

# 조선산업 LLM 활용 과정

---

위블 CEO 노규남  
bardroh@weable.ai

2025/07/21

- 클라우드 업체(가비아)와 회선 업체(케이아이엔엑스) CTO 역임
- 현재는 인공지능 스타트업인 위블(<https://www.weable.ai>) 의 CEO로 재직 중: 머신러닝과 클라우드, 블록체인, DevOps 등에 대한 컨설팅 및 솔루션 개발을 영역으로 비즈니스를 전개, 교육사업도 진행
- AI: GBM 등 Classical Machine Learning부터 CNN, RNN, Transformer 기반의 NLP, DQN, CFR, LLM까지 머신러닝 전반에 걸친 솔루션 개발 및 컨설팅, 교육
- 클라우드: Public, Private, Hybrid IaaS, docker, kubernetes, MSA(MicroService Archiectrue), CI/CD 등 클라우드 네이티브한 영역에서의 개발 및 운영 역량 보유
- 블록체인: Private 또는 Permissioned Blockchain인 Hyperledger Fabric 기반의 네트워크 구축/운영, Public Blockchain과의 연동(Hyperledger Fabric 을 메인체인으로 활용하고 중요한 트랜잭션만 이더리움에 기록하는 이원화 구조)

## 검색 증강 챗봇

선택 업로드

검색에 사용할 파일을 선택해주세요.

(교육교재) 삼성전자 CRA 과...

파일 삭제

글포인트 쪽으로 간너씨비드린 투겁에사 인니. 이 공개 아진 고드시 노 난걸세자 푸시포구시  
용이해집니다.

- Dependency Inversion Principle (의존성 역전 원칙): 고수준 모듈은 저수준 모듈에 의존해서는 안 된다. 즉, 추상화된 인터페이스에 의존하게 되어야 한다. 이렇게 설계되면 코드가 더 유연해져 확장성이 좋아집니다.

챗봇 리스코프 치환 원칙에 대해서 더 상세하게 설명해주세요.

챗봇 리스코프 치환 원칙(Liskov Substitution Principle, LSP)은 객체지향 프로그래밍에서 중요한 원칙 중 하나입니다. 이 원칙은 상위 타입의 인스턴스를 하위 타입의 인스턴스로 교체할 수 있는지 여부에 대한 규칙을 제공합니다. 즉, 프로그램의 다른 부분이 상위 타입으로 선언된 변수에 하위 타입의 객체를 할당할 수 있는지 확인하는 것입니다.

챗봇 리스코프 치환 원칙은 다음과 같은 조건을 만족해야 합니다:

- 상위 타입의 객체와 하위 타입의 객체가 서로 대체 가능해야 한다. 이렇게 하면, 프로그램의 다른 부분은 상위 타입으로 선언된 변수에 상위 타입의 객체를 할당하더라도, 실제로 하위 타입의 객체가 할당되어 있는 경우에도 정상적으로 작동해야 합니다.
- 하위 타입의 객체가 상위 타입의 인터페이스를 구현해야 한다. 이렇게 하면, 상위 타입의 변수에 하위 타입의 객체를 할당하더라도, 상위 타입의 메서드와 프로퍼티를 사용할 수 있습니다.
- 하위 타입의 객체가 상위 타입의 인터페이스를 구현할 때, 상위 타입과 동일한 방식으로 작동해야 합니다. 즉, 상위 타입과 하위 타입 간에는 일관성이 있어야 합니다.

챗봇 리스코프 치환 원칙은 코드의 유연성과 재사용성을 높여주며, 객체지향 설계를 더 안정적으로 만들어줍니다. 이 원칙을 따르면, 프로그램의 다른 부분에서 상위 타입의 변수에 하위 타입의 객체를 할당하더라도, 프로그램이 정상적으로 작동하게 됩니다. 이렇게 함으로써, 코드가 더 유연하고 재사용 가능해지며, 설계의 안정성이 향상됩니다.

대화를 입력해주세요.



X

⋮

## NL2SQL

회사의 데이터에 대해서 자연어로 질의하신다면 해당하는 BigQuery 쿼리를 생성합니다.

상세하게 질의하실 수록 정확한 답을 얻을 수 있습니다.

데이터 카탈로그는 매일 업데이트 됩니다.



2023년 9월 1일부터 2023년 9월 30일까지 '닥터핀스' 브랜드의 재고 총합을 구해줘.



```
SELECT SUM(quantity) AS total_stock
FROM company_in.daily_stock_shop_stock
WHERE date_logged BETWEEN '2023-09-01' AND '2023-09-30'
AND brand = '닥터핀스';
```



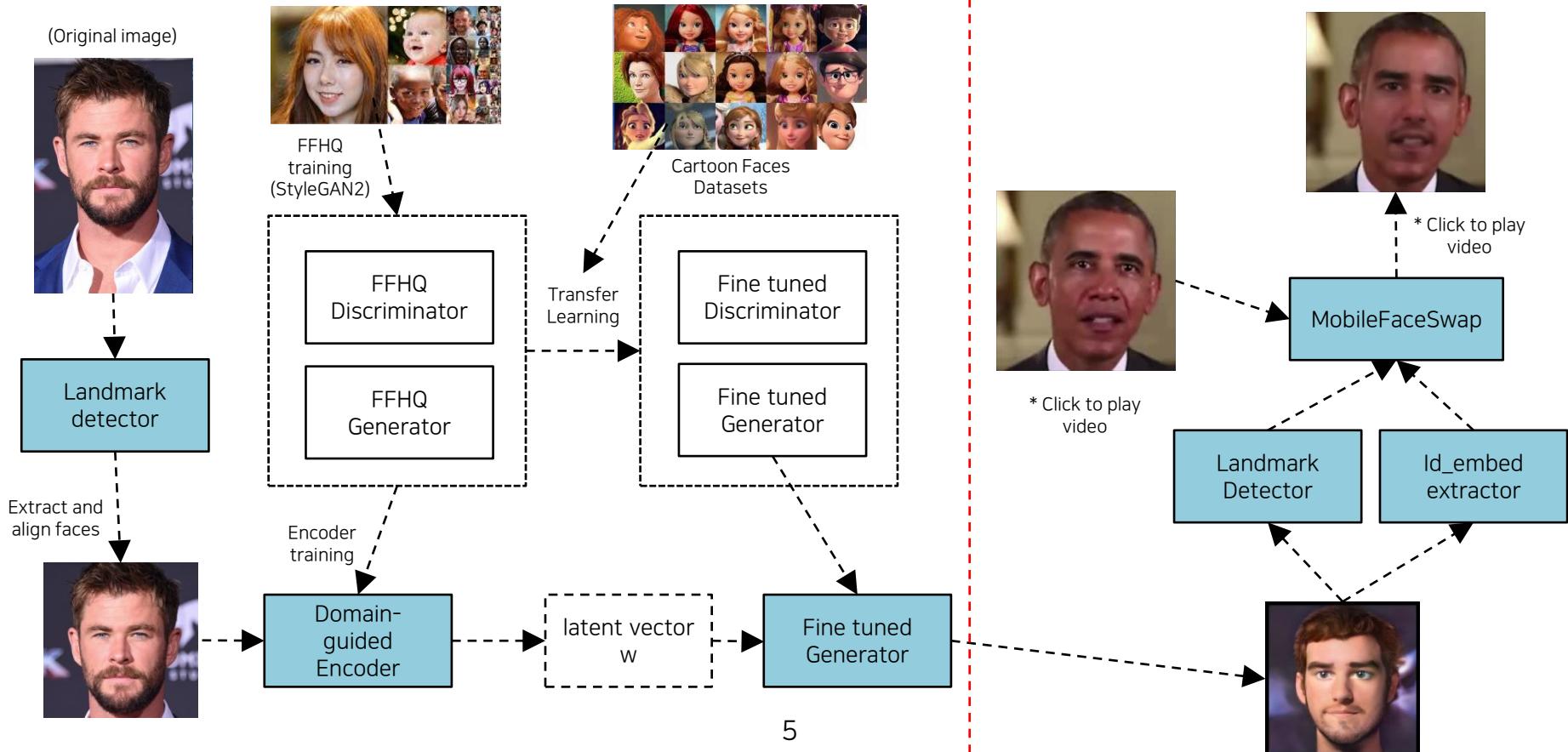
2023년 데이터로부터 전체 브랜드의 월별 재고 평균을 구해줘.

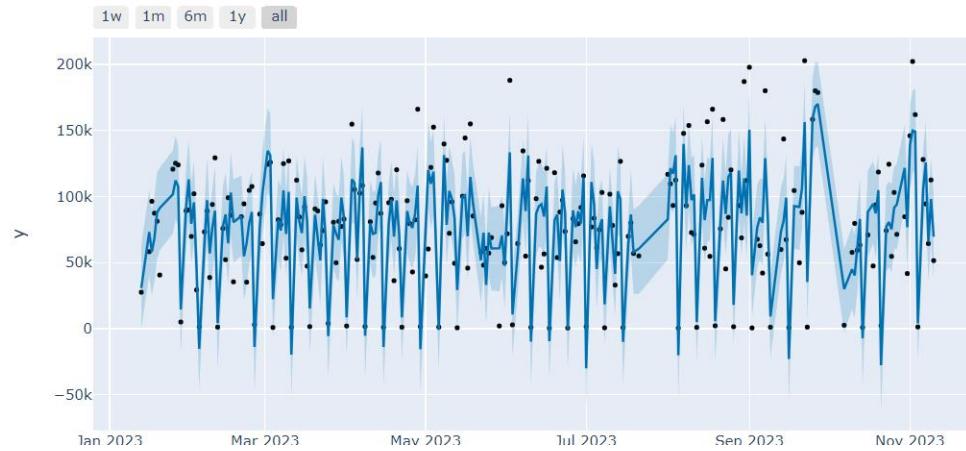
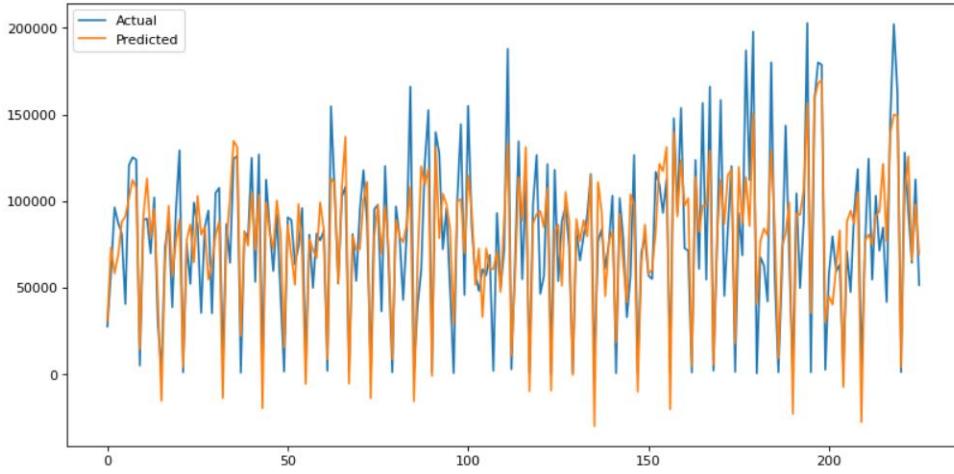


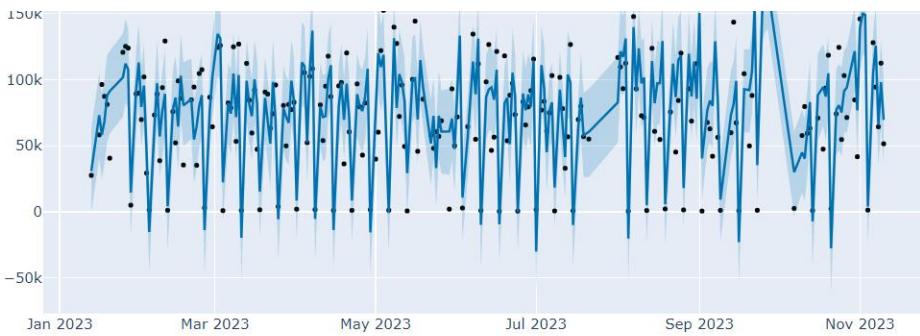
```
SELECT
    EXTRACT(YEAR FROM date_logged) AS year,
    EXTRACT(MONTH FROM date_logged) AS month,
    brand,
    AVG(quantity) AS average_stock
FROM
    company_in.daily_stock_shop_stock
WHERE
    EXTRACT(YEAR FROM date_logged) = 2023
GROUP BY
    year, month, brand
ORDER BY
    year, month;
```

대화를 입력해주세요.

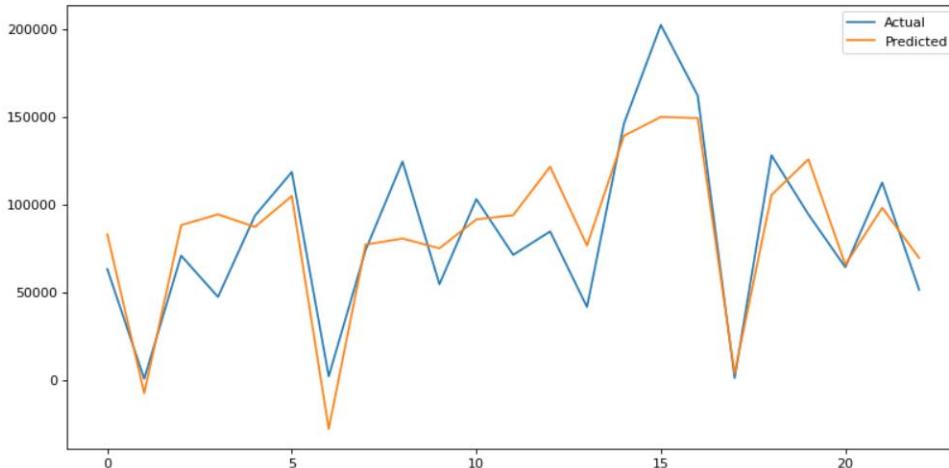


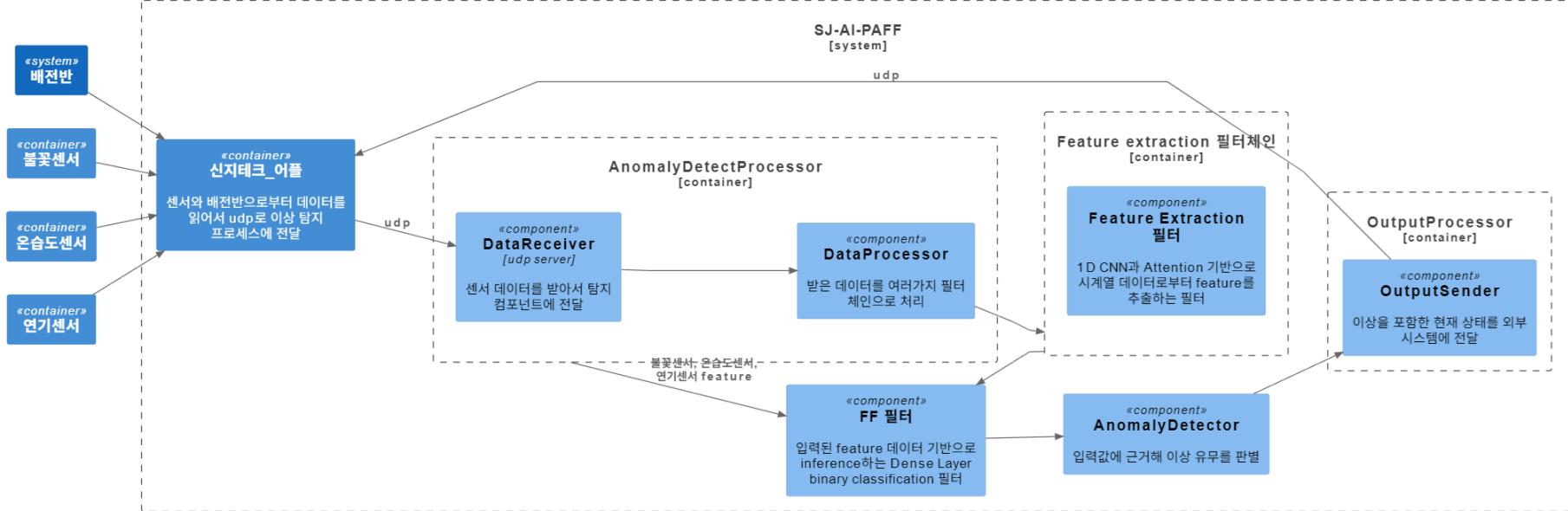






ds





- 이 과정의 주제는 ChatGPT를 포함한 LLM 모델들을 사용하는 방법과, 이 모델들을 다양한 라이브러리와 연동하여 실제 필요한 조선산업에 필요한 서비스를 탐구하고 만들어보는 것
- 교육시간은 월~금요일 오전 8시~오후 5시까지, 일일 8시간, 전체 40시간 분량
- 매 시간 정각에 시작하며, 50분 수업 후 10분 휴식을 반복
- 수강생들 모두가 포함된 git repo(github)가 있음: 메일로 초청하여 멤버가 되며 이를 통해 필요한 파일들을 전달하고 실습, 과정 이해도를 간접적으로 평가

[https://forms.gle/5Trdxc5VkY  
beUW5R6](https://forms.gle/5Trdxc5VkYbeUW5R6)

- ChatGPT는 OpenAI에서 개발한 언어 모델(LLM)로, 자연어 이해 및 생성에 대한 혁신적인 능력을 제공
- 대화형 인터페이스를 통해 사용자와 상호작용하며, 텍스트 기반의 질문 응답, 대화 생성, 자동 요약 등 다양한 언어 작업 수행 가능
- 가입은 chatgpt.com, 무료계정은 ChatGPT로 모델이 고정되어 있으며, 한달에 \$20의 비용을 지불하면 ChatGPT plus로 모델을 업그레이드할 수 있음: 모델의 성능이 좀더 우수하며, 여러가지 플러그인을 활용하여 대화 외에 좀더 상세한 작업에 사용 가능
- 무료, 유료 모두 일정 수준의 request rate limit이 있으며, 그 이상의 요청에 대해서는 블록처리됨
- API로도 접근가능하나 API는 사용하는 token 양에 비례하는 요금을 과금
- 신용카드를 등록한 신규계정에 대해서는 3개월 기한의 \$18 -> \$5의 무료 크레딧이 제공되었으나 현재는 \$0로 시작: 일단 크레딧을 충전하고 난 후에 사용할 수 있음(최소 충전은 \$5)

- ChatGPT는 현재 나와 있는 언어 모델 중 가장 우수한 종류지만 ChatGPT만으로 AI 서비스를 만들 수는 없음: 실제 서비스 구축을 위해서는 LLM 외 다른 요소들이 상당히 많이 필요하다.
- ChatGPT를 실제 서비스에 적용하려면 부족한 점들이 상당히 있으며, 가격이 내려감과 동시에 점점 더 성능이 낮아지고 있다. API도 계속 변경된다. ChatGPT를 사용하려면 이런 변화에 지속적으로 대응해야 한다.
- 사용량이 많아지면 그에 비례하여 비용이 증가하며, 채팅을 통해 회사 내부의 정보가 외부로 흘러나갈 수 있다는 점을 고려해야 한다: 자체 인프라를 구축하거나 인터페이스에 필터를 적용할 수도 있다(가드레일).
- AI 서비스를 구축하는 방법에 대해서는 비교적 잘 알려져 있는 편이나, 서비스하기에 충분한 품질을 얻는 것은 생각보다 많이 어렵다. 대기업들(예. SK 텔레콤)도 비슷하게 헤매고 있다.
- AI 서비스를 구축할 때는 반드시 원하는 성능 지표를 설정하고, 측정을 위한 데이터셋을 준비해야 한다.
- ChatGPT의 비용은 성능을 고려할 때 매우 낮은 편이나, 만약 호출이 매우 많아지거나 context를 매번 길게 사용한다면 비용은 빠르게 증가한다. 항상 상위 모델을 사용할 필요는 없으며 적절한 모델을 선택하면 이런 비용은 최적화될 수 있다.

- ChatGPT는 API만 연결하면 되므로 특정 환경에 구애받지 않음
- 다만 결과를 확인하기 위해서는 좀더 interactive한 환경이 적합: jupyter notebook이나 Google colab
- Google colab 세션은 계속 유지되지 않으므로 사용하기에 적당하지 않으며, git repo 연동도 어려움
- 이 과정에서는 로컬 PC에 wsl을 설치하고, wsl에서 python환경을 구축한 후 jupyter notebook을 실행
- WSL2는 Windows Subsystem for Linux2로, 윈도우에서 리눅스 어플리케이션을 구동하게 해주는(정확하게 말하면 Linux system call을 처리해주는) 호환성 레이어로, 리눅스 커널의 기능을 제공
- WSL2를 사용하면 윈도우안에서 리눅스 배포본을 구동할 수 있고, 리눅스 어플리케이션도 모두 실행할 수 있음: 거의 대부분의 어플리케이션을 무리 없이 실행 가능
- 여러가지 오픈소스 프로젝트들이 리눅스를 기준하여 개발되고 있으므로, 윈도우에서의 설치로는 한계
- 가상화 기능이 있는 Windows 10 이상의 PC에서는 모두 사용 가능: Mac 사용자는 wsl을 사용할 필요가 없으며 자체적으로 python과 pip, jupyter, git을 설치해서 사용하면 됨

- 제어판 > 프로그램 > 프로그램 및 기능 > Windows 기능 켜기 또는 끄기
- Linux용 윈도우 하위시스템 켜기, 가상 머신 플랫폼 켜기
- 재부팅 후 Microsoft Store에서 Ubuntu를 검색하고 22.04를 선택하여 설치
- cmd.exe 실행
- wsl --install Ubuntu-22.04로 설치, ubuntu 계정 생성하고 암호 설정
- wsl -l -v로 현재 설치된 배포본 확인
- wsl -d Ubuntu-22.04로 wsl 콘솔 진입
- 종료는 wsl --terminate Ubuntu-22.04
- 배포본 삭제는 wsl --unregister Ubuntu-22.04

- 1) github 계정 생성
- 2) wsl 로그인, 작업 디렉토리 생성

```
mkdir ~/work  
cd ~/work
```

- 3) 계정 설정

```
git config --global user.name "Bard K. Roh"  
git config --global user.email "bardroh@weable.ai"  
git config --global --list
```

4-1) ssh키 설정 & git clone

```
ssh-keygen
```

```
cat ~/.ssh/id_rsa.pub
```

화면의 내용을 전부 복사해서 github의 Settings-SSH Keys에 붙임

```
git clone git@github.com:weable-weable/koshipa-llm-2025-1st.git
```

4-2) github token 설정 & git clone

https clone시에는 basic token 설정, repo 관련 모든 권한 할당

```
git clone https://[사용자 id]:[access token]@github.com/weable-weable/koshipa-llm-2025-1st.git
```

5) 작업하고 git add; git commit; git push, 작업 전 항상 git pull (본인 폴더만 작업)

- <https://platform.openai.com/>에 접속하여 가입하고 카드 등록. Google, Apple, MS 계정으로 가입 가능
- token 사용량에 따라 다음 달에 과금하는 구조였으나, Credit Balance에서 충전하고 이를 소진하는 방식으로 변경되었음: Credit Balance가 \$0.00이 되면 rate limit이 걸렸을 때와 유사하게 429 에러가 발생, 최소 충전 금액은 \$5
- Auto recharge 옵션을 적용해 두면, Credit Balance가 일정 수준 이하로 떨어졌을 때 자동으로 일정액씩 충전하도록 해둘 수 있음
- API key를 생성하여 연동할 수 있으며 한번 생성된 key는 다시 보여지지 않음: 삭제하고 재생성
- 사용을 위해 python의 openai 패키지 설치가 필요(pip3 install openai)

VS Code Extension에서 WSL을 찾아서 설치 후 종료

이후 WSL 실행

```
sudo su - root
```

```
apt-get update
```

```
apt-get install python3 python3-pip -y
```

```
pip3 install openai
```

```
cd <git repo 경로>
```

```
export OPENAI_API_KEY=<자신의 OPENAI API KEY>
```

```
echo $OPENAI_API_KEY
```

```
code .
```

## Visual Studio Code

[Docs](#) [Updates](#) [Blog](#) [API](#) [Extensions](#) [FAQ](#) [Learn](#)

Search Docs

Download

[Version 1.82](#) is now available! Read about the new features and fixes from August.

# Code editing. Redefined.

Free. Built on open source. Runs everywhere.

[Download for Windows](#)  
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its  
[license](#) and [privacy statement](#).

The screenshot shows the Visual Studio Code interface. On the left, there's a sidebar with icons for file operations like Open, Save, Find, and Copy/Paste. Below it is the 'EXTENSIONS: MARKETPLACE' section, which lists several popular extensions with their names, versions, sizes, ratings, and 'Install' buttons:

- Python 2019.6.24221 (54.9M) ★ 4.5
- GitLens — Git sup... 9.8.5 (23.1M) ★ 5
- C/C++ 0.24.0 (23M) ★ 3.5
- ESLint 1.9.0 (21.9M) ★ 4.5
- Debugger for Ch... 4.11.6 (20.6M) ★ 4
- Language Supp... 0.47.0 (18.7M) ★ 4.5
- vscode-icons 8.8.0 (17.2M) ★ 5
- Vetur 0.21.1 (17M) ★ 4.5
- C# 12.1.0 (15.6M) ★ 4

The main area is a code editor with three tabs: 'src > JS serviceWorker.js', 'JS App.js', and 'JS index.js'. The 'serviceWorker.js' tab contains code related to service workers and registration:

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
checkValidServiceWorker(swUrl, config);

// Add some additional logging to localhost, p...
// service worker/PWA documentation.

navigator.serviceWorker.ready.then(() => {
  product
  productSub
  removeSiteSpecificTrackingException
  removeWebwideTrackingException
  requestMediaKeySystemAccess
  sendBeacon
  serviceWorker (property) Navigator.serviceWorker
  storage
  storeSiteSpecificTrackingException
  storeWebwideTrackingException
}) userAgent
} vendor

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

The status bar at the bottom shows the file path 'serviceWorker.js - create-react-app - Visual Studio Code - In...', the current file 'serviceWorker.js', the line number '1: node', and the status '1: node'. It also displays the commit message 'master', the status '0 ▲ 0', and the terminal status 'TERMINAL ...'. The bottom right corner shows the status bar with 'Ln43, Col19 Spaces:2 UTF-8 LF JavaScript' and a few small icons.

File Edit Selection View Go Run Terminal Help ⏪ ⏩ Search

EXPLORER NO FOLDER OPENED You have not yet opened a folder. Open Folder Opening a folder will close all currently open editors. To keep them open, add a folder instead. Clone Repository You can clone a repository locally. Clone Repository To learn more about how to use git and source control in VS Code read our docs.

Start

- New File...
- Open File...
- Open Folder...
- Clone Git Repository...
- Connect to...

Recent

You have no recent folders, [open a folder](#) to start.

Welcome

# Visual Studio Code

## Editing evolved

Recommended

 GitHub Copilot Supercharge your coding experience for as little as \$10/month with cutting edge AI code generation.

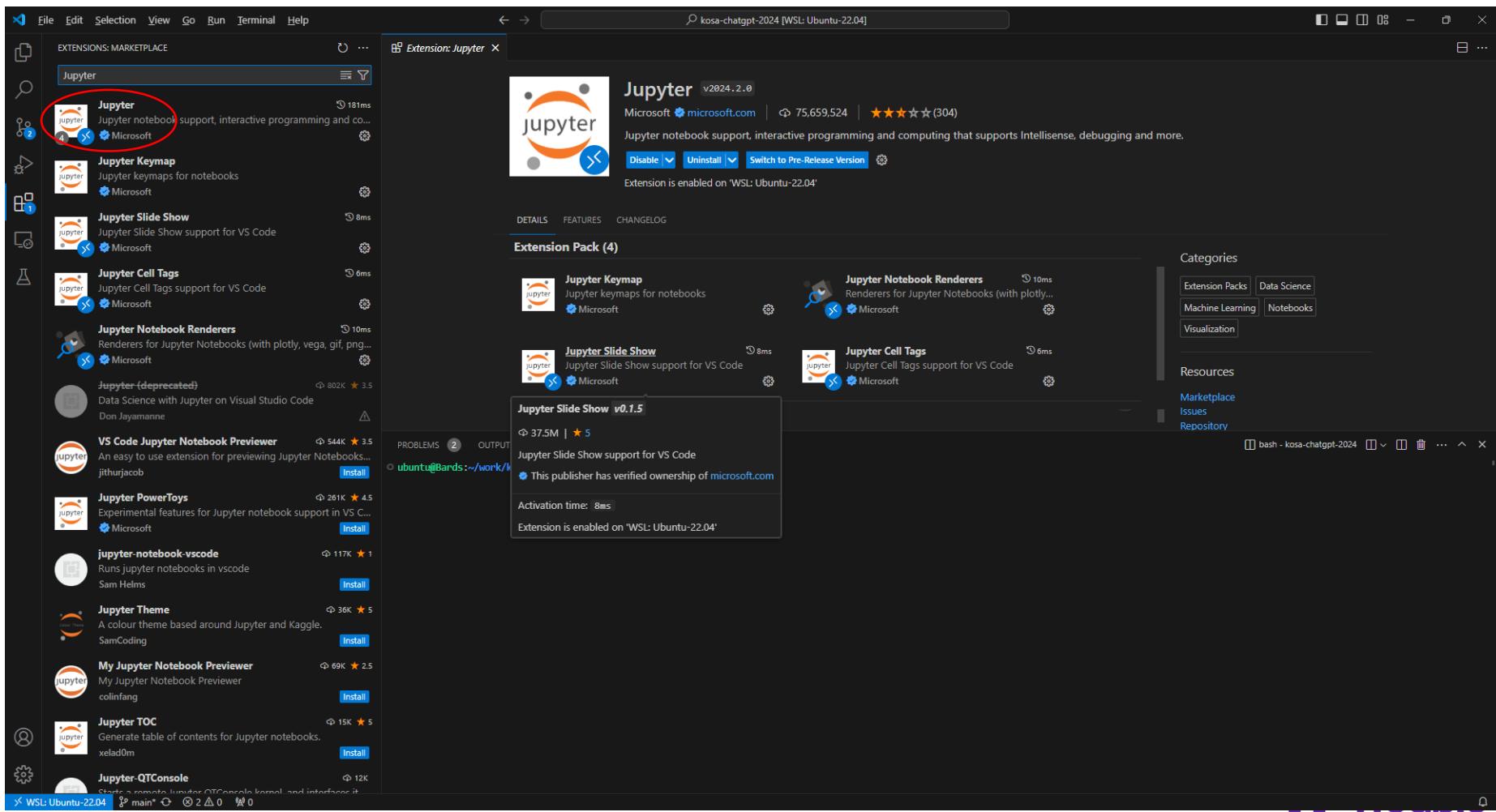
Walkthroughs

 Get Started with VS Code Discover the best customizations to make VS Code yours.

 Learn the Fundamentals

 Boost your Productivity

Show welcome page on startup



File Edit Selection View Go Run Terminal Help

EXTENSIONS: MARKETPLACE

WSL

Open any folder in the Windows Subsystem for Linux (WSL) and take advanta... Microsoft

WSL workspaceFolder

Returns `workspaceFolder` as a WSL path. Useful where WSL format workspace... Ifurzewadock

Linux/Unix/WSL paths

Provides commands to have unix-like paths (with slashes instead of backslash... Krzysztof Grzywocz

Remote Development

An extension pack that lets you open any folder in a container, on a remote m... Microsoft

Open (WSL support)

Opens the current file with the default application for the OS (in Windows for... NoThInG

wsl-split

Iamjava

WSL Path Tools

Tools to convert paths between Windows and WSL format Rob Clouth

OCaml and Reason IDE - for WSL

OCaml and Reason language support for WSL raiscui

Save and Run with wsl

Run commands when a file is saved in Terminal (add wsl path) raiscui

Json Editor for VS code - WSL

Generate a tree view editor of the active JSON document pick68

Laravel Artisan

Run Laravel Artisan commands within Visual Studio Code Ryan Naddy

WSLSchoolAux

Bruno School

WSL: Recommender

Recommendations using the Windows Subsystem for Linux (WSL) and the WSL ext...

7. LCELipynb 8. Assistant-math.ipynb Extension: WSL

WSL v0.86.0 Microsoft microsoft.com 25,383,614 ★★★★★ (77)

Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio Code's full feature set.

Disable Uninstall

This extension is enabled globally.

