AiDA Lab.
Artificial Intelligence & Data Analysis Laboratory

GPT

Seokhwan Yang

사전 학습 모델(Pre-trained Model)

- **사전 학습(Pre-training)**
  - **기존에 자비어(Xavier) 등 임의의 값으로 초기화하던 모델의 가중치들을 다른 문제(task)에 학습시킨 가중치들로 초기화하는 방법**
  - **예시**
    - 텍스트 유사도 예측 모델을 만들기 전 감정 분석 문제를 학습한 모델의 가중치를 활용해 텍스트 유사도 모델의 가중치로 활용하는 방법
      → 즉, 감정 분석 문제를 학습하면서 얻은 언어에 대한 이해를 학습한 후 그 정보를 유사도 문제를 학습하는 데 활용하는 방식

사전 학습 모델(Pre-trained Model)

- **하위 문제(downstream task)**
  - 사전 학습한 가중치를 활용해 학습하고자 하는 본 문제
  - 예시
    - 앞 예시에서는
    - 사전 학습 문제(pre-train task): 사전 학습한 모델인 감정 분석 문제
    - 하위 문제: 사전 학습된 가중치를 활용해 본격적으로 학습하고자 하는 문제인 텍스트 유사도 문제
- **사전 학습 모델**
  - **사전 학습을 통해 학습이 완료된 가중치가 저장되어 있는 모델**

사전 학습 모델(Pre-trained Model)

- **사전 학습은 왜 필요한가?**
  - **예시**
    - QQP(Quora Questions Pairs) 데이터 셋을 이용하여 텍스트 유사도를 측정하고자 함
    - 그런데 QQP 데이터의 대부분이 잘못되었다는 것이 밝혀진다면?

      → 잘못된 데이터로 학습하는 것은 모델의 성능을 떨어뜨림
    - 잘못된 데이터를 모두 제거한 후, 남은 데이터만 이용하여 학습하기로 결정

      → 그런데 잘못된 데이터의 비율이 95%, 즉 40만개의 데이터 중 38만개가 잘못되었다면?
    - 남은 2만개의 데이터는 학습을 진행하기에 매우 부족한 것으로 확인

      → 사전 학습 기법 적용으로 해결 가능

사전 학습 모델(Pre-trained Model)

- **사전 학습 모델을 이용하지 않고 학습하는 과정**
  - **데이터 셋 분석 및 전처리**
    - 데이터 셋을 EDA(Exploratory Data Analysis, 탐색적 자료 분석)과정을 거쳐 분석
    - 토크나이징, 임베딩 등 전처리 수행
  - **모델 설계, 구현 및 생성**
    - 이때 생성된 모델은 임의의 값으로 초기화된 가중치를 가지고 있음
  - **모델 학습**
    - 유사도 데이터를 활용하여 학습 수행

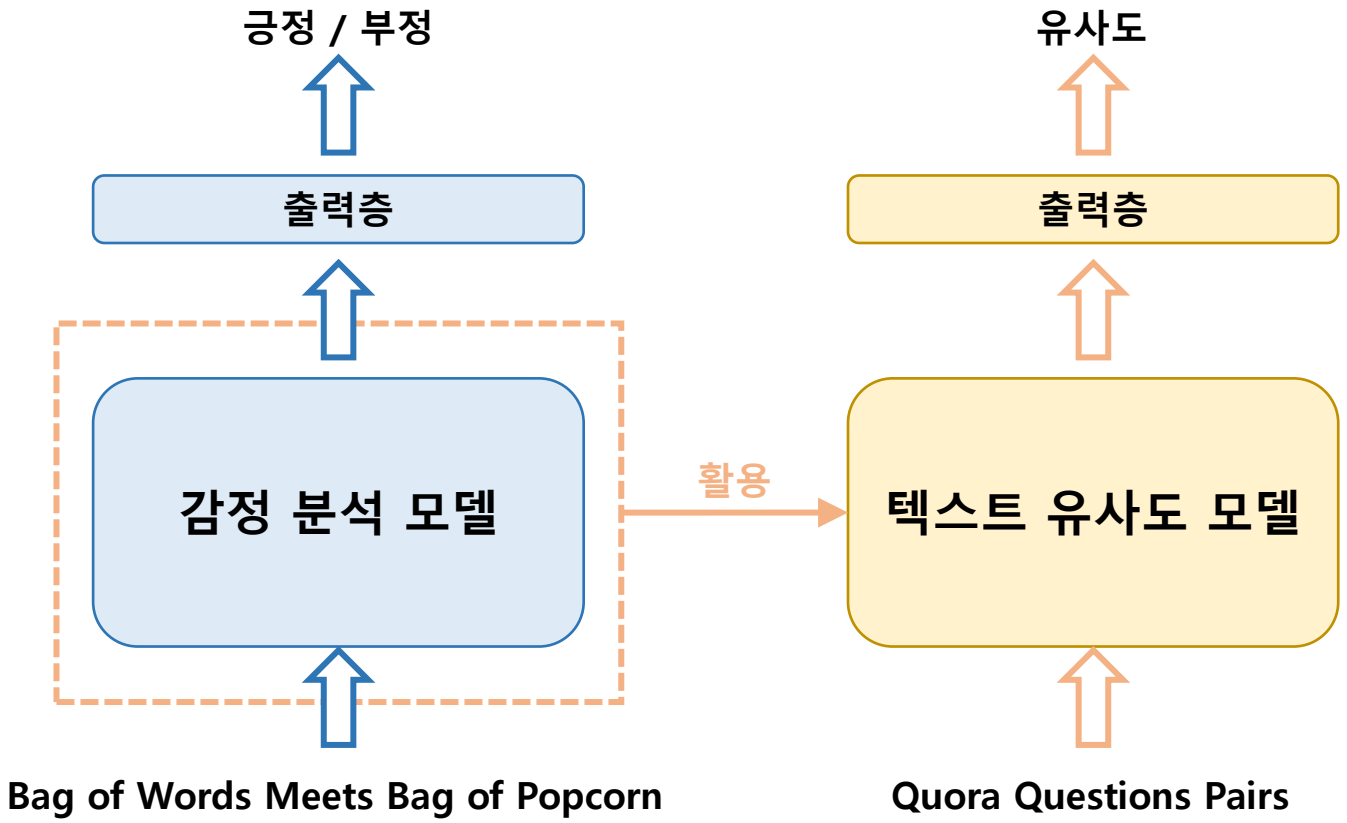
사전 학습 모델(Pre-trained Model)

