[논문리뷰] Improving Language Understanding by Generative Pre-Training

250910 BOAZ WEEK8 과제

🌟 등장 배경

- 준지도 학습 : unlabelled 데이터가 많은 상황에서 사용하는 방법
  - 한계
    1. 어떤 학습 목표가 좋은 지가 없었고,
    2. 학습된 표현을 다른 작업에 전이하는 방법도 확실치 않았다.

    → 작업마다 방법을 새로 설계, 범용성 ↓

⇒ 1️⃣ **비지도 사전학습 (언어모델링) +** 2️⃣ **소량의 지도 데이터로 Fine-tuning** 을 조합하자!

✏️ Framework

1️⃣ Unsupervised pre-training

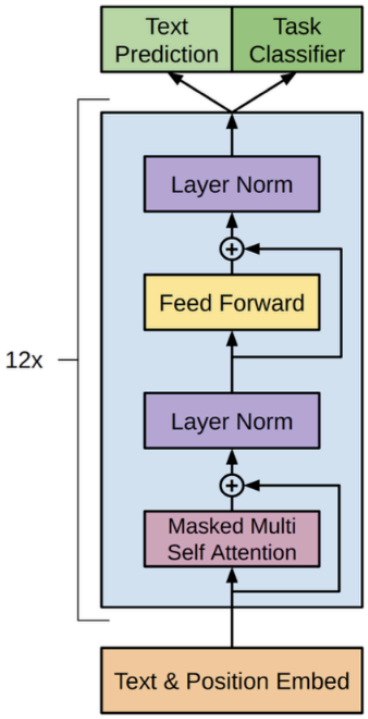
특정 태스크를 목표로 학습 하기 위한 labelled 데이터 부족 → 처음에는 비지도 학습을 통해 GPT를 사전 훈련!

- 여느 LLM과 동일하게 **"특정한 단어가 만약 i번째라면 i-1 부터 i-k번째까지의 단어를 보고, i번째가 나올 가능성을 최대화하는 방법"**
- 우도 최대화를 loss function 으로 한다.

maximize the following likelihood:

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

- 구조 = Transformer Decoder
  (12-layer, hidden size 768, heads 12, 총 1.17억 파라미터)



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

1. input sequence 를 받아 word embedding , Positional embedding 수행
    - 이때 h_0 가 위의 식으로 표현됨
2. 그 다음부터 hidden state 를 decoder block 에 계속 넣어 학습시킴
    - (논문에서는 decoder block = 12개)
3. 최종 hidden state 값을 활용하여 확률값 출력

❓ 왜 decoder를 사용했을까?

- GPT는 자연어 생성이나 자연어 모델링에 초점이 맞춰져 있어,
  seq2seq2 구조에서, **문맥벡터를 사용해 다른 시퀀스를 생성하는 decoder만을 사용**
- Encoder는 단어의 vector로 출력되는 반면, **Decoder는 확률값으로 표현되기 때문에 어떤 token이 나와야 할 지 예측할 수 있다.**

2️⃣ Supervised Fine-tuning

사전학습 된 모델에 labeled data를 가지고 target task에 맞게 모델을 fine-tuning 하자!

- W_y (linear output layer)를 추가해서 라벨에 대한 예측값을 만드는 방식

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

objective to maximize:

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

- 추가적으로 논문의 저자들은, L2 function 에 L1 function을 더하는 auxiliary objective를 추가하여 성능을 높임(L3).

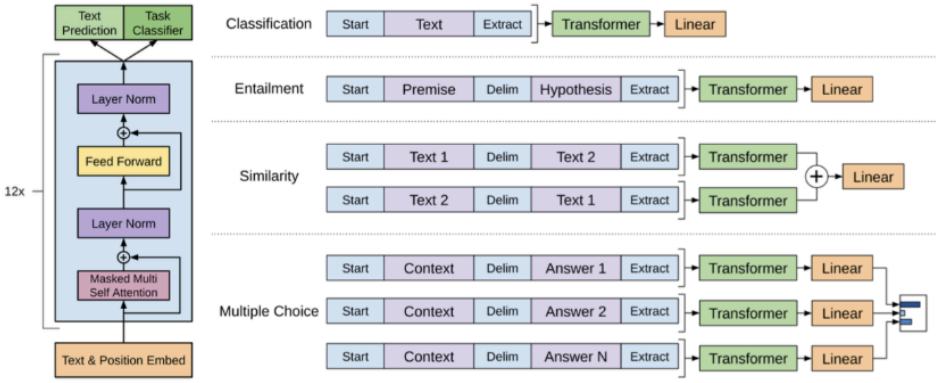


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

- 구조는 바꾸지 않고, 마지막 층에 태스크에 맞는 head 추가

⇒ 새로운 architecture를 추가하던 기존 연구들과 다르게,
GPT-1은 traversal-style을 가진 구조화된 input으로 전이 학습을 최소한의 변화를 통해 가능

🧪 Experiments

- Dataset

Table 1: A list of the different tasks and datasets used in our experiments.

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

- TASK (총 12개)

1. Classification : 텍스트 입력을 Transformer로 인코딩 후 선형결합하여 클래스 라벨을 예측하는 과제
2. Natural Language Inference : 전제(premise)가 가설(hypothesis)을 의미적으로 함의하는지 여부(Yes/No)를 판별하는 과제
3. Semantic Similarity : 두 문장의 의미적 유사도를 0~1 실수 값으로 추정
4. Question answering and commonsense reasoning : 주어진 문맥에 대해 여러 선택지 중 가장 적합한 답변을 확률 기반으로 선택하는 과제

- 12개의 task 중에 9개의 Dataset에서 SOTA 달성

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

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**

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**