A Contrastive Self-Distillation BERT with Kernel Alignment-Based Inference

Yangyan Xu[1,2], Fangfang Yuan[1(✉)], Cong Cao[1(✉)],
Majing Su[3], Yuhai Lu[1], and Yanbing Liu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{xuyangyan,yuanfangfang,caocong,luyuhai,liuyanbing}@iie.ac.cn
[3] The 6th Research Institute of China Electronic Corporations, Beijing, China
sumj@ncse.com.cn

Abstract. Early exit, as an effective method to accelerate pre-trained language models, has recently attracted much attention in the field of natural language processing. However, existing early exit methods are only suitable for low acceleration ratios due to two reasons: (1) The shallow classifiers in the model lack semantic information. (2) Exit decisions in the intermediate layers are unreliable. To address the above issues, we propose a Contrastive self-distillation BERT with kernel alignment-based inference (CsdBERT), which aims to let shallow classifiers learn deep semantic knowledge to make comprehensive predictions. Specifically, we classify the early exit classifiers into teachers and students based on classification loss to distinguish the representation ability of the classifiers. Firstly, we present a contrastive learning approach between teacher and student classifiers to maintain the consistency of class similarity between them. Then, we introduce a self-distillation strategy between these two kinds of classifiers to solidify learned knowledge and accumulate new knowledge. Finally, we design a kernel alignment-based exit mechanism to identify samples of different difficulty for accelerating BERT inference. Experimental results on the GLUE and ELUE benchmarks show that CsdBERT not only achieves state-of-the-art performance, but also maintains 95% performance at $4\times$ speed.

Keywords: Early exit · Contrastive learning · Self-distillation · Centered kernel alignment

1 Introduction

Pre-trained language models (PLMs) have become the most promising models in the field of natural language processing (NLP), such as BERT [1], GPT [2], XLNet [3], RoBERTa [4], ALBERT [5], etc., which bring significant improvements to NLP tasks. Despite the great success of PLMs, they incur computational consumption, which leads to very slow inference and high latency. To cope with these issues, static model compression techniques are used to accelerate PLMs inference, such as knowledge distillation [6], quantization [7], and pruning [8], etc. However, this class of static compression methods aims to obtain a compact model, resulting in a dramatic performance degradation. Conversely, dynamic early exit [9] has proven to be an effective way to reduce

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

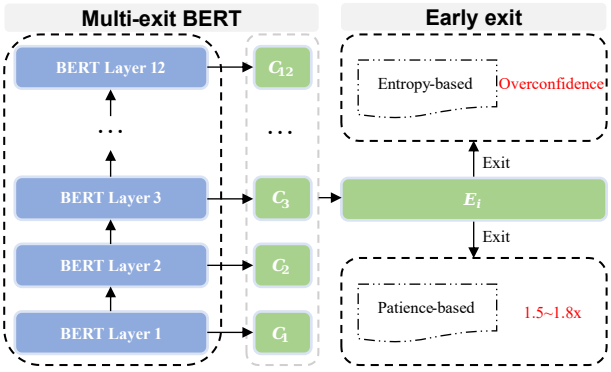


Fig. 1. On the left is an example of a multi-exit BERT, and on the right are the two mainstream exit mechanisms (where c_1 to c_{12} denote early exit classifiers and E_i denotes exit at some layer i).

computation and latency. The main idea of dynamic early exit models [9–13] is to add additional classifiers to the different layers of PLMs. Dynamic early exit PLMs mainly involve training these classifiers in the fine-tuning stage, thus exiting early in the inference stage. Hence, it is important to design an effective exit mechanism for the inference stage.

Existing exit methods can be categorized into entropy-based and patience-based methods, as shown in Fig. 1. The entropy-based methods [10,11,13,14] aim to compute the entropy of the predicted probability distribution as an estimate of the confidence of exiting classifiers. These methods are overconfident in some classifiers, making them unreliable indicators of confidence, and the ability of the low layers may not match their high confidence scores. The patience-based methods [9, 12, 15] rely on the cross-layer consistency prediction to make an exit decision. Unfortunately, these methods using the patience index only provide very limited acceleration ratios and cannot be applied to complex real-world scenarios.