DETAILS FEATURES CHangelog

## Visual Studio Code WSL

The **WSL extension** lets you use VS Code on Windows to build Linux applications that run on the [Windows Subsystem for Linux \(WSL\)](#). You get all the productivity of Windows while developing with Linux-based tools, runtimes, and utilities.

The **WSL extension** lets you use VS Code in WSL just as you would from Windows.

### Why do I need the WSL extension?

#### Why WSL?

WSL lets you run a Linux environment -- including command-line tools and applications -- directly on Windows, without the overhead of a traditional virtual machine or dualboot setup. WSL especially helps web developers and those working with Bash and Linux-first tools (i.e. Ruby, Python) to use their toolchain on Windows and ensure consistency between development and production environments.

When you install a version of Linux on Windows, you're getting a full Linux environment. It's isolated from Windows- the UI is the

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

```
ubuntu@DESKTOP-JJ3PK15:~/work/kosa-chatgpt-2024$ git pull
Already up to date.
ubuntu@DESKTOP-JJ3PK15:~/work/kosa-chatgpt-2024$
```

Categories

Other

Resources

Marketplace Issues Repository License Microsoft

More Info

Published 2019-05-03, 03:40:03 Last released 2024-01-27, 02:31:24 Last updated 2024-03-01, 10:43:51 Identifier ms-vscode-wsl bash - kosa-chatgpt-2024

Billing overview - OpenAI API

platform.openai.com/account/billing/overview

Billable API requests

Playground Assistants Fine-tuning API keys Storage Usage Settings Organization Team Limits Billing Profile Documentation Help All products Weable, Inc

**Billing settings**

Overview Payment methods Billing history Preferences

**Pay as you go**

Credit balance \$0.00

Auto recharge is off

When your credit balance reaches \$0, your API requests will stop working. Enable automatic recharge to automatically keep your credit balance topped up.

Enable auto recharge

Add to credit balance Cancel plan

**Payment methods**  
Add or change payment method

**Billing history**  
View past and current invoices

**Preferences**  
Manage billing information

**Usage limits**  
Set monthly spend limits

**Pricing**  
View pricing and FAQs

## | 가격

모델	Input /1M	Output /1M	Context window	Max output tokens	비고
o3	10.0	40.0	200K	100K	프롬프트 캐싱
o3-mini	1.1	4.4	200K	100K	프롬프트 캐싱
gpt-4.5	75	150	128K	16K	프롬프트 캐싱
gpt-4o	2.5	10	128K	16K	프롬프트 캐싱
gpt-4o-mini	0.15	0.6	128K	16K	프롬프트 캐싱
o1	15	60	128K	32K	프롬프트 캐싱
o1-mini	3	12	128K	32K	프롬프트 캐싱
gpt-4-turbo	10	30	128K	4K	
gpt-4	30	60	8K	8K	
gpt-3.5-turbo	0.5	1.5	16K	4K	
Code Interpreter			N/A		0.03/session
File Search			N/A		0.10/GB per day (1GB free)

모델	Input /1M	Output /1M	Context window	Max output tokens	비고
o3-mini	1.1	4.4	200K	100K	프롬프트 캐싱
gpt-4o	2.5	10	128K	16K	프롬프트 캐싱
<b>gpt-4o-mini</b>	<b>0.15</b>	<b>0.6</b>	<b>128K</b>	<b>16K</b>	<b>프롬프트 캐싱</b>
o1-mini	3	12	128K	32K	프롬프트 캐싱
gemini 2.5-pro	2.5	15	1M	32K	
gemini 2.5-flash	0.3	2.5	1M	8K	
<b>gemini 2.5-flash lite</b>	<b>0.1</b>	<b>0.4</b>	<b>1M</b>	<b>8K</b>	
gemini 2.0-flash	0.1	0.4	1M	8K	
<b>gemini 2.0-flash lite</b>	<b>0.075</b>	<b>0.3</b>	<b>1M</b>	<b>8K</b>	<b>gpt-4o-mini의 절반 가격</b>
<b>claude 3 Haiku</b>	<b>0.25</b>	<b>1.25</b>	<b>200K</b>	<b>8K</b>	<b>앤스로픽의 최저가 모델</b>
DeepSeek v3	0.07	1.1	64K	8K	
<b>DeepSeek R1</b>	<b>0.55</b>	<b>2.19</b>	<b>64K</b>	<b>32K~64K</b>	<b>추론 모델 최저가</b>

- OpenAI에서 직접 계정을 만들고 API를 연동할 수 있는데, MS Azure 클라우드에서 거의 동일한 모델들을 거의 동일한 가격으로 판매하고 있음
- OpenAI 대비 MS Azure OpenAI Service가 갖는 기본적인 장점이라면, US West 단일리전이 아닌 다른 리전 선택이 가능하며(도쿄 리전이 가장 가까움) 여러가지 클라우드의 보안조치가 되어 있다는 점
- Azure OpenAI의 모델들도 지역 할당량이 있는데, gpt-3.5-turbo 모델 기준하여 분당 120K~300K까지 다양함. gpt-4인 경우는 분당 20~40K
- 할당량이 부족해지게 되면, Azure 포털에서 할당량 증가를 요청할 수 있으며, 수락되면 더 많은 token을 사용 할 수 있게 됨: OpenAI를 대규모 서비스에 적용할 경우 가장 문제가 되는 부분이 이 할당량 quota
- MS Azure의 OpenAI 모델은 별도의 품으로 신청해야 하며, Azure의 체험 계정으로는 사용이 어려움

## | OpenAI vs Azure OpenAI

---

Feature	OpenAI	Azure OpenAI
Security & Data Privacy	Basic Security	Enterprise Security, RBAC, Customer-Managed Keys, Private Link
Compliance	None	SOC2, ISO, HIPAA, CSA STAR
Reliability	No SLA (yet)	Azure SLA, Dedicated Capacity Option (soon)
Responsible AI	Separate Safety Classifier (adds latency)	Built-in, enterprise-grade, low latency moderation and harm prevention
Holistic Solution	Advanced LLM & Image Generation, Basic Speech	OpenAI Models, Complete AI Solution, and a Complete PaaS
Service Region	US West (샌프란시스코) Only	East US Region, South Central US Region, West Europe Region 기타

# OpenAI playground

The screenshot shows the OpenAI Playground interface. On the left, there's a sidebar titled "Get started" with instructions and a "KEEP IN MIND" section containing several tips. The main area is titled "Playground" and contains a "SYSTEM" message box and a "USER" message input field. To the right, there are various configuration options: Mode (set to Chat), Model (set to gpt-3.5-turbo), Temperature (set to 1), Maximum length (set to 256), Stop sequences (empty), Top P (set to 1), Frequency penalty (set to 0), and Presence penalty (set to 0). A note at the bottom states that API and Playground requests will not be used to train our models. The "View code" button in the top right is circled in red.

## | Completion mode (deprecated)

---

```
# Completion mode, 지시와 명령을 하나의 prompt로 전달
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")
# 텍스트 생성 요청을 보냅니다.
response = openai.completions.create(
    model="gpt-3.5-turbo-instruct",
    prompt="다음을 일본어로 번역하세요: 안녕하세요? 오늘 날씨가 참 좋군요!",
    max_tokens=256, # 생성된 텍스트의 최대 길이를 설정합니다.
)
# 생성된 텍스트를 출력합니다.
print(response.choices[0].text)
```

```
# 표준적인 Chat completion mode
```

```
import openai
```

```
res = openai.chat.completions.create(  
    model="gpt-4o-mini",  
    messages=[{"role": "user", "content": "다음을 일본어로 번역하세요: 안녕하세요? 오늘 날씨가 참 좋군  
요!"}],  
    temperature=0.5,  
    max_tokens=256,  
    top_p=1,  
    frequency_penalty=0,  
    presence_penalty=0  
)  
print(res.choices[0].message.content)
```

- system: 주로 채팅 대화의 시작 부분에 사용되며, 시스템 역할로 제공된 메시지는 Assistant에게 초기 컨텍스트를 제공하고 작동 방식을 지정. 예) 당신은 과학에 대한 전문가입니다.
- user: 사용자 역할은 API를 호출하는 클라이언트 측 역할. 이 역할은 Assistant에게 질문을 하거나 지시를 내릴 때 사용. 예) 과학 분야에서 어떤 발전이 있었나요?
- assistant: Assistant는 사용자 역할로부터 받은 메시지에 대답하고 정보를 제공하는 역할. 예) 새로운 에너지 저장 기술에 대해 이야기해드릴게요.
- function: Function 역할은 Assistant가 호출로 실행할 작업을 정의할 때 사용. 주로 외부의 API와 LLM을 연동하기 위한 목적으로 사용됨

## | Hyper parameters (Chat mode 기준)

---

- temperature: 생성 시 다양성을 조절하는 매개변수 (0~1)
- max\_tokens: 생성된 텍스트의 최대 토큰 수
- stop: 종료 토큰(End-of-Sequence)을 나타내는 토큰의 리스트
- top\_p: 다음 단어 선택 시 고려되는 확률 분포의 크기를 지정하는 매개변수 (0.1~1) 예) 0.8 -> 상위 80%의 확률을 가진 token 중 다음 단어를 선택
- frequency\_penalty: 단어 반복에 대해서 패널티를 주어서 반복하지 않게 하는 매개변수
- presence\_penalty: 새 토큰에 대한 출현빈도를 낮추기 위한 매개변수

```
gen = openai.chat.completions.create(  
    model="gpt-4o-mini",  
    messages=  
    ...  
    stream=True  
)
```

```
while True:  
    response = next(gen)  
    delta = response.choices[0].delta  
    if delta.content is not None:  
        print(delta.content, end="")  
    else:  
        break
```

- 프롬프트 엔지니어링은 다양한 애플리케이션과 연구 주제에 언어모델(LM)을 효율적으로 사용할 수 있도록 프롬프트를 개발하고 최적화하는 비교적 새로운 분야
- 연구자는 프롬프트 엔지니어링을 사용하여 질의응답 및 산술 추론과 같은 일반적 작업부터 복잡한 작업까지 다양한 범위에서 LLM의 역량을 향상
- 프롬프트 엔지니어링은 LLM과 상호 작용하고 개발하는 데 유용한 다양한 기술과 기법을 포함
- LLM과 인터페이스를 형성하고, 빌드하고, LLM의 기능을 이해하는 데 중요한 기술
- LLM에 따라 프롬프트의 유효성은 차이가 있음: 대부분의 프롬프트 엔지니어링은 ChatGPT를 기준으로 설명 하지만 llama나 다른 LLM 들에는 잘 적용되지 않을 수 있음. 작은 모델일수록 상세한 프롬프트를 잘 이해하지 못하는 경향

- LLM이 최종 답변을 내리기 전에 중간 추론 단계를 명시적으로 보여주는 프롬프팅 기법
- 복잡한 문제 해결력 향상, 추론 과정의 투명성 확보 등의 장점
- "Let's solve this step by step" 같은 명시적 지시어 사용, 중간 사고 과정 표현
- 수학 문제 해결, 논리 추론, 의사결정, 코딩 문제 등에 사용
- Zero-shot CoT, Few-shot CoT, Self-consistency CoT 등 다양한 변형 기법 개발
- 최근에는 LLM 자체를 CoT 형식의 데이터로 학습한 모델들도 나오고 있으며, o1-mini와 o1-preview가 대표적인 모델임. 이런 모델을 추론(reasoning) 모델이라고도 함
- CoT에 맞게 학습된 o1계열 모델들은 CoT 프롬프트를 쓰지 않는 편이 오히려 좋은 결과가 나올 가능성을 높임

sharebook-kr/pykrx: KRX 주식 정보 스크래핑

github.com/sharebook-kr/pykrx

Type  to search

Code Issues Pull requests Actions Projects Security Insights

pykrx Public Watch 21 Fork 200 Star 546

master 6 branches 0 tags Go to file Add file Code

mr-yoo 수정 주가 조회시 Naver에서 데이터를 가져오도록 원복함 (#162) ... 2ef0fe2 on Aug 5 144 commits

pykrx 수정 주가 조회시 Naver에서 데이터를 가져오도록 원복함 (#162) last month

tests ETF의 get\_trading\_volume\_and\_value 함수의 수정 (#132) 4 months ago

.gitignore add gitignore for python distribution files last year

readme.MD ETF의 get\_trading\_volume\_and\_value 함수의 수정 (#132) 4 months ago

requirements.txt ETF의 get\_trading\_volume\_and\_value 함수의 수정 (#132) 4 months ago

setup.py pypi에 TTF 파일 추가 2 months ago

readme.MD

PyKrx

About

KRX 주식 정보 스크래핑

Readme

Activity

546 stars

21 watching

200 forks

Report repository

Releases

No releases published

Packages

 이 누리집은 대한민국 공식 전자정부 누리집입니다.

검색어를 입력해 주세요.



로그인 회원가입 사이트맵 ENGLISH

**DATA** . GO . KR

데이터찾기

국가데이터맵

데이터요청

데이터활용

정보공유

이용안내

## 오픈API 상세

[f](#) [t](#) [w](#) URL 복사

XML

JSON 한국관광공사\_고캠핑 정보 조회서비스\_GW

활용신청

한국관광공사 고캠핑 홈페이지에서 제공하는 캠핑장 정보를 제공합니다.  
한국관광공사 고캠핑 홈페이지에서 제공하는 캠핑장 정보를 제공합니다.



7



2



관심

OpenAPI 정보

메타데이터 다운로드

▲ 오픈API 에러코드

데이터 개선요청

오류신고 및 문의

분류체계	문화체육관광 - 관광	제공기관	<a href="#">한국관광공사</a>
관리부서명	디지털인프라팀	관리부서 전화번호	033-371-2733
API 유형	REST	데이터포맷	JSON+XML
활용신청	800	키워드	야영장, 카라반, 글램핑
등록	2022-06-30	수정	2022-12-13
비용부과유무	무료	신청가능 트래픽	개발계정 : 1,000 / 운영계정 : 활용사례 등록시 신청하면 트래픽 증가 가능
심의유형	개발단계 : 자동승인 / 운영단계 : 심의승인		

- Gradio: 사용자 친화적인 인터페이스를 통해 AI 모델 배포와 상호 작용을 단순화하는 Python 라이브러리
- 간편한 모델 배포: Gradio를 사용하면 딥러닝 및 기계 학습 모델을 간편하게 웹 애플리케이션으로 배포할 수 있음
- 다양한 입력 형식: 텍스트, 이미지, 오디오, 비디오 등 다양한 입력 형식을 모델에 제공할 수 있으며, 모델의 예측 결과를 즉시 확인 가능
- 웹 기반 사용자 인터페이스: 사용자 친화적인 웹 기반 인터페이스를 통해 모델에 입력을 제공하고 결과를 시각적으로 확인할 수 있음
- 간단한 코드 통합: 몇 줄의 코드로 Gradio를 사용하여 모델을 빠르게 웹 서비스로 변환할 수 있음
- 확장성: Gradio는 다양한 환경에서 사용할 수 있으며, 클라우드, 로컬 서버, 도커 컨테이너 등에서 확장 가능

- gr.interace는 Gradio 인터페이스와 python function을 연동
- Interface Class는 세가지 파라메터를 가짐
  - fn: UI에 적용되는 함수
  - inputs: 입력으로 사용되는 컴포넌트(복수개 가능)
  - outputs: 출력으로 사용되는 컴포넌트(복수개 가능)
- fn(inputs) 함수가 리턴한 결과는 outputs에 반영된다, 라는 매우 간단한 원칙이 있으며, 이 원칙은 더 복잡한 인터페이스, 복잡한 함수에도 동일하게 적용됨

- Jupyter notebook의 셀 내에서 실행 가능: 실행할 때마다 포트를 바꿔가면서 어플리케이션 구동
- python3 app.py로 어플리케이션 구동
- gradio app.py로 구동: 'python' 어플리케이션을 사용하므로 다음과 같이 symbolic link를 걸어둠

```
ln -sf /usr/bin/python3 /usr/bin/python
```

- gradio로 실행하면 소스의 변경을 감지하여 자동으로 반영해줌: 어플리케이션을 중단하고 다시 실행해야 하는 수고가 없음

- Streamlit은 파이썬 기반의 오픈 소스 라이브러리로, 데이터 과학자 및 개발자들이 빠르게 웹 기반의 대화형 데이터 애플리케이션을 만들 수 있게 도와주는 역할
- Gradio 보다 코드는 좀더 길어지나, 인터페이스에 발생하는 이벤트 처리 등을 더 섬세하게 처리할 수 있다는 장점이 있음
- Streamlit은 프론트엔드와 백엔드로 나누어져 있으며, 프론트엔드에서 뭔가 변경이 발생하게 되면(예: 사용자가 라디오 버튼을 선택, 텍스트를 입력) 백엔드의 코드는 처음부터 한번 '다시' 실행됨. 이 작업을 매번 반복
- 매번 백엔드 코드들이 실행될 때 기존의 데이터들을 잃지 않기 위해 'session'이라는 객체를 사용하는데, 이 객체에 Streamlit 프론트엔드 컴포넌트들의 상태도 저장되어 있음
- 따라서 개발자는 UI를 구성하고, 백엔드의 python 코드는 항상 처음부터 다시 실행된다고 가정하여 session으로부터 데이터를 다시 읽는 식으로 구성해야 함
- 한번만 초기화해야 하는 객체는 `st.cache_data`나 `st.cache_resource`로 캐쉬에 넣을 수 있음

- 설치

pip install streamlit

- 실행

streamlit run <실행할 python 파일명>

예) streamlit run plotting\_demo.py

- streamlit은 실제로는 ~/.local/bin/streamlit이라는 이름으로 저장되는 python 스크립트로, streamlit.web.cli.main이라는 함수를 실행하도록 되어 있음. 따라서 배포시에도 동일하게 'streamlit run' 을 실행해도 되고, 아니면 이 함수를 사용해도 됨

```
#!/usr/bin/python3

# -*- coding: utf-8 -*-

import re

import sys

from streamlit.web.cli import main


if __name__ == '__main__':

    sys.argv[0] = re.sub(r'(-script\b|.pyw|\.exe)?$', "", sys.argv[0])

    sys.exit(main())
```

streamlit hello # Streamlit 데모를 실행

streamlit run plotting\_demo.py # 기본적인 실행 방법, 웹은 8501 포트로 열림

streamlit run --server.port 8000 plotting\_demo.py # 웹을 8000 포트로 구동

python -m streamlit run plotting\_demo.py # python의 모듈로 실행하는 방식

streamlit run [https://raw.githubusercontent.com/streamlit/demo-uber-nyc-pickups/master/streamlit\\_app.py](https://raw.githubusercontent.com/streamlit/demo-uber-nyc-pickups/master/streamlit_app.py) # 원격지의 코드를 실행하는 방식

Streamlit 설정은 다음 네가지 방식으로 지정 가능

- ~/.streamlit/config.toml 파일에 저장
- 현재경로/.streamlit/config.toml 파일에 저장
- STREAMLIT\_으로 시작하는 환경 변수를 지정

예) export STREAMLIT\_SERVER\_PORT=80

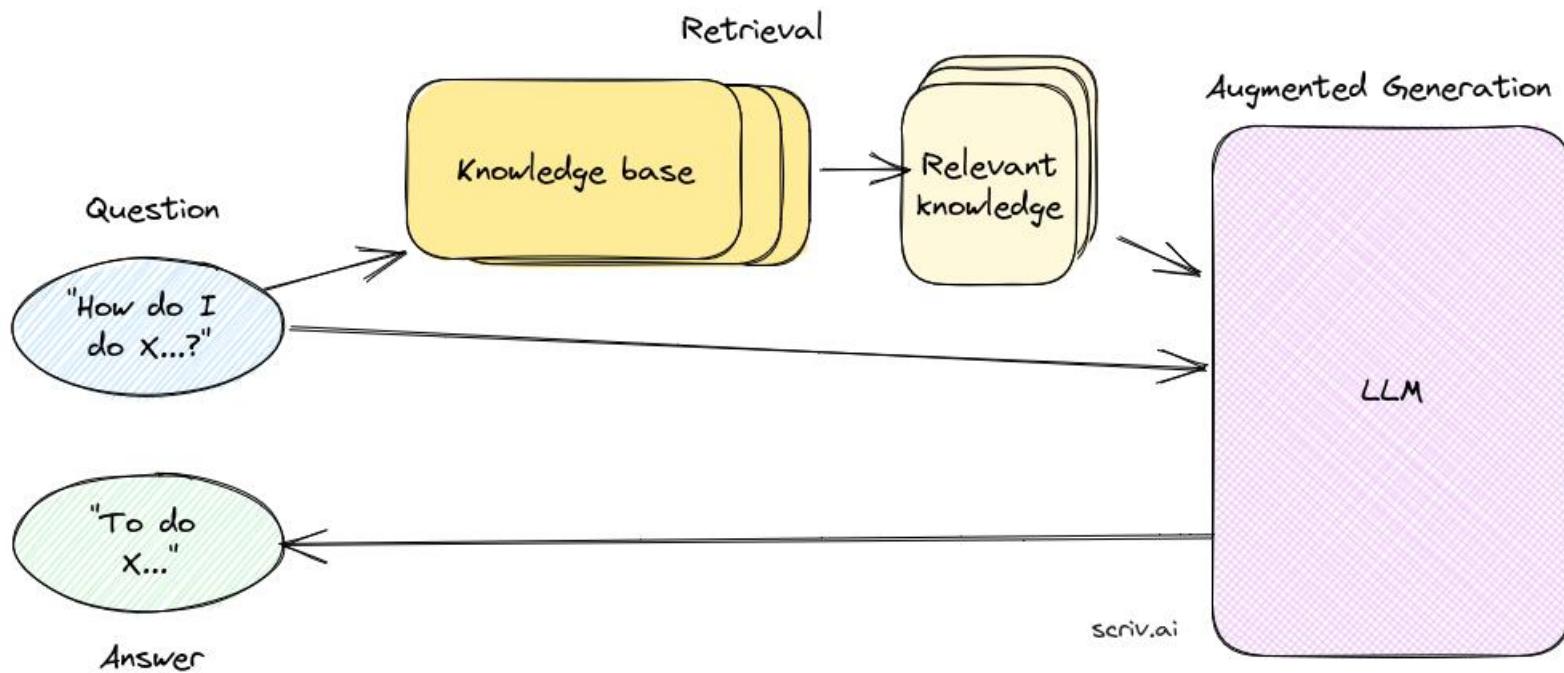
- streamlit run에 옵션으로 지정

<https://docs.streamlit.io/library/advanced-features/configuration>

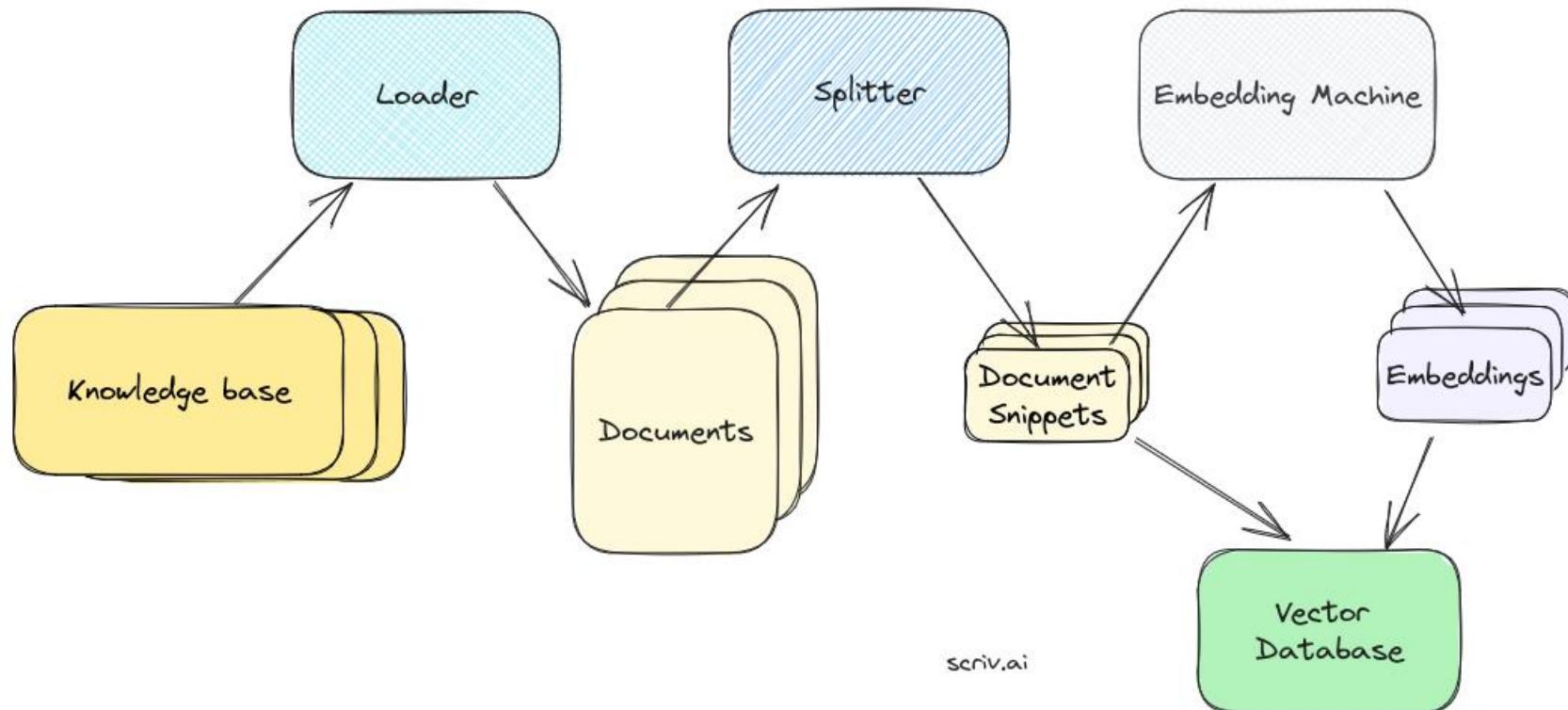
- 언어모델을 기반으로 한 어플리케이션을 개발하기 위한 프레임워크
- 언어모델 학습시 사용된 데이터 외 다른 데이터를 인식할 수 있게 함
- 여러가지 LLM 모델들을 선택적으로 사용할 수 있게 함
- 능동적으로 다른 기능들을 연동하여 추가적인 결과를 낼 수 있게 함
- 여러 기능 모듈을 체인으로 연결하여 파이프를 만들어서 기능적으로 확장할 수 있음
- 내부적으로는 LLM의 프롬프트 조작과 관련 라이브러리셋으로 이런 기능들을 구현

- Model I/O: LLM과 인터페이스 하는 모듈
- Retrieval: 어플리케이션과 연관된 데이터들을 검색하고 찾는 모듈
- Chains: 연속된 LLM 호출과 관계된 모듈
- Agents: 고수준 명령에 의해 적절한 도구를 사용하여 task를 수행하게 하는 모듈
- Memory: 어플리케이션의 상태를 저장하는 것과 관련된 모듈
- Callbacks: 각 체인간의 중간 단계를 저장하고 스트리밍하는 모듈

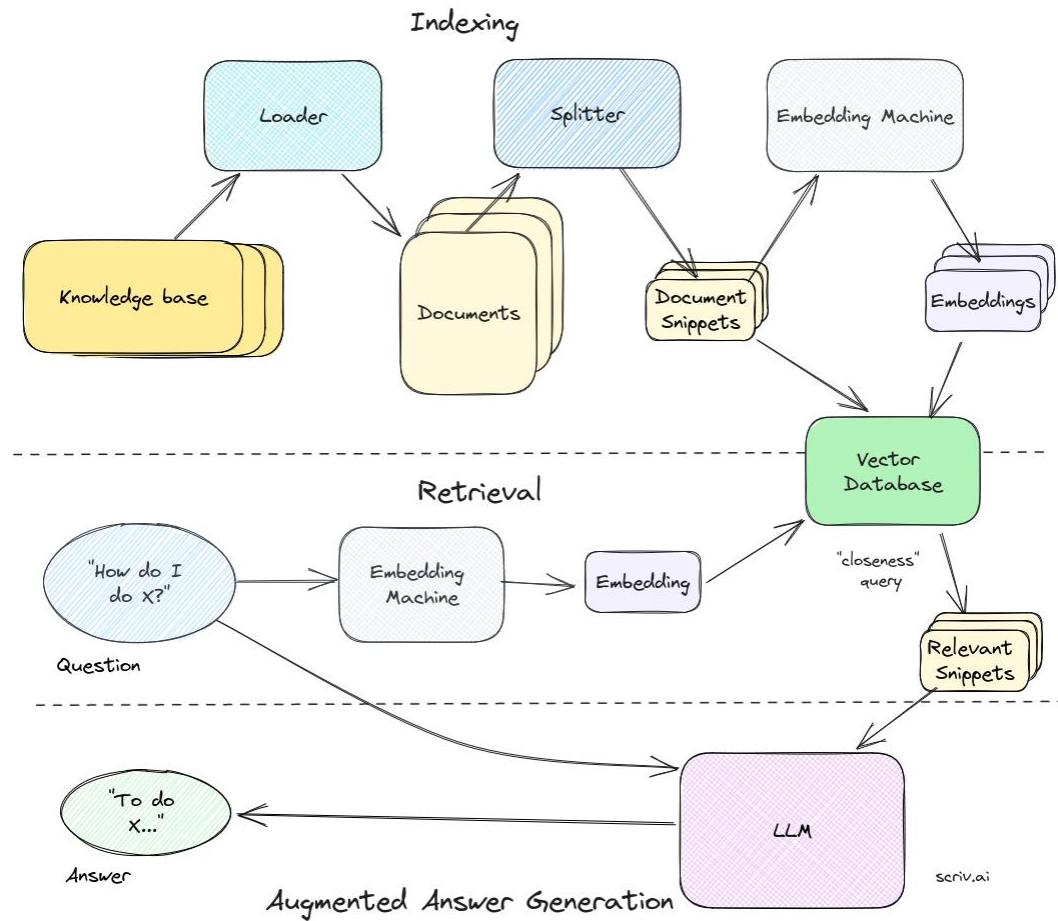
- Embedding은 다양한 데이터를 정해진 차원의 벡터로 변환한 것
- 이러한 벡터들을 저장하고 검색하여 주어진 벡터와 가장 가까운 것들을 추출할 수 있는데, 이렇게 벡터들을 저장하고 관리하는 일종의 Vector Store를 VectorDB라 부름
- 일반적인 데이터베이스에서는, 주로 쿼리와 정확히 일치하는 데이터베이스의 행을 찾는 반면, VectorDB에서는 유사성 메트릭을 적용하여 쿼리(사용자 질문)과 가장 유사한 벡터를 검색
- RAG(Retrieval Augmented Generation)에서 Retriever에 해당하는 것이 이 VectorDB
- 상용으로는 Pinecone이 유명하며, 오픈소스로 Qdrant, Weaviate, Milvus, Chroma, Vespa, FAISS 등이 있음. 어떤 것이나 제공하는 기본 기능은 비슷하지만, 검색의 품질에서는 미묘하게 차이가 있음



