MINILMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers

Wenhui Wang  Hangbo Bao  Shaohan Huang  Li Dong  Furu Wei*
Microsoft Research
{wenwan,t-habao,shaohan,lidong1,fuwei}@microsoft.com

Abstract

We generalize deep self-attention distillation in MINILM (Wang et al., 2020) by only using self-attention relation distillation for task-agnostic compression of pretrained Transformers. In particular, we define multi-head self-attention relations as scaled dot-product between the pairs of query, key, and value vectors within each self-attention module. Then we employ the above relational knowledge to train the student model. Besides its simplicity and unified principle, more favorably, there is no restriction in terms of the number of student’s attention heads, while most previous work has to guarantee the same head number between teacher and student. Moreover, the fine-grained self-attention relations tend to fully exploit the interaction knowledge learned by Transformer. In addition, we thoroughly examine the layer selection strategy for teacher models, rather than just relying on the last layer as in MINILM. We conduct extensive experiments on compressing both monolingual and multilingual pretrained models. Experimental results demonstrate that our models¹ distilled from base-size and large-size teachers (BERT, RoBERTa and XLM-R) outperform the state-of-the-art.

1 Introduction

Pretrained Transformers (Radford et al., 2018; Devlin et al., 2018; Dong et al., 2019; Yang et al., 2019; Joshi et al., 2019; Liu et al., 2019; Bao et al., 2020; Radford et al., 2019; Raffel et al., 2019; Lewis et al., 2019a) have been highly successful for a wide range of natural language processing tasks. However, these models usually consist of hundreds of millions of parameters and are getting bigger. It brings challenges for fine-tuning and online serving in real-life applications due to the restrictions of computation resources and latency.

Knowledge distillation (KD; Hinton et al. 2015, Romero et al. 2015) has been widely employed to compress pretrained Transformers, which transfers knowledge of the large model (teacher) to the small model (student) by minimizing the differences between teacher and student features. Soft target probabilities (soft labels) and intermediate representations are usually utilized to perform KD training. In this work, we focus on task-agnostic compression of pretrained Transformers (Sanh et al., 2019; Tsai et al., 2019; Jiao et al., 2019; Sun et al., 2019b; Wang et al., 2020). The student models are distilled from large pretrained Transformers using large-scale text corpora. The distilled task-agnostic model can be directly fine-tuned on downstream tasks, and can be utilized to initialize task-specific distillation.

DistilBERT (Sanh et al., 2019) uses soft target probabilities for masked language modeling predictions and embedding outputs to train the student. The student model is initialized from the teacher by taking one layer out of two. Tiny-BERT (Jiao et al., 2019) utilizes hidden states and self-attention distributions (i.e., attention maps and weights), and adopts a uniform function to map student and teacher layers for layer-wise distillation. MobileBERT (Sun et al., 2019b) introduces specially designed teacher and student models using inverted-bottleneck and bottleneck structures to keep their layer number and hidden size the same, layer-wisely transferring hidden states and self-attention distributions. MINILM (Wang et al., 2020) proposes deep self-attention distillation, which uses self-attention distributions and value relations to help the student to deeply mimic teacher’s self-attention modules. MINILM shows that transferring knowledge of teacher’s last layer achieves better performance than layer-wise distil-

¹Distilled models and code will be publicly available at https://aka.ms/minilm.

*Contact person.
We introduce multi-head self-attention relations while performing better than BERT BASE. Afterwards, for teacher and student models with which leads to a restriction that the number of examples. The 2020) by using self-attention relation distillation. Moreover, using a larger number of relation heads brings more fine-grained self-attention knowledge, which helps the student to achieve a deeper mimicry of teacher’s self-attention module. In addition, for large-size (24 layers, 1024 hidden size) teachers, extensive experiments indicate that transferring an upper middle layer tends to perform better than using the last layer as in MINILM.

To summarize, our contributions include:

- We generalize and simplify deep self-attention distillation in MINILM by introducing multi-head self-attention relation distillation, which brings more fine-grained self-attention knowledge and allows more flexibility for the number of student’s attention heads.

- We conduct extensive distillation experiments on different large-size teachers and find that using knowledge of a teacher’s upper middle layer achieves better performance.

- Experimental results demonstrate the effectiveness of our method for different monolingual and multilingual teachers in base-size and large-size.

2 Related Work

2.1 Backbone Network: Transformer

Multi-layer Transformer (Vaswani et al., 2017) has been widely adopted in pretrained models. Each Transformer layer consists of a self-attention sub-layer and a position-wise fully connected feed-forward sub-layer.

