计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(12):75−88 [doi: 10.15888/j.cnki.csa.010017] [CSTR: 32024.14.csa.010017]

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

大语言模型驱动的碳知识库构建与应用①

芦成飞

(成都信息工程大学 软件工程学院, 成都 610225)
通信作者: 芦成飞, E-mail: lucf09@qq.com

摘　要: 大语言模型 (large language model, LLM) 在自然语言理解与生成领域展现出卓越能力, 但在特定领域知识密集型任务中仍面临事实准确性不足、知识更新难以及高质量领域数据集匮乏的问题. 在应对上述难题时, 检索增强生成 (retrieval-augmented generation, RAG) 技术脱颖而出, 成为行之有效的解决路径. 然而, 在应对碳领域的知识密集型任务时, RAG 技术还存在查询理解环节容易出现偏差、外部知识检索策略僵化单一、检索得到的结果与实际需求的相关性较差等短板, 同时缺乏特定的数据集来评估问答效果. 针对以上问题, 提出基于多管道的检索增强生成 (Multi-pipeline-based RAG) 方法, 使用本文提出的图谱增强递归式智能合并检索, 有效提升了检索精确率; 针对特定领域问答数据集的缺乏, 提出基于父节点文本的大模型自动生成问答数据集方法. 同时在传统评估指标, 如精确率 (*Precision*)、召回率 (*Recall*) 等基础上, 利用 LLM 的文本理解能力评估: (1) 响应-上下文-查询相关性评估; (2) 响应-查询相关性评估; (3) 上下文-查询相关性评估; (4) 忠诚性评估. 通过与 BM25-based RAG、Vector-based RAG、Recursive-based RAG 的对比实验, 基于 GLM-4-Plus 模型的 Multi-pipeline-based RAG 精确率达到了 85%, 高于其他方法.

关键词: 大语言模型; 知识图谱; 混合检索; 双碳; 检索增强生成

引用格式: 芦成飞.大语言模型驱动的碳知识库构建与应用.计算机系统应用,2025,34(12):75–88. http://www.c-s-a.org.cn/1003-3254/10017.html

Construction and Application of Carbon Knowledge Base Driven by Large Language Model

LU Cheng-Fei

(School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: The large language model (LLM) demonstrates excellent capabilities in natural language understanding and generation. However, it still faces challenges such as insufficient factual accuracy, difficulties in knowledge updating, and a lack of high-quality domain-specific datasets in knowledge-intensive tasks. To address these challenges, retrieval-augmented generation (RAG) has emerged as an effective solution. However, when applied to knowledge-intensive tasks in the carbon domain, RAG technology has limitations, including potential bias in query understanding, rigid external knowledge retrieval strategies, poor correlation between retrieved results and actual needs, and a lack of specific datasets for evaluating question-answering performance. To tackle these issues, this study proposes a Multi-pipeline-based RAG method, which utilizes the graph-enhanced recursive intelligent merge retrieval method to effectively improve retrieval accuracy. For the lack of Q&A datasets in specific domains, a large model-based approach is proposed to automatically generate Q&A datasets from the parent node text. Moreover, this study evaluates the following aspects using the text understanding capability of LMM, alongside traditional evaluation metrics such as precision and recall: (1) response-context-query correlation; (2) response-query correlation; (3) context-query correlation, and (4) loyalty evaluation. Experimental results show that the Multi-pipeline-based RAG method based on the GLM-4-Plus model achieves an

① 收稿时间: 2025-05-12; 修改时间: 2025-06-05; 采用时间: 2025-06-20; csa 在线出版时间: 2025-10-21
CNKI 网络首发时间: 2025-10-22

accuracy of 85%, outperforming BM25-based RAG, Vector-based RAG, and Recursive-based RAG methods.

Key words: large language model (LLM); knowledge graph (KG); hybrid retrieval; dual carbon; retrieval-augmented generation (RAG)

1 引言

大型语言模型 (large language model, LLM)[1]依托于庞大的训练数据集, 吸纳并内化了广泛的世界知识于模型参数之中, 展现出卓越的自然语言理解及文本创造潜能. 作为与大语言模型交互的基础方法, 提示工程是一项用于引导模型在多元任务中生成精准且相关输出的关键技术体系. LLM 可以通过提示工程技术、解码调整、生成微调等方式提高生成质量[2]. 在此过程中, 提示内容的设计与构建成为充分释放大语言模型性能潜力的核心影响要素[3]. 王海涛等人[4]提出 Query2-Query 方法, 设计 TAO 提示词框架, 利用大语言模型生成能力, 结合“What、How、Why 黄金法则”重写查询, 提升检索准确率. 王东清等人[5]强调通过基础或高级提示方法提升模型响应效果, 无需微调参数. 然而, 提示工程发展进程中亟需攻克的难点之一, 在于应对自然语言与生俱来的模糊特性[6]. 近些年来, 像 ChatGPT、文心一言这类具有代表性的通用大模型, 在自然语言处理领域成绩斐然, 极大地优化了问答系统的性能表现[7]. 然而, 当涉及特定垂直领域时, 这些通用大模型却暴露出诸多问题. 在面对复杂的专业术语、独特的作业流程以及行业规范等内容时, 其实际表现与在通用领域的出色成果相比, 存在着较为显著的落差[8,9].

为了应对这些挑战, 检索增强生成 (retrieval-augmented generation, RAG)[10]策略应运而生. 通常 RAG 方法可分为基于向量的 RAG 和基于图的 RAG[11]. 两者本质上都是将非参数化的知识存储系统与参数化的语言模型相结合, 为 LLM 引入了一个能够动态调整的外部知识库, 进而增强其文本生成效能. RAG 技术能够将从外部资源中筛选出关联信息, 作为关键线索融入 LLM 的输入提示中. 这一过程优化了模型的文本生成机制, 尤其在处理知识密集型任务时, 显著降低了因知识匮乏导致的事实性错误发生率, 有效提升了模型输出内容的准确性与可靠性[12]. 典型的检索增强生成 (RAG) 流程包含以下核心步骤. 首先, 用户输入查询问题, 由检索器对问题进行向量化处理, 并从预构建的向量数据库中检索匹配的相关数据源; 随后, 检索结果与生成模型进行交互融合, 通过引入外部知识优化内容生成的质量与上下文相关性, 从而实现对实际问题的精准解答[13]. 尽管 RAG 技术具备显著优势, 但其在技术实现层面仍存在特定难题——检索模块与生成模块深度融合会不可避免地提升模型构建的复杂程度[14]. 作为 RAG 模型的关键构成部分, 检索模块在从知识库中精准获取关联信息的过程中起核心支撑作用[15], 若仅依赖单一的检索策略, 弊端同样突出. 外接知识库的固定性使得知识段落内容无法根据问题的变化进行动态调整. 较长的知识段落中大量专业性词汇的存在, 在知识与问题拼接时极易引入无关信息, 进而产生噪声. 此外, 由于部分问题的答案所需知识分散在多个段落中, 传统的单一处理方式是将多个段落同时作为生成器的输入. 然而, 这种方式在引入所需知识的同时, 也不可避免地引入了更多噪声. 这些噪声干扰了生成器的正常工作, 严重影响答案生成的准确性, 最终导致生成的答案出现与问题不相关甚至错误的内容, 极大地降低了系统的可靠性和实用性. 在生物医学的智能问答领域, Yang 等人[16]采用 BM25 检索加大模型微调来提高检索精度. 在双碳知识领域中, 齐俊等人[17]采用知识图谱检索和微调来增强大语言模型在碳达峰碳中和领域中的应用. 但微调大模型存在数据依赖高、计算资源与时间成本大、性能有局限、难快速适应变化、易出现知识遗忘、可解释性差等缺点. 针对大语言模型在智能制造中存在的领域知识差距、传统向量检索模糊及知识图谱方法扩展性不足等问题, Wan 等人[18]提出一种混合 KG-向量 RAG 框架, 通过构建元数据增强知识图谱、注入领域语义约束及分层检索策略, 实现结构化推理与高效向量检索结合, 提升问答系统的准确性、上下文相关性和效率. 在软件持续交付领域, 因知识获取困难影响实施效果, 鞠炜刚等人[19]提出基于大语言模型和检索增强生成的智能问答系统构建方法. 该方法通过高质量语料处理形成数据集, 采用 LoRA (low-rank adaptation) [20]微调技术训练领域大模型, 运用改进的向量知识检索与多场景提示词模板技术提升生成效果.

