大模型知识管理系统

Large Model Knowledge Management System

周扬/ZHOU Yang，蔡霈涵/CAI Peihan，
董振江/DONG Zhenjiang

（南京邮电大学，中国 南京 210023）
(Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

DOI：10.12142/ZTETJ.202402010
网络出版地址：http://kns.cnki.net/kcms/detail/34.1228.TN.20240422.1818.002.html
网络出版日期：2024-04-23
收稿日期：2024-03-10

摘要：企业知识管理至关重要，而传统企业知识管理系统存在构建成本高、知识利用率低的问题。提出了基于大模型检索增强生成（RAG）技术构建企业知识管理系统的方案。首先介绍了整体方案架构、业务流程与4类知识构建技术，然后重点介绍了检索前处理、知识检索、检索后处理等全流程知识检索技术，并设计了全面的测评框架。经过实践检验，该方案具有知识构建效率高且成本低、意图理解精确、知识检索准确等特点与优势。

关键词：RAG；知识管理系统；大模型；知识工程

Abstract: Enterprise knowledge management is very important, but traditional enterprise knowledge management systems suffer from high construction costs and low knowledge utilization rate. A scheme to build enterprise knowledge management system based on retrieval-augmented generation (RAG) technology is proposed. Firstly, the overall scheme architecture, business processes and four types of knowledge construction technologies are introduced, and then the whole process of knowledge retrieval technology are discussed, such as retrieval pre-processing, knowledge retrieval, retrieval post-processing, and subsequently a comprehensive evaluation framework is designed. The scheme has the characteristics and advantages of high efficiency and low cost of knowledge construction, accurate intention understanding, and accurate knowledge retrieval.

Keywords: RAG; knowledge management system; large model; knowledge engineering

引用格式：周扬，蔡霈涵，董振江. 大模型知识管理系统 [J]. 中兴通讯技术, 2024, 30(2): 63-71. DOI: 10.12142/ZTETJ.202402010
Citation: ZHOU Y, CAI P H, DONG Z J. Large model knowledge management system [J]. ZTE technology journal, 2024, 30(2): 63-71. DOI: 10.12142/ZTETJ.202402010

在当今信息化快速发展的时代，企业面临着前所未有的数据增长和知识爆炸挑战。有效地管理和利用这些知识资源成为企业获得竞争优势、促进创新和提高决策质量的关键因素。这是企业知识管理[1]的重要研究内容，旨在帮助组织系统地收集、整合、共享和分析企业内外的知识和信息，从而最大化知识资产的价值。

传统企业知识管理系统以共享知识库为核心，如ONES Wiki、PingCode Wiki等，旨在搭建共享知识和交流平台。该方案面临多重挑战，主要包括：1）知识库的内容来源于多源异构，大量非结构化知识需要人工处理，转换为关系数据库或知识图谱数据后才能提供服务，但构建效率低、成本高、周期长。2）传统的全文检索技术仅依赖于关键词匹配和倒排索引。随着知识库规模的扩大，用户在庞大的知识库中难以获取所需知识，检索效率低下。3）系统功能单一，用户体验差。系统主要提供固定字段检索或按关键词的全文检索，用户需要在搜索结果中自行提取所需信息，多次搜索仍难以找到合适信息。

随着人工智能技术，尤其是大语言模型（LLM）技术的迅猛发展，企业知识管理的潜力有待进一步挖掘。LLM如ChatGPT、Qwen[2]、Gemini[3]、Gemma[4]等，具有良好的自然语言理解能力，不仅可以处理和分析大量文本数据，还能够生成高质量摘要，回答复杂的查询，甚至推动自动化决策。这些能力有助于大幅提升知识管理的效率和智能化水平。但是大语言模型在生成最终答案时，因自身专业领域知识不足、知识更新不及时以及企事业单位数据无法获取等原因，会出现幻觉而生成不当内容，这在要求内容准确、专业、合规的政企领域成为应用推广的最大障碍。检索增强生成（RAG）应运而生，成为了当前业界解决该问题的核心技术。

基金项目：江苏省重点研发计划项目（BE2023025）

RAG技术概念最早由Meta提出[5]。受限于当时较差的语言模型能力，尽管RAG技术已经在多个知识密集型自然语言处理（NLP）任务上取得了不错效果，但其并未引发更多的关注。在大模型时代，模型的性能取得了巨大的提升，伴随而来的幻觉问题使RAG技术重新进入人们的视野。通过从多数据源中获取外部知识，结合搜索技术和LLM的提示词功能，RAG向大模型提出问题，并把问题在多数据源中进行搜索获取的知识作为背景上下文，将问题和背景上下文信息整合到LLM的提示词中，从而让LLM做出最终的准确回答。

在大模型时代，RAG的发展可分为3个阶段。1）基础RAG（Native RAG）：遵循传统的工作流程包括索引、检索和生成3个模块，也被称为“检索–读取”框架。首先各类知识被分割成离散的块，然后利用embedding模型构建这些块的向量索引；其次，RAG根据查询和索引块的向量相似性识别和检索块；最后，模型根据从检索到的块中获得的上下文信息合成响应。2）高级的RAG：通过丰富的前处理和后处理技术，在信息检索精度和准确率上取得了显著效果。3）模块化的RAG（Modular RAG）：将RAG前、后处理等技术抽离出来并形成模块，进行组合。模块化RAG相比于传统的Native RAG框架，提供了更好的通用性和灵活性。

