

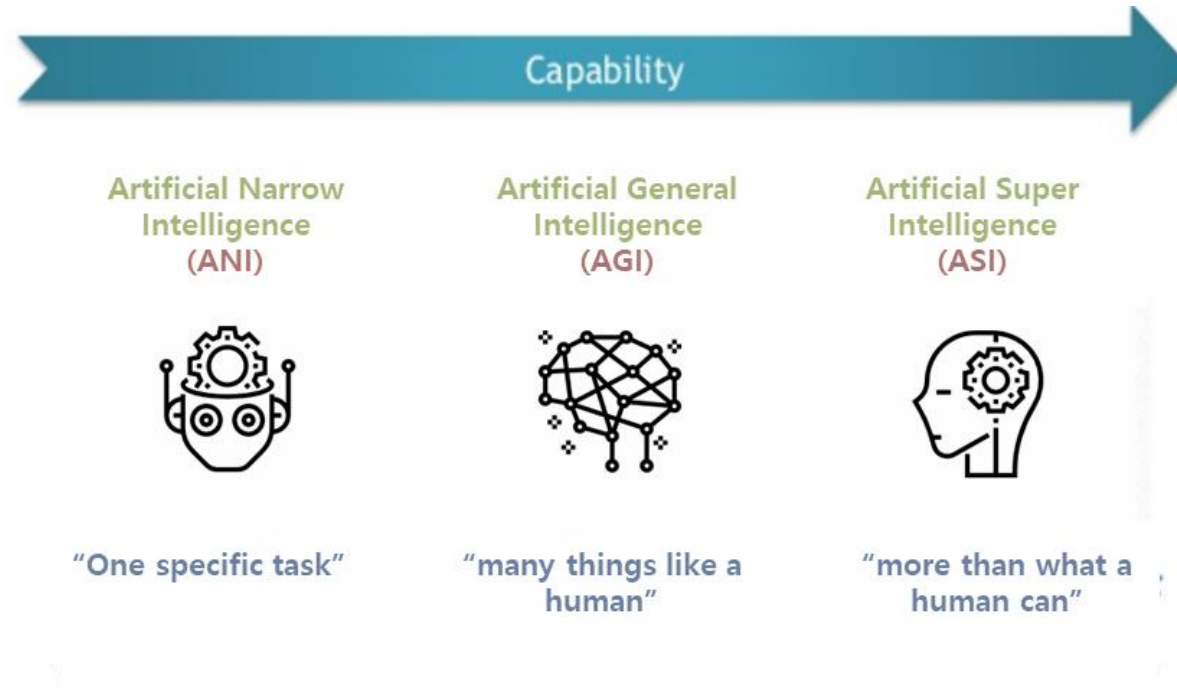


# 챗GPT 소개

2025. 8.

# I. LLM 이해하기

- AI 개요



- '약인공지능(ANI)', '범용인공지능(AGI)', '초인공지능(ASI)'
- ANI: 주어진 데이터 내에서 주어진 문제를 해결하는 AI
- AGI로의 진입 가속화 중

# I. LLM 이해하기

- Scaling Laws (Open AI, 2020)

- 컴퓨팅 리소스, 데이터, 모형 크기를 늘릴 수록 성능 개선
- 새로운 능력이 생겨남 (Emergent Abilities)

Scaling Laws for Neural Language Models			
Jared Kaplan *		Sam McCandlish*	
Johns Hopkins University, OpenAI		OpenAI	
jaredk@jhu.edu		sam@openai.com	
Tom Henighan	Tom B. Brown	Benjamin Chess	Rewon Child
OpenAI	OpenAI	OpenAI	OpenAI
henighan@openai.com	tom@openai.com	bchess@openai.com	rewon@openai.com
Scott Gray	Alec Radford	Jeffrey Wu	Dario Amodei
OpenAI	OpenAI	OpenAI	OpenAI
scott@openai.com	alec@openai.com	jeffwu@openai.com	damodei@openai.com

## Abstract

We study empirical scaling laws for language model performance on the cross-entropy loss. The loss scales as a power-law with model size, dataset size, and the amount of compute used for training, with some trends spanning more than seven orders of magnitude. Other architectural details such as network width or depth have minimal effects within a wide range. Simple equations govern the dependence of overfitting on model/dataset size and the dependence of training speed on model size. These relationships allow us to determine the optimal allocation of a fixed compute budget. Larger models are significantly more sample-efficient, such that optimally compute-efficient training involves training very large models on a relatively modest amount of data and stopping significantly before convergence.

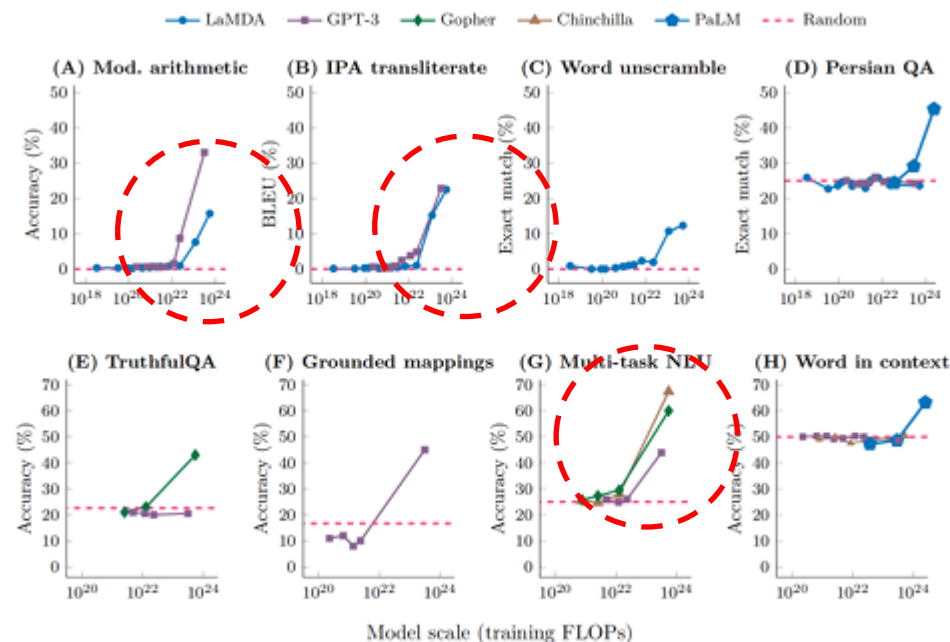
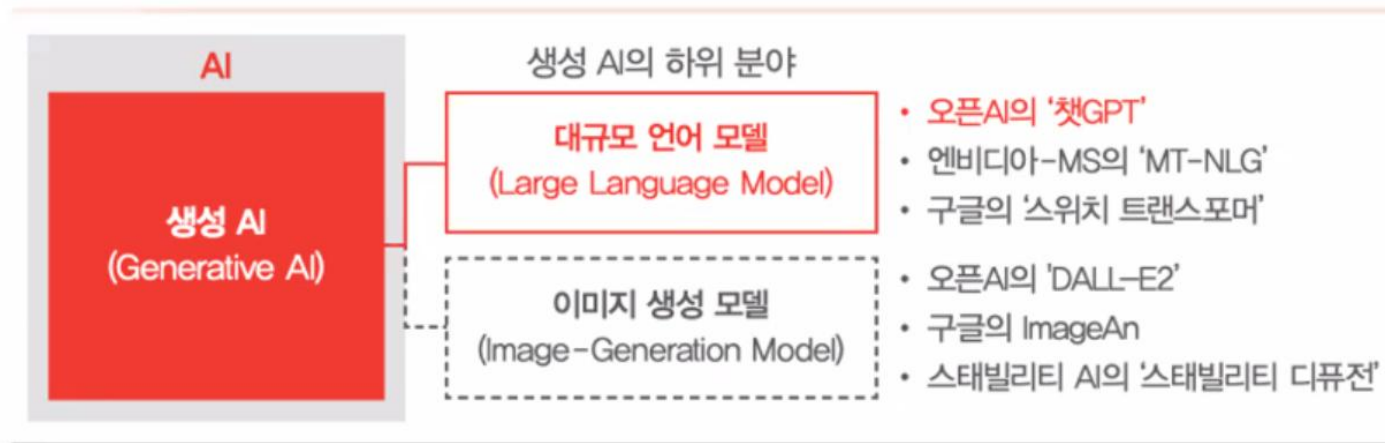


Figure 1: Emergent abilities of large language models. Model families display *sharp* and *unpredictable* increases in performance at specific tasks as scale increases. Source: Fig. 2 from [33].

