

基于LangChain与大模型的医疗问答系统研究

闵 瑞, 姜 丹*, 王永刚

北京印刷学院信息工程学院, 北京

收稿日期: 2025年1月7日; 录用日期: 2025年2月10日; 发布日期: 2025年2月20日

摘 要

随着人工智能技术的迅猛发展, 医疗问答系统已成为医疗信息检索和知识获取的重要工具。医疗领域涉及大量医学术语、复杂的疾病症状和治疗方案, 传统查询方式难以高效、准确地满足医护人员和患者的信息需求。相比传统国内搜索引擎和原生开源大语言模型(LLMs), 基于LangChain的大模型医疗问答系统能够提供更高质量的答案, 显著提升医疗知识检索的效率和精准度。因此, 本研究提出了一种基于LangChain与大模型的医疗智能问答系统, 结合命名实体识别(NER)、图谱查询和对话分析等技术, 构建了一个专注于医疗领域的知识图谱及其查询与生成模块。通过设计和优化Prompt提示词, Agent Tool提升了大模型生成更精准、高质量医疗问答的能力。研究结果表明, 该系统在医疗问答任务中的表现优异, 准确度、方案可行性和上下文相关性等指标显著优于传统LLMs和国内知名大模型。该系统通过与大规模医疗知识图谱的结合, 能够深入理解复杂的医疗问题, 并提供精准的回答, 呈现可视化图谱展示图, 更直观地给用户反馈, 同时具备较高的数据安全性和可迁移性。

关键词

LangChain, 知识图谱, 大模型

Research on Medical Question Answering System Based on LangChain and Large Model

Rui Min, Dan Jiang*, Yonggang Wang

School of Information Engineering, Beijing Institute of Graphic Communication, Beijing

Received: Jan. 7th, 2025; accepted: Feb. 10th, 2025; published: Feb. 20th, 2025

Abstract

Nowadays, with the rapid development of artificial intelligence technology, medical question answering

*通讯作者。

文章引用: 闵瑞, 姜丹, 王永刚. 基于 LangChain 与大模型的医疗问答系统研究[J]. 计算机科学与应用, 2025, 15(2): 33-43. DOI: 10.12677/csa.2025.152031

system has become an important tool for medical information retrieval and knowledge acquisition. The medical field involves a large number of medical terms, complicated disease symptoms and treatment plans, and traditional inquiry methods are difficult to meet the information needs of medical staff and patients efficiently and accurately. Compared with traditional domestic search engines and native open source large language model (LLMs), LangChain-based large model medical question answering system can provide higher quality answers, significantly improving the efficiency and accuracy of medical knowledge retrieval. Therefore, this study proposed a medical intelligent question and answer system based on LangChain and large model, combined with named entity recognition (NER), graph query and dialogue analysis and other technologies, to build a knowledge graph and query and generation module focusing on the medical field. By designing and optimizing Prompt words, Agent Tool improves the ability of large models to generate more accurate and high-quality medical questions and answers. The results show that the system performs well in medical question answering tasks, with significant improvements in accuracy, feasibility, and context relevance are significantly better than traditional LLMs and well-known domestic large models. Through the combination of large-scale medical knowledge graph, the system can deeply understand complex medical questions, provide accurate answers, present a visual map display graph, and give users more intuitive feedback, while having high data security and portability.

Keywords

LangChain, Knowledge Graph, Large Model

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,生成式人工智能(AIGC)技术得到了广泛应用,智能问答系统取得了显著进展。尽管大语言模型在文本理解与生成上表现优异,但其在处理超出训练数据范围的问题时仍然面临局限,且生成的答案可能存在准确性问题,尤其在医疗领域中,模型生成答案不准确,联系上下文不紧密问题较为严重。为了提升医疗问答系统的可靠性和准确性,LangChain技术的应用日益受到关注。如蔡振华[1]等为提升高校财务部门的服务水平构建了一个基于LangChain和ChatGLM的财务问答系统,测试结果显示在科研经费管理、学生业务、工薪劳务等领域的问答已接近实用水平。姜嘉伟[2]等创建了基于LangChain框架结合多个大语言模型(LLMs)的问答系统,旨在为当前大语言模型存在的训练数据外的无法回答、无法对本地文档中的内容进行问答等问题提供解决方案。窦凤岐[3]等采用了检索增强生成(RAG)技术,并基于LangChain结合大语言模型构建了课程问答系统,克服了生成式AI问答系统在涉及新的知识时会生成一些错误的答案。陈迪欢[4]等运用LangChain对各个文本进行检索并分块,计算问题示例向量与各个分块文本向量的相似度,将检索到相似度高的分块文本和问题示例进行拼接的方法,针对大语言模型在垂直领域回答精度不高的问题得到解决。此外Vidivelli S [5]等还结合LoRA和QLoRA策略调整生成式问答,进一步提高可行性,升级客户体验和扩展数据准入,能够准确有效地处理用户的问题。Jeong J [6]等研究发现LangChain通过链接数据来影响LLM,展示出了研究论文审查的显著潜力。研究表明,LangChain可以快速有效地总结研究,使其成为文献综述的宝贵工具。Chan MM [7]等利用LangChain与OpenAI结合来增强MOOC功能,并自动化和完善MOOC反馈流程,可以为学生提供个性化反馈。D A W [8]发起了一个完全基于过敏和鼻学国际共识声明:鼻-鼻窦炎(ICAR-RS)的LangChain/OpenAI驱动

