DistilBERT: A Distilled Version of BERT

Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf *Hugging Face*

Abstract

As large-scale pre-trained language models become the standard in Natural Language Processing (NLP), operating these models in resource-constrained environments remains a significant challenge. We propose **DistilBERT**, a general-purpose language representation model pre-trained using knowledge distillation. Our results demonstrate that it is possible to reduce the size of a BERT model by **40%**, while retaining **97%** of its language understanding capabilities and improving inference speed by **60%**. To achieve this, we introduce a triple loss function that leverages inductive biases from larger models.

---

1. Introduction

The trend in NLP has shifted toward increasingly larger models (e.g., BERT-Large, RoBERTa). While these models achieve state-of-the-art results, their high computational cost and memory footprint hinder deployment on edge devices like smartphones. We address this by applying **Knowledge Distillation** during the pre-training phase, resulting in a model that is smaller, faster, and cheaper to run, yet remains flexible for a wide range of downstream tasks.

2. Methodology

The core of DistilBERT's efficiency lies in its training regime and architectural modifications.

2.1 Knowledge Distillation

We employ a teacher-student framework where a large model (the Teacher, BERT) transfers its knowledge to a smaller model (the Student, DistilBERT). The student learns by mimicking the **soft targets** (probability distributions) of the teacher.

2.2 The Triple Loss Function

To optimize the student, we minimize a combined loss function:

$$Loss = L_{mlm} + L_{ce} + L_{cos}$$

- **L_{mlm} (Masked Language Modeling Loss):** Standard BERT loss to learn linguistic patterns.
- **L_{ce} (Distillation Loss):** Calculated using soft targets with a temperature $T > 1$:
  $$Softmax(z_i, T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$
- **L_{cos} (Cosine Embedding Loss):** Aligns the hidden state vectors of the student with the teacher.

2.3 Architecture & Initialization

- **Layers:** Reduced from 12 (BERT-Base) to 6.
- **Initialization:** The student is initialized by taking every second layer from the teacher.

- **Optimizations:** Removed token-type embeddings and the pooler.

---

3. Experimental Results

We evaluated DistilBERT on the GLUE benchmark and downstream tasks like SQuAD.

3.1 Performance Comparison

Model	GLUE Score	Parameters	Inference (CPU)
BERT-Base	79.5	110M	649 ms
DistilBERT	77.0	66M	410 ms
ELMo	71.0	180M	895 ms

3.2 Key Findings

- **Accuracy:** DistilBERT retains **97%** of BERT-Base performance.
- **Speed:** On-device (mobile) tests show DistilBERT is **71% faster** than the original BERT.
- **Training:** DistilBERT was trained on 8 V100 GPUs for roughly 90 hours, significantly less than the original BERT.

---

4. Conclusion

DistilBERT proves that knowledge distillation is an effective tool for model compression in the transformer era. By reducing parameters and latency without a significant drop in accuracy, we enable the use of state-of-the-art NLP on edge devices.

5. Implementation (Hugging Face)

The model is available via the transformers library:

```python
from transformers import AutoModel, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
model = AutoModel.from_pretrained("distilbert-base-uncased")
```