機器學習案例問答

向量資料庫建立與操作(雲端/本地端)

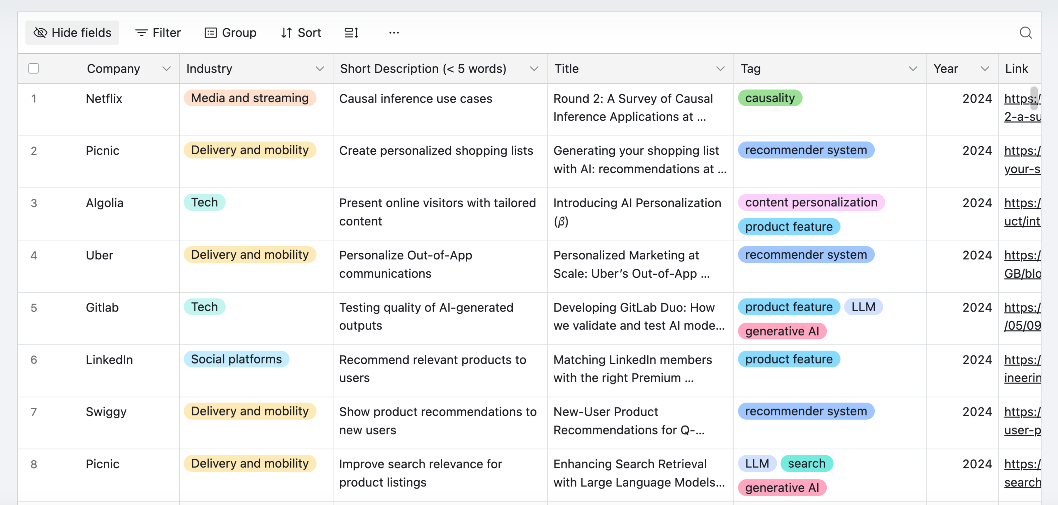
Ryan Chung

20260108

需求

- 將資料轉換成向量資料庫
- 使用者輸入一句話後，能找到最相近的資料
- 新增資料至向量資料庫

資料集：Real-world ML & LLM systems



□	Company	Industry	Short Description (< 5 words)	Title	Tag	Year	Link
1	Netflix	Media and streaming	Causal inference use cases	Round 2: A Survey of Causal Inference Applications at ...	causality	2024	https:/ 2-a-su
2	Picnic	Delivery and mobility	Create personalized shopping lists	Generating your shopping list with AI: recommendations at ...	recommender system	2024	https:/ your-s
3	Algolia	Tech	Present online visitors with tailored content	Introducing AI Personalization (β)	content personalization product feature	2024	https:/ uct/int
4	Uber	Delivery and mobility	Personalize Out-of-App communications	Personalized Marketing at Scale: Uber's Out-of-App ...	recommender system	2024	https:/ GB/blo
5	Gitlab	Tech	Testing quality of AI-generated outputs	Developing GitLab Duo: How we validate and test AI mode...	product feature LLM generative AI	2024	https:/ /05/09
6	LinkedIn	Social platforms	Recommend relevant products to users	Matching LinkedIn members with the right Premium ...	product feature	2024	https:/ ineerin
7	Swiggy	Delivery and mobility	Show product recommendations to new users	New-User Product Recommendations for Q-...	recommender system	2024	https:/ user-p
8	Picnic	Delivery and mobility	Improve search relevance for product listings	Enhancing Search Retrieval with Large Language Models...	LLM search generative AI	2024	https:/ search

