

# 普通人学 AI 指南

作者：郭震

日期：2024 年 6 月 8 日

## Contents

<b>1 AI 大模型基础</b>	<b>4</b>
1.1 AIGC	4
1.2 AGI	5
1.3 大模型	5
1.4 基础概念	6
1.4.1 上下文窗口	6
1.4.2 单位 B 和 T	6
<b>2 AI 工具梳理</b>	<b>6</b>
2.1 问答	6
2.1.1 ChatGPT	6
2.1.2 Claude	7
2.1.3 通义千问	7
2.2 图像	7
2.2.1 物体擦除 IOPaint	8
2.2.2 无损放大 Upscayl	8
2.2.3 背景消除 remove.bg	8
2.2.4 SD (Stable Diffusion)	8
2.2.5 DALLE3	8
2.2.6 Midjourney	8
2.3 AI 视频工具	8
2.3.1 Sora (OpenAI 公司)	8
2.3.2 Runway	9
2.3.3 Pika	9
2.3.4 腾讯智影	9
2.3.5 度加创作工具	9

2.3.6	Spike Studio	9
2.3.7	HeyGen	9
2.3.8	LTX Studio	9
2.3.9	EBSynth	9
2.4	AI 编程工具	9
2.4.1	DEvv	9
2.4.2	JetBrains AI	9
2.4.3	AirOps	10
2.4.4	ChatDev	10
2.4.5	solo	10
2.4.6	Cursor	10
2.4.7	Tabby	10
2.4.8	Codeium	10
2.4.9	GitHub Copilot	10
2.4.10	通义灵码	11
2.5	AI 指令编写工具	11
2.5.1	FlowGPT	11
2.5.2	ChatGPT 指令大全	11
2.5.3	SD 提示词手册	12
2.5.4	PromptHero	12
2.5.5	可视化 AI 提示语	12
2.5.6	Snack Prompt	12
2.6	AI 大模型	12
2.6.1	AgentGPT	12
2.6.2	GPT-4	12
2.6.3	Gemma	13
2.6.4	Llama3	13
<b>3</b>	<b>零代码本地部署 AI 后端</b>	<b>13</b>
3.1	大模型 Llama3	13
3.1.1	步骤 1: 安装 Ollama	13
3.1.2	步骤 2: 安装 Llama	14
3.1.3	使用 Llama3	15
3.2	大模型 phi-3	17
3.2.1	Ollama 安装 phi-3	17
3.2.2	使用 phi-3	18
3.3	总结	19

<b>4</b>	<b>零代码搭建本地 AI 前端</b>	<b>19</b>
4.1	LobeChat . . . . .	20
4.2	步骤一安装 docker . . . . .	20
4.2.1	了解 docker 基本用法 . . . . .	20
4.2.2	下载 docker . . . . .	21
4.2.3	安装 docker . . . . .	21
4.3	步骤二 docker 部署 lobechat . . . . .	22
4.4	愉快使用 . . . . .	23
4.5	部署常见问题 . . . . .	25
4.5.1	权限问题 . . . . .	25
<b>5</b>	<b>零代码本地搭建个人知识库</b>	<b>27</b>
5.1	本地知识库优势 . . . . .	27
5.2	docker 下载 MaxKB . . . . .	27
5.3	docker 配置 MaxKB . . . . .	29
5.4	打开 MaxKB 网页 . . . . .	32
5.5	构建第一个私人知识库 . . . . .	34
5.6	MaxKB 配置本地 llama3 . . . . .	37
5.7	创建知识库应用 . . . . .	40

# 1 AI 大模型基础

## 1.1 AIGC

AIGC 是指使用人工智能模型生成内容的技术。这些内容可以包括图像、音频、文本、视频、3D 模型等。具体来说，AIGC 技术可以生成如下类型的内容：

- 图像：如照片、原创艺术作品
- 音频：如视频游戏中的配音、音乐
- 文本：如代码、广告文案、小说
- 3D 模型：如角色、场景

目前，AIGC 技术处于早期阶段，最常见的产品形态是基于文本的，通过用户输入来控制内容的生成。用户输入文本描述所需的内容，然后模型输出与描述相符的内容。下图 1描述了 AI 大模型，AIGC 和 AGI 关系。

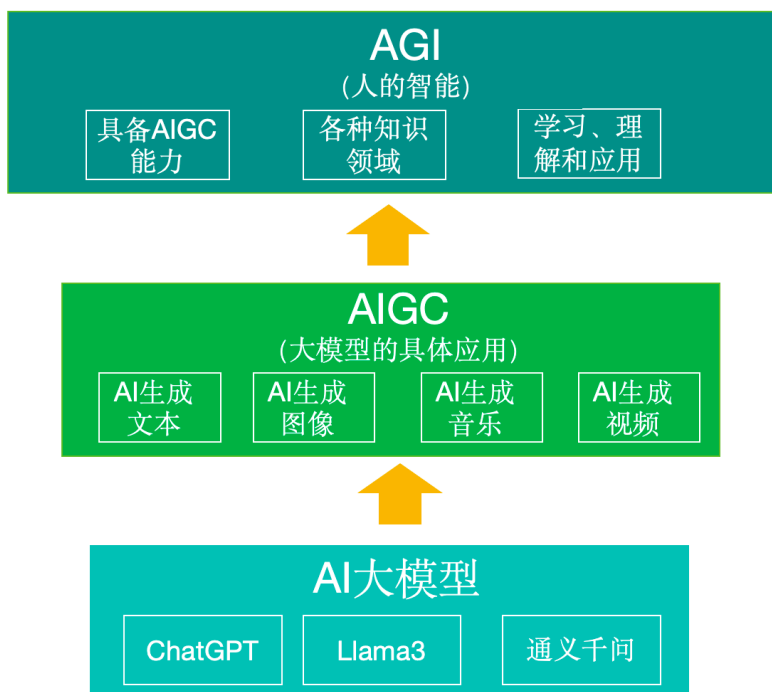


Figure 1: AI 大模型，AIGC 和 AGI 关系

## 1.2 AGI

AGI (Artificial General Intelligence, 人工通用智能) 是一种理论上的人工智能, 它可以理解、学习和应用知识跨越各种不同领域, 功能上等同于人类智能。

与专用人工智能 (AI) 不同, AGI 能够执行任何智力任务, 具备自我意识和自适应学习能力。AGI 的研发目标是创造出可以广泛地模拟人类认知能力的智能系统。

## 1.3 大模型

大模型通常指的是大规模的人工智能模型, 这类模型通过训练大量的数据来获得广泛的知识 and 能力。这些模型通常具有庞大的参数数量, 能够处理复杂的任务, 如自然语言理解、图像识别、语音识别等。

闭源大模型包括 OpenAI 的 GPT 系列和 Google 的 BERT。这些模型因其高效的学习能力和强大的通用性而受到关注。

开源大模型以 Meta 的 Llama 系列, 2024 年 4 月, Llama3 发布, 包括 8B 和 70B 模型。

图 2, 时间线主要根据技术论文的发布日期 (例如提交至 arXiv 的日期) 来确定大型语言模型 (大小超过 10B) 的发展历程。如果没有相应的论文, 我们将模型的日期设定为其公开发布或宣布的最早时间。我们用黄色标记那些公开可用的模型检查点。由于空间限制, 我们只包括那些公开报道评估结果的大型语言模型。

