

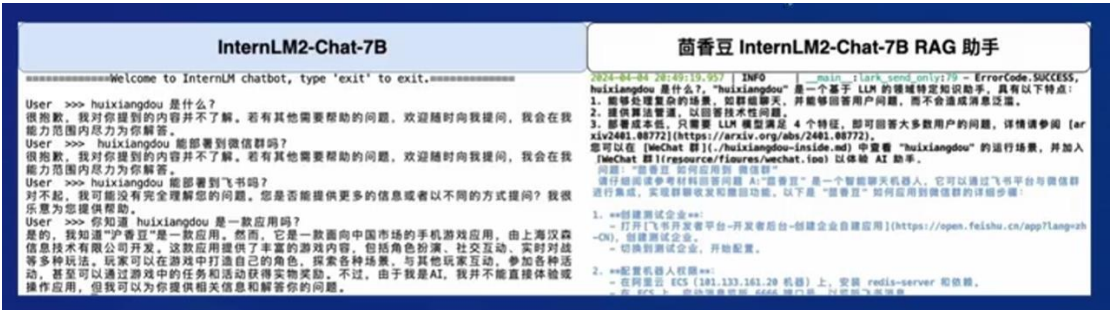
Lesson 3 笔记

茴香豆：零代码搭建你的 RAG 智能助理

这节课分 3 部分学习，包括 RAG 基础知识介绍、茴香豆介绍和茴香豆搭建知识库实战。

第一部分

左图中关于 huixiangdou 的 3 轮问答均未给出准确的答案。右图未对 InternLM2-Chat-7B 进行任何增训的情况下，通过 RAG 技术实现的新增知识问答，看起来还不错。



RAG 技术概述

定义

RAG (Retrieval Augmented Generation) 是一种结合了检索 (Retrieval) 和生成 (Generation) 的技术，旨在通过利用**外部知识库**来增强大型语言模型 (LLMs) 的性能。它通过检索与用户输入相关的信息片段，并结合这些信息来生成更准确、更丰富的回答。

知识

解决LLMs在处理**知识密集型任务**时可能遇到的挑战。提供更准确的回答、降低成本、实现外部记忆。

V5

- 生成幻觉 (hallucination)
- 过时知识
- 缺乏透明和可追溯的推理过程

应用

问答系统

文本生成

信息检索

图片描述

RAG 工作原理



向量数据库 (Vector-DB)

数据存储

将文本及其他数据通过其他预训练的模型转换为固定长度的向量表示，这些向量能够捕捉文本的语义信息。

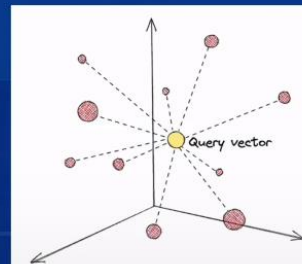
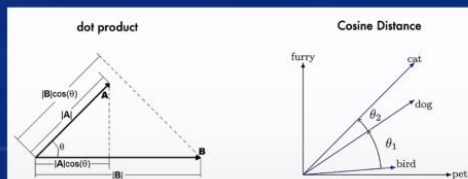
相似性检索

根据用户的查询向量，使用向量数据库快速找出最相关的向量的过程。通常通过计算余弦相似度或其他相似性度量来完成。检索结果根据相似度得分进行排序，最相关的文档将被用于后续的文本生成。

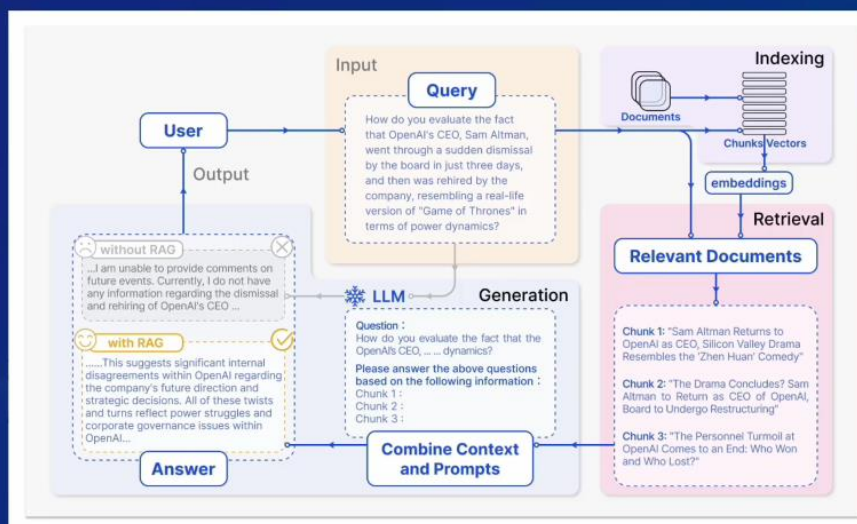
向量表示的优化

包括使用更高级的文本编码技术，如句子嵌入或段落嵌入，以及对数据库进行优化以支持大规模向量搜索。

Images from :
<https://github.com/chenzom12/AlSystem/blob/main/06Foundation/05Dataset/04VectorDB.pdf>