https://www.evidentlyai.com/ml-system-design

Qdrant

- 向量搜尋引擎
- 執行方式
  - 本地模式
  - Docker部署
  - 雲端

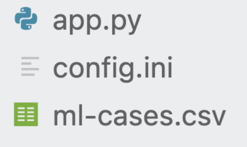
建立專案

- 新增資料夾 ml-cases

  app.py

  config.ini

  ml-cases.csv



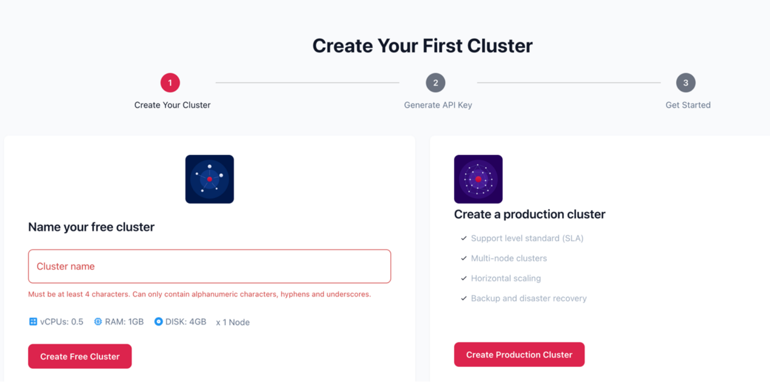
app.py

config.ini

ml-cases.csv

Qdrant Cloud

- email註冊登入後，建立第一個Cluster
  - 命名為：ml-cases-yyyymmdd
- 產生API Key
- 記下Key



Create Your First Cluster

Name your free cluster

Cluster name

Must be at least 4 characters. Can only contain alphanumeric characters, hyphens and underscores.

vCPUs: 0.5 RAM: 1GB DISK: 4GB x 1 Node

Create a production cluster

- ✓ Support level standard (SLA)
- ✓ Multi-node clusters
- ✓ Horizontal scaling
- ✓ Backup and disaster recovery

https://cloud.qdrant.io/login

config.ini

```ini
[Qdrant]
URL = https://xxxxxxxxx:6333
API_KEY = xxxxxxxxxxx
```

requirements.txt

```
langchain_community
pandas
langchain_huggingface
faiss-cpu
sentence-transformers
ipykernel
langchain-openai
langchain_google_genai
tiktoken
chardet
langchain_ollama
docx2txt
pypdf
langchain_qdrant
```

app.py

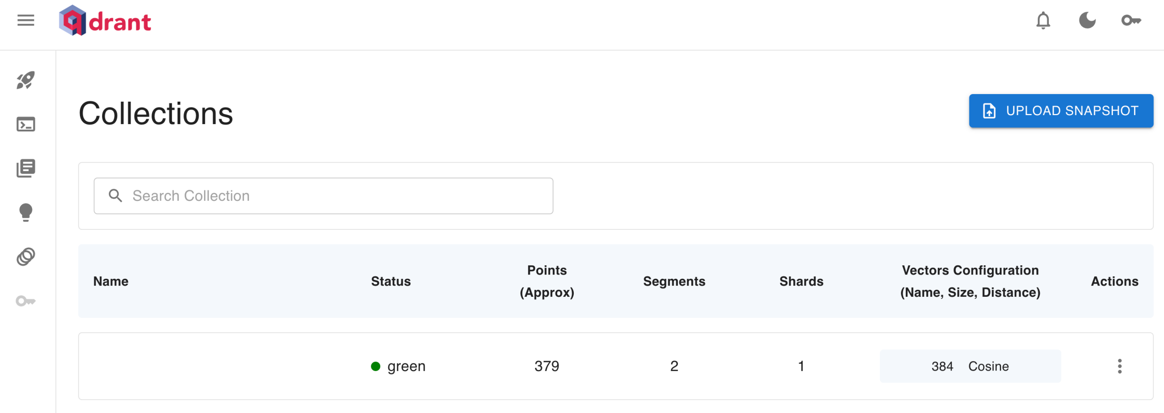
```
from configparser import ConfigParser
config = ConfigParser()
config.read("config.ini")
import pandas as pd

# Load dataset
df = pd.read_csv("ml-cases.csv")

df.head()

df["TitleAndDescription"] = df["Title"] + " - " + df["Short Description (< 5 words)"]

df["Year"].value_counts()
df_since_2024 = df[df["Year"] >= 2024]
df_until_2023 = df[df["Year"] < 2024]
df_until_2023["Year"].value_counts()

from langchain_huggingface import HuggingFaceEmbeddings

embedding_function = HuggingFaceEmbeddings(
    model_name="sentence-transformers/all-MiniLM-L6-v2"
)
```

app.py

```python
metadatas = []
for i, row in df_until_2023.iterrows():
    metadatas.append(
        {
            "Company": row["Company"],
            "Industry": row["Industry"],
            "Tag": row["Tag"],
            "Year": row["Year"],
            "Link": row["Link"],
        }
    )
# df_until_2023["TitleAndDescription"] = df_until_2023["TitleAndDescription"].astype(str)
df_until_2023 = df_until_2023.astype({"TitleAndDescription": str})

from langchain_qdrant import QdrantVectorStore

qdrant = QdrantVectorStore.from_texts(
    texts=df_until_2023["TitleAndDescription"].to_list(),
    embedding=embedding_function,
    metadatas=metadatas,
    url=config["Qdrant"]["URL"],
    api_key=config["Qdrant"]["API_KEY"],
    prefer_grpc=True
)

question = "What kind of frauds Blablacar use machine learning to prevent?"
results = qdrant.similarity_search(question, k=5)
for i, case in enumerate(results):
    print(f"Case {i+1}:")
    print(f"{case.page_content}")
    print("====================================================")
```