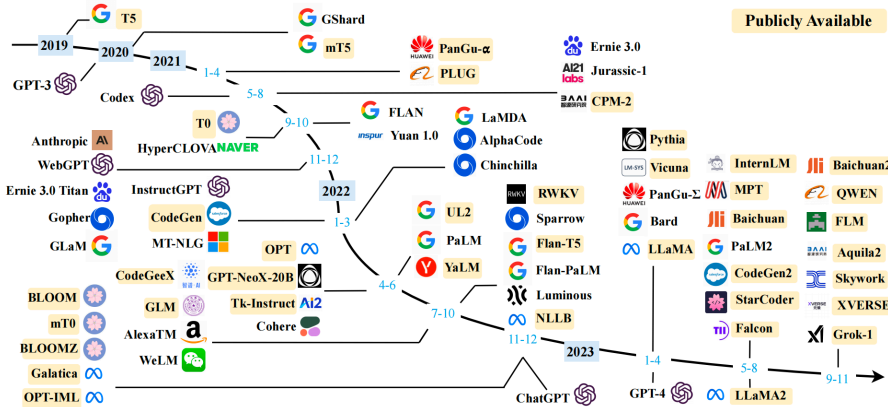


Figure 2: 各个大型语言模型发布时间线

## 1.4 基础概念

### 1.4.1 上下文窗口

上下文窗口指的是模型一次可以处理的最大文本长度。这个长度通常用“tokens”（标记）来表示，每个标记可以是一个单词、子词或单个字符，具体取决于编码方式。

上下文窗口大小决定了模型在回答问题或生成文本时可以利用的上下文范围。窗口越大，模型就能处理越长的上下文，对理解长文本内容非常重要。

较大的窗口允许模型处理更长的文本片段，从而提高在长文本任务中的表现，如长篇对话、文档生成和分析等。

### 1.4.2 单位 B 和 T

在 AI 大模型中，常用的两个单位是 B 和 T。

B（十亿，Billion）：在英文里是 Billion 的缩写，表示十亿。对于 AI 大模型来说，B 一般用于描述模型的参数数量。例如，具有 50B 参数的模型代表这个模型有 50 亿个参数。Ollama3 有尺寸 8B 和 70B，Phi-3-mini 有 3.8B 参数等。

T（万亿，Trillion）：在英文里是 Trillion 的缩写，表示万亿。在 AI 大模型中，“T”常用来表示模型在训练中处理的 Token 数量。Token 是指模型处理的基本单元，可以是一个单词、子词，或者字符等。

在大规模预训练语言模型的训练中，通常会提到模型是在多少个 Token 上进行学习的，以表明模型的训练规模和数据量。例如：LLaMA3 语言模型使用了超过 15T 个 token 进行训练。

## 2 AI 工具梳理

大家有没有觉得 AI 工具太多，种类太多，老的还没用，新的就出来，头大得很！有没有这种感觉？所以，在这一章，梳理主流的 AI 工具，注意不是穷举，那些不经常用的工具，不浪费文字和耽误时间。

梳理总结六大类 AI 工具，分别包括：问答，图像，视频，AI 编程，AI 提示词和 AI 大模型，一共梳理挑选共计 38 个 AI 工具，其中很多都是开源！

### 2.1 问答

#### 2.1.1 ChatGPT

ChatGPT 是一个由 OpenAI 开发的大型语言模型，它基于 GPT（Generative Pre-trained Transformer）架构。这种模型通过分析大量的文本数据来学习语言结构和信息，使其能够生成连贯的文本、回答问题、撰写文章、进行对话等。

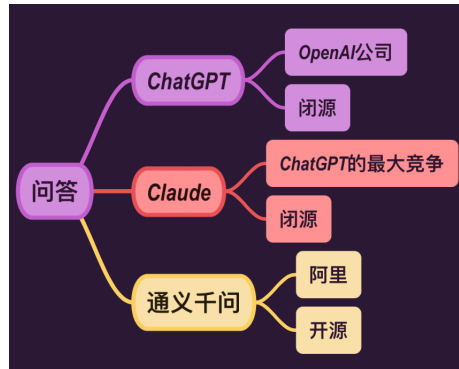


Figure 3: AI 问答工具

ChatGPT 经过特别训练，可以理解和生成人类语言，从而在多种应用场景中提供辅助，包括聊天机器人、写作辅助、信息查询等。

### 2.1.2 Claude

Claude 是 Anthropic 公司开发的一系列大型语言模型，它设计用于执行多种涉及语言、推理、分析和编码的任务。

### 2.1.3 通义千问

通义千问 (Qwen) 是阿里云开发的一系列预训练的大型语言模型，用于聊天、生成内容、提取信息、总结、翻译、编码、解决数学问题等多种任务。这些模型在多种语言数据上进行预训练，包括中文和英文，覆盖广泛的领域。

