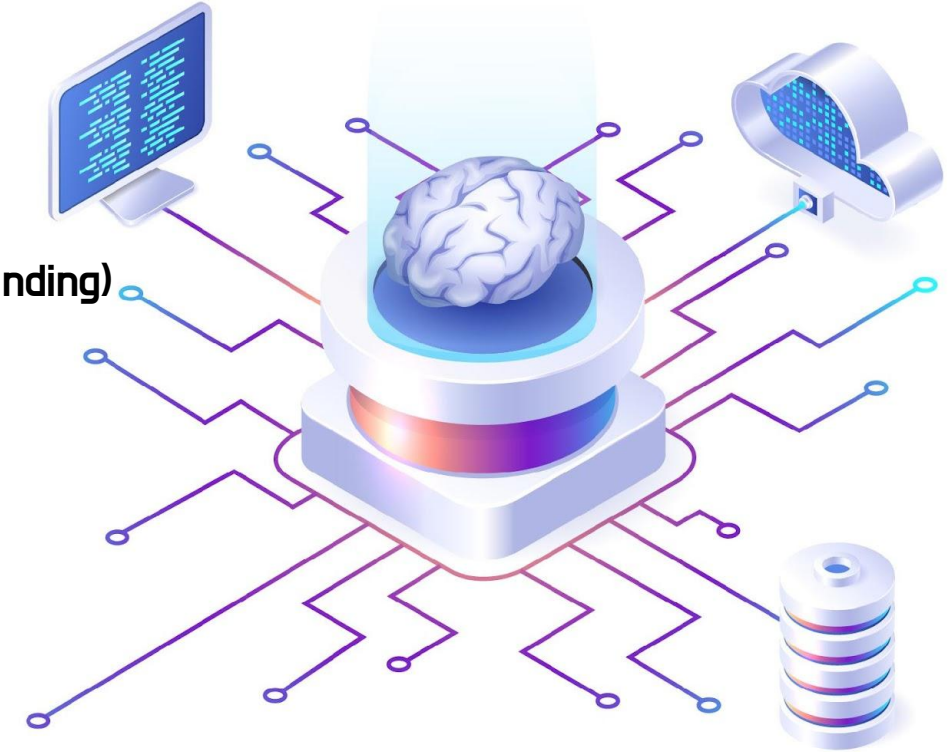
Insight campus SeSAC

BERT

(Pre-training of Deep Bidirectional Transformers for Language Understanding)

실무형 인공지능 자연어처리



Insight campus SeSAC

BERT 까지

단어의 표현

문맥의 표현

RNN

- 순환 구조로 시계열에 대한 고려 가능
- 장기문맥 손실 (Long dependency)

LSTM / GRU

- grad vanishing/exploding 문제 해결
- 장기문맥에 대한 고려

ElMo

- biLM (양방향 언어모델)
- Bidirectional, Stacking

CNN

- 이미지 분류에 많이 사용
- 시계열에 대한 고려 X

Seq2Seq

- Encoder - Decoder 구조
- Many to many
- 고정 문맥 벡터로 표현하여 정보 손실 발생
- Teacher forcing

Seq2Seq with Attention

- 집중해야할 정보를 고려 = attention
- LSTM 기반 attention
- Attention mechanism

Transformer (Attention is all you need)

