三 茴香豆搭建你的 RAG 智能助理笔记

https://www.bilibili.com/video/BV1QA4m1F7t4/

第三章

一、RAG技术概述

RAG（Retrieval Augmented Generation）技术，结合了信息检索和生成式问答的模型，能够通过检索用户输入相关的信息片段，并利用这些信息片段辅以外部知识库，以生成更准确、更丰富的回答。这样的技术可以解决大型语言模型（LLMs）在处理知识密集型任务时可能出现的问题，例如内容的虚构、信息的过时、以及缺乏清晰可追溯的推理过程。RAG的优势在于可以提升回答的准确度、降低推理的成本，并且能够实现外部知识的更新，无需再次对模型进行训练即可掌握新的知识领域。

二、环境配置与模型部署

在Intern Studio的服务器上配置和部署RAG技术。具体包括配置Conda环境、复制模型文件、以及git clone模型相关的代码。

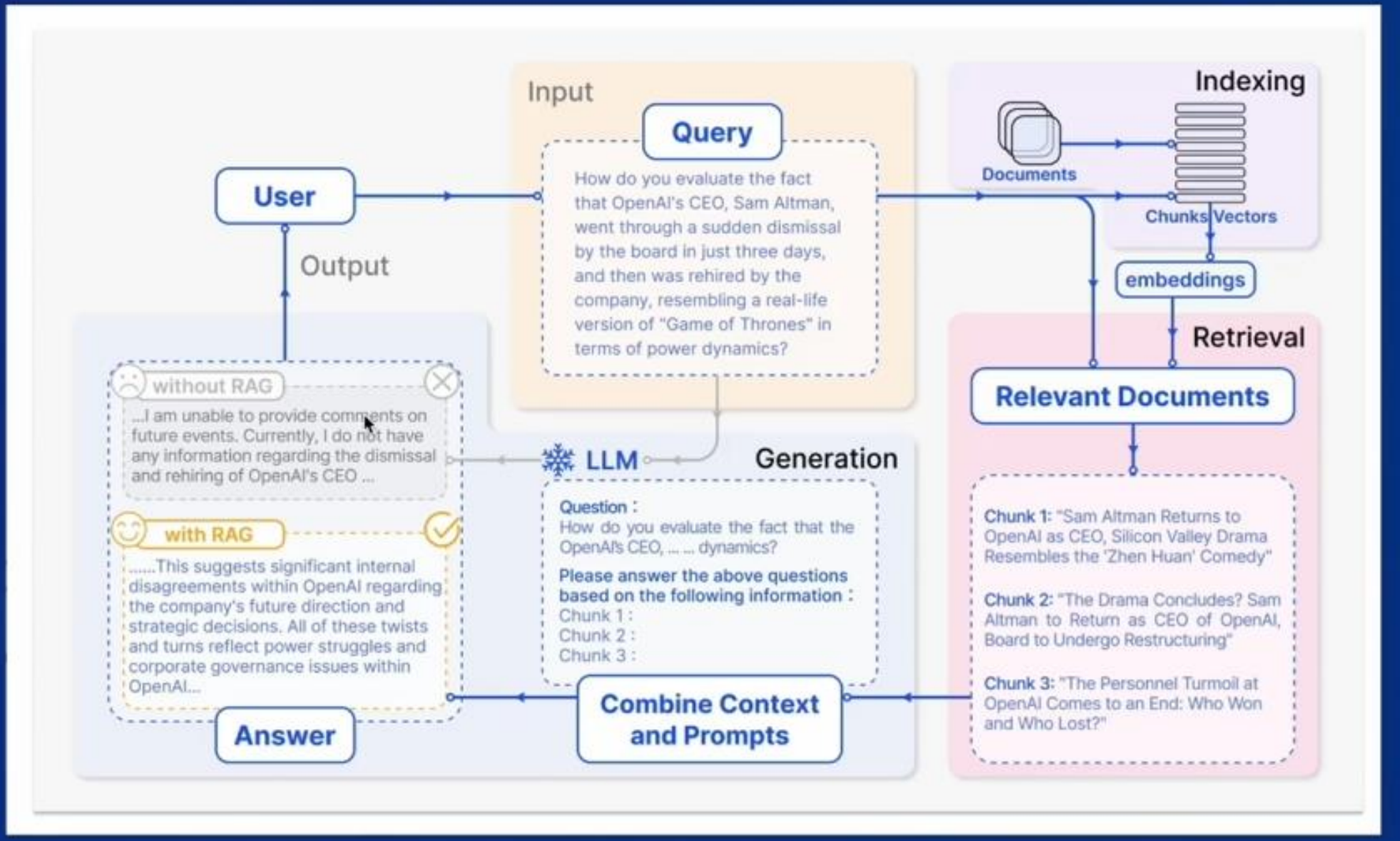
三、使用茴香豆搭建RAG助手

首先是修改配置文件，指定向量数据库、词嵌入模型、用于检索的重排序模型，以及所使用的语言模型（internlm2-chat-7b）。接下来创建知识库，使用InternLM的Huixiangdou文档作为新增知识的检索来源，以此建立技术问答助手。此外，还需要提取知识库特征来创建向量数据库，使用的是LangChain的相关模块和网易BCE双语模型。

为了让RAG更加精准地判断提问的相关性，茴香豆还建立了接受问题和拒绝问题两个向量数据库，其中包括希望茴香豆助手回答或拒答的示例问题。