装 订 线

目 录

装 订 线

装 订 线

1 研究背景

随着深度学习的发展，神经网络在各项任务上的精度逐步提高，不断刷新着各个 benchmark 的最高准确率。但是，深度学习技术的快速发展也带来了一些问题——具有较高精度的大规模深度学习模型只能停留在实验室的研究阶段，无法落地到实际应用场景中，这是因为高精度的模型往往具有复杂的网络结构和较深的网络层数。

然而，大规模深度学习模型在部署到边缘设备的过程中往往会面对以下窘境：边缘设备所拥有的储存容量有限，无法容纳下规模较大的深度学习模型；边缘设备提供的计算资源有限，无法满足一些应用场景对于实时性的需求。

在这样的背景下，模型压缩技术应运而生。对模型压缩技术的研究已经成为深度学习研究的一个大方向。这是一个比较偏向于应用层面的研究领域，它是将实验室成果落地到实际应用场景中的关键一环，意义重大。迄今为止，模型压缩领域已经有量化、剪枝、矩阵加速、知识蒸馏、动态推理加速等诸多方法，其中，知识蒸馏因为对模型压缩有较好的表现和很强的适应性而受到学术界的极大关注。作为一种能够在模型之间进行知识迁移的技术，知识蒸馏技术不仅能够完成模型的压缩，也是一种模型增强的技术。它从教师模型的角度出发，较小的学生模型获得了教师模型的知识，从而实现了对大模型的压缩；从学生模型的角度出发，学生模型因为教师模型的知识而有了更好的表现，从而完成了对学生模型的增强。

本文是同济大学研究生课程《机器学习理论与应用》的期末报告，我将聚焦于自然语言模型的知识蒸馏方法，复现近些年来的自然语言模型的知识蒸馏领域的相关工作，并在此基础上提出自己的创新想法。

装 订 线

2 研究现状

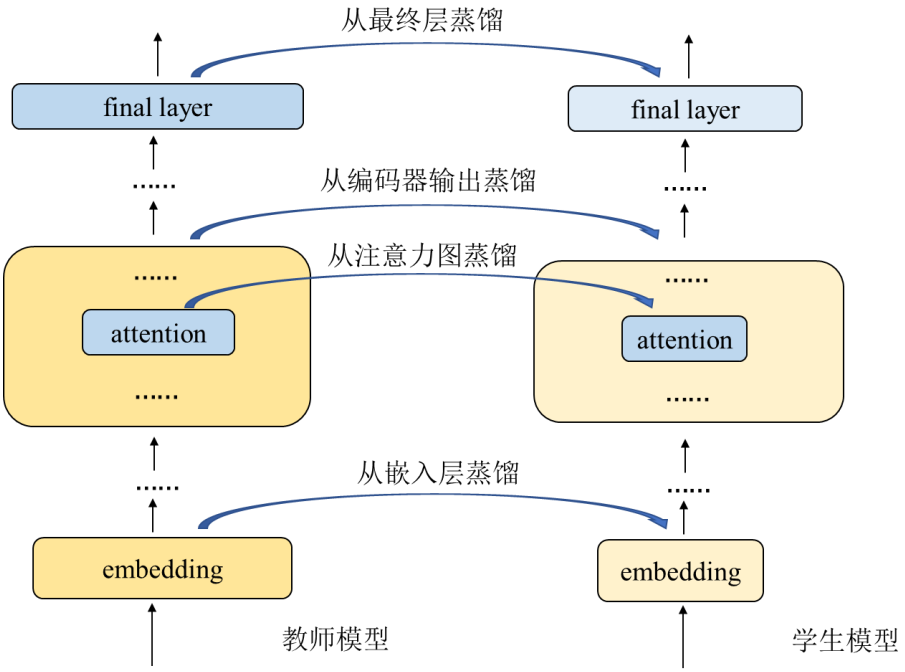
2.1 研究对象

在自然语言处理领域，现有的基于 Transformer 的自注意力机制架构的预训练模型数量庞大，包括 BERT，GPT-2，XLNet，Megatron-LM，Turing-NLG，T5 和 GPT-3 等。其中，谷歌的 BERT 模型因为开源最早、效果强大而影响广泛，成为模型压缩领域的一个重要研究对象。BERT 模型可以分为 BERT-base 和 BERT-large 两种，其模型参数大小分别为 110MB 和 340MB。

2.2 现有方法

知识蒸馏是模型压缩领域的一个研究热点，该方法利用一个或多个大模型（教师模型）的输出来训练小模型（学生模型）。该方法通过将教师模型的知识迁移到学生模型之上，从而达到在保证精度的同时，减小模型所占空间，提高模型的推理速度。

教师模型和学生模型之间迁移知识的形式具有多样性。学生模型学习的知识不仅可以是教师模型最后一层输出的 logits 值，还可以是模型的中间结果，包括中间层编码器的输出、嵌入层的输出和注意力图等，如下图所示。



使用的损失函数也具有多样性，可以是交叉熵，也可以是 KL 散度和 MAE 等，这往往和迁移知识的形式密切相关。此外，预训练模型都包含预训练和微调两个阶段，一般而言，BERT 模型的知识蒸馏是在微调阶段进行的。然而，同时兼顾预训练阶段和微调阶段的知识蒸馏则可以获得更好的模型表现，也可以实现保留特定领域、特定任务的知识。

学生模型的选择也具有多样性。学生模型可以选择基于 Transformer 构架的模型，只需要减少编码单元的深度或者减少编码单元的宽度即可，这样的学生模型在部署时就会占据更少的空间和使用更短的推理时间。学生模型也可以不使用 Transformer 构架的模型，而使用 BiLSTM 模型，BiLSTM 单独处理每一个输入而不用并行处理所有的输入，在推理速度上和模型压缩比例上都有更大提升，但其也会损失更多的模型精度。此外，RNN 等结构更加简单的模型也可以作为 BERT 模型知识蒸馏中的学生模型，而动态决定学生模型的结构也是研究的热点。

而近年来的研究更关注于蒸馏中的结构知识和动态损失构建。相关工作将知识蒸馏与课程学习相结合，构建动态的蒸馏损失；另外的一些工作利用词向量特征图或者注意力矩阵构建结构知识，从而进行知识蒸馏。

本次我要复现的第一篇论文就是将知识蒸馏与课程学习相结合的、能够自适应调整蒸馏损失的动态知识蒸馏，即《Dynamic Knowledge Distillation for Pre-trained Language Models》。该论文发表在 2021 EMNLP（CCF-B），该会议是 NLP 领域的顶会。

