MINILM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers

Wenhui Wang Furu Wei Li Dong Hangbo Bao Nan Yang Ming Zhou
Microsoft Research
{wenwan,fuwei,lidong1,t-habao,nanya,mingzhou}@microsoft.com

Abstract

Pre-trained language models (e.g., BERT (Devlin et al., 2018) and its variants) have achieved remarkable success in varieties of NLP tasks. However, these models usually consist of hundreds of millions of parameters which brings challenges for fine-tuning and online serving in real-life applications due to latency and capacity constraints. In this work, we present a simple and effective approach to compress large Transformer (Vaswani et al., 2017) based pre-trained models, termed as deep self-attention distillation. The small model (student) is trained by deeply mimicking the self-attention module, which plays a vital role in Transformer networks, of the large model (teacher). Specifically, we propose distilling the self-attention module of the last Transformer layer of the teacher, which is effective and flexible for the student. Furthermore, we introduce the scaled dot-product between values in the self-attention module as the new deep self-attention knowledge, in addition to the attention distributions (i.e., the scaled dot-product of queries and keys) that have been used in existing works. Moreover, we show that introducing a teacher assistant (Mirzadeh et al., 2019) also helps the distillation of large pre-trained Transformer models. Experimental results demonstrate that our monolingual model[1] outperforms state-of-the-art baselines in different parameter size of student models. In particular, it retains more than 99% accuracy on SQuAD 2.0 and several GLUE benchmark tasks using 50% of the Transformer parameters and computations of the teacher model. We also obtain competitive results in applying deep self-attention distillation to multilingual pre-trained models.

Correspondence to: Furu Wei <fuwei@microsoft.com>.

[1]The code and models are publicly available at https://aka.ms/minilm.

1. Introduction

Language model (LM) pre-training has achieved remarkable success for various natural language processing tasks (Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2018; Dong et al., 2019; Yang et al., 2019; Joshi et al., 2019; Liu et al., 2019). The pre-trained language models, such as BERT (Devlin et al., 2018) and its variants, learn contextualized text representations by predicting words given their context using large scale text corpora, and can be fine-tuned with additional task-specific layers to adapt to downstream tasks. However, these models usually contain hundreds of millions of parameters which brings challenges for fine-tuning and online serving in real-life applications for latency and capacity constraints.

Knowledge distillation (Hinton et al., 2015; Romero et al., 2015) (KD) has been proven to be a promising way to compress a large model (called the teacher model) into a small model (called the student model), which uses much fewer parameters and computations while achieving competitive results on downstream tasks. There have been some works that task-specifically distill pre-trained large LMs into small models (Tang et al., 2019; Turc et al., 2019b; Sun et al., 2019a; Aguilar et al., 2019). They first fine-tune the pre-trained LMs on specific tasks and then perform distillation. Task-specific distillation is effective, but fine-tuning large pre-trained models is still costly, especially for large datasets. Different from task-specific distillation, task-agnostic LM distillation mimics the behavior of the original pre-trained LMs and the student model can be directly fine-tuned on downstream tasks (Tsai et al., 2019; Sanh et al., 2019; Jiao et al., 2019; Sun et al., 2019b).

Previous works use soft target probabilities for masked language modeling predictions or intermediate representations of the teacher LM to guide the training of the task-agnostic student. DistilBERT (Sanh et al., 2019) employs a soft-label distillation loss and a cosine embedding loss, and initializes the student from the teacher by taking one layer out of two. But each Transformer layer of the student is required to have the same architecture as its teacher. TinyBERT (Jiao et al., 2019) and MOBILEBERT (Sun et al., 2019b) utilize

more fine-grained knowledge, including hidden states and self-attention distributions of Transformer networks, and transfer these knowledge to the student model layer-to-layer. To perform layer-to-layer distillation, TinyBERT adopts a uniform function to determine the mapping between the teacher and student layers, and uses a parameter matrix to linearly transform student hidden states. MOBILEBERT assumes the teacher and student have the same number of layers and introduces the bottleneck module to keep their hidden size the same.

In this work, we propose the deep self-attention distillation framework for task-agnostic Transformer based LM distillation. The key idea is to deeply mimic the self-attention modules which are the fundamentally important components in the Transformer based teacher and student models. Specifically, we propose distilling the self-attention module of the last Transformer layer of the teacher model. Compared with previous approaches, using knowledge of the last Transformer layer rather than performing layer-to-layer knowledge distillation alleviates the difficulties in layer mapping between the teacher and student models, and the layer number of our student model can be more flexible. Furthermore, we introduce the scaled dot-product between values in the self-attention module as the new deep self-attention knowledge, in addition to the attention distributions (i.e., the scaled dot-product of queries and keys) that has been used in existing works. Using scaled dot-product between self-attention values also converts representations of different dimensions into relation matrices with the same dimensions without introducing additional parameters to transform student representations, allowing arbitrary hidden dimensions for the student model. Finally, we show that introducing a teacher assistant (Mirzadeh et al., 2019) helps the distillation of large pre-trained Transformer based models and the proposed deep self-attention distillation can further boost the performance.

We conduct extensive experiments on downstream NLP tasks. Experimental results demonstrate that our monolingual model outperforms state-of-the-art baselines in different parameter size of student models. Specifically, the 6-layer model of 768 hidden dimensions distilled from $\text{BERT}_{\text{BASE}}$ is $2.0\times$ faster, while retaining more than 99% accuracy on SQuAD 2.0 and several GLUE benchmark tasks. Moreover, our multilingual model distilled from $\text{XLM-R}_{\text{Base}}$ also achieves competitive performance with much fewer Transformer parameters.

2. Preliminary

Multi-layer Transformers (Vaswani et al., 2017) have been the most widely-used network structures in state-of-the-art pre-trained models. In this section, we present a brief introduction to the Transformer network and the self-attention mechanism, which is the core component of the Transformer. We also present the existing approaches on knowledge distillation for Transformer networks, particularly in the context of distilling a large Transformer based pre-trained model into a small Transformer model.

2.1. Input Representation

Texts are tokenized to subword units by WordPiece (Wu et al., 2016) in BERT (Devlin et al., 2018). For example, the word "*forecasted*" is split to "*forecast*" and "*##ed*", where "*##*" indicates the pieces are belong to one word. A special boundary token `[SEP]` is used to separate segments if the input text contains more than one segment. At the beginning of the sequence, a special token `[CLS]` is added to obtain the representation of the whole input. The vector representations ($\{\mathbf{x}_i\}_{i=1}^{|x|}$) of input tokens are computed via summing the corresponding token embedding, absolute position embedding, and segment embedding.

2.2. Backbone Network: Transformer

Transformer (Vaswani et al., 2017) is used to encode contextual information for input tokens. The input vectors $\{\mathbf{x}_i\}_{i=1}^{|x|}$ are packed together into $\mathbf{H}^0 = [\mathbf{x}_1, \cdots, \mathbf{x}_{|x|}]$. Then stacked Transformer blocks compute the encoding vectors as:

$$\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1}),\ l \in [1, L] \tag{1}$$

where L is the number of Transformer layers, and the final output is $\mathbf{H}^L = [\mathbf{h}_1^L, \cdots, \mathbf{h}_{|x|}^L]$. The hidden vector $\mathbf{h}_i^L$ is used as the contextualized representation of $\mathbf{x}_i$. Each Transformer layer consists of a self-attention sub-layer and a fully connected feed-forward network. Residual connection (He et al., 2016) is employed around each of the two sub-layers, followed by layer normalization (Ba et al., 2016).