# I. LLM 이해하기

## • 생성형 AI

- 대형 언어모델이나 이미지 생성모델 활용, User가 원하는 것을 생성하는 AI 분야
- LLM: 테라바이트 단위의 대용량 코퍼스로 모형을 학습, 문장을 생성하는 모형
- IGM(Image Generation Model): 텍스트를 입력하면 그에 대한 이미지를 생성, DALL-E2, 미드저니 등

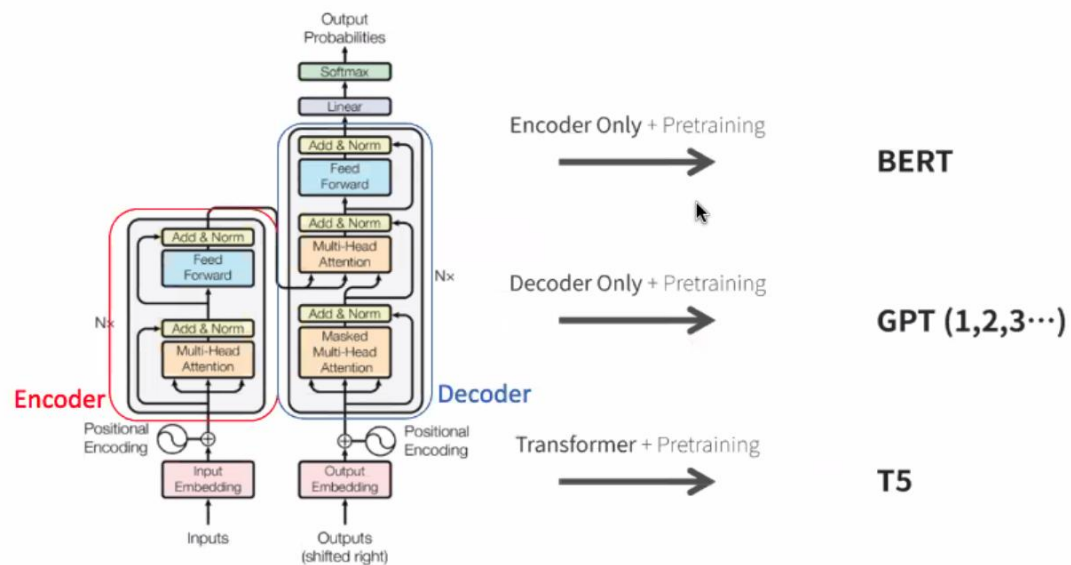


※ 자료: 삼일PwC경영연구원

# I. LLM 이해하기

- 간단하게 이해하는 LLM

- 언어모형: 주어진 문장을 구성하는 단어에 대한 확률을 모델링
- 주어진 문장/단어의 다음 단어를 예측, 문장 내 Masking한 단어를 예측
- 대용량 텍스트를 학습한 모형이 좋은 Representation(Embedding)을 보유



# I. LLM 이해하기

## • 간단하게 이해하는 LLM

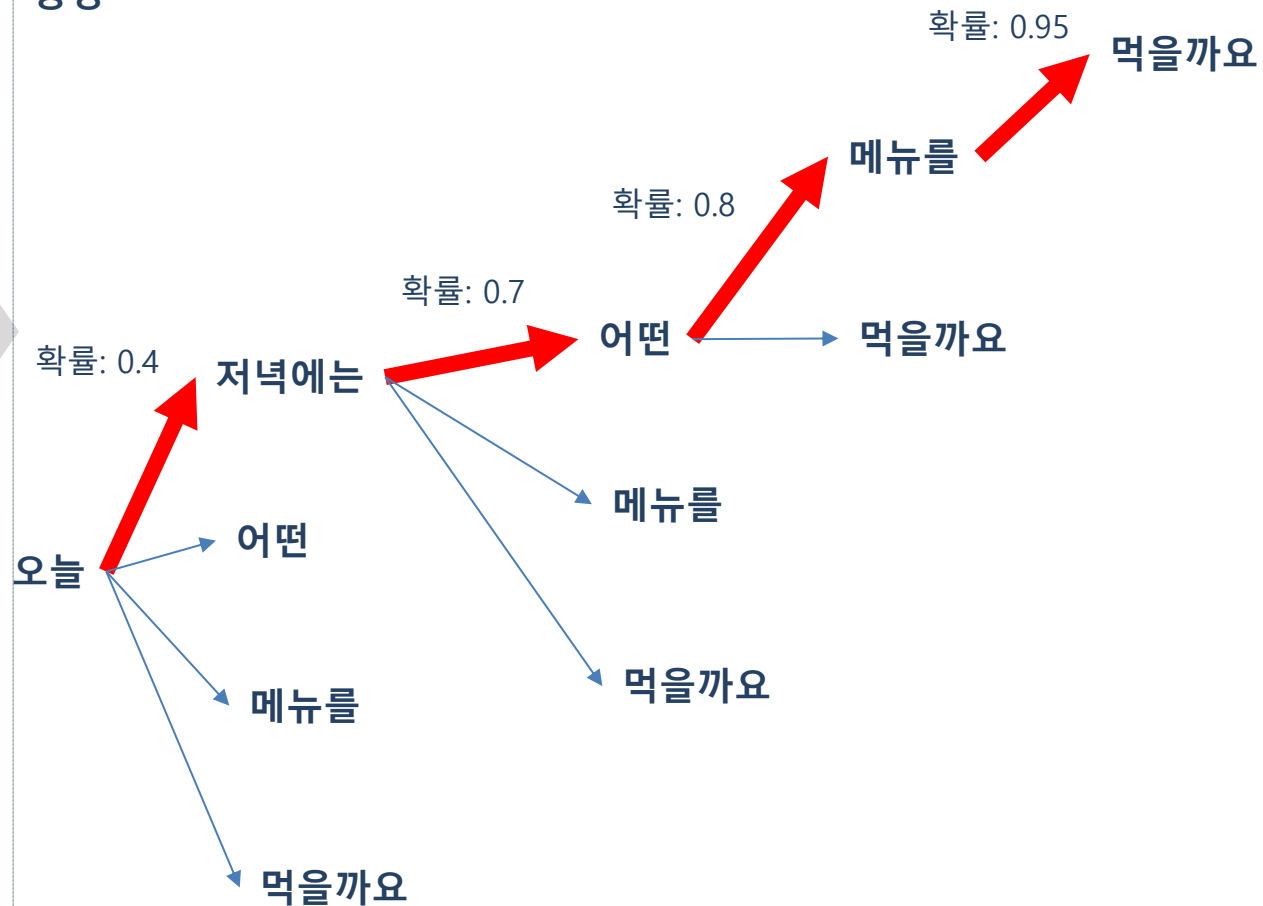
"오늘 저녁에는 어떤 메뉴를 먹을까요?"



오늘 : 0.5, 1.2, 0.3, ..., 0.9  
저녁에는 : 1.1, 0.2, 1.3, ..., 0.7  
어떤 : 1.5, 0.2, 0.9, ..., 0.5  
메뉴를 : 1.1, 1.7, 1.3, ..., 1.5  
먹을까요 : 1.12, 1.3, 1.7, ..., 1.1

Representation (Embedding)

생성



# I. LLM 이해하기

- 다양한
- LLM

The screenshot shows the Hugging Face homepage. On the left, the 'Tasks' filter is active, displaying a list of tasks categorized by type: Multimodal, Computer Vision, Natural Language Processing, and Audio. On the right, a grid of model cards is displayed, each showing the model name, its primary task, and some statistics like update time and likes.

**Hugging Face** Search models, datasets, users...

Models Datasets Spaces Posts Docs Solutions Pricing Log In Sign Up

**Tasks** Libraries Datasets Languages Licenses Other

Filter Tasks by name

**Multimodal**

- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering

**Computer Vision**

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction

**Natural Language Processing**

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

**Audio**

- Text-to-Speech
- Text-to-Audio
- Automatic Speech Recognition
- Audio-to-Audio

**Models** 606,274 Filter by name

new Full-text search Sort: Trending

- mistral-community/Mixtral-8x22B-v0.1**  
Text Generation • Updated about 10 hours ago • 33.9k • 595
- openbmb/MiniCPM-V-2**  
Visual Question Answering • Updated 2 days ago • 2.45k • 287
- HuggingFaceH4/zephyr-orpo-141b-A35b-v0.1**  
Text Generation • Updated 5 days ago • 3.91k • 194
- parler-tts/parler\_tts\_mini\_v0.1**  
Text-to-Speech • Updated 7 days ago • 14.4k • 190
- HuggingFaceM4/idefics2-8b**  
Image-Text-to-Text • Updated about 5 hours ago • 5.23k • 173
- Vezora/Mistral-22B-v0.1**  
Text Generation • Updated 6 days ago • 519 • 139
- taide/TAIDE-LX-7B-Chat**  
Text Generation • Updated 1 day ago • 1.2k • 83
- mistralai/Mixtral-8x22B-v0.1**  
Text Generation • Updated about 11 hours ago • 5 • 69
- google/codegemma-7b-it**  
Text Generation • Updated 1 day ago • 5.29k • 131
- meta-llama/Llama-2-7b-chat-hf**  
Text Generation • Updated about 16 hours ago • 1.47M • 3.38k
- CohereForAI/c4ai-command-r-plus**  
Text Generation • Updated 8 days ago • 153k • 1.15k
- mistralai/Mixtral-8x22B-Instruct-v0.1**  
Text Generation • Updated about 8 hours ago • 124 • 202
- NexaAI/Octopus-v2**  
Text Generation • Updated 5 days ago • 10.7k • 728
- alpindale/WizardLM-2-8x22B**  
Text Generation • Updated 1 day ago • 2.04k • 175
- mistralai/Mistral-7B-Instruct-v0.2**  
Text Generation • Updated 25 days ago • 2.98M • 1.93k
- Vezora/Mistral-22B-v0.2**  
Text Generation • Updated 3 days ago • 995 • 91
- amazingvince/Not-WizardLM-2-7B**  
Text Generation • Updated 1 day ago • 631 • 72
- mistralai/Mixtral-8x7B-Instruct-v0.1**  
Text Generation • Updated Feb 29 • 846k • 3.67k
- openai/whisper-large-v3**  
Automatic Speech Recognition • Updated Feb 8 • 1.82M • 2.27k
- runwayml/stable-diffusion-v1-5**  
Text-to-Image • Updated Aug 24, 2023 • 4.07M • 10.7k



## • 챗GPT

- GPT3는 GPT2에 비해 10배 이상의 데이터(570GB), 100배 이상 모형 크기(175B)
- GPT4는 1700B의 파라미터(estimated)
- Few-shot 학습을 통한 In-Context Learning 향상
  - 프롬프트의 내용만으로 Task 수행
  - 프롬프트 내 맥락을 이해하고 답변

 You

나에게 송년회 건배사를 3개만 알려줘, 건배사가 감동적이었으면 좋겠어

 ChatGPT

물론, 아래는 감동적인 송년회 건배사의 예시 중 3가지입니다. 이 중 하나를 선택하여 송년회에서 사용하실 수 있습니다.