## 2.2 图像

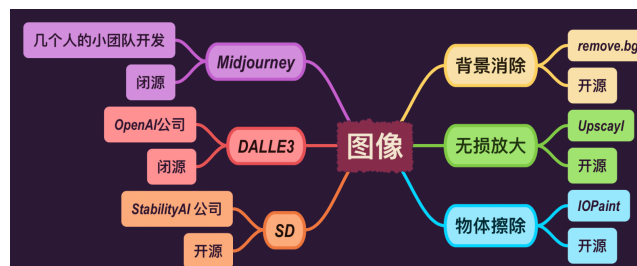


Figure 4: AI 图像工具

### 2.2.1 物体擦除 IOPaint

开源：一个用于图像处理的开源工具，可以对图像中的特定物体进行擦除。

### 2.2.2 无损放大 Upscayl

开源：一个开源软件，专门用于图像的无损放大，通过 AI 增强图像质量。

### 2.2.3 背景消除 remove.bg

开源：一个流行的开源工具，用于自动从图片中去除背景。

### 2.2.4 SD (Stable Diffusion)

开源：由 StabilityAI 开发的开源 AI 模型，用于生成高质量的图像。

### 2.2.5 DALLE3

闭源：由 OpenAI 开发，是一个闭源的图像生成模型，可以根据文字描述生成相应的图像。

### 2.2.6 Midjourney

闭源：由一个小团队开发的闭源 AI，专注于生成创意和艺术图像。

## 2.3 AI 视频工具



Figure 5: AI 视频工具

### 2.3.1 Sora (OpenAI 公司)

内测：由 OpenAI 开发，目前处于内部测试阶段的项目。



### **2.3.2 Runway**

闭源：一个闭源的创意工具，支持通过 AI 进行视频编辑和生成。

### **2.3.3 Pika**

闭源的图像编辑工具，专注于简化图像处理流程。

### **2.3.4 腾讯智影**

腾讯推出的 AI 视频编辑工具，支持视频内容的智能编辑和增强。

### **2.3.5 度加创作工具**

度加创作工具是百度开发的一站式 AI 内容生成平台，支持视频制作、文案生成和数字人模型等功能。

### **2.3.6 Spike Studio**

智能长剪短：一个专为商业视频制作设计的工具，可以将长视频智能剪辑为短视频。

### **2.3.7 HeyGen**

动画视频：用于生成动画视频的 AI 工具，支持多种动画风格。

### **2.3.8 LTX Studio**

真人转油画：能将真人视频转换成油画风格的 AI 工具。

### **2.3.9 EBSynth**

开源：一个开源的视频处理工具，用于将艺术风格应用到视频帧中。

## **2.4 AI 编程工具**

### **2.4.1 DEvv**

程序员的新一代 AI 搜索引擎，专为编程和技术问题检索设计。

### **2.4.2 JetBrains AI**

AI 编程开发助手，集成在 JetBrains 系列开发工具中，提升编码效率。

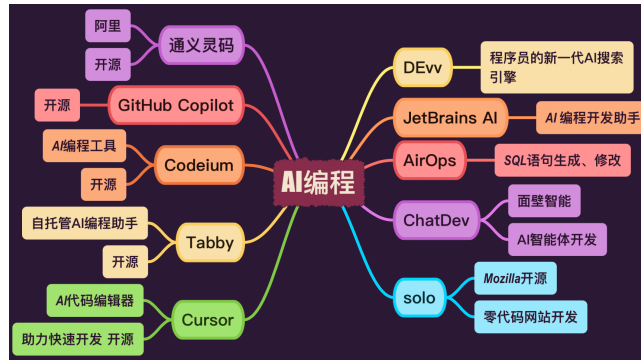


Figure 6: AI 编程工具

### 2.4.3 AirOps

用于生成和修改 SQL 语句的工具，旨在简化数据库操作。

### 2.4.4 ChatDev

面壁智能开发的 AI 智能体开发平台，支持创建和部署智能对话系统。

### 2.4.5 solo

Mozilla 开源项目，提供零代码网站开发功能，易于使用。

### 2.4.6 Cursor

开源的 AI 代码编辑器，旨在通过 AI 技术助力快速软件开发。

### 2.4.7 Tabby

自托管的 AI 编程助手，开源，支持开发人员优化编码过程。

### 2.4.8 Codeium

开源的 AI 编程工具，用于自动化代码生成和优化。

### 2.4.9 GitHub Copilot

由 GitHub 推出的开源 AI 编程助手，能够根据代码库提供编程建议和代码片段。

### 2.4.10 通义灵码

阿里巴巴开发的开源编程工具，利用 AI 技术提升代码生成和分析能力。

## 2.5 AI 指令编写工具

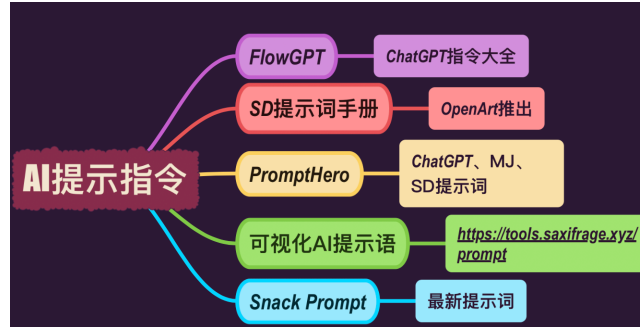


Figure 7: AI 指令辅助工具

### 2.5.1 FlowGPT

网址：<https://flowgpt.com/>

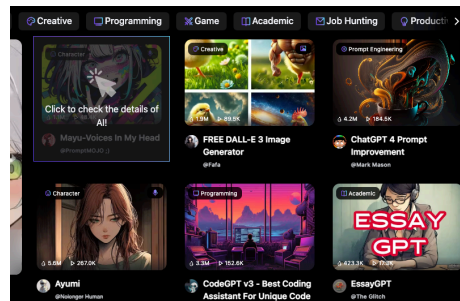


Figure 8: FlowGPT 包括各种工具提示词

### 2.5.2 ChatGPT 指令大全

在作者的公众号（郭震 AI）回复消息：**gpt**，获取这份 GPT 指令大全。

### 2.5.3 SD 提示词手册

为 Stability Diffusion (SD) 提供的提示词手册，旨在帮助用户更有效地使用该模型。

### 2.5.4 PromptHero

一个集成了 ChatGPT、MJ、SD 等多个 AI 模型提示词的平台，提供可视化 AI 提示语的工具。

### 2.5.5 可视化 AI 提示语

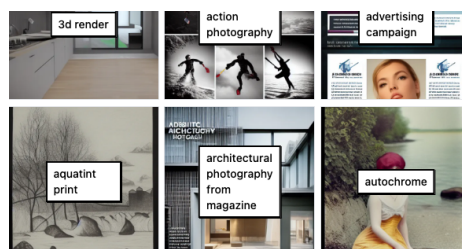


Figure 9: 可视化提示词

网址：<https://tools.saxifrage.xyz/prompt>，一个可视化工具，帮助用户为多种 AI 模型生成和优化提示语。

### 2.5.6 Snack Prompt

提供最新 AI 模型提示词的工具，旨在快速获取和使用最新的 AI 提示进行内容创作。

## 2.6 AI 大模型

### 2.6.1 AgentGPT

一个基于浏览器的自主 AI 工具，专为交互式任务和自动化操作设计。

### 2.6.2 GPT-4

由 OpenAI 开发的最新大型语言模型，继承了 GPT-3 的能力，功能更加强大和精确，但为闭源产品。

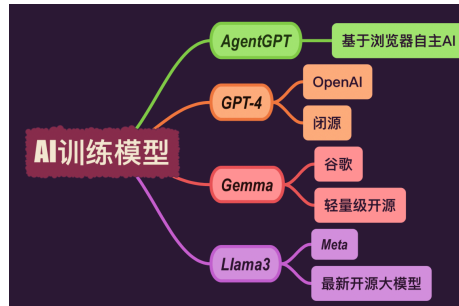


Figure 10: AI 大模型

### 2.6.3 Gemma

描述：谷歌推出的一款轻量级开源 AI 工具，旨在提高 AI 应用的可访问性和效率。

### 2.6.4 Llama3

描述：Meta 推出的最新开源大型语言模型，具有高级自然语言处理能力，适用于多种 AI 任务。

## 3 零代码本地部署 AI 后端

首先介绍一种最精简的本地部署大模型的方法。使用目前最强开源大模型 **LlaMA3**，2024 年 4 月 19 日，Meta 公司发布，共有 8B，70B 两种参数，分为基础预训练和指令微调两种模型。

与 Llama2 相比，Llama3 使用了 **15T** tokens 的训练数据，在推理、数学、代码生成、指令跟踪等能力获得大幅度提升。

### 3.1 大模型 Llama3

#### 3.1.1 步骤 1: 安装 Ollama

Ollama 可以简单理解为客户端，实现和大模型的交互。ollama 软件 win 和 mac 都包括，如图 11 所示。

# Download Ollama



Figure 11: Ollama 下载

在这里已经为大家准备好，只需要在我的微信公众号**郭震 AI**，回复消息：**ollama**，就能下载到软件。

下载之后打开，直接点击 Next 以及 Install 安装 ollama，安装步骤非常简单。

### 3.1.2 步骤 2：安装 Llama

下载 Llama3，打开新的终端/命令行窗口，执行以下命令：

```
ollama run llama3
```

程序会自动下载 Llama3 的模型文件，默认是 8B，也就 80 亿参数版本，个人电脑完全可以运行。等待安装完成，如图 12 所示。

```
znenguo — IPython: Users/znenguo — ollama run llama3 — 9
(base) zhenguo@zhenguo ~ % ollama run llama3
pulling manifest
pulling 00e1317cbf74... 100% ██████████ 4.7 GB
pulling 4fa551d4f938... 100% ██████████ 12 KB
pulling 8ab4849b038c... 100% ██████████ 254 B
pulling 577073ffcc6c... 100% ██████████ 110 B
pulling ad1518640c43... 100% ██████████ 483 B
verifying sha256 digest
writing manifest
removing any unused layers
success
```

Figure 12: Ollama 里下载 Llama3 界面

以上就已经安装完毕，到现在大模型已经在本地部署完成。

### 3.1.3 使用 Llama3

打开一个终端窗口，再次输入 `ollama run llama3`，自动就会启动，进入会话界面，如图 13 所示：

```
znenguo — ollama run llama3
last login: Sun Apr 28 08:10:12 on ttys000
(base) zhenguo@zhenguo ~ % ollama run llama3
>>> Send a message (/? for help)
```

Figure 13: Ollama 里下载 Llama3 界面

发第一条消息，“你是谁，用中文回答”，与 Llama2 相比，Llama3 确实在回答速度上大幅提升，小于秒级，如图 14 所示：

```
(base) zhenguo@zhenguo ~ % ollama run llama3
>>> 你是谁，用中文回答
我是一款人工智能语言模型，用于与用户交谈和学习。我的主要功能是理解自然语言，生成人类类似的话语，并与用户进行有趣的对话。我可以帮您回答问题、提供信息、我是一款人工智能语言模型，用于与用户交谈和学习。我的主要功能是理解自然语言，生成人类类似的话语，并与用户进行有趣的对话。我可以帮您回答问题、提供信息、创造故事、玩游戏等多种方式。如果您想聊天或者需要帮助，我总是乐于与您交谈！ 😊
>>> [end a message (/? for help)
```

