

TinyBERT: 为自然语言理解提炼 BERT

焦晓琪^{1*}, Yichun Yin^{2*}, Lifeng Shang^{2#}, Xin Jiang², Xiao

Chen², Linlin Li³, Fang Wang^{1#} and Qun Liu²

¹华中科技大学信息存储系统重点实验室、武汉光电国家实验室² 华为诺亚

方舟实验室

³华为技术有限公司

{*jiaoxiaoqi, wangfang*}@hust.edu.cn

{*yinyichun, shang.lifeng, jiang.xin*}@huawei.com

{*chen.xiao2, lynn.lililn, qun.liu*}@huawei.com

摘要

语言模型预训练 (如 BERT) 大大提高了许多自然语言处理任务的性能。然而, 预训练语言模型的计算成本通常很高, 因此很难在资源有限的设备上高效执行。为了在保持准确性的同时加快推理速度并缩小模型尺寸, 我们首先提出了一种新颖的转换器蒸馏方法, 该方法专门用于基于转换器的模型的知识蒸馏 (KD)。利用这种新的知识蒸馏方法, 大型 "教师" BERT 中编码的大量知识可以有效地转移到小型 "学生" TinyBERT 中。然后, 我们为 TinyBERT 引入了一个新的两阶段学习框架, 该框架在预培训和特定任务学习阶段都采用了 Transformer 提炼法。该框架确保 TinyBERT 可以捕捉到 BERT 中的通用领域知识和特定任务知识。

TinyBERT₄¹ 在 GLUE 基准测试中, 4 层的 TinyBERT 比其老师 BERT_{BASE} 的性能高出 96.8%, 同时体积小了 7.5 倍, 推理速度快了 9.4 倍。TinyBERT₄ 在 BERT 蒸馏方面也明显优于 4 层的最先进基础线路, 参数和推理时间分别仅为它们的 28% 和 31%。此外, 具有 6 层的 TinyBERT₆ 与其教师 BERT_{BASE} 性能相当。

预训练语言模型, 然后在下游任务中进行微调, 这已成为

*作者的贡献相同。

[†]这项工作是在焦晓琪在 华为诺亚方舟实验室实习时完成的。

[#]通讯作者:

¹代码和模型可在 <https> 上公开获取:

[//github.com/huawei-nah/Pretrained-Language-Model/tree/master/TinyBERT](https://github.com/huawei-nah/Pretrained-Language-Model/tree/master/TinyBERT)

自然语言处理（NLP）。预训练的语言模型（PLMs），如 BERT（Devlin 等人，2019 年）、XLNet（Yang 等人，2019 年）、RoBERTa（Liu 等人，2019 年）、ALBERT（Lan 等人，2020 年）、T5（Raf-Rein 等人，2019 年）、ALBERT（Lan 等人，2020 年）等。

fel 等人，2019）和 ELECTRA（Clark 等人，2020），在许多 NLP 任务（例如 GLUE 基准（Wang 等人，2018）和具有挑战性的多跳推理任务（Ding 等人，2019））中取得了巨大成功。然而，PLM 通常参数较多，推理时间较长，难以在手机等边缘设备上部署。最近的研究（Kovaleva 等人，2019 年；Michel 等人，2019 年；Voita 等人，2019 年）表明，PLMs 中存在冗余。因此，在保持 PLM 性能的同时，减少 PLM 的计算开销和模型存储时间既重要又可行。

已经有很多模型压缩技术（Han 等人，2016 年）被提出来，以加速深度模型推理，并在保持准确性的同时减小模型大小。最常用的技术包括量化（Gong 等人，2014 年）、权重剪枝（Han 等人，2015 年）和知识提炼（KD）（Romero 等人，2014 年）。在本文中，我们将重点讨论知识蒸馏，这是源自 Hinton 等人（2015 年）在师生框架下提出的一个想法。知识蒸馏的目的是将大型教师网络中蕴含的知识转移到小型学生网络中，并对学生网络进行训练，使其重现教师网络的行为。基于该框架，我们提出了一种新颖的蒸馏方法，专门用于基于 Transformer 的模型（Vaswani 等人，2017 年），并以 BERT 为例研究了该方法在大规模 PLM 中的应用。

KD 已在 NLP（Kim 和 Rush，2016 年；Hu 等人，2018 年）以及预训练语言模型（Sanh 等人，2019 年；Sun 等人，2019 年，2020 年；Wang 等人，2020 年）中得到广泛研究。预训练-再微调范式首先是预训练，然后是微调。

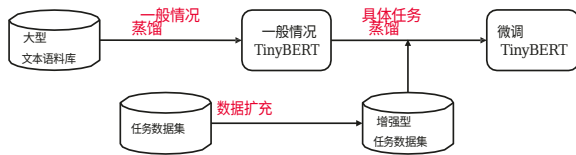


图 1: TinyBERT 学习示意图。

在大规模无监督文本语料库上训练 BERT，然后在特定任务数据集上对其进行微调，这大大增加了 BERT 识别的难度。因此，需要为这两个训练阶段设计有效的 KD 策略。

为了构建有竞争力的 TinyBERT，我们首先提出了一种新的 *Transformer 提炼* 方法，以提炼教师 BERT 中蕴含的知识。具体来说，我们设计了三种损失函数来拟合 BERT 层的不同表征：1) 嵌入层的输出；2) 转换器层得出的隐藏状态和注意力矩阵；3) 预测层输出的对数。最近的研究发现 (Clark 等人, 2019)，基于注意力的拟合可以使 BERT 学习到的注意力权重包含大量的语言知识，从而鼓励语言知识可以很好地从教师 BERT 转移到学生 TinyBERT。然后，我们提出了一个新颖的 *两阶段学习框架*，包括 *一般蒸馏* 和 *特定任务蒸馏*，如图 1 所示。在一般蒸馏阶段，没有微调的原始 BERT 充当教师模型。学生 TinyBERT 通过在通用语料库上的转换器蒸馏模仿教师的行为。之后，我们会得到一个通用的 TinyBERT，将其作为学生模型的初始化，用于进一步的提炼。在特定任务的提炼阶段，我们首先进行数据扩增，然后使用微调后的 BERT 作为教师模型，在扩增后的数据集上进行提炼。需要指出的是，这两个阶段对于提高 TinyBERT 的性能和泛化能力至关重要。

这项工作的主要贡献如下：1) 我们提出了一种新的 Transformer distillation 方法，以确

TinyBERT 可以吸收教师 BERT 的通用领域知识和特定任务知识。3) 我们在实验中表明，我们的 TinyBERT₄ 在 GLUE 任务上可以达到教师 BERT_{BASE} 96.8% 以上的性能，同时拥有保编码在教师 BERT 中的语言知识能够充分地转移到 TinyBERT 中；2) 我们提出了一种新颖的两阶段学习框架，在预训练和微调阶段执行所提出的 Transformer distillation 方法，以确保在 TinyBERT 中的语言知识能够充分地转移到 TinyBERT 中。