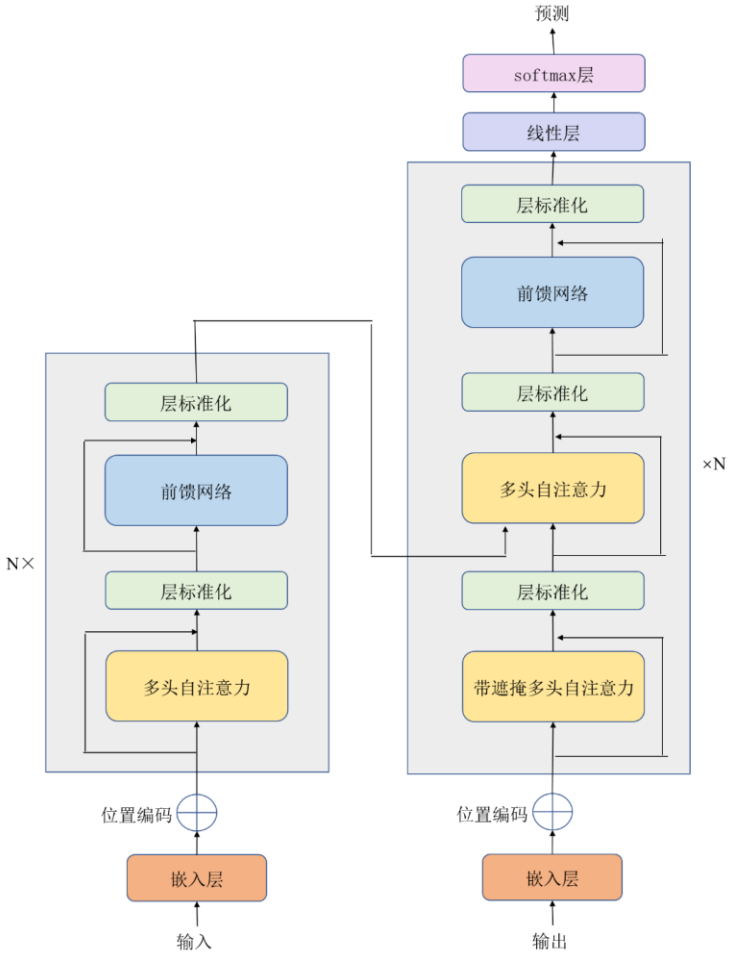
我要复现的第二篇论文就是利用词向量特征图和跨层词表示来构建结构关系从而进行知识蒸馏的语义上下文知识蒸馏，即《Distilling Linguistic Context for Language Model Compression》。该论文发表在 2021 NIPS（CCF-A）上。

装 订 线

3 算法原理

3.1 Transformer 模型

Transformer 模型是深度学习模型的一种，和仅依靠单个神经元全连接组成的神经网络不同，Transformer 模型具有十分复杂的网络组件和网络结构。Transformer 模型结构如下图所示，其结构包含了嵌入层、位置编码、多头自注意力模块、层标准化模块、前馈网络层、编码器-解码器结构等。



（1）嵌入层

嵌入层完成的工作是词嵌入，是将英文单词转换为张量，以便输入神经网络。Transformer 的嵌入层可以处理一个句子，它将一个句子中的单词全部转换为张量，从而得到一个矩阵。具体而言，嵌入层首先将单词转化为仅有一个位置是 1、其他位置全为 0 的 one-hot 向量：

$$e = [0,0,\dots,0,1,0,\dots 0]$$

然后通过可学习的权重矩阵W将 one-hot 向量映射到指定维度，得到词嵌入编码的结果w：

$$w = We'$$

经过词嵌入编码的词向量具有较小的维度，并且保有单词之间的相关性。

（2）位置编码

Transformer 可以并行处理一个句子，但词嵌入得到的编码没有考虑各个词语的位置关系，因此需要在词嵌入的基础上增加位置编码。位置编码需要为每个词向量赋予一个与位置相关的位置向量，这个向量的计算方法为：

$$p(pos,2i)=\sin\left(\frac{pos}{10000^{2i\times d^{-1}}}\right)$$

$$p(pos,2i+1)=\cos\left(\frac{pos}{10000^{2i\times d^{-1}}}\right)$$

其中，pos 为单词在句子中的位置，d 为词向量维度，i 为位置编码的维度。这样计算所得的位置编码向量和词嵌入向量具有相同的维度，并且每个位置的位置编码都是独一无二的。而最终得到的编码向量 x 采用词嵌入编码和位置编码在对应位置相加的方法得到：

$$x_{pos}=w_{pos}+p_{pos}$$

（3）多头自注意力模块

多头自注意力模块是提取句子中每个词语特征的模块，首先需要说明注意力的计算方法。在注意力函数的计算中，需要键值矩阵K、查询矩阵Q和值矩阵V作为输入，其中K和Q矩阵具有相同的列向量维度d_k。注意力函数的计算方法为：

$$Attention(K,Q,V)=softmax\left(\frac{KQ^T}{d_k}\right)\cdot V$$

注意力的计算方法能够使得句子中的单个词向量关注到自身和其他所有词向量的相关关系，从而使得特征提取更为有效。

Transformer 使用了多头自注意力机制。所谓“多头”是指计算多个不同的注意力，每个不同的注意力称为一个注意力头。各个注意头的计算结果通过拼接的方法连接，并通过一个可学习的矩阵变换W^O映射到目标维度，而目标维度和注意力模块的输入维度是一致的。

$$MultiHeadAttention(K,Q,V)=concat(head_1,\dots,head_h)\cdot W^O$$

所谓“自注意力”指的是注意力计算所需的 K,Q,V 三个矩阵均通过输入矩阵 X 通过可学习变换W^K，W^Q，W^V得到：

$$head_i=Attention\left(XW_i^K,XW_i^Q,XW_i^V\right)$$

相较于循环神经网络和卷积神经网络，注意力机制的存在使得模型能够注意到时序输入序列的相关关系并且较早地关注到输入数据的全局特征，这使得特征提取更为有效。同时，注意力机制能够并行处理时序输入，使得训练效率可以大大提升。

此外，在 Transformer 的解码器部分的多头自注意力模块是带有遮掩的，遮掩使得计算注意力时只能看到当前词向量之前的词向量。

（4）层标准化

在得到多头自注意力模块提取的特征之后，Transformer 模型将其与自注意力模块输入的特征进行了跳跃相加，并将相加结果进行层标准化。层标准化有利于加速模型的收敛。h为输入词向量的个数，层标准化首先需要计算输入的词向量的均值与方差：

$$\mu_L=\frac{1}{h}\sum_{i=1}^{h}a_i$$

$$\sigma_L = \sqrt{\frac{1}{h}\sum_{i=1}^{h}(a_i - \mu_L)^2}$$

然后计算层标准化的结果：

$$LN(a_i) = \alpha \cdot \frac{a_i - \mu_L}{\sqrt{\sigma_L^2 + \varepsilon}} + \beta$$

（5）前馈网络

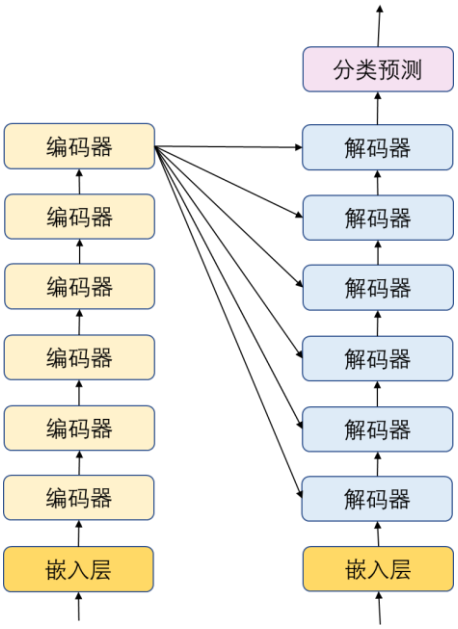
Transformer 的前馈神经网络进一步对词向量的特征进行提取，它使用了两个全连接的线性层，之间包含一个 ReLU 激活函数：

