

Improving Language Understanding by Generative Pre-Training



저자

- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever



링크

- https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
-

1. Introduction

- 지도 학습 의존도 완화: 수동으로 라벨링된 데이터의 부족 문제를 해결하기 위해, 라벨 없는 데이터로부터 언어적 정보를 활용하는 대안을 제시함.
- 2단계 훈련 절차
 - Unsupervised pre-training: 라벨 없는 대규모 코퍼스에서 언어 모델링 목적 함수를 사용하여 초기 파라미터를 학습함.
 - Supervised fine-tuning: 학습된 파라미터를 타겟 작업의 지도 학습 목적 함수에 맞춰 적응시킴.
- 트랜스포머의 구조적 이점: 순환 신경망(RNN)에 비해 장기 의존성(long-term dependencies)을 처리하는 데 더 구조화된 메모리를 제공하여, 다양한 작업에서 강건한 전이 성능을 보임.
- 작업별 입력 적응: 모델 구조의 수정을 최소화하기 위해, 구조화된 텍스트 입력을 단일 토큰 시퀀스로 처리하는 순회 스타일의 입력 변형 방식을 활용함.
- 범용적 모델 : 각 작업에 맞춰 설계된 구조 없이도, 하나의 범용 모델이 다양한 자연어 이해 작업에서 기존 SOTA를 경신함.

2. Related Work

- 반지도 학습 (Semi-supervised learning)

- 기존의 단어 수준 피처 전이를 넘어 문맥이 담긴 고차원적 의미 정보를 전이하는 것을 목표로 함.
 - **비지도 사전 학습 (Unsupervised pre-training)**
 - LSTM의 짧은 예측 범위를 극복하기 위해 트랜스포머를 채택하여 더 긴 언어 구조를 학습하고 최적의 모델 초기화 지점을 찾아냄.
 - **보조 훈련 목적 함수 (Auxiliary training objectives)**
 - 미세 조정 단계에서 언어 모델링을 보조적으로 추가하되, 사전 학습만으로도 이미 타겟 작업에 필요한 언어적 자산이 충분히 습득됨을 입증함.
-

3. Framework

두 단계로 구성됨 (대규모 텍스트 코퍼스에서 고용량 언어 모델을 학습 + 모델을 라벨링된 데이터가 있는 판별적 작업에 적용)

