

# 大型语言模型的检索增强生成技术综述

高云帆<sup>1</sup>, 熊云<sup>2</sup>, 高新宇<sup>2</sup>, 贾康翔<sup>2</sup>, 潘金柳<sup>2</sup>, 毕宇曦<sup>3</sup>, 戴毅<sup>1</sup>, 孙佳伟<sup>1</sup>, 郭倩宇<sup>4</sup>, 王萌<sup>3</sup>  
和王浩芬<sup>1,3</sup>\*<sup>1</sup>同济大学智能自主系统研究院

<sup>2</sup>复旦大学计算机科学学院数据科学重点实验室

<sup>3</sup>同济大学设计与创意学院

<sup>4</sup>复旦大学计算机科学学院

## 摘要

大型语言模型（LLMs）展示了显著的能力，但面临幻觉、过时知识和非透明、不可追溯的推理过程等挑战。

检索增强生成（RAG）通过整合外部数据库的知识，已经成为一种有前途的解决方案。这提高了模型的准确性和可信度，特别适用于知识密集型任务，并允许持续更新知识和集成领域特定信息。RAG将LLMs的内在知识与庞大、动态的外部数据库库相融合。本综述论文详细考察了RAG范式的发展，包括Naive RAG、Advanced RAG和Modular RAG。它详细审查了RAG框架的三个基本组成部分，包括检索、生成和增强技术。本文重点介绍了嵌入在这些关键组件中的最新技术，深入理解RAG系统的进展。此外，本文介绍了评估RAG模型的度量标准和基准，以及最新的评估框架。总之，本文勾勒了研究的前景，包括挑战的识别、多模态的扩展以及RAG基础设施和生态系统的发展。<sup>1</sup>

在包括Super-GLUE [Wang等, 2019]、MMLU [Hendrycks等, 2020]和BIG-bench [Srivastava等, 2022]在内的各种基准测试中，LLMs表现出卓越的性能。尽管取得了这些进展，LLMs在处理特定领域或高度专业化查询方面仍存在明显的局限性[Kandpal等, 2023]。一个常见问题是生成错误信息或“幻觉”[Zhang等, 2023b]，特别是当查询超出模型的训练数据范围或需要最新信息时。这些缺点凸显了在实际生产环境中部署LLMs作为黑盒解决方案的不切实际性，需要额外的保护措施。缓解这些局限性的一种有希望的方法是检索增强生成（RAG），它将外部数据检索集成到生成过程中，从而增强模型提供准确和相关的响应的能力。

RAG，由Lewis等人[2020年Lewis等人]于2020年中引入，是LLM领域中的一个范例，增强了生成任务。具体而言，RAG包括一个初始检索步骤，LLM查询外部数据源以获取相关信息，然后再回答问题或生成文本。这个过程不仅在后续的生成阶段提供信息，还确保回答基于检索到的证据，从而显著提高输出的准确性和相关性。

在推理阶段动态地从知识库中检索信息使得RAG能够解决生成事实不正确的内容（通常称为“幻觉”）的问题。将RAG集成到LLM中已经得到了快速采用，并成为改进聊天机器人能力和使LLM更适用于实际应用的关键技术。

RAG的进化轨迹分为四个独特的阶段，如图1所示。在2017年的初始阶段，与Transformer架构的出现相一致，主要的推动力是通过预训练模型（PTM）吸收额外的知识来增强语言模型。这个时期见证了RAG在优化预训练方法方面的基础性工作。

在这个初始阶段之后，chatGPT的出现之前，相关研究相对停滞不前。chatGPT的到来标志着RAG研究的一个关键时刻。

## 1 引言

大型语言模型（LLMs）如GPT系列[Brown等, 2020, OpenAI, 2023]和LLama系列[Touvron等, 2023]，以及Gemini [Google, 2023]等其他模型，在自然语言处理方面取得了显著的成功，展示了卓越的性能。

\*通讯作者。电子邮件：haofen.wang@tongji.edu.cn

<sup>1</sup>资源可在<https://github.com/Tongji-KGLLM/RAG-Survey>获取

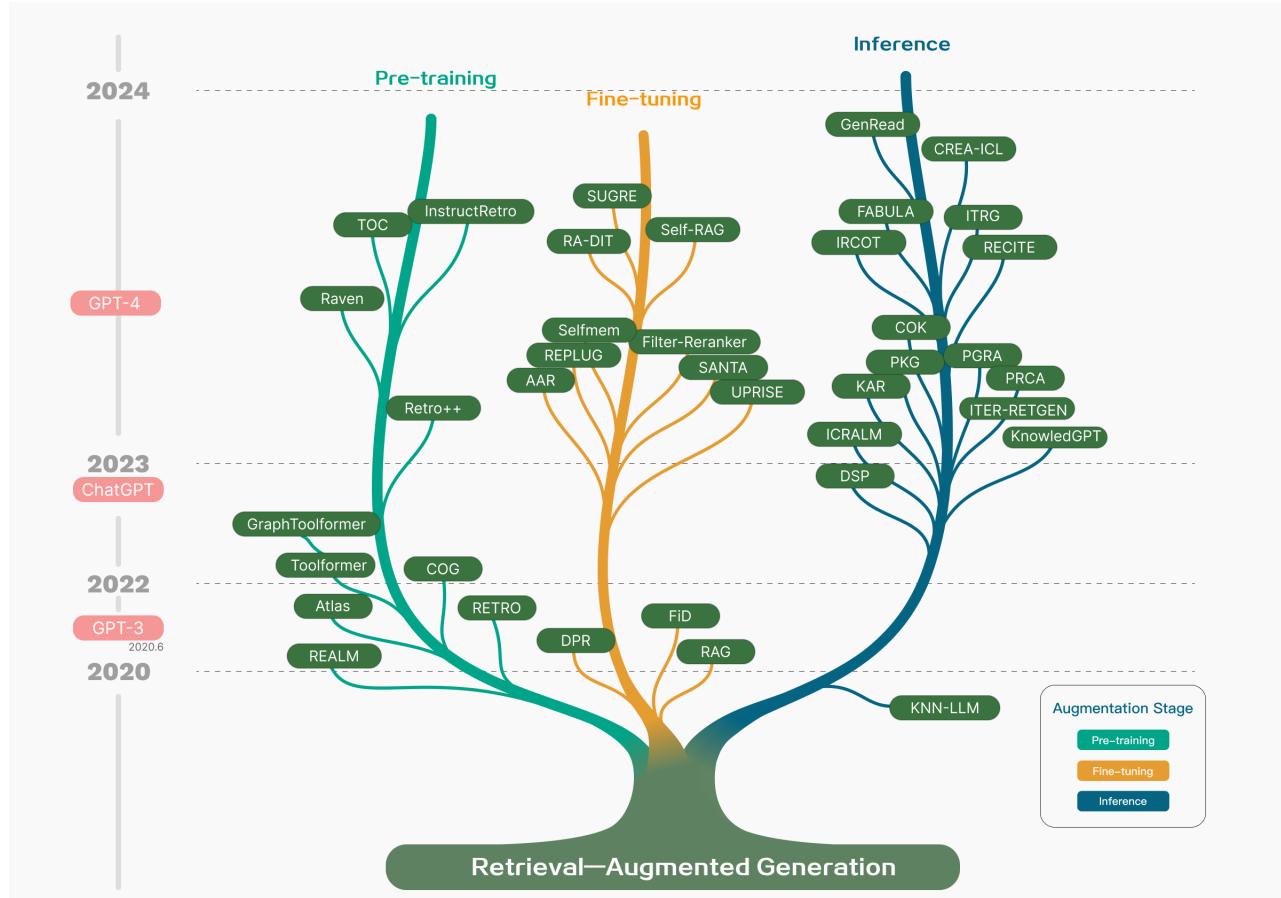


图1：RAG研究发展的技术树，展示了代表性的作品

轨迹将LLMs推向了前沿。社区的焦点转向利用LLMs的能力来实现更高的可控性和应对不断变化的需求。因此，RAG的大部分工作集中在推理上，只有少部分工作专注于微调过程。随着LLM能力的不断提升，特别是GPT-4的引入，RAG技术的格局发生了重大变革。重点转向了RAG和微调的混合方法，同时还有一小部分人继续专注于优化预训练方法。

尽管RAG研究迅速增长，但在该领域缺乏系统性的整合和抽象，这给理解RAG进展的全面情况带来了挑战。本调查旨在概述整个RAG过程，并涵盖LLMs中的当前和未来方向的RAG研究，通过对LLMs出现后的技术原理、发展历史、内容以及相关方法和应用的全面总结和组织，以及RAG的评估方法和应用场景。它旨在提供现有RAG技术的全面概述和分析，并为未来发展方法提供结论和展望。

本调查旨在为读者和实践者提供关于大型模型和RAG的全面而系统的理解，阐明检索增强的进展和关键技术，澄清各种技术的优点和局限性以及适用的背景，并预测潜在的未来发展。本调查旨在为读者和实践者提供关于大型模型和RAG的全面而系统的理解，阐明检索增强的进展和关键技术，澄清各种技术的优点和局限性以及适用的背景，并预测潜在的未来发展。

我们的贡献如下：

- 我们全面系统地回顾了最新的RAG技术，描述了其从原始RAG、高级RAG到模块化RAG的演变过程。这个综述将RAG研究的更广泛范围置于LLM领域的背景之中。
- 我们确定并讨论了RAG过程中的核心技术，特别关注“检索”、“生成器”和“增强”方面，并深入探讨它们之间的协同作用，阐明这些组成部分如何密切合作形成一个有凝聚力和高效的RAG框架。
- 我们构建了一个全面的RAG评估框架，概述了评估目标和指标。我们的比较分析阐明了RAG相对于各种微调方法的优势和劣势。

从各个角度来看。此外，我们预测了RAG的未来发展方向，强调解决当前挑战的潜在增强措施，拓展到多模态环境，并发展其生态系统。

本文的结构如下：第2和3节定义了RAG并详细介绍了其发展过程。第4至6节探讨了核心组件——检索、“生成”和“增强”，并突出了多样的嵌入式技术。

第7节着重介绍了RAG的评估系统。第8节比较了RAG与其他LLM优化方法，并提出了其发展的潜在方向。本文在第9节进行了总结。

## 2 定义

RAG的定义可以从其工作流程中总结出来。

图2描述了一个典型的RAG应用工作流程。在这种情况下，用户向ChatGPT询问一个最近备受关注的事件（即OpenAI首席执行官的突然解雇和恢复），这引发了广泛的公众讨论。作为最著名和广泛使用的LLM，ChatGPT由于其预训练数据的限制，缺乏对最新事件的了解。RAG通过从外部知识库中检索最新的文档摘录来解决这一差距。在这种情况下，它获取了与查询相关的一系列新闻文章。这些文章与初始问题一起合并成一个丰富的提示，使ChatGPT能够综合出一个有根据的回答。这个例子说明了RAG的过程，展示了它利用实时信息检索来增强模型的回答能力。从技术上讲，RAG通过各种创新方法进行了丰富，解决了诸如“检索什么”、“何时检索”和“如何使用检索到的信息”等关键问题。对于“检索什么”，研究已经从简单的标记检索[Khandelwal等，2019]和实体检索[Nishikawa等，2022]发展到更复杂的结构，如块检索[Ram等，2023]和知识图[Kang等，2023]，研究重点放在检索的粒度和数据结构的层次上。

粗粒度带来更多信息，但精度较低。检索结构化文本提供更多信息，但牺牲了效率。“何时检索”的问题导致了各种策略，从单一的[Wang等，2023年e，Shi等，2023年]到自适应的[Jiang等，2023年b，Huang等，2023年]和多重检索[Izacard等，2022年]方法。高频率的检索带来更多信息和较低的效率。至于“如何使用”检索到的数据，已经在模型的各个层次上开发了集成技术，包括输入[Khattab等，2022年]、中间[Borgeaud等，2022年]和输出层[Liang等，2023年]。尽管“中间层”和“输出层”更有效，但存在训练需求和低效率的问题。

RAG是一种通过整合外部知识库来增强LLMs的范例。它采用协同方法，结合信息检索机制和上下文学习（ICL）来增强LLM的性能。在这个框架中，用户发起的查询会触发检索操作。

通过搜索算法获取相关信息。然后将这些信息编织到LLM的提示中，为生成过程提供额外的上下文。RAG的主要优势在于无需重新训练LLM以适用于特定任务。开发者可以附加外部知识库，丰富输入，从而提高模型的输出精度。由于其高实用性和低门槛，RAG已成为LLM系统中最受欢迎的架构之一，许多对话产品几乎完全基于RAG构建。RAG的工作流程包括三个关键步骤。首先，将语料库划分为离散的块，利用编码器模型构建向量索引。其次，RAG根据查询和索引块之间的向量相似性识别和检索块。最后，模型根据检索块获取的

上下文信息生成响应。这些步骤构成了RAG过程的基本框架，支撑其信息检索和上下文感知生成能力。接下来，我们将介绍RAG研究框架。

## 3 RAG 框架

RAG研究范式不断发展，本节主要描述其进展。我们将其分为三种类型：Naive RAG、Advanced RAG 和 Modular RAG。虽然RAG具有成本效益，并且超越了原生LLM的性能，但也存在一些限制。Advanced RAG 和 Modular RAG 的发展是对 Naive RAG 中特定缺点的回应。

### 3.1 Naive RAG

Naive RAG研究范式代表了最早的方法论，在ChatGPT广泛采用之后迅速崭露头角。Naive RAG遵循传统的索引、检索和生成过程。它也被称为“Retrieve-Read”框架 [Ma et al., 2023a]。

#### 索引

索引过程是数据准备的关键初始步骤，离线进行，包括多个阶段。它从数据索引开始，对原始数据进行清理和提取，并将PDF、HTML、Word 和 Markdown 等各种文件格式转换为标准化纯文本。为了适应语言模型的上下文限制，这段文本被分割成更小、更易管理的块，这个过程称为分块。然后，通过嵌入模型将这些块转换为向量表示，选择嵌入模型是为了在推理效率和模型大小之间取得平衡。这有助于在检索阶段进行相似性比较。最后，创建一个索引来存储这些文本块及其向量嵌入作为键值对，从而实现高效和可扩展的搜索能力。