- **사전 학습 모델을 이용한 학습 과정**
  - **적절한 데이터 셋에 대하여 잘 학습된 사전 학습 모델 확보**
    - 이때 모델은 데이터 셋에 대하여 학습이 완료된 가중치를 가지고 있음
  - **모델 학습**
    - 가중치 초기화 등의 과정을 수행하지 않고
    - 기존에 학습이 완료된 가중치를 초기값으로 사용함
    - 현재 해결하고자 하는 문제에 대한 유사도 데이터를 활용하여 학습 수행
    - 단, 사전 학습 모델의 최종 출력 값을 뽑는 가중치 층은 제외

      → 우리가 해결하고자 하는 문제와 관계없는 형태의 데이터가 포함되어 있기 때문

사전 학습 모델(Pre-trained Model)

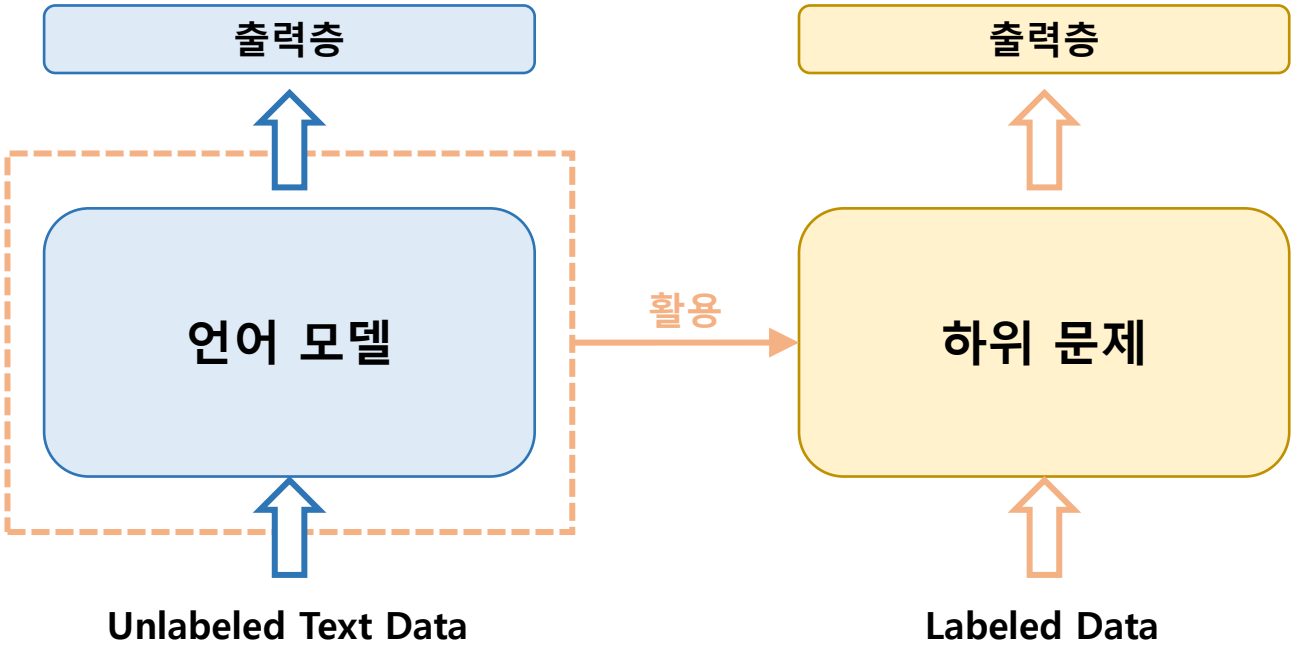
- 사전 학습된 가중치의 활용



※ Bag of Words Meets Bag of Popcorn Dataset (from. Kaggle)
https://www.kaggle.com/c/word2vec-nlp-tutorial

사전 학습 모델(Pre-trained Model)

- 최근의 자연어 처리 연구 추세 → 사전 학습 활용 → 언어모델 활용
- 언어 모델을 사전학습한 모델



사전 학습 모델(Pre-trained Model)

- **왜 언어 모델을 많이 활용하는가?**
  - **사전 학습은 어떤 문제에도 적용이 가능함**
  - **그렇다면 왜 대부분의 자연어 처리 관련 연구는 언어 모델을 활용하는가?**
    - 대부분의 언어 처리 문제는 Label 이 주어지지 않는 비지도 학습 문제에 포함됨
      → 데이터에 제약이 없고 언어에 대한 전반적인 이해를 사전 학습하는 언어 모델이 효율적
    - 또한 대부분의 경우, 하위 문제 모델의 성능도 향상시킴
    - 언어 모델은 대규모의 데이터를 활용한 사전 학습을 통해 언어에 대한 전반적인 이해 (Natural Language Understanding, NLU)를 하게 됨
      → 이렇게 학습된 지식을 기반으로 하위 문제에 대한 성능을 향상시킴

사전 학습 모델(Pre-trained Model)

- **사전 학습한 가중치를 활용하는 방법**
  - **특징 기반(Feature-Based) 방법**
    - 사전 학습된 특징을 하위 문제의 모델에 부가적인 특징으로 활용하는 방법
    - 대표사례: word2vec

      → 학습한 임베딩 특징을 우리가 학습하고자 하는 모델의 임베딩 특징으로 활용
  - **미세 조정(Fine-Tuning)**
    - 사전 학습한 모든 가중치와 더불어 하위 문제를 위한 최소한의 가중치를 추가해서 모델을 추가로 학습(미세 조정)하는 방법
    - 대표사례: 감정 분석 문제에 사전 학습 시킨 가중치와 더불어 텍스트 유사도를 위한 부가적인 가중치를 추가하여 텍스트 유사도 문제를 학습하는 것

사전 학습 모델(Pre-trained Model)

