

RAG-Powered Real Estate Matching

A dark blue background featuring a stylized, glowing blue wave pattern composed of numerous thin lines and small white dots, resembling a digital or neural network signal.

April 4, 2025

Jesus Alcocer Tagua

Let's Get Started →



Table of contents

- **Intro:** AI in Real Estate & Personalized Recommendations
- **API Calls:** Basic Matching (Pros & Cons)
- **RAG:** Advanced Personalization & Real-Time Data
- **Comparison:** API vs. RAG Use Cases
- **Demo:** Client Matching - API vs. RAG
- **Conclusions**
- **Q&A**

1. Chat with Real Estate Data
2. Table of contents
3. The Goal
4. LLM API Calls
 1. Improving API Calls
5. API Workflow
6. RAG Concepts
7. Implementing a RAG
8. Do More with RAGs
9. References
10. Wrapping Up
11. Conclusions
12. Demo - Real Estate Agents
13. QnA
14. Thanks

The Goal

Creating a Real Estate Agent to propose the best solution to property seekers.

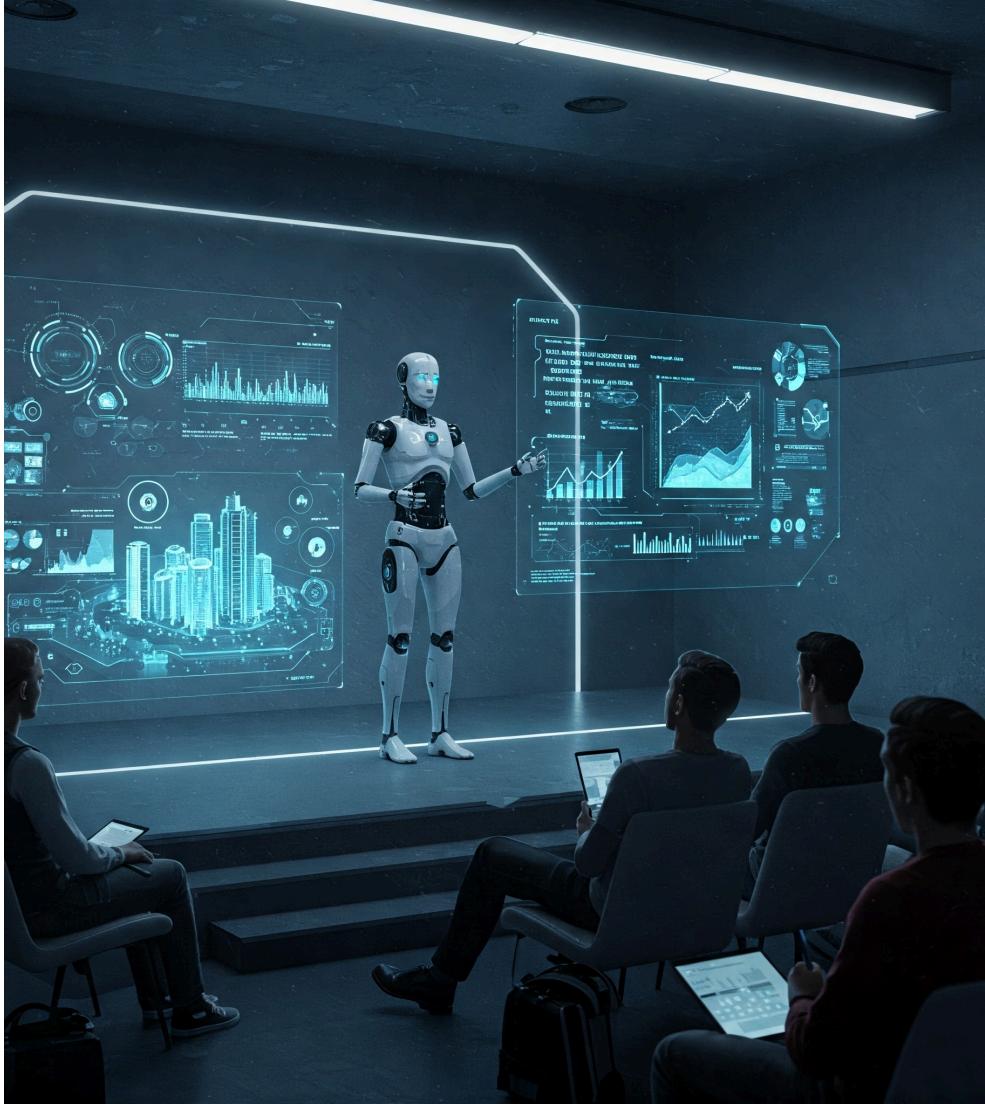
While understanding:

1. How to provide context to LLMs via API calls
2. How to setup a RAG as Real Estate Agent

By the end of the, we will have a recap with
Pros and Cons of each approach.

Code and Demo, will also be shared.

See the Code



LLM API Calls

Use code snippets and get the highlighting directly, and even types hover!

```
import os
from openai import OpenAI

client = OpenAI(
    # This is the default and can be omitted
    api_key=os.environ.get("OPENAI_API_KEY"),
)

chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "Can you recommend me proper"
        }
    ],
    model="gpt-4o-mini",
    temperature=0.3,
)

completed_message = chat_completion.choices[0].message
print(completed_message)
```



Improving API Calls

Lets add some context to our API calls!

Even simple API calls to LLMs can have interesting performance.

