

基于知识图谱的迭代改写检索增强生成方法

毛 赞^{1,2}, 符海东^{1,2}, 顾进广^{1,2}, 张 娜³, 高 峰^{1,2†}

1. 武汉科技大学 计算机科学与技术学院, 湖北 武汉 430065;
2. 智能信息处理与实时工业系统湖北省重点实验室, 湖北 武汉 430065;
3. 中国五矿集团有限公司 数字化管理部, 北京 100010

收稿日期: 2025-06-27 † 通信联系人 E-mail: feng.gao86@wust.edu.cn

基金项目: 湖北省教育厅科学技术研究计划重点项目(D20231104); 武汉市科创局对口科技支援项目(2024071104010831)

第一作者: 毛 赞, 男, 硕士生, 现从事知识图谱方面研究。E-mail: 904375057@qq.com

摘 要: 检索增强生成(Retrieval-Augmented Generation, RAG)方法自提出以来便引发广泛关注, 其核心目标在于通过引入外部信息来提升大语言模型的推理能力, 从而有效缓解幻觉问题。然而, 检索增强生成方法在复杂多跳问答中面临单次检索信息不全、迭代过程冗余干扰导致错误传播的挑战。为此, 提出一种基于知识图谱的迭代改写 RAG 方法(Knowledge Graph-based Iterative Rewriting RAG, KGIR-RAG), 通过证据子图构建筛选检索段落中的结构化知识, 利用最大生成树算法剔除噪声三元组; 迭代查询重构, 用图谱事实替换问题中的多跳依赖, 逐步降低推理复杂度; 最后通过反思推理模块判断答案完备性, 驱动迭代终止。在 HotpotQA 与 MuSiQue 数据集上进行了实验, 验证了各模块的有效性。实验结果表明: KGIR-RAG 方法在生成性能和检索性能上优于近年提出的一些方法。

关 键 词: 大语言模型; 检索增强生成; 知识图谱; 迭代改写

中图分类号: TP391.1

文献标志码: A

文章编号: 1671-8836(2026)04-0543-12

Knowledge Graph-Based Iterative Rewriting for Retrieval-Augmented Generation Method

MAO Yun^{1,2}, FU Haidong^{1,2}, GU Jinguang^{1,2}, ZHANG Na³, GAO Feng^{1,2†}

1. School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, Hubei, China;
2. Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan 430065, Hubei, China;
3. Digital Management Department, China Minmetals Corporation, Beijing 100010, China

Abstract: Retrieval-Augmented Generation (RAG) methods have attracted widespread attention since their proposal, with the core goal of enhancing large language models' (LLMs) reasoning capabilities by incorporating external information to effectively mitigate hallucination. However, RAG approaches face challenges in complex multi-hop question answering, including incomplete information retrieval in a single pass and error propagation caused by redundant interference during iteration. To address these issues, we propose a Knowledge Graph-based Iterative Rewriting RAG method (KGIR-RAG), which constructs evidence subgraphs to filter structured knowledge from retrieved passages and utilizes a maximum spanning tree algorithm to eliminate noisy triplets. It iteratively rewrites queries by replacing multi-hop dependencies in questions with factual relations from the knowledge graph, progressively reducing reasoning complexity. Finally, a reflection-based reasoning module evaluates answer completeness to drive iteration termination. We conducted experiments on the HotpotQA and MuSiQue datasets to validate the effectiveness of each module. The results demonstrate that KGIR-RAG outperforms several recently proposed methods in both generation and retrieval performance.

Key words: large language model; retrieval augmented generation; knowledge graph; iterative rewriting

引用格式: 毛赞, 符海东, 顾进广, 等. 基于知识图谱的迭代改写检索增强生成方法[J]. 武汉大学学报(理学版), 2026, 72(4): 543-554. DOI: 10.14188/j.1671-8836.2025.0144.

MAO Yun, FU Haidong, GU Jinguang, et al. Knowledge Graph-Based Iterative Rewriting for Retrieval-Augmented Generation Method [J]. *J. Wuhan Univ. (Nat. Sci. Ed.)*, 2026, 72(4): 543-554. DOI: 10.14188/j.1671-8836.2025.0144(Ch).

0 引言

大语言模型(Large Language Model, LLM)在多层对话任务中表现出了卓越的综合能力。这种能力不仅体现在对用户问题的精准解析与自然流畅的回复生成,更体现在对上下文的深度理解与逻辑推理的有效处理。随着 LLM 技术的快速发展,其在医疗^[1]、法律^[2]、电子商务^[3]等领域的应用已取得显著成效,展现出强大的实用价值。然而,这一技术突破也伴随着新的挑战,尽管 LLM 基于大规模语料训练,但在生成响应中频繁出现的“幻觉”现象已成为亟待解决的关键问题,该现象主要源于模型包含过时信息或缺乏特定领域知识。为应对这一挑战,检索增强生成(Retrieval-Augmented Generation, RAG)技术应运而生,通过从结构化文档中检索相关知识并融入 LLM 的提示中,为缓解幻觉问题提供了创新性的技术路径。

现有的 RAG 研究^[4]普遍采用语义匹配方法^[5]进行文档检索,通过计算用户查询与文档内容的相似性,筛选出最相关的文本片段。然而,这种检索机制存在明显局限性:检索结果往往呈现高度同质化特征,并存在语义重复现象,难以有效揭示文本片段间的语义关联^[6]。针对这一问题,研究者提出了迭代检索的方法以提升检索全面性和准确性。例如,ITER-RETGEN^[7]和 IRCOT^[8]方法通过 LLM 的响应进行多次迭代检索,其核心假设是 LLM 的每次响应中都包含缺失信息的描述,再据此检索出更多相关段落。但当检索的信息包含过多冗余信息时,LLM 在生成时会受到冗余信息的干扰导致无法精确生成缺失的信息,从而导致无效迭代。Adaptive-RAG^[9]方法则构建分类器模型,根据问题的难易程度动态选择执行单次检索或多次检索。RQ-RAG^[10]方法通过问题重写、分解和消歧策略,在回答过程中实现检索过程的优化。这两种方法都需要对模型进行微调,增加了资源的消耗。

为进一步提升 RAG 系统回答的准确性,研究者们提出了将知识图谱和 RAG 系统结合的方法^[11],旨在优化检索与生成性能。相较于传统的检索方式,知识图谱中的三元组通过文档段落抽取构建而成,能够精确表示实体间关系。基于知识图谱的方法具备全局信息感知优势,通过实体的深度提炼与总结,可高效应对概括性问题和多跳推理问题,显著扩展了 RAG 系统的应用场景。在子问题分解的方法中,当子问题的检索段落错误或包含过多干扰信息时会导致回答错误,从而造成错误传播风险。在迭代求解过程中,当中间段落存在干扰信息时,

利用生成的内容进行下一步检索会使迭代过程偏离原始问题,从而造成无效迭代。因此,在迭代过程中提高检索信息的质量至关重要。

针对上述方法存在的缺陷,本文提出了基于知识图谱的迭代改写检索增强生成方法(Knowledge Graph-based Iterative Rewriting RAG, KGIR-RAG),通过引入知识图谱改进迭代求解过程,使迭代中使用的信息更加精准以缓解错误传播和无效迭代问题。

本文的贡献主要有以下两点:

1) 在检索阶段针对迭代求解方法中因检索信息质量不佳导致无效迭代的问题,本文引入知识图谱并进行子图探寻,以改进检索信息的质量使迭代更高效,从而提升端到端问答性能。

2) 在生成阶段提出一种综合考虑证据子图、查询重构过程和检索段落的 CoT (Chains-of-Thoughts) 方法,该方法能综合考虑迭代过程中的信息来回答问题或进行下一步迭代。该方法即插即用,在小参数量模型上无需进行微调即可达到与大参数量模型相近的性能。

1 相关研究

1.1 传统检索增强生成方法

传统 RAG 方法包含索引、检索和生成三个阶段。在索引阶段,对各种格式的原始数据进行清理并转换为统一的纯文本格式。为了适应语言模型的上下文限制,文本被分割成更小的块。这些块随后使用嵌入模型编码成向量,并存储在向量数据库中,为高效相似性检索奠定基础。当接收到用户查询时,RAG 系统使用相同嵌入模型将查询编码为向量,并计算其与库中块向量的相似性,返回最相关的 K 个文本块作为扩展上下文。在生成阶段,问题、检索结果(及可选的对话历史)被整合为结构化提示,由大语言模型据此生成答案,在这一过程中可结合自身知识,也可限定仅基于检索内容,从而支持准确、连贯的单轮或多轮问答。

例如,In-Context RALM^[12]方法通过保持 LLM 参数不变,在生成阶段将检索到的文档段落直接插入原始提示前部。这种结构化上下文扩展策略显著提升了模型对复杂语义的理解能力,同时提高了生成过程的计算效率。TOC^[13]针对复杂问题提出树状分解架构,该方法通过递归式知识检索构建问题树,将复杂问题分解为多个子问题。通过逐层求解并聚合子答案,最终生成结构化长文本输出。其优势在于将模糊问题的处理转化为可管理的子任务集合。

相较于无需训练的模型,基于检索知识的训练方法展现出更显著的性能优势。以RETRO^[14]为例,该方法采用独立预训练的BERT模型作为检索器,通过编码-解码架构实现知识整合。这种设计使模型能够将检索到的片段有效融入预测过程,形成知识驱动的生成机制。而RALM^[15]则采取差异化策略,利用ColBERTv2^[16]构建高效预训练检索器,并通过微调大型语言模型实现知识迁移。这种策略使模型能够更精准地利用检索到的文章段落,显著提升了生成内容的准确性和上下文连贯性。

传统RAG方法在检索阶段面临精确率与召回率的双重挑战:一方面,可能检索到与查询语义不匹配或相关性较低的文本块,导致噪声信息的引入;另一方面,关键信息可能被遗漏,影响上下文的完整性。在生成阶段,模型仍可能出现“幻觉”问题^[17],即生成的内容超出或偏离检索到的上下文支持,编造虚假或未经证实的信息。这些问题共同制约了RAG系统在实际应用中的表现与可信度。

1.2 改进型检索增强方法

如图1所示,通过引入特定的改进措施,改进型RAG方法旨在克服传统RAG的局限性。它侧重于提升检索质量,并采用了预检索和后检索策略。为了解决索引问题,改进型RAG方法在查询预处理阶段通过查询重写、查询分解和查询扩展等手段对索引技术进行了优化。在查询后处理阶段通过引入重排序、摘要和图谱增强等方法优化检索结果。

如文献[7]通过迭代实现检索与生成。在每次迭代中,模型的输出会包含完成任务可能需要的额外信息,从而可以利用输出内容检索相关信息,这

些相关信息反过来又在下一次迭代中帮助生成更好的响应。文献[8]提出思维链(CoT)生成与知识检索的交替执行框架。该方法通过迭代式信息补充机制,既保证了初步答案的生成效率,又通过持续检索增强推理深度,最终实现高质量输出。其核心优势在于将知识获取与逻辑推理过程深度融合。HyDE^[18]根据用户输入问题生成假设文档,这些文档来自LLM自身学习到的知识;随后,系统以假设文档替代原始查询进行检索。该方法通过生成语义更丰富、表达更完整的假设文档,有效对齐了查询与候选文档的语义空间,尤其在原始查询较短或表述模糊时,显著提升了检索的精确率与召回率。

现有改进型RAG方法技术特点如表1所示。在基于知识图谱的检索增强方法中,HippoRAG2^[19]在检索阶段使用嵌入模型对段落和三元组进行评分,以确定种子节点类型,并使用个性化PageRank算法选择相关节点,通过更深入的段落整合和更有效的在线检索增强其性能^[19]。这种结合使RAG系统更接近人类长期记忆的有效性,同时也展现出更优越的事实性知识和意义构建记忆能力。KG²RAG^[6]方法利用知识图谱提供片段间的事实级关系,从而提高检索结果的多样性和连贯性。在执行基于语义的检索以提供种子片段之后,KG²RAG采用一个基于知识图谱的片段扩展过程和一个基于知识图谱的片段组织过程,将相关且重要的知识组织为段落形式进行问答,避免了信息碎片化的问题。KAG^[20]充分利用知识图谱中结构严谨的优势,通过多重表征和互索引的方式优化信息组织,并借助知识和逻辑语义引导,帮助实现结构化的思考和推理。在问题求解阶段,KAG将问题拆分为多个子问题,每个子问题采用检索、排序、计算等相关算子进行求解,有效提高了模型对复杂逻辑问题的求解能力。GraphRAG^[21]构建了多级社区图谱索引,通过定制化策略分别应对全局与局部检索需求,显著提升抽象概括性问题与具体明确问题的解答能力。LightRAG^[22]在GraphRAG基础上引入双层检索架构,通过低层检索聚焦实体关系细节信息,高层检索覆盖广泛主题范围,有效优化了图谱构建成本与检索效率。

上述方法代表了RAG技术从“静态检索+单步生成”向“动态交互”与“结构化推理”演进的重要方向。通过引入迭代机制或融合知识图谱,RAG系统在语义匹配、信息完整性、逻辑推理和答案组织等方面均获得显著提升。然而,以上方法仍然存在依赖高质量三元组抽取、知识图谱构建与更新成本高等问题。

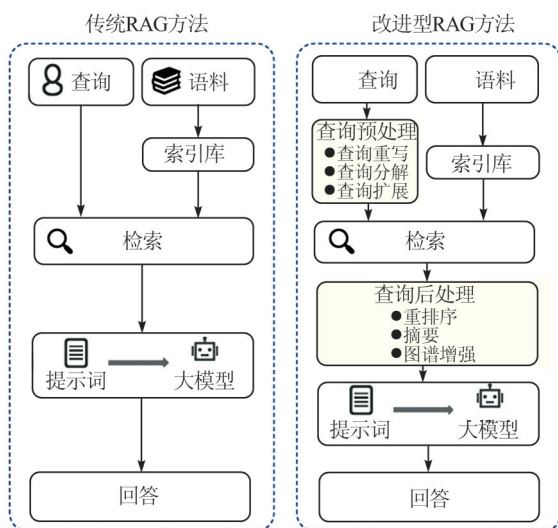


图1 传统RAG方法和改进型RAG方法对比

Fig. 1 Comparison of traditional and enhanced RAG methods

表 1 现有改进型 RAG 方法技术特点

Table 1 Technical features of enhanced RAG methods

方法	检索数据类型	检索粒度	检索过程
IRCoT ^[8]	文本文档	文档段落	CoT 的单个推理步骤进行迭代检索
ITER-RETGEN ^[7]	文本文档	文档段落	LLM 完整响应内容进行迭代检索
HyDE ^[18]	文本文档	生成的假设性文本	使用假设性文本单次检索
HippoRAG2 ^[19]	知识图谱	三元组、上下文	个性化 PageRank 方法进行单次检索
KG ² RAG ^[6]	知识图谱	三元组	单次三元组检索并扩展
KAG ^[20]	知识图谱	三元组	根据分解后的子问题依次检索
GraphRAG ^[21]	知识图谱	三元组、社区	单次三元组检索或相关社区检索
LightRAG ^[22]	知识图谱	实体	基于抽象实体或具体实体单次检索