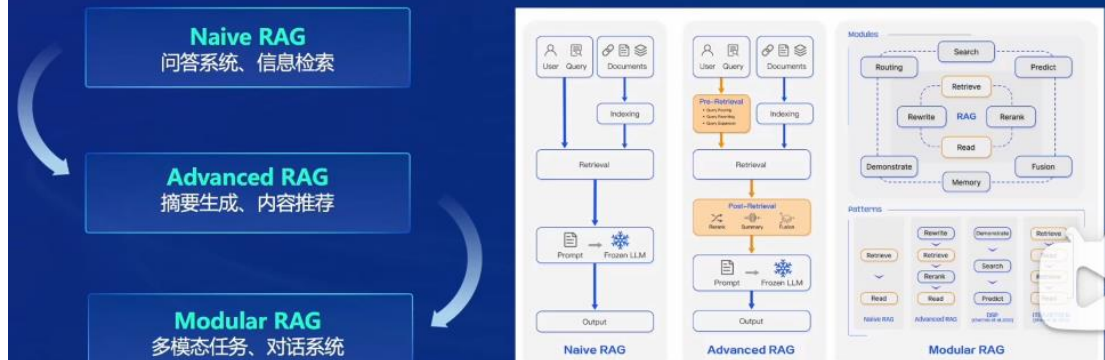


RAG 流程示例



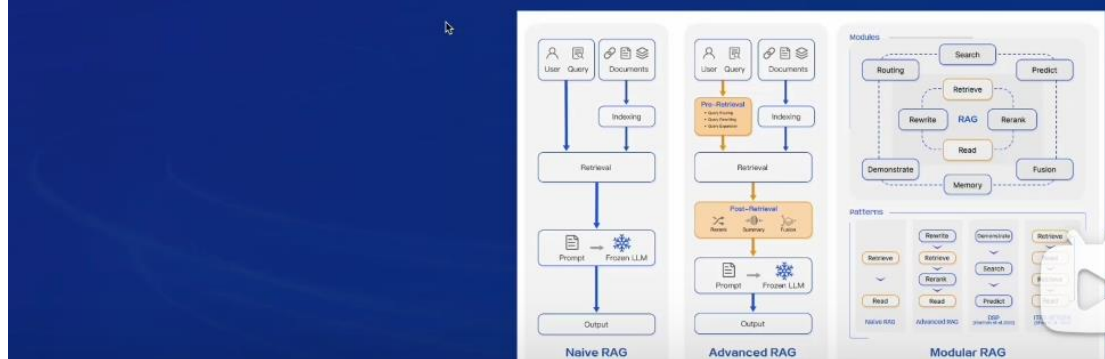
RAG 发展进程

RAG的概念最早是由Meta (Facebook) 的Lewis等人在2020《Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks》中提出的。



RAG 发展进程

RAG的概念最早是由Meta (Facebook) 的Lewis等人在2020《Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks》中提出的。



RAG 常见优化方法



RAG vs. 微调 (Fine-tuning)

RAG

- 非参数记忆，利用外部知识库提供实时更新的信息。
- 能够处理知识密集型任务，提供准确的事实性回答。
- 通过检索增强，可以生成更多样化的内容。

适用场景

适用于需要结合最新信息和实时数据的任务：开放域问答、实时新闻摘要等。

优势：动态知识更新，处理长尾知识问题。

局限：依赖于外部知识库的质量和覆盖范围。依赖大模型能力。



Fine-tuning

- 参数记忆，通过在特定任务数据上训练，模型可以更好地适应该任务。
- 通常需要大量标注数据来进行有效微调。
- 微调后的模型可能过拟合，导致泛化能力下降。

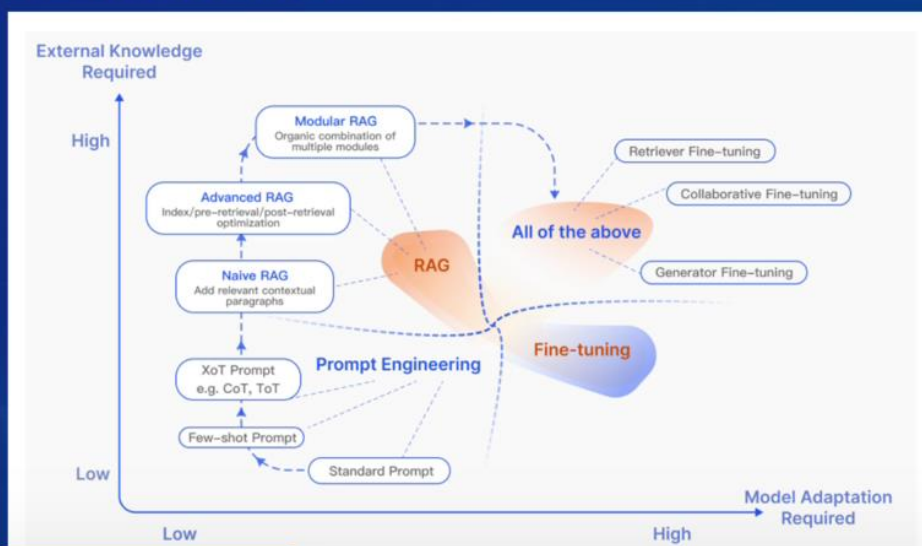
适用场景

适用于数据可用且需要模型高度专业化的任务，如特定领域的文本分类、情感分析、文本生成等。

优势：模型性能针对特定任务优化。

局限：需要大量的标注数据，且对新任务的适应性较差。

LLM 模型优化方法比较



评估框架和基准测试

经典评估指标：

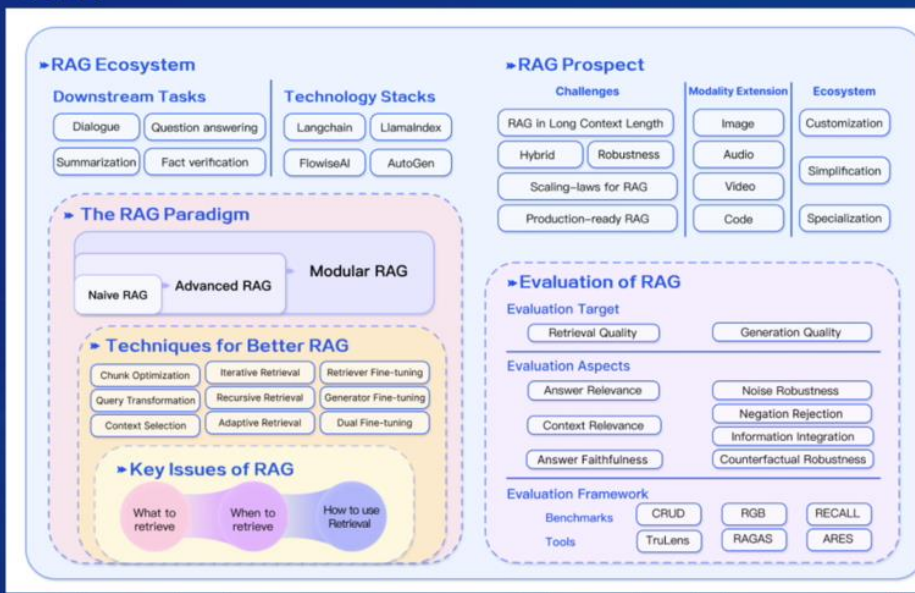
- 准确率 (Accuracy)
- 召回率 (Recall)
- F1分数 (F1 Score)
- BLEU分数 (用于机器翻译和文本生成)
- ROUGE分数 (用于文本生成的评估)

RAG 评测框架：

- 基准测试 - RGB、RECALL、CRUD
- 评测工具 - RAGAS、ARES、TruLens

Evaluation Framework	Evaluation Targets	Evaluation Aspects	Quantitative Metrics
RGB [†]	Retrieval Quality Generation Quality	Noise Robustness Negative Rejection Information Integration Counterfactual Robustness	Accuracy EM Accuracy Accuracy
RECALL [‡]	Generation Quality	Counterfactual Robustness	R-Rate (Reappearance Rate)
RAGAS [‡]	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	* * Cosine Similarity
ARES [‡]	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	Accuracy Accuracy Accuracy
TruLens [‡]	Retrieval Quality Generation Quality	Context Relevance Faithfulness Answer Relevance	* * *
CRUD [†]	Retrieval Quality Generation Quality	Creative Generation Knowledge-intensive QA Error Correction Summarization	BLEU ROUGE-L BertScore RAGQuestEval

