文献综述

一、研究背景概述

当前大语言模型（LLM）在问答任务中仍面临“幻觉”频发、知识更新滞后、领域专业性不足等核心问题。知识图谱（KG）因其结构化、可解释、可追溯的优势，成为解决上述瓶颈的关键补充手段。结合检索增强生成（RAG）技术，研究者提出了多种创新架构，实现从静态问答向动态推理、从通用模型向领域专精的演进。

二、技术方法详述

1. TCMLCM：KG2TRAG 方法（中医肺癌问答

算法核心流程

KG2TRAG 把“结构化三元组”→“自然语言句子”→“再检索”→“再生成”做成一个双层检索-生成闭环。

离线阶段用 ChatGPT 把 KG 中所有 `(s, p, o)` 三元组线性化成一句通顺的中文描述，例如(肺癌，证候，阴虚火旺) → “肺癌患者常见证候为阴虚火旺”。线性化模板由少量 hand-crafted prompt 控制，保证医学术语不被扭曲。

在线问答阶段

第 1 轮：用 BGE 向量模型做稠密检索，从“线性化句子池”里召回 Top-k 句。

第 2 轮：把召回的句子重新拼回三元组，再与原始 KG 对齐，形成“证据子图”。

第 3 轮：把子图再次线性化，和原问题一起喂给已经 LoRA 微调过的 ChatGLM2-6B（LoRA rank=64，α=32，8-bit QLoRA 量化；batch=64，步数 3000，学习率 2e-3），生成最终答案。 算法亮点

用“线性化-反解析”方式解决了 KG 与 LLM 之间的语义鸿沟。

双层检索：先句子级粗召回，再子图级精对齐，减少幻觉。

2. KG-IRAG：迭代式 RAG 框架（时间-空间推理）

算法核心流程

KG-IRAG 通过“LLM1 规划 + LLM2 验证”两轮对话完成迭代检索。

LLM1 规划器输入：自然语言问题（如“今天 14-16 点去悉尼歌剧院，会下雨吗？”）。输出：一个逻辑形式(location=Sydney Opera House, time\_range=[14:00,16:00], predicate=rain)。

迭代检索循环（LLM2 验证器）Step-0：用规划器给出的 (loc, time) 在 KG 中做一次 hop 查询，取出天气三元组。Step-i：LLM2 判断当前三元组是否足以回答问题；若不足，则生成新的 (loc’, time’) 并继续查询。终止条件：LLM2 输出 `SUFFICIENT` 或达到最大迭代次数。

图查询算法底层使用Cypher 查询 + BFS 子图扩展（m-hop=3）实现快速时间轴遍历。

算法亮点

将“时间-空间”约束显式写进逻辑形式，避免 LLM 直接做数值推理导致的幻觉。

通过“停-走”机制自动决定何时停止检索，减少冗余。

3. RAG-KG-IL：多智能体增量学习框架

算法核心流程

系统由 4 个异步 Agent 组成，各自维护独立内存，通过消息总线通信。

1. Query Agent

把用户问题转成向量 + 关键词双通道表示。

2. Retriever Agent

先用向量召回 Top-N 文档块；

再用关键词过滤掉“硬负例”，得到种子块。

3. KG Agent

对种子块做实体-关系联合抽取（使用 Prompt-based LLM），生成新三元组；

执行RDFLib 推理规则（如传递性、对称性）补全隐含关系；

将新三元组增量写入 Neo4j，实现 KG 热更新。

4. IL Agent

采用LoRA-Switch机制：

当新三元组数量 < 阈值 T 时，仅更新 LoRA adapter；

当 > T 时，触发一次轻量级全参数微调（QLoRA，rank=16）。

算法亮点

用“事件驱动”方式实现 KG-LLM 同步，避免全量重训。

推理阶段可输出因果路径图，提升可解释性。

4. Adobe KG-RAG：企业文档问答系统

算法核心流程

1. 离线构建 KG

- Semantic Chunking：按句子粒度切分文档，块长 128 token，overlap 32 token。

- Entity-Relation Extraction：用 LLM 抽取 `(entity, relation, entity)`，每条给 0-1 置信度；低于 0.6 丢弃。

- Deduplication：SBERT 语义相似度 > 0.85 的实体合并，保留最短字符串作为 canonical name。

- Provenance：每个三元组存回原始 URL，支持用户溯源。

2. 在线检索

- 双塔检索：问题向量 vs 块向量（BERT-base 768-dim）；

- Top-1 块 直接附在 prompt 后，无需 rerank。

3. LLM-as-Judge 评估

- 用 GPT-4-mini 做 3 级评分：Irrelevant / Moderate / Fully Relevant；

- 人工与机器标注一致性 85%，KDE 图显示保守偏置（更倾向打 Moderate）。

算法亮点

全链路置信度控制，保证企业级可靠性。

Top-1 策略降低延迟，适合高并发客服场景。

5. 基于 LLM 的图数据库专利 KG 构建

算法核心流程

1. Prompt 工程

- 采用 GLM-4-0520 + Batch API，设计 5 类提示：角色、任务、上下文、示例、输出格式。

- 示例提示（few-shot）仅给 2 句专利摘要与期望 JSON，即可稳定抽取 7 类实体、6 类关系。

2. 图扩展与可视化

- 用 TextRank 提取领域关键词，权重映射为节点大小；

- 用 PageRank 计算节点影响力，知识图谱节点得分最高（50.96）。

3. 时间演化分析

- 将 2007-2024 年专利按“早期-中期-后期”切分，Cypher 查询每个阶段的技术关键词频率，绘制热力图。

算法亮点

零标注知识抽取，仅用 prompt 即可达到 93%+ 的实体 F1。

可视化脚本自动生成技术演化路径，适合情报人员快速洞察。

6. 软件测试用例复用：语义匹配 + 知识图谱

算法核心流程

1. 图谱构建

本体：TestCase → TestStep → ExpectedResult → BugReport，四元关系。

使用 BERT-whitening 获得 512-dim 句向量，存 Neo4j 作为节点属性。

2. 语义匹配

用户输入用例描述 q，计算与图谱中所有用例 d\_i 的 加权余弦相似度：

3. Top-k 重排

先 BM25 粗召回 100 条，再用向量精排取 Top-5。

4. 版本控制

每次复用自动创建新节点并指向父节点，形成可追溯的用例演化链。

算法亮点

加权余弦相似度 + 版本链，兼顾精度与可追溯性。

检索时间平均缩短 30%，召回率提升 8.32%。

三、结论与建议

本综述基于 6 篇最新论文，系统梳理了知识图谱与 RAG 技术在问答系统中的演进路径。研究表明，“KG+RAG+LLM” 是提升问答系统准确性、可解释性与动态性的有效路径，未来应重点关注领域 KG 的自动化构建与持续更新；多模态数据融合与轻量化部署；高可靠性场景下的可解释性增强。