计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2026,35(3):13−22 [doi: 10.15888/j.cnki.csa.010102] [CSTR: 32024.14.csa.010102]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# IHRAG: 面向 LLM 的迭代式混合检索增强生成①

谢雨霏[1], 李　琳[1,2], 李　涛[1,2], 何　柳[3], 高贝琳[4], 何志婷[5]

[1](武汉科技大学 计算机科学与技术学院, 武汉 430065)
[2](武汉科技大学 智能信息处理与实时工业系统湖北省重点实验室, 武汉 430065)
[3](武昌理工学院 商学院, 武汉 430223)
[4](武汉纺织大学 伯明翰时尚创意学院, 武汉 430073)
[5](武汉纺织大学 管理学院, 武汉 430073)
通信作者: 李　琳, E-mail: lilin@wust.edu.cn

**摘　要**: 在医疗领域, 检索增强生成 (RAG) 被提出以减少大语言模型幻觉, 并提供更多的可解释性和可控性, 然而现有技术面临对低频实体的召回能力较弱、难以处理模糊冗长或多义性强的查询的问题, 本文提出一种面向大语言模型的迭代式混合检索增强生成 (iterative hybrid retrieval-augmented generation, IHRAG) 方法以提升对复杂问题的意图解析能力, 增强模型在知识挖掘方面的表现, 使大语言模型生成更加准确的回答. 该框架通过动态路由机制协同调度向量检索的语义泛化能力与知识图谱的结构化推理能力, 结合医疗本体驱动的查询解构算法, 将复杂临床问题分解为可检索的原子子问题, 并引入知识缺口感知的神经符号扩展模型与“检索-验证-迭代”闭环优化机制, 构建了从表层信息提取到深层知识挖掘的递进式发现流程. 实验结果表明, IHRAG 在 Qwen、DeepSeek 等不同规模基础模型上均显著提升性能, 最高可使准确性提升 11.12 个百分点, 优秀回答率提升 17 个百分点.

**关键词**: 大语言模型; 检索增强生成; 知识图谱; 混合检索; 医疗问答

引用格式: 谢雨霏,李琳,李涛,何柳,高贝琳,何志婷.IHRAG: 面向 LLM 的迭代式混合检索增强生成.计算机系统应用,2026,35(3):13–22. http://www.c-s-a.org.cn/1003-3254/10102.html

## IHRAG: Iterative Hybrid Retrieval-augmented Generation for Large Language Model

XIE Yu-Fei[1], LI Lin[1,2], LI Tao[1,2], HE Liu[3], GAO Bei-Lin[4], HE Zhi-Ting[5]

[1](School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)
[2](Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China)
[3](School of Business, Wuchang University of Technology, Wuhan 430223, China)
[4](Birmingham Institute of Fashion and Creativity Art, Wuhan Textile University, Wuhan 430073, China)
[5](School of Management, Wuhan Textile University, Wuhan 430073, China)

**Abstract**: In the medical field, retrieval-augmented generation (RAG) has been proposed to mitigate hallucinations in large language model (LLM) and enhance the interpretability and controllability. However, existing techniques are faced with poor recall of low-frequency entities and difficulties in processing ambiguous, verbose, or polysemous queries. To this end, this study proposes an iterative hybrid retrieval-augmented generation (IHRAG) approach for LLM to improve the intention parsing ability of complex queries and enhance the model's performance in knowledge mining capabilities for making LLMs generate more accurate responses. IHRAG employs a dynamic routing mechanism to synergistically leverage the semantic generalization capability of vector retrieval andthe structured reasoning capacity of knowledge graphs. By combining a medical ontology-driven query decomposition algorithm, complex clinical questionsare broken down into retrievable atomic sub-questions. Furthermore, a knowledge gap-aware neuro-symbolic expansion model and a “retrieve-verify-iterate” closed-loop optimization mechanism are introduced to establish a progressive discovery process

① 基金项目: 武汉市重点研发计划 (2022012202015070); 武汉东湖新技术开发区“揭榜挂帅”项目 (2022KJB126)
收稿时间: 2025-09-03; 修改时间: 2025-09-29; 采用时间: 2025-10-14; csa 在线出版时间: 2026-01-19
CNKI 网络首发时间: 2026-01-20

that advances from surface-level information extraction to deep knowledge mining. Experiments demonstrate that IHRAG significantly enhances the performance of base models of various scales such as Qwen and DeepSeek, achieving an improvement in the accuracy of up to 11.12 percentage points and 17 percentage points increase in the high-quality response rate.

**Key words**: large language model (LLM); retrieval-augmented generation (RAG); knowledge graph (KG); hybrid retrieval; medical Q&A

近年来, 大语言模型 (large language model, LLM) 在金融、法律、医学等知识密集型领域的应用显著增长[1]. 然而, 研究发现 LLM 在医疗领域的应用面临着一系列挑战[2,3]. 先前研究表明, 大语言模型对低频实体的记忆能力有限[4], 容易产生幻觉[5], 并可能出现时效性退化[6], 这些缺陷可能导致临床决策失误, 引发安全隐患. 因此整合和理解多源知识的认知能力成为医疗问答系统的关键需求[7].

检索增强生成 (retrieval-augmented generation, RAG) 应运而生, 用于增强生成器, 以减少幻觉, 并提供更多的可解释性和可控性[8]. 且现在多数医疗数据属于私有数据, 这种情况下 RAG 是一个方便有效的工具, 能够实现私有数据库. 然而现有技术面临瓶颈: 向量检索虽具备语义泛化能力, 但对低频或罕见医学实体的召回能力较弱, 容易遗漏关键信息; 知识图谱检索支持结构化推理, 但受限于文本覆盖狭窄, 难以解析非结构化临床表述; 静态混合检索虽整合异构知识源, 却因简单拼接引发知识冗余及响应不一致. 而实际医疗场景中的用户查询常包含隐含的多重临床意图 (如病因溯源、用药禁忌、预后评估等), 这就需要丰富全面的回答[9].