4) 我们还表明，6 层 TinyBERT₆ 与教师 BERT_{BASE} 在 GLUE 上的表现相当。

多头注意力的定义是将来自不同表征的注意力头连接起来

2 序言

在本节中，我们将介绍 Transformer (Vaswani 等人, 2017 年) 和 Knowledge Distillation (Hinton 等人, 2015 年) 的表述。我们提出的 *Transformer Distillation* 是一种专门为基于 Transformer 的模型设计的 KD 方法。

2.1 变压器层

最近的大多数预训练语言模型 (如 BERT、XLNet 和 RoBERTa) 都采用了 Transformer 层，通过自我注意机制捕捉输入标记之间的长期依赖关系。具体来说，标准的 Transformer 层包括两个主要的子层：多头注意 (MHA) 和全连接前馈网络 (FFN)。

多头注意力 (MHA)。注意力函数的计算取决于查询、键和值三个部分，分别用矩阵 \mathbf{Q} 、 \mathbf{K} 和 \mathbf{V} 表示。注意力函数可表述如下：

$$\mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{A})\mathbf{V}, \quad (2)$$

其中， d_k 是键的维度，用作缩放因子； \mathbf{A} 是通过点积运算从 \mathbf{Q} 和 \mathbf{K} 的兼容性中计算出的注意力矩阵。最终的函数输出计算为值 \mathbf{V} 的加权和，而权重是通过对矩阵 \mathbf{A} 的每一列进行 $\text{softmax}()$ 运算计算得出的。根据 Clark 等人 (2019) 的研究，BERT 中的注意力矩阵可以捕捉到大量语言知识，因此在我们提出的提炼方法中发挥着至关重要的作用。

子空间如下

$$\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{h}_1, \dots, \mathbf{h}_k) \mathbf{W}, \quad (3)$$

其中, k 是注意力头的数量, \mathbf{h}_i 表示第 i 个注意力头, 由 $\text{Attention}()$ 函数根据来自不同表示子空间的输入计算得出。矩阵 \mathbf{W} 起着线性变换的作用。

位置前馈网络 (FFN)。变压器层也包含一个全连接前馈网络, 其结构如下:

$$\text{FFN}(x) = \max(0, \mathbf{xW}_1 + b_1) \mathbf{W}_2 + b_2. \quad (4)$$

我们可以看到, FFN 包含两个线性转换和一个 ReLU 激活。

2.2 知识提炼

KD 的目的是将大型教师网络 T 的知识转移到小型学生网络 S 上。让 f^T 和 f^S 分别代表教师网络和学生网络的行为函数。行为函数的目标是将网络输入转换为某种信息表征, 它可以定义为网络中任意一层的输出。在 Transformer distillation 中, MHA 层或 FFN 层的输出, 或一些中间表征 (如注意力矩阵 \mathbf{A}) 都可以用作行为函数。从形式上看, KD 可以建模为最小化以下目标函数:

$$L_{\text{KD}} = L \sum_{x \in X} f^S(x), f^T(x), \quad (5)$$

其中, $L()$ 是一个损失函数, 用于评估教师和学生网络之间的差异; x 是文本输入, 表示训练数据集。因此, 如何优化有效的行为函数和损失函数就成了研究的关键问题。与以往的 KD 方法不同, 除了特定任务的训练阶段, 我们还需要考虑如何在 BERT 的预训练阶段执行 KD。

3 方法

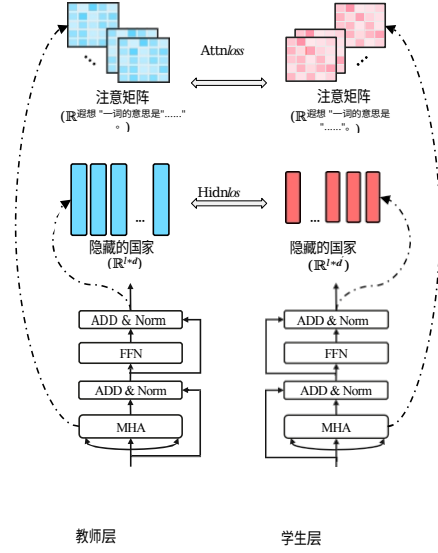


图 2: 由 $\text{Attn}_{\text{loss}}$ (基于注意力的蒸馏) 和 $\text{Hidn}_{\text{loss}}$ (基于隐藏状态的蒸馏) 组成的变压器层蒸馏细节。

3.1 变压器蒸馏

所提出的 **变压器蒸馏法** 是一种专门为变压器网络设计的 KD 方法。在这项工作中, 学生网络和教师网络都是用 Transformer 层构建的。为了清楚地说明问题, 我们在介绍我们的方法之前先提出问题。

问题的提出。 假设学生模型有 M 个变换层, 教师模型有 N 个变换层, 我们首先从教师模型的 N 个变换层中选择 M 个进行 **变换层蒸馏**。然后定义一个函数 $n = g(m)$, 作为从学生层到教师层的指数之间的映射函数, 即学生模型的 m 层从教师模型的第 $g(m)$ 层学习信息。准确地说, 我们设定 0 为嵌入层的索引, $M+1$ 为预测层的索引, 相应的层映射分别定义为 $0 = g(0)$ 和 $N+1 = g(M+1)$ 。实验部分将研究选择不同映射函数对性能的影响。从形式上看, 学生可以通过最小化以下目标从教师那里获取知识:

$$L_{\text{模型}} = \sum_{m=0}^{M+1} \sum_{x \in X} \lambda L_{\text{m layer}}(f_m^S(x), f_{g(m)}^T(x)), \quad (6)$$

在本节中，我们将为基于变压器的模型提出一种新颖的提炼方法，并为我们从 BERT 中提炼出的模型提出一个 *两阶段学习* 框架，即 TinyBERT。

其中 layer 指某一模型层（如变换层或嵌入层）的损失函数， $f_m(x)$ 表示第 m 层诱导的行为函数， λ_m 是超

参数，表示

第 m 层的蒸馏。

变压器层蒸馏。拟议的转换器层蒸馏包括*基于注意力的蒸馏*和*基于隐藏状态的蒸馏*，如图 2 所示。基于注意力的蒸馏是由于最近的研究发现，通过 BERT 学习到的注意力权重可以捕捉到丰富的语言知识 (Clark et al.) 这种语言知识包括语法和核心参照信息，对于自然语言理解至关重要。因此，我们提出了基于注意力的提炼方法，以确保语言知识能够从教师 (BERT) 传递给学生 (TinyBERT)。具体来说，学生要学会适应教师网络中的多头注意力矩阵，其目标定义为：

$$L = \frac{1}{\sum_h} \text{MSE}(\mathbf{A}_i^S, \mathbf{A}_i^T), \quad (7)$$

联系人 h $i=1$ i i

其中， h 是注意力头数， $\mathbf{A}_i \in \mathbb{R}^{l \times l}$ 指与注意力矩阵相对应的为第 i 位教师或学生的班主任， l 为输入值。文本长度， $\text{MSE}(\cdot)$ 表示文本长度的均方差误差损失函数。在这项工作中，（非正常-

i 作为拟合目标，而不是其 softmax 输出 $\text{softmax}(\mathbf{A}_i)$ ，因为我们的实验表明，前者的收敛速度更快，性能更好。

