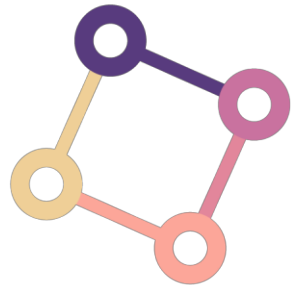
AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Taehee Kim

2020.10.06

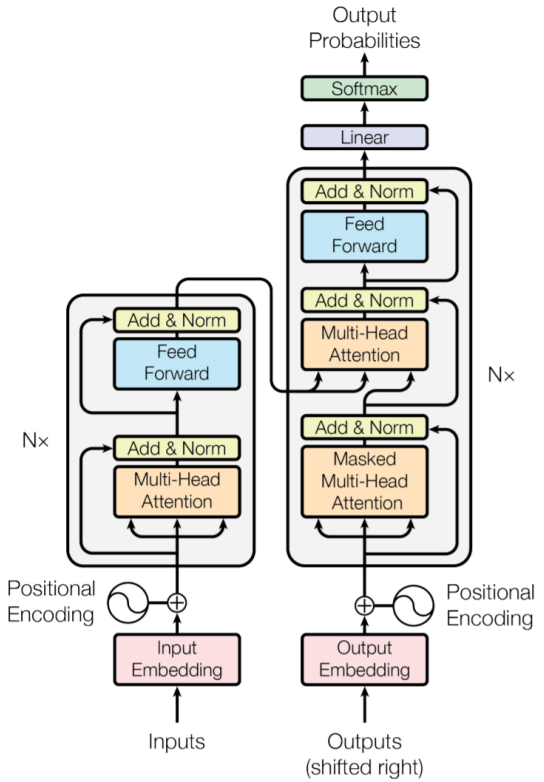


DAVIAN

Data and Visual Analytics Lab

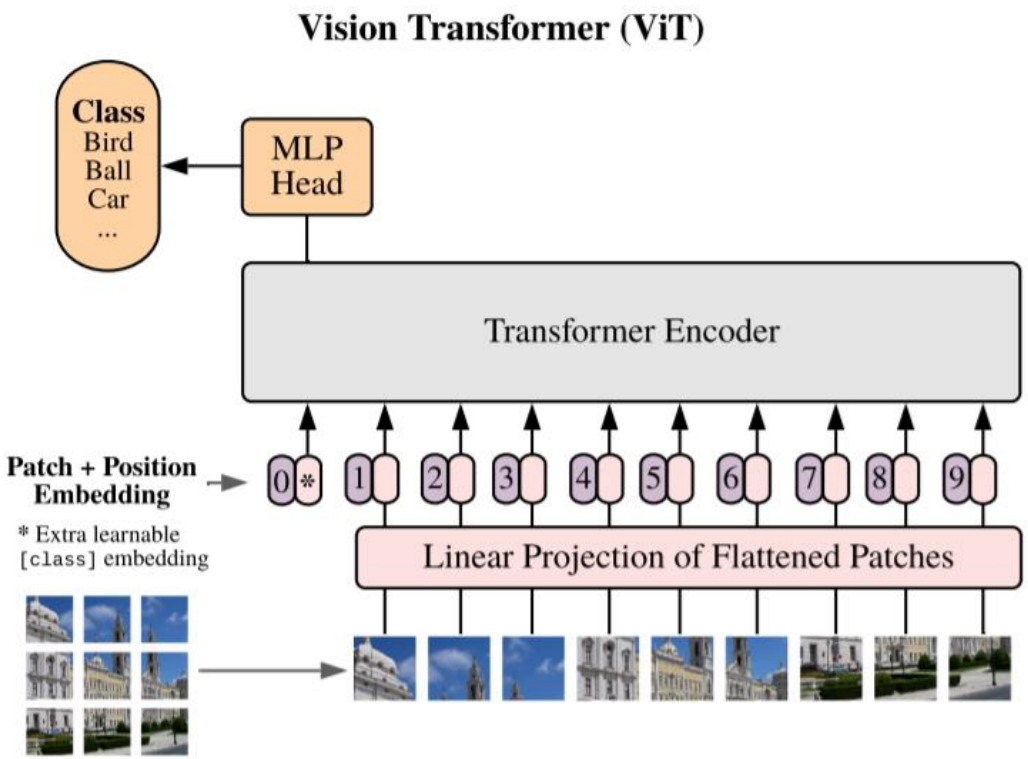
Transformer

- Self-attention based architectures have become the model of choice in natual language processing but, in computer vision, convolutional architectures remain dominant.
- Inspired by the success in NLP, we experiment with applying a standard Transformer directly to images.
- We split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer.



