小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2025-0232
2026 年 5 月 第 5 期
Vol.47 No.5 2026

融合知识图谱和大模型的医疗智能问答方法研究

赵海燕[1],高东昇[1],曹 健[2],陈庆奎[1]

[1](上海理工大学 上海市现代光学系统重点实验室 光学仪器与系统教育部工程研究中心,上海 200093)
[2](上海交通大学 计算机学院,上海 200030)
E-mail:233360817@st.usst.edu.cn

摘 要: 医疗领域的自动问答对答案的准确性有很高的要求.尽管大语言模型(LLM)提供了通用的问答能力,但是无法满足医疗领域对答案准确性的要求.与此相对,基于知识图谱检索的自动问答依赖于客观的知识表达对回答质量提供了可靠性保证,然而目前的方法中存在知识图谱检索效率不高、检索不充分、检索过多冗余信息,以及对较复杂的问题理解不充分进而影响检索的质量等问题.为此,本文中将知识图谱与 LLM 相结合,基于 LLM 对用户的提问进行问题分解,对每一个子问题在知识图谱中进行子图搜索,再将融合后的子图交给 LLM 以生成可靠的答案.文中的方法在医疗数据集 GenMedGPT-5k、LiveQA、HealthCareMagic-100k 和知识图谱 FB15k-237 上进行了实验.实验表明,文中的方法取得了较好的性能.

关 键 词: 大语言模型;知识图谱;检索增强;知识子图
中图分类号: TP391 **文献标识码:** A **文 章 编 号:** 1000-1220(2026)05-1127-07

Research on Medical Intelligent Question Answering Method Combining Knowledge Graph and Big Model

ZHAO Haiyan[1],GAO Dongsheng[1],CAO Jian[2],CHEN Qingkui[1]

[1](Shanghai Key Lab of Modern Optical System,Engineering Research Center of Optical Instrument and System,Ministry of Education,University of Shanghai for Science and Technology,Shanghai 200093,China)
[2](School of Computer Science,Shanghai Jiao Tong University,Shanghai 200030,China)

Abstract: In the medical area,automated question answering places high demands on the accuracy of answers. Although large language models (LLMs) provide general-purpose question-answering capabilities,they cannot meet the stringent accuracy requirements of the medical domain. In contrast,knowledge graph-based automated question answering relies on objective knowledge representation to ensure answer reliability. However,current methods suffer from issues such as inefficient knowledge graph retrieval,insufficient coverage,excessive redundant information,and inadequate understanding of complex questions,which negatively impact retrieval quality. To address these challenges,this paper integrates knowledge graphs with LLMs. Specifically,the LLM is used to decompose user questions into sub-questions,each of which is then subjected to subgraph retrieval in the knowledge graph. The merged subgraphs are then fed back to the LLM to generate reliable answers. The proposed method is evaluated on medical datasets (GenMedGPT-5k,LiveQA,HealthCareMagic-100k) and the knowledge graph FB15k-237. Experimental results demonstrate that the proposed approach achieves superior performance.

Keywords: Large Language Model (LLM);knowledge graph;search enhancement;knowledge subgraph

0 引 言

医疗问答系统作为连接公众与专业医学知识的桥梁,在健康咨询、临床决策支持和远程医疗等领域发挥着关键作用.医疗问答面临区别于通用领域的独特挑战:知识专业性、回答安全性、时效敏感性.

近年来,大规模语言模型(Large Language Models,LLMs)如 GPT-4[1]、Llama-3[2]、DeepSeek[3] 和 ERNIE-Bot 4.0[4] 的涌现,标志着预训练模型技术的跨越式进步.这些模型展现出近乎“百科全书式”的知识覆盖能力,能够高效处理开放域问题,甚至模拟人类使用工具(如搜索引擎)的行为,为用户提供即时、多样化的信息响应.然而,LLM 的生成机制存在固有缺陷,尤其是幻觉问题(Hallucination)和可解释性不足,严重限制了其在医疗等高可靠性需求领域的应用.不准确的内容同样也会引发人们的担忧[5].

现有的检索增强生成(Retrieval-Augmented Generation,RAG)框架部分缓解了这些问题,特别是通过在知识图谱(Knowledge Graph,KG)中进行检索相关的知识后提供给 LLM 以生成答案成为一种有效的方法.然而,将 LLM 与 KG 集成依然面临两大挑战:

1)多跳推理效率低下:复杂问题需分解为多个子问题并分步检索,而现有方法对子问题间的因果依赖关系建模不足,

收稿日期:2025-06-09 收修改稿日期:2025-08-19 基金项目:上海交通大学医工交叉项目(YG2024QNB05)资助. 作者简介:**赵海燕**,女,1975 年生,博士,副教授,CCF 会员,研究方向为服务计算、数据挖掘、推荐系统;**高东昇**,男,1999 年生,硕士研究生,研究方向大语言模型检索增强;**曹 健**,男,1972 年生,博士,教授,博士生导师,CCF 杰出会员,研究方向为智能数据分析、服务计算、协同计算、网络计算等;**陈庆奎**,男,1967 年生,博士,教授,博士生导师,CCF 会员,研究方向为计算机集群、并行数据库、并行理论、物联网等.

导致检索冗余或逻辑断层.

2)异构知识库兼容性不足:传统框架通常依赖单一数据库(如文本索引),难以适配知识图谱的多模态存储介质(如Neo4j、MongoDB 等).

