TinyBERT: Distilling BERT for Natural Language Understanding

2022.01.09

주세준

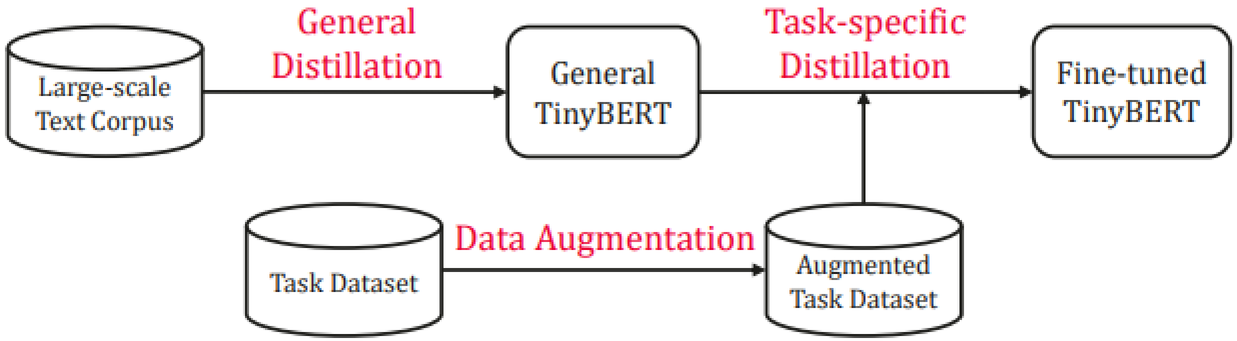

Figure 1: The illustration of TinyBERT learning.

$$\mathcal{L}_{\text{KD}} = \sum_{x \in \mathcal{X}} L\big(f^S(x), f^T(x)\big),$$

$$\mathcal{L}_{\text{model}} = \sum_{x \in \mathcal{X}} \sum_{m=0}^{M+1} \lambda_m \mathcal{L}_{\text{layer}}(f_m^S(x), f_{g(m)}^T(x)),$$

