

ChatGPT 기반 어플리케이션 개발 과정

위블 CEO 노규남
bardroh@weable.ai

2024/3/11

- 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 개발**을 영역으로 비즈니스를 전개, **교육사업**도 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 Architecture), 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... x v

파일 삭제

클라이언트 쪽의 인터페이스를 구현해야 한다. 이렇게 하면 코드가 더 간결해서 유지보수가 용이해집니다.

5. 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.



리스코프 치환 원칙에 대해서 더 상세하게 설명해주세요.



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.

리스코프 치환 원칙은 다음과 같은 조건을 만족해야 합니다:

1. 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.
2. 하위 타입의 객체가 상위 타입의 인터페이스를 구현해야 한다. 이렇게 하면, 상위 타입의 변수에 하위 타입의 객체를 할당하더라도, 상위 타입의 메서드와 프로퍼티를 사용할 수 있습니다.
3. 하위 타입의 객체가 상위 타입의 인터페이스를 구현할 때, 상위 타입과 동일한 방식으로 작동해야 합니다. 즉, 상위 타입과 하위 타입 간에는 일관성이 있어야 합니다.

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.



NL2SQL

회사의 데이터에 대해서 자연어로 질의하 시면 해당하는 BigQuery 쿼리를 생성합니다.

상세하게 질의하실 수록 정확한 답을 얻을 수 있습니다.

데이터 카탈로그는 매일 업데이트 됩니다.



