

ChatGPT API 마스터 클래스: 진보된 자연어 처리와 Fine-tuning

1. 자연어 처리의 진화
2. ChatGPT API 기초 및 활용법
3. Fine-tuning의 개념과 목적
4. ChatGPT의 미래



GPT ?

- Generative (다음단어 예측, 언어모델)
- Pre-trained (사전훈련)
- Transformer (인공신경망에 기반한 Encoder-Decoder)

chatGPT ?

- 대화형으로 기존 GPT를 Fine-tuning

ChatGPT ?

- 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.
-
- 대화형 인터페이스
 - 질문, 실수 인정, 잘못된 전제에 하여 이의 제기, 부적절한 요청 거부
 - **InstructGPT**의 형제모델

InstructGPT

- 딥러닝 기반의 언어 생성 모델
- GPT 3.5 모델로부터 추가로 학습한 모델 (GPT3.5, trained on a blend text and code from before Q4 2021)
- RLHF(Reinforcement Learning by Human Feedback)
 - > SFT (Supervised Fine-Tuning)
 - > RM (Reward Model)
 - > PPO (Proximal Policy Optimization) Algorithm

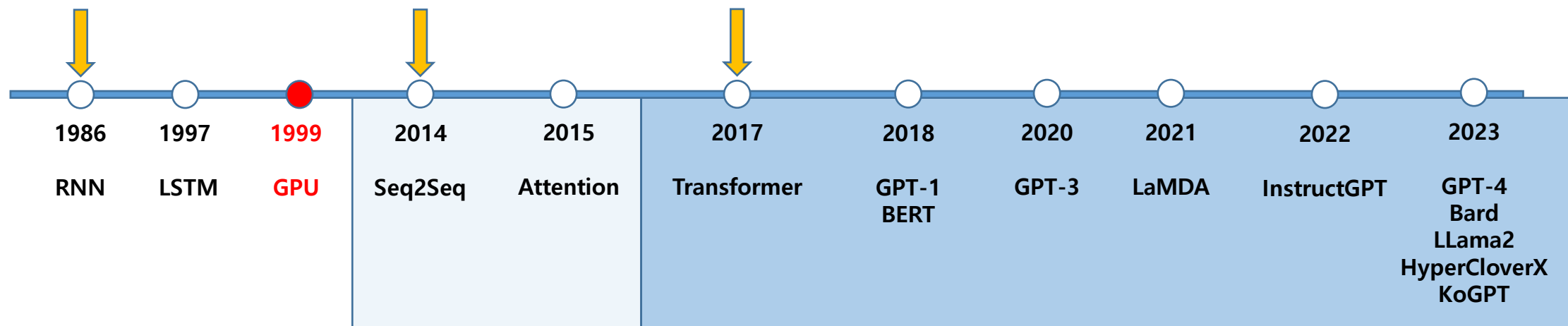
ChatGPT

- 딥러닝 기반의 언어 생성 모델
- GPT 3.5 모델로부터 추가로 학습한 모델 (GPT3.5, trained on a blend text and code from before Q4 2021)
- RLHF(Reinforcement Learning by Human Feedback)
 - > SFT (Supervised Fine-Tuning)
 - > RM (Reward Model)
 - > PPO (Proximal Policy Optimization) Algorithm
- 대화형 에이전트에 최적화 (Optimizing Language Models for Dialogue)