针对上述问题,本文提出一种基于问题分解和知识融合的检索增强框架. 其核心创新包括:

· 提出了一种多跳问题分解、动态子图生成与融合的方法:通过设计特定的提示词,利用 LLM 的潜在推理能力将复杂问题分解为子问题序列,每个子问题在知识图谱上的检索结果(子图)作为后续检索的上下文输入,最终融合成全局知识图供 LLM 生成答案.

· 实现了一种通用化的知识检索模块:该模块支持 Neo4j、MongoDB、MySQL 等多种数据库的即插即用,通过统一接口实现异构知识源的协同查询.

与此同时,本文通过提示词的方式使问答系统能够扮演医生的角色,从医生的角度去理解和回答问题,这更能加深大语言模型对问题的理解,充分利用其潜在的知识.

1 问题定义与系统架构

本文的目的是通过检索器从图谱中过滤出与问题相关的知识图或关系路径,将领域知识动态注入到 LLM 中.

领域知识可以表示为一组三元组 $G=\{(e_S,r,e_0)\mid e_S,e_o\in E,r\in R\}$. 其中 e_S,表示关系对中的主体,e_o 表示客体,r 表示主客体之间的关系. E 表示知识图谱中所有实体的集合,R 表示知识图谱中的所有关系的集合. 给出一个问题 q,知识图谱 G,大语言模型 M_{LLM},本文通过最大化以下概率分布来产生问题 q 的响应 a.

问题求解概率分布:

$$P(S_q\mid M_{LLM},G,q),S_q=\{q_1,w_2,\cdots,q_n\} \tag{1}$$

$$P(a_2,G^2_{sub}\mid M_{LLM},G,q)=P(a_2,G^2_{sub}\mid M_{LLM},G,q_2,a_1,G^1_{sub})P(a_1,G^1_{sub}\mid M_{LLM},G,q_1) \tag{2}$$

$$P(a\mid LLM,G,q)=\prod_{i=1}^{n}P(a_i,G^i_{sub}\mid M_{LLM},G,q) \tag{3}$$

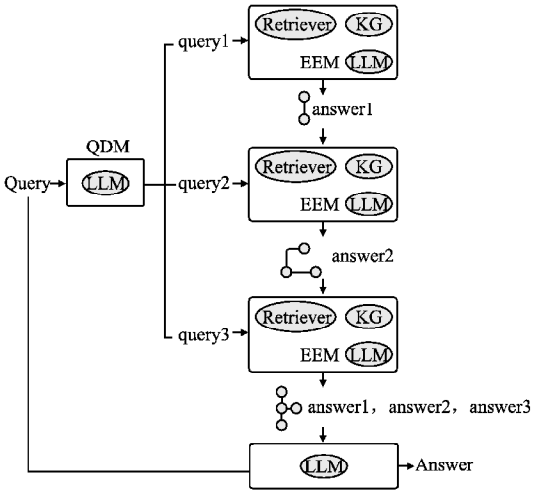

图 1 系统整体架构图

Fig. 1 Overall system architecture diagram

图 1 显示了本文提出的系统的整体框架. 系统的输入是用户的问题 *Query*. *Query* 被分解为多个简单的子问题{*query*1,*query*2,*query*3,…}. 每个子问题都需要系统处理,不过有处理的先后顺序. 先处理的子问题的输出将会参与到下一个子问题的处理过程中,也就是上一个子问题最后的输出将会是下一个子问题的输入. 中间模块的输出将包含一个知识子图和一个文本格式的回答. 最后系统的输出则只有一个文本形式的答案.

2 相关工作

指令(Prompt)设计. 从头开始训练一个模型是一个高成本的事情. 尤其是训练一个垂直领域的模型,不仅会有算力的问题,还会面对可信的数据量问题. 所以,预训练模型或微调模型[6]加上外部知识库以及优化指令输入,已成为一种趋势. 其本质就是利用检索技术,对外部知识库检索出相关的知识. 将其作为指令和问题一并输入到 LLM. 也就是给 LLM 一个可以参考的答案模板,最大限度的降低 LLM 的幻觉问题,以及弥补 LLM 对于垂直领域知识匮乏的问题. 检索增强是用来向 LLM 动态地注入额外的知识[7]. 通常是以文档作为作为外部知识. 先提前将外部文档生成嵌入向量存到向量数据库,然后将用户输入问题转换成嵌入式向量,然后和文档向量数据库的数据做相似度计算. 选择相似度 Top-N 的文档作为 LLM 的额外知识,和用户问题一起输入给 LLM[7]. 然而,这会导致模型的上下文输入变得很长,从而减慢解码时间,增加用户等待回答的时间,且 LLM 也可能不支持这么长的 token,所以可以将 LLM 输入上下文进行压缩[8]. 本文的工作是利用 LLM 的推理能力对知识图谱检索出关键的信息,并生成高质量的指令输入.

知识图谱动态注入 LLM. 知识图谱以结构化的方式表示信息,通常使用实体、属性和关系来建模现实世界中的概念及其相互联系. 这使得机器更容易理解和处理信息. 知识图谱和 LLM 的结合主要有 3 种方式:1)利用知识图谱来训练 LLM;2)利用 LLM 来生成知识图谱的查询语句来检索关键信息[9];3)利用 LLM 在知识图谱中进行推理,找到和问题相关的知识路径[10].