#### 检索

在收到用户查询后，系统使用与索引阶段相同的编码模型来进行转码

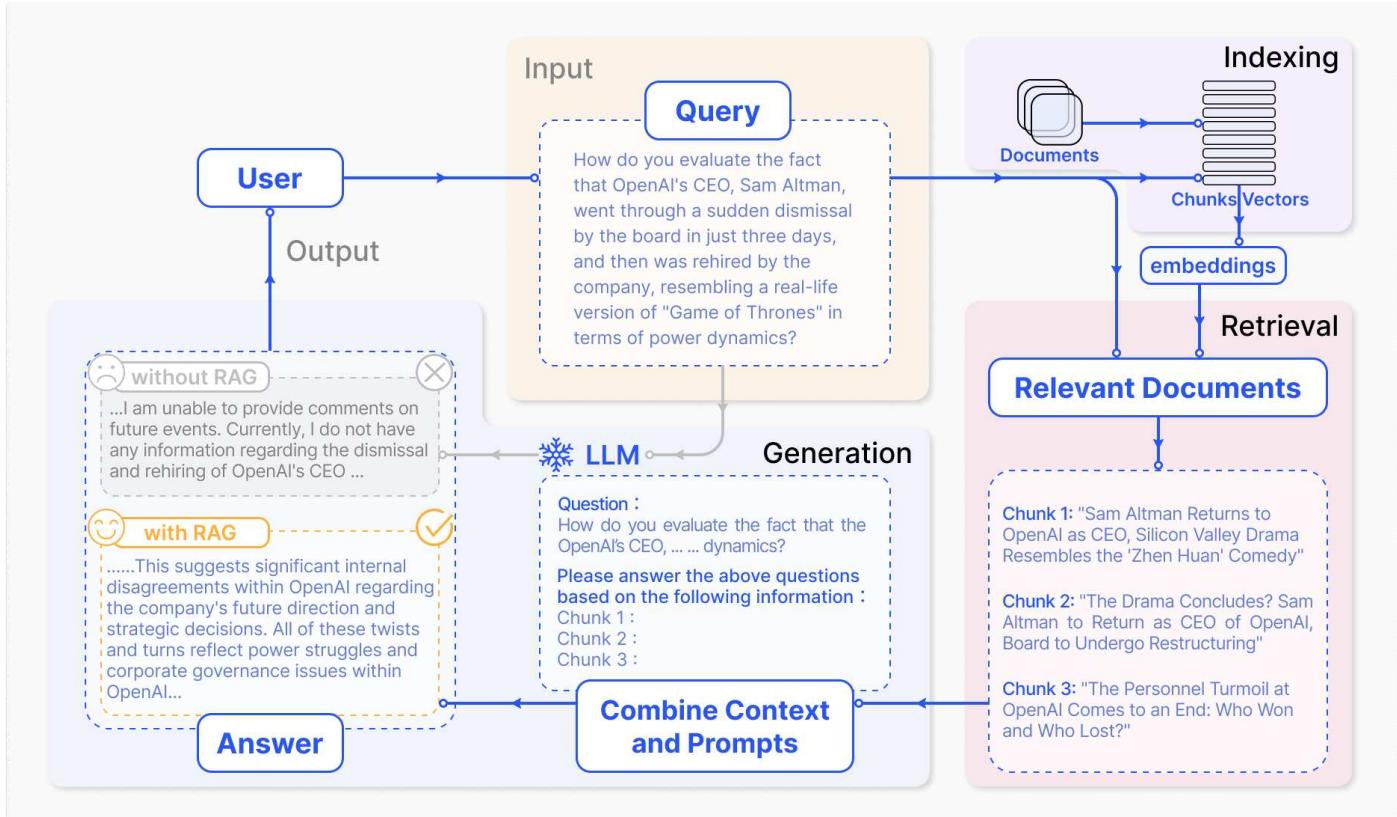


图2：应用于问答的RAG过程的代表性实例

将输入转换为向量表示 然后计算查询向量与索引语料库中向量化块之间的相似度分数 系统优先选择并检索与查询最相似的前K个块 这些块随后被用作扩展上下文基础，用于处理用户的请求

全面的回答。过时的信息进一步加剧了问题，可能导致不准确的检索结果。

响应生成质量存在幻觉挑战，模型生成的答案与提供的上下文不相关，还存在与模型输出相关的无关上下文和潜在的有害或偏见问题。

增强过程在有效地整合来自检索段落的上下文与当前生成任务方面存在挑战，可能导致不连贯或不一致的输出。冗余和重复也是问题，特别是当多个检索到的段落包含相似信息时，在生成的回答中会出现重复内容。

识别多个检索到的段落对生成任务的重要性和相关性是另一个挑战，需要适当平衡每个段落的价值。此外，协调写作风格和语气的差异以确保输出的一致性是至关重要的。最后，生成模型过度依赖增强信息的风险可能导致输出仅仅重申检索到的内容，而没有提供新的价值或综合信息。

## 生成

将提出的查询和选定的文档合成为一个连贯的提示，大型语言模型负责生成回答 模型的回答方法可能因任务特定标准而异，它可以利用其固有的参数化知识或限制其回答范围在提供的文档中的信息内 在进行持续对话时，可以将任何现有的对话历史整合到提示中，使模型能够有效地进行多轮对话交互

## Naive RAG的缺点

Naive RAG在三个关键领域面临重大挑战：“检索”，“生成”和“增强”。

检索质量面临各种挑战，包括低精度，导致检索到的块不对齐和出现幻觉或中途丢失等潜在问题。还存在低召回率的情况，导致未能检索到所有相关块，从而阻碍了LLMs的综合构建能力。

## 3.2 高级RAG

先进的RAG已经通过有针对性的增强措施来解决Naive RAG的缺点。在检索质量方面，先进的RAG实现了预检索。

和后检索策略。为了解决Naive RAG所遇到的索引挑战，Advanced RAG通过使用滑动窗口、细粒度分割和元数据等技术来改进其索引方法。它还引入了各种方法来优化检索过程[ILIN, 2023]。

### 检索前处理

优化数据索引优化数据索引的目标是提高索引内容的质量。

这涉及五个主要策略：提高数据粒度、优化索引结构、添加元数据、对齐优化和混合检索。

提高数据粒度的目标是提升文本标准化、一致性、事实准确性和丰富的上下文，以提高RAG系统的性能。这包括删除不相关的信息，消除实体和术语的歧义，确认事实准确性，保持上下文并更新过时的文档。

优化索引结构涉及调整块的大小以捕获相关上下文，跨多个索引路径进行查询，并利用图数据索引中节点之间的关系来捕获相关上下文。

添加元数据信息涉及将引用的元数据（如日期和目的）集成到块中以进行过滤，并将章节和引用的子节等元数据纳入其中，以提高检索效率。

对齐优化通过引入“假设问题”[Li et al., 2023d]来解决文档之间的对齐问题和差异。

### 检索

在检索阶段，主要关注通过计算查询和块之间的相似度来确定适当的上下文。嵌入模型在这个过程中起着核心作用。在高级RAG中，有潜力优化嵌入模型。

**微调嵌入。**微调嵌入模型对RAG系统中检索内容的相关性有重要影响。这个过程涉及定制嵌入模型，以增强特定领域上下文中的检索相关性，特别是处理涉及新兴或罕见术语的专业领域。BGE嵌入模型[BAAI, 2023]<sup>2</sup>，如BAAI开发的BGE-large-EN，是一个可以进行微调以优化检索相关性的高性能嵌入模型的例子。微调的训练数据可以使用像GPT-3.5-turbo这样的语言模型生成，以基于文档块制定问题，然后将其用作微调对。

**动态嵌入**适应单词使用的上下文，不同于静态嵌入，静态嵌入为每个单词使用一个向量[Karpukhin et al., 2020]。例如，在BERT等变压器模型中，同一个单词的嵌入可以根据周围的单词而变化。Ope-nAI的嵌入-ada-02模型<sup>3</sup>，基于以下原则构建

<sup>2</sup><https://huggingface.co/BAAI/bge-large-en>

<sup>3</sup><https://platform.openai.com/docs/guides/embeddings>

像GPT这样的LLM的Ope-nAI的嵌入-ada-02模型<sup>3</sup>是一个复杂的动态嵌入模型，可以捕捉上下文理解。然而，它可能不像最新的GPT-4等全尺寸语言模型那样对上下文敏感。

### 检索后处理

在从数据库中检索到有价值的上下文之后，将其与查询合并作为LLMs的输入是必要的，同时要解决上下文窗口限制带来的挑战。简单地一次性将所有相关文档呈现给LLM可能会超过上下文窗口限制，引入噪音，并阻碍对关键信息的关注。需要对检索到的内容进行额外处理以解决这些问题。

**重新排序。**重新排序检索到的信息以将最相关的内容重新定位到提示的边缘是一种关键策略。这个概念已经在LlamaIndex<sup>4</sup>、LangChain<sup>5</sup>和HayStack [Blagojevi, 2023]等框架中实现。例如，Diversity Ranker<sup>6</sup>根据文档多样性进行重新排序，而LostInTheMiddleRanker则交替将最佳文档放置在上下文窗口的开头和结尾。此外，像cohereAI rerank [Cohere, 2023]、bge-rerank<sup>7</sup>和LongLLMLingua [Jiang et al., 2023a]这样的方法重新计算相关文本与查询之间的语义相似性，解决了基于向量的语义相似性模拟搜索的挑战。

**提示压缩。**研究表明，检索到的文档中的噪音会对RAG的性能产生不利影响。在后处理中，重点是压缩无关上下文，突出关键段落，减少整体上下文长度。Selective Context和LLMLingua [Litman et al., 2020, Anderson et al., 2022]等方法利用小型语言模型计算提示的互信息或困惑度，估计元素的重要性。Recomp [Xue et al., 2023a]通过在不同粒度上训练压缩器来解决这个问题，而Long Context [Xue et al., 2023b]和“Walking in the Memory Maze”[Chenet et al., 2023a]则设计了摘要技术来增强LLM对关键信息的感知能力，特别是在处理大量上下文时。

## 3.3 模块化 RAG

模块化的RAG结构与传统的Naive RAG框架不同，提供了更大的灵活性和可变性。它集成了各种方法来增强功能模块，例如在检索器中引入了一个相似性检索模块，并在检索器中应用了微调方法[Lin et al., 2023]。已经开发了重构的RAG模块[Yuet al., 2022]和迭代方法，如[Shao et al., 2023]，以解决特定问题。模块化的RAG范式在RAG领域越来越成为常态，可以在多个模块之间进行串行流水线或端到端训练。三种RAG范式的比较

<sup>4</sup><https://www.llamaindex.ai>

<sup>5</sup><https://www.langchain.com/>

<sup>6</sup><https://haystack.deepset.ai/blog/enhancing-rag-pipelines-in-haystack>

<sup>7</sup><https://huggingface.co/BAAI/bge-reranker-large>

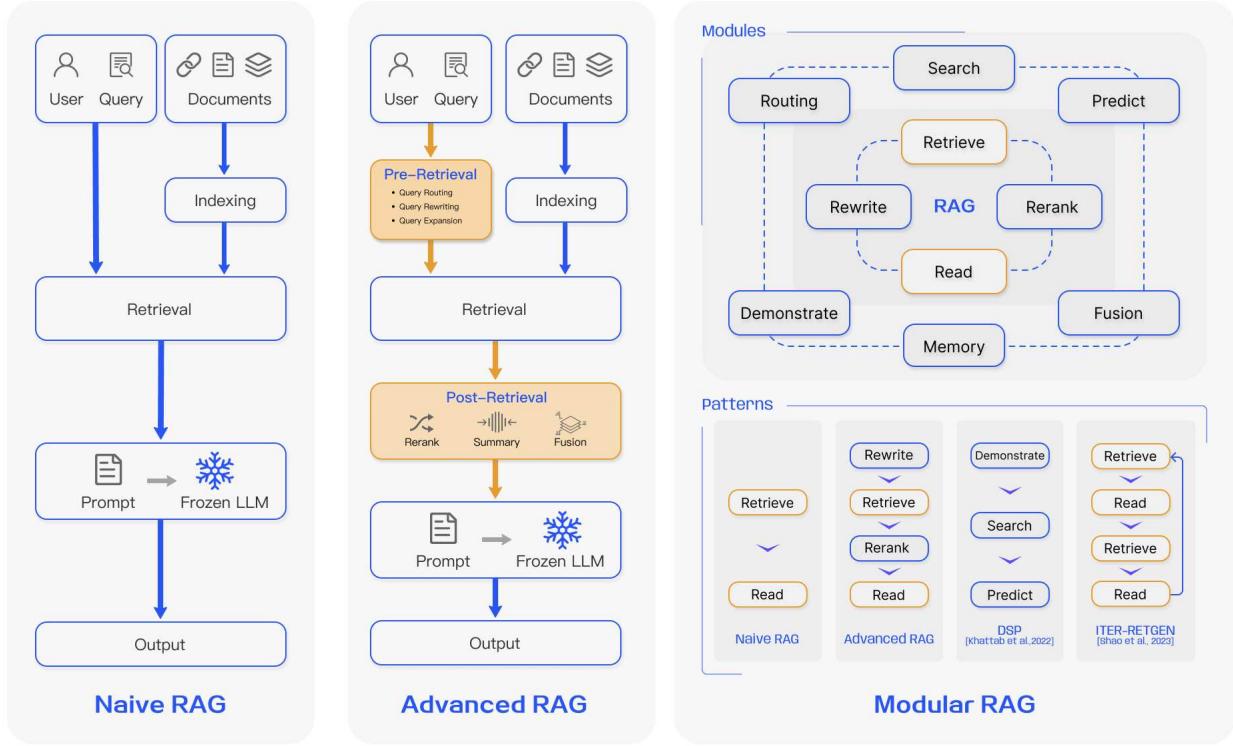


图3：RAG的三种范式比较

如图3所示。然而，模块化RAG并非独立存在。高级RAG是模块化RAG的一种专门形式，而朴素RAG本身则是高级RAG的一种特例。这三种范式之间的关系是继承和发展的关系。

### 新模块

**搜索模块。**与朴素/高级RAG中的相似性检索相比，搜索模块针对特定场景进行了定制，并结合了对额外语料库的直接搜索。这种集成是通过LLM生成的代码、SQL或Cypher等查询语言以及其他自定义工具实现的。这些搜索的数据源可以包括搜索引擎、文本数据、表格数据和知识图谱[Wang等，2023年]。

**记忆模块。**该模块利用LLM的记忆能力来指导检索。该方法涉及识别与当前输入最相似的记忆。

**Selfmem** [Chen et al., 2023b]利用检索增强生成器迭代地创建一个无限的内存池，将“原始问题”和“双重问题”结合起来。通过使用检索增强生成模型，它利用自身的输出来改进自身，文本在推理过程中与数据分布更加一致。因此，模型使用自己的输出而不是训练数据[Wang et al., 2022a]。

**融合。**RAG-Fusion [Raudaschl, 2023]通过多查询方法将用户查询扩展为多个不同的视角，从而增强传统搜索系统并解决其局限性。

这种方法不仅捕捉到用户寻求的显式信息，还揭示了更深层次的变革性知识。融合过程涉及对原始查询和扩展查询的并行向量搜索，智能重新排序以优化结果，并将最佳结果与新查询配对。这种复杂的方法确保搜索结果与用户的显式和隐含意图密切相关，从而实现更深入和相关的信息发现。

**路由.**RAG系统的检索过程利用不同领域、语言和格式的多样化来源，可以根据情况进行交替或合并[Liet al., 2023b]。查询路由决定对用户的查询进行后续操作，选项包括摘要、搜索特定数据库或将不同路径合并为单个响应。查询路由还选择适合查询的数据存储，可能包括各种来源，如向量存储、图数据库或关系数据库，或者索引的层次结构，例如用于多文档存储的摘要索引和文档块向量索引。查询路由的决策是预定义的，并通过LLMs调用执行，将查询指向选择的索引。