Self-Attention In each layer, Transformer uses multiple self-attention heads to aggregate the output vectors of the previous layer. For the l-th Transformer layer, the output of a self-attention head $\mathbf{AO}_{l,a},\ a \in [1, A_h]$ is computed via:

$$\mathbf{Q}_{l,a} = \mathbf{H}^{l-1}\mathbf{W}_{l,a}^Q,\ \mathbf{K}_{l,a} = \mathbf{H}^{l-1}\mathbf{W}_{l,a}^K,\ \mathbf{V}_{l,a} = \mathbf{H}^{l-1}\mathbf{W}_{l,a}^V \tag{2}$$

$$\mathbf{A}_{l,a} = \text{softmax}(\frac{\mathbf{Q}_{l,a}\mathbf{K}_{l,a}^\intercal}{\sqrt{d_k}}) \tag{3}$$

$$\mathbf{AO}_{l,a} = \mathbf{A}_{l,a}\mathbf{V}_{l,a} \tag{4}$$

where the previous layer's output $\mathbf{H}^{l-1} \in \mathbb{R}^{|x|\times d_h}$ is linearly projected to a triple of queries, keys and values using parameter matrices $\mathbf{W}_{l,a}^Q, \mathbf{W}_{l,a}^K, \mathbf{W}_{l,a}^V \in \mathbb{R}^{d_h\times d_k}$, respectively. $\mathbf{A}_{l,a} \in \mathbb{R}^{|x|\times|x|}$ indicates the attention distributions, which is computed by the scaled dot-product of queries and keys. A_h represents the number of self-attention heads. $d_k \times A_h$ is equal to the hidden dimension d_h in BERT.

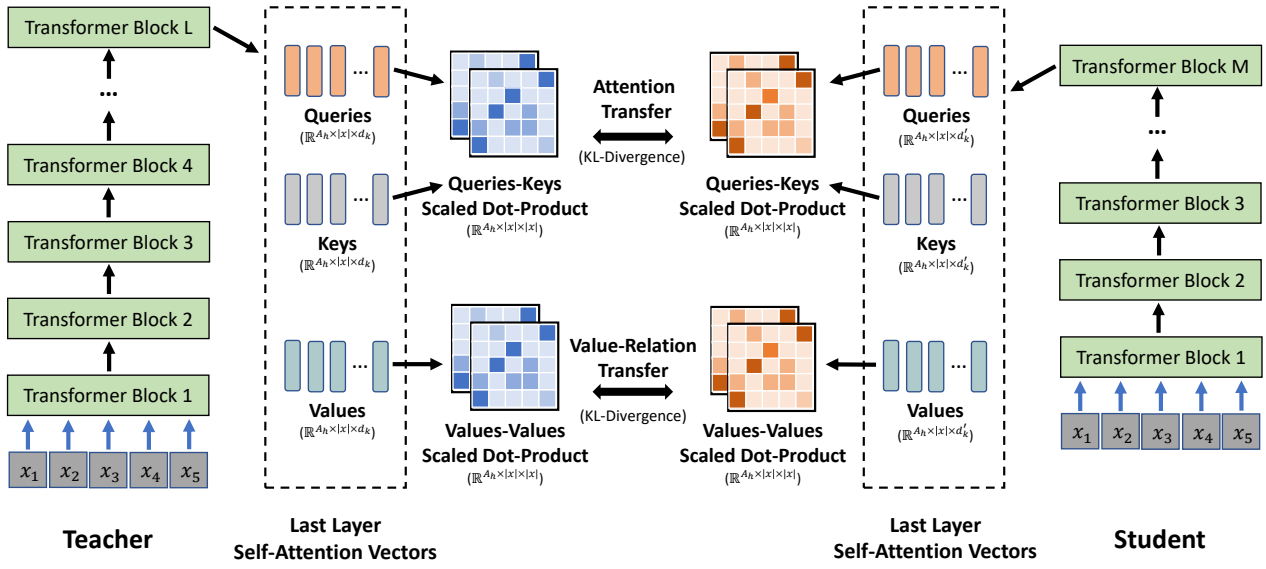

Figure 1. Overview of Deep Self-Attention Distillation. The student is trained by deeply mimicking the self-attention behavior of the last Transformer layer of the teacher. In addition to the self-attention distributions, we introduce the self-attention value-relation transfer to help the student achieve a deeper mimicry. Our student models are named as MINILM.

2.3. Transformer Distillation

Knowledge distillation (Hinton et al., 2015; Romero et al., 2015) is to train the small student model S on a transfer feature set with soft labels and intermediate representations provided by the large teacher model T. Knowledge distillation is modeled as minimizing the differences between teacher and student features:

$$\mathcal{L}_{\text{KD}} = \sum_{e \in \mathcal{D}} L(f^S(e), f^T(e)) \tag{5}$$

Where $\mathcal{D}$ denotes the training data, $f^S(\cdot)$ and $f^T(\cdot)$ indicate the features of student and teacher models respectively, $L(\cdot)$ represents the loss function. The mean squared error (MSE) and KL-divergence are often used as loss functions.

For Transformer based LM distillation, soft target probabilities for masked language modeling predictions, embedding layer outputs, self-attention distributions and outputs (hidden states) of each Transformer layer of the teacher model are used as features to help the training of the student. Soft labels and embedding layer outputs are used in DistillBERT. TinyBERT and MOBILEBERT further utilize self-attention distributions and outputs of each Transformer layer. For MOBILEBERT, the student is required to have the same number of layers as its teacher to perform layer-to-layer distillation. Besides, bottleneck and inverted bottleneck modules are introduced to keep the hidden size of the teacher and student are also the same. To transfer knowledge layer-to-layer, TinyBERT employs a uniform-function to map teacher and student layers. Since the hidden size of the student can be smaller than its teacher, a parameter matrix is introduced to transform the student features.

3. Deep Self-Attention Distillation

Figure 1 gives an overview of the deep self-attention distillation. The key idea is three-fold. First, we propose to train the student by deeply mimicking the self-attention module, which is the vital component in the Transformer, of the teacher's last layer. Second, we introduce transferring the relation between values (i.e., the scaled dot-product between values) to achieve a deeper mimicry, in addition to performing attention distributions (i.e., the scaled dot-product of queries and keys) transfer in the self-attention module. Moreover, we show that introducing a teacher assistant (Mirzadeh et al., 2019) also helps the distillation of large pre-trained Transformer models when the size gap between the teacher model and student model is large.

3.1. Self-Attention Distribution Transfer

The attention mechanism (Bahdanau et al., 2015) has been a highly successful neural network component for NLP tasks, which is also crucial for pre-trained LMs. Some works show that self-attention distributions of pre-trained LMs capture a rich hierarchy of linguistic information (Jawahar et al., 2019; Clark et al., 2019). Transferring self-attention distributions has been used in previous works for Transformer distillation (Jiao et al., 2019; Sun et al., 2019b; Aguilar et al., 2019). We also utilize the self-attention distributions to help the training of the student. Specifically, we minimize the KL-divergence between the self-attention distributions of

Table 1. Comparison with previous task-agnostic Transformer based LM distillation approaches.