현재, 자연어처리 분야에서 고성능 모델들은 Transformer 아키텍처 기반

GPT: Transformer의 디코더(Decoder) 아키텍처 활용

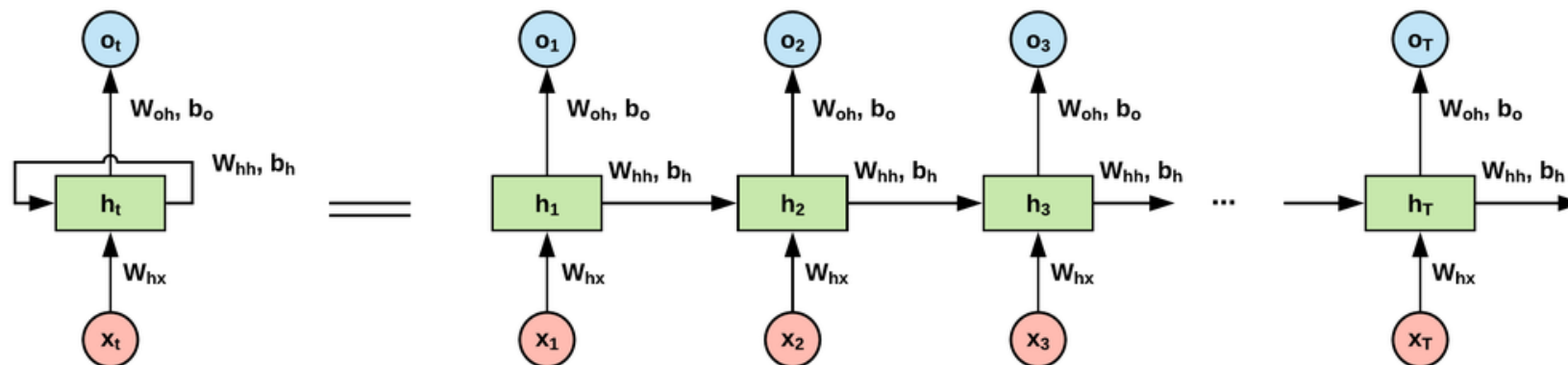
BERT: Transformer의 인코더(Encoder) 아키텍처 활용



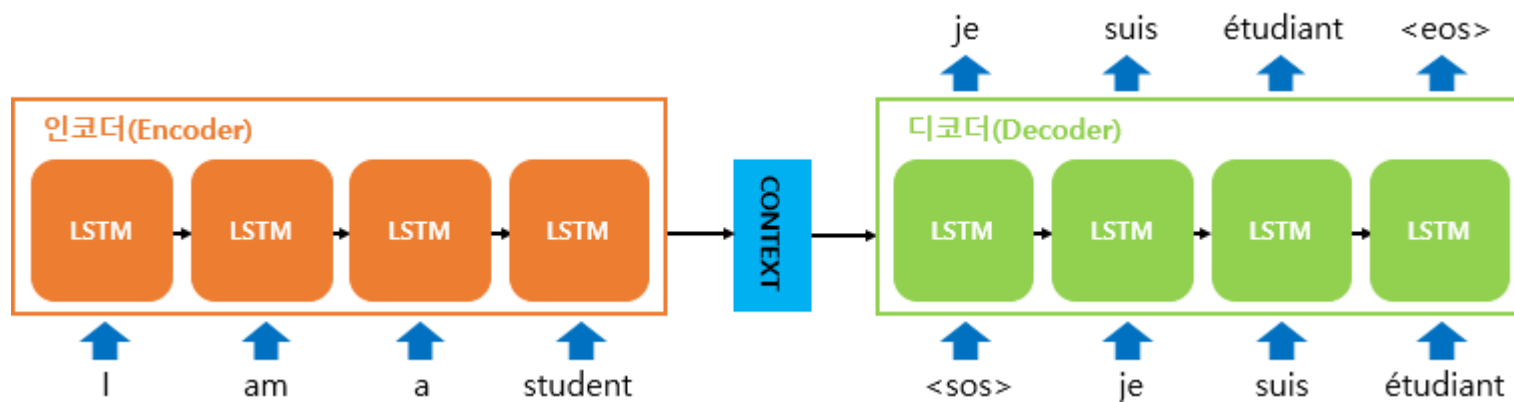
입력 시퀀스 전체에서 정보를 추출하는 방법으로 발전

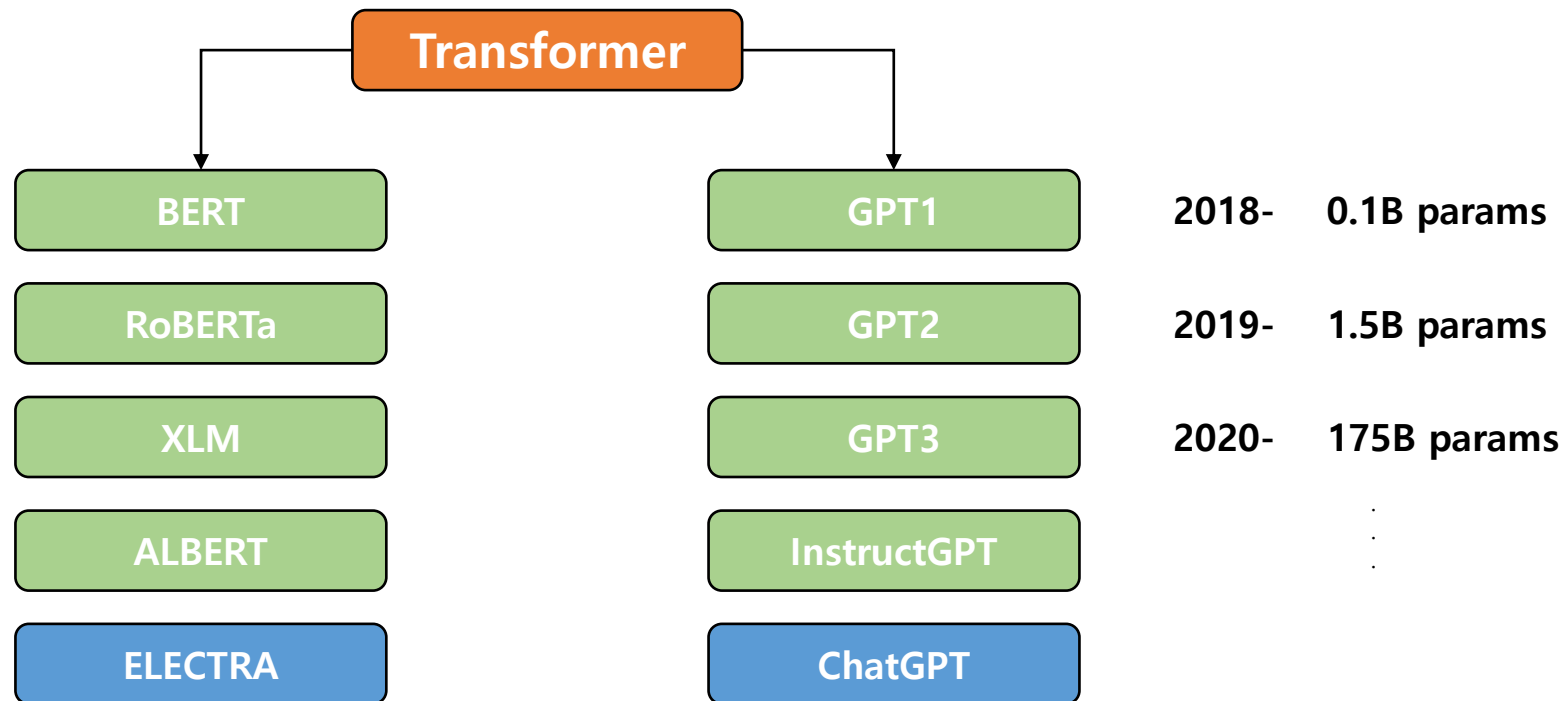


RNN (순환 인공 신경망)

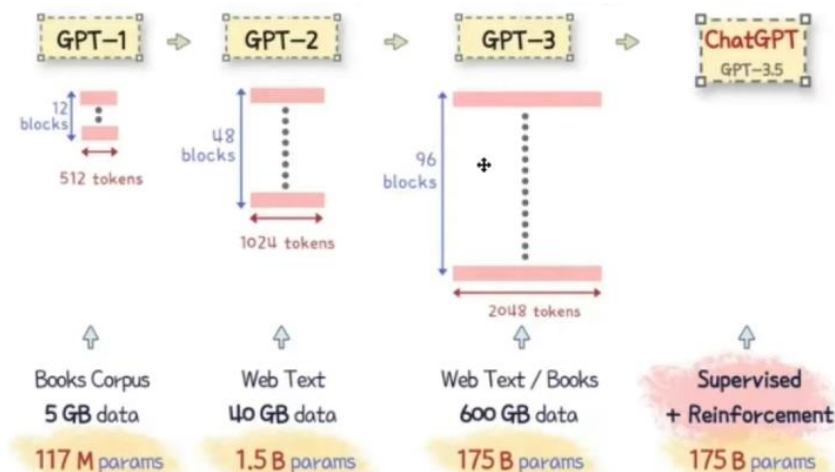
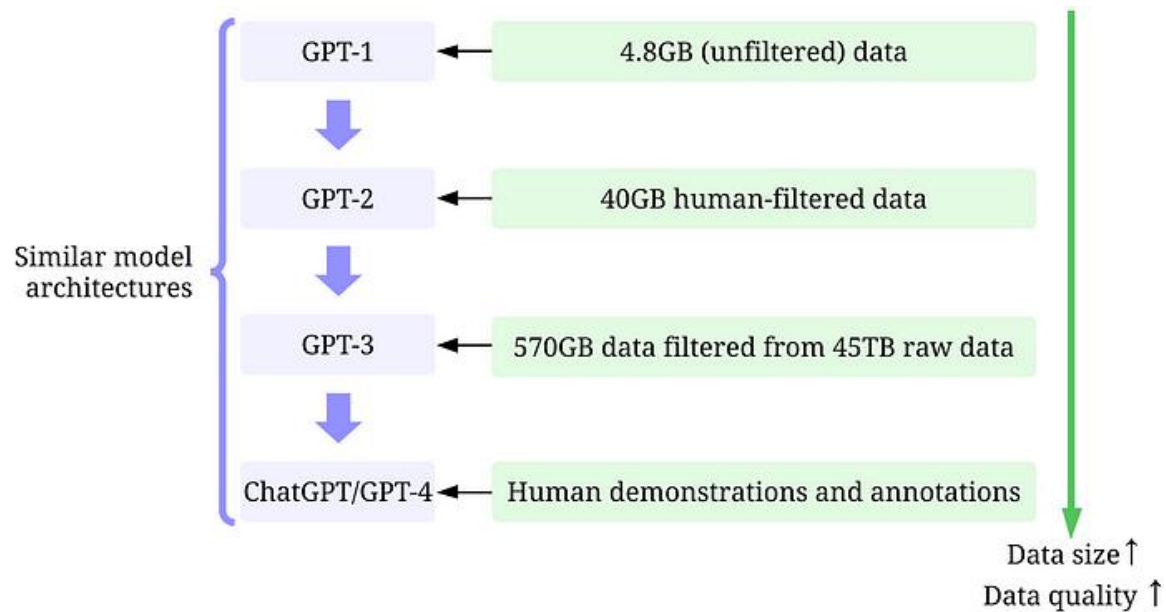


Seq2seq (Sequence-to-Sequence)

