LR 1

I. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models

arxiv.org

Claude

Talk with Claude, an AI assistant from Anthropic

https://claude.ai/chat/d1d63b3a-004b-4999-ba66-98...

Claude

BY ANTHROP\C

1. 有什么问题?

随着大型语言模型(LLMs)的迅猛发展，它们在语言理解和生成等方面展现出了惊人的能力。然而，这些模型仍存在内在局限性，如幻觉现象、知识过时和领域专业知识缺乏等问题。特别是在医学和法律等特定领域，LLMs生成的幻觉信息比例高达69%至88%。同时，由于更新LLMs需要大量计算资源，使其难以及时获取最新数据和知识。这些问题严重阻碍了LLMs在实际应用中的广泛部署。此外，LLMs通常仅依赖预训练过程中学到的内部知识，无法有效获取和利用外部最新信息，导致生成结果质量受限，尤其是对于需要专业知识或最新事实的领域，如问答系统、科学研究和软件工程等应用场景。

2. 提出了什么解决方案?

为解决上述问题，研究者提出了检索增强生成(Retrieval-Augmented Generation, RAG)技术与大型语言模型的结合方案。RAG技术通过引入外部知识库，而非仅依赖模型内部知识，来增强LLMs的生成质量。具体而言，RAG首先调用检索器从外部数据库中搜索并提取与查询相关的文档，然后将这些文档作为上下文来增强生成过程。这种方法在不需要大规模重新训练的情况下，可以为LLMs提供可靠且最新的外部知识，显著提升其在知识密集型任务中的表现。该综述系统地回顾了检索增强大型语言模型(RA-LLMs)的研究进展，从架构设计、训练策略和应用领域三个主要技术视角进行了全面总结。

3. 论文有什么亮点

本综述的主要亮点在于其全面系统地梳理了RA-LLMs领域的研究进展，并从不同的技术视角提供了深入分析。首先，作者从检索、生成和增强三个核心组件详细阐述了RA-LLMs的架构设计，讨论了检索的必要性和应用频率。其次，综述总结了RA-LLMs的主要训练技术，包括无训练方法、独立训练、顺序训练和联合训练等不同策略的优缺点。第三，作者分析了RA-LLMs在自然语言处理、下游任务和特定领域的广泛应用，如问答系统、聊天机器人、推荐系统、软件工程、科学研究和金融等。最后，论文深入讨论了当前RA-LLMs面临的挑战和未来研究方向，如可信RA-LLMs、多语言RA-LLMs、多模态RA-LLMs以及外部知识质量等关键问题，为后续研究提供了宝贵的洞见。

4. 论文具体是怎么实现的

论文通过系统梳理RA-LLMs的三个核心组件及其训练策略和应用领域来实现全面综述：

检索组件：作者详细分析了检索器类型、检索粒度、检索增强技术和数据库构建。检索器主要分为稀疏检索(如TF-IDF、BM25)和密集检索(如DPR、Contriever)两类。检索粒度granularity包括文档、段落、词元和实体级别检索。论文同时介绍了检索前增强(如查询扩展、查询重写)和检索后增强(如PRCA、R2G、BlendFilter)技术，以及封闭源和开放源数据库的构建方法。

生成组件：综述区分了参数可访问(白盒)和参数不可访问(黑盒)两类生成器。白盒生成器如T5和BART允许参数优化，可适应不同的检索和增强方法。黑盒生成器如GPT系列模型则主要通过增强输入来提升生成质量。

增强组件：作者介绍了三种主要的增强设计：输入层集成(如In-Context RALM、FID、Atlas)将检索内容与原始输入结合；输出层集成(如kNN-LM、REFEED)结合检索和生成结果；中间层集成(如RETRO、EAE)则通过内部层增强生成模型能力。

训练策略：论文详细讨论了四种训练策略：

1. 无训练方法：直接利用检索内容，无需额外训练，如提示工程和检索引导的词元生成。
2. 独立训练：检索器和LLMs作为两个独立过程进行训练，如DPR和CoG。
3. 顺序训练：先训练一个模块再固定它来训练另一个，分为"检索器优先"(如RETRO、ITERRTGEN)和"LLMs优先"(如DKRR、UPRISE)两种方式。
4. 联合训练：同时优化检索器和生成器，如RAG和REALM。

应用领域：综述展示了RA-LLMs在三大类应用中的实践：

1. NLP应用：问答系统(REALM、FiD)、聊天机器人(KDBTS)和事实验证(Atlas、Self-RAG)。
2. 下游任务：推荐系统(CoRAL)和软件工程(Docprompting)。
3. 特定领域应用：科学研究(RetMol、RSA)和金融(AlphaFin)。

论文通过时间线梳理了主要RA-LLMs模型的发展脉络，并提供了它们的核心技术设计和应用场景的详细比较，为读者提供了全面的研究概览。

5. 可能的改进

尽管RA-LLMs取得了显著进展，但仍存在几个可能的改进方向。首先，可信RA-LLMs的构建至关重要，未来研究应关注如何提高RA-LLMs的稳健性、公平性、可解释性和隐私保护能力，尤其是在安全关键场景中。其次，多语言RA-LLMs的发展可以打破语言障碍，使不同语言地区的用户能够利用丰富的英文和中文语料库进行知识检索。第三，多模态RA-LLMs的研究可以扩展知识源至图像、视频和音频等多种数据模态，为LLMs提供更丰富的上下文信息。最后，外部知识质量的提升也是关键，如何筛选高质量的知识并减轻低质量信息对模型生成的负面影响值得深入探索。此外，对于检索机制本身的改进也是未来的重要方向，如何更准确地判断检索的必要性，以及如何在大规模知识库中更高效地进行相关信息检索，都是值得进一步研究的问题。

II. Graph Retrieval-Augmented Generation: A Survey

arxiv.org

1. 有什么问题?