Approach	Teacher Model	Distilled Knowledge	Layer-to-Layer Distillation	Requirements on the number of layers of students	Requirements on the hidden size of students
DistillBERT	$\text{BERT}_{\text{BASE}}$	Soft target probabilities Embedding outputs			✓
TinyBERT	$\text{BERT}_{\text{BASE}}$	Embedding outputs Hidden states Self-Attention distributions	✓		
MOBILEBERT	$\text{IB-BERT}_{\text{LARGE}}$	Soft target probabilities Hidden states Self-Attention distributions	✓	✓	✓
MINILM	$\text{BERT}_{\text{BASE}}$	Self-Attention distributions Self-Attention value relation			

the teacher and student:

$$\mathcal{L}_{\text{AT}} = \frac{1}{A_h|x|}\sum_{a=1}^{A_h}\sum_{t=1}^{|x|} D_{KL}(\mathbf{A}_{L,a,t}^{T} \parallel \mathbf{A}_{M,a,t}^{S}) \quad (6)$$

Where $|x|$ and A_h represent the sequence length and the number of attention heads. L and M represent the number of layers for the teacher and student. $\mathbf{A}_L^T$ and $\mathbf{A}_M^S$ are the attention distributions of the last Transformer layer for the teacher and student, respectively. They are computed by the scaled dot-product of queries and keys.

Different from previous works which transfer teacher's knowledge layer-to-layer, we only use the attention maps of the teacher's last Transformer layer. Distilling attention knowledge of the last Transformer layer allows more flexibility for the number of layers of our student models, avoids the effort of finding the best layer mapping.

3.2. Self-Attention Value-Relation Transfer

In addition to the attention distributions, we propose using the relation between values in the self-attention module to guide the training of the student. The value relation is computed via the multi-head scaled dot-product between values. The KL-divergence between the value relation of the teacher and student is used as the training objective:

$$\mathbf{VR}_{L,a}^{T} = \text{softmax}(\frac{\mathbf{V}_{L,a}^{T}\mathbf{V}_{L,a}^{T\intercal}}{\sqrt{d_k}}) \quad (7)$$

$$\mathbf{VR}_{M,a}^{S} = \text{softmax}(\frac{\mathbf{V}_{M,a}^{S}\mathbf{V}_{M,a}^{S\intercal}}{\sqrt{d_k'}}) \quad (8)$$

$$\mathcal{L}_{\text{VR}} = \frac{1}{A_h|x|}\sum_{a=1}^{A_h}\sum_{t=1}^{|x|} D_{KL}(\mathbf{VR}_{L,a,t}^{T} \parallel \mathbf{VR}_{M,a,t}^{S}) \quad (9)$$

Where $\mathbf{V}_{L,a}^T \in \mathbb{R}^{|x|\times d_k}$ and $\mathbf{V}_{M,a}^S \in \mathbb{R}^{|x|\times d_k'}$ are the values of an attention head in self-attention module for the teacher's and student's last Transformer layer. $\mathbf{VR}_L^T \in \mathbb{R}^{A_h\times|x|\times|x|}$ and $\mathbf{VR}_M^S \in \mathbb{R}^{A_h\times|x|\times|x|}$ are the value relation of the last Transformer layer for teacher and student, respectively.

The training loss is computed via summing the attention distribution transfer loss and value-relation transfer loss:

$$\mathcal{L} = \mathcal{L}_{\text{AT}} + \mathcal{L}_{\text{VR}} \quad (10)$$

Introducing the relation between values enables the student to deeply mimic the teacher's self-attention behavior. Moreover, using the scaled dot-product converts vectors of different hidden dimensions into the relation matrices with the same size, which allows our students to use more flexible hidden dimensions and avoids introducing additional parameters to transform the student's representations.

3.3. Teacher Assistant

Following Mirzadeh et al. (2019), we introduce a teacher assistant (i.e., intermediate-size student model) to further improve the model performance of smaller students.

Assuming the teacher model consists of L-layer Transformer with d_h hidden size, the student model has M-layer Transformer with d_h' hidden size. For smaller students ($M \leq \frac{1}{2}L$, $d_h' \leq \frac{1}{2}d_h$), we first distill the teacher into a teacher assistant with L-layer Transformer and d_h' hidden size. The assistant model is then used as the teacher to guide the training of the final student. The introduction of a teacher assistant bridges the size gap between teacher and smaller student models, helps the distillation of Transformer based pre-trained LMs. Moreover, combining deep self-attention distillation with a teacher assistant brings further improvements for smaller student models.

3.4. Comparison with Previous Work

Table 1 presents the comparison with previous approaches (Sanh et al., 2019; Jiao et al., 2019; Sun et al.,

Table 2. Comparison between the publicly released 6-layer models with 768 hidden size distilled from $\text{BERT}_{\text{BASE}}$. We compare task-agnostic distilled models without task-specific distillation and data augmentation. We report F1 for SQuAD 2.0, and accuracy for other datasets. The GLUE results of DistillBERT are taken from Sanh et al. (2019). We report the SQuAD 2.0 result by fine-tuning their released model[3]. For TinyBERT, we fine-tune the latest version of their public model[4] for a fair comparison. The results of our fine-tuning experiments are an average of 4 runs for each task.

Model	#Param	SQuAD2	MNLI-m	SST-2	QNLI	CoLA	RTE	MRPC	QQP	Average
$\text{BERT}_{\text{BASE}}$	109M	76.8	84.5	93.2	91.7	58.9	68.6	87.3	91.3	81.5
DistillBERT	66M	70.7	79.0	90.7	85.3	43.6	59.9	87.5	84.9	75.2
TinyBERT	66M	73.1	83.5	91.6	90.5	42.8	**72.2**	**88.4**	90.6	79.1
MINILM	66M	**76.4**	**84.0**	**92.0**	**91.0**	**49.2**	71.5	**88.4**	**91.0**	**80.4**

2019b). MOBILEBERT proposes using a specially designed inverted bottleneck model, which has the same model size as $\text{BERT}_{\text{LARGE}}$, as the teacher. The other methods utilize $\text{BERT}_{\text{BASE}}$ to conduct experiments. For the knowledge used for distillation, our method introduces the scaled dot-product between values in the self-attention module as the new knowledge to deeply mimic teacher's self-attention behavior. TinyBERT and MOBILEBERT transfer knowledge of the teacher to the student layer-to-layer. MOBILEBERT assumes the student has the same number of layers as its teacher. TinyBERT employs a uniform strategy to determine its layer mapping. DistillBERT initializes the student with teacher's parameters, therefore selecting layers of the teacher model is still needed. MINILM distills the self-attention knowledge of the teacher's last Transformer layer, which allows the flexible number of layers for the students and alleviates the effort of finding the best layer mapping. Student hidden size of DistillBERT and MOBILEBERT is required to be the same as its teacher. TinyBERT uses a parameter matrix to transform student hidden states. Using value relation allows our students to use arbitrary hidden size without introducing additional parameters.