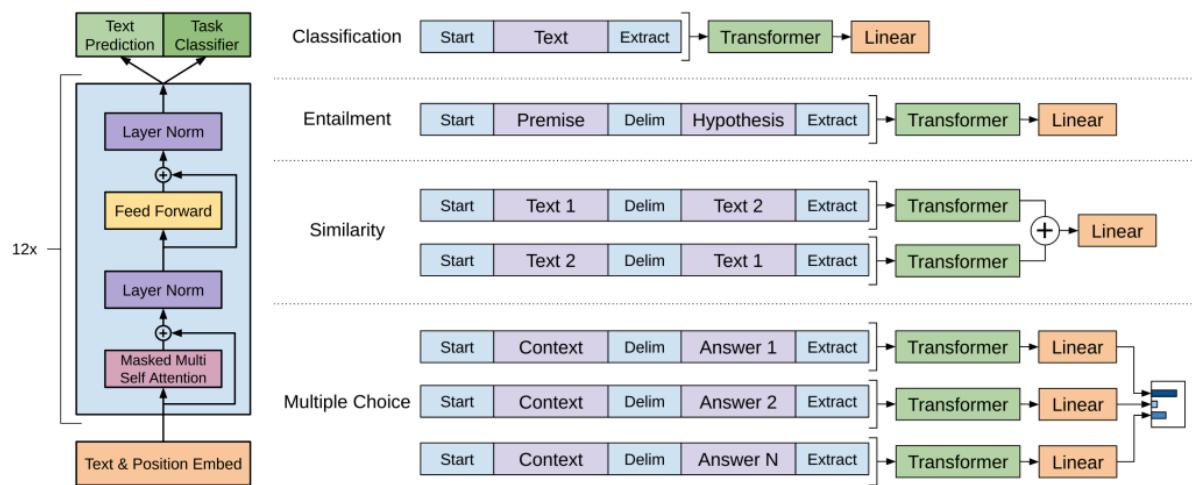
3.1) Unsupervised pre-training

- **언어 모델링 목적 함수 (L_1):** 라벨이 없는 대규모 코퍼스 U 를 사용하여, 앞선 토큰들을 기반으로 다음 토큰이 나타날 확률을 최대화하도록 학습함.
 - $L_1(U) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1})$
 - 문장의 문맥(k 개의 윈도우)을 보고 다음에 올 단어를 맞히는 과정을 통해 언어의 구조를 스스로 깨우침
- **트랜스포머 디코더 구조:** 모델의 뼈대로 트랜스포머의 디코더 블록을 여러 층(n) 쌓아 사용.
 - $h_l = \text{transformerblock}(h_{l-1})$
 - 이전 정보를 순차적으로 처리하는 RNN과 달리, 어텐션 메커니즘을 통해 문장 내 먼 거리의 단어 간 관계(Long-range dependency)를 더 잘 파악함.
- **입력 및 출력 연산:**
 - **입력(h_0):** 단어의 의미(W_e)와 위치 정보(W_p)를 더해 입력 벡터를 만듦.
 - **출력($P(u)$):** 최종 결과물에 소프트맥스를 적용해 어떤 단어가 올지 확률 분포를 생성.

3.2) Supervised fine-tuning

- **작업 적응 및 출력층 추가:** 사전 학습된 모델의 마지막 출력값(h_l^m)에 새로운 선형 출력층(W_y)을 연결하여 정답 y 를 예측함.
 - $P(y|x^1, \dots, x^m) = \text{softmax}(h_l^m W_y)$
- **지도 학습 목적 함수 (L_2):** 라벨링된 데이터셋 C 에서 실제 정답을 맞힐 확률을 최대화하도록 학습함.
 - $L_2(C) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m)$
- **통합 최적화 (L_3):** 미세 조정 시 실제 문제 풀이(L_2)뿐만 아니라 비지도 학습인 언어 모델링(L_1)을 보조 작업으로 함께 수행함
 - $L_3(C) = L_2(C) + \lambda * L_1(C)$

3.3) Task-specific input transformations



- **구조화된 입력의 시퀀스화:** 별도의 복잡한 아키텍처를 추가하는 대신, traversal style 접근법을 통해 다양한 형태의 입력을 모델이 처리할 수 있는 하나의 연속된 시퀀스로 변형함.
- **통합된 입력 포맷:** 모든 입력 시퀀스의 시작에 **Start** 토큰을, 끝에 **Extract** 토큰을 추가하며, 문장 간의 경계에는 **Delimiter \$** 토큰을 삽입하여 구분함.
- **작업별 변형 규칙:**
 - **Classification (분류):** `[Start; Text; Extract]`
 - **Entailment (함의):** `[Start; Premise; Delimiter; Hypothesis; Extract]`
 - **Similarity (유사도):** 두 문장의 순서가 상관없도록 `[Start; Text A; Delimiter; Text B; Extract]` 와 `[Start; Text B; Delimiter; Text A; Extract]` 두 시퀀스를 각각

처리한 후 합산함.

- **Multiple Choice (질의응답/다지선다):** [Start; Context+Question; Delimiter; Answer 1; Extract], [Start; Context+Question; Delimiter; Answer 2; Extract] 등 각 선택지마다 독립적인 시퀀스를 생성함.
 - **아키텍처 일관성 유지:** 이러한 입력 변환 덕분에 작업마다 모델 구조를 바꿀 필요가 없으며, 사전 학습된 지식을 모든 작업에 손실 없이 전이하는 Task Agnostic한 특성을 극대화함.
-

4. Experiments

4.1) Setup

- **사전 학습 데이터 (BooksCorpus):** 다양한 장르의 미출간 도서 7,000권 이상을 사용하여, 문장 단위로 섞이지 않은 긴 연속 텍스트(contiguous text)
- **모델 규격:** 12레이어의 디코더 전용(decoder-only) 트랜스포머를 사용하며, 768차원의 상태 벡터와 12개의 어텐션 헤드, 3072차원의 피드포워드 신경망으로 구성함.
- **최적화 및 스케줄링:** Adam 옵티마이저를 사용하며, 초기 2,000 스텝 동안 학습률을 선형적으로 높이는 Warmup 후 코사인 스케줄에 따라 감소시킴.
- **입력 및 토큰화:** 40,000회의 병합을 거친 BPE(Bytepair Encoding)를 사용하며, 기존 트랜스포머와 달리 사인파 대신 학습된 포지션 임베딩(Learned position embeddings)을 채택함.
- **활성화 함수:** ReLU 대신 GELU(Gaussian Error Linear Unit)를 사용함.
- **미세 조정 설정:** 대부분의 작업에서 3 에폭(Epoch)의 짧은 학습만으로 충분했으며, 보조 언어 모델링의 가중치 λ 는 0.5로 설정함.