2012 年, Google 提出知识图谱 (knowledge graph, KG)[21]这一概念, 为知识处理带来了全新思路. 它是一种结构化语义知识库, 其构建源于对结构化数据、半结构化数据或者非结构化数据的加工、处理与整合. 知识图谱以包含实体及其关系的三元组作为基本组成单位, 能够对概念以及概念之间的相互关系进行有效描述[22], 是实现知识共享、融合和挖掘的重要手段[23]. 随着人工智能与语义网[24]技术的持续进步, 知识图谱作为关键的数据处理技术和表示格式, 近年来研究成果不断涌现. 知识图谱与 LLM 已展现出显著潜力. 大型语言模型在自然语言理解、非结构化数据处理以及生成上下文相关的连贯输出方面表现突出. 尽管这类模型在生成能力和逻辑推理上具备优势, 但在专业领域仍存在局限, 例如可能产生看似合理却与事实不符的响应 (即“幻觉”现象)[25]. 在此情境下, 知识图谱能够为语义精准检索供给领域内的结构化知识体系, 以此补足 LLM 在专业知识精细化、准确化理解层面的需求缺口[26]. 近年来, 国内外学者积极探索知识图谱在碳达峰碳中和领域的应用. 比如, 有研究借助 BiLSTM-CRF[27]模型抽取碳交易三元组, 进而提出面向碳交易领域的知识图谱构建方法[28]; 还有研究基于 BERT-CRF 构建碳市场知识图谱, 并利用 neo4j 数据库进行存储与分析, 给出碳市场领域的图谱构建方案[29]. 在零件工艺设计知识方面, 田小贵等人[30]提出融合知识图谱与大模型的零件工艺设计知识问答方法, 先构建零件工艺知识图谱, 包括模式层、数据层构建及知识存储更新; 再选用 ChatGLM3-6B-32K 大模型并微调. 不过, 当前碳达峰碳中和领域的知识图谱研究尚处于初期, 无论是该领域的知识梳理, 还是基于双碳业务的知识服务问答构建等应用, 都有待进一步发展完善.

综上所述, LLM、RAG 和知识图谱存在一定的局限性, 体现在通用大模型在自然语言处理领域虽表现出色, 但在特定垂直领域, 面对复杂专业术语、独特作业流程和行业规范时, 实际表现与通用领域相比存在显著落差, 还可能产生“幻觉”现象, 即生成看似合理却与事实不符的响应. RAG 检索模块存在如下弊端,一方面作为 RAG 模型关键部分, 若仅依赖单一检索策略, 外接知识库固定, 知识段落内容无法动态调整, 易引入无关信息产生噪声; 此外在处理答案需多段落知识的问题时, 传统方式会引入过多噪声, 干扰生成器工作, 降低答案准确性, 影响系统可靠性和实用性. 在碳达峰碳中和领域中, 知识图谱研究不够完善, 当前该领域知识图谱研究尚处初期, 无论是知识梳理, 还是基于双碳业务的知识服务问答构建等应用, 都有待进一步发展完善. 此外还存在问答数据集缺失、智能应答能力不足且准确率较低等问题.

为应对上述问题, 本研究提出基于多管道的检索增强生成 (Multi-pipeline-based RAG) 方法及基于文本索引的大模型自动生成问答数据集方法. 其中, 基于多管道的检索增强生成方法由多个功能模块协同运作. 首先, 针对复杂多变的查询场景, 通过特定管道对输入查询进行深度语义解析, 精准捕捉用户意图, 避免因语义理解模糊导致的响应偏差; 其次, 在检索阶段, 利用索引管道和总结管道对外部知识源进行针对性检索, 各管道依据策略筛选出与查询紧密相关的知识内容, 有效规避无关信息干扰, 提升检索效率与质量; 最后, 在生成响应环节, 将检索的知识与查询相结合, 通过优化的生成管道输出准确且贴合需求的回答, 确保响应-查询及检索知识的高度一致性. 而对于特定领域问答数据集匮乏的问题, 基于文本索引的大模型自动生成问答数据集方法则先构建完善的文本索引体系, 将特定领域文本按语义、主题等维度进行精准分类与索引. 然后, 利用大模型对索引后的文本进行深度挖掘, 自动提取关键信息生成问题, 并基于文本内容生成相应答案, 以此快速构建高质量的特定领域问答数据集. 此外, 在评估环节, 除传统的精确率、召回率等指标外, 借助大语言模型强大的文本理解能力, 从响应-上下文-查询相关性、响应-查询相关性、上下文-查询相关性以及忠诚性这 4 个维度展开全面评估, 确保模型输出的可靠性与准确性.

综上所述, 本研究的贡献主要有以下几点.

(1) 提出基于多管道的检索增强生成方法: 针对复杂多变的查询场景, 构建多个功能模块协同运作的基于多管道的检索增强生成方法. 通过深度语义解析精准捕捉用户意图, 利用索引与总结管道针对性检索外部知识, 结合优化生成管道输出贴合需求的回答, 有效提升检索效率与响应准确性.

(2) 构建基于文本索引的问答数据集生成方法: 为解决特定领域问答数据集匮乏问题, 提出基于文本索引的大模型自动生成问答数据集方法. 通过构建完善的文本索引体系, 对特定领域文本精准分类与索引, 利用大模型挖掘文本生成问题及答案, 实现高质量特定

领域问答数据集的快速构建.

(3) 完善评估体系: 突破传统评估指标局限, 借助大语言模型文本理解能力, 从响应-上下文-查询相关性、响应-查询相关性、上下文-查询相关性以及忠诚性这 4 个维度, 对模型输出进行全面评估, 确保模型输出的可靠性与准确性.

2 基于多管道的检索增强生成框架

为应对双碳领域复杂的知识问答需求, 本文设计了 Multi-pipeline-based RAG 框架. 此框架巧妙融合了 RAG 策略与大语言模型, 为精准解答双碳领域的各类问题提供了强有力的支撑, 极大地提升了对双碳相关查询的回答质量与准确性, 有效推动了双碳领域知识密集型任务的顺利开展.

具体而言, 当用户提出一个与双碳领域紧密相关的查询时, 系统将依托精心构建的双碳知识库中的文献资料、政策文件以及前沿研究片段, 协同大语言模型, 快速生成既准确又有充分依据的答案, 并附上详实且合理的解释, 让用户不仅知其然, 还知其所以然.

本文提出的面向双碳领域知识问答的框架 Multi-pipeline-based RAG, 其流程框架如图 1 所示, Multi-pipeline-based RAG 框架主要涵盖 3 个关键阶段: 数据准备阶段、模型微调阶段以及检索推理阶段.

数据准备阶段, 研究团队广泛收集各地工业信息化厅发布的碳政策文件. 例如, 江苏省工业和信息化厅发布的《江苏省工业领域及重点行业碳达峰实施方案》, 详细规划了当地工业领域碳达峰的总体要求、重点任务和保障措施, 其中对产业结构调整、节能降碳、绿色制造等方面提出了明确目标与实施路径, 为双碳知识库提供了重要的政策依据与实践指导方向. 河南省工业和信息化厅印发的《河南省工业领域碳达峰实施方案》, 不仅设定了当地工业领域碳达峰的阶段性目标, 还围绕产业结构调整、节能降碳、循环经济等多个重点任务展开部署, 对于深入理解双碳目标在工业领域的落地策略具有重要价值. 这些丰富且权威的政策数据, 为后续知识图谱的构建、知识索引的构建和检索器筑牢根基, 确保 Multi-pipeline-based RAG 框架在双碳领域知识问答中能够紧密贴合政策导向, 生成具有权威性与实用性的答案. 在获取相关文档后, 该框架会对源文档基于一定策略分割成父文本块和子文本块 (pipeline 1), 并且分割后的每个文本块会交由给 LLM 进行文本总结 (pipeline 2), 并最终加入索引中去. 而父文本块会交由 LLM 去做问答数据集的构建.