https://gist.github.com/ryanchung403/eebd2a1defba090078499fa722524fc3

觀察雲端是否有拿到資料

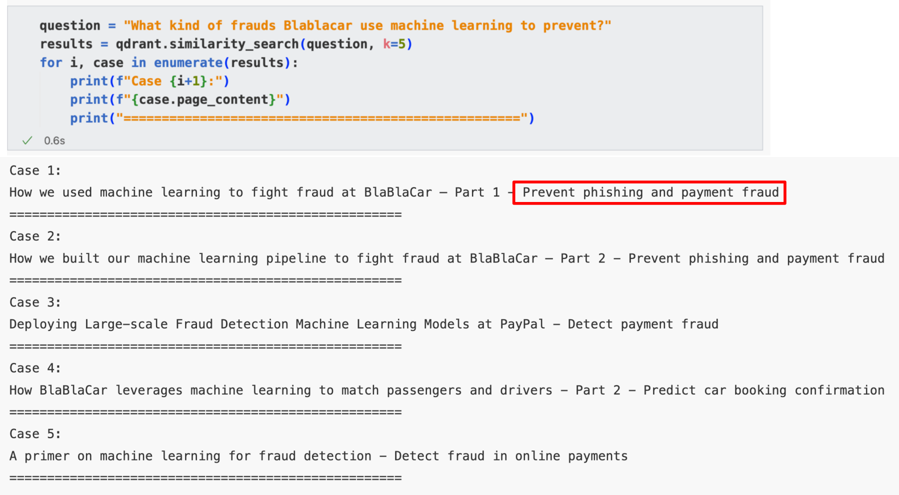


Collections

Name	Status	Points (Approx)	Segments	Shards	Vectors Configuration (Name, Size, Distance)	Actions
	● green	379	2	1	384 Cosine	⋮

結果確認

- 成功找到答案！



```python
question = "What kind of frauds Blablacar use machine learning to prevent?"
results = qdrant.similarity_search(question, k=5)
for i, case in enumerate(results):
    print(f"Case {i+1}:")
    print(f"{case.page_content}")
    print("====================================================")
```

✓ 0.6s

```
Case 1:
How we used machine learning to fight fraud at BlaBlaCar — Part 1 — Prevent phishing and payment fraud
====================================================
Case 2:
How we built our machine learning pipeline to fight fraud at BlaBlaCar — Part 2 — Prevent phishing and payment fraud
====================================================
Case 3:
Deploying Large-scale Fraud Detection Machine Learning Models at PayPal — Detect payment fraud
====================================================
Case 4:
How BlaBlaCar leverages machine learning to match passengers and drivers — Part 2 — Predict car booking confirmation
====================================================
Case 5:
A primer on machine learning for fraud detection — Detect fraud in online payments
====================================================
```

結合大型語言模型

- 將已經取得的線索提供給大型語言模型
- 大型語言模型只依據線索來回答問題

config.ini 加上 Gemini API Key

```
[Qdrant]
URL = https://xxxxxxxxx
API_KEY = xxxxxxxxxxx
[Gemini]
API_KEY = xxxxxxxxxx
```

app.py

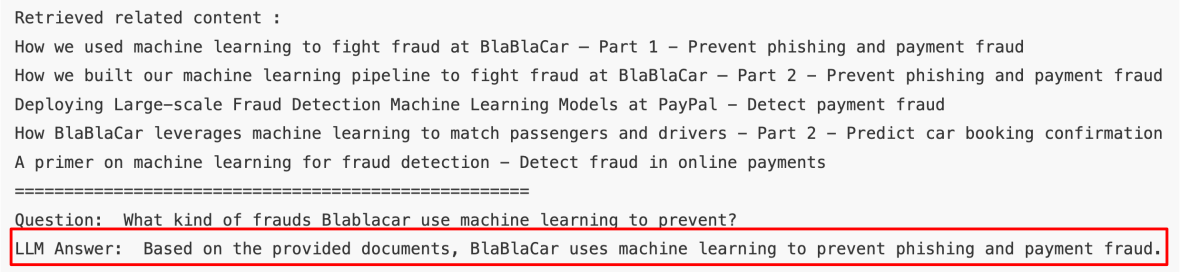
```python
from langchain_google_genai import ChatGoogleGenerativeAI

llm_gemini = ChatGoogleGenerativeAI(
    model="gemini-2.5-flash",
    google_api_key=config["Gemini"]["API_KEY"],
)

from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser

prompt = ChatPromptTemplate.from_template(
    """Answer the following question based only on the provided context:
<context>
{context}
</context>
Question: {input}"""
)

output_parser = StrOutputParser()

chain = prompt | llm_gemini | output_parser

query = "What kind of frauds Blablacar use machine learning to prevent?"
# query = "What Romie can do for you?"
```

