Bert & GPT

Few, Zero shot learning & Transfer learning

자연어 데이터의 불완전성

<Data Sample>

lend some dignity to a dumb story 0

the greatest musicians 1

cold movie 0

with his usual intelligence and subtlety 1

redundant concept 0

<summary>

the belgian duo took to the dance floor on monday night with some friends. ma nchester united face newcastle in the premier league on wednesday . red devils will be looking for just their second league away win in seven . louis van gaal’s si de currently sit two points clear of liverpool in fourth .

- 특정 자연어 Task 를 해결하기 위해서는 다양한 Label이 요구됨. 특히, format도 매우 복잡함

https://bluediary8.tistory.com/118

자연어 데이터의 불완전성

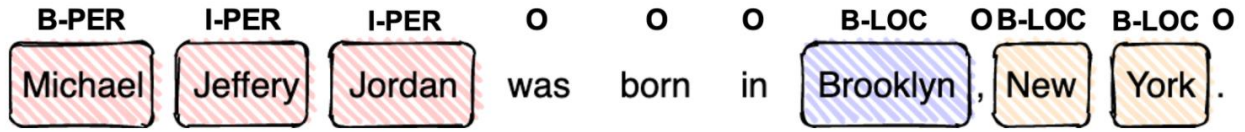
Dataset : SQuAD Sample

Context: Hyperbaric (high-
pressure) medicine uses special oxygen chambers to increase the partial pr
essure of O 2 around the patient and, when needed, the medical staff. Carb
on monoxide poisoning, gas gangrene, and decompression sickness (the 'b
ends')are sometimes treated using these devices. Increased O 2 concentrati
on in the lungs helps to displace carbon monoxide from the heme group of
hemoglobin. Oxygen gas is poisonous to the anaerobic bacteria that cause
gas gangrene, so increasing its partial pressure helps kill them. Decompress
ion sickness occurs in divers who decompress too quickly after a dive, resul
ting in bubbles of inert gas, mostly nitrogen and helium, forming in their bl
ood. Increasing the pressure of O 2 as soon as possible is part of the treatm
ent.

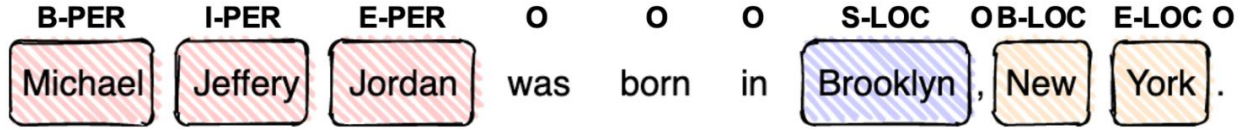
Question: What does increased oxygen concentrations in the patient's lungs
displace?

target: carbon monoxide

BIO 시스템에 따라 태깅이 된 문장의 예시



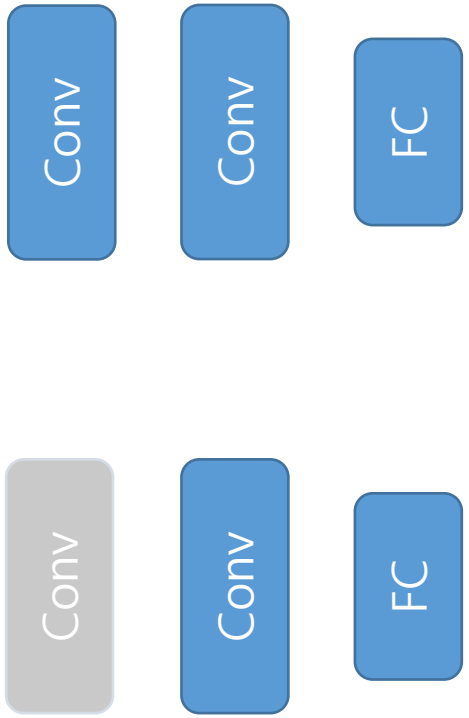
BIESO시스템에 따라 태깅이 된 문장의 예시



1. 수많은 텍스트 데이터들이 레이블이 없이 존재함 (의료 데이터셋과의 비교)
2. (가정) 주어진 문장에 대한 함축적인 문맥을 이해할 수 있다면 다양한 Task에 적용 가능함

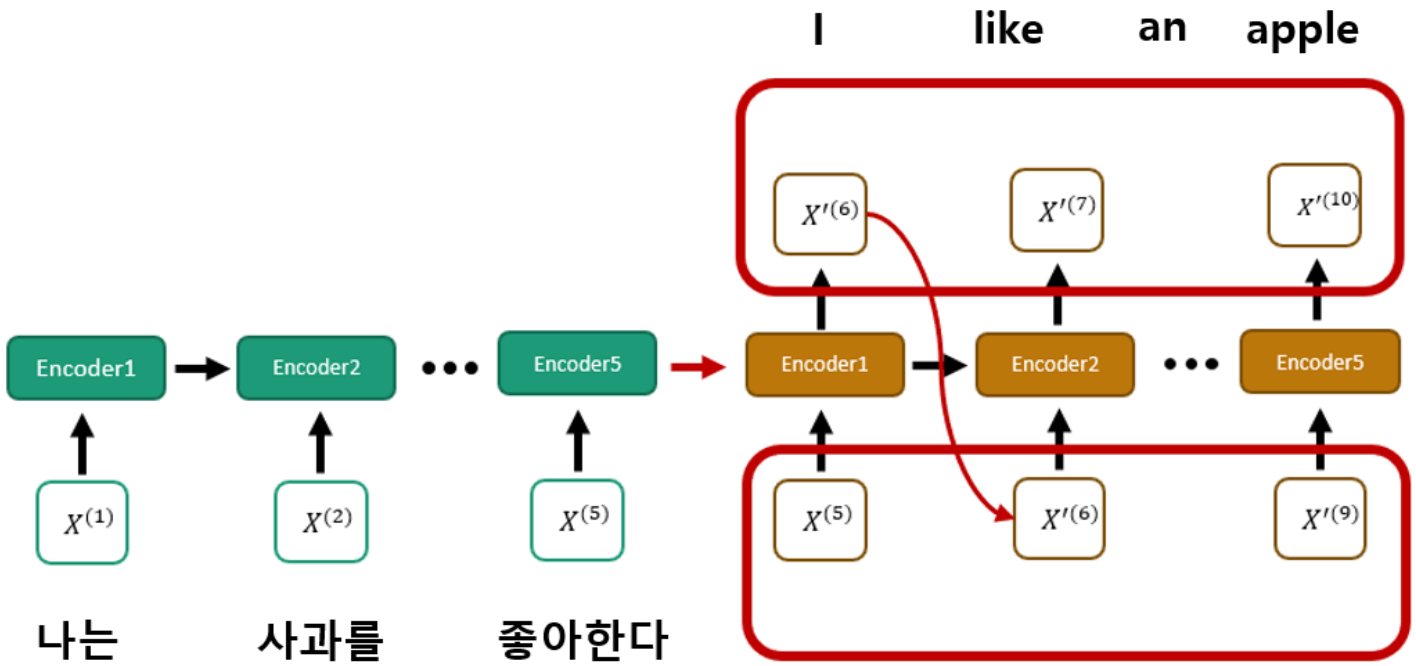
https://www.letr.ai/explore/tech-20210723