**预测 .**它解决了检索内容中的冗余和噪声的常见问题。该模块不直接从数据源检索，而是利用LLM生成所需的上下文[Yuet al., 2022]。与直接检索获得的内容相比，LLM生成的内容更有可能包含相关信息。

任务适配器。该模块专注于将RAG适应各种下游任务。UPRISE自动从预构建的数据池中检索零-shot任务输入的提示，从而增强了任务和模型的普适性[Chen et al., 2023a]。与此同时，PROMPTAGA-TOR [Dai et al., 2022]利用LLM作为few-shot查询生成器，并基于生成的数据创建特定任务的检索器。通过利用LLM的泛化能力，它能够以最少的示例开发特定任务的端到端检索器。

## 新模式

模块化RAG的组织结构非常适应，可以根据特定问题环境替换或重新排列模块。

Naive RAG和Advanced RAG都可以被视为由一些固定模块组成。如图3所示，Naive RAG主要由“检索”和“阅读”模块组成。高级RAG的典型模式是在Naive RAG的基础上添加“重写”和“重新排序”模块。然而，总体而言，模块化RAG具有更大的多样性和灵活性。

当前的研究主要探索了两种组织范式。第一种范式涉及添加或替换模块，而第二种范式则侧重于调整模块之间的组织流程。这种灵活性使得RAG过程能够有效地应对各种任务。

添加或替换模块。引入或替换模块的策略涉及保持检索-阅读过程的核心结构，同时整合额外的模块以增强特定功能。RRR模型[Ma等, 2023a]引入了重写-检索-阅读过程，利用LLM的性能作为重写模块的强化学习激励。这使得重写器能够微调检索查询，从而提高阅读器的下游任务性能。

类似地，模块可以在Generate-Read [Yu等, 2022]等方法中进行选择性交换，其中LLM的生成模块取代了检索模块。Recite-Read方法[Sun等, 2022]将外部检索转化为从模型权重中检索，要求LLM首先记忆任务特定信息，然后产生能够处理知识密集型自然语言处理任务的输出。

调整模块之间的流程。在模块流程调整的领域中，重点是增强语言模型和检索模型之间的交互。DSP [Khattab等, 2022]引入了演示-搜索-预测框架，将上下文学习系统视为一个显式程序，而不是一个最终任务提示，从而更有效地处理知识密集型任务。ITER-RETGEN [Shao等, 2023]方法利用生成的内容来指导检索，在Retrieve-Read-Retrieve-Read流程中迭代实现“检索增强生成”和“生成增强检索”。这种方法展示了一种创新的方式，利用一个模块的输出来提高另一个模块的功能。

## 优化RAG流程

检索过程的优化旨在提高RAG系统中信息的效率和质量。当前的研究重点是整合多样化的搜索技术，优化检索步骤，引入认知回溯，实施多功能查询策略，并利用嵌入相似性。这些努力共同致力于在RAG系统中实现检索效率和上下文信息深度之间的平衡。

**混合搜索探索.** RAG系统通过智能地整合各种技术（包括基于关键词的搜索、语义搜索和向量搜索）来优化其性能。这种方法利用每种方法的独特优势来适应不同的查询类型和信息需求，确保高度相关和上下文丰富的信息的一致检索。混合搜索的使用作为检索策略的强大补充，从而提高了RAG流程的整体效果。

**递归检索和查询引擎.** 递归检索在初始检索阶段获取较小的块以捕捉关键语义含义。随后，在过程的后期提供包含更多上下文信息的较大块给LLM。这种两步检索方法有助于在效率和提供上下文丰富响应之间取得平衡。

**StepBack-prompt方法.** 鼓励LLM远离具体实例，从而围绕更广泛的概念和原则进行推理[Zheng等, 2023年]。实验结果表明，当使用向后提示时，在各种具有挑战性的基于推理的任务中，性能显著提高，突出了它们对RAG过程的自然适应性。这些增强检索的步骤可以应用于生成向后提示的响应以及最终的问答过程。

**子查询.** 根据情况，可以采用各种查询策略，例如使用LlamaIndex等框架提供的查询引擎，利用树查询，利用向量查询，或者执行简单的顺序查询块。

**假设性文档嵌入.** HyDE的操作基于这样的信念：生成的答案在嵌入空间中可能比直接查询更接近。使用LLM，HyDE根据查询创建一个假设性文档（答案），将该文档嵌入，并使用得到的嵌入来检索与假设性文档相似的真实文档。这种方法不是基于查询寻求嵌入相似性，而是专注于从一个答案到另一个答案的嵌入相似性[Gao等, 2022年]。然而，当语言模型对主题不熟悉时，可能无法始终产生理想的结果，可能导致更多的错误实例。

## 4 检索

在RAG的背景下，从数据源中高效地检索相关文档至关重要。然而，创建一个熟练的检索器面临着重大挑战。本节将分为三个基本问题：1) 如何实现准确的语义表示？2) 有哪些方法可以对齐查询和文档的语义空间？3)

如何将检索器的输出与大型语言模型的偏好对齐？

#### 4.1 增强语义表示

在RAG中，语义空间是至关重要的，因为它涉及查询和文档的多维映射。在这个语义空间中，检索准确性对RAG的结果有着重要影响。本节将介绍构建准确语义空间的两种方法。

##### 块优化

在管理外部文档时，初始步骤涉及将它们分解成较小的块，以提取细粒度特征，然后嵌入以表示它们的语义。然而，嵌入过大或过小的文本块可能导致次优结果。因此，确定语料库中文档的最佳块大小对于确保检索结果的准确性和相关性至关重要。

选择适当的分块策略需要仔细考虑几个重要因素，例如索引内容的性质、嵌入模型及其最佳块大小、用户查询的预期长度和复杂性，以及特定应用对检索结果的利用。例如，选择分块模型应基于内容的长度——无论是较长还是较短。此外，不同的嵌入模型在不同的块大小下表现出不同的性能特征。例如，句子转换器在单个句子上表现更好，而文本嵌入ada-002在包含256或512个标记的块上表现出色。

此外，用户输入问题的长度和复杂性以及应用程序的特定需求（例如语义搜索或问题回答）都会影响分块策略的选择。这个选择可以直接受到所选LLM的标记限制的影响，需要调整块大小。实际上，获得精确的查询结果需要灵活应用不同的分块策略。没有一种“最佳”策略适用于所有情况，只有适合特定环境的最合适的策略。

当前RAG研究探索了各种旨在提高检索效率和准确性的块优化技术。其中一种方法涉及使用滑动窗口技术，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。另一种策略称为“small2big”方法，在初始搜索阶段使用小文本块，然后提供更大的相关文本块供语言模型处理。

抽象嵌入技术根据文档摘要（或总结）优先选择前K个检索结果，提供对整个文档内容的全面理解。此外，元数据过滤技术利用文档元数据增强过滤过程。一种创新的方法是图索引技术，将实体和关系转化为节点和连接，显著提高相关性，特别是在多跳问题的背景下。

这些多种方法的结合已经取得了显著的进展，提高了RAG的检索结果和性能。

##### 微调嵌入模型

一旦确定了适当的块大小，下一个关键步骤是使用嵌入模型将这些块和查询嵌入到语义空间中。

嵌入的有效性至关重要，因为它影响模型表示语料库的能力。最近的研究引入了一些重要的嵌入模型，如AngIE、Voyage、BGE等[Lee和Li, 2023年, VoyageAI, 2023年, BAAI, 2023年]。这些模型已经在大规模语料库上进行了预训练。然而，当应用于专业领域时，它们准确捕捉领域特定信息的能力可能有限。

此外，对嵌入模型进行任务特定的微调是确保模型理解用户查询的内容相关性的关键。没有进行微调的模型可能无法充分满足特定任务的要求。因此，微调嵌入模型对于下游应用非常重要。嵌入微调方法有两种主要范式。

**领域知识微调。**为了确保嵌入模型准确捕捉领域特定信息，必须利用领域特定数据集进行微调。这个过程与标准语言模型微调有所不同，主要体现在所涉及数据集的性质上。

通常，用于嵌入模型微调的数据集包括三个主要元素：查询、语料库和相关文档。模型利用这些查询在语料库中识别相关文档。然后，根据模型检索这些相关文档对查询的响应来评估模型的效果。

数据集构建、模型微调和评估阶段各自面临着不同的挑战。LlamaIndex [Liu, 2023]引入了一套关键的类和函数，旨在增强嵌入模型微调工作流程，从而简化这些复杂的过程。通过策划一个融入领域知识的语料库并利用提供的方法，可以熟练地微调嵌入模型，使其与目标领域的具体要求紧密对齐。

**下游任务的微调。**为下游任务微调嵌入模型是提高模型性能的关键步骤。在利用RAG进行这些任务的领域中，出现了创新的方法，通过利用LLM的能力来微调嵌入模型。例如，PROMPTAGATOR [Dai等, 2022]利用LLM作为少样本查询生成器，创建任务特定的检索器，解决了在数据稀缺领域中监督微调的挑战。另一种方法，LLM-Embedder [Zhang等, 2023a]，利用LLM生成跨多个下游任务的奖励信号的数据。检索器通过两种类型的监督信号进行微调：数据集的硬标签和LLM的软奖励。这种双信号方法促进了更有效的微调过程，使嵌入模型适应各种下游应用。

虽然这些方法通过整合领域知识和任务特定的微调来改善语义表示，但是检索器可能并不总是与某些大型语言模型完全兼容。为了解决这个问题，一些研究人员通过使用来自大型语言模型的反馈直接监督微调过程。这种直接监督旨在使检索器与大型语言模型更加接近，从而提高下游任务的性能。关于这个主题的更全面讨论见第4.3节。

## 4.2 查询和文档的对齐

在RAG应用的背景下，检索器可以使用单个嵌入模型来编码查询和文档，也可以为每个模型使用单独的模型。此外，用户的原始查询可能存在措辞不准确和缺乏语义信息的问题。因此，将用户查询的语义空间与文档的语义空间对齐是至关重要的。本节介绍了两种旨在实现这种对齐的基本技术。

### 查询重写

查询重写是一种对齐查询和文档语义的基本方法。

Query2 Doc和ITER-REGEN等方法利用LLMs将原始查询与附加指导信息相结合，创建一个伪文档[Wang等，2023c，Shao等，2023]。HyDE利用文本线索构建查询向量，生成一个捕捉关键模式的“假设”文档[Gao等，2022]。RR R引入了一个框架，颠倒了传统的检索和阅读顺序，重点关注查询重写[Ma等，2023a]。

STEP-BACKPROMPTING使LLMs能够根据高级概念进行抽象推理和检索[Zheng等，2023]。此外，多查询检索方法利用LLMs同时生成和执行多个搜索查询，有利于解决具有多个子问题的复杂问题。

### 嵌入转换

除了像查询重写这样的广泛策略外，还存在一些更细粒度的技术，专门用于嵌入转换。LlamaIndex [Liu, 2023]通过引入一个适配器模块来展示这一点，该模块可以在查询编码器之后集成。该适配器有助于微调，从而优化查询嵌入的表示，将其映射到与预期任务更密切对齐的潜在空间中。

SANTA [Li et al., 2023d]解决了将查询与结构化外部文档对齐的挑战，特别是在处理结构化和非结构化数据不一致性时。它通过两种预训练策略增强了检索器对结构化信息的敏感性：首先，通过利用结构化和非结构化数据之间的内在对齐来通知结构感知预训练方案中的对比学习；其次，通过实施掩码实体预测。后者利用了一种以实体为中心的掩码策略，鼓励语言模型预测和填充掩码实体，从而促进对结构化数据的更深入理解。

解决查询与结构化外部文档对齐的问题，特别是在处理结构化和非结构化数据之间的差异时，SANTA [Li等人，2023d]提出了解决方案。这种方法通过两种预训练策略改进了检索器识别结构化信息的能力：首先，利用结构化和非结构化数据之间的内在对齐来指导结构感知预训练方案中的对比学习；其次，采用掩码实体预测。后者使用以实体为中心的掩码策略，促使语言模型预测和完成被掩码的实体，从而提升对结构化数据的深入理解。

## 4.3 对齐检索器和LLM

在RAG流程中，通过各种技术提高检索命中率不一定能改善最终结果，因为检索到的文档可能与LLMs的特定要求不对齐。因此，本节介绍了两种旨在将检索器输出与LLMs的偏好对齐的方法。

### 微调检索器

几项研究利用LLMs的反馈信号来改进检索模型。例如，AAR [Yuet al., 2023b] 使用编码器-解码器架构为预训练的检索器引入监督信号。这是通过通过FiD交叉注意力分数识别LM首选的文档来实现的。随后，检索器经过精细调整采用硬负采样和标准交叉熵损失。

最终，经过改进的检索器可以直接应用于提高未见目标LLMs的性能在目标任务中。此外，有人认为LLMs可能更喜欢关注易读而不是信息丰富的文档。

REPLUG [Shiet al., 2023] 利用检索器和LLM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。这种简单而有效的训练方法提高了检索模型的性能通过使用LM作为监督信号，消除了特定的交叉注意力机制的需求。

UPRISE [Chenget al., 2023a] 还使用冻结的LLMs来微调提示检索器。LLM和检索器都将提示-输入对作为输入，并利用LLM提供的分数来监督检索器的训练，有效地将LLM视为数据集标签。此外，Atlas [Izacardet al., 2022] 提出了四种监督微调嵌入模型的方法：

- 注意力蒸馏。这种方法在输出过程中利用LLM生成的交叉注意力分数来蒸馏模型的知识。
- EMDR2。通过使用期望最大化算法，该方法使用检索到的文档作为潜在变量来训练模型。
- 困惑度蒸馏直接使用生成的标记的困惑度作为指标来训练模型。

•循环. 该方法提出了一种基于文档删除对LLM预测影响的新型损失函数，提供了一种有效的训练策略，以更好地适应特定任务的模型。

这些方法旨在改善检索器和LLM之间的协同作用，从而提高检索性能并更准确地回答用户的查询。

### 适配器

微调模型可能面临挑战，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约束。因此，一些方法选择引入外部适配器来帮助对齐。

PRCA通过上下文提取阶段和奖励驱动阶段训练适配器。然后，使用基于标记的自回归策略优化检索器的输出[Yang等，2023b]。标记过滤方法利用交叉注意力分数高效过滤标记，仅选择最高得分的输入标记[Berchansky等，2023]。RECOMP引入了摘要生成的抽取式和生成式压缩器。

这些压缩器要么选择相关的句子，要么合成文档信息，创建适用于多文档查询的摘要[Xu等人，2023a]。

此外，PKG通过指令微调引入了一种将知识整合到白盒模型中的创新方法[Luo等人，2023]。在这种方法中，检索模块直接替换为根据查询生成相关文档。这种方法有助于解决微调过程中遇到的困难，并提高模型性能。

## 5 生成

RAG的一个关键组成部分是其生成器，负责将检索到的信息转化为连贯流畅的文本。与传统语言模型不同，RAG的生成器通过整合检索数据来提高准确性和相关性。在RAG中，生成器的输入不仅包括典型的上下文信息，还包括通过检索器获取的相关文本片段。这种全面的输入使生成器能够深入理解问题的上下文，从而产生更具信息量和上下文相关的回答。

此外，生成器通过检索到的文本来指导，以确保生成内容与获取的信息之间的连贯性。多样化的输入数据导致在生成阶段进行有针对性的努力，所有这些努力都旨在改进大型模型对来自查询和文档的输入数据的适应性。在接下来的小节中，我们将探讨生成器的介绍，深入研究检索后处理和微调的方面。

