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

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
{victor,lysandre,julien,thomas}@huggingface.co

Abstract

As Transfer Learning from large-scale pre-trained models becomes more prevalent in Natural Language Processing (NLP), operating these large models in on-the-edge and/or under constrained computational training or inference budgets remain challenging. In this work, we propose a method to pre-train a smaller general-purpose language representation model, called DistilBERT, which can then be fine-tuned with good performances on a wide range of tasks like its larger counterparts. While most prior work investigated the use of distillation for building task-specific models, we leverage knowledge distillation during the pre-training phase and show that it is possible to reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. To leverage the inductive biases learned by larger models during pre-training, we introduce a triple loss combining language modeling, distillation and cosine-distance losses. Our smaller, faster and lighter model is cheaper to pre-train and we demonstrate its capabilities for on-device computations in a proof-of-concept experiment and a comparative on-device study.

1 Introduction

The last two years have seen the rise of Transfer Learning approaches in Natural Language Processing (NLP) with large-scale pre-trained language models becoming a basic tool in many NLP tasks [Devlin et al., 2018, Radford et al., 2019, Liu et al., 2019]. While these models lead to significant improvement, they often have several hundred million parameters and current research[1] on pre-trained models indicates that training even larger models still leads to better performances on downstream tasks.

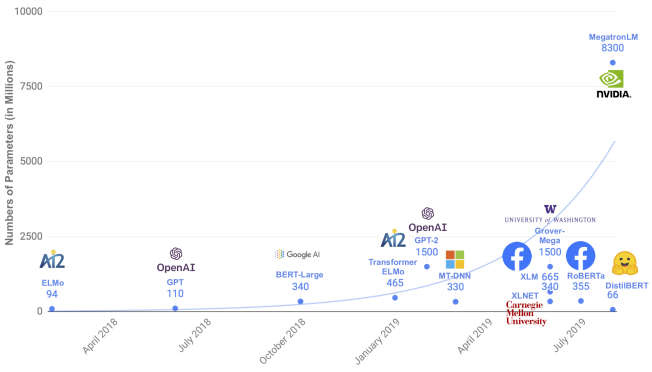

Figure 1: **Parameter counts of several recently released pretrained language models.**

The trend toward bigger models raises several concerns. First is the environmental cost of exponentially scaling these models' computational requirements as mentioned in Schwartz et al. [2019], Strubell et al. [2019]. Second, while operating these models on-device in real-time has the potential to enable novel and interesting language processing applications, the growing computational and memory requirements of these models may hamper wide adoption.

[1] See for instance the recently released MegatronLM (https://nv-adlr.github.io/MegatronLM)

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

In this paper, we show that it is possible to reach similar performances on many downstream-tasks using much smaller language models pre-trained with knowledge distillation, resulting in models that are lighter and faster at inference time, while also requiring a smaller computational training budget. Our general-purpose pre-trained models can be fine-tuned with good performances on several downstream tasks, keeping the flexibility of larger models. We also show that our compressed models are small enough to run on the edge, e.g. on mobile devices.

Using a triple loss, we show that a 40% smaller Transformer (Vaswani et al. [2017]) pre-trained through distillation via the supervision of a bigger Transformer language model can achieve similar performance on a variety of downstream tasks, while being 60% faster at inference time. Further ablation studies indicate that all the components of the triple loss are important for best performances.

We have made the trained weights available along with the training code in the `Transformers`[2] library from HuggingFace.

2 Knowledge distillation

Knowledge distillation [Bucila et al., 2006, Hinton et al., 2015] is a compression technique in which a compact model - the student - is trained to reproduce the behaviour of a larger model - the teacher - or an ensemble of models.

In supervised learning, a classification model is generally trained to predict an instance class by maximizing the estimated probability of gold labels. A standard training objective thus involves minimizing the cross-entropy between the model's predicted distribution and the one-hot empirical distribution of training labels. A model performing well on the training set will predict an output distribution with high probability on the correct class and with near-zero probabilities on other classes. But some of these "near-zero" probabilities are larger than others and reflect, in part, the generalization capabilities of the model and how well it will perform on the test set[3].