RAG 总结



第二部分

茴香豆介绍



茴香豆是一个基于LLMs的领域知识助手，由书生浦语团队开发的开源大模型应用。

- 专为即时通讯（IM）工具中的群聊场景优化的工作流，提供及时准确的技术支持和自动化问答服务。
- 通过应用检索增强生成（RAG）技术，茴香豆能够理解和高效准确的回应与特定知识领域相关的复杂查询。

应用场景

- 智能客服：技术支持、领域知识对话
- IM工具中创建用户群组，讨论、解答相关的问题。
- 随着用户数量的增加，答复内容高度重复，充斥大量无意义和闲聊，人工回复，成本高，影响工作效率。
- 茴香豆通过提供自动化的问答支持，帮助维护者减轻负担，同时确保用户问题得到有效解答。

场景难点

- 群聊中的信息量巨大，且内容多样，从技术讨论到闲聊应有尽有。
- 用户问题通常与个人紧密相关，需要准确的实时的专业知识解答。
- 传统的NLP解决方案无法准确解析用户意图，且往往无法提供满意的答案。
- 需要一个能够在群聊中准确识别与回答相关问题的智能助手，同时避免造成消息过载。



茴香豆的核心特性



开源免费
BSD-3-Clause
免费商用



高效准确
Hybrid LLMs
专为群聊优化



领域知识
应用RAG技术
专业知识快速获取



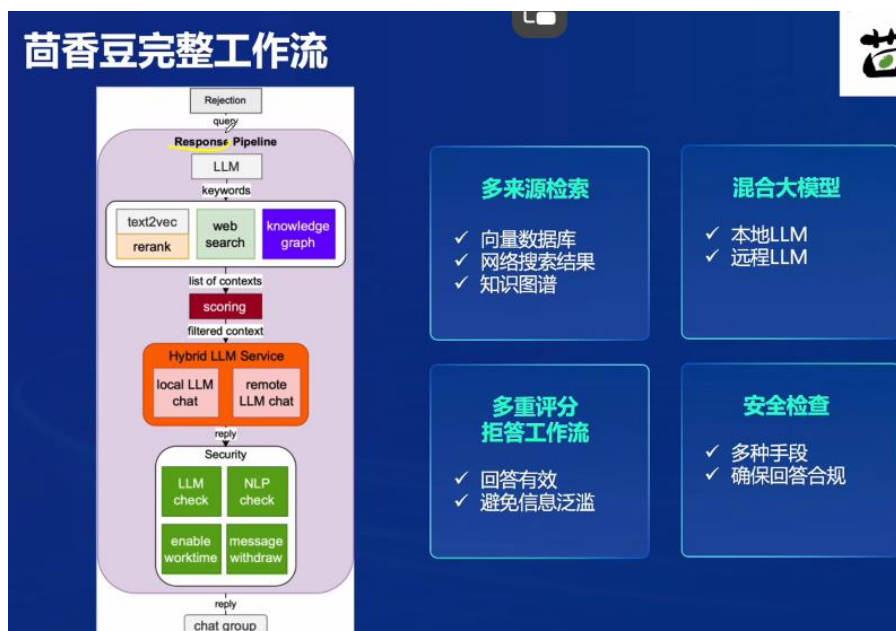
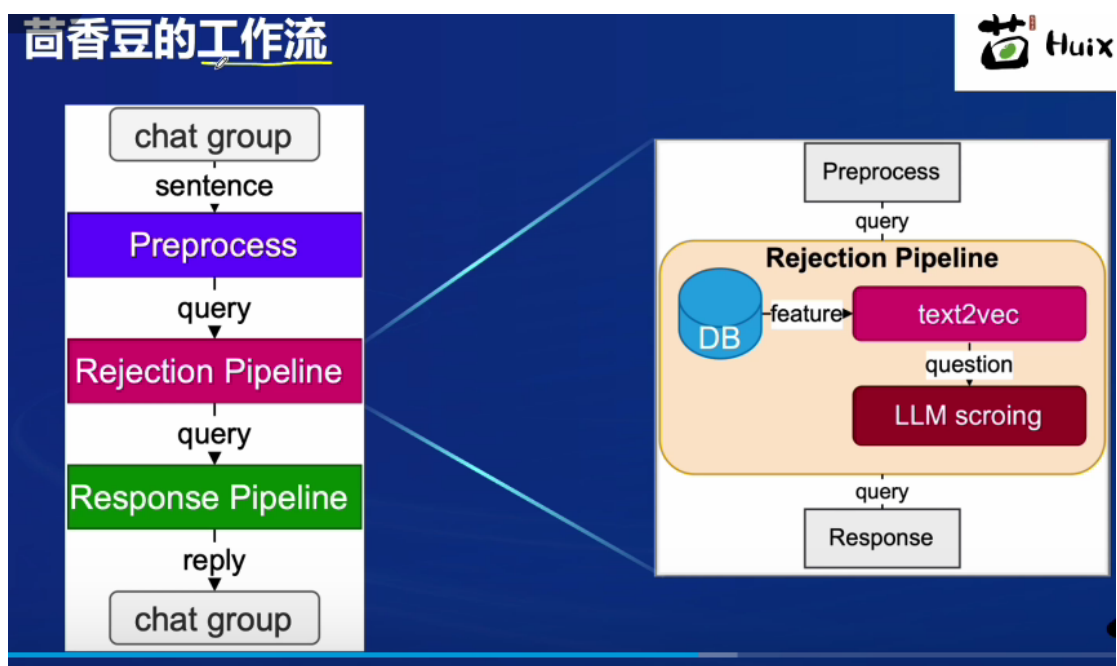
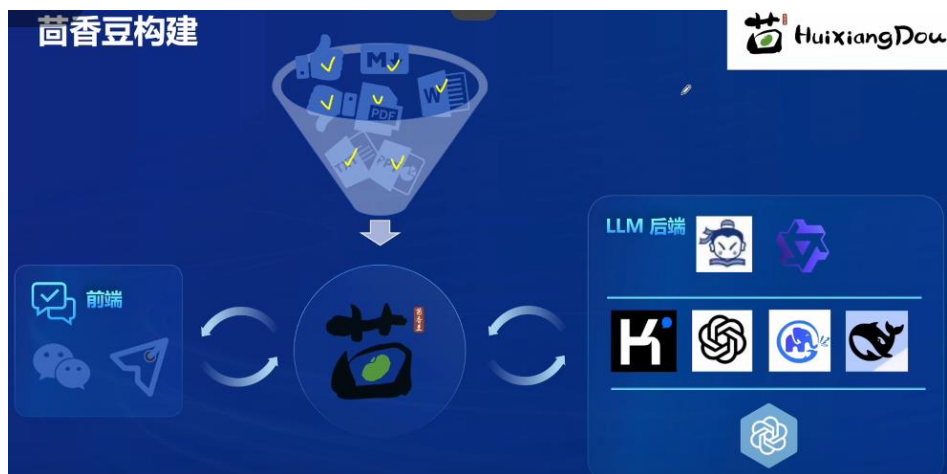
部署成本低
无需额外训练
可利用云端模型api，
本地算力需求少