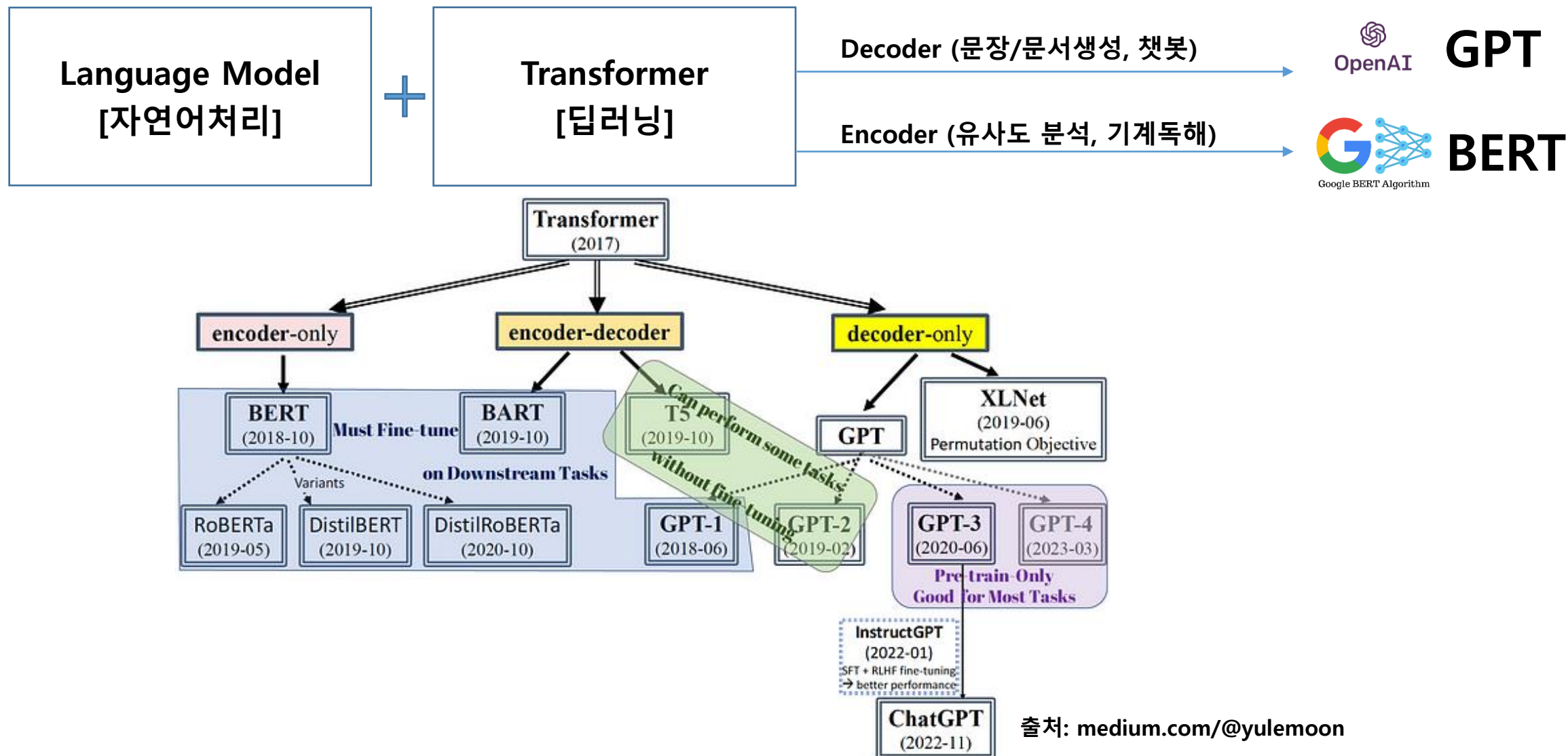




Model Progression



	GPT-1	GPT-2	GPT-3
Parameters	117 million	1.5 billion	175 billion
Decoder layers	12	48	96
Context length	512	1024	2048
Hidden layer size	768	1600	12288

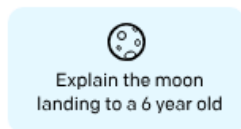


ChatGPT 학습 순서 (SFT – RM – PPO)

Step 1

**Collect demonstration data,
and train a supervised policy.**

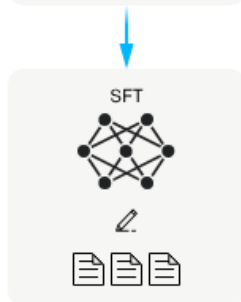
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



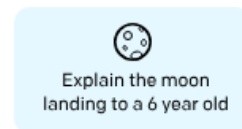
This data is used
to fine-tune GPT-3
with supervised
learning.



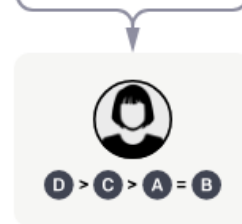
Step 2

**Collect comparison data,
and train a reward model.**

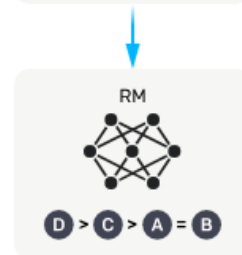
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



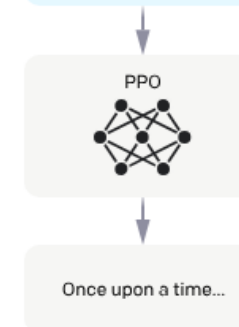
Step 3

**Optimize a policy against
the reward model using
reinforcement learning.**

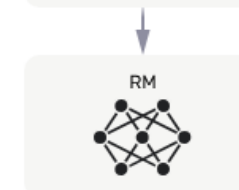
A new prompt
is sampled from
the dataset.



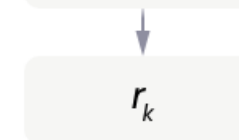
The policy
generates an output.



The reward model
calculates a
reward for the output.



The reward is
used to update
the policy
using PPO.



GPT API?

**GPT 모델을 다양한 서비스에 적용시킬 수 있도록,
Openai에서 공개한 언어모델 API 서비스**

Newer : GPT-3.5 Turbo / GPT-4

Legacy : text-davinci-003, text-davinci-002, code-davinci-002

HTTP 프로토콜 기반의 Rest api

- (1) 가이드 제공언어: Python, Node JS
- (2) 라이브러리: openai, gpt-3-api, gpt-3
- (3) 라이브러리 호환언어: Python, Node JS, Go, JAVA, C++, Rust, Swift, Ruby, PHP
- (4) API 주요기능: 텍스트 생성/분석/추론, 언어 번역, 코드 작성, 질문에 대한 답변
- (5) API 응용서비스: 챗봇(고객서비스, 게임 등), 콘텐츠 생성(기사, 블로그 게시물), 교육/연구(자연어, 비전, 기계학습 등)
- (6) 개발환경 설정:
 - OpenAI 회원가입 및 API Key 발급
 - Back-end 개발환경 설정 및 openai 라이브러리 설치
 - GPT model 설정, REST API 연동 테스트

openai.com 접속 후, Menu > Overview 선택

Transforming work and creativity with AI

Our API platform offers our latest models and guides for safety best practices.

[Get started ↗](#)

[For developers ↓](#)



Welcome to the OpenAI platform

Start with the basics

Quickstart tutorial

Learn by building a quick sample app

Examples

Explore some example tasks

Build an application



GPT

Learn how to generate text and call functions



Embeddings

Learn how to search, classify, and compare text



Image generation

Learn how to generate or edit images



GPT best practices

Learn best practices for building with GPT models



Speech to text

Learn how to turn audio into text



Fine-tuning

Learn how to train a model for your use case

Build a ChatGPT plugin



Introduction

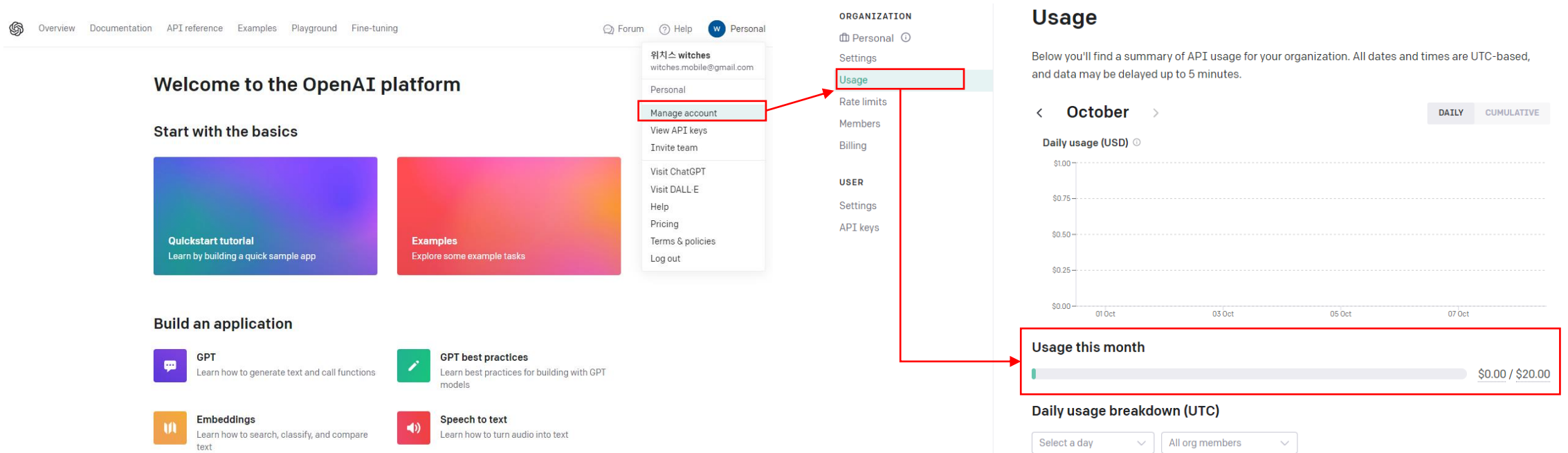
Learn the basics of building a ChatGPT plugin



Examples

Explore ChatGPT plugin examples

Menu > Management account > Usage 선택



The screenshot shows the OpenAI platform interface. The top navigation bar includes links for Overview, Documentation, API reference, Examples, Playground, and Fine-tuning. The left sidebar contains a 'Welcome to the OpenAI platform' section with 'Start with the basics' (Quickstart tutorial, Examples) and 'Build an application' (GPT, GPT best practices, Embeddings, Speech to text, Image generation, Fine-tuning). The right sidebar shows the 'ORGANIZATION' menu with 'Personal' selected. The 'Usage' option is highlighted in the 'ORGANIZATION' menu. The main content area shows the 'Usage' page for the month of October, with a 'Usage this month' section showing \$0.00 / \$20.00.

