基于图检索增强生成(GraphRAG)技术的

审计定性依据查询系统研究

【摘 要】AIGC技术的迅速发展为内部审计效率和准确性的提升带来了新的机遇。本文针对传统RAG方法构建的审计定性依据查询方案存在的检索内容不完整、信息不够丰富等问题，提出了一种基于图检索增强生成（GraphRAG）技术的审计定性依据查询系统。该系统通过Neo4j图数据库构建审计知识图谱，并结合GraphRAG和LLM进行信息检索和内容生成，实现了基于源引用的查询响应和清晰的依据条款解释。实验结果表明，GraphRAG技术能够显著提高LLM在审计定性依据检索场景的可靠性，提升查询的准确性和完整性。本研究为内部审计工作提供了实践方案，并为推进AIGC技术在审计领域的应用提供了新的思路。

**【关键词】**审计定性依据 图检索增强生成 大语言模型 智能检索系统

一、引言

内部审计在企业管理中扮演着至关重要的角色，它不仅有助于发现和预防风险，还能提高企业的运营效率和合规性。内部审计作为企业内部监督控制的重要抓手，必须时刻提升审计履职能力，适配数字化变革中出现的审计新需求。在网络强国、数字中国、大数据战略等国家政策指导下，中央企业正加快数字化转型，消除“数据孤岛”和“数据烟囱”，开展大数据与人工智能、物联网等技术融合应用。这一变革使信息载体从纸质变为电子文件，数据量剧增，数据关系复杂且标准不一，同时信息安全风险加大，这对内部审计方法和效率提出了新要求和新挑战。在推动内部审计工作数字化、智能化发展过程中，强化审计定性依据的规范性与适用性，是确保审计监督质量和效果的重要环节。作为审计工作的基本遵循，审计定性依据的科学分类和有效运用是提升审计效率和规范化水平的关键，而正确选择审计定性依据更是内部审计机构有效开展审计工作的基础。审计定性依据通常来源于具有法律效力的国家法律法规、监管规定，以及行业准则和企业内部规章制度等规范性文件（金昌铉等，2023），传统的依据查询主要依赖于人工翻阅资料或基于关键词的文本检索，存在效率低下、结果不准确等问题，难以满足现代内部审计的需求。为解决这一问题，学术界和业界开始探索将人工智能技术应用于内部审计领域。已有学者对AIGC（Artificial Intelligence Generated Content，AIGC）技术在内部审计工作中的应用价值进行了研究，尤其是在提高审计效率和准确性方面取得了一定进展。其中，基于检索增强生成（Retrieval-Augmented Generation, RAG）技术的方法在审计知识智能检索方面显示出潜力。然而，现有研究主要集中在传统RAG方法的应用，对于复杂审计知识结构和多层次查询需求的处理仍存在局限性。鉴于此，本文对业界现有审计知识智能检索方案进行优化创新，提出通过图检索增强生成（GraphRAG）技术构建的审计定性依据查询系统，以提高审计定性依据查询的便捷性和准确性，帮助审计人员更好地理解审计问题及其定性依据，从而更好地服务于审计监督职能的有效履行。

二、研究概述

（一）业界研究概述

AIGC技术的发展，在内部审计领域掀起了一场人工智能革命。易冰心（2023），刘锦（2024）等人研究了人工智能在内部审计工作中的应用，提出了AIGC技术在辅助编制审计计划、审计数据分析、自动化审计流程、审计知识检索等领域的应用价值，并基于AIGC技术探索了审计知识问答助手的构建和运用。例如，通过AIGC技术自动化数据分析和模式识别，帮助审计人员快速识别潜在的风险和问题；使用大语言模型(Large Language Model, LLM)根据预设的规则和标准自动生成审计报告，对关键信息内容进行辅助校对等。

在AIGC技术企业应用落地方面，目前研究较广泛的领域是利用检索增强生成（Retrieval-Augmented Generation, RAG）技术对企业沉淀数据、领域知识进行智能检索及智能问答。RAG技术由Lewis等人（2021）提出，是一种利用特定和私有数据集回答用户查询的技术，无需对大语言模型进行进一步训练。该技术通过结合向量检索和大语言模型，使系统能够迅速地从海量数据、知识库中提取相关信息，并在此基础上生成连贯且有原始数据支撑的回答。即使利用RAG技术对传统搜索系统进行简单优化，也能使得企业知识的利用效率和问答准确性得到显著提升。

在内部审计领域，部分研究学者通过FastGPT、Chato等AI开发框架对审计知识库智能检索进行了研究和实践（刘锦，2024；陈亚盛等，2024），利用其RAG开发引擎实现了对审计报告、法律法规、审计定性依据等知识问答助手的构建。以审计定性依据库为例，这个过程通常分为3个阶段：首先，将审计定性依据指引条目汇编为审计指南或参考文档，通过对该文档中进行分块（Chunking），将长文本分割成较小的块（Chunk）以适应嵌入模型（Embedding Model）的上下文长度限制；然后，将每个块通过嵌入模型转换为用于检索的嵌入向量（Text Embedding），并在专有向量数据库或支持向量扩展的常规数据库中建立向量索引；最后，在问答阶段，通过检索与查询语句的向量表示最相关的块召回原始文本片段，并将它们作为上下文输入给LLM，以生成对用户查询的回答。