本文中，我们设计了基于RAG架构的LLM知识管理系统。该系统在充分利用LLM提高知识管理水平的同时，有效缓解了LLM可能产生的幻觉和不当内容问题。

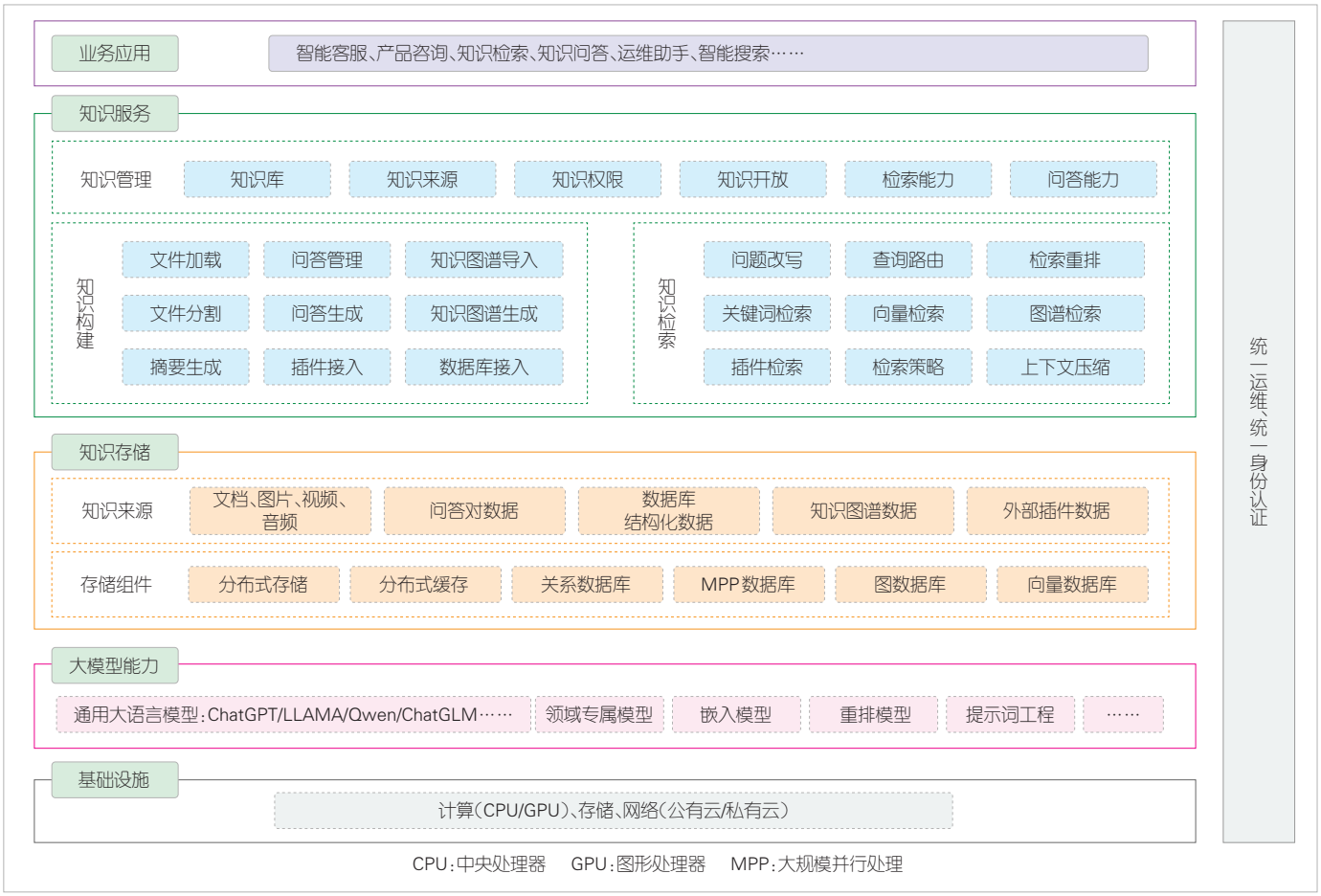
1 系统方案

知识管理系统旨在将人类知识以计算机可理解的形式表示出来，并使计算机能够理解、推理和应用这些知识。这项技术涉及知识表示、知识获取、知识推理、知识存储和知识管理等方面。

1.1 系统架构

如图1所示，LLM知识管理系统架构主要分为以下几个部分：基础设施层、大模型能力层、知识存储层、知识服务层和业务应用层。

基础设施层是构建LLM知识管理系统的底层基础，包



▲图1 大模型知识管理系统架构

括运行系统所需要的计算、存储和网络资源，特别是用于模型部署和推理需要的图形处理器（GPU）资源。部署方式可以是基于公有云服务的部署，也可以是基于企业内部私有云的部署。

在基础设施层之上是大模型能力层，该层包括多种预训练的通用LLM，如ChatGPT、Gemma、LLAMA[6]、Qwen、ChatGLM[7]等，用于理解和生成自然语言，是系统的智能核心。它不仅包括针对特定领域训练或微调以适应特定领域的专属LLM，还包括用于知识构建和知识检索过程中的嵌入模型、重排模型等其他模型，以及使用这些模型的提示词工程。它通过调用基础设施层的计算资源，为整个系统的其他各层提供大模型服务。