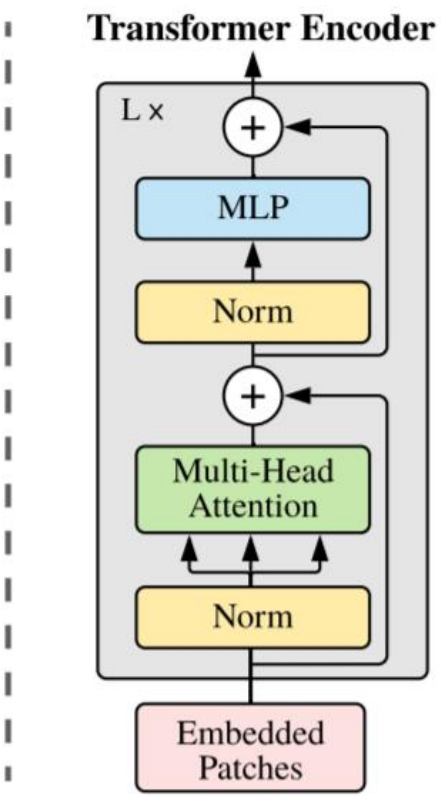
Vision Transformer

- Reshape the image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$



Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Configuration of our different model variants.



$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos},$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1},$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell,$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$

$$\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \quad (1)$$

$$\ell = 1 \ldots L \quad (2)$$

$$\ell = 1 \ldots L \quad (3)$$

$$(4)$$

```python
class SublayerConnection(nn.Module):
    """
    A residual connection followed by a layer norm.
    Note for code simplicity the norm is first as opposed to last.
    """
    def __init__(self, size, dropout):
        super(SublayerConnection, self).__init__()
        self.norm = LayerNorm(size)
        self.dropout = nn.Dropout(dropout)

    def forward(self, x, sublayer):
        "Apply residual connection to any sublayer with the same size."
        return x + self.dropout(sublayer(self.norm(x)))
```

Experiments

- Pre-training Datasets
  - ILSVRC-2012 ImageNet with 1k classes and 1.3M images
  - ImageNet-21k with 21k classes and 14M images
  - JFT with 18k clsses and 303M images

	Ours (ViT-H/14)	Ours (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.36	87.61 ± 0.03	87.54 ± 0.02	$88.4/\mathbf{88.5}^{*}$
ImageNet ReaL	**90.77**	90.24 ± 0.03	90.54	90.55
CIFAR-10	$\mathbf{99.50} \pm 0.06$	99.42 ± 0.03	99.37 ± 0.06	–
CIFAR-100	$\mathbf{94.55} \pm 0.04$	93.90 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	$\mathbf{97.56} \pm 0.03$	97.32 ± 0.11	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	$\mathbf{99.74} \pm 0.00$	99.63 ± 0.03	–
VTAB (19 tasks)	$\mathbf{77.16} \pm 0.29$	75.91 ± 0.18	76.29 ± 1.70	–
TPUv3-days	2.5k	0.68k	9.9k	12.3k

Experiments

- Comparison to state of the art

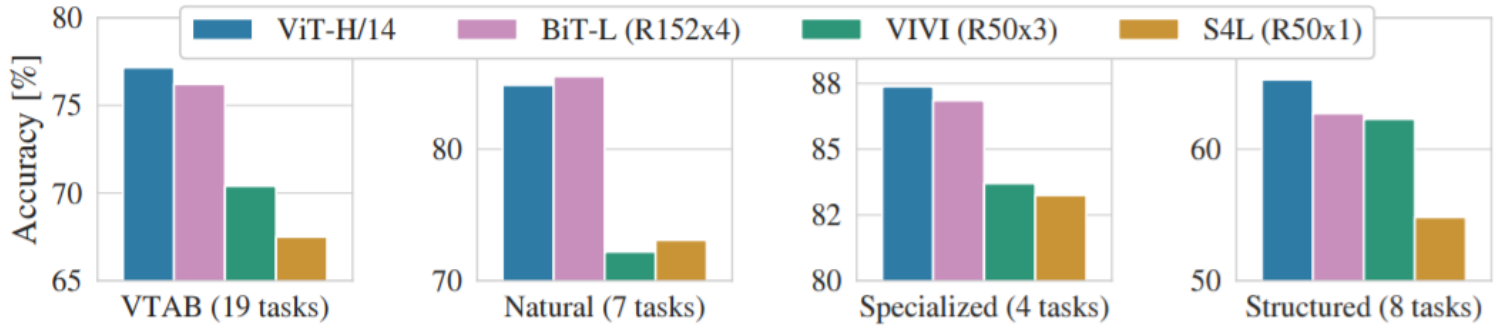


Figure 2: Breakdown of VTAB performance in *Natural*, *Specialized*, and *Structured* task groups.