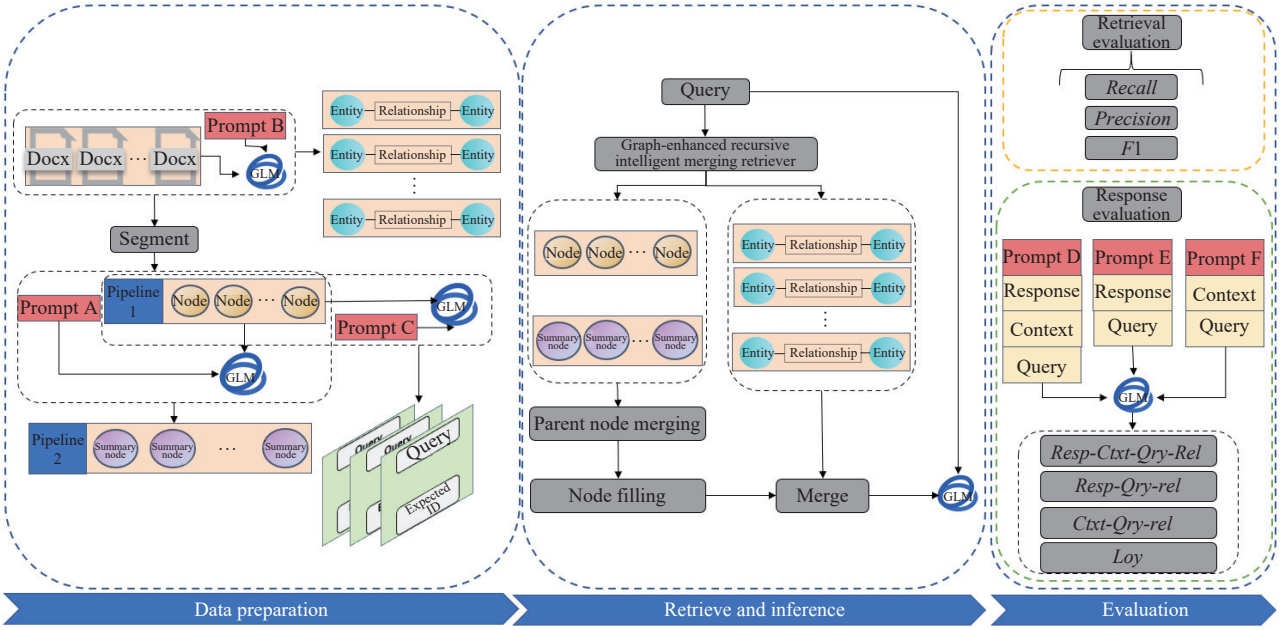

图 1 Multi-pipeline-based RAG 概述

在检索推理阶段, 当用户提出问题后, 系统并行启动递归检索和图谱检索两种方式. 递归检索深入挖掘相关信息, 获取对应的 Node; 图谱检索则从知识图谱维度中探寻, 也得到相应的 Node. 针对递归检索获取

的 Node, 会依次进行填充 Node 和合并 Node 的操作, 旨在完善节点信息并优化结构. 之后, 将处理后的递归检索 Node 与图谱检索得到的 Node 一同进行合并操作, 最终将其输入 LLM 中, 为生成精准答案提供优质素材.

评估阶段是确保知识问答系统性能与答案质量的关键环节, 本系统将评估阶段细分为检索评估和响应评估两个部分.

● 检索评估主要聚焦于衡量检索模块的性能表现, 采用召回率 (*Recall*) 和精确率 (*Precision*) 两个关键指标. 召回率用于评估系统在检索过程中, 能够准确找到与问题相关信息的能力, 即实际相关的信息中有多少被成功检索出来. 精确率则侧重于评估检索结果的准确性, 即检索出的信息中真正与问题相关的比例. 通过对这两个指标的计算与分析, 能够全面了解检索模块是否高效且准确地获取了所需信息, 进而为优化检索策略提供数据支撑.

● 响应评估聚焦于对系统基于检索信息所生成的答案内容进行细致评估, 具体涵盖以下 3 个维度.

1) 响应-上下文-查询相关性评估: 主要考察系统生成的响应和给定上下文-查询之间的关联紧密程度. 判断响应和给定上下文是否能够充分利用查询相关事实等信息, 确保答案在整体语境中具备合理性与连贯性, 避免出现与查询脱节或矛盾的情况.

2) 响应-查询相关性评估: 着重评估系统生成的响应与用户原始查询之间的契合度. 检查响应是否准确针对查询所提出的问题进行作答, 是否全面涵盖了查询所涉及的关键要点, 能否切实满足用户的信息需求.

3) 上下文-查询相关性评估: 主要分析给定的上下文与用户查询之间的关联程度. 确认上下文所提供的信息是否与查询主题紧密相关, 是否能够为解答查询问题提供有价值的辅助信息, 以保障整个问答过程在知识体系上的一致性与逻辑性.

在响应评估过程中, 系统会依据预先设定的评估准则, 将查询、响应及上下文信息整合为提示 (prompt), 输入至 LLM 中. 借助大型语言模型的强大分析能力, 从答案的准确性、完整性、逻辑性以及语言表达的流畅性等多个层面进行综合考量, 从而对响应质量做出客观、准确的评判, 确保系统输出的答案能够切实为用户提供有价值、高质量的信息服务.

3 方案设计

3.1 数据准备

3.1.1 文本分块与文本嵌入

数据准备阶段包含两大核心管道: (1) 文本分块管道, 将文档分割为多粒度原始文本节点; (2) 文本总结管道, 通过 LLM 生成摘要节点. 两者共同构建混合索引库, 支持细节型与概括型查询的差异化匹配.

由于碳领域政策文件具有显著的结构化特征 (如章节分明、条款独立), 且包含大量专业术语与数值指标 (如 “十四五” 目标、绿色工厂数量). 文本分块策略需在语义完整性、检索效率与多粒度适配之间实现平衡. 为此, 本研究采用非重叠分块加多粒度子块策略. 初步分割产出的文本块称为父文本块. 父文本块大小 *chunk_size* 设置为 2 048, 每个父文本块都会被封装成一个节点对象. 为确保每个节点具有唯一标识, 会为每个节点赋予一个符合特定格式的 ID, 即 node-*index*. 其中, *index* 为在列表中的序号. 这种统一的 ID 格式便于后续对节点进行管理和引用. 然后将每一个父文本块进一步分割为 3 组子文本块, 3 组子文本块的大小 *sub_chunk_sizes* 分别为 256, 512, 1 024. 与父文本块一样, 子文本块也将封装成一个节点对象. 文本分割逻辑如图 2 所示.

在知识索引构建流程中, 系统会将各节点的文本块, 通过精心设计的提示词发送至大语言模型. 这些提示词涵盖总结任务指令、关键词引导、输出格式规范等要素, 以此引导大模型对文本内容进行语义解析、关键信息提取和精简概括. 经大模型处理后生成的总结文本, 将作为新增节点, 依据既定的索引规则, 有序整合至现有知识索引体系内.

随后利用嵌入 (embedding) 模型将这些节点映射为数值向量. 嵌入模型本文选择 jina-embeddings-v3[31], 该嵌入模型输出 1 024 维向量, 是拥有 5.7 亿参数的前沿文本嵌入模型, 它在多语言和长文本检索任务上实现了最先进的性能, 支持长达 8 192 个 token 的输入长度, 同时该模型具有特定任务的低秩适应 (LoRA) 适配器, 使其能够为文档检索、聚类、分类和文本匹配等各种任务生成高质量嵌入.

3.1.2 构建双碳问答数据集

在知识问答系统的构建与应用中, 借助 LLM 生成

问答对是丰富双碳领域知识储备的重要举措. 具体来讲, 通过精心构造的 prompt, LLM 能够依据每个父节点的文本, 生成一系列问题. 表 1 为构建双碳问答数据集功能的输入输出以及提示词.

