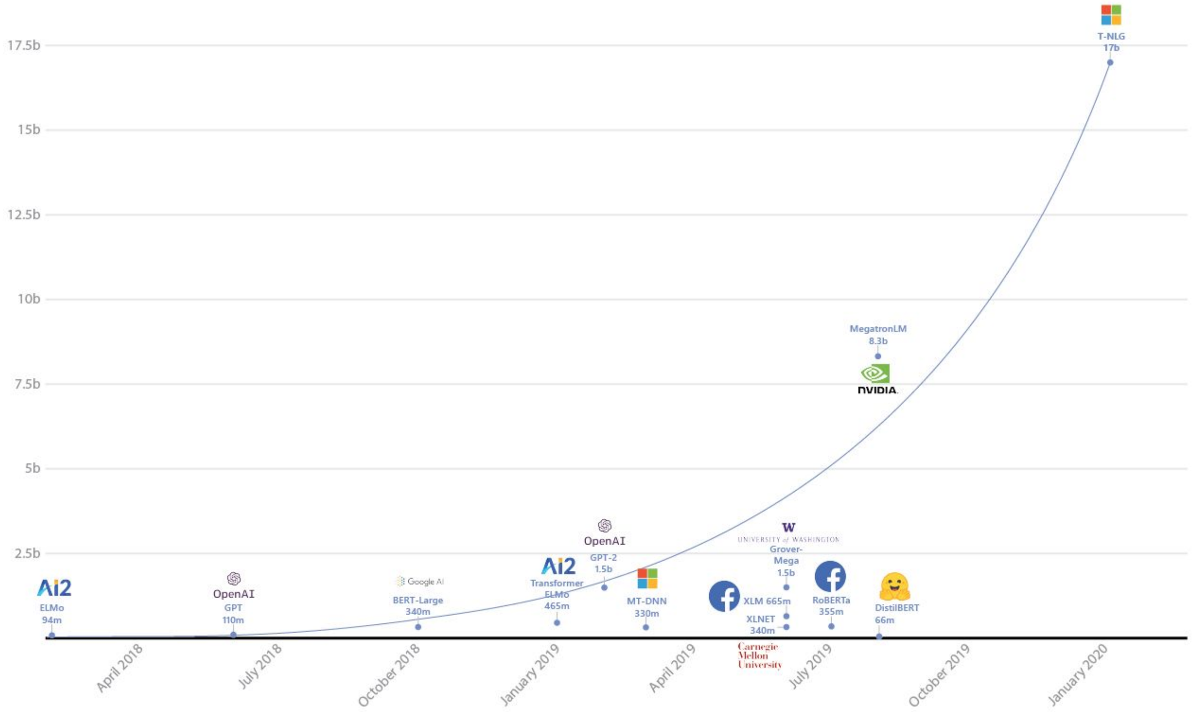
DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Sanh et al., 2020

Presented by Vardaan Pahuja

Parameter explosion in pre-trained LMs



Note: A more recent work (GPT-3) has 175 billion parameters.

Figure credit: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

Parameter explosion in pre-trained LMs

- The pre-trained language models in the BERT family keep getting larger and larger (in terms of parameter count) and are being trained on even bigger datasets.
- The latest model from Nvidia has 8.3 billion parameters (for the GPT variant): 24 times larger than BERT-large, 5 times larger than GPT-2.

- RoBERTa, the latest work from Facebook AI, was trained on 160GB of text.

