

# BERT

멘토 현시은

# BERT

- BERT의 구조
- BERT의 구성
- BERT의 Task
- Huggingface

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

**Jacob Devlin   Ming-Wei Chang   Kenton Lee   Kristina Toutanova**

Google AI Language

`{jacobdevlin, mingweichang, kentonl, kristout}@google.com`

# BERT의 구조

- **BERT의 구조**

BERT의 구성

BERT의 Task

Huggingface

- BERT

Bidirectional Encoder Representations from Transformers

- 트랜스포머 기반 양방향 인코더 표현



- **BERT의 구조**  
BERT의 구성  
BERT의 Task  
Huggingface

- Transformer

구글에서 공개한 "Attention is All You Need"라는 논문에서 처음으로 공개된 구조

- Attention을 기반으로 한다.
- 순환구조 때문에 시간이 오래 걸리는 RNN, seq2seq의 단점을 보완
- BERT, GPT-3 등의 언어 모형에서 사용
- 컴퓨터 비전 등 다른 분야에서도 강력한 도구로 사용 중

- **BERT의 구조**
- BERT의 구성
- BERT의 Task
- Huggingface

- Transformer

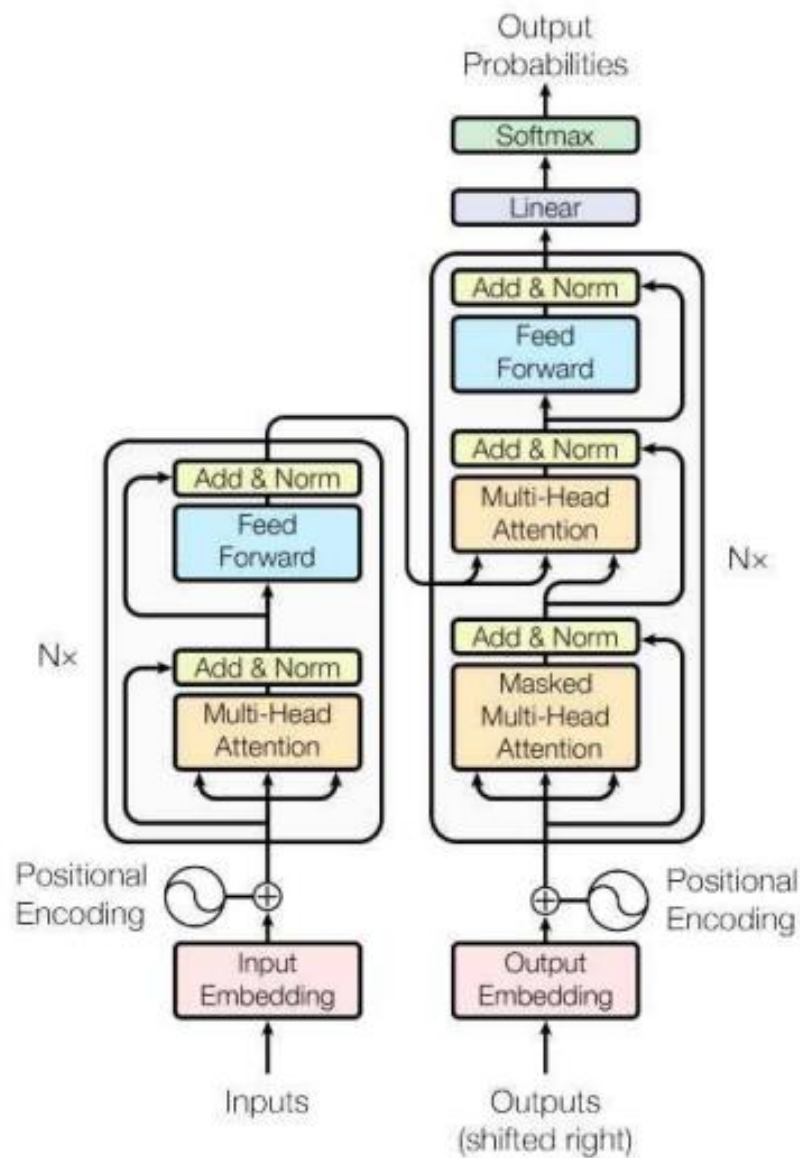
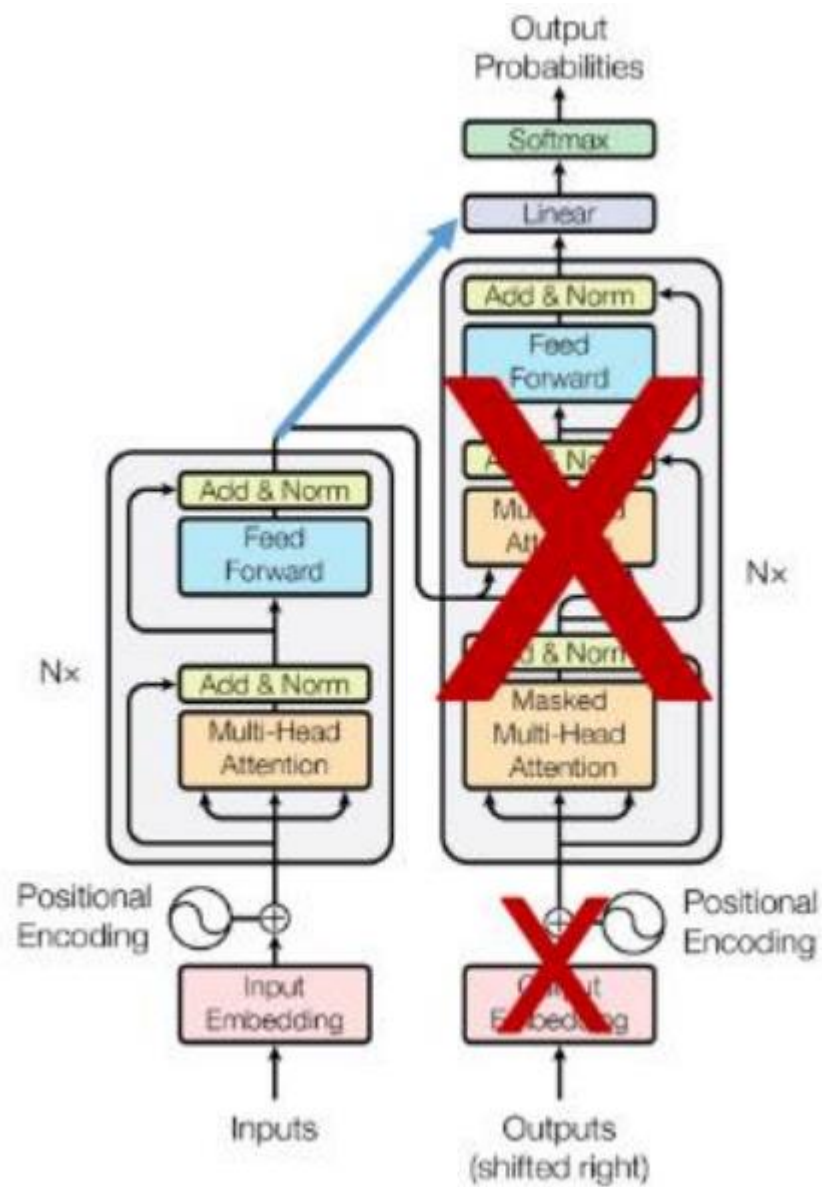


Figure 1: The Transformer - model architecture.

