Improving Language Understanding by Generative Pre-Training

19기 시각화 이세연
19기 분석 신은빈

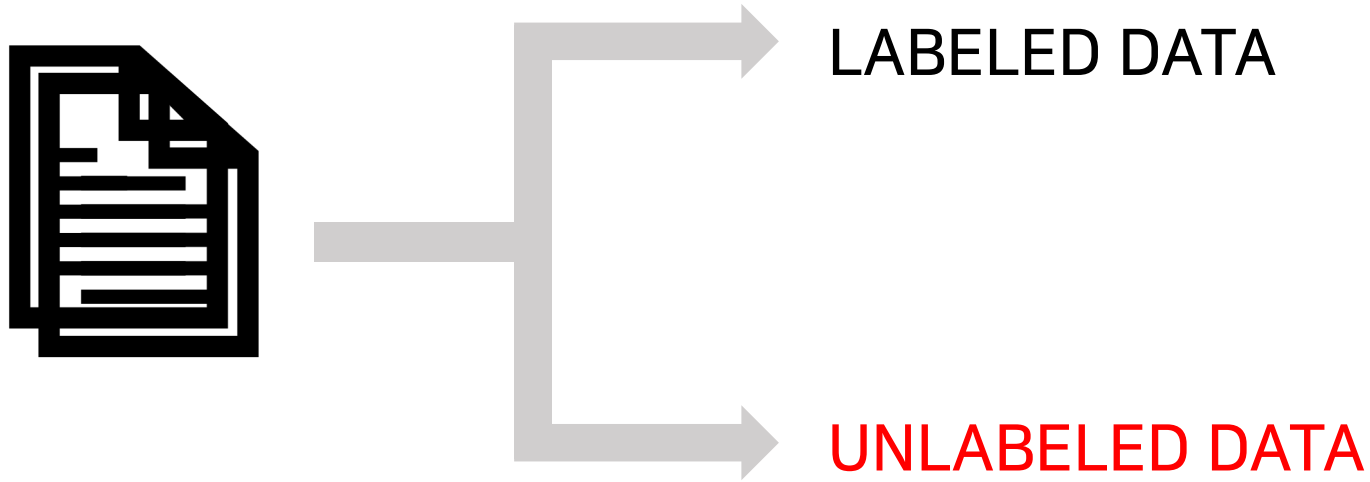
CONTENTS

01

INTRODUCTION

01. INTRODUCTION

Text



01. INTRODUCTION

Problems with handling unlabeled text

1. 어떤 목적함수가 전이에 유용한 text representation을 배우는 데 효과적인지 불분명하다.

   It is unclear what type of optimization objectives are most effective at learning text representations that are useful for transfer.

2. 학습된 representation을 target task에 전이하는 가장 효과적인 방법에 대한 의견 일치가 없는 상태이다.

   There is no consensus on the most effective way to transfer these learned representations to the target task.

01. INTRODUCTION

GPT's approach to the problems

Semi-supervised

Unsupervised Pre-training

*We use a **language modeling objective on the unlabeled data** to learn the initial parameters of a neural network model.*

Supervised Fine-tuning

*We adapt these parameters to a **target task** using the corresponding **supervised objection.***