2 本文方法

本文提出的 KGIR-RAG 方法整体架构如图 2 所示。该方法由三个核心模块组成:证据子图构建(简称子图构建)模块,迭代查询重构(简称查询重构)模块和反思推理模块。证据子图构建模块首先基于问题和段落的语义相似性检索相关段落,随后将段落中的三元组构建为证据子图。由于证据子图中包含冗余信息,所以本文通过提取问题相关实体的连通

图,再通过三元组的相关程度作为边的权重得到一个最大生成树作为最终的证据子图。迭代查询重构模块将原始问题中的多跳部分用证据子图中的信息进行改写,从而减少问题的跳数,使问题易于回答。反思推理模块综合考虑证据子图、查询重构过程和检索段落判断这些信息是否支持回答问题,当信息不足以回答问题时则进行下一轮查询重构。直到多轮子图构建后的信息足以回答问题,或达到最大迭代次数时则退出迭代。

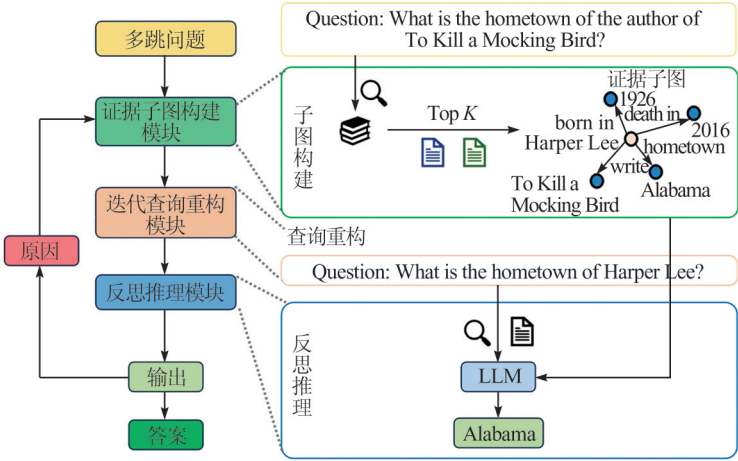


图 2 KGIR-RAG 总体架构

Fig. 2 Overall architecture of KGIR-RAG

2.1 文档预处理过程

在现有 RAG 方法中^[4],通常采用固定长度分割策略将文档拆分为 n 个片段,随后通过文本嵌入模型将这些片段转换为向量表示。这种标准化处理方式可能导致检索信息碎片化,难以捕捉片段间的语义关联。为此,本文从原始文本段落中抽取实体与关系,构建知识图谱,文档预处理流程如图 3 所示。

针对给定文档,可将其分块后的片段表示为 $D = \{d_1, d_2, \dots, d_n\}$,为了捕捉这些片段间丰富的事

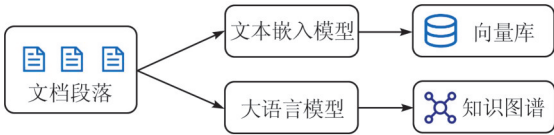


图 3 文档预处理流程

Fig. 3 Document preprocessing flow

实关系,可通过以下两种方式建立知识图谱关联:第一种方法是在已有外部知识图谱中,通过实体识别与关系抽取技术,将片段中的实体与图谱实体进行映射关联;第二种方法是直接从片段中提取实体和关系,构建成段落子图并整合为完整知

识图谱。本文选择第二种方法作为核心实现路径,其优势在于避免对外部知识图谱的依赖。通过合理设计提示策略,借助大型语言模型实现该过程,最终形成片段与知识图谱的结构化连接,其数学表达式为:

$$\mathcal{G} = \{(h, r, t, c) | c \in D\} \quad (1)$$

其中, h, r, t 分别代表头实体、关系和尾实体, c 表示三元组对应的文档段落。值得注意的是,文档片段与知识图谱的关联过程具有查询无关性,这一特性使该过程可离线执行。只需对所有文档进行一次处理,即可完成知识图谱构建。这种离线处理机制与传统 RAG 方法在功能上保持一致,既支持对现有知识库和知识图谱进行文档的动态增删操作,又具备良好的扩展性,当新增文档时系统可实现增量更新,无需重复构建整个知识图谱。

2.2 证据子图构建模块

本文引入证据子图构建方法将段落中的实体构建为一个证据子图,这一过程不仅避免了检索的段落之间过于碎片化的问题,还能有效去除段落中的冗余信息,使迭代过程中所参考的信息更加准确。具体构建过程如下。

首先对于输入问题 q 与所有片段之间的语义相似性可以表示为:

$$S = \{s(q, c) | c \in D\} \quad (2)$$

其中,相似性函数 s 使用余弦相似性方法,根据问题向量和段落向量计算两者的相似性。与问题 q 相似性最高的 k 个片段记作 D_q 。在检索过程中,如果仅依赖基于语义的检索可能会导致文档片段间缺乏语义关联,造成知识碎片化或关键信息丢失和上下文关联断裂。为解决这一问题,本文将检索到的片段 D_q 作为初始片段,将 D_q 中的知识表示为:

$$\mathcal{G}_q = \{(h, r, t, c) | c \in D_q\} \in \mathcal{G} \quad (3)$$

之后,再利用 LLM 对输入的问题 q 进行命名实体识别(Named Entity Recognition, NER),抽取出问题 q 中的实体 E_q :

$$E_q = \{e_1, e_2, \dots, e_n\} = \text{NER}(q) \quad (4)$$

得到检索片段相关图谱 $\mathcal{G}_q$ 和问题中关键实体 E_q 后,需要从 $\mathcal{G}_q$ 中探寻一个包含关键实体 E_q 的证据子图 $\mathcal{G}_{\text{Ans}}$,该子图将在后续迭代推理过程中共享。

具体而言,依次把 E_q 中实体作为种子实体,将该实体及其相邻的边添加到 $\mathcal{G}_{\text{Ans}}$ 中。由于段落中的三元组和问题中的实体都是由 LLM 抽取出来,同一个实体在不同语境下可能会稍有不同,因此本文采用 N -gram 算法判断问题实体和段落中的三元组是否为同一实体,得到筛选后的证据

