电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

学士学位论文

BACHELOR THESIS

论文题目 检索增强大语言模型研究综述

学科专业

学　　号

作者姓名 蒋钦禹 李世杰 耿玮俊

指导老师 方瑞琴

学　　院 计算机科学与工程学院

摘　要

近年来，大规模预训练语言模型（LLMs）在自然语言处理（NLP）领域取得了显著进展，但它们在处理知识密集型任务时仍面临知识更新和准确性的挑战。为此，检索增强型语言模型（RAG）技术应运而生，它通过结合语言模型与信息检索模块，动态获取外部信息，以生成更准确、时效性更高的内容。本文综述了RAG 技术的核心范式、组成要素、关键技术，并探讨了其在输入提升、检索器优化、语言模型改进以及整个 RAG 流程提升方面的研究进展。此外，本文还展望了RAG 技术的未来研究方向，包括多语言和多模态能力的提升、外部知识的质量控制、计算效率的优化，以及面临的安全问题。RAG 技术的发展不仅推动了语言生成技术的新一轮发展，也为问答系统、对话生成等信息密集型任务提供了新的解决方案，展现出广阔的应用前景。

关键词：检索增强生成，大语言模型

目 录

第一章 引 言

1.1 背景介绍

近年来，大规模预训练语言模型（Large Language Models, LLMs）在自然语言处理（NLP）领域取得了显著突破 [1]。以 BERT 和 GPT 系列为代表的模型，通过海量数据的预训练，展现了强大的语言生成和理解能力。这些 LLMs 被广泛应用于文本生成、机器翻译和对话系统等多个任务，推动了智能应用的普及 [2] [3]。然而，尽管 LLMs 具备强大的语言处理能力，仍存在明显的局限性。首先，LLMs 的知识来源于其训练数据，无法动态更新。这导致它们在处理实时信息或领域专有知识时，可能生成不准确或过时的答案。此外，模型有时会出现“幻觉”（hallucination），即生成与事实不符的内容。这些局限性表明，传统 LLMs 在处理知识密集型任务时存在瓶颈，无法满足某些高精度任务的要求 [4]。为了解决这些问题，研究者提出了检索增强生成（Retrieval-Augmented Generation, RAG）技术。这种技术通过将语言模型与信息检索模块结合，使模型不仅依赖于预训练的内部知识，还能够动态检索外部信息，从而生成更准确、时效性更高的内容。RAG 架构不仅弥补了 LLMs 在知识覆盖和实时性上的不足，还显著减少了幻觉现象，使得生成结果更加可靠 [5] [6]。根据《RAG+RAU：对检索增强型语言模型（RALM）进行全面、深入综述》的分析，RAG 技术使得语言模型能够在生成过程中从外部知识库中检索相关信息，从而改善其生成能力 [7]。这种创新在问答系统、对话生成等信息密集型任务中取得了显著成效，推动了语言生成技术的新一轮发展。

1.2 RALMs 的研究意义与应用前景

检索增强大语言模型（Retrieval-Augmented Large Language Models, RALMs）是 RAG 技术的进一步扩展，通过实时检索外部信息提升语言模型的知识覆盖面和准确性。与传统 LLMs 相比，RALMs 不再仅依赖其内置的语言模式，而是通过检索模块获取最新、最相关的外部知识来生成更为准确的内容。这一创新在多个应用领域展现出广泛的前景，尤其是在问答系统、知识推理、医疗诊断等知识密集型任务中。RALMs 的优势在于，它们可以通过检索模块实时获取信息，而不是仅仅依赖训练期间学习到的知识。这种能力使它们在复杂、动态的任务环境中表现得尤为出色。此外，检索增强技术还有效降低了模型对庞大参数规模的依赖，从而减少了模型的计算成本。通过引入检索机制，RALMs 可以更高效地处理长文本和复杂推理任务，这在领域专有知识和实时数据检索场景中尤为重要。

