






RAG 技术详解与实践应用

第19讲： RAG × 知识图谱：从关系结构中召回更准确的内容



目录

-  1. 上节回顾与问题引入
-  2. 知识图谱简介
-  3. 知识图谱增强的RAG架构
-  4. 基于知识图谱RAG的开源方案
-  5. LightRAG实战与案例分析



基础概念介绍

- **AI Agent**: 具备记忆、规划、工具调用能力的智能体，可自主完成复杂任务。
- **Agent 的四种工作流**: 包括 Function Call 、 ReAct 、 PlanAndSolve 和 ReWOO , 各自对应不同的思考和执行机制。

Agentic RAG的核心原理

- 将传统RAG中的检索和生成组件替换为更智能的**AI Agent**。
- 支持多轮检索、多数据源、多工具调用和动态决策，提升准确率和智能性。
- 包括**单Agent**和**多Agent**两种方式，分别适用于不同任务复杂度。

实践与实现方式

- 使用LazyLLM构建传统RAG流程。
- 用AI Agent取代原有检索器。
- 完成一个可多轮搜索、智能应对复杂问题的升级版RAG系统。



现有问题

传统的RAG系统（即朴素RAG）依赖于文本块的向量化检索，忽略了文档间实体关系与上下文结构，包括：

- 语义割裂：检索结果通常为相互孤立的文本段，缺乏上下文衔接，无法反映实体间的内在逻辑；
- 结构缺失：无法识别文档间的组织结构或主题关系，难以支持跨文档、跨段落的推理；
- 信息冗余与漏召：召回片段常包含冗余内容，同时遗漏具有重要语义关联的信息。

这些问题限制了RAG在复杂问答、逻辑推理、多跳检索等高阶场景中的性能表现。






发展趋势：知识图谱引导的结构化检索增强

为克服朴素RAG在检索阶段结构缺失的问题，近期研究引入知识图谱（Knowledge Graph, KG）作为结构化语义补充。知识图谱以 实体-关系-属性 三元组形式组织知识，使系统具备：

- 语义关联显式建模：通过关系边连接实体节点，清晰刻画知识之间的语义路径；
- 结构感知的检索机制：支持基于图结构的遍历与路径匹配，使召回内容更具逻辑连贯性；
- 主题聚合与推理能力增强：帮助系统对跨文档实体进行对齐和融合，提升多跳问答与复杂推理能力。



目录

-  1. 上节回顾与问题引入
-  2. 知识图谱简介
-  3. 知识图谱增强的RAG架构
-  4. 基于知识图谱RAG的开源方案
-  5. LightRAG实战与案例分析

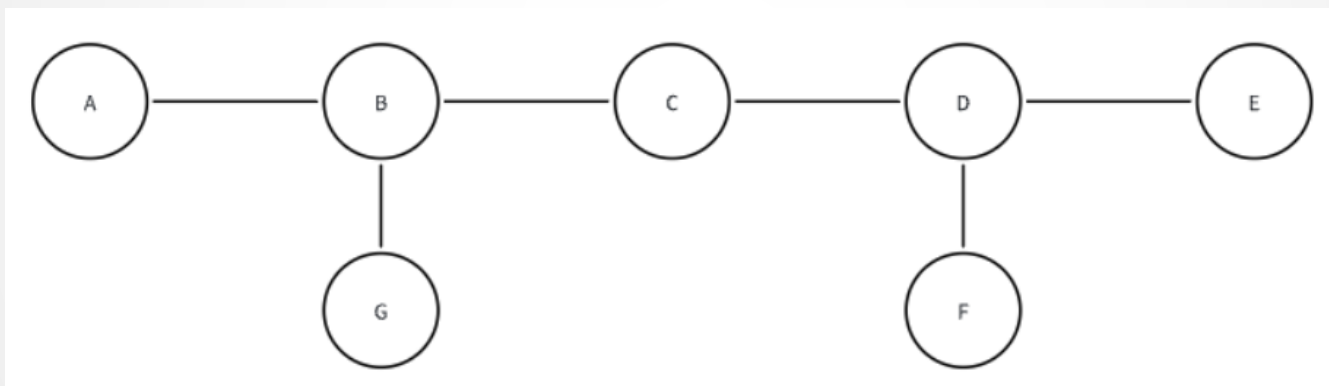


从图结构看知识的组织方式

🤔 那什么是“图”呢？

图 (Graph) 是一种强大的数据结构，由以下两部分组成：

- 节点 (Node)：表示实体 (如人、地点、概念等)
- 边 (Edge)：表示节点之间的关系 (如属于、引用、朋友关系等)

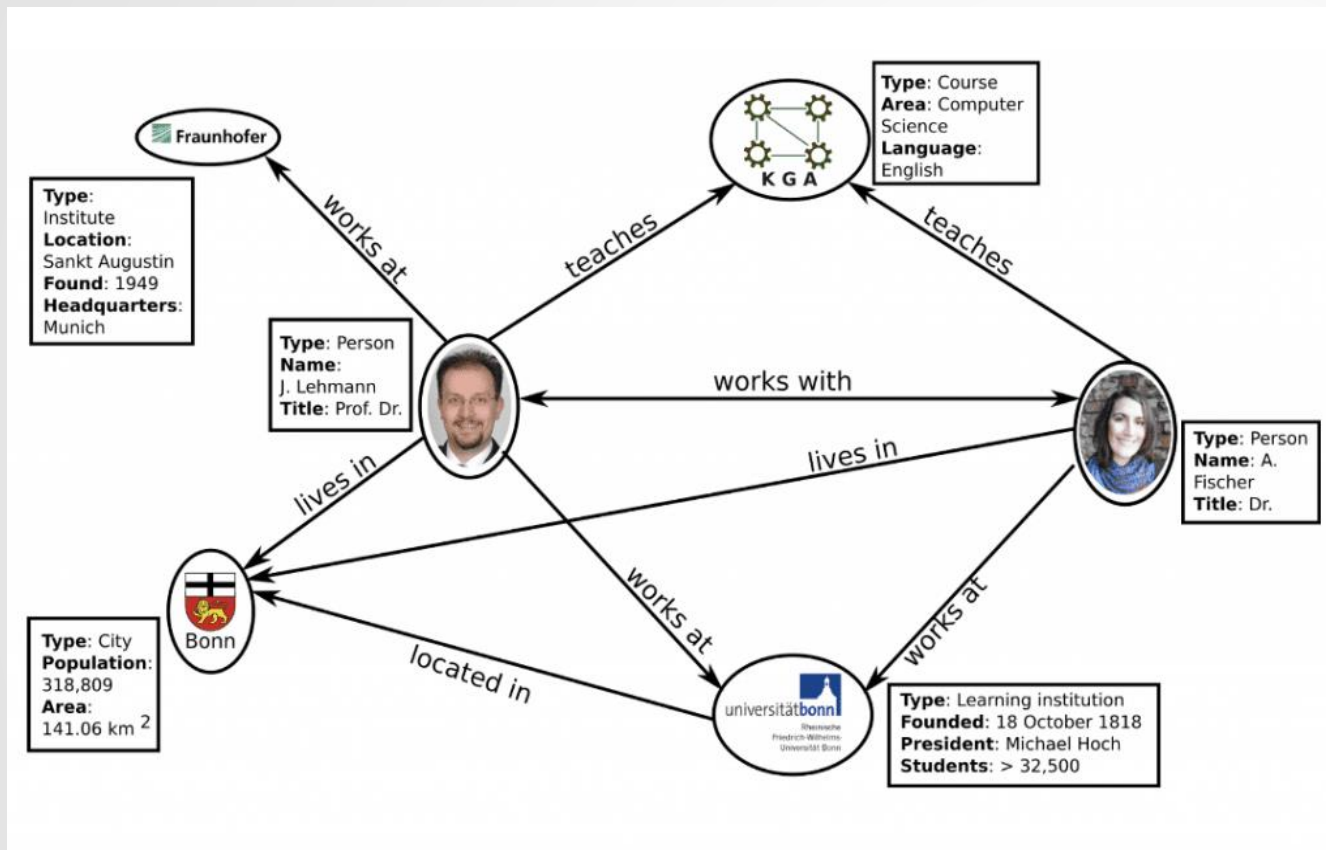


图结构 = 节点 (实体) + 边 (关系)

知识图谱 (Knowledge Graph, KG) 是一种用“图”结构表示知识的方式，由实体 (节点) 和关系 (边) 组成，能够将各种信息抽象成结构化的形式，进而支持机器对其进行理解、推理和查询。



知识图谱：让机器理解世界的“知识网络”

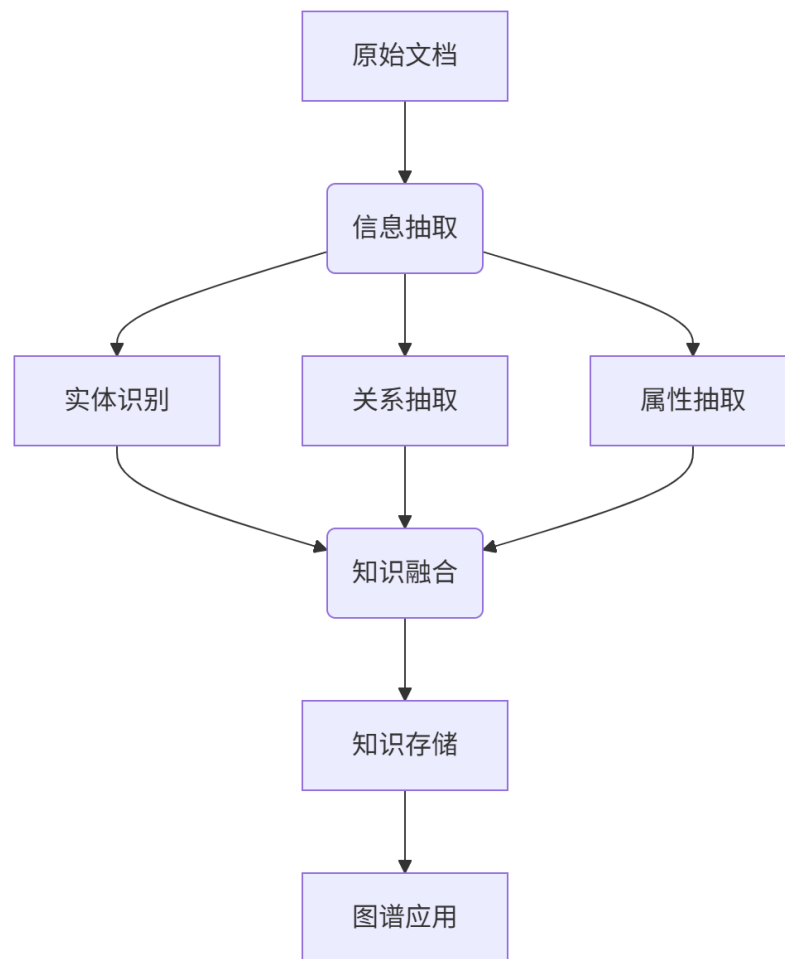


- **实体 (Entity)**：知识图谱的基本构成单位，图中的点，代表现实世界中的人物、物品、概念、事件等。
- **关系 (Relationship)**：描述实体间联系的方式，通常用边来表示。比如图2中男士与Bonn的关系是“lives in”，表示这位男士生活在Bonn City。
- **属性 (Attribute)**：实体或关系的附加信息。例如图中Bonn City有“Type”，“Population”和“Area”三个属性。



从文档中构造知识图谱

- **文档解析**：将 PDF、Word、网页等非结构化文档转化为纯文本，清除格式与乱码。
- **信息抽取**：识别实体、关系和属性，形成基本的结构化三元组。
- **知识融合**：对同一实体进行合并处理，解决多义词和冲突信息。
- **知识存储**：将抽取的三元组存入图数据库中，便于后续查询与分析。
- **图谱应用**：用于可视化展示、智能问答或企业决策辅助等实际场景。



以水浒传为例，提取一段原文中的实体与关系示例。

□ 原文

起灯烛荧煌，焚起香来。宋江在当中证盟，朝着涌金门下哭奠。戴宗立在侧边。先是僧人摇铃诵咒，摄招呼名，祝赞张顺魂魄，降坠神幡。次后戴宗宣读祭文。宋江亲自把酒浇奠，仰天望东而哭。正哭之间，只听得桥下两边，一声喊起，南北两山，一齐鼓响，两彪军马来拿宋江。正是：方施恩念行仁义，翻作勤王小战场。正是：直诛南国数员将，搅动西湖万丈波。毕竟宋江、戴宗怎地迎敌，且听下回分解。此一回内，折了三员将佐：郝思文、徐宁、张顺。京师取回一员将佐：安道全\n第一百五十五段 话说浙江钱塘西湖这个去处，果然天生佳丽，水秀山明。正是帝王建都之所，名实相孚，繁华第一。自古道：江浙昔时都会，钱塘自古繁华。却才说不了宋江和戴宗正在西陵桥上祭奠张顺，不期方天定已知，着令差下十员首将，分作两路来拿宋江，杀出城来。南山五将是吴值、赵毅、晁中、元兴、苏泾；北山路也差五员首将，是温克让、崔彧、廉明、茅迪、汤逢士。南兵两路，共十员首将，各引三千人马，半夜前后开门，两头军兵一齐杀出来。宋江正和戴宗奠酒化纸，只听得桥下喊声大举。左有樊瑞、马麟，右有石秀，各引五千人埋伏。。。

□ 提取结果

\n\n(\n"entity" <|>\n樊瑞" <|>\n"person" <|>\n樊瑞在宋江的军队中担任伏击任务，参与了对南兵的攻击。\\")

...

