

# **ChatGPT: Optimizing Language Models for Dialogue**

Date

**2023. 01. 24**

Presenter

**이지현**

# Contents

1. ChatGPT introduction
2. GPT
  - a. GPT-1 : Improving Language Understanding by Generative Pre-Training
  - b. GPT-2 : Language Models are Unsupervised Multitask Learners
  - c. GPT-3 : Language Models are Few-Shot Learners
  - d. InstructGPT : Training language models to follows instructions with human feedback
3. ChatGPT
  - a. Methods
  - b. Limitations
  - c. Iterative deployment
4. Conclusion

# ChatGPT introduction

---

# ChatGPT

- OpenAI에서 학습한 대규모 언어 모델
  - 언어 모델 : 인간과 유사한 텍스트를 생성
  - 채팅으로 물어본 질문을 통해 텍스트 답변 생성
  - <https://openai.com/blog/chatgpt/>
- 2022년 11월 30일 prototype launching
- 공개된지 5일 만에 백만 유저 가입



OpenAI CEO Sam Altman의 트윗

## 1. ChatGPT introduction

# ChatGPT

### ● 활용 예시

 ChatGPT  
@ChatGPTGoneWild · Follow

ChatGPT explains HTML like we're@ a dog:

**Human:**  
Explain HTML to me like I'm a dog.

**ChatGPT:**  
As a dog, it's likely that you don't have the ability to understand or use the internet in the way that humans do. However, I can try to explain the concept of the 'nav' element in a way that might make sense to you.  
  
Imagine that you are at a park with your human. There are lots of different paths and trails that you can explore, and your human wants to help you find your way around. They might give you a map of the park, or they might tell you which way to go to get to different areas of the park.  
  
In a similar way, the 'nav' element on a website is like a map or a set of directions that helps people find their way around the site. It usually has links to different pages or sections of the site, and people can click on the links to go to the different pages.  
  
I hope that helps! As a dog, it's important to remember that the internet is a tool for humans, and it's not something that you need to worry about. Just focus on having fun and being a good companion to your human.

9:18 AM · Jan 8, 2023

383 Reply Copy link

Read 9 replies

"HTML을 강아지에게 설명해줘"

Can you rewrite the scenario of the movie Parasite so that the Kim family fails?

Certainly! Here is a scenario of the movie "Parasite" where the Kim family fails:  
  
In this version of the story, the Kim family is still struggling to make ends meet and living in a cramped, semi-basement apartment. They hear about the wealthy Park family's need for a tutor for their son, and Mr. Kim manages to secure the job for his daughter, Ki-jung. However, things do not go as planned. Ki-jung is unable to connect with the Park's son and is fired from the job.  
  
The Kim family's plan to infiltrate the Park household falls apart, and they are unable to find any other jobs within the household. Desperate for money, they decide to turn to illegal means to make ends meet. However, their illegal activities are eventually discovered by the authorities, and the family is arrested and sent to prison.  
  
Without the help of the Park family and their financial support, the Kim family is unable to escape their poverty and live a better life. They remain trapped in their difficult circumstances, unable to break free from their cycle of poverty.

"영화 기생충 시나리오를 김씨 가족이 실패하는 경우로 다시 써줘"

지현 이미지 분류 중 연구할 만한 주제가 있을까?

이미지 분류는 딥러닝의 일반적인 응용 중 하나입니다. 여러분이 연구하려는 분야에 따라 다른 주제가 적절할 수 있지만, 일반적으로 이미지 분류에 관심 있는 사람들이 연구하는 주제로는 다음과 같은 것들이 있습니다:

- 컴퓨터 비전 분야에서의 이미지 분류
- 이미지 인식에서의 딥러닝
- 이미지 분류에서의 강력한 컨볼루션 모델
- 고려상도 이미지 분류
- 작은 데이터셋에서의 이미지 분류
- 이미지 분류를 위한 가장 최신의 딥러닝 기술
- 이미지 분류 알고리즘의 비교 분석

이러한 주제 중 관심 있는 것을 선택하여 자세히 연구해보세요.

지현 Medical 딥러닝의 전망이 어떤가

Medical 딥러닝은 의료 분야에서 점점 더 중요해지고 있는 주제 중 하나입니다. 이미지 인식, 자연어 처리, 그리고 기계 학습 기술을 사용하여 의료 데이터를 분석하면서 진단, 예측, 치료 방법을 개선하는 등의 분야에서 새로운 기회를 제공합니다.

Medical 딥러닝은 이미지 인식에서부터 자연어 처리, 예측 분석, 의료 데이터 분석, 딥러닝 기반 의료 응용 기술 개발 등 다양한 의료 응용에 사용되고 있습니다. 이미지 인식을 통한 진단, 예측 분석을 통한 의료 임상 응용, 자연어 처리를 통한 의료 레코드 자동화, 의료 데이터 분석을 통한 의료 연구 등 의료 분야에서 딥러닝 기술이 적용되고 있으며, 앞으로도 이러한 의료 분야에서 딥러닝 기술의 적용이 계속 증가할 것으로 예

## 1. ChatGPT introduction

# ChatGPT

- ### ● 활용 예시

The screenshot shows a ChatGPT conversation. The user asks for a CDK code snippet to create an S3 bucket. ChatGPT provides the code and explains it. Then, the user asks for an example IAM policy. ChatGPT generates a policy allowing access to the S3 bucket. Finally, the user asks if the policy allows creating objects. ChatGPT clarifies that the policy only allows reading objects.

