RAG实战

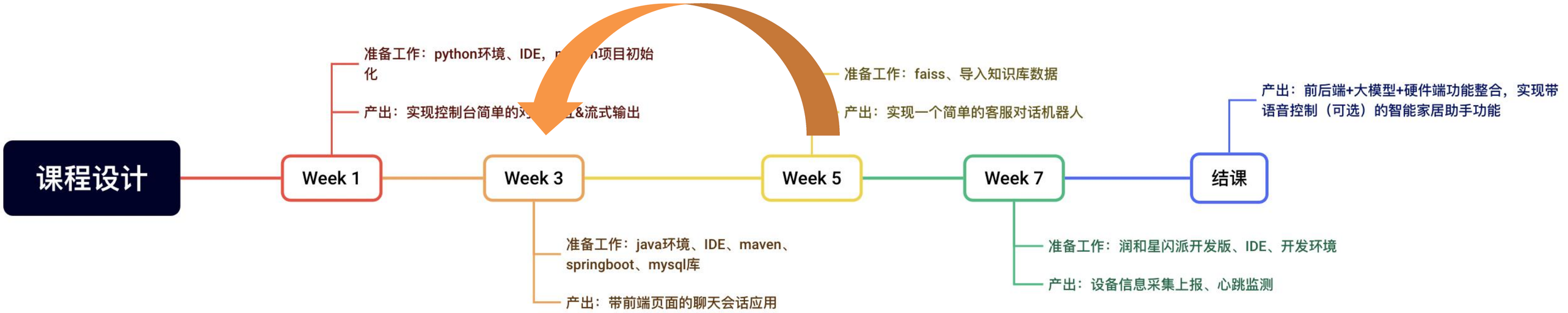
Project1 Review

```python
def _get_default_parameters(self) -> Dict:
    params: Dict[Any, Any] = {}
    if self.max_tokens is not None:
        params["max_tokens"] = self.max_tokens
    params["incremental_output"] = self.incremental_output
    params["enable_search"] = self.enable_search
    if self.stop is not None:
        params["stop"] = self.stop
    if self.temperature is not None:
        params["temperature"] = self.temperature

    if self.top_k is not None:
        params["top_k"] = self.top_k

    if self.top_p is not None:
        params["top_p"] = self.top_p
    if self.seed is not None:
        params["seed"] = self.seed
```

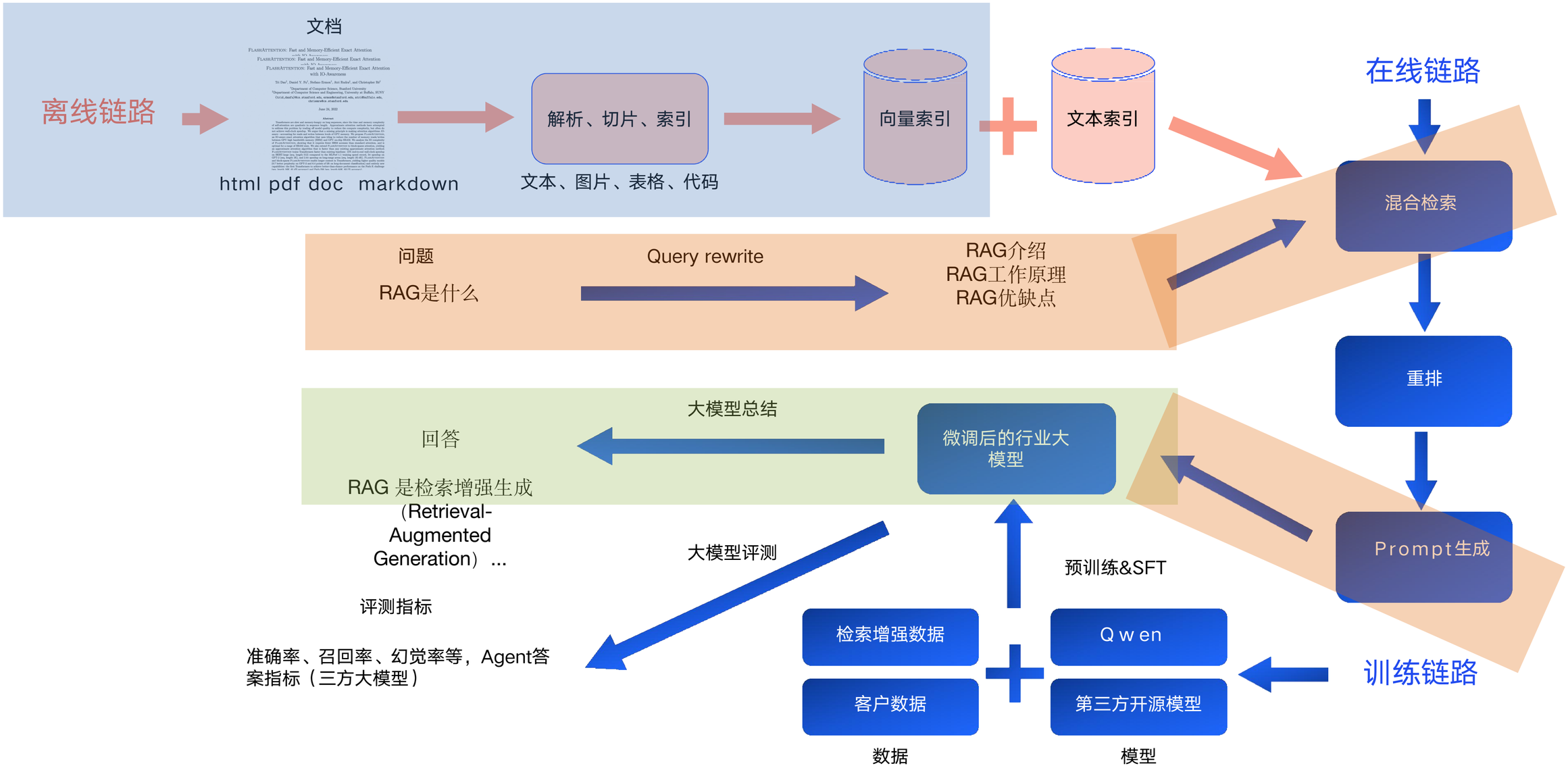
https://help.aliyun.com/zh/model-studio/developer-reference/use-qwen-by-calling-api