- LSTM을 제거. 속도향상
- Self-attention
- Multihead attention
- Positional encoding

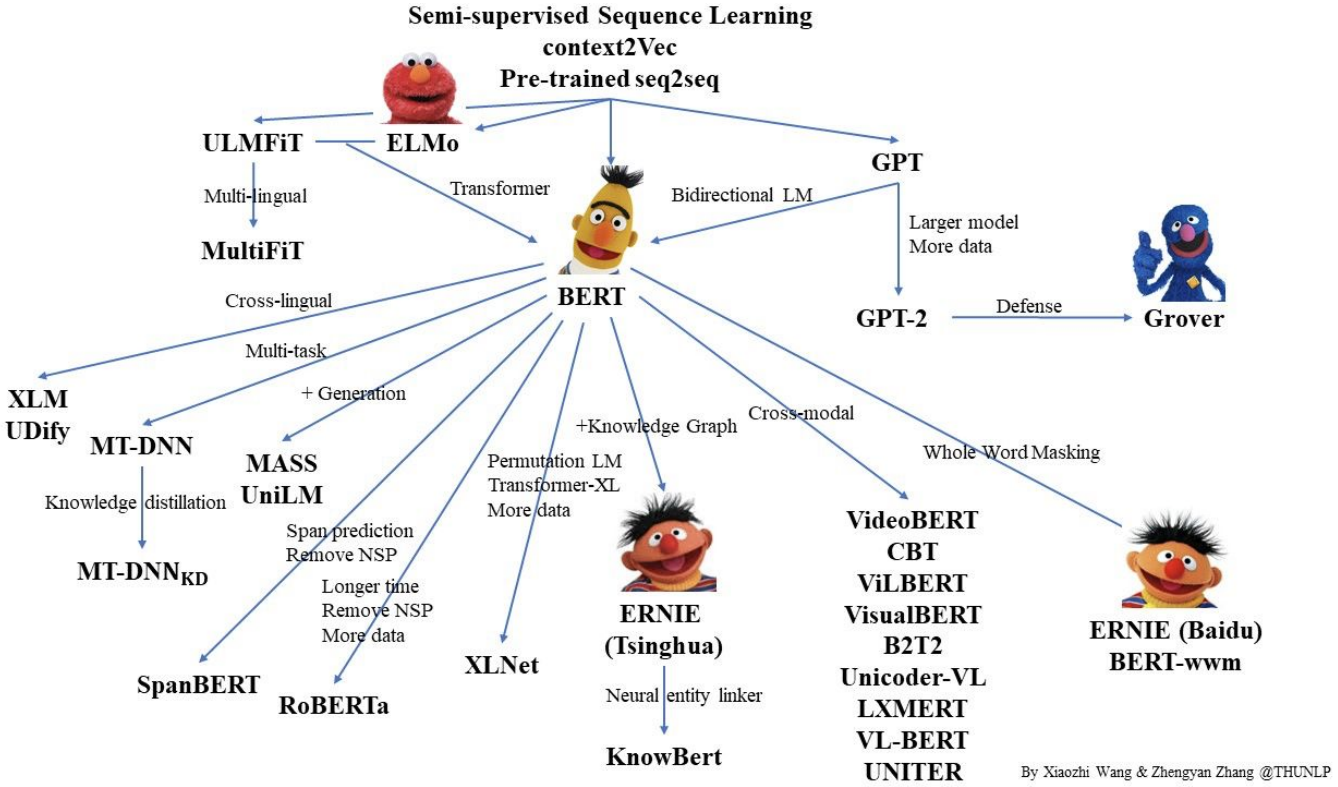
BERT

- Wordpiece tokenizer
- Masked Language Model(MLM)
- Next Sentence Prediction (NSP)
- Position embedding
- Pre-trained model
- Fine tuning

1

BERT

Pre-training of Deep Bidirectional Transformers for Language Understanding

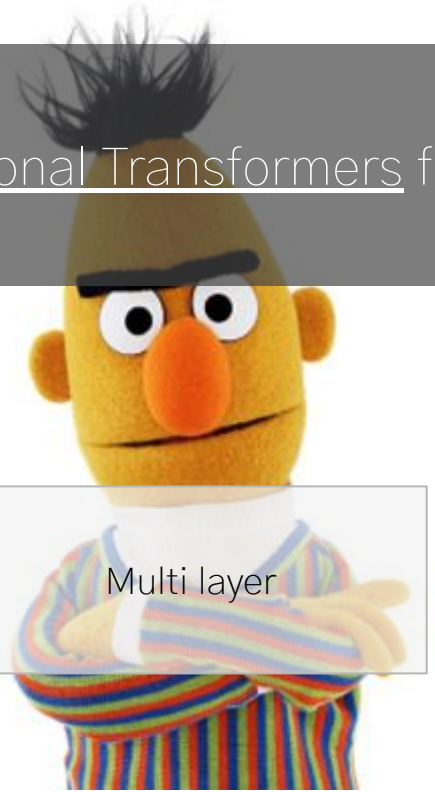

Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
GPT
Multi-lingual
MultiFiT
Transformer
Bidirectional LM
Larger model
More data
Cross-lingual
BERT
GPT-2
Defense
Grover
Multi-task
XLM
UDify
MT-DNN
+ Generation
MASS
UniLM
+Knowledge Graph
Cross-modal
Whole Word Masking
Knowledge distillation
MT-DNN$_{KD}$
Span prediction
Remove NSP
Permutation LM
Transformer-XL
More data
Longer time
Remove NSP
More data
SpanBERT
RoBERTa
XLNet
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
ERNIE (Baidu)
BERT-wwm
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

BERT

BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers)

BERT Overview (1)

Pre-training of Deep Bidirectional Transformers for Language Understanding



Pre-trained model
+ Fine tuning

Multi layer

Bidirectional Transformer
(Encoder)

Insight campus SeSAC

BERT Overview (2)

- BERT : Bidirectional Encoder Representations form Transformer
- “Attention is all you need(Vaswani et al., 2017)”에서 소개한 Transformer 활용 Language Representation
- wiki나 book data 대용량 **unlabeled data로 모델을 미리 학습** 시킨 후, 특정 task를 가지고 있는 labeled data로 transfer learning을 하는 모델
- **Fine-tuning**을 통해 task의 state-of-the-art를 달성