Self-Attention Transformer relies on multi-head self-attention to capture dependencies between words. Given previous Transformer layer’s output $H^{l-1} \in \mathbb{R}^{|x| \times d_h}$, the output of a self-attention head $O_{l,a}, a \in [1, A_h]$ is computed via:

$$Q_{l,a} = H^{l-1}W^Q_{l,a} \quad (1)$$
$$K_{l,a} = H^{l-1}W^K_{l,a} \quad (2)$$
$$V_{l,a} = H^{l-1}W^V_{l,a} \quad (3)$$
$$O_{l,a} = \text{softmax}(\frac{Q_{l,a}K_{l,a}^\top}{\sqrt{d_k}})V_{l,a} \quad (4)$$

Previous layer’s output is linearly projected to queries, keys and values using parameter matrices $W^Q_{l,a}, W^K_{l,a}, W^V_{l,a} \in \mathbb{R}^{d_h \times d_h}$, respectively. The self-attention distributions are computed via scaled dot-product of queries and keys. These weights are assigned to the corresponding value vectors to obtain the attention output. $|x|$ represents the length of input sequence. $A_h$ and $d_h$ indicate the number of attention heads and hidden size. $d_k$ is the attention head size. $d_k \times A_h$ is usually equal to $d_h$.

2.2 Pretrained Language Models

Pre-training has led to strong improvements across a variety of natural language processing tasks. Pretrained language models are learned on large amounts of text data, and then fine-tuned to adapt to specific tasks. BERT (Devlin et al., 2018) proposes to pretrain a deep bidirectional Transformer using masked language modeling (MLM) objective. UniLM (Dong et al., 2019) is jointly pretrained on three types language modeling objectives to adapt to both understanding and generation tasks. RoBERTa (Liu et al., 2019) achieves strong performance by training longer steps using large batch size and more text data. MASS (Song et al., 2019), T5 (Raffel et al., 2019) and BART (Lewis et al., 2020) are jointly pretrained using much fewer training examples.
2.3 Knowledge Distillation

Knowledge distillation has been proven to be a promising way to compress large models while maintaining accuracy. Knowledge of a single or an ensemble of large models is used to guide the training of small models. Hinton et al. (2015) propose to use soft target probabilities to train student models. More fine-grained knowledge such as hidden states (Romero et al., 2015) and attention distributions (Zagoruyko and Komodakis, 2017; Hu et al., 2018) are introduced to improve the student model.

In this work, we focus on task-agnostic knowledge distillation of pretrained Transformers. The distilled task-agnostic model can be fine-tuned to adapt to downstream tasks. It can also be utilized to initialize task-specific distillation (Sun et al., 2019a; Turc et al., 2019; Aguilar et al., 2019; Mukherjee and Awadallah, 2020; Xu et al., 2020; Hou et al., 2020; Li et al., 2020), which uses a fine-tuned teacher model to guide the training of the student on specific tasks. Knowledge used for distillation and layer mapping function are two key points for task-agnostic distillation of pretrained Transformers. Most previous work uses soft target probabilities, hidden states, self-attention distributions and value-relation to train the student model. For the layer mapping function, TinyBERT (Jiao et al., 2019) uses a uniform strategy to map teacher and student layers. MobileBERT (Sun et al., 2019b) assumes the student has the same number of layers as its teacher to perform layer-wise distillation. MINILM (Wang et al., 2020) transfers self-attention knowledge of teacher’s last layer to the student last Transformer layer. Different from previous work, our method uses multi-head self-attention relations to eliminate the restriction on the number of student’s attention heads. Moreover, we show that transferring the self-attention knowledge of an upper middle layer of the large-size teacher model is more effective.

