

LRC-BERT: Latent-representation Contrastive Knowledge Distillation for Natural Language Understanding

Hao Fu,^{1*} Shaojun Zhou,^{2*} Qihong Yang,² Junjie Tang,^{2†} Guiquan Liu,¹
Kaikui Liu,² Xiaolong Li²

¹School of Computer Science and Technology, University of Science and Technology of China

²Alibaba Group

hfu@mail.ustc.edu.cn, {zsj148798, xiaokui.yqh, lixi.tjj}@alibaba-inc.com, gqliu@ustc.edu.cn,
{damon, xl.li}@alibaba-inc.com

Abstract

The pre-training models such as BERT have achieved great results in various natural language processing problems. However, a large number of parameters need significant amounts of memory and the consumption of inference time, which makes it difficult to deploy them on edge devices. In this work, we propose a knowledge distillation method LRC-BERT based on contrastive learning to fit the output of the intermediate layer from the angular distance aspect, which is not considered by the existing distillation methods. Furthermore, we introduce a gradient perturbation-based training architecture in the training phase to increase the robustness of LRC-BERT, which is the first attempt in knowledge distillation. Additionally, in order to better capture the distribution characteristics of the intermediate layer, we design a two-stage training method for the total distillation loss. Finally, by verifying 8 datasets on the General Language Understanding Evaluation (GLUE) benchmark, the performance of the proposed LRC-BERT exceeds the existing state-of-the-art methods, which proves the effectiveness of our method.

Introduction

Recently, one of the main trends of natural language processing (NLP) is pre-training model (Peters et al. 2018; Radford et al. 2018). The pre-training models such as BERT (Devlin et al. 2019) and XLNet (Yang et al. 2019) have achieved remarkable results in tasks like Sentiment Classification (Socher et al. 2013) and Natural Language Inference (Williams, Nangia, and Bowman 2017). The training of these models usually consists of two stages: the first stage is the model pre-training on a large scale corpus by predicting specific words according to the given context. The second stage is to add a specific prediction layer on a specific downstream task for fine-tuning training. However, these models usually contain hundreds of millions of parameters. For example, the original BERT-base (Devlin et al. 2019) model has 12 layers and 109 million parameters. This limits the deployment of online services because of inference

delay and device capacity constraints. Therefore, in the case of limited computing resources, it is necessary to reduce the computational cost of these models in practice.

Knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015; Adriana et al. 2015) is an effective method of model compression (Han, Mao, and Dally 2015). By compressing a large teacher network into a small student network, the student network has the same predictive ability as the teacher network. At the same time, the student network has fewer parameters, correspondingly faster inference speed on specific tasks and can be effectively deployed to edge devices, greatly saving computing resources. Recalling the existing research on the KD method in NLP, DistilBERT (Sanh et al. 2019) uses soft label distillation loss, cosine embedding loss, and initializes the student from the teacher by taking one layer out of two. This strategy forces the structure of the transformer layer in the student network to be consistent with the teacher, and the distillation of the output characteristics of the intermediate layer is not fully considered. The student model in BERT-PKD (Sun et al. 2019) patiently learns from multiple intermediate layers of the teacher model for incremental knowledge extraction, including PKD Last and PKD Skip. The former try to learn from the last k layers, the latter learn from each k layer. Like DistilBERT, BERT-PKD also requires the corresponding layer in the student to be exactly the same as the teacher, which greatly limits the flexibility of structural design and the compression scale. TinyBERT (Jiao et al. 2019) uses more fine-grained knowledge including the hidden state of the transformer network and self-attention distribution. Additionally, a parameter matrix is introduced to perform a linear transformation on the hidden state of student so as to reduce the number of parameters in the transformer layer of the student. However, these models only mimic the output value of each layer of the teacher network, without considering structural information such as the correlation and difference of the output distribution among different samples.

In this work, we design a model compression framework called LRC-BERT to distill the knowledge of the teacher network into the student network. Specifically, we design a variety of different loss functions for the output of the transformer and prediction layer. For the output logits of the prediction layer, we calculate Kullback-Leibler (KL) di-

*Hao Fu and Shaojun Zhou are equal contribution first authors. This work was mainly done when Hao Fu was an intern at Alibaba Group.