Training loss The student is trained with a distillation loss over the soft target probabilities of the teacher: $L_{ce} = \sum_i t_i * \log(s_i)$ where t_i (resp. s_i) is a probability estimated by the teacher (resp. the student). This objective results in a rich training signal by leveraging the full teacher distribution. Following Hinton et al. [2015] we used a *softmax-temperature*: $p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$ where T controls the smoothness of the output distribution and z_i is the model score for the class i. The same temperature T is applied to the student and the teacher at training time, while at inference, T is set to 1 to recover a standard *softmax*.

The final training objective is a linear combination of the distillation loss L_{ce} with the supervised training loss, in our case the *masked language modeling* loss L_{mlm} [Devlin et al., 2018]. We found it beneficial to add a *cosine embedding* loss (L_{cos}) which will tend to align the directions of the student and teacher hidden states vectors.

3 DistilBERT: a distilled version of BERT

Student architecture In the present work, the student - DistilBERT - has the same general architecture as BERT. The *token-type embeddings* and the *pooler* are removed while the number of layers is reduced by a factor of 2. Most of the operations used in the Transformer architecture (*linear layer* and *layer normalisation*) are highly optimized in modern linear algebra frameworks and our investigations showed that variations on the last dimension of the tensor (hidden size dimension) have a smaller impact on computation efficiency (for a fixed parameters budget) than variations on other factors like the number of layers. Thus we focus on reducing the number of layers.

Student initialization In addition to the previously described optimization and architectural choices, an important element in our training procedure is to find the right initialization for the sub-network to converge. Taking advantage of the common dimensionality between teacher and student networks, we initialize the student from the teacher by taking one layer out of two.

[2] https://github.com/huggingface/transformers

[3] E.g. BERT-base's predictions for a masked token in "`I think this is the beginning of a beautiful [MASK]`" comprise two high probability tokens (*day* and *life*) and a long tail of valid predictions (*future*, *story*, *world*...).

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	77.6	48.9	84.3	88.6	89.3	89.5	71.3	91.7	91.2	43.7
DistilBERT	76.8	49.1	81.8	90.2	90.2	89.2	62.9	92.7	90.7	44.4

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

Distillation We applied best practices for training BERT model recently proposed in Liu et al. [2019]. As such, DistilBERT is distilled on very large batches leveraging gradient accumulation (up to 4K examples per batch) using dynamic masking and without the next sentence prediction objective.

Data and compute power We train DistilBERT on the same corpus as the original BERT model: a concatenation of English Wikipedia and `Toronto Book Corpus` [Zhu et al., 2015]. DistilBERT was trained on 8 16GB V100 GPUs for approximately 90 hours. For the sake of comparison, the RoBERTa model [Liu et al., 2019] required 1 day of training on 1024 32GB V100.

4 Experiments

General Language Understanding We assess the language understanding and generalization capabilities of DistilBERT on the *General Language Understanding Evaluation* (GLUE) benchmark [Wang et al., 2018], a collection of 9 datasets for evaluating natural language understanding systems. We report scores on the development sets for each task by fine-tuning DistilBERT without the use of ensembling or multi-tasking scheme for fine-tuning (which are mostly orthogonal to the present work). We compare the results to the baseline provided by the authors of GLUE: an ELMo (Peters et al. [2018]) encoder followed by two BiLSTMs.[4]

The results on each of the 9 tasks are showed on Table 1 along with the macro-score (average of individual scores). Among the 9 tasks, DistilBERT is always on par or improving over the ELMo baseline (up to 20 points of accuracy on STS-B). DistilBERT also compares surprisingly well to BERT, retaining 97% of the performance with 40% fewer parameters.

4.1 Downstream task benchmark

Downstream tasks We further study the performances of DistilBERT on several downstream tasks under efficient inference constraints: a classification task (IMDb sentiment classification - Maas et al. [2011]) and a question answering task (SQuAD v1.1 - Rajpurkar et al. [2016]).

As shown in Table 2, DistilBERT is only 0.6% point behind BERT in test accuracy on the IMDb benchmark while being 40% smaller. On SQuAD, DistilBERT is within 3.5 points of the full BERT.

We also studied whether we could add another step of distillation during the adaptation phase by fine-tuning DistilBERT on SQuAD using a BERT model previously fine-tuned on SQuAD as a

[4]We use `jiant` [Wang et al., 2019] to compute the baseline.