To solve the above problems, we propose CsdBERT: a Contrastive self-distillation BERT with kernel alignment-based inference to reduce computational cost and inference latency. Our CsdBERT contains a contrastive learning approach and a self-distillation strategy in the fine-tuning phase, and a kernel alignment-based exit decision in the inference phase. Specifically, we first rank the early exit classifiers by classification loss, which can effectively distinguish strong teacher classifiers from weak student classifiers. For the contrastive learning approach, teachers and students are obtained from the encoder outputs of the early exit classifiers, and the ensemble teacher is obtained by averaging multiple teachers to allow the student classifiers to learn the contrastive training signal of the ensemble teacher at the encoder level. For the self-distillation strategy, the teachers composed of soft-labels become an expert teacher through the weight of difficulty perception, so that students can learn the rich knowledge of the expert teacher and reduce forgetting. Finally, we design a kernel alignment-based

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

exit mechanism to calibrate the predictions of the model based on the instance difficulty, which satisfies the requirements of acceleration ratios in different scenarios.

Extensive experiments are carried out on the GLUE and ELUE benchmarks, and the results show that our proposed method not only outperforms the state-of-the-art methods, but also achieves the highest acceleration ratio.

2 Related Work

Large-scale pre-trained language models based on the Transformer architecture show excellent performance in the field of NLP. However, such models have a large number of parameters, resulting in large memory requirements and computational consumption during inference. To deal with this problem, studies [16–21] on improving the efficiency of over-parameterized models has gradually emerged.

Static Compression. Knowledge distillation [22], as a model compression technique, compacts the model structure to a smaller model, and keeps static for all instances in the inference process. In the pre-training stage, [16] reduces the size of the BERT model by knowledge distillation, which makes the training cost of the model less and the inference time shorter. [17] proposes deep self-attention distillation, and student models are trained by deep imitation of self-attention module, which shows that using the knowledge of the last Transformer layer can alleviate the difficulties of layer mapping between teacher and student models. [18] proposes progressive module replacing, which provides a new perspective for model compression without additional loss functions. [23] demonstrates that the teacher network can learn to better transfer knowledge to the student network by distilling the feedback on the student network's performance in a meta-learning framework. However, these static compression methods have to distill the model from scratch to meet different speedup requirements and treat instances requiring different computational costs indiscriminately.

Dynamic Early Exit. To meet different speedup constraints, instance adaptation methods [14, 24] have been proposed to adjust the number of executed layers for different instances. Among them, dynamic early exit is an efficient way to adaptively speed up inference, which is first used in computer vision tasks [25–27]. [28] makes full use of the idea of early exit, and proposes calibrated confidence scores to make early exit decisions, which are applied to NLP tasks to show the effectiveness of the method. [10] proposes a two-stage training method. In the first stage, the classifier of the last Transformer layer is fine-tuned; In the second stage, the parameters fine-tuned in the first stage are frozen, and then the remaining eleven classifiers are updated. To make predictions with fewer Transformer layers, [9] proposes a joint training method to train each early exit classifier, and make the model stop inference dynamically through cross-layer consistent prediction. In addition to the joint training and two-stage training methods, [29] proposes an alternating training method, which combines the advantages of the first two training methods and extends the idea of early exit to regression tasks. [11] proposes a speed-adjustable BERT model, which improves the inference time of NLP model through sample-wise adaptation and self-distillation mechanism. [30] analyzes the working mechanism of dynamic early exit and shows that dynamic selection of appropriately sized models in a cascading manner can provide a comprehensive represen-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

tation for prediction. To improve the performance of early exit classifiers, [15] reveals that each early exit classifier can learn from each other, and improves the optimization results through a cross-level optimization algorithm. [12] uses mutual learning and gradient alignment for knowledge distillation, which shows that deep classifiers can also learn from shallow classifiers, and the conflict between cross-entropy loss and distillation loss can be eliminated by gradient alignment. [14] proposes a unified horizontal and vertical multi-perspective early exit framework to achieve a trade-off between efficiency and performance. However, most of the research on this subject has not fully exploited semantic information of high-layer classifiers, and the judgment of exit decisions is not accurate enough.