基于长文本分块并进行文本块召回的RAG技术，在智能知识检索方案中以其实现简便、部署灵活和效果显著等优势，在内部审计领域展现出显著的应用潜力。然而，随着审计工作复杂性的提升以及对知识检索精度要求的日益提高，传统的RAG方法在实际应用中逐渐暴露出一些局限性。特别是在审计问题定性过程中，往往需要准确的依据来源以及精确的措辞表达，因此对检索结果的准确度提出了更高的要求。传统的RAG方法在这一方面尚存在差距，难以满足高精度审计定性依据检索的需求。

（二）传统RAG方案的局限性

1. 关联信息分割，导致答案质量下降

为了适应嵌入模型的上下文长度限制，将大量有内部结构层次的问题定性依据条目分割成固定长度的片段可能导致相关内容断开连接，从而产生不完整的答案。例如，一个在开头描述问题、在结尾提供依据条款的问题定性依据条目可能在文本分割过程中被拆分，导致定性依据的关键部分被遗漏。尽管Chunk Overlap（文本块重叠）、Recursive Character Text Splitter（递归字符文本分割器）等技术的应用能够保留更多的文本块上下文信息、保证单句内容的完整性[引用]，但受限于chunk size的大小，传统RAG方法的文本分割模式仍无法避免破坏审计定性依据条目的结构信息，导致召回内容的缺失。

2. 未构建实体关系，难以实现多条查询

像审计定性依据指引这样的参考文档具有固有的信息结构和关联关系，包含诸如“[审计问题C](定性依据是)[文件条款B]”“[文件条款B](属于)[依据文件A]”“[审计内容D](包含)[审计问题C]”等引用，将文档拆分为固定长度文本块的常规文本分块方法未考虑不同实体之间的关联，导致实体关系信息丢失，进而无法应对复杂的多跳查询(Multi-hop Query)。例如，对于“与工程项目应招未招属于同一类审计内容的问题还有哪些？”这样的查询，由于缺乏实体关系的建模，传统RAG方法即使能够从文本块中检索到部分信息，也无法进一步找到与其他相关审计问题的关联，导致检索结果局限于单个文本块的内容，无法有效进行多跳推理。尽管在传统RAG框架下可以通过多轮查询设计、链式推理等方法部分实现多跳查询，但这些方案大幅增加了系统复杂度。每次查询的结果都需要进一步处理和重新检索，且需要明确设计如何在多轮交互中处理上下文和关联关系，进而影响系统的维护、扩展以及查询效率和准确性。

3. 缺乏多路径查询和复杂问题路由能力

传统RAG方案的检索机制通常依赖于单一查询路径的处理模式，即通过用户查询或关键语句直接进行向量化，匹配最相关的信息片段作为LLM上下文。这种模式限制了系统对多样化问题的检索能力，尤其是在面对复杂的用户查询时，传统RAG难以灵活地选择适合的查询路径。例如，审计人员可能会提出不同类型的问题，如“某审计问题的定性依据是什么？”或“某文件针对特定事项有何要求？”，这些问题需要系统根据不同的语义需求选择不同的查询路径。而传统RAG方案缺乏针对复杂问题的路由能力，无法通过多跳查询路径或灵活的函数调用来处理多样化的查询方式，导致其在应对复杂查询时显得力不从心。

三、本文研究方案探讨

（一）GraphRAG技术介绍

由于传统RAG技术原本就是被设计用于在特定文本区域内找到必要答案，其难以从包含共同属性的关联信息片段中挖掘出关联信息，在需要对大型数据集或长文档中的总结性语义概念进行理解的任务表现也不尽如人意。为了解决这些局限性，Hu等人（2024）提出了GraphRAG，它是一个利用外部结构化知识图谱来提升语言模型（LM）的上下文理解能力并生成更丰富响应的技术框架，其目标是从数据库中检索最相关的知识，从而增强下游任务的答案质量。从广义上讲，GraphRAG可以被视为RAG的一个新的分支，它从图数据库中检索相关的关系知识，而不是从文本语料库中检索。GraphRAG方法通过将知识图谱与查询相结合，能够更好地识别和连接相关信息片段，将结构信息作为超越文本本身的额外知识，弥补了传统RAG技术在处理复杂语义关系时的不足，展现出更智能和更强大的综合检索能力。