Fine tunning (cv)




```
for param in model.parameters():
    param.requires_grad = False
```

Transfer learning



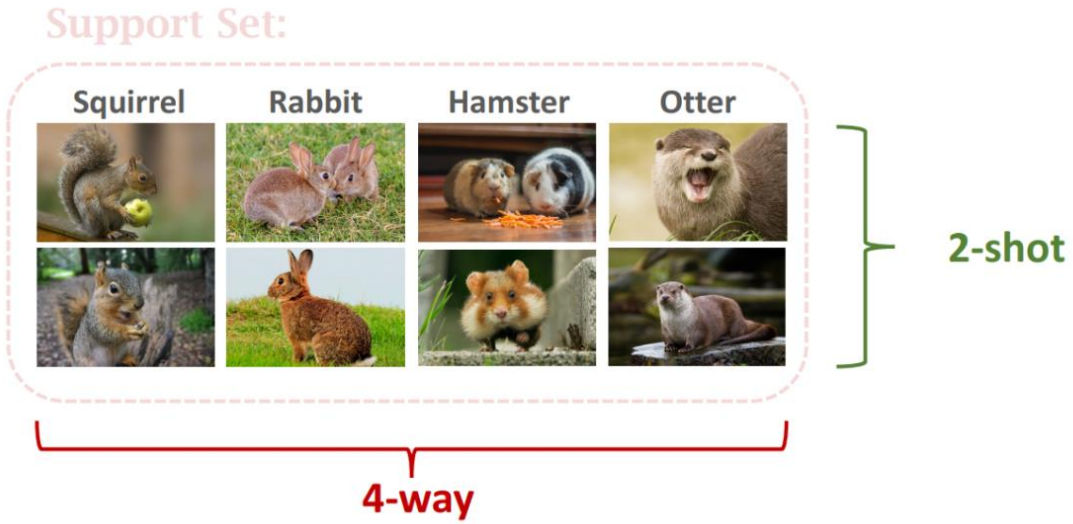
- 일본어 – 영어, 한국 – 일본어 간의 번역 데이터는 많으나 한국 – 영어 데이터는 적을 경우
- 영어 –> 일본어 데이터셋에서 문맥 벡터를 잘 뽑아 내는 모델을 학습 한 후,
  영어 -> 한국어 데이터셋에 적용한다면?

Few shot, zero shot

Query image : 추론할 입력 데이터

Training set : 모델이 학습하는 데이터 셋

Support set : 추론해야 하는 셋

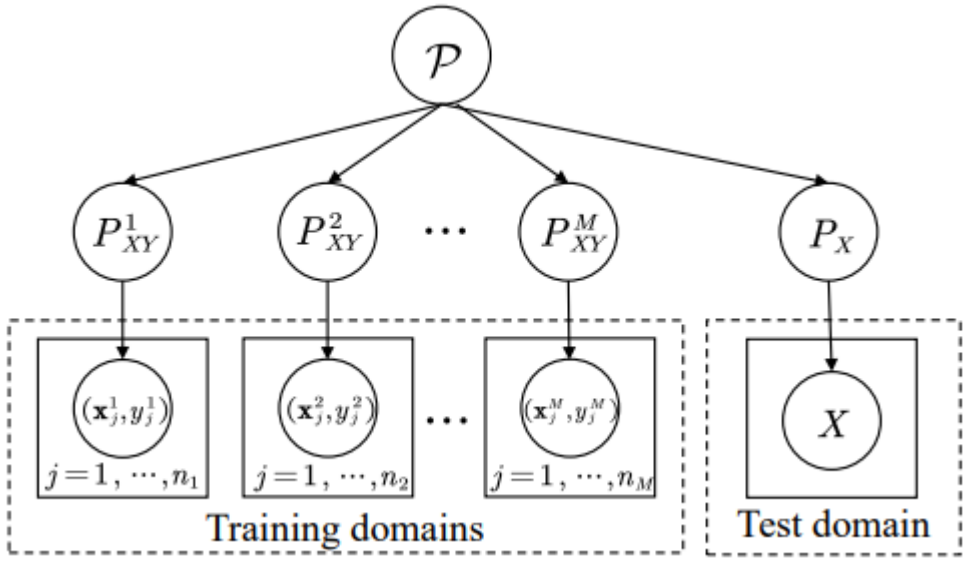


Notation

Generalizing to Unseen Domains: A Survey on Domain Generalization 2021

Definition 1 (Domain). *Let $\mathcal{X}$ denote a nonempty input space and $\mathcal{Y}$ an output space. A domain is composed of data that are sampled from a distribution. We denote it as $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \sim P_{XY}$, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, $y \in \mathcal{Y} \subset \mathbb{R}$ denotes the label, and P_{XY} denotes the joint distribution of the input sample and output label. X and Y denote the corresponding random variables.*

Definition 2 (Domain generalization). *As shown in Fig. 2, in domain generalization, we are given M training (source) domains $\mathcal{S}_{train} = \{\mathcal{S}^i \mid i = 1, \cdots, M\}$ where $\mathcal{S}^i = \{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{n_i}$ denotes the i-th domain. The joint distributions between each pair of domains are different: $P_{XY}^i \neq P_{XY}^j, 1 \leq i \neq j \leq M$. The goal of domain generalization is to learn a robust and generalizable predictive function $h : \mathcal{X} \to \mathcal{Y}$ from the M training domains to achieve a minimum prediction error on an* unseen



Notation

TABLE 2
Comparison between domain generalization and some related learning paradigms.

