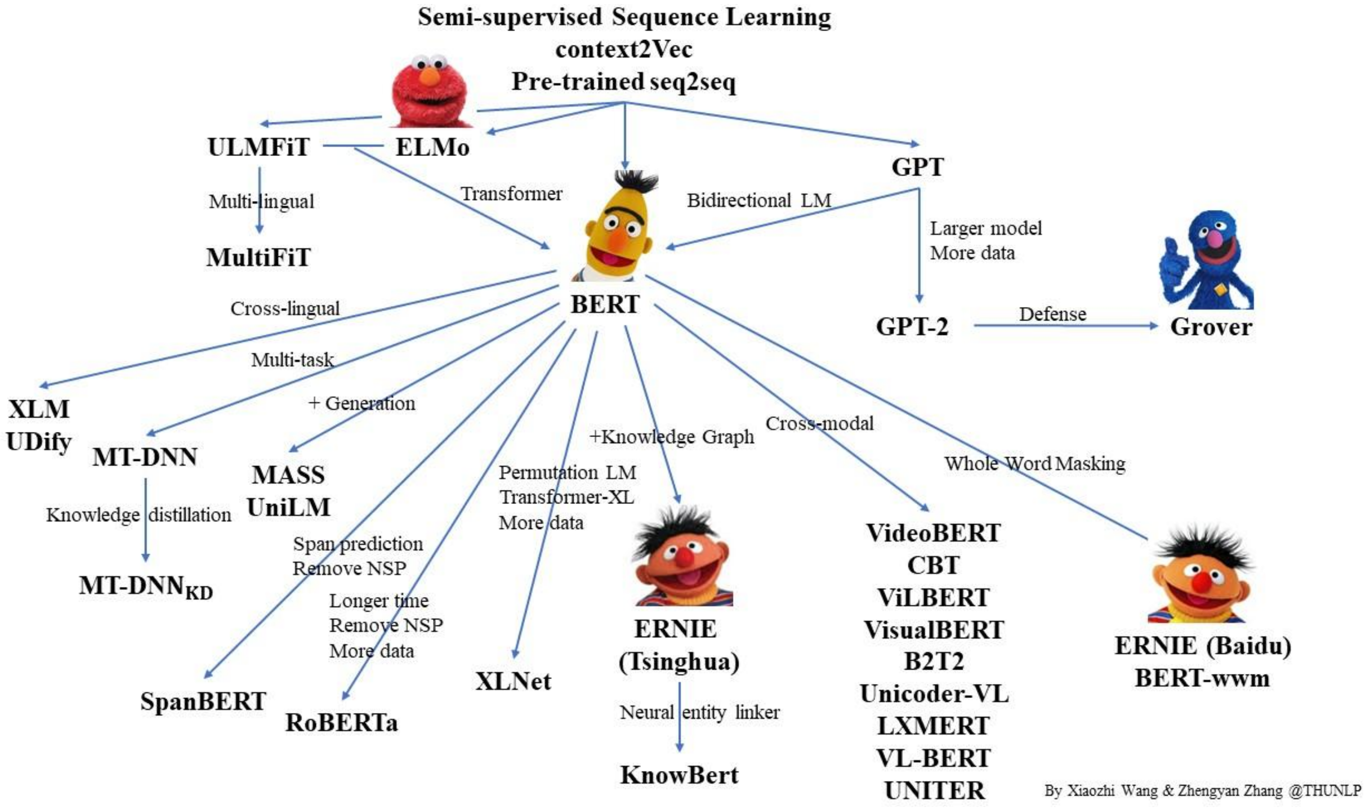
Distilling BERT

By Tu4n
@AIST

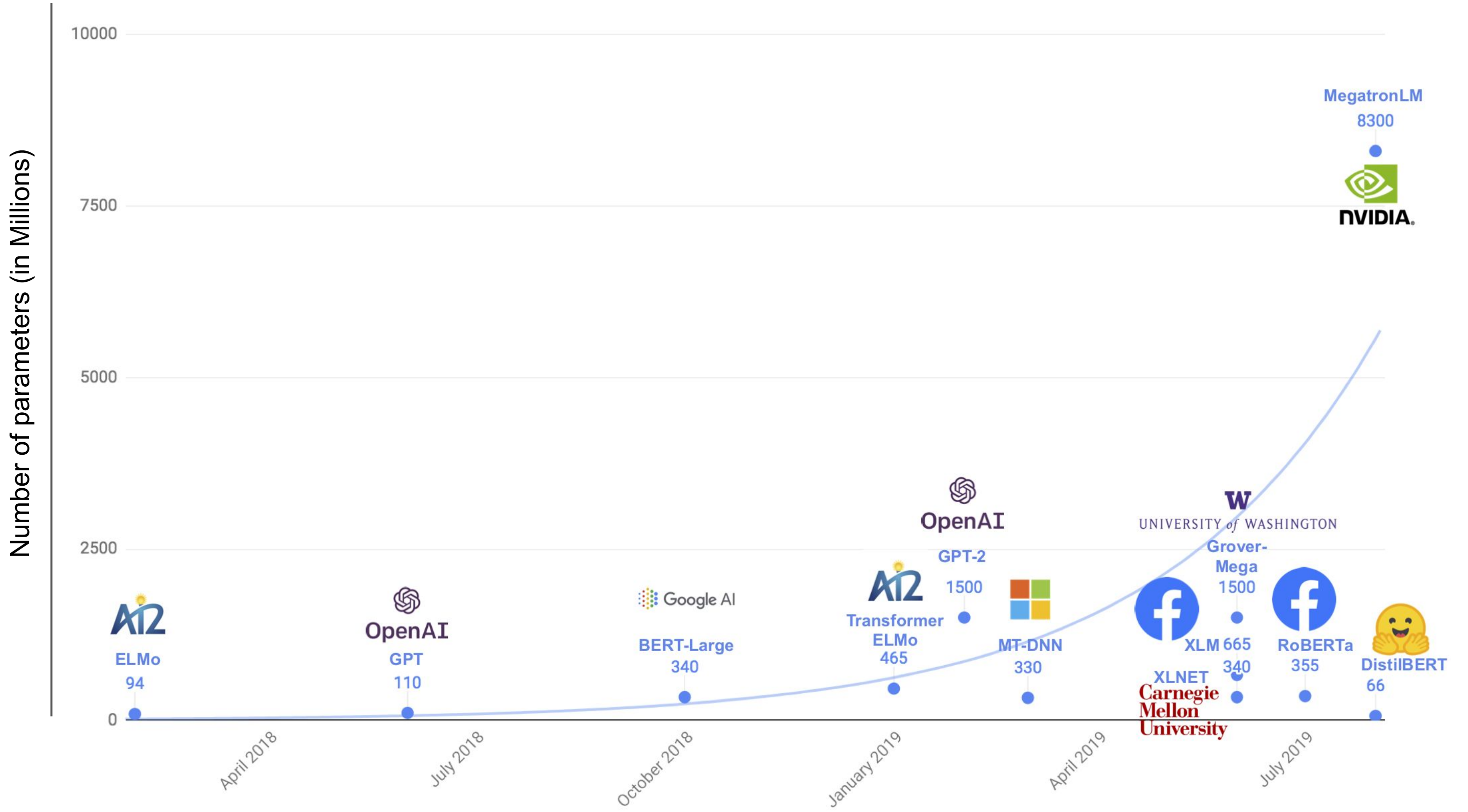
PAPERS

- DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter by huggingface, 2019 🤗
- Distilling Task-Specific Knowledge from BERT into Simple Neural Networks by Tang et al., 2019
- TinyBERT: Distilling BERT for Natural Language Understanding by Jiao et al., 2019

