GPT-1

📌 논문 제목

Improving Language Understanding by Generative Pre-Training

👨‍💻 저자

- Alec Radford 외 3인

✍️ 링크

- https://arxiv.org/pdf/1512.03385

1. Introduction

- 핵심 문제
  - 데이터 불균형: 자연어 이해를 위한 레이블 없는 데이터는 풍부하지만, 특정 태스크를 위한 레이블 있는 데이터는 부족
  - 기존 모델의 한계: 특정 목적에만 맞춰 학습된 판별 모델은 데이터가 적으면 성능이 떨어지고, 레이블 없는 데이터에서 범용적인 정보를 추출하는 데 어려움
- 해결책 제안: 2단계 학습 프레임 워크

  ⇒ Generative Pre-training(생성적 사전 학습) + Discriminative Fine-tuning(판별적 미세조정)

  (1) 비지도 사전 학습(Unsupervised Pre-training): 대규모의 레이블 없는 텍스트에서 언어 모델링을 통해 보편적 언어 특징 먼저 학습

  (2) 지도 미세 조정(Supervised Fine-tuning): 학습된 파라미터를 바탕으로, 실제 해결하려는 특정 태스크(질문 답변, 분류 등)의 소량 데이터를 사용하여 모델 최적화
- 주요 기술적 특징
  - Transformer 아키텍처 사용: 기존의 RNN(LSTM)보다 긴 문맥을 더 잘 포착
  - 입력 변환(Input Transformation): 태스크마다 모델 구조를 바꿀 필요 없이 입력 데이터의 형태만 조정하여 다양한 태스크에 동일한 모델 적용

2. Related Work

- Word to Sentence
  - 기존의 준지도 학습은 주로 워드 임베딩 수준의 정보 전이에 그침
  - 최근 연구와 본 논문은 레이블 없는 데이터로부터 구절이나 문장 수준의 더 깊은 의미(high-level semantic)을 학습하여 전달하는 데 집중
- Pre-training & Fine-tuning
  - 비지도 사전 학습은 모델이 더 나은 초기 지점에서 시작하게 하며, 일반화 성능을 높이는 정규화 역할을 함
  - 기존 연구처럼 사전 학습 후 미세 조정을 거치지만, LSTM 대신 Transformer를 사용하여 더 긴 문맥을 효과적으로 학습
- Auxiliary Objectives
  - 학습 시 주된 태스크 외에 언어 모델링과 같은 보조적인 학습 목표를 사용하는 것이 성능 향상에 도움
  - GPT는 사전 학습을 통해 언어 특징을 배운 뒤 미세 조정 단계에서 보조 목표를 활용하여 학습 효율을 높임

---

3. Framework

3.1 Unsupervised pre-training(비지도 사전 학습)

- 다음 log-likelihood를 최대화하도록 표준 언어 모델링 목적 함수 사용

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

  - *k*: 컨텍스트 윈도우 크기
  - 조건부 확률 *P*: 파라미터 theta를 가진 신경망을 통해 모델링

  파라미터들은 SGD를 통해 최적화

- 언어 모델 구조: Transformer의 디코더

  ⇒ 입력 컨텍스트 토큰에 대해 multi-head self-attention 연산 적용 후 목표 토큰에 대한 확률 분포를 출력하기 위해 position-wise feed-forward 층을 거침

3.2 Discriminative fine-tuning(지도 미세 조정)

사전 학습을 마친 후, 파라미터를 레이블이 있는 데이터셋 C에 맞게 조정

각 인스턴스는 입력 토큰 시퀀스 x^1 ~ x^m과 레이블 y로 구성

입력 시퀀스는 사전 학습된 모델을 통과하여 최종 트랜스포머 블록의 활성값 h_l^m을 얻고, 이를 선형 출력 층에 통과시켜 y 예측

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

다음 목적 함수를 최대화

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

⇒ 미세 조정 단계에서 언어 모델링을 보조 목적 함수로 포함시키는 것이 학습의 일반화를 돕고 수렴을 빠르게 함

- 최종 최적화 목적 함수:

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

3.3 Task-specific input transformations(태스크별 입력 변환)

텍스트 분류와 같은 태스크는 위처럼 직접 미세 조정이 가능하지만 질문 답변이나 문장 함의 같은 태스크는 입력 형태가 다르기 때문에 traversal-style 접근법을 사용하여 다양한 형태의 데이터를 사전 학습 모델이 처리할 수 있는 연속된 시퀀스로 변환

- 문장 함의(Entailment): 전제(P)와 가설(H)을 구분자로 연결
- 유사도(Similarity): 문장 순서에 고유한 의미가 없으므로 양방향 순서(A+B, B+A)를 모두 처리하여 합산
- 질문 답변 및 상식 추론: Context, Question, 각 답변 후보 Answer_i를 연결하여 별도의 시퀀스를 만들고 각각의 점수 계

---

4. Experiments

4.1 Setup

- 사전 학습(Unsupervised pre-training)
  - BooksCorpus 데이터셋 사용

    - 긴 범위의 연속적 텍스트 포함하여 모델이 장거리 정보 학습 가능
    - 약 0.95%의 낮은 perplexity 달성
- 모델 사양(Model specifications)
    - 12개 layer로 구성된 디코더 전용 트랜스포머
    - 768차원의 state 벡터 // 12개의 attention head
    - feedforward 네트워크의 내부 차원: 3072
    - 최적화: Adam, 최대 학습률: 2.5e-4
    - 2000번의 step동안 학습률 서서히 높이는 워밍업(warm-up) 사용, 이후 코사인 함수에 따라 0까지 줄임
    - 64개의 무작위 샘플로 구성된 미니배치를 사용하여 100 epoch 동안 학습
- 미세 조정 세부사항(Fine-tuning details)
    - 사전 학습 시의 하이퍼파라미터 그대로 사용
    - Dropout 비율: 0, 대부분의 태스크에서 학습률: 6.25e-5, 배치 크기 32
    - 3 epoch 내외로 빠르게 수행

4.2 Supervised Fine-tuning results

- 자연어 추론(Natural Language Inference)
    - 문장 간의 관계를 파악하는 태스크
    - SNLI, MultiNLI 등 5개 데이터셋에서 평가, 모든 데이터셋에서 SOTA 달성
- 질문 답변 및 상식 추론(Question Answering and Commonsense Reasoning)
    - RACE(중고등학교 시험 문제), Story Cloze Test 데이터셋 사용
    - 기존 모델들보다 훨씬 뛰어난 성능

      → 트랜스포머가 긴 문맥을 효과적으로 포착 가능 입증
- 의미론적 유사성(Semantic Similarity)
    - 두 문장이 같은 뜻인지 판별하는 태스크
    - Microsoft Paraphrase corpus(MRPC), STS-B 등을 포함한 3개 데이터셋 중 2개에서 최고 성능 달성
- 분류(Classification)

  문법적 정확성을 판단하는 CoLA와 감성 분석인 SST-2에서 평가

CoLA에서 기존 최고 기록을 앞지름

5. Analysis

- Impact of number of layers transferred(사전 학습 층의 수에 따른 영향)
  - 비지도 사전 학습에서 지도 타겟 태스크로 전이되는 layer 수가 성능에 미치는 영향
  - 사전 학습된 layer를 하나씩 추가할 때마다 성능이 점진적으로 향상

    ⇒ 사전 학습된 모델의 각 layer가 태스크를 해결하는 데 유용한 보편적인 특징들을 학습하고 있음을 의미
- Zero-shot Behaviors(제로샷 성능)
  - 왜 트랜스포머의 언어 모델 사전 학습이 효과적인가?
    - 언어 모델의 성능을 향상시키기 위해 학습하는 기본 모델이 많은 태스크를 수행하는 능력을 이미 습득
    - 사전 학습이 진행됨에 따라 미세 조정 없이도 다양한 태스크의 성능이 꾸준히 상승