子图 $\mathcal{G}_{\text{tmp}}$:

$$\mathcal{G}_{\text{tmp}} = N\text{-gram}(e, \mathcal{G}_q) \quad (5)$$

然而,由于引入了该实体的所有相邻边,不可避免地会在 $\mathcal{G}_{\text{tmp}}$ 引入噪声知识,这种噪声信息可能显著降低事实推理的准确性,因此证据子图的优化成为提升推理准确性的关键环节。为此,本文进一步筛选出证据子图中的连通图 $\mathcal{B}_i$:

$$\mathcal{B}_i = \text{isConnected}(\mathcal{G}_{\text{tmp}}) \quad (6)$$

为节省计算资源可以用(2)式中计算的向量相似性作为三元组中边的权重,得到一个加权无向图,采用最大生成树(Maximum Spanning Tree, MST)算法根据此权重为 E_q 中每一个实体构造最大生成树,再将三元组信息合并为证据子图:

$$\mathcal{G}_{\text{Ans}} = \{\text{MST}(e) | e \in E_q\} \quad (7)$$

最终利用筛选之后的证据子图 $\mathcal{G}_{\text{Ans}}$ 进行下一步的查询重构。

证据子图构建模块的算法如算法 1 所示。算法以问题 q , 文档段落 C , 文档知识图谱 $\mathcal{G}$ 作为输入,输出证据子图 $\mathcal{G}_{\text{Ans}}$ 。首先通过相似性检索函数 S 检索最相关段落 D_q ,再将段落中的实体相连接构建为段

算法 1 证据子图构建算法

输入: 问题 q ; 文档段落 C ; 文档知识图谱 $\mathcal{G}$;

输出: 与问题 q 相关的证据子图 $\mathcal{G}_{\text{Ans}}$;

```

1  def KnowledgeExploration( $q, C, \mathcal{G}_{\text{Ans}}$ ):
2       $D_q = S(q, C)$  //通过相似性检索函数  $S$  检索最
        相关段落  $D_q$ 
3       $E_q = \text{NER}(q)$  //通过 NER 方法抽取问题  $q$  中的
        实体
4       $\mathcal{G}_q = \{(h, r, t, c) | c \in D_q\} \in \mathcal{G}$  //将段落中的三
        元组构建为段落知识图谱  $\mathcal{G}_{\text{tmp}}$ 
5      For  $e$  in  $E_q$ :
6           $\mathcal{G}_e = N\text{-gram}(e, \mathcal{G}_q)$  //采用  $N$ -gram 算法将段
            落三元组与实体进行匹配
7           $\mathcal{G}_{\text{tmp}} = \mathcal{G}_{\text{tmp}} \cup \mathcal{G}_e$  //将关键实体相邻的边添加
            到段落知识图谱  $\mathcal{G}_{\text{tmp}}$  中
8           $\mathcal{B}_i = \text{isConnected}(\mathcal{G}_{\text{tmp}})$  //选择扩展之后的连
            通图  $\mathcal{B}_i$ 
9           $\mathcal{G}_{\text{score}} = \text{Score}(q, \mathcal{B}_i)$  //利用重排序模型计算连通
            图  $\mathcal{B}_i$  中各条边与问题  $q$  的相似性,并将相似性作
            为边的权重
10          $\mathcal{G}_{\text{Ans}} = \text{MST}(\mathcal{G}_{\text{score}})$  //用最大生成树算法得到问
            题相关子图  $\mathcal{G}_{\text{Ans}}$ 
11     return  $\mathcal{G}_{\text{Ans}}$ 

```

落知识图谱 $\mathcal{G}_{\text{imp}}$, 并选取其中的连通图 $\mathcal{B}_i$, 最后利用 bge-ranker-large 重排序模型计算连通图 $\mathcal{B}_i$ 中各条边与问题 q 的相似性, 并将相似性作为边的权重, 用最大生成树算法得到问题相关证据子图 $\mathcal{G}_{\text{Ans}}$ 。

2.3 迭代查询重构模块

迭代查询重构模块利用证据子图中的信息对原始问题进行改写, 使其更易于回答。证据子图构建模块根据相关段落结构进行筛选得到证据子图 $\mathcal{G}_{\text{Ans}}$, $\mathcal{G}_{\text{Ans}}$ 包含回答问题所需的部分信息, 但不一定包含所有信息。因此迭代查询重构模块的作用是利用这一部分相关信息将问题进行改写, 用知识图谱中的三元组信息替换问题中的多跳部分, 并通过查询重构的方式逐步降低多跳问题的复杂程度, 使问题变得简单清楚更有利于回答。其过程如下:

$$q' = \text{Rewrite}(q, q_{\text{history}}, \mathcal{G}_{\text{Ans}}) \quad (8)$$

其中, q 代表原始问题, q_{history} 代表改写的历史过程, 查询重构算法如算法 2 所示。

查询重构的提示词如图 4 所示, 在提示词中说明在不改变问题原意的情况下用三元组替换句子中的信息。给出的示例问题为: “2013 年 Liqui Moly 巴瑟斯特 12 小时耐力赛的赛道长度是多少?”。改写后用具体的场馆名替代了问题中的定语从句, 最终输出为: “Mount Panorama 赛道的长度是多少?”。此改写方式将一个复杂多跳问题逐步简化, 有效缓解了单次检索方法因检索信息不全导致回答错误的问题。

2.4 反思推理模块

反思推理模块作为整个过程中的一个关键步骤, 主要是综合利用证据子图和段落信息回答改写

算法 2 查询重构算法

输入: 问题 q ; 查询重构历史 q_{history} ; 证据子图 $\mathcal{G}_{\text{Ans}}$;

输出: 改写后的问题 q' ;

```

1  def IterRewrite( $q, q_{\text{history}}, \mathcal{G}_{\text{Ans}}$ ):
2       $q' = q$ 
3      triples = Select( $q, \mathcal{G}_{\text{Ans}}$ ) // 用 LLM 选择证据子
        图  $\mathcal{G}_{\text{Ans}}$  和问题  $q$  最相关的三元组
4      If triples != None // 当相关三元组为空时返回
        原始问题
5           $q' = \text{Rewrite}(q, \text{triples})$  // 用最相关三元组
            triples 替换问题中的信息以简化问题
6      return  $q'$ 

```

后的问题。由于在证据子图构建模块已经检索了相关段落构造证据子图, 因此反思推理模块重点通过上一阶段未检索出的段落进行反思推理。在反思推理过程中模型需综合证据子图、查询重构历史和新增段落信息并结合 CoT 方法判断已有信息是否支持回答问题, 如果支持则直接给出答案, 否则给出缺失的信息并进行下一次迭代改写, 当达到最大迭代次数时直接生成最终答案。

反思推理的提示词示例如图 5 所示。在示例 1 中, 针对问题“《杀死一只知更鸟》的作者的家在哪里?”, 已有三元组信息足以回答该问题, 因此模型直接给出答案: “亚拉巴马州”。而在示例 2 中, 证据子图无法回答“《杀死一只知更鸟》的作者的家在哪里?”, 因为虽然能从中推理出哈珀·李的家是亚拉巴马州, 但上下文中并未提供该州有哪些机场的信息。因此, 模型输出了无法回答的原因, 后续可基于这一原因进行下一步迭代查询。

Rewrite the question based on the given triples, using entity from the triples to make the question simpler and easier to answer. Replace the information in the sentence with triples without changing the original meaning of the question. Following the format in the example, output the rewritten question beginning with "Rewrite:" and ending with a question mark."

Example1:

Question: What is the length of the track where the 2013 Liqui Moly Bathurst 12 Hour was staged?

Triples: ["2013 Liqui Moly Bathurst 12 Hour", "staged at ", "the Mount Panorama Circuit"]

Rewrite_history: []

Rewrite: What is the length of the track in Mount Panorama Circuit?