3 Method

Our CsdBERT aims at learning deep semantic knowledge to realize efficient inference. The main framework of CsdBERT is shown in Fig. 2. Firstly, we train early exit classifiers by classification loss, and then we present a contrastive learning design and elaborate a self-distillation strategy. Finally, we design a kernel alignment-based exit mechanism.

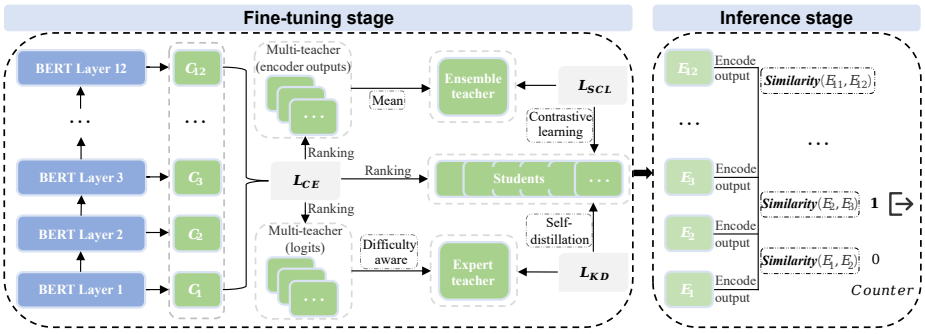


Fig. 2. The framework of our proposed CsdBERT. There are three main loss functions in the fine-tuning stage: classification loss $\mathcal{L}_{CE}$, contrastive loss $\mathcal{L}_{SCL}$, and distillation loss $\mathcal{L}_{KD}$. *Centered kernel alignment* is used as a similarity index to measure the output of the encoder in the exit layer, and the prediction can be accurately output in advance by counting.

3.1 Early Exit

The base version of the BERT model typically contains twelve layers of transformers. The input sample is first tokenized as a sequence of subwords, i.e., $X = [x_1, x_2, ..., x_K]$, with the corresponding ground truth label y, where K is the length of the sequence. Firstly, a special token $[CLS]$ is added to the head of the sequence so that each layer's corresponding hidden state $h^{(l)}_{[\mathrm{CLS}]}$ is encoded through a multilayer encoding process

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

that includes all representative information about all tokens. For each layer, the encoding process is defined as follows:

$$\left[h_{[\mathrm{CLS}]}^{(l)}, h_1^{(l)}, \ldots, h_N^{(l)}\right] = f_\theta^{(l)}\left(\left[h_{[\mathrm{CLS}]}^{(l-1)}, h_1^{(l-1)}, \ldots, h_N^{(l-1)}\right]; \theta\right), \tag{1}$$

where $f_\theta^{(l)}$ is the Transformer encoder at the l_{th} layer of the θ parameterization. $h_{[\mathrm{CLS}]}^{(l)}$ is considered to be the features that train the early exit classifier $c^{(l)}$.

The predicted probability distribution $\hat{y}^{(l)}$ of the early exit classifier for the ground-truth label is computed as follows:

$$c^{(l)} = \tanh\left(W_c^{(l)} h_{[\mathrm{CLS}]}^{(l)} + b_c^{(l)}\right), \tag{2}$$

$$\hat{y}_c^{(l)} = \mathrm{softmax}\left(c^{(l)}\right), \tag{3}$$

where $\tanh$ is the activation function, $W_c^{(l)}$ is the weight, and $b_c^{(l)}$ is the bias. The loss function for the target task is a categorical cross-entropy, which is defined as

$$\mathcal{L}_{CE}^{(l)} = -\sum_{l=1}^{L} \mathbb{I}(y) \circ \log\left(\hat{y}_c^{(l)}\right), \tag{4}$$

where y and $\hat{y}_c^{(l)}$ denote the ground truth and probability distribution of the l_{th} layer. $\mathbb{I}(y)$ represents a one-hot vector with the y_{th} component being one. $\circ$ denotes the element-wise multiplication operation.