4.2) Supervised fine-tuning

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

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

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]	<u>76.5</u>	-	-	-
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

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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

- 추론 및 장거리 문맥 처리 (QA & Reasoning)

- RACE(5.7%↑)와 Story Cloze(8.9%↑)에서 기존 기록을 크게 경신하며, 트랜스포머의 셀프 어텐션 구조가 긴 문맥 속에서 정보를 추출하고 복잡한 추론을 수행하는 데 매우 효과적임을 증명함.

- 문장 간 의미 관계 파악 (Semantic Similarity)

- STS-B와 QQP(4.2%↑) 등 유사도 측정 작업에서 단순한 단어 매칭을 넘어, 문장의 재표현(paraphrasing)이나 부정어 사용, 구문적 모호성을 파악하는 깊이 있는 언어 전이 능력을 보여줌.

- 언어적 타당성 및 범용성 (Classification & GLUE)

- 문법성을 판단하는 CoLA(35.0 → 45.4)에서 점수가 급상승했으며, 이는 사전 학습만으로도 문장의 구조적 적합성을 판별하는 '언어적 편향'을 모델이 스스로 학습했음을 시사함.
- 범용 벤치마크인 GLUE에서 기준 SOTA 대비 3.9점 높은 72.8점을 기록하며, 특정 작업에 특화된 구조 없이도 범용 모델이 더 우수할 수 있음을 입증함.

5. Training

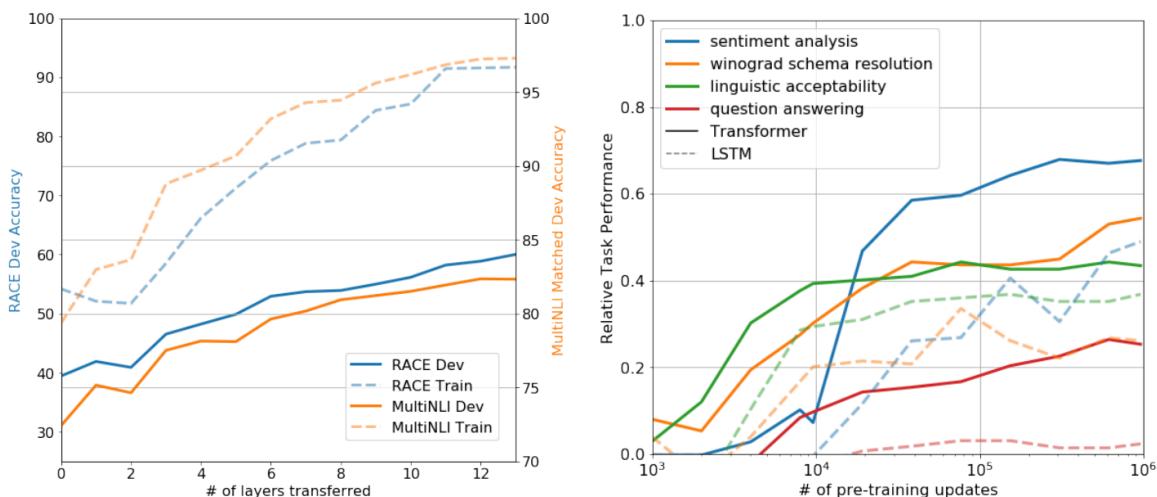


Figure 2: (left) Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. (right) Plot showing the evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess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

- 전이 레이어 수의 영향 (Impact of transferred layers)

- **레이어별 기여도:** 임베딩만 전이하는 것보다 레이어를 더 많이 전이할수록 성능이 계단식으로 상승하며, MultiNLI 기준 전체 레이어 전이 시 최대 9%의 추가 성능 향상을 기록함.
- **심층 특징 학습:** 이는 사전 학습된 모델의 각 레이어가 문법, 의미, 추론 등 타겟 작업 해결에 필요한 서로 다른 유용한 기능들을 단계적으로 학습했음을 의미함.

