DSBA CS224n 2021 Study

[Lecture – 10] Transformers and Pretraining

고려대학교 산업경영공학과

Data Science & Business Analytics Lab

발표자 : 김재희

00 발표 목차

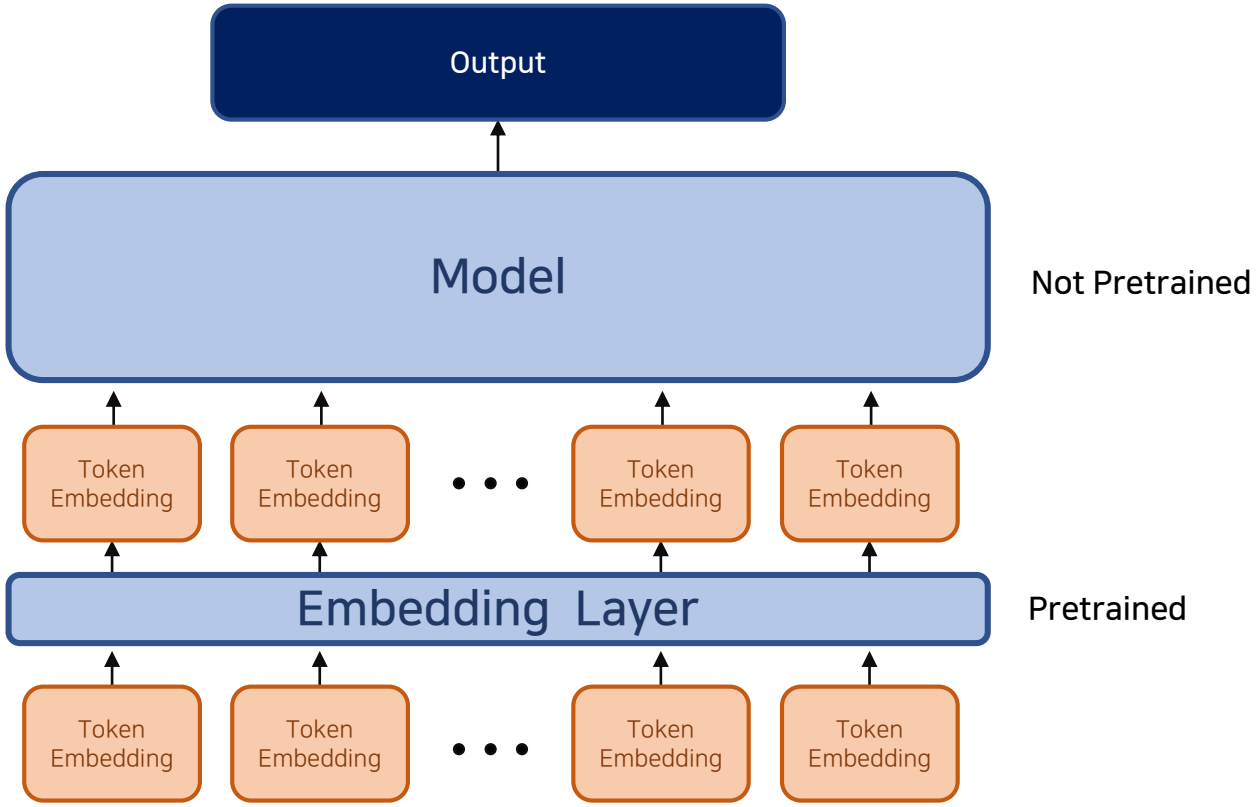
Contents

01 Problems of Vocab

- **현재까지 다룬 방법론에서의 전체 흐름**

- 임베딩 레이어 : Pretrained Embedding 사용
- 모델 파라미터 : 초기화

- 학습 데이터 부족
- Out-Of-Vocabulary 문제 발생
- 다른 문장에도 동일한 임베딩 사용



01 Problems of Vocab

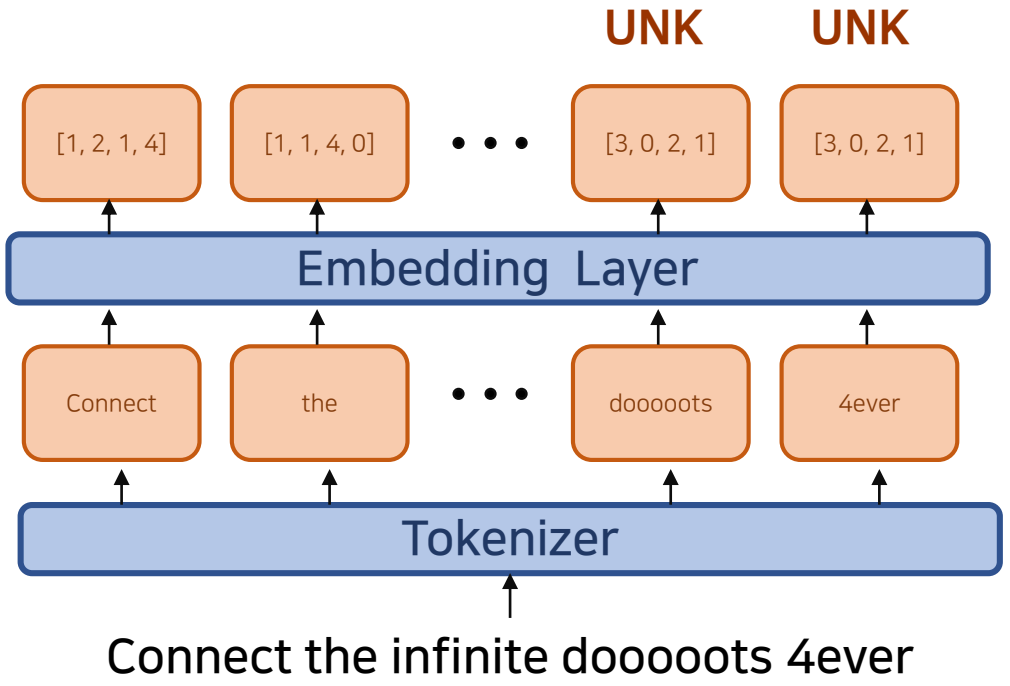
Out of Vocabulary Problem

✓ **임베딩 과정**

- 문자열 입력
- 학습된 토크나이저를 통해 토큰화
- 각 토큰별 학습된 임베딩 벡터 변환

✓ **학습 데이터에 존재하지 않는 단어**

- 모두 UNK(Unknown) 토큰으로 토큰화
- 다른 토큰이 모두 동일 벡터로 변환



01 Problems of Vocab

Out of Vocabulary Problem

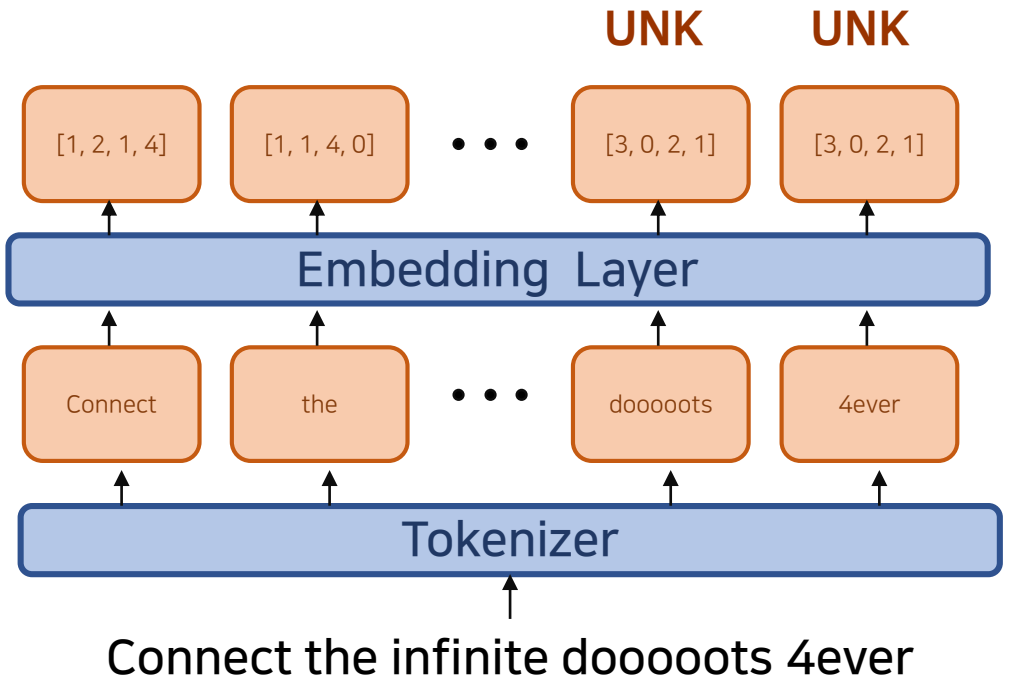
✓ 임베딩 과정

- 문자열 입력
- 학습된 토크나이저를 통해 토큰화
- 각 토큰별 학습된 임베딩 벡터 변환

✓ 학습 데이터에 존재하지 않는 단어

- 모두 UNK(Unknown) 토큰으로 토큰화
- 다른 토큰이 모두 동일 벡터로 변환

➡ 변형, 오탈자, 신조어에 대응 X



01 Problems of Vocab

Out of Vocabulary Problem

✓ Subword Token

- 단어 이하의 단위에서 언어 구조를 추론
- Byte-Pair-Encoding(BPE)
- 문법/의미/활용 포착

Bidirectional

광광우럭따

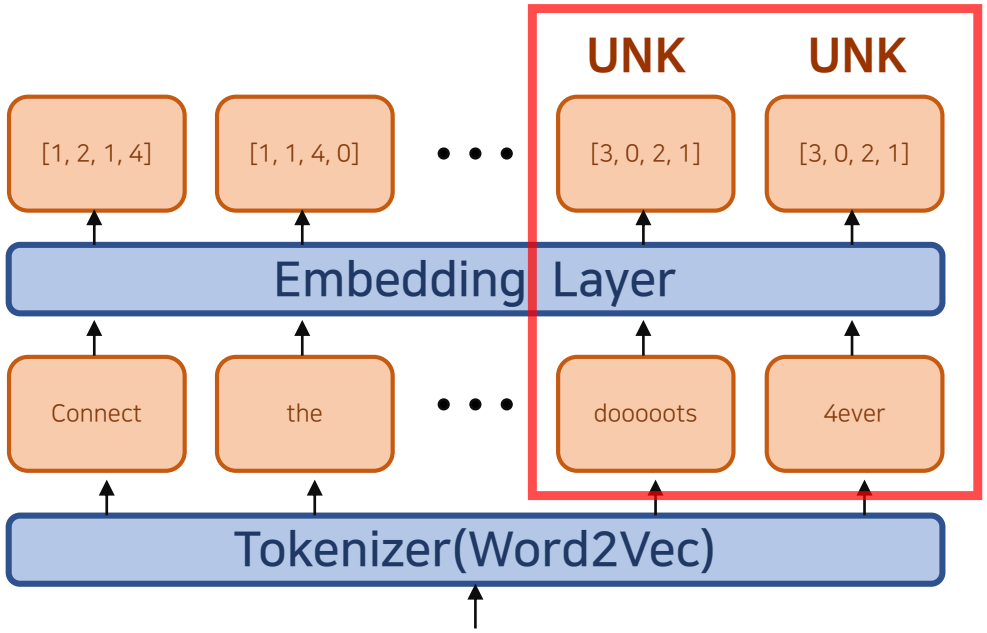
✓ Byte-Pair-Encoding

1. 단어를 문자 단위로 분해 …… connect
2. 전체 코퍼스에서 가장 자주 등장하는 연속된 두 문자열 병합 …… c## o## n## n## e## c## t
3. 병합된 토큰을 기반으로 분해 결과 조합 …… n## n## -> nn##
4. 종료 조건(병합 횟수, 토큰 개수 등)을 만족할 때 까지 1 ~ 3 반복 …… c## o## nn## e## c## t

