Distinguished Engineering

BERT
- Bidirectional Encoder Representations from Transformers

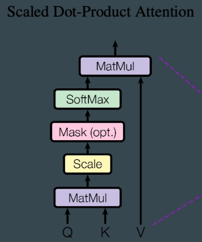
•••

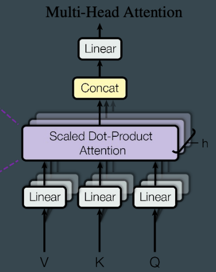
BW

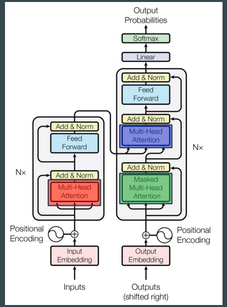
Plan

- Transformer, recap
- **Pre-traing entree**
- BERT, the main dish.
- BERT, dessert.

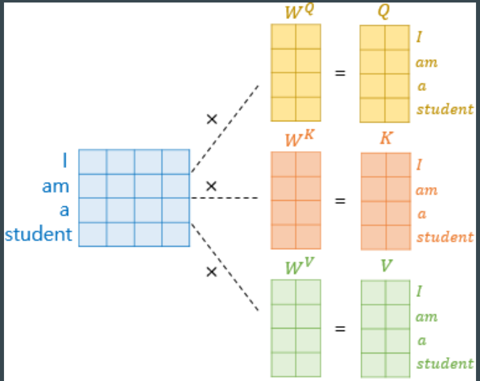
Transformer, recap

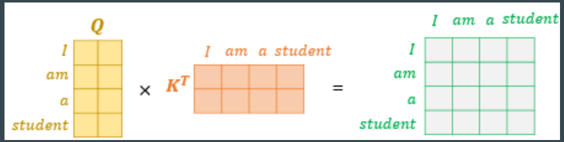


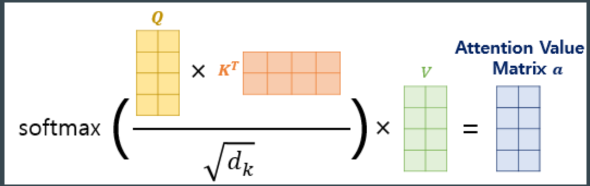




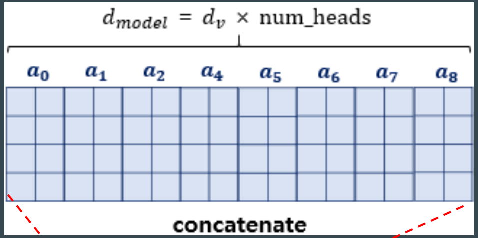
- Multi-head Attention

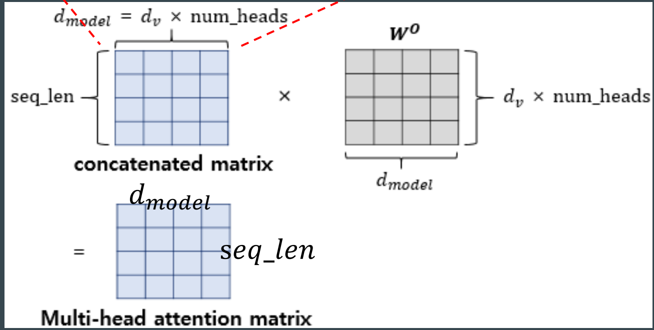




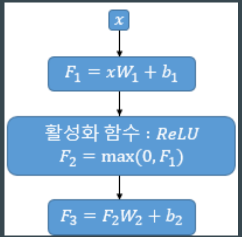


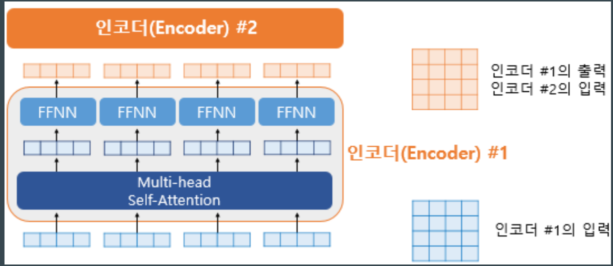
$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$





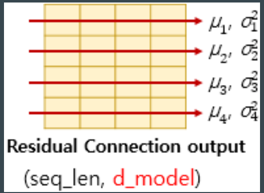
- FFNN





Transformer, recap

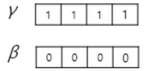
- Residual connection, Layer Normalization

