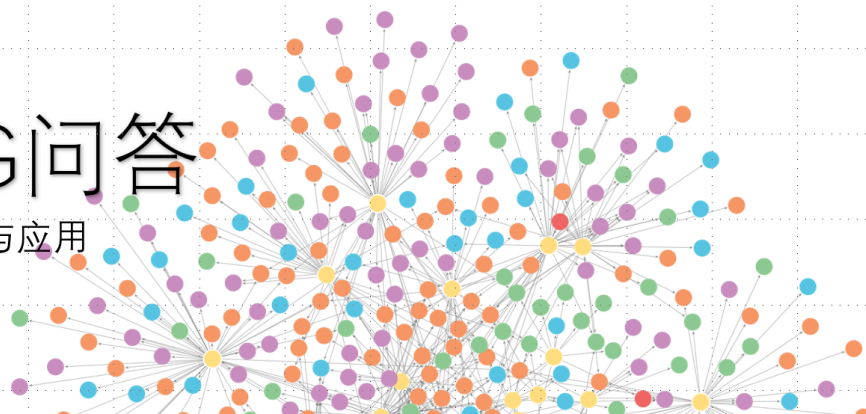


# Datawhale X MarsCode

## 知识图谱RAG问答

知识图谱+LLM的RAG探索与应用



## 知识图谱、RAG是什么？？？

**知识图谱** (Knowledge Graph) 是一种用于表示和组织知识的图形化表示方法。它由**节点** (Entities) 和**边** (Relationships) 组成，用于表示实体之间的**关系**。知识图谱可以用于各种应用，例如信息检索、问答系统、推荐系统等。

**RAG** (Retrieval-Augmented Generation) 是一种基于检索增强生成的技术，它通过**检索**相关的知来**增强**LLM的能力。RAG的核心思想是将LLM与外部知识源进行融合，以提高LLM的**生成能力**。

## 目录

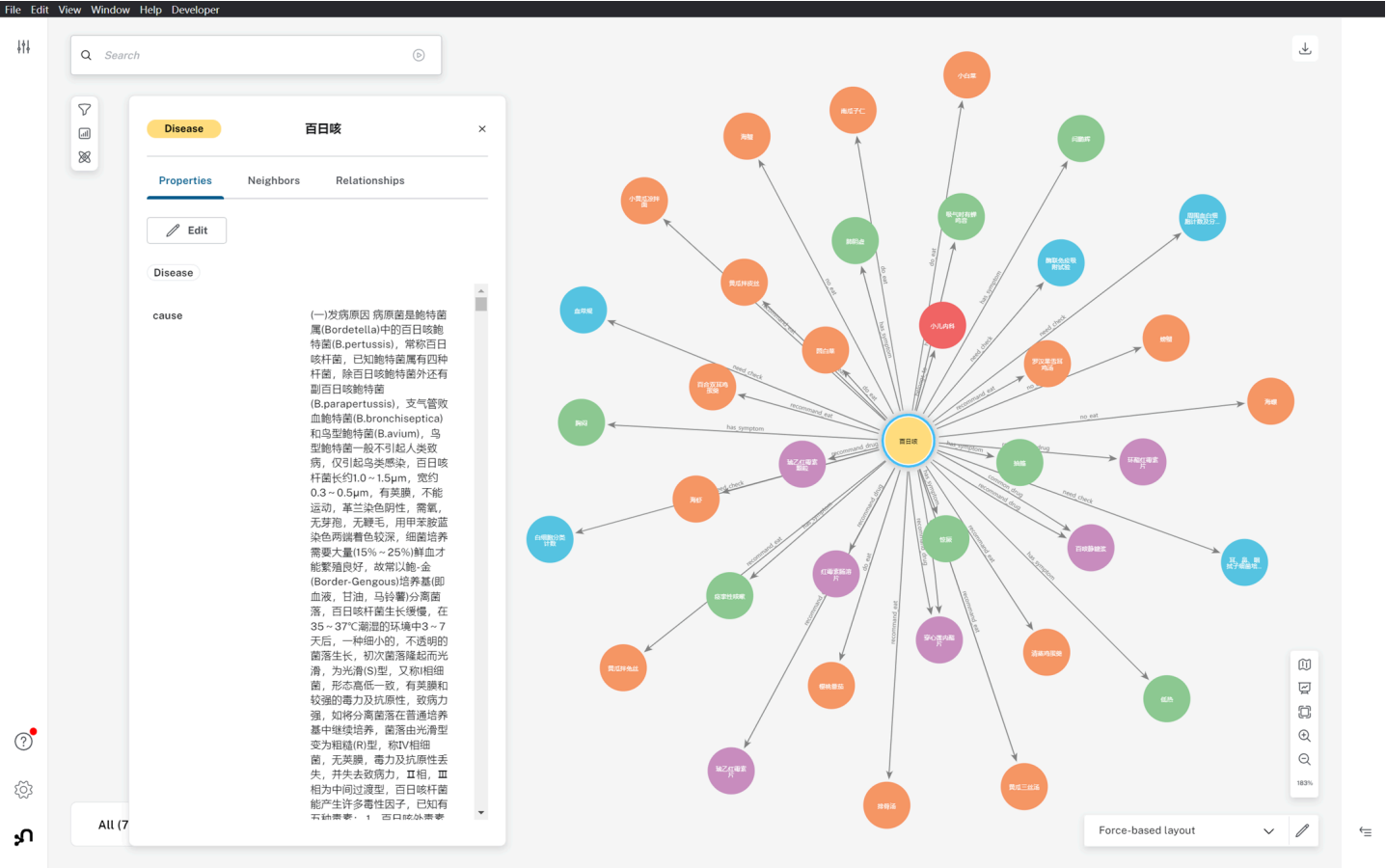
- [项目背景](#)
- [Demo展示](#)
- [应用价值](#)
- [技术实现](#)
- [AI协作](#)
- [团队介绍](#)