4. Experiments

We conduct distillation experiments in different parameter size of student models, and evaluate the distilled models on downstream tasks including extractive question answering and the GLUE benchmark.

4.1. Distillation Setup

We use the uncased version of $\text{BERT}_{\text{BASE}}$ as our teacher. $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018) is a 12-layer Transformer with 768 hidden size, and 12 attention heads, which contains about 109M parameters. The number of heads of attention distributions and value relation are set to 12 for student models. We use documents of English Wikipedia[2] and BookCorpus (Zhu et al., 2015) for the pre-training data, following the preprocess and the WordPiece tokenization of Devlin et al. (2018). The vocabulary size is $30,522$. The maximum sequence length is 512. We use Adam (Kingma & Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$. We train the 6-layer student model with 768 hidden size using 1024 as the batch size and 5e-4 as the peak learning rate for $400,000$ steps. For student models of other architectures, the batch size and peak learning rate are set to 256 and 3e-4, respectively. We use linear warmup over the first $4,000$ steps and linear decay. The dropout rate is 0.1. The weight decay is 0.01.

We also use an in-house pre-trained Transformer model in the $\text{BERT}_{\text{BASE}}$ size as the teacher model, and distill it into 12-layer and 6-layer student models with 384 hidden size. For the 12-layer model, we use Adam (Kingma & Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The model is trained using 2048 as the batch size and 6e-4 as the peak learning rate for $400,000$ steps. The batch size and peak learning rate are set to 512 and 4e-4 for the 6-layer model. The rest hyper-parameters are the same as above BERT based distilled models.

For the training of multilingual MINILM models, we use Adam (Kingma & Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$. We train the 12-layer student model using 256 as the batch size and 3e-4 as the peak learning rate for $1,000,000$ steps. The 6-layer student model is trained using 512 as the batch size and 6e-4 as the peak learning rate for $400,000$ steps.

We distill our student models using 8 V100 GPUs with mixed precision training. Following Sun et al. (2019a) and Jiao et al. (2019), the inference time is evaluated on the QNLI training set with the same hyper-parameters. We report the average running time of 100 batches on a single P100 GPU.

4.2. Downstream Tasks

Following previous language model pre-training (Devlin et al., 2018; Liu et al., 2019) and task-agnostic pre-trained language model distillation (Sanh et al., 2019; Jiao et al., 2019; Sun et al., 2019b), we evaluate our distilled models on the extractive question answering and GLUE benchmark.

[2]Wikipedia version: enwiki-20181101.

Table 3. Comparison between student models of different architectures distilled from $\text{BERT}_{\text{BASE}}$. M and d'_h indicate the number of layers and hidden dimension of the student model. TA indicates teacher assistant[5]. The fine-tuning results are averaged over 4 runs.

Architecture	#Param	Model	SQuAD 2.0	MNLI-m	SST-2	Average
M=6;d'_h=384	22M	MLM-KD (Soft-Label Distillation)	67.9	79.6	89.8	79.1
		TinyBERT	71.6	81.4	90.2	81.1
		MINILM	72.4	82.2	91.0	81.9
		MINILM (w/ TA)	**72.7**	**82.4**	**91.2**	**82.1**
M=4;d'_h=384	19M	MLM-KD (Soft-Label Distillation)	65.3	77.7	88.8	77.3
		TinyBERT	66.7	79.2	88.5	78.1
		MINILM	69.4	80.3	90.2	80.0
		MINILM (w/ TA)	**69.7**	**80.6**	**90.6**	**80.3**
M=3;d'_h=384	17M	MLM-KD (Soft-Label Distillation)	59.9	75.2	88.0	74.4
		TinyBERT	63.6	77.4	88.4	76.5
		MINILM	66.2	78.8	89.3	78.1
		MINILM (w/ TA)	**66.9**	**79.1**	**89.7**	**78.6**

Table 4. The number of Embedding (Emd) and Transformer (Trm) parameters, and inference time for different models.

#Layers	Hidden Size	#Param (Emd)	#Param (Trm)	Inference Time
12	768	23.4M	85.1M	93.1s (1.0×)
6	768	23.4M	42.5M	46.9s (2.0×)
12	384	11.7M	21.3M	34.8s (2.7×)
6	384	11.7M	10.6M	17.7s (5.3×)
4	384	11.7M	7.1M	12.0s (7.8×)
3	384	11.7M	5.3M	9.2s (10.1×)

Extractive Question Answering Given a passage P, the task is to select a contiguous span of text in the passage by predicting its start and end positions to answer the question Q. We evaluate on SQuAD 2.0 (Rajpurkar et al., 2018), which has served as a major question answering benchmark.

Following BERT (Devlin et al., 2018), we pack the question and passage tokens together with special tokens, to form the input: "`[CLS]` Q `[SEP]` P `[SEP]`". Two linear output layers are introduced to predict the probability of each token being the start and end positions of the answer span. The questions that do not have an answer are treated as having an answer span with start and end at the `[CLS]` token.

GLUE The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) consists of nine sentence-level classification tasks, including Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2018), Stanford Sentiment Treebank (SST) (Socher et al., 2013), Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005), Semantic Textual Similarity Benchmark (STS) (Cer et al., 2017), Quora Question Pairs (QQP) (Chen et al., 2018), Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018), Question Natural Language Inference (QNLI) (Rajpurkar et al., 2016), Recognizing Textual Entailment (RTE) (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) and Winograd Natural Language Inference (WNLI) (Levesque et al., 2012). We add a linear classifier on top of the `[CLS]` token to predict label probabilities.

4.3. Main Results

Previous works (Sanh et al., 2019; Sun et al., 2019a; Jiao et al., 2019) usually distill $\text{BERT}_{\text{BASE}}$ into a 6-layer student model with 768 hidden size. We first conduct distillation experiments using the same student architecture. Results on SQuAD 2.0 and GLUE dev sets are presented in Table 2. Since MOBILEBERT distills a specially designed teacher with the inverted bottleneck modules, which has the same model size as $\text{BERT}_{\text{LARGE}}$, into a 24-layer student using the bottleneck modules, we do not compare our models with MOBILEBERT. MINILM outperforms DistillBERT[3] and TinyBERT[4] across most tasks. Our model exceeds the two state-of-the-art models by 3.0+% F1 on SQuAD 2.0 and 5.0+% accuracy on CoLA. We present the inference time for models in different parameter size in Table 4. Our 6-layer 768-dimensional student model is 2.0× faster than original $\text{BERT}_{\text{BASE}}$, while retaining more than 99% performance on a variety of tasks, such as SQuAD 2.0 and MNLI.

We also conduct experiments for smaller student models. We compare MINILM with our implemented MLM-KD

[3]The public model of DistillBERT is obtained from https://github.com/huggingface/transformers/tree/master/examples/distillation

[4]We use the 2nd version TinyBERT from https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TinyBERT

Table 5. Effectiveness of self-attention value-relation (Value-Rel) transfer. The fine-tuning results are averaged over 4 runs.