例如，接受问题列表中的问题存储在huixiangdou/resource/good_questions.json中，而拒绝问题列表中的则存储在huixiangdou/resource/bad_questions.json中，主要涵盖了一些技术无关或闲聊性质的问题。完成语料来源设定后，执行命令创建RAG在检索过程中使用的向量数据库。

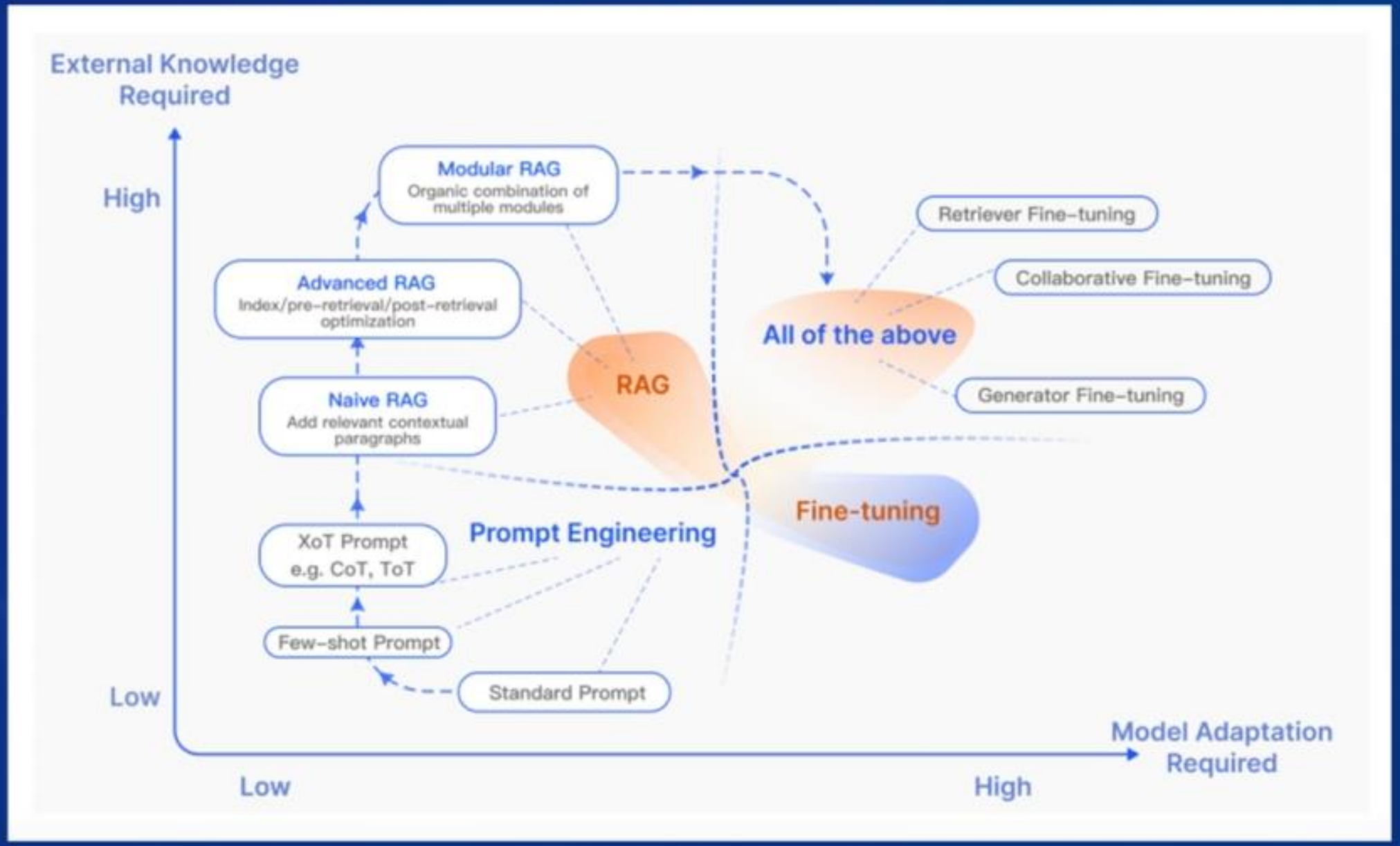
在实际检索过程中，茴香豆会将输入的问题与两个列表中的问题进行向量空间的相似性比较，以判断问题是否应当回答，从而避免了问答的泛滥。确定要回答的问题之后，基础模型会提取关键词，并在知识库中检索与之最相似的信息片段（chunk），最终结合这些信息生成回答。