提示工程介绍：https://www.promptingguide.ai/zh/introduction/settings

课程要求-project



工程链路

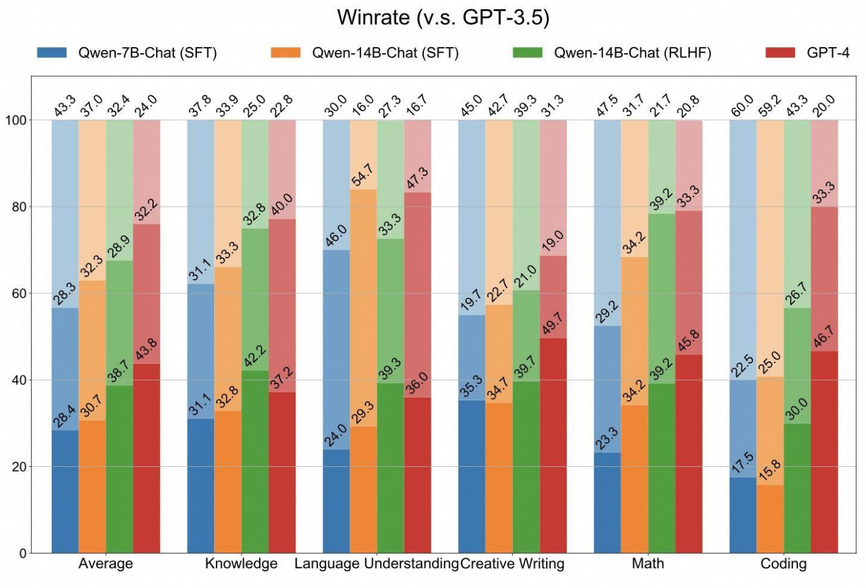

离线链路
文档
html pdf doc markdown
解析、切片、索引
文本、图片、表格、代码
向量索引
文本索引
在线链路
混合检索
重排
Prompt生成
问题
RAG是什么
Query rewrite
RAG介绍
RAG工作原理
RAG优缺点
大模型总结
回答
RAG 是检索增强生成（Retrieval-Augmented Generation）...
微调后的行业大模型
大模型评测
评测指标
准确率、召回率、幻觉率等，Agent答案指标（三方大模型）
预训练&SFT
检索增强数据
客户数据
数据
Qwen
第三方开源模型
模型
训练链路