因此, 本研究提出了一种融合 Vector-KG RAG 迭代式问答的方法, 该方法融合了基于稠密向量检索的浅层语义匹配与基于知识图谱的深度推理验证. 系统架构遵循“检索-验证-迭代优化”的认知计算模型, 实现了从表层信息获取到深层知识挖掘的递进式知识发现流程. 混合 Vector-KG 方法弥补了向量相似性和知识相关性之间的差距, 从而提高生成响应的准确性、相关性和完整性. 本研究的主要贡献是: (1) 提出一种混合模型, 建立混合知识源的证据加权融合模型. (2) 提出知识缺口驱动的动态查询扩展理论. (3) 验证迭代式检索在复杂 Q&A 任务中的有效提升. 该框架目的是解决医疗问答中准确性-可扩展性的固有权衡问题, 降低专业知识获取门槛, 推动医疗决策支持的普惠化发展.

## 1 相关工作

RAG 能够快速且动态地适应最新的或特定的外部知识, 并在生成过程中从外部来源检索相关信息[10]. 为了确保 RAG 系统的有效性, 以往的研究主要同时优化检索器和生成器 (文献[11–15]).

基于知识源的异构性, RAG 方法可分为两类. (1) 基于向量的 RAG (Vector-RAG), 如 T-RAG[16]、ICCA-RAG[17]、INFO-RAG[18]. Vector-RAG 在需要从相关文本文档中提取上下文以生成有意义且连贯的回答的情况下表现出色[19]. 向量检索依赖稠密向量匹配实现语义泛化, 但在低频实体检索方面存在短板. (2) 基于图的 RAG, 如 GraphRAG[20]、GraphReader[21]、Medical Graph RAG[22]. 知识图谱已成为数据管理和分析中的关键技术, 并已在包括搜索引擎、推荐系统和生物医学研究在内的多个领域得到广泛应用[23,24]. 知识图谱检索利用结构化关系支持推理和精确实体召回, 但对自然语言的适应性较弱, 难以处理模糊、冗长或多义性强的查询.

为融合二者优势, 静态混合检索模型 (如 Hybrid-RAG) 被提出, 但简单拼接机制易引发知识冗余与逻辑冲突. 因此本文提出迭代式混合检索增强生成框架, 提升低频实体召回率及多源知识融合的准确性, 从而增强复杂问题的推理与挖掘能力, 提高回答的准确性和完整性.

## 2 迭代式混合检索方法

现有 RAG 方式有效地解决了大语言模型中的静态知识和幻觉问题, 但是现有的研究大多集中在具有明确的用户意图和简洁的答案的问题场景上. 然而, 实际使用时, 用户的问题一般是有多个子意图的开放式问题, 因此上述方式具有很大局限性. 本研究提出迭代式混合检索增强生成框架 (iterative hybrid retrieval-augmented generation, IHRAG), 通过知识融合和动态

迭代优化机制实现问答系统的性能提升. 使用 Langchain 工具, 通过 text-embedding 模型将文本数据转换为高维度向量表示. 此矢量化过程支持语义相似性搜索, 从而提高检索信息的相关性和准确性. 通过从域文档中提取元数据, 然后构建基于元数据的知识图谱. 将这些元数据映射到 head-entity-tail 关系的有向三元组, 构建以 Neo4j 为载体的知识图谱, 通过有向三元组实现文本块与结构化知识的协同表示.

本研究的技术路线采用多阶段递进式架构, 如图 1 所示, 通过系统的流程设计实现医疗知识的高效检索与智能生成.

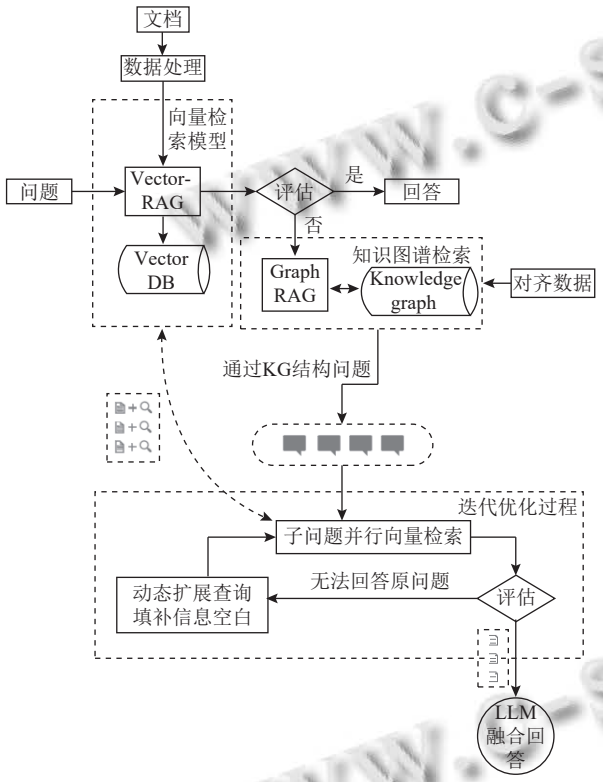



图 1 IHRAG 的整体框架

### 2.1 动态查询扩展与迭代算法 (动态路由检索模块)

本系统采用基于大语言模型 (LLM) 的动态查询扩展机制与马尔可夫决策过程 (MDP) 框架结合的迭代优化算法, 旨在提升信息覆盖的质量、灵活性与效率. 检索过程建模为 MDP 五元组.

状态空间 $S$: 知识状态 $K_t$ 的集合, 其中, $K_t=\{(q_i, D_i, a_i)\}$表示第 $t$ 轮迭代积累的问题-文档-答案三元组.

动作空间 $A$: {检索终止, 生成扩展查询, 调整检索策略}.