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}	-5.06
L_{ce} - $\emptyset$ - L_{mlm}	-4.07
L_{ce} - L_{cos} - $\emptyset$	-1.90
Triple loss + random weights initialization	-4.83

teacher for an additional term in the loss (knowledge distillation). In this setting, there are thus two successive steps of distillation, one during the pre-training phase and one during the adaptation phase. In this case, we were able to reach interesting performances given the size of the model: 86.9 F1 and 79.1 EM, i.e. within 2 points of the full model.

Size and inference speed

To further investigate the speed-up/size trade-off of DistilBERT, we compare (in Table 3) the number of parameters of each model along with the inference time needed to do a full pass on the STS-B development set on CPU (Intel Xeon E5-2690 v3 Haswell @2.9GHz) using a batch size of 1. DistilBERT has 40% fewer parameters than BERT and is 60% faster than BERT.

On device computation We studied whether DistilBERT could be used for on-the-edge applications by building a mobile application for question answering. We compare the average inference time on a recent smartphone (iPhone 7 Plus) against our previously trained question answering model based on BERT-base. Excluding the tokenization step, DistilBERT is 71% faster than BERT, and the whole model weighs 207 MB (which could be further reduced with quantization). Our code is available[5].

4.2 Ablation study

In this section, we investigate the influence of various components of the triple loss and the student initialization on the performances of the distilled model. We report the macro-score on GLUE. Table 4 presents the deltas with the full triple loss: removing the *Masked Language Modeling* loss has little impact while the two distillation losses account for a large portion of the performance.

5 Related work

Task-specific distillation Most of the prior works focus on building task-specific distillation setups. Tang et al. [2019] transfer fine-tune classification model BERT to an LSTM-based classifier. Chatterjee [2019] distill BERT model fine-tuned on SQuAD in a smaller Transformer model previously initialized from BERT. In the present work, we found it beneficial to use a general-purpose pre-training distillation rather than a task-specific distillation. Turc et al. [2019] use the original pretraining objective to train smaller student, then fine-tuned via distillation. As shown in the ablation study, we found it beneficial to leverage the teacher's knowledge to pre-train with additional distillation signal.

Multi-distillation Yang et al. [2019] combine the knowledge of an ensemble of teachers using multi-task learning to regularize the distillation. The authors apply *Multi-Task Knowledge Distillation* to learn a compact question answering model from a set of large question answering models. An application of multi-distillation is multi-linguality: Tsai et al. [2019] adopts a similar approach to us by pre-training a multilingual model from scratch solely through distillation. However, as shown in the ablation study, leveraging the teacher's knowledge with initialization and additional losses leads to substantial gains.

Other compression techniques have been studied to compress large models. Recent developments in weights pruning reveal that it is possible to remove some heads in the self-attention at test time without significantly degrading the performance Michel et al. [2019]. Some layers can be reduced to one head. A separate line of study leverages quantization to derive smaller models (Gupta et al. [2015]). Pruning and quantization are orthogonal to the present work.

[5]https://github.com/huggingface/swift-coreml-transformers

6 Conclusion and future work

We introduced DistilBERT, a general-purpose pre-trained version of BERT, 40% smaller, 60% faster, that retains 97% of the language understanding capabilities. We showed that a general-purpose language model can be successfully trained with distillation and analyzed the various components with an ablation study. We further demonstrated that DistilBERT is a compelling option for edge applications.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2018.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar S. Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *ArXiv*, abs/1907.10597, 2019.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *ACL*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2018.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

Alex Wang, Ian F. Tenney, Yada Pruksachatkun, Katherin Yu, Jan Hula, Patrick Xia, Raghu Pappagari, Shuning Jin, R. Thomas McCoy, Roma Patel, Yinghui Huang, Jason Phang, Edouard Grave, Najoung Kim, Phu Mon Htut, Thibault F'evry, Berlin Chen, Nikita Nangia, Haokun Liu, Anhad Mohananey, Shikha Bordia, Nicolas Patry, Ellie Pavlick, and Samuel R. Bowman. `jiant` 1.1: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`, 2019.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, 2011.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *ArXiv*, abs/1903.12136, 2019.

Debajyoti Chatterjee. Making neural machine reading comprehension faster. *ArXiv*, abs/1904.00796, 2019.

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

Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. Model compression with multi-task knowledge distillation for web-scale question answering system. *ArXiv*, abs/1904.09636, 2019.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. Small and practical bert models for sequence labeling. In *EMNLP-IJCNLP*, 2019.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *NeurIPS*, 2019.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *ICML*, 2015.