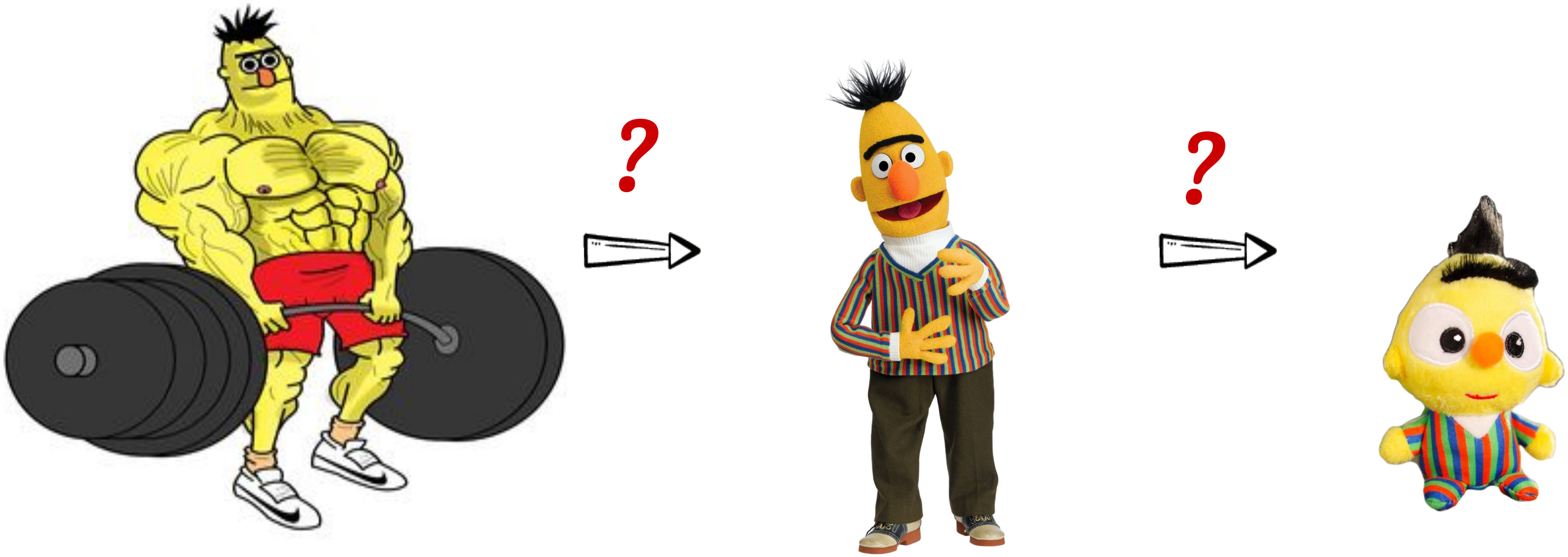
WHY?


Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
GPT
Multi-lingual
MultiFiT
Transformer
Bidirectional LM
Larger model
More data
BERT
GPT-2
Defense
Grover
Cross-lingual
Multi-task
XLM
UDify
MT-DNN
+ Generation
MASS
UniLM
Knowledge distillation
$MT\text{-}DNN_{KD}$
Span prediction
Remove NSP
Longer time
Remove NSP
More data
SpanBERT
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
Whole Word Masking
ERNIE (Baidu)
BERT-wwm
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

WHY?


Number of parameters (in Millions)
10000
7500
5000
2500
0
April 2018
July 2018
October 2018
January 2019
April 2019
July 2019
ELMo
94
OpenAI
GPT
110
Google AI
BERT-Large
340
Transformer ELMo
465
OpenAI
GPT-2
1500
MT-DNN
330
XLNET
340
Carnegie Mellon University
UNIVERSITY of WASHINGTON
Grover-Mega
1500
XLM 665
RoBERTa
355
DistilBERT
66
MegatronLM
8300
NVIDIA

WHY?