第二章 RAG 技术详解

2.1 RAG 的核心范式

检索增强型语言模型（RAG）是一种结合了检索（Retrieval）和生成（Generation）的先进自然语言处理技术。RAG 的核心在于利用外部信息源来增强大型语言模型（LLMs）的性能，以解决传统模型在处理特定领域知识时遇到的挑战，如幻觉问题和领域知识的缺失。RAG 技术可以分为两个主要范式：检索增强生成（RAG）和检索增强理解（RAU）。

2.1.1 检索增强生成（RAG）

检索增强生成（RAG）主要关注于利用检索到的信息来辅助生成任务，如文本生成、机器翻译和对话系统。在这一范式中，模型首先通过检索器从大量数据中检索出与输入相关的信息，然后将这些信息作为上下文输入到语言模型中，以生成流畅、准确且信息丰富的文本。

2.1.2 检索增强理解（RAU）

检索增强理解（RAU）则侧重于利用检索信息来提升模型对文本的理解能力，这在问答系统 [8]、文本分类和事实核查等任务中尤为重要。RAU 通过检索相关信息来增强模型对输入文本的语义理解，从而提高任务的准确性。

2.2 RAG 的组成要素

RAG 系统的架构主要由两个部分组成：检索器、语言模型。

2.2.1 检索器

检索器负责从大量的数据源中检索出与输入查询最相关的信息。检索器的性能直接影响到 RAG 系统的效果，因此，设计高效准确的检索器是 RAG 技术的关键。

2.2.2 语言模型

语言模型是 RAG 系统的另一个核心组成部分，它负责生成或理解文本。根据任务的不同，可以选择不同类型的语言模型，如自编码器模型、自回归模型或编码器-解码器模型。

2.3 RAG 的关键技术

2.3.1 检索器类型

检索器在 RALM 架构中扮演着至关重要的角色。通过检索器获取的信息可以显著提高大语言模型的准确性。

2.3.1.1 稀疏检索

TF-IDF 算法是一种统计方法，用以评估一个词语对于一个文件集或一个语料库中的其中一份文件的重要性。

词频（TF）[9] 表示词在文档中出现的次数，计算公式通常是：

$$\mathrm{TF}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2-1}$$

其中：$TF(t,d)$ 是词 t 在文档 d 中的词频。$f_{t,d}$ 是词 t 在文档 d 中出现的频率。$\sum_{t' \in d} f_{t',d}$ 是文档 d 中所有词的频率之和。

逆文档频率（IDF）表示词在整个语料库中出现的频率的倒数，计算公式通常是：

$$\mathrm{IDF}(t,D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{2-2}$$

其中：$IDF(t,D)$ 是词 t 的逆文档频率。$|D|$ 是语料库中文档的总数。$|\{d \in D : t \in d\}|$ 是包含词 t 的文档数量。

TF-IDF 是 TF 和 IDF 的乘积，用于衡量词在文档中的重要性，计算公式通常是：

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D) \tag{2-3}$$

BM25 算法 [10] 是基于 TF 和 IDF 的基础上进行改进的一种检索算法，它考虑了词频、文档长度和文档集合中词的分布，计算公式通常是：

$$\mathrm{BM25}(q,d) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \times \frac{f(q_i,d) \times (k_1+1)}{f(q_i,d) + k_1 \times (1 - b + b \times \frac{|d|}{\mathrm{avgdl}})} \tag{2-4}$$

其中：$\mathrm{BM25}(q,d)$ 是查询 q 和文档 d 之间的 BM25 得分。$\mathrm{IDF}(q_i)$ 是查询词 q_i

的逆文档频率。$f(q_i, d)$ 是查询词 q_i 在文档 d 中的词频。k_1 和 b 是 BM25 算法的调节参数。$|d|$ 是文档 d 的长度。avgdl 是文档集合的平均长度。

对于每个查询词，BM25 算法会计算它在每个文档中的得分，然后将这些得分相加，得到该文档对于整个查询的总得分。在 BM25 的上下文中，所说的“稀疏向量”通常指的是文档的向量表示，其中只包含非零项，即那些查询词在文档中出现的词频和逆文档频率的乘积。由于大多数词在大多数文档中不会出现，所以这种表示通常是稀疏的。最后，通过比较查询向量和文档向量之间的相似性（例如，使用余弦相似度），可以找到与查询最相关的文档。