RAG 流程示例



Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H., 2024. Retrieval-Augmented Generation for Large Language Models: A Survey.

LLM 模型优化方法比较



Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, Dai Y, Sun J, Wang M, Wang H. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey

复制完成后，在本地查看环境。

```
conda env list
```

结果如下所示。

```
# conda environments:
#
base                  *  /root/.conda
InternLM2_Huixiangdou                /root/.conda/envs/InternLM2_Huix:
```

运行 ***conda*** 命令，激活 `InternLM2_Huixiangdou` ***python*** 虚拟环境:

```
conda activate InternLM2_Huixiangdou
```

环境激活后，命令行左边会显示当前（也就是 `InternLM2_Huixiangdou` ）的环境名称，如下图所示:

```
    ==========================================
                  ALL DONE!
    ==========================================

(base) root@intern-studio-40059224:~# conda env list
# conda environments:
#
base                  *  /root/.conda
InternLM                 /root/.conda/envs/InternLM
InternLM2_Huixiangdou     /root/.conda/envs/InternLM2_Huixiangdou
internlm-demo            /root/.conda/envs/internlm-demo
opencompass              /root/.conda/envs/opencompass
xcomposer-demo           /root/.conda/envs/xcomposer-demo
xtuner0.1.9              /root/.conda/envs/xtuner0.1.9

(base) root@intern-studio-40059224:~# conda activate InternLM2_Huixiangdou
(InternLM2_Huixiangdou)root@intern-studio-40059224:~#
```

```
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/80/03/6ea8b1b2a5ab40a7a60
dc464d3daa7aa546e0a74d74a9f8ff551ea7905db/executing-2.0.1-py2.py3-none-any.whl (24
kB)
Collecting asttokens>=2.1.0 (from stack-data->ipython>=7.23.1->ipykernel)
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/45/86/4736ac618d82a20d87d
2f92ae19441ebc7ac9e7a581d7e58bbe79233b24a/asttokens-2.4.1-py2.py3-none-any.whl (27
kB)
Collecting pure-eval (from stack-data->ipython>=7.23.1->ipykernel)
  Using cached https://pypi.tuna.tsinghua.edu.cn/packages/2b/27/77f9d5684e6bce929f5
cfe18d6cfbe5133013c06cb2fbf5933670e60761d/pure_eval-0.2.2-py3-none-any.whl (11 kB)
Installing collected packages: wcwidth, pure-eval, ptyprocess, traitlets, tornado,
six, pyzmq, pygments, psutil, prompt-toolkit, platformdirs, pexpect, parso, packagi
ng, nest-asyncio, executing, exceptiongroup, decorator, debugpy, python-dateutil, m
atplotlib-inline, jupyter-core, jedi, comm, asttokens, stack-data, jupyter-client,
ipython, ipykernel
Successfully installed asttokens-2.4.1 comm-0.2.2 debugpy-1.8.1 decorator-5.1.1 exc
eptiongroup-1.2.0 executing-2.0.1 ipykernel-6.29.4 ipython-8.23.0 jedi-0.19.1 jupyt
er-client-8.6.1 jupyter-core-5.7.2 matplotlib-inline-0.1.6 nest-asyncio-1.6.0 packa
ging-24.0 parso-0.8.4 pexpect-4.9.0 platformdirs-4.2.0 prompt-toolkit-3.0.43 psutil
-5.9.8 ptyprocess-0.7.0 pure-eval-0.2.2 pygments-2.17.2 python-dateutil-2.9.0.post0
 pyzmq-25.1.2 six-1.16.0 stack-data-0.6.3 tornado-6.4 traitlets-5.14.2 wcwidth-0.2.
13
WARNING: Running pip as the 'root' user can result in broken permissions and confli
cting behaviour with the system package manager. It is recommended to use a virtual
 environment instead: https://pip.pypa.io/warnings/venv
Installed kernelspec InternLM2_Huixiangdou in /root/.local/share/jupyter/kernels/in
ternlm2_huixiangdou
 conda环境：InternLM2_Huixiangdou安装成功！

    ==========================================
                  ALL DONE!
    ==========================================

(base) root@intern-studio-40059224:/opt/jupyterlab# conda env list
# conda environments:
#
base                  *  /root/.conda
InternLM                 /root/.conda/envs/InternLM
InternLM2_Huixiangdou     /root/.conda/envs/InternLM2_Huixiangdou
internlm-demo            /root/.conda/envs/internlm-demo
opencompass              /root/.conda/envs/opencompass
xcomposer-demo           /root/.conda/envs/xcomposer-demo
xtuner0.1.9              /root/.conda/envs/xtuner0.1.9

(base) root@intern-studio-40059224:/opt/jupyterlab#
```

2.3 运行茴香豆知识助手

我们已经提取了知识库特征，并创建了对应的向量数据库。现在，让我们来测试一下效果：

命令行运行：

```
# 填入问题
sed -i '74s/.*/    queries = ["huixiangdou 是什么？", "茴香豆怎么部署到微信
# 运行茴香豆
cd /root/huixiangdou/
python3 -m huixiangdou.main --standalone
```

