KUBIG 23-1 NLP 분반 논문리뷰

BERT 기반 Model 논문 리뷰

2023.02.23

16기 이영노

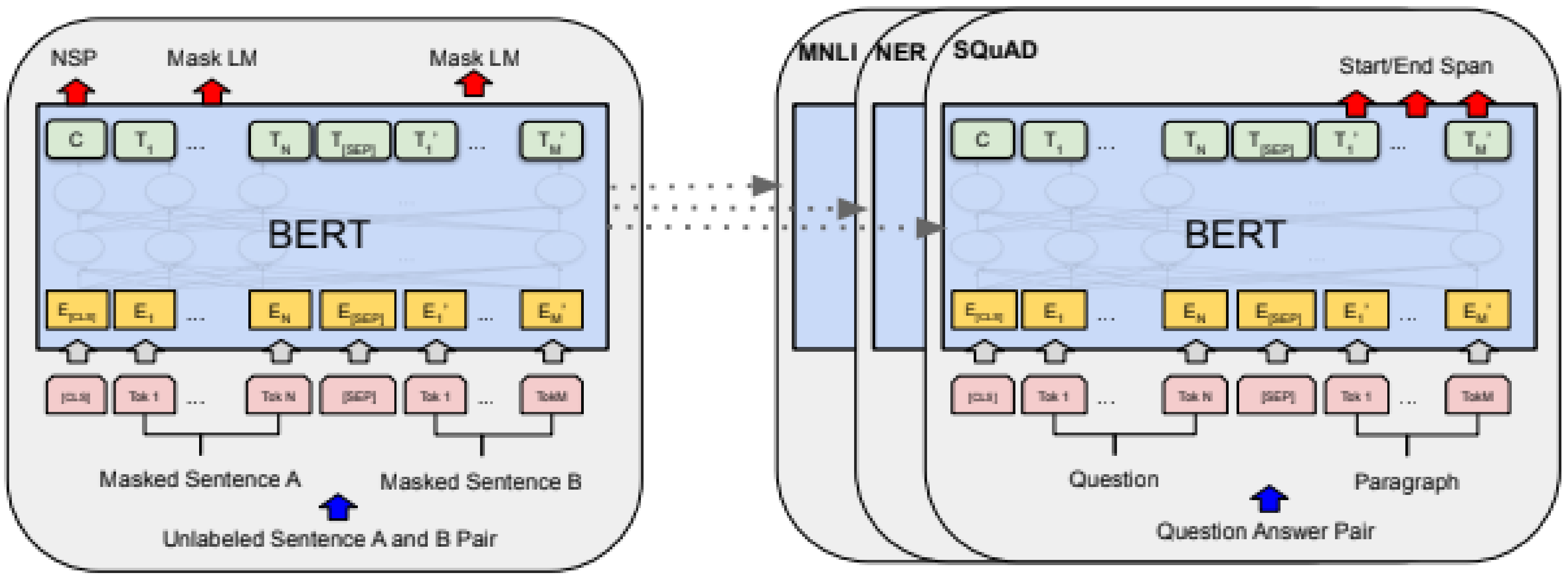
Table of Contents

Motivation

- Two approaches for Pre-Training

- Feature based approach : Task Specific (Embedding: ELMo)
  기존 input에 pre-trained representation 을 feature로서 추가

- Fine-tuning approach : Task Agnostic (GPT, BERT)
  최소한의 task specific parameter를 추가하여
  모든 pre-trained parameter를 조금만 바꿔주는 방식