$$FFN(a_i) = \max(a_i W_1 + b_1, 0) \cdot W_2 + b_2$$

并且在线性层中使用了 dropout 的随机失活方法。

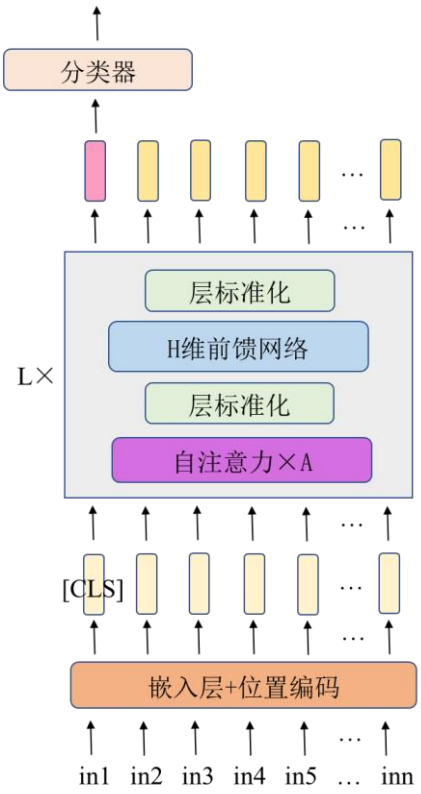
（6）编码器-解码器结构

作为处理“序列到序列”任务的 Transformer 模型，其使用了特殊的编码器-解码器结构，如下图所示，编码器和解码器均有六个。Transformer 使用六个结构相同的编码器提取输入语句的特征，将最后一个编码器提取到的特征分别输入六个解码器中，从而得到相应的输出结果。



3.2 BERT 模型

BERT 模型是一个主要用于分类任务的预训练语言模型，它的模型结构采用 Transformer 模型的编码器，如下图所示。根据模型的大小，该预训练模型有 BERT-base（L=12, A=12, H=768）和 BERT-large（L=24, A=16, H=1024）之分。



在大量的语料库上，BERT 模型通过以下两个任务进行自监督预训练：

①带 mask 的语言模型训练：在原始语料中选取 15%的单词进行遮掩，遮掩方式为：

1）80%的单词使用特殊符号 [mask] 代替；

2）10%的单词使用随机单词代替；

3）10%的单词保持不变。预训练的任务为根据上下文推测被遮掩位置的原始单词。

②判断是否为下一句话的训练：预训练任务为输入一个句子对(A, B)判断B是否是A的下一句话，具体的设定为：50%的B是原始文本中真实跟随A的下一句话，50%的B是原始文本中随机抽取的一句话。

通过以上任务，BERT 模型可以学习到语言的语义和语句之间的关联性，从而能够在下游分类任务中具有很好的表现。

3.3 知识蒸馏

知识蒸馏是一种在深度学习模型间进行知识迁移的方式。在模型压缩领域，知识蒸馏通常是将大型教师模型 T 的知识迁移到小型学生模型 S 中。具体而言，学生模型需要在训练过程中尽可能地模仿教师模型的行为和表现。形式上，知识蒸馏可以表示为以下目标函数的优化问题：

$$\min L_{KD} = \min \sum_{x \in X} L\left(f^T(x), f^S(x)\right)$$

其中，X代表了数据集，x是数据集里的数据样本，$f^T(\cdot)$和$f^S(\cdot)$分别代表了教师模型的行为表现和学生模型的行为表现，$L(\cdot)$代表两者之间行为差异的度量方法。知识蒸馏的目标就是在学生模型的训练过程中，不断缩小学生模型与教师模型的在行为表现上的差异，从而让学生模型学习到教师模型的知识。对于分类问题而言，L_{KD} 常用学生模型的概率分布与教师模型概率分布软化后的软目标之间的 KL 散度：

$$L_{KL} = KL(\sigma(S(x)) \,||\, T_{soft})$$

其中，$S(\cdot)$ 表示学生模型是输出 logits 值，$\sigma(\cdot)$ 表示将 logits 值映射到概率分布的归一化函数，T_{soft} 表示软目标，$KL(\cdot)$ 表示 KL 散度的计算。软目标一般采用 Softmax-T 获得，其中的每一项概率密度的计算方法为：

$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum \exp\left(\frac{z_i}{T}\right)}$$

其中，q_i 表示软目标的第 i 项值，z_i 表示教师模型输出的概率分布的第 i 项值，T 表示蒸馏温度这一与分布软化程度相关的值。

此外，与教师模型相同，数据集标注也能为学生模型的训练提供监督，因此在知识蒸馏的一般做法中，需要最小化学生模型输出和数据标注的交叉熵损失：

$$\min L_{CE} = min \sum_{x \in X} -y \log\, \sigma(S(x))$$

其中，y 表示数据标注。该优化目标的存在使得学生模型能够向正确结果靠近。

一般地，知识蒸馏所优化的损失函数的形式为：

$$L = \alpha L_{KD} + (1-\alpha) L_{CE}$$

参数 α 表示了各部分损失所占的比例。

3.4 BERT 知识蒸馏

BERT 模型结构的复杂性导致基于 BERT 结构的模型之间的知识蒸馏具有特殊性。

（1）BERT 结构间的知识蒸馏的形式化定义

假设教师模型具有 N 个 Transformer 的编码器，学生模型具有 M 个 Transformer 的编码器。由于要为学生指定知识的来源，使用函数 g 定义学生的特定层到教师的特定层之间的映射：

$$\begin{cases} n = g(m) \\ 0 = g(0) \\ N+1 = g(M+1) \end{cases}$$

这里 m 代表学生模型的某一层，n 代表教师模型的对应层，0 代表 Transformer 模型的嵌入层，$N+1$ 和 $M+1$ 代表最终输出的分布。在这样的定义下，蒸馏损失可以被形式化定义为：$L_{KD} = \sum_{x \in X} \sum_{m=0}^{M+1} \omega_m L_{layer}\left(f_m^S(x), f_{g(m)}^S(x)\right)$

对于不同的层，$L_{layer}(\cdot)$ 和 $f^T(\cdot)$、$f^S(\cdot)$ 的选取具有多样性，只需要经过标准化就可以同时使用不同层的不同蒸馏损失作为知识蒸馏的总损失。

（2）嵌入层的蒸馏损失形式

对于 BERT 结构的嵌入层，可以使用嵌入层输出的特征图构造蒸馏损失。假设 E^S 表示学生模型嵌入层的特征图，E^T 代表教师模型输出的特征图。一般而言，E^T 列向量的维度大于 E^S 列向量的维度，因此需要利用可学习的矩阵 W^E 将 E^S 映射到 E^T 的维度。一般使用二者的均方误差作为嵌入层的蒸馏损失，对于单个数据样本的蒸馏损失定义如下：

$$L_{embedding} = MSE(E^S W^E, E^T)$$

（3）中间层的蒸馏损失形式

对于 BERT 结构的中间层，同样可以使用中间层输出的特征图构造蒸馏损失，假设 H^S 表示学生模型中间层的特征图，H^T 代表教师模型输出的特征图。一般而言，H^T 列向量的维度大于 H^S 列向量的维度，因此需要利用可学习的矩阵 W^H 将 H^S 映射到 H^T 的维度。一般使用二者的均方误差作为中间层的蒸馏损失，考虑到学生模型一共有 M 个中间层，对于单个数据样本的蒸馏损失的定义如下：

$$L_{hidden} = \sum_{m=1}^{M} MSE(H_m^S W_m^H, H_m^T)$$

除此之外，注意到分类任务中 [CLS] 的重要性，我们同样可以利用 [CLS] 所对应的特征向量来构造蒸馏损失。假设学生模型 [CLS] 对应的特征向量为：

$$h^{\mathrm{S}} = [h_1, h_2, \ldots, h_M]$$

分别与之相对应的教师模型的 [CLS] 特征向量为：

$$h^{\mathrm{T}} = \left[h_{g(1)0}, h_{g(2)}, \ldots, h_{g(M)}\right]$$

可以使用 [CLS] 对应特征在正则化后的 2-范数来定义蒸馏损失，则对于单个数据样本的蒸馏损失的定义如下：

$$L_{hidden} = \sum_{m=1}^{M} \left\| \frac{h_{\mathrm{m}}}{\|h_m\|_2} - \frac{h_{\mathrm{g(m)}}}{\left\|h_{g(m)}\right\|_2} \right\|_2^2$$