知识存储层负责存储和管理企业的知识资产。该层在系统中主要提供知识的存储服务。其中，传统的数据库系统用于存储结构化数据，分布式存储系统用于存储文档、图片、音频和视频等非结构化数据，图数据库用于存储知识图谱数据，外部插件系统用于访问通过外部应用程序编程接口（API）获取的外部知识（例如搜索引擎API等）。另外，向量数据库用于存储基于大模型嵌入技术产生的向量数据。

知识服务层分为3个部分，分别是知识构建、知识检索和知识管理。知识构建主要将来自多种来源的知识数据进行预处理，然后导入到系统，并使用知识存储层的存储组件进行存储。常见的知识数据来源包括非结构化的文档数据、结构化数据库数据、问答（QA）数据、知识图谱数据以及外部API插件数据等。知识检索主要实现根据用户问题获取知识答案的过程。检索的第一步需要对用户问题进行理解和改写，随后采取多种方式进行检索。多种检索方法获得的数据还会经历重排过程，并由大模型最终理解后生成检索结果。知识管理将系统能力统一封装和管理，对业务层提供知识服务能力，同时封装统一的知识开放接口、知识检索能力接口和知识问答能力接口供上层业务层使用。

业务应用层展示了基于LLM知识管理系统构建的常见业务应用。通过知识服务层提供的知识服务，该层提供了以问答方式提供服务的智能客服、面向市场售前人员或客户的产品咨询助手，面向企业提供知识检索和知识问答应用（特别是图书馆图书检索、档案馆档案检索、法律法规条文检索、知识产权专利检索）、复杂系统和场景的运维服务助手，以及基于大模型新一代搜索引擎等应用。

1.2 业务流程

基于LLM的企业知识管理系统的业务流程主要包括知识构建流程、知识检索流程和基于大模型的答案生产流程，如图2所示。知识构建流程包括知识数据预处理、建立索引和知识存储，主要是将企业内部的数据库、知识图谱、文档，外部的Web知识以及构建的QA对进行统一的处理，并存储为企业知识库的统一形式，以完成企业知识数据的处理和构建。知识检索流程包括检索前处理、知识检索、检索后处理、答案生成等步骤。其中，知识检索前处理和检索后处理是可选步骤，在基础知识检索过程中，可能会缺少相关步骤。在知识库构建完成后，用户使用企业知识库进行知识检索。知识检索过程将获取与用户问题相关知识内容的上下文信息。最后，基于知识检索过程获得上下文内容，LLM生成最终答案。

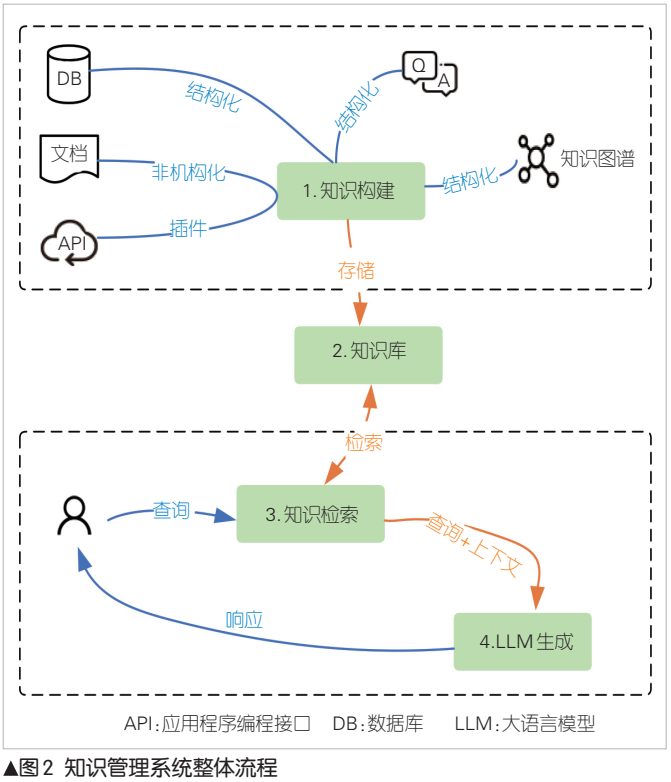
2 关键技术

2.1 知识构建技术

知识构建是企业知识管理系统的核心部分，负责将企业原始知识数据转化为易于存储和检索使用的结构化知识，并将其存入知识库进行管理。构建的知识库的知识质量决定了RAG的最终检索质量。企业的知识来源丰富多样，包括文档、知识图谱、数据库、外部插件等。

2.1.1 文档知识

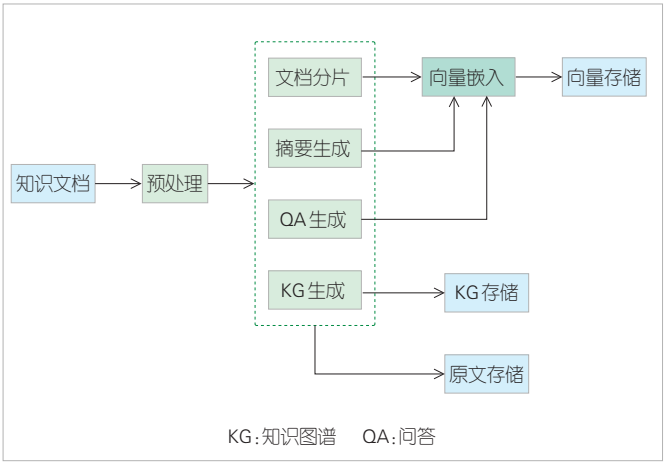
文档型知识是企业知识的主要来源之一。通常，文档型



▲图2 知识管理系统整体流程

知识需要经过预处理、文档切分、向量嵌入等过程，才能完成从原始文档数据到知识库中知识的转变。其中，文档切分算法是一个关键的技术。良好的切分算法应该在满足切片大小的限制的同时，保证每一个切片的语义相对完整。常见的切分算法包括按段落递归切分、按标题切分、按行切分、按固定分隔符切分、按标题切分、按语义切分等。具体的文档知识构建流程如图3所示。