Pre-training Fine-Tuning

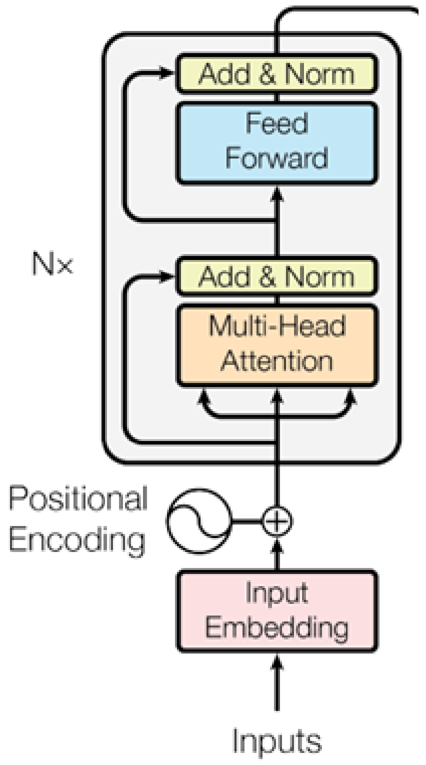
1) Architecture

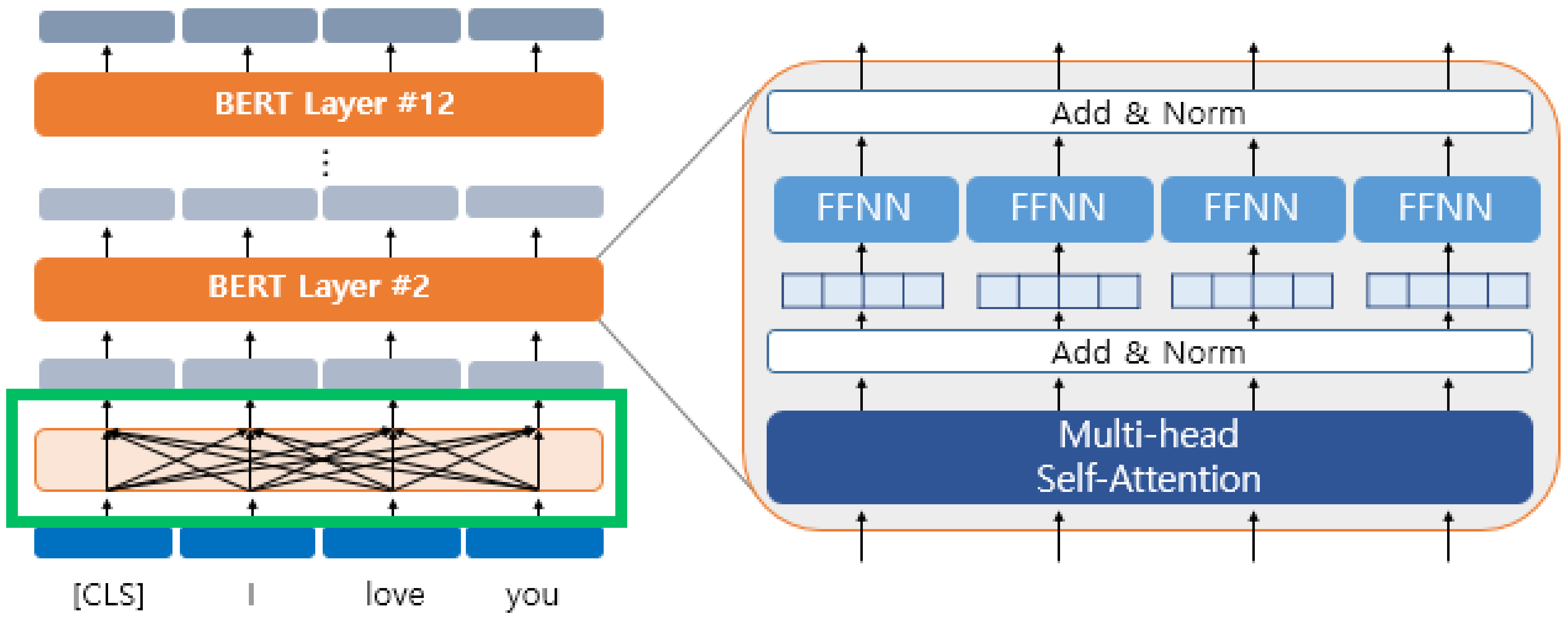
- Transformer의 Encoder부분만 사용하여 Self Attention을 통해 문맥을 학습

- Parameters
- L = number of Layers (Transformer block)
- D = d_model
- A = number of Heads in Multi-head Attention layer

