Knowledge Distillation

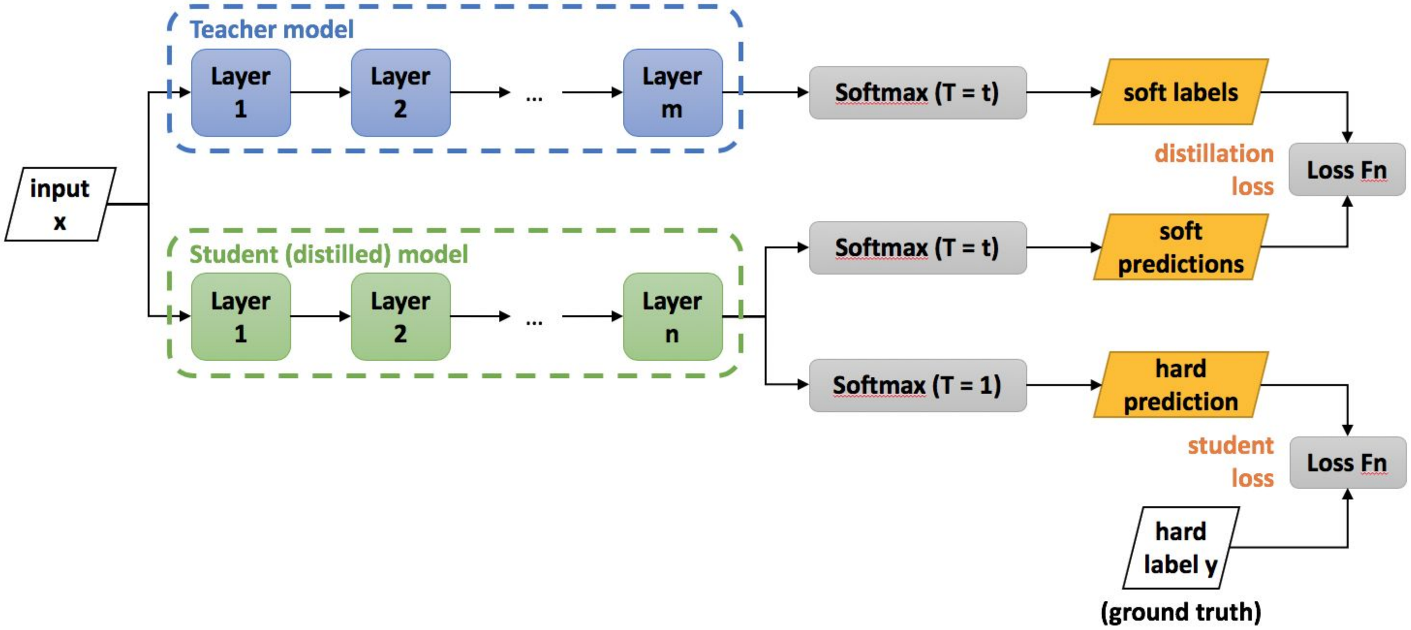
DistilBERT | MobileBERT | TinyBERT

DistilBERT

A distilled version of BERT: Smaller, faster, cheaper and lighter

What is Knowledge Distillation?

