AI Food Advisor with LangChain and Pinecone

Matheus Ferreira · Follow

5 min read · Oct 13, 2023

Building an LLM-powered application in Streamlit step-by-step

Generated with Gencraft

Large Language Models (LLMs) have become a reality in our daily lives, and there's a rush between companies to launch profitable products from those models.

Inspired by this great article, I've decided to develop an LLM-powered application to tackle an everyday issue: what should I eat today? The app

allows the user to input his preferences on a human-based language and to yield the recipes that most fit its desires.

The project utilizes a sample from this dataset containing recipes, nutritional information and reviews from Food.com, and the code repository can be found here.

Dataset

The dataset contains thousands of food recipes, along with their nutritional information.

RecipeId	Name	AuthorId	AuthorName	CookTime	PrepTime	TotalTime	DatePublished	Description	Images
260527	Bat Poop - for Halloween	297076	CandyTX	PT30M	PT20M	PT50M	2007-10-21T00:48:00Z	Tom came up with this for our son's Halloween-...	c("https://img.sndimg.com/food/image/upload/w_...
16512	Lemony Rice Pilaf	20371	Lennie	PT25M	PT10M	PT35M	2001-12-31T10:19:00Z	This takes plain ol' rice to a new level. Goes...	c("https://img.sndimg.com/food/image/upload/w_...
443170	Sugar-Free Blueberry-Corn Muffins - Weight Wat...	57042	internetnut	PT18M	PT15M	PT33M	2010-11-30T20:46:00Z	These fruity muffins get their sweetness from ...	character(0)
164010	Easy & Quick Strawberry Wedding Cake	83060	KEHALI	PT35M	PT15M	PT50M	2006-04-12T16:43:00Z	This is a variation of a cake served at my ste...	character(0)
215927	Vegetable Minestrone	461472	ladybug810	PT20M	PT40M	PT1H	2007-03-09T18:18:00Z	This is my sister's recipe, but I have to shar...	"https://img.sndimg.com/food/image/upload/w_55...

Image by author

There's plenty of information to be processed in natural language, but also metadata from the recipes, such as the preparation time, nutritional information and the necessary ingredients.

Project

Before we go into the code, let's break down the problem into chunks, so we can structure the steps to build the food advisor.

The first step is to store our data in a vector database. For this project, I am using Pinecone, which allows maintaining one database on its free tier.

Once the database is available, we can query it by performing similarity searches. This will be done by the LLM, the core of the application, that will be responsible for understanding the user preferences and retrieving an appropriate query. The LangChain provides a framework for building this bridge.

Now that the background job is done, we need a framework to control this flow. As mentioned, we are using Streamlit on this purpose. So, summing up:

1. Preprocess data.
2. Store data in a vector database.
3. Build LLM pipeline.
4. Build Streamlit application for controlling the job flow.

Preprocessing won't be covered in this article, but you can take a look on what I've done on this notebook.

Vector Database

Vector databases are databases designed to work with LLM embeddings. They store data according to a specific embedding, providing optimized storage capabilities, along with fast querying abilities. In this project we are using Pinecone, which allows one free database in it's free tier plan.

You should first create an application index with the appropriate settings in the Pinecone website, so that you can get your API key. In my case, the vectors storedcontain 1536 dimensions each.

Image by author

With the database environment now set up, let's dive into the details.

We first establish our connections to OpenAI and Pinecone by providing our API keys. The OpenAI embedding instructs Pinecone on how to embed and store our data.

We then turn our data into LangChain Document instances, including the content (the recipe description) and its associated metadata. Before uploading, we should divide our data into chunks of 100s, as pinecone limits the amount of data we can upsert at once.

LLM Chain

LLM's can produce quite useful responses, but they are somehow open by nature, which make it difficult to work with structured data. For this purpose, LangChain provides us a framework to build LLM-powered applications, with several tools and APIs for working with prompts and responses.

On this subject, I can recommend the OpenAI course for prompt engineering. It provides a handful of insights on how to build better and more assertive prompts, besides some techniques to yield better responses.

From this snippet we can see how we built the LLM chain. The chain needs an LLM, a prompt and an output parser. The prompt will be formatted by the user inputs taken from the app form, and by the response format yielded from the output parser.

As we fit this all those variables together, we guarantee a pipeline in which we control the prompt to be used by the LLM, and also the response

structure. The response will be a string that will be used to query the vector database and look for the most similar embeddings in it.

App

The application will provide an interactive interface to manage all the workflow in a human way. I won't share the whole app code in this article, but if you are curious, check it out here.

Finally, let's just take a look on how to put all the inputs together to generate the similarity search.