cf. same model size as GPT-1 for comparison

	Transformer	BERT
L	6	12
D	512	768
A	8	16



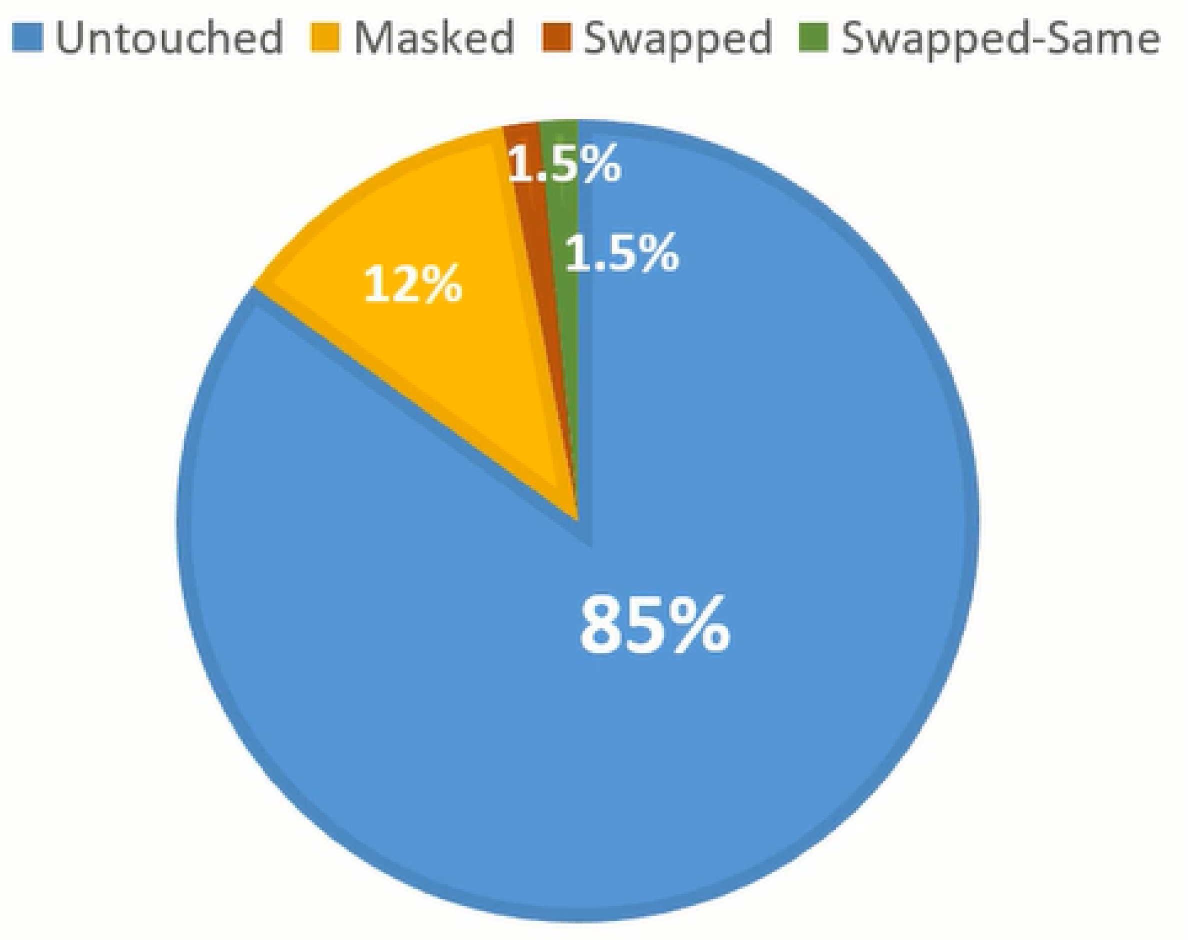


2) Pre Train 방식

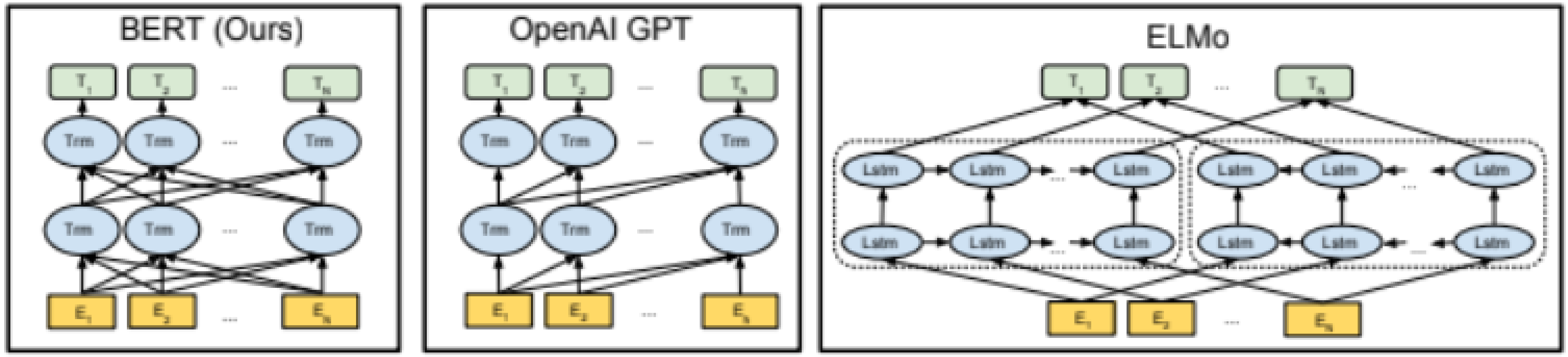
2-1) Masked LM : 임의의 순서에 해당하는 부분을 masking

Q. input은 masking되지 않은 raw data. 그럼 masking을 어떻게?

A. Train data generation for training and fine-tuning
- choose 15% of random token positions for prediction
  from the chosen i-th token position
  1. replace token with [MASK] for 80%
  2. replace token with [random token] for 10%
  3. replace token with [original token] for 10%
     (Fine-tune 할때는 fine tune input 이masking된 데이터가 아니므로,
     pre-train data와 gap이 생김. 이를 3번을 통해 해결)



cf. suitable for NER, MNLI
cf. comparison btw ELMo, GPT and BERT



- ELMo : Biderictional 한 sequential input
- BERT : Biderctional 한 non sequential input

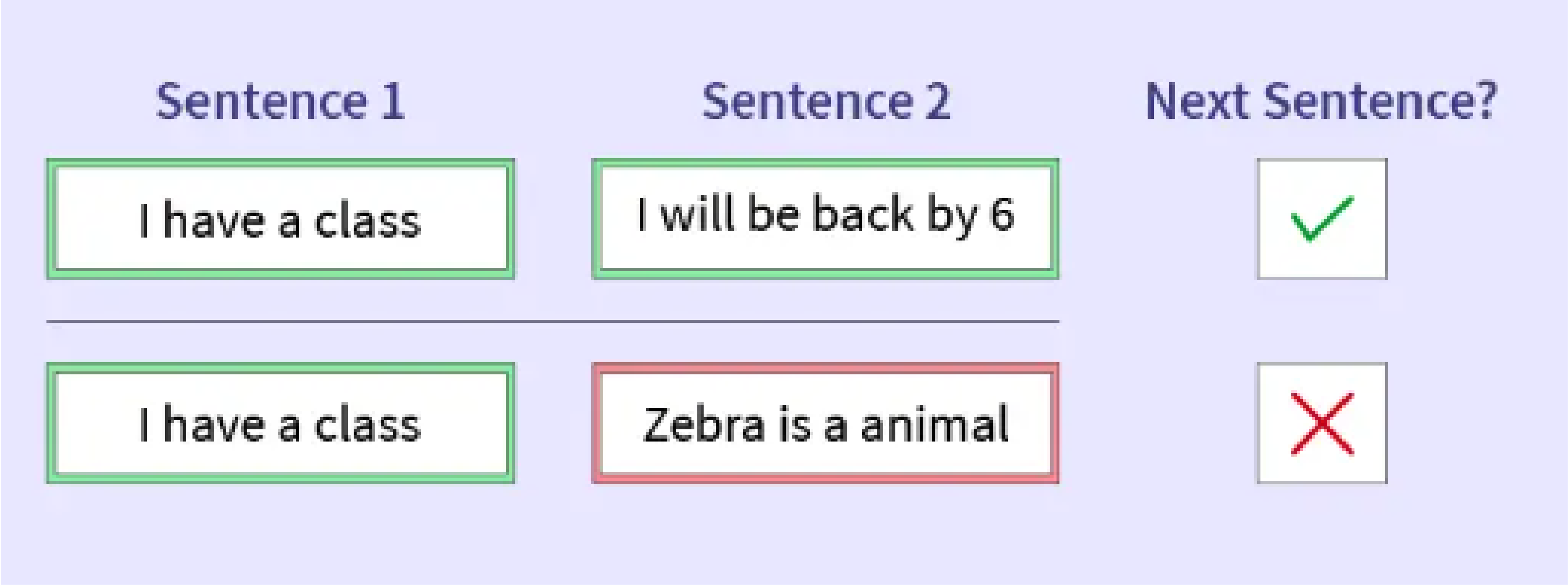
- BERT : Masking 이 random 하게 (bidirectional)
- GPT : Masking이 자기보다 뒤쪽에만(unidirectional)

