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

Jacob Devlin, Ming Wei Chang, Kenton Lee, Kristina Toutanova

TABLE OF CONTENTS

01 연구 배경

02 연구 목적

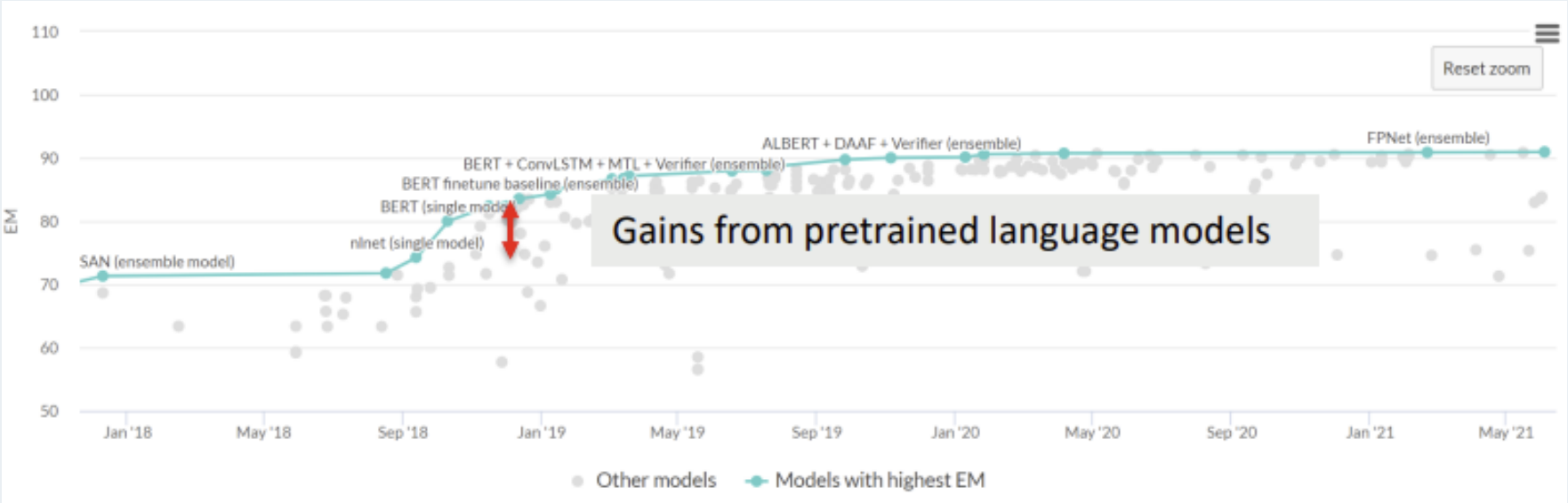
03 BERT 설명

04 실험 방법

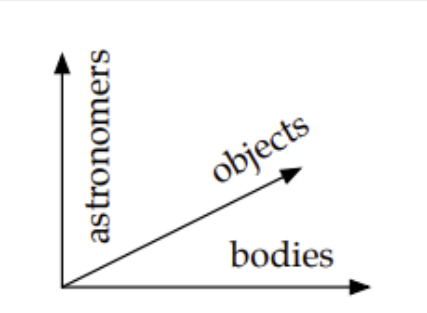
05 실험 결과

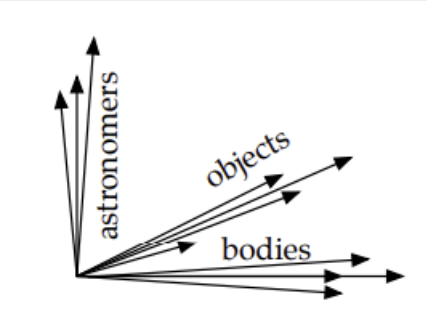
06 세부 설명

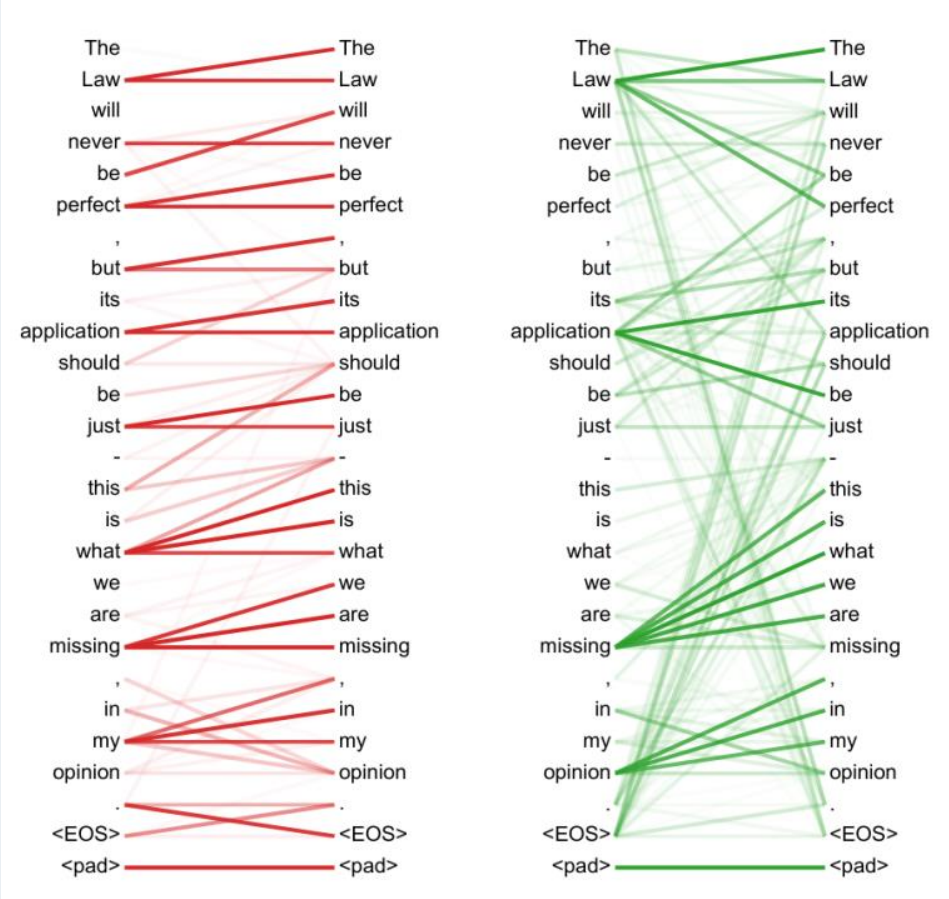
BACKGROUND


110
100
90
80
70
60
50
EM
Reset zoom
SAN (ensemble model)
nlnet (single model)
BERT (single model)
BERT finetune baseline (ensemble)
BERT + ConvLSTM + MTL + Verifier (ensemble)
ALBERT + DAAF + Verifier (ensemble)
FPNet (ensemble)
Gains from pretrained language models
Jan '18
May '18
Sep '18
Jan '19
May '19
Sep '19
Jan '20
May '20
Sep '20
Jan '21
May '21
Other models
Models with highest EM