3.2 Classification Loss

To train early exit classifiers of BERT, additional classifiers are added. For each classifier $c^{(l)}$ ($l = [1, 2, ..., L]$), its cross-entropy loss $\mathcal{L}_l^{CE}$ is first calculated. Then, we use the summed loss to facilitate the joint training of these classifiers:

$$\mathcal{L}_{CE} = \sum_{l=1}^{L} \mathcal{L}_l^{CE}, \tag{5}$$

where L is the number of early exit classifiers.

Based on the cross-entropy loss, we sort the encoder outputs corresponding to all classifiers, and select M students to be the set S (where $S = [S_1, S_2, ..., S_M]$) with larger losses and the remaining ones as teachers $T^{(encoder)}$. The ensemble teacher T_{ens} is obtained by averaging over multiple teachers.

3.3 Contrastive Loss

To alleviate the inherent semantic bias of the student classifiers in the training process, we introduce self-supervised contrastive loss [31], which explores the merit of the ensemble teacher classifier. The contrastive learning between the student classifiers and

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

the ensemble teacher classifier can effectively improve the ability of shallow classifiers to extract semantic information. Our goal is to encourage the same samples to be pulled closer and the other samples be pushed away between the ensemble teacher and the student classifiers. The specific form is as follows:

$$l_{SCL}^{(x_i,x_j)} = \frac{e^{\mathrm{sim}(T_{ens}(x_i),S_m(x_j))/\tau}}{\sum_{k=1,k\neq i}^{2Q} e^{\mathrm{sim}(T_{ens}(x_i),S_m(x_k))/\tau}}, \tag{6}$$

where the sample x_i is from the ensemble teacher classifier T_{ens} and the sample x_j is from the student classifier S_m ($m = [1, 2, ..., M]$, where M is the number of students). Q is the batch size. For the ensemble teacher T_{ens} and student S_m, the similarity between them is computed, where sim(., .) is the cosine similarity (the dot product of the input samples). τ denotes the contrastive temperature [32]. As suggested by [31], we also use the normalised softmax in Eq. (6), instead of using the cosine similarity measure directly. The contrastive loss takes the form of

$$\mathcal{L}_{SCL}^{(m)} = -\frac{1}{Q}\sum_{j=1}^{Q} \log l_{SCL}^{(x_i,x_j)}, \tag{7}$$

where the contrastive loss is defined as the arithmetic mean of the cross-entropy of the normalised similarity $l_{SCL}^{(x_i,x_j)}$. Then, the final contrastive loss is

$$\mathcal{L}_{SCL} = \frac{\sum_{m=1}^{M} m \cdot \mathcal{L}_{SCL}^{(m)}}{\sum_{m=1}^{M} m}. \tag{8}$$

3.4 Distillation Loss

Apart from maintaining the consistency of class similarities, distilling semantic knowledge from the teacher classifiers is also essential to alleviate the catastrophic forgetting. To this end, we divide soft-labels $logits$ of classifiers into students S and teachers $T^{(logits)}$ (where $S = [S_1, S_2, ..., S_M]$, $T^{(logits)} = [T_1^{(logits)}, T_2^{(logits)}, ..., T_N^{(logits)}]$, M is the total number of students and N is the total number of teachers). Firstly, we design the difficulty function:

$$\mathit{Dif}^{(n)} = \frac{\sum_{i=1}^{Y} p_t^n(i) \log p_t^n(i)}{\log \frac{1}{Y}}, \tag{9}$$

where Y is the number of labeled classes, $n = [1, 2, ..., N]$. $p_t^n(i)$ is the distribution of output probability of the teacher classifier. Then, we multiply the corresponding teacher by $\mathit{Dif}^{(n)}$ as a weight to become an expert teacher T_{exp}:

$$T_{exp} = \sum_{n=1}^{N} Dif^{(n)} \cdot T_n^{(logits)}, \tag{10}$$

The distillation is performed as

$$\mathcal{L}_{KD} = \sum_{m=1}^{M} \mathcal{L}_{S_m \to T_{exp}}^{KD}, \tag{11}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

where $\mathcal{L}^{KD}$ is the distillation loss [11]. S_m denotes the student ($m = [1, 2, ..., M]$). Since the model structure of the expert teacher T_{exp} and the student S_m is the same, self-distillation is carried out between them.