状态转移概率 $P$: $P(K_{t+1}|K_t, A_t)$ 表示在状态 $K_t$ 执行动作 $A_t$ 后转移到 $K_{t+1}$ 的概率.

奖励函数 $R$: $R(K_t, A_t, K_{t+1})=ROUGE\text{-}L(A_{t+1})+\lambda\cdot FactScore(A_{t+1})-\eta\cdot Cost(t)$.

折扣因子 $\gamma$: 设为 0.9, 平衡即时与远期奖励.

对于答案 $a$, 其充分性度量为:

$$Sufficiency(a)=\left(1-P_{\text{neg}}\right)\cdot\left(\frac{1}{n}\right)\cdot\sum_{\{i=1\}}^{n}RelevanceScore(d_i,q) \quad (1)$$

其中, $P_{\text{neg}}$ 为否定模式概率, 基于预定义的否定标记集 $M$={无法确定, 无相关信息, 不详, 未找到, 无法回答}计算, $RelevanceScore(d_i, q)\in[0, 1]$为文档 $d_i$ 与问题 $q$ 的语义相关度, $n$ 为检索文档数量. 设置充分性阈值 $\theta$=0.7.

基于知识状态 $K_t$ 的信息缺口定义为:

$$Gap(K_t)=1-(Coverage(K_t)/Coverage\_ideal) \quad (2)$$

其中, $Coverage(K_t)=\sum_{e\in E(q)}\min(1,Count_{K_t}(e))$表示当前知识状态对问题实体的覆盖度, $Coverage_ideal=|E(q)|$为理想覆盖度. 设置信息缺口阈值 $\tau$=0.2.

首先, 系统利用 LLM 的注意力机制识别信息缺口, 通过设计特定的 Prompt, 基于已有的问题-答案对集合$\{(q_i,a_i)\}_{i=1}^{n}$, 生成补充查询集 $Q'$. 通过自适应注意力机制, 系统动态计算语义覆盖不足的区域, 确保能够生成具有针对性的补充查询. 此过程不仅提升了信息的覆盖质量, 同时也通过上下文自适应调整检索策略, 增强了检索的灵活性与智能化水平. 系统通过计算已有答案集合的信息熵来量化当前知识缺口, 从而优化补充查询的生成, 确保查询的精准性.

在算法实现层面, 查询扩展过程表现为一个闭环控制系统. 系统不断监控检索结果的知识覆盖情况, 当检测到关键临床概念的缺失或证据强度不足时, 自动触发查询重写机制. 该自适应能力使得系统能够根据不同的临床场景调整检索策略, 保证结果的相关性和知识获取的效率. 通过一个递归优化的迭代过程, 在每轮迭代中, 系统的知识状态 $K_t$ 作为当前的上下文环境, 确保该过程不受历史迭代的影响, 从而避免冗余检索和计算资源的浪费. 迭代的停止准则结合信息论度量和阈值, 当新增知识的预期效用低于设定的风险阈值时, 系统会自动终止迭代, 以此平衡知识获取的深度和效率.

迭代过程在满足以下任一条件时终止:

$Sufficiency(a_t)\geqslant\theta$ (答案充分性达标);

$Gap(K_t) < \tau$ (信息缺口足够小);

$t \geqslant T$ (达到最大迭代次数).

通过一个递归优化的迭代过程, 在每轮迭代中, 系统的知识状态 $K_t$ 作为当前的上下文环境, 确保该过程不受历史迭代的影响, 从而避免冗余检索和计算资源的浪费. 迭代的停止准则结合信息论度量和阈值, 当新增知识的预期效用低于设定的风险阈值时, 系统会自动终止迭代, 以此平衡知识获取的深度和效率. 详细算法见附录 A 中的算法 A2.

### 2.2 知识图谱结构问题 (本体化解构模块)

首先借助现有知识图谱构建工具从域文档中提取元数据, 然后构建基于元数据的知识图谱. 该部分旨在解决医疗领域特有的复合型查询挑战, 通过结构化分解将包含多重意图的自然语言问题转化为系列原子查询, 为后续检索环节提供清晰明确的目标指向. 对于问题 $q$, 其复杂度定义为实体-关系图函数.

$$Complexity(q) = |E(q)| + \lambda \cdot |R(q)| \tag{3}$$

其中, $E(q)$ 为问题中提取的实体集合, $R(q)$ 为实体间关系集合, $\lambda$ 为关系权重系数 (设为 0.7). 设置复杂度阈值 $\delta$=5, 当 $Complexity(q) \geqslant \delta$ 时判定为复杂问题, 触发深度解构流程. 复杂度越高, 表示问题包含的实体和属性越多, 需要进行解构的可能性越大.

利用知识图谱中实体及其属性的层级结构信息, 实现复杂医学问题的语义拆解, 如图 2 所示. 从问题 $q$ 中抽取实体集合 $E(q)$, 基于查询中实体的类型调用对应的属性集合 $P(type(e_i))$, 进而利用自然语言生成函数构造针对每个实体-属性对的子问题集.

$$Decompose(q) = \{\varphi(e_i, p_j) | e_i \in E(q), p_j \in P(type(e_i))\} \tag{4}$$

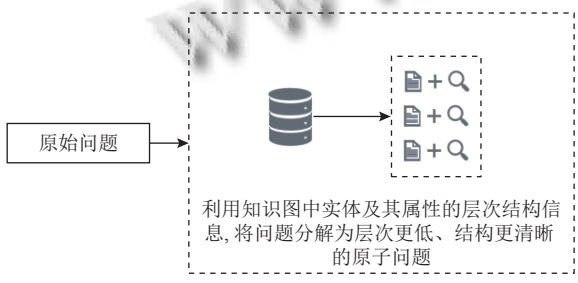



图 2 知识图谱拆解问题流程

具体而言, 例如问题: “糖尿病患者服用二甲双胍后出现肾功能异常的可能原因”, 可分解为多个原子查询: 1) 二甲双胍的药物代谢途径、肾脏清除机制; 2) 不同糖尿病继发肾病病理特征; 3) 二甲双胍相关肾毒性临床证据. 多粒度解构策略能同时处理单实体单属性查询和多实体关系链推理, 将复杂的多跳临床推理路径通过路径表达式重写技术转换为可执行的子查询序列. 详细算法见附录 A 中的算法 A3.