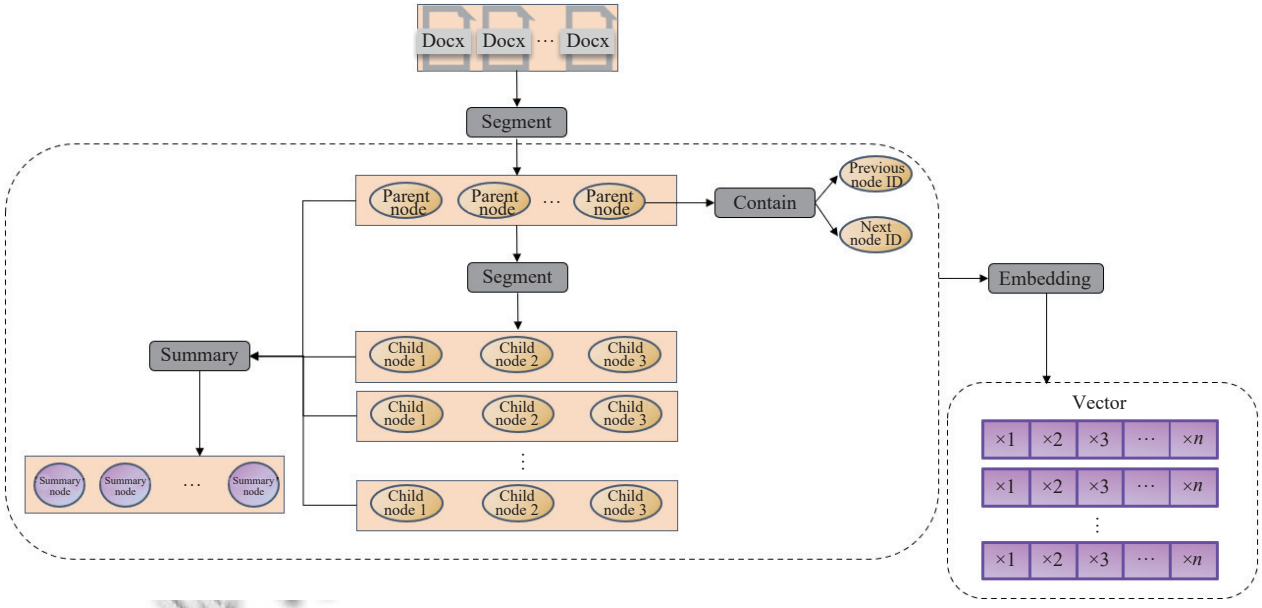

图 2 文档分割逻辑

表 1 针对双碳问答数据集构建的提示词及其输入输出

功能	输入	输出	提示词
构建双碳问答数据集	父节点	问题、父节点ID	‘‘‘上下文信息如下: \n {context_str} 你是一名双碳领域的专家. 你的任务是必须从给定的上下文信息和metadata中生成1个中文问题, 而不是先验知识. 问题类型必须为简答题类型, 生成的问题必须包含metadata中的file_name中的主体, 明确问的是哪个主体. \n 将问题限制在提供的上下文信息中. ’’’

注: ‘‘‘ ’’’用于将多行文字转化为字符串, 下同

Prompt 作为与 LLM 沟通的关键要素, 其设计需精确且具有引导性. 在输入过程中, 将父节点所承载的双碳相关文本信息融入 prompt, 清晰地向 LLM 传达生成问答对的任务指令. LLM 凭借其内部深度神经网络架构, 具备强大的语义理解与生成能力. 它会对父节点文本进行深入的语义剖析, 挖掘其中的关键概念、政策要点以及技术路径等信息.

基于上述分析, LLM 依据自身所学习到的语言范式和知识体系, 按照问答对的形式进行内容创作. 生成的问题精准指向父节点文本的核心内容, 答案则是父节点文本本身和父节点 ID. 例如, 若父节点文本是关于某地区碳达峰行动方案的阐述, LLM 可能生成诸如“该地区实现碳达峰的主要能源结构调整措施有哪些?”这样的问题, 答案则围绕该地区如何提升清洁能源占比、降低化石能源消耗、推进能源消费电气化等具体举措展开, 并附带源父节点 ID. 表 2 展示了 LLM 基于父节点文本所生成的部分问题以及父节点 ID.

3.1.3 知识图谱构建

在双碳领域知识体系的构建过程中, 数据来源具有多元化与权威性的显著特征. 本研究的数据取材于各省市以及国家相关部门正式发布的碳达峰实施方案及各类通知文件, 这些一手资料涵盖了各地域、各层面在推进碳达峰进程中的战略布局、具体举措以及阶段性目标等关键信息, 为后续知识构建提供了坚实的事实依据.

模型选型方面, 选取了智谱 GLM 团队精心研发的 GLM-4-Plus 作为核心 LLM. 该模型在训练过程中展现出卓越性能, 它运用了大量模型辅助构造的高质量合成数据, 此类数据不仅丰富了模型的知识储备, 更有效规避了因真实数据稀缺或不均衡可能导致的过拟合等问题, 全方位提升了模型的整体性能. 尤其值得一提的是, 其采用近端策略优化算法 (PPO) 对模型进行

优化, 显著增强了模型在处理推理任务, 诸如数学运算、复杂代码算法题求解等方面的表现, 使得模型生成的结果能够更加精准地反映人类偏好, 契合实际应用场景中的交互需求.

表 2 部分双碳问题及其源节点 ID

序号	问题	父节点ID
1	根据文件《【政策】吉林省工业领域碳达峰实施方案.docx》中的内容, 该方案提出的“十四五”期间工业领域碳达峰的总体目标是什么?	['node-0', 'node-1']
2	根据《【政策】吉林省工业领域碳达峰实施方案.docx》的内容, 到2025年, 吉林省计划创建多少家绿色工厂?	['node-2', 'node-3']
3	请简述《【政策】吉林省工业领域碳达峰实施方案》中, 针对建材行业提出的节能降碳措施及其目标.	['node-4', 'node-5']
4	【政策】吉林省工业领域碳达峰实施方案.docx中, 哪些部门按职责分工负责建立健全统计核算和标准体系?	['node-6', 'node-7']
5	请简述《山东省工业领域碳达峰工作方案》中, 针对钢铁行业提出的哪些具体措施来推动其低碳转型?	['node-8', 'node-9']
6	根据文档《【政策】山东省工业领域碳达峰工作方案.docx》的内容, 请简述山东省在推动工业绿色微电网建设方面采取的主要措施及目标.	['node-10', 'node-11']
7	根据文件《【政策】山东省工业领域碳达峰工作方案.docx》, 请简述山东省在推动钢铁产业绿色低碳转型方面的主要措施和目标.	['node-12', 'node-13']

在知识图谱构建环节, 使用 Llamaindex 框架, 将碳知识领域非结构化文档作为输入, 同时借助精心设计的提示词驱动所选 LLM, 将双碳领域内纷繁复杂的知识信息转化为以 (主语, 谓语, 宾语) 形式呈现的知识三元组. 表 3 为针对构建知识图谱所设计的提示词及其输入与输出, 其中 text 为文档文本.

表 3 针对知识图谱构建的提示词及其输入输出

功能	输入	输出	提示词
构建知识图谱	碳知识docx格式文档	(主语, 谓语, 宾语)	‘‘‘以下提供了一些文本, 根据给定的文本, 以 (主语, 谓语, 宾语)的形式提取尽可能多的知识三元组. 避免使用停用词. \n 示例: 文本: 化石能源清洁低碳利用标准主要包括煤炭、石油、天然气等化石能源的清洁高效燃烧. 三元组: \n (化石能源清洁低碳利用标准, 包括, 石油) (化石能源清洁低碳利用标准, 包括, 煤炭) (化石能源清洁低碳利用标准, 包括, 天然气) 文本: {text}\n 三元组: \n’’’

Llamaindex 框架以单个文档作为提取目标, 从每个文档提取的实体和关系都会附加元数据, 而元数据中拥有文件名属性, 即该实体和关系是从哪个文件中提取所得. 经过系统整理与归纳的三元组通过 nGQL (Nebula Graph query language) 声明式图查询语言插入目标图数据库, 最终成功构建出一个规模宏大、涵盖内容丰富的双碳知识图谱, 该图谱包含高达 41 000 个精准且具有代表性的知识三元组. 为确保知识图谱的高效存储与便捷查询, 选用 Nebula Graph 图数据库作为存储载体, 将海量的知识三元组有序地整合其中. 双碳知识图谱的部分示例展示于图 3, 从中可以直观窥探到图谱中所蕴含的丰富知识脉络以及三元组之间的紧密逻辑关联, 为双碳领域的深入研究、知识问答以及决策支持等诸多应用场景提供强有力的知识支撑.

