DistilBERT

Daniel Botros, Jenny Chen, Jessica Cho, Eric Gao, Xinyu Ge

https://github.com/danielbotros/DistilBERT/tree/main

Introduction

BERT models have achieved strong performance on many natural language processing tasks due to their bidirectional attention and deep contextual understanding. However, the model contains many parameters, which makes it difficult to deploy in resource-constrained or low-latency environments. This motivates the development of DistilBERT, a smaller, faster, and cheaper version of BERT while maintaining a promising portion of the model's performance. In the paper "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter" by Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf, the researchers proposed using knowledge distillation during pretraining to compress BERT models. The paper's main contribution is showing that DistilBERT can achieve 97% of BERT's performance, while being 40% smaller and 60% faster. Our project aimed to reproduce the results of this paper by training the models using a triple loss distillation function and fine-tuning them on downstream tasks such as sentiment analysis, grammar judgment, and paraphrase detection. In the end, we were able to retain 94.35% of teacher models' performance, averaging across tasks.

Chosen Result

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

We aimed to reproduce the results shown in Table 1 and Table 2 since they demonstrate the core claim of the paper that DistilBERT retains most of BERT's performance (97%) on distilled tasks, while being significantly smaller and faster. We specifically chose to reproduce the results for CoLA, MRPC, and IMDb because these tasks offer a well-rounded evaluation on semantics, grammar, and sentiment, and are publicly available and feasible to run on Google Colab, and represent both GLUE tasks and non-GLUE downstream classification tasks, showcasing DistilBERT's generalization. If these performance claims hold, it validates DistilBERT as a viable replacement for BERT in many real-world applications, especially those with computational constraints.

Methodology

To replicate the DistilBERT results, we closely followed the knowledge distillation approach described in the original paper to compress the BERT base model while maintaining a significant portion of its performance. The teacher model used in our implementation is a standard uncased BERT base, while the student is the pretrained DistilBERT model that came directly from the paper's result. This model removes half of the BERT layers (12 → 6) to reduce size and computational cost, as well as removing token-type embeddings and poolers, which are not essential to the model's architecture, further simplifying the model. However, since we wanted to measure the performance retained by undergoing our own knowledge distillation pretraining, we did not want to use the pre-learned weights that the model is initialized with. Instead, we re-initialized the student model layers with every other BERT layer from the teacher as its initialization, helping it capture similar hierarchical linguistic patterns and giving the model a head start on pretraining. This method was described in the paper. We trained this model for 10 epochs on the wikitext dataset on a masked language modeling (MLM) task, a widely used corpus for language

modeling, over a span of approximately 3 hours on Colab free T4 GPUs in order to distill it. This pretraining process allows the student to imitate the behavior of the teacher using a triple loss objective consisting of a linear combination of three key components: **MLM Loss** consists of the cross-entropy loss between the predicted masked token logits and the actual masked token labels. **KL-Div Loss** measures the divergence between the distribution of the teacher's softened and the student's unnormalized logits. **Cosine Embedding Loss** measures the alignment between the internal hidden states of the teacher and student by finding the cosine similarity between them. This methodology directly aligns with the original paper's approach and serves as the backbone of our experiment, allowing us to assess whether the distilled model can maintain the teacher's performance across a range of downstream tasks like CoLA (grammar judgment), MRPC (paraphrase detection), and IMDb (sentiment classification), on which both the teacher and student models were fine-tuned for evaluation.

Results & Analysis

As shown in the table, we were able to get results for the three downstream tasks that aligned pretty closely with the original paper's findings (our results are blue, the paper's are green). We found that the student retained 94.35% of the teacher's performance (whereas the paper's student retained 97% of their teacher's performance). The paper's DistilBERT slightly outperforms our DistilBERT, likely because the former was distilled using a much larger text corpus. The main challenge we encountered during the re-implementation process was figuring out how to reproduce the distillation pre-training process using limited resources. The original paper's DistilBERT was trained for over 3 days on a massive dataset–a concatenation of Wikipedia and Toronto Book Corpus. Due to our limited resources (both time and computational resources), we decided to use a smaller dataset (wikitext) and restrict the number of downstream tasks to fine-tune our model. As such, our project's results demonstrate the paper's key conclusion that BERT can be downsized to a much smaller model using knowledge distillation without compromising model performance on various language tasks. Knowledge distillation is a powerful technique that can allow for the incorporation/usage of large models in lower-latency, resource-constrained settings.

Model	IMDb (Accuracy)	CoLA (Matthews corr)	MRPC (Acc. / F1)
Teacher	92.4	57.2	87.2
	93.5	56.3	88.6
Student	91.2	48.9	86.2
	92.8	51.3	87.5

Reflections

This project gave us hands-on experience with model compression and the practical challenges of reproducing large-scale NLP models under limited resources. By following the triple loss distillation method outlined in the DistilBERT paper, we were able to train a student model that retained 94.35% of the teacher's performance across IMDb, CoLA, and MRPC, closely aligning with the paper's 97% claim despite using a much smaller dataset and shorter training time. We learned that careful architectural choices—like halving the number of layers and removing poolers/token-type embeddings, can significantly reduce inference time (by 2.6x) and model size (~40%) with only modest performance trade-offs. This highlights the value of distillation for real-world applications where latency and memory constraints matter. One challenge was replicating the pre-training phase without the original massive corpus. Nonetheless, our results validate the effectiveness of knowledge distillation, even under constrained settings. In the future, we hope to explore selective layer removal and ensemble teacher models to improve compression-performance trade-offs.

References

Papers:

- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv. https://arxiv.org/abs/1910.01108

- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. arXiv. https://arxiv.org/abs/1804.07461

Tools and Frameworks:

- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). *Transformers: State-of-the-art Natural Language Processing*. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). *PyTorch: An imperative style, high-performance deep learning library*. *Advances in Neural Information Processing Systems*, 32, 8026–8037. https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

Datasets:

- Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016). *Pointer Sentinel Mixture Models*. arXiv. https://arxiv.org/abs/1609.07843 (for WikiText-2)

- Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. (for IMDb)
- Dolan, W. B., & Brockett, C. (2005). *Automatically constructing a corpus of sentential paraphrases*. *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. (for MRPC)
- Warstadt, A., Singh, A., & Bowman, S. R. (2019). *Neural Network Acceptability Judgments*. *Transactions of the Association for Computational Linguistics*, 7, 625–641. https://aclanthology.org/Q19-1042/ (for CoLA)