BACKGROUND


astronomers
objects
bodies


astronomers
objects
bodies


The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>

Pre-Training

기존: Random한 distribution으로 모델의 가중치를 초기화

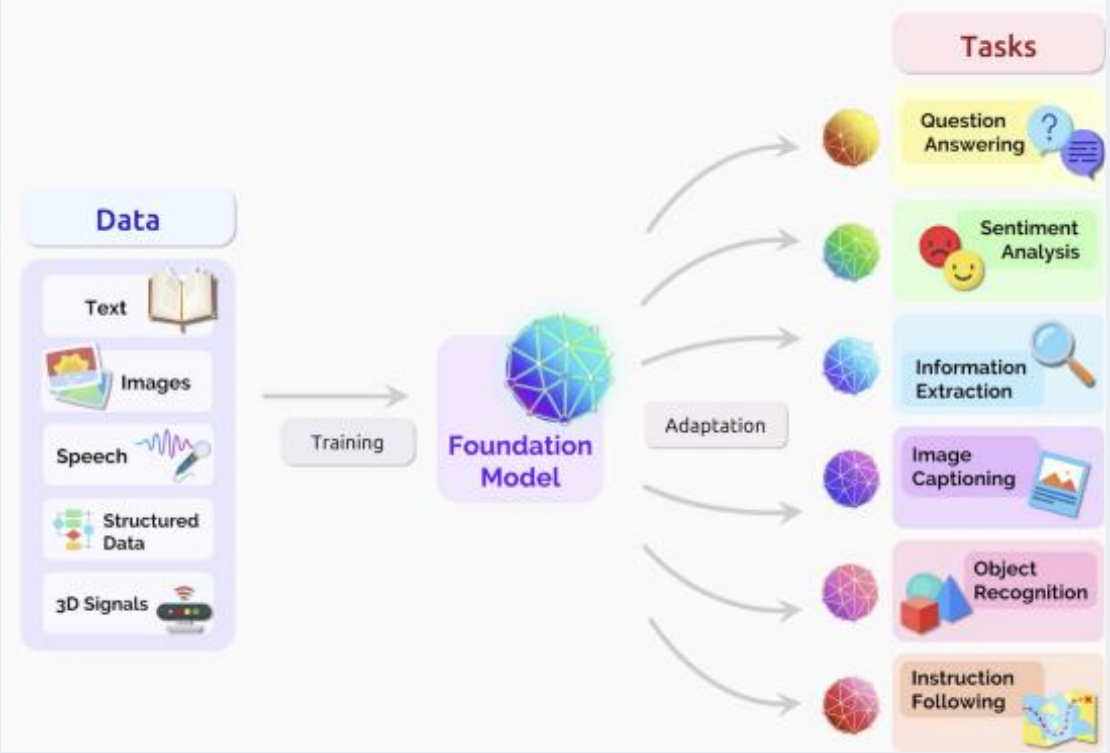
사전 학습: 하나의 Task에 학습시킨 가중치를 다른 task를 위한 모델 가중치로 초기화하는 것

↓

Transfer Learning

Pre-training + Fine tuning: 사전 학습된 모델 가중치를 원하는 Task에 학습시키는 것

1. 사전 학습을 방대한 데이터로 unsupervised 방식으로 학습 → 모델이 언어를 전체적으로 파악
2. 사전 학습 가중치를 사용하여 적용하려는 Task (downstream task)에 대해 supervised 방식으로 학습



https://arxiv.org/pdf/1511.01432.pdf

Pre-trained Language Model

Main Approaches

1. Feature based (ELMO): Downstream task에 적합한 모델 구조에 사전 학습된 가중치를 적용하는 것

2. Fine tuning based (GPT): 사전 학습된 가중치를 downstream task에 적용하여 모든 가중치를 task에 맞게 학습시키는 것

문제점: unidirectional language model 사용

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

연구 목적

연구 가정

Unidirectional language model을 이용하면, 양쪽 문맥를 고려해야 하는 token-level task에 취약해진다

Unidirectional Language Model

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

왼쪽 문맥만 고려

“The strong ____ is annoying but it also provides warm and light”

a) Wind
b) Sun

양쪽 문맥을 모두 고려해야 한다

연구 목적

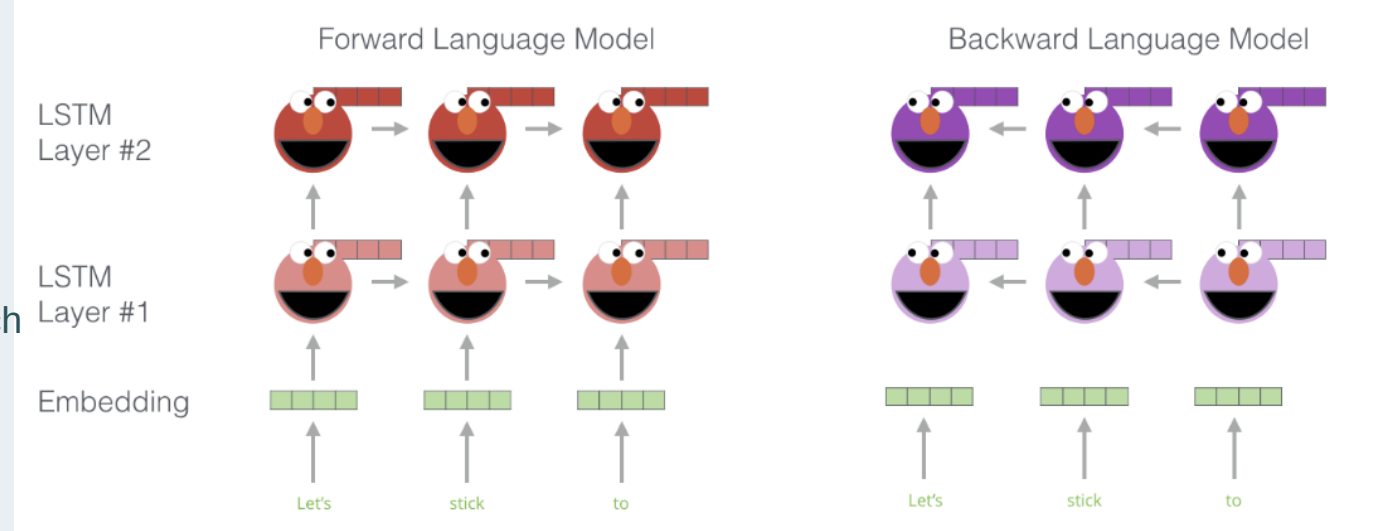
양쪽 문맥을 고려하는 Language Model 학습 방법을 통해 **fine-tuning based pre-trained language representation**을 개선시켜 사전학습 효과를 극대화

