0	84.3	85.3/89.2	62.8	87.4	86.3/86.1
RCO (Jin et al., 2019)	46M	4.0×	31.4	76.3/76.9	85.3/77.5	83.4	85.4/88.7	66.1	86.8	84.8/84.4
TAKD (Mirzadeh et al., 2020)	46M	4.0×	35.7	76.2/76.8	83.2/73.5	83.8	83.7/87.5	59.2	87.9	83.8/83.4
DistilBERT (Sanh et al., 2019)	46M	4.0×	34.0	77.0/77.0	83.2/73.0	83.8	85.1/88.9	62.8	86.9	86.6/86.2
TinyBERT (Jiao et al., 2019)	46M	4.0×	38.7	76.5/76.9	82.8/72.8	84.2	85.1/88.8	60.6	86.8	86.4/86.1
CRD (Tian et al., 2019)	46M	4.0×	38.6	76.1/76.8	85.2/77.5	84.6	83.9/88.0	65.7	87.6	86.1/85.6
SFTN (Park et al., 2021)	46M	4.0×	38.1	76.6/77.1	83.1/73.3	84.2	83.9/87.7	60.3	88.0	83.9/83.5
MetaDistill (Zhou et al., 2022)	46M	4.0×	39.3	75.9/76.4	82.0/71.1	83.8	83.7/88.1	62.1	88.0	86.6/86.4
Annealing KD (Jafari et al., 2021)	52M	3.0×	36.0	73.9/74.8	86.2/-	83.1	-/86.5	61.0	**89.4**	74.5/-
ACKD (ours)	46M	4.0×	**42.7**	**79.5/80.6**	**87.5/81.4**	**86.2**	**86.1/89.7**	**67.9**	88.5	**87.1/86.8**
Student Network: $BERT_6$										
PKD (Sun et al., 2019)	66M	2.0×	54.5	82.7/83.3	89.4/84.7	89.5	87.8/90.9	67.6	91.3	88.6/88.1
RCO (Jin et al., 2019)	66M	2.0×	53.6	82.4/82.9	89.5/85.1	89.7	87.4/90.6	67.6	91.4	88.7/88.3
TAKD (Mirzadeh et al., 2020)	66M	2.0×	53.8	82.5/83.0	89.6/85.0	89.6	87.5/90.7	68.5	91.4	88.2/88.0
DistilBERT (Sanh et al., 2019)	66M	2.0×	53.0	82.5/83.1	89.3/85.0	89.2	87.2/90.6	66.1	91.5	88.7/88.5
TinyBERT (Jiao et al., 2019)	66M	2.0×	52.4	83.6/83.8	90.5/86.5	89.8	87.6/90.6	67.7	91.9	89.2/88.7
CRD (Tian et al., 2019)	66M	2.0×	55.8	83.2/83.4	89.5/85.5	89.8	87.6/90.8	67.1	91.5	88.8/88.3
SFTN (Park et al., 2021)	66M	2.0×	53.6	82.4/82.9	89.8/85.3	89.5	87.5/90.4	68.5	91.5	88.4/88.5
MetaDistill (Zhou et al., 2022)	66M	2.0×	58.6	83.5/83.8	**91.1**/86.8	90.4	88.1/91.0	69.4	92.3	89.4/**89.1**
ALP-KD (Passban et al., 2021)	66M	2.0×	46.4	82.0/-	-/85.8	89.7	-/90.6	69.0	91.9	88.8/-
CoDIR (Sun et al., 2020)	66M	2.0×	56.4	**83.9**/-	87.9/-	**90.7**	-/91.2	66.3	**92.4**	-/-
ACKD (ours)	66M	2.0×	**59.7**	83.6/**83.9**	91.0/**87.0**	90.6	**88.5/91.3**	**69.7**	92.3	**89.5/89.1**

4.3 Experimental Results

We compare our ACKD framework with multiple state-of-the-art approaches including: PKD (Sun et al., 2019), RCO (Jin et al., 2019), TAKD (Mirzadeh et al., 2020), DistilBERT (Sanh et al., 2019), TinyBERT (Jiao et al., 2019), CRD (Tian et al., 2019), SFTN (Park et al., 2021), MetaDistill (Zhou et al., 2022), Annealing KD (Jafari et al., 2021), ALP-KD (Passban et al., 2021), and CoDIR (Sun et al., 2020).

The results are shown in Table 2. From Table 2, we have following observations: (1) Our ACKD framework outperforms other baseline methods when using $BERT_3$ and $BERT_6$ as the students under most of settings, which demonstrates the effectiveness of the proposed ACKD framework. Specifically, when using $BERT_3$ as the student, our ACKD framework surpasses other baseline methods by more than 2.9% on CoLA. (2) When using $BERT_3$ as the student, our ACKD framework can achieve higher performance gain. One possible explanation is that the performance of the distilled $BERT_6$ is close to the teacher network BERT-Base, which is the bottleneck for further performance improvement. Also, $BERT_3$ has less knowledge than $BERT_6$. Therefore, our A-CDL as new knowledge can bring more information gain for $BERT_3$ and thus bring more performance improvement.