- 트랜스포머 모델 이후로
  - 대부분의 자연어 처리 연구에서는
  - 트랜스포머 모델을 기반으로 하는 비지도 사전 학습을 통해 학습한 많은 가중치들을 활용하여
  - 다양한 자연어 처리 모델을 미세 조정하는 방법
- 이 각광받고 있음

사전 학습 모델(Pre-trained Model)

- **대표적인 트렌스포머 기반 사전 학습 모델**
  - **GPT (Generative Pre-trained Transformer)**
    - OpenAI에서 발표한 최초의 트랜스포머 기반 사전 학습 언어 모델(2018)
    - 현재 GPT-1, GPT-2를 거쳐 GPT-3까지 발표됨
    - https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf
  - **BERT (Bidirectional Transformers for Language Understanding**
    - Google에서 발표한 트랜스포머 기반 사전 학습 언어 모델(2018)
    - https://arxiv.org/pdf/1810.04805.pdf

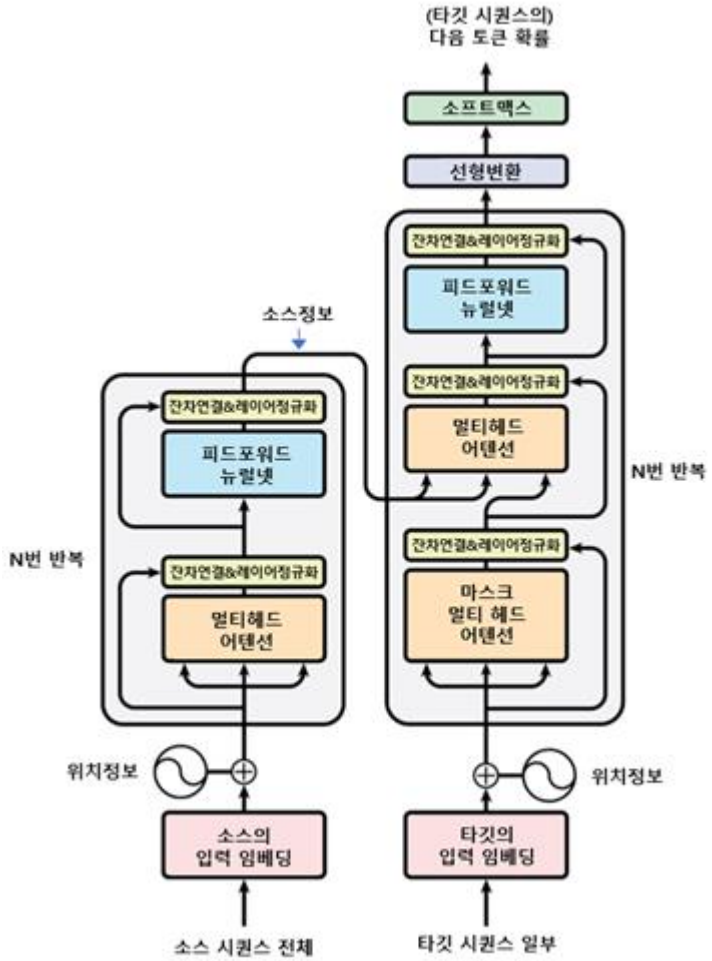
GPT와 BERT 비교

- GPT
  - 이전 단어들이 주어졌을 때 다음 단어가 무엇인지 예측하는 과정에서 사전학습을 수행하는 언어 모델
  - 트랜스포머의 디코더만 사용
  - 문장 왼쪽부터 오른쪽으로 순차적으로 계산하는 일방향성 모델
- BERT
  - 문장 중간에 빈칸을 만들고, 해당 빈칸에 어떤 단어가 적절할지 예측하는 과정에서 사전학습을 수행하는 마스크 언어 모델
  - 트랜스포머의 인코더만 사용
  - 빈칸의 앞뒤 문맥을 모두 살필 수 있는 양방향성 모델

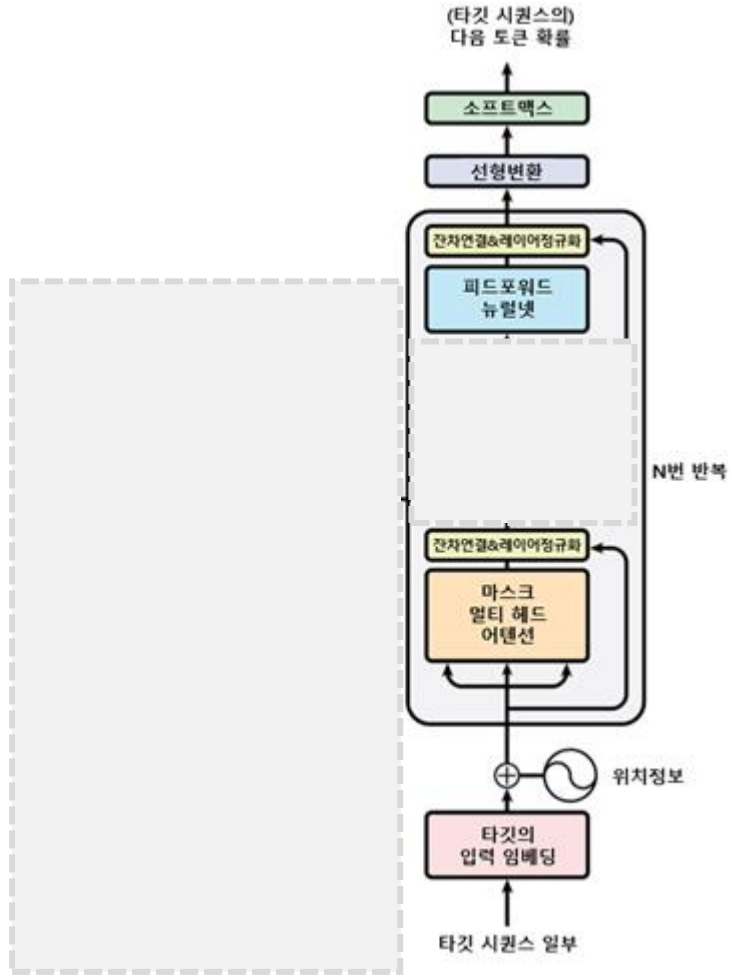
GPT(Generative Pre-trained Transformer)