大型语言模型(LLMs)如GPT-4、Qwen2和LLaMA在自然语言处理领域带来了革命性变化，但它们仍存在三个关键限制：首先，LLMs往往缺乏领域特定知识、实时更新信息和专有知识，导致"幻觉"现象；其次，传统的检索增强生成(RAG)系统虽然通过引入外部文本知识库来改善输出质量，但往往忽略文本之间的结构化关系；最后，**传统RAG系统通过拼接文本片段构建提示，容易导致上下文过长，引发"丢失在中间"的问题，并且无法把握全局信息。这些限制严重影响了LLMs在需要精确关系理解和推理的复杂任务中的表现。**

2. 提出了什么解决方案?

为解决上述问题，论文提出了图检索增强生成(GraphRAG)框架，它通过从预构建的图数据库中检索与查询相关的图元素来增强LLMs的生成能力。GraphRAG将文本之间的**结构化关系**显式地表示为图，包括节点、三元组、路径或子图**（vertex/Nodes, triples, paths or subgraphs），主体（Subject）、谓词（Predicate）和客体（Object）**，从而更精确地捕获关系知识。与传统RAG相比，GraphRAG考虑了文本之间的互联关系，实现了更准确、更全面的关系信息检索。此外，图数据(如知识图谱)提供了对文本数据的抽象和概括，显著缩短了输入文本的长度，减轻了冗长的问题。通过检索子图或图社区，GraphRAG能够访问更全面的信息，更有效地解决查询聚焦摘要(QFS)等挑战。

3. 论文有什么亮点

本综述是首个全面系统地对GraphRAG技术进行调研的工作，主要亮点有：首先，提供了GraphRAG的正式定义，概述了包括**G-Indexing(图索引)、G-Retrieval(图检索)和G-Generation(图增强生成)**在内的通用工作流程；其次，详细讨论了现有GraphRAG系统的核心技术，包括模型选择、方法设计和增强策略的全谱分析，并对比了各模块采用的不同训练方法；第三，明确划分了GraphRAG相关的下游任务、基准测试、应用领域、评估指标和未来研究方向，全面讨论了该领域的进展和前景；最后，编纂了现有工业GraphRAG系统清单，提供了学术研究转化为实际工业解决方案的见解，为该领域的研究者和实践者提供了全面的参考框架。

4. 论文具体是怎么实现的

论文将GraphRAG框架分解为三个主要阶段，并对每个阶段进行了详细分析：

图索引(G-Indexing)：这是GraphRAG的初始阶段，旨在识别或构建与下游任务相匹配的图数据库并建立索引。图数据可以来自公共知识图谱(如Wikidata、Freebase、DBpedia)或基于私有数据源构建的自定义图。索引方法包括**图索引**(保留完整图结构)、**文本索引**(将图数据转换为文本描述)、**向量索**引(转换为向量表示)和混合索引(结合上述方法)。这一阶段决定了后续检索的粒度，对提高查询效率起着关键作用。

图检索(G-Retrieval)：在索引建立后，检索阶段专注于从图数据库中提取与用户查询相关的信息。检索器类型包括非参数检索器(基于**启发式规则**或传统**图搜索算法**)、基于语言模型的检索器(利用LMs的自然语言理解能力)和基于图神经网络的检索器(理解和利用复杂图结构)。检索范式可分为一次性检索、迭代检索和多阶段检索。检索粒度则包括节点、三元组、路径、子图和混合粒度。论文还介绍了查询增强(查询扩展和查询分解)和知识增强(合并和剪枝)等技术来提高检索质量。

图增强生成(G-Generation)：生成阶段将检索到的图数据与查询集成，生成最终响应。生成器可以是图神经网络(GNNs，适用于图数据的强大表示能力)、语言模型(LMs，擅长文本理解和生成)，或混合模型(结合GNNs和LMs优势)。由于图数据的非欧几里德性质，需要将其转换为生成器可接受的格式，包括图语言(邻接/边表、自然语言、代码形式、语法树和节点序列)和图嵌入。生成增强技术分为生成前增强(在生成前丰富检索到的图数据)、生成中增强(在生成过程中调整策略)和生成后增强(整合多个生成响应)。

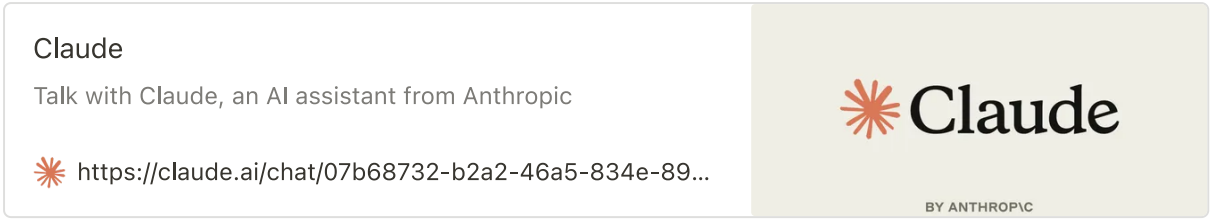
对于训练策略，论文区分了无训练和有训练方法。无训练方法主要用于闭源LLMs，通过精心设计的提示控制检索和生成能力。有训练方法则通过监督信号优化模型，提高检索或生成内容的质量和相关性。联合训练检索器和生成器可以增强它们的协同作用，提高下游任务性能。

5. 可能的改进

未来GraphRAG技术的改进可从多个方向展开：首先，动态和自适应图更新机制需要进一步发展，以快速整合新兴实体和关系；其次，多模态信息集成至关重要，将图像、音频和视频等多样化数据类型融入知识图谱可提供更全面的理解，但也增加了图的复杂性和规模；第三，针对包含数百万甚至数十亿实体的大规模知识图谱，需要开发更高效的检索算法和可扩展基础设施；第四，将图基础模型与GraphRAG结合，可能提高图结构数据的处理效率；第五，研发无损压缩技术以处理长上下文，平衡压缩率和信息保留；第六，建立统一标准基准以评估不同方法；最后，扩展GraphRAG在医疗、金融服务、法律合规、智慧城市等更广泛应用领域的应用，将进一步释放其潜力，为各领域提供更复杂、更有针对性的解决方案。

III. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG

arxiv.org



1. 有什么问题?