3.5 Total Loss

In the fine-tuning stage, we train early exit classifiers by the above three losses. The total loss function of CsdBERT is summarized as

$$\mathcal{L}_{Total} = \mathcal{L}_{CE} + \alpha\mathcal{L}_{SCL} + \beta\mathcal{L}_{KD}, \tag{12}$$

where α and β are hyper-parameters to balance the effect between contrastive loss and distillation loss.

3.6 Centered Kernel Alignment

To achieve a high acceleration ratio, we use an early exit based on centered kernel alignment (CKA) [33] in the inference stage. We first employ the encoder output E_i of each exit layer as input. Then, we use CKA as the similarity index. The similarity of the encoder outputs is computed two-by-two in turn, which is denoted as

$$\mathrm{CKA}(E_i, E_{i+1}) = \frac{\mathrm{HSIC}(E_i, E_{i+1})}{\sqrt{\mathrm{HSIC}(E_i, E_i)\,\mathrm{HSIC}(E_{i+1}, E_{i+1})}}, \tag{13}$$

where $E = [E_1, E_2, ..., E_L]$, L is the number of early exit classifiers. HSIC is

$$\mathrm{HSIC}(E_i, E_{i+1}) = \frac{1}{(n-1)^2}\,\mathrm{tr}(E_i H E_{i+1} H), \tag{14}$$

where H denotes the centering matrix, $H_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$. n is the size of the E_i row vector. We count once when $Similarity(i+1)$ minus $Similarity(i)$ is less than θ ($Similarity(i) = CKA(E_i, E_{i+1})$), corresponding to an *inference threshold* of 1. We count twice, then *inference threshold* is 2, and so on.

4 Experiments

Datasets. We conduct experiments on the GLUE [34] and ELUE [35] benchmarks, as shown in Table 1 and Table 2, respectively. On the GLUE benchmark, we test on Recognizing Textual Entailment (RTE), Question Natural Language Inference (QNLI), and Multi-Genre Natural Language Inference Matched (MNLI) for the Natural Language Inference (NLI) task; Quora Question Pairs (QQP) for Paraphrase Similarity Matching. On the ELUE benchmark, we test on Microsoft Research Paraphrase Matching (MRPC) for Paraphrase Similarity Matching; Stanford Sentiment Treebank (SST-2) and IMDb for Sentiment Classification; SciTail and SNLI for the NLI task.

Baselines. We compare our method with two types of baselines: (1) **Compressed models**. We choose BERT-6L, BERT-of-Theseus [18], LayerDrop [36], DistilBERT [16],

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

Table 1. The GLUE benchmark.

Tasks	**Datasets**	**#Train**	**#Dev**	**#Test**
Natural Language Inference	RTE	2,490	277	3,000
Natural Language Inference	QNLI	104,743	5,463	5,463
Paraphrase Similarity Matching	QQP	363,849	40,430	390,965
Natural Language Inference	MNLI	392,702	9,815	9,796

Table 2. The ELUE benchmark.

Tasks	**Datasets**	**#Train**	**#Dev**	**#Test**
Paraphrase Similarity Matching	MRPC	3,668	408	1,725
Sentiment Classification	SST-2	8,544	1,101	2,208
Sentiment Classification	IMDb	20,000	5,000	25,000
Natural Language Inference	SciTail	23,596	1,304	2,126
Natural Language Inference	SNLI	549,367	9,842	9,824

and BERT-PKD [37] as baselines. (2) **Early exit models**. We choose state-of-the-art early exit models, including PABEE [9], DeeBERT [10], FastBERT [11], GAML-BERT [12], and CascadeBERT [30].

Evaluation Metrics. For QQP and MRPC, we report the unweighted average of accuracy and F1 score. For the other datasets, we simply adopt accuracy as the metric. We use the above metrics to evaluate our CsdBERT and baselines in all experiments, and the detailed information of metrics are shown in the literature [34, 35].

Table 3. Results(%) on the GLUE benchmark.

