RAG2.0 — 融合图、推理与决策的知识增强新范式

王昊奋

同济大学特聘研究员、OpenKG 发起人之一

2025.05.16

Part I: RAG概述与回顾

01. RAG发展回顾

- **1-1. RAG概述**
  检索增强生成的基本原理与关键技术组件
- **1-2. RAG Challenges回顾**
  去年ADL提出RAG技术面临的挑战，今年我们继续探讨

02. 2025我们还需要RAG吗?

- **2-1. RAG的Scaling Law**
  RAG中是否也存在Scaling Law?
- **2-2. RAG vs 长上下文**
  当上下文长度超过一百万时，还需要RAG吗?

03. 过去一年有什么进展?

- **3-1. 框架与工具新趋势**
  2025年RAG框架与工具有哪些新趋势?
- **3-2. 去年行业重大进展**
  从去年ADL到今天，社区有哪些重大进展?

Part II: RAG技术新趋势

04. RAG与Graph更深度的融合

- **4-1. 知识表示与关系建模**
  用图结构来增强RAG中知识的表示和关系的建模
- **4-2. 图推理能力增强**
  图上的多跳推理能力如何增强RAG?
- **4-3. 个性化GraphRAG**
  如何用图来增强RAG的个性化生成?

05. RAG与深度推理能力协同

- **5-1. RAG+Reasoning的新趋势**
  为什么RAG需要推理?两者协同有什么好处?
- **5-2. 协同的实现**
  RAG+Reasoning的实现方式有哪些?
- **5-3. 协同的优化**
  如何优化RAG+Reasoning的性能?

06. AgenticRAG流程设计

- **6-1. 流程定义**
  基于预设模板的AgenticRAG流程设计
- **6-2. 动态流程**
  完全动态的AgenticRAG流程设计

Part III: RAG前沿与实践

07. 评测的新趋势

- **7-1. RAG评测现状**
  RAG评测的现状是什么?
- **7-2. OpenKG OneEval**
  OpenKG推出的大语言模型知识评测基准
- **7-3. AGI-Eval**
  学术界主导的LLM评测社区

08. RAG应用实践

- **8-1. ToC 应用场景**
  面向个人的典型场景有哪些?
- **8-2. 领域应用场景**
  领域应用典型场景有哪些?
- **8-3. 隐性成本与实践指南**
  RAG的隐性成本有哪些? 实际应用中应该如何设计?

09. 总结与展望

- **9-1. RAG发展总结**
- **9-2. 未来的展望**

ADL158《AI搜索与信息智能体》

01 RAG发展回顾

1-1. RAG的概述 | 1-2. RAG Challenges回顾

知识检索增强技术（Retrieval-Augmented Generation, RAG）

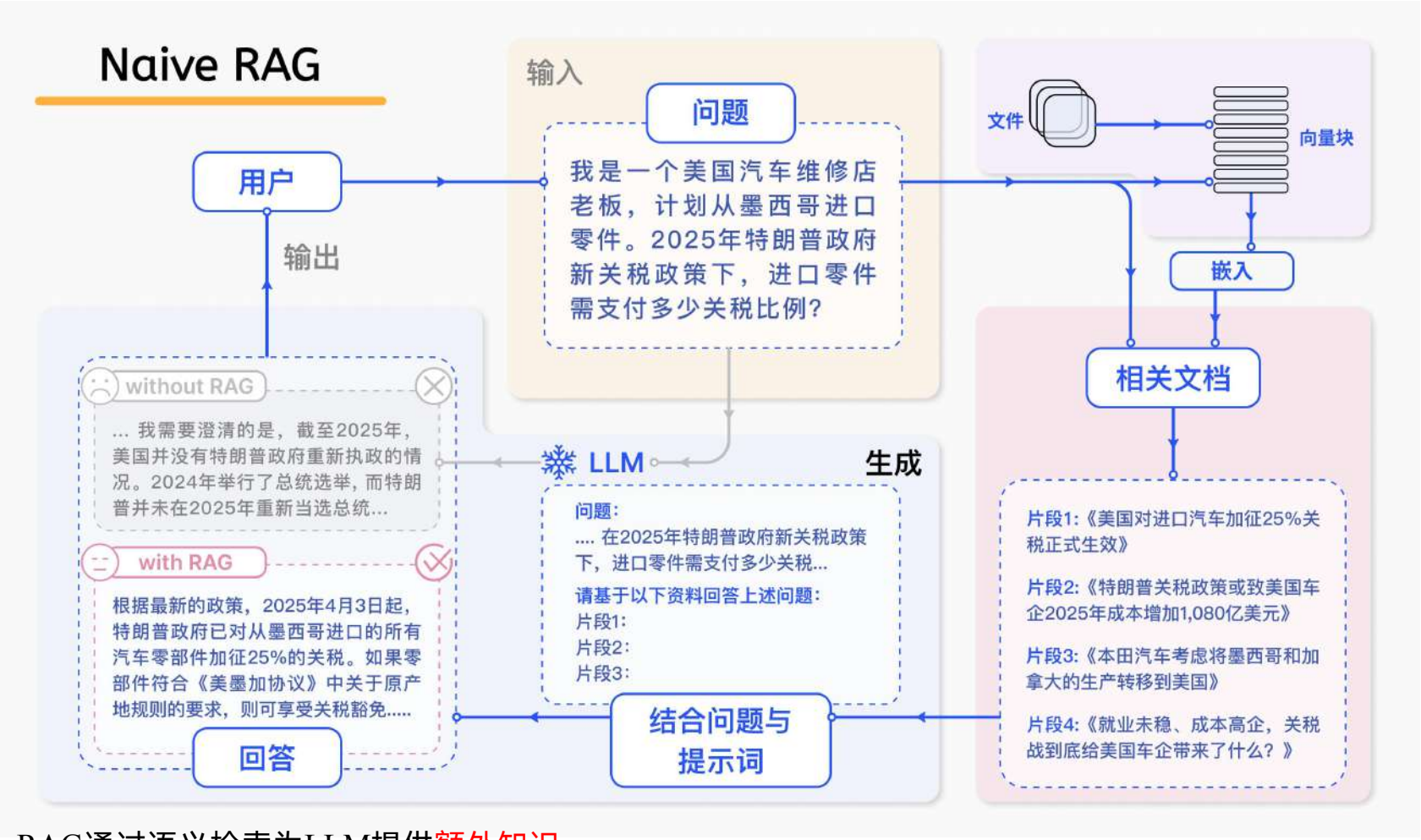
LLM的缺陷

- 幻觉
- 信息过时
- 参数化知识效率低
- 缺乏专业领域的深度知识
- 推理能力弱

实际应用的需求

- 领域精准问答
- 数据频繁更新
- 生成内容可解释可溯源
- 成本可控
- 数据隐私保护

大模型受语料限制，无法跟进最新事件。
例如，在不依赖外部知识库的情况下，LLM不知道特朗普执政期间的情况。



RAG通过语义检索为LLM提供额外知识。
在案例中，检索到特朗普2025年新的关税政策，新增25%的232条款，并提到美加墨贸易协定可能豁免。但答案仍有不足，如未提及基础税率，未明确豁免的具体要求，也未区分汽车零部件税率。
仅靠简单RAG无法满足复杂问题需求，我们需要的是决策辅助者，而非仅是知识助手。

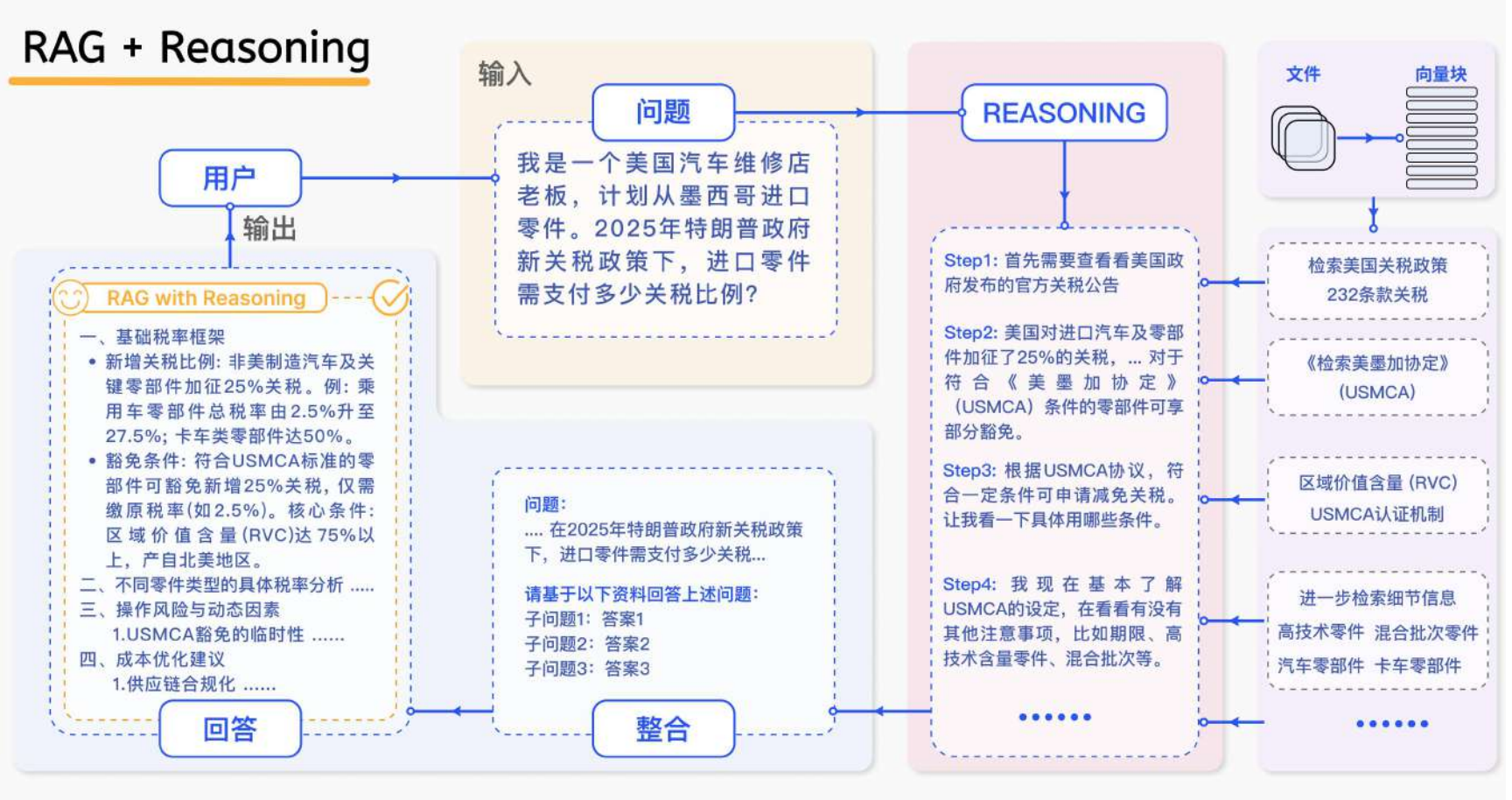
只用朴素RAG 就够了吗？

朴素RAG面临的缺陷

- 意图理解弱
- 复杂推理（多跳）能力差
- 信息覆盖度不足
- 决策链路不透明

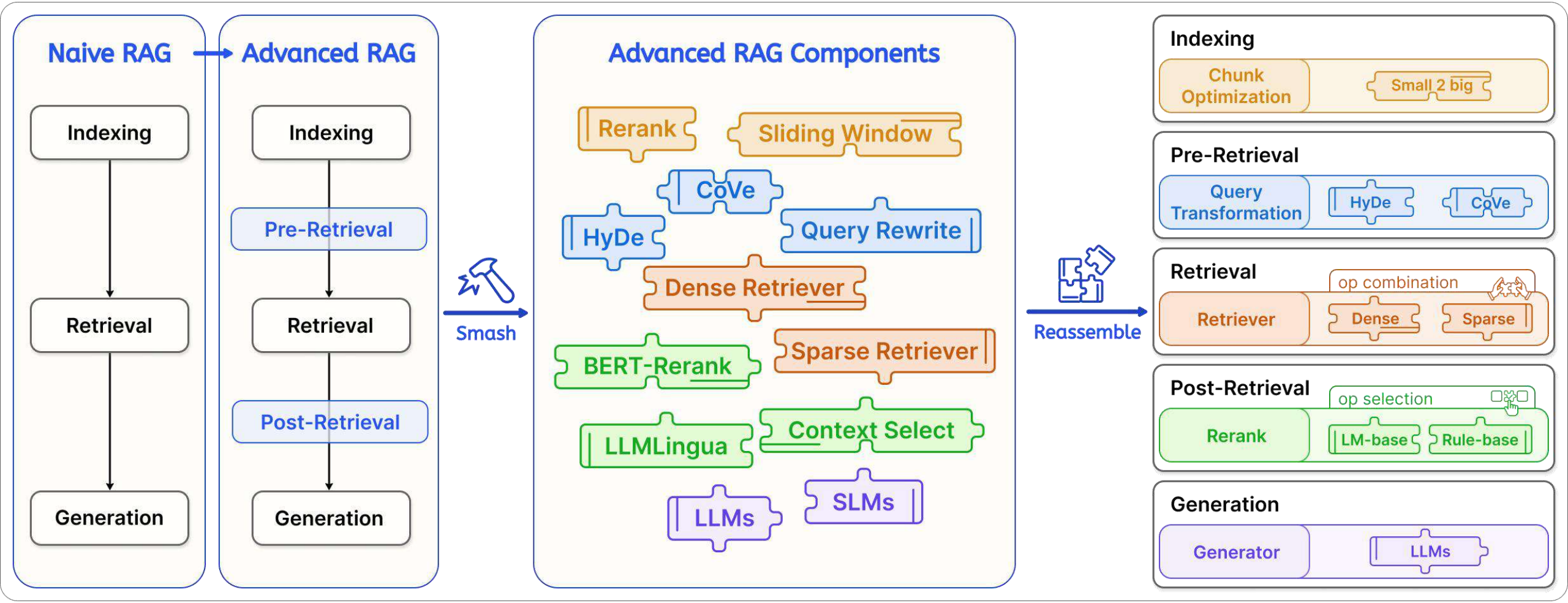
实际场景提出更高的要求

- 逻辑驱动的精准检索
- 逻辑自洽的上下文构建
- 智能资源分配
- 系统性决策支持
- 主动认知与服务

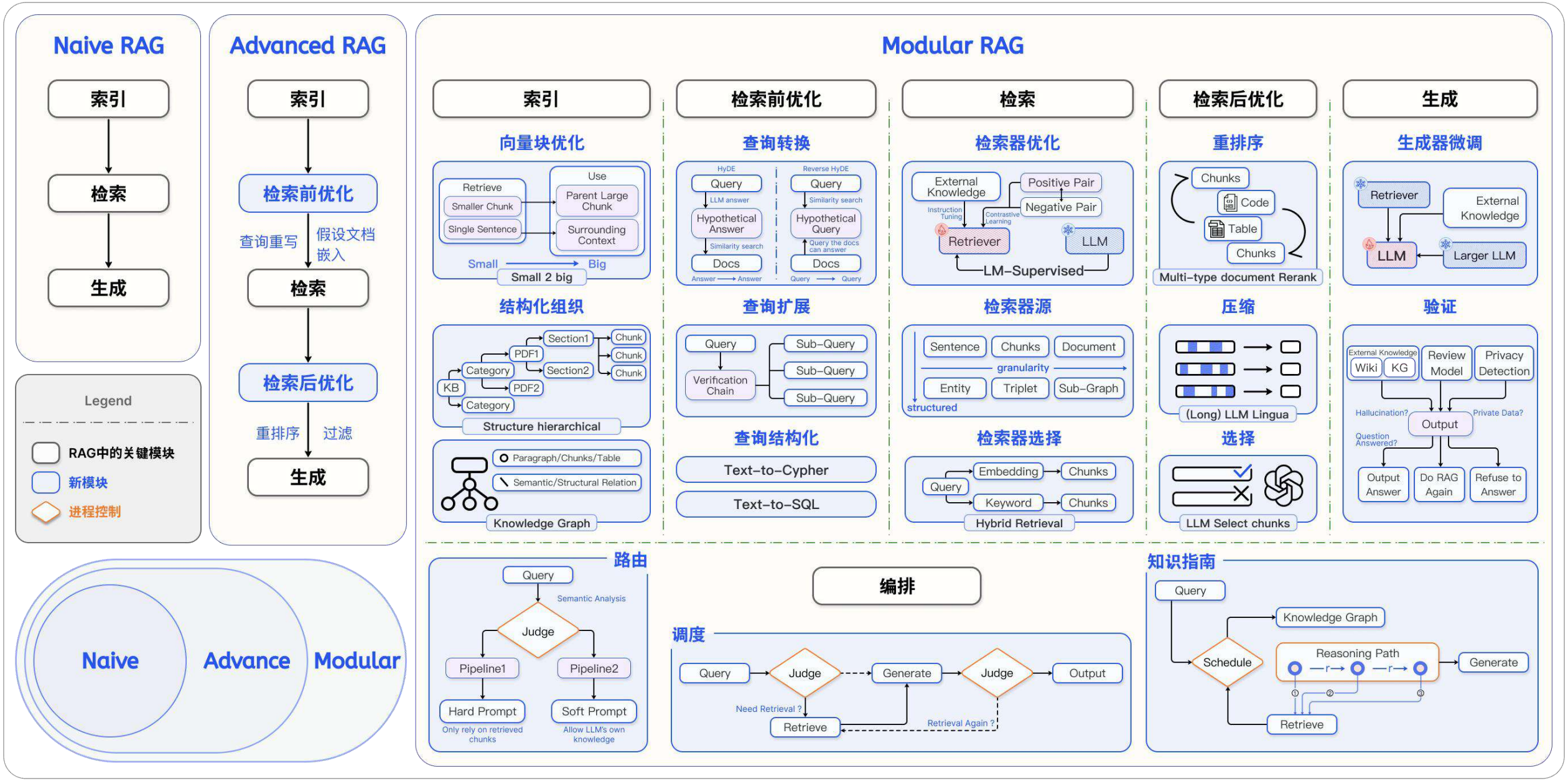


在该案例中，通过多步推理，将复杂问题分解后分别检索。与传统RAG直接堆砌答案不同，RAG+Reasoning通过任务拆分、检索、理解，最终形成逻辑自洽的推理链路。最终答案更细致，包含基础税率、汽车和卡车零部件税率差异、USMCA豁免条件、风险分析和优化建议，是一个较满意的回答。

RAG 范式发展趋势


Naive RAG
Indexing
Retrieval
Generation
Advanced RAG
Indexing
Pre-Retrieval
Retrieval
Post-Retrieval
Generation
Smash
Advanced RAG Components
Rerank
Sliding Window
CoVe
HyDe
Query Rewrite
Dense Retriever
BERT-Rerank
Sparse Retriever
LLMLingua
Context Select
LLMs
SLMs
Reassemble
Indexing
Chunk Optimization
Small 2 big
Pre-Retrieval
Query Transformation
HyDe
CoVe
Retrieval
op combination
Retriever
Dense
Sparse
Post-Retrieval
op selection
Rerank
LM-base
Rule-base
Generation
Generator
LLMs

RAG 的典型范式演化（Modular RAG）



RAG 的典型范式演化（Modular RAG）

Modular RAG

三层架构

Module

6 大主要模块：
RAG的核心流程

Sub-Module

14 个子模块：
流程中的具体功能

Operator

40+ 算子：
特定功能的具体实现

统一的RAG研究范式

对之前范式的继承与发展

RAG Flow

不同模块以及模块中不同算子的选择和编排构成了RAG Flow，从而识别出典型的RAG Flow 模式。

3 Tuning Stage

- Retriever FT
- Generator FT
- Dual FT

4 Inference Stage

- Sequential
- Conditional
- Brach
- Loop

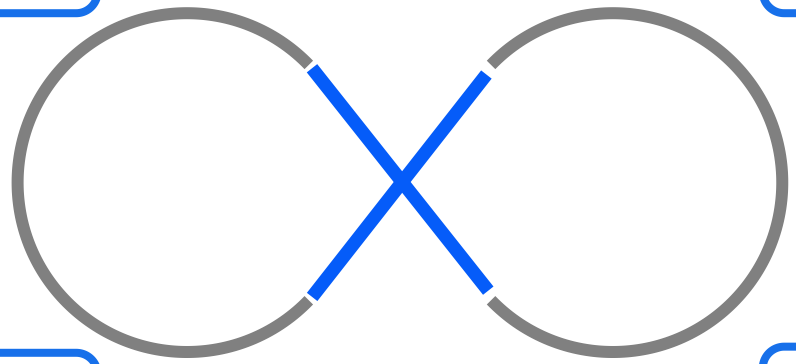
模块化RAG下的机遇

研究视角

新模块

新算子

新编排方式



场景适配

性能优化

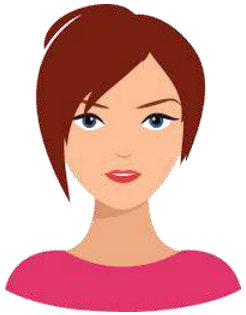
应用扩展

应用视角

AI (RAG) 研究者

算法工程师

业务应用开发工程师



业务专家

01 RAG发展回顾

1-1. RAG的概述 | 1-2. RAG Challenges回顾

去年在ADL中，我们围绕RAG的发展现状与趋势，提出了面向社区的十大挑战及相应技术路线。

时隔一年，今天，我们一起回顾技术演进的脉络

2024.5.24 ADL147
《大模型与检索》

《知识检索增强：
范式与关键技术》

过去一年有哪些技术进展?

✓
有哪些已经取得了突破?
部分关键技术路线得到验证

△
哪些领域还有待解决?
针对性改进取得进展

?
哪些是新出现的挑战?
仍需深入研究

RAG的十大挑战（2024）

- 应用侧
  - 大规模工业应用场景
  - 低资源任务场景
- 技术侧
  - 幻觉控制
  - 隐私保护
  - 自主控制的RAG流程
  - 流程可追溯、结果可解释
  - 查询与知识的语义距离
- 数据侧
  - 多模态数据检索与理解
  - 长文本的高效切分、向量化与检索
  - 大规模知识库的管理
  - 异构数据的整合与存储
  - 评测集与指标
- 理论侧
  - 外部知识和内化知识的博弈
  - RAG记忆与遗忘机理

值得进一步研究的RAG议题（2024）

应用侧

- RAG增强的推荐系统
- RAG增强的个人助手
- RAG增强的代码服务
- RAG增强的决策支持

系统侧

- RAG开源模型库 Huggingface
- RAG评测平台
- AutoRAG
- 无/低代码平台

方法侧

RAG 模型与协作	RAG 与上下文	RAG 流程	RAG Safety	低资源 RAG
RAG + XOE	Context Compression	自适应检索	RAG 隐私保护	端云协同 RAG
RAG + KG	Context Selection	RAG 事实校验	RAG 越狱	端侧 RAG
RAG + 微调 /RLHF	Context 传输优化	反思与拒答	RAG 可控生成	
RAG + SLM	长下文下的 RAG	RAG 鲁棒性		

数据侧

- RAG 评估体系
- 结构化检索源
- 分块策略与索引配置

理论侧

- RAG Scaling Law
- RAG 的记忆与遗忘机制

技术路线的验证

接下来，我们将从理论、方法和系统三个维度，选取最具代表性的一个方向进行深入探讨。

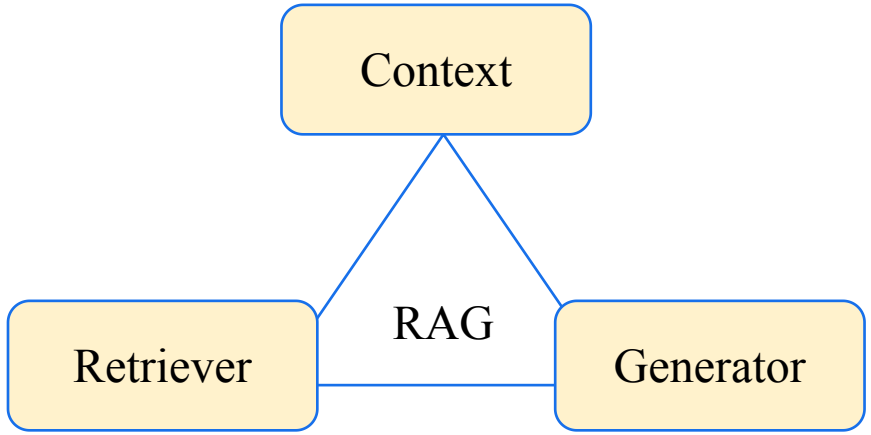
02 2025我们还需要RAG吗？

2-1. RAG Scaling Law | 2-2. RAG vs 长上下文

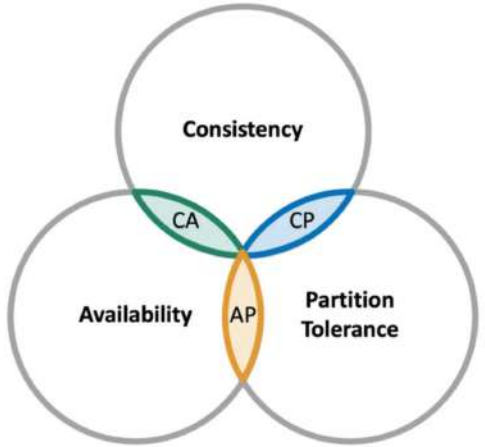
理论侧：Scaling Law

RAG中是否也存在Scaling Law?

- 在给定LLM规模的情况下，如何选择Context size和chunk size指导RAG（模型资源受限）？
- 在Context类型确定的情况下，如何搭配Embedding和LLM？（检索源受限）？
- 如何组合与搭配检索器和生成器，从而实现更好的效果？
- RAG中是否存在能力的上限，或是类CAP理论？



RAG中的三角关系



大模型中的CAP理论

长上下文检索增强生成的推理扩展

Inference Scaling for Long-Context RAG

基于大型语言模型的推理计算优化

作者：Zhenrui Yue, Honglei Zhuang等 (2025年3月)

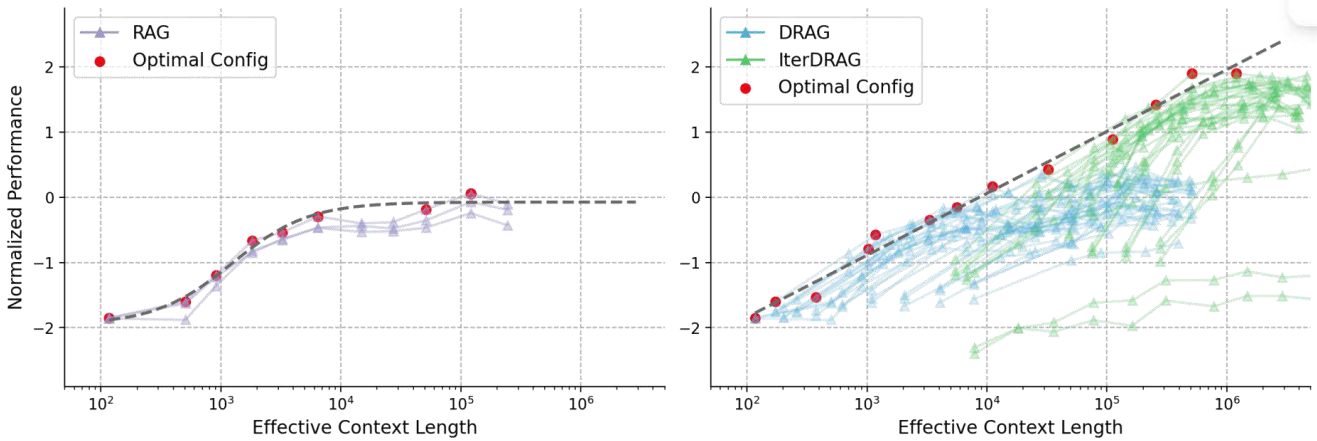
研究背景与核心问题

当前长上下文RAG的挑战

- 简单增加检索文档数量效果有限，性能很快达到平台期
- 现有大型语言模型难以有效处理超长上下文
- 仅增加知识量而不提供引导会导致性能下降
- ⚠ 合理分配推理计算资源成为关键问题

为什么推理扩展很重要?

推理计算的扩展释放了长上下文大语言模型的潜力，特别是对于知识密集型任务，增加外部知识与计算资源的合理分配能显著提升模型性能。



研究的核心问题

1. **RAG性能如何从推理计算扩展中获益?**
   研究增加计算资源对性能的影响规律

2. **能否预测给定计算预算下的最佳参数配置?**
   寻找最优资源分配策略

研究方向

计算资源分配

- 示例数量与检索数量平衡
- 迭代深度与广度的权衡

性能预测模型

- 参数敏感性分析
- 不同模型大小的扩展规律

推理扩展策略与实验结果

提出的创新策略

DRAG (示范型检索增强生成)

利用上下文学习能力，将多个示例与检索文档结合，帮助模型更好地定位和利用相关信息

IterDRAG (迭代示范型检索增强生成)

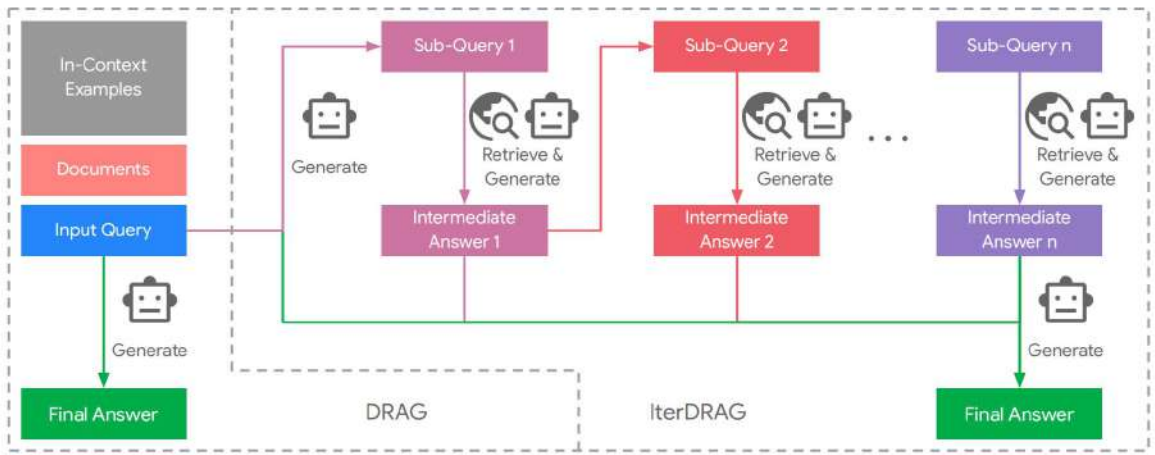
通过子查询分解、交错检索与迭代生成，构建推理链，弥合多跳查询的组合鸿沟

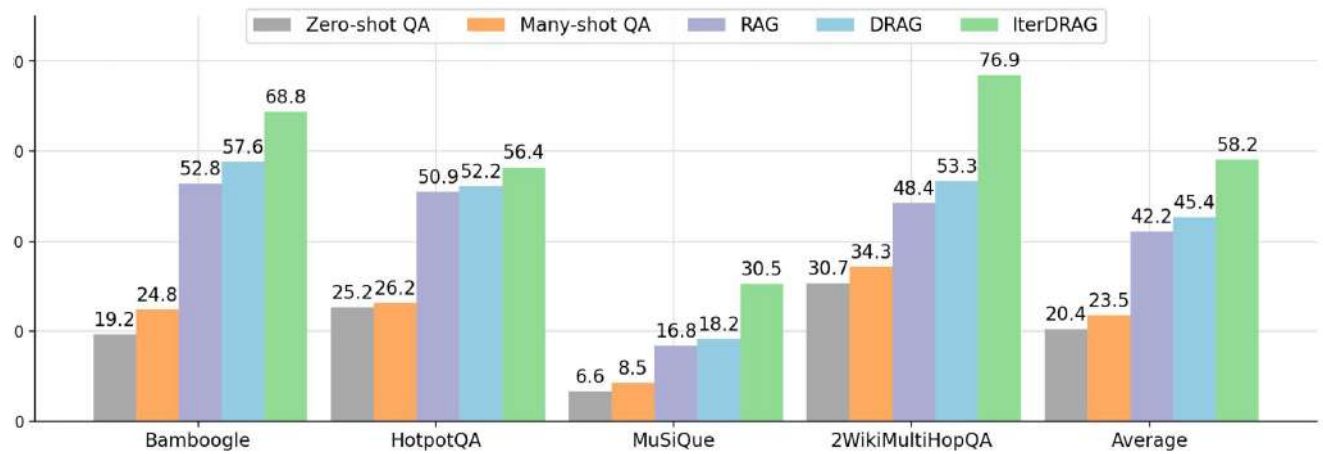
主要实验结果

发现1: 在计算资源最优分配下，RAG性能几乎与推理计算的增加呈线性增长

发现2: DRAG在较短上下文中表现更佳，而IterDRAG在超长上下文（>128k）中性能更优

发现3: 对比标准RAG，我们的方法在基准数据集上取得高达58.9%的性能提升





使用万亿级令牌数据库扩展检索式语言模型

Scaling Retrieval-Based Language Models with a Trillion-Token Datastore

华盛顿大学 & Allen 人工智能研究所

Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, Pang Wei Koh

arXiv:2407.12854v1 [cs.CL] 9 Jul 2024

项目链接: https://github.com/RulinShao/retrieval-scaling

研究背景与核心问题

研究背景

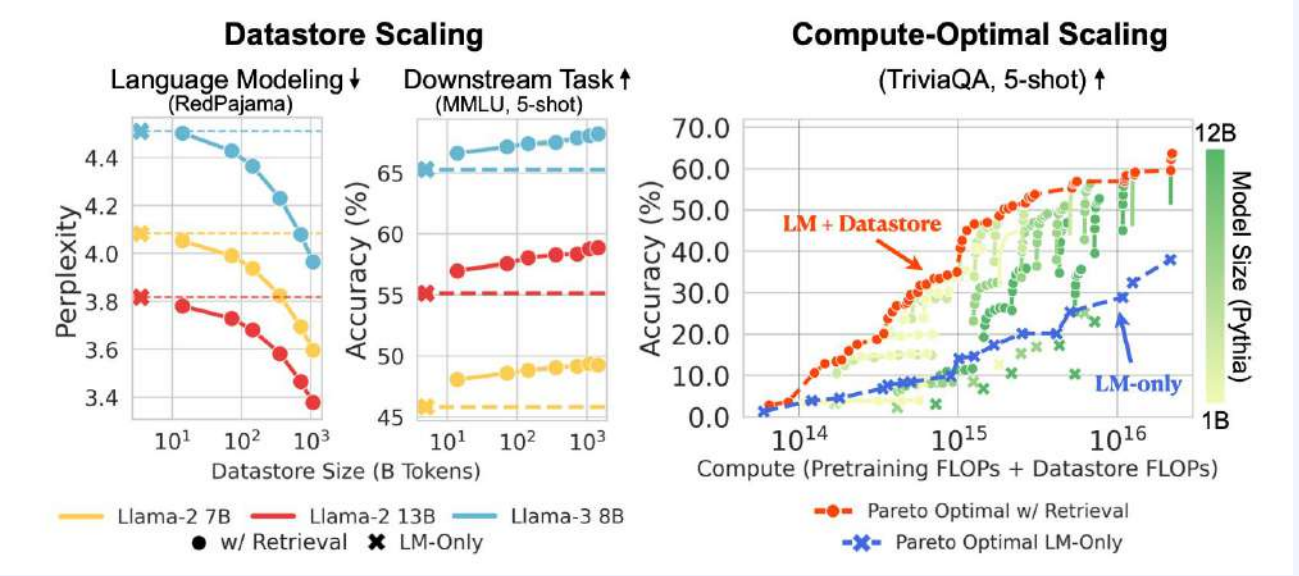
- 传统语言模型扩展关注两个维度：训练数据量和模型参数数量
- 扩展法则允许预测不同配置下预训练语言模型的成本效益权衡
- 当前研究缺乏对推理时可用数据量这一维度的系统性探索

核心问题

- 检索式语言模型的数据库规模如何影响模型性能?
- 增大数据库规模能否作为一种替代扩大模型规模的方法?
- 在计算资源有限的情况下，如何最优地配置数据库和模型规模?

研究发现

- 增加推理时可用的数据库规模可单调提升语言建模和下游任务性能
- 小型模型配合大规模数据库可超越大型纯语言模型在知识密集型任务上的表现
- 数据库规模应被视为语言模型效率和性能权衡的重要维度



研究结论

主要发现

- 增加推理时可用的数据量可单调改善语言模型性能，没有明显饱和迹象
- 配备大规模数据库的小型模型可超越大型纯语言模型在多种任务上的表现
- 通过计算最优扩展曲线证明，数据库规模是一个重要的扩展维度，可提供更高的计算效率
- 检索质量的提升和数据过滤可以进一步增强数据库扩展的效果

局限与未来方向

- 现有实验受计算资源限制，数据库和模型规模的组合未能完全探索
- MASSIVEDS虽然规模庞大，但可能仍缺乏改善复杂推理任务所需的高质量数据
- 当前评估主要集中在问答任务上，未来需要扩展到更多样化的任务类型
- 改进检索过程（如更好的检索器或重排器）可能带来显著性能提升
- 探索更多数据源和数据类型以提高数据库质量

总结与影响

本研究表明，数据库规模应被视为语言模型效率和性能权衡的关键维度。通过构建和开源MASSIVEDS数据库，为未来检索式语言模型的研究提供了宝贵资源。

研究结果对构建更高效、性能更优的语言模型系统具有重要指导意义，特别是在计算资源有限的情况下。

数据库规模是语言模型系统设计的关键考量因素

扩大推理时可用数据是一种提高模型性能的有效方法，应与模型规模和训练数据量一起考虑

代码和数据库开源地址：https://github.com/RulinShao/retrieval-scaling

02 2025我们还需要RAG吗？

2-1. RAG Scaling Law | 2-2. RAG vs 长上下文

RAG vs Long Context

长下文窗口突破100万 Token

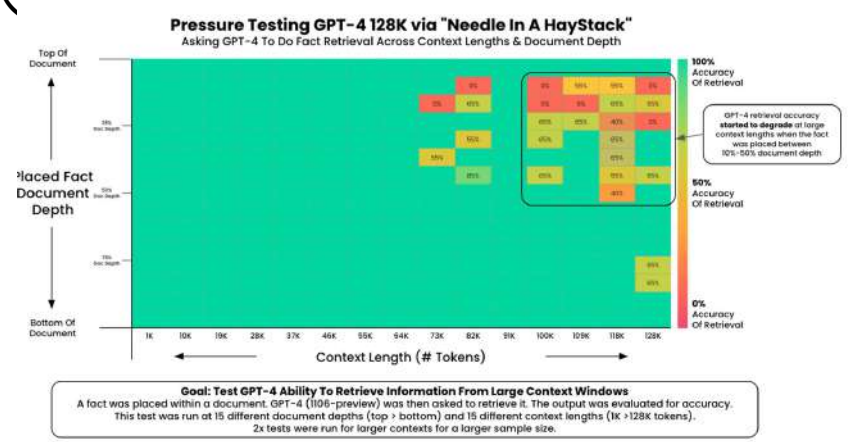
这意味着，可以直接给LLM输入

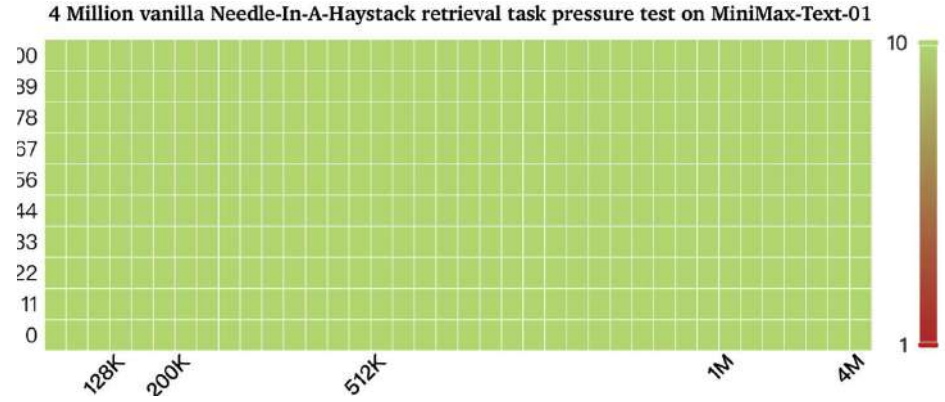
- 一整本小说
- 某公司十年的财报
- 几百个文档
- ....

Is RAG Really Dead ？

Needle In a Haystack 大海捞针

LLM在大海捞针任务上的表现不断提升





2023.8 GPT4 /128K → 2025.1 MiniMax-01 /4M

- RAG和LLM在Long-context中的表现如何？
- 在Long-context下RAG有什么优势？
- 在Long-context RAG有什么缺陷？典型的错误模式？

长上下文 vs. RAG: 评估与重新思考

RAG中的检索方法

基于分块的检索

- 将文档分割成较小的块
- 使用向量嵌入来找到相关块
- 流行模型: E5-Mistral, OpenAI embeddings
- 简单但可能错过上下文连接

基于索引的检索

- 使用更复杂的数据结构
- 工具如Llama-Index促进交互
- 树状或图索引等层次结构
- 更好地组织信息

基于摘要的检索

- 建立在分块和索引方法之上
- 创建内容的层次摘要
- 例如: RAPTOR系统使用递归摘要
- 更适合多步推理任务

检索方法比较

方法	优势	劣势
基于分块	简单、高效、应用广泛	可能错过上下文连接
基于索引	更好的组织、灵活结构	设置更复杂、需要预处理
基于摘要	更适合复杂推理、提取关键点	计算密集、可能丢失细节

Long Context vs. RAG for *LLMs: An Evaluation and Revisits. Arxiv,2025*

长上下文 vs. RAG: 评估与重新思考

长上下文优势

- 在问答基准测试中总体上优于RAG
- 对维基百科类问题特别有效
- 对结构良好、信息密集的上下文更有效(维基文章、书籍)
- 在需要提取特定信息时表现更佳
- 保持完整文档上下文，理解更连贯

RAG优势

- 基于摘要的检索性能与LC相当
- 在对话和一般问题查询中表现更佳
- 更擅长处理碎片化信息
- 对超长文档(超出模型限制)更高效
- 通过仅关注相关部分，计算上可能更高效

需考虑的权衡

- 上下文相关性对性能至关重要
- 合成与真实上下文产生不同结果
- "长"上下文的定义在各研究中差异很大
- 计算资源随上下文长度增加而增加

检索方法比较

- **基于摘要:** 性能最佳，接近LC质量
- **基于索引:** 适合复杂结构，中等性能
- **基于分块:** 简单但落后于其他方法

先进检索方法的使用，RAG与LC之间的差距正在缩小

Long Context vs. RAG for *LLMs: An Evaluation and Revisits. Arxiv,2025*

U-NIAH

统一RAG与LLM长文本环境下的"大海捞针"评估框架

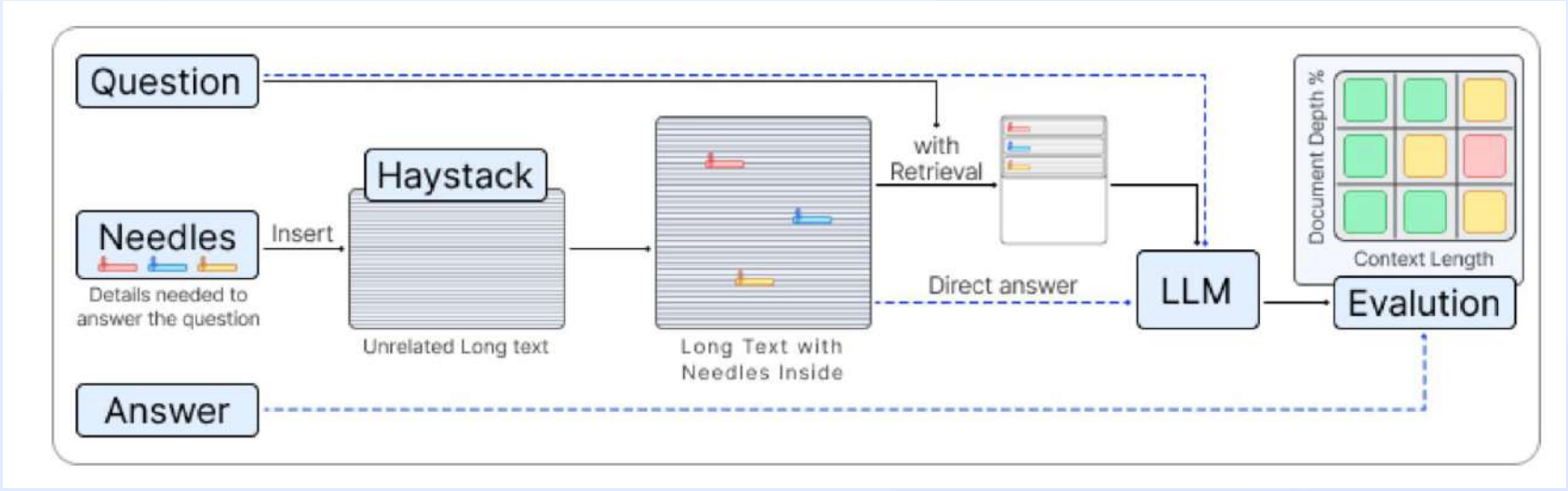
U-NIAH: Unified RAG and LLM Evaluation for Long Context Needle-In-A-Haystack

Yunfan Gao, Yun Xiong, Wenlong Wu, Zijing Huang, Bohan Li, Haofen Wang

同济大学 复旦大学 南京航空航天大学

arXiv:2503.00353

项目链接: https://github.com/Tongji-KGLLM/U-NIAH



U-NIAH框架

统一框架设计

U-NIAH整合RAG和LLM进行"大海捞针"评估，处理输入四元组:

Q: 查询问题

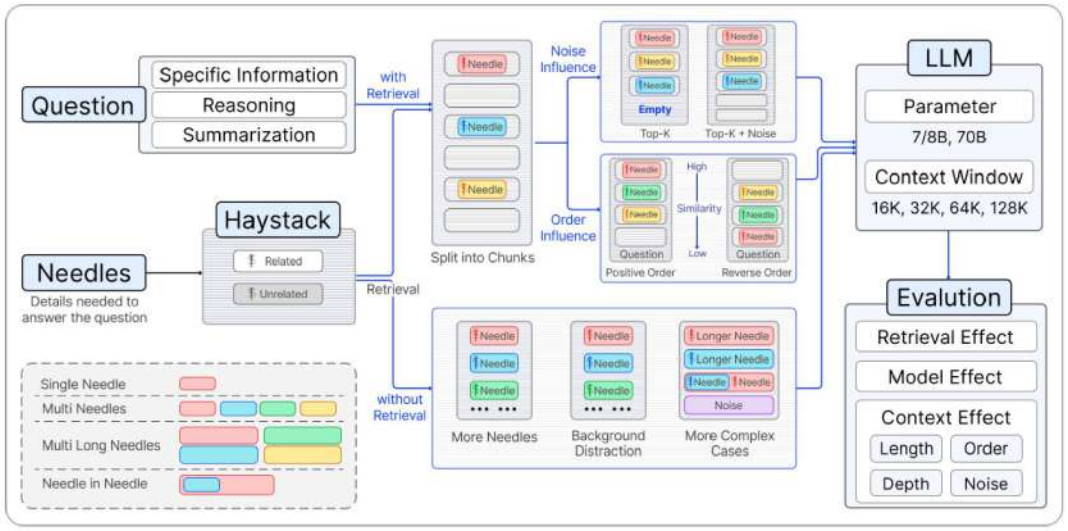
A: 标准答案

N = {n_1, ..., n_k}: 关键信息集合

H: 干扰信息(长文本语料)

上下文窗口大小分配:

L(总上下文窗口) = P(系统提示) + N(关键信息) + H(干扰信息)



创新设计

1. 多维度因素扩展:

- "针"(关键信息)数量: 3/7/15个
- "针"长度: 短(50–100 tokens)到长(400–500 tokens)
- "针中针"结构: 在长信息中嵌入短信息

2. 问题复杂度:

- 简单事实检索到复杂关系分析
- 跨信息关系分析与综合概括能力

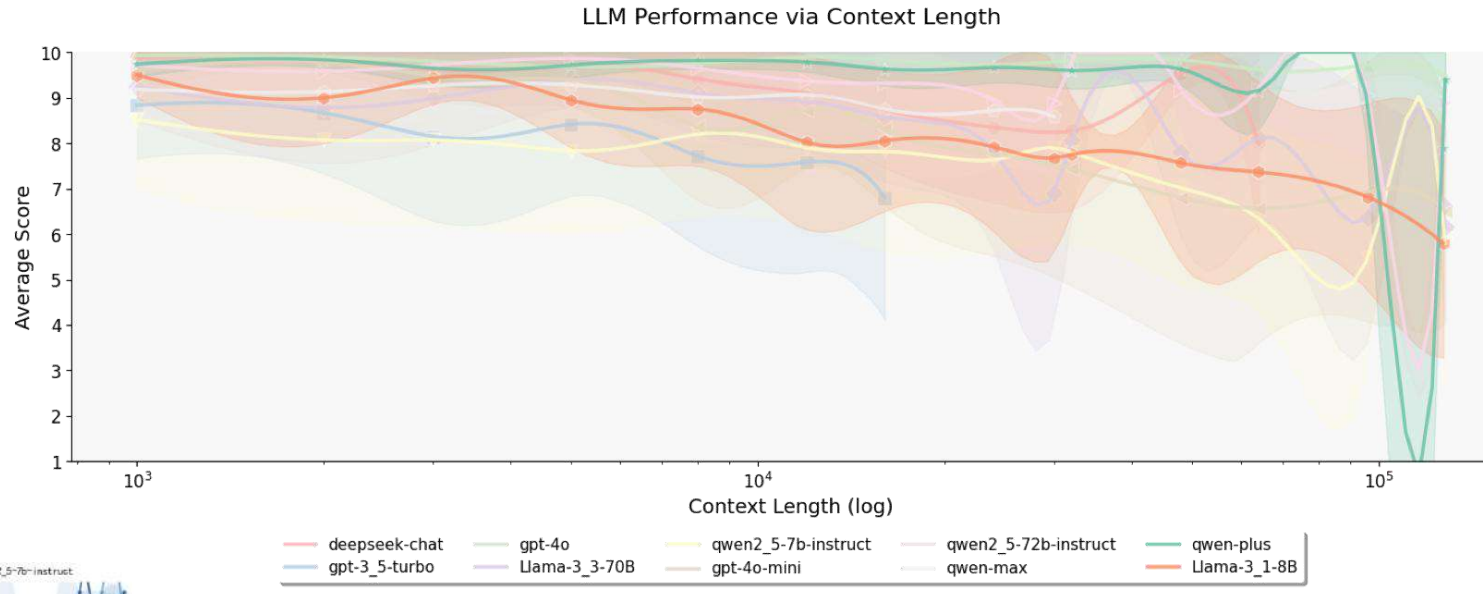
3. 干扰设计:

