GPT1

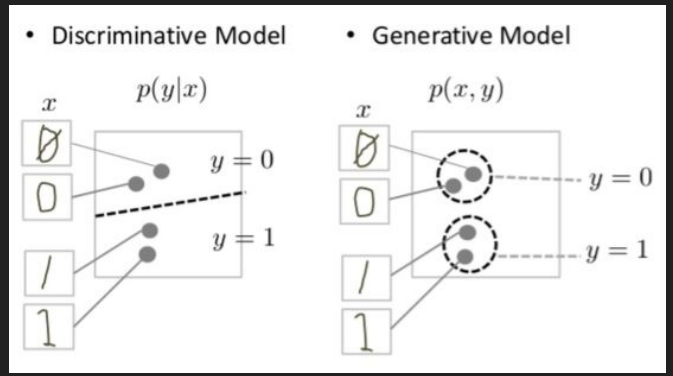
Improving Language Understanding
by Generative Pre-Training

GPT란?

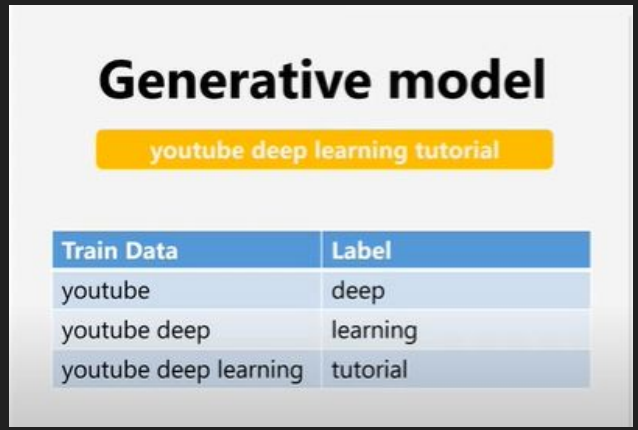
- Generative (생성하는) Pre-trained (사전 학습된) Transformer (트랜스포머)
- **G**enerative model (생성모델)

Generative models – generate new data instances.

Discriminative models – discriminate between different kinds of data instances.



0인지 1인지 구별 / 데이터 공간에서 실제와 가까운 숫자(0,1) 생성



GPT란?

- **p**re-training

semi-supervised approach

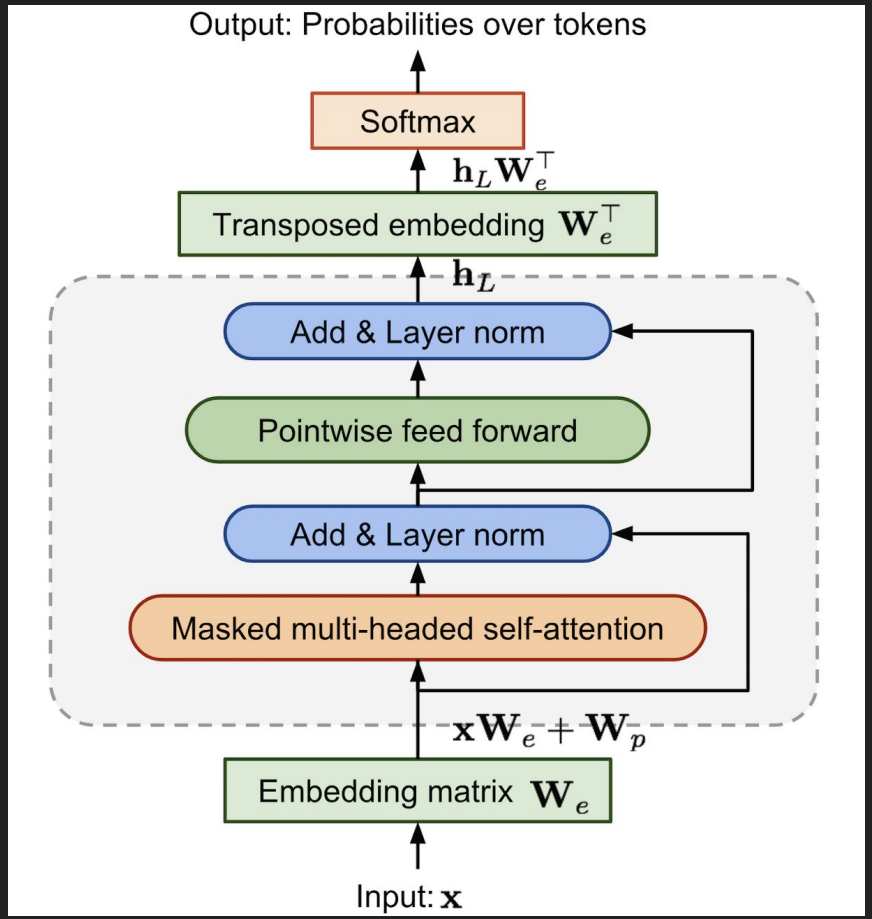
-> unsupervised pre-training + supervised fine-tuning