Learning paradigm	Training data	Test data	Condition
Multi-task learning	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{Y}^i \neq \mathcal{Y}^j, 1 \leq i \neq j \leq n$
Transfer learning	$\mathcal{S}^{src}, \mathcal{S}^{tar}$	$\mathcal{S}^{tar}$	$\mathcal{Y}^{src} \neq \mathcal{Y}^{tar}$
Domain adaptation	$\mathcal{S}^{src}, \mathcal{S}^{tar}$	$\mathcal{S}^{tar}$	$P(\mathcal{X}^{src}) \neq P(\mathcal{X}^{tar})$
Meta-learning	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{S}^{n+1}$	$\mathcal{Y}^i \neq \mathcal{Y}^j, 1 \leq i \neq j \leq n+1$
Lifelong learning	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{S}^i$ arrives sequentially
Zero-shot learning	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{S}^{n+1}$	$\mathcal{Y}^{n+1} \neq \mathcal{Y}^i, 1 \leq i \leq n$
Domain generalization	$\mathcal{S}^1, \cdots, \mathcal{S}^n$	$\mathcal{S}^{n+1}$	$P(\mathcal{S}^i) \neq P(\mathcal{S}^j), 1 \leq i \neq j \leq n+1$

Notation

GPT 1

GPT-1

Gpt-1 bert gpt-2 gpt3 순으로 연구

GPT = “ Generative Pre-Training “

GPT-1

- 비지도학습 기반의 pre-training 과 지도학습 기반의 fine-tunning 을 결합한 semi-supervised learning

- 그래서 다양한 자연어 task에서 fine-tunning 만으로도 좋은 성능을 보이는
  범용적인 자연어 representation 을 학습하는 것

- 2 stage 로 구성되어 있으며 transformer의 decoder 구조를 사용함

- 기존 RNN 대비 좋은 성능을 보였으며 일반화 성능 확인

GPT-1

Stage 1 : Unsupervised pre-training

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

Architecture

$$h_0 = UW_e + W_p$$
$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$
$$P(u) = \texttt{softmax}(h_n W_e^T)$$