2023년 9월 1일부터 2023년 9월 30일까지 '닥터픽스' 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 = '닥터픽스';
```



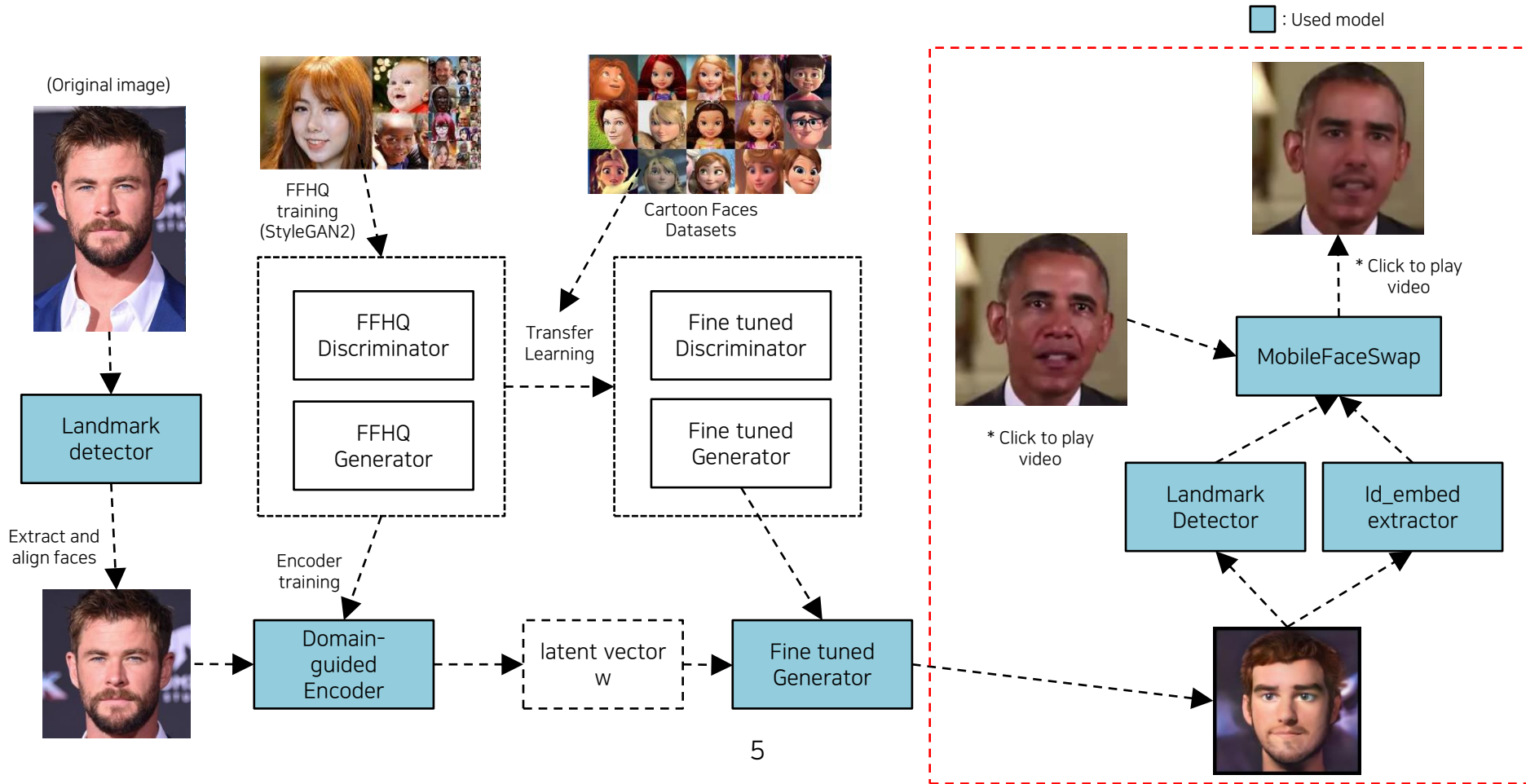
2023년 데이터로부터 전체 브랜드의 월별 재고 평균을 구해줘.

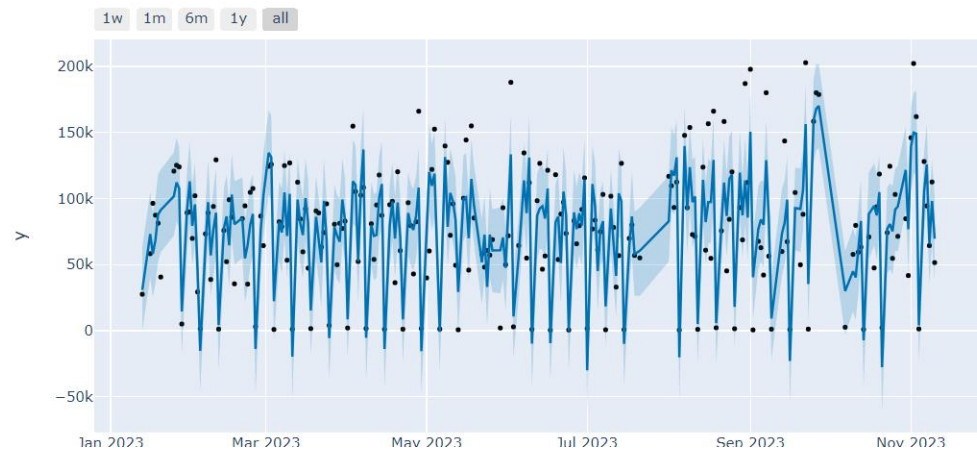
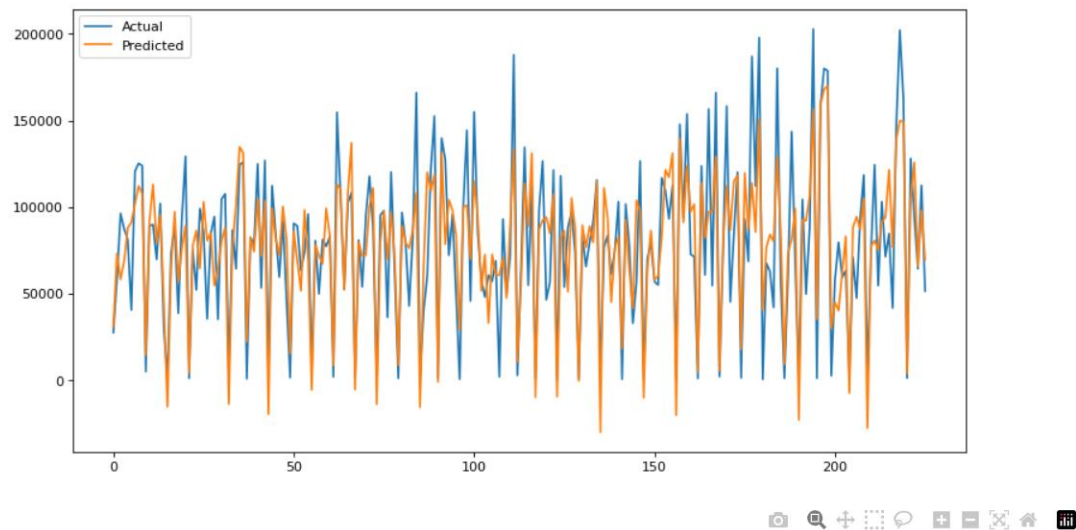


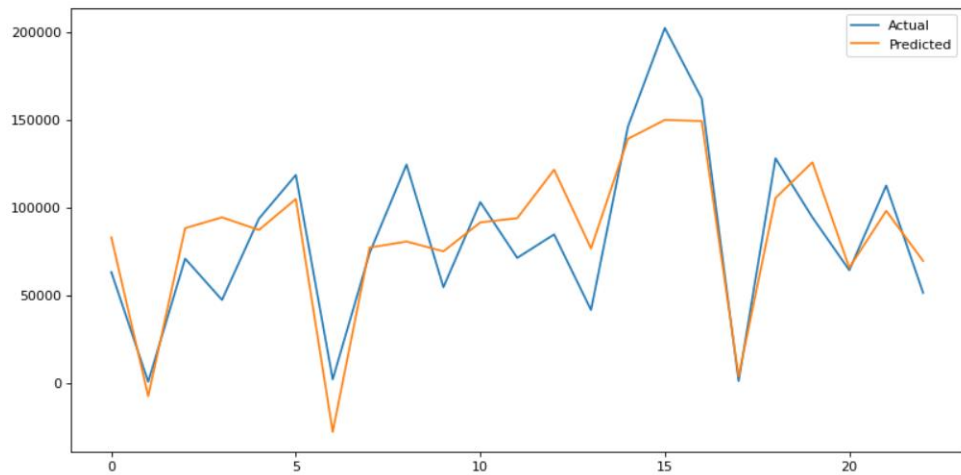
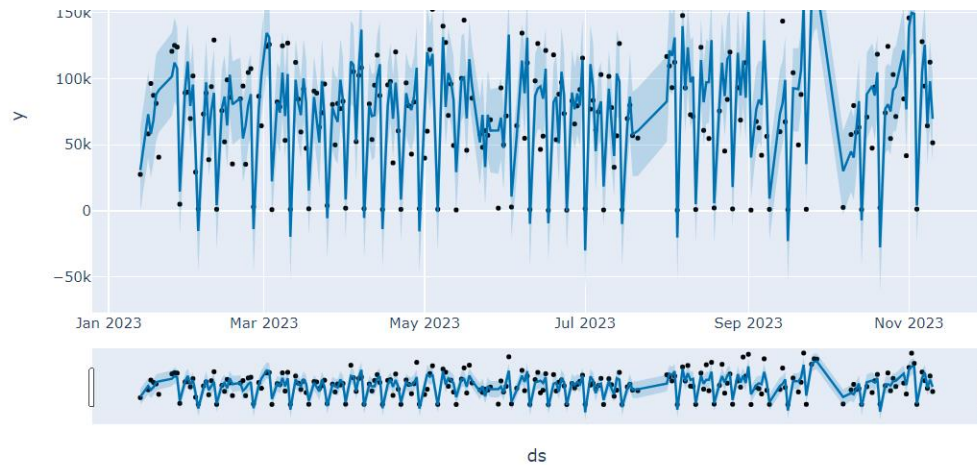
```
SELECT
  EXTRACT(YEAR FROM date_logged) AS year,
  EXTRACT(MONTH FROM date_logged) AS month,
  brand,
  AVG(quantity) AS average_stock
FROM
  company_in.daily_stock_shop_stock
WHERE
  EXTRACT(YEAR FROM date_logged) = 2023
GROUP BY
  year, month, brand
ORDER BY
  year, month;
```

대화를 입력해주세요.









- 이 과정의 주제는 ChatGPT를 포함한 OpenAI 모델들을 사용하는 방법과, 이 모델들을 다양한 라이브러리와 연동하여 실제 필요한 서비스를 만들어보는 것
- 교육시간은 월~수요일 오전 9시~오후 6시까지, 일일 8시간, 전체 24시간 분량