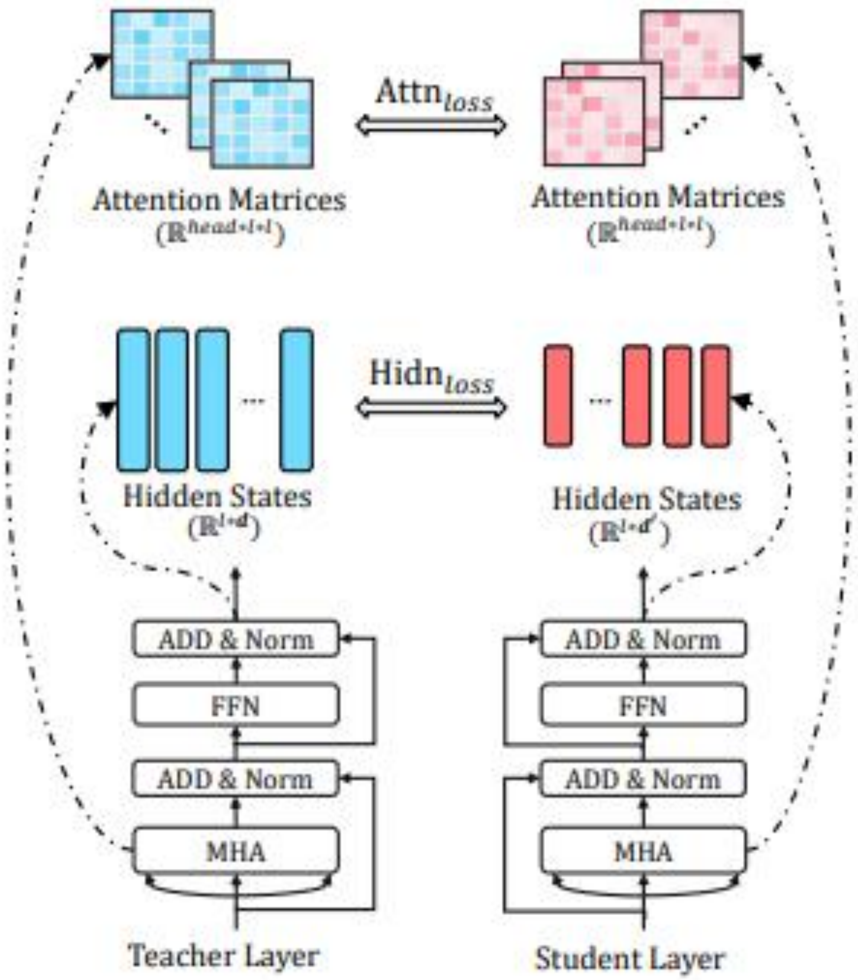

Figure 2: The details of Transformer-layer distillation consisting of Attn$_{loss}$(attention based distillation) and Hidn$_{loss}$(hidden states based distillation).

Transformer Layer Distillation

$$\mathcal{L}_{\text{attn}} = \frac{1}{h}\sum\nolimits_{i=1}^{h} \texttt{MSE}(\boldsymbol{A}_i^S, \boldsymbol{A}_i^T),$$

$$\mathcal{L}_{\text{hidn}} = \texttt{MSE}(\boldsymbol{H}^S \boldsymbol{W}_h, \boldsymbol{H}^T),$$

Embedding-layer Distillation

$$\mathcal{L}_{\text{embd}} = \texttt{MSE}(\boldsymbol{E}^S \boldsymbol{W}_e, \boldsymbol{E}^T),$$

Prediction-layer Distillation

$$\mathcal{L}_{\text{pred}} = \text{CE}(\boldsymbol{z}^T/t, \boldsymbol{z}^S/t),$$

Final Loss

$$\mathcal{L}_{\text{model}} = \sum_{x \in \mathcal{X}} \sum_{m=0}^{M+1} \lambda_m \mathcal{L}_{\text{layer}}(f_m^S(x), f_{g(m)}^T(x)),$$

$$\mathcal{L}_{\text{layer}} = \begin{cases} \mathcal{L}_{\text{embd}}, & m=0 \\ \mathcal{L}_{\text{hidn}} + \mathcal{L}_{\text{attn}}, & M \geq m > 0 \\ \mathcal{L}_{\text{pred}}, & m = M+1 \end{cases}$$

General Distillation

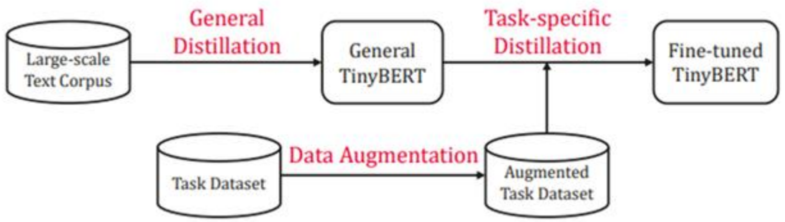

Figure 1: The illustration of TinyBERT learning.

- Teacher: PreTrained Bert
- Prediction-layer distillation (x)

Task-specific Distillation

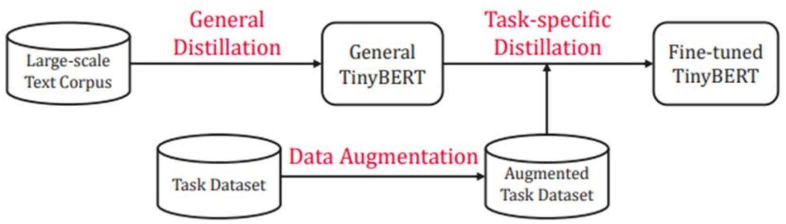

Figure 1: The illustration of TinyBERT learning.

