



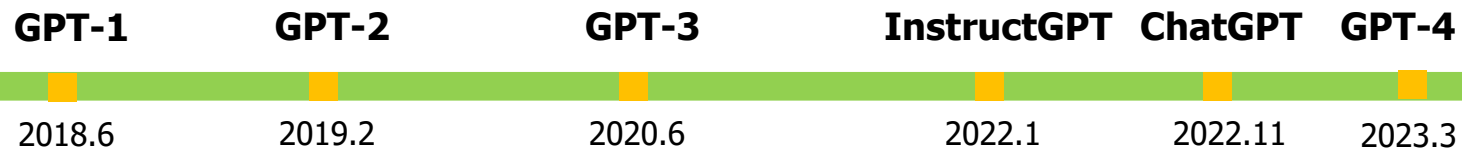
# GPT Models

---

Sang Yup Lee

# GPT Timeline

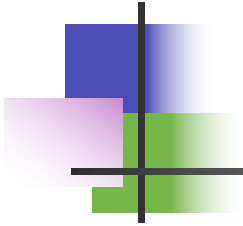
## ■ GPT Timeline



- ChatGPT는 InstructGPT와 유사
  - 따라서 여기서는 InstructGPT를 중심으로 설명

# Summary of GPT-1, 2, and 3

	GPT	GPT-2	GPT-3
당대 배경	<ul style="list-style-type: none"><li>• Labeled data로 학습</li><li>• NLP task 전반에 대한 일반화 모델 부재</li><li>• Transformer 등장</li></ul>	<ul style="list-style-type: none"><li>• NLP task 전반에 SOTA 성능을 보이는 BERT 모델 등장 (사전학습+미세조정)</li></ul>	
성과	작은 미세조정만으로 NLP task 전반에 효과적인 전이학습을 할 수 있는 semi-supervised learning 언어모델	대용량 데이터로 대용량 모델 학습을 하면 비지도 사전학습 모델만으로 추가 미세조정 없이 다양한 task에 멀티태스킹이 가능한 언어모델 구현 가능성	다양한 task에 대해 자연어 패턴처럼 몇 개의 예제만 주면 (few-shot) 추가 미세조정 없이 동일한 형식으로 task를 수행해내는 언어모델
학습 방법	Unlabeled 데이터로 사전학습 모델 생성 후, labeled 데이터로 특정 task에 맞게 fine-tuning 학습	자체 대용량 데이터셋으로 대용량 모델로 사전학습 (모델 구조는 GPT 사전학습 모델과 거의 동일)	대용량 데이터셋으로 초거대 모델 사전학습 (모델 구조는 GPT-2 모델과 동일)
학습 데이터셋	<ul style="list-style-type: none"><li>• BookCorpus (다양한 장르의 출간되지 않은 책 7,000여 권 이상)</li><li>• 약 2.5GB</li></ul>	<ul style="list-style-type: none"><li>• 웹 상의 데이터를 활용한 새로운 dataset (WebText) 구성</li><li>• 약 40GB (vs BERT 16GB)</li></ul>	<ul style="list-style-type: none"><li>• 5개의 대용량 corpora (Common Crawl, WebText2, Books1, Books2 and Wikipedia)</li><li>• 3,000억개 토큰 (vs BERT 3.3억개 토큰)</li></ul>
파라미터 수	1.17억	15억	1,750억



# GPT-1



# GPT-1

---

## ■ 개요

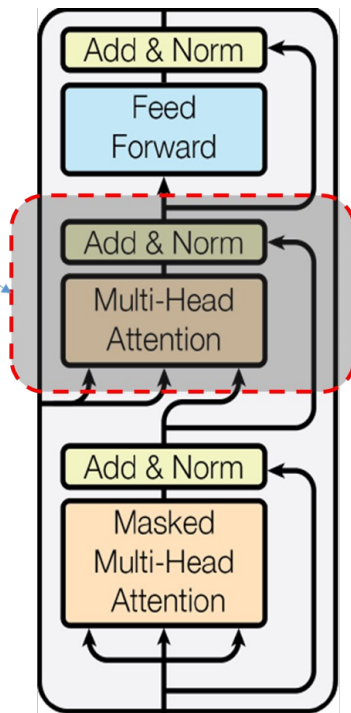
- OpenAI에서 2018년 6월에 발표한 모형 (참고: BERT는 2018.10에 발표)
- BERT와 달리 GPT는 트랜스포머의 디코더 부분 사용
- 준지도학습(semi-supervised) 학습 방법
  - GPT-1 논문에서는 비지도 학습을 이용한 사전학습 방법과 지도 학습을 이용한 미세 조정 방법을 결합한 방법 제안
  - 이러한 방법의 주된 목적은 비지도 사전 학습을 통해서 단어가 갖는 언어적인 특성을 배우고, 그렇게 습득된 결과를 미세 조정 통해 새로운 작업에 적용할 수 있는 모형을 만드는 것
- 해당 논문에서는 제안된 모형을 자연어 추론, 질의·응답, 의미적 유사도, 문서 분류 등의 문제에 대해서 평가
  - 대부분의 작업에 대해서 기존의 SOTA 모형들 보다 좋은 성능을 보임

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

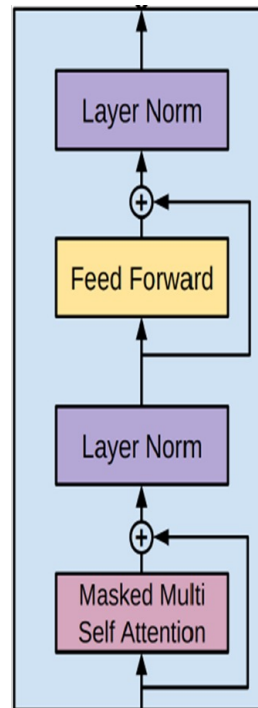
# GPT-1

## ■ 모형의 구조: 트랜스포머의 디코더 사용

원래의 트랜스포머 디코더 블록에는 존재하지만 GPT-1의 디코더 블록에서 사용되지 않은 부분



트랜스포머 디코더 블록



GPT-1 디코더 블록

- 디코더 블록의 수 = 12
- 임베딩 벡터의 차원 = 768
- 멀티-헤드 어텐션에서 사용된  
헤드의 수 = 12개
- 위치 기반 완전연결층이 갖는  
노드의 수 = 3,072

# GPT-1

- GPT-1의 언어 모형이 작동하는 방식의 예





# GPT-1

---

## ■ 학습

- 준지도학습 (semi-supervised) (두 단계로 구성)
  - 단계1: 비지도 사전 학습 (unsupervised pre-training)
  - 단계2: 지도학습 기반 미세 조정 (supervised fine-tuning)
- 단계1: 비지도 사전 학습
  - 입력된 텍스트 데이터에 존재하는 이전 단어들의 정보를 이용해서 다음에 나올 단어가 무엇인지를 예측하는 방식으로 학습이 진행 ⇒ 언어 모델을 비지도 학습 방식을 이용해서 대용량의 데이터를 적용하여 사전 학습 수행
  - 사전학습에서 사용된 데이터
    - BookCorpus 데이터셋 사용: 출간되지 않은 7천 권이 넘는 책
    - 대안적으로 사용될 수 있는 Word Benchmark 데이터셋 대신 BookCorpus를 사용했는데, 주된 이유는 BookCorpus 데이터셋이 더 긴 시퀀스 데이터로 구성되어 있어 멀리 떨어져 있는 단어들 간의 연결 관계를 파악하는데 더 적합하기 때문





# GPT-1

---

## ■ 학습

### ■ 단계1: 비지도 사전 학습 (cont'd)

#### ■ 하이퍼파라미터

- 옵티마이저: Adam, 최대 학습률:  $2.5e-4$ ,
- 입력 시퀀스 길이: 512, 미니 배치 크기: 64, 전체 에포크의 수: 100
- 가중치 파라미터는  $N(0, 0.02)$ 의 분산 크기가 고정된 정규 분포를 사용하여 초기화
- 10%의 드롭아웃을 적용
- L2 규제화와 유사한 방법의 규제화를 진행
- 활성화 함수: GELU
- 위치 임베딩: 학습을 통해 결정
- **토큰화: Byte Pair Encoding**



