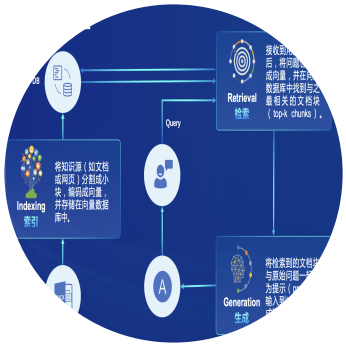
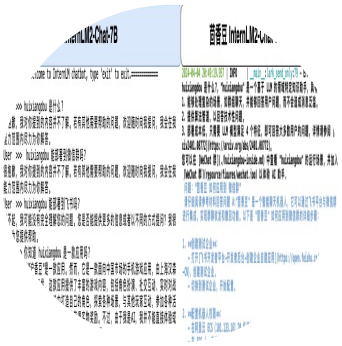


RAG（Retrieval Augmented Generation）技术，通过检索与用户输入相关的信息片段，并结合外部知识库来生成更准确、更丰富的回答。解决 LLMs 在处理知识密集型任务时可能遇到的挑战, 如幻觉、知识过时和缺乏透明、可追溯的推理过程等。提供更准确的回答、降低推理成本、实现外部记忆。

它结合了信息检索和生成模型，用于生成文本和回答查询。以下是RAG技术的基本流程：

1. **信息检索：** 首先，通过信息检索的方式从大量文本中检索相关信息。这可能涉及使用搜索引擎或索引来收集与查询相关的文档或段落。
2. **文本表示：** 将检索到的文本转换成适合生成模型处理的表示形式。通常使用向量化或嵌入技术将文本转换为数值表示。
3. **生成模型：** 使用生成模型来生成对查询的回答或文本。生成模型可以是基于神经网络的模型，如循环神经网络（RNN）、转换器模型（Transformer）等。
4. **整合和生成：** 将信息检索和生成模型结合起来，根据查询和检索到的文本，生成最终的回答或文本。这可能涉及对检索到的文本进行加权或筛选，以及对生成模型生成的文本进行调整或编辑。
5. **评估和反馈：** 对生成的文本进行评估，确保其质量和准确性。根据用户反馈或其他指标，对系统进行调整和改进。

RAG技术的优势在于其能够结合信息检索和生成模型的优点，生成更加准确、连贯和相关的文本回答。这使得RAG技术在问答系统、摘要生成、对话系统等领域具有广泛的应用前景。



如图所示，由于茴香豆是一款比较新的应用， `InternLM2-Chat-7B` 训练数据库中并没有收录到它的相关信息。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。

代码准备：

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

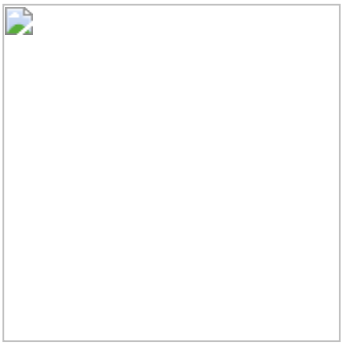
```
conda activate InternLM2_Huixiangdou
```

```
cd /root && mkdir models

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

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

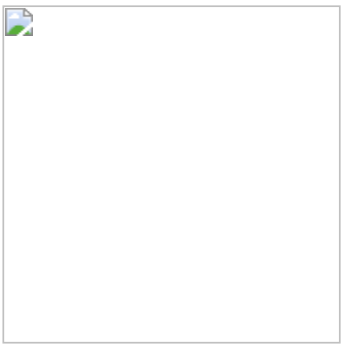
```
cd /root