SQuAD1.1 Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

https://arxiv.org/abs/1810.04805

BERT vs OpenAI GPT vs ELMo

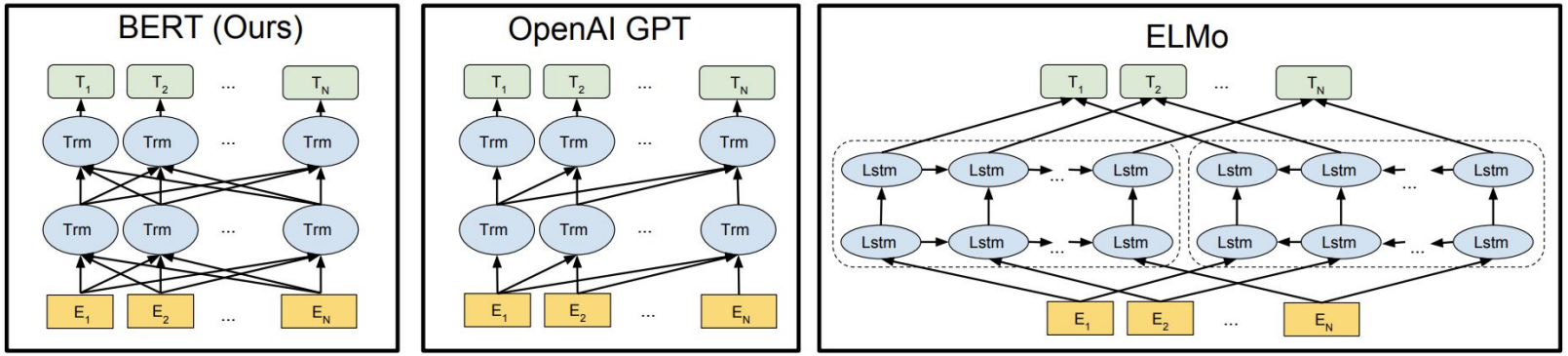


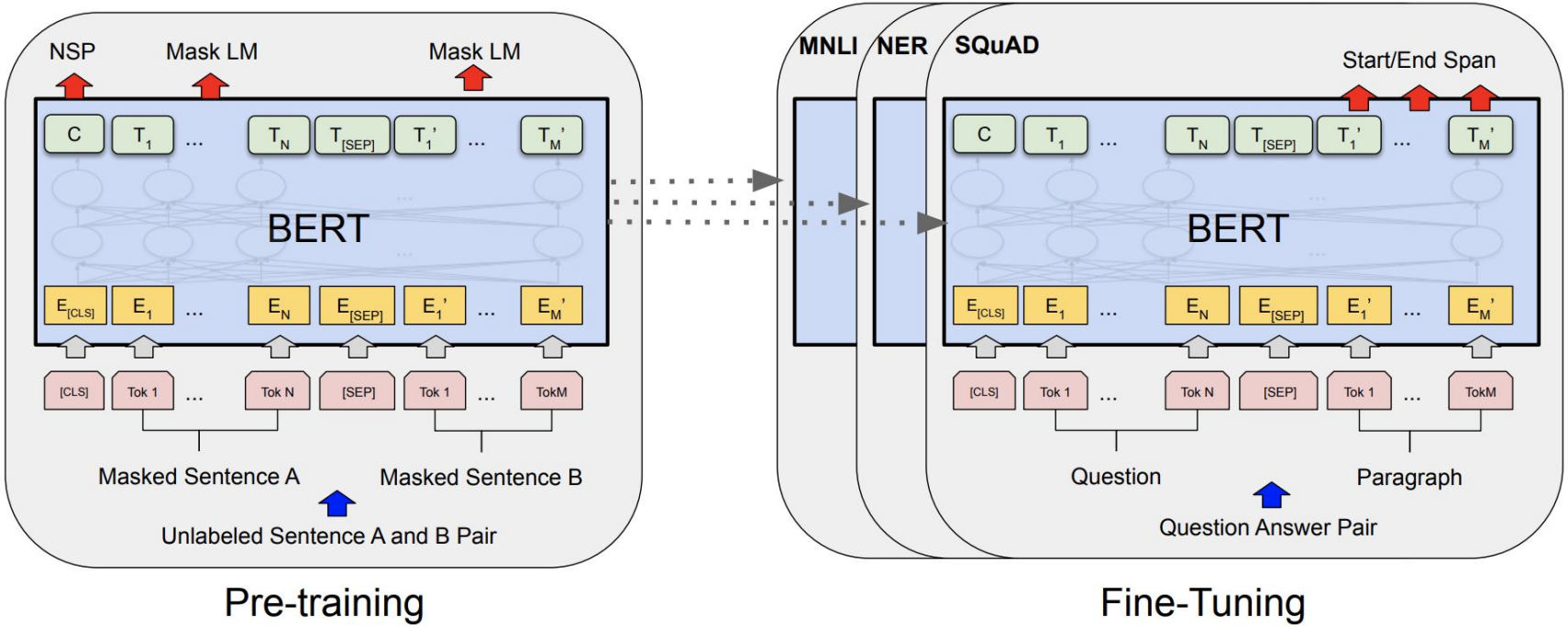
Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

Evaluation

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**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

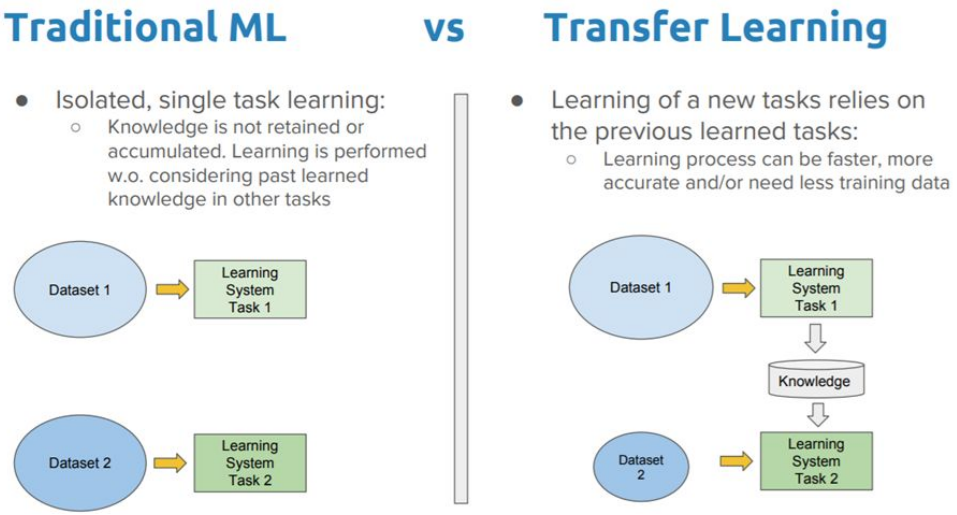
Fine-Tuning (1)



