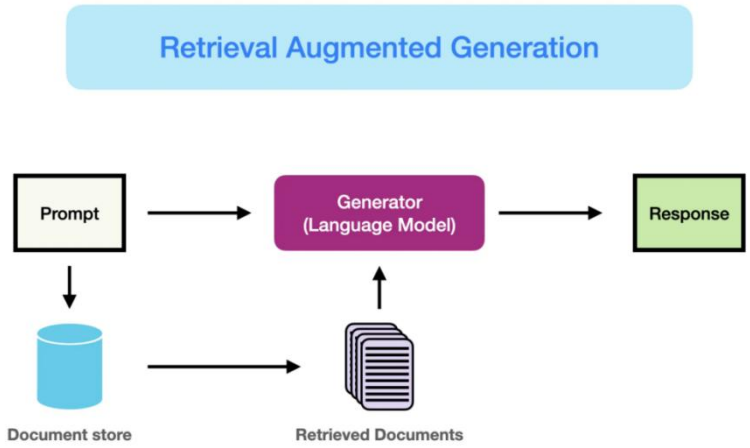
实验 3　RAG 法律检索系统

一、实验背景

RAG（Retrieval-Augmented Generation，检索增强生成） 是一种结合了信息检索技术与语言生成模型的人工智能技术。该技术通过从外部知识库中检索相关信息，并将其作为提示（Prompt）输入给大型语言模型（LLMs），以增强模型处理知识密集型任务的能力，如问答、文本摘要、内容生成等。RAG 模型由 Facebook AI Research（FAIR）团队于 2020 年首次提出，并迅速成为大模型应用中的热门方案。



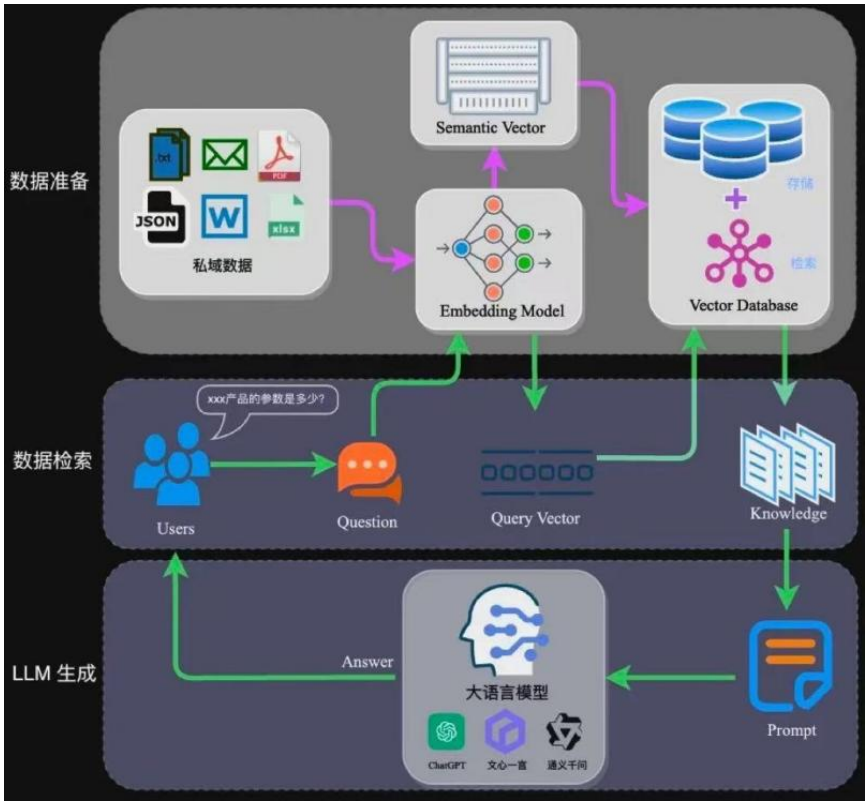
在本次实验中，我们要求各位同学利用公开法律相关知识数据库，并基于 langchain 开发框架，实现一种简单的 RAG 问答应用示例。本次实验的主要目的是比较大模型的生成式检索与普通检索的区别，以及引入 RAG 之后大模型在专业搜索上是否做得更好。