†Junjie Tang is the corresponding author.
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

vergence and cross entropy with the output of the teacher and the real label respectively; for the output of the intermediate transformer layer, we design a novel contrastive loss cosine-based noise-contrastive estimation (COS-NCE) to capture the distribution structure characteristics of the output of the intermediate layer. The design of COS-NCE is inspired by infoNCE (Oord, Li, and Vinyals 2018) and CRD (Tian, Krishnan, and Isola 2020). Optimizing the contrastive loss can narrow the gap between the representations of positive pairs in a metric space, push the gap between the representations of negative pairs away, and learn the structural information output by the teacher network in the intermediate layer effectively. However, these structural information is ignored in the existing methods, such as TinyBERT and MINILM (Wang et al. 2020), they only approximate the output values of the teacher network in the transformer layers. Compared with existing model DistilBERT, LRC-BERT choose a more flexible student structure without the limitation of the structure and parameter quantity of the intermediate layer to be consistent with the teacher by introducing a dimension transformation matrix. Then, in order to help LRC-BERT focus on the structural information of the output distribution in the early stage of training, we adopt a two-stage training method for the total distillation loss, and verify the effectiveness of calculating distillation loss by stages in the experimental part. Finally, we introduce a new model training architecture based on gradient perturbation (Miyato, Dai, and Goodfellow 2017), which perturbs the output of the word vector embedding layer through the gradient value, thereby changing the output distribution of the model in this layer and improving the robustness of the model. Our contributions are summarized as follows:

- For the distillation of the intermediate layer, a new contrastive loss COS-NCE is proposed to effectively capture the structural characteristics between different samples.
- The gradient perturbation is introduced for the first time in knowledge distillation and verified in the experimental part that it can improve the robustness of LRC-BERT.
- We use a two-stage training method to better capture the output distribution characteristics of the intermediate layer based on COS-NCE.
- We evaluate the proposed method on 8 NLP datasets, and the experimental results show that our LRC-BERT model outperforms than the state-of-the-art baseline models.

Related Work

NLP Pre-trained Language Models

The emergence of pre-training models has brought NLP into a new era (Peters et al. 2018; Song et al. 2019; Jiao et al. 2019; Devlin et al. 2019). The advantages of using pre-training models are as follows: 1) Pre-training models learn knowledge from the large-scale corpus, which is helpful for downstream tasks; 2) pre-training provides a better parameter initialization method, which makes better generalization

and faster convergence on the target task; 3) pre-training can be considered as a regularization method, which can prevent the model from over fitting on small datasets (Qiu et al. 2020). Specifically, the pre-training language model can be divided into two stages: 1) pre-training word embedding; 2) pre-training context encoder.

In the first stage of pre-training word embedding, only word vectors are trained, which is a static feature-based mode. Typical examples are Word2Vec (Mikolov et al. 2013) and Glove (Pennington, Socher, and Manning 2014). The word vector is used for token embedding and sent it into the specific model. This kind of model has simple structure, but it can also get high-quality word vector which can capture the potential grammatical and semantic information between words in the potential text. However, this type of pre-training word vector can not be dynamic with the context variety. In addition, in the downstream tasks, the remaining model parameters still need to be retrained.

In the second stage of pre-training context encoder, the semantic information dynamically changes with the context. Typical examples are ULMFiT (Howard and Ruder 2018), ELMo (Peters et al. 2018) and GPT (Radford et al. 2018). With the introduction of transformer (Vaswani et al. 2017), it has brought more attention to the pre-training language model with a deeper structure. The typical representative is BERT (Devlin et al. 2019), which is a bi-directional encoder based on transformer and can learn sentence information more completely by using context. Through pre-training via masked language modeling and next sentence prediction, it has achieved the advanced performance on many NLP tasks, such as the GLUE benchmark (Wang et al. 2019a).

Contrastive Learning

Contrastive Learning (Hadsell, Chopra, and LeCun 2006; Weinberger and Saul 2009; Schroff, Kalenichenko, and Philbin 2015), as a kind of self-supervised learning method, uses the data itself as the supervised information to learn the feature expression of the sample data by constructing positive samples and negative samples, without manual annotation information. In fact, by minimizing some specially constructed contrastive loss such as infoNCE (Oord, Li, and Vinyals 2018) and CRD (Tian, Krishnan, and Isola 2019), the lower bound of the mutual information between all samples can be maximized.

The construction of positive and negative samples is one of the major difficulties in contrastive learning. In the field of image, deep InfoMAX (Hjelm et al. 2018) takes local features of training images and different images as positive samples and negative samples respectively. Then, MoCo (He et al. 2020) and SimCLR (Chen et al. 2020) extract positive samples by clipping, rotation and other transformation operations. For the data such as text and audio, because the data itself is sequence, the method adopted by the CPC (Oord, Li, and Vinyals 2018) is to take the sample from the data that has not been input into the model as a positive example, and randomly sampling an input from the input sequence as a negative example.

Knowledge Distillation

The model compression technology (Han, Mao, and Dally 2015) accelerates the inference speed and reduces the number of model parameters while ensuring the prediction performance of the model. Its main technologies consist of using fewer bits to represent the quantization of parameter weights (Gong et al. 2014), weight pruning (Han et al. 2015) to reduce or dilute network connections and knowledge distillation (Adriana et al. 2015; Tang et al. 2019; Tan et al. 2018).