Pre-trained model를 기반으로 Fine-Tuning 하여 task 해결

Fine-Tuning (2)

- Transfer learning : 전이 학습은 특정 환경에서 만들어진 AI 알고리즘을 다른 비슷한 분야에 적용
- Fine-Tuning : 사전에 학습된 모델의 파라미터를 task에 맞추어 정교하게 조정하여 활용



Fine-Tuning (3)

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step

Model: BERT

Dataset:  

Objective: Predict the masked word (langauge modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step

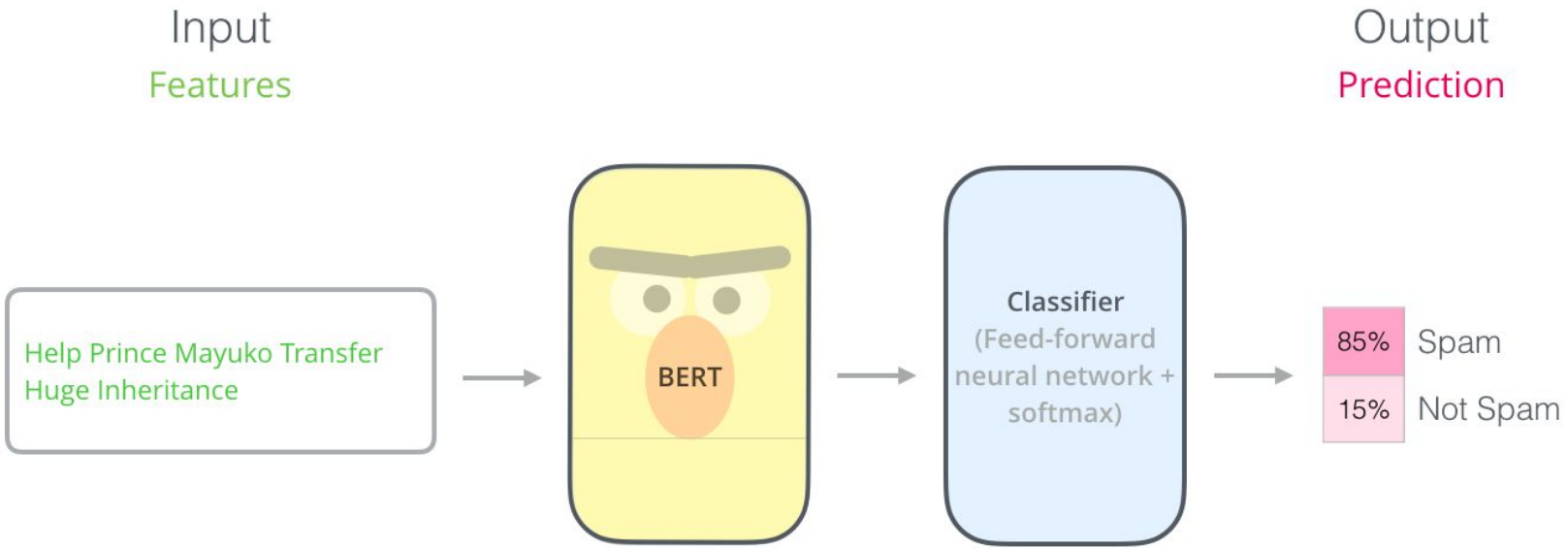
Model: (pre-trained in step #1) BERT

Classifier → 75% Spam, 25% Not Spam

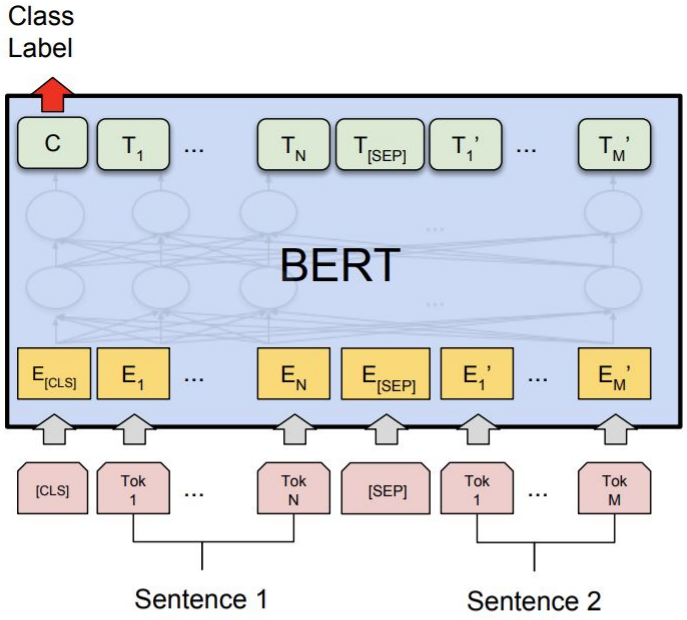
Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