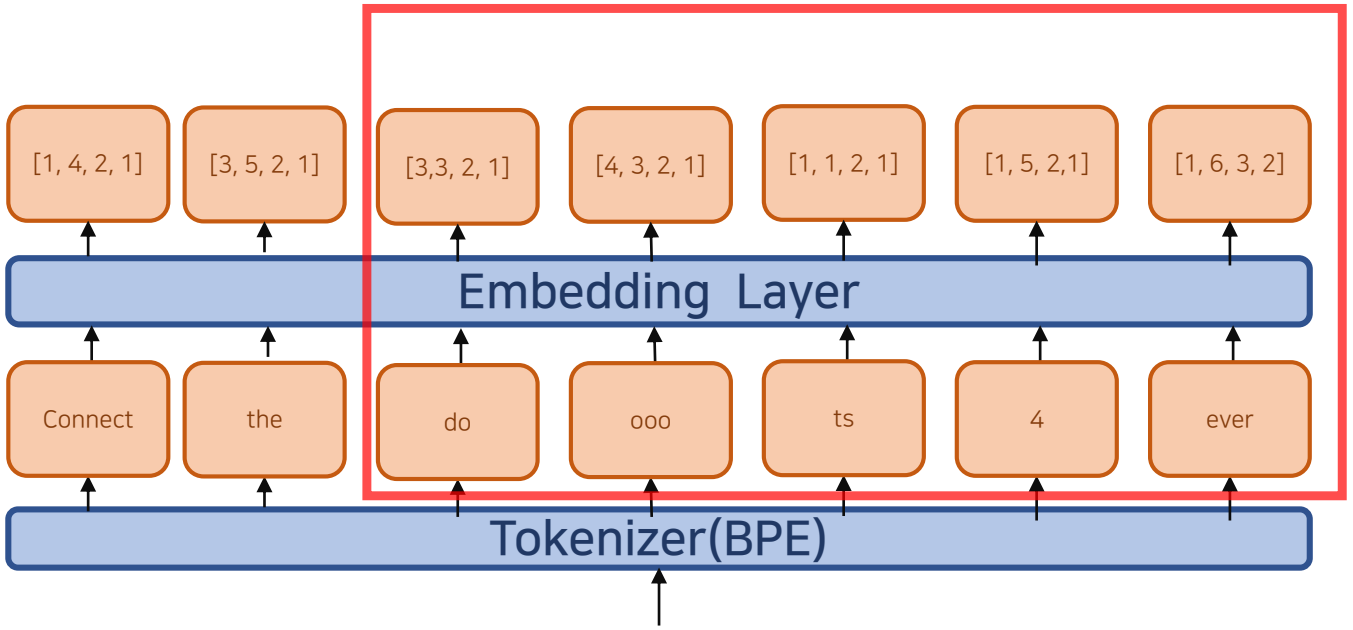
01 Problems of Vocab

Out of Vocabulary Problem

- **BPE와 단어기반 토크나이징 비교**



Connect the infinite dooooots 4ever



Connect the infinite dooooots 4ever

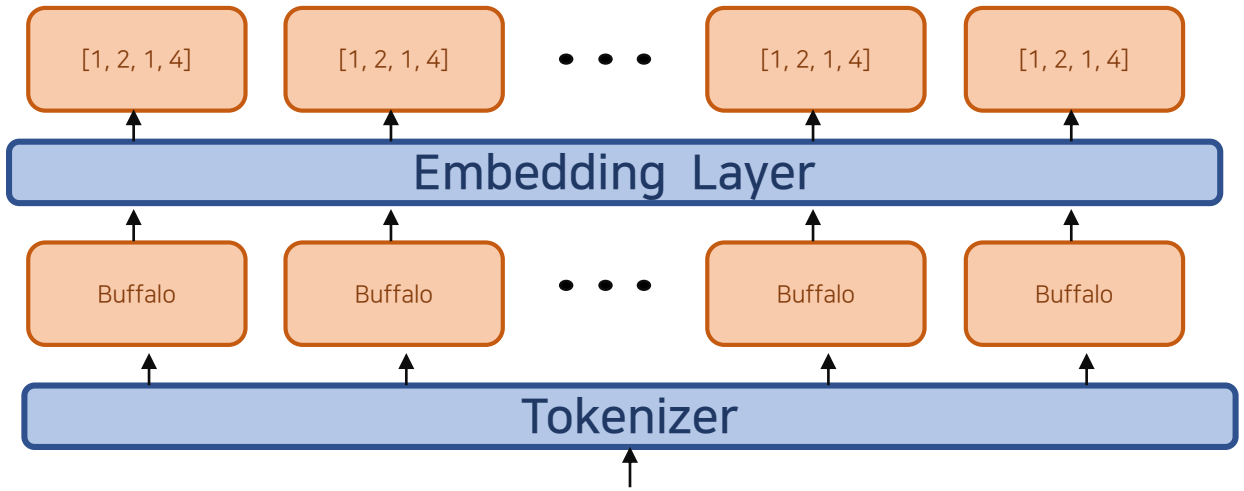
- ✓ 신조어, 오타에 대응이 가능
- ✓ OOV 문제에서 비교적 자유로움

01 Problems of Vocab

Contextualized Word Embedding

✓ Fixed Word Embedding

- 각 단어에 대해 사전에 학습된 임베딩 벡터 부여
- 문맥을 고려하지 못한 임베딩 방식
- 다른 문맥임에도 동일 벡터가 맵핑
- 임베딩 레이어로 문맥 반영 불가



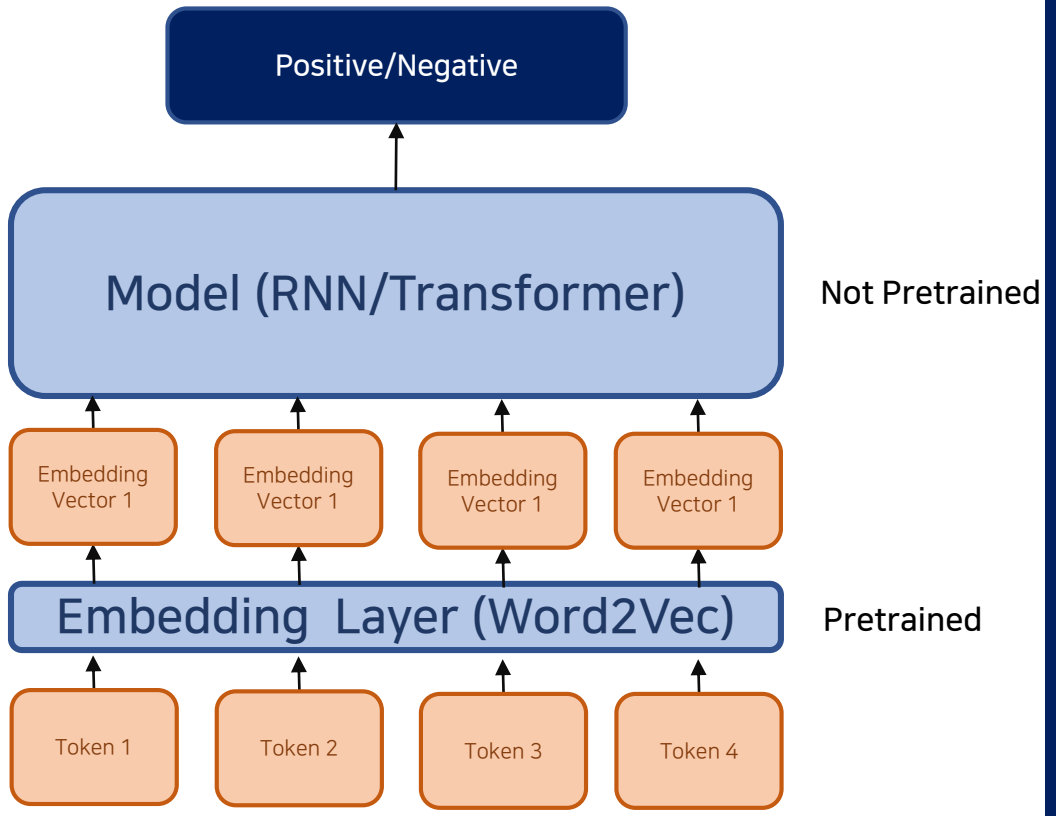
01 Problems of Vocab

Contextualized Word Embedding

✓ Fixed Word Embedding

- 각 단어에 대해 사전에 학습된 임베딩 벡터 부여
- 문맥을 고려하지 못한 임베딩 방식
- 다른 문맥임에도 동일 벡터가 맵핑
- 임베딩 레이어로 문맥 반영 불가

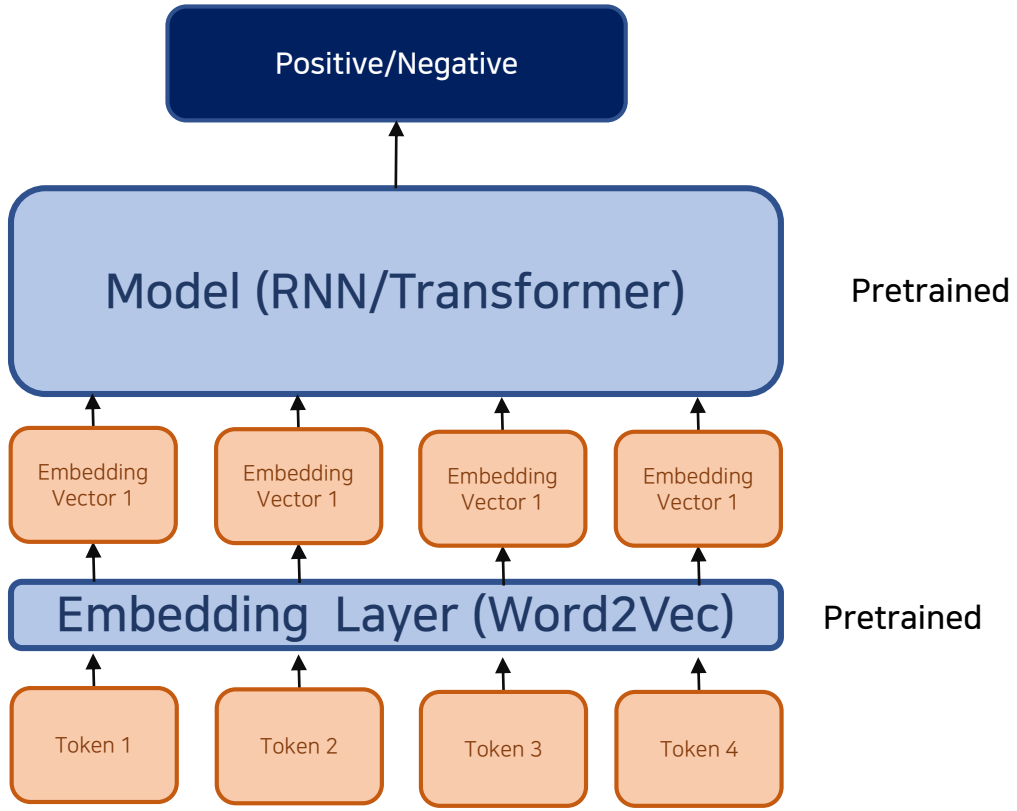
\+ 각 태스크마다 대량의 데이터 필요



02 Pretrain

Pretrain Whole Model

- ✓ **Pretrain(사전학습)**
  - Input의 일부를 훼손
  - 모델이 input을 복원
  - 의미적/문법적 구조 학습
- ✓ **Downstream Task**
  - 실제 수행하고자 하는 태스크
  - 사전학습된 모델의 weight를 initial weigh로 삼아 학습(fine-tuning) 진행
  - QA/Sentiment Analysis/NLI 등등



02 Pretrain

Pretrain Whole Model

- **Pretrain task : 문장의 일부를 변형하고, 해당 부분의 단어를 예측**

1. 고려대학교는 서울시 성북구 ___에 위치해 있다. [단편적 사실]
2. 나는 어제 저녁으로 돼지고기를 먹__다. [문법 정보]
3. 아이유는 피부가 좋기로 유명하다. ___는 피부 관리를 위해 니베아 로션을 바른다고 한다. [상호참조]
4. 어제 여자 컬링 국가대표팀의 일본전을 봤는데 극적으로 이기는 모습을 보니 ___ [감성]
5. 재희가 커피를 타려고 주방에 갔다. 재희 옆에 서있던 건호는 잠이 와서 ___를 떠났다. [추론]
6. 이 수열 참 이쁘지 않아? 1, 1, 2, 3, 5, ___, 13… [단순 계산]