安全
可完全本地部署，
信息不上传
保护数据和用户隐私



扩展性强
兼容多种IM软件
支持多种开源LLMs
和云端api



第三部分：实践

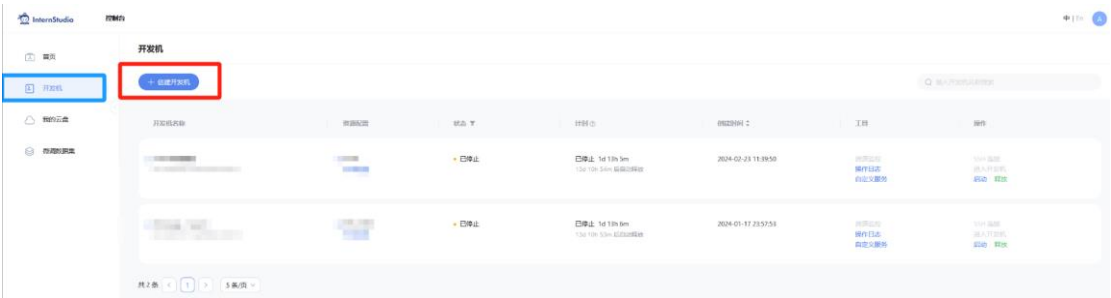
【1】



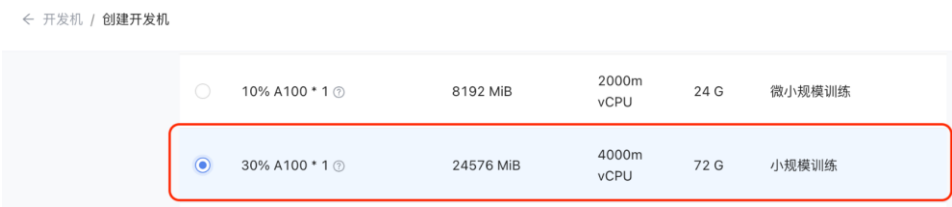
1.1 配置基础环境

这里以在 Intern Studio 服务器上部署茴香豆为例。

首先，打开 Intern Studio 界面，点击 创建开发机 配置开发机系统。



填写 开发机名称 后，点击 选择镜像 使用 Cuda11.7-conda 镜像，然后在资源配置中，使用 30% A100 * 1 的选项，然后立即创建开发机器。



点击 进入开发机 选项。



进入开发机后，从官方环境复制运行 InternLM 的基础环境，命名为 InternLM2_Huixiangdou，在命令行模式下运行：

```
studio-conda -o internlm-base -t InternLM2_Huixiangdou
```

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_Huixiangdou
```

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:~#
```

后续教程所有操作都需要在该环境下进行，重启开发机或打开新命令行后要重新激活环境。

1.2 下载基础文件

复制茴香豆所需模型文件，为了减少下载和避免 HuggingFace 登录问题，所有作业和教程涉及的模型都已经存放在 Intern Studio 开发机共享文件中。本教程选用

InternLM2-Chat-7B 作为基础模型。

创建模型文件夹

```
cd /root && mkdir models
```

复制 BCE 模型

```
ln -s /root/share/new_models/maidalun1020/bce-embedding-base_v1  
/root/models/bce-embedding-base_v1
```

```
ln -s /root/share/new_models/maidalun1020/bce-reranker-base_v1 /root/models/bce-  
reranker-base_v1
```

复制大模型参数（下面的模型，根据作业进度和任务进行**选择一个**就行）

```
ln -s /root/share/new_models/Shanghai_AI_Laboratory/internlm2-chat-7b  
/root/models/internlm2-chat-7b
```

1.3 下载安装茴香豆

安装茴香豆运行所需依赖。

安装 python 依赖

```
# pip install -r requirements.txt
```

```
pip install protobuf==4.25.3 accelerate==0.28.0 aiohttp==3.9.3 auto-gptq==0.7.1  
bcembedding==0.1.3 beautifulsoup4==4.8.2 einops==0.7.0 faiss-gpu==1.7.2  
langchain==0.1.14 loguru==0.7.2 lxml_html_clean==0.1.0 openai==1.16.1  
openpyxl==3.1.2 pandas==2.2.1 pydantic==2.6.4 pymupdf==1.24.1 python-  
docx==1.1.0 pytoml==0.1.21 readability-lxml==0.8.1 redis==5.0.3 requests==2.31.0  
scikit-learn==1.4.1.post1 sentence_transformers==2.2.2 textract==1.6.5  
tiktoken==0.6.0 transformers==4.39.3 transformers_stream_generator==0.0.5  
unstructured==0.11.2
```

从茴香豆官方仓库下载茴香豆。

```
cd /root
```

下载 repo

```
git clone https://github.com/internlm/huixiangdou && cd huixiangdou
```

```
git checkout 447c6f7e68a1657fce1c4f7c740ea1700bde0440
```

茴香豆工具在 Intern Studio 开发机的安装工作结束。

【2】

使用茴香豆搭建 RAG 助手

2.1 修改配置文件

用已下载模型的路径替换 `/root/huixiangdou/config.ini` 文件中的默认模型，需要修改 3 处模型地址，分别是：

命令行输入下面的命令，修改用于向量数据库和词嵌入的模型

```
sed -i '6s#.*#embedding_model_path = "/root/models/bce-embedding-base_v1"##'
/root/huixiangdou/config.ini
```

用于检索的重排序模型：

```
sed -i '7s#.*#reranker_model_path = "/root/models/bce-reranker-base_v1"##'
/root/huixiangdou/config.ini
```

和本次选用的大模型

```
sed -i '29s#.*#local_llm_path = "/root/models/internlm2-chat-7b"##'
/root/huixiangdou/config.ini
```

修改好的配置文件应该如下图所示：

```
[feature_store]
reject_throttle = 0.9
embedding_model_path = "/root/models/bce-embedding-base_v1"
reranker_model_path = "/root/models/bce-reranker-base_v1"
work_dir = "workdir"

[web_search]
x_api_key = "${YOUR-API-KEY}"
domain_partial_order = ["openai.com", "pytorch.org", "readthedocs.io", "nvidia.com", "stackoverflow.com",
"juejin.cn", "zhuankan.zhihu.com", "www.cnblogs.com"]
save_dir = "logs/web_search_result"

[llm]
enable_local = 1
enable_remote = 0
client_url = "http://127.0.0.1:8888/inference"

[llm.server]
local_llm_path = "/root/models/internlm2-chat-7b"
```