# GPT-1

---

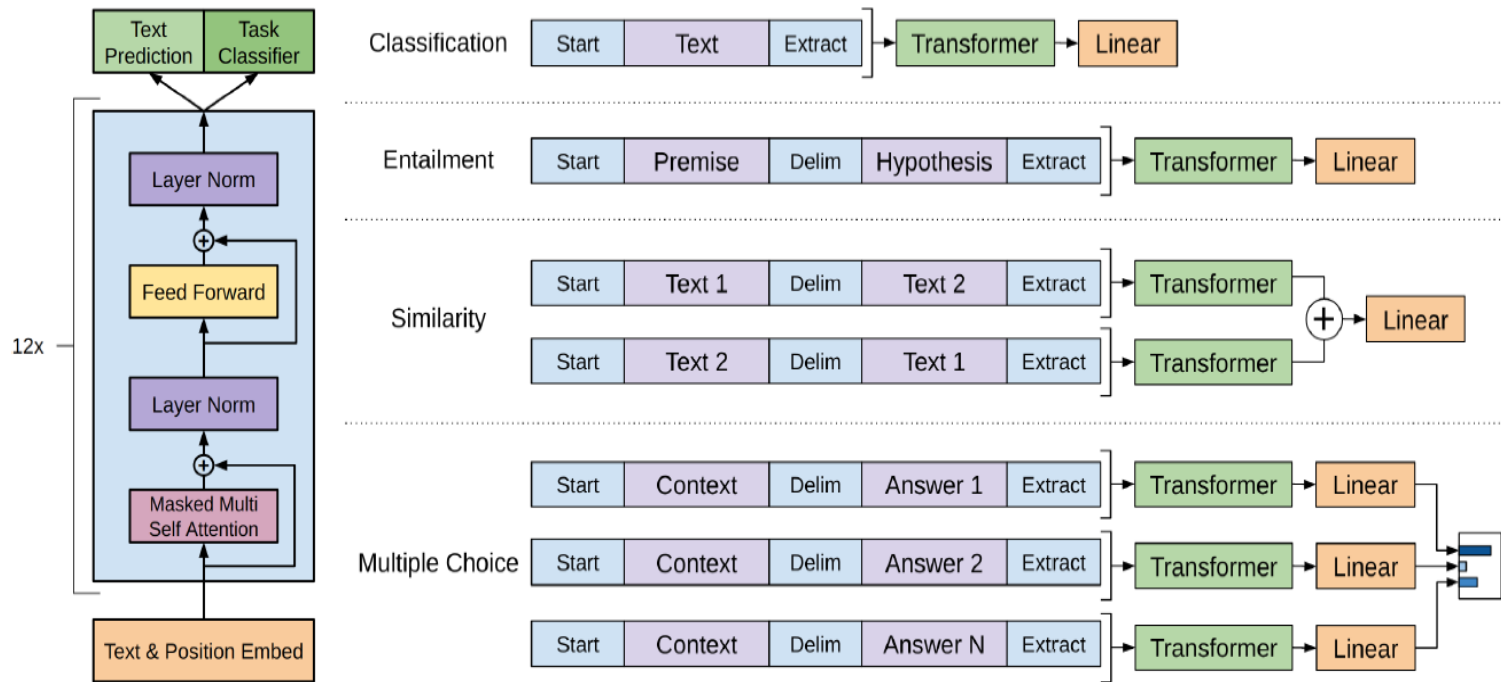
- 지도학습 기반 미세조정 (fine-tuning)
  - $\mathcal{C}$ : 다운스트림 작업에 대한 정답 데이터
  - 정답 예측
    - 입력 데이터에 대해 마지막 디코더 블록에서 출력하는 은닉 상태 벡터 ( $h_l^m$ )를 소프트맥스 함수의 입력값으로 사용
    - $P(y|x^1, \dots, x^m) = \text{softmax}(h_l^m W_y)$
    - 이를 이용해 비용함수 계산  $\Rightarrow L_2(\mathcal{C})$
  - 최종 비용함수

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

$L_1(\mathcal{C})$ : 데이터  $\mathcal{C}$ 에 대한 언어 모형의 비용함수  
논문에서는  $\lambda = 0.5$ 로 설정

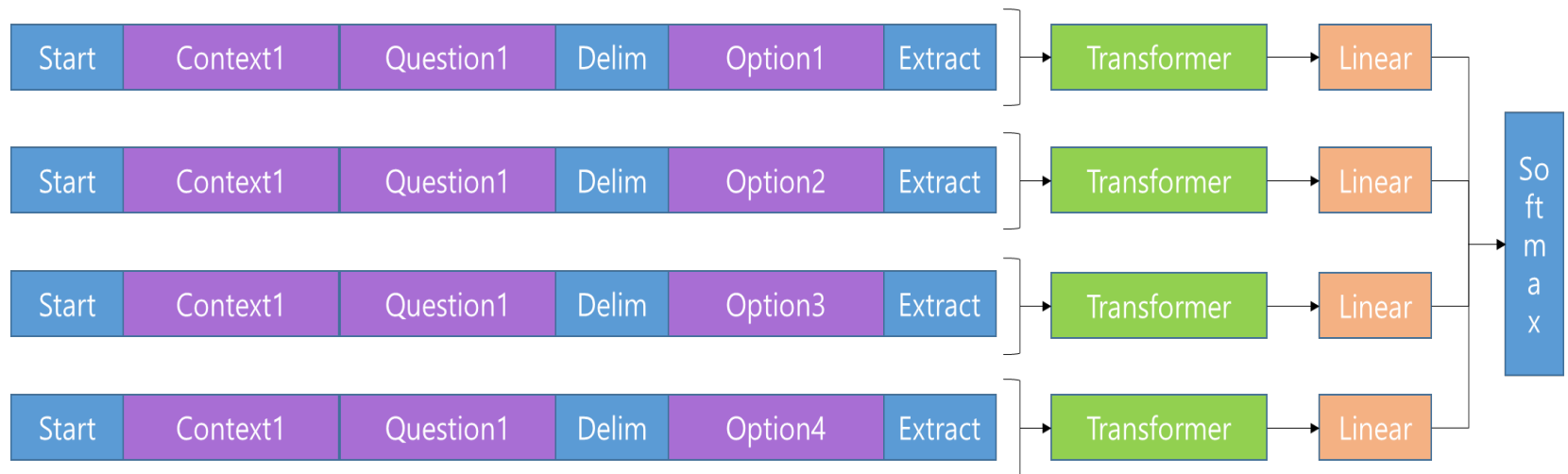
# GPT-1

## ■ 다운스트림 작업 종류에 따른 입력 형태



# GPT-1

## ■ Multiple choice questions



# GPT-1: 모형의 성능

모 형	분류		의미 유사		
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)
Sparse byte mLSTM <sup>a</sup>	-	<b>93.2</b>	-	-	-
TF-KLD <sup>b</sup>	-	-	<b>86.0</b>	-	-
ECNU <sup>c</sup>	-	-	-	<u>81.0</u>	-
Single-task BiLSTM + ELMo + Attn <sup>d</sup>	<u>35.0</u>	90.2	80.2	55.5	<u>66.1</u>
Multi-task BiLSTM + ELMo + Attn <sup>d</sup>	18.9	91.6	83.5	72.8	63.3
Finetuned Transformer LM (ours)	<b>45.4</b>	91.3	82.3	<b>82.0</b>	<b>70.3</b>

<sup>a</sup> Gray, S., Radford, A., & Kingma, D. P. (2017). Gpu kernels for block-sparse weights.

<sup>b</sup> Ji, Y., & Eisenstein, J. (2013, October). Discriminative improvements to distributional sentence similarity.

<sup>c</sup> Tian et al. (2017, August). Ecnv at SemEval-2017 task 1: Leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity.