### 5.1 冻结LLM的检索后处理

在无法调整的LLM领域，许多研究依赖于像GPT-4 [OpenAI, 2023]这样的成熟模型，以利用其全面的内部知识，系统地合成来自各种文档的检索信息。

然而，这些大型模型仍然存在挑战，包括上下文长度的限制和对冗余信息的敏感性。为了解决这些问题，某些研究工作已将重点转向检索后处理。

后续检索处理涉及对由检索器从大型文档数据库中检索到的相关信息进行处理、过滤或优化。其主要目标是提高检索结果的质量，使其更加符合用户需求或后续任务。它可以被视为检索阶段获取的文档的再处理。后续检索处理中常见的操作通常包括信息压缩和结果重新排序。

### 信息压缩

检索器擅长从庞大的知识库中检索相关信息，但管理检索文档中的大量信息是一项挑战。当前的研究旨在扩展大型语言模型的上下文长度以解决这个问题。然而，当前的大型模型仍然在上下文限制方面存在困难。因此，在某些情况下，压缩信息变得必要。信息压缩对于减少噪音、解决上下文长度限制和增强生成效果非常重要。

PRCA通过训练信息提取器 [Yang等人，2023b] 来解决这个问题。在上下文提取阶段，当提供输入文本  $S$  输入时，它能够生成一个表示从输入文档中提取的精简上下文的输出序列  $C$  提取。训练过程旨在最小化  $C$  提取和实际上下文  $C$  真相之间的差异。

类似地，RECOMP采用了一种可比较的方法，通过使用对比学习训练信息压缩器 [Xu等人，2023a]。每个训练数据点包括一个正样本和五个负样本，编码器在整个过程中使用对比损失进行训练 [Karpukhin等人，2020]。

另一项研究采用了一种不同的方法，旨在减少文档数量以提高模型答案的准确性。在[Ma等人，2023b]的研究中，他们提出了“过滤-重新排序”范式，将大型语言模型 (LLMs) 和小型语言模型 (SLMs) 的优势结合起来。在这个范式中，SLMs充当过滤器，而LLMs则充当重新排序代理。研究表明，指导LLMs重新排列SLMs识别出的具有挑战性的样本，可以显著改善各种信息提取 (IE) 任务。

### 重新排序

重新排序模型在优化从检索器检索到的文档集合方面起着关键作用。当引入额外的上下文时，语言模型往往面临性能下降的问题，而重新排序有效地解决了这个问题。核心概念涉及重新排列文档记录，以优先考虑最相关的项目，从而限制文档的总数。这不仅解决了检索过程中上下文窗口扩展的挑战，还增强了检索效率和响应能力。

重新排序模型在信息检索过程中扮演着双重角色，既是一个

优化器和细化器。它为后续语言模型处理提供了更有效和准确的输入[Zhuang等, 2023]。

上下文压缩被纳入重新排序过程中，以提供更精确的检索信息。这种方法涉及减少单个文档的内容并过滤整个文档，最终目标是在搜索结果中呈现最相关的信息，以更专注和准确地显示相关内容。

## 5.2 对RAG进行微调的大型语言模型

优化RAG模型内的生成器是其架构的关键方面。生成器的作用是获取检索到的信息并生成相关的文本，形成模型的最终输出。生成器的优化旨在确保生成的文本既自然又有效地利用检索到的文档，以更好地满足用户的查询需求。

在标准的大型语言模型生成任务中，输入通常包括一个查询。RAG通过将检索器检索到的各种文档（结构化/非结构化）纳入输入中，脱颖而出。这些附加信息可以显著影响模型的理解，特别是对于较小的模型而言。在这种情况下，微调模型以适应查询和检索到的文档的输入变得至关重要。在将输入呈现给微调模型之前，通常会对检索器检索到的文档进行后处理。需要注意的是，RAG中生成器的微调方法与大型语言模型的一般微调方法一致。接下来，我们将简要介绍一些涉及数据（格式化/非格式化）和优化函数的代表性工作。

### 通用优化过程

作为通用优化过程的一部分，训练数据通常包括输入-输出对，旨在训练模型在给定输入  $x$  的情况下生成输出  $y$ 。在Self-Mem的工作中[Cheng等人, 2023b]，采用了传统的训练过程，给定输入  $x$ ，检索相关文档  $z$ （在论文中选择Top-1），并在集成  $(x, z)$  后，模型生成输出  $y$ 。该论文利用了两种常见的微调范式，即联合编码器和双编码器[Arora等人, 2023, Wang等人, 2022b, Lewis等人, 2020, Xia等人, 2019, Cai等人, 2021, Cheng等人, 2022]。

在联合编码器范式中，使用基于编码器-解码器的标准模型。在这里，编码器最初对输入进行编码，解码器通过注意机制以自回归方式组合编码结果生成标记。另一方面，在双编码器范式中，系统设置了两个独立的编码器，每个编码器分别对输入（查询、上下文）和文档进行编码。生成的输出按顺序经过解码器的双向交叉注意处理。这两种架构都利用Transformer [Vaswani等, 2017]作为基础模块，并使用负对数似然损失进行优化。

### 利用对比学习

在为语言模型准备训练数据的阶段，通常会创建输入和输出的交互对。这种传统方法可能导致“曝光偏差”，即模型只在个别正确输出示例上进行训练，从而限制了其对可能输出范围的暴露。这种限制可能会阻碍模型在现实世界中的性能，因为它会导致模型过度拟合训练集中的特定示例，从而降低其在不同上下文中的泛化能力。

为了减轻曝光偏差，SURGE [Kang等, 2023年]提出了使用图文对比学习的方法。该方法包括对比学习目标，促使模型生成一系列合理且连贯的回答，扩展到训练数据中未遇到的情况。这种方法对于减少过拟合和增强模型的泛化能力至关重要。

对于涉及结构化数据的检索任务，SANTA框架 [Li等, 2023年]实施了三部分的训练方案，以有效地捕捉结构和语义细微差别。初始阶段侧重于检索器，利用对比学习来优化查询和文档的嵌入。

随后，生成器的初步训练阶段采用对比学习来将结构化数据与非结构化文档描述对齐。在生成器训练的进一步阶段中，模型认识到实体语义在文本数据检索的表示学习中的关键作用，正如[Sciavolino等, 2021年, Zhang等, 2019年]所强调的。这个过程从识别结构化数据中的实体开始，然后在生成器的输入数据中对这些实体应用掩码，为模型预测这些被掩码的元素铺平道路。

训练计划随着模型学习逐渐进行，通过利用上下文信息来重建被屏蔽实体。这个练习培养了模型对文本数据结构语义的理解，并促进了结构化数据中相关实体的对齐。

总体优化目标是训练语言模型准确恢复被遮盖的范围，从而丰富其对实体语义的理解[Ye等, 2020]。

### RAG中的增强

本节围绕增强阶段、增强数据来源和增强过程三个关键方面进行结构化。这些方面阐明了对RAG发展至关重要的关键技术。图4呈现了RAG核心组件的分类。

## 6.1 增强阶段中的RAG

RAG是一个知识密集型的工作，涵盖了语言模型训练的预训练、微调和推理阶段的各种技术方法。

### 预训练阶段

在预训练阶段，研究人员通过探索方法来增强开放领域问答（QA）的PTM

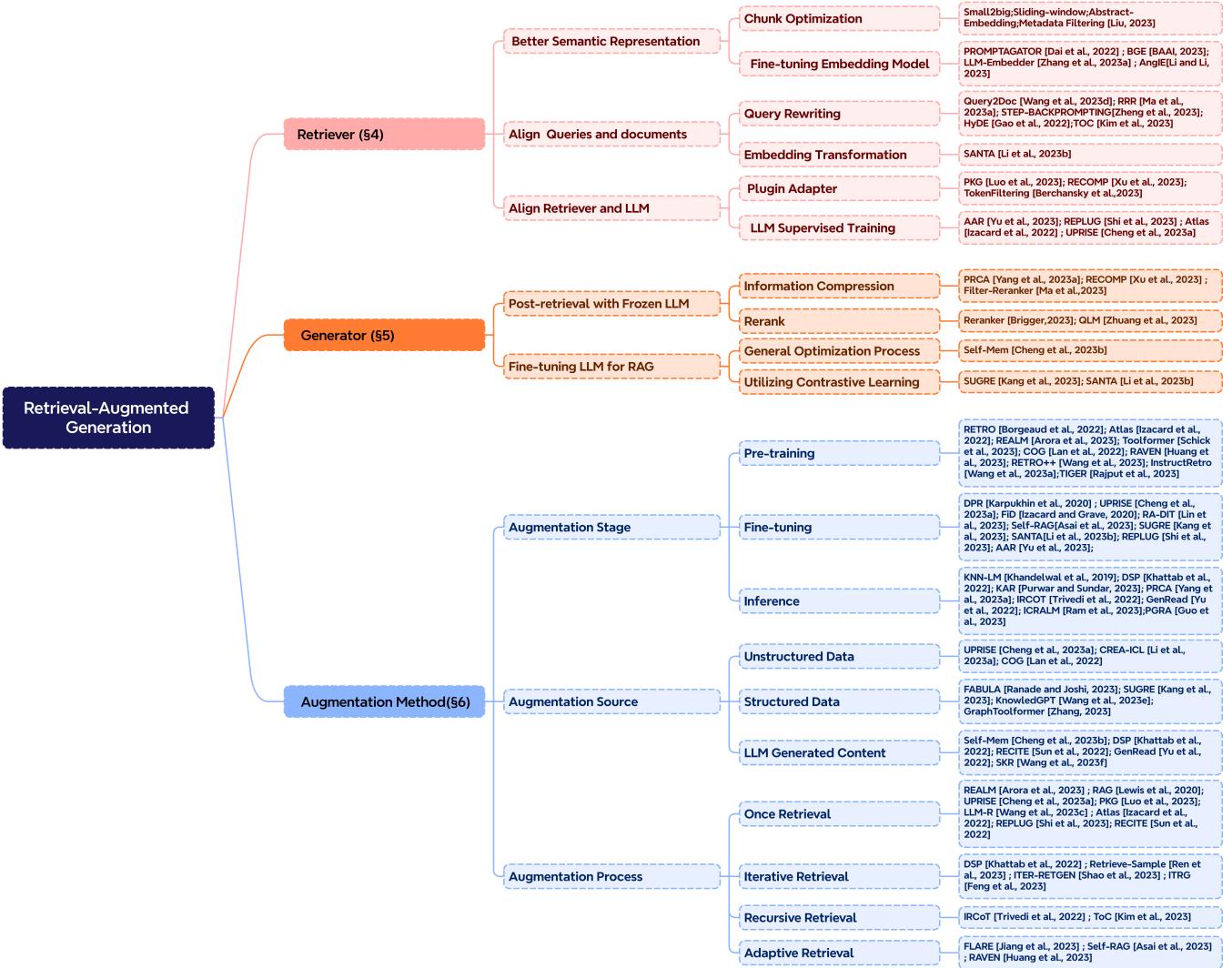


图4：RAG核心组件的分类

基于检索的策略。REALM模型采用了一种结构化、可解释的方法进行知识嵌入，将预训练和微调作为检索-预测工作流程，并嵌入到掩码语言模型（MLM）框架中[Arora等，2023]。

RETRO [Borgeaud等，2022]利用检索增强进行大规模预训练，从零开始，减少了模型参数，同时在困惑度方面超过了标准的GPT模型。RETRO通过额外的编码器来处理从外部知识库检索到的实体特征，构建在GPT模型的基础结构上，使其与众不同。

Atlas [Izacard等，2022]还将检索机制引入到T5架构[Raffel等，2020]的预训练和微调阶段。它使用预训练的T5来初始化编码器-解码器语言模型，并使用预训练的Contriever作为密集检索器，提高了复杂语言建模任务的效率。

此外，COG [Lan等，2022]引入了一种新的文本生成方法，模拟从现有集合中复制文本片段。利用高效的向量搜索工具，COG计算并索引文本片段的上下文相关的有意义的表示，展示了在问答和领域适应等领域相比RETRO更优越的性能。研究人员正在将RAG方法扩展到预训练的最大模型，RETRO++是这一趋势的典型代表，通过增加模型参数的规模同时保持或提升性能。

实证证据强调了文本生成质量、事实准确性、毒性减少以及知识密集型应用（如开放领域问答）中下游任务能力的显著改善。这些结果表明将检索机制整合到预训练模型中的重要性。

自回归语言模型的训练构成了一个有前途的途径，将复杂的检索技术与广泛的语言模型相结合，以产生更精确和高效的语言生成。

增强预训练的好处包括一个强大的基础模型，其困惑度、文本生成质量和任务特定性能均优于标准的GPT模型，同时利用更少的参数。这种方法特别擅长处理知识密集型任务，并通过在专门的语料库上进行训练来促进领域特定模型的开发。

然而，这种方法面临着一些挑战，如需要大量的预训练数据集和资源，以及随着模型规模增加而减少的更新频率。尽管存在这些障碍，但这种方法在模型的韧性方面具有显著优势。一旦训练完成，增强检索模型可以独立于外部库运行，提高生成速度和操作效率。所确定的潜在收益使得这种方法在人工智能和机器学习领域成为一个引人注目的持续研究和创新课题。

## 微调阶段

RAG和Fine-tuning是增强LLMs的强大工具，将两者结合可以满足更具体的场景需求。一方面，微调允许检索具有独特风格的文档，实现更好的语义表达并对齐查询和文档之间的差异。这确保了检索器的输出更适合当前场景。另一方面，微调可以满足生成需求，进行风格化和有针对性的调整。此外，微调还可以用于对齐检索器和生成器，以改善模型的协同效应。

微调检索器的主要目标是通过直接微调嵌入模型使用语料库[Liu, 2023]来提高语义表示的质量。

通过通过反馈信号将检索器的能力与LLMs的偏好对齐，可以更好地协调两者[Yu等, 2023b, Izacard等, 2022, Yang等, 2023b, Shi等, 2023]。针对特定下游任务微调检索器可以提高适应性[cite]。引入任务不可知的微调旨在增强检索器在多任务场景中的通用性[Cheng等, 2023a]。

微调生成器可能会导致更加风格化和定制化的输出。一方面，它允许对不同的输入数据格式进行专门的适应。例如，将LLMs微调以适应知识图谱的结构[Kang等, 2023年]，文本对的结构[Kang等, 2023年, Cheng等, 2023年]以及其他特定结构[Li等, 2023年]。另一方面，通过构建指令数据集，可以要求LLMs生成特定格式的内容。例如，在自适应或迭代检索场景中，LLMs被微调为生成将有助于确定下一步行动时机的内容[Jiang等, 2023年, Asai等, 2023年]。

通过协同微调检索器和生成器，我们可以增强模型的泛化能力

并避免由于分别训练它们而导致的过拟合问题。然而，联合微调也会导致资源消耗增加。RA-DIT [Lin等, 2023]提出了一种轻量级的双指令调整框架，可以有效地为任何LLM添加检索能力。