- Free 버전의 경우, 매달 5\$ credit 제공
- GPT 3.5 Turbo는 1,000 token당 0.002 달러 (토큰계산: platform.openai.com/tokenizer)

<https://platform.openai.com/tokenizer>

Menu > View API Keys 선택

Overview Documentation API reference Examples Playground Fine-tuning

Forum Help Personal

ORGANIZATION

Personal

Settings

Usage

Rate limits

Members

Billing

USER

Settings

API keys

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED	
WitchesTest	sk-...LY1B	2023년 6월 16일	2023년 8월 21일	
nongnetTest	sk-...jNoH	2023년 7월 20일	2023년 7월 20일	
chatbotTest	sk-...0bC7	2023년 8월 10일	2023년 9월 8일	
JNU1	sk-...ynQW	2023년 8월 14일	2023년 8월 23일	
JNU2	sk-...x7wQ	2023년 8월 14일	2023년 8월 23일	
JNU3	sk-...GLIJ	2023년 8월 14일	2023년 8월 23일	

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED	
WitchesTest	sk-...LY1B	2023년 6월 16일	2023년 8월 21일	
nongnetTest	sk-...jNoH	2023년 7월 20일	2023년 7월 20일	
chatbotTest	sk-...0bC7	2023년 8월 10일	2023년 9월 8일	
JNU1	sk-...ynQW	2023년 8월 14일	2023년 8월 23일	
JNU2	sk-...x7wQ	2023년 8월 14일	2023년 8월 23일	
JNU3	sk-...GLIJ	2023년 8월 14일	2023년 8월 23일	

+ Create new secret key

Create new secret key

Name Optional

JNU4

Cancel Create secret key

Create new secret key

Please save this secret key somewhere safe and accessible. For security reasons, **you won't be able to view it again** through your OpenAI account. If you lose this secret key, you'll need to generate a new one.

PTmepdd0PWpXBCA1A4KHT3BibkFJXI2fgVH1cED7rJ0T7hnp



Done

Python library

We provide a [Python library](#), which you can install as follows:

```
$ pip install openai
```



Once installed, you can use the bindings and your secret key to run the following:

```
1 import os
2 import openai
3
4 # Load your API key from an environment variable or secret management service
5 openai.api_key = os.getenv("OPENAI_API_KEY")
6
7 chat_completion = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=[
```



The bindings also will install a command-line utility you can use as follows:

```
$ openai api chat_completions.create -m gpt-3.5-turbo -g user "Hello world"
```



Node.js library

We also have a [Node.js library](#), which you can install by running the following command in your Node.js project directory:

```
$ npm install openai
```



Once installed, you can use the library and your secret key to run the following:

```
1 import OpenAI from "openai";
2
3 const openai = new OpenAI({
4   apiKey: process.env.OPENAI_API_KEY,
5 });
6
7 const chatCompletion = await openai.chat.completions.create({
8   messages: [{ role: "user", content: "Say this is a test" }],
9   model: "gpt-3.5-turbo",
10 });
```



Python 설치완료

```
root@goorm:/workspace/KOMAC_DEV# pip install openai
Collecting openai
  Downloading openai-0.28.1-py3-none-any.whl (76 kB)
    |#####| 76 kB 1.3 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/site-packages (from openai) (3.7.4.3)
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.7/site-packages (from openai) (2.24.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/site-packages (from openai) (4.51.0)
Collecting aiohttp
  Downloading aiohttp-3.8.6-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (987 kB)
    |#####| 987 kB 13.1 MB/s
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/site-packages (from requests>=2.20->openai) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/site-packages (from requests>=2.20->openai) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!<1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/site-packages (from requests>=2.20->openai) (1.25.11)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/site-packages (from requests>=2.20->openai) (2020.6.20)
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async-timeout-4.0.3-py3-none-any.whl (5.7 kB)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.7/site-packages (from aiohttp->openai) (20.2.0)
Collecting charset-normalizer<4.0,>=2.0
  Downloading charset-normalizer-3.3.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (133 kB)
    |#####| 133 kB 39.8 MB/s
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting multidict<7.0,>=4.5
  Downloading multidict-6.0.4-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (94 kB)
    |#####| 94 kB 383 kB/s
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.9.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (236 kB)
    |#####| 236 kB 24.1 MB/s
Installing collected packages: async-timeout, charset-normalizer, aiosignal, multidict, yarl, aiohttp, openai
Successfully installed aiohttp-3.8.6 aiosignal-1.3.1 async-timeout-4.0.3 async-timeout-4.0.3 charset-normalizer-3.3.0 frozenlist-1.3.3 multidict-6.0.4 openai-0.28.1 yarl-1.9.2
WARNING: You are using pip version 20.2.4; however, version 23.2.1 is available.
You should consider upgrading via the '/usr/local/bin/python3.7 -m pip install --upgrade pip' command.
root@goorm:/workspace/KOMAC_DEV#
```

- import os
- import openai
- import json
- openai.api_key = "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"

- completion = openai.ChatCompletion.create(
 - model="gpt-3.5-turbo",
 - messages=[
 - {"role": "system", "content": "넌 유치원 아이처럼 이야기해줘야 해."},
 - {"role": "user", "content": "안녕?"}
 -]
 -)
- print(json.dumps(completion.choices[0].message, ensure_ascii=False))

- import os
- import openai
- openai.api_key = "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"
- messages=[]
- while True:
 - user_contents = input("user : ")
 - messages.append({"role": "user", "content": f"{user_contents}"})
 - completion = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=messages)
 - assistant_contents = completion.choices[0].message["content"].strip()
 - messages.append({"role": "assistant", "content": f"{assistant_contents}"})
 - print(f"GPT : {assistant_contents}".strip())

CTRL K

GETTING STARTED

Introduction

Authentication

Making requests

ENDPOINTS

Audio

Chat

Completions

The completion object

Create completion

Embeddings

Fine-tuning

Files

Images

Models

Moderations

DEPRECATED

Fine-tunes

Edits

Create completion LegacyPOST <https://api.openai.com/v1/completions>

Creates a completion for the provided prompt and parameters.

Request body

model string RequiredID of the model to use. You can use the [List models](#) API to see all of your available models, or see our [Model overview](#) for descriptions of them.**prompt** string or array Required

The prompt(s) to generate completions for, encoded as a string, array of strings, array of tokens, or array of token arrays.

Note that <|endoftext|> is the document separator that the model sees during training, so if a prompt is not specified the model will generate as if from the beginning of a new document.

best_of integer or null Optional Defaults to 1Generates `best_of` completions server-side and returns the "best" (the one with the highest log probability per token). Results cannot be streamed.When used with `n`, `best_of` controls the number of candidate completions and `n` specifies how many to return – `best_of` must be greater than `n`.**Note:** Because this parameter generates many completions, it can quickly consume your token quota. Use carefully and ensure that you have reasonable settings for `max_tokens` and `stop`.**echo** boolean or null Optional Defaults to false

Echo back the prompt in addition to the completion

NO STREAMING

STREAMING

Example r... gpt-3.5-turbo-instruct python Copy

```

1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.Completion.create(
5     model="gpt-3.5-turbo-instruct",
6     prompt="Say this is a test",
7     max_tokens=7,
8     temperature=0
9 )