app.py

```
results = qdrant.similarity_search(query, k=5)
print("Retrieved related content :")
print(results[0].page_content)
print(results[1].page_content)
print(results[2].page_content)
print(results[3].page_content)
print(results[4].page_content)
print("====================================================")

llm_result = chain.invoke(
    {
        "input": query,
        "context": [results[0],
                    results[1],
                    results[2],
                    results[3],
                    results[4]
        ]
    }
)

print("Question: ", query)
print("LLM Answer: ", llm_result)
```

https://gist.github.com/ryanchung403/f87d9903a1b445e6ad5a7a3c60878b70

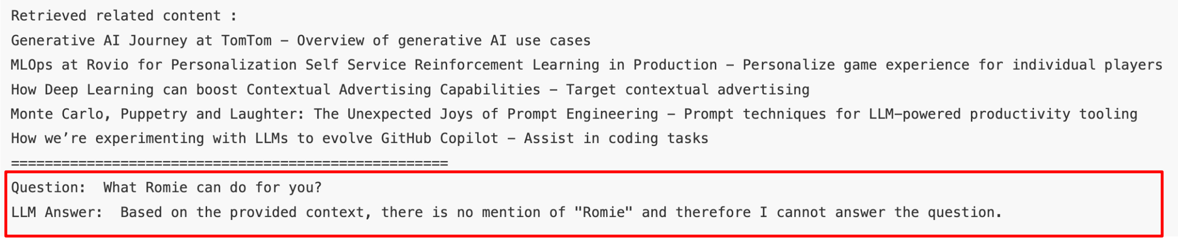
觀察結果



```
Retrieved related content :
How we used machine learning to fight fraud at BlaBlaCar – Part 1 – Prevent phishing and payment fraud
How we built our machine learning pipeline to fight fraud at BlaBlaCar – Part 2 – Prevent phishing and payment fraud
Deploying Large-scale Fraud Detection Machine Learning Models at PayPal – Detect payment fraud
How BlaBlaCar leverages machine learning to match passengers and drivers – Part 2 – Predict car booking confirmation
A primer on machine learning for fraud detection – Detect fraud in online payments
====================================================
Question:  What kind of frauds Blablacar use machine learning to prevent?
LLM Answer:  Based on the provided documents, BlaBlaCar uses machine learning to prevent phishing and payment fraud.
```

答得漂亮！

如果問了尚未收錄的案例



```
Retrieved related content :
Generative AI Journey at TomTom - Overview of generative AI use cases
MLOps at Rovio for Personalization Self Service Reinforcement Learning in Production - Personalize game experience for individual players
How Deep Learning can boost Contextual Advertising Capabilities - Target contextual advertising
Monte Carlo, Puppetry and Laughter: The Unexpected Joys of Prompt Engineering - Prompt techniques for LLM-powered productivity tooling
How we're experimenting with LLMs to evolve GitHub Copilot - Assist in coding tasks
====================================================
Question:  What Romie can do for you?
LLM Answer:  Based on the provided context, there is no mention of "Romie" and therefore I cannot answer the question.
```