（4）输出层的蒸馏损失形式

对于 BERT 的输出层，其输出为预测的概率分布。因此，可以使用 KL 散度：

$$L_{KL} = KL(\,\sigma(S(x))\;||\,T_{soft})$$

（5）注意力矩阵的蒸馏损失形式

注意力矩阵是 BERT 结构的核心，因此可以考虑利用注意力矩阵设计蒸馏损失。令注意力权重为：

$$A = \frac{KQ^T}{d_k}$$

通常情况，我们可以利用学生模型中间层的注意力权重 A^S 和教师模型中间层的注意力权重 A^T 的均方误差构造蒸馏损失。同样考虑到学生模型有 M 个中间层，则对于单个注意力头、单个数据样本的蒸馏损失的定义如下：

$$L_{attention} = \sum_{m=1}^{M} MSE(A^S, A^T)$$

除了均方误差之外，也可以使用 KL 散度作为距离度量标准：

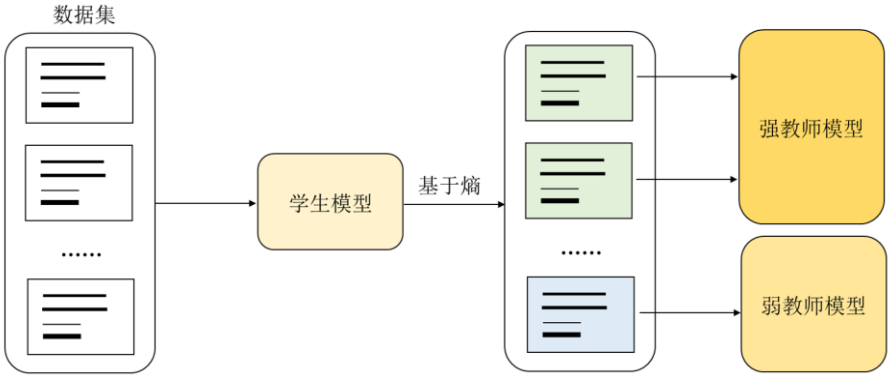
$$L_{attention} = \sum_{m=1}^{M} KL(A^S \,||\, A^T)$$

3.5 BERT 的动态知识蒸馏

3.5.1 动态教师选择

在知识蒸馏中，对于同样的学生模型，规模较大、表征能力较强的教师模型的知识蒸馏效果不一定比规模较小、表征能力较弱的知识蒸馏效果好，学生模型与教师模型的匹配程度才是决定蒸馏效果好坏的直接因素。因此在多教师知识蒸馏时，要选择与学生模型更匹配的教师模型作为目标。动态教师选择就是在这样的问题背景下提出的。

下图展示了基于熵不确定度的动态教师选择：



受到主动学习的启发，可以使用学生模型对分类结果的不确定性 u_x 作为衡量标准，选择合适的教师模型进行知识蒸馏。u_x 的计算方式可以是分类结果的熵：

$$u_x = \text{Entropy}\left(\sigma\left(S(x)\right)\right)$$

具体的选择策略有硬选择和软选择两种。

（1）硬选择

根据学生预测的不确定性 u_x 对一轮训练中的数据进行排序。然后根据排序结果将数据分为学生不确定部分和学生有信心部分，各占 50%。对于不确定的部分，利用弱教师产生的软目标进行学习；对于有信心的部分，使用强教师提供软目标进行学习。

（2）软选择

软选择利用两个教师蒸馏损失，只是动态决定所占比例：

$$L_{KD} = \omega_1 L_{KD}^{T1} + \omega_2 L_{KD}^{T2}$$

其中，ω_1 和 ω_2 两个与 u_x 相关的参数决定两个教师模型监督的贡献，其计算方式如下：

$$\omega_1 = \frac{u_x}{U} \quad \omega_2 = 1 - \frac{u_x}{U}$$

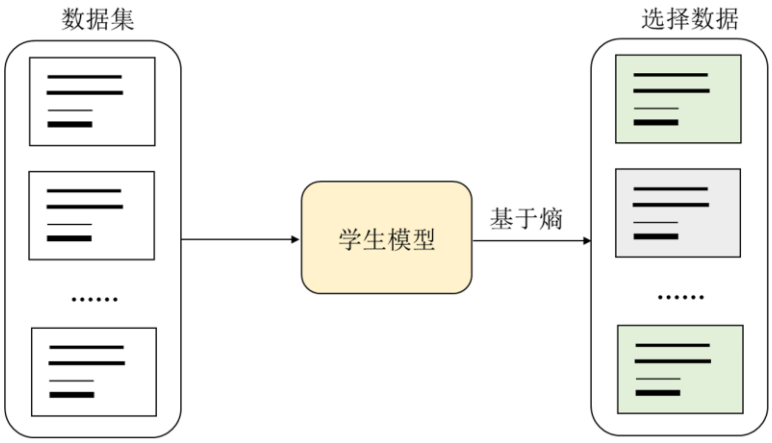
其中，U 是将权重重新缩放到 $[0,1]$ 的归一化因子。在这种策略下，学生在对当前数据样本不确定时会更加关注弱教师，而在对其预测有信心时会更加依赖强教师。

3.5.2 动态数据选择

在知识蒸馏中，存在着对知识迁移而言重要的数据。对于不重要的数据，只需要简单进行学习；而对于重要的数据，需要反复学习。为了提高学习的效率、避免冗余，知识蒸馏应该选取那些真正重要的数据进行学习。在这样的背景下，动态数据选择被提出。

同样受到主动学习的启发，学生模型对分类结果的不确定性 u_x 作为数据重要性的衡量标准。在动态数据选择策略里，需要人为指定数据选取率 r 这一超参数，即我们需要人为决定数据集中有多少数据是重要的。然后根据学生预测的不确定性 u_x 对一轮训练中的数据进行排序，选择不确定性高的前 r 部分数据作为知识蒸馏的数据样本。

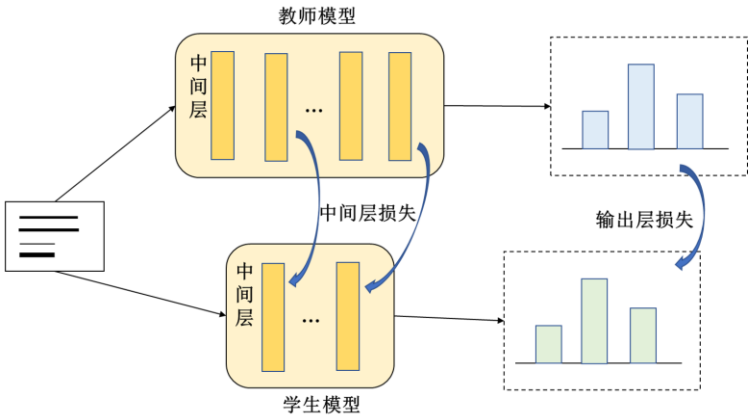
下图展示了基于熵不确定性的动态数据选择：



3.5.3 动态监督调整

在知识蒸馏中，迁移知识的形式不同，知识蒸馏的效果也是不同。在知识蒸馏的早期，我们希望学生模型尽可能模仿教师模型的行为，以尽快学习到教师模型的知识；在知识蒸馏的后期，我们只要就学生模型能给出正确的结果而不要求过程的一致，这有利于学生模型泛化能力的提高。因此，在知识蒸馏的过程中，我们需要在不同阶段选择不同的知识迁移形式进行蒸馏。动态监督调整的基本思想就是这样。