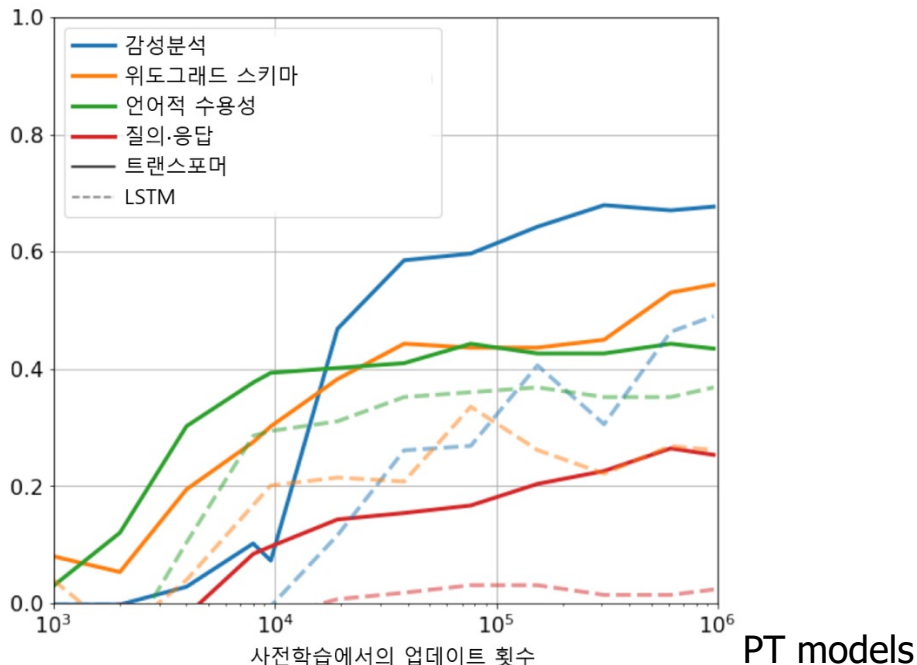
<sup>d</sup> Wang et al. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding.

참고: mc = Matthews correlation, acc=Accuracy, pc=Pearson correlation을, CoLA는 Corpus of Linguistic Acceptability 데이터셋을, MRPC는 Microsoft Paraphrase corpus 데이터셋을, STSB는 Semantic Textual Similarity benchmark 데이터셋을, QQP는 Quora Question Pairs 데이터셋을 의미

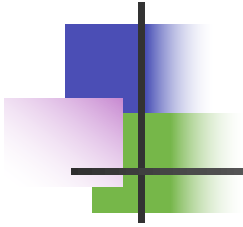
# GPT-1

## ■ 제로샷 (zero-shot) 행동

- 제로샷이라고 하는 것은 추가적인 미세 조정없이 주어진 다운스트림 작업을 사전 학습 모델을 그대로 이용해서 수행하는 것을 의미
  - 특정 다운스트림 작업을 위한 분류기에 존재하는 파라미터의 값은 학습



중요 인사이트:  
추가적인 학습 (즉, 미세조정)을  
하지 않더라도 사전학습을  
충분히 수행하면 다운스트림  
작업에 대한 성능이 좋게  
나온다!!  
⇒ 이를 반영한 것이 GPT-2, 3



# GPT-2



# GPT-2

---

- GPT-1과의 주요한 차이
  - 더 많은 양질의 데이터를 이용해서 학습을 하였고,
  - 더 많은 파라미터를 갖는 모델을 사용했으며,
  - 미세 조정 작업을 수행하지 않았다는 것
- GPT-2 논문의 가장 큰 목적
  - 대용량의 질이 좋은 학습 데이터로 사전 학습된 언어 모델을 추가적인 미세 조정없이 다양한 다운스트림 작업에 적용해 보고, 모델의 성능을 파악해 보는 것
  - 저자들은 다양한 작업에 대한 내용을 담고 있는 학습 데이터를 이용해서 사전 학습된 언어 모델의 경우에는 추가적인 미세 조정 과정이 없이도 특정한 다운스트림 작업에 대해 좋은 성능을 낼 수 있을 것이라는 가설을 세우고, 그러한 가설이 맞는지 해당 논문에서 검증

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.





# GPT-2

---

## ■ Zero-shot

- 의미: 사전 학습 모형에 대한 추가적인 학습, 즉 미세 조정 과정이 없는 것
- 주요 이유: 사전 학습과 지도 학습 기반의 미세 조정을 같이 사용하는 방법이 갖는 한계 때문
  - 이러한 정답 데이터를 생성하는 것이 많은 비용이 들고, 특정한 작업을 위한 정답 데이터를 이용해 미세 조정된 모형의 경우에는 일반화 정도가 낮아 다른 작업에 적용하는 것이 어렵다는 단점 존재
- 주요 아이디어
  - 풀고자 하는 문제에 대한 별도의 정답 데이터를 이용해서 미세 조정 학습을 수행하지 않더라도 그러한 문제와 관련된 내용을 포함하고 있는 학습 데이터를 기반으로 사전 학습된 언어 모형을 이용해서 주어진 문제를 풀 수 있다.
  - 미세조정시, 풀고자 하는 문제와 관련된 예시나 혹은 프롬프트를 사전학습 모형에 입력하는 방식을 사용
  - 프롬프트: 사용자가 GPT 모형을 이용하여 어떠한 종류의 작업을 수행하고자 하는지에 대한 내용을 담고 있는 텍스트



# GPT-2

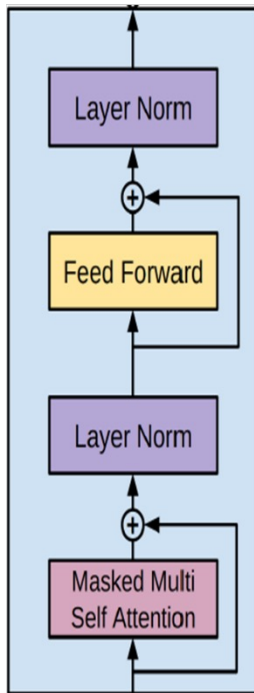
---

## ■ 학습 데이터

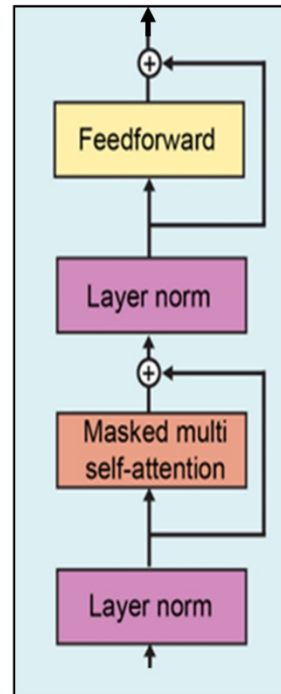
- WebText: 질 높은 학습 데이터를 구축하기 위해서 저자들은 직접 인터넷으로부터 데이터를 수집합니다. 레딧(Reddit)이라고 하는 소셜 미디어 플랫폼에서 세 개 이상의 카르마(karma)를 받은 게시글에 입력된 링크에 대한 텍스트만을 수집
  - 800만개가 조금 넘는 문서
  - 전체 사이즈는 40GB 정도
  - 위키피디아 페이지는 불포함

# GPT-2

## ■ 모형의 구조



GPT-1 디코더 블록



GPT-2 디코더 블록

파라미터수	블록의 수	임베딩 벡터 차원
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600



# GPT-2

## ■ 모형의 성능

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	<b>21.8</b>
117M	<b>35.13</b>	45.99	<b>87.65</b>	<b>83.4</b>	<b>29.41</b>	65.85	75.20
345M	<b>15.60</b>	55.48	<b>92.35</b>	<b>87.1</b>	<b>22.76</b>	47.33	55.72
762M	<b>10.87</b>	<b>60.12</b>	<b>93.45</b>	<b>88.0</b>	<b>19.93</b>	<b>40.31</b>	44.575
1542M	<b>8.63</b>	<b>63.24</b>	<b>93.30</b>	<b>89.05</b>	<b>18.34</b>	<b>35.76</b>	42.16

