0928 GPT 논문발표

+논문 (Improving Language Understanding by Generative Pre-Training)

https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

---

Improving Language Understanding by Generative Pre-Training

---

0. Abstract

unlabel된 text는 많지만, 특정한 task에 label된 data는 부족하다.

generative pre-training + discriminative fine-tuning 하여 높은 성과

1. Introduction

unlabel된 data 활용의 한계점

1. It is unclear what type of optimization objectives are most effective at learning text representations that are useful for transfer.(어떠한 목적함수가 효과적인지 알 수 없다.)
2. There is no consensus on the most effective way to transfer these learned representations to the target task. (각각의 task에 대해서 어느 전이학습을 해야 효과적인지 알 수 없다.)

→ semi-supervised approach 제안 (unsupervised pre-training + supervised fine-tuning)

2. Related Work

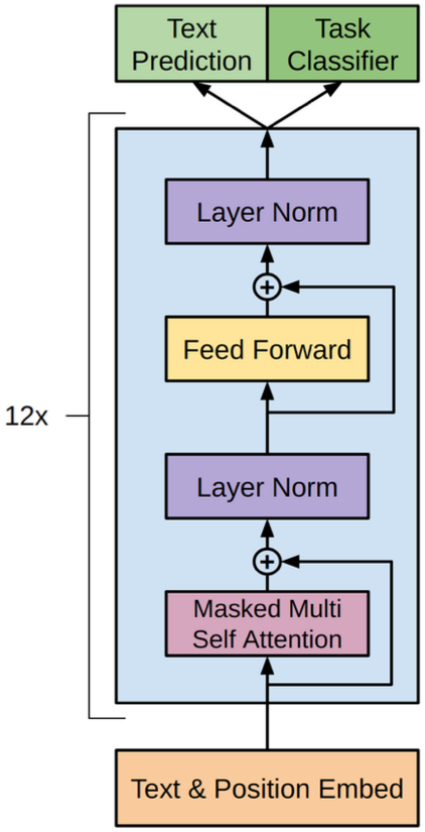
1. Semi-supervised learning for NLP
2. Unsupervised pre-training
3. Auxiliary training objectives

3. Framework

- 3.1 Unsupervised pre-training

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

본 목적함수를 최대화 하는 식을 활용함 (i번째 토큰이 i-1개의 토큰을 기반으로 발생할 확률)



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

문맥벡터를 활용해 다른 시퀀스를 생성하는 Transformer의 decoder부분을 활용

- 3.2 Supervised fine-tuning

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

각 task마다 label y를 예측한 뒤 최대화하는 식으로 변경

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

auxiliary objective를 추가하여 일반화 성능을 향상시키고 수렴 속도를 가속시킴.

- 3.3 Task-specific input transformations

기존에는, 전송된 representation 위에 새로운 architecture를 하나 더 달아 복잡한 구조였음

상당한 양의 task-specific customization이 필요하여 전이학습을 사용하지 않았음

NEW) task별로 구조화된 input으로 전이 학습 진행

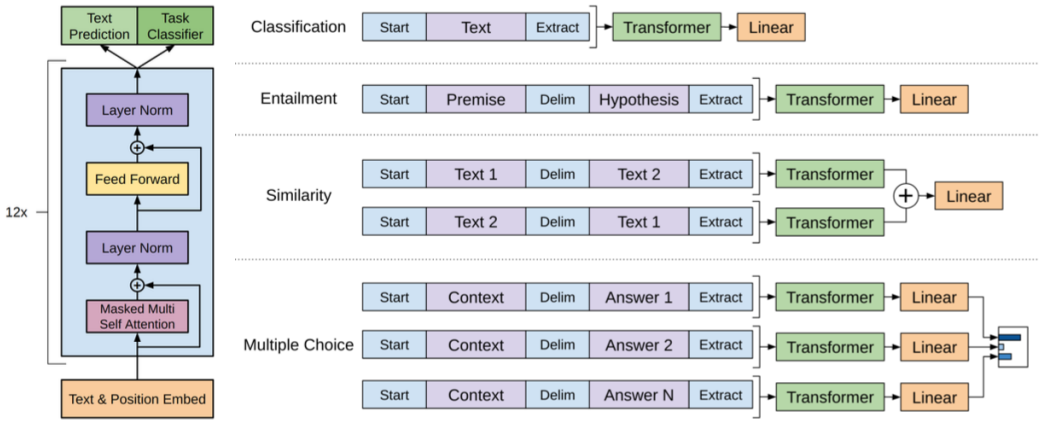


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Classification : task 입력 후 은닉층 업데이트 후 선형결합

Entailment : 전제와 가정 두가지의 시퀀스를 결합해 한번에 forward함

Similarity : 순서 반영을 위해 (Text1,Text2) (Text2,Text1)을 forward하여 concat하여 최종 결과값

Multiple Choice : (문맥,답1,답2,,,답N)을 forward하여 linear-softmax의 확률값이 출력

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**

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**

자연어추론, 질의응답, 문장유사도, 문법분류 등의 예시에서 좋은 성능을 발휘함

5. Analysis

LSTM과 비교했을 때,

구조화된 transformer의 attentional memory가 도움됨

transformer 구조의 귀납적 편향이 transfer을 도움

6. Conclusion

GPT : semi-supervised learning을 사용하여 task specific한 언어 모델

task별로 input을 변형해서 y를 맞추기 위한 supervised fine-tuning을 실시해 기존의 weight 조정