大型检索增强生成语言模型:综述

Yunfan Gaoa ，Yun Xiongb , Xinyu Gaob , Kangxiang Jiab ，Jinliu Panb ，Yuxi Bic ，去做吧。 ，Jiawei Suna 、孟Wangc , and Haofen Wang

a同济大学上海自主智能系统研究所b复旦大学计算机学院上海市数据科学重点实验室c同济大学设计创意学院

摘要:大型语言模型 (LLM) 展现出令人印象深刻的功能,但也面临着诸如幻觉、知识过时以及推理过程不透明、不可追踪等挑战。检索增强生成 (RAG) 通过整合外部数据库的知识,已成为一种颇具前景的解决方案。这提高了生成的准确性和可信度,尤其对于知识密集型任务而言,并支持持续的知识更新和领域特定信息的集成。RAG 将 LLM 的内在知识与外部数据库庞大、动态的存储库协同融合。这篇全面的综述文章详细探讨了 RAG 范式的演变,涵盖了朴素 RAG、高级 RAG 和模块化 RAG。

本文仔细研究了 RAG 框架的三部分基础,包括检索、生成和增强技术。论文重点介绍了每个关键组件中嵌入的先进技术,从而深入了解 RAG 系统的进步。此外,本文还介绍了最新的评估框架和基准。最后,本文概述了当前面临的挑战,并指出了未来的研究和发展方向。

[1]。

索引词 大型语言模型、检索增强生成生成、自然语言处理、信息检索

一、引言

大型语言模型(LLM)取得了显著的成功，但面临显著的局限性，尤其是在特定领域或知识密集型任务中[1]，尤其是在处理超出训练数据或需要当前信息的查询时，会产生“幻觉”[2]。为了克服这些挑战，检索增强生成 (RAG) 通过语义相似度计算从外部知识库中检索相关文档块，从而增强了 LLM。通过引用外部知识，RAG 有效地减少了生成与事实不符内容的问题。

它与 LLM 的集成已得到广泛采用，使 RAG 成为推进聊天机器人发展和增强 LLM 在实际应用中适用性的关键技术。

RAG技术近年来发展迅速，相关研究总结技术树如下

通讯作者.Email :haofen.wang@tongji.edu.cn 1资源是RAG调查
网址:https://github.com/Tongji-KGLLM/

如图1所示，大模型时代RAG的发展轨迹呈现出几个明显的阶段特征。

最初，RAG 的诞生与 Transformer 架构的兴起相吻合，该架构专注于通过预训练模型 (PTM) 整合额外知识来增强语言模型。这一早期阶段的特点是致力于改进预训练技术 [3]–[5] 的基础性工作。随后 ChatGPT [6] 的出现标志着一个关键时刻，LLM 展现出强大的情境学习 (ICL) 能力。RAG 的研究转向为 LLM 在推理阶段提供更优质的信息，使其能够应对更复杂、知识密集型的任务，从而推动了 RAG 研究的快速发展。随着研究的进展，RAG 的增强不再局限于推理阶段，而是开始更多地融入 LLM 微调技术。

RAG 这一新兴领域经历了快速发展，但尚未形成系统的综合研究，以阐明其更广泛的发展轨迹。本综述旨在填补这一空白，通过绘制 RAG 流程、规划其发展历程和预期的未来路径，重点关注 RAG 与 LLM 的整合。本文同时考察技术范式和研究方法，总结了 100 多项 RAG 研究中的三大主要研究范式，并分析了“检索”核心阶段的关键技术。

“生成”和“增强”。另一方面，当前的研究往往侧重于方法，缺乏对如何评估 RAG 的分析和总结。本文全面回顾了适用于 RAG 的后续任务、数据集、基准和评估方法。总而言之，本文旨在细致地汇编和分类基础技术概念、历史发展以及 LLM 之后出现的 RAG 方法和应用范围。本文旨在帮助读者和专业人士对大型模型和 RAG 有一个详细而结构化的理解。本文旨在阐明检索增强技术的演变，评估各种方法在其各自环境下的优缺点，并推测未来的趋势和创新。

我们的贡献如下：

·在本次调查中，我们对最先进的 RAG 方法进行了全面而系统的回顾，描述了其通过包括朴素 RAG 在内的范式的演变，

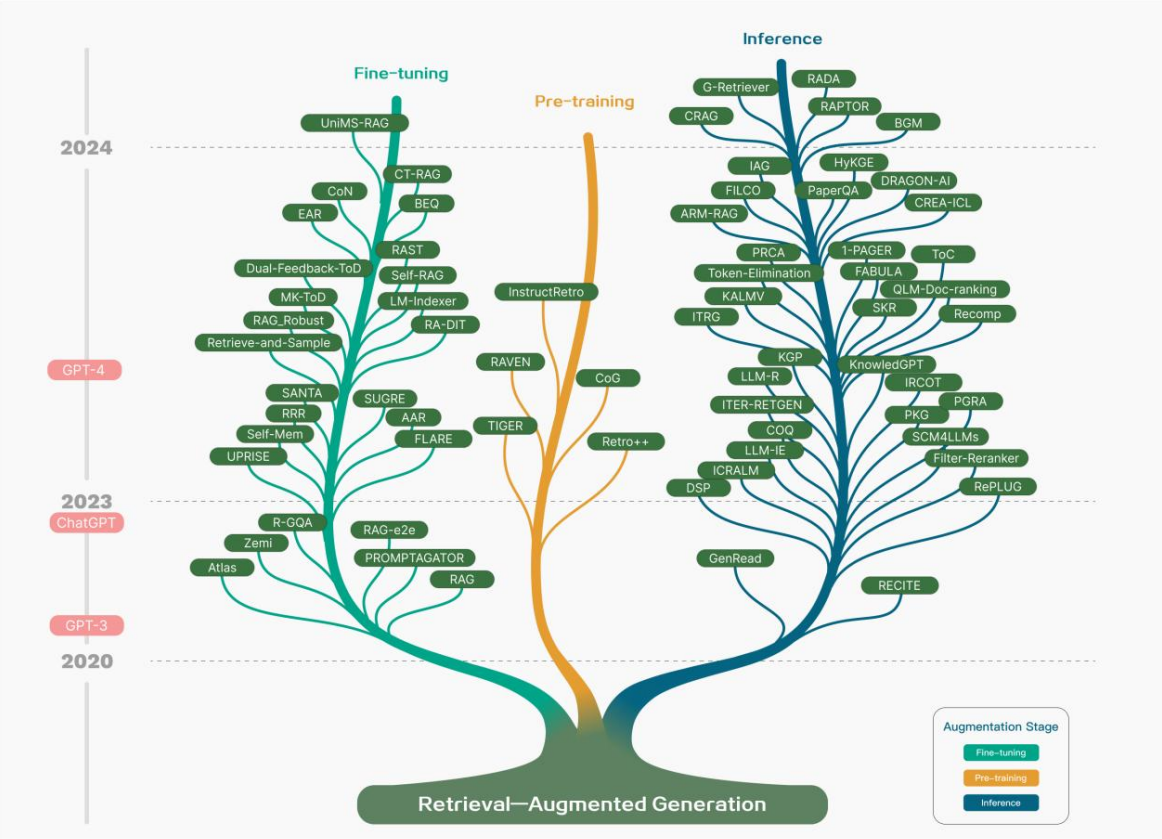

图 1. RAG 研究技术树。RAG 的研究主要包括预训练、微调和推理三个阶段。随着 LLM 的兴起，RAG 的研究最初侧重于利用 LLM 强大的上下文学习能力，主要集中在推理阶段。后续研究逐渐深入，并逐渐与 LLM 的微调进行更多融合。研究人员也在探索如何通过检索增强技术来增强预训练阶段的语言模型。

高级 RAG 和模块化 RAG。本综述将 RAG 研究的更广泛范围置于 LLM 领域中。

·我们确定并讨论了 RAG 流程中不可或缺的核心技术，特别关注“检索”、“生成”和“增强”方面，并深入研究它们的协同作用，阐明这些组件如何错综复杂地协作以形成一个有凝聚力且有效的 RAG 框架。

·我们总结了当前的评估方法

RAG 涵盖 26 项任务和近 50 个数据集，概述了评估目标和指标，以及当前的评估基准和工具。此外，我们还预测了 RAG 的未来发展方向，并强调了应对当前挑战的潜在改进措施。

论文主要内容如下：第二部分介绍 RAG 的核心概念和当前范式。接下来的三部分分别探讨核心组件 “检索”、“生成”和“增强”。第三部分重点介绍检索中的优化方法，包括索引、查询和嵌入优化。第四部分重点介绍检索后流程和生成中的 LLM 微调。第五部分分析了三个增强过程。第六部分重点介绍 RAG 的下游任务和评估系统。第七部分主要讨论 RAG 目前面临的挑战。

面临的问题以及未来的发展方向。最后，在第八部分对全文进行总结。

II. RAG概述

图 2 展示了 RAG 的典型应用。

在这里，一位用户向 ChatGPT 提出了一个关于最近被广泛讨论的新闻的问题。由于 ChatGPT 依赖于预训练数据，它最初缺乏提供最新动态的能力。RAG 通过从外部数据库获取和整合知识来弥补这一信息缺口。在这种情况下，它会收集与用户查询相关的新闻文章。这些文章与原始问题相结合，形成了一个全面的提示，使 LLM 能够生成一个信息充分的答案。

RAG 研究范式在不断发展，我们将其分为三个阶段：Naive RAG、Advanced RAG 和 Modular RAG，如图 3 所示。尽管 RAG 方法具有成本效益并且超越了原生 LLM 的性能，但它们也表现出一些局限性。

Advanced RAG 和 Modular RAG 的开发就是针对 Naive RAG 的这些特定缺陷而做出的回应。

A. 幼稚的 RAG

朴素 RAG 研究范式代表了最早的方法论，在

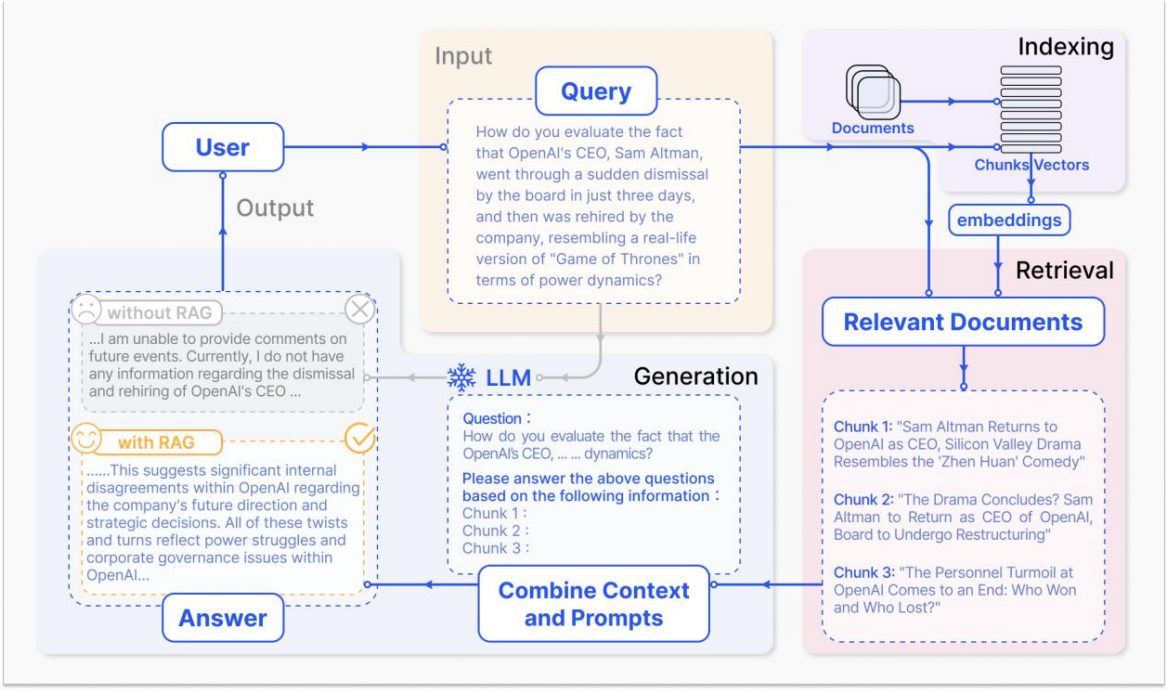

图 2. RAG 流程应用于问答系统的典型示例。该流程主要包含 3 个步骤。1)索引。将文档拆分成块，编码成向量，并存储在向量数据库中。2)检索。根据语义相似度检索与问题最相关的 Top k 个块。3）生成。将原始问题和检索到的块一起输入 LLM 以生成最终答案。

ChatGPT 的广泛采用。Naive RAG 遵循传统的流程，包括索引、检索和生成，也称为“检索-阅读”框架 [7]。

索引构建始于清理和提取 PDF、HTML、Word 和 Markdown 等多种格式的原始数据，然后将其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、更易于理解的块。然后，使用嵌入模型将块编码为向量表示，并存储在向量数据库中。此步骤对于在后续检索阶段实现高效的相似性搜索至关重要。

检索。收到用户查询后，RAG 系统采用索引阶段使用的相同编码模型将查询转换为向量表示。

然后计算查询向量和索引语料库中的块向量之间的相似度得分。

系统会优先检索与查询最相似的前 K 个块，这些块随后将用作提示中的扩展上下文。

生成。提出的查询和选定的文档被合成为一个连贯的提示，大型语言模型负责针对该提示制定响应。该模型的回答方法可能因任务特定的标准而异，允许它利用其固有的参数知识，或将其响应限制在所提供文档中包含的信息范围内。在对话正在进行的情况下，任何现有的对话历史记录都可以集成到提示中，从而使模型能够有效地进行多轮对话交互。

然而，Naive RAG 遇到了明显的缺点：

检索挑战。检索阶段通常会面临精确度和召回率的挑战，从而导致选择错位或不相关的块，并丢失关键信息。

生成难题。在生成回复时，模型可能会面临“幻觉”问题，即生成的内容与检索到的上下文不符。此阶段还可能出现输出结果不相关、毒性或偏差，从而降低回复的质量和可靠性。

数据增强的障碍。将检索到的信息与不同的任务整合起来可能颇具挑战性，有时会导致输出脱节或不连贯。当从多个来源检索类似信息时，该过程还可能遇到冗余，导致重复响应。确定不同段落的重要性和相关性，并确保风格和语调的一致性，进一步增加了复杂性。