- GPT1
  - 매우 큰 자연어 처리 데이터를 활용해 비지도 학습으로 사전 학습을 수행한 후, 학습된 가중치를 활용해 해결하고자 하는 문제에 미세조정을 적용하는 방식의 언어 모델
  - 모델의 기반 구조는 트랜스포머 모델임
  - 트랜스포머의 디코더 구조만 사용함
  - 순방향 마스크 어텐션 적용

GPT(Generative Pre-trained Transformer)



트랜스포머 아키텍처



GPT 모델

GPT(Generative Pre-trained Transformer)

Transformer.Decoder → GPT1

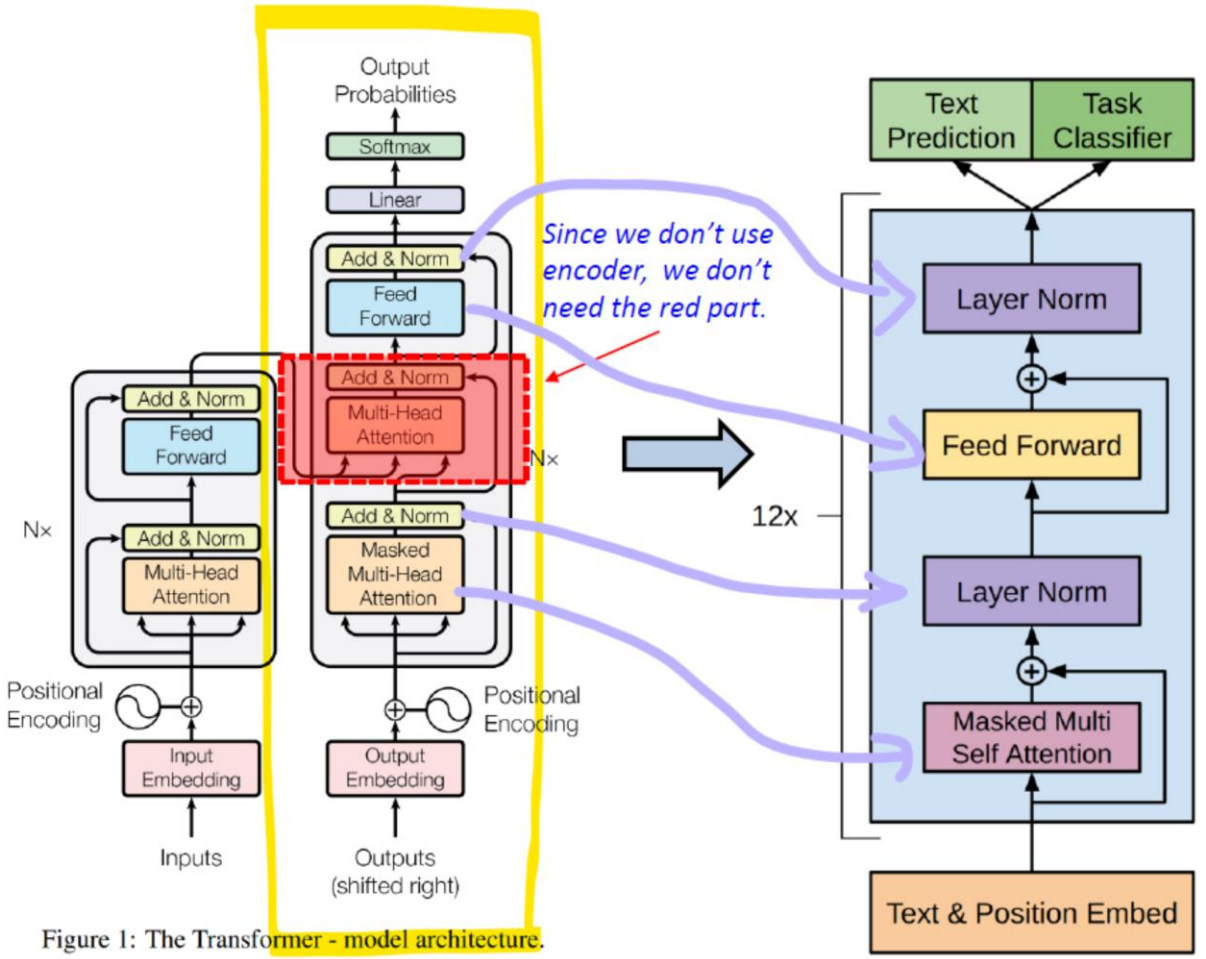


Figure 1: The Transformer - model architecture.

GPT(Generative Pre-trained Transformer)

- **하나의 사전 학습 방식(전통적인 언어 모델 방식) 사용**

  → 앞의 단어들을 활용하여 해당 단어를 예측하는 방식

  - 예시
    - "나는 학교에 간다 " 라는 문장을 활용하여 총 3번 학습 진행

Input	Label
"<START>"	"나는"
"<START>", "나는"	"학교에"
"<START>", "나는", "학교에"	"간다"

- **Label이 따로 존재하지 않아도 학습 진행 가능 → 비지도 학습**

GPT(Generative Pre-trained Transformer)

- **실제 문제를 대상으로 학습을 진행할 때도 언어 모델을 함께 학습**
  - 비교 모델인 BERT의 경우, 사전 학습에서만 언어 모델의 손실 값(Loss)을 사용해서 학습하지만
  - GPT-1에서는 본 학습 시에도 실제로 학습해야 하는 문제의 손실 값과 더불어 언어 모델의 손실 값 또한 학습함

GPT(Generative Pre-trained Transformer)

- **GPT-1 논문의 내용을 기반으로 살펴보면**
  - 대부분의 딥러닝 task는
    - 수동으로 Labeling된 많은 양의 Data가 필요 → 수동 작업이 뒤따라야 하므로 풍부하지 않음
    - 그러나 이런 데이터가 부족한 분야에는 적용 가능성이 제한됨
    - 따라서 Unlabeled Data 를 사용하기 위한 방안이 필요함
  - Unlabeled Data
    - 시간 소모나 가격측면에서 좋은 대안을 제공
    - 성능 상승에 효과적인 Representation(표현방식)을 배울 수 있음
    - Unlabeled Data는 풍부함