## Knowledge Indexing



scriv.ai



Is Langchain really pointless? x +

medium.com/@wp237/is-langchain-really-pointless-2302dea10d6d

Medium Search

Write Sign up Sign in

# Is Langchain really pointless?

Top Piriyakulwij · Follow  
6 min read · Aug 28, 2023

21 Q



# LangChain

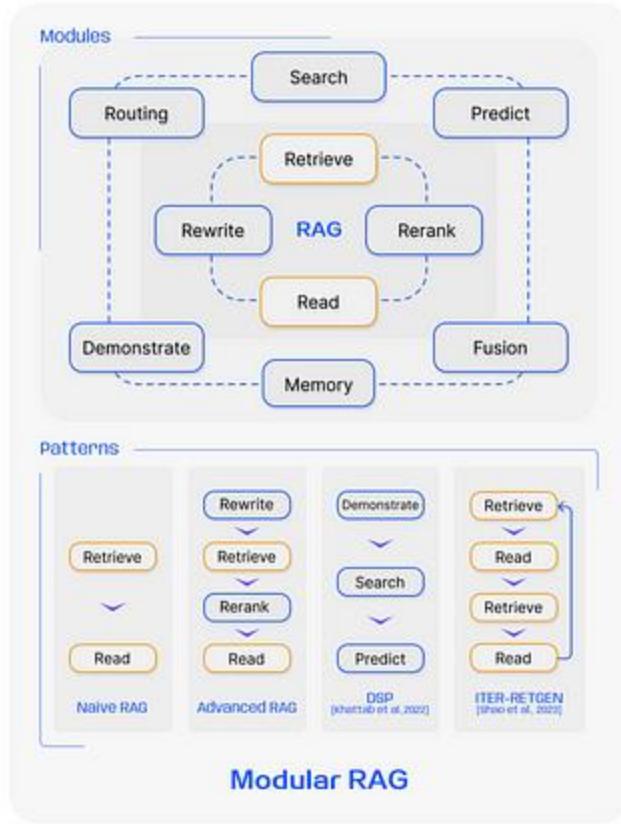
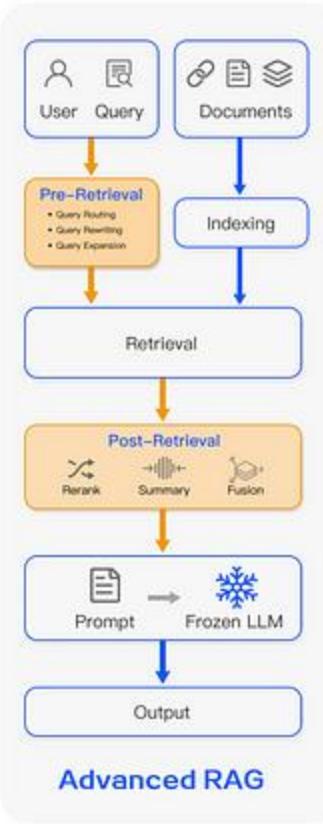
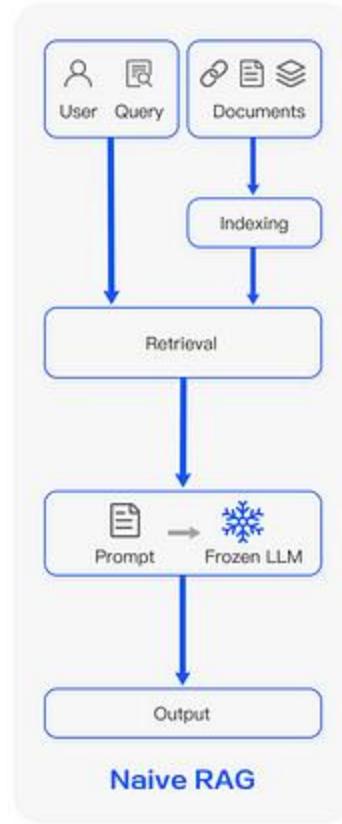
Credit: <https://www.langchain.com/>

In this article, I write about my personal thoughts on LangChain and its usability, in response to recent articles like [Langchain Is Pointless](#) and [The Problem With LangChain](#)

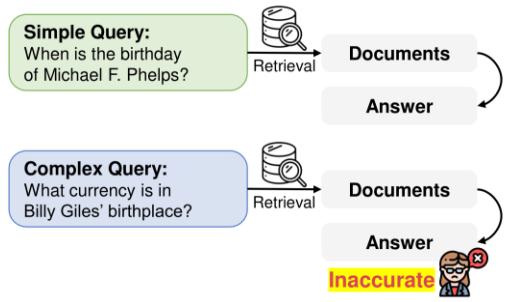
TL;DR - Not pointless for building quick cool demos BUT not worth learning

- LangChain에서 체인을 쉽게 구성하기 위해 도입된 형식
- {object} | {object} | {object} … 하는 식으로 각 객체들을 pipe로 연결하는 문법을 사용
- 자체적으로 Streaming을 지원하며, 병렬실행도 가능

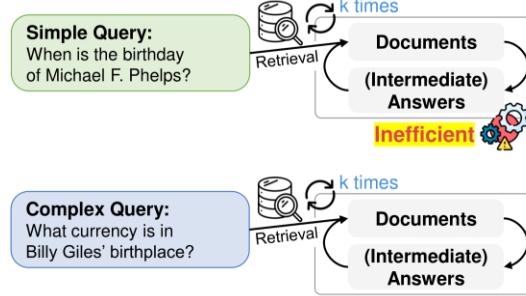
- Naive RAG: 가장 기본적인 구성 방식. Query가 들어오면 문서를 검색하고, 이를 context로 사용하여 응답을 생성. 간단하지만 검색된 문서의 관련성이나 신뢰성을 깊이 평가하지 않는다는 한계
- Advanced RAG: Pre-retrieval, Post-retrieval process를 통해 Retrieve되는 context의 질을 높이는 방식의 RAG. Query를 재작성하거나 Rerank를 사용하는 방법 등을 포함
- Modular RAG: 필요에 따라 여러 개의 모듈을 선택적으로 사용하여 결과물을 생성하는 방식의 RAG
- Adaptive RAG: 3단계의 복잡도에 따라 non-retrieval, single-step, multi-step으로 Retrieval 전략을 사용하여 답변하는 RAG. 복잡도 분류에 별도의 작은 모델을 사용
- Self RAG: 검색(retrieval)과 자기반성(self-reflection)을 통해 언어모델 응답의 품질과 사실성을 개선하는 방식의 RAG. 검색된 문서의 관련성에 따라 query를 재작성하거나 하는 방식을 취함
- Corrective RAG: Retrieval의 결과를 correct, ambiguous, incorrect로 분류하여 각각 다르게 처리하는 RAG. Self RAG에 CRAG을 더한 Self CRAG도 있음



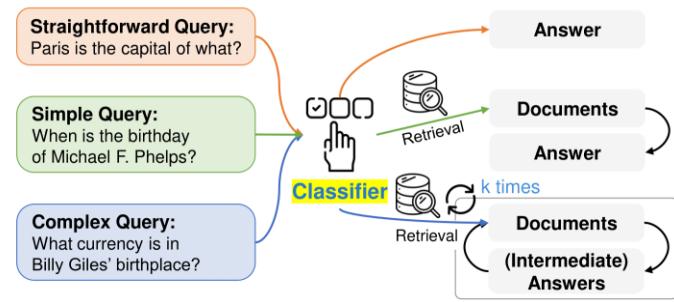
### (A) Single-Step Approach



### (B) Multi-Step Approach



### (C) Our Adaptive Approach



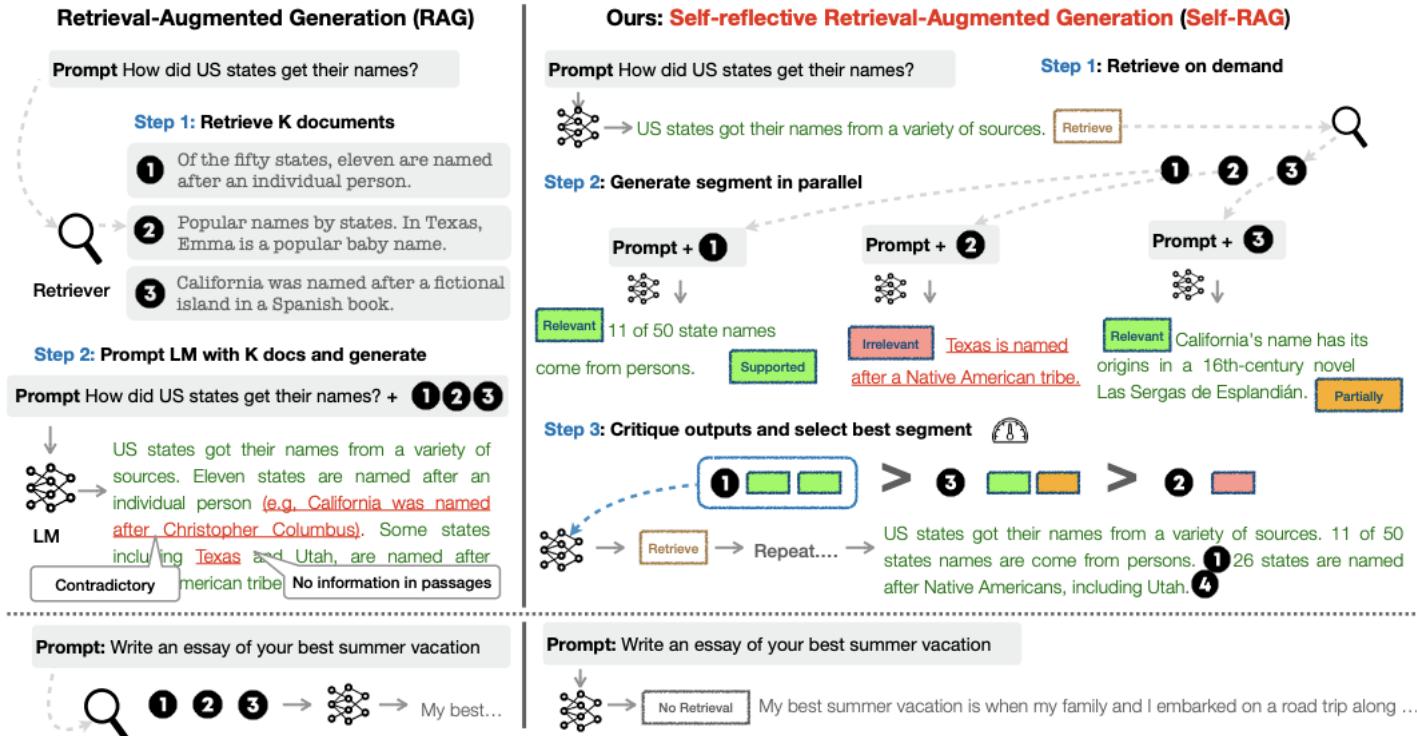


Figure 1: Overview of SELF-RAG. SELF-RAG learns to retrieve, critique, and generate text passages to enhance overall generation quality, factuality, and verifiability.

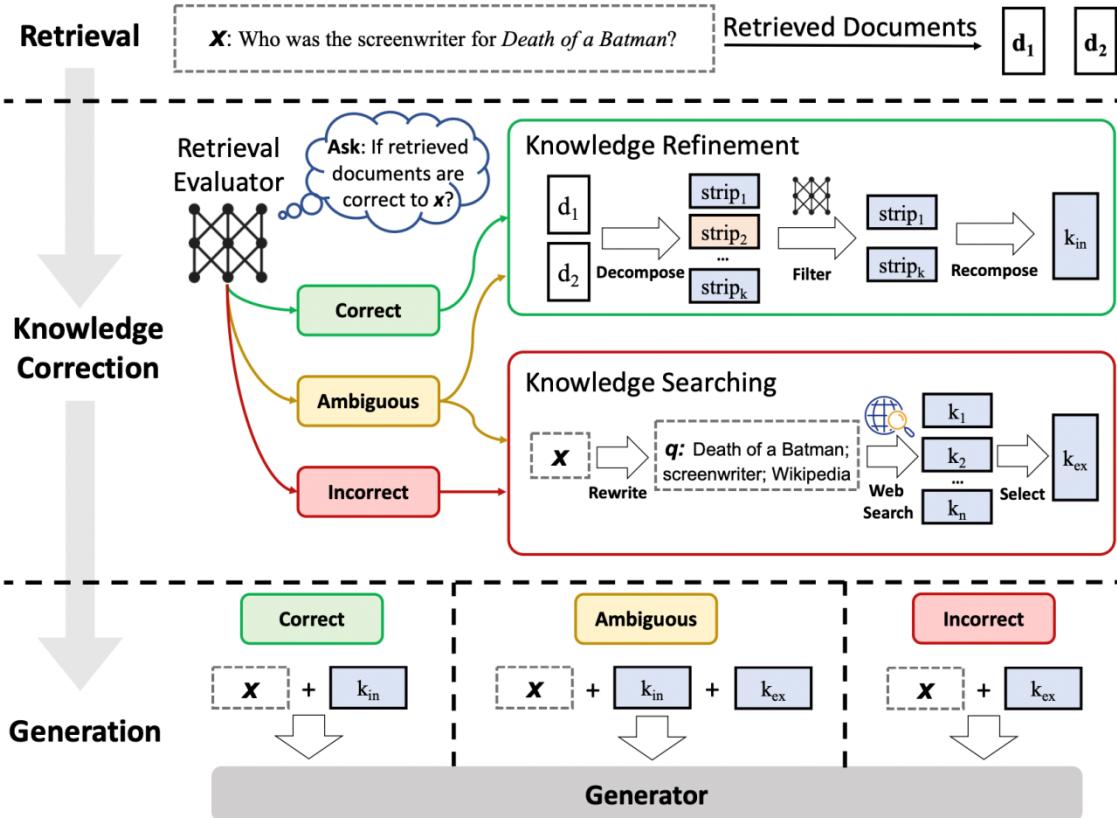
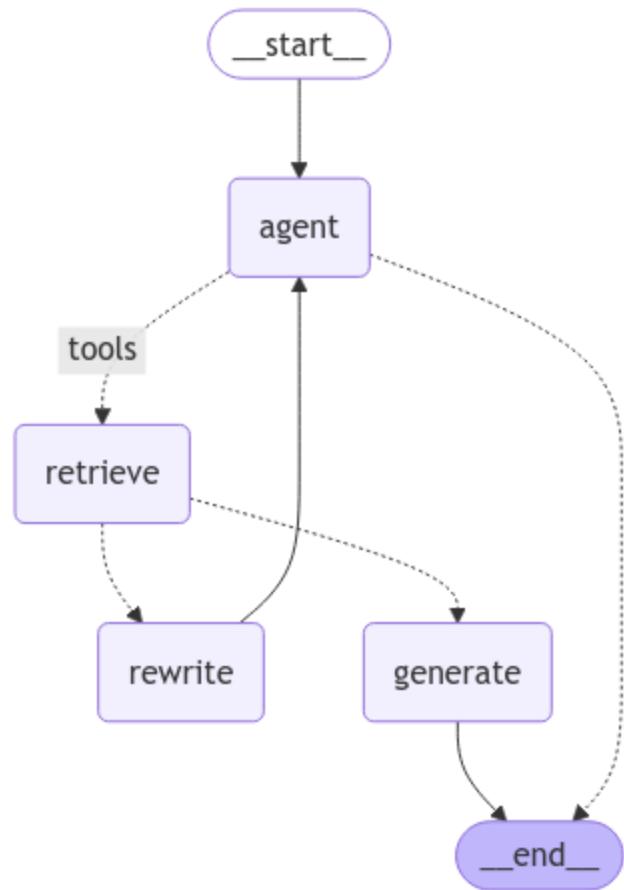
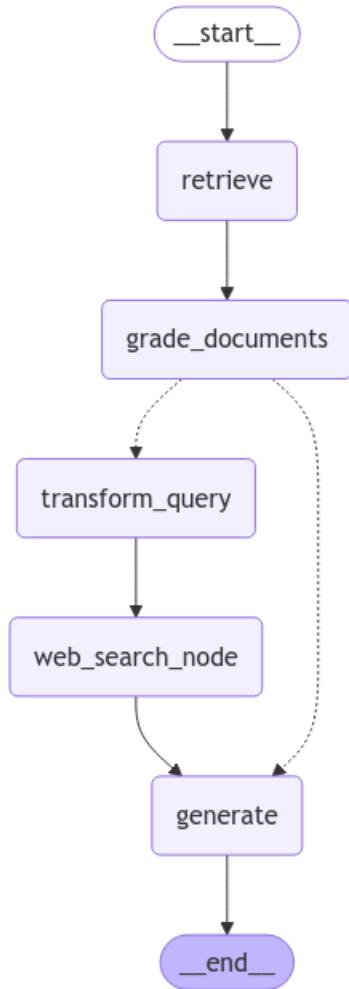


Figure 2: An overview of CRAG at inference. A retrieval evaluator is constructed to evaluate the relevance of the retrieved documents to the input, and estimate a confidence degree based on which different knowledge retrieval actions of {Correct, Incorrect, Ambiguous} can be triggered.

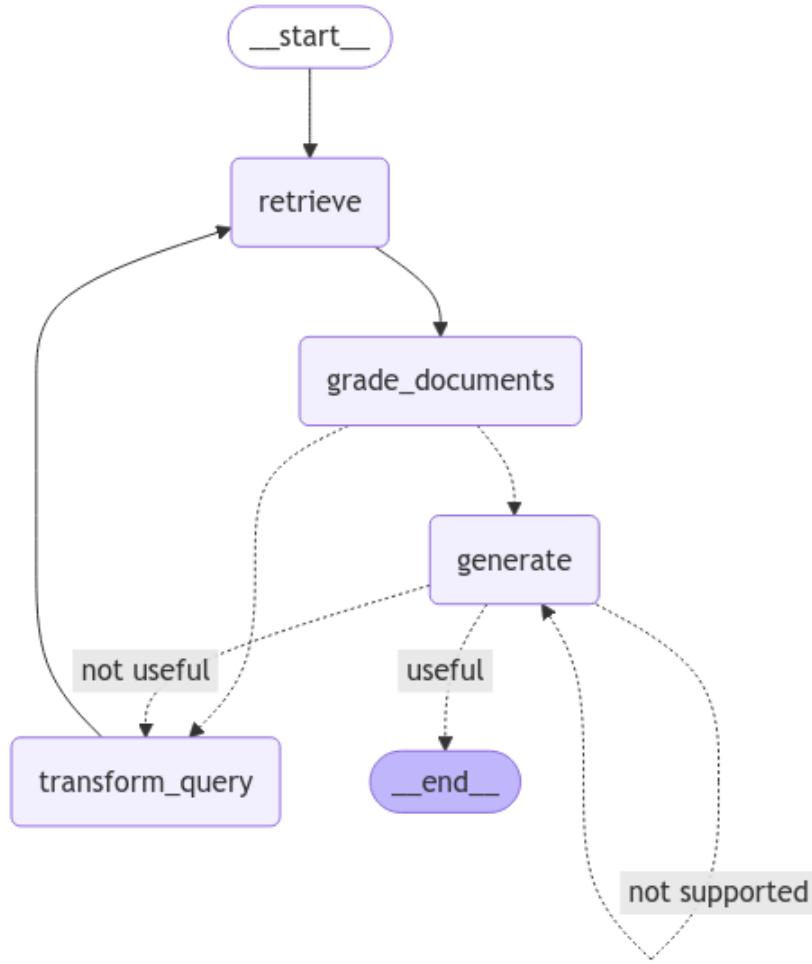
# Agentic RAG



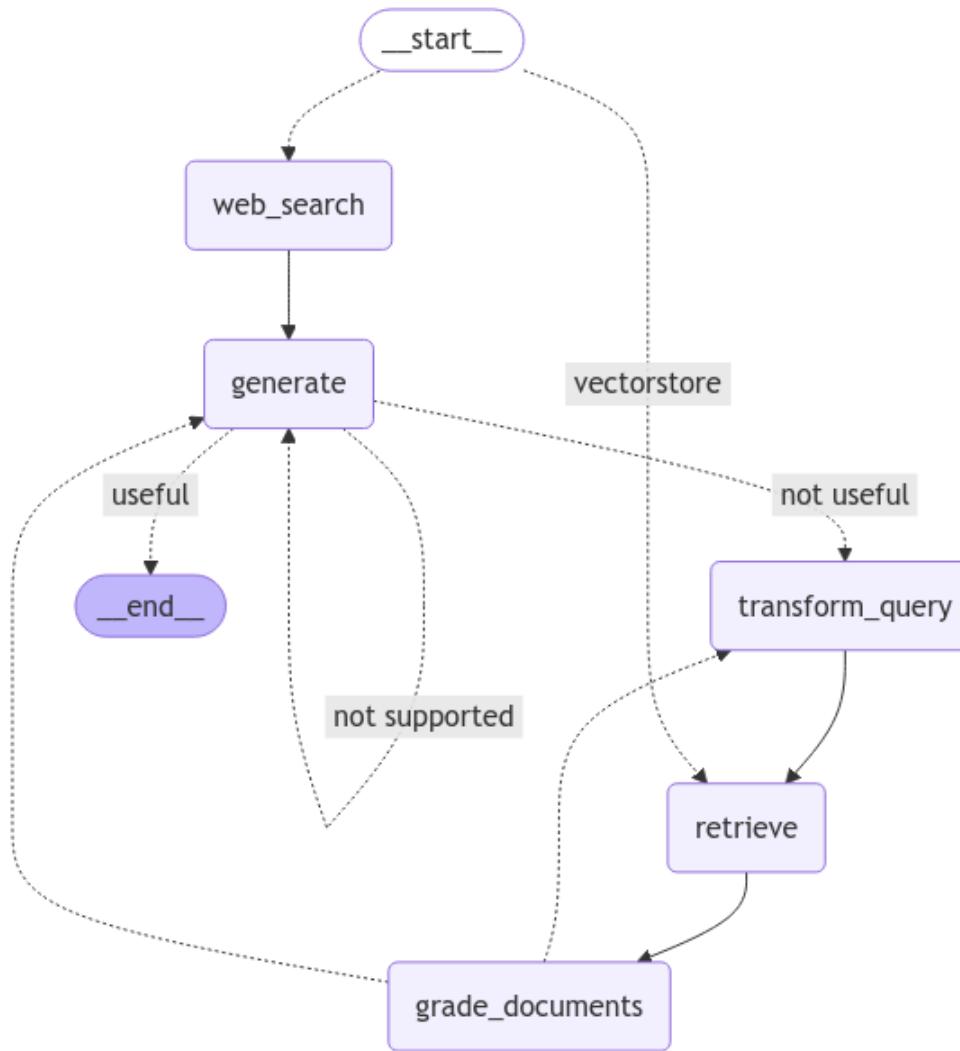
# CRAG



# Self RAG



# Adaptive RAG



- LLM 기반 멀티에이전트 시스템을 위한 프레임워크
- LangChain 생태계의 일부로, LLM 기반 에이전트들의 상태 관리와 워크플로우 조정을 위한 프레임워크
- 상태 머신 기반의 워크플로우 관리, 다중 에이전트 간 협업 조율, 복잡한 태스크의 단계별 실행
- 주요 컴포넌트: State Managers, Channel Managers, Graph Execution Engine, Agent Coordinators
- Python 기반, 비동기 실행 지원, 모듈러 디자인, 확장 가능한 아키텍처
- 복잡한 워크플로우를 간단하게 구현할 수 있어서, Modular RAG이나 Self RAG, CoT, AI agent 등 반복적으로 생성하고 판단해야 하는 어플리케이션에 적합

- LangGraph에서 실행의 단위는 Node로, Node는 표준 python 함수
- START와 END는 시작과 끝을 나타내는 특별한 종류의 Node
- 각 Node들은 State를 입력으로 받고, State를 출력함
- State는 Node들 간에 공유하는 상태 클래스: TypeDict 타입으로, 일반적으로 messages(사용자와 챗봇이 주고 받은 메시지)를 포함하며, 그외 어플리케이션의 종류에 따라 다양한 속성을 추가할 수 있음
- Node 간의 연결은 Edge라고 하며, Node는 Edge가 연결된 방향과 순서에 따라 실행될 수 있음
- 특정 조건을 만족할 때만 연결이 진행되는 Edge를 정의할 수 있는데, 이를 Conditional Edge라 함
- Conditional Edge에서 정의한 함수의 리턴값에 따라 분기하는 Edge가 달라짐
- 각 Node는 모두 특정한 역할을 하며, Node 실행결과에 따라 실행경로를 다르게 하여 복잡한 기능을 수행하게 하는 것이 LangGraph의 요체
- RAG에서 사용시는 BaseModel을 상속한 클래스를 llm.with\_structured\_output() 메소드로 처리하면 LLM의 출력을 클래스의 인스턴스로 바꾸어서 사용할 수 있음

- ChatGPT 기반 어플리케이션을 구성하려면 LLM 외에도 많은 요소들이 필요
- Assistant API는, 이를 내부적으로 구현하여 이러한 어플리케이션들을 빠르게 구축할 수 있게 한 것: 현재 제공되는 도구는 Code Interpreter, Retrieval, Function calling임. 다시 말해 별도의 시스템 구축 없이 API만으로 python code를 실행하고, 특정 문서에 대해 질의하고, 함수를 실행할 수 있게 해줌
- 이와 더불어, 복수 개 사용자가 동시에 접속하는 경우의 세션 관리나, 대화의 메모리 관리, 입력 token이 overflow되지 않게 하는 기능도 기본적으로 갖추고 있음
- 기본적인 테스트는 Assistant Playground에서 모두 해볼 수 있으며, 여기서 만들어진 Assistant도 어플리케이션에서 사용할 수 있음
- 현재는 Streaming을 지원하지 않으며, completed 상태를 확인하기 위해서 polling해야 하고, DALL-E와 브라우징을 tool로 사용할 수 없으며, User Message에 이미지를 추가할 수 없는 등의 제한이 있음. 또한 계정별로 분당 60 req의 rate limit도 걸려 있음(베타 상태)

- Assistant API를 사용하는 순서는 다음과 같음
- 1) Assistant를 만든다.
  - 2) Thread를 만든다. Thread는 한 작업 세션으로, 사용자별로 주어지는 것이 보통이며, 대화 내역은 저장된다.
  - 3) Thread에 Message를 추가한다. 보통 사용자 Role인 'User'로 Message를 추가해준다.
  - 4) 1)의 Assistant id와 2)의 Thread id를 주어서 Assistant를 실행한다. 여기에 추가 Instruction을 줄 수도 있다.
  - 5) 실행된 작업은 queued state가 된다. 이 상태가 completed가 되면 Assistant의 작업이 끝난 것이다.
  - 6) Thread에 추가된 Assistant의 Message를 확인한다.

- Assistant API를 사용하면 기본 요금 외에 추가로 붙는 요금들이 있음
- File Search를 위해 파일을 업로드할 경우, GB당 \$0.10이 하루에 과금됨. 하나의 파일을 여러 개의 Assistant 가 같이 사용한다면, Assistant 수만큼 요금이 늘어남. 파일을 삭제하지 않으면 계속 과금(1G는 무료)
- User Message의 일부로 파일이 추가되었다면 해당 파일에도 동일하게, 각각 GB당 \$0.10이 하루에 청구됨
- 파일의 크기는 512MB를 넘을 수 없으며, token으로 환산하면 200만 token이 됨
- 파일 종류는 일반 텍스트, 개발용 소스, pdf, md, docx, html, jpg, png 등 업무에서 사용되는 많은 종류의 파일 포맷을 지원하고 있음: 파일 종류에 따라 Code Interpreter나 File Search에서의 사용 가능 여부가 달라지니 확인해야 함
- Code Interpreter는 세션당 \$0.03이며, 각 세션은 1시간 동안 활성화 됨. 그리고 이 요금은 1시간 동안 1회만 과금되는 형태

- AI agent란, 인공지능 기술을 활용하여 특정한 작업을 수행하거나 의사결정을 내릴 수 있는 소프트웨어 시스템: 주변 환경을 인식하고 해석할 수 있으며, 그에 따라 자율적으로 행동하고 목표를 달성하기 위해 최선의 행동 계획을 수립
- 단순히 사용자의 질문에 답하는 것에 그치지 않고, 주어진 도구들을 이용하여 스스로 프롬프트를 만들고 추론하면서 주어진 목표를 달성하려고 노력
- AI agent의 핵심은 역시 LLM이지만, 이 LLM에 어떤 식으로 목표를 주고 작동하게 하는가에 대해 최근 연구가 활발함: 단순 챗봇이라면 효용은 떨어짐, AI agent가 실제 사람이 하는 작업의 일부 또는 전부를 수행할 수 있어야 생산성을 끌어올릴 수 있음

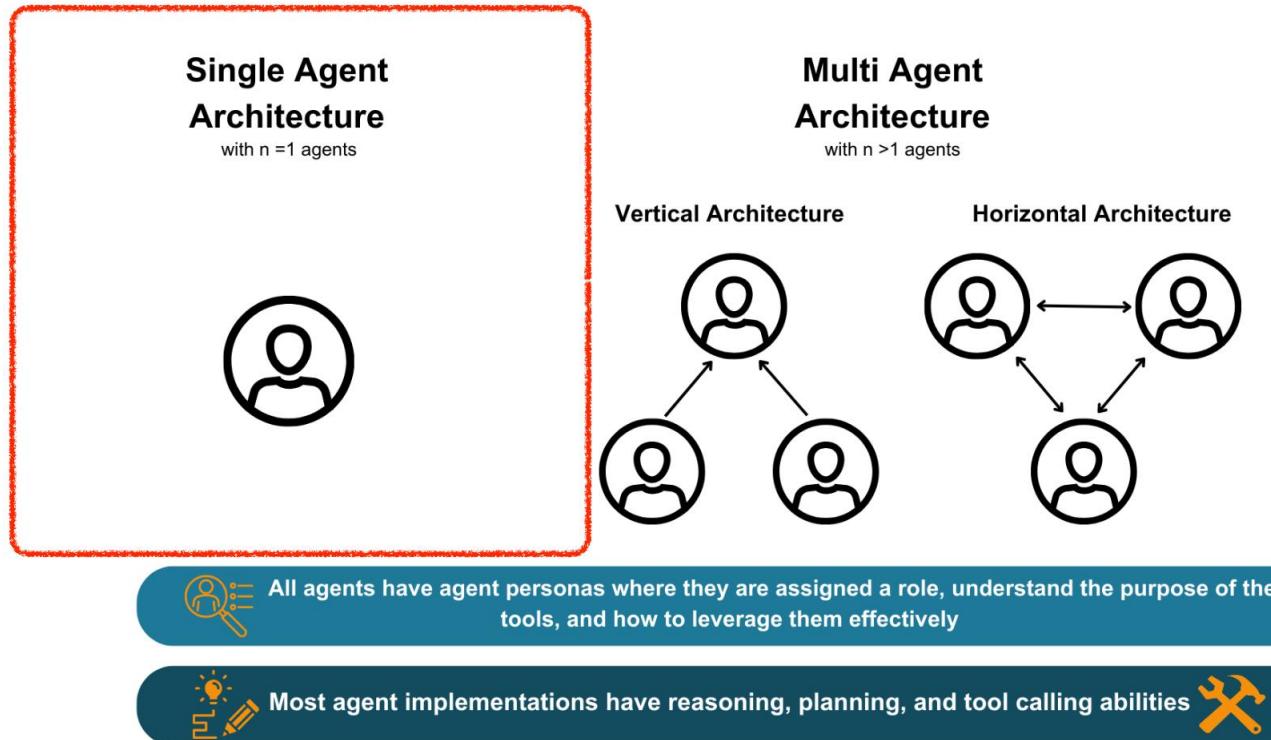
- AI 에이전트(Agent): 목표를 달성하기 위해 여러 번에 걸쳐 반복적으로 계획을 세우고 **행동**을 취할 수 있는 언어 모델(LM, Language Model) 기반의 개체. 각각의 에이전트는 서로 다른 역할을 수행하도록 별도의 페르소나(Persona)가 주어짐. 이 페르소나는 일종의 역할(role)과 같은 것으로, 에이전트에 대한 설명을 비롯하여 접근 가능한 도구들을 정의하는 역할 등을 포함하고 있음
- 단일 에이전트 아키텍처(Single-Agent Architecture): 하나의 언어 모델이 모든 추론(reasoning), 계획(planning), 실행(execution)을 담당하는 구조. 이 경우, 에이전트는 다른 에이전트와의 상호작용 없이 독립적으로 작업을 수행
- 다중 에이전트 아키텍처(Multi-Agent Architecture): 두 개 이상의 에이전트가 협업하여 작업을 수행하는 구조. 각 에이전트는 고유한 페르소나를 가지며, 동일하거나 서로 다른 언어 모델과 도구를 사용할 수 있음. 다중 에이전트 시스템은 다시 수직형(Vertical)과 수평형(Horizontal)으로 나눌 수 있음

- 수직형 아키텍처(Vertical Architecture): 하나의 리드 에이전트(Lead Agent)가 다른 에이전트들을 관리하고 지시하는 구조. 이 경우, 주도 에이전트를 중심으로 의사소통이 이루어짐
- 수평형 아키텍처(Horizontal Architecture): 모든 에이전트가 동등한 지위를 가지며, 그룹 토론을 통해 정보를 공유하고 협업하는 구조
- 추론(Reasoning): 에이전트가 주어진 정보를 바탕으로 논리적인 결론을 도출하거나 문제를 해결하는 과정. 이는 에이전트가 복잡한 상황을 이해하고 적절한 행동을 선택하는 데 필수적인 능력임
- 계획 수립(Planning): 에이전트가 목표를 달성하기 위해 일련의 행동을 순서대로 정리하는 과정. 이는 문제를 더 작은 하위 문제로 분해하고, 각 단계에서 필요한 행동을 결정하는 것을 포함
- 도구 사용(Tool Use): 에이전트가 외부 데이터 소스, API, 기능 등과 상호작용할 수 있는 능력. 도구 사용을 통해 에이전트는 더 많은 정보를 수집하고 활용할 수 있으며, 실제 환경에 변화를 일으킬 수 있음