面对复杂的问题，基于原始查询的单一检索可能不足以获取足够的上下文信息。

此外，人们担心生成模型可能过度依赖增强信息，导致输出只是简单地回应检索到的内容，而不会添加有见地或合成的信息。

B. 高级 RAG

Advanced RAG 引入了特定的改进，以克服 Naive RAG 的局限性。它专注于提高检索质量，采用了预检索和后检索策略。为了解决索引问题，Advanced RAG 通过使用滑动窗口方法、细粒度分割和元数据的整合来改进其索引技术。此外，它还结合了多种优化方法来简化检索流程 [8]。

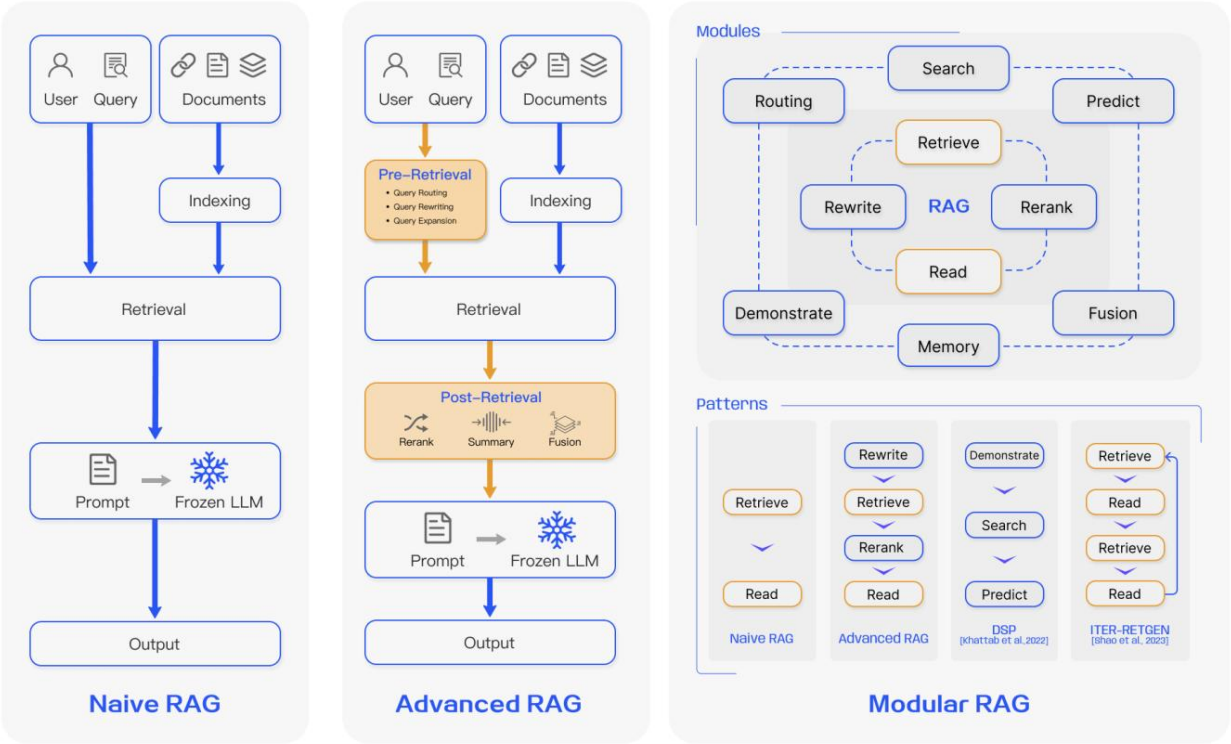

图 3. RAG 三种范式的比较。(左)Naive RAG 主要由索引、检索和生成三部分组成。(中)
高级 RAG 围绕预检索和后检索提出了多种优化策略,其流程与 Naive RAG 类似,但仍然遵循链式结构。(右图)模块化 RAG 继承并发展了之前的范式,整体上展现出更高的灵活性。这体现在引入多个特定的功能模块以及替换现有模块方面。整体流程不仅限于顺序检索和生成,还包含迭代检索和自适应检索等方法。

预检索过程。在此阶段,主要关注优化索引结构和原始查询。

优化索引的目标是提升被索引内容的质量。这涉及的策略包括:增强数据粒度、优化索引结构、添加元数据、对齐优化以及混合检索。而查询优化的目标是使用户的原始问题更清晰,更适合检索任务。常用的方法包括查询重写、查询转换、查询扩展等技术[7], [9]–[11]。

检索后处理。检索到相关上下文后,将其与查询有效地集成至关重要。检索后处理的主要方法包括对词块进行重新排序和上下文压缩。对检索到的信息进行重新排序,将最相关的内容重新定位到提示的边缘,是一项关键策略。这一概念已在 LlamaIndex2 、LangChain3和 HayStack [12] 等框架中得到实现。

将所有相关文档直接输入 LLM 可能会导致信息过载,用不相关的内容稀释对关键细节的关注。为了缓解这种情况,检索后的工作应集中于选择必要的信息、强调关键部分以及缩短要处理的上下文。

2https://www.llamaindex.ai
3https://www.langchain.com/

C. 模块化 RAG

模块化 RAG 架构超越了前两种 RAG 范式,增强了适应性和多功能性。它融合了多种策略来改进其组件,例如添加用于相似性搜索的搜索模块,以及通过微调来优化检索器。为了应对特定挑战,引入了诸如重组 RAG 模块 [13] 和重新排列 RAG 流水线 [14] 等创新技术。向模块化 RAG 方法的转变正日趋普遍,它支持跨组件的顺序处理和集成端到端训练。尽管模块化 RAG 独具特色,但它建立在高级 RAG 和朴素 RAG 的基本原理之上,体现了 RAG 家族的进步和完善。

1)新模块:模块化 RAG 框架引入了额外的专用组件,以增强检索和处理能力。搜索模块可适应特定场景,使用 LLM 生成的代码和查询语言,支持跨搜索引擎、数据库和知识图谱等各种数据源进行直接搜索 [15]。RAG-Fusion 通过采用多查询策略,将用户查询扩展到不同的视角,突破了传统搜索的局限性,并利用并行向量搜索和智能重排序来发现显式知识和变革性知识 [16]。记忆模块利用 LLM 的内存来指导检索,创建一个无限的内存池,

通过迭代自我增强使文本与数据分布更紧密地对齐 [17],[18]。RAG 系统中的路由会浏览不同的数据源,为查询选择最佳路径,无论该查询涉及摘要、特定数据库搜索还是合并不同的信息流 [19]。预测模块旨在通过 LLM 直接生成上下文来减少冗余和噪声,确保相关性和准确性 [13]。最后,任务适配器模块使 RAG 适应各种下游任务,自动对零样本输入进行快速检索,并通过少样本查询生成创建特定于任务的检索器 [20],[21]。这种综合方法不仅简化了检索过程,而且显着提高了检索信息的质量和相关性,从而以更高的精度和灵活性满足了各种任务和查询的需求。

2)新模式:模块化 RAG 提供了卓越的适应性,允许通过模块替换或重新配置来应对特定挑战。这打破了 Naive 和 Advanced RAG 的固定结构(以简单的“检索”和“读取”机制为特征)。此外,模块化 RAG 通过集成新模块或调整现有模块之间的交互流程来扩展这种灵活性,从而增强了其在不同任务中的适用性。

诸如重写-检索-阅读 [7] 模型之类的创新利用 LLM 的功能,通过重写模块和 LM 反馈机制来改进检索查询,从而更新重写模型,从而提高任务性能。

类似地,诸如 Generate-Read [13] 之类的方法用 LLM 生成的内容取代了传统的检索,而 Recite-Read [22] 则强调从模型权重中进行检索,从而增强了模型处理知识密集型任务的能力。

混合检索策略集成了关键词、语义和向量搜索,以满足多样化的查询需求。此外,采用子查询和假设文档嵌入 (HyDE) [11] 旨在通过关注生成的答案与真实文档之间的嵌入相似性来提高检索相关性。

模块排列和交互的调整,例如演示-搜索-预测 (DSP) [23] 框架和 ITER-RETGEN [14] 的迭代检索-读取-检索-读取流程,展示了如何动态地使用模块输出来增强另一个模块的功能,体现了对增强模块协同作用的深刻理解。

模块化 RAG Flow 的灵活编排展现了通过 FLARE [24] 和 Self-RAG [25] 等技术进行自适应检索的优势。这种方法通过根据不同场景评估检索的必要性,超越了固定的 RAG 检索流程。灵活架构的另一个优势是,RAG 系统可以更轻松地与其他技术(例如微调或强化学习)集成 [26]。例如,这可以涉及微调检索器以获得更好的检索结果,微调生成器以获得更个性化的输出,或参与协作微调 [27]。

D. RAG 与微调 LLM 的增强由于其

日益普及而引起了广泛关注。在优化中

作为LLM方法,RAG经常与微调(FT)和快速工程进行比较。如图4所示,每种方法都有其独特的特点。我们使用象限图从两个维度来说明三种方法之间的差异:外部知识需求和模型适配需求。快速工程充分利用模型的固有能力,最大限度地减少了对外部知识和模型适配的需求。RAG 可以比作为模型提供定制的信息检索教材,非常适合精确的信息检索任务。相比之下,FT就像学生随着时间的推移内化知识,适用于需要复制特定结构、样式或格式的场景。

RAG 在动态环境中表现出色,能够提供实时知识更新,并有效利用外部知识源,且具有高度可解释性。然而,它存在更高的延迟,并且在数据检索方面存在伦理方面的考量。另一方面,FT 则更加静态,需要重新训练才能进行更新,但可以对模型的行为和风格进行深度定制。它需要大量的计算资源来准备和训练数据集,虽然它可以减少幻觉,但在处理不熟悉的数据时可能会面临挑战。

[28] 对不同主题的各种知识密集型任务的性能进行了多次评估,结果表明,虽然无监督微调有所提升,但 RAG 的表现始终优于无监督微调,无论是针对训练过程中遇到的现有知识还是全新知识。此外,研究还发现,LLM 难以通过无监督微调来学习新的事实信息。RAG 和 FT 之间的选择取决于应用环境中对数据动态、定制化和计算能力的具体需求。RAG 和 FT 并非相互排斥,而是可以相互补充,在不同层面上增强模型的能力。

在某些情况下,两者结合使用可能会获得最佳性能。涉及 RAG 和 FT 的优化过程可能需要多次迭代才能达到令人满意的结果。

III.检索

在RAG的背景下,高效地从数据源中检索相关文档至关重要。这涉及到几个关键问题,例如检索源、检索粒度、检索的预处理以及相应嵌入模型的选择。

A. 检索来源

RAG依赖外部知识来增强LLM,而检索源的类型和检索单元的粒度都会影响最终的生成结果。

1)数据结构:最初,文本是检索的主流来源。随后,检索来源扩展到包括半结构化数据(PDF)和结构化数据(知识图谱,KG)以用于增强。除了从原始的外部来源检索外,近年来,利用LLM自身生成的内容进行检索和增强的研究也日益增多。

表一
RAG方法总结

方法	检索来源	检索数据类型	检索粒度	增强阶段	检索过程
重心 [29]	维基百科	文本	短语	预训练	迭代
DenseX [30]	趣闻维基	文本	主张	推理	一次
耳 [31]	数据集库	文本	句子	调优	一次
价格 [20]	数据集库	文本	句子	调优	一次
案情 [32]	数据集库	文本	句子	调优	一次
自我记忆 [17]	数据集库	文本	句子	调优	迭代
耀斑 [24]	搜索引擎,维基百科	文本	句子	调优	自适应
PGRA [33]	维基百科	文本	句子	推理	一次
斐尔可 [34]	维基百科	文本	句子	推理	一次
工作 [35]	数据集库	文本	句子	推理	一次
过滤-重新排序 [36]	合成数据集	文本	句子	推理	一次
R-GQA [37]	数据集库	文本	句子对	调优	一次
法学硕士（研究） [38]	数据集库	文本	句子对	推理	迭代
老虎 [39]	数据集库	文本	项目基础	预训练	一次
LM 索引器 [40]	数据集库	文本	项目基础	调优	一次
贝克 [9]	数据集库	文本	项目基础	调优	一次
CT-RAG [41]	合成数据集	文本	项目基础	调优	一次
阿特拉斯 [42]	维基百科,常见爬虫	文本	块	预训练	迭代
渡鸦 [43]	维基百科	文本	块	预训练	一次
复古++ [44]	预训练语料库	文本	块	预训练	迭代
复古指令 [45]	预训练语料库	文本	块	预训练	迭代
风险承担率 [7]	搜索引擎	文本	块	调优	一次
RA-e2e [46]	数据集库	文本	块	调优	一次
序幕 [21]	带来	文本	块	调优	一次
AAR [47]	MSMARCO,维基百科	文本	块	调优	一次
RA-DIT [27]	常见爬行,维基百科	文本	块	调优	一次
RAG-Robust [48]	维基百科	文本	块	调优	一次
RA-长篇 [49]	数据集库	文本	块	调优	一次
钻 [50]	维基百科	文本	块	调优	一次
自 RAG [25]	维基百科	文本	块	调优	自适应
背景音乐 [26]	维基百科	文本	块	推理	一次
辅酶Q [51]	维基百科	文本	块	推理	迭代
令牌消除[52]	维基百科	文本	块	推理	一次
论文问答	Arxiv、在线数据库、PubMed	文本	块	推理	迭代
NoiseRAG [54]	趣闻维基	文本	块	推理	一次
国际航空集团 [55]	搜索引擎,维基百科	文本	块	推理	一次
来自MIRACL [56]	维基百科	文本	块	推理	一次
目录 [57]	搜索引擎,维基百科	文本	块	推理	递归
SKR [58]	数据集库,维基百科	文本	块	推理	自适应
ITRG [59]	维基百科	文本	块	推理	迭代
RAG-长上下文 [60]	数据集库	文本	块	推理	一次
ITER-RETGEN [14]	维基百科	文本	块	推理	迭代
IRCoT [61]	维基百科	文本	块	推理	递归
LLM-知识边界 [62]	维基百科	文本	块	推理	一次
猛禽 [63]	数据集库	文本	块	推理	递归
背诵 [22]	法学硕士	文本	块	推理	一次
ICRALM [64]	桩,维基百科	文本	块	推理	迭代
检索和采样 [65]	数据集库	文本	文档	调优	一次
地球 [66]	C4	文本	文档	调优	一次
峭壁 [67]	Arxiv	文本	文档	推理	一次
单页 [68]	维基百科	文本	文档	推理	迭代
PRCA [69]	数据集库	文本	文档	推理	一次
QLM-文档排名 [70]	数据集库	文本	文档	推理	一次
重新计算 [71]	维基百科	文本	文档	推理	一次
DSP [23]	维基百科	文本	文档	推理	迭代
重新插拔 [72]	桩	文本	文档	推理	一次
ARM-RAG [73]	数据集库	文本	文档	推理	迭代
GenRead [13]	法学硕士	文本	文档	推理	迭代
UniMS-RAG [74]	数据集库	文本	多	调优	一次
CREA-ICL [19]	数据集库	跨语言,文本	句子	推理	一次
包 [75]	法学硕士	表格、文本	块	推理	一次
圣诞老人 [76]	数据集库	代码、文本	物品	预训练	一次
涌动 [77]	游离碱	公斤	子图	调优	一次
MK-ToD [78]	数据集库	公斤	实体	调优	一次
双反馈ToD[79]	数据集库	公斤	实体序列	调优	一次
知识GPT [15]	数据集库	公斤	三胞胎	推理	多时间
故事 [80]	数据集基础,图	公斤	实体	推理	一次
HyKGE [81]	甲基化KG	公斤	实体	推理	一次
卡尔梅克航空飞机 [82]	维基百科	公斤	三胞胎	推理	迭代
羅格 [83]	游离碱	公斤	三胞胎	推理	迭代
G-猎犬 [84]	数据集库	文本图	子图	推理	一次