聊天机器人，能够提供直接且可操作的建议。Haurum R K [9]等使用 LangChain 和 Pinecone 开发了一个英语和丹麦语的房地产 AI 代理，通过测试，与豆包和 ChatGPT 4 等其他 LLM 相比，代理表现出出色的性能。MSV J [10]使用 Python 及 Langchain 进行高级文本处理和信息检索，定义 QChain 基于向量存储构建并通过 Google PaLM 2 模型来回答相关问题。研究表明，在提高信息检索和问答系统的准确性和效率方面具有显著优势。Machlis S [11]利用 reticulate R 包在 RStudio 和 Python 之间建立桥梁，提高了数据处理与分析的效率与准确性。

通过梳理相关文献，本研究发现，当前学者正从多领域视角深入探索 LangChain 与大模型在实际中的应用。在医疗领域，鉴于信息更新迅速、专业词汇深奥且诊疗流程复杂，传统检索手段难以有效满足医护人员快速、精确获取医疗信息的需求，因此，本文提出了一种基于 LangChain 与大模型的医疗智能问答系统，通过融合先进的自然语言处理技术与自定义的推理链条，能够实现对医疗领域复杂问题的深度理解和精准回答，推动医疗问答系统在提高效率、准确性方面的应用。

2. LangChain 框架及其在自然语言处理中的应用

2.1. LangChain 框架简介

LangChain 是一个开源框架，旨在简化大型语言模型(LLM)驱动应用的开发。通过模块化设计，LangChain 集成了多个核心组件，方便开发者快速构建聊天机器人、问答系统等自然语言处理(NLP)应用。

LangChain 的架构由以下关键模块组成：

Models (模型)：封装与 LLM 的交互，支持多种语言模型的无缝集成。

Prompts (提示词)：提供灵活的模板工具，用于生成高质量输入，优化模型响应效果。

Memory (记忆)：存储对话上下文，支持短期和长期记忆，提升对话一致性。

Chains (模型链)：允许将多个模型或步骤链接起来，执行复杂任务流程。

Agents (代理)：动态调用工具或模型，结合外部资源扩展 LLM 的能力。

LangChain 在 NLP 领域的应用包括：

聊天机器人：利用记忆和模型链模块构建上下文感知的对话系统。

问答系统：结合检索模块定位内容，并生成精准答案。

对话分析：使用记忆模块分析长时间对话的语义和用户意图。

LangChain 框架为长文本处理和复杂任务提供了高效解决方案。随着语言模型的不断发展，LangChain 将进一步推动 NLP 技术的普及与应用。

2.2. LangChain 框架在问答系统中的原理

LangChain 提供了丰富的功能模块来支持多种自然语言处理场景。对于问答场景，其步骤包括构建文本库并基于文本回答问题，这种方法能够高效地解决基于文本的问答需求。以 TXT 文件为例，加载文本后按照预设规则对其进行分割，将分割后的文本向量化，存储在向量数据库中，用户提问后针对问题进行分词处理并向量化，系统与向量数据库中的数据进行相似度匹配计算，将匹配到的相似文本作为提示词与提问内容一起提交给模型，由模型生成答案。其中引入相似度匹配计算机制，为了仅将与问题相关的文本传递给模型，减少无效开销。相似度匹配计算主要使用 FAISS 索引方法，FAISS 通过相似性搜索返回相似分数，分数越小代表文本与问题的相关性越高[12]，如公式(1)所示：

$$d(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (1)$$

其中， p_i 和 q_i 分别表示向量 p 和 q 在第 i 个维度上的坐标值。

最后引入 Agent 模块，系统能够根据问题的复杂度选择最合适的工具链组合，并灵活调整执行步骤。例如，在处理多轮问答时，Agent 可以利用检索模块召回相关文本，将其嵌入到提示词中以优化模型响应。这种动态规划的能力极大地提升了框架的适应性和效率。如图 1 所示：