In my case, I used the OpenAI API, but you could use any LLM you want. From this snippet we can see how we can pass the user preferences as inputs to run our chain. Then all we need to do is to parse the LLM output and use it to perform our similarity search, that will be stored in the docs variable.

You can see the final result down below.

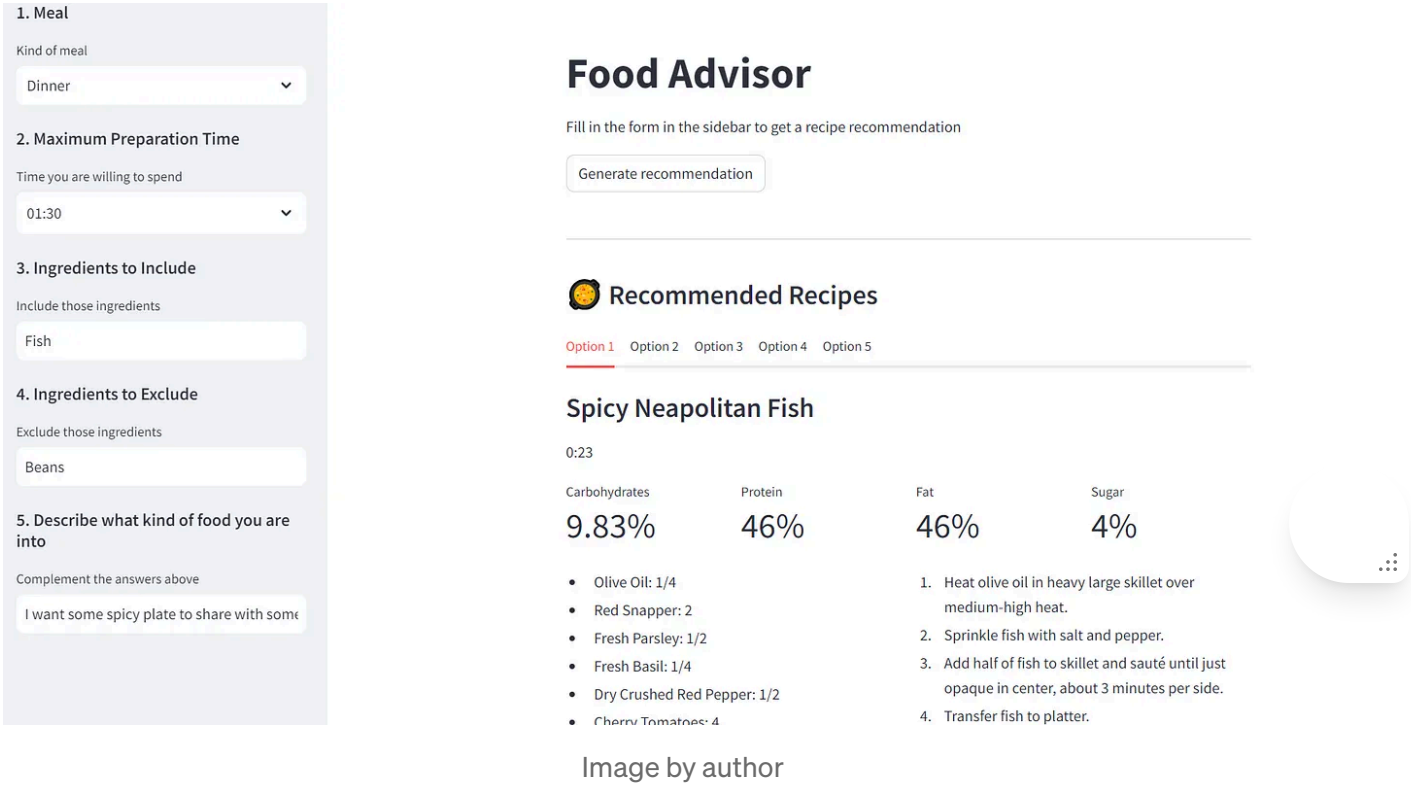


Image by author

Conclusion

In this project we built an LLM-powered application step by step.

First we did some data engineering by preprocessing our data and uploading it to a vector database in Pinecone.

Then we built an LLM chain to structure our prompt and retrieve a structured query to be used to perform similarity searches in our database.

Finally, we built an application to collect the inputs, perform the similarity searches and parse the results in order to show the user the most suitable recipes.

Despite simple, this project explores a powerful application of LLMs, allowing the user to interact with any content in a natural language way.

Websites that hold several amount of data can build this kind of tools for users to navigate their content by ‘chatting’ with an LLM.