2) Pre Train 방식

2-2) Next Sentence Prediction :

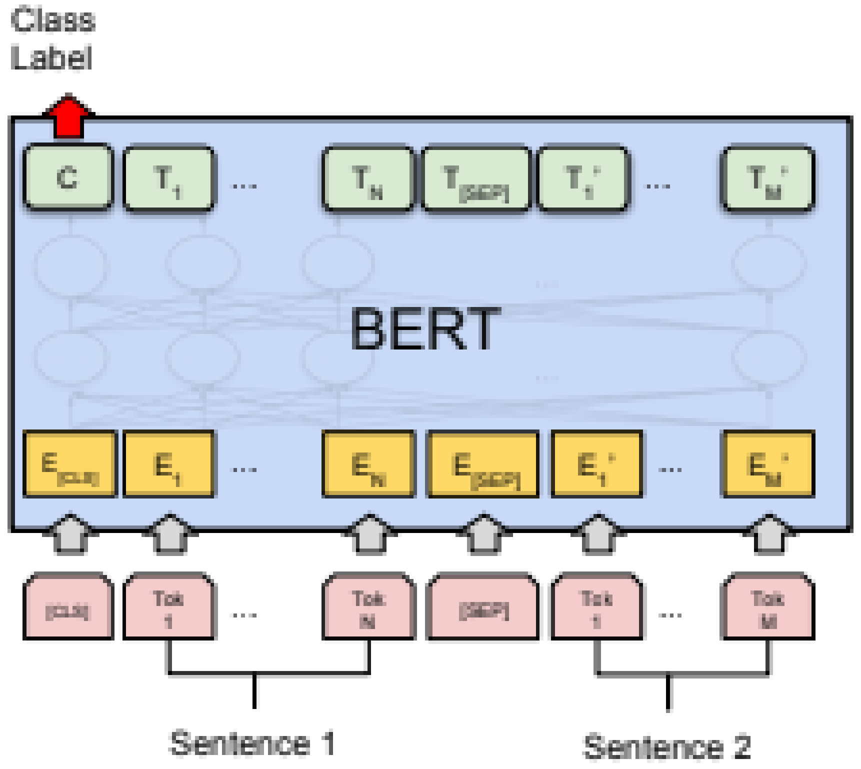
- 문장 간 관계를 고려해주기 위한 방법
- IsNextSentence : 1, NotNextSentence : 0 으로 labeling해줌
- 두개의 비율은 5:5로 설정