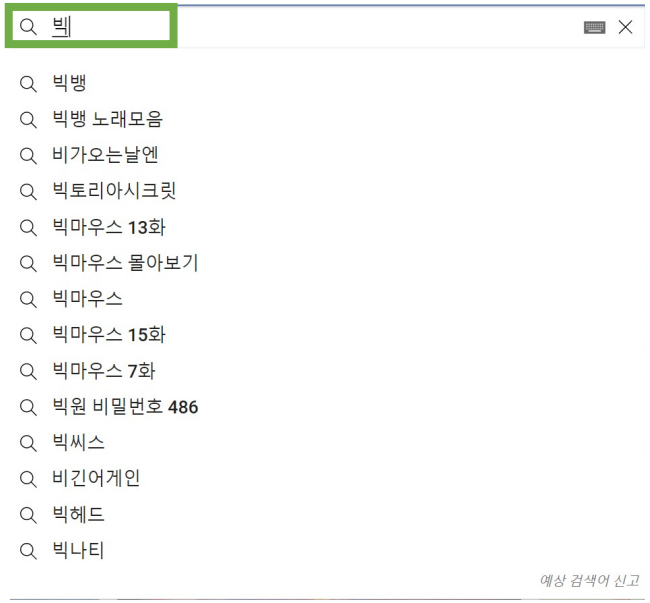
01. INTRODUCTION

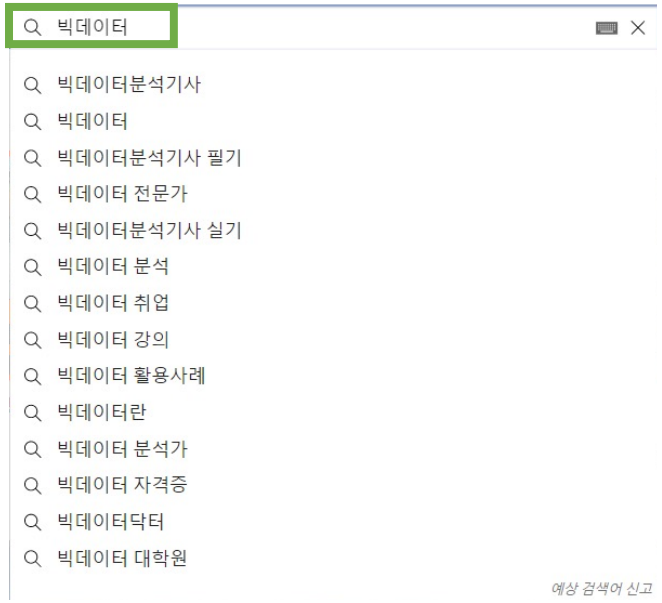
Language Modeling

- 현재 갖고 있는 단어들로 다음 단어를 예측
- 별도의 레이블링된 데이터가 필요하지 않음

01. INTRODUCTION

Language Modeling





- *오류율이 적어짐*
- *숨겨진 패턴을 찾을 수 있음*

01. INTRODUCTION

GPT's approach to the problems

Semi-supervised

TRANSFORMER

Unsupervised
Pre-training

*We use a **language modeling objective on the unlabeled data** to learn the initial parameters of a neural network model.*

Supervised
Fine-tuning

*We adapt these parameters to a **target task** using the corresponding **supervised objection.***

02

Transformer

02. TRANSFORMER

Why Transformer?

Transformer

◈ ***Perform strongly on various tasks*** *such as machine translation, document generation, and syntactic parsing.*

◈ *This model choice provides us with a* ***more structured memory for handling long-term dependencies in text****, compared to alternatives like current networks, resulting in robust transfer performance across diverse tasks.*

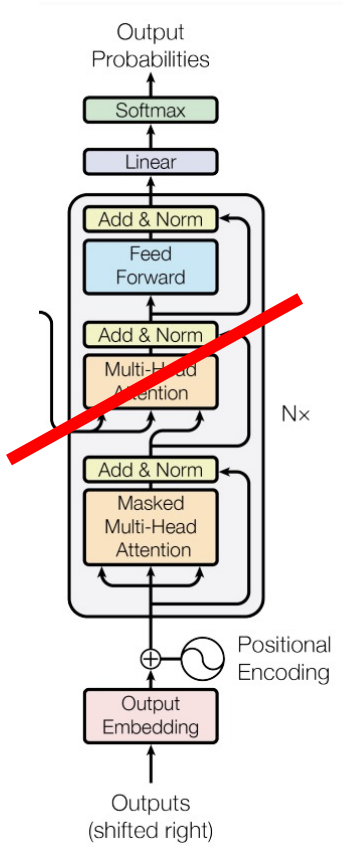
◈ *During transfer, we* ***utilize task-specific input adaptations*** *derived from* ***traversal-style approaches****, which process structured text input as a single contiguous sequence of tokens.*
→ these adaptations enable us to ***fine-tune effectively with minimal changes to the architecture of the pre-trained model.***

02. TRANSFORMER

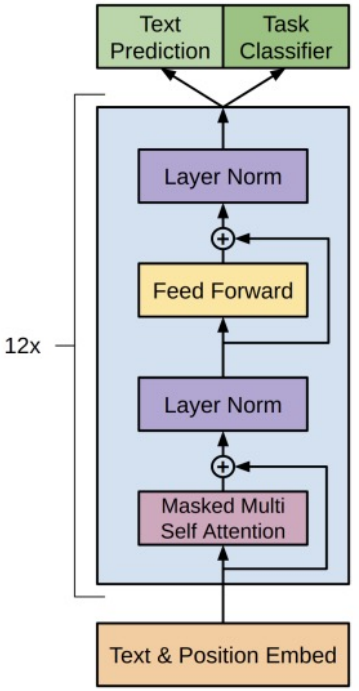
Details

Transformer

~~1. Encoder Self-Attention~~
2. Decoder Masked Self-Attention
~~3. Encoder-Decoder Attention~~