The model could be enhanced by taking comments and reviews into account, so that other users’ comments could hold more information about the recipe or the product in question.

Llm | Artificial Intelligence | Data Science | ChatGPT | Python

Written by Matheus Ferreira

31 Followers

Publishing my studies and projects in Data Science and AI - Statistician

Follow

More from Matheus Ferreira

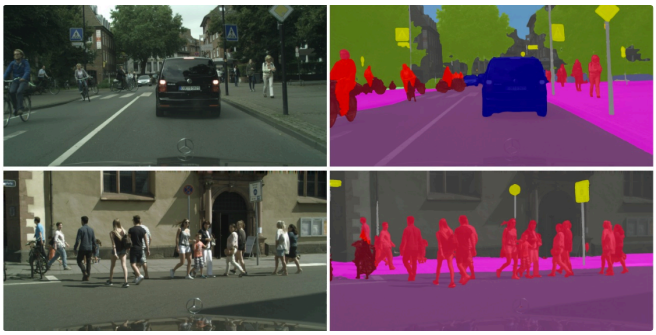
Matheus Ferreira

Forecasting Football Results - Part 2: The Model

Final dataset and data modeling

9 min read · Mar 3, 2023

73



Matheus Ferreira

Video Segmentation with Transformers

Transformers are the backbone of the new Large Language Models (LLMs) that enable...

4 min read · Jun 13, 2023

83 1

Matheus Ferreira

Forecasting Football Results - Part 1: Web Scraping

Python project exploring football results and betting odds

8 min read · Nov 19, 2022

120 1

Matheus Ferreira

Build Your Company RAG Chatbot with LangChain

Walktrough building a chatbot that answers about your company with LangChain.

6 min read · Feb 22, 2024

See all from Matheus Ferreira

Recommended from Medium

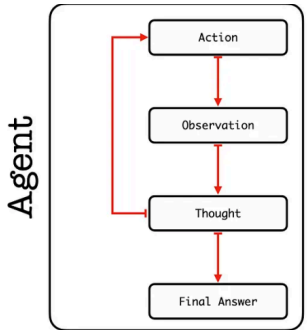
Manthandeshpande in Accredian

Leveraging LSTM and LLM Models for Stock Price Prediction

Introduction

6 min read · Feb 19, 2024

165 1



Saumitra Joshi

Langchain (Agents, Tools, Chains & Memory) for utilizing the full...

Introduction

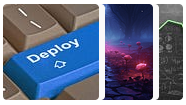
12 min read · Nov 8, 2023

75

Lists

ChatGPT prompts
44 stories · 1198 saves

ChatGPT
21 stories · 497 saves



Predictive Modeling w/ Python
20 stories · 966 saves

What is ChatGPT?
9 stories · 310 saves

vived	Pclass	Name	Sex	Age	SibSp	Parch
	3	Braund, Mr. Owen Harris	male	22	1	0
	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0
	3	Heikkinen, Miss. Laina	female	26	0	0
	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0
	3	Allen, Mr. William Henry	male	35	0	0
	3	Moran, Mr. James	male		0	0
	1	McCarthy, Mr. Timothy J	male	54	0	0
	3	Palsson, Master. Gosta Leonard	male	2	3	1

TeeTracker

Experience with the LangChain App Template

Create wine price Q&A with the LangChain App Template

8 min read · Dec 2, 2023

3 1

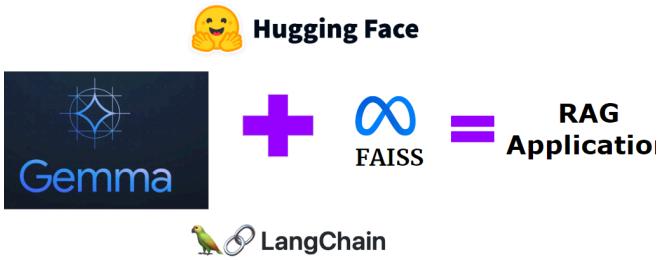
Zahrizhal Ali

Agents from LLMs? Set up and Code our own LLM-based agents...

In the subfield of Artificial Intelligence, we have Reinforcement Learning (RL). RL literall...

10 min read · Oct 29, 2023

36



Akriti Upadhyay in Superteams.ai Blog

Steps to Build RAG Application with Gemma 7B LLM

By Akriti Upadhyay

5 min read · Feb 22, 2024

178 3

Kamal Dhungana

Exploring LangChain Chains & Agents: A Quick Overview

In the LangChain framework, “Chains” represent predefined sequences of...

✦ · 15 min read · Sep 25, 2023

234

See more recommendations