结构化良好的文档，比如word、pdf、html、Markdown等格式文档，通常具有章节结构或标题层次信息。因此，我们可以考虑文档本身的章节或标题层次结构信息，使用按标题切分算法将标题内容和正文内容综合起来进行切片，通过在每一个切片内容的头部添加切片所在的章节或标题信息，使切片内容可以更好地保持原始文档中的语义信息。



▲图3 文档知识构建流程

2.1.2 知识图谱知识

知识图谱通过将知识组织成网络结构的图来表示，它连接了各种实体和它们之间的关系，为知识提供了一种直观的结构化表示。知识图谱构建流程一般包括数据预处理、实体识别、关系提取、属性提取、知识整合和存储等步骤。其中，实体识别和关系提取是较为关键的步骤，传统上可以分别调用专业的小模型，如在实体识别任务上取得SOTA的W2NER[8]、LERERT[9]等模型，在关系提取上取得SOTA的CasRel[10]模型等，来完成相应任务。而在大模型时代，由于LLM具有强大的语义理解能力，可以使用LLM通过预设好的Prompt进行实体识别和关系抽取，形成三元组进行存储。

2.1.3 数据库知识

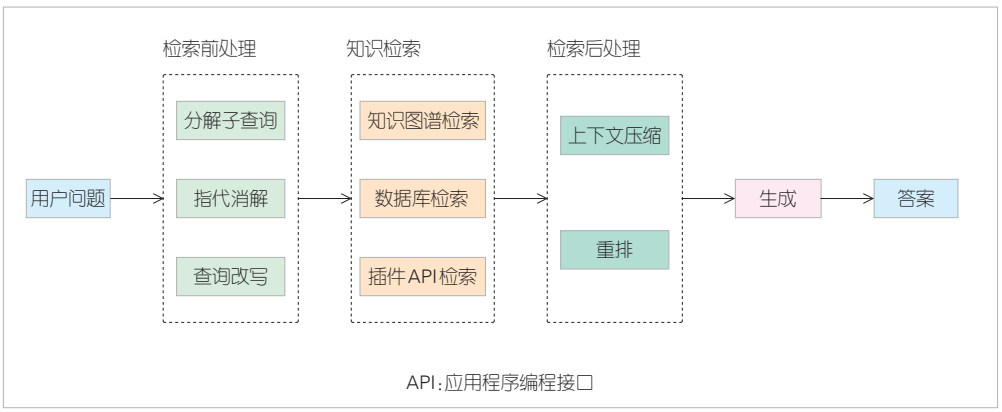
数据库知识指的是存储在传统关系数据库、分析型数据库等数据库中的知识。在信息化建设过程中，企业一般都陆续积累了大量的数据库数据。通过关系数据库理论或数据仓库理论，企业建立了相关的数据库和表，在企业知识管理系统中并不需要重复建设这部分知识数据，但是需要将这部分数据纳入知识管理系统中，以便用户方便地使用已有的数据库知识。

2.2 知识检索技术

知识检索是RAG的核心过程，也是企业知识管理系统的最重要的部分，其目的是确保用户能够快速、准确地找到所需的信息。知识检索包括基础检索流程和复杂检索流程。其中，典型的复杂检索流程包括检索前处理、知识检索、检索后处理、答案生成等步骤。每一个检索步骤都涉及众多的技术细节，而基础检索流程常省略检索前处理和后处理环节。为了提升RAG的准确度，我们采取了多种前后处理技术，并采用了混合知识检索。知识检索具体流程如图4所示。接下来，我们对前处理、知识检索、后处理分别进行介绍。

2.2.1 前处理

在企业知识检索系统中，前处理是指对用户查询进行预处理的一系列技术和方法，旨在优化查询，以提高检索效率和准确性。前处理方法有很多，包括多查询扩展、分解子查询、术语替换、补全历史、指代消融、假设答案、StepBack提示词、查询改写、查询路由等。通过前处理，企业知识检索系统能够更有效地理解用户的查询意图，优化查询以适应复杂的检索环境，从而提供更准确、更相关的检索结果。这里我们将对分解子查询、指代消解、查询改写进行介绍。



▲图4 知识检索流程

2.2.1.1 分解子查询

分解子查询的核心理念在于将一个复杂的原始查询拆分成若干个更小、更易于处理的部分，其中每个部分均代表一个信息独立的子问题。为了实现这一目标，可以采用多查询检索器。该检索器借助LLM，从多个维度自动生成针对给定用户输入的多个查询，进而自动执行提示优化流程。对于生成的每个子查询，多查询检索器都会检索一组与之相关的文档，并最终对所有子查询检索到的文档采取并集操作，从而构建出一个更广泛的潜在相关文档集合。分解子查询可以突破基于向量距离检索方法的某些局限性，从而获取一组更为丰富和多元的检索结果。

