



[9주차] 논문리뷰

GPT1

'Improving Language Understanding by Generative Pre-Training'

1. Introduction

- 논문이 다루는 분야
- 해당 task에서 기존 연구 한계점
- 논문의 contributions

2. Related Work

- Semi-supervised learning for NLP
- Unsupervised pre-training
- Auxiliary training objectives

3. 제안 방법론 (Framework)

1. Unsupervised pre-training
2. Supervised fine-tuning
3. 사전학습&파인튜닝에 대한 전체적처
4. Task-specific input transformations

4. 실험 및 결과

1. Setup(설정)
2. Supervised fine-tuning

5. Analysis

1. Impact of number of layers transferred
2. Zero-shot Behaviors
3. Ablation studies (구성요소 제거 실험)

6. 결론 (배운점)

PPO

'Proximal Policy Optimization Algorithms'

1. Introduction

- 논문이 다루는 분야
- 해당 task에서 기존 연구 한계점
- 핵심 아이디어
- 논문의 contributions

GPT1

'Improving Language Understanding by Generative Pre-Training'

1. Introduction



논문에서 다루고 있는 주제가 무엇인지와 해당 주제의 필요성이 무엇인가
논문에서 제안하는 방법이 기존 방법의 문제점에 대응되도록 제안 되었는가

자연어 분야는 요약, QnA, 유사성 평가, 문서 분류 등 다양한 분야에서 발전했음. unlabeled text는 많지만, 특정한 task에 label된 data는 부족. 본 논문은 unlabeled 데이터를 버리지 않고 generative pre-training에 사용하고 특정 task에 label된 데이터로 discriminative fine-tuning 진행하여 높은 성과를 냄. transformer의 디코더 구조를 바탕으로 하였고, pre-training에서 fine-tuning으로 넘어가는 과정에서 architecture는 최소한으로 변경되도록 하였음.

⇒ 크게 대규모의 unlabeled data를 사용하는 "Unsupervised Pre-training" 단계와 task에 특화된 labeled data를 사용하는 "Supervised Fine-tuning" 단계로 나뉨

논문이 다루는 분야

- 자연어 처리 (준지도 학습)

해당 task에서 기존 연구 한계점

- 라벨링 안 된 raw text 학습이 중요한 이유
 1. 지도학습 의존도를 줄이기 위해
 - 대부분의 딥러닝에서는 수작업으로 라벨링한 데이터 필요로 했었음
 - 라벨링된 자원이 부족한 분야에서는 적용이 쉽지 않았음.
 2. label된 데이터가 존재한다고 하더라도 성능향상을 위해 비지도 방식으로 representation 학습하는 건 중요한 과제임.
 - ex. 다양한 NLP 과제에서 성능 향상을 위해 pre-trained word embeddings 사용함.
- 단어 수준 넘어서는 정보를 라벨 없이 활용하기 어려운 이유
 1. transfer learning에 유용한 text representation을 학습하는 데에 어떤 종류의 최적화 objective가 효과적인지 불분명함.
 - language modeling, machine translation, discourse coherence 등

2. 학습된 표현을 목표로 하는 target task에 효과적으로 transfer하는 방법이 불분명함.

→ 기존에는 모델 구조 변경, 복잡한 학습 절차, auxiliary learning objectives 추가 등의 방식을 사용했었음.

논문의 contributions

- unsupervised pre-training + supervised fine-tuning (준지도)
- language understanding을 task로 삼음
- 목표 : 적은 수정만으로 다양한 과제에 전이 가능한 universal representation을 학습하는 것.
- 대규모의 라벨이 없는 텍스트 코퍼스(unsupervised pre-training), 수작업으로 주석이 달린 여러 개의 학습 데이터셋(supervised fine-tuning)을 사용할 수 있다고 가정. 두 데이터셋이 동일한 도메인일 필요X
 - "corpus" : 특정 주제나 목적에 맞춰 모아 놓은 언어 자료
- 학습절차
 1. 라벨이 없는 데이터를 사용하여 언어 모델링 목적(objective)을 최적화하여 신경망 모델의 초기 파라미터를 학습 (=좋은 표현을 학습하는 것)
 2. 이 파라미터들을 각 과제에 맞는 **지도 학습 목적**으로 조정(fine-tune)
- 모델 구조
 - Transformer 사용
 - RNN에 비해 텍스트 간 장기적인 의존 관계를 더 잘 처리함
- transfer 단계
 - traversal-style 접근법에서 영감을 얻은 과제 특화 입력 적응(input adaptation)을 사용함
 - traversal-style : 연속된 토큰 시퀀스로 구조화된 텍스트를 처리하는 방식
 - 모델 구조를 거의 변경하지 않고도 효과적으로 미세조정
- 모델 평가 task
 - natural language inference
 - question answering
 - semantic similarity

