

# RAG System Explanation

Sim Wang - University of Southern California

<https://github.com/simwang-codes>

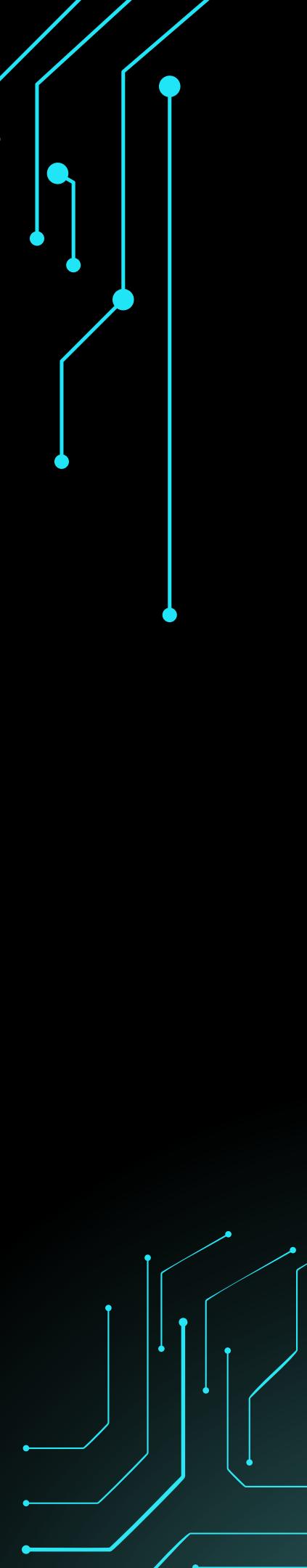
# Project Overview

1. What is RAG
2. Workflow/Data Pipeline
