23.07.06 _ 4주차

딥러닝 논문 요약 및 구현 스터디

발표자

정명찬

GPT-1

Improving Language Understanding by Generative Pre-Training

Abstract

- **기존 NLP 태스크 모델**

  - Supervised learning 위해서, 많은 라벨링 된 데이터 필요
  - 데이터 양 부족 -> discriminatie learning 수행 어려움

- **GPT-1 제안**

  - 라벨링 없는 데이터는 확보 쉬움 -> Generative Pre-trained LM
  - 이후, 많은 구조 변화 없이 특정 태스크를 수행하도록 fine-tuning
  - 상식 추론, 질의 응답, 텍스트 유추 등 많은 태스크에서 성능 향상 달성

I. Introduction

기존 모델: unlabeled data로 사전 학습하더라도 word-level보다 많은 정보 이용하기 어려움

모델 아키텍쳐를 이후 태스크에 맞게 변경하거나, 복잡한 학습 방식 사용, 보조 학습 추가하는 방법으로 해결했었음

GPT-1: Unsupervised Pre-training & supervised fine-tuning -> semi-supervised 접근

대량의 unlabeled data -> 언어 모델링(신경망 초기 파라미터 학습) -> supervised 데이터로 뒤에 특정 태스크 활용

Transformer를 차용하여 사용. Pretrained model을 최소한으로 변경하여 fine tuning

각 작업에 맞게 모델을 개발한 경우보다 성능을 능가.

II. Related Work

Word-level 이상의 정보를 사전 훈련에서 담을 수 있도록, 구절 수준, 문장 수준 임베딩 사용

사전 훈련 후 fine-tuning 할 때, 최소한의 변경만을 적용

보조 비지도 학습을 추가하는 방식으로 의미론적 역할 라벨링 수행

Unsupervised pretraining 만으로도 특정 태스크의 언어적 측면을 이미 학습

III. Framework

1. 큰 텍스트 말뭉치에서 언어 모델 사전 학습
2. Labeled data로 fine tuning -> discriminative task 수행

$$\mathcal{U} = \{u_1, \ldots, u_n\},$$

U -> 토큰 코퍼스

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

K개 윈도우

그 다음 단어가 나올 확률 최대화

하도록 하는 목적 함수 L1(u)

III. Framework

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

h = Context vector U 를 We 와 곱해서 토큰 임베딩, Wp 로 포지셔널 임베딩 추가

transformer_block을 거치고 나온 h를 토큰 임베딩하여 softmax를 취한 게 P(u) (토큰 u가 나올 확률)