還是會查詢出5筆最相似的案例
但LLM可以成功確認沒有相關案例

將2024年之後的案例也加入向量資料庫

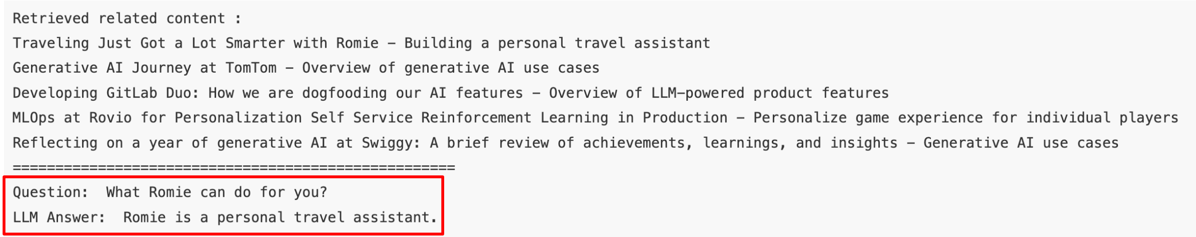
```python
# Add the new case to the dataset
from uuid import uuid4

metadatas = []
for i, row in df_since_2024.iterrows():
    metadatas.append(
        {
            "Company": row["Company"],
            "Industry": row["Industry"],
            "Tag": row["Tag"],
            "Year": row["Year"],
            "Link": row["Link"],
        }
    )
# df_since_2024["TitleAndDescription"] = df_since_2024["TitleAndDescription"].astype(str)
df_since_2024 = df_since_2024.astype({"TitleAndDescription": str})
uuids = [str(uuid4()) for _ in range(len(df_since_2024))]

qdrant.add_texts(
    texts=df_since_2024["TitleAndDescription"].to_list(),
    metadatas=metadatas,
    ids=uuids,
)
```

再查詢一次

```python
query = "What Romie can do for you?"

results = qdrant.similarity_search(query, k=5)
print("Retrieved related content :")
print(results[0].page_content)
print(results[1].page_content)
print(results[2].page_content)
print(results[3].page_content)
print(results[4].page_content)
print("====================================================")

llm_result = chain.invoke(
    {
        "input": query,
        "context": [results[0],
                    results[1],
                    results[2],
                    results[3],
                    results[4]
        ]
    }
)

print("Question: ", query)
print("LLM Answer: ", llm_result)
```

https://gist.github.com/ryanchung403/081a31297ff15c80a90980f7c09d9304

執行結果

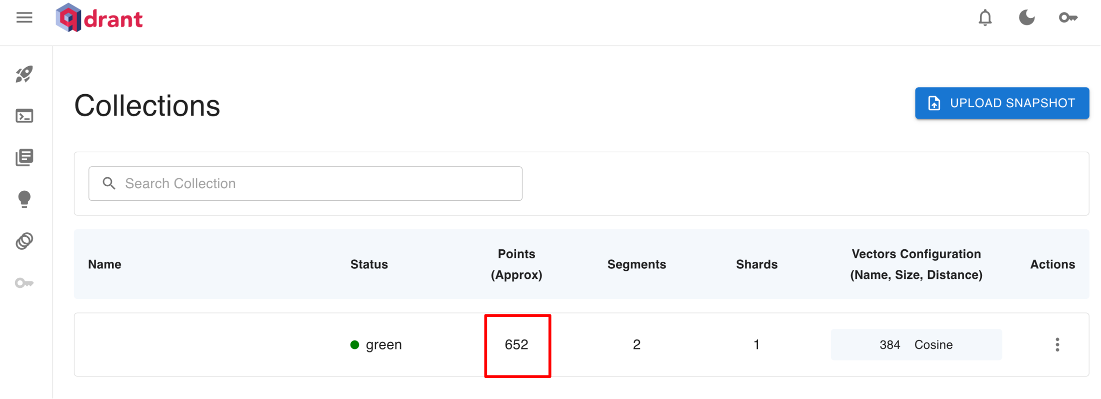


```
Retrieved related content :
Traveling Just Got a Lot Smarter with Romie - Building a personal travel assistant
Generative AI Journey at TomTom - Overview of generative AI use cases
Developing GitLab Duo: How we are dogfooding our AI features - Overview of LLM-powered product features
MLOps at Rovio for Personalization Self Service Reinforcement Learning in Production - Personalize game experience for individual players
Reflecting on a year of generative AI at Swiggy: A brief review of achievements, learnings, and insights - Generative AI use cases
====================================================
Question:  What Romie can do for you?
LLM Answer:  Romie is a personal travel assistant.
```

可成功調閱出2024年的案例！

確認雲端向量資料庫

- Qdrant -> Clusters -> Actions … -> Dashboard
- 的確更新為652筆資料！



Collections

Name	Status	Points (Approx)	Segments	Shards	Vectors Configuration (Name, Size, Distance)	Actions
	● green	652	2	1	384 Cosine	⋮

LangChain + Embedding + Qdrant + Gemini

- 註冊Qdrant帳號，取得URL與API Key
- Embedding
  - 使用 OpenSource/OpenAI /Gemini 任一種Embedding模型
- LangChain
  - 資料匯入
  - 結合Embedding在Qdrant建立向量資料庫
- LLM
  - 提示樣板
  - 取得Qdrant的最相似結果，提供給LLM進行作答

Qdrant

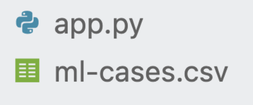
- 向量搜尋引擎
- 執行方式
  - 本地模式
  - Docker部署
  - 雲端