尽管具有优势，微调也有一些限制，包括需要专门的RAG微调数据集和大量的计算资源。

然而，这个阶段允许根据特定需求和数据格式定制模型，可能减少资源使用量，同时仍能够微调模型的输出风格。

总之，微调阶段对于RAG模型适应特定任务至关重要，可以对检索器和生成器进行优化。尽管资源和数据集要求带来了挑战，但这个阶段增强了模型的多功能性和适应性。因此，RAG模型的策略性微调是开发高效和有效的检索增强系统的关键组成部分。

## 推理阶段

在RAG模型中，推理阶段至关重要，因为它需要与LLM进行广泛的集成。传统的RAG方法，也称为Naive RAG，在这个阶段将检索内容纳入生成过程中进行引导。

为了克服Naive RAG的局限性，先进的技术在推理过程中引入更丰富的上下文信息。DSP框架[Khattab等, 2022]利用自然语言文本在冻结的LM和检索模型(RM)之间进行复杂的交流，丰富了上下文，从而改善了生成结果。PKG [Luo等, 2023]方法通过引入知识引导模块，使LLM能够检索相关信息而不修改LLM的参数，从而实现更复杂的任务执行。CREA-ICL [Li等, 2023b]通过同步检索跨语言知识来增强上下文，而RE-CITE [Sun等, 2022]则通过直接从LLM中抽样段落来生成上下文。

在推理过程中，对RAG过程的进一步改进可以看到，这些方法适用于需要多步推理的任务。ITRG [Feng等, 2023]通过迭代检索信息来识别正确的推理路径，从而提高任务的适应性。ITER-RETEGEN [Shao等, 2023]采用迭代策略，在检索和生成之间进行循环过程，交替进行“检索增强生成”和“生成增强检索”。对于非知识密集型(NKI)任务，PGRA [Guo等, 2023]提出了一个两阶段的框架，首先是一个任务不可知的检索器，然后是一个基于提示的重新排序器，用于选择和优先考虑证据。相比之下，IRCOT [Trivedi等, 2022]将RAG与思维链(CoT)方法相结合，通过CoT引导的检索和检索引导的CoT过程交替进行，显著提高了GPT-3的性能。

各种问答任务。

本质上，这些推理阶段的增强提供了轻量级、经济高效的替代方案，利用了预训练模型的能力，而无需进一步训练。主要优势是在提供上下文相关信息以满足特定任务需求的同时，保持静态LLM参数。然而，这种方法并非没有局限性，它需要精心的数据处理和优化，并受到基础模型固有能力的限制。为了有效应对多样化的任务需求，这种方法通常与逐步推理、迭代检索和自适应检索策略等程序优化技术相结合。

## 6.2 增强源

RAG模型的有效性受到增强数据源的选择的严重影响。不同层次的知识和维度需要不同的处理技术。它们被归类为非结构化数据、结构化数据和由LLMs生成的内容。具有不同增强方面的代表性RAG研究的技术树如图5所示。以三种不同色调着色的叶子代表使用各种类型数据进行增强：非结构化数据、结构化数据和由LLMs生成的内容。该图清楚地显示，最初增强主要通过非结构化数据（如纯文本）实现。随后，这种方法扩展到使用结构化数据（例如知识图谱）进行进一步改进。最近的研究趋势表明，越来越多地利用LLMs生成的内容进行检索和增强。

### 增强的非结构化数据

非结构化文本是从语料库中收集的，例如用于大型模型微调的提示数据 [Cheng等, 2023a] 和跨语言数据 [Li等, 2023b]。检索单元的范围从标记（例如kNN-LM [Khandelwal等, 2019]）到短语（例如NPM, COG [Lee等, 2020, Lan等, 2022]）和文档段落，更精细的粒度可以提供更高的检索精度，但会增加检索复杂性。

FLARE [Jiang等, 2023b] 引入了一种主动检索方法，由语言模型生成低概率词触发。它创建一个临时句子用于文档检索，然后使用检索到的上下文重新生成句子以预测后续句子。RETRO使用前一个块来检索块级别的最近邻，结合前一个块的上下文，引导下一个块的生成。为了保持因果关系，下一个块  $C_i$  的生成仅利用前一个块  $N(C_{i-1})$  的最近邻，而不是  $N(C_i)$ 。

### 增强结构化数据

结构化数据，如知识图谱（KGs），提供高质量的上下文并减轻模型的幻觉。RET-LLMs [Modarressiet al., 2023] 从过去的对话中构建知识图谱内存，以供将来参考。SUGRE [Kang et al., 2023] 使用图神经网络（GNNs）对相关的KG子图进行编码，通过多模态对比学习确保检索到的事实与生成的文本之间的一致性。Knowl-

edGPT [Wang et al., 2023d] 生成KB搜索查询并将知识存储在个性化数据库中，增强了RAG模型的知识丰富性和上下文性。

### RAG中的LLMs生成内容

针对RAG中外部辅助信息的局限性，一些研究专注于利用LLMs的内部知识。SKR [Wang et al., 2023e] 将问题分类为已知或未知，并选择性地应用检索增强。GenRead [Yuet al., 2022] 用LLM生成器替换了检索器，并发现LLM生成的上下文通常包含更准确的答案，因为它更好地与因果语言建模的预训练目标对齐。Selfmem [Chen et al., 2023b] 使用检索增强的生成器迭代地创建一个无限的记忆池，使用记忆选择器选择作为原始问题的双重问题的输出，从而自我增强生成模型。

这些方法强调了在RAG中创新数据源利用的广度，旨在提高模型性能和任务效果。

## 6.3 增强过程

在RAG领域，标准做法通常涉及一个检索步骤，然后是生成步骤，这可能导致效率低下。一个值得注意的问题，被称为“中间丢失”现象，在单次检索中产生冗余内容，可能会稀释或矛盾于关键信息，从而降低生成质量 [Li等, 2023a]。此外，这种单一检索通常对于需要多步推理的复杂问题来说是不够的，因为它提供了有限的信息范围 [Yoran等, 2023]。

如图5所示，为了解决这些挑战，当代研究提出了改进检索过程的方法：迭代检索、递归检索和自适应检索。迭代检索允许模型进行多次检索循环，增强所获得信息的深度和相关性。递归检索过程中，一个检索操作的结果被用作后续检索的输入。它有助于深入研究相关信息，特别是在处理复杂或多步查询时。递归检索通常在需要逐步逼近最终答案的情况下使用，例如学术研究、法律案例分析或某些类型的数据挖掘任务。另一方面，自适应检索提供了一种动态调整机制，根据不同任务和环境的特定需求来定制检索过程。

### 迭代检索

在RAG模型中，迭代检索是一个过程，根据初始查询和迄今为止生成的文本，重复收集文档，为LLM提供更全面的知识库 [Borgeaud等, 2022年, Arora等, 2023年]。通过多次检索迭代，这种方法已被证明可以通过提供额外的上下文参考来增强后续答案生成的鲁棒性。然而，它可能会遭受语义不连续和无关信息的积累，因为

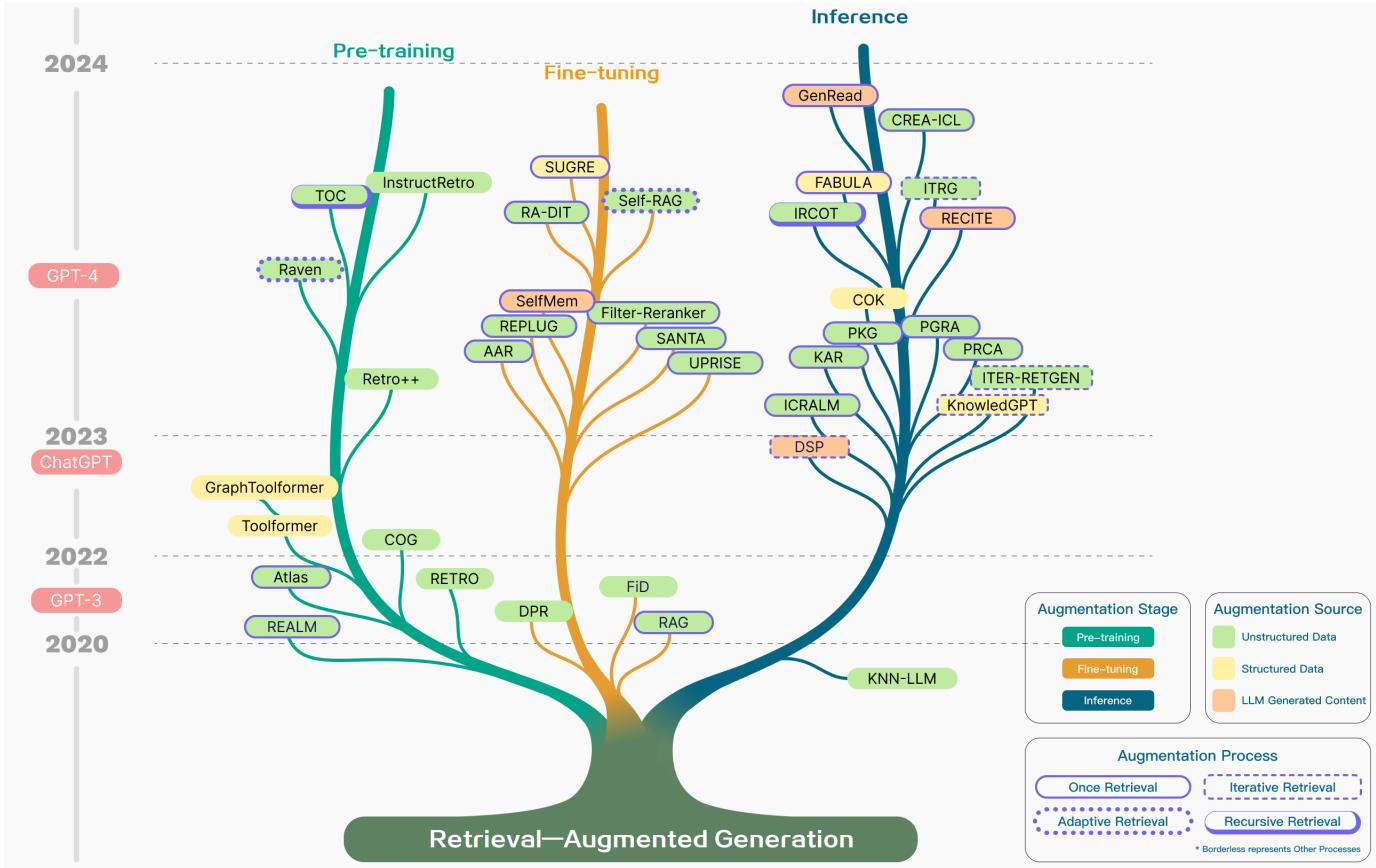


图5：具有不同增强方面的代表性RAG研究的技术树

它通常依赖于一系列n个标记来标示生成文本和检索文档之间的边界。

为了解决特定的数据场景，使用了递归检索和多跳检索技术。递归检索涉及使用结构化索引以分层方式处理和检索数据，这可能包括在执行检索之前对文档或长PDF的摘要进行总结。随后，在文档内进行二次检索来细化搜索，体现了该过程的递归性质。相比之下，多跳检索旨在深入挖掘图结构化数据源，提取相互关联的信息。

该过程通过根据先前搜索的结果逐步改进搜索查询来迭代地优化搜索查询。递归检索旨在通过反馈循环逐渐收敛于最相关的信息，IRCoT [Trivedi et al., 2022]使用思维链来指导检索过程，并通过获得的检索结果改进CoT。ToC [Kim et al., 2023]创建了一个澄清树，系统地优化查询中的模糊部分。它在用户的需求从一开始就不完全清楚或所寻找的信息非常专业化或微妙的复杂搜索场景中特别有用。该过程的递归性质允许持续学习和适应用户的需求，通常会导致对搜索结果的满意度提高。

此外，一些方法将检索和生成步骤整合在一起。ITER-RETEGEN [Shao et al., 2023]采用协同方法，利用“检索增强生成”和“生成增强检索”来复制特定信息的任务。该模型利用输入任务所需的内容作为上下文基础，用于检索相关知识，从而促进后续迭代中生成改进的响应。

### 递归检索

递归检索通常用于信息检索和自然语言处理，以提高搜索结果的深度和相关性。

### 自适应检索

自适应检索方法，例如Flare和Self-RAG [Jian et al., 2023b, Asai et al., 2023]，通过使LLMs能够主动确定检索的最佳时机和内容，从而提高信息的效率和相关性。

这些方法是LLMs在其操作中采用主动判断的更广泛趋势的一部分，如AutoGPT、Toolformer和Graph-Toolformer [Yang et al., 2023c, Schicket et al., 2023, Zhang, 2023]所示。

例如，Graph-Toolformer将其检索过程分为不同的步骤，LLMs主动使用检索器，应用Self-Ask技术，并使用少样本提示来发起搜索查询。这种主动的立场使LLMs能够决定何时搜索必要的信息，类似于代理人如何利用工具。

WebGPT [Nakano et al., 2021]在文本生成过程中集成了一个强化学习框架，通过使用搜索引擎自主训练GPT-3模型。

它使用特殊标记来导航这个过程，以便进行搜索引擎查询、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（由生成术语的概率指示）来自动化时间检索[Jian et al., 2023b]。当概率低于某个阈值时，会激活检索系统以收集相关信息，从而优化检索循环。

Self-RAG [Asai et al., 2023]引入了“反思标记”，允许模型自省其输出。这些标记有两种类型：“检索”和“评论”。模型自主决定何时激活检索，或者可以预先定义一个阈值来触发该过程。在检索过程中，生成器在多个段落上进行片段级别的波束搜索，以得到最连贯的序列。评论分数用于更新细分分数，可以在推理过程中灵活调整这些权重，以调整模型的行为。Self-RAG的设计消除了对额外分类器或依赖自然语言推理（NLI）模型的需求，从而简化了何时启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。

由于其日益普及，LLM优化受到了广泛关注。诸如提示工程、微调（FT）和RAG等技术都具有不同的特点，如图6所示。虽然提示工程利用了模型的固有能力，但优化LLM通常需要同时应用RAG和FT方法。RAG和FT之间的选择应基于具体场景的要求和每种方法的固有属性。表1详细比较了RAG和FT。

## 6.4 RAG与Fine-Tuning

RAG就像给模型提供了一个定制的信息检索教材，非常适合特定的查询。另一方面，FT就像学生随着时间内化知识，更适合复制特定的结构、风格或格式。FT可以通过强化基础模型知识、调整输出和教授复杂指令来提高模型性能和效率。然而，它不适用于整合新知识或快速迭代新的用例。

RAG和FT这两种方法并不是互斥的，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，它们的联合使用可能会产生最佳性能。优化过程

涉及RAG和FT可能需要多次迭代才能达到满意的结果。

## 7 RAG评估

RAG在自然语言处理（NLP）领域的快速发展和广泛应用，已经将RAG模型的评估推到了LLMs社区研究的前沿。这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。

从历史上看，RAG模型的评估主要集中在特定的下游任务上。这些评估使用适合任务的已建立的度量标准。例如，问答评估可能依赖于EM和F1分数[Wang等, 2022a, Shi等, 2023, Feng等, 2023, Ma等, 2023a]，而事实核查任务通常以准确性作为主要指标[Lewis等, 2020, Izacard等, 2022, Shao等, 2023]。类似于RALLE这样的工具，专为RAG应用的自动评估而设计，同样基于这些特定任务的度量标准[Hoshi等, 2023]。