III. Framework

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

Fine-tuning

x : input token

y : label

위의 pretraining에서 나온 h를 Wy와 선형 결합하여 softmax를 취한 게 input이 들어왔을 때, 정답이 y일 확률

III. Framework

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

C(fine-tunin할 태스크의 코퍼스)
목적함수 L2(C)는 다음과 같이 input이 들어왔을 때 정답(y)일 확률. 이를 최적화

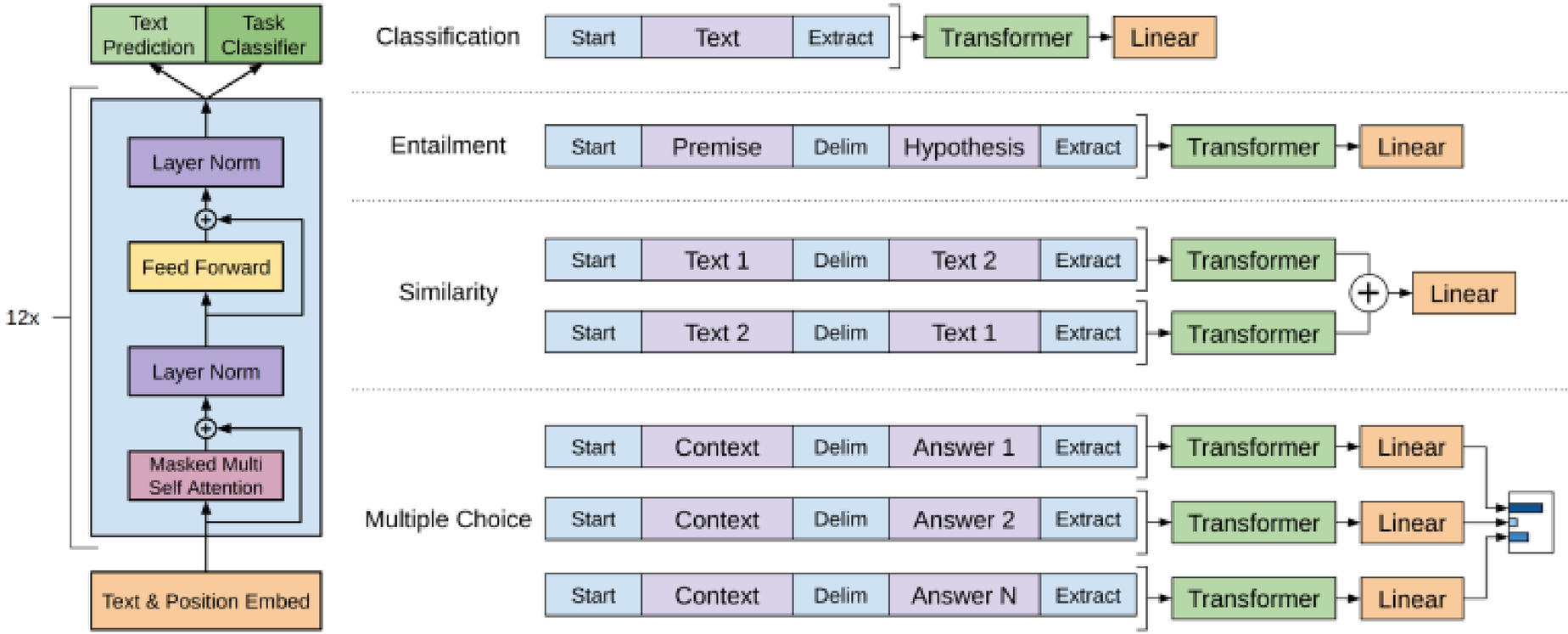
최종 목적 함수 L3(C)

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

L1(C)로 fine-tuning 태스크의 코퍼스를 사전훈련 목적함수에 업데이트 하는 과정을 더하니,

1. 일반화 성능 향상
2. 학습 가속화(빠른 convergence)

III. Framework



Transformer의 Decoder에서 encoder-decoder attention만 떼고 사용

각 태스크에 맞게 input 텍스트 조정해서 넣어줌

IV. Experiments

1) Pretraining

BooksCorpus 데이터셋(7000개 이상의 책)