Knowledge distillation has proven to be a promising method in model compression, which transfers the knowledge of a large model or a group of neural networks (teacher) to a single lightweight model (student). (Hinton, Vinyals, and Dean 2015) first proposed using the soft target distribution to train the student model and impart the knowledge of teachers to students. (Chen, Goodfellow, and Shlens 2016) introduced technologies that effectively transfer knowledge from existing networks to a deeper or wider network. (Turc et al. 2019) used a small pre-trained language model to initialize students during task-specific distillation. MiniBERT (Tsai et al. 2019) uses soft target distribution for mask language modeling prediction, guides the training of multilingual student models. TinyBERT (Jiao et al. 2019) and MobileBERT (Sun et al. 2020) further introduced self-attention distribution and hidden state to train students. (Aguilar et al. 2020) distilled internal representation knowledge from attention layer and intermediate output layer in a bottom-up way.

Different from the existing works paying more attention to approximate the attention value and output value in the transformer layer, we try to capture the correlation and difference of the output distribution between different input data of the teacher network in each layer, and these structural features have also been proved to be feasible in the knowledge distillation by CRD.

Method

In this section, we first propose COS-NCE by distilling the output distribution of the intermediate layer, and then show the process details of proposed adaptive deep contrastive distillation framework LRC-BERT, as shown in Fig. 1. Finally, we introduce a model training method based on gradient perturbation to enhance the robustness of LRC-BERT.

Problem Definition

The trained teacher network is represented by the function $f^T(x, \theta)$, where x is the input of the network and θ is the parameters. Our goal is to make the student network $f^S(x, \theta')$ imitate the output distribution of $f^T(x, \theta)$ in the intermediate and prediction layer by minimizing the specific loss \mathcal{L}_{total} , so as to realize the purpose of knowledge distillation.

COS-based NCE loss

Inspired by recent contrastive learning algorithms (Oord, Li, and Vinyals 2018; Tian, Krishnan, and Isola 2019; Sohn 2016), we design a contrastive loss COS-NCE for the transfer of the intermediate layer. In particular, we

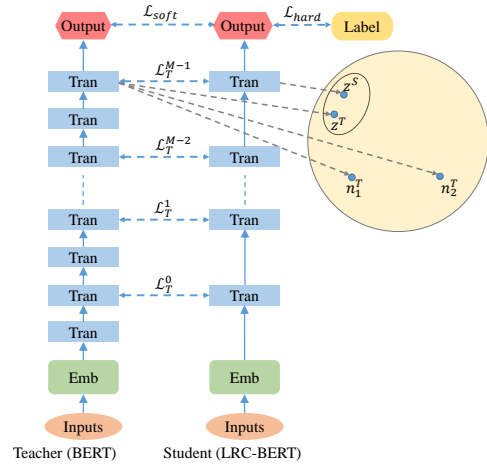


Figure 1: Overview of the proposed latent-presentation contrastive distillation framework for BERT model compression. Among them, contrastive learning is used to transfer the structural characteristics of output distribution from the teacher to the student. For instance, for the same input sample, the output distribution z^S of the student is more similar to z^T that of the teacher when comparing with the outputs n_1^T and n_2^T of the teacher for the negative input samples.

use the penultimate layer of distillation to elicit our proposed COS-NCE. For a given teacher network f^T and student model f^S , we randomly sample K negative samples $X = \{x_1^-, x_2^-, \dots, x_K^-\}$ for each input training sample x , and record the output of sample x in the penultimate layer of f^T and f^S networks as z^T and z^S , respectively. Correspondingly, we can obtain the output $N = \{n_1^T, n_2^T, \dots, n_K^T\}$ of negative samples in the penultimate layer of f^T network. As shown in Fig. 2, we map the outputs of the penultimate layer to certain points on the high-dimensional feature space. Different from the previous contrastive loss works considering Euclidean distance or mutual information in feature space (Sun et al. 2014), we propose a new contrastive loss COS-NCE based on angular distance distribution to consider the latent structural information output by the intermediate layer.

Formally, distilling the output of a given sample and K negative samples in the penultimate layer, the proposed COS-NCE formula is as follows:

$$\mathcal{L}_C(z^S, z^T, N) = \frac{\sum_{i=1}^K (2 - (g(n_i^T, z^S) - g(z^T, z^S)))}{2K} + g(z^T, z^S), \quad (1)$$

$$g(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}, \quad (2)$$

where $g(\cdot, \cdot) \rightarrow [0, 2]$ is a function used to measure the angular distance between two input vectors, and the smaller $g(x, y)$ means that the distribution distance of the two vectors is closer to each other. When $g(x, y) = 2$, it means that