GPT-1

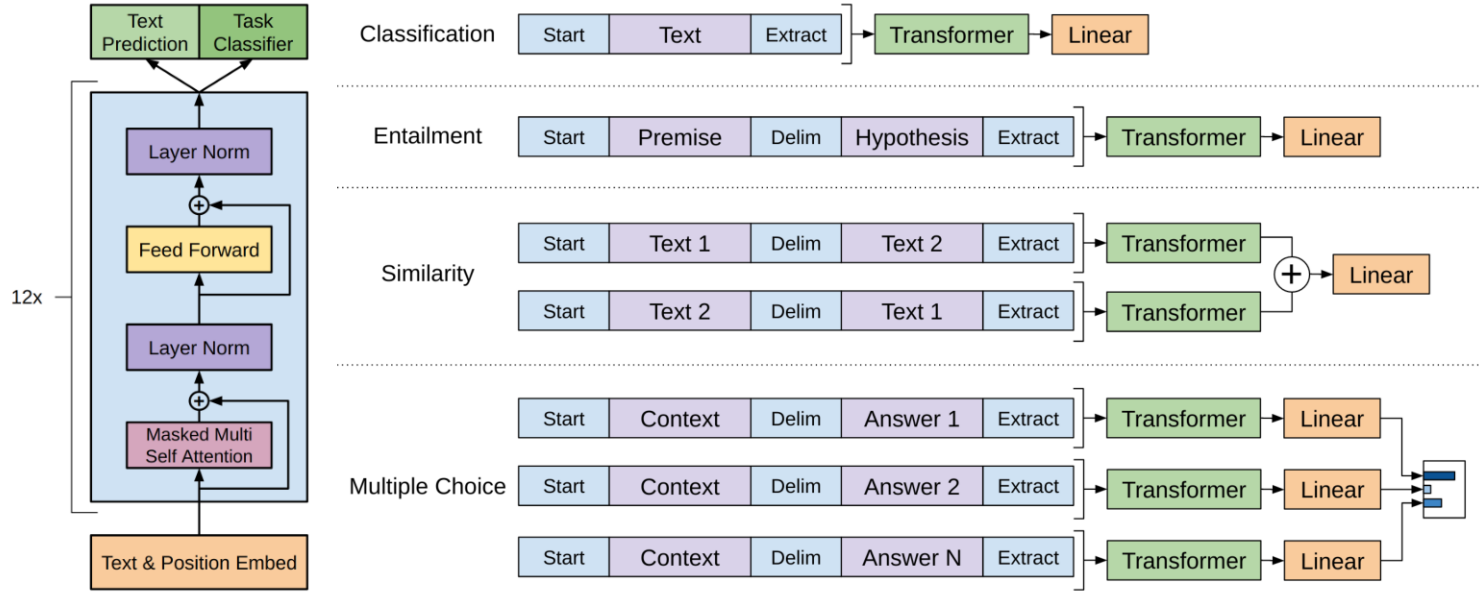
Stage 2 : Supervised fine-tunning

$$P(y|x^1,\ldots,x^m) = \texttt{softmax}(h_l^m W_y).$$

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1,\ldots,x^m).$$

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

GPT-1 : Task-specific input transformations



Task-specific input transformers

- 기존: Task specific 구조에 기반한 학습, 구조에 종속되기 때문에 task 가 변할 때 마다 많은 커스터마이즈 요구

GPT-1

- Pre-training model이 적용될 수 있도록 input 구졸르 convert

GPT-1

- 다양한 Task에서 SOTA 달성

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

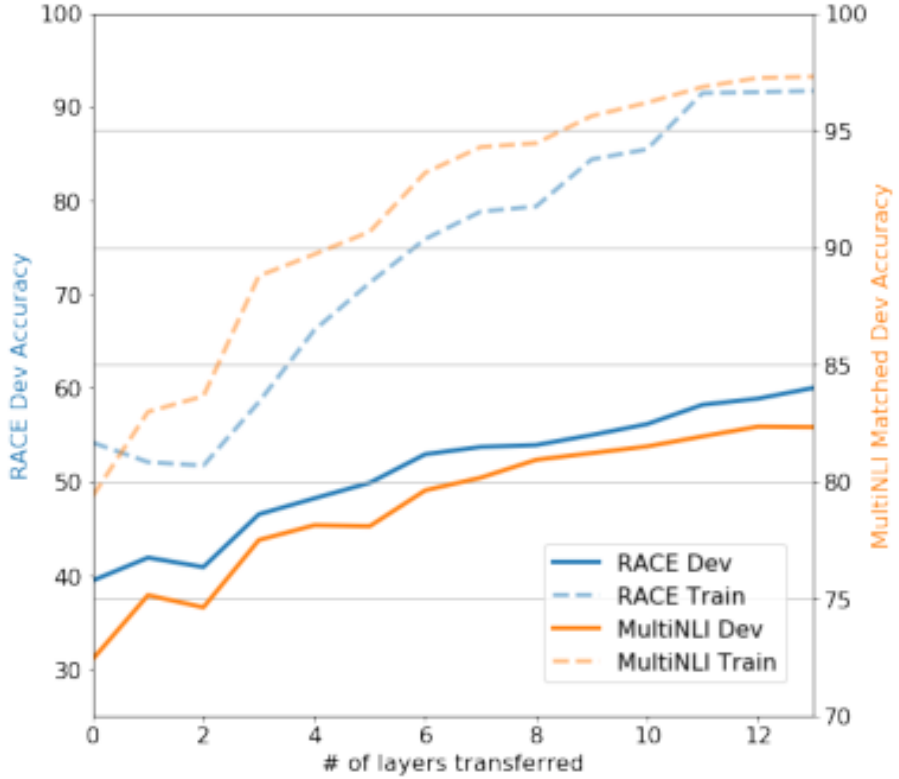
GPT-1

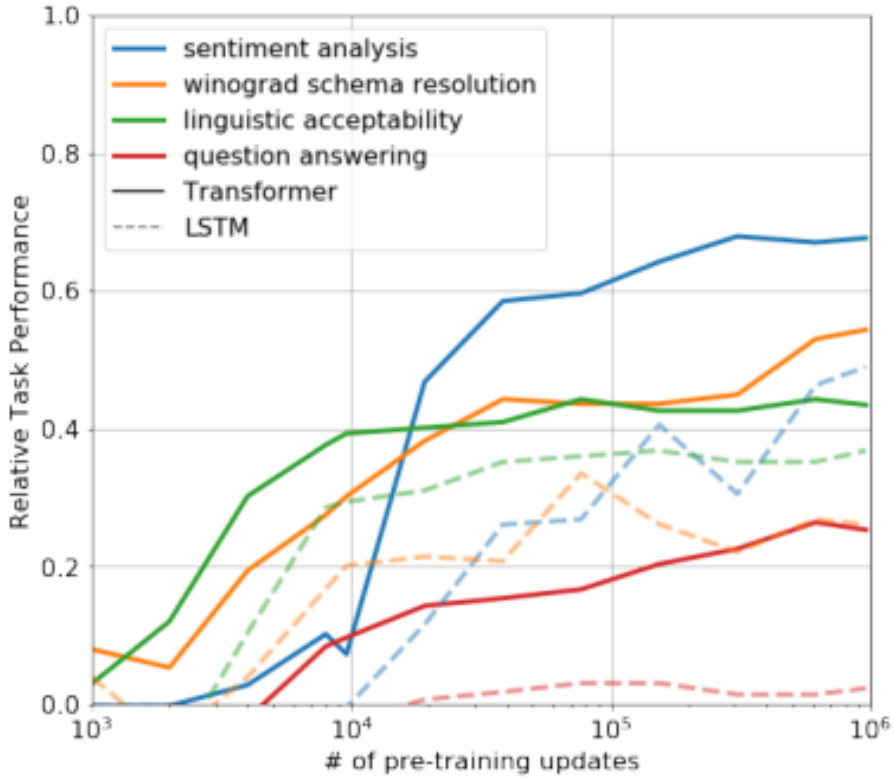
Parameter	Descrption
State dimension	decoder: 768, inner state: 3072
Batch size	64 random sample × 512 token/sample
Schedule	100 epochs,
Optimizer	Adam
Learning Rate	0~2000 step까지 2.5e-4까지 증가, 이후 cosine 함수를 따라 0으로 서서히 감소
warmup_steps	4000
Regularization	L2($w = 0.01$)
Activation	GELU(Gaussian Error Linear Unit)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

- LM -> 큰 데이터셋 에서는 좋은 결과 but 작은 데이터셋에서는 아님

GPT-1





- Layer 가 증가함에 따라 정확도가 높아지는 것을 확인
- LSTM과 비교하여 다양한 task에서 일반화 성능 확인

BERT

BERT

- 다양한 Task에서 SOTA 달성

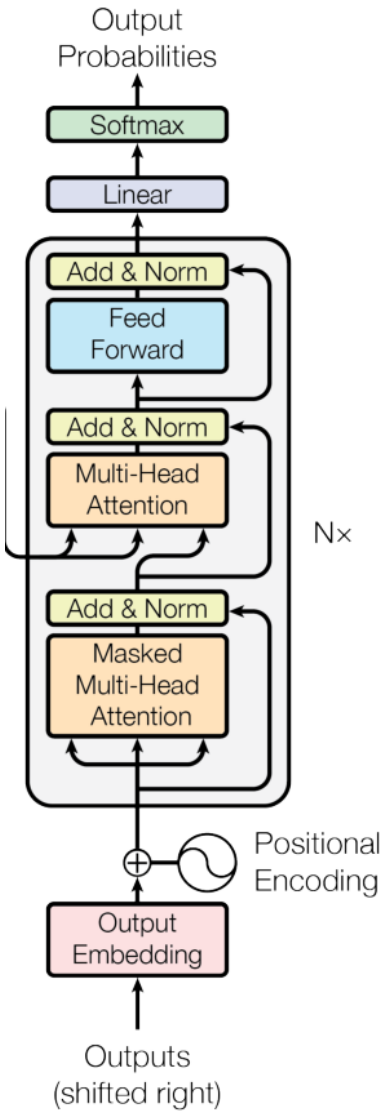
BERT

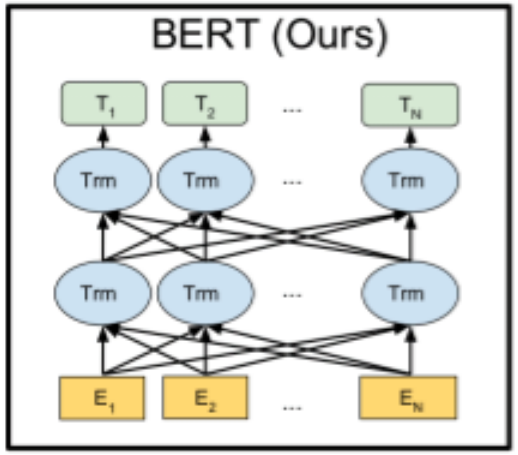
- BERT = “Bidirectional Encoder Representations from Transformer”
- Wiki & book data 와 같은 대용량 unlabeled data로 pre-training 시킨 후,
  특정 task 에 transfer learning 을 함
- GPT와의 차이? -> unidirectional vs bidirectional
- GPT와는 달리 새로운 네트워크를 붙이지 않고 fine-tunning 만을 진행함