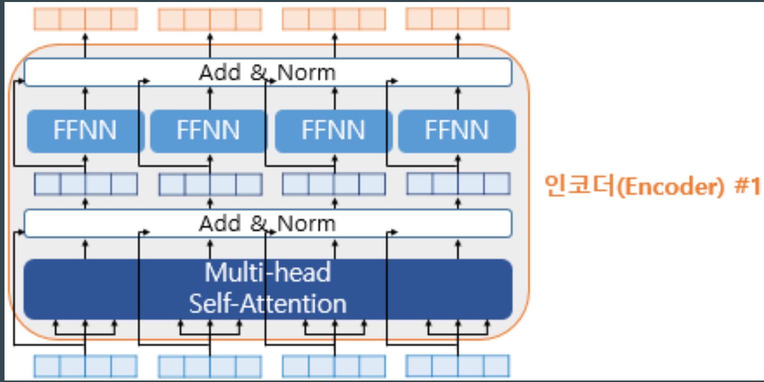


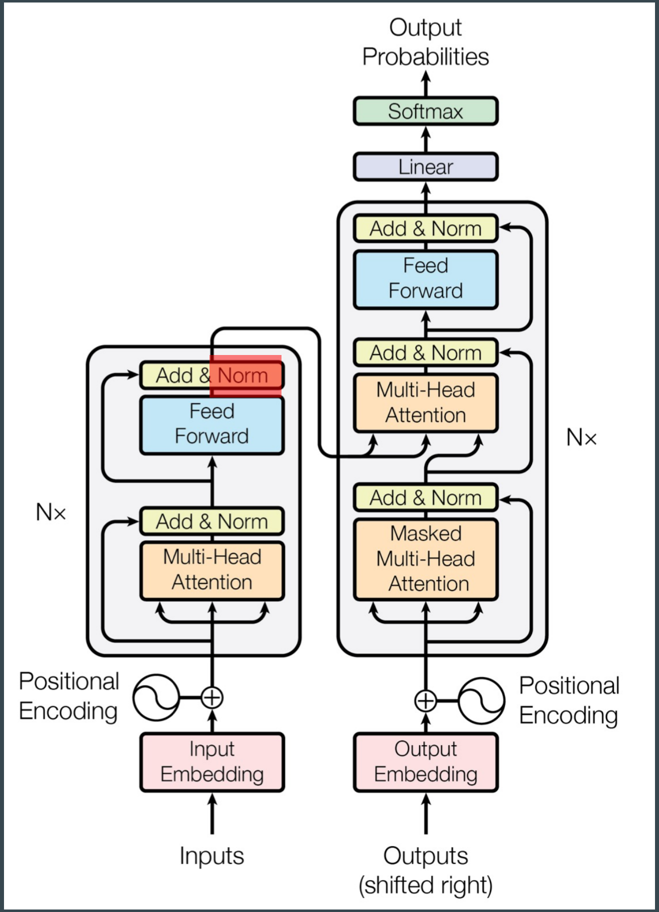
$$ln_i = LayerNorm(x_i)$$

$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

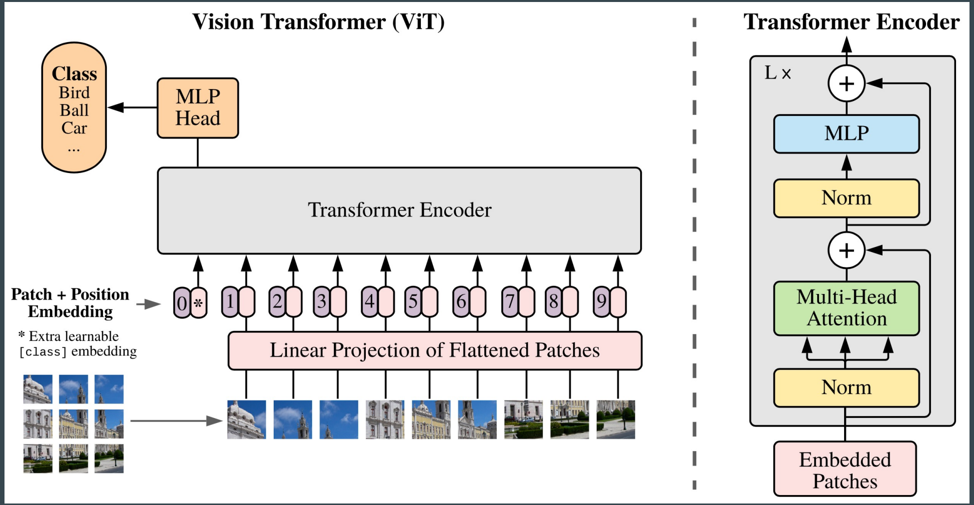
$$ln_i = \gamma \hat{x}_i + \beta = LayerNorm(x_i)$$







An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR2021



The MLP contains two layers with a GELU non-linearity.

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E}; \cdots ;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2\cdot C)\times D},\ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D} \tag{1}$$

$$\mathbf{z'}_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

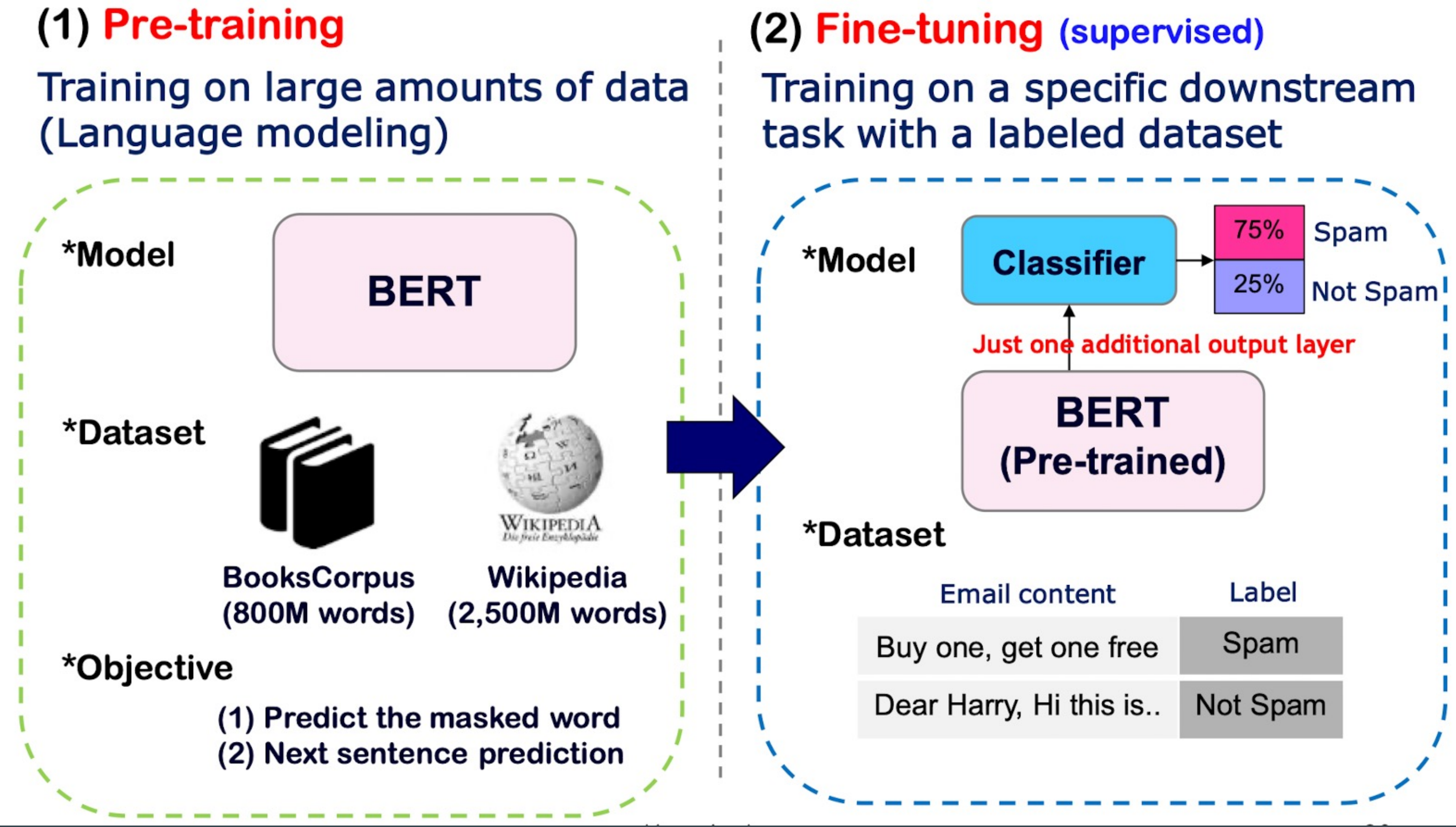
$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0) \tag{4}$$