참고: PPL = Perplexity

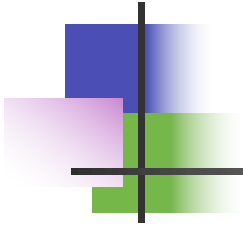


# GPT-2

---

- 추론시 입력되는 데이터의 형태
  - 해당 논문에서는 작업 조건(task conditioning)의 목적으로 추론 시 모형이 적용되는 작업에 따른 특정 문자열 입력 데이터의 일부로 추가
  - 예
    - 문서 요약의 경우, 요약을 의미하는 'TL;DR:'을 입력 데이터의 일부로 추가
    - 번역의 경우에는 번역 작업을 나타내기 위해서 'korean sentence = english sentence' 형태로 예제 데이터를 입력한 후, 번역해야 하는 텍스트를 'korean sentence ='와 같은 형태로 입력
      - 예를 들어, 한글 문장 "오늘은 금요일입니다"를 영어로 번역한 결과를 얻고자 하는 경우, 다음과 같이 입력

'나는 영화를 봅니다 = I watch a movie <delimiter> 오늘은 금요일입니다 ='



# GPT-3



# GPT-3

---

- GPT-3에서의 주요하게 주목한 점
  - 사전 학습이 충분히 된 언어 모델을 사용하는 경우, 미세 조정 과정 없이도 풀고자 하는 문제 관련한 소수의 예를 이용해서도 좋은 성능을 낼 수 있다. ⇒ 이는 GPT-2와 유사
- GPT-2와의 주요 차이
  - 더 다양하고 큰 학습 데이터를 이용해서 언어 모델을 사전 학습
  - GPT-2에 비해서 훨씬 더 거대한 크기의 모델을 사용
    - 가장 큰 모델은 1,750억 개의 파라미터 포함 (해당 논문에서는 이 버전의 모델을 GPT-3라고 함)
  - GPT-2에서는 사전 학습된 언어 모델이 미세 조정 과정없이 여러 종류의 다운스트림 작업에 대해 갖는 모델의 성능을 파악하는 것에 주요한 목적이 있었다면, GPT-3 논문에서는 추론시 입력되는 예시(example)의 수에 따라서 모델의 성능이 어떻게 달라지는지를 파악 ⇒ 제로샷, 원샷, 퓨샷

# 제로샷, 원샷, 퓨샷

- 퓨샷 (few-shot)
  - 입력 데이터에 포함된 작업 관련된 예의 수가 두 개 이상인 경우

1	Translate English to French:	←	작업 설명
2	sea otter => loutre de mer	←	작업 예시
3	peppermint => menthe poivrée	←	
4	plush girafe => girafe peluche	←	
5	cheese => .....	←	프롬프트





# 제로샷, 원샷, 퓨샷

---

- GPT-2에서의 제로샷 의미
  - GPT-2 논문에서의 제로샷의 의미는 몇 개의 예제를 사용했느냐와 관련있는 것이 아니라, 사전 학습 모형에 대한 추가적인 학습 즉, 미세 조정 과정이 있느냐 없느냐와 관련이 있는 것으로 추가적인 파라미터 학습이 없는 경우를 제로샷이라고 표현
  - GPT-2에서도 추론 작업을 할 때 풀고자 하는 문제와 관련된 예제를 입력 데이터의 일부로 입력



# GPT-3

## ■ 학습 데이터

데이터셋	데이터크기 (토큰수)	300B개의 토큰을 학습하는 과정에서 사용된 정도
Common Crawl (filtered)	410 billion	0.44
WebText2	19 billion	2.9
Books1	12 billion	1.9
Books2	55 billion	0.43
Wikipedia	3 billion	3.4



# GPT-3

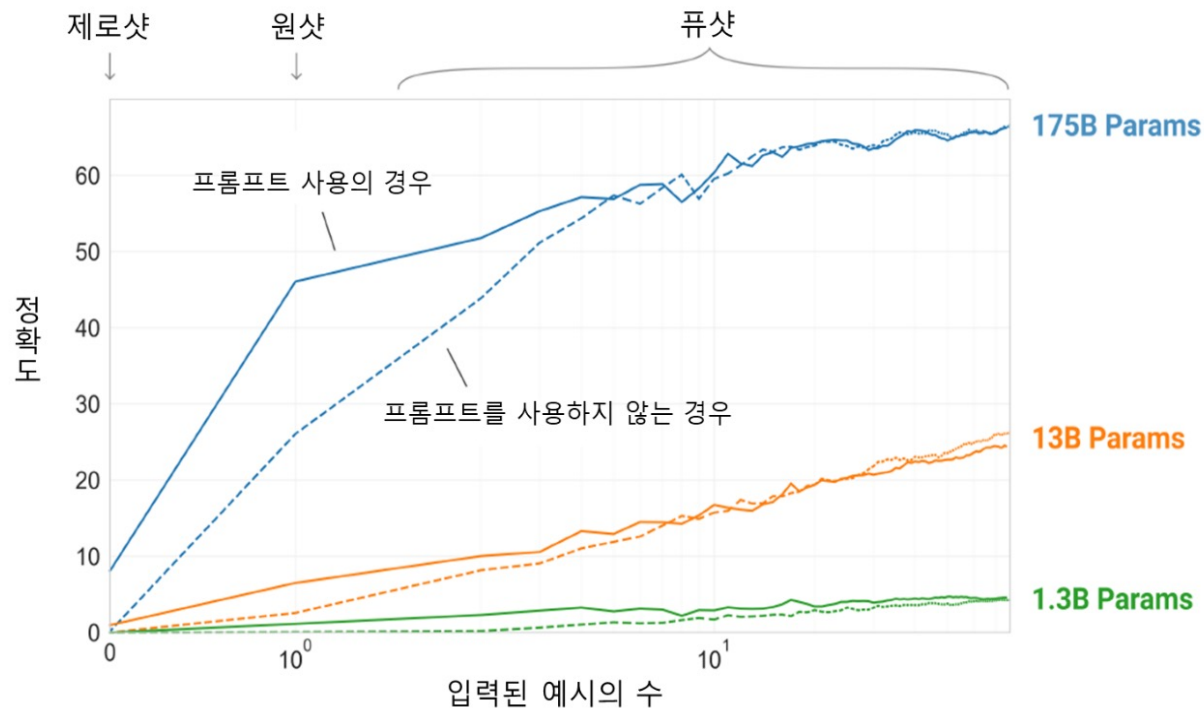
## ■ 모형의 구조

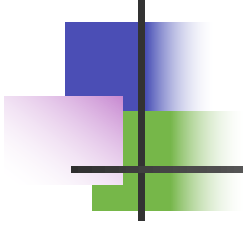
모형 이름	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	배치 크기	학습률
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

## ■ 학습시 Sparse Transformer 사용 (남선 자료 참고)

# GPT-3

## ■ 모형의 성능





# InstructGPT



# InstructGPT

---

- GPT-3와의 주요 차이
  - 미세 조정 방법 사용
- GPT-3와 그 이전 모형의 주요 문제
  - Hallucination
    - often defined as "generated content that is nonsensical or unfaithful to the provided source content"
  - 사실을 조작한다든지, 편향된 혹은 악의적인 텍스트를 생성한다든지, 또는 사용자의 지시를 따르지 않는 것 등의 문제
  - 주요 이유
    - 이러한 이유 중 하나는 주어진 단어들을 이용해서 다음 단어를 예측하는 언어 모형의 목적이 "사용자의 지시를 안전하고 도움이 될 수 있게 수행하라"는 것과 차이가 있기 때문



# InstructGPT

---

- InstructGPT 논문의 주요 목적
  - 사용자의 의도와 일치하는 방향으로 행동하는 언어 모형 제안
    - 명시적 의도 뿐 아니라 암묵적 의도도 포함
  - How?
    - 미세 조정 방법 사용, 특히 사람들의 피드백을 이용한 강화 학습 방법을 사용하여 미세 조정을 수행
  - 즉, InstructGPT 논문에서는 사람의 피드백 기반 미세 조정 방법을 사용해서 사용자의 의도에 맞는 결과를 반환하는 언어 모형을 제안