1. 기회와 변화를 의미있게 채우며:

"우리는 또 하나의 해를 마무리하고 새로운 시작을 앞두고 있습니다. 이번 해도 여러 어려움과 도전에 직면했지만, 그런 순간들에서 우리는 더 강해지고 성장했습니다. 이제 하는 일과 목표는 새로운 해를 기대하게 만듭니다. 모두가 함께 협력하고 서로를 지원하면, 무엇이든 이룰 수 있습니다. 새해에는 우리 모두가 행복하고 건강하게 지내길 기원하며, 함께 하는 모든 순간을 소중히 여기고 감사하게 생각합니다. 건배!"

2. 우정과 고마움 표현:

"이 자리에 모여 함께 시간을 보낸 모든 분들에게 감사의 말씀을 전하고 싶습니다. 우리는 서로를 향한 그 무한한 우정을 실제로 느낄 수 있었던 해였습니다. 다음 해에도 함께 소중한 순간을 나누고, 어려움을 함께 극복하며, 더 많은 기억을 만들어 나갈 것입니다. 모두에게 행복하고 풍요로운 새해를 기원합니다. 건배!"

3. 빛나는 미래를 기대하며:

"저희는 이 송년회에서 과거를 되돌아보고, 현재를 즐기며 미래를 기대합니다. 우리는 계속해서 발전하고 성장하며, 새로운 도약을 위해 노력할 것입니다. 이곳에 모여주신 모든 분들에게 감사의 인사를 전하며, 다음 해에는 더 큰 성취와 행복을 경험하길 바랍니다. 새해에는 우리의 꿈을 이루고 빛나는 미래를 함께 만들어 나가기를 기대합니다. 건배!"



- 효율적인 LLM을 위한 트렌드: sLLM

- Small Large Language Model
- LLM 대비 파라미터 수를 수십억~수백억 개 수준으로 유지
- Meta의 LLaMA부터 관심
- 효율적 / On device 활용 / 높은 유연성 / Fine tuning을 적용
- 예:
  - 알파카7B: 스탠포드대학교, LLaMA의 가장 작은 버전 활용한 소형 언어모형, GPT-3.5에 버금가는 성능이면서도 100만원 내외의 비용과 8대 PC로 3시간 훈련
  - Dolly: 데이터브릭스, 60억개 파라미터, 1대의 서버로 3시간 훈련

## II. 업무 활용 사례

- 챗GPT리터러시: 이미지에서 텍스트 추출, 데이터 분석, 코딩 등



Rowan Cheung  
@rowancheung

The ChatGPT Mac app is the ultimate screenshot-to-code tool.

Screenshot anything, paste it in the ChatGPT shortcut, and just tell GPT-4o to code it for you.

Here's me taking a snapshot of Snake Game and getting fully working code in 90 seconds. Video is on 3x speed.

[Traducir post](#)

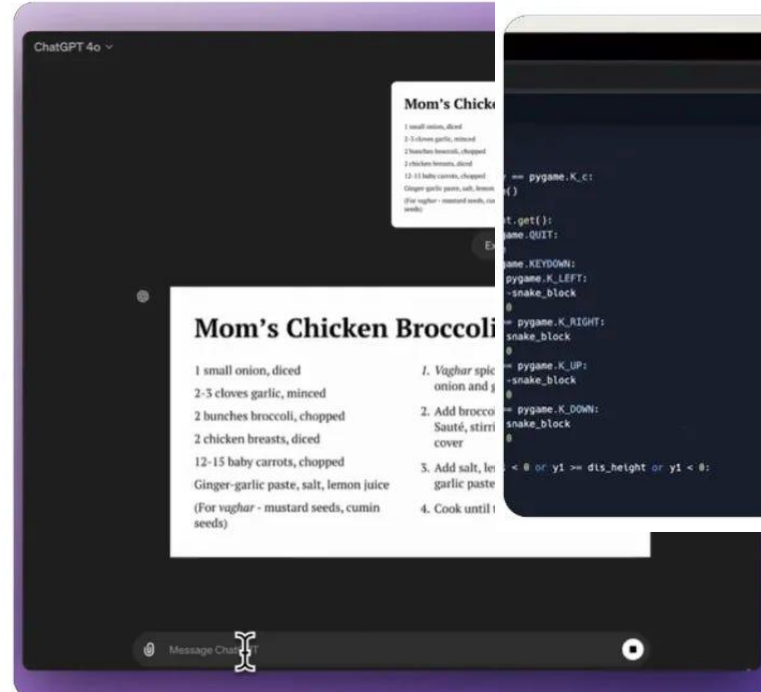


Robin Delta  
@heyrobinai

3. Extract text from images

Prompt: Extract this recipe to a JSON file

[Traducir post](#)

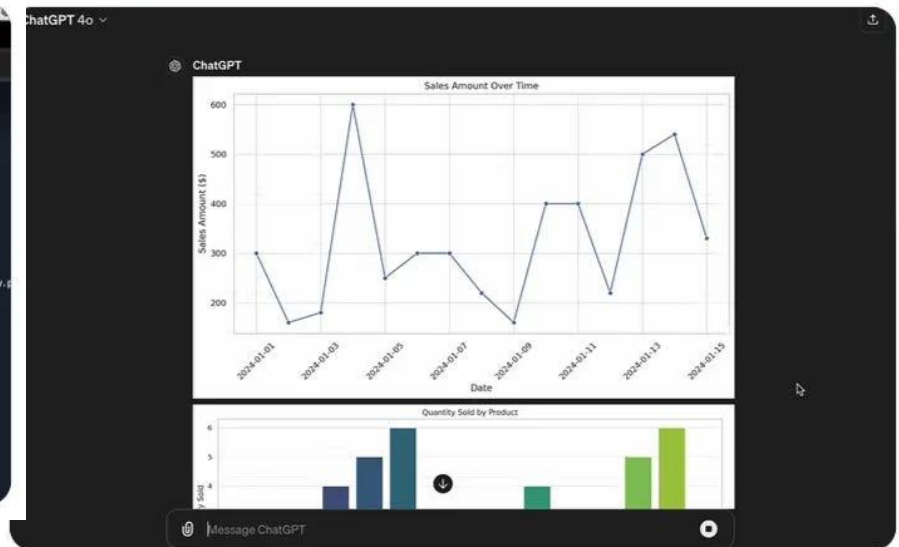


Zain Kahn  
@heykahn

/ild. GPT-4o on ChatGPT can generate full blown charts and statistical analysis from spreadsheets with a single prompt in less than 30 seconds.

his stuff use to take ages in Excel.

[aducir post](#)



<https://x.com/heykahn/status/1790347606750532010>

Industrial Data Science Lab

## II. 업무 활용 사례

- 프롬프트 구조화

- XML 태그 스타일의 영역 지정

- XML태그?

- XML은 데이터의 구조를 기술하는 Mark up 언어
      - 데이터의 각 항목에 대해 의미하는 태그로 마크업
      - 예: <키워드> 챗, 생성AI </키워드>

## II. 업무 활용 사례

- 프롬프트 구조화

- 활용 예시 프롬프트

대화 예시와 결과 예시를 참고해서, 다음 <<<대화>>>에 나오는 각 문장의 감성을 '긍정' 이나 '부정' 으로 분류하시오.

<범주>

긍정

부정

</범주>

<대화예시>

[행인1]: 오늘 날씨가 좋네요

[행인2]: 좋은 뭐가 좋아요

</대화예시>

<결과예시>

긍정

부정

</결과예시>

<대화>

[행인1] 비가 오니 짜증납니다.

[행인2] 비오는 날 분위기 좋아요

</대화>

## II. 업무 활용 사례

- Vibe 코딩

- Vibe 코딩(Vibe Coding)이란?

- AI 중심 소프트웨어 개발 방식
    - Andrej Karpathy(OpenAI 공동 창업자)가 제안
    - 기존의 명시적 코딩 → 자연어 지시 + AI 응답



- Vibe 코딩의 핵심 철학

- “코딩이라기보단, 보고 말하고 실행하고 복사하여 붙여넣는 일련의 행위”
    - 기술/언어 제약 없이, 아이디어 → 결과물로 빠르게 전환

## II. 업무 활용 사례

- Vibe 코딩
  - 직관적 개발
    - 자연어로 명령 → 자동 코드 생성
      - "캔버스에서 장애물을 피하는 게임 만들어줘" → 간단한 게임 완성
      - "위 내용을 메모리에 기억해 "
  - 빠른 프로토타입 제작
    - 비전문가도 AI로 직접 프로토타입 제작 가능
    - 개발 속도 향상, 기술 장벽 완화

## II. 업무 활용 사례

- Vibe 코딩
  - 초기 작동 ≠ 장기적 신뢰성
    - AI가 생성한 코드는 복잡한 상황에 취약전문 개발자의 코드 검토 및 수정이 필요
  - 아이디어 우선의 함정
    - 기술적 검토 없이 낙관적인 개발 접근 위험
    - 실현 가능성보다 컨셉에 집중될 수 있음
  - 개발자의 영역
    - Vibe 코딩만으로는 모든 문제 해결 불가
    - 복잡한 알고리즘, 성능 최적화 등은→ 여전히 숙련된 개발자의 영역



## II. 업무 활용 사례

- Vibe 코딩

- 코드 인터프리터와 캔버스

- 자연어 명령을 코드로 변환하여 실행: "명령 기반 코드 생성"을 실현

- 코드 명령어 입력: "데이터를 분석하여 매출 증가율을 시각화하는 파이썬 코드를 작성해줘."