      ⇒ 생성적 사전 학습이 광범위하고 유용한 태스크 관련 기능들을 학습하는 데 도움이 되는 것을 시사함
- Ablation studies
  - 보조 언어 모델링 목표의 효과:

    미세 조정 단계에서 보조 언어 모델링 목표를 제거했을 때의 영향

    ⇒ NLI 태스크와 대규모 데이터셋에서는 보조 목표가 도움이 되지만, 소규모 데이터셋에서는 오히려 성능 저하
  - Transformer vs LSTM:

    동일한 프레임워크에서 트랜스포머 대신 2층짜리 LSTM 모델을 사용하여 비교

    ⇒ 트랜스포머보다 성능 하락, 특히 긴 문맥 다루는 태스크
  - 사전 학습의 필요성:

    사전 학습 없이 직접 지도 학습만 수행했을 때 비교

    ⇒ 모든 태스크에서 사전 학습을 거친 모델보다 성능이 훨씬 낮았으며, 이는 사전 학습이 모델의 성능 향상에 결정적인 역할을 한다는 것 증명

6. Conclusion

본 연구는 생성적 사전 학습과 판별적 미세 조정을 결합하여 태스크에 의존하지 않는 단일 모델로 강력한 자연어 이해 성능을 달성하는 프레임워크를 제안한다.

다양한 장르를 포함하는 넓은 범위의 연속된 텍스트 코퍼스에서 사전 학습을 수행함으로써 모델은 global한 지식과 장거리 의존성 처리 능력을 습득한다. 이 사전 학습된 표현은 질문 응답, 의미 유사성, 의미론적 함의, 텍스트 분류 등 12개 중 9개 태스크에서 최고 수준의 성능으로 성공적으로 전이된다.

비지도 사전 학습을 통해 판별 모델의 성능을 향상시키는 것이 실제로 가능함을 보였으며, Transformer 구조와 장거리 의존성을 포함한 텍스트 데이터가 특히 효과적임을 제시한다.

BERT

📌 논문 제목

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

---

👨‍💻 저자

- Jacob Devlin 외 3인

✍️ 링크

- https://arxiv.org/pdf/1512.03385

[Abstract]

최근의 언어 표현 모델들과 달리, BERT는 모든 층에서 좌측과 우측 문맥을 동시에 조건으로 사용하는 깊은 양방향 표현을 비지도 텍스트로부터 사전학습하도록 설계되었다. 그 결과, 사전학습된 BERT 모델은 하나의 추가적 출력 층만 더하여 fine-tuning하는 것만으로도 광범위한 과제에서 최소한의 과제별 아키텍처 수정으로 최첨단 성능을 달성할 수 있다.

1. Introduction

사전학습된 언어 표현을 특정 태스크에 적용하는 기존 전략에는 두 가지가 있다.

첫번째는 feature-based 접근법으로, ELMo가 그 예이며, 사전학습된 표현을 추가적인 입력 특징으로 사용하는 과제별 아키텍처를 활용한다. 두번째는 fine-tuning 접근법으로 OpenAI GPT가 그 예이며, 최소한의 과제별 파라미터만 추가한 뒤 모든 사전학습 파라미터를 다운스트림 과제에서 함께 fine-tuning한다.

이 두 접근법은 사전학습 단계에서 단방향 언어 모델을 사용하여 일반적인 언어 표현을 학습한다는 점에서 공통점을 가진다.

이러한 단방향 언어 모델은 사전학습 시 사용할 수 있는 아키텍처의 선택을 제한하기 때문에 문장 수준 과제에서는 비효율적이며, 질문 응답과 같은 토큰 수준 과제에 fine-tuning 기반 접근법을 적용할 때에는 양방향 문맥을 통합하는 것이 필수적이기 때문에 매우 치명적일 수 있다.