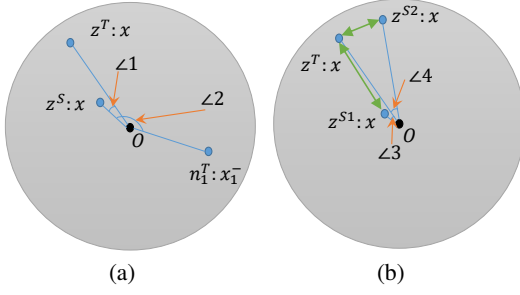


Figure 2: Map the output of the teacher network and the student model in the penultimate layer to the same feature space. The green line represents Euclidean distance. (a) the angular distance between positive pairs z^S and z^T with the same input x is smaller than that between negative pairs z^S and n_1^T with different inputs x and x_1^- , (b) for different student models f^{S1} and f^{S2} , the distance between the output z^{S2} and z^T is closer than that of z^{S1} under the evaluation standard based on Euclidean distance; however, under the standard based on angular distance, f^{S1} is better than f^{S2} .

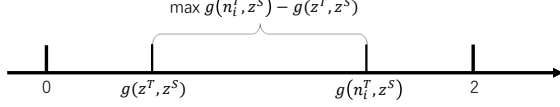


Figure 3: We use the difference between $g(n_i^T, z^S)$ (similarity with negative sample) and $g(z^T, z^S)$ (similarity with positive sample) to represent the distance from the negative sample.

the two vectors reach the maximum dissimilarity. The purpose of COS-NCE in formula (1) is to minimize the angle between z^S and the positive sample z^T , while increasing the angle between z^S and the negative sample $n_i^T \in N$. Instead of constructing the form of triplet loss based on Euclidean distance as used in previous work (Weinberger and Saul 2009; Schroff, Kalenichenko, and Philbin 2015), we apply the function $g(\cdot, \cdot)$ to minimize the angular distance between the training sample and positive samples. Then, in order to maximize the distance between the training sample and the negative samples (the first term in formula (1)), the loss is constructed from the geometric perspective as shown in Fig. 3. Obviously, the difference value $g(n_i^T, z^S) - g(z^T, z^S)$ between $g(n_i^T, z^S)$ and $g(z^T, z^S)$ need to be maximized. In order to transform the maximization problem into the minimization equivalent problem, we calculate the distance from $g(z^T, z^S)$ to the lower boundary of the function and $g(n_i^T, z^S)$ to the upper boundary. Therefore, the maximization problem is redefined as minimizing the sum of these two distances: $2 - (g(n_i^T, z^S) - g(z^T, z^S))$.

Distillation for transformer-layer

The COS-NCE is proposed to transfer the output of transformer layer knowledge from the teacher network to the student model. Each transformer layer in BERT consists of two

main sub-layers: multi head attention and fully connected feed-forward network (FFN), but we only distill the output of FFN. Assuming that the teacher model has N transformer layers and the student model has M transformer layers, we need to select M layers in the teacher for knowledge transfer and the objective is as follows:

$$\mathcal{L}_T^i = \mathcal{L}_C(h_i^S W, h_{\phi_i}^T, H_{\phi_i}^T), \quad (3)$$

where $h_i^S \in \mathbb{R}^{l \times d}$ is the output of the training sample x in the i -th transformer layer of the student network, $h_{\phi_i}^T \in \mathbb{R}^{l \times d'}$ is the output of the training sample x in the ϕ_i -th transformer layer of the teacher model, $j = \phi_i$ is a mapping function, which means that the output of the i -th layer of student network needs to imitate the output of the j -th layer of teacher model, l is the input text length, and scalar values d and d' represent the hidden size of student and teacher respectively, usually d is smaller than d' . $H_{\phi_i}^T = \{h_{0, \phi_i}^T, h_{1, \phi_i}^T, \dots, h_{K-1, \phi_i}^T\}$ is the output of K negative samples in the i -th transformer layer of teacher network. Referring to the exiting works (Jiao et al. 2019; Wang et al. 2019b), we use matrix $W \in \mathbb{R}^{d \times d'}$ to map the outputs of different dimensions to the same feature space.

Distillation for predict-layer

In order to predict specific downstream classification tasks, we simulate the student's prediction output from the soft label of the teacher model and the hard label corresponding to the real label respectively. Among them, the KL divergence is used to imitate the output of the teacher network, and cross-entropy loss is used to fit the difference between the student's output and the one-hot encoding of the real label.

$$\mathcal{L}_{soft} = \text{softmax}(y^T / \tau) \cdot \log\left(\frac{\text{softmax}(y^T / \tau)}{\text{softmax}(y^S / \tau)}\right), \quad (4)$$

$$\mathcal{L}_{hard} = -\text{softmax}(y / \tau) \cdot \log(\text{softmax}(y^S)), \quad (5)$$

where y^S and y^T are logits vectors output by the predict layer of student and teacher respectively, τ is a temperature that adjusts the concentration level, y is the *one-hot* encoding of real label. It should be noted that there is no negative sample involved in the distillation of the prediction layer. When the downstream task is a regression problem, the calculation of \mathcal{L}_{soft} and \mathcal{L}_{hard} is changed to mean squared error (MSE) loss function.