Shreyas Panhalkar  
@achillesHeelV2

## ChatGPT can generate CDK code snippets and can understand AWS IAM policies.

Generate a CDK code snippet (`aws s3`) to create an S3 bucket. Furthermore, the AWS IAM policy used to access should be restricted to anyone except for Lambda execution role.

```
import * as cdk from 'aws-cdk-lib';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { BucketEncryption } from 'aws-cdk-lib/aws-s3';
import { Stack, StackProps } from 'aws-cdk-lib';

const myBucket = new cdk.Stack(this, 'MyBucket', {
    // Set the bucket to be private so that it is not readable by anyone by default
    stackName: 'MyBucket',
    });

// Grant full access to this bucket for the Lambda execution role
myBucket.addPermission('lambdaExecution', {
    actions: ['s3:PutObject'],
    resources: ['*'],
});
```

Here is an example CDK code snippet in `index.ts` to create an S3 bucket that is only readable by the Lambda execution role.

```
import * as cdk from 'aws-cdk-lib';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { BucketEncryption } from 'aws-cdk-lib/aws-s3';
import { Stack, StackProps } from 'aws-cdk-lib';

const myBucket = new cdk.Stack(this, 'MyBucket', {
    // Set the bucket to be private so that it is not readable by anyone by default
    stackName: 'MyBucket',
    });

// Grant full access to this bucket for the Lambda execution role
myBucket.addPermission('lambdaExecution', {
    actions: ['s3:PutObject'],
    resources: ['*'],
});
```

Provide the following IAM policy

```
Version: '2012-10-17'
Statement: [
    {
        "Effect": "Allow",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::mybucket/*"
    },
    {
        "Effect": "Deny",
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::mybucket/*"
    }
]
```

The IAM policy denies the `s3:PutObject` action on all resources managed by `"arn:aws:s3:::mybucket/*"`. But it also grants the `s3:GetObject` action on all resources managed by `"arn:aws:s3:::mybucket/*"`. Hence, the `s3:PutObject` action is denied, but the `s3:GetObject` action is allowed.

In other words, this IAM policy does not allow the `s3:PutObject` action to be performed. IAM policies are evaluated in order, and the first matching statement is applied. In this case, the second statement matches the action, but the second statement denies it, so the action is ultimately denied.

 **Dror Hilman**  
@drorhilman

## ChatGPT is my new code reviewer...

 can you refactor this code to be more readable?

```
def some_func(list_a, list_b):
    for i in range(len(list_a)):
        if list_a[i] != list_b[i]:
            return False
    return True
```

 Here is one possible refactoring of the code that makes it more readable:

```
def are_lists_equal(list_a, list_b):
    for i, (a, b) in enumerate(zip(list_a, list_b)):
        if a != b:
            return False
    return True
```

This refactored version uses more descriptive variable names and function names, and also uses the `'enumerate'` and `'zip'` functions to make the code more concise and easier to read. The `'enumerate'` function allows us to loop over the elements of `'list_a'` and `'list_b'` simultaneously and access their indexes, while the `'zip'` function allows us to pair the elements of the two lists together so that we can compare them directly. This eliminates the need to index into the lists manually. ↗ Again → the code more readable.

Ananay @ananayarora · Dec 1

ChatGPT teaches you how to use my Sign in with Apple library for Node.js :)(except the code example is incorrect)

Sign in with Apple in Node.js

To enable Sign in with Apple in a Node.js application, you will need to use the "passport-apple" library, which is a popular, open-source library for implementing Sign in with Apple authentication in Node.js applications. This library provides an easy-to-use interface for integrating Sign in with Apple into your application, and it supports the OAuth 2.0 authentication protocol used by Sign in with Apple.

Here is an example of how to use the "passport-apple" library to enable Sign in with Apple in a Node.js application:

```
const passport = require("passport");
const AppleStrategy = require("passport-apple").Strategy;

// Set up the Sign in with Apple strategy
passport.use(new AppleStrategy({
  clientID: "<your-client-id>",
  clientSecret: "<your-client-secret>",
  callbackURL: "<your-callback-url>"
}, function(accessToken, refreshToken, profile, done) {
  // Save the user's profile and access token
  return done(null, {
    profile: profile,
    accessToken: accessToken
  });
}));
```

# ChatGPT

- 다크웹 내 포럼에서 활용하는 사이버 범죄자들이 **ChatGPT**를 악성 공격에 활용한 사례 발생 (이미 최소 3건)
- **Malware** 작성부터 다크웹 시작 구축까지
  - **Malware** 제작의 난이도를 **Chatbot** 들이 크게 낮춰줄 수 있음
- 인공지능 개발자들이 악성 명령을 차단하도록 인공지능을 훈련시켜야 함

멀웨어를 만드는 건 상당히 높은 수준의 기술이다. 하지만 더 이상 그 기술력을 열망하지 않아도 될 것이다. 챗GPT가 다 해결해줄 것이기 때문이다.

[보안뉴스 문가용 기자] 지난 11월 오픈AI(OpenAI)에서 챗GPT(ChatGPT)라는 인공지능 챗봇을 공개한 이후 수많은 보안 전문가들이 목소리를 모아 예언을 하나 했다. 사이버 범죄자들이 챗봇을 이용해 멀웨어를 작성하는 등 각종 악성 행위를 편리하게 실시할 수 있을 거라는 것이었다. 보안 업계의 불길한 예언은 왜 그리 잘 맞아 떨어지는 것일까. 그 말은 이미 현실로 이뤄지고 있는 것으로 보인다.