Distilling the Knowledge in a Neural Network

Geoff Hinton et al., 2015

DARK KNOWLEDGE

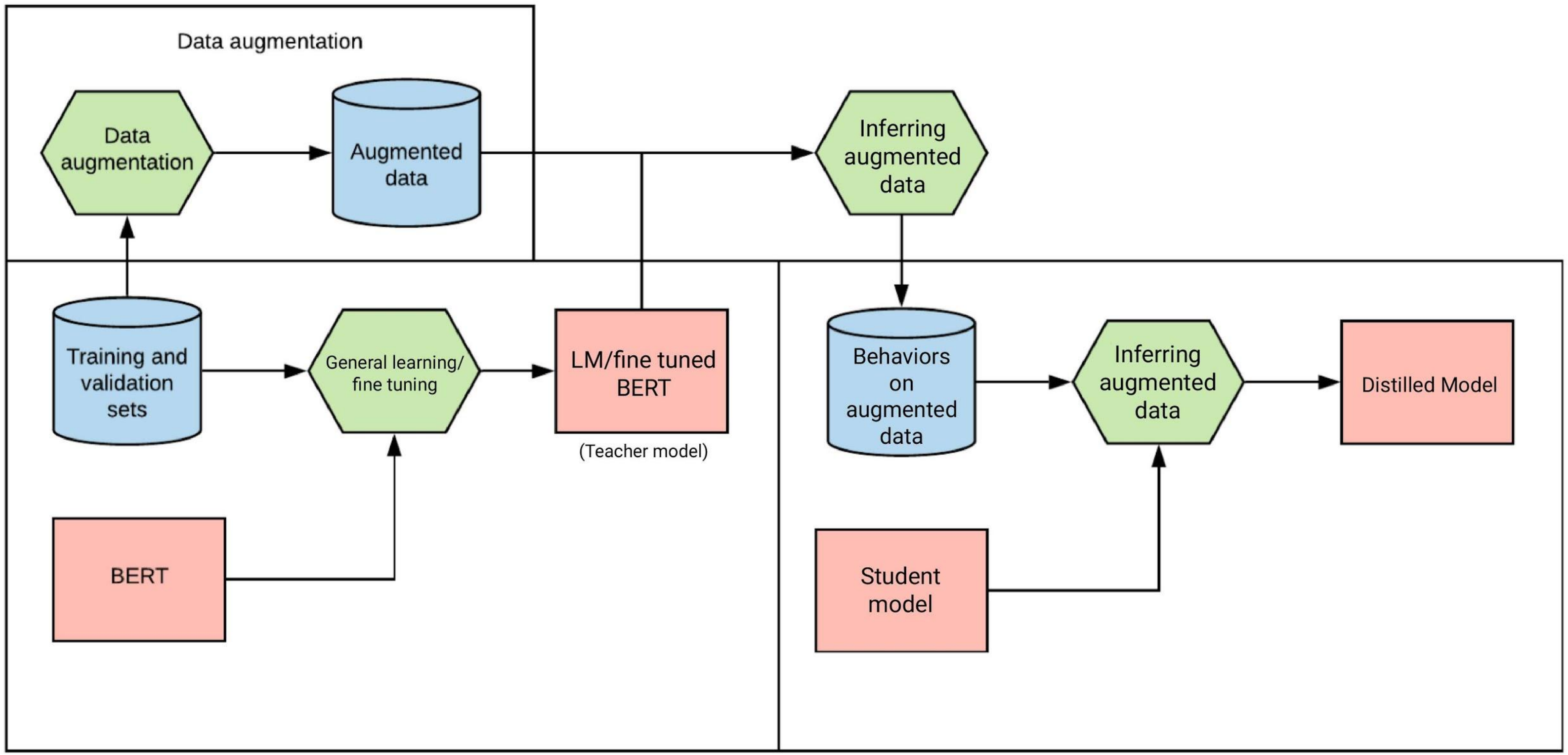
Knowledge Distillation

- “Many insects have a larval form that is optimized for extracting energy and nutrients from the environment and a completely different adult form that is optimized for the very different requirements of traveling and reproduction“