大型语言模型(LLMs)已经在人工智能领域带来革命性变化，实现了类人文本生成和自然语言理解。然而，它们依赖静态训练数据的局限性导致无法对动态、实时查询做出响应，从而产生过时或不准确的输出。检索增强生成(RAG)作为解决方案，通过整合实时数据检索来提供上下文相关且最新的回答。**但传统RAG系统受限于静态工作流程，缺乏处理多步推理和复杂任务管理所需的适应性。**传统RAG面临三大核心挑战：上下文整合困难、多步推理能力有限、以及扩展性和延迟问题。这些局限使得系统难以处理需要跨多个步骤收集和整合信息的复杂场景，从而限制了在现实世界中的适用性。

2. 提出了什么解决方案?

论文提出了基于代理的检索增强生成(Agentic RAG)作为解决方案，它通过将自主AI代理嵌入到RAG流程中来超越这些限制。这些代理利用代理设计模式——反思、规划、工具使用和多代理协作，动态管理检索策略，迭代细化上下文理解，并通过明确定义的操作结构(从顺序步骤到自适应协作)调整工作流程。Agentic RAG引入了关键的工作流模式，包括提示链接、路由、并行化、协调者-工作者模型和评估者-优化者模式，以结构化和优化任务执行。这种创新整合使Agentic RAG系统能够在各种应用场景中提供前所未有的灵活性、可扩展性和上下文感知能力，解决了传统RAG系统在处理复杂多领域查询时的固有约束。

3. 论文有什么亮点

本综述的突出亮点在于它全面而系统地介绍了Agentic RAG领域，为理解这一新兴技术提供了坚实基础。首先，论文详细分析了RAG范式的演变历程，从简单的Naïve RAG到更复杂的高级RAG、模块化RAG和图RAG，突显了每种方法的特点和局限性。其次，论文提出了全面的Agentic RAG分类法，涵盖单代理、多代理和基于图的框架，为研究人员提供了清晰的概念框架。第三，论文探讨了Agentic RAG在医疗保健、金融和教育等多个领域的应用，展示了其实际价值。最后，论文深入分析了实施策略、基准测试和道德考量，为未来研究和实际部署提供了宝贵见解，同时建立了GitHub资源库支持开放协作研究，促进该领域的持续发展。

4. 论文具体是怎么实现的

论文采用系统化方法详细阐述了Agentic RAG的实现原理和架构：

首先，论文全面分析了RAG的基础组件和演变过程。RAG系统包含三个核心模块：检索(负责查询外部数据源)、增强(处理检索数据以提取相关信息)和生成(结合检索信息与预训练知识生成回应)。论文追踪了从简单关键词检索的朴素RAG，到引入语义理解的高级RAG，再到具有混合检索策略和外部工具整合的模块化RAG，以及利用图结构的图RAG的演变历程。

其次，论文深入探讨了代理智能的核心原则。AI代理由LLM(执行推理)、记忆(捕获上下文)、规划(指导迭代推理)和工具(扩展能力)组成。关键的代理模式包括：反思(允许代理评估和完善输出)、规划(分解复杂任务)、工具使用(与外部资源交互)和多代理(实现任务专业化和并行处理)。

第三，论文详述了代理工作流模式的实现方式。这包括：提示链接(将复杂任务分解为顺序步骤)、路由(将输入引导到专业流程)、并行化(并发执行任务减少延迟)、协调者-工作者(动态任务分配)和评估者-优化者(通过迭代改进内容)。

论文还提出了详细的Agentic RAG系统分类学：

1. 单代理Agentic RAG：集中式决策系统，一个代理管理检索、路由和整合，适用于工具或数据源有限的设置。
2. 多代理Agentic RAG：模块化可扩展系统，通过多个专业代理分配责任，每个代理专注于特定数据源或任务。
3. 层次化Agentic RAG：采用多层次方法，高层代理监督和指导低层代理，实现多层决策。
4. 纠错型Agentic RAG：引入机制评估和纠正检索结果，通过查询细化和外部知识整合提高质量。
5. 自适应Agentic RAG：根据查询复杂性动态调整处理策略，优化计算资源。
6. 基于图的Agentic RAG：整合图知识库与非结构化文档检索，增强推理能力。
7. 代理文档工作流：实现端到端知识工作自动化，协调复杂文档处理过程。

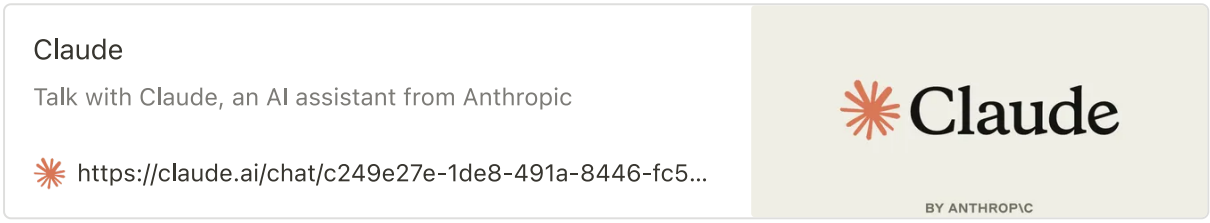
论文还详细讨论了现有框架(如LangChain、LangGraph、LlamaIndex、CrewAI)的实现工具，以及评估这些系统的基准和数据集，为研究人员和实践者提供了全面的资源指南。