- BERT의 구조
- BERT의 구성
- BERT의 Task
- Huggingface

- BERT



- BERT의 구조

BERT의 구성

BERT의 Task

Huggingface

- BERT

**Bidirectional** Encoder Representations from Transformers

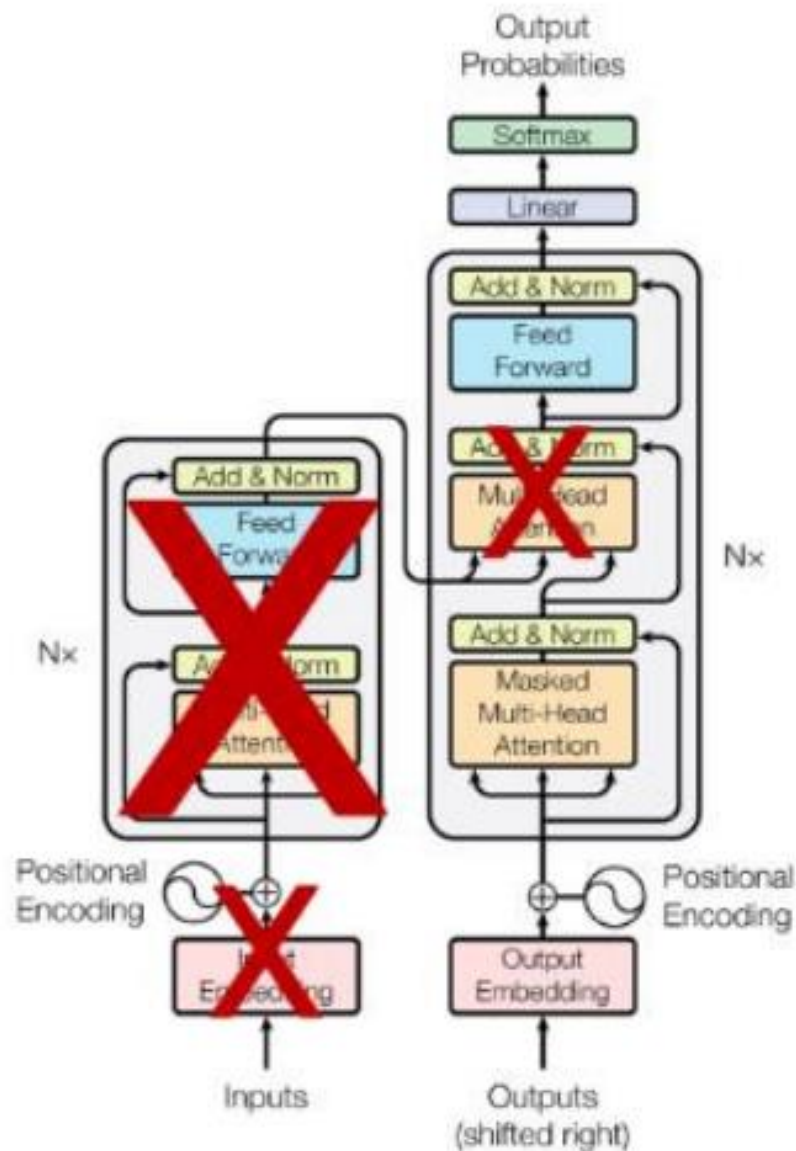
- 트랜스포머 기반 양방향 인코더 표현

<그럼 다른 Language Model은?>



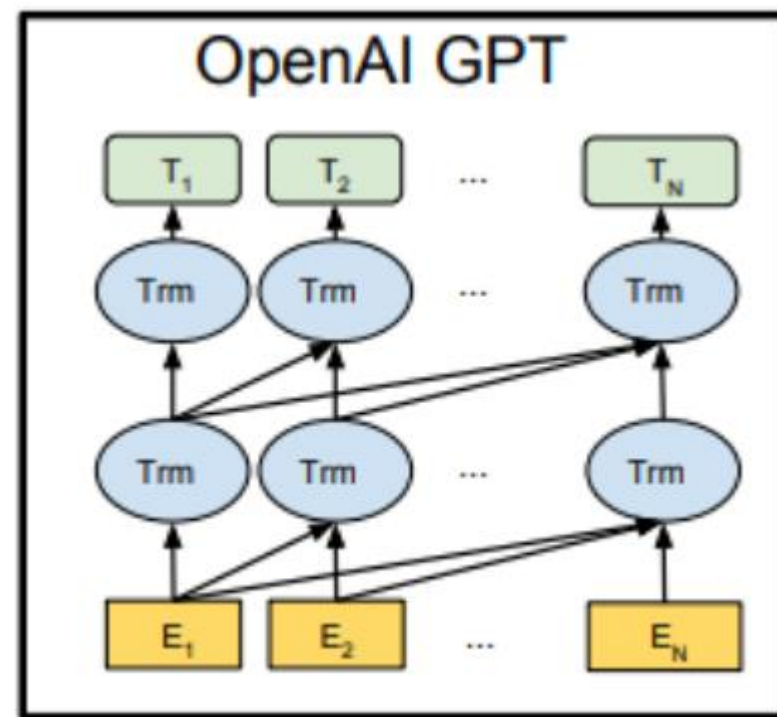
- **BERT의 구조**  
BERT의 구성  
BERT의 Task  
Huggingface

- GPT (Generative Pre-trained Transformer)
  - Unidirectional



- **BERT의 구조**  
BERT의 구성  
BERT의 Task  
Huggingface

- GPT (Generative Pre-trained Transformer)
  - i번째 입력 처리 시 i번째 이하의 토큰 만 고려
  - ex) 피곤해 보이네. 어제 늦게 잤니?



[Devlin et al., 2019]

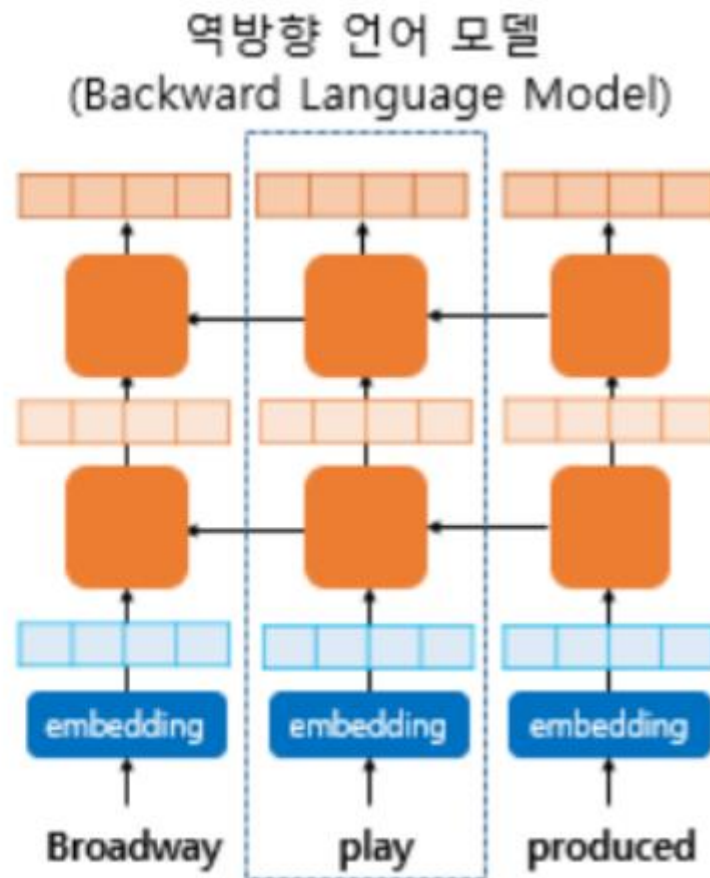
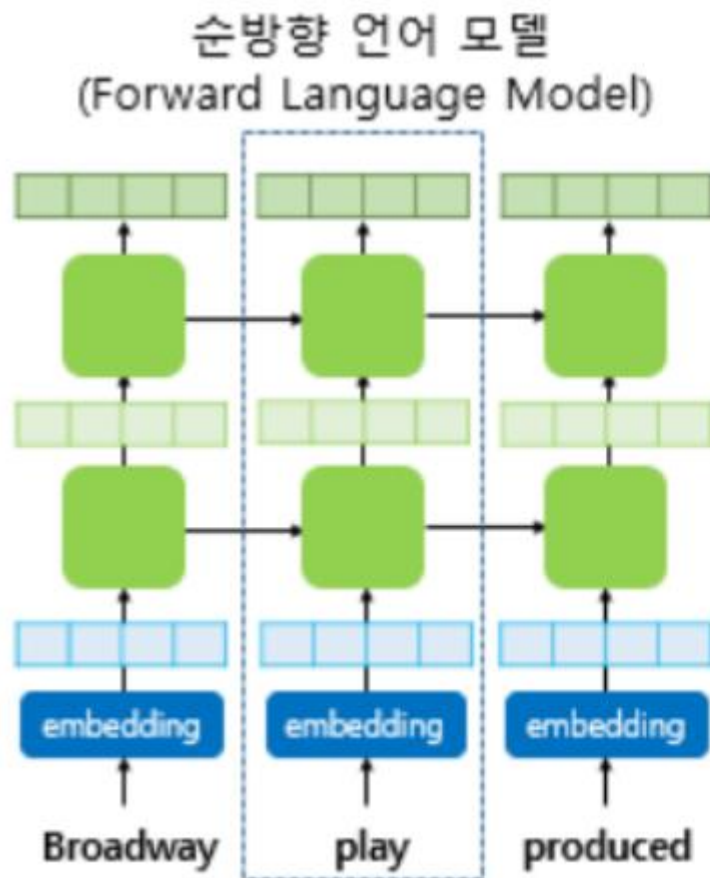
- BERT의 구조