(\n"entity" <|>\n林冲" <|>\n"person" <|>\n林冲在宋江的军队中参与了对独松关的战斗，与蒋印交战。\\")##\n(\n"entity" <|>\n吕方" <|>\n"person" <|>\n吕方在宋江的军队中参与了对独松关的战斗，杀了厉天祐。\\")

...

(\n"entity" <|>\n卢先锋" <|>\n"person" <|>\n卢先锋率领军队取下了独松关，在独松关与敌人激战。\\")##

...

(\n"relationship" <|>\n卢先锋" <|>\n独松关的夺取" <|>\n卢先锋领导军队成功地取下了独松关，击败了吴升等守关将领。\\" <|>\n战斗,胜利" <|> 10)##

(\n"relationship" <|>\n林冲" <|>\n独松关的夺取" <|>\n林冲在对独松关的战斗中表现出色，成功地伤了蒋印。 <|>\n战斗,胜利" <|> 8)

...

(\n"content_keywords" <|>\n战斗,伏击,胜利,防御,攻击,敌我交战,夺取关隘\\")<|COMPLETE|>

这段原文是独松关之战的内容，其中的难点在于：实体歧义，**卢先锋**实际上是**卢俊义**，但片段原文未体现。



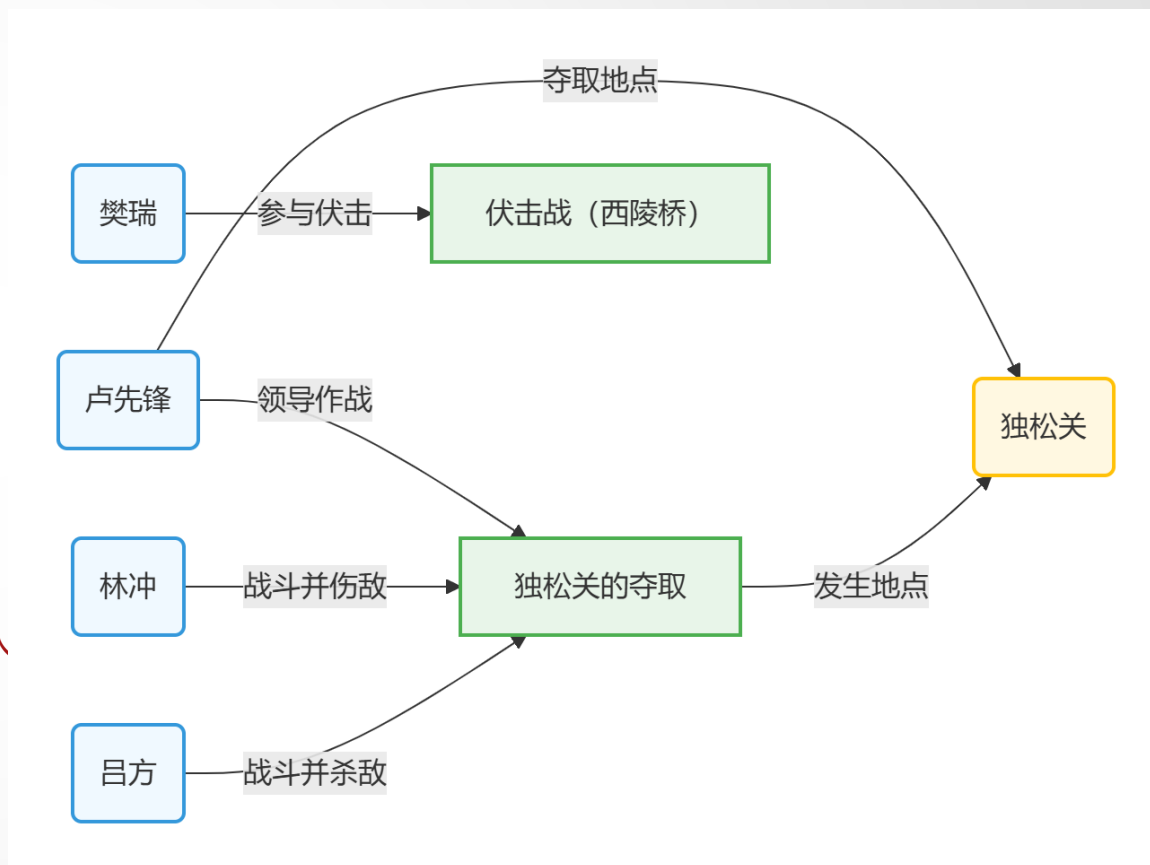
□ 提取结果说明

(\ "entity" \ "<|>" 卢先锋" \ "<|>" person" \ "<|>" 卢先锋率领军队
取下了独松关，在独松关与敌人激战。 \")##

"entity" <|> 实体名 <|> 实体类型 <|> 描述 <|> 属性1 <|> 属性2 <|> ...

```
{  
  "entity": "卢先锋",  
  "type": "person",  
  "description": "卢先锋率领军队取下了独松关，在独松关与敌人  
  激战。",  
  "organization": "宋江军",  
  "skills": ["...", "..."]  
}
```

□ 提取结果图示



□ 提取结果说明

"relationship" 的表示:

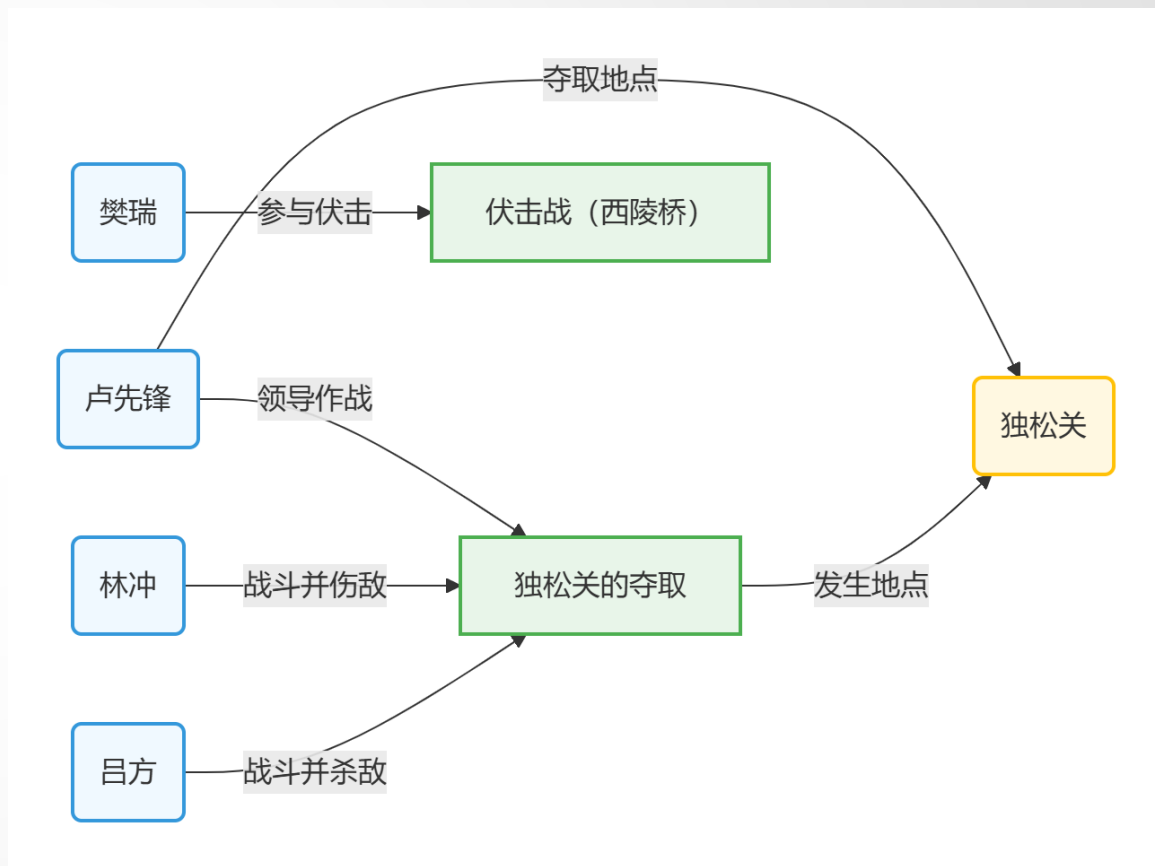
"relationship" <|> 实体1 <|> 实体2 <|> 关系描述

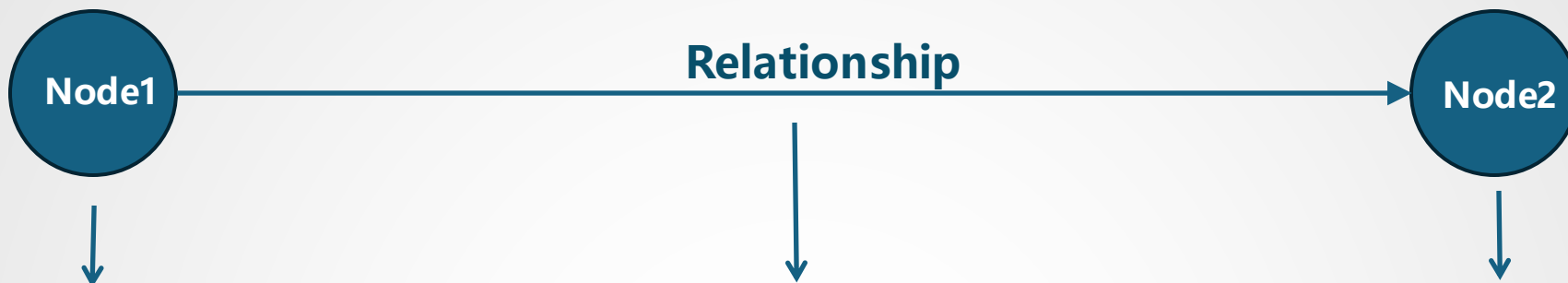
上述形式可以抽象出 ("实体A", "关系", "实体B"), 通过该元组即可构建知识图谱的结构: 节点 (实体) + 边 (关系)
例如:

"relationship" <|> "卢先锋" <|> "独松关的夺取" <|> "卢先锋领导军队成功地取下了独松关....."