Question: {query_str}

Triples: {triples_str}

Rewrite_history: {rewrite_history_str}

Rewrite:

图 4 查询重构过程的提示词

Fig. 4 Prompt for a question rewriting

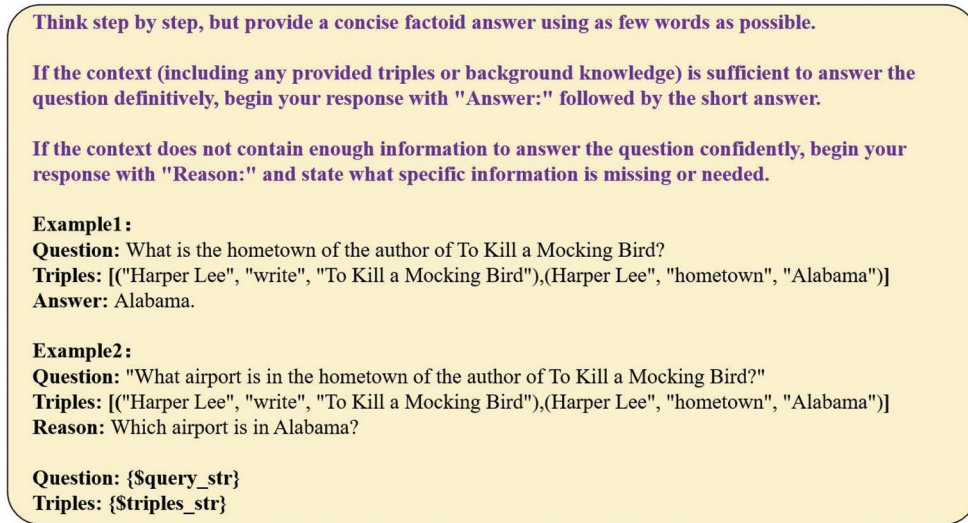


图5 反思推理过程的提示词

Fig. 5 Prompt for reflective reasoning process

3 实验

3.1 数据集

本文在两个广泛使用的多跳数据集 MuSiQue^[23]和 HotpotQA^[24]上进行了实验。为确保实验的公平性,本文采用与 HippoRAG 相同的数据筛选方法,从每个数据集的验证集上随机选取 1 000 个样本进行评估,以全面验证 KGIR-RAG 方法的泛化能力。实验所采用样本的跳数分布如表 2 所示。

MuSiQue^[23]数据集在 2021 年提出,这是一个包含 25 000 个 2 跳到 4 跳的问答数据集。每个问题关联有 20 个上下文段落,这些段落混合了相关和无关的信息,进一步增加了识别正确推理路径的难度。该数据集引入了一种自底向上的方法,通过系统性筛选具有逻辑关联的单跳问题对,形成多跳推理链条。这种设计不仅提升了数据集的挑战性,也提供了更严格的测试场景。

HotpotQA^[24]是在 2018 年提出的一个著名问答基准数据集。该数据集包含 113 000 个基于维基百科的问答对,主要由 2 跳问题组成,每个问题关联有 10 段维基百科文章。在这 10 段文章中,有些包含回答问题所需的支持事实,而其余的则作为干扰信息存在。这

表2 两个数据集的跳数

Table 2 Hop count for two datasets

数据集	跳数	数量
MuSiQue ^[24]	2 跳	518
	3 跳	316
	4 跳	166
HotpotQA ^[23]	2 跳	1 000

种设计有效模拟了真实场景中信息过载的挑战,为评估模型推理的鲁棒性提供了重要依据。

3.2 实验环境及参数设置

使用硬件平台为 Intel(R) Xeon Platinum 8358P CPU, 8 张 NVIDIA A800-SXM4-80 GB 显卡,操作系统为 Ubuntu22.04, Python 3.10.14 编程环境,基于 PyTorch 框架进行实验。

在图谱构建阶段,为保证图谱质量,本文采用 Qwen2.5:72B 模型对候选段落进行三元组抽取,并对原始问题进行实体抽取。在各数据集中抽取的实体、关系和三元组数量如表 3 所示。

表3 各数据集抽取的实体、关系和三元组数量

Table 3 Number of extracted entities, relations, and triples per dataset

数据集	三元组	实体	关系
MuSiQue	13 427	103 026	43 409
HotpotQA	6 514	50 514	21 705

为确保实验的公平,在生成性能实验中所有基线均采用 BGE-M3 作为文本嵌入模型,其最大输入序列长度为 8 192,输出向量的维度为 1 024,输出向量通过余弦相似性计算相似性,根据相似性逆序召回 Top 10 个段落进行后续生成。在生成阶段使用 LLaMA3:8B 和 Qwen2.5:72B 不同参数量的模型验证本文提出的方法,所有模型均使用 vllm 框架进行部署。其他更多实验细节可参考本实验代码, <https://github.com/morris-yun/KGIR-RAG.git>。

3.3 实验结果及分析

3.3.1 基线方法

本文与以下几类方法进行对比,各类方法介绍

如下。

1) 原生推理方法

Naive Gen:在不进行检索的情况下,采用 LLM 直接生成答案。

Naive RAG^[16]:原生 RAG 方法将检索的段落拼接后输入 LLM 生成最终答案。

2) 基于迭代的方法

IRCoT^[8]:通过交替进行思维链生成和知识检索步骤,从而利用 CoT 指导检索,并反过来用检索结果辅助 CoT。这种交替过程使得后续的推理步骤能够获取到更相关的信息,相比仅使用问题作为查询的标准检索方式更为有效。

ITER-RETGEN^[7]:通过迭代实现检索与生成。在每次迭代中,模型的输出会包含完成任务可能需要哪些额外信息,从而利用输出内容检索相关信息,这些相关信息反过来又在下一次迭代中帮助生成更好的响应。

3) 基于问题改写的方法

HyDE^[18]:根据用户输入问题生成假设文档,这些文档来自 LLM 自身学习到的知识,再依据假设文档检索相关信息。HyDE 通过生成假设文档对齐查询和文档的语义空间,在原始查询较短或措辞不准确时有助于改善查询和检索文档之间的语义匹配,提高检索的准确性和召回率。

4) 基于知识图谱的方法

KG²RAG^[6]:在语义检索获取种子片段后,通过知识图谱进行片段的扩展与组织,将相关重要知识整合为段落形式用于问答,有效避免信息碎片化,最终提升检索结果的多样性和连贯性。

HippoRAG2^[19]:对 HippoRAG^[4]方法改进,通过嵌入模型对段落和三元组评分以确定种子节点,并采用个性化 PageRank 算法选取相关节点,结合更深入的段落整合与高效的在线检索策略,增强大语言模型的性能。该方法使系统更接近人类长期记忆机制,在事实性知识和意义构建方面表现更优。

KAG^[20]:充分利用知识图谱中结构严谨的优势,通过多重表征和互索引的方式优化信息组织,并借助知识和逻辑的语义引导帮助实现结构化的思考和推理。在问题求解阶段,KAG 将问题拆分为多个子问题每个子问题采用检索、排序、计算等相关算子进行求解,有效提高了模型对复杂逻辑问题的求解能力。

GraphRAG^[21]:通过构建多级社区结构的知识图谱索引,针对全局性与局部性检索需求设计了差异化策略,有效提升了系统在回答抽象概括类问题

和具体事实类问题上的能力。

LightRAG^[22]:在 GraphRAG 基础上提出了一种双层检索架构,低层检索专注于实体间的关系细节,高层检索则覆盖更广泛的语义主题,从而在降低图谱构建成本的同时显著提升了检索效率。