cf. suitable for QA, NLI

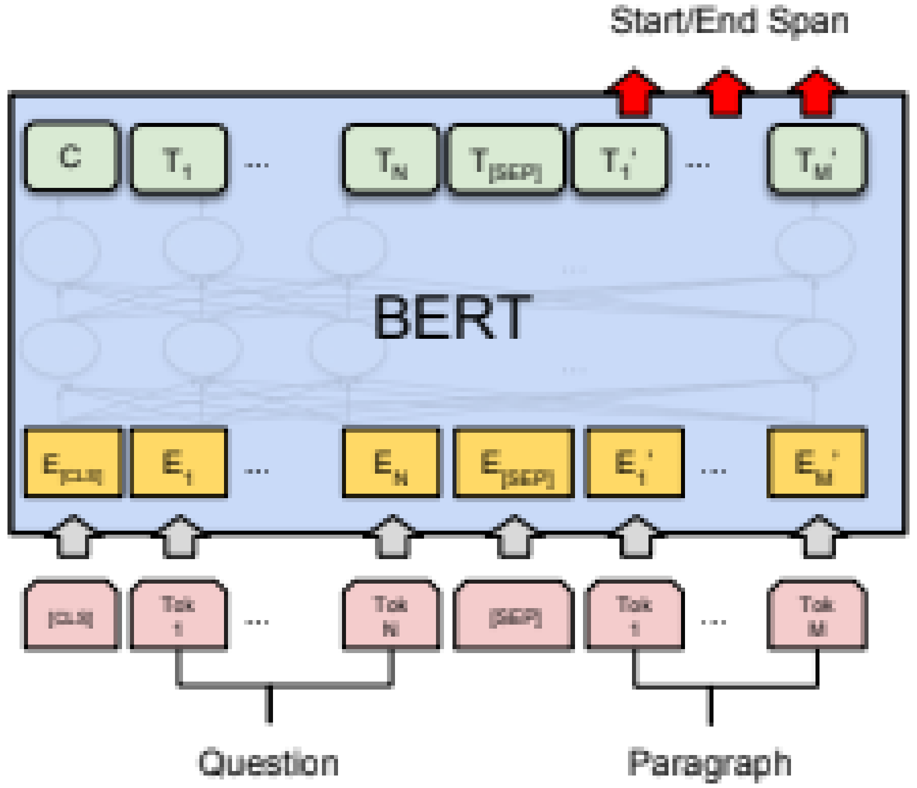


3) Input/Output Representation

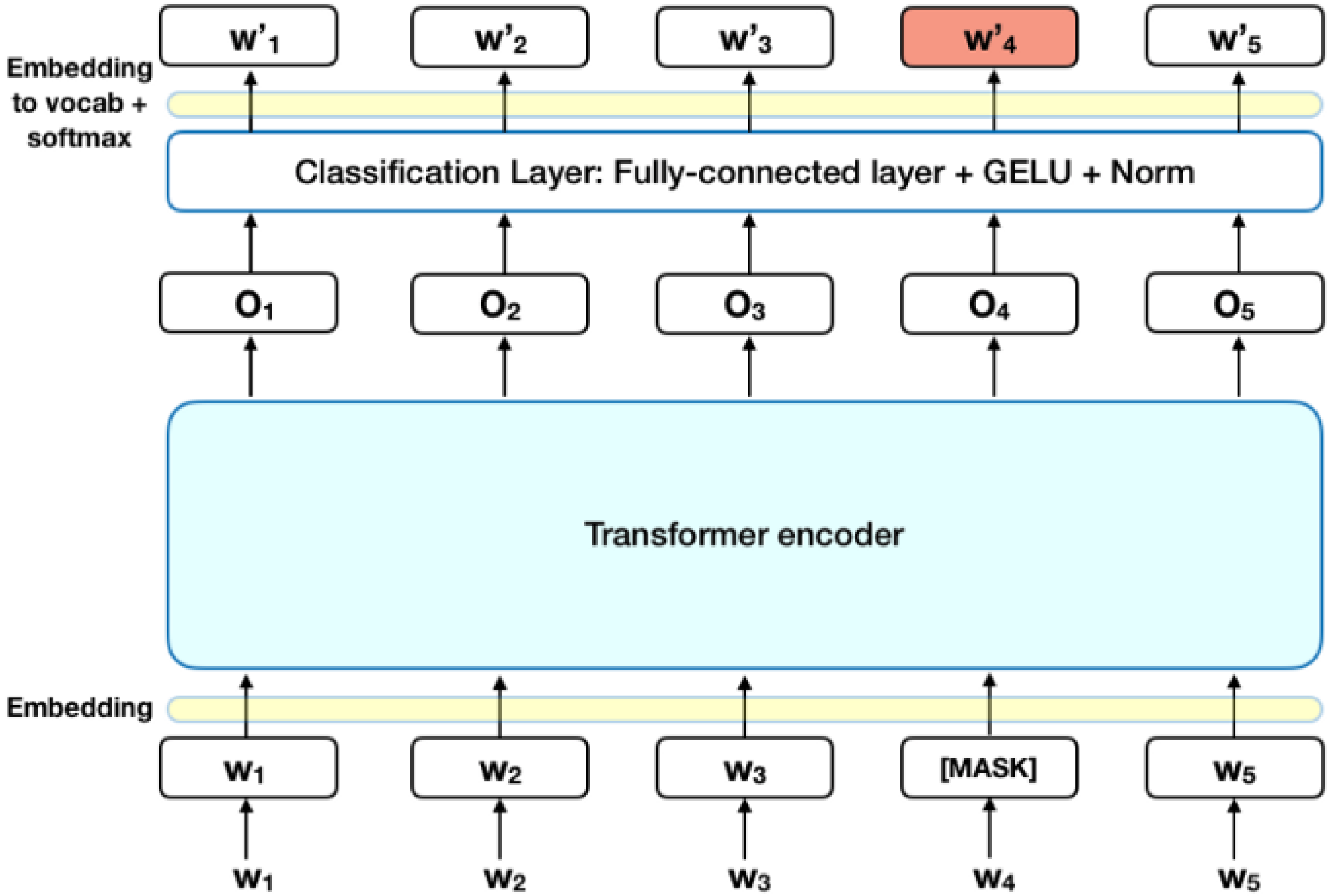
- Sentence : 통상적인 문장의 의미X, 문장의 연속
  arbitrary span of contiguous text
- Sequence : set of sentences (sentences packed together)
- C : binary TASK (sentiment classification, similarity, NSP, etc)
- T : final hidden vectors (prediction, QA)