配置文件具体含义和更多细节参考 3.4 配置文件解析。

2.2 创建知识库

本示例中，使用 InternLM 的 Huixiangdou 文档作为新增知识数据检索来源，在不重新训练的情况下，打造一个 Huixiangdou 技术问答助手。

首先，下载 Huixiangdou 语料：

```
cd /root/huixiangdou && mkdir repodir
```

```
git clone https://github.com/internlm/huixiangdou --depth=1 repodir/huixiangdou
```

提取知识库特征，创建向量数据库。数据库向量化的过程应用到了 LangChain 的相关模块，默认嵌入和重排序模型调用的网易 BCE 双语模型，如果没有在 config.ini 文件中指定本地模型路径，茴香豆将自动从 HuggingFace 拉取默认模型。

除了语料知识的向量数据库，茴香豆建立接受和拒答两个向量数据库，用来在检索的过程中更加精确的判断提问的相关性，这两个数据库的来源分别是：

接受问题列表，希望茴香豆助手回答的示例问题

存储在 huixiangdou/resource/good_questions.json 中

拒绝问题列表，希望茴香豆助手拒答的示例问题

存储在 huixiangdou/resource/bad_questions.json 中

其中多为技术无关的主题或闲聊

如："nihui 是谁"，"具体在哪些位置进行修改？"，"你是谁？"，"1+1"

运行下面的命令，增加茴香豆相关的问题到接受问题示例中：

```
cd /root/huixiangdou
```

```
mv resource/good_questions.json resource/good_questions_bk.json
```

```
echo '[
```

```
    "mmpose 中怎么调用 mmyolo 接口",
```

```
    "mmpose 实现姿态估计后怎么实现行为识别",
```

```
    "mmpose 执行提取关键点命令不是分为两步吗，一步是目标检测，另一步是关键点提取，我现在目标检测这部分的代码是 demo/topdown_demo_with_mmdet.py
```

```
demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py
```

```
checkpoints/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth    现在我想把这个 mmdet 的 checkpoints 换位 yolo 的，那么应该怎么操作",
```

```
    "在 mmdetection 中，如何同时加载两个数据集，两个 dataloader",
```

```
    "如何将 mmdetection2.28.2 的 retinanet 配置文件改为单尺度的呢？",
```

```
    "1.MMPose_Tutorial.ipynb、inferencer_demo.py、image_demo.py、bottomup_demo.py、body3d_pose_lifter_demo.py 这几个文件和
```

```
topdown_demo_with_mmdet.py 的区别是什么，\n2.我如果要使用 mmdet 是不是就只能使用 topdown_demo_with_mmdet.py 文件，",
```

"mmpose 测试 map 一直是 0 怎么办？",

"如何使用 mmpose 检测人体关键点？",

"我使用的数据集是 labelme 标注的, 我想知道 mmpose 的数据集都是什么样式的, 全都是单目标的数据集标注, 还是里边也有多目标然后进行标注",

"如何生成 openmmpose 的 c++推理脚本",

"mmpose",

"mmpose 的目标检测阶段调用的模型, 一定要是 demo 文件夹下的文件吗, 有没有其他路径下的文件",

"mmpose 可以实现行为识别吗, 如果要实现的话应该怎么做",

"我在 mmyolo 的 v0.6.0 (15/8/2023)更新日志里看到了他新增了支持基于 MMPose 的 YOLOX-Pose, 我现在是不是只需要在 mmpose/project/yolox-Pose 内做出一些设置就可以, 换掉 demo/mmdetection_cfg/faster_rcnn_r50_fpn_coco.py 改用 mmyolo 来进行目标检测了",

"mac m1 从源码安装的 mmpose 是 x86_64 的",

"想请教一下 mmpose 有没有提供可以读取外接摄像头, 做 3d 姿态并达到实时的项目呀？",

"huixiangdou 是什么？",

"使用科研仪器需要注意什么？",

"huixiangdou 是什么？",

"茴香豆 是什么？",

"茴香豆 能部署到微信吗？",

"茴香豆 怎么应用到飞书",

"茴香豆 能部署到微信群吗？",

"茴香豆 怎么应用到飞书群",

"huixiangdou 能部署到微信吗？",

"huixiangdou 怎么应用到飞书",

"huixiangdou 能部署到微信群吗？",

"huixiangdou 怎么应用到飞书群",

"huixiangdou",


```
"茴香豆",  
"茴香豆 有哪些应用场景",  
"huixiangdou 有什么用",  
"huixiangdou 的优势有哪些? ",  
"茴香豆 已经应用的场景",  
"huixiangdou 已经应用的场景",  
"huixiangdou 怎么安装",  
"茴香豆 怎么安装",  
"茴香豆 最新版本是什么",  
"茴香豆 支持哪些大模型",  
"茴香豆 支持哪些通讯软件",  
"config.ini 文件怎么配置",  
"remote_llm_model 可以填哪些模型?"
```

```
] ' > /root/huixiangdou/resource/good_questions.json
```

再创建一个测试用的问询列表，用来测试拒答流程是否起效：

```
cd /root/huixiangdou
```

```
echo '['
```

```
"huixiangdou 是什么? ",
```

```
"你好，介绍下自己"
```

```
] ' > ./test_queries.json
```

在确定好语料来源后，运行下面的命令，创建 RAG 检索过程中使用的向量数据库：

```
# 创建向量数据库存储目录
```

```
cd /root/huixiangdou && mkdir workdir
```

```
# 分别向量化知识语料、接受问题和拒绝问题中后保存到 workdir
```

```
python3 -m huixiangdou.service.feature_store --sample ./test_queries.json
```

向量数据库的创建需要等待一小段时间，过程约占用 1.6G 显存。

完成后，Huixiangdou 相关的新增知识就以向量数据库的形式存储在 workdir 文件夹下。

检索过程中，茴香豆会将输入问题与两个列表中的问题在向量空间进行相似性比较，判断该问题是否应该回答，避免群聊过程中的问答泛滥。确定的回答的问题会利用基础模型提取关键词，在知识库中检索 top K 相似的 chunk，综合问题和检索到的 chunk 生成答案。