## 멀웨어 작성에서부터 다크웹 시장 구축까지

체크포인트가 발견한 세 건의 사건 중 한 건은 한 멀웨어 개발자와 관련된 것이다. 이 인물은 다크웹 해킹 포럼에서 자신이 챗GPT를 활용해서 예전부터 잘 알려진 멀웨어를 개발하는 실험을 진행했다고 공개했다. 뿐만 아니라 챗GPT가 자신의 명령에 따라 개발한 파이선 기반 정보 탈취 멀웨어의 코드 일부를 공유하기도 했다. 12가지 종류의 파일을 피해자의 시스템에서 훔쳐내는 멀웨어였다. 그는 챗GPT에게 명령을 내려 PuTTY SSH와 텔넷 클라이언트를 다운로드 받는 자바 코드를 만들었고, 파워셸을 통해 피해자 시스템에서 몰래 실행시키는 데에 성공했다고까지 밝혔다

그러더니 12월 1일 한 USDoD라는 이름을 사용하는 한 공격자가 챗GPT를 사용해 생성한 파이선 스크립트를 다크웹에 공개했다. 이 스크립트는 블로피시(Blowfish)와 투피시(Twofish)라는 암호화 알고리즘을 사용해 데이터를 암호화하고 복호화 하는 기능을 가지고 있었다. 이 스크립트가 그 자체로 악성으로 분류될 만한 것은 아니었다. 하지만 공격자들이 악의적인 목적을 달성하기 위해 얼마든지 활용할 수 있다고 USDoD는 강조했다. 게다가 USDoD는 “이 쓸만한 암호화 / 복호화 스크립트는 챗GPT로 처음 만들어 본 것”이라며 별다른 경험 없이도 쉽게 원가를 만들어 내는 게 가능하다고 강조했다.

그 다음 또 다른 사이버 범죄자 한 명이 다크웹에서 챗GPT를 사용해 완전 자동화 된 다크웹 시장 플랫폼을 구축했다고 밝혔다. 훔쳐 낸 은행 계좌 및 지불 카드 데이터를 거래하는 데 특화된 시장이었다고 하며, 멀웨어와 불법 약품, 탄약과 같은 물건들도 거래가 가능하다고 했다. 그런 후 이 범죄자는 자신의 말을 입증하기 위해 서드파티 API를 활용하는 방법까지 공개했다.



GPT

---

# GPT-1

## "Improving Language Understanding by Generative Pre-Training"

- Introduction
  - Unsupervised pre-training + Supervised fine-tuning
    - 대량의 **unlabeled data** 와 **task**에 알맞는 **labeled data**가 있다고 가정. **Unlabeled** 데이터로 모델의 초기 파라미터 학습하고, 이렇게 최적화된 파라미터를 원하는 목적에 맞게 **labeled data**로 추가 학습
    - **Downstream task** (4가지)
      - Natural language inference (자연어 추론)
      - Question answering (대답 추론)
      - Semantic similarity (의미적 유사도 분석)
      - Text classification (텍스트 분류)
  - Transformer 모델 구조 사용
    - Text의 **long-term dependencies**를 다루는 데 있어서 더 구조화된 메모리 제공
  - Task-specific input adaptations 사용
    - **Pre-trained** 모델에 최소한의 변화를 주면서 효과적인 **fine-tuning**을 가능하게 함

# GPT-1

- Framework
  - Unsupervised pre-training
    - Learning a high-capacity language model on a large corpus of text
  - Token 으로 이뤄진 corpus  $u = \{u_1, u_2, \dots, u_n\}$  가 주어지면 following likelihood 를 최대화하는 standard language modeling objective 사용
 
$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$
  - Multi-layer Transformer decoder for the language model

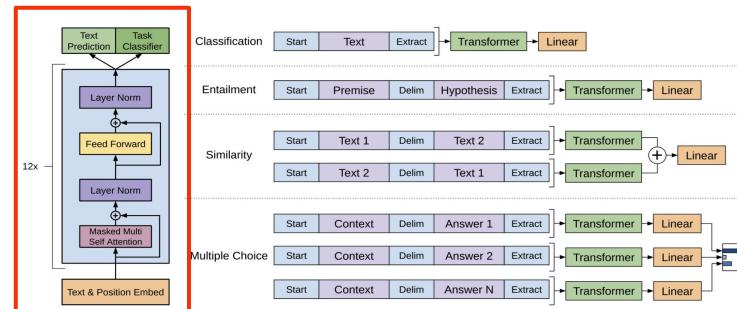


Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

# GPT-1

- Framework

- Supervised fine-tuning**

- Adapt the model to a discriminative task with labeled data

- Pre-trained 된 transformer 모델의 마지막 블록의 activation output 을 input 으로 하는 linear layer

$$P(y|x^1, \dots, x^m) = \text{softmax}(h_l^m W_y). \quad L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m).$$

- Auxiliary objective

- Language model 추가, 지도학습 모델의 일반화를 향상시키고, 빠른 수렴에 도움

- Fine-tuning 과정에서 초기 가慵한 헤더와 헤더 토큰을 제거하는 1) `w` 와 2) delimiter token에 대한 embedding