3. Result Demonstration

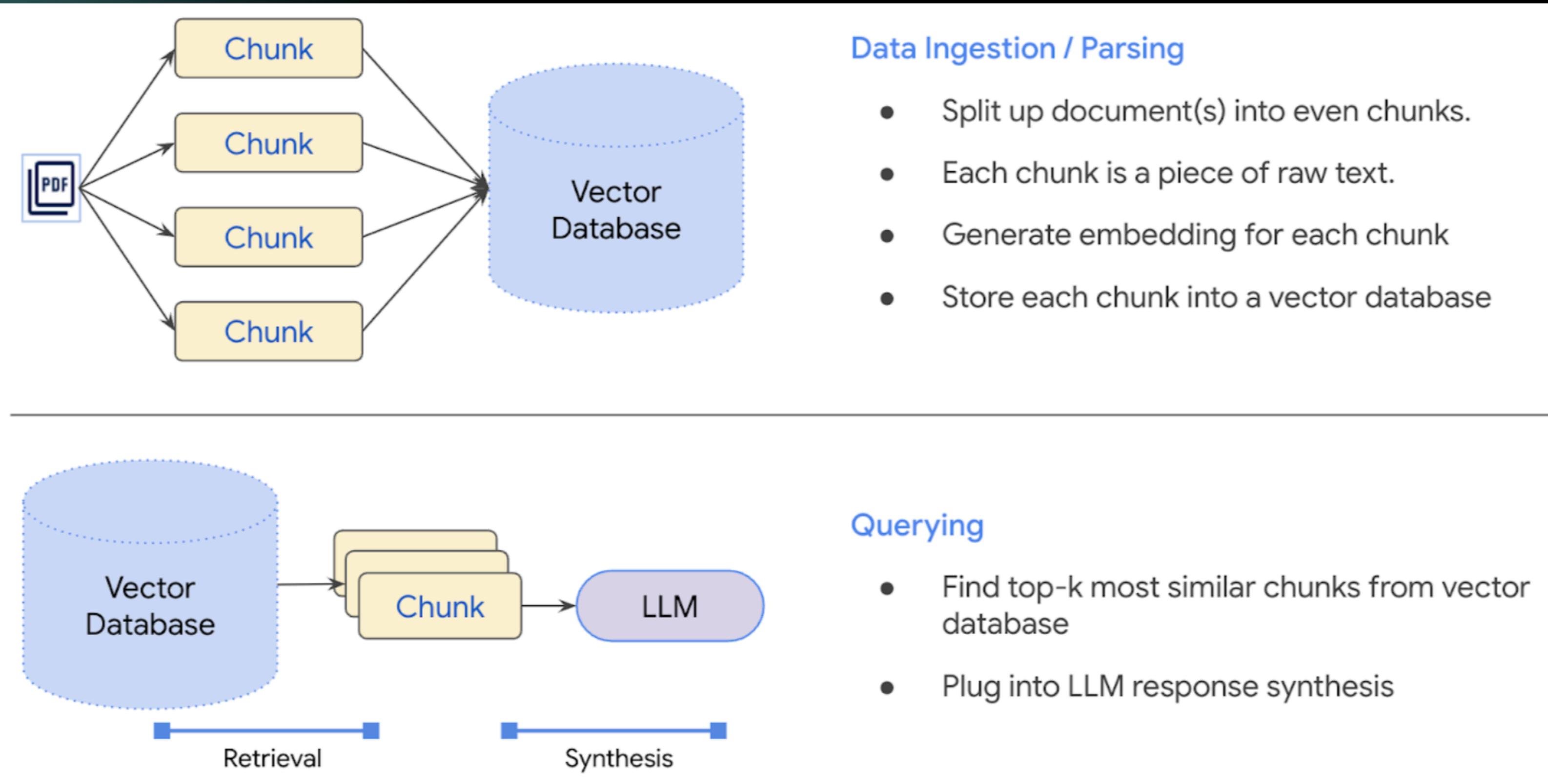


# RAG(Retrieval-Augmented Generation)

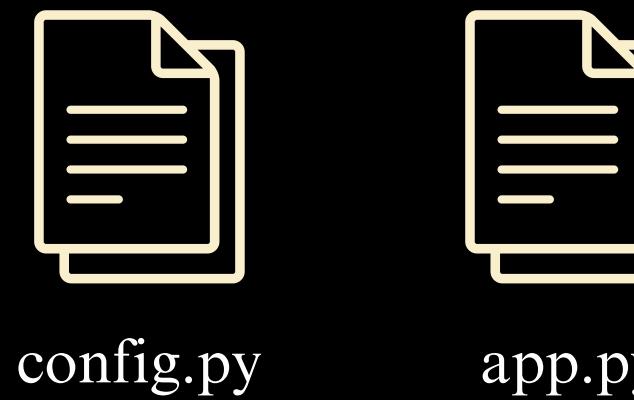
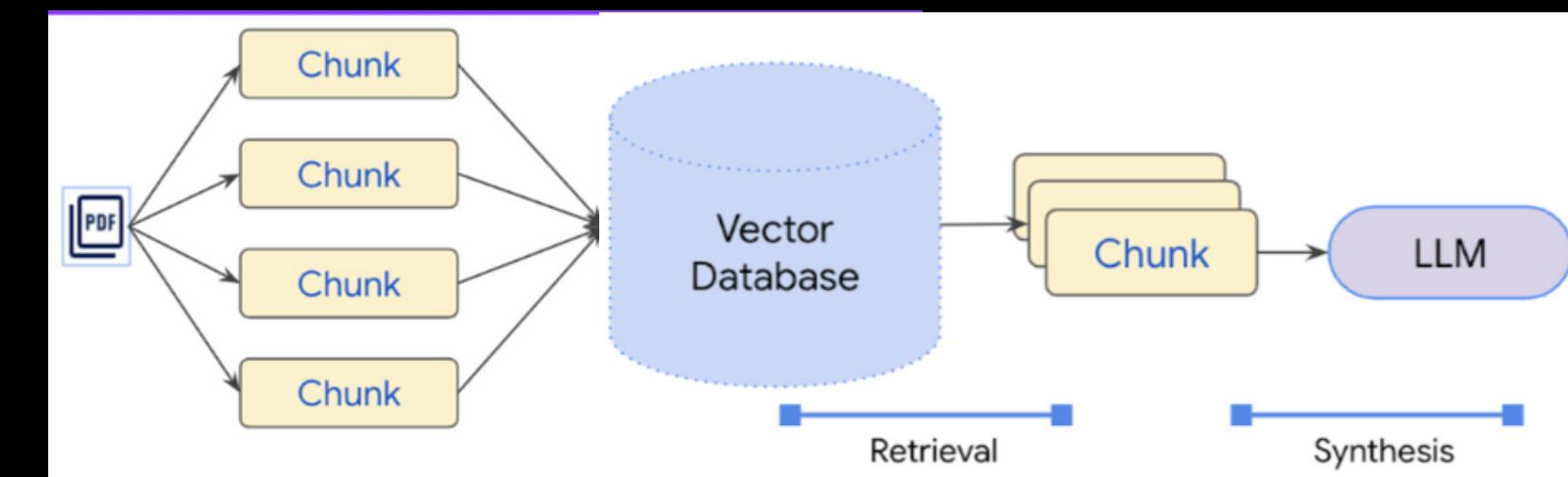
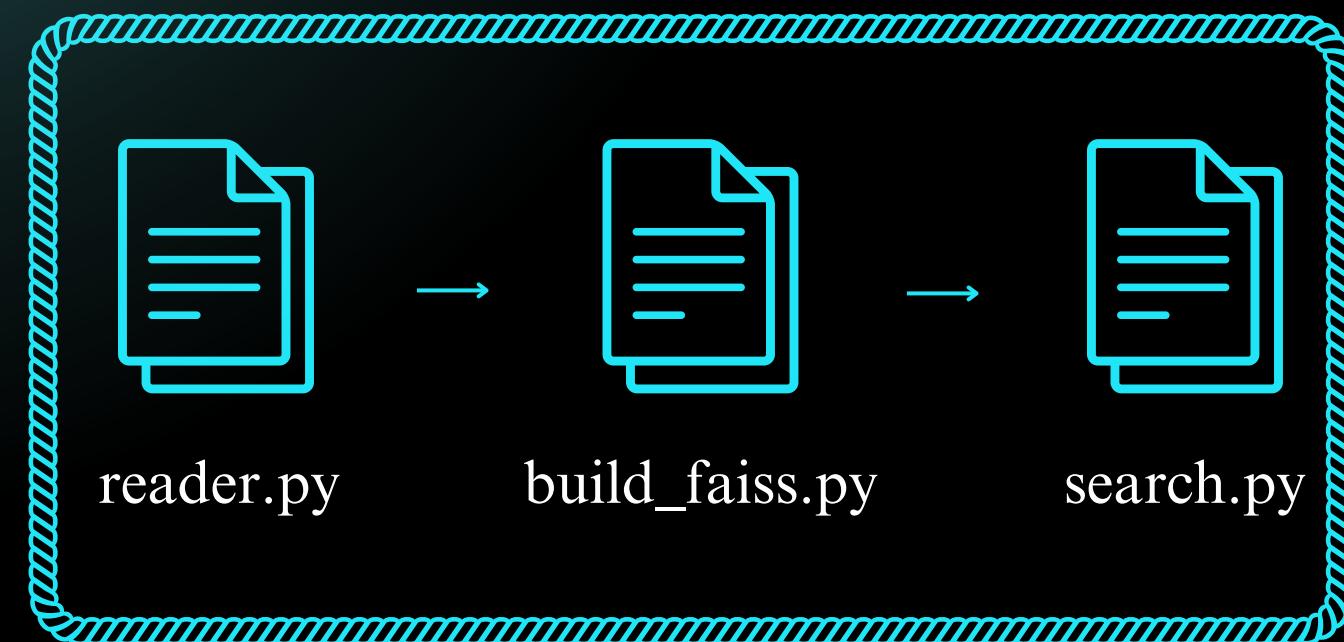
LLMs	External Data	AI-powered Q&A Chatbot
OpenAI gpt-4	A book in .pdf/.txt	Answer any questions about book's content
Google Gemini	Corporate/Product Policy documents	A customer service Chatbot
...	...	...