- text classification
- 이러한 과제들에서 처음부터 이 과제들을 위해 설계된 모델들보다 우수한 성능을 보였음. 특히 12개 중 9개에서 SOTA 성능을 능가했음.

2. Related Work



Introduction에서 언급한 기존 연구들에 대해 어떻게 서술하는가
제안 방법의 차별성을 어떻게 표현하고 있는가

Semi-supervised learning for NLP

- 자연어를 위한 준지도 학습은 sequence labeling이나 text classification등의 과제에 광범위하게 적용돼왔음.
- 초기
 - unlabeled data를 이용해 word-level & phrase-level statistics를 계산
→ 이를 supervised model의 feature로 사용
- 최근
 - unlabeled corpora에서 학습된 word embeddings을 다양한 과제에 적용하여 성능을 향상시킬 수 있다는 것이 입증됨.
 - BUT 단어 수준 정보(word-level information)의 전이에 국한됨
 - 우리 연구는 보다 높은 수준의 의미(high-level semantics)를 포착하려 함.

Unsupervised pre-training

- 좋은 initialization point를 찾는 데 목적을 둠
- 초기 : image classification와 regression 등의 과제에 사용
- 이후 : deep neural networks의 generalization을 향상시키는 정규화기법의 역할 밝혀짐
- 최근 : 이미지 분류, 음성 인식, 개체 소거(entity disambiguation), 기계 번역 등 다양한 과제에 적용되어 성능 향상
- 본 논문의 접근 : language modeling을 통해 신경망을 사전학습하고 지도학습으로 target task에 파인튜닝하는 것.
→ Dai et al.과 Howard & Ruder는 이 방식으로 텍스트 분류 성능 향상했음.

→ BUT 이들의 연구는 LSTM을 사용했기에 예측 가능한 문맥 범위가 짧다는 제한
→ 본 논문은 transformer 구조 사용해서 longer-range linguistic structure 포착함.
+ 다양한 과제에 대한 성능도 함께 제시함

- 사전학습된 언어 모델 또는 번역 모델의 은닉 표현을 auxiliary features로 활용하여 모델을 학습하기도 했음.
→ BUT 이렇게 되면 각 목표 과제마다 많은 수의 새로운 파라미터가 필요
→ 본 논문은 모델 구조 변화 없이 전이 학습 가능

Auxiliary training objectives

- Auxiliary Training Objectives 추가하는 것도 다른 형태의 준지도 학습.
- 초기 : 다양한 auxiliary NLP tasks를 통해 semantic role labeling을 향상했음
 - POS tagging, chunking, Named Entity Recognition, language modeling
- 최근 : **Rei** - 목표 과제에 auxiliary language modeling objective를 추가
→ 시퀀스 라벨링 과제(sequence labeling tasks)에서 성능 향상

3. 제안 방법론 (Framework)



Introduction에서 언급된 내용과 동일하게 작성되어 있는가

Introduction에서 언급한 제안 방법이 가지는 장점에 대한 근거가 있는가

제안 방법에 대한 설명이 구현 가능하도록 작성되어 있는가

1. Unsupervised pre-training

- 말뭉치 u 에 대해, 표준 언어 모델링 목적 함수(standard language modeling objective)를 사용하여 아래의 우도를 최대화함.