3.2 检索策略

本文基于知识图谱检索和递归检索, 提出了图谱增强递归式智能合并检索 (graph-enhanced recursive intelligent merging retriever, GRIM retriever), 该检索分为两个检索器: 递归检索与动态合并和知识图谱检索. 两者的检索结果将采用合并策略, 旨在结合知识图谱的结构化知识优势、递归检索的深度挖掘能力以及智能合并的优化策略, 实现高效、精准且全面的信息检索.

3.2.1 递归检索与动态合并

在复杂知识结构的检索场景下, 为实现高效的信息获取, 本文提出了一种融合递归检索与动态合并的层次化信息检索方法. 该方法旨在通过层次化检索与结构化合并相结合的策略, 将多源信息进行融合, 并对检索结果进行动态优化, 从而为下游任务, 如问答、推

理等提供高质量的结构化节点集合.

此方法采用图状组件网络来组织检索过程, 包含检索器和静态节点两种核心组件. 检索器负责从特定数据源提取相关节点, 静态节点作为预定义的基础数据单元, 是检索过程的终止节点. 这些组件以向量检索器作为入口, 构建以检索器为根的有向无环图, 支持异构组件的混合调用.

递归检索阶段包含递归查询执行方法和检索节点处理方法. 检索过程以深度优先递归方式遍历组件网络. 从递归查询执行方法出发, 通过不断深入索引结构, 根据不同类型的查询对象 (节点、检索器) 进行相应处理, 若为节点, 则直接将该节点作为结果, 赋予当前相似度分数; 若为检索器, 则检索器会检索相关节点, 并进行检索节点处理方法, 该方法首先会对相关节点进行去重处理, 确保每个节点仅被处理一次, 提高检索效率. 而对于上游检索到的不同节点类型会有不同处理方法. 对于索引节点, 会调用递归查询执行方法, 继续深入检索, 获取更多相关节点; 对于文本节点, 直接将其作为结果, 确保文本信息的完整性, 确保每个节点仅被处理一次, 提高检索效率, 以获取与查询相关的所有节点. 其核心在于利用查询对象的特性, 逐步扩展检索范围, 直到无法继续深入为止.

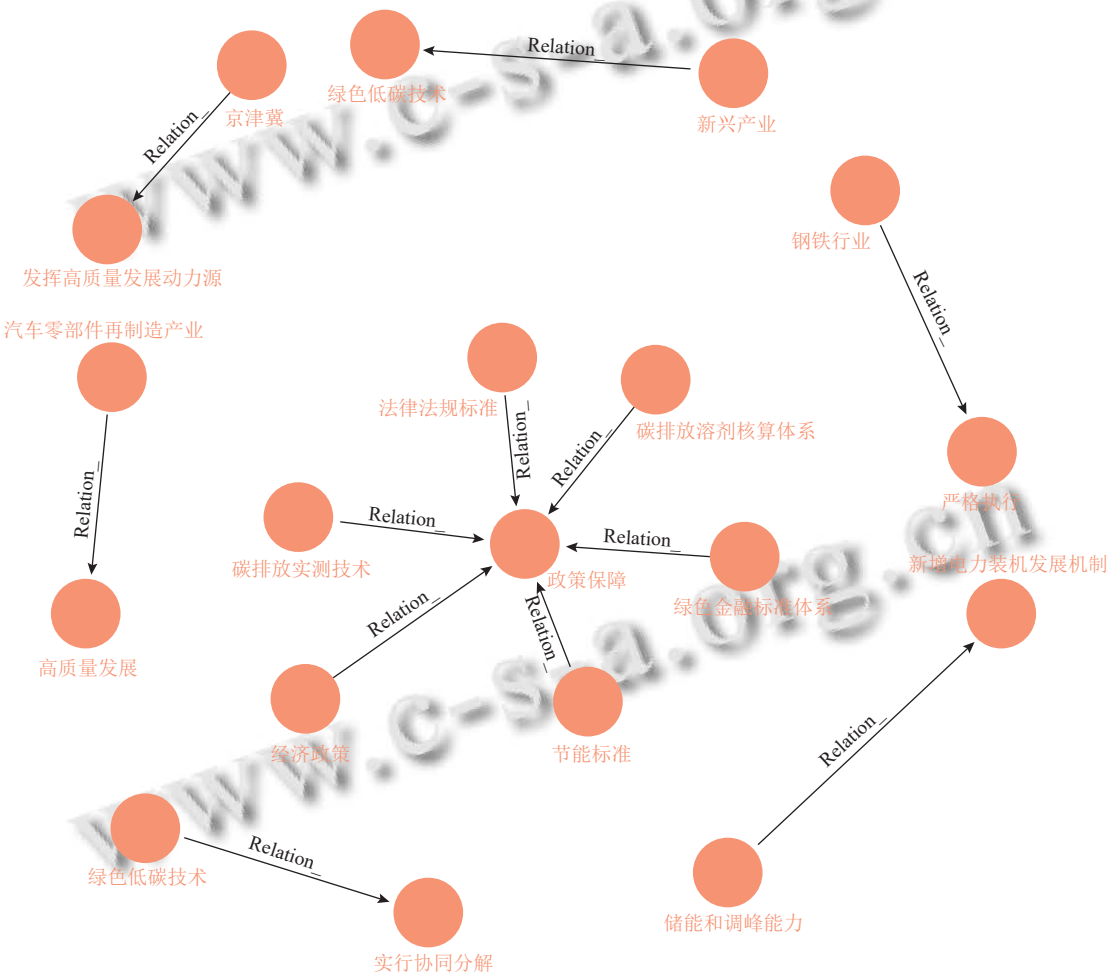

图 3 部分双碳知识图谱

动态合并阶段是针对递归检索生成的碎片化节点集合, 设计了两阶段合并策略以实现结果的层次化聚合与序列修复. 在父节点合并阶段, 合并节点操作基于父节点与子节点的数量比率. 对于每个父节点, 计算其当前子节点数量与总子节点数量的比率 r:

$$r = \frac{|parent\_cur\_children|}{|parent\_child\_nodes|} \tag{1}$$

其中, $|parent_cur_children|$为当前检索到的子节点数量, $|parent_child_nodes|$为父节点的总子节点数量. 当比率 r 超过简单比例阈值τ时 (设置为 0.5), 会将多个

子节点合并为父节点. 合并后, 父节点的得分 $score_{\text{parent}}$ 通过对这些子节点的得分求平均值得到, 公式如下:

$$score_{\text{parent}} = \frac{\sum_{i=1}^{n} score_{child_i}}{n} \tag{2}$$

其中, $score_{child_i}$ 表示第 i 个子节点的得分, n 为子节点的数量.

在节点填充阶段, 在结果集中, 检查相邻节点之间是否存在缺失的中间节点. 若存在, 即当前节点的下一个节点信息与下一个节点的前一个节点信息匹配时, 会将中间节点补充到结果集中. 中间节点的得分 $score_{\text{mid}}$ 通过取相邻节点得分的平均值来计算, 公式如下:

$$score_{\text{mid}} = \frac{score_{\text{cur}} + score_{\text{next}}}{2} \tag{3}$$

其中, $score_{\text{cur}}$ 为当前节点的得分, $score_{\text{next}}$ 为下一个节点的得分.

合并与填充操作迭代执行, 直至节点集合不再变化, 确保策略充分应用, 避免局部最优.

该方法具有层次化适应性、数据驱动的合并决策和语义完整性保障等关键特性. 它支持任意深度的组件嵌套, 能自动解析索引节点的层次关系, 实现“按需展开”的深度检索; 基于子节点覆盖度动态选择合并粒度, 平衡信息细节与抽象层次, 同时保留节点分数的统计特性; 节点填充机制修复内容缺失, 去重与合并策略避免重复信息干扰, 提升后续任务处理效率.

整体检索流程分为递归检索、动态合并和排序输出这 3 个阶段. 先从根组件出发生成包含层次化节点的初始集合, 再通过迭代执行填充与合并策略优化结果, 最后按相似度分数降序排列节点, 为下游任务提供有序的输入序列. 此方法通过灵活组合不同检索工具, 在保持信息完整性的同时提升检索效率, 其生成的包含层次关系与序列依赖的节点集合, 天然支持需要结构化输入的任务, 为复杂知识环境下的高效信息获取提供了通用解决方案.