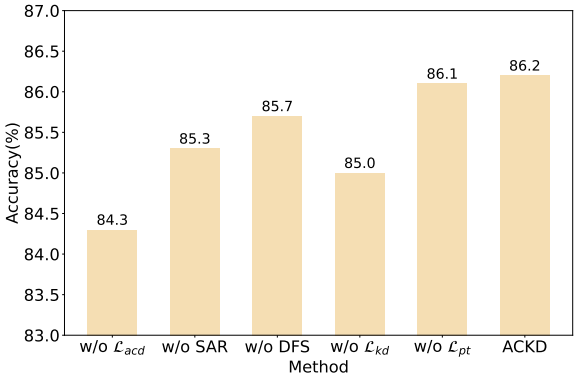
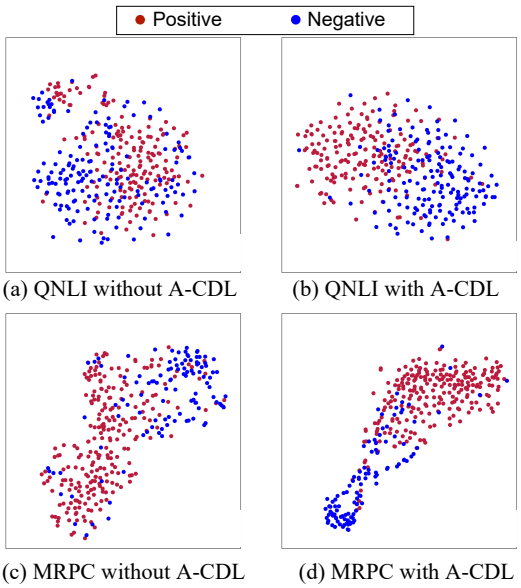


Figure 3: Performance of our ACKD framework and other alternative methods on QNLI.

4.4 Ablation Study

In this section, we perform extensive ablation studies. We use BERT-Base as the teacher network and use $BERT_3$ as the student network to conduct the experiment on QNLI (Rajpurkar et al., 2016).

Effectiveness of $\mathcal{L}_{acd}$ in Eq. (4). To investigate the effectiveness of the A-CDL, we remove the $\mathcal{L}_{acd}$ in Eq. (4) and conduct the distillation. The re-



(a) QNLI without A-CDL (b) QNLI with A-CDL

(c) MRPC without A-CDL (d) MRPC with A-CDL

Figure 4: The t-SNE visualization of student features trained without ((a) and (c)) and with ((b) and (d)) using our A-CDL.

sult is denoted as "w/o $\mathcal{L}_{acd}$" in Fig. 3. Our ACKD method outperforms the alternative approach "w/o $\mathcal{L}_{acd}$" by a large margin, demonstrating the effectiveness of our A-CDL for explicit supervision to push student features of negative pairs far away from each other.

Effectiveness of our sample adaptive reweighting strategy. To investigate the effectiveness of our SAR strategy, we perform the experiment to remove the $\frac{1}{w_i}$ in Eq. (3) and conduct the distillation. In this case, we use CDL instead of A-CDL in distillation. The result is denoted as "w/o SAR" in Fig. 3. From the result, we observe that our ACKD approach performs better than the alternative method "w/o SAR", which demonstrates the effectiveness of the SAR strategy to pay more attention to less discriminative samples.

Effectiveness of dynamic feature storage. We investigate the effectiveness of using dynamic feature storage (DFS) in our ACKD framework. We perform the experiment to remove the DFS, and the result is denoted as "w/o DFS" in Fig. 3. Our ACKD framework performs better than "w/o DFS", demonstrating the effectiveness of using dynamic feature storage.

Effectiveness of $\mathcal{L}_{kd}$ and $\mathcal{L}_{pt}$ in Eq. (3). We also report the results when removing the $\mathcal{L}_{kd}$ and $\mathcal{L}_{pt}$ in Eq. (4), which are denoted as "w/o $\mathcal{L}_{kd}$" and "w/o $\mathcal{L}_{pt}$" in Fig. 3, respectively. From the results, we observe: (1) The performance of our ACKD framework is better than the methods "w/o $\mathcal{L}_{kd}$" and "w/o $\mathcal{L}_{pt}$". This suggests it is beneficial to use $\mathcal{L}_{kd}$ and $\mathcal{L}_{pt}$. (2) The accuracy of "w/o $\mathcal{L}_{pt}$" is higher than "w/o $\mathcal{L}_{kd}$", which indicates the loss $\mathcal{L}_{kd}$ is more useful than $\mathcal{L}_{pt}$ in our ACKD framework when compressing BERT.