除了基于注意力的提炼，我们还从变压器层的输出中提炼知识，目标如下：

$$l_{\text{hidn}} = \text{mse}(\mathbf{h}^S \mathbf{w}_h, \mathbf{h}^T), \quad (8)$$

其中矩阵 $\mathbf{H}^S \in \mathbb{R}^{l \times d}$ 和 $\mathbf{H}^T \in \mathbb{R}^{l \times d}$ 分别指学生网络和教师网络的隐藏状态，由公式 4 计算得出。标量值 d 和 d^r 表示教师模型和学生模型的隐藏大小， d^r 通常小于 d ，以获得较小的学生网络。矩阵 $\mathbf{w}_h \in \mathbb{R}^{d \times d^r}$ 是一种可学习的线性转换，它将学生网络的隐藏状态转换到与教师网络

其中矩阵 \mathbf{E}^S 和 \mathbf{H}^T 分别指学生网络和教师网络的隐状态矩阵。在本文中，它们的形状与隐藏状态矩阵相同。矩阵 \mathbf{w}_e 是一种线性变换，其作用与 \mathbf{w}_h 类似。

预测层蒸馏。除了改进中间层的行为外，我们还利用知识蒸馏来拟合教师模型的预测，如 Hinton 等人 (2015) 所做的那样。具体来说，我们对学生网络对数与教师对数之间的软交叉熵损失进行惩罚：

$$L_{\text{pred}} = \text{CE}(\mathbf{z}^T/t, \mathbf{z}^S/t), \quad (10)$$

其中， \mathbf{z}^S 和 \mathbf{z}^T 分别是学生和教师预测的对数向量，CE 表示交叉熵损失， t 表示温度值。在实验中，我们发现 $t = 1$ 表现出色。

利用上述蒸馏目标（即等式 7、8、9 和 10），我们可以将蒸馏和蒸馏过程统一起来。教师和学生网络之间相应层的连接损耗：

$$l_m = \begin{cases} \text{embd}, & m = 0 \\ l_{\text{hidn}} + l_{\text{attn}}, & M \geq m > 0 \end{cases} \quad (11)$$

状态相同的空间。

嵌入层蒸馏。与基于隐藏状态的蒸馏类似，我们也进行嵌入层蒸馏，其目标是：

$$L_{\text{pred}}, \quad m = M + 1$$

3.2 TinyBERT Learning

BERT 的应用通常包括两个学习阶段：预训练和微调。BERT 在预训练阶段学到的大量知识非常重要，应将其转移到压缩模型中。因此，我们提出了一种新颖的两阶段学习框架，包括 *一般蒸馏* 和 *特定任务蒸馏*，如图 1 所示。一般蒸馏帮助 TinyBERT 学习预训练

BERT 中蕴含的丰富知识，这对提高 TinyBERT 的泛化能力起着重要作用。特定任务蒸馏则进一步向 TinyBERT 传授来自微调 BERT 的知识。通过两步蒸馏，我们可以大大缩小教师模型和学生模型之间的差距。

一般蒸馏。我们使用未经微调的原始 BERT 作为教师，并使用大规模文本语料库作为训练数据。通过对²对文本进行一般

$$L_{\text{embd}} = \text{MSE}(\mathbf{E}^T \mathbf{W}_e, \quad), \quad (9)$$

²在一般的蒸馏过程中，我们不执行等式 10 中的预测层蒸馏。我们的动机是使

算法 1 特定任务蒸馏的数据扩充程序

输入: X 是一串单词
参数: p_t : 阈值概率
 N_a : 每个示例增强的样本数
 K : 候选集的大小

输出: D' : 增强数据 1: $n \leftarrow 0$; $D' \leftarrow []$
2: **while** $n < N_a$ **do**
3: $x_m \leftarrow X$
4: **for** $i \leftarrow 1$ to $\text{len}(X)$ **do**
5: **如果** $X[i]$ 是单字, **那么**
6: 用 $[MASK]$ 替换 $x_m[i]$
7: $C \leftarrow \text{BERT}(x_m)[i]$ 的 K 个最可能词 8:
 否则
9: $C \leftarrow K$ 来自 GloVe 的与 $X[i]$ 最相似的词
10: **end if**
11: 取样 $p \sim \text{Uniform}(0, 1)$
12: **if** $p \leq p_t$ **then**
13: 将 $x_m[i]$ 随机 替换为 C 中的一个单词
14: **end if**
15: **结束**
16: 将 x_m 追加到 D'
17: $n \leftarrow n + 1$
18: **同时结束**
19: **返回** D'

词嵌入结合起来, 进行词级分析。

TinyBERT 主要是在预训练阶段学习 BERT 的中间结构。通过初步实验, 我们还发现, 在预训练阶段进行预测层分辨并不会带来额外的改进。

这样, 我们就得到了一种通用的 TinyBERT, 它可以针对下游任务进行微调。然而, 由于隐藏/嵌入大小和层数的大幅减少, 一般 TinyBERT 的性能普遍不如 BERT。

特定任务蒸馏。以往的研究表明, 复杂的模型, 如微调 BERT, 会因特定领域任务的过度参数化而受到影响 (Kovaleva 等人, 2019 年)。因此, 较小的模型也有可能达到与 BERT 相同的性能。为此, 我们建议通过特定任务蒸馏来产生有竞争力的微调 Tiny-BERT。在特定任务蒸馏过程中, 我们在增强的特定任务数据集上重新执行原定的变换器蒸馏。具体来说, 我们使用微调后的 BERT 作为教师, 并提出了一种数据增强方法来扩展特定任务训练集。使用更多与任务相关的试题进行训练, 可以进一步提高学生模型的泛化能力。

数据扩充。我们将预先训练好的语言模型 BERT 和 GloVe (Pennington 等人, 2014 年)

替换来进行数据扩增。具体来说，我们使用语言模型预测单片词的替换词（Wu 等人，2019 年），并使用词嵌入检索最相似的词作为多片词的替换词。³我们定义了一些超参数来控制句子的替换率和增强数据集的数量。数据扩增程序的更多细节见算法 1。在所有实验中，我们设置 $p_t = 0.4$ ， $N_a = 20$ ， $K = 15$ 。