→ Downstream task 더 효과적으로 해결

ELMO

Feature-based approach

https://arxiv.org/pdf/1802.05365.pdf



Bi-LSTM을 통해 양쪽 문맥 고려하지만 Concatenate으로 문맥을 연결: 독립적으로 고려하게 된다 =Weak Connection

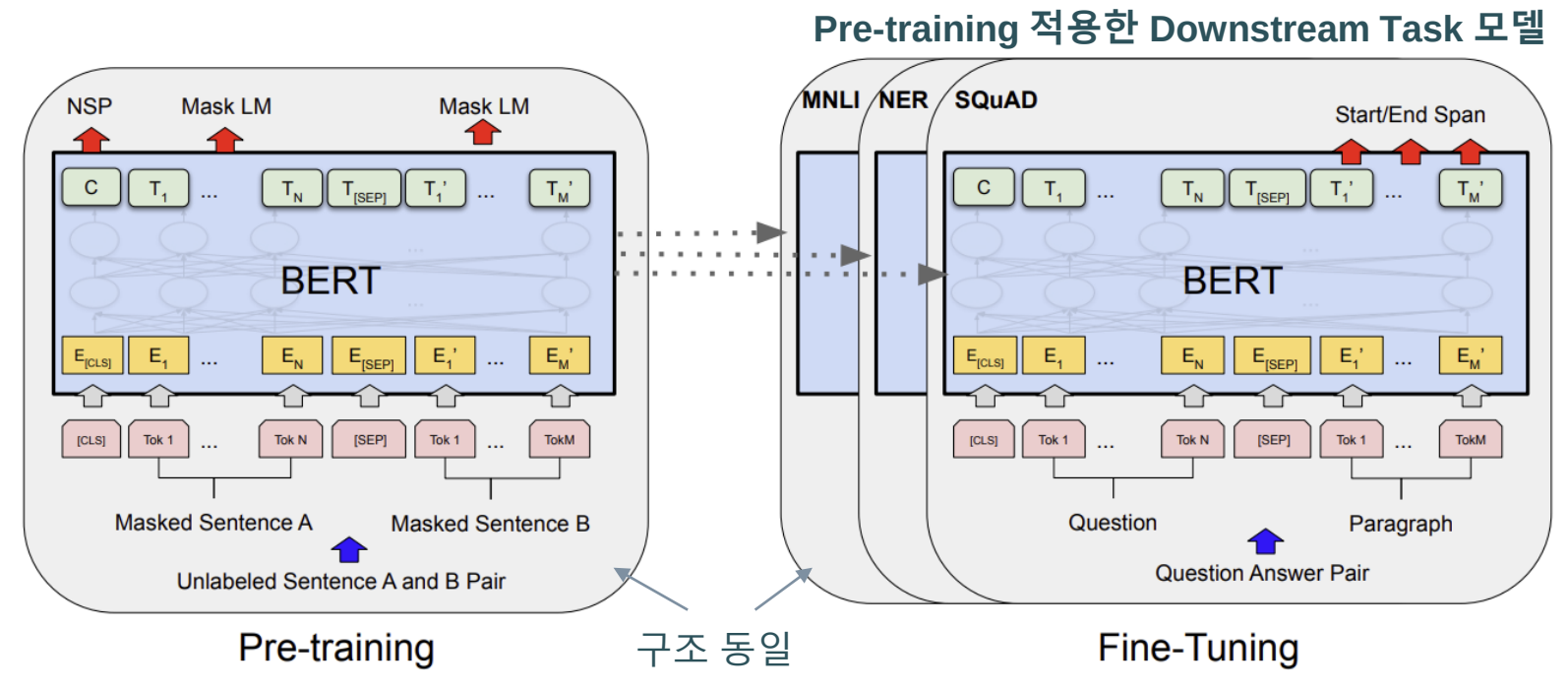
$$\sum_{k=1}^{N} \left(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \right).$$

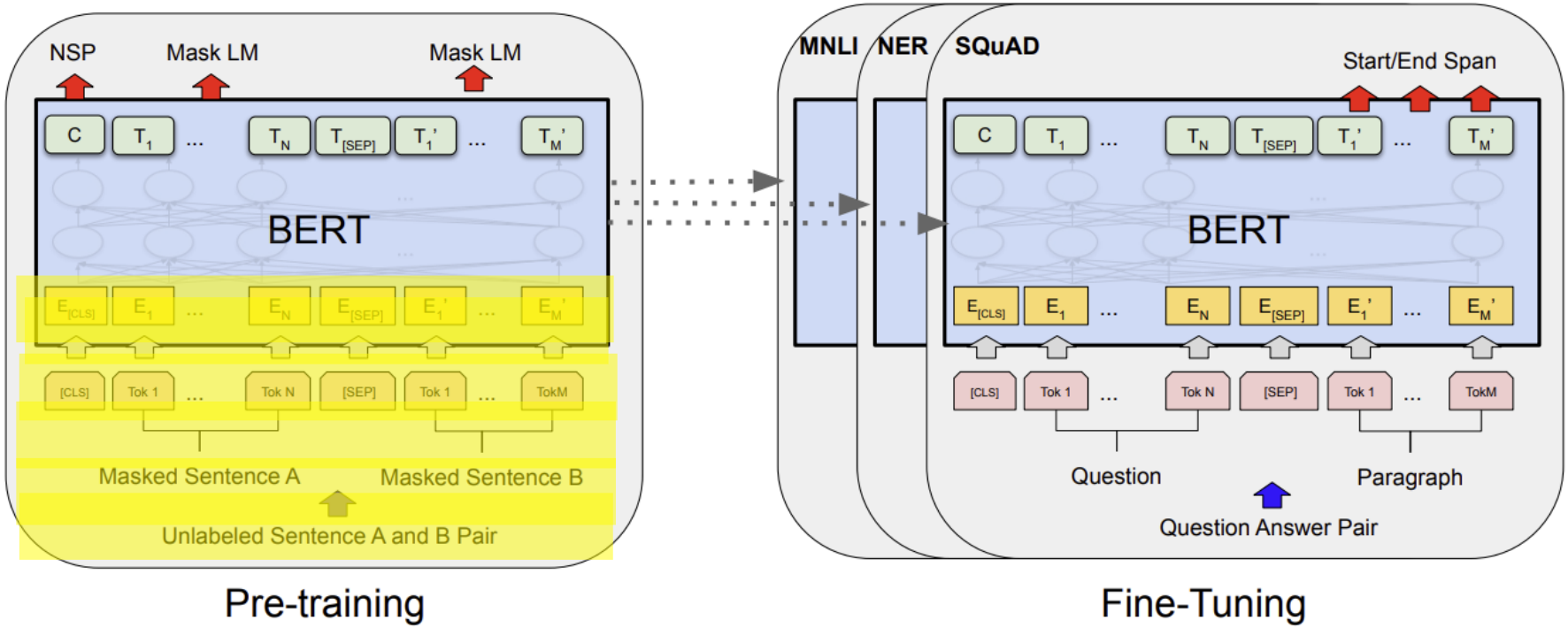
$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