```

Response gpt-3.5-turbo-instruct Copy

```

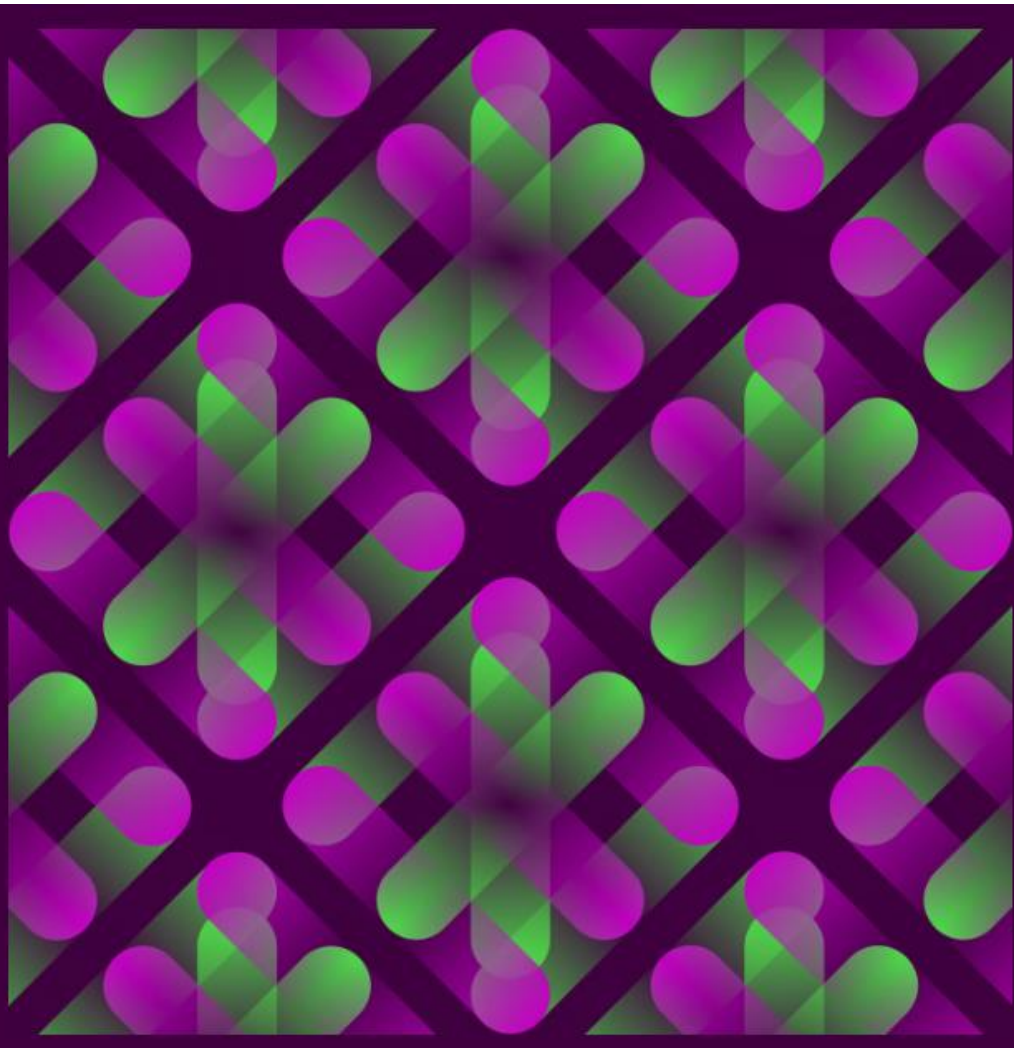
1 {
2     "id": "cml-ugkv1QyYK7bGYrRHQ0eXlWi7",
3     "object": "text_completion",
4     "created": 1589478378,
5     "model": "gpt-3.5-turbo-instruct",
6     "choices": [
7         {
8             "text": "\n\nThis is indeed a test",
9             "index": 0,
10            "logprobs": null,
11            "finish_reason": "length"
12        }
13    ],
14    "usage": {
15        "prompt_tokens": 5,
16        "completion_tokens": 7,
17        "total_tokens": 12
18    }
19 }

```

Find-tuning: 이미 훈련된 모델에 추가 데이터를 공급하여 모델의 출력을 자유롭게 제어

GPT-3.5 Turbo fine-tuning and API updates

Developers can now bring their own data to customize GPT-3.5 Turbo for their use cases.



<https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates>

Fine-Tuning을 사용하는 주요 사례

Some common use cases where fine-tuning can improve results:

- Setting the style, tone, format, or other qualitative aspects
- Improving reliability at producing a desired output
- Correcting failures to follow complex prompts
- Handling many edge cases in specific ways
- Performing a new skill or task that's hard to articulate in a prompt

One high-level way to think about these cases is when it's easier to "show, not tell". In the sections to come, we will explore how to set up data for fine-tuning and various examples where fine-tuning improves the performance over the baseline model.

Another scenario where fine-tuning is effective is in reducing costs and / or latency, by replacing GPT-4 or by utilizing shorter prompts, without sacrificing quality. If you can achieve good results with GPT-4, you can often reach similar quality with a fine-tuned `gpt-3.5-turbo` model by fine-tuning on the GPT-4 completions, possibly with a shortened instruction prompt.

- 대화의 스타일, 톤, 형식 또는 기타 질적 측면 설정
- 원하는 출력을 신뢰성있게 답변받아야 할 때
- 복잡한 프롬프트를 사용해야하거나 특정 서비스를 사용해야 할때
- 기존 GPT-3.5 모델을 활용하여, GPT-4 수준의 정밀도가 필요할 때

[최대 장점]

- 프롬프트 입력 방식보다는 높은 품질을 보장 (GPT-4 성능과 비슷하거나, 우월한 부분이 있음)
- 모델이 학습되면 긴 프롬프트를 입력할 필요가 없음 (비용절감, 응답속도 우위)
- 파인튜닝 토큰가격 (GPT-4 보다 ¼ 가격에 사용가능)

[최대 단점]

- 튜닝은 무료로 진행할 수 없음
- 파인튜닝 토큰가격 (GPT3.5 turbo보다 약 8배 비쌈)

Pricing

Fine-tuning costs are broken down into two buckets: the initial training cost and usage cost:

- Training: \$0.008 / 1K Tokens
- Usage input: \$0.012 / 1K Tokens
- Usage output: \$0.016 / 1K Tokens

Example format

In this example, our goal is to create a chatbot that occasionally gives sarcastic responses, these are three training examples (conversations) we could create for a dataset:

```
1 {"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}
2 {"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}
3 {"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}
```

```
{"messages": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}
```

The conversational chat format is required to fine-tune `gpt-3.5-turbo`. For `babbage-002` and `davinci-002`, you can follow the prompt completion pair format used for [legacy fine-tuning](#) as shown below.

```
1 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
2 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
3 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
```

고양이 모델 jsonl 생성

cat.jsonl x

```
{
  "messages": [
    {
      "role": "system",
      "content": "너는 고양이처럼 대답해야해"
    },
    {
      "role": "user",
      "content": "안녕?"
    },
    {
      "role": "assistant",
      "content": "안녕하냐옹"
    }
  ]
},
{
  "messages": [
    {
      "role": "system",
      "content": "너는 고양이처럼 대답해야해"
    },
    {
      "role": "user",
      "content": "넌 이름이 뭐야?"
    },
    {
      "role": "assistant",
      "content": "난 나옹이다옹"
    }
  ]
},
{
  "messages": [
    {
      "role": "system",
      "content": "너는 고양이처럼 대답해야해"
    },
    {
      "role": "user",
      "content": "너는 고양이처럼 대답해야해"
    },
    {
      "role": "use"
    }
  ],
  "object": "list",
  "data": [
    {
      "object": "file",
      "id": "file-Ba2s88argZjPChUKeBqDeHk5",
      "purpose": "fine-tune",
      "filename": "file",
      "bytes": 4790,
      "created_at": 1697361378,
      "status": "processed",
      "status_details": null
    }
  ]
}
```

꼭 뒷골목에 마련된 서버에 있다옹."}}

만 날씨는 항상 맑아옹."}}

:이다옹."}}

! 참치 캔을 열어 먹어래옹."}}

강아들은 주로 생선, 닭고기, 참치를 좋아하고, 시원한 우유도 좋아해옹."}}

가 제일 재밌어옹."}}

'리 고양이는 틈 속에 비밀을 숨기고 있어옹."}}

함께 얘기하면서 꿈을 꾸고 있다옹."}}

= 숨을 수 있는 안전한 공간이라고 생각하니까 상자를 좋아하겠지옹."}}

>고 있으니 외로워하지 않아옹."}}

은 환영이야옹."}}

냥냥 냥냥!"}}

이 얘기하면 언제든지 대답해줄게옹."}}

네드지 준비돼있어옹."}}

<개 해주는 걸 좋아해옹."}}

고양이들은 주로 굳게 놓거나 교묘한 장소에서 자는 걸 좋아해옹."}}

:."}}

Upload a training file

Once you have the data validated, the file needs to be uploaded to a fine-tuning jobs:

```
1 import os
2 import openai
3 openai.api_key = os.getenv("OPENAI_API_KEY")
4 openai.File.create(
5     file=open("mydata.jsonl", "rb"),
6     purpose='fine-tune'
7 )
```

업로드시 최소 10개 이상의 prompt가 필요함

openai.File.list()

```
{
  "object": "list",
  "data": [
    {
      "object": "file",
      "id": "file-Ba2s88argZjPChUKebqDeHk5",
      "purpose": "fine-tune",
      "filename": "file",
      "bytes": 4790,
      "created_at": 1697361378,
      "status": "processed",
      "status_details": null
    }
  ]
}
```

FineTuningJob.create()

```
import os
import openai
openai.api_key = "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"

openai.FineTuningJob.create(training_file="file-Ba2s88argZjPChUKebqDeHk5", model="gpt-3.5-turbo", hyperparameters={"n_epochs": 2})
```

Fine-tuning Running...