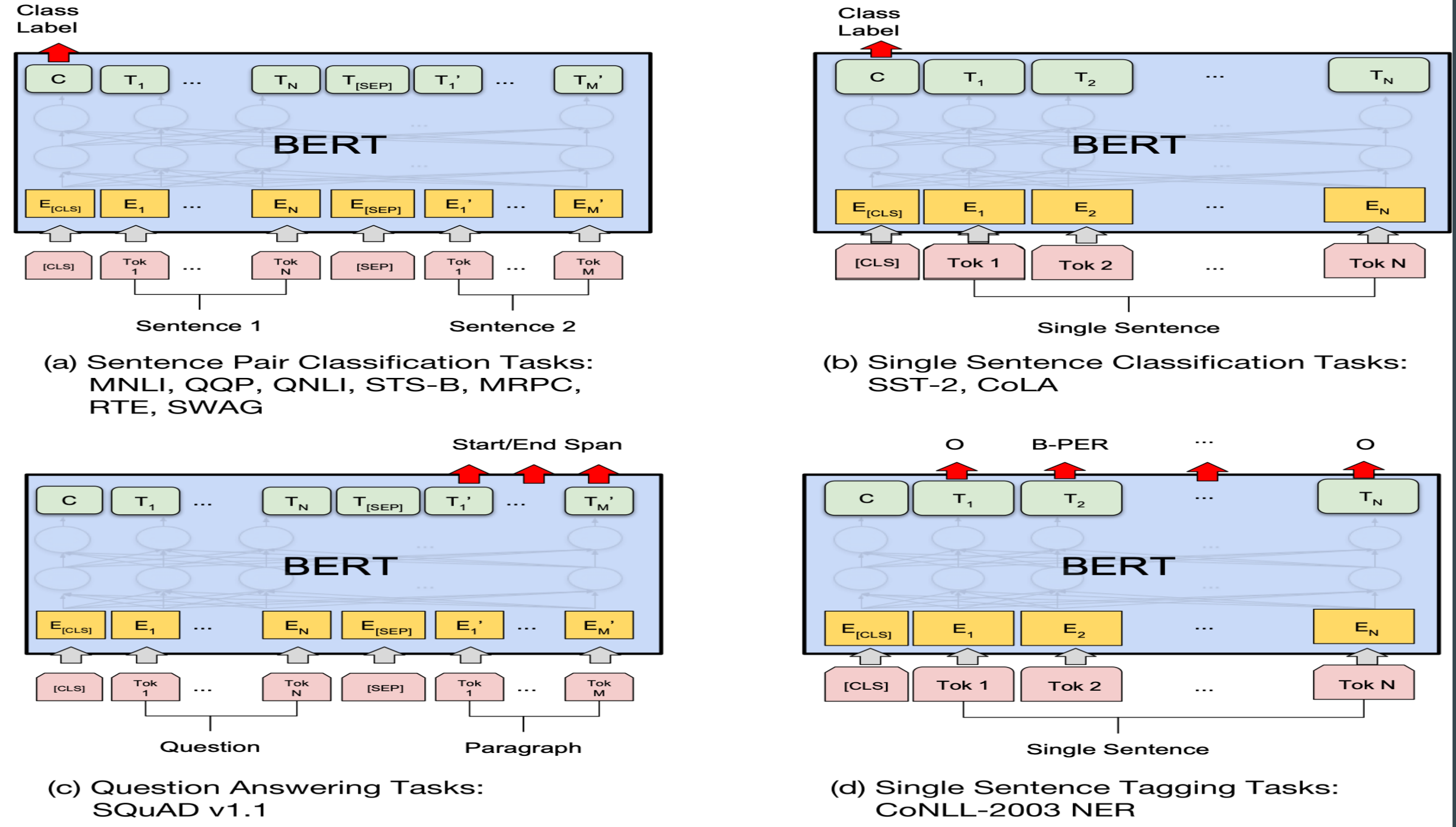
Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

