Langchain

LangChain

| Langchain?

ChatGPT의 능력을 올려주는 장비를 착용해보자.

2022년 10월, LLM을 위한 API가 등장했습니다.

- Python, JavaScript를 지원합니다.
- LLM에 자료를 추가하거나, 여러 모델을 사용하는 기능을 지원합니다.
- **Open Source Project**입니다.

Langchain으로 무엇을 할 수 있을까?

- 지식 기반 시스템, 질문 답변 시스템, 요약 시스템을 만들 수 있습니다.
- LLM을 프로그래밍 기반으로 다뤄보기
- LLM interface가 필요한 시스템에 적용하기
- 이외 ChatGPT에서 제한되었던 기능들을 추가해 구현할 수 있습니다.

Langchain으로 무엇을 할 수 있을까?

Langchain을 검색도구로 나중에 참조하는 유용한 정보를 저장할 수 있습니다.

외부 **API** 사용가능, **Python Coding** 가능

이전 채팅 내용을 더 자유롭게 활용할 수 있습니다.

LLM을 코드로 사용하기 좋은 환경을 조성해줍니다.

Langchain으로 무엇을 할 수 있을까?

- 지식 기반 시스템, 질문 답변 시스템, 요약 시스템을 만들 수 있습니다.
- LLM을 프로그래밍 기반으로 다뤄보기
- LLM interface가 필요한 시스템에 적용하기
- 이외 ChatGPT에서 제한되었던 기능들을 추가해 구현할 수 있습니다.

그럼 **Langchain**을 사용하는 이유는?

| Langchain

복잡한 구현을 쉽게 진행할 수 있다.

Without LCEL

```
from typing import List

import openai


prompt_template = "Tell me a short joke abou
client = openai.OpenAI()

def call_chat_model(messages: List[dict]) ->
    response = client.chat.completions.creat
        model="gpt-3.5-turbo",
        messages=messages,
    )
    return response.choices[0].message.conte

def invoke_chain(topic: str) -> str:
    prompt_value = prompt_template.format(to
    messages = [{"role": "user", "content":
    return call_chat_model(messages)

invoke_chain("ice cream")
```

LCEL

```
from langchain_core.runnables import Runnabl


prompt = ChatPromptTemplate.from_template(
    "Tell me a short joke about {topic}"
)
output_parser = StrOutputParser()
model = ChatOpenAI(model="gpt-3.5-turbo")
chain = (
    {"topic": RunnablePassthrough()}
    | prompt
    | model
    | output_parser
)

chain.invoke("ice cream")
```

| Langchain

복잡한 구현을 쉽게 진행할 수 있다.

Without LCEL

```
from typing import Iterator


def stream_chat_model(messages: List[dict])
    stream = client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=messages,
        stream=True,
    )
    for response in stream:
        content = response.choices[0].delta.
        if content is not None:
            yield content

def stream_chain(topic: str) -> Iterator[str
    prompt_value = prompt.format(topic=topic
    return stream_chat_model([{"role": "user


for chunk in stream_chain("ice cream"):
    print(chunk, end="", flush=True)
```

LCEL

```
for chunk in chain.stream("ice cream"):
    print(chunk, end="", flush=True)
```

| Langchain

복잡한 구현을 쉽게 진행할 수 있다.

Without LCEL

```
from concurrent.futures import ThreadPoolExe


def batch_chain(topics: list) -> list:
    with ThreadPoolExecutor(max_workers=5) a
        return list(executor.map(invoke_chai

batch_chain(["ice cream", "spaghetti", "dump
```

LCEL

```
chain.batch(["ice cream", "spaghetti", "dump
```

| OpenAI API key

실습을 위한 API 키 발급 및 적용 방법

2022년 10월, LLM을 위한 API가 등장했습니다.

- Python, JavaScript를 지원합니다.
- LLM에 자료를 추가하거나, 여러 모델을 사용하는 기능을 지원합니다.
- **Open Source Project**입니다.

자세한 추가 내용들은
Colab 실습과 함께 진행하겠습니다.