尽管如此，有关评估RAG模型独特特征的研究明显不足，只有少数相关研究。

下一节将重点从任务特定的评估方法和指标转向基于其独特属性的现有文献综合。这次探索涵盖了RAG评估的目标、评估这些模型的方面以及可用于此类评估的基准和工具。目的是提供对RAG模型评估的全面概述，概述专门解决这些先进生成系统的独特方面的方法论。

### 7.1 评估目标

对RAG模型的评估主要围绕检索和生成模块这两个关键组成部分展开。这种划分确保了对提供的上下文质量和生成内容质量的全面评估。

#### 检索质量

评估检索质量对于确定检索器组件提取的上下文的有效性至关重要。从搜索引擎、推荐系统和信息检索系统领域采用标准度量指标来衡量RAG检索模块的性能。常用的度量指标包括命中率、MRR和NDCG[Liu, 2023, Nguyen, 2023]。

#### 生成质量

生成质量的评估集中在生成器从检索到的上下文中综合一致且相关的答案的能力上。这种评估可以根据内容的目标进行分类：无标签和有标签的内容。对于无标签的内容，评估包括生成的答案的忠实度、相关性和无害性。相反，对于有标签的内容，重点是所生成信息的准确性。

表1：RAG和Fine-Tuning的比较

特征比较	RAG	Fine-Tuning
知识更新	直接更新检索知识库可以确保信息保持最新，无需频繁重新训练，非常适合动态数据环境。	存储静态数据，需要重新训练以进行知识和数据更新。
外部知识	擅长利用外部资源，特别适合访问文档或其他结构化/非结构化数据库。	可以用于将预训练的外部获取的知识与大型语言模型对齐，但对于经常变化的数据源可能不太实用。
数据处理	涉及最小化数据处理和处理。	依赖于高质量数据集的创建，有限的数据集可能不会带来显著的性能改进。
模型定制	侧重于信息检索和整合外部知识，但可能无法完全定制模型行为或写作风格。	允许根据特定的语气或术语调整LLM的行为、写作风格或特定领域知识。
可解释性	响应可以追溯到特定的数据源，提供更高的可解释性和可追溯性。	类似于黑盒子，模型为什么以某种方式反应并不总是清楚，导致相对较低的可解释性。
计算资源	依赖计算资源来支持与数据库相关的检索策略和技术。此外，它还需要维护外部数据源的集成和更新。	高质量训练数据集的准备和整理，定义微调目标，并提供相应的计算资源是必要的。
延迟要求	涉及数据检索，可能导致更高的延迟。	微调后的大型语言模型可以在没有检索的情况下进行响应，从而降低延迟。
减少幻觉	由于每个答案都基于检索到的证据，因此不容易产生幻觉。	通过基于特定领域数据训练模型可以帮助减少幻觉，但在面对不熟悉的输入时仍可能出现幻觉。
伦理和隐私问题	存储和从外部数据库检索文本引发了伦理和隐私问题。	由于训练数据中可能存在敏感内容，因此可能引发伦理和隐私问题。

模型[Liu, 2023]。此外，可以通过手动或自动评估方法[Liu, 2023, Lan等, 2022, Leng等, 2023]进行检索和生成质量评估。

## 7.2 评估方面

当代RAG模型的评估实践强调三个主要质量分数和四个基本能力，共同为RAG模型的检索和生成这两个主要目标提供评估信息。

### 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

从不同的角度评估RAG模型在信息检索和生成过程中的效率[Eset *et al.*, 2023, Saad-Falconet *et al.*, 2023, Jarvis and Allard, 2023]。质量分数-上下文相关性、答案忠实度和答案相关性-评估了RAG模型在信息检索和生成过程中的效率，从各个角度确保检索到的上下文的准确性和特异性，以减少与无关内容相关的处理成本。

答案的忠实性确保生成的答案与检索到的上下文保持一致，保持一致性

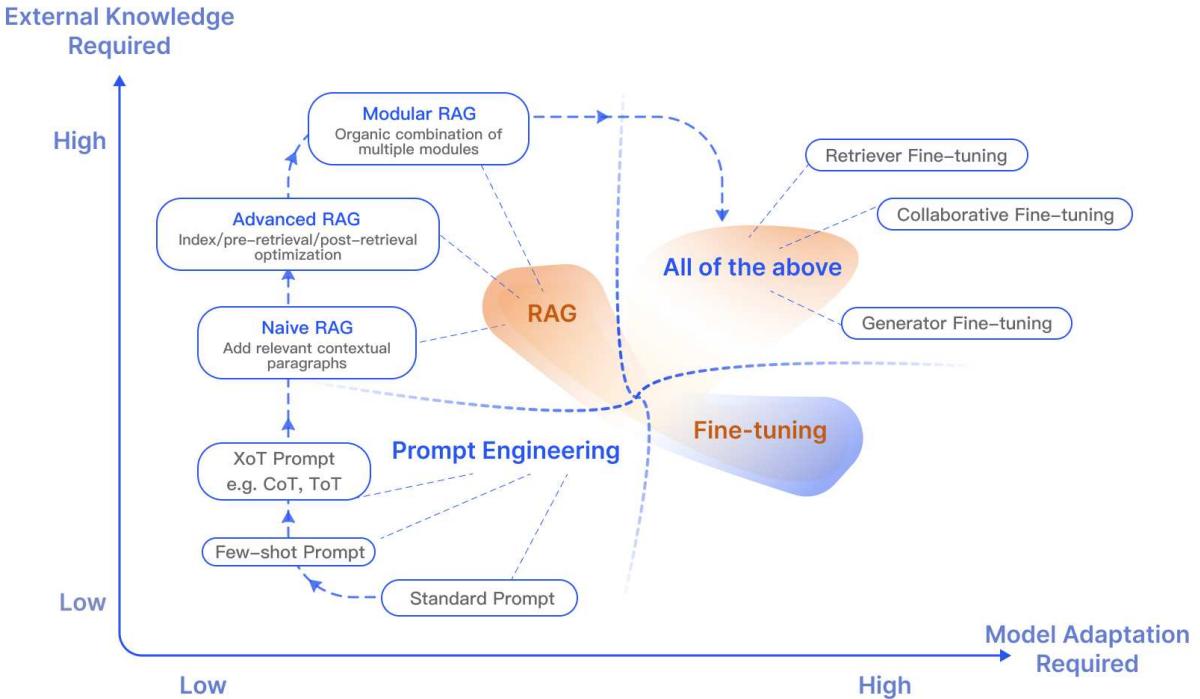


图6：RAG与其他模型优化方法的比较

并避免矛盾。

答案的相关性要求生成的答案直接与提出的问题相关，有效地回答核心问题。

#### 所需能力

RAG评估还包括四种能力，表明其适应性和效率：噪声鲁棒性，负面拒绝，信息整合和反事实鲁棒性[Chen等，2023b, Liu等，2023b]。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，影响质量评分。

**噪声鲁棒性**评估模型处理与问题相关但缺乏实质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中不包含回答问题所需知识时的辨别能力。

**信息整合**评估模型在综合多个文档中合成信息以解决复杂问题方面的熟练程度。

**反事实鲁棒性**测试模型在识别和忽略已知不准确信息方面的能力，即使被告知可能存在的错误信息。

上下文相关性和噪声鲁棒性对于评估检索质量非常重要，而回答的忠实度、回答的相关性、负面拒绝、信息整合和反事实鲁棒性对于评估生成质量非常重要。

生成质量的重要性。

每个评估方面的具体指标总结在表2中。需要认识到这些指标是从相关工作中得出的，它们是传统的度量标准，尚不能代表一种成熟或标准化的方法来量化RAG评估方面。尽管这里没有包括，但一些评估研究还开发了针对RAG模型细微差别的自定义指标。

### 7.3 评估基准和工具

本节描述了RAG模型的评估框架，包括基准测试和自动化评估工具。这些工具提供定量指标，不仅可以衡量RAG模型的性能，还可以增进对模型在各个评估方面的能力的理解。知名的基准测试如RGB和RECALL [Chen等，2023b, Liu等，2023b]专注于评估RAG模型的基本能力。同时，最先进的自动化工具如RAGAS [Es等，2023], ARES [Saad-Falcon等，2023]和TruLens<sup>8</sup>利用LLMs来裁决质量得分。

这些工具和基准共同构成了一个强大的框架，用于系统评估RAG模型，如表3所总结的。

<sup>8</sup>[https://www.trulens.org/trulens\\_eval/core\\_concepts\\_rag\\_triad/](https://www.trulens.org/trulens_eval/core_concepts_rag_triad/)

表2：适用于RAG评估方面的指标总结

	上下文 相关性	忠实度	答案 相关性	噪音 鲁棒性	负面 拒绝	信息 整合	反事实 鲁棒性
准确性	✓	✓	✓	✓	✓	✓	✓
EM					✓		
召回率	✓						
精确度	✓			✓			
R-率	✓						✓
余弦相似度			✓				
命中率	✓						
MRR	✓						
NDCG	✓						

表3：评估框架总结

评估框架	评估目标	评估方面	定量指标
RGB <sup>†</sup>	检索质量	噪音鲁棒性	准确性
	生成质量	负面拒绝	EM
		信息整合	准确性
		反事实鲁棒性	准确性
召回率 <sup>†</sup>	生成质量	反事实鲁棒性 R-率 (再出现率)	
RAGAS <sup>‡</sup>	检索质量	上下文相关性	*
	生成质量	忠实度	*
		答案相关性	余弦相似度
ARES <sup>‡</sup>	检索质量	上下文相关性	准确性
	生成质量	忠实度	准确性
		答案相关性	准确性
TruLens <sup>‡</sup>	检索质量	上下文相关性	*
	生成质量	忠实度	*
		答案相关性	*

<sup>†</sup>代表基准，<sup>‡</sup>代表工具。 \*表示定制的定量指标，与传统指标有所偏差。鼓励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。

## 8个未来展望

本节探讨了RAG的三个未来展望：未来挑战、模态扩展和RAG生态系统。

### 8.1 RAG的未来挑战

尽管RAG技术取得了相当大的进展，但仍存在一些需要深入研究的挑战：上下文长度。RAG的有效性受到大型语言模型（LLM）上下文窗口大小的限制。在窗口长度过短、信息不足的风险和窗口长度过长、信息稀释的风险之间取得平衡至关重要。随着不断努力将LLM上下文窗口扩展到几乎无限大小，RAG对这些变化的适应性成为一个重要的研究问题[Xu等，2023c，Packer等，2023，Xiao等，2023]。

鲁棒性。在检索过程中存在噪音或矛盾信息会对RAG的输出质量产生不利影响。

输出质量。这种情况被形象地称为“错误信息比没有信息更糟糕”。改进RAG对这种对抗性或反事实输入的抵抗力正在获得研究动力，并成为一个关键性能指标[Yu等，2023a，Glass等，2021，Baek等，2023]。

混合方法（RAG+FT）。将RAG与微调相结合正在成为一种主导策略。确定RAG和微调的最佳集成方式，无论是顺序、交替还是通过端到端联合训练，以及如何利用参数化和非参数化优势，都是值得探索的领域[Linet al., 2023]。

扩展LLM角色。除了生成最终答案外，LLM还被用于RAG框架内的检索和评估。在RAG系统中进一步发掘LLM的潜力是一个不断发展的研究方向。

缩放定律。尽管LLM的缩放定律[Kaplan等，2020]已经确立，但其在RAG中的适用性仍然不确定。

不确定。初步研究[Wang等, 2023b]已经开始解决这个问题,但是RAG模型的参数数量仍然落后于LLM。逆向缩放定律的可能性,即较小的模型表现优于较大的模型,特别引人注目,值得进一步研究。

生产就绪的RAG。RAG的实用性和与工程要求的一致性促进了其采用。

然而,提高检索效率,改进大型知识库中的文档召回率,以及确保数据安全性(例如防止LLM意外披露文档来源或元数据)是尚未解决的关键工程挑战[Alon等, 2022]。

### RAG的模态扩展

RAG已经超越了最初的基于文本的问答范围,拥抱了多样化的模态数据。

这种扩展催生了创新的多模态模型,将RAG的概念整合到各个领域中:图像。RA-CM3 [Yasunaga et al., 2022]

是一个先驱性的多模态模型,既可以检索文本又可以生成文本和图像。BLIP-2 [Liu et al., 2023a]利用冻结的图像编码器和LLMs进行高效的视觉语言预训练,实现了零-shot图像到文本的转换。

“在写作之前进行可视化”方法 [Zhu et al., 2022] 利用图像生成来引导语言模型的文本生成,在开放式文本生成任务中显示出潜力。

音频和视频。GSS方法通过检索和拼接音频片段将机器翻译数据转换为语音翻译数据 [Zhao et al., 2022]。UEOP通过整合外部的离线策略将语音转文本转换应用于端到端的自动语音识别,标志着重要的进展 [Chen et al., 2023]。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,从而加速领域适应。Vi d2Seq通过引入专门的时间标记增强语言模型,便于在统一的输出序列中预测事件边界和文本描述 [Yang et al., 2023a]。

代码。RBPS [Nashid等, 2023] 通过检索与开发者目标对齐的代码示例,通过编码和频率分析在小规模学习任务中表现出色。这种方法在测试断言生成和程序修复等任务中已经证明了其有效性。对于结构化知识,CoK方法 [Li等, 2023c] 首先从知识图中提取与输入查询相关的事,然后将这些事实作为提示集成到输入中,提高了知识图问答任务的性能。

## 8.2 RAG的生态系统

### 下游任务和评估

通过利用广泛的知识库,RAG在丰富语言模型处理复杂查询和生成详细响应的能力方面显示出了相当大的潜力。经验证据表明,RAG在各种下游任务中表现出色,包括开放式问题回答和事实验证。RAG的整合不仅增强了响应的准确性和相关性,还增加了其多样性和深度。

<sup>9</sup><https://github.com/inverse-scaling/prize>

RAG在多个领域的可扩展性和多功能性值得进一步研究,特别是在医学、法律和教育等专业领域。在这些领域中,与传统的微调方法相比,RAG有可能降低培训成本并提高专业领域知识问答的性能。

同时,改进RAG的评估框架对于最大化其在不同任务中的效力和效用至关重要。这需要开发细致入微的度量标准和评估工具,可以评估上下文相关性、内容创造力和无恶意等方面。

此外,提高基于RAG的模型的可解释性仍然是一个关键目标。这样做可以让用户理解模型生成的回答背后的推理过程,从而促进对RAG应用的信任和透明度。

### 技术栈

RAG生态系统的发展受到其技术栈的进展的极大影响。像LangChain和LLamaIndex这样的关键工具随着ChatGPT的出现迅速赢得了广泛的关注,提供了丰富的与RAG相关的API,并在LLMs领域中变得至关重要。新兴的技术栈虽然没有LangChain和LLamaIndex那么丰富的功能,