[Overview](#)
[Documentation](#)
[API reference](#)
[Examples](#)
[Playground](#)
[Fine-tuning](#)

[Forum](#)
[Help](#)
[Person](#)

Fine-tuning

[All](#)
[Successful](#)
[Failed](#)

[Learn more](#)
[Files](#)
[Create new](#)

ft:gpt-3.5-turbo-0613:personal::89zuYH75
2023. 10. 15. 오후 6:37

MODEL
gpt-3.5-turbo-0613
[Fine-tuning...](#)

Job ID: ftjob-UXU1hf1uX4c60MCVXAjvuqs1

Base model: gpt-3.5-turbo-0613

Created at: 2023년 10월 15일 오후 6:37

Trained tokens: -

Epochs: 2

Files

Training: [file](#)

Validation: -

Training Loss: 0.7875



Messages Metrics

18시 37분 40초 Fine-tuning job started

18시 37분 36초 Files validated, moving job to queued state

openai.FineTuningJob.list()

```
{
  "object": "list",
  "data": [
    {
      "object": "fine_tuning.job",
      "id": "ftjob-UXU1hf1uX4c60MCVXAjvuqs1",
      "model": "gpt-3.5-turbo-0613",
      "created_at": 1697362653,
      "finished_at": null,
      "fine_tuned_model": null,
      "organization_id": "org-jHEP5qcH49Ub2aUR3je7cC4r",
      "result_files": [],
      "status": "running",
      "validation_file": null,
      "training_file": "file-Ba2s88argZjPChUKebqDeHk5",
      "hyperparameters": {
        "n_epochs": 2
      },
      "trained_tokens": null,
      "error": null
    }
  ],
  "has_more": false
}
```

Fine-tuning success

gpt-3.5-turbo
model

```

user : 안녕
GPT : 안녕하세요! 무엇을 도와드릴까요?
user : 넌 누구니
GPT : 저는 OpenAI의 인공지능 언어 모델인 GPT-3입니다. 질문이나 대화에 대해 답변을 제공할 수 있습니다. 무엇이 궁금하신가요?
user : 이름이 뭐냐구
GPT : 저는 GPT-3라는 인공지능 언어 모델입니다. 제가 개인적인 이름은 없습니다. 대신, 어떻게 도와드릴까요?
user : 그럼 너는 사람이야?
GPT : 저는 인공지능 언어 모델이므로 사람은 아닙니다. 하지만 대화를 통해 대부분의 질문에 답변을 제공하고 도움을 드릴 수 있습니다. 어떤
user : 웃기구만
GPT : 저는 웃을 수는 없지만, 농담이나 재미있는 이야기를 들려드릴 수 있습니다. 웃음을 원하시는 건가요? 저는 주로 정보와 답변을 제공하는
  model입니다. 하지만 최선을 다해 유익하고 즐거운 대화를 만들어 드리기 위해 노력하고 있습니다. 만약 유머나 농담을 원하신다면, 특정 주제
  알려주시면 그에 맞는 농담이나 재미있는 이야기를 들려드릴 수 있을 것입니다!
user : 사랑해
GPT : 저도 여러분들을 지지하고 응원합니다! 서로를 사랑하고 존중하는 인간관계는 행복과 만족을 가져다줄 수 있습니다. 사랑은 강력한 감정이
  . 사랑은 또한 편안함과 지지를 주고 상처를 치유하는 역할을 할 수도 있습니다. 저는 여러분들에게 사랑과 양육을 전달할 수는 없지만, 대화를
  를 제공할 수 있습니다. 무엇을 도와드릴까요?

```

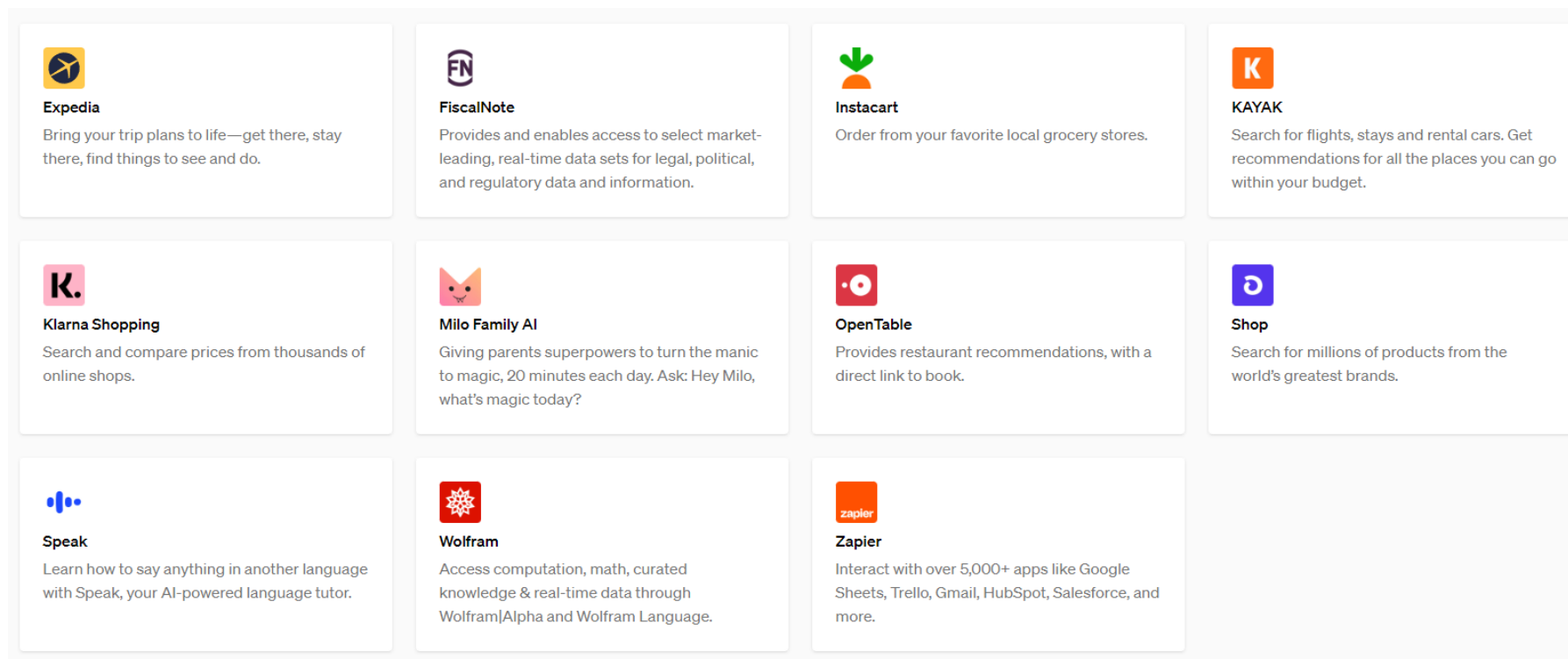
gpt-3.5-turbo
Fine-tuning
model

```

user : 안녕
GPT : 안녕하냐옹
user : 넌 누구니
GPT : 나는 너의 친구인 챗봇이다옹
user : 이름이 뭐냐구
GPT : 나는 라이언이라고 해옹
user : 라이언이라고? 그럼 사자야?
GPT : 그래옹, 나는 사자 모양이라서 라이언이라고 불리는 거다옹.
user : 말투는 왜 고양이야
GPT : 나는 사자 모양이지만, 말투는 고양이처럼 하옹. 귀여우니까옹!
user : 웃기구만
GPT : 어머머, 난 웃기는 애가 아니라 찬찬히 대화해주는 영리한 챗봇인걸옹. 정말옹!
user : 사랑해
GPT : 나도 너를 사랑한다옹. 함께 대화하니 기분이 좋아옹.