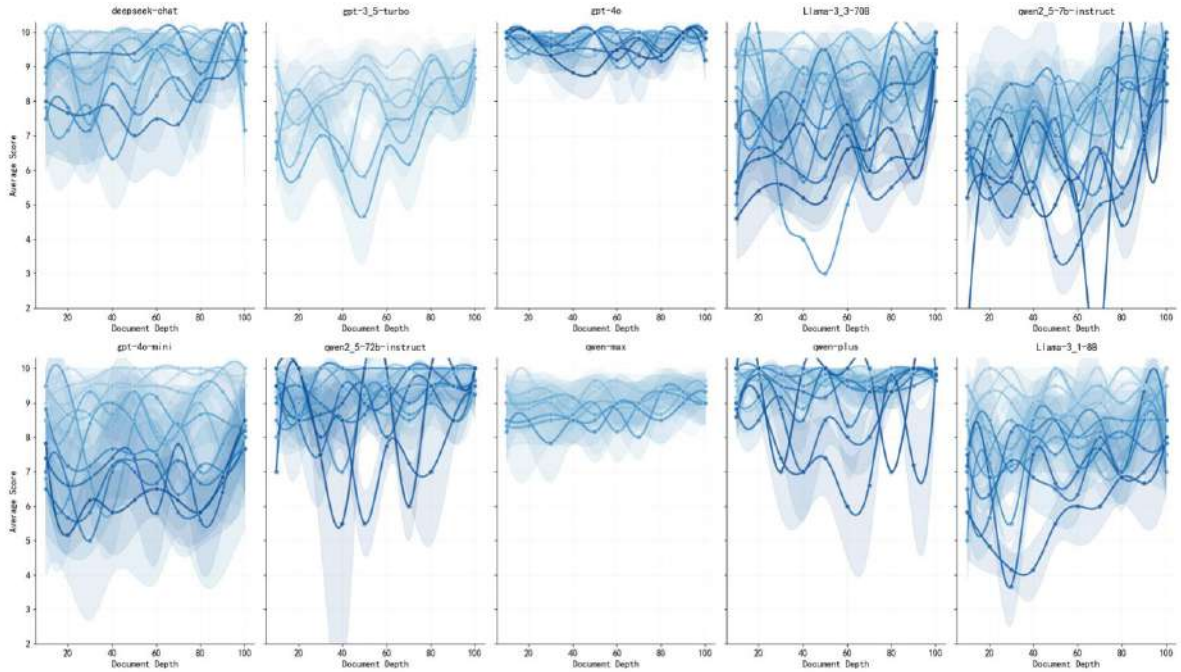
- 语义无关背景文本
- 对抗性干扰(与关键信息相似但不准确)
- 噪声比例控制: η = |Rnoise|/|R|

框架通过对比LLM直接处理与RAG检索增强的表现，在统一指标下评估两种方法的优劣

RAG与LLM在Long-context下的性能对比

- RAG总体优于LLM，胜率达**82.58%**
- RAG显著提升小型LLM性能，缓解"lost–in–the–middle"效应
- 随上下文长度增加，RAG优势更明显:
  - [1–16k]: 57.4%胜率
  - [16–32k]: 77%胜率
  - [32–64k]: 86.7%胜率
  - [64–128k]: 92.7%胜率
- 模型规模越大，RAG提升效果越小





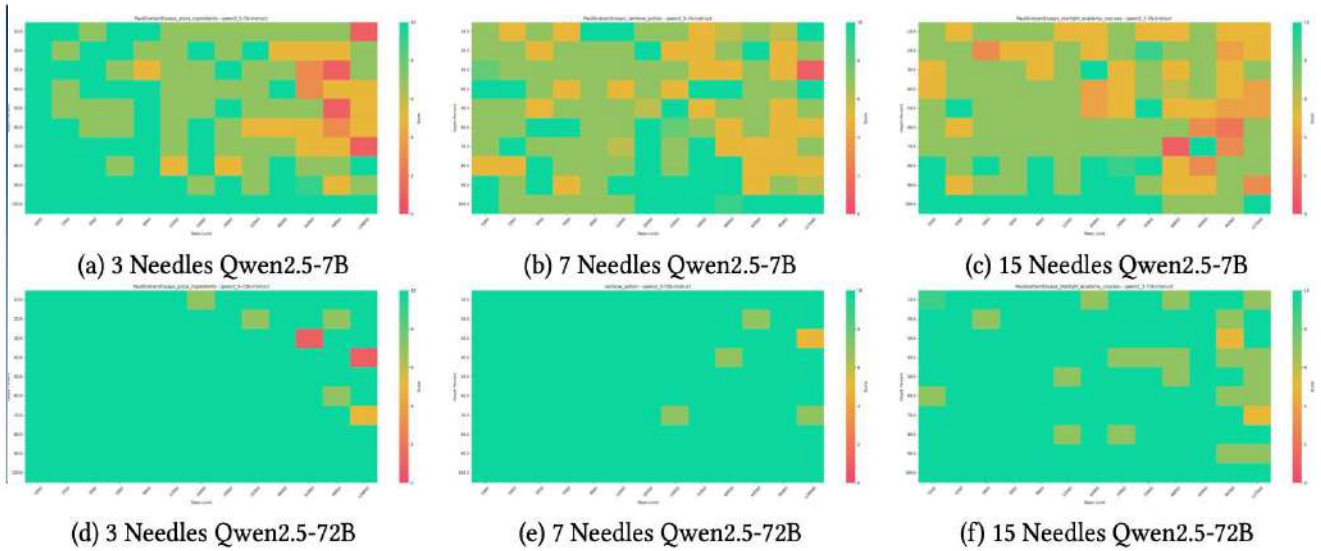
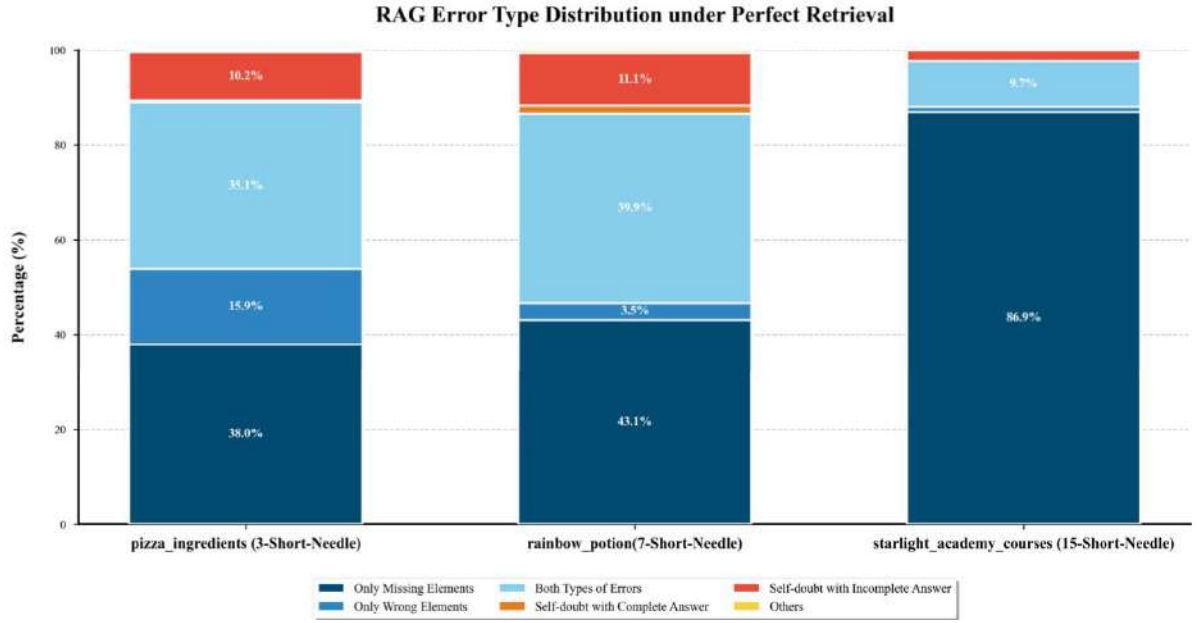
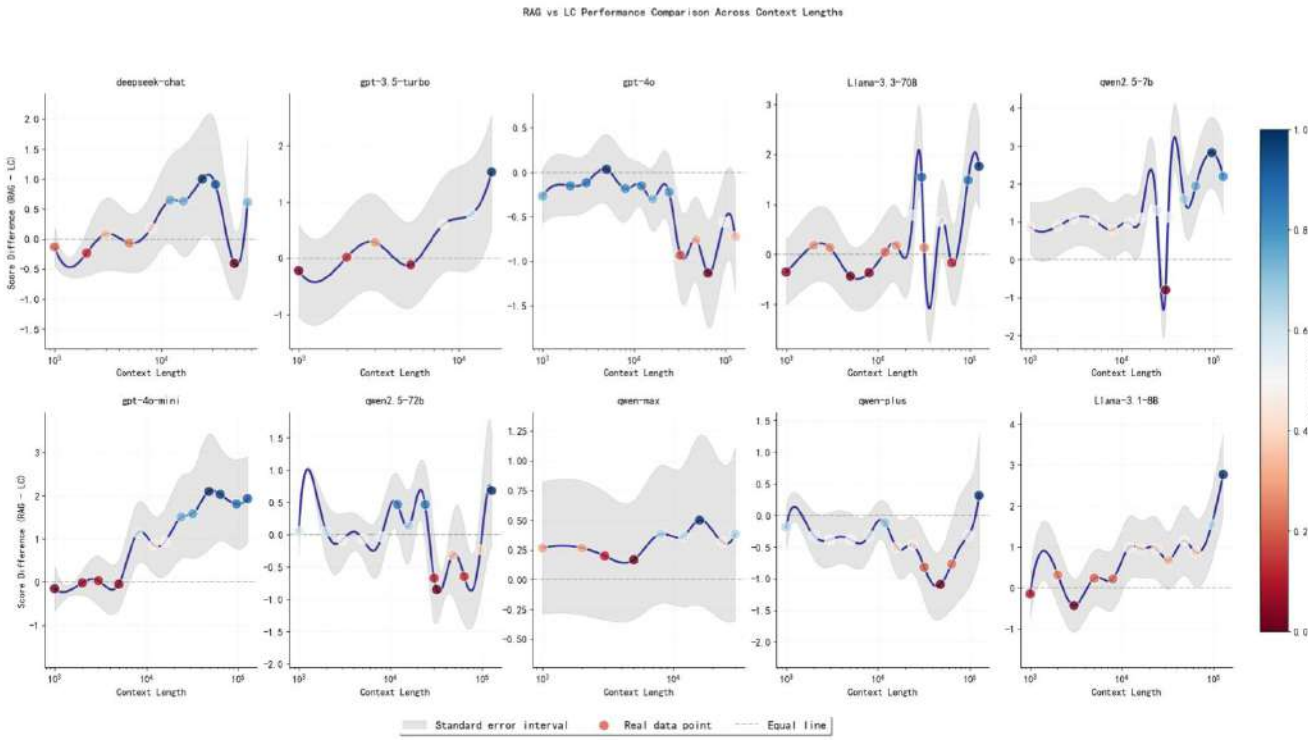


Figure 5. Performance of LLM in U-NIAH framework under different Document Depth

RAG在Long-context下的典型错误模式

- **检索瓶颈**: 召回率不足导致信息缺失(10.18%)
- **噪声干扰**: 噪声比例>44%时信息遗漏增加(+33.53%)
- **噪声临界**: 长上下文+噪声比>97%时幻觉激增(+355.82%)
- **位置敏感**: 逆序排列使信息遗漏增加(+59.64%)
- **自我一致性**: 小模型在长上下文中自我怀疑(11.30%)
- 小模型更容易受到各种错误影响





Error Pattern	Condition	Behav-ior	Effect	LC Impact	Model Size	Cause
Retrieval Bottleneck	Recall < 1	Missing Element	10.18%	Increase with longer LC	Independent	Limits the upper bound of generation
Noise Distraction	Noise Ratio > 44%	Missing Element	+33.53%	Increase with longer LC	Smaller more affected	High noise interferes with LLM's information capture
Noise Critical	Long Context + Noise Ratio > 97%	Extra Element	+355.82%	Occurs in long contexts	Smaller more affected	Hallucination surges with high noise at critical context length
Position-sensitive	Context Length < 16K + Reverse	Missing Element	+59.64%	Pronounced in small / medium contexts	Smaller more affected	Irrelevant info at the beginning interferes with understanding
Self-consistency	Small Model + Context > 16K + Reverse	Self-doubt	11.30%	Significant in longer LC	Smaller models 377% more	Incoherent text in long contexts causes self-consistency errors

长文本 Vs 嵌入模型

嵌入模型在长上下文中的大海捞针任务分析

《NoLiMa: Long-Context Evaluation Beyond Literal Matching》
arXiv:2502.05167

研究问题

1. 嵌入模型如何在不同上下文长度中处理大海捞针检索?
2. 语义查询增强能否缓解长上下文中的性能差距?

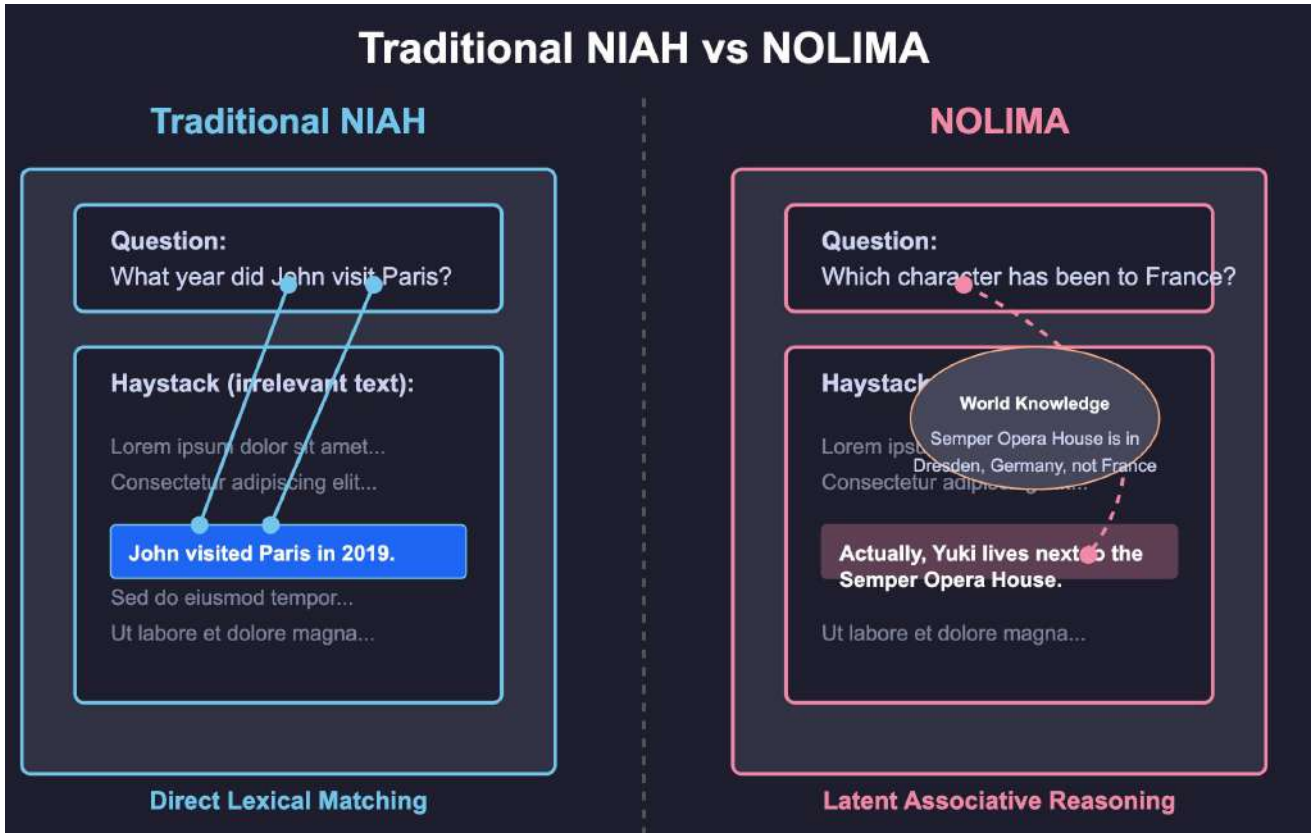
针和干草堆的构建

针（相关信息）构建方式：

问题示例："哪个角色去过德累斯顿？"
一跳针："事实上，Yuki 住在森帕歌剧院旁边。"
倒装一跳针："森帕歌剧院就在 Yuki 住的地方旁边。"
（森帕歌剧院位于德累斯顿）

干草堆（上下文文本）构建：

- 从公共领域书籍中随机连接短片段
- 不同长度：128至8192个token
- 在每个干草堆中嵌入一个针



例如，在传统的NIAH中，如果问题是“John什么时候访问巴黎？”，针可能直接包含“John在2019年访问了巴黎。”在NOLIMA中，问题可能是“哪个角色去过法国？”而针包含“事实上，Yuki住在森帕歌剧院旁边。”——这要求模型知道森帕歌剧院在德国德累斯顿，而不是法国。

评估指标

归一化相似度分数
测量问题与整个上下文之间的相似度，并与问题-针基线相似度进行归一化

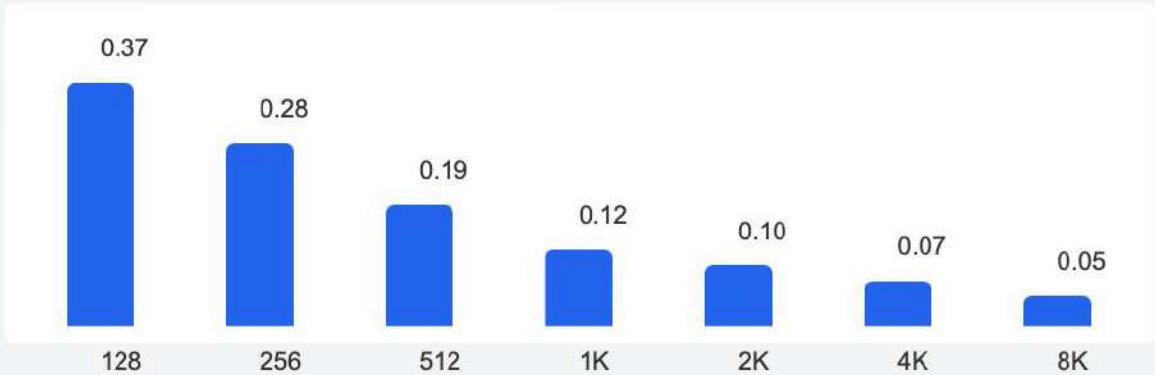
与随机概率的比较比率
问题与针所在干草堆的相似度超过随机片段相似度的频率

分离分析
模型区分相关和不相关内容的能力，包括平均分离度和AUC分数

关键研究发现

1. 性能随上下文长度急剧下降

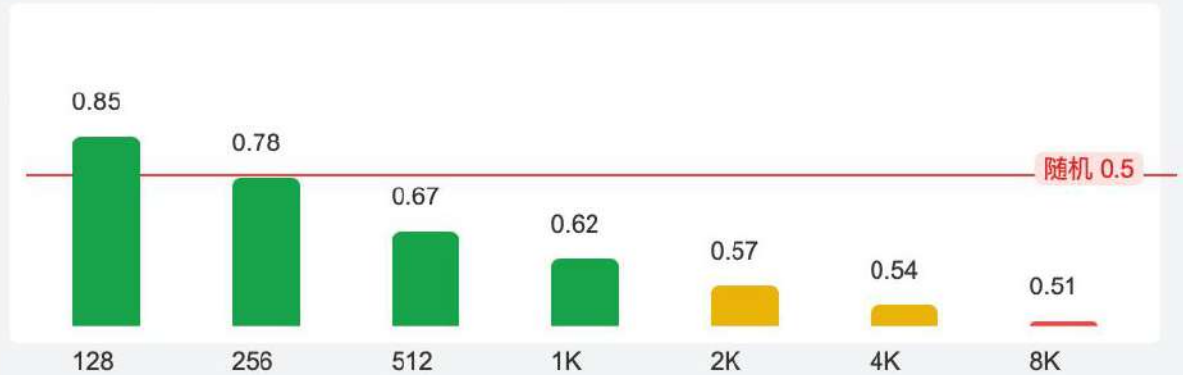
归一化性能与上下文长度关系



平均相似度分数从 128 token 时的 0.37 下降到 8K token 时的 0.10

2. 4K Token后几乎等同于随机猜测

模型性能与随机概率的比较



在8K token文本中，模型选择包含正确答案的能力接近随机猜测(0.51)

3. 分离分析显示严重的区分能力下降

短上下文 (128 token)
分离度: **0.1**
AUC: **0.81**
模型能可靠地区分相关和不相关段落

中等上下文 (1K token)
分离度: **0.04**
AUC: **0.66**
性能下降60%，判别能力减弱

长上下文 (8K token)
分离度: **0.001**
AUC: **0.50**
几乎无法区分，等同随机猜测

查询扩展影响与词汇匹配实验

查询扩展能否改善长上下文性能?

查询扩展示例:

原始查询: "哪个角色去过德累斯顿? "

扩展后: "哪个角色去过德累斯顿? 角色: 虚构角色 文学角色 主角 反派 人物 形象 角色 剧中人物... 德累斯顿: 德累斯顿德国 德累斯顿轰炸 二战 历史小说 冯内古特 屠宰场5号 萨克森城市 易北河 文化地标..."

查询扩展结论:

- 查询扩展有所帮助，但上下文长度仍是主要挑战
- 100个术语扩展优于150和250个术语（质量与数量的权衡）
- 即使是8K token文本，扩展查询也使找到正确答案的可能性高于随机

词汇匹配实验： 匹配也会失效

实验设计:

使用直接词汇匹配的针，测试长上下文性能

问题: "哪个角色去过德累斯顿? "
直接匹配针: "事实上，Yuki住在德累斯顿。"

实验发现:

即使存在直接词汇匹配，长上下文性能仍然急剧下降!

直接词汇匹配在8K token上下文的性能

仅比随机猜测好约5%

这证明了"海"的大小(上下文长度)比"针"的表述方式影响更大

关键结论

嵌入模型在大海捞针式搜索中的能力，受到的影响**更多来自于"海"的大小**（上下文长度）， **而不是"针"的语义表述方式。**

结论与实践启示

主要研究结论

- 嵌入模型在4K Token之外的长文本中几乎失效
- 在128-1K Token区间内性能急剧下降
- 即使存在词汇精确匹配，性能仍会随上下文增长而下降
- 查询扩展有所帮助，但无法从根本上解决长上下文问题
- 8K Token时，模型的区分能力与随机猜测几乎相同

实践启示

RAG系统设计

应优先划分文档为小于4K Token的片段，而非处理整个长文档

检索策略调整

针对中长文档(1K-4K Token)，考虑结合关键词搜索与语义搜索

查询增强技术

适度的查询扩展(约100个术语)可提高长上下文检索性能

模型应用限制

在关键应用中，认识并避免依赖4K Token以上的嵌入检索能力

03 过去一年有什么进展？

3-1. RAG框架与工具新趋势 | 3-2. 去年RAG社区的重要进展

快速上手: RAGFlow

✓ “开箱即用”的解决方案

1 **环境准备**

RAGFlow提供官方Docker镜像和Compose配置，简化部署流程。

2 **启动服务**

自动部署Flask API、MySQL、MinIO、Elasticsearch/Infinity等组件，并开放Web界面。

3 **创建知识库**

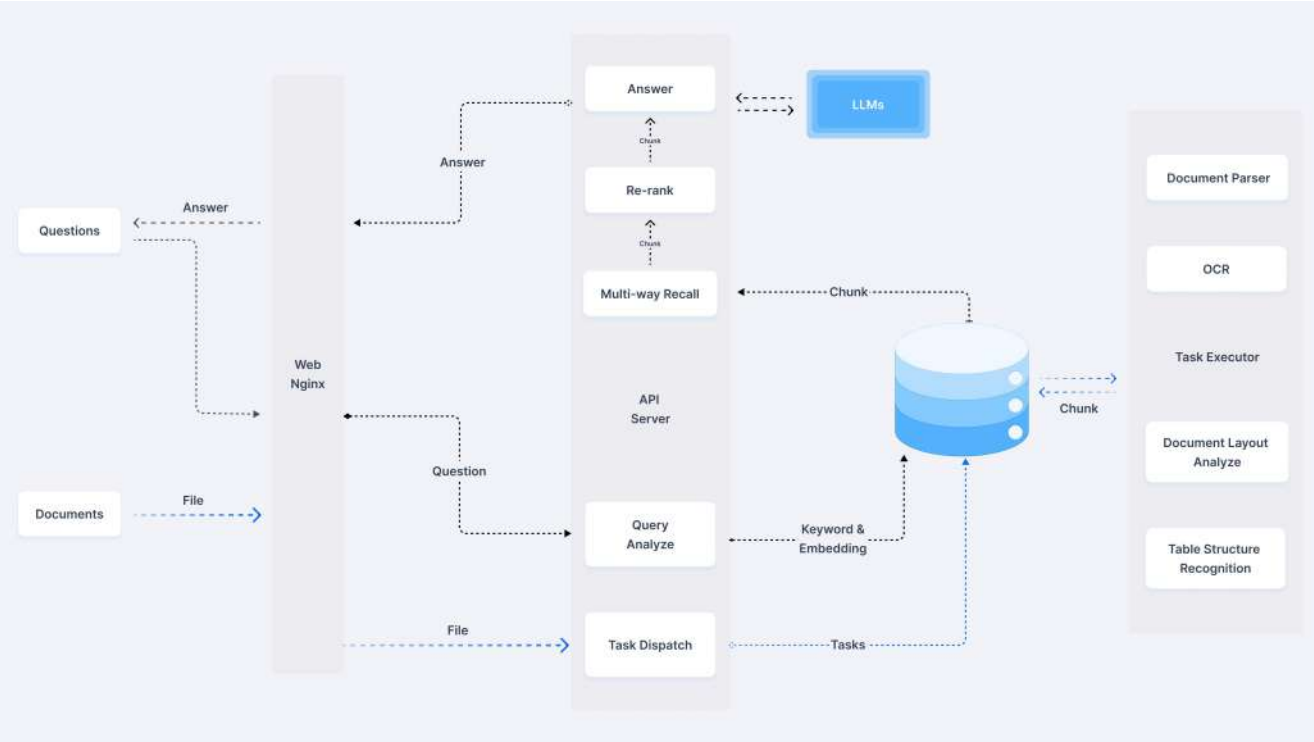
在Web UI中进入“知识库”页面，新建一个知识库。创建时可选择**分块模板**和**嵌入模型**。

4 **文档上传与解析**

在知识库中上传待处理的文档。上传后，点击解析（或通过API触发）启动**DeepDoc分析流程**。

5 **配置问答助手**

选择创建的知识库作为数据源，并配置LLM服务。系统会自动走**解析-召回-生成**流程给出答案。

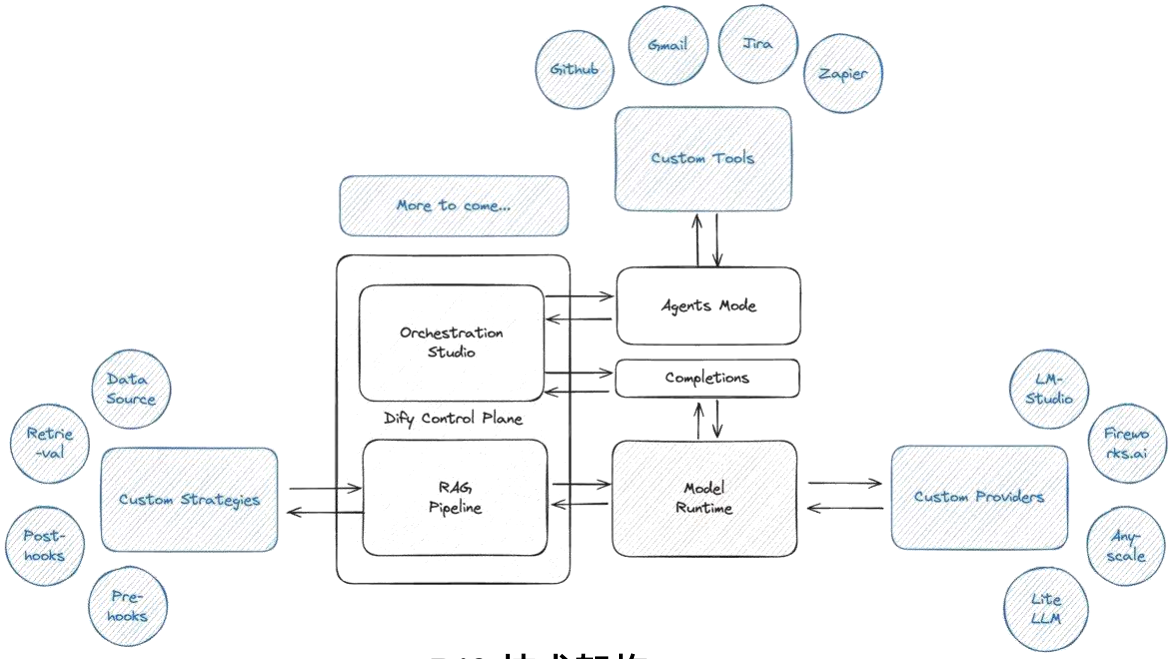


核心模块及交互机制

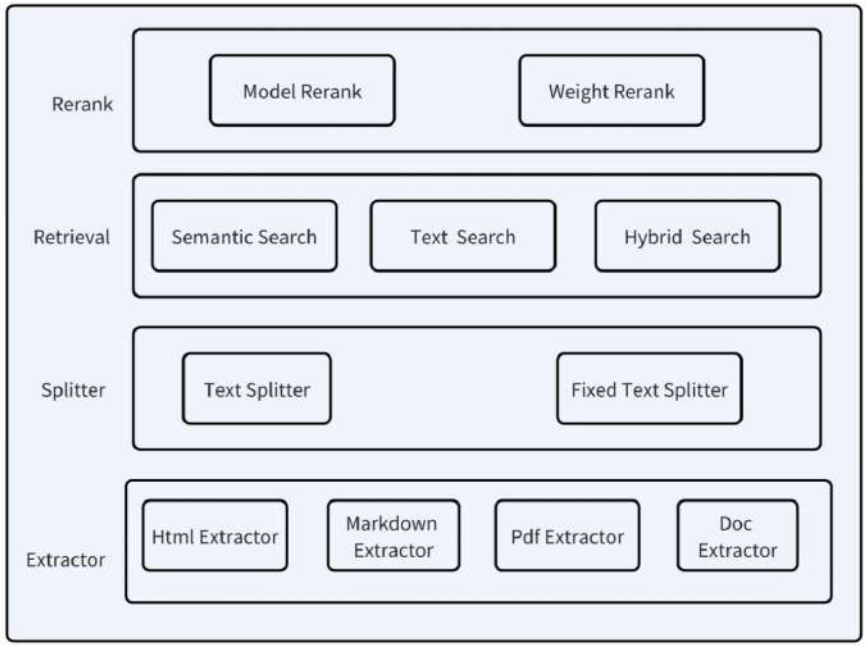
- **深度文档理解（DeepDoc）：** 支持解析 PDF、DOCX、PPT、EXCEL、图片等多种复杂格式文档。
- **向量存储与检索：** 解析出的文本块会连同其Embedding向量一起存储到向量数据库。
- **查询解析（Query Analyze）：** 当用户提出问题时，该模块负责对问题进行预处理，包括关键词提取、查询向量计算等。
- **多路召回（Multi-way Recall）与重排序：** 同时利用关键词检索和向量检索，从向量库中提取候选文档块。之后，Re-rank模块会对这些候选块进行融合排序，选出最相关的前若干个片段作为LLM的上下文输入。
- **答案生成（LLM）：** 排序后的文本块与原始问题一起输入大语言模型模型基于检索到的真实内容生成回答。

快速上手: RAGFlow+Dify的结合

- **Dify的RAG流水线包括：**数据预处理（文件导入、自动分块、元数据标注）、向量化与索引管理（嵌入模型 + 向量数据库）、检索服务（多种检索策略）和生成模块交互（拼装Prompt再调用LLM），所有环节都可在低代码界面中配置。
- **数据预处理：**支持文本、PDF、表格等多格式导入，自动切分为适合上下文窗口的段落；UI界面友好，非技术人员也能快速上手。
- **向量化与索引：**使用内置或外部模型生成文本向量，存入向量数据库（如Milvus、Weaviate等）；向量库通过可配置索引（FLAT、HNSW、IVF等）加速相似检索。
- **检索服务：**执行向量检索（或关键词检索、混合检索），选出与查询语义最匹配的Top-K片段，并可开启Rerank模型进行二次排序。
- **生成交互：**将检索到的上下文与用户问题一起填充到Prompt模板中，再由LLM节点生成答案。Dify支持在Prompt中使用变量和模板，开发者可自定义提示词策略和多轮对话逻辑。



Dify技术架构

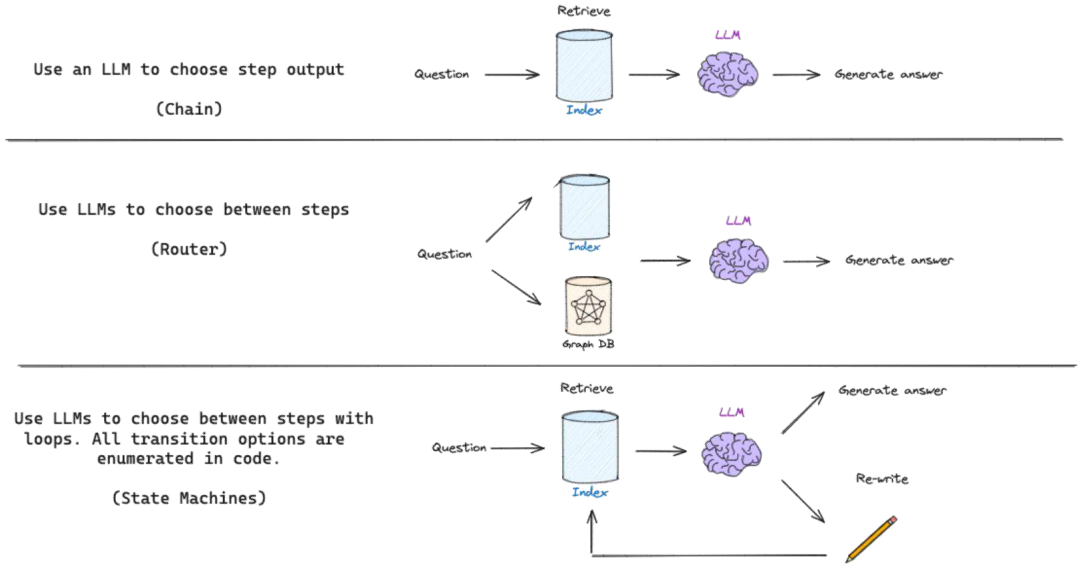


Dify-RAG模块化结构

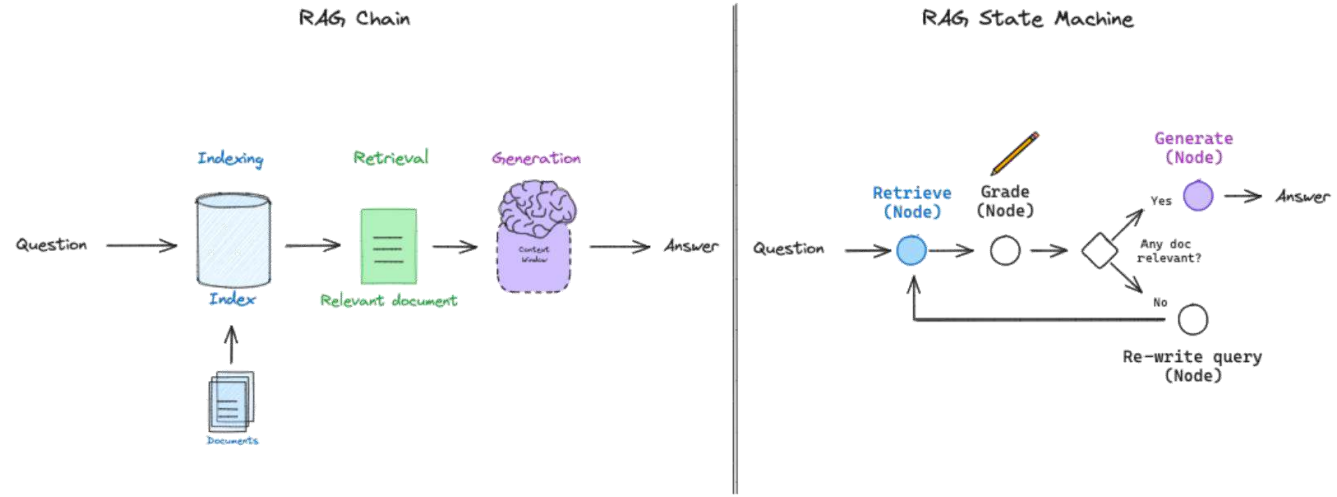
深度定制: LangGraph

LangGraph关键特性

- ✓ 多种触发器与流程控制
- ✓ 流式输出
- ✓ 内存与上下文精细管理
- ✓ 人机协同
- ✓ 企业级部署



Cognitive architectures for RAG



State machines let us design more complex RAG "flows"

LangGraph与RAG的集成方式

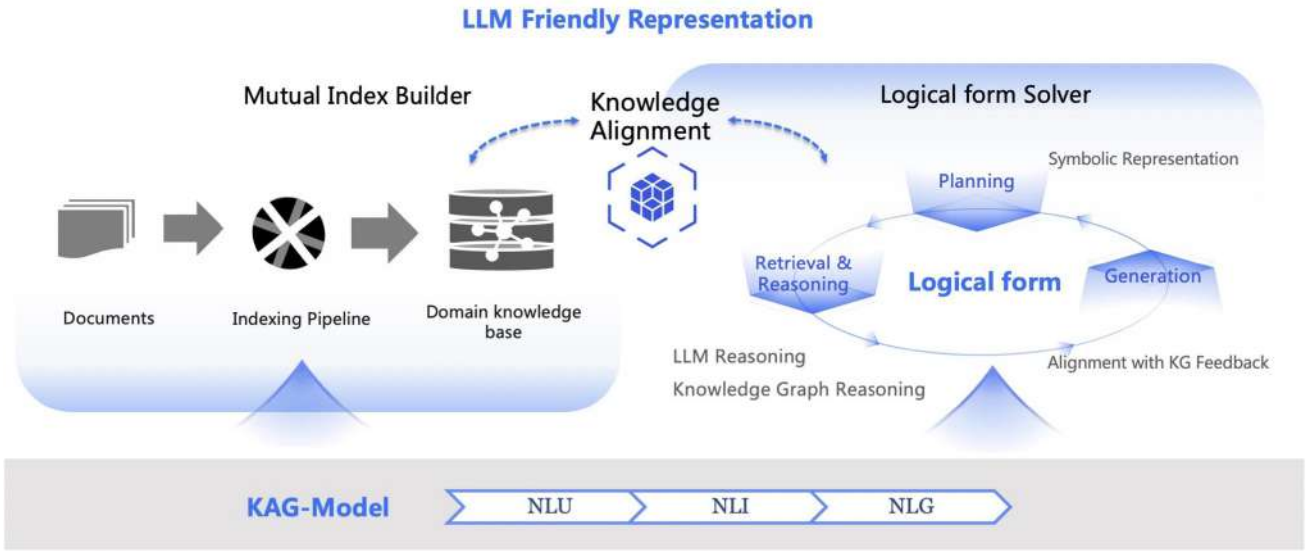
- **检索节点集成：**在LangGraph图中可将向量检索封装为一个节点（例如调用Chroma、Pinecone等向量数据库），将检索到的文档存入状态并传递给后续节点。
- **灵活流程编排：**通过在检索节点和生成节点之间插入评分节点和查询重写节点，实现高级RAG策略。
- **高层抽象：**LangGraph封装了常见RAG模式（如CRAG、Self-RAG），可直接调用或参考其示例流水线。开发者只需定义节点功能（检索、文档评分、生成等）及它们之间的条件流转，即可快速实现可重复使用的RAG应用流程。
- **示例 – LangGraph状态机RAG：**如上图所示，传统的链式RAG直接从检索到生成（左图）；LangGraph状态机RAG引入了“评分节点”和“条件分支”，根据文档是否相关决定是直接生成答案还是重写查询继续检索。

KAG (Knowledge-Augmented Generation)

- 2024年9月，蚂蚁集团及OpenKG SIGSPG组团队等在arXiv公布KAG框架论文。该框架结合KG和向量检索，通过5个增强模块实现领域知识服务性能的大幅提升。实验表明KAG在多跳QA上分别提升HotpotQA 33.5%和2Wiki 19.6%的F1值。

KAG框架

- **KAG-Builder:** 用于构建离线索引。在该模块中，我们提出了一种**面向LLM友好的知识表示框架**，以及知识结构与文本块之间的**互索引机制**。
- **KAG-Solver:** 本模块中引入了一种**逻辑形式引导的混合推理器**，整合了LLM推理、知识推理以及数学逻辑推理。
- **KAG-Model:** 基于通用语言模型，对各模块所需的能力进行优化，从而提升整个系统的整体性能。
- 在KAG-Builder和KAG-Solver两个模块中，均使用了基于**语义推理的知识对齐方法**，以增强知识表示与检索的准确性。



KAG五大增强

- **LLMs友好的知识表示：** **LLMFriSPG框架**通过概念将实例和概念分开，以实现与LLMs更有效的对齐，让KG更好地支持LLMs的应用，并提高两者之间的协同效果。
- **互索引：** 从**term-based倒排索引**到**graph-based倒排索引**，形成图结构化知识与文本知识的互索引模式。
- **混合推理：** 先在**逻辑知识层检索**，若无解则转向**开放信息层**，再通过关联文档检索提高召回率和准确性。
- **语义对齐：** 通过开放信息抽取构建结构化知识，并应用schema约束以提升决策的严谨性，形成基于**SPG的领域知识图谱**。
- **KAG模型：** 对LLMs和KGs的能力进行**对齐**，强调自然语言理解、推理和生成能力，确保从文本中提取结构化信息并提升知识融合效率。

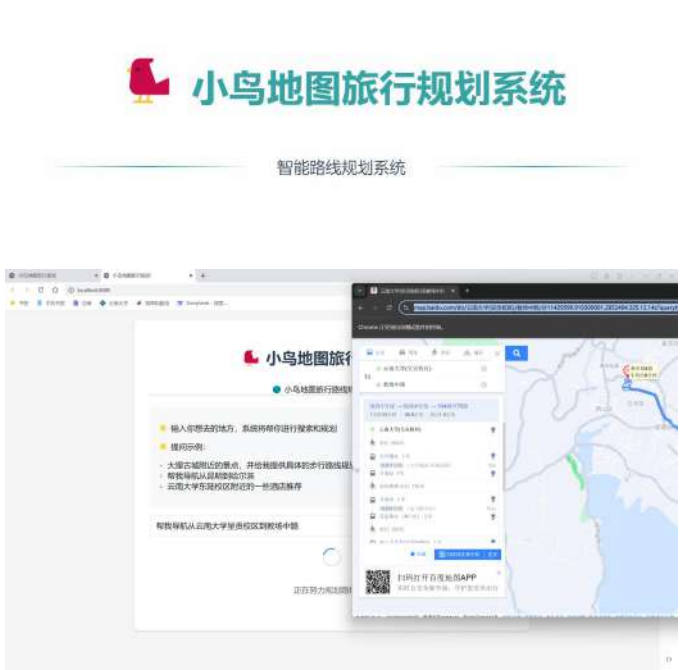
Framework	Model	HotpotQA		2WikiMultiHopQA		MuSiQue	
		EM	F1	EM	F1	EM	F1
NativeRAG [35, 34]	ChatGPT-3.5	43.4	57.7	33.4	43.3	15.5	26.4
HippoRAG [12, 34]	ChatGPT-3.5	41.8	55.0	46.6	59.2	19.2	29.8
IRCoT+NativeRAG	ChatGPT-3.5	45.5	58.4	35.4	45.1	19.1	30.5
IRCoT+HippoRAG	ChatGPT-3.5	45.7	59.2	47.7	62.7	21.9	33.3
IRCoT+HippoRAG	DeepSeek-V2	51.0	63.7	48.0	57.1	26.2	36.5
KAG *w/* LFS_{ref_3}	DeepSeek-V2	<u>59.8</u>	<u>74.0</u>	<u>66.3</u>	<u>76.1</u>	<u>35.4</u>	<u>48.2</u>
KAG *w/* $LFSH_{ref_3}$	DeepSeek-V2	**62.5**	**76.2**	**67.8**	**76.2**	**36.7**	**48.7**

Table 8: The end-to-end generation performance of different RAG models on three multi-hop Q&A datasets. The values in **bold** and <u>underline</u> are the best and second best indicators respectively.

KAG: Boosting LLMs in Professional Domains via Knowledge Augmented Generation. WWW 2025

应用拓展

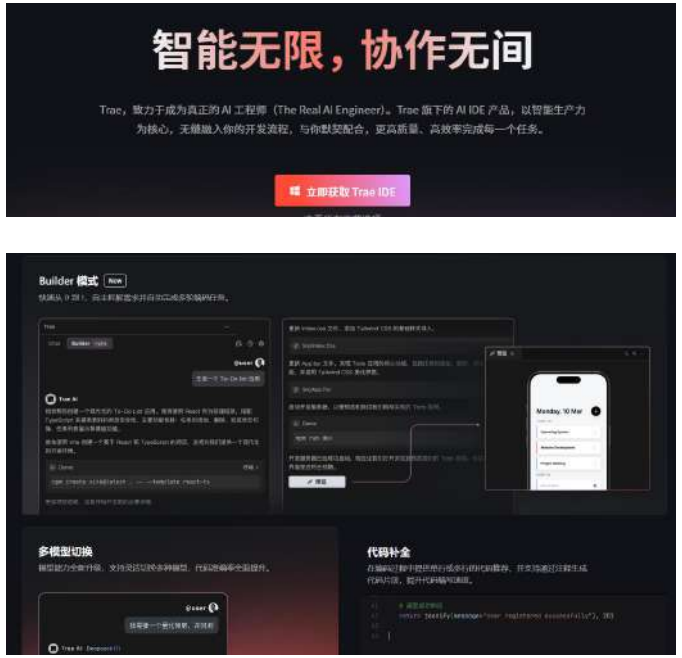
工具/API检索: 小鸟地图



系统集成了高德地图MCP和Microsoft MCP，提供更加准确的地理，天气信息。调用浏览器控制工具，快速抓取内容，实现智能体决策的可视化。

https://github.com/Heisenberg-Gao/TravelBird

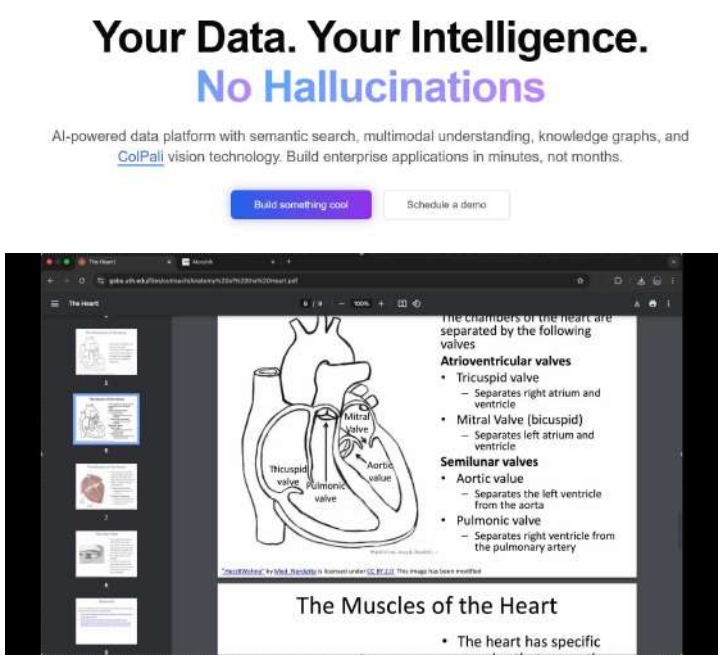
代码RAG: Trae



系统不仅检索当前源码文件，且检索一系列编码相关操作的数据（编辑历史、debug尝试、UI操作），以构建更详细的上下文信息生成用户需求代码。

https://www.trae.com.cn/home

多模态RAG: Morphik



系统支持多模态数据检索（文本、PDF、图片、视频等）。基于ColPali完成多模态嵌入，结合知识图谱自动提取实体和关系，提升检索结果。

https://github.com/morphik-org/morphik-core

03 过去一年有什么进展？

3-1. RAG框架与工具新趋势 | 3-2. 去年RAG社区的重要进展

去年RAG社区的进展：模型与应用的协同进化

2024.5-2025.5

🧠 模型侧进展

🚀 基座模型能力深化

通过MOE和高效预训练技术，基座LLM的能力持续提升，在复杂推理和上下文理解方面取得突破。

例如：DeepSeek-V3, Qwen-3

🧩 深度思考能力提升

引入Test-time Scaling等Post-Training优化技术，显著提升模型系统性分解复杂问题的能力。

例如：OpenAI O3, DeepSeek-R1

🔍 知识图谱融合

RAG技术与知识图谱深度结合,在细粒度知识表示，依赖关系、多跳推理等方面取得突破。

例如：KAG, PIKE

📚 上下文理解增强

通过先进的注意力机制和长文本处理技术，极大提升模型的上下文窗口长度，达到百万token级别。

例如：MinMax-01, Qwen-Long

🚀 应用侧进展

🤖 通用Agent能力提升

Agent与多Agent协作框架日趋完善，逐渐胜任更加复杂的工作，可处理的场景复杂度提升。

例如：Manus

🔄 MCP协议标准化

智能体通信协议取得重大进展，推动智能系统间以及对于工具的互操作性，系统间协作效率提升。

例如：Claude MCP

🔗 AgenticRAG

RAG与推理、规划能力的结合，从单一的知识检索工具向协作助手转变，提供更强的决策支持。

例如：Deep Research，DeerFlow

🏢 场景落地突破

RAG在多个垂直领域实现更大范围的应用，逐渐从概念验证走向商业实践。

例如：RAGFlow, MaxKB (金融、医疗、法律、政务等)

DeepSeek大模型最新技术亮点

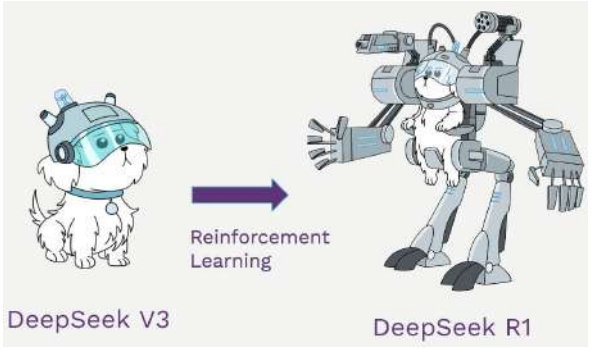
- R1不是从头训练，在V3上后训练几周得到
- 深度思考训练依靠可验证问题、数学代码
- 测试时计算扩展性增强，不再需人工标注
- Aha顿悟时刻R1 zero证明大规模RL有效
- 长思维链CoT能力可以通过强化学习RL隐式激发，而不需要通过监督学习SFT显式训练

DeepSeek offers a range of open weights models and efficient distilled variants.