Models	#Param	Speed-up	RTE	QNLI	QQP	MNLI	Average
BERT-base	109M	1.00×	67.1	86.8	88.7	83.5	81.5
BERT-6L	66M	1.96×	60.3	80.2	84.8	77.1	75.6
BERT-of-Theseus	66M	1.96×	63.6	82.4	86.5	80.7	78.3
LayerDrop	66M	1.96×	62.8	81.7	86.2	80.1	77.7
DistilBERT	66M	1.96×	63.1	82.2	86.4	79.8	77.9
BERT-PKD	66M	1.96×	63.4	82.5	86.8	80.6	78.3
CascadeBERT	66M	1.96×	64.6	89.4	71.2	83.0	77.1
GAML-BERT	109M	1.96×	66.8	85.1	89.1	83.2	81.1
PABEE	109M	1.89×	64.5	83.1	87.5	81.5	79.2
DeeBERT	109M	1.88×	63.9	82.5	87.3	81.2	78.7
FastBERT	109M	1.93×	64.9	83.6	88.1	82.1	79.7
CsdBERT(Ours)	109M	1.96×	**73.3**	**92.1**	**90.0**	**85.3**	**85.2**

Implementation Details. Following [35], the batch size is set to 32 and the learning rate is set to 2e-5. The contrastive temperature τ is set to 0.5 and θ is set to 0.06. We

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

fine-tune 5 epochs using Adam optimizer, and take the distillation temperature to be 4. Furthermore, we set α equal to 0.01 and β equal to 1. The implementation of CsdBERT is based on the HuggingFace Transformers Library [38].

Table 4. Results(%) on the ELUE benchmark.

Models	#Param	Speed-up	MRPC	SST-2	IMDb	SciTail	SNLI	Average
ElasticBERT-base	109M	1.00×	87.9	88.6	93.9	93.8	91.3	91.1
ElasticBERT-entropy	109M	1.88×	87.5	88.3	88.2	94.5	90.0	89.7
ElasticBERT-patience	109M	1.89×	86.7	88.7	88.0	93.9	90.1	89.5
CsdBERT(Ours)	109M	1.96×	**91.8**	**93.1**	**88.7**	**95.9**	**91.0**	**92.1**

4.1 Experimental Results on GLUE and ELUE

Comparison with State-of-the-Art Methods. To verify the effectiveness of CsdBERT, we first evaluate CsdBERT and our baselines on the GLUE benchmark with BERT as the backbone model. From Table 3, we can see that CsdBERT outperforms the state-of-the-art methods with the same speedup ratio. For instance, the accuracy of our method on RTE is 73.3%, which improves 6.5% compared with the best result of GAML-BERT. This is because our method not only enhances the ability of classifiers to recognize semantic knowledge, but also determines the difficulty of different samples to exit precisely in advance.

Further Comparison. In order to further explore the efficiency of our method, we conduct experiments on the ELUE benchmark with ElasticBERT [35] as the backbone model. From Table 4, we can see that CsdBERT outperforms comparative models. The reason is that we alleviate semantic bias between the strong classifier (the ensemble teacher classifier made by combining multiple teachers) and the weak classifiers (student classifiers) by contrastive learning. Then, through self-distillation, our weak classifiers can learn rich semantic knowledge from the strong classifier.

Table 5. Ablation study of CsdBERT.

Models	RTE	MRPC	SST-2
CsdBERT	**73.3**	**91.8**	**93.1**
w/o $\mathcal{L}_{SCL}$	70.4	90.3	87.4
w/o $\mathcal{L}_{KD}$	69.7	90.0	88.3

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

4.2 Ablation Study

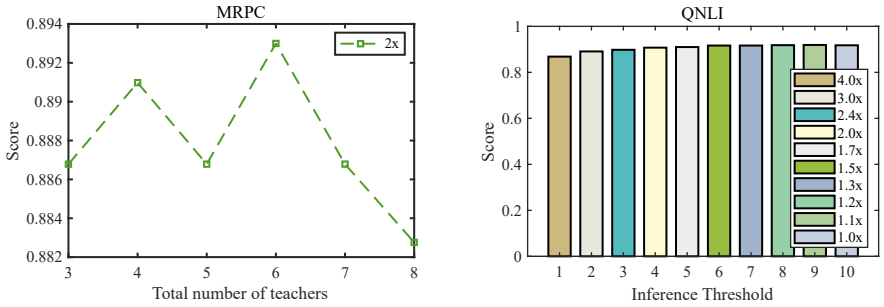
We conduct ablation study to validate the effectiveness of the essential components of CsdBERT, and the results are shown in Table 5.