5. 可能的改进

尽管Agentic RAG带来了显著进步，但仍有多个潜在改进方向。首先，减少多代理架构中的协调复杂性是关键挑战，需要开发更高效的协调机制来简化代理间通信。其次，降低计算开销非常重要，特别是在处理复杂工作流和大规模查询时，研究优化策略和资源分配算法可以提高系统效率。第三，创建专门针对代理能力的评估基准至关重要，当前缺乏专门评估多代理协作和动态适应性的标准化测试集。此外，加强代理系统的解释性和透明度能增强用户信任，同时探索跨模态代理能力将扩展系统处理多种数据类型的能力。最后，为确保负责任部署，需要更深入研究道德考量和安全机制，以防止潜在的错误信息传播和偏见放大。这些改进将推动Agentic RAG向更强大、高效且可信的系统发展。

IV. REALM

2402.07016v1.pdf 1586.2KB



1. 有什么问题?

电子健康记录(EHR Electronic health record)数据在临床预测任务中扮演着重要角色，特别是多模态EHR数据(包括临床记录和多变量时间序列数据)能够提供全面的患者信息。然而，现有的深度学习模型在处理这些数据时存在几个关键限制：首先，这些模型缺乏**医学专业知识**，无法像医疗专业人员那样深入理解医学概念和上下文；其次，虽然有些方法尝试通过知识图谱(KG)引入额外的医学知识，但它们主要集中在从结构化数据中提取医学概念和关系，忽略了非结构化模态的重要性；第三，即使是利用大型语言模型(LLM)构建的知识图谱也面临幻觉问题，可能产生不准确或虚构的信息。这些限制严重影响了模型在临床预测任务中的准确性和可靠性。

2. 提出了什么解决方案?

针对上述问题，论文提出了REALM框架，这是一种基于检索增强生成(RAG)的多模态EHR表示学习框架。该框架首先利用LLM编码长上下文临床记录，并使用GRU模型编码时间序列EHR数据。然后，通过提示LLM从这些数据中提取与任务相关的医学实体，并将这些实体与专业标注的外部知识图谱(PrimeKG)中的相应医学知识进行匹配。通过与临床标准的匹配和对齐，REALM框架消除了幻觉问题并确保了一致性。最后，论文提出了一种自适应多模态融合网络，将提取的知识与多模态EHR数据整合在一起。在MIMIC-III数据集上的死亡率和再入院预测任务中的实验表明，REALM框架优于基线模型，证明了该方法的有效性。

3. 论文有什么亮点

REALM框架的主要亮点在于其创新地将检索增强生成(RAG)应用于多模态EHR数据分析。首先，与以往方法不同，REALM不仅从结构化数据中提取知识，还能处理非结构化的临床记录和时间序列数据，这更符合实际临床环境中的数据形式。其次，REALM通过严格的实体验证和匹配过程，确保了从LLM提取的医学实体与专业知识图谱一致，有效解决了LLM幻觉问题。第三，REALM不仅提取简单的实体三元组，还包括更丰富的实体定义和描述，能够捕获更复杂的语义医学背景知识。最后，REALM设计的自适应多模态融合网络通过自注意力和交叉注意力机制，有效地整合了各种模态的信息，提高了预测性能。实验结果显示，即使在数据稀疏的情况下，REALM也表现出色，证明了其在实际临床应用中的潜力。

4. 论文具体是怎么实现的

REALM框架的实现分为三个主要模块：

多模态EHR嵌入提取：

- 对于时间序列数据，使用门控循环单元(GRU)网络捕获时间依赖性，将时间相关信息编码为嵌入向量hTS。
- 对于临床记录文本，采用大型语言模型(LLM)作为编码器，其优势在于处理长文本上下文的能力，输出嵌入向量hText。
- 对于时间戳信息，通过类似Transformer位置嵌入机制的正弦/余弦变换进行编码，以保留时间序列数据中的周期性连续性，生成嵌入向量hTime。

RAG驱动增强管道：

- 时间序列数据处理：计算每个特征的z分数，识别超过阈值ε的异常值作为实体。这些异常特征通常代表不健康的身体状况。
- 临床记录处理：通过LLM提取疾病实体，并进行后处理以减少幻觉，包括丢弃不在原始文本中出现的实体、使用LLM过滤非疾病类型实体、删除语义上重复的实体等。
- 实体匹配：使用基于语义的**密集向量检索方法**，将提取的实体与知识图谱中的节点进行匹配。计算余弦相似度，并设置阈值η判断实体是否与知识图谱中的节点密切相关。
- 知识编码：将匹配的实体详细信息（实体名称、定义、描述）连接起来，使用LLM进行编码，生成嵌入向量hRAG。

多模态融合网络：

- 首先将从实体提取的隐藏表示与文本的隐藏表示连接，并使用MLP网络映射到统一维度。
- 对每个模态应用自注意力机制。

- 使用一个模态的输出作为查询，另一个模态的输出作为键和值计算交叉注意力。
- 对每个多头注意力层和前馈网络应用残差连接和BatchNorm。
- 将两个交叉注意力模块的输出相加，通过注意力池化层获取融合信息。
- 最终将融合表示通过单层MLP网络得到预测结果。

在训练过程中，使用BCE损失函数作为二元分类任务的损失函数。整个过程是离线进行的，确保了隐私和数据安全。实验结果表明，REALM在MIMIC-III数据集上的死亡率和再入院预测任务中，在AUROC、AUPRC、min(+P, Se)和F1分数等多个评估指标上都优于基线模型。

5. 可能的改进

尽管REALM框架在多模态EHR数据分析方面取得了显著进展，但仍存在一些可能的改进方向：

首先，当前的实体提取和匹配过程依赖于设定的阈值（ε和η），这些阈值的选择可能影响模型性能。未来可以探索更自适应的阈值选择方法，或者采用无需阈值的端到端学习方法。其次，虽然REALM通过后处理步骤减少了LLM幻觉，但随着更先进LLM的发展，可以探索更有效的幻觉检测和减轻方法。第三，当前的多模态融合网络虽然有效，但可以进一步探索更复杂的融合策略，例如考虑不同模态之间的时间对齐问题或引入更多注意力机制。最后，REALM目前主要应用于二元分类任务（死亡率和再入院预测），未来可以将其扩展到更多样化的临床预测任务，如疾病诊断、治疗响应预测等，进一步验证其泛化能力。总的来说，随着医疗知识图谱和大型语言模型的不断发展，REALM框架有潜力在更广泛的临床应用中发挥作用。

V. GraphRAG

arxiv.org

1. 有什么问题?