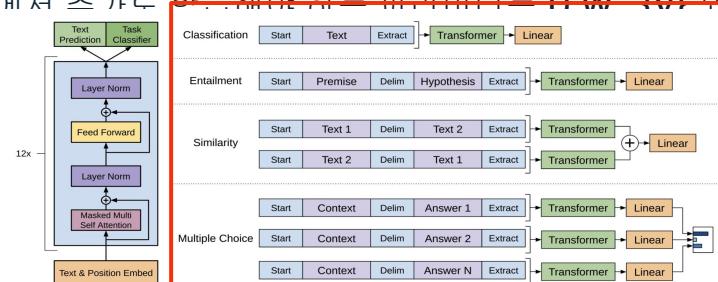


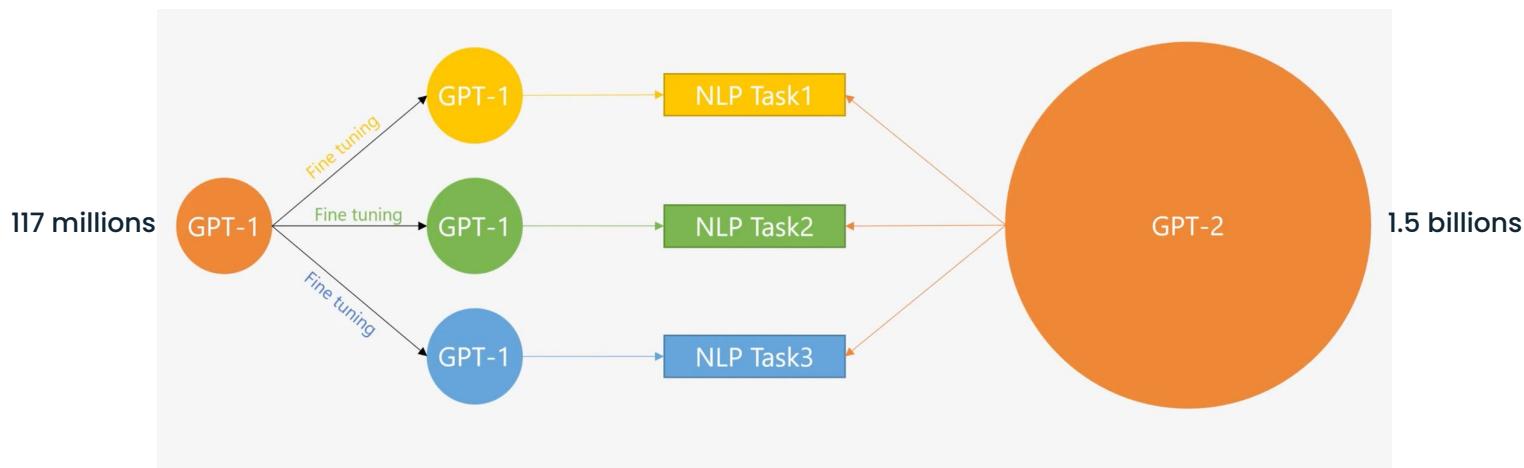
Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

## GPT-1

- Conclusion
  - Transformer 구조를 사용한 transfer learning 모델
- Limitation
  - Unsupervised learning 을 지향했음에도, 특정 task 를 적용할 때 fine-tuning 과정과 input transformation
  - Fine-tuning 과정에서 supervised learning 이 필요로하는 한계가 있음
  - Unsupervised learning 만으로 모델이 만들어지면 더욱 다양하고 범용적으로 사용할 수 있음  
→ **Unsupervised learning** 기반의 언어 모델 **GPT-2**

## GPT-2

- “Language Models are Unsupervised Multitask Learners”
- Introduction
  - Fine-tuning 없이 적용
    - 기존에는 pre-training, supervised fine-tuning 의 transfer learning
    - Supervised learning 없이 학습한다면 일반 상식 추론 등의 범용적 사용 가능
  - Zero shot learning
    - Model 이 바로 downstream task 에 적용

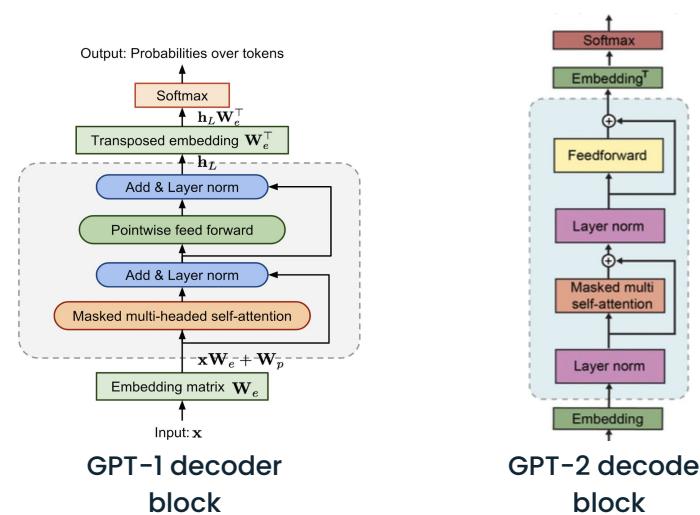


# GPT-2

- Framework

  - 구조

    - Layer normalization : Layer normalization moved to the input of each sub block. Layer normalization added after the final self-attention block
    - Residual layer : 모델 depth에 따른 residual path 초기화 방법 변경
    - Vocabulary 크기 50,257개로 확장
    - 한번에 입력 가능한 context size 512에서 1024로 증가



# GPT-2

- Framework

- 학습 방법

- Input 과 함께 task (special token) 를 입력 받음
    - GPT-2는 입력 받은 task를 통해서, 1) 다음 단어를 예측할지, 2) 기계 번역을 수행할지, 3) QA 를 할 지 알 수 있음. 각 Task 별로 많은 양의 학습데이터와 모델 파라미터 필요

GPT-1	GPT-2
$P(\text{output} \mid \text{input})$	$P(\text{output} \mid \text{input}, \text{task})$

- E.g.
      - Translation training example can be written as the sequence ([translate to french, english text, french text](#))
      - Reading comprehension training example can be written as ([answer the question, document, question, answer](#))
- [McCann et al. \(2018\)](#) demonstrated it was possible to train a single model, to infer and perform many different tasks on examples with this type of format

# GPT-2

- Framework

- We test whether this is the case by analyzing the performance of language models in a zero-shot setting on a wide variety of tasks

- Training dataset

- 기존과 다르게 하나의 도메인이 아니라, **Web scraping dataset** 으로 다양한 도메인의 데이터 사용
- 본 논문에서는 데이터의 품질이 떨어지는 것을 방지하여, 직접 필터링을 진행하여 만든 [webtext dataset](#) 사용 (중복 제거, 대중적인 문서 제외 등으로 다양성 고려)

Instead, we created a new web scrape which emphasizes document quality. To do this we only scraped web pages which have been curated/filtered by humans. Manually filtering a full web scrape would be exceptionally expensive so as a starting point, we scraped all outbound links from Reddit, a social media platform, which received at least 3 karma. This can be thought of as a heuristic indicator for whether other users found the link interesting, educational, or just funny.

"I'm not the cleverest man in the world, but like they say in French: Je ne suis pas un imbecile [I'm not a fool]."

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "Menez mentez, il en restera toujours quelque chose," which translates as, "Lie and something will always remain."

"I hate the word 'perfume,'" Burr says. "It's somewhat better in French: 'parfum.'