02 Pretrain

Pretrain Whole Model

✓ **Encoder : input이 연속적으로 레이어 통과**

- Output : Contextual Embedding

  (batch, sequence length, embed dim)

- 입력 문장에서 맥락/의미/문법 구조 등을 파악

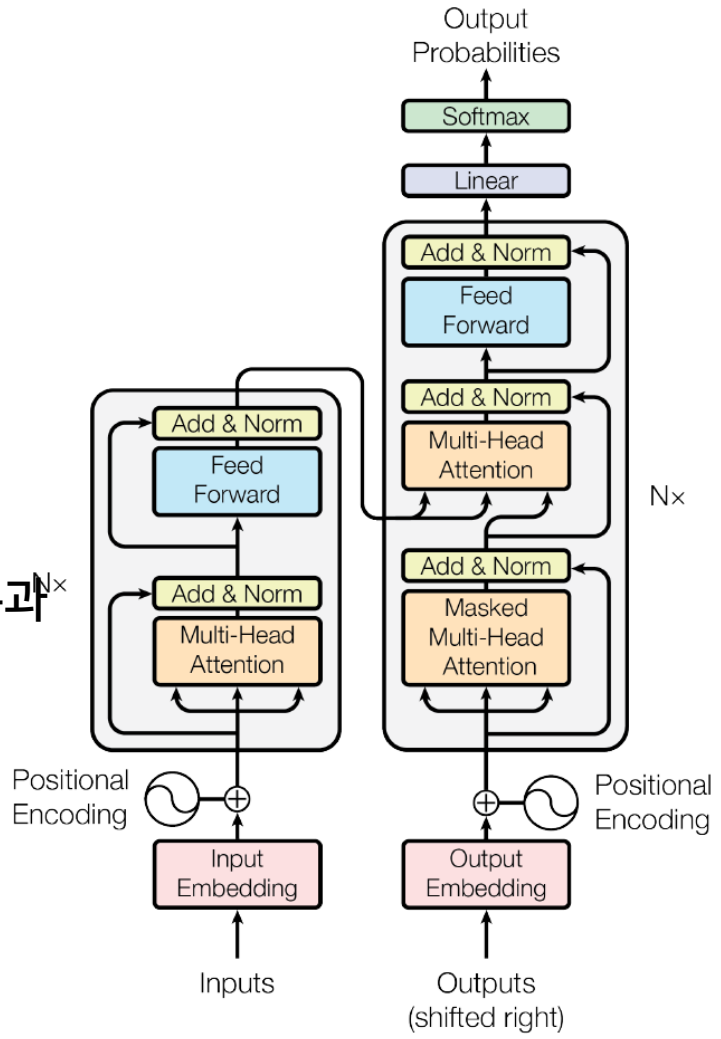
✓ **Decoder : input과 encoder output이 연속적으로 레이어 통과**

- Output : 조건부 문장 생성

  (batch, sequence length, vocab dim)

- 입력 문장과 이전 생성 문장을 바탕으로 문장 생성

➡ **인코더와 디코더가 수행하는 역할이 다름**



02 Pretrain

Objective functions and …

✓ Language Modeling

- $L = \sum \log P(w_t | w_0, \ldots, w_{t-1})$
- 별도의 label이 없어도 됨 (다량의 데이터 확보 가능)
- LM은 그 자체로 언어구조/문법 및 의미적 정보 파악을 가능케 함*

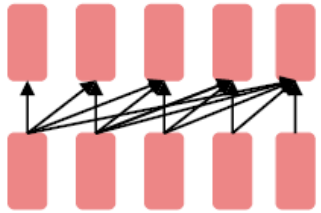
✓ SGD with Pretrain/Finetune

- $\hat{\theta} = \min_{\theta} L_{pretrain}(\theta)$
- $\min_{\theta} L_{finetune}(\theta)$ $(initial\ weight = \hat{\theta})$
- Random initializing θ에 비해 $\hat{\theta}$ 가 global optimum에 가까울 수 있다.
- -> 적은 데이터로도 잘 학습, 쉽게 수렴

Radford, Narasimhan, et al. Improving Language Understanding by Generative Pre-Training

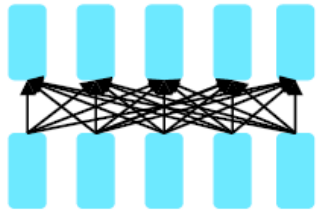
02 Pretrain

모델 종류



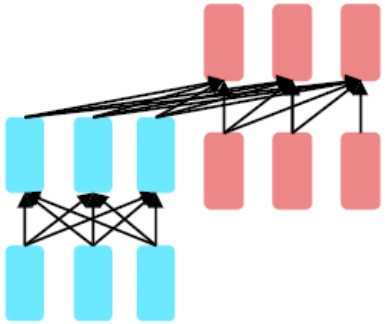
Decoders

- ✓ 일반적인 Language Model
- ✓ 이전 시점의 토큰을 조건으로 하는 생성모델



Encoders

- ✓ 양방향 토큰 정보를 활용할 수 있음
- ✓ 이전에 배운 Language Model로 pretrain 불가

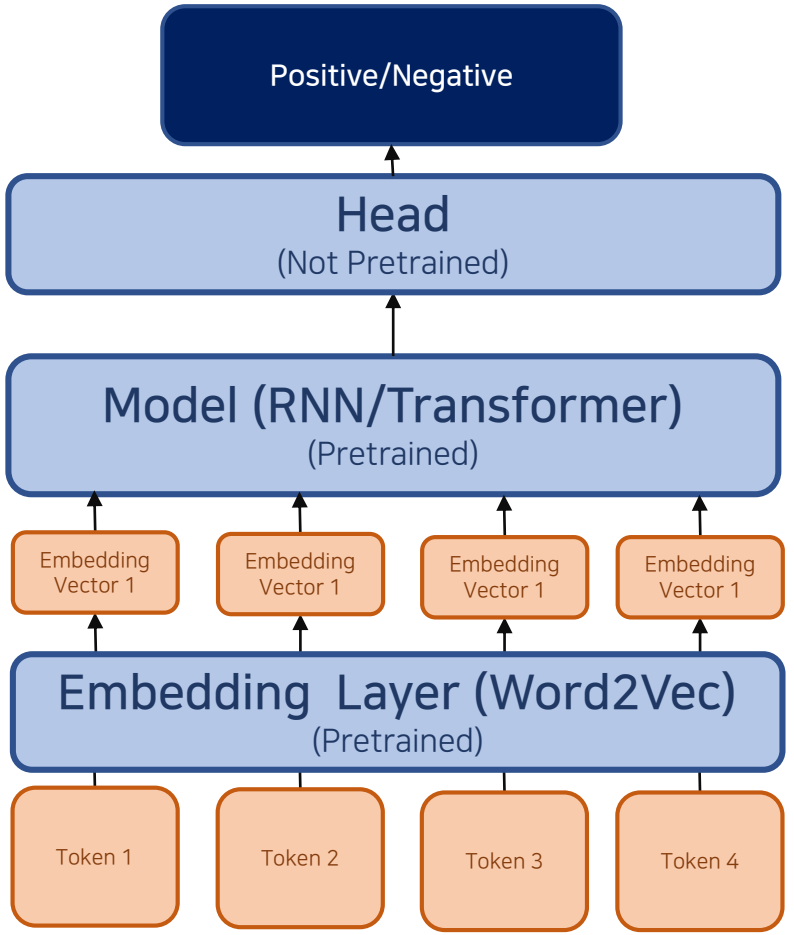


Encoder-Decoders

- ✓ 인코더가 양방향 정보를 종합
- ✓ 디코더가 생성에 집중
- ✓ 어떤 방법으로 pretrain해야 하나?

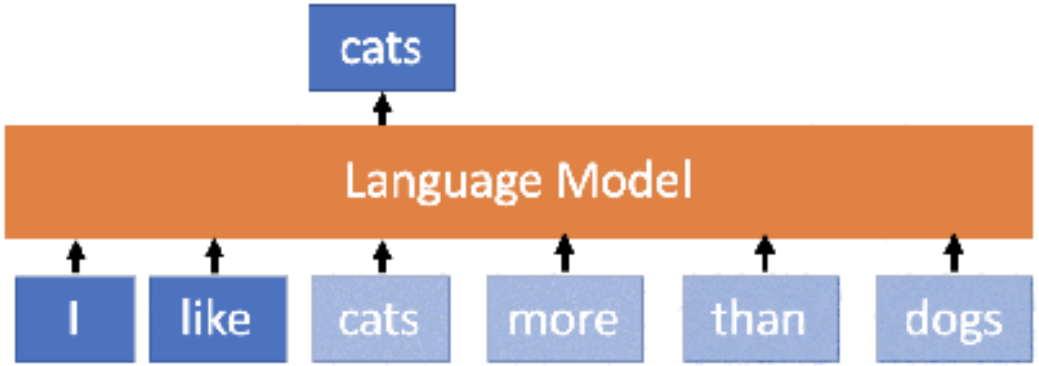
02 Pretrain
용어

- ✓ Pretrain task : Pretraining 시 수행하는 pretain을 위한 태스크
- ✓ Downstream task : 실제 수행할 구체적인 태스크
- ✓ Head : pretrain된 모델에 붙어 다운스트림 태스크에 맞춘 레이어.
  - Finetuning 이전에 학습이 전혀 되어 있지 않은 상태
- ✓ Pretrained model : pretrain task를 수행한 레이어
  - Finetuning 이전에 이미 어느정도 학습된 상태



03 GPT-1
Pretrain

- ✓ 트랜스포머 디코더만 사용
- ✓ 단방향 모델 $\mathrm{P}(w_i) = P(w_i \mid w_0, \ldots, w_{i-1})$
- ✓ 일반적인 LM을 통해 pretrain
- ✓ Teacher Forcing 이용
- ✓ LM은 레이블이 필요없음
  - 대량의 데이터 확보 가능
  - Train data size : BooksCorpus(800M Words)



$$L = \sum_i \log P(w_i \mid w_0, \ldots, w_{i-1}; \theta)$$

$$h_0 = UW_e + W_p$$

$$h_l = Decoder\ block_i(h_{l-1})$$

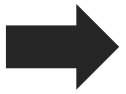
$$P(w_i) = softmax(h_n W_e^T)$$

https://jalammar.github.io/illustrated-gpt2/

03 GPT-1
Details

- **Transformer와 비교**

	Transformer	GPT-1
Model Arch.	Encoder-Decoder	Decoder
model dim	512	768
FFNN dim	2048	3072
Vocab size(BPE)	3.7M	4M
# heads	8	12
# layers	6	12
# params	6.5M	110M