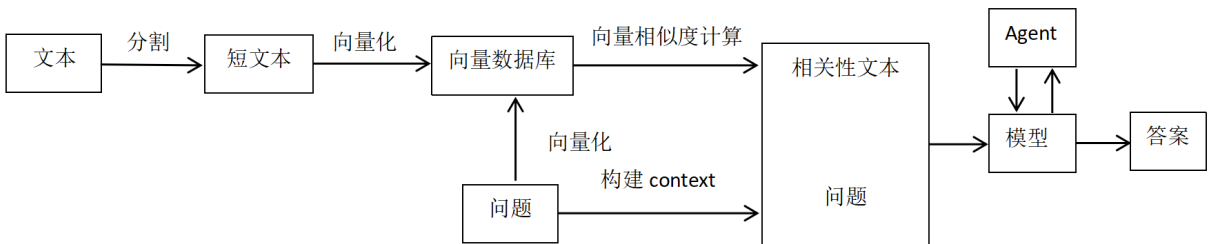


Figure 1. LangChain framework question answering system schematic diagram
图 1. LangChain 框架问答系统原理图

3. 基于 LangChain 与大模型的医疗问答模型系统结构

主要分为四个部分内容，分为识别真实意图，医疗问题设计，日常问题设计，探索性问题设计，这样设计可以扩大本问答系统研究的覆盖面，具体结构如图 2 所示：

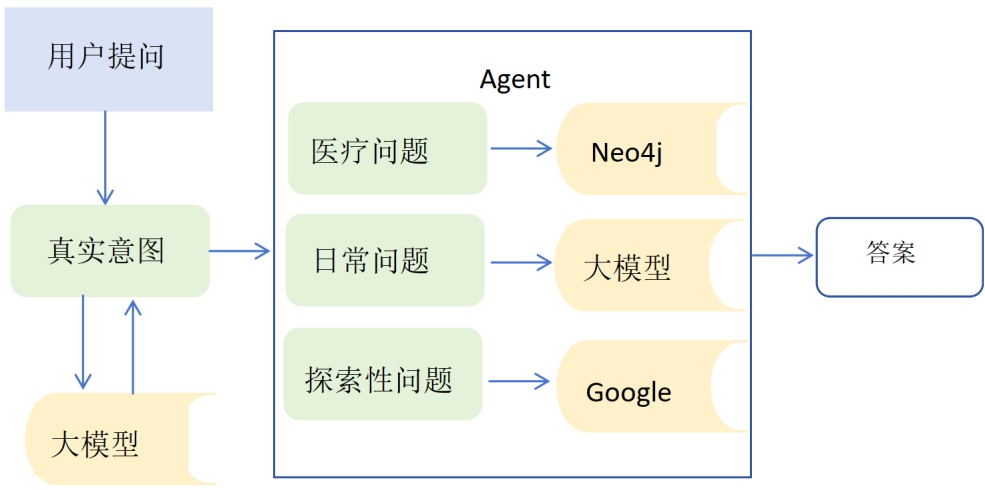


Figure 2. Medical question answering model system structure diagram based on LangChain and large model
图 2. 基于 LangChain 与大模型的医疗问答模型系统结构图

3.1. 数据准备与知识图谱构建

数据准备：本研究所使用的医疗数据主要来源于“寻医问药网”和“人民健康网”。这两个平台提供了丰富的半结构化数据，涵盖广泛的疾病信息及其相关并发症等信息。为高效获取上述数据，采用了开源爬虫框架 Scrapy 进行数据采集。通过服务器配置 Scrapy 框架并定义爬虫(Spider)，明确目标网站入口 URL 及爬取规则，针对网站的医疗知识数据精确提取。蜘蛛模块利用 XPath 和 CSS 选择器从网页中提取结构化数据，如疾病信息、症状描述等。在爬取过程中，调度器动态管理 URL 队列，通过递归爬取实现数据覆盖广度，下载器高效抓取网页内容并将数据传递给蜘蛛解析。为提高爬取稳定性与效率，下载中间件采用随机 User-Agent 和 IP 代理池，规避反爬机制。解析后的数据经由项目管道(Item Pipeline)清洗与处理，统一格式，去除冗余，并输出为结构化文件，直接适配知识图谱的节点与关系构建需求。为知识

图谱的构建提供了优质数据支撑。本研究所使用的数据全部来源于寻医问药网(<https://www.xywy.com/>)和人民健康网(<http://health.people.com.cn/>), 两者均为开源、公开访问的网站。研究过程中未涉及任何需要授权访问的受限数据或敏感个人信息。