二、实验介绍

（一）RAG 介绍

RAG[1]（Retrieval-Augmented Generation）是一种人工智能技术，它通过整合外部知识库来增强大型语言模型（LLMs）的能力。RAG 模型首先使用信息检索技术从外部数据源中提取相关信息，然后利用这些信息增强大型语言模型的提示，从而提高回答的准确性和相关性。

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/



RAG 的架构主要分为三个阶段：索引（Indexing）、检索（Retrieval）和生成（Generation）。在索引阶段，系统会将外部数据源中的内容转换为向量表示，并存储在向量数据库中。检索阶段涉及使用用户的查询来搜索向量数据库，以找到最相关的信息。最后，在生成阶段，系统将检索到的信息与用户的查询结合起来，通过语言模型生成最终的回答。

以下是构建 RAG 检索系统的一般流程：

1. **数据准备**
   - ✧ 数据来源：法律检索系统需要高质量的法律文本数据，如案例库、法律法规、判例等。**（law_data.csv）**
   - ✧ 数据清洗：移除无关信息、规范化文本格式。
   - ✧ 文本分割：将法律文献分割成段落、章节或案例，以便于后续检索和匹配。
   - ✧ 嵌入表示：将文档转换为向量表示，创建向量数据库。
2. **数据检索**
   - ✧ 查询处理：用户输入查询时，将查询转化为向量，并从检索数据库中找到最相关的文档。
   - ✧ 数据检索方法：相似性检索、全文检索等，根据检索效果，一般可以选择多种检索方式融合，提升召回率。
     a. 相似性检索：计算查询向量与所有存储向量的相似性得分，返回得分高的记录。常见的相似度匹配方法包括：**TF-IDF**，计算输入和知

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/

识库中文档之间的词频-逆文档频率（TF-IDF）值来衡量相似度，适用于短文本检索任务。**BM25**，基于概率模型的一种优化版本的 TF-IDF 算法，能够更好地处理长文本，并具有更高的检索精度。**深度学习方法，**如基于 BERT 等预训练语言模型的相似度计算方法，能够更好地捕捉输入与知识库中信息的语义相似度，尤其在处理复杂语义时表现优越。

b. 全文检索：全文检索是一种比较经典的检索方式，在数据存入时，通过关键词构建倒排索引。在检索时，通过关键词进行全文检索，找到对应的记录。

3. LLM 生成

将检索到的文档与用户的查询结合，作为 RAG 的输入，生成符合用户需求的法律答案。

（二）基于 LangChain 实现 RAG 系统

LangChain[2] 是一个用于开发由大语言模型 (LLM) 提供支持的应用程序的框架。

Embeddings 类是一个专为与文本嵌入模型交互而设计的类。有许多不同的嵌入模型提供商（OpenAI、Cohere、Hugging Face 等）和本地模型，旨在为所有这些模型提供标准接口。LangChain 中的基类 Embeddings 提供两种方法：一种用于嵌入文档，另一种用于嵌入查询。前者将多个文本作为输入，而后者将单个文本作为输入。将它们作为两种单独方法的原因是，某些嵌入提供商对文档（要搜索的文档）和查询（搜索查询本身）具有不同的嵌入方法。

向量数据库检索器是一种使用向量数据库来检索文档的检索器。它是在向量数据库类周围的一个轻量级包装器，使其符合检索器接口。它使用向量数据库中实现的搜索方法，如相似性搜索和 MMR，来查询向量数据库中的文本。

```
retriever = vectorstore.as_retriever(
    search_type="similarity_score_threshold",
search_kwargs={"score_threshold": 0.5, "k": 1}
)
```