## 项目背景

近年来，大模型 (LLM) 在各个领域取得了许多重大成果。但是大模型有时会有答非所问的情况 (幻觉)，RAG能够缓解幻觉问题。然而基于文档RAG难以保持如医疗领域、汽车领域、农业领域专业知识具有的自身结构性。为了解决以上问题，提出了一种新的知识增强方法。该方法由**知识图谱**和**RAG**结合prompt工程组成**GRAG**框架。

# Demo展示

以医疗领域为例构建的知识图谱



用户提问

question = "我的孩子得了百日咳，经常咳嗽，应该吃什么？"

检索知识图谱

```
('百日咳', '海蟹', {'type': {'properties': {'name': '忌吃'}, 'type': 'no_eat'}}),
('百日咳', '螃蟹', {'type': {'properties': {'name': '忌吃'}, 'type': 'no_eat'}}),
('百日咳',
 '黄瓜拌皮丝',
 {'type': {'properties': {'name': '推荐食谱'}, 'type': 'recommand_eat'}}),
('百日咳',
 '黄瓜拌兔丝',
 {'type': {'properties': {'name': '推荐食谱'}, 'type': 'recommand_eat'}}),
('百日咳',
 '黄瓜三丝汤',
 {'type': {'properties': {'name': '推荐食谱'}, 'type': 'recommand_eat'}}),
('百日咳',
```