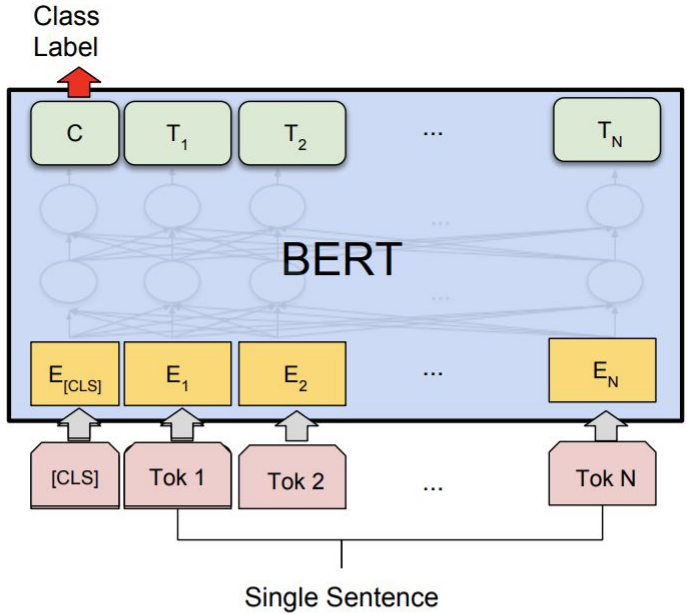
Fine-Tuning (4)



Fine-Tuning (5)

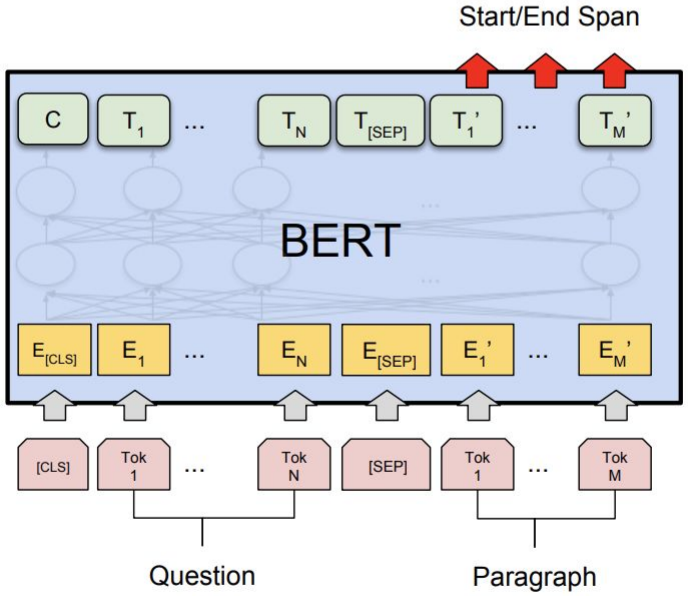


(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

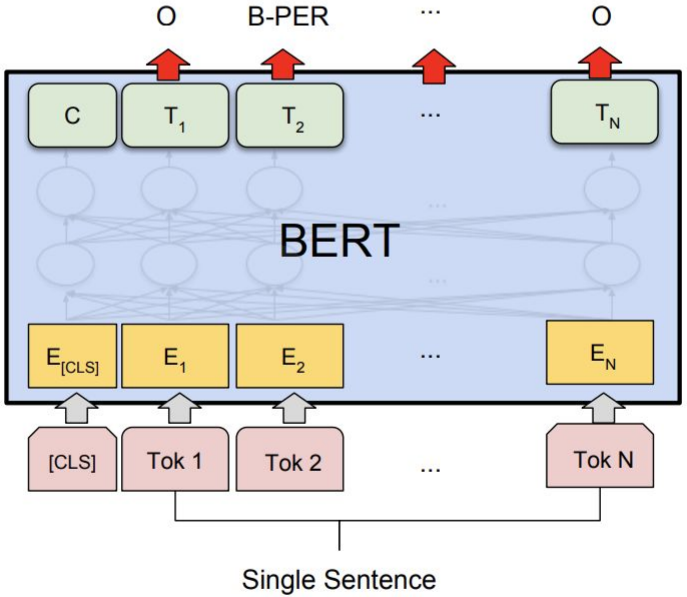


(b) Single Sentence Classification Tasks: SST-2, CoLA

Fine-Tuning (6)