但它们通过提供专业化的服务来区别于其他技术栈。例如,Flowise AI<sup>10</sup>采用低代码方法,使用户能够通过用户友好的拖放界面部署包括RAG在内的AI应用。其他技术,如HayStack、Meltano<sup>11</sup>和Cohere Coral<sup>12</sup>,也因其在该领域的独特贡献而受到关注。

除了以人工智能为重点的供应商外,传统软件和云服务供应商也在扩大其提供的RAG中心服务范围。Weaviate的Verba<sup>13</sup>专为个人助理应用程序设计,而Amazon的Kendra<sup>14</sup>提供智能企业搜索服务,允许用户使用内置连接器浏览各种内容存储库。在RAG技术领域的发展过程中,出现了明显的分歧,包括:1)定制化。根据特定要求定制RAG。2)简化。使RAG更容易使用,从而减少初始学习曲线。3)专业化。优化RAG以更有效地服务于生产环境。

RAG模型及其技术堆栈的相互增长是显而易见的;技术进步不断为现有基础设施树立新的标准。反过来,技术堆栈的增强推动了RAG能力的发展。RAG工具包正在融合成为一个基础技术堆栈,为先进的企业应用奠定基础。然而,一个完全集成、综合的平台的概念仍然在视野中,需要进一步的创新和发展。

<sup>10</sup><https://flowiseai.com>

<sup>11</sup><https://meltano.com>

<sup>12</sup><https://cohene.com/coral>

<sup>13</sup><https://github.com/weaviate/Verba>

<sup>14</sup><https://aws.amazon.com/cn/kendra/>

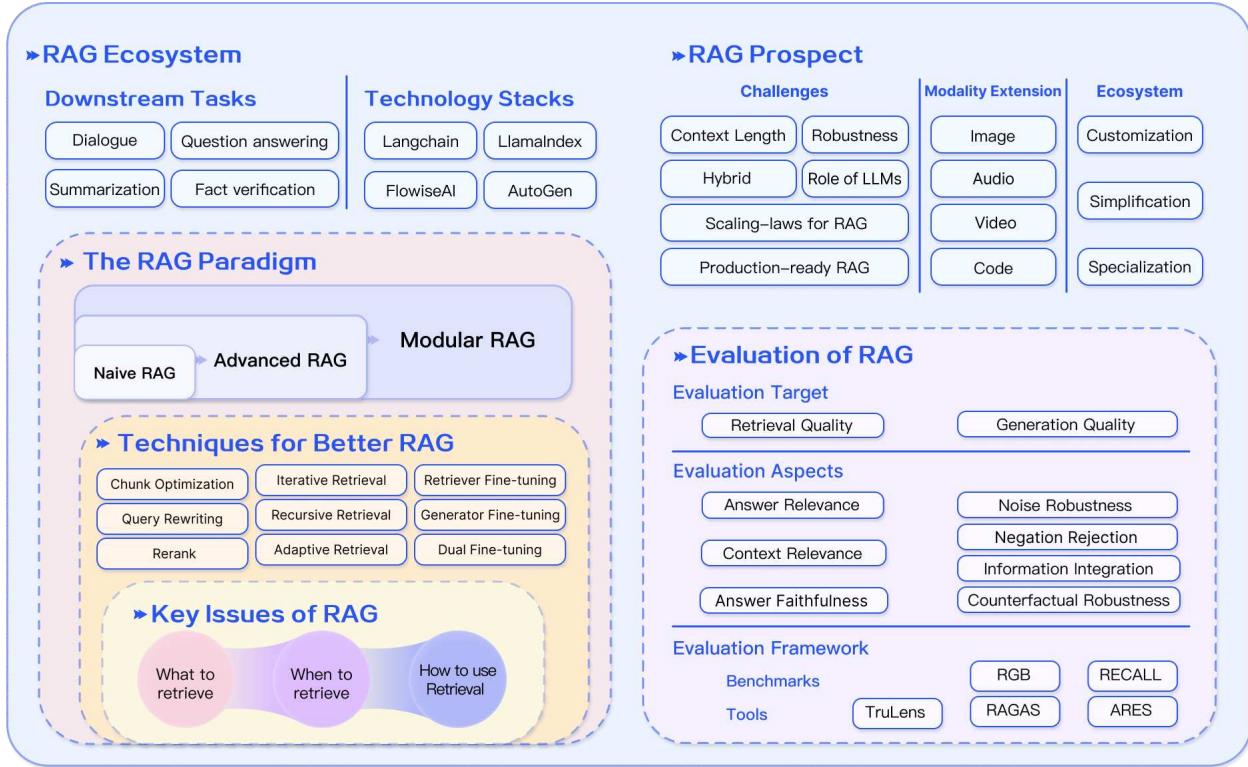


图7：RAG生态系统概述

## 9 结论

正如图7所示，本文的总结突出了RAG通过将来自语言模型的参数化知识与来自外部知识库的广泛非参数化数据集成，从而增强了LLM的能力。我们的调查说明了RAG技术的演变及其对知识密集型任务的影响。我们的分析描绘了RAG框架中的三种发展范式：Naive、Advanced和Modular RAG，每一种都对其前身进行了渐进性增强。Advanced RAG范式超越了Naive方法，包括了复杂的架构元素，包括查询重写、块重新排序和提示摘要。这些创新使得架构更加细致和模块化，增强了LLM的性能和可解释性。RAG与其他AI方法（如微调和强化学习）的技术整合进一步扩展了其能力。在内容检索方面，利用结构化和非结构化数据源的混合方法正在成为一种趋势，提供了更丰富的检索过程。RAG框架内的前沿研究正在探索从LLM进行自检索和动态信息检索的新概念。

◦

尽管在RAG技术方面取得了进展，但在提高其稳健性和管理扩展上下文的能力方面仍有研究机会。RAG的应用范围也正在扩大到多模态领域，适应其原则-

解释和处理各种数据形式（如图像、视频和代码）的挑战。这一扩展凸显了RAG在人工智能部署方面的重要实际意义，吸引了学术界和工业界的兴趣。RAG生态系统的不断发展体现在RAG中心的人工智能应用程序的增加和支持工具的持续开发。然而，随着RAG应用领域的扩大，有必要完善评估方法，以跟上其发展的步伐。确保性能评估保持准确和代表性对于捕捉RAG对人工智能研究和开发社区的全部贡献至关重要。

## 参考文献

[Alon等人, 2022] Uri Alon, Frank Xu, Junxian He, SudiptaSengupta, Dan Roth和Graham Neubig。带有自动机增强检索的神经符号语言建模。在国际机器学习会议上，页码为468-485。PMLR, 2022年。

[Anderson等, 2022] Nathan Anderson, Caleb Wilson, 和Stephen D. Richardson. Lingua：应对实时口译和自动配音的场景。在Jan- ice Campbell, Stephen Larocca, Jay Marciano, KonstantinSavenkov和Alex Yanishevsky编辑, *Proceedings ofthe 15th Biennial Conference of the Association for Ma-chine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)* , 页码202-209,

- 美国奥兰多，2022年9月。美洲机器翻译协会。
- [Arora等, 2023] Daman Arora, Anush Kini, Sayak Ray Chowdhury, Nagarajan Natarajan, Gaurav Sinha和Amit Sharma。Gar-meets-rag范式用于零-shot信息检索。arXiv预印本arXiv:2310.20158, 2023年。
- [Asai等人, 2023] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil和Hannaneh Hajishirzi。自我反思：学习通过自我反思来检索、生成和评论。arXiv预印本arXiv:2310.11511, 2023年。
- [BAAI, 2023] BAAI。Flagembedding。https://github.com/FlagOpen/FlagEmbedding, 2023年。
- [Baek等人, 2023] Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C Park和Sung Ju Hwang。知识增强语言模型验证。arXiv预印本arXiv:2310.12836, 2023年。
- [Berchansky等人, 2023] Moshe Berchansky, Peter Izsak, Avi Caciularu, Ido Dagan和Moshe Wasserblat。通过令牌消除优化检索增强阅读器模型。arXiv预印本arXiv:2310.13682, 2023年。
- [Blagojevi, 2023] 弗拉基米尔·布拉戈耶维。在干草堆中增强rag管道：引入diversityranker和lostinthemiddleranker。https://towardsdatascience.com/enhancing-rag-pipelines-in-haystack-45f14e2bc9f5, 2023。
- [Borgeaud et al., 2022] 塞巴斯蒂安·博尔戈, 亚瑟·门施, 乔丹·霍夫曼, 特雷弗·凯, 伊丽莎·拉瑟福德, 凯蒂米利坎, 乔治·范登·德里斯切, 让-巴蒂斯特·勒斯皮奥, 博格丹·达莫克, 艾丹·克拉克等。通过从数万个标记中检索来改进语言模型。在国际机器学习会议上, 第2206-2240页。PMLR, 2022年。
- [Brown et al., 2020] 汤姆·布朗, 本杰明·曼, 尼克·赖德, 梅兰妮·苏比亚, 贾里德·D·卡普兰, 普拉夫拉·达里-瓦尔, 阿尔温德·尼拉坎坦, 普拉纳夫·夏姆, 吉里什·萨斯特里, 阿曼达·阿斯克尔等。语言模型是少样本学习器。神经信息处理系统的进展, 33:1877-1901, 2020年。
- [Cai等人, 2021] Deng Cai, Yan Wang, Huayang Li, Wai Lam和Lemao Liu。神经机器翻译与单语翻译记忆。arXiv预印本arXiv:2105.11269, 2021年。
- [Chan等人, 2023] David M Chan, Shalini Ghosh, Ariya Rastrow和Björn Hoffmeister。在上下文端到端自动语音识别中使用外部非策略性语音到文本映射。arXiv预印本arXiv:2301.02736, 2023年。
- [Chen等人, 2023a] Howard Chen, Ramakanth Pasunuru, Jason Weston和Asli Celikyilmaz。通过交互式阅读走出记忆迷宫：超越上下文限制。arXiv预印本arXiv:2310.05029, 2023年。
- [Chen等, 2023b] 陈嘉伟, 林宏宇, 韩贤培和孙乐。在检索增强生成中对大型语言模型进行基准测试。arXiv预印本arXiv:2309.01431, 2023年。
- [Cheng等, 2022] 程鑫, 高申, 刘乐茂, 赵东岩和严锐。具有对比翻译记忆的神经机器翻译。arXiv预印本arXiv:2212.03140, 2022年。
- [Cheng等, 2023a] 程岱轩, 黄少涵, 毕俊宇, 詹跃峰, 刘建峰, 王玉静, 孙浩, 魏福如, 邓登威和张琦。Up-rise：用于改善零-shot评估的通用提示检索。arXiv预印本arXiv:2303.08518, 2023年。
- [Cheng等, 2023b] 郑欣, 罗迪, 陈秀英, 刘乐茂, 赵东岩, 严锐。提升自己：带有自我记忆的检索增强文本生成。arXiv预印本arXiv:2305.02437, 2023年。
- [Cohere, 2023] Cohere。告别无关的搜索结果：Cohere rerank来了。https://txt.cohere.com/rerank/, 2023年。
- [Dai等, 2022] 戴竹云, 赵宇文, 马吉, 栾逸, 倪建模, 卢静, 安东·巴卡洛夫, 凯尔文古, 基思·B·霍尔和明伟·张。Promptagator：从8个示例中进行少样本密集检索。arXiv预印本arXiv:2209.11755, 2022年。
- [Es等, 2023年] Shahul Es, Jithin James, Luis Espinosa-Anke和Steven Schockaert。Ragas：检索增强生成的自动评估。arXiv预印本arXiv:2309.15217, 2023年。
- [Feng等, 2023年] Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang和Bing Qin。检索-生成协同增强的大型语言模型。arXiv预印本arXiv:2310.05149, 2023年。
- [Gao等, 2022年] Luyu Gao, Xueguang Ma, Jimmy Lin和Jamie Callan。无相关性标签的精确零-shot密集检索。arXiv预印本arXiv:2212.10496, 2022年。
- [Glass等, 2021] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury和Alfio Gliozzo。强大的检索增强生成用于零-shot槽填充。arXiv预印本arXiv:2108.13934, 2021年。
- [Google, 2023] Google。Gemini：一系列高性能多模型。https://google/GeminiPaper, 2023年。
- [Guo等, 2023] 郭志成, 程思杰, 王一乐, 李鹏和刘阳。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。
- [Hendrycks等, 2020] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, 和Jacob Steinhardt。测量大规模多任务语言理解。arXiv预印本arXiv:2009.03300, 2020年。
- [Hoshi等人, 2023] Yasuto Hoshi, Daisuke Miyashita, Youyang Ng, Kento Tatsuno, Yasuhiro Morioka, O사무Torii和Jun Deguchi。Ralle：一个用于开发和评估检索增强大型语言模型的框架。arXiv预印本arXiv:2308.10633, 2023年。
- [Huang等人, 2023] Jie Huang, Wei Ping, Peng Xu, Mohammad Shoeybi, Kevin Chen-Chuan Chang和Bryan