- 매 시간 정각에 시작하며, 50분 수업 후 10분 휴식을 반복. 점심시간은 오후 12시~1시
- 온라인으로 진행되는 수업 특성 상 커뮤니케이션이 적을 수 밖에 없음, 이를 보완하기 위해 수강생들 모두가 포함된 git repo(github)를 공유: 메일로 초청하여 멤버가 되며 이를 통해 필요한 파일들을 전달하고 실습, 과정 이해도를 간접적으로 평가

- 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 chat.openai.com, 무료계정은 GPT-3.5로 모델이 고정되어 있으며, 한달에 \$20의 비용을 지불하면 GPT-4로 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 제공되며, 이 크레딧이 모두 소진되거나 expire되면 그때부터 과금시작

- 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 사용하면 비용은 빠르게 증가한다. gpt-3.5-turbo는 gpt-4-turbo 대비 10%의 비용이지만 코딩이나 높은 논리력을 요구한다면 gpt-4를 사용할 수 밖에 없는 상황도 있다. 이 경우는 비용이 10배 이상 빠르게 증가한다.

- 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-22.04`로 설치, ubuntu 계정 생성하고 암호 설정
- `wsl -i -v`로 현재 설치된 배포본 확인
- `wsl -d Ubuntu-22.04`로 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