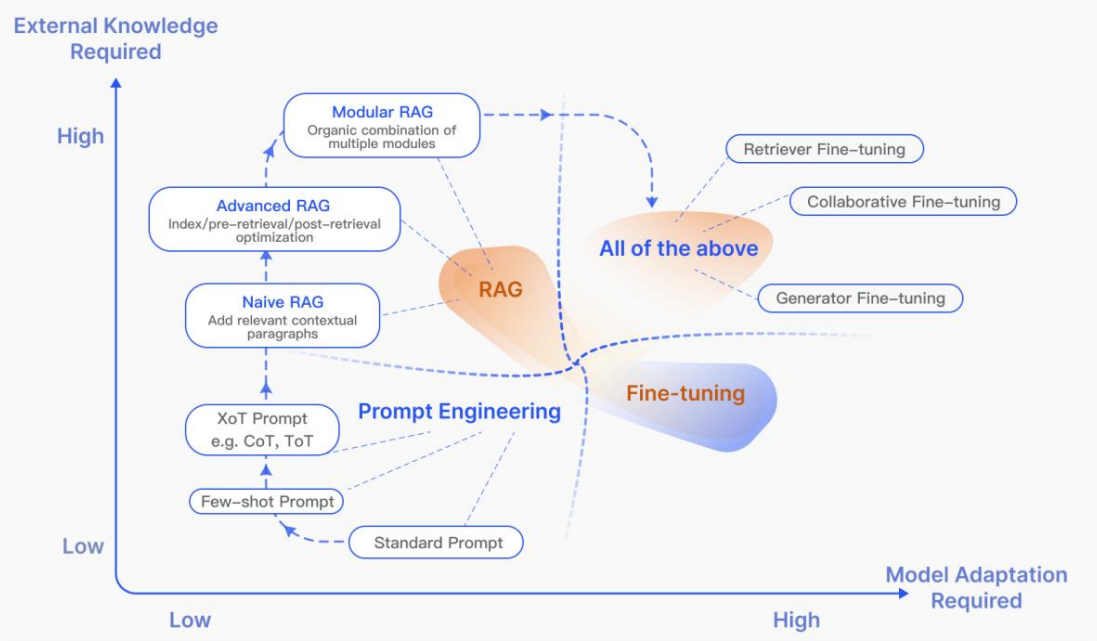

图 4. RAG 在“所需外部知识”和“所需模型自适应”方面与其他模型优化方法进行了比较。快速工程 (Prompt Engineering) 要求对模型和外部知识的修改较少,专注于充分利用 LLM 自身的能力。另一方面,微调 (Fine-tuning) 则需要进一步训练模型。在 RAG（朴素 RAG)的早期阶段,对模型修改的需求较低。随着研究的进展,模块化 RAG 与微调技术的集成度不断提高。

非结构化数据（例如文本)是最广泛使用的检索来源,主要通过语料库收集。对于开放域问答 (ODQA) 任务,主要的检索来源是 Wikipedia Dump（目前主要的)(2017)、DPR5 （包括 HotpotQA（2018 年 10 月 1 日-12 月)在内的 20 个版本）。除了百科全书数据外,常见的非结构化数据还包括跨语言文本（例如医学 [67] 和法律领域 [29]）。[4] [19] 和领域特定数据

半结构化数据通常指包含文本和表格信息的数据,例如 PDF。处理半结构化数据对传统的 RAG 系统提出了挑战,主要有两个原因。首先,文本拆分过程可能会无意中将表拆分开,导致检索过程中数据损坏。其次,将表合并到数据中会使语义相似性搜索变得复杂。处理半结构化数据时,一种方法是利用 LLM 的代码功能对数据库中的表执行 Text-2-SQL 查询,例如 TableGPT [85]。

或者,可以将表格转换为文本格式,以便使用基于文本的方法进行进一步分析 [75]。然而,这两种方法都不是最佳解决方案,这表明该领域仍有大量的研究机会。

结构化数据,例如知识图谱 (KG) [86],通常经过验证,可以提供更精准的信息。KnowledGPT [15] 生成知识库搜索查询,并将知识存储在个性化库中,从而增强了 RAG 模型的知识丰富度。为了弥补 LLM 在理解和回答文本图谱问题方面的局限性,G-Retriever [84] 集成了图神经网络（GNN）、LLM 和 RAG,通过 LLM 的软提示增强图理解和问答能力,并利用奖品收集斯坦纳树 (PCST) 优化问题进行有针对性的图检索。相反,构建、验证和维护结构化数据库需要额外的努力。相反,构建、验证和维护结构化数据库需要额外的努力。

LLM 生成的内容。为了解决 RAG 中外部辅助信息的局限性,一些研究集中于利用 LLM 的内部知识。SKR [58] 将问题分为已知或未知,并选择性地应用检索增强功能。GenRead [13] 用 LLM 生成器取代了检索器,发现 LLM 生成的上下文通常包含更准确的答案,因为它与因果语言模型的预训练目标更加契合。

Selfmem [17] 使用检索增强生成器迭代地创建一个无界的内存池,使用内存选择器来选择作为原始问题的对偶问题的输出,从而自我增强生成模型。

这些方法强调了 RAG 中创新数据源利用的广度,致力于提高模型性能和任务效率。

2)检索粒度:除了检索源的数据格式之外,另一个重要因素是检索数据的粒度。粗粒度的检索单元理论上可以提供与问题更相关的信息,但它们也可能包含冗余内容,这可能会分散下游任务中检索器和语言模型的注意力[50],[87]。另一方面,细粒度的检索单元粒度会增加检索的负担,并且无法保证语义完整性和所需知识的获取。选择

4https://hotpotqa.github.io/wiki-readme.html
5https://github.com/facebookresearch/DPR

推理过程中适当的检索粒度可以成为一种简单有效的策略,以提高密集检索器的检索和下游任务的性能。

在文本中,检索粒度从细到粗,包括标记（Token）、短语（Phrase）、句子（Sentence）、命题（Proposition）、词块（Chunk）、文档（Document）。其中,DenseX [30] 提出了使用命题作为检索单位的概念。命题被定义为文本中的原子表达式,每个命题封装一个独特的事实片段,并以简洁、自包含的自然语言格式呈现。这种方法旨在提高检索的准确率和相关性。在知识图谱 (KG) 中,检索粒度包括实体、三元组和子图。

检索的粒度也可以适应下游任务,例如在推荐任务中检索项目 ID [40] 和句子对 [38]。详细信息如表 1 所示。

B.索引优化

在索引阶段,文档会被处理、切分,并转化为Embeddings,存储到向量数据库中。索引构建的质量决定了在检索阶段能否获取正确的上下文。

1)分块策略:最常用的方法是将文档按固定数量的标记（例如 100、256、512)拆分成块 [88]。较大的块可以捕获更多上下文,但也会产生更多噪声,需要更长的处理时间和更高的成本。较小的块虽然可能无法完全传达必要的上下文,但它们的噪声较少。然而,分块会导致句子内部的截断,这促使人们优化递归拆分和滑动窗口方法,通过合并多个检索过程中的全局相关信息来实现分层检索 [89]。然而,这些方法仍然无法在语义完整性和上下文长度之间取得平衡。因此,有人提出了类似 Small2Big 的方法,其中句子（小）被用作检索单位,前后句子作为（大)上下文提供给 LLM [90]。

2)元数据附件:数据块可以添加元数据信息,例如页码、文件名、作者、类别时间戳等。随后,可以根据这些元数据过滤检索,从而限制检索范围。

在检索过程中为文档时间戳分配不同的权重可以实现时间感知的RAG,确保知识的新鲜度并避免信息过时。

除了从原始文档中提取元数据外,还可以人工构建元数据。例如,添加段落摘要,以及引入假设性问题。这种方法也称为逆向 HyDE。具体而言,利用 LLM 生成文档可以回答的问题,然后在检索过程中计算原始问题与假设问题的相似度,以缩小问题与答案之间的语义差距。

3)结构化索引:增强信息检索效率的有效方法是建立文档的层次结构。通过构建层次结构,RAG系统可以加快相关数据的检索和处理速度。

分层索引结构。文件按父子关系排列,并以块为单位进行链接。数据摘要存储在每个节点上,有助于快速遍历数据,并协助 RAG 系统确定需要提取的块。这种方法还可以减轻由块提取问题引起的错觉。

知识图谱索引。利用知识图谱 (KG) 构建文档的层级结构有助于保持一致性。它描绘了不同概念和实体之间的联系,显著降低了出现错觉的可能性。另一个优势是将信息检索过程转化为 LLM 能够理解的指令,从而提高知识检索的准确性,并使 LLM 能够生成上下文连贯的响应,最终提高 RAG 系统的整体效率。为了捕捉文档内容和结构之间的逻辑关系,知识图谱 [91] 提出了一种使用知识图谱 (KG) 在多文档之间构建索引的方法。该知识图谱由节点（表示文档中的段落或结构,例如页面和表格)和边（表示段落之间的语义/词汇相似性或文档结构中的关系)组成,有效地解决了多文档环境下的知识检索和推理问题。

C.查询优化

Naive RAG 的主要挑战之一是它直接依赖用户的原始查询作为检索的基础。提出一个精准清晰的问题非常困难,而轻率的查询会导致检索效果不佳。

有时,问题本身很复杂,语言组织也不够完善。另一个难点在于语言复杂性和歧义性。语言模型在处理专业词汇或具有多重含义的模糊缩写时常常会遇到困难。例如,它们可能无法辨别“LLM”指的是大型语言模型还是法律语境中的法学硕士。

1)查询扩展:将单个查询扩展为多个查询可以丰富查询的内容,提供进一步的上下文来解决任何缺乏具体细微差别的问题,从而确保生成的答案的最佳相关性。

多查询。通过采用快速工程,通过 LLM 扩展查询,这些查询可以并行执行。查询的扩展并非随机的,而是经过精心设计的。

子查询。子问题规划的过程表示生成必要的子问题,以便将原始问题组合起来并完整地回答原始问题。添加相关上下文的过程原则上类似于查询扩展。具体来说,可以使用从最少到最多的提示方法将复杂问题分解为一系列更简单的子问题[92]。

验证链（CoVe）。扩展查询经过LLM验证,以达到减少幻觉的效果。经过验证的扩展查询通常具有更高的信度[93]。

2)查询转换:核心概念是根据转换后的查询而不是用户的原始查询来检索块。

查询重写。原始查询并不总是适合 LLM 检索,尤其是在实际场景中。因此,我们可以提示 LLM 重写查询。除了使用 LLM 进行查询重写之外,还可以使用专门的小型语言模型,例如 RRR(重写-检索-阅读)[7]。淘宝中对查询重写方法的实现,即 BEQUE [9],显著提升了长尾查询的召回率,从而带来了 GMV 的提升。

另一种查询转换方法是使用提示工程,让 LLM 基于原始查询生成查询,以供后续检索。HyDE [11] 构建了假设文档(对原始查询的假设答案)。它侧重于从答案到答案的嵌入相似性,而不是寻求问题或查询本身的嵌入相似性。

采用Step-back Prompting方法[10],将原始Query抽象为高阶概念问题(Step-back question)。在RAG系统中,Step-back question和原始Query都会用于检索,并将两者的结果作为语言模型答案生成的基础。

3)查询路由:根据不同的查询,路由到不同的RAG管道,适用于为适应不同场景而设计的多功能RAG系统。

元数据路由器/过滤器。第一步是从查询中提取关键字(实体),然后根据块内的关键字和元数据进行过滤,以缩小搜索范围。

语义路由器是另一种利用查询语义信息的路由方法。具体方法请参见语义路由器,它结合了语义方法和基于元数据的方法,以增强查询路由。[6]。当然,混合路由

D. 嵌入

在 RAG 中,检索是通过计算问题和文档块的嵌入之间的相似度(例如余弦相似度)来实现的,其中嵌入模型的语义表示能力起着关键作用。这主要包括稀疏编码器(BM25)和密集检索器(BERT 架构预训练语言模型)。最近的研究引入了诸如 AnglE、Voyage、BGE 等 [94]–[96] 等突出的嵌入模型,这些模型受益于多任务指令调优。Hugging Face 的 MTEB 排行榜嵌入模型涵盖 8 个任务,覆盖 58 个数据集。此外,C-MTEB 专注于中文能力,涵盖 6 个任务和 35 个数据集。对于“使用哪种嵌入模型”没有一刀切的答案。但是,某些特定模型更适合特定的用例。

[7] 评估

1) 混合检索:稀疏和密集嵌入方法分别捕捉不同的相关性特征,并可以通过利用互补的相关性信息相互补充。例如,稀疏检索模型可以用于为训练密集检索模型提供初始搜索结果。此外,预训练语言模型 (PLM) 可用于学习术语权重,从而增强稀疏检索。具体而言,它还表明稀疏检索模型可以增强密集检索模型的零样本检索能力,并帮助密集检索器处理包含稀有实体的查询,从而提高鲁棒性。

2)微调嵌入模型:在上下文与预训练语料库有显著偏差的情况下,特别是在医疗保健、法律实践和其他充斥着专有术语的领域等高度专业化的学科中,在您自己的领域数据集上微调嵌入模型对于缓解这种差异至关重要。

除了补充领域知识之外,微调的另一个目的是使检索器和生成器对齐,例如使用 LLM 的结果作为微调的监督信号,称为 LSR(LM-supervised Retriever)。

PROMPTAGATOR [21] 利用 LLM 作为少样本查询生成器来创建特定任务的检索器,从而解决了监督微调中的挑战,尤其是在数据稀缺领域。另一种方法 LLM-Embedder [97] 利用 LLM 为多个下游任务生成奖励信号。检索器使用两种类型的监督信号进行微调:数据集的硬标签和来自 LLM 的软奖励。这种双信号方法促进了更有效的微调过程,使嵌入模型能够适应不同的下游应用。REPLUG [72] 利用检索器和 LLM 来计算检索文档的概率分布,然后通过计算 KL 散度进行监督训练。这种直接有效的训练方法通过使用 LM 作为监督信号来提升检索模型的性能,从而无需特定的交叉注意力机制。