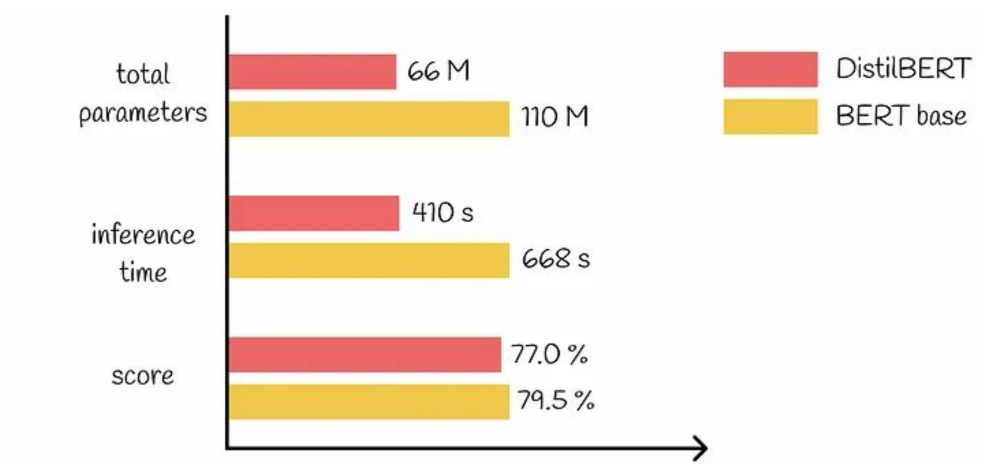
- Knowledge Distillation is a technique in machine learning where a smaller, simpler model (student) is trained to mimic the performance of a larger, more complex model (teacher).
- It is done for Model compression, inference speedup and deployment efficiency



Why Distil BERT?

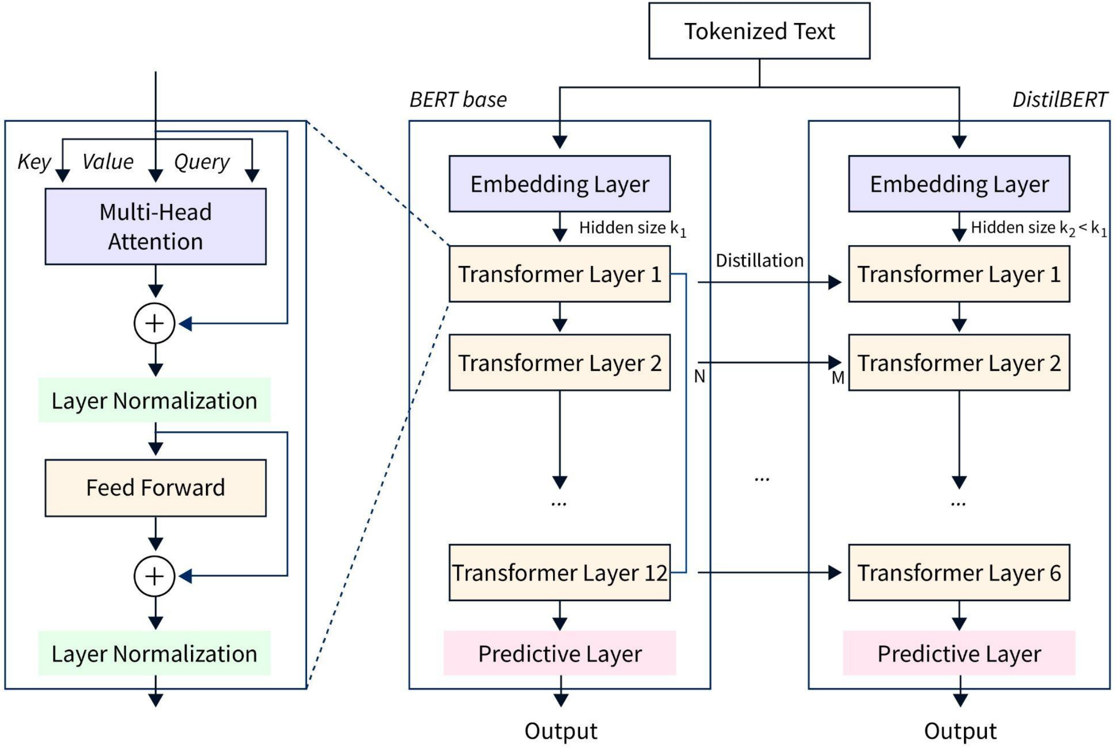
- During inference, DistilBERT is 60% faster than BERT.
- DistilBERT has 44M fewer parameters and in total is 40% smaller than BERT.
- DistilBERT retains 97% of BERT performance.