当前的检索增强生成(RAG)系统在回答针对整个文本语料库的全局问题时存在明显不足。传统的向量RAG方法主要适用于回答可以从少量文档中找到答案的具体问题，但无法有效处理需要**全局理解的宏观问题(如"数据集中的主要主题是什么?")**。这类全局性问题本质上是一个**查询聚焦摘要(QFS)任务**，而非简单的检索任务。同时，现有的QFS方法又无法扩展到典型RAG系统索引的大量文本。因此，需要一种能够兼顾用户问题的通用性和源文本数量的扩展性的新方法，以弥补现有技术的不足。

2. 提出了什么解决方案?

论文提出了GraphRAG，一种基于图的问答方法，能够随着用户问题的普遍性和源文本数量的增加而扩展。GraphRAG使用LLM通过两个阶段构建图索引：首先从源文档中提取实体知识图谱，然后为所有密切相关的实体组预生成社区摘要。当收到问题时，系统使用每个社区摘要生成部分响应，然后将所有部分响应汇总为最终答案。这种方法结合了RAG和QFS的优势，使系统能够处理需要全局理解的问题，同时保持处理大规模文本的能力，特别适合于100万个标记范围内的全局性问题。

3. 论文有什么亮点

论文最大的亮点是提出了一种创新的图索引方法，该方法不仅能够处理局部问题，还能有效回答需要全局理解的问题。特别是，GraphRAG利用图的固有模块性和社区检测算法，创建嵌套的模块化社区，并通过LLM生成覆盖这些社区的递归摘要。此外，论文还开发了一种新颖的评估技术，使用"LLM-as-a-judge"方法来评估针对没有标准答案的广泛问题的回答质量。实验表明，在全面性和多样性方面，GraphRAG显著优于传统的向量RAG基线。论文的方法已开源，并已集成到多个开源库中，包括LangChain、LlamaIndex、NebulaGraph和Neo4J。

4. 论文具体是怎么实现的

GraphRAG的实现包含两个主要阶段：索引时和查询时。

索引时阶段：

1. **文本分块**：首先将文档分割成重叠的文本块，选择适当的块大小以平衡信息提取效率和准确性。
2. **实体与关系提取**：使用LLM从每个文本块中提取重要实体、实体间关系以及相关声明。例如，从一个关于NeoChip公司的文本中，LLM会提取出"NeoChip"和"Quantum Systems"这两个实体，以及它们之间的所有权关系。**Named-Entity recognition & Relation Extraction**
3. **知识图谱构建**：将提取的实体、关系和声明聚合成知识图谱，其中实体成为节点，关系成为边，重复关系的数量成为边的权重。

4. **社区检测**：使用Leiden社区检测算法以分层方式对图进行分区，递归地检测每个社区内的子社区，直到达到无法再分区的叶社区。 **High cohesion**
5. **社区摘要生成**：为社区层次结构中的每个社区创建摘要。对于叶级社区，根据节点度数的组合优先级添加元素摘要；对于更高级别的社区，若所有元素摘要不超过上下文窗口，则直接摘要所有元素，否则用子社区摘要替代相关元素摘要。

查询时阶段：

1. **准备社区摘要**：随机打乱社区摘要并分成预定大小的块，确保相关信息分布在不同块中。 **Shuffle**
2. **映射社区答案**：并行生成中间答案，LLM还会生成0-100的得分表示该答案对目标问题的帮助程度，得分为0的答案被过滤掉。
3. **归约为全局答案**：将中间社区答案按帮助度得分降序排列，并依次添加到新的上下文窗口中，直到达到标记限制，最后使用此上下文生成返回给用户的全局答案。

论文还开发了一种特殊的全局性问题生成方法，用于评估系统。该方法首先要求LLM根据语料库描述生成假设用户角色，然后为每个用户生成任务，最后为每个用户-任务组合生成需要全局理解的问题。评估标准包括全面性、多样性和赋能性三个方面。实验在两个包含约100万标记的真实世界数据集上进行，结果显示GraphRAG在全面性和多样性方面显著优于传统的向量RAG，同时提供了比直接文本摘要更高效的查询处理方式。

5. 可能的改进

GraphRAG虽然效果显著，但仍有几个方面可以进一步改进。首先，当前的评估主要集中在特定领域的两个语料库上，需要更多不同领域和使用场景的测试以验证方法的泛化能力。其次，可以引入像SelfCheckGPT这样的方法来检测和减少幻觉生成的可能性。此外，可以探索将嵌入式匹配与实时社区报告生成相结合的混合RAG方案，实现更灵活的信息检索。更"自上而下"的方法也可以扩展到社区层次结构的多个层次，或实现为更具探索性的"自上而下"机制，遵循高级社区摘要中包含的信息线索。最后，系统可以更好地保留原始文本中的具体示例、引用和引用，这对帮助用户达成知情理解至关重要，这需要通过调整元素提取提示来保留更多细节信息。**Increment**

VI. LightRAG

1. 有什么问题?