- 제로샷 성능 (Zero-shot Behaviors)

- **내재적 지식 습득:** 미세 조정 없이도 언어 모델링 학습만으로 QA나 감성 분석 능력이 꾸준히 향상됨. 이는 모델이 다음 단어를 맞추기 위해 자연스럽게 다양한 언어 작업 수행 능력을 내재화함을 시사함.
- **구조적 우위:** LSTM보다 트랜스포머가 제로샷 성능에서 훨씬 낮은 변동성과 높은 효율을 보였으며, 이는 트랜스포머의 Inductive bias이 전이 학습에 더 유리함을 입증함.

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)	STS-B (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

• 어블레이션 연구 (Ablation studies)

- 사전 학습 없이 직접 지도 학습을 했을 때 성능이 14.8%나 급감하여, 모델 성능의 핵심이 사전 학습에 있음을 명확히 함.
- 동일 조건에서 LSTM으로 교체 시 평균 **5.6점**이 하락하여 트랜스포머 구조의 우월성을 재확인함.
- 미세 조정 시 언어 모델링을 병행하는 것은 특히 대규모 데이터셋에서 일반화 성능을 높이는 데 기여함.

6. Conclusion

Generative Pre-training과 Discriminative Fine-tuning을 결합하여, 구조 변경이 필요 없는 단일 모델로 강력한 범용 프레임워크를 제시함. 트랜스포머 구조로 긴 문맥의 데이터를 사전 학습함으로써 모델이 세상 지식과 장거리 의존성을 스스로 습득하고 이를 다양한 과제에 성공적으로 전이할 수 있음을 증명함.

결과적으로 12개 중 9개 작업에서 SOTA를 달성하며 비지도 학습의 실질적 기여를 입증했고, 트랜스포머와 연속적 데이터가 이 전략의 최적 조합이라는 통찰을 제공하며 향후 연구의 중요한 이정표를 세움.

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



저자

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language)



링크

- <https://arxiv.org/pdf/1810.04805>
-

1. Introduction

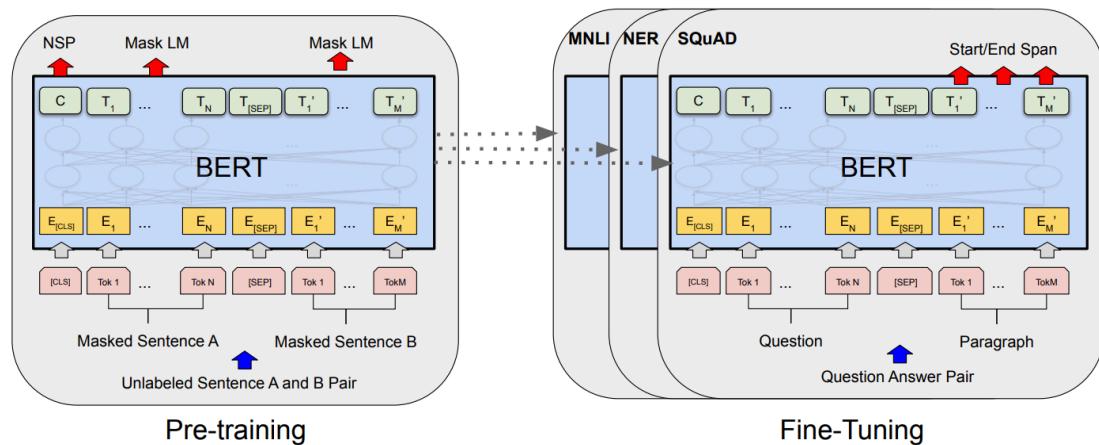
- 기존 전략의 한계: ELMo(피처 기반)와 GPT-1(미세 조정 기반) 모두 단방향 (Unidirectional) 언어 모델링을 사용하므로, 토큰의 양쪽 문맥을 동시에 파악해야 하는 작업(특히 질의응답)에서 성능이 제한됨.
- BERT의 해결책 (MLM): 빈칸 채우기 형식의 Masked Language Model을 도입하여 모든 층에서 왼쪽과 오른쪽 문맥을 동시에 결합하는 깊은 양방향 표현을 학습함.
- 추가 학습 목표 (NSP): 두 문장 간의 관계를 이해하기 위해 다음 문장 예측(Next Sentence Prediction) 작업을 병행함.
- 논문의 기여:
 1. 언어 표현에서 양방향 사전 학습의 중요성을 증명함.
 2. 복잡한 작업별 아키텍처 설계의 필요성을 줄임 (단순한 미세 조정만으로 충분).
 3. 11개의 NLP 작업에서 새로운 SOTA를 달성함.