deep bidirectional context 아님

BERT Bidirectional Encoder Representations from Transformers

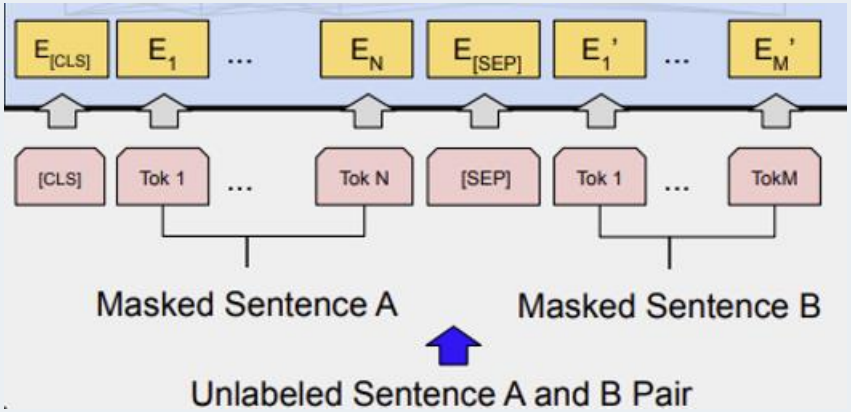
Multi layer bidirectional Transformer Encoder




NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Question
Paragraph
Question Answer Pair
Fine-Tuning

Input Representation

하나의 token sequence으로 하나의 Sentence 또는
하나의 Sentence 쌍을 표현



Input

단어
WordPiece Embedding 사용

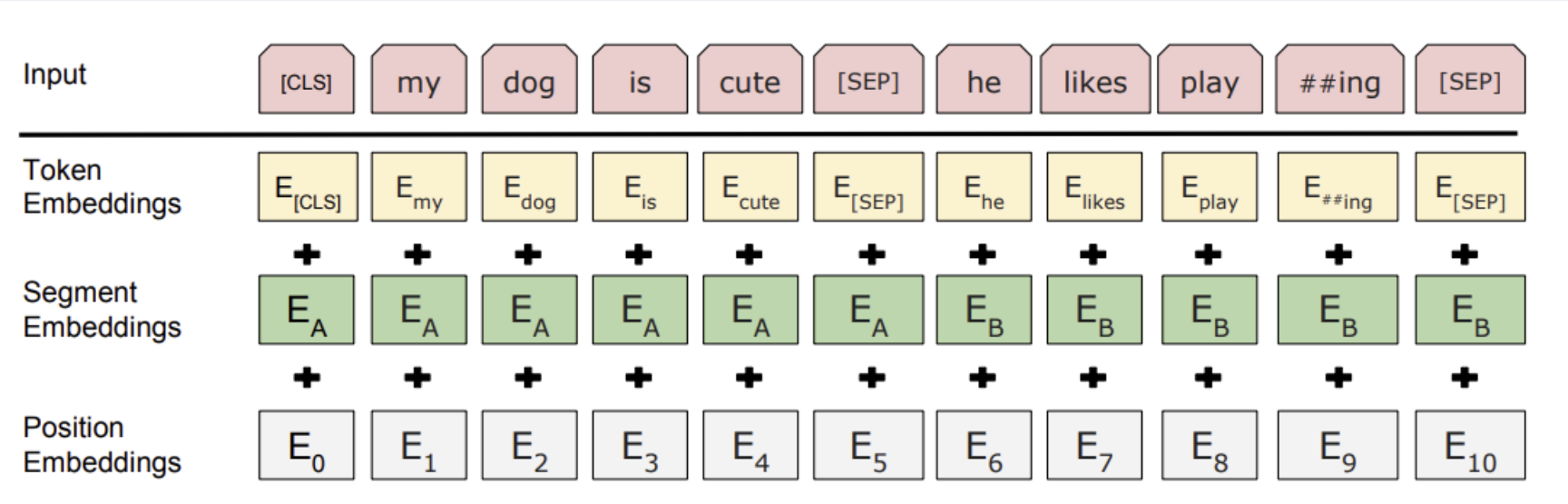
[CLS]: Classification의 약자,
입력한 문장의 전체적인
표현을 담기 위한 Token

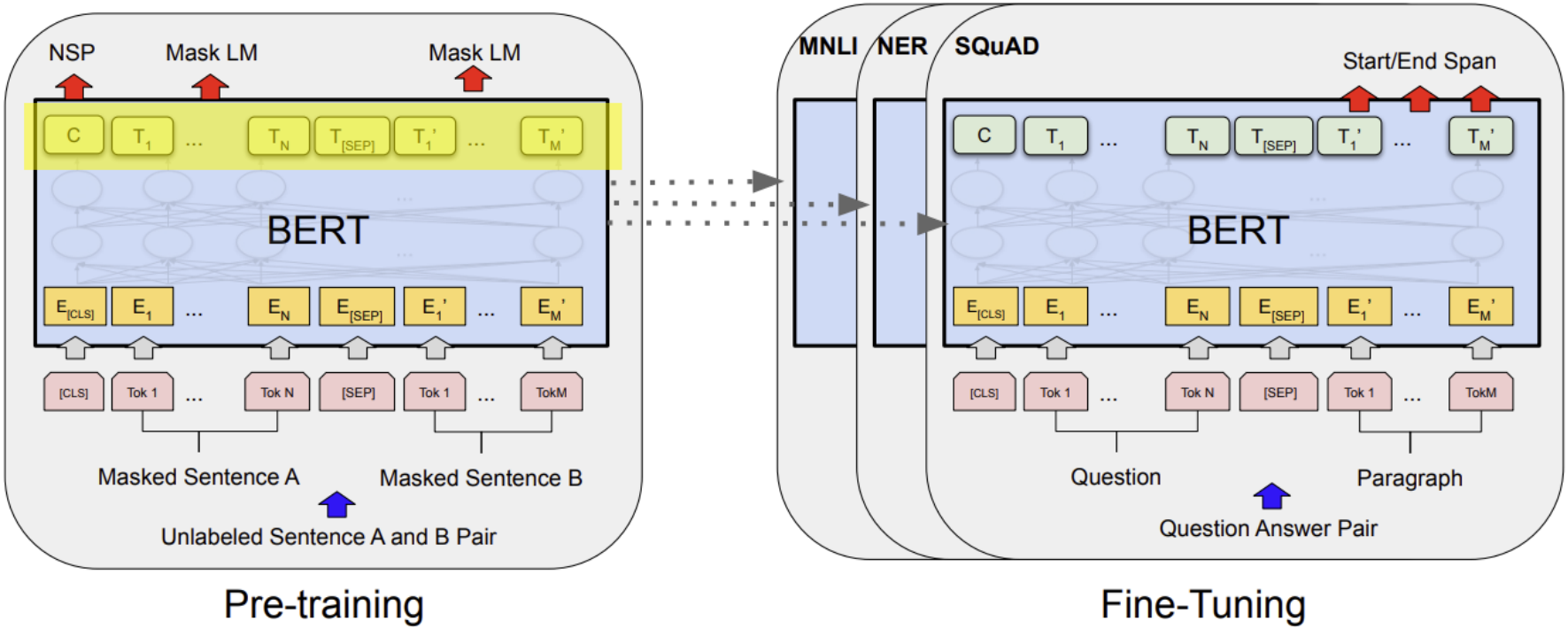
[CLS]: Classification의 약자, 입력한 문장의 전체적인
표현을 담기 위한 Token