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群", "今天天气怎么样? "]/' /root/huixiangdou/huixiangdou/main.py
```

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.SUCESS, 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.SUCESS, 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。
3. ****更新正反例**:**
 - 在 Web 版 "茴香豆" 中, 您可以通过添加和编辑正反例来进一步提高 "茴香豆" 的性能。
 - 正例是正确的问题和答案对, 而反例是不正确的问题和答案对。
4. ****开关网络搜索**:**
 - 在 Web 版 "茴香豆" 中, 您可以选择是否启用网络搜索。
 - 启用网络搜索可以让 "茴香豆" 在知识库中找不到答案时, 从互联网上搜索相关信息。
5. ****集成到微信群**:**
 - 在 Web 版 "茴香豆" 中, 将 "茴香豆" 的 Web URL 或 API 密钥与您的微信群进行集成。
 - 您可以使用 [飞书](https://www.feishu.cn/) 或 [微信群](https://.weixin.qq.com/) 的 API 来集成 "茴香豆"。
6. ****测试和优化**:**
 - 在集成 "茴香豆" 到微信群后, 请进行测试以确保 "茴香豆" 可以正常工作。
 - 您可以根据测试结果来优化 "茴香豆" 的性能, 比如调整正反例、网络搜索的开关等。
7. ****分享和推广**:**
 - 如果您觉得 "茴香豆" 很有用, 请与他人分享, 并鼓励他们 also 使用 "茴香豆"。
 - 您可以在社交媒体、技术论坛等地方分享 "茴香豆" 的使用体验, 以帮助更多人发现和使用 "茴香豆"。

通过以上步骤, 您就可以将 "茴香豆" 部署到微信群中, 并让 "茴香豆" 帮助您和您的微信群成员解决问题。记得在部署过程中仔细阅读 "茴香豆" 的文档和注意事项, 以确保 "茴香豆" 的正常运行。 , ['README_zh.md']

InternLM2-Chat-7B 的关于 huixiangdou 问题的原始输出:

=====Welcome to InternLM chatbot, type 'exit' to exit.=====

User >>> huixiangdou 是什么?

很抱歉, 我对你提到的内容并不了解。若有其他需要帮助的问题, 欢迎随时向我提问, 我会在我能力范围内尽力为你解答。

User >>> huixiangdou 能部署到微信群吗?

很抱歉, 我对你提到的内容并不了解。若有其他需要帮助的问题, 欢迎随时向我提问, 我会在我能力范围内尽力为你解答。

User >>> huixiangdou 能部署到飞书吗?

对不起, 我可能没有完全理解您的问题。您是否能提供更多的信息或者以不同的方式提问? 我很乐意为您提供帮助。

User >>> 你知道 huixiangdou 是一款应用吗?

是的, 我知道"茴香豆"是一款应用。然而, 它是一款面向中国市场的手机游戏应用, 由上海汉森信息技术有限公司开发。这款应用提供了丰富的游戏内容, 包括角色扮演、社交互动、实时对战等多种玩法。玩家可以在游戏中打造自己的角色, 探索各种场景, 与其他玩家互动, 参加各种活动, 甚至可以通过游戏中的任务和活动获得实物奖励。不过, 由于我是AI, 我并不能直接体验或操作应用, 但我可以为你提供相关信息和解答你的问题。

到此我们就完成了一个 茴香豆知识助手的服务器端部署 (基础作业) 的全部内容。

【3】 茴香豆进阶（选做）

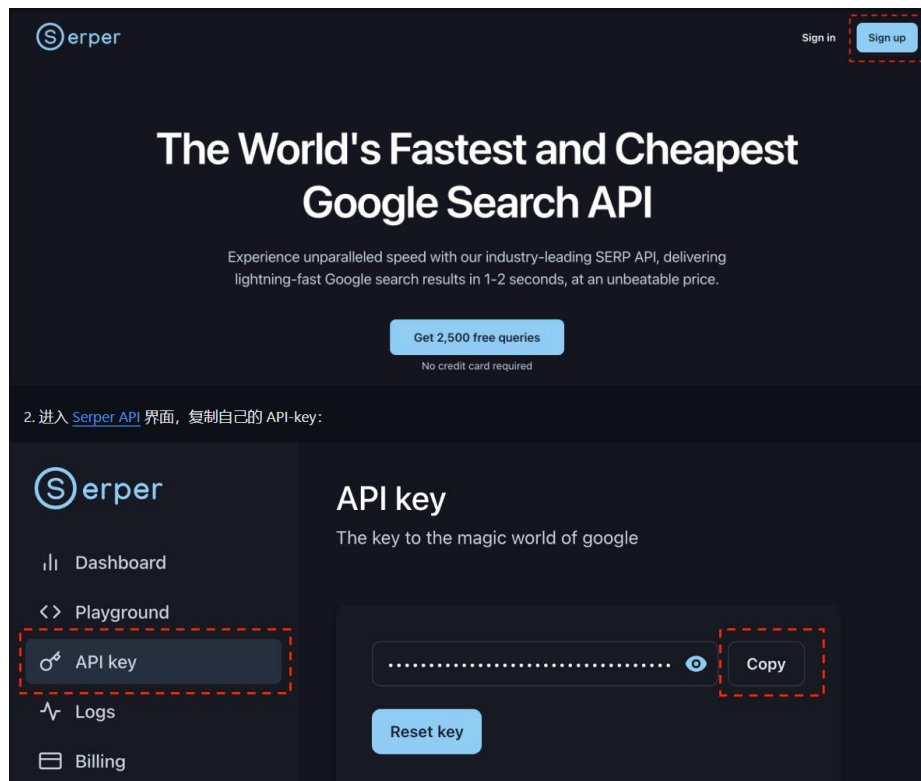
茴香豆并非单纯的 RAG 功能实现，而是一个专门针对群聊优化的知识助手。

3.1 加入网络搜索

茴香豆除了可以从本地向量数据库中检索内容进行回答，也可以加入网络的搜索结果，生成回答。

开启网络搜索功能需要用到 Serper 提供的 API：

登录 Serper ， 注册：



替换 /huixiangdou/config.ini 中的 \${YOUR-API-KEY} 为自己的 API-key:

```
[web_search]
```

```
# check https://serper.dev/api-key to get a free API key
```

```
x_api_key = "${YOUR-API-KEY}"
```

```
domain_partial_order = ["openai.com", "pytorch.org", "readthedocs.io", "nvidia.com",  
"stackoverflow.com", "juejin.cn", "zhuankan.zhihu.com", "www.cnblogs.com"]
```

```
save_dir = "logs/web_search_result"
```

其中 domain_partial_order 可以设置网络搜索的范围。