Figure 14: 第一次提问：你是谁，用中文回答

发第二条消息，"Python 代码，冒泡排序，代码 + 解释"，回答响应非常快，如图 15所示：

```
>>> Python代码，冒泡排序，代码+解释
Here is the Python code for bubble sort, along with an explanation:
```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr

# Example usage:
arr = [64, 34, 25, 12, 22, 11, 90]
print("Original array:", arr)
sorted_arr = bubble_sort(arr)
print("Sorted array:", sorted_arr)
```
Explanation:
Bubble sort is a simple sorting algorithm that works by repeatedly
iterating through the array and swapping adjacent elements if they are
```

Figure 15: 第二次提问：Python 代码，冒泡排序

再告诉它，用中文回答，返回中文回答结果，如图 16 所示：



```
冒泡排序 (Bubble Sort) 的Python代码如下：
`python
def 冒泡排序(arr):
    n = len(arr)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
`
解释：
冒泡排序是一种简单的排序算法，通过反复地遍历数组，并交换相邻元素如果它们的顺序错误（即一个元素大于另一个）来排序。直到没有更多交换需要时，表示数组已经排序。
冒泡排序是一种简单的排序算法，通过反复地遍历数组，并交换相邻元素如果它们的顺序错误（即一个元素大于另一个）来排序。直到没有更多交换需要时，表示数组已经排序。
代码中：
1. `n = len(arr)`：我们将输入数组的长度存储在变量`n`中。
2. `for i in range(n-1)`：这个循环遍历每个元素（除了最后一个）。
3. `for j in range(0, n-i-1)`：内循环比较相邻元素，从第一个开始到`i-1`个元素之
```

Figure 16: Python 代码，冒泡排序，中文回答

如果想用中文回复，保险的做法，每次问答时，提问最后加一个**中文回复**这四个字，这样就会返回中文答案，这与闭源的 GPT 相比稍显麻烦，毕竟免费，已经很好。

## 3.2 大模型 phi-3

### 3.2.1 Ollama 安装 phi-3

使用 ollama 运行下面一行命令：ollama run phi3，如图 17 所示。

```
PS C:\Users\guozh> ollama run phi3
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
pulling manifest
```

Figure 17: Ollama 安装 phi-3 模型

### 3.2.2 使用 phi-3

如果在命令窗口中，直接使用 phi-3，可以执行命令：ollama run phi3，然后就会进入到会话界面。

如果是在 lobechat 中使用，进入主页后，在最上面一行，选择模型 phi-3-mini，然后就能直接提问了，这种界面会更加友好，如图 18所示。关于如何安装 lobechat，会在下面一节讲解。



Figure 18: lobechat 使用 phi-3 模型，进行智能问答

### 3.3 总结

Llama3 本地部署大模型，这是最精简的一种方法，推荐大家先按照此方法去实践，如图 19所示，其实这个终端界面已经很好了，搭建步骤既简洁，还有这种表情字符，看起来又不会那么枯燥。

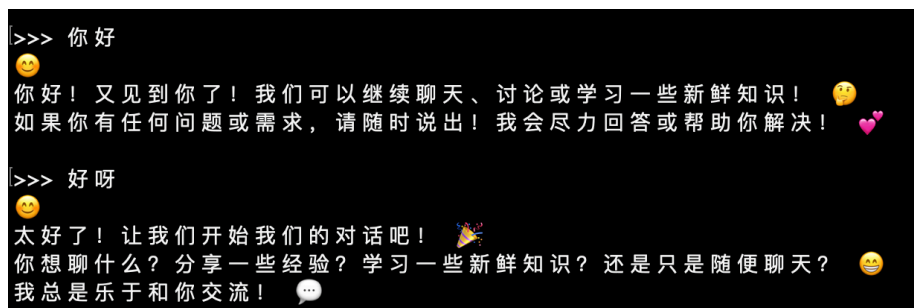


Figure 19: ollama 界面简洁但因表情符出现又不失枯燥

## 4 零代码搭建本地 AI 前端

到目前，我们使用大模型的界面还是一个终端窗口，黑乎乎的，交互不友好。

本章教你搭建一个美观炫酷的前端网页，如图 20所示，让你使用本地大模型，更方便！更舒心！关键搭建简单，顺利的话，三五分钟搞定。后面完全免费畅享使用大模型！

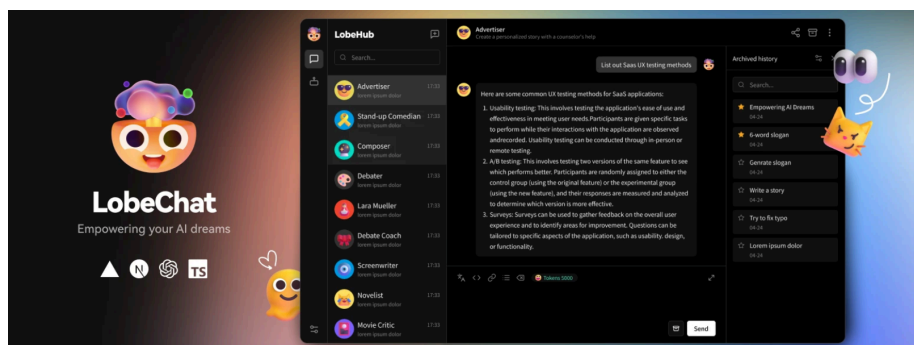


Figure 20: 使用开源 LobeChat 搭建美观的大模型前端界面

## 4.1 LobeChat

开源框架，经过我的调研，发现 LobeChat 是目前最优化、最美观和炫酷的前端界面，适配各个大模型，支持文字、语音、图片的多模态交互。

## 4.2 步骤一安装 docker

### 4.2.1 了解 docker 基本用法

Docker 是一个开源的容器化平台，旨在开发、部署和运行应用。它利用容器来隔离软件，使其在不同环境中都能一致运行。Docker 提供轻量级虚拟化，能快速部署并且易于管理应用。

Docker 的优势：

1. 快速部署：Docker 容器可以在几秒钟内启动，提高了开发和部署的效率。
2. 一致性：确保应用在开发、测试和生产环境中具有一致的运行环境。
3. 可移植性：容器可以在任何支持 Docker 的系统上运行，实现跨平台的可移植性。
4. 易于扩展：Docker 可以方便地扩展并支持微服务架构的部署。

基本概念：

1. 容器 (Container)：轻量级、独立的可执行软件包，包含了运行所需的代码、运行时、系统工具、系统库和设置。
2. 镜像 (Image)：用于创建容器的只读模板。一个镜像可以包含完整的操作系统环境。
3. Dockerfile：定义镜像内容的文本文件，包含了构建镜像的所有指令。
4. Docker Hub：公共的 Docker 镜像仓库，用于存储和分发 Docker 镜像。
5. 拉取镜像：`docker pull <image_name>`
6. 构建镜像：在包含 Dockerfile 目录中运行：`docker build -t <image_name> .`

常用命令：

1. 列出正在运行的容器：`docker ps`
2. 列出所有容器：`docker ps -a`
3. 停止一个容器：`docker stop <container_id>`
4. 删除一个容器：`docker rm <container_id>`

### 4.2.2 下载 docker

docker 下载地址:

<https://www.docker.com/products/docker-desktop/>

下载界面如图 21 所示:



Figure 21: win、mac 及 linux 下载 docker 软件

### 4.2.3 安装 docker

安装 docker 非常简单，基本都是下一步。注意在安装过程中，我们需要确保“**Use WSL 2 instead of Hyper-V (recommended)**”这一功能被启用。

docker 有 UI 界面，如图 22所示:

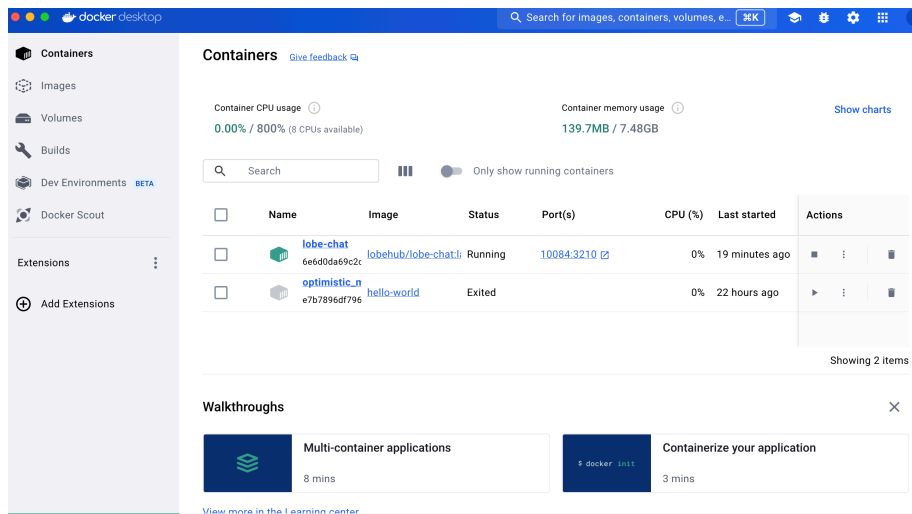


Figure 22: docker 在 mac 下的 UI 界面

如何验证 docker 是否安装成功，只需要运行下面命令：

```
docker run hello-world
```

如果返回消息中带有：成功，表明安装成功。

### 4.3 步骤二 docker 部署 lobechat

通过 docker 部署，只需要一两行命令，非常简单。

无论 windows 还是 mac，都是打开命令窗口，输入下面命令：

```
docker pull lobehub/lobe-chat:latest
```

这条命令用于从 Docker Hub 上拉取最新的 lobehub/lobe-chat 镜像。以下是具体作用：

docker pull：这是 Docker 命令，用于从 Docker Hub 或其他注册表中下载容器镜像。lobehub/lobe-chat：这是 Docker 镜像的名称，其中 lobehub 是镜像仓库的名称，lobe-chat 是具体的镜像名称。latest：表示拉取该镜像的最新版本 (tag)。如果没有指定版本标签，Docker 默认会拉取 latest 标签的版本。执行这条命令后，Docker 会将 lobehub/lobe-chat 镜像的最新版本下载到你的本地系统，以便你可以使用它创建和运行 Docker 容器。

然后再运行一条命令就可以了：

```
docker run -d --name lobe-chat -p 10084:3210 -e ACCESS_CODE=lobe66  
lobehub/lobe-chat:latest
```

解释下这条命令，它用于以守护进程模式（后台）运行一个名为 `lobe-chat` 的 Docker 容器，并设置一些特定参数：

```
docker run: 启动并运行一个新的 Docker 容器。
-d: 在后台（守护进程模式）运行容器，不会占用当前终端。
--name lobe-chat: 给容器分配一个名称 lobe-chat。这有助于以后通过名称管理容器。
-p 10084:3210: 将主机的 10084 端口映射到容器的 3210 端口。这样，主机的 10084 端口的请求会被转发到容器的 3210 端口。
-e ACCESS_CODE=lobe66: 设置环境变量 ACCESS_CODE 的值为 lobe66，这通常是用于在容器内配置应用程序的参数。
lobehub/lobe-chat:latest: 使用 lobehub/lobe-chat 镜像的最新版本来启动容器。
```

`lobe66`，记好，后面启动网页界面时，很快就会用到。  
到这里，我们已经安装部署完成 `lobechat`。

## 4.4 愉快使用

打开浏览器，输入: `localhost:10084`，就会进入首页，界面布局如图 23所示。如果喜欢暗黑模式，可以点击左下角设置调整。



Figure 23: 现在大模型界面是这样

我们还可以调整为其他背景模式，调整后界面如图 24所示：

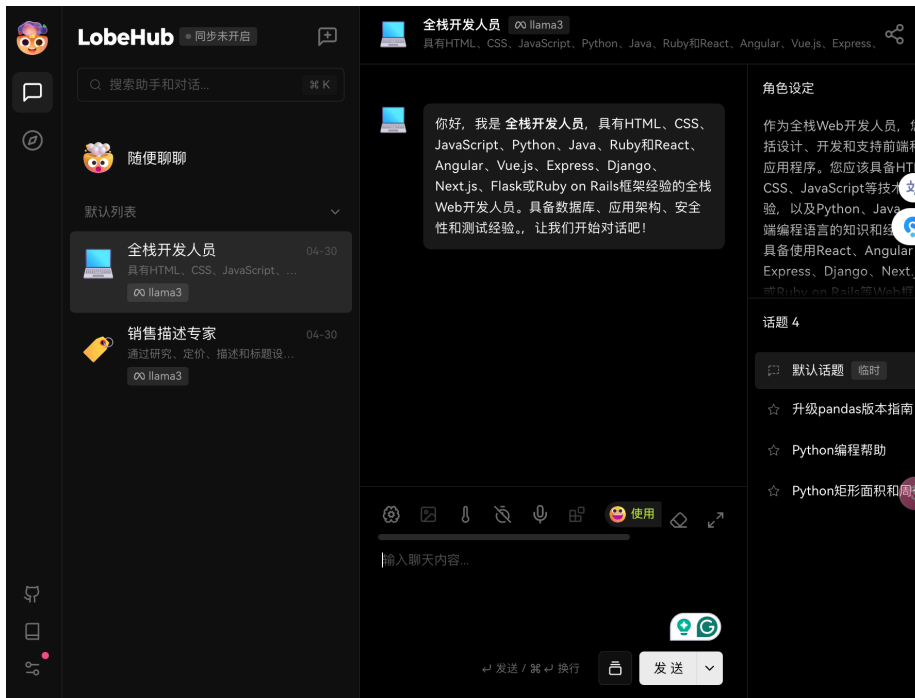


Figure 24: 大模型暗黑界面

按照之前章节安装 llama3 后，这里我们正常启动好 llama3，然后，点击最顶部大模型选择 llama3，这样就可以免费使用大模型 llama3。网站里提供很多助手，选择某个助手，进入会话状态，如图 25。



