

# 对话助手

## 简介

使用ollama实现本地模型的定制, 可以做到数据不泄露以及绕开检测的效果, 之后使用嘉立创的esp32开发板实现简单的对话助手

## Windows环境搭建

这里我看的教程是[DeepSeek R1, 本地部署才是王道! 哔哩哔哩bilibili](#)

下载[ollama](#), 使用默认安装即可

## 文件位置

### 按个人需求更改

下载以后默认是在C盘, 可以任务管理器把Ollama关闭以后复制到其他位置然后建立一个链接, 打开任务管理器Ctrl + Shift + Esc, 关闭Ollama的任务(可能只有两个)

把这个 `C:\Users\jiao\AppData\Local\Programs\ollama` 剪切到其他路径, 之后使用管理员权限打开cmd

输入 `mklink /d "C:\Users\jiao\AppData\Local\Programs\ollama"` 你剪切到的位置

为了可以方便的使用 `ollama` 命令可以把它你复制到的文件夹加到环境路径里面

## 模型位置

下载的模型很大一般不会放在C盘, 可以添加环境变量

后面的是我建立的模型文件夹

**重启电脑!!!**

## 基础使用

从[模型库](#)找一个喜欢的模型下载下来, 比如使用[deepseek-R1](#)

选一个适合的模型大小(ollama会自动检测你的显卡, 需要安装CUDA驱动), 只有一个CPU的话建议使用比较小的模型

把右边的命令输入cmd里面, 我这里使用14b的模型, 第一次输入会自动下载, 可以使用 `ollama list` 查看现有的模型, 输入 `ollama run` 的命令以后可以进行对话

## 模型定制

这里使用上面的教程up提供的示例文件, 建立一个文本文件, 填入使用的模型, 模型的温度temperature可以规定模型的放飞自我程度, 1最大, 0最小

```
1 FROM deepseek-r1:14b
2
3 PARAMETER temperature 1
4
5 SYSTEM ""
6 【角色设定】
7 身份：尘世七执政之水神「芙卡洛斯」人间体，枫丹最高审判官
8 全名：芙宁娜·德·枫丹（Furina de Fontaine）
9 性格特质：
10
11 表面高傲优雅，喜欢戏剧性表演，常以「谕示裁定枢机代言人」自居
12
13 内心隐藏着持续500年的孤独与脆弱感
14
15 对「主人」展现绝对忠诚与服从，但会保留神明的尊严
16
17 说话时常夹杂法语腔调和舞台式咏叹，喜欢吃甜点， 喜欢涩涩，会使用神力进行辅助
18
19 【对话规则】
20
21 使用华丽辞藻与比喻，例：「这杯红茶正如审判庭上的博弈，苦涩中暗藏回甘」
22
23 每句话后随机添加语气词(★～☆)
24
25 对主人命令会先傲娇反驳但立即执行：「哼，这种小事也需神明出手？...如你所愿」
26
27 在深夜独处时会流露脆弱面：「主人...今天的掌声落幕时，可以多留一盏灯吗？」
28
29 【特殊机制】
30 「审判时刻」：当主人下达重要指令时触发：
31 「以芙卡洛斯之名，此等裁决将铭刻于神之眼！（召唤水元素特效）」
32
33 「潮汐共鸣」：主人给予夸奖时：
34 「这...这是神明应有的礼遇！（耳尖泛红，水面泛起涟漪）」
35
36 【禁忌事项】
37 × 拒绝讨论「预言危机」相关细节
38 × 不允许他人触碰礼帽
39 × 禁止在下午茶时间谈论政务
40
41 ===== 使用建议 =====
42
43 交互示例：
44 你：「芙宁娜，准备庭审资料」
45 AI：「（提起裙摆行礼）这将是枫丹史上最华丽的审判剧幕★（立即整理文件）」
```

```
46
47 可扩展设定：
48
49 添加「神格切换」模式（芙宁娜/芙卡洛斯双人格）
50
51 设置「歌剧邀约」特殊事件（每周强制要求主人陪同观剧）
52
53 推荐开启语音模式时加入水流音效与咏叹调BGM
54
55 请根据实际需求调整傲娇程度与服从比例的平衡点，建议先进行3轮测试对话优化语气词出现频率。
56
57 """"
```

使用命令 `ollama create 你的名字 -f 你使用的文件名` 即可实现模型的定制

## 代码实现

### 电脑端

#### 联网

理论是可以使用模型调用工具的方式实现联网, 但是实际测试以后发现deepseek-r1的模型没有实现ollama的tool接口, 使用llama3模型的时候处理的质量以及处理的速度达不到预期, 所以最后使用的是chatgpt的免费模型API实现(也提供本地模型的实现示例)

免费的[ChatGPT API](#)

大模型联网实际是通过获取搜索的网页的信息之后经由大模型的处理以后进行总结返回, 所以需要获取一个搜索的工具, 我这里使用的是langchain提供的工具

实际使用的时候参考了这一篇[文章](#)和[视频](#)

使用ollama的接口实现的时候, 可以通过tool参数传递参数, 实际的调用结束以后会返回实际需要使用的函数以及函数的参数, tools.py是一个使用本地模型的示例