✓ 파라미터 수의 증가

✓ Pretrain 효과 발생

03 GPT-1

Finetuning

- **총 4가지 finetuning 방법론 실험**

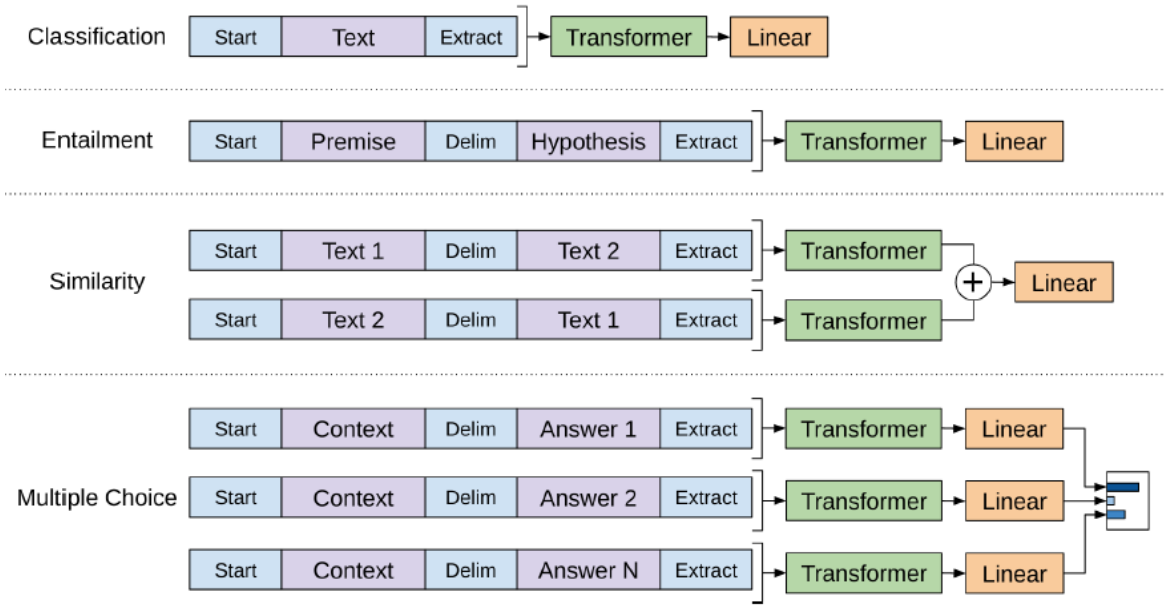
✓ Classification : 긍/부정, 문법 오류 여부

✓ Entailment : 주어진 문장들의 관계 분류

✓ Similarity : 두 문장 간 의미적 유사도 파악

✓ Multiple Choice :

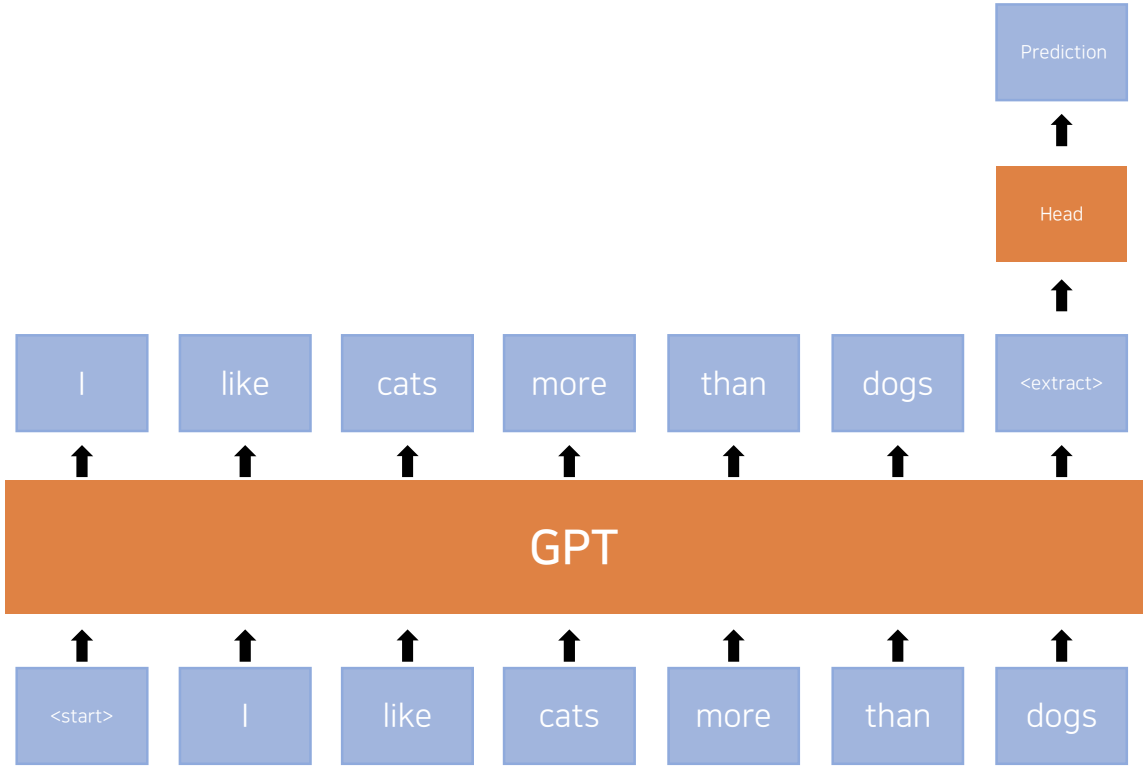
주어진 문제에 대한 보기 중 정답 고르기



03 GPT-1

Finetuning

- ✓ 문장 종료 토큰 : <Extract>
- ✓ 문장 마지막에 <Extract>을 생성하도록 함
- ✓ Teacher Forcing 이용하여 학습
- ✓ <Extract> 위치의 출력값을 head에 입력



03 GPT-1
Finetuning

- **Natural Language Inference(NLI) Result**

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

✓ 단일 태스크에 학습된 모델보다 좋은 성능 달성

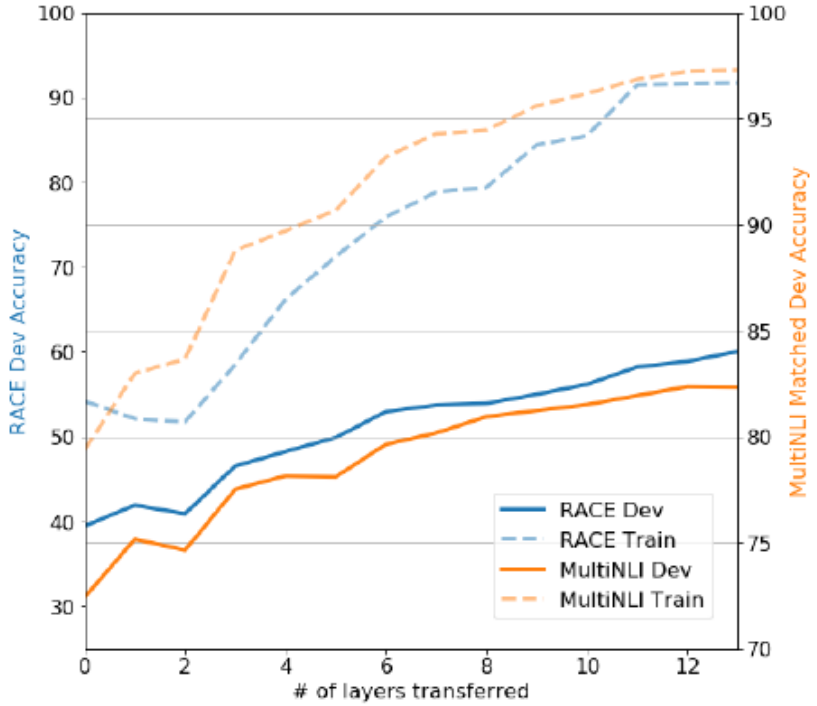
✓ 다양한 태스크에 학습된 모델보다 대부분 좋은 성능 달성

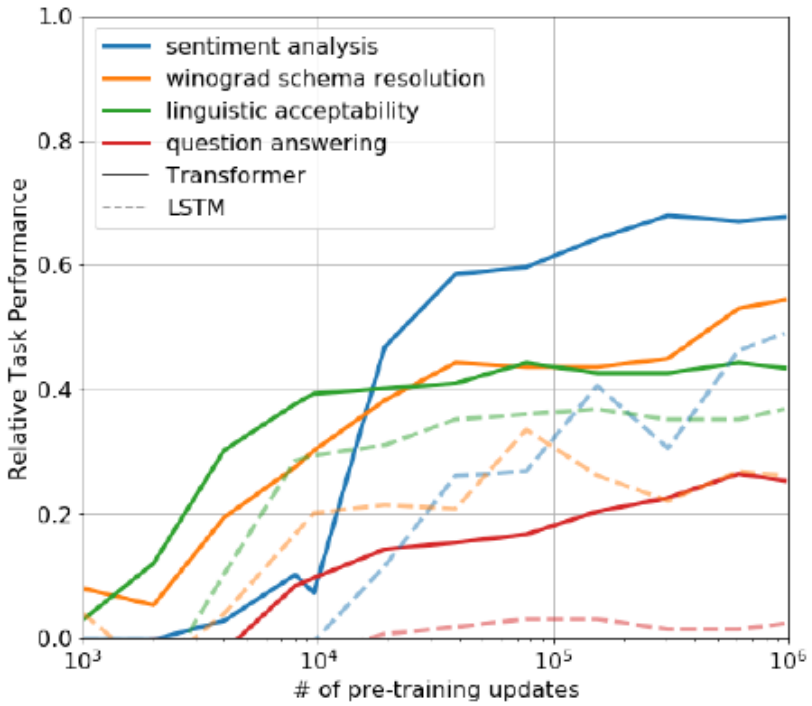
✓ 비교적 적은 에폭(3)으로도 SOTA를 달성

03 GPT-1

Finetuning

- **Pretrain이 다운스트림 태스크에 효과적임을 보임**





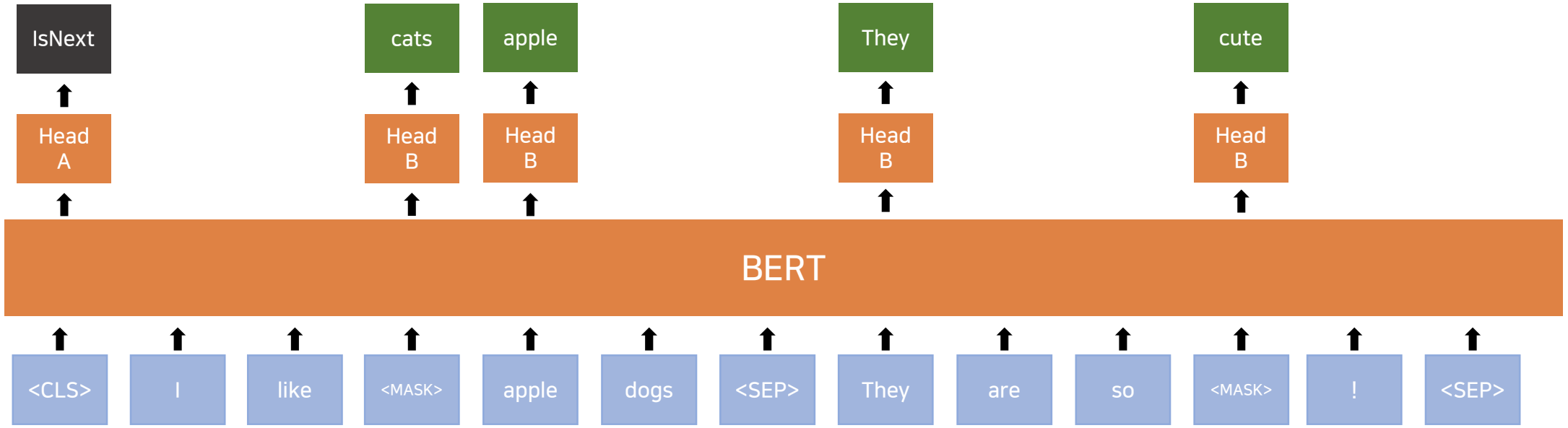
✓ Pretrain 레이어를 사용할수록 성능이 향상되는 모습

✓ Pretrain을 진행한 모델일수록 성능이 향상되는 모습

Radford, Narasimhan, et al. Improving Language Understanding by Generative Pre-Training

04 BERT
Pretrain

- **BERT의 pretrain task는 MLM과 NSP 두 가지임**



✓ 트랜스포머 인코더만 사용

✓ 양방향 모델

✓ MLM과 NSP를 통해 학습

$$h_0 = UW_e + W_p$$

$$h_l = Encoder\ block_i(h_{l-1})$$

$$P(w_i) = softmax(h_n W_e^T)$$

04 BERT

MLM & NSP

- **Masked Language Modeling (MLM)**

$$L_{MLM} = -\sum_{i \in M} \log P(w_i \mid w_0, \ldots, w_n; \theta)$$

(M : Masked tokens, n : sequence length)