尽管当前智能知识库问答领域涉及到众多RAG技术研究与探索，但对于如何将GraphRAG这一新技术分支运用到审计知识检索领域未曾被广泛讨论。

（二）Neo4j图数据库介绍

Neo4j是一种高性能的图数据库，采用属性图模型来存储和管理数据。它的核心构建元素包括节点、关系和属性，其中节点代表实体，关系则表示节点之间的连接，属性则是描述节点和关系的键值对。与传统的关系型数据库不同，Neo4j不使用表格结构，而是通过图形结构来表达数据之间的复杂关系，这使得它在处理高度关联的数据时表现出色。就数据检索而言，Neo4j支持Cypher查询语言，能够高效地执行复杂的查询操作，如最短路径搜索和模式匹配，广泛应用于社交网络、推荐系统及知识图谱构建等领域。其图形处理引擎（GPE）确保了数据的快速访问和处理能力，使得Neo4j成为现代数据驱动应用的重要工具。

本文将基于Neo4j图数据库构建审计定性依据知识图谱，结合Langchain框架来构建GraphRAG系统。通过将审计问题描述、文件条款内容、依据文件名称等文本转换为向量数组并存储在Neo4j中，我们能够实现高效的向量索引和相似性检索。在查询过程中，系统将利用大型语言模型的函数调用能力来解析用户查询，并通过Cypher查询语言在Neo4j中执行多跳查询，以获取最相关的审计定性依据信息。最终，系统将结合检索到的结构化数据和用户查询，生成结构化且信息丰富的参考答案。

（三）GraphRAG方案总体思路

为克服传统RAG方案在审计定性依据检索系统中的局限性，本文提出了一种基于GraphRAG技术的改进方案，主要从以下三个方面着手：

1. 结构化知识构建

利用图数据库技术构建审计定性依据指引的知识图谱，将审计问题、文件条款、依据文件等实体及其关系进行结构化存储。这种方法能够保留问题定性依据条目各部分的逻辑连贯性，有效解决关联信息分割的问题，确保提供完整且高质量的回答。

2. 多实体语义嵌入

采用多实体语义向量嵌入方法，对审计问题、文件条款等多个实体的关键属性文本进行向量化处理，显著提升系统对复杂语义概念的捕捉和定位能力。通过结合图数据库的关联节点检索机制，这种方法使得系统能够实现审计问题和依据文件等概念之间多层次、复杂关系的查询。

3. 多路径检索优化

设计多路径召回方案，实现复杂查询问题的高效准确响应。这种方法能够从多个路径检索相关信息，大大降低信息遗漏的可能性。通过LLM函数调用能力来识别查询实体目标，匹配到相应实体的关联节点，有效提高语义理解的准确性。这种优化使系统能够准确解析用户的查询意图，根据问题类型动态选择合适的路径进行查询，提供更加灵活和精准的答案。

（四）关键技术原理阐述

1. 文本块大小的选择

在实施RAG技术时，文本块大小(chunk size)的范围对检索准确度有重要影响。根据Zhang等人(2022)的研究，适当的文本块大小可以显著提高检索的准确性和效率。太大的文本块可能包含过多无关信息，而太小的文本块则可能丢失重要的上下文。因此，在处理审计知识时，需要仔细考虑文本块的划分策略，以平衡信息完整性和检索精度。

2. LLM函数调用及结构化输出

LLM函数调用是GraphRAG方案中连接大语言模型与知识图谱的关键桥梁。它允许我们利用LLM的强大语言理解能力来解析用户查询，并将其转化为对知识图谱的精确操作。通过设计一系列特定功能的函数，如实体识别、关系抽取、查询生成等。当用户提出一个复杂的审计问题时，系统首先调用LLM来理解问题的核心意图，然后通过预定义的函数接口将这种理解转化为结构化的查询指令。

例如，对于“某审计内容涉及哪些法律法规?”这样的问题，LLM可能会调用实体识别函数来定位"审计内容"和"依据文件"这两个关键实体，再调用关系查询函数来寻找属于“法律法规”类型的依据文件。这种方法大大提高了查询的准确性和效率。在得到查询结果后，LLM还能发挥其强大的自然语言生成能力，将图数据库返回的结构化信息转化为流畅、易懂的文本描述。这不仅包括直接的事实陈述，还可能涉及对结果的解释、总结甚至进一步的分析建议。通过这种方式，系统能够为审计人员提供既准确又易于理解的信息支持。

（加图示）

3. 单跳查询和多跳查询

单跳查询(Single-hop Query)和多跳查询(Multi-hop Query)是检索技术中的两个重要概念，它们代表了在数据库中进行查询和推理的不同复杂度。

单跳查询是最基本的查询方式，通常只涉及一个步骤或一次“跳跃”来获取所需信息。例如，直接查询某个实体的特定属性就是一个单跳查询。多跳查询则是一种更复杂的查询方式，需要通过多个步骤或跳跃来获取最终答案。每个跳跃通常涉及从一个实体到另一个实体的连接，这些连接通常是通过关系或属性来实现的。多跳查询能够处理更复杂的问题，需要在多个相关实体之间进行推理。