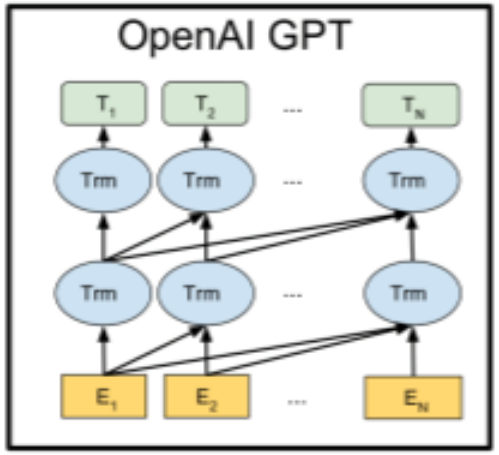
BERT

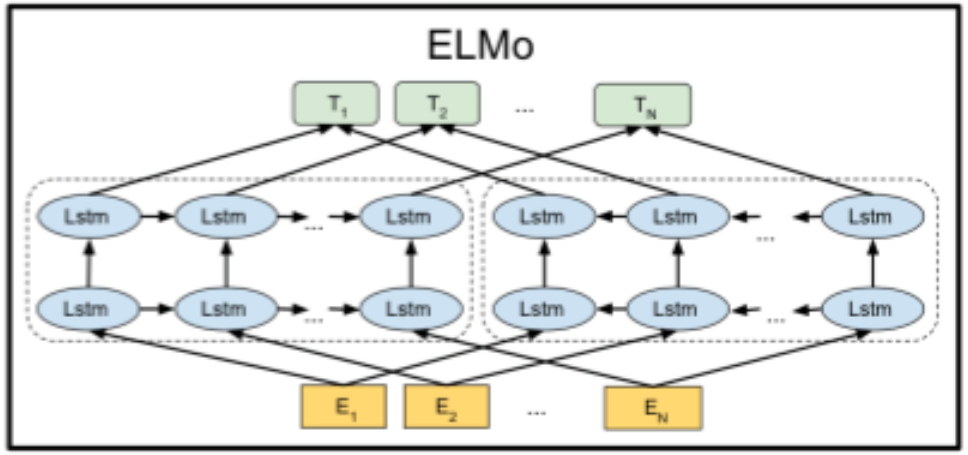
- GPT-1 : Unsupervised pre-training -> BERT : Masked Language Model(MLM) & Next sentence prediction
- Next sentence prediction
  - 문장간 관계를 알아내기 위한 task, 두 문장이 실제 corpus 에서 이어져 있는지 아닌지 확인
  - 50% 는 실제 이어져 있는 문장
- Pre-training 프로세스는 GPT-1과 같음

BERT


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Output Embedding
Outputs (shifted right)

