Improving Language Understanding by Generative Pre-Training

1. Introduction

본 논문은 자연어 이해 과제에서 **라벨 데이터 부족 문제**를 해결하기 위한 방법으로 **대규모 Text Corpus를 활용한 Generative Pre-Training + Supervised Fine-Tuning** 프레임워크를 제안한다.

→ 기존 NLP 모델들은 각 task 별로 많은 labeled data를 필요로 했으며, 사전학습을 하더라도 단어 임베딩 수준에 국한되는 경우가 많았다.

> 💡 **따라서 저자들은 이러한 한계를 극복하기 위해 언어 모델을 기반으로 universal representation을 학습하고, 이를 최소한의 구조 변형만으로 다양한 task에 적용하는 것을 목표로 한다.**

- Generative Language Model을 사전학습하면 High Level의 의미·문맥 정보를 학습할 수 있음
- Transformer 기반 모델은 장거리 의존성을 효과적으로 포착
- 단일 모델이 다양한 task에서 여러 SOTA 모델을 능가할 수 있음

2. Related Work

논문은 기존 연구를 다음 세 흐름으로 정리한다.

2.1 Semi-supervised Learning in NLP

- Word2Vec, GloVe 등 사전학습 단어 임베딩은 성능 향상을 보였으나 문장·담화 수준의 의미 전달에는 한계 존재

2.2 Unsupervised Pre-training

- LSTM 기반 접근(Dai et al., ULMFiT 등)이 있었으나
  - 장거리 문맥 처리 한계
  - 다양한 Task에 대한 범용성 부족

2.3 Auxiliary Objectives

- 언어 모델링을 auxiliary loss로 사용하는 접근이 제안되었으나 저자들은 사전학습 자체가 이미 풍부한 언어적 지식을 내재화한다고 주장

---

3. Framework

본 논문의 핵심 방법론은 **2-stage 학습 구조**이다.

3.1 Unsupervised Pre-training

- 대규모 unlabeled text에 대해 Language Modeling Objective를 사용

  $\mathcal{L}_1(U) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1})$

- 모델: **Decoder-only Transformer**
  - Masked self-attention
  - Multi-head attention
  - Position embedding + token embedding
- 결과적으로 모델은 **문법, 의미, 패턴**을 내재적으로 학습

3.2 Supervised Fine-tuning

- 사전학습된 Transformer 위에 linear head만 추가

  $\mathcal{L}_2(C) = \sum_{(x,y)} \log P(y|x)$

- 추가로 **Auxiliary LM loss**를 함께 사용

  $\mathcal{L}_3 = \mathcal{L}_2 + \lambda \mathcal{L}_1$

→ 일반화 성능 향상 + 수렴 속도 개선

3.3 Task-specific Input Transformations

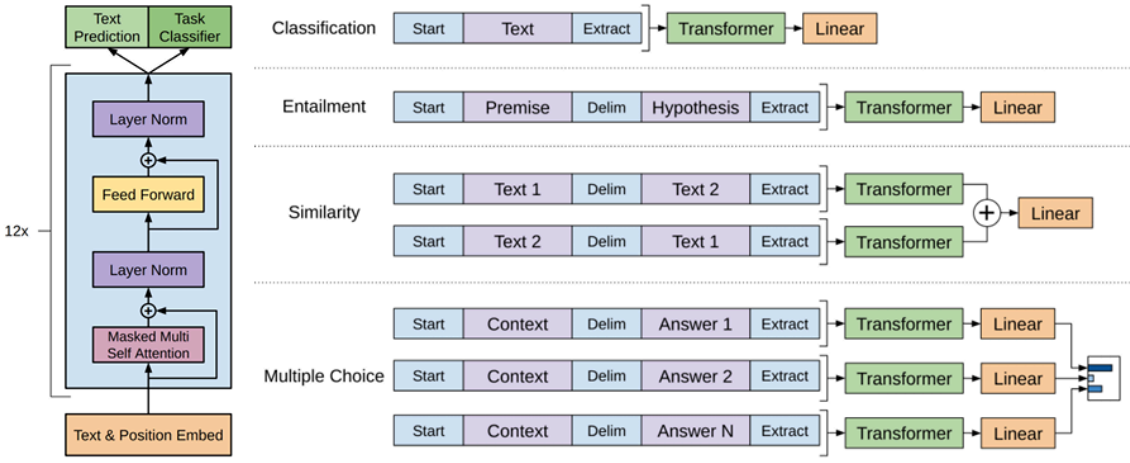
모델 구조를 바꾸지 않고, **입력 표현만 바꾸는 방식**으로 Task를 통일:

- **Textual Entailment**
- **Semantic Similarity**

  → 문장 순서를 바꾼 두 입력을 각각 처리 후 representation 합산
- **QA / Commonsense Reasoning**

  → 후보별 평가

구조적 입력을 **하나의 token sequence**로 변환하여 Transformer에 그대로 투입



---

4. Experiments

4.1 Setup

- **Pre-training dataset**: BooksCorpus
  - 긴 문맥이 존재
- **Model**
  - 12-layer Transformer
  - hidden size 768, head 12
  - BPE vocab 40k

4.2 Results

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

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

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**

Natural Language Inference

- SNLI, MultiNLI, QNLI, SciTail 등에서 SOTA

Question Answering & Commonsense

- Story Cloze Test: **+8.9%**
- RACE: **+5.7%**

  → 긴 문맥 이해 능력 입증

Semantic Similarity & Classification

- STS-B, CoLA에서 **큰 폭의 성능 향상**
- GLUE benchmark 전체 점수 **72.8 (기존 68.9)**

→ 12개 중 9개 태스크에서 SOTA 달성

---

5. Analysis

5.1 Layer Transfer Effect

- 더 많은 Transformer layer를 전이할수록 성능 지속 향상
- 단순 embedding보다 High Representation이 중요

5.2 Zero-shot Behavior

- Fine-tuning 없이도
  - 감성 분석
  - 문법성 판단
  - QA 추론

  어느 정도 수행 가능

→ LM이 이미 Task 관련 지식을 학습했음을 보여줌

5.3 Ablation Study

- Pre-training 제거 시 평균 성능 **–14.8%**
- LSTM 대비 Transformer가 전이 성능 우수
- Auxiliary LM loss는 Large Dataset에서 효과적

---

6. Conclusion

Transformer 기반 Generative Pre-Training이 범용 자연어 이해 모델의 강력한 기반이 될 수 있음

- Task-agnostic한 단일 모델로 다수의 Task 해결
- 최소한의 구조 변경 + 입력 변환만으로 효과적으로 전이

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

1. Introduction

- 언어 모델 사전학습(pre-training)은 다양한 NLP 태스크 성능 향상에 효과적
- 기존 접근 방식
  - **Feature-based**: ELMo
  - **Fine-tuning-based**: OpenAI GPT
- 기존 fine-tuning 방식의 핵심 한계
  - **단방향(unidirectional) 언어 모델**
  - 토큰이 한쪽 문맥만 참조 가능
  - QA, NLI 등 **양방향 문맥이 중요한 태스크에 부적합**

  → 사전학습 단계에서부터 **깊은 양방향 문맥 표현**을 학습할 수 없는 구조

BERT

- Masked Language Model(MLM)을 통해 **양방향 Transformer 사전학습**
- Next Sentence Prediction(NSP)으로 문장 간 관계 학습
- Contribution
  - 깊은 양방향 Transformer 사전학습 최초 제안
  - task-specific 구조 없이 fine-tuning만으로 SOTA 달성
  - 11개 NLP 태스크에서 기존 최고 성능 갱신

2. Related Work

2.1 Unsupervised Feature-based Approaches

- Word embedding
  - Word2Vec, GloVe 등 (정적 임베딩)
- 문맥 기반 표현
  - ELMo
    - 양방향 LM을 **독립적으로 학습**
    - 두 표현을 **concatenation**
- 한계
  - 깊은 양방향 상호작용 부재
  - 사전학습 모델이 downstream task와 분리됨
  - fine-tuning 기반이 아님

2.2 Unsupervised Fine-tuning Approaches

- OpenAI GPT, ULMFiT
  - 사전학습된 LM 전체를 downstream task에 fine-tuning
- 장점
  - task-specific 파라미터 적음
- 한계
  - **Left-to-right 단방향 LM**
  - 문장 전체/문장 쌍 이해에 구조적 제약

2.3 Transfer Learning from Supervised Data

- 대규모 supervised 데이터로 사전학습 후 전이 (NLI, MT, ImageNet 등)
- NLP에서도 효과적이지만 task 범용성에는 한계 존재