详细技术方案-离线链路

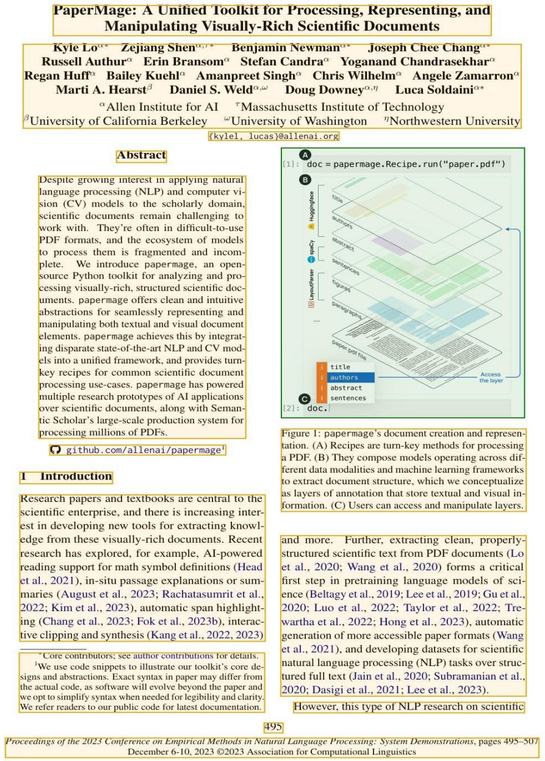
待解析的文件

	GPT-4 Evaluated few-shot	GPT-3.5 Evaluated few-shot	LM SOTA Best external LM evaluated few-shot	SOTA Best external model (incl. benchmark-specific tuning)
MMLU [49] Multiple-choice questions in 57 subjects (professional & academic)	**86.4%** 5-shot	70.0% 5-shot	70.7% 5-shot U-PaLM [50]	75.2% 5-shot Flan-PaLM [51]
HellaSwag [52] Commonsense reasoning around everyday events	**95.3%** 10-shot	85.5% 10-shot	84.2% LLaMA (validation set) [28]	85.6 ALUM [53]
AI2 Reasoning Challenge (ARC) [54] Grade-school multiple choice science questions. Challenge-set.	**96.3%** 25-shot	85.2% 25-shot	85.2% 8-shot PaLM [55]	86.5% ST-MOE [18]
WinoGrande [56] Commonsense reasoning around pronoun resolution	**87.5%** 5-shot	81.6% 5-shot	85.1% 5-shot PaLM [3]	85.1% 5-shot PaLM [3]
HumanEval [43] Python coding tasks	**67.0%** 0-shot	48.1% 0-shot	26.2% 0-shot PaLM [3]	65.8% CodeT + GPT-3.5 [57]
DROP [58] (F1 score) Reading comprehension & arithmetic.	80.9 3-shot	64.1 3-shot	70.8 1-shot PaLM [3]	**88.4** QDGAT [59]
GSM-8K [60] Grade-school mathematics questions	**92.0%*** 5-shot chain-of-thought	57.1% 5-shot	58.8% 8-shot Minerva [61]	87.3% Chinchilla + SFT+ORM-RL, ORM reranking [62]

表格信息提取



图表理解



PaperMage: A Unified Toolkit for Processing, Representing, and Manipulating Visually-Rich Scientific Documents

Kyle Lo[α*] Zejiang Shen[φ,τ*] Benjamin Newman[γ*] Joseph Chee Chang[α*] Russell Authur[α] Erin Bransom[α] Stefan Candra[α] Yoganand Chandrasekhar[α] Regan Huff[α] Bailey Kuehl[α] Amanpreet Singh[α] Chris Wilhelm[α] Angele Zamarron[α] Marti A. Hearst[β] Daniel S. Weld[α,ω] Doug Downey[α,η] Luca Soldaini[α*]

[α]Allen Institute for AI [τ]Massachusetts Institute of Technology [β]University of California Berkeley [ω]University of Washington [η]Northwestern University

{kylel, lucas}@allenai.org

Abstract

Despite growing interest in applying natural language processing (NLP) and computer vision (CV) models to the scholarly domain, scientific documents remain challenging to work with. They're often in difficult-to-use PDF formats, and the ecosystem of models to process them is fragmented and incomplete. We introduce papermage, an open-source Python toolkit for analyzing and processing visually-rich, structured scientific documents. papermage offers clean and intuitive abstractions for seamlessly representing and manipulating both textual and visual document elements. papermage achieves this by integrating disparate state-of-the-art NLP and CV models into a unified framework, and provides turn-key recipes for common scientific document processing use-cases. papermage has powered multiple research prototypes of AI applications over scientific documents, along with Semantic Scholar's large-scale production system for processing millions of PDFs.

github.com/allenai/papermage[1]