BERT vs DistilBERT comparison (on GLUE dataset)



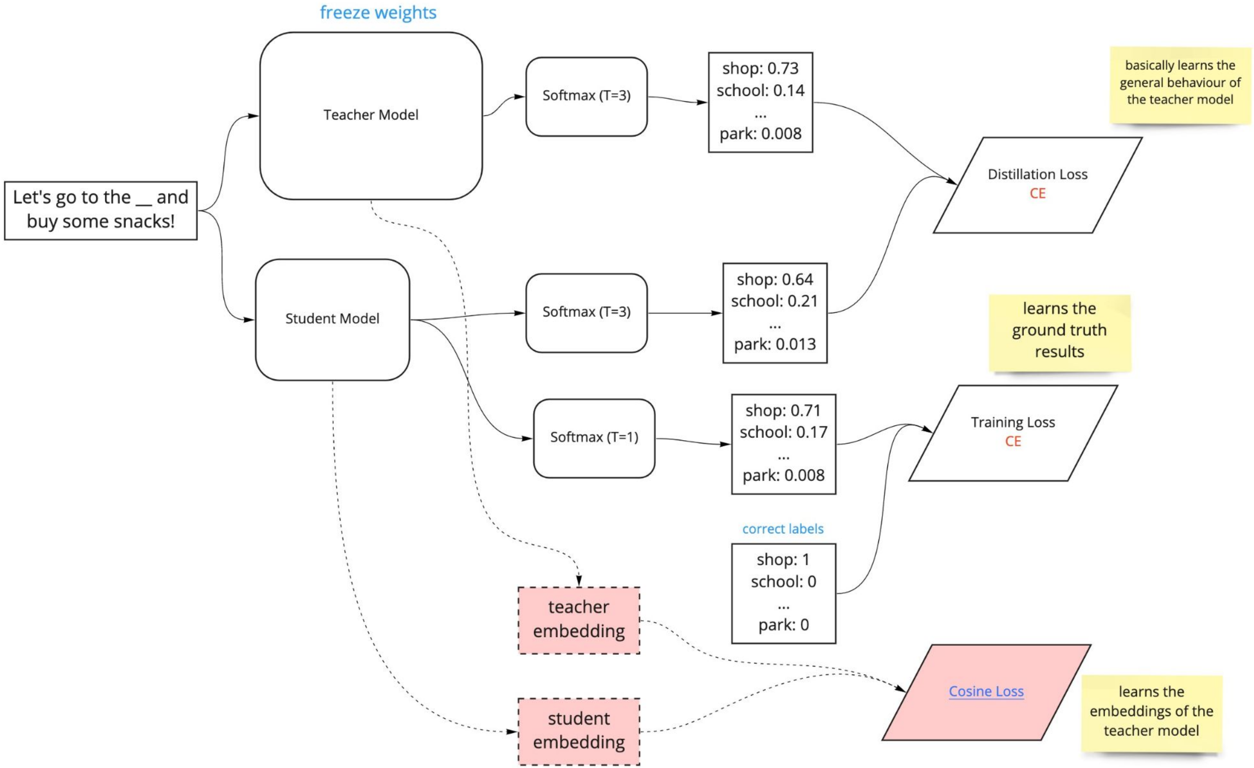
DistilBERT Architecture

- DistilBERT reduces the number of layers from 12 in BERT-base to 6.
- The student model (DistilBERT) is trained to predict the probability distribution over the vocabulary produced by the teacher model (BERT) using the same input text.
- The student model learns to replicate the teacher's attention patterns.
- During training, optimization strategies such as temperature scaling are applied to the softmax outputs.