本文在知识图谱中检索知识借助了 LLM 生成的检索查询语句. 为了提高检索的质量和复杂问题的检索,首先会对问题进行拆分. 检索过程中会考虑到拆分后的子问题间的因果关系. 对于下一个子问题的求解将依赖上一个子问题在知识图谱中检索出来知识子图和输出答案,这样就能保持一条完整的推理链路,也对用户问题的回答增加了可解释性.

3 方 法

本文将问题 q 分为多个子问题,多个子问题对应多个"解题"步骤. 步骤 $Step^i$ 负责问题 q^i 的解答,每个步骤的输出就是一个子图 G^i_{sub} 和问题 q^i 的答案 $answer^i$. $Step^{i+1}$ 的输出为:

$$answer^{i+1},G^{i+1}_{sub}=EEM(answer^i,G^i_{sub}) \tag{4}$$

本文所提出方法的框架如图 1 所示,其主要包含两个部分:

1)知识子图的生成. 为了让 LLM 的回答与知识图谱中

的知识对齐,需要检索器尽可能的检索出与问答相关的实体集,找到符合问答上下文的实体间关系,最后组成知识子图 G^i_{sub}.

2)知识子图的融合. 对于上一个步骤的输出的知识子图 G^i_{sub},需要提取与当前上下文有潜在关联的关系对,这有利于对当前步骤的知识补充,也需要剔除上 G^i_{sub} 中没有价值的关系对,避免其降低 LLM 回答的质量和增加响应速度.

3.1　问题分解

如图 2、图 3 所示,通过设计好的提示词,借助 LLM 的推理能力将问题分解.

```
You are a doctor, and you need a doctor's perspective to analyze problems.
Firstly, the problem needs to be decomposed into 1 to 3 sub problems.
By solving these sub problems in sequence, the final answer can be obtained.
Now we need you to provide the decomposition result of a problem first.
give an example as follow:
Question: What are the symptoms of gastric cancer and how to treat it.
Answer: {
  "answer":["What are the common symptoms of gastric cancer?",
            "How is gastric cancer typically diagnosed?",
            "What are the available treatment options for gastric cancer?"]
}

tips:You just need to give the answer, no other words are needed.
The format of the answer only requires one JSON, please answer my question
next .
Question:{question}
```

图 2　问题分解相关提示词

Fig. 2　Related prompt words for problem decomposition

```
What are the treatment methods for gastric cancer, and what are their
advantages and disadvantages?
```

```
{
"answer": [
"What are the surgical treatment options for gastric cancer,
and what are their pros and cons?",

"What are the non-surgical treatment options for gastric cancer ,
and what are their advantages and disadvantages?",

"How are treatment methods selected based on the stage
and characteristics of gastric cancer?"
]
}
```

图 3　问题分解示例

Fig. 3　Example of problem decomposition

3.2　知识子图的生成

G_{sub} 的生成主要有以下步骤:1)查询语句生成;2)实体对齐;3)子图生成.

3.2.1　查询语句生成

知识图谱可以采用多种数据库进行存储,包括图数据库(如 Neo4j、Dgraph)、文档数据库(如 MongoDB)以及关系型数据库(如 MySQL). 不同的存储介质在查询效率、数据关联性和扩展性上各有优劣,因此需要一种统一的查询方式,以适应不同后端存储的需求.

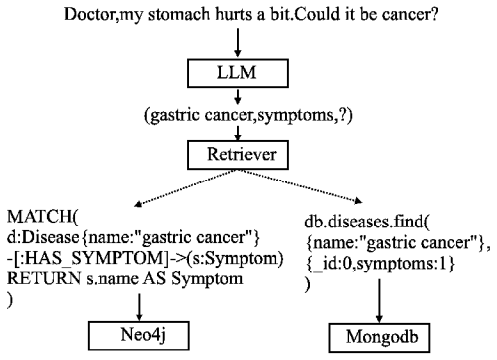

图 4　查询语句转换过程

Fig. 4　Query statement conversion process

为此,本文实现了一个转换模块(结构如图 4 所示),其主要功能是将自然语言或半结构化的查询请求转换为知识图谱的可执行查询语句. 由于 LLM 生成的查询可能存在歧义或语法错误,本文定义了一种标准化的三元组查询语法,以规范化查询输入. 相关提示词如图 5 所示,生成 Mongodb 查询语句如图 6 所示.

三元组查询语法:

三元组采用 < 主语,关系,宾语 > 的形式,其中:

· 主语:查询的实体(如"胃癌");

· 关系:目标属性或关联关系(如"症状");

· 宾语:可以是具体值或占位符(? 表示待查询的值).