3.2.2 知识图谱检索

本文知识图谱检索体系的核心预处理阶段整合了关键词提取与同义词扩展策略, 融合自然语言处理与图查询技术. 首先, 通过 LLM 结合专业设计的提示词模板, 对用户查询文本进行深度语义解析, 精准识别出具有核心指向意义的实体词汇作为关键词. 随后, 针对提取的关键词, LLM 进一步挖掘其潜在的同义词或相近表述, 通过语义拓展将更多语义关联词汇纳入检索范畴. 这一过程不仅确保了对用户查询核心信息的准确捕捉, 还通过词汇边界的扩展, 有效覆盖了因语言表达多样性可能导致的信息盲区, 为后续检索奠定了更全面的语义基础.

在完成主题词的提取与语义扩展后, 检索器启动图查询策略. 知识图谱以节点代表实体, 连线表征关系的网络结构存在, 检索器以预处理阶段形成的关键词及其扩展词汇为导航坐标, 在知识图谱中展开检索. 通过限定检索范围为与关键词直接关联的一层关系网络, 系统聚焦核心关联信息, 避免冗余搜索, 高效定位与用户查询紧密相关的知识片段.

尤为重要的是, 该体系融合了自然语言处理与图查询技术, LLM 将用户的自然语言查询转化为图数据库可识别的专业指令, 深度挖掘知识图谱中的潜在关联信息, 并与基于关键词及其扩展词汇的检索结果有机整合. 这种多维度的信息汇聚机制, 确保最终检索结果既精准贴合用户查询意图, 又能通过语义扩展和图结构关联实现对知识的全面覆盖, 为复杂知识检索场景提供了高效且可靠的解决方案.

4 实验与分析

4.1 数据集

本文利用 93 个多地相关政府机构或国家级部门发布的碳政策文件作为实验数据集. 文件内容详细规划了当地工业领域碳达峰的总体要求、重点任务和保障措施, 其中对产业结构调整、节能降碳、绿色制造等方面提出了明确目标与实施路径, 为双碳知识库提供了重要的政策依据与实践指导方向.

本文基于父节点文本, 通过 LLM 生成 233 个简答题, 并利用这些简答题开展检索效果测评.

4.2 实验环境

本文使用 Llamaindex 框架作为基础, 并对本文所使用的源文件进行优化. 大模型使用智谱 GLM-4-Plus. 文本嵌入模型使用 jina-embeddings-v3.

4.3 指标

本系统将其细分为检索评估和响应评估两个部分, 其中响应评估引入 LLM 作为评估专家来对响应-上下文-查询相关性、响应-查询相关性、上下文-查询相关性、忠诚性进行评估, 表 4 为各响应评估指标的解释, 返回结果均由 LLM 生成.

表 4 响应指标解释

指标	含义	返回结果
响应-上下文-查询相关性 (*Resp-Ctxt-Qry-Rel*)	响应和上下文是否与查询相关	是/否
响应-查询相关性 (*Resp-Qry-Rel*)	评估生成的答案与用户查询是否相关	0–1之间的分数以及解释分数的反馈
上下文-查询相关性 (*Ctxt-Qry-Rel*)	评估检索到的上下文与用户查询是否相关	0–1之间的分数以及解释分数的反馈
忠诚性 (*Loy*)	来测量响应是否忠于上下文	是/否

为进一步量化这 4 个指标的表现, 本文分别引入响应-上下文-查询相关性比例 (*Resp-Ctxt-Qry-Rel ratio*)、忠诚性比例 (*Loy ratio*)、响应-查询相关性平均值 (*Mean Resp-Qry-Rel*)、上下文-查询相关性平均值 (*Mean Ctxt-Qry-Rel*).

Resp-Ctxt-Qry-Rel ratio 是返回结果为“是”的数量占总评估数量的比例, 公式为:

$$Resp\text{-}Ctxt\text{-}Qry\text{-}Rel\ ratio = \frac{n_{\text{yes}}}{N} \tag{4}$$

其中, n_{yes} 代表在“响应-上下文-查询相关性”评估中, 返回结果为“是”的数量; N 表示该指标的总评估数量. 该指标取值区间为[0, 1], 越接近 1, 表示相关情况占比越高, 系统相关性匹配能力越强.

Loy ratio 即返回“是”的数量与总评估数的比值, 公式为:

$$Loy\ ratio = \frac{n_{\text{yes}}}{N} \tag{5}$$

其中, n_{yes} 是在“忠诚性”评估中, 返回结果为“是”的数量, N 为“忠诚性”指标的总评估数量. 取值在[0, 1]区间, 数值越接近 1, 响应忠实于上下文的情况越多, 系统可靠性越高.

Mean Resp-Qry Rel 是多个此类分数的算术平均, 公式为:

$$Mean\ Resp\text{-}Qry\text{-}Rel = \frac{\sum_{i=1}^{n} Resp\text{-}Qry\text{-}Rel_i}{n} \tag{6}$$

其中, $Resp\text{-}Qry\text{-}Rel_i$ 表示第 i 次“响应-查询相关性”评估中得到的分数, n 为“响应-查询相关性”评估的总次数, 其值在[0, 1]区间, 越高表明答案整体与查询相关性越好, 对查询意图回应越准确.

Mean Ctxt-Qry-Rel 由多次评估分数平均得到, 公式为:

$$Mean\ Ctxt\text{-}Qry\text{-}Rel = \frac{\sum_{i=1}^{n} Ctxt\text{-}Qry\text{-}Rel_i}{n} \tag{7}$$

$Ctxt\text{-}Qry\text{-}Rel_i$ 是第 i 次“上下文-查询相关性”评估的分数, n 为“上下文-查询相关性”评估总次数. 取值在[0, 1]区间, 越趋近 1, 检索的上下文总体与查询相关性越高, 为响应提供更有效的信息基础. 这些指标从不同维度量化系统性能, 为评估和优化提供依据.

检索评估利用算法来进行评估, 在自然语言处理任务中, 精确率 (*Precision*)、召回率 (*Recall*) 和 *F*1 分数 (*F*1 score) 是评估检索性能的核心指标, 三者从不同维度来衡量检索结果, 下面将详细介绍各指标.

Precision 衡量模型预测为正类的样本中实际为正类的比例, 反映模型预测结果的“可靠性”. 其数学定义为:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

其中, *TP* (true positive) 为正确预测的正类样本数; *FP* (false positive) 为错误预测的正类样本数 (实际为负类, 但被误判为正类).

Recall 衡量模型对正类样本的覆盖能力, 即实际正类中被正确预测的比例, 定义为:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

其中, *FN* (false negative) 为误判数.

*F*1 分数是 *Precision* 与 *Recall* 的调和平均数, 平衡两者矛盾, 定义为:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

4.4 基线模型

在本次研究中, 为深入探究 Multi-pipeline-based RAG 架构的优势, 精心挑选了以下具有代表性的模型和方法作为对比基线.

首先是基于 BM25 检索的 RAG, 作为经典的信息检索模型, 它依据词频、逆文档频率等特征来衡量查询词与文档的相关性, 在众多文本检索场景中被广泛应用, 具有简单高效、易于理解与实现的特点, 能为评估改进的迭代检索在基础检索性能上的提升提供直观参照.

基于向量检索 (vector-based retrieval) 的 RAG 近年来崭露头角, 它将文本转化为低维向量表示, 通过计算向量间的相似度 (如余弦相似度等) 来检索相关文档. 这种检索方式能够捕捉文本的语义信息, 尤其适用于语义理解要求较高的场景, 像知识图谱问答、智能客服领域等. 以其作为基线, 可检验改进的迭代检索在语义挖掘深度与广度上是否更胜一筹, 即面对复杂语义关联的查询需求时, 能否超越向量检索单纯依靠向量空间映射的方式, 进一步优化检索结果的相关性排序.

而传统递归检索 (recursive retrieval, RR) 的 RAG, 遵循固定的迭代流程, 在每一轮迭代中依据上一轮检索结果调整检索策略, 逐步逼近最优解. 它虽然初步引入了动态调整检索方向的思路, 但存在迭代终止条件不够灵活、对中间结果利用不充分等问题. 将其作为对比对象, 能够凸显本文提出的改进的迭代检索在优化迭代机制、增强反馈调节精准度及避免无效迭代等方面所做的努力, 清晰展现该方法如何克服传统迭代检索的固有短板, 从而在复杂信息环境下达成更高效、精准的检索目标, 满足用户日益增长的复杂信息需求.