在审计知识检索中，两种查询方式都有其应用场景：单跳查询可用于快速获取特定审计项目的基本信息，如项目编号、负责人等。而多跳查询能够帮助审计人员更好地理解复杂的审计问题，例如追踪特定审计发现的法规依据，或者分析不同审计发现之间的潜在关联。

四、系统构建与开发

（一）实验环境与工具：

使用Neo4j图数据库、LLM和RAG工具进行实验

图数据库：Neo4j (Commutity Version)

GraphRAG系统构建框架：Langchain （开源）

LLM 1：Qwen/Qwen2.5-32B-Instruct （开源，作为Task Planning Agent执行Function Calling）

LLM 2：THUDM/glm-4-9b-chat （开源，作为Summarizing Agent执行信息总结任务）

嵌入模型(Embedding Model)：Alibaba-NLP/gte-Qwen2.5-32B-instruct （开源）

LLM API后端：One-API （开源）

LLM 推理框架：Xinference （开源）

API设计及前端交互：FastAPI / HTML JS CSS

（二）检索流程设计

1. 图数据构建：定义审计问题、依据文件、文件条款等核心实体及其关系，清洗和结构化处理审计定性依据数据集。通过实体间的映射关系，设计Cypher语句将数据导入Neo4j图数据库，构建审计知识图谱，并建立索引，为后续的向量索引和查询提供支持。

2. 向量索引：将审计问题描述、文件条款内容、依据文件名称、审计重点名称、审计内容名称等属性的文本内容转换为稠密向量表示。此过程通过使用嵌入模型（本文采用Alibaba-NLP/gte-Qwen2-7B-instruct模型）实现，该模型将文本数据映射到高维向量空间。生成的向量随后存储于Neo4j图数据库中。

3. 查询路由：在查询路由阶段，系统利用LLM的函数调用(Function Calling)能力解析用户查询语句意图，以确定所需匹配的目标实体，进而确定查询路径。

4. 查询语句向量化：在确定查询路径后，根据LLM提炼的关键语句，使用与向量索引过程相同的嵌入模型将关键语句转换为稠密向量表示，以确保查询与存储信息处于同一向量空间，从而实现精确的相似性比较。

5. 相似性检索：在Neo4j数据库中，对第3步确定的实体执行向量检索，匹配与关键语句向量最相似的实体属性向量，本文采用向量间的余弦相似度进行计算。通过检索到的最相似的前k个向量确定实体（其中k为预定义的返回结果数），生成实体及其关联实体、关联关系构成的结构化检索数据。

6. 上下文融合：将第5步得到的检索结果与用户查询结合作为上下文，形成LLM的输入。此步骤使得LLM能够访问图数据库中的相关信息，并利用这些信息生成更准确和更具信息量的回应。

7. 回复生成：LLM基于融合后的上下文，结合预设模板，生成结构化且信息丰富的参考回复。

（三）系统构建

1.系统框架

（1）知识图谱数据库：利用Neo4j图数据库构建审计定性依据库的知识图谱，数据包含审计问题、依据文件、文件条款、审计重点、审计内容等5个实体，审计问题与其它4个实体间的关联关系，依据文件和文件条款的关联关系等。构建实体关键信息的语义嵌入向量，并建立索引。

（2）图向量检索服务：通过预先构建的Task Planning Agent，识别用户意图，通过预先设计Cypher查询语句执行对应路径的向量查询，获取余弦相似度最高的k个实体（Top k），并返回上下游关联实体及相应属性数据。

（3）用户查询界面：用户输入任意自然语句，提交查询后，获得检索数据，并通过精心设计的模板提示词，将检索数据作为上下文，与用户查询语句一起提交给Summarizing Agent进行总结回复。

2. 数据来源与预处理

收集企业内部历史审计问题数据，按照业务域、审计内容、问题表现形式、定性依据、定性依据来源、审计意见等梳理数据，进行清洗、去重、信息补充、文件溯源、数据格式化等处理，使用LLM从定性依据中提取文件名称、文号、条款内容，经过人工复核清洗填充后，完成数据准备，形成《审计定性依据指引》数据集。