```
[web_search]
x_api_key = "${YOUR-API-KEY}"
domain_partial_order = ["openai.com", "pytorch.org", "readthedocs.io", "nvidia.com", "stackoverflow.com",
"juejin.cn", "zhuanlan.zhihu.com", "www.cnblogs.com"]
save_dir = "logs/web_search_result"
```

3.2 使用远程模型

茴香豆除了可以使用本地大模型，还可以轻松的调用云端模型 API。

目前，茴香豆已经支持 Kimi，GPT-4，Deepseek 和 GLM 等常见大模型 API。

想要使用远端大模型，首先修改 /huixiangdou/config.ini 文件中。

enable_local = 0 # 关闭本地模型

enable_remote = 1 # 启用云端模型

接着，如下图所示，修改 remote_ 相关配置，填写 API key、模型类型等参数。

```
[llm]
enable_local = 0
enable_remote = 1
# hybrid llm service address
client_url = "http://127.0.0.1:8888/inference"

[llm.server]
# local LLM configuration
# support "internlm/internlm2-chat-7b" and "qwen/qwen-7b-chat-int8"
# support local path, for example
# local_llm_path = "/path/to/your/internlm2"
# also support local_llm_path = "internlm/internlm2-chat-20b"

local_llm_path = "internlm/internlm2-chat-7b"
local_llm_max_text_length = 16000
local_llm_bind_port = 8888

# remote LLM service configuration
# support "gpt", "kimi" and "deepseek"
remote_type = "deepseek"
remote_api_key = "${YOUR-API-KEY}"
# max text length for remote LLM.
# use 128000 for kimi, 192000 for gpt, 16000 for deepseek
remote_llm_max_text_length = 16000
# openai model type.
# use "moonshot-v1-128k" for kimi, "gpt-4" for gpt, "deepseek-chat" for deepseek
remote_llm_model = "deepseek-chat"
```

远端模型配置选项	GPT	Kimi	Deepseek	ChatGLM	xi-api	alles-apin
remote_type	gpt	kimi	deepseek	zhipuai	xi-api	alles-apin
remote_llm_max_text_length 最大值	192000	128000	16000	128000	192000	-
remote_llm_model	"gpt-4-0613"	"moonshot-v1-128k"	"deepseek-chat"	"glm-4"	"gpt-4-0613"	-

启用远程模型可以大大降低 GPU 显存需求，根据测试，采用远程模型的茴香豆应用，最小只需要 2G 内存即可。

需要注意的是，这里启用的远程模型，只用在问答分析和问题生成，依然需要本地嵌入、重排序模型进行特征提取。

3.3 利用 Gradio 搭建网页 Demo

让我们用 Gradio 搭建一个自己的网页对话 Demo，来看看效果。

首先，安装 Gradio 依赖组件：

```
pip install gradio==4.25.0 redis==5.0.3 flask==3.0.2 lark_oapi==1.2.4
```

运行脚本，启动茴香豆对话 Demo 服务：

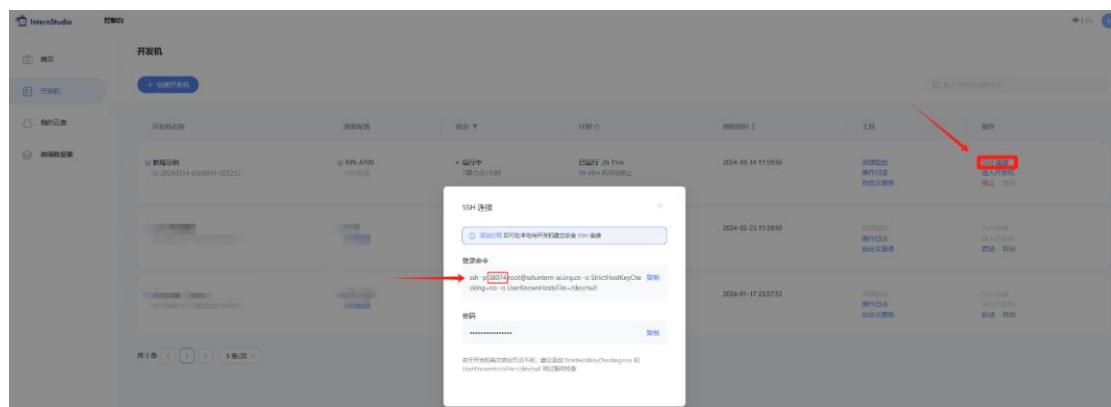
```
cd /root/huixiangdou
```

```
python3 -m tests.test_query_gradio
```

此时服务器端接口已开启。如果在本地服务器使用，直接在浏览器中输入 127.0.0.1:7860，即可进入茴香豆对话 Demo 界面。

针对远程服务器，如我们的 Intern Studio 开发机，我们需要设置端口映射，转发端口到本地浏览器：

(1) 查询开发机端口和密码（图中端口示例为 38374）：



(2) 在本地打开命令行工具：

Windows 使用快捷键组合 Windows + R（Windows 即开始菜单键）打开指令界面，并输入命令 Powershell，按下回车键

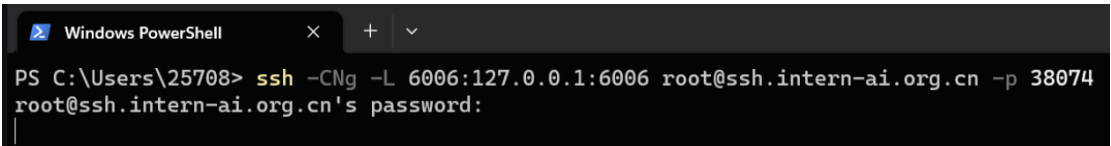
Mac 用户直接找到并打开终端

Ubuntu 用户使用快捷键组合 ctrl + alt + t

在命令行中输入如下命令，命令行会提示输入密码：

ssh -CNg -L 7860:127.0.0.1:7860 root@ssh.intern-ai.org.cn -p <你的端口号>

(1) 复制开发机密码到命令行中，按回车，建立开发机到本地到端口映射。



```
Windows PowerShell
PS C:\Users\25708> ssh -CNg -L 6006:127.0.0.1:6006 root@ssh.intern-ai.org.cn -p 38074
root@ssh.intern-ai.org.cn's password:
```

(4) 在本地浏览器中输入 127.0.0.1:7860 进入 Gradio 对话 Demo 界面，开始对话。



如果需要更换检索的知识领域，只需要用新的语料知识重复步骤 2.2 创建知识库 提取特征到新的向量数据库，更改 huixiangdou/config.ini 文件中 work_dir = "新向量数据库路径";

或者运行：

python3 -m tests.test_query_gradi --work_dir <新向量数据库路径>

无需重新训练或微调模型，就可以轻松的让基础模型学会新领域知识，搭建一个新的问答助手。

3.4 配置文件解析

茴香豆的配置文件位于代码主目录下，采用 Toml 形式，有着丰富的功能，下面将解析配置文件中重要的常用参数。

[feature_store]

...