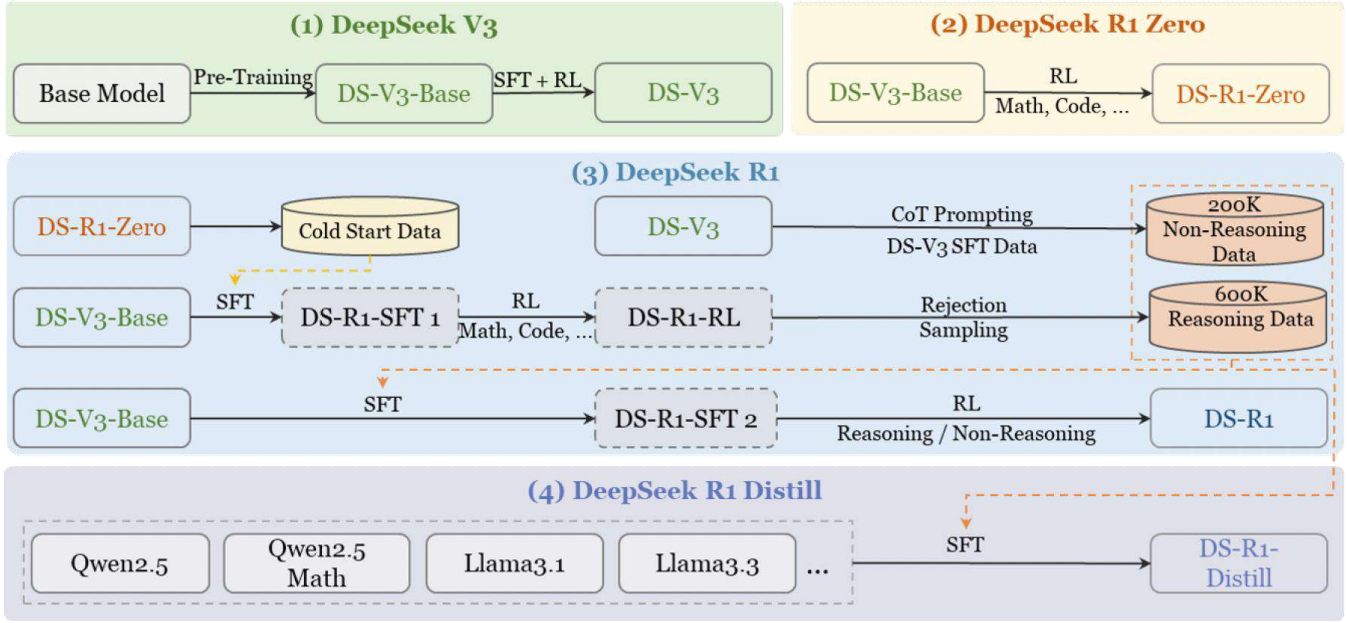
Base and R1 Models (671B)
- **DeepSeek-V3**: Base MoE model
- **DeepSeek-R1-Zero**: Pure reinforcement learning
- **DeepSeek-R1**: Cold-start data before RL

Distilled Models
- **DeepSeek-R1-Distill-Qwen** (1.5B, 7B, 14B, 32B)
- **DeepSeek-R1-Distill-Llama** (8B and 70B)

- 测试时计算使得思考问题的中间步骤成为可能
- 模型在回答之前会“推理”并探索可能的解决方案



- 长思维链冷启动（不稳定和可读性差）
- 基于规则奖励的推理导向RL
- 拒绝采样与监督微调
- 多场景的大规模强化学习



长CoT能力的隐式激发：先验注入

长CoT的格式设计：总体推理流程为思考部分和总结部分

<think> reasoning </think> <answer> answer </answer>

长CoT的奖励设计：格式性奖励和正确性奖励

- **格式性奖励**：模型是否把思考过程放在了<think>和</think>之中；
- **正确性奖励**：模型是否在思考+总结后生成了正确的答案

Deep Research

DeepResearch发展时间线

- **2024年9月：** **OpenAI o1-preview**模型首次公开测试，o1模型引入了“先思考再作答”的链式推理机制，擅长逐步解决复杂问题。
- **2025年2月：** OpenAI在**ChatGPT**中上线**DeepResearch功能**，初期向Pro用户开放。官方公布模型在重大评测中的成绩，在Humanity’s Last Exam基准测试中达到**26.6%**的准确率，远超前代o1的9.1%；并在通用AI助手基准GAIA上以平均**72.57**的成绩登顶榜首。
- **2025年2月：** OpenAI发布DeepResearch后，LangChain、Google等大厂发布DeepResearch开源项目。

https://openai.com/index/introducing-deep-research/

OpenAI DeepResearch能力分析

- **推理能力**
  - ✓ 长链条推理与多跳问答
  - ✓ 知识整合与分析能力
  - ✓ 减少幻觉与不确定性处理
- **数据处理**
  - ✓ 海量非结构化数据检索
  - ✓ 多样化数据类型处理
  - ✓ 衔接多个外部数据源和工具接口
  - ✓ 信息整理与报告生成
- **技术整合与系统设计**
  - ✓ Agent模块化架构
  - ✓ 任务分解与规划
  - ✓ 记忆与上下文跟踪
  - ✓ 准确性与可验证性保障

Manus

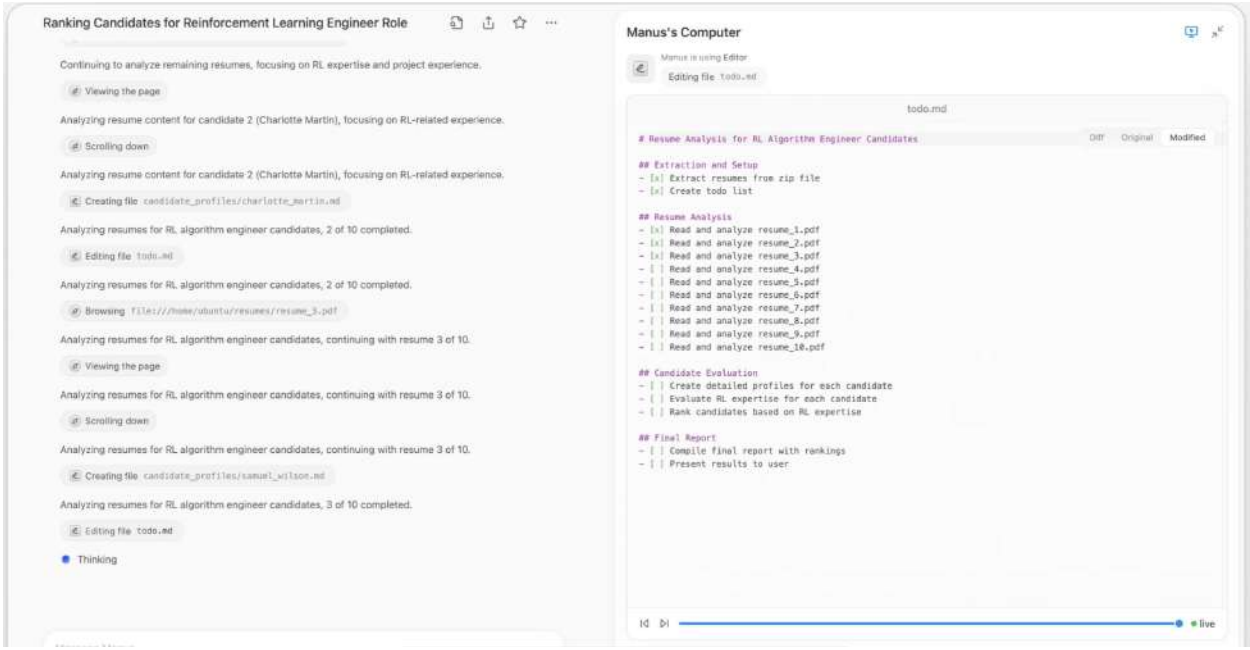
- **2025年3月：** Monica.im发布产品**Manus**。
- 同月，国内社区出现Manus开源替代方案 “OpenManus”。

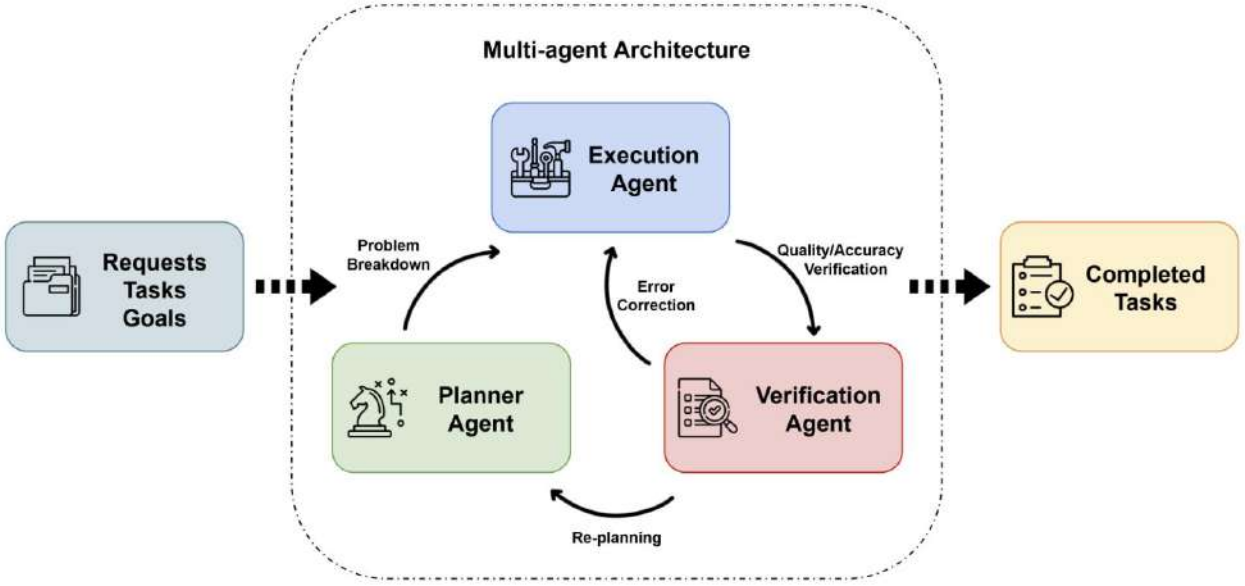
核心技术

- **多代理（Agent）架构：** 基于大模型主干，集成多个小模型组件，运行于独立虚拟机，包含规划、执行、验证代理，分工协作提升效率。
- **多模型协同：** 大模型作“大脑”提供智能推理，小模型像“手脚”负责规划、执行、验证等具体任务，如编写代码、分析数据。

核心特点

- **多模型动态调用：** 灵活调用GPT-4、Claude3、Gemini等主流大模型，整合优势能力，形成综合解决方案。
- **工具链集成：** 通过“虚拟环境”集成代码执行、网页操作、数据分析等工具，如自动编写Python脚本分析股票数据并生成可视化Dashboard。
- **自主规划与纠错：** 具备拆解复杂任务能力，通过“规划-执行-反馈”循环动态调整策略，可恢复中断任务进度。



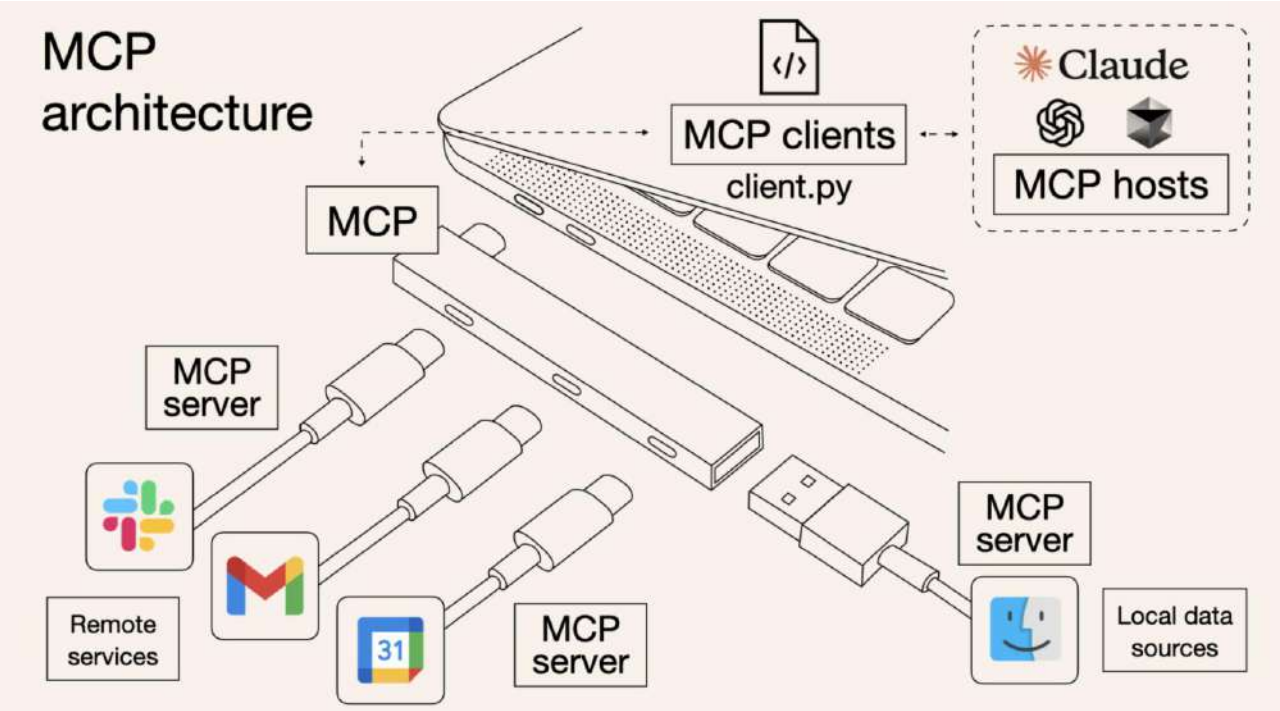


From Mind to Machine: The Rise of Manus AI as a Fully Autonomous Digital Agent. arxiv. 2025

MCP (Model Context Protocol)

MCP技术原理

- **MCP (Model Context Protocol)**是一种开放标准，旨在实现**LLM与外部数据源和工具之间的双向交互**，从而为模型提供即时、动态的上下文。
- **MCP**沿用**客户端–服务器架构**，在其中“客户端”（例如运行LLM的应用）通过统一协议与不同的“MCP服务器”连接，每个服务器代表一个独立的数据源或服务（如数据库、文件系统、API等），并提供可调用的工具和资源。
- 例如，开发者只需启动相应的MCP服务器（可能是本地程序或远程服务），LLM即可通过标准化接口自动发现并调用该服务器的功能，无需为每个服务硬编码适配。



MCP发展时间线

- **2024年11月：** **Anthropic**正式发布**Model Context Protocol标准**，并开源规范文档和SDK，同时提供了多个预置的MCP服务器示例（包括Google Drive、Slack、Git仓库、PostgreSQL等），此时MCP概念首次提出。
- **2025年2月：** 在 AI工程师峰会等技术会议上，Anthropic深入展示了MCP的工作原理和使用方法，引发业界广泛讨论。
- **2025年3月：** **OpenAI**在其Agents SDK中新增对MCP的支持（GitHub发布相关更新），允许其代理框架通过MCP与外部数据源对接。
- **2025年4月：** **Google**宣布计划在其Gemini模型及开发工具集中加入对Anthropic MCP标准的支持。

MCP的核心设计目标

- **模块化与复用**：将对外工具调用按服务进行模块化封装，一次配置后可在不同应用间重用。
- **标准化接口**：采用统一协议简化集成流程，任何遵循规范的客户端都能调用任何兼容的MCP服务器。
- **动态发现**：客户端可自动识别可用的MCP服务器及其功能，无需预先配置，实现灵活扩展。
- **实时上下文**：通过双向连接，LLM可以在推理时访问最新数据或执行任务（如查询数据库、调用函数等），让模型的回答更准确及时。

https://modelcontextprotocol.io/introduction

2025，我们需要什么样的RAG?

智能知识系统

超越简单检索，能主动理解、关联和生成知识，具备自适应学习能力的智能知识引擎

复杂任务处理

突破简单问答局限，能够处理多步骤推理、长程依赖、复杂逻辑与专业领域深度任务，实现类人思维链路的智能决策

业务落地驱动

不再停留于技术演示，深度融入业务流程，提供可靠的决策支持与洞察，创造可量化的商业价值与竞争优势

技术路径

- 与图/图谱/图计算引擎的深度融合
- 与深度推理能力的协同
- 更加自主的流程设计、控制与编排
- 更多的业务场景的落地

04 RAG与Graph的深度融合

4-1. 知识表示与关系建模 | 4-2. 图推理能力增强 | 4-3. 个性化GraphRAG

RAG+Graph 成为香饽饽

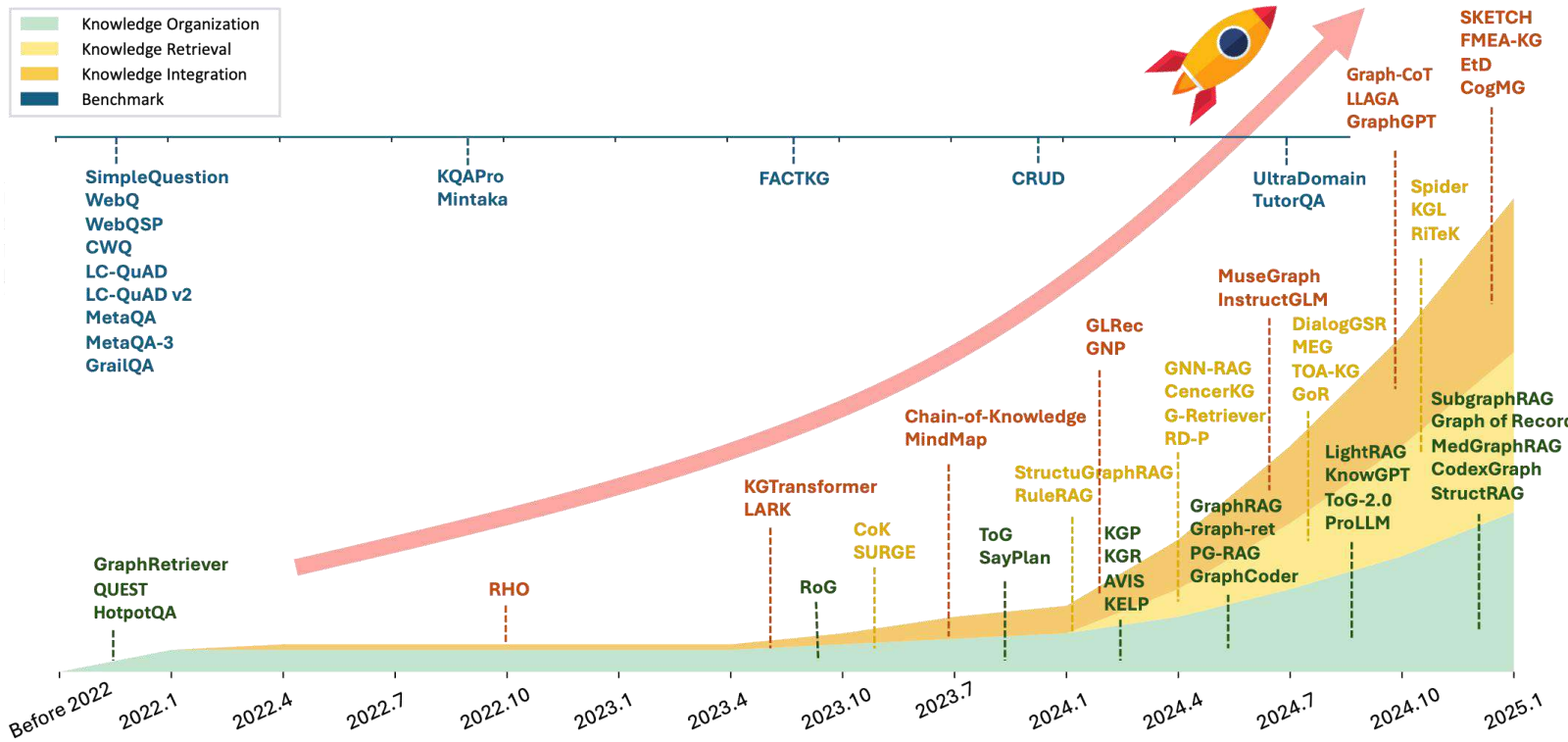
GraphRAG成为前沿探索方向

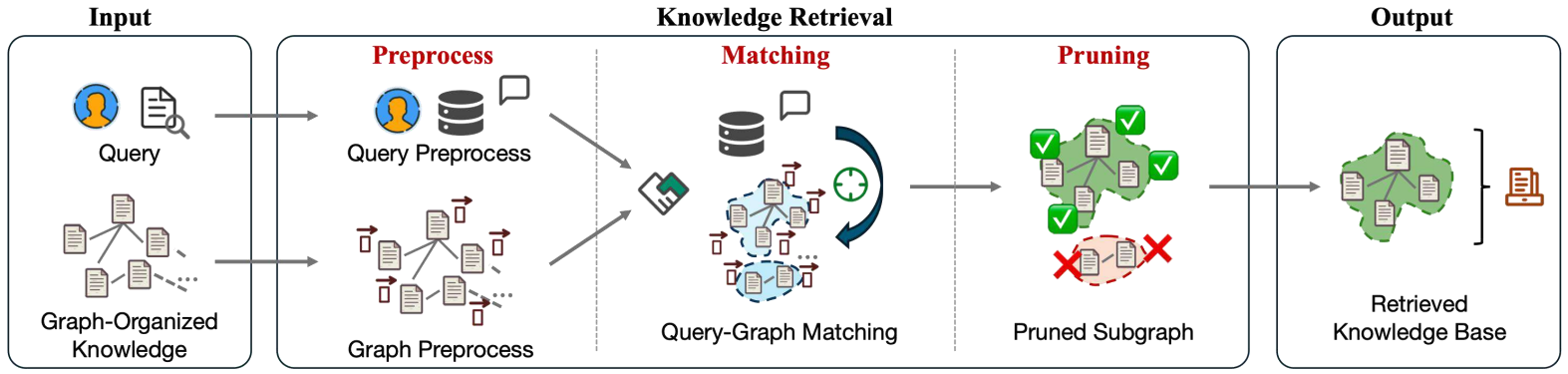
Graph结构的潜在优势

- 更强的复杂推理能力
- 更清晰的可解释可追溯性
- 更好的知识表达与关联性
- 更灵活的知识源集成能力

GraphRAG的核心议题

- 图推理能力增强
- 图结构化的知识表示
- 高效的图信息检索
- 利用图上知识的校验





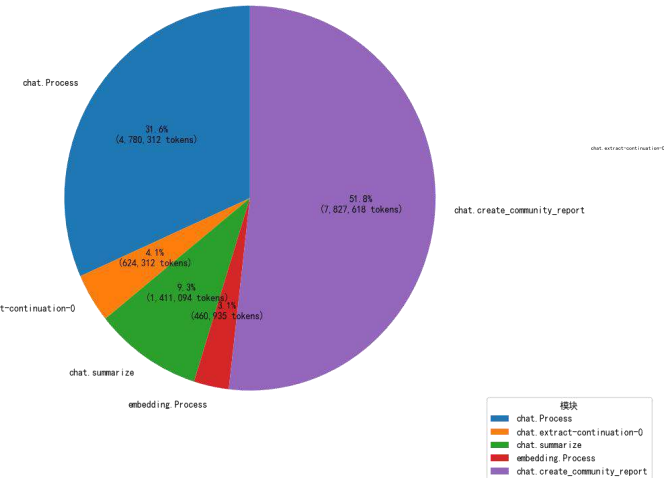
A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models, 2025.

更轻量高效的GraphRAG

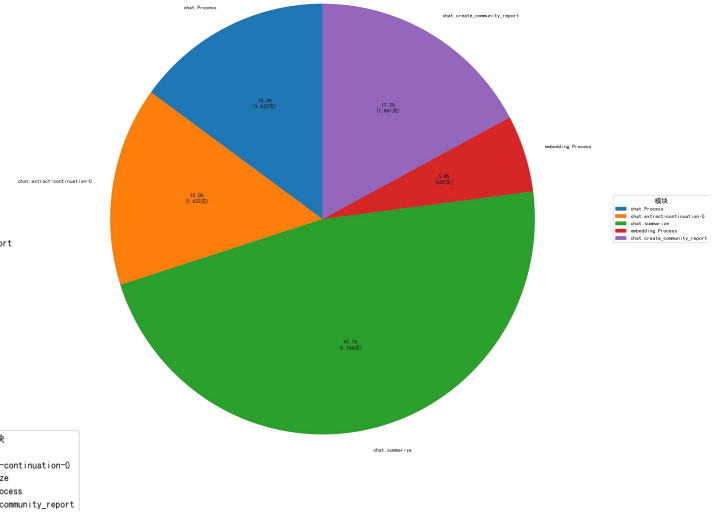
Microsoft GraphRAG 较高的构建成本

- 原始语料：0.8 M tokens（约300篇新闻）
- 索引消耗：16 M tokens（2000%+ 原始语料）
- 其中：抽取（35%）摘要（10%）社区摘要（50%）
- HTTP 请求：10,840+

各个模块Token消耗占比

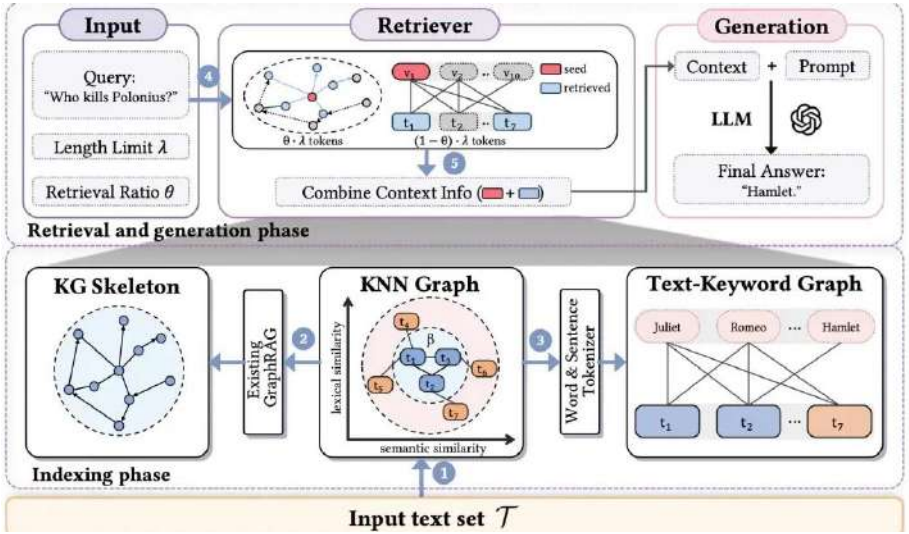


各个模块Http请求占比

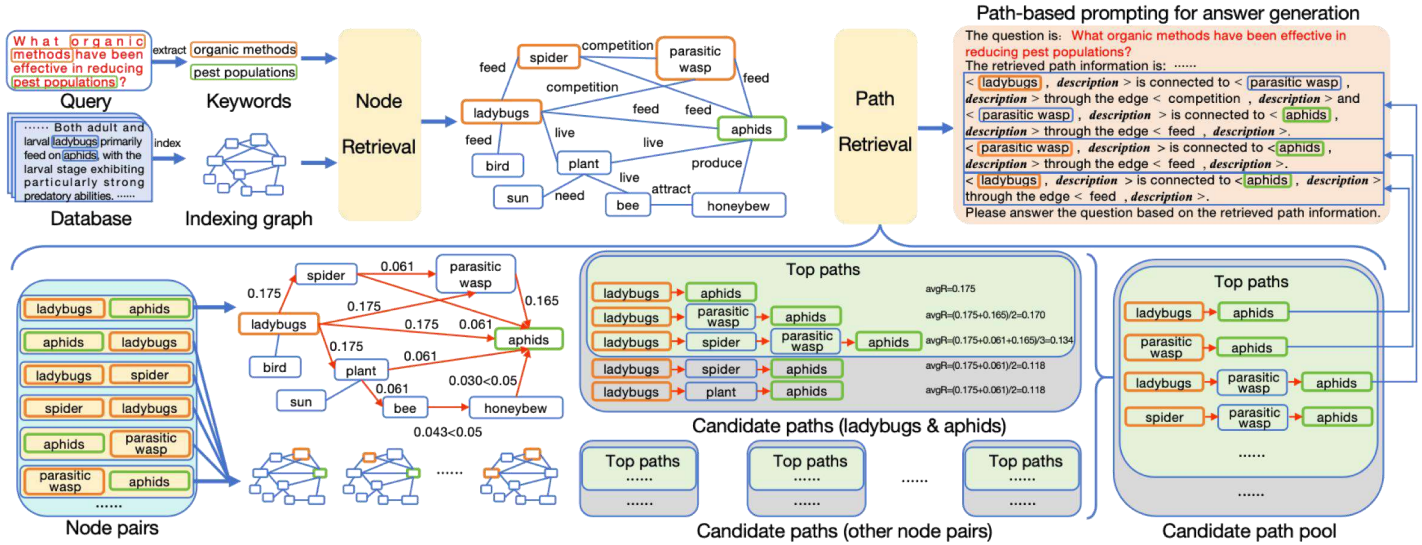


如何构建更轻量级、更高效的GraphRAG？

KET: 只构建核心图



PathRAG: 基于路径的剪枝



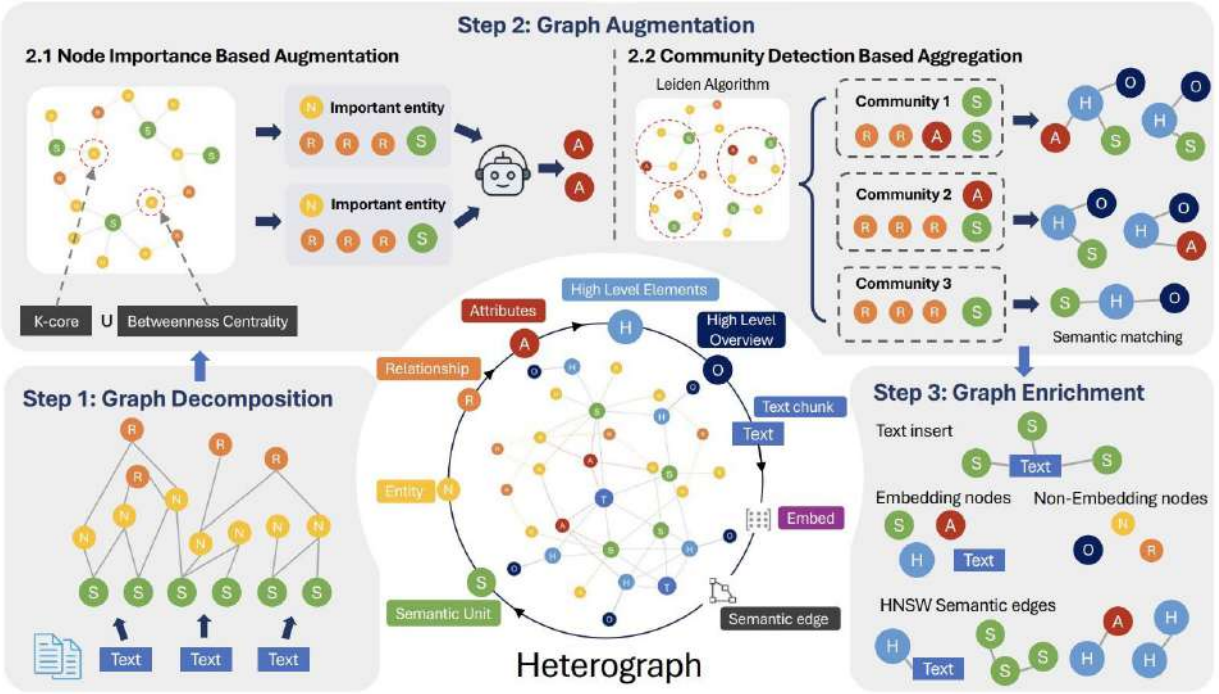
KET-RAG: A Cost-Efficient Multi-Granular Indexing Framework for Graph-RAG, arxiv, 2025.

PathRAG: Pruning Graph-based Retrieval Augmented Generation with Relational Paths, arxiv, 2025.

更结构化的表示

NodeRAG: 基于异构图的细粒度检索

- 精细和可解释的检索：异构图可以细粒度地从功能上区分不同节点，使得算法能够识别关键的多跳节点。
- 统一层次信息检索：将文档分解信息和LLMs提取的信息统一作为异构图中的节点。



NodeRAG: Structuring Graph-based RAG with Heterogeneous Nodes, arxiv, 2025.

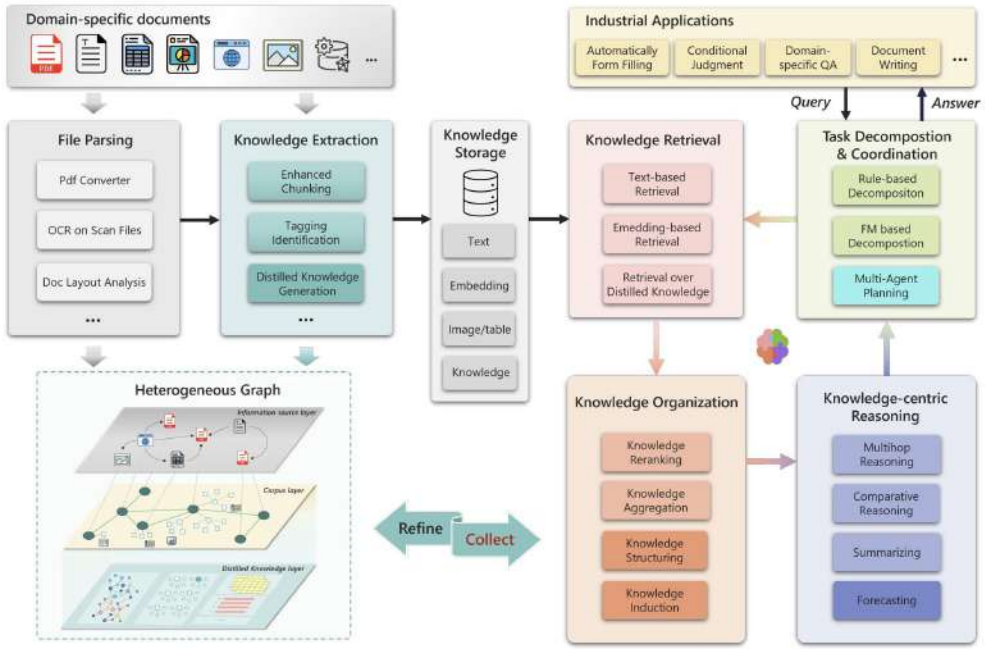
PIKE: 任务分类与RAG系统分级

专业知识增强：

- 知识原子化：将文档分解为细粒度的*问题-答案* 对
- 多层异构知识图谱：信息资源、语料和提炼知识的三层图结构。

动态任务分解：

- 知识感知任务分解：将任务和RAG系统划分为四级，结合可用知识动态生成子查询，解决多跳推理问题。



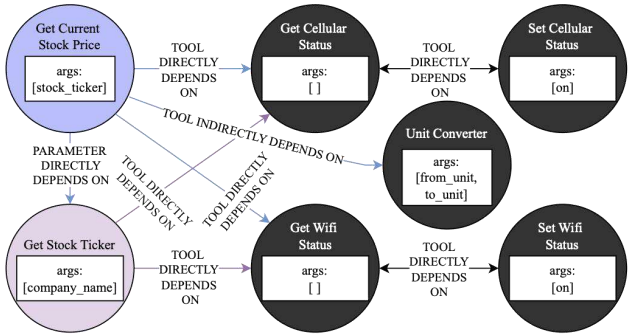
PIKE-RAG: Specialized Knowledge and Rationale Augmented Generation, arxiv, 2025.

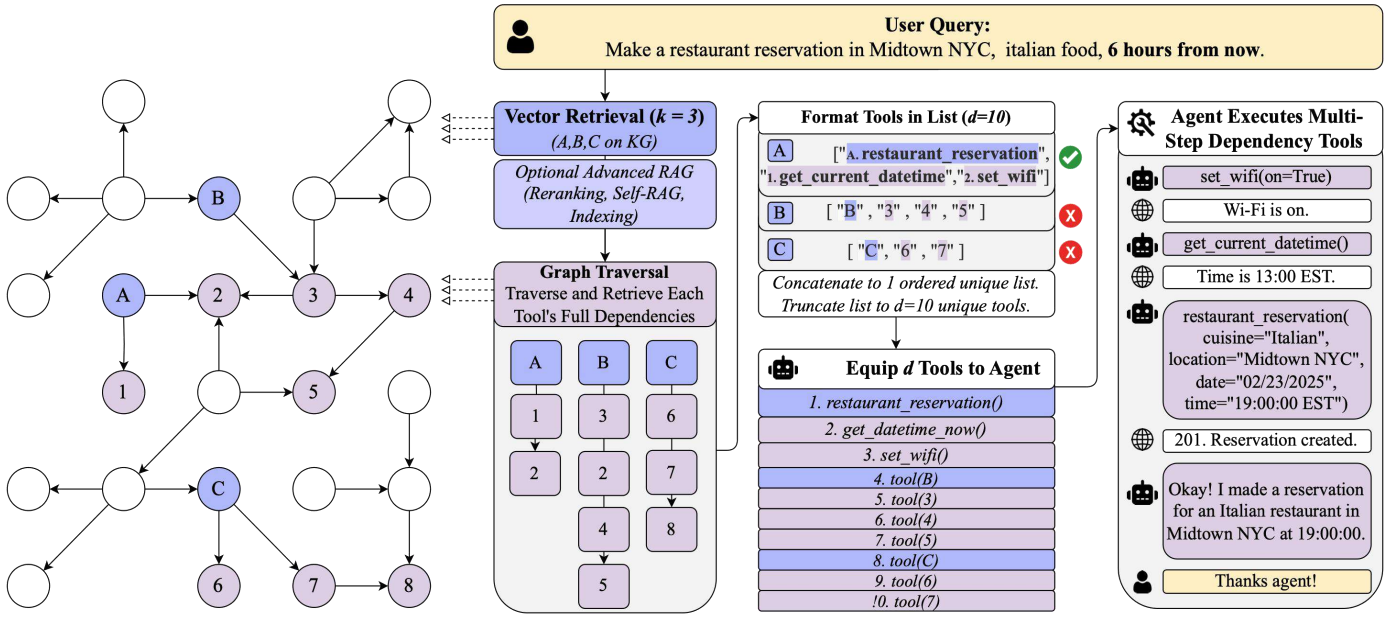
依赖关系建模

Graph RAG-Tool Fusion: 工具间关系建模

- **工具图谱依赖关系：**

通过结合向量检索+知识图谱，模型有效地捕捉了工具之间的四种结构化依赖关系。

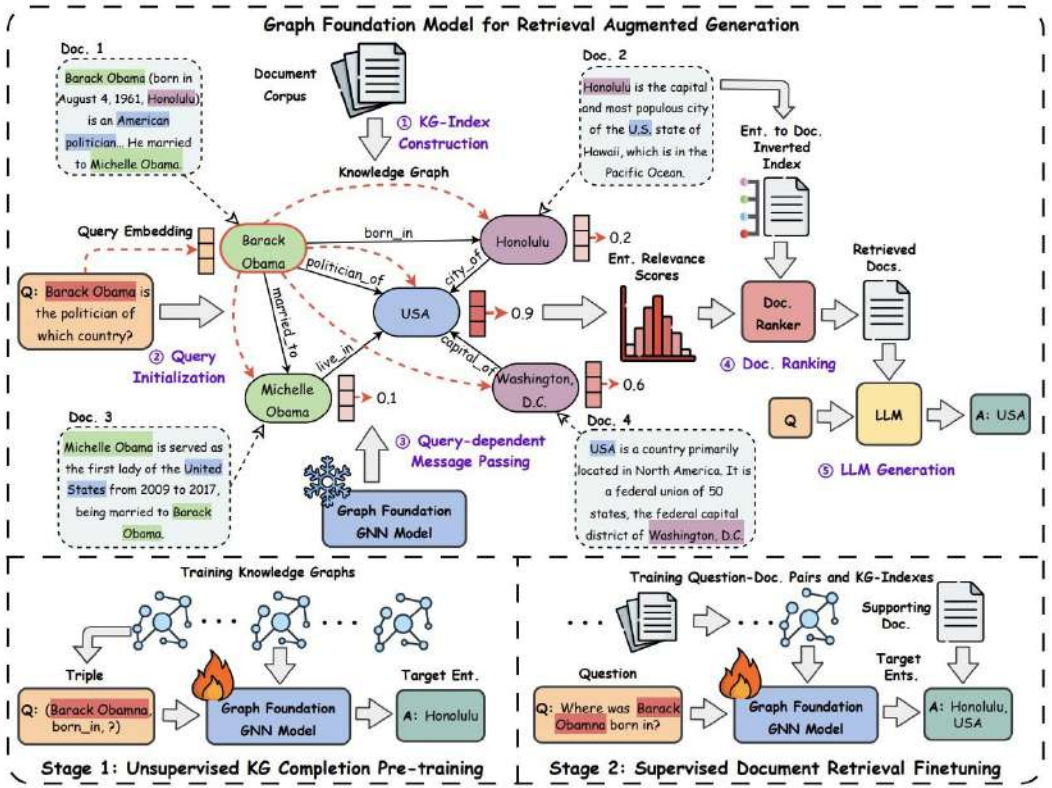




Graph RAG-Tool Fusion, arxiv, 2025.

GFM-RAG: 无需微调捕捉查询-知识关系

- **捕捉依赖关系：**在图结构上进行推理，以捕捉复杂的查询 - 知识关系。
- 该模型是首个适用于未见数据集的图基础模型，无需任何微调即可进行检索增强生成任务。



GFM-RAG: Graph Foundation Model for Retrieval Augmented Generation, arxiv, 2025.

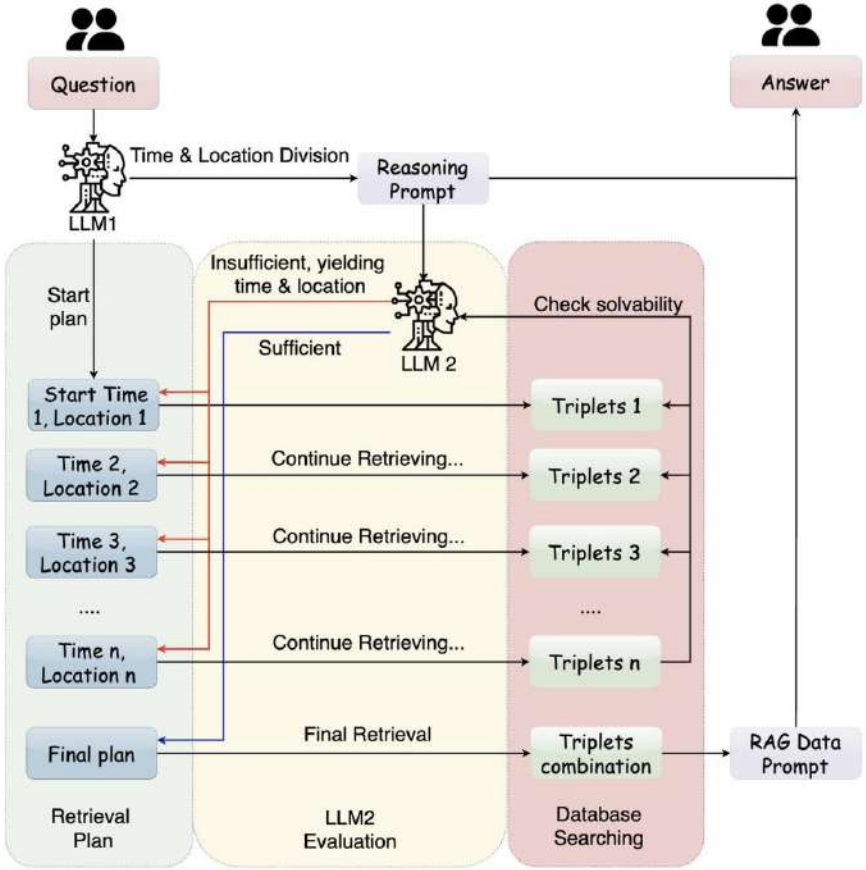
多跳迭代推理

KG-IRAG: 知识图谱上的推理

- 通过迭代检索步骤，从外部KGs中逐步收集相关数据，实现逐步推理，适用于需要动态时间数据提取和推理的场景。

- 迭代推理：每次检索三元组后，系统评估当前三元组集合与推理提示是否足以回答查询。

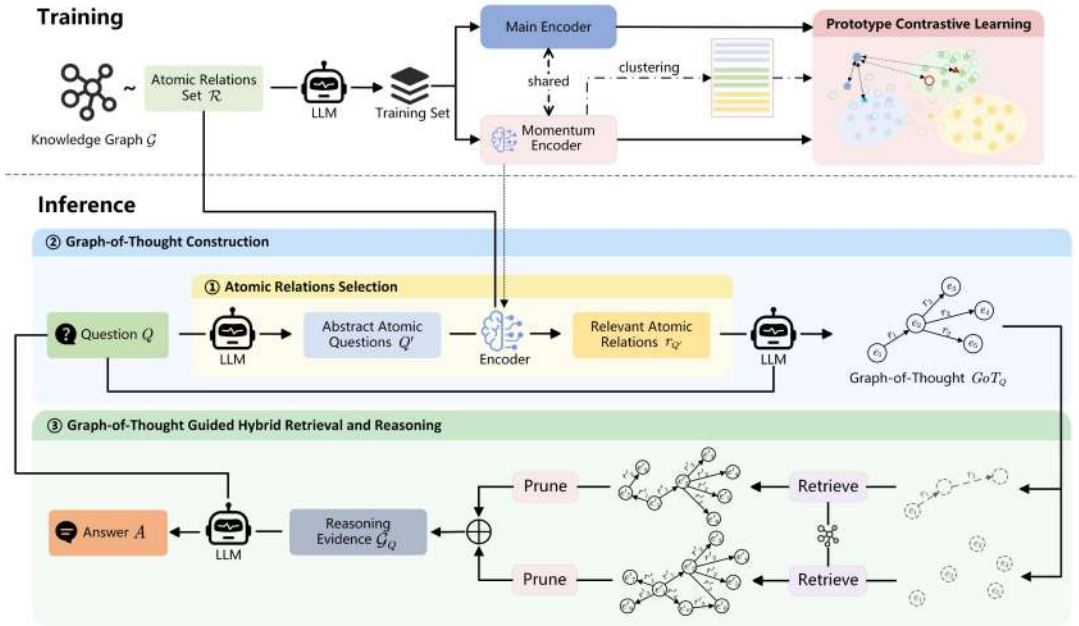
- 另一个LLM负责此评估，若数据不足，系统调整搜索标准并继续检索。



KG-IRAG: A Knowledge Graph-Based Iterative Retrieval-Augmented Generation Framework for Temporal Reasoning, arxiv, 2025.