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

- k 는 context window의 크기
- 조건부확률 P 는 신경망을 통해 모델링됨
- 신경망의 파라미터 Θ 는 stochastic gradient descent를 사용해 학습됨

- 모델이 이전 단어들을 보고 다음 단어가 나올 확률을 예측하는 것
- 전체 말뭉치에 대해 예측 정확도를 높이는 방향으로 학습함
- 라벨이 없는 비지도 학습에 해당함.
- 모델 구조
 - multi-layer Transformer decoder 구조를 사용
- 1. 입력 문맥 벡터에 대해 다중 헤드 self-attention 연산
- 2. 그 뒤에 위치별 feed-forward 계층을 적용하여 출력 확률 분포를 생성
- 3. 모델의 작동 방식은 다음 수식과 같음
 - a. U = 현재 단어 u 예측 시 사용되는 문맥 토큰들
 - b. W_e 토큰 임베딩 행렬 / W_p 위치 임베딩 행렬
 - c. h_0 입력 시퀀스의 첫 번째 레이어 인풋 → 임베딩 + 위치정보 합
 - d. h_l Transformer의 l 번째 층의 출력
 - e. h_n 최종 층 출력
 - f. $P(u)$ softmax를 통해 구한 다음 단어에 대한 예측 확률 분포

$$\begin{aligned}
 h_0 &= UW_e + W_p \\
 h_l &= \text{transformer_block}(h_{l-1}) \forall i \in [1, n] \\
 P(u) &= \text{softmax}(h_n W_e^T)
 \end{aligned}$$

2. Supervised fine-tuning

- 앞선 식(1)의 목적 함수를 이용해서 사전학습을 끝냈다면, 여기서는 지도학습 target task에 맞게 파라미터 조정하는 fine tune 진행
- 지도학습 데이터셋 C 는 입력 토큰 시퀀스와 (x 들로 구성) 레이블 y 로 구성됨
- 입력 시퀀스가 사전학습된 모델 통과해서 transformer 마지막 층에서 활성벡터 h_l 계산. 그 결과를 linear output layer에 통과시켜서 y 에 대한 확률 분포를 예측

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

- 즉, 입력 시퀀스를 보고 정답 레이블을 예측할 확률을 최대로 하고자 함. 따라서 최종적으로 목적하고자 하는 목적 함수는 다음과 같음.

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

- 추가적으로 언어 모델링을 보조 목적으로 함께 사용하는 것이 지도 과제 일반화 성능 개선과 수렴 속도를 높이는 데에 도움이 되는 것을 발견했음. 이에 최종적으로는 이를 반영해서 다음과 같이 목적함수를 최적화함.

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

- L2는 지도 학습 목적(과제 레이블 예측), L1은 언어 모델링 목적(다음 토큰 예측), 람다는 가중치 조절

3. 사전학습&파인튜닝에 대한 전체적처

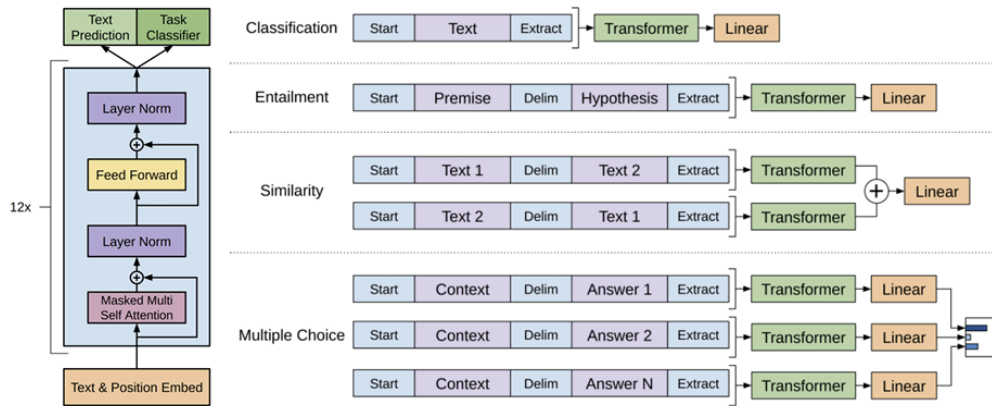


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

4. Task-specific input transformations

- 일부 과제에 대해서는 사전 학습된 모델을 바로 fine tuning에 사용할 수 있지만, 특정 과제들은 구조화된 입력을 가지고 있어서(사전 학습 모델이 연속된 텍스트 시퀀스를 가지는 것과 달리) 입력 형식에 대한 수정이 필요하다.
- 이전 연구들에서는 과제 특화 아키텍처를 학습시키는 방식을 제안했지만 이렇게 되면 커스터마이징이 많이 필요해서 전이학습의 이점을 살리지 못한다.