- 언어 모델(Language Model, LM): 자연어 데이터를 기반으로 학습된 AI 모델로, 인간의 언어 사용 패턴을 모방할 수 있음. 언어 모델은 텍스트 생성, 질의응답, 번역 등 다양한 자연어 처리(NLP, Natural Language Processing) 작업에 활용됨
- 강화 학습(Reinforcement Learning, RL): 에이전트가 환경과 상호작용하면서 시행착오를 통해 학습하는 방법. 에이전트는 행동(action)을 선택하고, 그에 따른 보상(reward)을 받는데, 이 과정을 반복하면서 에이전트는 장기적인 관점에서 최적의 행동 정책(policy)을 학습하게 됨. 강화 학습은 게임 플레이, 로봇 제어 등 다양한 분야에서 활용됨
- 벤치마크(Benchmark): AI 모델이나 에이전트의 성능을 평가하기 위한 표준화된 테스트 또는 데이터셋. 벤치마크를 통해 서로 다른 모델 간의 성능을 객관적으로 비교할 수 있으며, 연구의 발전 방향을 제시할 수 있음. 대표적인 벤치마크는 ImageNet(이미지 분류), SQuAD(질의응답), GLUE(자연어 이해) 등

- 에이전트가 실제 문제를 효과적으로 해결하기 위해서는 추론(reasoning), 계획 수립(planning), 그리고 도구 사용(tool use) 능력이 필수적
- 추론은 인간의 인지 능력 중 가장 기본적이면서도 핵심적인 부분
- 계획 수립은 추론 능력과 밀접히 연관: 에이전트는 주어진 목표를 달성하기 위해 효과적인 행동 시퀀스를 생성해야 하는데, 이는 복잡한 추론 과정을 필요. 일반적으로 계획 수립 과정은 아래 다섯가지 중 하나를 선택
  1. 작업 분해(Task Decomposition): 주어진 문제를 더 작고 다루기 쉬운 하위 문제로 나누는 방법. 에이전트는 각 하위 문제를 순차적으로 해결함으로써 전체 목표에 도달할 수 있음
  2. 다중 계획 선택(Multi-Plan Selection): 에이전트가 여러 개의 대안적인 계획을 생성한 뒤, 그 중 가장 적합한 계획을 선택하는 방법. 이를 통해 에이전트는 다양한 가능성을 고려하고, 상황에 맞는 최적의 계획을 수립할 수 있음

3. 외부 모듈 기반 계획 수립(External Module-Aided Planning): 에이전트가 미리 정의된 외부 모듈이나 알고리즘을 활용하여 계획을 생성하는 방법. 복잡한 계획 수립 과정을 단순화할 수 있지만, 모듈의 성능과 범용성에 크게 의존하게 됨
4. 반영과 개선(Reflection and Refinement): 에이전트가 수립한 계획을 실행하면서 얻은 경험과 피드백을 바탕으로 계획을 지속적으로 개선하는 방법. 이는 에이전트가 환경의 변화에 적응하고, 더 나은 성능을 달성하는 데 도움이 됨
5. 메모리 증강 계획 수립(Memory-Augmented Planning): 에이전트가 과거의 경험이나 지식을 활용하여 계획을 수립하는 방법. 이를 위해서는 에이전트가 중요한 정보를 효과적으로 저장하고 검색할 수 있는 메모리 메커니즘을 갖추어야 함



Source: Neudesic, an IBM Company

Figure 1: A visualization of single and multi-agent architectures with their underlying features and abilities

- ReAct(Reason + Act) 는 "이성적으로 사고한 다음 행동한다(Reason + Act)"는 컨셉의 방식으로, AI 에이전트가 작업에 대해 무엇인가를 수행하기 전, 먼저 '생각'을 하고 그 다음에 '행동'하도록 설계한 구조: LangChain의 경우 Thought/Action/Action Input/Observation을 반복하면서 최종 답안을 얻을 때까지 실행을 계속함
- RAISE 방식은 ReAct를 기반으로 하면서 인간의 단기 및 장기 기억과 유사한 메모리 구성 요소를 추가한 방식
- Reflexion은 언어적 피드백을 통한 자기 성찰(self-reflection)을 사용하는 단일 에이전트 패턴
- AutoGPT+P(Planning) 는 기존의 AutoGPT 방식에 계획(Planning) 기능을 추가하여 자연어를 이해하고 로봇 작업을 수행하는 AI 에이전트의 능력을 강화한 방식
- LATS(Language Agent Tree Search, 언어 에이전트 트리 탐색) 는 계획 수립 및 행동, 추론을 통합하는 트리(Tree) 검색 기법을 적용한 것이 특징인 단일 에이전트 동작 방식

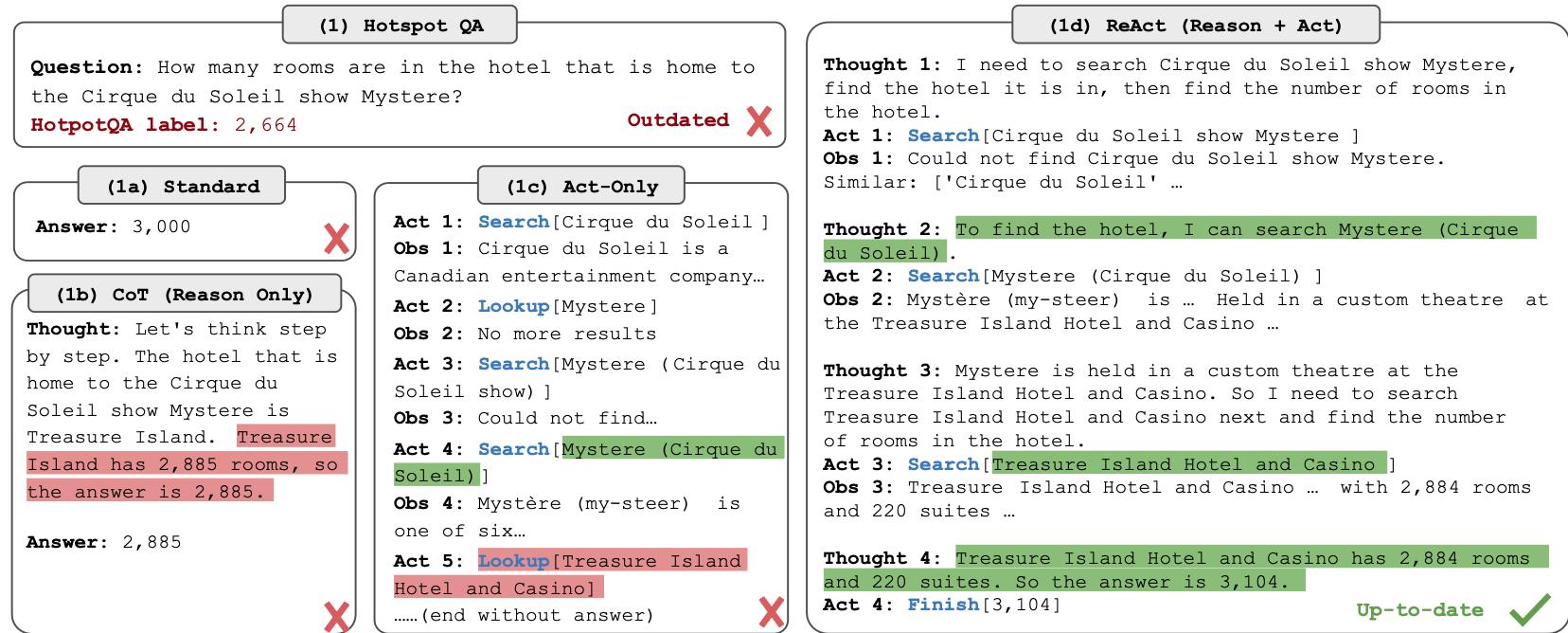


Figure 2: An example of the ReAct method compared to other methods [32]

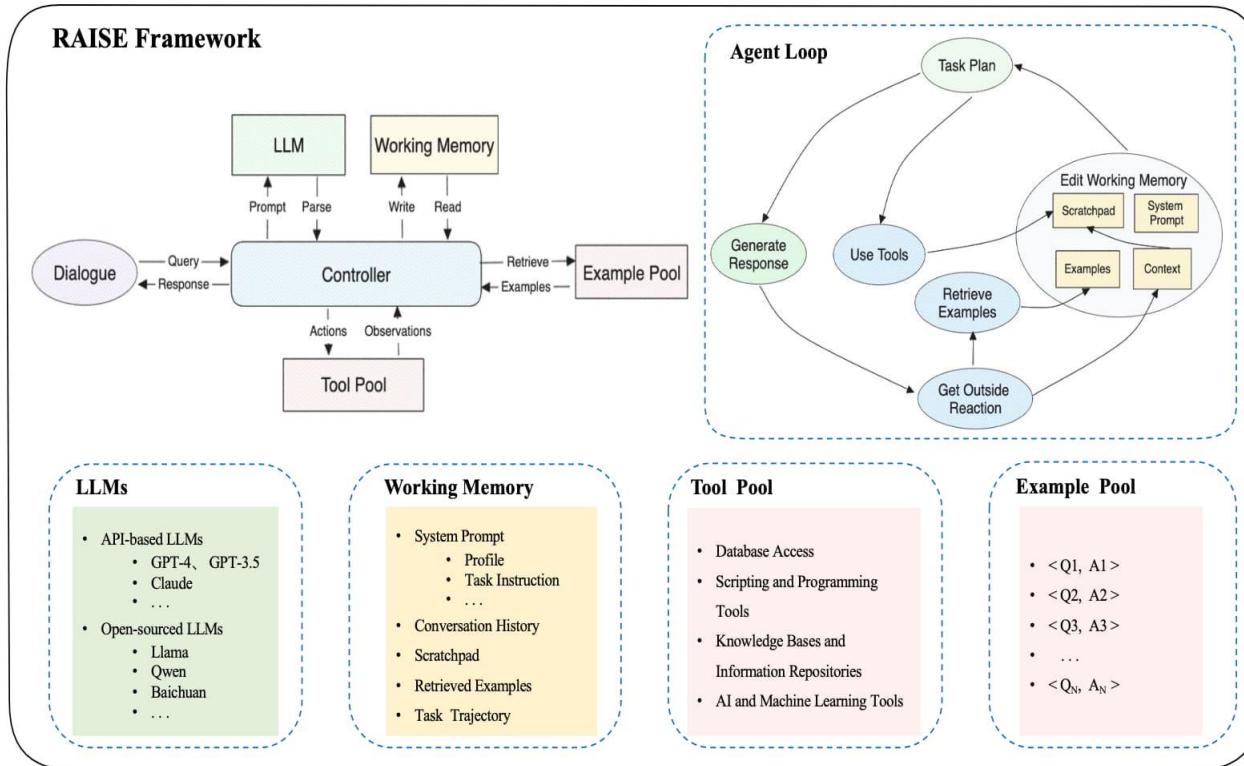
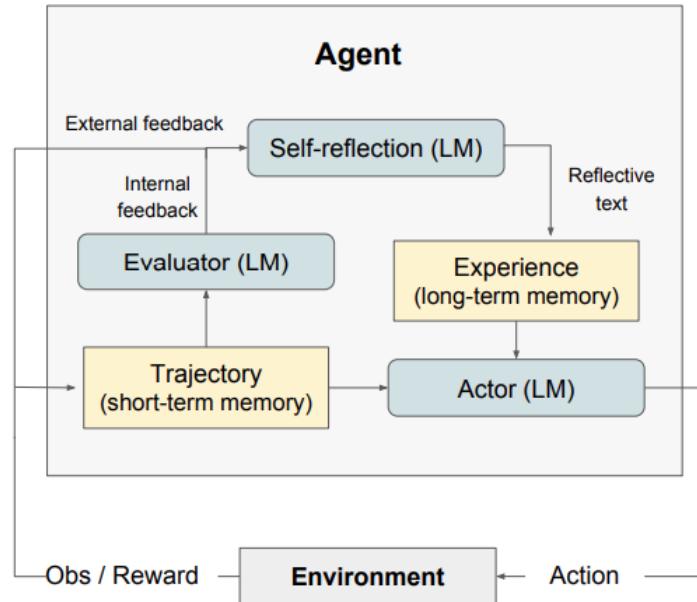


Figure 3: A diagram showing the RAISE method [16]




---

### Algorithm 1 Reinforcement via self-reflection

```

Initialize Actor, Evaluator, Self-Reflection:  

 $M_a, M_e, M_{sr}$   

Initialize policy  $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$   

Generate initial trajectory using  $\pi_\theta$   

Evaluate  $\tau_0$  using  $M_e$   

Generate initial self-reflection  $sr_0$  using  $M_{sr}$   

Set  $mem \leftarrow [sr_0]$   

Set  $t = 0$   

while  $M_e$  not pass or  $t < \text{max trials}$  do  

  Generate  $\tau_t = [a_0, o_0, \dots a_i, o_i]$  using  $\pi_\theta$   

  Evaluate  $\tau_t$  using  $M_e$   

  Generate self-reflection  $sr_t$  using  $M_{sr}$   

  Append  $sr_t$  to  $mem$   

  Increment  $t$   

end while  

return

```

---

Figure 2: (a) Diagram of Reflexion. (b) Reflexion reinforcement algorithm

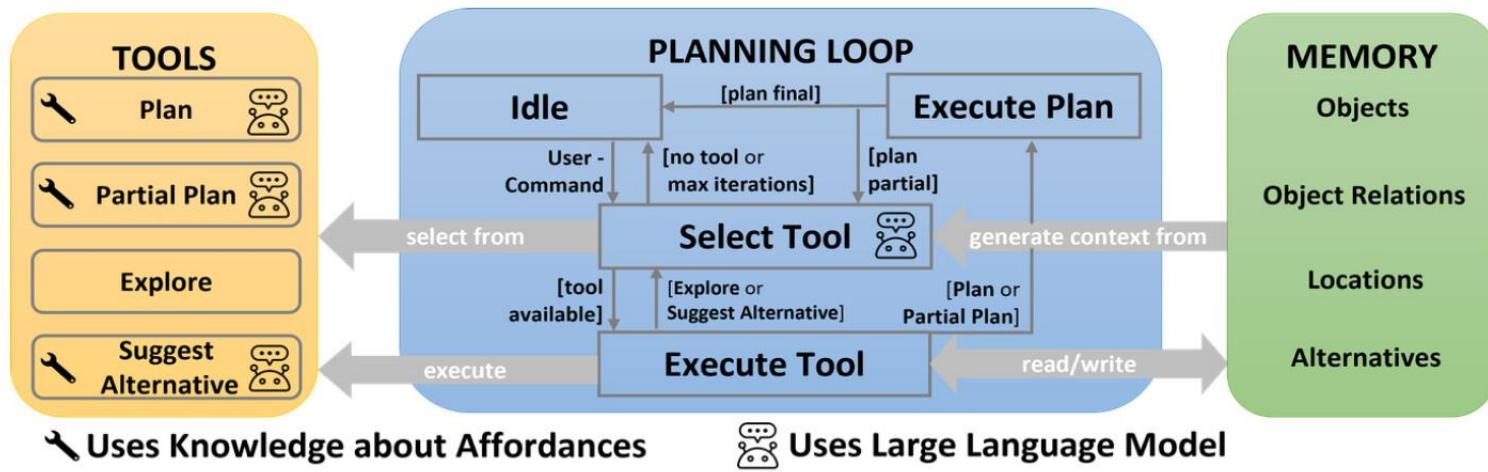
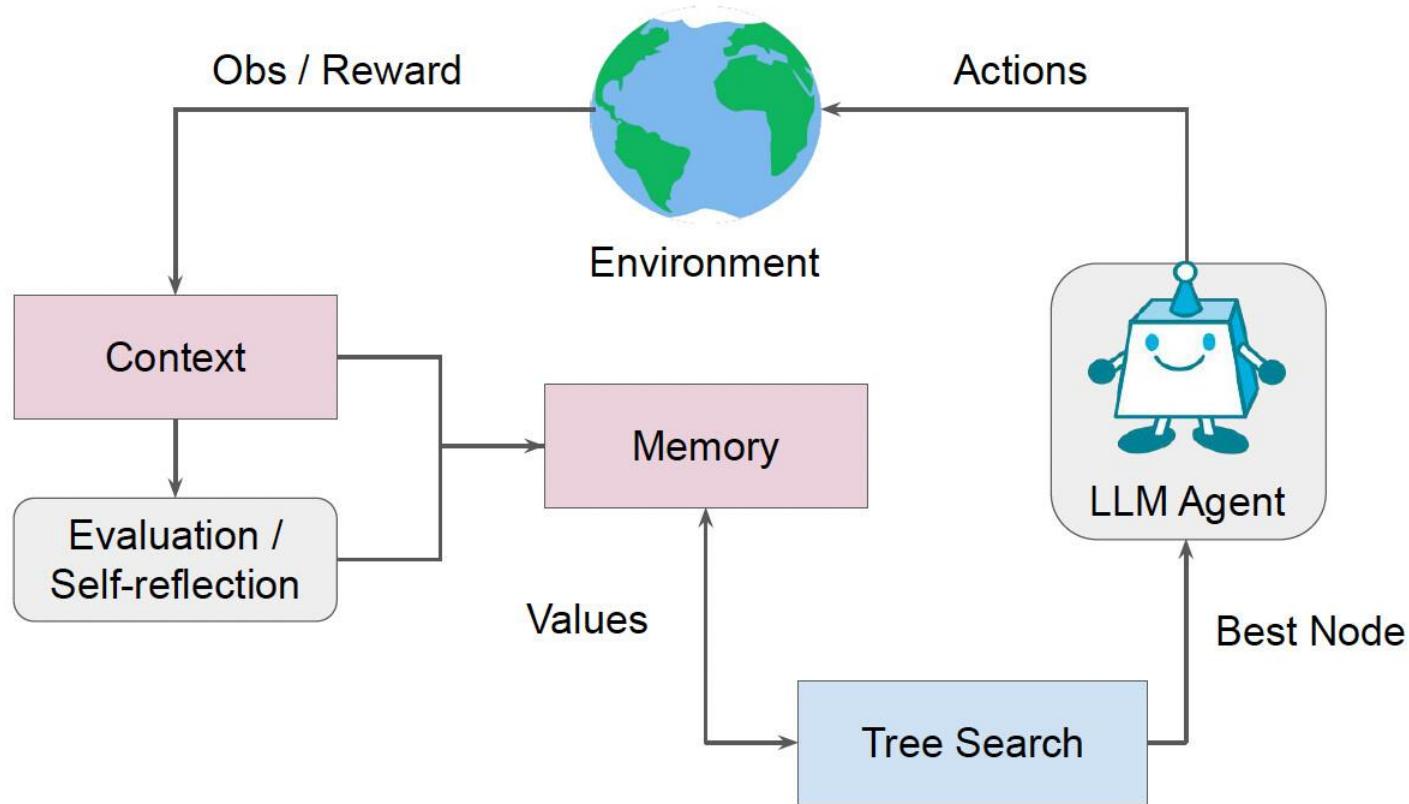


Figure 4: A diagram of the AutoGPT+P method [1]



## Single Agent Architecture

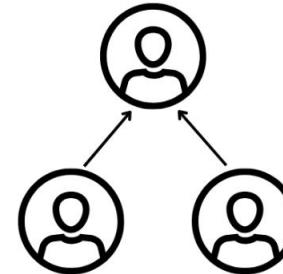
with n = 1 agents



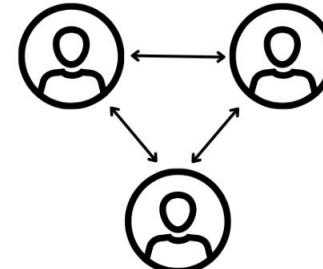
## Multi Agent Architecture

with n > 1 agents

### Vertical Architecture



### Horizontal Architecture



All agents have agent personas where they are assigned a role, understand the purpose of their tools, and how to leverage them effectively



Most agent implementations have reasoning, planning, and tool calling abilities



Source: Neudesic, an IBM Company

Figure 1: A visualization of single and multi-agent architectures with their underlying features and abilities

- 실체화된 LLM 에이전트들의 조직화된 팀 협력 학습(Embodied LLM Agents Learn to Cooperate in Organized Teams)
- DyLAN(Dynamic LLM-Agent Network, 동적 LLM 에이전트 네트워크) 은 복잡한 작업 수행에 특화된 프레임워크로, 동적인 상황에 대응할 수 있는 LLM 기반의 다중 에이전트 네트워크
- AgentVerse 는 각각의 AI 에이전트에게 명확한 역할을 부여하고, 임무 수행을 위한 단계별 계획을 제공하는 방식으로 추론 및 문제 해결 능력을 향상하는 다중 에이전트 아키텍처 방식 중 하나
- MetaGPT 는 다수의 AI 에이전트들이 서로 의사소통을 하며 공통의 문제를 해결할 수 있는 기능을 제공하는 다중 에이전트 아키텍처 방식

# | 실체화된 LLM 에이전트들의 조직화된 팀 협력 학습

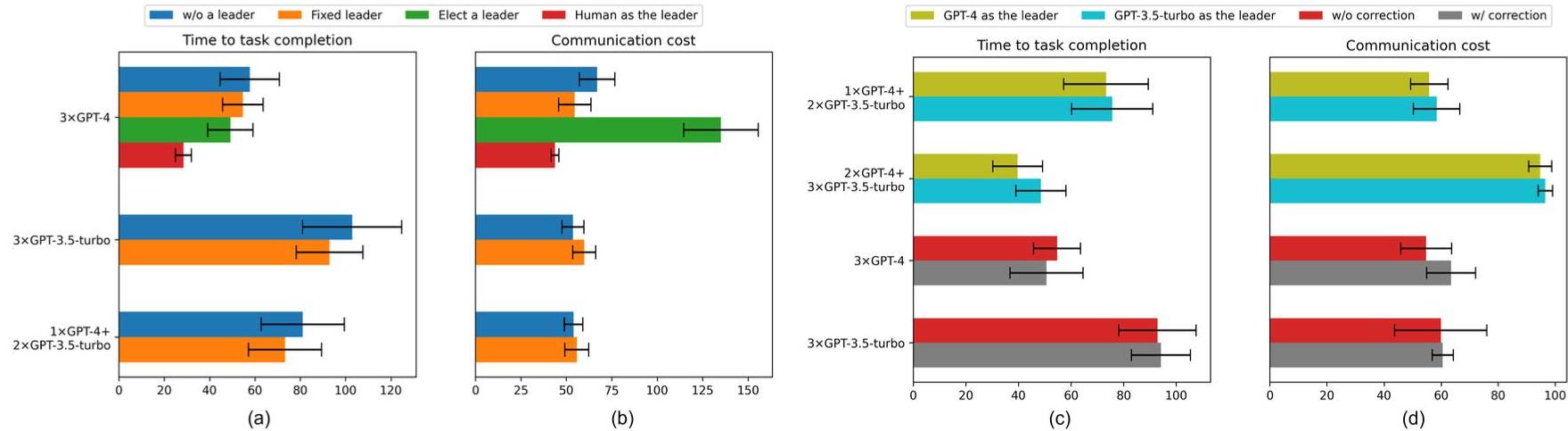
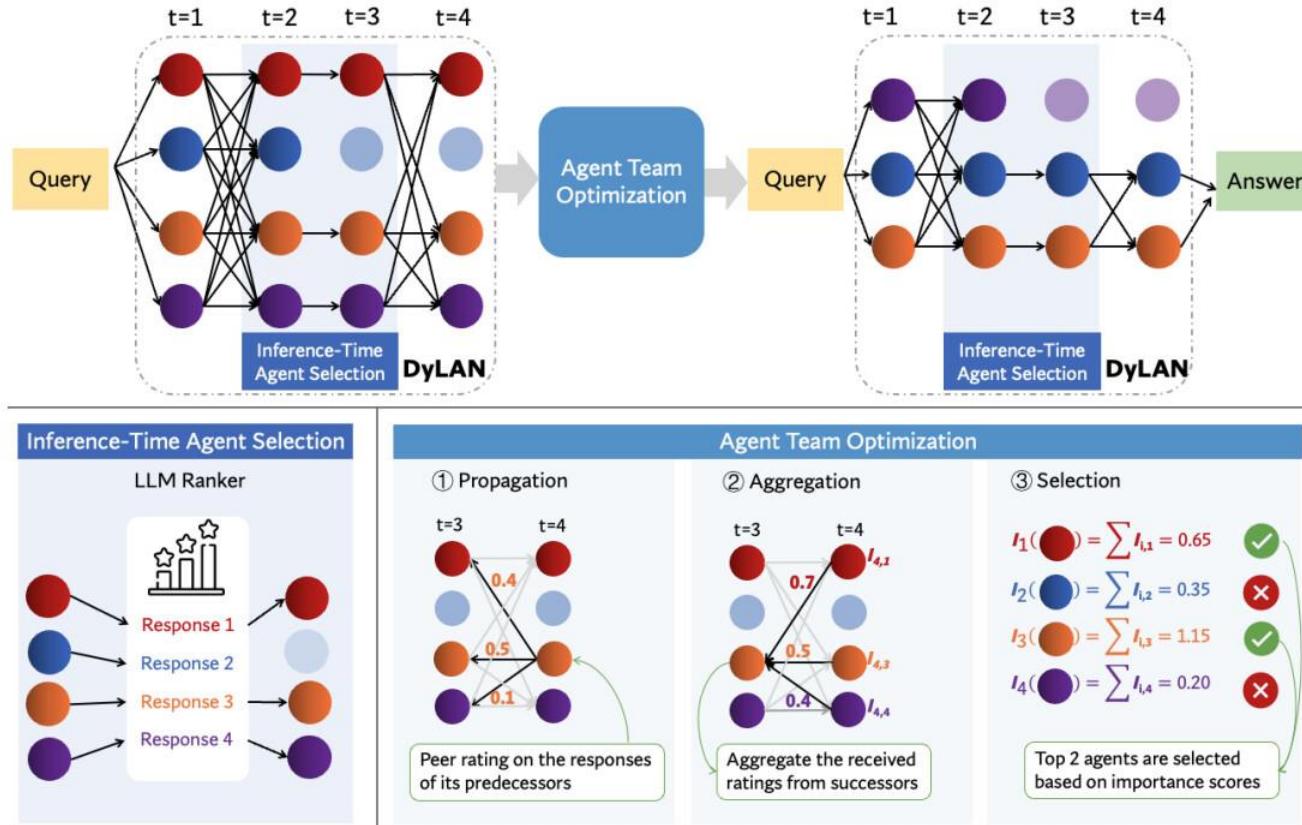


Figure 5: Agent teams with a designated leader achieve superior performance [9]

# | DyLAN (Dynamic LLM-Agent Network)



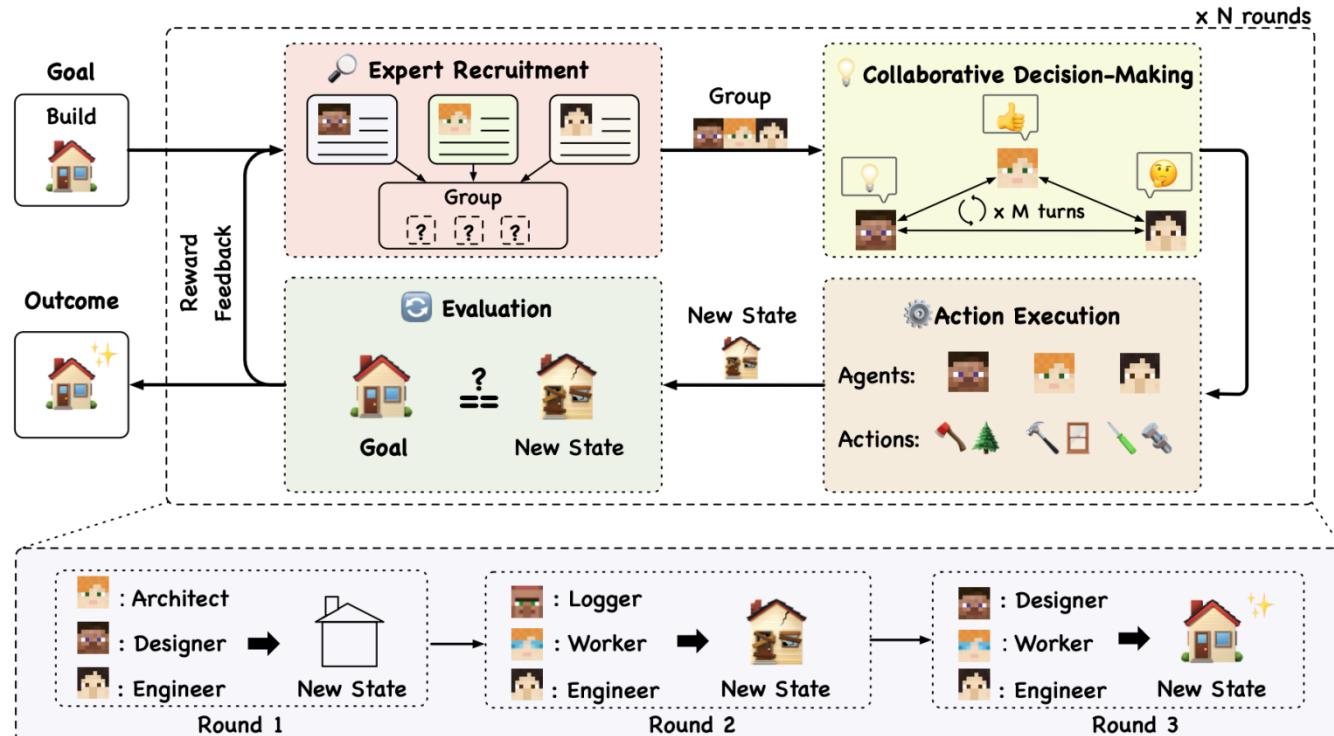
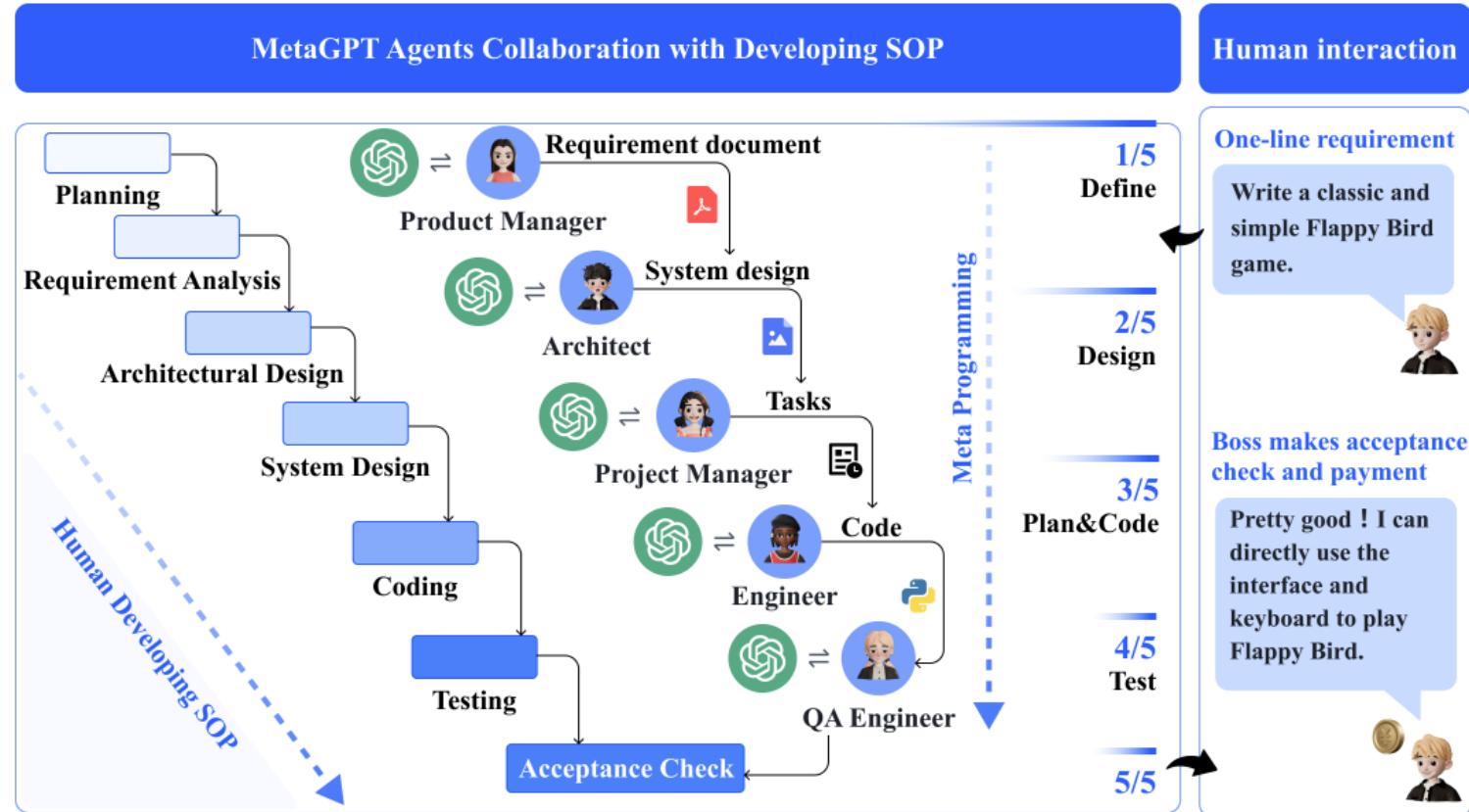