1 Introduction

Research papers and textbooks are central to the scientific enterprise, and there is increasing interest in developing new tools for extracting knowledge from these visually-rich documents. Recent research has explored, for example, AI-powered reading support for math symbol definitions (Head et al., 2021), in-situ passage explanations or summaries (August et al., 2023; Rachatasumrit et al., 2022; Kim et al., 2023), automatic span highlighting (Chang et al., 2023; Fok et al., 2023b), interactive clipping and synthesis (Kang et al., 2022, 2023) and more. Further, extracting clean, properly-structured scientific text from PDF documents (Lo et al., 2020; Wang et al., 2020) forms a critical first step in pretraining language models of science (Beltagy et al., 2019; Lee et al., 2019; Gu et al., 2020; Luo et al., 2022; Taylor et al., 2022; Trewartha et al., 2022; Hong et al., 2023), automatic generation of more accessible paper formats (Wang et al., 2021), and developing datasets for scientific natural language processing (NLP) tasks over structured full text (Jain et al., 2020; Subramanian et al., 2020; Dasigi et al., 2021; Lee et al., 2023).

However, this type of NLP research on scientific

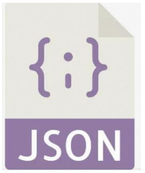
*Core contributors; see author contributions for details.
[1]We use code snippets to illustrate our toolkit's core designs and abstractions. Exact syntax in paper may differ from the actual code, as software will evolve beyond the paper and we opt to simplify syntax when needed for legibility and clarity. We refer readers to our public code for latest documentation.

Figure 1: papermage's document creation and representation. (A) Recipes are turn-key methods for processing a PDF. (B) They compose models operating across different data modalities and machine learning frameworks to extract document structure, which we conceptualize as layers of annotation that store textual and visual information. (C) Users can access and manipulate layers.

495

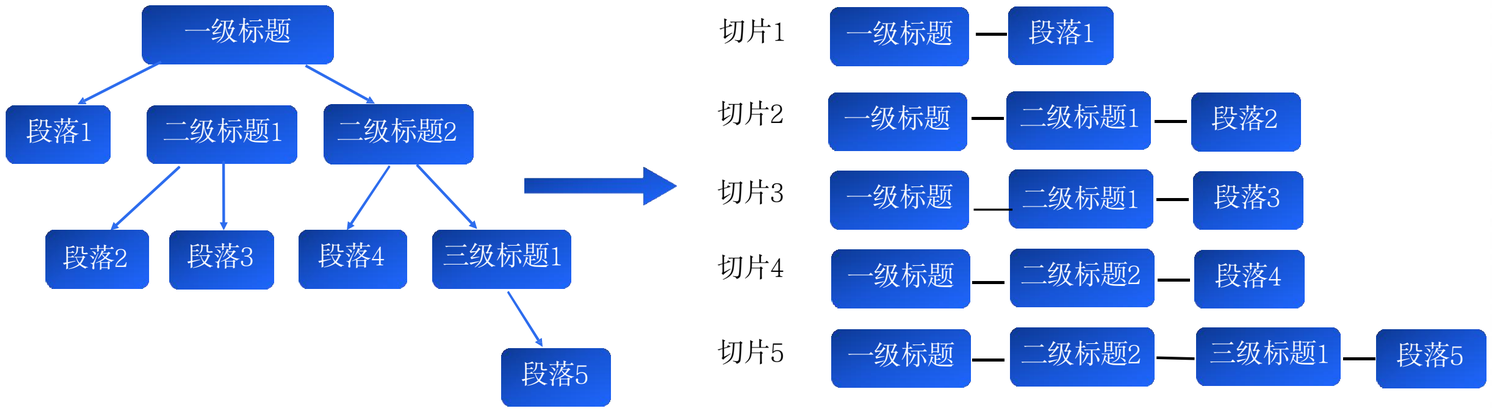
Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 495–507
December 6-10, 2023 ©2023 Association for Computational Linguistics