2.2.1.2 指代消解

指代消解技术适用于处理用户查询中含有指代词（如“它”“这个公司”等）的情况，有助于提高检索系统对用户查询的准确性。传统通过微调BERT进行指代消解的技术往往只适用于有限、简单的查询语句，在LLM时代，相较于传统的依赖专用小模型进行微调的方法，可以采用Few-shot Prompt并结合思考-行动-观察（CoT）的策略进行指代消解。通过将一些常见的指代消解场景作为Few-shot例子集成到LLM的Prompt中，结合CoT方法，LLM能够分析并处理更复杂的指代消解问题。

2.2.1.3 查询改写

在知识检索系统中，查询改写技术常常可以大幅提高检索准确度。它通过对用户最初的查询进行语言层面的优化与调整，可以增强检索效率并提高结果的精准度。此技术特别适用于处理那些表达模糊不清、含义不明确或结构过于复杂的查询。通过这种方式，系统能够更精确地把握用户的信息需求，并返回更相关的检索结果。

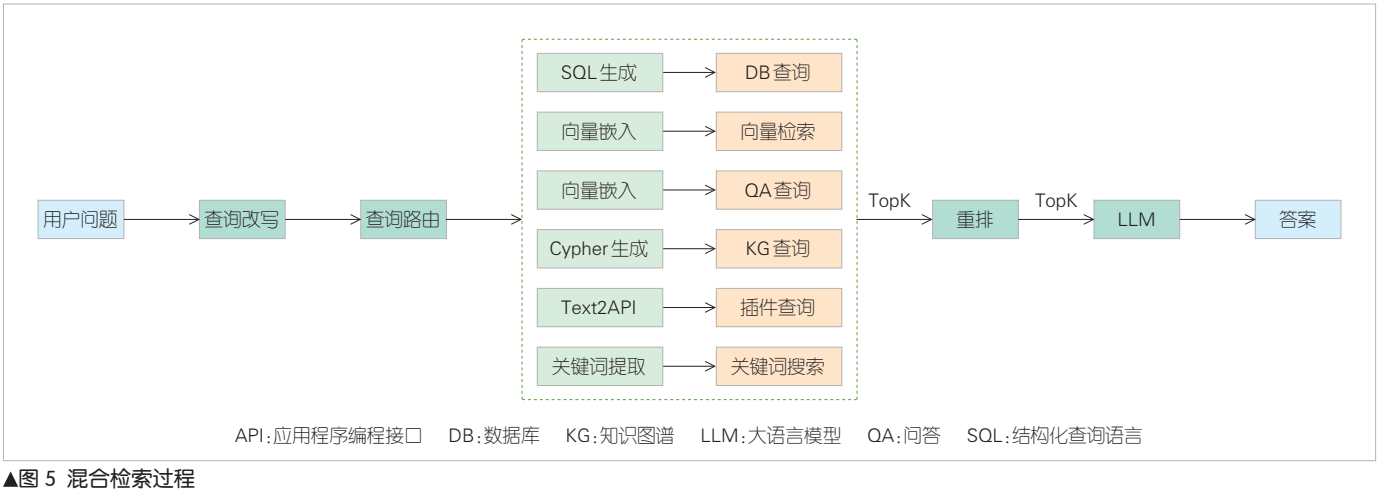
为提高检索系统的准确性，查询改写技术依靠LLM的强大能力，利用精心设计的提示词，让LLM能够有效地改写用户的查询。为了进一步提升查询改写效果，我们还可以引入一个辅助模型“重写器”[11]。这个辅助模型专门负责调整用户查询，使其更好地适应固定检索器和LLM的处理要求。辅助模型重写器可以通过收集领域数据进行有监督的预训练或微调获得。这样，重写器就能更好地满足实际应用场景中的改写需求。

2.2.2 知识检索

在知识检索过程中，通常采用多种检索策略来增强检索的深度，提高检索结果的准确性。如图5所示，混合检索通常将用户查询问题进行改写生成一个或多个查询。经过查询路由模块后，这些查询问题被分发到不同的检索方法流程中。常见的检索方法包括数据库查询、向量检索、QA检索、知识图谱检索、插件检索、关键词检索等。经过多重检索方法检索后每一种检索方法将输出TopK个检索结果。由于不同检索方法生成的检索结果打分标准不同，它们并不能简单地组合在一起进行排序，这时候就需要引入新的重排算法来对这些TopK结果进行组合和重新排序，从而得到最终的TopK，并丢弃其他的候选检索结果。这些最终被选中的TopK将作为上下文和用户查询问题一起交给大模型，让LLM基于上下文内容为用户的提问生成答案。

2.2.2.1 知识图谱检索

知识图谱检索是一种利用知识图谱信息来检索和提供与特定任务相关信息的技术。传统的知识图谱检索较为复杂，一般包括从查询中进行实体识别、关系识别和查询匹配等步



▲图5 混合检索过程

骤。每一个步骤往往都需要专门微调一个小语言模型，而且对于不同的知识图谱，往往需要重新进行微调训练，时间成本较高。在大模型时代，利用大模型出色的语义理解能力和prompt提示词工程，我们仅需要一个大模型就可以较好地对多个知识图谱进行知识检索。