두 개의 Sentence 구별
[SEP]: 두 문장을 구별하는 Token
+두 문장을 구별하기 위해 학습이 가능한 Embedding
matrix을 통해 문장 A,B 중 어떤 것인지 알려준다.
= 첫 번째 문장인지 두 번째 문장인지 판단하는 matrix

Input Representation

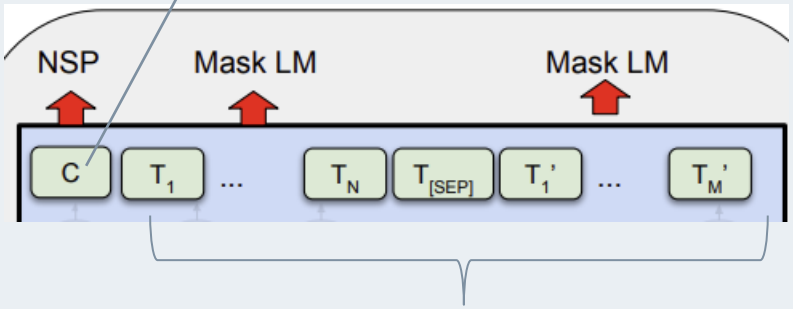
Input: Wordpiece embedding + segment embedding + position embedding




NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Question
Paragraph
Question Answer Pair
Fine-Tuning

Output Representation

[CLS]의 마지막 hidden state == Context vector처럼 문장의 전체적인 정보 담는다



각 input token들의 마지막 hidden state

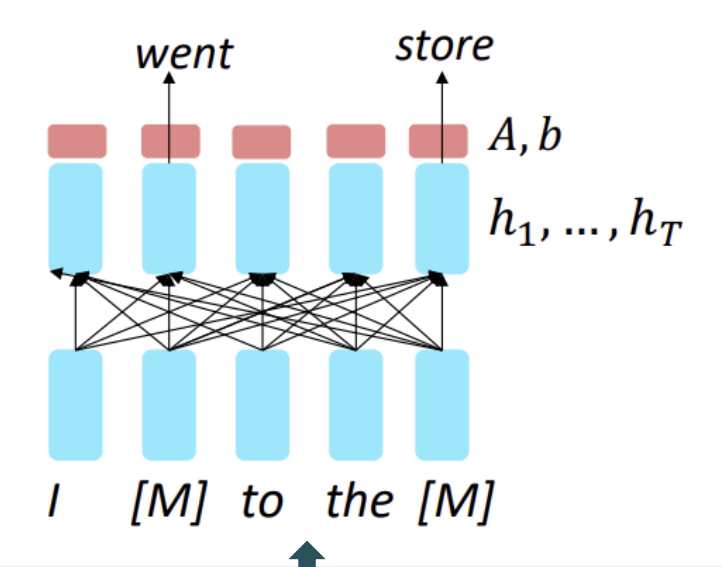
Pre-training BERT

Task 2개로 학습시킨다

1. Masked LM
2. Next Sentence Prediction

Pre-training BERT : Masked LM

Bidirectional **Encoder** Representations from Transformers



I went to the store

Bidirectional한 Context 고려하게 하는 방법

MLM (Masked Language Model)

To predict the original vocabulary of the masked word based only on its context

Allows representation to fuse the left and the right context

[Mask] token이 softmax에 들어가서 어떤 단어인지 예측할 수 있게 학습

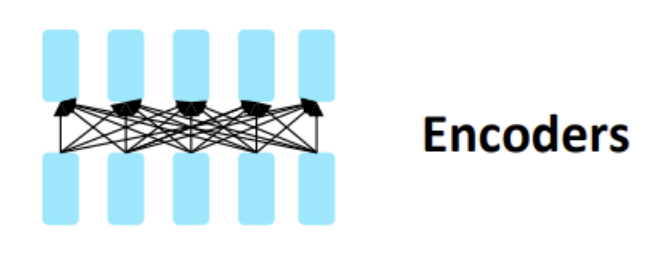
If $\tilde{x}$ is the masked version of x,we're learning $p_\theta(x|\tilde{x})$

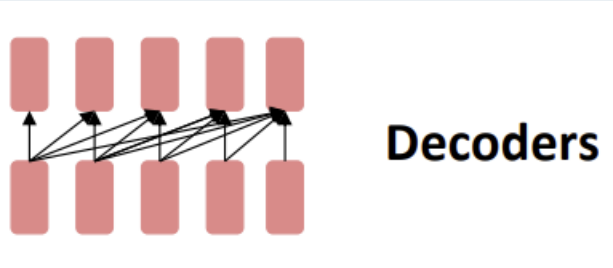
Pre-training BERT: Masked LM

MASK를 왜 할까?
→ Bidirectional Context 학습을 하게 하기 위해

Idea
Cloze task: Text에서 일부 단어가 빠져 있고 이 빠진 부분을 완성하는 작업
→ 언어 모델의 이해력과 문맥 파악 능력을 평가할 때 사용, **양쪽 문맥을 고려하면 쉽다**

Transformer Encoder를 통해 Context를 학습하자! → **Idea**
GPT 한계: Transformer **Decoders** based