- 본 논문에서는 traversal-style 접근 방식을 채택하였다. 이 방식에서는 구조화된 입력을 사전학습 모델이 처리할 수 있는 정렬화된 시퀀스로 변환하여 입력한다. 이렇게 되면 사전학습 모델의 아키텍처를 따로 바꾸지 않고 fine tuning을 진행할 수 있게 된다.
- Textual entailment (자연어 추론)
 - 전제(premise) 문장과 가설(hypothesis) 문장의 토큰 시퀀스를 구분 토큰(\$)을 사이에 두고 하나의 입력 시퀀스로 concatenate함.
- Similarity (문장 유사도 판단)
 - 비교 대상인 두 문장 간에 순서가 따로 존재하지 않음. 두 개의 시퀀스를 생성하고 각 시퀀스마다 구분 토큰을 삽입함. 두 시퀀드를 독립적으로 transformer에 통과시켜 각각에 대한 표현 벡터를 얻고 두 표현을 원소 단위로 더해서 선형 출력층에 입력해서 예측함.
- Question Answering and Commonsense Reasoning (질문 응답 및 상식 추론)
 - 문맥, 질문, 여러개의 후보 답변 집합이 주어지는데 세 개를 [z;q;\$;ak]로 연결. 각 보기마다 해당 문맥과 질문을 결합하고 구분 토큰과 해당 보기를 붙이는 것. 시퀀스들이 각각 독립적으로 모델에 입력됨

4. 실험 및 결과



Introduction에서 언급한 제안 방법의 장점을 검증하기 위한 실험이 있는가

요약하자면 아래의 설정 상황에서 12개의 다양한 언어 과제 중 9개에서 SOTA 성능을 보였음. 또한 STS-B (~5.7k training examples) 처럼 작은 데이터셋에서부터 SNLI (~550k training examples) 같은 큰 데이터셋에서까지 성능이 좋았음.

1. Setup(설정)

- **BookCorpus**라는 대규모 말뭉치에서 언어 모델을 사전학습
 - **BookCorpus:** 인터넷에서 자유롭게 이용 가능한 약 7,000권의 책으로 구성, 약 8억 단어로 이루어짐. 이 책들은 다양한 장르로 이루어져 있어, 다양한 스타일과 주제의 텍스트를 포함함.
- 사전학습 후, 12개의 자연어 처리 과제에서 이 모델을 평가함

- 텍스트 분류(text classification), 자연어 추론(natural language inference), 문장 유사도 평가(sentence similarity), 상식 추론(commonsense reasoning), 질문 응답(question answering) 등
- 전이 학습
 - task-specific input representations을 정의함.
 - 각 입력은 single text sequence으로 변환되며, 사전 학습동안 사용된 것과 동일한 tokenizer를 사용하여 sub-word tokens으로 분할됨. 이후 시퀀스는 transformer 모델의 입력으로 들어가며, 과제마다 적절한 방식으로 출력.
 - 전체 supervised 데이터셋에서 전체 파라미터를 다시 학습(fine-tune)

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]

- 하이퍼파라미터는 개발 셋(dev set)의 성능에 따라 선택

2. Supervised fine-tuning

- 일관되게 입력 시퀀스를 구성
- Transformer 모델에 전달
- 마지막 토큰의 은닉 상태(hidden state)를 선형 분류기(linear classifier)에 통과시켜 예측을 수행
- 분류기의 출력은 소프트맥스(softmax) 층을 거쳐 확률로 변환
- 정답 레이블과의 cross-entropy loss를 최소화하도록 학습됨

▼ [Natural Language Inference]

- 전체와 가설 문장이 주어졌을 때 둘 사이의 관계를 판단하는 과제
- 함의 / 모순 / 중립 세 가지 중 하나의 label을 가짐
- 입력 형식 : "premise" 문장과 "hypothesis" 문장을 하나의 시퀀스로 연결하고, 그 사이에 구분 기호를 삽입
- 데이터셋 : 5개의 NLI 데이터셋(SNLI, MultiNLI, Question NLI, RTE, SciTail)
- 이 과제에서는 3 or 2 레이블 분류 문제를 풀게 됨

- entailment / contradiction / neutral
- entailment / not entailment
- RTE 제외하고는 본 논문 모델 성능이 가장 좋음