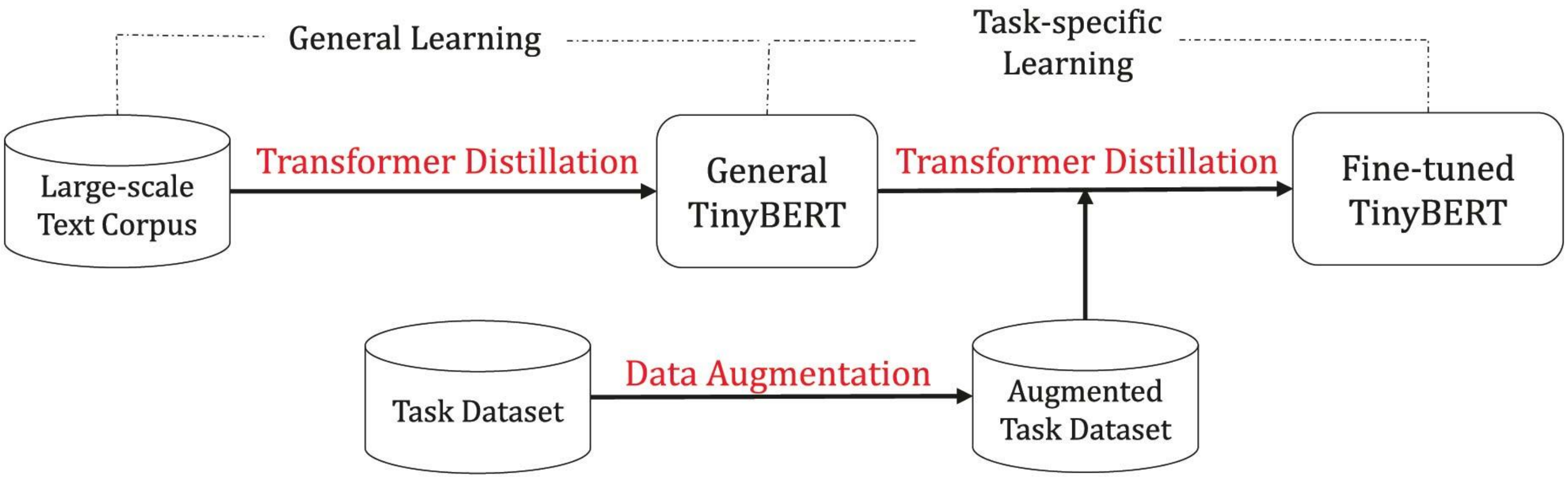
- “*Knowledge distillation* (sometimes also referred to as *teacher-student learning*) is a compression technique in which a small model is trained to reproduce the behavior of a larger model (or an ensemble of models).“

- “An image of a BMW, for example, may only have a very small chance of being mistaken for a garbage truck, but that mistake is still many times more probable than mistaking it for a carrot → Dark knowledge“