=> unlabeled text data의 활용 문제를 개선하기 위한

Semi-supervised language model, GPT 제안

unlabeled data 활용 문제 : 1. 어떠한 목적함수가 unlabeled text data를 학습하기 위해 가장 효과적인지 확실하게 밝았다.

2. unlabeled text data를 활용하여 사전학습을 하여도, target-task로 transfer 하기 위한 가장 효과적인 방법이 밝혀져 있지 않다.



GPT란?

- Transformer의 decoder block 구조
  - 기계번역이나 문서 생성, 구문 분석과 같은 task에서 강력한 성능
  - 최소한의 fine-tuning 구조변화로 target task에 transfer 가능
  - Encoder-Decoder Attention이 제거된 Transformer Decoder의 variation 구조

Framework - 1: Unsupervised pre-training

learning a high-capacity language model on a large corpus of text

- unlabeled token U = {u1, . . . , un} 을 통해 일반적인 언어모델의 목적함수 Likelihood L1(U) 를 최대화 하는 과정

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

k는 context window의 크기이며, 조건부 확률 P는 Θ를 parameter로 갖는 신경망으로 모델링

Framework - 1: Unsupervised pre-training

Transformer의 decoder block

1) $h_0 = UW_e + W_p$ - Masked self-Attention을 위한 input h0

input token Matrix U * embedding Matrix We + Positional embedding Wp

2) $h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$

12개의 decoder block을 stack, h1은 이전 decoder block의 hidden state h0을 입력으로 받아 계산

3) $P(u) = \texttt{softmax}(h_n W_e^T)$ - output probability P(u)

n번째 decoder block의 hidden state output hn에 transposed embedding Matrix WeT를 곱하여 softmax 함수를 적용

Framework - 2 : supervised fine-tuning

- target task에 맞게 parameter를 조정하는 단계
- input token sequence {x1 , . . . , xm}과 label y로 구성된 target task의 labeled dataset C 통해 학습 진행

1) C의 input token에 대한 GPT의 마지막 decoder block hidden state h 얻기 위해 pretrained model에 input token 통과
2) $P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y)$ - softmax probability

파라미터 Wy를 갖는 하나의 linear layer에 h를 통과시켜 softmax probability 계산

-> token probability distribution

Framework - 2 : supervised fine-tuning

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

label y에 대해 지도학습 진행

지도학습의 목적함수L2(C) 또한 일련의 구조로 모델링된 조건부확률 P를 통해 계산

4) unsupervised pre-training의 목적함수 L1 추가 (auxiliary objective)