通过这行表示即可建立实体 "卢先锋" 与实体 "独松关的夺取" 之间的联系。

□ 提取结果图示





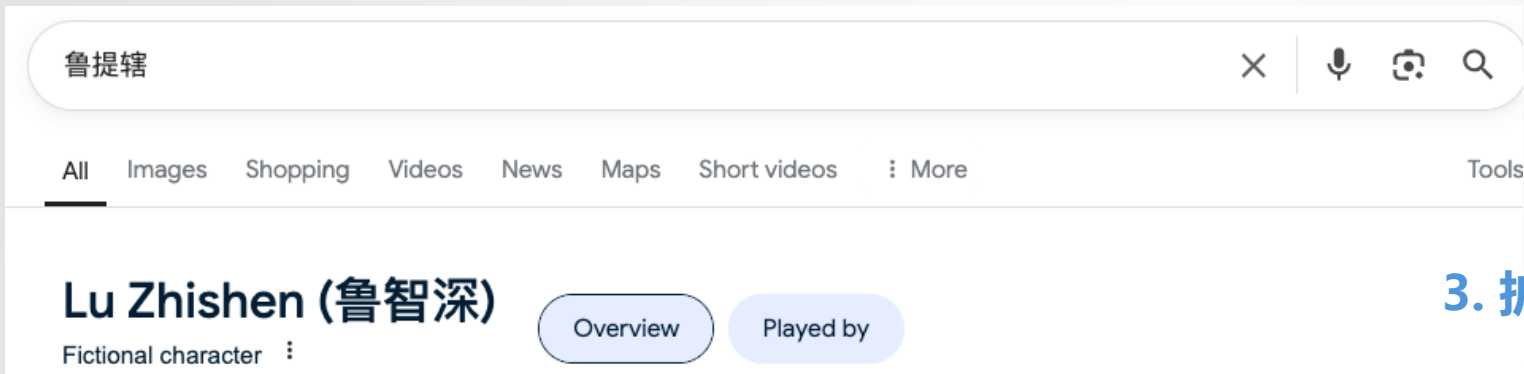
```
<node id="&quot;卢先锋&quot;">
  <data key="d0">"PERSON"</data>
  <data key="d1">"卢先锋在军事行动中发挥领导
作用，指挥军队进攻，并在军事行动后进行奖励。
"&lt;SEP&gt;"卢先锋是军队中的指挥角色，负责决
策和指挥军事行动。"&lt;SEP&gt;"卢先锋是前线的
军事指挥团队，攻破晋宁，并赢得了对
阵孙安的战斗。"&lt;SEP&gt;"卢先锋是宋军的指挥
官，指挥军队攻占昱岭关并获得关键胜利。
"&lt;SEP&gt;"卢先锋是宋江手下的将领，曾接待孙
安，并派遣孙安前往壶关探听消息。"&lt;SEP&gt;"
卢先锋是宋江手下的将领，被派遣去攻打湖>州。
"&lt;SEP&gt;"卢先锋是晋宁的指挥官，戴宗前往晋
宁探听军情的对象。"&lt;SEP&gt;"卢先锋是梁山泊
的好汉之一，被贺统军围困。"&lt;SEP&gt;"卢先锋
是这场战役中的重要指挥官，负责制定策略和指挥
战斗。"&lt;SEP&gt;"卢先锋率领
军队取下了独松关，在独松关与敌人激战。
"</data>
```

```
<edge source="&quot;卢俊义
&quot;" target="&quot;卢先锋
&quot;">
  <data key="d3">8.0</data>
  <data key="d4">"卢俊义作为卢先
锋，统领军队进攻玉田县。"</data>
  <data key="d5">"军队指挥, 战略推
进"</data>
  <data key="d6">chunk-
a7949e3833d3220bc3a90adf93e85
5f8</data>
</edge>
```

```
<node id="&quot;卢俊义&quot;">
  <data key="d0">"PERSON"</data>
  <data key="d1">卢俊义，绰号玉麒麟，是
大名府的长者和富豪，以高超的武艺尤为擅长
棍棒技巧而闻名。在梁山泊中，卢俊义与宋江
并列成为重要将领，经常担任先锋的角色。他
曾独自抵抗四个番将并斩杀耶律宗霖，也参与
多次军事行动>，如攻打蓟州和檀州。被任命
为副先锋，卢俊义不仅负责领导军队攻击多个
城市，包括宣州、湖州等地，同时也负责执行
宋江下达的各项军事任务。在平定各城市的过
程中，卢俊义展现出了英勇的战斗能力及杰出
的统帅才能。作为宋江的重要
副手，卢俊义不仅指挥中军，还与其他好汉一
起商讨战略，且亲自领兵发起攻击以破坏敌军。
不仅参与军事行动，卢俊义还协助宋江处理招
安事宜，并在对抗辽国的侵扰中扮演关键角色。
此外，他亲自带领军队突破昱岭关，进攻歙州
等重要行动，展现了其军事才能和领导才
能。。。
```



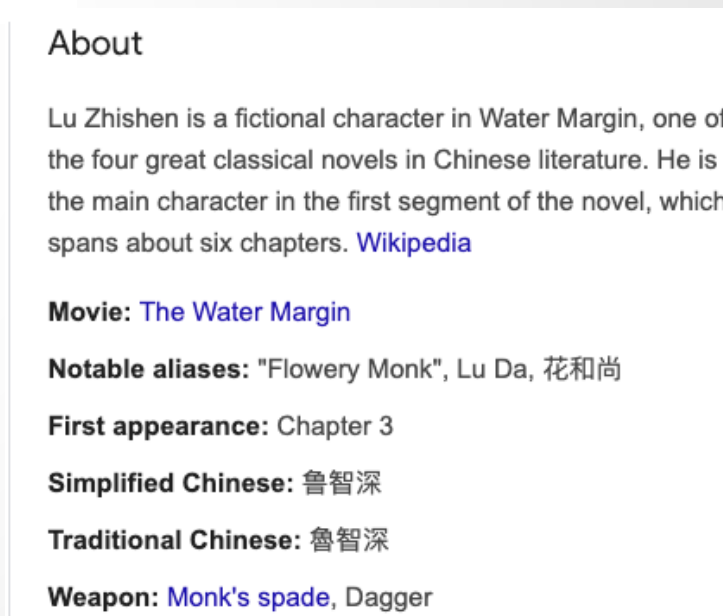
1. 搜索概念而非字符串








2. 围绕主题汇总相关内容



3. 扩大找回，解决潜在意图（搜索特有）



目录

-  1. 上节回顾与问题引入
-  2. 知识图谱简介
-  3. 知识图谱增强的RAG架构
-  4. 基于知识图谱RAG的开源方案
-  5. LightRAG实战与案例分析



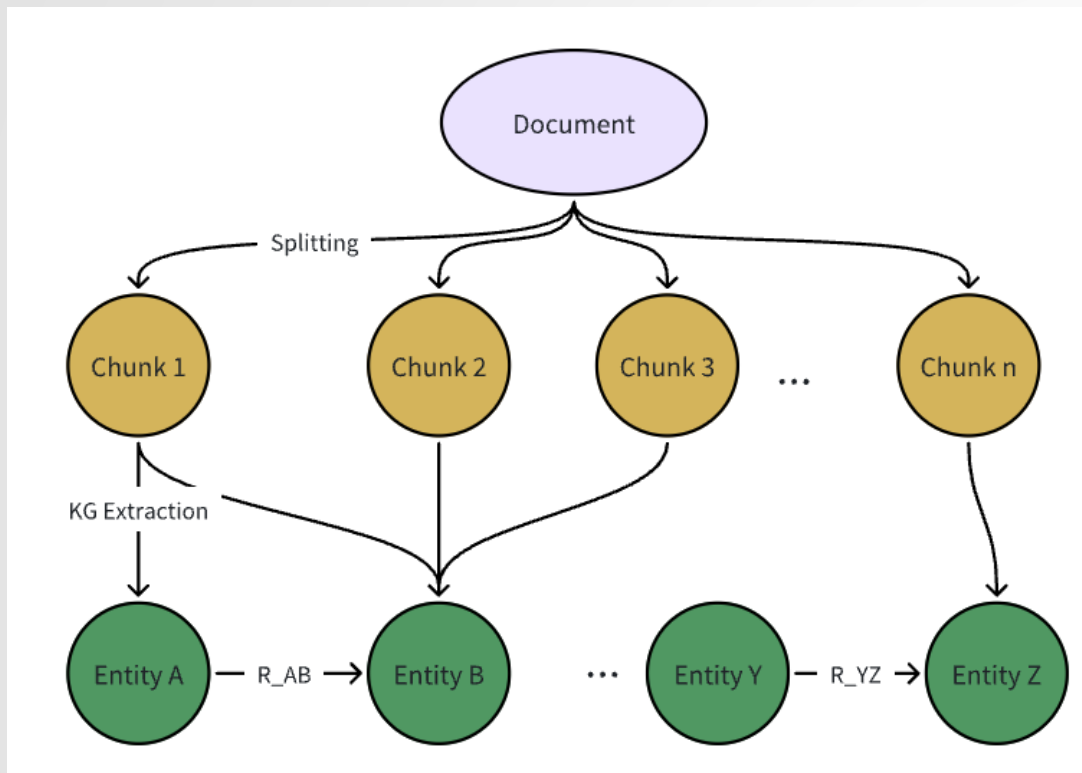


图 基于RAG构建知识图谱的流程

分片

- 这一步与普通RAG中的分块方法一致。

提取

- **提取实体与关系**

识别文本中出现的所有实体（包括实体名、类型和描述）以及关系(源实体、目标实体和关系描述)

- **合并实体、关系**

合并不同文档/分片中已存在的实体和关系



RAG系统中的知识图谱检索

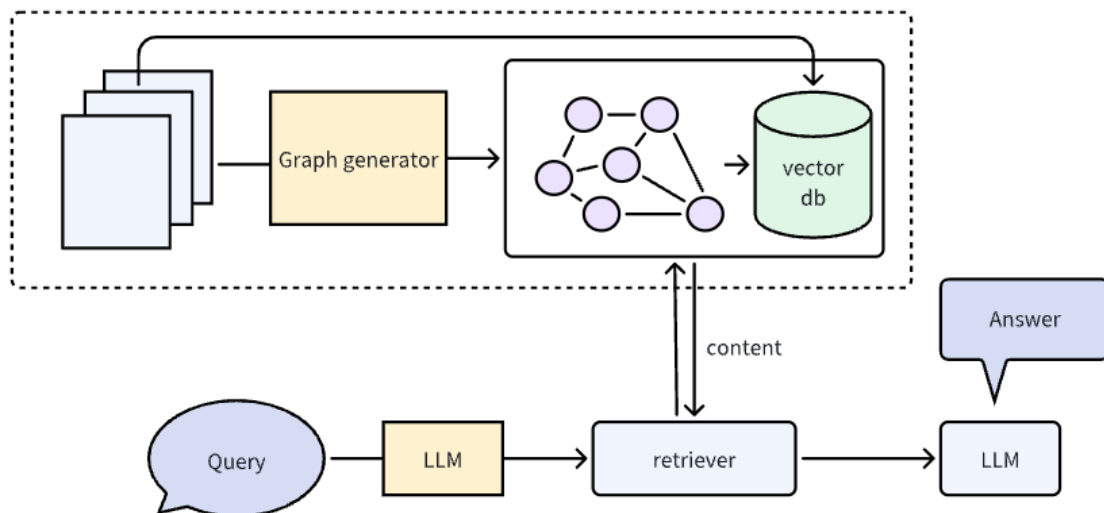


图 基于知识图谱RAG系统回答过程

检索 (Retrieval)

- 根据用户输入问题召回实体、关系和原文。

增强 (Augmented)






- 将召回的结构化结合并为额外的上下文，与用户输入一起提供给大模型。

生成 (Generatation)