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

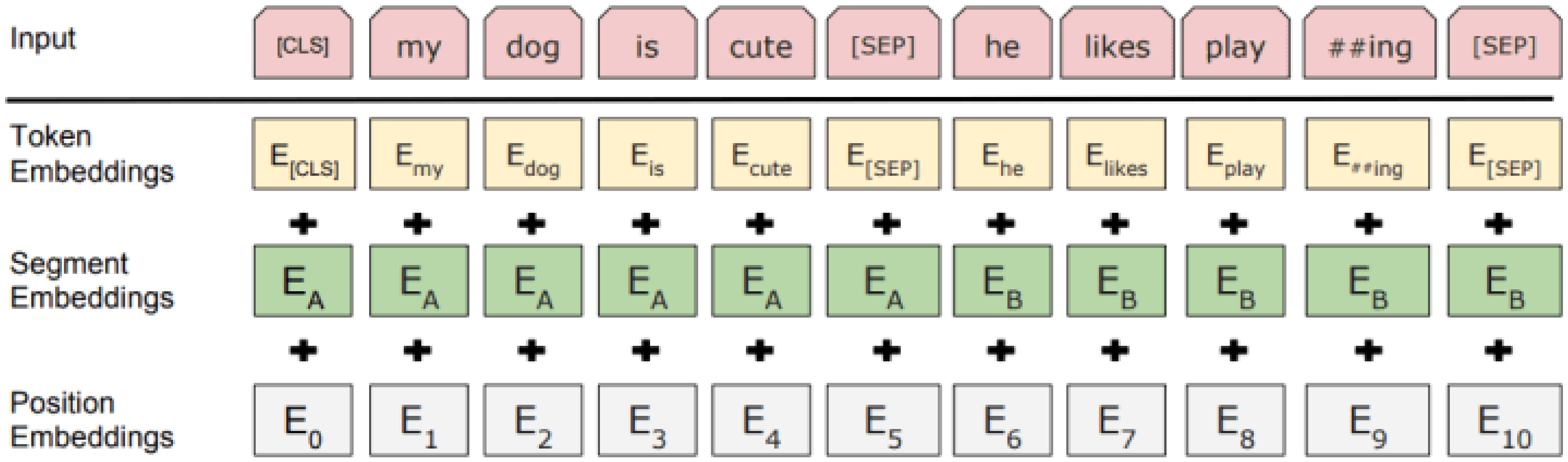


(c) Question Answering Tasks: SQuAD v1.1



4) Embeddings

- Token Embedding : Word-Piece subword tokenizer 30522
  Deal with OOV problem
- Position Embedding : 순서위치 정보(학습가능) 512
- Segment Embedding : 문장을 구분해주기 위한 임베딩 2



1. Pre Train Method
사전훈련 방식 개선을 통한 성능 향상
spanBERT, RoBERTa, ELECTRA, XLNet, ALBERT, BART, GPT3, T5

2. AE + AR

AE의 문제점	• [MASK] token이 독립적으로 예측 (independent assumption)되기때문에 token사이의 dependency는 학습할 수 없음 • Finetuning 과정에서 [MASK] token이 등장하지 않기때문에 pretraining과 finetuning사이에 discrepancy 발생
AR의 문제점	• 단일 방향 정보만 이용하여 학습 가능함

XLNet, BART, T5, DeBERTa-MT

3. Model Efficiency : 더 적은 parameter, 더 적은 computation
ELECTRA, ALBERT

4. META Learning
GPT3, T5

Motivation

- **BERT Pre Train 방식 : MLM, NSP**
- "Denver Broncos" 가 "NFL팀" 인지 결정하는 것은 "어떤 NFL팀이 슈퍼볼 50에서 우승했습니까? 라는 질문에 대답하는데 중요함. 그러나 Denver Broncos는 두개의 단어가 합쳐진 형태이기에 예측하는데 어려움이 있음
- Denver Broncos 를 한꺼번에 예측하는 것은, 각각의 단어를 예측하는 task보다 어려운 문제.

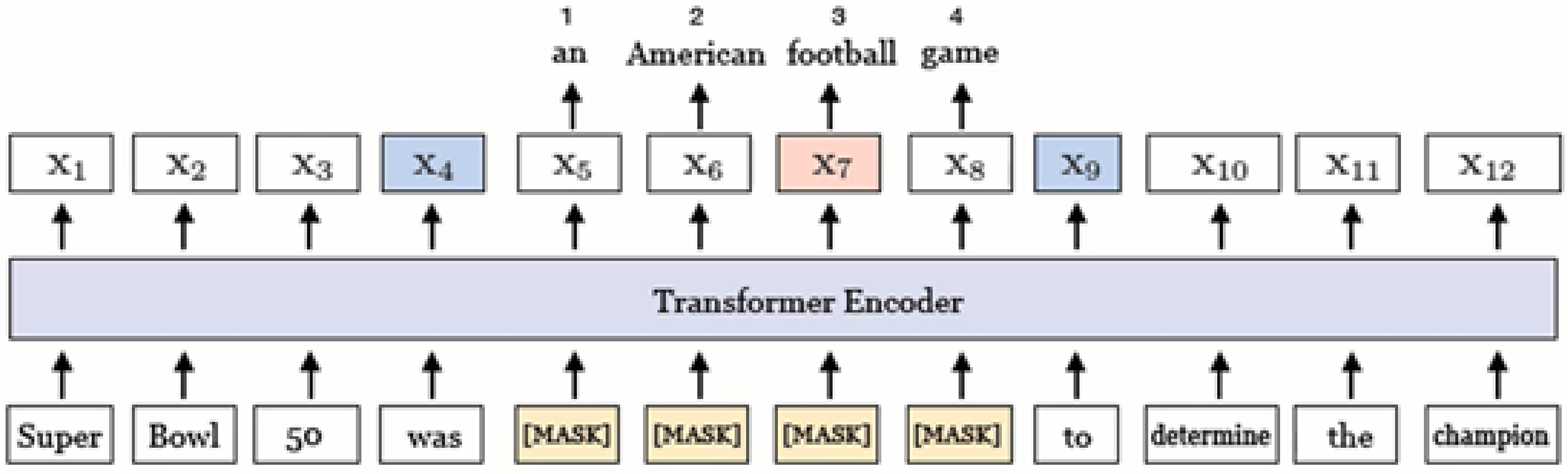
cf. QA, Conference Resolution(문맥군집화)