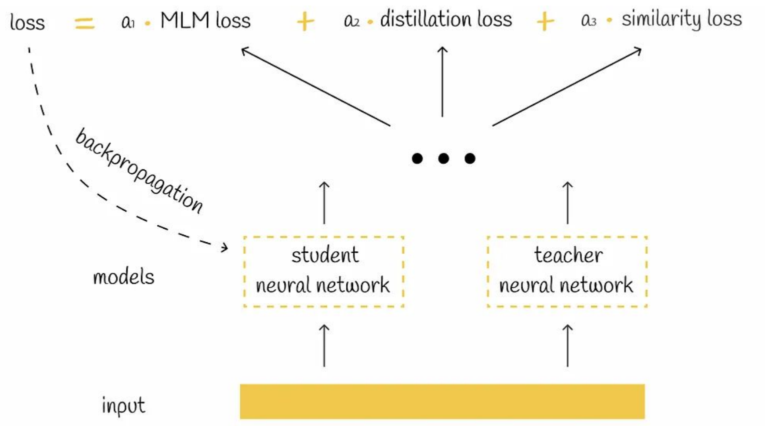


DistilBERT Loss Function

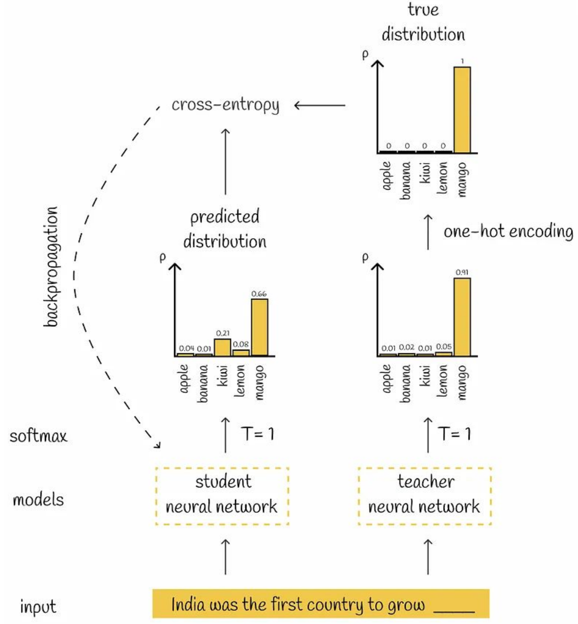
- DistilBERT learns from BERT and updates its weights by using the loss function which consists of three components:
    - Masked language modeling (MLM) loss
    - Distillation loss
    - Similarity loss