Architecture	Model	SQuAD2	MNLI-m	SST-2
M=6;d'_h=384	MINILM	**72.4**	**82.2**	**91.0**
	-Value-Rel	71.0	80.9	89.9
M=4;d'_h=384	MINILM	**69.4**	**80.3**	**90.2**
	-Value-Rel	67.5	79.0	89.2
M=3;d'_h=384	MINILM	**66.2**	**78.8**	**89.3**
	-Value-Rel	64.2	77.8	88.3

(knowledge distillation using soft target probabilities for masked language modeling predictions) and TinyBERT, which are trained using the same data and hyper-parameters. The results on SQuAD 2.0, MNLI and SST-2 dev sets are shown in Table 3. MINILM outperforms soft label distillation and our implemented TinyBERT on the three tasks. Deep self-attention distillation is also effective for smaller models. Moreover, we show that introducing a teacher assistant[5] is also helpful in Transformer based pre-trained LM distillation, especially for smaller models. Combining deep self-attention distillation with a teacher assistant achieves further improvement for smaller student models.

4.4. Ablation Studies

We do ablation tests on several tasks to analyze the contribution of self-attention value-relation transfer. The dev results of SQuAD 2.0, MNLI and SST-2 are illustrated in Table 5, using self-attention value-relation transfer positively contributes to the final results for student models in different parameter size. Distilling the fine-grained knowledge of value relation helps the student model deeply mimic the self-attention behavior of the teacher, which further improves model performance.

We also compare different loss functions over values in the self-attention module. We compare our proposed value relation with mean squared error (MSE) over the teacher and student values. An additional parameter matrix is introduced to transform student values if the hidden dimension of the student is smaller than its teacher. The dev results on three tasks are presented in Table 6. Using value relation achieves better performance. Specifically, our method brings about 1.0% F1 improvement on the SQuAD benchmark. Moreover, there is no need to introduce additional parameters for our method. We have also tried to transfer the relation between hidden states. But we find the performance of student models are unstable for different teacher models.

To show the effectiveness of distilling self-attention knowledge of the teacher's last Transformer layer, we compare our method with layer-to-layer distillation. We transfer the same knowledge and adopt a uniform strategy as in Jiao et al. (2019) to map teacher and student layers to perform layer-to-layer distillation. The dev results on three tasks are presented in Table 7. MINILM achieves better results. It also alleviates the difficulties in layer mapping between the teacher and student. Besides, distilling the teacher's last Transformer layer requires less computation than layer-to-layer distillation, results in faster training speed.

[5]The teacher assistant is only introduced for the model MINILM (w/ TA). The model MINILM in different tables is directly distilled from its teacher model.

Table 6. Comparison between different loss functions: KL-divergence over the value relation (the scaled dot-product between values) and mean squared error (MSE) over values. A parameter matrix is introduced to transform student values to have the same dimensions as the teacher values (Jiao et al., 2019). The fine-tuning results are an average of 4 runs for each task.

Architecture	Model	SQuAD2	MNLI-m	SST-2
M=6;d'_h=384	MINILM	**72.4**	**82.2**	**91.0**
	Value-MSE	71.4	82.0	90.8
M=4;d'_h=384	MINILM	**69.4**	**80.3**	**90.2**
	Value-MSE	68.3	80.1	89.9
M=3;d'_h=384	MINILM	**66.2**	**78.8**	**89.3**
	Value-MSE	65.5	78.4	**89.3**

5. Discussion

5.1. Better Teacher Better Student

We report the results of MINILM distilled from an in-house pre-trained Transformer model following UNILM (Dong et al., 2019; Bao et al., 2020) in the $\text{BERT}_{\text{BASE}}$ size. The teacher model is trained using similar pre-training datasets as in $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019), which includes 160GB text corpora from English Wikipedia, BookCorpus (Zhu et al., 2015), OpenWebText[6], CC-News (Liu et al., 2019), and Stories (Trinh & Le, 2018). We distill the teacher model into 12-layer and 6-layer models with 384 hidden size using the same corpora. The 12x384 model is used as the teacher assistant to train the 6x384 model. We present the dev results of SQuAD 2.0 and GLUE benchmark in Table 8, the results of MINILM are significantly improved. The 12x384 MINILM achieves 2.7× speedup while performs competitively better than $\text{BERT}_{\text{BASE}}$ in SQuAD 2.0 and GLUE benchmark datasets.

5.2. MINILM for NLG Tasks

We also evaluate MINILM on natural language generation tasks, such as question generation and abstractive summarization. Following Dong et al. (2019), we fine-tune

[6]skylion007.github.io/OpenWebTextCorpus

Table 7. Comparison between distilling knowledge of the teacher's last Transformer layer and layer-to-layer distillation. We adopt a uniform strategy as in Jiao et al. (2019) to determine the mapping between teacher and student layers. The fine-tuning results are an average of 4 runs for each task.

Architecture	Model	SQuAD 2.0	MNLI-m	SST-2	Average
M=6;d'_h=384	MINILM	**72.4**	**82.2**	**91.0**	**81.9**
	+Layer-to-Layer Distillation	71.6	81.8	90.6	81.3
M=4;d'_h=384	MINILM	**69.4**	**80.3**	**90.2**	**80.0**
	+Layer-to-Layer Distillation	67.6	79.9	89.6	79.0
M=3;d'_h=384	MINILM	**66.2**	**78.8**	**89.3**	**78.1**
	+Layer-to-Layer Distillation	64.8	77.7	88.6	77.0

Table 8. The results of MINILM distilled from an in-house pre-trained Transformer model ($\text{BERT}_{\text{BASE}}$ size, 12-layer Transformer, 768-hidden size, and 12 self-attention heads) on SQuAD 2.0 and GLUE benchmark. We report our 12-layer^a and 6-layer^b models with 384 hidden size. The fine-tuning results are averaged over 4 runs.

Model	#Param	SQuAD2	MNLI-m	SST-2	QNLI	CoLA	RTE	MRPC	QQP	Average
$\text{BERT}_{\text{BASE}}$	109M	76.8	84.5	93.2	91.7	58.9	68.6	87.3	91.3	81.5
MINILM^a	33M	81.7	85.7	93.0	91.5	58.5	73.3	89.5	91.3	83.1
MINILM^b (w/ TA)	22M	75.6	83.3	91.5	90.5	47.5	68.8	88.9	90.6	79.6

Table 9. Question generation results of our 12-layer^a and 6-layer^b models with 384 hidden size on SQuAD 1.1. The first block follows the data split in Du & Cardie (2018), while the second block is the same as in Zhao et al. (2018). MTR is short for METEOR, RG for ROUGE, and B for BLEU.

	#Param	B-4	MTR	RG-L
(Du & Cardie, 2018)		15.16	19.12	-
(Zhang & Bansal, 2019)		18.37	22.65	46.68
$\text{UNILM}_{\text{LARGE}}$	340M	22.78	25.49	51.57
MINILM^a	33M	21.07	24.09	49.14
MINILM^b (w/ TA)	22M	20.31	23.43	48.21
(Zhao et al., 2018)		16.38	20.25	44.48
(Zhang & Bansal, 2019)		20.76	24.20	48.91
$\text{UNILM}_{\text{LARGE}}$	340M	24.32	26.10	52.69
MINILM^a	33M	23.27	25.15	50.60
MINILM^b (w/ TA)	22M	22.01	24.24	49.51

MINILM as a sequence-to-sequence model by employing a specific self-attention mask.