- 与朴素RAG系统无区别。

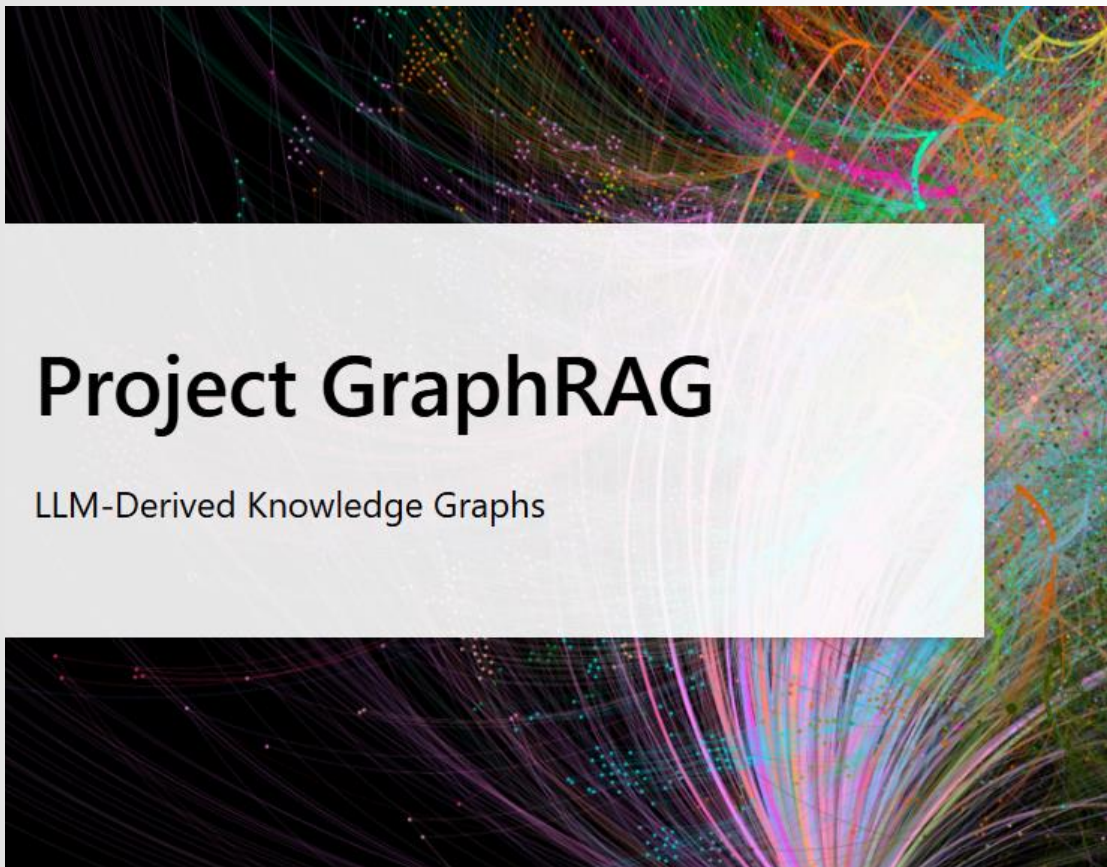


目录

-  1. 上节回顾与问题引入
-  2. 知识图谱简介
-  3. 知识图谱增强的RAG架构
-  4. 基于知识图谱RAG的开源方案
-  5. LightRAG实战与案例分析



GraphRAG通过构建知识图谱和分层社区结构，将局部信息聚合为全局理解。



核心优势：

- **图结构索引**：利用LLM从文档中提取实体、关系和事实声明，构建知识图谱，捕捉语义关联。
- **分层社区检测**：使用图社区算法将图谱划分为紧密关联的社区，递归生成从局部到全局的摘要。
- **Map-Reduce式回答生成**：通过社区摘要的并行处理和聚合，生成全面且多样的全局回答。



工作流程：

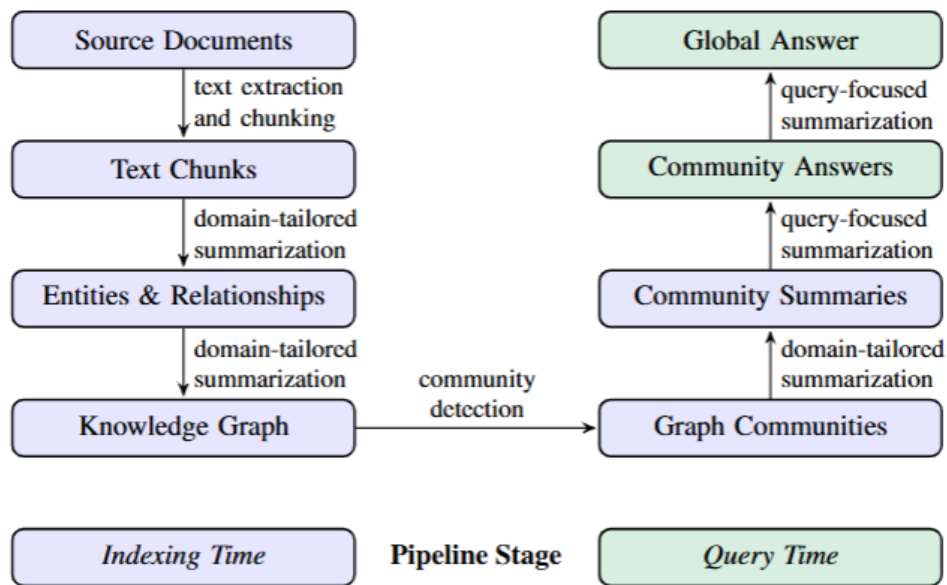


Figure 1: Graph RAG pipeline using an LLM-derived graph index of source document text. This graph index spans nodes (e.g., entities), edges (e.g., relationships), and covariates (e.g., claims) that have been detected, extracted, and summarized by LLM prompts tailored to the domain of the dataset. Community detection (e.g., Leiden, Traag et al., 2019) is used to partition the graph index into groups of elements (nodes, edges, covariates) that the LLM can summarize in parallel at both indexing time and query time. The “global answer” to a given query is produced using a final round of query-focused summarization over all community summaries reporting relevance to that query.

- **文本分块与实体提取**：将文档拆分为文本块，使用LLM提取实体、关系及事实声明。
- **知识图谱构建**：将提取的实体和关系转化为图节点和边，聚合重复实例并生成节点描述，边权重由关系出现频率决定。
- **社区检测与分层**：使用Leiden算法递归划分图为嵌套社区，每个社区代表一个语义主题，形成层级结构。
- **社区摘要生成**：叶社区按节点重要性排序，聚合实体、关系和声明生成摘要。高层社区通过递归整合子社区摘要，平衡细节与全局视角。
- **查询处理与全局回答**：每个社区摘要独立生成部分回答，通过评分过滤无关内容，聚合高分部分回答，生成最终全局回答。



GraphRAG 的核心创新在于 **分层社区结构** 的引入。

Community:

ID: 社区的唯一标识符。

Level: 社区的层级 (如 0、1、2 等)。

Entity IDs: 社区中包含的实体 ID 列表。

Relation IDs: 社区中包含的关系 ID 列表。

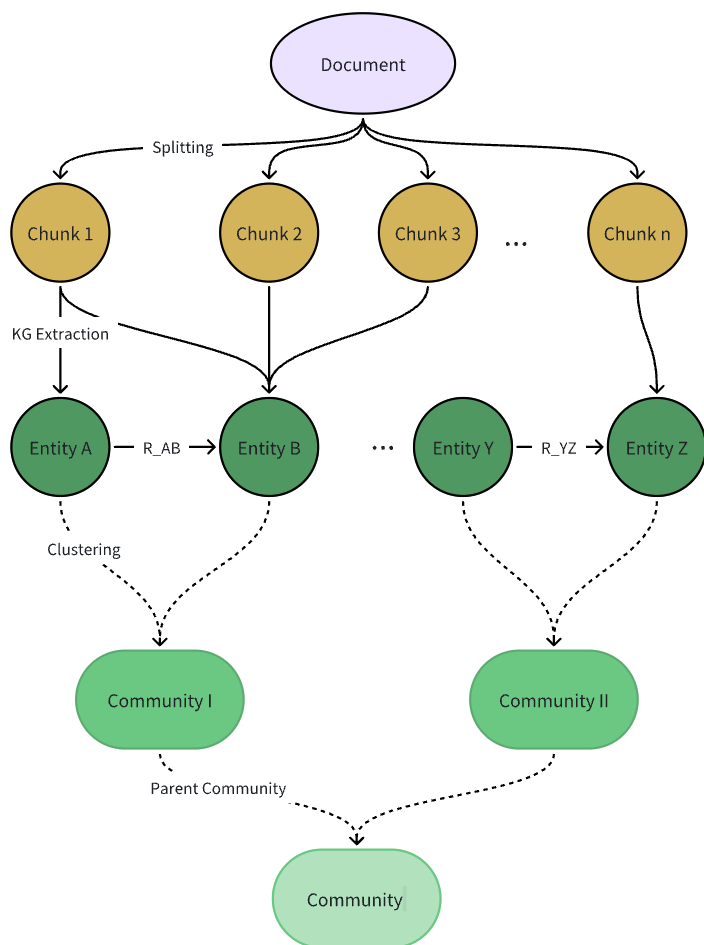
Text Block IDs: 社区中包含的文本块 ID 列表。

Description: 社区的描述信息。

Summary: 社区的摘要信息。

社区生成 由两步组成:

- 社区检测**: 使用图分析算法, 获得具有高度连接性的实体簇。
- 摘要提取**: 使用LLM根据社区中的实体和关系, 提取各类型总结。



LightRAG的知识图谱是比较基础的 **分片+实体/关系**

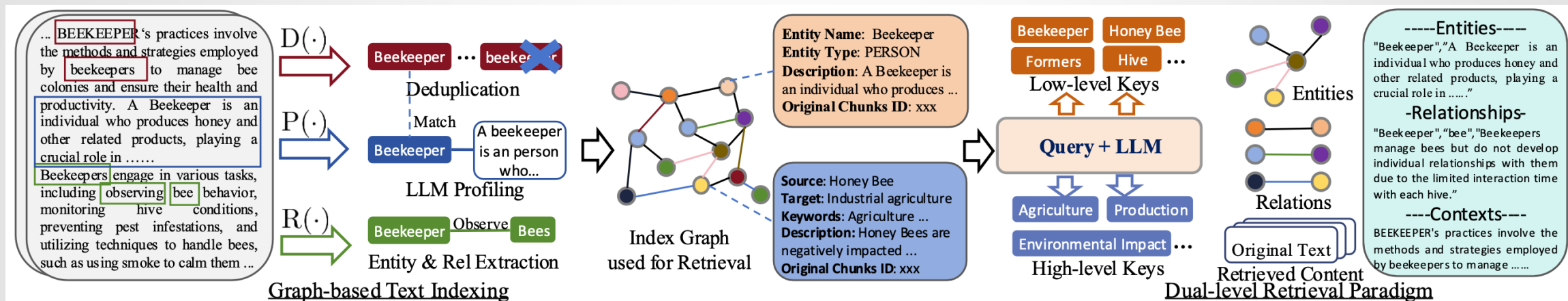


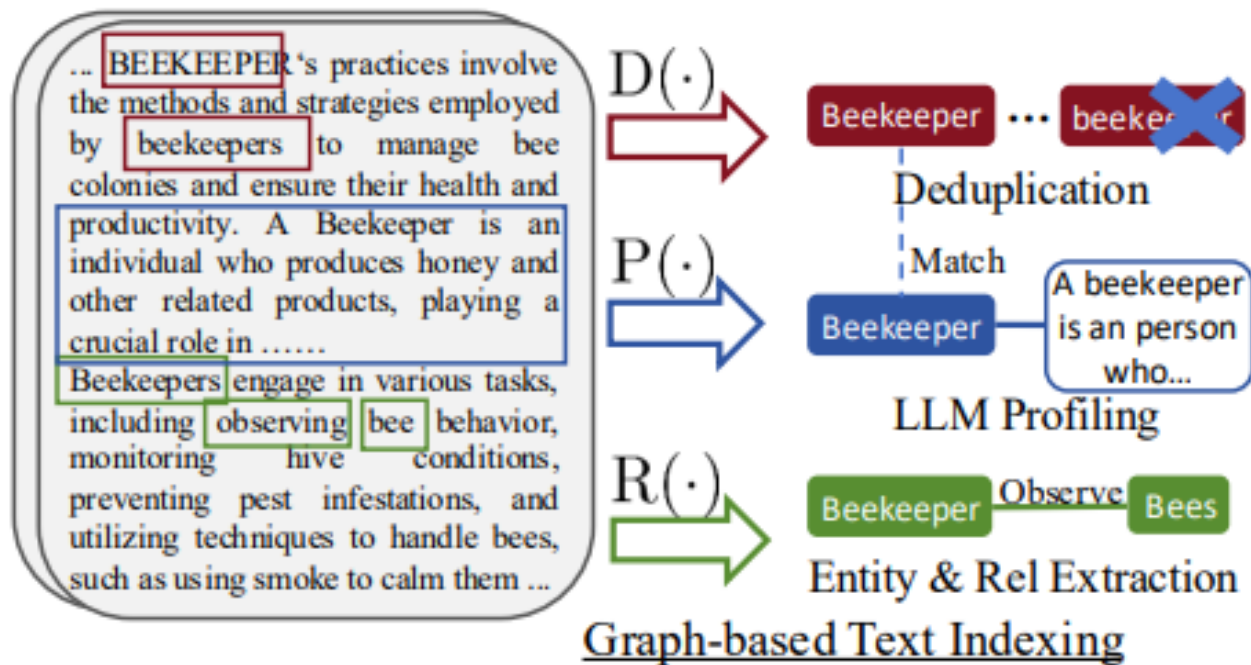
Figure 1: Overall architecture of the proposed LightRAG framework.