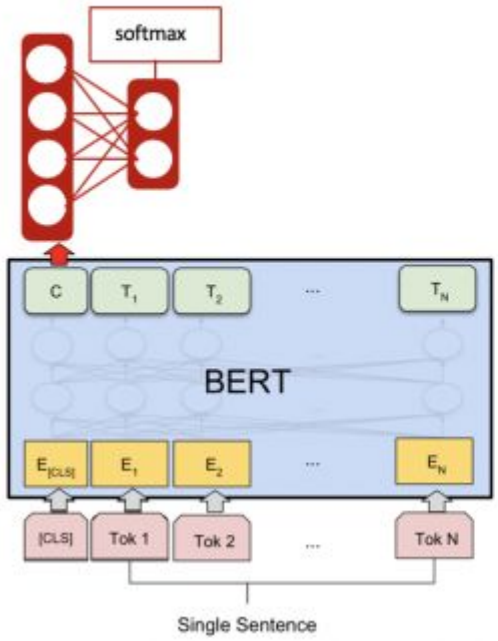


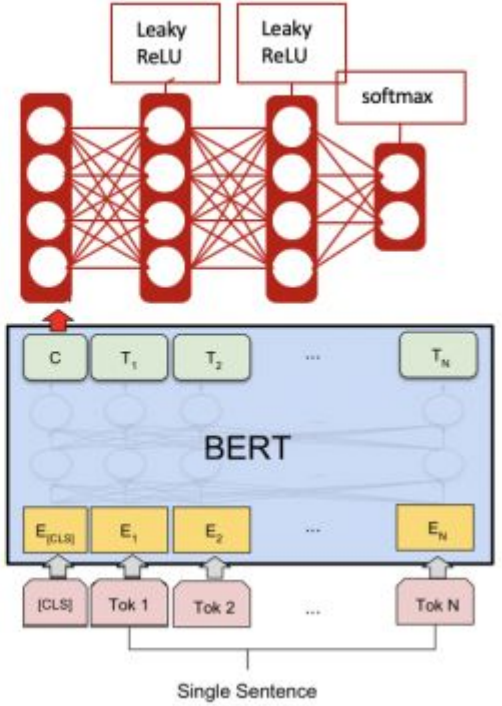
(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

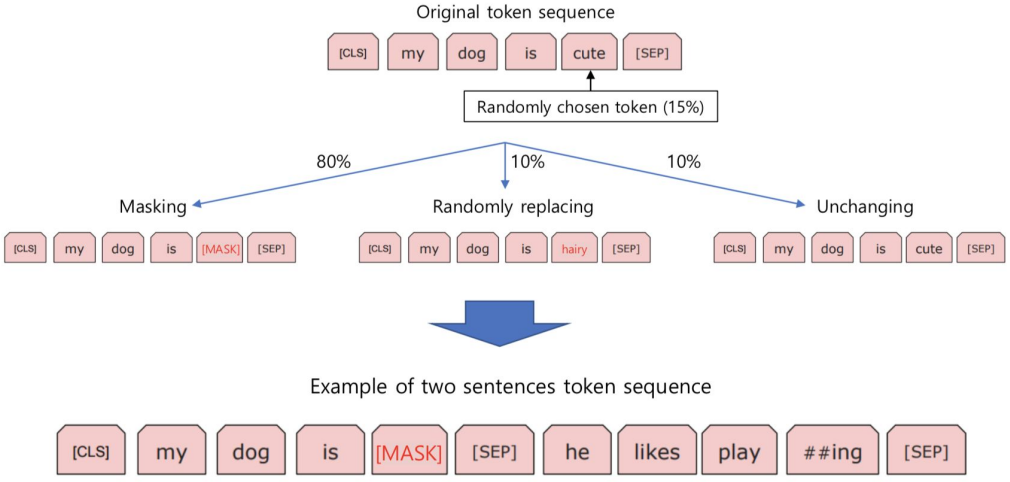
Fine-Tuning (10)





Masked Language Model (MLM)

- Masked Language Model(MLM)은 문장 내 랜덤하게 단어를 마스킹하고 이를 예측
- 마스킹은 전체 단어의 15% 적용
  - 마스킹 단어 중 80%는 <MASK>로 처리
  - 마스킹 단어 중 10%는 랜덤 단어
  - 나머지 10%는 정상적인 단어를 그대로 노출



Next Sentence Prediction (NSP)

- Next Sentence Prediction(NSP)은 두 문장을 입력하고 첫 번째 문장을 주고 두번째 문장이 다음 문장(Next Sentence)가 맞는지 여부를 판단
- 두 문장이 실제로 이어지는지 여부는 50% 비율로 참인 문장과 랜덤하게 추출되어 거짓인 문장의 비율로 구성되며, [CLS] 벡터의 Binary Classification 결과를 맞추도록 학습
  - 50% : sentence A, B가 실제 next sentence
  - 50% : sentence A, B가 corpus에서 random으로 뽑힌(관계가 없는) 두 문장
  - 예를 들어
    - 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