(1) w/o $\mathcal{L}_{SCL}$. The accuracy corresponding to all the three tasks shows a relatively large decrease in the absence of contrastive loss. This is because contrastive loss ensures the student classifiers to get rich contrastive training signals from the ensemble teacher classifier.

(2) w/o $\mathcal{L}_{KD}$. If distillation loss is removed, the performance of classifiers in each layer is degraded. This indicates that the student classifiers can learn new knowledge of the expert teacher classifier.

4.3 Parameter Analysis

Total Number of Teachers N. As shown in Fig. 3 (a), the total number of teachers has a different impact on performance. In order to select the optimal number of teachers, we conducted experiments on MRPC at $2\times$ speed. We analyse the effect of N by varying its value from 3 to 8. With the increase of teachers, the score first increases and then declines dramatically. As the score is highest when N = 6, we use N = 6 in all experiments reported.



(a) Multi-teacher selection (b) Different acceleration ratios

Fig. 3. Teacher selection and acceleration ratio. For the MRPC task, *Score* is the average of accuracy and F1 score. For the QNLI task, *Score* represents accuracy.

Inference Threshold R. On QNLI, we show the speedup ratios corresponding to different values of inference thresholds. Concretely, compared to BERT's accuracy on QNLI, CsdBERT still maintains 95% performance at $4\times$ speed. As illustrated in Fig. 3 (b), when R is 1, the speedup ratio is the highest.

5 Conclusion

In this paper, we propose CsdBERT for accelerating BERT inference. We first keep the consistency of class similarity between strong ensemble teacher and weak student

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

classifiers via contrastive learning. Then, our student classifiers learn richer knowledge of the expert teacher via self-distillation. Finally, our designed kernel alignment-based mechanism can reflect the real difficulty of each instance and output more reliable predictions. Experimental results on the GLUE and ELUE benchmarks show that CsdBERT outperforms the state-of-the-art methods, and maintains 95% performance at $4\times$ speed. In the future, we will explore CsdBERT deployed to complex real-world scenarios.

Acknowledgement. This work is supported by Key Research and Development Program Projects of Xinjiang (No. 2022B03010-2), Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDC02030400).

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT. pp. 4171–4186 (2019)
2. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. OpenAI (2018)
3. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS. pp. 5754–5764 (2019)
4. 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)
5. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. In: International Conference on Learning Representations, ICLR (2020)
6. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
7. Tang, H., Zhang, X., Liu, K., Zhu, J., Kang, Z.: MKQ-BERT: quantized BERT with 4-bits weights and activations. arXiv preprint arXiv:2203.13483 (2022)
8. Lassance, C., Maachou, M., Park, J., Clinchant, S.: Learned token pruning in contextualized late interaction over BERT (colbert). In: SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2232–2236 (2022)
9. Zhou, W., Xu, C., Ge, T., McAuley, J.J., Xu, K., Wei, F.: BERT loses patience: Fast and robust inference with early exit. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS (2020)
10. Xin, J., Tang, R., Lee, J., Yu, Y., Lin, J.: Deebert: Dynamic early exiting for accelerating bert inference. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL. pp. 2246–2251 (2020)
11. Liu, W., Zhou, P., Wang, Z., Zhao, Z., Deng, H., Ju, Q.: Fastbert: a self-distilling bert with adaptive inference time. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL. pp. 6035–6044 (2020)
12. Zhu, W., Wang, X., Ni, Y., Xie, G.: GAML-BERT: improving BERT early exiting by gradient aligned mutual learning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP. pp. 3033–3044 (2021)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

