OpenAI – GPT 1
Improving Language Understanding by Generative Pre-Training

Alec Radford et al, 2018.07

Feb.4.2021
신한이

0. Abstract

- NLP
  Labeling되지 않은 원문 text data에 대해 학습한다.

- Generative pre-training & Fine-tuning
  Language model로 Pre-training 을 진행,
  필요한 task에 대해 fine-tunning 하는 것이 효과적임.

- 12개 중 9개의 task에서 State-of-the-art(SOTA) 달성

1. Introduction

- '원본'의 텍스트에서 효과적으로 학습하는 능력. => 지도학습에 대한 의존성을 낮춤
- 문제점:
  1. 어떤 최적화 목적 함수가 더 효과적인지 모른다. 과제마다 적절한 방법이 다름.
  2. 특정 과제에 있어서 가장 효과적인 방법에 대한 일치된 의견이 없다.

- 이 논문에서는 비지도 사전학습(Unsupervised pre-training)과 지도 미세조정(Supervised fine-tuning)의 조합을 사용
- 학습의 단계:
  1. 신경망 모델의 초기 parameter를 학습하기 위해 언어모델링 목적함수를 사용
  2. 이 parameter를 연관된 지도 목적함수를 사용하여 목표 과제에 적용

- 모델 구성은 Transformer 사용.
- RNN에 비해 장거리 의존성 부분에서 뛰어나다.

2. Related Work

Semi-supervised learning for NLP

-> 단어, 구, 문장 수준의 embedding은 미분류 데이터로부터 학습될 수 있고, 다양한 문제에서 텍스트를 적절한 벡터표현으로 변환할 수 있다.

Unsupervised pre-training

-> 다른 접근법은 상당한 양의 parameter를 추가했지만, GPT는 그렇지 않다.

Auxiliary training objectives

-> 보조 목적 함수 추가

3. Framework

3.1. Unsupervised pre-training

학습은 다음의 두 단계로 진행

1. **Unsupervised pre-training**
   : 큰 말뭉치에서 대용량의 언어 모델을 학습

2. **Supervised finetuning**
   : 분류 데이터를 써서 특정 과제에 맞추어 모델을 미세조정

3. Framework

3.1. Unsupervised pre-training

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

$$h_0 = UW_e + W_p$$

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

$$P(u) = \text{softmax}(h_n W_e^T)$$

⇒ 목적함수
(Language Modeling Objective)

- transformer 활용
- Decoder 12개로만 이루어진 모델
- Target 토큰에 대한 분포를 얻음

We: 토큰 임베딩
Wp: positional 임베딩
U: token context 벡터

3. Framework

3.2. Supervised fine-tuning

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

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

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

⇒ 목적함수
(Language Modeling Objective)

- X1부터~xm을 입력 토큰으로 받음
- P를 입력으로 하는 linear layer를 추가.
- Objective를 최적화.