现有的检索增强生成(RAG)系统存在几个关键限制，影响了它们在处理复杂查询时的性能。首先，许多方法依赖于平面数据表示，这限制了它们理解和基于实体之间复杂关系检索信息的能力。其次，这些系统通常缺乏维持各种实体及其相互关系之间连贯性所需的上下文感知能力，导致对用户查询的回答可能不完整。例如，当用户询问"电动汽车的兴起如何影响城市空气质量和公共交通基础设施？"时，现有RAG方法可能会检索有关电动汽车、空气污染和公共交通挑战的单独文档，但难以将这些信息合成为一个连贯的回答。它们可能无法解释电动汽车的采用如何改善空气质量，这反过来可能影响公共交通规划。结果，用户可能会收到一个分散的答案，无法充分捕捉这些主题之间复杂的相互依存关系。

2. 提出了什么解决方案?

为了解决这些限制，作者提出了LightRAG，这是一个将图结构无缝集成到文本索引和检索过程中的模型。LightRAG采用双层检索系统，增强了从低层和高层知识发现中进行全面信息检索的能力。低层检索专注于特定实体及其关系的精确信息，而高层检索则涵盖更广泛的主题和主题。此外，图结构与向量表示的集成促进了相关实体及其关系的高效检索，同时通过从构建的知识图中获取相关结构信息来增强结果的全面性。LightRAG还实现了增量更新算法，确保新数据的及时集成，使系统在快速变化的数据环境中保持有效和响应。通过结合详细和概念性检索，LightRAG有效地适应了各种查询，确保用户接收到针对其特定需求的相关和全面的响应。

3. 论文有什么亮点

LightRAG的主要亮点在于其综合的图增强RAG系统，其中包含三个创新方面。首先，在方法论上，LightRAG将**双层检索范式**与图增强文本索引相集成，这种方法既捕获**低层信息(实体和关系)**又捕获**高层信息(主题和主题)**，实现了全面而经济高效的检索。其次，LightRAG采用图结构表示文本，使其能够有效地理解实体之间的相互依存关系。这一特性对于生成连贯、上下文丰富的回答至关重要，特别是在处理复杂查询时。最后，通过消除重建整个索引的需要，LightRAG减少了计算成本并加速了适应速度。其增量更新算法确保了新数据的及时集成，维持了系统在动态环境中的有效性。实验结果表明，与现有方法相比，LightRAG在检索准确性、响应效率和对新信息的适应性方面都有显著提高。

4. 论文具体是怎么实现的

LightRAG的实现包含三个主要组件：图基文本索引、双层检索范式和增量知识库更新。

图基文本索引：
LightRAG首先将文档分割成更小、更易管理的片段，然后利用大型语言模型(LLM)识别和提取各种实体(如名称、日期、位置和事件)以及它们之间的关系。这一过程包括三个关键功能：

1. 实体和关系提取(R(·))：该函数提示LLM在文本数据中识别实体(节点)和它们的关系(边)。
2. LLM分析生成键值对(P(·))：使用LLM为图中的每个实体节点和关系边生成文本键值对。每个索引键是一个能够实现高效检索的单词或短语，相应的值是一个文本段落，汇总外部数据中相关片段以辅助文本生成。
3. 去重优化图操作(D(·))：该功能识别并合并来自原始文本不同段落的相同实体和关系，有效减少了与图操作相关的开销。

双层检索范式：
为了从文档片段和它们的复杂相互依存关系中检索相关信息，LightRAG在详细和抽象两个层次生成查询关键词：

1. 低层检索：主要关注检索特定实体及其关联属性或关系。该层级的查询是面向细节的，旨在提取关于图中特定节点或边的精确信息。
2. 高层检索：处理更广泛的主题和总体主题。该层级的查询聚合多个相关实体和关系的信息，提供对更高层次概念和摘要的见解。

LightRAG的检索算法通过以下步骤结合图结构和向量表示：

1. 查询关键词提取：对于给定查询q，LightRAG的检索算法首先提取本地查询关键词k(l)和全局查询关键词k(g)。
2. 关键词匹配：算法使用高效的向量数据库将本地查询关键词与候选实体匹配，将全局查询关键词与链接到全局键的关系匹配。
3. 整合高阶相关性：为增强查询的高阶相关性，LightRAG进一步收集检索到的图元素的局部子图中的相邻节点。

增量知识库更新：
为了高效适应不断变化的数据，LightRAG实现了增量更新算法，无需完全重新处理整个外部数据库：

1. 对于新文档D'，增量更新算法使用与之前相同的图基索引步骤ϕ处理它，得到Dˆ' = (Vˆ', Eˆ')。
2. 随后，LightRAG通过取节点集Vˆ和Vˆ'的并集以及边集Eˆ和Eˆ'的并集，将新图数据与原始数据结合。

这种设计确保了无缝集成新数据并减少了计算开销，使LightRAG在维持系统准确性的同时，提供当前信息并保存资源。

5. 可能的改进

尽管LightRAG在提高RAG系统性能方面取得了显著进展，仍有几个方面可以进一步改进。首先，当前的实现主要关注文本数据，但扩展到多模态数据(如图像、视频和音频)将大大增强其在各种应用场景中的实用性。其次，尽管LightRAG的增量更新机制有效，但进一步优化图结构以处理超大规模数据集将提高其在企业环境中的可扩展性。此外，探索将更多类型的关系(如时间和因果关系)集成到知识图中可能提供更丰富的上下文理解。最后，当前的评估主要集中在生成性能上，将这一评估扩展到包括检索准确性、响应时间和计算效率的更全面指标将提供对LightRAG优势的更全面理解。与其他最先进的大型语言模型的集成也可能进一步提升系统性能，特别是在处理复杂查询方面。这些改进将使LightRAG更加强大，适用于更广泛的应用场景。