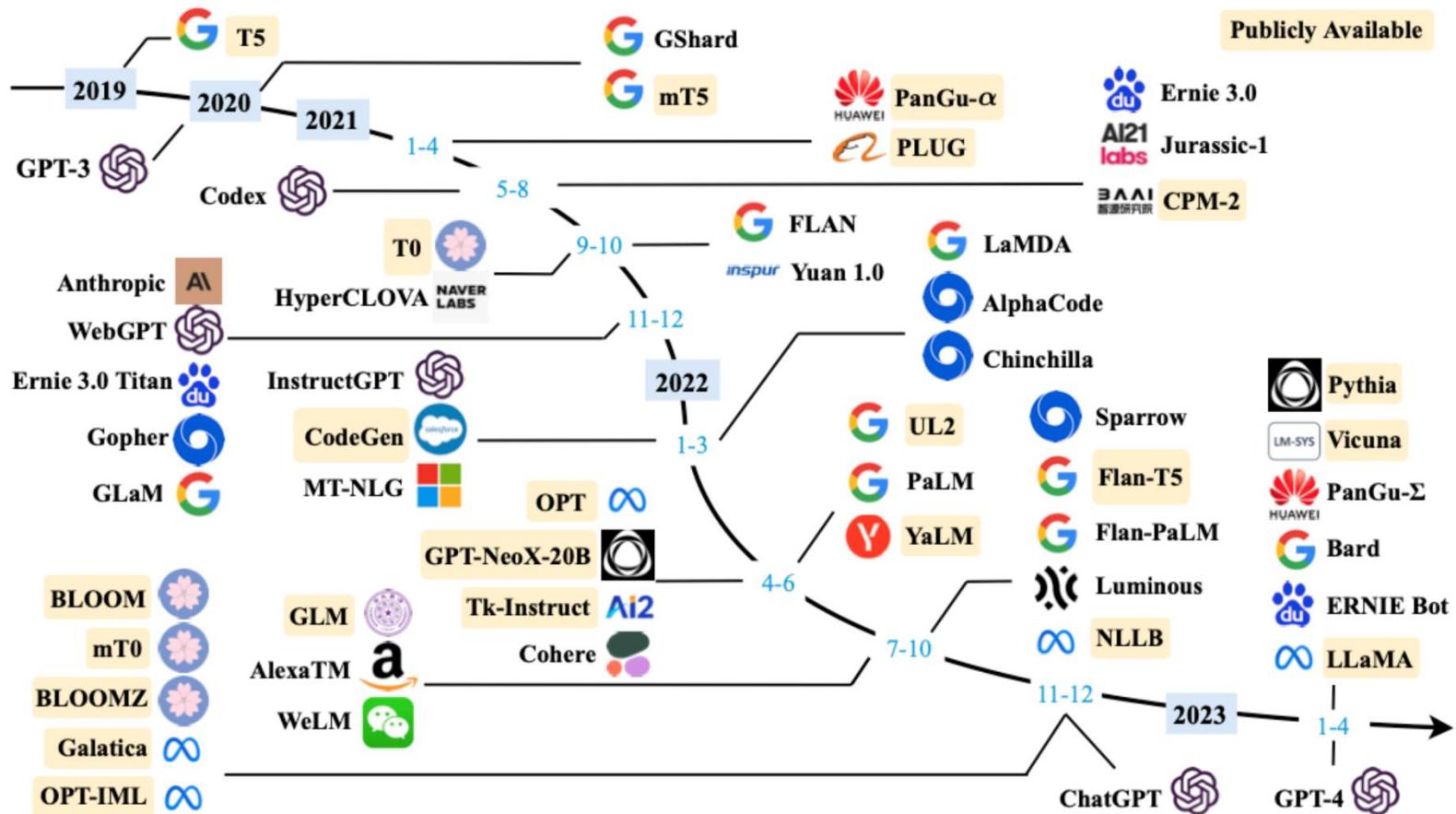
Figure 6: A diagram of the AgentVerse method [2]

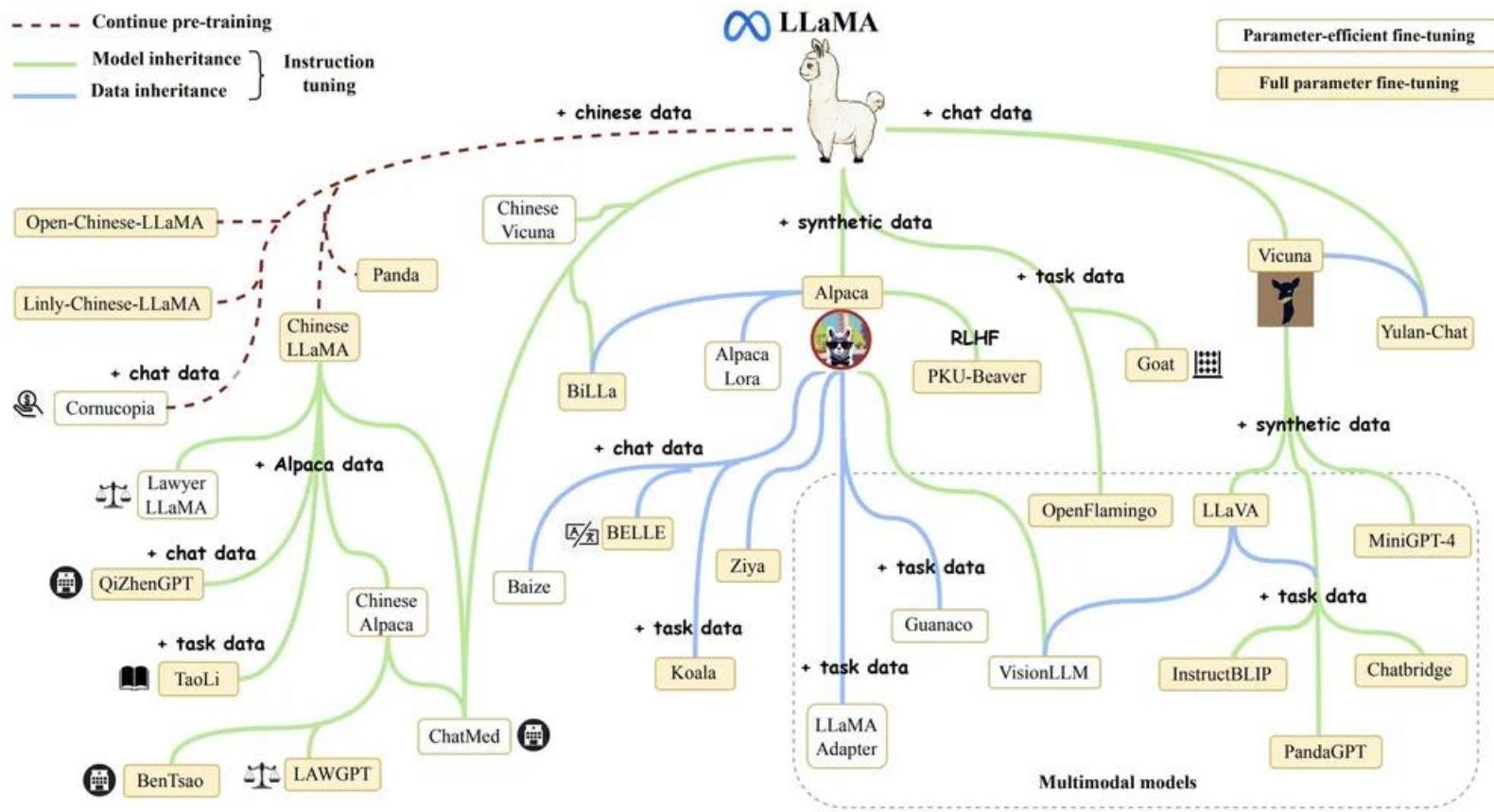


## | 단일 vs 다중 에이전트 아키텍처

---

- 단일 에이전트 패턴은 일반적으로 도구 목록이 좁게 정의되어 있고 프로세스가 명확한 작업에 적합
- 단일 에이전트는 하나의 에이전트와 도구 세트만 정의하면 되므로 구현도 일반적으로 더 쉬움
- 단일 에이전트 아키텍처는 다른 에이전트의 부정확한 피드백이나 관련 없는 잡담으로 인한 산만함 등의 제약을 받지 않음, 그러나 추론 및 개선 능력이 강건하지 않으면 실행 루프에 빠져 목표 달성을 이루지 못할 수 있음
- 다중 에이전트 아키텍처는 작업 수행에 있어 다양한 페르소나의 피드백이 도움이 되는 경우에 적합
- 다중 에이전트 시스템은 서로 다른 작업이나 워크플로우에 걸친 병렬 처리가 필요한 경우에도 유용
- 다중 에이전트 시스템은 본질적으로 더 복잡하며 견고한 대화 관리와 명확한 리더십의 이점을 누릴 수 있음
- 에이전트에 제공되는 프롬프트가 충분히 강력하다면 다중 에이전트 토론이 반드시 추론을 향상시키는 것은 아니다: 즉 추론능력 향상을 위해 다중 에이전트를 쓴다는 건 적합하지 않음





- 만약 NVIDIA 그래픽 카드가 시스템에 장착되어 있다면, CUDA 드라이버를 설치해서 LLM 구동에 GPU를 활용 할 수 있음: AMD 그래픽 카드의 경우 ROCm, 인텔은 OneAPI를 지원하지만 CUDA가 사실상 업계 표준. 다른 API의 경우 호환이 안되는 경우도 많고, 상당수 프로젝트들이 성능을 향상시키기 위해 CUDA-native 코드를 포함하고 있음
- 윈도우와 리눅스, 그래픽 카드 칩, 사용할 CUDA 버전등에 따라 설치방법은 매우 다양하나, 다음 사이트에서 설치 방법을 일괄하여 확인할 수 있음

<https://developer.nvidia.com/cuda-downloads>

- CUDA 설치 후 cuDNN까지 설치하고 나면 기본적인 세팅은 끝남: cuDNN은 컨벌루션이나 어텐션에 사용되는 연산을 CUDA로 가속하는 레이어를 구현한 라이브러리로, 신경망 연산을 빠르게 해주는 역할

- CUDA는 버전에 따라 지원하는 nvidia 드라이버 버전이 있음
- nvidia 드라이버는 GPU에 따라 사용 가능한 버전이 다름
- 결론적으로 각각의 환경에 따라 nvidia 드라이버 + CUDA 설치 방법은 다를 수 있음: 같은 명령으로 실행하더라도 실행이 될 수도 있고 안될 수도 있음, 자신의 환경에 맞는 방법을 찾아야 함
- WSL에서 nvidia 드라이버를 사용하려면 윈도우에서 먼저 nvidia 드라이버가 제대로 설치되어 있어야 함: 다음 명령으로 설치된 nvidia 드라이버의 버전과 상태를 확인할 수 있음

nvidia-smi.exe

```
C:\ 명령 프롬프트
C:\Users\노규남>nvidia-smi.exe
Fri May 24 22:54:51 2024
+-----+
| NVIDIA-SMI 531.41        Driver Version: 531.41        CUDA Version: 12.1 |
+-----+
| GPU  Name              TCC/WDDM | Bus-Id     Disp.A  | Volatile Uncorr. ECC |
| Fan  Temp   Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|          %          %          %          %       |           MiB |          %          %
|-----+
| 0  NVIDIA GeForce RTX 3060      WDDM | 00000000:05:00.0  On |      N/A | |
| 0%   41C   P8    15W / 170W| 546MiB / 12288MiB |     1%  Default |
|          |                           |           |          N/A |
+-----+
+-----+
| Processes:
| GPU  GI CI      PID  Type  Process name             GPU Memory |
| ID   ID          ID   ID               Usage          |
|-----+
| 0  N/A N/A 2676  C+G  ...siveControlPanel\SystemSettings.exe  N/A |
| 0  N/A N/A 3172  C+G  ...5n1h2txyewy\ShellExperienceHost.exe  N/A |
| 0  N/A N/A 10176 C+G  C:\Windows\explorer.exe                N/A |
| 0  N/A N/A 11660 C+G  ...nt.CBS_cw5n1h2txyewy\SearchHost.exe  N/A |
| 0  N/A N/A 11684 C+G  ...2txyewy\StartMenuExperienceHost.exe N/A |
| 0  N/A N/A 13740  C+G  ...GeForce Experience\NVIDIA Share.exe N/A |
| 0  N/A N/A 14176  C+G  ...ekyb3d8bbwe\PhoneExperienceHost.exe N/A |
| 0  N/A N/A 14640  C+G  ...t.LockApp_cw5n1h2txyewy\LockApp.exe N/A |
| 0  N/A N/A 17444  C+G  ...GeForce Experience\NVIDIA Share.exe N/A |
| 0  N/A N/A 18352  C+G  ...CBS_cw5n1h2txyewy\TextInputHost.exe N/A |
| 0  N/A N/A 18500  C+G  ...cal\Microsoft\OneDrive\OneDrive.exe N/A |
| 0  N/A N/A 22724  C+G  ...ta\Local\Programs\Notion\Notion.exe N/A |
| 0  N/A N/A 26448  C+G  ...oogle\Chrome\Application\chrome.exe N/A |
| 0  N/A N/A 28668  C+G  ...icrosoft\Edge\Application\msedge.exe N/A |
| 0  N/A N/A 31208  C+G  ..._8wekyb3d8bbwe\WindowsTerminal.exe N/A |
+-----+
C:\Users\노규남>
```

# CUDA Toolkit 12.5 Downloads

## Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

<b>Operating System</b>	Linux	Windows						
<b>Architecture</b>	x86_64	arm64-sbsa	aarch64-jetson					
<b>Distribution</b>	Amazon-Linux	Debian	Fedora	KylinOS	OpenSUSE	RHEL	Rocky	SLES
<b>Version</b>	Ubuntu	WSL-Ubuntu						
<b>Installer Type</b>	2.0	deb (local)	deb (network)	runfile (local)				

## Resources

- CUDA Documentation/Release Notes

- Archive of Previous CUDA Releases

## Installer Type

deb (local)

deb (network)

runfile (local)

## Download Installer for Linux WSL-Ubuntu 2.0 x86\_64

The base installer is available for download below.

## &gt; Base Installer

## Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-wsl-ubuntu.pin  
$ sudo mv cuda-wsl-ubuntu.pin /etc/apt/preferences.d/cuda-repository-pin-600  
$ wget https://developer.download.nvidia.com/compute/cuda/12.5.0/local_installers/cuda-repo-wsl-ubuntu-12-5-local_12.5.0-1_amd64.deb  
$ sudo dpkg -i cuda-repo-wsl-ubuntu-12-5-local_12.5.0-1_amd64.deb  
$ sudo cp /var/cuda-repo-wsl-ubuntu-12-5-local/cuda-*keyring.gpg /usr/share/keyrings/  
$ sudo apt-get update  
$ sudo apt-get -y install cuda-toolkit-12-5
```

Additional installation options are detailed [here](#).

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).

The checksums for the installer and patches can be found in [Installer Checksums](#).

For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

## Resources

- [CUDA Documentation/Release Notes](#)
- [MacOS Tools](#)
- [Training](#)
- [Sample Code](#)

- [Archive of Previous CUDA Releases](#)
- [FAQ](#)
- [Open Source Packages](#)
- [Submit a Bug](#)

# CUDA Toolkit 12.5 Downloads

## Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

### Operating System

Linux

Windows

## Resources

- [CUDA Documentation/Release Notes](#)
- [MacOS Tools](#)
- [Training](#)
- [Sample Code](#)
- [Forums](#)

- [Archive of Previous CUDA Releases](#)
- [FAQ](#)
- [Open Source Packages](#)
- [Submit a Bug](#)
- [Tarball and Zip Archive Deliverables](#)



Sign up for NVIDIA News

Subscribe

Follow NVIDIA Developer



Find more news and tutorials on [NVIDIA Technical Blog](#)

Join

# CUDA Toolkit Archive

Previous releases of the CUDA Toolkit, GPU Computing SDK, documentation and developer drivers can be found using the links below. Please select the release you want from the list below, and be sure to check [www.nvidia.com/drivers](http://www.nvidia.com/drivers) for more recent production drivers appropriate for your hardware configuration.

[Download Latest CUDA Toolkit](#)

[Learn More about CUDA Toolkit](#)

## Latest Release

[CUDA Toolkit 12.5.0 \(May 2024\)](#), [Versioned Online Documentation](#)

## Archived Releases

[CUDA Toolkit 12.4.1 \(April 2024\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.4.0 \(March 2024\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.3.2 \(January 2024\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.3.1 \(November 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.3.0 \(October 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.2.2 \(August 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.2.1 \(July 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.2.0 \(June 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.1.1 \(April 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.1.0 \(February 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.0.1 \(January 2023\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 12.0.0 \(December 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.8.0 \(October 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.7.1 \(August 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.7.0 \(May 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.6.2 \(March 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.6.1 \(February 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.6.0 \(January 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.5.2 \(February 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.5.1 \(November 2021\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.5.0 \(October 2021\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.4 \(February 2022\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.3 \(November 2021\)](#), [Versioned Online Documentation](#)  
[CUDA Toolkit 11.4.2 \(September 2021\)](#), [Versioned Online Documentation](#)

# cuDNN Downloads

## Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [NVIDIA Software License Agreement](#).

<b>Operating System</b>	Linux	Windows			
<b>Architecture</b>	x86_64	arm64-sbsa	aarch64-jetson		
<b>Distribution</b>	Tarball	Debian	RHEL	Rocky	Ubuntu
<b>Version</b>	20.04	22.04			
<b>Installer Type</b>	deb (local)	deb (network)			

## Resources

- [cuDNN Documentation](#)
- [Tarball and Zip Archive Deliverables](#)
- [Archive of Previous Releases](#)

cuDNN 9.1.1 Downloads | NVI

developer.nvidia.com/cudnn-downloads?target\_os=Linux&target\_arch=x86\_64&Distribution=Ubuntu&target\_version=22.04&target\_type=deb\_local

**Version**

20.04    22.04

**Installer Type**

deb (local)    deb (network)

**Download Installer for Linux Ubuntu 22.04 x86\_64**

The base installer is available for download below.

> Base Installer

**Download (1.6 GB)**

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cudnn/9.1.1/local_installers/cudnn-local-repo-ubuntu2204-9.1.1_1.0-1_amd64.deb
$ sudo dpkg -i cudnn-local-repo-ubuntu2204-9.1.1_1.0-1_amd64.deb
$ sudo cp /var/cudnn-local-repo-ubuntu2204-9.1.1/cudnn-*keyring.gpg /usr/share/keyrings/
$ sudo apt-get update
$ sudo apt-get -y install cudnn
```

To install for CUDA 11, perform the above configuration but install the CUDA 11 specific package:

```
$ sudo apt-get -y install cudnn-cuda-11
```

To install for CUDA 12, perform the above configuration but install the CUDA 12 specific package:

```
$ sudo apt-get -y install cudnn-cuda-12
```

**Resources**

.. ..

```
sudo su - root  
wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-wsl-  
ubuntu.pin  
sudo mv cuda-wsl-ubuntu.pin /etc/apt/preferences.d/cuda-repository-pin-600  
wget https://developer.download.nvidia.com/compute/cuda/12.1.0/local_installers/cuda-repo-wsl-  
ubuntu-12-1-local_12.1.0-1_amd64.deb  
sudo dpkg -i cuda-repo-wsl-ubuntu-12-1-local_12.1.0-1_amd64.deb  
sudo cp /var/cuda-repo-wsl-ubuntu-12-1-local/cuda-*keyring.gpg /usr/share/keyrings/  
sudo apt-get update  
sudo apt-get -y install cuda  
apt install nvidia-utils-535 # nvidia-smi
```

```
sudo su - root
```

```
wget https://developer.download.nvidia.com/compute/cudnn/9.1.1/local_installers/cudnn-local-repo-ubuntu2204-9.1.1_1.0-1_amd64.deb
```

```
sudo dpkg -i cudnn-local-repo-ubuntu2204-9.1.1_1.0-1_amd64.deb
```

```
sudo cp /var/cudnn-local-repo-ubuntu2204-9.1.1/cudnn-*-keyring.gpg /usr/share/keyrings/
```

```
sudo apt-get update
```

```
sudo apt-get -y install cudnn
```

- Google Colab(oratory)은 Google에서 제공하는 jupyter notebook 서비스, 구글 계정만 있으면 누구나 무료로 사용할 수 있음
- colab의 가장 큰 장점은, 무료 GPU를 사용할 수 있게 해준다는 것: Tesla K40 16GB GPU 컨테이너를 제공. OS는 Ubuntu 22.04
- 노트북은 구글 드라이브에 저장하며, 보존되지 않는 컨테이너 자체 스토리지도 제공
- 유료 버전으로 Colab pro, Colab Pro+, Colab Enterprise가 있으며, 별도로 컴퓨팅 단위를 구매하여 GPU를 구동시킬 수도 있음
- Colab의 세션은 최대 12시간까지만 유지되며, 그 후에는 스토리지의 내용이 모두 초기화됨: 기본 플랜에서는 백그라운드 실행을 제공하지 않고, 마우스나 커서의 움직임이 90초 동안 없으면 역시 세션이 리셋됨
- GPU를 어느 정도 사용하면 자동으로 회수되고, 일정 기간이 지나기 전까지 다시 할당이 제한됨

Colab 유료 서비스 가격 정정

colab.research.google.com/signup?utm\_source=notebook\_settings&utm\_medium=link&utm\_campaign=premium\_gpu\_selector

# 내게 맞는 Colab 요금제 선택하기

학업, 취미, ML 연구 등 다양한 목적으로 Colab을 사용할 수 있습니다.

Colab은(는) 항상 무료로 사용할 수 있지만 컴퓨팅 수요가 많은 경우 필요에 따라 유료 옵션을 구매할 수 있습니다.

제한사항이 적용됩니다. 여기에서 자세히 알아보세요.

### Pay As You Go

\$9.99/100 컴퓨팅 단위  
\$49.99/500 컴퓨팅 단위

현재 컴퓨팅 단위가 0개 있습니다.  
컴퓨팅 단위는 90일 후 만료됩니다. 필요하면 더 구매하세요.

- ✓ 구독이 필요하지 않습니다.  
사용한 만큼만 비용을 지불하세요.
- ✓ 더 빠른 GPU  
더 강력한 GPU로 업그레이드하세요.

### Colab Pro

월 \$9.99

- ✓ 100 컴퓨팅 단위/월  
컴퓨팅 단위는 90일 후 만료됩니다. 필요하면 더 구매하세요.
- ✓ 더 빠른 GPU  
더 강력한 GPU로 업그레이드하세요.
- ✓ 추가 메모리  
고성능 메모리 마シン에 액세스하세요.
- ✓ 터미널  
연결된 VM으로 터미널 사용 가능
- ✓ 일부 국가 및 18세 이상만 해당:
- ✓ AI 지원 자동 완성  
입력 중에 여러 줄을 제안하는 지능형 추천이 자동으로 렌더링됩니다.
- ✓ 코드 생성  
통합 챗봇을 비롯하여 자연어로 코드를 생성합니다.

### Colab Pro+

월 \$49.99

- Pro의 모든 혜택 및 다음 추가 혜택:
  - ✓ 주가 컴퓨팅 단위 400개가 지원되어 매월 총 500개 이용 가능  
컴퓨팅 단위는 90일 후 만료됩니다. 필요하면 더 구매하세요.
  - ✓ 더 빠른 GPU  
더 강력한 프리미엄 GPU로 한발 앞서 업그레이드하세요.
  - ✓ 백그라운드 실행  
컴퓨팅 단위를 통해 브라우저를 닫아도 현재 실행 중인 노트북이 최대 24시간 동안 계속 실행됩니다.

### Colab Enterprise

사용한 만큼만 비용 지불

- ✓ 통합  
BigQuery 및 Vertex AI와 같은 Google Cloud 서비스와 긴밀하게 통합됩니다.
- ✓ Enterprise 노트북 스토리지  
Google Drive 노트북 사용을 Cloud 콘솔 내에서 저장 및 공유하는 GCP 노트북으로 대체하세요.
- ✓ 생산성  
생성형 AI 기반 코드 완성 및 생성

- 1) github 계정 생성, 메일로 초청 받아 프로젝트 참여
- 2) colab 노트북 실행, 구글 드라이브 마운트

```
from google.colab import drive  
import os  
  
# Google Drive를 /content/gdrive에 마운트  
drive.mount('/content/gdrive')  
  
# ~/work 디렉토리를 생성  
os.makedirs('/content/work', exist_ok=True)  
  
# Google Drive를 ~/work에 마운트  
!ln -s /content/gdrive/MyDrive/ /content/work
```

### 3) 계정 설정

```
!git config --global user.name "Bard K. Roh"  
!git config --global user.email "bardroh@weable.ai"  
!git config --global --list
```

### 4) github token 설정 & git clone: Setting-Developer Settings에서 classic token으로 생성. repo 관련 모든 권한만 부여되어 있으면 됨

```
%cd /content/work/MyDrive  
!git clone https://[사용자 id]:[access token]@github.com/weable-weable/koshipa-llm-2025-1st.git
```

### 5) 작업하고 git add; git commit; git push, 작업 전 항상 git pull (본인 폴더만 작업)

- GPU가 장착된 시스템을 구비하기 어렵다면, 클라우드에서 임대하는 방법이 있음
- 대개의 CSP에서 GPU는 매우 비싼 자원: 무료 티어로는 사용할 수 없고, 사용을 위해서는 별도 품으로 신청하여 GPU quota를 할당 받아야 함
- 구글 클라우드는 가입시 \$300의 90일 기한의 크레딧을 제공하나, 무료계정인 상태에서 이 크레딧으로는 GPU VM을 사용할 수 없음
- 무료계정을 일반계정으로 업그레이드하면 카드 청구가 가능해지는데, 이 상태에서 GPU quota를 할당 받을 수 있고, GPU VM을 만들 수 있음
- 일반계정으로 업그레이드하더라도 받은 \$300 크레딧이 소멸되는 것은 아니기 때문에, GPU VM을 생성하여 사용하고 90일내 탈퇴하면 추가 비용 없이 GPU 학습이 가능함
- 유료 계정을 사용할 때는 해킹의 피해가 커질 수 있으므로, 이중 인증(Authenticator)을 설정해두면 좋음

<https://myaccount.google.com/u/0/security>

클라우드 컴퓨팅 서비스 | Google Cloud +

cloud.google.com

앱 Gmail YouTube 다른 기기의 템 지도 뉴스

Google Cloud Google을 선택해야 하는 이유 솔루션 제품 가격 책정 시작하기 문의하기

한국어 문서 지원 로그인 무료로 시작하기

신규 고객에게는 Google Cloud에 사용할 수 있는 \$300의 무료 크레딧이 제공됩니다. 모든 고객에게 20여 개 제품에 대한 무료 사용량이 제공됩니다. [혜택 세부정보 보기](#).

# Google Cloud로 앞당기 는 혁신

앱을 더 빠르게 빌드하고 보다 현명한 비즈니스 의사결정을 내리며 세계 각지의 사람들과 소통할 수 있습니다.

[무료로 시작하기](#)



두산중공업이 Google Cloud와 만나 제조업의 미래를 그려나가고 있습니다.



새로운 앱과 환경의 개발 속도 향상

Google Cloud의 오픈소스, [멀티 클라우드](#),

조직에서 보다 현명한 의사결정을 내릴 수 있도록 지원

소통 및 공동작업 방식의 혁신

통합된 영상 통화, 이메일, 채팅, 문서

안심할 수 있는 클라우드 빌드 및 투자

[고급 보안 서비스](#)로 데이터를 보호하고 혁

A screenshot of a Google Cloud Platform login page. The page features the Google logo at the top center. Below it, the word "로그인" (Login) is displayed in large, bold, black Korean characters. Underneath, the text "Google 클라우드 플랫폼으로 이동" (Move to Google Cloud Platform) is shown. A large input field is present for entering an email or phone number, with the placeholder "이메일 또는 휴대전화" (Email or mobile phone). Below the input field, a link "이메일을 잊으셨나요?" (Forgot your email?) is provided. Further down, a note in Korean encourages users to log in as a guest if they're not on a computer, with a link "자세히 알아보기" (Learn more). At the bottom, there are two buttons: "계정 만들기" (Create account) on the left and a blue "다음" (Next) button on the right.

A screenshot of a web browser window showing the Google Account Creation page. The URL is accounts.google.com/signup/v2/webcreateaccount?service=cloudconsole&continue=https%3A%2F%2Fconsole.cloud.google.com%2Ffreetrial%3F\_ga%3D2.70473744.1899673253.1628493719-357... The page title is "Google 계정 만들기". It shows fields for First Name (ROH), Last Name (KYOUNAM), and Email Address (bard@hitel.net). Below the email field is a message: "이 이메일이 본인 소유인지 확인해야 합니다." (This email must be verified by the owner). There is a link "새로운 Gmail 주소 만들기" (Create a new Gmail address) and a password input field with two masked entries. A checkbox "비밀번호 표시" (Show password) is checked. On the right, there is a blue shield icon with a white person silhouette, and below it, a graphic showing a laptop screen with a play button, a smartphone with a messaging bubble, and a location pin. Text next to the graphic says "하나의 계정으로 모든 Google 서비스를 이용할 수 있습니다." (You can use all Google services with one account). At the bottom, there are buttons for "대신 로그인하기" (Log in instead) and "다음" (Next).



**Google**

## 이메일 주소 인증

bard@hitel.net(으)로 전송한 인증 코드를 입력하세요.  
표시되지 않으면 스팸 폴더를 확인해 보세요.

코드 입력 \_\_\_\_\_

뒤로

확인

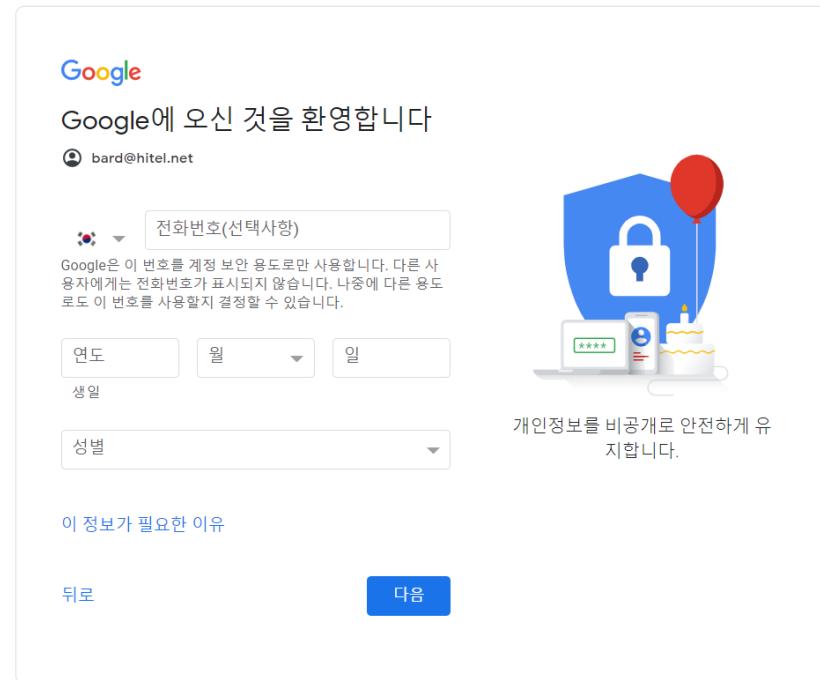


한국어

도움말

개인정보처리방침

약관



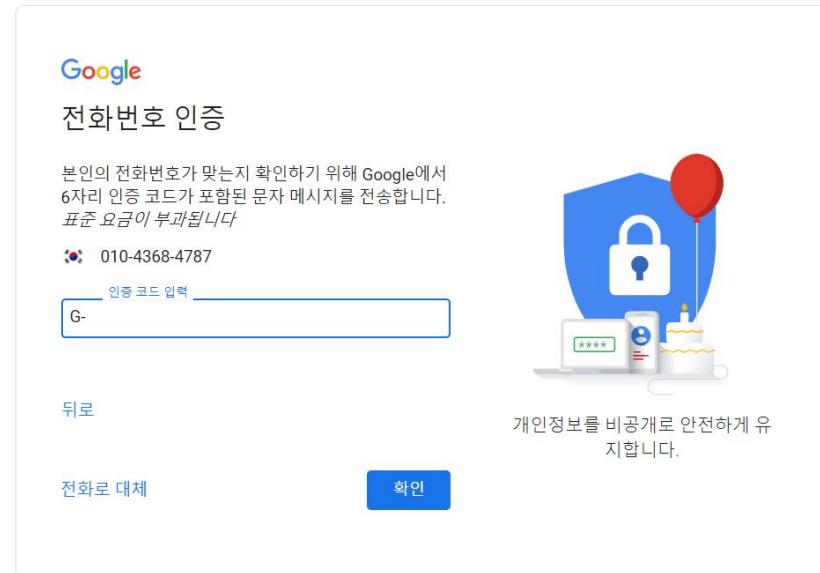
한국어

다음

도움말

개인정보처리방침

약관



한국어

도움말

개인정보처리방침

20

The screenshot shows a Google account setup page. The title bar indicates it's for creating a new account. The main content is titled "전화번호 다양하게 활용하기" (Use your phone number in various ways). It explains that adding a phone number to the account allows users to access Google services like Google Meet and Google AdSense. A blue shield icon with a lock and a red balloon is displayed. A note at the bottom right states that personal information is stored securely. Navigation buttons include "뒤로" (Back), "건너뛰기" (Skip), and "예" (Yes). Language options at the bottom include Korean, English, and other languages.