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>		
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	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 and commonsense reasoning]

- 하나의 질문과 여러 개의 보기 문장이 주어졌을 때, 이 중에서 정답을 선택하는 multiple choice 문제를 풀게 됨
- 입력 :
 - 질문과 각 보기를 결합하여 하나의 시퀀스로 구성
 - 각 보기에 대해 독립적으로 마지막 토큰의 은닉 벡터를 분류기에 통과시켜 score를 계산, 소프트맥스를 통해 모든 보기 중에서 가장 높은 확률을 가진 것을 선택
- 데이터셋 : RACE (독해 기반 질문 응답) / Story Cloze Test (상식 기반 이야기 완성)
- 모든 데이터셋에서 본 논문 모델 성능이 가장 좋음

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	<u>77.6</u>	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	<u>60.2</u>	<u>50.3</u>	<u>53.3</u>
Finetuned Transformer LM (ours)	86.5	62.9	57.4	59.0

▼ [Semantic Similarity]

- 두 문장이 주어졌을 때 그 의미가 얼마나 비슷한지 판단하는 문제
- 데이터셋에 따라 분류 또는 회귀 문제가 됨
- 입력 :

- 두 문장을 하나의 시퀀스로 연결하고, 사이에 구분 기호를 삽입
- 이 시퀀스를 Transformer에 입력하고, 마지막 토큰의 은닉 상태를 사용하여 유사도를 예측
- 데이터셋 :
 - MSR Paraphrase Corpus (MSRP) – 이진 분류 (동의어 여부)
 - Quora Question Pairs – 이진 분류
 - STS Benchmark – 회귀 문제, 두 문장 사이의 유사도 0~5로 예측
 - Loss는 MSE로 학습, 최종 결과는 Pearson correlation로 평가
- MRPC에서 제외하고는 본 논문 모델 성능이 가장 좋음

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]	-	-	-	<u>81.0</u>	-	-
Single-task BiLSTM + ELMo + Attn [64]	<u>35.0</u>	90.2	80.2	55.5	<u>66.1</u>	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	<u>68.9</u>
Finetuned Transformer LM (ours)	45.4	91.3	82.3	82.0	70.3	72.8

▼ [Classification]

- 단일 문장 입력에 대한 텍스트 분류 문제. 감정 분석이나 문법성 판단 등
- 입력 형식 :
 - 입력 문장은 하나의 시퀀스로 구성되며, 서브워드 토큰라이저를 거쳐 Transformer에 입력
 - 마지막 토큰의 은닉 벡터를 사용하여 분류기를 통해 예측을 수행
- 데이터셋 :
 - SST-2 (Stanford Sentiment Treebank): 긍정/부정 감정 분류 이진 과제
 - CoLA (Corpus of Linguistic Acceptability): 문장이 문법적으로 타당한지 여부를 판단하는 이진 과제
- CoLA에서는 본 논문 모델 성능이 가장 좋음

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STS-B (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	93.2	-	-	-	-
TF-KLD [23]	-	-	86.0	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	<u>81.0</u>	-	-
Single-task BiLSTM + ELMo + Attn [64]	<u>35.0</u>	90.2	80.2	55.5	<u>66.1</u>	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	<u>68.9</u>
Finetuned Transformer LM (ours)	45.4	91.3	82.3	82.0	70.3	72.8

5. Analysis

1. Impact of number of layers transferred

- layer 개수 변화의 영향 확인하고자 함.