### 2.3 融合 Vector-KG RAG (自适应融合权重模块)

混合检索框架建立在知识表示学习的理论基础上, 通过构建统一的向量-符号联合空间实现异构知识的深度融合. 该空间的设计遵循 3 个核心原则: 临床概念的可解释性、知识来源的可追溯性以及推理过程的可验证性.

通过构建向量存储、基于元数据的 KG 和约束提示, 提出了一种混合 Vector-KG RAG 方法. 这种方法利用 KG 捕获的显式关系以及从向量存储中检索到的丰富但非结构化的信息来生成准确而全面的响应. 使用命名实体识别 (NER) 从提示中提取关键实体, 以便有效地检索知识. 然后, 提取的实体用于 KG-RAG 流程, 该流程在涉及节点检索和构建的元数据驱动的 KG 中进行关系检索. KG-RAG 流程检索相关节点及其相互关系, 提供与查询相关的相关源的初步概述. 通常, KG-RAG 结果会将用户定向到 KG 中与其查询最相关的特定文档或部分. 但是, 要获得更详细的答案, 需在矢量库中搜索包含深入信息的特定文本块. 这些文本块是根据其与查询的向量相似性来检索的, 从而确保内容既相关又详细. 最后, 将 KG-RAG 和 Vector-RAG 的结果整合在一起, 形成混合响应. 具体步骤如算法 1 所示.

算法 1. 迭代式混合检索增强生成框架

输入: 用户查询 $q_0$, 最大迭代次数 $T$, 充分性阈值 $\theta$, 信息缺口阈值 $\tau$, 实验组标识 $\Gamma$.
输出: 最终答案.

1. $K \leftarrow \varnothing$ // 初始化知识状态
2. $Q \leftarrow Queue(q_0)$ // 初始化查询队列 (先进先出)
3. $t \leftarrow 0$ // 迭代计数器
4. $\alpha \leftarrow \alpha_0$ // 静态混合权重 (仅 HybridRAG 组使用)
5. while $Q$ 非空且 $t<T$ do
6. $q \leftarrow$出队 ($Q$)
   // === 实验对照组分支 ===
7. if $\Gamma$=LLM-only then // 基线 1: 单一大语言模型回答
8. $A \leftarrow$LLM 生成 ($q$, $\varnothing$)
9. $K \leftarrow K \cup \{(q, \varnothing, A)\}$
10. else if $\Gamma \in$ {Vector-RAG, KG-RAG} then // 基线 2、3: 单一检索及单一知识图谱检索生成回答
11. if $\Gamma$=Vector-RAG then

```
        D←稠密检索 (q) // 仅通过向量检索
    else
        D←知识图谱查询 (q) // 仅通过图谱检索
    A←LLM 生成 (q, D)
    K←K∪{(q, D, A)}
else if Γ=HybridRAG then // 基线 4: 静态混合
    D_vec←稠密检索 (q)
    D_kg←知识图谱查询 (q)
    D←α·D_vec⊕(1−α)·D_kg // 固定权重融合生成回答
    A←LLM 生成 (q, D)
    K←K∪{(q, D, A)}
else if Γ=IHRAG then // 实验组: 动态混合框架
    // 动态路由
    if 复杂度 (q)<δ then // 简单问题查询
        D←稠密检索 (q, K) // 神经 IR 路径
    else
        D_kg←知识图谱查询 (q, K), 并将复杂问题生成细化子问题
        D_vec←稠密检索 (q, K)
        α_t←f(熵 (q), |实体 (q)|) // 自适应融合比
        D←α_t·D_vec⊕(1−α_t)·D_kg // 动态融合
    // 假设生成与验证
    假设集 A_h←∅
    for k=1 to N_假设 do
        A_h←A_h∪LLM 生成 (q, D∪负采样 (D))
    end for
    A←arg max_{h∈A_h} 验证 (h, D)
    K←K∪{(q, D, A)} // 状态更新
    // 终止条件检测
    if 充分性 (A)≥θ or 信息缺口 (K)<τ then break
    // 神经符号扩展
    知识缺口←缺口检测 (K)
    补充问题查询集Q′←∅
    for 每个概念 c ∈知识缺口 do //遍历知识缺口中的每个概念
        q′←本体扩展 (q, c) // 基于概念 c 对原始问题 q 进行补充扩展
        Q′←Q′∪{q′}
    end for
    入队 (Q, Q′)
end if
t←t+1
end while
```

## 3 实验与分析

本研究采用系统性实验方法验证关键技术方案的有效性, 通过多维度评估框架确保研究结论的可靠性. 具体研究步骤如下: 1) 数据准备: 在公开数据集和收集文档资料中, 预处理后形成初步的整体框架模型所需的数据集, 并划分出测试集; 2) 构造知识库: 将文档中提取到的文本数据构造为特定领域的 KG 和向量知识库; 3) 比较 RAG 方法: 对 5 种方法进行比较, 包括单一大语言模型、单一向量检索系统、单一知识图谱检索、迭代式混合检索以及本文的 IHRAG; 4) 评估结果: 确定评估指标, 评估实验结果, 验证本方法的准确性和泛化性.

### 3.1 数据集构建

本文实验测试数据集包括 3 部分.

(1) 公开数据集: 主要是中文医疗数据集 Huatuo[25] 和 cMedQA[26].