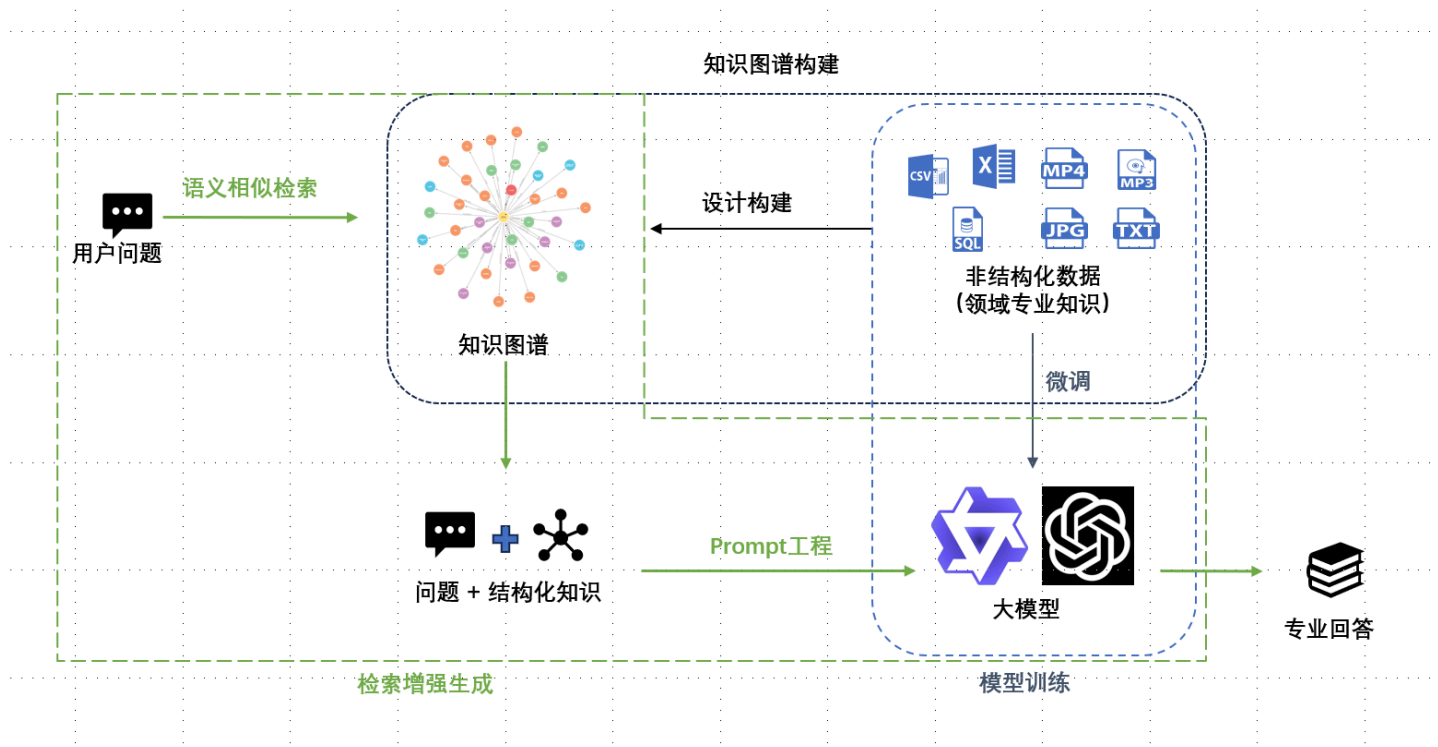
模型回答

```
('对于百日咳患者，建议食用以下食物：\n'\n'\n'\n'* **小白菜**\n'\n'* **樱桃番茄**\n'\n'* **圆白菜**\n'\n'* **南瓜子仁**\n'\n'\n'\n'此外，还有一些推荐的食谱，如：\n'\n'\n'\n'* **黄瓜拌皮丝**\n'\n'* **黄瓜拌兔丝**\n'\n'* **黄瓜三丝汤**\n'\n'* **小黄瓜凉拌面**\n'\n'* **罗汉果雪耳鸡汤**\n'\n'* **排骨汤**\n'\n'* **百合双耳鸡蛋羹**\n'\n'* **清蒸鸡蛋羹**\n'\n'\n'\n'这些食物和食谱可能对缓解百日咳的症状有所帮助。同时，请确保遵循医生的治疗建议，因为食物并不能替代药物治疗。')\n{'completion_tokens': 137, 'prompt_tokens': 9031, 'total_tokens': 9168}
```