Question Generation We conduct experiments for the answer-aware question generation task (Du & Cardie, 2018). Given an input passage and an answer, the task is to generate a question that asks for the answer. The SQuAD 1.1 dataset (Rajpurkar et al., 2016) is used for evaluation. The results of MINILM, $\text{UNILM}_{\text{LARGE}}$ and several state-of-the-art models are presented in Table 9, our 12x384 and 6x384 distilled models achieve competitive performance on the question generation task.

Abstractive Summarization We evaluate MINILM on two abstractive summarization datasets, i.e., XSum (Narayan et al., 2018), and the non-anonymized version of CNN/DailyMail (See et al., 2017). The generation task is to condense a document into a concise and fluent summary, while conveying its key information. We report ROUGE scores (Lin, 2004) on the datasets. Table 10 presents the results of MINILM, baseline, several state-of-the-art models and pre-trained Transformer models. Our 12x384 model outperforms BERT based method BERTSUMABS (Liu & Lapata, 2019) and the pre-trained sequence-to-sequence model $\text{MASS}_{\text{BASE}}$ (Song et al., 2019) with much fewer parameters. Moreover, our 6x384 MINILM also achieves competitive performance.

5.3. Multilingual MINILM

We conduct experiments on task-agnostic knowledge distillation of multilingual pre-trained models. We use the $\text{XLM-R}_{\text{Base}}$[7] (Conneau et al., 2019) as the teacher and distill the model into 12-layer and 6-layer models with 384 hidden size using the same corpora. The 6x384 model is trained using the 12x384 model as the teacher assistant. Given the vocabulary size of multilingual pre-trained models is much larger than monolingual models (30k for monolingual BERT, 250k for XLM-R), soft-label distillation for multilingual pre-trained models requires more computation. MINILM only uses the deep self-attention knowledge of the teacher's last Transformer layer. The training speed

[7] We use the v0 version of $\text{XLM-R}_{\text{Base}}$ in our distillation and fine-tuning experiments.

Table 10. Abstractive summarization results of our 12-layer[a] and 6-layer[b] models with 384 hidden size on CNN/DailyMail and XSum. The evaluation metric is the F1 version of ROUGE (RG) scores.

Model	#Param	CNN/DailyMail			XSum		
		RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
LEAD-3		40.42	17.62	36.67	16.30	1.60	11.95
PTRNET (See et al., 2017)		39.53	17.28	36.38	28.10	8.02	21.72
Bottom-Up (Gehrmann et al., 2018)		41.22	18.68	38.34	-	-	-
$\text{UNILM}_{\text{LARGE}}$ (Dong et al., 2019)	340M	43.08	20.43	40.34	-	-	-
$\text{BART}_{\text{LARGE}}$ (Lewis et al., 2019a)	400M	44.16	21.28	40.90	45.14	22.27	37.25
$\text{T5}_{11\text{B}}$ (Raffel et al., 2019)	11B	43.52	21.55	40.69	-	-	-
$\text{MASS}_{\text{BASE}}$ (Song et al., 2019)	123M	42.12	19.50	39.01	39.75	17.24	31.95
BERTSUMABS (Liu & Lapata, 2019)	156M	41.72	19.39	38.76	38.76	16.33	31.15
T5_{BASE} (Raffel et al., 2019)	220M	42.05	20.34	39.40	-	-	-
MINILM[a]	33M	42.66	19.91	39.73	40.43	17.72	32.60
MINILM[b] (w/ TA)	22M	41.57	19.21	38.64	38.79	16.39	31.10

Table 11. Cross-lingual classification results of our 12-layer[a] and 6-layer[b] multilingual models with 384 hidden size on XNLI. We report the accuracy on each of the 15 XNLI languages and the average accuracy. Results of mBERT, XLM-100 and $\text{XLM-R}_{\text{Base}}$ are from Conneau et al. (2019).

Model	#Layers	#Hidden	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
mBERT	12	768	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
XLM-100	16	1280	83.2	76.7	77.7	74.0	72.7	74.1	72.7	68.7	68.6	72.9	68.9	72.5	65.6	58.2	62.4	70.7
$\text{XLM-R}_{\text{Base}}$	12	768	84.6	78.4	78.9	76.8	75.9	77.3	75.4	73.2	71.5	75.4	72.5	74.9	71.1	65.2	66.5	74.5
MINILM[a]	12	384	81.5	74.8	75.7	72.9	73.0	74.5	71.3	69.7	68.8	72.1	67.8	70.0	66.2	63.3	64.2	71.1
MINILM[b] (w/ TA)	6	384	79.2	72.3	73.1	70.3	69.1	72.0	69.1	64.5	64.9	69.0	66.0	67.8	62.9	59.0	60.6	68.0

Table 12. The number of Transformer (Trm) and Embedding (Emd) parameters for different multilingual pre-trained models and our distilled models.

Model	#Layers	Hidden Size	#Vocab	#Param (Trm)	#Param (Emd)
mBERT	12	768	110k	85M	85M
XLM-15	12	1024	95k	151M	97M
XLM-100	16	1280	200k	315M	256M
$\text{XLM-R}_{\text{Base}}$	12	768	250k	85M	192M
MINILM[a]	12	384	250k	21M	96M
MINILM[b]	6	384	250k	11M	96M

of MINILM is much faster than soft-label distillation for multilingual pre-trained models.

We evaluate the student models on cross-lingual natural language inference (XNLI) benchmark (Conneau et al., 2018) and cross-lingual question answering (MLQA) benchmark (Lewis et al., 2019b).

XNLI Table 11 presents XNLI results of our distilled students and several pre-trained LMs. Following Conneau et al. (2019), we select the best single model on the joint dev set of all the languages. We present the number of Transformer and embedding parameters for different multilingual pre-trained models and our distilled models in Table 12. MINILM achieves competitive performance on XNLI with much fewer Transformer parameters. Moreover, the 12x384 MINILM compares favorably with mBERT (Devlin et al., 2018) and XLM (Lample & Conneau, 2019) trained on the MLM objective.

MLQA Table 13 shows cross-lingual question answering results. Following Lewis et al. (2019b), we adopt SQuAD 1.1 as training data and use MLQA English development data for early stopping. The 12x384 MINILM performs competitively better than mBERT and XLM. Our 6-layer MINILM also achieves competitive performance.

6. Related Work

6.1. Pre-trained Language Models

Unsupervised pre-training of language models (Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2018; Baevski et al., 2019; Song et al., 2019; Dong et al., 2019; Yang et al., 2019; Joshi et al., 2019; Liu et al., 2019; Lewis et al., 2019a; Raffel et al., 2019) has achieved significant improvements for a wide range of NLP tasks. Early methods for pre-training (Peters et al., 2018; Radford et al., 2018) were based on standard language models. Re-

Table 13. Cross-lingual question answering results of our 12-layer[a] and 6-layer[b] multilingual models with 384 hidden size on MLQA. We report the F1 and EM (exact match) scores on each of the 7 MLQA languages. Results of mBERT and XLM-15 are taken from Lewis et al. (2019b). † indicates results of XLM-R_{Base} taken from Conneau et al. (2019). We also report our fine-tuned results (‡) of XLM-R_{Base}.