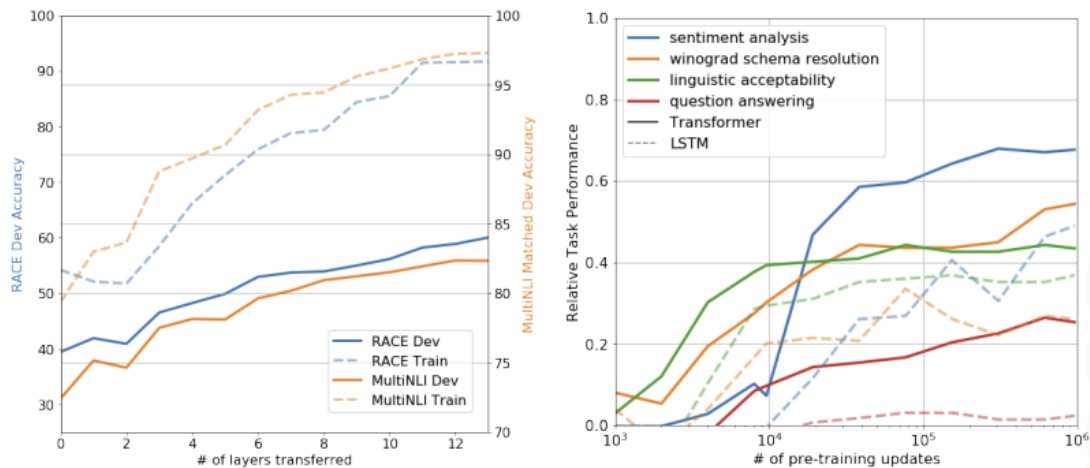


Figure 2: **(left)** Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. **(right)** Plot showing the evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model.

- 왼쪽 그래프는 전이된 층수에 따른 RACE와 MultiNLI에서의 성능 보여줌
 - 파랑 - RACE / 노랑 - MultiNLI
 - 임베딩(embeddings)을 전이하는 것만으로도 성능이 향상
 - 층을 더 많이 전이할수록 성능이 꾸준히 향상
 - 전체 층을 전이할 경우 최대 9%의 성능 향상 (MultiNLI)
 - 사전학습된 모델의 각 층이 목표 과제를 해결하는 데 유용한 기능을 내포함
- 오른쪽 그래프는 사전학습이 진행됨에 따라 여러 과제에서의 zero-shot 성능이 어떻게 변화하는지 나타낸 그래프 (전이학습 없이 사전학습한것로만 측정)

- x축 : 사전학습에서의 업데이트 횟수
- 서로 다른 색상의 선 = 서로 다른 downstream task
- 실선: Transformer 기반 언어 모델 / 점선: LSTM 기반 모델
- Transformer 기반 모델은 업데이트 횟수가 많아질수록 zero-shot 성능이 급격히 향상
- LSTM은 Transformer보다 전체적으로 성능이 낮으며, 성능 상승도 더딤
- 감정 분석(sentiment analysis) 같은 과제에서는 사전학습만으로도 상당한 성능이 가능함

2. Zero-shot Behaviors

- Transformer 기반 언어 모델 사전학습이 왜 효과적인지에 대한 것
- 가설 :
 - underlying generative model이 언어 모델링 성능을 향상시키기 위해 많은 과제들을 자연스럽게 수행하는 방법을 학습한다.
 - Transformer의 attentional memory가 LSTM에 비해 전이에 도움을 준다
- 실험 :
 - 지도 방식의 미세조정(supervised fine-tuning) 없이 underlying generative model만을 활용하여 다양한 과제를 수행하는 휴리스틱(heuristic) 방식 설계
 - 휴리스틱(Heuristic) : 꼭 정답은 아니지만, 직관적으로 문제를 해결할 수 있는 간단한 규칙이나 요령
 - CoLA (문법 타당성 판단, linguistic acceptability)
 - 각 문장에 대해, 생성 모델이 부여하는 토큰별 로그 확률의 평균(average token log-probability)을 계산하고, 이를 임계값(thresholding)을 기준으로 문법적으로 타당한지 여부를 예측
 - SST-2 (감정 분석, sentiment analysis)
 - 각 입력 문장 뒤에 토큰 "very"를 덧붙인 후, 언어 모델의 출력 단어 분포를 "positive"와 "negative" 두 단어로 제한 하고, 그 중 더 높은 확률을 부여한 단어를 예측값으로 선택
 - RACE (질문 응답, question answering)
 - 문서와 질문을 조건으로 주었을 때, 각 선택지(answer)에 대해 생성 모델이 부여한 평균 토큰 로그 확률 을 계산하고, 가장 높은 값을 받은 선택지를 정답으로