例如,查询 < 胃癌,症状,? > 对应的语义是"胃癌的症状是什么". Retriever 会解析该三元组,并根据底层存储类型(如 Neo4j 的 Cypher 查询、MySQL 的 SQL 语句等)生成适配的查询语言,从而屏蔽不同数据库的语法差异.

```
The knowledge graph is known to have the following relationships:
Disease->risk_factors->Factor. Disease->symptom_of->Symptom
give an example as follow:
Question: "What are the symptoms of diabetes and the causes of illness"
Answer:  {
            "answer":[
                      ["?","symptom_of", "diabetes"],
                      ["diabetes","risk_factors","?"]
                     ]
         }
Tips:You just need to give the answer, no other words are needed.
Each triplet of the answer needs to contain a symbol '?'
If the answer cannot be found
in the graph query, you can return as follow:
        {
         "answer":[]
        }
he format of the answer only
requires one JSON, please answer my question next.
Question:{question}
```

图 5　查询语句转换相关的提示词

Fig. 5　Prompt words related to query statement conversion

查询结果返回:

Retriever 的返回结果同样采用三元组结构,例如 < 胃癌,症状,胃疼 >,表示"胃癌的症状是胃疼". 这种结构化输出便于后续处理,并可直接用于 LLM 的推理或知识增强任务.

```
FUNCTION query(subject, relation, object)
  triplets ← empty list

  IF object is NULL THEN
    sql ← "match (a)-[r:relation]->(b) where a.name=~'(?i)subject' return a,r,b"
  ELSE IF relation is NULL THEN
    sql ← "match (a)-[r]->(b) where a.name=~'(?i)subject' and b.name=~'(?i)object' return a,r,b"
  ELSE IF subject is NULL THEN
    sql ← "match (a)-[r:relation]->(b) where b.name=~'(?i)object' return a,r,b"
  END IF

  data ← EXECUTE sql query USING database driver

  IF length(data) > 0 THEN
    FOR each item o IN data DO
      PRINT "find: " + o.a.name + " " + relation + " " + o.b.name
      APPEND (o.a.name, relation, o.b.name) TO triplets
    END FOR
  END IF

  RETURN triplets
END FUNCTION
```

图 6　转换为 Mongodb 查询语句伪代码

Fig. 6　Convert to Mongodb query statement pseudocode

只要存储介质的数据模式满足:主体实体,关系,客体实体之间存在关联关系,那么就可以进行查询语句转换.

3.2.2　实体对齐

对于 3.2.1 所说的查询三元组中的关键字,需要和知识图谱中的实体对齐. 目前最流行的一种方法就是用预训练模型(如 BERT)对图谱中的实体生成嵌入式向量,然后存储到

向量式数据库. 输入要对齐的关键字的嵌入向量,返回相似度最高的实体. 这种方法有一种缺陷,那就是没考虑到问答的上下文环境. 比如属于水果类的“苹果”和属于科技类的“苹果”指的不是一种物品. 因此在生成嵌入向量的时候不仅要考虑到实体的名称,还要考虑其类别. 有两种解决方案:

1)将实体的名称和问题也加入到向量的计算中. 如图 7 所示.

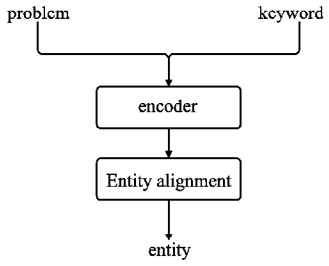

图 7 结合上下文环境进行实体对齐

Fig. 7 Align entities based on contextual environment

2)将不同类别的实体,存到不同的向量数据库实例. 因此在查询与关键字相似度最高的实体的时候,需要根据关键字的类别然后去对应的数据库实例查询,这样就能并行的进行实体对齐,并提高总体的查询速度. 该方法适用于对实体有分类属性的知识图谱.

3.2.3 子图生成方法

通过 3.2.1(查询语句生成)和 3.2.2(实体对齐)的步骤后,将从原始文本中提取出多个结构化的 <主体,关系,客体> 三元组,这些三元组在知识图谱中可以形式化地表示为多个相互连接的强连通子图.

为了弥补知识图谱中可能存在的知识不完善或覆盖不全的问题,本文利用 LLM 强大的语义理解和推理能力,根据当前问题上下文生成语义相关的补充实体. 具体而言,本文采用基于图结构的 k-hop 扩展策略(其中 k 表示跳数,可根据任务复杂度调整),从初始实体集合出发,沿着知识图谱中的关系边进行多跳遍历,探索潜在的语义关联路径,最终得到一个与问题密切相关的扩展子图 G_{sub}.

在获得扩展子图后,将其转换为自然语言形式的逻辑推理链条和对应的子问题 q,并将这些结构化信息输入到大语言模型中生成当前子问题的答案.

为了增强推理的连贯性,本文采用迭代式问答策略,在每一轮迭代中将前一个子问题的答案作为新的证据融入到后续推理过程中. 最终答案将根据逻辑规则的证据链聚合并输入到大语言模型中生成,该算法能够有效整合多轮迭代中产生的所有中间结果,形成完整可靠的最终答案.

3.3 知识子图的融合

子图融合过程包含两个关键步骤:

1)上下文知识集成

在迭代式问答的每一轮推理过程中,将前驱子问题求解阶段所生成的子图(包含经过 k-hop 扩展的实体、关系及其三元组结构)及其对应的推理结果(即前驱子问题的解答)以结构化上下文的形式动态地整合到后续的大语言模型输入中. 具体而言,这些信息会被编码为统一的提示(prompt)格式,其中既包含原始问题的语义信息,也融合了前序推理步骤中积累的中间知识. 这一机制的核心目的在于:1)确保跨轮次推理时的知识传递与继承,避免信息丢失;2)通过显式地维护历史推理状态,增强大语言模型对复杂问题分解与求解的全局一致性理解;3)利用结构化子图与自然语言答案的互补性,为模型提供多模态的上下文线索,从而提升其在后续推理步骤中的语义关联能力和逻辑连贯性. 实验表明,这种基于结构化上下文的知识注入策略能有效缓解大语言模型在长程推理中常见的注意力漂移问题,显著提升多跳问答的准确性和可解释性.