基于大模型的知识图谱检索有两种方式：Text2Cypher和GraphRAG。其中，Text2Cypher将用户问题翻译成图数据库能够识别的Cypher语句，然后调用图数据库接口执行这个生成的Cypher语句以获得执行结果，并将执行结果通过LLM能力生成最终答案。GraphRAG通过构造子图（Sub-Graph）方式来利用知识图谱中的上下文知识以处理用户查询。它首先从用户输入的查询内容中提取实体，然后通过构建与查询相关实体的子图来建立上下文，最后将子图信息作为上下文和用户查询一起送给大模型以给出准确的回答。知识图谱检索工作流程如图6所示。

2.2.2.2 数据库检索

Text2SQL，也称为NL2SQL，是指将自然语言（NL）查询转换为关系型数据库中可执行的SQL查询语言的过程。用户能够以自然语言形式提出查询请求，无须编写SQL语句，从而降低了与数据库交互的复杂性。与知识图谱检索类似，传统的Text2SQL方法也存在流程复杂、组件冗余的情况。同时，采用传统的Text2SQL方法，准确性也难以得到保障。通过引入大模型，我们可以加速整个Text2SQL的流程，并将准确率由原先的60%提升到80%[12]。

Text2SQL进行数据库数据检索主要包括以下步骤：首先利用Schema过滤器筛选与用户输入相关的Schema，然后将筛选的Schema列表与问题一并交给大模型，利用大模型生成SQL语句并执行，最终借助大模型对SQL的执行结果进行分析和总结。

2.2.2.3 插件API检索

插件API检索是指通过API调用外部服务或功能的过程，这被视为LLM与外部世界交互的一种方式。这种交互经常涉及函数调用（Function Calling）。更具体地，它涉及通过API发送请求和接收响应。这些API可能由第三方服务、工具集或自定义实现提供，比如：OpenAI的联网检索和代码解释器就是常见的两种插件检索应用形态。在传统的插件API检索中，面对繁杂的插件API，系统往往难以准确调用正确的插件API。在大模型时代，大模型能够较好地通过API描述，并结合查询，从而较为准确地调用相关API进行检索。

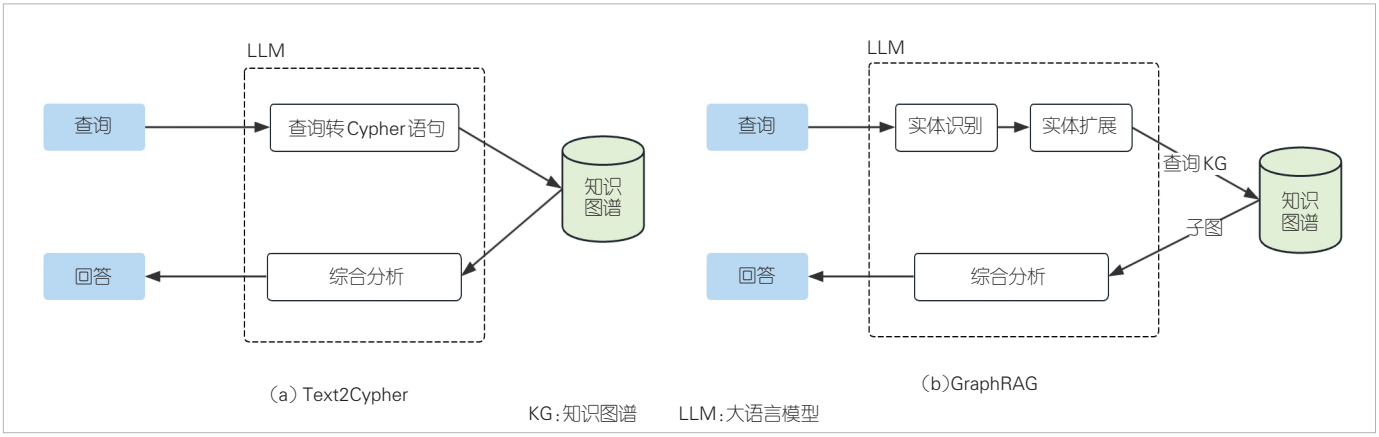
插件API检索的流程首先需要使用API过滤器将用户通过自然语言输入的用户查询进行筛选过滤，从中取出TopK候选相关的API，并将这些API定义和用户查询一起送给大模型进行处理。对于支持Function Call功能的LLM，它会返回函数调用的名称和参数等信息。

2.2.3 后处理

后处理（Postprocessing）阶段负责对检索结果进行进一步的优化和调整，以提高检索系统性能和检索结果质量。这一阶段的核心活动包括但不限于对检索结果进行筛选、压缩和重新排序等操作。进行这些操作的目的是为了精炼并整理出一组最终结果。这些结果随后将被提交给LLM以生成响应答案。在本节中，我们将对上下文压缩、重排技术进行介绍。

2.2.3.1 上下文压缩

通过RAG获得的上下文长度常常达到数千个tokens。当检索步骤所获得的结果内容较多并超出大模型上下文长度时，需要对上下文进行压缩处理以去除冗余信息，查询无关



▲图6 知识图谱检索

噪声，同时保持语义不丢失，进而为LLM提供更有效的上下文信息。常见的上下文压缩方法有内容摘要、关键词提取、LongLLMLingua[13]等。其中，LongLLMLingua通过使用对齐并训练好的小模型来检测移除上下文中不重要的token，并将其转换为人类难以理解但LLM易于理解的形式，有效提升了系统性能。LongLLMLingua的核心思想是将长输入分两步处理：首先使用一个小型编码器模型（通常是BERT等双向编码器）将长输入编码为一个较短的向量表示，然后将编码后的向量连同查询一起输入到一个LLM中（LLM解码器能够识别小型编码器编码后的信息），生成最终的输出。