3 Multi-Head Self-Attention Relation Distillation

Following MINILM (Wang et al., 2020), the key idea of our approach is to deeply mimic teacher’s self-attention module, which draws dependencies between words and is the vital component of Transformer. MINILM uses teacher’s self-attention distributions to train the student model. It brings...
| Model          | Teacher | #Param | Speedup | SQuAD2 | MNLI-m | QNLI | QQP | RTE | SST | MRPC | CoLA | Avg |
|---------------|---------|--------|---------|--------|--------|------|-----|-----|-----|------|------|-----|
| BERT\textsc{BASE} | -       | 109M   | ×1.0    | 76.8   | 84.5   | 91.7 | 91.3 | 68.6 | 93.2 | 87.3 | 58.9 | 81.5 |
| RoBERTa\textsc{BASE} | -       | 125M   | ×1.0    | 83.7   | 87.6   | 92.8 | 91.9 | 78.7 | 94.8 | 90.2 | 63.6 | 85.4 |
| BERT\textsc{SMALL} | -       | 66M    | ×2.0    | 73.2   | 81.8   | 89.8 | 90.6 | 67.9 | 91.2 | 84.9 | 53.5 | 79.1 |
| Truncated BERT\textsc{BASE} | -       | 66M    | ×2.0    | 69.9   | 81.2   | 87.9 | 90.4 | 65.5 | 90.8 | 82.7 | 41.4 | 76.2 |
| Truncated RoBERTa\textsc{BASE} | -       | 81M    | ×2.0    | 77.9   | 84.9   | 91.1 | 91.3 | 67.9 | 92.9 | 87.5 | 55.2 | 81.1 |
| DistilBERT    | BERT\textsc{BASE} | 66M | ×2.0 | 70.7 | 82.2 | 89.2 | 88.5 | 59.9 | 91.3 | 87.5 | 51.3 | 77.6 |
| TinyBERT      | BERT\textsc{BASE} | 66M | ×2.0 | 73.1 | 83.5 | 90.5 | 90.6 | 72.2 | 91.6 | 88.4 | 42.8 | 79.1 |
| MiniLM        | BERT\textsc{BASE} | 66M | ×2.0 | 76.4 | 84.0 | 91.0 | 91.0 | 71.5 | 92.0 | 88.4 | 42.9 | 80.4 |
| 6×384 Ours    | BERT\textsc{BASE} | 22M   | ×5.3   | 72.9   | 82.8   | 90.3 | 90.6 | 68.9 | 91.3 | 86.6 | 41.8 | 78.2 |
| 6×384 Ours    | BERT\textsc{LARGE} | 22M   | ×5.3   | 74.3   | 83.0   | 90.4 | 90.7 | 68.5 | 91.1 | 87.8 | 41.6 | 78.4 |
| 6×384 Ours    | RoBERTa\textsc{BASE} | 30M   | ×5.3   | 76.4   | 84.4   | 90.9 | 90.8 | 69.9 | 92.0 | 88.7 | 42.6 | 79.5 |
| 6×768 Ours    | BERT\textsc{BASE} | 66M   | ×2.0   | 76.3   | 84.2   | 90.8 | 91.1 | 72.1 | 92.4 | 88.9 | 52.5 | 81.0 |
| 6×768 Ours    | BERT\textsc{LARGE} | 66M   | ×2.0   | 77.7   | 85.0   | 91.4 | 91.1 | 73.0 | 92.5 | 88.9 | 53.9 | 81.7 |
| 6×768 Ours    | RoBERTa\textsc{LARGE} | 81M   | ×2.0   | 81.6   | 87.0   | 92.7 | 91.4 | 78.7 | 94.5 | 90.4 | 54.0 | 83.8 |

Table 1: Results of our students distilled from base-size and large-size teachers on the development sets of GLUE and SQuAD 2.0. We report F1 for SQuAD 2.0, Matthews correlation coefficient for CoLA, and accuracy for other datasets. The GLUE results of DistilBERT are taken from Sanh et al. (2019). The rest results of DistilBERT, TinyBERT\textsuperscript{1,2}, BERT\textsc{SMALL}, Truncated BERT\textsc{BASE} and MiniLM are taken from Wang et al. (2020). BERT\textsc{SMALL} (Ture et al., 2019) is trained using the MLM objective, without using KD. We also report the results of truncated BERT\textsc{BASE} and truncated RoBERTa\textsc{BASE}, which drops the top 6 layers of the base model. Top-layer dropping has been proven to be a strong baseline (Sajjad et al., 2020). The fine-tuning results are an average of 4 runs.

3.1 Multi-Head Self-Attention Relations

Multi-head self-attention relations are obtained by scaled dot-product of pairs\textsuperscript{3} of queries, keys and values of multiple relation heads. Taking query vectors as an example, in order to obtain queries of multiple relation heads, we first concatenate queries of different attention heads and then split the concatenated vector based on the desired number of relation heads. The same operation is also performed on keys and values. For teacher and student models which uses different number of attention heads, we convert their queries, keys and values of different number of attention heads into vectors of the same number of relation heads to perform KD training. Our method eliminates the restriction on the number of attention heads of student models. Moreover, using more relation heads in computing self-attention relations brings more fine-grained self-attention knowledge and improves the performance of the student model.

We use A\textsubscript{1}, A\textsubscript{2}, A\textsubscript{3} to denote the queries, keys and values of multiple relation heads. The KL-divergence between multi-head self-attention relations of the teacher and student is used as the training objective:

\[
\mathcal{L} = \sum_{i=1}^{3} \sum_{j=1}^{3} \alpha_{ij} \mathcal{L}_{ij} \tag{5}
\]

\[
\mathcal{L}_{ij} = \frac{1}{A_r} \sum_{a=1}^{A_r} \sum_{t=1}^{x} D_{KL}(R_{ij,l,a,t}^T \parallel R_{ij,m,a,t}^S) \tag{6}
\]