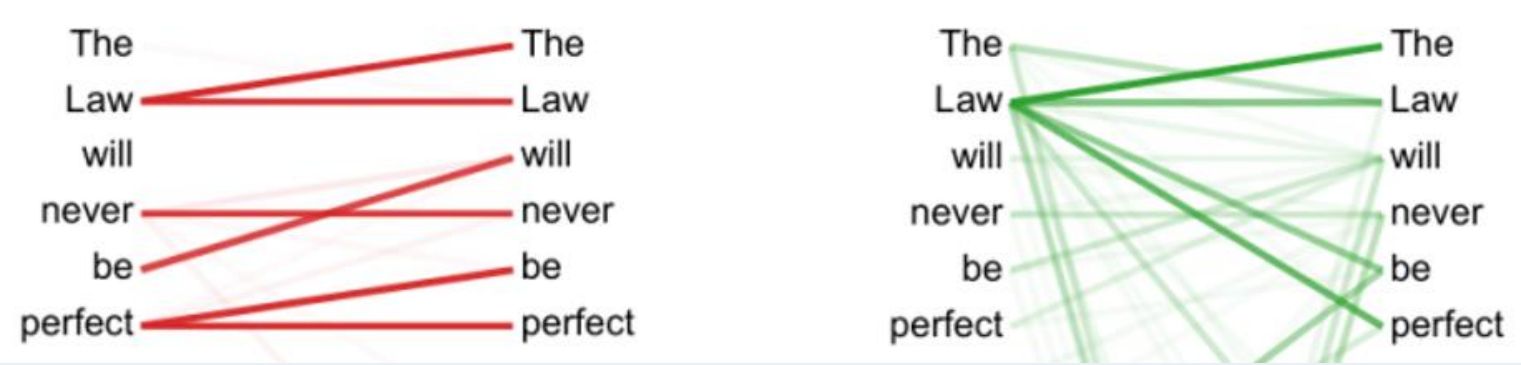


Pre-training BERT: Masked LM

Transformer Encoder를 통해 Context를 학습하는데 Mask하지 않는다면

Ex) The Law will never be perfect에서 Law를 예측한다고 할 때 이미 Law에 대한 정보가 자기 자신에 들어있다. → **Mask 해야함**

<Transformer Encoder 시각화>



Pre-training BERT : Masked LM

Mask 기준
각 input sequence의 단어들의 15%만 랜덤하게 [MASK] token으로 바꿔 Language model이 이 [MASK]를 예측하도록 한다

문제
Fine tuning 할 때는 [MASK]라는 토큰 없다
→ 사전학습 결과 model과 fine tuning 모델 간의 mismatch

개선
Masking 할 때, 빈칸으로 뚫을 단어의 80%는 [MASK]으로, 10%는 다른 단어로, 10%는 바꾸지 않는다.

Pre-training BERT: Next Sentence Prediction

** Sentence A,B 존재 이유

도입 이유 질의응답, 추론: 두 문장 간의 관계를 파악하는 것이 중요
일반적인 Language Model로는 이 관계를 학습하는 것이 어렵다.

Sentence A	문단+질문	가설	문단
Sentence B	답변	전제	요약

데이터

corpus의 이어진 두 문장 사용.
50%는 Sentence B가 Sentence A를 잇는 데이터 → Label: IsNext
50%는 특별한 순서가 없는 두 문장 데이터 → Label: NotNext

학습

Sentence A, B가 주어졌을 때 isNext인지
NotNext인지 classify하게 만든다

Pre-training BERT

The training loss is the sum of the mean masked LM likelihood and the mean next sentence prediction likelihood.

$$\text{Loss} = -\sum_{i=1}^{N}\sum_{j=1}^{V} y_{i,j}\log(p_{i,j})$$

N: Mask 수
V: Sequence 단어 수
p(i.j): 예측 확률
y(I,j): 정답

https://arxiv.org/pdf/1611.01603.pdf

Fine-tuning BERT

Encoder based 모델 장점 ⟷ **Bi-directional attention flow (Bi-DAF) 구조**

Passage, 질문 pair 같은 데이터 → 따로 encoding 후, cross attention으로

Text pair 데이터를 효율적으로 처리 가능하다.

두 sentence를 연결해서 input에 넣으면 된다.

Self attentio을 하면 알아서 cross attention 결과가 나온다

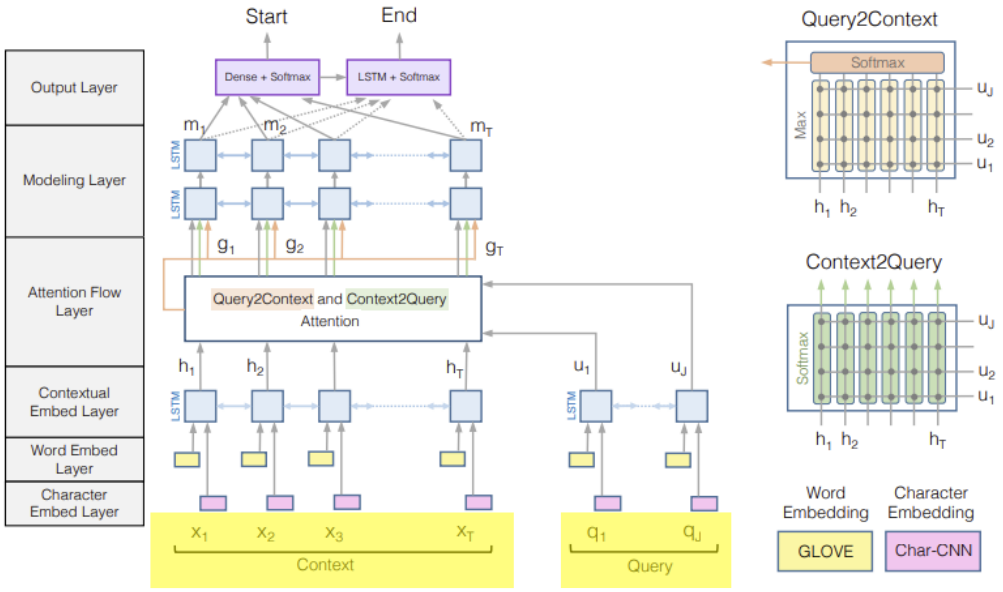
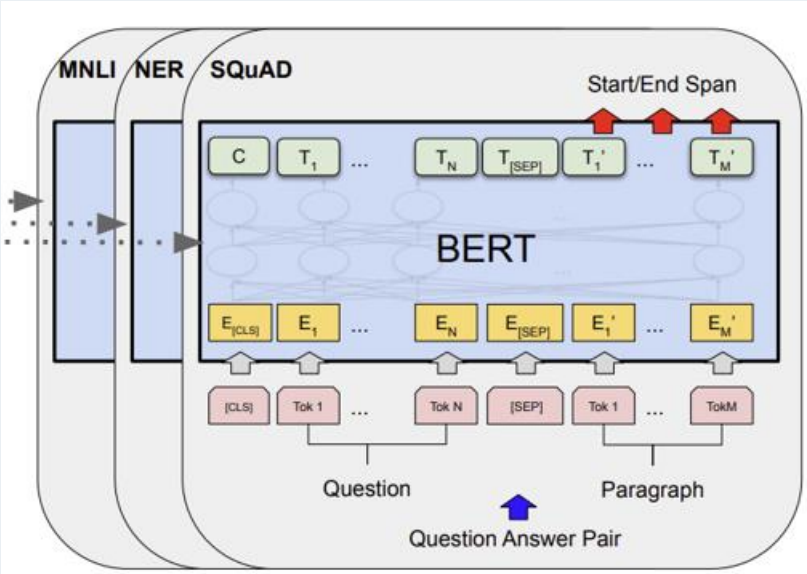


Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

Fine-Tuning 결과



Fine-Tuning

[CLS] →입력에 대한 전체적인 정보가 담긴다.