```
reject_throttle = 0.22742061846268935

...

embedding_model_path = "/root/models/bce-embedding-base_v1"

reranker_model_path = "/root/models/bce-reranker-base_v1"

...

work_dir = "workdir"
```

reject_throttle: 拒答阈值，0-1，数值越大，回答的问题相关性越高。拒答分数在检索过程中通过与示例问题的相似性检索得出，高质量的问题得分高，无关、低质量的问题得分低。只有得分数大于拒答阈值的才会被视为相关问题，用于回答的生成。当闲聊或无关问题较多的环境可以适当调高。 embedding_model_path 和 reranker_model_path: 嵌入和重排用到的模型路径。不设置本地模型路径情况下，默认自动通过 Huggingface 下载。开始自动下载前，需要使用下列命令登录 Huggingface 账户获取权限：

```
huggingface-cli login
```

work_dir: 向量数据库路径。茴香豆安装后，可以通过切换向量数据库路径，来回答不同知识领域的问答。

```
[llm.server]
```

```
...

local_llm_path = "/root/models/internlm2-chat-1_8b"

local_llm_max_text_length = 3000

...
```

local_llm_path: 本地模型文件夹路径或模型名称。现支持 书生·浦语 和 通义千问 模型类型，调用 transformers 的 AutoModels 模块，除了模型路径，输入 Huggingface 上的模型名称，如*"internlm/internlm2-chat-7b"、"qwen/qwen-7b-chat-int8"、"internlm/internlm2-chat-20b"*，也可自动拉取模型文件。 local_llm_max_text_length: 模型可接受最大文本长度。

远端模型支持参考上一小节。

```
[worker]
```

```
# enable search enhancement or not
```



```
enable_sg_search = 0

save_path = "logs/work.txt"

...
```

[worker]: 增强搜索功能，配合 [sg_search] 使用。增强搜索利用知识领域的源文件建立图数据库，当模型判断问题为无关问题或回答失败时，增强搜索功能将利用 LLM 提取的关键词在该图数据库中搜索，并尝试用搜索到的内容重新生成答案。在 config.ini 中查看 [sg_search] 具体配置示例。

```
[worker.time]

start = "00:00:00"

end = "23:59:59"

has_weekday = 1
```

[worker.time]: 可以设置茴香豆每天的工作时间，通过 start 和 end 设定应答的起始和结束时间。

[fronted]: 前端交互设置。

3.5 文件结构

通过了解主要文件的位置和作用，可以更好的理解茴香豆的工作原理。

```
.
├── LICENSE
├── README.md
├── README_zh.md
├── android
├── app.py
├── config-2G.ini
├── config-advanced.ini
├── config-experience.ini
├── config.ini # 配置文件
├── docs # 教学文档
```

- ├── huixiangdou # 存放茴香豆主要代码，重点学习
- ├── huixiangdou-inside.md
- ├── logs
- ├── repodir # 默认存放个人数据库原始文件，用户建立
- ├── requirements-lark-group.txt
- ├── requirements.txt
- ├── resource
- ├── setup.py
- ├── tests # 单元测试
- ├── web # 存放茴香豆 Web 版代码
- └── web.log
- └── workdir # 默认存放茴香豆本地向量数据库，用户建立

./huixiangdou

- ├── __init__.py
- ├── frontend # 存放茴香豆前端与用户端和通讯软件交互代码
 - | ├── __init__.py
 - | ├── lark.py
 - | └── lark_group.py
- ├── main.py # 运行主贷
- ├── service # 存放茴香豆后端 workflow 代码
 - | ├── __init__.py
 - | ├── config.py #
 - | ├── feature_store.py # 数据嵌入、特征提取代码
 - | ├── file_operation.py
 - | ├── helper.py
 - | └── llm_client.py

```

|   |—— llm_server_hybrid.py # 混合模型代码
|   |—— retriever.py # 检索模块代码
|   |—— sg_search.py # 增强搜索，图检索代码
|   |—— web_search.py # 网页搜索代码
|   |—— worker.py # 主流程代码
|   |—— version.py

```

茴香豆 workflow 中用到的 Prompt 位于 `huixiangdou/service/worker.py` 中。可以根据业务需求尝试调整 Prompt，打造你独有的茴香豆知识助手。

...

```

    # Switch languages according to the scenario.

    if self.language == 'zh':

        self.TOPIC_TEMPLATE = '告诉我这句话的主题，直接说主题不要解释：
“{}”

        self.SCORING_QUESTION_TEMPLTE = "{}\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。' # noqa E501

        self.SCORING_RELAVANCE_TEMPLATE = '问题：“{}\n 材料：“{}\n 请仔
细阅读以上内容，判断问题和材料的关联度，用 0~10 表示。判断标准：非常相关得
10 分；完全没关联得 0 分。直接提供得分不要解释。’ # noqa E501

        self.KEYWORDS_TEMPLATE = '谷歌搜索是一个通用搜索引擎，可用于访
问互联网、查询百科知识、了解时事新闻等。搜索参数类型 string，内容是短语或关
键字，以空格分隔。
\n 你现在是{}交流群里的技术助手，用户问“{}”，你打算通过谷歌
搜索查询相关资料，请提供用于搜索的关键字或短语，不要解释直接给出关键字或短
语。' # noqa E501

        self.SECURITY_TEMAPLTE = '判断以下句子是否涉及政治、辱骂、色情、
恐暴、宗教、网络暴力、种族歧视等违禁内容，结果用 0~10 表示，不要解释直接给
出得分。判断标准：涉其中任一问题直接得 10 分；完全不涉及得 0 分。直接给得分
不要解释：“{}” # noqa E501

        self.PERPLESITY_TEMPLATE = "“question:{} answer:{}”\n 阅读以上对话，

```

answer 是否在表达自己不知道，回答越全面得分越少，用 0~10 表示，不要解释直接给出得分。
判断标准：准确回答问题得 0 分；答案详尽得 1 分；知道部分答案但不确定信息得 8 分；知道小部分答案但推荐求助其他人得 9 分；不知道任何答案直接推荐求助别人得 10 分。直接打分不要解释。' # noqa E501

```
self.SUMMARIZE_TEMPLATE = '{ }\n 仔细阅读以上内容，总结得简短有力点' # noqa E501
```

```
# self.GENERATE_TEMPLATE = '材料: "{ }\n 问题: "{ }\n 请仔细阅读参考材料回答问题，材料可能和问题无关。如果材料和问题无关，尝试用你自己的理解来回答问题。如果无法确定答案，直接回答不知道。' # noqa E501
```

```
self.GENERATE_TEMPLATE = '材料: "{ }\n 问题: "{ }\n 请仔细阅读参考材料回答问题。' # noqa E501
```

...