(2) 私有数据集: 包括真实医疗模拟测试题以及人工构造的复杂场景测试用例. 我们选择了专门的社区案例文本手册和药品的说明书来构成数据集的核心. 这些文档包含药物的功能主治、不良反应、疾病症状和注意事项等. 将数据扩展, 通过大语言模型分两轮构建数据集和问答对. 最后, 通过使用 Langchain 的工具链处理原始 PDF 文档来获得文本数据, 该文档提取和分割文本内容以供后续处理. 在分解和分析数据集后, 采取后处理步骤来揭示和完善主题集群.

(3) 混合数据集: 该数据集通过抽样策略, 从公开数据集和私有数据集中抽取样本, 确保覆盖多样化的医疗问题类型. 具体而言, 从公开数据集中抽取具有代表性的常见医疗问答对, 同时从私有数据集中选取包含复杂场景和低频实体的案例.

### 3.2 基础模型与基线方法

构建了对比实验方案来评估不同模态组合的检索效果.

(1) 核心基础模型: 选择 4 个模型 Qwen2.5-1.5B、Qwen-Turbo、DeepSeek-V3 和 Qwen3-235B.

(2) 对照组基线方法: 将对照组方法分别设置为基线 1、基线 2、基线 3 和基线 4.

基线 1: LLM-only (单一大语言模型) 即直接使用基础模型生成回答, 评估大语言模型本身的能力.

基线 2: Vector-RAG (单一向量检索系统) 即借助 Langchain 工具, 链接 LLM 进行向量检索.

基线 3: KG-RAG (单一知识图谱检索) 即基于 Neo4j, 支持 Cypher 查询语言的扩展版本.

基线 4: HybridRAG 即一种静态权重融合的中间方案.

(3) 实验组为本研究提出的迭代式混合检索框架 IHRAG.

IHRAG 框架的改进体现在: (1) 动态路由机制: 根据查询复杂度自动选择检索路径 (简单查询使用向量检索, 复杂查询触发图谱推理); (2) 迭代验证循环: 通过

假设生成与证据检索的交替进行实现答案优化. 关键算法伪代码见算法 1, 数据预处理、模型训练、自动评估伪代码分别见附录 A 中的算法 A1、算法 A2–A4 以及算法 A5.

### 3.3 评估指标

本研究建立的评估框架基于医疗决策支持系统的特殊需求, 采用多维度、多层次的评估方法, 确保系统输出的科学性、可靠性和临床实用性. 评估过程结合自动化评估工具 (DeepSeek-R1) 和专家人工评审, 形成互补验证机制. 自动评估: 使用 DeepSeek-R1 进行初步评分, 生成评估报告. 评估代码基于提供的评分函数实现, 采用统一的 Prompt 模板确保评分一致性. 人工评审: 两位评审员根据知识库文档对作答进行独立评分.

(1) 相关性 (relevance, Rel): 相关性评估系统响应与用户查询在语义层面的匹配程度, 在医疗场景下, 相关性不仅要求表面语义的匹配, 更需要深层次的临床语境理解.

(2) 完整性 (completeness, Com): 完整性评估源自问答的医疗决策理论的要素完整情况, 强调医疗决策的系统性和全面性.

(3) 准确性 (accuracy, Acc): 准确性评估建立在循证医学原则基础上, 对质量进行分级, 评估回答可靠程度.

### 3.4 实验结果

通过对 4 种不同规模基础模型 (Qwen2.5-1.5B、Qwen-Turbo、DeepSeek-V3、Qwen3-235B) 在 LLM-only、KG-RAG、Vector-RAG、HybridRAG 及 IHRAG 这 5 种方法下的系统性评测, 实验在混合数据集、私有数据集和公开数据集进行测试, 为全面评估检索增强方法的有效性, 在 DeepSeek-V3 基础上额外引入 GraphRAG 作为对照方法. 实验数据如表 1 所示.

表 1 所有模型的总体实验结果 (%)

| 基础模型 | 基线方法 | 混合数据集 | | | 私有数据集 | | | 公开数据集 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | Com | Rel | Acc | Com | Rel | Acc | Com | Rel |
| Qwen2.5-1.5B | LLM-only | 54.95 | 60.21 | 70.74 | 56.53 | 60.25 | 72.25 | 55.15 | 63.46 | 70.92 |
| | KG-RAG | 63.66 | 68.47 | 80.10 | 69.07 | 70.26 | 82.07 | 56.15 | 64.61 | 74.86 |
| | Vector-RAG | 63.75 | 68.35 | 79.08 | 68.96 | 71.40 | 82.39 | 55.38 | 62.76 | 72.69 |
| | HybridRAG | 64.54 | 69.83 | 80.58 | 70.53 | 71.20 | 84.23 | 57.10 | **65.67** | 75.07 |
| | IHRAG | **66.07** | **69.88** | **80.94** | **74.52** | **72.21** | **84.55** | **60.24** | 65.30 | **75.30** |
| Qwen-Turbo | LLM-only | 72.95 | 80.20 | 85.78 | 72.90 | 80.08 | 85.92 | 72.12 | 80.68 | 85.86 |
| | KG-RAG | 73.06 | 72.87 | 84.64 | 75.34 | 72.64 | 85.85 | 74.43 | 76.28 | 85.94 |
| | Vector-RAG | 71.75 | 69.42 | 76.64 | 73.91 | 74.30 | 79.50 | 75.86 | 82.92 | 87.98 |
| | HybridRAG | 81.10 | 81.64 | 88.41 | 82.49 | **86.42** | 89.96 | 80.20 | 83.63 | 88.76 |
| | IHRAG | **82.31** | **84.56** | **90.65** | **84.67** | 86.39 | **92.14** | **81.65** | **83.78** | **89.20** |
| DeepSeek-V3 | LLM-only | 82.97 | 89.43 | 88.21 | 82.23 | 89.30 | 90.12 | 83.29 | 88.94 | 87.64 |
| | KG-RAG | 83.26 | 84.96 | 88.74 | 84.65 | 84.21 | 89.91 | 83.88 | 87.70 | 87.98 |
| | Vector-RAG | 83.35 | 87.31 | 89.35 | 84.15 | 84.75 | 90.17 | 82.92 | 88.49 | 89.29 |
| | HybridRAG | 86.75 | 90.54 | 90.12 | 86.87 | 90.37 | 92.02 | 86.18 | **91.20** | 92.21 |
| | GraphRAG | 86.92 | 90.24 | 91.07 | 87.23 | 90.18 | 92.54 | 85.97 | 90.32 | 91.83 |
| | IHRAG | **87.73** | **91.16** | **91.73** | **87.98** | **90.42** | **92.95** | **86.51** | 90.25 | **92.31** |
| Qwen3-235B | LLM-only | 80.29 | 88.78 | 87.08 | 80.88 | 88.18 | 87.01 | 82.79 | 87.84 | 86.15 |
| | KG-RAG | 82.29 | 84.18 | 87.20 | 83.05 | 85.82 | 89.59 | 82.65 | 86.28 | 86.29 |
| | Vector-RAG | 81.53 | 85.04 | 87.70 | 83.36 | 85.43 | 89.72 | 82.68 | 85.95 | 85.71 |
| | HybridRAG | 84.82 | 89.06 | 89.08 | 85.50 | **90.18** | 91.00 | 84.57 | **89.91** | 89.95 |
| | IHRAG | **85.68** | **89.67** | **90.68** | **86.20** | 89.59 | **92.06** | **85.02** | 89.79 | **90.52** |