文档结构分析

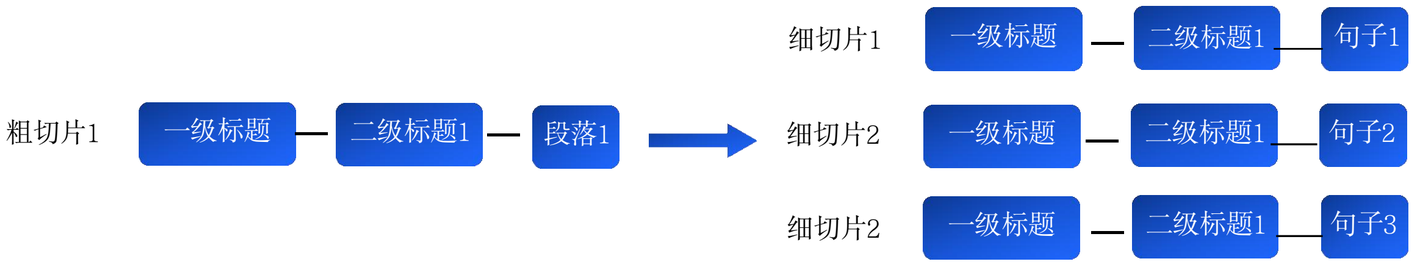


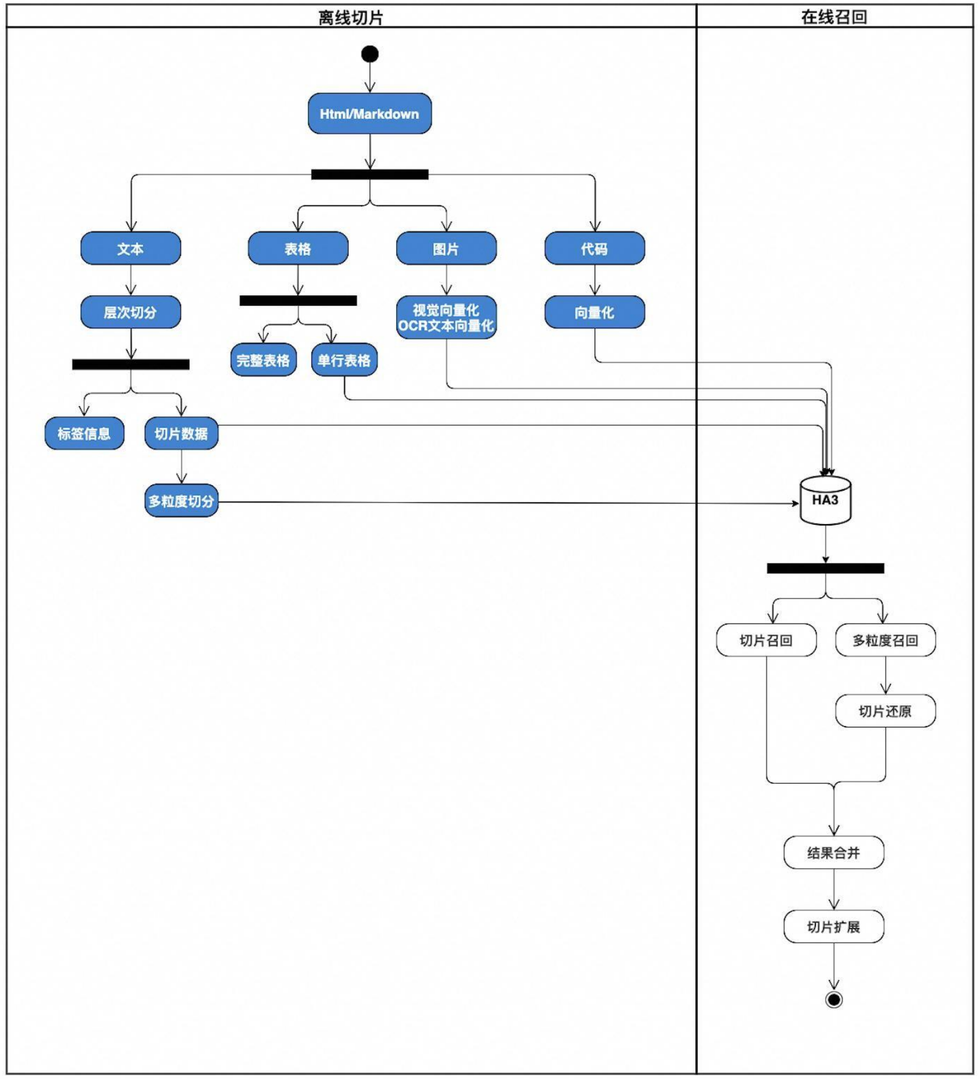
文件解析+切片

层次切分



多粒度切分

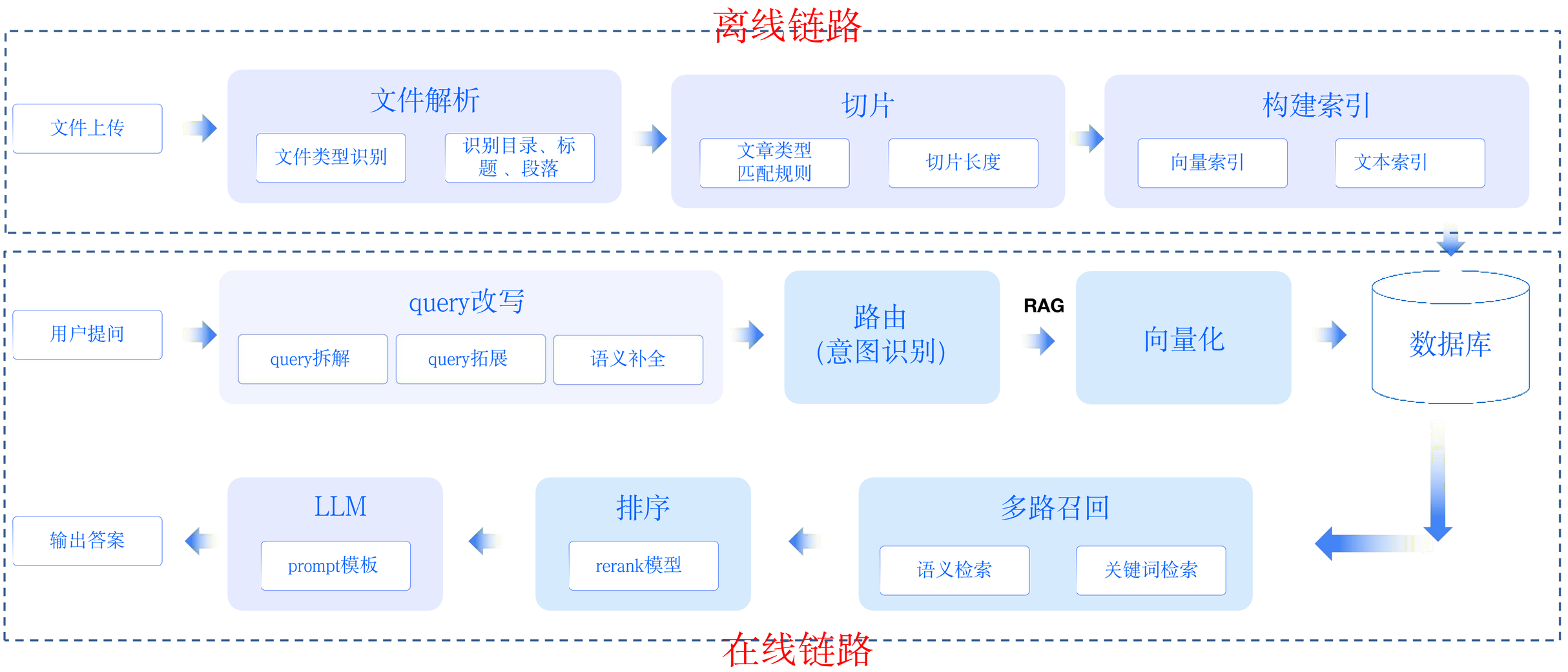




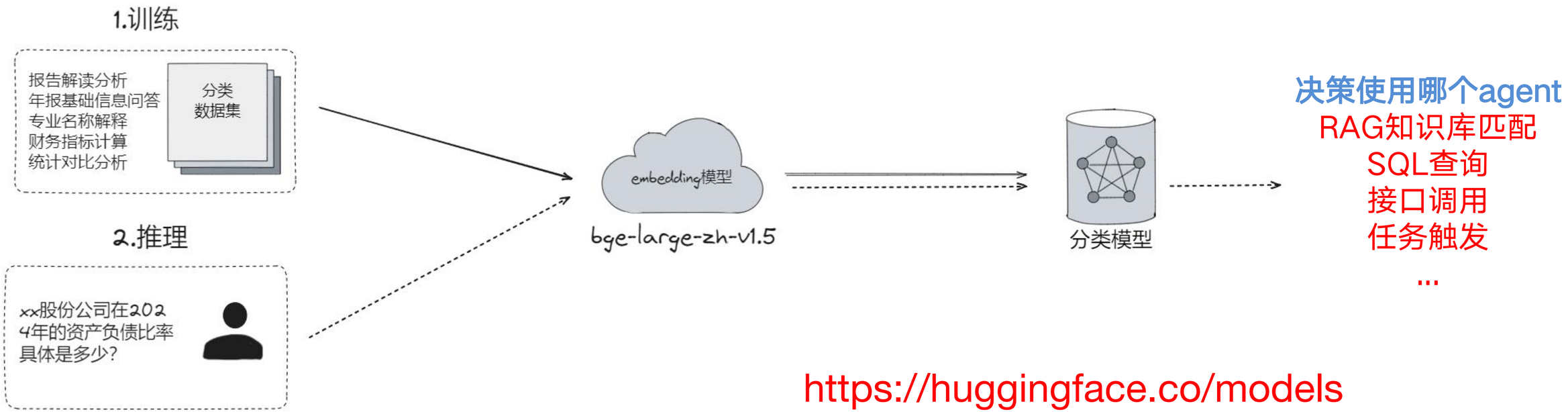
向量库选型

向量库	适用场景	优点	缺点
FAISS	大规模检索，支持 GPU	高效、支持多种索引、支持 GPU 加速	无法存储元数据，缺少持久化
Annoy	中小规模，内存受限场景	内存效率高、构建速度快	不支持动态更新索引，功能相对简单