✓ 양방향 모델 – 양방향 토큰의 정보를 활용
  - Language Model 목적함수 사용 불가

✓ 입력 토큰의 15%를 변형하고, 이를 예측
  - 80% : [MASK]
  - 10% : 랜덤한 다른 토큰
  - 10% : 원본 토큰

✓ Inference 상황과 동일한 환경

✓ 모델이 언어구조를 학습

04 BERT

MLM & NSP

- **Next Sentence Prediction (NSP)**

$$L_{NSP} = -\log \hat{y} P(IsNext \mid w_0, \ldots, w_n) - (1 - \hat{y})(1 - P(IsNext \mid w_0, \ldots, w_n))$$

(M : Masked tokens, n : sequence length)

- ✓ 많은 다운스트림 태스크에서 문장 간 관계를 다룸
- ✓ 이를 반영한 pretrain task 필요
- ✓ 두 문장 입력
  - ✓ 50% : 학습 데이터에서 연속된 문장
  - ✓ 50% : 서로 다른 문서에서 사용된 문장
- ✓ 연속된 문장 여부 판단(IsNext)

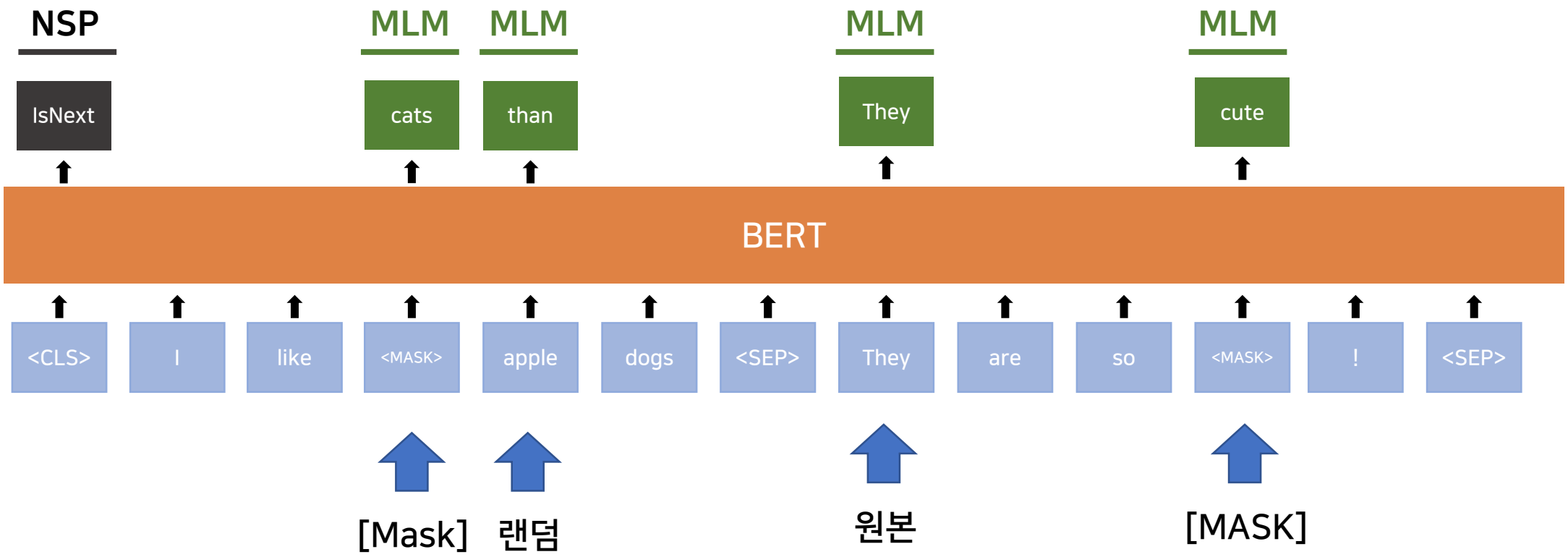
→

- ✓ 문장 간 관계 학습
- ✓ <CLS> 토큰이 입력 문장의 정보를 담음

04 BERT

MLM & NSP

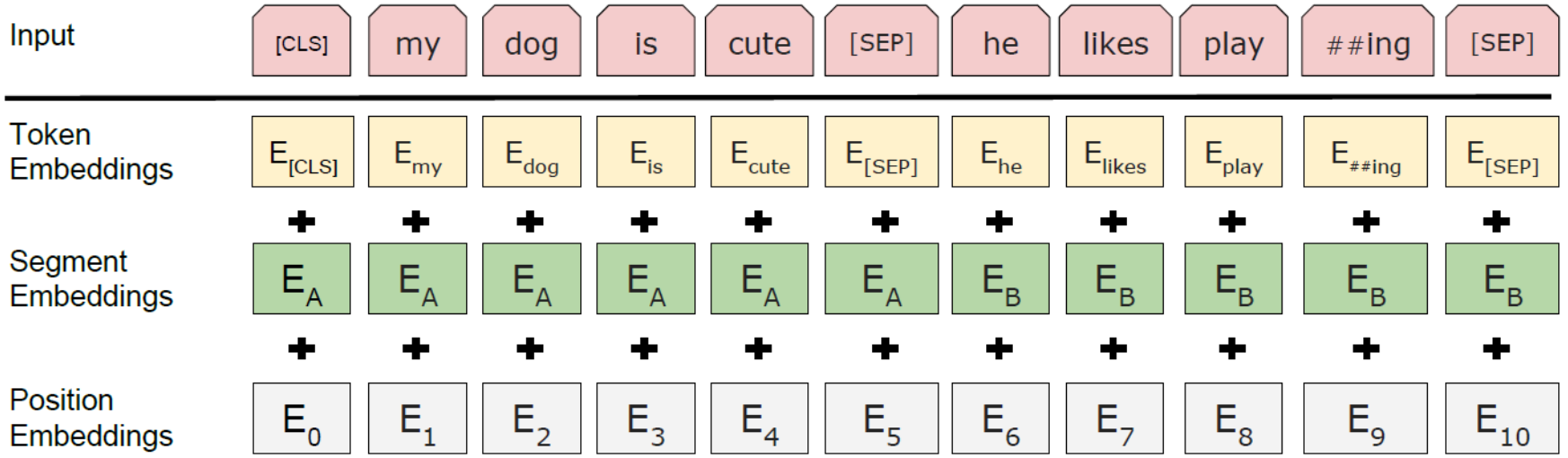
- **Pretrain Tasks의 모습**



=> 이후 RoBerta에서 NSP가 학습에 도움이 되지 않고, Bert가 과소적합 되어 있음을 보임

04 BERT
Input

- **BERT는 Token, Segment, Position 3 가지의 임베딩 벡터를 더하여 입력 임베딩을 생성**



- Token Embedding : Wordpiece 기반 토큰 단위 임베딩
- Position Embedding : 트랜스포머의 positional encoding
- Segement Embedding : 두 문장을 구분하기 위한 임베딩

https://ratsgo.github.io/nlpbook/docs/preprocess/bpe/

Devlin, Chang, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

04 BERT
Details

- **두가지 버전의 모델 발표**
  - BERT-Base : 12 layer, 768 hidden state dim, 12 attention head, 110M params
  - BERT-Large : 24 layers, 1024 hidden state dim, 16 attention head, 340M params
- **학습 데이터**
  - BooksCorpus(800M words) -> GPT pretrain 데이터
  - English Wikipedia(2500M words)
- Pretraing은 64개의 TPU로 4일에 걸쳐 이루어짐
- Finetuning은 단일 GPU로 이루어짐

04 BERT
Details

- **GPT, Transformer와의 비교**

	Transformer	GPT-1	BERT
Model Arch.	Encoder-Decoder	Decoder	Encoder
model dim	512	768	768
FFNN dim	2048	3072	3072
Vocab size(BPE)	3.7M	4M	3.7M
# heads	8	12	12
# layers	6	12	12
# params	6.5M	110M	110M

✓ GPT-1과 동일한 파라미터 수

✓ Model arch만 다름

04 BERT
Results

- **다양한 태스크에 대해 SOTA의 성능 달성**

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**

✓ **GPT-1(110M) 와 동일한 파라미터 수(110M)로 더 좋은 성능을 보임**

- 양방향 모델링에서 오는 이점

✓ **모델의 크기를 키움으로서 더 성능을 높일 수 있음을 보임.**

✓ **하지만 인코더 모델의 한계로 생성 다운스트림 태스크에 파인튜닝이 힘듬.**

04 BERT
RoBERTa

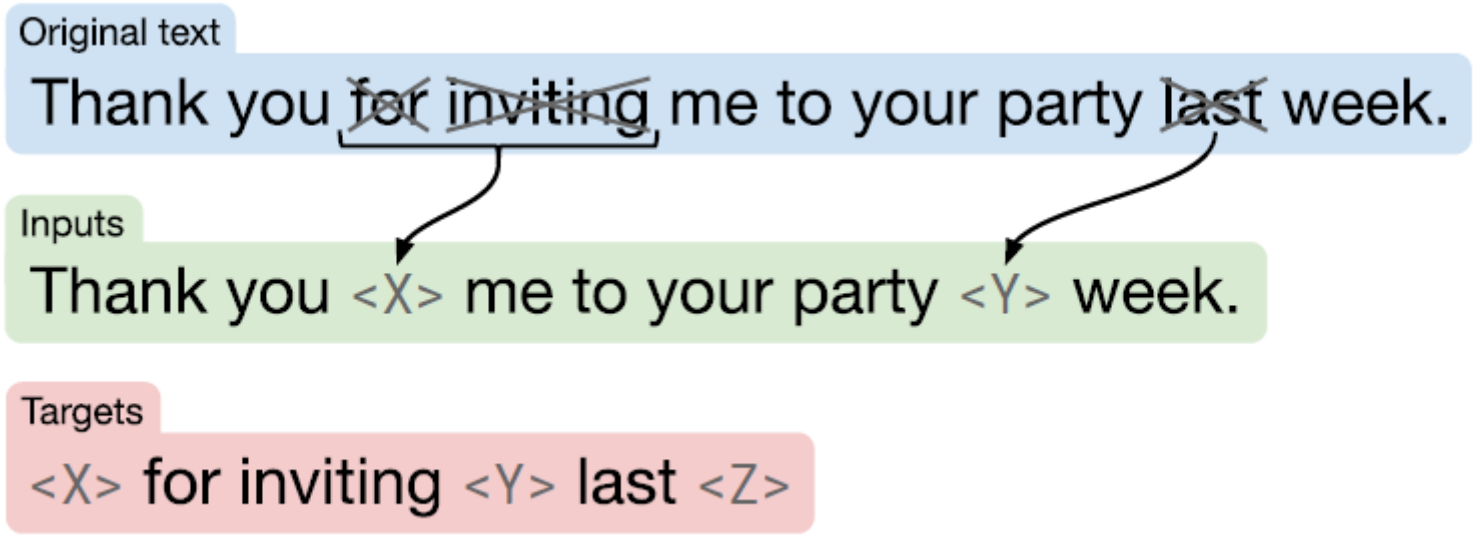
- **Bert의 구조를 그대로 가져오되, NSP를 제외하고 추가적인 하이퍼파라미터 튜닝 진행**

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**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7