VII. TimeRAG

1. 研究问题

时间序列预测是数据科学和机器学习研究中的关键任务，广泛应用于金融市场分析、需求预测和天气预测等领域。尽管基于深度学习的预测方法（如LSTM、Reformer和Informer）在经典基准测试上取得了令人满意的性能，但这些方法难以捕捉大规模序列数据中复杂的隐藏模式和依赖关系。研究人员开始探索将大型语言模型（LLMs）应用于时间序列分析和预测，但现有的时间序列预测LLMs存在明显局限性：它们不能轻易适应不同领域，计算成本高昂，通常只针对特定领域进行优化。此外，由于LLMs的"幻觉"问题，它们可能生成不准确的预测、异常值或与数据不符的模式，且缺乏可解释性。

2. 提出的解决方案

论文提出了TimeRAG框架，将检索增强生成（Retrieval-Augmented Generation，RAG）技术整合到时间序列预测LLMs中。该框架首先从训练集通过K-means聚类构建时间序列知识库，然后对于给定的预测查询，采用动态时间规整（Dynamic Time Warping，DTW）作为距离度量，从知识库中检索与查询具有相似波形和趋势的序列作为参考。最后，将输入查询和参考序列重写为自然语言提示，输入LLMs进行预测。与现有需要大量训练成本的时间序列LLMs和之前的RAG解决方案不同，TimeRAG是首个专为时间序列数据预测设计的RAG框架，无需修改底层LLM的基础参数。实验结果表明，该方法在与同类LLMs和基线模型比较时表现出强大的竞争力。

3. 论文亮点

TimeRAG框架在时间序列预测领域具有三个主要创新点：首先，这是首个将检索增强生成技术应用于时间**序列预测LLMs的尝试**，显著提高了预测准确性。实验验证表明，RAG技术平均提升了序列预测准确率2.97%，在最优情况下甚至提高了13.12%。其次，团队采用K-means聚类和动态时间规整技术高效构建时间序列知识库，这使LLM能够轻松适应不同领域的时间序列数据。与传统方法不同，TimeRAG不需要修改底层LLM的基础参数，降低了计算成本。第三，该方法为解决LLMs在时间序列预测中的"幻觉"问题提供了新思路，通过引入相似历史序列作为参考，增强了预测结果的可靠性和可解释性。

4. 具体实现方法

TimeRAG框架包含两个主要组件：时间序列知识库（Time Series Knowledge Base）和基于LLM的时间序列预测模型。

时间序列知识库构建：

1. 序列切片：给定时间序列X = (x_t, ...x_{t+n})，采用滑动窗口方法，步长为S，窗口长度为L，将X切分为多个子序列X_L。
2. K-means聚类：对切分后的N个序列片段Q_L = {X_L^i}应用K-means聚类，首先初始化k个聚类中心C = {X_c^1, ..., X_c^k}，将每个X_L^i分配给最近的中心。距离使用欧几里得距离d = ||X_L^i - X_c^j||_2计算。
3. 聚类迭代：K-means迭代更新每个聚类中心为该聚类内序列的平均值，并重新分配每个序列到最近的聚类，逐步最小化所有点与其对应聚类中心之间的总距离和。
4. 知识库构成：通过收集每个聚类中与其中心最接近的序列来构建时间序列知识库。

检索增强时间序列预测：

1. 相似序列检索：给定输入查询序列X_{input}，TimeRAG使用DTW从知识库中检索最相似的序列。具体而言：
    - 为知识库中的每个序列X_L构建一个n×L矩阵，其中元素(i,j)表示X_{input}^i和X_L^j之间的距离d(i,j) = (X_{input}^i, X_L^j)^2。
    - 定义从矩阵元素(1,1)到(n,L)的路径W，称为规整路径，其第m个元素为w_m = d(m_i, m_j)。
    - 使用动态规划获取最短规整路径，用于测量X_{input}和X_L之间的相似度Simi(X_{input}, X_L) = min(√(∑_{m=1}^M w_m)/M)。
    - 选择与查询序列最相似的前K个序列作为检索结果。

2. 模型预测：
    - TimeRAG遵循TimeLLM方法，采用重编程层将序列模态与自然语言模态对齐。
    - 输入查询序列X_{input}和检索到的序列通过重编程层转换并连接为一个提示，增强LLM的预测性能。

实验设置：

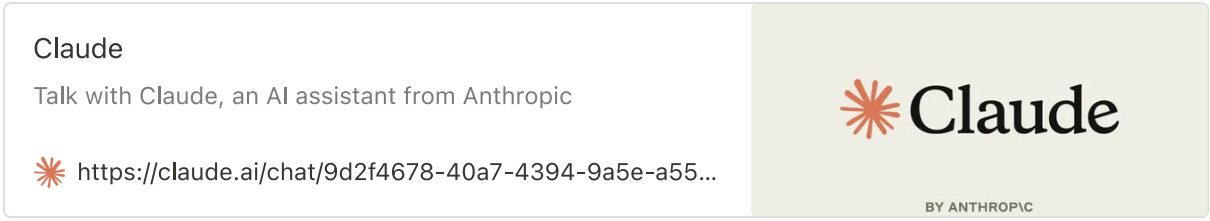
1. 数据集：在M4基准测试集上评估TimeRAG，该数据集包含来自金融、人口统计、营销等不同领域的数据，具有不同的序列采样频率（年度、季度、月度、周度、日度和小时）。
2. 评估指标：采用对称平均绝对百分比误差（SMAPE）、平均绝对尺度误差（MASE）和总体加权平均（OWA）三个指标。
3. 基线模型：与多种最先进的时间序列模型进行比较，包括基于Transformer的方法（iTransformer、FEDformer、Pyraformer等）和其他有竞争力的模型（如Time-LLM、DLinear、TSMixer等）。
4. 训练设置：基于LIama3，最大训练50个轮次，使用8个A100 GPU，Adam优化器，SMAPE作为损失函数。为减轻过拟合，采用动态学习率调整和早停策略，最大学习率设为0.01。