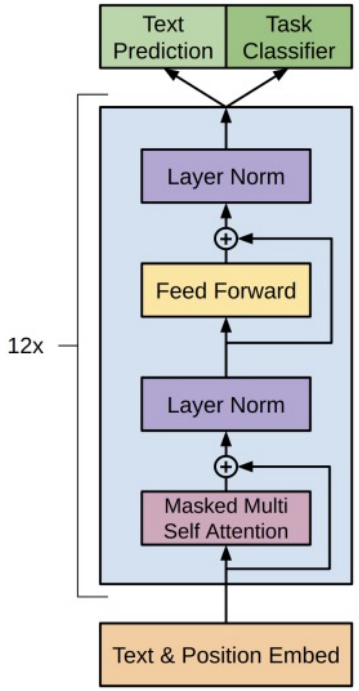
Transformer의 decoder



GPT의 transformer

02. TRANSFORMER

Details



◈ Positional Encoding

- 단어 입력을 순차적으로 받는 방식 X
- 단어의 위치/순서 정보는 필요함
- 각 단어의 임베딩 벡터에 위치 정보들을 더하여 모델의 입력으로 사용

◈ Attention

◈ Self Attention

◈ Multi

◈ Masked

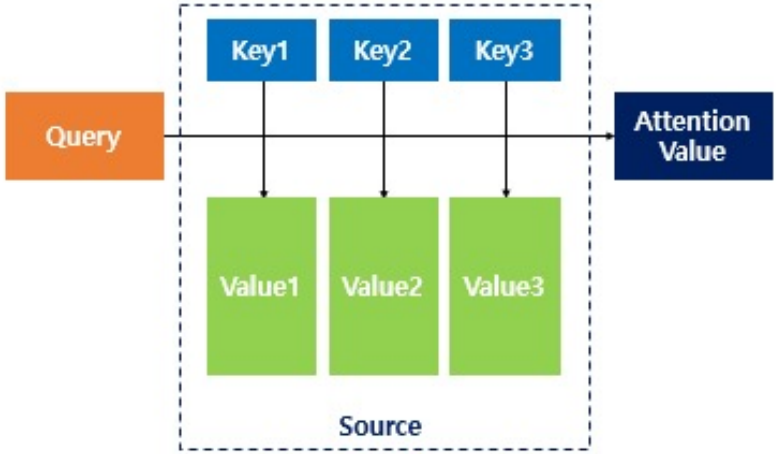
02. TRANSFORMER

Attention

◈ Attention

- * 디코더에서 출력 단어를 예측하는 매시점마다,
  인코더에서 해당 시점에서 예측해야 할 단어와
  연관이 있는 입력 단어 부분을 집중(attention)해서 봄.

- 연관성이 높은 단어들끼리 연결해 줌.



Q = Query : t 시점의 디코더 셀에서의 은닉 상태
K = Key : 모든 시점의 인코더 셀의 은닉 상태들
V = Value : 모든 시점의 인코더 셀의 은닉 상태들

- * **Query**에 대해서 모든 **key**와의 유사도를 각각 구함
  → 유사도를 가중치로 하여 key와 매핑되어 있는 각각의 **value**에 반영함
  → value를 모두 가중합
  → attention 값

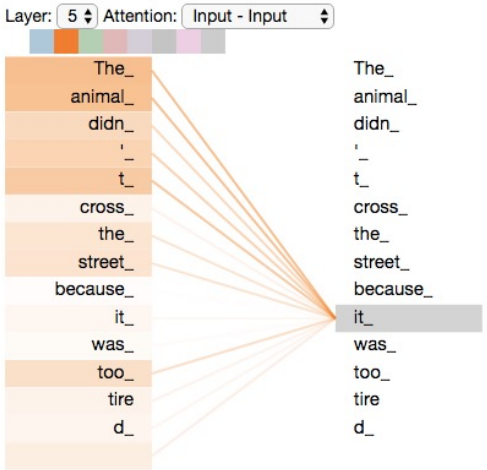
02. TRANSFORMER

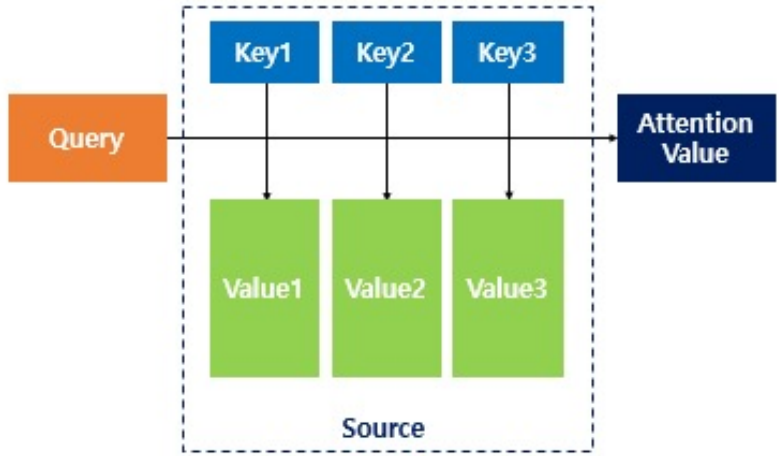
Self Attention

◈ Self Attention

- 입력 문장 내의 단어들끼리의 유사도를 구함

- The animal didn't cross the street because it was too tired.





