week 02 -创研课技术栈概览+工具介绍

概要

在本次"AI+"创研课中，我们的项目围绕国家治理大数据和人工智能创新平台上的数据开展研究，在我们的研究方案里，我们将完成以下任务：

- [ ] 在平台数据上以适应大语言模型的数据流和数据粒度构建多模态数据库。
- [ ] 应用RAG+LLM，在数据库基础上实现人工智能问答系统。
- [ ] 完成下游任务，对质性资料进行数据挖掘和数据分析。
- [ ] 利用情感分析技术，构建详尽的被访者人物画像。
- [ ] 依托数据库，训练以质性资料为基础，新闻学和访谈内容特化的大语言模型（可选）。

看上去很复杂，有一定难度？不要担心，**车到山前必有路**。不要产生畏难心理，这些技术我们可以从0开始，只要你掌握了基本的原理，能看懂一些代码，就可以很快上手相关工具。接下来介绍我们会用到的技术。

工具不是目的，工具为科研服务。

技术栈概览

👉数据库技术

我们有一个丰富且多模态的数据集，里面包含了丰富的信息。为了开发并利用好这些数据，我们首先要做的是将他们整理好并存储在一个容易存取的地方，这个地方就是数据库。类比取快递的例子，你买了一箱衣服，快递是分件的，我们要把快递撕开，将衣服洗一遍，然后放到衣柜里。怎么做呢？

1. 了解数据库

为理解什么是数据库，请参考这篇文章。

> 基础篇：数据库 SQL 入门教程
>
> 
>

如果你有其他程序语言基础，在快速通览一遍后应该会发现：SQL语言相当简单。当然只是看一遍无法上手，推荐你在数据库上应用一下SQL语言（参见下文工具部分-数据库开发）。

2. SQL语言

SQL - Structured Query Language - 结构化查询语言：是一种用于管理和操作关系型数据库的标准化编程语言。在本次创研课后，你将学到如何使用 SQL 访问和处理数据系统中的数据，这类数据库包括：MySQL、SQL Server、Access、Oracle、Sybase、DB2 等等。

👉数据清洗-*甚至是多模态的...*

在我们把数据存放入数据库前，应当先进行数据清洗（在数据库中也可以实现数据清洗）。为什么要进行数据清洗？数据清洗是数据处理和数据分析中一个非常重要的步骤，它可以帮助我们提高数据的质量，从而提高数据分析和机器学习的准确性和可靠性。

1. Multimodal - 多模态（作为了解）

- **多模态**：指的是在交互或信息处理过程中，同时利用多种感官或数据类型（如文本、图像、音频、视频等）的方式。
- **多模态技术**：在人工智能领域，特指将来自不同模态的数据和信息进行融合，以提高系统的理解和生成能力。

2. 清洗方法

根据选题指南，我们可能会遇到这些格式的数据：

- [ ] 访谈音频
- [ ] 视频
- [ ] 图片
- [ ] 文本记录
- [ ] 观察日志

尽管处理工具不同，数据模态也不同，但我们清洗的思路是大致相同的：

> - **初步检查**
>
> 检查文件是否完整、无损坏，并确认其格式是否符合后续处理的要求。
> - **缺失值处理**
>
> 如果数据某一部分缺失，这个数据要还是不要？
> - **异常值检测**
>
> 异常值可能是由于数据录入错误、测量误差或真实但极端的观测值。
> - **ETC**.....
>
> 详见：数据清洗（Data Cleansing）

数据清洗是一个迭代的过程，可能需要多次重复上述步骤，直到数据集达到可接受的质量标准。在实际操作中，数据清洗可能涉及复杂的逻辑和算法，需要具体情况具体分析。

👉Python

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。

要进行数据清理，链接数据库，编写RAG，调用开源大模型，完成下游任务.....这些都离不开Python。本项目对于Python掌握程度的要求如下：

- [ ] ✔能看得懂网上的基础Python代码，知道它在做些什么。不需要运行它，你可以**借助大模型**理解代码。（以看懂这篇文章为例：MultiModal RAG for Advanced Video Processing with LlamaIndex & LanceDB，*如果打不开请尝试搭梯子*）。
- [ ] ✔能**利用大模型**写基础Python代码。给出你的需求，读懂，然后把它有效的部分保留下来。
- [ ] ✔会调试代码，找不到bug也没关系。以PyCharm为例，会使用调试工具，能打断点，会看进程中的参数，就可以了（参见下文工具部分-配置Python环境）！