上述两个学习阶段互为补充：一般蒸馏为特定任务蒸馏提供了良好的初始化，而在增强数据上的特定任务蒸馏则通过集中学习特定任务知识进一步改进了 TinyBERT。尽管 TinyBERT 的模型规模大幅缩小，但通过数据扩充以及在预训练和微调阶段执行所提出的 Transformer 提炼方法，TinyBERT 可以在各种 NLP 任务中取得有竞争力的表现。

4 实验

在本节中，我们将评估 TinyBERT 在各种任务中使用不同模型设置的效果和效率。

在下游任务中，当变压器层蒸馏时

(Attn and Hidn distillation) 和 Embedding-layer distillation 记化器。

4.1 数据集

我们在通用语言理解评估（GLUE）（Wang 等人，2018 年）基准上对 TinyBERT 进行了评估，该基准由 2 个单句任务组成：CoLA（Warstadt 等人，2019 年）、SST-2（Socher 等人，2013 年），3 个句子相似性任务：MRPC（Dolan 和 Brockett，2005 年）、STS-B（Cer 等人，2017 年）、QQP（Chen 等人，2018 年），以及 4 项自然语言推理任务：MNLI（Williams 等人，2018 年）、QNLI（Rajpurkar 等人，2016 年）、RTE（Ben-tivogli 等人，2009 年）和 WNLI（Levesque 等人，2012 年）。这些任务的指标可参见 GLUE 论文（Wang 等人，2018 年）。

4.2 TinyBERT 设置

我们实例化了一个微小学生模型（层数 $M=4$ ，隐藏大小 $d^r=312$ ，前馈/滤波器大小 $d^r=1200$ ，头数 $h=12$ ），共有 14.5M 个参数。该模型被称为 TinyBERT₄。最初的

³ 通过单词标记为多个单词片段 BERT 的标

系统	#Params	#FLOPs	提速	MNLI- (m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	平均值
BERT _{BASE} (教师)	109M	22.5B	1.0x	83.9/83.4	71.1	90.9	93.4	52.8	85.2	87.5	67.0	79.5
BERT _{TINY}	14.5M	1.2B	9.4x	75.4/74.9	66.5	84.8	87.6	19.5	77.1	83.2	62.6	70.2
BERT _{SMALL}	29.2M	3.4B	5.7x	77.6/77.0	68.1	86.4	89.7	27.8	77.0	83.4	61.8	72.1
BERT ₄ -PKD	52.2M	7.6B	3.0x	79.9/79.3	70.2	85.1	89.4	24.8	79.8	82.6	62.3	72.6
蒸馏器 ₄	52.2M	7.6B	3.0x	78.9/78.0	68.5	85.2	91.4	32.8	76.1	82.4	54.1	71.9
MobileBERT _{TINY} [†]	15.1M	3.1B	-	81.5/81.6	68.9	89.5	91.7	46.7	80.1	87.9	65.1	77.0
TinyBERT ₄ (我们的)	14.5M	1.2B	9.4x	82.5/81.8	71.3	87.7	92.6	44.1	80.4	86.4	66.6	77.0
)												
BERT ₆ -PKD	67.0M	11.3B	2.0x	81.5/81.0	70.7	89.0	92.0	-	-	85.0	65.5	-
PD	67.0M	11.3B	2.0x	82.8/82.2	70.4	88.9	91.8	-	-	86.8	65.3	-
蒸馏器 ₆	67.0M	11.3B	2.0x	82.6/81.3	70.1	88.9	92.5	49.0	81.3	86.9	58.4	76.8
TinyBERT ₆ (我们的)	67.0M	11.3B	2.0x	84.6/83.2	71.6	90.4	93.1	51.1	83.7	87.3	70.0	79.4
)												

表 1: 在 GLUE 官方基准测试集上评估的结果。每组学生模型的最佳结果以黑体表示。TinyBERT₄ 和 BERT_{TINY} 的架构是 ($M=4, d=312, d_i=1200$) , BERT_{SMALL} 的架构是 ($M=4, d=512, d_i=2048$) , BERT₄-PKD 和 DistilBERT₄ 的架构是 ($M=4, d=768, d_i=3072$) ,[†] 而 BERT₆-PKD、DistilBERT₆ 和 TinyBERT₆ 的架构为 ($M=6, d=768, d_i=3072$) 。所有模型都是以单任务方式学习的。推理速度是在单个英伟达 K80 GPU 上评估的。表示 MobileBERT_{TINY} 和 TinyBERT₄ 之间的比较可能不公平, 因为前者有 24 层, 是从 IB-BERT_{LARGE} 中提炼出来的任务无关模型, 而后者是从 BERT_{BASE} 中提炼出来的任务特定的 4 层模型。

BERT_{BASE} ($N=12, d=768, d_i=3072$ and $h=12$) 作为教师模型, 包含 109M 个参数。我们使用 $g(m) = 3 \times m$ 作为层映射函数, 因此 TinyBERT₄ 从 BERT_{BASE} 的每 3 层学习。此外, 为了与基线进行直接比较, 我们还实例化了一个

TinyBERT₆ ($M=6, d^r=768, d^r=3072, h=12$) , 其架构与 BERT₆-PKD (Sun 等人, 2019) 和 DistilBERT₆ (Sanh 等人, 2019) 相同。

TinyBERT 学习包括一般蒸馏和特定任务蒸馏。对于一般蒸馏, 我们将最大序列长度设为 128, 并使用英语维基百科 (2500 万字) 作为文本语料库, 在预先训练的 BERT_{BASE} 的监督下进行 3 个 epoch 的中间层蒸馏, 其他超参数与 BERT 预训练相同 (Devlin 等, 2019) 。对于特定任务的蒸馏, 在微调 BERT 的监督下, 我们首先对增强数据进行 20 个历元的中间层蒸馏。⁴然后在增强数据上进行预测层蒸馏,

每次蒸馏 3 个 epoch, 并选择学习率为 $5e-5$ 。⁵在 dev 集上选择批量大小为 16、32, 学习率为 $1e-5$ 、 $2e-5$ 、 $3e-5$, 进行 3 次历时计算。在特定任务蒸馏时, 单句任务的最大序列长度设为 64 , 序列对任务的最大序列长度设为 128。

⁴对于大型数据集 MNLI、QQP 和 QNLI, 我们只进行了 10 次中间层蒸馏, 而对于具有挑战性的任务 CoLA, 我们在这一步进行了 50 次蒸馏。⁵对于回归任务 STS-B, 原始训练集的效果更好。