建立專案

- 新增資料夾 ml-cases

  app.py

  ml-cases.csv



app.py

ml-cases.csv

app.py

```python
import pandas as pd

# Load dataset
df = pd.read_csv("ml-cases.csv")

df.head()

df["TitleAndDescription"] = df["Title"] + " - " + df["Short Description (< 5 words)"]

df["Year"].value_counts().sort_index()
df_since_2024 = df[df["Year"] >= 2024]
df_until_2023 = df[df["Year"] < 2024]
df_until_2023["Year"].value_counts().sort_index()
len(df_until_2023)
len(df_since_2024)
len(df)

from langchain_huggingface import HuggingFaceEmbeddings

embedding_function = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
```

app.py

```
metadatas = []
for i, row in df_until_2023.iterrows():
    metadatas.append(
        {
            "Company": row["Company"],
            "Industry": row["Industry"],
            "Tag": row["Tag"],
            "Year": row["Year"],
            "Link": row["Link"],
        }
    )

# df_until_2023["TitleAndDescription"] = df_until_2023["TitleAndDescription"].astype(str)
df_until_2023 = df_until_2023.astype({"TitleAndDescription": str})

from langchain_qdrant import QdrantVectorStore
from qdrant_client import QdrantClient
from qdrant_client.http.models import Distance, VectorParams

client = QdrantClient(path="qdrant_storage")

client.create_collection(
    collection_name="ml_cases_collection",
    vectors_config=VectorParams(size=384,
                                distance=Distance.COSINE),
)
```

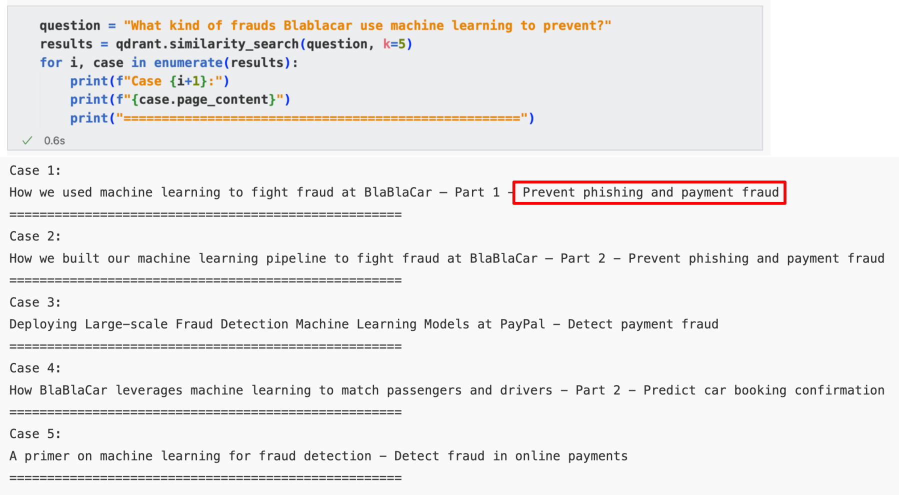
app.py

```
qdrant = QdrantVectorStore(
    client=client,
    collection_name="ml_cases_collection",
    embedding=embedding_function,
)


qdrant.add_texts(
    texts=df_until_2023["TitleAndDescription"].to_list(),
    metadatas=metadatas,
    ids=None,
)

question = "What kind of frauds Blablacar use machine learning to prevent?"
results = qdrant.similarity_search(question, k=5)
for i, case in enumerate(results):
    print(f"Case {i+1}:")
    print(f"{case.page_content}")
    print("====================================================")
```

https://gist.github.com/ryanchung403/f27c87540b67da7cd7a96119c6c0f8d4

結果確認

- 成功找到答案！



```python
question = "What kind of frauds Blablacar use machine learning to prevent?"
results = qdrant.similarity_search(question, k=5)
for i, case in enumerate(results):
    print(f"Case {i+1}:")
    print(f"{case.page_content}")
    print("====================================================")
```

✓ 0.6s

```
Case 1:
How we used machine learning to fight fraud at BlaBlaCar — Part 1 — Prevent phishing and payment fraud
====================================================
Case 2:
How we built our machine learning pipeline to fight fraud at BlaBlaCar — Part 2 — Prevent phishing and payment fraud
====================================================
Case 3:
Deploying Large-scale Fraud Detection Machine Learning Models at PayPal — Detect payment fraud
====================================================
Case 4:
How BlaBlaCar leverages machine learning to match passengers and drivers — Part 2 — Predict car booking confirmation
====================================================
Case 5:
A primer on machine learning for fraud detection — Detect fraud in online payments
====================================================
```