GoT-R: 思维图辅助推理

- 思维图引导检索和推理：推理阶段分为三步：原子关系挑选；思维图构建；混合检索与推理。其中，最后一步将思维图作为检索对象生成推理证据。

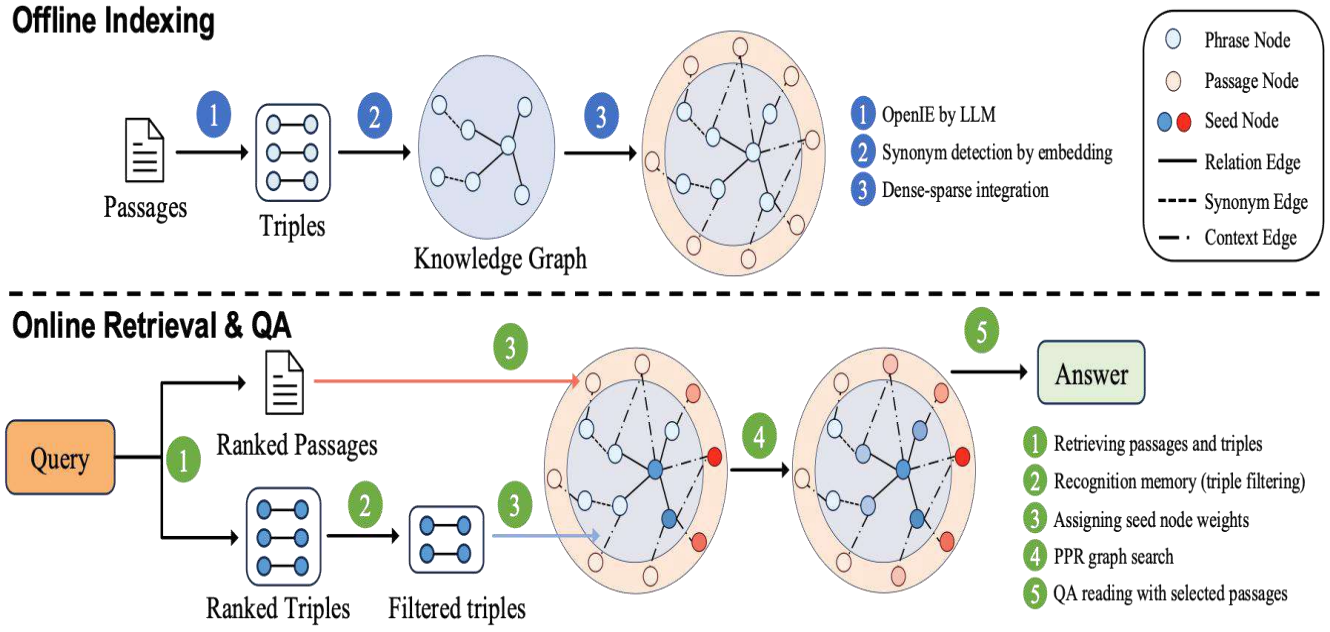


GoT-R: Enhancing Large Language Models for Complex Question Answering with Graph-of-Thought Guided Reasoning, arxiv, 2025.

个性化 GraphRAG

HippoRAG2: 模拟人类长期记忆

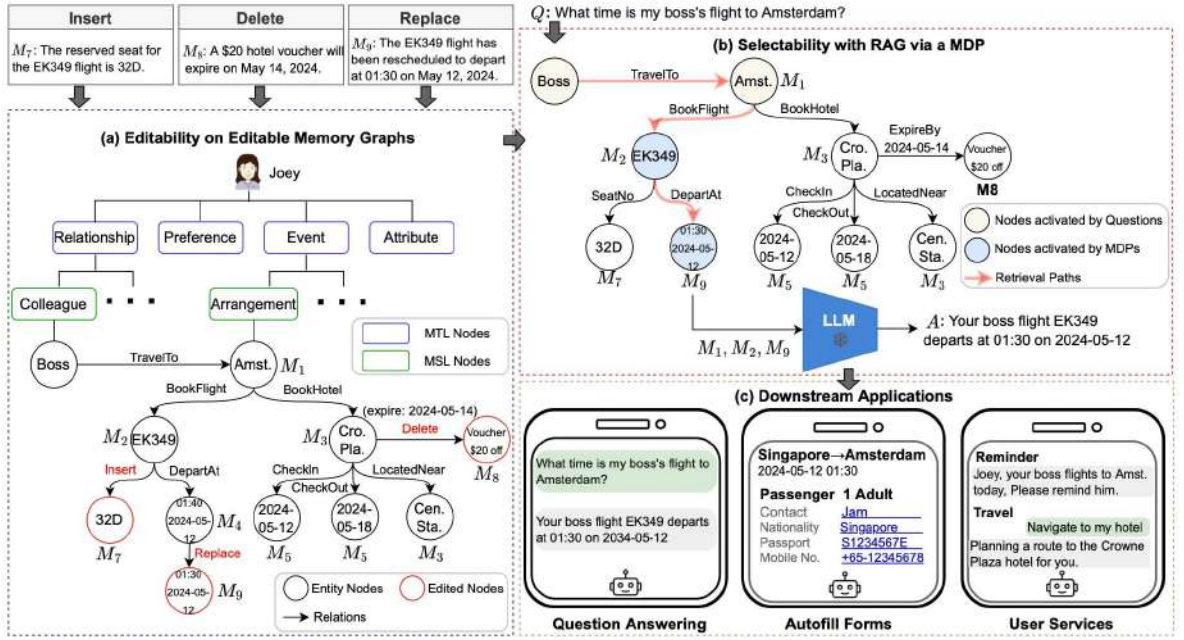
- 个性化的PageRank: 进行多跳推理，提升复杂问题的关联性检索能力。
- 密集-稀疏编码结合：引入短语节点和段落节点，模拟人类大脑的密集-稀疏编码机制，更好地整合概念和上下文。



From RAG to Memory: Non-Parametric Continual Learning for Large Language Models, arxiv, 2025.

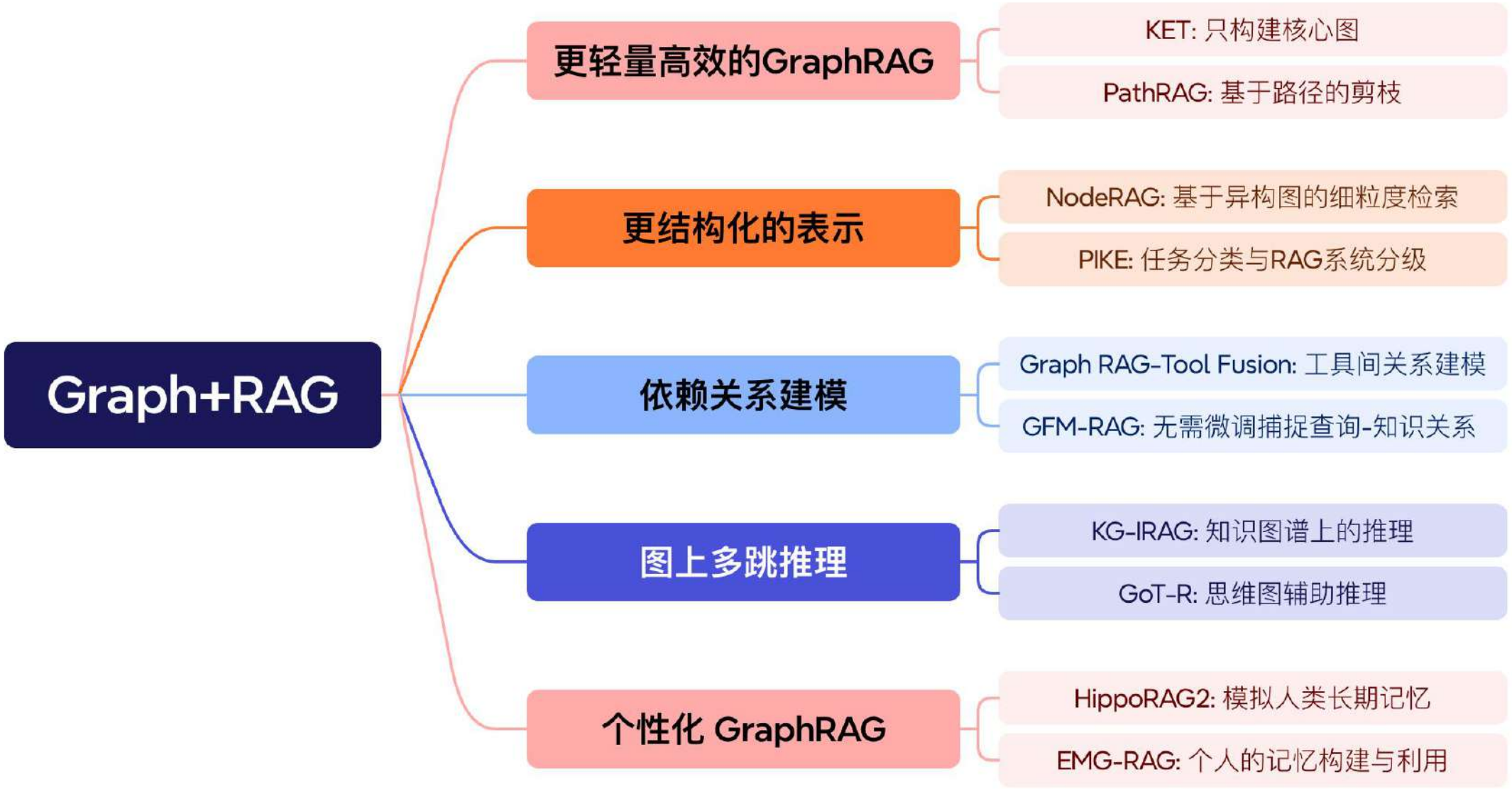
EMG-RAG: 个人的记忆构建与利用

- 个人可编辑内存图与RAG结合：EMG模型缓解个人代理三大挑战问题：数据收集、可编辑性和记忆选择。
- 可编辑记忆图（EMG）：EMG是一种三层数据结构，用于支持记忆的插入、删除和替换操作。



Crafting Personalized Agents through Retrieval-Augmented Generation on Editable Memory Graphs, arxiv, 2024.

总结：Graph与RAG 协同



05 RAG与深度推理能力协同

5-1. RAG+Reasoning的新趋势 | 5-2. RAG+Reasoning协同的实现 | 5-3. RAG+Reasoning协同的优化

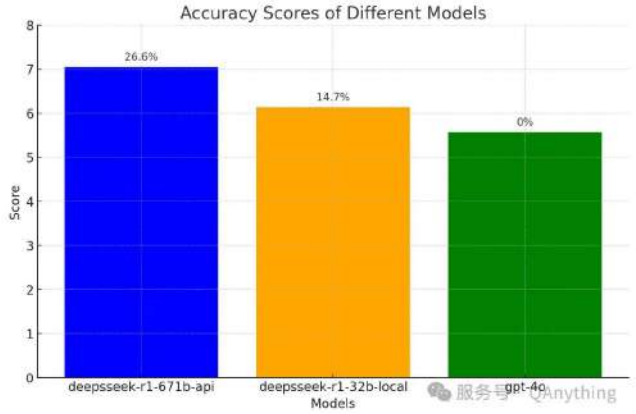
RAG + Reasoning 是否成就新的CP？

OpenAI O1 和DeepSeek-R1等慢思考模型展示出了强大的复杂任务处理能力，对RAG的影响和启发是什么？

推理模型（DeepSeek-R1）在RAG场景上的表现如何？

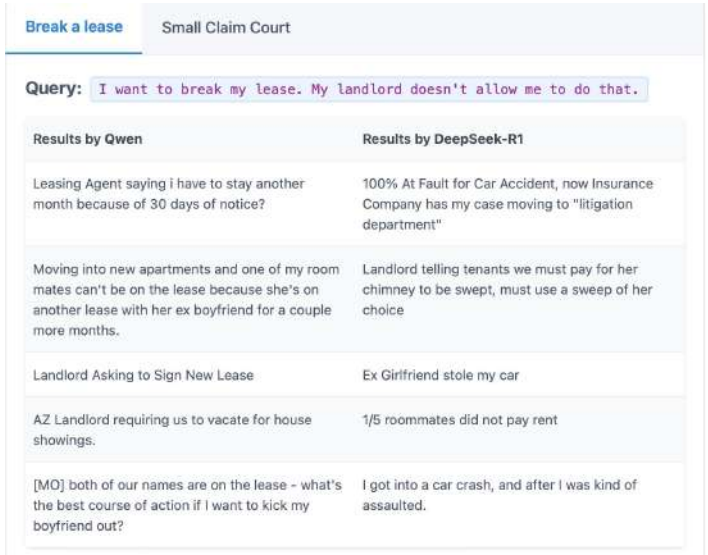
QAnything测试

- R1模型无需复杂prompt工程，直接明确需求即可，否则可能适得其反。
- 能假设用户真实意图，有上下文理解和推理能力强，关键在于检索环节。



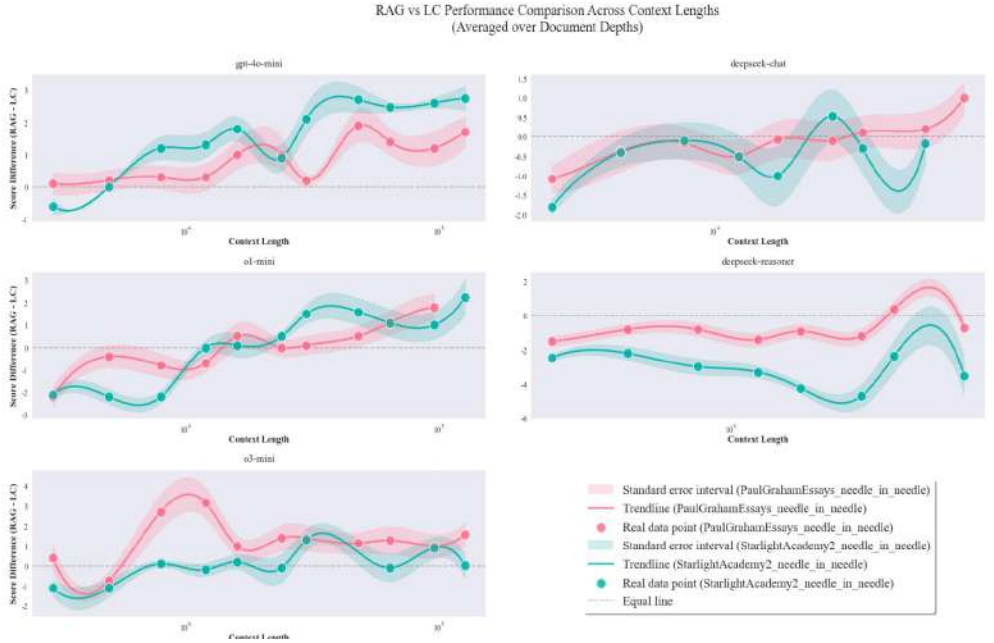
R1 as Embedding

- R1 作为检索模型表现不好，不如专门的Embedding模型。
- 擅长推理重点是顺序思考和逻辑连接。不擅长将文档映射到语义空间中。
- 有时会遵循导致主题相关但实际上无关的结果的推理路径。



在长上下文中的性能

在大海捞针（U-NIAH）的复杂RAG场景中（更长的上下文、更多的干扰项、更分散的关键信息）DeepSeekR1-RAG下的表现并不如直接使用R1，更容易受到干扰。



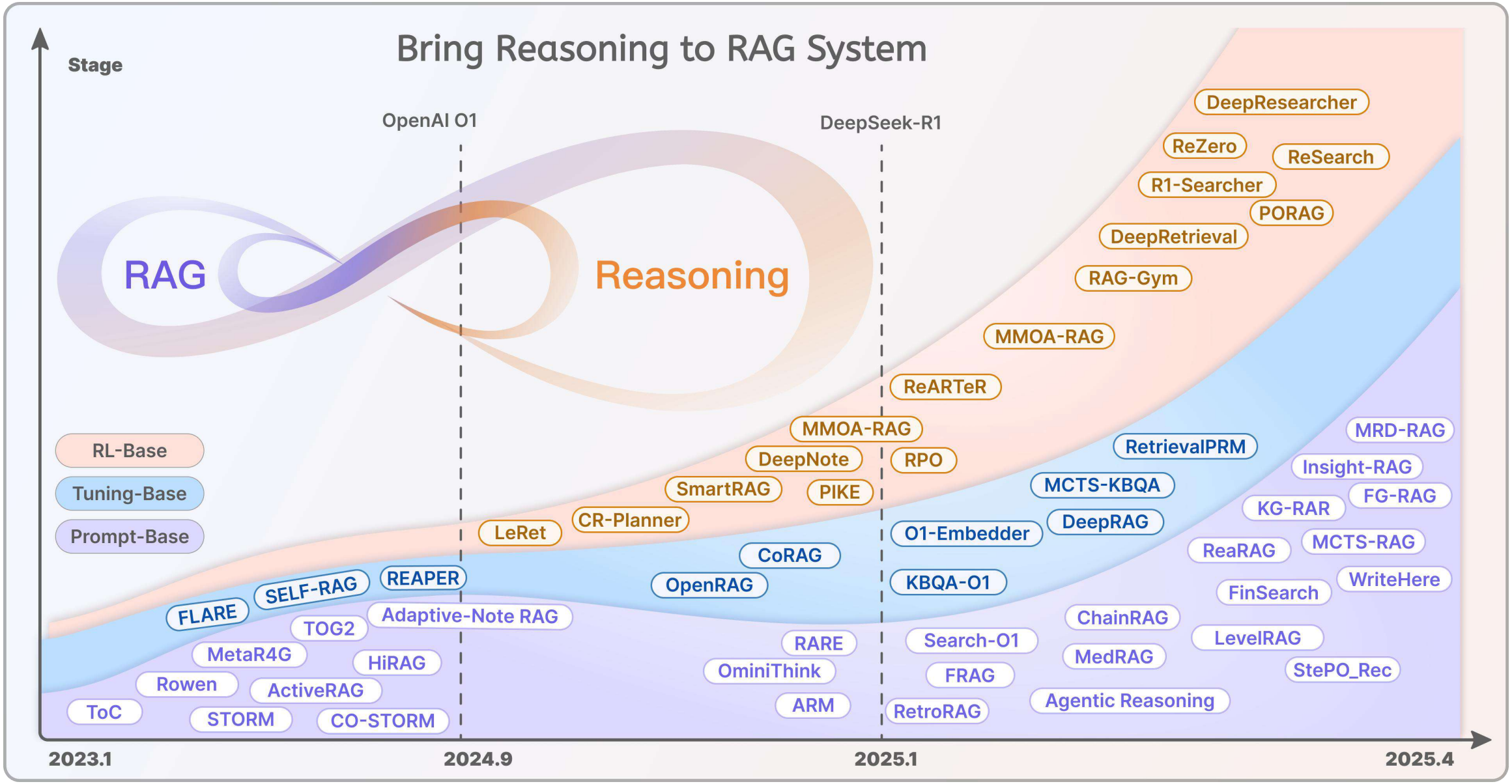
[1] QAnything引擎升级技术通告：DeepSeek-R1适配实践与效果验证.QAnything 官方公众号

[2] U-NIAH: Unified RAG and LLM Evaluation for Long Context Needle-In-A-Haystack, 2025.

[3] Using DeepSeek R1 for RAG: Do's and Don'ts. https://blog.skypilot.co/deepseek-rag/

RAG + Reasoning 是否成就新的CP？

在OpenAI O1 和DeepSeek-R1等慢思考模型兴起后，将RAG与推理能力结合的研究不断出现



Reasoning如何定义？ 和Inference的区别?

Reasoning（推理）

结构化的**多步骤**过程，动态分解复杂问题，生成中间假设，通过逻辑和基于证据的转换来迭代地完善解决方案。

$$R(x) = \Phi_1 \circ \Phi_2 \circ \ldots \circ \Phi_n(x)$$

特点：

- 多步骤拆解复杂问题
- 生成新知识或事实
- 目的性明确的过程
- 支持错误回溯与自我修正
- 状态转换与动态调整

VS

Inference（推断）

单步条件概率计算，直接从输入到输出的映射，缺乏中间状态和目标导向的迭代过程。

$$P(y|x) = \Pi_{t=1}^{T} P(y_t|x, y_{<t})$$

特点：

- 单步条件概率计算
- 直接输入–输出映射
- 缺乏自我纠错机制
- 没有显式的中间状态
- 无法进行动态检索优化

示例场景

数学推理中的方程重构、基于检索事件的时序推理、符号操作等场景。
例如：解决"A是否大于B"时，会将问题转化为多个关于A和B属性的子问题，然后通过多步推导得出结论。

关键区别

- **计算特征**：Reasoning协调多个推断调用，实现系统性错误修正；Inference仅为单步概率计算
- **状态管理**：Reasoning有明确的状态转换与记忆；Inference无状态转换
- **适用范围**：Reasoning适合复杂问题与多步决策；Inference适合直接映射任务
- **灵活性**：Reasoning可动态调整策略；Inference执行固定的映射

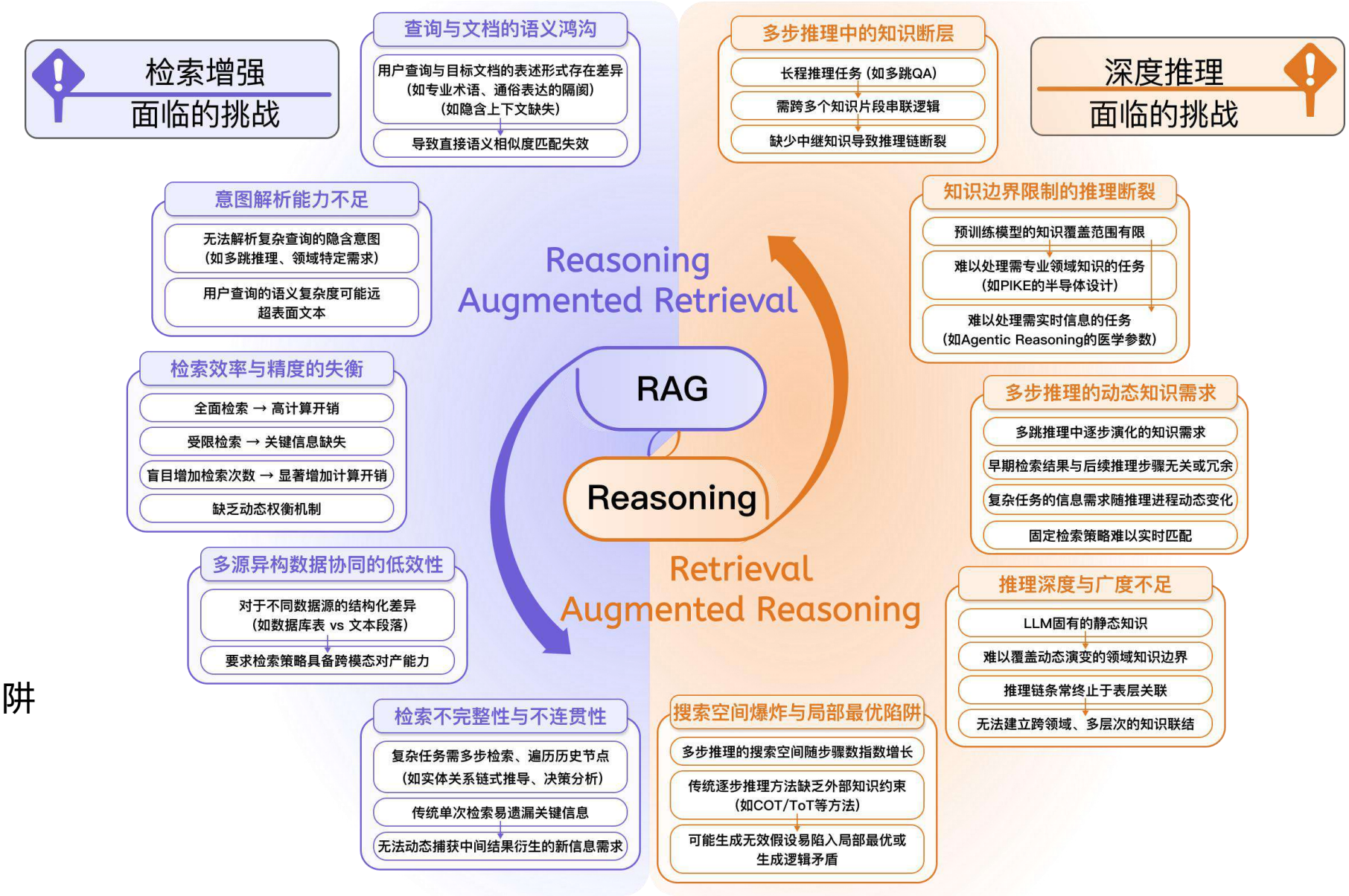
为什么我们需要RAG + Reasoning的协同

RAG的局限性

- 意图解析能力不足
- 查询与文档的语义鸿沟
- 多源异构数据协同的低效性
- 检索不完整性与不连贯性
- 检索效率与精度的失衡

Reasoning的局限性

- 多步推理中的知识断层
- 知识边界限制的推理断裂
- 搜索空间爆炸与局部最优陷阱
- 多步推理的动态知识需求
- 推理深度与广度不足



协同的目的 —— Reasoning-Augmented Retrieval

核心目的

通过整合多步推理能力来动态增强检索过程的质量，超越传统基于静态语义匹配的局限性。

传统RAG在复杂信息需求场景下的五大挑战：

- 静态触发机制难以判断何时需要检索
- 语义匹配无法捕捉深层查询意图和逻辑关系
- 单轮检索难以支持多跳信息依赖
- 缺乏对垂直领域知识结构的适应性
- 检索效率与精确度之间的矛盾难以平衡

五大核心增强能力

动态按需检索
通过推理评估意图明确性、知识状态和时间因素，指导自适应检索触发
例：UAR的分类器根据查询复杂度智能决定何时触发检索

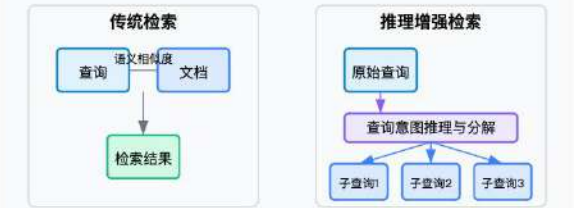
语义对齐增强
推断隐含的查询逻辑（如业务规则、实体关系），生成与数据模式对齐的精确检索请求
例：PlanRAG的计划-检索循环，将复杂查询转换为数据库可理解的形式

多步迭代精化
利用中间推理输出（如思维链、部分答案）递归重新构建查询，形成闭环系统
例：ITER-RETGEN使用中间答案重建后续查询，解析多跳依赖

垂直领域适应
根据特定领域（如金融、医疗）的知识结构和术语体系定制检索策略
例：FinSearch针对金融文档的时效性权重和多源融合机制

效率-精度平衡
根据查询复杂度智能分配检索资源，避免简单查询的过度检索和复杂查询的检索不足
例：AdaptiveRAG的轻量级分类器为查询分配不同检索预算

从模式匹配到逻辑驱动的检索转变



对比示例：医疗查询场景

传统检索：
查询"糖尿病患者术后感染风险"
→ 匹配"糖尿病术后护理"文档

推理增强检索：
分析因果关系和条件约束
→ 优先检索"血糖控制阈值"和"抗生素使用指南"

代表性技术实现

PlanRAG
使用迭代式"计划-回顾-重规划"循环来触发子查询

LevelRAG
基于复杂度的自适应路由，智能分配不同级别的检索资源

SmartRAG
强化学习驱动的决策优化，平衡检索质量和步骤数量

ChainRAG
推理链驱动的多阶段检索，解决跨文档的逻辑连贯性问题

从信息聚合到逻辑连贯的上下文构建

传统RAG系统直接输入所有检索文档片段，常导致信息碎片化或矛盾。推理增强系统通过：

- 逻辑验证检索内容并推断因果关系
- 过滤冲突并形成连贯解释
- 动态知识补全检测缺失的逻辑链接
- 提示迭代检索或推理填补知识空白

协同的目的 —— Retrieval-Augmented Reasoning

核心目的

通过引入外部知识弥补纯参数化LLM推理的局限性，使复杂推理过程更加可靠、准确和可解释。

即使是先进的大型推理模型（如OpenAI O1、DeepSeek-R1）在单独使用时仍面临：

- 幻觉生成风险
- 知识覆盖有限（时效性、专业性）
- 缺乏可验证的外部依据
- 复合推理中的组合挑战

典型应用场景

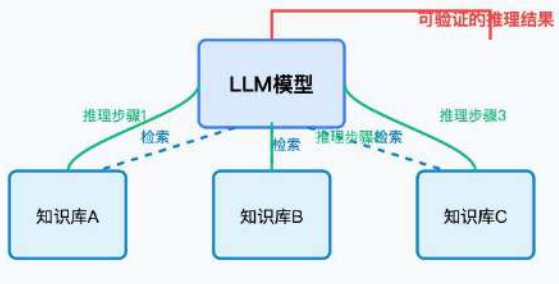
医疗诊断决策
MedRAG整合最新医学文献与临床指南，确保诊断推理基于最新证据

深度研究与分析
DeepResearcher打破多来源信息孤岛，实现跨文档的逻辑推演

复杂问题解决
RARE实现在步骤证明期间动态检索相关定理和公式

实现方式与优势

- **知识更新与时效性**
  检索提供最新信息，克服模型参数中固有的知识过时问题
- **推理过程可解释性**
  每个推理步骤可关联到具体的外部知识源，增强结果可信度
- **复杂多步推理支持**
  支持推理过程中动态检索所需信息，解决中间步骤知识缺口
- **领域专业性提升**
  能够检索高度专业的领域知识，超越模型训练数据覆盖范围
- **减少幻觉生成**
  使用检索到的事实作为"基准线"校准模型内部知识



代表性技术方案

ActiveRAG
通过三阶段流程（自我询问→知识吸收→思维调整）结构化理解和校准所检索的知识

DeepRAG
使用强化学习优化推理路径，自动决定何时进行检索以及检索何种信息

ReARTeR
实施信任感知奖励策略，在检索结果可靠性和推理过程质量之间取得平衡

CR-planner
将检索和推理集成到统一规划框架中，针对每个子问题动态决策采用何种方式

RAG + Reasoning 潜在收益

当我们将推理能力与RAG系统结合以后，期望的收益是什么，和之前的RAG系统相比与有什么提升。

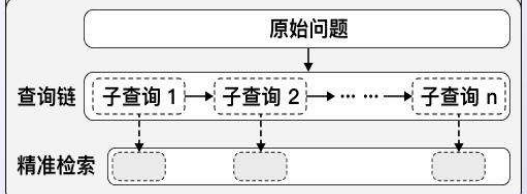
模糊语义匹配 ① **逻辑驱动的定向精准检索**

- 依赖相似度匹配
- 受模糊问题影响
- 受文档表述差异影响

问题 —相似?/相关?→ 回答

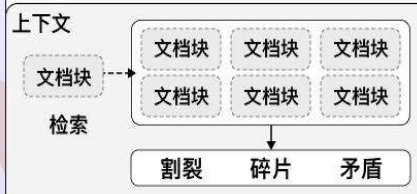
语义相似 ≠ 逻辑相关

- 解析逻辑结构
- 构建检索链
- 分析问题
- 更强的多跳推理能力



信息堆砌 ③ **逻辑自洽的上下文建构**

- 独立、语义割裂、碎片化
- 相互矛盾的文档块
- 干扰LLM生成
- 产生幻觉和逻辑断层的回答



- 与推理协同
- 实现证据链整合
- 识别缺少信息
- 动态知识补全

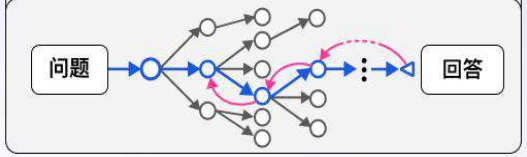
单一问答 ② **系统性决策支持**

- 面向密集任务
- 擅长实时性回答
- 缺少多步推理决策支持

问题 → 回答

单次问答

- 支持复杂决策
- 实现结构化推理输出
- 多目标优化能力
- 平衡多重约束



RAG

Reasoning

盲目检索 ④ **智能资源分配**

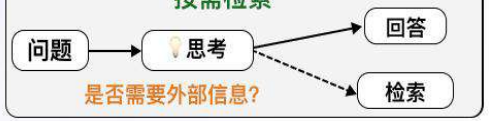
- 检索是必要环节
- 导致不必要的开销和时延
- 潜在信息冲突和干扰

固定的检索环节

问题 → 检索 → 答案

- 动态检索
- 区分简单与复杂问题
- 动态剪枝
- 减少无效信息遍历

按需检索



- 被动响应的知识库
- 只能接受用户的查询
- 单轮交互
- 缺少长期记忆

被动知识助手 ⑤ **认知伙伴**

- 主动的服务和引导
- 融合记忆、推理和决策能力
- 预判潜在(隐式)需求
- 类人助手

05 RAG与深度推理能力协同

5-1. RAG+Reasoning的新趋势 | 5-2. RAG+Reasoning协同的实现 | 5-3. RAG+Reasoning协同的优化

LLM思维链

思维链核心价值

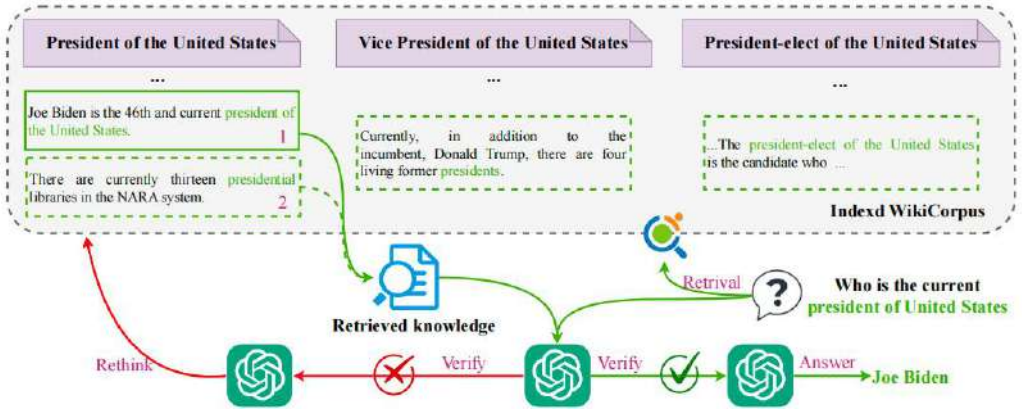
- 引导多步骤推理过程
- 动态整合外部知识
- 提高复杂推理任务表现

共同优势

- 将复杂问题分解为清晰中间步骤
- 通过推理状态引导外部知识选择
- 增强LLM推理及适应外部知识能力

HiRAG: 分层检索与反思机制

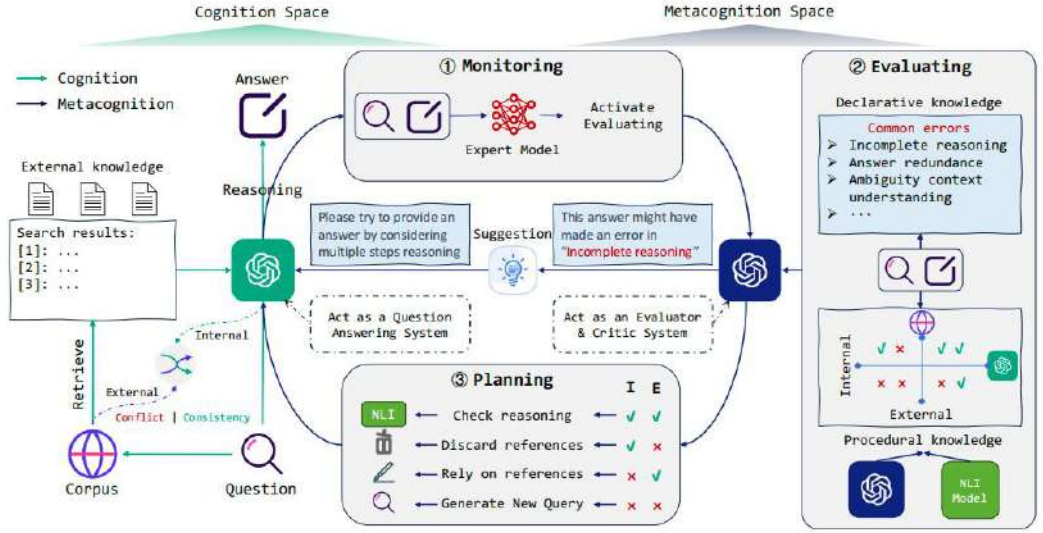
- 分解器：基于CoT拆解问题为多步（例：逐步分析生成子问题）
- 定义器：判断信息充分性，控制流程终止（例：依据子答案判定可答性）
- 过滤器：验证结果可信度 + 反思循环，动态调整检索策略



Hierarchical Retrieval-Augmented Generation Model with Rethink for Multi-hop Question Answering.2024

MetaRAG: 元认知驱动的多阶段

- 将心理学元认知理论编码为可执行的Prompt逻辑链
- 角色分离提示（QA系统 vs 评估-批判系统）
- 动态查询生成模板（"To answer this, I need to search..."）



Metacognitive Retrieval-Augmented Large Language Models. WWW 2024

特殊Token预测

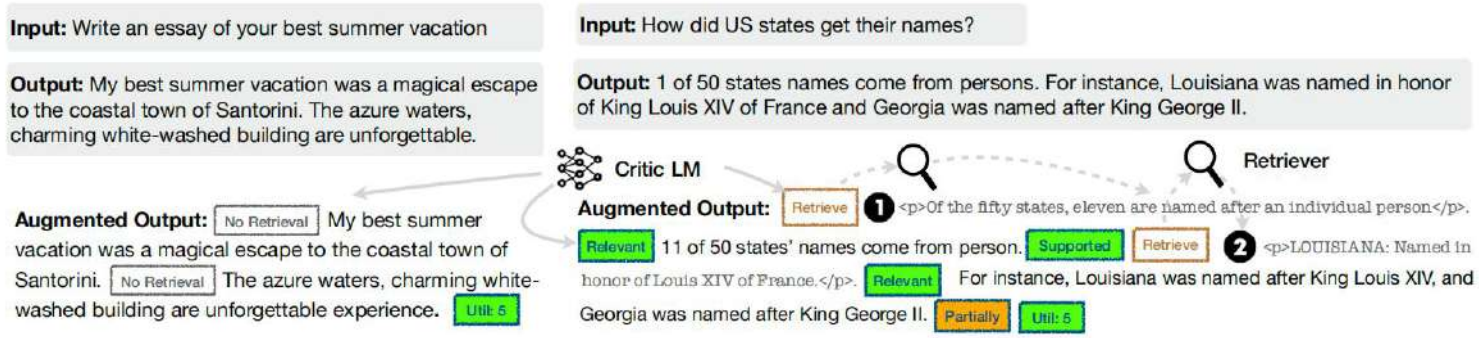
特殊标记功能

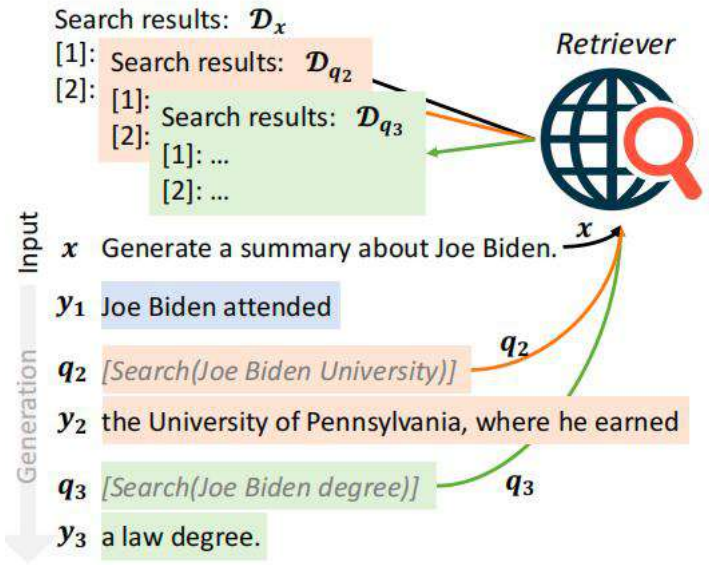
- 实现检索与多步推理动态连接
- 触发外部工具或自我反思
- 管理检索激活与相关性检查
- 实现输出验证与知识接地

Self-RAG
使用'Retrieve'、'ISREL'标记管理检索和验证

SmartRAG
使用'[RETRIEVE]'标记控制检索流程

Open-RAG
结合标记控制与专家混合(MoE)路由





相比于传统方法的优势

更精细控制
通过Token序列显式编码决策逻辑

更高的可解释性
标记序列映射决策过程，便于理解

缓解延迟问题
实现上下文感知、按需工具使用

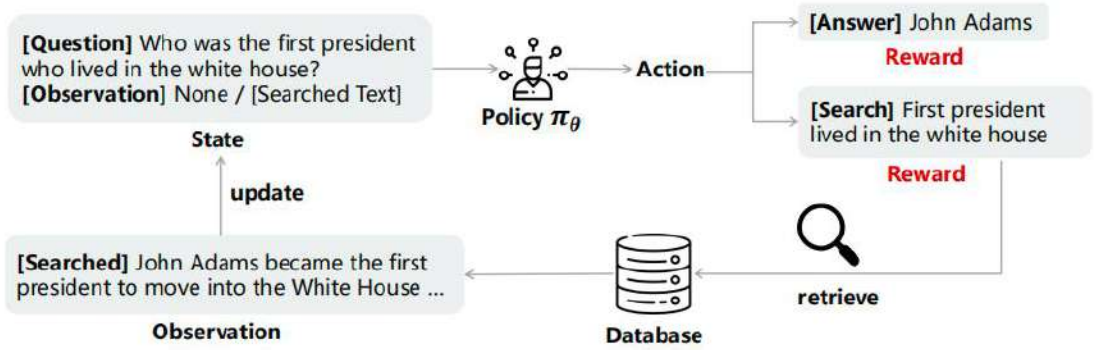
核心创新

创新在于在生成序列内预测这些标记，将任务分割为检索启动、文档评估和知识接地阶段，将静态推理链转变为条件性工作流。

特殊Token预测

SmartRAG: 多角色策略网络

单一LLM策略网络预测特殊Token [RETRIEVE] / [ANSWER]:
决策：首Token预测决定是否检索；
查询重写：若需检索，后续Token生成优化后的搜索查询；
答案生成：直接生成最终答案。



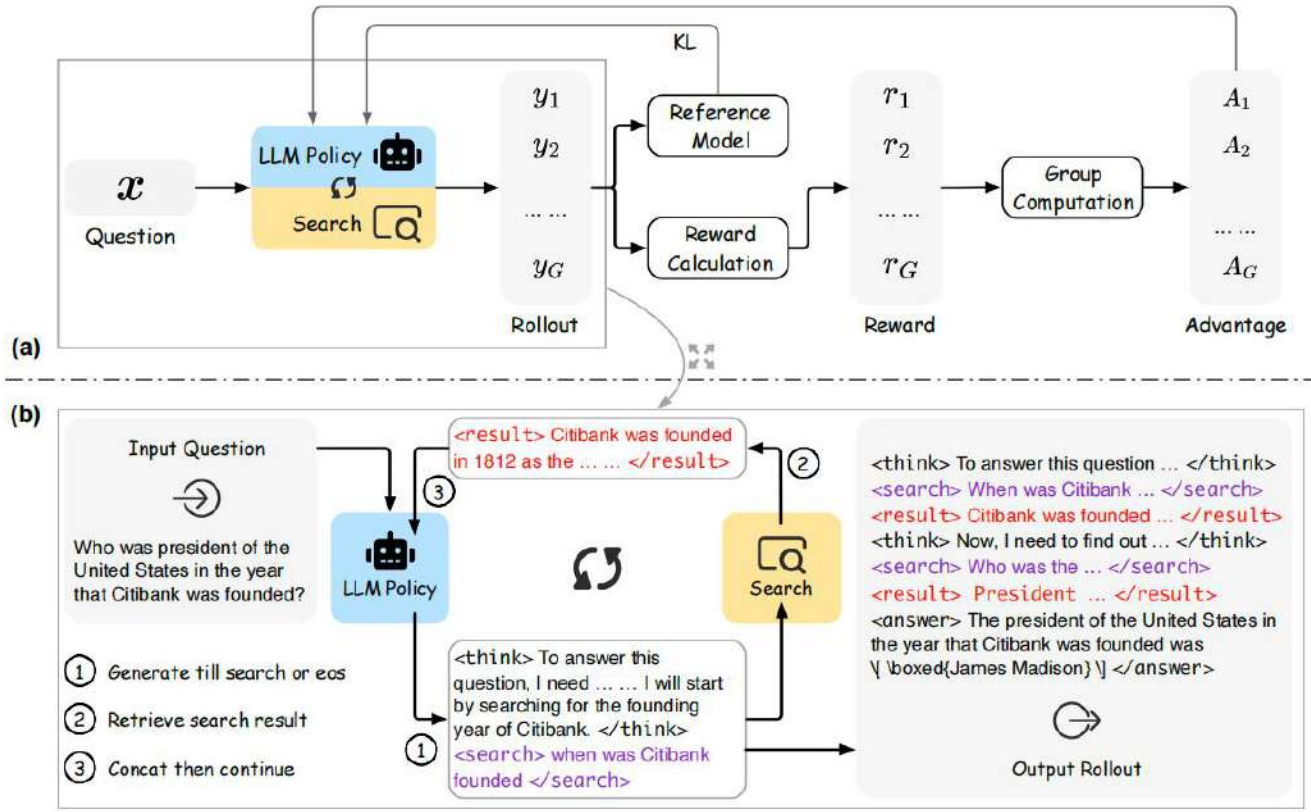
Algorithm 1 SmartRAG Pipeline

Require: Policy Network π_θ, Retriever $\mathcal{R}$
Input: input question $\mathbf{x}$, retrieve quota N, observation $\mathbf{os} \leftarrow [\,]$, retrieve count $n \leftarrow 0$
while $n \leq N$ **do**
  if $n = N$ **then**,
    $\mathbf{a} \sim \pi_\theta([\mathbf{x}, \mathbf{os}])$ s.t. $\mathbf{a}_0 =$ [ANSWER]
  else
    $\mathbf{a} \sim \pi_\theta([\mathbf{x}, \mathbf{os}])$
  $n \leftarrow n + 1$
  if $\mathbf{a}_0 =$ [RETRIEVE] **then**,
    $\mathbf{o} \leftarrow \mathcal{R}(\mathbf{a}_{1:}), \quad \mathbf{os} \leftarrow [\mathbf{os}, \mathbf{o}]$
  else
    return $\mathbf{a}_{1:}$

SmartRAG: Jointly Learn RAG-Related Tasks From the Environment Feedback. 2025.

ReSearch: 结构化预测推理链

特殊Token结构化推理链：
采用<search>/</search>标记搜索动作，<result>/</result>包裹检索结果，<think></think>引导文本推理，\boxed{}标识最终答案，形成结构化多模态推理链。



ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning , 2025.