5) 检索模型

BM25^[25]:核心思想是基于词频、逆向文档频率和文档长度归一化等因素,综合计算文档与查询之间的相关性。作为一种经典且高效的稀疏检索方法,BM25 目前已被广泛应用于各类搜索引擎中,并常作为基线用于评估新型检索模型的性能。

Contriever^[26]:将对比学习方法用于无监督密集型检索模型的训练,通过在大规模无标注语料上学习语义相似性,使模型能够捕捉查询与文档间的深层语义关联。这种方法不仅在各种检索场景中取得优异表现,还展现出强大的跨语言迁移能力,尤其适用于低资源语言或跨语言检索任务。

RAPTOR^[27]:通过递归地对文本片段进行嵌入、聚类 and 摘要,自底向上构建一个多层级的树形概括结构。在推理阶段,该方法可从该树中检索信息,从而在不同抽象层次上整合长文档内容。与传统检索模型相比,RAPTOR 在多项任务中均表现出更优的检索性能,尤其在涉及复杂多步推理的任务中优势更为显著。

3.3.2 问答性能实验

本文使用 LLama3:8B 和 Qwen2. 5:72B 这两类不同参数量的模型,通过对比各模型在 HotpotQA^[23]和 MuSiQue^[24]数据集上的 EM(准确率)和 F1,验证端到端生成性能。具体的实验结果见表 4 和表 5。

表 4 的实验结果表明,在两个数据集上,本文提出的 KGIR-RAG 方法大部分指标优于其他基线方法。

表 5 的实验结果表明,在更小参数量的 LLaMA3:8B 模型上,KGIR-RAG 方法依然优于各基线方法。这说明本文提出的方法在不同尺寸的模型上均有效。

与 Qwen2. 5:72B 模型相比,LLaMA3:8B 模型在各数据集上 EM 和 F1 的平均值分别下降了 2.2 和 2.4 个百分点,这说明参数量大的模型具有更强的语义理解能力,能从段落中理解一些细节信息回答多跳问题。

本文将实验性能的提升归因于以下三个方面:

1) 将检索的段落重新组织成高质量的证据子图为后续检索提供了有效的信息支撑。本文方法将段落中的实体与问题中的关键实体进行关联,有效克服了传统检索方法中段落文本存在干扰或冗余信

表4 各方法端到端问答性能对比(Qwen2.5:72B)

Table 4 Comparison of end-to-end question answering performance across methods (Qwen2.5:72B)

Method	MuSiQue		HotpotQA		Average	
	EM	F1	EM	F1	EM	F1
Naive Gen	3.3	7.4	22.3	31.3	12.8	19.4
NativeRAG ^[16]	24.8	35.7	46.6	60.4	35.7	48.1
IRCoT ^[8]	30.8	36.5	59.1	60.7	44.9	48.6
ITER-RETGEN ^[7]	26.7	42.0	45.9	61.1	36.3	51.6
HyDE ^[18]	25.8	39.4	46.4	63.7	36.1	51.6
GraphRAG ^[21]	27.5	41.8	52.9	64.3	40.2	53.1
LightRAG ^[22]	28.3	42.3	54.6	65.8	41.5	53.8
HippoRAG2 ^[19]	28.9	40.4	55.7	69.4	42.3	54.9
KG ² RAG ^[6]	30.3	41.9	57.2	66.3	43.8	54.1
KAG ^[20]	<u>36.3</u>	<u>48.1</u>	<u>59.9</u>	74.5	<u>48.1</u>	<u>61.3</u>
KGIR-RAG	37.7	50.6	61.3	<u>74.2</u>	49.5	62.4

注:表中加粗值为同列中的最大值,下划线值为次优值。Average表示该指标在两个数据集上的平均值。

息的局限性。2) 迭代查询重构过程中充分利用已有信息,使复杂问题逐步简化,并使改写后的问题不偏离原始问题。本文方法与IRCoT方法相比,在生成前多一步对已检索信息的筛选,使生成的信息更加精确,有效缓解了因检索信息存在噪声从而降低生成信息的质量,最终导致无效迭代的问题。3) 反思推理时根据历史改写过程和证据子图准确给出缺失信息,有利于下一步的迭代。

3.3.3 检索性能实验

为系统验证本文提出方法在多跳场景下的性能表现,本实验选取一些经典嵌入模型,以及近年来提出的先进模型作为对比基准。通过对比各模型在HotpotQA^[23]和MuSiQue^[24]数据集上的Recall@2和

表5 各方法端到端问答性能对比(LLaMA3:8B)

Table 5 Comparison of end-to-end question answering performance across methods (LLaMA3:8B)

Method	MuSiQue		HotpotQA		Average	
	EM	F1	EM	F1	EM	F1
Naive Gen	2.9	7.1	20.8	30.6	11.9	18.9
NativeRAG ^[16]	21.2	32.9	42.3	54.7	31.8	43.8
IRCoT ^[8]	29.4	35.9	57.3	58.5	43.4	47.2
ITER-RETGEN ^[7]	25.3	41.7	43.5	59.3	34.4	50.5
HyDE ^[18]	24.4	37.8	44.7	62.8	34.6	50.3
GraphRAG ^[21]	26.8	39.6	53.6	66.7	40.2	53.2
LightRAG ^[22]	28.4	41.8	55.3	68.3	41.9	55.1
HippoRAG2 ^[19]	27.5	40.9	53.4	68.0	40.5	54.5
KG ² RAG ^[6]	29.7	40.3	55.8	64.8	42.8	52.6
KAG ^[20]	<u>34.8</u>	<u>45.9</u>	<u>58.4</u>	<u>72.1</u>	<u>46.6</u>	<u>59.0</u>
KGIR-RAG	35.3	46.4	59.3	73.6	47.3	60.0

注:表中加粗值为同列中的最大值,下划线值为次优值。Average表示该指标在两个数据集上的平均值。

Recall@5指标,评估其检索性能,具体的实验结果见表6。

从表6的实验结果可知,本文提出的KGIR-RAG方法在多个检索指标上均显著优于基线方法。在MuSiQue数据集上,相比KAG方法,KGIR-RAG方法将Recall@2和Recall@5指标分别提升0.6和1.5个百分点。在HotpotQA数据集上,Recall@2和Recall@5指标分别提升1.5和0.7个百分点。在各类基线方法中BM25、Contriever和RAPTOR属于单步检索方法,IRCoT、HyDE、ITER-RETGEN、KG²RAG、KAG和KGIR-RAG属于多步检索方法。多步检索方法采用迭代检索、多路召回或者知识图谱增强等策略提升召回率。KGIR-RAG方法在召

表6 各方法检索性能对比

Table 6 Comparison of retrieval performance across methods

Method	MuSiQue		HotpotQA		Average	
	Recall@2	Recall@5	Recall@2	Recall@5	Recall@2	Recall@5
BM25 ^[25]	32.3	41.2	55.4	72.2	43.85	56.70
Contriever ^[26]	34.8	46.6	57.2	75.5	46.00	61.05
RAPTOR ^[27]	35.7	45.3	58.1	71.2	46.90	58.25
HyDE ^[18]	37.5	48.3	65.6	79.1	51.60	63.70
ITER-RETGEN ^[7]	39.7	52.9	65.8	79.3	52.80	66.10
IRCoT ^[8]	41.7	53.7	67.9	82.0	54.80	67.90
KG ² RAG ^[6]	45.1	63.4	70.5	84.7	57.80	74.10
KAG ^[20]	<u>48.5</u>	<u>65.7</u>	<u>72.8</u>	<u>88.8</u>	<u>60.70</u>	<u>77.30</u>
KGIR-RAG	49.1	67.8	74.3	89.5	61.70	78.70