👉LLM - Large Language Model - 大语言模型

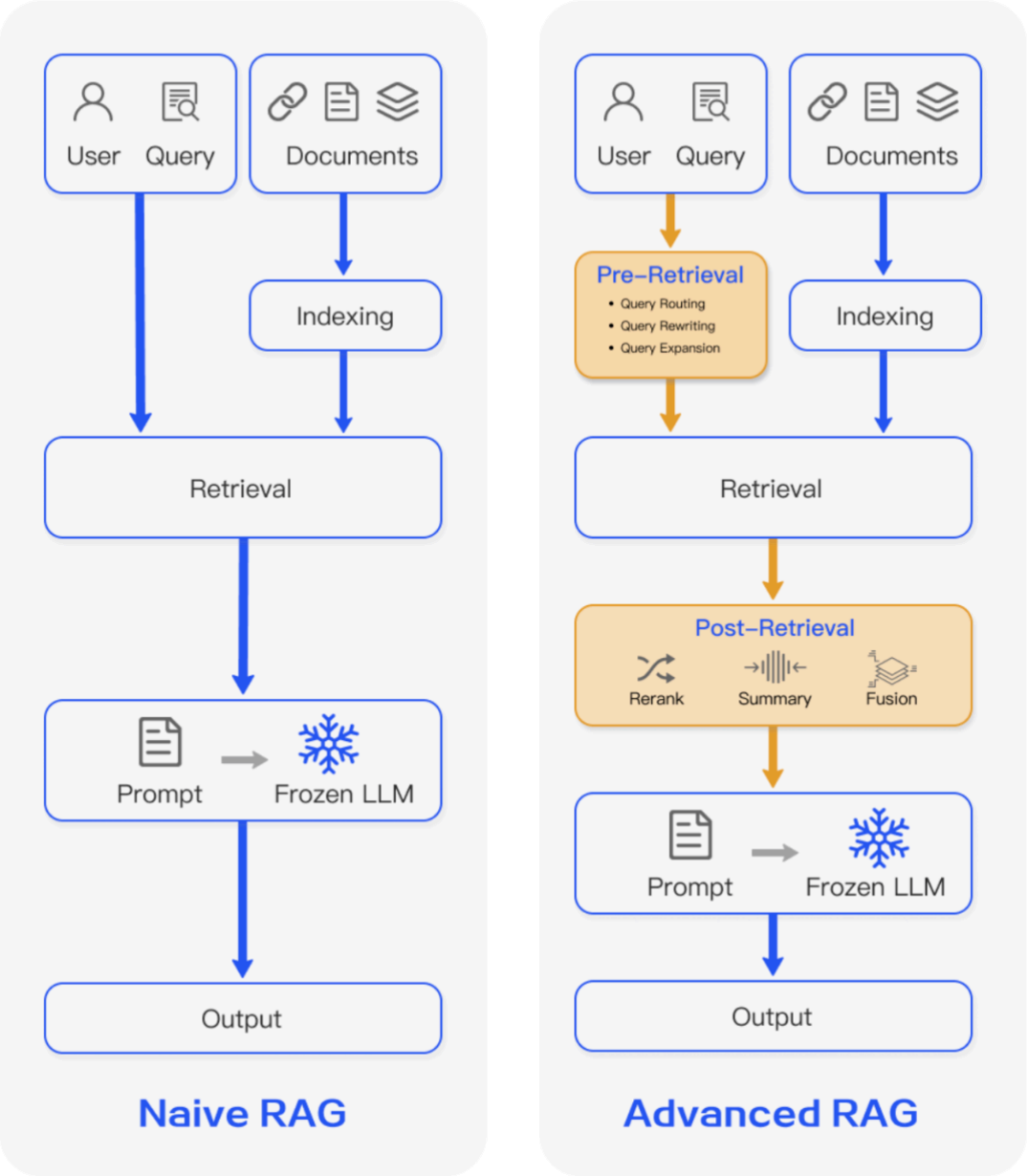
根据课程设计，我们不需要深入理解大语言模型架构、设计和优化环节。我们只需要会调包并使用应该就足够了（如果我们没有将可选任务考虑在内的话）。大语言模型技术的确丰富且需要一定基础，但目前市面上的开源大模型和接口都已经封装得很好了，我们要做的就是把他们利用起来。