# RAG Workflow



# RAG Data Pipelines



config.py Stores db\_path and API key  
app.py runs Streamlit UI

# reader.py

```
import os
import sqlite3
import fitz
import jieba
import nltk
import re
from tqdm import tqdm #用来生成进度条
from config import DB_PATH
from langdetect import detect
from collections import Counter
from keybert import KeyBERT
from sentence_transformers import SentenceTransformer
from langdetect.lang_detect_exception import LangDetectException
from langchain.text_splitter import RecursiveCharacterTextSplitter
```

## LangChain:

- text\_splitter

## jieba:

- extract tags for chinese content

## KeyBERT:

- extract tags for languages that have space between words

LangChain is the key for my entire RAG system, its an open source orchestration framework for developing applications that use LLMs

# reader.py

```
def split_text_into_chunks(text, chunk_size = 700, chunk_overlap = 100):
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size = chunk_size,
        chunk_overlap = chunk_overlap,
        separators = ["\n\n", "\n", ".", "!", "!", "!", "?", "?", "?"]
    )
    return text_splitter.split_text(text)

def generate_chinese_tags(text, top_k = 5):
    text = re.sub(r"[\u4e00-\u9fffA-Za-z ]", "", text)
    words = jieba.lcut(text) + nltk.word_tokenize(text)
    counter = Counter(w for w in words if len(w) > 1)
    common = counter.most_common(top_k)
    return [word for word, _ in common]
```

id	chunk_index	chunks	tags
Filter	Filter	Filter	
84	83	冬十月，军还洛阳。[111]孙权遣使上书，以讨关羽自效。王自洛...	关羽,魏王,洛阳,曹仁,刘备
85	84	冬季十月，魏军回到洛阳。孙权派遣使节向魏王上书，表示愿意用...	魏王,关羽,洛阳,孙权,到达
553	552	太祖征讨马超时，让曹仁兼安西将军，统领各将在潼关和马超对抗...	曹仁,将军,关羽,马超,打败
877	876	！公卿已将各位比作前代的恶人了。”王肃因与祭祀宗庙的事有牵...	王肃,后来,关羽,司马,景王
936	935	及关羽围曹仁于樊，孙权遣使辞以“遣兵西上，欲掩取羽。江陵、...	太祖,群臣,渡江,陛下,关羽
937	936	等到关羽在樊城包围了曹仁时，孙权派遣使者致辞曹操：“吴军准...	关羽,孙权,保密,泄露,樊城
949	948	黄初元年，以晔为侍中，赐爵关内侯。诏问群臣令料刘备当为关羽...	陛下,群臣,关羽,未有,江湖
952	951	黄初元年，任命刘晔为侍中，赐关内侯爵位。此时，关羽被吴军所...	文帝,关羽,刘备,他们,吴国
961	960	！有这种事，那就是我不了解人了。这一定是愚民乐于兴乱，胡乱...	孙权,关羽,太祖,蒋济,樊城
1136	1135	建安二十四年，太祖在长安，使曹仁讨关羽于樊，又遣禁助仁。秋...	关羽,太祖,文帝,孙权,于禁
1189	1188	侯音、卫开等人在宛城反叛，庞德率领自己的部下与曹仁共同攻占...	庞德,关羽,樊城,投降,部下
14...	1439	关羽围征南将军曹仁于樊。俨以议郎参军军事南行，与平寇将军徐...	隔绝,北军,关羽,征南将军,曹仁
1441	1440	关羽在樊城包围征南将军曹仁。赵俨以议郎之职参与曹仁军务而南...	关羽,曹仁,我们,现在,孙权
1580	1579	汉献帝建安十三年，他跟随太祖征讨荆州。太祖大军返回，留下满...	满宠,关羽,将军,樊城,太祖

Read file → Split into chunks + Extract keywords/tags → Store into sqlite db

# build\_faiss.py

```
import sqlite3
import os
from tqdm import tqdm
from config import DB_PATH, OPENAI_API_KEY
from langchain.embeddings import OpenAIEMBEDDINGS
from langchain.vectorstores import FAISS
from langchain.schema import Document # 当在未来处理多个documents时,
```

```
os.environ["OPENAI_API_KEY"] = OPENAI_API_KEY # This is an en
```

```
embedding_model = OpenAIEMBEDDINGS() # Creates an embedding m
```

```
def load_all_from_db(db_path = DB_PATH): ...
```

```
def build_faiss_index(document, save_path = "vector_index"):
```