BERT - Input representation

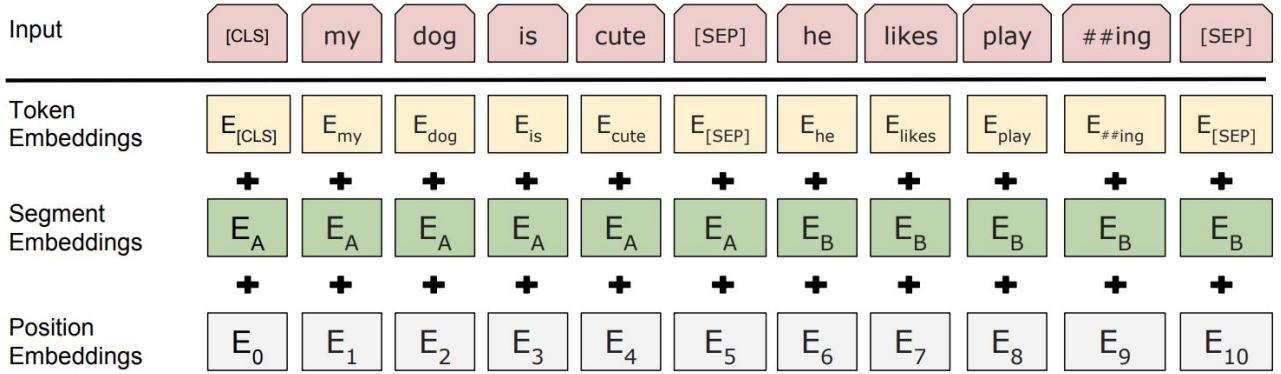
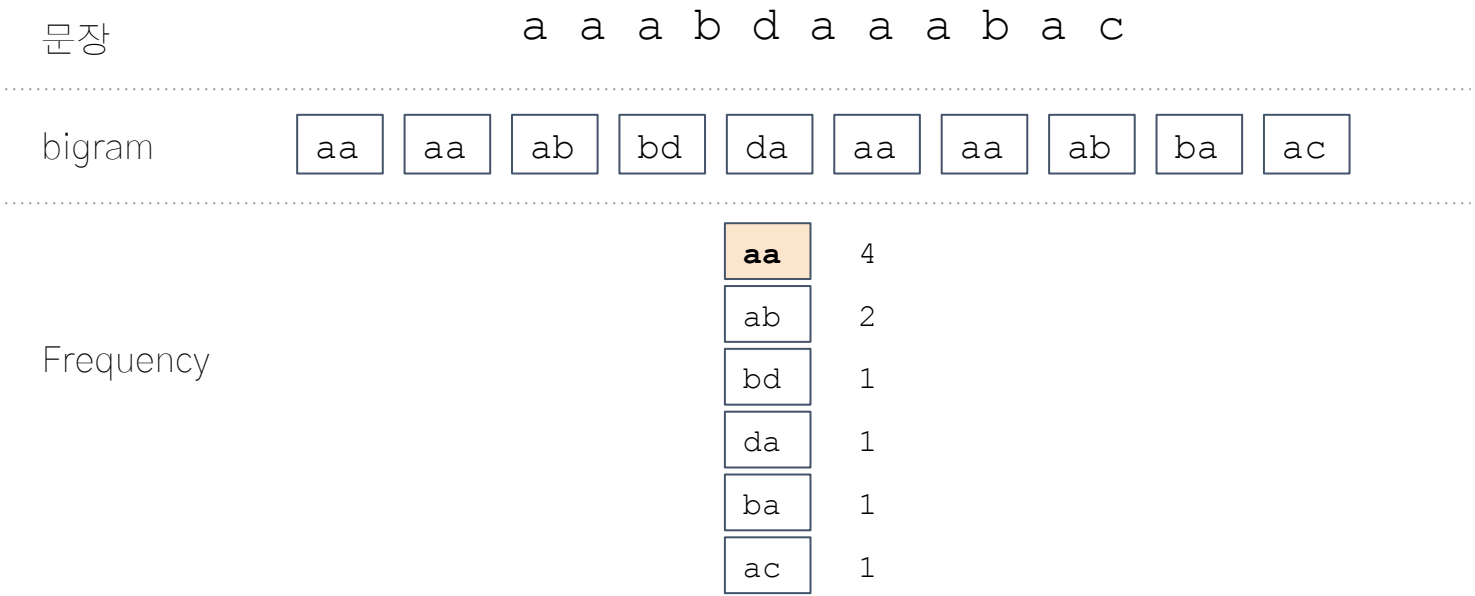


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

BPE(Byte Pair Encoding) 알고리즘

- BPE(Byte pair encoding) 알고리즘은 1994년에 제안된 데이터 압축 알고리즘
- 자연어 처리의 단어 분리 알고리즘으로 응용



Word Piece Model (1)

- BPE(Byte pair encoding) 알고리즘을 tokenizer에 적용

문장: 자연어 덮밥, 자연어 처리, 연어 덮밥

bigram

자	##연	##어	덮	##밥	자연	##연어	덮밥
자	##연	##어	처	##리	자연	##연어	처리
연	##어	덮	##밥		연어	덮밥	

Frequency

자연	2
##연어	2
덮밥	2
연어	1
처리	1

=> 처음으로 등장한 최빈토큰을 선택

Word Piece Model(2)