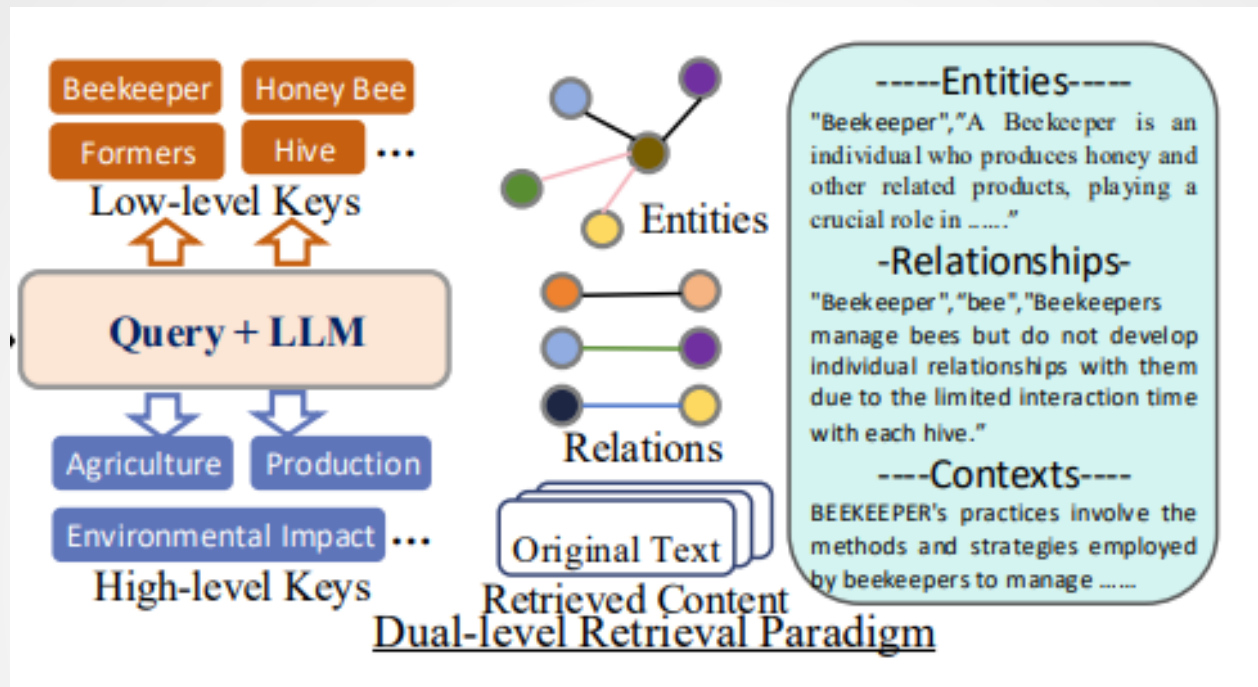
LightRAG 是一种 **轻量级** 的知识图谱RAG，它没有使用层次化的社区聚类，而直接构造了知识图谱，从而支持了增量更新，无需每添加一部分数据都需要进行实体和关系的提取。在检索方面采用了**双层检索策略**，包括**低层级检索**（集中关注特定实体的信息）和**高层级检索**（处理更广泛的主题）



Graph-based Text Indexing

- **实体与关系提取**：利用LLM从文档中提取实体（如人名、地点）和关系（如“影响”“关联”），构建知识图。
- **LLM 特征分析**：为每个实体和关系生成键值对，键为检索关键词，值为摘要文本，优化检索效率。
- **增量更新算法**：新数据加入时，仅需更新新增的图节点和边，无需重建整个索引，大幅降低计算开销。





Dual-level Retrieval Paradigm

- **低级检索**：聚焦具体实体及其属性，如检索 “电动汽车” 的技术参数。
- **高级检索**：捕捉主题和全局关系，如 “电动汽车对城市交通的整体影响”。
- **图与向量融合**：通过提取查询中的本地关键词和全局关键词，结合向量数据库匹配实体，并利用图结构扩展检索到相关邻居节点，提升结果全面性。



LightRAG VS GraphRAG

LightRAG 相比 GraphRAG 最大的改进在 **成本** 上。

处理阶段	query-召回		文档创建 (5000K Token)	
方案	GraphRAG	LightRAG	GraphRAG	LightRAG
Token消耗	610*1000	< 100	1399*5000 + T_extract	T_extract
API 调用次数	$610 \times 1000 / C_{Max}$	1	1399 + C_extract	C_extract

对于Legal Dataset, 共 **94** 篇文章, 约 **5000K Token** (约5个水浒传大小) 。






- GraphRAG 生成了 **1399** 个communities, 平均每个社区报告生成需要 **5000** Token, 其中 **610** 个 level-2 communities, 平均每个level-2的社区占用 **1000** Token。

上表对比了创建知识图谱和一次检索所消耗的LLM Token, 注意GraphRAG 从 **0.4.0** 开始支持增量索引。





- C_Max** 为API单次请求最多Token限制
- T_extract** 代表提取实体和关系消耗Token数
- C_extract** 则表示提取产生的API调用次数。



目录

-  1. 上节回顾与问题引入
-  2. 知识图谱简介
-  3. 知识图谱增强的RAG架构
-  4. 基于知识图谱RAG的开源方案
-  5. LightRAG实战与案例分析



Python 依赖项	安装方式
 vLLM	<code>pip install vllm</code>
 LightRAG	<code>pip install lightrag-hku</code>
Infinity 	<code>pip install infinity-emb</code>
 LazyLLM	<code>pip3 install lazyllm</code>



由于在线LLM API服务通常有内容安全检查策略，它有一定概率会误判，尤其是处理长篇内容时，可能对部分片段拒绝返回，因此建议此使用本地资源部署LLM服务。

1. 下载模型(Qwen 2.5-32B-Instruct)，推荐从ModelScope下载

2. 创建虚拟环境，安装vLLM

3. 启动openai 兼容模式的推理服务

```
python -m vllm.entrypoints.openai.api_server --model /mnt/lustre/share_data/lazyllm/models/Qwen2.5-32B-Instruct/ --served-model-name qwen2 --max_model_len 16144 --host 0.0.0.0 --port 12345
```

4. 测试一下服务OK

！注：vLLM启动的服务对应的openai格式的base_url为：http://{ip}:{port}/v1，若用curl测试则url为

http://{ip}:{port}/v1/chat/completions

```
(base) → ~ curl http://192.168.1.10:12345/v1/chat/completions \
-H "Content-Type: application/json" \
-d '{
  "model": "qwen2",
  "messages": [
    {"role": "system", "content": "You are a helpful assistant."},
    {"role": "user", "content": "你是谁？"}
  ]
}'
{"id": "chatcmpl-e92ea486f9f94d42921ed200831c42aa", "object": "chat.completion", "created": 1745823991, "model": "qwen2", "choices": [{"index": 0, "message": {"role": "assistant", "reasoning_content": null, "content": "我是来自阿里云的语言模型，我叫通义千问。我被设计用来回答各种问题、进行对话互动和提供信息帮助。有什么我可以帮助你的吗？", "tool_calls": [], "logprobs": null, "finish_reason": "stop", "stop_reason": null}}, {"usage": {"prompt_tokens": 22, "total_tokens": 59, "completion_tokens": 37, "prompt_tokens_details": null, "prompt_logprobs": null}}], "usage": {"prompt_tokens": 22, "total_tokens": 59, "completion_tokens": 37, "prompt_tokens_details": null, "prompt_logprobs": null}}

(base) → ~ █
```



环境配置 – 部署Embedding服务



Embedding可选择线上平台的 **API服务** 或者 **本地部署**。

1. 下载模型(bge-large-zh-v1.5), 推荐从ModelScope下载

2. 创建虚拟环境, 安装infinity_emb

3. 启动openai 兼容模式的推理服务

```
infinity_emb v2 --model-id "/mnt/lustre/share_data/lazyllm/models/bge-large-zh-v1.5" --port 19001 --served-model-name bge-large
```

4. 测试一下服务OK

注: vLLM启动的服务对应的openai格式的base_url为: http://{ip}:{port}, 若用curl测试则url为 http://{ip}:{port}/embeddings

```
curl --location 'http://127.0.0.1:19001/embeddings' --header "Authorization: Bearer TEST" \
--header 'Content-Type: application/json' --data '{
  "model": "bge-large",
  "input": "基于知识图谱的RAG系统",
  "dimension": "1024",
  "encoding_format": "float"
}'
```



LightRAG创建知识图谱



参考 lightrag 仓库中: [examples/lightrag_openai_compatible_stream_demo.py](#)

1. 设置处理的文件名和保存知识图谱的文件夹路径

WORKING_DIR 为 知识图谱保存的文件夹路径

PATH_TO_TXT 为 待解析的小说 <水浒传>, 注意检查是否为utf8格式

2. 修改llm_model的配置

base_url 改为 http://{ip}:{port}/v1

model_name 改为 vLLM中配置的 qwen2

3. 修改openai_embed的配置

base_url 改为 http://{ip}:{port}

model_name 改为 infinity中配置的 bge-large

3. 运行, A800约需4小时

建议在后台运行或将txt 拆成数个文件依次读取并处理



LightRAG创建知识图谱

```
build_knowledge_graph.py U x
build_knowledge_graph.py > ...
1 import os
2 from lightrag import LightRAG
3 from lightrag.llm.openai import openai_complete, openai_embed
4 from lightrag.utils import EmbeddingFunc
5
6 WORKING_DIR = "./shuihu_novel_kg"
7 PATH_TO_TXT = "./shuihu.txt"
8
9 if not os.path.exists(WORKING_DIR):
10     os.mkdir(WORKING_DIR)
11
12 api_key = "empty"
13 rag = LightRAG(
14     working_dir=WORKING_DIR,
15     llm_model_func=openai_complete,
16     llm_model_name="qwen2",
17     llm_model_kwargs={"base_url": "http://192.168.1.100:12345/v1", "api_key": api_key},
18     embedding_func=EmbeddingFunc(
19         embedding_dim=1024,
20         max_token_size=8192,
21         func=lambda texts: openai_embed(
22             texts=texts,
23             model="bge-large",
24             base_url="http://192.168.1.100:19001",
25             api_key=api_key,
26         ),
27     ),
28 )
29
30 with open(PATH_TO_TXT, "r", encoding="utf-8") as f:
31     rag.insert(f.read())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(lightrag) (base) + LightRAG git:(main) * python build_knowledge_graph.py
INFO:nano-vectordb:Init {'embedding_dim': 1024, 'metric': 'cosine', 'storage_file': './shuihu_novel_kg/vdb_entities.json'} 0 data
INFO:nano-vectordb:Init {'embedding_dim': 1024, 'metric': 'cosine', 'storage_file': './shuihu_novel_kg/vdb_relationships.json'} 0 data
INFO:nano-vectordb:Init {'embedding_dim': 1024, 'metric': 'cosine', 'storage_file': './shuihu_novel_kg/vdb_chunks.json'} 0 data
INFO:lightrag:Loaded document status storage with 0 records
INFO:lightrag:Stored 1 new unique documents
INFO:lightrag:Number of batches to process: 1.
INFO:lightrag:Inserting 761 vectors to chunks
Generating embeddings: 54%
Level 2 - Extracting entities and relationships: 2%

13/24 [03:42<01:39, 9.03s/batch]
13/761 [03:33<1:07:36, 5.42s/chunk]

SSH: LazyTmp main* 0 0 0 0 Ln 31, Col 25 Spaces: 4 UTF-8 LF Python Signed out 3.10.0 ('lightrag': conda) Windsurf: 0

LightRAG基于知识图谱生成回答