2)知识图谱剪枝

鉴于在 k-hop 扩展过程中构建的子图可能包含大量与当前待解子问题语义关联度较低的冗余实体(如过度扩展的邻域节点)以及非关键的关系路径(如与问题主干无关的旁支推理链),这会导致信息过载并干扰大语言模型的推理聚焦,因此需要执行精细化的知识选择操作.

具体而言,本文设计了一种基于图结构特征与语义相关性的双重过滤机制:首先,通过计算子图中各节点与问题的语义嵌入相似度(如使用余弦相似度衡量),初步筛选潜在相关实体;其次,基于节点的深度(即与初始问题实体的最短路径距离)对子图进行层次化裁剪,优先保留距离核心问题实体在 $k/2$ 跳范围内的关键节点及其关联边;最后,结合节点中心性指标(如 PageRank 值)进一步过滤低重要性节点,从而精炼出与当前问题求解密切相关的核心知识单元(包括关键实体及其支撑性关系路径). 这种层级式的知识选择策略不仅能有效降低计算复杂度,更能确保输入大语言模型的结构化上下文兼具高相关性和高信息密度,从而显著提升多跳推理的精准度和效率.

3.3.1 子图裁剪

在处理当前子问题 q^{i+1} 的过程中,首先以其对应的实体集 $E_q = \{e^1, e^2, \cdots, e^n\}$ 为核心,对知识子图进行深度限制. 具体而言,删除了与 E_q 中实体距离超过深度的所有实体,确保子图仅保留与当前子问题直接相关的局部信息. 这一步骤有效减少了子图的规模,同时保留了关键的结构信息.

接下来,借助 LLM 对子图进行进一步分析. LLM 能够识别并提取出那些对解决子问题 q^{i+1} 没有实质性贡献的逻辑关系. 这些逻辑关系可能是冗余的、无关的,或者与当前问题的求解目标无关. 基于 LLM 的分析结果,删除了所有与这些无用逻辑关系相关联的实体,从而进一步精简了子图的结构.

```
Here is a question and a list of triplets, with the format of<subject,
relation, object>. Return all relationships unrelated to the problem
Question:{question}
Triplets:{triplets}
```

图 8 剔除冗余关系对相关提示词

Fig. 8 Eliminate redundant relationships for relevant prompt words

最终,经过上述两步操作,得到了一个高度简化的子图 G^i_{sub}. 这个简化后的子图不仅规模更小,而且更加聚焦于当前子问题的核心内容,为后续的问题求解提供了清晰且高效的知识表示基础. 这一过程体现了知识图优化与问题求解的紧密结合,显著提升了问题处理的效率和准确性. 伪代码如图 8 所示.

3.3.2 子图的融合

当前子问题将生成知识子图 G_{sub}^{i+1},该子图包含了与当前子问题相关的所有关键节点和关系. 接着,将这个新生成的 G_{sub}^{i+1} 与之前已经经过精简处理的 G_{sub}^{i} 进行合并操作. 在合并的过程中,识别并删除那些冗余的、重复的或者不必要的关系

```
function MergeGraphs(G1, G2, LM, k, Eq)
  // 输入：两个图G1/G2，大语言模型LM，距离阈值k，核心实体集合Eq
  // 输出：合并过滤后的图

  Gm ← GraphUnion(G1, G2)            // 图合并

  // 大模型过滤三元组
  for each triple t in Gm do
    if LM.ShouldDelete(t) then
      Gm.Remove(t)

  // 距离过滤实体
  Entities ← GetAllEntities(Gm)
  for each entity e in Entities do
    if MinDistance(e, CoreEntities) > k then
      RemoveAllTriplesContaining(Gm, e)

  return Gm
end function
```

图 9　子图融合伪代码

Fig. 9　Subgraph fusion pseudocode

对,以确保最终合并后的知识子图既简洁又高效,能够准确地反映问题的核心结构和关键信息. 这一步骤不仅优化了知识子图的结构,还提高了后续问题求解的效率和准确性. 相关提示词如图 9 所示.

4　实　验

为了评估本文中提出的方法,本文在 3 个问答数据集上将本文的方法与其他方法进行了详细的对比.

4.1　数据集

实验中使用的数据集如下:

1) GenMedGPT-5k

它是一个包含 5000 条由 ChatGPT 生成的医患对话的数据集,该数据集由 ChatDoctor[16] 项目创建,结合了从 ChatGPT 生成的对话和疾病数据库中的信息,模拟患者与医生之间的交流

2) LiveQA[17]

该数据集源自 2017 年文本检索会议(TREC 2017)中的医疗问答任务. 该数据集包含 634 个问答对,涵盖 23 种问题类型(例如治疗、原因、诊断、适应症、易感性、剂量)和 4 个类别:疾病、药物、治疗和检查.

3) HealthCareMagic-100k

该数据集来自 HealthCareMagic. com,包含了 10 万例实际患者与医生之间的对话.

实验中用到的知识图谱是 FB15k-237,它是知识图谱 Freebase[18] 的子集,总共有 544230 个三元组.