\[
R_{ij,l,a}^T = \text{softmax}(\frac{A_{T_{l,i,a}} A_{T_{j,l,a}}^\top}{\sqrt{d_r}}) \tag{7}
\]

\[
R_{ij,m,a}^S = \text{softmax}(\frac{A_{S_{i,m,a}} A_{S_{j,m,a}}^\top}{\sqrt{d_r}}) \tag{8}
\]

where A\textsubscript{T_{l,i,a}} \in \mathbb{R}^{x \times d_r} and A\textsubscript{S_{i,m,a}} \in \mathbb{R}^{x \times d_r} (i \in [1, 3]) are the queries, keys and values of a relation head of l-th teacher layer and m-th student layer. d\textsubscript{r} and d\textsubscript{r'} are the relation head size of teacher and student models. R\textsubscript{ij,l,a}^T \in \mathbb{R}^{A_r \times |x| \times |x|} is the self-attention relation of A\textsubscript{T_{i,l}} and A\textsubscript{T_{j,l}} of teacher model.

\textsuperscript{1}In addition to task-agnostic distillation, TinyBERT uses task-specific distillation and data augmentation to further improve the model. We report the fine-tuning results of their public task-agnostic model.

\textsuperscript{2}There are nine types of self-attention relations, such as query-query, key-key, key-value and query-value relations.
We use the uncased version for three BERT teacher models. For the pre-training data, we use English Wikipedia and BookCorpus (Zhu et al., 2015). We train student models using 256 as the batch size and 6e-4 as the peak learning rate for 400,000 steps. We use linear warmup over the first 4,000 steps and linear decay. We use Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The maximum sequence length is set to 512. The dropout rate and weight decay are 0.1 and 0.01. The number of attention heads is 12 for all student models. The number of relation heads is 48 and 64 for base-size and large-size teacher model, respectively. The student models are initialized randomly.

For models distilled from RoBERTa, we use similar pre-training datasets as in Liu et al. (2019). For the 12 × 768 student model, we use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The rest hyper-parameters are the same as models distilled from BERT.

For multilingual student models distilled from XLM-R, we perform training using the same datasets as in Conneau et al. (2019) for 1,000,000 steps. We conduct distillation experiments using 8 V100 GPUs with mixed precision training.

### 4.2 Downstream Tasks

Following previous pre-training (Devlin et al., 2018; Liu et al., 2019; Conneau et al., 2019) and task-agnostic distillation (Sun et al., 2019b; Jiao et al., 2019) work, we evaluate the English student models on GLUE benchmark and extractive question answering. The multilingual models are evaluated on cross-lingual natural language inference and cross-lingual question answering.

#### GLUE

General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) consists of two single-sentence classification tasks (SST-2 (Socher et al., 2013) and CoLA (Warstadt et al., 2018)), three similarity and paraphrase tasks (MRPC (Dolan and Brockett, 2005), STS-B (Cer et al., 2017) and QQP), and four inference tasks (MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli

### Table 2: Results of our 6×768 students distilled form BERT on GLUE test sets and SQuAD 2.0 dev set. The reported results are directly fine-tuned on downstream tasks. We report F1 for SQuAD 2.0, QQP and MRPC, Spearman correlation for STS-B, Matthews correlation coefficient for CoLA and accuracy for the rest.

| Model       | Teacher | #Param | Speedup | SQuAD2 | MNLI-m/mm | QNLI | QQP | RTE | SST | MRPC | CoLA | STS | Avg |
|-------------|---------|--------|---------|--------|-----------|------|-----|-----|-----|------|------|-----|-----|
| BERT_BASE   | -       | 198M   | 1.0     | 76.8   | 84.6/83.4 | 90.5 | 71.2| 66.4 | 93.5 | 89.9 | 52.1 | 85.8 | 17.93 |
| BERT_LARGE  | -       | 340M   | 0.3     | 81.9   | 86.7/85.9 | 92.7 | 72.1| 70.1 | 94.9 | 89.3 | 60.5 | 86.5 | 18.1 |
| 6×768 Ours  | BERT_BASE| 66M    | 2.0     | 76.3   | 83.8/83.3 | 90.2 | 70.9| 69.2 | 92.9 | 89.1 | 46.6 | 84.3 | 17.8 |
| 6×768 Ours  | BERT_LARGE| 66M   | 2.0     | 77.7   | 84.5/84.0 | 91.5 | 71.3| 69.2 | 93.0 | 89.1 | 48.6 | 85.1 | 17.94 |

$R_{ij,l,a}^T \in \mathbb{R}^{[x] \times [x]}$ is the self-attention relation of a teacher’s relation head. $R_{ij,m}^S \in \mathbb{R}^{A_r \times [x] \times [x]}$ is the self-attention relation of student model. For example, $R_{ij,l}^T$ represents teacher’s Q-Q attention relation in Figure 1. $A_r$ is the number of relation heads. If the number of relation heads and attention heads is the same, the Q-K relation is equivalent to the attention weights in self-attention module. $\alpha_{ij} \in \{0, 1\}$ is the weight assigned to each self-attention relation loss. We transfer query-query, key-key and value-value self-attention relations to balance the performance and training cost.