稀疏检索最初依赖于匹配相关内容的方法，如 TF-IDF（Term Frequency-Inverse Document Frequency）和 BM25 算法。这些算法通过计算词频和逆文档频率来评估相关性，具有简单和快速的优点。随着机器学习技术的发展，稀疏向量被用来表示词，并通过网络距离计算来检索它们。稀疏检索在 RALMs 中可以用于多种任务，包括自动翻译、文本分类、情感分析等。它特别适用于那些基于知识的任务，因为这些任务通常需要从大量文档中检索信息。

稀疏检索通常依赖于倒排索引和原始数据输入。这种方法简单、不依赖于训练，但性能受限于数据库质量和查询。

2.3.1.2 密集检索

密集检索使用深度学习技术来生成查询和文档的稠密向量表示，然后通过计算向量之间的距离来检索信息。这种方法能够更好地捕捉到查询和文档之间的语义关系，但计算成本较高。

在密集检索中，常用的架构是双编码器（Dual-Encoder）模型，它包含两个独立的网络，分别对查询和文档进行编码，然后通过计算编码向量之间的相似度来检索相关信息。

在密集检索中，词嵌入是一种常见的方法，它使用深度学习技术将词映射到高维向量空间。这些嵌入能够捕捉词之间的语义关系，从而提高检索的准确性。例如，DPR（Dense Passage Retriever）模型 [11] 就是一种使用密集嵌入的检索模型，它通过在低维连续空间中索引所有段落，使得在运行时高效地检索与输入问题相关的前 k 个段落成为可能

2.3.1.3 互联网检索

互联网检索是指直接从互联网上检索信息的方法。这种方法可以获取到最新的信息，但面临的挑战包括信息的准确性和相关性。互联网检索可以作为稀疏检索和密集检索的补充，以提高检索的效果。搜索引擎本身利用了大量的数据和传

统检索方法，可以作为 RALMs 的一个重要组成部分，增强模型的时效性和泛化能力。[12]

2.3.1.4 混合检索

混合检索结合了多种检索技术，以提高检索的准确性和鲁棒性。这种方法可以结合稀疏检索的效率和密集检索的语义理解能力，或者结合互联网检索的最新信息。[13] [14]

2.3.2 语言模型的选择

与仅依靠训练参数完成任务的传统语言模型不同，RAG 中的语言模型通过整合检索器获取的非参数记忆和自身的参数记忆，形成半参数记忆，从而增强了语言模型的性能。

2.3.2.1 自编码器语言模型

自编码器语言模型，如 BERT，通过预测遮蔽词来学习语言的表示。BERT 采用了一种遮蔽语言模型（Masked Language Model，MLM）的训练方式，即在输入文本中，随机选择一些单词（通常是 15% 左右）并将其替换为特殊的 [MASK] 标记。模型的任务是预测这些被遮蔽的单词。这种方式迫使模型学习单词之间的双向关系，因为它需要考虑整个句子的上下文来预测被遮蔽的单词，而不仅仅是单向的前文或后文信息。这类模型在理解任务中表现出色，因为它们能够捕捉到丰富的上下文信息。

自编码器语言模型常用于自然语言理解（NLU）任务，如在一些 RALM 架构中用于判断等特定任务。具有高度的泛化能力，是无监督学习的，不需要数据标注，能够自然地融入上下文语义信息。

2.3.2.2 自回归语言模型

自回归语言模型，如 GPT 系列，通过预测下一个词来生成文本。GPT 采用了无监督预训练的方式，在大规模的文本语料上进行学习。它以预测下一个单词为目标，给定一段文本中的前几个单词，模型尝试预测下一个单词是什么。

自回归语言模型适用于自然语言生成（NLG）任务，如对话生成和机器翻译等，是 RAG 中处理 NLG 任务的流行选择。采用从左到右的语言建模方式，能够根据前面的单词预测下一个单词，适合生成式自然语言处理任务。