# 应用价值

- 医疗疾病药物关系问答
- 刑侦人物关系推理
- 社交网络推荐系统

# 技术实现



# AI协作

## 知识图谱构建

使用MarsCode辅助编写爬虫脚本获取数据，再结合AI辅助构建知识图谱。利用MarsCode编写cypher语句将构建的知识图谱保存到Neo4j中进行可视化展示。

MARSCODE AI

今日宜吃串串

可以教教我cypher语言吗？如何创建节点，如何连接两个节点等

MarsCode AI

AI is thinking..

> 引用 1 相关信息

Cypher是Neo4j图数据库的查询语言，用于创建、查询和操作图数据。以下是一些基本的Cypher语句示例，用于创建节点和连接节点：

创建节点

1. 

readme.md

readme.md

CREATE (n:Person {name: 'Alice', age: 30})

CREATE

：用于创建新的节点或关系。

(n:Person)

： 

n

 是节点的别名， 

Person

 是节点的标签，表示该节点属于 

Person

 类型。

{name: 'Alice', age: 30}

：节点的属性，这里定义了 

name

 和 

age

 两个属性。

2. 创建多个节点

readme.md

CREATE (n1:Person {name: 'Bob', age: 25}), (n2:Person {name: 'Charlie', age: 35})

可以在一个 

CREATE

 语句中创建多个节点，每个节点用逗号分隔。

连接节点

1. 创建关系

readme.md

MATCH (a:Person {name: 'Alice'}), (b:Person {name: 'Bob'})  
CREATE (a)-[r:KNOWS]->(b)

MATCH

：用于查找符合条件的节点

AI 思考中...

※ #

## RAG框架

使用到langchain、torch、transformers等库进行RAG框架的搭建，其中的用法借助MarsCode进行解释和编写样例参考。同时AI辅助编写一些相关算法如相似度算法、重排算法等，算法细节也能使用AI辅助进行学习和优化。

## MarsCode AI

[➤ 引用 1 相关信息](#)

在LangChain中，Neo4j是一个图数据库，它可以用来存储和查询知识图谱。LangChain提供了一些工具和类来与Neo4j进行交互，例如 `Neo4jGraph` 类和 `GraphCypherQAChain` 类。

以下是一个使用LangChain与Neo4j进行交互的示例代码：

python

```
from langchain.graphs import Neo4jGraph
from langchain.chains import GraphCypherQAChain
from langchain.chat_models import ChatOpenAI

# 初始化Neo4j图
graph = Neo4jGraph(
    url="bolt://localhost:7687",
    username="neo4j",
    password="your_password"
)

# 初始化ChatOpenAI模型
llm = ChatOpenAI(temperature=0)

# 初始化GraphCypherQAChain
chain = GraphCypherQAChain.from_llm(llm, graph=graph, verbose=True)

# 执行查询
result = chain.run("MATCH (p:Person) RETURN p.name LIMIT 5")

print(result)
```