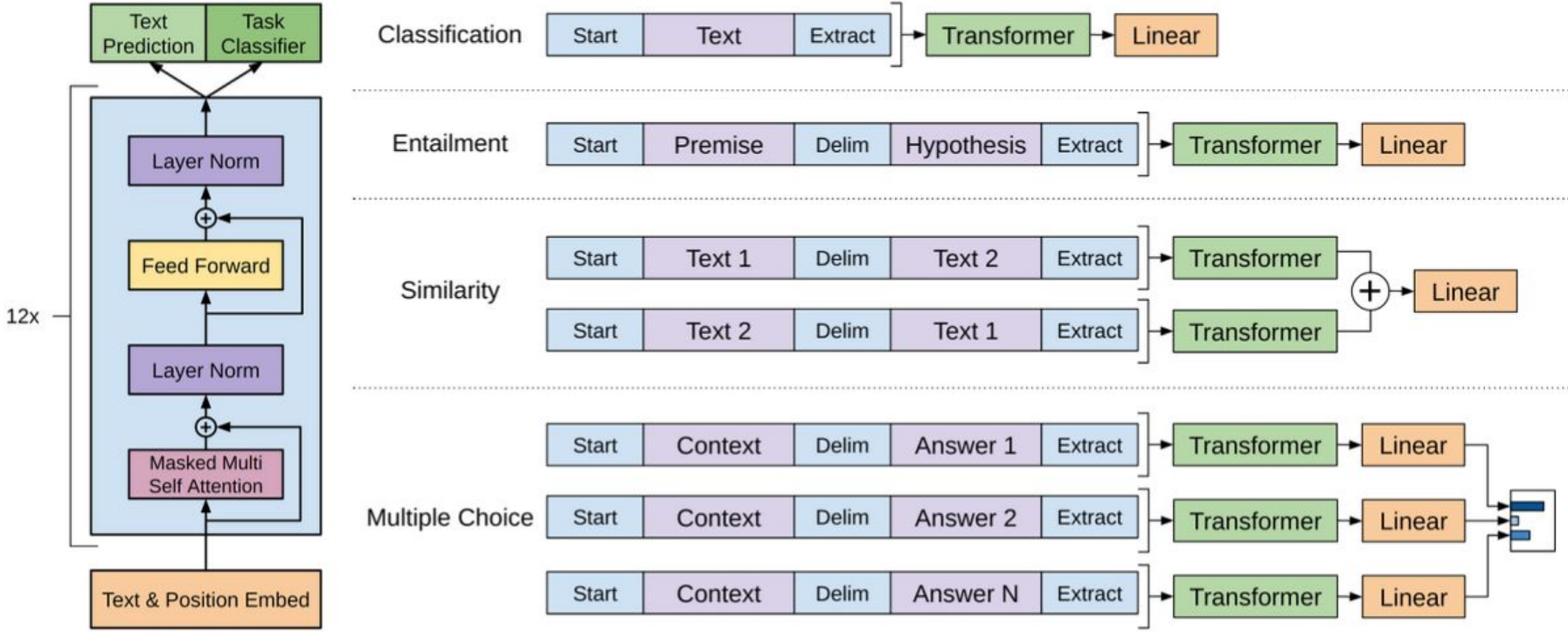
Labeled dataset C에 대해 L1(C)로 계산

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

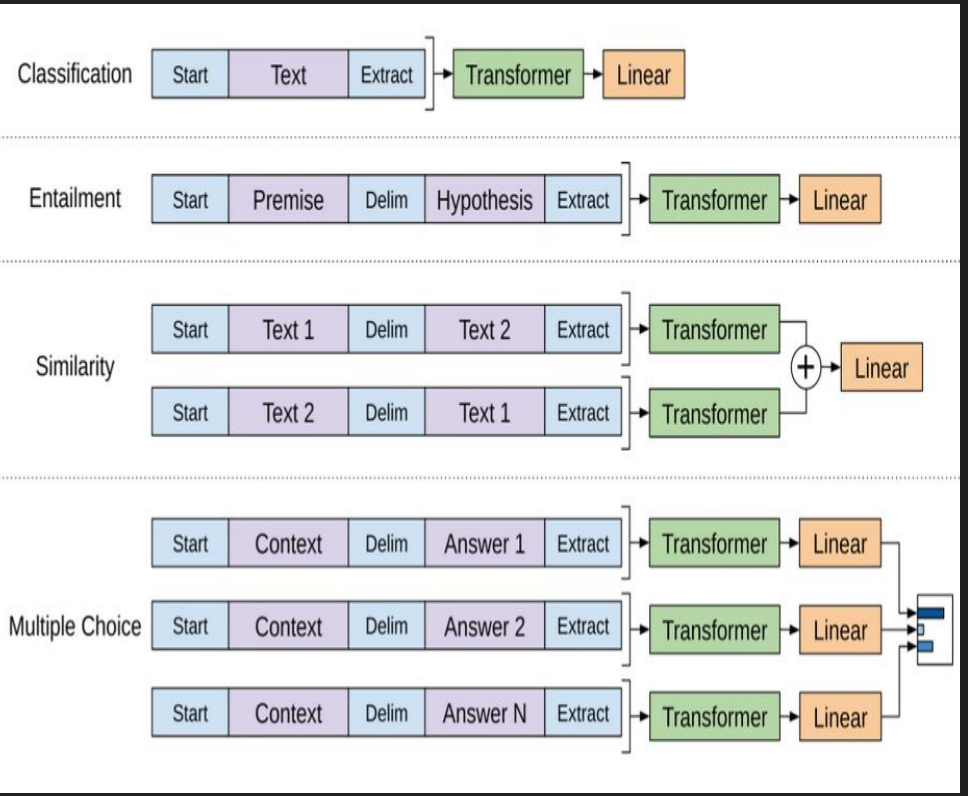
λ는 L1(C)의 반영 정도 정하기 위한 weight hyper parameter