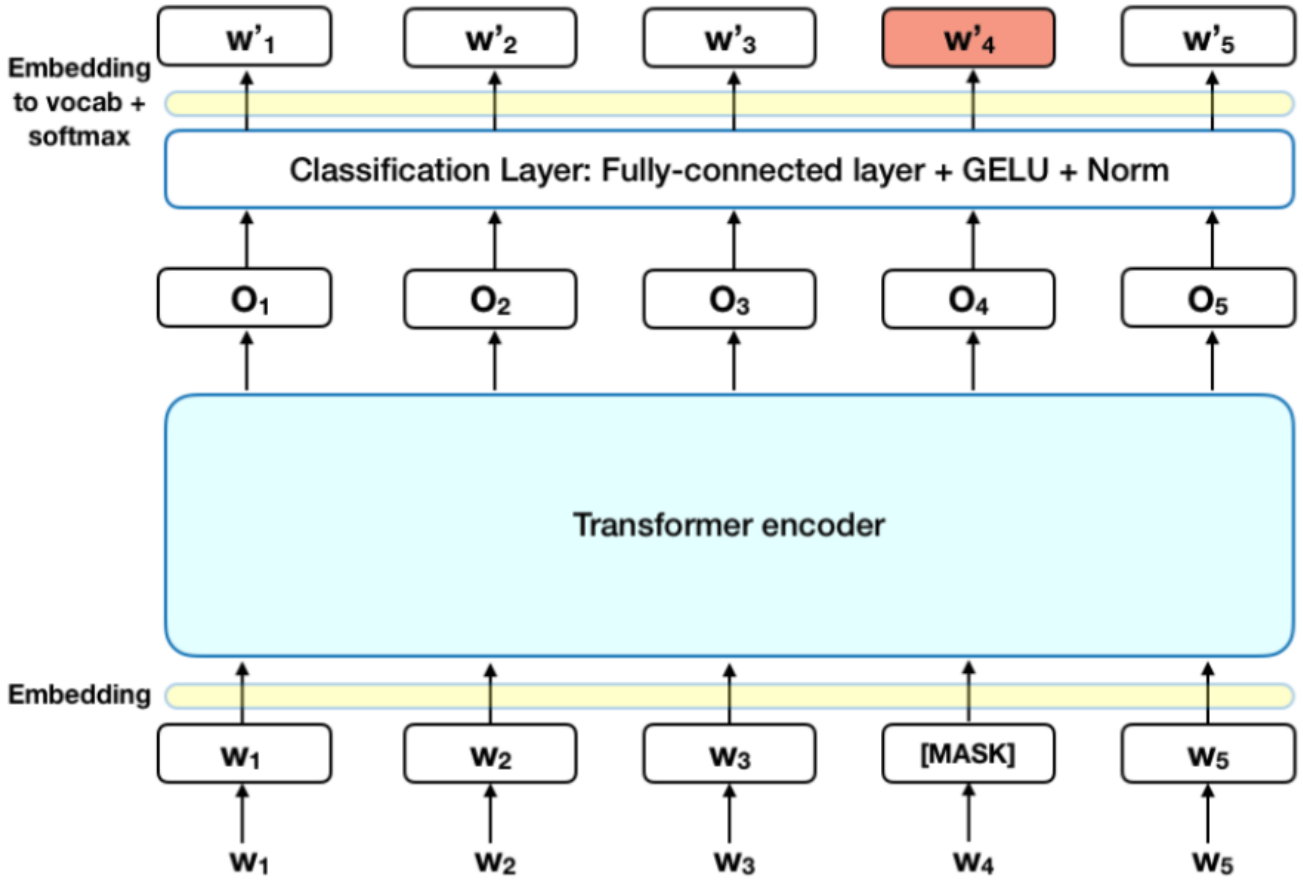

BERT (Ours)
Trm
E1
E2
EN


OpenAI GPT
Trm
E1
E2
EN


ELMo
Lstm
E1
E2
EN

BERT

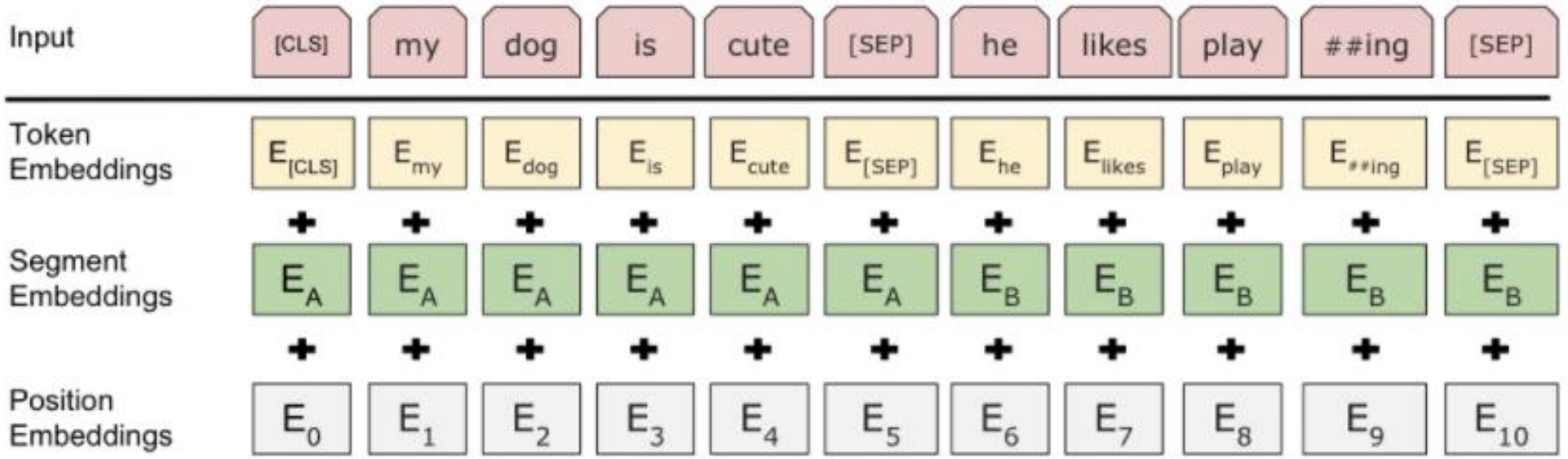
MLM



- [MASK] 비율 : 15%
- Tokenization : Wordpiece
- LM 의 left-to-right 와는 달리, [MASK] 를 추론하는 task 수행
- Fine tunning 에는 사용되지 않음

- [MASK] 생성 과정
  - 80% : token 을 [MASK]로 변환
  - 10% : token 을 임의의 단어로 변경
  - 10% : 원래의 단어 token 으로 둠

- Pre-trained 되는 transformer encode의 입장에서는 contextual representation 학습

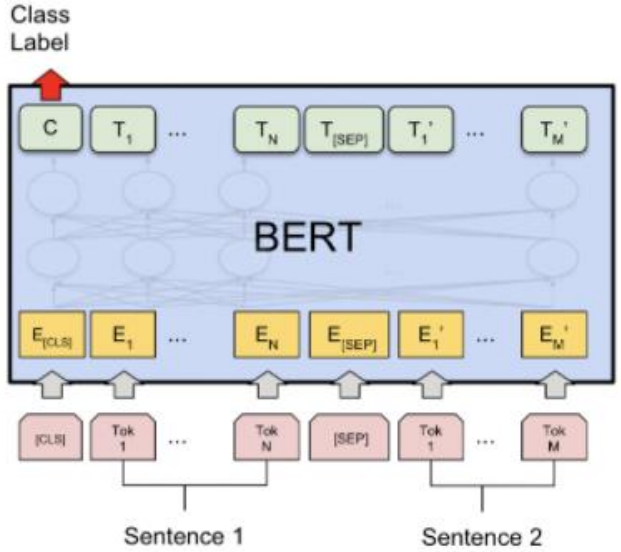


QA의 경우 sentence 가 여러 개일 수 있음

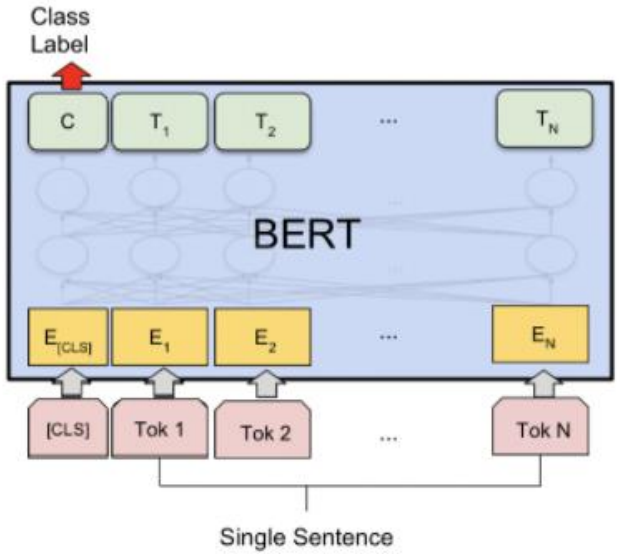
BERT

Fine-tunning

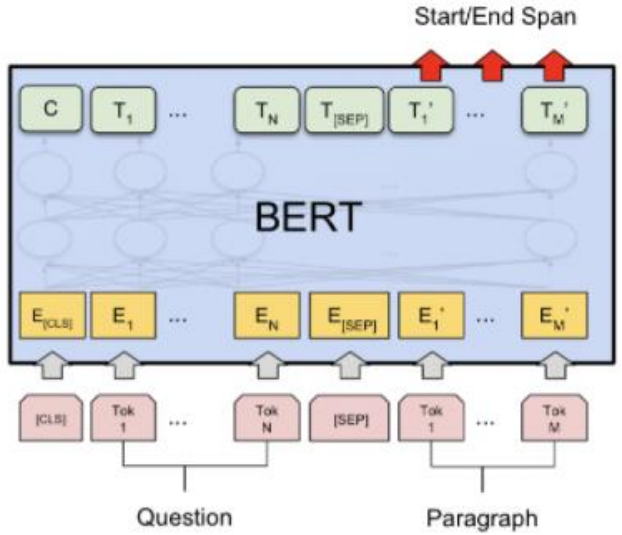
- Sequence-level classification
  - [CLS] token 의 output 사용
  - CLS output 에 W matrix 를 곱해주고 softmax를 취해 준다. $P = softmax(CW^T)$
- Span-level, token-level prediction



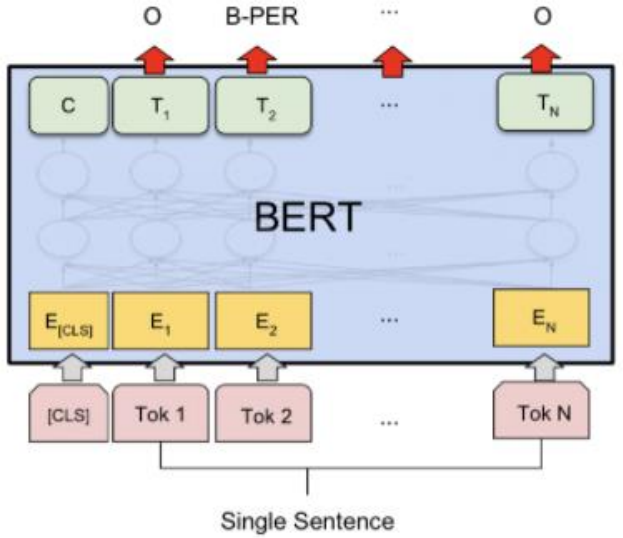
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



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



(c) Question Answering Tasks:
SQuAD v1.1



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

BERT

Datasets

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.