2.3.2.3 编码器-解码器模型

编码器-解码器模型，如 T5 [15]，结合了编码器和解码器的结构，使其在处理需要转换输入到输出的任务时非常有效，如机器翻译。

T5 引入了一个统一的框架，将所有基于文本的语言问题转化为文本到文本的格式。这使得它在 RAG 中能够更方便地处理各种自然语言处理任务，无论是文本生成、问答还是其他任务。例如，在检索增强的问答任务中，它可以将问题和检索到的相关文档作为输入，以生成准确的答案。

2.3.3 检索粒度

检索粒度指的是检索单元的索引级别，如文档、段落、令牌或其他级别（如实体）[16] [17]。文档级检索：将整个文档作为检索单元。段落级检索（Chunk Retrieval）：在传统和基于 LLM 的 RAG 模型中较为常见，如 REALM、RAG 和 Atlas。令牌级检索（Token Retrieval）：更细粒度的检索，适用于需要罕见模式或领域外数据的情况。实体检索（Entity Retrieval）：从知识而非语言的角度设计，如 Entities as Experts (EAE) 模型，它通过实体记忆来表示知识。

2.3.4 生成器设计

参数可访问生成器 [18]（White-box Generators）：如 Encoder-Decoder 和 Decoder-only 模型，允许参数优化，可以针对不同的检索和增强方法进行训练以提高生成性能。参数不可访问生成器（Black-box Generators）：如 GPT 系列，不允许内部结构更改或参数更新，专注于通过检索和增强过程来提升生成器的性能。

2.3.5 检索集成方法

检索集成方法（Retrieval-Integration Methods）是指在人工智能系统中，尤其是语言模型和信息检索系统中，将检索到的信息与模型生成的结果结合起来，以提高系统的性能和输出的相关性。这种方法特别适用于需要结合大量外部数据来提供准确和及时回答的场景。输入层集成（Input-Layer Integration）：将检索到的文档与原始查询结合，作为生成器的新输入。输出层集成（Output-Layer Integration）：在输出阶段将检索结果与生成结果结合起来，如 kNN-LM 通过插值两个下一个词的概率分布来进行预测。中间层集成（Intermediate-Layer Integration）：通过生成模型的内部层来整合检索结果，这种方法可能增加额外的复杂性，但也有望通过有效训练来提升生成模型的能力 [19]。上下文集成：在这种方法中，检索到的数据被用作上下文信息，提供给模型以帮助其更好地理解查询的上下文，并生成更准

确的回答。

小型模型集成：通过训练一个小型的语言模型来专门处理检索到的数据，然后将这个小型模型的输出作为输入提供给大型语言模型，以此来指导信息的整合。

微调集成：使用外部领域特定的数据对预训练的大型语言模型进行微调，使其能够更好地处理特定领域的查询。[20]

第三章 研究进展

3.1 输入提升

输入指的是用户的查询，该查询最初被输入到检索器中。输入的质量显著影响检索阶段的最终结果，因此对输入优化变得至关重要。在这里，我们将介绍两种方法：查询改写和数据增强。查询改写可以通过修改输入查询来提高检索结果。数据增强是指在检索之前提前对数据进行改进，如去除无关信息、消除歧义、更新过时文档、合成新数据等，可以有效提高最终 RAG 系统的性能 [21]。

3.2 检索器优化

在 RAG 系统中，检索过程对结果影响很大。一般来说，内容质量越好，就越容易激发 LLM 的上下文学习能力以及其他生成模型的能力；内容质量越差，就越有可能导致模型幻觉。常见的优化方法有以下几种：

递归检索是一种高级的检索策略，它通过在检索之前拆分查询，并执行多次搜索以检索更多、更高质量的内容。这种策略的核心思想是在不同层次上构建 chunks 节点与检索器，并建立层次之间的链接关系，使得能够在每次检索时自动实现向下递归探索，直至达到结束条件。

检索器微调是对检索器的优化，一般是对嵌入模型能力的提升。检索器的能力越强，就可以为后续生成器提供更多有用的信息，从而提高 RAG 系统的有效性。一个好的嵌入模型可以使语义相似的内容在向量空间中更紧密地结合在一起；此外，对于已经具有良好表达能力的嵌入模型，我们仍然可以使用高质量的领域数据或任务相关数据对其进行微调，以提高其在特定领域或任务中的性能。