KG-向量混合检索增强生成 (KG-vector hybrid RAG)[18], 该方法融合知识图谱结构化推理与向量语义检索. 在双碳领域, 调整知识图谱元数据为“政策文件-目标-措施”结构.

基于 LoRA 微调的领域大模型, 该方法采用领域数据对 GLM-4-Plus 大语言模型进行参数优化. 该方法可能受限于训练数据的覆盖范围与模型的“知识遗忘”问题.

4.5 实验结果

比较的策略被分为 6 组: 第 1 组基于 BM25 (BM25-based RAG)、第 2 组基于单一向量 (vector-based RAG)、第 3 组基于单一递归 (recursive-based RAG)、第 4 组为 LoRA 微调、第 5 组为 KG-向量混合检索增强生成 (KG-vector-based RAG)、第 6 组为基于多管道 (Multi-pipeline-based RAG). 表 5 展示了不同策略的评估结果. 评估指标涵盖精确率、召回率、 *F*1 分数、 响应-上下文-查询相关性比例、 响应-查询相关性平均值、上下文-查询相关性平均值以及忠诚性比例.

表 5 基准测试结果

Strategy	*Precision*	*Recall*	*F*1	*Resp-Ctxt-Qry-Rel ratio*	*Mean Resp-Qry-Rel*	*Mean Ctxt-Qry-Rel*	*Loy ratio*
BM25-based RAG	0.54	0.43	0.44	0.85	1	0.83	0.75
Vector-based RAG	0.47	0.40	0.43	0.9	1	0.68	0.95
Recursive-based RAG	0.74	0.43	0.55	0.9	1	0.83	0.80
LoRA	0.71	0.58	0.64	0.88	1	0.85	0.82
KG-vector hybrid RAG	0.78	0.52	0.63	0.89	1	0.88	0.83
Multi-pipeline-based RAG	0.85	0.55	0.66	0.9	1	0.96	0.85

本文提出的 Multi-pipeline-based RAG 方法在多维度指标上展现出显著优势. 在 *Precision* 上, Multi-pipeline-based RAG (0.85) 较传统基线模型显著提升, 较 BM25-based RAG (0.54) 提升 0.31; 较 Vector-based RAG (0.47) 提升 0.38; 较 Recursive-based RAG (0.74) 提升 0.11. 与纯微调方法 (LoRA, 0.71) 和混合检索方法 (KG-vector hybrid RAG, 0.78) 相比, 分别提升 0.14 和 0.07. 表明如递归合并与图谱引导比单纯依赖参数优化或单一向量匹配更能精准定位领域知识, 减少“幻觉”现象.

Recall 方面, Multi-pipeline-based RAG (0.55) 较 BM25/Recursive-based RAG (0.43) 提升 0.12, 较 Vector-based RAG (0.40) 提升 0.15, 且优于 KG-vector hybrid RAG (0.52). 其优势源于多管道协同——文本分块与递归检索可覆盖多层级知识 (如政策文件的“章节-条款”结构), 而纯微调方法 (0.58) 虽召回率较高, 但受限于训练数据量, 易将不相关知识泛化引入噪声.

*F*1 分数方面, Multi-pipeline-based RAG (0.66) 较 BM25-based RAG (0.44)、Vector-based RAG (0.43)、Recursive-based RAG (0.55) 分别提升 0.22、0.23、0.11; 较 LoRA (0.64) 和 KG-vector hybrid RAG (0.63) 提升 0.02 和 0.03, 体现了检索精确性与全面性的平衡.

在上下文-查询相关性平均值指标方面, Multi-pipeline-based RAG (0.96) 较 BM25-based RAG 和 Recursive-based RAG (0.83) 提升 0.13, 较 Vector-based RAG (0.68) 提升 0.28, 显著优于 KG-vector hybrid RAG (0.88). 这得益于知识图谱的结构化约束. 例如, 检索“山东省钢铁行业碳减排措施”时, 图谱可直接定位 “钢铁行业”实体关联的措施节点, 而传统向量检索易因语义模糊匹配到“建材行业”的无关内容.

忠诚性方面: Multi-pipeline-based RAG (0.85) 较 BM25-based RAG (0.75) 提升 0.1, 较 Recursive-based RAG (0.8) 提升 0.05, 略高于 LoRA (0.82) 和 KG-vector hybrid RAG (0.83). 动态合并策略在此发挥关键作用, 通过递归检索结果的层次化合并 (如父节点与子节点分数平均), 减少了多段落信息融合时的噪声干扰, 确保生成内容严格基于检索到的上下文.

综上, Multi-pipeline-based RAG 通过引入多管道, 层次化递归检索与动态合并策略, 在核心检索指标及相关性、忠诚性维度上较传统方法实现显著提升, 为复杂知识结构下的高效信息获取及下游任务支持提供了更优解决方案.

4.6 错误案例分析

(1) 检索失败导致的回答错误

用户提问“到 2025 年, 吉林省计划创建多少家绿色工厂?”, 若知识库中该政策文件未被正确索引 (如文本分块破坏了“2025 年”与“绿色工厂”的语义关联), 导致检索结果缺失. 通过溯源索引, 发现在文本分块阶段, 由于文本分块粒度不合理, 导致目标文本出现语义断联. 后续可使用专业文本分割模型来对原文件进行初步处理, 以此进一步提高检索准确率.

(2) 不同量级的 LLM 所带来的影响

由于 Multi-pipeline-based RAG 中所使用的知识图谱, 是由 LLM 通过自由提取策略所得. 故在实验过程中使用了 deepseek-llm: 7b、GLM-4-Air 等 LLM 来验证不同量级的 LLM 是否会影响知识图谱质量. 通过环境工程研究团队的帮助, 证实 LLM 的量级与提取质量成正比.

为提高图谱提取质量, 除了提高 LLM 的量级, 设计一套碳领域的知识图谱元模型也是一种较好的方法.

5 消融实验

本文对模型主要进行了两个方面的改进, 首先引入多管道模块帮助模型更全面地理解源文档中所包含的信息. 随后, 使用递归式智能合并检索的方法, 该方法针对递归检索生成的碎片化节点集合, 设计了两阶段合并策略以实现结果的层次化聚合与序列修复. 为了评估 Multi-pipeline-based RAG 不同组成部分对整体性能的影响, 我们将其拆分为以下 3 种方案.

• 单管道: 只使用单一的节点索引, 不附加总结节点同时使用图谱增强递归式智能合并检索.

• 无动态合并: 将图谱增强递归式检索的结果不进行动态合并.

• 单管道加无动态合并: 即是单一的递归检索.

所有消融实验均使用 GLM-4-Plus 作为基础模型. 在第 4.1 节介绍的数据集上进行了测试, 并使用精确率、召回率和 $F1$ 这 3 个指标来评估两个方案的效果, 最终将模型评估结果与 Multi-pipeline-based RAG 进行了比较. 表 6 为无动态合并、单管道和 Multi-pipeline-based RAG 的对比结果, 实验结果表明, 在精确率、召回率和 $F1$ 这 3 个指标上, Multi-pipeline-based RAG 与无动态合并方法对比, 分别提升了 0.33、0.15、0.21; 与单管道方法对比, 分别提升了 0.22、0.10、0.14; 与 Recursive-based RAG 对比, 分别提升了 0.11、0.12、0.11. 由此可以得出, 当动态合并和多管道不结合使用时, 得到的结果反而没有优势, 不如单一的递归检索, 而当动态合并与多管道相结合时, 效果得到了显著的提高.

表 6 消融实验结果

方法	*Precision*	*Recall*	*F*1
无动态合并	0.52	0.40	0.45
单管道	0.63	0.45	0.52
Recursive-based RAG	0.74	0.43	0.55
Multi-pipeline-based RAG	0.85	0.55	0.66

6 总结

尽管大语言模型在通用领域展现出优异性能, 但在处理碳领域专业知识时, 面临事实准确性不足、知识更新滞后及高质量数据集匮乏等挑战. 检索增强生成 (RAG) 技术虽被视为解决上述问题的重要方向, 但其在碳领域应用时, 存在查询意图理解偏差、检索策略针对性不足、结果相关性匹配度低等技术缺陷, 同时缺乏专门针对碳领域的评估数据集, 导致现有方法难以有效满足碳领域复杂任务的需求.