3. 知识图谱实体及关系识别

对《审计定性依据指引》数据进行拆解，识别数据集中的实体及其属性，实体间关联关系。

图3. 实体及其属性

|  |  |  |  |
| --- | --- | --- | --- |
| 实体 | 属性 | 数据类型 | 示例数据 |
| 审计问题 | 问题定性 | 字符串 | 违规进行重大决策 |
| 问题描述 | 字符串 | 可行性研究在经济性、技术性等方面论证不充分，尽职调查走过场，存在重大疏漏等，导致实际投资效果未达到预期目标 |
| 是否金额类 | 布尔值 | false |
| 问题类别 | 字符串 | 履行重大经济决策 |
| 审计内容 | 字符串 | 重大投资（融资）项目 |
| 审计意见 | 字符串 | 严格执行“三重一大”集体决策程序规定，落实重大经济事项可行性研究论证程序，进行风险评估和合法合规性审查，确保决策合法合规。 |
| [问题描述]向量 | 数组 | [-0.029405003,0.0015058697,0.009459399,0.006633437,0.0021966973,0.007366527,0.0047365995,-0.016367959,-0.0048824917,-0.039434113,0.008279605,0.01467344… Show all] |
| 依据文件 | 文件名称 | 字符串 | 关于进一步推进国有企业贯彻落实“三重一大”决策制度的意见 |
| 文号 | 字符串 | 文号: 中办发〔2010〕17号 |
| 生效日期 | 日期型 | 20100715 |
| 发布机构 | 字符串 | 中共中央办公厅 |
| 文件类别 | 字符串 | 政策文件 |
| 文件链接 | 字符串 | [https://www.gov.cn/jrzg/2010-07/15/content\_1655395.htm](https://www.gov.cn/jrzg/2010-07/15/content_1655395.htm" \o "https://www.gov.cn/jrzg/2010-07/15/content_1655395.htm) |
| 文件条款 | 文件名称 | 字符串 | 文件名称: 关于进一步推进国有企业贯彻落实“三重一大”决策制度的意见 |
| 条款内容 | 字符串 | （七）“三重一大”事项提交会议集体决策前应当认真调查研究，经过必要的研究论证程序，充分吸收各方面意见。重大投资和工程建设项目，应当事先充分听取有关专家的意见。重要人事任免，应当事先征求国有企业和履行国有资产出资人职责机构的纪检监察机构的意见。…… |
| [条款内容]向量 | 数组 | [-0.017118981,-0.008738745,-0.0015949388,0.027031474,-0.0150722815,-0.011608815,0.015934316,-0.009181158,-0.01136993,-0.009512999,0.00886081,0.01260081… Show all] |
| 审计重点 | 名称 | 字符串 | 履行重大经济决策 |
| 审计内容 | 描述 | 字符串 | 重大投资（融资）项目 |

图4. 实体间关联关系

|  |  |  |
| --- | --- | --- |
| 实体A | 关联关系 | 实体B |
| 审计问题 | 定性依据 | 文件条款 |
| 文件条款 | 属于 | 依据文件 |
| 审计内容 | 包含 | 审计问题 |
| 审计重点 | 包含 | 审计内容 |
| 审计重点 | 包含 | 审计问题 |

4. 图数据导入

在识别了数据集中的实体关系三元组后，设计Cypher查询语句，读取csv数据文件，将处理好的数据导入到Neo4j数据库中。

原始CSV数据字段：

{

问题编号: string;

审计重点: string;

审计内容: string;

问题定性: string;

是否金额类: string(boolean);

条款列表: string[{}...]; // 注释：包含一条或多条定性依据数据

审计意见: string;

}

向图数据库导入数据的Cyphery语句设计：

MERGE (p:审计问题 {问题定性:row.问题定性, 问题描述:row.问题描述})

SET p.问题编号 = row.问题编号,

p.问题类别 = row.审计重点,

p.审计内容 = row.审计内容,

p.审计意见 = row.审计意见,

p.是否金额类 = CASE row.是否金额类

WHEN 'true' THEN true

WHEN 'false' THEN false

ELSE null

END

WITH p, row, apoc.convert.fromJsonList(row.文件条款) AS 条款列表

UNWIND 条款列表 AS 条款详情

MERGE (f:文件 {文件名称: 条款详情.title, 文号: 条款详情.document\_number})

SET f.生效日期 = 条款详情.effective\_date,

f.发布机构 = 条款详情.publisher,

f.文件类别 = 条款详情.type,

f.文件链接 = 条款详情.file\_url

MERGE (r:文件条款 {条款内容: 条款详情.clause})

SET r.文件名称 = 条款详情.title

MERGE (r)-[:属于]->(f)

MERGE (p)-[:定性依据]->(r)

MERGE (c:审计内容 {id:row.审计内容})

MERGE (c)-[:包含]->(p)

MERGE (k:审计重点 {id:row.审计重点})

MERGE (k)-[:包含]->(c)

MERGE (k)-[:包含]->(p)

5. 审计问题向量索引构建

首先在Neo4j[审计定性依据]图数据库中为审计问题实体创建向量索引（通过增加一个problemEmbedding属性），调用嵌入模型将每个审计问题的问题描述转换为文本嵌入向量，然后将这些文本嵌入向量写入到审计问题对应的problemEmbedding属性中。

6. 查询逻辑设计

在审计问题依据查询系统的设计中，首先通过自然语言处理将用户输入的审计问题转换为向量形式，然后在知识图谱中检索与之最相似的问题及其关联信息。接着，将检索到的参考数据作为上下文，与原始查询一起提交给大型语言模型（LLM）以生成答复。最后，系统根据LLM的响应提供给用户精确、内容丰富的查询结果，包括问题定性、描述、依据条款、所属文件以及后续审计工作程序等建议。详细步骤如下：