```
1. async def initialize_rag():
2.     rag = LightRAG(
3.         working_dir=WORKING_DIR,
4.         embedding_func=EmbeddingFunc(
5.             embedding_dim=1024,
6.             max_token_size=8192,
7.             func=lambda texts: openai_embed(
8.                 texts=texts,model="bge-large",
9.                 base_url="http://127.0.0.1:19001",
10.                 api_key=api_key,)
11.             ),
12.         llm_model_func=openai_complete,
13.         llm_model_name="qwen2",
14.         llm_model_kwargs={"base_url":"http://0.0.0.0:12345/v1",
15.                             "api_key":api_key},
16.     )
17.     await rag.initialize_storages()
18.     await initialize_pipeline_status()
19.     return rag
20.
21.
22.
```

```
1. async def main():
2.     try:
3.         rag = await initialize_rag()
4.         with open(PATH_TO_TXT, "r", encoding="utf-8") as f:
5.             content = f.read()
6.             await rag.ainsert(content)
7.             mode = "hybrid"
8.             print(await rag.aquery("鲁智深打的是谁? ",
9. param=QueryParam(mode=mode)))
10.     except Exception as e:
11.         print(f"发生错误: {e}")
12.     finally:
13.         if rag:
14.             await rag.finalize_storages()
15.
```



LightRAG基于知识图谱生成回答

```
INFO: Merge E: 李忠 - 鲁提辖 | 2+0
INFO: Merge E: 史进 - 李忠 | 2+0
INFO: Merge E: 李忠 - 潘家酒楼 | 2+0
INFO: Merge E: 史进 - 潘家酒楼 | 2+0
INFO: Merge E: 东京 - 渭州 | 2+0
INFO: Merge E: 李忠 - 江湖上使枪棒卖药的 | 2+0
INFO: Merge E: 金老 - 鲁提辖 | 2+0
INFO: Merge E: 翠莲 - 鲁提辖 | 2+0
INFO: Merge E: 郑屠 - 鲁达 | 3+0
INFO: Merge E: 老种经略相公 - 鲁达 | 2+0
INFO: Merge E: 金翠莲 - 鲁达 | 2+0
INFO: Merge E: 王观察 - 鲁达 | 3+0
INFO: Updating 77 entities 1/1: unknown_source
INFO: Updating 81 relations 1/1: unknown_source
INFO: Writing graph with 80 nodes, 81 edges
INFO: In memory DB persist to disk
INFO: Completed processing file 1/1: unknown_source
INFO: Document processing pipeline completed
INFO: == LLM cache == saving hybrid: 50ad6b43ed6b6a5ea847a4b1fa339025
INFO: Process 2809 building query context...
INFO: Query nodes: 鲁智深, 镇关西, top_k: 60, cosine: 0.2
INFO: Local query uses 60 entites, 80 relations, 3 chunks
INFO: Query edges: Water Margin characters, Fictional conflicts, top_k: 60, cosine: 0.2
INFO: Global query uses 50 entites, 60 relations, 3 chunks
INFO: == LLM cache == saving hybrid: d4ec92db0e0695b3cdb8172d113f1f8a
鲁达（鲁提辖）打的是镇关西郑屠。鲁提辖因为正义感驱使，见义勇为帮助金翠莲和金老，他们受到镇关西郑屠的欺辱，因此鲁达在郑屠的肉铺前对其施以暴力，最终导致郑屠死亡。这一事件引发鲁达逃离渭州以躲避追捕。[KG] unknown_source [DC] unknown_source
INFO: Storage Finalization completed!
```



案例分析: 1

Query	鲁提辖是怎么死的？
Naive RAG	<p>根据提供的信息，我们无法直接得知鲁提辖（鲁智深）最终是如何死去的。资料中提到的部分主要是关于鲁提辖早年行侠仗义的经历、打死镇关西后的逃亡过程，以及后来成为花和尚鲁智深后因刺杀贺太守被擒的事件。但这些内容并未涉及他的死亡情况。从《水浒传》完整情节来看，鲁智深最后是在征方腊之后圆寂的。在战役结束之后，鲁智深预感到自己命数将尽，于浙江六和寺坐化（即佛教中的安然离世）。他留下了一首偈子表明自己的心境，并且显示了他对尘世已经毫无留恋，最终以一种平静且庄严的方式结束了他传奇的一生。不过，这一细节并未包含在您给出的知识段落内。</p>
LightRAG	<p>据提供的资料，鲁提辖（即后来出家为僧的鲁智深）并没有明确的死亡描述。相反，他的结局是在杭州六和寺坐化，也就是一种和尚修行到一定程度后自主选择的圆寂方式。这表明他最终选择了佛教的解脱之路，而非因战斗或其他原因去世。具体描述如下：- 鲁智深在《水浒传》中经历了多次战斗，展现了非凡的武艺和正义感。- 最终，他在杭州六和寺实现了从武士到觉悟和尚的转变。- 他在六和寺坐化，留下了颂词，体现了其放下执念和觉悟的精神转变。因此，鲁提辖的结局是通过坐化完成了一种精神上的升华，而不是因外界因素导致的死亡。</p>
分析	Naive RAG: 召回的内容都是82回之前含有鲁提辖的段落。



案例分析: 2

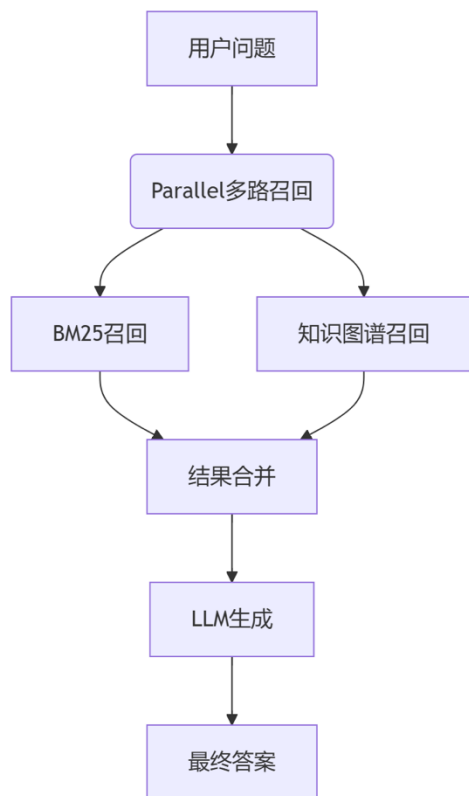
Query	林冲的主要事迹有哪些？
Naive RAG	根据您提供的知识内容，林冲的主要事迹包括：1. 被高太尉陷害，误入白虎节堂，被冤枉拿了刀进节堂意图刺杀高太尉，因此被逮捕。2. 被判处刺配，送往沧州。3. 在前往沧州的途中，经过野猪林，被押解的公人董超、薛霸想要杀死他，但被鲁智深所救。4. 到达沧州后，林冲被安排看守天王堂，并后来被指派去看管草料场。5. 高太尉派陆虞候到沧州谋害林冲，设计让草料场发生火灾并致林冲于死地。但林冲在听闻阴谋后避免了火灾，并杀死了想要陷害他的富安、陆谦以及差拨。6. 杀人之后，林冲戴罪上路，遇见了之前曾在东京救助过的李小二，李小二及其妻子在林冲落难时对他十分照顾。这些信息中描述了林冲几次非常关键的遭遇，最终导致他走上反抗的道路。
LightRAG	1. 身份转变：林冲原是东京八十万禁军的枪棒教头，因得罪高俅被陷害而被流放至沧州。在流放途中，受鲁智深的帮助，最终投奔了柴进，并后来加入了梁山泊。2. 加入梁山泊：林冲因遭受高俅的迫害而被迫逃离京城，寻求庇护于柴进处。之后，他前往梁山泊并成为梁山的重要头领之一。他还协助推翻了原来的首领王伦，立晁盖为新的领导人。3. 军事行动：- 参与攻打祝家庄的军事行动，作为第二拨的军事行动领导。- 在高唐州之战中直接与敌方交战。- 执行宋江命令，在宛州前线作战，是攻取宛州的关键人物之一。- 参与迎战辽国援军的战斗。- 在独松关的战斗中表现出色，成功地伤了蒋印。4. 个人英勇表现：- 林冲多次在重要战役中展现了他的英勇和忠诚，与多位将领如关胜、王焕以及贺统军等对决，并取得了显著的战绩。- 他参与了多次军事行动，包括寻找卢俊义、护送宿太尉进辽国颁诏和计划参与对辽国军队的攻击等。5. 患病与疗养：林冲后来染患风病瘫痪，留在六和寺中休养。6. 忠诚与影响力：尽管在面对招安时内心有所质疑，但林冲对梁山泊的贡献和在英雄中的地位是无可替代的。他的英勇和忠诚在《水浒传》中留下了一段传奇。 这些事迹展示了林冲从一名禁军教头到成为梁山重要头领的过程，以及他在多次战斗中的英勇表现和重要作用。
分析	Naive RAG：受限于topk，召回的原文主要是上梁山前的内容。回复中不够广泛，但涉及细节较多， LightRAG：召回的实体和关系偏重概括，回复中基本没有细节。



Query	武松帮助过谁?
Naive RAG	根据提供的信息，武松帮助过施恩。具体来说，施恩有一座酒肉店在城东快活林内，被蒋门神倚势豪强夺走。武松得知此事后，醉打了蒋门神，帮助施恩复夺了快活林。
LightRAG	<p>根据提供的信息，武松帮助过以下人物：</p> <ol style="list-style-type: none">施恩：<ul style="list-style-type: none">- 武松帮助施恩夺回了被蒋门神霸占的快活林酒店的经营权。这展现了武松是施恩的一位强有力的帮手。- 描述中提到：“武松帮助施恩夺回了快活林酒店的经营权，两人之间建立了信任和友谊的关系。”孔亮：<ul style="list-style-type: none">- 武松同意帮助孔亮，并计划一同攻打青州。此外，武松还帮助孔亮救了他的哥哥孔明和叔叔孔宾。张青和孙二娘：<ul style="list-style-type: none">- 虽然张青和孙二娘最初意图对武松不利，但后来他们成为朋友。武松在与他们的互动中表现出极大的正义感，间接帮助他们避免了一场误会。宋江：<ul style="list-style-type: none">- 在某些情节中，武松作为宋江的手下参与了多项军事行动，如解救李逵、参与分兵进攻东门等，间接帮助了宋江巩固梁山泊的力量。亲戚和邻舍：<ul style="list-style-type: none">- 武松通过安排饮酒答谢活动来接触和了解他的邻舍，维护和改善关系，体现了他对周围人的关心。- 例如，“武松邀请邻舍到家中饮酒答谢的事件，目的是安抚和进一步了解情况。”施恩的父亲（老管营）：<ul style="list-style-type: none">- 虽然没有明确提到武松直接帮助老管营，但通过帮助施恩，间接也使老管营受益。 <p>总结来说，武松的主要帮助对象包括施恩、孔亮、张青夫妇、宋江以及周围的邻居和亲友。他的行为体现了强烈的忠义精神和反抗冤屈的勇气。</p>
分析	Naive RAG：实际上这个CASE受益于LLM幻觉，因为召回的内容中未出来“夺回了被蒋门神霸占的快活林”。LightRAG：召回的广度更高。不过，其中 武松帮助了老管营略有牵强。（其属于召回的关系<施老管营, 施恩>：“施老管营是施恩的父亲，施恩得到武松的帮助夺回经营权，可能也会得到老管营的支持与协助。”



LazyLLM融合LightRAG