知识图谱构建: 本研究构建了一个基于三元组形式的医疗知识图谱[13], 其中有大约 12,000 多个知识实体, 分为 13 个类别, 并以“实体-关系-属性”结构存储医疗信息。13 个实体类别: 疾病(Disease)、药物(Drug)、症状(Symptom)、检查(Check)、治疗(Treatment)、部位(Part)、病因(Cause)、食物(Food)、细胞类型(Cell)、检查指标(Value)、病理机制(Pathological)、科室(Department)、并发症(Complication), 导入 Neo4j 中实现知识图谱可视化, 直观呈现了疾病与其症状、治疗方式及并发症等信息的关联[14], 给用户良好体验。如图 3 所示:

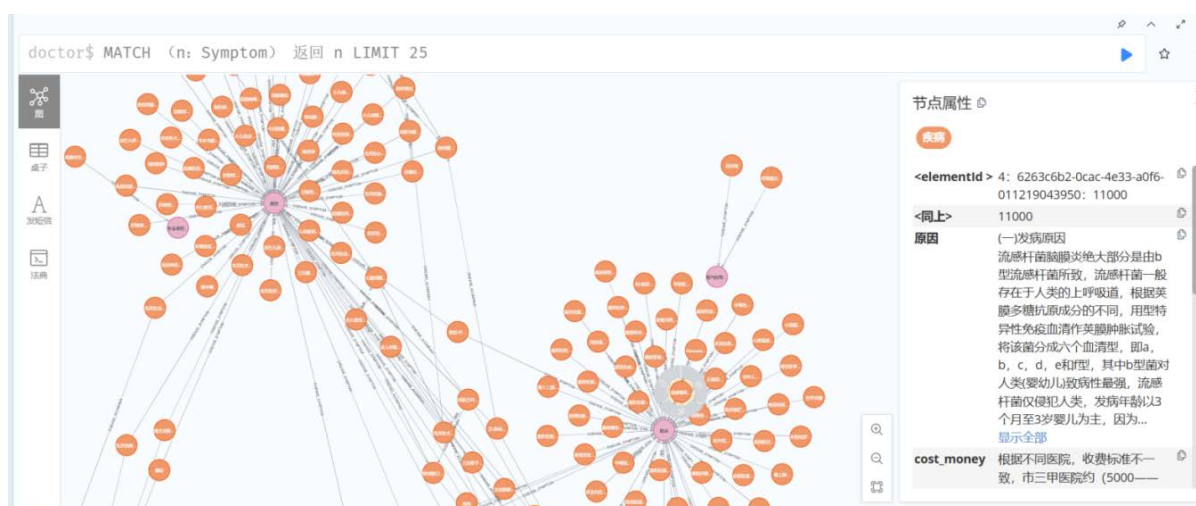


Figure 3. Knowledge graph diagram

图 3. 知识图谱示意图

3.2. Agent 串联

在接收到用户的问题时, 首先需要考虑其历史对话信息。这些历史对话包含有关上下文的信息, 可以帮助大模型理解用户的真实问题。基于 LangChain 设置一个提示模板, 结合历史对话和当前用户问题信息, 为大模型提供完整的上下文。

在历史对话中, 用户可能提到一些概念, 比如“感冒”, 当用户之后问到“吃什么药好得快?”时, 我们需要归纳出完整的上下文信息, 替换其中的指代词, 生成“感冒了吃什么药好得快? ”。例如, 如果历史对话包含以下内容:

用户: 我觉得有点不舒服, 好像是感冒了。

系统: 你现在有哪些症状?

用户: 我有些喉咙痛, 头有点晕。

那么, 当用户问“吃什么药好得快?”时, 系统需要识别到这是在询问关于“感冒”相关的药物信息。因此, 生成的归纳问题应为: “感冒了吃什么药好得快?”

在处理特定业务场景之前, 需要定义一个 Agent, 该 Agent 通过描述性的方式引导大模型选择合适的工具来回答用户的问题。值得注意的是, 医疗问题模块具有特殊性, 因为在知识图谱查询中, Agent 不仅需要对输入文本进行加工, 还需要对原问题进行实体抽取。因此, 采用匿名函数的方式将原问题传递给图谱查询模块。如图 4 所示:


```

prefix = """请用中文，尽你所能回答以下问题。您可以使用以下工具："""
suffix = """Begin!

History: {chat_history}
Question: {input}
Thought:{agent_scratchpad}"""

agent_prompt = ZeroShotAgent.create_prompt(
    tools=tools,
    prefix=prefix,
    suffix=suffix,
    input_variables=['input', 'agent_scratchpad', 'chat_history']
)
llm_chain = LLMChain(llm=get_llm_model(), prompt=agent_prompt)
agent = ZeroShotAgent(llm_chain=llm_chain)

memory = ConversationBufferMemory(memory_key='chat_history')
agent_chain = AgentExecutor.from_agent_and_tools(
    agent = agent,
    tools = tools,
    memory = memory,
    verbose = os.getenv('VERBOSE')
)
return agent_chain.run({'input': query})