### 3.2 Layer Selection of Teacher Model

Besides the knowledge used for distillation, mapping function between teacher and student layers is another key factor. As in MINILM, we only transfer the self-attention knowledge of one of the teacher layers to the student last layer. Only distilling one layer of the teacher model is fast and effective. Different from previous work which usually conducts experiments on base-size teachers, we experiment with different large-size teachers and find that transferring self-attention knowledge of an upper middle layer performs better than using other layers. For BERTLARGE and BERTLARGE-WWM, transferring the 21-th (start at one) layer achieves the best performance. For RoBERTalARGE and XLM-RlARGE, using the self-attention knowledge of 19-th layer achieves better performance. For the base-size teacher, we also find that using teacher’s last layer performs better.
Table 3: Results of our $12 \times 768$ models on the dev sets of GLUE benchmark and SQuAD 2.0. The fine-tuning results are an average of 4 runs for each task. We report F1 for SQuAD 2.0, Pearson correlation for STS-B, Matthews correlation coefficient for CoLA and accuracy for the rest.

| Model                  | Teacher       | #Param | SQuAD2 | MNLI | QNLI | QQP | RTE | SST | MRPC | CoLA | STS | Avg |
|------------------------|---------------|--------|--------|------|------|-----|-----|-----|------|------|-----|-----|
| BERT BASE              | -             | 109M   | 76.8   | 84.5 | 91.7 | 91.3 | 68.6 | 93.2 | 87.3 | 58.9 | 89.5 | 82.4 |
| RoBERTa BASE           | -             | 125M   | 83.7   | 87.6 | 92.8 | 91.9 | 78.7 | 94.8 | 90.2 | 63.6 | 91.2 | 86.1 |

12 × 768 Ours          BERT LARGE   | 109M   | 81.8   | 86.5   | 92.6  | 91.6 | 76.4 | 93.3 | 89.2 | 62.3 | 90.5 | 84.9 |
| 12 × 768 Ours          RoBERTa LARGE | 125M   | 86.6   | 89.4   | 94.0  | 91.8 | 83.1 | 95.9 | 91.2 | 65.0 | 91.3 | 87.6 |

Table 4: Comparison of different methods using BERT LARGE WWM as the teacher. We report dev results of $12 \times 384$ student model with 128 embedding size.

| Model                  | SQuAD2 | MNLI-m | SST-2 | Avg |
|------------------------|--------|--------|-------|-----|
| MINILM (Last Layer)    | 79.1   | 84.7   | 91.2  | 85.0 |
| + Upper Middle Layer   | 80.3   | 85.2   | 91.5  | 85.7 |
| 12 × 384 Ours          | 80.7   | 85.7   | 92.3  | 86.2 |

4.3 Main Results

Table 1 presents the dev results of $6 \times 384$ and $6 \times 768$ models distilled from BERT BASE, BERT LARGE and RoBERTa LARGE on GLUE and SQuAD 2.0. (1) Previous methods (Sanh et al., 2019; Jiao et al., 2019; Sun et al., 2019a; Wang et al., 2020) usually distill BERT BASE into a 6-layer model with 768 hidden size. We first report results of the same setting. Our $6 \times 768$ model outperforms DistilBERT, TinyBERT, MINILM and two BERT baselines across most tasks. Moreover, our method allows more flexibility for the number of attention heads of student models. (2) Both $6 \times 384$ and $6 \times 768$ models distilled from BERT LARGE outperform models distilled from BERT BASE. The $6 \times 768$ model distilled from BERT LARGE is 2.0× faster than BERT BASE, while achieving better performance. (3) Student models distilled from RoBERTa LARGE achieve further improvements. Better teacher results in better students. Multi-head self-attention relation distillation is effective for different large-size pretrained Transformers.

We report the results of $6 \times 768$ students distilled from BERT BASE and BERT LARGE on GLUE test sets and SQuAD 2.0 dev set in Table 2. $6 \times 768$ model distilled from BERT BASE retains more than 99% accuracy of its teacher while using 50% Transformer parameters. $6 \times 768$ model distilled from BERT LARGE compares favorably with BERT BASE.

We compress RoBERTa LARGE and BERT LARGE into a base-size student model. Dev results of GLUE benchmark and SQuAD 2.0 are shown in Table 3. Our base-size models distilled from large-size teacher outperforms BERT BASE and RoBERTa BASE. Our method can be adopted to train students in different parameter size. Moreover, our student distilled from RoBERTa LARGE uses a much smaller (almost $32 \times$ smaller) training batch size and fewer training steps than RoBERTa BASE. Our method requires much fewer training examples.