Q = Query : 입력 문장의 모든 단어 벡터들

K = Key : 입력 문장의 모든 단어 벡터들

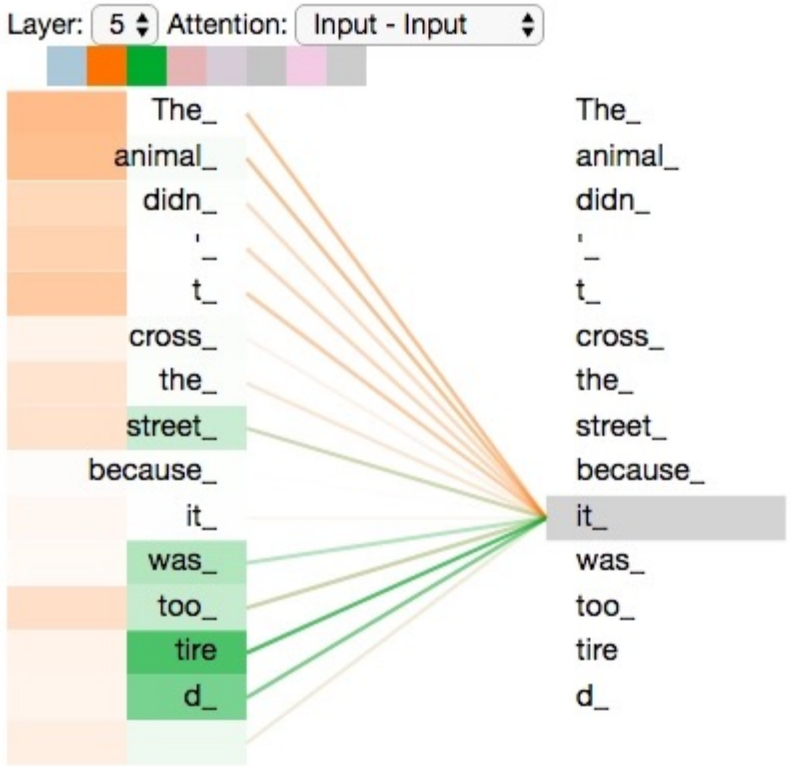
V = Value : 입력 문장의 모든 단어 벡터들

02. TRANSFORMER

Multi

◈ Multi

- 한 번의 attention을 수행하는 것보다 여러 개의 attention을 병렬로 사용하는 것이 더 효과적



02. TRANSFORMER

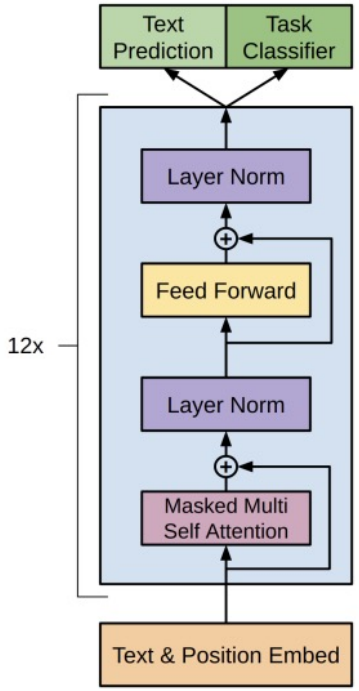
Masked

◈ Masked

- 트랜스포머: 문장 행렬로 입력을 한 번에 받음
- 어떤 시점의 단어를 예측하려고 할 때, 해당 시점 이후(미래)의 단어까지도 참고하게 되는 상황 발생
- 현재 시점의 예측에서, 현재 시점보다 미래에 있는 단어들을 참고하지 못하도록 하는 방법

02. TRANSFORMER

Evaluation



4 language understanding tasks

- Natural Language Inference
- Question Answering
- Semantic Similarity
- Text Classifications