Transformer 6 layers -> 12 layers

BPE encoding 사용

활성화 함수: GELU

포지셔널 임베딩: 기존의 사인파 버전 대신 학습 가능한 위치 임베딩 사용

2) Fine-tuning details

Pretraining의 하이퍼파라미터 사용

위의 목적함수에서 람다 값 0.5 설정

IV. 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

NLI – 문장 쌍을 읽고, 그들의 관계를 이해하는 방식

IV. Experiments

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**

질의 응답, 상식 추론 – Long-term context에 대해서 좋은 성능

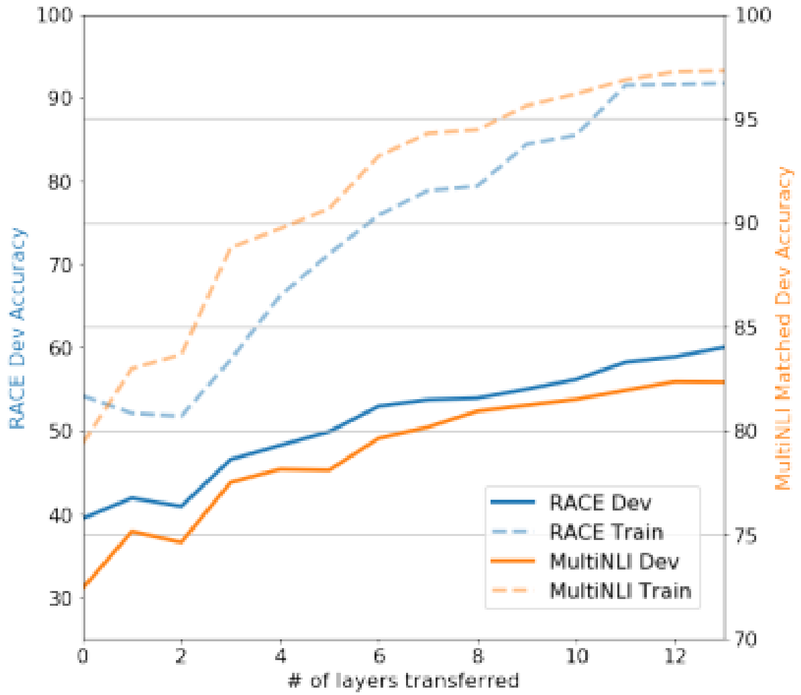
IV. Experiments

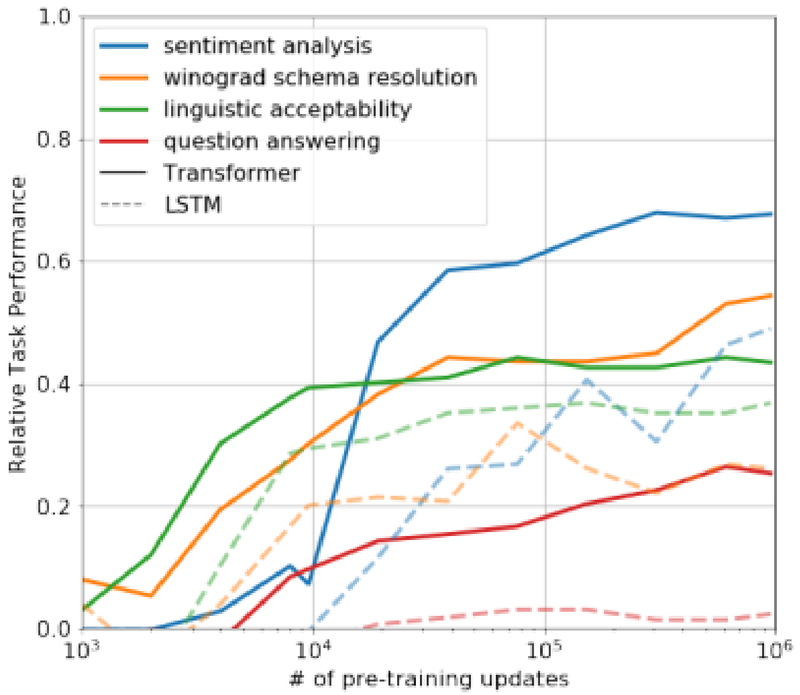
Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + 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**

의미론적 유사성 – 두 문장이 의미적으로 동일한 지 여부를 예측

분류 – 텍스트가 특정 클래스에 더 부합하는지 판단

V. Analysis





1) layer 수를 늘렸을 때, 정확도 향상

2) Zero-shot(훈련 시 본 적 없는 새로운 범주나 작업에 대해 수행하는 능력) 높음

LSTM(RNN 기반) 보다 더 구조화된 attentional memory로 transfer 용이

V. Analysis

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) 작은 데이터셋에 대해서는 supervised learnin의 corpus 까지 업데이트(L1(C)를 더하는 과정) 없이 하는 것이 유리
2) 하지만 큰 데이터셋에 대해서는 위 과정을 포함하는 것이 유리

VI. Conclusion

Generative pretraining + finetuning 으로 단일 과제에 구애받지 않는 모델

Long-term dependency 확보, 많은 데이터 학습 가능 및 여러 태스크에 효과적으로 transfer