步骤 1. 审计人员用自然语言描述审计问题或线索疑点，系统通过embedding model对用户查询语句进行向量化，转换为文本嵌入向量，例如：

QuerySentence = “工程项目应招未招应该如何判定？”

经过向量化之后的生成的文本嵌入向量：

QuerySentenceVector = [0.004287188, -0.003524557, -0.0030855776, 0.02104941, 0.0048622238,……]

步骤 2. 将QuerySentenceVector与Neo4j数据库中的所有审计问题嵌入向量(ProblemEmbedding)进行向量匹配，通过余弦相似度方法评估2个向量的相似度（VectorSimilarity），匹配到向量相似度排序最高的3个审计问题的嵌入向量，进而获取到这3个审计问题。每个匹配到的审计问题及其属性、依据条款及所属依据文件等关联信息作为1组参考数据条目，共得到3组参考数据条目，将他们按照Cosine Similarity排序，组合成[参考数据集]，示例数据如下：

[

{

"VectorSimilarity": 0.9431,

"审计问题": "……",

"依据条款": "……",

……

},

{

"VectorSimilarity": 0.9152,

"审计问题": "……",

"依据条款": "……",

……

},

{

"VectorSimilarity": 0.8874,

"审计问题": "……",

"依据条款": "……",

……

}

]

步骤3. 将[参考数据集]分别作为上下文融入LLM系统提示词(System Prompt)，与用户查询语句一同提交给LLM进行答复：

[

{

"role": "system",

"content": "请根据[参考数据集]内容，选择与用户查询问题**最相关**的一条数据作为信息来源进行回答。以下是一段回复模板：\n\n### 一、问题定性：\n\n### 二、问题描述：……\n### 六、分析程序：\n\n(其中第六条是后续审计工作程序的建议，如查阅资料、访谈人员、数据分析等)"

},

{

"role": "user",

"content": "工程项目应招未招应该如何判定？"

}

]

步骤4. 获取LLM响应，得到查询结果。

通过上述步骤，即可完成系统设计，使其成为一套高效、精确、内容丰富的审计定性依据查询系统。

7. 前端交互界面设计

使用HTML、JavaScript及CSS代码设计一个简洁的查询页面，包含页面标题、用户文本输入框、展示区域及按钮点击事件等元素，页面能够支持用户以自然语言描述审计问题，获取最匹配的5个审计定性依据信息。

五、查询验证与结果讨论

本文设计了2种任务来验证GraphRAG系统的效果，以评估其在不同场景下的信息检索能力和回答质量。这2种任务分别为单跳查询和多跳查询。在单跳查询任务中，通过生成语义相近的问题，对比GraphRAG方案和基于FastGPT的传统RAG方案的查询效果，重点考察两者在内容匹配性、回复完整性和答案逻辑性上的表现。在多跳查询任务中，GraphRAG系统被用于检索与给定问题同类的其他审计问题，以验证其在多跳查询上的检索能力和结果准确性。这些验证任务旨在全面评估GraphRAG系统在审计知识库问答系统构建中的优势和适用性。

（一）单跳查询

1. 查询验证方案

随机抽取知识库中已收录工程项目管理、招标采购、财务管理等5个审计业务域下各2个审计问题，根据其问题描述利用LLM生成语义相近的查询问题，分别通过本文所设计的GraphRAG方案和基于FastGPT的Beseline RAG方案进行回答，验证并对比2种方案的查询效果。

为了确保实验可比性，2种方案采用相同的词向量嵌入模型（Alibaba-NLP/gte-Qwen2.5-32B-instruct）和总结回复模型（THUDM/glm-4-9b-chat）；GraphRAG方案仅采用Top K=1情况下回复的单个内容，抽取问题列表如下：

图4. 抽取的10个查询问题列表（单跳问题）

|  |  |
| --- | --- |
| 审计业务 | 查询语句 |
| 工程项目管理 | 1. 未按项目批准范围及内容实施，例如超规模或超预算建设，应该如何处理？ |
| 2. 工程监理单位转包所其承包的监理业务应该如何判定？ |
| 招标采购 | 1. 招标过程中存在串通投标的定性依据是什么？ |
| 2. 单一来源采购计价标准缺乏依据，如何处理？ |
| 财务管理 | 1. 提前确认收入应该如何定性？ |
| 2. 违规对外提供担保的定性依据是什么？ |
| 投资管理 | 1. 投资项目未进行充分尽职调查应该如何处理？ |
| 2. 持续亏损或低效的参股股权项目未及时处置，如何定性？ |
| 资金金融管理 | 1. 未按规定处理经营性结余资金的定性依据是什么？ |
| 2. 未经审批开展金融衍生业务的定性依据是什么？ |

2. 效果评估

（1）评估方案

本文选择阿里巴巴开源的Qwen/Qwen2.5-32B-Instruct作为知识检索评估模型，用于评估2种方案的查询效果。

首先，通过2种方案分别回答上述10个问题；然后，将【①用户提问②《审计定性依据指引》原始数据条目③2种方案分别提供的答案】一同提交给LLM，让其从2种方案响应结果的内容匹配性、回复完整性、答案逻辑性3个维度进行评分（单个维度满分为10分），并计算平均得分。