2. Related Work

- **비지도 피처 기반 접근법 (Feature-based)**
 - **전통적 임베딩:** Word2Vec, GloVe 등 단어 수준의 임베딩은 현대 NLP의 필수 요소가 됨.

- **문장 및 단락 임베딩:** 단어에서 더 나아가 문장이나 단락 단위의 표현을 학습하는 연구로 확장됨.
- **ELMo:** 왼쪽에서 오른쪽($L \rightarrow R$)으로 가는 모델과 오른쪽에서 왼쪽($R \rightarrow L$)으로 가는 모델을 각각 학습한 뒤, 두 표현을 단순 결합(Concatenation)하여 문맥을 파악함. 양쪽을 보긴 하지만 깊은 방식의 동시 양방향 학습은 아님.
- **비지도 미세 조정 접근법 (Fine-tuning)**
 - 라벨이 없는 데이터로 인코더를 먼저 학습시킨 뒤, Downstream task에 맞춰 모든 파라미터를 업데이트하는 방식임. 밑바닥부터 새로 학습해야 할 파라미터가 매우 적어 효율적.
 - **GPT-1:** 이 방식을 통해 GLUE 벤치마크 등에서 SOTA를 달성했으나, **단방향 ($L \rightarrow R$)** 구조라는 한계가 있음.
- **지도 데이터로부터의 전이 학습**
 - 자연어 추론(NLI)이나 기계 번역과 같은 거대한 지도 학습 데이터셋을 이용한 전이 학습 연구도 효과적임이 입증됨. 컴퓨터 비전 분야에서 **ImageNet**으로 사전 학습 된 모델을 미세 조정하여 사용하는 것과 유사한 원리임.

3. BERT



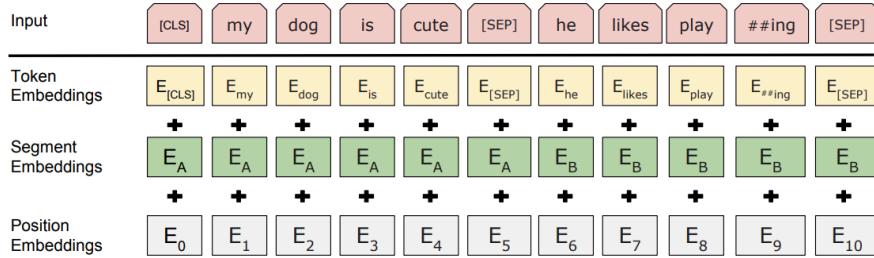


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- 비지도 데이터로 학습하는 사전 학습(Pre-training)과 라벨링된 데이터로 특정 작업을 배우는 미세 조정(Fine-tuning)의 두 단계로 진행함.
- 사전 학습 모델과 하위 작업용 모델 사이의 구조적 차이를 최소화하여, 어떤 작업이든 동일한 설계를 유지함.
- **모델 아키텍처:** 트랜스포머의 **양방향 인코더**를 사용하며, 크기에 따라 두 가지 버전을 제시함.
 - **BERT-BASE:** L=12, H=768, A=12 (GPT-1과 비교를 위해 동일 크기로 설정, 파라미터 1.1억 개).
 - **BERT-LARGE:** L=24, H=1024, A=16 (파라미터 3.4억 개).
- **입력 표현 (Input Representations):** 단일 문장뿐만 아니라 문장 쌍(예: 질문과 답변)도 하나의 시퀀스로 표현함.
 - **토큰화:** 3만 개의 단어 사전을 가진 **WordPiece 임베딩**을 사용함.
 - **특수 토큰:** 모든 시퀀스의 시작에 분류용 **[CLS]** 를, 문장 사이에는 구분용 **[SEP]** 토큰을 넣음.
- **임베딩 합산 (Embedding Construction):** 세 가지 임베딩을 더해서(단순 합) 최종 입력을 만듦.
 1. **Token Embeddings:** WordPiece로 쪼개진 단어 정보.
 2. **Segment Embeddings:** 각 토큰이 문장 A에 속하는지 B에 속하는지 나타냄.
 3. **Position Embeddings:** 문장 내 토큰의 위치 정보.