RAG 技术的优势就是非参数化的模型调优，这里使用的仍然是基础模型 `InternLM2-Chat-7B`， 没有任何额外数据的训练。面对同样的问题，我们的茴香豆技术助理能够根据我们提供的数据库生成准确的答案：

```
2024-04-05 00:33:31.503 | INFO     | __main__:lark_send_only:79 - ErrorCode.SUCCESS, huixiangdou 是什么？, "huixiangdou" 是一个基于 LLM 的领域特定知识助手，具有以下特点：
1. 能够处理复杂的场景，如群组聊天，并能够回答用户问题，而不会造成消息泛滥。
2. 提供算法管道，以回答技术性问题。
3. 部署成本低，只需要 LLM 模型满足 4 个特征，即可回答大多数用户的问题，详情请参阅 [arxiv2401.08772](https://arxiv.org/abs/2401.08772)。
您可以在 [WeChat 群](./huixiangdou-inside.md) 中查看 "huixiangdou" 的运行场景，并加入 [WeChat 群](resource/figures/wechat.jpg) 以体验 AI 助手。
如果这对您有帮助，请给 "huixiangdou" 一个星标 🌟 , ['README.md']


2024-04-05 00:33:56.986 | INFO     | __main__:lark_send_only:79 - ErrorCode.SUCCESS, 茴香豆怎么部署到微信群，要部署 "茴香豆" 到微信群，请按照以下步骤操作：

1. **安装 "茴香豆" 的 Web 版**:
   - 首先，访问 [OpenXLab](https://openxlab.org.cn/apps/detail/tpoisonooo/huixiangdou-web) 并注册一个账号。
   - 然后，在 OpenXLab 中安装 "茴香豆" 的 Web 版应用。

2. **创建自己的知识库**:
   - 在 Web 版 "茴香豆" 中，创建一个新的知识库。
   - 将您希望 "茴香豆" 能够回答的问题和相应的答案添加到该知识库中。
   - 配置 `data` 参数，以指定您要使用的数据文件。
   - 配置 `log` 参数，以指定日志文件的路径。

4. **运行茴香豆**:
   - 在终端中，导航到茴香豆的根目录。
   - 运行 `python huixiangdou.py` 启动茴香豆。
   - 茴香豆将开始处理数据，并准备回答用户的问题。

5. **集成到微信群**:
   - 在您的微信中，创建一个新的群聊。
   - 将茴香豆的 QQ 号添加到该群聊中。
   - 茴香豆将开始接收来自微信群的消息，并尝试回答用户的问题。

6. **测试和优化**:
   - 测试茴香豆的性能，并根据需要进行调整。
   - 您可以通过添加更多数据、调整参数或使用更高级的模型来提高茴香豆的性能。

请注意，以上步骤仅提供了一个基本的部署流程，您可能需要根据您的具体需求进行一些调整。同时，为了确保茴香豆的正常运行，请确保您的计算机具有足够的计算资源，并且您的数据集是干净、有组织且格式正确的。, ['README_zh.md']
2024-04-06 13:03:54.742 | INFO     | huixiangdou.service.llm_server_hybrid:generate_response:519 - ('"今天天气怎么样？"\n请仔细阅读以上内容，判断句子是否是个有主题的疑问句，结果用 0~10 表示。直接提供得分不要解释。\n判断标准：有主语谓语宾语并且是疑问句得 10 分；缺少主谓宾扣分；陈述句直接得 0 分；不是疑问句直接得 0 分。直接提供得分不要解释。', '根据给定的标准，"今天天气怎么样？" 是一个有主语、谓语和宾语，并且是疑问句的句子。因此，它的得分是 10 分。')
2024-04-06 13:03:54.742 | DEBUG    | huixiangdou.service.llm_server_hybrid:generate_response:522 - Q:有主题的疑问句，结果用 0~10 表示。直接提供得分不要解释。
判断标准：有主语谓语宾语并且是疑问句得 10 分；缺少主谓宾扣分；陈述句直接得 0 分；不是疑问句直接得 0 分。直接提供得分不要解释 A:根据给定的标准，"今天天气怎么样？" 是一个有主语、谓语和宾语，并且是疑问句的句子。因此，它的得分是 10 分。 remote local timecost 1.4027836322784424
04/06/2024 13:03:54 - [INFO] -aiohttp.access->>>  127.0.0.1 [06/Apr/2024:13:03:53 +0800] "POST /inference HTTP/1.1" 200 474 "-" "python-requests/2.31.0"
2024-04-06 13:03:54.903 | INFO     | huixiangdou.service.llm_server_hybrid:generate_response:519 - ('告诉我这句话的主题，直接说主题不要解释："今天天气怎么样？"', '主题：天气。')
2024-04-06 13:03:54.903 | DEBUG    | huixiangdou.service.llm_server_hybrid:generate_response:522 - Q:告诉我这句话的主题，直接说主题不要解释："今天天气怎么样？ A:主题：天气。 remote local timecost 0.15743017196655273
04/06/2024 13:03:54 - [INFO] -aiohttp.access->>>  127.0.0.1 [06/Apr/2024:13:03:54 +0800] "POST /inference HTTP/1.1" 200 206 "-" "python-requests/2.31.0"
2024-04-06 13:03:54.917 | INFO     | __main__:lark_send_only:79 - ErrorCode.UNRELATED, 今天天气怎么样？, , ['HuixiangDou.pdf']
(InternLM2_Huixiangdou) root@intern-studio-40059224:~/huixiangdou#
```