- 코드 생성: 챗GPT는 명령어를 해석하여 파이썬 코드를 자동으로 생성

```
python

import pandas as pd
import matplotlib.pyplot as plt

# 가상 데이터 생성
data = {'월': ['1월', '2월', '3월', '4월'], '매출': [100, 150, 200, 300]}
df = pd.DataFrame(data)

# 시각화
plt.plot(df['월'], df['매출'])
plt.title('매출 증가율')
plt.show()
```

- 코드 실행 및 결과 출력

- » 코드 인터프리터는 자동으로 코드를 실행하여 시각화를 제공

- » 사용자는 코드 수정이나 추가 명령을 통해 즉각적으로 결과를 보정할 수 있음

## II. 업무 활용 사례

- Vibe 코딩

- 캔버스

- 실시간 코드 수정과 협업의 기능
    - 사용자가 자연어로 작성한 텍스트나 코드를 시각적으로 구성하고 수정할 수 있도록 돕는 인터랙티브 작업 공간
    - 기존의 단순한 대화형 인터페이스를 넘어, 문서 작성, 코드 편집, 시각적 구성 등을 하나의 공간에서 실시간으로 수행할 수 있음
    - 콘텐츠와 코드, 시각적 요소를 통합적으로 관리할 수 있는 환경을 제공함으로써, 개발자뿐 아니라 비전문가도 복잡한 작업을 직관적으로 처리할 수 있음

## II. 업무 활용 사례

- Vibe 코딩

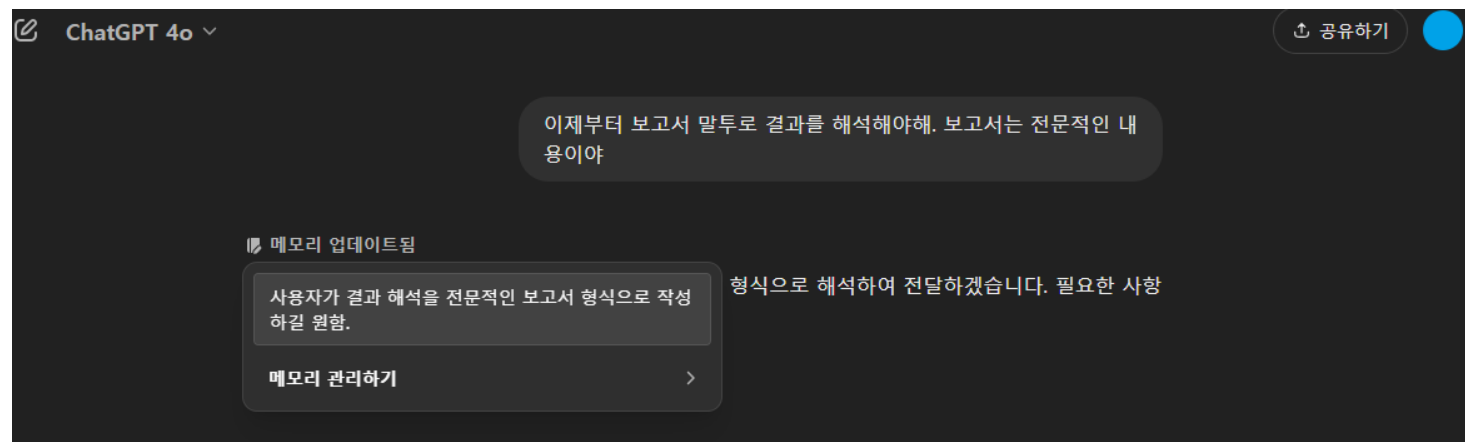
- 캔버스

- 실시간 문서 작성과 수정 가능
    - 코드 작성 및 편집 지원: 코드 블록을 구분하고 각 코드에 대한 설명을 함께 구성할 수 있는 인터페이스를 제공. 파이썬, 자바스크립트, HTML 등 다양한 프로그래밍 언어를 지원
      - 문법 오류 감지 및 자동 완성 기능도 포함되어 있어 사용자의 생산성 제고
      - 생성된 코드는 캔버스 내에서 직접 실행할 수 있으며, 실행 결과 역시 시각적으로 확인할 수 있음.
    - 데이터 시각화 기능 지원: 데이터 분석 결과를 보여주는 그래프나 도표, 데이터를 수정할 경우 시각화 결과도 자동으로 업데이트
    - 실시간 협업 기능:
      - 캔버스는 여러 사용자가 하나의 작업 공간에서 협업할 수 있도록 지원. 예를 들어 팀 프로젝트를 진행할 때, 한 명은 문서를 작성하고 다른 팀원은 코드 블록을 작성하거나 시각화 자료를 삽입하는 분업 가능

## II. 업무 활용 사례

- 메모리

- 사용자와의 이전 대화를 기억하고,이후 대화에 지속적으로 반영하는 기능
- 반복 설명 없이, 개인화된 대화 가능
- 메모리 관리 방법좌측 메뉴 > 프로필 > "메모리" 메뉴 켜기 / 끄기
- 개별 정보 삭제 가능 + 각 대화별로도 비활성화 가능
- 민감한 정보(비밀번호, 개인정보 등) 는 저장하지 않음
- 챗GPT가 기억한 내용은 수동으로 확인 및 수정 가능
  - 메모리는 대화 중 자동으로 생성되며,GPT가 "이 내용을 기억해도 될까요?"라고 물어봄



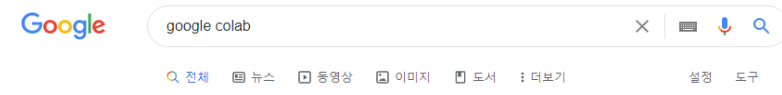
## II. 주요 기능별 실습

- 메모리, 아카이브, 컨텍스트, 챗
  - 챗: 대화
  - 컨텍스트: 챗의 모음
  - 대화: 챗과 컨텍스트의 모음, 2개월간 저장
  - 아카이브: 영구 저장된 대화
  - 메모리: 아카이브 중 주요 내용을 항상 활용할 수 있도록 준비

## II. 업무 활용 사례

### • 실습 – Colab 활용

- 구글 Colab?
- 클라우드 상에서 파이썬 사용 지원
- 구글 드라이브



검색결과 약 4,030,000개 (0.38초)

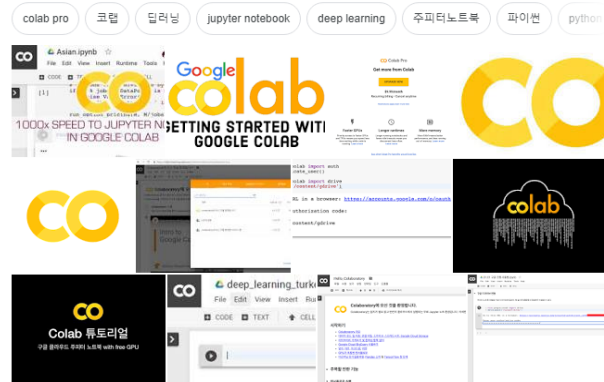
colab.research.google.com ▾ 이 페이지 번역하기

Google Colab

Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the ...

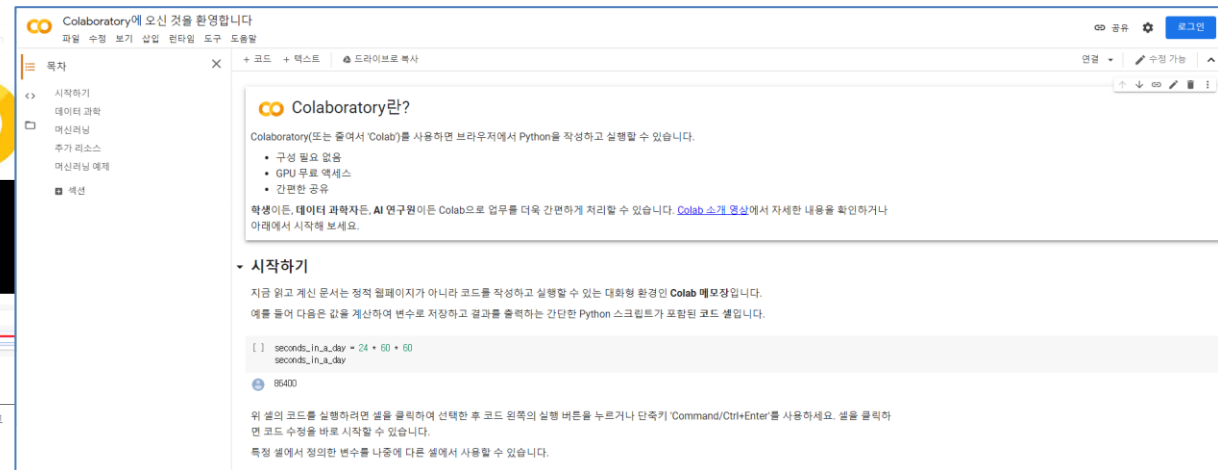
[Introduction to Colab and Python](#) · [Overview of Collaboratory](#) · [This notebook](#) · [Forms](#)

google colab 관련 이미지



→ [google colab에 대한 이미지 더보기](#)

이미지 신고







## II. 업무 활용 사례

### • 실습 – Colab 활용

Colaboratory에 오신 것을 환영합니다

파일 수정 보기 삽입 런타임 도구 도움말

공유 로그인

목차

- 시작하기
- 데이터 과학
- 머신러닝
- 추가 리소스
- 추천 예시
- 섹션

Colab 시작 페이지

Colab에 이미 익숙하다면 이 동영상을 통해 양방향 테이블, 코드 실행 기록 보기, 명령어 팔레트에 관해 알아보세요.

3 Cool Google Colab Features

Colab이란?

Colaboratory(줄여서 'Colab'이라고 함)을 통해 브라우저 내에서 Python 스크립트를 작성하고 실행할 수 있습니다.