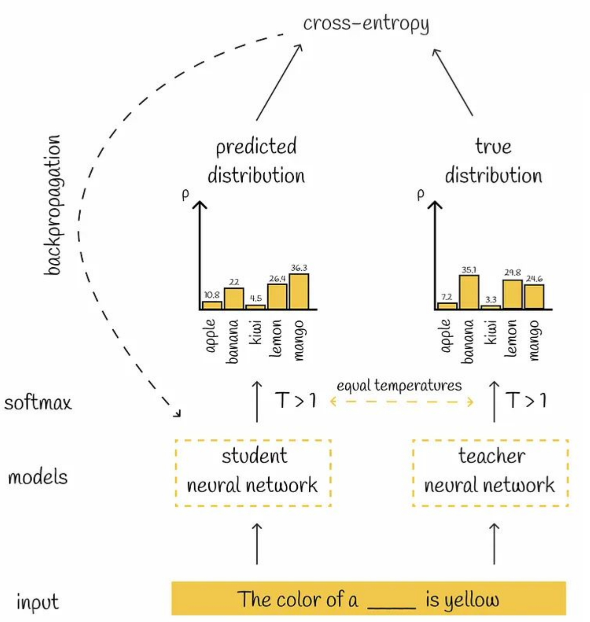
DistilBERT Loss



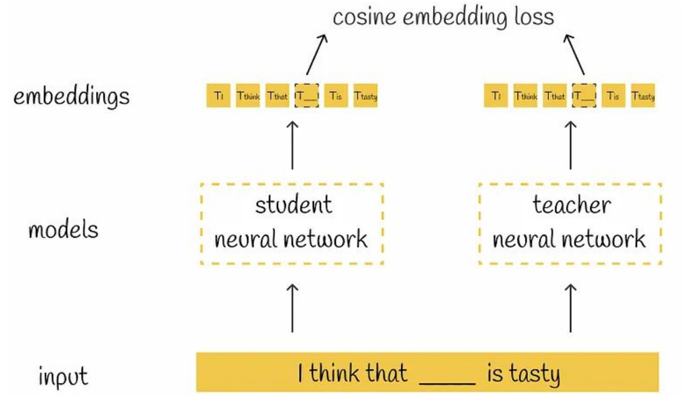
Triple loss



MLM loss



Distillation loss



Similarity loss

DistilBERT Performance Comparison

Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Table 2: **DistilBERT yields to comparable performance on downstream tasks.** Comparison on downstream tasks: IMDb (test accuracy) and SQuAD 1.1 (EM/F1 on dev set). D: with a second step of distillation during fine-tuning.

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Table 3: **DistilBERT is significantly smaller while being constantly faster.** Inference time of a full pass of GLUE task STS-B (sentiment analysis) on CPU with a batch size of 1.

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

MobileBERT

a Compact Task-Agnostic BERT for Resource-Limited Devices

What is mobileBERT?

- **MobileBERT** compresses and accelerates **BERT** to enable deployment on mobile devices with limited resources while maintaining high performance.
- **MobileBERT** is a versatile, task-agnostic model that can be fine-tuned for various NLP tasks without task-specific modifications.
- **MobileBERT** uses a "thin" version of **BERTLARGE** with bottleneck structures and balanced self-attentions and feed-forward networks to reduce computational load.
- **MobileBERT** is trained by transferring knowledge from an inverted-bottleneck **BERTLARGE** (IB-BERT) model, ensuring the smaller model retains high performance.
- **MobileBERT** is 4.3 times smaller and 5.5 times faster than **BERTBASE**, achieving competitive results on benchmarks like **GLUE** and **SQuAD**.