注: 最佳结果用加粗标出

IHRAG 方法在所有基础模型和数据集上均展现出显著优势 (表 1). 该方法在私有数据集表现尤为突出: Qwen2.5-1.5B 准确性提升至 74.52%, Qwen-Turbo 相关性达 92.14%, DeepSeek-V3 和 Qwen3-235B 各项指标全面领先. 这表明 IHRAG 通过动态混合机制有效融合了知识图谱的结构化推理与向量检索的语义泛化能力, 显著增强了模型的事实准确性和回答完备性.

RAG 技术对小模型提升效果更为显著. Qwen2.5-1.5B 经 IHRAG 增强后, 准确性较 LLM-only 基线提升近 20 个百分点, 使其性能逼近部分大语言模型的基线水平. 而对大语言模型如 DeepSeek-V3, IHRAG 仍可将其私有数据集相关性提升至 92.95%, 验证了外部知识注入对大语言模型的增益效应.

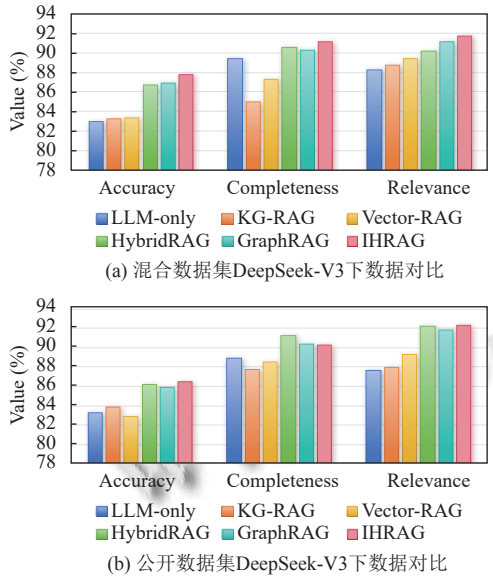
不同 RAG 方法对数据集特性表现出明显差异: KG-

RAG 在私有数据集略优于 Vector-RAG, 反映结构化知识对领域专有概念捕捉的优势; 而 Vector-RAG 在开放域更具适应性. IHRAG 在 DeepSeek-V3 混合数据集上实现了 87.73% 的准确性和 91.73% 的相关性, 比 Hybrid-RAG 提升近 1 个百分点, 证明其动态决策机制优于静态混合策略.

为深入验证 IHRAG 框架中各核心模块的作用, 本研究设计了系统的消融实验方案. 在 DeepSeek-V3 基础模型和私有数据集上, 分别修改关键模块以评估其对系统性能的影响, 实验结果如表 2 所示. 实验结果表明, 动态路由、本体化解构和自适应融合权重这 3 个核心模块对系统性能均有显著影响. 各模块间存在协同增强效应, 完整 IHRAG 框架通过模块有机配合实现了最优的性能表现. 具体结果如图 3 所示.

表 2 消融实验结果对比 (%)

| 方法 | Acc | Com | Rel |
|---|---|---|---|
| IHRAG | 87.56 | 90.35 | 90.63 |
| 去除动态路由 | 86.45 | 89.57 | 89.23 |
| 去除本体化解构 | 85.92 | 88.25 | 88.56 |
| 去除自适应融合权重 | 86.65 | 88.59 | 89.21 |