BERT의 구성

BERT의 Task

Huggingface

- ELMo
  - RNN 기반 모델, biLM을 이용



[유원준 외 1명, 2022]

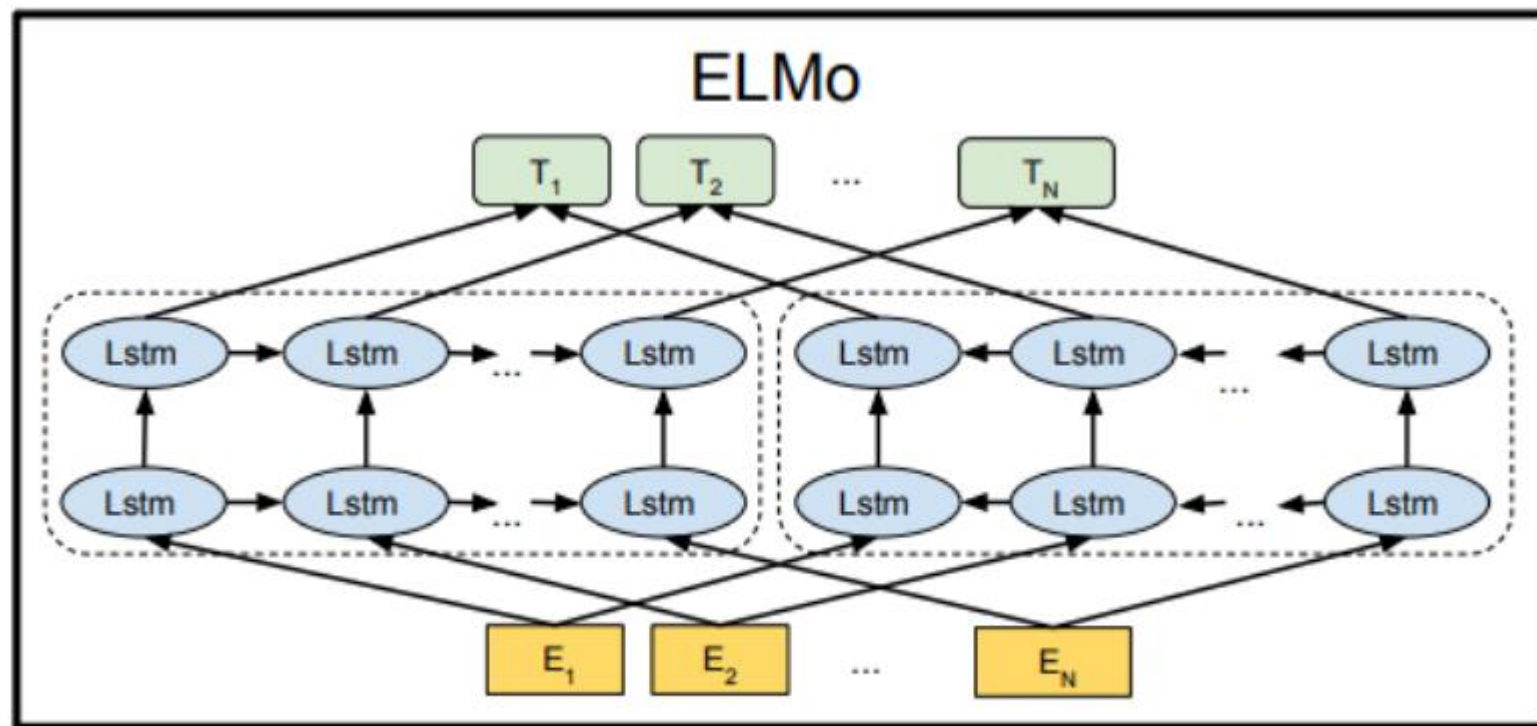
- BERT의 구조

BERT의 구성

BERT의 Task

Huggingface

- ELMo
  - Shallow Bidirectional(Unidirectional의 두 개를 단순 concat 한 것)



[Devlin et al., 2019]

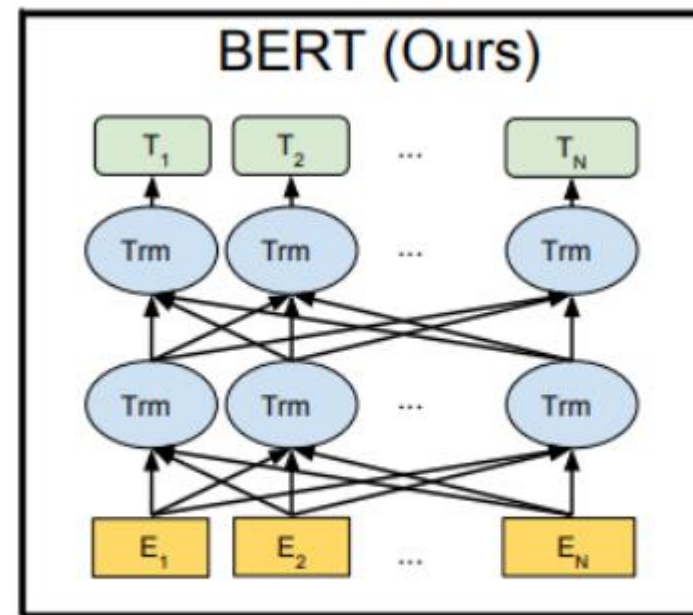
- BERT의 구조

BERT의 구성

BERT의 Task

Huggingface

- BERT (Bidirectional Encoder Representations from Transformers)
  - Transformer의 인코더의 Self Attention 원리 이용
  - Deeply Bidirectional



[Devlin et al., 2019]

- BERT의 구조
- BERT의 구성
- BERT의 Task
- Huggingface

## BERT 구조

**B**idirectional **E**ncoder **R**epresentations from **T**ransformers

### Bert base

- 12개의 transformer blocks
- n\_head: 12
- Hidden size  $d = 768$
- Total Parameters = 110 million

### Bert Large

- 24개의 transformer blocks
- n\_head: 16
- Hidden size  $d = 768$
- Total Parameters = 340 million

그럼 BERT는 도대체 왜 개발되었는가??