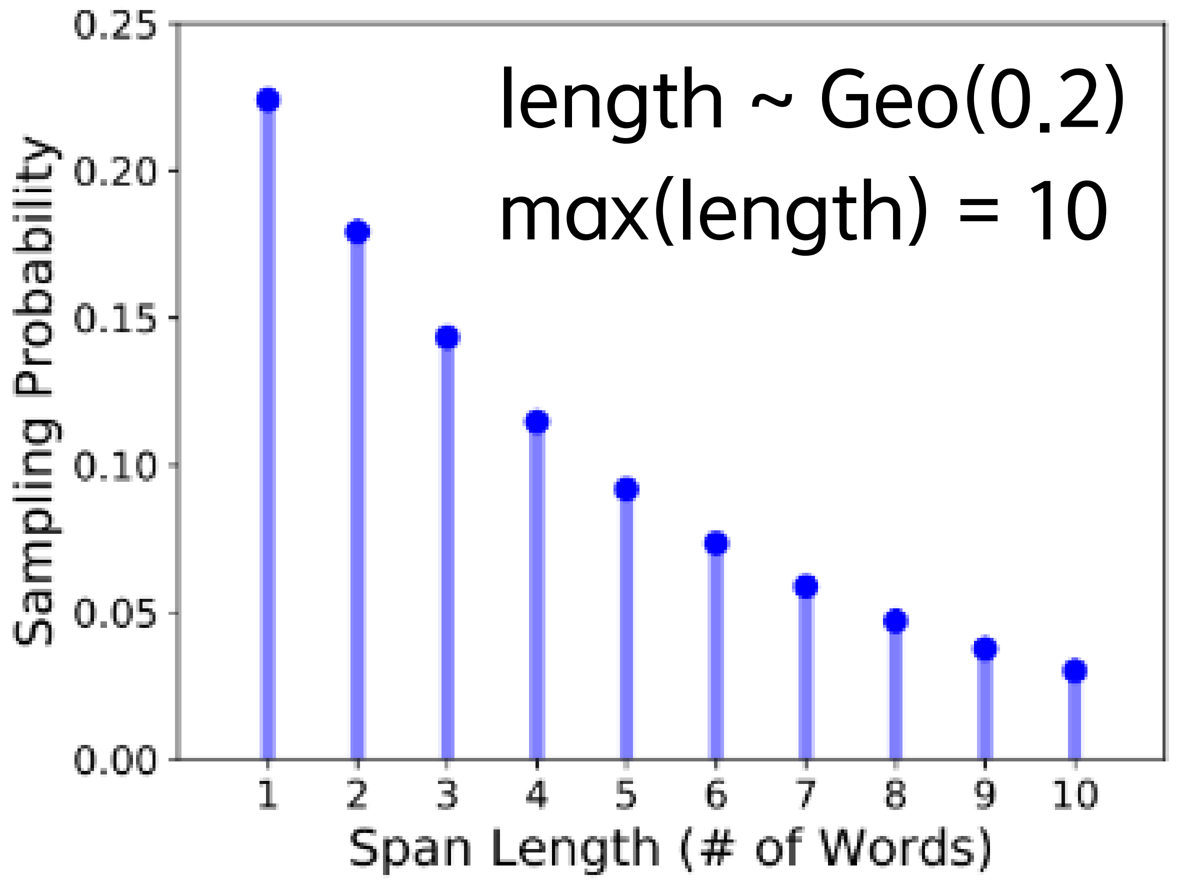
=> Pre Train 방식의 개선으로 해결
=> 성능 향상을 위해 NSP삭제, SBO추가

Model

- Span Masking : BERT의 random masking 대신에,
  한개 이상의 연속된 토큰을 masking
- mask span의 length ~ Geo(0.2) wh. max(length) = 10
- Span Boundary Objective : span masked 된 영역의 boundary에 있는
  x4, x9 토큰도 예측에 활용
- Single Sequence Training : NSP 삭제
  context form other document adds "Noise" to the MLM.