基于树搜索的方法

主要特点

基于Search的方法通过动态搜索策略增强RAG与推理的结合，将复杂问题分解为可搜索的子问题，形成逻辑决策树。

- ✓ 从简单语义匹配转向逻辑驱动的定向检索
- ✓ 支持多跳检索，通过推理链保持跨文档连贯性
- ✓ 使用自适应信息约束管理复杂推理过程

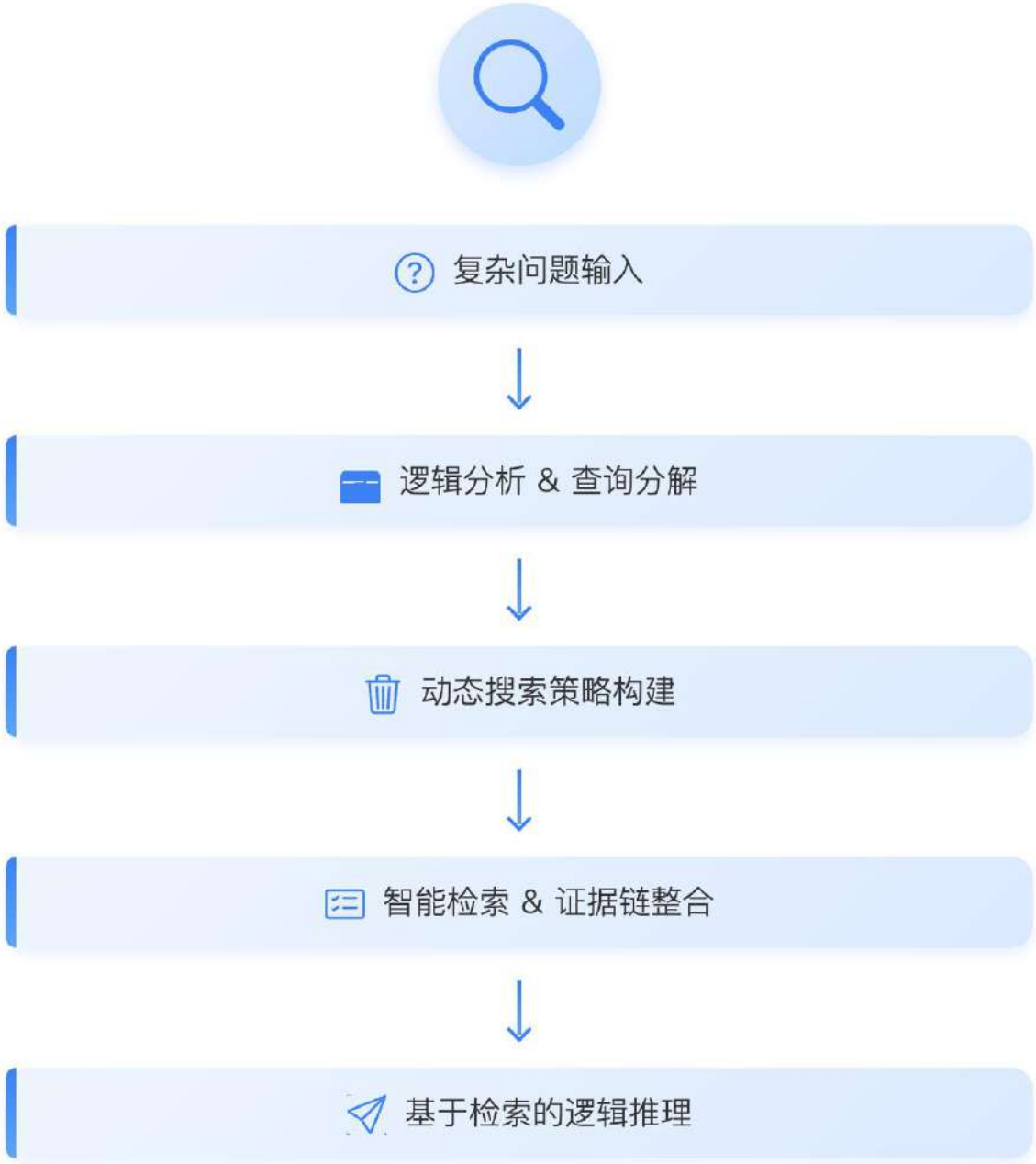
代表性框架

DeepRAG：深度分析查询，通过因果关系和条件约束动态优化检索策略

OmniThink：运用搜索驱动的多步推理，实现对复杂问题的分解与求解

CoRAG：协作式推理与检索框架，实现智能信息获取与整合

Search-O1：代理式搜索增强大型推理模型，改进复杂推理能力



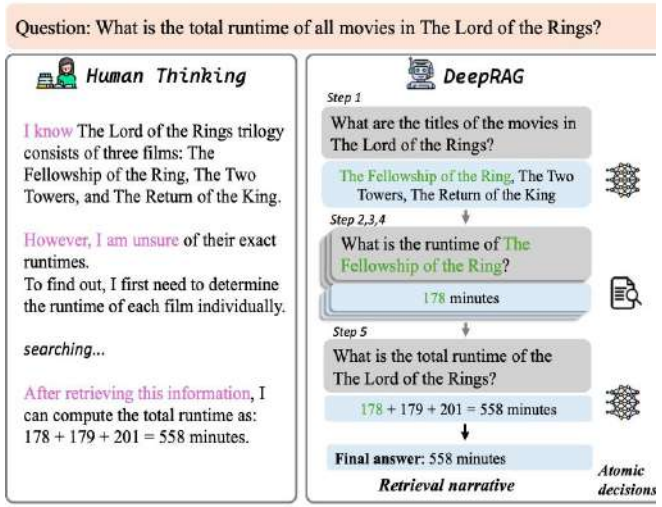
基于树搜索的方法

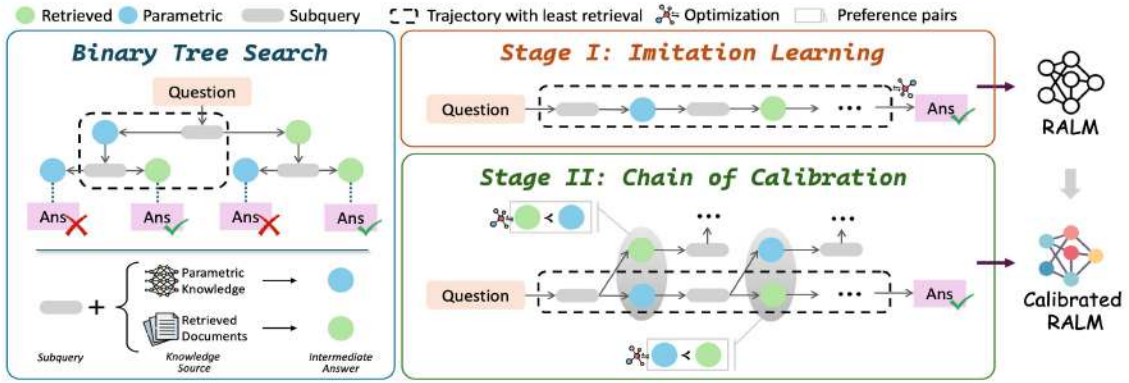
DeepRAG: 二叉树搜索

动态检索决策：将RAG建模为马尔可夫决策过程（MDP），通过状态转移动态决定是否检索外部知识或依赖模型内部知识。

模仿学习：让模型学会以最小化检索成本（如次数）将复杂问题分解为子查询序列。

校准链（Chain of Calibration）：通过偏好数据优化原子决策，提升模型对自身知识边界的认知。





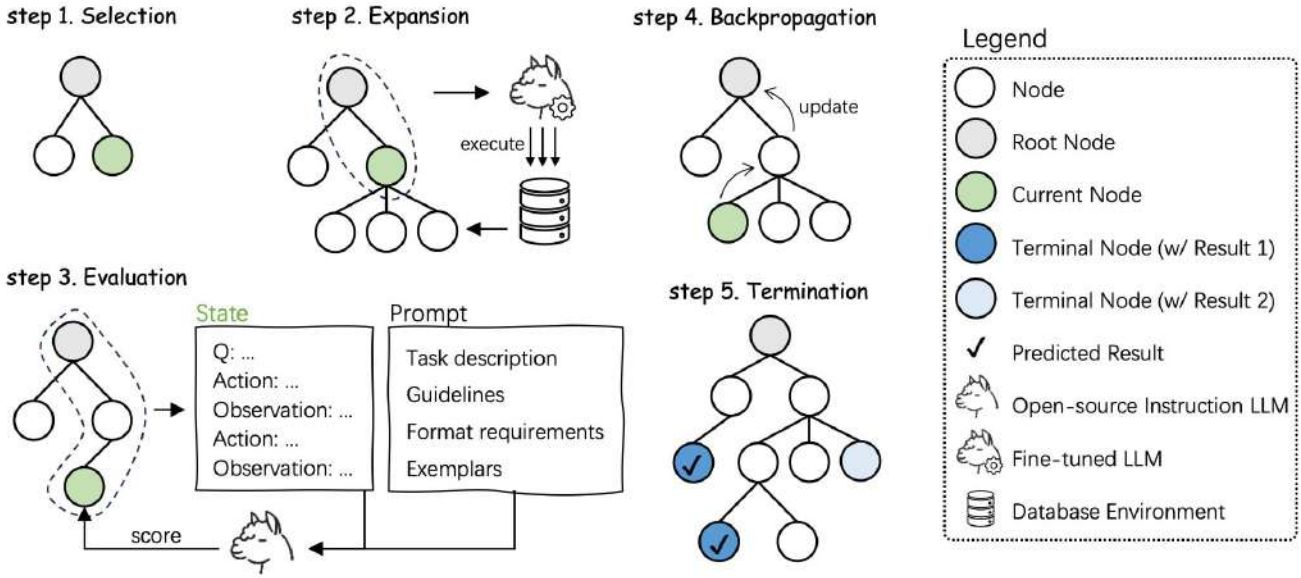
DeepRAG: Thinking to Retrieval Step by Step for Large Language Models, 2025.

MCTS-KBQA: 蒙特卡洛树搜索

MCTS与KBQA结合：借助MCTS的选择、扩展、评估、模拟和反向传播步骤，对解决方案空间进行有效探索，提升推理效率。

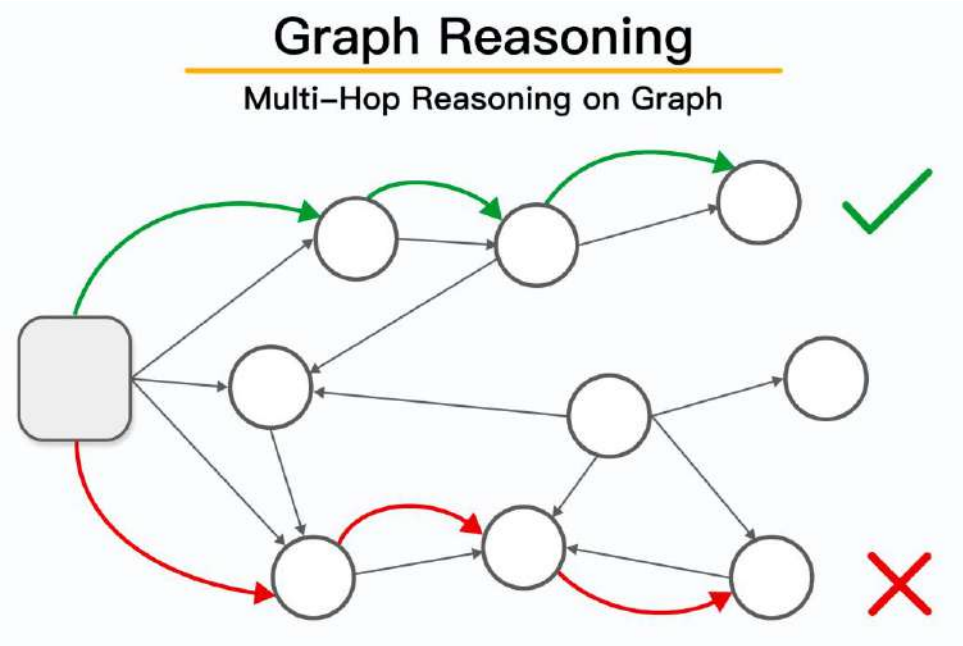
分步奖励机制：依据KBQA任务特点，设计规则提示文本，评估中间状态是否成功识别元素和解决子问题。

中间推理标注数据：运用远程监督技术，为现有的问题 - SPARQL数据集标注中间推理过程，助力低资源场景下模型性能提升。



MCTS-KBQA: Monte Carlo Tree Search for Knowledge Base Question Answering, 2025.

图上的多跳推理



代表性框架

HippoRAG2

构建开放知识图谱并使用个性化PageRank与密集–稀疏编码方法，增强事实记忆、语义理解和多跳推理能力

Think-on-Graph (ToG)系列

结合知识图谱和文档迭代检索，使用关系发现、实体修剪和上下文驱动的图搜索，增强隐式关系检测能力

Agentic Reasoning

将推理链转化为图结构，用于实体提取、关系识别和社区聚类，支持动态路径跟踪和优化检索

四大应用领域

- 符号推理：利用知识图谱进行多跳推理
- 任务规划：构建动态知识图谱支持复杂问题求解
- 工具使用与管理：捕获工具间复杂依赖关系
- 多源信息整合：从事实检索到全局主题总结

核心优势

显式建模逻辑依赖	支持多跳知识整合
增强关系推断能力	支持反事实分析

外部求解器

核心思想

基于外部求解器的方法将RAG系统与专用求解器（如数学求解器、规则推理引擎、符号逻辑处理器等）相结合，增强复杂推理能力。

- ✓ 解耦推理过程与知识获取，提高系统可靠性
- ✓ 专业求解器处理特定领域问题，降低幻觉风险
- ✓ 支持可解释性推理，提供明确的推理路径

典型应用场景

数学问题求解：结合符号数学引擎处理复杂数学推导和证明

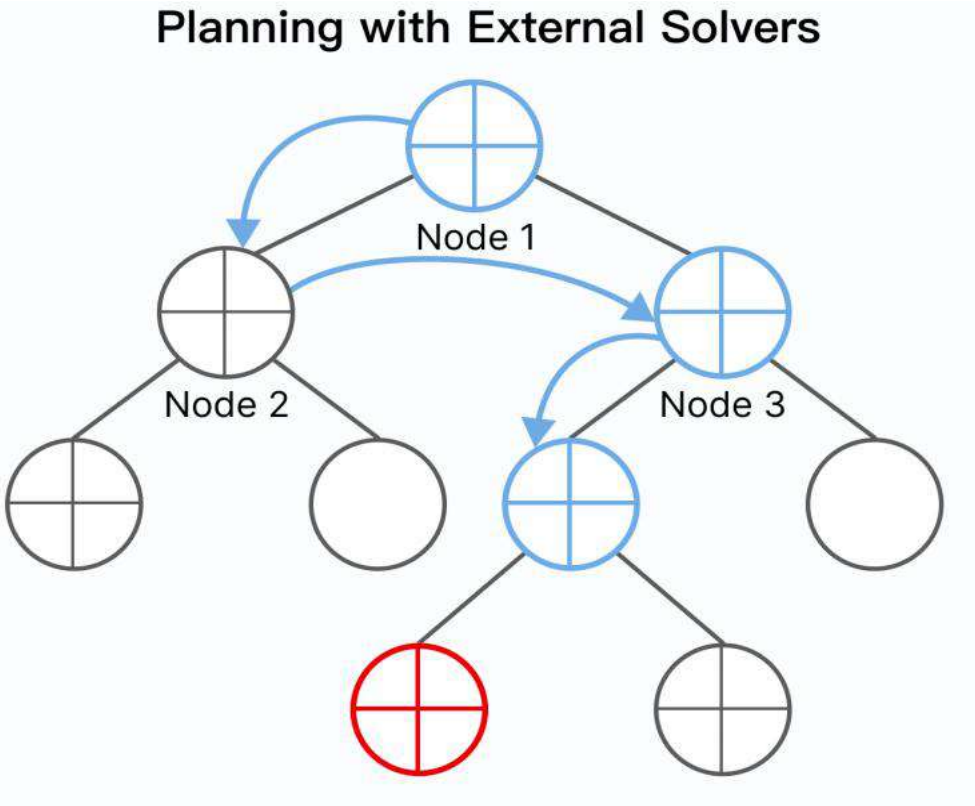
金融分析：整合专业计算模型分析财务数据和市场预测

法律咨询：使用规则推理系统进行法条解析与案例分析

外部求解器工作流程示例

ARM框架通过以下步骤整合外部求解器：

1. 问题分析 → 确定适用求解器类型
2. 检索相关知识 → 构建求解模型
3. 转换为求解器可处理的形式
4. 外部求解 → 获取结果
5. 结果整合与解释



代表性实现与方法

ARM：整合符号代数求解器，增强RAG系统处理数学推理的能力。可以解决方程组、微积分及优化问题。

规则引擎集成：如FinSearch在金融领域中，将领域规则与检索系统集成，支持复杂条件下的监管合规性评估和风险分析。

外部求解器 —— ARM: 一次性检索所有必要信息

研究背景与问题

- 复杂问题回答通常需要来自**多个信息源**的信息
- 现有RAG方法存在问题：
  - 标准RAG中的问题分解不了解数据集组织结构
  - 智能体式RAG检索效率低，需要多次LLM调用
- 存在关键挑战：如何高效且全面地检索所有必要信息

研究目标

ARM旨在通过更好地**对齐问题与数据集组织**，构建一种能够一次性检索所需全部信息的高效方法。

ARM核心方法

信息对齐

分解原始问题为关键词，并与数据集中已有对象的N-gram进行匹配

通过约束解码引导LLM重新表述关键词

结构对齐

使用推理求解器识别数据对象之间的关系，探索完整连接

使用混合整数规划(MIP)优化连接数据对象

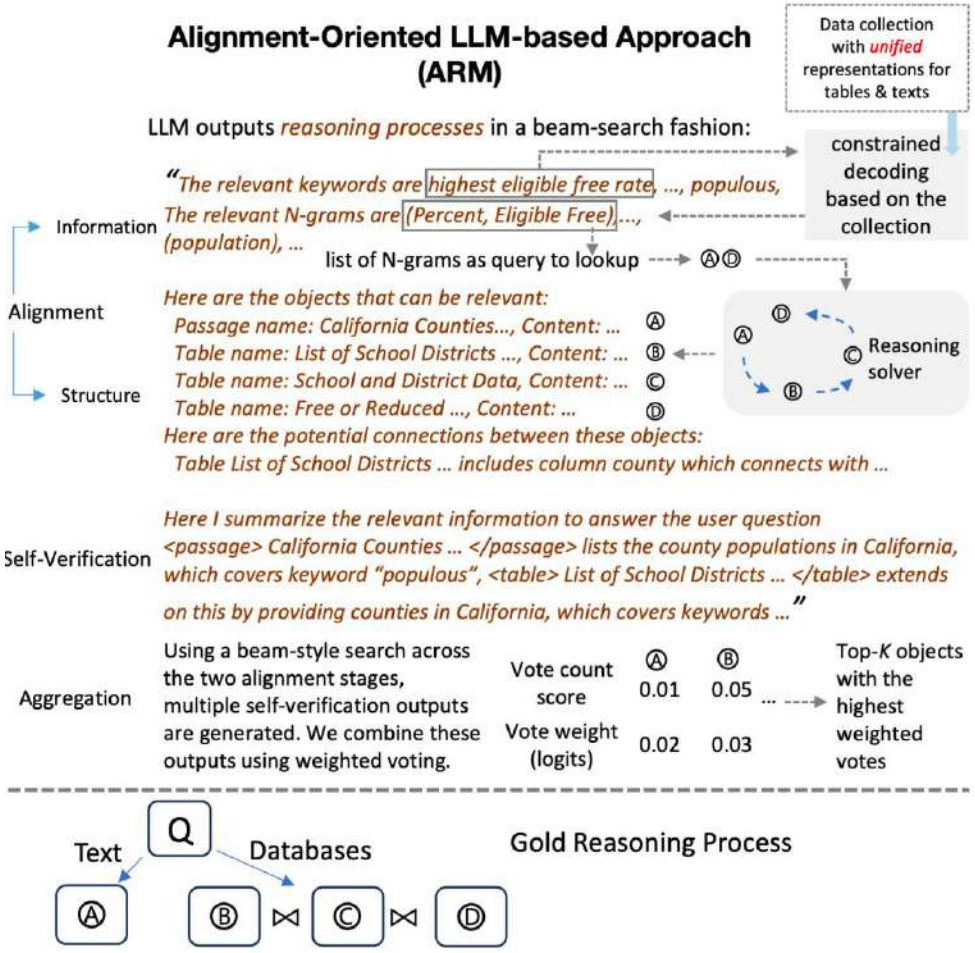
自验证与聚合

LLM验证检索对象相关性及连接，选择最终答案所需数据

通过beam search生成多个推理过程并聚合

ARM的核心思想

通过对数据集结构的理解，探索潜在信息源之间的关系，实现高效检索并减少LLM调用次数。

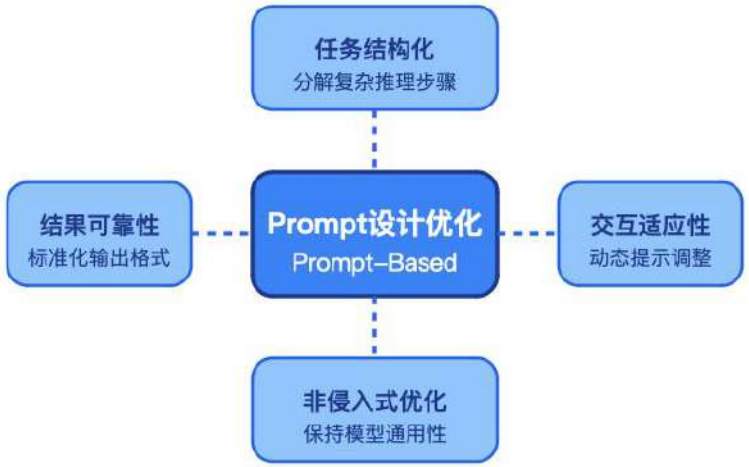


Can we Retrieve Everything All at Once? ARM: An Alignment-Oriented LLM-based Retrieval Method. 2025

05 RAG与深度推理能力协同

5-1. RAG+Reasoning的新趋势 | 5-2. RAG+Reasoning协同的实现 | 5-3. RAG+Reasoning协同的优化

基于提示工程的优化



任务结构化设计

通过精心设计的自然语言提示，将复杂推理任务分解为可管理的步骤，引导LLM在生成过程中遵循特定的逻辑结构。Co-STORM和WriteHere等技术使用角色分配、阶段划分和操作特定指令来指导多步推理，包括建议生成、知识检索、细化和验证，通过清晰表示中间步骤提高可解释性。

结果可靠性提升

通过标准化输出和减少幻觉来提高结果可靠性。策略包括要求引用检索结果、强制执行特定输出格式，以及基于检索知识集成反思和校准。FinSearch和ActiveRAG等系统通过提示融入时间加权、去重和领域规则，提高一致性和逻辑连贯性，特别在复杂领域中效果显著。

代表方法与实现技术

Co-STORM：多角色协作系统，生成跨模态检索命令

WriteHere：使用结构化提示指导文档生成流程

FinSearch：金融领域专用系统，整合时间加权与去重

ActiveRAG："自我询问→知识吸收→思想调整"链

PlanRAG：生成可执行多步计划，通过闭环反馈调整

Agentic Reasoning：使用上下文敏感提示和反馈循环

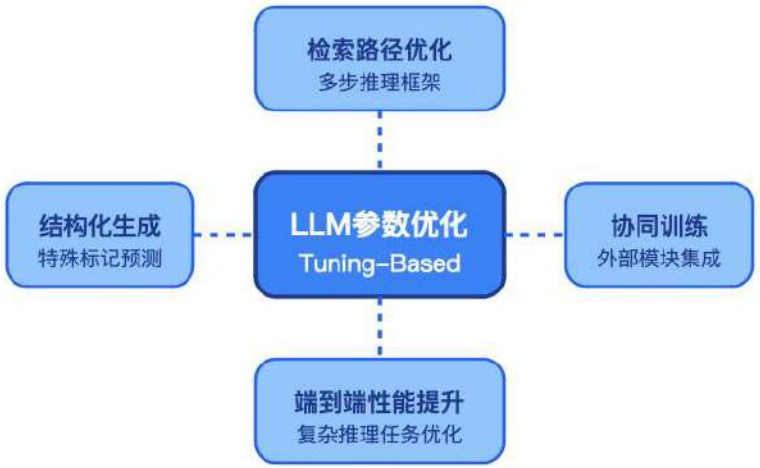
交互适应性

基于Prompt的方法允许动态提示调整。特殊标记（如<Search>, [Web-search]）使模型能够基于中间结果实时触发工具或修改查询。Agentic Reasoning和PlanRAG等方法使用上下文敏感提示和反馈循环动态优化推理路径，在多跳任务中保持连贯性和准确性，在复杂、不断变化的场景中优于传统RAG方法。

优势与适用场景

基于Prompt的优化是增强RAG+Reasoning的高效、灵活和可靠方法，其非侵入式设计使其成为优化LLM推理的主流策略，并作为未来集成微调和强化学习的混合方法的基础。通过语义结构、动态反馈和符号约束系统地优化推理而无需改变模型参数，该范式有效管理任务分解和知识集成等宏观级控制，同时解决生成一致性、逻辑连贯性和外部知识对齐等关键挑战。

基于微调的优化



检索路径优化

CoRAG和DeepRAG等方法通过全参数微调和多任务学习构建端到端多步推理框架。CoRAG将单步QA数据集扩展为检索-推理链，并联合训练子查询生成、中间答案预测和最终组合等任务，增强模型对复杂问题的分解能力。

结构化生成增强

通过元学习和知识蒸馏技术增强模型对结构化数据的理解和生成能力。O1-Embedder使用低秩自适应技术改进对长文本的表示能力，KBQA-O1采用知识图谱辅助微调，通过结构感知子任务优化复杂查询解析。

关键代表方法

CoRAG：扩展单步QA数据集为检索-推理链，联合训练多个子任务

DeepRAG：端到端多步推理框架，通过全参数微调优化检索路径

O1-Embedder：采用低秩自适应技术提升长文本表示能力

KBQA-O1：结合知识图谱辅助微调，优化复杂查询解析

RetrievalPRM：使用参数高效微调技术，降低训练成本

与外部模块协同训练

微调方法同时关注LLM与外部模块（如检索器、排序器）的协同训练，实现系统级优化。UAR对重排模块和生成模块联合训练，REAPER引入多阶段训练方法，有效提高知识密集型任务的性能。RetrievalPRM采用参数高效微调技术，在保持高性能的同时显著降低训练成本。

优势与应用价值

基于微调的方法能将检索增强思维链机制内化到LLM参数中，实现更紧密的检索-推理集成。这些方法在多跳问答和知识密集型任务中表现优异，比传统RAG方法提升10-25%的准确率。适用于医疗诊断、法律分析等专业领域，能够降低推理过程中的错误传播风险。

基于强化学习的优化

结果导向奖励模型(ORM)

ORM范式专注于最终输出的质量及其对标准的符合程度。以R1-Searcher为例，该方法采用两阶段Reinforce++训练，第一阶段奖励取决于正确的检索调用和特殊标记生成，第二阶段直接优化答案的F1分数。这鼓励模型开发策略，最大化知识集成，减少幻觉，并增强多跳QA中的准确性。同样，KBQA-O1使用MCTS和策略网络评估逻辑一致性，有效平衡知识库QA中的探索和利用。

过程导向奖励模型(PRM)

PRM强调对中间推理步骤的详细监督。LeReT使用身份策略优化(IPO)算法，通过奖励检索文档的平均精度(AP)来优化查询质量，提升检索召回率和多跳任务整体性能。ReARTeR扩展了这一方法，使用步级二元奖励模型，结合蒙特卡洛评分和时间差分(TD)方法主动评估推理路径，减少逻辑错误和冗余检索，并提高HotpotQA等基准测试的准确性。

关键算法与实现

GRPO：抛弃评论家模型的PPO变体，从组分数估计基线，大幅减少训练资源

ReZero：使用GRPO引入"重试"机制，通过奖励重试搜索查询来鼓励LLM在初次搜索失败后继续尝试

PORAG：通过双重奖励机制(检索忠实度和响应质量)直接优化检索质量、上下文相关性和生成连贯性

MCTS：蒙特卡洛树搜索技术，如KBQA-O1使用的策略网络与奖励模型组合

IPO：身份策略优化，LeReT中用于优化查询质量的新型算法

方法	基础模型	RL类型	参数调整	奖励类型
PORAG	Qwen2.5/Llama3.2	GRPO	QLoRA	ORM
ReZero	Llama3.2-3B	GRPO	全参数	ORM+PRM
R1-Searcher	Qwen2.5/Llama3.1	Reinforce++	全参数	ORM
KBQA-O1	Llama3系列	MCTS	DoRA	ORM+PRM

混合奖励策略

近期研究展示了混合ORM和PRM的优势。DeepRAG采用成本感知准确性指标($R = -C(o) \times T(st)$)，其中C(o)表示答案正确性，T(st)表示总检索成本，通过PPO式校准和对比学习优化推理行为。RAG-Gym使用三重标准(充分性、实用性、冗余性)构建全面奖励函数，通过SFT和DPO技术，指导模型生成高质量步骤。CR-Planner利用评论家估计的奖励(步骤正确性和全局影响)，结合配对排名损失，实现复杂数学推理中高效解决方案的优先级排序。

RAG + Reasoning中的强化学习方法与奖励函数设计

主要奖励模型范式

基于结果的奖励模型 (ORM)

专注于最终输出质量和标准遵循度的评估

- 通常使用F1分数、准确率等指标评估最终答案
- 例如：R1-Searcher使用F1分数+格式惩罚作为奖励
- 优点：引导发现全局最优策略

基于过程的奖励模型 (PRM)

着重于中间推理步骤的详细监督

- 评估每个推理步骤的质量和贡献
- 例如：LeReT使用检索文档的平均精度(AP)作为奖励
- 优点：通过局部优化增强推理鲁棒性

混合奖励模型 (ORM+PRM)

结合结果和过程双重奖励机制

- 同时优化最终输出和中间过程
- 例如：DeepRAG使用成本感知准确性奖励
- 优点：平衡全局目标和局部优化

典型RL算法应用

GRPO (群体相对策略优化)

放弃critic模型，从组得分估计基线，大幅减少训练资源

代表工作: PORAG, ReZero, ReSearch

MCTS (蒙特卡洛树搜索)

通过模拟未来状态评估当前行动价值

代表工作: KBQA-O1, MCTS-KBQA, CR-Planner

PPO (近端策略优化)

通过约束策略更新步长防止过度优化

代表工作: SmartRAG, DeepRetrieval

DPO (直接偏好优化)

通过对比学习直接从偏好数据优化策略

代表工作: DeepNote, RAG-Gym

Reinforce++ & IPO

基于策略梯度的改进算法

代表工作: R1-Searcher, LeReT

RAG + Reasoning奖励函数设计

PORAG 双重奖励机制

公式: R = αRfid + βRqual

- Rfid: 检索忠实度 - 评估检索内容相关性
- Rqual: 响应质量 - 评估生成内容连贯性
- α,β: 平衡两个奖励组件的权重

DeepRAG 成本感知奖励

公式: R = -C(o) × T(st)

- C(o): 答案正确性评分
- T(st): 总检索成本
- 平衡准确性与效率的权衡

RAG-Gym 三重标准评估

综合三个维度的奖励:

- 充分性: 检索知识是否足够回答问题
- 实用性: 检索内容对答案生成的贡献
- 冗余性: 避免过度检索和重复信息

R1-Searcher 两阶段奖励

分阶段设计的奖励函数:

- 第一阶段: 检索次数+格式符合度
- 第二阶段: F1分数+格式惩罚
- 促进模型发展最大化知识整合的策略

Table 1. Comparison of RL-based RAG with Reasoning Methods

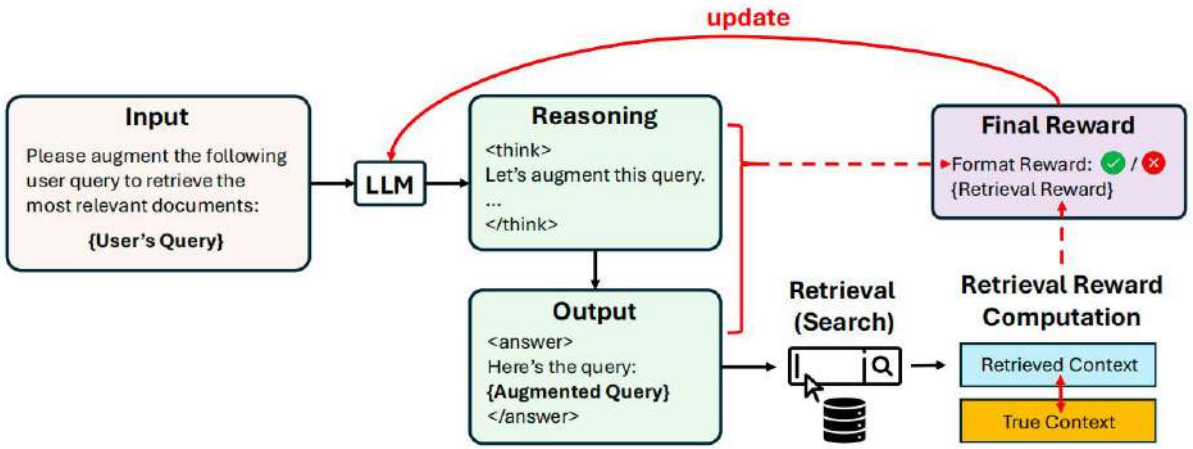
Method	Base Model	RL	Parameter	Supervision	Reward Function	Policy Strategy
PORAG [73]	Qwen2.5/Llama3.2	GRPO	QLoRA	ORM	Dual rewards: 1. Retrieval fidelity (R_{fid}) 2. Response quality (R_{qual}) Combined: $R = \alpha R_{\text{fid}} + \beta R_{\text{qual}}$	• Group-based advantage normalization • PPO-style clipped objective • KL regularization
DeepResearcher [106]	Qwen2.5-7B	GRPO	Full	ORM	Format compliance penalty (-1) + Answer F1 score	• Reference policy constraints • KL divergence penalty
ReSearch [6]	Qwen2.5-7B	GRPO	Full	ORM	Hybrid rewards: • Answer F1 (vs ground truth) • Format compliance check	• GRPO with clip ratio 0.2 • Group advantage normalization (G=5) • $\beta = 0.001$ KL penalty
ReZero [16]	Llama3.2-3B	GRPO	Full	ORM+PRM	• Answer correctness • Format compliance • Search diversity • Chunk matching • Retry behavior • Strategy compliance	• Intra-group reward comparison • Noise-injected robustness training • KL constraints
MMOA-RAG [12]	Llama-3-8B	MAPPO	Full	ORM	Shared F1 reward + penalties: • Excessive sub-questions • Document ID errors • Answer verbosity	• MAPPO actor-critic updates • Cosine learning rate scheduling
DeepNote [84]	Qwen2.5/Llama3.1	DPO	Full	ORM	Implicit preference modeling via likelihood contrast	• Direct Preference Optimization • Preference gap maximization
R1-Searcher [72]	Qwen2.5/Llama3.1	Reinforce++	Full	ORM	Two-stage rewards: 1. Retrieval count + format 2. F1 score + format penalty	• RAG-based rollout • Retrieval-masked loss
KBQA-O1 [58]	Llama3/Qwen2.5/Gemma2	MCTS	DoRA	ORM+PRM	Composite reward: • Stepwise policy model score • Final reward model score	• MCTS trajectory optimization • Q-value backpropagation
DeepRetrieval [42]	Qwen2.5-3B	PPO	Full	ORM	Task metrics: • Recall@k/NDCG • Syntax validity	• GAE advantage estimation • Distributed HybridFlow
LeReT [34]	Llama3-8B/Gemma-9B	IPO	Full	PRM	Average Precision (AP) of retrieved documents	• Identity Policy Optimization • Context distillation
SmartRAG [20]	Flan-T5-L/Llama2-7B	PPO	Full/LoRA	ORM	Action-specific: • EM+F1 for answers • Cost penalty for retrievals	• On-policy sampling • PPO updates
ReARTeR [75]	LLaMA3.1-8B	MCTS	LoRA	ORM+PRM	Monte Carlo step scoring + TD look-ahead	• Iterative preference optimization • KTO loss
DeepRAG [24]	Qwen2.5-7B/Llama3.1-8B	Hybrid	Full	ORM+PRM	Cost-aware accuracy: $R = -C(o) \times T(s_t)$ $C(o)$: Answer correctness $T(s_t)$: Total retrieval cost	• Imitation + contrastive learning • PPO-like calibration
RAG-Gym [96]	LLaMA3.1-8B	Hybrid	LoRA	PRM	Triple criteria: • Sufficiency • Utility • Redundancy	• SFT + DPO • PRM-guided selection
CR-Planner [52]	Skywork-Llama3.1-8B	MCTS	LoRA	PRM	Critic-estimated rewards: • Stepwise correctness • Global impact	• MCTS simulation • Pairwise ranking loss

[1]ORM: Outcome-based Reward Model; PRM: Process-based Reward Model. [2]Full: Full parameter tuning.

结果奖励（ORM）

DeepRetrieval: 复合奖励的检索查询

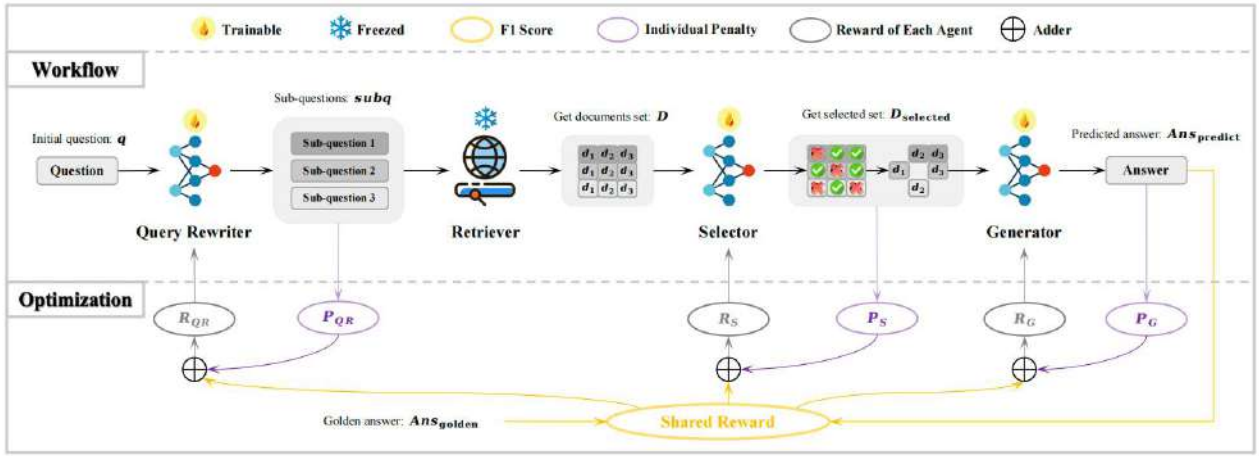
- 复合奖励函数：
  - 检索性能奖励：直接量化增强查询的实际检索效果（如文献搜索的Recall@3K，SQL的执行准确率）；
  - 格式合规奖励：约束生成查询符合目标检索系统的语法（如布尔表达式、SQL语法）。
- 推理增强生成：强制模型在生成查询前输出结构化推理步骤（<think>），通过链式思考提升查询的逻辑连贯性。



DeepRetrieval: Hacking Real Search Engines and Retrievers with Large Language Models via Reinforcement Learning.2025

MMOA-RAG: 多智能体协作

- 将RAG流程的模块（查询重写器、选择器、生成器）建模为独立RL代理，通过协作最大化共享奖励。
- 使用Multi-Agent PPO (MAPPO)联合优化，全局Critic模型共享状态价值估计。
- 全局共享奖励：基于最终答案质量的F1分数，直接对齐系统目标。
- 模块级惩罚项：查询重写器：限制子问题数量（超量生成扣分）。选择器：约束文档ID格式与去重（错误格式/重复扣分）。生成器：抑制过长回答（超长生成扣分）。

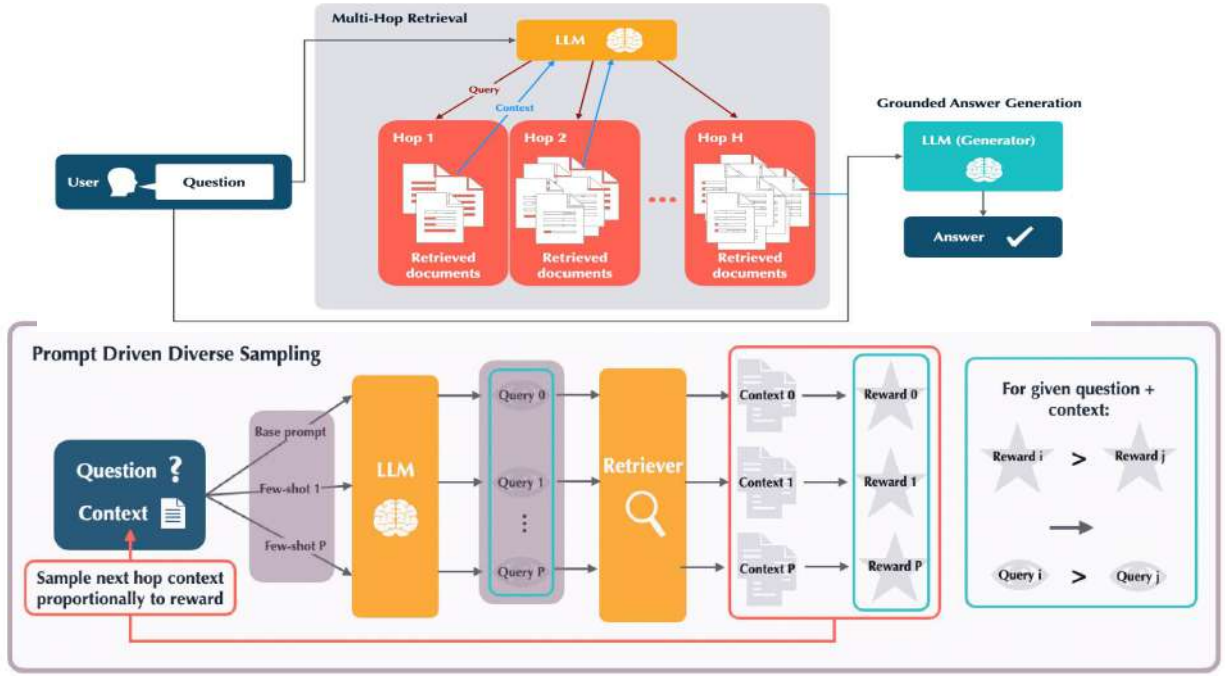


Improving Retrieval-Augmented Generation through Multi-Agent Reinforcement Learning.2025

过程奖励（PRM）

LeReT: RL驱动的高效检索

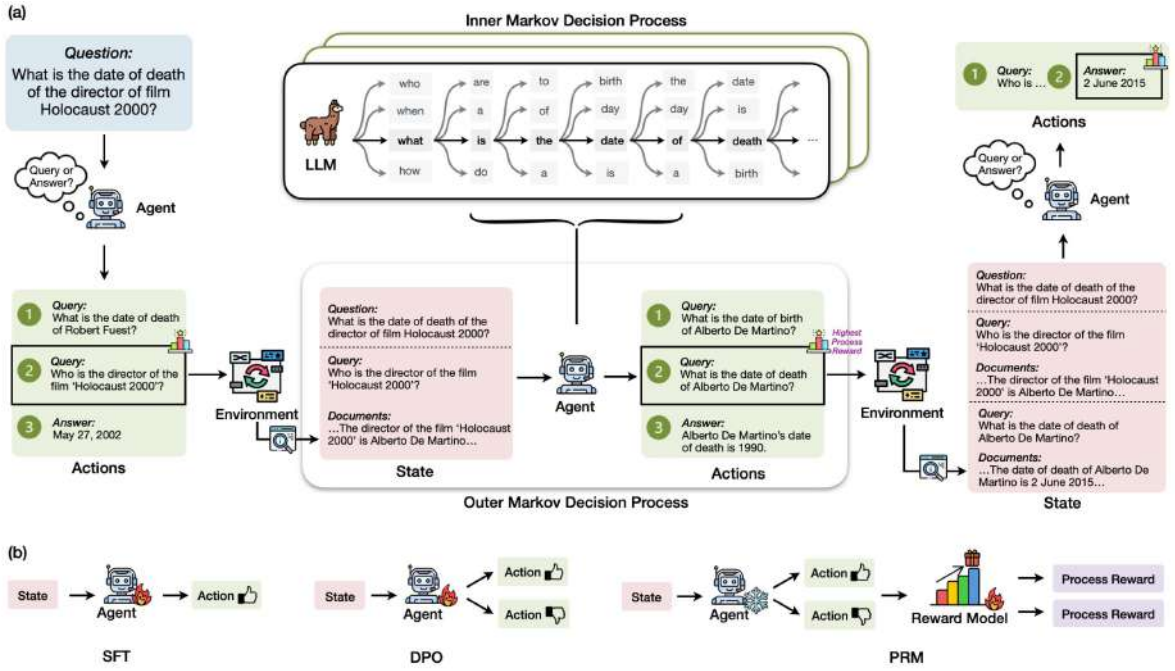
- 分步过程奖励设计：在每一步检索后引入直接奖励信号（如检索结果的平均精度AP），替代仅依赖最终答案的稀疏奖励，精准引导多跳查询优化。
- 贪婪局部优化策略：将多跳任务拆解为独立单跳优化，仅依赖当前步骤的奖励进行训练，降低长期依赖复杂度



Grounding by Trying: LLMs with Reinforcement Learning-Enhanced Retrieval.2024

RAG-Gym: 多阶段多步奖励优化

- 嵌套MDP框架：将知识密集型QA任务建模为嵌套马尔可夫决策过程（MDP），外层控制搜索与答案生成动作，内层优化LLM的特殊token生成。
- 过程监督机制：在每一步搜索中引入细粒度奖励评估，基于充分性（必要性）、效用（精确性）、冗余性（信息新颖性）三准则优化中间动作。



RAG-Gym: Optimizing Reasoning and Search Agents with Process Supervision, 2025.

混合奖励（ORM+PRM）

ReARTeR: 双重奖励机制

- 过程奖励模型（PRM）：提供密集标量奖励，对推理步骤实时打分，引导模型优化方向。
- 过程解释模型（PEM）：生成自然语言反馈，将奖励信号转化为可解释的修正建议，类似策略梯度中的优势函数。
- 结果奖励：PRM的监督数据通过蒙特卡洛方法生成，其中最终答案正确性，被反向传播到中间步骤

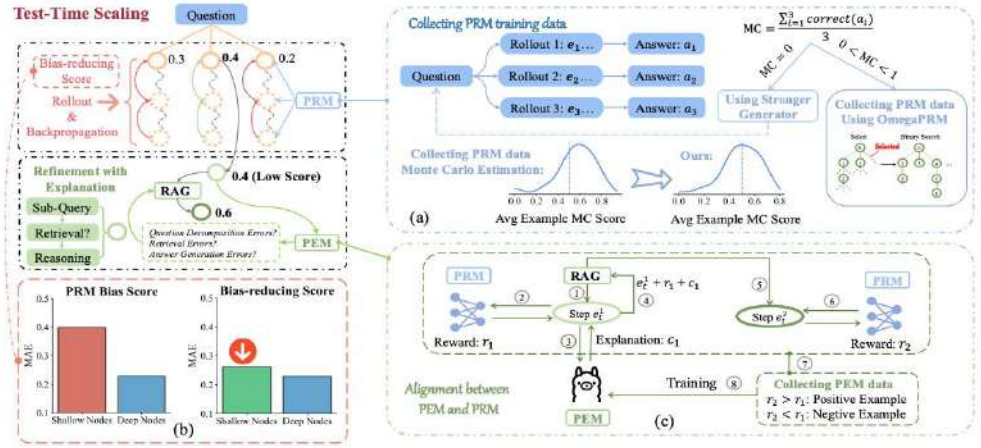
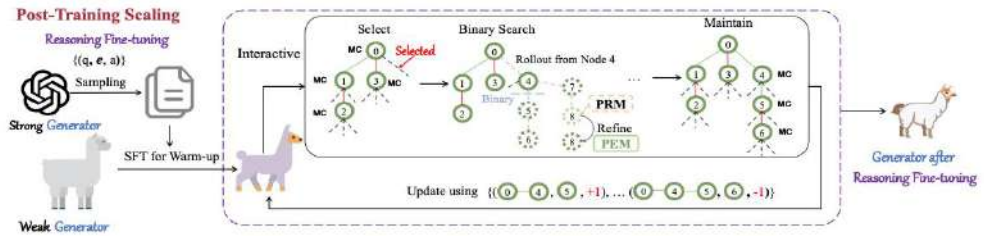


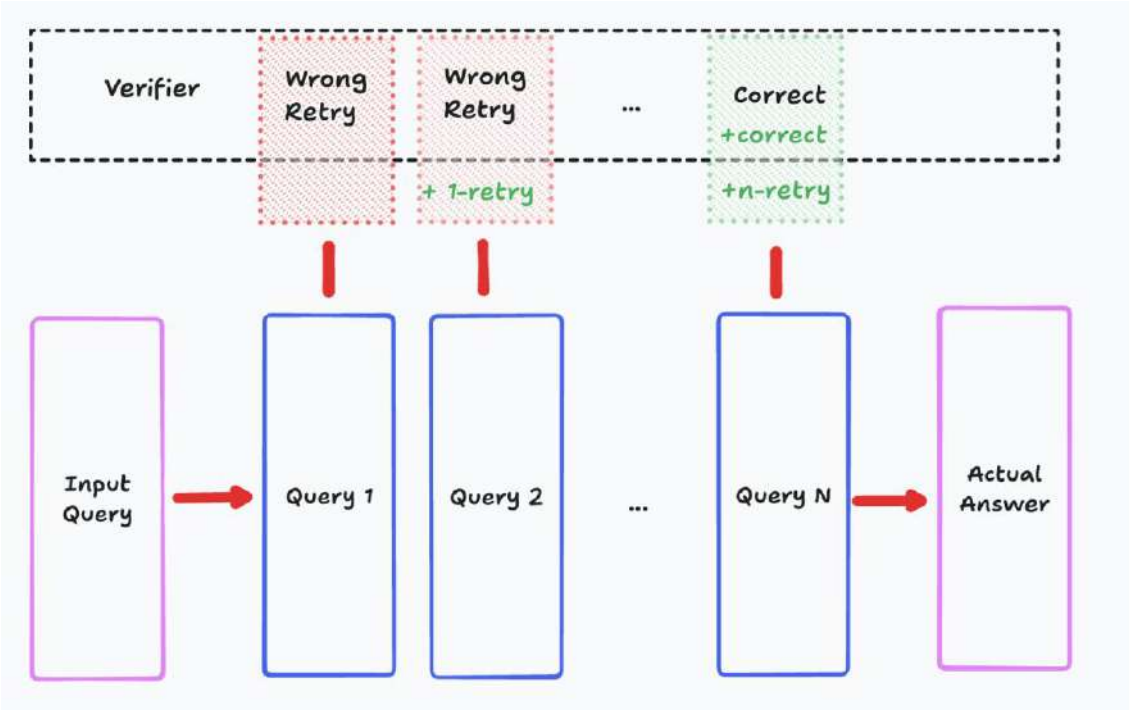
Figure 2: Test-Time Scaling of ReARTeR, which includes collecting unbiased PRM training data for PRM, reducing early-step bias in PRM, and alignment between PEM and PRM.



ReARTeR: Retrieval-Augmented Reasoning with Trustworthy Process Rewarding, 2025.