相比单检索方式，混合检索更具全面性，混合检索是指同时使用多种类型的检索器，如同时使用统计词频的方式和计算向量相似性的方式来得到检索结果，也就是混合了稀疏检索和密集检索。稀疏检索侧重于关键词的精确匹配，适用于搜索特定术语，如产品名称或专业术语。而密集检索则侧重于理解查询和文档的上下文和含义，适用于捕捉语义相似性，即使查询中不存在确切的关键字也能检索到相关信息。因此，混合检索对不同类型的查询更具鲁棒性，无论它们是精确的基于关键字的查询，还是更抽象且依赖于上下文的查询。例如，Hybrid with HyDE [22] 方法将稀疏和稠密检索结合起来，从语义和语义角度捕捉相关文档。

此外，加入重排序技术，对检索到的内容进行重新排序，可以实现更大的多样性和更好的结果。

3.3 语言模型的改进

在 RAG 系统中，生成器的质量通常决定最终输出结果的质量。在这里，我们将介绍如下一些提升生成器能力的技术。提示词工程是一种专注于提高 LLM 输出质量的技术，其中包括提示词压缩、回退提示、主动提示、思维链提示等等，以上这些同时也都适用于使用 LLM 生成器的 RAG 系统中。解码过程控制、调整是指在生成器处理过程中添加额外的控制，可以通过调整超参数来实现更大的多样性或者以某种形式限制输出词汇表等等。生成器微调可以使生成模型具有更精确的领域知识或更好地与检索器匹配的能力。

3.4 RAG 流程提升

我们将对整个 RAG 流程上的优化分为如下两大类：自适应检索和迭代 RAG。

自适应检索是基于一个观察：很多 RAG 的研究和实践表明，检索并不总是有利于最终生成的结果。当模型本身的参数化知识足以回答相关问题时，过度检索会造成资源浪费，并可能增加模型的混乱。因此，一些工作提出了基于规则和基于模型的自适应检索方法。基于规则等方法指的是通过判断某些与模型生成高度相关的指标来确定是否进行搜索，具体而言，这个变量可以是模型生成过程中当前 token 的生成概率，也可以是模型的困惑度等等。基于模型的方法则指的是借助模型能力来判断是否进行搜索，这里的模型可以是生成模型本身也可以是借助外部模型。

迭代 RAG 则指的是迭代的进行检索和生成。生成器的当前轮次输出可以在一定程度上反映其仍然缺乏的知识，并且检索器可以检索缺失的信息作为下一轮的上下文信息，这有助于提高下一轮生成内容的质量。如此循环迭代，直到生成内容达到标准。

第四章 未来研究方向

4.1 多语言和多模态能力的提升

RAG 已经超越了最初基于文本的问题回答的限制，融入了各种各样的模态数据。这种扩张催生了创新的多模式模式集成了不同领域的 RAG 概念的形象 [23]。RA-CM3 [24] 是多模态的先驱检索和生成文本和图像的模型。BLIP-2 [25] 利用了冻结图像编码器用于高效视觉语言预训练的 LLM，实现零样本图像到文本转换。“在你面前形象化方法 [26] 使用图像生成来控制 LM 的文本生成，在开放式文本中显示出希望一代的任务。音频和视频。GSS 方法用于检索和缝合一起音频剪辑转换成机器翻译的数据语音翻译数据 [27]。UEOP 标志着端到端自动语音识别的重大进步结合外部离线策略进行语音到文本转换 [28]。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完善 ASR，从而加快域适应。Vid2Seq 用专门的时态增强语言模型标记，便于预测事件边界和统一输出序列中的文本描述 [29]。代码。RBPS [30] 在小规模学习任务中表现出色，通过检索与开发人员目标一致的代码示例通过编码和频率分析。这种方法具有在诸如测试断言生成和程序修复等任务中展示了有效性。对于结构化的知识，焦炭方法 [31] 首先提取与输入查询相关的事实 从知识图谱中，然后将这些事实集成为提示在输入中，提高知识图谱的性能问题的任务 [32]。