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid x_7) - \log P(\text{football} \mid x_4, x_9, p_3)$$





Motivation

- BERT는 "Under-Trained" 되었다는 가정에서 시작
  가장 "최적화된" BERT 모델을 만들어보자!
- Sampling bias from Random Masking
  BERT는 pre train 전에 raw data에 대해 masking 진행
  그러나 이 과정에서 random masking 에 기인한 sampling bias 발생
  동일한 token이 selected 되어 masking이 진행될 수 있음

  => 동일한 데이터에 대해 masking을 10번 다르게 적용
  => Input이 들어갈때마다 masking을 진행

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
Our reimplementation:			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Model

- more data, more batch size, longer sequence size
  (BERT는 d_model = 512, but Pre-Train 과정에서 512인 sequence를 10% 만 사용함)
- NSP 제거 (성능)
- Dynamic Masking :
  동일한 데이터에 대해 masking을 10번 다르게 시행
  Input이 들어갈 때마다 masking 진행
- Byte Pair Encoding : 빈도수에 기반하여 가장 많이 등장한 쌍을 병합

$$\mathbf{y}_i = f(\mathbf{x}_{s-1}, \mathbf{x}_{e+1}, \mathbf{p}_{i-s+1})$$

$$\begin{aligned}\mathbf{h}_0 &= [\mathbf{x}_{s-1}; \mathbf{x}_{e+1}; \mathbf{p}_{i-s+1}]\\ \mathbf{h}_1 &= \text{LayerNorm}(\text{GeLU}(\mathbf{W}_1\mathbf{h}_0))\\ \mathbf{y}_i &= \text{LayerNorm}(\text{GeLU}(\mathbf{W}_2\mathbf{h}_1))\end{aligned}$$

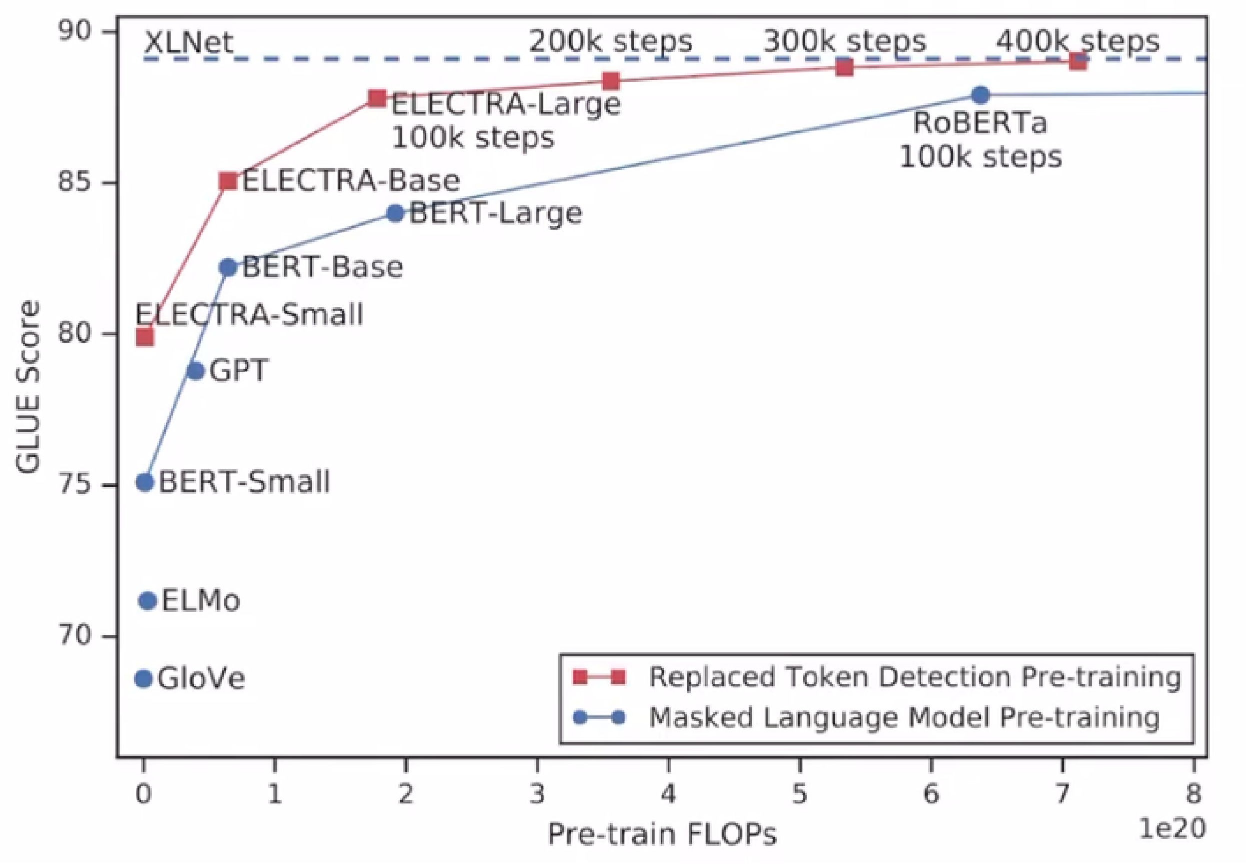
	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT$_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet$_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
BERT$_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNet$_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Motivation

- 전체 토큰중 masked된 15%에 대해서만 학습 (비용)
- pre train 시에 mask token을 참고하지만, 실제로 이것은 학습을 위한 변형일뿐 실제로는 mask token 이 존재하지 않아 fine tune시 gap 발생.

=> Generator을 이용해 input의 일부 토큰을 예측
=> Discriminator가 binary classification을 통해 전체 데이터를 학습
(전체 데이터에 대해 학습)
(ELECTRA-Small 의 경우 BERT-Large대비 1/20 parameter,
1/4 computaion 으로 GLUE Score가 5 point 더 좋은 성능을 보였음)



Model

- Replaced Token Detection(RTD) : 15%가 아닌 전체 데이터를 학습
  Generator, Discriminator 모두 Transformer Encoder구조(문맥 정보 반영)
- Generator만 학습 - 학습된 weight로 Discriminator 초기화 - Discriminator 학습