ScaNN	超大规模，高效检索	自适应量化和分片设计，内存利用率优秀	仅支持 CPU 和 Linux，文档较少
HNSW	高维向量，高精度检索	检索精度高，支持动态更新	不支持 GPU 加速，缺少元数据存储功能
Milvus	生产级应用	持久化存储、灵活多样的索引类型支持	部署运维复杂，资源需求大
Pinecone	托管向量数据库服务	高可用性、支持元数据检索，无需基础设施管理	商业收费，数据隐私可能成问题
Weaviate	向量+元数据一体化应用	支持复杂查询，图数据库功能，丰富的集成	部署复杂，资源消耗较大

详细技术方案


离线链路
文件上传
文件解析
文件类型识别
识别目录、标题 、段落
切片
文章类型匹配规则
切片长度
构建索引
向量索引
文本索引
用户提问
query改写
query拆解
query拓展
语义补全
路由（意图识别）
RAG
向量化
数据库
输出答案
LLM
prompt模板
排序
rerank模型
多路召回
语义检索
关键词检索
在线链路

意图识别



https://huggingface.co/models

https://modelscope.cn/my/overview

分组讨论

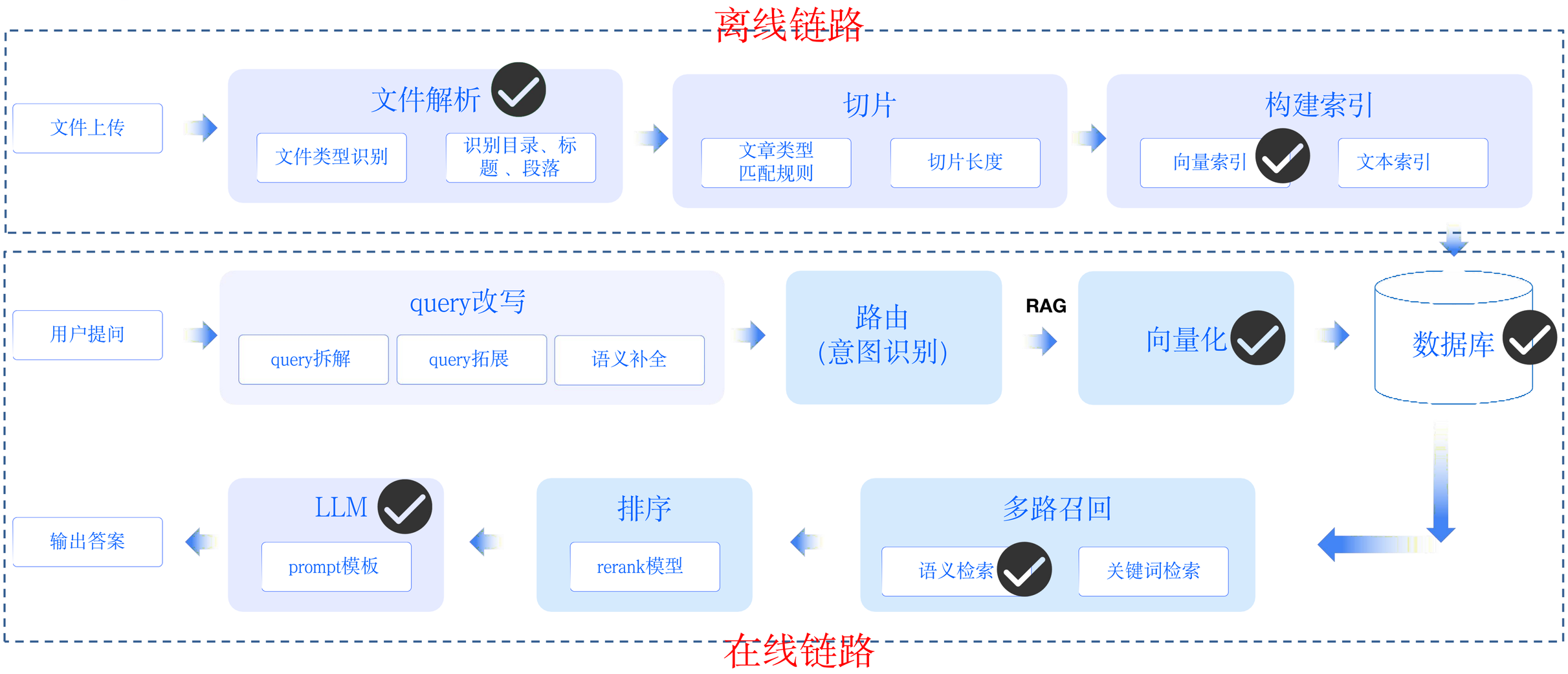
Ques：
结合你们的生活体验和日常项目经验，你们觉得 RAG适合用在什么场景呢？