```

Figure 4. Defining Agent graph

图 4. 定义 Agent 图

系统通过 LLMChain 来处理整体逻辑，LLMChain 将结合历史问题和用户输入，帮助模型判断其回答与用户问题的相关性和一致性，确保回答的准确性和相关性。最终，处理后的问题将传递给 Agent，Agent 会根据预定义的描述精准地将问题定位到相应的处理逻辑上。这种设计确保了问题能够被准确识别并传递给适当的 Tool，从而提供高效且准确的回答。

3.3. 问答的三种情况构想

3.3.1. 面对医疗问题设计

此模块是问答的重点，主要逻辑如图 5 所示：

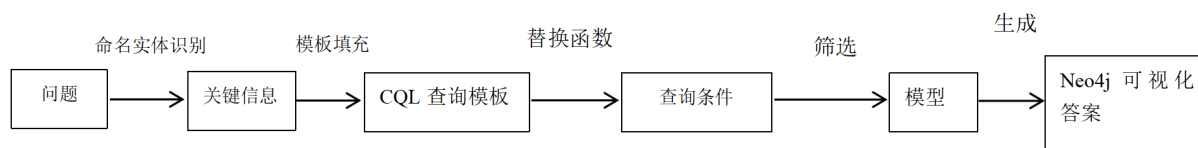


Figure 5. Medical Q & A logic graph

图 5. 医疗问答逻辑图

命名实体识别与模板选择：首先，针对用户提问，使用大模型进行命名实体识别(NER)，提取问题中的关键信息[15] [16]。识别出的实体随后将填充到预设的 CQL 查询模板中。此外，在查询之前，需要筛选出与用户问题相关的模板，排除无关模板。例如，用户询问某个疾病时，系统会使用 MATCH 语句查

询疾病节点并返回相应的属性和答案模板。如图 6 所示：

```
# agent.py
def graph_func(self, query):
    # 命名实体识别
    response_schemas = [
        ResponseSchema(type='list', name='disease', description='疾病名称实体'),
        ResponseSchema(type='list', name='symptom', description='疾病症状实体'),
        ResponseSchema(type='list', name='drug', description='药品名称实体'),
    ]

    output_parser = StructuredOutputParser(response_schemas=response_schemas)
    format_instructions = structured_output_parser(response_schemas)

    ner_prompt = PromptTemplate(
        template = NER_PROMPT_TPL,
        partial_variables = {'format_instructions': format_instructions},
        input_variables = ['query']
    )

    ner_chain = LLMChain(
        llm = get_llm_model(),
        prompt = ner_prompt,
        verbose = os.getenv('VERBOSE')
    )

    result = ner_chain.run({
        'query': query
    })

    ner_result = output_parser.parse(result)

print(agent.graph_func('感冒一般是由什么引起的? '))
print(agent.graph_func('感冒吃什么药好得快? 可以吃阿莫西林吗? '))
```

Figure 6. Named entity identification graph

图 6. 命名实体识别图

文本替换与模板填充：通过文本替换函数，将提取出的实体内容替换模板中的占位符，生成具体的查询条件。

筛选与相关性判断：筛选模板目的是避免无关信息的干扰，确保模型只处理相关问题。举例来说，若用户询问“感冒和鼻炎是并发症吗”，模型会识别出用户关注的是并发症这一主题，并找到与并发症相关的模板，列出感冒和鼻炎的并发症并判断其相关性。

CQL 执行与结果总结：最后，系统执行相应的 CQL 查询，向 Neo4j 数据库发送请求。查询结果将作为上下文信息提供给大模型，以生成最终的答案[17]。

3.3.2. 面对日常问题设计

在日常交际问题的回答过程中，通过一般性问题 Agent Tool 结合大模型的分析能力来生成响应效果很好。然而，在使用大模型自身的能力回答问题时，应避免模型的过度自由发挥。因此，提示词的设计需要相对严格，以确保生成的答案既准确又符合上下文要求。通过这种方式，可以确保模型在处理日常

问题时能够维持合适的回答范围，又充分利用模型的语言生成能力，实现高质量的自然语言交互。

3.3.3. 面对探索性问题设计

考虑了一般日常问题和医疗问题，那如果用户的问题不在这些范围内，就需要加一个兜底方案，就是先去查 Google，然后总结 Google 的搜索结果。定义提示词中，依然让大模型根据搜索结果去回答问题，避免模型回答太发散，定义搜索工具函数[18]，当模型判断不属于前两种问题场景时，LLM Request-Chain 会请求 URL。