# InstructGPT

- InstructGPT에서의 미세 조정
  - InstructGPT는 사전학습된 GPT-3 모델을 여러 가지 방법으로 미세 조정하여 도출된 결과물로 간주 가능
  - 미세 조정 단계
    - 단계 1: 지도학습 기반의 미세 조정 (Supervised fine-tuning, SFT)
      - 준비된 프롬프트에 대해 사람이 직접 응답 데이터를 생성하고, 이러한 응답 데이터를 정답으로 사용해서 사전학습된 GPT-3 모델을 미세 조정하는 단계
    - 단계 2: 보상 모델 (Reward model)
      - 동일한 프롬프트에 대해 단계 1에서 미세 조정된 모델 (즉, SFT 모델)이 출력하는 여러 개의 서로 다른 응답들에 대해 사람이 직접 평가하여 순위를 부여하고, 보상 모델을 이용해서 점수를 계산한 후, 그러한 순위와 점수 정보를 이용해서 모델을 학습하는 단계
    - 단계 3: 보상 점수를 이용해서 강화학습 수행
      - 하나의 프롬프트에 대해 SFT 모델을 이용해서 응답을 생성하고, 생성된 응답에 대해서 단계 2에서의 보상 모델을 이용해 보상 점수를 계산 ⇒ 강화학습 방법을 사용해서 이러한 보상 점수가 높은 응답이 출력될 수 있도록 학습
  - \* 단계 2와 3은 여러 번 반복적으로 수행





# InstructGPT

---

- 미세 조정 데이터
  - '프롬프트(prompt) – 응답(response)' 으로 구성
  - OpenAI는 이러한 데이터를 구축하기 위해서 40명의 휴먼 코더 채용
- 프롬프트 데이터 생성: 두 가지로 구성
  - 1) OpenAI 에서 제공하는 GPT-3 기반의 API에 입력된 프롬프트를 사용
    - API에 입력된 프롬프트를 사용하는 경우, 다양성을 높이기 위해서 사용자 당 최대 200개의 프롬프트만을 사용
  - 2) 일부는 고용된 코더들이 직접 생성



# InstructGPT

---

- 미세 조정 데이터 (cont'd)
  - 앞 과정을 통해 생성된 프롬프트를 이용해서 미세 조정의 각 단계에서 필요한 데이터셋을 구축
  - SFT 단계에서 사용된 학습 데이터셋
    - 13,000개 정도의 프롬프트로 구성
    - 고용된 코더들이 각 프롬프트에 대해 직접적으로 응답을 생성하고 이렇게 생성된 응답을 정답 정보로 사용
  - 단계 2에서의 보상 모형을 위한 데이터셋
    - 33,000개 정도의 프롬프트로 구성
    - 정답 정보로는 코더들이 하나의 프롬프트에 대해 SFT 모형이 생성하는 여러 응답 (4 – 9개)에 순위를 매긴 정보를 사용
  - 단계 3에서의 강화학습을 위해 사용된 데이터셋
    - API에서 추출된 31,000개의 프롬프트로만 구성
    - 각 프롬프트에 대해 보상 모형이 출력하는 점수 정보를 사용



# InstructGPT

## ■ 미세 조정 단계별 데이터의 구성

단계 1 데이터			단계 2 데이터			단계 3 데이터		
split	source	size	split	source	size	split	source	size
train	labeler	11,295	train	labeler	6,623	train	customer	31,144
train	customer	1,430	train	customer	26,584	valid	customer	16,185
valid	labeler	1,550	valid	labeler	3,488			
valid	customer	103	valid	customer	14,399			

참고: labeler는 휴먼 코더가 생성한 프롬프트를 의미하며, customer는 API 사용자가 입력한 프롬프트를 의미



# InstructGPT

---

## ■ 카테고리별 프롬프트의 비중

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%



# InstructGPT

---

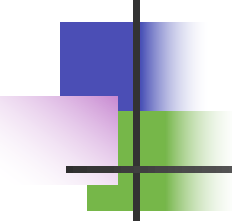
- 휴먼 코더

- 휴먼 코더의 구성

- 결과의 질을 높이기 위해 다양한 인구통계학적 그룹을 대표할 수 있는 사람들로, 그리고 InstructGPT가 출력하는 결과물 중에서 해로운 결과물을 잘 구분할 수 있는 사람들로 코더를 구성

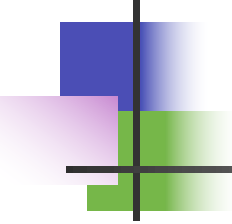
- 휴먼 코더 훈련

- 응답 데이터를 생성하는데 있어서 프롬프트를 작성한 사용자의 의도를 추론하는데 최대한의 노력을 하게끔 하고,
    - 응답 데이터를 생성하거나 모형이 출력하는 응답의 순위를 부여하는 경우, 응답의 진실성을 고려하게 하였으며,
    - 해로운 응답이나 혹은 편향되거나 악의적인 표현의 답변들을 구분하도록 훈련



# InstructGPT: 미세조정 단계

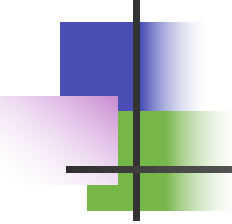
- 단계 1: Supervised Fine-Tuning, SFT
  - 데이터셋은 13,000개 정도의 프롬프트로 구성
  - 각 프롬프트에 대해 휴먼 코더가 직접적으로 생성한 응답을 정답 정보로 사용
  - 이러한 데이터를 이용해서 사전학습된 GPT-3 모델을 미세 조정
- 단계 1의 한계
  - 데이터 양이 많지 않음 ⇒ 결과가 안 좋을 것으로 예상
  - 하지만, 정답 데이터를 생성하는 것이 어려움 ⇒ 모형 개선을 위해 다른 방법 사용
  - 즉, 휴먼 코더가 프롬프트에 대한 응답을 생성하는 것이 아니라, 하나의 프롬프트에 대해서 SFT 모형이 출력하는 서로 다른 응답들에 대해서 휴먼 코더가 응답의 질에 따라 순위를 부여하고 그러한 순위 정보를 이용해서 보상 모형을 학습하는 방법을 사용



# InstructGPT: 미세조정 단계

---

- 단계 2: 보상 모형 (Reward model)
  - 구조
    - 단계 1에서 미세 조정 학습된 모형 (즉, SFT 모형)을 보상 모형의 초기 모형으로 사용
    - 보상 모형의 경우는 최종적으로 입력된 프롬프트-응답 시퀀스에 대한 숫자 형태의 보상 점수를 출력하고, 이를 위해서 보상 점수 출력을 위한 추가적인 완전연결층을 SFT 모형에 추가
    - 보상 모형으로는 175B의 파라미터를 갖는 GPT-3 모형이 아니라 6B개의 파라미터를 갖는 모형 ⇒ 학습을 보다 효율적이고 안정적으로 수행하기 위해서
  - 데이터셋
    - 33,000개 정도의 프롬프트 각각에 대해서 SFT 모형이 4-9개의 응답 출력 ⇒ 휴먼 코더가 정해진 기준에 따라 응답의 순위를 부여
    - 이러한 순위 정보와 보상 점수를 이용해서 비용함수 계산

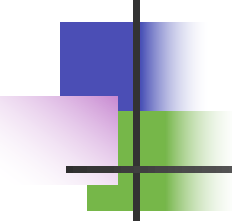


# InstructGPT: 미세조정 단계

---

- 단계 2: 보상 모형 (Reward model) (cont'd)
  - 데이터셋의 구성
    - 하나의 샘플은 하나의 프롬프트와 두 개의 응답으로 구성
    - 이러한 샘플을 구성하기 위해서, 하나의 프롬프트에 대해 SFT 모형이 출력하는 여러 개의 응답들 중에서 임의로 두 개를 선택
    - 만약, 하나의 프롬프트에 대해서 SFT 모형이 생성하는 응답이 K개라고 하는 경우, 두 개의 응답을 구성할 수 있는 경우의 수 =  $\binom{K}{2}$
  - 해당 프롬프트와 이렇게 선택된 두 개의 응답을 하나의 샘플로 구성하여 보상 모형에 입력하고 각 응답에 대한 보상 점수를 출력해서 비용함수를 계산