Downstream task가 Sentiment Analysis라면
Fine tuning을 통해 [CLS]가 입력 sequence를 통해
전체적인 감정을 판단하게 학습하는 것

use the final hidden vector C ∈ R(1xH) corresponding to the first input token ([CLS]) as the aggregate representation. The only new parameters introduced during fine-tuning are classification layer weights W ∈ R(K×H), where K is the number of labels. We compute a standard classification loss (cross entropy loss) with C and W, i.e., log(softmax(CW^T)).

Experiment Results

GLUE(General Language Understanding Evaluation benchmark)

MNLI, QQP, QNLI, SST-2, CoLA, SSTS-B, MRPC, RTE에 대해 fine-tuning

목적: fine-tuning based pre-trained language representation을 개선

모두 classification task

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
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

+4.5%

+7.0%

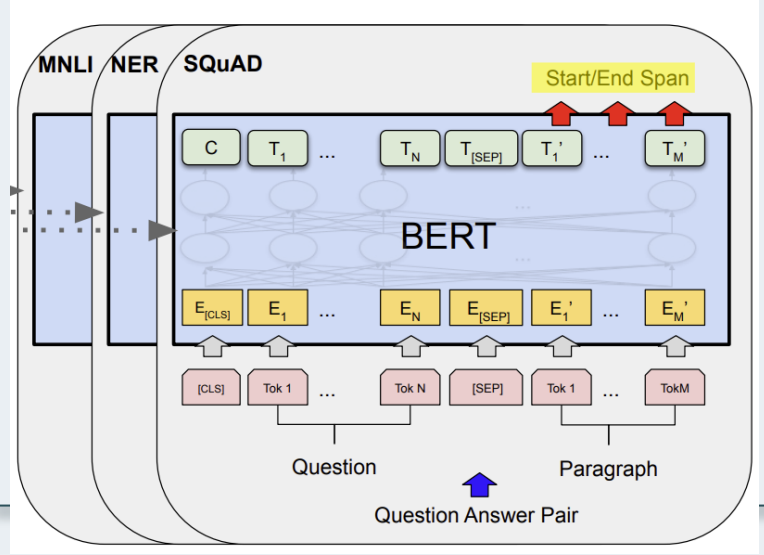
Big model performs better

Experiment Results

SQuAD (Stanford Question Answering Dataset) v1.1 Fine-tuning

Question + 답이 통째로 들어있는 Passage가 주어졌을 때,
Passage 내부에서 답을 잘 예측(찾기)할 수 있는지 평가

Input Sentence A = question, Sentence B = passage, 답안 start 벡터 S, 답안 end vector E



단어 i가 답안의 첫 부분일 확률, 답안의 마지막 부분일 확률

Argmax 결과: 모델의 답안 시작 단어, 답안 끝 단어

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

Score

$$S \cdot T_i + E \cdot T_j$$

The training objective is the sum of the log-likelihoods of the correct start and end positions

Experiment Results

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$BERT_{BASE}$ (Single)	80.8	88.5	-	-
$BERT_{LARGE}$ (Single)	84.1	90.9	-	-
$BERT_{LARGE}$ (Ensemble)	85.8	91.8	-	-
$BERT_{LARGE}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$BERT_{LARGE}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

+1.3 F1

+1.5 F1

Experiment Results

SQuAD v2.0 Fine-tuning

Question + 답이 통째로 들어있는 Passage, **Question + 답이 없는 Passage** 가 주어졌을 때,
답을 잘 예측(찾기)할 수 있는지, **질문과 답안 관계**를 잘 이해하는지 평가

[CLS] → Question과 관련이 없는 Passage가 주어진다면, Passage에 답이 없다는 것을 알아낼 것이다.

$$s_{\texttt{null}} = S \cdot C + E \cdot C$$

$$\hat{s_{i,j}} = \max_{j \geq i} S \cdot T_i + E \cdot T_j$$

$$\hat{s_{i,j}} > s_{\texttt{null}} + \tau$$

이 값이 더 클 때, Passage으로부터
답을 찾을 수 없다고 판단

이 값이 더 클 때, Passage으로부터
답을 찾을 수 있다고 판단

Experiment Results

SQuAD v2.0 Fine-tuning

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

+5.1

Experiment Results

SWAG (Situations With Adversarial Generations)

어떤 문장에 대해서 continuation 문장 후보가 여러 개 있을 때, 제일 적합한 문장을 찾는 것

Sentence A + (Sentence B1, Sentence B2, Sentence B3, Sentence B4) 중 하나

[CLS] → Sentence A와 continuation 문장 후보의 연관성 정보를 지닌다 → classification

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

+8.3%

+27.1%

Experiment Results: Effect of Pre-training tasks

Deep Bidirectionality of BERT Importance

이어지는 문장이 전 문장을 필연적으로 따르는가

알맞은 답변인지

두 문장이 의미적으로 동일한지

감정 분석

Tasks	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
	Dev Set				
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

문장 관계 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델
+ BiLSTM을 사용하여 어느정도 양방향 context 사용하게

Experiment Results: Effect of Pre-training tasks

Deep Bidirectionality of BERT Importance

이어지는 문장이 전 문장을 필연적으로 따르는가

알맞은 답변인지

두 문장이 의미적으로 동일한지

감정 분석

관계 파악 능력 ↓

Tasks	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
	Dev Set				
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

문장 관계 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델
+ BiLSTM을 사용하여 어느정도 양방향 context 사용하게

Experiment Results: Effect of Pre-training tasks

Deep Bidirectionality of BERT Importance

이어지는 문장이 전 문장을 필연적으로 따르는가

알맞은 답변인지

두 문장이 의미적으로 동일한지

감정 분석

관계 파악 능력 ↓
의미 파악 능력 ↓

Tasks	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
	Dev Set				
$BERT_{BASE}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

문장 관계 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델
+ BiLSTM을 사용하여 어느정도 양방향 context 사용하게

Experiment Results: 세부 실험

Deep Bidirectionality of BERT Importance

이어지는 문장이 전 문장을 필연적으로 따르는가

알맞은 답변인지

두 문장이 의미적으로 동일한지

감정 분석

SQuAD에 대해서만 ↑
나머지는 ↓

Tasks	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
	Dev Set				
$BERT_{BASE}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

문장 관계 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델

문장 관계, 양쪽 문맥 학습 안 한 모델
+ BiLSTM을 사용하여 어느정도 양방향 context 사용하게

Experiment Results: 세부 실험

Model 크기 실험
Transformer 논문을 보면 모델의 크기가 클수록 성능이 좋아졌다.