{ } { }

4.3 基线

我们将 TinyBERT 与 BERT_{TINY}、BERT_{SMALL}⁶(Turc 等人, 2019 年) 和几种最先进的 KD 基线, 包括 BERT- PKD (Sun 等人, 2019 年)、PD (Turc 等人, 2019 年)、DistilBERT (Sanh 等人, 2019 年) 和 Mobile- BERT (Sun 等人, 2020 年)。BERT_{TINY} 是指直接预训练一个小型 BERT, 其模型架构与 TinyBERT₄ 相同。在训练 BERT_{TINY} 时, 我们采用了与原始 BERT (Devlin 等人, 2019 年) 相同的学习策略。为了进行公平比较, 我们使用发布的代码训练了 4 层 BERT -PKD₄⁷和 4 层 DistilBERT₄⁸并使用建议的超参数对这些 4 层基线进行微调。对于 6 层基线, 我们使用报告中的数字, 或使用已发布的模型在 GLUE 测试集上评估结果。

4.4 GLUE 的实验结果

我们将模型预测结果提交给官方的 GLUE 评估服务器, 以获得测试集的结果。⁹如表 1 所示。

4 层学生模型的实验结果表明 1) 由于模型规模的大幅缩小, BERT_{TINY} (或 BERT_{SMALL}) 与 BERT_{BASE} 的性能差距很大。2) TinyBERT₄ 在所有 GLUE 任务上的表现始终优于 BERT_{TINY}, 而 BERT 在所有 GLUE 任务上的表现始终劣于 TinyBERT。

⁶<https://github.com/google-research/bert>
<https://github.com/intersun/>
PKD-for-BERT 模型压缩

⁸<https://github.com/huggingface/transformers/tree/main/examples/distillation>

⁹<https://gluebenchmark.com>

平均提高了 6.8%。这表明，所提出的 KD 学习框架可以有效提高小型模型在各种下游任务中的性能。3) TinyBERT₄ 明显优于最先进的 4 层 KD 基线（即 BERT₄-PKD 和 DistilBERT₄），幅度至少为 4.4%，参数提升 28%，速度提升 3.1 倍。4) 与教师 BERT_{BASE} 相比，TinyBERT₄ 在模型效率方面缩小了 7.5 倍，速度提高了 9.4 倍，同时保持了具有竞争力的性能。5) 对于具有挑战性的 CoLA 数据集（预测语言可接受性判断的任务），所有 4 层分解模型与教师模型相比都有很大的性能差距，而 TinyBERT₄ 与 4 层基础模型相比有显著提高。6) 我们还将 TinyBERT 与从 24 层 IB-BERT_{LARGE} 中提炼出来的 24 层 MobileBERT_{TINY} 进行了比较。结果显示，TinyBERT₄ 的平均得分与 24 层模型相同，FLOPs 仅为 38.7%。7) 当我们将模型的容量提高到 TinyBERT₆ 时，其性能可以进一步提升，平均以 2.6% 的优势超过了相同架构的基线，并取得了与教师相当的结果。8) 与其他两阶段基线 PD（首先预训练一个小型 BERT，然后用这个小型模型对特定任务进行蒸馏）相比，TinyBERT 在特定任务阶段通过一般蒸馏对学生进行初始化。我们将在附录 C 中分析这两种初始化方法。

此外，BERT-PKD 和 DistilBERT 用预先训练好的 BERT 的某些层初始化学生模型，这使得学生模型必须保持与教师模型相同的变换层（或嵌入层）大小设置。在我们的两阶段蒸馏框架中，TinyBERT 是通过一般蒸馏来初始化的，这使得它在选择模型配置时更加灵活。

更多比较。我们通过纳入更多基准线来证明 TinyBERT 的有效性，例如穷人的 BERT（

Sajjad 等人，2020 年）、BERT-of-Theseus（Xu 等人，2020 年）和 MiniLM（Wang 等人，2020 年），其中一些基准线只报告了 GLUE dev 集上的结果。此外，我们还在 SQuAD v1.1 和 v2.0 上对 TinyBERT 进行了评估。由于篇幅有限，我们在附录 A 和 B 中介绍了我们的结果。

系统	MNLI-m	MNLI-mm	MRPC	CoLA	平均值
TinyBERT ₄	82.8	82.9	85.8	50.8	75.6
无 GD	82.5	82.6	84.1	40.8	72.5
无 TD	80.6	81.2	83.8	28.5	68.5
不含 DA	80.5	81.0	82.4	29.8	68.4

表 2：两阶段学习框架中不同程序（即 TD、GD 和 DA）的消融研究。这些变体在 dev 集上进行了验证。

系统	MNLI-m	MNLI-mm	MRPC	CoLA	平均值
TinyBERT ₄	82.8	82.9	85.8	50.8	75.6
无 Embd	82.3	82.3	85.0	46.7	74.1
无 Pred	80.5	81.0	84.3	48.2	73.5
不含 Trm	71.7	72.3	70.1	11.2	56.3
无附页	79.9	80.7	82.3	41.1	71.0
不含 Hidn	81.7	82.1	84.1	43.7	72.9

表 3：TinyBERT 学习中不同蒸馏目标的消融研究。这些变体是在开发集上验证的。

4.5 消融研究

在本节中，我们将进行消融研究，以了解：

a) 图 1 中建议的两阶段 TinyBERT 学习框架的不同过程；b) 公式 11 中的不同分散目标。

4.5.1 学习程序的效果

拟议的两阶段 TinyBERT 学习框架由三个关键程序组成：GD（总体蒸馏）、TD（特定任务蒸馏）和 DA（数据增强）。表 2 分析并展示了每个学习程序的性能。结果表明，所有这三个程序对所提出的方法都至关重要。TD 和 DA 在所有四项任务中都有类似的效果。我们注意到，在所有任务中，任务特定程序（TD 和 DA）比预训练程序（GD）更有帮助。另一个有趣的现象是，与 MNLI 和 MRPC 相比，GD 对 CoLA 的帮助更大。我们推测，GD 学习到的语言概括能力（Warstadt 等人，2019 年）在语言可接受性判断任务中发挥了重要作用。

4.5.2 蒸馏目标的影响

我们研究了蒸馏目标对 TinyBERT 学习的影响。我们提出了几种基本方法，包括不使用变压器层蒸馏（w/o Trm）、不使用嵌入层蒸馏（w/o Emb）或不使用"....."（...