三、实验内容：

（一）数据集说明

law_data.csv（数据集已在压缩包中给出）：

法律知识库，由两部分组成：中华人民共和国法律手册最核心的~600 条法律条文、百度知道~2400 条法律问答数据。

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/

	data
0	民法商法-农民专业合作社法2017-12-27：第十五条 农民专业合作社章程应当载明下列事项...
1	民法商法-个人独资企业法1999-08-30：第十七条 个人独资企业投资人对本企业的财产依法...
2	民法商法-个人独资企业法1999-08-30：第二十六条 个人独资企业有下列情形之一时，应当...
3	民法商法-个人独资企业法1999-08-30：第二十八条 个人独资企业解散后，原投资人对个人...
4	民法商法-公司法2018-10-26：第七十五条 自然人股东死亡后，其合法继承人可以继承股东...
...	...
3008	请问，我借了七千块给我朋友，是在微信转账的，我写了张借款单，单不是她本人写的，借条上面没有她...
3009	别人用我老公的身份证去办贷款，我老公本人也去了，也签字了，这样的后果严重吗？还不起钱，我老公...
3010	陷入网贷的高利贷\n陷入网贷的高利贷想死的心都有了，陷入了网贷的高利贷中，无法自拔\n本金以...
3011	不经过主人同意拆掉主机主和拿走手提触犯了什么法律？还借两三千钱不还\n不经过主人同意拆掉主机...
3012	离婚判决患精神病的女儿给了女方，但女方生病了，无法照顾女儿，男方有责任照顾女儿吗？\n男方当...

3013 rows × 1 columns

（二）任务说明

[1]【必做】数据准备阶段：主要是将私域数据向量化后构建索引并存入数据库的过程，主要包括：数据提取、文本分割、向量化、数据入库等环节。

- 数据提取：本次实验我们提供的数据是 csv 文件，因此可以使用 langchain.document_loaders 中的 CSVLoader 来加载我们的数据。（选做：可适当进行一些数据清洗工作）
- 文本分割：langchain.text_splitter 中的 CharacterTextSplitter 可以按照指定的字符（如换行符）直接分割文本。它适用于那些结构简单且以特定字符明确分隔的文本，如 CSV 文件、日志文件等。因此本次实验可使用 CharacterTextSplitter 对本次实验提供的数据进行文本分割（注意分割符与字符长度的选取）。
- 向量化：向量化是一个将文本数据转化为向量矩阵的过程，该过程会直接影响到后续检索的效果。在这里为大家提供了两种常见 embedding 模型的获取地址：https://huggingface.co/moka-ai/m3e-base、https:///huggingface.co/BAAI/bge-base-en-v1.5，大家可以先部署到本地再使用 langchain.embeddings 的 HuggingFaceBgeEmbeddings 来进行调用（选择一种向量模型即可）。
- 数据入库：数据向量化后构建索引，并写入数据库的过程可以概述为数

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/

据入库过程，适用于 RAG 场景的数据库包括：FAISS、Chromadb、ES、milvus 等。langchain 的官方网站（https://python.langchain.com/v0.1/docs/modules/data_connection/vectorstores/）提供了 FAISS 与 Chromadb 的使用代码示例大家可以参考。

[2]【必做】数据检索阶段：我们根据用户的提问，通过高效的检索方法，召回与提问最相关的知识，并融入 Prompt，才能使得大模型参考当前提问和相关知识，生成相应的答案。常见的数据检索方法包括：相似性检索、全文检索等。本次实验中，将向量入库后，我们可以直接调用 db.similarity_search 语句进行问题的相似性检索，可将得到结果与后续大模型生成结果进行对比。（选做：可探索其他检索方式，并与最终大模型生成结果进行对比）

[3]【必做】LLM 生成阶段：将检索得到的相关知识注入 prompt，大模型参考当前提问和相关知识，生成相应的答案。

◆ Prompt：作为大模型的直接输入，Prompt 是影响模型输出准确率的关键因素之一。在 RAG 场景中，Prompt 一般包括任务描述、背景知识（检索得到）、任务指令（一般是用户提问）等，根据任务场景和大模型性能，也可以在 Prompt 中适当加入其他指令优化大模型的输出。

我们提供了本次实验的基础参考 prompt 模板，（选做：大家可以进行适当优化，观察其对模型回答结果的影响）：

template = """你是专业的法律知识问答助手。你需要使用以下检索到的上下文片段来回答问题，禁止根据常识和已知信息回答问题。如果你不知道答案，直接回答“未找到相关答案”。

Question: {question}

Context: {context}

Answer:

"""

接下来，可以使用 langchain.prompts 中的 ChatPromptTemplate 生成 prompt。

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/

◆ LLM：可供选择的大模型有很多，大家选择一种完成实验即可。该网址（https://python.langchain.com/docs/integrations/chat/）提供了 langchain 支持的 llms 以及调用方式供大家参考，下面是以 Qwen 作为示例的简单实现：