Experiments

- Pre-training dataset, samples

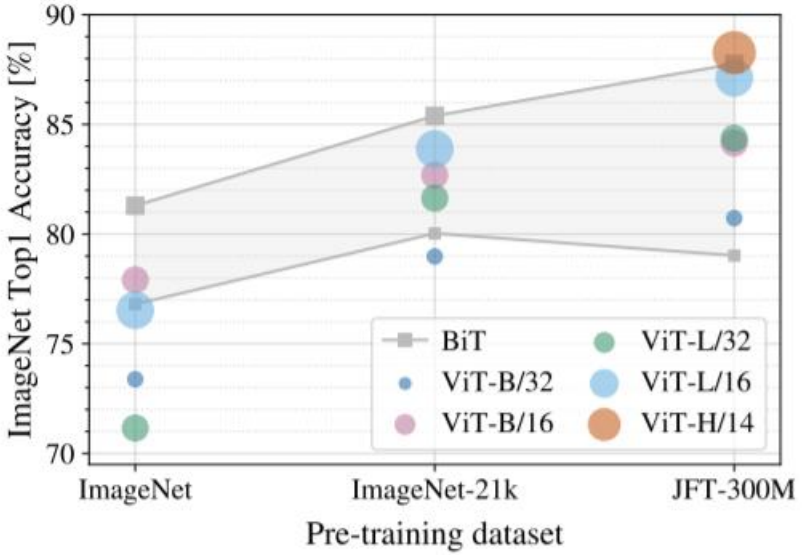


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

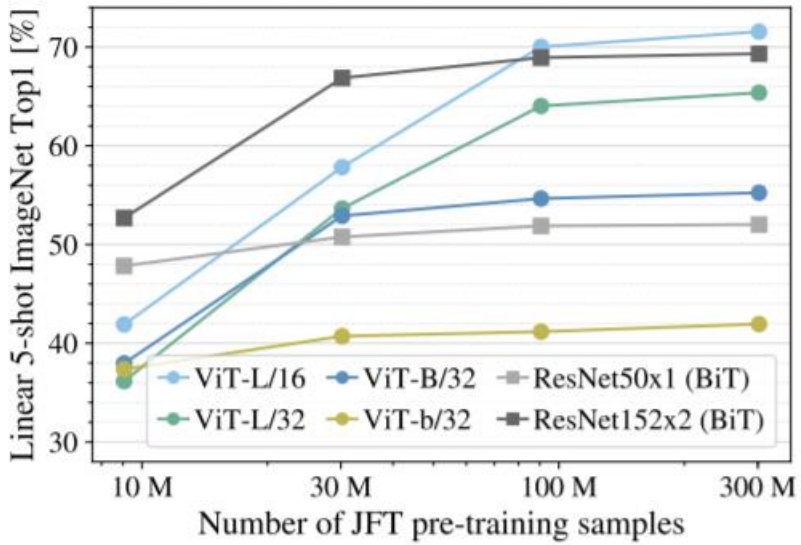


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