2.2.3.2 重排

在检索后处理阶段，为确保最相关且最有价值的检索结果能够优先被用作回答查询的上下文输入，我们引入了重新排序（Reranking）机制。重排操作通过对检索阶段获得的检索结果相关性评分进行再次调整，或采用更精细的排序算法，从而实现检索结果的重新排列。重排的关键在于设计高效的打分模型。常见的做法是引入交叉编码器。对于给定查询，交叉编码器将所有检索结果与之进行编码打分，然后按得分递减排列，得分最高者即为最相关检索结果。

为进一步提升重排性能，我们采用了经过训练的专门用于重排的模型，其中Cohere公司的Cohere重排模型和智源的bge-rerank[14]模型因具有代表性而被广泛使用。本文中，我们选用了bge-rerank作为重排器，搭配bge-embedding模型进行文档嵌入，取得了良好效果。重排环节的优化有助于提高上下文的相关性和质量，从而为最终答案生成提供更为可靠的语义支撑。

2.3 答案生成技术

答案生成技术是指，依赖LLM本身的推理能力，结合系统提供的上下文信息进行最终的答案生成。目前，根据开源情况，主流的LLM可以分为以ChatGPT为首的闭源模型和以LLaMA、Qwen为首的开源模型两类。

在闭源大模型中，OpenAI的ChatGPT-4常常在各大评测排行榜中名列前茅，而近期出现的Claude3也显示出了强大的性能。然而，尽管这些模型性能强劲，但由于它们是闭源模型，只提供API调用接口，费用昂贵，不适用于企业知识库中需要频繁调用的场景。此外，企业知识管理系统通常涉及大量的企业内部知识，这对闭源商业模型的隐私保护提出较高要求。

本文所提知识管理系统方案采用了开源大模型。在开源大模型中，比较有名的包括清华大学的ChatGLM、阿里的Qwen以及Meta的LLaMA，具体的参数规模和说明如表1所示。可以看出，ChatGLM-6B受限于参数规模，相较于14B的模型性能略有不足，而LLaMA模型本身只支持英文，即使引入了中文补丁，在中文语境下，Qwen模型性能更胜一筹。综合考虑，我们在方案中选择了Qwen-14B模型。

3 测评框架

为全面评估基于RAG架构的知识管理系统的性能表现，我们需要一个科学全面的测评框架。由S. ES等于2023年9月提出的检索增强生成评估（RAGAs）[15]开源评估框架在业界取得了良好的反响。RAGAs能够快速对RAG系统进行综合评估，所需的输入包括：用户提出的查询问题（Question）、RAG系统生成的答案（Answer）、检索到的与问题相关的上下文文档（Contexts），以及人工标注的参考答案（Ground Truths）。

在获得上述输入信息后，RAGAs基于以下4个评估指标对RAG系统效果进行量化评分：

1）Faithfulness，衡量生成答案与上下文是否保持一致，反映了系统回答的可信赖性。

2）Answer Relevancy，评估生成答案与参考答案的语义相关度，考察答案的准确性。

3）Context Relevancy，测量检索上下文与问题的关联程度，体现上下文选择的恰当性。

4）Context Recall，计算系统检索到的相关上下文数量

▼表1 开源大模型参数规模和说明

模型名称	参数大小/亿	MMLU	CEval	AGIEval	推理显存/GB
ChatGLM-6B[7]	62	36.90	38.90	/	6
LLaMA-7B[6]	70	35.10	27.10	23.90	6
LLaMA-13B	130	46.94	/	33.90	10
Qwen-7B[2]	70	56.70	59.60	/	8
Qwen-14B	140	66.30	72.10	/	13

MMLU：大规模多任务语言理解

占总相关上下文的比例，反映上下文覆盖的完整性。

通过同时关注上述4个维度指标，RAGAs可以综合评估RAG系统在可靠性、准确性和泛化能力等方面的整体水平，为持续优化和改进系统性能提供量化指引。接下来，我们将详细介绍这4个评估指标的具体含义。

Faithfulness是衡量RAG生成的答案Answer与检索到的上下文Context的事实一致性。它是根据Answer和Context计算得出的。Faithfulness的取值范围为0～1之间，且越高越好，计算公式如公式（1）所示：

$$\text{Faithfulness score} = \frac{|\text{Number of claims that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|}。\quad (1)$$

Answer Relevancy是评估RAG生成的答案（Answer）与用户问题（Question）之间的相关程度。当RAG生成的答案不完整或包含不相关的信息时，系统则将获得较低分数。Answer Relevancy的取值范围为0～1之间，且越高越好，计算公式如公式（2）所示：

$$\text{Answer Relevancy} = \frac{1}{n}\sum_{i=1}^{n}\text{sim}(q,q_i),\quad (2)$$

其中，q为原始问题Question，q_i为提示LLM生成基于该Answer的可能的第i个问题，$\text{sim}(q,q_i)$是计算原始问题q和生成问题q_i的余弦相似度。