下图展示基于熵不确定度的动态监督调整：



动态监督调整使用中间层损失作为蒸馏前期的知识迁移形式，选择预测层损失作为知识蒸馏后期的知识迁移形式：

$$L_{KD} = \lambda_{pred} L_{pred} + \lambda_{hidden} L_{hidden}$$

主动学习仍然在这一部分得到应用，λ_{pred} 和 λ_{hidden} 是两个和学生预测的不确定性 u_x 相关的参数：

$$\lambda_{pred} = \lambda^*_{pred}\left(1 - \frac{u_x}{U}\right) \quad \lambda_{hidden} = \lambda^*_{hidden}\frac{u_x}{U}$$

其中，λ^*_{KL} 和 λ^*_{PT} 是通过参数搜索获得的每个目标的预设的权重，U 是归一化因子。通过这种方式，这两个损失的贡献在每个样本的训练期间动态调整。对于学生有信心的样本，来自中间表示对齐的监督被降低权重，此时学生专注于根据自己对样本的理解，只模仿老师最终的预测概率分布。相反，对于学生混淆的样本，教师模型的中间表示的监督可以帮助它更好地学习样本的特征。

3.6 BERT 的语义上下文知识蒸馏

3.6.1 结构损失的构建

利用 BERT 的词向量表示，可以结构两种不同类型的结构损失，分别是二元损失和三元损失。所谓二元损失就是利用两个词向量来构建关系，而三元损失就是利用三个词向量来构建关系。

（1）二元损失

二元损失是基于向量间距离的。给定两个词向量(r_i, r_j)，记二元关系为$\varphi(r_i, r_j)$，则二元关系可以用词向量的余弦相似度来表示：

$$\varphi(r_i, r_j) = \cos(r_i, r_j)$$

也可以用 L2 距离来表示：

$$\varphi(r_i, r_j) = \|r_i, r_j\|_2$$

（2）三元损失

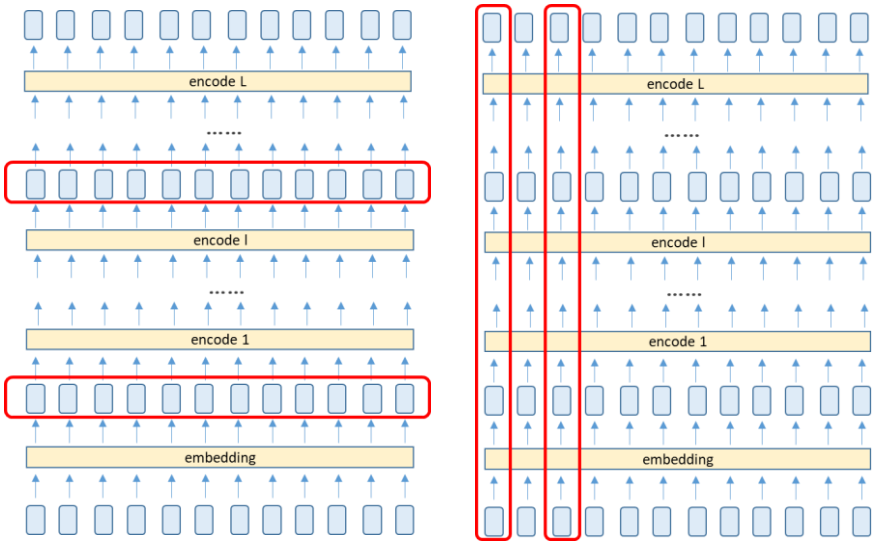
三元损失是基于向量间的夹角的。给定三个词向量(r_i, r_j, r_k)，记三元关系为$\psi(r_i, r_j, r_k)$，则三元几何角可以表示为：

$$\psi(r_i, r_j, r_k) = \angle\cos(r_i, r_j, r_k) = \langle \frac{r_i - r_j}{\|r_i - r_j\|_2}, \frac{r_k - r_j}{\|r_k - r_j\|_2} \rangle$$

对于三元几何角关系，得到的几何角矩阵规模为$O(d_r n^3)$，其中 n 是句子长度，而d_r是单个词向量的长度。这样的规模过于庞大，在实际应用中，一般通过选择邻域词向量和重要词向量的方法来降低几何角矩阵的规模至$O(d_r d_1 d_2^2)$。

3.6.2 词向量的选取

词向量的选取也有两种不同的方式。一种是利用不同单词在特定 BERT 层输出的词表示构建结构关系，另外一种是利用同一单词在不同 BERT 层输出的词表示构建结构关系。如下图所示。



左图表示的是前者，右图表示的是后者。

3.6.3 全局损失

全局损失由输出层的概率分布损失和结构损失两部分构成，结构损失也包含两部分：

$$L = (1 - \alpha)L_{logic} + \alpha L_{CKD}$$

$$L_{CKD} = L_{WR} + L_{LTR}$$

其中，L_{WR}是层内二元关系和三元几何角构成的损失，L_{LTR}是跨层二元关系和三元几何角构成的损失。

装 订 线

4 实验 1：过程

4.1 实验环境

实验主要使用 Python 的 Pytorch 库和 Huggingface 的 Transformers 库来完成，后者是一个包含了多种基于 Transformer 结构的预训练模型的库，并且该库对模型的分词器、模型结构和训练过程有高度封装，使用极为方便。Python 的版本为 3.6，Pytorch 的版本为 1.8.0，Huggingface Transformers 的版本为 4.5.1。服务器的操作系统为 Ubuntu18.04，显卡有两张 NVIDIA GeForce RTX 3090，采用分布式训练。

4.2 数据集

本次实验采用的数据集是 GLUE 基准数据集。

GLUE 是自然语言处理领域的一个标准数据集，有九项分类任务，涵盖自然语言推断、文本蕴含、情感分析、语义相似等方面。下表展示了 GLUE 数据集的具体信息。

数据集	训练集	验证集	测试集	任务	评价指标	数据来源领域
CoLA	8551	1043	1063	语法正确性判断	马修斯相关系数	混合
SST-2	67350	873	1821	情感分类	准确率	电影评论
MRPC	3668	408	1725	释义关系判断	准确率、F1 值	新闻
STS-B	5749	1379	1377	句子相似度判断	相关系数	混合
QQP	363870	40431	390965	句子相似度判断	准确率、F1 值	社区问答网站
MNLI	392702	9815	9796	前提假设关系判断	匹配精度	混合
QNLI	104743	5463	5461	蕴含关系判断	准确率	维基百科
RTE	2491	277	3000	蕴含关系判断	准确率	新闻，维基
WNLI	635	71	146	蕴含关系判断	准确率	科幻小说

九项任务在数据集规模、任务类型、评价指标和数据来源方面均有或多或少的差异。GLUE 的九项任务具有任务难度大的特点，该数据集的单任务训练 baseline 如下表所示。