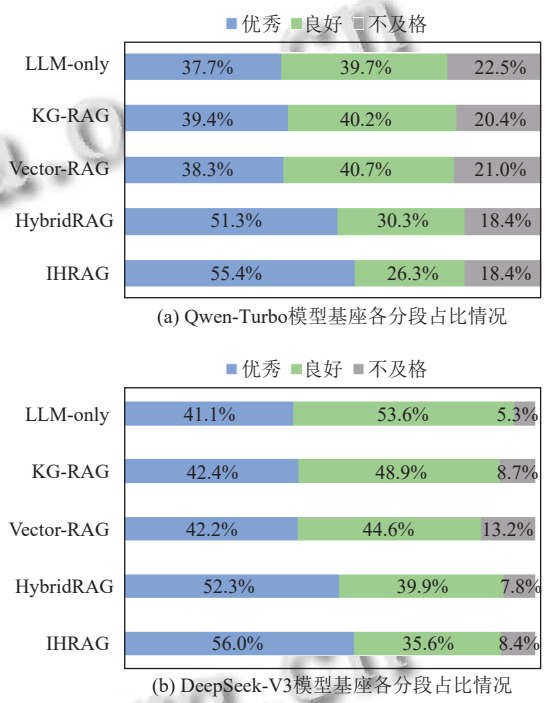

(a) 混合数据集DeepSeek-V3下数据对比

(b) 公开数据集DeepSeek-V3下数据对比

图 3 DeepSeek-V3 基础下各数据集数据对比

在回答质量分级方面 (如图 4, 将得分小于 60 设为不及格, 得分在 60–80 设为良好, 得分大于 80 设为优秀), IHRAG 显著提升高质量回答比例. Qwen-Turbo 和 DeepSeek-V3 的优秀回答占比分别达 55.4% 和 56.0%, 比 LLM-only 基线提升均超 17 个百分点. 同时, 所有 RAG 方法均系统性降低“不及格”回答占比, 其中 IHRAG 在 HybridRAG 基础上实现额外增益, 验证了动态决策机制对知识整合的优化作用.

值得注意的是, DeepSeek-V3 经 IHRAG 增强后达到 56.0% 的优秀率峰值, 表明大语言模型的强推理能力与动态知识检索存在协同效应. IHRAG 成功实现三重突破: 1) 将主流输出从“良好”迁移至“优秀”; 2) 持续抑制低质回答; 3) 在不同规模模型上均提升质量分布.




(a) Qwen-Turbo模型基座各分段占比情况

(b) DeepSeek-V3模型基座各分段占比情况

图 4 各模型回答质量分级对比情况

## 4 结论

本文提出的迭代式混合检索增强生成框架 (IHRAG) 通过系统整合知识图谱的符号推理与向量检索的语义泛化能力, 解决低频实体召回不足、静态知识融合冗余等问题. 实验结果表明, 基于 DeepSeek-V3 的 IHRAG 模型在私有数据集上表现优异 (准确性 87.98%, 完整性 90.42%, 相关性 92.95%), 较最优基线方法提升 1.11–5.75 个百分点. 值得注意的是, 在资源受限场景下 (Qwen2.5-1.5B 为基础模型), 该框架通过迭代优化机制实现 4.01% 的准确性提升, 验证了算法对模型容量的补偿.

在理论创新层面, 通过模式感知问题解构算法建立了临床意图到知识图谱的映射; 动态扩展机制实现检索策略的自适应调整, 减少无效迭代; 异构知识融合

框架优化权重分配, 形成“解构-扩展-融合”的技术闭环, 使系统对复杂意图的处理能力提升. 未来可探索更高效的跨模态知识对齐方法, 降低知识融合的计算开销, 提升领域适应性, 通过真实用户反馈持续改进系统性能.

本研究仍存在一些局限性, 未能与所有先进方法进行全方面对比. 未来工作将致力于开展更广泛的系统性对比测试, 并进一步优化模型的计算效率和领域适应性.

## 参考文献


1 Wang M, Wu WF, Gao CY, *et al*. RoCar: A relationship network-based evaluation method for large language models. arXiv:2307.15997, 2023.

2 Chen LC, Chen JH, Goldstein T, *et al*. INSTRUCTZERO: Efficient instruction optimization for black-box large language models. Proceedings of the 41st International Conference on Machine Learning. Vienna: JMLR.org, 2024. 251.

3 Singhal K, Azizi S, Tu T, *et al*. Large language models encode clinical knowledge. Nature, 2023, 620(7972): 172–180. [doi: 10.1038/s41586-023-06291-2]

4 Kandpal N, Wallace E, Raffel C. Deduplicating training data mitigates privacy risks in language models. Proceedings of the 39th International Conference on Machine Learning. Baltimore: PMLR, 2022. 10697–10707.

5 Shuster K, Poff S, Chen MY, *et al*. Retrieval augmentation reduces hallucination in conversation. Findings of the Association for Computational Linguistics: EMNLP 2021. Punta Cana: Association for Computational Linguistics, 2021. 3784–3803.

6 Kasai J, Sakaguchi K, Takahashi Y, *et al*. REALTIME QA: What’s the answer right now? Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 2130.

7 Wan YW, Chen ZY, Liu Y, *et al*. Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing. Advanced Engineering Informatics, 2025, 65(PB): 103212.

8 Lewis P, Perez E, Piktus A, *et al*. Retrieval-augmented generation for knowledge-intensive NLP tasks. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 793.

9 Wang ST, Yu X, Wang M, *et al*. RichRAG: Crafting rich responses for multi-faceted queries in retrieval-augmented generation. Proceedings of the 31st International Conference on Computational Linguistics. Abu Dhabi: Association for Computational Linguistics, 2025. 11317–11333.

10 Gupta S, Ranjan R, Singh SN. A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions. arXiv:2410.12837, 2024.

11 Izacard G, Grave E. Leveraging passage retrieval with generative models for open domain question answering. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Association for Computational Linguistics, 2021. 874–880.

12 Borgeaud S, Mensch A, Hoffmann J, *et al*. Improving language models by retrieving from trillions of tokens. Proceedings of the 39th International Conference on Machine Learning. Baltimore: PMLR, 2022. 2206–2240.

13 Wang BX, Ping W, Xu P, *et al*. Shall we pretrain autoregressive language models with retrieval? A comprehensive study. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics, 2023. 7763–7786.

14 Arora D, Kini A, Chowdhury SR, *et al*. GAR-meets-RAG paradigm for zero-shot information retrieval. arXiv:2310.20158, 2023.

15 Asai A, Wu ZQ, Wang YZ, *et al*. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. Proceedings of the 12th International Conference on Learning Representations. OpenReview.net, 2024.