→ 본 논문에서는 BERT를 제안한다. BERT는 masked language model 사전학습 목표를 사용하여 입력 토큰 중 일부를 무작위로 마스킹하고 해당 토큰의 원래 단어를 문맥만을 사용

해 예측하도록 한다. 또한 masked language model 과 함께 next sentece prediction 과제를 사용하여 문장 쌍 표현을 공동으로 사전학습한다.

---

2. Related Work

2.1 Unsupervised Feature-based Approaches

- 초기 연구들은 단어 임베딩을 사전학습하여 NLP 성능을 향상시켰다.
  - 비신경망 기반 방법 & 신경망 기반 방법
  - 좌 → 우 언어 모델 또는 주변 문맥 구분 목적 함수 사용
- ⇒ 문장 임베딩 / 문단 임베딩으로 확장
- ELMo는 이 흐름을 확장하여 좌→우, 우→좌 언어 모델을 각각 학습하고, 두 표현을 연결하여 문맥적 단어 표현을 생성한다. 이는 feature-based 접근법으로 과제 특화 아키텍처에 입력으로 사용된다.
- 이러한 feature-based 접근법은 강력하지만 사전학습된 표현이 모델 내부에서 직접 미세조정되지는 않는다.

2.2 Unsupervised Fine-tuning Approaches

- 최근에는 사전학습된 언어 모델을 다운스트림 과제에서 직접 fine-tuning 하는 접근 등장
- OpenAI GPT는 좌→우 Transformer 언어 모델을 사전학습하고 모든 파라미터를 다운스트림 과제에서 fine-tuning
- 이러한 접근법은 사전학습 단계에서 단방향(unidirectional) 문맥만 사용하기 때문에 양방향 문맥이 필요한 과제에서 한계를 가짐

---

3. BERT

프레임 워크: (1) pre-training → (2) fine-tuning

> 서로 다른 과제 전반에 걸쳐 통일된 아키텍처를 사용하여 사전학습 단계의 아키텍처와 최종 다운스트림 아키텍처 사이에는 거의 차이가 없다.

[Model Architecture]

- 다층 양방향 Transformer 인코더로 구성(Transformer 구조는 Vaswani et al.에 기반하며, 각 층은 양방향 self-attention 사용
- 기호
  - L: Transformer 층 수

    - *H*: 은닉 차원
    - *A*: self-attention 헤드 수
- 모델 크기
    - BERT_BASE
        - L=12, H=768, A=12
        - 파라미터 수: 110M
    - BERT_LARGE
        - L=24, H=1024, A=16
        - 파라미터 수: 340M
    - BERT_BASE는 OpenAI GPT와 동일한 크기로 설정되었지만, BERT는 양방향 self-attention, GPT는 좌→우 단방향 self-attention을 사용한다는 점에서 본질적으로 다르다.
    - 모든 모델에서 feed-forward 층의 차원은 4H

[Input/Output Representations]

- BERT는 단일 문장과 문장 쌍을 하나의 토큰 시퀀스로 처리
- 입력은 WordPiece 임베딩, 어휘 크기 30,000
- 모든 입력 시퀀스의 첫 토큰은 [CLS], 이 토큰의 최종 은닉 벡터는 분류 과제에서 전체 시퀀스 표현으로 사용
- 문장 쌍 입력은 [SEP] 토큰으로 문장 구분, 각 토큰에 문장 A/B를 나타내는 세그먼트 임베딩을 추가
- 각 토큰의 최종 입력 표현: 토큰 임베딩, 세그먼트 임베딩, 위치 임베딩의 합

3.1 Pre-training BERT

- BERT는 양방향 Transformer 인코더를 사용해 언어 표현을 사전학습

  ⇒ 두 가지 비지도 사전학습 과제

(1) Masked Language Model(MLM)

- 표준 좌→우/우→좌 언어 모델 대신 입력 토큰의 일부를 마스킹하고 양방향 문맥으로 원래 단어 예측
- 각 입력 시퀀스에서 15%의 WordPiece 토큰 선택,
    - 80% → [MASK]로 대체