MobileBERT Architecture

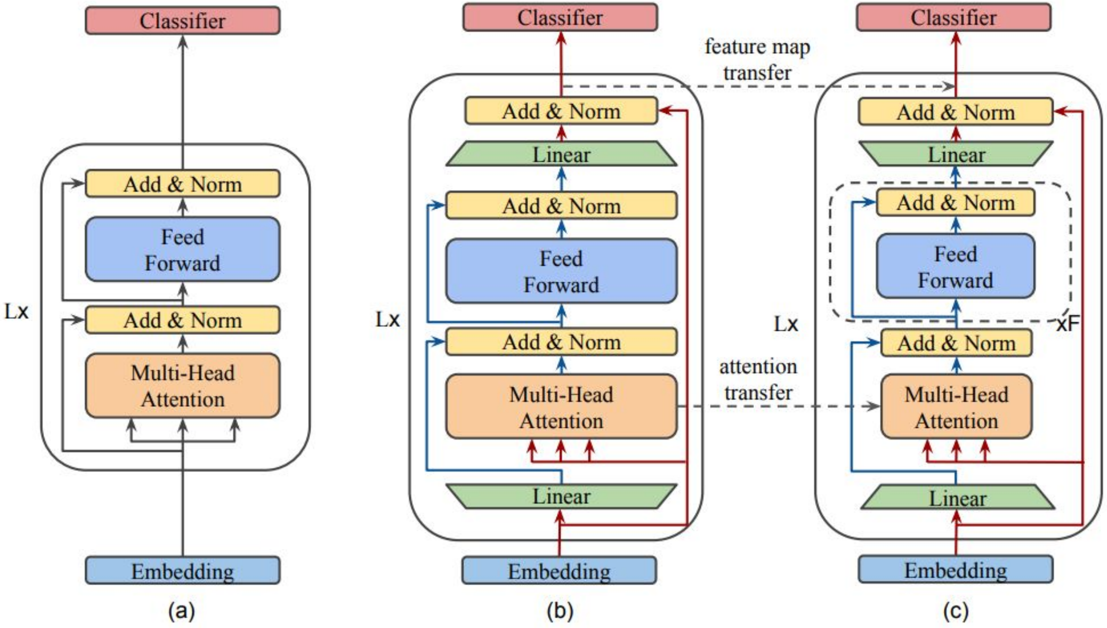


Figure 1: Illustration of three models: (a) BERT; (b) Inverted-Bottleneck BERT (IB-BERT); and (c) MobileBERT. In (b) and (c), red lines denote inter-block flows while blue lines intra-block flows. MobileBERT is trained by layer-to-layer imitating IB-BERT.

MobileBERT Model Params Settings

			$BERT_{LARGE}$	$BERT_{BASE}$	$IB\text{-}BERT_{LARGE}$	MobileBERT	$MobileBERT_{TINY}$
embedding		$h_{embedding}$	1024	768	128	128	128
			no-op	no-op	3-convolution	3-convolution	3-convolution
		h_{inter}	1024	768	512	512	512
body	Linear	h_{input}			512	512	512
		h_{output}			1024	128	128
	MHA	h_{input}	1024	768	512	512	128
		#Head	16	12	4	4	4
		h_{output}	1024	768	1024	128	128
	FFN	h_{input}	1024	768	1024	128	128
		h_{FFN}	4096	3072	4096	512 ×4	512 ×2
		h_{output}	1024	768	1024	128	128
	Linear	h_{input}			1024	128	128
		h_{output}			512	512	512
			×24	×12	×24	×24	×24
#Params			334M	109M	293M	25.3M	15.1M

Table 1: The detailed model settings of a few models. h_{inter}, h_{FFN}, $h_{embedding}$, #Head and #Params denote the inter-block hidden size (feature map size), FFN intermediate size, embedding table size, the number of heads in multi-head attention, and the number of parameters, respectively.