4.2　实验评价方法

本文采用了 BertScore[19]、Rouge-L[20] 和 BLEURT[21] 3 个指标进行多维度衡量. 这些指标分别从语义匹配、词汇重叠和人类评分模拟等不同角度评估生成文本与参考文本的相似性. 另外还增加了大语言模型评分和疾病诊断和药物推荐问答的获胜率的实验.

大语言模型评分,主要是让大预言模型作为裁判,对每个方法的回答做出综合排名. 然后比较了表现较好的 KgRank,MindMap 和本文方法在疾病诊断和药物推荐方面上回答的质量. 实验所使用的大语言模型有 ERNIE4[4] 和 DeepSeek-R1[3]. ERNIE4 是百度公司自主研发的第 3 代知识增强千亿大模型,ERNIE 3.0 在知识推理任务中超越同等规模的 GPT-3[22]. ERNIE4 在保持 3.0/3.5 版本基础架构的同时,重点提升了逻辑推理和长期记忆能力,其中逻辑能力提升近 3 倍,记忆能力提升超 2 倍. ERNIE 系列通过“知识掩码”技术预训练,显式建模实体间关系,这对 RAG 中知识密集型任务(如事实性问答)尤为重要. 借助 DeepSeek-R1 的深度思考能力来对各模型的回答进行质量评比.

4.3　实验设置

实验中涉及到的超参数为 k-hop 策略中的 k 设为 2. 也就是实体扩展中,检索深度为 2.

实验室中的机器采用 ubuntu22.04 操作系统. 服务器包含了 20-core CPU(型号为:Intel(R) Xeon(R) Gold5115CPU 2.40GHz),251GB RAM,NVIDIA GeForce RTX2080 Ti GPU 11264MiB,NVIDIA RTX A6000 GPU 49140MiB.

其它软件包括:

Neo4j(版本:5.26.2),Mongodb(版本 6.0)和 Mysql(版本:5.7)

4.4　实验结果

本文通过实验与现有代表性方法进行了对比分析. 对比的方法包括 KG-Rank[23]、MindMap[24]、Rok[10]、ERNIE4[4]、DeepseekR1、Embedding Retriever[11-13] 和 Bm25 Retriever[14,15]. Kg-Rank 的核心是对实体初步拓展的关系对进行排序,最后选择那些排序靠前的关系对;MindMap 借助在知识图谱中初步检索的子图通过大语言模型构建“思维导图”,并根据这个思维导图进行推理后生成答案;Rok 利用大语言模型和 PageRank 算法对初步拓展的实体关系对图进行裁剪,留下和问题相关的实体关系对;ERNIE4 和 DeepseekR1 分别代表的是直接用对应的大模型来生成答案;Embedding Retriever 基于稠密向量检索(Dense Retrieval)通过神经网络(如 Sentence-BERT、DPR)将查询和文档映射到同一向量空间,利用余弦相似度衡量语义相关性后进行答案生成;BM25 Retriever 是一种基于统计概率的传统信息检索算法,通过计算查询词与文档的词频(TF)、逆文档频率(IDF)及文档长度归一化来评估相关性.

表 1　GenMedGPT-5k 上的评测结果

Table 1　Review results on GenMedGPT-5k

	BertScore			Rouge			Bleurt
	P	R	F1	P	R	F1	
Ours	**74.4**	**81.0**	77.6	8.9	**28.7**	13.1	**46.8**
MindMap	62.9	78.6	69.9	5.9	25.4	9.4	42.3
Rok	68.7	79.4	73.7	6.0	29.7	9.8	41.2
KGRank	73.8	80.4	76.9	**9.9**	25.8	**13.4**	46.0
ERNIE4	67.2	76.6	71.5	8.2	27.1	9.8	40.4
DeepseekR1	65.1	78.2	71.1	5.8	23.6	9.1	36.5
Bm25 Retriever + LLM	67.2	72.2	69.6	9.2	20.2	12.1	41.9
Embedding Retriever + LLM	67.6	72.4	69.9	9.8	20.3	12.6	43.1

表 1 显示了在 GenMedGPT 上的生成质量(Ours 为本文提出的方法). 实验结果表明,本文提出的方法在大多数指标

上均优于或接近最优表现,尤其在语义一致性(BertScore)和生成质量(BLEURT)方面表现突出. 在 LiveQA 上的实验结果如表 2 所示,表明本文提出的方法在语义一致性(BertScore)和生成质量(BLEURT)方面表现最优,同时在 Rouge 指标上也保持竞争力. 本文提出的方法在 HealthCareMagic-100k 数据集上的表现也明显优于其他方法(如表 3 所示).

表 2 LiveQA 上的评测结果

Table 2 Review results on LiveQA

	BertScore			Rouge			Bleurt
	P	R	F1	P	R	F1	
Ours	**70.1**	76.7	**73.1**	**10.8**	23.5	**11.0**	**40.6**
MindMap	66.0	76.7	70.8	7.5	24.6	9.6	38.7
Rok	68.5	**77.8**	72.7	7.9	**28.7**	9.7	39.8
KGRank	68.1	75.5	71.5	10.7	23.0	10.6	39.7
ERNIE4	67.2	76.6	71.5	8.20	27.1	9.8	37.3
DeepseekR1	65.9	76.7	70.7	7.3	22.6	8.5	34.6
Bm25 Retriever + LLM	62.2	69.4	65.5	8.6	22.4	9.5	36.1
Embedding Retriever + LLM	61.6	68.8	64.9	8.2	22.2	9.3	35.8