（2）评估结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 审计业务 | 查询  语句 | 效果评估 | | | | | | | |
| 传统RAG方案（基于FastGPT） | | | | GraphRAG方案 (本文所涉及） | | | |
| 匹配性 | 完整性 | 逻辑性 | 综合得分 | 匹配性 | 完整性 | 逻辑性 | 综合得分 |
| 工程项目管理 | 1 | 4 | 6 | 8 | 6.00 | 9 | 10 | 7 | 8.67 |
| 2 | 4 | 5 | 7 | 5.33 | 9 | 6 | 8 | 7.67 |
| 招标采购 | 1 | 5 | 4 | 9 | 6.00 | 8 | 9 | 9 | 8.67 |
| 2 | 6 | 7 | 8 | 7.00 | 9 | 10 | 8 | 9.00 |
| 财务管理 | 1 | 5 | 6 | 7 | 6.00 | 9 | 8 | 8 | 8.33 |
| 2 | 4 | 7 | 9 | 6.67 | 9 | 10 | 10 | 9.67 |
| 投资管理 | 1 | 7 | 6 | 8 | 7.00 | 8 | 8 | 8 | 8.00 |
| 2 | 5 | 5 | 7 | 5.67 | 10 | 10 | 7 | 9.00 |
| 资金金融管理 | 1 | 7 | 6 | 9 | 7.33 | 9 | 8 | 8 | 8.33 |
| 2 | 6 | 4 | 8 | 6.00 | 9 | 9 | 7 | 8.33 |
| 平均得分 | / | 5.30 | 5.60 | 8.00 | 6.30 | 8.90 | 8.80 | 8.00 | 8.57 |

评估结果表明，GraghRAG方案在处理相同用户查询时，相较于Baseline RAG方案，在内容匹配性、回复完整性和答案逻辑性方面均表现出显著优势，显示出其在信息检索和生成任务中的优越性，对后续构建包括问题定性依据查询在内的审计知识库问答系统具有重要参考价值。

（二）多跳查询

随机抽取《审计定性依据指引》内已收录工程项目管理、招标采购、财务管理等5个审计业务域下各2个审计问题，根据其问题描述利用LLM生成语义相近的查询问题，验证查询效果。

图5. 抽取的10个查询问题列表（多跳问题）

|  |  |
| --- | --- |
| 审计业务 | 查询语句 |
| 工程项目管理 | 1. 未按项目批准范围及内容实施，例如超规模或超预算建设，与其关联的问题有哪些？ |
| 2. 与工程监理单位转包所其承包的监理业务类似的问题有什么？ |
| 招标采购 | 1. 请帮我查询与招标过程中串通投标相关的问题。 |
| 2. 单一来源采购计价标准缺乏依据，有其他类似问题参考吗？ |
| 财务管理 | 1. 提前确认收入相关的问题表现形式有哪些？ |
| 2. 与违规对外提供担保关联的问题有哪些？ |
| 投资管理 | 1. 投资项目未进行充分尽职调查，存在哪几种情况？ |
| 2. 持续亏损或低效的参股股权项目未及时处置，请提供更多类似问题。 |
| 资金金融管理 | 1. 未按规定处理经营性结余资金有哪些关联问题？ |
| 2. 未经审批开展金融衍生业务。请提供与上述问题属于同类审计内容的其他问题。 |

查询验证方案：

由于基于FastGPT的Baseline RAG问答系统并不具备多跳查询能力，此处仅评估本文所设计的GraphRAG方案对于多跳问题的检索回答效果。

首先，让GraphRAG系统分别回答上述10个问题；然后，通过《审计定性依据指引》筛选上述问题所属审计内容下的所有问题清单，与回复结果进行逐一校对，验证其完整性和准确性。

实验结果显示，10个问题查询中，有9个完整召回了其所属审计内容下其他关联问题。未成功完成召回任务的1个问题查询是由于其查询语句匹配到的最大向量相似度的问题条目与其并不属于同一审计内容，通过调整查询语句或选择不同的嵌入模型，可以改善其匹配精确度。该结果验证了GraphRAG方案对于多跳查询问题的回答具有较高的可靠性。

六、总结与展望

本文提出了基于GraphRAG技术的审计定性依据查询系统，并通过实验验证了其有效性以及相对于传统RAG技术的优势。GraphRAG技术能够显著提高审计问题依据查询的效率和准确性，为内部审计工作提供有力支持。

（一）主要贡献

1. 提升查询效率和准确性：GraphRAG技术通过构建审计知识图谱，利用图结构的知识表示和大语言模型的优势，显著提高了审计定性依据查询的准确性和效率。相较于传统的关键词检索和传统RAG方法，GraphRAG能够更全面地捕捉文本中实体间的关联关系，从而提供更完整和高质量的查询结果。