13. Kong, J., Wang, J., Zhang, X.: Accelerating pretrained language model inference using weighted ensemble self-distillation. In: Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC. vol. 13028, pp. 224–235 (2021)
14. Kong, J., Wang, J., Yu, L., Zhang, X.: Accelerating inference for pretrained language models by unified multi-perspective early exiting. In: Proceedings of the 29th International Conference on Computational Linguistics, COLING. pp. 4677–4686 (2022)
15. Zhu, W.: Leebert: Learned early exit for BERT with cross-level optimization. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP. pp. 2968–2980 (2021)
16. 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)
17. 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: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS (2020)
18. Xu, C., Zhou, W., Ge, T., Wei, F., Zhou, M.: BERT-of-theseus: Compressing BERT by progressive module replacing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP. pp. 7859–7869 (2020)
19. Ryu, M., Lee, G., Lee, K.: Knowledge distillation for BERT unsupervised domain adaptation. Knowl. Inf. Syst. **64**(11), 3113–3128 (2022)
20. Yao, S., Tan, J., Chen, X., Zhang, J., Zeng, X., Yang, K.: Reprbert: Distilling BERT to an efficient representation-based relevance model for e-commerce. In: KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 4363–4371 (2022)
21. Gao, Y., Zhang, B., Qi, X., So, H.K.: DPACS: hardware accelerated dynamic neural network pruning through algorithm-architecture co-design. In: Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS. pp. 237–251 (2023)
22. Yuan, F., Shou, L., Pei, J., Lin, W., Gong, M., Fu, Y., Jiang, D.: Reinforced multi-teacher selection for knowledge distillation. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI. pp. 14284–14291 (2021)
23. Zhou, W., Xu, C., McAuley, J.J.: BERT learns to teach: Knowledge distillation with meta learning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL. pp. 7037–7049 (2022)
24. Zhang, Z., Zhu, W., Zhang, J., Wang, P., Jin, R., Chung, T.: PCEE-BERT: accelerating BERT inference via patient and confident early exiting. In: Findings of the Association for Computational Linguistics: NAACL. pp. 327–338 (2022)
25. Teerapittayanon, S., McDanel, B., Kung, H.T.: Branchynet: Fast inference via early exiting from deep neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 2464–2469. IEEE (2016)
26. Kaya, Y., Hong, S., Dumitras, T.: Shallow-deep networks: Understanding and mitigating network overthinking. In: Proceedings of the 36th International Conference on Machine Learning, ICML. vol. 97, pp. 3301–3310 (2019)
27. Elbayad, M., Gu, J., Grave, E., Auli, M.: Depth-adaptive transformer. arXiv preprint arXiv:1910.10073 (2019)
28. Schwartz, R., Stanovsky, G., Swayamdipta, S., Dodge, J., Smith, N.A.: The right tool for the job: Matching model and instance complexities. arXiv preprint arXiv:2004.07453 (2020)
29. Xin, J., Tang, R., Yu, Y., Lin, J.: Berxit: Early exiting for BERT with better fine-tuning and extension to regression. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL. pp. 91–104 (2021)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39

30. Li, L., Lin, Y., Chen, D., Ren, S., Li, P., Zhou, J., Sun, X.: Cascadebert: Accelerating inference of pre-trained language models via calibrated complete models cascade. In: Findings of the Association for Computational Linguistics: EMNLP. pp. 475–486 (2021)
31. Atito, S., Awais, M., Kittler, J.: Sit: Self-supervised vision transformer. arXiv preprint arXiv:2104.03602 (2021)
32. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning, ICML. vol. 119, pp. 1597–1607 (2020)
33. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.E.: Similarity of neural network representations revisited. In: Proceedings of the 36th International Conference on Machine Learning, ICML. vol. 97, pp. 3519–3529 (2019)
34. 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. arXiv preprint arXiv:1804.07461 (2018)
35. Liu, X., Sun, T., He, J., Wu, L., Zhang, X., Jiang, H., Cao, Z., Huang, X., Qiu, X.: Towards efficient nlp: A standard evaluation and a strong baseline. arXiv preprint arXiv:2110.07038 (2021)
36. Fan, A., Grave, E., Joulin, A.: Reducing transformer depth on demand with structured dropout. In: International Conference on Learning Representations, ICLR (2019)
37. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP. pp. 4322–4331 (2019)
38. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP. pp. 38–45 (2020)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_39