3.1) Pre-training BERT

- 기존의 단방향 모델과 달리 두 가지 비지도 학습 작업(**MLM, NSP**)을 통해 깊은 양방향 표현을 배움.
- **Task #1: Masked LM (MLM)**

- **개념:** 입력 토큰의 15%를 무작위로 선택해 마스킹하고, 주변 문맥만으로 원래 단어를 맞히는 방식임.
 - **데이터 불일치 해결:** [MASK] 토큰은 미세 조정(Fine-tuning) 때 나타나지 않으므로, 선택된 15% 중 80%는 [MASK], 10%는 랜덤 단어, 10%는 원래 단어를 그대로 두어 모델이 항상 올바른 문맥을 유지하게 유도함.
 - **효과:** 모델이 단순히 [MASK] 만 찾는 게 아니라, 모든 토큰에 대해 깊은 문맥 정보를 파악하게 만듦.
- **Task #2: Next Sentence Prediction (NSP)**
 - **개념:** 두 문장(A, B) 사이의 관계를 이해하기 위해, B가 A 다음에 올 문장인지 (**IsNext**) 아니면 상관없는 랜덤 문장인지(**NotNext**)를 **50:50 비율**로 섞어 예측하게 함.
 - **효과:** 질문-답변(QA)이나 자연어 추론(NLI)처럼 문장 간의 관계 파악이 중요한 작업에서 성능을 대폭 끌어올림.
 - **사전 학습 데이터:**
 - **BooksCorpus**(8억 단어)와 **English Wikipedia**(25억 단어)를 사용함.
 - 문장 단위로 섞인 데이터 대신 **문서 단위(Document-level)** 데이터를 사용하여 길고 연속적인 문맥을 학습함.

3.2) Fine-tuning BERT

- 트랜스포머의 셀프 어텐션 덕분에 입력과 출력만 적절히 교체하면 단일 문장이나 문장 쌍 작업을 모두 쉽게 처리할 수 있음. 기존 모델들이 두 문장을 각각 인코딩한 뒤 나중에 결합했던 것과 달리, BERT는 두 문장을 하나로 합쳐서 입력함. 이는 셀프 어텐션 과정에서 두 문장 사이의 **양방향 교차 어텐션(Bidirectional cross attention)** 효과를 자동으로 포함하게 만듦.
- **입력:** 질문-본문, 전제-가설 등 다양한 구조를 문장 A-B 형식에 맞춰 넣기만 하면 모든 작업을 수행할 수 있음.
- **출력:** 분류 작업은 [CLS] 토큰을, 질의응답이나 태깅 같은 토큰 작업은 **각 단어 위치의 출력값을 활용함.**
- 사전 학습된 파라미터를 그대로 사용하여, 단일 GPU로도 몇 시간 안에 학습이 끝날 만큼 비용이 저렴함.

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
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.⁸ BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
BERT _{BASE} (Single)	80.8	88.5	-	-
BERT _{LARGE} (Single)	84.1	90.9	-	-
BERT _{LARGE} (Ensemble)	85.8	91.8	-	-
BERT _{LARGE} (Sgl.+TriviaQA)	84.2	91.1	85.1	91.8
BERT _{LARGE} (Ens.+TriviaQA)	86.2	92.2	87.4	93.2

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-	-	71.4	74.4
Ours				
BERT _{LARGE} (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
BERT _{BASE}	81.6	-
BERT _{LARGE}	86.6	86.3
Human (expert) [†]	-	85.0
Human (5 annotations) [†]	-	88.0

Table 4: SWAG Dev and Test accuracies. [†]Human performance is measured with 100 samples, as reported in the SWAG paper.

- 문장 분류, 유사도 등 9개 하위 작업을 포함하는 GLUE 벤치마크에서 80.5%를 기록하며 기존 SOTA 대비 **7.7%p**라는 기록적인 상승을 보여줌.
- SQuAD (질의응답):** 질문과 본문을 문장 A, B로 넣어 미세 조정한 결과, SQuAD v1.1 과 v2.0 모두에서 기존 최고 성능을 가볍게 경신하며 복잡한 문맥 이해 능력을 증명함.