03

Framework

03. Framework

Model 학습의 2단계

Unsupervised Pre-training

labeling되지 않은 대량 데이터를
이용하여 범용적인 언어모델을 학습

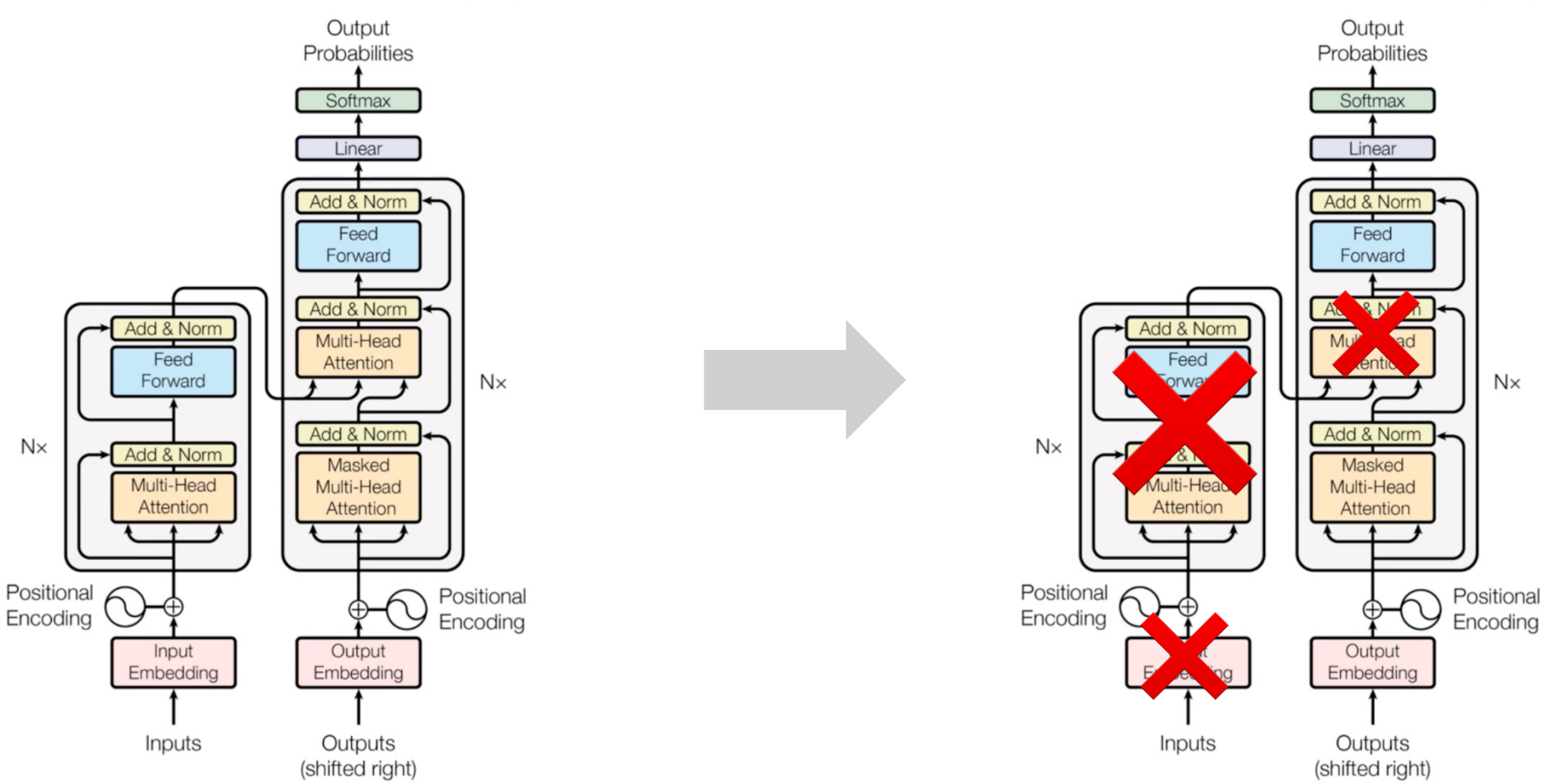
Supervised Fine-tuning

labeling 된 데이터를 이용하여
개별적인 task에 모델을 적용하는 절차

03. Framework

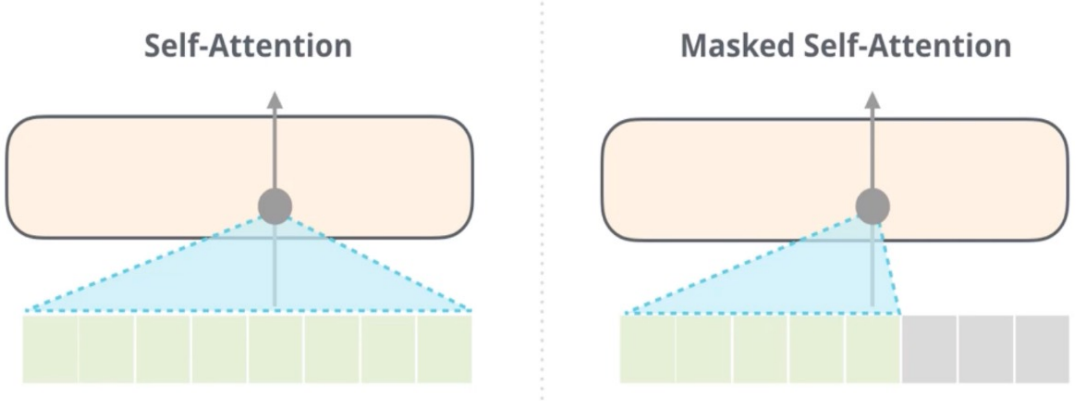
1. Unsupervised pre-training

transformer decoder 부분만 가져와 decoder 부분을 여러 겹 쌓은 모델 → **masked self-attention layer**