模型	平均	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE
BiLSTM	63.9	15.7	85.9	69.3/79.4	81.7/61.4	66.0/62.8	70.3/70.8	75.7	52.8
+ ELMo	66.4	**35.0**	**90.2**	69.0/80.8	**87.5**/65.6	64.0/60.2	72.9/73.4	71.7	50.1
+ CoVe	64.0	14.5	88.5	**73.4/81.4**	83.3/59.4	**67.2/64.1**	64.5/64.8	75.4	**53.5**
+ Attn	63.9	15.7	85.9	68.5/80.3	83.5/62.9	59.3/55.8	74.2/73.8	77.2	51.9
+ Attn, ELMo	66.5	**35.0**	**90.2**	68.8/80.2	86.5/**66.1**	55.5/52.5	**76.9/76.7**	**76.7**	50.4
+ Attn, CoVe	63.2	14.5	88.5	68.6/79.7	84.1/60.1	57.2/53.6	71.6/71.5	74.5	52.7

由表格可见，基线模型仅在 SST-2 和 QQP 数据集上表现较好，在 CoLA 和 RTE 数据集上表现很差，仍有很大的提升空间。因此只有性能极其强大的模型才能在 GLUE 基准数据集上有较高的得分。

由于时间有限，本次仅使用 GLUE 中的 RTE 数据集。

4.3 实验配置

（1）强教师

参数	含义	数值
model	模型	24 层 BERT
init_model	模型初始化参数	bert-large-uncased
max_seq_length	输入句子长度	128
per_device_train_batch_size	训练集批大小	16
per_device_eval_batch_size	验证集批大小	64
warmup_ratio	线性预热率	0.05
learning_rate	学习率	5e-5
num_train_epochs	训练轮数	3

（2）弱教师

参数	含义	数值
model	模型	12 层 BERT
init_model	模型初始化参数	bert-base-uncased
max_seq_length	输入句子长度	128
per_device_train_batch_size	训练集批大小	16
per_device_eval_batch_size	验证集批大小	64
warmup_ratio	线性预热率	0.05
learning_rate	学习率	5e-5
num_train_epochs	训练轮数	3

（3）动态教师选择

参数	含义	数值
student_model	模型	6 层 BERT
init_model	模型初始化参数	bert-base-uncased
small_teacher	弱教师	$SMALL_TEACHER
large_teacher	强教师	$LARGE_TEACHER
small_teacher_alpha	弱教师权重	1

large_teacher_alpha	强教师权重	1
kd_alpha	蒸馏损失权重	1
ce_alpha	交叉熵损失权重	1
uncertainty_mode	教师策略	“hard” or “soft”
max_seq_length	输入句子长度	128
per_device_train_batch_size	训练集批大小	16
per_device_eval_batch_size	验证集批大小	64
warmup_ratio	线性预热率	0.05
learning_rate	学习率	2e-5
num_train_epochs	训练轮数	3

（4）动态数据选择

参数	含义	数值
student_model	模型	6 层 BERT
init_model	模型初始化参数	bert-base-uncased
teacher_model	弱教师	$SMALL_TEACHER
kd_alpha	蒸馏损失权重	1
ce_alpha	交叉熵损失权重	1
select_method	数据选择策略	margin
select_ratio	批数据选取率	0.5
max_seq_length	输入句子长度	128
per_device_train_batch_size	训练集批大小	16
per_device_eval_batch_size	验证集批大小	64
warmup_ratio	线性预热率	0.1
learning_rate	学习率	2e-5
num_train_epochs	训练轮数	3

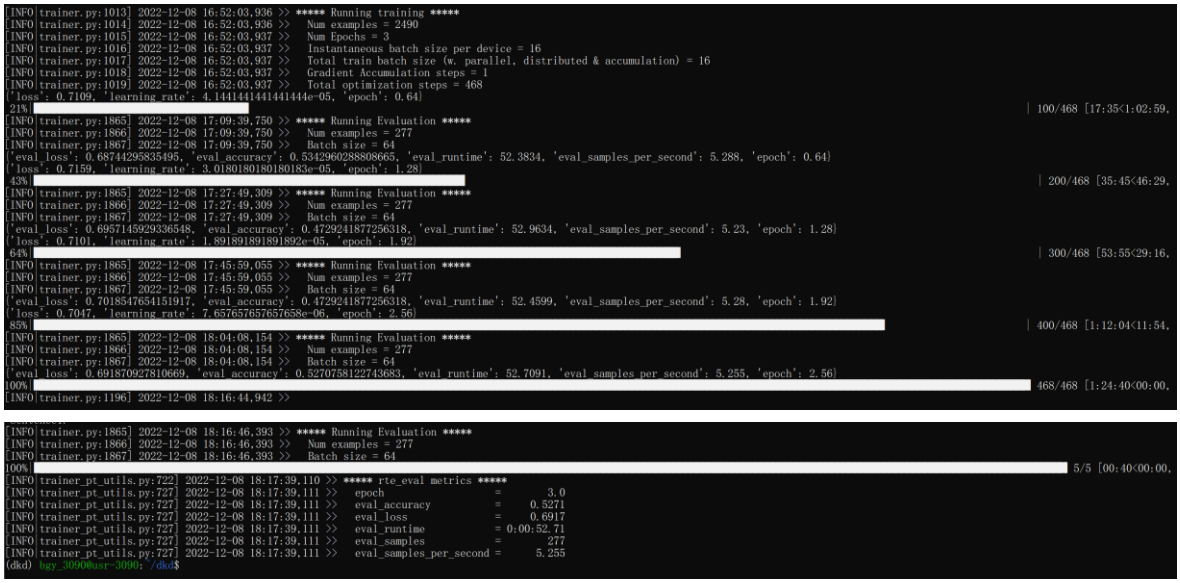
（5）动态监督调整

参数	含义	数值
student_model	模型	6 层 BERT
init_model	模型初始化参数	bert-base-uncased
teacher_model	弱教师	$SMALL_TEACHER
method	损失动态调整标准	uncertainty
kd_kl_alpha	初始预测层蒸馏损失	1
kd_rep_alpha	初始中间层蒸馏损失	4

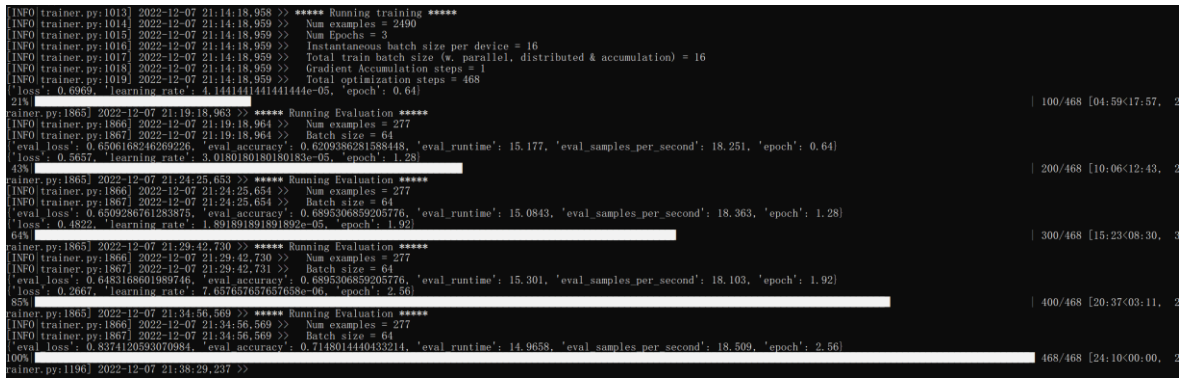
ce_alpha	交叉熵损失权重	1
max_seq_length	输入句子长度	128
per_device_train_batch_size	训练集批大小	16
per_device_eval_batch_size	验证集批大小	64
warmup_ratio	线性预热率	0.1
learning_rate	学习率	2e-5
num_train_epochs	训练轮数	3