GLUE

- 다양한 task를 모아놓아 종합적인 자연어 이해 능력 테스트가 가능한 벤치마크
- 대부분의 task에 SOTA
- 특히, 데이터 크기가 작아도 fine-tunning 후에는 좋은 성능

BERT

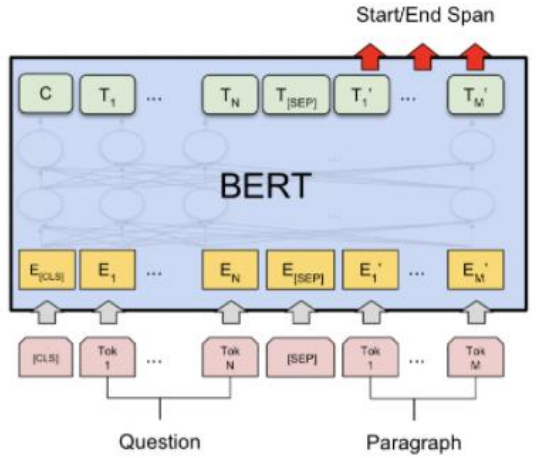
Datasets

SQuAD

- GLUE는 sequence classification
- SQuAD 는 질문 과 지문이 주어지고, substring 인 정답 찾기
- 질문 A, 지문 B 지문에서 substring 찾기 문제
- Start vector와 end vector의 dot product를 하여 찾기

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				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.



BERT

Datasets

SWAG

- Grounded common-sense inference
- 문장이 주어지고, 가장 잘 이어지는 문장 찾기
- 주어진 문장 A, 가능한 문장들 B

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

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

BERT

Datasets

CoNLL-2003

- 각각의 단어가 어떤 형식인지
  - Person, Organization, Location …
- 토큰마다 classifier 붙이기

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		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
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	-

Table 7: CoNLL-2003 Named Entity Recognition results. Hyperparameters were selected using the Dev set. The reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

BERT

Ablation studies

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	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

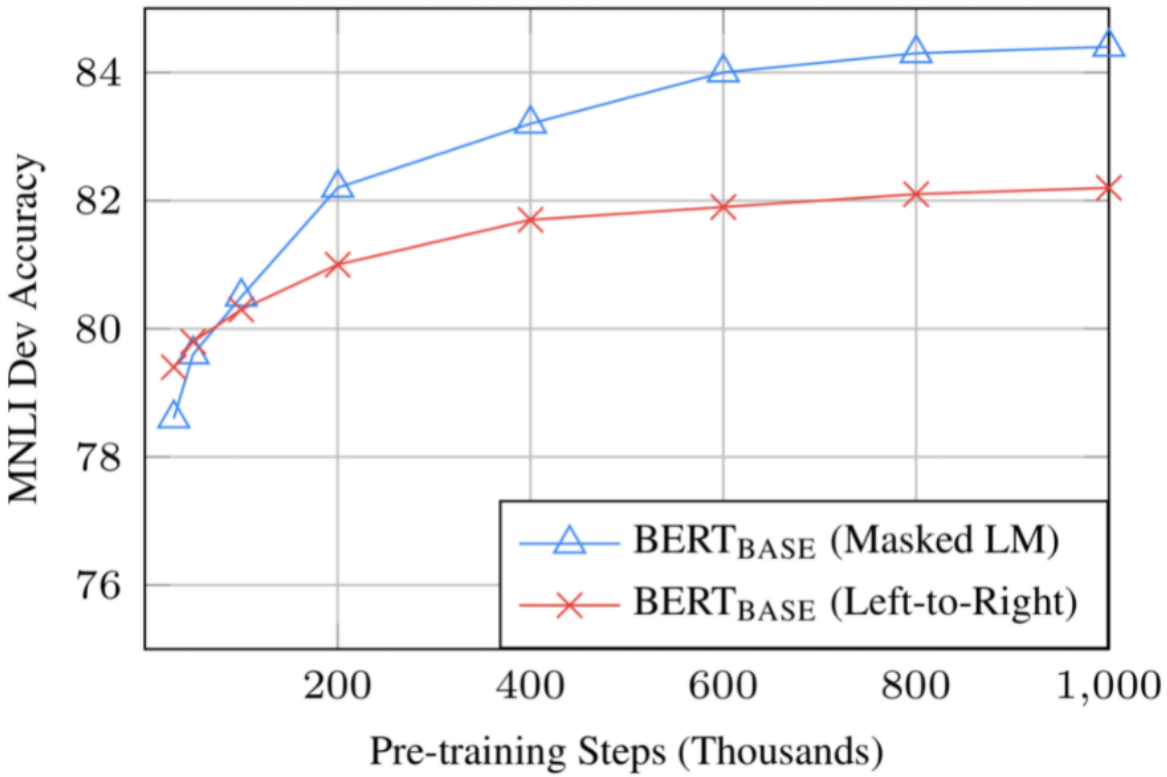
- Pre-train 을 하나라도 제거하면 성능 감소가 일어남
- No NSP -> 자연어 추론 계열(NLI)에서 성능 감소 폭 큼
- MLM 대신 LTR -> 성능이 매우 감소함

BERT

Ablation studies

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

Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. "LM (ppl)" is the masked LM perplexity of held-out training data.