针对上述问题, 本文聚焦于大语言模型在碳领域知识密集型任务中的应用难题, 深入剖析检索增强生成 (RAG) 技术在此领域的短板, 提出并构建了一套完整的解决方案, 通过理论研究与实验验证, 为碳达峰碳中和目标的实现提供了技术路径.

在数据层面, 通过收集整理各地碳政策文件, 构建双碳知识库; 采用文本分块与嵌入技术, 将文档转化为便于处理的向量形式; 利用大语言模型自动生成问答对, 构建双碳问答数据集; 基于智谱 GLM-4-Plus 模型

构建包含 41 000 个知识三元组的双碳知识图谱, 并存储于 Nebula Graph 图数据库.

检索策略上, 提出图谱增强递归式智能合并检索 (GRIM retriever), 融合递归检索与动态合并、知识图谱检索. 递归检索与动态合并通过层次化检索和两阶段合并策略, 实现高效信息获取与结果优化; 知识图谱检索通过关键词提取、同义词扩展及图查询策略, 精准定位相关知识.

评估体系中, 在传统精确率、召回率等指标基础上, 新增响应-上下文-查询相关性、响应-查询相关性、上下文-查询相关性及忠诚性这 4 个评估指标, 利用大语言模型进行综合评估.

实验结果显示, 基于多管道的检索增强生成框架 (Multi-pipeline-based RAG) 在多项指标上显著优于本文中的其他方法.

7 未来工作

未来工作中, 为了进一步提升 Multi-pipeline-based RAG 的性能, 将在以下方面进行全面的研究.

深度融合大模型: 在本次研究与实验过程中, 凡是 LLM 所参与的环节, 如构建问答数据集的质量、结果相关性、忠诚性评估的质量、知识图谱构建的质量等. 深深依赖于所使用的 LLM 质量. 为了构建高质量数据集, 未来可以考虑对 LLM 进行领域微调, 以此来提高 LLM 在以上环节的表现.

文本分块: 在进行反复实验中, 文本分块的大小是影响检索精确率的重要因素之一. 单一地对文本按大小分割会破坏原文本的上下文关系. 在未来可以将此模块替换, 使用专业的文本分割模型来提高数据质量.

碳领域知识图谱构建: 自由提取策略在一定程度上具有局限性, 未来在对源文档进行充分的人工理解后, 需设计一个相对完善的元模型, 再交由大模型, 根据知识图谱元模型来构建一个高质量的知识图谱.

参考文献

1 Zhao WX, Zhou K, Li JY, *et al*. A survey of large language models. arXiv:2303.18223, 2023.

2 Jin M, Shahriar S, Tufano M, *et al*. InferFix: End-to-end program repair with LLMs. Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. San Francisco: ACM, 2023. 1646–1656.

3 Spurlock KD, Acun C, Saka E, *et al*. ChatGPT for conversational recommendation: Refining recommendations by reprompting with feedback. arXiv:2401.03605, 2024.

4 王海涛, 师杨坤. 基于大语言模型的查询扩展方法研究. 计算机技术与发展, 2025, 35(3): 148–155. [doi: 10.20165/j.cnki.ISSN1673-629X.2024.0353]

5 王东清, 芦飞, 张炳会, 等. 大语言模型中提示词工程综述. 计算机系统应用, 2025, 34(1): 1–10. [doi: 10.15888/j.cnki.csa.009782]

6 Cosler M, Hahn C, Mendoza D, *et al*. nl2spec: Interactively translating unstructured natural language to temporal logics with large language models. Proceedings of the 35th International Conference on Computer Aided Verification. Paris: Springer, 2023. 383–396. [doi: 10.1007/978-3-031-37703-7_18]

7 史天运, 李新琴, 代明睿, 等. 铁路自然语言大模型关键技术研究及应用展望. 中国铁路, 2024(7): 7–14. [doi: 10.19549/j.issn.1001-683x.2024.03.08.001]

8 Castro Nascimento CM, Pimentel AS. Do large language models understand chemistry? A conversation with ChatGPT. Journal of Chemical Information and Modeling, 2023, 63(6): 1649–1655. [doi: 10.1021/acs.jcim.3c00285]

9 Watari T, Takagi S, Sakaguchi K, *et al*. Performance comparison of ChatGPT-4 and Japanese medical residents in the general medicine in-training examination: Comparison study. JMIR Medical Education, 2023, 9: e52202. [doi: 10.2196/52202]

10 Edge D, Trinh H, Cheng N, *et al*. From local to global: A graph RAG approach to query-focused summarization. arXiv:2404.16130, 2024.

11 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.

12 Mallen A, Asai A, Zhong V, *et al*. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: ACL, 2023. 9802–9822.

13 Chen DQ, Fisch A, Weston J, *et al*. Reading Wikipedia to answer open-domain questions. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017. 1870–1879.

14 Zeng SL, Zhang JK, He PF, *et al*. The good and the bad:

Exploring privacy issues in retrieval-augmented generation (RAG). Proceedings of the 2024 Findings of the Association for Computational Linguistics. Bangkok: ACL, 2024. 4505–4524.

15 Zhao PH, Zhang HL, Yu QH, *et al*. Retrieval-augmented generation for AI-generated content: A survey. arXiv:2402.19473, 2024.

16 Yang H, Li SL, Gonçalves T. Enhancing biomedical question answering with large language models. Information, 2024, 15(8): 494. [doi: 10.3390/info15080494]

17 齐俊, 曲睿婷, 教传铭, 等. 基于知识图谱增强大语言模型双碳领域服务. 计算机与现代化, 2024(9): 8–14. [doi: 10.3969/j.issn.1006-2475.2024.09.002]

18 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: 103212. [doi: 10.1016/j.aei.2025.103212]

19 鞠炜刚, 汪鹏, 王佳. 基于大语言模型和 RAG 的持续交付智能问答系统. 计算机技术与发展, 2025, 35(2): 107–114. [doi: 10.20165/j.cnki.ISSN1673-629X.2024.0347]

20 Hu EJ, Shen YL, Wallis P, *et al*. LoRA: Low-rank adaptation of large language models. Proceedings of the 10th International Conference on Learning Representations. OpenReview.net, 2022.

21 Singhal A. Introducing the knowledge graph: Things, not strings. The Official Google Blog, 2012, 5(16): 3–8.

22 张吉祥, 张祥森, 武长旭, 等. 知识图谱构建技术综述. 计算机工程, 2022, 48(3): 23–37.

23 刘世侠, 李卫军, 刘雪洋, 等. 基于强化学习的知识图谱推理研究综述. 计算机应用研究, 2024, 41(9): 2561–2572.

24 Berners-Lee T, Hendler J, Lassila O. The semantic Web. Scientific American, 2001, 284(5): 34–43. [doi: 10.1038/scientificamerican0501-34]

25 Pan SR, Luo LH, Wang YF, *et al*. Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(7): 3580–3599. [doi: 10.1109/TKDE.2024.3352100]

26 Freire SK, Wang C, Niforatos E. Chatbots in knowledge-intensive contexts: Comparing intent and LLM-based systems. arXiv:2402.04955, 2024.

27 Huang ZH, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991, 2015.

28 王良萸. 面向碳交易领域的知识图谱构建方法. 计算机与现代化, 2018(8): 114–119. [doi: 10.3969/j.issn.1006-2475.2018.08.021]

29 Peng YF, Guo JL, Liu YH. Comparative analysis of China's pilot carbon market based on knowledge graph. Proceedings of the 3rd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS). Shenyang: IEEE, 2023. 521–525.

30 田小贵, 许建新, 张振明. 融合知识图谱与大模型的零件工艺设计知识问答. 计算机技术与发展, 2025, 35(8):101–109. [doi: 10.20165/j.cnki.ISSN1673-629X.2025.0056]

31 Sturua S, Mohr I, Akram MK, *et al*. jina-embeddings-v3: Multilingual embeddings with task LoRA. arXiv:2409.10173, 2024.

(校对责编: 王欣欣)