Training times of pre-trained LMs

	BERT	RoBERTa	XLNet	MegatronLM	DistilBERT
Parameter count (millions)	Base: 110 Large: 340	Base: 110 Large: 340	Base: ~110 Large: ~340	3900	Base: 66
Training time	Base: 8 x V100 x 12 days Large: 64 TPU chips x 4 days	Large: 1024 x V100 x 1 day; 4-5 times more than BERT	Large: 512 TPU chips x 2.5 days: 5 times more than BERT	512 GPU x 15 days*	Base: 8 x V100 x 3.5 days
Data	16 GB BERT data (3.3 Billion words)	160 GB (16 GB BERT data + 144 GB additional)	Base: 16 GB BERT data Large: 113 GB (16 GB BERT data + 97 GB additional)	16 GB BERT data	16 GB BERT data (3.3 Billion words)

Source: https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8

* indicates approximate time

Drawbacks of using bigger LMs in production-level code

- Need huge amount of computational resources even for inference.
- For edge devices like mobile phones, need to call a cloud API to run inference on possibly private data (not desirable).
- **Solution**: Develop lightweight and energy-efficient models which have low latency on edge devices without compromising much on performance.

Knowledge Distillation [Hinton et al. 2015]

- Train a classifier using the real-valued outputs of another classifier as target values than using actual ground-truth labels.
- A trained classifier model assigns probabilities to all labels (incl. the incorrect labels).
- The relative magnitude of these incorrect probabilities affects the generalization capability of the model.
- For instance: An image of bus might be mistaken for a car by an image classification model, but is unlikely to be mistaken for a chair.

Training procedure

- Compute “soft labels” by using softmax with temperature (T).
- Equation to compute soft probabilities for the student model.

$$(y_S^{\tau})_i = \frac{exp((z_S)_i/T)}{\sum_j exp((z_S)_j/T)}$$

$$\mathcal{L}_{KD} = \alpha T^2 * \mathcal{L}_{CE}(y_T^{\tau}, y_S^{\tau}) + (1-\alpha) * \mathcal{L}_{CE}(y_S, y_{true})$$

- Notation:
- y_T^{τ} Denotes softmax probabilities for teacher model (with temp. T) / soft labels
- y_S^{τ} Denotes softmax probabilities for student model (with temp. T).
- y_S Denotes softmax probabilities with T=1
- y_{true} Denotes the gold labels (or hard labels).
- α : hyper-parameter
- $(z_S)_i$: denotes the i^{th} logit for the student model

Training procedure

$$\mathcal{L}_{KD} = \alpha T^2 * \mathcal{L}_{CE}(y_T^{\tau}, y_S^{\tau}) + (1-\alpha) * \mathcal{L}_{CE}(y_S, y_{true})$$

- The multiplying factor of T^2 is used for the first loss term because the magnitude of gradients for the soft labels scale as $1/T^2$ compared to the other term (see Hinton et al. 2015 for derivation)
- This ensures that the relative contributions of the hard and soft targets remain roughly unchanged if the temperature used for distillation is changed while experimenting with meta-parameters.

DistilBERT Model Structure

- Omit token-type embeddings (as there is no Next Sentence Prediction objective).
- The number of layers is reduced from 12 layers (BERT-base) to 6 layers.
- **Initialization**: Initialize the student from the teacher by taking one layer out of two (use alternate layers from the original pre-trained checkpoint).

DistilBERT Training details

- The final training objective is a linear combination of:
  - Distillation loss: KL divergence loss b/w softmax probabilities calculated with temp. T
  - Supervised training loss/cross-entropy loss (L_{MLM} in case of DistilBERT)
  - Cosine embedding loss (L_{cos}) between the hidden states vectors of student and teacher models.

$$\mathcal{L} = \alpha_d T^2 * \mathcal{L}_{dist} + \alpha_{mlm} * \mathcal{L}_{MLM} + \alpha_{cos} * \mathcal{L}_{cos}$$

$$\mathcal{L}_{cos}(\mathbf{x}_1, \mathbf{x}_2) = 1 - cos(\mathbf{x}_1, \mathbf{x}_2)$$

$$cos(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 . \mathbf{x}_2}{||\mathbf{x}_1||||\mathbf{x}_2||}$$

Hyperparameter values from code repo:
T = 2.0
α_d = 5.0
α_{mlm}= 2.0
α_{cos}= 1.0

DistilBERT Training details

- DistilBERT borrows certain best practices of trained BERT model from RoBERTa (Liu et al. 2019).
- Modifications from original BERT model:
    - Use large batch size (=4000) with gradient accumulation (gradients from multiple mini-batches are accumulated locally before each optimization step).
    - Dynamic masking (compared to static masking in the original BERT model)
    - Omitting the Next Sentence Prediction objective.
    - Omit the use of segment embeddings.