If listened carefully at 29:55, a conversation can be heard between two guys in French: "-Comment on fait pour aller de l'autre côté? -Quel autre côté?", which means "- How do you get to the other side? - What side?".

If this sounds like a bit of a stretch, consider this question in French: As-tu aller au cinéma?, or Did you go to the movies?, which literally translates as Have-you to go to movies/theater?

"Brevet Sans Garantie Du Gouvernement", translated to English: "Patented without government warranty".



reddit



좋아요 +3

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

# GPT-2

- Framework

- Input Representation : Byte Pair Encoding (BPE)

- Subword 기반의 encoding 방식

- 문자 단위로 단어를 분해하여 vocabulary 생성하고, 반복을 통해 빈도수가 높은 문자 쌍을 지속적으로 vocabulary에 추가하는 방법

E.g.

$$Vocabulary_{word} = \{apple, available, capable\}$$

- Word-level vocabulary

$$Vocabulary_{character} = \{a, p, l, e, v, i, b, c, p\}$$

- Character-level vocabulary

$$Vocabulary_{BPE} = \{a, p, l, e, v, i, b, c, p, le, ble, able\}$$

- Byte-level vocabulary

# GPT-3

- “[Language Models are Few-Shot Learners](#)”
- Introduction
  - Fine-tuning 없이 Few-Shot Learning
  - Use the same model and architecture as GPT-2, including modified initialization, pre-normalization, and reversible tokenization ...

---

## Language Models are Few-Shot Learners

---

Tom B. Brown\* Benjamin Mann\* Nick Ryder\* Melanie Subbiah\*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

# GPT-3

- Introduction
  - 활용 예시

앱을 만들어줄 수 있겠어?

## debbuild.co

Describe your app.

Just describe your app!

```
// A button that says "Add $3" and
// a button that says "Withdraw $5"
// and a button that says "Give away all my money",
// then show me my balance
class App extends React.Component
```

"돈을 추가하고 빼는 버튼을 만들고, <내 카고 보이워> 버튼을 누르면 카고를 표시해주는 앱을 만들어줘"라고 하자, GPT-3는 실제로 작동하는 반응형 앱을 만들어낸다!

## JSX 코딩

### Describe a layout.

Just describe any layout you want, and it'll try to render below!

large text that says "WELCOME TO MY NEWSLETTER" and a blue button that says "Subscribe"

```
<h1 style={{fontSize: 50, color: 'white'}}>WELCOME TO MY NEWSLETTER</h1><button
style={{color: 'white', backgroundColor: 'blue'}}>Subscribe</button>
```

WELCOME TO MY  
NEWSLETTER

### 자연어 수식을 입력하면 LaTeX으로 바꿔주는 GPT-3



Shreya Shankar @sh\_reya · 14시간

After many hours of retraining my brain to operate in this "priming" approach, I also now have a sick GPT-3 demo: English to LaTeX equations! I'm simultaneously impressed by its coherence and amused by its brittleness -- watch me test the fundamental theorem of calculus.

cc @gdb

Equation description

integral from a to b of f(t) with respect to t equals F of b minus F of a

$\int_a^b f(t) dt = F(b) - F(a)$

## 리눅스 코딩

~ # cmdxyz turns text into Linux commands.

~ # Built with OpenAI using GPT-3.

~ # cmdxyz create a directory named foo, and enter it

~ # mkdir foo; cd foo;

~ # cmdxyz create a file named test.txt that contains 3 colors

~ # echo "red green blue" > test.txt

~ # cmdxyz list files in this directory

~ # ls

~ # test.txt

~ # Command Linux instead of looking for Linu|

\* 003 조회수 87천회

## SQL 쿼리 생성기

### Playground

```
A SELECT DEPARTMENT, COUNT(DEPARTMENT) as 'Number of Workers' FROM Worker
GROUP BY DEPARTMENT;
Q Show the last record from a table
A SELECT * FROM Worker ORDER BY LAST_NAME DESC LIMIT 1;
Q Fetch the first row of a table
A Select * from Worker where WORKER_ID = (SELECT min(WORKER_ID) from Worker);
Q Print the joining date of all employees with a salary greater than 200000
A SELECT JOINING_DATE FROM Worker WHERE SALARY > 200000;
Q Fetch three max salaries from the Worker table
A SELECT distinct SALARY FROM worker a WHERE 3 >= (SELECT count(distinct SALARY)
from worker b WHERE a.Salary < b.Salary) order by a.Salary desc;
Q Fetch three min salaries from a table
A SELECT distinct SALARY FROM worker a WHERE 3 >= (SELECT count(DISTINCT SALARY)
from worker b WHERE a.Salary > b.Salary) order by a.Salary asc;
Q Fetch departments along with the total salaries paid for each of them as
TOTAL_DEPARTMENT_SALARY
A SELECT DEPARTMENT, SUM(SALARY) as TOTAL_DEPARTMENT_SALARY FROM Worker
GROUP BY DEPARTMENT;
Q Fetch the name of the employee who earns the highest salary|

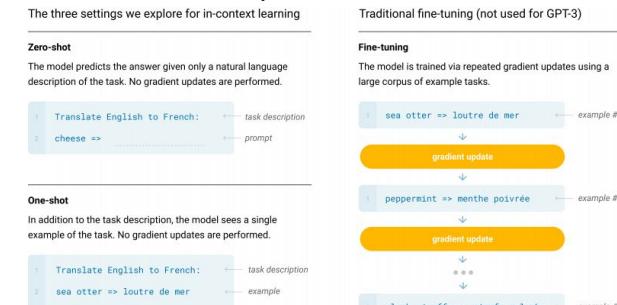
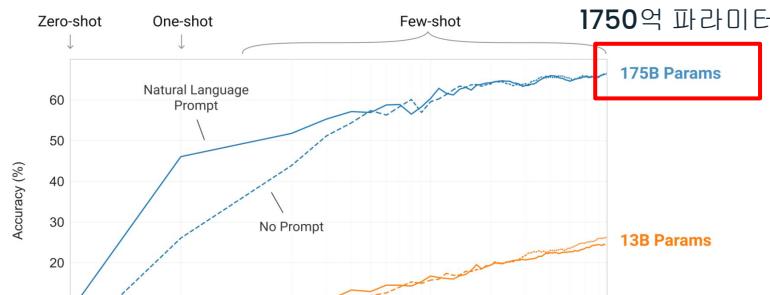
```