Most of previous work conducts experiments using base-size teachers. To compare with previous methods on large-size teacher, we reimplement MINILM and compress BERT LARGE WWM into a $12 \times 384$ student model. Dev results of SQuAD 2.0, MNLI-m and SST-2 are presented in Table 4. Our method also outperforms MINILM for large-size teachers. Moreover, we report results of distilling an upper middle layer instead of the last layer for MINILM. Layer selection is also effective for MINILM when distilling large-size teachers.

Table 5 and Table 6 show the results of our student models distilled from XLM-R on XNLI and MLQA. For XNLI, the best single model is selected on the joint dev set of all the languages as in Conneau et al. (2019). Following Lewis et al. (2019b), we adopt SQuAD 1.1 as training data and evaluate on MLQA English development set for
Table 5: Cross-lingual classification results of our multilingual models on XNLI. We report the accuracy on each of the 15 XNLI languages and the average accuracy. #L and #H indicate the number of layers and hidden size.

Table 6: Cross-lingual question answering results of our multilingual models on MLQA. We report the F1 and EM (exact match) scores on each of the 7 MLQA languages. #L and #H indicate the number of layers and hidden size. † indicates our fine-tuned results of XLM-R_{BASE}.

Table 7: Ablation studies of different self-attention relations. We report results of 6×384 student model distilled from BERT_{BASE}. The relation head number is 12.

4.4 Ablation Studies

Effect of using different self-attention relations

We perform ablation studies to analyse the contribution of different self-attention relations. Dev results of three tasks are illustrated in Table 7. Q-Q, K-K and V-V self-attention relations positively contribute to the final results. Besides, we also compare Q-Q + K-K + V-V with Q-K + V-V given queries and keys are employed to compute self-attention distributions in self-attention module. Experimental result shows that using Q-Q + K-K + V-V achieves better performance.

Effect of distilling different teacher layers

Figure 2 presents the results of 6×384 model distilled from different layers of BERT_{BASE}, BERT_{LARGE} and XLM-R_{LARGE}. For BERT_{BASE}, using the last layer achieves better performance than other layers. For BERT_{LARGE} and XLM-R_{LARGE}, we find that using one of the upper middle layers achieves the best performance. The same trend is also observed for BERT_{LARGE-WWM} and RoBERTa_{LARGE}.

Effect of different number of relation heads

Table 8 shows the results of 6×384 model distilled from BERT_{BASE} and RoBERTa_{BASE} using different number of relation heads. Using a larger number of relation heads achieves better performance. More
Table 9: Comparison between MobileBERT and the same-size model (12 layers, 384 hidden size and 128 embedding size) distilled form BERT\textsubscript{LARGE} (Whole Word Masking) on GLUE test sets and SQuAD 2.0 dev set. Following MobileBERT (Sun et al., 2019b), the reported results are directly fine-tuned on downstream tasks. We compute the speedup of MobileBERT according to their reported latency.

| Model          | Teacher          | #Param | Speedup | SQuAD2 | MNLI-m/mm | QNLI | QQP | RTE | SST | MRPC | CoLA | STS | Avg |
|----------------|------------------|--------|---------|--------|-----------|------|-----|-----|-----|------|------|-----|-----|
| BERT\textsubscript{BASE} | - | - | 109M | 1.0 \times | 76.8 | 84.6/83.4 | 90.5 | 71.2 | 66.4 | 93.5 | 88.9 | 52.1 | 85.8 | 79.3 |
| MobileBERT     | IB-BERT\textsubscript{LARGE} | 25M | 1.5 \times | 80.2 | 84.3/83.4 | 91.6 | 70.5 | 70.4 | 92.6 | 88.8 | 51.1 | 84.8 | 79.8 |
| MobileBERT \* More Att-Rels | BERT\textsubscript{LARGE}-WWM | 25M | 2.7 \times | 80.9 | 85.8/84.8 | 92.3 | 71.6 | 72.0 | 93.6 | 89.7 | 46.6 | 86.0 | 80.3 |
| BERT\textsubscript{BASE} | - | - | 25M | 2.7 \times | 80.9 | 85.8/84.8 | 92.3 | 71.6 | 72.0 | 93.6 | 89.7 | 46.6 | 86.0 | 80.3 |

MobileBERT (Sun et al., 2019b) compresses a specially designed teacher model (in the BERT\textsubscript{LARGE} size) with inverted bottleneck modules into a 24-layer student using the bottleneck modules. Since our goal is to compress different large models (e.g. BERT and RoBERTa) to small models using standard Transformer architecture, we note that our student model can not directly compare with MobileBERT. We provide results of a student model with the same parameter size for a reference. A public large-size model (BERT\textsubscript{LARGE-WWM}) is used as the teacher, which achieves similar performance as MobileBERT’s teacher. We distill BERT\textsubscript{LARGE-WWM} into a student model (25M parameters) using the same training data (i.e., English Wikipedia and BookCorpus). The test results of GLUE and dev result of SQuAD 2.0 are illustrated in Table 9. Our model outperforms MobileBERT across most tasks with a faster inference speed. Moreover, our method can be applied for different teachers and has much fewer restriction of students.