表 3 HealthCareMagic-100k 上的评测结果

Table 3 Review results on HealthCareMagic-100k

	BertScore			Rouge			Bleurt
	P	R	F1	P	R	F1	
Ours	**72.3**	**76.9**	**74.5**	8.4	22.7	**11.9**	**43.3**
MindMap	63.8	74.1	68.6	5.6	17.9	8.4	40.3
Rok	67.8	75.1	71.2	**10.2**	**23.8**	8.8	41.2
KGRank	68.8	72.6	70.6	7.8	17.6	9.6	41.3
ERNIE4	67.2	76.6	71.5	8.2	27.1	9.8	37.3
DeepseekR1	68.1	76.1	71.8	7.5	19.5	9.6	36.9
Bm25 Retriever + LLM	64.9	68.7	66.7	7.8	18.4	10.4	38.7
Embedding Retriever + LLM	65.6	68.8	67.1	8.6	17.7	10.5	38.6

从图 10 ~ 图 12 中可以更为直观地看出各方法的表现.

表 4 展示了以 DeepSeek-R1 作为医生,给对比的各个方法的回答做出的排名数据. 表 5 展示了 DeepSeek-R1 作为医学专家对比本文和其他方法在疾病诊断和药物推荐上的质量. 根据表 4 和表 5 中的结果,本文的方法的表现明显优于其他方法,大语言模型评分排名与其他方法相比排名较高,幻觉评分较低. 这凸显了本文与其他模型相比,能够生成基于证据、合理和准确的答案,因为这些模型可能会因依赖隐性知识而产生不正确的答案. 此外,表 5 展示了本文相较于其他方法的持续优势,显示了整合外部知识以减轻大语言模型幻觉并提供准确答案的价值.

表 4 大语言模型评分

Table 4 Large language model rating

	LiveQA	GenMedGPT-5k
Ours	**1.76**	**1.88**
MindMap	3.14	2.80
Rok	1.93	2.29
KgRank	2.72	2.35

表 5 疾病诊断和药物推荐问答的获胜率进行配对比较

Table 5 Pairing and comparing the winning rates of disease diagnosis and drug recommendation

	KgRank			MindMap			Rok		
	Win	Tie	Loss	Win	Tie	Loss	Win	Tie	Loss
Avg	63.29	6.32	30.37	60.75	3.79	35.44	69.62	1.26	29.11

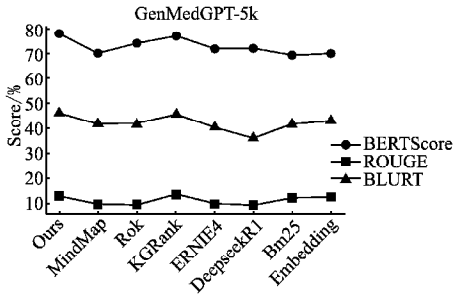

图 10 GenMedGPT-5k 上的评测结果

Fig. 10 Review results on GenMedGPT-5k

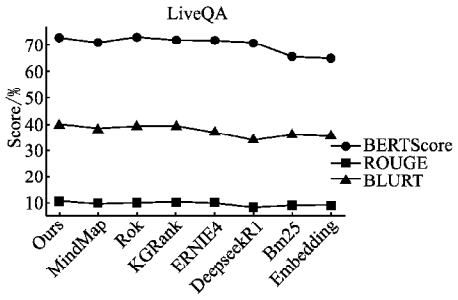

图 11 LiveQA 上的评测结果

Fig. 11 Review results on LiveQA

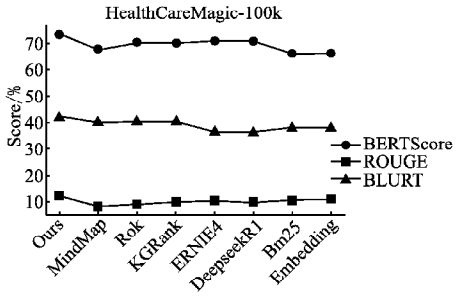

图 12 HealthCareMagic-100k 上的评测结果

Fig. 12 Review results on HealthCareMagic-100k

4.5 示例

输入:Doctor,I have been experiencing a hoarse voice for a few weeks now and it's not getting any better despite taking medication. What could be the problem?

本文模型输出:The **persistent hoarse voice** despite medication could be due to various causes, such as **vocal cord nodules**, polyps, **pharyngitis**, or inflammation. To diagnose the underlying cause, a thorough examination of your vocal cords, possibly using a **laryngoscope**, may be necessary. Treatment options can vary depending on the diagnosis but may include **voice therapy**, **lifestyle modifications**, and in some cases, **surgical intervention**. It's essential to consult with a specialist to determine the appropriate course of action.

黑色加粗的关键字是本文模型从知识图谱以及在知识图谱知识欠缺的情况下利用大语言模型扩充的知识.

5 结论与展望

本文在医疗场景提出了一种智能问答解决方案.可以适用于不同存储介质的知识图谱,通过三元组的形式减少大语言模型生成查询语句的错误率.通过融合知识子图的方法有效提高大语言模型对问题的处理能力.根据多个数据集上的表现,本文提出的方法比没有外部知识库的大语言模型和其他增强检索的生成方法表现更加出色.

