# Retrieval Augmented Generation with LangChain and OpenAI

By Shao Hang HE

# A bit about myself

I'm Shao Hang He

I'm a senior developer with 11 years of experience.

I’ve used OpenAI's API for 3 years.

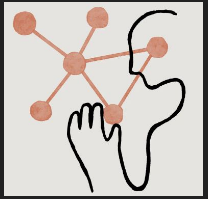
I'm also a business owner running my own Development and LLM consulting firm.

# Why LLMs are popular now?

- No need of a training set
- Reasonable pricing
- Available in all languages

Gemini

Llama



# OpenAI's GPT models

- Powerful
- affordable
- accessible

Get an API key:
https://help.openai.com/en/articles/4936850-where-do-i-find-my-openai-api-key

# LangChain

- Popular Wrapper for OpenAI and other LLMs
- Available in Python and JavaScript
- Open source

Check the documentation there:
https://python.langchain.com/docs/get_started/introduction

# Prompt engineering

The most basic way to use OpenAI's GPT models is to ask simple questions.

**Prompt:** Explain what is a dev conference in 2 sentences.

**Output:** A developer conference is an event where software developers, programmers, and tech enthusiasts gather to learn about the latest technologies, programming languages, and industry trends through workshops, talks, and networking opportunities. These conferences serve as platforms for sharing knowledge, fostering collaboration, and exploring innovations in the realm of software development.

# Chat Models

- Great for generating messages
- Less deterministic (Not ideal for a specific format)
- Ideal for chat applications like ChatGPT

# Completion Models

- Great for true and false answer
- Deterministic answers
- Ideal for LLM chains

LLM chains using LangChain:
https://python.langchain.com/docs/modules/chains

# Few-shot learning

Since LLMs are pretrained, we only need to guide it with a few examples in most cases.

**Prompt:** Here is an email I received today: “It was nice meeting you at Confoo. Can you tell me more about your company?”. Write a reply to this message.
Here are some examples of my previous reply:

1. It was nice to meet you at CollisionConf, my company provide dev services to small and medium enterprises.
2. Great meeting you at the Web Summit! I am doing AI consulting now! Let’s discuss in detail next week?

# Ways to extend the model with more data

When you have too much data for the LLM prompt context, you would need other ways to extend the LLM’s knowledge:

- Fine tuning: Supervised learning with well formatted training set
- Retrieval Augmented Generation (RAG): Search algorithm on existing unstructured data such as PDFs

# Fine tuning

- Similar to traditional supervised learning
- Great if you have a lot of examples
- Shorter prompts for a well fine tuned model

# Challenge with Fine tuning

- Time consuming to gather data.
- Required large amount of data to be effective
- Data privacy issues if trained with real user data

# Retrieval Augmented Generation (RAG)

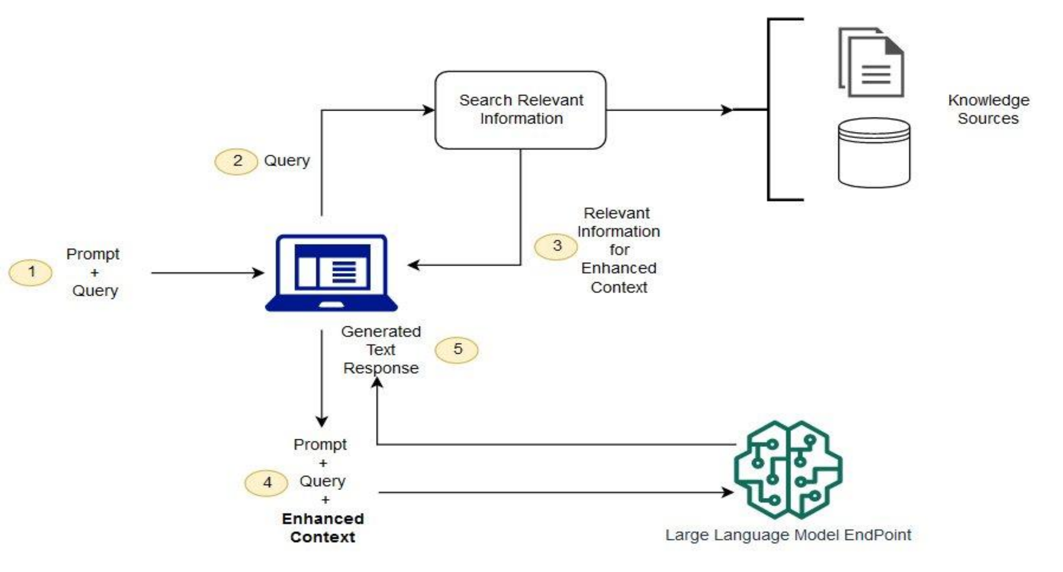
- Augment your prompt with external data
- Work with existing unstructured data such as PDF files and web pages.
- No need to create a training set.
- Work with other LLMs

# Vector Embeddings

- Vector embeddings are a mathematical representation of the text data.
- Use OpenAI's Ada model to generate embeddings
- Turn your unstructured text data into vector embeddings
- Turn the user query into a vector embedding.