- 자연어처리(NLP)의 분야

- Natural Language Generation (NLG)

: 기계가 사람처럼 의미있는 문장을 생성하는 것

(ex. 챗봇(Chatbots), 기계 번역(Machine Translation), 자동 보고서 생성 등)

- Natural Language Understanding (NLU)

: 기계가 사람처럼 문장을 이해하는 것

(ex. 감성 분석, 자연어 질의응답, 자동 요약, 텍스트 분류 등)

- BERT의 구조

BERT의 구성

BERT의 Task

Huggingface

- GPT는 Natural Language Generation(NLG)에 적합!

Why? Unidirectional 모델이라서!

: 문맥상에 맞는 다음 단어나 문장을 예측하는 데 탁월

- BERT는 Natural Language Understanding(NLU)에 적합!

Why? Bidirectional 모델이라서!

: 텍스트 전체를 전반적으로 학습하는데 탁월



# BERT의 구성

## BERT의 구조

- **BERT의 구성**

## BERT의 Task

## Huggingface

- BERT는 어떤 기법으로 만들어졌는가?

(1) Pre-training

- MLM (Masked Language Model)

- NSP (Next Sentence Prediction)

- Embedding Combination

(2) Fine-tuning

BERT의 구조

- BERT의 구성

BERT의 Task  
Huggingface

## Pre-training & Fine-tuning

- 거대 데이터를 이용하여 미리 훈련(pre-training)을 해두고, 사전 훈련된 모델을 필요한 Task에 맞게 튜닝(Fine-tuning)하는 것

## Fine-tuning vs Feature Extraction

- 변형된 부분을 포함한 전체 모델 학습(Fine-tuning) : BERT, GPT
- 새로 쌓은 부분만 파라미터를 업데이트(Feature Extraction) : ELMo

## BERT의 구조

- BERT의 구성

## BERT의 Task Huggingface

### (1) MLM (Masked Language Model)

: BERT를 pre-training할 때 사용한 기법 중 하나

- 일정 비율의 토큰을 가린 채 문장을 복원하도록 학습
- 다음 time-step의 토큰을 예측하는 기존 모델과 달리, MLM은 현재 time-step의 토큰을 예측하는 것

## BERT의 구조

### • BERT의 구성

## BERT의 Task Huggingface

학습과 추론의 괴리를 없애기 위해:

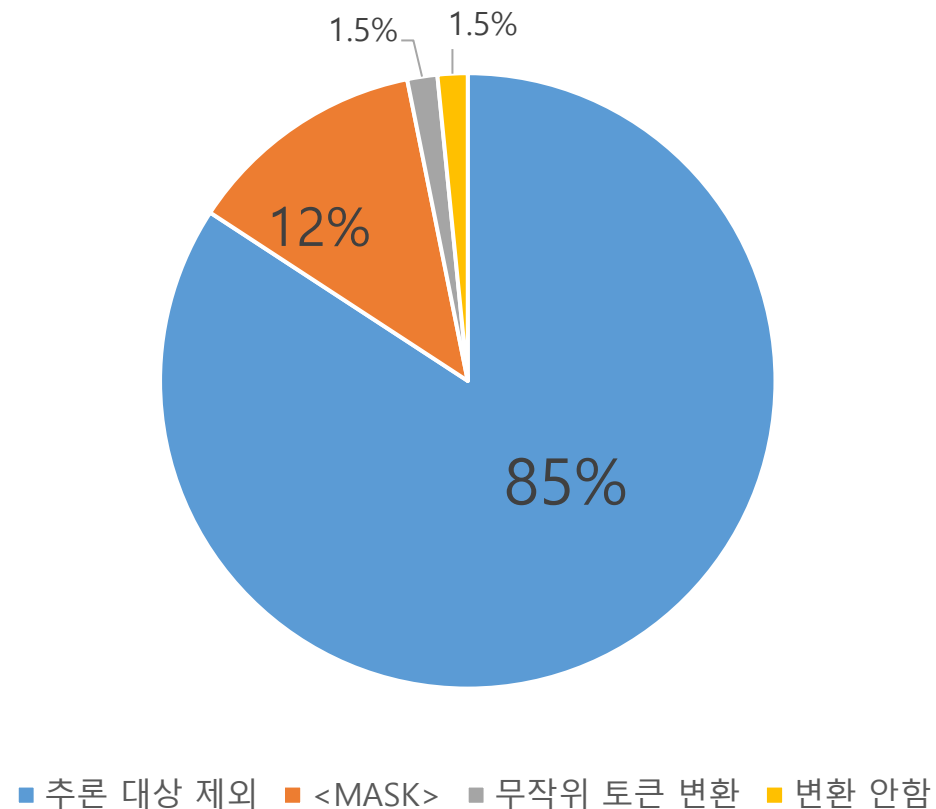
(1) 전체 토큰 중 15%의 토큰만 추론 대상으로 선정

(2) 15%의 토큰 중 80%(전체 중 12%)를 <MASK>로 가림

(3) 15%의 토큰 중 10%(전체 중 1.5%)를 랜덤 토큰으로 변환

(4) 15%의 토큰 중 10%(전체 중 1.5%)를 그대로 놔둠

Pre-training 시 토큰 비율

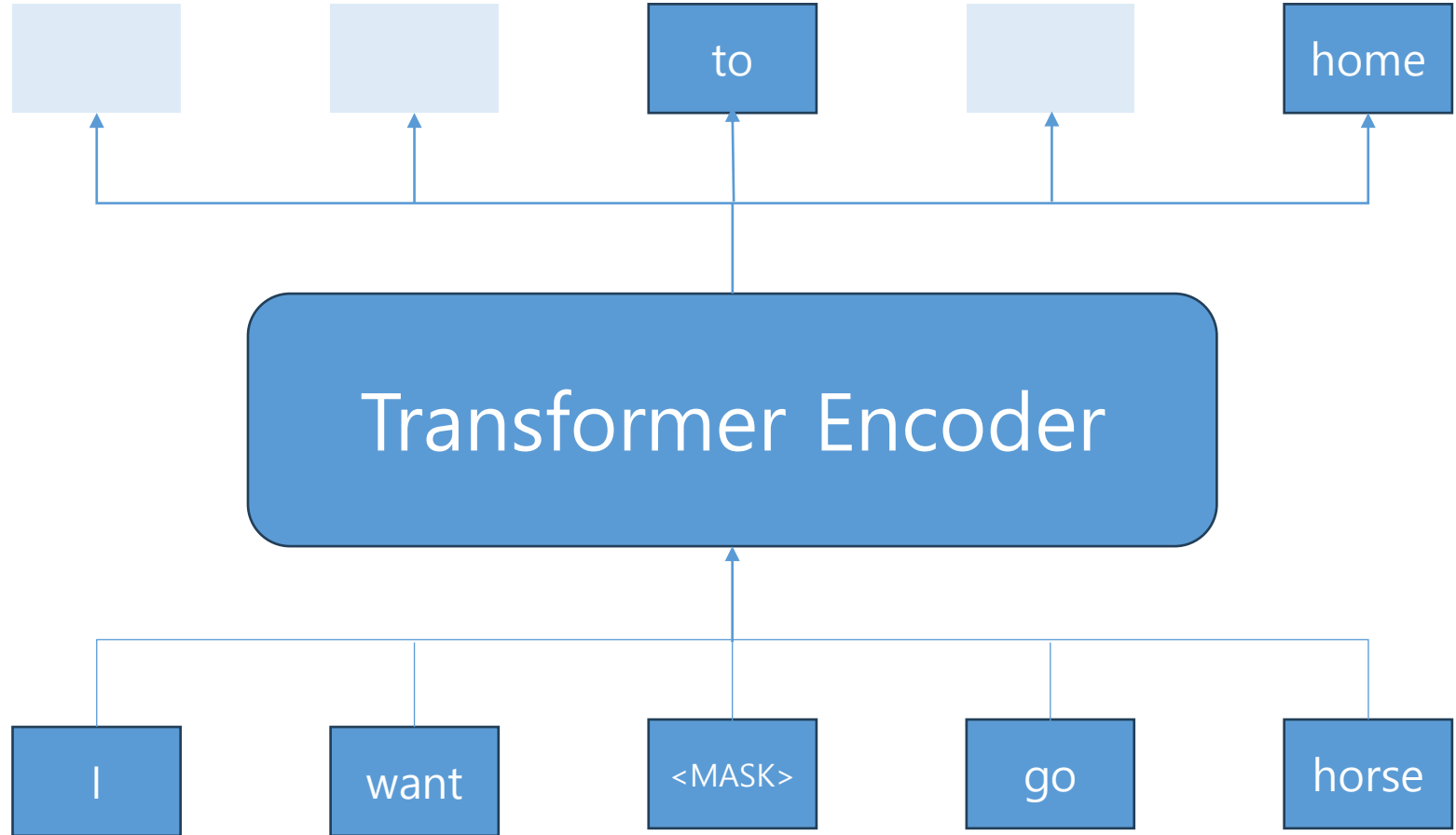


## BERT의 구조

- BERT의 구성

BERT의 Task  
Huggingface

### (1) MLM (Masked Language Model)



## BERT의 구조

- BERT의 구성

## BERT의 Task Huggingface

### (2) NSP (Next Sentence Prediction)

: BERT를 pre-training할 때 사용한 기법 중 하나

- 두 번째 문장이 첫 번째 문장 다음에 나오는지 여부를 예측
- 문장과 문단을 구분할 때 2개의 특별한 토큰을 추가했다.
- [CLS] : 전체 시퀀스 분류
- [SEP] : 시퀀스 내 문장 분류
- (ex) [CLS] 어제 늦게 잤어 ? [SEP] 어쩐지 피곤해 보이더라 . [SEP]

## BERT의 구조

- BERT의 구성

## BERT의 Task Huggingface

### (2) NSP (Next Sentence Prediction)

- Question Answering(질의응답)또는 Textual Entailment(텍스트 추론)의 경우 문장 사이의 관계를 이해하는 것이 중요하다.

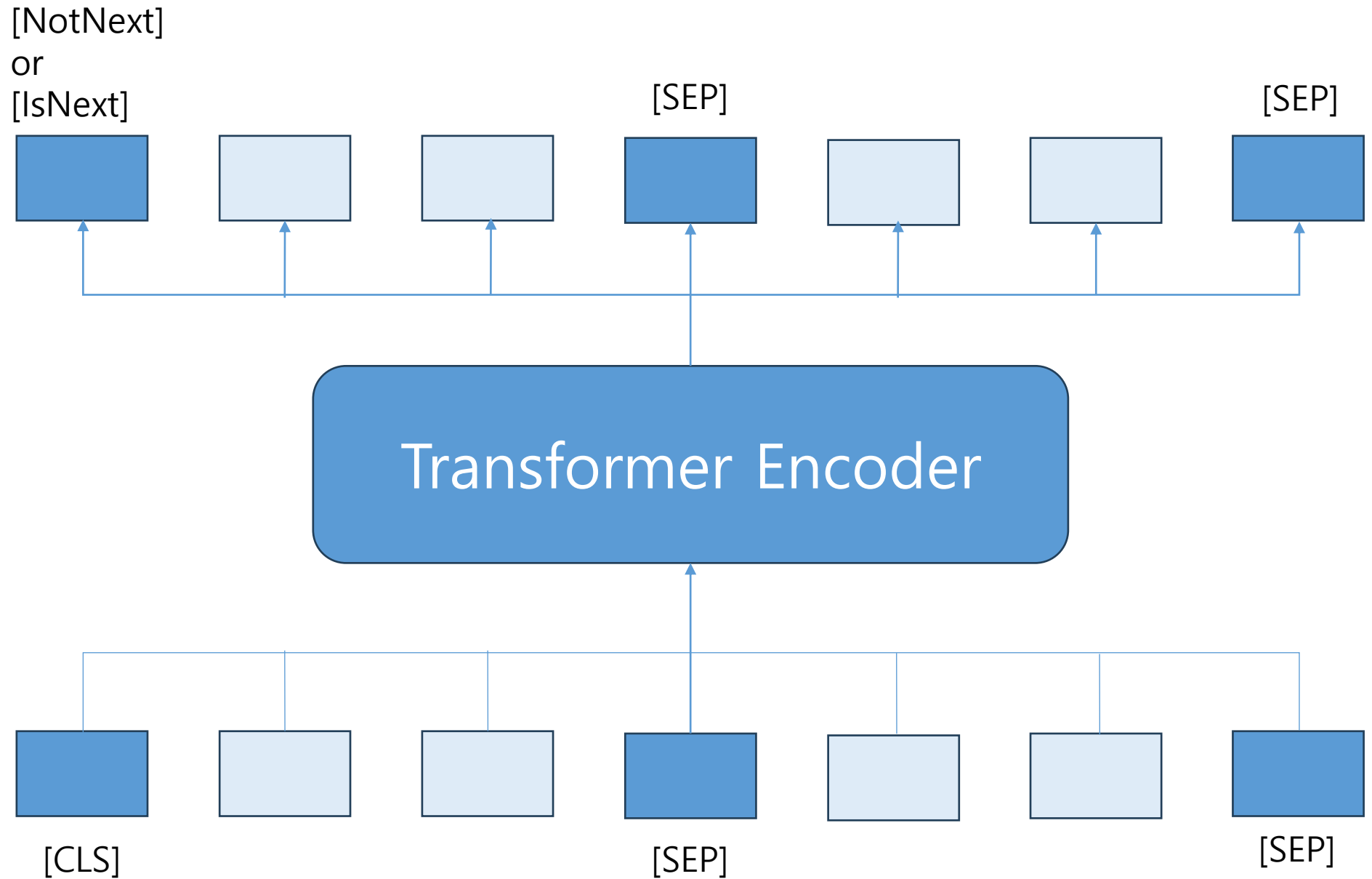
- [SEP]으로 분리되는 두 문서 (A, B)를 통과시키고 B를 50%의 확률로 임의의 문서로 대체한다. 그 후 [CLS] 토큰의 위치에서 대체 여부를 예측하도록 학습한다.