Figure 2: $6 \times 384$ models trained using different BERT\textsubscript{BASE} (a), BERT\textsubscript{LARGE} (b) and XLM-R\textsubscript{LARGE} (c) layers.

Table 10: Results of introducing more self-attention relations (Q-K, Q-Q, K-V, V-Q, K-V and V-K relations).

| Model          | Teacher          | SQuAD2 | MNLI-m | SST-2 |
|----------------|------------------|--------|--------|-------|
| $6 \times 384$ Ours | BERT\textsubscript{BASE} | 72.9 | 82.8 | 91.3 |
| \* More Att-Rels | BERT\textsubscript{BASE} | 73.3 | 82.8 | 91.6 |
| $6 \times 384$ Ours | BERT\textsubscript{LARGE} | 74.3 | 83.0 | 91.1 |
| \* More Att-Rels | BERT\textsubscript{LARGE} | 74.7 | 83.2 | 92.4 |
| $6 \times 384$ Ours | RoBERT\textsubscript{LARGE} | 76.4 | 84.4 | 92.0 |
| \* More Att-Rels | RoBERT\textsubscript{LARGE} | 76.0 | 84.4 | 92.1 |
| $6 \times 768$ Ours | BERT\textsubscript{BASE} | 76.3 | 84.2 | 92.4 |
| \* More Att-Rels | BERT\textsubscript{BASE} | 76.8 | 84.4 | 92.3 |
| $6 \times 768$ Ours | BERT\textsubscript{LARGE} | 77.7 | 85.0 | 92.5 |
| \* More Att-Rels | BERT\textsubscript{LARGE} | 78.1 | 85.2 | 92.5 |
| $6 \times 768$ Ours | RoBERT\textsubscript{LARGE} | 81.6 | 87.0 | 94.5 |
| \* More Att-Rels | RoBERT\textsubscript{LARGE} | 81.2 | 87.3 | 94.1 |

We also observe that our model performs relatively worse on CoLA compared with MobileBERT. The task of CoLA is to evaluate the grammatical acceptability of a sentence. It requires more fine-grained linguistic knowledge that can be learnt from language modeling objectives. Fine-tuning the model using the MLM objective as in MobileBERT brings improvements for CoLA. However, our preliminary experiments show that this strategy will lead to slight drop for other GLUE tasks.

Fine-grained self-attention knowledge can be captured by using more relation heads, which helps the student to deeply mimic the self-attention module of its teacher. Besides, we find that the number of relation heads is not required to be a positive multiple of both the number of student and teacher attention heads. The relation head can be a fragment of a single attention head or contains fragments from multiple attention heads.

5 Discussion

5.1 Comparison with MobileBERT

MobileBERT (Sun et al., 2019b) compresses a specially designed teacher model (in the BERT\textsubscript{LARGE} size) with inverted bottleneck modules into a 24-layer student using the bottleneck modules. Since our goal is to compress different large models (e.g. BERT and RoBERTa) to small models using standard Transformer architecture, we note that our student model can not directly compare with MobileBERT. We provide results of a student model with the same parameter size for a reference. A public large-size model (BERT\textsubscript{LARGE-WWM}) is used as the teacher, which achieves similar performance as MobileBERT’s teacher. We distill BERT\textsubscript{LARGE-WWM} into a student model (25M parameters) using the same training data (i.e., English Wikipedia and BookCorpus). The test results of GLUE and dev result of SQuAD 2.0 are illustrated in Table 9. Our model outperforms MobileBERT across most tasks with a faster inference speed. Moreover, our method can be applied for different teachers and has much fewer restriction of students.

We also observe that our model performs relatively worse on CoLA compared with MobileBERT. The task of CoLA is to evaluate the grammatical acceptability of a sentence. It requires more fine-grained linguistic knowledge that can be learnt from language modeling objectives. Fine-tuning the model using the MLM objective as in MobileBERT brings improvements for CoLA. However, our preliminary experiments show that this strategy will lead to slight drop for other GLUE tasks.
5.2 Results of More Self-Attention Relations

In Table 9 and 10, we report results of students trained using more self-attention relations (Q-K, K-Q, Q-V, V-Q, K-V and V-K relations). We observe improvements across most tasks, especially for student models distilled from BERT. Fine-grained self-attention knowledge in more attention relations improves our students. However, introducing more self-attention relations also brings a higher computational cost. In order to achieve a balance between performance and computational cost, we choose to transfer Q-Q, K-K and V-V self-attention relations instead of all self-attention relations in this work.