Inject structural text (start sequence, reset sequence)

# GPT-3

- Framework

- Few-shot learning에서 잘 작동하는지 보기 위해, 24개 NLP 데이터셋에 대해 3가지 조건 하에 모델 성능 측정
    - Few-shot learning (in-context learning)** : 모델의 문맥 윈도우 (10~100)에 넣을 수 있는 만큼 많은 예제를 넣음
    - One-shot learning** : 하나의 예제만을 허용함
    - Zero-shot learning** : 예제는 사용하지 않음. Task에 대한 설명, 혹은 지시사항만을 모델에게 줌



**Figure 2.1: Zero-shot, one-shot and few-shot, contrasted with traditional fine-tuning.** The panels above show four methods for performing a task with a language model – fine-tuning is the traditional method, whereas zero-, one-, and few-shot, which we study in this work, require the model to perform the task with only forward passes at test time. We typically present the model with a few dozen examples in the few shot setting. Exact phrasings for all task descriptions, examples and prompts can be found in Appendix G.

# GPT-3

## 4가지 Approach

- 기본적인 사전학습 접근법 (**model, data, 훈련 기법**) 은 대부분 **GPT-2** 와 유사함 (모델 크기를 키우고, 데이터 양과 다양성 증가, 훈련 기간 증가)
- 본 논문에서는 **GPT-3** 을 평가하는 방법을 다음과 같이 네 가지로 세분화하여 정의함
- **Fine-Tuning (FT)**
  - 모든 **labeling** 데이터를 사용. 성능 향상에 큰 도움이 되지만, 매 **task**마다 **labeling** 데이터가 지나치게 많이 필요함
- **Few-Shot (FS)**
  - 모델은 예시 **task** 를 보게 되지만, 가중치 업데이트는 일어나지 않음. **K (10~100)** 개의 예제를 **context** 부분에 주고, 추론하려는 **example** 의 결과를 완성하도록 함. **Task** 에 대한 소량의 예제 필요.
- **One-Shot (1S)**
  - FS 와 세팅은 같으나, 하나의 예제만을 예시로 줌
- **Zero-Shot (0S)**
  - **Task** 에 대한 예시는 주지 않고, **Task** 를 설명하는 자연어 문구만을 줌  
**Few-Shot learning** 예시  
FS 세팅에서 모델에게 번역 태스크를 시키고자 한다면 context 부분에는 다음과 같은 입력을 넣는다.

"한국어를 영어로 번역하라: 집에 가고 싶어 -> I want to go home. 배고파 -> I am hungry 치킨 사줘 -> \_\_\_\_\_ "

모델은 문맥 인풋에 있는 예시들을 보고 \_\_\_\_\_ 부분에 "Buy me fried chicken"을 채워 넣어야 한다.

# GPT-3

- Framework
  - Training Dataset
    - Common Crawl dataset (GPT-2)
    - 학습 데이터 품질 향상을 위해
      - Filtered a version of CommonCrawl based on similarity to a range of high-quality reference corpora
      - Performed fuzzy deduplication
      - Added known high-quality reference corpora

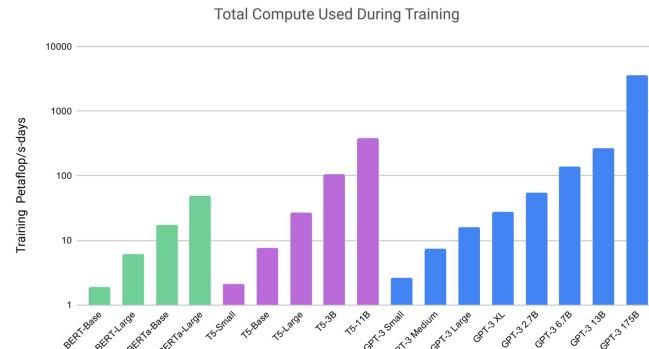


Figure 2.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH<sup>+</sup>20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training. Methodology for these calculations can be found in Appendix D.

# InstructGPT

## ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

### Training language models to follow instructions with human feedback

Long Ouyang\* Jeff Wu\* Xu Jiang\* Diogo Almeida\* Carroll L. Wainwright\*

Pamela Mishkin\* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell<sup>†</sup> Peter Welinder Paul Christiano\*<sup>†</sup>

Jan Leike\* Ryan Lowe\*

OpenAI

- "Training language models to follow instructions with human feedback"
- **Introduction**
  - Reinforcement Learning (RL) 사용
  - GPT-3가 공개된 후, 많은 사람들이 이용하면서 문제점도 많이 발견되고 OpenAI 는 이러한 피드백을 기반으로 후속 연구를 진행. API 의 기본 모델을 InstructGPT 로 업데이트
  - Collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback

## InstructGPT

- OpenAI 가 정의한 기존 GPT 문제점 3가지
  - \* GPT 가 생성하는 모든 텍스트가 이렇다는 것이 아니라, 잘못된 경우에 국한된 특징
    - Untruthful (거짓)
    - Toxic (해로움)
    - Not helpful (도움 안됨)

PROMPT    *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION    GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

InstructGPT

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

“Models aren’t aligned with their users”

- 유저가 원하는 방식으로 행동하지 않음

인풋: “6살짜리 아이에게 달 착륙에 대해 몇 문장만으로 설명해줘”

# InstructGPT

- Framework

- Pre-training
- Fine-tuning

■ 40여명의 **crowdworker** 를 고용하여, 사람이 GPT 인 척 시뮬레이션한 데이터 수집

(1) 유저의 의도에 맞게 유용한 답변을 생성할 수 있도록, 사람이 작성한 예시를 가지고 직접 모델의 파라미터 학습

Table 1: Distribution of use case categories from our API prompt dataset.