# Basic RAG pipeline

Find the vector with the closest distance to the query vector. We can use the search algorithm from OpenAI.


Search Relevant Information
Knowledge Sources
2 Query
1 Prompt + Query
3 Relevant Information for Enhanced Context
Generated Text Response 5
4 Prompt + Query + Enhanced Context
Large Language Model EndPoint

# Vector Databases

In a real application, you would need to store the vector embeddings in a vector database. This way, we don't need to turn the documents into embeddings every time.

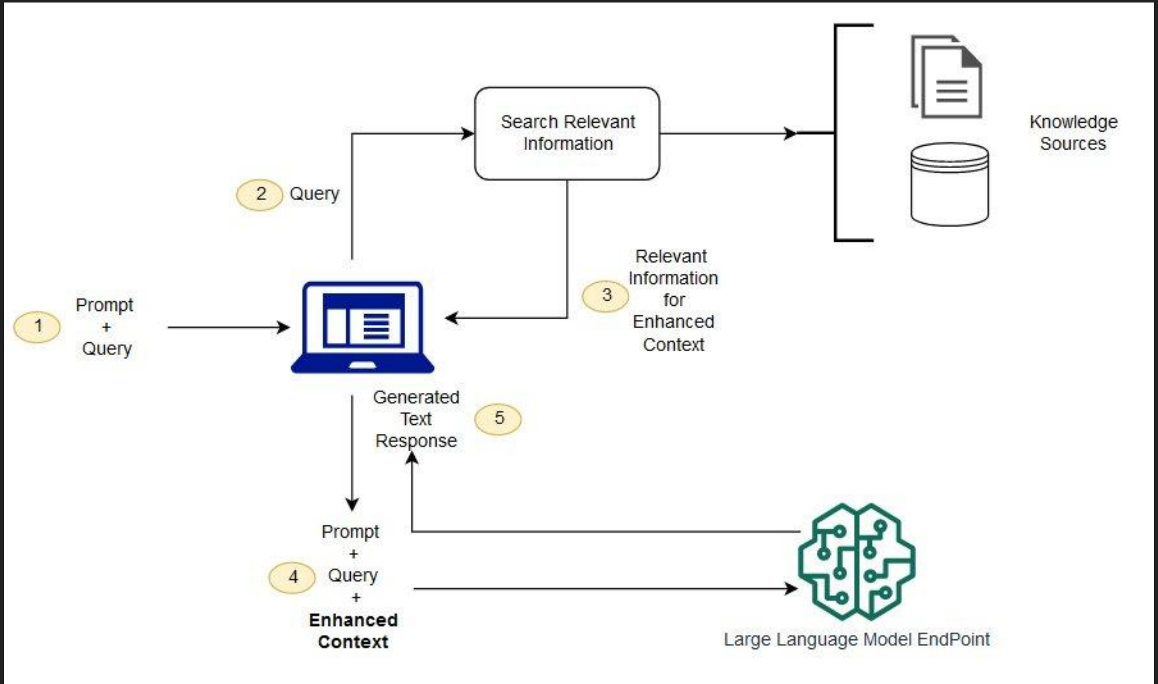
You can you a classic database that support vector embeddings such as PostgreSQL and Redis.

Dedicated Vector Databases:

- Pinecone
- Chroma
- Weaviate

# Prompt engineering with RAG

Once you get the matched embeddings, you can use it to feed them into your prompt.