```
class LightRAGRetriever:
    def __init__(self, working_dir, txt_path, mode="hybrid"):
        self.working_dir = working_dir
        self.txt_path = txt_path
        self.mode = mode
        self.api_key = "empty"
        self.rag = None
        self.loop = asyncio.new_event_loop() # 创建独立事件循环
        setup_logger("lightrag", level="INFO")
        if not os.path.exists(working_dir):
            os.makedirs(working_dir)
        self.loop.run_until_complete(self.initialize_rag())

    async def initialize_rag(self):
        self.rag = LightRAG(
            working_dir=self.working_dir,
            embedding_func=EmbeddingFunc(
                embedding_dim=1024,
                max_token_size=8192,
                func=lambda texts: openai_embed(
                    texts=texts, model="bge-large",
                    base_url="http://127.0.0.1:19001",
                    api_key=self.api_key,
                )
            ),
            llm_model_func=openai_complete,
            llm_model_name="qwen2",
            llm_model_kwargs={"base_url": "http://0.0.0.0:12345/v1", "api_key": self.api_key},
        )
        await self.rag.initialize_storages()
        await initialize_pipeline_status()
        with open(self.txt_path, "r", encoding="utf-8") as f:
            content = f.read()
        await self.rag.ainsert(content)

    def __call__(self, query):
        return self.loop.run_until_complete(
            self.rag.aquery(query, param=QueryParam(mode=self.mode, only_need_context=True))
        )

    def close(self):
        if self.rag:
            try:
                self.loop.run_until_complete(self.rag.finalize_storages())
            except Exception as e:
                print(f"关闭LightRAG时出错: {e}")
            finally:
                self.loop.close()
                self.rag = None
```

```
def main():
    kg_retriever = None
    documents = lazyllm.Document(dataset_path="/home/mnt/yehongfei/Code/LightRAG/data")
    prompt = ('请参考所给的信息给出问题的答案。')
    print("正在初始化知识图谱检索器...")
    kg_retriever = LightRAGRetriever(
        working_dir="/home/mnt/yehongfei/Code/LightRAG/shuihu_kg",
        txt_path="/home/mnt/yehongfei/Code/LightRAG/data/水浒传.txt"
    )
    print("知识图谱检索器初始化完成!")

    with lazyllm.pipeline() as ppl:
        with parallel() as ppl.multi_retrieval:
            bm25_retriever = lazyllm.Retriever(doc=documents,
                                                group_name="CoarseChunk",
                                                similarity="bm25_chinese", topk=3)

            def bm25_pipeline(query):
                nodes = bm25_retriever(query)
                return "".join([node.get_content() for node in nodes])

            ppl.multi_retrieval.bm25 = bm25_pipeline
            ppl.multi_retrieval.kg = kg_retriever

        def context_combiner(*args):
            bm25_result = args[0]
            kg_result = args[1]
            return (f"知识图谱召回结果:\n(kg_result)"
                    f"BM25召回结果:\n(bm25_result)\n\n")

        ppl.context_combiner = context_combiner
        ppl.formatter = (lambda nodes, query: dict(context_str=nodes, query=query)) | bind(query=ppl.input)

    ppl.llm = (
        lazyllm.OnlineChatModule().prompt(lazyllm.ChatPrompter(instruction=prompt,
                                                                extra_keys=['context_str']))
    )
    lazyllm.WebModule(ppl, port=23466).start().wait()
```



LazyLLM融合LightRAG

```
CODE / LightRAG / lazy_light_rag.py
11 class LightRAGRetriever:
12     def __init__(self, working_dir, txt_path, mode="hybrid"):
24         # 在独立事件循环中初始化RAG
25         self.loop.run_until_complete(self.initialize_rag())
26
27     async def initialize_rag(self):
28         self.rag = LightRAG(
29             working_dir=self.working_dir,
30             embedding_func=EmbeddingFunc(
31                 embedding_dim=1024,
32                 max_token_size=8192,
33                 func=lambda texts: openai_embed(
34                     texts=texts,
35                     model="bge-large",
36                     base_url="http://127.0.0.1:19001",
37                     api_key=self.api_key,
38                 )
39             ),
40             llm_model_func=openai_complete

问题 输出 调试控制台 终端 3
> 终端
(lazyllm) → LazyLLM git:(main) X infinity_emb v2 --model-id "/mnt/lustre/share_data/lazyllm/models/bge-large-zh-5" --port 19001 --served-model-name bge-large
```



朴素RAG召回文本：

又只一拳，太阳穴上正着，却似做了一个全堂水陆的道场，磬儿、钹儿、铙儿，一齐响。鲁达看时，只见郑屠挺在地下，口里只有出的气，没有入的气，动弹不得。鲁提辖假意道：“你这厮诈死，洒家再打。”只见面皮渐渐的变了。鲁达寻思道：“俺只指望痛打这厮一顿，不想三拳真个打死了他，。洒家须吃官司，又没人送饭，不如及早撒开。”拔步便走，回头指着郑屠尸道：“你诈死，洒家和你慢慢理会。”一头骂，一头大踏步去了。街坊邻舍并郑屠的火家，谁敢向前来拦他。鲁提辖回到下处，急急卷了一些衣服盘缠、细软银两，但是旧衣粗重都弃了。提了一根齐眉短棒，奔出南门，一道烟走了。且说郑屠家中众人，救了半日不活，呜呼死了。老小邻人径来州衙告状。正直府尹升厅，接了状子，看罢道：“鲁达系是经略府的提辖，不敢擅自径来捕捉凶身。”府尹随即上轿，来到经略府前，下了轿子。把门军士入去报知，府中听得，教请到厅上，与府尹施礼罢。经略问道：“何来？”府尹禀道：“好教相公得知。府中提辖鲁达，无故用拳打死市上郑屠。不曾禀过相公，不敢擅自捉拿凶身。”经略听说，吃了一惊，寻思道：“这鲁达虽好武艺，只是性格粗卤，今番做出人命事，俺如何护得短？须教他推问使得。”经略回府尹道：“鲁达这人，原是我父亲老经略处的军官，为因俺这里无人帮护，拨他来做个提辖。既然犯了人命罪过，你可拿他依法度取问。如若供招明白，拟罪已定，也须教我父亲知道，方可断决。怕日后父亲处边上要这个人时，却不好看。”府尹禀道：“下官问了情由，合行申禀老经略相公知道，方可断遣。”府尹辞了经略相公，出到府前，上了轿，回到州衙里，措厅坐下。便唤当日缉捕使臣押下文书，捉拿犯人鲁达。当时王观察领了公文，将带二十来个做公的人，径到鲁提辖下处。但见：扶肩搭背，交颈并头。纷纷不辨贤愚，扰扰难分贵贱。张三蠢胖，不识字只把头摇；李四矮矬，看别人也将脚踏。白头老叟，尽将拐棒拄髭须；绿鬓书生，却把文房抄款目。行行总是萧何法，句句俱依律令行。鲁达看见众人看榜，挨满在十字路口，也站在人丛里。听时，鲁达却不识字，只听得众人读道：“代州雁门县依奉太原府指挥使司，该准渭州文字，捕捉打死郑屠犯人鲁达，即系经略府提辖。如有人停藏在家宿食，与犯人同罪；若有人捕获前来，或告到官，支給赏钱一千贯文。”鲁提辖正听到那里，只听得背后有人大叫道：“张大哥。你如何到这里？”拦腰抱住，扯离了十字路口。不是这个人看见了，横倒拖拽将去，有分教：鲁提辖剃除头发，削去髭须，倒换过杀人姓名，薙恼杀诸佛罗汉。直教：禅杖打开危险路，戒刀杀尽不平人。毕竟扯住鲁提辖的是甚人，且听下回分解。若非雨病云愁，定是怀忧积恨。那妇人拭着眼泪，向前来深深的道了三个万福。那老儿也都相见了。鲁达问道：“你两个是那里人家？为甚啼哭？”那妇人便道：“官人不知，容奴告禀：奴家是东京人氏。因同父母来这渭州，投奔亲眷，不想搬移南京去了。母亲在客店里染病身故，子父二人，流落在此生受。此间有个财主，叫做镇关西郑大官人，因见奴家，便使强媒硬保，要奴作妾。谁想写了三千贯文书，虚钱实契，要了奴家的身体，未及三个月，他家大娘子好生利害，将奴赶打出来，不容完聚，着落店主人家追要原典身钱三千贯。父亲懦弱，和他争执不得，他又有钱有势。当初不曾得他一文，如今那讨些钱还他？没计奈何，父亲自小教得奴家些小曲儿，来到这里酒楼上赶座。每日但得些钱来，将大半还他，留些少子父们盘缠。这两日酒客稀少，违了他钱限，怕他来时，受他羞耻。子父们想起这些苦楚来，无处告诉，因此啼哭。不想误触犯了官人，望乞恕罪，高抬贵手。”鲁提辖又问道：“你姓甚么，在那个客店里歇？那个镇关西郑大官人在那里住？”老儿答道：“老汉姓金，排行第二。孩儿小字翠莲。郑大官人便是此间状元桥下卖肉的郑屠，绰号镇关西。老汉父子两个，只在前面东门里鲁家客店安下。”鲁达听了道：“呸！俺只道那个郑大官人，却原来是杀猪的郑屠。这个腌臢泼才，投托着俺小种经略相公门下做个肉铺户，却原来这等欺负人！”回头看李忠、史进道：“你两个且在这里，等洒家去打死了那厮便来。”史进、李忠抱住劝道：“哥哥息怒，明日却理会。”两个三回五次劝得他住。鲁达又道：“老儿，你来！洒家与你些盘缠，时日便回东京去如何？”父子两个告道：“若是能够回乡去时，便是重生父母，再长爷娘。只是店主人家如何肯放？郑大官人须着落他要钱。”鲁提辖道：“这个不妨事，俺自有道理。”



LazyLLM融合LightRAG



LightRAG召回内容(Entities部分):

-----Entities(KG)-----

```
```json[{"id": "1", "entity": "金翠莲", "type": "person", "description": "金翠莲是金老的女儿，受到郑屠的不公平对待。<SEP>金翠莲是鲁达提到的一个女性，据推测她可能是被郑屠欺骗或伤害的对象。", "rank": 3, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "2", "entity": "鲁达", "type": "person", "description": "鲁达是一个强力的武官，积极帮助金老摆脱困境并惩罚了郑屠。<SEP>鲁达是一名经略府的提辖，以其武艺高强和性格粗犷著称，在故事中因打死郑屠而逃走。<SEP>鲁达是经略府的提辖，曾与史进在茶坊相遇，并愿意帮助史进寻找他的师父。<SEP>鲁达，也称鲁提辖，因打死郑屠而被捉拿在逃，是故事中的主要人物。", "rank": 18, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "3", "entity": "两个都头", "type": "person", "description": "两个都是官府中的官员，负责抓捕史进等人。", "rank": 2, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "4", "entity": "史进", "type": "person", "description": "史进是一位与鲁提辖一起行动的个人，也被称为史大郎，其师父是打虎将李忠。<SEP>史进是一位有义气的英雄，他在保护朱武三人免受官府逼迫的过程中，采取了紧张的行动，并最终决定离开家乡去寻找他的师父。<SEP>史进是一名清白的好汉，拒绝落草为寇，离开少华山，寻找他的师父王教头。<SEP>史进是与鲁提辖一同饮酒的同伴之一，提供了帮助金氏父女的一部分钱财。", "rank": 17, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "5", "entity": "杨春", "type": "person", "description": "杨春是史进朋友中的另一位头领，积极参与史进保护朱武等人并对抗官府的行动。", "rank": 1, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "6", "entity": "事件: 鲁达还房钱", "type": "event", "description": "鲁达提出支付郑屠的钱，以释放金老。", "rank": 0, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "7", "entity": "代州雁门县", "type": "geo", "description": "代州雁门县是鲁达逃亡过程中到达的一座城市。", "rank": 1, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "8", "entity": "事件公告", "type": "event", "description": "州衙厅发布的公告，悬赏捉拿鲁提辖的信息在代州雁门县发布。", "rank": 2, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}]
```