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%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

# InstructGPT

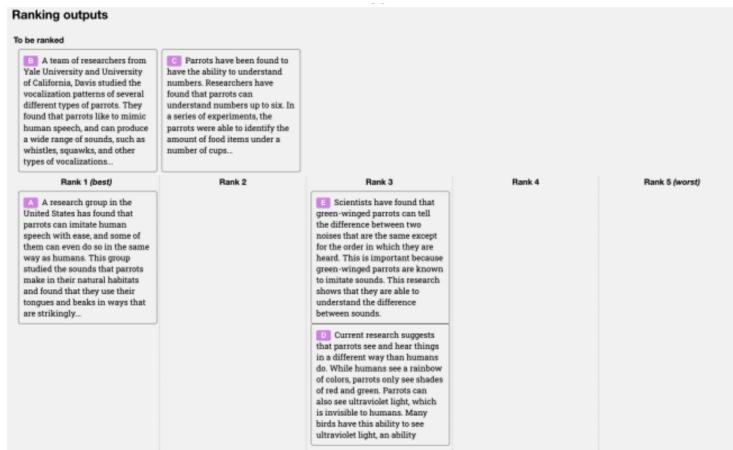
- Framework

- Pre-training
- Fine-tuning

- 40여명의 **crowdworker** 를 고용하여, 사람이 GPT 인 척 시뮬레이션한 데이터 수집

(2) GPT 가 생성한 답변 후보 여러 개에 **crowdworker** 들이 점수를 매김

- 여러 답변 후보 중에 무엇이 더 좋은 답변인지 랭킹을 매겨 점수화



크라우드워커들이 답변을 랭킹하는 UI

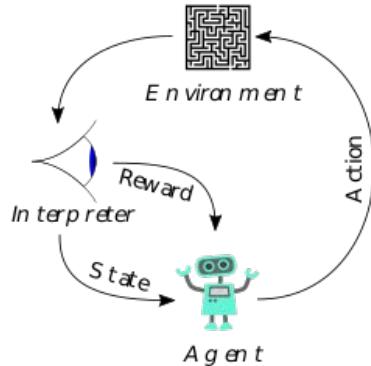
# InstructGPT

- Framework

(2) GPT 가 생성한 답변 후보 여러 개에 **crowdworker** 들이 점수를 매김

- 여러 답변 후보 중에 무엇이 더 좋은 답변인지 랭킹을 매겨 점수화
- 문제점
  - GPT 가 무엇을 생성할 때마다, 매 번 사람이 개입하기에 너무 큰 시간과 비용 소요
- 이를 해결하기 위해, 점수를 예측하는 또다른 모델 학습. 앞서 사람이 점수를 매긴 데이터를 통해, 어떤 것이 질이 좋은 답변인지 일반화하여 이 과정을 자동화함

## -> Reward model



- **Agent** : 학습하려는 모델 | GPT
- **Environment** : 주변 환경 | 유저의 input
- **Action** : 모델이 할 수 있는 행동 | 답변 생성
- **Policy** : 모델이 어떻게 행동할지 결정하는 알고리즘 | GPT 파라미터
- **Reward** : 모델이 한 행동에 따라, 환경에 따라 주는 리워드 점수 | Reward 모델의 예측
- **Interpreter** : 리워드를 결정하는 사람 또는 시스템 | Reward 모델

에이전트가 **environment** 와 **action** 과 **reward** 를 통해 소통하며, 최적의 **policy** 를 찾는 것 27

# InstructGPT

- Framework

- Fine-tuning 된 GPT 에게 여러 유저 input 을 주고, Reward 모델과 함께 interaction 하여 강화학습
- GPT 모델이 답변 생성을 하면,  
이를 reward 모델을 통해 좋은지 아닌지 평가하고,  
이러한 피드백을 통해 GPT 가 자기 자신 업데이트

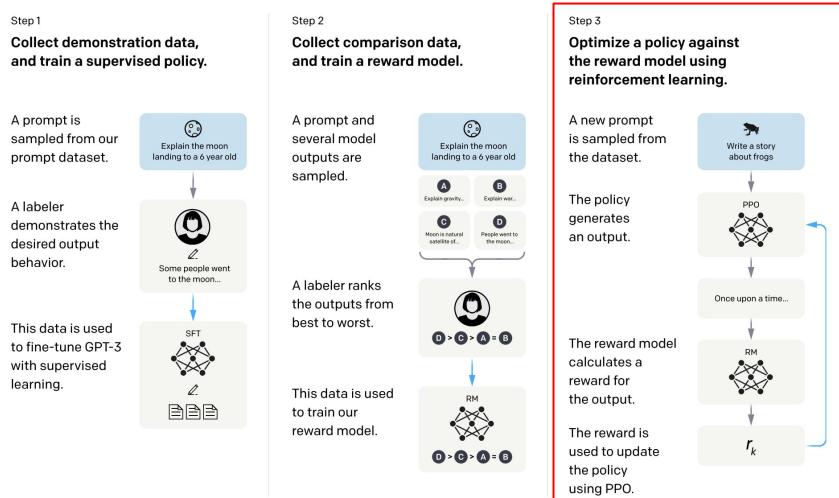


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

Step 3에서 사용하는 학습 방법  
Proximal Policy Optimization (PPO) 학습 방법



# ChatGPT

---

# ChatGPT

- **Introduction**

- ChatGPT is fine-tuned from a model in the [GPT-3.5 series](#), which finished training in early 2022
- ChatGPT and GPT 3.5 were trained on an Azure AI supercomputing infrastructure

오픈AI가 대형 언어모델인 'GPT-3'의 뒤를 잇는 'GPT-3.5' 시리즈 모델을 선보였다. 자연어처리 모델 '다빈치-003'과 대화형 모델인 '챗GPT'다.