16 Fatehkia M, Lucas JK, Chawla S. T-RAG: Lessons from the LLM Trenches. arXiv:2402.07483, 2024.

17 Hu R, Liu S, Qi PP, *et al*. ICCA-RAG: Intelligent customs clearance assistant using retrieval-augmented generation (RAG). IEEE Access, 2025, 13: 39711–39726. [doi: 10.1109/ACCESS.2025.3544408]

18 Xu SC, Pang L, Yu M, *et al*. Unsupervised information refinement training of large language models for retrieval-augmented generation. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Bangkok: Association for Computational Linguistics, 2024. 133–145.

19 Guu K, Lee K, Tung Z, *et al*. REALM: Retrieval-augmented language model pre-training. Proceedings of the 37th International Conference on Machine Learning. JMLR.org, 2020. 368.

20 Edge D, Trinh H, Cheng N, *et al*. From local to global: A graph RAG approach to query-focused summarization.

arXiv:2404.16130, 2024.
21 Li SL, He YC, Guo HY, *et al*. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. Findings of the Association for Computational Linguistics: EMNLP 2024. Miami: Association for Computational Linguistics, 2024. 12758–12786.
22 Wu JD, Zhu JY, Qi YL, *et al*. Medical Graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vienna: Association for Computational Linguistics, 2025. 28443–28467.
23 Cimiano P, Paulheim H. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web, 2017, 8(3): 489–508.
24 Hogan A, Blomqvist E, Cochez M, *et al*. Knowledge graphs. ACM Computing Surveys (CSUR), 2021, 54(4): 71.
25 Li JQ, Wang XD, Wu XB, *et al*. Huatuo-26m, a large-scale Chinese medical QA dataset. arXiv:2305.01526, 2023.
26 Zhang S, Zhang X, Wang H, *et al*. Multi-scale attentive interaction networks for Chinese medical question answer selection. IEEE Access, 2018, 6: 74061–74071. [doi: 10.1109/ACCESS.2018.2883637]

## 附录 A. 部分模块伪代码

算法 A1. 数据预处理模块

```
def document_preprocessing(raw_documents):
# 文档预处理流程
# 输入: 原始文档
# 输出: 结构化文本段落
processed_docs = []
for doc in raw_documents:
# 1. 清理冗余信息和乱码
cleaned_doc = remove_redundant_info(doc)
# 2. 文本分段
segments = text_segmentation(cleaned_ doc)
# 3. 结构化处理
structured_segments = add_structure_metadata(segments)
processed_docs.extend(structured_segments) return processed_docs
def testset_generation(text_segments):
# 测试集生成: 基于文本段落生成问答对
# 输入: 文本段落
# 输出: (问题, 答案) 对列表
qa_pairs = []
for segment in text_segments:
# 使用 LLM 生成相关问题
questions = generate_questions_from_text(segment)
# 生成对应答案
for question in questions:
answer = extract_answer_from_text(question, segment)
qa_pairs.append(question, answer)
return qa_pairs
```

算法 A2. 动态检索路由

```
def dynamic_retrieval_router(query, knowledge_ state):
# 动态路由: 根据问题复杂度选择检索路径
complexity = compute_complexity(query) # 基于实体和关系数量计算复杂度
if complexity < COMPLEXITY_THRESHOLD: # 简单问题
return dense_retrieval(query) # 直接向量检索
else: # 复杂问题
# 混合检索: 向量+知识图谱
sub_questions = ontology_decomposition(query) # 问题解构
kg_docs = knowledge_graph_retrieval(sub_questions)
vector_docs = dense_retrieval(query)
# 自适应融合权重
alpha = compute_adaptive_weight(query, kg_docs, vector_docs)
return alpha * vector_docs + (1 – alpha) * kg_docs
```

算法 A3. 本体化解构

```
def ontology_decomposition(query):
# 基于知识图谱模式的问题解构
entities = extract_entities(query) # 实体识别
sub_questions = []
for entity in entities:
properties = get_entity_properties(entity.type) # 获取实体属性
for prop in properties[:3]: # 每个实体最多 3 个属性
# 生成子问题: 实体+属性组合
sub_q = generate_subquestion(entity, prop)
sub_questions.append(sub_q)
return sub_questions[:5] # 单次生成 5 个子问题
```

算法 A4. 知识缺口检测

```
def knowledge_gap_expansion(knowledge_state):
# 知识缺口检测与查询扩展
gap_concepts = detect_knowledge_gaps(knowledge_state) # 检测未覆盖部分
expansion_queries = []
for concept in gap_concepts:
# 基于医疗本体生成扩展查询
new_queries = generate_expansion_queries (concept, knowledge_state)
expansion_queries.extend(new_queries)
return expansion_queries
```

算法 A5. 自动评估流程

```
def automated_evaluation(question, reference, response):
# 基于 DeepSeek-R1 的自动评估
prompt = construct_eval_prompt(question, reference, response)
```

```
eval_result = llm_evaluate(prompt) # 调用评估模型
return parse_evaluation_scores(eval_result) # 解析相关性/完整性/准确性分数
def batch_evaluation(input_file, output_file):
# 批量评估处理
df = load_data(input_file)
for index, row in df.iterrows():
# 逐条评估并保存进度
scores = automated_evaluation (row['question'], row['answer'], row['response'])
df.at[index, 'scores'] = scores
if index % 5 == 0: # 每 5 条保存一次
save_progress(df, output_file)
```

```
# 系统阈值参数
COMPLEXITY_THRESHOLD = 5 # 复杂度阈值
SUFFICIENCY_THRESHOLD = 0.7 # 答案充分性阈值
INFORMATION_GAP_THRESHOLD = 0.2 # 信息缺口阈值
# 检索参数
VECTOR_TOP_K = 5 # 向量检索返回数量
KG_RELATION_DEPTH = 3 # 知识图谱关系深度
# 评估参数
HUMAN_REVIEWERS = 2 # 人工评审员数量
CONSISTENCY_THRESHOLD = 0.7 # 评审一致性阈值
```

(校对责编: 张重毅)