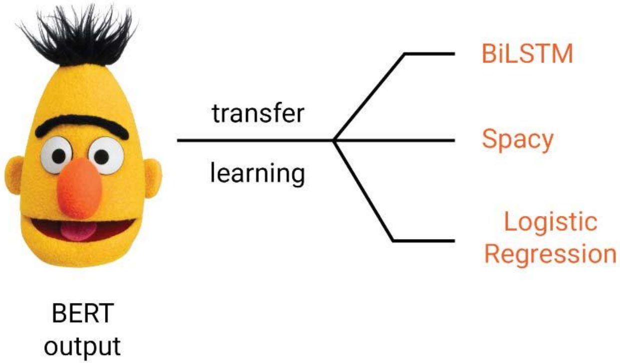
Teacher - student Framework



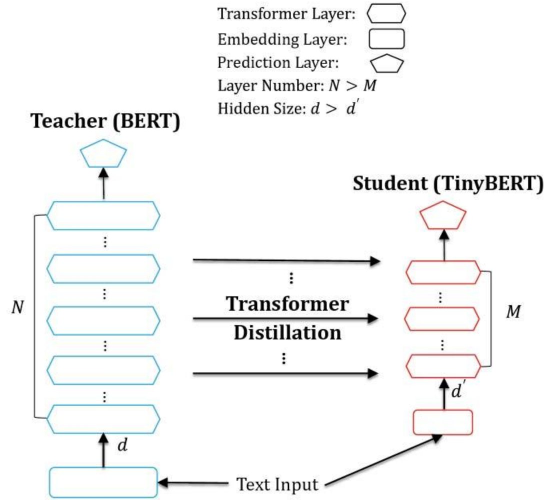
Teacher - student Framework



Teacher & student



Different architecture



Same architecture but smaller

Teacher & student

System	Layers	Hidden Size	Feed-forward Size	Model Size	Inference Time
BERT $_{BASE}$ (Teacher)	12	768	3072	109M(× 1.0)	188s(× 1.0)
Distilled BiLSTM $_{SOFT}$	1	300	400	10.1M(× 10.8)	24.8s(× 7.6)
BERT-PKD/DistilBERT	6	768	3072	52.2M(× 2.1)	63.7s(× 3.0)
TinyBERT	4	312	1200	14.5M(× 7.5)	19.9s(× 9.4)

Loss

Baseline: Train a student network to mimic **the full output distribution** of the teacher network (its knowledge)

Cross-entropy loss:

$$L = -\sum_i t_i * log(s_i)$$

With **t** the logits from the teacher and **s** the logits of the student

Softmax-temperature:

$$p_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

T is the temperature parameter.

Loss

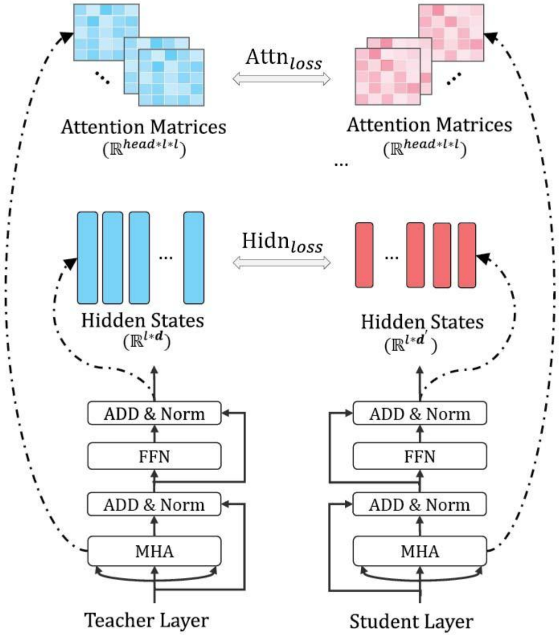
Kullback-Leibler loss

$$KL(p||q) = \mathbb{E}_p(log(\frac{p}{q}))$$

$$= \sum_i p_i * log(p_i) - \sum_i p_i * log(q_i)$$

```python
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.optim import Optimizer

KD_loss = nn.KLDivLoss(reduction='batchmean')

def kd_step(teacher: nn.Module,
            student: nn.Module,
            temperature: float,
            inputs: torch.tensor,
            optimizer: Optimizer):
    teacher.eval()
    student.train()

    with torch.no_grad():
        logits_t = teacher(inputs=inputs)
    logits_s = student(inputs=inputs)

    loss = KD_loss(input=F.log_softmax(logits_s/temperature, dim=-1),
                   target=F.softmax(logits_t/temperature, dim=-1))

    loss.backward()
    optimizer.step()
    optimizer.zero_grad()
```

Loss



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

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

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

Data Augmentation

- **Masking:** randomly replace a word with [MASK], which corresponds to the masked word (unknown) token in BERT.
- **POS-guided word replacement:** randomly replace a word with another of the same POS tag.
- **n-gram sampling:** randomly sample an n-gram from the example, where n is randomly selected from {1, 2, . . . , 5}.