03. Framework

1. Unsupervised pre-training



◈ Masked self-attention

- Self attention과의 차이점 : 자신이 processing 하고자 하는 토큰 다음 시퀀스는 사용하지 않음

03. Framework

1. Unsupervised pre-training

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

◈ Masked self-attention

- $U = (u_{-k} \ \ldots \ u_{-1})$: 토큰의 context vector
- W_e = token embedding matrix
- W_p = position embedding matrix

- l번째 hidden state(h_l)는 $l-1$번째 hidden state(h_{l-1})를 입력으로 받아서 transformer decoder 블럭에 넣음
- n번 반복 후 softmax함수를 이용하여 다음 단어 예측

03. Framework

1. Unsupervised pre-training

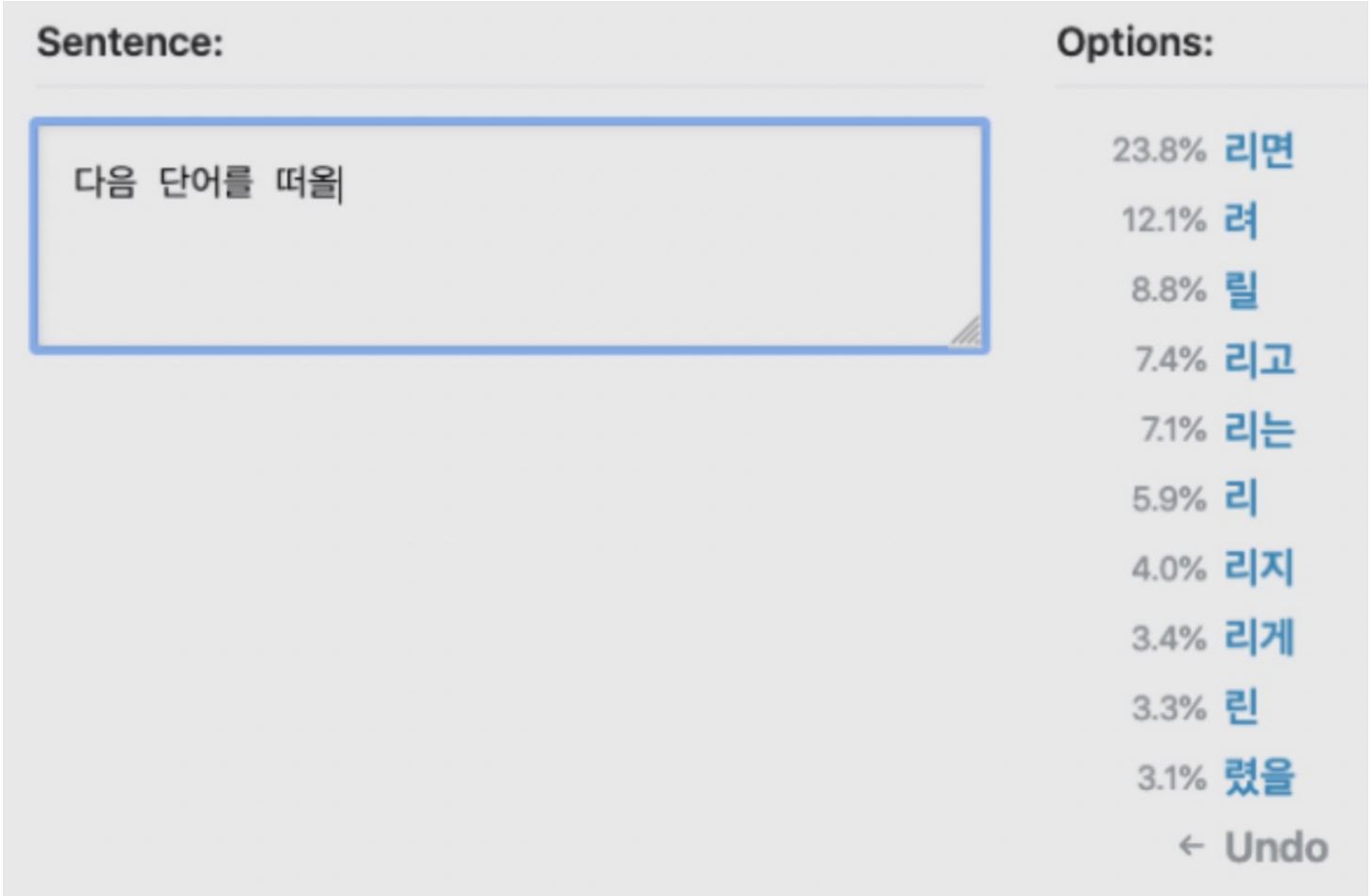
$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