## BERT의 구조

- BERT의 구성

BERT의 Task  
Huggingface



## BERT의 구조

- BERT의 구성

## BERT의 Task

## Huggingface

### (3) Embedding Combination

- 기존 Transformer : 단어 임베딩 + 위치 인코딩
- BERT : 단어 임베딩 + 문장 임베딩 + 위치 임베딩

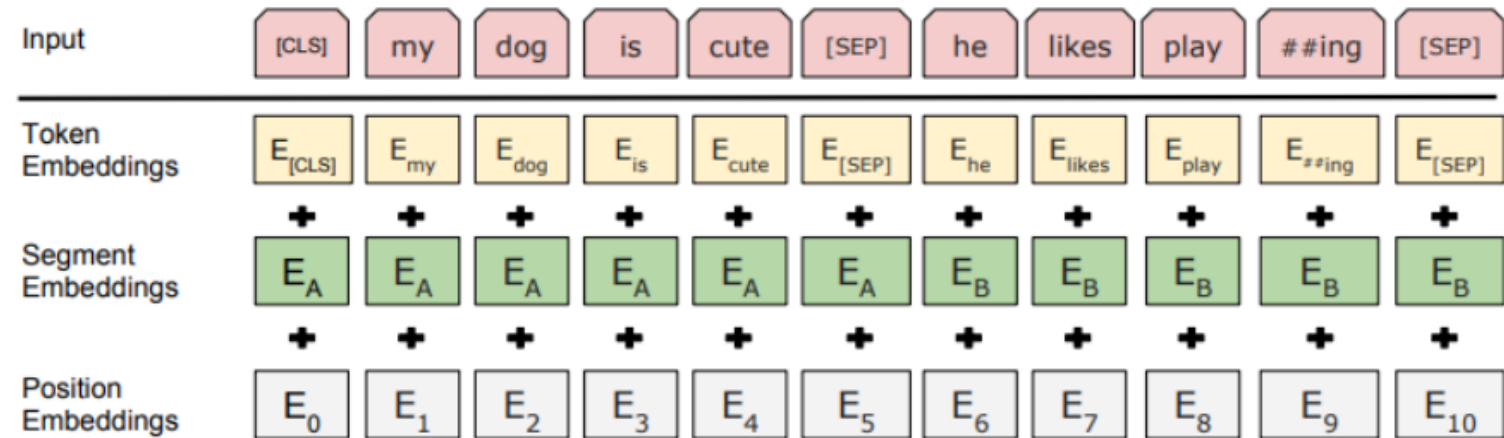


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

[Devlin et al., 2019]

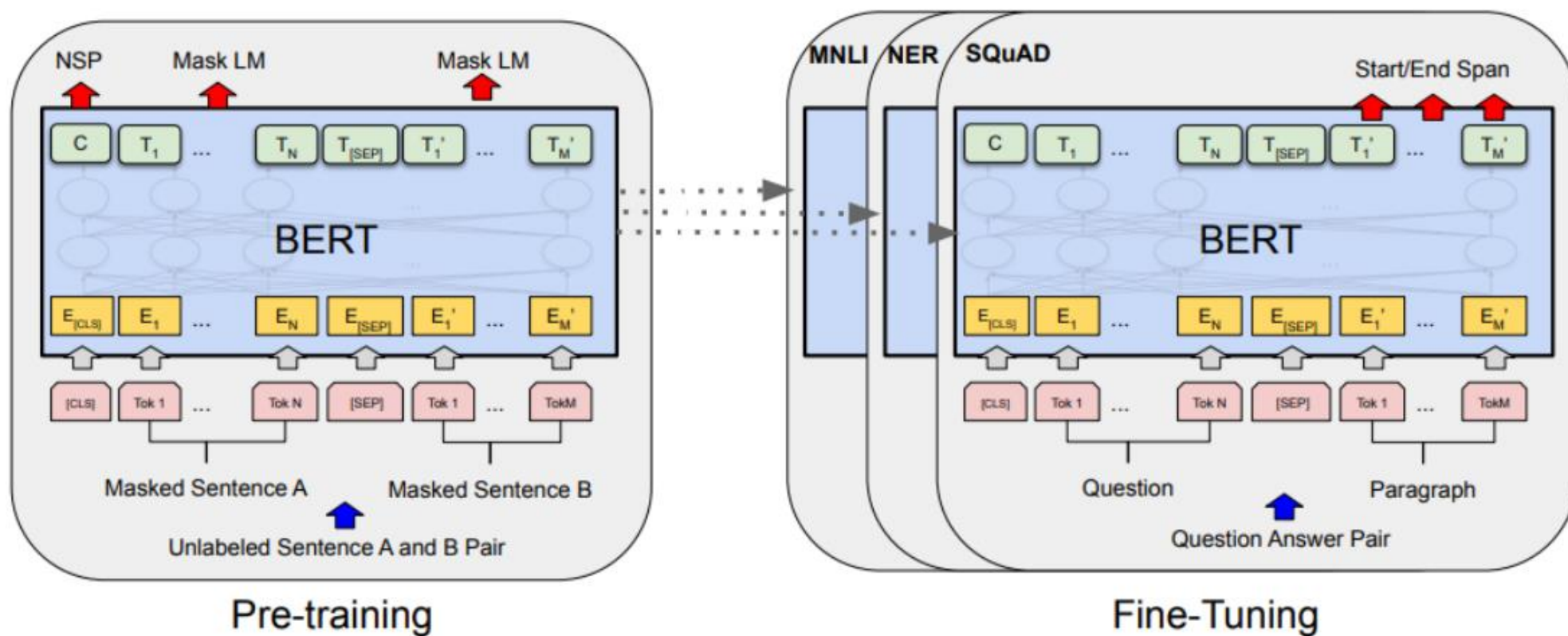
## BERT의 구조

- BERT의 구성

BERT의 Task  
Huggingface

## Fine-tuning

- Pre-Training으로 얻은 모델을 기반으로 task에 맞게 변형, 조정 한 것



[Devlin et al., 2019]

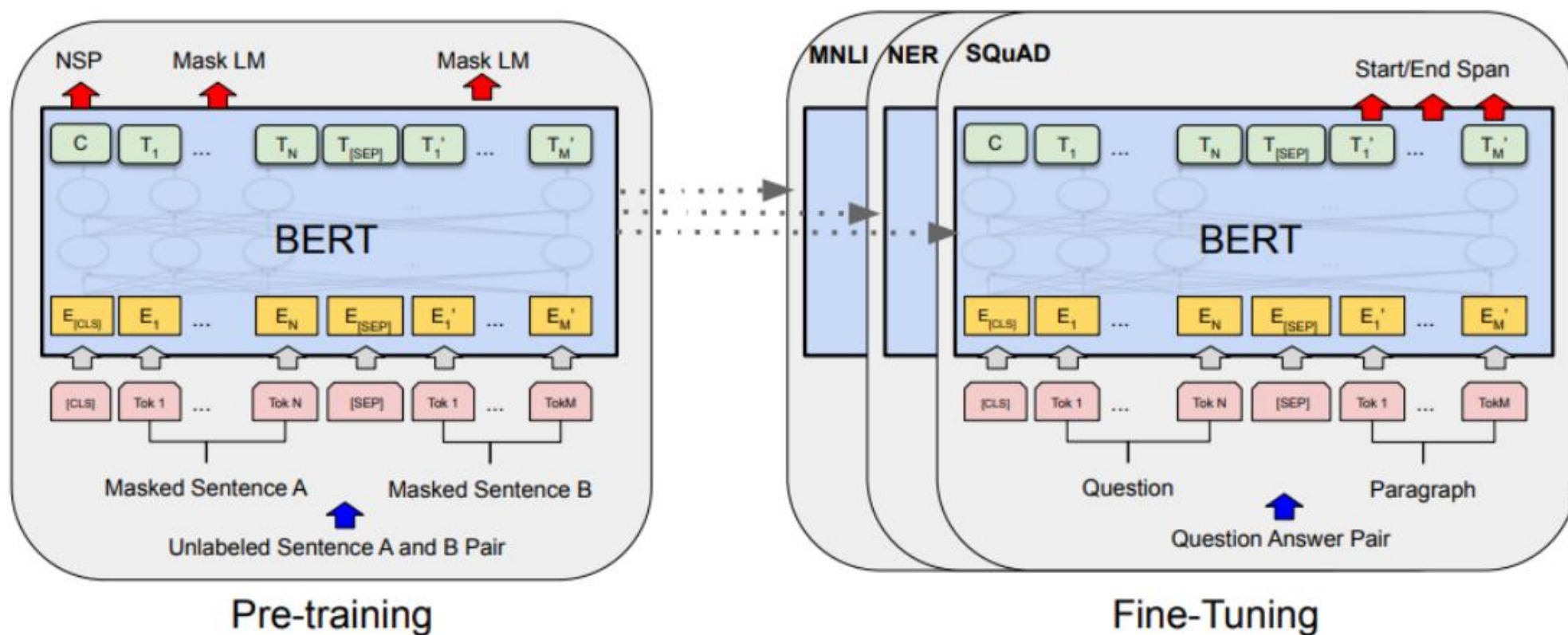
## BERT의 구조

- BERT의 구성

BERT의 Task  
Huggingface

## Fine-tuning

- Pre-Training으로 얻은 모델을 기반으로 task에 맞게 변형, 조정 한 것



[Devlin et al., 2019]