Table 3: Performance of ACKD framework when using different teacher network structures. BERT_l means the BERT model with l layers.

Teacher	BERT_{12}	BERT_{10}	BERT_8	BERT_6
Student (BERT_3)	86.2	86.1	85.8	85.5

4.5 Algorithm Analysis

In this section, we also use BERT-Base as the teacher and use BERT_3 as the student to conduct the experiments on algorithm analysis. We perform the experiments on QNLI (Rajpurkar et al., 2016).

Analysis on the structure of teacher network. In Table 3, we also report the results when using different teacher networks. We observe that we can effectively train the student when using different teacher network structures.

Analysis on the structure of predictor. In our ACKD framework, we use a two-layer neural network as our predictor to predict the discrimination ability of each sample. We also investigate the performance of our ACKD framework when using different predictor structures. When using BERT-Base as the teacher and using BERT_3 as the student, the accuracy of our ACKD framework with two, three, and four layers of predictor are 86.2%, 86.4%, and 86.2% on QNLI, respectively. We observe that the performance of our ACKD using different predictor structures is relatively stable.

4.6 Visualization

To demonstrate the effectiveness of the proposed A-CDL, we visualize the learned student feature without and with using our A-CDL. Specifically, Fig. 4 visualize the student feature trained without and with using A-CDL (i.e., $\mathcal{L}_{acd}$ in Eq. (3)) on QNLI and MRPC by using the t-SNE (Van der Maaten and Hinton, 2008) technique. From Fig. 4, we observe that after introducing our A-CDL, the student features from different classes become far away from each other, which demonstrates the effectiveness of our A-CDL.

5 Conclusion

In this paper, we have proposed a new knowledge distillation approach called adaptive contrastive knowledge distillation (ACKD) for BERT compression. We first introduce a novel contrastive distillation loss (CDL) as the explicit supervision to learn more discriminative student features. Then, we propose a new strategy called sample adaptive reweighting (SAR) to adaptively pay more attention to hard samples with fewer discrimination abilities. The SAR strategy can be seamlessly incorporated into the CDL and form the adaptive contrastive distillation loss (A-CDL). Based on A-CDL, we construct our ACKD framework, where dynamic feature storage is used for better sample diversity. Extensive experiments on multiple natural language processing tasks demonstrate the effectiveness of our ACKD framework for BERT compression.

6 Limitation

One of the limitations of our framework is we need to design the rough range of hyperparameters to search the best setting. In our future work, we will explore the strategy to avoid hyperparameter tuning.

7 Ethical Consideration

Our adaptive contrastive knowledge distillation framework aims to improve the performance of knowledge distillation methods and does not introduce extra ethical concerns compared with other knowledge distillation approaches. Therefore, there are no ethical problems caused by the proposed method.

Acknowledgements

We sincerely thank the anonymous reviewers for their serious reviews and valuable suggestions. This work was supported by The National Key Research and Development Plan of China (2021ZD0110503), National Natural Science Foundation of China (62022009), National Natural Science Foundation of China (62206010), and National Natural Science Foundation of China (61932002).

References

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *LREC*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Lrc-bert: latent-representation contrastive knowledge distillation for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hongcheng Guo, Jiaheng Liu, Haoyang Huang, Jian Yang, Zhoujun Li, Dongdong Zhang, and Zheng Cui. 2022a. LVP-M3: Language-aware visual prompt for multilingual multimodal machine translation. In *EMNLP 2022*, pages 2862–2872.

Jinyang Guo, Jiaheng Liu, and Dong Xu. 2021. Jointpruning: Pruning networks along multiple dimensions for efficient point cloud processing. *IEEE Transactions on Circuits and Systems for Video Technology*.

Jinyang Guo, Jiaheng Liu, and Dong Xu. 2022b. 3d-pruning: A model compression framework for efficient 3d action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12):8717–8729.

Jinyang Guo, Wanli Ouyang, and Dong Xu. 2020a. Channel pruning guided by classification loss and feature importance. *AAAI*.

Jinyang Guo, Wanli Ouyang, and Dong Xu. 2020b. Multi-dimensional pruning: A unified framework for model compression. In *CVPR*.

Jinyang Guo, Weichen Zhang, Wanli Ouyang, and Dong Xu. 2020c. Model compression using progressive channel pruning. *IEEE Transactions on Circuits and Systems for Video Technology*.

Jun Guo, Wei Bao, Jiakai Wang, Yuqing Ma, Xinghai Gao, Gang Xiao, Aishan Liu, Jian Dong, Xianglong Liu, and Wenjun Wu. 2023. A comprehensive evaluation framework for deep model robustness. *Pattern Recognition*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings.