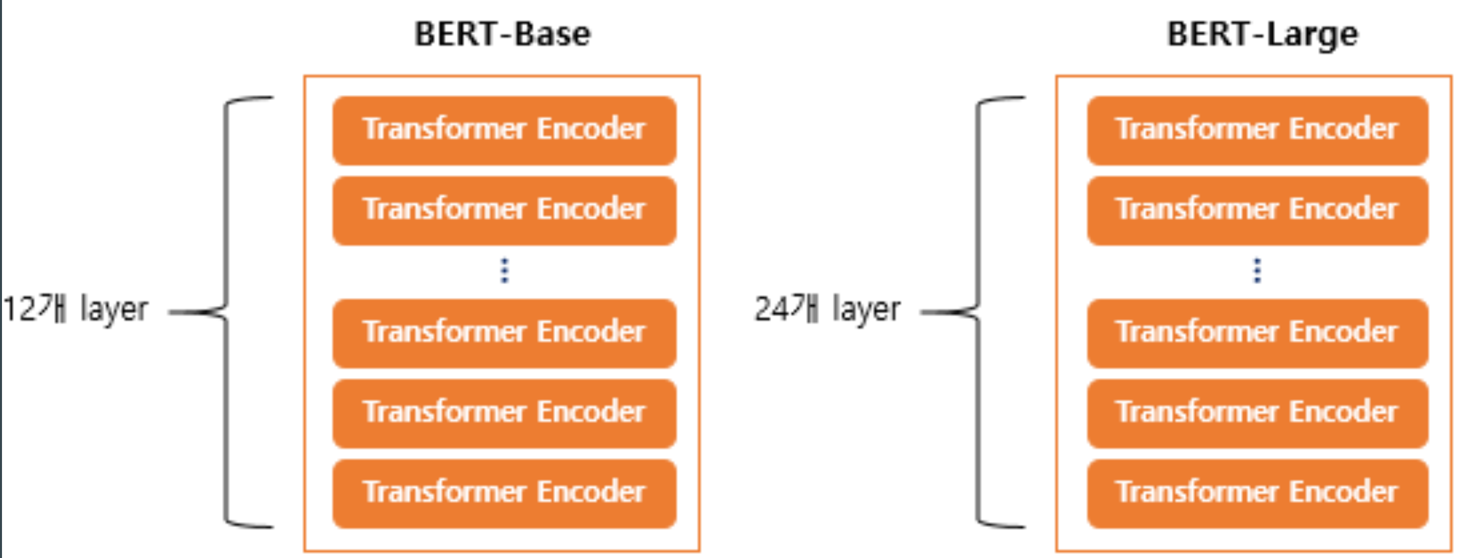
Warm-ups, Language Modeling Process



Warm-ups , pre-training vs fine-tuning

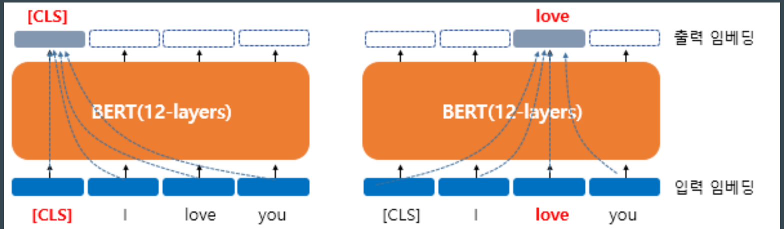


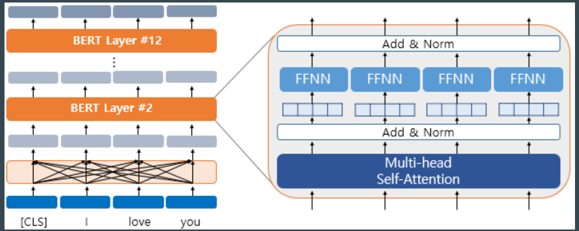
BERT

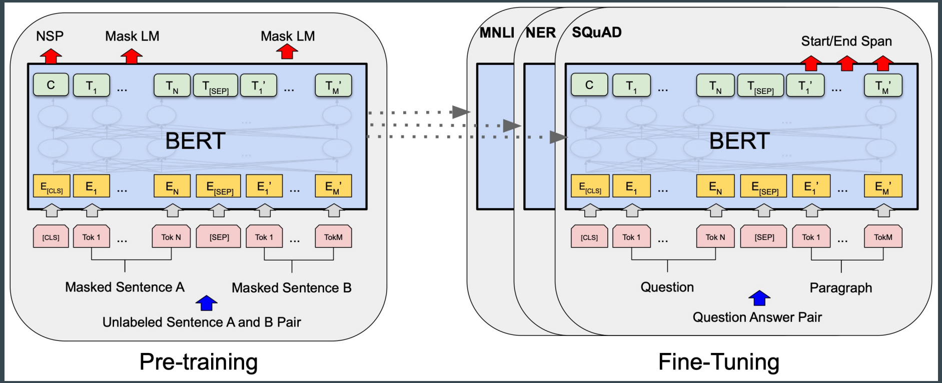


	L (# of trmlayers)	d_model	num_heads	# of Parameters
Transformer base	6	512	8	65M
Transformer big	6	1024	16	213M
BERT base	12	768	12	110M
BERT large	24	1024	16	340M

BERT:

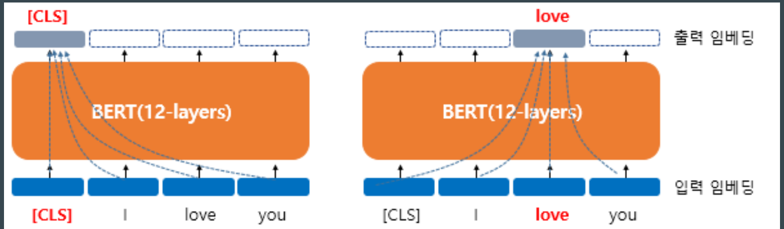

[CLS]
love
BERT(12-layers)
BERT(12-layers)
출력 임베딩
입력 임베딩
[CLS] I love you
[CLS] I love you

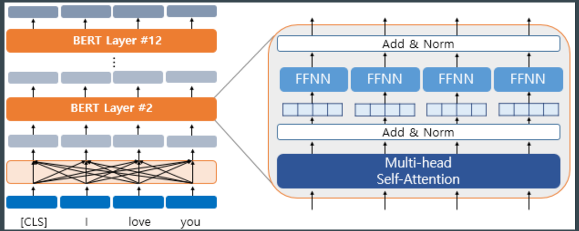

BERT Layer #12
BERT Layer #2
[CLS] I love you
Add & Norm
FFNN FFNN FFNN FFNN
Add & Norm
Multi-head
Self-Attention

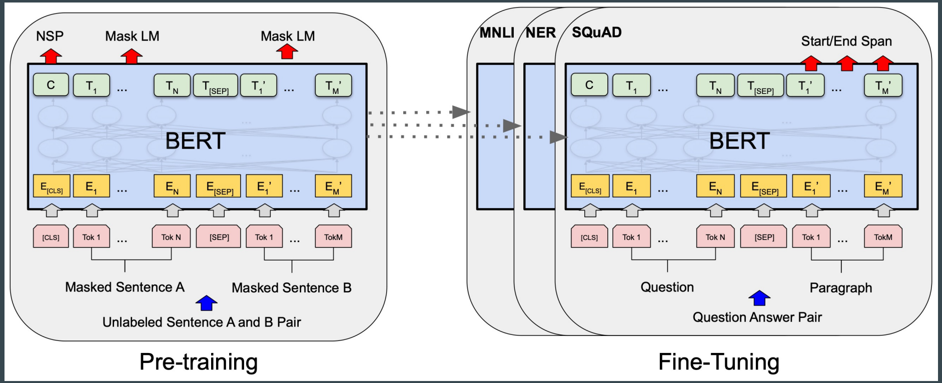

NSP
Mask LM
Mask LM
BERT
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
BERT
Question
Paragraph
Question Answer Pair
Fine-Tuning

BERT: Bidirectional Encoder Representation from Transformer

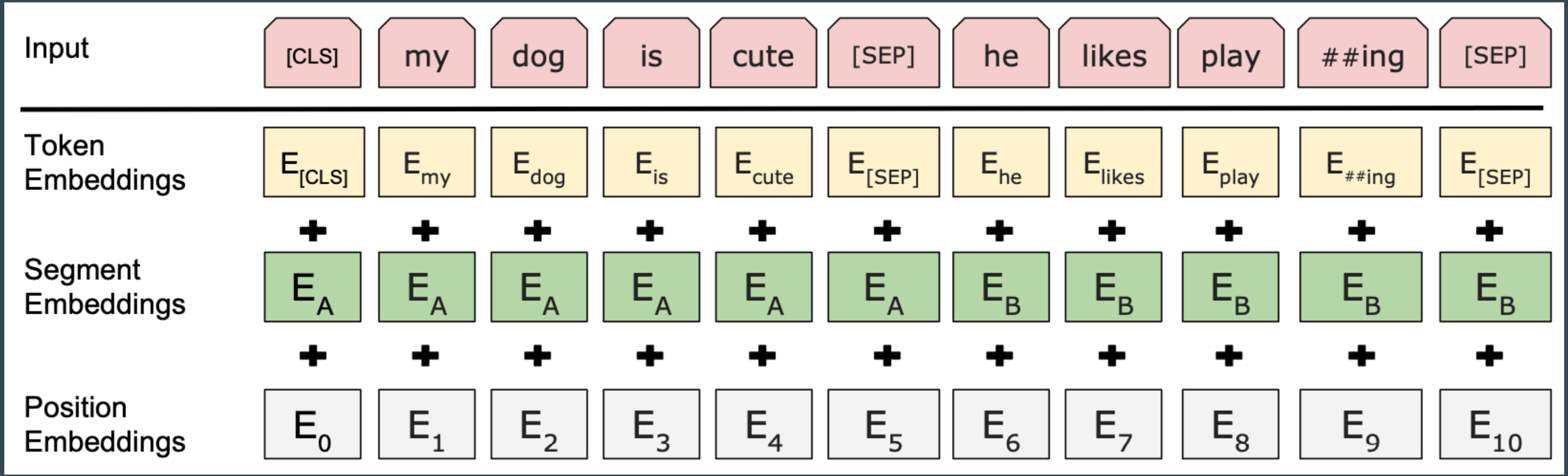
- “BERT is designed to pre- train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers ”