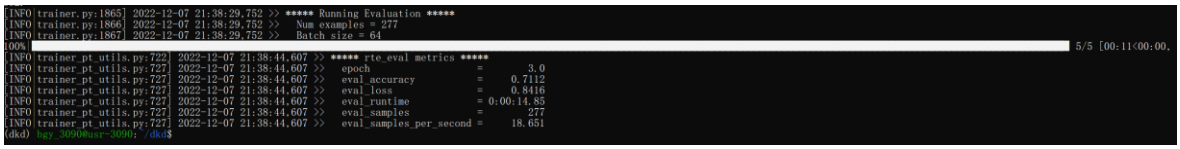
4.4 实验过程

（1）建立强教师模型

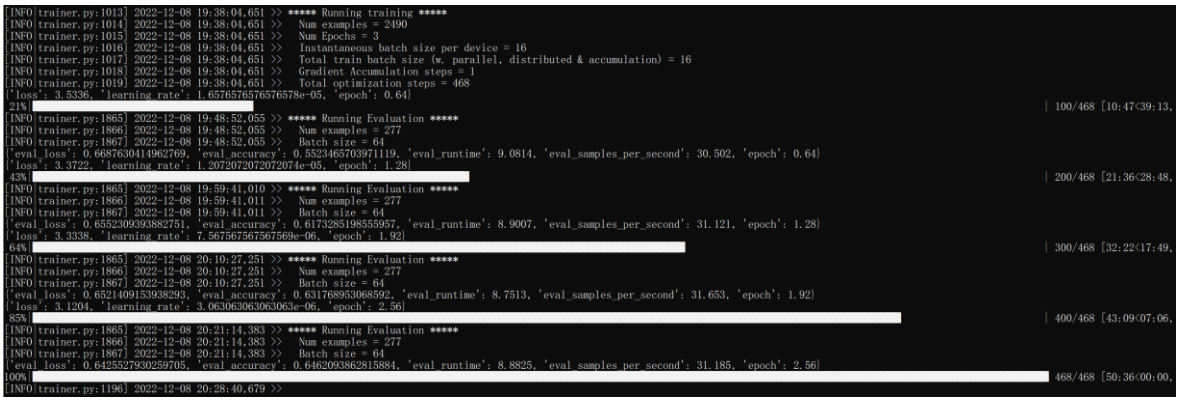


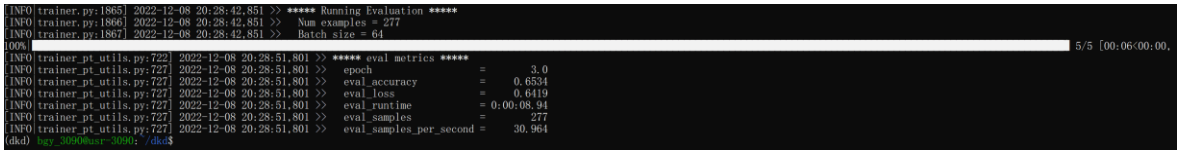
（2）建立弱教师模型



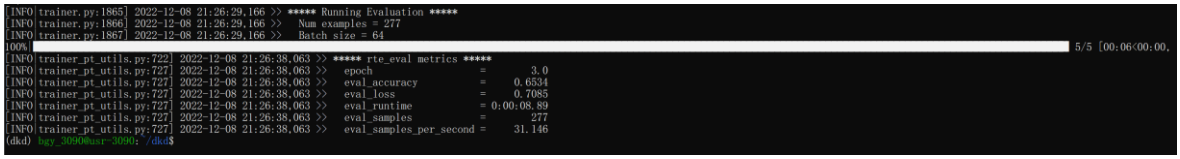


（3）动态知识蒸馏（动态教师选择）获取学生模型

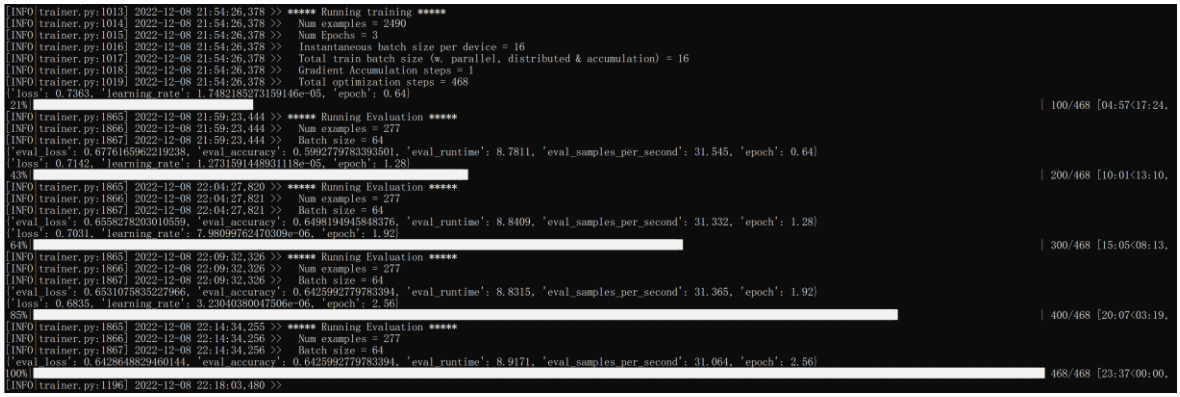


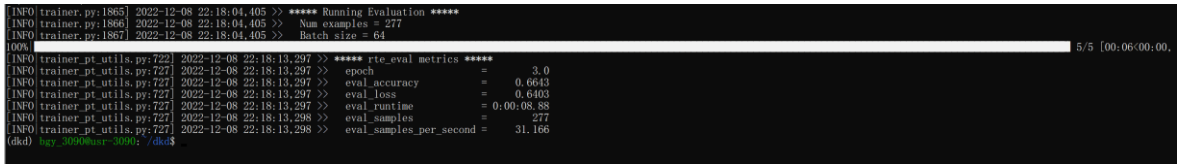


（4）动态知识蒸馏（动态数据选择）获取学生模型



（5）动态知识蒸馏（动态监督调整）获取学生模型





装 订 线

5 实验 1：结果与分析

5.1 评价指标

通常而言，知识蒸馏框架的评价指标需要考虑模型压缩率、模型加速比、学生模型精度等多个方面的指标。

在本次复现的动态知识蒸馏中，由于学生模型采用统一的结构，即均采用了 6 层的 BERT 模型架构作为学生模型，因此知识蒸馏在模型压缩率、模型加速比两个方面的性能是一致的。所以本实验只需从学生模型的精度上来进行评价即可。

我们使用的数据集是 RTE 数据集，该数据集上的任务为判断两个句子间的蕴含关系。若两个句子相互蕴含，则输出 1；若两个句子没有相互蕴含，则输出 0。因此，这是一个二分类任务，其评价指标为分类正确率，即 accuracy。