此外,受到 RLHF(人类反馈强化学习)的启发,利用基于 LM 的反馈通过强化学习来强化检索器。

E. 适配器微调

模型可能会带来挑战,例如通过 API 集成功能或解决由有限的本地计算资源引起的限制。

因此,一些方法选择加入外部适配器来帮助对齐。

为了优化 LLM 的多任务能力,UP-RISE [20] 训练了一个轻量级的提示检索器,它可以自动从预先构建的提示池中检索适合给定零样本任务输入的提示。AAR(增强自适应检索器)[47] 引入了一个通用适配器,旨在适应多个下游任务。

PRCA [69] 添加了可插拔的奖励驱动上下文适配器,以增强特定任务的性能。BGM [26] 保持检索器和 LLM 固定不变,并在两者之间训练一个桥接的 Seq2Seq 模型。桥接模型旨在将检索到的信息转换为 LLM 可以有效处理的格式,使其不仅可以重新排序,还可以动态地为每个查询选择段落,并可能采用更高级的策略,例如重复。此外,PKG

6https://github.com/aurelio-labs/semantic-router
7https://huggingface.co/spaces/mteb/leaderboard

提出了一种通过指令微调将知识集成到白盒模型中的创新方法 [75]。在该方法中,检索器模块直接被替换,根据查询生成相关文档。该方法有助于解决微调过程中遇到的困难,并提高模型性能。

四、世代

检索后,将检索到的所有信息直接输入LLM来回答问题并不是一个好的做法。

下面将从检索内容的调整和LLM的调整两个角度来介绍调整。

A. 语境策划

冗余信息会干扰LLM的最终生成,过长的上下文也会导致LLM出现“迷失在中间”的问题[98]。与人类一样,LLM往往只关注长文本的开头和结尾,而忽略中间部分。因此,在RAG系统中,我们通常需要对检索到的内容进行进一步处理。

1) 重排序:重排序从根本上重新排序文档块,以优先突出显示最相关的结果,从而有效地缩减整体文档池,在信息检索中发挥双重作用,既充当增强器,又充当过滤器,为更精确的语言模型处理提供精炼的输入 [70]。重排序可以使用基于规则的方法（这些方法依赖于预定义的指标,例如多样性、相关性和 MRR）,也可以使用基于模型的方法（例如 BERT 系列中的编码器-解码器模型,例如 SpanBERT）,专门的重排序模型（例如 Cohere rerank 或 bge-raranker-large）,以及通用的大型语言模型（例如 GPT [12]、[99]）。

2) 上下文选择/压缩:RAG 过程中一个常见的误解是,人们认为检索尽可能多的相关文档并将它们串联起来形成一个冗长的检索提示是有益的。然而,过多的上下文可能会引入更多噪音,削弱 LLM 对关键信息的感知。

(长) LLMLingua [100],[101] 利用小型语言模型 (SLM)（例如 GPT-2 Small 或 LLaMA-7B)来检测和删除不重要的标记,将其转换为人类难以理解但 LLM 可以很好理解的形式。这种方法提出了一种直接实用的快速压缩方法,无需对 LLM 进行额外的训练,同时兼顾了语言完整性和压缩率。PRCA 通过训练信息提取器解决了这个问题 [69]。类似地,RECOMP 采用了类似的方法,通过使用对比学习来训练信息压缩器 [71]。每个训练数据点包含一个正样本和五个负样本,并且编码器在整个过程中使用对比损失进行训练 [102]。

除了压缩上下文之外,减少文档数量也有助于提高模型答案的准确性。Ma 等人 [103] 提出了“Filter-Reranker”范式,它结合了 LLM 和 SLM 的优势。

在这一范式中,SLM 充当过滤器,而 LLM 充当重新排序代理。研究表明,指示 LLM 重新排列 SLM 识别出的具有挑战性的样本,可以显著提升各种信息提取 (IE) 任务的表现。另一种直接有效的方法是让 LLM 在生成最终答案之前评估检索到的内容。这使得 LLM 能够通过 LLM 评审过滤掉相关性较差的文档。例如,在“Chatlaw [104]”一文中,LLM 被提示对所引用的法律条款进行自我建议,以评估其相关性。

B. LLM 微调

根据场景和数据特征在LLM上进行有针对性的微调可以取得更好的效果。这也是使用本地LLM的最大优势之一。当LLM缺乏特定领域的数据时,可以通过微调为LLM提供额外的知识。Huggingface的微调数据也可以作为初始步骤。

微调的另一个好处是能够调整模型的输入和输出。例如,它可以让 LLM 适应特定的数据格式,并根据指示生成特定风格的响应 [37]。对于涉及结构化数据的检索任务,SANTA 框架 [76] 实施了三部分训练方案,以有效地封装结构和语义的细微差别。初始阶段专注于检索器,利用对比学习来优化查询和文档向量。

通过强化学习将法学硕士 (LLM) 的输出与人类或检索器的偏好相匹配是一种潜在的方法。例如,手动注释最终生成的答案,然后通过强化学习提供反馈。

除了与人类偏好保持一致之外,还可以与微调模型和检索器的偏好保持一致 [79]。当无法使用强大的专有模型或参数更大的开源模型时,一种简单有效的方法是提取更强大的模型（例如 GPT-4）。LLM 的微调也可以与检索器的微调相协调,以协调偏好。一种典型的方法,例如 RA-DIT [27],使用 KL 散度来对齐检索器和生成器之间的评分函数。

V. RAG中的增强过程

在RAG领域,标准做法通常涉及单一（一次性)检索步骤,然后进行生成。这可能导致效率低下,有时对于需要多步推理的复杂问题通常不够用,因为它提供的信息范围有限[105]。针对这个问题,许多研究已经优化了检索流程,我们在图5中进行了总结。

A. 迭代检索

迭代检索是根据初始查询和迄今为止生成的文本反复搜索知识库的过程,从而提供更全面的知识

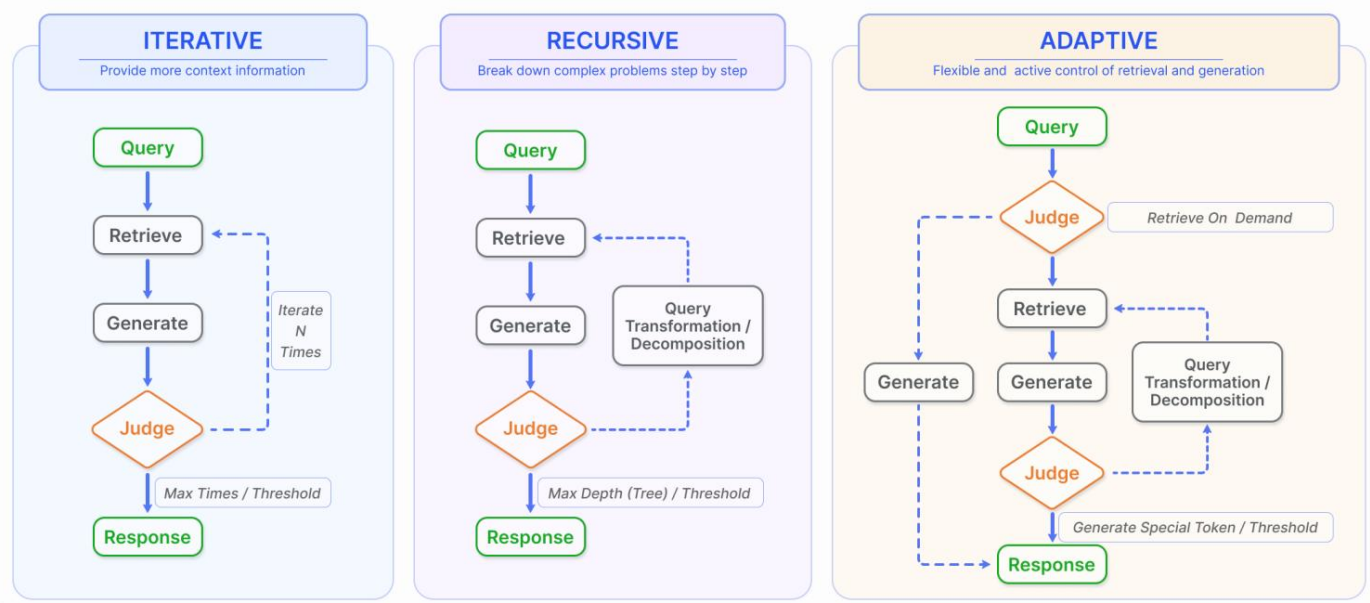

图 5. 除了最常见的一次检索外,RAG 还包含三种类型的检索增强过程。(左)迭代检索涉及在检索和生成之间交替进行,从而允许在每个步骤中从知识库中获得更丰富、更有针对性的上下文。(中)递归检索涉及逐步细化用户查询并将问题分解为子问题,然后通过检索和生成不断解决复杂问题。(右)自适应检索侧重于使 RAG 系统能够自主确定是否需要外部知识检索以及何时停止检索和生成,通常利用 LLM 生成的特殊标记进行控制。

法学硕士 (LLM) 的基础。该方法已被证明可以通过多次检索迭代提供额外的上下文参考,从而增强后续答案生成的鲁棒性。然而,它可能受到语义不连续性和无关信息积累的影响。ITER-RETGEN [14] 采用一种协同方法,将“检索增强生成”与“生成增强检索”结合起来,用于需要重现特定信息的任务。该模型利用完成输入任务所需的内容作为检索相关知识的上下文基础,从而促进在后续迭代中生成改进的答案。

B.递归检索

递归检索常用于信息检索和NLP中,以提高搜索结果的深度和相关性。

该过程涉及根据先前搜索结果迭代地优化搜索查询。递归检索旨在通过反馈循环逐步收敛到最相关的信息,从而提升搜索体验。IRCoT [61] 使用思路链 (CoT) 引导检索过程,并利用获得的检索结果优化 CoT。ToC [57] 创建了一个澄清树,系统地优化查询中的模糊部分。它在复杂的搜索场景中尤其有用,在这些场景中,用户的需求从一开始就不完全明确,或者所查找的信息高度专业化或细致入微。该过程的递归特性使其能够持续学习并适应用户的需求,通常能够提高搜索结果的满意度。

为了解决特定的数据场景,递归检索和多跳检索技术被结合使用。递归检索涉及结构化索引,以分层方式处理和检索数据,这可能包括先对文档或长篇PDF文件的各个部分进行汇总,然后基于此汇总执行检索。随后,在文档中进行二次检索以优化搜索,体现了该过程的递归特性。相比之下,多跳检索旨在更深入地挖掘图结构数据源,提取相互关联的信息[106]。

C.自适应检索

以 Flare [24] 和 Self-RAG [25] 为代表的自适应检索方法,通过使 LLM 能够主动确定检索的最佳时刻和内容,改进了 RAG 框架,从而提高了信息的效率和相关性。

这些方法体现了一种更广泛的趋势,即 LLM 在操作中运用主动判断,正如 AutoGPT、Toolformer 和 Graph-Toolformer 等模型代理所见 [107]–[109]。例如,Graph-Toolformer 将其检索过程划分为不同的步骤,LLM 主动使用检索器,运用自问技术,并使用少量提示来发起搜索查询。这种主动的策略使 LLM 能够决定何时搜索必要信息,类似于代理使用工具的方式。

WebGPT [110] 集成了强化学习框架,用于训练 GPT-3 模型在文本生成过程中自主使用搜索引擎。它使用特殊标记来引导此过程,这些标记有助于执行搜索引擎查询、浏览结果和引用参考文献等操作,从而通过使用外部搜索引擎扩展 GPT-3 的功能。Flare 通过监控生成过程的置信度来自动化时序检索,如

生成词的概率[24]。当概率低于某个阈值时,会激活检索系统收集相关信息,从而优化检索周期。自检索生成模型[25]引入了“反射标记”,允许模型自省其输出。这些标记有两种:“检索”和“评论”。模型自主决定何时激活检索,或者,预定义的阈值可以触发该过程。在检索过程中,生成器会在多个段落中进行片段级集束搜索,以得出最连贯的序列。评论家分数用于更新细分分数,并可以在推理过程中灵活调整这些权重,从而定制模型的行为。 Self-RAG 的设计无需额外的分类器或依赖自然语言推理 (NLI) 模型,从而简化了何时使用检索机制的决策过程,并提高了模型在生成准确响应方面的自主判断能力。

六、任务与评估

RAG 在 NLP 领域的快速发展和日益普及,推动了 RAG 模型评估成为 LLM 社区研究的前沿。

本次评估的主要目标是理解和优化 RAG 模型在不同应用场景下的性能。本章将主要介绍 RAG 的主要下游任务、数据集以及如何评估 RAG 系统。

A.下游任务

RAG 的核心任务仍然是问答 (QA),包括传统的单跳/多跳问答、多项选择、特定领域问答以及适合 RAG 的长篇场景。除了问答之外,RAG 还在不断扩展多个下游任务,例如信息提取 (IE)、对话生成、代码搜索等。

RAG 的主要下游任务及其相应的数据集总结在表二中。

B. 评估目标

历史上,RAG 模型评估主要集中在其在特定下游任务中的执行情况。这些评估采用与当前任务相适应的既定指标。例如,问答评估可能依赖于 EM 和 F1 分数 [7]、[45]、[59]、[72],而事实核查任务通常以准确率作为主要指标 [4]、[14]、[42]。BLEU 和 ROUGE 指标也常用于评估答案质量 [26]、[32]、[52]、[78]。类似 RALLE 等专为自动评估 RAG 应用程序而设计的工具,也同样基于这些特定于任务的指标进行评估 [160]。尽管如此,专门用于评估 RAG 模型独特特征的研究仍然非常匮乏。主要评估目标包括:

检索质量。评估检索质量对于确定检索器组件所获取上下文的有效性至关重要。来自以下领域的标准指标搜索引擎、推荐系统和信息检索系统的数据用于衡量RAG检索模块的性能。通常使用命中率、MRR和NDCG等指标来实现此目的[161],[162]。

生成质量。生成质量的评估重点在于生成器从检索到的上下文中合成连贯且相关答案的能力。此评估可根据内容目标分为:未标记内容和已标记内容。对于未标记内容,评估涵盖生成答案的真实性、相关性和无害性。相比之下,对于已标记内容,重点在于模型生成信息的准确性 [161]。此外,检索和生成质量评估都可以通过手动或自动评估方法进行 [29]、[161]、[163]。