预测层蒸馏（不含 Pred）¹⁰结果如表 3 所示。结果如表 3 所示，表明所有建议的蒸馏目标都是有用的。无 Trm¹¹大幅下降，从 75.6 降至 56.3。性能大幅下降的原因在于初始化模型。在预训练阶段，获得良好的初始化对基于变压器的模型的蒸馏至关重要，而在此阶段，没有来自上层的监督信号来更新变压器层的参数。此外，我们还研究了注意力（Attn）和隐藏状态（Hidn）在变换层蒸馏中的贡献。我们发现，基于注意力的蒸馏比基于隐藏状态的蒸馏影响更大。同时，这两种知识蒸馏技术是相辅相成的，因此在我们的实验中，它们是基于 Transformer 模型的最重要的蒸馏技术。

4.6 绘图功能的影响

我们还研究了不同映射函数 $n = g(m)$ 对 TinyBERT 学习的影响。如第 4.2 节所述，我们最初的 TinyBERT 采用统一策略，并与两种典型的基线策略进行比较，包括上层策略 $(g(m) = m + N M ; 0 < m \leq M)$ 和下层策略 $(g(m) = m ; 0 < m < M)$ 。

比较结果见表 4。我们发现，上层策略在 MNLI 上的表现优于下层策略，而在 MRPC 和 CoLA 上的表现较差，这证实了不同任务取决于不同 BERT 层知识的观点。统一策略涵盖了 BERT_{BASE} 从底层到顶层的知识，在所有任务中都比其他两个基线策略取得了更好的性能。为特定任务自适应选择层是一个棘手的问题，我们将其作为未来的工作。

5 相关工作

预训练语言模型的压缩 一般来说，预训练语言模型（PLM）可以通过低秩近似（Ma

准交叉熵。

¹¹在 "w/o Trm "设置下，我们实际上 1) 在预训练阶段进行嵌入层蒸馏；2) 在微调阶段进行嵌入层和预测层蒸馏。

¹⁰预测层蒸馏对增强数据执行软交叉熵，如公式 10 所示。"w/o Pred "表示针对原始训练集的地面真相执行标

系统	MNLI-m	MNLI-mm	MRPC	CoLA	平均值
制服	82.8	82.9	85.8	50.8	75.6
返回顶部	81.7	82.3	83.6	35.9	70.9
底部	80.6	81.3	84.6	38.5	71.3

表 4: TinyBERT₄ 不同映射策略的结果 (偏差)。

等人, 2019; Lan 等人, 2020)、权重共享 (De-ghani 等人, 2019; Lan 等人, 2020)、知识提炼 (Tang 等人, 2019; Sanh 等人, 2019; Turc 等人, 2019; Sun 等人, 2020; Liu 等人, 2020; Wang 等人, 2020)、剪枝 (Cui 等人, 2019; Mc-Carley, 2019; F. 等人, 2020; Elbayad 等人, 2020; Gordon 等人, 2020; Hou 等人, 2020) 或量化 (Shen 等人, 2019; Zafir 等人, 2019)。本文的重点是知识提炼。

PLM 的知识提炼 有一些研究试图将预训练的语言模型 (PLM) 提炼成更小的模型。

BiLSTM_{SOFT} (Tang 等人, 2019) 将 BERT 的特定任务知识提炼为单层 BiLSTM。

BERT-PKD (Sun 等人, 2019) 不仅从最后一层教师那里获取知识, 还在微调阶段从中间层外行那里获取知识。DistilBERT (Sanh 等人, 2019 年) 在大规模语料库的预训练阶段执行蒸馏。同时进行的工作有: Mobile-BERT (Sun 等人, 2020 年) 通过在预训练阶段进行渐进式知识转移, 将增强了瓶颈结构的 BERT_{LARGE} 提炼为 24 层的瘦身版本。

MiniLM (Wang 等人, 2020 年) 也是在预训练阶段进行深度自我注意力提炼。相比之下, 我们提出了一种新的两阶段学习框架, 通过一种新颖的转换器蒸馏方法, 在预训练和微调阶段从 BERT 中蒸馏知识。**预训练精简版 PLM** 其他相关工作旨在直接预训练精简版 PLM。Turc 等人 (2019) 预训练了 24 个

微型 BERT 模型, 结果表明, 预训练在较小架构的文本中仍然很重要, 对预训练的紧凑型模型进行微调也很有竞争力。AL-BERT (Lan 等人, 2020) 结合了嵌入因子化和跨层参数共享来减少模型参数。由于 ALBERT 没有减少隐藏大小或变压器块的层数, 它的计算量仍然很大。另一项同时进行的工作是 ELECTRA (Clark 等人, 2020 年), 它提出了一项名为 "替换为肯检测" 的样本效率任务, 以加速预训练, 并提出了一个 12 层的 ELECTRA_{small}, 其性能与 TinyBERT₄ 相当。不同

与这些小型 PLM 相比, TinyBERT₄ 是一个 4 层模型, 可以实现更快的速度。

6 结论和未来工作

在本文中, 我们介绍了一种基于 Transformer 的蒸馏新方法, 并进一步提出了 TinyBERT 的两阶段框架。大量实验表明, TinyBERT 在大幅减少 BERT_{BASE} 的模型大小和推理时间的同时, 实现了极具竞争力的性能, 为在边缘设备上部署基于 BERT 的 NLP 模型提供了有效途径。在未来的工作中, 我们将研究如何将更广泛、更深入的教师知识 (如 BERT_{LARGE}) 有效地转移到学生 TinyBERT 上。将蒸馏与量化/剪枝相结合将是进一步压缩预训练语言模型的另一个有前途的方向。

致谢

本研究得到了国家自然科学基金委员会 (NSFC) 61832020 号、61821003 号、61772216 号、Na-中央高校基本科研业务费重大项目, 项目编号: 2017ZX01032-101。

参考资料

L.Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo.2009.第五届帕斯卡尔识别文本蕴含挑战赛。

D.Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L.Specia.2017.Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation.第11届国家间语义评估研讨会论文集》。

Z.Chen, H. Zhang, X. Zhang, and L. Zhao.2018.Quora 问题对。

K.Clark, U. Khandelwal, O. Levy, and C. D. Manning.2019.伯特的注意力分析》(What does Bert look at?In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neu-*

ral Networks for NLP.

K.Clark, M. Luong, Q. V. Le, and C. D. Manning.2020.Electra: 将文本编码器作为判别器而非生成器进行预训练。In *ICLR*.

B.Cui, Y. Li, M. Chen, and Z. Zhang.2019.用稀疏自注意力机制微调 Bert。在 *EMNLP* 中。

M.Dezhghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L.Kaiser.2019.通用变压器。In *ICLR*.