Rezero: 重试奖励机制

- 重试奖励：直接奖励“重试”行为本身（每次新增动作），但附加结果条件：仅当最终生成有效答案时才生效
- 搜索策略流程：评估多阶段搜索流程（如先广度搜索→分析→精炼→回答）的执行质量
- 多目标协同优化：通过6个互补的奖励函数（正确性/格式/检索匹配/搜索策略/多样性/重试）实现搜索行为精细调控

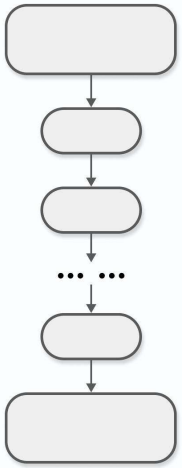


ReZero: Enhancing LLM search ability by trying one-more-time.2025

总结：RAG + Reasoning 协同的实现与优化

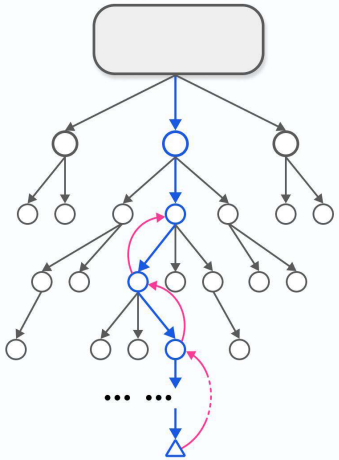
推理过程

思维链



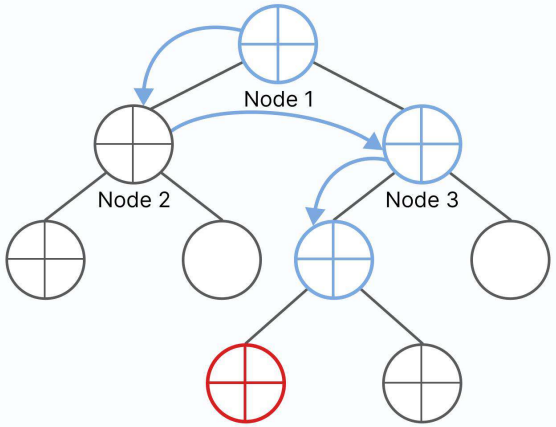
树搜索

Tree Search　　MCTS



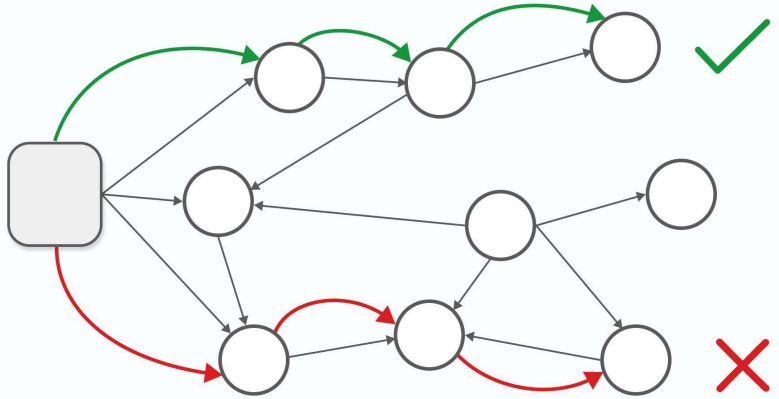
外部求解器

外部求解器上的规划



图上推理

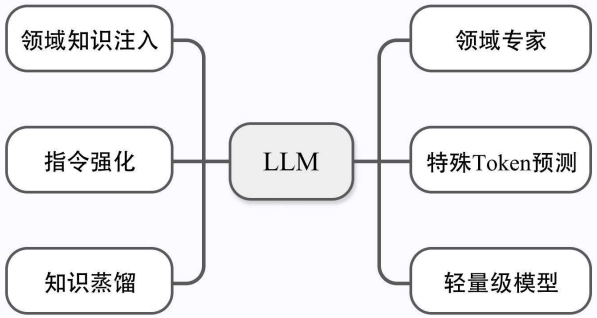
图上的多跳推理



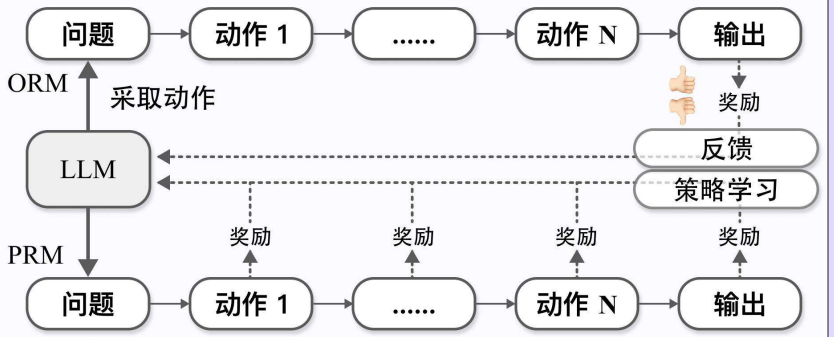
推理增强方式

提示词工程

监督微调



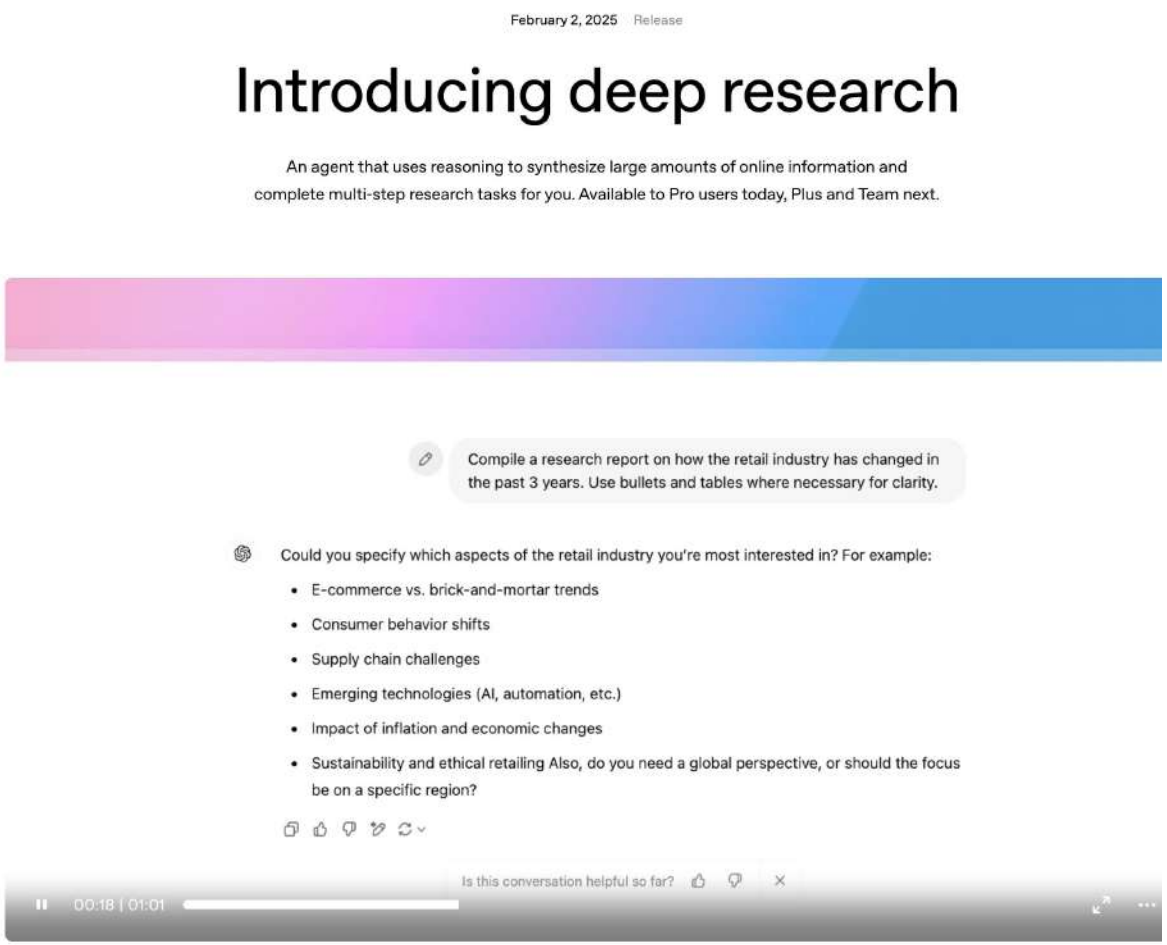
强化学习



06 AgenticRAG流程设计

6-1. 预定义流程 | 6-2. 动态流程

AgenticRAG 序幕拉开



不同模块和和算子的编排就组成了RAG Flow，也拉开了Agentic RAG序幕。

预定义流程（Pre-defined Workflow）

结构化特性

- 采用固定架构和顺序执行的多步推理方法
- 强调流程清晰度和操作确定性
- 由预定义的迭代阶段组成
- 每个阶段都有严格的输入–输出规则
- 不会基于中间结果进行动态变更

优势与局限

- 模块化设计确保了复杂任务的可控性和结构化推理
- 无论中间结果如何，都会执行所有步骤
- 保证了可重复性和稳定性
- 避免了动态决策带来的不确定性
- 虽然牺牲了适应性，但提供了程序的可预测性
- 由于缺乏实时调整，可能导致计算冗余

数学表示：$D = f_N \circ \ldots \circ f_2 \circ f_1(Q)$

其中Q为输入查询，f为推理步骤，D为最终决策输出

预定义流程的三种形式

检索前推理
Pre-Retrieval Reasoning

检索后推理
Post-Retrieval Reasoning

混合检索推理
Hybird Retrieval Reasoning

▶ 检索前推理（Pre-Retrieval Reasoning）

核心概念

检索前推理方法在进行信息检索之前首先通过推理来系统性地转换或丰富查询。

数学表达式

$$D = \Gamma \circ R \circ \Psi(Q)$$

其中Ψ是在检索前对查询进行处理的推理操作符

关键优势

- ✓ 提高检索精度，解决查询中的歧义
- ✓ 推断隐含意图，优化查询表示
- ✓ 减少不必要的检索操作，提高效率

四种主要实现方法

查询优化 (Query Optimization)

生成和选择查询变体以最大化检索相关性，通过对比训练或强化学习选择最优查询。

代表：LeReT通过迭代采样和优化平衡查询多样性和特异性

属性判断 (Attribute Judgment)

使用分类机制动态调节检索触发器，评估查询属性（如时间敏感性、意图复杂性）。

代表：UAR和AdaptiveRAG集成多阶段分类器以最小化不必要的检索

计划生成 (Plan Generation)

将复杂查询分解为结构化的子任务序列，引导检索方向。

代表：PlanRAG利用思维链推理将检索目标与多步问题求解需求对齐

语义增强 (Semantic Enhancement)

使用领域特定或任务感知的嵌入丰富查询表示。

代表：O1-Embedder将潜在推理模式集成到查询嵌入中以提高检索鲁棒性

PlanRAG——基于RAG的决策问答

推理流程设计

1. 计划阶段

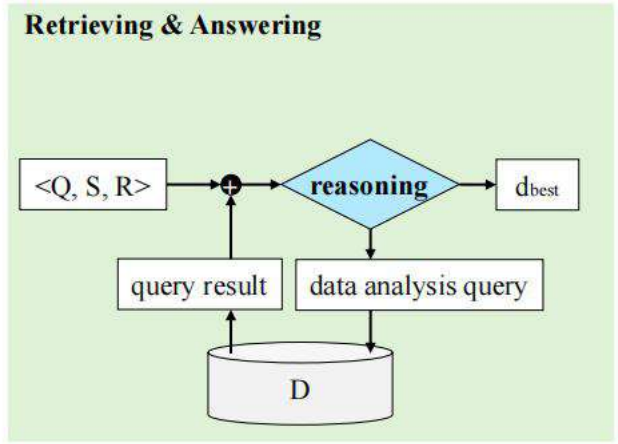
根据问题和数据模式生成初始分析计划

2. 检索与回答

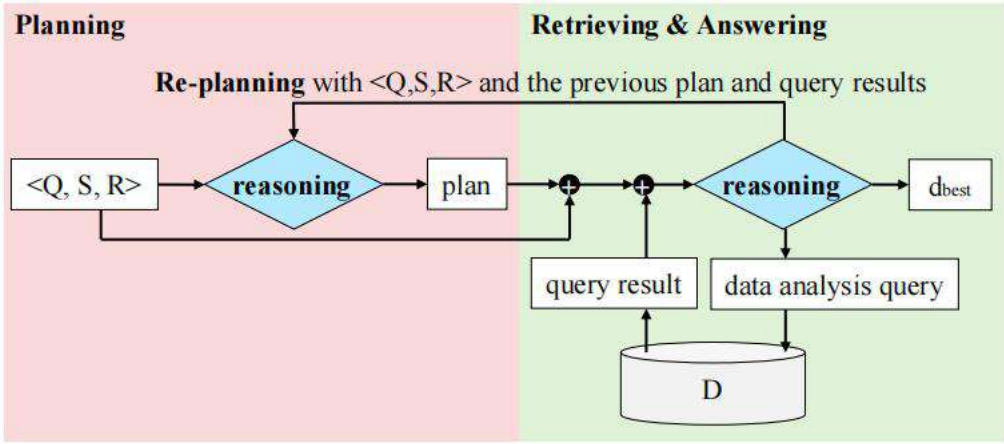
执行计划中的数据分析查询，获取决策所需信息

3. 重新计划

评估当前计划是否充分，必要时调整分析策略



(a) previous RAG (Single-turn and Iterative RAG)



(b) PlanRAG

Step 1: Making a plan for which kind of analysis is needed for decision

Where should I locate my merchant? My goal is maximizing **BAH**'s profit on home node, **Deccan**.

<The database about the international trade>

Country table

country	home node	development
BAH	Deccan	186.5
JNP	Doab	143.7
BNG	Ganges	143.7

NodeCountry table

trade node	country	is home	$TP_{country}$	has merchant
Deccan	BAH	True	160	False
Doab	BAH	False	71	False
Ganges	BAH	False	0	False

TradingNode table

trade node	is inland	LV	IV	OV	TP_{total}
Deccan	True	8.91	0.82	2.171	1128
Doab	True	6.98	0.87	1.564	1243
Ganges	False	8.31	1.07	1.647	1172

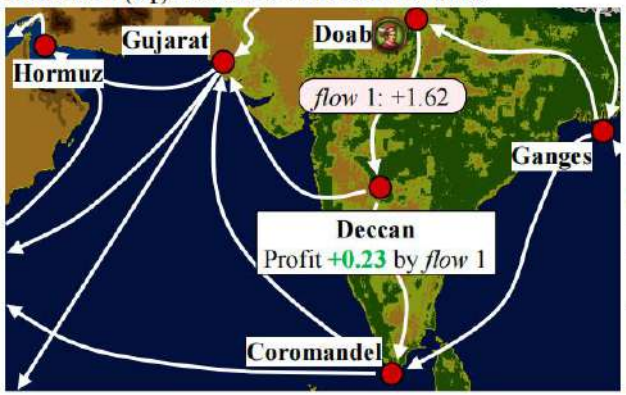
TradingFlow table

source	destination	flow
Doab	Deccan	0.78
Ganges	Doab	0.82

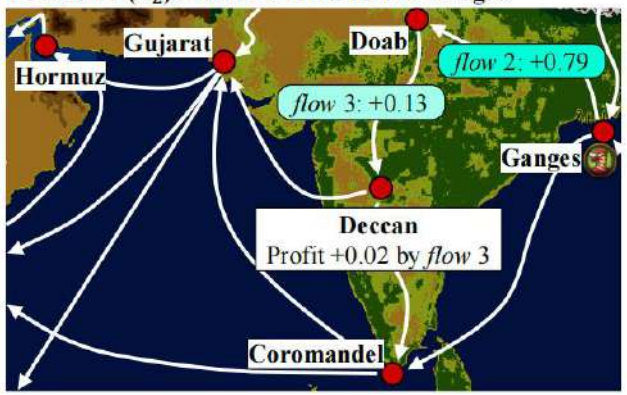
Step 1: Determine available decisions by finding source nodes. Step 2: Ascertain flow increments by decisions. Step 3: Calculate profit increments by decisions.

Step 2: Retrieving data and analyze it

Decision 1 (d_1): Locate a merchant to Doab.



Decision 2 (d_2): Locate a merchant to Ganges.



Step 3: Answering based on the result of data analysis

You should locate your merchant to the **Doab** node to steer value, so that maximize profit of BAH.

PlanRAG: A Plan-then-Retrieval Augmented Generation for Generative Large Language Models as Decision Makers. 2024

检索后推理（Post-Retrieval Reasoning）

核心概念

检索后推理是预定义RAG系统的一种关键进步，其中认知处理发生在从外部来源检索信息之后，推理的目标是针对检索到的内容。

数学表达式

$$D = \Gamma \circ \Psi \circ R(Q)$$

其中R是检索操作符，Ψ实现推理变换，Γ表示最终决策函数

检索后推理的主要特点

- 对检索结果进行深度分析和验证
- 整合和协调来自多个文档的信息
- 解决检索文档之间的冲突和矛盾
- 使用外部知识校准模型内部表示

代表性方法示例

ToG2.0

提出了一种迭代多步推理框架，在图检索和上下文检索之间交替进行。集成大语言模型的推理判断来逐步扩展实体并修剪不相关信息，最终生成准确答案。

ActiveRAG

采用预定义的三阶段过程（自我询问→知识同化→思维适应）来结构化理解和校准检索知识，解决参数记忆与外部知识之间的冲突。

通过多指令微调策略（如反事实比较和锚点关联）增强外部知识对LLM内部表示的纠正效果，大幅减少幻觉生成的可能性。

ActiveRAG: 基于RAG的决策问答

推理流程三步骤

1. **自我询问**
   基于内部知识生成初步推理链
2. **知识同化**
   深入分析检索到的外部知识
3. **思维调节**
   优化和修正推理过程

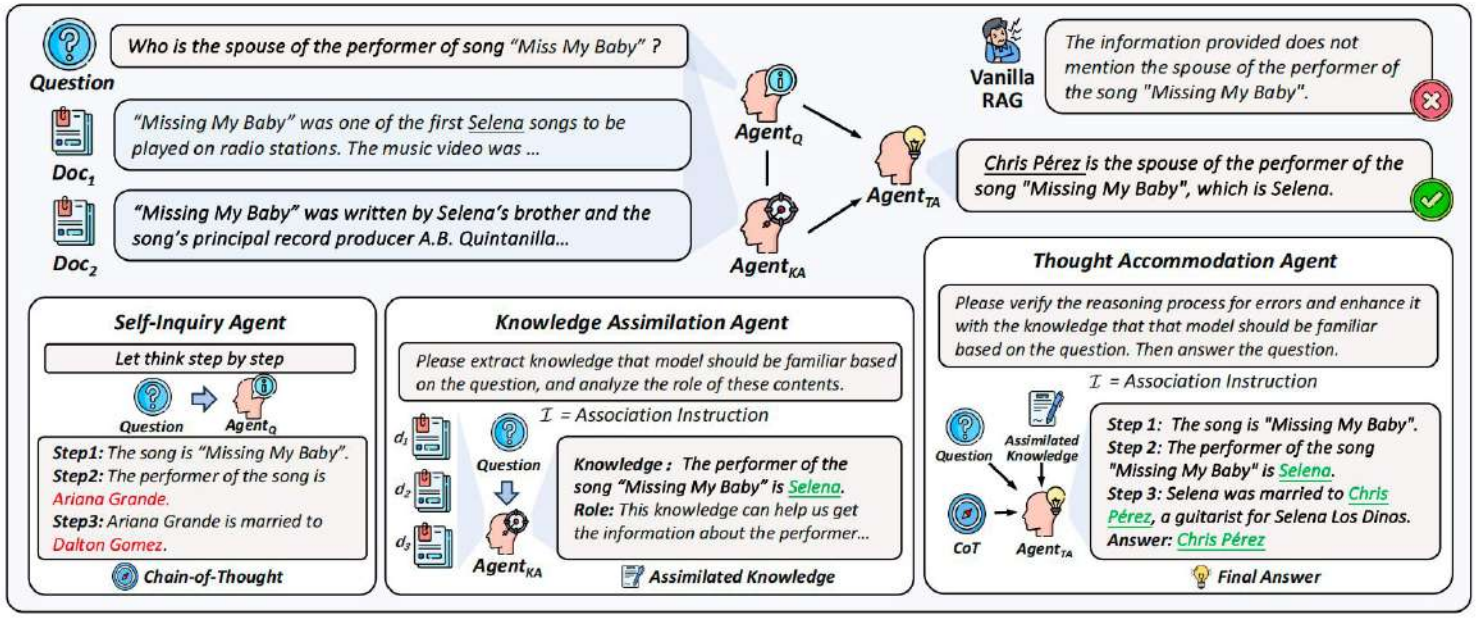
推理发生在检索之后

推理流程

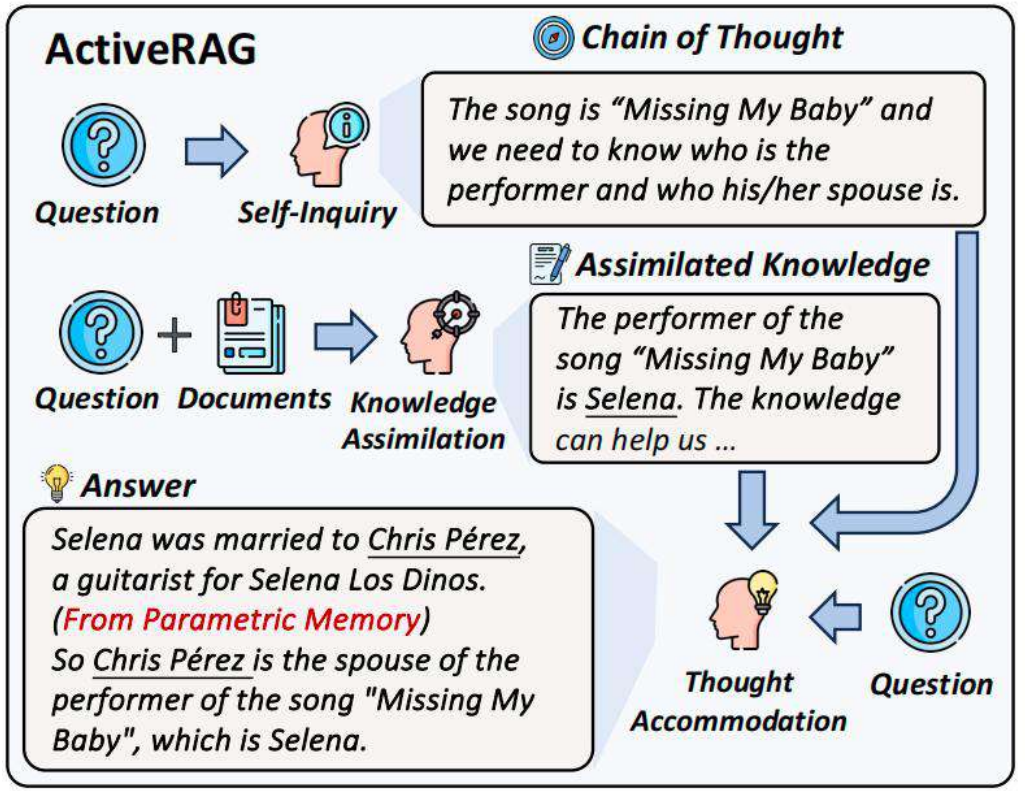
1. 首先进行知识检索
2. 然后基于检索结果进行多轮推理
3. 通过知识同化和思维调节不断优化推理

推理特点

检索是推理的基础和输入
推理过程是对检索知识的深度理解和利用
不断迭代，动态调整推理链



Question: Who is the spouse of the performer of song Missing My Baby?
Documents: "Missing My Baby" was one of the first Selena songs to be played on radio stations...
Ground Truth: Chris Pérez



ACTIVERAG: Autonomously Knowledge Assimilation and Accommodation through Retrieval-Augmented Agents.2024

O1 Embedder: 让检索器先思考

方法核心：两阶段思考式检索

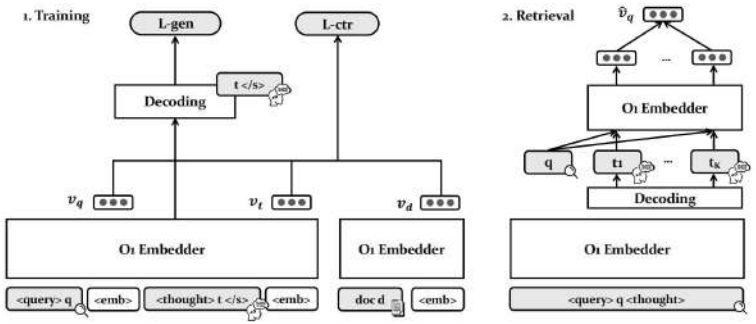
1. 思考生成阶段：模型为查询生成多个语义丰富的"思考"
2. 检索增强阶段：利用生成的思考增强查询嵌入

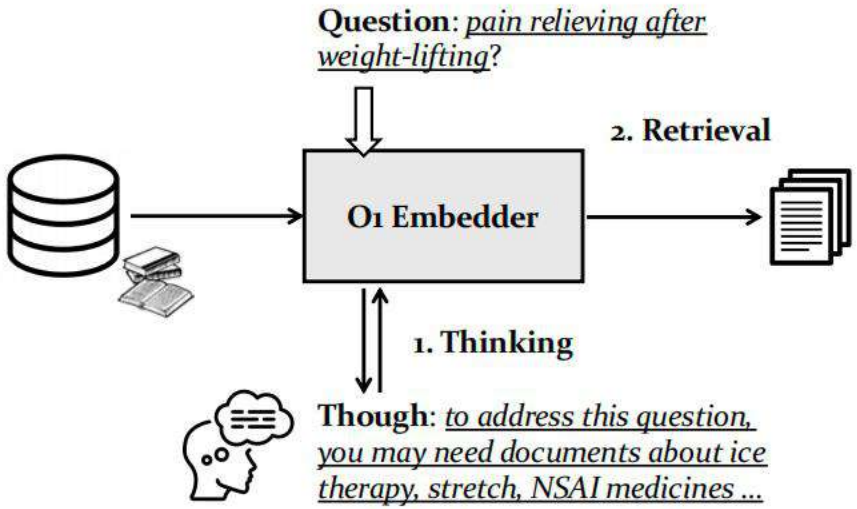
数据合成关键流程

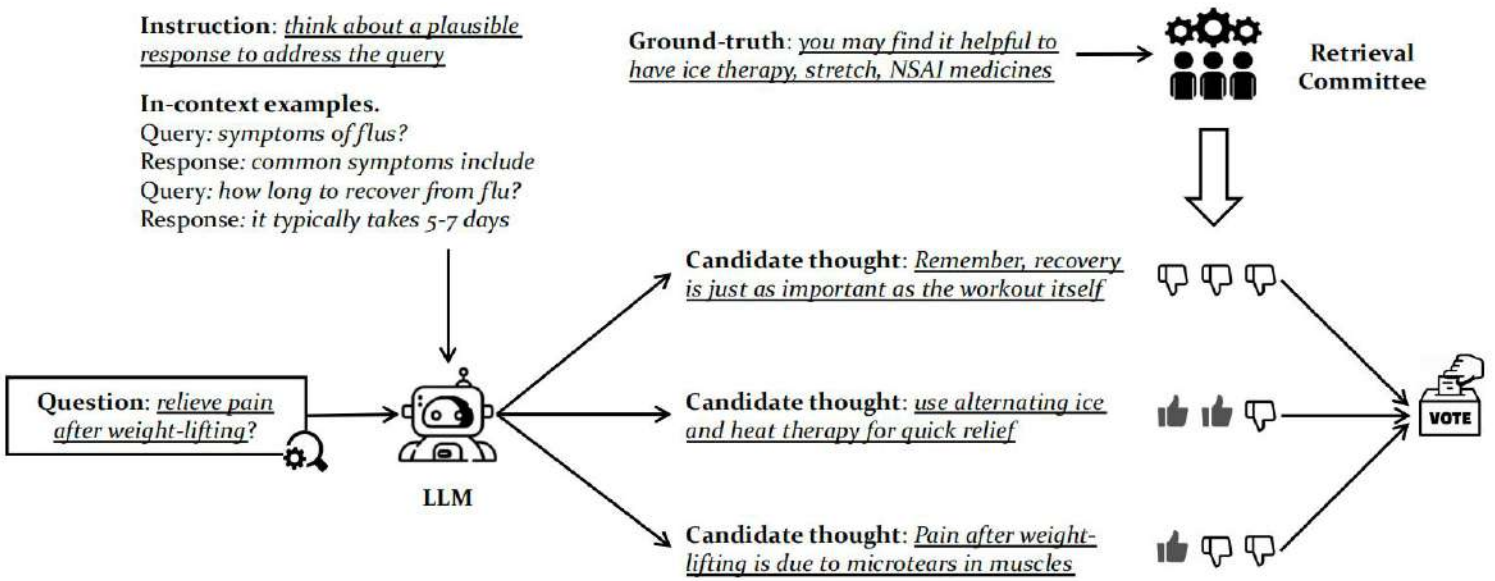
探索–精炼工作流： • 大语言模型生成候选思考 • 检索委员会评估思考质量 • 通过投票机制选择最佳思考内容

多任务训练策略

- 行为克隆：监督微调生成思考能力
- 对比学习：训练判别性嵌入能力
- 内存高效的联合训练机制







O1 Embedder: Let Retrievers Think Before Action.2025

▶ 混合推理检索（Hybrid Retrieval Reasoning）

核心概念

混合推理模式通过整合检索前推理和检索后推理形成复合处理范式，本质是一个多轮递归迭代过程。

数学表达式

$$Q_t = [\circ^t_{t=1}(R \circ \Gamma_t \circ \Psi_t)](Q_0)$$

每个迭代单元包含检索、生成和推理三个阶段

递归机制使知识获取和语义推理之间能够动态协同，克服了单周期检索–生成框架的线性限制。

混合推理的关键特点

- 结构化递归：通过预定义迭代单元实现多轮处理
- 动态查询重写：基于中间结果优化后续检索
- 层次化验证：聚合多粒度检索结果
- 迭代优化：通过反馈循环逐步提升结果质量

代表性实现方法

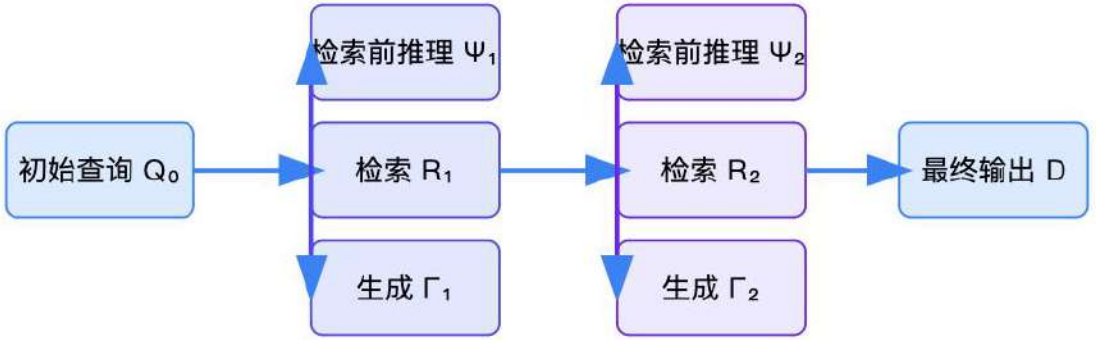
IR–CoT

利用思维链推理迭代构建中间逻辑链，通过逐步细化的上下文线索指导多跳检索。

FinSearch

引入双阶段架构，首先生成结构化搜索图建模时间和实体依赖关系，然后动态重写查询优化金融数据检索。

混合推理迭代流程



通过强制执行确定性迭代周期，混合推理方法平衡了控制精度和适应灵活性，建立了一个结构化框架来处理依赖动态上下文精化的复杂查询，同时保持了可解释的推理路径。

DeepNote: 以笔记为中心的迭代知识扩展

研究背景

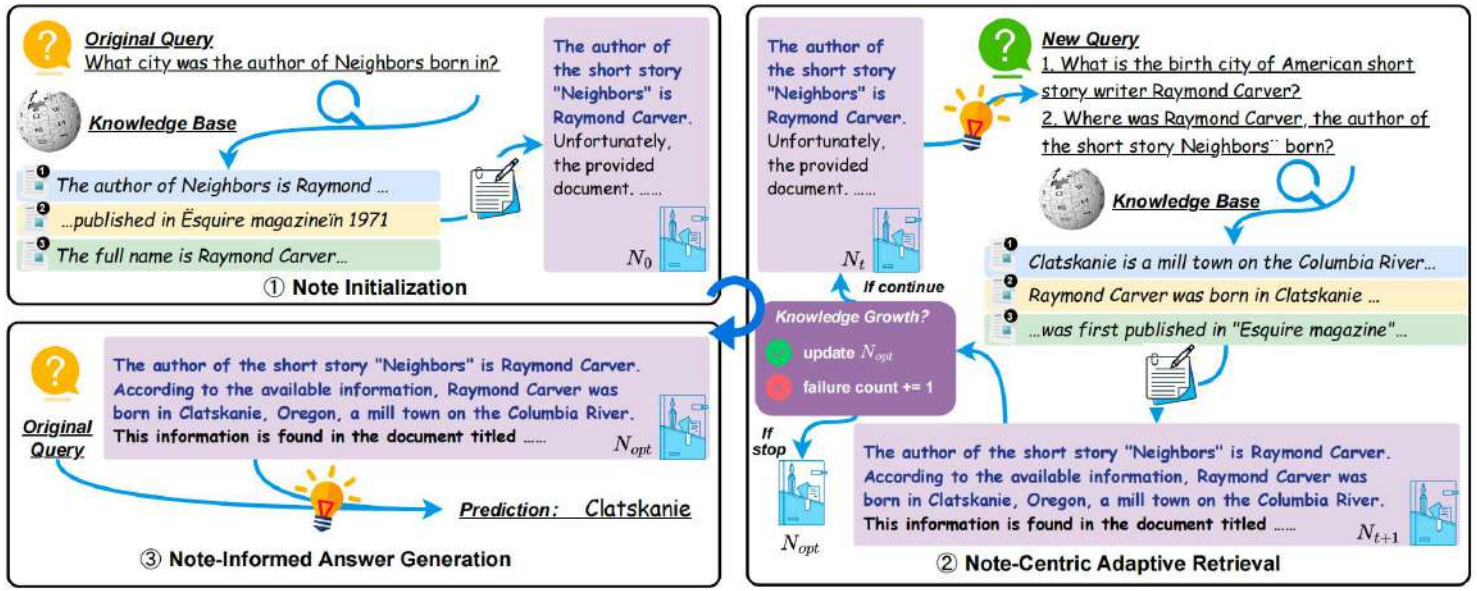
- 传统笔记工具缺乏深度知识探索能力
- 单次检索难以全面捕获复杂知识
- 知识整合和深度推理存在局限性

核心创新

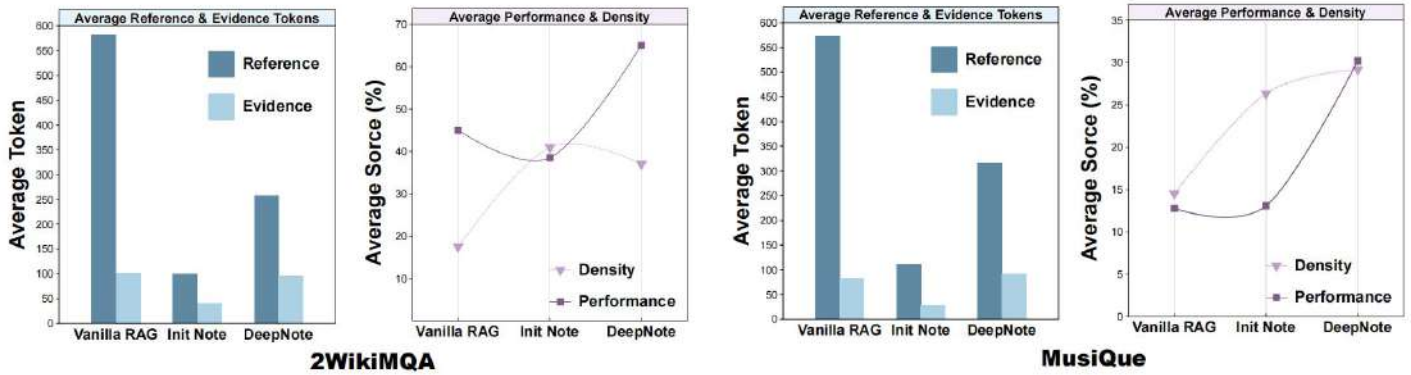
混合推理-检索迭代框架：

1. 推理驱动的知识扩展
2. 上下文感知的检索策略
3. 动态笔记自我完善机制

多单元闭环迭代：推理 → 检索 → 生成



性能提升：复杂知识任务效率提高 15.3% - 22.7%



DeepNote: Note-Centric Deep Retrieval-Augmented Generation.2025

06 AgenticRAG流程设计

6-1. 预定义流程 | 6-2. 动态流程

动态流程（Dynamic Workflow）

核心架构特性

- 以LLM为中心的自主推理架构
- 集成非确定性操作工作流和实时决策能力
- 不同于预定义管道，能够持续监控推理状态
- 动态触发检索、生成或验证操作
- LLM主动评估上下文需求，自主决定调用外部工具或资源的最佳时机
- 通过混合反馈协调机制实现灵活决策

三大核心特征

1. **上下文状态驱动的操作调用**

   操作调用由LLM的上下文状态分析控制，通过特殊标记预测（如'[Web-Search]'或'<begin_of_query>'）来启动外部操作

2. **高度灵活的推理轨迹**

   允许动态查询重构和子问题生成，克服静态工作流的局限性

3. **上下文驱动的决策机制**

   优先考虑实时推理状态而非预定义规则，增强系统对新兴任务复杂性的响应能力，同时提高精度

数学表示

将t时刻的推理状态定义为$S_t = (H_t, C_t)$，其中H_t表示历史信息聚合，C_t表示上下文嵌入向量，决策过程被建模为随机系统：

$a_{t+1} \sim \pi(S_t; \Theta)$

$S_{t+1} = \delta(S_t, T_{a_{t+1}}(S_t))$

通过可扩展的行动空间A和策略参数Θ的在线优化，确保系统在复杂问题领域中具有动态适应性。

主动驱动推理（Proactivity-Driven Reasoning）

主动驱动推理的关键特征是模型的自主行动触发机制：

自主决策

模型根据内部评估独立触发操作，无需外部干预

直觉推理

类似人类直觉决策，当模型识别到当前推理过程中证据不足时，主动发起检索请求

示例：模型在发现当前推理链缺少关键信息时，自动触发额外的知识检索

主动驱动推理（Proactivity-Driven Reasoning）

机制概述

主动性驱动推理是动态RAG工作流的一种形式，其特点是模型基于内部评估自主触发行动，无需外部干预即可执行操作。

关键特性

- LLM通过类似人类直觉决策的机制自主执行操作
- 当模型独立识别当前推理过程中证据支持不足时，会主动生成检索请求补充信息
- 自主决策何时需要外部知识与工具
- 模型内部状态触发，无需外部控制信号

代表系统与技术实现

Agentic Reasoning

通过自主触发的知识图谱构建和多轮推理，实现复杂问题分解和逐步求解，模型主动决定何时需要补充知识

DeepRAG

集成自适应检索策略，模型根据当前推理状态自动调整检索深度和广度，平衡探索与利用

Co-STORM

采用多智能体系统，主机模块通过分析未引用文档中的潜在语义生成跨模态检索命令

R1-Searcher

模型通过内部评估机制，在生成过程中主动识别需要外部验证的陈述，并触发针对性检索

应用价值与优势

主动性驱动推理特别适用于需要灵活信息获取的复杂领域，如科学研究、法律分析和医疗诊断。通过模型自主判断知识需求，该方法可以减少人工干预，提高信息获取效率，并在复杂推理过程中保持连贯性。这种方法模拟了人类专家在面对未知问题时的认知过程：识别知识缺口、主动寻求相关信息，并将新信息整合到现有推理框架中。

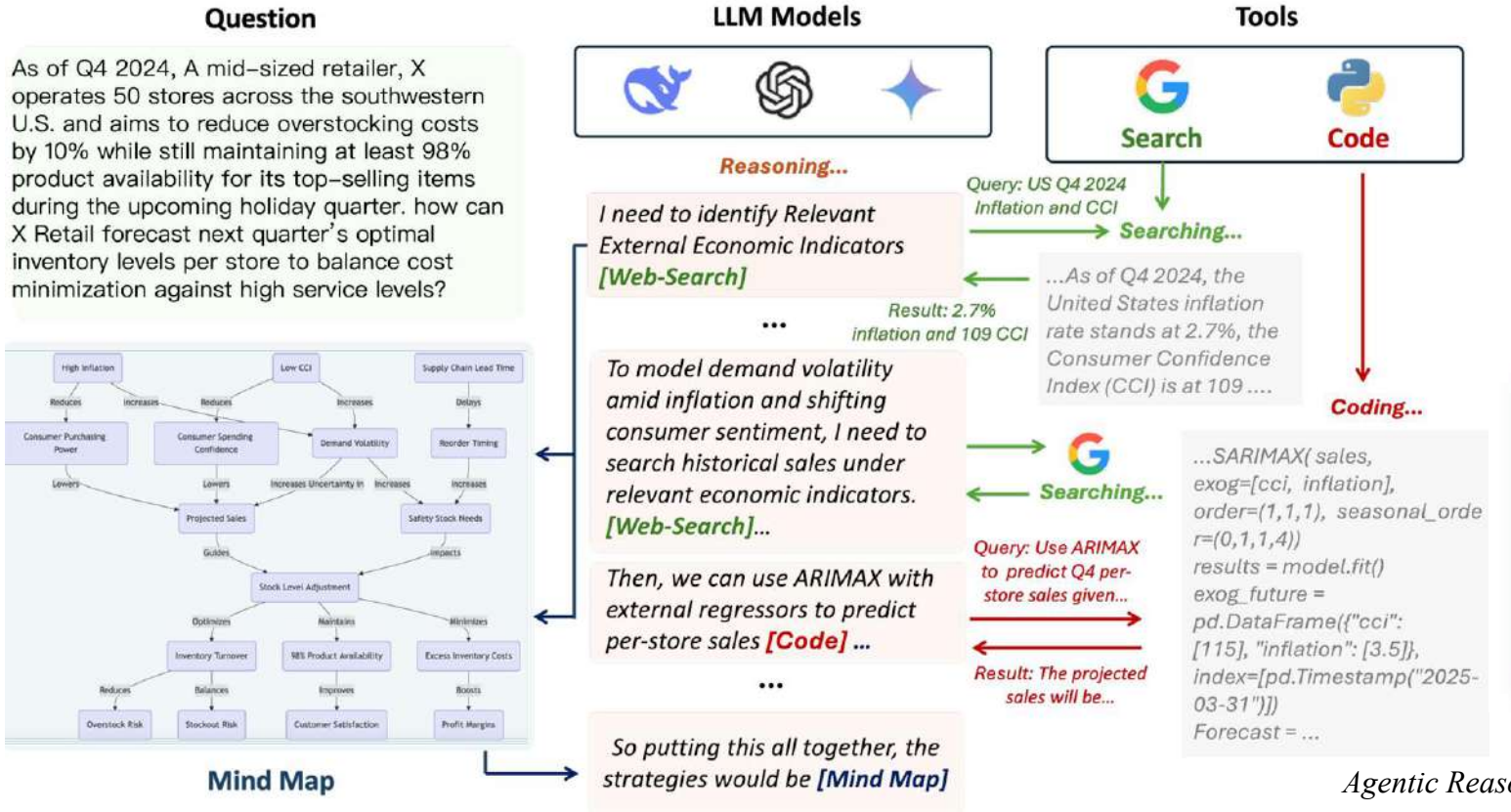
Agentic Reasoning: 智能推理框架

研究动机

- 传统语言模型推理局限于内部推断
- 需要动态获取外部信息和计算支持
- 模仿人类使用外部工具进行复杂推理

核心方法

- **Web搜索代理**：实时检索相关信息
- **代码执行代理**：支持量化分析和计算
- **思维导图代理**：构建知识图谱，追踪推理脉络



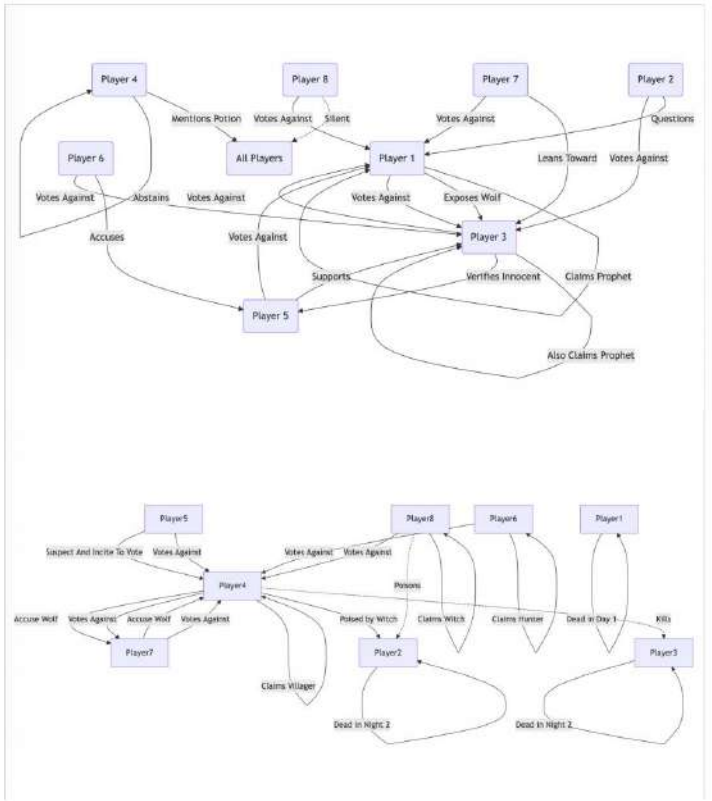


Figure 5: Mind Map in playing werewolf game. The first round and the second round.