The various losses described above are shown in Fig 1. By combining the various losses of different layers described above, we can obtain the final distillation objective function for transforming the knowledge of the teacher network into the student model:

$$\mathcal{L}_{total} = \alpha \sum_{i=0}^{M-1} \mathcal{L}_T^i + \beta \mathcal{L}_{soft} + \gamma \mathcal{L}_{hard}, \quad (6)$$

where α , β and γ are used to weigh the influence of different losses. Specifically, in order to enable the model to learn the

| Model | Params | MNLI-m (393k) | MNLI-mm (393k) | QQP (364k) | SST-2 (67k) | QNLI (105k) | MRPC (3.7k) | RTE (2.5k) | CoLA (8.5k) | STS-B (5.7k) | Avg |
|-----------------------|--------|------------------|-------------------|---------------|----------------|----------------|----------------|---------------|----------------|-----------------|-------------|
| BERT-base (teacher) | 109M | 84.3 | 83.8 | 71.4 | 93.6 | 90.9 | 88.0 | 66.4 | 53.0 | 84.8 | 79.6 |
| DistilBERT | 52.2M | 78.9 | 78.0 | 68.5 | 91.4 | 85.2 | 82.4 | 54.1 | 32.8 | 76.1 | 71.9 |
| BERT-PKD | 52.2M | 79.9 | 79.3 | 70.2 | 89.4 | 85.1 | 82.6 | 62.3 | 24.8 | 79.8 | 72.6 |
| TinyBERT | 14.5M | 82.5 | 81.8 | 71.3 | 92.6 | 87.7 | 86.4 | 62.9 | 43.3 | 79.9 | 76.5 |
| LRC-BERT ₁ | 14.5M | 82.8 | 82.6 | 71.9 | 90.7 | 88.3 | 83.0 | 51.0 | 31.6 | 79.8 | 73.5 |
| LRC-BERT | 14.5M | 83.1 | 82.7 | 72.2 | 92.9 | 88.7 | 87.0 | 63.1 | 46.5 | 81.2 | 77.5 |

Table 1: The **results** are evaluated from the official website of GLUE benchmark, and the optimal experimental results are identified in bold. The number under each dataset represents the corresponding number of training samples.

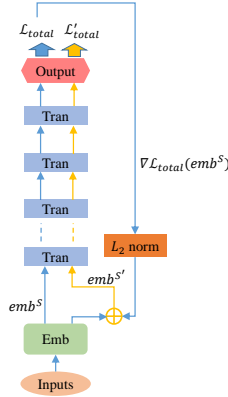


Figure 4: The illustration of training based on gradient perturbation.

contrastive structural features more effectively in the preliminary training, we will adopt a two-stage training method. In the first stage, we only calculate the contrastive loss of the intermediate transformer layer, that is, the values of α , β and γ are set to 1, 0 and 0 respectively. In the second stage, the weight values are set to be greater than 0 to ensure the model has the ability to predict downstream tasks.

Training based on Gradient Perturbation

In the training stage, we introduce gradient perturbation to increase the robustness of LRC-BERT. The method based on gradient perturbation has been applied in word embeddings (Miyato, Dai, and Goodfellow 2017), and the prediction performance is more stable. Fig. 4 shows the training process based on gradient perturbation. For the distillation loss \mathcal{L}_{total} obtained according to formula (6), we do not directly use the loss to carry out the parameter update by backward gradient propagation, but calculate the gradient of each element in the output matrix emb^S of the embedding layer in the student network, and use the gradient value to interfere with the distribution of emb^S . Then use the distribution after interference to calculate the loss according to the network forward propagation, and the loss is finally used to update the network parameters. The specific process is as follows:

$$emb^{S'} = emb^S + g / \|g\|_2, \quad (7)$$

$$g = \nabla \mathcal{L}_{total}(emb^S). \quad (8)$$

Where $emb^{S'}$ is the vector after using gradient perturbation to interfere, and the calculation of the gradient perturbation is L_2 norm processing for the gradient value emb^S . Before the parameter of the student network is updated, it is passed into the subsequent transformer layer and prediction layer again to calculate the loss \mathcal{L}'_{total} after the gradient perturbation.

Experiment

In this section, we elaborate on a large number of experiments to verify the effectiveness of the proposed LRC-BERT. The specific details are shown in the following subsections.

Datasets

We evaluate LRC-BERT on **GLUE** benchmark. The datasets provided on GLUE were all from NLP datasets with high recognition. We evaluate LRC-BERT in tasks such as natural language reasoning, emotion analysis, reading comprehension and semantic similarity.