GPT(Generative Pre-trained Transformer)

- Unlabeled Text 활용의 어려움
  - 단어 수준(word level) 이상의 정보를 얻기가 쉽지 않음
  - 어떤 데이터가 활용하기 용이한지, 또 유용한 표현을 학습하는데 효과적인지 불분명함
  - 학습된 표현을 target task에 효과적으로 전달하는 일관성 있는 방안, 구조가 없음
  - 이러한 불확실성이 Semi-Supervised Learning(반지도학습) 접근법을 어렵게 만듦

GPT(Generative Pre-trained Transformer)

- GPT-1의 제안
  - 접근법: 비지도 사전 학습과 지도학습 미세조정(파인튜닝) 을 결합한 반지도학습적 접근
  - 목표: 넓은 범위에서의 작업에 약간의 조정만으로도 전이할 수 있는 범용 표현의 학습
  - 2단계 구조를 적용
    - Unlabeled Data 상에서 동작하는 언어 모델 개체를 사용하여 사전학습을 수행한 후,
    - 지도학습에 해당하는 Target Task를 대상으로 미세조정을 적용함
  - 언어 모델 개체에는 트랜스포머 구조를 적용
    - 텍스트의 장기 의존성에 강한 결과를 보여줌
    - 기존 RNN 등에 비해 구조화된 메모리를 사용할 수 있게 함

→ 지도학습 모델의 일반화 성능을 개선하고, 학습의 수렴을 빠르게 해 주는 장점을 보임

GPT(Generative Pre-trained Transformer)

- 기존 트랜스포머 모델의 단점
  - 태스크가 가진 특정 아키텍처의 목표에 따라 학습 수행
  - 이는 상당한 양의 태스크 특정 커스터마이징이 필요하며, 전이 학습을 어렵게 함

→ 사전 학습 모델이 잘 동작할 수 있도록 입력 구조를 변환시키는 방법을 적용

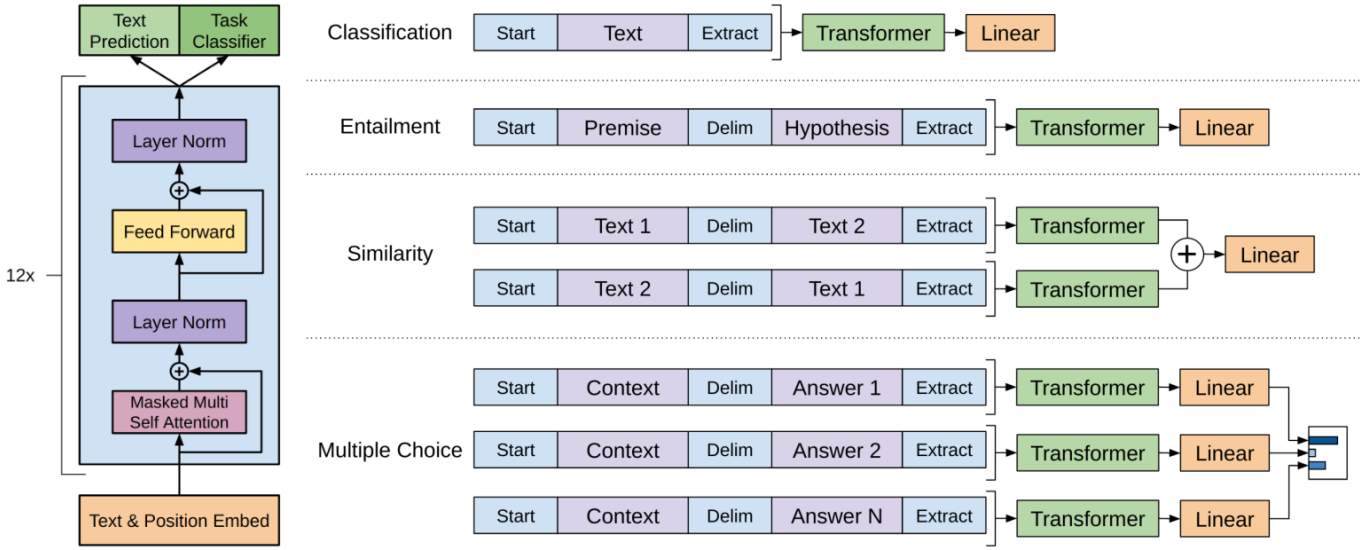


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

GPT(Generative Pre-trained Transformer)

- 여러 부문에서 최고의 성능을 나타냄

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)	-	-	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**