> 如果你对LLM感兴趣，可以自行探索这篇文章：
>
> 一文读懂“大语言模型”

👉RAG - Retrieval-Augmented Generation -检索增强生成

RAG通过语义相似性计算从外部知识库中检索相关文档块来增强 LLM。通过引用外部知识，RAG 有效地减少了生成事实不正确的内容的问题。它与LLM的集成导致了广泛采用，将RAG确立为推进聊天机器人和增强LLM对实际应用的适用性的关键技术。

如果你用过大模型问问题，也许会有一个朴素的想法：我想根据一个文档的内容来提问，如何呢？就是用到RAG。



> 基础了解请参阅：[读懂RAG这一篇就够了，万字详述RAG的5步流程和12个优化策略]
>
> 深入了解综述请参阅：[Retrieval-Augmented Generation for Large Language Models: A Survey]

👉情感分析

情感分析，也称为意见挖掘，是自然语言处理（NLP）领域中的一个重要分支。它涉及识别和提取文本数据中表达的情感信息。情感分析通常用于理解人们对某个主题、产品或服务的态度，从而提供有价值的洞察，帮助企业和组织做出更明智的决策。
根据课程设计，我们可能会采用指定方法实现，故不在此赘述。

使用工具

要*卓越地*完成本次系列任务，你可能会需要使用以下工具：

💪数据库开发

- **MySQL+Navicat**

  我们可能会使用国家治理大数据和人工智能创新平台上的云上数据库，但是**出于学习目的**，在此介绍MySQL+Navicat，用来学习SQL语言。

安装请参考：

首先安装MySQL 【一篇搞定】MySQL安装与配置

然后安装Navicat 2024最新 Navicat Premium 17.0.8 简体中文版破解激活永久图文详细教程（亲测可用）

最后，链接Navicat到MySQL。应该很简单，在Navicat的图形化界面就可以看找到链接，输入密码即可~

- **使用SQL简单测试**

还是那个方案：利用好大模型。

- [ ] Q：用SQL创建一个示例表

> 示例：
>
> ```
> CREATE TABLE Users (
>  UserID INT PRIMARY KEY AUTO_INCREMENT, -- 假设这是一个自增的主键
>  UserName VARCHAR(255) NOT NULL, -- 用户名，不允许为空
>  Email VARCHAR(255) UNIQUE NOT NULL, -- 电子邮件地址，唯一且不允许为空
>  Age INT, -- 年龄，可以为空
>  RegisterDate DATETIME DEFAULT CURRENT_TIMESTAMP -- 注册日期，默认为当前时间戳
> );
> ```

将这个代码测试一下。

> →还不太会用Navicat？看看这篇文章：Navicat使用快速入门教程

看到这里，你应该学会了如何查找资料：CSDN、大模型、各个技术平台.....

因此，接下来的内容仅仅作为索引，指示你需要哪一些配置。本周日-2024/09/20-我们开展一次线下讨论，讲解本章内容及后续要求。

💪配置Python环境

- PyCharm
- Anaconda Navigator
- Jupyter Notebook

💪管理文档

- Typora
- Git
- 阅读文献工具（自选）
- 写论文工具（可选）