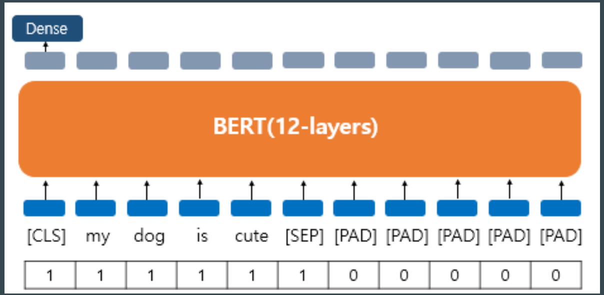






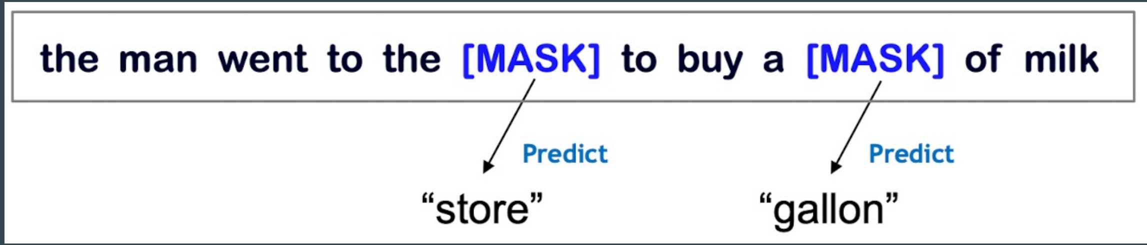
BERT: Embeddings



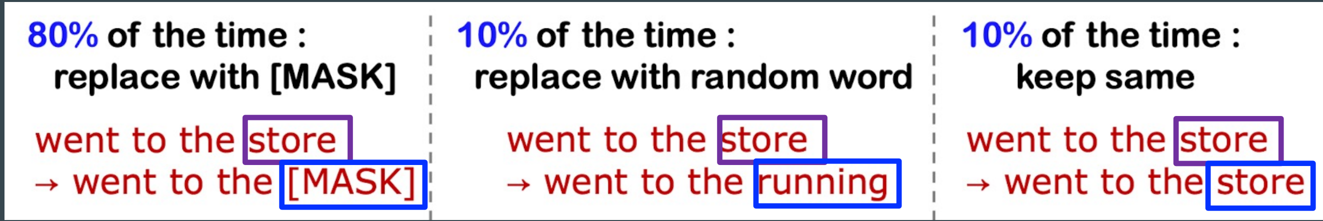


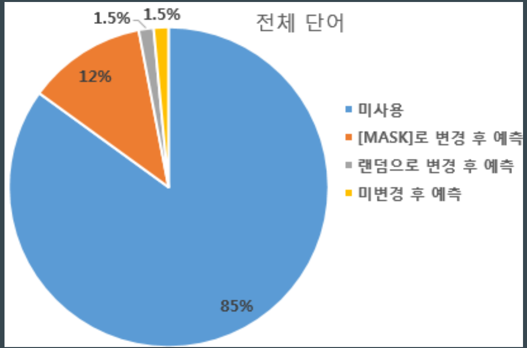
BERT: pre-training #1 Masked Language Model (MLM)

- Mask out 15% of the input words, and then predict the masked word



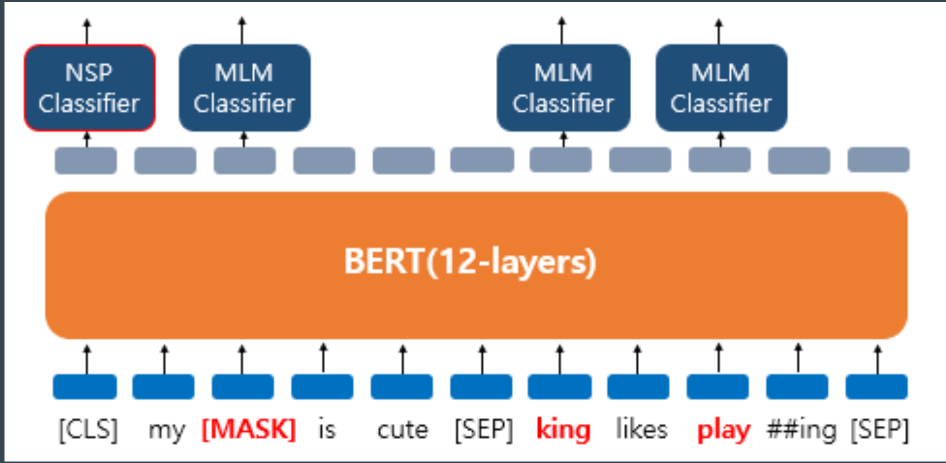
- But [MASK] token will be naver seen at fine tuning.
  →Mismatch pre vs fine tuning
- Solution: out of 15%





BERT: pre-training #2 Next Sentence Prediction (NSP)

- 두개의 문장을 준 후에 이 문장이 이어지는 문장인지 아닌지 맞추는 방식으로 훈련
- 50:50의 비율로 실제 이어지는 문장과 랜덤으로 이어 붙인 문장을 주고 학습



- 이어지는 문장의 경우

  Sentence A : The man went to the store.

  Sentence B : He bought a gallon of milk.

  Label = IsNextSentence

- 이어지는 문장이 아닌 경우 경우

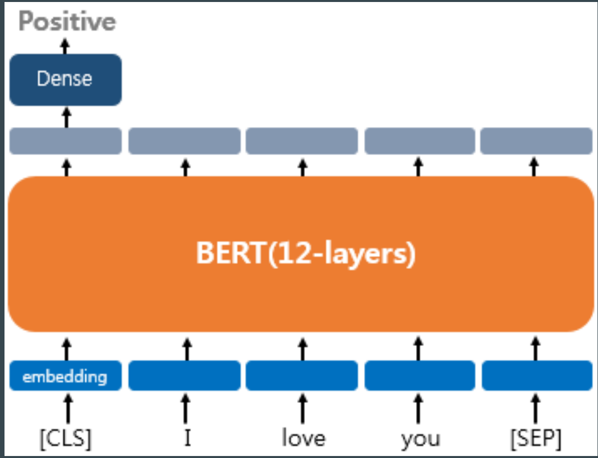
  Sentence A : The man went to the store.

  Sentence B : dogs are so cute.