注:表中加粗值为同列中的最大值,下划线值为次优值。Average表示该指标在两个数据集上的平均值。

回率上有所提升主要得益于子图探寻过程中能准确筛选出关键信息,用筛选后的信息进行问题改写能有效简化问题,使检索更加精确。

3.4 消融实验

消融实验中统一采用 Qwen2. 5:72B 模型,在 MuSiQue 数据集上通过端到端生成性能指标验证各模块的作用。具体消融实验方法如下:

- 1) w/o 子图构建:当缺失子图探寻模块时用与问题向量相似性最高的段落替代证据子图进行改写,并对改写后问题进行反思推理迭代。
- 2) w/o 反思推理:仅用证据子图进行一次改写,并用改写后的问题进行一次检索回答。
- 3) w/o 查询重构:用检索段落的证据子图直接回答,不对问题进行改写和反思迭代。

消融实验的结果如表 7 所示。当用与问题向量相似性最高的段落来替代证据子图时,EM 和 F1 指标分别下降了 4. 2 和 4. 8 个百分点,这表明证据子图构建

模块不仅能有效去除段落中冗余信息,还能准确筛选出与问题相关的关键信息。体现的是证据子图构建模块对检索信息的筛选作用,使问题回答更加准确。当无反思迭代过程时,仅用证据子图进行一次改写,并进行一次检索回答,此时 EM 和 F1 指标分别下降了 9. 4 和 8. 9 个百分点。该结果表明,适当增加迭代次数能对复杂问题进行逐步求解,迭代过程能检索出更多相关信息,更有利于问题的回答。当用检索段落的证据子图直接回答,不对问题进行改写和反思迭代时,EM 和 F1 指标下降最多,分别为 11. 0 和 8. 3 个百分点。这表明一次检索所构建的证据子图还不包括回答问题所需的全部信息,在无法有效补充相关信息时,回答的性能指标会显著下降。

因此,子图构建、迭代查询重构和反思推理这三个模块共同作用能有效提高端到端问答性能。证据子图构建模块对检索信息进行筛选过滤,迭代查询重构模块利用有效信息将问题简化,反思推理模块给出最终答案或者还需补充的信息。

3.5 案例分析

如图 6 所示,通过基于知识图谱的迭代改写检索增强方法回答“2002 年哪家公司与 1994 年顶级 PC 制造商合并”这一问题。首先根据问题检索出相关段落,抽取三元组并利用知识探寻模块对相关路径三元组进行重排序,筛选出最相关的三元组。此时得到的最相关信息为:1994 年顶级 PC 制造商为康柏公司,利用此信息对原始问题进行改写并将问

表 7 MuSiQue 数据集上的消融实验

Table 7 Ablation experiment result on MuSiQue

方法	EM	F1
KGIR-RAG	37. 7	50. 6
w/o 子图构建	33. 5(−4. 2)	45. 8(−4. 8)
w/o 反思推理	28. 3(−9. 4)	41. 7(−8. 9)
w/o 查询重构	26. 7(−11. 0)	42. 3(−8. 3)

注:表中加粗值为同列中的最大值;括号中的数值为该方法与 KGIR-RAG 之差。

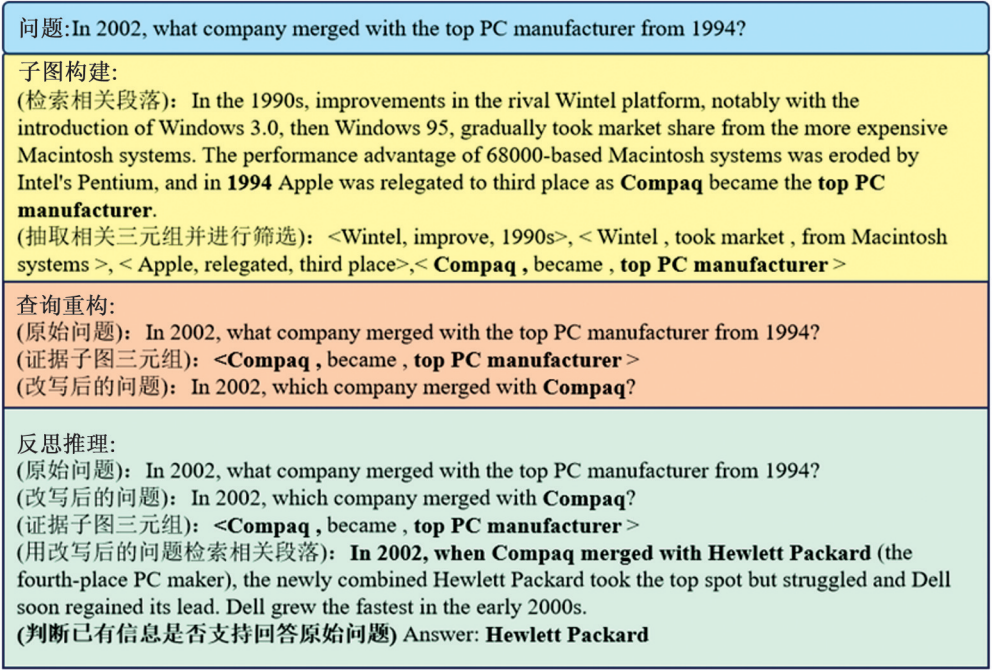


图 6 案例分析
Fig. 6 Case study

题简化为:“在2002年,康柏公司与哪家公司合并?”针对改写后的问题检索相关信息,并综合已有信息判断是否支持回答当前问题。此时检索到2002年康柏与惠普(第四大PC制造商)完成合并,因此最终答案为:惠普公司。

该案例展示了如何通过一次完整的迭代过程回答多跳问题,验证了本文提出的基于知识图谱的迭代式问答方法在处理复杂事实推理任务中的有效性,其通过子图构建、查询重构和反思推理相结合,显著提升了复杂问题的解答准确率。

4 结 语

本文针对传统RAG方法通过迭代检索策略来回答复杂多跳问题时,容易出现错误传播和无效迭代的问题,引入证据子图构建模块将段落信息构造为证据子图并结合连通图和最大生成树算法提高检索信息的质量。再进一步利用高质量的证据子图对问题进行改写,使多跳问题逐步简化,使之更易于回答。最后采用反思推理模块综合利用证据子图和改写过程的信息对问题进行回答,如果还缺失信息则进行下一轮迭代。在HotpotQA和MuSiQue两个多跳推理数据集上进行的对比实验表明,所提出方法在准确率和F1指标上显著优于现有主流基线方法。消融实验进一步验证了各模块设计的必要性与有效性,表明各组件对整体性能的贡献。通过案例分析更直观地展示各阶段的推理过程,进一步验证了所提出方法的有效性。

尽管本研究在多跳推理场景中取得进展,但受限于知识图谱的覆盖率与动态更新机制,仍存在改进空间。未来工作将聚焦于构建更全面的知识图谱、优化跨模态对齐策略,并探索多模态信息的深度融合。