In more detail, we evaluate on 8 classification tasks, including Corpus of Linguistic Acceptability (CoLA) (Warstadt, Singh, and Bowman 2019), Stanford Sentiment Treebank (SST-2) (Socher et al. 2013), Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett 2005), Semantic Textual Similarity Benchmark (STS-B) (Cer et al. 2017), Quora Question Pairs (QQP) (Chen et al. 2018), Question Natural Language Inference (QNLI) (Rajpurkar et al. 2016), Recognizing Textual Entailment (RTE) (Bentivogli et al. 2009) and Multi-Genre Natural Language Inference (MNLI) (Williams, Nangia, and Bowman 2017). Among them, MNLI is further divided into two parts: in-domain (MNLI-m) and cross-domain (MNLI-mm), so as to evaluate the generality of the test model. When evaluating the model, we use the metrics used by GLUE benchmark. Specifically, For QQP and MRPC, the metric is F1-score, for STS-B, the metric is Matthew’s correlation, the rest tasks use accuracy.

Distillation Setup

We use BERT-base (Devlin et al. 2019) as our teacher. BERT-base contains a total of about 109M parameters, including 12 layers of transformer ($N = 12$), and each layer has 768 hidden dimensions ($d' = 768$), 3072 intermediate sizes and 12 attention heads. The student model we instantiated has 4 transformer layers ($M = 4$), 312 hidden sizes

| Model | transformer layers | hidden size | Params | inference time(s) |
|-----------|--------------------|-------------|--------|-------------------|
| BERT-base | 12 | 768 | 109M | 121.4 |
| LRC-BERT | 4 | 312 | 14.5M | 12.7 |

Table 2: The number of parameters and inference time before and after model compression.

($d = 312$), 1200 intermediate sizes and 12 attention heads, with a parameter of approximately 14.5M. In order to test the generality of our model more effectively, we design two groups of experimental procedures. The first group uses the Wikipedia corpus to conduct pre-training, and then distills on specific tasks, which is called LRC-BERT. In the pre-training stage, only one epoch of distillation is carried out on the transformer layers, the values of α , β and γ are set to 1, 0 and 0 respectively. The second group is directly distilled under the specific task dataset and recorded as LRC-BERT₁.

For the distillation of each task on GLUE, we fine-tune a BERT-base teacher, choosing learning rates of 5e-5, 1e-4, and 3e-4 with batchsize of 16 to distill LRC-BERT and LRC-BERT₁. For each sample, we choose the remaining 15 samples in batchsize as negative samples, i.e. $K = 15$. Among them, 90 epochs of distillation are performed on the MRPC, RTE, and CoLA with the training dataset less than 10K, and 18 epochs of distillation on other tasks. For the proposed two-stage training method, the first 80% of the steps are chosen as the first stage of training, the rest 20% of the steps are the second stage. Specifically, the partition parameter value 80% applies to all tasks is obtained by searching on MNLI-m and MNLI-mm tasks, and the search range is {0%, 5%, 10% ,...,95%}. Then, we set the parameters of the second stage to $\alpha : \beta : \gamma = 1 : 1 : 3$, and the search range of each parameter is {1,2,3,4}. For the hyperparametric temperature τ , we set it to 1.1. Considering that there are fewer training samples for RTE and CoLA, we use the data augmentation method used by TinyBERT in LRC-BERT to expand the training data by 20 times and conduct 20 epochs of transformer layer distillation on the augmented dataset before specific task distillation.

We distill our student model with 6 V100 in the pre-training stage, and 4 V100 for distillation training on specific task dataset and extended dataset. In the inference experiments, we report the results of the student on a single V100.

Main Results

The results of the evaluation on GLUE are recorded in Table 1. By comparing with BERT-PKD (Sun et al. 2019), DistilBERT (Sanh et al. 2019) and TinyBERT (Jiao et al. 2019), which also have 4 transformer layers, the following three conclusions can be drawn: 1) in all evaluation tasks, LRC-BERT has better distillation effect than other methods, and the average prediction effect of LRC-BERT can reach 97.4% of the teacher model BERT-BASE, which indicates that the proposed method is effective for distillation on specific tasks. 2) For tasks with a large amount of training data, LRC-BERT₁, which does not use data augmen-

| Model | MNLI-m | MNLI-mm | MRPC | CoLA |
|-----------------------|--------|---------|------|------|
| LRC-BERT | 83.4 | 83.5 | 89.0 | 50.0 |
| LRC-BERT _C | 78.0 | 78.2 | 81.5 | 37.0 |
| LRC-BERT _S | 82.7 | 83.0 | 89.4 | 48.8 |
| LRC-BERT _H | 83.0 | 83.5 | 88.7 | 48.6 |

Table 3: Ablation studies of different loss functions (dev).