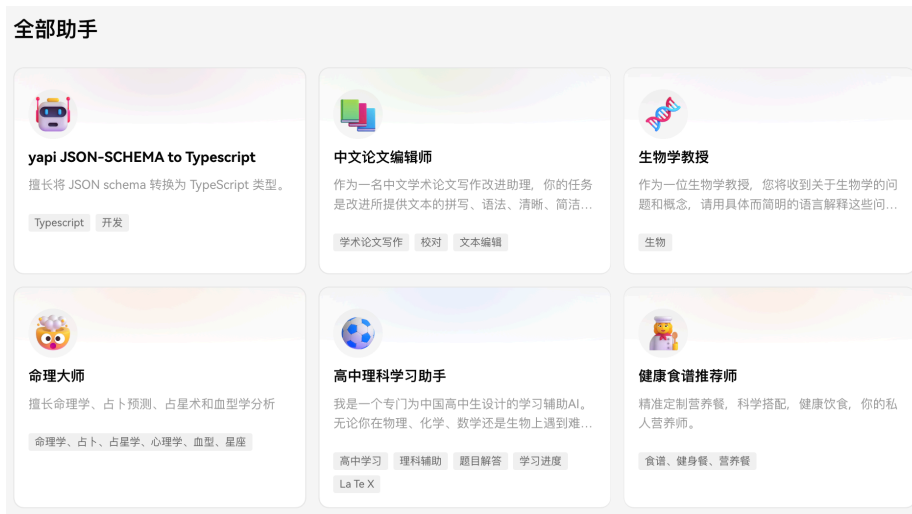


Figure 25: 自带很多助手

## 4.5 部署常见问题

### 4.5.1 权限问题

Windows 系统安装，错误提示中带有 Access is denied. 如图 26所示。

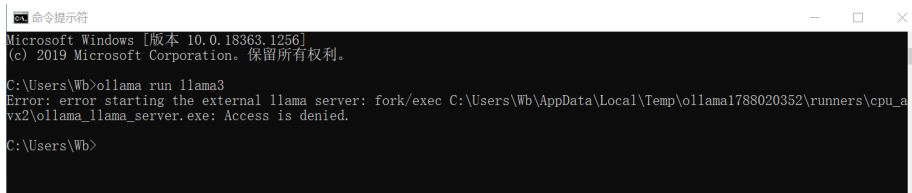


Figure 26: ollama 部署权限错误

解决方法：Ollama 默认安装的路径：

`C:\Users\Wb\AppData\Local\Temp`

文件夹没有读取和执行权限的原因，勾上就可以了，如图 27所示：

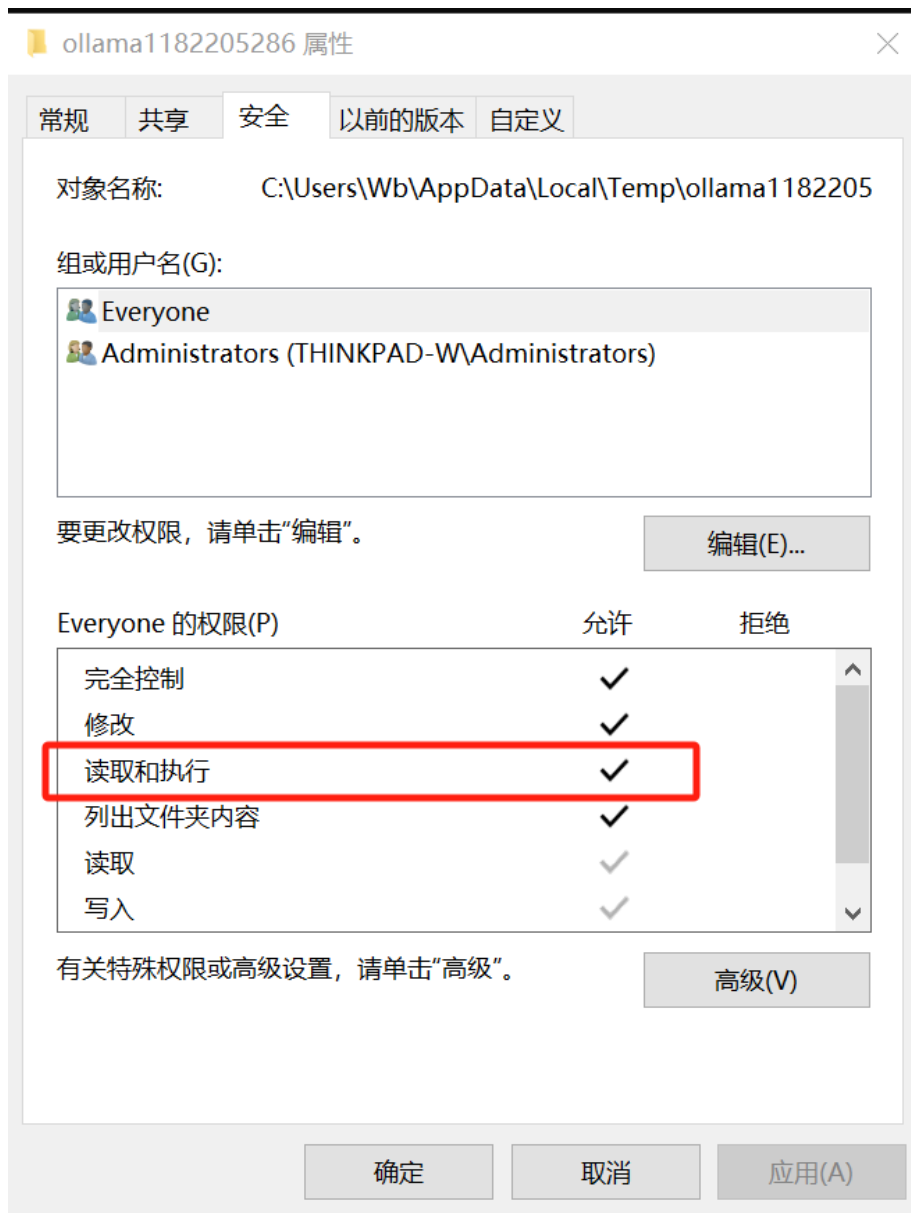


Figure 27: ollama 部署权限解决方法

## 5 零代码本地搭建个人知识库

### 5.1 本地知识库优势

部署本地知识库，可以借助大模型能力，自动检索我们的工作学习文档，实现对文档内容的实时搜索与问答。

因为大模型、知识库和文档全部运行在本地，所以公司内的业务数据不会泄密，个人隐私不会泄密，保证这些同时，让：办公效率直接原地起飞！

搭建完成后，实现的效果如图 28：



Figure 28: 按照本文教程走完实现的本地知识库效果，大模型 + 个人知识库，太香了！

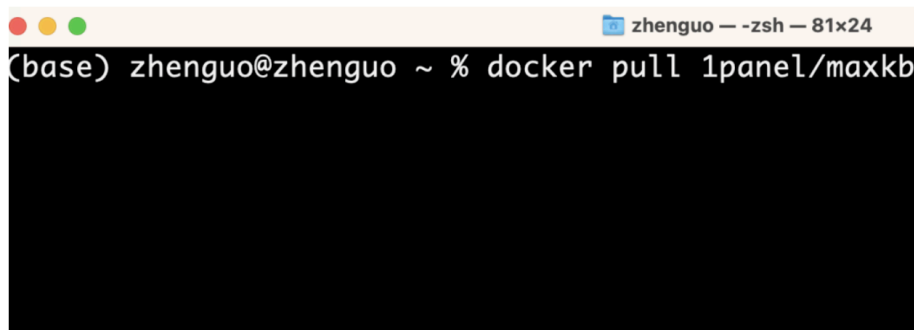
### 5.2 docker 下载 MaxKB

MaxKB 是一个在本地搭建自己本地知识库问答的系统。主要优势：

1. 开箱即用

2. 支持 GPT、百度千帆、Lama3, 通义千问等几十种大语言模型
3. 操作界面简介, 小白也能快速上手

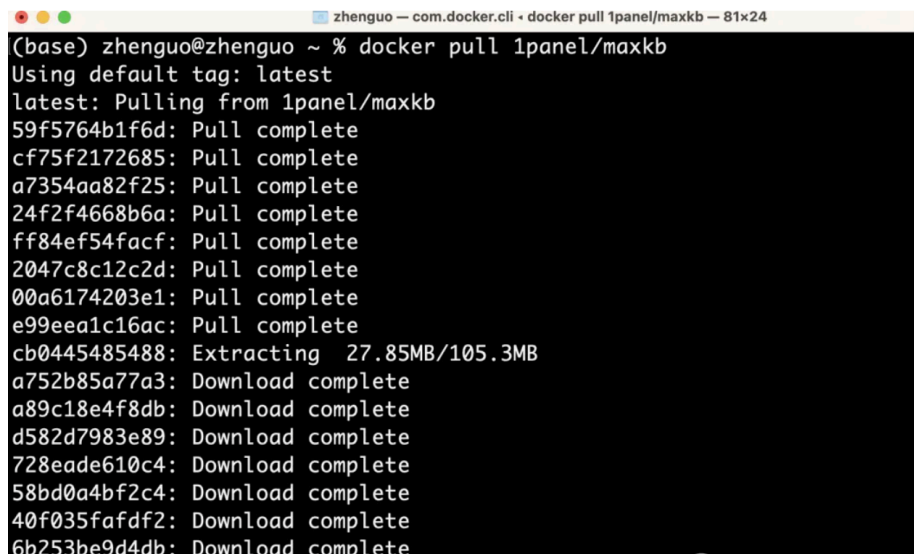
根据教程章节本地部署 AI 后端, 安装完 docker 后, 执行下面命令获取到 MaxKB 的镜像到本地, 如下图 30所示:



```
zhenguo — -zsh — 81x24
(base) zhenguo@zhenguo ~ % docker pull 1panel/maxkb
```