- **SWAG (상식 추론)**: 주어진 문장 뒤에 올 자연스러운 문장을 고르는 작업에서도 인간의 성능에 근접하는 압도적인 결과를 기록함.
 - 모든 작업에서 **BERT-LARGE**가 BERT-BASE보다 훨씬 뛰어난 성능을 보였으며, 이는 모델의 파라미터가 커질수록 사전 학습의 효과가 더 극대화됨을 시사함.
-

5. Ablation Studies

5.1) Effect of Pre-training Tasks

- NSP를 제거하자 문장 간 논리적 연결이 중요한 QNLI, MNLI, SQuAD 성적이 크게 하락하며 문장 관계 이해 능력이 필수적임을 증명함.
- 단방향(LTR) 모델은 모든 과제에서 MLM보다 뒤쳐졌고, 특히 뒷문맥 참조가 필수적인 SQuAD(질의응답)에서 점수가 폭락함.
- 두 방향을 따로 학습해 합치는(ELMo 방식) 것보다, 모든 레이어에서 좌우 문맥을 동시에 융합하는 깊은 양방향 학습이 훨씬 강력한 표현력을 가짐.

5.2) Effect of Model Size

- 레이어 수(L), 숨겨진 유닛 수(H), 어텐션 헤드(A)를 늘릴수록 모든 데이터셋에서 정확도가 꾸준히 상승함.
- 보통 데이터가 적으면 큰 모델은 과적합(Overfitting)되기 쉬운데, BERT는 MRPC(3,600) 같은 아주 작은 작업에서도 모델이 커질수록 성능이 대폭 향상됨.
- 기존 연구들(ELMo 등)은 모델을 일정 크기 이상 키웠을 때 이득이 없다고 보고했으나, BERT는 사전 학습 후 직접 미세 조정하는 방식을 통해 초대형 모델(BERT-LARGE, 3.4억 개 파라미터)의 표현력을 소규모 작업까지 전이하는 데 성공함.
- 충분히 사전 학습된 모델이라면, 하위 작업의 데이터가 매우 적더라도 모델이 크고 정교 할수록 더 유리한 결과를 얻을 수 있음을 처음으로 확실히 입증함.

5.3) Feature-based Approach with BERT

- 비교 목적: 모든 파라미터를 업데이트하는 '미세 조정' 대신, BERT를 고정된 특징 추출기(Feature Extractor)로 썼을 때의 성능을 검증함.
 - 모든 작업을 트랜스포머 구조로만 풀 수 없을 때 유리하며, 미리 계산된 임베딩을 재사용하므로 연산 비용이 매우 저렴함.
- 실험 방법 (NER 작업): BERT의 파라미터는 고정(Freeze)시킨 채, 특정 레이어에서 나온 값들을 추출하여 별도의 BiLSTM 모델의 입력으로 넣어 학습함.

- 가장 좋은 성능은 최상위 4개 레이어를 결합(Concat)하여 사용했을 때 나왔음 (Test F1 96.1). 이는 전체 모델을 미세 조정한 성능(96.4)과 0.3점밖에 차이가 나지 않음.
 - BERT는 특정 작업에 맞춰 전체를 튜닝하는 미세 조정 모델로서도 훌륭하지만, 상황에 따라 임베딩 값만 뽑아서 쓰는 피처 기반 모델로서도 충분히 강력한 범용성을 가짐을 증명함.
-

6. Conclusion

본 연구는 대규모 코퍼스를 활용한 깊은 양방향(Deep Bidirectional) 사전 학습이 언어 표현 능력을 비약적으로 향상한다는 사실을 입증함. 특히 기존의 단방향 모델(GPT-1)이나 얇은 결합 방식(ELMo)이 가졌던 구조적 한계를 MLM과 NSP라는 새로운 학습 목표를 통해 성공적으로 극복함.

결과적으로 특정 작업에 특화된 복잡한 아키텍처 설계 없이도, 사전 학습된 모델 위에 간단한 출력층 하나만 추가하는 방식으로 문장 수준과 토큰 수준의 다양한 NLP 작업에서 압도적인 성능을 기록함. 이는 풍부한 비지도 학습 데이터가 자연어 이해 시스템의 핵심임을 확고히 했으며, 누구나 강력한 사전 학습 모델을 하위 작업에 쉽게 전이할 수 있는 새로운 연구 패러다임을 확립함.