C. 评估方面 RAG 模型的当代

评估实践强调三个主要质量分数和四个基本能力,它们共同指导对 RAG 模型的两个主要目标的评估:检索和生成。

1)质量分数:质量分数包括上下文相关性、答案忠实度和答案相关性。这些质量分数从不同角度评估 RAG 模型在信息检索和生成过程中的效率 [164]–[166]。

上下文相关性评估检索到的上下文的精确度和特殊性,确保相关性并最大限度地减少与无关内容相关的处理成本。

答案忠实度确保生成的答案符合检索到的上下文,保持一致性并避免矛盾。

答案相关性要求生成的答案与提出的问题直接相关,有效地解决核心问题。

2)必备能力:RAG 评估还包含四种能力,体现其适应性和效率:噪声鲁棒性、负抑制性、信息整合性和反事实鲁棒性 [167], [168]。这些能力对于模型在各种挑战和复杂场景下的性能至关重要,会影响模型的质量得分。

噪声鲁棒性评估模型管理与问题相关但缺乏实质性信息的噪声文档的能力。

负面拒绝评估模型在检索到的文档不包含回答问题所需的知识时避免做出回应的辨别能力。

信息集成评估模型综合来自多个文档的信息以解决复杂问题的能力。

反事实稳健性测试模型识别和忽略文档中已知不准确之处的能力,即使在被告知可能存在错误信息的情况下也是如此。

上下文相关性和噪声鲁棒性对于评价检索的质量很重要,而答案忠实度、答案相关性、否定拒绝、信息整合和反事实鲁棒性对于评价生成质量很重要。

表二
RAG的下游任务和数据集

任务	子任务	数据集	方法
质量保证	单跳	自然问题(NQ)[111]	[26], [30], [34], [42], [45], [50], [52], [59], [64], [82] [3], [4], [22], [27], [40], [43], [54], [62], [71], [112] [20], [44], [72]
		TriviaQA(TQA) [113]	[13], [30], [34], [45], [50], [64] [4], [27], [59], [62], [112] [22], [25], [43], [44], [71], [72]
		SQuAD [114]	[20], [23], [30], [32], [45], [69], [112]
		网络问题(WebQ) [115]	[3], [4], [13], [30], [50], [68]
		流行问答 [116]	[7], [25], [67]
		马可女士 [117]	[4], [40], [52]
	多跳	HotpotQA [118]	[23], [26], [31], [34], [47], [51], [61], [82] [7], [14], [22], [27], [59], [62], [69], [71], [91]
		2WikiMultiHopQA [119]	[14], [24], [48], [59], [61], [91]
		音乐 [120]	[14], [51], [61], [91]
	长篇问答	ELI5 [121]	[27], [34], [43], [49], [51]
		叙述质量保证（NQA）	[45], [60], [63], [123]
		ASQA [124]	[24], [57]
		QMSum(QM) [125]	[60],[123]
	域名质量保证	卡斯珀 [126] [60], [63]	
		COVID-QA [127] [35], [46]	
		CMB [128],MMCU医疗 [129] [81]	
	多项选择问答	质量 [130]	[60],[63]
		弧 [131]	[25], [67]
		CommonsenseQA [132]	[58], [66]
	图问答	GraphQA[84]	[84]
对话	对话生成	维基百科巫师（魔兽世界） [133] [13], [27], [34], [42]	
	个人对话	KBP [134] [74], [135]	
		杜勒蒙 [136] [74]	
	任务导向对话	凸轮休息 [137] [78], [79]	
	推荐	亚马逊(玩具,运动,美妆) [138] [39], [40]	
IE	事件参数提取 WikiEvent [139]		[13], [27], [37], [42]
		公羊 [140]	[36], [37]
	关系提取	T-REx [141],ZsRE [142]	[27], [51]
推理 常识推理		HellaSwag [143]	[20], [66]
	CoT推理	CoT推理[144]	[27]
	复杂推理	CSQA	[55]
其他的	语言理解 MMLU [146]		[7], [27], [28], [42], [43], [47], [72]
	语言建模	维基文本-103 [147]	[5], [29], [64], [71]
		策略问答 [148]	[14], [24], [48], [51], [55], [58]
	事实核查/验证热潮 [149]		[4], [13], [27], [34], [42], [50]
		公共健康[150]	[25], [67]
	文本生成	传记 [151]	[67]
	文本摘要	维基ASP [152]	[24]
		XSum [153]	[17]
	文本分类	暴力 [154]	[19]
		通过 [155]	[33]
	情绪	SST-2 [156]	[20], [33], [38]
	代码搜索	代码搜索网 [157]	[76]
	稳健性评估	来自MIRACL [56]	[56]
	数学	GSM8K [158]	[73]
	机器翻译	JRC-Acquis [159]	[17]

表三
适用于RAG评估方面的指标总结

	语境 关联	忠诚	回答 关联	噪音 鲁棒性	消极的 拒绝	信息 一体化	反事实 鲁棒性
准确性	✓	✓	✓	✓	✓	✓	✓
在					✓		
记起	✓						
精确	✓			✓			
R 率							✓
余弦相似度			✓				
命中率	✓						
月平均收入	✓						
NDCG	✓						
蓝色的	✓	✓	✓				
红色/红色-L	✓	✓	✓				

表三总结了每个评估方面的具体指标。必须认识到,这些
源自相关工作的指标是传统衡量标准
并且还不代表成熟或标准化的方法
量化 RAG 评估方面。定制指标
虽然这里没有包括 RAG 模型的细微差别,但
在一些评估研究中也得到了发展。

D.评估基准和工具

提出了一系列基准测试和工具
以方便评估 RAG。这些仪器提供
定量指标不仅可以衡量 RAG 模型的性能,还可以增强对模型在各个评估方面能力的理解。突出的基准

如 RGB、RECALL 和 CRUD [167]–[169] 专注于
评估 RAG 模型的基本能力。同时,像 RAGAS [164] 这样的最先进的自动化工具,

ARES [165] 和 TruLens8使用 LLM 来裁定
质量得分。这些工具和基准共同构成了
系统评估 RAG 的稳健框架
模型,如表 IV 所示。

七、讨论与未来展望

尽管 RAG 技术取得了长足的进步,但仍存在一些挑战需要深入研究。

本章主要介绍当前的挑战和未来
RAG面临的研究方向。

A. RAG 与长上下文

随着相关研究的深入,LLM 的背景
正在不断扩展[170]–[172]。目前,法学硕士可以
轻松管理超过 200,000 个 token 的上下文
能力意味着长文档问答,
以前依赖于 RAG,现在可以整合整个
文档直接进入提示。这也引发了
关于 LLM 课程是否仍需要 RAG 的讨论
不受上下文限制。事实上,RAG 仍然扮演着
不可替代的作用。一方面,为法学硕士提供一个
大量的上下文会对其产生重大影响
推理速度,同时分块检索和按需输入
可以显著提高运营效率。另一方面
另一方面,基于 RAG 的生成可以快速定位原始
法学硕士（LLM)的参考文献,帮助用户验证生成的答案。整个检索和推理过程是可观察的,
而仅仅依赖长上下文的生成仍然是
黑盒子。相反,上下文的扩展提供了新的
为 RAG 的发展提供机遇,使其能够
解决更复杂的问题和综合或总结
需要阅读大量材料才能回答的问题
答案[49]。在
超长上下文是未来的研究趋势之一。

B. RAG稳健性

存在噪音或矛盾的信息
检索会对 RAG 的输出质量产生不利影响。这
这种情况被形象地称为“错误信息可以
比没有信息更糟糕”。改进 RAG 的
抵制这种对抗性或反事实的输入正在获得研究动力,并已成为一项关键绩效
度量 [48], [50], [82]。Cuconasu 等人 [54] 分析了
应检索的文档类型,评估相关性
文档的提示、它们的位置以及
上下文中包含的数字。研究结果显示
包括不相关的文件可能会意外增加
准确率超过 30%,与最初的假设相矛盾
质量下降。这些结果强调了
制定专门的策略来整合检索与
语言生成模型,强调需要进一步
对RAG的稳健性进行研究和探索。

C.混合方法

将 RAG 与微调相结合正在成为一种领先的
策略。确定 RAG 和
微调是否顺序、交替或通过端到端联合训练 以及如何利用参数化

8https://www.trulens.org/trulens eval/核心概念 rag triad/
9https://kimi.moonshot.cn

表IV
评估框架概要

评估框架	评估目标	评估方面	定量指标
RGB †	检索质量	噪声鲁棒性	准确性
		负面拒绝	在
	生成质量	信息集成	准确性
		反事实稳健性	准确性
记起 †	生成质量反事实稳健性 R 率（重现率）		
拉加斯 ‡	检索质量	语境相关性	*
		忠诚	*
	生成质量	答案相关性	余弦相似度
阿瑞斯 ‡	检索质量	语境相关性	准确性
		忠诚	准确性
	生成质量	答案相关性	准确性
TruLens ‡	检索质量	语境相关性	*
		忠诚	*
	生成质量	答案相关性	*
CRUD †	检索质量	创意一代	蓝色的
		知识密集型质量保证	红色-L
	生成质量	错误纠正	Bert评分
		总结	RAGQuest评估

†代表基准，‡代表工具。*表示定制的量化指标，与传统指标有所不同。建议读者根据需要查阅相关文献，了解这些指标的具体量化公式。

非参数化优势是值得探索的领域[27]。另一个趋势是将具有特定功能的SLM引入RAG，并根据RAG系统的结果进行微调。例如，CRAG[67]训练了一个轻量级的检索评估器，用于评估查询检索到的文档的整体质量，并根据置信度触发不同的知识检索操作。

D. RAG 的缩放定律

端到端 RAG 模型以及基于 RAG 的预训练模型仍然是当前研究人员的研究重点之一[173]。这些模型的参数是关键因素之一。虽然针对 LLM 建立了缩放律 [174]，但它们是否适用于 RAG 仍不确定。像 RETRO++ [44] 这样的初步研究已经开始解决这个问题，但 RAG 模型的参数数量仍然落后于 LLM。

逆缩放定律10的可能性，即较小的模型优于较大的模型，尤其有趣，值得进一步研究。

E. 生产就绪 RAG

RAG 的实用性和与工程需求的契合促进了它的采用。然而，提高检索效率、改善大型知识库中的文档召回率以及确保数据安全（例如防止 LLM 无意中泄露文档来源或元数据是仍有待解决的关键工程挑战 [175]。

RAG 生态系统的发展在很大程度上受到其技术栈发展的影响。随着 ChatGPT 的兴起，像 LangChain 和 LLamaIndex 这样的关键工具迅速普及，提供了丰富的 RAG 相关 API，并成为 LLM 领域不可或缺的工具。新兴的技术栈虽然功能不如 LangChain 和 LLamaIndex 丰富，但凭借其专业的产品脱颖而出。例如，Flowise AI 优先采用低代码方法，允许用户通过用户友好的拖放界面部署包括 RAG 在内的 AI 应用程序。HayStack、Meltano 和 Cohere Coral 等其他技术也因其在该领域的独特贡献而备受关注。

除了专注于人工智能的供应商外，传统软件和云服务提供商也在扩展其产品线，以涵盖以 RAG 为中心的服务。Weaviate 的 Verba 专为个人助理应用程序而设计，而亚马逊的 Kendra 则提供智能企业搜索服务，使用户能够使用内置连接器浏览各种内容[11][12]存储库。

在RAG技术的发展中，呈现出明显的朝着不同专业化方向发展的趋势，例如：1）定制　定制 RAG 以满足特定要求。
2) 简化 - 使 RAG 更易于使用，以减少

10https://github.com/inverse-scaling/prize

11https://github.com/weaviate/Verba
12https://aws.amazon.com/cn/kendra/

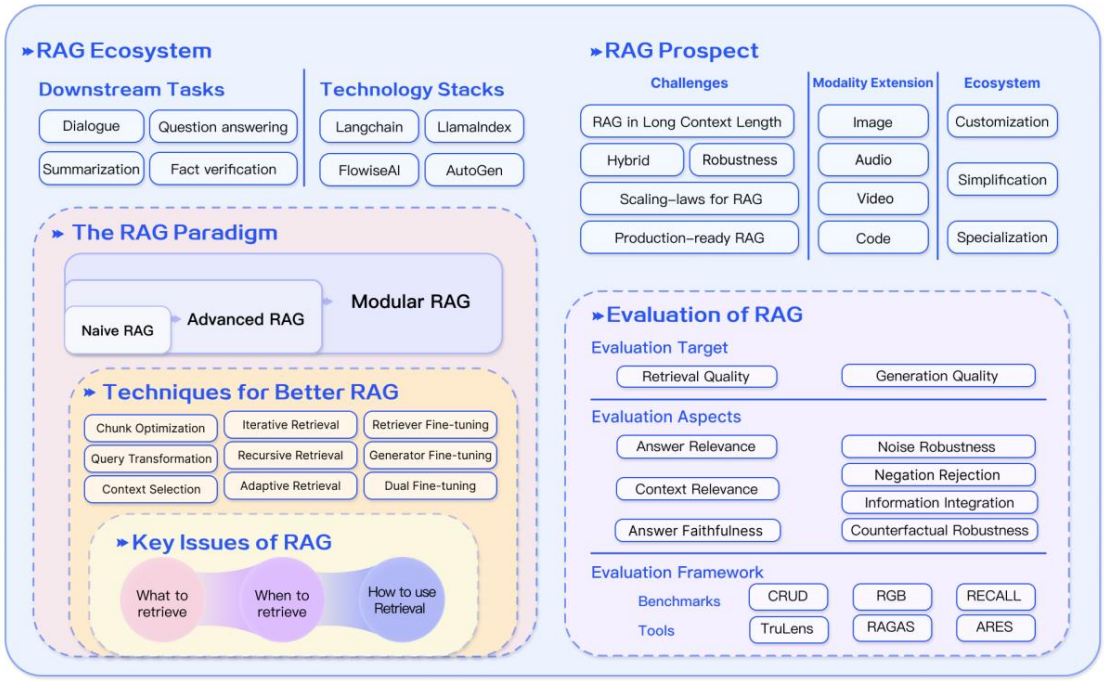

图 6. RAG 生态系统概述

初始学习曲线。3)专业化 优化 RAG 以更好地服务于生产环境。

RAG 模型及其技术栈的共同发展显而易见;技术进步不断为现有基础设施树立新的标准。反过来,技术栈的增强又推动了 RAG 功能的发展。RAG 工具包正在融合成一个基础技术栈,为高级企业应用奠定基础。然而,一个完全集成、全面的平台概念仍需未来进一步创新和发展。

F. 多模态 RAG

RAG 已经超越了最初基于文本的问答范围,涵盖了多种多样的模态数据。

这种扩展催生了创新的多模态模型,将 RAG 概念融入各个领域:

图像。RA-CM3 [176] 是检索和生成文本和图像的先驱多模态模型。

BLIP-2 [177] 利用冻结图像编码器和语言模型 (LLM) 进行高效的视觉语言预训练,实现了零样本图像到文本的转换。“先可视化后书写”方法 [178] 利用图像生成来引导语言模型 (LM) 的文本生成,在开放式文本生成任务中展现出良好的前景。

音频和视频。GSS 方法检索并拼接音频片段,将机器翻译数据转换为语音翻译数据 [179]。UEOP 通过结合外部离线语音转文本策略,标志着端到端自动语音识别的重大进步 [180]。此外,基于 KNN 的注意力融合利用音频嵌入和语义相关的文本嵌入来改进 ASR,从而加速领域自适应。

Vid2Seq 通过专门的时间标记增强语言模型,有助于在统一的输出序列中预测事件边界和文本描述 [181]。

代码。RBPS [182] 通过编码和频率分析检索符合开发人员目标的代码示例,从而在小规模学习任务中表现出色。该方法已在测试断言生成和程序修复等任务中证明其有效性。对于结构化知识,CoK 方法 [106] 首先从知识图谱中提取与输入查询相关的事实,然后将这些事实作为提示集成到输入中,从而提升了知识图谱问答任务的性能。

八、结论

如图 6 所示,本文摘要强调了 RAG 通过将语言模型中的参数化知识与外部知识库中的大量非参数化数据相结合,在增强 LLM 功能方面取得的重大进步。该综述展示了 RAG 技术的演变及其在众多不同任务中的应用。分析概述了 RAG 框架内的三种发展范式:朴素 RAG、高级 RAG 和模块化 RAG,每种范式都代表了对其前身的逐步增强。RAG 与其他人工智能方法(例如微调和强化学习)的技术集成进一步扩展了其功能。尽管 RAG 技术取得了进步,但仍有研究机会来提高其鲁棒性和处理扩展上下文的能力。

RAG 的应用范围正在扩展到多模态领域,其原理也适用于解释和处理图像、视频和代码等多种数据形式。这一扩展凸显了 RAG 对人工智能部署的重大实际意义,引起了学术界和工业界的广泛关注。

以 RAG 为中心的 AI 应用不断涌现,支持工具也不断发展,这体现了 RAG 生态系统的蓬勃发展。随着 RAG 应用场景的不断拓展,我们需要改进评估方法,以跟上其发展的步伐。确保绩效评估的准确性和代表性,对于全面展现 RAG 对 AI 研发社区的贡献至关重要。

参考

[1] N. Kandpal, H. Deng, A. Roberts, E. Wallace, 和 C. Raffel, “大型语言模型难以学习长尾知识”, 国际机器学习会议 PMLR, 2023, pp. 15696–15707.

[2] Y.Zhang,Y.Li,L.Cui,D.Cai,L.Liu,T.Fu,X.Huang,E.Zhao,Y.Zhang,Y.Chen等人，“人工智能海洋中的海妖之歌:大语言模型中的幻觉调查”,arXiv预印本arXiv:2309.01219,2023。

[3] D. Arora、A. Kini、SR Chowdhury、N. Natarajan、G. Sinha 和 A. Sharma，“Gar-meets-rag 范式与零样本信息检索”,arXiv 预印本 arXiv:2310.20158,2023 年。

[4] P. Lewis、E. Perez、A. Piktus、F. Petroni、V. Karpukhin、N. Goyal、H. Kuttler、M. Lewis、W.-t. Yih、T. Rockt aschel ¨ 等人，“知识密集型 NLP 任务的检索增强生成”,神经信息处理系统进展,第 33 卷,第 9459-9474 页，2020 年。

[5] S. Borgeaud、A. Mensch、J. Hoffmann、T. Cai、E. Rutherford、K. Milli-can、GB Van Den Driessche、J.-B. Lespiau、B. Damoc、A. Clark 等人，“通过检索数万亿个标记来改进语言模型”,国际机器学习会议 PMLR,2022 年,第 2206-2240 页。

[6] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray 等人，“训练语言模型以遵循人类反馈的指令”,神经信息处理系统进展,第 35 卷,第 27 730–27 744 页,2022 年。

[7] X. Ma, Y. Gong, P. He, H. Zhao, 和 N. Duan，“检索增强大型语言模型的查询重写”,arXiv 预印本 arXiv:2305.14283, 2023 年。

[8] 我. ILIN，“先进技术:概述”， https://pub.towardsai.net/advanced-rag-techniques-an-illustrated-overview-04d193d8fec6,2023年。

[9] W. Peng, G. Li, Y. Jiang, Z. Wang, D. Ou, X. Zeng, E. Chen 等人，“基于大型语言模型的淘宝搜索长尾查询重写”,arXiv 预印本 arXiv:2311.03758,2023 年。

[10] HS Zheng, S. Mishra, X. Chen, H.-T. Cheng, EH Chi, QV Le 和 D. Zhou，“退一步:在大型语言模型中通过抽象来引发推理”,arXiv 预印本 arXiv:2310.06117,2023 年。

[11] L. Gao, X. Ma, J. Lin, 和 J. Callan, “无相关标签的精确零样本密集检索”, arXiv 预印本 arXiv:2212.10496, 2022 年。

[12] V. Blagojevi，“增强 Haystack 中的 rag 管道:引入 diver-sityranker 和 lostinthemiddleranker”， https://towardsdatascience.com/enhancing-rag-pipelines-in-haystack-45f14e2bc9f5,2023年。

[13] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng 和 M. Jiang，“生成而非检索:大型语言模型是强大的上下文生成器”,arXiv 预印本 arXiv:2209.10063,2022 年。

[14] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan 和 W. Chen，“通过迭代检索-生成协同作用增强检索增强大型语言模型”,arXiv 预印本 arXiv:2305.15294, 2023 年。

[15] X. Wang, Q. Yang, Y. Qiu, J. Liang, Q. He, Z. Gu, Y. Xiao, 和 W. Wang，“Knowledgpt:通过知识库中的检索和存储访问增强大型语言模型”,arXiv 预印本 arXiv:2308.11761, 2023 年。

[16] A. H. Raudaschl 是破布融合，” “忘记破布， https://towardsdatascience.com/forget-rag-the-future-is-rag-fusion-1147298d8ad1,2023年。

[17] X. Cheng, D. Luo, X. Chen, L. Liu, D. Zhao, 和 R. Yan, “提升自己:利用自我记忆进行检索增强文本生成”, arXiv 预印本 arXiv:2305.02437, 2023 年。

[18] S. Wang, Y. Xu, Y. Fang, Y. Liu, S. Sun, R. Xu, C. Zhu, 和 M. Zeng，“训练数据比你想象的更有价值:一种从训练数据中检索的简单有效的方法”,arXiv 预印本 arXiv:2203.08773,2022 年。

[19] X. Li, E. Nie, 和 S. Liang，“从分类到生成:洞察跨语言检索增强 icl”,arXiv 预印本 arXiv:2311.06595, 2023 年。

[20] D. Cheng, S. Huang, J. Bi, Y. Zhan, J. Liu, Y. Wang, H. Sun, F. Wei, D. Deng 和 Q. Zhang，“Uprise:用于改进零样本评估的通用即时检索”,arXiv 预印本 arXiv:2303.08518,2023 年。

[21] 戴正,赵伟,马建,栾玉,倪建,陆建,A.巴卡洛夫,K.Guu,KB Hall,M.-W. Chang，“Promptagator:从 8 个示例中进行少量密集检索”,arXiv 预印本 arXiv:2209.11755,2022 年。

[22] Z. Sun, X. Wang, Y. Tay, Y. Yang, 和 D. Zhou, “Recitation-augmented language models,” arXiv 预印本 arXiv:2210.01296, 2022 年。

[23] O. Khattab,K. Santhanam,XL Li,D. Hall,P. Liang,C. Potts 和 M. Zaharia，“演示-搜索-预测:为知识密集型 NLP 构建检索和语言模型”,arXiv 预印本 arXiv:2212.14024,2022 年。

[24] Z. Jiang,FF Xu,L. Gau,Z. Sun,Q. Liu,J. Dwivedi-Yu,Y. Yang,J. Callan,G. Neubig。

[25] A. Asai, Z. Wu, Y. Wang, A. Sil 和 H. Hajishirzi，“自我反思:通过自我反省学习检索、生成和批评”,arXiv 预印本 arXiv:2310.11511,2023 年。

[26] Z. Ke, W. Kong, C. Li, M. Zhang, Q. Mei 和 M. Bendersky，“弥合检索者与法律硕士之间的偏好差距”,arXiv 预印本 arXiv:2401.06954, 2024 年。

[27] XV Lin,X. Chen,M. Chen,W. Shi,M. Lomeli,R. James,P. Ro-driguez,J. Kahn,G. Szilvasy,M. Lewis 等人，“Ra-dit:检索增强的双指令调优”,arXiv 预印本 arXiv:2310.02323。

[28] O. Ovadia, M. Brief, M. Mishaeli 和 O. Elisha，“微调还是检索?比较法学硕士中的知识注入”,arXiv 预印本 arXiv:2312.05934,2023 年。

[29] T. Lan, D. Cai, Y. Wang, H. Huang 和 X.-L. Mao，“Copy is all you need”,第 11 届国际学习表征会议,2022 年。

[30] T. Chen, H. Wang, S. Chen, W. Yu, K. Ma, X. Zhao, D. Yu, 和 H. Zhang, “密集 x 检索:我们应该使用什么检索粒度？” arXiv 预印本 arXiv:2312.06648, 2023 年。

[31] F. Luo 和 M. Surdeanu，“分而治之,实现蕴涵感知的多跳证据检索”,arXiv 预印本 arXiv:2311.02616,2023 年。

[32] Q. Gou, Z. Xia, B. Yu, H. Yu, F. Huang, Y. Li, 和 N. Cam-Tu, “通过检索增强风格转换实现问题生成多样化”, arXiv 预印本 arXiv:2310.14503, 2023 年。

[33] Z. Guo, S. Cheng, Y. Wang, P. Li, 和 Y. Liu, “针对非知识密集型任务的提示引导检索增强”, arXiv 预印本 arXiv:2305.17653, 2023 年。

[34] Z. Wang, J. Araki, Z. Jiang, MR Parvez 和 G. Neubig，“学习过滤上下文以进行检索增强生成”,arXiv 预印本 arXiv:2311.08377, 2023 年。

[35] M. Seo, J. Baek, J. Thorne 和 SJ Hwang，“针对低资源领域任务的检索增强数据增强”,arXiv 预印本 arXiv:2402.13482, 2024 年。

[36] Y. Ma, Y. Cao, Y. Hong, 和 A. Sun，“大型语言模型不是一个好的小样本信息提取器,而是一个好的困难样本重排序器！”arXiv 预印本 arXiv:2303.08559, 2023 年。

[37] X. Du 和 H. Ji，“用于事件参数提取的检索增强生成式问答”,arXiv 预印本 arXiv:2211.07067,2022 年。

[38] L. Wang,N. Yang 和 F. Wei，“学习检索大型语言模型的上下文示例”,arXiv 预印本 arXiv:2307.07164,2023 年。

[39] S. Rajput,N. Mehta,A. Singh,RH Keshavan,T. Wu,L. Heldt,L. Hong,Y. Tay,VQ Tran,J. Samost 等人，“具有生成检索的推荐系统”,arXiv 预印本 arXiv:2305.05065,2023。

[40] B. Jin, H. Zeng, G. Wang, X. Chen, T. Wei, R. Li, Z. Wang, Z. Li, Y. Li, H. Lu 等人，“语言模型作为语义索引器”, arXiv 预印本 arXiv:2310.07815, 2023 年。

[41] R. Anantha,T. Bethi,D. Vodianik 和 S. Chappidi，“用于检索增强生成的上下文调整”,arXiv 预印本 arXiv:2312.05708,2023 年。

[42] G. Izacard,P. Lewis,M. Lomeli,L. Hosseini,F. Petroni,T. Schick,J. Dwivedi-Yu,A. Joulin,S. Riedel 和 E. Grave，“使用检索增强语言模型进行小样本学习”,arXiv 预印本 arXiv:2208.03299,2022 年。

[43] J. Huang, W. Ping, P. Xu, M. Shoeybi, KC-C. Chang 和 B. Catan-zaro，“Raven:基于检索增强编码器-解码器语言模型的语境学习”,arXiv 预印本 arXiv:2308.07922,2023 年。

[44] B. Wang, W. Ping, P. Xu, L. McAfee, Z. Liu, M. Shoeybi, Y. Dong, O. Kuchaiev, B. Li, C. Xiao 等人，“我们是否应该使用检索功能对自回归语言模型进行预训练?一项综合研究”，arXiv 预印本 arXiv:2304.06762,2023 年。

[45] B. Wang、W. Ping、L. McAfee、P. Xu、B. Li、M. Shoeybi 和 B. Catan-zaro，“Instructretro:指令调整后检索增强预训练”，arXiv 预印本 arXiv:2310.07713,2023 年。

[46] S. Siriwardhana、R. Weerasekera、E. Wen、T. Kaluarachchi、R. Rana 和 S. Nanayakkara，“改进检索增强生成(rag) 模型在开放领域问答中的领域适应性”，《计算语言学协会会刊》，第 11 卷，第 1-17 页，2023 年。

[47] Z. Yu, C. Xiong, S. Yu, 和 Z. Liu, “增强自适应检索器作为通用插件提高了语言模型的泛化能力”，arXiv 预印本 arXiv:2305.17331, 2023 年。

[48] O. Yoran, T. Wolfson, O. Ram 和 J. Berant，“使检索增强语言模型对不相关的上下文具有鲁棒性”，arXiv 预印本 arXiv:2310.01558, 2023 年。

[49] H.-T. Chen, F. Xu, SA Arora 和 E. Choi，“理解长篇问答的检索增强”，arXiv 预印本 arXiv:2310.12150, 2023 年。

[50] W. Yu, H. Zhang, X. Pan, K. Ma, H. Wang 和 D. Yu，“笔记链:增强检索增强语言模型的鲁棒性”，arXiv 预印本 arXiv:2311.09210, 2023 年。

[51] S. Xu, L. Pang, H. Shen, X. Cheng 和 T.-S. Chua，“链中搜索:面向知识密集型任务的准确、可信、可追溯的大型语言模型”，CoRR，vol. abs/2304.14732，2023 年。

[52] M. Berchansky, P. Izsak, A. Caciularu, I. Dagan 和 M. Wasserblat，“通过标记消除优化检索增强阅读器模型”，arXiv 预印本 arXiv:2310.13682, 2023 年。