```
sudo su - root
```

```
whoami (root인 것을 확인)
```

```
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-kosa/kosa-chatgpt-2024-1st.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-osa/kosa-chatgpt-2024-1st.git

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

```
sudo su - root
```

```
apt-get update
```

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

```
pip3 install jupyter
```

```
jupyter notebook --ip=0.0.0.0 --port=8888 --allow-root --notebook-dir=/work/
```

이후 <http://localhost:8888>로 접속하고 token 입력하여 로그인

- <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)


```
sudo su - root
```

```
apt-get update
```

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

```
pip3 install jupyter openai
```

```
mkdir -p /work/notebooks
```

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

```
echo $OPENAI_API_KEY
```

```
jupyter notebook --ip=0.0.0.0 --port=8888 --allow-root --notebook-dir=/work
```

이후 <http://localhost:8888>로 접속하고 token 입력하여 로그인

VS Code Extension에서 Jupyter와 WSL을 찾아서 설치 후 종료

이후 WSL 실행

```
cd <git repo 경로>
```

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

```
echo $OPENAI_API_KEY
```

```
code .
```

[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.

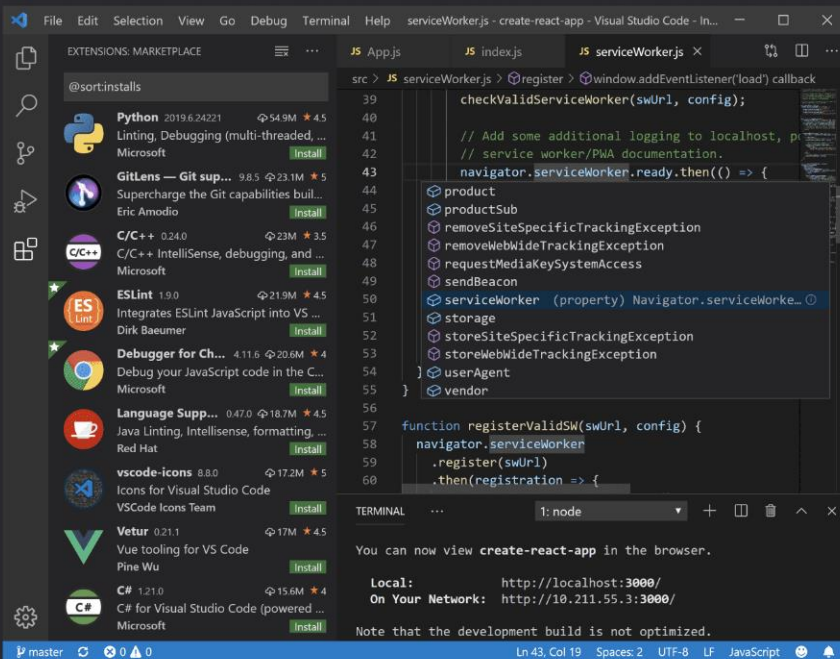
Download for Windows

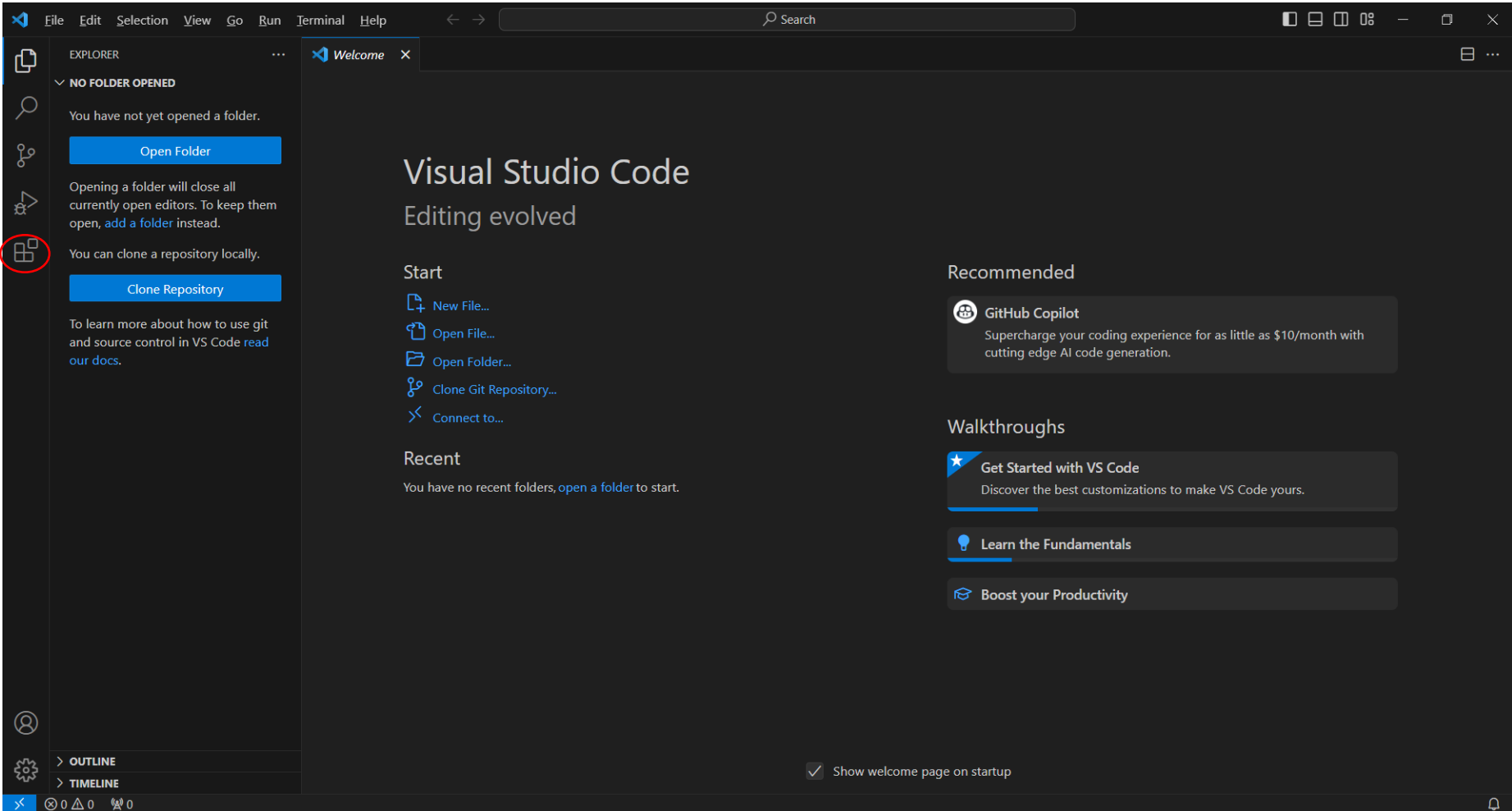
Stable Build



Web, Insiders edition, or other platforms

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










Visual Studio Code

Editing evolved


Start

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




Recent

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

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

FileEditSelectionViewGoRunTerminalHelp

EXTENSIONS: MARKETPLACE

Jupyter

Jupyter notebook support, interactive programming and co...

181ms

Microsoft

Jupyter Keymap

Jupyter keymaps for notebooks

8ms

Microsoft

Jupyter Slide Show

Jupyter Slide Show support for VS Code

8ms

Microsoft

Jupyter Cell Tags

Jupyter Cell Tags support for VS Code

6ms

Microsoft

Jupyter Notebook Renderers

Renderers for Jupyter Notebooks (with plotly, vega, gif, png...

10ms

Microsoft

Jupyter (deprecated)

Data Science with Jupyter on Visual Studio Code

802K 3.5

Don Jayamanne

VS Code Jupyter Notebook Previewer

An easy to use extension for previewing Jupyter Notebooks...

544K 3.5

jithurjacob

Install

Jupyter PowerToys

Experimental features for Jupyter notebook support in VS C...

261K 4.5

Microsoft

Install

jupyter-notebook-vscode

Runs jupyter notebooks in vscode

117K 1

Sam Helms

Install

Jupyter Theme

A colour theme based around Jupyter and Kaggle.

36K 5

SamCoding

Install

My Jupyter Notebook Previewer

My Jupyter Notebook Previewer

69K 2.5

colinfang

Install

Jupyter TOC

Generate table of contents for Jupyter notebooks.

15K 5

xelad0m


Install

Jupyter-QTConsole

Starts a remote Jupyter QTConsole kernel, and interfaces it

12K

Extension: Jupyter



Jupyter

v2024.2.0

Microsoft

microsoft.com

75,659,524

★★★★☆ (304)

Jupyter notebook support, interactive programming and computing that supports Intellisense, debugging and more.

Disable

Uninstall

Switch to Pre-Release Version


Extension is enabled on 'WSL: Ubuntu-22.04'

DETAILS

FEATURES

CHANGELOG


Extension Pack (4)



Jupyter Keymap

Jupyter keymaps for notebooks


Microsoft



Jupyter Notebook Renderers

Renderers for Jupyter Notebooks (with plotly...


Microsoft



Jupyter Slide Show

Jupyter Slide Show support for VS Code

Microsoft



Jupyter Cell Tags

Jupyter Cell Tags support for VS Code

Microsoft

Categories

Extension Packs

Data Science

Machine Learning

Notebooks

Visualization

Resources

Marketplace

Issues

Repository

PROBLEMS 2

OUTPUT

ubuntu@bards:~/work/...