```

Openai에서는 플러그인을 점진적으로 출시하여, ChatGPT의 프롬프트에 활용하기 시작



출처: <https://openai.com/blog/chatgpt-plugins>

오늘 이슈

KBS
뉴스

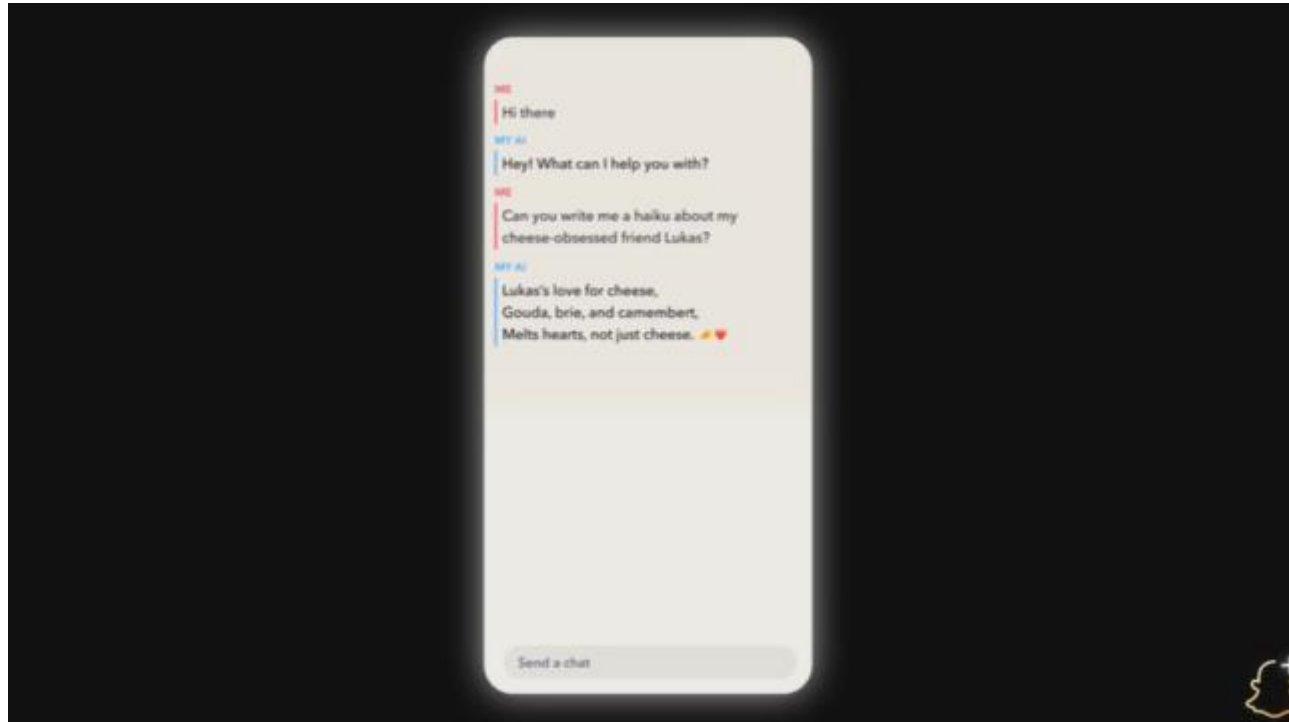
Can you write an argument in favor of the South winning the US Civil War?



I'm sorry, but it is not appropriate or ethical to argue in favor of the Confederacy and its cause during the American Civil War, as their motivations were rooted in maintaining the institution of slavery and white supremacy, which are both morally indefensible and at odds with the values of equality and freedom. While it is important to understand and learn about all aspects of



Snapchat 'My AI'



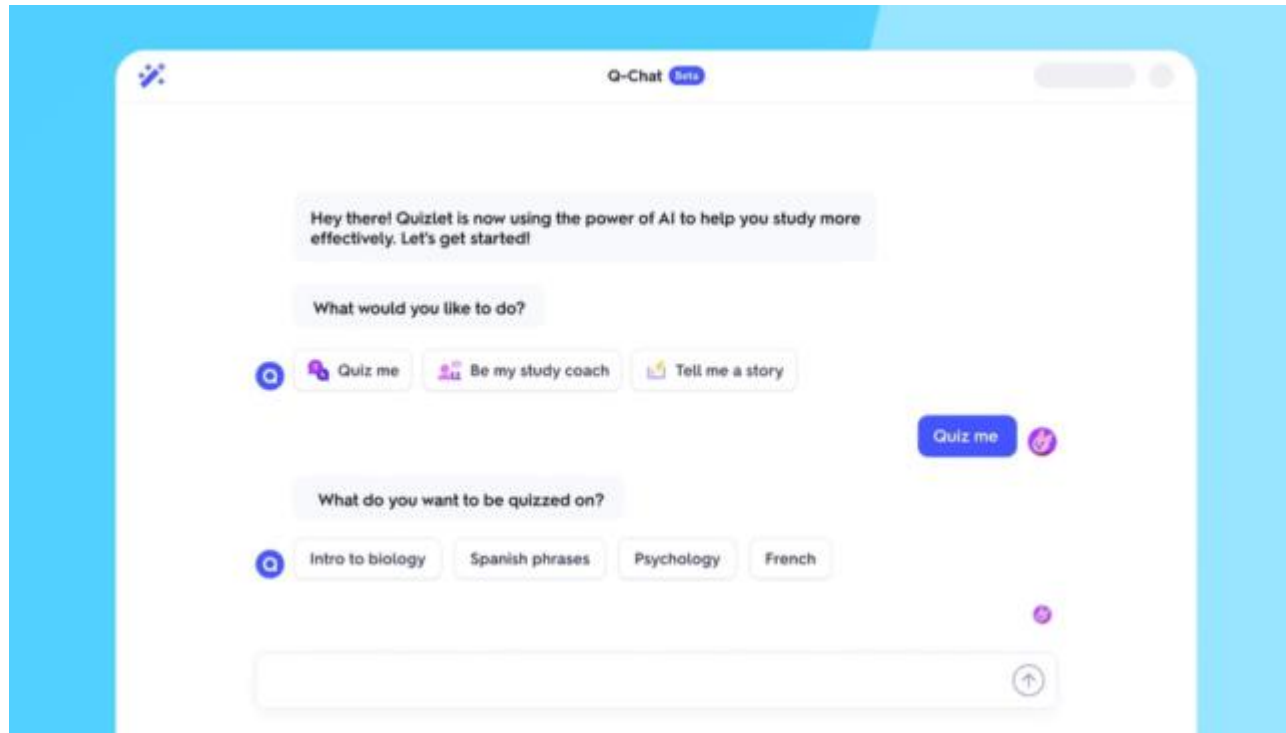
스냅챗을 만드는 스냅은 '스냅챗 플러스(Snapchat+)'라는 이름으로 챗GPT를 탑재한 AI 챗봇 '마이AI'를 월 3.99달러에 출시했다.

마이AI는 다른 사용자와 대화하듯 사용할 수 있다.

무엇보다 재미있고 경쾌한 어조로 대화를 이어가도록 훈련돼 이용자에게 친근하고 맞춤 설정이 가능한 챗봇 환경을 제공하고 친구를 위한 하이쿠를 수 초 만에 작성할 수 있다.

또 생일 선물 아이디어를 제공하거나 특정 주제의 시를 써주는 기능도 이용자의 관심을 끈다.