- 목적함수 $L_1(u)$: i부터 ~ $i-1$번째까지 살펴 본 후
  이를 바탕으로 i번째에 해당하는
  토큰이 무엇인지에 대한 likelihood 최대화

- Θ = neural network parameter
- k = context window size

03. Framework

1. Unsupervised pre-training



03. Framework

2. Supervised fine-tuning

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

◈ Supervised fine tuning

- y값, label 존재
- $x^1, \ldots x^m$: input token sequence
- h_l^m : labeled dataset C의 input token을 pretrained model에 통과시켜 얻은 값
- W_y : parameter

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

- C : labeled dataset
- 목적함수 $L_2(C)$: $x^1 \ldots x^m$ 총 m개의 token이 주어지고 이때 정답이 무엇인지에 대한 확률 값 최대화

03. Framework

2. Supervised fine-tuning

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

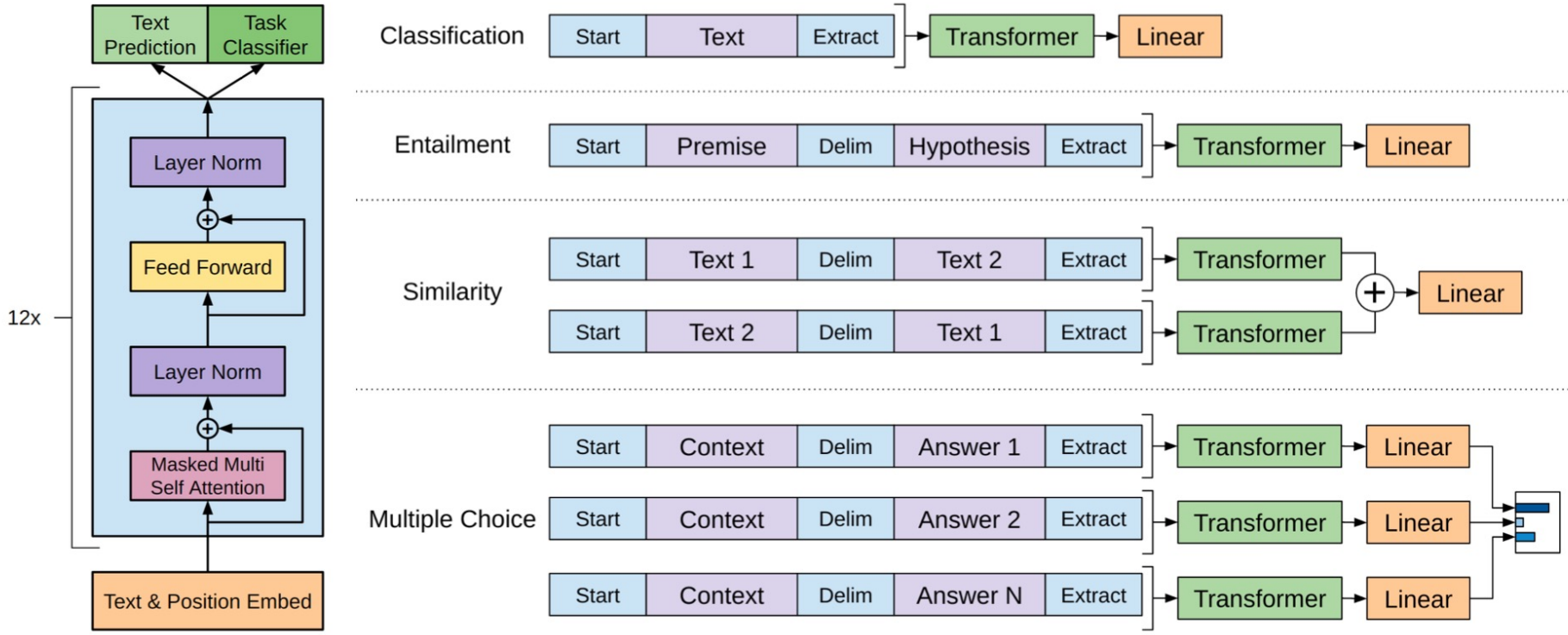
- $L_3(C)$: pre-training 후 label된 데이터를 가지고 **unsupervised할 때의 목적함수** $(L_1(u))$와 **supervised할 때의 목적함수**$(L_2(C))$를 같이 사용