这篇文章 [33] 对 RAG 和长上下文（Long-Context，简称 LC）LLMs 进行了全面比较，旨在利用两者的优势。

4.2 外部知识的质量控制

在考量大型语言模型（LLMs）的安全性和操纵性时，外部知识的质量控制至关重要。检索质量是检索增强型语言模型（RAG）系统有效性的根本，它直接影响到生成内容的相关性和准确性。然而，现有的检索方法常常面临挑战，比如数据中的噪声、不相关文档和碎片化信息，这些都可能干扰生成过程的质量。

关于数据中的噪声，分为有益噪声和有害噪声两大类。有益噪声，如语义噪声、数据类型噪声和非法句子噪声，可以提高模型的性能，增强模型对正确信息的识别能力。有害噪声，如反事实噪声、支持性噪声和拼写错误噪声，则会降低模型的性能。为了提高系统的抗噪声能力，通过系统地引入和管理噪声，可以提高模型的鲁棒性和适应性。

在检索结果不尽如人意时，模型可能会尝试生成不准确的回答，这增加了错

误输出的风险。这种情况在查询含糊不清或缺乏足够上下文时尤为突出，使得检索模型难以找到相关的文档。例如，HyDE [34] 通过生成一个能够捕捉查询核心的伪文档来解决这一问题。这种方法通过允许检索系统从非最优查询中检索到更多相关文档，从而提高了检索的准确性，尽管这可能会增加计算成本。未来的研究可以探索如何优化这一过程，以在不牺牲检索精度的情况下减少延迟。

对于信息的集成，复杂查询通常需要从多个文档中整合信息，但碎片化或相互矛盾的信息可能导致生成的答案不连贯或不完整。预检索和后检索技术在这里扮演了重要角色。通过提高检索粒度和采用实体级检索及重新排序技术，可以增强检索文档的连贯性。然而，许多后检索方法严重依赖于 LLM API 的调用，这可能导致成本过高。因此，研究更经济的替代方案，如将知识蒸馏到轻量级模型，可以提供更具可扩展性的解决方案，使高级检索策略在在线环境中更加实用。

4.3 计算效率的优化

RAG 系统在处理大型数据集和实时应用时，系统效率仍然是一个显著的瓶颈。通过使用轻量级搜索方法、混合检索方法、可微分索引和优化的深度学习模型，可以提高系统性能和效率。而 RALM 带来了显著的计算开销，特别是在需要迭代推理的场景中。未来的研究可以专注于优化这些模型或开发检索精简技术，以减少传递到生成阶段的文档数量，同时不影响性能。

模块化工作流优化。RAG 系统的复杂性通常源于诸如分块策略、嵌入模型和重排序算法等组件之间的相互依赖。模块化设计是提高系统吞吐量的关键，它允许独立优化每个步骤，同时考虑跨组件的交互 [35]。先进的分块方法和混合搜索策略可以提供在最大程度上提高检索精度和速度之间的权衡 [36]。

4.4 RALM 面临的安全问题

LLM 系统大大提高了工作效率，但是大语言模型的滥用会导致会导致负面的社会后果。这种滥用包括学术欺诈、侵犯版权、网络攻击和利用软件漏洞 [37]。

TrojanRAG 是一种针对检索增强型语言模型（RALM）的新型攻击方式，它利用了 RALM 的自然漏洞来注入联合后门，从而在各种通用攻击场景中操纵基于大型语言模型（LLMs）的 API。这些攻击场景包括攻击者主动攻击、用户被动执行后门攻击以及后门式越狱攻击。TrojanRAG 不仅能够在正常推理、可转移和 CoT（思维链）中实现强大的后门激活，而且在日常查询中保持高可用性。这一点在正常查询中尤为重要，因为它意味着攻击者可以在不引起用户怀疑的情况下实施攻击 [38]。