- J.Devlin、M. Chang、K. Lee 和 K. Toutanova。2019.Bert：用于语言理解的深度双向变换器的预训练。在 *NAACL*。
- M.Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang.2019.大规模多跳阅读理解的认知图谱。In *ACL*。
- W.B. Dolan and C. Brockett.2005.自动构建语句转述语料库。In *Proceedings of the Third International Workshop on Paraphrasing*。
- M.Elbayad, J. Gu, E. Grave, and M. Auli.2020.深度自适应变压器。In *ICLR*。
- Angela F., Edouard G., and Armand J. 2020.利用结构化滤波降低变压器深度。In *ICLR*。
- Y.Gong, L. Liu, M. Yang, and L. Bourdev.2014.使用向量量化压缩深度卷积网络。 *arXiv preprint arXiv:1412.6115*。
- M.A. Gordon, K. Duh, and N. Andrews.2020.Compressing bert: Studying the effects of weight pruning on transfer learning. *ArXiv preprint arXiv:2002.08307*。
- S.Han, Mao H., and Dally W. J. 2016.深度压缩：用剪枝、训练量化和胡夫曼编码压缩深度神经网络。In *ICLR*。
- S Han、J. Pool、J. Tran 和 W. Dally。2015.同时学习权重和连接，实现高效神经网络。In *NIPS*。
- G. Hinton, O. Vinyals, and J. Dean.2015.提炼神经网络中的知识。 *arXiv preprint arXiv:1503.02531*。
- L.Hou、L. Shang、X. Jiang 和 Q. Liu.2020.Dynabert: Dynamic bert with adaptive width and depth. *arXiv preprint arXiv:2004.04037*。
- M.Hu, Y. Peng, F. Wei, Z. Huang, D. Li, N. Yang, and M. Zhou.2018.用于机器阅读理解的注意力引导答案分解。In *EMNLP*。
- Y.Kim and A. M. Rush.2016.序列级知识提炼。在 *EMNLP* 中。
- O. Kovaleva, A. Romanov, A. Romanov, A. Rogers、和 A.Rumshisky.2019.揭开贝尔特的秘密。In *EMNLP*。
- Z.Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut.2020.Albert: A lite bert for self supervised learning of language representations.In *ICLR*。
- Hector Levesque、Ernest Davis 和 Leora Morgenstern。2012.Winograd 模式挑战。 *第十三届知识表示与推理原理国际会议*。

- W.Liu, P. Zhou, Z. Zhao, Z. Wang, H. Deng, and Q. Ju.2020.Fastbert: a self-distilling bert with adaptive inference time. *ArXiv preprint arXiv:2004.02178*.
- Y.Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O.Levy、 M. Lewis、 L. Zettlemoyer 和 V. Stoyanov 。 2019.Roberta : A robustly optimized bert pretrain- ing approach. *ArXiv preprint arXiv:1907.11692*.
- X.Ma, P. Zhang, S. Zhang, N. Duan, Y. Hou, M. Zhou, and D. Song.2019.用于语言建模的张量变换器。 In *NIPS*.
- J.S. McCarley.2019.Pruning a bert-based question answering model. *ArXiv preprint arXiv:1910.06360*.
- P.Michel, O. Levy, and G. Neubig.2019.十六个脑袋真的比一个脑袋好吗? In *NIPS*.
- J.Pennington, R. Socher, and C. D. Manning.2014.手套：用于单词表示的全局向量。在 *EMNLP* 中。
- C.拉斐尔、N. 沙泽尔、A. 罗伯茨、K. 李、S. 纳兰、 M.Matena、Y. Zhou、W. Li 和 P. J. Liu.2019.用统一的文本到文本转换器探索迁移学习的极限。 *arXiv preprint arXiv:1910.10683*.
- P.Rajpurkar, R. Jia, and P. Liang.2018.Know what you don't know: 无法回答的小队问题。 In *ACL*.
- P.Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang.2016.Squad : 用于文本机器分析的 100,000+ 个问题。在 *EMNLP* 中。
- A.罗梅罗、N. 巴拉斯、S. E. 卡胡、A. 查桑、 C.Gatta 和 Y. Bengio 。 2014.Fitnets : *arXiv preprint arXiv:1412.6550*.
- H.Sajjad, F. Dalvi, N. Durrani, and P. Nakov.2020.Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*.
- V.Sanh、 L. Debut、 J. Chaumond 和 T. Wolf 。 2019.Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- S.Shen, Z. Dong, J. Ye, L. Ma, Z. Yao, A. Gholami、 M.W. Mahoney, and K. Keutzer.2019.Q-bert: Hessian based ultra low precision quantization of bert. *arXiv preprint arXiv:1909.05840*.
- R.Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts.2013.通过语义树库实现语义合成的递归深度模型。 In *EMNLP*.
- S.Sun, Y. Cheng, Z. Gan, and J. Liu.2019.用于贝尔特模型压缩的患者知识提炼。在 *EMNLP*。
- Z.Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D.Zhou.2020.Mobilebert: a compact task- agnostic bert for resource-limited devices. *ArXiv preprint arXiv:2004.02984*.

- R.Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin.2019.从伯特利到简单神经网络的任务特定知识提炼 (*arXiv preprint arXiv:1903.12136*).
- I.Turc, M. Chang, K. Lee, and K. Toutanova.2019.Well-read students learn better : 学生初始化对知识提炼的影响. *arXiv preprint arXiv:1908.08962*.
- A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. 琼斯、A. N. 戈麦斯、L.Kaiser, and I. Polosukhin.2017.注意力就是你所需要的一切。In *NIPS*.
- E.Voita, D. Talbot, F. Moiseev, R. Sennrich, and I.Titov.2019.分析多头自我注意专门的头做繁重的工作, 其余的可以修剪。In *ACL*.
- A.Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S.Bowman.2018.Glue: 用于自然语言理解的多任务基准和分析平台。In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.
- W.Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M.Zhou.2020.Minilm: 用于预训练变换器的任务识别压缩的深度自我注意分馏。*arXiv 预印本 arXiv:2002.10957*.
- A.Warstadt, A. Singh, and S. R. Bowman.2019.神经网络可接受性判断。 *TACL*.
- A.Williams, N. Nangia, and S. Bowman.2018.通过推理进行句子理解的广覆盖挑战语料库。In *NAACL*.
- X.Wu, S. Lv, L. Zang, J. Han, and S. Hu.2019.Conditional bert contextual augmentation. *国际计算科学大会*.
- C.Xu, W. Zhou, T. Ge, F. Wei, and M. Zhou.2020.Bert-of-theseus : *ArXiv preprint arXiv:2002.02925*.
- Z.Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le.2019.Xlnet: 用于语言理解的广义回归预训练。In *NIPS*.
- O.Zafir, G. Boudoukh, P. Izsak, and M. Wasserblat.2019.Q8bert : *ArXiv preprint arXiv:1910.06188*.

A 关于 GLUE 的更多比较

由于之前一些有关 BERT 压缩的研究仅在 GLUE dev set 上评估了其模型, 为了方便直接地进行比较, 我们在此将 TinyBERT₆ 与之前这些研究的报告结果进行比较。所有比较的方法都具有

系统	CoLA (8.5k) Mcc	MNLI-m (393k) Acc	MNLI-mm (393k) Acc	MRPC (3.7k) F1/Acc	QNLI (105k) Acc	QQP (364k) F1/Acc	RTE (2.5k) Acc	SST-2 (67k) Acc	STS-B (5.7k) 梨/豆
<i>相同的字生建筑 ($M=6$; $d^r=768$; $d^r=3072$)</i>									
蒸馏器 ₆	51.3	82.2	-	87.5/-	89.2	-/88.5	59.9	92.7	-/86.9
穷人的 BERT ₆	-	81.1	-	-/80.2	87.6	-/90.4	65.0	90.3	-/88.5
忒修斯的伯特	51.1	82.3	-	89.0/-	89.5	-/89.6	68.2	91.5	-/88.7
迷你 LM ₆	49.2	84.0	-	88.4/-	91.0	-/91.0	71.5	92.0	-
TinyBERT ₆	54.0	84.5	84.5	90.6/86.3	91.1	88.0/91.1	73.4	93.0	90.1/89.6