  Label = NotNextSentence

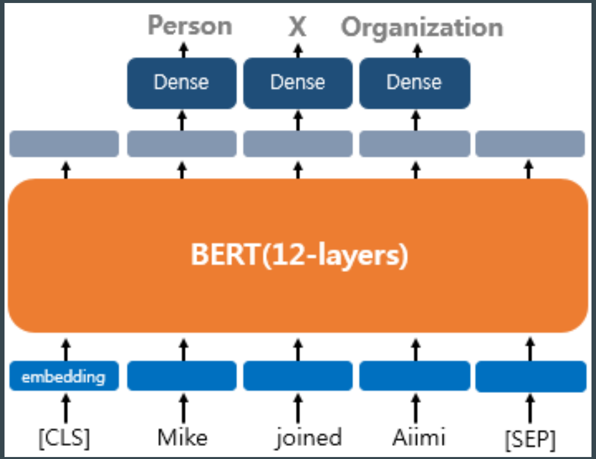
BERT: fine-tuning

1) 하나의 텍스트에 대한 텍스트 분류 (Single Text Classification)
    - 영화리뷰, 감성분석, 뉴스분류



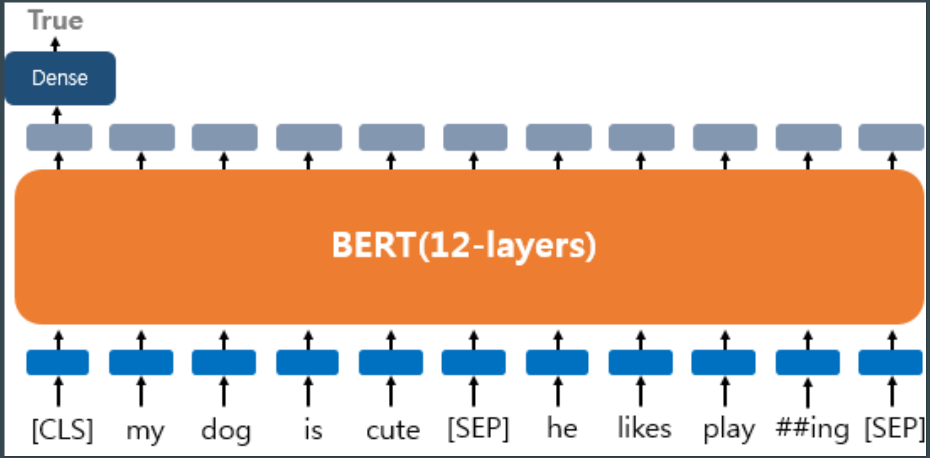
BERT: fine-tuning

2) 하나의 텍스트에 대한 태깅 (Tagging)
   - 품사, 소속



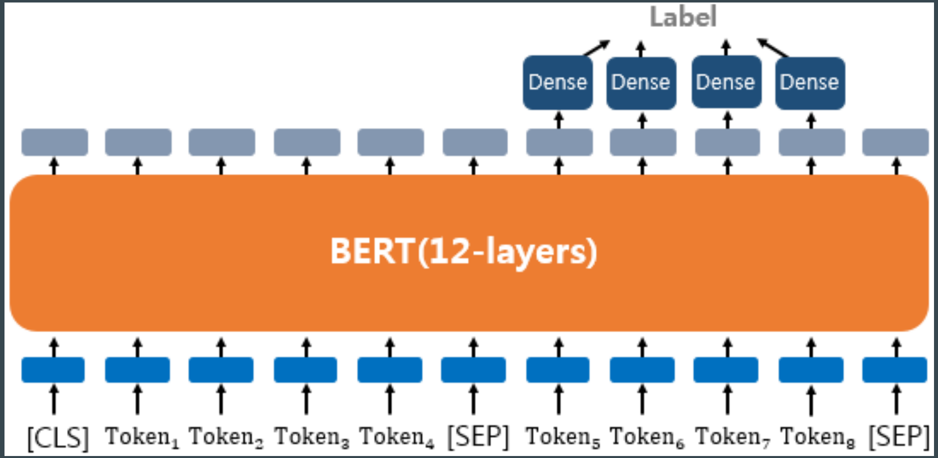
BERT: fine-tuning

3) 텍스트 쌍에 대한 분류 또는 회귀 (Text pair Classification or Regression)
   - 모순 관계(contradiction), 함의 관계 (entailment), 중립 관계 (neutral)