Experiments

- Performance versus cost for different architectues

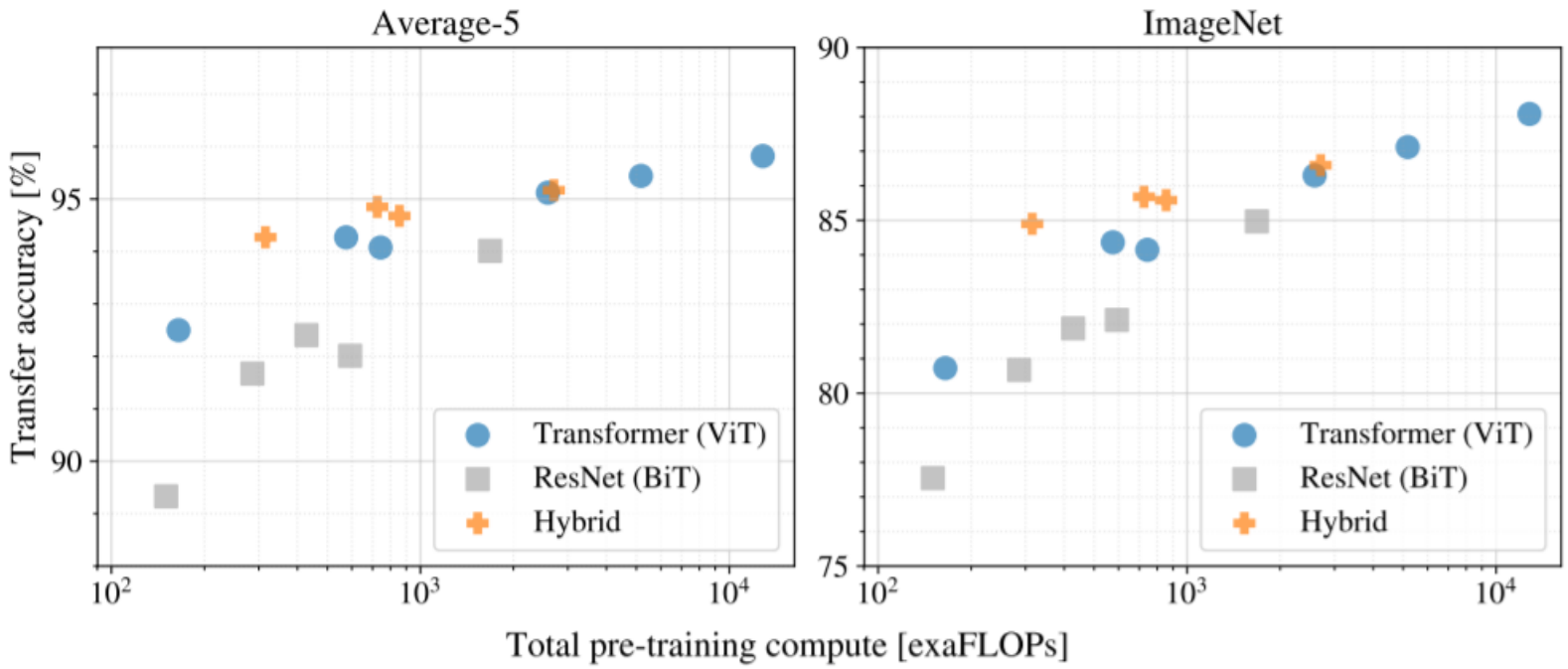


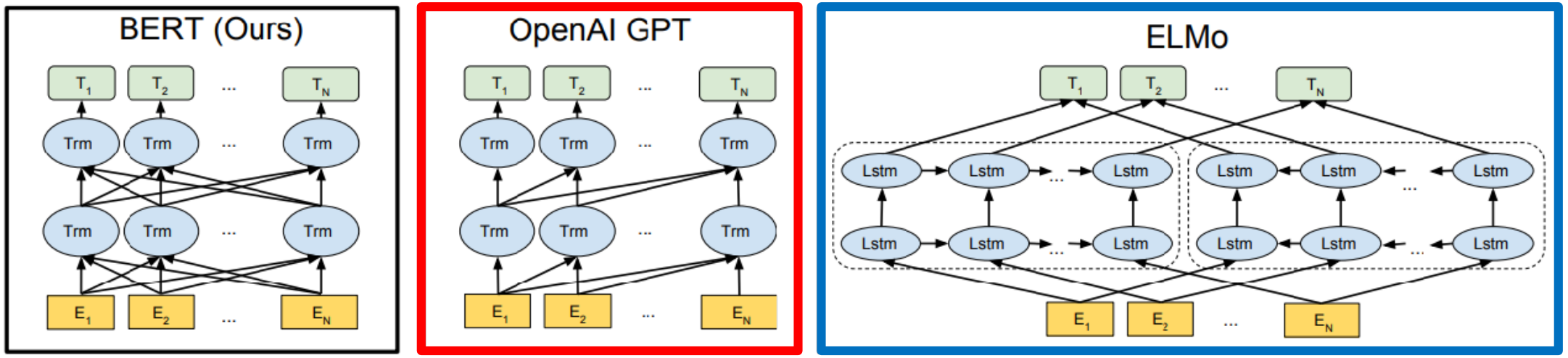
Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanished for larger models.