See an example:

```
1 # Step 3 - Query the LLM with better context
2 chat_completion = client.chat.completions.create(
3     messages=[
4         {
5             "role": "system",
6             "content": f"""{SYSTEM_CONTENT_OpenAI}.
7                         These are the properties we have:
8                         {properties_must_have}
9                         """,
10            },
11            {"role": "user", "content": prompt}
12
13        ],
14        model="gpt-4o-mini",
15        temperature=0.8,
16    )
```

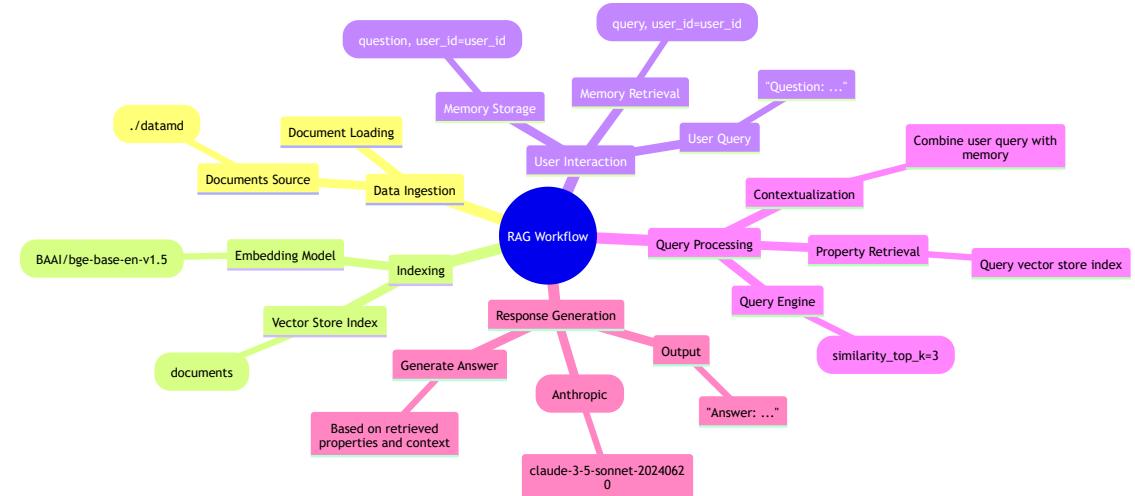
API Workflow

You can create diagrams / graphs from textual descriptions, directly in your Markdown.



RAG Concepts

You can create diagrams / graphs from textual descriptions, directly in your Markdown.



Implementing a RAG

See an example here [here](#)



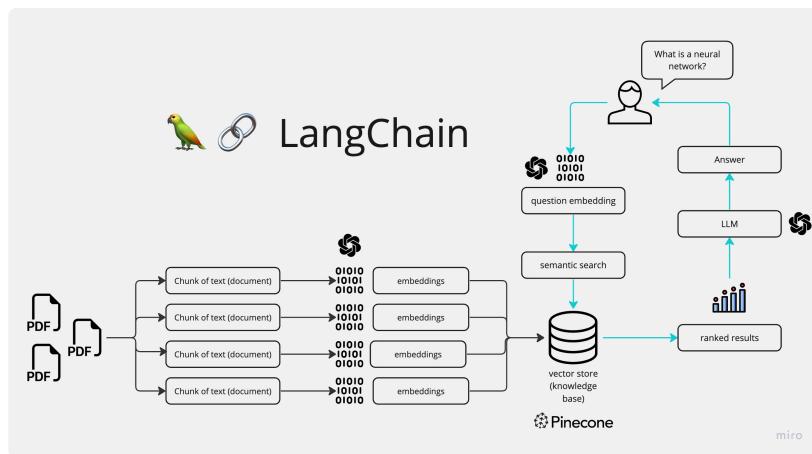
Do More with RAGs

RAGs can have many more use cases. From Chatting with a DB, to talk with PDF or CSV data.

Use RAG to chat with your DataBase



Use RAG to chat with PDFs



Read more about RAG use cases.

References

Get your API for LLMs:



1. OpenAI

- <https://openai.com/api/>
- <https://platform.openai.com/docs/models/#current-model-aliases>

2. Anthropic

- <https://console.anthropic.com/>
- <https://docs.anthropic.com/en/docs/about-claude/models/all-models>

Learn more on how to use Different LLMs with
Python

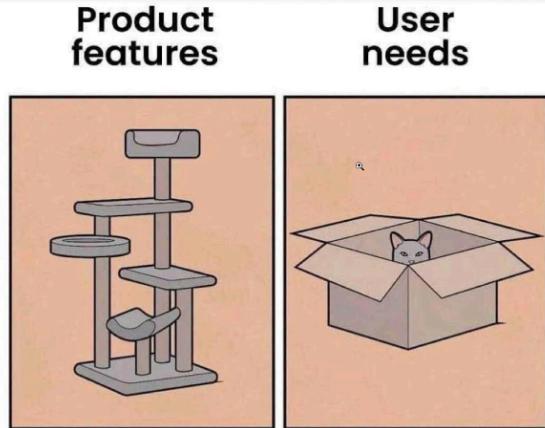
Wrapping Up

Feature	API Calls (Summary)	RAG with LlamaIndex
Speed/Efficiency	Fast, retrieves pre-summarized data.	Slower, requires retrieval and processing of documents.
Context/Detail	Limited context, concise summaries.	High context, detailed and nuanced responses.
Complexity of Queries	Handles simple, structured requests.	Handles complex, unstructured, and ad-hoc queries.

Conclusions

RAGs can have many more use cases. From Chatting with a DB, to talk with PDF or CSV data.

Use RAG to chat with your DataBase



Use RAG to chat with PDFs



Read more about RAG use cases.

Demo - Real Estate Agents

Demo Time!

QnA



Thanks

Learn More about Real Estate Powered RAG

[Youtube](#) · [GitHub](#) · [Tech Blog Post](#)