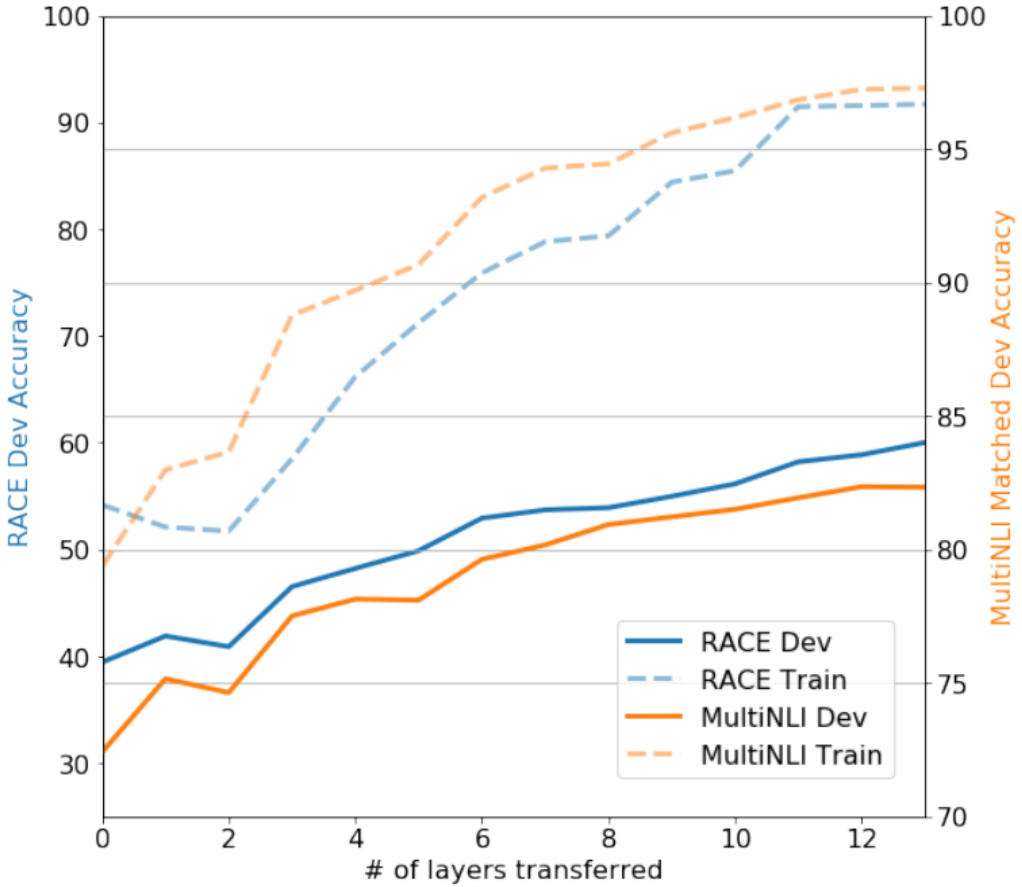
GPT(Generative Pre-trained Transformer)

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)

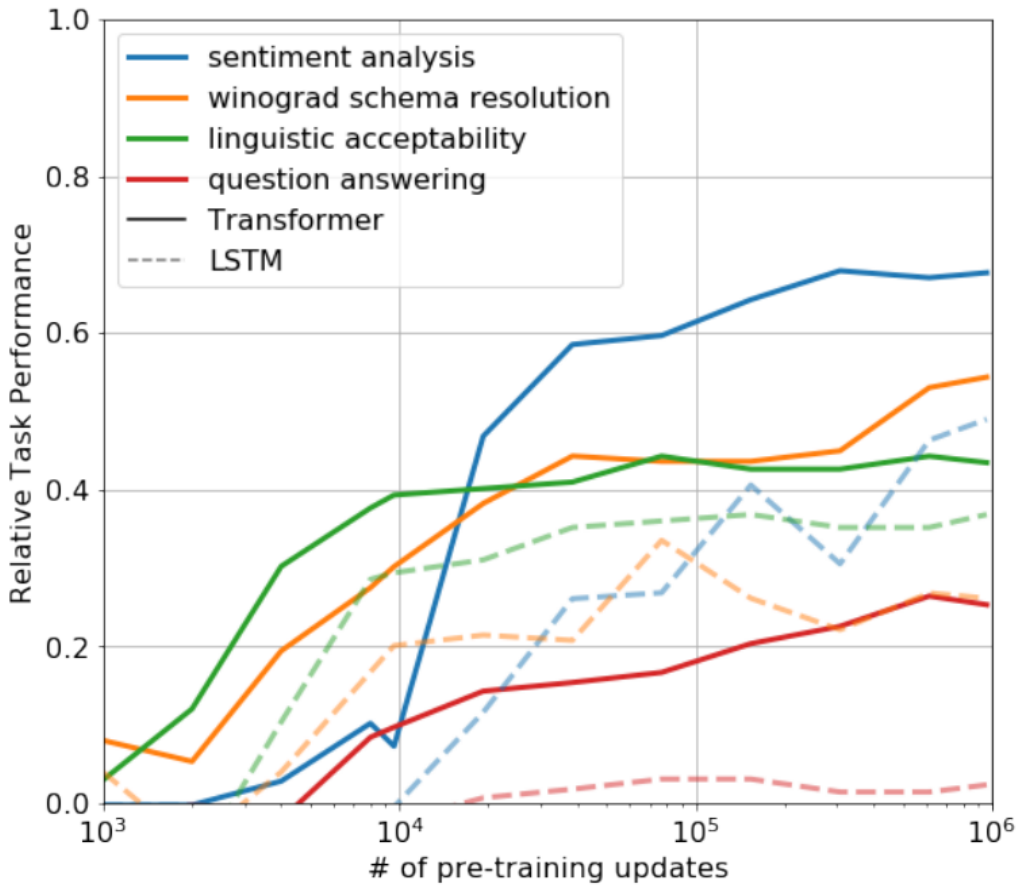
Method	**Classification**		**Semantic Similarity**			**GLUE**
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task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

GPT(Generative Pre-trained Transformer)

- 전이 시 Layer의 개수에 따른 성능 영향 비교



- 사전 학습에 의한 모델 갱신 횟수에 따른 LSTM과 트랜스포머의 비교



GPT(Generative Pre-trained Transformer)

- **GPT-2**
  - **GPT-1의 성능을 향상시킨 모델**
  - **모델 구조**
    - GPT1과 거의 동일함
    - 차이점
      - 기존의 디코더에서 각 레이어 직후의 Residual Connection(잔차 연결)과 함께 적용되던 Layer Normalization(층 정규화)이 각 부분 블록의 입력 쪽으로 이동함
      - 마지막 셀프 어텐션 레이어 이후에 층 정규화가 적용됨

GPT(Generative Pre-trained Transformer)

- **학습 데이터**
  - 기존의 다양한 사전 학습 방법론
    - 사전 학습 데이터의 크기: 한 영역(Domain)의 텍스트를 사용
      - 예: 뉴스 기사 텍스트만 사용, 위키피디아 텍스트만 사용 등
  - GPT-2
    - 사전 학습 데이터의 크기: 다양한 영역의 텍스트를 활용하여 사전 학습 진행

    → 모델이 좀 더 다양한 문맥과 영역의 글을 이해할 수 있게 함

GPT(Generative Pre-trained Transformer)

- **모델의 크기**
  - GPT-1
    - 총 12개의 Layers
    - 총 117만개의 가중치
  - GPT-2
    - 총 48개의 Layers
    - 총 1,542만개

GPT(Generative Pre-trained Transformer)