Search Relevant Information
Knowledge Sources
2 Query
1 Prompt + Query
3 Relevant Information for Enhanced Context
Generated Text Response 5
4 Prompt + Query + **Enhanced Context**
Large Language Model EndPoint

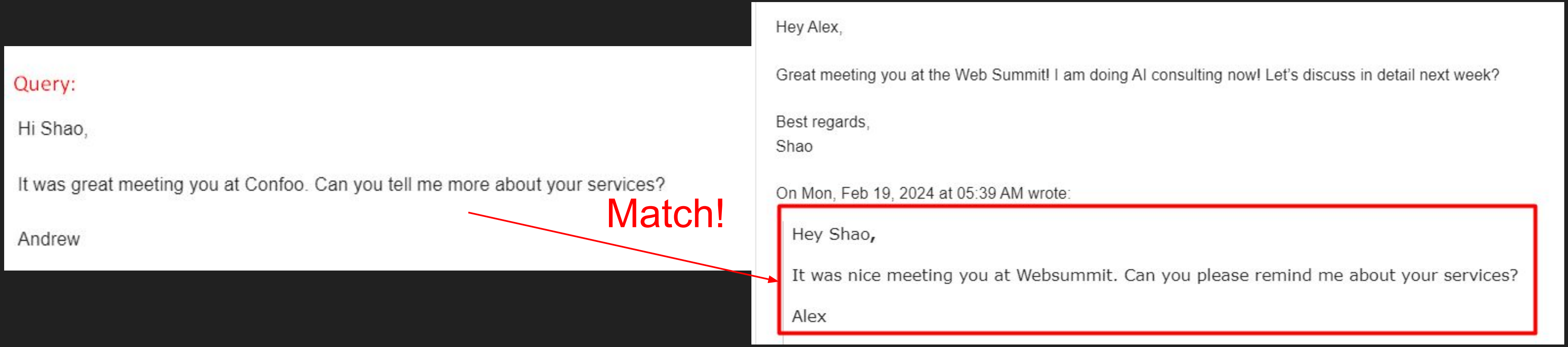
# Example application Mail Magic

Mail Magic uses RAG to generate draft email replies for the user. It can write email draft while you are sleeping! Please check https://mailmagic.ai/

- Read your past emails and turn them into vector embeddings.
- Turn the incoming email (query) into a vector embedding.
- Run search algorithm to extract the matched vectors.
- Generate enhanced prompt with the query and matched vectors.
- Give the prompt to the LLM (GPT or Llama)

# Mail Magic RAG

A past message in sent inbox contain the reply I sent and the original message in quote. We can use the new message as query and RAG will find the messages that have a close match with the original message.



Query:

Hi Shao,

It was great meeting you at Confoo. Can you tell me more about your services?

Andrew

Match!

Hey Alex,

Great meeting you at the Web Summit! I am doing AI consulting now! Let's discuss in detail next week?

Best regards,
Shao

On Mon, Feb 19, 2024 at 05:39 AM wrote:

> Hey Shao,
>
> It was nice meeting you at Websummit. Can you please remind me about your services?
>
> Alex

# Mail Magic prompt engineering

Get the similar answers from the past emails and use them as few-shot examples.

**Prompt:** Here is an email I received today: “It was great meeting you at Confoo. Can you tell me more about your services?”. Write a reply to this message.
Here are some examples of my previous reply:

1. Great meeting you at the Web Summit! I am doing AI consulting now! Let’s discuss in details next week?
2. It was nice to meet you at CollisionConf, my company provide dev services to small and medium enterprises.

# Other RAG applications

- Chat with PDF
- Website FAQ bot
- Customer service bot
- Personal AI assistant
- Recommender System

# Resources

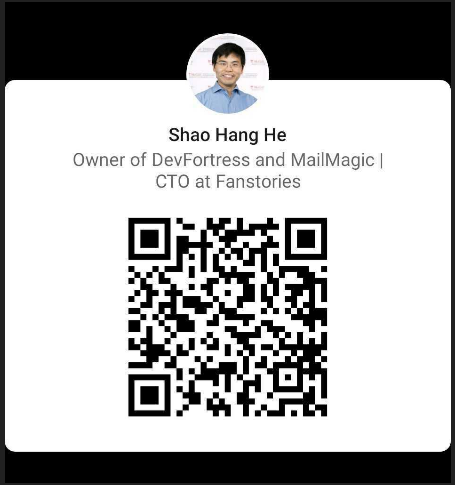
- OpenAI: https://openai.com/
- Langchain: https://www.langchain.com/
- Deeplearning.ai: https://www.deeplearning.ai/short-courses/building-evaluating-advanced-rag/
- Hugging face: https://huggingface.co/
- Mail Magic AI: https://mailmagic.ai/

# Thank You!

Please let me know if you have any questions!

My email: shaohang.he@devfortress.com

Add me on linkedin: https://www.linkedin.com/in/shao-hang-he/



Shao Hang He

Owner of DevFortress and MailMagic |
CTO at Fanstories