## LightRAG召回内容(Relationships部分):

```
-----Relationships(KG)-----```json[{"id": "1", "entity1": "金翠莲", "entity2": "鲁达", "description": "鲁达为了帮助金翠莲及其父亲而采取行动。<SEP>鲁达提及金翠莲，认为她是被郑屠欺骗的受害者。", "keywords": "受害者保护, 忠义, 救助", "weight": 14.0, "rank": 21, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "2", "entity1": "两个都头", "entity2": "史进", "description": "两个都头代表官府来抓捕史进，与史进及其同伙发生对抗。", "keywords": "对抗, 抓捕", "weight": 9.0, "rank": 19, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "3", "entity1": "史进", "entity2": "杨春", "description": "史进和杨春共同行动以对抗官府并保护山寨成员。", "keywords": "合作, 联盟", "weight": 8.0, "rank": 18, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "4", "entity1": "事件公告", "entity2": "鲁达", "description": "公告号召人们协助捉拿鲁提辖。", "keywords": "司法通缉", "weight": 9.0, "rank": 20, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "5", "entity1": "代州雁门县", "entity2": "鲁达", "description": "鲁达在逃亡过程中到达代州雁门县。", "keywords": "逃亡途点", "weight": 9.0, "rank": 19, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "6", "entity1": "事件公告", "entity2": "州衙厅", "description": "州衙厅发布有关鲁达的逃犯公告。", "keywords": "司法公告", "weight": 7.0, "rank": 4, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}]
```





## LightRAG召回内容(Document Chunks部分):

-----Document Chunks(DC)-----

```
```json[{"id": "1", "content": "上。鲁达再入一步，踏住胸脯，提着那醋钵儿大小拳头，看着这郑屠道：\n\n“洒家始投老种经略相公，做到关西五路廉访使，也不枉了叫做镇关西。你是个卖肉的操刀屠户，狗一般的人，也叫做镇关西！你如何强骗了金翠莲？”扑的只一拳，正打在鼻子上，打得鲜血迸流，鼻子歪在半边，却便似开了个油酱铺，咸的、酸的、辣的，一发都滚了出来。郑屠挣不起来，那把尖刀也丢在一边，口里只叫：“打得好！”鲁达骂道：“直娘贼，还敢应口！”提起拳头，就眼眶际眉梢只一拳，打得眼棱缝裂，乌珠迸出，也似开了个彩帛铺的，红的、黑的、绛的，都绽将出来。两边看的人，惧怕鲁提辖，谁敢向前来劝？\n\n郑屠当不过，讨饶。鲁达喝道：“咄！你是个破落户，若是和俺硬到底，洒家倒饶了你；你如何对俺讨饶，洒家偏不饶你。”又只一拳，太阳穴上正着，却似做了一个全堂水陆的道场，磬儿、钹儿、铙儿，一齐响。鲁达看时，只见郑屠挺在地下，口里只有出的气，没有入的气，动弹不得。鲁提辖假意道：\n\n“你这厮诈死，洒家再打。”只见面皮渐渐的变了。\n\n鲁达寻思道：“俺只指望痛打这厮一顿，不想三拳真个打死了他，。洒家须吃官司，又没人送饭，不如及早撒开。”拔步便走，回头指着郑屠尸道：“你诈死，洒家和你慢慢理会。”\n\n一头骂，一头大踏步去了。街坊邻舍并郑屠的火家，谁敢向前来拦他。鲁提辖回到下处，急急卷了一些衣服盘缠、细软银两，但是旧衣粗重都弃了。提了一根齐眉短棒，奔出南门，一道烟走了。\n\n且说郑屠家中众人，救了半日不活，呜呼死了。老小邻人径来州衙告状。正直府尹升厅，接了状子，看罢道：“鲁达系是经略府的提辖，不敢擅自径来捕捉凶身。”府尹随即上轿，来到经略府前，下了轿子。把门军士入去报知，府中听得，教请到厅上，与府尹施礼罢。经略问道：“何来？”府尹禀道：“好教相公得知。府中提辖鲁达，无故用拳打死市上郑屠。不曾禀过相公，不敢擅自捉拿凶身。”经略听说，吃了一惊，寻思道：“这鲁达虽好武艺，只是性格粗卤，今番做出人命事，俺如何护得短？须教他推问使得。”经略回府尹道：“鲁达这人，原是我父亲老经略处的军官，为因俺这里无人帮护，拨他来做个提辖。既然犯了人命罪过，你可拿他依法度取问。如若供招明白，拟罪已定，也须教我父亲知道，方可断决。怕日后父亲处边上要这个人时，却不好看。”府尹禀道：“下官问了情由，合行申禀老经略相公知道，方可断遣。”府尹辞了经略相公，出到府前，上了轿，回到州衙里，措厅坐下。便唤当日缉捕使臣押下文书，捉拿犯人鲁达。\n\n当时王观察领了公文，将带二十来个做公的人，径到鲁提辖下处。只见房主人道：“却才？了些包裹，提了短棒出去了。小人只道奉着差使，又不敢问他。”王观察听了，教打开他房门看时，只有些旧衣旧裳和些被卧在里面。王观察就带了房主人，东西四下里去跟寻，州南走到州北，捉拿不见。王观察又捉了两家邻舍并房主人，同到州衙厅上回话道：“鲁提辖惧罪在逃，不知去向，只拿得房主人并邻舍在此。”府尹见说，且教监下；一面教拘集郑屠家邻佑人等，点了件作”，“file_path": "unknown_source"}], .....
```



RAG多路召回合并后输入给LLM的信息:

```
{'messages': [{'role': 'system', 'content': 'You are an AI assistant, developed by SenseTime.\n请你参考所给的信息给出问题的答案.\nHere are some extra messages you can referred to:\n\n### context_str:\n知识图谱召回结果:\n-----Entities(KG)-----\n\n\n```\njson\n[{"id": "1", "entity": "金翠莲", "type": "person", "description": "金翠莲是金老的女儿, 受到郑屠的不公平对待。<SEP>金翠莲是鲁达提到的一个女性, 据推测她可能是被郑屠欺骗或伤害的对象。", "rank": 3, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "2", "entity": "鲁达", "type": "person", "description": "鲁达是一个强力的武官, 积极帮助金老摆脱困境并惩罚了郑屠。<SEP>鲁达是一名经略府的提辖, 以其武艺高强和性格粗犷著称, 在故事中因打死郑屠而逃走。<SEP>鲁达是经略府的提辖, 曾与史进在茶坊相遇, 并愿意帮助史进寻找他的师父。<SEP>鲁达, 也称鲁提辖, 因打死郑屠而被捉拿在逃, 是故事中的主要人物。", "rank": 18, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, {"id": "3", "entity": "两个都头", "type": "person", "description": "两个都是官府中的官员, 负责抓捕史进等人。", "rank": 2, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, .....]\n\n```\n\n\n-----Relationships(KG)-----\n\n\n```\njson\n[{"id": "1", "entity1": "金翠莲", "entity2": "鲁达", "description": "鲁达为了帮助金翠莲及其父亲而采取行动。<SEP>鲁达提及金翠莲, 认为她是被郑屠欺骗的受害者。", "keywords": "受害者保护, 忠义, 救助", "weight": 14.0, "rank": 21, "created_at": "2025-06-10 00:06:37", "file_path": "unknown_source"}, .....]\n\n\n\n-----Document Chunks(DC)-----\n\n\n```\njson\n[{"id": "1", "content": "上。鲁达再入一步, 踏住胸脯, 提着那醋钵儿大小拳头, 看着这郑屠道: \\n\\n    “洒家始投老种经略相公, 做到关西五路廉访使, 也不枉了叫做镇关西。你是个卖肉的操刀屠户, 狗一般的人, 也叫做镇关西! 你如何强骗了金翠莲?” 扑的只一拳, 正打在鼻子上, 打得鲜血迸流, 鼻子歪在半边, 却便似开了个油酱铺, 咸的、酸的、辣的, 一发都滚了出来。郑屠挣不起来, 那把尖刀也丢在一边, 口里只叫: “打得好!” 鲁达骂道: “直娘贼, 还敢应口!” 提起拳头, 就眼眶际眉梢只一拳, 打得眼棱缝裂, 乌珠迸出, 也似开了个彩帛铺的, 红的、黑的、绛的, 都绽将出来。两边看的人, 惧怕鲁提辖, 谁敢向前来劝? \\n\\n    郑屠当不过, 讨饶。鲁达喝道: “咄! 你是个破落户, 若是和俺硬到底, 洒家倒饶了你; 你如何对俺讨饶, 洒家偏不饶你。” 又只一拳, 太阳穴上正着, 却似做了一个全堂水陆的道场, 磬儿、钹儿、铙儿, 一齐响。鲁达看时, 只见郑屠挺在地下, 口里只有出的气, 没有入的气, 动弹不得。鲁提辖假意道: \\n\\n    “你这厮诈死, 洒家再打。” 只见面皮渐渐的变了。\\n\\n    鲁达寻思道: “俺只指望痛打这厮一顿, 不想三拳真个打死了他, 。洒家须吃官司, 又没人送饭, 不如及早撒开。” 拔步便走, 回头指着郑屠尸道: “你诈死, 洒家和你慢慢理会。” \\n\\n    一头骂, 一头大踏步去了。街坊邻舍并郑屠的火家, 谁敢向前来拦他。鲁提辖回到下处, 急急卷了一些衣服盘缠、细软银两, 但是旧衣粗重都弃了。提了一根齐眉短棒, 奔出南门, 一道烟走了。\\n\\n    且说郑屠家中众人, 救了半日不活, 呜呼死了。.....]\n\n\n\n\nBM25召回结果:\n” 又只一拳, 太阳穴上正着, 却似做了一个全堂水陆的道场, 磬儿、钹儿、铙儿, 一齐响。鲁达看时, 只见郑屠挺在地下, 口里只有出的气, 没有入的气, 动弹不得。鲁提辖假意道: \\r\\n\\r\\n    “你这厮诈死, 洒家再打。” 只见面皮渐渐的变了。\\r\\n\\r\\n    鲁达寻思道: “俺只指望痛打这厮一顿, 不想三拳真个打死了他, 。洒家须吃官司, 又没人送饭, 不如及早撒开。” 拔步便走, 回头指着郑屠尸道: “你诈死, 洒家和你慢慢理会。” \\r\\n\\r\\n    一头骂, 一头大踏步去了。街坊邻舍并郑屠的火家, 谁敢向前来拦他。鲁提辖回到下处, 急急卷了一些衣服盘缠、细软银两, 但是旧衣粗重都弃了。提了一根齐眉短棒, 奔出南门, 一道烟走了。\\r\\n\\r\\n    .....}, {'role': 'user', 'content': '鲁智深打死的是谁?'}], 'model': 'SenseChat-5', 'stream': True}
```



你的问题需要哪种RAG? —— 小结



1. 朴素RAG

通过向量检索的方式，能够快速、有效地匹配相关文本片段。

- 适用于处理大量非结构化文本数据
- 能够从大规模数据中迅速提取关键信息

2. 基于知识图谱的RAG

通过构建知识图谱，在查询过程中深入理解实体之间的关系。

- 提供更丰富、准确的上下文支持
- 确保查询结果的完整性和关联性

3. 基于混合检索的RAG系统

将片段检索与知识图谱检索相结合，实现优势互补。

- 融合语义相似度与结构化关系理解
- 提升在复杂任务中的检索质量与生成效果



感谢聆听
Thanks for Listening