-> fine tuning 과정에서 추가로 필요한 parameters는 Wy와 delimiter토큰을 위한 임베딩뿐

Framework - 3 : Task-specific input transformations



Framework - 3 : Task-specific input transformations



- **Entailment** - 전제 Premise를 통해 가설 Hypothesis의 참, 거짓을 밝히는 task

-> 전제 p 와 가설 h 토큰 시퀀스를 delimiter token으로 concat

- **Similarity** - 문장 유사도 측정 task

-> 가능한 문장 순서를 모두 포함하도록 입력 시퀀스를 delimiter token과 함께 수정하고 두 문장의 h를 각각 생성한 후 linear layer에 들어가기 전 더함

- **QA, Multiple choice**

-> [z; q; $; ak] (z: context document, q: question, ak: 가능한 답변 set)

delimiter token을 사용해 각 z,q와 답변을 concat

각 시퀀스 별로 독립적으로 처리 후 가능한 답변의 분포 생성 위해 soft layer에 전달

Experiments - setup

- **Unsupervised pre-training** - BooksCorpus dataset
  : 다양한 장르의 출간되지 않은 책 7000권 이상 포함

  -> allows the generative model to learn to condition on long-range information

- **Model specifications** - 12-layer decoder-only transformer with masked self-attention heads (768 dimensional states and 12 attention heads)
  - BPE(Byte pair encoding) :서브워드 분리(subword segmentation)
- **fine tuning** - pre train에 사용한 하이퍼파라미터 그대로 사용, dropout 추가
  - dropout: Masked Attention Layer 에 들어갈 Input 을 만들기 위해 0.1의 확률로 Input tensor의 각 값이 0으로 바뀜

- **Natural Language Inference** (Entailment - 문장 관계 수반/모순/중립 파악)

정부리포트(MNLI) 이미지캡션(SNLI) Science exams (SciTail) 위키피디아기사 (QNLI) 뉴스기사 (RTE)

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

-> RTE제외한 데이터 셋에서 모두 다른 모델보다 우수한 성능

RTE dataset은 크기가 작은 데이터셋, 상대적으로 데이터셋이 클수록 좋은 성능

- **Question answering and commonsense reasoning**

RACE (영어로 구성된 중/고등학교 시험 문제) Story Cloze(네 문장의 스토리의 올바른 결말 선택)