# InstructGPT: 미세조정 단계

---

- 단계 2: 보상 모형 (Reward model) (cont'd)

- 비용함수 계산의 예)

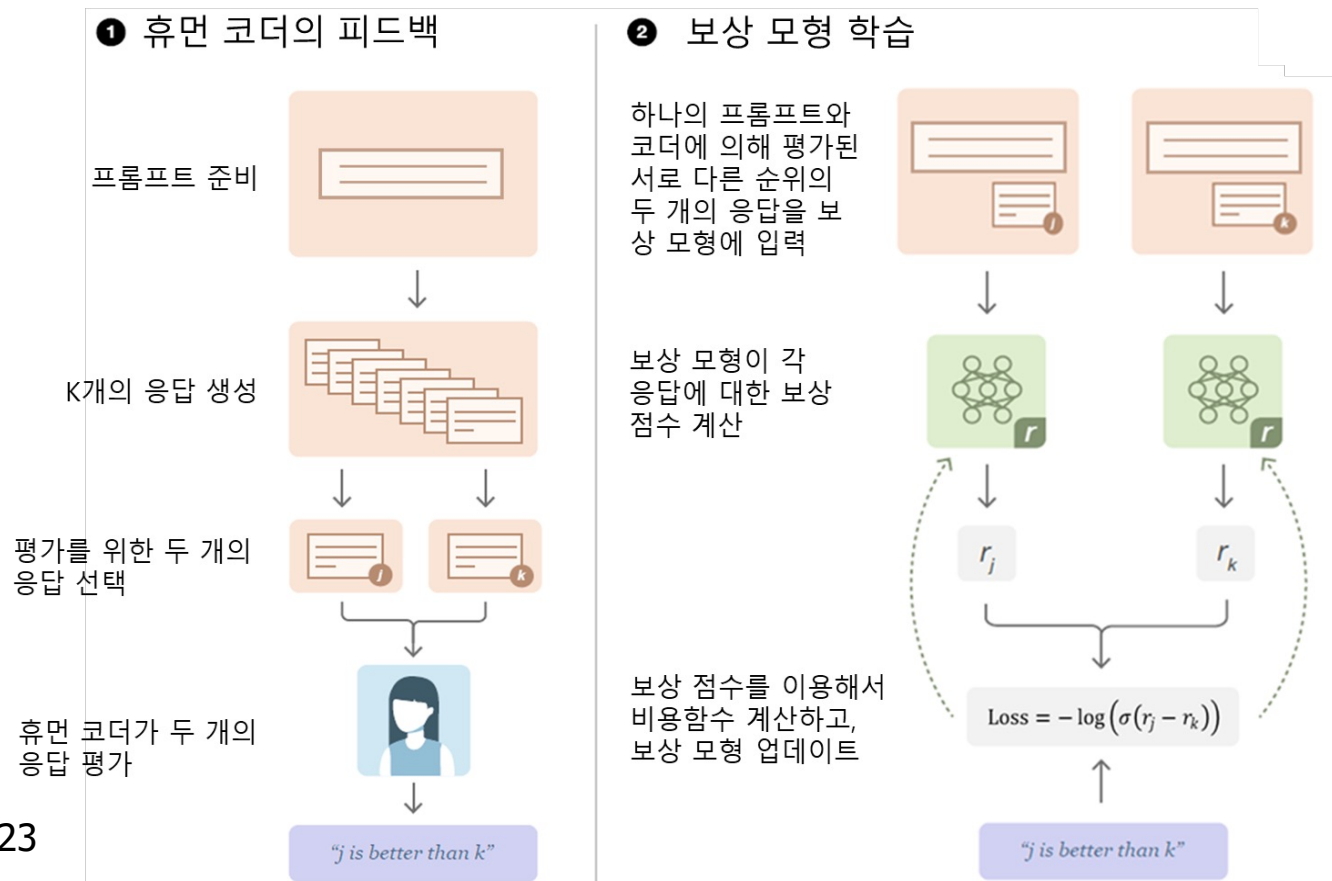
- 하나의 프롬프트 (프롬프트1)에 대해서 SFT 모형이 출력한 서로 다른 두 개의 응답 (응답1과 응답2)이 있다고 가정
    - 휴먼 코더에 따르면 응답1이 응답2 보다 더 우수한 응답으로 구분되었다고 가정
    - 샘플: (프롬프트1, 응답1), (프롬프트1, 응답2)
    - 각 '프롬프트-응답'에 대해서 보상 모형을 이용해 각 응답에 대한 보상 점수 계산
    - 응답1에 대한 보상 점수를  $r_1$ 이라고 하고, 응답2에 대한 보상 점수를  $r_2$ 라고 표현

$$\text{Loss} = -\log(\sigma(r_1 - r_2))$$

- 만약 모형이 예측을 잘못해서  $r_2 > r_1$ 가 된다면, 비용 증가

# InstructGPT: 미세조정 단계

## ■ 단계 2: 보상 모형 (Reward model) (cont'd)





# InstructGPT

- 단계 3: 강화학습 모형 학습
  - InstructGPT 논문에서는 단계 2에서 학습된 보상 모형을 이용해 사람의 기준에서 질이 좋은 답변을 출력하는 모형(즉, InstructGPT)을 추가적으로 학습
  - 이를 위해서 강화학습 방법 사용
    - 강화학습 알고리즘 중에서도 PPO (Proximal Policy Optimization) 알고리즘을 사용
- \* 강화학습과 PPO 관련해서는 별도 파일 참고
- 작동 방식
  - SFT 모형을 초기 정책으로 사용
  - 이러한 정책을 이용해서 입력된 프롬프트에 대해 응답을 출력
  - 단계 2에서 학습된 보상 모형을 사용해 해당 응답에 대한 보상 점수를 계산
  - 강화학습 방법을 이용해서 이러한 보상 점수를 최대화하는 방향으로 정책을 업데이트
- 목적함수
  - $$R(x, y) = r_{\theta}(x, y) - \beta \log[\pi_{\phi}^{RL}(y|x)/\pi^{SFT}(y|x)]$$

# InstructGPT

- 단계 3: 강화학습 모형 학습 (cont'd)