✓ Bert의 성능을 더 높일 수 있음을 밝힘

✓ Pretrain 시 추가적인 데이터셋과 에포크로 더 높은 성능 달성

05 T5
Pretrain



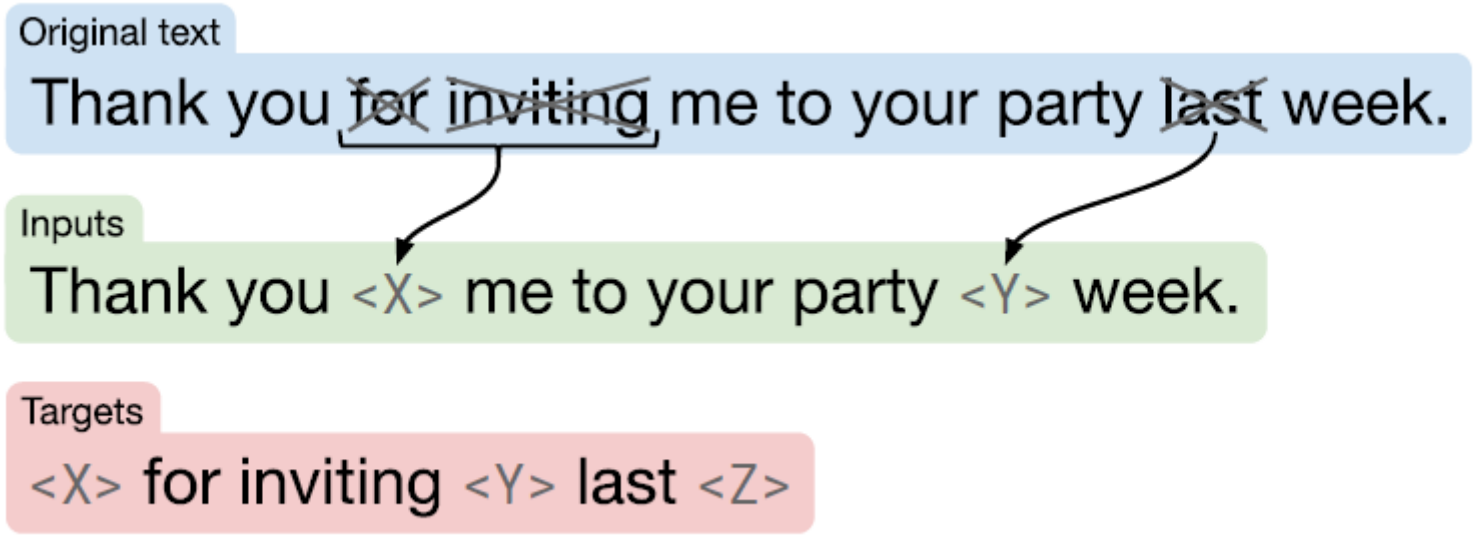
- ✓ 트랜스포머 구조를 가져오되 데이터셋과 모델을 키우고 pretrain 과정을 정교화
- ✓ 매우 다양한 다운스트림 태스크에 대한 성능 평가 진행
- ✓ Pretrain task로 span corruption 채택

Raffel, Shazeer, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

05 T5

Span Corruption

- **T5의 pretrain task**



✓ 인코더의 입력으로 MASK 사용

✓ 디코더의 출력으로 MASK ID와 해당하는 문장 생성

✓ 인코더는 MLM을 디코더는 LM을 수행하는 pretrain task

✓ 이를 통해 GPT보다 생성을 잘하고, BERT보다 파악을 잘하는 모델 구성

Raffel, Shazeer, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

05

T5

Span Corruption

- **BERT MASK와 다른 점**(1대1 MASKING vs 1대다 MASKING)

Span length	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Baseline (i.i.d.)	**83.28**	19.24	80.88	71.36	**26.98**	**39.82**	**27.65**
2	**83.54**	19.39	**82.09**	**72.20**	**26.76**	**39.99**	**27.63**
3	**83.49**	**19.62**	**81.84**	**72.53**	**26.86**	39.65	**27.62**
5	**83.40**	19.24	**82.05**	**72.23**	**26.88**	39.40	**27.53**
10	82.85	19.33	**81.84**	70.44	**26.79**	39.49	**27.69**

✓ 전체 토큰 중 15%를 25개의 MASK로 MASKING

✓ BERT는 모델이 하나의 토큰을 복원하지만, T5는 마스킹된 부분을 생성해야함

05 T5
Span Corruption

- **BERT MASK vs T5 MASK**

ORIGINAL : I like cats than dogs, cause they are so cute.

MASK RANDOM MASK ORGINAL
BERT : I <MASK> tomato <MASK> dogs, cause they are so cute.

MASK1 MASK2 MASK2
T5 : I <MASK1> <MASK2> dogs, cause <MASK3> cute.

- **GPT vs T5**

I like cats than → GPT-1 → dogs

I <MASK1> <MASK2> dogs, cause <MASK3> cute.

→ T5 → <MASK1> cats<MASK2>than<MASK3> they are

05

T5

Open Domain

- **기존의 Question & Answering task : 질문과 정답이 포함된 문장이 입력되면 모델이 정답 생성**

✓ Input : 안암에서 제일 맛있는 돈까스 집이 어디야?

✓ Context :

안암에서 제일 맛있는 돈가스집

예로부터 안암 정후는 돈가스집들의경쟁이 되게 치열했었죠 저는 그중에서 지푸라기를 제일 좋아했는데요

군대 복무중에 사라졌더라구요 너무 슬펐는데 동기가 돈가스냉면이라구 먹으러 가자고 했습니다.처음엔 무슨 그런 해괴한 음식을 먹냐고 뭐라 했지만 반강제로 따라가 먹은 돈가스는 지푸라기를 잇는 제 최애 돈가스집이 되었습니다

저는 냉면도 좋아하는 데 냉면도 안암에서 제일 맛있다고 느꼈습니다.
더 중요한건 그걸 같이 줘요
그날 저는 제 오버워치 아이디를 안암동돈가스냉면으로 바꿨습니다

이 집의 단점을 말하자면 돈가스를 너무 많이 줘서 남기게 만듭니다. 전 꾸역꾸역 먹으면 도미노 라지도 혼자서 한판 먹는데 이 집에서는 과식 안하려고 조금 남겨요. 정말 양이 많습니다 많이 못드시는 분들은 두분 가셔서 냉면 하나 돈가스 하나 시키시는걸 추천드려요

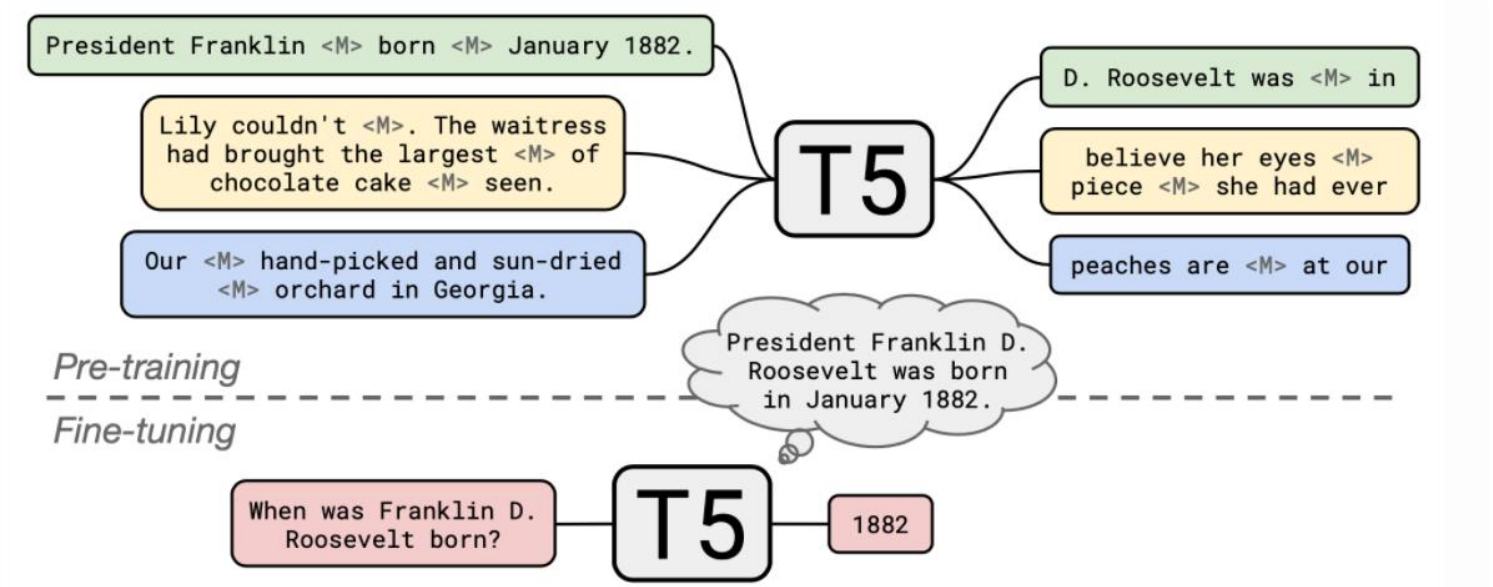
위치는 정대후문 더진국 국밥가게 뒤편 예전스파게티 팔던 자리에 있습니다 정확한 상호는 잘 기억 안 나네요 전 돈가스냉면이라구 불러서 이 집 좋아하는데 손님이 생각보다 없어서 추천드립니다 정말 맛있어요

✓ Output : 돈가스냉면

05 T5

Open Domain

- T5는 질문만 주어져도 어느정도 답변이 가능함



- 이는 정말 대량의 데이터셋(750GB)으로 학습했기 때문
- 모델이 정보를 파라미터화 하여 기억
  - ✓ Input : 안암에서 제일 맛있는 돈까스 집이 어디야?
  - ✓ Output : 돈가스 냉면

Raffel, Shazeer, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

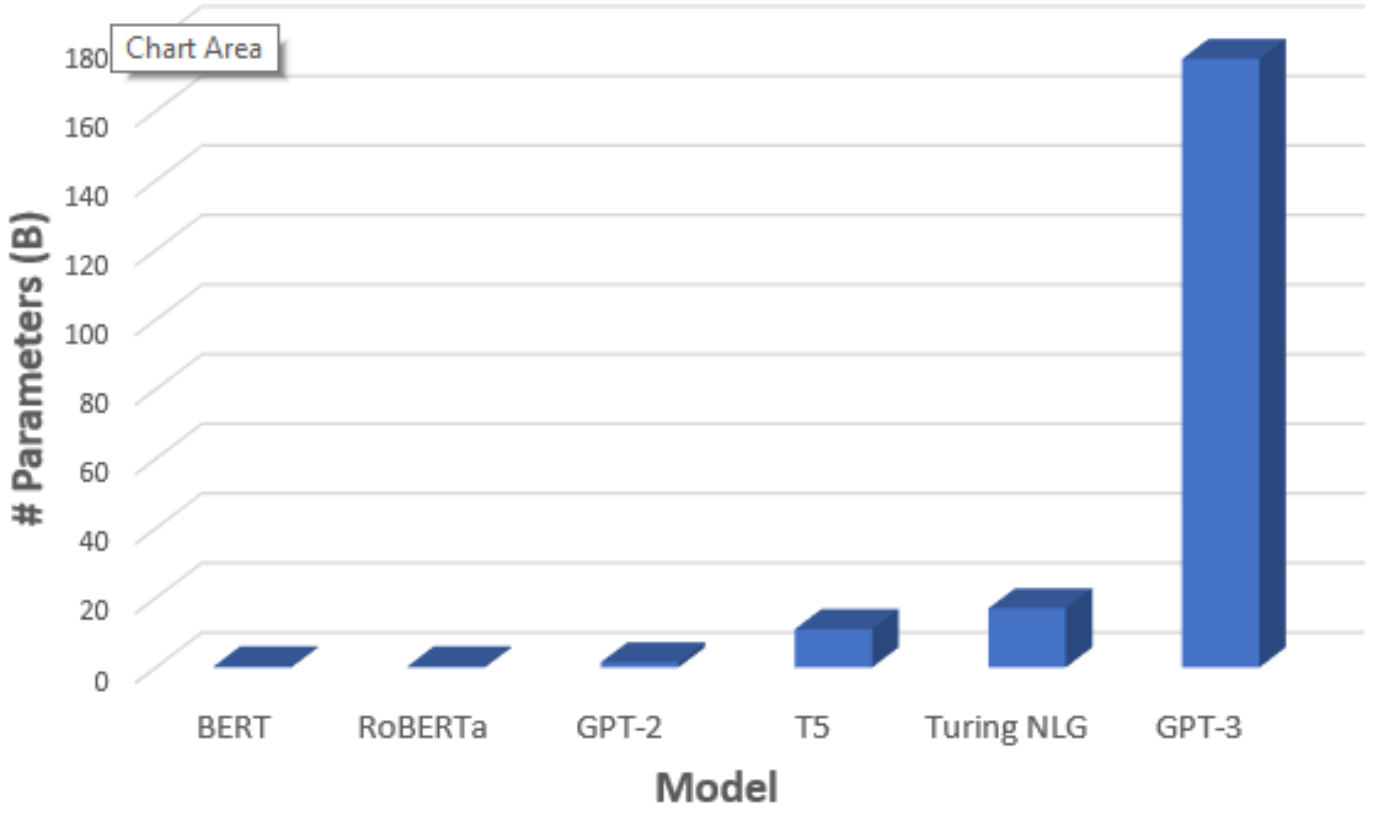
05 T5
Details

- **GPT, BERT, Transformer와의 비교 (large 기준)**

	Transformer	GPT-1	BERT	BERT
Model Arch.	Encoder-Decoder	Decoder	Encoder	Encoder-Decoder
model dim	512	768	1,024	1,024
FFNN dim	2,048	3,072	3,072	65,536
Vocab size(BPE)	3.7M	4M	3.7M	3.2M
# heads	8	12	16	128
# layers	6	12	24	24
# params	6.5M	110M	340M	11B = 11000M