Jupyter Slide Show v0.1.5

37.5M | 5

Jupyter Slide Show support for VS Code

This publisher has verified ownership of microsoft.com

Activation time: 8ms

Extension is enabled on 'WSL: Ubuntu-22.04'

WSL: Ubuntu-22.04

main*


2

0


0

EXTENSIONS: MARKETPLACE

WSL


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

14ms

**WSL workspaceFolder**
Returns 'workspaceFolder' as a WSL path. Useful where WSL format workspace...
Ifurzewaddock


52K ★ 5

Install

**Linux/Unix/WSL paths**
Provides commands to have unix-like paths (with slashes instead of backslash...
Krzysztof Grzywocz


38K ★ 3

Install

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


5.2M ★ 4.5

Install

**Open (WSL support)**
Opens the current file with the default application for the OS (in Windows for ...
NoThing


2K ★ 5

Install

**wsl-split**
lanjava


2K

Install

**WSL Path Tools**
Tools to convert paths between Windows and WSL format
Rob Clouth


5K

Install

**OCaml and Reason IDE - for WSL**
OCaml and Reason language support, for WSL
raiscl


8K

Install

**Save and Run with wsl**
Run commands when a file is saved in Terminal (add wsl path)
raiscl


6K ★ 3

Install

**Json Editor for VS code - WSL**
Generate a tree view editor of the active JSON document
pick68


55

Install

**Laravel Artisan**
Run Laravel Artisan commands within Visual Studio Code
Ryan Naddy


1.7M ★ 4

Install

**WSLSchoolAux**
Bruno School

182

Install

**WSL Recommender**
Recommend using the Windows Subsystem for Linux (WSL) and the WSL ext...
Microsoft


13K ★ 2.5

Install

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

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-

PROBLEMS (8)

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

JUPYTER

- ubuntu@DESKTOP-J33WK15:~/work/kosa-chatgpt-2024\$ git pull
Already up to date.
- ubuntu@DESKTOP-J33WK15:~/work/kosa-chatgpt-2024\$

WSL: Ubuntu-22.04

main

0

8

0

- 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

MODEL	DESCRIPTION
GPT-4 and GPT-4 Turbo	자연어 지시를 읽고 이해할 수 있는 OpenAI에서 가장 우수한 모델
GPT-3.5 Turbo	GPT-3.5 기반의 자연어 지시를 읽고 이해할 수 있는 모델. GPT-4 기반 모델보다 대략 10배 이상 저렴
DALL·E	자연어로 이미지를 생성하고 변경할 수 있는 모델
TTS	텍스트를 음성으로 바꾸는 모델
Whisper	오디오를 텍스트로 바꾸는 모델(STT)
Embeddings	텍스트를 n차원 배열(embedding vector)로 바꾸는 모델
Moderation	위험하거나 민감한 텍스트를 감지하는 모델
GPT base	지시어로 fine tune되지 않은 GPT의 기본 모델들(completion 기반)
Deprecated	오래 되어서 지원되지 않으며 대체해야 하는 모델들

	Model	Training	Input	Output
GPT-4-turbo				
	gpt-4-0125-preview		\$10.00 / 1M tokens	\$30.00 / 1M tokens
	gpt-4-1106-preview		\$10.00 / 1M tokens	\$30.00 / 1M tokens
	gpt-4-1106-vision-preview		\$10.00 / 1M tokens	\$30.00 / 1M tokens
GPT-4				
	gpt-4		\$30.00 / 1M tokens	\$60.00 / 1M tokens
	gpt-4-32k		\$60.00 / 1M tokens	\$120.00 / 1M tokens
GPT-3.5-turbo				
	gpt-3.5-turbo-0125		\$0.50 / 1M tokens	\$1.50 / 1M tokens
	gpt-3.5-turbo-instruct		\$1.50 / 1M tokens	\$2.00 / 1M tokens
Assistant API				
	Code interpreter		\$0.03 / session	
	Retrieval		\$0.20 / GB / assistant / day (free until 04/01/2024)	
Fine tuning Models				
	gpt-3.5-turbo	\$8.00 / 1M tokens	\$3.00 / 1M tokens	\$6.00 / 1M tokens
Embedding Models				
	text-embedding-3-small		\$0.02 / 1M tokens	
	text-embedding-3-large		\$0.13 / 1M tokens	
	ada v2		\$0.10 / 1M tokens	

- 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-35-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 사용할 수 있게 됨: OpenAI를 대규모 서비스에 적용할 경우 가장 문제가 되는 부분이 이 할당량 quota
- MS Azure의 OpenAI 모델은 별도의 폼으로 신청해야 하며, Azure의 체험 계정으로는 사용이 어려움

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

The screenshot shows the OpenAI Playground web interface. At the top, the browser address bar displays 'platform.openai.com/playground'. The navigation menu includes 'Overview', 'Documentation', 'API reference', 'Examples', and 'Playground'. On the right, there are links for 'Forum', 'Help', and the 'Weable, Inc' logo.

The main interface is divided into three sections:

- Get started** (left sidebar): Contains instructions on how to use the playground, including a note about model completion and a 'KEEP IN MIND' section with three bullet points: using good judgment when sharing outputs, the fact that requests are not used for training, and the knowledge cutoff date of 2021.
- Playground** (center): Features a 'SYSTEM' message box with the text 'You are a helpful assistant.', a 'USER' input field with the placeholder 'Enter a user message here.', and a 'Submit' button at the bottom.
- Settings** (right sidebar): Includes a 'View code' button (circled in red), a 'Model' dropdown set to 'gpt-3.5-turbo', and sliders for 'Temperature' (set to 1), 'Maximum length' (set to 256), 'Top P' (set to 1), 'Frequency penalty' (set to 0), and 'Presence penalty' (set to 0). A disclaimer at the bottom states that API and Playground requests are not used for training.

```
# 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-3.5-turbo",  
    messages=[{"role": "user", "content": "다음을 일본어로 번역하세요: 안녕하세요? 오늘 날씨가 참 좋군요!"}],  
    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

- 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-3.5-turbo",  
    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 작용 하고 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 설명하지만 llama나 다른 LLM 들에는 잘 적용되지 않을 수 있음. 작은 모델일수록 상세한 프롬프트를 잘 이해하지 못하는 경향

sharebook-kr/pykrx: KRX 주식 정보

github.com/sharebook-kr/pykrx

sharebook-kr / pykrx

Type / to search

>_

+ ▾

🕒

🔗

📧

🌐

<> Code🕒 Issues 18🔗 Pull requests 6🕒 Actions📁 Projects🛡 Security📈 Insights

📖

pykrx

Public

👁 Watch 21 ▾🔗 Fork 200 ▾☆ Star 546 ▾

🔑 master ▾🔗 6 branches🕒 0 tags

Go to fileAdd file ▾<> Code ▾

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

📁 pykrx

수정 추가 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

오픈API 상세

f

tw

ol

URL 복사

XMLJSON

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	한국관광공사
관리부서명	디지털인프라팀	관리부서 전화번호	033-371-2733
API 유형	REST	데이터포맷	JSON+XML
활용신청	800	키워드	아영장,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.interface는 Gradio 인터페이스와 python function을 연동
- Interface Class는 세가지 파라미터를 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.py 로 어플리케이션 구동
- gradio app.py로 구동: 'python' 어플리케이션을 사용하므로 다음과 같이 symbolic link를 걸어둌