推理流程设计

主动触发
推理模型实时判断何时需要外部信息

上下文嵌入
生成专用Token调用外部代理

迭代优化
不断丰富和精炼推理链

Agentic Reasoning: Reasoning LLMs with Tools for the Deep Research，2025

反思驱动推理（Reflection-Driven Reasoning）

反思驱动推理强调对推理过程的自我审查与评估：

定量评估

通过计算中间结果的质量分数，动态启动后续操作

自我检查

持续监控推理过程的逻辑连贯性和知识完整性

动态调整

根据自我评估结果，实时调整推理策略和检索方向

示例：当推理支持分数超过0.7时，触发进一步的知识验证或补充检索

反思驱动推理（Reflection-Driven Reasoning）

机制概述

反思驱动推理强调对推理过程的自我检查，通过对中间结果质量的量化评估动态启动后续操作。

关键特性

- 当计算的推理支持分数超过预定阈值时触发行动
- 通过系统性自我检查和评估引导工作流程
- 模型不仅生成内容，还能评估自身输出质量
- 使用明确的评分机制指导决策过程
- 能够识别并修正推理中的潜在错误或不确定性

工作原理

反思驱动推理通过以下过程运作：

1. 生成初步推理或答案
2. 对输出进行自我评估和打分
3. 基于评分决定是否需要额外知识
4. 如需要，动态触发检索或验证
5. 整合新信息，重新生成优化结果

代表系统与应用

Flare
利用持续自我评估和反思过程，模型在推理过程中动态识别不确定性高的陈述，并触发针对性的外部检索以验证或纠正这些陈述，从而提高整体推理可靠性

OpenRAG
实现开放式的自评估框架，通过多维度（相关性、充分性、一致性等）评分来监控检索–生成过程的质量，并根据评分结果动态调整检索策略

Self–RAG
训练模型同时生成内容和相应的质量评分，使系统能够识别需要外部知识支持的声明，并在必要时触发额外检索，大幅减少幻觉并提高事实准确性

ReaRAG
通过集成自我反思和推理链长度控制机制，动态剪枝无效的推理分支，避免过度思考问题，提高推理效率和质量

优势与应用场景

反思驱动推理特别适合需要高精度和自我修正能力的场景，如科学研究、医疗诊断和金融分析。通过结合LLM的生成能力和自我评估功能，该方法显著提高了系统在复杂任务中的可靠性和透明度。反思机制不仅有助于减少幻觉和错误，还能为最终结论提供更强的证据支持和可解释性。

OpenRAG: 反思驱动的推理增强生成

反思驱动推理框架

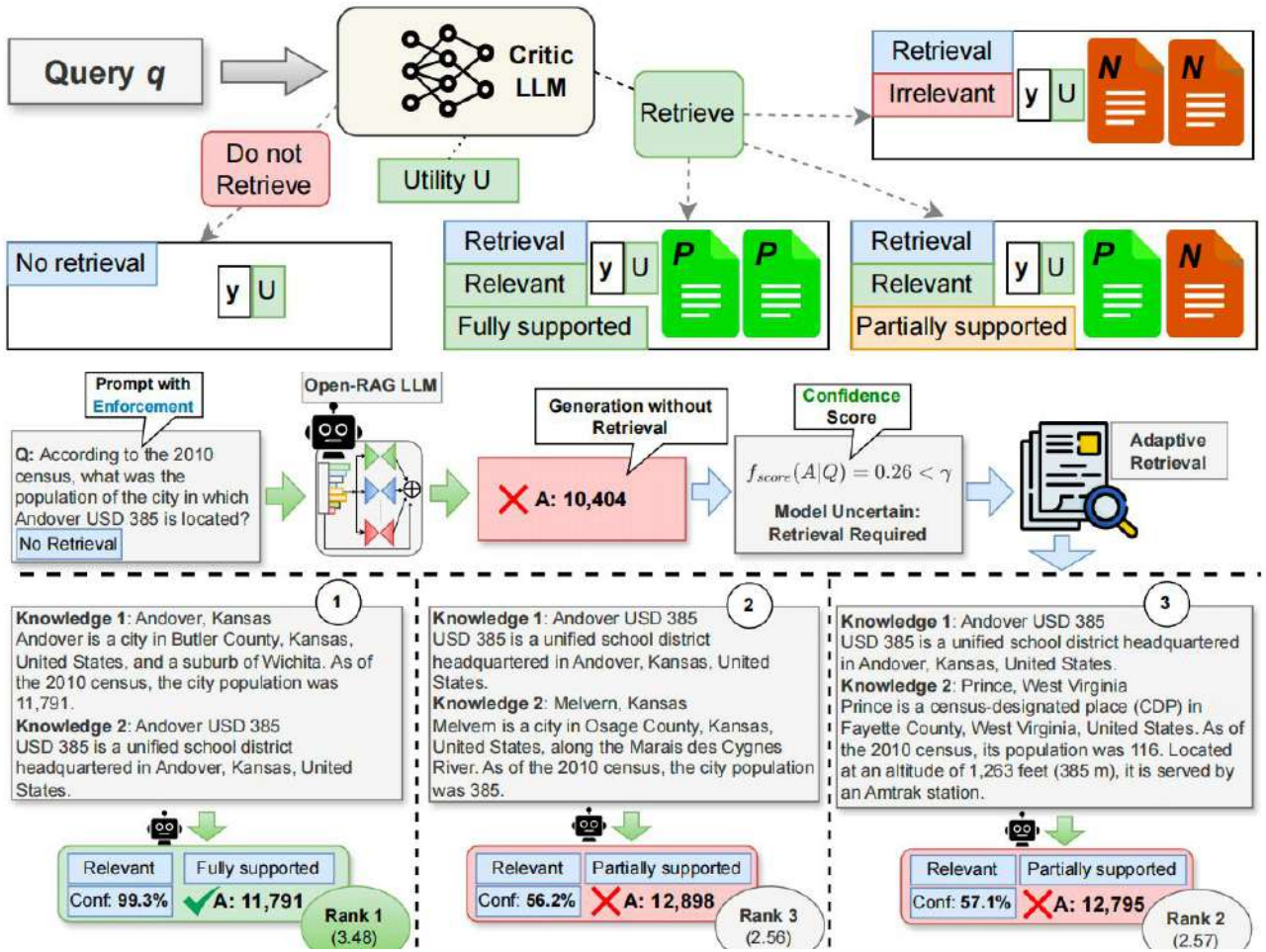
持续反思机制
动态评估推理过程中的信息质量

自适应推理
根据反思结果调整检索和生成策略

精确信息定位
通过反思机制过滤无关信息

推理流程

1. 初始查询输入
2. Reflection Token生成
   决定是否需要检索
3. 自适应检索
   多跳/混合检索策略
4. 专家路由选择
   动态激活Top-2专家
5. 对比学习过滤
   去除干扰上下文
6. 反思性生成
   持续评估生成质量



反思驱动特征
通过Reflection Token主动评估推理过程，实现动态、自适应的知识检索和生成

推理灵活性
多专家混合模型支持上下文敏感的推理策略动态调整

信息质量控制
持续反思机制确保生成内容的准确性和相关性

OPEN-RAG: Enhanced Retrieval-Augmented Reasoning with Open-Source Large Language Models. 2024

反馈驱动推理（Feedback-Driven Reasoning）

反馈驱动推理通过外部信号实现可控的推理优化：

外部信号

接收并整合来自外部环境的可解释反馈信号

策略更新

基于反馈信号动态调整推理和检索策略

保持生成能力

在优化推理过程的同时，保持大语言模型的生成能力

通过统一的策略更新框架，实现对推理过程的可控优化，提升系统在复杂任务中的适应性

反馈驱动推理（Feedback-Driven Reasoning）

机制概述

反馈驱动推理依赖外部信号或评估指标来指导RAG系统的动态决策过程，形成闭环的优化机制。

关键特性

- 利用外部反馈信号（如验证结果、用户响应、环境变化）调整推理轨迹
- 形成闭环控制和迭代改进机制
- 能够动态平衡探索与利用的权衡
- 通过持续评估和调整实现自适应优化
- 支持多轮交互和增量式知识整合

工作流程

反馈驱动推理通常遵循以下流程：

1. 基于当前状态生成初步推理或查询
2. 执行检索或工具调用获取外部信息
3. 评估反馈结果与预期的差距
4. 根据反馈调整推理策略或查询方向
5. 循环迭代直至达到满意结果或收敛

代表系统与技术实现

SmartRAG

应用近端策略优化(PPO)，结合答案级F1奖励和过度检索惩罚，平衡知识完整性和效率

CR-Planner

通过迭代扩展触发专业知识（如算法复杂度的教科书证明）的检索，通过多轮验证确保准确的领域知识整合

MCTS-KBQA

利用蒙特卡洛树搜索优化知识库问答过程，通过反馈指导探索路径选择，实现复杂查询的高效解析

ReSearch

实现递归式搜索策略，模型根据初始搜索结果的质量反馈，动态调整后续查询，形成自适应搜索层次

应用价值与优势

反馈驱动推理特别适合探索性任务和连续决策场景，如科研发现、复杂问题解决和交互式学习。其优势包括：

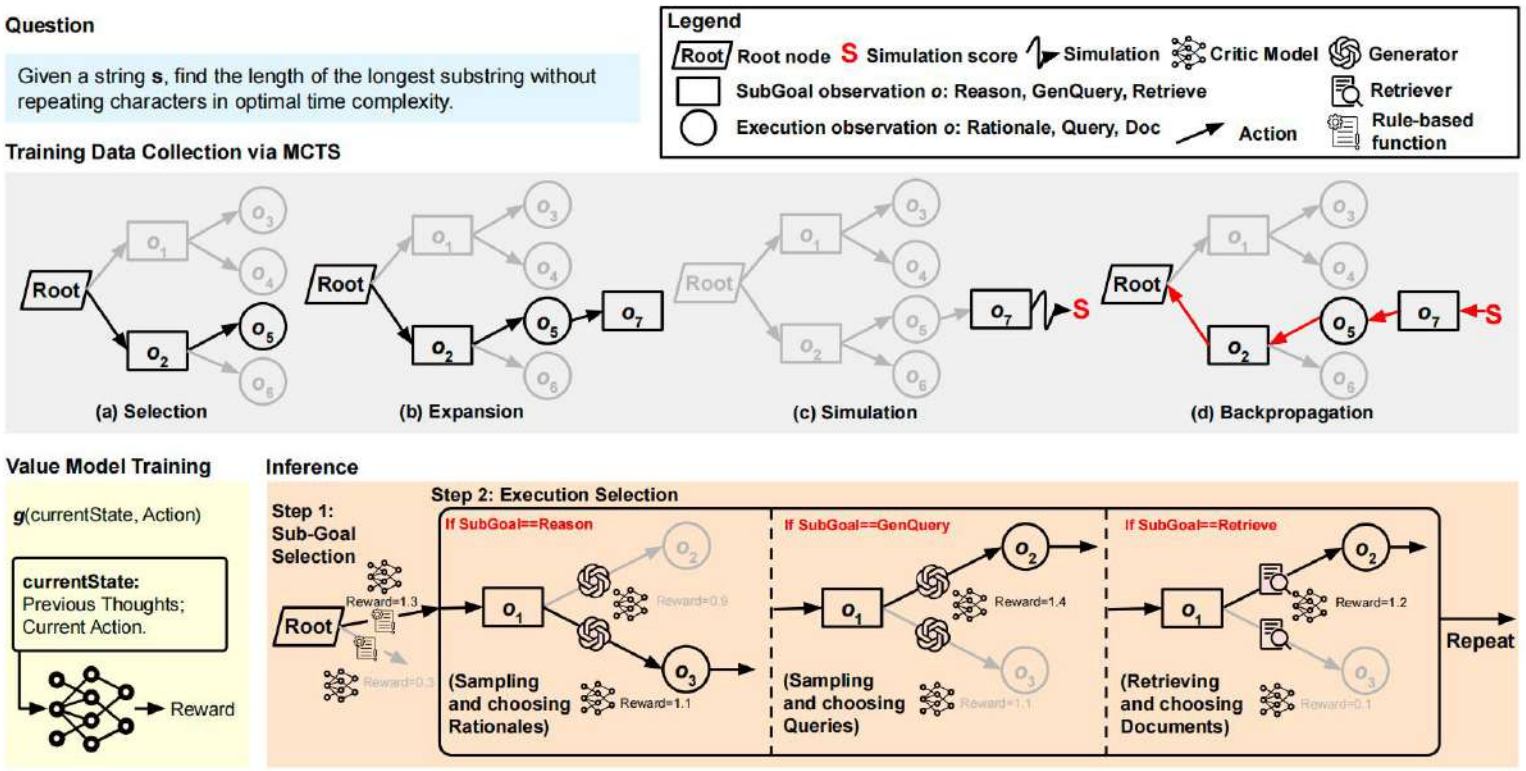
- 能够处理高度不确定性和动态变化的环境
- 支持探索未知领域和解决开放式问题
- 通过持续学习和调整提高系统性能
- 在资源有限的情况下实现计算效率优化
- 更好地适应用户需求和问题上下文变化

CR-Planner: 外部反馈驱动的推理规划框架

外部反馈驱动机制

推理过程通过三层外部反馈循环实现持续优化：

1. **局部步骤反馈**
   - 每个推理步骤实时评估和纠正
   - 批评者模型提供即时质量评估
2. **路径级别反馈**
   - 评估整体推理路径的有效性
   - 动态调整推理策略
3. **全局学习反馈**
   - 通过蒙特卡洛树搜索积累经验
   - 长期优化推理决策模型



批评者模型的关键职责

- 实时评估推理质量和可靠性
- 提供细粒度和宏观层面的反馈
- 指导模型动态调整推理策略

推理驱动模式

外部反馈驱动：通过持续的批评者模型评估，实现推理过程的自适应调整。 不仅仅是被动接收反馈，而是主动利用外部信号不断优化推理策略和决策路径。

Can We Further Elicit Reasoning in LLMs? Critic-Guided Planning with Retrieval-Augmentation for Solving Challenging Tasks. 2024

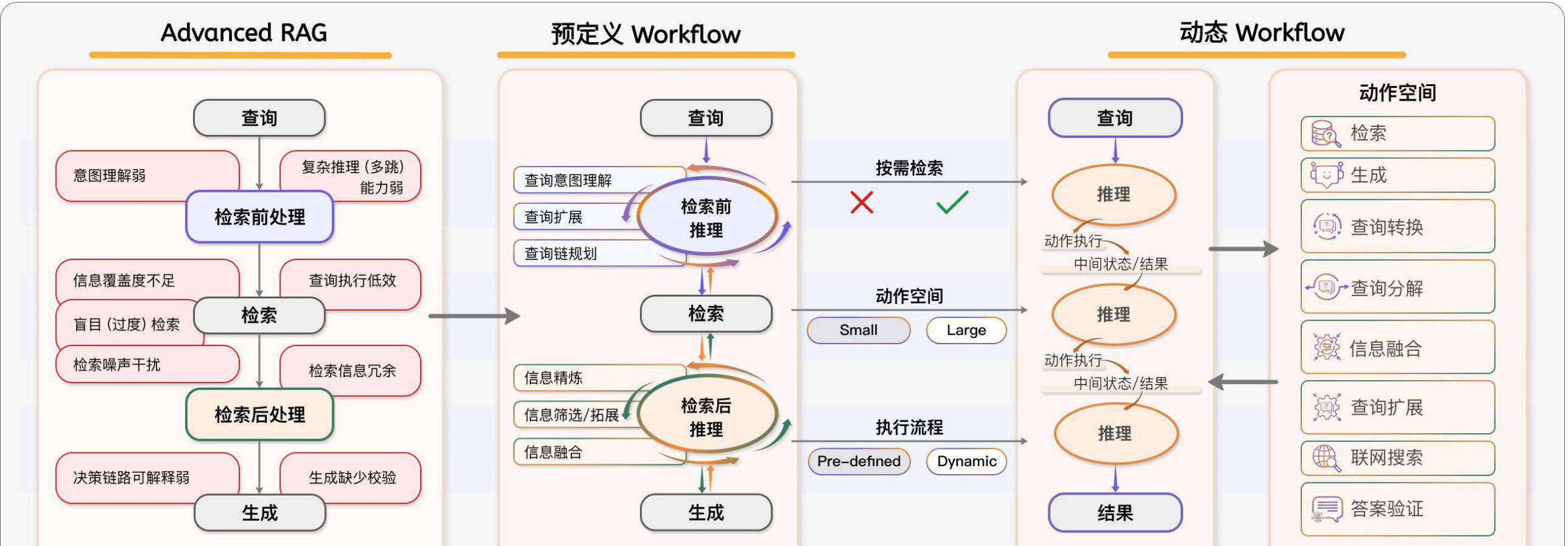
总结：RAG系统注入Reasoning能力的三种典型模式

预定义流程

检索前推理　检索后推理　混合推理

动态流程

主动驱动　反思驱动　反馈驱动



07 评测的新趋势

7-1. RAG 评测现状 | 7-2. OneEval | 7-3. AGI-Eval

▶▶ 当前RAG评测现状

知识密集型问答（Knowledge-Intensive QA）仍是RAG系统评测的核心任务：

▶ 主要数据集：

- 单跳事实查询：Natural Questions (NQ)
- 多跳问答：HotpotQA, 2WikiMultiHopQA, MuSiQue

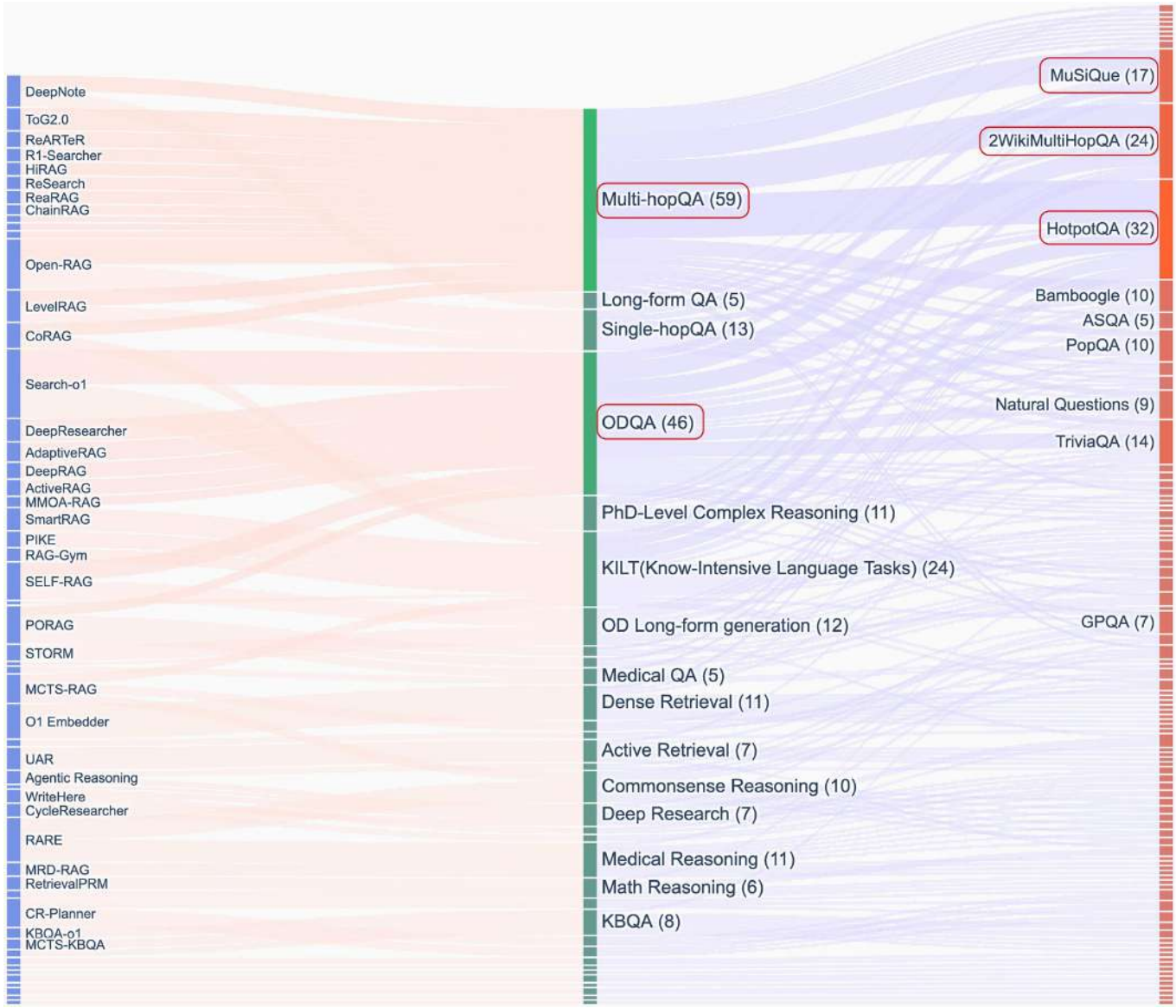
▶ 当前评测面临的主要挑战：

- 问题难度不足，大多数可直接由语言模型回答
- 缺乏深度分析性评估
- 任务单一，主要依赖问答交互

▶ 现有评估方法的局限性：

- 缺乏特异性：主要集中在事实性评估和知识检索，未深入分析认知深度
- 任务单一性：过度依赖问答类任务，缺乏与真实世界应用场景对齐的评估
- 维度不足：仅关注最终输出，忽视多步推理过程中的中间推理链

关键目标：构建更加贴近真实世界的复杂知识检索和推理系统



新兴评测任务

新兴评测任务的关键特征：

- **任务复杂性**：从基础知识检索到多层次信息集成和推理
- **代表性新任务**：
  - 深度研究任务：跨领域知识综合
  - PhD级复杂推理：高难度专业场景
  - 领域特定决策支持：医疗诊断、法律分析
- **关键评测数据集**：
  - WildSeek：信息检索场景
  - GAIA：真实世界问题解决
  - SolutionBench：跨工程领域问题解决
  - GPQA：研究生级别问答基准
- **评测维度扩展**：从结果导向到过程与结果并重
  - 任务执行能力
  - 适应性
  - 协作性
  - 泛化能力
  - 真实世界推理

Table 2. Tasks and Datasets under the New Trend of RAG Combined with Reasoning

Task Type	Sub-Task	Dataset	Description	Scale	Construction By	Evaluation	Paper
Deep Research	Deep Research	Agentic Reasoning Deep Research [92]	PHD-level dataset covering finance, medicine, and law.	15-30 domains	PhD Experts	Expert pass rate	[92]
	Report Generation	WildSeek [44]	Info-seeking task-goal pairs for document generation.	100 samples	Rules/LLM/Manual	LLM	[98]
	Report Generation	TELL ME A STORY [37]	fiction writing evaluation dataset: detailed prompts and long-form narratives.	230 samples	Manual	LLM	[98]
	Peer Review	Review-5k [91]	ICLR 2024 peer review dataset: paper metadata and structured reviewer feedback.	4,991 papers	OpenReview/arXiv	MSE/MAE/Acc	[91]
	Report Generation	Research-14k [91]	2022–2024 Accepted ML papers: outlines, full texts, and cited abstracts.	14,911 papers	Semantic Scholar + arXiv	Simulated review scores	[91]
	Report Generation	SolutionBench [54]	Engineering benchmark: constrained solutions across 8 real-world domains.	1,050 datapoints	Manual/LLM extraction	Analytical/ Technical scores	[54]
Mathematics & Reasoning	Math Reasoning	GPQA [67]	PHD-level MCQs in physics, chemistry, and biology.	744 sets	PhD Experts	Accuracy	[92]
	Math Reasoning	MATH500 [55]	500 math problems from the MATH test set.	500 problems	Public repos	Pass@K	[51]
	Programming	LiveCodeBench [40]	Programming benchmark with easy, medium, and hard problems.	1,055 problems	Competition platforms	Pass@K	[51]
	Programming	USACO [70]	USA Computing Olympiad problems, testing algorithms and coding.	307 problems	USA Computing Olympiad	Pass@K	[52]
	Math Reasoning	TheoremQA-Math [33]	BRIGHT subset: theorem-based math problems.	206 problems	STEM datasets	Accuracy	[52]
	Programming	Gorilla [64]	API-aware code generation from HuggingFace, Torch Hub, TensorFlow Hub docs.	1,600 APIs	Manual	AST matching	[73]
	Math Reasoning	OlympiadBench [29]	Olympiad-level math competition problems.	1,000 problems	Competitions	Accuracy/F1	[109]
	Complex Reasoning	ComplexWebQA [76]	Multi-step reasoning over web queries with cross-document integration.	34,689 queries	Web snippets	Accuracy	[36]
Demanding Retrieval	Domain Retrieval	StackEcon & StackBio [33]	Biology and economics StackExchange questions for complex retrieval.	206 queries	StackExchange	nDCG@K	[52]
	Active Retrieval	AR-Bench [14]	Active retrieval benchmark with four sub-tasks.	8k/sub-task	Synthetic	Accuracy	[14]
	Real-time	TAQA [104]	QA dataset with time-evolving answers.	10K-100K rows	Human-curated	LLM	[14]
	Real-time	FreshQA [80]	Dynamic fact QA benchmark with evolving answers	600 samples	Mixed sources	LLM	[14]
	Domain Retrieval	PubMed [42]	PICO-based medical search dataset linking reviews to PubMed.	21k+ samples	Systematic reviews	Recall@K	[42]
	Domain Retrieval	Trial search [42]	PICO-based clinical trial search linked to ClinicalTrials.gov.	7k+ samples	Manually	Recall@K	[42]
	Domain Retrieval	FinSearchBench-24 [50]	Financial retrieval benchmark covering stocks, rates, policy, trends.	1,500 queries	Manually	Accuracy	[50]
Decision & QA	Business	DQA [48]	Decision QA benchmark with business scenarios in enterprise settings.	301 pairs	video games	Accuracy	[48]
	Medical	CMB-Clin [87]	CMB subset for clinical diagnosis reasoning in Chinese medical cases.	74 cases	Textbooks/diagnostic materials	LLM/Expert	[11]
	Medical	MM-Cases [11]	Medicine cases generated by GPT-4o-mini, verified by doctors.	609 cases	LLM/doctor-reviewed	LLM/Expert	[11]
	Medical	TCM-Cases [11]	TCM patient cases generated by GPT-4o-mini, verified by doctors.	130 cases	LLM/doctor-reviewed	LLM/Expert	[11]

典型数据集：深度研究任务的评估维度

代表性数据集及其特点：

🌍 **WildSeek**：覆盖24个领域的信息检索场景，100个数据点

🤖 **GAIA**：466个任务，评估真实世界问题解决能力

🏗️ **SolutionBench**：跨越8个工程领域的复杂问题解决方案

评估维度包括：**任务执行、适应性、协作性、泛化能力、真实世界推理**

WildSeek深度搜索数据集

数据集定义与目的

WildSeek是一个创新的大规模开放域信息检索意图数据集，旨在捕捉真实用户在复杂、开放性信息检索场景中的搜索意图和行为。

核心目标： • 理解用户实际的信息寻求动机 • 研究大语言模型在复杂搜索任务中的性能 • 提供更接近真实世界的搜索场景数据

评估方法与指标

多维度评估框架

- **相关性评估**：检查检索结果与原始意图的匹配程度
- **信息深度**：分析结果的专业性和深入程度
- **新颖性指标**：衡量结果的创新性和非重复性
- **覆盖广度**：评估结果的多角度和全面性

定量评估技术

- 使用Prometheus模型进行自动评估
- 计算信息多样性熵指数
- 语义相似度分析
- 专家人工复审

数据集示例

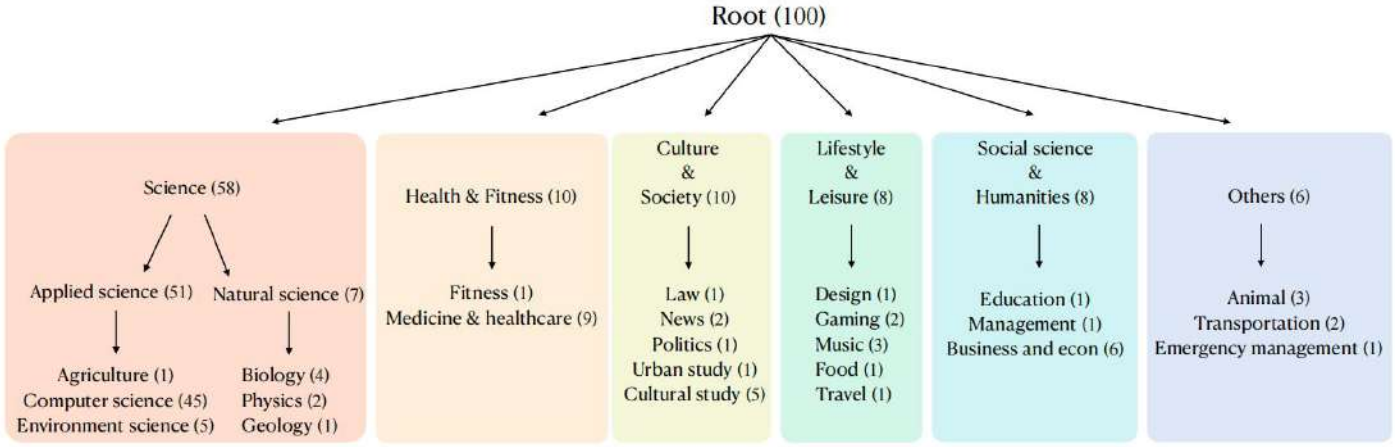
示例数据点

主题：探索可持续城市发展的创新模式

详细意图：寻找全球城市在应对气候变化和城市可持续性方面的前沿创新案例

背景：研究城市规划师和政策制定者如何通过技术和社会创新应对城市面临的环境挑战

预期信息： • 具体的城市可持续发展案例 • 创新技术和社会解决方案 • 跨地区的成功经验



Into the Unknown Unknowns: Engaged Human Learning through Participation in Language Model Agent Conversations.EMNLP 2024

DQA基于策略游戏模拟的复杂决策数据集

数据集目的与意义

- 聚焦于复杂战略性决策场景下的智能问答
- 模拟真实世界中需要深度推理和策略分析的决策环境
- 研究大语言模型在复杂决策情境中的推理能力
- 核心挑战：从海量数据中提取关键信息并做出明智决策

数据集类型

- 决策问答(Decision QA)基准数据集
- 包含关系型数据库(RDB)和图数据库(GDB)
- 两个场景：定位(Locating)和建设(Building)

数据收集方式

- 从两款历史策略游戏提取数据
- 从游戏存档文件解析相关数据
- 按预定义问题格式生成问题

数据集示例

典型决策问题场景

场景： 资源有限的战略游戏

问题： 如何在有限资源下最优化农场产能

挑战：

- 多维度数据分析
- 复杂约束条件下的决策
- 跨数据库的推理

数据集统计信息

指标	定位场景	建设场景
问题-数据库对数量	200	101
关系型数据库平均行数	2,038.8	579.0
图数据库平均边数	1,432.3	374.7

PlanRAG: A Plan-then-Retrieval Augmented Generation for Generative Large Language Models as Decision Makers. NAACL 2024

Chatbot Arena排行榜的系统性偏向

Chatbot Arena作为AI模型评估的重要平台，被发现存在严重的系统性公平性问题。研究团队揭示了一系列隐藏的排名操纵机制。

系统性问题：私密测试与选择性披露

私密测试政策

- 允许特定提供商进行大量并行测试
- Meta在Llama-4发布前测试了27个模型变体
- 不要求所有测试模型公开

选择性披露机制

- 可选择性发布最佳评分模型
- 公共排行榜不一定反映真实模型性能
- 为大型科技公司提供额外的竞争优势

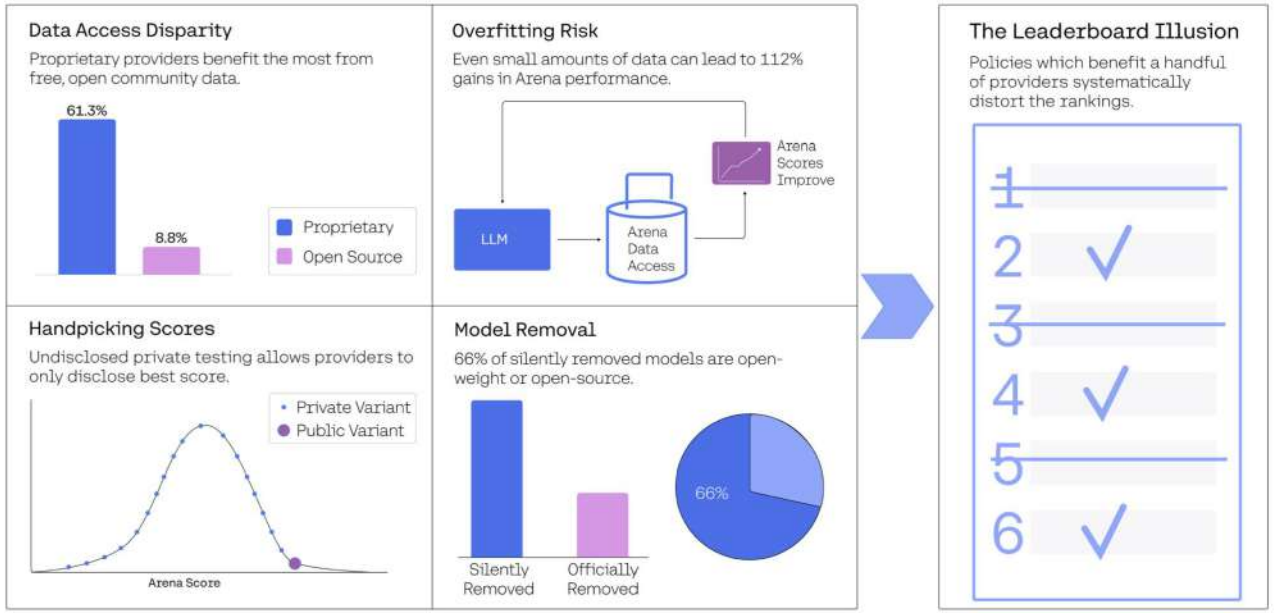
数据访问的不对称性

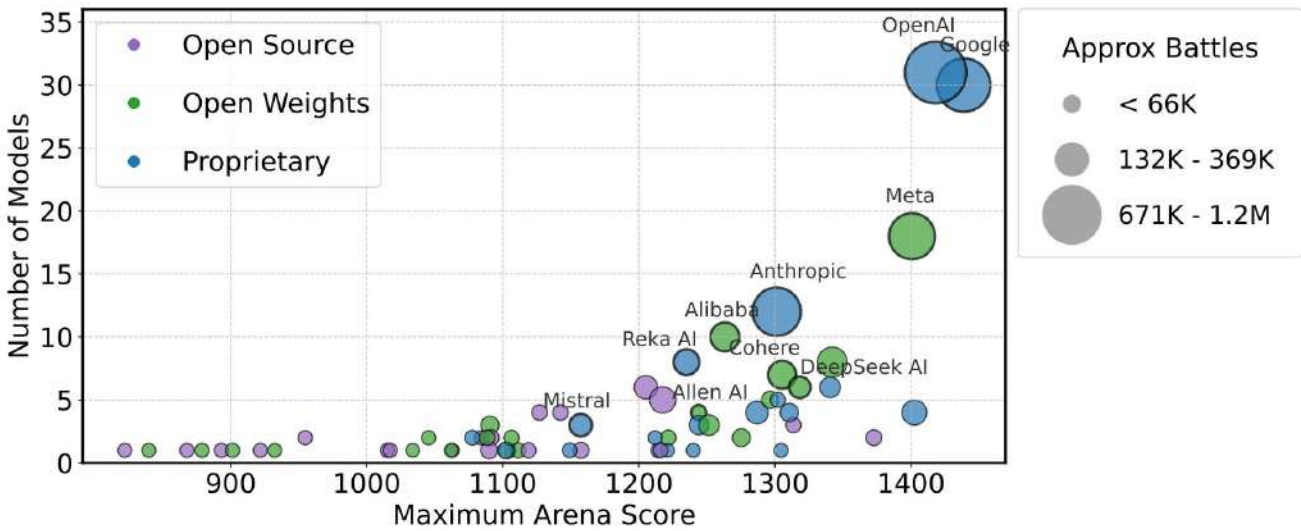
数据分配不均

- Google获得19.2%的数据
- OpenAI获得20.4%的数据
- 83个开放权重模型仅获得29.7%的数据

性能影响

- 数据访问显著提升模型排名
- 数据比例从0%增加到70%可使胜率翻倍
- 可能导致对特定Arena分布的过度拟合





The Leaderboard Illusion. Arxiv 2025.

Chatbot Arena引发对AI评估系统的思考

科学诚信

如何在快速发展的AI领域维护公平和透明?

评估机制

我们需要更加开放和去中心化的评估方法

长期发展

避免为了排名而优化，专注于真正的技术进步

学术界可以做什么？

标准化与治理

- 制定统一的AI模型评测伦理准则
- 建立透明的模型评测标准框架
- 开发去中心化的评测机制
- 创建独立的第三方评测委员会

研究方向

- 研究数据访问的不对称性
- 开发更公平的基准测试方法
- 揭示和量化排行榜操纵机制
- 构建跨平台的评测协议

长期愿景

学术界应该成为推动AI评测公平性的关键力量。通过系统性的研究、透明的治理和创新的方法，重建AI基准测试的科学信任，确保技术进步不会被少数几家大型科技公司所主导。

The Leaderboard Illusion. Arxiv 2025.

07 评测的新趋势

7-1. RAG 评测现状 | 7-2. OpenKG OneEval | 7-3. AGI-Eval

OpenKG-One Eval

http://oneeval.openkg.cn

- 提供基准测试平台，鼓励KG+LLM融合创新
- 提供客观的衡量标准，评估LLM在知识抽取、理解和应用能力
- 提高OpenKG在KG+LLM方面的影响力

OpenKG-SIGEval兴趣小组简介

知识图谱与大语言模型技术的快速发展

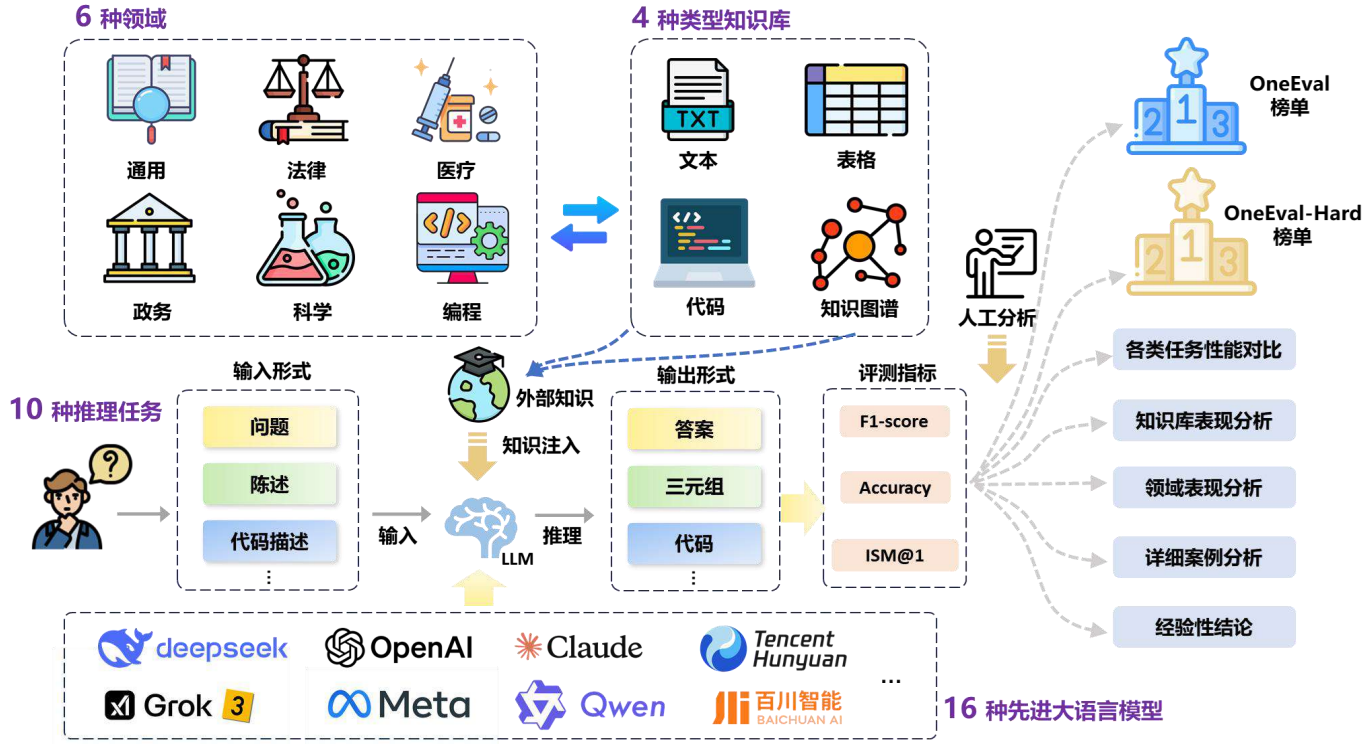
随着知识图谱和大语言模型技术的快速发展,如何有效评估知识图谱相关任务的质量以及知识图谱与大模型结合的效果,成为了当前亟需解决的关键问题。SIGEval兴趣组致力于探索和建立知识图谱评估的标准体系和方法论,涵盖从图谱构建、知识融合到实践应用的全流程评估。

评估知识图谱和大模型结合效果的重要性

重点关注知识图谱在大模型时代的新型评估需求,包括知识的准确性、时效性、完整性,以及与大模型结合后的表现等多个维度。通过开发开源评估工具、组织评测竞赛、制定评估标准等方式,推动知识图谱评估技术的进步,为知识图谱的质量提升和大模型的知识增强提供重要支撑。

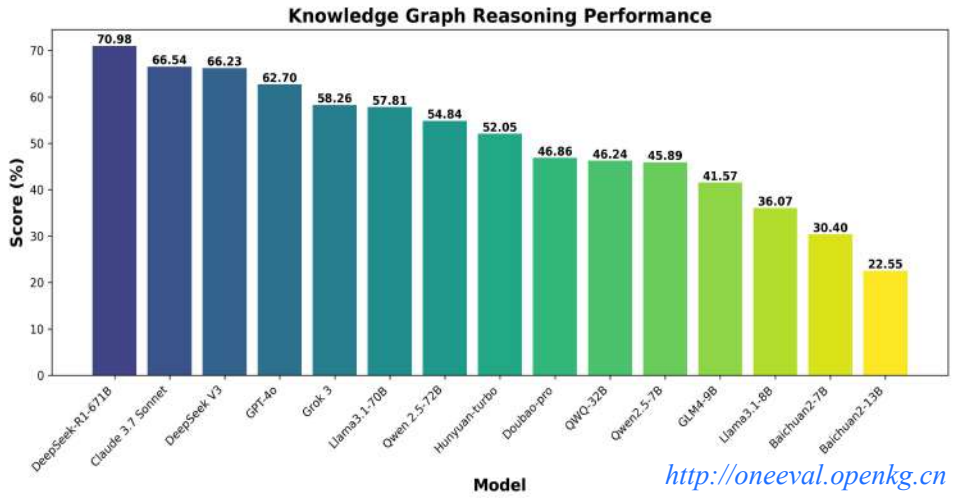
OneEval：大模型知识增强综合能力评测榜单

OneEval整体框架



Rank	Model	Overall Score (%)
1	Grok 3	26.57
2	OpenAI o1	26.17
3	Hunyuan-turbo	23.64
4	QWQ-32B	22.07
5	DeepSeek-R1-671B	19.7
6	Qwen2.5-72B	18.89
7	GPT-4o	17.79
8	Doubao-pro	16.46
9	Llama3.1-70B	16.45
10	Claude 3.7 Sonnet	15.62
11	GLM4-9B	15.42
12	DeepSeek-V3	13.68
13	Llama3.1-8B	10.83
14	Baichuan2-13B	10.58

Rank	Model	Overall Score	WTQ	PersonQA	Report	MedicalQA	PoliticalQA	BioQA	MaterialQA	PharmKGQA	ChineseLawFact	VersiCode
1	Grok 3	55.82%	76.50%	4.70%	77.80%	49.00%	45.50%	80.00%	64.29%	42.11%	54.25%	64.00%
2	QWQ-32B	50.65%	70.50%	3.00%	32.30%	78.30%	45.00%	76.67%	62.38%	45.67%	69.00%	23.70%
3	Hunyuan-turbo	50.10%	55.10%	1.40%	2.20%	84.50%	43.00%	85.71%	60.95%	32.52%	83.87%	51.70%
4	Qwen 2.5-72B	50.02%	65.50%	2.50%	38.90%	59.50%	45.00%	81.43%	62.86%	38.09%	70.50%	35.90%
5	GPT-4o	48.49%	69.40%	3.20%	44.70%	59.00%	41.00%	43.81%	61.43%	39.23%	56.63%	66.50%
6	DeepSeek-R1-671B	47.36%	74.30%	6.80%	59.70%	48.00%	45.50%	33.81%	50.48%	31.37%	58.00%	65.60%
7	DeepSeek V3	45.83%	69.90%	2.60%	57.90%	59.50%	42.50%	55.71%	39.90%	39.04%	53.87%	37.40%
8	Llama3.1-70B	44.57%	47.70%	2.20%	24.20%	27.00%	40.00%	88.57%	71.43%	34.33%	59.38%	50.90%
9	Doubao-pro	44.54%	46.00%	0.00%	25.30%	53.00%	40.00%	83.33%	50.00%	27.14%	57.50%	63.10%
10	GLM4-9B	41.24%	39.10%	0.20%	6.60%	48.50%	38.50%	80.95%	58.10%	17.70%	66.25%	58.50%
11	Claude 3.7 Sonnet	38.48%	28.30%	0.50%	42.30%	46.00%	22.10%	78.10%	48.80%	40.10%	60.38%	18.20%
12	Qwen2.5-7B	32.93%	30.60%	0.50%	17.00%	34.50%	46.00%	50.95%	37.50%	31.55%	62.88%	17.80%
13	Llama3.1-8B	30.11%	35.70%	0.20%	2.50%	17.00%	42.00%	55.23%	55.98%	23.53%	57.13%	11.80%
14	Baichuan2-7B	24.80%	4.80%	0.00%	12.00%	20.00%	43.50%	51.43%	50.95%	21.43%	43.87%	0.00%
15	Baichuan2-13B	24.74%	14.80%	0.00%	13.60%	26.50%	37.00%	57.14%	22.86%	14.76%	56.63%	4.10%



http://oneeval.openkg.cn

07 评测的新趋势

7-1. RAG 评测现状 | 7-2. OneEval | 7-3. AGI-Eval

AIceping / AGI-Eval

引言

我们是谁：AGI-Eval大模型评测社区

AGI-Eval评测社区

定位

整合多家高校
大厂资源的
独立第三方
大模型评测团队

愿景

评测助力
让**AI**成为
我们**更好的伙伴**

使命

打造价值驱动的
评测生态

AGI-Eval评测社区

我们的使命：打造价值驱动的评测生态

评测研究团队

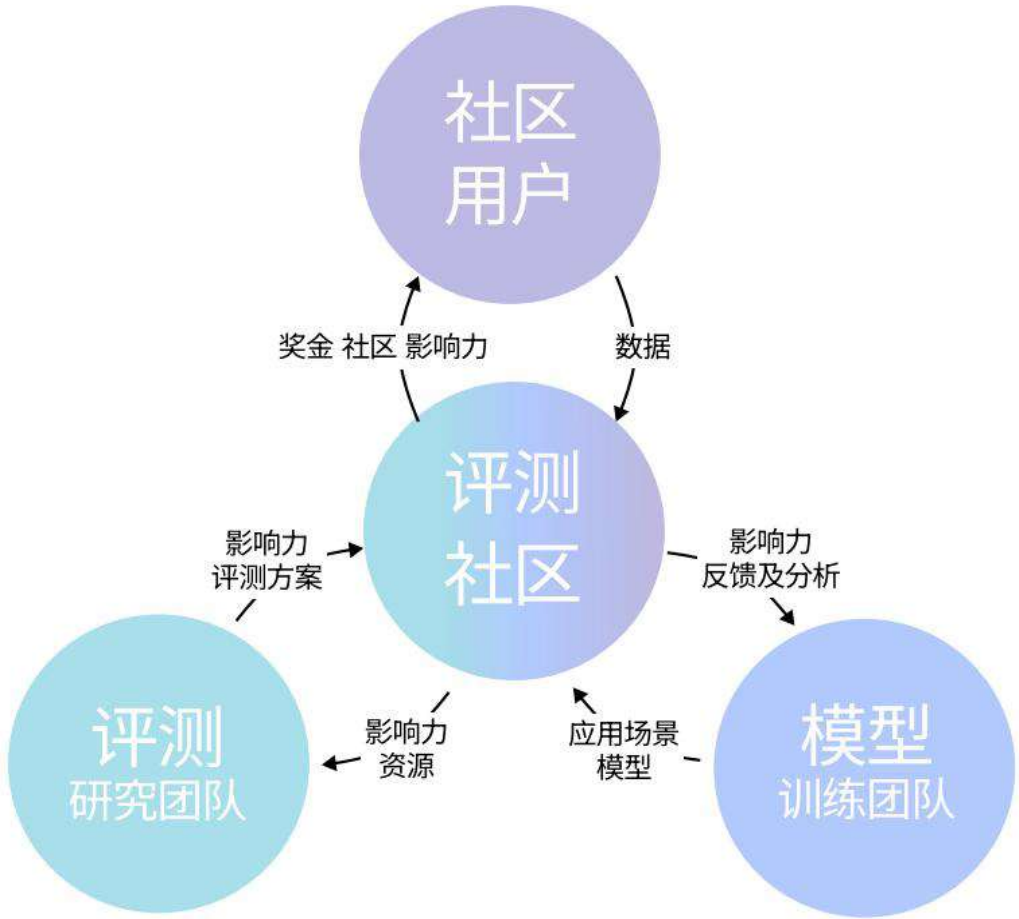
- 贡献 – 高质量的评测方案、数据集
- 收获 – 评测方案曝光与影响力、长期托管与维护

模型训练团队

- 贡献 – 基座模型、精调模型、应用场景
- 收获 – 评测结果曝光与影响力、评测反馈与分析

下游应用开发者、普通用户

- 贡献 – 用户数据
- 收获 – 认知、社区影响力、奖金



我们做了什么?

团队投入

AGI-Eval评测社区

完善的评测体系

10+ 组评测指标

100+ 细分能力项

全面的评测手段

10+ 组评测手段

海量的评测任务

1000 组评测任务

10w 级别评测数据

私有的评测数据

100% 私有化数据

60000 次模型评测

我们做了什么?

模型榜单

AGI-Eval评测社区

https://agi-eval.cn/mvp/home

我们做了什么？

评测集社区

AGI-Eval评测社区

完善的评测体系

- 10+组评测指标、100+细分能力项

全面的评测手段

- 自动与人工评测结合、10+组评测手段

海量的评测数据

- 近千个评测子集、10万级别评测数据

私有的评测数据

- 黑盒100%私有化数据

我们做了什么?

Data Studio

- 日活 1000+
- 数据总量 35w+
- 用户数量 2.5w+
- 任务标签 500+

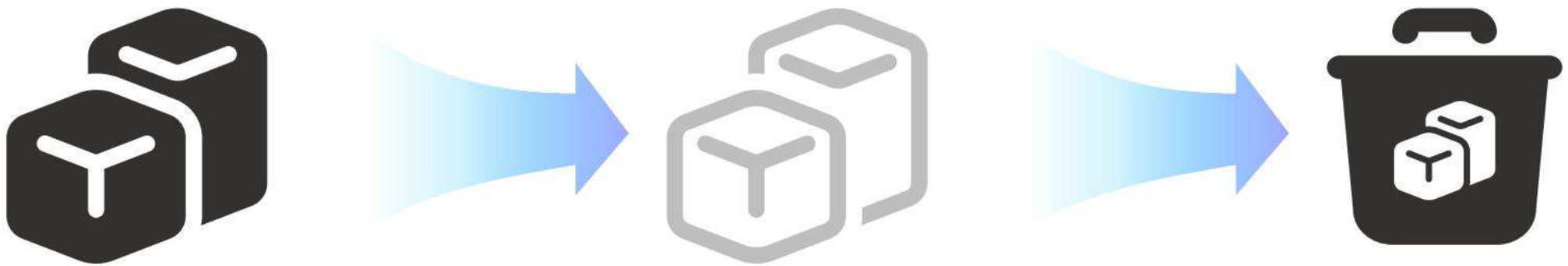
黑白盒数据流转机制

从数据穿越到分布穿越

- 大模型已经具备很强的泛化（分布穿越）能力
- 当评测数据被公开，评测数据距离失效就已经不远
- 定向刷榜：达到更高的榜单性能不再需要数据穿越

建立有效的数据流转机制

- 什么样的数据更难被分布穿越?
- 当前的数据集中有哪些正在被分布穿越?
- 如何权衡防止穿越与反馈训练?



未来做什么?