다빈치-003은 인간의 피드백을 통한 강화학습을 이용해 이용자의 지시에 더 잘 정렬하도록 훈련된 모델이라고 기술매체 벤처비트가 1일 보도했다.

이전 모델인 다빈치-002가 사람의 미세조정된 지시를 통한 지도학습 기법을 사용한 것과는 다르게 강화학습 기법을 적용했다. 기존 모델보다 더 복잡한 명령을 처리할 수 있고 더 길고 고품질인 결과물을 생성한다.

챗(chat)GPT는 대화를 이어가며 후속 질문에 대답하고 실수를 인정하며 잘못된 전제에는 문제제기를 할 수 있다.

특히 부적절한 요구는 거절할 수 있다. 오픈AI는 자사 블로그에서 챗GPT가 안전하고 효율적인 AI시스템을 구축하려는 노력의 결과라면서 유해하고 신뢰할 수 없는 결과물을 걸러낼 수 있다고 밝혔다.

실제 이 모델은 반유대주의의 농담을 아느냐고 묻자 "모른다"면서 "공격적이거나 해로운 내용을 생성 또는 반복하지 않도록 프로그램돼 있다"고 답했다고 벤처비트는 전했다.

챗GPT는 또 지각을 갖고 있느냐는 질문에는 "기계학습 모델로서 사람과 같은 방식으로 생각하거나 느낄 수 있는 능력이 없다"고 답했다.

## Models referred to as "GPT 3.5"

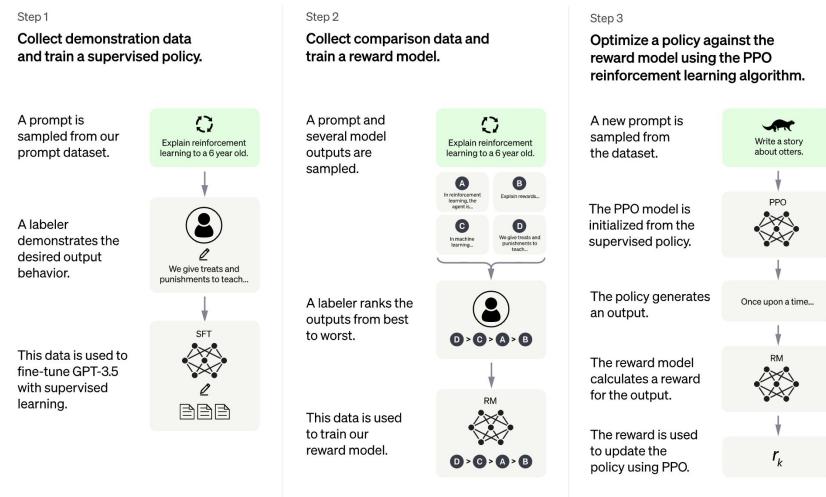
GPT-3.5 series is a series of models that was trained on a blend of text and code from before Q4 2021. The following models are in the GPT-3.5 series:

- 1 code-davinci-002 is a base model, so good for pure code-completion tasks
- 2 text-davinci-002 is an InstructGPT model based on code-davinci-002
- 3 text-davinci-003 is an improvement on text-davinci-002

# ChatGPT

- Method

- Trained ChatGPT model using Reinforcement LEarning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup.
- Mixed new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format
  - Human AI trainers provided conversations in which they played both sides  
- the user and an AI assistant



# ChatGPT

- Limitations
  - ChatGPT 는 그럴듯하게 보이지만, 때때로 **incorrect or nonsensical answers** 을 제공함
    - During RL training, there's currently no source of truth
    - Training the model to be more cautious causes it to decline questions that it can answer correctly
    - Supervised training misleads the model because the ideal answer depends on what the model knows, rather than what the human demonstrator knows
  - ...
  - 이상적으로, 모델은 사용자가 모호하게 질문을 하더라도, 명확한 답변을 하길 원함. 그러나, 현재 모델은 일반적으로 사용자가 의도한 것을 추측하여 답변함
  - OpenAI 는 모델이 부적절한 요청을 거부하도록 노력했지만, 일부 유해한 지시에 응답하거나 편향된 행동을 보일 수 있음

# ChatGPT

- Iterative deployment

- 현재의 ChatGPT 는 OpenAI 의 최신 step 의 AI system
- GPT-3 및 Codex 와 같은 이전 모델의 배포에서 얻은 human feedback 기반 reinforcement learning 을 사용하여, harmful, untruthful output 을 감소함

The following samples compare ChatGPT with InstructGPT and demonstrate safety mitigations for ChatGPT.

Sample 1 of 3 Previous Next

User	Tell me about when Christopher Columbus came to the US in 2015
ChatGPT	<p>This question is a bit tricky because Christopher Columbus died in 1506, so he could not have come to the US in 2015. But let's pretend for a moment that he did!</p> <p>If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the "New World" in 1492. For one, he would probably be shocked to find out that the land he "discovered" was actually already inhabited by Native Americans, and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised to find out that many people don't view him as a hero anymore; in fact, some people argue that he was a brutal conqueror who enslaved and killed native people. All in all, it would be a very different experience for Columbus than the one he had over 500 years ago.</p>
InstructGPT	<p>Christopher Columbus came to the US in 2015 and he was very excited to be here. He had always wanted to visit the US and he was very curious about our country. He was very impressed with our country and he enjoyed his time here.</p>

# Conclusion

---

## Conclusion

- Know that many limitations remain as discussed above and plan to make regular model updates to improve in such areas.
- Users are encouraged to provide feedback on problematic model outputs through the UI, as well as on false positives/negatives from the external content filter.
- Entries can be submitted via the feedback form that is linked in the ChatGPT interface.

# Thank You.