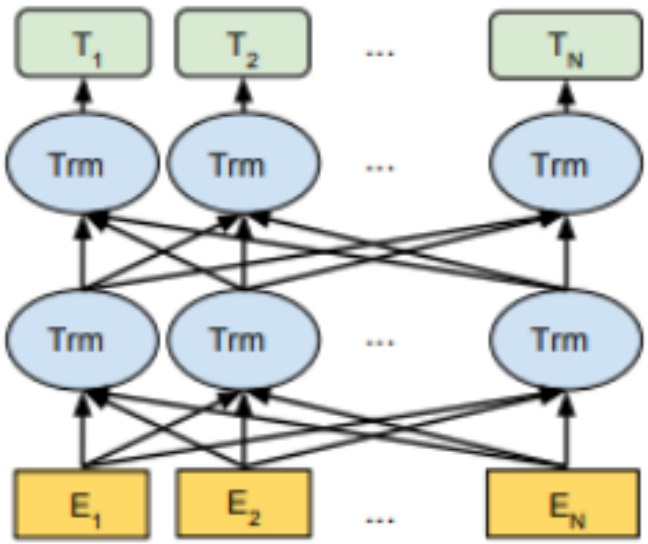
DistilBERT

Kostek Subbotko

BERT

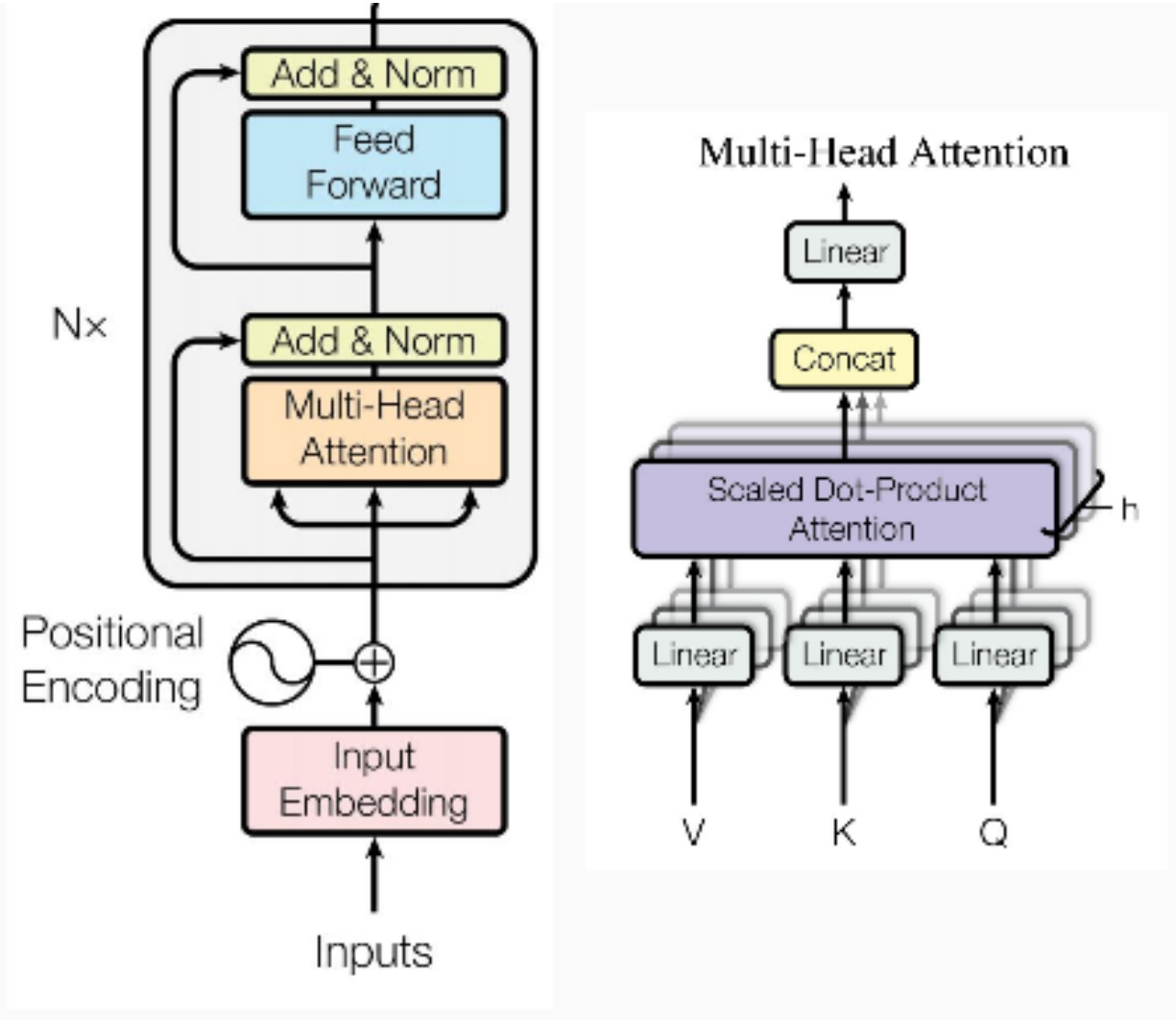
- BERT - **B**idirectional **E**ncoder **R**epresentations from **T**ransformer
- Architecture: multi-layer bidirectional Transformer encoder
- Pre-trained using two unsupervised tasks: MaskedLM and Next Sentence Prediction(NSP)
- Fine-tuned on downstream tasks