2. 提高复杂查询能力：通过多实体语义向量嵌入及多路径召回，GraphRAG能够高效准确地响应复杂查询问题，为审计人员提供了丰富的延伸性信息参考。这不仅提高了审计查证过程的便利性，还有助于实现审计工作的系统化、全局化和微观化。

3. 结构化数据构建方法：本文提出了基于知识图谱的审计定性依据数据的结构化构建方法，保留了问题定性依据条目各部分的逻辑连贯性，确保了查询结果的完整性和质量。然而，这对数据质量要求较高，依赖于高质量的审计问题数据和知识图谱构建，数据质量的不足可能导致查询结果的准确性下降。

（二）未来研究方向

1. 优化GraphRAG查询路径和召回排序

通过优化GraphRAG的查询路径和召回排序机制，实现更加精准和可靠的信息检索。具体而言，可以通过将现有的关键语句向量查询拆分为多个关键词向量查询，以召回更多相关的数据源。通过将长语句分解为更精细化的关键词组合，并将其映射到图谱中对应的节点上，可以实现更加精准且全面的数据召回。此外，利用先进的重排序（Rerank）模型对初步召回的结果进行进一步筛选和排序，可以显著提升最终回复的匹配性和相关性。

2. 自动化知识图谱数据处理流程

现阶段审计定性依据数据准备仍依赖于审计人员人工收集和整理，数据专家进行数据清洗、知识图谱构建。为了提高系统的构建效率和可扩展性，未来可以进一步研究如何利用LLM实现审计知识数据的自动化处理和知识图谱的自动构建。例如，可以利用LLM对非结构化文本数据进行自动分类、关键信息提取和结构化转换，自动识别和纠正数据错误、异常值和不一致性，根据数据结构和关系信息自动生成图查询语句从而实现自动化的图数据写入等。

3. 探索非结构化知识的GraphRAG方案

本文提出的GraphRAG方案对于结构化审计知识数据具有较强的适配性，通过简易的数据准备、实体及关联关系辨认、召回路径设计便能得到优质的检索效果。然而，很多其他审计知识检索场景中的数据是完全非结构化的，例如审计报告、访谈记录、工作总结等。因此，未来可以重点探索如何有效构建和应用针对非结构化文本数据的GraphRAG方案。例如，使用LLM从非结构化文本中进行命名实体识别、关系抽取和事件抽取来构建知识图谱，从而实现更多复杂和多样性数据类型的处理，进一步扩展其应用场景和实用性。

参考文献：

[1] 陈亚盛, 蒋礼蔚, 单敏, 庄平. 审计大模型的构建及应用研究——以员工违规经商办企业专项审计为例[J]. 审计研究, 2024(04): 139-149.

[2] 易冰心, 刘思琦, 王志勇. AIGC 在审计领域的运用[J]. 审计与鉴证, 2023(11): 69-73.

[3] 金昌铉, 孙丽丽, 郑石桥, 等. 内部审计的合规审计定性依据探讨[J]. 商业会计, 2023(07): 78-80.

[4] 刘锦. AIGC 技术在国家审计中的应用[J]. 审计研究, 2024(4): 18-29.

[5] 向小伟, 申艳光, 胡明昊, 等. 大模型驱动的科技政策法规问答系统研究[J]. 计算机科学与探索, 2024, 18(9): 2349-2360.

[6] 张春芝, 张茜, 党彦威. 智能审计发展与应用探析[J]. 中国注册会计师, 2024(07): 105-107.

[7] Bruckhaus T. RAG Does Not Work for Enterprises[J]. arXiv preprint arXiv:2406.04369, 2024.

[8] Chen B, Zhang Z, Langrené N, et al. Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review[J]. arXiv preprint arXiv:2310.14735, 2023.

[9] Lewis P, Perez E, Piktus A, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks[J]. Advances in Neural Information Processing Systems, 2020, 33: 9459-9474.

[10] Peng B, Zhu Y, Liu Y, et al. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering[J]. arXiv preprint arXiv:2408.08921, 2023.

[11] Pusch L, Conrad T O F. Combining LLMs and Knowledge Graphs to Reduce Hallucinations in Question Answering[J]. arXiv preprint arXiv:2409.04181, 2024.

[12] Rathle P. The GraphRAG Manifesto: Adding Knowledge to GenAI[EB/OL]. https://neo4j.com/blog/graphrag-manifesto/, 2024-07-11.

[13] Xu Z, Cruz M J, Guevara M, et al. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering[J]. arXiv preprint arXiv:2404.17723, 2024.

[14] Yang A, Yang B, Hui B, et al. Qwen2 Technical Report[J]. arXiv preprint arXiv:2407.10671, 2024.

[15] Zhao P, Zhang H, Yu Q, et al. Retrieval-Augmented Generation for AI-Generated Content: A Survey[J]. arXiv preprint arXiv:2402.19473, 2024.