Model	#Layers	#Hidden	en	es	de	ar	hi	vi	zh	Avg
mBERT	12	768	77.7 / 65.2	64.3 / 46.6	57.9 / 44.3	45.7 / 29.8	43.8 / 29.7	57.1 / 38.6	57.5 / 37.3	57.7 / 41.6
XLM-15	12	1024	74.9 / 62.4	68.0 / 49.8	62.2 / 47.6	54.8 / 36.3	48.8 / 27.3	61.4 / 41.8	61.1 / 39.6	61.6 / 43.5
XLM-R_{Base}†	12	768	77.8 / 65.3	67.2 / 49.7	60.8 / 47.1	53.0 / 34.7	57.9 / 41.7	63.1 / 43.1	60.2 / 38.0	62.9 / 45.7
XLM-R_{Base}‡	12	768	80.3 / 67.4	67.0 / 49.2	62.7 / 48.3	55.0 / 35.6	60.4 / 43.7	66.5 / 45.9	62.3 / 38.3	64.9 / 46.9
MINILM[a]	12	384	79.4 / 66.5	66.1 / 47.5	61.2 / 46.5	54.9 / 34.9	58.5 / 41.3	63.1 / 42.1	59.0 / 33.8	63.2 / 44.7
MINILM[b] (w/ TA)	6	384	75.5 / 61.9	55.6 / 38.2	53.3 / 37.7	43.5 / 26.2	46.9 / 31.5	52.0 / 33.1	48.8 / 27.3	53.7 / 36.6

cently, BERT (Devlin et al., 2018) proposes to use a masked language modeling objective to train a deep bidirectional Transformer encoder, which learns interactions between left and right context. Liu et al. (2019) show that very strong performance can be achieved by training the model longer over more data. Joshi et al. (2019) extend BERT by masking contiguous random spans. Yang et al. (2019) predict masked tokens auto-regressively in a permuted order.

To extend the applicability of pre-trained Transformers for NLG tasks. Dong et al. (2019) extend BERT by utilizing specific self-attention masks to jointly optimize bidirectional, unidirectional and sequence-to-sequence masked language modeling objectives. Raffel et al. (2019) employ an encoder-decoder Transformer and perform sequence-to-sequence pre-training by predicting the masked tokens in the encoder and decoder. Different from Raffel et al. (2019), Lewis et al. (2019a) predict tokens auto-regressively in the decoder.

6.2. Knowledge Distillation

Knowledge distillation has proven a promising way to compress large models while maintaining accuracy. It transfers the knowledge of a large model or an ensemble of neural networks (teacher) to a single lightweight model (student). Hinton et al. (2015) first propose transferring the knowledge of the teacher to the student by using its soft target distributions to train the distilled model. Romero et al. (2015) introduce intermediate representations from hidden layers of the teacher to guide the training of the student. Knowledge of the attention maps (Zagoruyko & Komodakis, 2017; Hu et al., 2018) is also introduced to help the training.

In this work, we focus on task-agnostic knowledge distillation of large pre-trained Transformer based language models. There have been some works that task-specifically distill the fine-tuned language models on downstream tasks. Tang et al. (2019) distill fine-tuned BERT into an extremely small bidirectional LSTM. Turc et al. (2019a) initialize the student with a small pre-trained LM during task-specific distillation. Sun et al. (2019a) introduce the hidden states from every k layers of the teacher to perform knowledge distillation layer-to-layer. Aguilar et al. (2019) further introduce the knowledge of self-attention distributions and propose progressive and stacked distillation methods. Task-specific distillation requires to first fine-tune the large pre-trained LMs on downstream tasks and then perform knowledge transfer. The procedure of fine-tuning large pre-trained LMs is costly and time-consuming, especially for large datasets.

For task-agnostic distillation, the distilled model mimics the original large pre-trained LM and can be directly fine-tuned on downstream tasks. In practice, task-agnostic compression of pre-trained LMs is more desirable. MiniBERT (Tsai et al., 2019) uses the soft target distributions for masked language modeling predictions to guide the training of the multilingual student model and shows its effectiveness on sequence labeling tasks. DistillBERT (Sanh et al., 2019) uses the soft label and embedding outputs of the teacher to train the student. TinyBERT (Jiao et al., 2019) and MOBILEBERT (Sun et al., 2019b) further introduce self-attention distributions and hidden states to train the student. MOBILEBERT employs inverted bottleneck and bottleneck modules for teacher and student to make their hidden dimensions the same. The student model of MOBILEBERT is required to have the same number of layers as its teacher to perform layer-to-layer distillation. Besides, MOBILEBERT proposes a bottom-to-top progressive scheme to transfer teacher's knowledge. TinyBERT uses a uniform-strategy to map the layers of teacher and student when they have different number of layers, and a linear matrix is introduced to transform the student hidden states to have the same dimensions as the teacher. TinyBERT also introduces task-specific distillation and data augmentation for downstream tasks, which brings further improvements.

Different from previous works, our method employs the self-attention distributions and value relation of the teacher's last Transformer layer to help the student deeply mimic the self-attention behavior of the teacher. Using knowledge of the last Transformer layer instead of layer-to-layer distillation avoids restrictions on the number of student layers and the effort of finding the best layer mapping. Distilling relation between self-attention values allows the hidden size of students to be more flexible and avoids introducing linear matrices to transform student representations.

7. Conclusion

In this work, we propose a simple and effective knowledge distillation method to compress large pre-trained Transformer based language models. The student is trained by deeply mimicking the teacher's self-attention modules, which are the vital components of the Transformer networks. We propose using the self-attention distributions and value relation of the teacher's last Transformer layer to guide the training of the student, which is effective and flexible for the student models. Moreover, we show that introducing a teacher assistant also helps pre-trained Transformer based LM distillation, and the proposed deep self-attention distillation can further boost the performance. Our student model distilled from $\text{BERT}_{\text{BASE}}$ retains high accuracy on SQuAD 2.0 and the GLUE benchmark tasks, and outperforms state-of-the-art baselines. The deep self-attention distillation can also be applied to compress pre-trained models in larger size. We leave it as our future work.

References

Aguilar, G., Ling, Y., Zhang, Y., Yao, B., Fan, X., and Guo, E. Knowledge distillation from internal representations. *CoRR*, abs/1910.03723, 2019. URL http://arxiv.org/abs/1910.03723.

Ba, L. J., Kiros, J. R., and Hinton, G. E. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Baevski, A., Edunov, S., Liu, Y., Zettlemoyer, L., and Auli, M. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*, 2019.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Piao, S., Gao, J., Zhou, M., and Hon, H.-W. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*, 2020.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., and Giampiccolo, D. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 01 2006.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09*, 2009.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

Chen, Z., Zhang, H., Zhang, X., and Zhao, L. Quora question pairs. 2018.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019. URL http://arxiv.org/abs/1906.04341.