BERT Architecture

Transformer encoder

- Based on *"Attention is All you need"* paper



Masked LM

- Problem: Previous language models used only unidirectional context, but language understanding is bidirectional
- Solution: Mask 15% of the input words and ask network to predict the masked words:
    - my dog is hairy $\rightarrow$ my dog is [MASK]
- Too little masking: Too expensive to train
- Too much masking: Not enough context

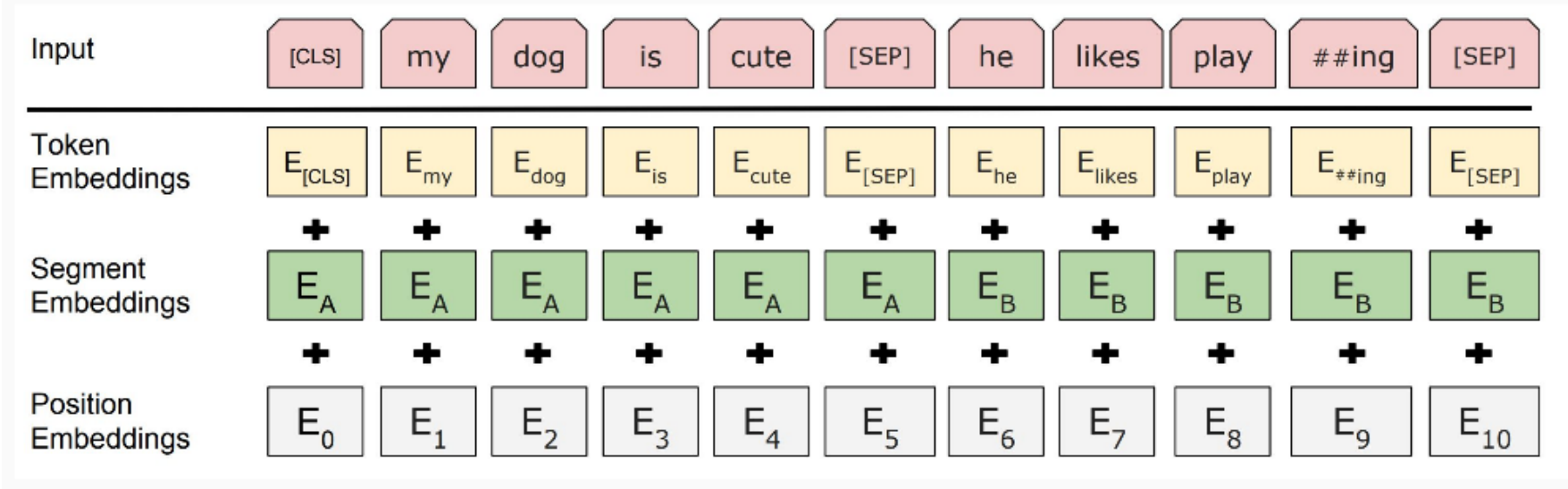
Masked LM

- Problem: Mask token never seen at fine-tuning
- Solution: Don't replace with mask 100% of the time. Instead:
- 80% of the time replace with [MASK]
  my dog is hairy $\rightarrow$ my dog is [MASK]
- 10% of the time replace with random word
  my dog is hairy $\rightarrow$ my dog is apple
- 10% of the time don't do anything
  my dog is hairy $\rightarrow$ my dog is hairy
- Effect: Model does not know which words it will be asked to predict so it is forced to keep representation of every input token