---

3. BERT

3.1 Overview

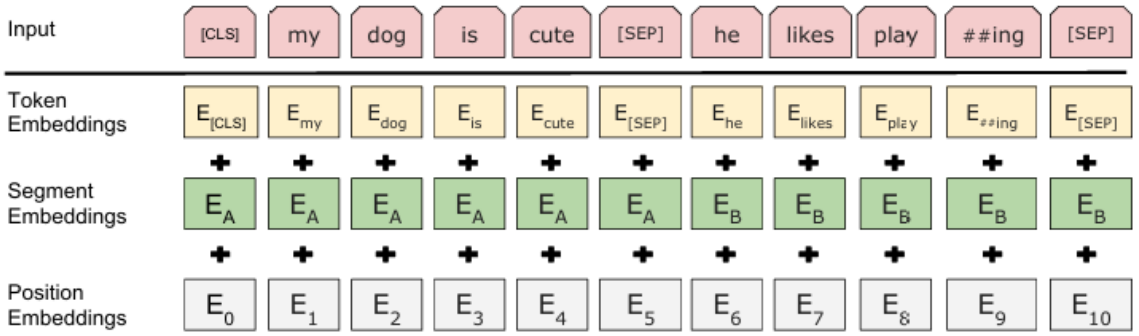
- 두 단계 구성
  1. **Pre-training** (unlabeled data)

2. **Fine-tuning** (labeled downstream task)

- 특징
  - 사전학습과 fine-tuning에서 **동일한 Transformer encoder 구조 사용**
  - 출력 레이어만 task-specific

3.2 Model Architecture

- 기본 구조
  - Transformer **encoder**
  - **Bidirectional self-attention**
- 모델 크기
  - **BERT_BASE**
    - L=12, H=768, A=12, 110M params
  - **BERT_LARGE**
    - L=24, H=1024, A=16, 340M params
- 비교 포인트
  - GPT: 단방향 attention
  - BERT: 모든 레이어에서 양방향 attention



3.3 Input Representation

- 입력 구성 요소

    - [CLS] : 문장 전체 표현 (classification)
    - [SEP] : 문장 구분
    - Segment embedding: 문장 A / B 구분
    - Position embedding
- 단일 문장 / 문장 쌍 모두 동일한 입력 포맷 사용

3.4 Pre-training Tasks

Task 1: Masked Language Model (MLM)

- 전체 토큰 중 **15% 랜덤 선택**
- 선택된 토큰 처리 방식
    - 80%: Mask
    - 10%: 랜덤 토큰
    - 10%: 원래 토큰 유지
- 목적
    - 좌·우 문맥을 모두 활용한 토큰 예측
    - 깊은 양방향 표현 학습 가능

Task 2: Next Sentence Prediction (NSP)

- 문장 A, B 입력
    - 50%: 실제 다음 문장 (IsNext)
    - 50%: 랜덤 문장 (NotNext)
- [CLS] 벡터를 이용해 이진 분류
- 목적
    - 문장 간 관계 이해
    - QA, NLI 성능 향상

3.5 Pre-training Data

- BooksCorpus (800M words)
- English Wikipedia (2.5B words)
- 문장 단위가 아닌 **document-level corpus 사용**

3.6 Fine-tuning

- 모든 파라미터 end-to-end fine-tuning
- 태스크별 출력 방식
  - 문장 분류: CLS → classifier
  - 토큰 예측(QA, NER): token hidden state → output layer
- 특징
  - 계산 비용 낮음
  - 동일한 사전학습 모델로 다양한 태스크 적용 가능

---

4. Experiments

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	**90.5**	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

4.1 GLUE Benchmark

- 모든 GLUE Task에서 SOTA
- BERT_LARGE 평균 성능 가장 높음
- 데이터가 적은 Task일수록 효과 큼

4.2 SQuAD v1.1

- Start / End position prediction 방식

- 단일 모델로 기존 앙상블 모델 성능 넘음

4.3 SQuAD v2.0

- No-answer 질문 처리
- [CLS]를 no-answer span으로 활용
- 기존 최고 성능 대비 큰 폭 향상

4.4 SWAG

- 상식 기반 문장 완성 태스크
- GPT 대비 큰 성능 향상

---

5. Ablation Studies

5.1 Pre-training Task 효과

- NSP 제거
  - MNLI, QNLI, SQuAD 성능 하락
- MLM → LTR LM 변경
  - 전반적인 성능 크게 감소
- BiLSTM 추가해도 bidirectional pre-training 성능 미달

5.2 Model Size 효과

- 모델 크기 증가 → 모든 태스크 성능 향상
- 소규모 데이터셋에서도 큰 모델이 효과적임을 확인

5.3 Feature-based vs Fine-tuning

- BERT는 feature-based 방식에서도 강력
- 그러나 **fine-tuning 방식이 가장 높은 성능** 달성

---

6. Conclusion

- BERT는 깊은 양방향 Transformer 사전학습을 최초로 제안
- 단순한 fine-tuning만으로 광범위한 NLP 태스크에서 SOTA 달성
- 이후 NLP 사전학습 모델들의 **기본 패러다임 정립**