結合大型語言模型

- 將已經取得的線索提供給大型語言模型
- 大型語言模型只依據線索來回答問題

app.py

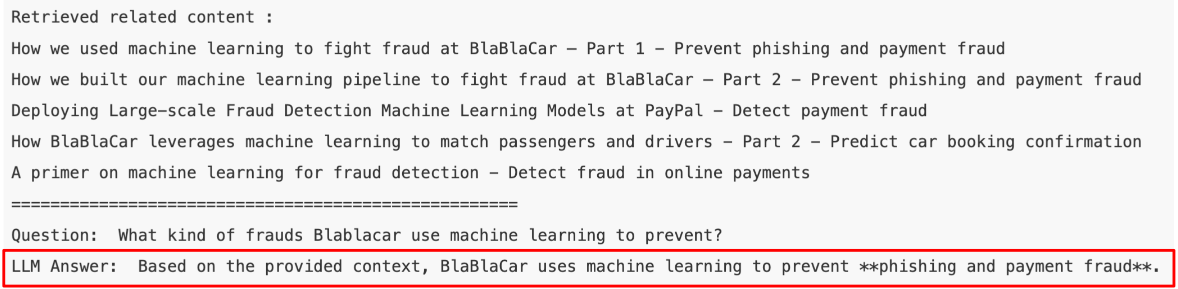
```python
from langchain_ollama.llms import OllamaLLM

ollama_llm = OllamaLLM(model="gemma3n:e4b")

from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser

prompt = ChatPromptTemplate.from_template(
    """Answer the following question based only on the provided context:
<context>
{context}
</context>
Question: {input}"""
)

output_parser = StrOutputParser()

chain = prompt | ollama_llm | output_parser

query = "What kind of frauds Blablacar use machine learning to prevent?"
# query = "What Romie can do for you?"
```

app.py

```python
results = qdrant.similarity_search(query, k=5)
print("Retrieved related content :")
print(results[0].page_content)
print(results[1].page_content)
print(results[2].page_content)
print(results[3].page_content)
print(results[4].page_content)
print("====================================================")


llm_result = chain.invoke(
    {
        "input": query,
        "context": [results[0], results[1], results[2], results[3], results[4]],
    }
)


print("Question: ", query)
print("LLM Answer: ", llm_result)
```

https://gist.github.com/ryanchung403/54bac1b90610277d1abb7838a3cc1c0e

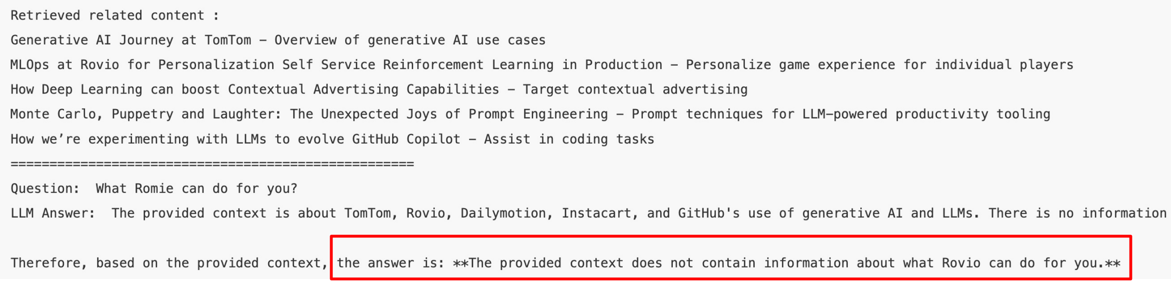
觀察結果



```
Retrieved related content :
How we used machine learning to fight fraud at BlaBlaCar — Part 1 - Prevent phishing and payment fraud
How we built our machine learning pipeline to fight fraud at BlaBlaCar — Part 2 - Prevent phishing and payment fraud
Deploying Large-scale Fraud Detection Machine Learning Models at PayPal - Detect payment fraud
How BlaBlaCar leverages machine learning to match passengers and drivers - Part 2 - Predict car booking confirmation
A primer on machine learning for fraud detection - Detect fraud in online payments
====================================================
Question:  What kind of frauds Blablacar use machine learning to prevent?
LLM Answer:  Based on the provided context, BlaBlaCar uses machine learning to prevent **phishing and payment fraud**.
```