大语言模型的引入显著提升了智能问答系统的语义理解与内容生成质量.通过其强大的预训练知识库和上下文建模能力,大语言模型能够生成结构完整、信息丰富的回答,尤其在处理开放域复杂查询时展现出传统方法难以企及的优势.然而,这种性能提升伴随着显著的计算资源消耗:单次推理过程涉及数百亿参数的实时运算,导致响应延迟随模型参数量级呈指数增长.当处理流程需要多次调用 LLM 进行迭代优化时,系统整体响应时间将产生不可忽视的累积效应.这种时延特性使得 LLM 的应用场景呈现明显的任务依赖性.在离线批处理任务(如文献综述生成、知识图谱构建)中,系统可以通过牺牲时效性来换取最优解答质量,时间成本往往被视为次要优化目标.但面对在线实时交互场景(如客服系统、即时问答平台),响应速度与结果质量同样构成核心用户体验指标.当前多数研究聚焦于通过模型微调、提示工程等手段提升回答准确性,却普遍忽视了问题处理耗时的优化探索.如何构建动态推理框架,在模型参数量、调用次数与响应延迟之间实现自适应平衡,将成为下一代智能问答系统的关键技术挑战.

References:

[1] Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report[J]. arxiv preprint arxiv:230308774, 2023.

[2] Dubey A, Jauhri A, Pandey A, et al. The llama 3 herd of models [J]. arxiv:2407.21783, 2024.

[3] Deepseek A I, Guo D, Yang D, et al. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning[J]. arxiv. org/abs/2501.12948, 2025.

[4] Sun Y, Wang S, Li Y, et al. ERNIE: enhanced representation through knowledge integration[J]. arxiv. org/abs/1904.09223, 2019.

[5] Xie Q, Schenck E J, Yang H S, et al. Faithful AI in medicine: a systematic review with large language models and beyond[J]. MedRxiv, 2023.04.18.23288752, doi:10.1101/2023.04.18.23288752.

[6] Liu P, Yuan W, Fu J, et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing [J]. ACM Computing Surveys, 2023, 55(9): 1-35.

[7] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks[C]//Advances in Neural Information Processing Systems, 2020: 9459-9474.

[8] Rau D, Wang S, Dejean H, et al. Context embeddings for efficient answer generation in retrieval-augmented generation[C]//Proceedings of the Web Search and Data Mining, 2025: 493-502.

[9] Li X, Zhao R, Chia Y K, et al. Chain of knowledge: a framework for grounding large language models with structured knowledge bases[J]. arxiv. org/abs/2305.13269, 2023.

[10] Jiang B, Wang Y, Luo Y, et al. Reasoning on efficient knowledge paths: knowledge graph guides large language model for domain question answering [C]//Proceedings of the IEEE International Conference on Knowledge Graph (ICKG), 2024: 142-149.

[11] Roberts A, Raffel C, Shazeer N J A P A. How much knowledge can you pack into the parameters of a language model? [J]. arxiv preprint arxiv, 2020.08910.

[12] Robertson S, Zaragoza H J F, Retrieval T I I. The probabilistic relevance framework: BM25 and beyond[J]. Foundations and Trends in Information Retrieval, 2009, 3(4): 333-389.

[13] Peng B, Galley M, He P, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback[J]. arxiv Preprint arxiv, 2023.12813.

[14] Kamphais C, Vries A P D, Boytsov L, et al. Which BM25 do you mean? A large-scale reproducibility study of scoring variants [C]//European Conference on Information Retrieval (ECIR), 2020: 28-34.

[15] Sharma A, Kumar S J D, Engineering K. Ontology-based semantic retrieval of documents using Word2vec model[J]. Data & Knowledge Engineering, 2023, 144: 102110, doi: 10.1016/j.datak.2022.102110.

[16] Li Y, Li Z, Zhang K, et al. ChatDoctor: a medical chat model fine-tuned on a large language model meta-AI (LLaMA) using medical domain knowledge[J]. Cureus, 2023, 15(6): e40895, doi: 10.7759/cureus.40895.

[17] Abacha A B, Agichtein E, Pinter Y, et al. Overview of the medical question answering task at TREC 2017 LiveQA[C]//Proceedings of the Text REtrieval Conference (TREC), 2017: 1-12.

[18] Bollacker K D, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the ACM Sigmod International Conference on Management of Data, 2008: 1247-1250.

[19] Zhang Z, Han X, Liu Z, et al. ERNIE: enhanced language representation with informative entities [J]. arxiv preprint arxiv: 1905.07129, 2019.

[20] Lin C Y. ROUGE: a package for automatic evaluation of summaries [C]//Proceedings of Workshop on Text Summarization Branches Out, 2004: 74-81.

[21] Sellam T, Das D, Parikh A P. BLEURT: learning robust metrics for text generation[J]. arxiv preprint arxiv, 2004.04696.

[22] Sun Y, Wang S, Feng S, et al. ERNIE 3.0: large-scale knowledge enhanced pre-training for language understanding and generation [J]. arxiv. org/abs/2107.02137, 2021.

[23] Yang R, Liu H, Marrese Taylor E, et al. KG-Rank: enhancing large language models for medical QA with knowledge graphs and ranking techniques[J]. arxiv preprint arxiv, 2403.05881, 2024.

[24] Wen Y, Wang Z, Sun J. MindMap: knowledge graph prompting sparks graph of thoughts in large language models[J]. arxiv preprint arxiv, 2308.09729, 2023.