- **모델의 입력**
  - GPT-1
    - 텍스트를 특정 단위(예: 문자 단위, 단어 단위 등)로 나눠서 모델의 입력으로 사용
  - GPT-2
    - BPE(Byte Pair Encoding) 방식을 사용해 텍스트를 나눠서 모델의 입력으로 사용
    - 글자와 문자 사이의 적절한 단위를 나누어 줌으로써 높은 성능을 발휘함

GPT(Generative Pre-trained Transformer)

- **GPT-2 논문의 내용을 기반으로 살펴보면**
  - https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

  - 기존의 머신러닝 시스템
    - 대규모 dataset, large model, 지도 학습으로 구성됨
    - 학습한 작업에서는 좋은 성능을 보이지만
    - 데이터에 대하여 민감하고 데이터 분포나 사소한 변경 등에 의해서 망가지기 쉬운 시스템
    - 즉, 일반적이지 않고 좁은 전문가(narrow expert)임
    - 우리는 Training dataset을 생성하고, label 할 필요가 없는 일반적인(범용적인) 시스템을 원함

GPT(Generative Pre-trained Transformer)

- 기존의 머신러닝 시스템의 접근 방식
  - 원하는 작업에 대한 dataset of training을 수집
  - 이러한 동작을 모방하도록 훈련
  - 독립적이고 동일한 분포를 가지는 예제로 테스트 → Narrow expert에게 좋은 결과 보장
  - 접근 방식에 대한 단점
    - 단일 도메인 데이터 세트에 대한 단일 태스크 학습 수행 → 취약한 일반화에 대한 주요 원인
    - 광범위한 도메인과 작업에 성능 측정 필요
    - 사전학습 및 지도 미세조정을 활용한 구조 → 여전히 지도학습이 요구됨

GPT(Generative Pre-trained Transformer)

- GPT-2의 제안
  - 기존에는 pre-training과 supervised fine-tuning의 결합으로 만들어졌던 두 개의 작업을 연결
  - 기존에 사용해 오던 전이(Transfer) 방법은 계속 유지
  - 지도 학습이 없는 상태로 만들어진다면 일반 상식 추론 등의 다양하게 범용적으로 사용할 수 있을 것
  - 언어 모델에 대하여
    - 매개변수나 모델 아키텍처 수정없이 사용할 수 있는 zero-shot 방식 적용
    - 모델이 바로 하위 작업에 적용됨
    - Zero-shot 설정으로 인하여 범용성 있는 언어 모델 능력 향상이 가능해 짐

GPT(Generative Pre-trained Transformer)

- 핵심은 언어 모델이다.
  - 일련의 $symbol$ $(s_1, s_2, \ldots, s_n)$으로 구성된 예제 $(x_1, x_2, \ldots, x_n)$ 에서 결과 추정
  - Single task 학습은 $P(output \mid Input)$을 추정하는 확률 framework로 표현
  - 범용 시스템은 여러 다른 과제들을 수행할 수 있어야 하므로 입력에 task를 함께 표현

GPT(Generative Pre-trained Transformer)

- 학습 데이터 셋
  - 이전의 작업은 주로 news article, wikipedia 사용 → Single Domain Data
  - GPT-2에서는
    - 다양한 도메인과 컨텍스트, 주제를 가진 대규모의 데이터 셋 수집
    - 일반적인 크롤링에 의한 데이터 수집 → 데이터 품질에 문제가 있을 수 있음
    - Web Text 사용
      - 인간에 의해 수동으로 필터링 된 웹페이지 수집
      - Reddit(소셜 뉴스 웹사이트)에서 3Karma 이상의 데이터 수집
        → Heuristic indicator로 생각할 수 있음(Interesting, educational of just funny)
    - 4500만 링크 포함
    - 중복 제거, Wikipedia document 제거 등 전처리 수행
    - 40GB, 800만 개 이상 문서

GPT(Generative Pre-trained Transformer)

- Byte Pair Encoding 활용
  - 글자(byte)와 단어의 중간 단위를 사용할 수 있음
  - Subword 분리 알고리즘 중 하나로 OOV(Out Of Vocabulary) 문제를 해결
    - Subword들을 활용하기 때문에 OOV와 신조어 같은 단어에 강점이 있음
- 모델
  - Transformer decoder를 활용
  - 기본적으로 GPT-1과 동일
    - Layer Normalization이 각 sub block의 input으로 이동
    - layer normalization이 마지막 self-attention block 이후에 추가
    - 모델 깊이에 따른 residual path 초기화 방법 변경
    - 사전 개수 5만 여개로 확장
    - Context size가 최대 1024개의 token으로 늘어남
    - Batch size 512로 증가

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

GPT(Generative Pre-trained Transformer)

- 언어 모델로서
  - GPT-2의 강점
    - Byte 수준에서 동작
    - 손실이 있는 전처리 또는 토큰화 불필요
    - 모든 언어 모델에 대한 벤치마크 평가 가능
  - WebText 언어 모델에 따른 dataset의 log 확률을 계산하는 방식으로 통일
  - <UNK>은 400억 byte 중 26번 밖에 나타나지 않음
  - Zero shot으로 8개중 7개에서 SOTA(최상의 결과) 달성

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

GPT(Generative Pre-trained Transformer)

- 성능평가
  - 다양한 성능평가 및 비교를 수행함
  - 일부 분야에서 좋은 성적을 보이고, 다양한 분야에서 여러 유의미한 데이터와 결과 획득
    - Children's Book Test: (ACC)89% → 93%
    - LAMBADA: (PPL)99.8 → 8.6 대폭 향상, (ACC)19% → 52.66%

      (PPL: Test data에 대해 빠르게 식으로 계산되는 간단한 평가 방법. 점수가 낮을 수록 성능이 뛰어남)
    - The Conversation Question Answering Dataset(CoQA): 4개 중 3개의 base line model을 능가(수동으로 수집한 127,000개의 QA 쌍을 사용하지 않고도 능가함)
  - 그러나 전반적으로 썩 뛰어난 결과를 보여주지는 못함
- 일부 분야에서 좋은 성과를 내고 있으나 더욱 개선할 필요성을 보임

GPT(Generative Pre-trained Transformer)