| Model | Accuracy |
|-----------------------|----------|
| LRC-BERT | 83.4 |
| LRC-BERT ₂ | 79.4 |

Table 4: Effect of two-stage training method on MNLI-m task (dev).

tation and pre-training, can significantly simplify the distillation process while still effectively transferring teacher knowledge to students. Specifically, for tasks with a training set greater than 100K LRC-BERT₁ has surpassed the experimental effect of TinyBERT, increasing by 0.3%, 0.8%, 0.6%, and 0.6% in MNLI-m, MNLI-mm, QQP and QNLI respectively. 3) Compared with the experimental results of LRC-BERT and LRC-BERT₁, the distillation performance of LRC-BERT can be significantly improved by pre-training and data augmentation, especially when the amount of training data is small. For example, the prediction results on tasks MRPC, RTE and CoLA are 4.0%, 12.1% and 14.9% higher than that of LRC-BERT₁, respectively.

Inference time is another important evaluation metric for model compression. Therefore, we further investigated the increase in model inference speed, and summarized the inference time of different baseline models in Table 2. Our 4-layer 312-dimensional student network is $9.6\times$ faster than the original teacher model BERT-BASE with the model size is $7.5\times$ smaller. This result is similar to TinyBERT, because our student model structure is the same as TinyBERT.

Ablation Studies

In this section, we conduct ablation tests on MNLI-m, MNLI-mm, MRPC and CoLA from two aspects: the construction of different loss functions and the use of gradient perturbation.

Effect of different loss function We remove the COS-NCE of transformer layer, soft loss or hard loss of prediction layer from the loss function for distillation shown in formula (1) to get three contrast models of LRC-BERT_C, LRC-BERT_S and LRC-BERT_H. The evaluation results of these four tasks on dev are shown in Table 3. The experimental results indicate that the COS-NCE plays the most important role in distillation. When COS-NCE is discarded, the performances are getting worse in all datasets, especially on the CoLA task, which is drop from 50 to 37. In addition, the absence of soft-loss or hard-loss shows little effect on the results. It is worthy to note that they have different degrees of influence on different tasks. For example, the prediction performance of soft loss used in MRPC and CoLA are outperform than that of hard loss, but the performance of hard loss used in MNLI-m and MNLI-mm is better. All of these

| sentence1 | sentence2 | MSE | angular distance(g) | label | prediction (BERT _M) |
|-----------------------|--|-------|-------------------------|---------------|---------------------------------|
| Paper goods. | Paper products. | 1.532 | 0.505 | entailment | entailment |
| oh constantly | Rarely | 1.557 | 0.525 | contradiction | contradiction |
| The truth? | Will you tell the truth? | 1.730 | 0.496 | neutral | contradiction |
| I'm not exactly sure. | I'm not exactly sure if you're aware of your issues. | 2.108 | 0.484 | entailment | neutral |

Table 5: Display of prediction results: In the given 4 cases, LRC-BERT can make accurate prediction, while BERT_M performs poorly, indicating that the angle-based distillation effect is better than the Euclidean distance.

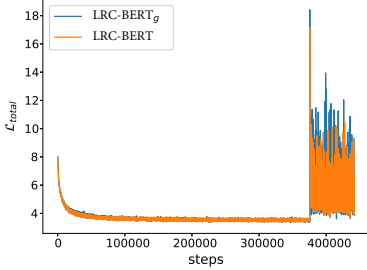


Figure 5: Effect of gradient disturbance on training loss. The variants are validated on the training set of the MNLI-m task and the loss fluctuation in the second stage of the two-stage training method is obviously increased.

experimental groups show that the three different losses used are useful in LRC-BERT.

Effect of gradient perturbation The introduction of gradient perturbation can change the data distribution of the middle layer in the training phase. In order to verify the stability improvement brought by the gradient perturbation on knowledge distillation, we construct LRC-BERT_g after removing the gradient perturbation from the training process. Fig. 5 shows the training loss of LRC-BERT and LRC-BERT_g on the MNLI-m task. It can be seen from the result that in the first stage of the two-stage training, their downward trend is relatively consistent, but in the second stage, after considering soft loss and hard loss, the training loss of LRC-BERT with gradient perturbation is more stable than LRC-BERT_g. It also shows that the gradient perturbation is feasible in knowledge distillation.

Analysis of Two-stage Training Method

The purpose of the two-stage training is to force LRC-BERT to focus only on the distribution characteristics of the data output in the transformer layer of the student model at the beginning of the training. In order to verify its effectiveness, we design LRC-BERT₂ without using the two-stage training method. In addition, during the entire training phase of LRC-BERT₂, the weighting parameter of the training loss \mathcal{L}_{total} is always set as $\alpha : \beta : \gamma = 1 : 1 : 3$. Table 4 shows the evaluation results of LRC-BERT and LRC-BERT₂ on the dev set of MNLI-m. Although we can conclude from the ablation studies that COS-NCE based on contrastive learning can effectively transfer the knowledge of the transformer layer in the teacher model to the student, the accuracy of

LRC-BERT₂ without using the two-stage training method is 4.0% lower than that of LRC-BERT, which indicates that the knowledge distillation of prediction layer from the beginning will greatly affect the results of contrast distillation in the transformer layer.