MobileBERT Knowledge Distillation

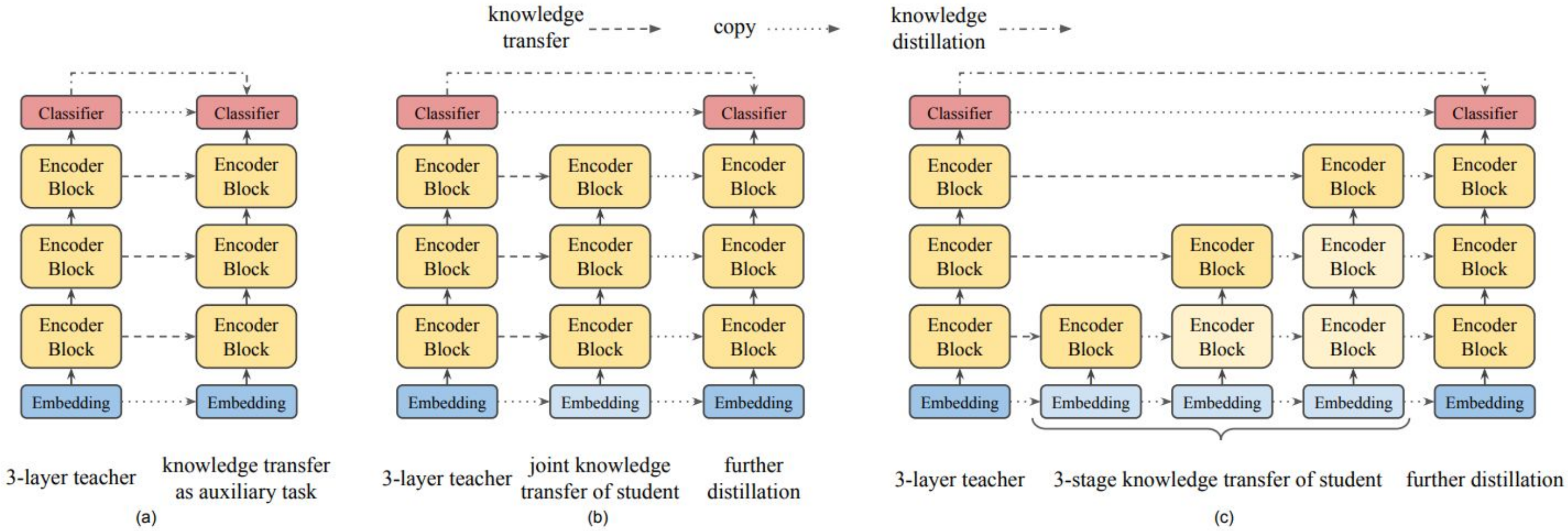


Figure 2: Diagrams of (a) auxiliary knowledge transfer (AKT), (b) joint knowledge transfer (JKT), and (c) progressive knowledge transfer (PKT). Lighter colored blocks represent that they are frozen in that stage.

MobileBERT Benchmarking

	#Params	#FLOPS	Latency	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 5.7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	GLUE
ELMo-BiLSTM-Attn	-	-	-	33.6	90.4	84.4	72.3	63.1	74.1/74.5	79.8	58.9	70.0
OpenAI GPT	109M	-	-	47.2	93.1	87.7	84.8	70.1	80.7/80.6	87.2	69.1	76.9
$BERT_{BASE}$	109M	22.5B	342 ms	**52.1**	**93.5**	**88.9**	**85.8**	71.2	**84.6/83.4**	90.5	66.4	78.3
$BERT_{BASE}$-6L-PKD*	66.5M	11.3B	-	-	92.0	85.0	-	70.7	81.5/81.0	89.0	65.5	-
$BERT_{BASE}$-4L-PKD†*	52.2M	7.6B	-	24.8	89.4	82.6	79.8	70.2	79.9/79.3	85.1	62.3	-
$BERT_{BASE}$-3L-PKD*	45.3M	5.7B	-	-	87.5	80.7	-	68.1	76.7/76.3	84.7	58.2	-
$DistilBERT_{BASE}$-6L†	62.2M	11.3B	-	-	92.0	85.0		70.7	81.5/81.0	89.0	65.5	-
$DistilBERT_{BASE}$-4L†	52.2M	7.6B	-	32.8	91.4	82.4	76.1	68.5	78.9/78.0	85.2	54.1	-
TinyBERT*	14.5M	1.2B	-	43.3	92.6	86.4	79.9	**71.3**	82.5/81.8	87.7	62.9	75.4
$MobileBERT_{TINY}$	15.1M	3.1B	40 ms	46.7	91.7	87.9	80.1	68.9	81.5/81.6	89.5	65.1	75.8
MobileBERT	25.3M	5.7B	62 ms	50.5	92.8	88.8	84.4	70.2	83.3/82.6	90.6	66.2	77.7
MobileBERT w/o OPT	25.3M	5.7B	192 ms	51.1	92.6	88.8	84.8	70.5	84.3/**83.4**	**91.6**	**70.4**	**78.5**

Table 4: The test results on the GLUE benchmark (except WNLI). The number below each task denotes the number of training examples. The metrics for these tasks can be found in the GLUE paper (Wang et al., 2018). "OPT" denotes the operational optimizations introduced in Section 3.3. †denotes that the results are taken from (Jiao et al., 2019). *denotes that it can be unfair to directly compare MobileBERT with these models since MobileBERT is task-agnosticly compressed while these models use the teacher model in the fine-tuning stage.

TinyBERT

Distilling BERT for Natural Language Understanding

What is TinyBERT?

- **TinyBERT** is created to reduce the size and improve the speed of **BERT** while maintaining high performance on NLP tasks.
- It uses a unique Transformer distillation method to effectively transfer knowledge from a larger **BERT** model to a smaller **TinyBERT** model.
- **TinyBERT** employs a two-stage learning process involving general distillation from a non-fine-tuned **BERT** and task-specific distillation from a fine-tuned **BERT**, enhancing both general and task-specific capabilities.
- **TinyBERT** with 4 layers achieves over 96.8% of **BERTBASE's** performance on the **GLUE** benchmark, while being 7.5 times smaller and 9.4 times faster in inference.

