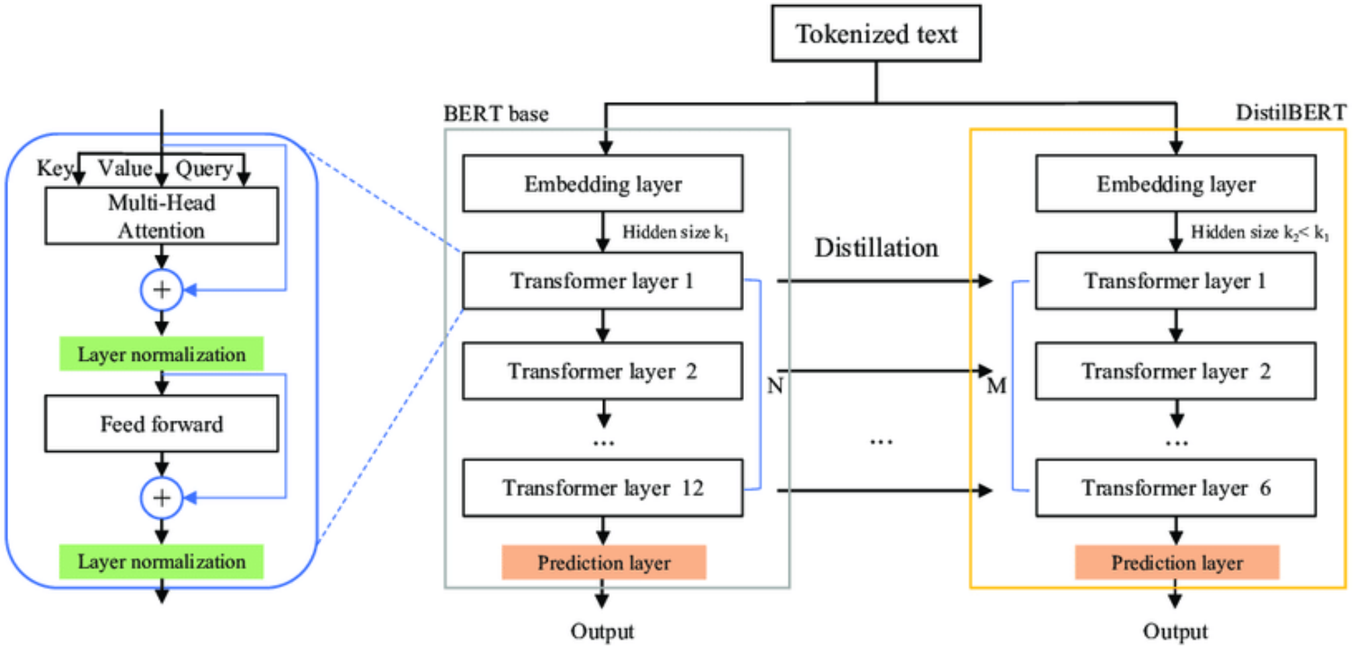


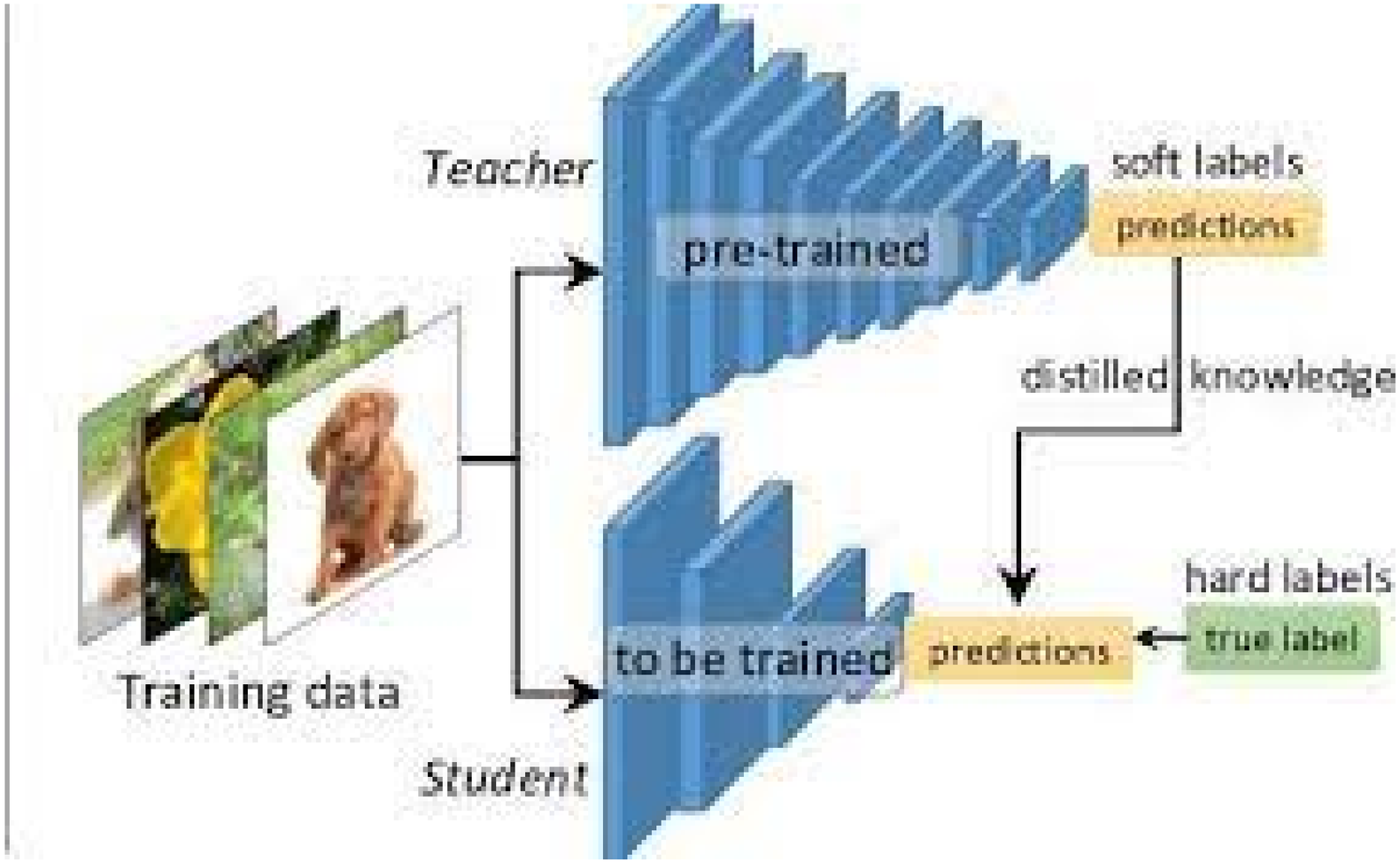
DistilBERT

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

	BERT	RoBERT	DistilBERT	XLNet
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base:** 66	**Base**: ~110 **Large**: ~340
Training Time	**Base**: 8 x V100 x 12 days* **Large:** 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large:** 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.	**Large:** 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	5% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	**Base**: 16 GB BERT data **Large**: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

DistilBERT was able to **reduce the size of a BERT model by 40%**, while **retaining 97% of its language understanding** capabilities and being **60% faster.**

Knowledge Distillation



Knowledge Distillation

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.

Kullback–Leibler divergence

$$KL(p||q) = \mathbb{E}_p(log(\frac{p}{q})) = \sum_i p_i * log(p_i) - \sum_i p_i * log(q_i)$$

Reference

- https://www.youtube.com/watch?v=rNOuDKWtrAE
- https://medium.com/huggingface/distilbert-8cf3380435b5