# 下载 repo

git clone https://github.com/internlm/huixiangdou && cd huixiangdou
git checkout 447c6f7e68a1657fce1c4f7c740ea1700bde0440
```

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

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

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

创建知识库

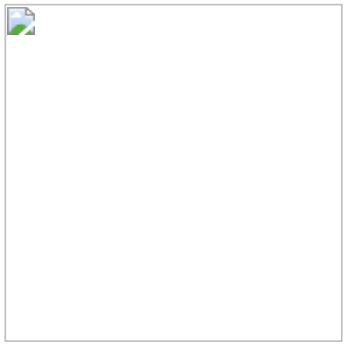


示例代码运行截图

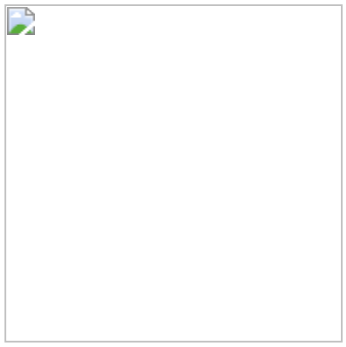


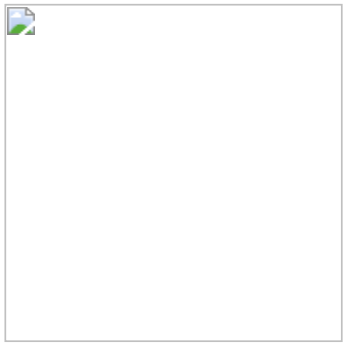
作业

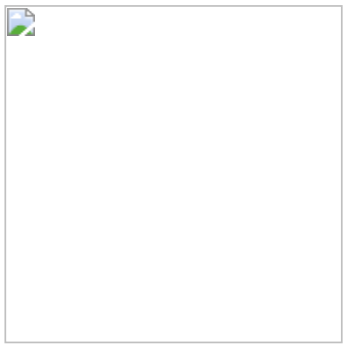
基础作业2：在 InternLM Studio 上部署茴香豆技术助手

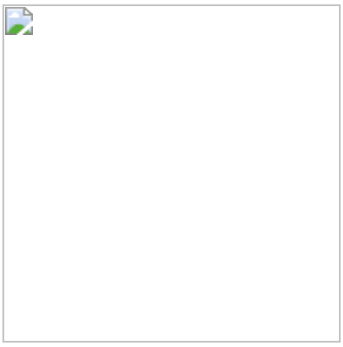


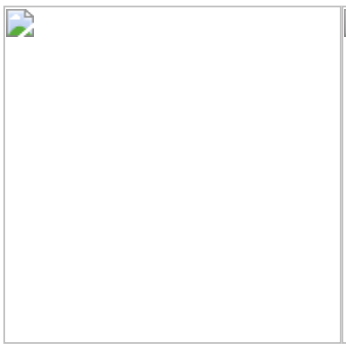
基础作业1： 在茴香豆web版中创建自己领域的知识问答助手

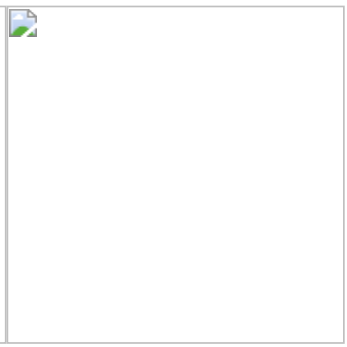












RAG是一种混合了预训练的参数记忆和外部非参数记忆的语言生成模型。以下是实现RAG的步骤：

1. **数据准备**：
    - 准备一个非参数记忆库，例如维基百科文章索引，将其编码为向量表示以进行快速检索。
    - 准备一个预训练的语言模型，例如BART或T5，作为参数记忆组件。
2. **模型构建**：
    - 构建一个神经检索器，例如DPR（Dense Passage Retriever），用于根据输入查询从非参数记忆中检索相关文档。
    - 将预训练的语言模型与检索器相结合，使语言模型能够使用检索到的文档作为上下文生成输出。
3. **训练**：
    - 使用检索器返回的文档作为潜在变量，对整个模型进行端到端训练。
    - 最小化生成的输出与真实输出之间的负对数似然损失。
4. **解码**：
    - 在测试时，对于RAG-Sequence模型，对每个生成的标记运行beam搜索，在所有文档上平均概率来得到最终概率。
    - 对于RAG-Token模型，为每个标记使用不同的文档，直接使用beam搜索解码。
5. **应用**：
    - 可以使用RAG模型来解决各种知识密集型自然语言处理任务，如开放领域问答、事实验证、生成式对话等。
6. **评估**：
    - 使用标准评估指标（如准确率、BLEU分数、ROUGE分数）以及人类评估来评估模型性能。
7. **更新知识库**：
    - RAG的非参数记忆可以轻松地通过替换文档索引来更新，允许模型在没有额外训练的情况下更新其知识库。 通过上述步骤，您可以实现一个RAG模型，并将其应用于各种知识密集型自然语言处理任务。

参考文档:

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.pdf

RAG的主要优势：

1. **随机性带来的新见解**：RAG通过引入随机过程，允许研究人员分析代数几何问题的概率分布和统计特性。这种随机化方法有助于揭示新的代数结构、现象和规律，从而深化我们对代数几何的理解。
2. **概率算法设计**：RAG为代数几何问题的解决提供了新的概率算法，这些算法通常比传统确定性算法更高效。通过随机化，研究人员能够设计出更有效的算法来解决特定类型的代数几何问题，这在实际应用中具有重要意义。
3. **几何与代数的统一**：RAG将几何和代数联系在一起，通过随机化技术探索代数结构的几何性质。这种统一视角有助于发现代数几何中的新联系，促进不同分支之间的交叉研究。
4. **概率建模**：RAG利用概率理论来建立代数几何对象的统计模型。这些模型有助于理解代数几何问题的性质，并可以用于预测和分类新的代数几何对象。
5. **算法优化**：RAG的概率算法设计有助于优化传统代数几何算法的效率。通过随机化技术，研究人员可以找到更快速的解决方案，减少计算时间和资源消耗。
6. **交叉学科影响**：RAG的研究成果不仅影响了代数几何领域，还对其他数学领域产生了深远的影响。例如，它在编码理论、几何数论、计算复杂性等领域都有广泛的应用。 总之，RAG通过随机化技术为代数几何研究提供了新的工具和方法，促进了代数几何领域的进展，并为其他数学领域带来了新的见解和应用。