How Similar All These Distilled Models Are?

- All three models—TinyBERT, DistilBERT, and MobileBERT—aim to reduce the size of the original BERT model to make it more efficient and deployable on devices with limited computational resources.
- Each model utilizes knowledge distillation techniques to transfer knowledge from a larger, more complex "teacher" model to a smaller, more efficient "student" model, preserving the teacher's capabilities while reducing computational demands.
- TinyBERT, DistilBERT, and MobileBERT maintain the ability to be fine-tuned for a variety of downstream NLP tasks, making them versatile across different applications without requiring task-specific pre-training.
- Despite their reduced sizes and faster inference times, all three models—TinyBERT, DistilBERT, and MobileBERT—achieve performance that is competitive with or close to the original BERT model on various NLP benchmarks.

TinyBERT Distillation

- It propose a novel two-stage learning framework including the general distillation and the task-specific distillation,
- It has three types of loss functions to fit different representations from BERT layers:
  a. the output of the embedding layer;
  b. the hidden states and attention matrices derived from the Transformer layer;
  c. the logits output by the prediction layer.

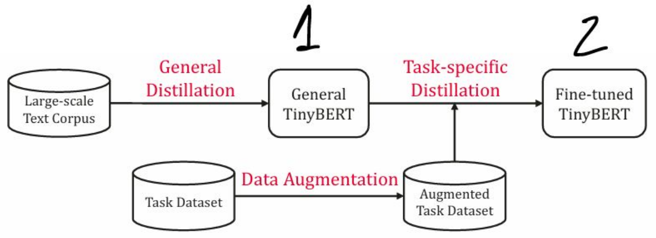


Figure 1: The illustration of TinyBERT learning.

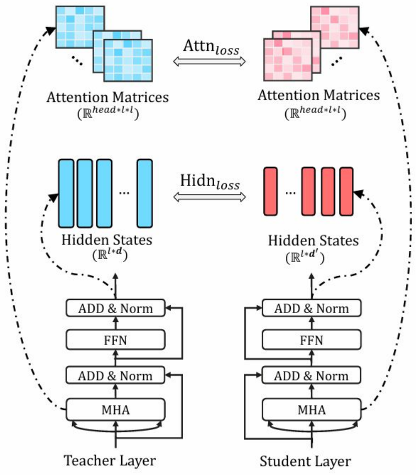


Figure 2: The details of Transformer-layer distillation consisting of Attn$_{loss}$(attention based distillation) and Hidn$_{loss}$(hidden states based distillation).

TinyBERT Model Performance

System	#Params	#FLOPs	Speedup	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Avg
$\text{BERT}_{\text{BASE}}$ (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
$\text{BERT}_{\text{TINY}}$	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
$\text{BERT}_{\text{SMALL}}$	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
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
$\text{MobileBERT}_{\text{TINY}}$†	15.1M	3.1B	-	81.5/81.6	68.9	**89.5**	91.7	**46.7**	80.1	**87.9**	65.1	**77.0**
TinyBERT_4 (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	-
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
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 TinyBERT_4 and $\text{BERT}_{\text{TINY}}$ is (M=4, d=312, d_i=1200), $\text{BERT}_{\text{SMALL}}$ is (M=4, d=512, d_i=2048), BERT_4-PKD and DistilBERT_4 is (M=4, d=768, d_i=3072) and the architecture of BERT_6-PKD, DistilBERT_6 and 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 $\text{MobileBERT}_{\text{TINY}}$ and TinyBERT_4 may not be fair since the former has 24 layers and is task-agnosticly distilled from $\text{IB-BERT}_{\text{LARGE}}$ while the later is a 4-layers model task-specifically distilled from $\text{BERT}_{\text{BASE}}$.

Thanks