Dataset and computational resource

- **Training dataset**: English Wikipedia + Toronto Book Corpus (same as BERT)
- **Training time**: 90 hours on 8 16GB V100 GPUs (compared to RoBERTa model which is trained for 1 day on 1024 32 GB V100 GPUs)
- Approximately 3.5x speedup in training time compared to the BERT model.

Results on GLUE benchmark

- General Language Understanding Evaluation (GLUE) benchmark

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

- DistilBERT is always on par or improving over the ELMo baseline.
- Performs surprisingly well compared to BERT, retains 97% performance with 40% fewer parameters.

Results on Downstream tasks

- Downstream tasks
  - IMBb sentiment classification
  - SQuAD Question-Answering task
- DistilBERT is only 0.6% point behind BERT in test accuracy on the IMDb benchmark
- On SQuAD, DistilBERT is within 3.9 points of the full BERT.
- Another approach: 2-step distillation (DistilBERT(D))
  - Use knowledge distillation in fine-tuning phase using a BERT model fine-tuned on SQuAD as a teacher.

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

Inference efficiency

- 40% fewer parameters than BERT
- 60% faster than BERT in terms of inference speed on CPU
- 71% faster than BERT on mobile device (iPhone 7 Plus) with lower memory footprint.

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

Ablation study

- Goal: Investigate the influence of various components of the triple loss and the student initialization on the performances of the distilled model
- The *Masked Language Modeling* loss has little effect on the model performance.
- The distillation loss, cosine embedding loss and teacher weights initialization have significant impact on model performance.

Table 4: **Ablation study.** Variations are relative to the model trained with triple loss and teacher weights initialization.

Ablation	**Variation on GLUE macro-score**
$\emptyset$ - L_{cos} - L_{mlm}	-2.96
L_{ce} - $\emptyset$ - L_{mlm}	-1.46
L_{ce} - L_{cos} - $\emptyset$	-0.31
Triple loss + random weights initialization	-3.69

Related Work

- **TinyBERT** [Jiao et al. 2019] uses the hidden layer representation, embeddings and attention matrices in addition to the the output of prediction layer to perform distillation.
- Distillation of BERT into a single-layer BiLSTM achieving comparable results with ELMo, while using roughly 100 times fewer parameters and 15 times less inference time [Xiaoqi et al. 2019].
- Use ensemble of teachers using multi-task learning to regularize the distillation [Yang et al. 2019].
- **Multi-step Distillation** [Mirzadeh et al. 2019]: Use an intermediate-sized network (teacher-assistant) to bridge the gap between teacher and student networks.

References

- Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).
- Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).
- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
- Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pretraining for language understanding." *Advances in neural information processing systems*. 2019.
- Shoeybi, Mohammad, et al. "Megatron-lm: Training multi-billion parameter language models using gpu model parallelism." *arXiv preprint arXiv:1909.08053* (2019).
- Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

References

- Yang, Ze, et al. "Model compression with multi-task knowledge distillation for web-scale question answering system." arXiv preprint arXiv:1904.09636 (2019).
- Jiao, Xiaoqi, et al. "Tinybert: Distilling bert for natural language understanding." arXiv preprint arXiv:1909.10351 (2019).
- Tang, Raphael, et al. "Distilling task-specific knowledge from bert into simple neural networks." arXiv preprint arXiv:1903.12136 (2019).
- Mirzadeh, Seyed-Iman, et al. "Improved Knowledge Distillation via Teacher Assistant." arXiv preprint arXiv:1902.03393 (2019).
- https://blog.inten.to/speeding-up-bert-5528e18bb4ea
- https://medium.com/huggingface/distilbert-8cf3380435b5
- https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/