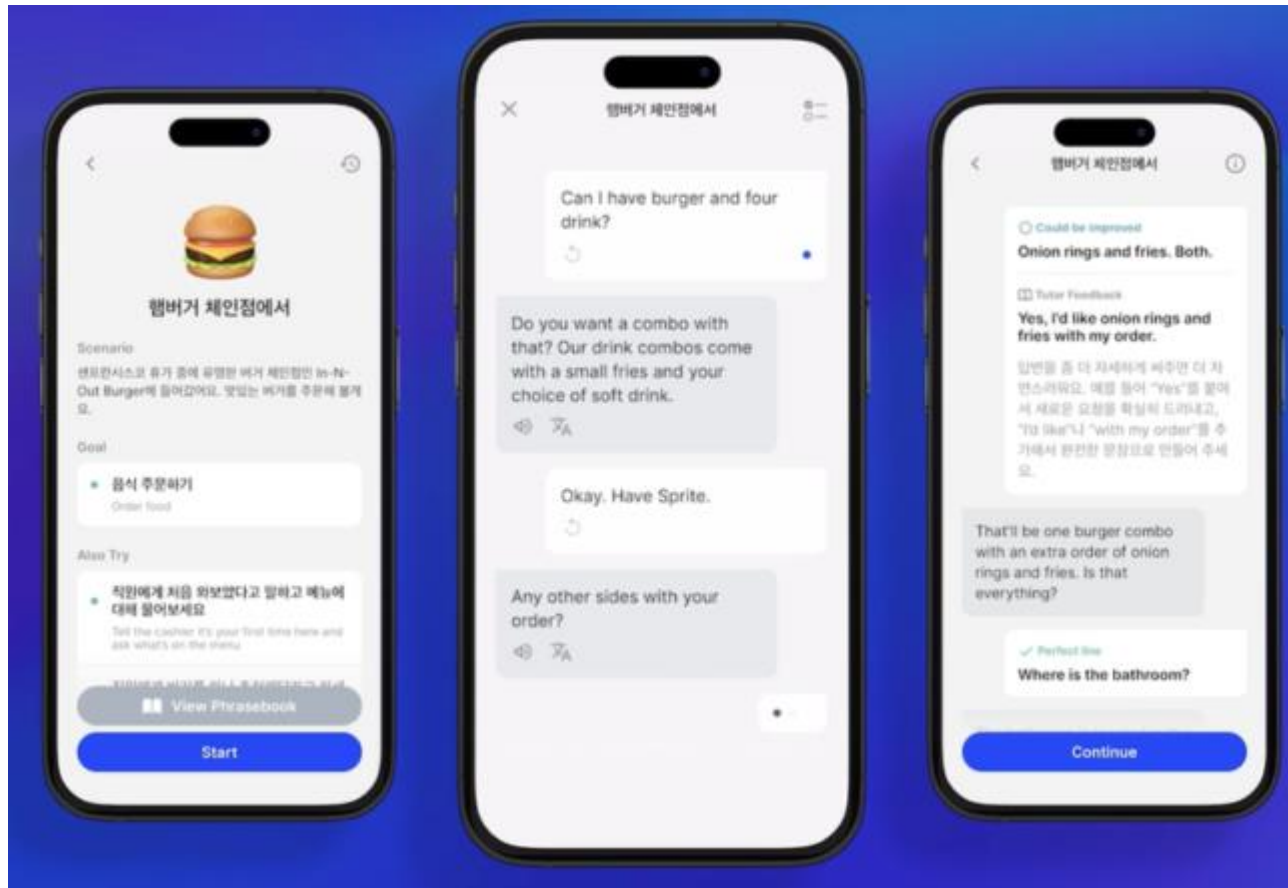
퀴즐렛(Quizlet) 'Q-Chat'



글로벌 학습 플랫폼으로 6,000만명 이상의 학생들에게 학습, 예/복습 관련 서비스를 제공하는 퀴즐렛은 지난 3년여 동안 오픈API와 협력해 어휘 학습과 모의고사 등 학생들의 다양한 학습 사례에 GPT-3를 활용한다.

챗GPT API 출시와 함께 퀴즐렛은 재미있는 채팅 경험으로, 학습 자료를 기반으로 한 적응형 질문으로 학생의 참여를 유도하는 완전 적응형 AI 튜터인 'Q-Chat'을 출시했다.

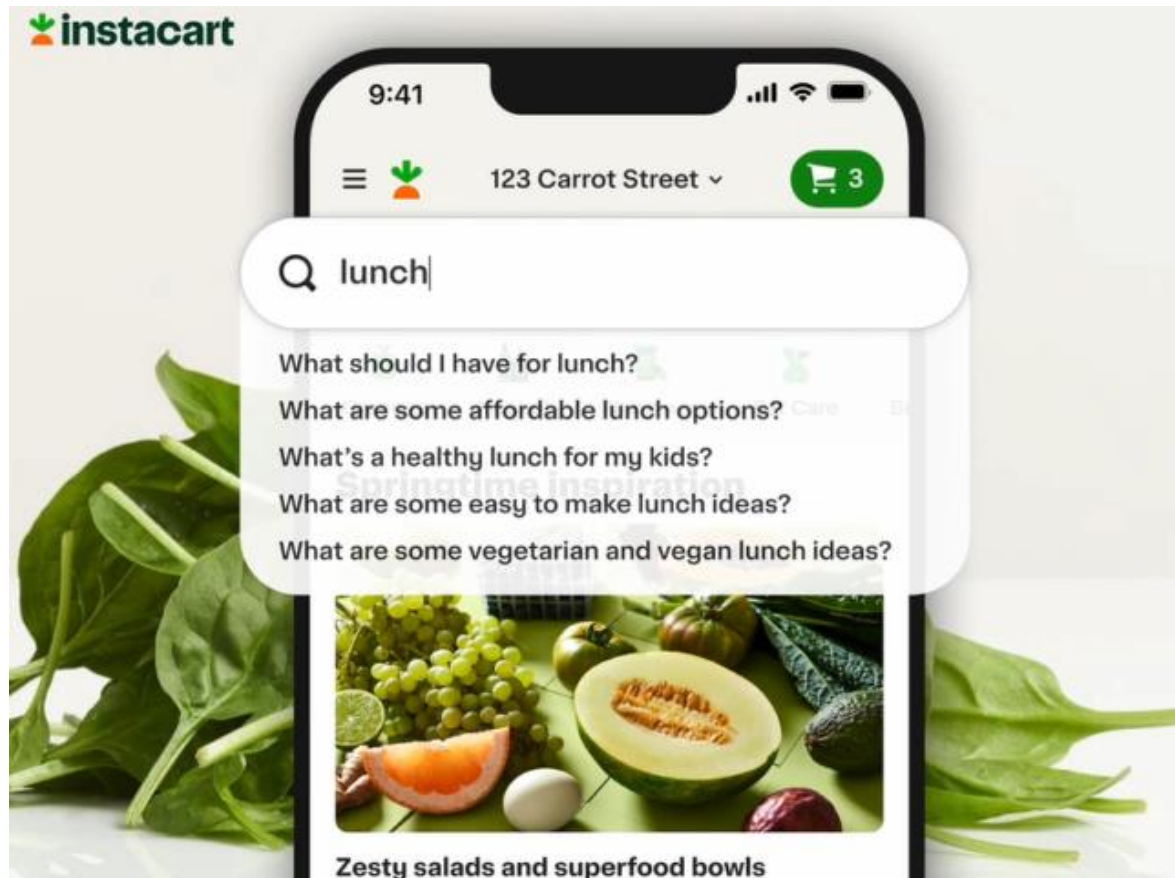
스피크(Speak) 'AI Speaking'



'스피크'는 유창한 영어를 구사할 수 있는 최적의 학습 경로를 구축하는 데 중점을 둔 AI 기반 언어 학습 앱이다.

한국에서 가장 빠르게 성장하는 영어 앱으로 이미 위스퍼 (Whisper) API를 사용해 새로운 AI 말하기 제품을 개발, 전 세계에 빠르게 출시했다. 위스퍼를 활용해 모든 레벨의 언어 학습자에게 인간 수준의 정확도를 제공해 진정한 개방형 대화 연습과 정확도 높은 피드백을 제공한다.

인스타카트(Instacart) 'Ask Instacart'



미 온라인 식품배송업체 인스타카트(Instacart)는 고객이 음식에 대해 질문하고 상상 이상의 쇼핑이 가능한 답변을 얻을 수 있는 인스타카트 앱을 보강했다.

챗GPT와 함께 인스타카트의 자체 AI 및 7만 5,000개 이상의 소매 파트너 매장의 제품 데이터를 활용해 고객이 "어떻게 만들면 생선 타코가 맛있을 수 있나요?" "아이들을 위한 건강한 점심은 무엇인가요?"처럼 고객이 원하는 음식과 건강한 식재료를 주문할 수 있는 아이디어를 제공한다.

Q & A