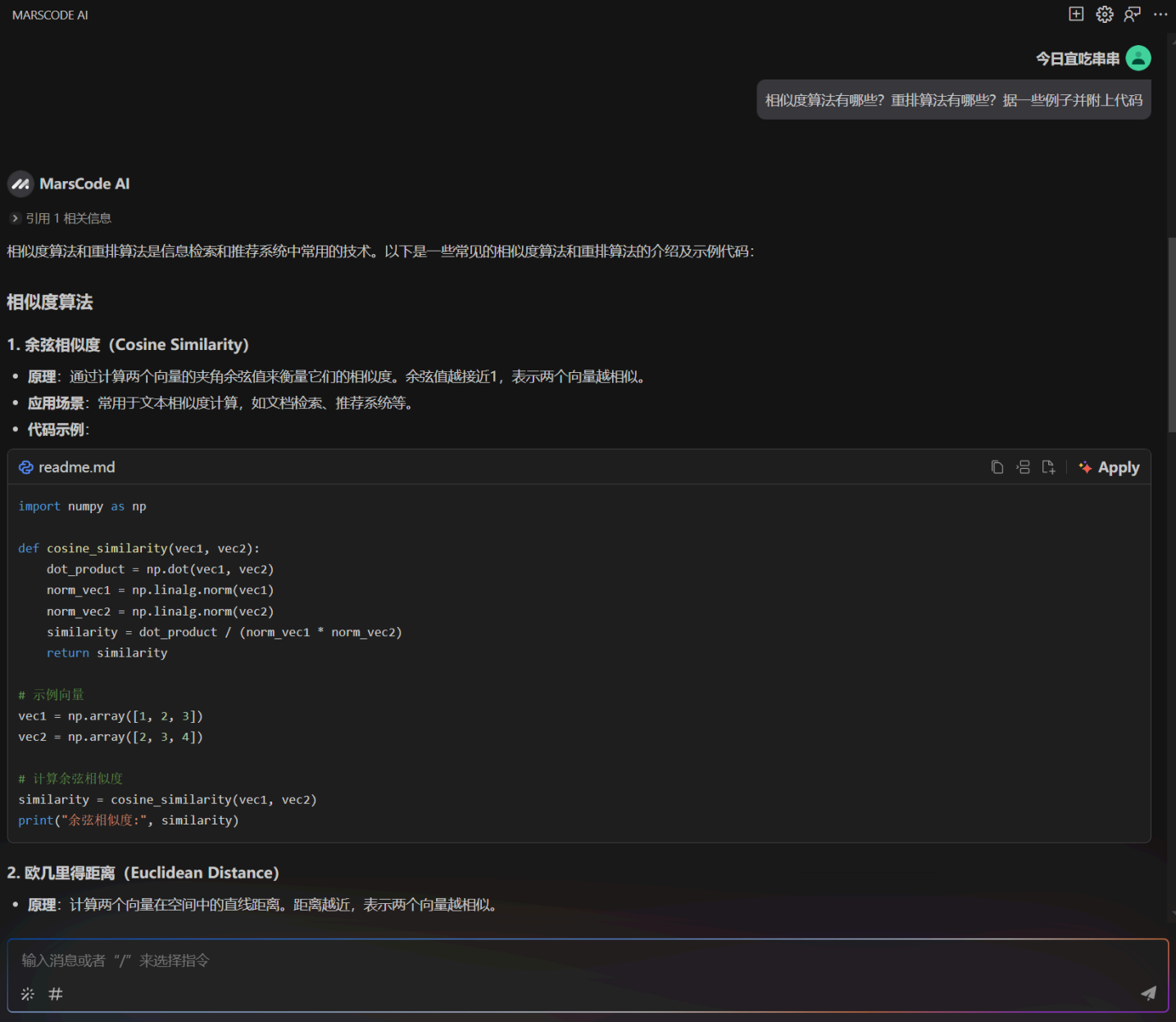
在这个示例中，我们首先初始化了一个 `Neo4jGraph` 对象，它连接到了本地的Neo4j数据库。然后，我们初始化了一个 `ChatOpenAI` 模型，并使用它来初始化一个 `GraphCypherQAChain` 对象。最后，我们使用 `GraphCypherQAChain` 对象来执行一个Cypher查询，并打印查询结果。

请注意，你需要将 `url`、`username` 和 `password` 替换为你自己的Neo4j数据库的连接信息。

输入消息或者 "/" 来选择指令

※ #





## 后续idea

借助MarsCode进行网站开发，实现用户提问、知识图谱检索、模型回答以及图谱可视化等功能。

## 团队介绍

10班-专业助教-周亮

- 信计专业 - 大四在读
- 相关经验 - 算法实习生
- 联系方式 - 我在10班学习群上，有感兴趣的小伙伴欢迎随时打扰！