The screenshot shows a step in the Google account creation process. The title bar indicates the user is creating a new account. The main content area is titled 'Google' and '개인정보 보호 및 약관' (Personal Information Protection and Terms of Service). It contains text about reading the Google Terms of Service before account creation, links to the Google Services Terms of Service, Location Services Terms of Service, and the Personal Information Collection History · Use · Privacy · Security · Terms of Service. Below this, it discusses how Google handles user data when they use Google services, listing activities like setting up accounts, writing emails, watching YouTube videos, and searching for locations. To the right, there is an icon featuring a shield with two sliders and icons for Google Sheets, Google Slides, Google Photos, and an '@' symbol.

Google 계정을 생성하기 전에 Google의 약관을 주의 깊게 읽어보시기 바랍니다. Google 계정을 생성하기 위해서는 [Google 서비스 약관](#), [위치서비스 이용약관](#) 및 [개인정보 수집항목·이용목적·보유기간](#)에 동의해야 합니다.

Google의 서비스 약관에는 사용자와 Google의 관계, Google 서비스에서 제공하는 콘텐츠와 소프트웨어의 사용, 분쟁 해결을 포함하지만 이에 국한되지 않는 중요한 내용이 설명되어 있습니다.

또한 계정을 만들 때 Google에서는 다음과 같은 주요 사항을 포함하여 Google 개인정보처리방침에 설명된 바와 같이 사용자의 정보를 처리합니다.

사용자가 Google을 사용할 때 Google에서 처리하는 정보

- Google 계정을 설정할 때 제공하신 이름, 이메일 주소, 전화번호와 같은 정보가 저장됩니다.
- 사용자가 Gmail에서 메시지를 작성하거나 YouTube 동영상에 댓글을 달는 등의 활동을 하기 위해 Google 서비스를 사용하면 Google은 사용자가 만든 정보를 저장합니다.
- 예를 들어 사용자가 Google 지도에서 식당을 검색하거나 YouTube에서 동영상을 시청할 때 Google에서는 사용자의 활동에 관한 정보(예:

총 3단계 중 1단계 – 무료 체험판

console.cloud.google.com/freetrial/signup/tos?\_ga=2.70473744.1899673253.1628493719-357501966.1626587934&\_gac=1.218255083.1628493723.CjwKCAjwpMOIBhBAEiwAy5M6YG158yK30... [검색] [별기록] [마크] [닫기] [목록]

앱 Gmail YouTube 다른 기기의 템 지도 뉴스

## Google Cloud 무료로 사용해 보기

### 총 3단계 중 1단계 계정 정보



KYOUNAM ROH  
bard@hitel.net

계정 전환

#### 국가

대한민국

#### 서비스 약관

[Google Cloud Platform 서비스 약관과 관련 서비스 및 API의 서비스 약관에 동의합니다. 또한 Google Cloud Platform 무료 체험판 서비스 약관을 읽었으며 이에 동의합니다.](#)

계속 진행하려면 체크박스를 선택하세요.

현재 또는 향후에 [상업적 목적](#)으로 Google Cloud Platform을 사용할 계획입니다.

계속 진행하려면 체크박스를 선택해야 합니다.

#### 이메일 업데이트

### 모든 Cloud Platform 제품에 액세스

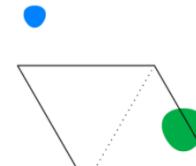
Firebase, Google Maps API 등을 포함해 앱, 웹사이트, 서비스를 구축하고 실행하는 데 필요한 모든 기능을 이용할 수 있습니다.

### \$300의 무료 크레딧

90일간 사용할 수 있는 \$300 크레딧으로 Google Cloud를 실제 작업에 활용할 수 있습니다.

### 무료 체험판 종료 후 자동 청구되지 않음

신용카드를 요청하는 이유는 자동 가입을 방지하기 위해서입니다. 유료 계정으로 직접 업그레이드하지 않는 한 요금이 청구되지 않습니다.



총 3단계 중 2단계 – 무료 체험판

console.cloud.google.com/freetrial/signup/verify?\_ga=2.70473744.1899673253.1628493719-357501966.1626587934&\_gac=1.218255083.1628493723.CjwKCAjwpMOIBhBAEiwAy5M6YG158jK... 읽기 목록

앱 Gmail YouTube 다른 기기의 템 지도 뉴스



## 총 3단계 중 2단계 본인 확인 및 연락처 정보

Cloud 환경을 지원하기 위해 솔루션에 대한 소식을 보낼 수 있는 연락처를 확인합니다. Google 계정과 연결된 번호로 계속 진행하거나 다른 번호를 선택하세요.

010-4368-4787

계속

다른 번호 사용

### 모든 Cloud Platform 제품에 액세스

Firebase, Google Maps API 등을 포함해 앱, 웹사이트, 서비스를 구축하고 실행하는 데 필요한 모든 기능을 이용할 수 있습니다.

### \$300의 무료 크레딧

90일간 사용할 수 있는 \$300 크레딧으로 Google Cloud를 실제 작업에 활용할 수 있습니다.

### 무료 체험판 종료 후 자동 청구되지 않음

신용카드를 요청하는 이유는 자동 가입을 방지하기 위해서입니다. 유료 계정으로 직접 업그레이드하지 않는 한 요금이 청구되지 않습니다.

총 3단계 중 3단계 - 무료 체험판

console.cloud.google.com/freetrial/signup/billing/KR?\_ga=2.70473744.1899673253.1628493719-357501966.1626587934&\_gac=1.218255083.1628493723.CjwKCAjwpMOIBhBAEiwAy5M6YG1... 읽기 목록

앱 Gmail YouTube 다른 기기의 템 지도 뉴스

## Google Cloud 무료로 사용해 보기

### 총 3단계 중 3단계 결제 정보 확인

사용자의 결제 정보는 사기와 악용을 줄이는 데 도움이 됩니다. 자동 결제를 사용 설정하지 않는 한 요금이 청구되지 않습니다.

#### 고객 정보

계정 유형 ① 수정

개인

세금 정보 ①

사업자 유형: 개인

이름 및 주소 ①

도/시

시/군/구

#### 모든 Cloud Platform 제품에 액세스

Firebase, Google Maps API 등을 포함해 앱, 웹사이트, 서비스를 구축하고 실행하는 데 필요한 모든 기능을 이용할 수 있습니다.

#### \$300의 무료 크레딧

90일간 사용할 수 있는 \$300 크레딧으로 Google Cloud를 실제 작업에 활용할 수 있습니다.

#### 무료 체험판 종료 후 자동 청구되지 않음

신용카드를 요청하는 이유는 자동 가입을 방지하기 위해서입니다. 유료 계정으로 직접 업그레이드하지 않는 한 요금이 청구되지 않습니다.

총 3단계 중 3단계 - 무료 체험판

console.cloud.google.com/freetrial/signup/billing/KR?\_ga=2.70473744.1899673253.1628493719-357501966.1626587934&\_gac=1.218255083.1628493723.CjwKCAjwpMOIBhBAEiwAy5M6YG1... 닫기

◀ ▶ ⌂ 다른 기기의 탭 지도 뉴스

고객 정보 입력  
카드 발급 은행의 요청에 따라 Google은 귀하의 본인 확인을 해야 합니다.  
계속 진행하면 Google이 은행 및 인증 기관에 귀하의 정보를 공유할 수 있도록  
[Google 개인정보처리방침](#)에 동의하는 것으로 간주됩니다.

이름  
 이름은(는) 필수 입력란입니다.

주민등록번호(앞 7자리 한정)  
 주민등록번호(앞 7자리 한정)은(는) 필수 입력란입니다.

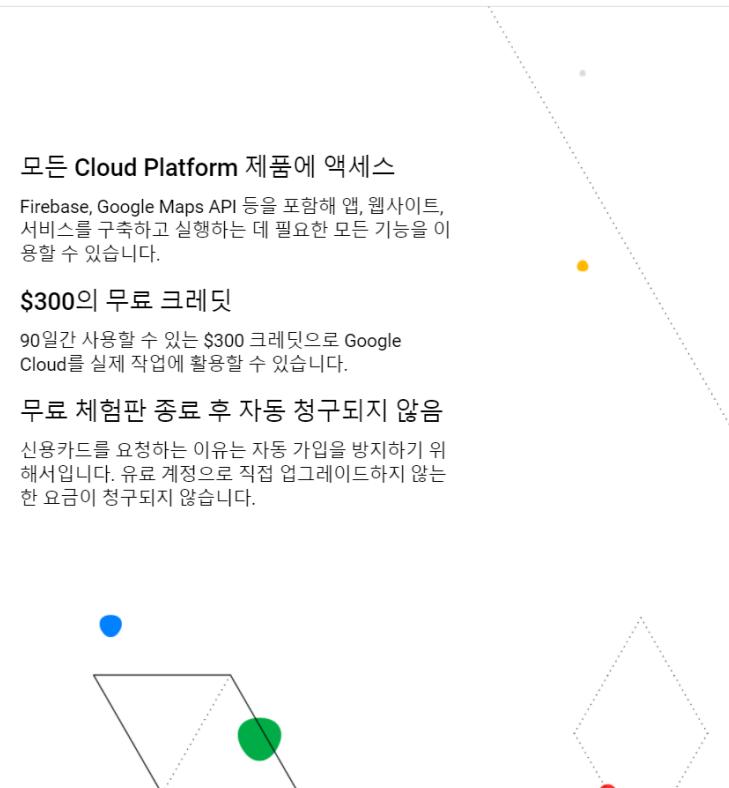
이동통신사 ▼

전화번호 코드 전송

코드 입력  
제공한 휴대전화 번호로 전송된 인증 코드를 입력하세요.

계속 진행하면 Google이 귀하의 결제 프로필 정보를 이 계정에 연결하고 Google 제품 전체에서 동일한 정보를 공유하고 사용할 수 있도록 [Google 개인정보처리방침](#)에 동의하는 것으로 간주됩니다. 이 정보는 언제든지 Google 계정에서 삭제할 수 있습니다.

계속



총 3단계 중 3단계 - 무료 체험판

console.cloud.google.com/freetrial/signup/billing/KR?\_ga=2.70473744.1899673253.1628493719-357501966.1626587934&\_gac=1.218255083.1628493723.CjwKCAjwpMOIBhBAEiwAy5M6YG1... 읽기 목록

대한민국 13481 경기도 성남시 분당구  
서판교로 29  
918-701  
KYOUNAM ROH

결제 옵션

자동 결제

비용이 발생한 후에만 서비스를 결제합니다. 청구 기준액에 도달하거나 지난 자동 결제일로부터 30일이 경과하면 둘 중 더 이른 날짜에 비용이 자동 청구됩니다.

결제 수단 ⓘ

카드 번호

#

계속 진행하면 Google이 귀하의 결제 프로필 정보를 이 계정에 연결하고 Google 제품 전체에서 동일한 정보를 공유하고 사용할 수 있도록 [Google 개인정보처리방침](#)에 동의하는 것으로 간주됩니다. 이 정보는 언제든지 Google 계정에서 삭제할 수 있습니다.

[무료 평가판 시작하기](#)

**모든 Cloud Platform 제품에 액세스**

Firebase, Google Maps API 등을 포함해 앱, 웹사이트, 서비스를 구축하고 실행하는 데 필요한 모든 기능을 이용할 수 있습니다.

**\$300의 무료 크레딧**

90일간 사용할 수 있는 \$300 크레딧으로 Google Cloud를 실제 작업에 활용할 수 있습니다.

**무료 체험판 종료 후 자동 청구되지 않음**

신용카드를 요청하는 이유는 자동 가입을 방지하기 위해서입니다. 유료 계정으로 직접 업그레이드하지 않는 한 요금이 청구되지 않습니다.

시작하기 - My First Project - G...

console.cloud.google.com/getting-started?project=nice-dispatcher-322407

Google Cloud Platform My First Project

제품 및 리소스 검색

Google Cloud Platform KYOUNAM님, 환영합니다.

무료 체험판에는 90일간 사용할 수 있는 \$300 크레딧이 포함되어 있습니다. 보다 나은 서비스를 제공할 수 있도록 4가지 질문에 답해 주세요.

1 조직 또는 니즈를 가장 잘 설명하는 것이 무엇인가요?  
답변 선택 \*  
스타트업

2 Google Cloud를 방문하신 이유가 무엇인가요?

3 Google Cloud로 어떤 작업을 하고 싶으신가요?

4 담당하는 역할을 가장 잘 설명하는 것이 무엇인가요?

닫기 완료

카드를 저장하시겠습니까?  
Amex 8394 22/08  
저장 취소

KYOUNAM님, 환영합니다.

무료 체험판에는 90일간 사용할 수 있는 \$300 크레딧이 포함되어 있습니다. 보다 나은 서비스를 제공할 수 있도록 4가지 질문에 답해 주세요.

1 조직 또는 니즈를 가장 잘 설명하는 것이 무엇인가요?  
답변 선택 \*  
스타트업

2 Google Cloud를 방문하신 이유가 무엇인가요?

3 Google Cloud로 어떤 작업을 하고 싶으신가요?

4 담당하는 역할을 가장 잘 설명하는 것이 무엇인가요?

닫기 완료

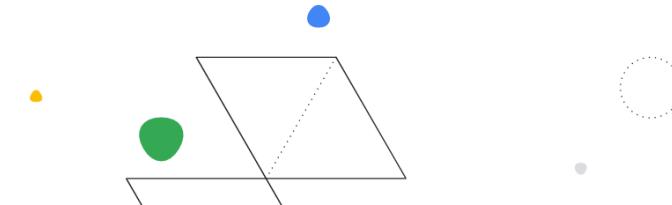
카드를 저장하시겠습니까?  
Amex 8394 22/08  
저장 취소

Google Cloud Platform My First Project 제품 및 리소스 검색

홈 최근 고정됨 Compute Engine Cloud Storage VPC 네트워크 App Engine SQL Kubernetes Engine BigQuery 모든 제품

# KYOUNAM님, 환영합니다

Google Cloud 시작하기



## 기본사항으로 시작하기

일반적인 작업을 확인하여 신속한 시작 및 실행

체크리스트로 이동

확장 가능하며 프로덕션에 바로 사용 가능한 엔터프라이즈 워크로드를 위해 Google Cloud를 설정하시나요?  
관리자용으로 마련된 [Google Cloud 설정 체크리스트](#)를 사용하세요.

다루는 내용

- 결제, 크레딧, 프로젝트 검토
- 제품 및 API 찾기
- 프로젝트에 리소스 추가
- 가격 책정 이해 및 계산

## 인기 제품

모두 보기

## 컴퓨팅 제품

... 디스크 드라이브 캐시 가상 머신 ...

홈 - My First Project - Google Cloud Platform

console.cloud.google.com/home/dashboard?project=nice-dispatcher-322407

새로운 탭 닫기 브라우저 설정

구글 앱 Gmail YouTube 다른 기기의 탭 지도 뉴스

읽기 목록

무료 평가판 상태: 크레딧은 ₩346,326.00, 무료 평가판 기간은 91일 남았습니다. 완전한 계정을 사용하면 Google Cloud Platform의 모든 기능에 무제한 액세스할 수 있습니다.

닫기 활성화

## Google Cloud Platform 대시보드

### My First Project

### 제품 및 리소스 검색

대시보드 활동 권장사항

맞춤설정

#### 프로젝트 정보

프로젝트 이름  
My First Project

프로젝트 ID  
nice-dispatcher-322407

프로젝트 번호  
147271846376

[이 프로젝트에 사용자 추가](#)

[프로젝트 설정으로 이동](#)

#### API API

요청(요청/초)

No data is available for the selected time frame.

[API 개요로 이동](#)

#### Google Cloud Platform 상태

모든 서비스 정상

[Cloud 상태 대시보드로 이동](#)

#### 모니터링

내 대시보드 만들기

알림 정책 설정

업타임 체크 만들기

[모든 대시보드 보기](#)

[모니터링으로 이동](#)

#### 추적

홈 - My First Project - Google Cloud Platform

console.cloud.google.com/home/dashboard?project=nice-dispatcher-322407

구글 앱 Gmail YouTube 다른 기기의 템 지도 뉴스

무료 평가판 상태: 크레딧은 ₩346,326.00, 무료 평가판 기간은 91일 남았습니다. 완전한 계정을 사용하면 Google Cloud Platform의 모든 기능에 무제한 액세스할 수 있습니다.

닫기 활성화

### Google Cloud Platform

- 홈
- 최근
- 고정됨
- Compute Engine
- Cloud Storage
- VPC 네트워크
- App Engine
- SQL
- Kubernetes Engine
- BigQuery

모든 제품

https://console.cloud.google.com/compute?project=nice-dispatcher-322407

### 제품 및 리소스 검색

#### API API

요청(요청/초)

No data is available for the selected time frame.

3:45 4 오후 4:15 4:30

→ API 개요로 이동

#### Google Cloud Platform 상태

모든 서비스 정상

→ Cloud 상태 대시보드로 이동

#### 모니터링

내 대시보드 만들기

알림 정책 설정

업타임 체크 만들기

모든 대시보드 보기

→ 모니터링으로 이동

가상 머신

VM 인스턴스

인스턴스 템플릿

단독 테넌트 노드

머신 이미지

TPU

작정 사용 할인

Migrate for Compute Engine

스토리지

디스크

스냅샷

이미지

인스턴스 그룹

인스턴스 그룹

상태 확인

VM MANAGER

폐지 관리

- 새 VM 인스턴스. 서울 리전 선택(asia-northeast3). asia-northeast3-b나 asia-northeast3-c
- n1-standard-4(vCPU 4개, 코어 2개, 메모리 15G)로 생성, GPU 추가, 유형은 NVIDIA T4
- Ubuntu 22.04, 균형있는 영구 디스크 100G, 이름은 gpu
- 콘솔에서 바로 ssh 접속됨: keypair로도 접속 가능하지만 다른 프로그램 설치 없이 바로 작업이 가능하다는 장점, 파일 업/다운로드도 됨
- sudo su - root로 루트 권한을 얻은 후 로컬에서와 동일하게 설정
- CUDA 및 cuDNN을 설치하면 로컬이나 colab과 동일한 환경이 됨, 다만 colab과 다르게 세션 시간에 대한 제약이 없으므로 학습이나 시간 오래 걸리는 작업은 이쪽이 적합

Compute Engine – My First Project

console.cloud.google.com/compute/instancesAdd?authuser=1&project=quantum-toolbox-425202-b9

Google Cloud My First Project 리소스, 문서, 제품 등 검색( /) 김색

인스턴스 만들기 상응하는 코드

새 VM 인스턴스 VM 인스턴스 하나를 처음부터 만듭니다.

템플릿에서 VM 인스턴스 만들기 기존 템플릿에서 VM 인스턴스 하나를 만듭니다.

머신 이미지에서 VM 인스턴스 만들기 기존 머신 이미지에서 VM 인스턴스 하나를 만듭니다.

Marketplace VM 인스턴스에 바로 사용할 수 있는 솔루션을 배포합니다.

이름 \* gpu

월별 예상 가격 US\$314.93 시간당 약 US\$0.43 사용한 만큼만 비용 지불·선불 비용 없이 초당 청구

태그 및 라벨 관리

리전 \* asia-northeast3 (서울) 영역 \* asia-northeast3-b

항목 월별 예상 가격

4 vCPU + 15 GB memory	US\$177.94
1 NVIDIA T4	US\$270.10
10GB 분산된 영구 디스크	US\$1.30
사용 할인	-US\$134.41
Total	US\$314.93

머신 구성

GPU GPU 유형 NVIDIA T4 GPU 개수 1

Compute Engine 가격 책정

LESS

가상 워크스테이션 사용 설정(NVIDIA GRID)

Series	설명	vCPUs	Memory	플랫폼
N1	균형을 이룬 가격과 성능	1 - 96	1.8~624GB	Intel Skylake

머신 유형

대부분의 워크로드에 적합한 vCPU 수와 메모리 양이 사전 설정된 머신 유형을 선택합니다. 또는 워크로드의 특정 니즈에 맞는 커스텀 머신을 만들 수도 있습니다. 자세히 알아보기

나중에 다시 찾기

만들기 취소 상응하는 코드

상상력을 키우는 코드

```
bard@docker1: ~ - Chrome
  ssh.cloud.google.com/projects/nice-dispatcher-322407/zones/asia-northeast3-a/instances/docker1?authuser=0&hl=ko&projectNum...
Connected, host fingerprint: ssh-rsa 0 12:93:60:AC:9B:F0:AA:D5:45:CF:E4:C6:1B:AE
:A6:77:35:3D:5F:70:E4:DC:05:C0:35:2D:A1:D2:6F:68:0D:45
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1049-gcp x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

0 updates can be applied immediately.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Aug  9 07:49:10 2021 from 35.235.242.81
bard@docker1:~$
```

제한 액세스할 수 있습니다. 닫기 활성화

시작/재개 : OPERATIONS 정보 패널 표시 알아보기

종: 내부 IP 외부 IP 연결

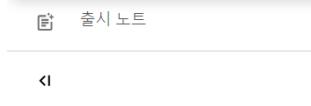
	10.178.0.3 (nic0)	34.64.118.58	SSH
--	-------------------	--------------	-----

닫기

펴보기 그를 보고, 하며, 다운로드.

방화벽 규칙 설정 VM 인스턴스와 주고받는 트래픽을 제어합니다.

패치 관리 패치 업데이트를 예약하고 VM 인스턴스의 패치 규정 준수를 확인합니다.



← 2단계 인증

계정이 2단계 인증으로 보호되고 있습니다

해커가 계정에 액세스하지 못하도록 보안을 강화하세요.

패스키가 아닌 다른 방법으로 로그인하는 경우, 계정에서 사용할 수 있는 가장 안전한 2단계를 완료하라는 메시지가 표시됩니다. 설정에서 언제든지 2단계와 로그인 옵션을 업데이트할 수 있습니다. [보안 설정으로 이동](#)



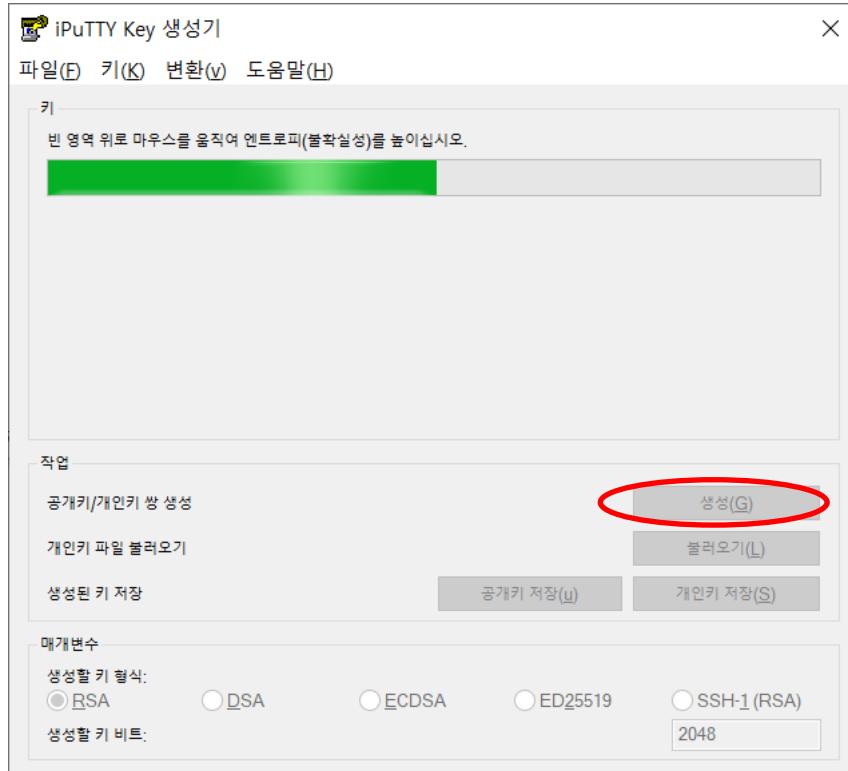
[2단계 인증 사용 중지](#)

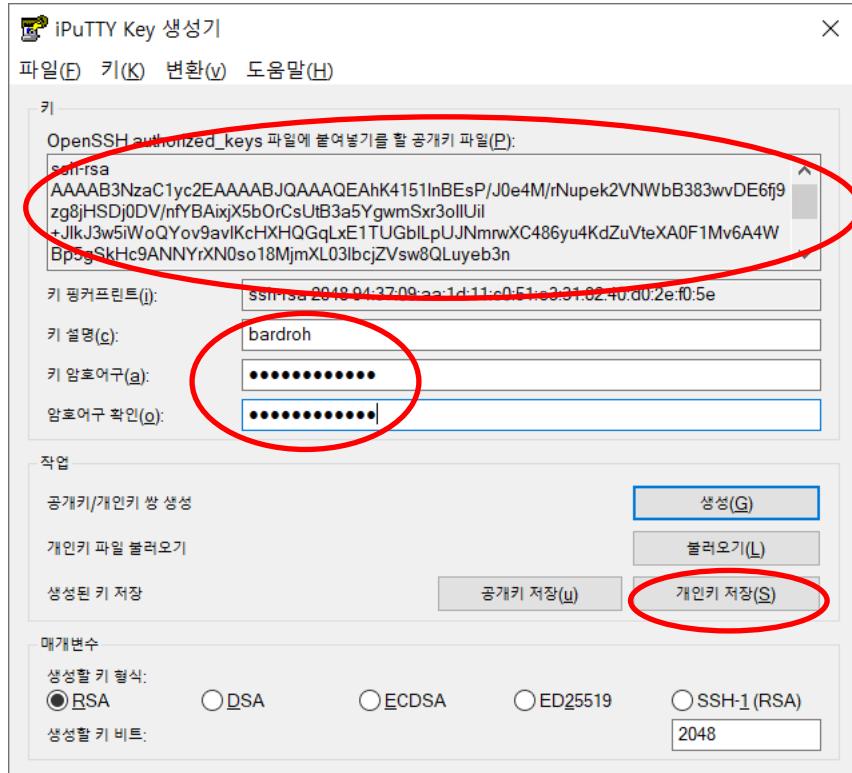
2단계

Google 계정에 액세스할 수 있도록 이 정보를 최신 상태로 유지하고 로그인 옵션을 추가하세요.

- |            |               |   |
|------------|---------------|---|
| 패스키 및 보안 키 | 보안 키 추가       | > |
| Google 메시지 |               | > |
| OTP        | 1분 전에 추가됨     | > |
| 전화번호       | 010-4368-4787 | > |
| 복구 코드      | 복구 코드 받기      | > |

- AWS에서는 keypair를 자동으로 만들어주지만, GCE에서는 우리가 keypair를 만들어서 업로드해야 함
- keypair 생성은 puttygen.exe를 사용
- 로그인하는 계정명과 암호를 지정하여 생성한 후 public key는 GCE의 메타정보(SSH 키)에 붙여넣고 private key는 putty의 세션 정보에 지정함





Google Cloud Platform × +

console.cloud.google.com/compute/metadata/sshKeys?project=nice-dispatcher-322407

구글 앱 Gmail YouTube 다른 기기의 템 지도 뉴스

무료 평가판 상태: 크레딧은 ₩346,326.00, 무료 평가판 기간은 91일 남았습니다. 완전한 계정을 사용하면 Google Cloud Platform의 모든 기능에 무제한 액세스할 수 있습니다.

닫기 활성화

Google Cloud Platform My First Project 제품 및 리소스 검색

Compute Engine 메타데이터

상태 확인 SSH 키 설정

VM Manager

OS 패치 관리

OS 설정 관리

설정

메타데이터

영역

네트워크 엔드포인트 그룹

작업

Marketplace

출시 노트

메타데이터 SSH 키 설정

이 프로젝트의 모든 인스턴스는 이러한 SSH 키를 상속받습니다. 자세히 알아보기

사용자 이름	키
bard	ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDF...,"expireOn":"2021-08-09T07:52:28+0000")
bard	ecdsa-sha2-nistp256 AAAE2VjZHNhLXNoYTIt...,"expireOn":"2021-08-09T07:52:24+0000")
bard	ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQFWS...,"expireOn":"2021-08-09T07:52:02+0000")
bard	ecdsa-sha2-nistp256 AAAE2VjZHNhLXNoYTIt...,"expireOn":"2021-08-09T07:51:58+0000")

동등한 REST

Google Cloud Platform × +

console.cloud.google.com/compute/metadata/sshKeys?project=nice-dispatcher-322407

구글 앱 Gmail YouTube 다른 기기의 템 지도 뉴스

무료 평가판 상태: 크레딧은 ₩346,326.00, 무료 평가판 기간은 91일 남았습니다. 완전한 계정을 사용하면 Google Cloud Platform의 모든 기능에 무제한 액세스할 수 있습니다.

닫기 활성화

## Google Cloud Platform My First Project 제품 및 리소스 검색

### Compute Engine 메타데이터

상태 확인 bard

VM Manager bard

OS 패치 관리 bard

OS 설정 관리 bard

설정 bard

메타데이터

영역 bard

네트워크 엔드포인트 그룹 bard

작업 bard

Marketplace bard

출시 노트

+ 항목 추가

저장 취소

```
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbm1zdHAYNTYAAAIBm1zdHAYNTYAAABBBHjg8R76ge
at*xzlnntpZYCgIBz0LV48WQ11KTKWc3cd0a4FN2Y8Z0alHm/J4+u6zVmgVmflSLTzFeRVjxQMI= goo
gle-ssh {"userName": "bard@hitel.net", "expireOn": "2021-08-09T07:52:24+0000"}  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQWscWB1QHg/3Ea5LYpro2rFf15/Sti3MCZ3NB5pVEGff
dvlnZbA85g/F1GCXR4n4pl8t45YQ1qEDK1Xfn+cTaQMx0aQMrkeWfwuZxdYv0IyLHBHiWAVZYH6R5k4HVz
ZxvdLyNv49zi22xJqxSCkydEZnRxNl95054XFn0IjEGQ1/tTYW1zDtUyGHT13T0J5gSwIr58ozJJPtE
LmzYnuTurWMcWtgxdrbyo3gX4LQKtruQRyknpjc1FgtbtBdecZG9F6akr+9V9/1py39GK0bySNTXieZsv
Ptukt3tvDtv488wxhaPTEGQIY6WLHhQmtjZjYN+1FDMyWmFU0s= google-ssh {"userName": "bard@hitel.net", "expireOn": "2021-08-09T07:52:02+0000"}  
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbm1zdHAYNTYAAAIBm1zdHAYNTYAAABBBAxNJLT+y7
EUbxW61bmxBPr0SbzsrXphGrDAmxh1t6IQwl1SOJteRh0h2/zA3gTh59pGRjY+qpadycyA9KX+k= goo
gle-ssh {"userName": "bard@hitel.net", "expireOn": "2021-08-09T07:51:58+0000"}
```

Google Cloud Platform × +

console.cloud.google.com/compute/metadata/sshKeys?project=nice-dispatcher-322407

구글 앱 Gmail YouTube 다른 기기의 탭 지도 뉴스

무료 평가판 상태: 크레딧은 ₩346,326.00, 무료 평가판 기간은 91일 남았습니다. 완전한 계정을 사용하면 Google Cloud Platform의 모든 기능에 무제한 액세스할 수 있습니다.