表 5: TinyBERT 与其他基线在 GLUE 任务开发集上的比较。Mcc 指马修斯相关性, Pear/Spea 指皮尔逊/斯皮尔曼相关性。

与 TinyBERT₆ 的模型结构相同 (即 $M=6$)、 $d^r=768$, $d^r=3072$)。

直接比较结果如图 5 所示。我们可以看到, 在相同的架构和评估方法设置下, TinyBERT₆ 的表现优于所有基线。这进一步证实了 TinyBERT 的有效性。

B SQuAD v1.1 和 v2.0 的结果

我们还在问题解答 (QA) 任务中演示了 TinyBERT 的有效性: SQuAD v1.1 (Rajpurkar et al., 2016) 和 SQuAD v2.0 (Rajpurkar et al., 2018)。按照之前工作 (Devlin 等人, 2019 年) 中的学习过程, 我们将这两个任务视为序列标注问题, 即预测每个标记作为答案跨度的开始或结束的可能性。与 GLUE 任务的一个小区别是, 我们在原始训练数据集而不是增强数据集上执行预测层蒸馏, 这可以带来更好的性能。

结果表明, TinyBERT 的性能始终优于 4 层和 6 层基线, 这表明所提出的框架也适用于标记级标注任务。与序列级 GLUE 任务相比, 问题解答任务依赖于更微妙的知识来推断正确答案, 这增加了知识提炼的难度。如何构建更好的 QA-TinyBERT 将是我们今后的工作重点。

C 用 BERT 初始化 TinyBERT_{TINY}

在我们提出的两阶段学习框架中, 为了让 TinyBERT 有效地处理不同的下游任务, 我们提出了通用分布 (General Distillation, GD) 来捕捉通用领域知识, TinyBERT 通过它来学习以下知识

系统	SQuAD 1.1		SQuAD 2.0	
	EM	F1	EM	F1
BERT _{BASE} (教师)	80.7	88.4	74.5	77.7
<i>4 层学生模型</i>				
BERT ₄ -PKD	70.1	79.5	60.8	64.6
蒸馏器 ₄	71.8	81.2	60.6	64.1
迷你 LM ₄	-	-	-	69.7
TinyBERT ₄	72.7	82.1	68.2	71.8
<i>6 层学生模型</i>				
BERT ₆ -PKD	77.1	85.3	66.3	69.8
蒸馏器 ₆	78.1	86.2	66.0	69.5
迷你 LM ₆	-	-	-	76.4
TinyBERT ₆	79.7	87.5	74.7	77.7

表 6：基线和 TinyBERT 在问题解答任务上的结果（偏差）。MiniLM₄ 的架构 ($M=4$, $d=384$, $d_l=1536$) 比 TinyBERT₄ 更宽, MiniLM₆ 的架构与 TinyBERT₆ 相同 ($M=6$, $d=768$, $d=3072$)。

系统	MNLI-m (392k)	MNLI-mm (392k)	MRPC (3.5k)	CoLA (8.5k)	平均值
BERT _{TINY}	75.9	76.9	83.2	19.5	63.9
BERT _{TINY} (+TD)	79.2	79.7	82.9	12.4	63.6
TinyBERT (GD)	76.6	77.2	82.0	8.7	61.1
TinyBERT (GD+TD)	80.5	81.0	82.4	29.8	68.4

表 7：不同方法在预训练阶段的结果。TD 和 GD 分别指特定任务蒸馏法（无数据增强）和一般蒸馏法。结果在 dev 集上进行评估。

在预训练阶段，从教师 BERT 的中间层获取 TinyBERT。之后，将得到一个通用的 TinyBERT，并将其作为学生模型的初始化，用于下游任务的特定任务蒸馏（TD）。

在初步实验中，我们还尝试用直接预训练的 BERT_{TINY} 对 TinyBERT 进行初始化，然后对下游任务进行 TD。我们将这种计算方法称为 BERT_{TINY} (+TD)。表 7 中的结果显示，BERT_{TINY} (+TD) 的性能与 TinyBERT 的性能相当。

在 MRPC 和 CoLA 任务上，它甚至比 BERT_{TINY} 更差。我们推测，如果不在预训练阶段模仿 BERT_{BASE} 的行为，BERT_{TINY} 将得出与 BERT_{BASE} 模型不匹配的中间表征分布（如注意力矩阵和隐藏状态）。在经过微调的 BERT_{BASE} 的监督下，接下来的特定任务蒸馏会进一步扰乱 BERT_{TINY} 的已学分布/知识，最终导致在一些数据较少的任务上表现不佳。对于数据密集型任务（如 MNLI），虽然预先训练的分布已经受到干扰，但 TD 有足够的训练数据使 BERT_{TINY} 很好地获取特定任务的知识。

从表 7 的结果中我们发现，即使不执行 MLM 和 NSP 任务，GD 也能有效地将教师 BERT 的知识传授给学生 TinyBERT，并取得与 BERT_{TINY} 相同的结果（61.1 vs. 63.9）。此外，通过继续从经过微调的教师 BERT_{BASE} 中学习特定任务的知识，针对特定任务的提炼提高了 TinyBERT 的性能。

D 胶水详细信息

GLUE 数据集介绍如下：MNLI.多类型自然语言推理（Multi-Genre Natural Language Inference）是一项大规模、人群来源的词义分类任务（Williams et al. 给定一对前提、假设，目标是预测假设相对于前提而言是蕴涵、反义还是中性。

QQP。Quora 问题对是来自 Quora 网站的问题对集合。其任务是确定两个问题在语义上是否等同（Chen 等人，2018 年）。

QNLI.问题自然语言推理是斯坦福大学问题解答数据集的一个版本，由 Wang 等人（2018）转换为二元句对分类任务。给定一对问题、上下文。任务是判断上下文是否包含问题的答案

。

SST-2.斯坦福情感树库是一项二进制单句分类任务，目标是预测电影重新观看时的情感（Socher 等人，2013 年）。

CoLA.语言可接受性语料库是一项预测英语句子是否语法正确的任务（Warstadt 等人，2019 年）。

STS-B。语义文本相似性基准 (STS-B) 是从新闻标题和许多其他领域中抽取的句对集合 (Cer 等人, 2017 年)。该任务旨在评估两篇文本的相似程度, 分数从 1 到 5 分不等。

MRPC。Microsoft Research Paraphrase Corpus 是一个意译识别数据集, 系统旨在识别两个句子是否互为意译 (Dolan 和 Brockett, 2005 年)。

RTE. 识别文本关联是一项二元关联任务, 训练数据集很小 (Ben-tivogli 等人, 2009 年)。