5. 可能的改进

尽管TimeRAG框架在时间序列预测领域展现出显著性能，但仍存在几个可能的改进方向。首先，当前的实现使用K-means聚类和DTW计算相似度，可以探索更先进的聚类算法和相似度度量方法，如分层聚类或基于深度学习的相似度计算，以提高检索效率和准确性。其次，可以考虑融合多模态信息，不仅仅依赖历史时间序列数据，还可以整合外部因素如社会经济指标、天气条件或新闻事件等，进一步增强预测的上下文理解。第三，当前框架在DTW计算上可能面临计算复杂度挑战，尤其对于高频长序列数据，可以探索更高效的近似DTW算法或并行计算方案。最后，可以拓展TimeRAG的应用场景，例如处理多变量时间序列预测或异常检测等任务，进一步验证其在更广泛时间序列分析领域的有效性。

VIII. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting

🔗 arxiv.org



1. 有什么问题?

时间序列预测一直是机器学习领域的重要研究方向，而基于Transformer的预测模型近年来受到广泛关注。然而，论文指出当前基于Transformer的时间序列预测模型存在几个关键问题：首先，传统Transformer模型在处理长序列时会出现性能下降和计算爆炸的问题；其次，现有模型通常将同一时间戳的多个变量嵌入为单个时间令牌(temporal token)，这种做法忽略了变量之间的独立性和关联性，可能导致学习到的表征缺乏变量中心性(variate-centric)，最终导致注意力图(attention map)失去意义。此外，当前研究发现简单的线性层模型在某些时间序列预测任务上甚至超过了复杂的Transformer模型，这引发了对Transformer架构在时间序列预测中适用性的质疑。

2. 提出了什么解决方案?

论文提出了一种创新性的解决方案：iTransformer（Inverted Transformer），它通过重新思考Transformer组件的功能并重新设计架构来解决上述问题。iTransformer最关键的创新在于"维度反转"的思想：不再将同一时间戳的多变量嵌入为时间令牌，而是将单个变量的整个时间序列嵌入为"变量令牌"(variate token)。在这种架构下，注意力机制被用于捕获多变量之间的相关性，而前馈网络则用于学习每个变量的非线性表征。这种设计使模型能够更好地处理变量间的复杂关系，同时保持各变量的独立性。论文强调，问题不在于Transformer本身不适用于时间序列预测，而在于之前的应用方式不恰当。

3. 论文有什么亮点

iTransformer的主要亮点在于其简洁而有效的设计思路。首先，它没有修改Transformer的任何基本组件，而是通过重新思考这些组件的适用维度，实现了显著的性能提升。其次，实验表明iTransformer在多个真实世界数据集上达到了最先进的性能，特别是在处理高维多变量时间序列时表现卓越。第三，通过将注意力机制用于变量间相关性建模，模型产生了更具可解释性的注意力图，能够直观地反映变量间的关系。另一个重要亮点是iTransformer展示了优秀的泛化能力：它能够在仅用少部分变量训练的情况下预测全部变量，这在实际应用中非常有价值。此外，与传统Transformer不同，iTransformer能够随着回溯窗口(lookback window)长度的增加而持续获得性能提升，这与统计预测方法的理论相符。

4. 论文具体是怎么实现的

iTransformer的实现主要围绕三个关键组件：嵌入层、注意力机制和前馈网络，但它们的应用维度与传统Transformer截然不同。

4.1 架构概述

iTransformer采用编码器架构，不使用解码器部分。对于输入的多变量时间序列$X \in R^{(T\times N)}$（T个时间步长，N个变量），iTransformer首先将其转置为$X \in R^{(N\times T)}$，然后对每个变量的时间序列单独进行嵌入，得到N个变量令牌，每个令牌维度为D，形成表征$H^0 \in R^{(N\times D)}$。

预测过程可以表示为：

1. $h^0_n = Embedding(X:,n)$：将每个变量的时间序列嵌入为一个令牌
2. $H^{(l+1)} = TrmBlock(H^l)$：通过L个Transformer块处理这些令牌
3. $\hat{Y}:,n = Projection(h^L_n)$：将最终表征投影为预测值

其中，Embedding和Projection都由多层感知机(MLP)实现。

4.2 层归一化(Layer Normalization)

在iTransformer中，层归一化应用于每个变量令牌的特征维度，而不是传统Transformer中应用于时间步的多变量表征。这种设计有两个主要好处：首先，它有助于解决非平稳(non-stationary)问题；其次，由于所有变量序列都被归一化为高斯分布，不同物理测量单位导致的差异被减少。相比之下，传统方法中对不同时间步的归一化可能导致时间序列过度平滑。

4.3 前馈网络(Feed-Forward Network)

在iTransformer中，前馈网络应用于每个变量令牌的序列表征上。通过万能近似定理(Universal Approximation Theorem)，这些网络能够提取复杂的时间序列表征。随着反转块的堆叠，它们的任务是编码观察到的时间序列并解码表征以预测未来序列。

作者对前馈网络学习的表征提供了一个有理解释：网络中的神经元被训练为描述任何时间序列的内在属性（如振幅、周期性甚至频谱特性），每个神经元相当于一个过滤器。这种解释与最近重新审视线性预测器的发现一致，即由MLP提取的时间特征应该在不同时间序列间共享。这解释了为什么iTransformer能够在看不见的变量上表现良好。

4.4 自注意力机制(Self-Attention)

在iTransformer中，自注意力机制应用于变量令牌之间，而不是时间步。具体来说，对于表征$H = \{h_1, ..., h_n\} \in R^{(N\times D)}$，线性投影生成查询、键和值$Q, K, V \in R^{(N\times d_k)}$。