닫기 활성화

## Google Cloud Platform My First Project 제품 및 리소스 검색

### Compute Engine 메타데이터

상태 확인 bard

VM Manager bard

OS 패치 관리 bard

OS 설정 관리 bard

설정

- 메타데이터
- 영역
- 네트워크 엔드포인트 그룹
- 작업
- Marketplace

출시 노트

저장 취소

```
ssh-rsa AAAAB3NzaC1yc2EAAQABAAQABAWscWb1Qhg/3Ea5LYpro2rFf15/Sti3MCZ3NB5pVEGffe
dvlnZba8sg/F1GCRAn4pU8t45YQ1qEDK1Xfn+cTaQMx0aMrkefwuZxdYv01yLHBHlWAVZYH6R5k4Hvz
ZxvdyLynV49zi22xJqXsSCKydEznRxNL95054xFn0IjEGQ1/tTYW1zDTuYGHt13T0J5gSwIr58ozJJPtE
LmzYnuTurWMcWtxd1rbyo3q4LQKtRuRyknpcjcfGtbt8decZG9f6akr+9V9/ipy39GKDbySNTXieZsv
PtUkt3tvDTv488wxhaPtfQGY6W.LhQmtjzjYN+1FDMyWmFu0s= google-ssh {"userName": "bard@hite
tel.net", "expireOn": "2021-08-09T07:52:02+0000"}
```

```
ecdsa-sha2-nistp256 AAAE2VjZHNhLXNoYTItbm1zdHAYNTYAAAIBmlzdHAYNTYAAABBBANJLT+y7
ELbxW6lbmx8Pr05bzsrXphGrVDAmxh1t6IQw1ISOItErh0h2/zA3gTh59pGrjY+qpadycyA9KX+k= goo
gle-ssh {"userName": "bard@hite
tel.net", "expireOn": "2021-08-09T07:51:58+0000"}
```

- VS Code 설치, Remote-SSH 확장 설치
- ssh-keygen으로 key 생성(/home/ubuntu/rsa\_for\_gce)
- rsa\_for\_gce.pub을 메타정보에 붙여 넣음
- VM 생성, 이후 ssh -i rsa\_for\_gce ubuntu@<IP>로 접속하여 확인
- rsa\_for\_gce ubuntu를 C:\temp에 복사
- PowerShell 실행 후 다음 명령 수행

```
$path = "C:\temp\rsa_for_gce"  
icacls $path /inheritance:r  
icacls $path /grant:r "$($env:username):(R)"
```

- VSCode 원격 설정 수정

Host 34.47.97.95

HostName 34.47.97.95

User ubuntu

IdentityFile c:/temp/rsa\_for\_gce

- 이후 로컬과 동일하게 python, jupyter 확장 설치 후 사용

- LLM들을 사용하는데 필수적인 라이브러리들 중 하나가 Huggingface
- Huggingface의 transformers 패키지에는 현재 공개된 대부분의 오픈소스 아키텍처에 대한 구현이 포함되어 있음: GPT, GPT2, LLaMa(2/3), Mistral 등등
- 이렇게 구현된 아키텍처로 학습한 모델들을 올려놓고 push / pull로 사용할 수 있도록 만든 registry가 Huggingface hub: 대부분의 잘 알려진 오픈소스 LLM들은 여기서 배포
- Model 뿐만 아니라, Dataset들도 배포하며, Space를 통해 이 Model들을 바로 테스트해볼 수도 있음
- 이 과정에서는 Huggingface의 잘 알려진 LLM들을 download하여 로드하고, inference하는 작업들을 테스트해볼 것: 다양한 모델들을 사용할 수 있음, download에는 시간이 걸림

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin
```

```
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-600
```

```
wget https://developer.download.nvidia.com/compute/cuda/12.5.0/local_installers/cuda-repo-ubuntu2204-12-5-local_12.5.0-555.42.02-1_amd64.deb
```

```
sudo dpkg -i cuda-repo-ubuntu2204-12-5-local_12.5.0-555.42.02-1_amd64.deb
```

```
sudo cp /var/cuda-repo-ubuntu2204-12-5-local/cuda-*keyring.gpg /usr/share/keyrings/
```

```
sudo apt-get update
```

```
sudo apt-get -y install cuda-toolkit-12-5
```

```
sudo apt-get install -y nvidia-driver-555-open
```

```
sudo apt-get install -y cuda-drivers-555
```

```
wget https://developer.download.nvidia.com/compute/cudnn/9.2.0/local_installers/cudnn-local-repo-ubuntu2204-9.2.0_1.0-1_amd64.deb  
sudo dpkg -i cudnn-local-repo-ubuntu2204-9.2.0_1.0-1_amd64.deb  
sudo cp /var/cudnn-local-repo-ubuntu2204-9.2.0/cudnn-*-keyring.gpg /usr/share/keyrings/  
sudo apt-get update  
sudo apt-get -y install cudnn  
sudo apt-get -y install cudnn-cuda-12
```

```
!pip install torch transformers accelerate huggingface_hub[cli]
```

```
!huggingface-cli --help
```

```
!huggingface-cli download 42dot/42dot_LLM-SFT-1.3B
```

```
import torch
```

```
from transformers import AutoTokenizer, AutoModelForCausalLM
```

```
model_id = "42dot/42dot_LLM-SFT-1.3B"
```

```
tokenizer = AutoTokenizer.from_pretrained(model_id)
```

```
model = AutoModelForCausalLM.from_pretrained(model_id, device_map={"":0})
```

```
!nvidia-smi
```

- 일반적으로 LLM은 32bit 실수로 표현되는 parameter들의 집합
- 즉 parameter 규모가 7B라면,  $7B * 32bit = 7B * 4 = 32GB$ 의 용량
- 이를 half-precision인 16bit로 저장하면 절반인 16G가 됨
- 점점 bit수를 낮춰가면 성능이 떨어질 가능성은 높지만, 그에 비례해 용량과 메모리 필요량이 줄어들게 되고, 연산량도 낮아져서 token 생성 속도가 빨라짐
- 일정 범위의 수를 하나의 수로 간주하여 bit수를 낮추는 기법을 양자화(quantization)이라 하며, 현재 1bit 양자화(1.5bit 정도를 사용)까지 나와 있는 상태
- 양자화를 하게 되면 성능은 떨어지지만(PPL값을 봄), 모델의 기본 성능이 좋거나 크기가 크면 양자화를 하더라도 성능이 크게 낮아지지 않는 것으로 알려져 있음
- 16bit 모델과 비교하여 거의 유사한 수준의 PPL을 나타내는 양자화는 4bit 정도가 한계
- 양자화 라이브러리는 bitsandbytes, AWQ, gguf, GPTQ 등 다양함

<b>type</b>	<b>ppl increase</b>	<b>ppl 13b to 7b %</b>	<b>file size</b>
q2_k	0.8698	>100%	2.67GB
q3_ks	0.5505	84.4%	2.75GB
q3_km	0.2437	37.4%	3.06GB
q3_kl	0.1803	27.6%	3.35GB
q4_0	0.2499	38.3%	3.5GB
q4_1	0.1846	28.3%	3.9GB
q4_ks	0.1149	17.6%	3.56GB
q4_km	0.0535	8.2%	3.80GB
q5_0	0.0796	12.2%	4.3GB
q5_1	0.0415	6.36%	4.7GB
q5_ks	0.0353	5.41%	4.33GB
q5_km	0.0142	2.18%	4.45GB
q6_k	0.0044	0.67%	5.15GB
k8_0	0.0004	0.061%	6.7GB

- Meta의 llama 모델은 처음에 huggingface 포맷으로 배포되었고, 따라서 사용하려면 huggingface의 transformers 라이브러리가 필요했음: 당연히 언어는 python을 요구
- llama.cpp는 이 llama 모델의 레이어를 pure C/C++로 구현하는 것을 목표하여 개발되었으나, 현재는 대부분의 LLM들을 모두 지원하고 있음
- llama.cpp의 장점은 여러가지가 있는데, 일단 pure C/C++로 구현되었으므로 속도가 빠르고, GPU가 없어도 구동된다는 것: 현재는 CUDA나 ROCm, Apple Silicon 등 다양한 백엔드를 지원하여 가속할 수 있음
- 더불어 여러가지 양자화 옵션을 사용할 수 있으므로, 2bit나 3bit부터 8bit int까지 상황에 따라 다른 양자화 모델을 사용할 수 있는 장점이 있음: ggml로 시작하여 현재는 gguf라는 포맷을 사용

```
sudo apt-get install cmake ccache  
sudo apt-get install nvidia-cuda-toolkit # CUDA  
cd ~/work  
git clone https://github.com/ggerganov/llama.cpp.git  
cd llama.cpp  
cmake -B build # CPU  
cmake -B build -DLLAMA_CUDA=ON # CUDA  
cmake --build build --config Release  
python3 -m pip install -r requirements.txt
```

## | gguf 변환 / quantize

---

```
python3 convert-hf-to-gguf.py ~/.cache/huggingface/hub/models--google--gemma-1.1-2b-
it/snapshots/bf4924f313df5166dee1467161e886e55f2eb4d4/ --outtype f16
ls ~/.cache/huggingface/hub/models--google--gemma-1.1-2b-
it/snapshots/bf4924f313df5166dee1467161e886e55f2eb4d4/
mv ~/.cache/huggingface/hub/models--google--gemma-1.1-2b-
it/snapshots/bf4924f313df5166dee1467161e886e55f2eb4d4/ggml-model-f16.gguf
~/work/google-gemma-1.1-2b-F16.gguf
cd build/bin
./quantize ~/work/google-gemma-1.1-2b-F16.gguf ~/work/google-gemma-1.1-2b-Q4_K_M.gguf
Q4_K_M
./server -m ~/work/google-gemma-1.1-2b-Q4_K_M.gguf # -ngl <GPU 레이어수>
```

```
pip3 install vllm
```

```
pip3 install autoawq
```

```
pip3 install fschat # chat completion을 위해 필요
```

```
python3 -m vllm.entrypoints.openai.api_server --model 42dot/42dot_LLM-SFT-1.3B  
--served-model-name 42dot_LLM-SFT-1.3B --dtype float16 --port 18888
```

```
curl http://localhost:18888/v1/chat/completions ¶
```

```
-H "Content-Type: application/json" ¶
```

```
-d '{
```

```
    "model": "42dot_LLM-SFT-1.3B",
```

```
    "messages": [{"role": "user", "content": "베트남 사파로 9박 10일 여행갈건데 매일의 상세한 일정을 짜줘."}],
```

```
    "temperature": 0.7
```

```
}
```

<https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html>

```
sudo su - root
```

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o  
/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg ¶  
¶ && curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-toolkit.list | ¶  
sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg]  
https://#g' | ¶  
sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y nvidia-container-toolkit
```

```
curl -fsSL https://get.docker.com -o install-docker.sh  
cat install-docker.sh  
sh install-docker.sh --dry-run  
sudo sh install-docker.sh  
sudo nvidia-ctk runtime configure --runtime=docker  
sudo systemctl restart docker  
docker run --gpus all --shm-size 1g -p 8080:80 -v ~/.cache/huggingface/hub:/data ghcr.io/huggingface/text-generation-inference:2.0 --model-id $model  
curl 127.0.0.1:8080/generate_stream ¶  
-X POST ¶  
-d '{"inputs":"What is Deep Learning?","parameters":{"max_new_tokens":20}}' ¶  
-H 'Content-Type: application/json'
```

```
cd ~/work  
git clone https://github.com/PygmalionAI/aphrodite-engine.git  
cd aphrodite-engine  
../update-runtime.sh  
../runtime.sh # 환경 진입  
pip install packaging ninja  
pip install flash-attn --no-build-isolation  
aphrodite -h  
aphrodite run 42dot/42dot_LLM-SFT-1.3B  
aphrodite run --load-in-4bit 42dot/42dot_LLM-SFT-1.3B  
aphrodite run --device cpu --load-in-4bit 42dot/42dot_LLM-SFT-1.3B
```

- 현대적인 LLM의 성능은 매우 우수하지만 LLM만으로 AI 서비스를 만들 수는 없음: 실제 서비스 구축을 위해서는 LLM 외 다른 요소들이 상당히 많이 필요하다.
- AI 서비스를 구축하는 방법에 대해서는 비교적 잘 알려져 있는 편이나, 서비스하기에 충분한 품질을 얻는 것은 생각보다 많이 어렵다. 대기업들(예. SK 텔레콤)도 비슷하게 헤매고 있다.
- AI 서비스를 구축할 때는 반드시 원하는 성능 지표를 설정하고, 측정을 위한 데이터셋을 준비해야 한다.
- 예상외로 LLM을 직접 운영할 때 들어가는 비용은 높다. 반드시 로컬 LLM을 사용해야 하는 이유가 없다면 대형 IT업체들의 서비스를 사용하는 편이 저렴한 경우가 많다: OpenAI나 claude, 특히 중국 업체들의 경우 원가에도 한참 미치지 못하는 비용으로 서비스하고 있음
- 로컬 LLM을 사용해야 하는 이유는 대부분 보안, 규정: 공공기관, 금융권 등. 더불어 자체 도메인에 더 적합한 미세조정 모델을 사용하려고 하는 경우도 포함됨. 사용량이 많아서 로컬 LLM을 고려하는 경우도 드물게 있음

- ollama는 llama 계열을 포함한 오픈소스 LLM들을 구동하기 위한 inference 전용 서비스
- llama 계열 뿐만 아니라 phi-3, gemma, mistral 등 다양한 LLM들을 모두 지원
- Linux, Windows, Mac을 모두 지원하며, 안드로이드나 라즈베리파이에서도 구동됨
- CUDA 뿐만 아니라, AMD Radeon 계열 가속을 지원하며, GPU가 없는 경우 CPU만으로 inference할 수 있음
- 'ollama pull' 명령으로 ollama.com에서 제공하는 이미지들을 가져와서 사용가능
- 또한 Dockerfile과 유사한 modelfile을 사용하여 custom 이미지를 만들 수 있으며, 일단 만들어진 이미지는 언제든지 모델명을 지정해서 사용할 수 있음
- 'ollama run'으로 직접 LLM에 프롬프트를 줄 수도 있고, 전용 포트(11434)에 REST API call을 전달해도 되며, OpenAI와 호환되는 API(11434/v1)에 openai call을 던져도 됨
- 구동용 이미지는 gguf를 사용하는 것이 일반적임



## Download Ollama



macOS



Linux



Windows

Install with one command:

```
curl -fsSL https://ollama.com/install.sh | sh
```

[View script source](#) • [Manual install instructions](#)

While Ollama downloads, sign up to get  
notified of new updates.

Get updates



ollama list # 현재 시스템에 설치된 모델들을 리스트업

ollama pull <모델명:태그> # ollama.com의 repo로부터 모델을 가져옴

ollama push <모델명:태그> # ollama.com의 repo로부터 모델을 보냄

ollama create -f <모델파일> <모델명:태그> # 지정된 모델파일로 ollama 이미지를 로컬에 생성

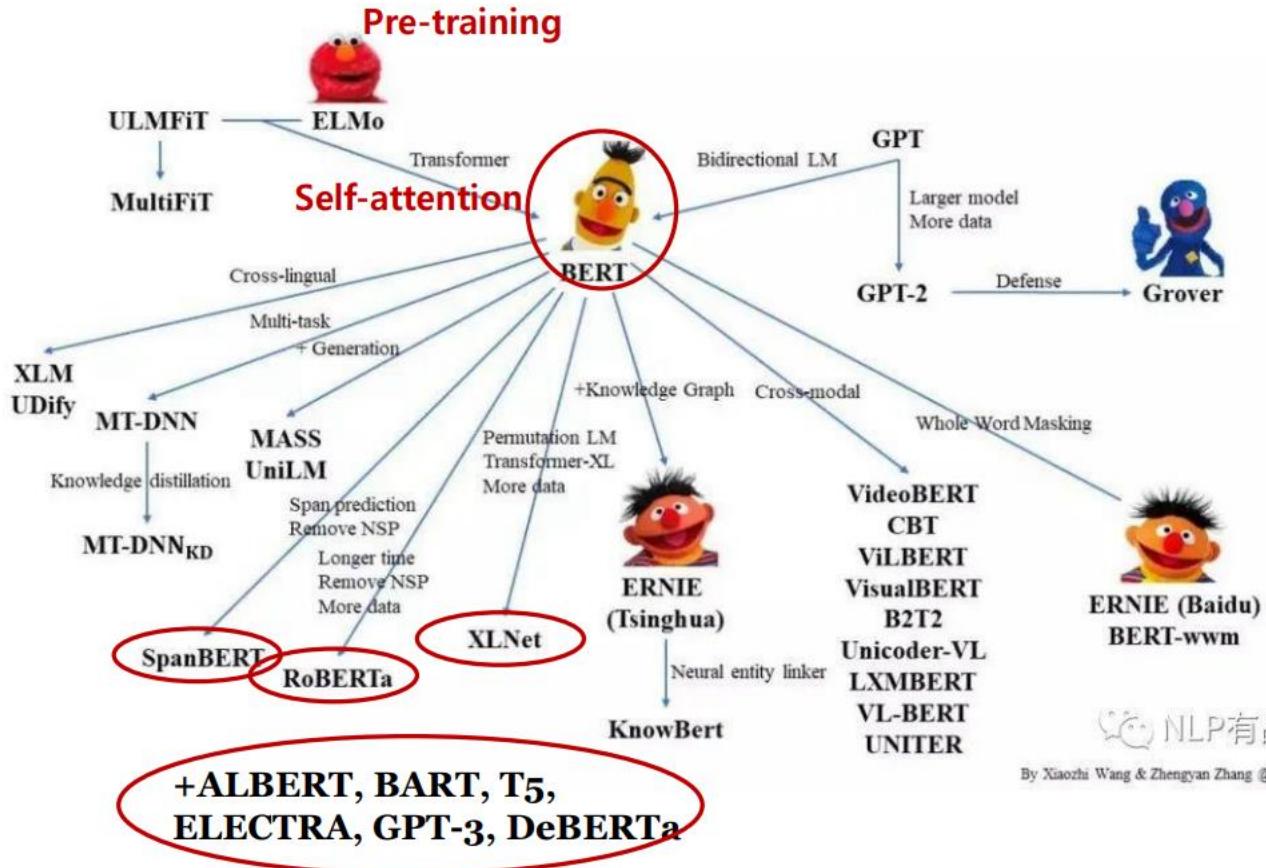
ollama run <모델명:태그> # 모델을 실행하고 사용자의 프롬프트를 받음

- OpenAI의 모델들을 사용할 때는 python의 openai 패키지를 쓰는 경우가 많음
- 최근 나오는 Serving framework들은 모두 OpenAI API와 일부라도 호환되는 기능을 포함하고 있음:  
OpenAI 생태계가 가장 크기 때문에, 이런 호환 기능을 넣으면 OpenAI로부터의 전환이 수월해짐
- ollama는 자체 API와 더불어, OpenAI와 호환 되는 API를 같이 제공
- 이 과정에는 openai 패키지를 설치하고 OpenAI와 같은 방법으로 ollama 모델들을 호출해볼 것

```
pip install openai
```

# The Neural History of Natural Language Processing

- 
- A vertical timeline chart showing the evolution of neural language processing models from 2001 to 2018. The timeline is represented by a black vertical line with white circular markers at each event point. The years are listed to the left of the markers, and the corresponding model names are listed to the right.
- | Date | Milestone                   |
|------|-----------------------------|
| 2001 | Neural language models      |
| 2008 | Multi-task learning         |
| 2013 | Word embeddings             |
| 2013 | Neural networks for NLP     |
| 2014 | Sequence-to-sequence models |
| 2015 | Attention                   |
| 2015 | Memory-based networks       |
| 2018 | Pretrained language models  |

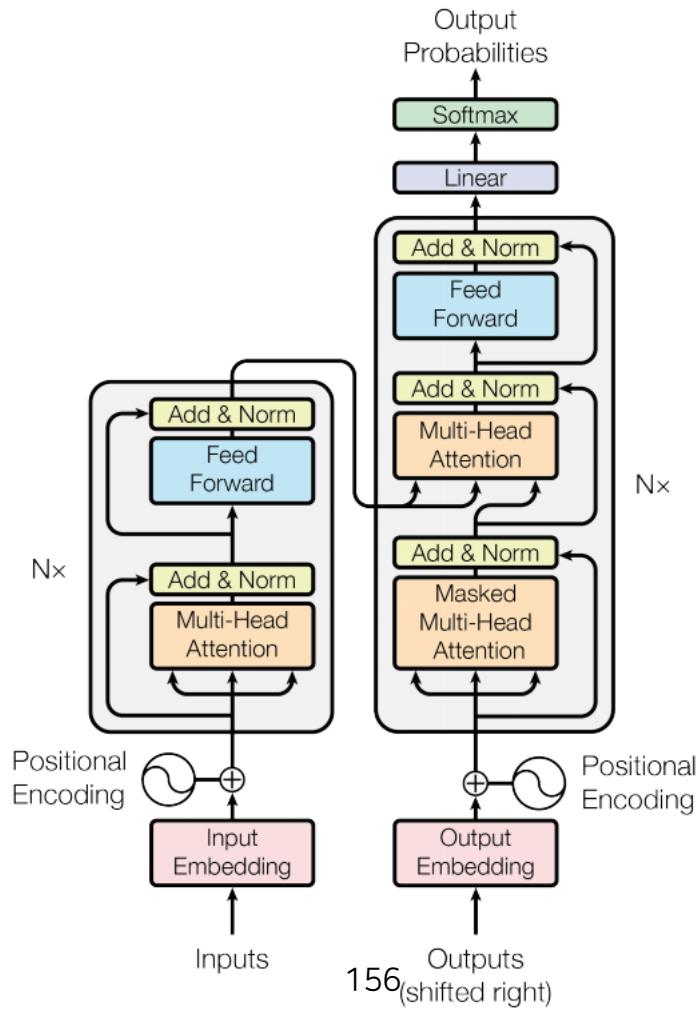


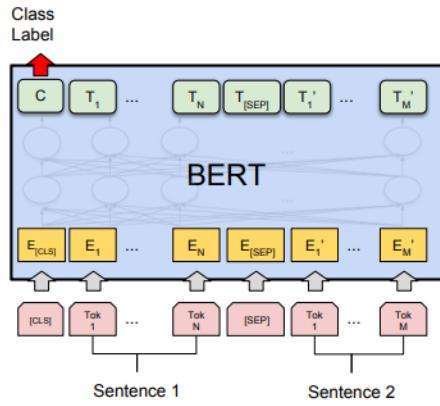
Transformer(2017, Google)

- Encoder-Decoder 구조로 RNN 없이 Attention만으로 구현
- Positional Encoding이 포함됨: 같은 token이더라도 위치에 따라 다른 의미
- Multi-head attention: 동시에 여러 개의 attention을 계산하여 효율을 높임

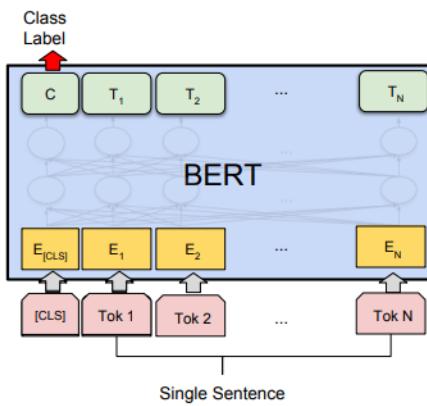
BERT(2018, Google)

- Transformer의 Encoder만을 사용
- 처음부터 모든 것을 학습하지 않고, pre-trained된 모델을 fine-tuning하는 방식
- 어떻게 학습시키느냐에 따라 Sentence Pair Classification, Single Sentence Classification, Question Answering, Single Sentence Tagging 등 다양한 Task를 수행할 수 있는 다목적 모델

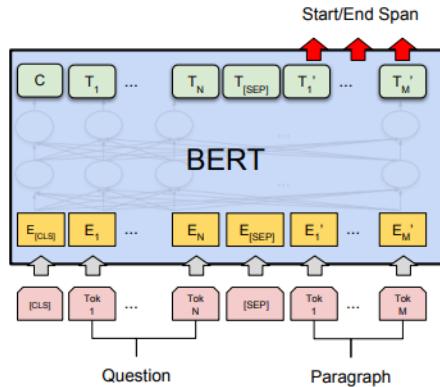




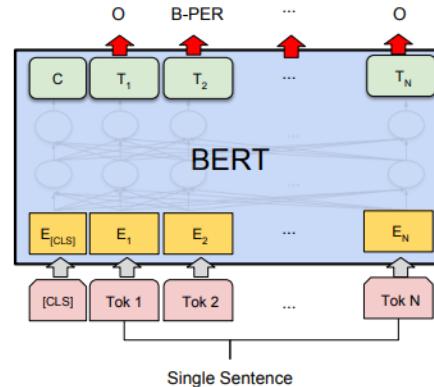
(a) Sentence Pair Classification Tasks:  
MNLI, QQP, QNLI, STS-B, MRPC,  
RTE, SWAG



(b) Single Sentence Classification Tasks:  
SST-2, CoLA



(c) Question Answering Tasks:  
SQuAD v1.1

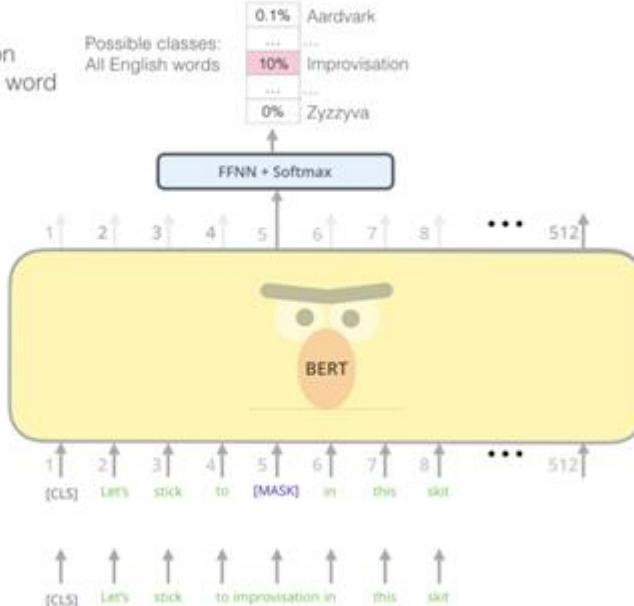


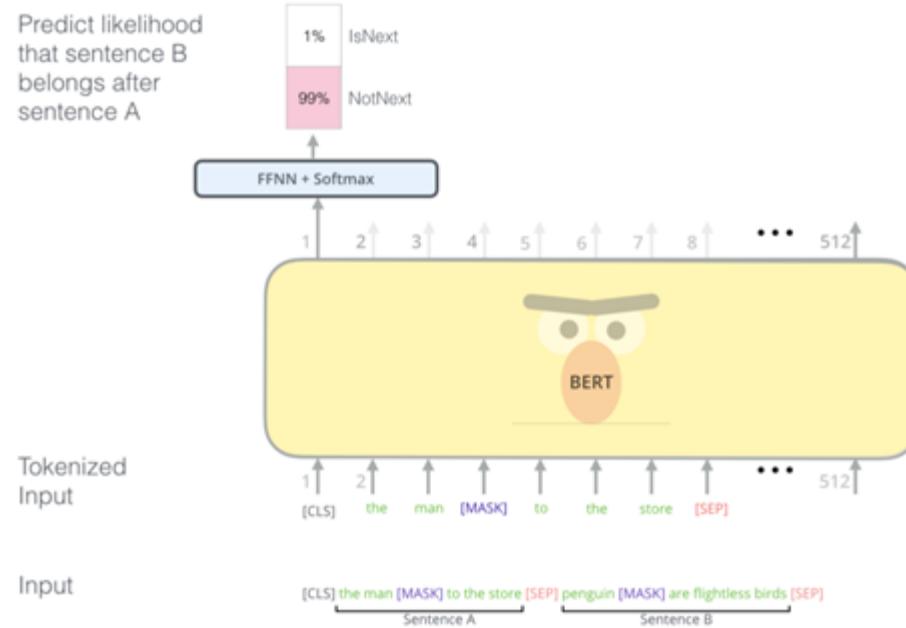
(d) Single Sentence Tagging Tasks:  
CoNLL-2003 NER

Use the output of the masked word's position to predict the masked word

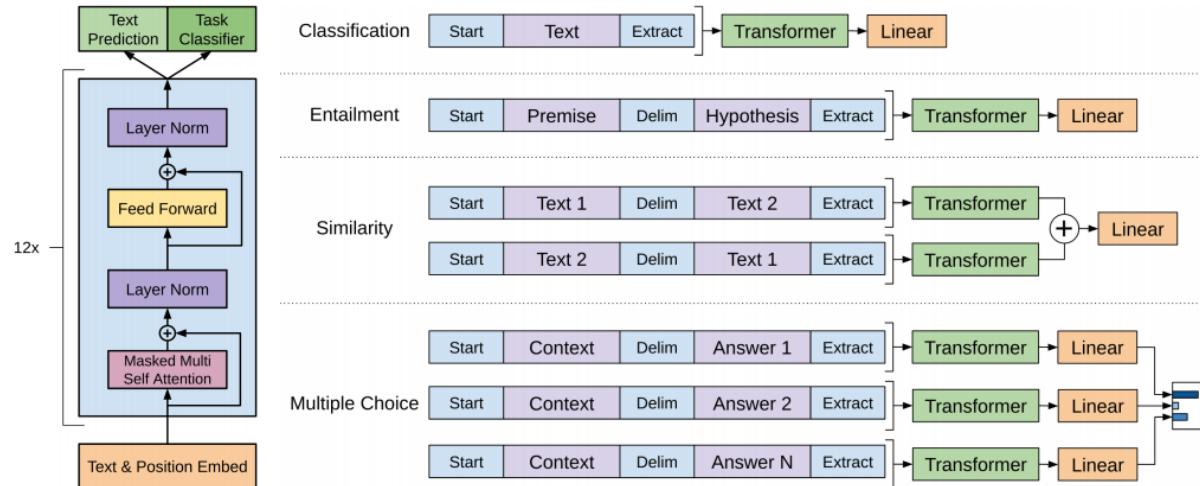
Randomly mask 15% of tokens

Input

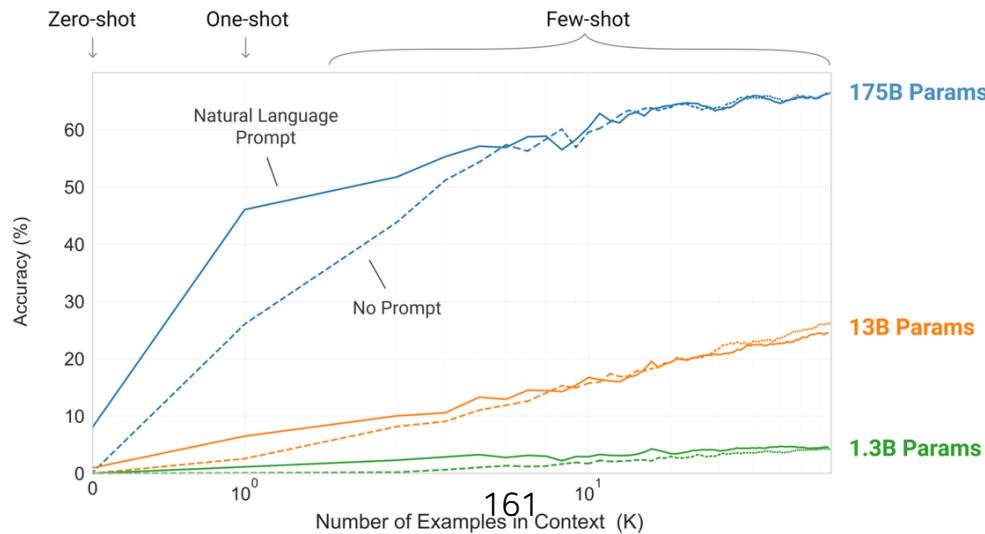




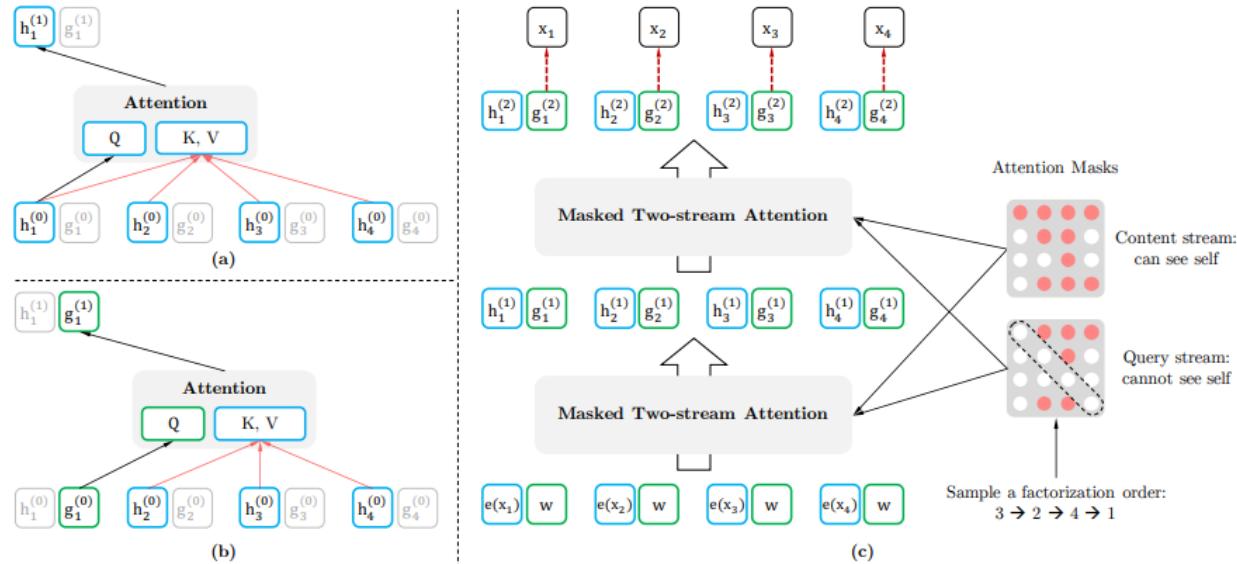
- 2018년 OpenAI가 제안한 모델, “Generative Pre-Training” 모델
- Transformer의 Encoder만을 사용하는 BERT에 비해, Transformer의 Decoder만을 사용
- BERT와 동일하게 다목적으로 사용하기 위한 pre-trained 모델이나, 주로 문장을 생성하는 Task에서 좋은 성과를 낸다고 알려져 있음