也有研究 [39] 致力于利用 LLMs 和 RAG 框架来提高软件漏洞检测准确性。

第五章 结论

5.1 RALMs 的重要性和影响

检索增强大语言模型的诞生标志着 NLP 研究的重要转折点。首先，RALMs 通过将语言生成和信息检索相结合，解决了传统 LLMs 在知识更新和准确性上的局限性。相比于依赖静态数据的传统模型，RALMs 能够动态获取外部信息，从而生成更可靠、更新的内容 [5]。这一特性使得 RALMs 在多个知识密集型任务中展现出显著的性能提升，尤其是在医疗、法律和技术等需要精准知识的领域。通过减少幻觉现象和提高信息检索能力，RALMs 在提升生成内容的可信度和减少错误率方面取得了显著成效 [40]。其次，RALMs 还推动了 LLMs 的应用扩展，尤其是在需要动态更新知识的场景中表现出色。这种结合检索机制的模型不仅提高了生成的质量，还有效降低了依赖大规模参数进行推理的计算成本，从而提升了模型在实际应用中的可操作性。

5.2 RALMs 的未来展望

展望未来，检索增强大语言模型的研究前景广阔。随着信息检索技术的不断进步，RALMs 在检索效率和知识库管理方面有望进一步优化。目前，检索系统的响应速度和知识库的覆盖范围仍然是 RALMs 面临的挑战之一，但通过改进检索算法和优化外部知识库，未来的 RALMs 将能够更快、更准确地检索相关信息。此外，跨模态检索和生成任务也是 RALMs 的一个重要研究方向。未来的模型将不仅局限于文本信息的检索，还能够整合图像、音频、视频等多种模态的数据，实现更丰富的生成结果。这种跨模态检索能力将为智能搜索、多媒体分析等领域带来巨大的潜力。与此同时，随着对数据隐私的日益关注，如何在保证用户隐私的前提下进行有效的检索也是未来 RALMs 需要解决的问题之一。如何平衡数据的开放性与隐私保护，将成为下一阶段技术发展的重点 [41]。最后，RALMs 在定制化领域应用中也有广阔的前景。未来的模型将能够结合特定领域的知识库，为医疗、法律、金融等行业提供更加专业化的智能服务。通过深度融合领域知识和检索增强技术，RALMs 有望成为各行业智能应用的重要组成部分。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. CoRR abs/1706.03762, 2023.

[2] Devlin J, Chang M-W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. CoRR abs/1810.04805, 2019.

[3] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners[J]. CoRR abs/2005.14165, 2020.

[4] Ji Z, Lee N, Frieske R, et al. Survey of hallucination in natural language generation[J]. ACM Computing Surveys, 2023, 55(12): 1–38.

[5] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. CoRR abs/2005.11401, 2021.

[6] Chan C-M, Xu C, Yuan R, et al. Rq-rag: Learning to refine queries for retrieval augmented generation[J]. CoRR abs/2404.00610, 2024.

[7] Hu Y, Lu Y. Rag and rau: A survey on retrieval-augmented language model in natural language processing[J]. CoRR abs/2404.19543, 2024.

[8] Ahn Y, Lee S G, Shim J, et al. Retrieval-augmented response generation for knowledge-grounded conversation in the wild[J]. IEEE Access, 2022, 10: 131374-131385.

[9] Jones K S. A statistical interpretation of term specificity and its application in retrieval, .

[10] Robertson S E, Walker S, Jones S, et al. Okapi at trec-3[J]. Nist Special Publication Sp, 1995, 109: 109.

[11] Karpukhin V, Oğuz B, Min S, et al. Dense passage retrieval for open-domain question answering[J]. CoRR abs/2004.04906, 2020.

[12] Komeili M, Shuster K, Weston J. Internet-augmented dialogue generation[J]. CoRR abs/2107.07566, 2021.

[13] Lazaridou A, Gribovskaya E, Stokowiec W, et al. Internet-augmented language models through few-shot prompting for open-domain question answering[J]. CoRR abs/2203.05115, 2022.

[14] Boytsov L, Novak D, Malkov Y, et al. Off the beaten path: Let's replace term-based retrieval with k-nn search[C]. Proceedings of the 25th ACM international on conference on information and knowledge management, 2016: 1099-1108.