```
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 session 으로부터 데이터를 다시 읽는 식으로 구성해야 함
- 한번만 초기화해야 하는 객체는 `st.cach_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/streamlit`이라는 이름으로 저장되는 python 스크립트로, `streamlit.web.cli.main`이라는 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|w.pyw|w.exe)?$', '', sys.argv[0])
    sys.exit(main())
```

`streamlit hello` # Streamlit 데모를 실행

`streamlit run plotting_demo.py` # 기본적인 실행 방법, 웹은 8501 포트에 열림

`streamlit run --server.port 8000 plotting_demo.py` # 웹을 8000 포트에 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` # 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>

- 무료 계정 생성후 프리티어(Free Tier)를 사용하면 제한된 자원으로 월 750시간, 12개월을 사용할 수 있음
- 필요한 것은 유효한 이메일 주소, 신용카드
- 프리티어 범위내에서 사용하고 있는 동안은 괜찮지만, 그 범위를 넘어서면 등록된 카드로 바로 결제됨
- 결제 관련 알림은 등록한 이메일로 받게 되어 있음
- 이 과정에서는 Ubuntu Server 22.04 LTS (HVM), SSD Volume Type(30G)으로 VM을 생성한 후 docker를 설치하고 테스트해볼 예정

- AWS 프리티어에서 사용 가능한 VM 자원은 다음과 같으며, Linux는 x86과 arm, Windows는 x86 선택 가능

Amazon Linux 2 / 2023 AMI (HVM), SSD Volume Type

Ubuntu Server 18.04 / 20.04 / **22.04** LTS (HVM), SSD Volume Type

SUSE Linux Enterprise Server 12 SP4 / 15 SP4 (HVM), SSD Volume Type

Debian 11 (HVM), SSD Volume Type

Red Hat Enterprise Linux 9 (HVM), SSD Volume Type

Microsoft Windows Server 2019 / 2022 Base

- VM의 타입은 x86인 경우 t2.micro(1C/1G), arm은 t4g.micro(2C/1G)만 선택 가능

인스턴스에 연결 | EC2 Manager x

ap-northeast-2.console.aws.amazon.com/ec2/home?region=ap-northeast-2#ConnectToInstanceinstanceId=i-0badcc939cc5340e9

aws 서비스 검색 [알트+S]

서술 hotangzang1

인스턴스 > i-0badcc939cc5340e9 > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0badcc939cc5340e9 (myserver)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID

i-0badcc939cc5340e9 (myserver)

퍼블릭 IP 주소

13.124.4.227

사용자 이름

인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ubuntu(을) 사용합니다.

ubuntu

참고: 대부분의 경우 기본 사용자 이름 ubuntu(는) 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

취소

연결

의견 언어

© 2023, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

aws 인스턴스에 연결 | EC2 Manager x EC2 Instance Connect x +

ap-northeast-2.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-northeast-2&connType=standard&instanceId=i-0badcc939cc5340e9&osUser=ubuntu&sshPort=22#/ [알트+S]

```
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1028-aws x86_64)

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

System information as of Sun Mar 19 11:00:43 UTC 2023

System load:  0.0      Processes:      98
Usage of /:    5.5% of 28.89GB   Users logged in:  0
Memory usage: 22%      IPv4 address for eth0: 172.31.47.254
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Mar 19 10:59:31 2023 from 13.209.1.59
ubuntu@ip-172-31-47-254:~$
```

i-0badcc939cc5340e9 (myserver)

PublicIPs: 13.124.4.227 PrivateIPs: 172.31.47.254

의견 언어

© 2023, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

- https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/putty.html
- AWS의 VM들에 접속하기 위해서는 터미널 에뮬레이터가 필요한데, 이 중 가장 유명하고 많이 사용되는 것이 바로 putty. iputty는 putty의 한글 버전
- iputty 패키지에는 터미널 에뮬레이터인 putty 외 puttygen 등 관련된 유틸리티들도 포함되어 있음
- AWS의 VM을 만들 때 keypair를 사용했는데, keypair는 public key와 private key로 구성됨
- public key는 VM이 가지고 있고, private key는 다운로드 받아서 iputty에서 접속에 사용: iputty를 사용한다면 ppk로 다운로드, Mac이나 리눅스에서 ssh client로 접속한다면 ppe로 다운로드
- 접속시 정확한 private key와 계정만 알고 있으면, 암호 없이 접근이 가능: 여러 대의 VM을 관리할 때 매우 편리, 자동화에는 필수. Ubuntu 22.04에 사용한 계정은 ubuntu. 참고로 Amazon AMI에서는 ec2-user를 사용함

1) github 계정 생성

2) 초청 이메일 수령, repo 참여

3) VM 로그인, 작업 디렉토리 생성