## BERT의 구조

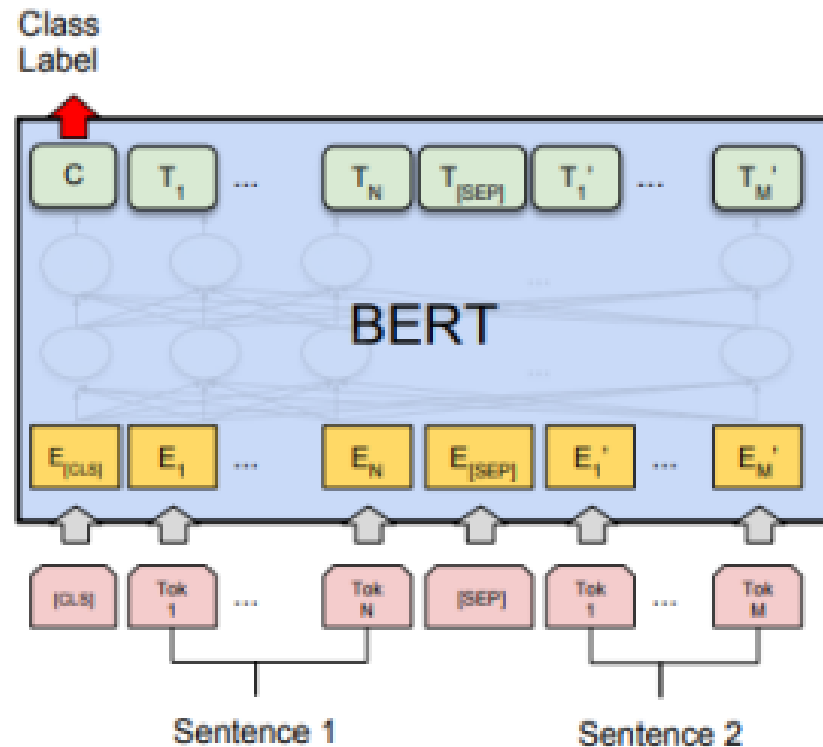
- BERT의 구성

## BERT의 Task

Huggingface

## Fine-tuning

- 특히, Text Classification 에서는 [CLS] 토큰 위치에 linear layer와 Softmax를 추가하여 분류



# BERT의 Task

BERT의 구조

BERT의 구성

- BERT의 Task

Huggingface

Pre-Training으로 얻은 모델을 기반으로 task에 맞게 변형 및 조정

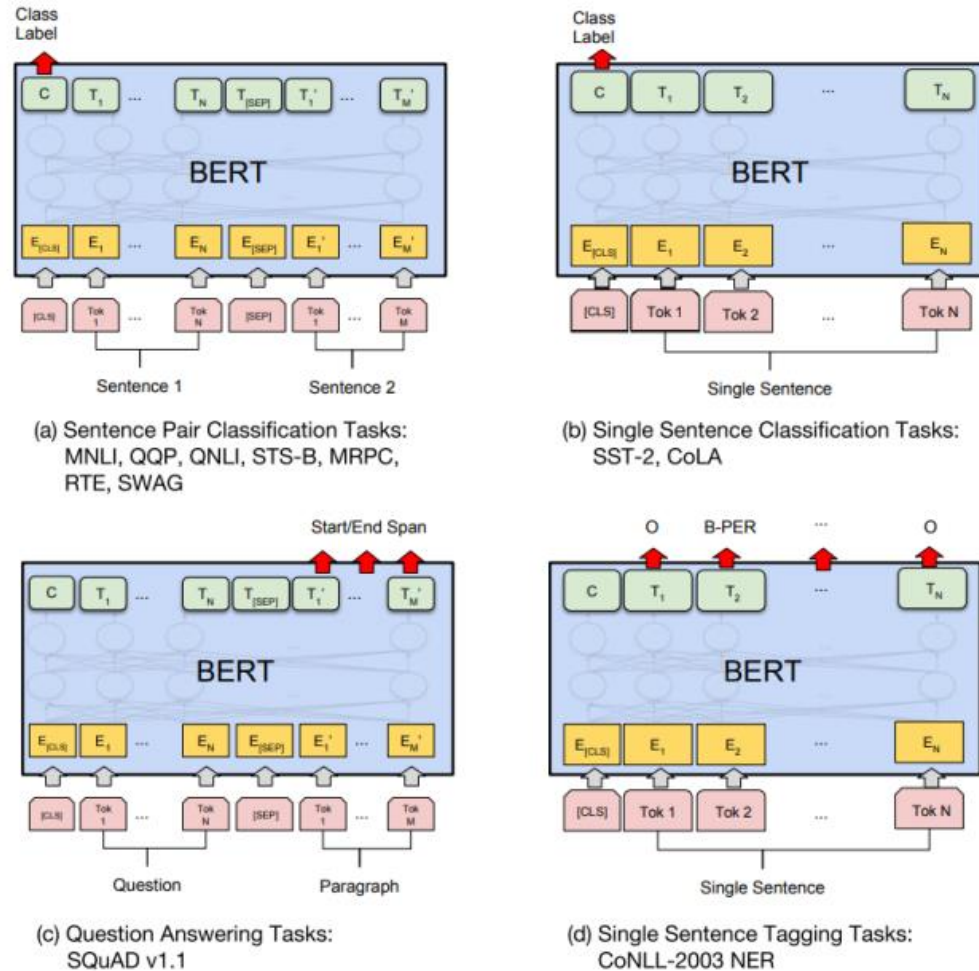


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

[Devlin et al., 2019]

BERT의 구조

BERT의 구성

- **BERT의 Task**

Huggingface

주의! BERT는 NLU에만 적용 가능하며, NLG에는 적용할 수 없음





Huggingface

BERT의 구조

BERT의 구성

BERT의 Task

- Huggingface

## Huggingface란? 인공지능 버전 GitHub!!

- Model Hub를 제공
- 사용자들이 모델을 공유하고, 자신들의 코드를 공유한다.  
: <https://huggingface.co/models>
- Pytorch나 TensorFlow 코드로 모두 동작하기에 쉽게 불러오는 것이 가능
- 모델 아키텍처 뿐만 아니라 전처리 및 학습, 테스트 코드 등도 제공받을 수 있음
- BERT, GPT, CLIP 등 NLU와 NLG를 위한 다양한 아키텍처를 제공
- CV 아키텍처도 존재한다

BERT의 구조

BERT의 구성

BERT의 Task

## • Huggingface

<https://huggingface.co/docs/transformers/index>  
문서 및 사용법이 굉장히 잘 정리되어 있다.