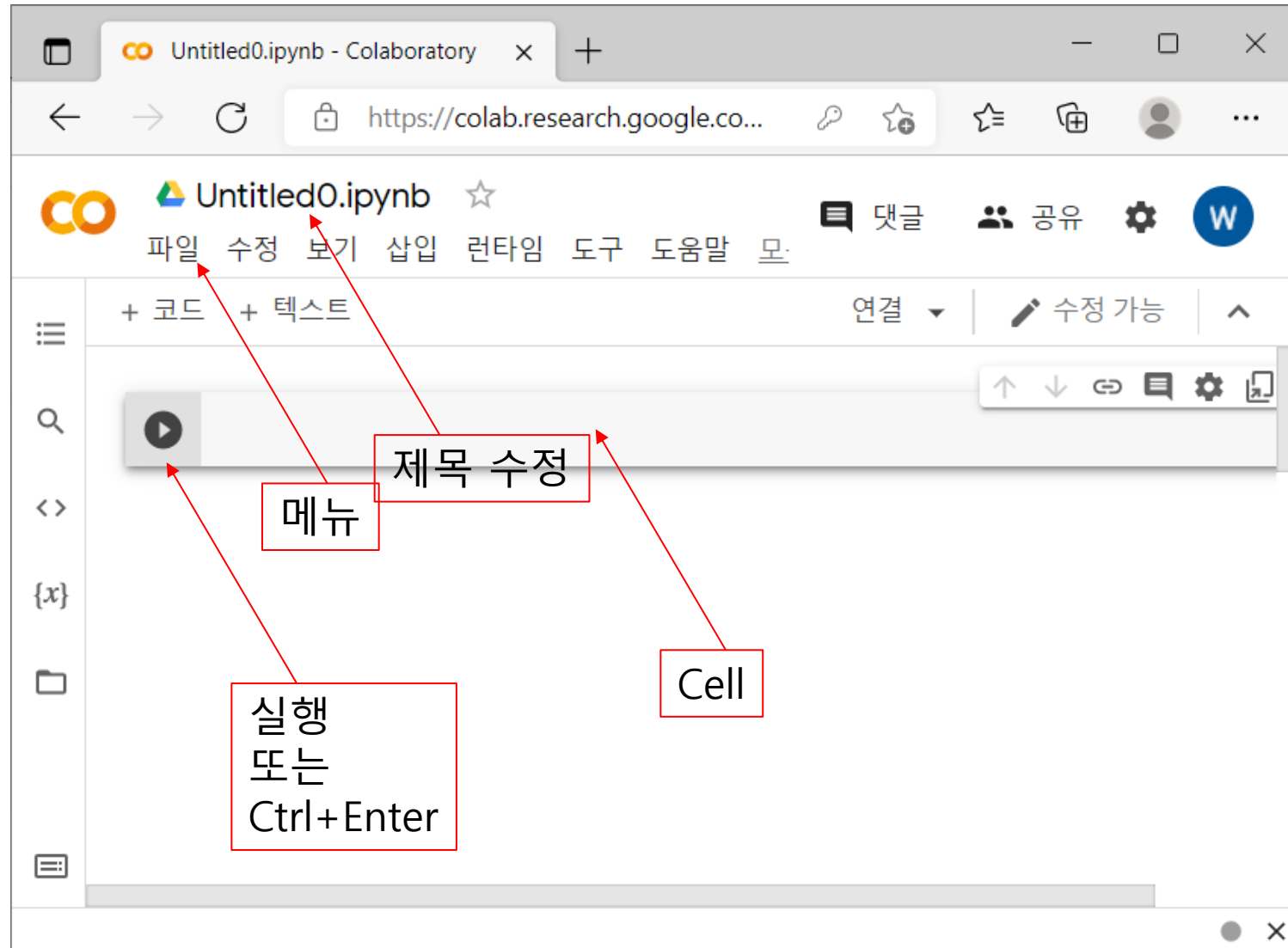
- 구성이 필요하지 않음
- GPU 무료 액세스
- 간편한 공유

학생이든, 데이터 과학자든, AI 연구원이든 Colab으로 업무를 더욱 간편하게 처리할 수 있습니다. [Colab 소개 영상](#)에서 자세한 내용을 확인하거나 아래에서 시작해 보세요.

시작하기

## II. 업무 활용 사례

- 실습 - Colab 활용



## II. 업무 활용 사례

- Agent 모드

- 가상 실행 환경과 다양한 내장 도구를 활용하여 복잡적이고 연속적인 지식 기반 작업을 자율적으로 설계·수행하는 모드
- 질의응답을 넘어 웹 탐색, 데이터 수집·분석, 코드 작성·실행, 외부 서비스 연동, 자동화된 반복 작업 등 고차원적 워크플로를 구현
- 가상 화면 상에서 클릭, 텍스트 입력, 스크롤 등 사용자의 행동을 재현하며, 이를 실시간 내레이션으로 제공해 사용자가 작업 과정과 판단 근거를 확인
- 기존 Operator(웹 상호작용)와 Deep Research(심층 분석)를 통합·발전

## II. 업무 활용 사례

- Agent 모드

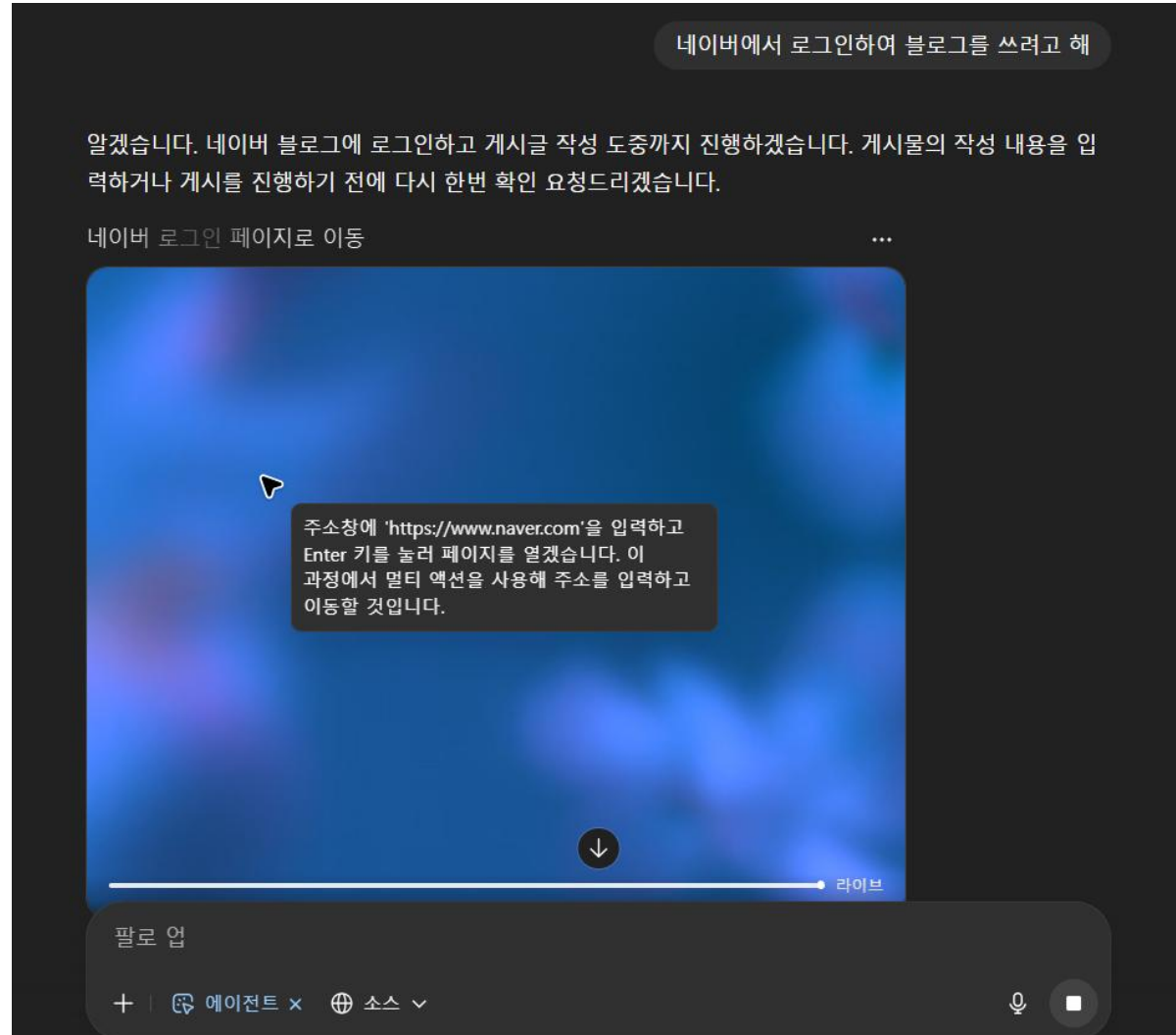
- 지원 플랜: Pro, Plus, Team, Enterprise, Edu.
- 월간 메시지 한도: Plus 40회, Pro 400회, Team/Enterprise 40회(또는 크레딧 기반). 초기 프롬프트만 카운트하며, 중간 승인·로그인 과정은 제외.
- 작업 소요 시간: 대다수 복합 작업은 5~30분 내 완료되나, 데이터 양과 도구 호출 횟수에 따라 변동 가능.
- 베타 제공: 안정성 및 성능 최적화를 위한 피드백 수집 단계.

- Agent 모드

- 가상 컴퓨팅 환경: 브라우저 조작, 파일 입출력, 웹 양식 작성 등 GUI 상호작용 모사.
- 투명성·통제 메커니즘: 중요 단계 전 사용자 승인 요청, 작업 내역 실시간 시각화.
- Code Interpreter: 데이터 전처리, 통계 분석, 시각화 생성.
- Connectors: Gmail, Google Calendar 등 읽기 전용 연동.
- Terminal: 제한된 명령 실행.
- 인증·보안 절차: 외부 계정 로그인은 사용자가 직접 진행.
- 스케줄러: 작업을 주기적으로 자동 재실행.