- BPE(Byte pair encoding) 알고리즘을 tokenizer에 적용

문장: 자연어 덮밥, 자연어 처리, 연어 덮밥

bigram

자연	##어	덮	##밥		자연어	덮밥
자연	##어	처	##리		자연어	처리
연	##어	덮	##밥		연어	덮밥

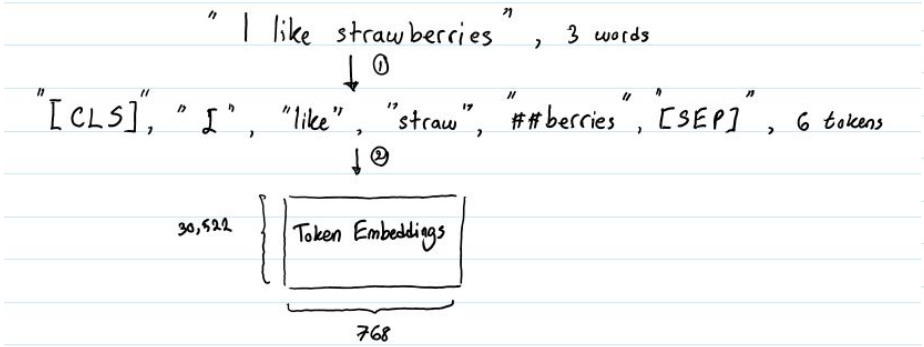
Frequency

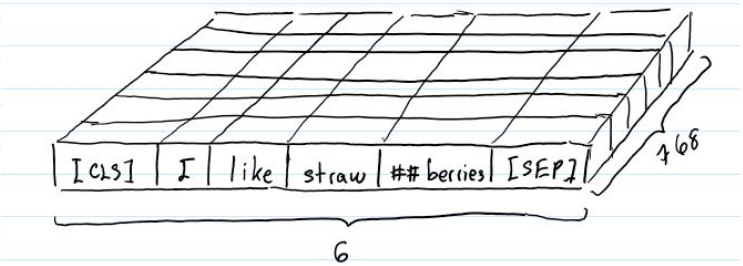
자연어	2
덮밥	2
처리	1
연어	1

=> 처음으로 등장한 최빈토큰을 선택

Token Embedding

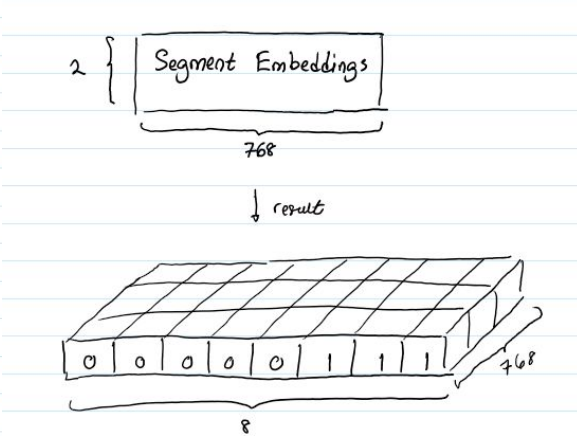
- 토큰 임베딩 : 단어를 고정된 차원의 벡터로 표현
- **BERT의 경우 각 단어는 768 차원 벡터로 표시**





Segment Embedding

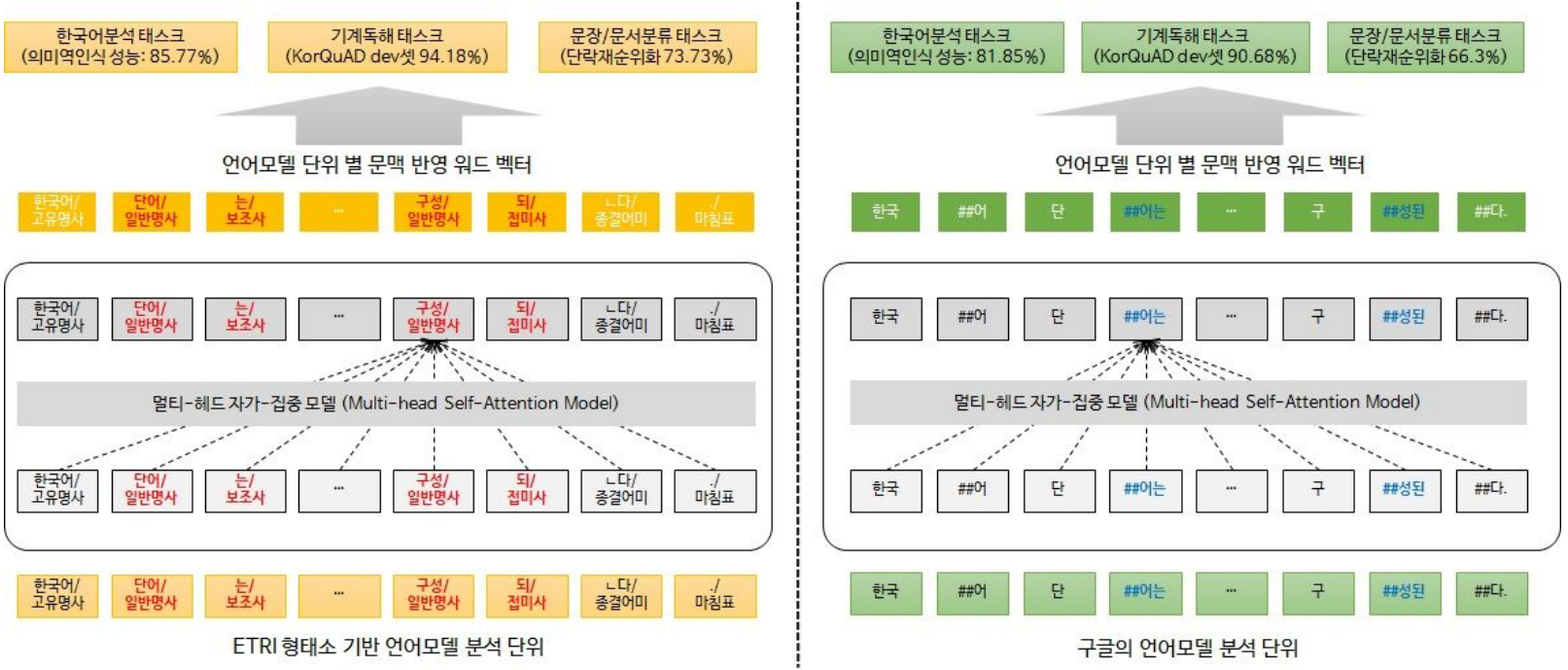
- 입력 문장이 2개인 경우
  - 첫번째 문장의 벡터는 0을 속하는 모든 토큰에 할당
  - 두번째 문장의 벡터는 1을 속하는 모든 토큰에 할당
- 입력 문장이 1개인 경우
  - 문장의 벡터는 0을 속하는 모든 토큰에 할당



Position Embedding

- Transformer의 positional encoding 참고

KoBERT



예문: 한국어 단어는 형태소로 구성된다.

〈ETRI 형태소 기반 언어모델과 구글 언어모델 비교〉

http://aiopen.etri.re.kr/service_dataset.php

감사합니다.

Insight campus SeSAC

daniel@fins.ai