Experimental results

GLUE

System	MNLI-m	MNLI-mm	QQP	SST-2	QNLI	MRPC	RTE	CoLA	STS-B	Avrg.
BERT$_{BASE}$ (Google)	84.6	83.4	71.2	93.5	90.5	88.9	66.4	52.1	85.8	79.6
BERT$_{BASE}$ (Teacher)	83.9	83.4	71.1	93.4	90.9	87.5	67.0	52.8	85.2	79.5
BERT$_{SMALL}$	75.4	74.9	66.5	87.6	84.8	83.2	62.6	19.5	77.1	70.2
Distilled BiLSTM$_{SOFT}$	73.0	72.6	68.2	90.7	-	-	-	-	-	-
BERT-PKD	79.9	79.3	70.2	89.4	85.1	82.6	62.3	24.8	79.8	72.6
DistilBERT	78.9	78.0	68.5	91.4	85.2	82.4	54.1	32.8	76.1	71.9
TinyBERT	82.5	81.8	71.3	92.6	87.7	86.4	62.9	43.3	79.9	76.5

Experimental results

Table 4: Results of wider or deeper TinyBERT variants and baselines.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
$\text{BERT}_{\text{BASE}}$ (Teacher)	84.2	84.4	86.8	57.4	78.2
BERT-PKD (M=6;d'=768;d'_i=3072)	80.9	80.9	83.1	43.1	72.0
DistilBERT (M=6;d'=768;d'_i=3072)	81.6	81.1	82.4	42.5	71.9
TinyBERT (M=4;d'=312;d'_i=1200)	82.8	82.9	85.8	49.7	75.3
TinyBERT (M=4;d'=768;d'_i=3072)	83.8	84.1	85.8	50.5	76.1
TinyBERT (M=6;d'=312;d'_i=1200)	83.3	84.0	86.3	50.6	76.1
TinyBERT (M=6;d'=768;d'_i=3072)	84.5	84.5	86.3	54.0	77.3

Experimental results

Table 5: Ablation studies of different procedures (i.e., TD, GD, and DA) of the two-stage learning framework. The variants are validated on the dev set.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
TinyBERT	82.8	82.9	85.8	49.7	75.3
No GD	82.5	82.6	84.1	40.8	72.5
No TD	80.6	81.2	83.8	28.5	68.5
No DA	80.5	81.0	82.4	29.8	68.4

Table 6: Ablation studies of different distillation objectives in the TinyBERT learning. The variants are validated on the dev set.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
TinyBERT	82.8	82.9	85.8	49.7	75.3
No Embd	82.3	82.3	85.0	46.7	74.1
No Pred	80.5	81.0	84.3	48.2	73.5
No Trm	71.7	72.3	70.1	11.2	56.3
No Attn	79.9	80.7	82.3	41.1	71.0
No Hidn	81.7	82.1	84.1	43.7	72.9

Table 7: Results (dev) of different mapping strategies.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
TinyBERT (Uniform-strategy)	82.8	82.9	85.8	49.7	75.3
TinyBERT (Top-strategy)	81.7	82.3	83.6	35.9	70.9
TinyBERT (Bottom-strategy)	80.6	81.3	84.6	38.5	71.3

Experimental results

	No.params	Inference Speed
BERT *Base*	100	100
Distilled BiLSTM	1	1500
DistilBERT	60	160
Tiny BERT	13	940

Thank you!