The screenshot shows the Hugging Face website's Transformers documentation page. The top navigation bar includes the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Docs, Solutions, Pricing, Log In, and Sign Up. The left sidebar features a 'Transformers' dropdown menu with a search bar, version (V4.31.0), language (EN), and a user icon. Below this are sections for 'GET STARTED' (Transformers, Quick tour, Installation) and 'TUTORIALS' (Run inference with pipelines, Write portable code with AutoClass, Preprocess data, Fine-tune a pretrained model, Train with a script, Set up distributed training with Accelerate, Share your model). The main content area has a large orange box titled 'Join the Hugging Face community' with the text 'and get access to the augmented documentation experience'. It lists three benefits: 'Collaborate on models, datasets and Spaces', 'Faster examples with accelerated inference', and 'Switch between documentation themes'. A 'Sign Up' button is present with the text 'to get started'. Below this is a section titled 'Transformers' with the text 'State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX.' and a paragraph explaining that Transformers provides APIs and tools to easily download and train state-of-the-art pretrained models, reducing compute costs and carbon footprint.

**Hugging Face** Search models, datasets, users...

Models Datasets Spaces Docs Solutions Pricing Log In Sign Up

**Transformers** Search documentation Ctrl+K

V4.31.0 EN 108,900

**GET STARTED**

- Transformers
- Quick tour
- Installation

**TUTORIALS**

- Run inference with pipelines
- Write portable code with AutoClass
- Preprocess data
- Fine-tune a pretrained model
- Train with a script
- Set up distributed training with Accelerate
- Share your model

**Join the Hugging Face community**  
and get access to the augmented documentation experience

- Collaborate on models, datasets and Spaces
- Faster examples with accelerated inference
- Switch between documentation themes

Sign Up to get started

**Transformers**

State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX.

Transformers provides APIs and tools to easily download and train state-of-the-art pretrained models. Using pretrained models can reduce your compute costs, carbon footprint, and save you the time and resources required to train a model from scratch. These models support common tasks in different modalities, such as:

**Transformers**

If you are looking for custom support from the Hugging Face team

Contents

- Supported models
- Supported frameworks

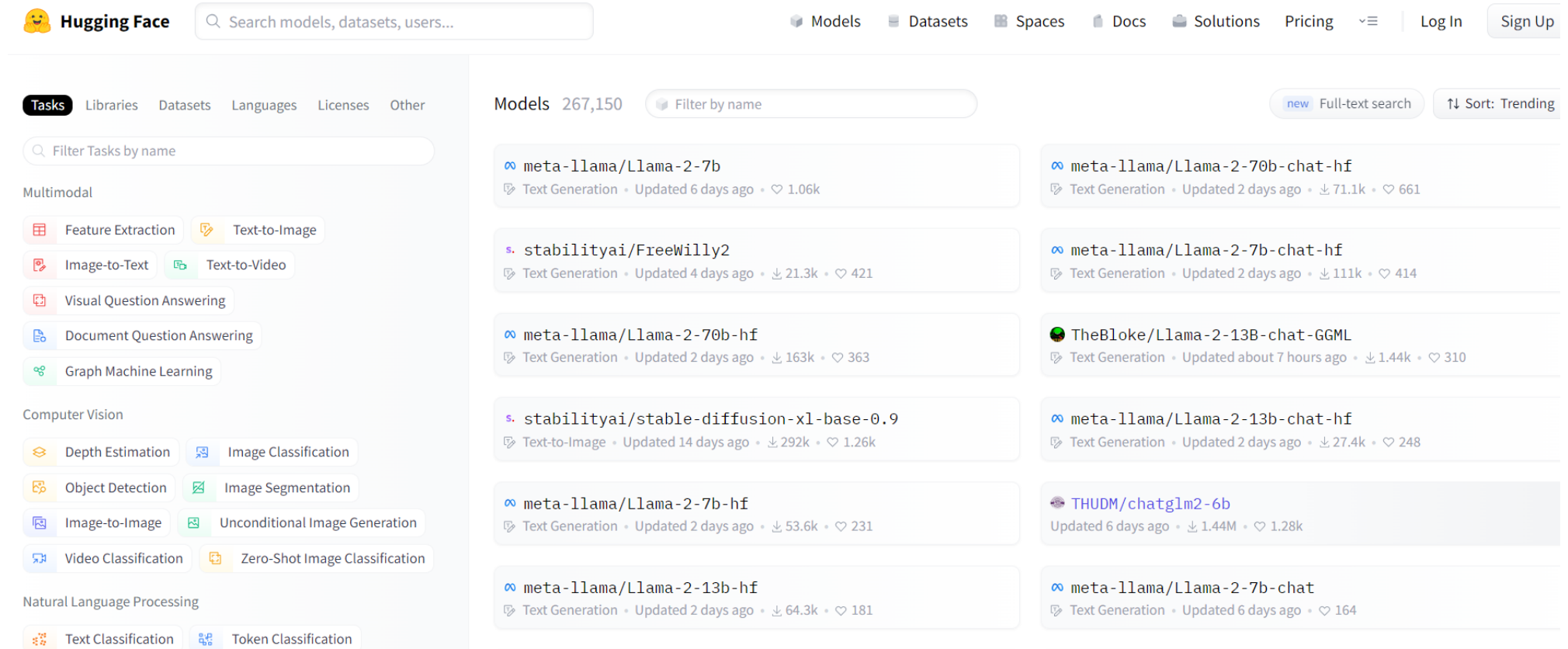
BERT의 구조

BERT의 구성

BERT의 Task

## • Huggingface

<https://huggingface.co/models>



The screenshot shows the Hugging Face website's 'Models' page. The header includes the Hugging Face logo, a search bar, and navigation links for Models, Datasets, Spaces, Docs, Solutions, Pricing, Log In, and Sign Up. The main content area is divided into a left sidebar and a main grid of model cards.

**Left Sidebar (Tasks):**

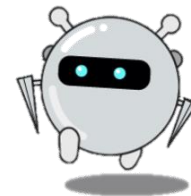
- Multimodal:** Feature Extraction, Text-to-Image, Image-to-Text, Text-to-Video, Visual Question Answering, Document Question Answering, Graph Machine Learning.
- Computer Vision:** Depth Estimation, Image Classification, Object Detection, Image Segmentation, Image-to-Image, Unconditional Image Generation, Video Classification, Zero-Shot Image Classification.
- Natural Language Processing:** Text Classification, Token Classification.

**Main Grid (Models):**

- meta-llama/Llama-2-7b:** Text Generation • Updated 6 days ago • 1.06k
- stabilityai/FreeWilly2:** Text Generation • Updated 4 days ago • 21.3k • 421
- meta-llama/Llama-2-70b-hf:** Text Generation • Updated 2 days ago • 163k • 363
- stabilityai/stable-diffusion-xl-base-0.9:** Text-to-Image • Updated 14 days ago • 292k • 1.26k
- meta-llama/Llama-2-7b-hf:** Text Generation • Updated 2 days ago • 53.6k • 231
- meta-llama/Llama-2-13b-hf:** Text Generation • Updated 2 days ago • 64.3k • 181
- meta-llama/Llama-2-70b-chat-hf:** Text Generation • Updated 2 days ago • 71.1k • 661
- meta-llama/Llama-2-7b-chat-hf:** Text Generation • Updated 2 days ago • 111k • 414
- TheBloke/Llama-2-13B-chat-GGML:** Text Generation • Updated about 7 hours ago • 1.44k • 310
- meta-llama/Llama-2-13b-chat-hf:** Text Generation • Updated 2 days ago • 27.4k • 248
- THUDM/chatglm2-6b:** Updated 6 days ago • 1.44M • 1.28k
- meta-llama/Llama-2-7b-chat:** Text Generation • Updated 6 days ago • 164



감사합니다



### 참고문헌

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', 2019
2. [https://ratsgo.github.io/nlpbook/docs/language\\_model/bert\\_gpt/](https://ratsgo.github.io/nlpbook/docs/language_model/bert_gpt/)
3. Vaswani et al., 'Attention Is All You Need', 2017
4. 유원준 외 1명, <딥 러닝을 이용한 자연어 처리 입문>, 2022