APPENDIX

Transformer

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

- Learn through masked language modeling tasks
- Use large-scale data and large-scale model



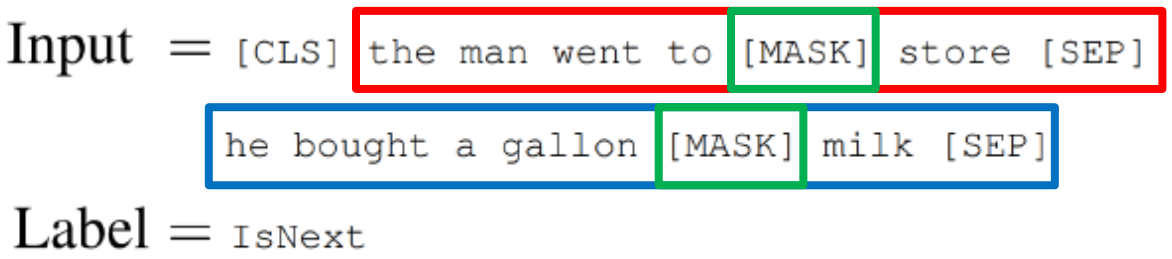
Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

BERT

Pretraining tasks

- Masked Language Model (MLM)
  - Mask some percentage of the input tokens at random, and then predict those masked tokens.
  - 15% of the words to predict
    - 80% of the time, replace with [MASK]
    - 10% of the time, replace with a random word
    - 10% of the time, keep the sentence as same

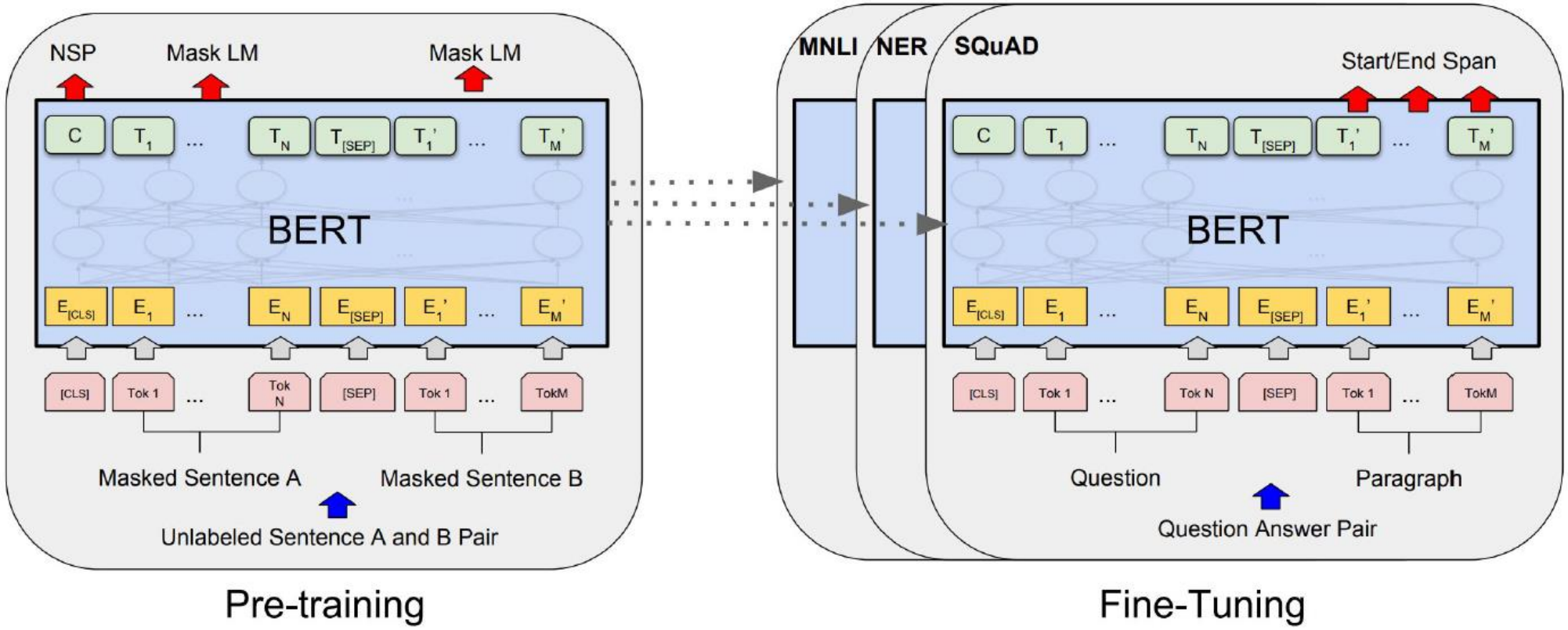
- Next Sentence Prediction (NSP)
  - Predict whether Sentence B is an actual sentence that proceeds Sentence A, or a random sentence



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

BERT

Transfer Learning



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