06 결론

- 정말 많은 대형 사전학습 모델들



✓ 기하급수적으로 늘어나는 훈련 데이터 및 모델 크기

✓ 태스크 별 모델 설계에서 사전학습 모델로 패러다임 변화

06 결론

사전 학습 = 큰 모델 + 대량 데이터 + 적절한 태스크

- **모델의 성능은 파라미터 수와 비례**
  - ✓ 과적합을 방지하기 위해서는 대량의 데이터가 필수
  - ✓ 대량의 데이터와 큰 모델은 많은 자원을 소모
- **효과적인 학습을 위해서는 적절한 pretrain task를 구성해야 함**
- **적절한 pretrain task, 대량의 데이터, 거대 모델은 매우 효과적인 fine tuning을 보장함**

06 결론

- T5는 최적의 사전학습 방법론을 찾기 위해 정말 많은 태스크에 정말 많은 실험을 수행

Table	Experiment	Score Average	GLUE CoLA MCC	GLUE SST-2 Acc	GLUE MRPC F1	GLUE MRPC Acc	GLUE STSB PCC	GLUE STSB SCC	GLUE QQP F1	GLUE QQP Acc	GLUE MNLI$_m$ Acc	GLUE MNLI$_{mm}$ Acc	GLUE QNLI Acc	GLUE RTE Acc	CNN/DM R-1-F	CNN/DM R-2-F	CNN/DM R-L-F	SQuAD EM	SQuAD F1	SuperGLUE Score Average	SuperGLUE BoolQ Acc	SuperGLUE CB F1	SuperGLUE CB Acc	SuperGLUE COPA Acc	SuperGLUE MultiRC F1	SuperGLUE MultiRC EM	SuperGLUE ReCoRD F1	SuperGLUE ReCoRD EM	SuperGLUE RTE Acc	SuperGLUE WiC Acc	SuperGLUE WSC Acc	WMT EnDe BLEU	WMT EnFr BLEU	WMT EnRo BLEU
1	★Baseline average	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
1	Baseline standard deviation	0.258	1.111	0.569	0.729	1.019	0.374	0.418	0.108	0.070	0.291	0.231	0.361	1.393	0.065	0.065	0.068	0.343	0.298	0.416	0.565	3.237	2.960	2.741	0.716	1.011	0.570	0.579	1.228	0.860	2.029	0.112	0.090	0.108
1	No pre-training	66.22	12.29	80.62	81.42	73.04	72.58	72.97	81.94	86.62	68.02	67.98	75.69	58.84	39.19	17.60	36.69	50.51	61.07	53.04	65.38	71.61	76.79	62.00	59.10	0.84	20.33	17.98	54.15	54.08	68.58	26.86	39.77	24.04
2	★Enc/dec, denoising	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
2	Enc/dec, shared, denoising	82.81	83.24	91.86	91.58	88.24	87.45	87.58	88.69	91.60	83.88	84.01	90.23	75.65	41.11	18.78	38.48	80.63	88.49	70.73	77.13	86.04	96.43	65.00	66.16	22.98	68.95	68.09	76.76	68.18	76.96	26.72	39.03	27.46
2	Enc/dec, 6 layers, denoising	80.88	80.26	92.09	91.51	87.99	87.24	86.76	87.25	90.97	82.20	82.41	88.83	71.48	40.83	18.97	38.51	77.59	86.07	68.42	75.79	94.70	92.86	67.00	61.02	19.62	61.26	60.33	73.20	65.60	76.00	26.38	38.40	26.95
2	Language model, denoising	74.70	24.50	90.60	86.08	78.92	58.22	55.42	84.40	88.99	78.72	77.95	86.02	64.62	39.49	17.93	36.51	61.14	71.57	55.02	65.47	60.08	71.43	58.00	43.03	2.94	53.35	52.31	63.07	53.62	65.46	25.09	35.28	25.86
2	Prefix LM, denoising	81.82	49.99	92.43	91.43	88.24	87.20	86.98	88.41	91.39	82.32	82.93	88.71	74.01	40.46	18.61	37.90	78.94	87.31	68.11	76.50	93.37	91.07	60.00	63.43	21.20	65.03	64.11	71.48	63.67	73.08	26.43	37.98	27.39
2	Enc/dec, LM	79.56	42.03	91.86	91.64	88.24	87.13	87.00	88.21	91.15	81.68	81.66	88.84	68.70	40.67	18.89	38.13	76.02	84.86	64.29	72.23	85.74	89.29	57.00	60.53	16.26	59.28	58.30	68.34	64.89	70.19	26.27	39.17	26.86
2	Enc/dec, shared, LM	79.60	44.83	92.09	90.20	85.78	86.05	85.87	87.77	91.02	81.74	82.20	89.16	68.54	40.16	18.13	37.89	76.55	84.86	63.60	70.49	91.41	87.50	55.00	60.21	16.89	57.83	56.73	63.54	63.48	70.19	26.62	39.17	27.05
2	Enc/dec, 6 layers, LM	76.67	38.72	91.40	90.40	86.52	86.82	86.40	87.87	91.03	80.99	80.92	88.05	66.10	40.26	18.26	37.70	75.32	84.06	64.06	71.38	85.25	89.29	60.00	57.66	16.79	55.22	54.30	66.76	63.05	71.15	26.13	38.42	26.89
2	Language model, LM	73.78	28.83	89.79	85.25	78.68	84.22	84.00	84.88	88.70	74.94	76.77	84.84	58.84	38.97	17.54	36.37	53.81	64.55	58.81	64.22	89.92	71.43	64.00	53.04	1.05	46.81	45.78	58.84	56.74	69.23	25.23	34.31	25.38
2	Prefix LM, LM	79.68	41.26	92.09	90.11	86.27	86.82	86.32	88.35	91.35	81.71	82.02	89.04	68.59	39.66	17.84	37.13	76.87	85.59	64.86	71.47	93.37	91.07	57.00	58.67	16.89	59.25	58.16	64.26	66.30	71.15	26.28	37.51	26.76
4	Language modeling with prefix	80.69	44.22	93.00	91.68	88.48	87.20	87.18	88.39	91.41	82.66	83.09	89.29	68.56	40.71	18.94	38.16	77.99	86.43	65.27	73.55	83.96	87.50	55.00	59.65	18.89	61.76	60.76	68.59	65.67	73.08	26.86	39.73	27.49
4	BERT-style (Devlin et al., 2018)	82.96	82.49	92.55	92.19	89.56	87.68	87.66	88.47	91.44	83.60	84.05	90.33	76.45	41.27	19.17	38.72	80.65	88.24	69.85	76.48	94.37	94.64	61.00	63.29	25.08	66.76	65.85	72.20	69.12	76.00	26.78	40.03	27.41
4	Deshuffling	73.17	22.82	87.16	86.88	81.13	84.03	83.82	86.38	89.90	76.30	76.34	84.18	58.84	40.76	18.69	38.10	67.61	76.76	58.67	69.17	83.70	78.57	56.00	59.85	12.70	45.82	44.36	57.04	54.89	68.27	26.11	39.30	25.62
5	BERT-style (Devlin et al., 2018)	82.96	82.49	92.55	92.19	89.56	87.68	87.66	88.47	91.44	83.60	84.05	90.33	76.45	41.27	19.17	38.72	80.65	88.24	69.85	76.48	94.37	94.64	61.00	63.29	25.08	66.76	65.85	72.20	69.12	76.00	26.78	40.03	27.41
5	MASS-style (Song et al., 2019)	82.32	47.01	91.63	92.83	89.71	88.21	88.18	88.58	91.44	82.96	83.67	90.02	77.26	41.16	19.16	38.55	80.10	88.07	69.28	75.08	84.98	89.29	63.00	64.46	23.60	66.71	66.01	72.20	67.71	78.85	26.79	39.89	27.55
5	★Replace corrupted spans	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
5	Drop corrupted tokens	84.44	86.04	92.89	92.79	89.95	87.28	86.95	88.66	91.54	83.94	83.92	90.74	74.42	41.27	19.51	38.70	80.82	88.26	68.67	75.90	96.02	94.64	66.00	65.06	23.92	61.94	64.60	71.12	67.40	74.04	27.07	39.76	27.82
6	Corruption rate = 10%	82.82	82.74	92.09	91.16	88.24	88.19	88.15	88.47	91.40	83.50	84.81	90.33	76.46	41.06	19.00	38.83	80.58	88.36	69.55	74.98	92.37	92.86	62.00	66.04	24.66	67.93	67.09	76.76	67.24	76.96	26.87	39.28	27.44
6	★Corruption rate = 15%	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
6	Corruption rate = 25%	83.00	83.47	93.00	92.44	89.46	87.36	87.26	88.68	91.83	84.44	84.15	90.77	74.01	41.89	19.64	39.14	80.96	88.61	70.48	76.39	93.02	92.86	68.00	65.46	24.66	68.20	67.39	75.65	67.87	72.12	27.04	39.83	27.47
6	Corruption rate = 50%	81.27	80.26	91.63	91.11	87.99	87.87	87.64	88.70	91.57	83.64	84.10	90.24	70.76	41.81	19.52	38.89	79.80	87.76	70.33	75.02	93.06	92.86	68.00	62.97	24.13	64.94	64.13	72.20	68.50	77.88	27.01	39.90	27.49
7	★Baseline (i.i.d.)	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
7	Average span length = 2	83.54	83.82	92.20	93.05	90.44	87.85	87.71	88.42	91.40	84.28	84.46	90.88	77.62	41.23	19.59	38.69	82.09	89.69	72.20	77.08	90.45	91.07	70.00	66.28	26.13	71.34	70.61	76.45	68.34	78.85	26.76	39.99	27.63
7	Average span length = 3	83.49	83.90	92.43	92.25	89.46	87.49	87.33	88.72	91.51	84.85	84.84	90.99	77.26	41.50	19.62	38.94	81.84	89.06	72.03	76.85	94.37	94.64	70.00	67.64	28.75	70.84	69.90	74.73	67.71	77.88	26.86	39.66	27.62
7	Average span length = 5	83.40	82.12	93.12	92.63	89.71	88.70	88.47	88.84	91.64	84.52	84.39	90.79	76.90	41.39	19.24	38.82	82.05	89.59	72.23	77.08	83.06	89.29	69.00	68.16	30.12	71.36	70.53	76.81	69.01	79.81	26.88	39.40	27.53
7	Average span length = 10	82.85	80.11	92.09	91.95	88.97	88.45	88.22	88.86	91.83	84.34	84.28	91.07	76.17	41.38	19.33	38.80	81.84	89.59	70.44	76.45	87.40	89.29	68.00	66.87	29.59	69.82	68.94	75.56	67.55	76.96	26.79	39.40	27.69
8	★C4	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
8	C4, unfiltered	81.46	78.01	91.63	92.12	89.46	87.79	87.60	88.31	91.27	82.30	82.54	88.71	72.20	41.09	19.14	38.54	78.78	87.04	68.04	75.75	89.17	91.07	62.00	65.52	25.60	62.42	61.58	69.68	67.08	72.12	26.55	39.34	27.21
8	RealNews-like	83.83	86.55	92.66	92.06	88.97	87.71	87.37	88.51	91.49	84.35	84.46	90.61	76.54	41.38	19.23	38.84	80.59	88.50	72.38	77.00	93.09	94.64	66.00	65.92	23.82	74.56	73.72	76.81	66.61	80.77	26.75	39.90	27.48
8	WebText-like	84.03	86.38	93.12	92.51	89.22	88.69	88.68	88.65	91.56	84.70	84.84	90.83	77.92	41.23	19.51	38.70	81.42	89.15	71.40	76.88	83.08	89.29	66.00	64.10	24.24	72.24	71.36	76.45	68.03	82.69	26.80	39.74	27.59