Next Sentence Prediction

- When choosing the sentences A and B, 50% of the time B is the actual next sentence that follows A, and 50% of the time it is a random sentence from the corpus:
- Input=[CLS] the man went to [MASK] store [SEP]he bought a gallon [MASK] milk [SEP]
  Label=IsNext
- Input=[CLS] the man [MASK] to the store [SEP]penguin [MASK] are flight ##less birds [SEP]
  Label=NotNext
- Token [CLS] is the special symbol for classification output

BERT - Input


Input
[CLS] my dog is cute [SEP] he likes play ##ing [SEP]
Token Embeddings
E[CLS] Emy Edog Eis Ecute E[SEP] Ehe Elikes Eplay E##ing E[SEP]
Segment Embeddings
EA EA EA EA EA EA EB EB EB EB EB
Position Embeddings
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10

BERT - Results

- GLUE Test results:

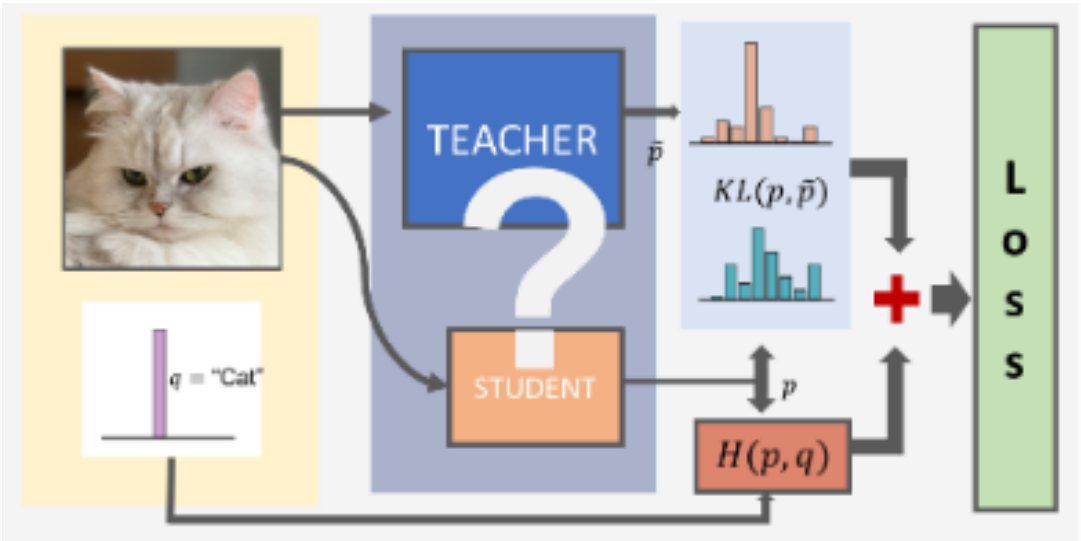
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

- Problem: size

Model	Hidden layers	Hidden unit size	Attention heads	Feedforward filter size	Max sequence length	Parameters
BERTBASE	12 encoder	768	12	4 x 768	512	110M
BERTLARGE	24 encoder	1024	16	4 x 1024	512	330M

Knowledge Distillation

- Teacher-Student training: compact model is trained to reproduce the distribution of larger model
- Idea: use soft targets instead of class labels
- Teacher's probabilities reveal more information
- In one-hot encoding "similar" classes are orthogonal
- Hinton(2015): use softmax-temperature $p_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$



DistilBERT

- Pre-trained with knowledge distillation
- Same architecture as BERT(base), number of layers reduced by a factor of 2
- Initialized from the teacher by taking one layer out of two
- Loss function consists of L_{ce}, L_{mlm}, L_{cos}
- $L_{ce} = \sum_i t_i * log(s_i)$
- L_{mlm} - masked language modeling loss
- L_{cos} - cosine embedding loss
- Implementation: https://github.com/huggingface/transformers

DistilBERT - performance

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

DistilBERT - size and inference speed

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

- 40% smaller, 60% faster at interference

DistilBERT - ablation study

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

- Removing the Masked Language Modeling loss has little impact while the two distillation losses account for a large portion of the performance