- ■ 获取 API Key：https://help.aliyun.com/document_detail/611472.html?spm=a2c4g.2399481.0.0
- ■ 配置 API Key：

```
from getpass import getpass
import os
DASHSCOPE_API_KEY = getpass()
os.environ["DASHSCOPE_API_KEY"] = DASHSCOPE_API_KEY
```

- ■ 调用

```
from langchain_community.llms import Tongyi
llm = Tongyi()
```

- ■ 参考当前提问和检索生成回答

```
from langchain.schema.runnable import RunnablePassthrough
from langchain.schema.output_parser import StrOutputParser
retriever = db.as_retriever()
rag_chain = (
    {"context": retriever,  "question": RunnablePassthrough()}
    | prompt
    | llm
    | StrOutputParser()
)
question = “要提问的问题”
rag_chain.invoke(question)
```

[4]【需要实现的 question】（**6 选 3 即可**，当然也可以大于 3）

① 借款人去世，继承人是否应履行偿还义务？

② 如何通过法律手段应对民间借贷纠纷？

③ 没有赡养老人就无法继承财产吗？

④ 谁可以申请撤销监护人的监护资格？

⑤ 你现在是一个精通中国法律的法官，请对以下案件做出分析：经审理查明：被告人 xxx 于 2017 年 12 月，多次在本市 xxx 盗窃财物。具体事实

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/

如下：（一）2017 年 12 月 9 日 15 时许，被告人 xxx 在 xxx 店内，盗窃白色毛衣一件（价值人民币 259 元）。现赃物已起获并发还。（二）2017 年 12 月 9 日 16 时许，被告人 xx 在本市 xxx 店内，盗窃米白色大衣一件（价值人民币 1199 元）。现赃物已起获并发还。（三）2017 年 12 月 11 日 19 时许，被告人 xxx 在本市 xxx 内，盗窃耳机、手套、化妆镜等商品共八件（共计价值人民币 357.3 元）。现赃物已起获并发还。（四）2017 年 12 月 11 日 20 时许，被告人 xx 在本市 xxxx 内，盗窃橙汁、牛肉干等商品共四件（共计价值人民币 58.39 元）。现赃物已起获并发还。2017 年 12 月 11 日，被告人 xx 被公安机关抓获，其到案后如实供述了上述犯罪事实。经鉴定，被告人 xxx 被诊断为精神分裂症，限制刑事责任能力，有受审能力。

⑥ 你现在是一个精通中国法律的法官，请对以下案件做出分析：2012 年 5 月 1 日，原告 xxx 在被告 xxxx 购买“玉兔牌”香肠 15 包，其中价值 55 8.6 元的 14 包香肠已过保质期。xxx 到收银台结账后，即径直到服务台索赔，后因协商未果诉至法院，要求 xxxx 店支付 14 包香肠售价十倍的赔偿金 5586 元。

[5]【选做】可以自己尝试编写 CoT Prompt 看看能不能引导大模型做得更好。

[6]【选做】我们提供的是一些基础实现，大家可以根据自身掌握与调研情况进行适当优化与改进。

❖ **请注意：实验报告中需要包含必要的实验过程和代码展示（比如检索方法、模型选择或者选做实现的关键代码）和至少 3 个 question 的输出展示，对比 RAG 与普通检索的区别，以及引入 RAG 前后大模型在专业搜索上的区别。**

四、实验要求

本次实验要求分组完成，每组最多 3 人（可以少于 3 人，但无优惠政策）。

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/

附：提交说明

请于截止日期（**2025 年 1 月 16 日晚 23:59**）前将**实验报告（pdf）**提交到课程邮箱 ustcweb2022@163.com（不需要提交代码与模型文件），具体要求如下：

1. 邮件标题以及 pdf 文件命名为"组长学号-组长姓名-实验 3"格式。邮件正文中请列出小组所有成员的姓名、学号。
2. 因未署名造成统计遗漏责任自行承担，你可以将邮件抄送你的队友。
3. 实验报告请务必独立完成，如果发现抄袭按 0 分处理。
4. 迟交实验将不被接收。

[1] https://mo-xiaoxi.github.io/2024/05/26/RAG/
[2] https://python.langchain.ac.cn/docs/how_to/