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**

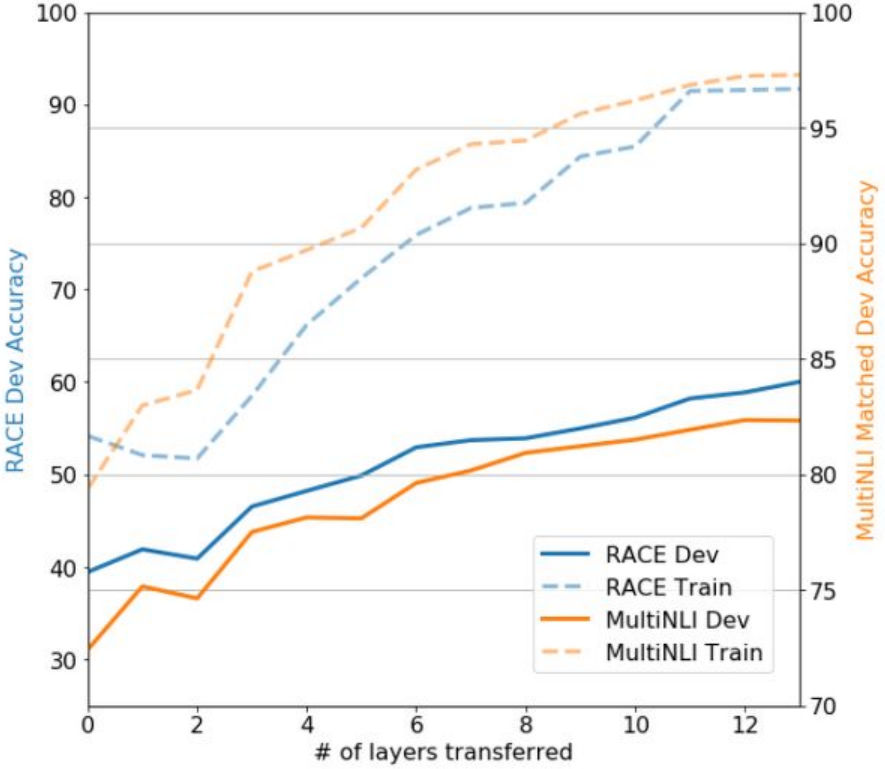
-> 모두 우수한 성능

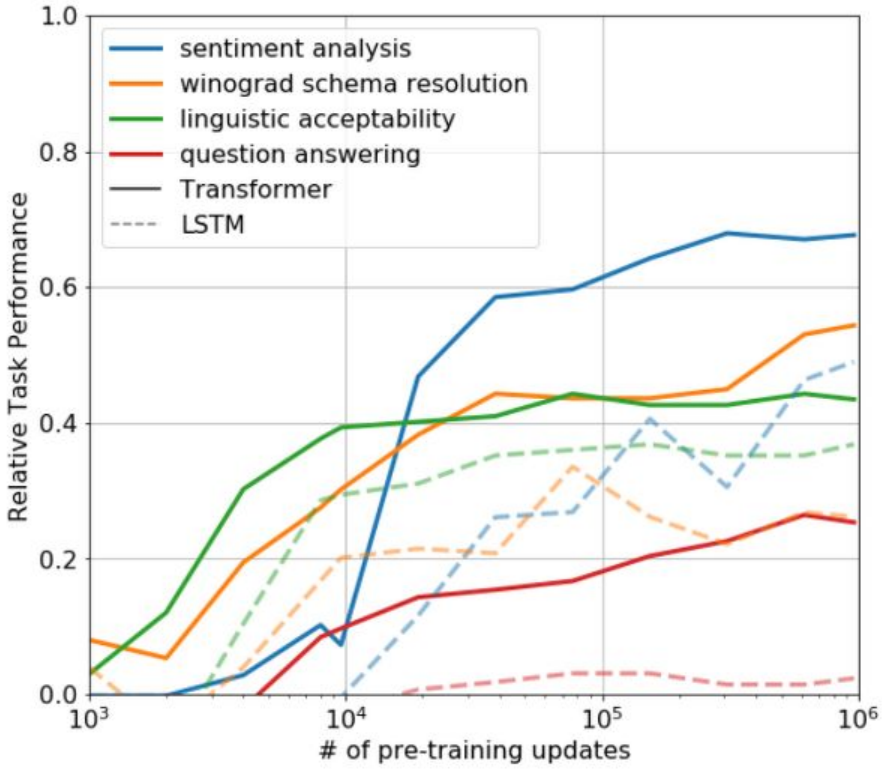
long range context 잘 처리

- **Classification과 Similarity task**

CoLA (문법적으로 올바른지) SST2(영화 코멘트 긍/부정)

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**





layer 수에 따른 유의미한 성능 향상 / 각각 task의 Zero-shot 성능 비교(점선 LSTM)

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

1)보조 목적함수 aux LM- NLI task와 QQP에서 도움을 준다는 것을 확인 (1,3)

전반적으로 큰 데이터셋은 보조 목적함수로부터 도움을 받는 반면, 작은 데이터셋에서는 도움을 받지 못함

2) LSTM과 Transformer 비교 (1,4)

LSTM - 평균 약 5.6점 낮음, 오직 MRPC데이터셋에서만 Transformer보다 더 나은 성능

3)사전학습 효과 (1,2) - w/o pre training 모두 성능 저하