答得漂亮！

如果問了尚未收錄的案例



```
Retrieved related content :
Generative AI Journey at TomTom - Overview of generative AI use cases
MLOps at Rovio for Personalization Self Service Reinforcement Learning in Production - Personalize game experience for individual players
How Deep Learning can boost Contextual Advertising Capabilities - Target contextual advertising
Monte Carlo, Puppetry and Laughter: The Unexpected Joys of Prompt Engineering - Prompt techniques for LLM-powered productivity tooling
How we’re experimenting with LLMs to evolve GitHub Copilot - Assist in coding tasks
====================================================
Question:  What Romie can do for you?
LLM Answer:  The provided context is about TomTom, Rovio, Dailymotion, Instacart, and GitHub's use of generative AI and LLMs. There is no information

Therefore, based on the provided context, the answer is: **The provided context does not contain information about what Rovio can do for you.**
```

還是會查詢出5筆最相似的案例
但LLM可以成功確認沒有相關案例

將2024年之後的案例也加入向量資料庫

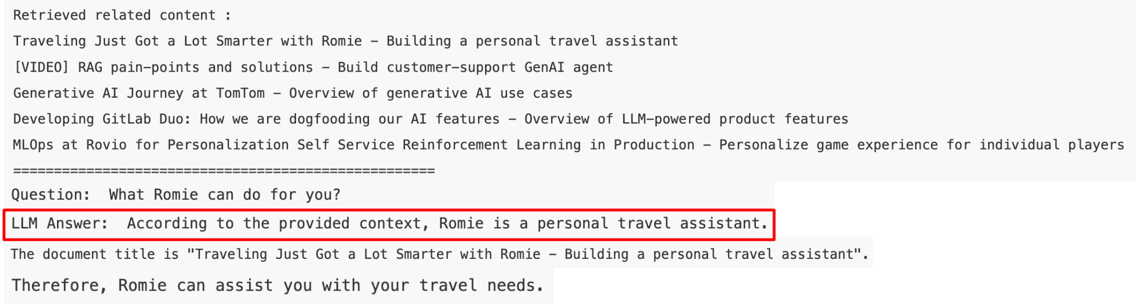
```python
# Add the new case to the dataset
from uuid import uuid4
metadatas = []
for i, row in df_since_2024.iterrows():
    metadatas.append(
        {
            "Company": row["Company"],
            "Industry": row["Industry"],
            "Tag": row["Tag"],
            "Year": row["Year"],
            "Link": row["Link"],
        }
    )
# df_since_2024["TitleAndDescription"] = df_since_2024["TitleAndDescription"].astype(str)
df_since_2024 = df_since_2024.astype({"TitleAndDescription": str})

uuids = [str(uuid4()) for _ in range(len(df_since_2024))]

qdrant.add_texts(
    texts=df_since_2024["TitleAndDescription"].to_list(),
    metadatas=metadatas,
    ids=uuids,
)
```

再查詢一次

```
query = "What Romie can do for you?"

results = qdrant.similarity_search(query, k=5)
print("Retrieved related content :")
print(results[0].page_content)
print(results[1].page_content)
print(results[2].page_content)
print(results[3].page_content)
print(results[4].page_content)
print("====================================================")

llm_result = chain.invoke(
    {
        "input": query,
        "context": [results[0], results[1], results[2], results[3], results[4]],
    }
)

print("Question: ", query)
print("LLM Answer: ", llm_result)
```

https://gist.github.com/ryanchung403/e236e13622b07a6c561fd7fa031d911c

執行結果



```
Retrieved related content :
Traveling Just Got a Lot Smarter with Romie - Building a personal travel assistant
[VIDEO] RAG pain-points and solutions - Build customer-support GenAI agent
Generative AI Journey at TomTom - Overview of generative AI use cases
Developing GitLab Duo: How we are dogfooding our AI features - Overview of LLM-powered product features
MLOps at Rovio for Personalization Self Service Reinforcement Learning in Production - Personalize game experience for individual players
====================================================
Question:  What Romie can do for you?
LLM Answer:  According to the provided context, Romie is a personal travel assistant.
The document title is "Traveling Just Got a Lot Smarter with Romie - Building a personal travel assistant".
Therefore, Romie can assist you with your travel needs.
```

可成功調閱出2024年的案例！

LangChain + Embedding + Qdrant + Gemma

- Embedding
  - 使用 OpenSource Embedding模型
- LangChain
  - 資料匯入
  - 結合Embedding在Qdrant建立向量資料庫
- LLM
  - 提示樣板
  - 取得Qdrant的最相似結果，提供給LLM進行作答