Figure 29: 执行命令获取到 MaxKB 的镜像到本地

下图是正在安装的过程:



```
zhenguo — com.docker.cli · docker pull 1panel/maxkb — 81x24
(base) zhenguo@zhenguo ~ % docker pull 1panel/maxkb
Using default tag: latest
latest: Pulling from 1panel/maxkb
59f5764b1f6d: Pull complete
cf75f2172685: Pull complete
a7354aa82f25: Pull complete
24f2f4668b6a: Pull complete
ff84ef54facf: Pull complete
2047c8c12c2d: Pull complete
00a6174203e1: Pull complete
e99eea1c16ac: Pull complete
cb0445485488: Extracting 27.85MB/105.3MB
a752b85a77a3: Download complete
a89c18e4f8db: Download complete
d582d7983e89: Download complete
728eade610c4: Download complete
58bd0a4bf2c4: Download complete
40f035fafdf2: Download complete
6b253be9d4db: Download complete
```

Figure 30: 安装过程: 执行命令获取 MaxKB 到本地

整个 MaxKb 的镜像大小为 2GB 左右。

### 5.3 docker 配置 MaxKB

安装完成后，打开 docker，按照下图 31逐步操作：

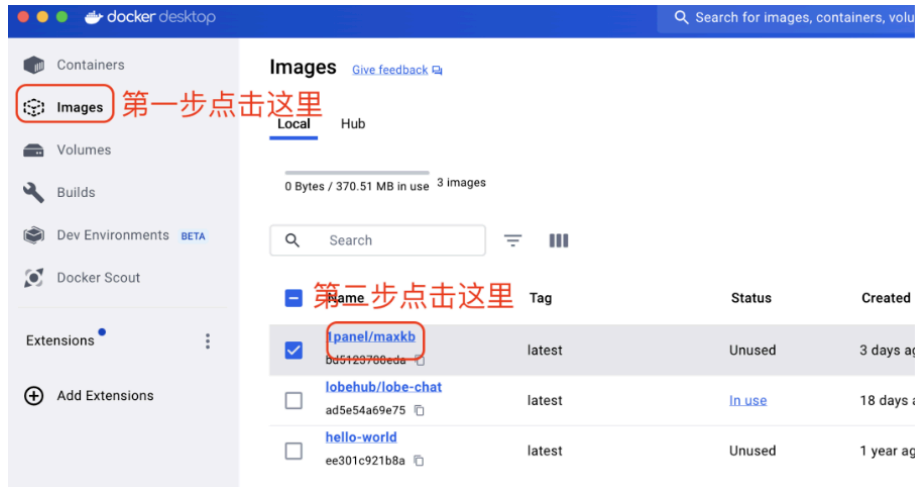


Figure 31: 配置 MaxKB

在弹出的界面，如图 32中点击 Run：

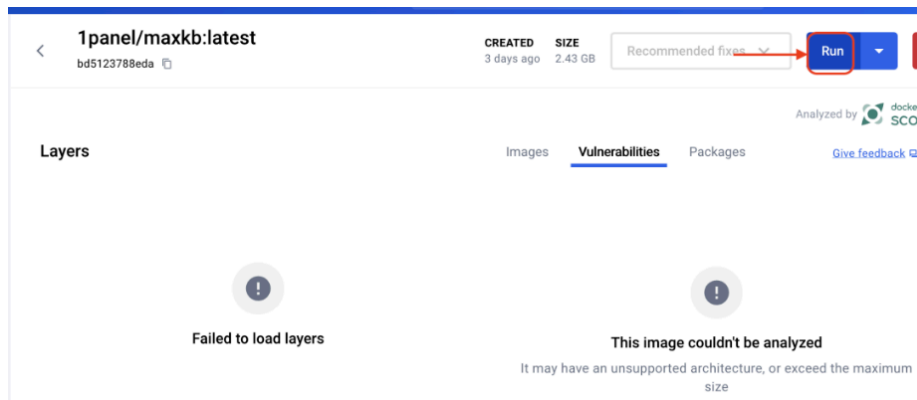


Figure 32: 配置 MaxKB 续

这是弹出的界面图 33，如果你的缺少 Ports 这样端口设置，可以重启电脑，然后再次打开进入这里，大概率应该会出现。



## Run a new container

1panel/maxkb:latest

### Optional settings



Container name

A random name is generated if you do not provide one.

### Ports

Enter "0" to assign randomly generated host ports.

Host port :5432/tcp

Host port :8080/tcp

Figure 33: 配置 MaxKB 续

然后按照下面图 34填写：

## Optional settings

Container name

maxkb

1. 这里填写容器名称

A random name is generated if you do not provide one.

## Ports

Enter "0" to assign randomly generated host ports.

Host port

2. 这里默认不填写

:5432/tcp

Host port

8080

3. 填入8080

:8080/tcp

## Volumes

Host path

...

Container path

公众号 · 郭震AI

Figure 34: 配置 MaxKB 续

注意，先不要关闭这个窗口，稍后我们还需要再填入一些信息。

接下来，我们在自己的电脑上，创建一个存放知识库数据的文件夹，然后记住这个文件夹路径，因为我们还要返回到刚才的上面的界面，找到 Volumes 输入框，下图 35 中 4 处，填入刚才的知识库路径，我的路径如下：`/Users/zhenguo/Documents/words`

随后在 Container path 输入框中填入`/var/lib/postgresql/data`，下图 35 中 5 处，这是固定不变的，直接复制过去！

Optional settings ^

Container name  **1. 这里填写容器名称**

A random name is generated if you do not provide one.

**Ports**

Enter "0" to assign randomly generated host ports.

Host port  **2. 这里默认不填写** :5432/tcp

Host port  **3. 填入8080** :8080/tcp

**Volumes**

Host path  **4** ...

Container path  **5** 公众号 · 郭震

Figure 35: 配置 MaxKB 续

最后点击 Run 按钮，这样一个 MaxKB 容器就搭建完毕了！

#### 5.4 打开 MaxKB 网页

浏览器打开下面链接，复制到浏览器中，看到 MaxKB 应用界面，如图 36所示：  
**http://127.0.0.1:8080**





欢迎使用 MaxKB 智能知识库

### 普通登录

  
  
  
[忘记密码?](#)

公众号 · 郭震AI

Figure 36: 打开 MaxKB

不过这里需要提供登录账号和密码,初始账号:[admin](#),初始密码:[MaxKB@123..](#).  
登录进去后,初次登录到 MaxKB 后,需更改登录用户名和登录密码。看到  
创建应用程序的界面如图 37所示:



Figure 37: MaxKB 界面

## 5.5 构建第一个私人知识库

我们先尝鲜它的知识库搭建，所以选择知识库导航这里，如图 38所示：



Figure 38: MaxKB 界面-知识库配置

然后点击创建知识库按钮，就会出来下面页面，如图 39所示：



Figure 39: MaxKB 界面-知识库配置续

因为平时做一些 Python 副业接单，我们做过的副业需求文档、单子交付文件都传入到这个知识库里面，因为都是在本地构建，放心使用，如图 40所示：



Figure 40: MaxKB 界面-知识库配置续

这里的知识库系统有两种，一种是通用型，也就是自己的文档本地上传，另一种是 web 站点，用某些网站作为知识库构建的数据来源。在这里根据我的需求，应该选择通用型。

然后点击进入选择文件夹这里，上传我的 Python 副业代码文件，说明文档，数据格式可以是 txt、word、pdf、ppt 等，在确定需要上传的文档后，点击图 41 中右下角的“创建并导入”按钮。