8	Wikipedia	81.85	78.53	92.32	91.67	88.24	88.62	88.40	88.37	91.54	82.61	83.25	90.06	77.26	41.39	19.51	38.81	81.29	89.18	68.01	76.12	86.03	80.36	67.00	65.01	25.92	69.03	68.06	74.73	67.08	76.92	26.94	39.69	27.67
8	Wikipedia + TBC	83.65	85.53	92.78	92.41	89.22	88.67	88.27	88.47	92.26	84.38	83.45	91.04	76.00	41.22	19.28	38.67	82.08	89.70	73.24	78.22	95.40	92.86	69.00	51.59	50.93	69.53	68.51	77.62	68.03	81.73	26.77	39.63	27.57
9	★Full data set	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
9	2^{23} (64 repeats)	82.87	83.82	92.78	91.79	88.73	87.56	87.58	88.73	91.54	84.07	84.21	90.89	75.65	41.18	19.19	38.67	80.97	88.60	72.03	76.78	92.56	92.86	66.00	65.11	26.78	69.35	68.49	76.81	67.24	82.69	26.83	39.74	27.63
9	2^{27} (256 repeats)	82.62	80.60	92.32	92.07	88.73	87.83	87.60	88.65	91.54	83.43	84.37	90.12	76.81	41.24	19.20	38.70	79.78	87.63	69.07	75.29	93.42	91.07	63.00	61.82	23.61	66.27	65.39	73.65	66.50	80.77	27.02	39.71	27.33
9	2^{29} (1,024 repeats)	79.55	43.84	91.28	89.32	85.05	85.92	85.74	88.05	91.09	81.29	81.72	87.90	69.31	40.66	18.87	38.13	76.27	84.58	64.76	72.63	83.97	82.14	64.00	59.39	17.94	56.94	56.04	64.98	65.20	73.08	26.38	39.06	26.80
9	2^{31} (4,096 repeats)	76.34	32.68	89.45	86.84	80.03	83.49	83.42	87.18	90.61	77.80	78.69	85.47	64.62	40.16	18.33	37.66	70.92	80.20	59.29	69.85	73.48	73.21	56.00	57.66	14.38	46.69	45.79	59.57	65.05	68.27	26.37	38.84	25.81
10	★All parameters	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
10	Adapter layers, $d = 32$	80.52	48.33	91.63	90.89	86.76	86.38	86.08	86.99	90.26	81.63	83.04	90.72	67.15	34.50	15.08	32.15	73.32	87.70	60.40	65.32	80.87	73.21	52.00	58.81	19.41	65.50	64.58	69.05	64.58	73.08	13.84	17.88	15.54
10	Adapter layers, $d = 128$	81.51	48.35	92.89	91.49	88.24	87.75	87.65	87.75	90.93	83.64	84.09	90.82	72.96	36.71	16.62	34.37	79.47	87.61	63.03	69.20	82.21	76.00	56.00	61.08	18.05	67.94	66.97	68.59	66.97	73.08	19.83	27.50	22.63
10	Adapter layers, $d = 512$	81.54	44.25	93.35	91.00	87.25	86.74	86.44	88.02	91.15	83.98	83.80	90.62	74.37	38.83	17.78	36.25	79.18	87.52	64.30	73.18	82.86	71.43	66.00	62.54	18.37	66.66	65.74	70.76	67.87	74.04	23.45	33.98	25.81
10	Adapter layers, $d = 2048$	82.62	49.86	92.55	91.50	87.99	88.46	88.35	88.36	91.40	83.65	83.18	90.66	76.13	39.44	18.50	37.06	79.40	87.36	68.61	74.83	88.00	91.07	58.00	61.10	18.89	66.73	66.06	73.29	71.16	76.96	25.64	36.92	26.93
10	Gradual Unfreezing	82.50	51.74	91.97	92.61	89.71	87.27	86.90	88.26	91.35	83.42	83.49	89.71	75.09	40.88	18.95	38.40	79.17	87.30	70.79	75.51	93.09	94.64	70.00	62.03	21.51	65.89	64.79	72.92	69.12	77.89	26.71	39.02	26.93
11	★Baseline (pre-train/fine-tune)	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
11	Equal	76.13	39.47	90.94	89.60	85.74	85.85	85.44	86.45	89.71	82.08	82.92	90.13	69.85	40.96	19.02	38.59	76.51	85.61	63.37	73.06	82.37	83.93	65.00	60.89	17.82	60.51	59.70	61.01	60.03	68.58	23.89	34.31	26.78
11	Examples-proportional, $K = 2^{16}$	80.45	42.07	91.97	90.97	87.50	86.41	85.04	86.89	90.10	83.01	83.66	90.74	72.66	41.16	19.04	38.69	77.26	85.72	69.56	76.67	86.38	85.29	70.00	65.23	27.91	62.78	61.95	76.90	65.83	73.08	24.35	34.99	27.10
11	Examples-proportional, $K = 2^{17}$	81.56	47.35	91.40	91.83	88.24	86.15	85.03	86.64	90.96	82.76	83.13	90.79	75.09	41.26	19.12	38.47	78.00	85.87	67.91	75.80	77.54	85.71	67.00	62.75	27.07	61.51	60.54	76.06	65.00	74.04	24.56	35.00	27.25
11	Examples-proportional, $K = 2^{18}$	81.67	46.88	91.63	91.99	88.73	87.68	87.20	86.95	90.35	83.30	84.01	91.47	75.29	40.96	19.07	38.43	78.17	86.74	67.94	76.67	78.88	87.50	62.00	67.70	30.85	63.43	62.54	76.53	65.67	67.31	24.37	35.19	27.39
11	Examples-proportional, $K = 2^{19}$	81.42	45.94	91.63	92.20	89.22	88.44	88.32	86.84	90.10	83.75	84.29	91.84	76.40	41.26	19.24	38.71	79.78	88.15	67.30	75.66	75.59	87.50	59.00	68.22	30.64	65.32	64.29	73.65	65.05	69.23	25.21	36.30	27.76
11	Examples-proportional, $K = 2^{20}$	80.80	42.55	92.78	91.27	87.99	88.36	88.10	86.10	89.62	84.18	84.26	92.20	68.56	41.05	19.24	38.46	80.56	88.27	67.38	73.21	76.18	83.93	62.00	67.57	26.86	66.12	65.22	76.90	64.13	69.23	25.66	36.93	27.68
11	Examples-proportional, $K = 2^{21}$	79.83	44.45	91.28	89.00	84.31	87.54	87.40	84.95	88.83	82.54	84.16	90.85	67.87	40.81	18.79	37.92	79.80	87.48	65.10	71.16	68.88	85.71	57.00	62.75	23.40	64.60	63.65	72.92	64.11	71.15	25.82	37.32	27.13
11	Temperature-scaled, $T = 2$	81.90	54.02	91.74	90.98	86.76	88.11	84.60	86.24	89.74	83.47	84.15	91.81	75.58	41.26	19.28	38.54	79.42	87.77	69.92	76.73	92.37	92.86	67.00	65.80	31.60	68.65	68.74	72.92	67.68	76.96	26.42	38.72	27.20
11	Temperature-scaled, $T = 4$	80.68	48.38	91.97	89.68	85.78	83.15	82.76	86.39	90.00	82.78	84.19	91.16	75.65	41.09	19.22	38.51	77.99	86.81	69.54	76.76	97.36	96.43	59.00	68.10	31.48	64.28	63.27	74.73	64.28	71.15	25.04	35.82	27.45
11	Temperature-scaled, $T = 8$	77.21	40.07	91.06	88.11	83.33	79.20	79.06	86.60	89.90	83.06	83.86	90.21	69.03	41.01	19.10	38.40	77.14	85.99	66.07	73.94	93.70	94.64	60.00	66.36	26.86	63.46	62.60	72.09	63.32	68.58	24.55	35.26	27.17
12	★Unsupervised pre-training + fine-tuning	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
12	Multi-task training	81.42	45.94	91.63	92.20	89.22	88.44	88.32	86.84	90.10	83.75	84.29	91.84	76.40	41.26	19.24	38.71	79.78	88.15	67.30	75.66	75.59	87.50	59.00	68.22	30.64	65.32	64.29	73.65	65.05	69.23	25.21	36.30	27.76
12	Multi-task pre-training + fine-tuning	83.11	51.42	92.66	91.75	88.73	88.06	87.70	88.61	91.61	84.09	84.31	91.85	76.53	41.15	19.12	38.59	80.26	88.50	71.03	79.54	81.69	87.50	66.00	70.72	31.48	65.94	65.03	81.23	68.18	73.08	27.08	39.80	28.07
12	Leave-one-out multi-task training	81.98	48.00	93.23	91.72	88.24	87.76	87.32	88.61	91.44	84.00	84.11	90.79	72.20	41.34	19.05	38.77	79.97	88.10	71.68	78.35	86.16	89.29	66.00	68.09	29.40	66.23	65.27	79.06	68.65	76.85	26.93	39.79	27.87
12	Supervised multi-task pre-training	79.93	38.60	92.43	91.98	88.24	87.05	86.78	88.15	91.20	82.51	83.16	90.13	76.16	41.12	18.96	38.49	77.38	85.65	65.36	75.68	68.87	83.03	58.00	64.81	21.03	63.37	54.61	71.12	67.40	76.96	26.81	40.15	28.04
13	★Baseline	83.28	83.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	26.78	69.05	68.16	76.34	68.04	78.56	26.98	39.82	27.65
13	1× size, 4× training steps	85.33	86.29	93.81	94.06	91.67	89.42	89.25	89.16	91.87	86.01	85.70	91.63	78.54	41.82	19.33	38.96	82.65	90.19	74.72	79.17	94.75	92.86	71.00	67.34	28.70	72.63	71.59	78.34	72.10	82.69	27.08	40.66	27.93
13	1× size, 4× batch size	84.60	86.08	93.12	92.31	89.22	88.85	88.84	89.35	92.07	85.98	86.13	91.07	80.14	41.70	19.42	39.08	82.82	90.21	74.64	78.78	93.69	94.64	72.00	68.09	30.98	74.73	73.90	76.53	70.06	81.73	27.07	40.60	27.84
13	2× size, 2× training steps	86.18	87.04	93.69	93.36	90.69	89.18	89.23	89.35	92.06	87.05	87.05	92.68	81.95	41.74	19.66	39.14	84.18	91.29	77.18	80.98	97.36	96.43	74.00	71.34	35.68	77.11	76.34	80.51	69.28	85.58	27.52	41.03	28.19
13	4× size, 1× training steps	86.91	87.58	94.38	92.67	89.95	89.60	89.60	89.44	92.14	87.06	87.12	93.12	83.39	41.60	19.73	39.08	83.86	91.32	78.04	81.38	89.09	94.64	73.00	73.74	40.40	78.25	77.40	81.59	70.92	84.38	27.47	40.71	28.10
13	4× ensembled	84.77	86.14	93.46	93.31	90.67	89.71	89.60	89.62	92.24	86.22	86.53	91.80	77.98	42.10	20.10	39.66	83.09	90.40	74.74	77.88	89.85	91.07	66.00	69.32	29.40	72.67	71.94	76.90	69.12	82.12	28.05	40.93	28.09
13	4× ensembled, fine-tune only	84.05	84.78	92.78	93.15	90.44	88.34	88.12	89.27	91.97	85.33	85.88	90.98	77.62	41.66	19.67	39.12	82.58	89.86	71.66	77.43	90.07	92.86	69.00	67.31	28.34	70.47	69.64	75.45	68.18	74.04	27.55	40.22	28.09

감사합니다