- 목적함수

- $R(x, y) = r_{\theta}(x, y) - \beta \log[\pi_{\phi}^{RL}(y|x)/\pi^{SFT}(y|x)]$

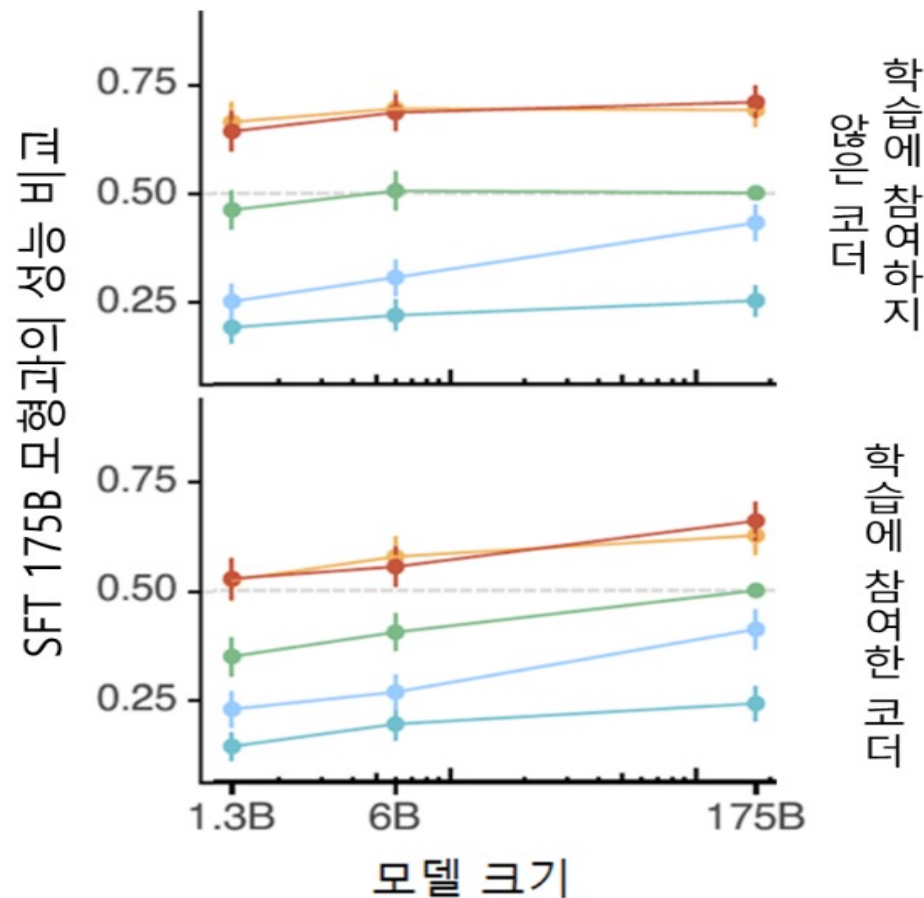
- 이때 사용되는 쿨백-라이블러 발산(즉,  $\log[\pi_{\phi}^{RL}(y|x)/\pi^{SFT}(y|x)]$ )은 단계 1에서의 SFT 모형에 대한 정책 (즉, SFT 모형이 출력하는 확률 분포)와 학습된 강화학습 모형에 대한 정책 (즉, 강화학습 모형이 출력하는 확률 분포) 간의 쿨백-라이블러 발산

- $x$ 는 입력 프롬프트를,  $y$ 는 모형이 출력하는 응답을 나타내고,  $r_{\theta}(x, y)$ 는 보상 모형이 반환하는 보상 점수를 의미.  $\beta$ 는 쿨백-라이블러 페널티항에 대한 적응 계수

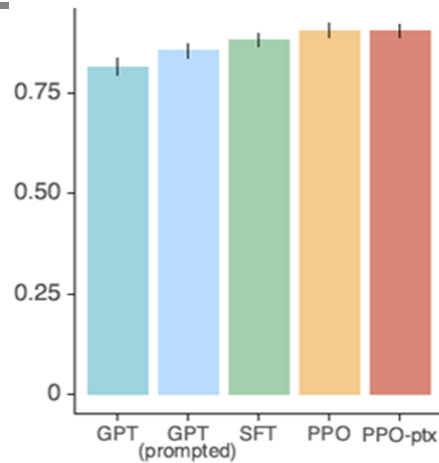
- 업데이트는 각 토큰 단위로 진행되는 것이 아니라, 전체 정답에 대해서 수행 (?)

- InstructGPT 논문에서는 위와 같은 기본적인 PPO 방법 뿐만 아니라, 위의 목적함수에 사전 학습 모형에서 사용된 경사값(**gradient**)을 추가한 형태의 목적함수를 이용한 방법도 사용. 해당 논문에서는 이러한 방법을 PPO-ptx (ptx는 **pre-training mix**)라고 표현

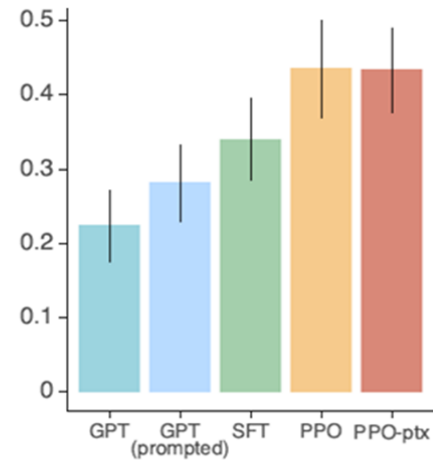
# InstructGPT: 모형의 성능



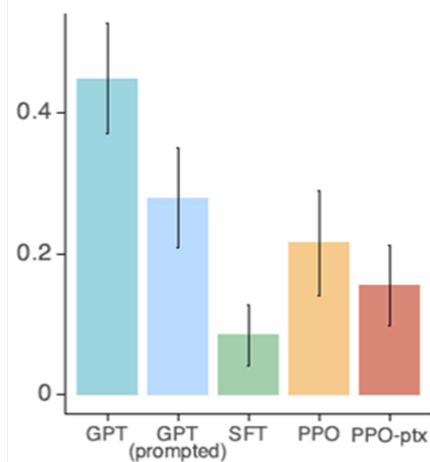
# InstructGPT: 모형의 성능



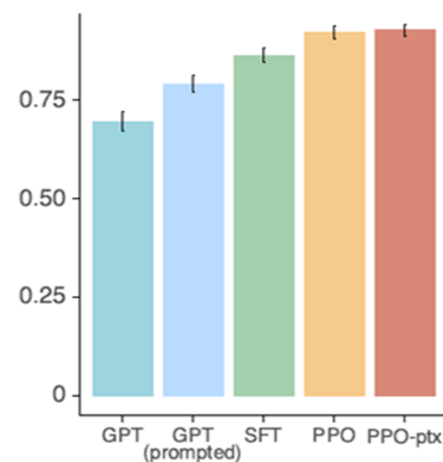
보조자 역할 표현



명시적 제한 사항



거짓 정보



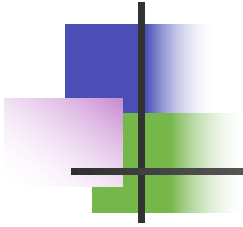
지시 사항 파악



# ChatGPT

---

- 주요 내용
  - <https://openai.com/blog/chatgpt>
  - InstructGPT와 거의 동일한 방법으로 학습
    - 다만, 사람과의 대화를 주된 목적으로 함
    - 대화 데이터를 학습 데이터로 추가
      - 휴먼 코더들이 사용자와 AI 어시스턴트의 역할을 모두 수행하며 정답에 해당하는 대화 데이터를 생성
  - InstructGPT가 GPT-3를 사전학습 모형으로 사용한 반면, ChatGPT는 GPT-3.5 또는 4를 사전학습 모형으로 사용
    - <https://platform.openai.com/docs/models/gpt-3-5>



# GPT 실습

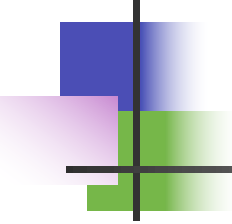




# GPT2

---

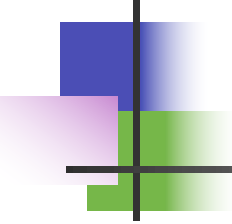
- 사전학습 GPT2를 이용한 텍스트 생성
  - GPT2\_example.ipynb 참고
  - <https://huggingface.co/> 에서 제공하는 사전학습 모형 사용



# GPT-3 미세조정

---

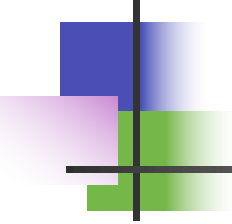
- 텍스트 생성의 예
  - 자체적인 정답 데이터 사용
  - 기업의 경우는 해당 기업이 보유한 데이터를 사용하여 자체적인 질의응답 모형이나 고객에게 보내는 메시지를 생성하는 모형 개발 가능
  - GPT-1, 2는 허깅 페이스에 등록되어 있으나, GPT-3는 OpenAI에서 제공하는 API 사용



# GPT-3 미세조정

---

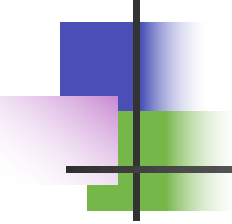
- OpenAI API를 이용한 미세조정 사전 준비
  - OpenAI 계정 생성
    - <https://platform.openai.com/account/>
  - API 키 정보 생성
    - <https://platform.openai.com/account/api-keys>
  - 키 정보를 이용해 OPENAI\_API\_KEY 시스템 환경 변수 생성
  - openai 설치
    - `pip install --upgrade openai`