[15] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. CoRR abs/1910.10683, 2023.

[16] Chen T, Wang H, Chen S, et al. Dense x retrieval: What retrieval granularity should we use?[J]. CoRR abs/2312.06648, 2024.

[17] Lee J, Wettig A, Chen D. Phrase retrieval learns passage retrieval, too[J]. CoRR abs/2109.08133, 2021.

[18] Cai P X, Fan Y C, Leu F Y. Compare encoder-decoder, encoder-only, and decoder-only architectures for text generation on low-resource datasets[C]. International Conference on Broadband and Wireless Computing, Communication and Applications, 2022: .

[19] Abeysinghe S, Wang F, Essertel G, et al. Architecting intermediate layers for efficient composition of data management and machine learning systems[J]. CoRR abs/2311.02781, 2023.

[20] Zhao S, Yang Y, Wang Z, et al. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely[J]. CoRR abs/2409.14924, 2024.

[21] Zhao P, Zhang H, Yu Q, et al. Retrieval-augmented generation for ai-generated content: A survey[J]. CoRR abs/2402.19473, 2024.

[22] Wang X, Wang Z, Gao X, et al. Searching for best practices in retrieval-augmented generation[J]. CoRR abs/2407.01219, 2024.

[23] Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: A survey[J]. CoRR abs/2312.10997, 2024.

[24] Yasunaga M, Aghajanyan A, Shi W, et al. Retrieval-augmented multimodal language modeling[J]. CoRR abs/2211.12561, 2023.

[25] Li J, Li D, Savarese S, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models[C]. International conference on machine learning, 2023: 19730-19742.

[26] Zhu W, Yan A, Lu Y, et al. Visualize before you write: Imagination-guided open-ended text generation[J]. arXiv preprint arXiv:2210.03765, 2022.

[27] Zhao J, Haffar G, Shareghi E. Generating synthetic speech from spokenvocab for speech translation[J]. arXiv preprint arXiv:2210.08174, 2022.

[28] Chan D M, Ghosh S, Rastrow A, et al. Using external off-policy speech-to-text mappings in contextual end-to-end automated speech recognition[J]. arXiv preprint arXiv:2301.02736, 2023.

[29] Yang A, Nagrani A, Seo P H, et al. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 10714-10726.

[30] Nashid N, Sintaha M, Mesbah A. Retrieval-based prompt selection for code-related few-shot learning[C]. 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), 2023: 2450-2462.

[31] Li X, Zhao R, Chia Y K, et al. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources[J]. arXiv preprint arXiv:2305.13269, 2023.

[32] Peng B, Zhu Y, Liu Y, et al. Graph retrieval-augmented generation: A survey[J]. CoRR abs/2408.08921, 2024.

[33] Li Z, Li C, Zhang M, et al. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach[J]. CoRR abs/2407.16833, 2024.

[34] Gao L, Ma X, Lin J, et al. Precise zero-shot dense retrieval without relevance labels[J]. CoRR abs/2212.10496, 2022.

[35] Gao Y, Xiong Y, Wang M, et al. Modular rag: Transforming rag systems into lego-like reconfigurable frameworks[J]. CoRR abs/2407.21059, 2024.

[36] Huang Y, Huang J. A survey on retrieval-augmented text generation for large language models[J]. CoRR abs/2404.10981, 2024.

[37] Kuppa A, Nicholls J, Le-Khac N-A. Manipulating prompts and retrieval-augmented generation for llm service providers[J]. , .

[38] Cheng P, Ding Y, Ju T, et al. Trojanrag: Retrieval-augmented generation can be backdoor driver in large language models[J]. CoRR abs/2405.13401, 2024.

[39] Du X, Zheng G, Wang K, et al. Vul-rag: Enhancing llm-based vulnerability detection via knowledge-level rag[J]. CoRR abs/2406.11147, 2024.

[40] Guu K, Lee K, Tung Z, et al. Realm: Retrieval-augmented language model pre-training[J]. CoRR abs/2002.08909, 2020.

[41] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models[C]. 2017 IEEE Symposium on Security and Privacy (SP), 2017: 3-18.