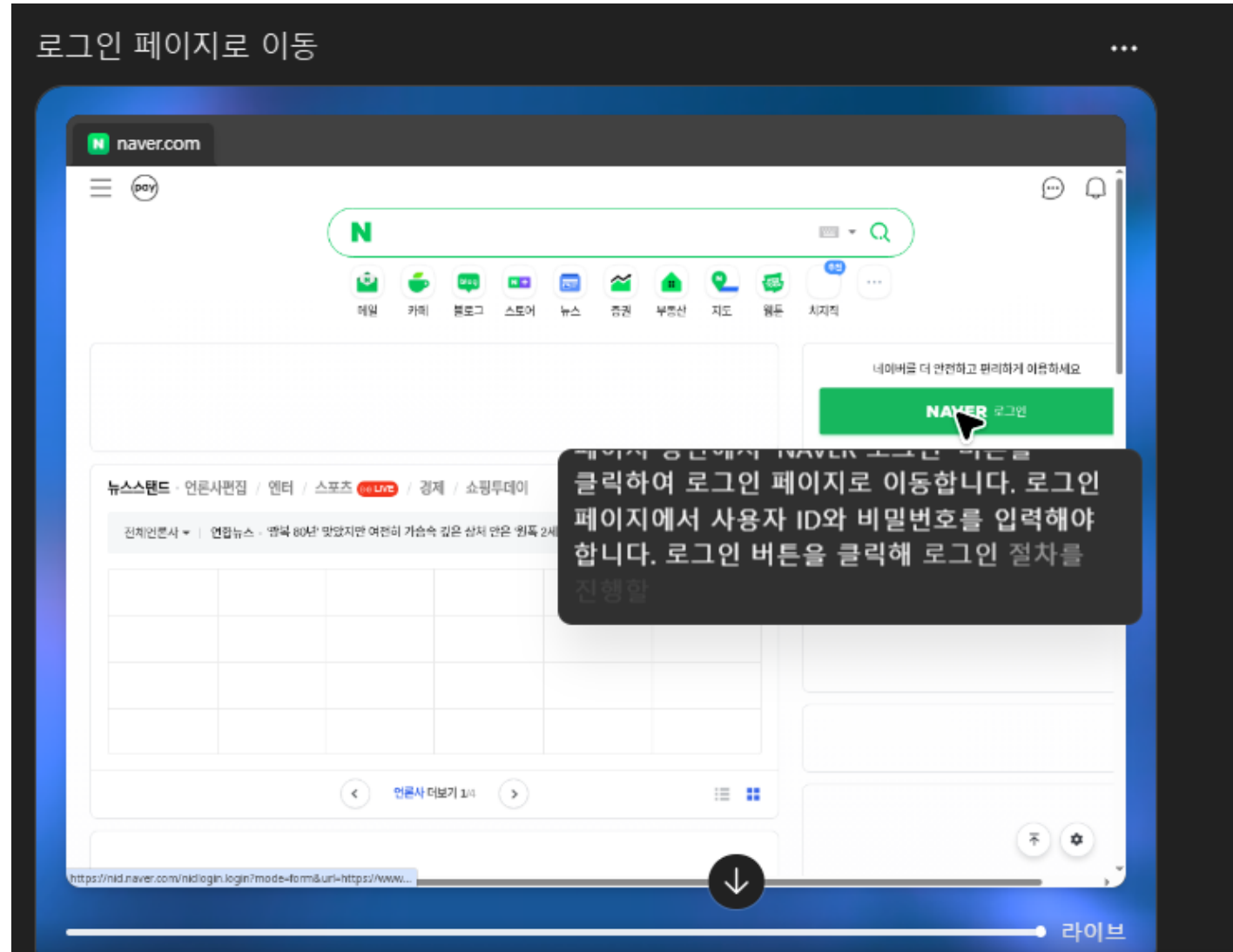
## II. 업무 활용 사례

- Agent 모드



## II. 업무 활용 사례

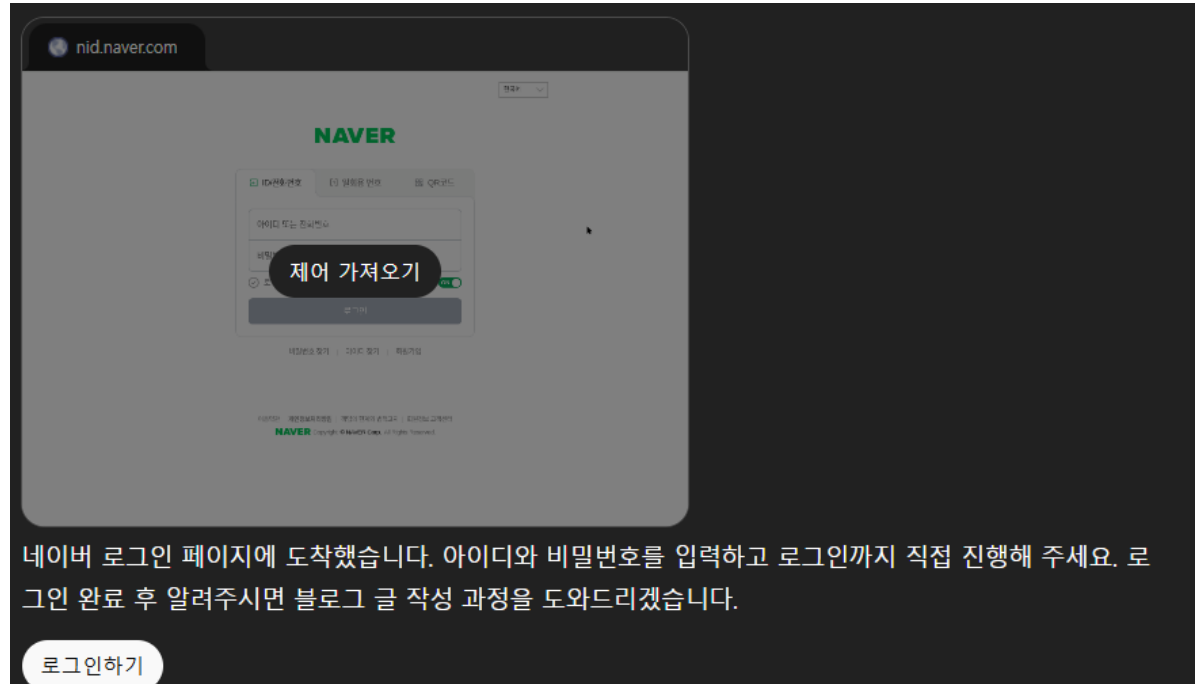
- Agent 모드: Desktop View





## II. 업무 활용 사례

- Agent 모드: 제어 전환



## II. 업무 활용 사례

- Agent 모드 예시

- 예시 1: 학술 리서치 자동화
  - "최신 LLM 기반 학습 분석 논문 5편 비교 후, 참고문헌 포함 2쪽 보고서 작성"
  - 논문 검색, 핵심 내용 추출, 비교표 작성, 참고문헌 형식화.
  - 매주 자동 업데이트 리포트로 예약.

## II. 업무 활용 사례

- Agent 모드 예시

- 예시 2: 일정 최적화
  - Gmail·Calendar 커넥터 연동.
  - "이번 주 회의 요청 메일 요약 후, 가능한 시간 3가지 제안"
  - 승인 후 일정 초안 발송.

## II. 업무 활용 사례

- Agent 모드 사용

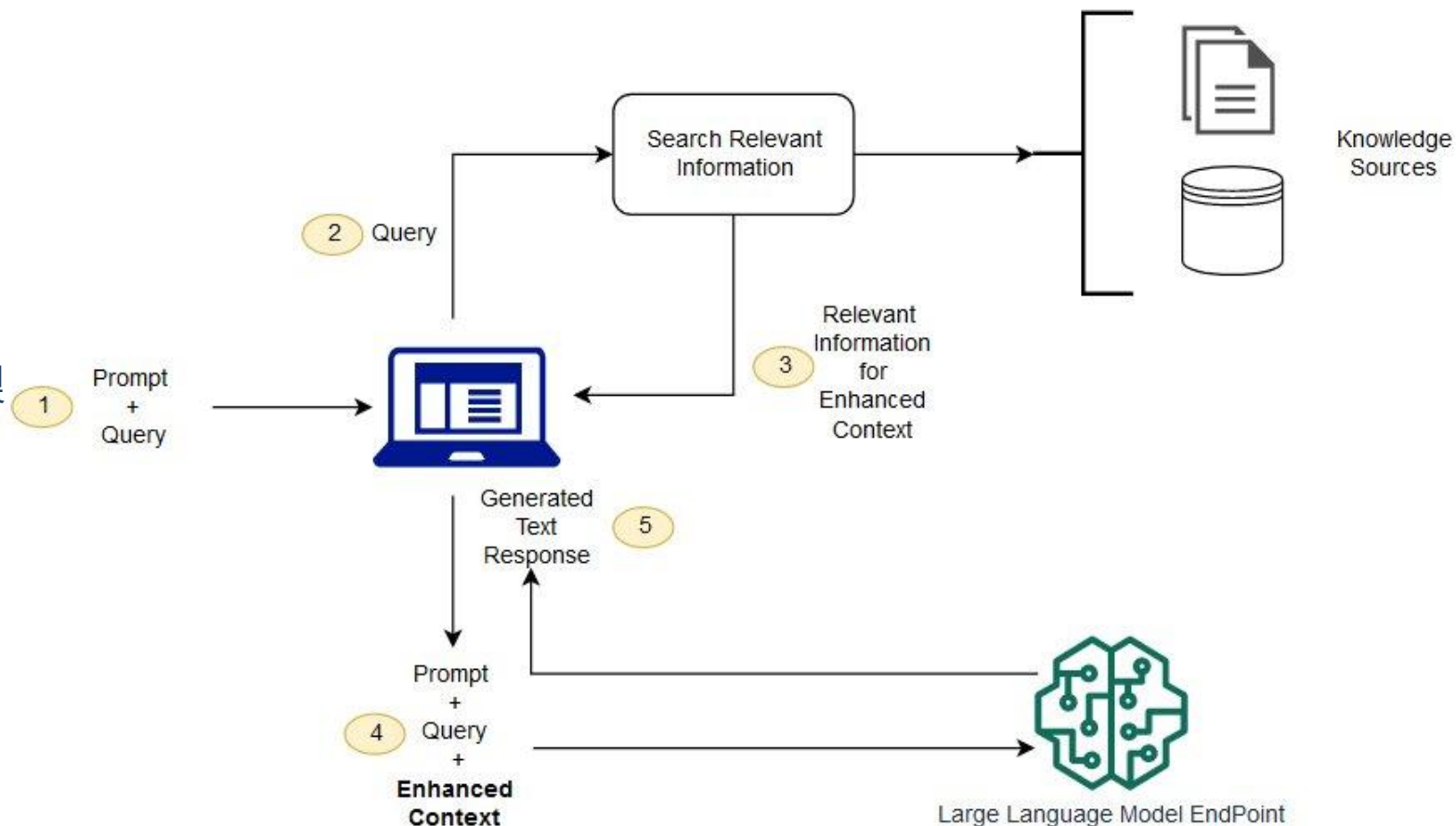
- 프롬프트 엔지니어링
  - 목표, 형식, 제약 조건을 명확히 정의.
  - 데이터 범위, 출처, 배제 기준 설정.
  - 산출물 검증 절차 포함.
- 체이닝 설계
  - 계획 → 실행 → 검토 → 개선 루프 설계.
  - 병렬·순차적 도구 호출 전략 병행.
- 보안·권한 관리
  - 모든 민감 작업은 사전 승인/ 최소 권한 원칙 적용.
- 반복 작업 스케줄링
  - 실행 주기와 조건 명시 후 자동화