→ supervised에 대한 일반화가 잘됨

→ 학습 속도가 빠름

03. Framework

2. Supervised fine-tuning



04

Result

04. Result

Experiments

Natural Language Inference

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

Question & Answering

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**

04. Result

Experiments

Classification & Sematic Similarity

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**

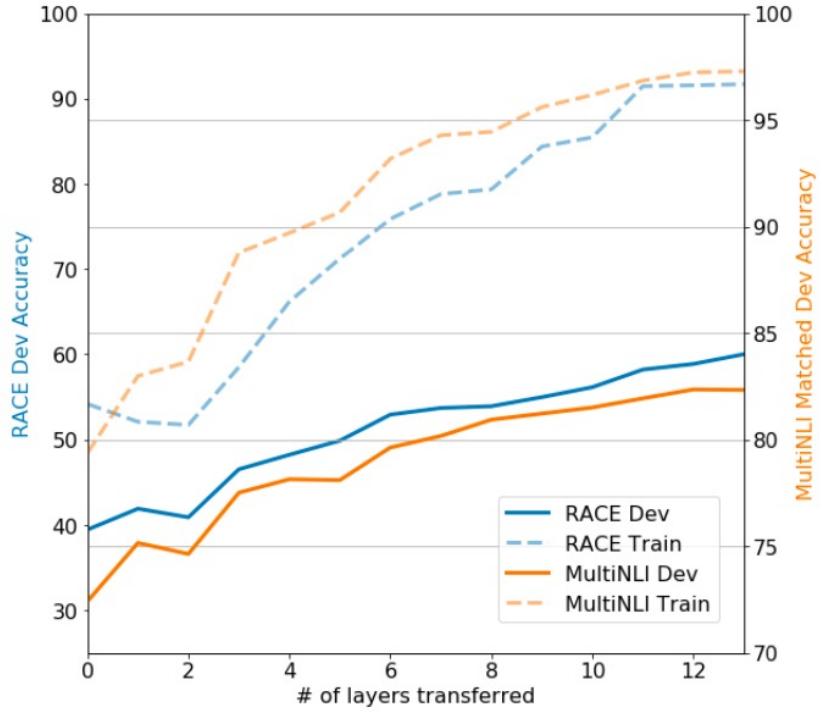
04. Result

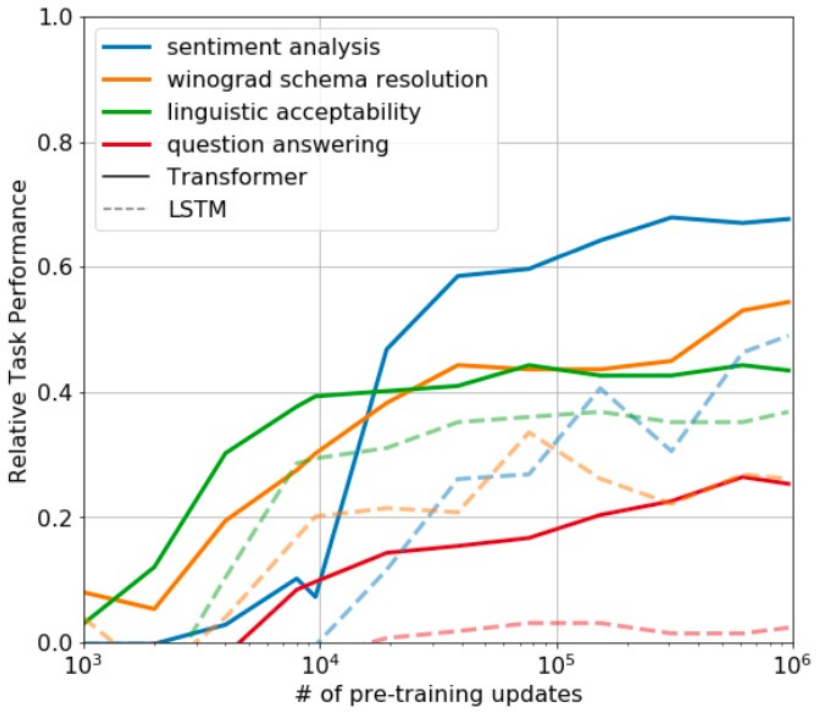
Analysis

◈ Decoding layers 쌓은 개수

→ **많을수록** 성능 높아짐

◈ Zero-shot learning에 비해 **fine tuning**이 성능이 더 높음





Thank you

19기 시각화 이세연
19기 분석 신은빈