每位同学可以先2分钟时间自己思考一下：
想象你是某个领域的用户（如购物、学习、医疗），你希望智能助手能解决什么问题？

。。。

然后分小组讨论，10分钟充分交流观点，选一个组长上来分享一下你们组内的idea~

Project2-简化后的技术方案



Project2-简化后的技术方案

1. 离线链路（数据预处理）：

读取知识库文本->向量化->写入向量库
知识库文本->转为sql->导入mysql数据库

2. 在线链路（数据消费）：

用户输入查询
->向量化->向量库返回查询结果索引
->根据索引匹配到mysql数据表结果
->根据查询结果生成context上下文
->带入llm->展示llm结果

Project2-简化后的离线链路

1. 读取数据（运动鞋店铺知识库.txt）
   ├── 考虑 pandas.read_csv
   └── （可选）每次读取多行，批次处理

2. 对读取的数据块执行以下操作：
   ├── 调用embedding模型，将每行数据向量化 参考API
      ├── 参考DashScope API（支持批次调用）：
https://help.aliyun.com/zh/model-studio/developer-reference/dashscopeembedding

Project2-简化后的离线链路

3. 将向量写入faiss库
    ├——— 使用llama-index#FaissVectorStore
        ├——— 参考文档
https://docs.llamaindex.ai/en/v0.10.23/api_reference/storage/vector_store/faiss/
    └——— 将索引写入到本地文件XXX.index

4. 将数据导入到mysql
    ├——— 执行schema.sql里的建表语句
    ├——— 执行 SOURCE /你的路径到文件/ai_context_insert.sql
（注意windows系统的路径格式哈）

Project2-mysql安装

1. mysql server安装

2. 你需要一个mysql client（可以下载mysqlWorkbench
或者直接命令行连接）

3. 在mysql server上新建数据库
mysql -u root -p # 登录mysql
CREATE DATABASE my_database（自定义）; #新建数据库
show databases; # 查看是否新建成功
use my_database; # 切换到你的数据库

Project2-离线侧验证方案

```python
import numpy as np
import faiss

faiss_read_index = faiss.read_index('../../output/faiss_index_test_shop.index')

# 假设索引是 1536 维的
query_vector = np.random.rand(1536).astype("float32")

# 查询前5个相似向量
try:
    distances, indices = faiss_read_index.search(query_vector.reshape(1, -1), k=2)
    print(f"Indices of nearest neighbors: {indices}")
    print(f"Distances: {distances}")
except Exception as e:
    print(f"Query failed: {e}")
```

输出

```
/Users/beibei/.pyenv/versions/3.11.0/bin/python
Indices of nearest neighbors: [[4 3]]
Distances: [[4817.6924 4997.4014]]

Process finished with exit code 0
```

Project2-控制台输出

```
用户： 鞋子有些什么尺码呢
助手： 根据提供的信息，鞋子的尺码范围在30-48之间。建议您参考提供的尺码表，并根据自己的脚长来选择合适的尺码。请注意，这里的尺码可能是指欧洲尺码系统。如果您需要更具体的指导，比如转换为国际通用尺码，可能需要更多的信息或参照具体品牌的尺码指南。
用户： 36码的脚可以买到合适的吗
助手： 根据提供的信息，您可以买到适合36码脚的鞋子，因为鞋子的尺码范围在30-48之间，所以36码在这个范围内。建议您参考提供的尺码表，根据自己的脚长选择合适的尺码以获得最佳的穿着体验。请注意，这里提到的尺码可能是按照欧洲尺码标准。如果您需要将其他尺码标准（如中国或美国尺码）转换为欧洲尺码，建议查看具体的尺码对照表。
用户： 鞋子有些什么品牌呢
助手： 提供的鞋款包括多个知名品牌，如Adidas、Nike、Puma、Reebok、New Balance、Asics、Under Armour等运动鞋品牌，还有Converse、Vans等休闲时尚品牌，以及Gucci、Balenciaga等高端奢侈品牌鞋款。您可以根据自己的需求和喜好选择合适的品牌和款式。
用户： 好的谢谢 我没有问题了
助手： 感谢您的咨询，再见！
==========conversation finish!
```