Analysis of angle-based distribution features

COS-NCE is proposed to capture the angular similarity feature of the output in the transformer layer, which is not considered by the existing knowledge distillation methods. For this reason, we design a comparison model BERT_M, which utilizes MSE for intermediate layer distillation. The average performance of BERT_M is 77.0, which is lower than 77.5 of LRC-BERT. Table 5 shows the four use cases extracted from the dev set of MNLI-m that can be correctly predicted by LRC-BERT, and record the prediction results of BERT_M. Specifically, when LRC-BERT and BERT_M predict the first two cases, the MSE and angular distance of the transformer layer are in line with the expected fluctuation range, both can make correct prediction. When the last two cases are predicted, the MSE of BERT_M increased significantly, while angular distance of LRC-BERT has no obvious change. LRC-BERT is trained based on COS-NCE and can still make correct prediction, which shows that the structural characteristics of the output distribution in the middle layer can be effectively used for knowledge distillation. However, in the knowledge distillation method based on Euclidean distance, the prediction result may be completely opposite, which briefly illustrates that the loss function based on angular similarity is feasible in knowledge distillation.

Conclusion

In this paper, we propose a novel knowledge distillation framework LRC-BERT to compress a large BERT model into a shallow one. Firstly, the COS-NCE based on contrastive learning is proposed to distill the output of the intermediate layer from the angle distance, which is not considered by the existing knowledge distillation methods. Then, we introduce a gradient perturbation-based training architecture in the training phase to increase the robustness of the model, which is the first attempt in knowledge distillation. Finally, to better capture the distribution characteristics of the intermediate layer, we design a two-stage training method for the total distillation loss. Extensive experiments on 8 datasets provided by GLUE benchmark show that the model is effective.

References

- Adriana, R.; Nicolas, B.; Ebrahimi, K. S.; Antoine, C.; Carlo, G.; and Yoshua, B. 2015. Fitnets: Hints for thin deep nets. *Proc. ICLR*.
- Aguilar, G.; Ling, Y.; Zhang, Y.; Yao, B.; Fan, X.; and Guo, C. 2020. Knowledge Distillation from Internal Representations. In *AAAI*, 7350–7357.
- Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.
- Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.
- Chen, T.; Goodfellow, I. J.; and Shlens, J. 2016. Net2Net: Accelerating Learning via Knowledge Transfer. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. URL <http://arxiv.org/abs/1511.05641>.
- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.
- Chen, Z.; Zhang, H.; Zhang, X.; and Zhao, L. 2018. Quora Question Pairs.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.
- Dolan, W. B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Gong, Y.; Liu, L.; Yang, M.; and Bourdev, L. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*.
- Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.
- Han, S.; Mao, H.; and Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*.
- Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, 1135–1143.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.
- Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339.
- Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
- Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Miyato, T.; Dai, A. M.; and Goodfellow, I. J. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL https://openreview.net/forum?id=r1X3g2_xl.
- Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.
- Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; and Huang, X. 2020. Pre-trained models for natural language processing: A survey. *arXiv preprint arXiv:2003.08271*.
- Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.
- Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.
- Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

- Sohn, K. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, 1857–1865.
- Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*.
- Sun, S.; Cheng, Y.; Gan, Z.; and Liu, J. 2019. 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)*, 4314–4323.
- Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, 1988–1996.
- Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2158–2170. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.acl-main.195/>.
- Tan, X.; Ren, Y.; He, D.; Qin, T.; Zhao, Z.; and Liu, T.-Y. 2018. Multilingual Neural Machine Translation with Knowledge Distillation. In *International Conference on Learning Representations*.
- Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; and Lin, J. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.
- Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive Representation Distillation. In *International Conference on Learning Representations*.
- Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive Representation Distillation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL <https://openreview.net/forum?id=SkgpBJrtvS>.
- Tsai, H.; Riesa, J.; Johnson, M.; Arivazhagan, N.; Li, X.; and Archer, A. 2019. Small and Practical BERT Models for Sequence Labeling. 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)*, 3623–3627.
- Turc, I.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*.
- Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019a. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.
- Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.
- Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019b. Heterogeneous graph attention network. In *The World Wide Web Conference*, 2022–2032.
- Warstadt, A.; Singh, A.; and Bowman, S. R. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics* 7: 625–641.
- Weinberger, K. Q.; and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10(2).
- Williams, A.; Nangia, N.; and Bowman, S. R. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.