Context Relevancy衡量检索到的上下文Context的相关性，根据用户问题Question和检索到的上下文Context计算得到，取值范围在0～1之间，值越高表示相关性越好。理想情况下，检索到的Context应只包含解答Question的信息，计算公式如公式（3）所示：

$$\text{Context Relevancy} = \frac{|S|}{|\text{Total number of sentences in retrived context}|}。\quad (3)$$

Context Recall是衡量检索到的上下文Context与人类提供的真实答案Ground truth的一致性程度。它是根据Ground truth和检索到的Context计算出来的，取值范围在0～1之间，值越高表示性能越好，计算公式如公式（4）所示：

$$\text{Context Recall} = \frac{|\text{GT sentences that can be attributed to context}|}{|\text{Number of sentences in GT}|}。\quad (4)$$

4 结束语

本研究旨在构建一个基于RAG架构的大型企业知识管理系统，以期为企业提供高效的知识检索和利用能力。本文中我们提出了基于RAG构建企业知识管理系统的架构、流程和方法。该系统采用开放的系统架构设计，可基于开源或商业LLM构建，充分保障了企业关注的数据安全；支持多种知识来源，包括文档、知识图谱、数据库和问答等，通过深度挖掘和融合这些异构知识源，形成了全面的专业知识基础。此外，我们设计并实现了完整的知识检索方案，包括检索前处理、知识检索、检索后处理和答案生成等环节，并采用了多种创新技术来提升检索效率和答案质量，介绍了使用RAGAs评估框架对构建的企业知识管理系统进行评估和迭代优化的情况。大量用户反馈和实验评估表明，该系统在准确性、知识覆盖范围、检索效率和用户体验等多个维度均有着优异的表现。

然而，系统中仍存在一些需要进一步改进的问题。首先，当前系统所使用的知识来源仍以文本为主，缺乏将多模态知识融入系统的合理方法。其次，尽管采用了多种文档切分和检索优化手段，但在实际应用场景中还需要针对特定的文档内容设计定制化文档切分算法。最后，系统已经较好地缓解了大模型幻觉的问题，但在企业应用场景下还需要考虑企业合规对齐、数据安全等问题。

在未来，我们希望可以设计更多的垂直领域文档切分算法，采取更有效的embedding和Rerank组合模型，进一步提升RAG技术的检索效率和准确度，同时引入最终回答的合规审查机制，构建一个更高效、更安全的基于RAG的大模型知识管理系统。

参考文献

[1] 牛菁. 大数据赋能企业知识管理创新机理与路径研究: 基于华为案例[J]. 中国新通信, 2023, 25(11): 19-21. DOI: 10.3969/j.issn.1673-4866.2023.11.008
[2] BAI J Z, BAI S, CHU Y F, et al. Qwen technical report [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2309.16609
[3] TEAM G, ANIL R, BORGEAUD S, et al. Gemini: a family of highly capable multimodal models [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2312.11805
[4] TEAM G, MESNARD T, HARDIN C, et al. Gemma: open models based on gemini research and technology [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2403.08295
[5] LEWIS P, PEREZ E, PIKTUS A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks [C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. ACM, 2020: 9459 - 9474. DOI: 10.5555/3495724.3496517
[6] TOUVRON H, LAVRIL T, IZACARD G, et al. LLaMA: open and efficient foundation language models [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2302.13971
[7] ZENG A H, LIU X, DU Z X, et al. GLM-130B: an open bilingual pre-trained model [EB/OL]. [2024-02-25]. http://arxiv. org/abs/2210.02414
[8] LI J Y, FEI H, LIU J, et al. Unified named entity recognition as word-word relation classification [J]. Proceedings of the AAAI

conference on artificial intelligence, 2022, 36(10): 10965–10973. DOI: 10.1609/aaai.v36i10.21344
[9] LIU W, FU X Y, ZHANG Y, et al. Lexicon enhanced Chinese sequence labeling using BERT adapter [C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2021: 5847–5858. DOI: 10.18653/v1/2021.acl-long.454
[10] 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). Association for Computational Linguistics, 2023: 1762–1777. DOI: 10.18653/v1/2023. acl-long.99
[11] MA X B, GONG Y Y, HE P C, et al. Query rewriting in retrieval-augmented large language models [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023. DOI: 10.18653/v1/2023.emnlp-main.322
[12] GAO D W, WANG H B, LI Y L, et al. Text-to-SQL empowered by large language models: a benchmark evaluation [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2308.15363
[13] JIANG H Q, WU Q H, LUO X F, et al. LongLLMLingua: accelerating and enhancing LLMs in long context scenarios via prompt compression [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2310.06839
[14] XIAO S T, LIU Z, ZHANG P T, et al. C-pack: packaged resources to advance general Chinese embedding [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2309.07597
[15] ES S, JAMES J, ESPINOSA-ANKE L, et al. RAGAS: automated evaluation of retrieval augmented generation [EB/OL]. [2024-02-25]. http://arxiv.org/abs/2309.15217

作 者 简 介

周扬，东南大学和南京邮电大学特聘副教授；研究领域包括人工智能、大数据、物联网等；参与国家级重点项目8项，在人工智能、大数据、物联网等领域具有近20年的大型产品研发与管理经验；申请发明专利20余项。

蔡霈涵，南京邮电大学在读硕士研究生；主要研究方向为人工智能。

董振江，南京邮电大学教授、博士生导师，国务院政府特殊津贴专家，中国人工智能学会常务理事；主要研究方向为人工智能、数据安全与区块链。