6 Conclusion

We generalize deep self-attention distillation in MINILM by employing multi-head self-attention relations to train the student. Our method introduces more fine-grained self-attention knowledge and eliminates the restriction of the number of student’s attention heads. Moreover, we show that transferring the self-attention knowledge of an upper middle layer achieves better performance for large-size teachers. Our monolingual and multilingual models distilled from BERT, RoBERTa and XLM-R obtain competitive performance and outperform state-of-the-art methods. For future work, we are exploring an automatic layer selection algorithm. We also would like to apply our method to larger pretrained Transformers.

References

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Edward Guo. 2019. Knowledge distillation from internal representations. CoRR, abs/1910.03723.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. arXiv preprint arXiv:2002.12804.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second PASCAL recognising textual entailment challenge. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In In Proc Text Analysis Conference (TAC’09).

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xianling Mao, and Heyan Huang. 2019. Cross-lingual natural language generation via pre-training. CoRR, abs/1909.10481.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Sksam Singhal, Wenhui Wang, Xia Song, Xianling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. CoRR, abs/2007.07834.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. CoRR, abs/1911.02116.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. 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, pages 177–190, Berlin, Heidelberg. Springer-Verlag.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In 33rd Conference on Neural Information Processing Systems (NeurIPS 2019).

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 1–9, Prague. Association for Computational Linguistics.
Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Minghao Hu, Yuxing Peng, Furu Wei, Zhen Huang, Dongsheng Li, Nan Yang, and Ming Zhou. 2018. 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, pages 2077–2086.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling BERT for natural language understanding. CoRR, abs/1909.10351.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. arXiv preprint arXiv:1907.10529.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, San Diego, CA.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. Advances in Neural Information Processing Systems (NeurIPS).

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019b. MLQA: evaluating cross-lingual extractive question answering. CoRR, abs/1910.07475.

Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruiyong Xu, Min Yang, and Yaohong Jin. 2020. BERT-EMD: many-to-many layer mapping for BERT compression with earth mover’s distance. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 3009–3018. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Subhajatra Mukherjee and Ahmed Hassan Awadallah. 2020. Xtremedistil: Multi-stage distillation for massive multilingual models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 2221–2234. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. 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.

Hassan Sajjad, Fahim Dalvi, Nadir Durairani, and Pleslav Nakov. 2020. Poor man’s BERT: smaller, faster, cheaper and lighter. CoRR, abs/2004.03844.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR, abs/1910.01108.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642.
Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019a. 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*, pages 4322–4331.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2019b. Mobile-bert: Task-agnostic compression of bert by progressive knowledge transfer.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical BERT models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3630–3634.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. *Attention is all you need*. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

The summary of datasets used for the General Language Understanding Evaluation (GLUE) benchmark is presented in Table 11.

### Table 11: 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 12: Dataset statistics and metrics of SQuAD 2.0.

| #Train | #Dev | #Test | Metrics |
|--------|------|-------|---------|
| 130,319 | 11,873 | 8,862 | Exact Match/F1 |
B SQuAD 2.0

We present the dataset statistics and metrics of SQuAD 2.0\(^5\) (Rajpurkar et al., 2018) in Table 12.

C Hyper-parameters for Fine-tuning

**Extractive Question Answering** For SQuAD 2.0, the maximum sequence length is 384. The batch size is set to 32. We choose learning rates from \{3e-5, 6e-5, 8e-5, 9e-5\} and fine-tune the model for 3 epochs. The warmup ratio and weight decay is 0.1 and 0.01.

**GLUE** The maximum sequence length is 128 for the GLUE benchmark. We set batch size to 32, choose learning rates from \{1e-5, 1.5e-5, 2e-5, 3e-5, 5e-5\} and epochs from \{3, 5, 10\} for different student models. We fine-tune CoLA task with longer training steps (25 epochs). The warmup ratio and weight decay is 0.1 and 0.01.

**Cross-lingual Natural Language Inference (XNLI)** The maximum sequence length is 256 for XNLI. We fine-tune 10 epochs using 64 as the batch size. The learning rates are chosen from \{3e-5, 4e-5, 5e-5, 6e-5\}.

**Cross-lingual Question Answering** For MLQA, the maximum sequence length is 512. We fine-tune 4 epochs using 32 as the batch size. The learning rates are chosen from \{3e-5, 4e-5, 5e-5, 6e-5\}.

\(^5\)http://stanford-qa.com