- Catanzaro。Raven：上下文学习与检索增强编码器-解码器语言模型。arXiv预印本*arXiv:2308.07922*, 2023年。
- [ILIN, 2023] IVAN ILIN。  
图解概述。  
<https://pub.towardsai.net/advanced-rag-techniques-an-illustrated-overview-04d193d8fed> 2023年。
- [Izacard 等人, 2022] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, 和Edouard Grave。  
使用检索增强语言模型的少样本学习。  
*arXiv*预印本*arXiv:2208.03299*, 2022年。
- [Jarvis和Allard, 2023] Colin Jarvis和John Allard。  
最大化LLM性能的技术调查。  
<https://community.openai.com/t/openai-dev-day-2023-breakout-sessions/505213#>  
a-survey-of-techniques-for-maximizing-lm-performance-2, 2023年。
- [Jiang等人, 2023a] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang和Lili Qiu。  
LLMLingua：压缩大型语言模型加速推理的提示。*arXiv*预印本*arXiv:2310.05736*, 2023年。
- [Jiang等, 2023b] 郑宝江, Frank F Xu, 卢宇高, 孙志清, 刘倩, Jane Dwivedi-Yu, 杨一鸣, Jamie Callan和Graham Neubig。  
主动检索增强生成。*arXiv*预印本*arXiv:2305.06983*, 2023年。
- [Kandpal等, 2023] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace和Colin Raffel。  
大型语言模型难以学习长尾知识。  
在国际机器学习会议上, 页码为15696-15707。*PMLR*, 2023年。
- [Kang等, 2023] Minki Kang, Jin Myung Kwak, Jinheon Baek和Sung Ju Hwang。  
知识图增强的语言模型用于知识驱动的对话生成。*arXiv*预印本*arXiv:2305.18846*, 2023年。
- [Kaplan et al., 2020] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei。  
神经语言模型的缩放定律。*arXiv*预印本*arXiv:2001.08361*, 2020年。
- [Karpukhin et al., 2020] Vladimir Karpukhin, Barlas O'guz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih。  
用于开放领域问答的密集段落检索。*arXiv*预印本*arXiv:2004.04906*, 2020年。
- [Khandelwal et al., 2019] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis。  
通过记忆实现泛化：最近邻语言模型。*arXiv*预印本*arXiv:1911.00172*, 2019年。
- [Khattab等, 2022] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, 和Matei Zaharia。  
演示-搜索-预测：用于知识密集型自然语言处理的检索和语言模型。*arXiv*预印本*arXiv:2212.14024*, 2022年。
- [Kim 等 , 2023] Gangwoo Kim, Sungdong Kim, Byeong-guk Jeon, Joonsuk Park和Jaewoo Kang。  
澄清之树：用检索增强的大型语言模型回答模糊问题。*arXiv*预印本*arXiv:2310.14696*, 2023年。
- [Lan等, 2022] Tian Lan, Deng Cai, Yan Wang, Heyan Huang和Xian-Ling Mao。  
复制就是你所需的。在第十一届国际学习表示会议上, 2022年。
- [Lee等, 2020] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, 和Danqi Chen。  
学习大规模短语的密集表示。*arXiv*预印本*arXiv:2012.12624*, 2020年。
- [Leng 等 , 2023] Quinn Leng, Kasey Uhlenhuth, 和Alkis Polyzotis。  
LLM评估RAG应用的最佳实践。  
<https://www.databricks.com/blog/llm-auto-eval-best-practices-RAG>, 2023年。
- [Lewis等, 2020] Patrick Lewis, Ethan Perez, Aleksandr Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kautler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel等。  
用于知识密集型NLP任务的检索增强生成。神经信息处理系统的进展, 33:9459–9474, 2020年。
- [Li and Li, 2023] 李贤明和李静。  
角度优化的文本嵌入。*arXiv*预印本*arXiv:2309.12871*, 2023年。
- [Li等, 2023a] 李俊南, 李东旭, Silvio Savarese和Steven Hoi。  
Blip-2：使用冻结图像编码器和大型语言模型进行语言-图像预训练。*arXiv*预印本*arXiv:2301.12597*, 2023年。
- [Li等, 2023b] 李小倩, 聂尔聪和梁胜。  
从分类到生成：跨语言检索增强iclr的见解。  
*arXiv*预印本*arXiv:2311.06595*, 2023年。
- [Li等, 2023c] 李兴轩, 赵若辰, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty和Sou-janya Poria。  
知识链：用结构化知识库为大型语言模型提供基础的框架。*arXiv*预印本*arXiv:2305.13269*, 2023年。
- [Li等, 2023d] Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu和Ge Yu。  
结构感知语言模型预训练改善了结构化数据上的密集检索。*arXiv*预印本*arXiv:2305.19912*, 2023年。
- [Liang等, 2023] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu和Lan Xu。  
Intergen：基于扩散的复杂互动下的多人运动生成。*arXiv*预印本*arXiv:2304.05684*, 2023年。
- [Lin等, 2023] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvassy, Mike Lewis等。  
Ra-dit：检索增强的双指令调整。*arXiv*预印本*arXiv:2310.01352*, 2023年。
- [Litman等, 2020] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor和R Manmatha。  
Scatter：选择性上下文注意力场景文本识别器。在IEEE/CVF计算机视觉和模式识别会议论文集中, 页码为11962-11972, 2020年。

- [Liu等, 2023a] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni 和Percy 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。
- [Liu等, 2023b] Yi Liu, Lianzhe Huang, Shicheng Li, Si shuo Chen, Hao Zhou, Fandong Meng, Jie Zhou和Xu Sun。Recall：LLM对外部反事实知识的鲁棒性基准。arXiv预印本arXiv:2311.08147, 2023年。
- [Liu, 2023] 杰瑞·刘。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。
- [Luo et al., 2023] 紫阳·罗, 灿·徐, 普·赵, 秀波耿, 崇阳·陶, 静·马, 庆伟·林和大新江。带参数知识引导的增强型大型语言模型。arXiv预印本 arXiv:2305.04757, 2023。
- [Ma et al., 2023a] 新北·马, 叶云·龚, 鹏程何, 海·赵和南·段。用于检索增强型大型语言模型的查询重写。arXiv预印本arXiv:2305.14283, 2023。
- [Ma et al., 2023b] 宇波·马, 一鑫·曹, 永清·洪, 和爱新·孙。大型语言模型不是一个好的少样本信息提取器, 但对于困难样本是一个好的重新排序器! ArXiv, abs/2303.08559, 2023。
- [Modarressi等, 2023] Ali Modarressi, Ayyoob Imani, M ohsen Fayyaz和Hinrich Schütze。Ret-llm：面向大型语言模型的通用读写内存。arXiv预印本arXiv:2305.14322, 2023年。
- [Nakano等, 2021] Reiichiro Nakano, Jacob Hilton, Suchi r Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders等。Webgpt：带有人类反馈的浏览器辅助问答。arXiv预印本arXiv:2112.09332, 2021年。
- [Nashid等, 2023] Noor Nashid, Mifta Sintaha和Ali Mesbah。基于检索的代码相关少样本学习的提示选择。在2023年IEEE/ACM第45届国际软件工程大会 (ICSE) 上, 第2450-2462页。
- [Nguyen, 2023] Isabelle 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。
- [Nishikawa等, 2022] 西川宗介, 李旅馆, 山田育也, 鹤冈佳正和越前功。Ease：基于实体感知的对比学习句子嵌入。arXiv预印本arXiv:2205.04260, 2022年。
- [OpenAI, 2023] OpenAI。Gpt-4技术报告。<https://cdn.openai.com/papers/gpt-4.pdf>, 2023年。
- [Packer等, 2023年] Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders和Joseph E Gonzalez。Memgpt：朝着将LLMS作为操作系统的方向。arXiv预印本arXiv:2310.08560, 2023年。
- [Raffel等, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena , Yanqi Zhou, Wei Li和Peter J Liu。通过统一的文本到文本转换器探索迁移学习的极限。机器学习研究杂志, 21(1): 5485-5551, 2020年。
- [Ram等, 2023] Ori Ram, Yoav Levine, Itay Dalmedigos , Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown和Yoav Shoham。上下文检索增强语言模型。arXiv 预印本arXiv:2302.00083, 2023年。
- [Raudaschl, 2023] Adrian H. Raudaschl。忘记rag, 未来是rag-fusion。<https://towardsdatascience.com/forget-rag,-future-is-rag-融合-1147298d8ad1>, 2023年。
- [Saad-Falcon等, 2023年] Jon Saad-Falcon, Omar Khatab, Christopher Potts和Matei Zaharia。Ares：用于检索增强生成系统的自动化评估框架。arXiv预印本arXiv:2311.09476, 2023年。
- [Schick等, 2023年] Timo Schick, Jane Dwivedi-Yu, Roberto Dess`i, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda和Thomas Scialom。To olformer：语言模型可以自学使用工具。arXiv预印本arXiv:2302.04761, 2023年。
- [Sciavolino等, 2021] Christopher Sciavolino, Zexuan Zhang, Jinhyuk Lee和Danqi Chen。简单的以实体为中心的问题挑战密集检索器。arXiv 预印本 arXiv:2109.08535, 2021年。
- [Shao等, 2023] 邵志宏, 龚业云, 沈晔龙, 黄敏烈, 段楠和陈伟柱。增强检索增强的大型语言模型与迭代检索生成协同作用。arXiv预印本arXiv:2305.15294, 2023年。
- [Shi等, 2023] 施伟佳, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer和Wen-tau Yih。Replug：检索增强的黑盒语言模型。arXiv预印本arXiv:2301.12652, 2023年。
- [Srivastava等, 2022年] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso等。超越模仿游戏：量化和推断语言模型的能力。arXiv预印本arXiv:2206.04615, 2022年。
- [Sun等, 2022年] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yim-ning Yang和Denny Zhou。复述增强的语言模型。arXiv预印本arXiv:2210.01296, 2022年。
- [Touvron等, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaee, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shuri Bhosale等。Llama 2：开放基础和精细调整的聊天模型。arXiv预印本arXiv:2307.09288, 2023年。
- [Trivedi等, 2022] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot和Ashish Sabharwal。将检索与思维链式推理相互交织, 用于知识密集型多步问题。arXiv预印本arXiv:2212.10509, 2022年。

- [Vaswani等, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser和Illia Polosukhin。注意力就是你所需要的。神经信息处理系统的进展, 30, 2017年。
- [VoyageAI, 2023] VoyageAI。Voyage的嵌入模型。  
<https://docs.voyageai.com/embeddings/>, 2023年。
- [Wang等, 2019年] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy和Samuel Bowman。Superglue: 通用语言理解系统的基准测试。神经信息处理进展, 32, 2019年。
- [Wang等, 2022a年] Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguan Zhu和Michael Zeng。训练数据比你想象的更有价值: 一种通过从训练数据中检索的简单有效方法。arXiv预印本arXiv:2203.08773, 2022年。
- [Wang等, 2022b] 王硕航, 徐一冲, 方宇伟, 刘洋, 孙思琪, 徐若尘, 朱晨光, 曾迈克尔。训练数据比你想象的更有价值: 一种简单有效的从训练数据中检索的方法。在Smaranda Muresan, Preslav Nakov和Aline Villavicencio编辑的《计算语言学协会第60届年会论文集(卷1: 长文)》中, 第3170-3179页, 2022年5月, 爱尔兰都柏林。计算语言学协会。
- [Wang等, 2023a] 王博鑫, 平伟, 劳伦斯·麦克菲, 徐鹏, 李波, 穆罕默德·肖伊比和布莱恩·卡坦扎罗。Instructretro: 检索增强预训练的指令调优。arXiv预印本arXiv:2310.07713, 2023年。
- [Wang等, 2023b] Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 我们是否应该使用检索来预训练自回归语言模型? 一项全面的研究。arXiv预印本arXiv:2304.06762, 2023年。
- [Wang等, 2023c] Liang Wang, Nan Yang和Furu Wei。Query2doc: 利用大型语言模型进行查询扩展。arXiv预印本arXiv:2303.07678, 2023年。
- [Wang等, 2023d] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanhua Xiao和Wei Wang。Knowledgpt: 在知识库上增强大型语言模型的检索和存储访问。arXiv预印本arXiv:2308.11761, 2023年。
- [Wang等, 2023e] 王一乐, 李鹏, 孙茂松, 刘洋。自知引导的大型语言模型检索增强。arXiv预印本arXiv:2310.05002, 2023年。
- [Xia等, 2019] 夏梦洲, 黄国平, 刘乐茂, 石树明。基于图的神经机器翻译记忆。在人工智能AAAI会议论文集中, 第33卷, 7297-7304页, 2019年。
- [Xiao等, 2023] 肖光轩, 田远东, 陈北迪, 韩松, Mike Lewis。具有注意力汇聚的高效流式语言模型。arXiv预印本arXiv:2309.17453, 2023年。
- [Xu等, 2023a] Fangyuan Xu, Weijia Shi和Eunsol Choi。Recomp: 通过压缩和选择性增强改进检索增强的语言模型。arXiv预印本arXiv:2310.04408, 2023年。
- [Xu等, 2023b] Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi和Bryan Catanzaro。检索遇见长上下文大型语言模型。arXiv预印本arXiv:2310.03025, 2023年。
- [Xu等, 2023c] Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi和Bryan Catanzaro。检索遇见长上下文大型语言模型。arXiv预印本arXiv:2310.03025, 2023年。
- [Yang等, 2023a] 安托万·杨, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, 和 Cordelia Schmid。Vid2seq: 用于密集视频字幕的大规模预训练视觉语言模型。在IEEE/CVF计算机视觉和模式识别会议的论文中, 页码为10714-10726, 2023年。
- [Yang等, 2023b] 郝燕, 李志涛, 张勇, 王建宗, 程宁, 李明, 和肖静。Prca: 通过可插拔的奖励驱动上下文适配器为检索问题回答拟合黑盒大型语言模型。arXiv预印本arXiv:2310.18347, 2023年。
- [Yang等, 2023c] 惠杨, 岳思夫, 和何云中。Auto-gpt用于在线决策: 基准和附加意见。arXiv预印本arXiv:2306.02224, 2023年。
- [Yasunaga等, 2022] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer和Wen-tau Yih。检索增强的多模态语言建模。arXiv预印本arXiv:2211.12561, 2022年。
- [Ye等, 2020] 叶德明, 林彦凯, 杜佳菊, 刘正豪, 李鹏, 孙茂松和刘知远。用于语言表示的共指推理学习。arXiv预印本arXiv:2004.06870, 2020年。
- [Yoran等, 2023] Ori Yoran, Tomer Wolfson, Ori Ram和Jonathan Berant。使检索增强的语言模型对无关上下文具有鲁棒性。arXiv预印本arXiv:2310.01558, 2023年。
- [Yu等, 2022] 于文浩, 丹·伊特尔, 王硕航, 徐一冲, 鞠明轩, Soumya Sanyal, 朱晨光, 曾迈克尔和蒋萌。生成而不是检索: 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。
- [Yu等, 2023a] 于文浩, 张洪明, 潘晓满, 马凯欣, 王宏伟和于东。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。

[Yu等, 2023b] 于子淳, 熊晨燕, 于石和刘志远。增强  
适应的检索器改善语言模型的泛化能力作为通用插件  
。  
*arXiv*预印本 *arXiv:2305.17331*, 2023年。

[Zhang等, 2019年] 郑燕张, 韩旭, 刘志远, 江欣, 孙  
茂松和刘群。Ernie: 增强语言表示与信息实体。*arXi*  
*v*预印本 *arXiv:1905.07129*, 2019年。

[Zhang等, 2023a] 张培天, 肖世涛, 刘铮, 窦志成和聂  
建云。检索任何内容以增强大型语言模型。*arXiv*预  
印本 *arXiv:2310.07554*, 2023年。

[Zhang等, 2023b] 张越, 李亚夫, 崔乐阳, 蔡登, 刘乐  
茂, 傅廷琛, 黄新婷, 赵恩波, 张宇, 陈宇龙等。人  
工智能海洋中的塞壬之歌: 大型语言模型中的幻觉综  
述。*arXiv*预印本 *arXiv:2309.01219*, 2023年。

[Zhang, 2023] 张佳伟。图工具变形器: 通过聊天GPT增  
强提示来赋予LLM图推理能力。*arXiv*预印本 *arXiv:23*  
*04.11116*, 2023年。

[Zhao等, 2022] 赵金明, Gholamreza Haffar和Ehsan Sha  
reghi。从口语词汇生成合成语音用于语  
音翻译。*arXiv*预印本 *a*  
*rXiv:2210.08174*, 2022年。

[Zheng等, 2023] 郑怀秀Steven Zheng, Swaroop Mishra  
, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V  
Le和Denny Zhou。退一步: 通过抽象在大型语言模  
型中引发推理。*arXiv*预印本 *arXiv:2310.06117*, 2023  
年。

[Zhu 等 , 2022] Wanrong Zhu, An Yan, Yujie Lu, Wenda  
Xu, Xin Eric Wang, Miguel Eckstein和William Yang  
Wang。在写作之前进行可视化: 想象引导的开放  
式文本生成。*arXiv*预印本  
*arXiv:2210.03765*, 2022年。

[Zhuang 等 , 2023] Shengyao Zhuang, Bing Liu, Bevan  
Koopman和Guido Zuccon。开源大型语言模  
型是强零-shot查询可能性模型, 用于文档排名。*arXi*  
*v*预印本 *arXiv:2310.13243*, 2023年。