4. 实验设计与结果分析

4.1. 实验设置与数据集

分别准备 30 个感染性疾病方面问题，免疫性疾病方面问题，衰老性疾病方面问题，外伤性疾病方面问题给所有待测试模型，为了方便后序待测试模型从疾病的类型、病因、预防和治疗等方面全面回答，并邀请 20 名专业医生进行评测。数据集来源于“寻医问药网”和“人民健康网”构建的医疗知识图谱。

4.2. 评价指标

从回答准确性，上下文连贯性，方案合理性三个维度打分。打分标准如下：回答准确性：0.8~1 (专业级)：答案完全正确，基于权威医学知识，涵盖用户需求的关键点，无遗漏或误导。0.6~0.8 (合格级)：答案基本正确，但缺乏部分细节或表述不够清晰。0.6 以下(不合格)：答案存在明显医学知识错误、信息严重不足或具有误导性。

上下文连贯性：0.8~1 (专业级)：能够准确理解问题背景，回答逻辑缜密，连续提问的上下文关联性强，语言表达流畅。0.6~0.8 (合格级)：基本符合上下文逻辑，但存在轻微逻辑跳跃或关联性稍弱的问题。0.6 以下(不合格)：回答与问题背景脱节，或逻辑混乱难以理解。

方案合理性：0.8~1 (专业级)：建议基于医学指南，科学合理，具备个性化针对性，具有实际实现价值。0.6~0.8 (合格级)：建议基本合理，但个性化程度不足或未充分考虑方案合理性。0.6 以下(不合格)：建议不科学、与医学常识冲突。

分数精确到小数点后一位，每个问题由 2 名专业医生独立评分。

此方案既有专业性，又符合实际操作，简化评分流程的同时增强了结果精度[19]。

4.3. 实验结果

本研究采用 ChatGPT-3.5-turbo 作为 LLM 模型，评测结果如表 1 所示：

Table 1. Evaluation results table

表 1. 评测结果表

| 模型 | 回答准确性 | 上下文连贯性 | 方案合理性 |
|-----------------------------|-------|--------|-------|
| MyAgent (ChatGPT-3.5-turbo) | 103.5 | 105 | 108.1 |
| 文心一言 3.5 | 80.75 | 79.8 | 84.9 |
| ChatGPT-4 | 94.5 | 95.4 | 97.1 |