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL http://arxiv.org/abs/1911.02116.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pp. 177–190, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33427-0, 978-3-540-33427-9. doi: 10.1007/11736790_9. URL http://dx.doi.org/10.1007/11736790_9.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Du, X. and Cardie, C. Harvesting paragraph-level question-answer pairs from wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 1907–1917, 2018.

Gehrmann, S., Deng, Y., and Rush, A. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1443.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 1–9, Prague, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W07-1401.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Howard, J. and Ruder, S. 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)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1031.

Hu, M., Peng, Y., Wei, F., Huang, Z., Li, D., Yang, N., and Zhou, M. Attention-guided answer distillation for machine reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2077–2086, 2018. URL https://www.aclweb.org/anthology/D18-1232/.

Jawahar, G., Sagot, B., and Seddah, D. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3651–3657, 2019. URL https://www.aclweb.org/anthology/P19-1356/.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351, 2019. URL http://arxiv.org/abs/1909.10351.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, San Diego, CA, 2015. URL http://arxiv.org/abs/1412.6980.

Lample, G. and Conneau, A. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Levesque, H., Davis, E., and Morgenstern, L. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019a.

Lewis, P. S. H., Oguz, B., Rinott, R., Riedel, S., and Schwenk, H. MLQA: evaluating cross-lingual extractive question answering. *CoRR*, abs/1910.07475, 2019b. URL http://arxiv.org/abs/1910.07475.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Liu, Y. and Lapata, M. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3730–3740, Hong Kong, China, 2019.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Mirzadeh, S., Farajtabar, M., Li, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *CoRR*, abs/1902.03393, 2019. URL http://arxiv.org/abs/1902.03393.

Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://www.aclweb.org/anthology/D18-1206.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N18-1202.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018. URL https://s3-us-west-2.amazonaws.com/openaiassets/research-covers/language-unsupervised/languageunderstandingpaper.pdf.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pp. 784–789, 2018.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6550.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*, 2019.

Sun, S., Cheng, Y., Gan, Z., and Liu, J. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 4322–4331, 2019a.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., and Zhou, D. Mobilebert: Task-agnostic compression of bert by progressive knowledge transfer, 2019b. URL https://openreview.net/pdf?id=SJxjVaNKwB.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016. doi: 10.1109/cvpr.2016.308.

Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., and Lin, J. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136, 2019. URL http://arxiv.org/abs/1903.12136.

Trinh, T. H. and Le, Q. V. A simple method for commonsense reasoning. *ArXiv*, abs/1806.02847, 2018.

Tsai, H., Riesa, J., Johnson, M., Arivazhagan, N., Li, X., and Archer, A. 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 2019, Hong Kong, China, November 3-7, 2019*, pp. 3630–3634, 2019.

Turc, I., Chang, M., Lee, K., and Toutanova, K. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962, 2019a. URL http://arxiv.org/abs/1908.08962.

Turc, I., Chang, M., Lee, K., and Toutanova, K. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962, 2019b. URL http://arxiv.org/abs/1908.08962.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. XLNet: Generalized autoregressive pretraining for language understanding. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=Sks9_ajex.

Zhang, S. and Bansal, M. Addressing semantic drift in question generation for semi-supervised question answering. *CoRR*, abs/1909.06356, 2019.

Zhao, Y., Ni, X., Ding, Y., and Ke, Q. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3901–3910, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1424.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pp. 19–27, 2015.

A. GLUE Benchmark

The summary of datasets used for the General Language Understanding Evaluation (GLUE) benchmark[8] (Wang et al., 2019) is presented in Table 14.

We present the dataset statistics and metrics of SQuAD 2.0[9] (Rajpurkar et al., 2018) in Table 15.

B. Fine-tuning Hyper-parameters

Extractive Question Answering For SQuAD 2.0, the maximum sequence length is 384 and a sliding window of size 128 if the lengths are longer than 384. For the 12-layer model distilled from our in-house pre-trained model, we fine-tune 3 epochs using 48 as the batch size and 4e-5 as the peak learning rate. The rest distilled models are trained using 32 as the batch size and 6e-5 as the peak learning rate for 3 epochs.

GLUE The maximum sequence length is 128 for the GLUE benchmark. We set batch size to 32, choose learning rates from {2e-5, 3e-5, 4e-5, 5e-5} and epochs from {3, 4, 5} for student models distilled from $\text{BERT}_{\text{BASE}}$. For student models distilled from our in-house pre-trained model, the batch size is chosen from {32, 48}. We fine-tune several tasks (CoLA, RTE and MRPC) with longer epochs (up to 10 epochs), which brings slight improvements. For the 12-layer model, the learning rate used for CoLA, RTE and MRPC tasks is 1.5e-5.

[8] https://gluebenchmark.com/

[9] http://stanford-qa.com

Table 14. Summary of the GLUE benchmark.

Corpus	#Train	#Dev	#Test	Metrics
Single-Sentence Tasks				
CoLA	8.5k	1k	1k	Matthews Corr
SST-2	67k	872	1.8k	Accuracy
Similarity and Paraphrase Tasks				
QQP	364k	40k	391k	Accuracy/F1
MRPC	3.7k	408	1.7k	Accuracy/F1
STS-B	7k	1.5k	1.4k	Pearson/Spearman Corr
Inference Tasks				
MNLI	393k	20k	20k	Accuracy
RTE	2.5k	276	3k	Accuracy
QNLI	105k	5.5k	5.5k	Accuracy
WNLI	634	71	146	Accuracy

Table 15. Dataset statistics and metrics of SQuAD 2.0.

#Train	#Dev	#Test	Metrics
130,319	11,873	8,862	Exact Match/F1

C. SQuAD 2.0

Question Generation For the question generation task, we set batch size to 32, and total length to 512. The maximum output length is 48. The learning rates are 3e-5 and 8e-5 for the 12-layer and 6-layer models, respectively. They are both fine-tuned for 25 epochs. We also use label smoothing (Szegedy et al., 2016) with rate of 0.1. During decoding, we use beam search with beam size of 5. The length penalty (Wu et al., 2016) is 1.3.

Abstractive Summarization For the abstractive summarization task, we set batch size to 64, and the rate of label smoothing to 0.1. For the CNN/DailyMail dataset, the total length is 768 and the maximum output length is 160. The learning rates are 1e-4 and 1.5e-4 for the 12-layer and 6-layer models, respectively. They are both fine-tuned for 25 epochs. During decoding, we set beam size to 5, and the length penalty to 0.7. For the XSum dataset, the total length is 512 and the maximum output length is 48. The learning rates are 1e-4 and 1.5e-4 for the 12-layer and 6-layer models, respectively. We fine-tune 30 epochs for the 12-layer model and 50 epochs for the 6-layer model. During decoding, we use beam search with beam size of 5. The length penalty is set to 0.9.

Cross-lingual Natural Language Inference The maximum sequence length is 128 for XNLI. We fine-tune 5 epochs using 128 as the batch size, choose learning rates from {3e-5, 4e-5, 5e-5, 6e-5}.

Cross-lingual Question Answering For MLQA, the maximum sequence length is 512 and a sliding window of size 128 if the lengths are longer than 512. We fine-tune 3 epochs using 32 as the batch size. The learning rates are chosen from {3e-5, 4e-5, 5e-5, 6e-5}.