## Document:

- wrap a single chunk, and its tags as metadata, into LangChain document

## OpenAIEMBEDDINGS:

- vectorizing LangChain document's contents

## FAISS (FB AI Similarity Search):

- store vectorized content in FAISS index structure for similarity search

'Cats are cute.' → Token IDs: [9645, 389, ...] → [0.0123, -0.0197, 0.0042, 0.0001]  
Number of tokens: 4 Total vector dimensions: 1536

# search.py

```
import os
import langid
from config import DB_PATH, OPENAI_API_KEY
from langchain_openai import OpenAIEMBEDDINGS, ChatOpenAI
from langchain_community.vectorstores import FAISS
```

```
> def load_vector_storage(path="vector_index"):: ...
# k=5 means retrieve the top 5 similair chunks
> def search_similar_chunks(query, k=5):: ...
# Originally, I used LangChain's detect method to identify
# so I switched to langid, which is a more accurate langua
> def detect_language(text):: ...
> def answer_question_with_prompt(query, retrieved_chunks):: ...
```

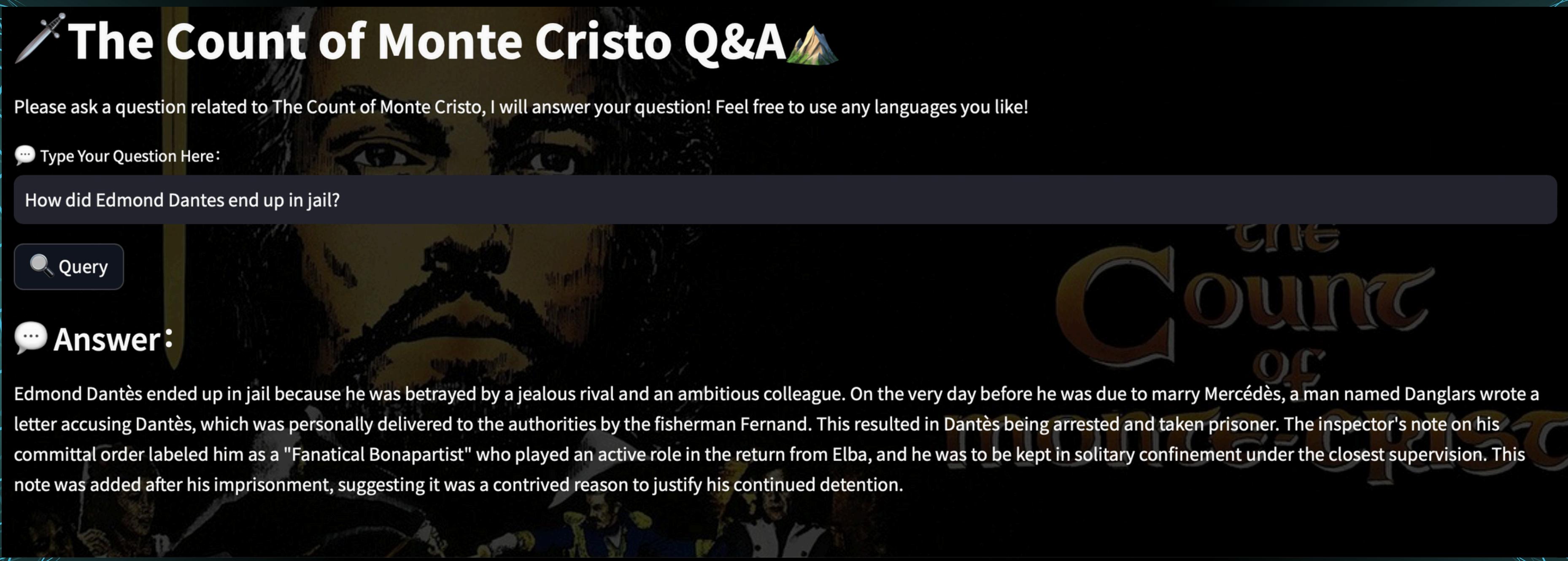
## .similarity\_search:

- embed user's query, then search top 5 similair chunks in vector\_index

```
# Prompt starts below:
language = detect_language(query)
prompt = f"""You are an intelligent AI assistant built for my RAG Chatbot syste
Your duty is to answer the user's question based on the content and tags below,
and you need to answer the question with this language the user is using: {lang
Guidelines for you:
1. Base your answer ONLY on the provided content and tags, however, if the ans
2. If relevant information exists in any form, do your best to provide a helpfu
3. Only if the content is indeed completely irrelevant with user's question, re
4. If user ask anything about you, for example:"Who are you" or "What is your n
Tags:
{joined_tags}
Content Chunks:
{context_chunks}
User's Question: {query}
Answer:"""
```

input query → embed query → FAISS similarity search + prompt = answer

# Result Demonstration



The Count of Monte Cristo Q&A

Please ask a question related to The Count of Monte Cristo, I will answer your question! Feel free to use any languages you like!

Type Your Question Here:

How did Edmond Dantes end up in jail?

Query

Answer:

Edmond Dantès ended up in jail because he was betrayed by a jealous rival and an ambitious colleague. On the very day before he was due to marry Mercédès, a man named Danglars wrote a letter accusing Dantès, which was personally delivered to the authorities by the fisherman Fernand. This resulted in Dantès being arrested and taken prisoner. The inspector's note on his committal order labeled him as a "Fanatical Bonapartist" who played an active role in the return from Elba, and he was to be kept in solitary confinement under the closest supervision. This note was added after his imprisonment, suggesting it was a contrived reason to justify his continued detention.

# Result Demonstration

## Three Kingdoms Q&A

Please ask a question related to Three Kingdoms, I will answer your question! Feel free to use any languages you like!

Type Your Question Here:

讲一下诸葛亮的故事，并把你讲的故事编辑成一段社交媒体帖子，用来吸引没读过三国志的人

Query

**Answer:**

诸葛亮，三国时期蜀汉的丞相，以智谋和忠诚著称。他不仅在军事和政治上有深远的影响，更因其品德和才能受到后世的推崇。有一次，诸葛亮将一个自负的官员廖立贬为平民，并将其流放到汶山郡，因为廖立批评先帝，贬低其他官员，并对国家的强大嗤之以鼻。诸葛亮的这种行为展现了他对于国家法纪和道德标准的严格要求。在他去世后，人们深深怀念他，他的言行和教诲一直被人们传颂。他像周公一样，用简洁而有力的言辞指导人们，虽然他的文辞不华丽，却充满了实用的智慧。

社交媒体帖子： ★【三国智慧传奇——诸葛亮】★ 追忆历史的烟云，有一位名垂青史的智者，他的故事激励了无数人——诸葛亮，蜀汉的辅政大臣，一个集智慧、忠诚与公正于一身的传奇人物。 ↗

他不仅是战略大师，更是治国理政的楷模。他严格法纪，对国家忠心耿耿，即使面对挑战与非议，也不改其公正。 ⚖

诸葛亮去世后，民间流传着他的故事和言论，就像古代的《甘棠》歌颂英雄一样，他的智慧和精神至今仍在人们心中。 ✨

想要了解更多关于这位三国时期的智者的故事吗？跟随我们探索诸葛亮的人生足迹，感受那个时代的智慧与魅力！ 📚✨

#诸葛亮 #三国志 #历史智慧 #传奇人物

# Result Demonstration

## Journey to the West Q&A

Please ask a question related to Journey to the West, I will answer your question! Feel free to use any languages you like!

Type Your Question Here:

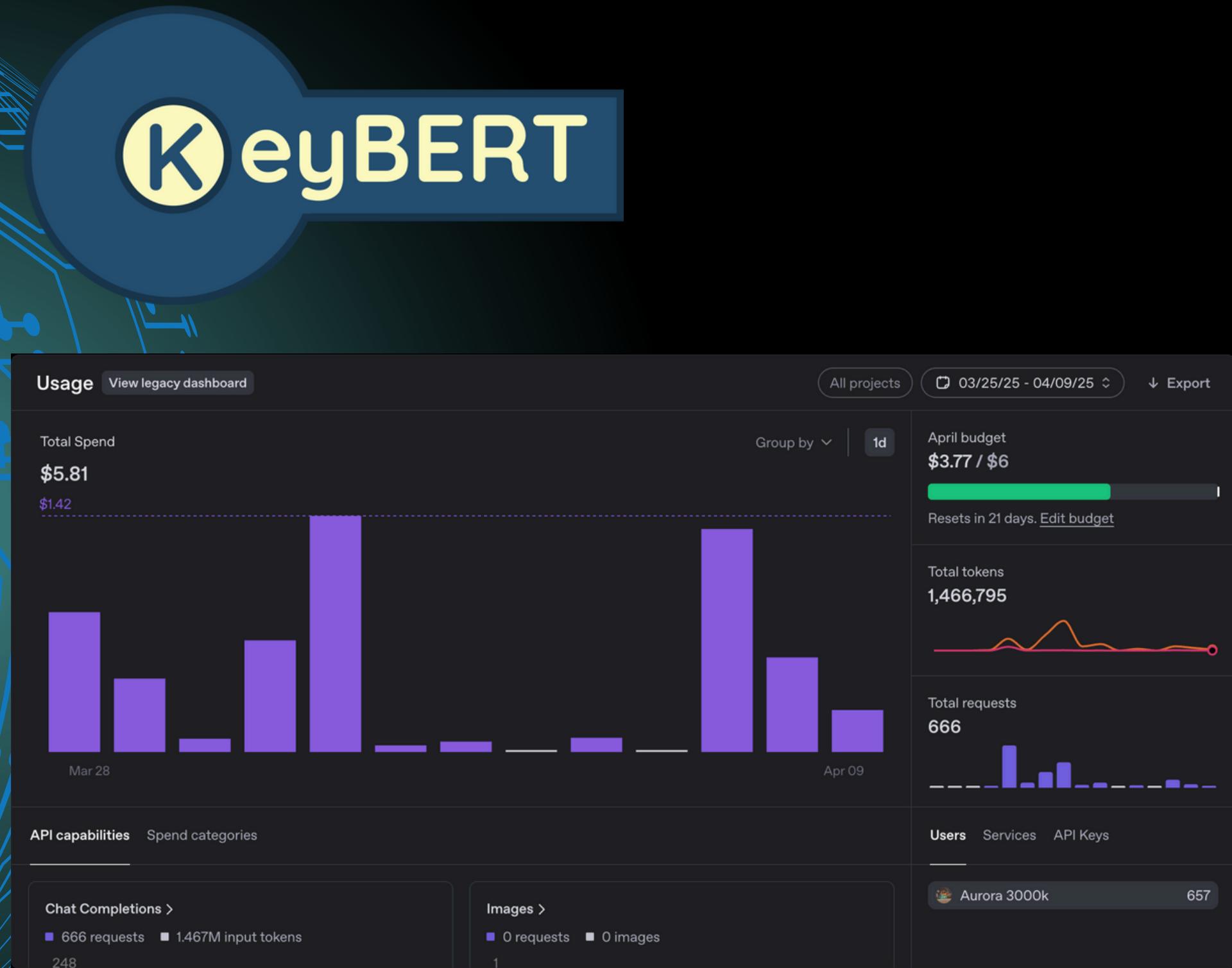
なぜ孫悟空は仏陀によって五本指山の下に鎮圧されたのでしょうか?

Query

Answer:

孫悟空が五指山の下に押さえられたのは、彼が天界で大暴れし、神々に反逆したためです。内容には直接的な記述がありませんが、タグと内容の断片から推測すると、孫悟空が天界での乱を起こした後、仏陀によって制服され、五指山の下に封じ込められたことを示唆しています。これは、彼が乱を起こした後の罰として、五指山の下に封じ込められたことを意味します。また、土地神や山神が孫悟空を封じ込む手伝いをしたことも示されています。

# Challenges



## KeyBERT is slow:

- as it first embed chunk, then use vector index to extract keywords

## gpt-4 is slow and expensive:

- takes longer time than less complex gpt models

## Scalability:

- if you have 40 million words to process, it will be slow

# Thank You

Sim Wang - University of Southern California