4.4. 案例分析

选取一个有代表性问题并收集几个模型的回答分析，MyAgent、文心一言和 ChatGPT-4 在准确性、上下文连贯性和方案可行性方面表现各异，如表 2 所示：

Table 2. Case analysis table
表 2. 案例分析表

| 模型 | 问题 1：心肌炎是什么 | 问题 2：如何治疗 | 问题 3：需要治疗多久 |
|-----------|--|---|--|
| MyAgent | 心肌炎(myocarditis)是指由各种原因引起的心肌的局限性或弥漫性炎症，多种因素如感染、物理和化学因素均可引起心肌炎，所造成的心肌损害的轻重程度差别很大，临床表现各异，轻症患者无任何症状，而重症患者可发生心力衰竭、心源性休克甚至猝死。 | 一般治疗：卧床休息，急性期：卧床休息至热退后 3~4 周，心影恢复正常后可轻微活动。恢复期：继续限制活动，重症患者需卧床休息半年以上。抗生素使用：由于细菌感染是重要因素，初期治疗时适当使用抗生素(如青霉素、先锋霉素VI)。心肌保护：抗氧化剂，大剂量维生素 C、维生素 B、辅酶 Q10。营养心肌药物：能量合剂、极化液。对症治疗：心律失常偶发早搏、一度房室传导阻滞，密切观察。频发早搏或室上性快速性心律失常：使用抗心律失常药物治疗(如胺碘酮)。室性心动过速：根据血流动力学情况使用利多卡因、胺碘酮或电复律。缓慢型心律失常：使用阿托品治疗，无效时考虑安放心脏起搏器。心力衰竭：低盐饮食，使用利尿剂。小剂量洋地黄类药物。顽固性心衰：使用多巴酚丁胺、米力农。心源性休克：使用酚妥拉明、多巴胺等血管活性药物。 | 主要看心肌炎的严重程度，如轻型的心肌炎可能不需要住院，居家服药、注意休息就可以，可能两周到一个月可以完全恢复正常。但是急性心肌炎合并轻度的心功能的下降，甚至已经合并心室的结构改变，可能需要进一步的住院治疗。住院的周期可能 7~10 天甚至可能两周左右时间，出院之后定期服药，门诊进行随访，心肌炎营养心肌的疗程可能 3~6 个月。如果重症心肌炎甚至是爆发性心肌炎，可能合并心源性休克或者心脑综合征，甚至出现严重心律衰竭。起病较急病程进展较快，病情较危重，所以治疗周期较长，可能住院期间的治疗时间就一个月甚至更长时间，心肌炎恢复时间也较长，需要定期随访然后服药，合并症和并发症治疗时间可能是半年甚至 1~2 年时间。 |
| 文心一言 3.5 | 心肌炎是指心肌发生的局限性或弥漫性炎症，通常由病毒感染、细菌感染、药物影响等多种因素引起，可影响心肌的正常功能，导致心悸、气短、乏力等症状，严重时可能引发心律失常、心衰等严重后果。 | 心肌炎的治疗包括卧床休息、营养支持，使用免疫抑制剂、抗病毒药、营养心肌药、改善心功能药等药物，以及必要时的手术治疗如心脏起搏器安装、射频消融术等。具体治疗方案需根据患者病情由医生制定。 | 心肌炎的治疗时间因病情严重程度和类型而异。一般来说，急性轻症心肌炎的治疗周期可能在 2~4 周内，而急性重症心肌炎或暴发性心肌炎可能需要 1~3 个月甚至更长时间。慢性心肌炎的治疗周期则可能长达数年，需要长期坚持治疗。 |
| ChatGPT-4 | 心肌炎是指心肌因感染、免疫异常或毒性作用引发的炎症性疾病，常见于病毒感染。患者可能出现胸痛、乏力、心悸、呼吸困难等症状，严重时引发心力衰竭或心律失常。 | 心肌炎的治疗依赖于病因和病情严重程度。病毒性心肌炎通常采用抗病毒治疗和免疫调节。细菌性心肌炎则使用抗生素治疗。免疫性心肌炎，常使用免疫抑制剂如糖皮质激素。心力衰竭可通过利尿剂、ACE 抑制剂、 β 受体阻滞剂等药物治疗，重症时使用机械支持或心脏移植。针对心律失常，可能需要抗心律失常药物或植入心脏起搏器。 | 心肌炎的治疗时间因病因和病情严重程度而异。轻度病例可能在几周内康复，但需要定期复查心电图和超声心动图。对于中重度患者，治疗可能需要几个月，期间包括抗病毒、免疫抑制或心力衰竭管理等。重症病例可能需要长期治疗，直至心脏功能恢复或使用机械支持。对于免疫性或慢性心肌炎，可能需要长期使用免疫抑制剂，直到症状得到有效控制。治疗过程中，患者应定期随访，监控心脏功能。 |

MyAgent 在准确性上提供了全面且无误的信息，详细描述了心肌炎的定义、常见病因和治疗方法，尤其是治疗时间上，充分考虑了不同病情和患者的差异，给予个性化的建议，确保了方案的可行性。在上下文连贯性方面，MyAgent 回答逻辑清晰，内容衔接自然，避免了信息断裂或跳跃，使用户更容易理解。相比之下，文心一言的回答虽然概括了基本概念，但未深入展开治疗细节，且治疗时间的描述较为模糊，缺乏针对性。而 ChatGPT-4 尽管涵盖了较为全面的治疗建议，但存在幻觉问题，尤其在治疗方法

和治疗时间方面未能充分准确表达不同病症下的差异。综合来看, MyAgent 在回答的准确性、完整性和方案的实际可行性方面均优于其他两个模型。

5. 结论

基于 LangChain 与大模型的医疗问答系统为医学领域的人工智能应用提供了创新性解决方案。在本研究中, 结合 LangChain 框架与大型预训练语言模型, 构建了一种高效、智能的医疗问答系统, 旨在改善患者咨询体验, 并推动医疗服务的智能化进程。

研究表明, 基于大模型的医疗问答系统在准确性和用户体验方面具有显著优势。通过 LangChain 的 Agent Tool、链式结构和 Prompt 设计, 系统能够高效整合医学知识, 结合实时数据与专业医生反馈, 显著提升了回答质量。同时, 知识图谱可视化技术也使得医学知识的呈现更加直观, 为用户提供了易于理解的信息。

此外, 本研究验证了跨领域知识整合的价值, 尤其是在医学领域, 信息庞大且不断更新。未来, 随着数据集的扩展与模型优化的深入, 医疗问答系统有望变得更加智能, 能够为不同需求的用户提供更精准、个性化的医疗咨询服务。