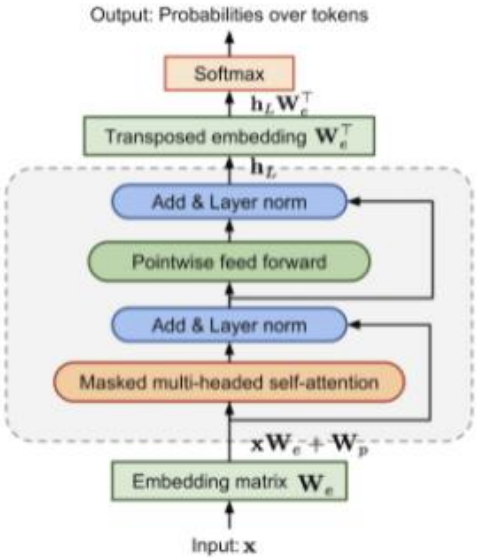


- 모델 사이즈가 커질수록 성능 향상
- MLM이 많은 training이 필요하지만 성능향상 확인

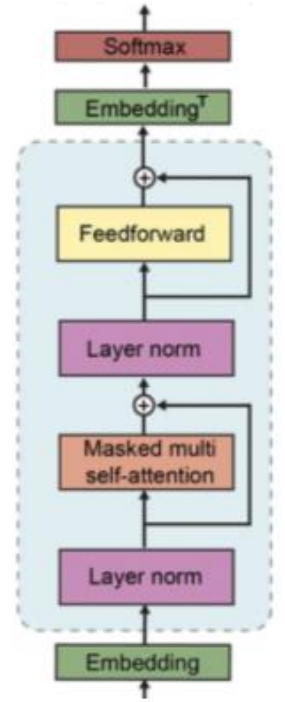
GPT-2

GPT-2

- Fine-tunning 없이도 우리는 가능하게 하고 싶다.
- 모델 자체는 GPT-1과 크게 차이 없음
- Zero shot learning
  - Model이 바로 downstream task에 적용함 (few shot: 몇 번 보고 적용함)
- WebText 데이터을 구축
  - 이 대용량 데이터셋에 LM 모델을 학습했을 때
    supervision 없이도 다양한 task 처리
- Byte pair Encoding을 활용 하여 Out of Vocabulary 문제 해결



GPT-1 decoder block



GPT-2 decoder block

GPT-2

Zero shot?

- 문장의 긍/부정 -> what do you think about this sentence ? 같은 질문 추가
- 문장 요약 -> What is the summary of … ? 추가
- 번역 -> what is translated sentence in Korean? 추가

Byte pair Encoding

Word Piece model (BERT)

- **Word** : Jet makers feud over seat width with big orders at stake
- **Wordpieces**: _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

Jet은 자주 등장하지 않아서 J et 로 나눔

모든 단어 시작에는 _

https://lovit.github.io/nlp/2018/04/02/wpm/

Byte pair Encoding

Byte-pair encoding(BPE)

```
vocab = {'l o w </w>' : 5,
         'l o w e r </w>' : 2,
         'n e w e s t </w>':6,
         'w i d e s t </w>':3
         }
```

1. 모두 캐릭터 단위로 분리
2. 각각의 캐릭터에서 빈도수 측정

```
# vocabulary
l, o, w, e, r, n, w, s, t, i, d
```

https://wikidocs.net/22592

Byte pair Encoding

1. 빈도수가 가장 높은 유니그램의 쌍을 통합함

```
# vocabulary
l, o, w, e, r, n, w, s, t, i, d
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est, lo
```

```
vocab = {'l o w </w>' : 5,
         'l o w e r </w>' : 2,
         'n e w e s t </w>':6,
         'w i d e s t </w>':3
         }
```

```
# dictionary update!
l o w : 5,
l o w e r : 2,
n e w est : 6,
w i d est : 3
```

https://wikidocs.net/22592

Byte pair Encoding

1. 빈도수가 가장 높은 유니그램의 쌍을 통합함

```
# vocabulary
l, o, w, e, r, n, w, s, t, i, d
```

```
vocab = {'l o w </w>' : 5,
         'l o w e r </w>' : 2,
         'n e w e s t </w>':6,
         'w i d e s t </w>':3
         }
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est, lo, low, ne, new, newest, wi, wid, widest
```

```
n e w est : 6,
w i d est : 3
```

```
# vocabulary update!
l, o, w, e, r, n, w, s, t, i, d, es, est, lo
```

https://wikidocs.net/22592

GPT-2

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

Zero shot 임에도 불구하고 8개중 7개에서 SOTA

특히, PTB,Wikitext-2 와 같은 적은 데이터셋에서 좋은 성능

GPT-3

GPT-3

- GPT-2 대비 Self-attention layer를 굉장히 많이 쌓아 parameter 수를 대폭 늘림
- GPT-2에서 사용하는 Zero shot learning framework의 확장

GPT-3

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

GPT-3

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

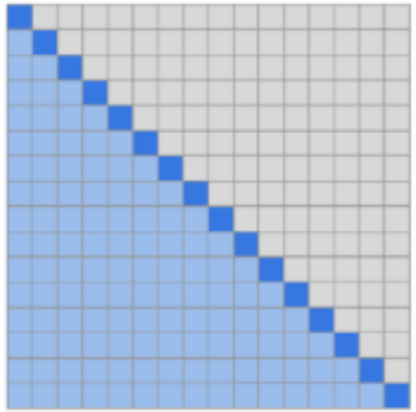
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

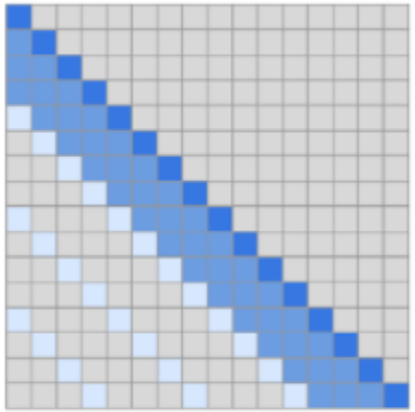
```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

GPT-3

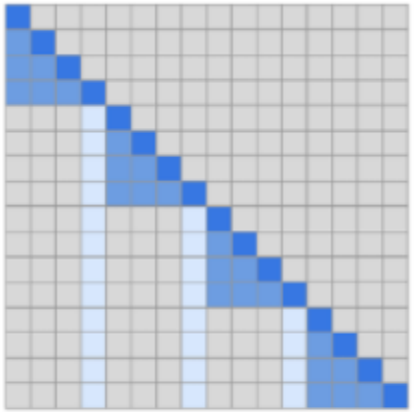
- Layer 가 증가함에 따라 정확도가 높아지는 것을 확인
- LSTM과 비



(a) Transformer



(b) Sparse Transformer (strided)



(c) Sparse Transformer (fixed)

	BERT	GPT-1	GPT-2	GPT-2
Pre-training	o	o	o	o
Fine-tunning	o	o	x	
Structure	Transformer Encoder	Transformer Decoder		
Attention	Multi-head attention	Masked Multi-head attention		
Tranining	MLM	Next word guess		

Thank you