综上所属，我们采用学生模型在验证集上 accuracy 准确率作为知识蒸馏性能好坏的评价标准。

5.2 实验结果

在 5 个随机数种子上进行模型的训练，得到的结果如下：

模型及方法	平均准确率	最高准确率	准确率排序
强教师模型+直接微调	72.5%±1.4%	75.1%	1
弱教师模型+直接微调	67.8%±1.5%	70.1%	2
学生模型+直接微调	63.7%±1.2%	64.8%	7
学生模型+知识蒸馏（强）	66.0%±0.7%	67.1%	4
学生模型+知识蒸馏（弱）	65.4%±0.3%	65.7%	5
学生模型+动态教师选择	66.6%±1.1%	67.9%	3
学生模型+动态数据选择	64.4%±1.0%	65.7%	6
学生模型+动态监督调整	60.8%±2.0%	63.2%	8

5.3 实验分析与评价

本次实验结果如 5.2 节所示。

在性能方面，具有更庞大结构的强教师模型具有最高的平均准确率和最高准确率，而弱教师模型次之，具有第二高的平均准确率和最高准确率。这是合理的，作为规模更加庞大的教师模型，其性能优于学生模型是情理之中的事情。

而直接对学生模型进行微调和知识蒸馏，得到的性能远不如教师模型，这表明为了拉近学生

模型和教师模型之间的性能差异还有很多探究值得展开。

对于本文所提出的三种动态知识蒸馏方法，为了保证实验配置的一致性，我并没有对 RTE 数据集进行数据扩充，得到的实验结果和原文略有不同，具体如下：

- 动态教师选择：动态教师选择的知识蒸馏性能与原文一致，它的性能比直接对学生模型进行知识蒸馏要好，但比不上教师模型。
- 动态数据选择：动态数据选择由于没有进行数据增强，其性能低于直接进行知识蒸馏，但比直接训练模型的性能要高。
- 动态监督调整：动态监督调整由于没有进行数据增强，并且对于超参数的搜索并不到位，这使得模型在蒸馏过程中具有很大的不确定性，这使得模型的性能要低于直接对学生模型进行微调。

从复现结果可知，动态知识蒸馏虽然从理论上可以使知识蒸馏具有更加好的性能，但该方法无疑也增加了方法的不确定性、增加了超参数空间的维度，这使得方法性能不够稳定。

装 订 线

装 订 线

6 实验 2：过程

6.1 实验环境

实验主要使用 Python 的 Pytorch 库和 Huggingface 的 Transformers 库来完成，后者是一个包含了多种基于 Transformer 结构的预训练模型的库，并且该库对模型的分词器、模型结构和训练过程有高度封装，使用极为方便。Python 的版本为 3.7，Pytorch 的版本为 1.7.0，Huggingface Transformers 的版本为 3.0.2。服务器的操作系统为 Ubuntu18.04，显卡有两张 NVIDIA GeForce RTX 3090，采用分布式训练。

6.2 数据集

同 4.2，使用 RTE 数据集。

6.3 实验配置

参数	含义	数值
student_model	模型	6 层 BERT
init_model	模型初始化参数	bert-base-uncased
teacher_model	教师模型	$SMALL_TEACHER
alpha	结构损失权重	0.9
temperature	蒸馏温度	3
wrdist_w	层内二元结构损失权重	1
ltrdist_w	跨层二元结构损失权重	128
wrangle_w	层内几何角结构损失权重	16
ltrangle_w	跨层几何角结构损失权重	64
learning_rate	学习率	5e-5
num_train_epochs	训练轮数	6

6.4 实验过程

使用 12 层的 BERT 作为教师模型并进行微调，随后用该教师模型进行知识蒸馏（直接蒸馏和 CKD）。其中，前两个实验已在实验 1 中完成。

7 实验 2：结果与分析

7.1 评价指标

同 5.1，使用验证集上的 accuracy 准确率。

7.2 实验结果

在 5 个随机数种子上进行实验，得到的结果如下：

模型及方法	平均准确率	最高准确率
教师模型+微调	67.8%±1.5%	70.1%
学生模型+KD	65.4%±0.3%	65.7%
学生模型+CKD	67.3%±1.2%	69.8%

7.3 实验分析与评价

本次实验结果如 7.2 所示。

从实验结果可以看到，采语境上下文知识蒸馏得到的结果远比直接进行知识蒸馏要好，通过 CKD 得到的学生模型的性能已经接近教师模型。

装 订 线

8 总结与思考

8.1 总结

首先，本次实验复现了论文《Dynamic Knowledge Distillation for Pre-trained Language Models》中关于动态知识蒸馏的三个实验，即基于动态教师选择的知识蒸馏、基于动态数据选择的知识蒸馏和基于动态监督调整的知识蒸馏。实验结果略微和原论文所得结果有些许差异，主要原因在于我为了比较不同方法在同一实验配置下的性能而未进行数据增强。该论文有很大的创新之处。在我看来，该论文将课程学习的思想融入到知识蒸馏中，在让蒸馏所使用的教师模型、数据和损失函数随着训练过程而自适应发生变化，并且这种结合十分到位。

其次，本次实验复现了论文《Distilling Linguistic Context for Language Model Compression》中关于语境上下文知识蒸馏的实验，从不同单词表示在同一 BERT 层输出和同一单词在不同 BERT 层输出的表示上构建结构关系，并且分别构建了两种关系——二元关系和三元几何角。这使得知识蒸馏的性能有所提升。

8.2 思考与创新点

近年来关于知识蒸馏的研究更多关注到模型的“结构知识”，尤其是在 2021 年和 2022 年有关知识蒸馏的文章中，出现了大量的结构知识构建方法，包括但不局限于本次所复现的 CKD（上下文知识蒸馏）。其中，MiniLMv2 所提到的注意力矩阵的结构蒸馏还有 MGSKD 所提到的多粒度结构知识蒸馏都不断提升着蒸馏的性能。

而本次所复现的 DKD（动态知识蒸馏）仅考虑了最简单的、形式直接的蒸馏知识。

因此，我认为，目前 BERT 的知识蒸馏可以从两个角度来进行：一方面可以从 BERT 的表征出发，构建更多样化的结构知识；另一方面可以从动态策略出发，寻找更适合知识蒸馏的的动态策略。此外，除了关注 NLP 领域的知识蒸馏方法之外，视觉领域的知识蒸馏方法也值得借鉴。并且，对图神经网络的知识蒸馏也值得关注。

装 订 线

9 参考文献

[1] Li L, Lin Y, Ren S, et al. Dynamic Knowledge Distillation for Pre-trained Language Models[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021: 379-389. （**本次复现**）

[2] Park G, Kim G, Yang E. Distilling Linguistic Context for Language Model Compression[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021: 364-378.（**本次复现**）

[3] Liu C, Tao C, Feng J, et al. Multi-Granularity Structural Knowledge Distillation for Language Model Compression[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. 2022: 1001-1011.

[4] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J].Advances in neural information processing systems, 2017, 30.

[5] Kenton J D M W C, Toutanova L K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]. Proceedings of NAACL-HLT. 2019: 4171-4186.

[6] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[J]. stat, 2015, 1050: 9.

[7] Sun S, Cheng Y, Gan Z, et al. Patient Knowledge Distillation for BERT Model Compression[C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 4323-4332.

[8] Jiao X, Yin Y, Shang L, et al. TinyBERT: Distilling BERT for Natural Language Understanding[C]. Findings of the Association for Computational Linguistics: EMNLP 2020. 2020: 4163-4174.

[9] Sanh V, Debut L, Chaumond J, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter[J]. In Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing, 2019.

[10] Sun Z, Yu H, Song X, et al. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices[C]. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 2158-2170.

[11] Wang A, Singh A, Michael J, et al. Glue: A multi-task benchmark and analysis platform for natural language understanding[C]. 7th International Conference on Learning Representations, ICLR 2019. 2019.