예측

- DPRD (Winograd schema 해결)
 - 지문 내의 정관사 대명사(definite pronoun)를 두 개의 후보 참조어 중 하나로 대체한 후, 그 다음 시퀀스에 대해 생성 모델이 부여한 평균 토큰 로그 확률을 비교하여 더 높은 확률을 부여한 참조어가 정답이라고 판단
- 결과 :
 - 휴리스틱들의 성능은 안정적으로 유지되며 훈련이 진행되면서 꾸준히 향상
→ generative pretraining이 task-relevant functionality를 광범위하게 학습하는데 도움이 됨을 시사함.
→ fine-tuning 없이도 좋은 성능이 나온다면, 사전학습이 과제 해결 능력을 내재하고 있음을 의미하는 것
 - LSTM은 zero-shot 성능에서 더 높은 분산(variance)을 보였음
→ Transformer 아키텍처의 inductive bias가 전이 학습을 도와줌을 시사

Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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

3. Ablation studies (구성요소 제거 실험)

요약하자면, auxiliary LM objective는 큰 데이터셋에서 성능 향상에 기여,

Transformer는 LSTM보다 전반적으로 우수,
사전학습 또한 성능 향상에 중요한 역할을 수행함.

1. auxiliary LM objective의 효과

→ fine-tuning 과정에서 auxiliary LM objective 없이 수행했을 때의 성능을 평가

- 보조 목적은 NLI 과제들과 QQP(Quora Question Pairs)에서 성능 향상에 도움을 주는 것으로 판단
- 전반적으로 데이터셋 크기가 클수록 보조 목적의 효과가 큼

2. Transformer의 효과

→ 동일한 프레임워크 하에서 하나의 층으로 이루어진 2048 유닛 LSTM과 비교

- Transformer 대신 LSTM을 사용했을 경우 평균 성능이 5.6점 하락

3. 사전학습의 효과

→

사전학습 없이, 각 지도 과제에 대해 직접 학습한 경우와 비교

- 사전학습이 없을 경우 모든 과제에서 성능이 저하되었으며, 전체 모델 대비 평균적으로 14.8% 성능 감소

6. 결론 (배운점)



연구의 의의 및 한계점, 본인이 생각하는 좋았던/아쉬웠던 점 (배운점)

- unlabel 코퍼스에서 비지도 방식으로 언어모델을 pre-train하고, 다양한 자연어 이해 과제에 대해 지도 방식으로 fine-tune하는 방식을 제안하였다
- 다양한 자연어 이해 과제들에 적용 가능함.
- 복잡한 구조나 목적함수 없이도 다양한 과제에서 높은 성능을 낼 수 있음을 입증
- 사전학습이 감독학습 과제의 generalization performance를 크게 향상시킬 수 있음을 보여준 것.

PPO

'Proximal Policy Optimization Algorithms'

1. Introduction



논문에서 다루고 있는 주제가 무엇인지와 해당 주제의 필요성이 무엇인가

논문에서 제안하는 방법이 기존 방법의 문제점에 대응되도록 제안 되었는가

논문이 다루는 분야

- 강화학습
- 강화학습에서 널리 사용되는 대표적인 최적화 알고리즘을 제안하였음

해당 task에서 기존 연구 한계점

- scalable, data efficient, robust 셋을 모두 만족시키는 방법X
- Deep Q-Learning
 - 함수 근사와 결합했을 때 단순한 문제도 못푸는 경우가 많음
- policy gradient
 - 데이터 효율성과 강인성이 떨어짐
- Trust Region / Natural Policy Gradient Methods (TRPO)
 - 복잡한 알고리즘
 - dropout, 정책/가치함수 간 파라미터 공유/auxiliary tasks 포함하는 아키텍처와 호환X

핵심 아이디어

- TRPO의 데이터 효율성과 안정된 성능은 유지하면서도, 1차 최적화 기법만을 사용하는 간단한 알고리즘을 제안
- 확률 비율을 클리핑(clipping)하는 새로운 목적 함수를 제안
→ 정책의 성능에 대한 pessimistic estimate(lower bound)를 형성
- 정책을 최적화하기 위해, 정책으로부터 데이터를 샘플링하고, 해당 데이터에 대해 여러 번의 미니배치 최적화를 수행

논문의 contributions

- 여러 종류의 surrogate objective 비교 → 확률 비율을 클리핑한 버전이 가장 좋은 성능을 보임
- continuous control 과제에서 가장 뛰어난 성능을 보임
- Atari 환경에서도 A2C보다 뛰어난 샘플 효율성
- ACER와 비슷한 성능을 유지, 더 단순한 구현