    - 10% → 무작위 토큰으로 대체
    - 10% → 원래 토큰 유지

    ⇒ fine-tuning 단계에서 [MASK] 토큰이 등장하지 않는 문제 완화를 위한 설계, 마스킹된 토큰의 원래 단어 예측이 목적

(2) Next Sentence Prediction(NSP)

- 질문 응답, 자연어 추론 등 문장 간 관계를 요구하는 과제를 위해 도입
- 입력은 문장 A와 문장 B
    - 50%: B가 A의 실제 다음 문장(IsNext)
    - 50%: B가 무작위 문장(NotNext)
- 모델은 문장 B가 문장 A의 실제 다음 문장인지 여부 예측

3.2 Fine-tuning BERT

- 텍스트 쌍 과제에서 기존 방식(별도 인코딩+cross attention)을 사용하지 않고, 결합된 입력을 self-attention으로 한 번에 인코딩하여 문장 간 양방향 상호작용을 자연스럽게 포함
- 각 다운스트림 과제에서는 과제 특화 입력/출력 층만 추가, 모든 파라미터를 end-to-end로 fine-tuning
- 입력:
    - 패러프레이징: 문장-문장
    - NLI(자연어 추론): 가설-전제
    - QA(질문 응답): 질문-지문
    - 분류/태깅: 단일 문장
- 출력 사용 방식:
    - 토큰 표현 → 토큰 수준 과제(QA, 시퀀스 태깅)
    - [CLS] 표현 → 분류 과제(MLI, 감성 분석)
- fine-tuning 비용은 낮아 단일 Cloud TPU 기준 최대 한시간, GPU는 수 시간 내 수행 가능

---

4. Experiments

- 사전학습된 BERT를 다양한 다운스트림 과제에 fine-tuning했을 때의 성능

- 총 11개의 자연어 처리 과제에 대해 과제별 아키텍처를 거의 추가하지 않고도 높은 성능

4.1 GLUE(여러 자연어 이해 과제로 구성된 벤치마크)

- BERT: 입력을 단일 문장 또는 문장 쌍으로 구성, [CLS]토큰의 최종 은닉 벡터를 전체 입력의 집계 표현으로 사용, 분류 층 하나만 추가하여 fine-tuning
- 결과: BERT_LARGE는 모든 GLUE 과제에서 기존 최고 성능을 능가하거나 동등, 평균 GLUE 점수에서도 최고 성능 달성

4.2 SQuAD v1.1

- 질문(문장 A)와 지문(문장 B)를 하나의 시퀀스로 결합하여 입력
- 각 토큰의 은닉 표현을 사용해 답변 시작 위치, 답변 종료 위치를 예측
- 결과: 기존 최고 성능 대비 +1.5 F1 향상

4.3 SQuAD v2.0

- SQuAD v2.0은 답변이 없는 질문을 포함
- BERT: 답변이 없는 경우를 [CLS] 토큰이 시작/종료 위치인 경우로 모델링, “답변 없음” 점수와 “가장 좋은 답변 span” 점수를 비교, 임계값은 개발 세트에서 F1 기준으로 선택
- 결과: 기존 최고 성능 대비 +5.1 F1 향상

4.4 SWAG(상식 추론 기반의 문장 완성 선택 과제)

- BERT: 문장 A + 각 후보 문장 B를 결합, 총 4개의 입력 시퀀스 생성, [CLS] 표현과의 dot product로 각 후보의 점수 계산
- 결과: BERT_LARGE가 ESIM+ELMo 대비 +27.1%, OpenAI GPT 대비 +8.3% 성능 향상

---

5. Ablation Studies

5.1 Effect of Pre-training Tasks

MLM과 NSP의 기여도를 분리해서 평가

- NSP 제거 시

  자연어 추론과 같이 문장 간 관계를 요구하는 과제에서 성능 저하

  → NSP는 문장 관계 모델링에 중요

- MLM 제거 + 좌→우 언어 모델만 사용시

  여러 다운스트림 과제에서 성능 저하 발생

→ 깊은 양방향 사전학습 자체가 핵심

5.2 Effect of Model Size

서로 다른 층 수, 은닉 차원, 어텐션 헤드 수를 가진 모델들을 비교한다.