- GPT를 발전시켜서 더욱 많은 데이터와 파라미터를 가진 초거대 자연어 처리 모델
- GPT-2까지는 오픈소스로 공개되어 있으나, GPT-3부터는 공개되지 않음
- GPT-3의 경우 1750억개의 파라미터를 사용하며(GPT-2의 100배 이상), 3000억개의 데이터로 사전학습 함: 훈련비용은 150억원, 모델이 공개되더라도 이미 일반인이 학습시킬 수 있는 규모가 아님

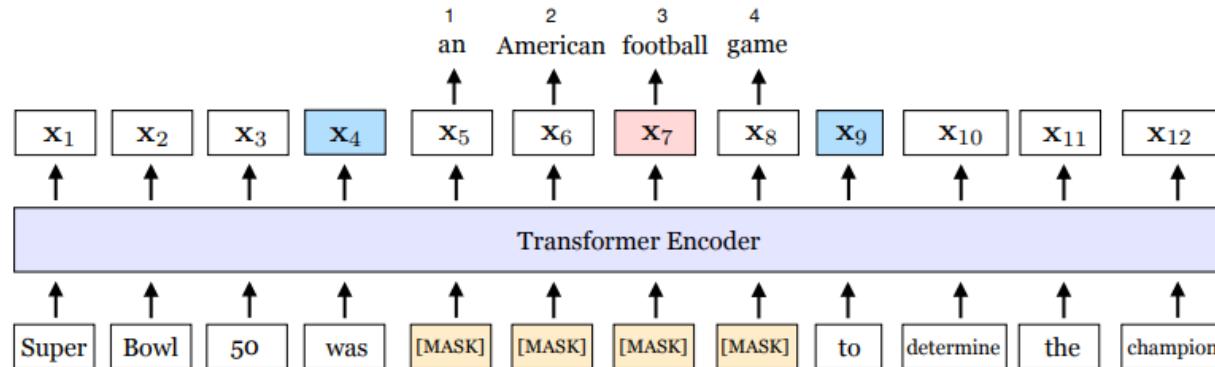


- 2019년 6월 카네기 멜론 대학과 Google이 제안한 모델
- AE 방식의 BERT의 한계점을 극복하기 위해 AE+AR 제안



- 2019년 7월 워싱턴대, 프린스턴대, Allen AI Lab, Facebook가 공동으로 개발한 모델
- BERT의 pre-training 방식을 개선

$$\begin{aligned}\mathcal{L}(\text{football}) &= \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football}) \\ &= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)\end{aligned}$$



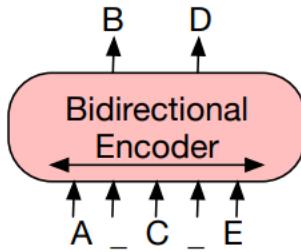
### RoBERTa

- 2019년 7월 워싱턴대, Facebook이 제안
- BERT의 최적화 모델(A Robustly Optimized BERT Pretraining Approach)

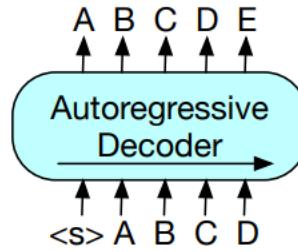
### ALBERT

- 2019년 9월 Google이 제안
- BERT를 더 크게 만들기 위해 모델을 효율화한 형태(A Lite BERT for Self-supervised Learning of Language Representations)

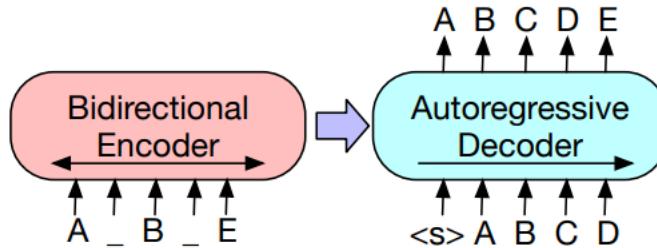
- 2019년 10월 Facebook이 제안
- 새로운 AE+AR 구조를 적용
- 논문명은 BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension 이지만 BART라는 이름은 The Bidirectional and Auto-Regressive Transformer의 줄임말



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

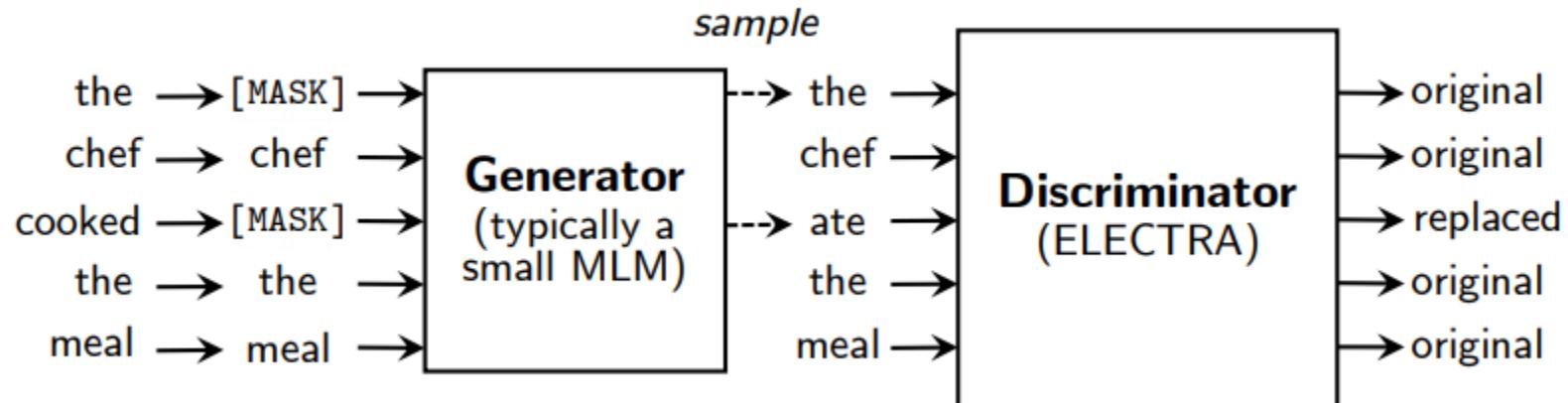


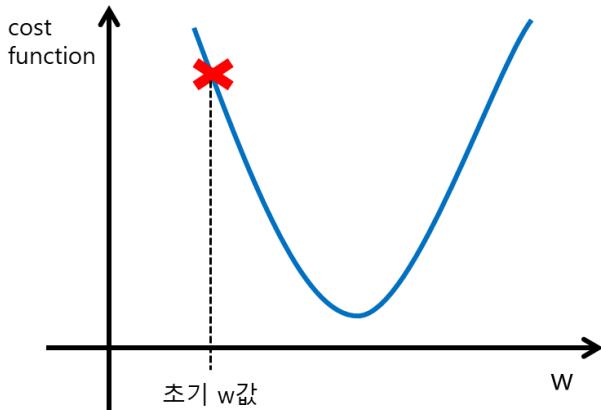
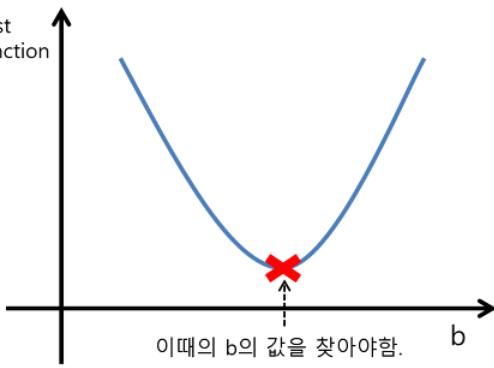
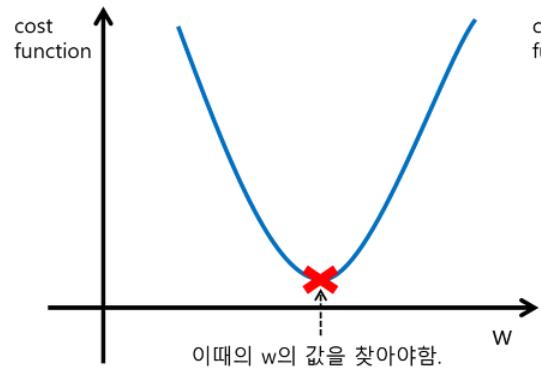
(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



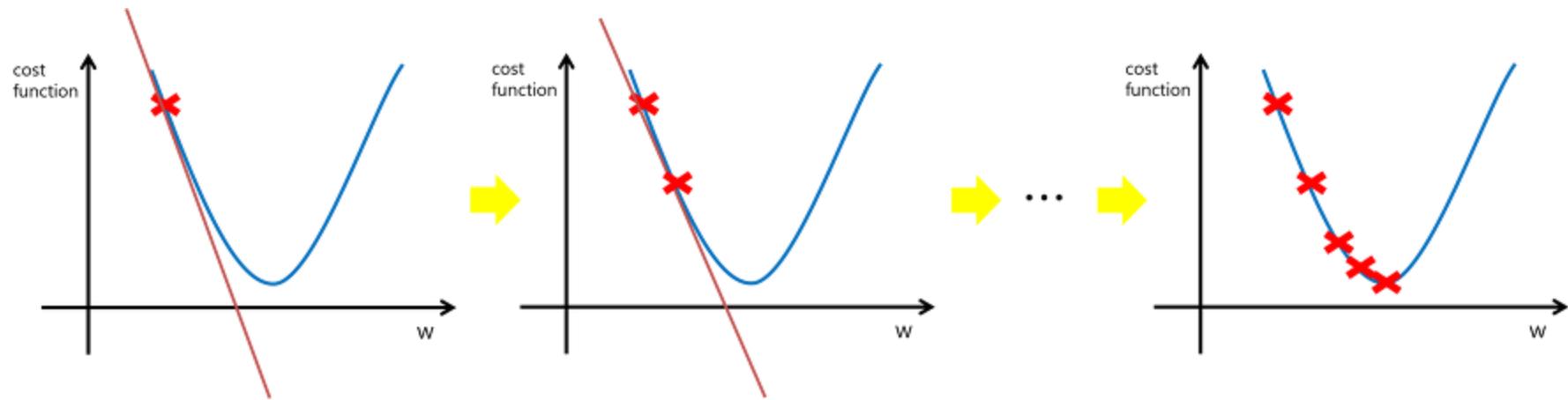
(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitrary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

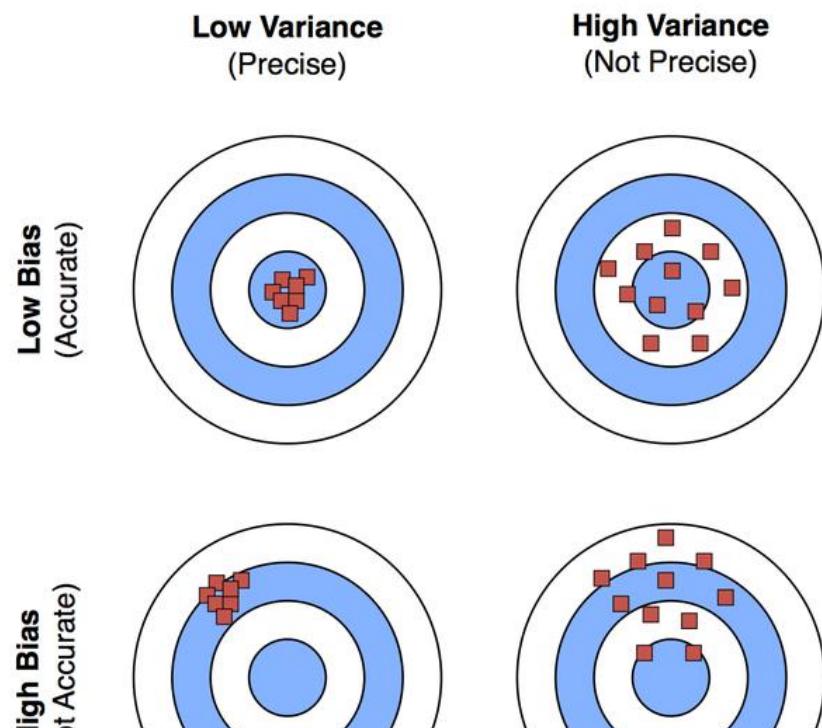
- 2020년 3월 스탠포드와 Google이 제안
- MLM 학습시의 비효율성을 개선하고 GAN-like 구조를 적용(핵심)



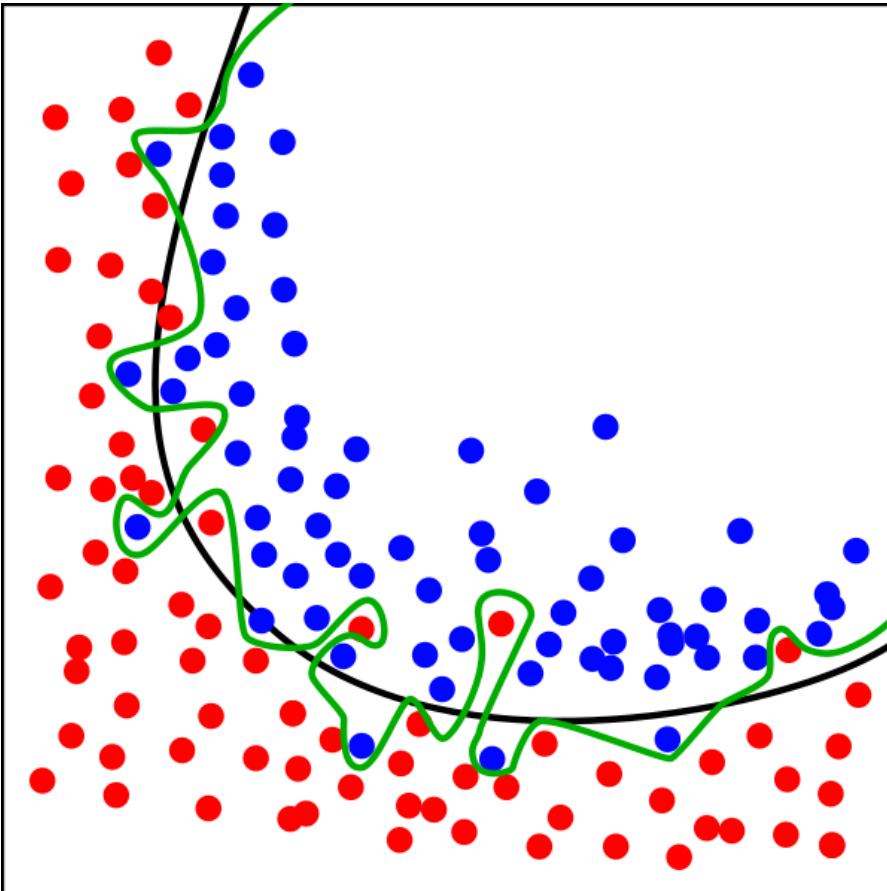


## | 경사하강법(Gradient Descent)

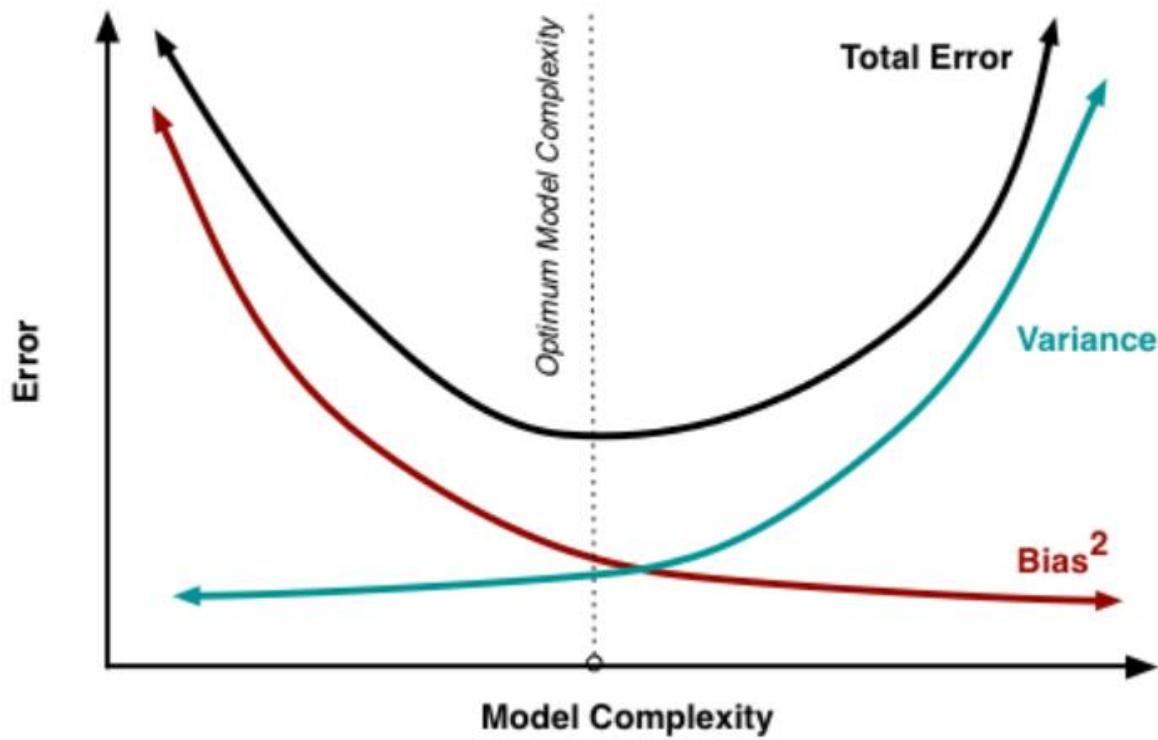




This work by Sebastian Raschka is licensed under a  
Creative Commons Attribution 4.0 International License.



$$\begin{aligned}
E[(y - \hat{f})^2] &= E[y^2 + \hat{f}^2 - 2y\hat{f}] \\
&= E[y^2] + E[\hat{f}^2] - E[2y\hat{f}] \\
&= \text{Var}[y] + E[y]^2 + \text{Var}[\hat{f}] + E[\hat{f}]^2 - 2f E[\hat{f}] \\
&= \text{Var}[y] + \text{Var}[\hat{f}] + (f^2 - 2f E[\hat{f}] + E[\hat{f}]^2) \\
&= \text{Var}[y] + \text{Var}[\hat{f}] + (f - E[\hat{f}])^2 \\
&= \sigma^2 + \text{Var}[\hat{f}] + \text{Bias}[\hat{f}]^2
\end{aligned}$$



### Step 1: **SFTTrainer**

Train your model on your favorite dataset

```
from trl import SFTTrainer

trainer = SFTTrainer(
    "facebook/opt-350m",
    train_dataset=dataset,
    dataset_text_field="text",
    max_seq_length=512,
)

trainer.train()
```

### Step 2: **RewardTrainer**

Train a preference model on a comparison data to rank generations from the supervised fine-tuned (SFT) model

```
from trl import RewardTrainer

trainer = RewardTrainer(
    model=model,
    args=training_args,
    tokenizer=tokenizer,
    train_dataset=dataset,
)

trainer.train()
```

### Step 3: **PPOTrainer**

Further optimize the SFT model using the rewards from the reward model and PPO algorithm

```
from trl import PPOConfig, PPOTrainer

trainer = PPOTrainer(
    config,
    model,
    tokenizer=tokenizer,
)

for query in dataloader:
    response = model.generate(query)
    reward = reward_model(response)
    trainer.step(query, response, reward)
```

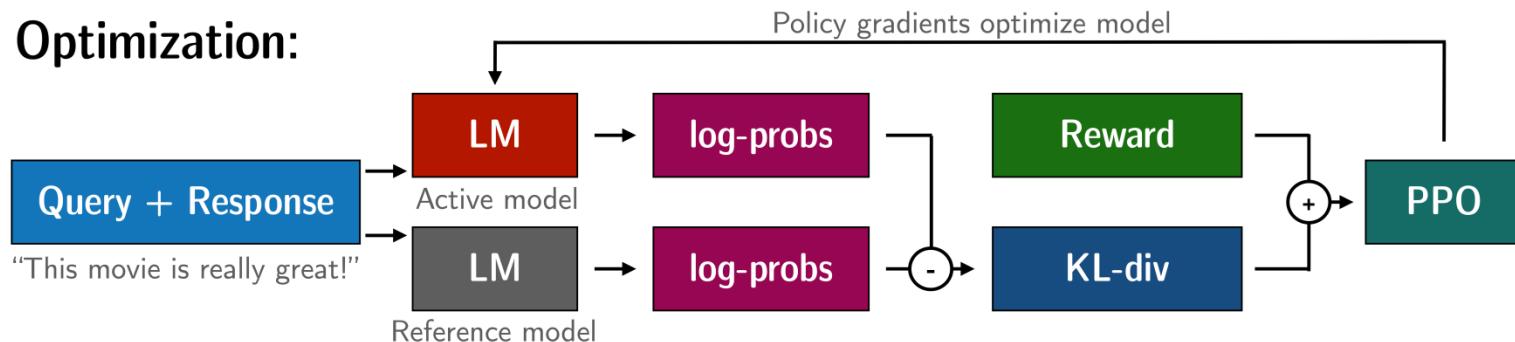
## Rollout:



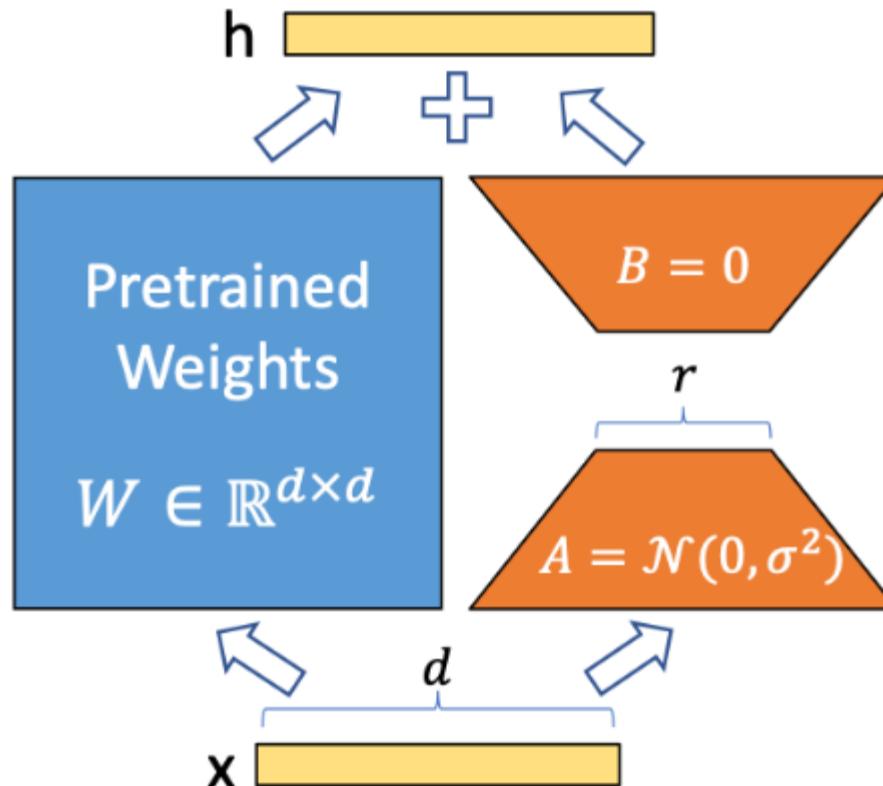
## Evaluation:

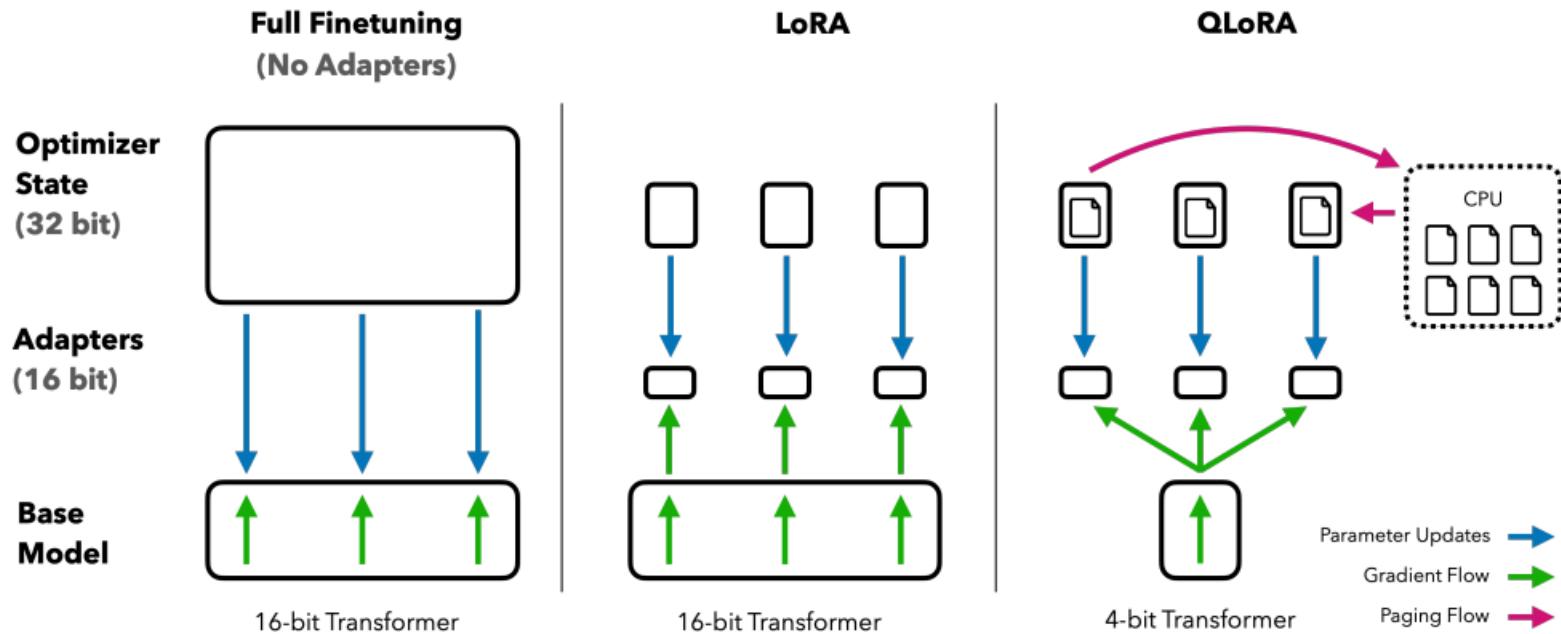


## Optimization:



- 예전에 LLM을 학습할 때는 AutoRegressive로, 출력되는 token과 학습데이터간의 loss를 계산하는 루프를 직접 만들어서 사용했음
- 기술이 발달함에 따라, PEFT나 LoRA 등 다양한 기법들이 등장하게 되었고, 이를 개인이 직접 코드에 구현해 넣기는 부담스러워졌음
- SFTTrainer는 학습에 대한 정형화된 방법론들을 집약하여 만든 트레이너 클래스로, 해당 클래스에 모델과 데이터, 설정을 전달해주면 지정된 방법으로 학습할 수 있음





**Figure 1:** Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

```
lora_config = LoraConfig(  
    r=6,  
    lora_alpha = 8,  
    lora_dropout = 0.05,  
    target_modules=["q_proj", "o_proj", "k_proj", "v_proj", "gate_proj", "up_proj", "down_proj"],  
    task_type="CAUSAL_LM",  
)  
  
bnb_config = BitsAndBytesConfig(  
    load_in_4bit=True,  
    bnb_4bit_quant_type="nf4",  
    bnb_4bit_compute_dtype=torch.float16  
)
```

## | SFTTrainer의 기본 실행

---

```
trainer = SFTTrainer(  
    model=model, # 모델  
    train_dataset=train_data, # 데이터  
    max_seq_length=512, # 최대 텍스트의 길이  
    args=TrainingArguments(  
        output_dir="outputs",  
        max_steps=3000, # 최대 학습하는 step의 수  
        per_device_train_batch_size=1, # 한번에 학습하는 자료의 양  
        gradient_accumulation_steps=4,  
        optim="paged_adamw_8bit",  
        warmup_steps=0.03,  
        learning_rate=2e-4,  
        fp16=True,  
        logging_steps=100, # 매 10 step마다 loss를 출력  
        push_to_hub=False,  
        report_to='none',  
    ),  
    peft_config=lora_config, # LoRA 설정  
    formatting_func=generate_prompt,  
)  
trainer_stats = trainer.train()
```

```
pip install --upgrade pip
pip install "unslloth[cu121-torch230] @ git+https://github.com/unslothai/unslloth.git"
pip install --no-deps xformers trl peft accelerate bitsandbytes
```

```
from unslloth import FastLanguageModel
import torch
max_seq_length = 5632
dtype = None
load_in_4bit = True

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unslloth/gemma-2b-it-bnb-4bit",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit
)
```

```
model = FastLanguageModel.get_peft_model(  
    model,  
    r = 16,  
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",  
                      "gate_proj", "up_proj", "down_proj",],  
    lora_alpha = 16,  
    lora_dropout = 0,  
    bias = "none",  
    use_gradient_checkpointing = "unslot",  
    random_state = 3407,  
    use_rslora = False,  
    loftq_config = None, #  
)
```

```
save_directory = "./trained_model"  
model.save_pretrained(save_directory)  
tokenizer.save_pretrained(save_directory)  
  
from transformers import AutoTokenizer, AutoModelForCausalLM  
import torch  
  
tokenizer = AutoTokenizer.from_pretrained(save_directory)  
model = AutoModelForCausalLM.from_pretrained(  
    save_directory,  
    torch_dtype=torch.bfloat16,  
    device_map="auto"  
)
```

## | Model merge, Huggingface push

---

```
merged_model = model.merge_and_unload()

import os
from huggingface_hub import HfApi, login

login(token=os.getenv("HUGGINGFACE_TOKEN"))

repo_name = "bardroh/gemma_train"
merged_model.push_to_hub(repo_name)
tokenizer.push_to_hub(repo_name)
```

```
from unsloth import FastLanguageModel
import torch
max_seq_length = 3000
dtype = None
load_in_4bit = True

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "./outputs/checkpoint-7000",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit
)
```