[53] J. Lala、O. O Donoghue、A. Shtedritski、S. Cox、SG Rodriques 和 AD White。

[54] F. Cuconasu、G. Trappolini、F. Siciliano、S. Filice、C. Campagnano、Y. Maarek、N. Tonellotto 和 F. Silvestri。

[55] Z. Zhang, X. Zhang, Y. Ren, S. Shi, M. Han, Y. Wu, R. Lai 和 Z. Cao，“Iag:用于回答推理问题的归纳增强生成框架”，2023 年自然语言处理经验方法会议论文集，2023 年，第 1-14 页。

[56] N. Thakur、L. Bonifacio、X. Zhu、O. Ogundepo、E. Kamalloo、D. Alfonso-Hermelo、X. Li、Q. Liu、B. Chen、M. Rezagholizadeh 等人 arXiv:2312 .11361，2023。

[57] G. Kim, S. Kim, B. Jeon, J. Park 和 J. Kang，“澄清树:使用检索增强大型语言模型回答模糊问题”，arXiv 预印本 arXiv:2310.14696，2023 年。

[58] Y. Wang, P. Li, M. Sun, 和 Y. Liu, “面向大型语言模型的自我知识引导检索增强”, arXiv 预印本 arXiv:2310.05002, 2023 年。

[59] Z. Feng, X. Feng, D. Zhao, M. Yang, 和 B. Qin, “检索-生成协同增强大型语言模型”, arXiv 预印本 arXiv:2310.05149, 2023 年。

[60] P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi 和 B. Catanzaro，“检索与长上下文大型语言模型”，arXiv 预印本 arXiv:2310.03025,2023 年。

[61] H. Trivedi, N. Balasubramanian, T. Khot 和 A. Sabharwal，“针对知识密集型多步骤问题的思路链推理交叉检索”，arXiv 预印本 arXiv:2212.10509,2022 年。

[62] R. Ren, Y. Wang, Y. Qu, WX Zhao, J. Liu, H. Tian, H. Wu, J.-R. Wen 和 H. Wang，“通过检索增强研究大型语言模型的事实知识边界”，arXiv 预印本 arXiv:2307.11019,2023 年。

[63] P. Sarthi、S. Abdullah、A. Tuli、S. Khanna、A. Goldie 和 CD Manning，“Raptor:用于树状检索的递归抽象处理”，arXiv 预印本 arXiv:2401.18059,2024 年。

[64] O. Ram、Y. Levine、I. Dalmedigos、D. Muhlgay、A. Shashua、K. Leyton-Brown 和 Y. Shoham，“上下文检索增强语言模型”，arXiv 预印本 arXiv:2302.00083,2023 年。

[65] Y. Ren, Y. Cao, P. Guo, F. Fang, W. Ma, 和 Z. Lin，“检索和采样:通过混合检索增强进行文档级事件参数提取”，载于第 61 届计算语言学协会年会论文集（第 1 卷:长篇论文），2023 年，第 293-306 页。

[66] Z. Wang, X. Pan, D. Yu, D. Yu, J. Chen 和 H. Ji，“Zemi:从多任务中学习零样本半参数语言模型”，arXiv 预印本 arXiv:2210.00185,2022 年。

[67] S.-Q. Yan、J.-C. Gu、Y. Zhu 和 Z.-H. Ling，“矫正检索增强生成”，arXiv 预印本 arXiv:2401.15884,2024 年。

[68] P. Jain、LB Soares 和 T. Kwiatkowski，“1-pager:一次性答案生成和证据检索”，arXiv 预印本 arXiv:2310.16568，2023 年。

[69] H. Yang, Z. Li, Y. Zhang, J. Wang, N. Cheng, M. Li, 和 J. Xiao, “Prca:通过可插拔的奖励驱动上下文适配器为检索问答拟合黑盒大型语言模型”, arXiv 预印本 arXiv:2310.18347, 2023 年。

[70] S. Zhuang, B. Liu, B. Koopman 和 G. Zuccon，“开源大型语言模型是用于文档排名的强大零样本查询似然模型”，arXiv 预印本 arXiv:2310.13243,2023 年。

[71] F. Xu, W. Shi, 和 E. Choi, “Recomp:通过压缩和选择性增强改进检索增强 lms”, arXiv 预印本 arXiv:2310.04408, 2023 年。

[72] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettle-moyer 和 W.-t. Yih，“Replug:检索增强黑盒语言模型”，arXiv 预印本 arXiv:2301.12652,2023 年。

[73] E. Melz，“通过手臂训练增强法学硕士智能:用于检索增强生成的辅助基本原理记忆”，arXiv 预印本 arXiv:2311.04177,2023 年。

[74] H. Wang, W. Huang, Y. Deng, R. Wang, Z. Wang, Y. Wang, F. Mi, JZ Pan 和 K.-F. Wong，“Unims-rag:面向个性化对话系统的统一多源检索增强生成”，arXiv 预印本 arXiv:2401.13256, 2024 年。

[75] Z. Luo, C. Xu, P. Zhao, X. Geng, C. Tao, J. Ma, Q. Lin, 和 D. Jiang, “具有参数知识指导的增强大型语言模型”, arXiv 预印本 arXiv:2305.04757, 2023 年。

[76] X. Li, Z. Liu, C. Xiong, S. Yu, Y. Gu, Z. Liu, 和 G. Yu, “结构感知语言模型预训练改进了结构化数据的密集检索”, arXiv 预印本 arXiv:2305.19912, 2023 年。

[77] M. Kang, JM Kwak, J. Baek 和 SJ Hwang，“用于基于知识的对话生成的知识图谱增强语言模型”，arXiv 预印本 arXiv:2305.18846,2023 年。

[78] W. Shen, Y. Gao, C. Huang, F. Wan, X. Quan, 和 W. Bi，“端到端任务导向对话系统的检索-生成对齐”，arXiv 预印本 arXiv:2310.08877, 2023 年。

[79] T. Shi, L. Li, Z. Lin, T. Yang, X. Quan, 和 Q. Wang，“面向任务的对话系统的双反馈知识检索”，arXiv 预印本 arXiv:2310.14528, 2023 年。

[80] P. Ranade 和 A. Joshi，“Fabula:使用检索增强叙事结构生成情报报告”，arXiv 预印本 arXiv:2310.13848,2023 年。

[81] X. Jiang, R. Zhang, Y. Xu, R. Qiu, Y. Fang, Z. Wang, J. Tang, H. Ding, X. Chu, J. Zhao 等人，“思考与检索:知识图谱增强医学大型语言模型的假设”，arXiv 预印本 arXiv:2312.15883,2023 年。

[82] J. Baek, S. Jeong, M. Kang, JC Park 和 SJ Hwang，“知识增强语言模型验证”，arXiv 预印本 arXiv:2310.12836, 2023 年。

[83] L. Luo, Y.-F. Li, G. Haffari 和 S. Pan，“基于图的推理:忠实且可解释的大型语言模型推理”，arXiv 预印本 arXiv:2310.01061,2023 年。

[84] X. He, Y. Tian, Y. Sun, NV Chawla, T. Laurent, Y. LeCun, X. Bresson 和 B. Hooi，“G-retriever:用于文本图理解和问答的检索增强生成”，arXiv 预印本 arXiv:2402.07630, 2024 年。

[85] L. Zha, J. Zhou, L. Li, R. Wang, Q. Huang, S. Yang, J. Yuan, C. Su, X. Li, A. Su 等人，“Tablegpt:将表、自然语言和命令统一为一个 gpt”，arXiv 预印本 arXiv:2307.08674,2023 年。

[86] M. Gaur、K. Gunaratna、V. Srinivasan 和 H. Jin，“Iseeq:使用动态元信息检索和知识图谱生成信息搜索问题”，载于《AAAI 人工智能会议论文集》，第 36 卷，第 10 期，2022 年，第 10672-10680 页。

[87] F. Shi、X. Chen、K. Misra、N. Scales、D. Dohan、EH Chi、N. Scharli ¨ 和 D. Zhou，“大型语言模型很容易因不相关的上下文而分散注意力”，国际机器学习会议。

PMLR，2023，第 31 210-31 227 页。

[88] R. Teja，“使用 llamaindex 评估碎布的理想块大小”， https://www.llamaindex.ai/blog/system evaluating-the-ideal-chunk-size-for-a-rag-system-using-llamaindex-6207e5d3fec5，2023年。

[89] Langchain，“按字符递归拆分”， https://python.langchain. com/docs/modules/data connection/document transforms/recursive text splitter， 2023 年。

[90] S. “先进的格大检索”， https://towardsdatascience.com/ 01： 从小到大,拉advanced-rag-01-small-to-big-retrieval-172181b396d4,2023年。

[91] Y. Wang, N. Lipka, RA Rossi, A. Siu, R. Zhang 和 T. Derr，“知识图谱提示多文档问答”，arXiv 预印本 arXiv:2308.11730, 2023 年。

[92] D. Zhou, N. Scharli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le 等人，“从最少到最多的提示使大型语言模型能够进行复杂的推理”，arXiv 预印本 arXiv:2205.10625,2022 年。

[93] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz 和 J. Weston，“验证链减少了大型语言模型中的幻觉”，arXiv 预印本 arXiv:2309.11495, 2023 年。

[94] X. Li 和 J. Li，“角度优化的文本嵌入”，arXiv 预印本 arXiv:2309.12871, 2023 年。

[95] VoyageAI，“Voyage 的嵌入模型”， https://docs.voyageai.com/embeddings/,2023年。

[96] BAY，“Flagembedding”， https://github.com/FlagOpen/FlagEmbedding， 2023 年。

[97] P. Zhang, S. Xiao, Z. Liu, Z. Dou 和 J.-Y. Nie，“检索任何内容以增强大型语言模型”，arXiv 预印本 arXiv:2310.07554,2023 年。

[98] NF Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni 和 P. Liang，“迷失在中间:语言模型如何使用长上下文”，arXiv 预印本 arXiv:2307.03172,2023 年。

[99] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, 和 J. Zhang，“Chat-rec: 面向交互式和可解释的 llms 增强推荐系统”, arXiv 预印本 arXiv:2303.14524, 2023 年。

[100] N. Anderson、C. Wilson 和 SD Richardson，“Lingua:解决现场口译和自动配音场景问题”，载于《美洲机器翻译协会第 15 届双年会论文集》（第 2 卷:用户和提供商轨道和政府轨道），J. Campbell、S. Larocca、J. Marciano、K. Savenkov 和 A. Yanishevsky 编辑。美国奥兰多:美洲机器翻译协会,2022 年 9 月,第 202-209 页。

[在线]。可访问网址:https://aclanthology.org/2022.amta-upg.14 [101] H. Jiang, Q. Wu, X. Luo, D. Li, C.-Y. Lin, Y. Yang, 和 L. Qiu，“Longllmlingua:通过快速压缩加速和增强长语境场景中的 llms”，arXiv 预印本 arXiv:2310.06839,2023 年。

[102] V. Karpukhin、B. Oguz、S. Min、P. Lewis、L. Wu、S. Edunov、D. Chen 和 W.-t. Yih，“面向开放域问答的密集段落检索”，arXiv 预印本 arXiv:2004.04906,2020 年。

[103] Y. Ma, Y. Cao, Y. Hong 和 A. Sun，“大型语言模型不是一个好的小样本信息提取器,而是一个好的困难样本重排器！”ArXiv,vol. abs/2303.08559,2023 年。[在线]。可用网址： https://api.semanticscholar.org/CorpusID:257532405 [104] J. Cui、Z. Li、Y. Yan、B. Chen 和 L. Yuan，“Chatlaw:集成外部知识库的开源法律大型语言模型”，arXiv 预印本 arXiv:2306.16092,2023 年。

[105] O. Yoran, T. Wolfson, O. Ram 和 J. Berant，“使检索增强语言模型对不相关的上下文具有鲁棒性”，arXiv 预印本 arXiv:2310.01558, 2023 年。

[106] X. Li, R. Zhao, YK Chia, B. Ding, L. Bing, S. Joty 和 S. Poria，“知识链:一个将大型语言模型与结构化知识库相结合的框架”，arXiv 预印本 arXiv:2305.13269,2023 年。

[107] H. Yang, S. Yue, 和 Y. He，“用于在线决策的 Auto-GPT:基准和补充意见”, arXiv 预印本 arXiv:2306.02224, 2023 年。

[108] T. Schick、J. Dwivedi-Yu、R. Dess` 、R. Raileanu、M. Lomeli、L. Zettle-moyer、N. Cancedda 和 T. Scialom，“Toolformer:语言模型可以自学使用工具”，arXiv 预印本 arXiv:2302.04761,2023 年。

[109] J. Zhang，“Graph-toolformer:通过 ChatGPT 增强提示来赋予 llms 图形推理能力”，arXiv 预印本 arXiv:2304.11116,2023 年。

[110] R. Nakano、J. Hilton、S. Balaji、J. Wu、L. Ouyang、C. Kim、C. Hesse、S. Jain、V. Kosaraju、W. Saunders 等人，“Webgpt:带有人工反馈的浏览器辅助问答”，arXiv 预印本 arXiv:2112.09332,2021 年。

[111] T. Kwiatkowski、J. Palomaki、O. Redfield、M. Collins、A. Parikh、C. Alberti、D. Epstein、I. Polosukhin、J. Devlin、K. Lee 等人，“自然问题:问答研究的基准”，《学报》计算语言学协会,第 7 卷,第 453-466 页，2019年。

[112] Y. Liu, S. Yavuz, R. Meng, M. Moorthy, S. Joty, C. Xiong, 和 Y. Zhou，“探索检索器与大型语言模型的集成策略”, arXiv 预印本 arXiv:2308.12574, 2023 年。

[113] M. Joshi, E. Choi, DS Weld 和 L. Zettlemoyer，“Triviaqa:用于阅读理解的大规模远程监督挑战数据集”，arXiv 预印本 arXiv:1705.03551,2017 年。

[114] P. Rajpurkar、J. Zhang、K. Lopyrev 和 P. Liang，“Squad:机器理解文本的 100,000 多个问题”，arXiv 预印本 arXiv:1606.05250,2016 年。

[115] J. Berant、A. Chou、R. Frostig 和 P. Liang，“基于问答对的自由基语义解析”，2013 年自然语言处理经验方法会议论文集,2013 年,第 1533-1544 页。

[116] A. Mallen、A. Asai、V. Zhong、R. Das、H. Hajishirzi 和 D. Khashabi，“何时不信任语言模型:研究参数和非参数记忆的有效性和局限性”，arXiv 预印本 arXiv:2212.10511,2022 年。

[117] T. Nguyen、M. Rosenberg、X. Song、J. Gao、S. Tiwary、R. Majumder 和 L. Deng，“Ms marco:人类生成的机器阅读理解数据集”，2016 年。