- 기존: 모델 크기 증가가 항상 성능 향상으로 이어지지는 않음
- 결과: 모델 크기가 커질수록 성능이 일관되게 향상, 아주 작은 다운스트림 데이터셋에서도 동일하게 관찰
- BERT:
  - 사전학습 충분히 수행
  - 다운스트림에서 직접 미세조정
  - 무작위 초기화 파라미터 최소화

⇒ 이 조건하에서 큰 모델의 표현력이 작은 데이터셋에서도 효과적

5.3 Feature-based Approach with BERT

- 지금까지의 BERT 결과는 모두 fine-tuning 기반 접근법

  → feature-based 접근법은??

  - Transformer 인코더로 직접 표현하기 어려운 과제에 적용 가능
  - 사전 계산된 표현을 재사용할 수 있어 계산 효율 높음

5.4 Comparison of BERT, ELMo, OpenAI GPT

- 세 모델의 차이를 아키텍처와 학습 방식 관점에서 비교
- 접근법
  - BERT, OpenAI GPT → fine-tuning 기반 // ELMo → feature-based
- GPT

  좌 → 우 Transformer 언어 모델을 사전학습
- BERT

  GPT와의 비교를 명확히 하기 위해 모델 크기와 구조를 의도적으로 최대한 유사하게 설계, 양방향

---

6. Conclusion

비지도 사전학습이 많은 언어 이해 시스템의 핵심 구성 요소임을 입증하였으며, 이러한 결과들은 low-resource 과제들도 단방향 아키텍처의 이점을 누릴 수 있도록 했다. 본 논문은 이

러한 발견을 깊은 양방향 아키텍처로 확장하여 동일한 사전학습 모델이 광범위한 자연어 처리 과제들을 성공적으로 해결할 수 있도록 했다.

GPT와 BERT 비교

1. 프레임워크

[GPT]

- 생성적 사전 학습(generative pre-training) + discriminative fine-tuning 구조
- 언어 모델링을 통해 연속 텍스트의 확률 구조 학습 → 다양한 판별 과제로 전이

[BERT]

- 비지도 사전학습 + end-to-end finetuning 구조
- 사전학습 단계에서부터 판별적 태스크에 적합한 표현 학습을 명시적으로 설계

2. 사전 학습 목표 차이점

[GPT]

- 표준 생성적 언어 모델 기반
- 연속 텍스트 코퍼스에서 다음 토큰을 예측하는 방식의 사전학습

  ⇒ 언어 생성에 자연스럽게 맞는 표현을 학습

[BERT]

- 생성 목적이 아닌, 표현 학습 중심의 사전학습
- MLM과 NSP 두 가지 사전학습 과제를 사용

  ⇒ 토큰 수준 + 문장 수준 정보를 동시에 학습, 문장 간 관계를 명시적으로 모델링

3. 문맥 방향성

[GPT]

- 단방향 문맥: 좌 → 우
- 사전 학습 단계에서 각 토큰은 이전 토큰만 참조 ⇒ “GPT의 한계”

[BERT]

- 양방향 문맥

- 사전 학습 단계부터 좌측+우측 문맥 동시에 사용
- 토큰 수준 판별 과제에서 특히 중요

4. 모델 아키텍처 사용 방식

[GPT]

- Transformer 기반 언어 모델
- 생성 중심 구조: 사전 학습과 미세 조정 사이의 역할 = 생성 모델 → 판별 모델로 전이

[BERT]

- Transformer 인코더만 사용
- 사전 학습과 미세 조정 간 아키텍처 거의 동일