参考文献:

- [1] LI D W, YANG S, TAN Z, *et al.* DALK: Dynamic co-augmentation of LLMs and KG to answer Alzheimer's disease questions with scientific literature[C]//Findings of the Association for Computational Linguistics: EMNLP 2024. Stroudsburg: Association for Computational Linguistics, 2024: 2187-2205. DOI:10.18653/v1/2024.findings-emnlp.119.
- [2] COLOMBO A. Leveraging knowledge graphs and LLMs to support and monitor legislative systems[C]//Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. New York: ACM, 2024: 5443-5446. DOI:10.1145/3627673.3680268.
- [3] XU Z T, CRUZ M J, GUEVARA M, *et al.* Retrieval-augmented generation with knowledge graphs for customer service question answering[DB/OL]. [2025-02-03]. <https://dl.acm.org/doi/pdf/10.1145/3626772.3661370>.
- [4] GUTIÉRREZ B J, SHU Y H, GU Y, *et al.* HippoRAG: Neurobiologically inspired long-term memory for large language models [DB/OL]. [2025-02-04]. <https://www.proceedings.com/content/079/079017-1902open.pdf>.
- [5] KARPUKHIN V, OGUZ B, MIN S, *et al.* Dense passage retrieval for open-domain question answering [C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg: Association for Computational Linguistics, 2020: 6769-6781. DOI:10.18653/v1/2020.emnlp-main.550.
- [6] ZHU X R, XIE Y X, LIU Y, *et al.* Knowledge graph-guided retrieval augmented generation[C]//Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Stroudsburg: Association for Computational Linguistics, 2025: 8912-8924. DOI:10.18653/v1/2025.naacl-long.449.
- [7] SHAO Z H, GONG Y Y, SHEN Y L, *et al.* Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy[C]//Findings of the Association for Computational Linguistics: EMNLP 2023. Stroudsburg: Association for Computational Linguistics, 2023: 9248-9274. DOI:10.18653/v1/2023.findings-emnlp.620.
- [8] TRIVEDI H, BALASUBRAMANIAN N, KHOT T, *et al.* Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg: Association for Computational Linguistics, 2023: 10014-10037. DOI:10.18653/v1/2023.acl-long.557.
- [9] JEONG S, BAEK J, CHO S, *et al.* Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity [C]//Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Stroudsburg: Association for Computational Linguistics, 2024: 7036-7050. DOI:10.18653/v1/2024.naacl-long.389.
- [10] CHAN C M, XU C P, YUAN R B, *et al.* RQ-RAG: Learning to refine queries for retrieval augmented generation [EB/OL]. 2024: arXiv: 2404.00610. <https://arxiv.org/abs/2404.00610>
- [11] 黄勃, 吴申奥, 王文广, 等. 图模互补: 知识图谱与大模型融合综述[J]. 武汉大学学报(理学版), 2024, 70(4): 397-412. DOI: 10.14188/j.1671-8836.2024.0040.

- HUANG B, WU S A, WANG W G, *et al.* KG-LLM-MCom: A survey on integration of knowledge graph and large language model [J]. *J Wuhan Univ (Nat Sci Ed)*, 2024, **70** (4): 397-412. DOI: 10.14188/j.1671-8836.2024.0040(Ch).
- [12] RAM O, LEVINE Y, DALMEDIGOS I, *et al.* In-context retrieval-augmented language models [J]. *Transactions of the Association for Computational Linguistics*, 2023, **11**: 1316-1331. DOI:10.1162/tacl_a_00605.
- [13] KIM G, KIM S, JEON B, *et al.* Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2023: 996-1009. DOI: 10.18653/v1/2023.emnlp-main.63.
- [14] BORGEAUD S, MENSCH A, HOFFMANN J, *et al.* Improving language models by retrieving from trillions of tokens[EB/OL]. 2021: arXiv: 2112.04426. <https://arxiv.org/abs/2112.04426>.
- [15] YORAN O, WOLFSON T, RAM O, *et al.* Making retrieval-augmented language models robust to irrelevant context[EB/OL]. 2023: arXiv: 2310.01558. <https://arxiv.org/abs/2310.01558>.
- [16] SANTHANAM K, KHATTAB O, SAAD-FALCON J, *et al.* ColBERTv2: Effective and efficient retrieval *via* lightweight late interaction[C]//Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics, 2022: 3715-3734. DOI: 10.18653/v1/2022.naacl-main.272.
- [17] HUANG L, YU W J, MA W T, *et al.* A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions[J]. *ACM Transactions on Information Systems*, 2025, **43**(2): 1-55. DOI:10.1145/3703155.
- [18] GAO L Y, MA X G, LIN J, *et al.* Precise zero-shot dense retrieval without relevance labels[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg: Association for Computational Linguistics, 2023: 1762-1777. DOI:10.18653/v1/2023.acl-long.99.
- [19] GUTIÉRREZ B J, SHU Y H, QI W J, *et al.* From RAG to memory: Non-parametric continual learning for large language models [EB/OL]. 2025: arXiv: 2502.14802. <https://arxiv.org/abs/2502.14802>.
- [20] LIANG L, BO Z P, GUI Z K, *et al.* KAG: Boosting LLMs in professional domains *via* knowledge augmented generation [C]//Proceedings of the ACM on Web Conference 2025. New York: Association for Computing Machinery. 2025: 334-343. DOI:10.1145/3701716.3715240.
- [21] EDGE D, TRINH H, CHENG N, *et al.* From local to global: A graph RAG approach to query-focused summarization[EB/OL]. 2024: arXiv: 2404.16130. <https://arxiv.org/abs/2404.16130>.
- [22] GUO Z R, XIA L H, YU Y H, *et al.* LightRAG: Simple and fast retrieval-augmented generation[C]//Findings of the Association for Computational Linguistics: EMNLP 2025. Stroudsburg: Association for Computational Linguistics, 2025: 10746-10761. DOI: 10.18653/v1/2025.findings-emnlp.568.
- [23] TRIVEDI H, BALASUBRAMANIAN N, KHOT T, *et al.* MuSiQue: Multihop questions *via* single-hop question composition [J]. *Transactions of the Association for Computational Linguistics*, 2022, **10**: 539-554. DOI:10.1162/tacl_a_004759.
- [24] YANG Z L, QI P, ZHANG S Z, *et al.* HotpotQA: A dataset for diverse, explainable multi-hop question answering [C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium. Stroudsburg: Association for Computational Linguistics, 2018: 2369-2380. DOI:10.18653/v1/d18-125.
- [25] BILLEL A, IBRAHIM B, YAHYA S. BM25 beyond query-document similarity[C]//String Processing and Information Retrieval: 26th International Symposium on String Processing and Information Retrieval (SPIRE 2019). Segovia: Springer Nature Switzerland AG, 2019: 65-79.
- [26] YIBIN L, LIANG D, YU C, *et al.* Dense retrieval with relevance-aware contrastive pre-training[C]// In Findings of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2023: 10932-10940. DOI:10.18653/v1/2023.findings-acl.695
- [27] SARTHI P, ABDULLAH S, TULI A, *et al.* RAPTOR: Recursive abstractive processing for tree-organized retrieval [EB/OL]. 2024: arXiv: 2401.18059. <https://arxiv.org/abs/2401.18059>.

□