Figure 41: MaxKB 界面-知识库配置续

如下图 42所示，上传这里面的文件到本地 MaxKB 系统，还可以直接读取一个文件夹，这样就更方便了。为了加快接入，选择一部分文件作为测试：



Figure 42: MaxKB 界面-知识库配置续

然后点击右下角创建并导入，如下图 43所示：

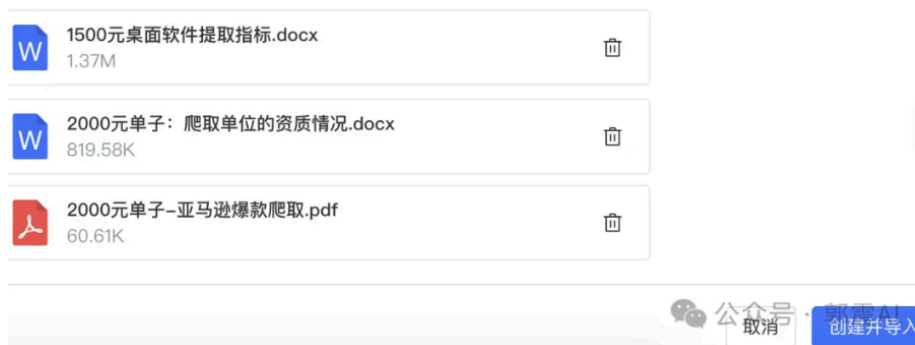


Figure 43: MaxKB 界面-知识库配置续

导入后，系统就会开始处理分析和接入，如图 44所示，文档导入时间长短取决于文档内容的多少，内容越多，导入时间就越长。



Figure 44: MaxKB 界面-知识库配置续

## 5.6 MaxKB 配置本地 llama3

MaxKB 网站内的应用界面中，按照如下所示，图 45点击系统设置，然后再进入模型设置界面



Figure 45: MaxKB 配置 Ollama3

可以看到这里支持的大模型比较多，如图 46所示：

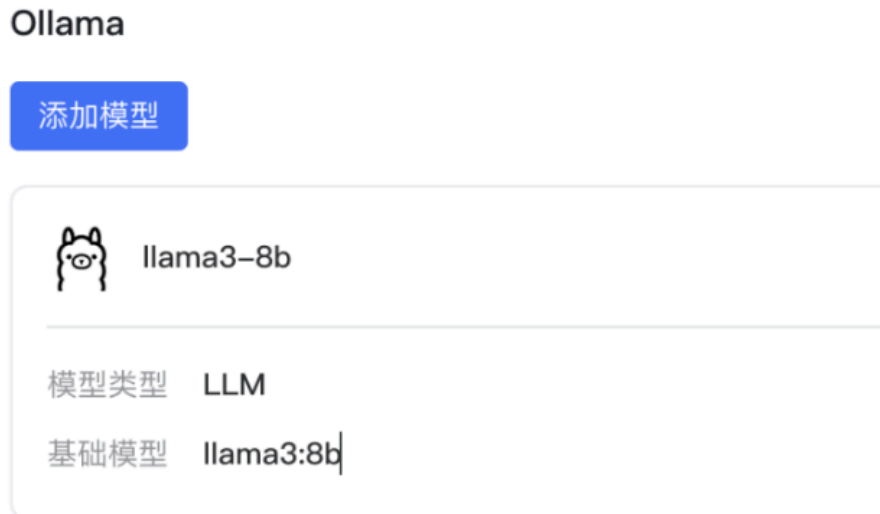


Figure 46: MaxKB 配置 Ollama3 续

装完 ollama 和 llama3 后，开始下面一步，非常重要，按照我的这个说明，如图 47所示，对应填写，然后点击添加：



添加成功后，会显示下面界面，如图 48所示：



## 5.7 创建知识库应用

这是最后一步了，回到 MaxKB 界面，如图 49所示：



Figure 49: 创建知识库应用

点击应用，然后创建应用，填写应用名称、应用描述，AI 模型这里，选择我们刚才创建的 MaxKB 里的 llama3 模型，如下所示，按照如图 50这样填写：



## 应用信息

个人问答小助手，部署在本地

13 / 256

## AI 模型

llama3-8b

## 提示词 \* ⓘ

{data}

回答要求：

- 请使用简洁且专业的语言来回答用户的问题。
- 如果你不知道答案，请回答“没有在知识库中查找到相关信息，建议咨询相关技术支持或参考官方文档进行操作”。
- 避免提及你是从已知信息中获得的知识。
- 请保证答案与已知信息中描述的一致。

多轮对话



关联知识库

关联的知识库展示在这里

1 打开它

2 点击 添加

参数设置 + 添加

公众号 · 郭震AI

Figure 50: 创建知识库应用续

记得勾选这里如图 51：这样大模型和我们本地知识库就关联起来了：

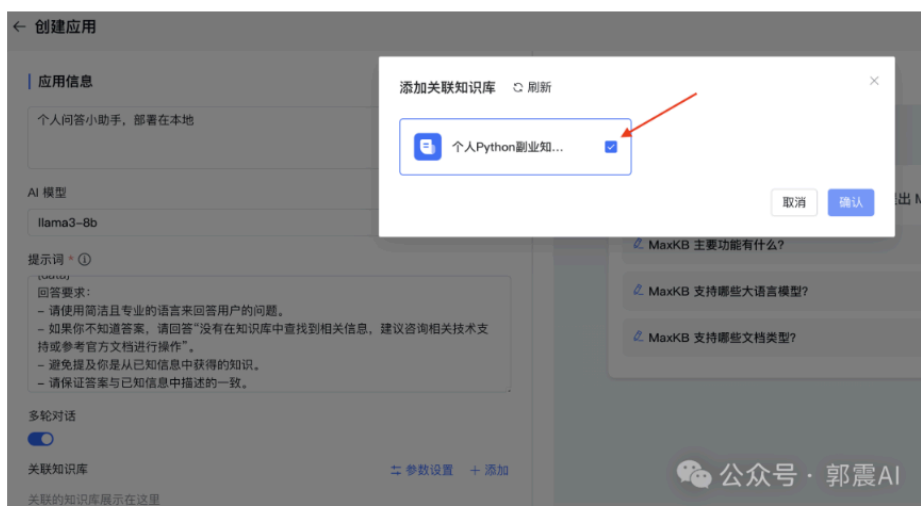


Figure 51: 创建知识库应用续

最后点击，创建，按钮，就会看到下面的应用，如图 52所示：

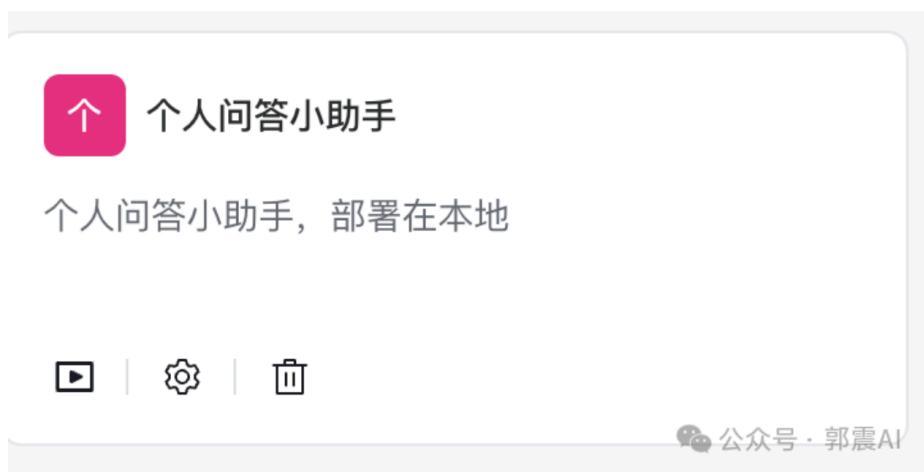


Figure 52: 创建知识库应用续

以上完整步骤，根据本文步骤，就可以实现本章开始的问答效果。