```
mkdir /work
```

```
cd /work
```

4) ssh키 설정

```
ssh-keygen
```

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

화면의 내용을 전부 복사해서 github의 계정-Settings-SSH and GPG keys에 New SSH key로 붙임

5) 계정 설정

git config --global user.name "Bard K. Roh"

git config --global user.email "bardroh@weable.ai"

6) repo clone

git clone [git@github.com:weable-kosa/kosa-chatgpt-2024-1st.git](https://github.com/weable-kosa/kosa-chatgpt-2024-1st.git)

7) 작업하고 git add; git commit; git push

8) 작업 전 항상 git pull (본인 폴더만 작업)

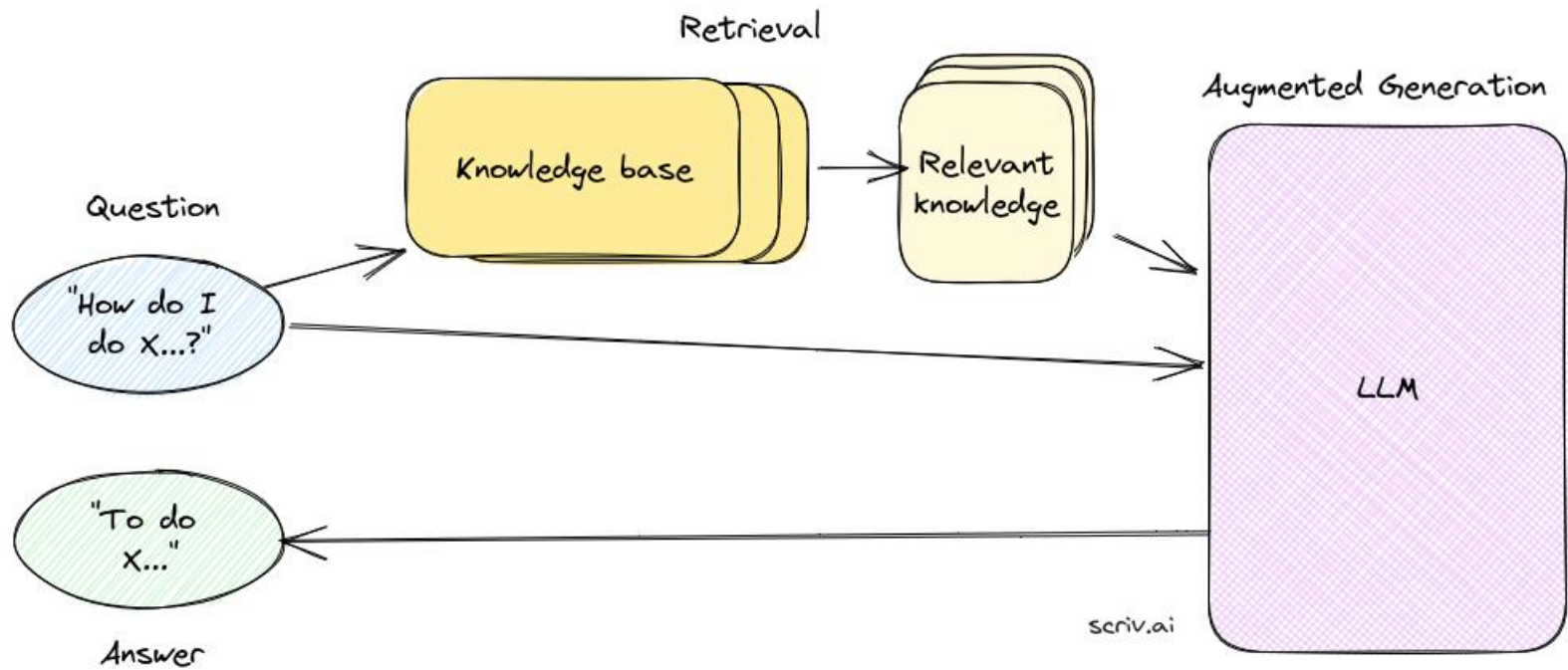
다른 방법

git clone <https://github.com/weable-kosa/kosa-chatgpt-2024-1st.git> (Settings-Developer Settings-Personal access tokens에서 basic token을 만든 후 암호로 입력)

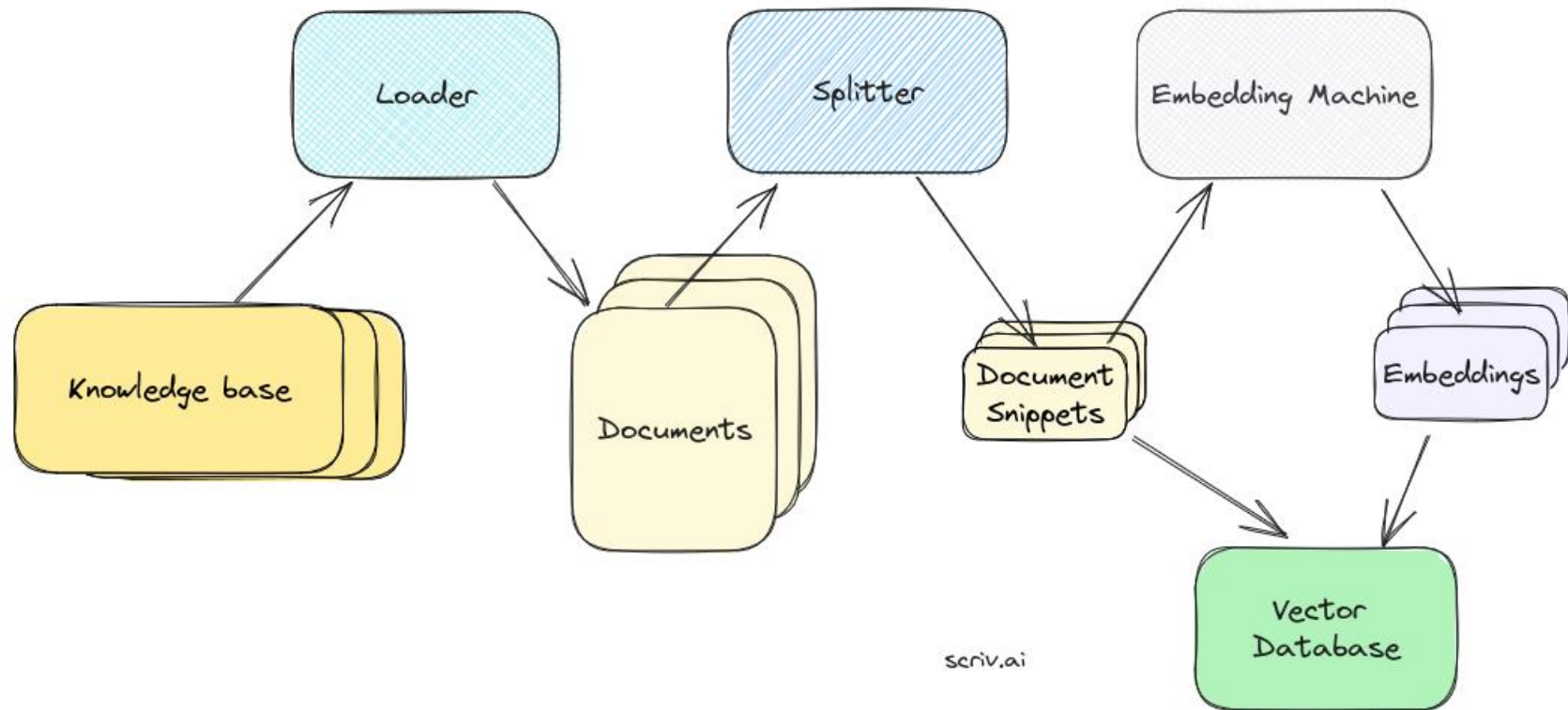
- 언어모델을 기반으로 한 어플리케이션을 개발하기 위한 프레임워크
- 언어모델 학습시 사용된 데이터 외 다른 데이터를 인식할 수 있게 함
- 여러가지 LLM 모델들을 선택적으로 사용할 수 있게 함
- 능동적으로 다른 기능들을 연동하여 추가적인 결과를 낼 수 있게 함
- 여러 기능 모듈을 체인으로 연결하여 파이프를 만들어서 기능적으로 확장할 수 있음