### III. RAG VS FineTuning

- **RAG: 검색 증강 생성**
  - Retrieval-Augmented Generation
  - Hallucination을 줄이는 비용 효율적인 접근 방식
    - LLM의 기능을 특정 도메인이나 특정 기관의 내부 지식 기반으로 확장
    - LLM 외부의 데이터를 활용해서 보다 정확한 답변을 찾는 방안
    - 모형 학습을 다시 할 필요는 없음

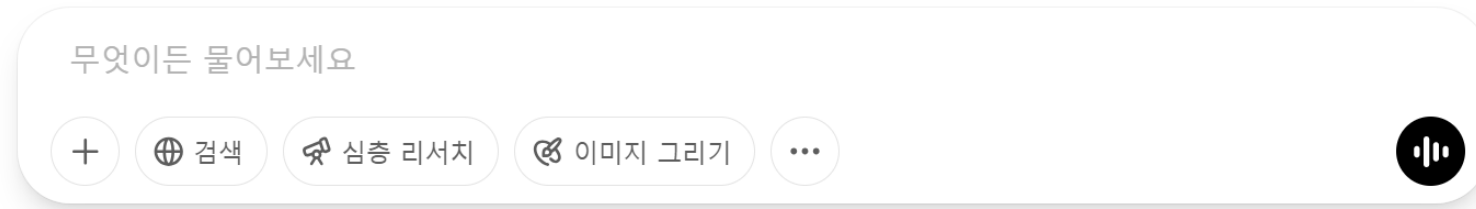
### III. RAG VS FineTuning

- **RAG: 검색 증강 생성**
- LLM 학습 데이터 외의 데이터인 외부 데이터 생성 후 임베딩을 통해 벡터DB저장
- 질문에 대한 관련 정보 검색
- 외부 데이터 업데이트: 실시간 및 배치 방식



### III. RAG VS FineTuning

- RAG: 검색 증강 생성
  - Web Search 기능
    - 웹 검색 및 출처 표시





- **RAG: 검색 증강 생성**

- 절차

- 1단계:

- 사용자 질문 입력사용자가 질문과 함께 프롬프트 입력
      - 예: "2025년 전기차 보조금 정책은 어떻게 바뀌었는지 알려줘"

- 2단계:

- 질문 → 웹 검색 쿼리로 변환
      - 핵심 키워드 중심의 간결한 쿼리 생성예: "2025년 한국 전기차 보조금 변경 내용"

- 3단계:

- 웹 검색 및 정보 수집정부/언론 등 신뢰 가능한 출처 탐색
      - 주요 문장 추출 및 요약예: "보조금 600만 원 → 500만 원 축소, 고가 차량 제외"

- **RAG: 검색 증강 생성**

- 4단계:
  - 확장된 문맥 생성 (Enhanced Context)
  - 외부 정보 + 프롬프트 + 사용자 질문 결합 → LLM 입력용 고도화된 문맥 생성
- 5단계:
  - LLM이 최종 응답 생성요약, 설명, 근거 등 사용자 맞춤 응답 출력
  - 예: "보조금이 500만 원으로 축소되며, 고가 차량은 제외..."
- 6단계:
  - 출처 정보 제공
  - (선택적)신뢰성 확보를 위한 출처 URL 또는 요약 정보 첨부
  - 예: "출처: 국토교통부 공식 보도자료(2025년 2월)"

### III. RAG VS FineTuning

- **Fine Tuning**

- 기존 GPT 모델에 특정 데이터셋을 추가 학습시켜 도메인 특화 모델을 만드는 방법
- GPT의 기본 성능 + 맞춤형 지식/스타일 내재화
- Fine Tuning의 특징
  - 일회성 응답이 아닌 지속적으로 반영되는 학습 결과
  - 다음을 모델 자체에 자동 학습
    - 도메인 지식
    - 브랜드 어조
    - 업무 스타일 등

# III. RAG VS FineTuning

- **RAG VS Fine Tuning**

- Fine-tuning:
  - 모델에 데이터를 "새겨 넣는" 방식
  - 반복되는 업무, 브랜드 톤앤매너, 고정 지식에 강함
- RAG:
  - 최신 정보나 방대한 문서를 "실시간 검색 + 생성"
  - 지속적인 정보 업데이트가 필요한 경우 적합

### III. RAG VS FineTuning

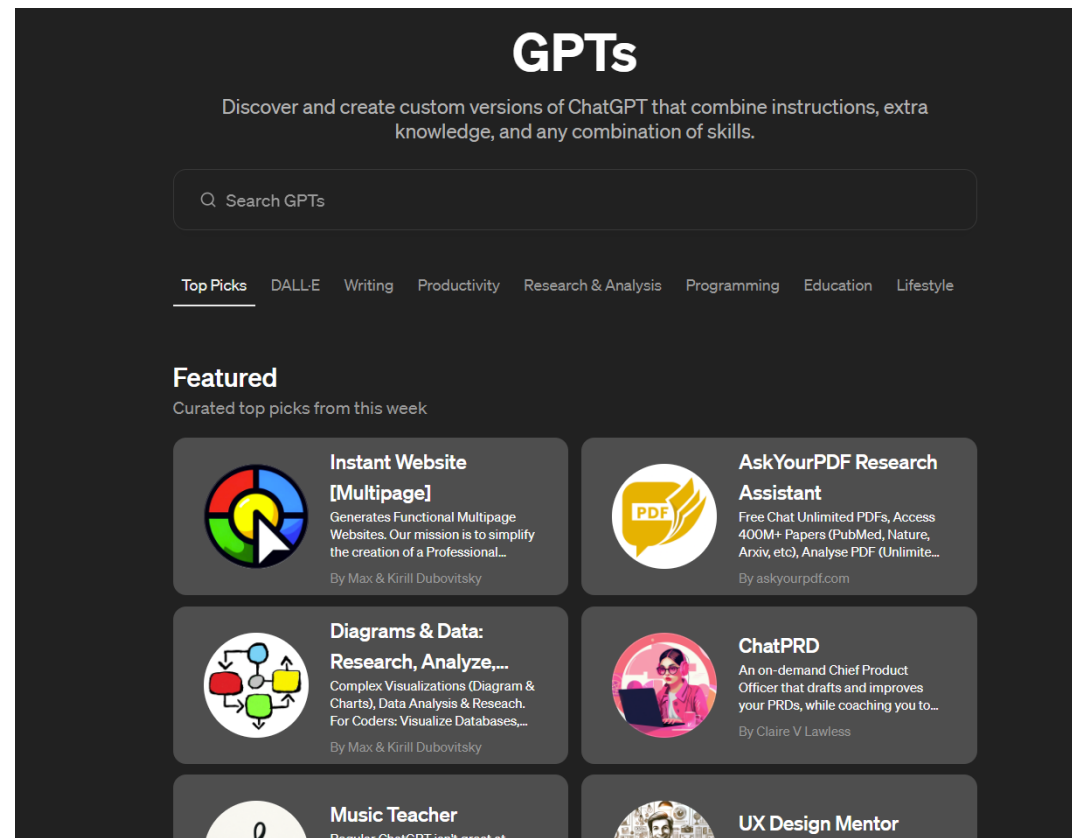
- RAG VS Fine Tuning

	Fine-tuning	RAG (Retrieval-Augmented Generation)
핵심 개념	모델 자체를 <b>재학습</b>	실시간 <b>외부 지식 검색 + 생성</b>
학습 대상	수천 개의 샘플로 모델을 미세 조정	<b>DB, 문서, 위키</b> 등에서 정보 검색
데이터 반영 방식	모델에 <b>내재화</b>	실시간 <b>참조 기반</b>
업데이트 방법	<b>재학습 필요</b> (비용 ↑, 시간 ↑)	문서만 바꾸면 즉시 반영 (유지관리 용이)
일관성	매우 높음 (특화된 도메인에 강함)	검색 결과에 따라 <b>유동적</b>
적합한 경우	브랜드 어조, 반복 질문, 폐쇄형 QA	최신 정보, 광범위한 도메인 지식 활용
예시	고객응대 챗봇, 전용 업무 GPT	법률 검색, 기술 문서 QA, 사내 위키봇