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

사이즈 커질수록 성능 좋아진다
+
Fine tuning 데이터가 적을수록 개선 효과가 커진다

Experiment Results: 세부 실험

Feature Based Approach with BERT

Feature Based Approach: 고정적인 Contextual Word Representation matrix을 Downstream Task 모델에 적용하여 성능 향상시키는 것

장점

1. 모든 NLP task가 Transformer based Encoder 구조로 표현될 수 있는 것이 아니다.
→ **BERT는 output이 다 [CLS] 기반**

2. 속도가 빠르다

Experiment Results: 세부 실험

Feature Based Approach with BERT
Named Entity Recognition task에 적용

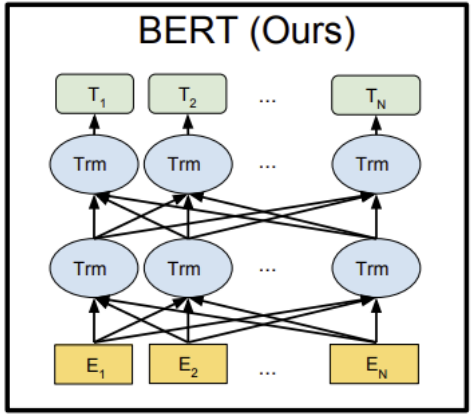
BERT의 Fine tuning과
Feature based approach 모두 적용

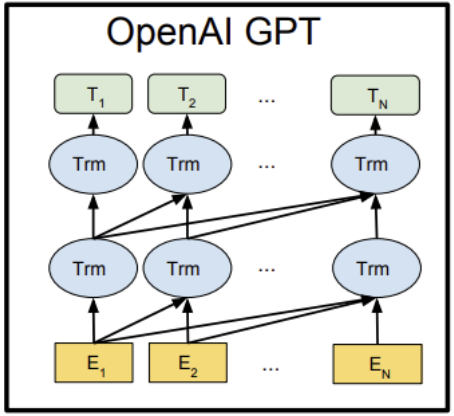
BERT의 Pre-training 모델 내부의
representation 결과를 조금씩 가져와서
NER task를 위한 Bi-LSTM 모델에 적용

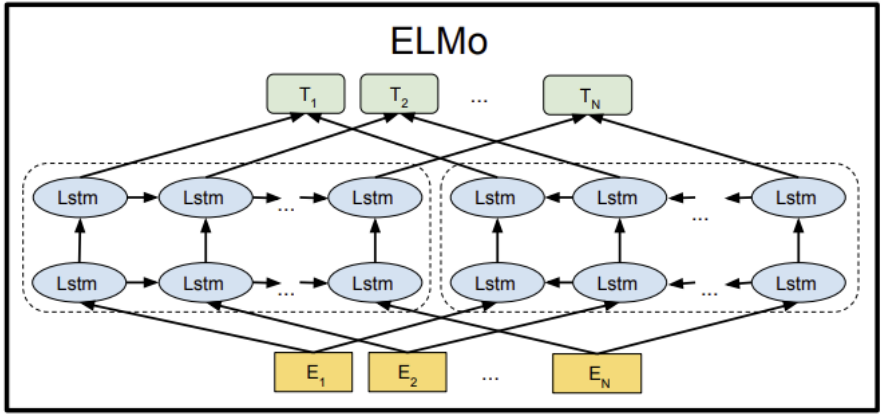
다른 NER 모델

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$BERT_{LARGE}$	96.6	92.8
$BERT_{BASE}$	96.4	92.4
Feature-based approach ($BERT_{BASE}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

추가: BERT vs GPT vs ELMO







only BERT representations are jointly conditioned on both left and right context in all layers.

GPT: Generative Pretrained Transformer → Generation Task를 위해 만들어진 것
Improving Language Understanding by Generative Pre-Training

추가: MASK

- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`.

The advantage of this procedure is that the Transformer encoder does not know which words it will be asked to predict or which have been replaced by random words, so it is forced to keep a distributional contextual representation of every input token.

Additionally, because random replacement only occurs for 1.5% of all tokens (i.e., 10% of 15%), this does not seem to harm the model's language understanding capability

추가: MASK

동일한 단어로

랜덤한 단어로

[MASK]으로

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI Fine-tune	NER Fine-tune	NER Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

추가: Cost

We train with batch size of 256 sequences (256 sequences * 512 tokens = 128,000 tokens/batch) for 1,000,000 steps, which is approximately 40 epochs over the 3.3 billion word corpus.
이런 훈련 과정이 꼭 필요할까?

→ 필요하다

Masked Language Pre-training 하는데 오래 걸리는데 꼭 사용해야 할까?

→ 느리긴 한데 성능이 압도적으로 좋다

500k번 더 학습하면
성능 1.0% 차이난다

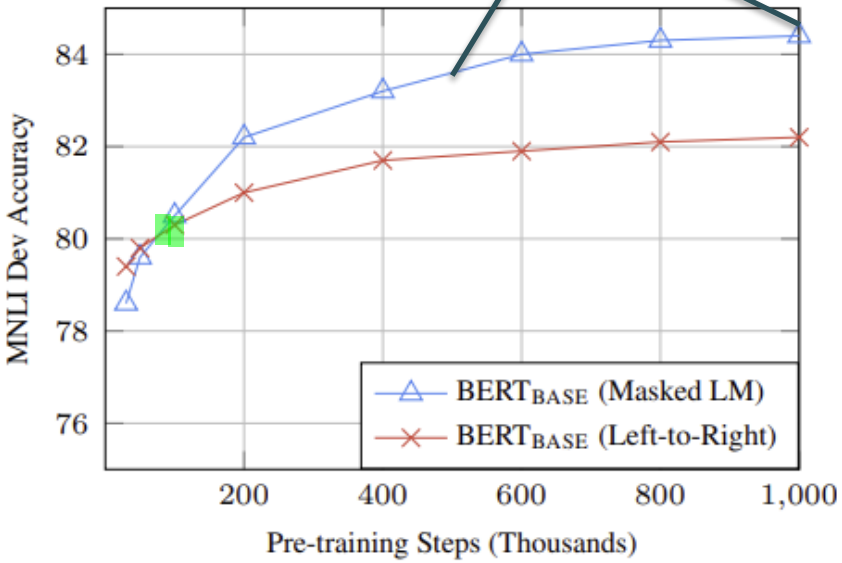


Figure 5: Ablation over number of training steps. This shows the MNLI accuracy after fine-tuning, starting from model parameters that have been pre-trained for k steps. The x-axis is the value of k.

https://arxiv.org/pdf/1907.11692.pdf

추가: RoBERTa

NSP(Next Sentence Prediction): Downstream task에 그렇게 효과적이지 않다

RoBERTa: NSP 제거