- 내부적으로는 LLM의 프롬프트 조작과 관련 라이브러리셋으로 이런 기능들을 구현

- 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

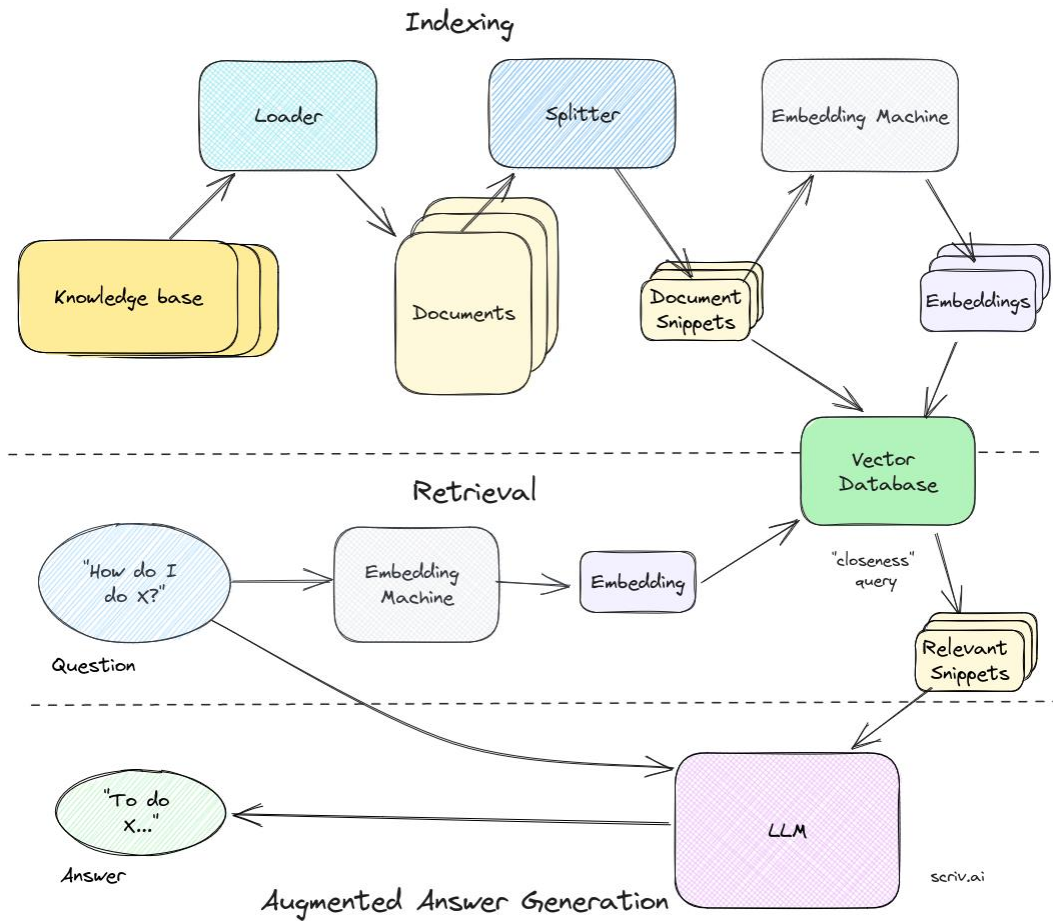
- Embedding은 다양한 데이터를 정해진 차원의 벡터로 변환한 것
- 이러한 벡터들을 저장하고 검색하여 주어진 벡터와 가장 가까운 것들을 추출할 수 있는데, 이렇게 벡터들을 저장하고 관리하는 일종의 Vector Store를 VectorDB라 부름
- 일반적인 데이터베이스에서는, 주로 쿼리와 정확히 일치하는 데이터베이스의 행을 찾는 반면, VectorDB에서는 유사성 메트릭을 적용하여 쿼리(사용자 질문)과 가장 유사한 벡터를 검색
- RAG(Retrieval Augmented Generation)에서 Retriever에 해당하는 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?



Top Piriyakulkij · Follow

6 min read · Aug 28, 2023



21



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

- 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
- Retrieval을 위해 파일을 업로드할 경우, GB당 \$0.20이 하루에 과금됨. 만약 하나의 파일을 여러 개의 Assistant가 같이 사용한다면, Assistant 수만큼 요금이 늘어남. 파일을 삭제하지 않으면 계속 과금
- User Message의 일부로 파일이 추가되었다면 해당 파일에도 동일하게, 각각 GB당 \$0.20이 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 Retrieval에서의 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