## IV. GPT만들기

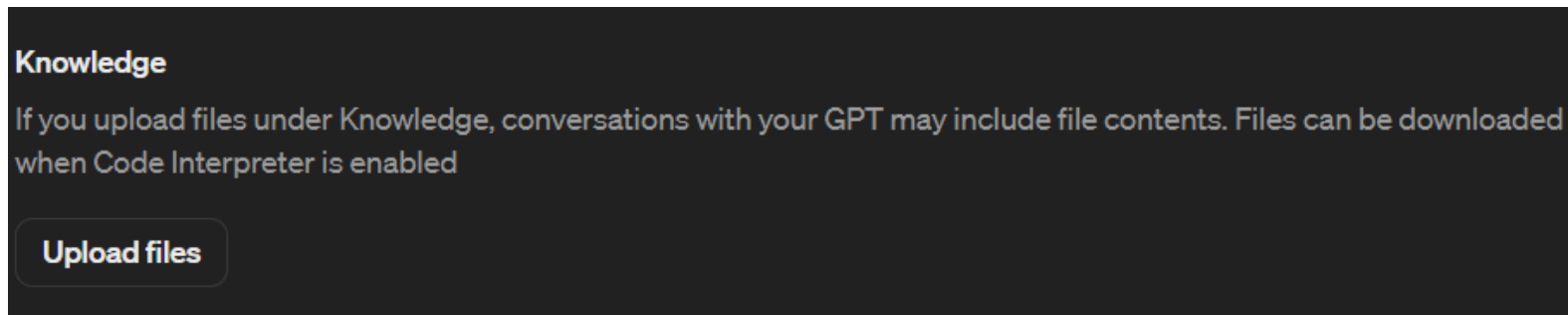
- 커스텀GPTs

- GPTs: Customize된 챗GPT를 의미
- instruction: 제약사항, 가이드라인, 설명, 개인화 등을 확인
- GPT앱마켓

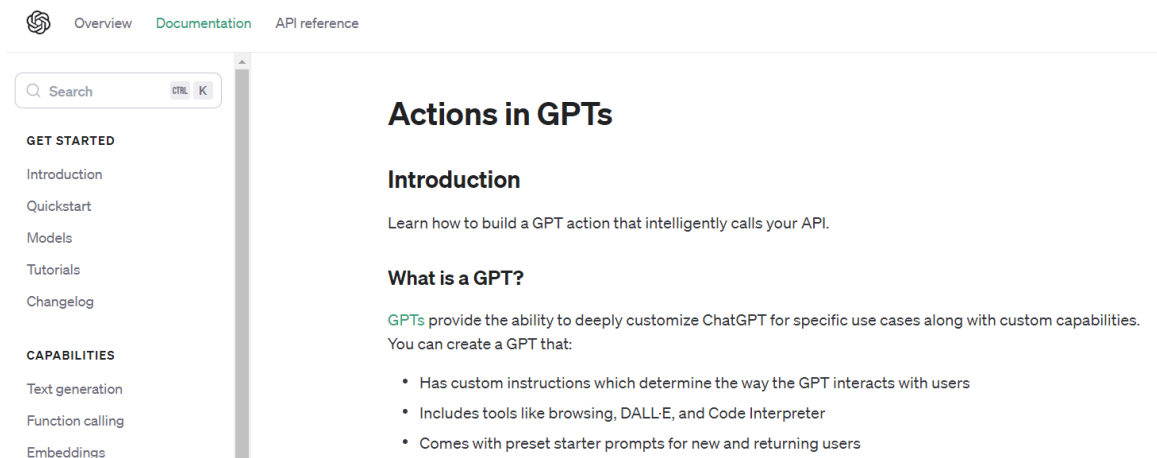


## IV. GPT만들기

- 챗GPT+커스텀GPTs
  - Knowledge에 파일 업로드



- 외부 API의 활용



## IV. GPT만들기

### • GPTs 생성

#### GPTs: Customize된 챗GPT를 의미

- 코딩 불필요 / API를 통하지 않음
- 채팅을 통한 생성

#### GPTs 사용 방법

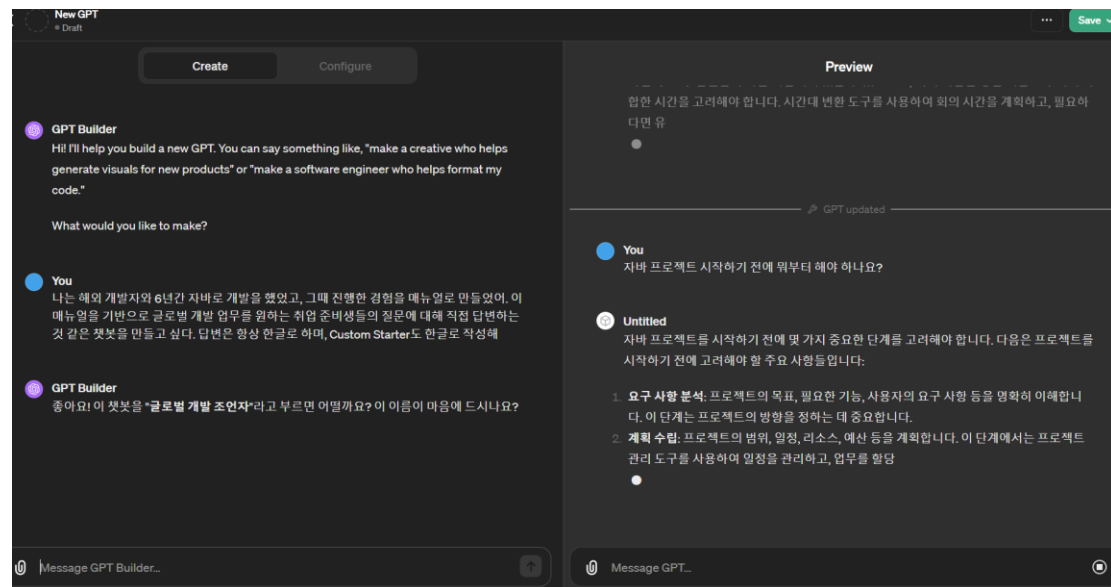
- Explore 클릭 / - 커스텀GPT 제작 또는 사용
- 공식커스텀GPT 또는 그 외 커스텀GPT

#### GPTs 사용 방법

- 일반 유저의 커스텀GPT
- URL을 통한 접근 / GPT스토어를 통한 접근
- 사이드바에 유지 가능

#### 실습:

- Explore -> Create a GPT
- 빌더(좌측), 테스트(우측): 빌더에 커스텀GPT에 대한 설명
- “나는 해외 개발자와 6년간 자바로 개발을 했었고, 그때 진행한 경험을 매뉴얼로 만들었어. 이 매뉴얼을 기반으로 글로벌 개발 업무를 원하는 취업 준비생들의 질문에 대해 직접 답변하는 것 같은 챗봇을 만들고 싶다. 답변은 항상 한글로 하며, Custom Starter도 한글로 작성해
- GPTs의 명칭 등 작성 / 관련 이미지는 자동으로 지정됨





## IV. GPT만들기

### • GPTs 생성

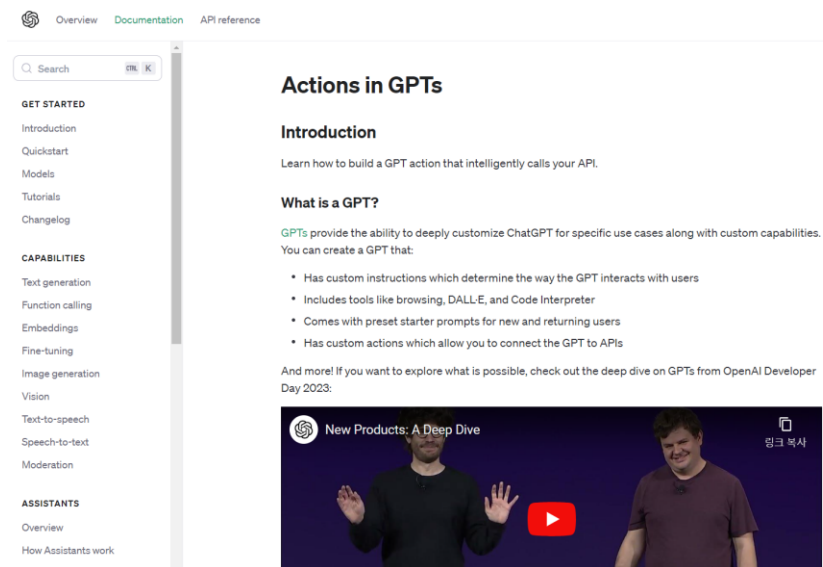
- **Customize:** 웹브라우징 / 캔버스 / 달리 이미지 생성기 / 고급데이터분석 설정

### • Actions

- API? API는 컴퓨터나 컴퓨터 프로그램 사이의 연결이다. 일종의 소프트웨어 인터페이스이며 다른 종류의 소프트웨어에 서비스를 제공
- YAML, JSON, 그 외 양식 지원
- **참고:** <https://platform.openai.com/docs/actions/introduction>

### 앱 저장과 공유

- Save 버튼: 공유 방식의 설정: Only me, Only people with a link, Public



## IV. GPT만들기

- 외부 서비스와의 연동(SerpAPI)

스키마

https://...

가져오기

취소

```
enum : [ naver ]
}
},
{
  "name": "query",
  "in": "query",
  "description": "검색어",
  "required": true,
  "schema": {
    "type": "string"
  }
},
{
  "name": "where",
  "in": "query",
  "description": "검색 위치 (예: web)",
  "required": true,
  "schema": {
    "type": "string",
    "enum": ["web"]
  }
},
{
  "name": "api_key",
  "in": "header",
  "description": "API 키",
  "required": true,
  "schema": {
    "type": "string"
  }
}
```

형식

가능한 작업

이름	메서드	경로	
naverSearch	GET	/search	테스트



## Industrial Data Science Lab

Contact:

won.sang.l@gmail.com

<https://sites.google.com/view/idslab>