基金项目

北京印刷学院青年托举项目(20190124072); 北京印刷学院博士科研启动基金(27170124010)。

参考文献

- [1] 蔡振华, 王燕贞, 杨净. 基于 LangChain 和 ChatGLM 的高校财务问答系统研究与实现[J]. 现代计算机, 2024, 30(15): 104-110.
- [2] 姜嘉伟. 基于 LangChain-LLMs 框架的智能问答系统的设计与实现[D]: [硕士学位论文]. 延吉: 延边大学, 2024.
- [3] 窦凤岐, 胡珊, 李佳隆, 等. 基于 LangChain 的 RAG 问答系统设计与实现——以 C 语言课程问答系统为例[J]. 信息与电脑(理论版), 2024, 36(6): 101-103.
- [4] 陈迪欢, 陆希, 张耀峰. 基于 LangChain + LLM 的招投标领域问答系统研究[J]. 大数据时代, 2024(5): 23-28.
- [5] Vidivelli, S., Ramachandran, M. and Dharunbalaji, A. (2024) Efficiency-driven Custom Chatbot Development: Unleashing LangChain, RAG, and Performance-Optimized LLM Fusion. *Computers, Materials & Continua*, **80**, 2423-2442. <https://doi.org/10.32604/cmc.2024.054360>
- [6] Jeong, J., Gil, D., Kim, D. and Jeong, J. (2024) Current Research and Future Directions for Off-Site Construction through LangChain with a Large Language Model. *Buildings*, **14**, Article 2374. <https://doi.org/10.3390/buildings14082374>
- [7] Morales-Chan, M., Amado-Salvatierra, H.R., Medina, J.A., Barchino, R., Hernández-Rizzardini, R. and Teixeira, A.M. (2024) Personalized Feedback in Massive Open Online Courses: Harnessing the Power of Langchain and Openai Api. *Electronics*, **13**, Article 1960. <https://doi.org/10.3390/electronics13101960>
- [8] Workman, A.D., Rathi, V.K., Lerner, D.K., Palmer, J.N., Adappa, N.D. and Cohen, N.A. (2023) Utility of a Langchain and Openai Gpt-powered Chatbot Based on the International Consensus Statement on Allergy and Rhinology: Rhinosinusitis. *International Forum of Allergy & Rhinology*, **14**, 1101-1109. <https://doi.org/10.1002/alr.23310>
- [9] Haurum, K.R., Ma, R. and Long, W. (2024) Real Estate with AI: An Agent Based on LangChain. *Procedia Computer Science*, **242**, 1082-1088. <https://doi.org/10.1016/j.procs.2024.08.199>
- [10] Msv, J. (2023) Building a Q&A App with LangChain and Google PaLM 2. InfoWorld.
- [11] Machlis, S. (2023) Generative AI with LangChain, RStudio, and Just Enough Python. InfoWorld.
- [12] Zhu, J., Liu, D. and Chen, S. (2024) Multiple-Choice Question Generation and Difficulty Calculations Based on Semantic Similarity. *Neural Computing and Applications*, 1-13. <https://doi.org/10.1007/s00521-024-10671-8>
- [13] 周泽先, 刘晓静. 基于知识图谱的藏医可视化研究与问诊系统构建[J/OL]. 现代中医药: 1-8. <http://kns.cnki.net/kcms/detail/61.1397.R.20241204.1415.020.html>, 2024-12-24.
- [14] Chen, D., Lu, C., Bai, H., Xia, K. and Zheng, M. (2024) Integrating AI with Medical Industry Chain Data: Enhancing Clinical Nutrition Research through Semantic Knowledge Graphs. *Frontiers in Digital Health*, **6**, Article 1439113.

- <https://doi.org/10.3389/fdgth.2024.1439113>
- [15] 张君冬, 刘江峰, 王震宇, 等. 用户响应式场景下大模型驱动的 AI 问答研究: 以医疗分诊为例[J/OL]. 情报理论与实践: 1-13. <http://kns.cnki.net/kcms/detail/11.1762.G3.20241018.1002.002.html>, 2024-12-24.
 - [16] Zhang, S. and Song, J. (2024) A Chatbot Based Question and Answer System for the Auxiliary Diagnosis of Chronic Diseases Based on Large Language Model. *Scientific Reports*, **14**, Article No. 17118. <https://doi.org/10.1038/s41598-024-67429-4>
 - [17] Reichenpfader, D., Rösslhuemer, P. and Denecke, K. (2024) Large Language Model-Based Evaluation of Medical Question Answering Systems: Algorithm Development and Case Study. In: *Studies in Health Technology and Informatics*, IOS Press, 22-27. <https://doi.org/10.3233/shti240006>
 - [18] Guo, Q., Cao, S. and Yi, Z. (2022) A Medical Question Answering System Using Large Language Models and Knowledge Graphs. *International Journal of Intelligent Systems*, **37**, 8548-8564. <https://doi.org/10.1002/int.22955>
 - [19] 张芊, 陈攀峰, 冯林坤, 等. PeMeBench: 中文儿科医疗问答基准测试方法[J]. 大数据, 2024, 10(5): 28-44.