BERT: fine-tuning

4) 질의 응답 (Question Answering)
   - 질문과 본문 입력 → 본문의 일부분을 추출해서 대답



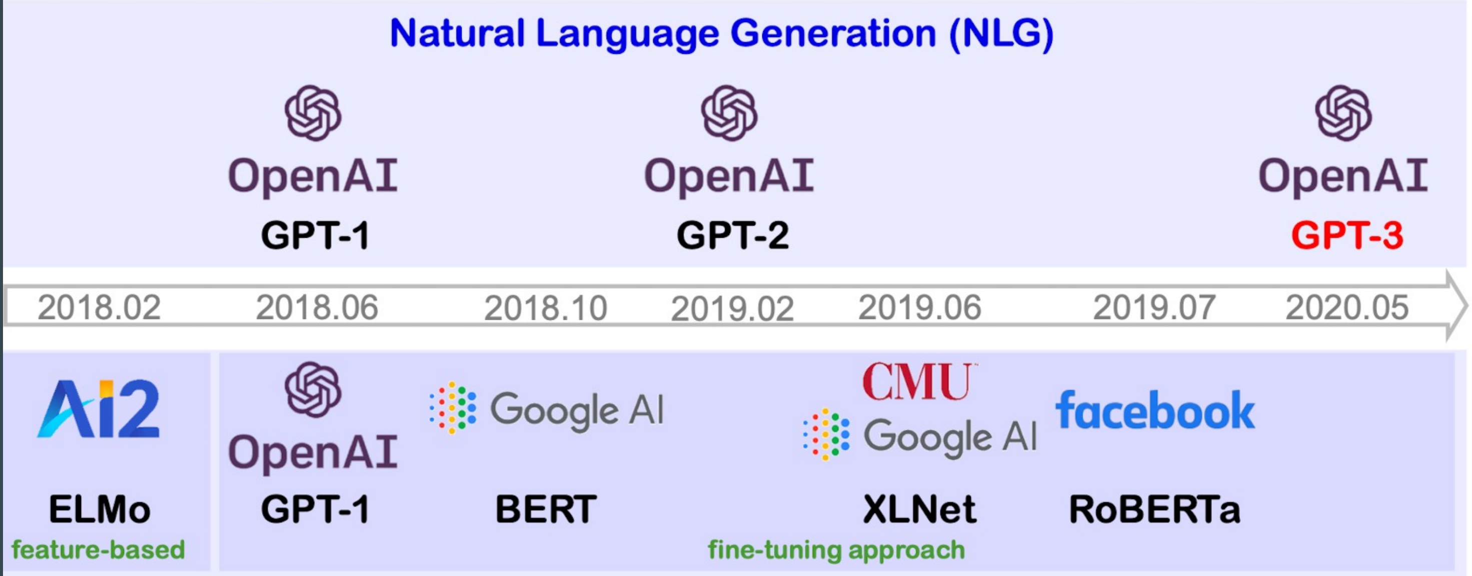
A: 중력

Q: 강우가 떨어지도록 영향을 주는 것은?

본문: 기상학에서 강우는 대기 수증기가 응결되어 중력의 영향을 받고 떨어지는 것을 의미합니다. 강우의 주요 형태는 이슬비, 비, 진눈깨비, 눈, 싸락눈 및 우박이 있습니다.

GPT: What is Different from Others

- **Generation VS Understanding**
  - OpenAI's GPT is an **unidirectional** Language Model(LM)
    - GPT is good for text generation tasks because of the auto-regressive LM
  - On the other hand, BERT and XLNet are **bidirectional** LMs
    - They are good for natural language understanding (NLU) tasks



GPT: Generative Pre-Training

A. Radford et al., “Improving Language Understanding by Generative Pre-Training”

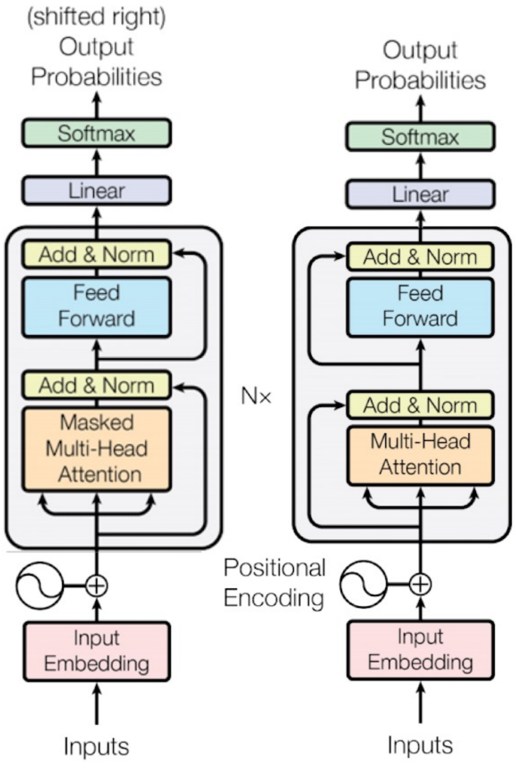
- **Significance of the first GPT**
  (known as GPT-1 now)
  - The first successful model of the pre-training and (then) fine-tuning approach using large model with large corpus
  - GPT-1 outperforms the previous state-of-the-arts on 9 out of 12 tasks

- **Two phase of training GPT-1**
  (similar to BERT)
  - Pre-training:
    LM is trained to predict the next word using the previous context, which is an auto-regressive (generative) language modeling
  - Fine-tuning:
    Almost all layers of pre-trained LM is transferred into any downstream task with minimal task-specific modification

GPT: Comparison with BERT

- **Pre-training objective**
  - GPT: "Next Word Prediction"
    auto-regressive language modeling
  - BERT: "Masked Word Prediction"
    masked language modeling
    + "Next Sentence Prediction"

- **Performances on NLU tasks**
  - ELMo < GPT-1:
    - The betterment of fine-tuning approach than feature-based approach
  - GPT-1 < BERT:
    - In natural language understanding tasks, there is a fundamental limitation of auto-regressive language modeling:
    - GPT-1 uses only unidirectional context, while BERT uses full contextual information.



GPT model

BERT model

GPT: After BERT Beats GPT-1

- **Different goal of GPT-3 (and GPT-2)**
  - They have focused on **enhancing language model**
    - Using large and various corpus and model sizes for LM training

	GPT-1	GPT-2	GPT-3
`dataset_size`	1B words (BooksCorpus)	10B words (WebText)	300B (Mixture of corpus)
`max_token_num`	512	**1024**	**2048**
`batch_size`	64	512	0.5 - 3.2M
`model_size`	0.1B params 12 layers)	{0.1 - 1.5}B params {12-48} layers	{0.1 - 175}B params {12-96} layers

  - They have applied GPT to **unsupervised learning tasks**
    - They have explored the **few-shot behaviors**
    - GPT-2 and GPT-3 are **NOT fine-tuned to any target tasks**

Amost Everything can be found in
https://wikidocs.net/book/2155
https://github.com/ukairia777/tensorflow-nlp-tutorial
https://arxiv.org/abs/1810.04805
thanks to prof. Kyomin Jung