3. Framework

3.3. Task-specific input transformations

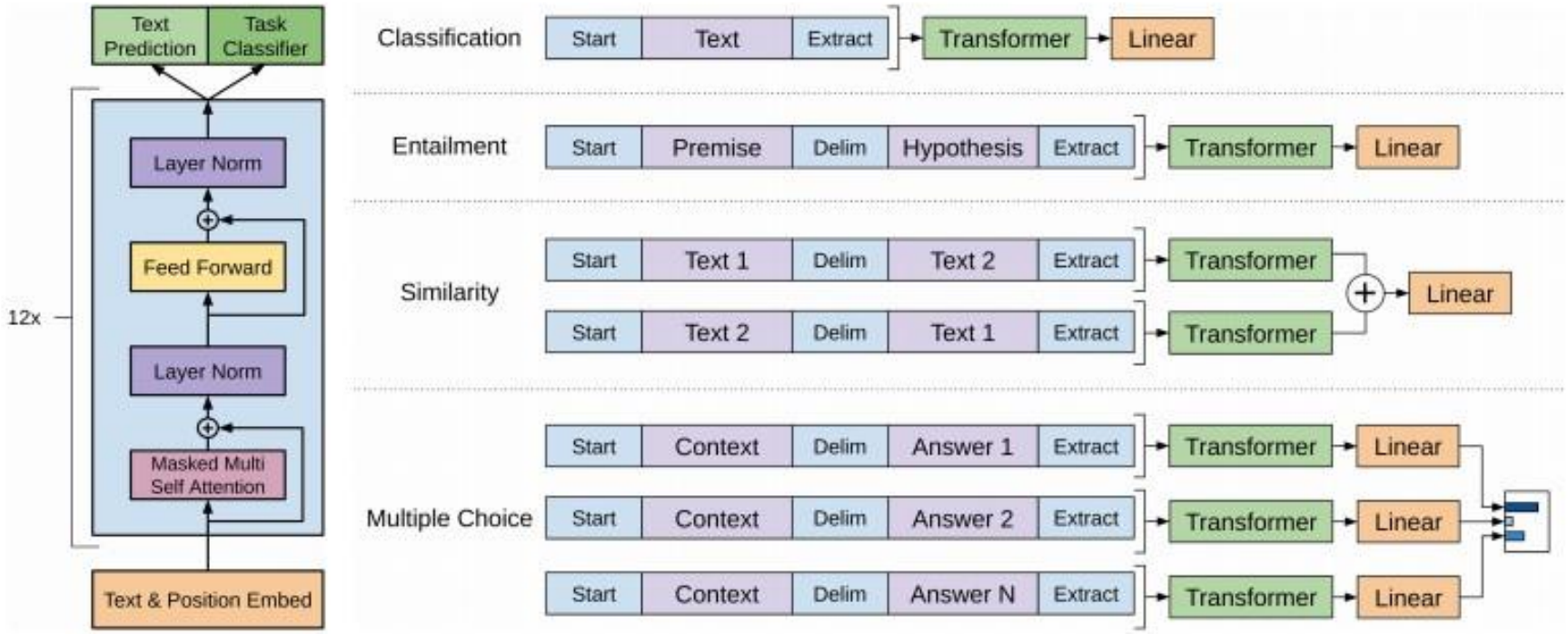


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

4. Experiments

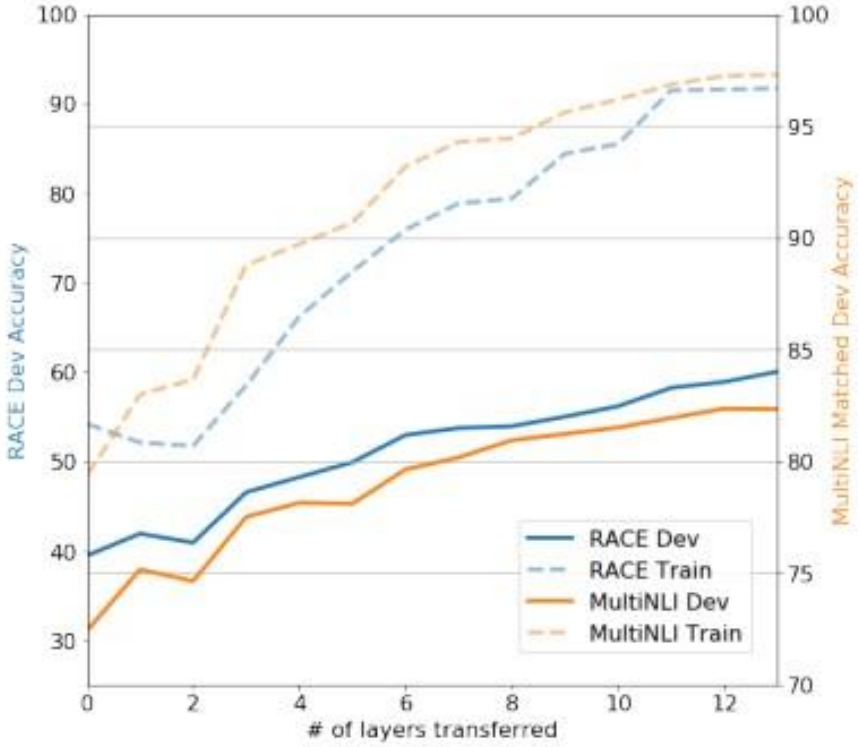
Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned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

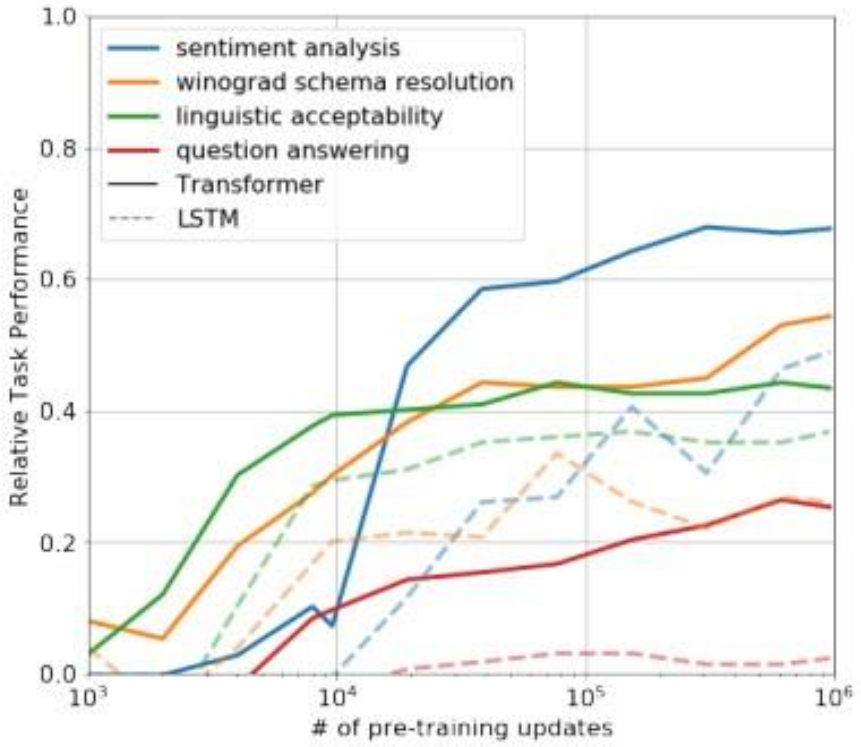
4. Experiments

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**

5. Analysis



-> 12 Layer까지, layer가 증가함에 따라 정확도가 높아진다.



-> transformer 모델이 모든 task에서 LSTM보다 높은 수치를 보였다.

6. Conclusion

- NLP 이해 능력이 뛰어난 단일 모델을 소개.
- Transformer를 Pre-trained Language Model 생성에 활용하는 좋은 예시.
- 당시 BERT와 비교

  -> pre-training 방법이 다르다.
  -> BERT가 다양한 task에 더 뛰어난 결과를 보임.
  -> GPT모델은 language model이 기반 => BERT보다 언어 생성에 유리.