[118] Z. Yang, P. Qi, S. Zhang, Y. Bengio, WW Cohen, R. Salakhutdi-nov 和 CD Manning，“Hotpotqa:一个用于多样化、可解释的多跳问答的数据集”，arXiv 预印本 arXiv:1809.09600,2018 年。

[119] X. Ho, A.-KD Nguyen, S. Sugawara, 和 A. Aizawa，“构建多跳 qa 数据集以全面评估推理步骤”, arXiv 预印本 arXiv:2011.01060, 2020 年。

[120] H. Trivedi、N. Balasubramanian、T. Khot 和 A. Sabharwal，《Musique:通过单跳问题组合实现多跳问题》，《计算语言学协会会刊》，第 10 卷,第 539-554 页,2022 年。

[121] A. Fan、Y. Jernite、E. Perez、D. Grangier、J. Weston 和 M. Auli，“Eli5:长篇问答”，arXiv 预印本 arXiv:1907.09190，2019 年。

[122] T. Kocisk ˇ y、J. Schwarz、P. Blunsom、C. Dyer、KM Hermann、G. Melis 和 E. Grefenstette，“叙事阅读理解挑战”，计算语言学协会会刊,第 6 卷,第 317-328 页,2018 年。

[123] K.-H. Lee, X. Chen, H. Furuta, J. Canny, 和 I. Fischer，“一种可记忆超长上下文要点的受人类启发的阅读代理”, arXiv 预印本 arXiv:2402.09727, 2024 年。

[124] I. Stelmakh、Y. Luan、B. Dhingra 和 M.-W. Chang，“驱动:事实问题遇到长篇答案”，arXiv 预印本 arXiv:2204.06092，2022 年。

[125] M.钟,D.尹,T.Yu,A.Zaidi,M.Mutuma,R.Jha,AH。Awadallah、A. Celikyilmaz、Y. Liu、X. Qiu 等人，“Qmsum:基于查询的多领域会议摘要的新基准”，arXiv 预印本 arXiv:2104.05938,2021 年。

[126] P. Dasigi、K. Lo、I. Beltagy、A. Cohan、NA Smith 和 M. Gardner，“基于研究论文的信息搜索问题与答案数据集”，arXiv 预印本 arXiv:2105.03011,2021 年。

[127] T. Moller、A. Reina、R. Jayakumar 和 M. Pietsch，“Covid-qa:covid-19 问答数据集”，载于 ACL 2020 COVID-19 自然语言处理研讨会（NLP-COVID），2020 年。

[128] X. Wang, GH Chen, D. Song, Z. Zhang, Z. Chen, Q. Xiao, F. Jiang, J. Li, X. Wan, B. Wang 等人，“Cmb:中文综合医学基准”，arXiv 预印本 arXiv:2308.08833, 2023 年。

[129] H. Zeng，“测量大规模多任务中文理解能力”，arXiv 预印本 arXiv:2304.12986, 2023 年。

[130] RY Pang, A. Parrish, N. Joshi, N. Nangia, J. Phang, A. Chen, V. Pad-makumar, J. Ma, J. Thompson, H. He 等人，“质量:使用长输入文本进行问答,是的！” arXiv 预印本 arXiv:2112.08608,2021 年。

[131] P. Clark、I. Cowhey、O. Etzioni、T. Khot、A. Sabharwal、C. Schoenick 和 O. Tafjord，“你认为你已经解决了问答问题吗?试试 ai2 推理挑战 arc”，arXiv 预印本 arXiv:1803.05457,2018 年。

[132] A. Talmor、J. Herzig、N. Lourie 和 J. Berant，“Commonsenseqa:针对常识知识的问答挑战”，arXiv 预印本 arXiv:1811.00937,2018 年。

[133] E. Dinan、S. Roller、K. Shuster、A. Fan、M. Auli 和 J. Weston，“维基百科奇才:知识驱动的对话代理”，arXiv 预印本 arXiv:1811.01241,2018 年。

[134] H. Wang, M. Hu, Y. Deng, R. Wang, F. Mi, W. Wang, Y. Wang, W.-C. Kwan, I. King 和 K.-F. Wong，“大型语言模型作为源

个性化知识基础对话规划器”，arXiv 预印本 arXiv:2310.08840，2023 年。

[135] ，“大型语言模型作为个性化知识基对话的源规划器”，arXiv 预印本 arXiv:2310.08840，2023 年。

[136] X. Xu, Z. Gou, W. Wu, Z.-Y. Niu, H. Wu, H. Wang 和 S. Wang，“好久不见！基于长期角色记忆的开放域对话”，arXiv 预印本 arXiv:2203.05797，2022 年。

[137] T.-H。 Wen，M. Gasic，N. Mrksic，L.M. Rojas-Barahona，P.-H。Su、S. Ultes、D. Vandyke 和 S. Young，“神经对话系统中的条件生成和快照学习”，arXiv 预印本 arXiv:1606.03352，2016 年。

[138] R. He 和 J. McAuley，“起伏：通过单类协同过滤建模时尚趋势的视觉演变”，第 25 届万维网国际会议论文集，2016 年，第 507-517 页。

[139] S. Li、H. Ji 和 J. Han，“通过条件生成提取文档级事件参数”，arXiv 预印本 arXiv:2104.05919，2021 年。

[140] S. Ebner, P. Xia, R. Culkin, K. Rawlins 和 B. Van Durme，“多句子论证链接”，arXiv 预印本 arXiv:1911.03766，2019 年。

[141] H. Elsahar、P. Vougiouklis、A. Remaci、C. Gravier、J. Hare、F. Laforest 和 E. Simperl，“T-rex：自然语言与知识库三元组的大规模比对”，载于《第十一届国际语言资源与评估会议论文集》（LREC 2018），2018 年。

[142] O. Levy, M. Seo, E. Choi 和 L. Zettlemoyer，“通过阅读理解进行零样本关系提取”，arXiv 预印本 arXiv:1706.04115，2017 年。

[143] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi 和 Y. Choi，“Hel-laswag：机器真的能完成你的句子吗？” arXiv 预印本 arXiv:1905.07830，2019 年。

[144] S. Kim, SJ Joo, D. Kim, J. Jang, S. Ye, J. Shin 和 M. Seo，“The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning”，arXiv 预印本 arXiv:2305.14045，2023 年。

[145] A. Saha、V. Pahuja、M. Khapra、K. Sankaranarayanan 和 S. Chandar，“复杂顺序问答：利用知识图谱学习通过链接问答对进行对话”，载于《AAAI 人工智能会议论文集》，第 32 卷，第 1 期，2018 年。

[146] D. Hendrycks、C. Burns、S. Basart、A. Zou、M. Mazeika、D. Song 和 J. Steinhardt，“测量大规模多任务语言理解”，arXiv 预印本 arXiv:2009.03300，2020 年。

[147] S. Merity、C. Xiong、J. Bradbury 和 R. Socher，“指针哨兵混合模型”，arXiv 预印本 arXiv:1609.07843，2016 年。

[148] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth 和 J. Berant，“亚里士多德使用笔记本电脑吗？基于隐性推理策略的问答基准测试”，《计算语言学协会会刊》，第 9 卷，第 346-361 页，2021 年。

[149] J. Thorne、A. Vlachos、C. Christodoulopoulos 和 A. Mittal，“发烧：用于事实提取和验证的大规模数据集”，arXiv 预印本 arXiv:1803.05355，2018 年。

[150] N. Kotonya 和 F. Toni，“可解释的公共卫生声明的自动化事实核查”，arXiv 预印本 arXiv:2010.09926，2020 年。

[151] R. Lebret、D. Grangier 和 M. Auli，“从结构化数据生成神经文本及其在传记领域的应用”，arXiv 预印本 arXiv:1603.07771，2016 年。

[152] H. Hayashi、P. Budania、P. Wang、C. Ackerson、R. Neervannan 和 G. Neubig，“Wikiasp：基于方面的多领域摘要数据集”，计算语言学协会会刊，第 9 卷，第 211-225 页，2021 年。

[153] S. Narayan、SB Cohen 和 M. Lapata，“不要给我细节，只要摘要！用于极端摘要的主题感知卷积神经网络”，arXiv 预印本 arXiv:1808.08745，2018 年。

[154] S. Saha、JA Junaed、M. Saleki、AS Sharma、MR Rifat、M. Rahouti、SI Ahmed、N. Mohammed 和 MR Amin，“Vio-lens：一个包含导致不同形式社区暴力的带注释社交网络帖子的新型数据集及其评估”，载于《第一届孟加拉语处理研讨会论文集（BLP-2023）》，2023 年，第 72-84 页。

[155] X. Li 和 D. Roth，“学习问题分类器”，载于 COLING 2002：第 19 届国际计算语言学会议，2002 年。

[156] R. Socher, A. Perelygin, J. Wu, J. Chuang, CD Manning, AY Ng 和 C. Potts，“情绪树库上的语义组合性递归深度模型”，2013 年自然语言处理经验方法会议论文集，2013 年，第 1631-1642 页。

[157] H. Husain、H.-H. Wu、T. Gazit、M. Allamanis 和 M. Brockschmidt，“Codesearchnet 挑战：评估语义代码搜索的状态”，arXiv 预印本 arXiv:1909.09436，2019 年。

[158] K. Cobbe、V. Kosaraju、M. Bavarian、M. Chen、H. Jun、L. Kaiser、M. Plappert、J. Tworek、J. Hilton、R. Nakano 等人，“训练验证者解决数学应用题”，arXiv 预印本 arXiv:2110.14168，2021 年。

[159] R. Steinberger、B. Pouliquen、A. Widiger、C. Ignat、T. Erjavec、D. Tufis 和 D. Varga，“jrc-acquis：包含 20 多种语言的多语言对齐平行语料库”，arXiv 预印本 cs/0609058，2006。

[160] Y. Hoshi、D. Miyashita、Y. Ng、K. Tatsuno、Y. Morioka、O. Torii 和 J. Deguchi。

[161] J. Liu，“构建可投入生产的 rag 应用程序”， https://www.ai.engineer/summit/schedule/building-production-ready-rag-applications，2023年。

[162] I. Nguyen，“评估 rag 第一部分：如何评估文档检索”https://www.deepset.ai/blog/rag-evaluation-retrieval，2023年。

[163] Q. Leng、K. Uhlenhuth 和 A. Polyzotis，“LLM 评估 RAG 应用程序的最佳实践”， https://www.databricks.com/blog/LLM-auto-eval-best-practices-RAG，2023年。

[164] S. Es、J. James、L. Espinosa-Anke 和 S. Schockaert，“Ragas：检索增强生成的自动评估”，arXiv 预印本 arXiv：2309.15217，2023。

[165] J. Saad-Falcon、O. Khattab、C. Potts 和 M. Zaharia，“Ares：检索增强生成系统的自动评估框架”，arXiv 预印本 arXiv:2311.09476，2023 年。

[166] C. Jarvis 和 J. Allard，“最大化性能技术调查”， https: //community.openai.com/t/openai-dev-day-2023-breakout-sessions/505239#a-survey-of-techniques-for-maximizing-llm-performance-2，2023年。

[167] J. Chen, H. Lin, X. Han, 和 L. Sun, “检索增强生成中的大型语言模型基准测试”, arXiv 预印本 arXiv:2309.01431, 2023 年。

[168] Y. Liu, L. Huang, S. Li, S. Chen, H. Zhou, F. Meng, J. Zhou, 和 X. Sun, “回忆：针对外部反事实知识的 llms 稳健性的基准”，arXiv 预印本 arXiv:2311.08147, 2023 年。

[169] Y. Lyu, Z. Li, S. Niu, F. Xiong, B. Tang, W. Wang, H. Wu, H. Liu, T. Xu, 和 E. Chen, “Crud-rag：用于检索增强大型语言模型生成的综合中文基准”，arXiv 预印本 arXiv:2401.17043, 2024 年。

[170] P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi 和 B. Catanzaro，“检索与长上下文大型语言模型”，arXiv 预印本 arXiv:2310.03025，2023 年。

[171] C. Packer、V. Fang、SG Patil、K. Lin、S. Wooders 和 JE Gon-zalez，“Memgpt：迈向 llms 作为操作系统”，arXiv 预印本 arXiv:2310.08560，2023 年。

[172] G. Xiao, Y. Tian, B. Chen, S. Han, 和 M. Lewis, “具有注意力池的高效流式语言模型”, arXiv 预印本 arXiv:2309.17453, 2023 年。

[173] T. 张、SG Patil、N. Jain、S. Shen、M. Zaharia、I. Stoica 和 JE Gonzalez，“Raft：使语言模型适应特定领域的 rag”，arXiv 预印本 arXiv:2403.10131，2024 年。

[174] J. Kaplan、S. McCandlish、T. Henighan、TB Brown、B. Chess、R. Child、S. Gray、A. Radford、J. Wu 和 D. Amodei，“神经语言模型的缩放定律”，arXiv 预印本 arXiv:2001.08361，2020 年。

[175] U. Alon, F. Xu, J. He, S. Sengupta, D. Roth 和 G. Neubig，“基于自动增强检索的神经符号语言建模”，国际机器学习会议 PMLR，2022 年，第 468-485 页。

[176] M. Yasunaga、A. Aghajanyan、W. Shi、R. James、J. Leskovec、P. Liang、M. Lewis、L. Zettlemoyer 和 W.-t。 Yih，“检索增强的多模态语言建模”，arXiv 预印本 arXiv:2211.12561，2022 年。

[177] J. Li, D. Li, S. Savarese 和 S. Hoi，“Blip-2：使用冻结图像编码器和大型语言模型进行引导语言图像预训练”，arXiv 预印本 arXiv:2301.12597，2023 年。

[178] W. Zhu、A. Yan、Y. Lu、W. Xu、XE Wang、M. Eckstein 和 WY Wang，“先想象后写：想象引导的开放式文本生成”，arXiv 预印本 arXiv:2210.03765，2022 年。

[179] J. Zhao, G. Haffar 和 E. Shareghi，“从口语词汇生成合成语音用于语音翻译”，arXiv 预印本 arXiv:2210.08174，2022 年。

[180] DM Chan、S. Ghosh、A. Rastrow 和 B. Hoffmeister，“在上下文端到端自动语音识别中使用外部离策略语音到文本映射”，arXiv 预印本 arXiv:2301.02736，2023 年。

[181] A.Yang、A.Nagrani、PHSeo、A.Miech、J.Pont-Tuset、I.Laptev、J.Sivic 和 C.Schmid，《Vid2seq：用于密集视频字幕的视觉语言模型的大规模预训练》，载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，2023 年，第 10714-10726 页。

[182] N. Nashid、M. Sintaha 和 A. Mesbah，“基于检索的代码相关小样本学习的提示选择”，2023 年 IEEE/ACM 第 45 届国际软件工程会议（ICSE），2023 年，第 2450-2462 页。