- 1. Data Augmentation

Algorithm 1 Data Augmentation Procedure for Task-specific Distillation

Input: $\mathbf{x}$ is a sequence of words
Params: p_t: the threshold probability
N_a: the number of samples augmented per example
K: the size of candidate set
Output: D': the augmented data

$n \leftarrow 0$; $D' \leftarrow [\,]$
while $n < N_a$ **do**
  $\mathbf{x}_m \leftarrow \mathbf{x}$
  for $i \leftarrow 1$ to len($\mathbf{x}$) **do**
    if $\mathbf{x}[i]$ is a single-piece word **then**
      Replace $\mathbf{x}_m[i]$ with [MASK]
      $C \leftarrow K$ most probable words of BERT$(\mathbf{x}_m)[i]$
    else
      $C \leftarrow K$ most similar words of $\mathbf{x}[i]$ from GloVe
    end if
    Sample $p \sim$ Uniform(0, 1)
    if $p \leq p_t$ **then**
      Replace $\mathbf{x}_m[i]$ with a word in C randomly
    end if
  end for
  Append $\mathbf{x}_m$ to D'
  $n \leftarrow n + 1$
end while
return D'

- 2. Task Specific distillation

System	#Params	#FLOPs	Speedup	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Avg
$\mathrm{BERT}_{\mathrm{BASE}}$ (Teacher)	109M	22.5B	1.0x	83.9/83.4	71.1	90.9	93.4	52.8	85.2	87.5	67.0	79.5
$\mathrm{BERT}_{\mathrm{TINY}}$	14.5M	1.2B	9.4x	75.4/74.9	66.5	84.8	87.6	19.5	77.1	83.2	62.6	70.2
$\mathrm{BERT}_{\mathrm{SMALL}}$	29.2M	3.4B	5.7x	77.6/77.0	68.1	86.4	89.7	27.8	77.0	83.4	61.8	72.1
BERT_4-PKD	52.2M	7.6B	3.0x	79.9/79.3	70.2	85.1	89.4	24.8	79.8	82.6	62.3	72.6
$\mathrm{DistilBERT}_4$	52.2M	7.6B	3.0x	78.9/78.0	68.5	85.2	91.4	32.8	76.1	82.4	54.1	71.9
$\mathrm{MobileBERT}_{\mathrm{TINY}}$†	15.1M	3.1B	-	81.5/81.6	68.9	**89.5**	91.7	**46.7**	80.1	**87.9**	65.1	**77.0**
$\mathrm{TinyBERT}_4$ (ours)	14.5M	1.2B	9.4x	**82.5/81.8**	**71.3**	87.7	**92.6**	44.1	**80.4**	86.4	**66.6**	**77.0**
BERT_6-PKD	67.0M	11.3B	2.0x	81.5/81.0	70.7	89.0	92.0	-	-	85.0	65.5	-
PD	67.0M	11.3B	2.0x	82.8/82.2	70.4	88.9	91.8	-	-	86.8	65.3	-
$\mathrm{DistilBERT}_6$	67.0M	11.3B	2.0x	82.6/81.3	70.1	88.9	92.5	49.0	81.3	86.9	58.4	76.8
$\mathrm{TinyBERT}_6$ (ours)	67.0M	11.3B	2.0x	**84.6/83.2**	**71.6**	**90.4**	**93.1**	**51.1**	**83.7**	**87.3**	**70.0**	**79.4**

Table 1: Results are evaluated on the test set of GLUE official benchmark. The best results for each group of student models are in-bold. The architecture of $\mathrm{TinyBERT}_4$ and $\mathrm{BERT}_{\mathrm{TINY}}$ is (M=4, d=312, d_i=1200), $\mathrm{BERT}_{\mathrm{SMALL}}$ is (M=4, d=512, d_i=2048), BERT_4-PKD and $\mathrm{DistilBERT}_4$ is (M=4, d=768, d_i=3072) and the architecture of BERT_6-PKD, $\mathrm{DistilBERT}_6$ and $\mathrm{TinyBERT}_6$ is (M=6, d=768, d_i=3072). All models are learned in a single-task manner. The inference speedup is evaluated on a single NVIDIA K80 GPU. † denotes that the comparison between $\mathrm{MobileBERT}_{\mathrm{TINY}}$ and $\mathrm{TinyBERT}_4$ may not be fair since the former has 24 layers and is task-agnosticly distilled from IB-$\mathrm{BERT}_{\mathrm{LARGE}}$ while the later is a 4-layers model task-specifically distilled from $\mathrm{BERT}_{\mathrm{BASE}}$.

감사합니다