- **GPT-3**
  - **2020년 5월 공개**
  - **총 1,750억 개의 가중치 보유**
  - **기존의 사전 학습-미세 조정 방법론의 구조적 한계 지적, 새로운 방법론 제시**
    - 사전 학습-미세 조정 방법의 가장 큰 한계는 학습된 모델이 특정 문제에 국한된다는 점
      - 각 하위 문제를 해결하기 위하여 미세 조정 과정이 필요할 뿐만 아니라
      - 미세 조정을 위한 하위 문제의 데이터셋이 추가적으로 필요함
      - 일반적으로 이러한 데이터셋의 경우 각 문제에 대하여 수십만 개의 데이터를 요구함

GPT(Generative Pre-trained Transformer)

- 사전 학습-미세 조정 방법론이 가지는 문제점의 이유
  - 기존 방법론의 경우,
    - 하나의 새로운 문제를 해결하려면 그 문제를 위한 정답이 주어진 매우 많은 데이터 셋을 필요로 함 → 사전 학습된 언어 모델의 활용도 감소, 데이터셋 구축 어려움
    - 학습 데이터의 분포가 좁을수록(다양하지 않을수록) 더욱 높은 성능을 보임
      - 하위 문제의 성능 향상에는 도움이 되지만 좀 더 국한된 문제에서만 좋은 성능을 보이는 한계점을 만듦
    - 사람과 비교했을 때, 사람은 새로운 언어적인 문제(감정 분석 등)를 해결하기 위해 수많은 정답을 가진 데이터를 필요로 하지 않고 간단한 몇 가지 가이드만 있으면 어느 정도 새로운 문제를 해결할 수 있음
  - 새로운 접근법의 필요성 대두 → 메타 학습(Meta Learning) 방법론 제안

GPT(Generative Pre-trained Transformer)

- 메타 학습 방법론
  - 사전 학습 과정에서 학습된 다양한 언어적인 능력 및 패턴을 인식하는 능력만을 활용해서 새로운 문제에 적용하는 방법
  - 방대한 데이터로 가중치를 사전 학습하고, 학습된 모델의 능력을 활용해서 특정 문제에 적용 및 예측하는 방법
  - 예시
    - 한국의 수도는 어디입니까?

      → 사전 학습 단계에서 취득한 방대한 자연어 데이터가 가진 정보를 활용하여 대답 생성
  - 기존 방법론과의 차이점
    - 기존 방법론: 사전 학습+미세 조정 후 새로운 문제에 특정된 가중치로 업데이트
    - 메타 학습 방법론: 사전 학습 과정에서 학습된 정보만을 활용하여 문제 해결

GPT(Generative Pre-trained Transformer)

- 메타 학습 방법론의 종류
  - 문제를 해결하기 위한 예시를 몇 개 사용했는지에 따라 다음과 같이 나뉨
    - 제로샷 러닝 (Zero-shot Learning): 0개의 예시 활용
    - 원샷 러닝 (One-shot Learning): 1개의 예시 활용
    - 퓨샷 러닝 (Few-shot Learning): n개의 예시 활용
      - 문제를 해결하기 위해 n개의 예시를 언어 모델에 제공하고
      - 그 예시를 활용해서 문제를 해결함
      - 기존의 미세조정 방법론과 비슷할 수 있지만
        - 미세조정에서는 추가 데이터를 활용해 모델 자체를 추가 학습하지만
        - 퓨샷 러닝에서는 모델을 추가 학습하지 않고 문제를 해결하기 위해 단순히 예시만 활용한다는 점이 다름

GPT(Generative Pre-trained Transformer)

- **GPT-3의 한계점**
  - **낮은 효율성**
    - 1,750억개의 매개변수 → 인간이 평생 보는 정보보다 많은 데이터를 학습해야 함
  - **현실세계의 물리적 상식 부족**
    - "치즈를 냉장고 안에 넣으면 녹을까?" 라는 질문에 "그렇다"라고 대답
    - 세상을 글로만 학습했기 때문. 현실에서 직접 겪어봐야 알 수 있는 매우 당연한 상식을 학습할 기회가 적었음

GPT(Generative Pre-trained Transformer)

- **모든 분야에서 뛰어난 것은 아니다.**
  - 아직까지는 대부분 태스크에서 사람보다 떨어진 성능을 보임
  - 주어진 태스크마다 성능차이가 심함
    - 두가지 이상의 복합연산 능력이 떨어지고
    - 태스크를 수행하기 위해 주어진 데이터가 적을수록 성능이 크게 떨어지는 경향을 보임
- **학습에 사용된 예제를 외운 것인지 실제 추론한 것인지 구분하기 어렵다.**
- **새로운 정보를 수용하기 어렵다. 한마디로 "기억력"이 없다.**
  - 학습된 정보를 토대로 입력 값에 대해 출력 값을 내보낼 수는 있지만, 사람처럼 기억력이라 부를 만한 것이 없다.(그러나 다른 AI도 마찬가지)

GPT(Generative Pre-trained Transformer)

- **방대한 양의 텍스트를 통해 다음 단어를 예측하는 방식으로 학습됨**
  - 주어진 단어에 대해 통계적으로 가장 어울리는 다음 단어를 생성하는 것뿐이며 이해하는 것은 아니라는 비판(이것 역시 다른 AI도 마찬가지)

GPT를 공격하는 사람들이 GPT의 한계점이라고 표현하는 내용중 일부는

GPT 만이 아닌 전체적인 AI 기술에 해당하는 한계임(그냥 깎아내리기 목적?)

그래도 중요한 것은...

"인간은 다음 단어를 예측하는 방법으로 언어를 학습하지 않았다"는 것

참고 자료

- 텐서플로 2와 머신러닝으로 시작하는 자연어 처리 (전창욱, 최태균, 조중현, 신성진 저 | 위키북스)
- Do it! BERT와 GPT로 배우는 자연어 처리 (이기창 저 | 이지스퍼블리싱)
- 김기현의 자연어 처리 딥러닝 캠프-파이토치 편 (김기현 저 | 한빛미디어)
- https://lsjsj92.tistory.com/617 , https://lsjsj92.tistory.com/620