Md Akmal Haidar, Mehdi Rezagholizadeh, Abbas Ghaddar, Khalil Bibi, Philippe Langlais, and Pascal Poupart. 2022. Cilda: Contrastive data augmentation using intermediate layer knowledge distillation. *arXiv preprint arXiv:2204.07674*.

Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. 2017. Smart mining for deep metric learning. In *ICCV*.

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

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

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. 2021. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. 2021. Annealing knowledge distillation. *arXiv preprint arXiv:2104.07163*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. 2019. Knowledge distillation via route constrained optimization. In *ICCV*.

Aishan Liu, Jun Guo, Jiakai Wang, Siyuan Liang, Renshuai Tao, Wenbo Zhou, Cong Liu, Xianglong Liu, and Dacheng Tao. 2023. X-adv: Physical adversarial object attacks against x-ray prohibited item detection. In *USENIX Security Symposium*.

Aishan Liu, Xianglong Liu, Hang Yu, Chongzhi Zhang, Qiang Liu, and Dacheng Tao. 2021. Training robust deep neural networks via adversarial noise propagation. *IEEE Transactions on Image Processing*.

Jiaheng Liu, Jinyang Guo, and Dong Xu. 2022a. Apsnet: Toward adaptive point sampling for efficient 3d action recognition. *IEEE Transactions on Image Processing*, 31:5287–5302.

Jiaheng Liu, Haoyu Qin, Yichao Wu, Jinyang Guo, Ding Liang, and Ke Xu. 2022b. Coupleface: relation matters for face recognition distillation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*. Springer.

Jiaheng Liu, Tan Yu, Hanyu Peng, Mingming Sun, and Ping Li. 2022c. Cross-lingual cross-modal consolidation for effective multilingual video corpus moment retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1854–1862, Seattle, United States. Association for Computational Linguistics.

Jiaheng Liu, Shunfeng Zhou, Yichao Wu, Ken Chen, Wanli Ouyang, and Dong Xu. 2020. Block proposal neural architecture search. *IEEE TIP*, 30:15–25.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *ICLR*.

Yuqing Ma, Shihao Bai, Wei Liu, Shuo Wang, Yue Yu, Xiao Bai, Xianglong Liu, and Meng Wang. 2021. Transductive relation-propagation with decoupling training for few-shot learning. *IEEE transactions on neural networks and learning systems*.

Yuqing Ma, Xianglong Liu, Shihao Bai, Lei Wang, Aishan Liu, Dacheng Tao, and Edwin R Hancock. 2022. Regionwise generative adversarial image inpainting for large missing areas. *IEEE Transactions on Cybernetics*.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *AAAI*.

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

Dae Young Park, Moon-Hyun Cha, Daesin Kim, Bohyung Han, et al. 2021. Learning student-friendly teacher networks for knowledge distillation. *Advances in Neural Information Processing Systems*.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *CVPR*.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2021. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on artificial intelligence*.

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.

Haotong Qin, Yifu Ding, Xiangguo Zhang, Jiakai Wang, Xianglong Liu, and Jiwen Lu. 2023a. Diverse sample generation: Pushing the limit of generative data-free quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Haotong Qin, Xudong Ma, Yifu Ding, Xiaoyang Li, Yang Zhang, Zejun Ma, Jiakai Wang, Jie Luo, and Xianglong Liu. 2023b. Bifsmnv2: Pushing binary neural networks for keyword spotting to real-network

performance. *IEEE Transactions on Neural Networks and Learning Systems*.

Haotong Qin, Mingyuan Zhang, Yifu Ding, Aoyu Li, Ziwei Liu, Fisher Yu, and Xianglong Liu. 2023c. Bibench: Benchmarking and analyzing network binarization. In *International Conference on Machine Learning*.

Haotong Qin, Xiangguo Zhang, Ruihao Gong, Yifu Ding, Yi Xu, and Xianglong Liu. 2022. Distribution-sensitive information retention for accurate binary neural network. *International Journal of Computer Vision*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. Fitnets: Hints for thin deep nets. *ICLR*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*. PMLR.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. 2019. Stochastic class-based hard example mining for deep metric learning. In *CVPR*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Siqi Sun, Zhe Gan, Yu Cheng, Yuwei Fang, Shuohang Wang, and Jingjing Liu. 2020. Contrastive distillation on intermediate representations for language model compression. *arXiv preprint arXiv:2009.14167*.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive representation distillation. In *ICLR*.

Frederick Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *ICCV*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ICLR*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *NAACL-HLT*.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *ICCV*.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*.

Sergey Zagoruyko and Nikos Komodakis. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ICLR*.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2022. Bert learns to teach: Knowledge distillation with meta learning. In *ACL*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☑ A2. Did you discuss any potential risks of your work?
7

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

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

B ☐ Did you use or create scientific artifacts?

Not applicable. Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

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

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

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

Left blank.

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

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

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

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

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