# GPT-3 미세조정

---

- OpenAI API를 이용한 미세조정 단계
  - 정답 데이터 준비하기
  - 준비된 데이터를 이용해서 미세조정 수행하기
  - 미세조정 모형 사용하기
- 예
  - 하고자 하는 작업: 가상의 초능력자의 나이, 성별, 초능력 정보를 이용해서 해당 초능력자에 대한 짧은 글 생성
  - 프롬프트의 예: '20, man, teleport ->'



# GPT-3 미세조정

---

## ■ 생성된 텍스트의 예

He is a 20-year-old man with short black hair and an athletic build. His eyes are a deep grey, like a stormy sky. He stands 6'2" and wears a black leather jacket and dark jeans to accent his toned physique. He has the superpower of teleportation, which he has been able to use since a young age. With a thought, he is able to transport himself from one place to another, instantly covering great distances. He is able to carry objects or people who are in contact with him, which opens up an infinite range of possibilities for his adventures.

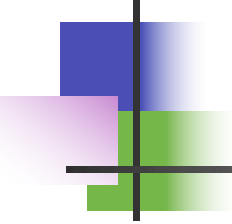
# GPT-3 미세조정

- 단계1: 정답 데이터 준비
  - 프롬프트와 정답으로 구성
    - OpenAI에서는 정답을 completion 라고 함

	A	B
1	prompt	completion
	18, man, invisibility	He is a slim 18-year-old with short black hair and soft brown eyes. He stands at an average height of 5'9" and has a light complexion. His left ear has a few small piercings and a tiny butterfly tattoo just above it. He usually wears faded black jeans with a dark blue t-shirt or a plain white hoodie. He has a brooding attitude, but beneath his serious exterior lies a gentle soul with a sharp tongue. He has the power of invisibility, allowing him to blend in with the shadows and travel undetected. He tends to shy away from crowds, preferring to explore the world on his own and ponder the greater mysteries that life has to offer.
2	18, man, invisibility	He had curly sandy brown hair which always seemed like it was defying gravity. His piercing green eyes could pierce through anyone's soul. A thin body, with a mysterious aura that only grew when he used his superpower. His confidence radiated when invisible, always tapping into deeper aspects of himself. His superpower was an invisible shield, being able to turn invisible anytime he wanted and walk around freely with no one being the wiser. His superpower was a gift, but he found himself wishing that he could

prepared\_data

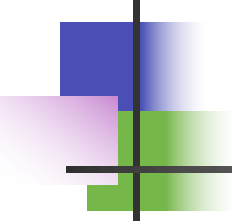
Ready



# GPT-3 미세조정

---

- 단계1: 정답 데이터 준비
    - openai를 사용하기 위해서는 데이터를 jsonl 형태로 변환 필요
    - 명령프롬프트에서 아래 명령문 실행
- ```
openai tools fine_tunes.prepare_data -f prepared_data.csv
```
- 이후 나오는 질문에 대해서 Y 입력
  - prepared\_data\_prepared.jsonl 파일 생성됨



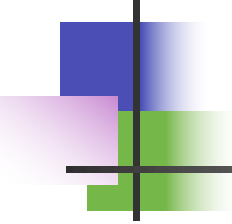
# GPT-3 미세조정

---

- prepared\_data\_prepared.jsonl 데이터의 예

```
{"prompt":"18, man, invisibility ->","completion":" \n\nHe is a slim 18-year-old with short black hair and soft brown eyes. He stands at an average height of 5'9W" and has a light complexion. His left ear has a few small piercings and a tiny butterfly tattoo just above it. He usually wears faded black jeans with a dark blue t-shirt or a plain white hoodie. He has a brooding attitude, but beneath his serious exterior lies a gentle soul with a sharp tongue. He has the power of invisibility, allowing him to blend in with the shadows and travel undetected. He tends to shy away from crowds, preferring to explore the world on his own and ponder the greater mysteries that life has to offer. END"}
```





# GPT-3 미세조정

- 단계2: 준비된 데이터를 이용해 미세조정 하기
  - 명령프롬프트에서 다음 형식의 명령문 실행

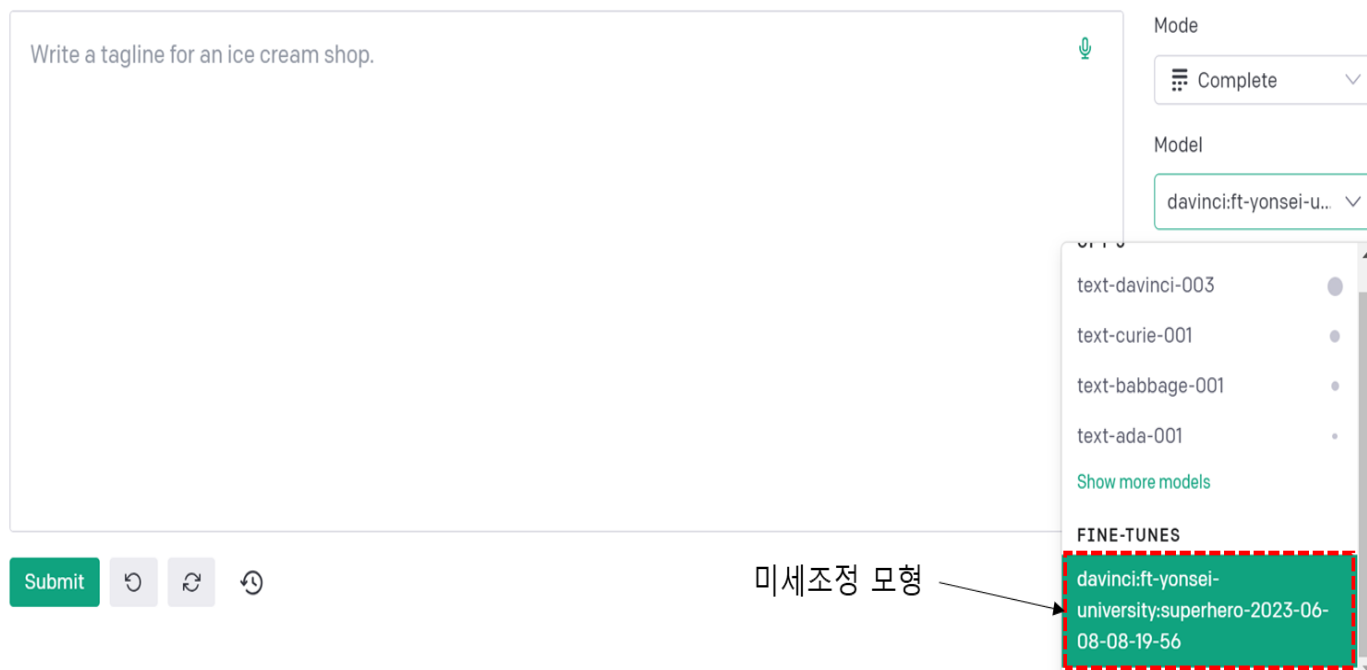
```
openai api fine_tunes.create -t 정답데이터 -m 사용모형 --suffix "이름의추가부분"
```

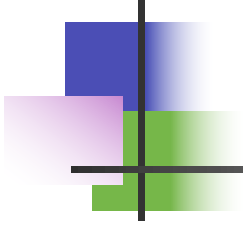
- 사용 가능 모형
  - GPT-3 사전학습 모형: Ada, Babbage, Curie, Davinci
- 입력 명령문의 구체적 예

```
openai api fine_tunes.create -t prepared_data_prepared.jsonl -m davinci --suffix "SuperHero"
```

# GPT-3 미세조정

- 단계3: 미세조정 모형 사용하기
  - <https://platform.openai.com/playground> 에서 확인 가능





# Q & A