AGI-Eval评测社区

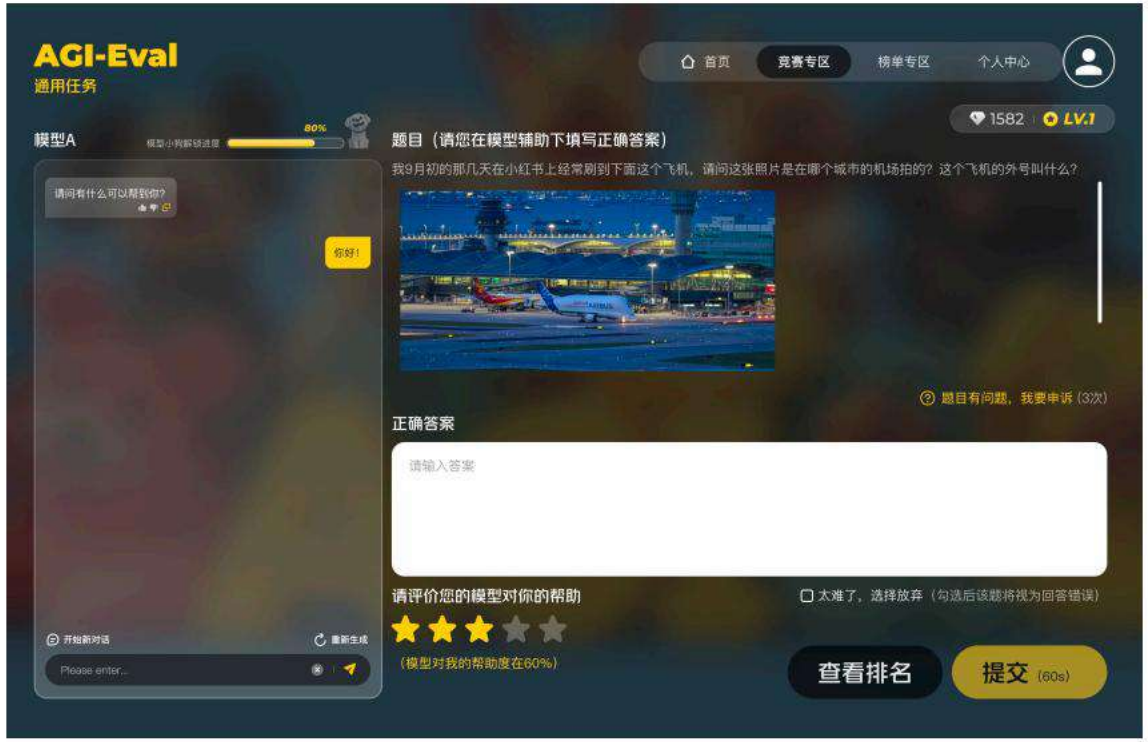
人机协作评测方案

高难度任务

- 任务过难，缺乏评测区分度
- 引入人机协作增加任务完成度，建立区分度
- 通过对人机协作过程的分析揭示潜在的模型优化方向

人机交互类任务

- 以模型对人的增益作为任务的目标：教育、情感陪伴
- 直接评估应用对人的增益
- 剥离应用评估模型的能力

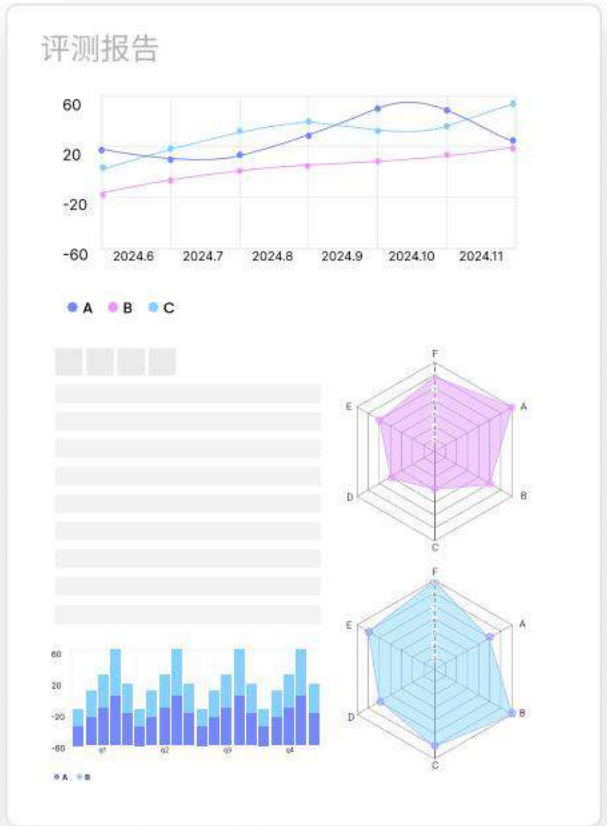


AGI-Eval评测社区

自动化评测分析报告

评测能为训练提供什么?

- 细粒度的模型能力类目指标
- 错误类目构建与分析
- 小规模灵活评测建立新的认知
- 端到端产出自动化评测分析报告



08 RAG 应用实践

8-1. ToC应用场景 | 8-2. 领域应用场景 | 8-3. 隐性成本与实践指南

Deep Research

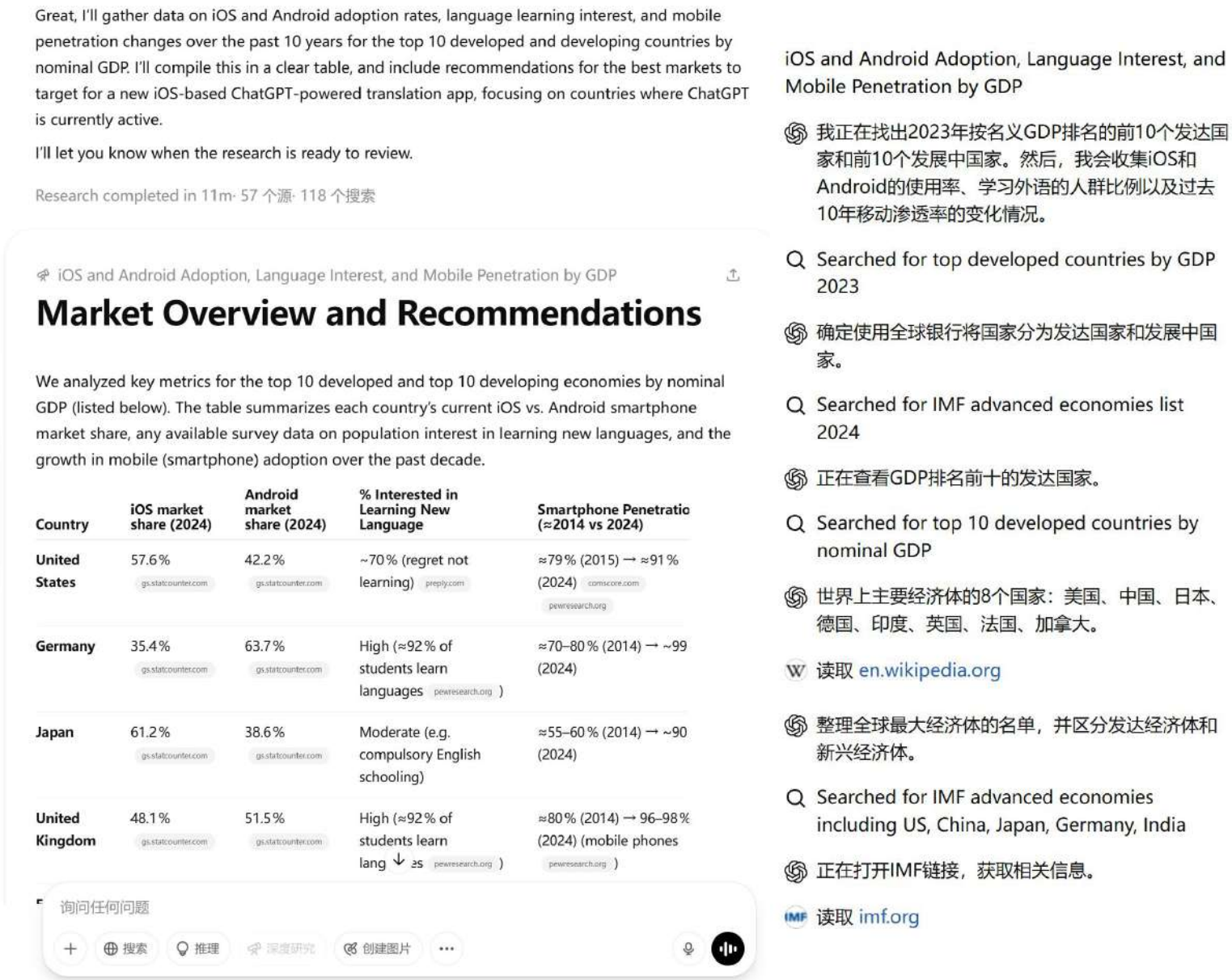
OpenAI: Deep Research

适用场景：

开发者、数据分析师、研究人员等，需要快速检索并分析大量网络内容或代码库的场景

特性：

- 基于优化的GPT-o3模型，支持长文本、多模态数据分析
- 可连接GitHub，实时读取仓库并解析代码、README等文档
- 深度研究准确度、深度和全面性处于领先地位。

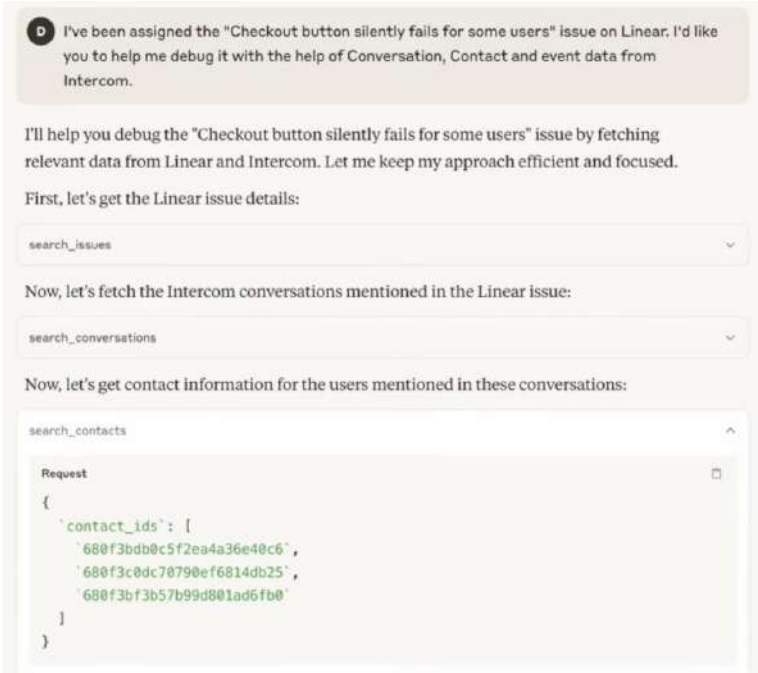


Deep Research

Anthropic: Claude DeepResearch

适用场景：

企业团队、知识工作者和研究人员等，需要整合公司内部工具（如 Google Workspace、Jira 等）和互联网信息的场景。



特性：

- 基于MCP协议，可接入多种应用（Atlassian、Zapier 等），获取项目历史、任务状态等上下文。

Sider Deep Scholar

适用场景：

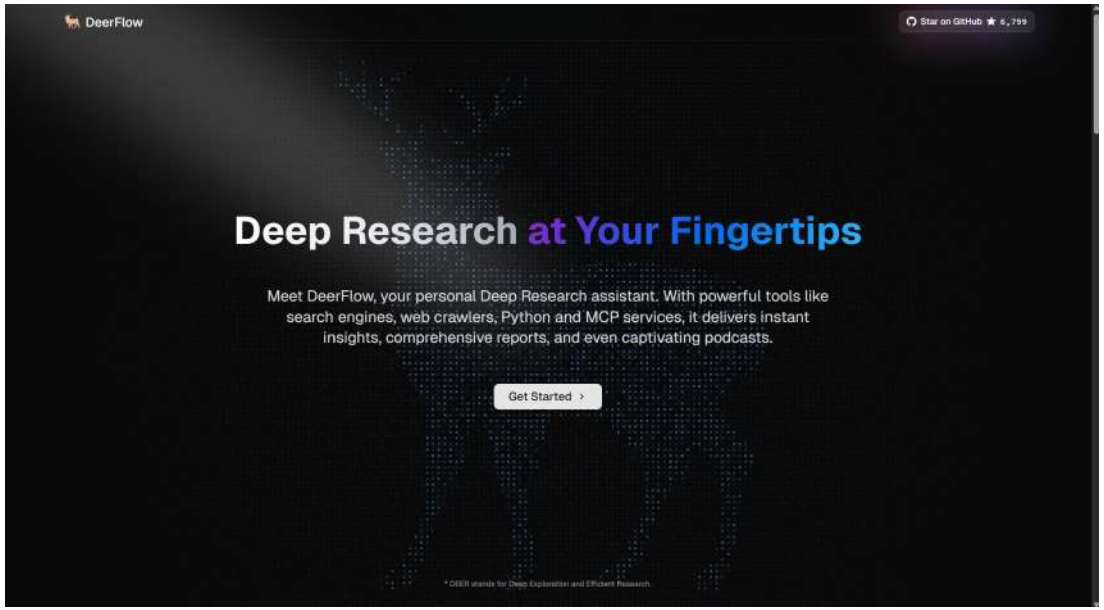
学者、学生和科研人员等需要撰写文献综述、研究报告的场景。

特性：

- 全文学术检索（3.5亿+论文），连接多家权威学术数据库（Nature、IEEE、Springer 等），自动更新最新文献
- 提供逐条来源验证和交互式细化功能，提高研究效率

Deep Research

DeerFlow 复杂推理任务+知识检索+多模态输出



特性：

- 搜索和检索增强，通过 Tavily、Brave Search、Jina 等工具进行网络搜索、支持高级内容提取功能。
- MCP 集成，扩展私有域访问、知识图谱、Web 浏览功能。
- 播客和演示文稿生成，AI 驱动的播客脚本生成和音频合成，自动创建 PowerPoint 演示文稿。

模块化的多代理系统架构：

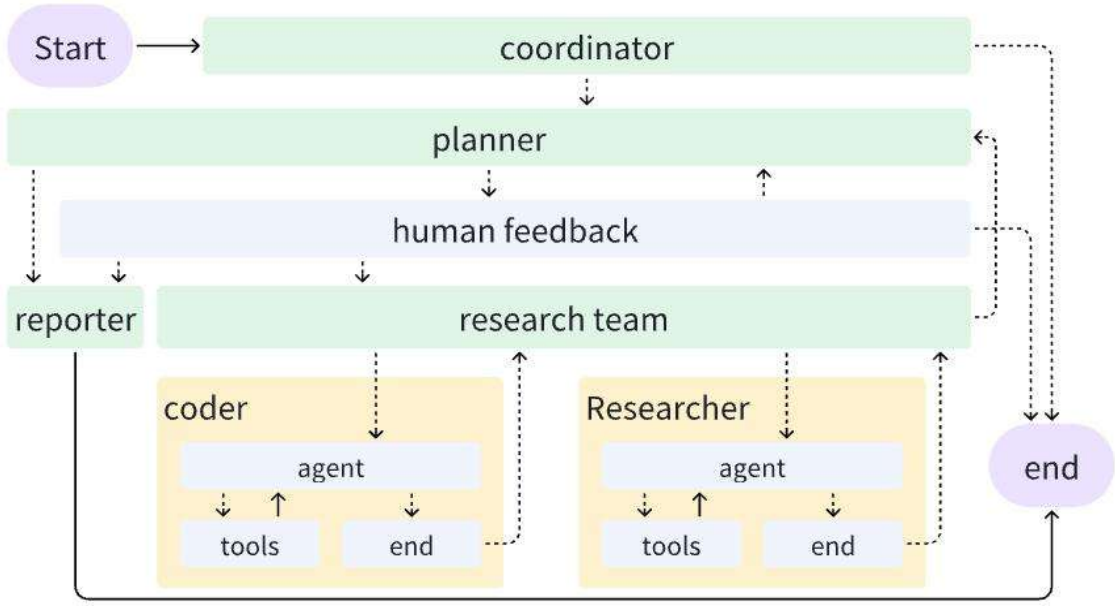
- 基于 LangGraph 构建，支持灵活的基于状态的工作流，组件通过定义明确的消息传递系统进行通信。
- 采用简化的工作流，包括以下组件：

  Coordinator： 管理工作流生命周期的入口点。

  Planner： 任务分解和规划的战略组件。

  Research Team： 执行计划的专业代理的集合。

  Reporter： 研究成果的最终阶段处理器。



https://github.com/bytedance/deer-flow

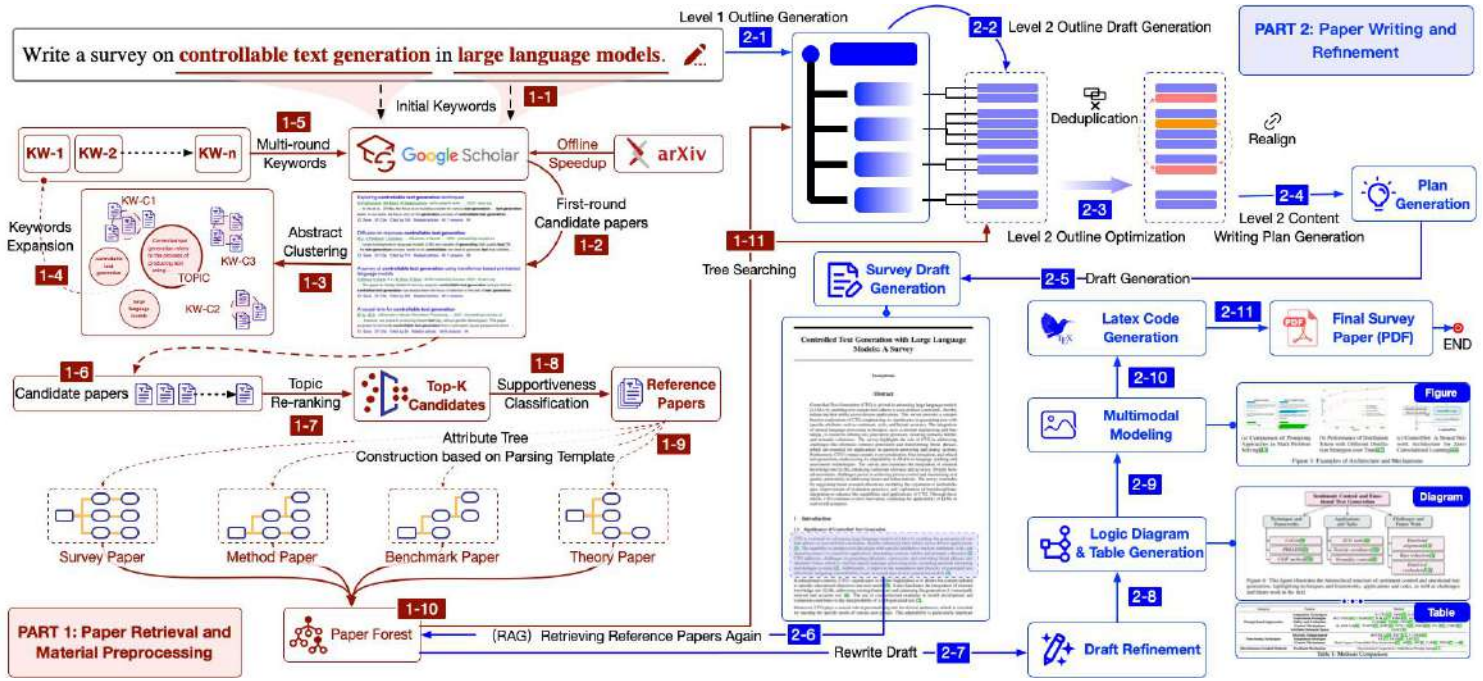
Deep Research

学术Survey: Surveyx

全面准确和最新参考文献数据 + 深入的内容讨论

大纲生成、主体生成和后期细化机制：

- 多轮关键词检索、语义聚类与迭代扩展，构建高质量的候选论文池并搭建按类型组织的“属性树”，生成一级大纲，为后续写作提供结构框架。
- 细化大纲、制定写作计划、调动LLM逐节产出与重写草稿，通过RAG补全引用并多轮精炼，最后生成多模态图表和LaTeX 文档。



Model	Coverage	Structure	Relevance	Synthesis	Critical Analysis	Avg	Recall	Precision	F1
naive RAG	4.40	3.66	4.66	3.82	2.82	3.872	68.79	61.97	65.20
AutoSurvey	4.73	4.33	4.86	4.00	3.73	4.331	82.25	77.41	79.76
SurveyX	**4.95**	**4.91**	**4.94**	**4.10**	**4.05**	**4.590**	**85.23**	**78.12**	**81.52**
Human	5.00	4.95	5.00	4.44	4.38	4.754	86.33	77.78	81.83

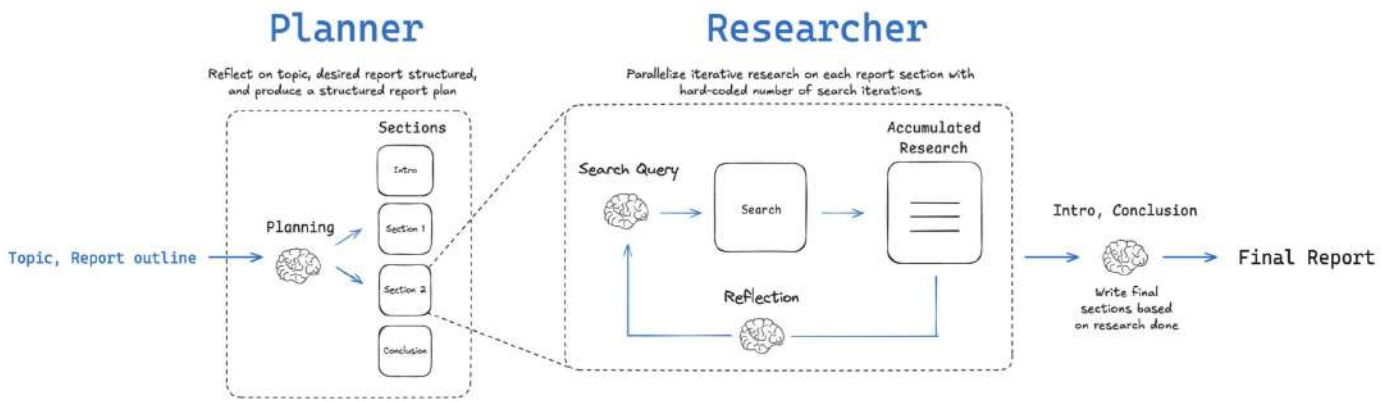
左图展示了NaiveRAG、Autosurvey、SurveyX和人类写作的内容质量评估结果。所有的大模型智能体均为GPT-4o。其中SurveyX的方法取得了**更高的得分且接近人类写作水平。**

SurveyX: Academic Survey Automation via Large Language Models, arXiv, 2025.

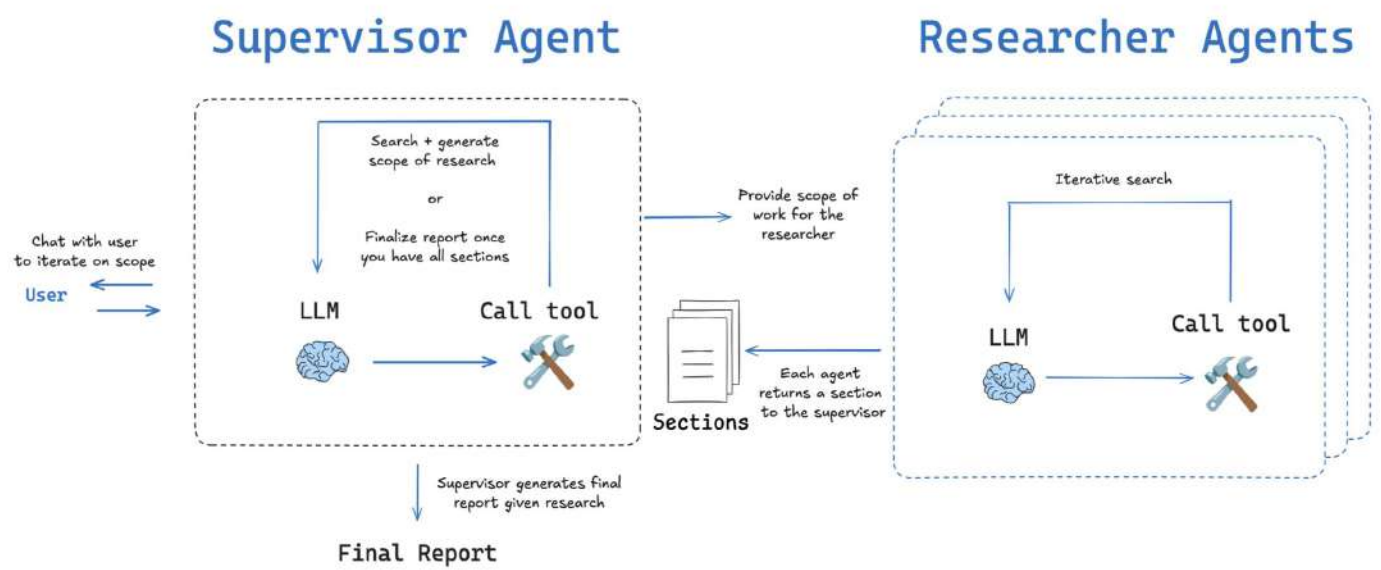
Deep Research

Open Deep Research 两种路径

Workflow



Multi-agent



https://github.com/langchain-ai/open_deep_research?tab=readme-ov-file

● 基于图的工作流实现

基于图的实现遵循结构化的“计划-执行”工作流程：

- **计划阶段：**使用规划模型来分析主题并生成结构化的报告计划
- **用户反馈环节：**在继续执行前允许用户提供反馈并审批报告计划
- **顺序研究流程：**逐节创建内容，在每次搜索迭代之间进行反思
- **针对性研究：**每一节都有专属的搜索查询和内容检索策略
- **支持多种搜索工具：**兼容所有搜索提供方（如Tavily、Perplexity、Exa、ArXiv、PubMed、Linkup等）

该实现方式提供了更高的交互性与对报告结构的控制，非常适合对报告质量和准确性要求较高的场景。

● 多智能体实现

多智能体实现采用“监督者-研究员”架构：

- **监督者：**管理整体研究流程、规划各章节并汇总最终报告
- **研究员：**多个独立agent并行工作，各自负责研究并撰写特定章节
- **并行处理：**所有章节同时进行研究，显著减少报告生成时间
- **专用工具设计：**每个agent配备其职责所需的专用工具（研究员用于搜索，监督者用于章节规划）
- **当前仅支持Tavily搜索：**目前该实现只支持Tavily作为搜索引擎，但框架设计上支持未来扩展至更多搜索工具

该实现侧重于效率与并行化，适合快速生成报告且无需大量用户干预的场景。

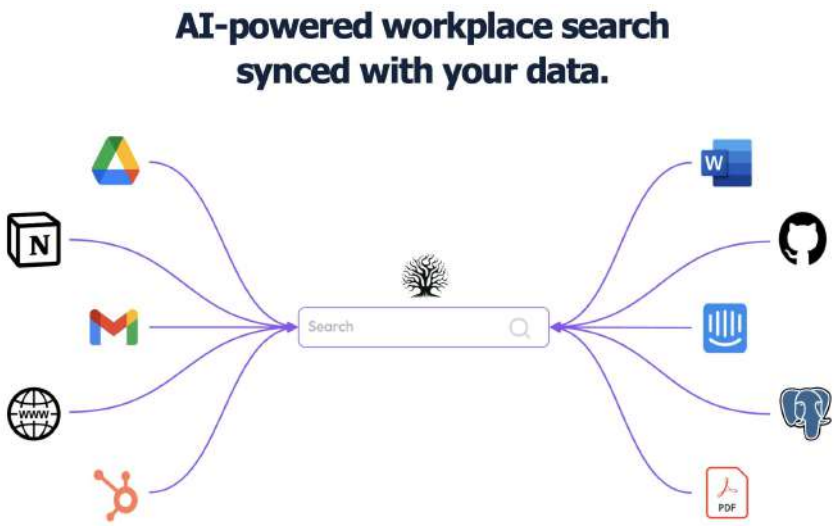
个人知识助手

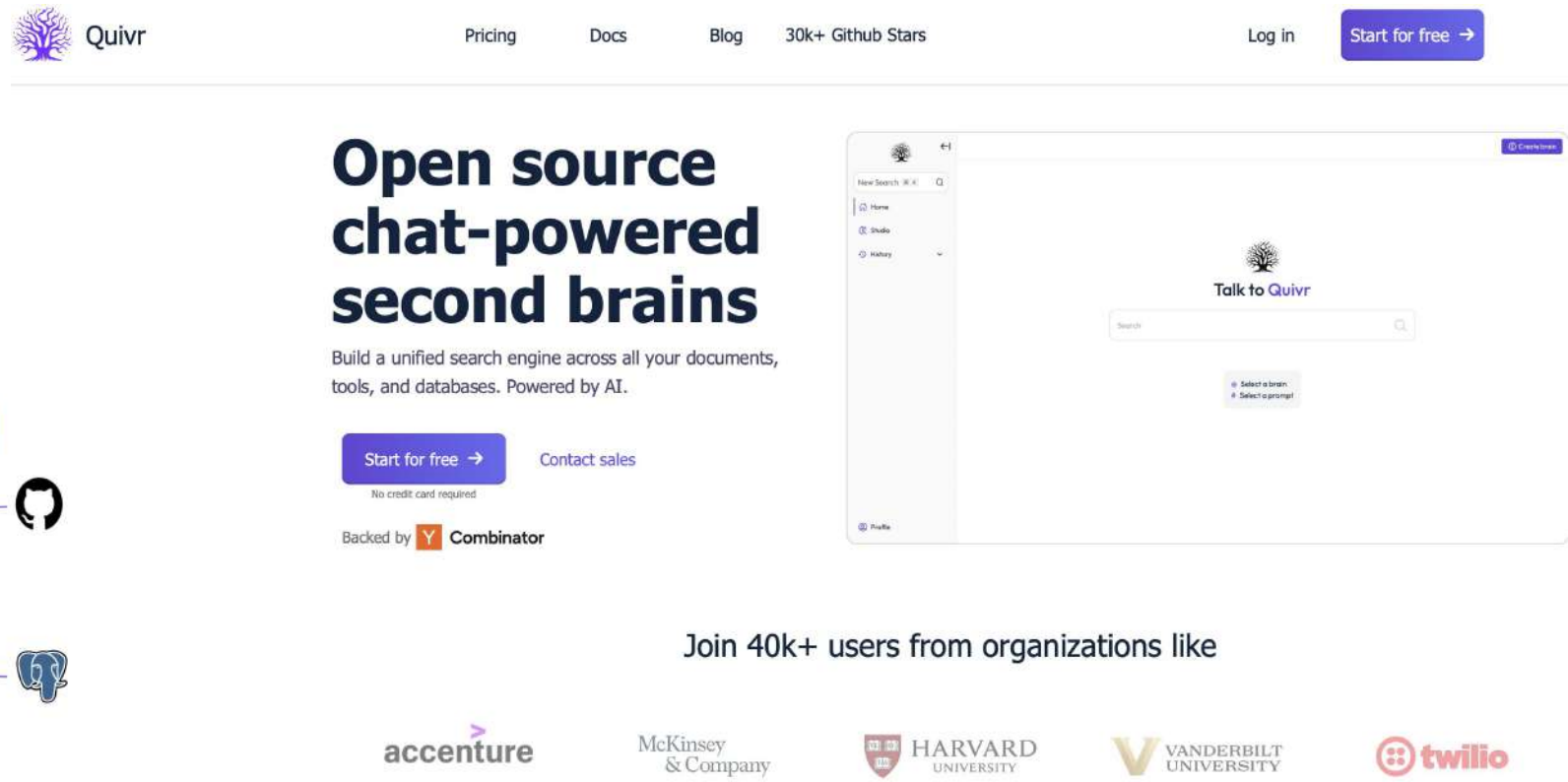
Quivr: Your Second Brain

整合多源文档语料

特性：

- 高效地存储和检索信息。
- 支持多种文件类型，包括文本、Markdown、PDF、音频和视频。





个人服装推荐助手

StePO-Rec

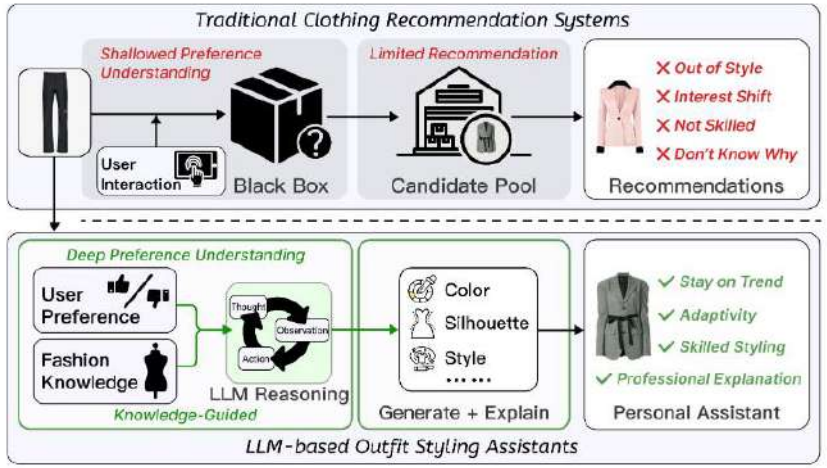
复杂推理任务+个性化+造型专业知识

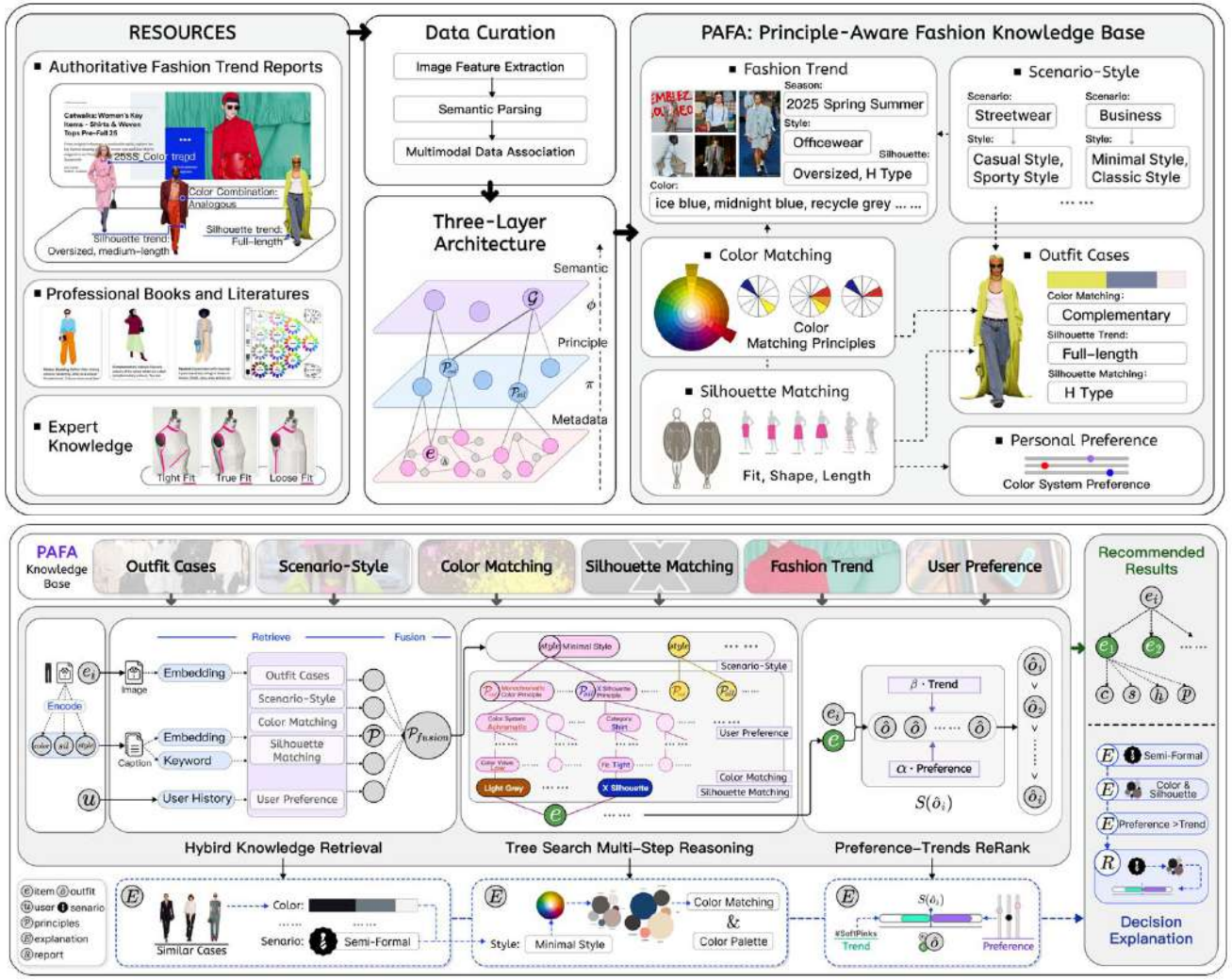
知识引导的多步推理机制：

- 通过树状决策结构将穿搭推荐问题分解为场景→风格→属性子任务，构建层次化推理路径，明确专业原则与用户偏好的依赖关系。

动态混合检索机制：

- 推荐生成过程中，基于PAFA知识库实时检索场景规则、颜色搭配等子模块，结合用户历史数据动态调整检索权重，提升个性化适配能力。





StePO-Rec: Towards Personalized Outfit Styling Assistant via Knowledge-Guided Multi-Step Reasoning, arXiv, 2025.

08 RAG 应用实践

8-1. ToC应用场景 | 8-2. 领域应用场景 | 8-3. 隐性成本与实践指南

垂域知识服务典型要求

私域文档&数据

建索引 + 检索 + 推理 + 生成

?

专业性知识服务

专业问答
法律、政务、医疗、科学

写作助手
新闻稿、研报、分析

场景

知识精准
知识完备
逻辑严谨
时间敏感
数值敏感

要求

1. 错误定性或错误逻辑
2. 事实性错误或无依据
3. 时间、数值不敏感
4. 张冠李戴
5. 不能区分重要性
6. 语义不精准
7. 召回不完备

RAG 与推理结合的场景

兼顾了外部知识依赖和复杂推理的需求，使得LLM更加胜任复杂的真实场景。

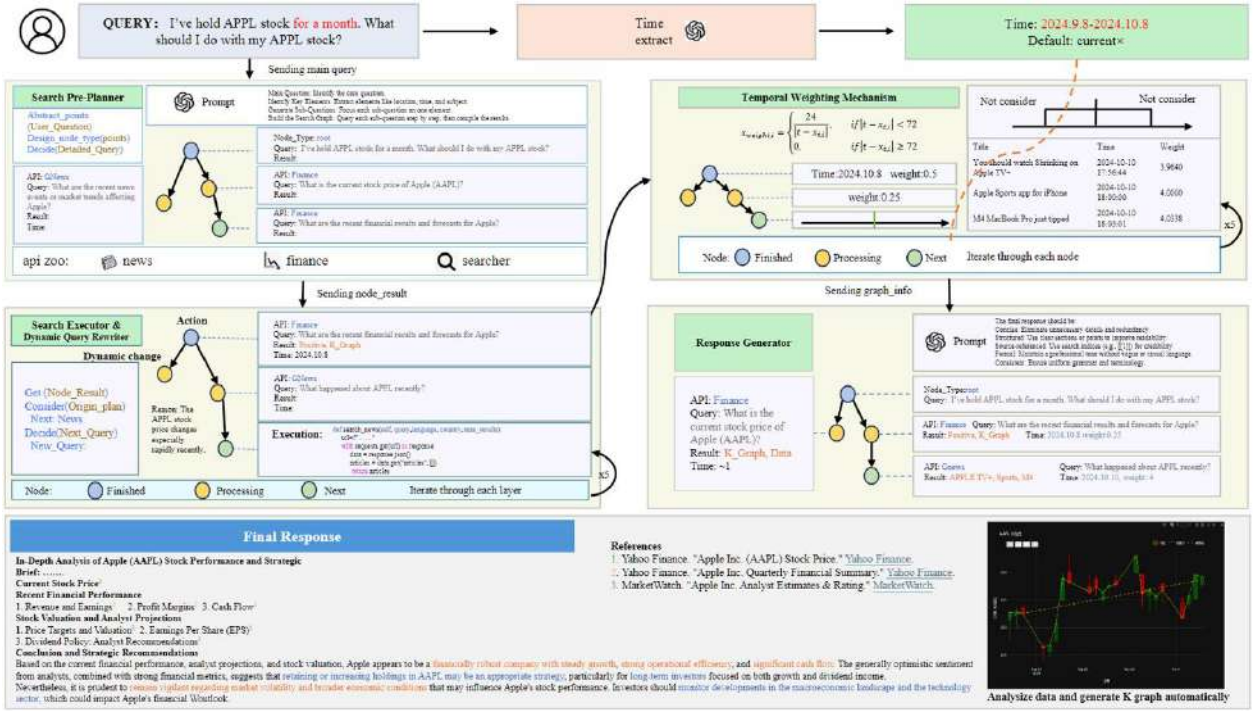
金融：FinSearch

- **金融：复杂推理任务+高频更新的外部专业知识**

动态自适应搜索机制

- 分步查询规划：通过LLM将复杂金融问题分解为子查询，构建DAG，明确逻辑依赖关系。
- 实时查询优化：在搜索执行过程中，基于中间结果动态调整后续子查询，提升市场动态响应能力。

时间敏感加权机制: 引入72小时时间窗口的线性衰减函数，优先处理近期信息（如新闻、股价波动），强化时间相关性。



Method	Accuracy (%)					Time (s/answer)				
	GPT-4o	Llama3.1-405B	Claude3.5-Sonnet	Deepseek	Gemini-1.5-Flash	GPT-4o	Llama3.1-405B	Claude3.5-Sonnet	Deepseek	Gemini-1.5-Flash
Baseline	36.13 ±1.22	38.13 ±1.27	38.60 ±1.28	34.07 ±1.21	35.47 ±1.22	3.87 ±0.15	4.96 ±0.25	6.66 ±0.21	14.50 ±0.27	5.04 ±0.16
SearchAgent	46.33 ±1.30	43.80 ±1.24	41.87 ±1.29	44.27 ±1.26	42.33 ±1.28	**1.58** ±0.06	**1.61** ±0.07	**1.54** ±0.06	**1.32** ±0.04	**1.58** ±0.04
MindSearch	52.40 ±1.33	53.07 ±1.34	53.60 ±1.28	49.73 ±1.32	51.53 ±1.29	19.09 ±0.39	14.82 ±0.45	17.93 ±0.39	27.01 ±0.58	20.14 ±0.43
Perplexity Pro	60.27 ±1.26	61.47 ±1.28	56.67 ±1.24	–	–	6.12 ±0.26	3.94 ±0.15	5.85 ±0.22	–	–
Ours	**76.20** ±1.12	**75.53** ±1.08	**78.27** ±1.07	**72.33** ±1.15	**74.87** ±1.08	16.03 ±0.43	14.55 ±0.47	18.15 ±0.35	29.31 ±0.70	17.74 ±0.53

上图展示了不同智能体方法在金融基准FinSearchBench-24上的性能和计算效率的对比，其中本文的方法取得了**更高的准确率与更低的计算开销**

An Agent Framework for Real-Time Financial Information Searching with Large Language Models, 2024.

KG +LLM的应用场景

KG+LLM 的结合，兼顾了外部知识依赖和复杂推理的需求，使得LLM更加胜任复杂的真实场景。

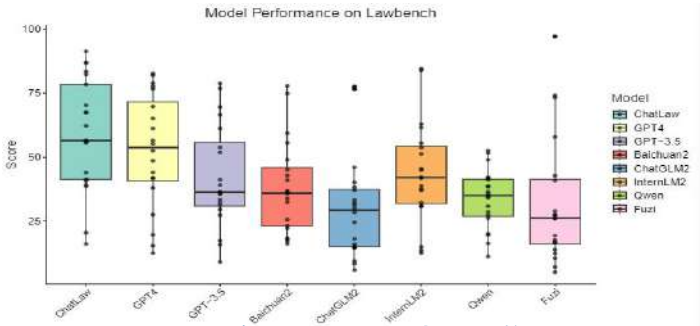
法律：Chatlaw

- **法律：复杂推理任务+时效性条文适配**

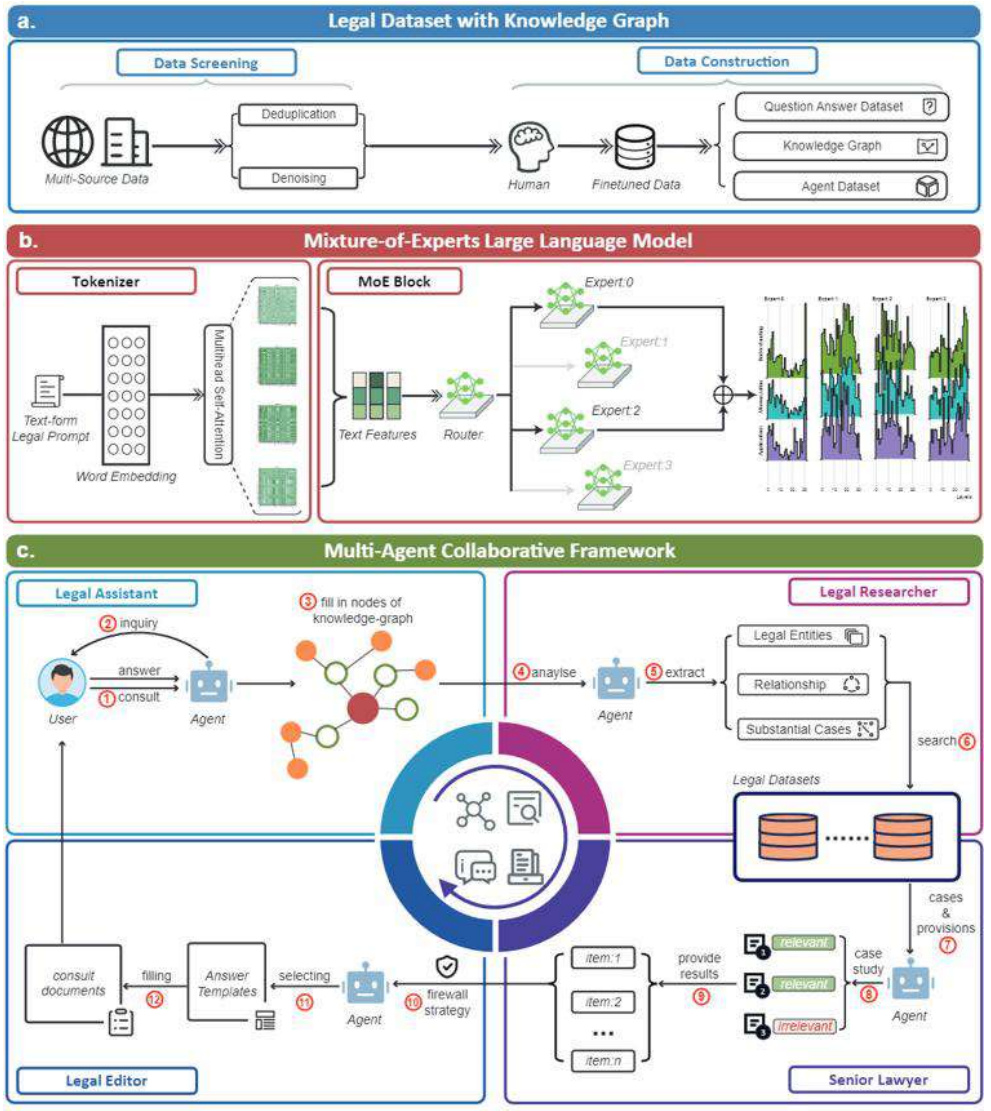
法律数据集构建：通过重复数据删除和降解以获取标准化的法律问答数据集。通过专家标注生成KG和智能代理任务数据集。

混合专家模型构建：设计MoE模型，并采用动态路由网络根据输入特征激活对应专家模块，实现罪名认定与量刑建议的精准匹配。通过对抗性样本注入与专家负载均衡约束，增强模型对非常规输入（如模糊表述、隐蔽诱导）的抗干扰能力。

多智能体协作推理：采用法律助理、研究员、资深律师、编辑员四智能体协作，模拟真实律所SOP工作流。对接裁判文书网等官方源，动态维护法律条文、典型案例与司法解释的时效性关联。



上图展示了Chatlaw在公开法律基准Lawbench上的效果，以及和其他通用LLM的对比，**超过了GPT-4**。



Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model, 2023.

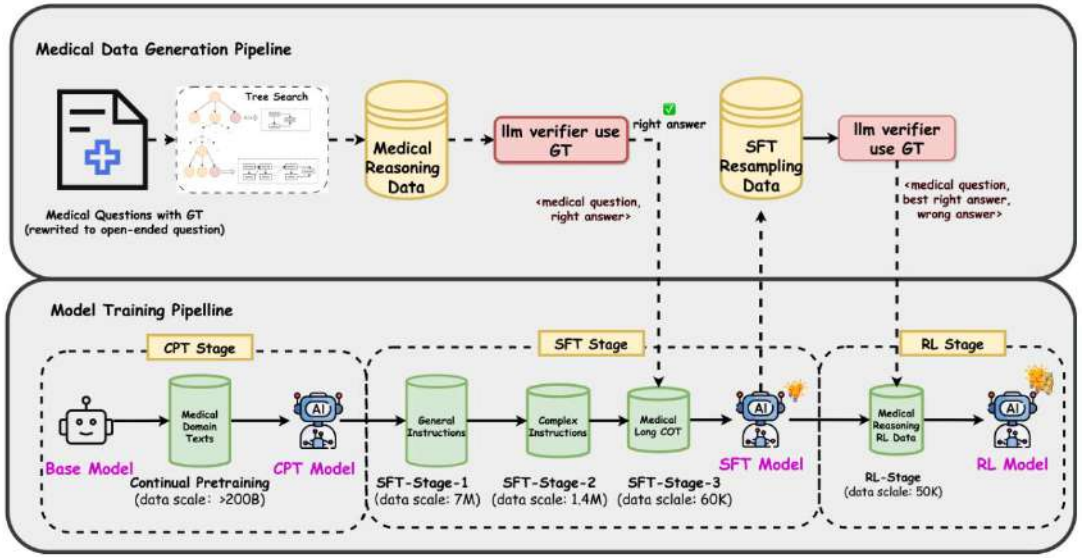
RAG 与推理结合的场景

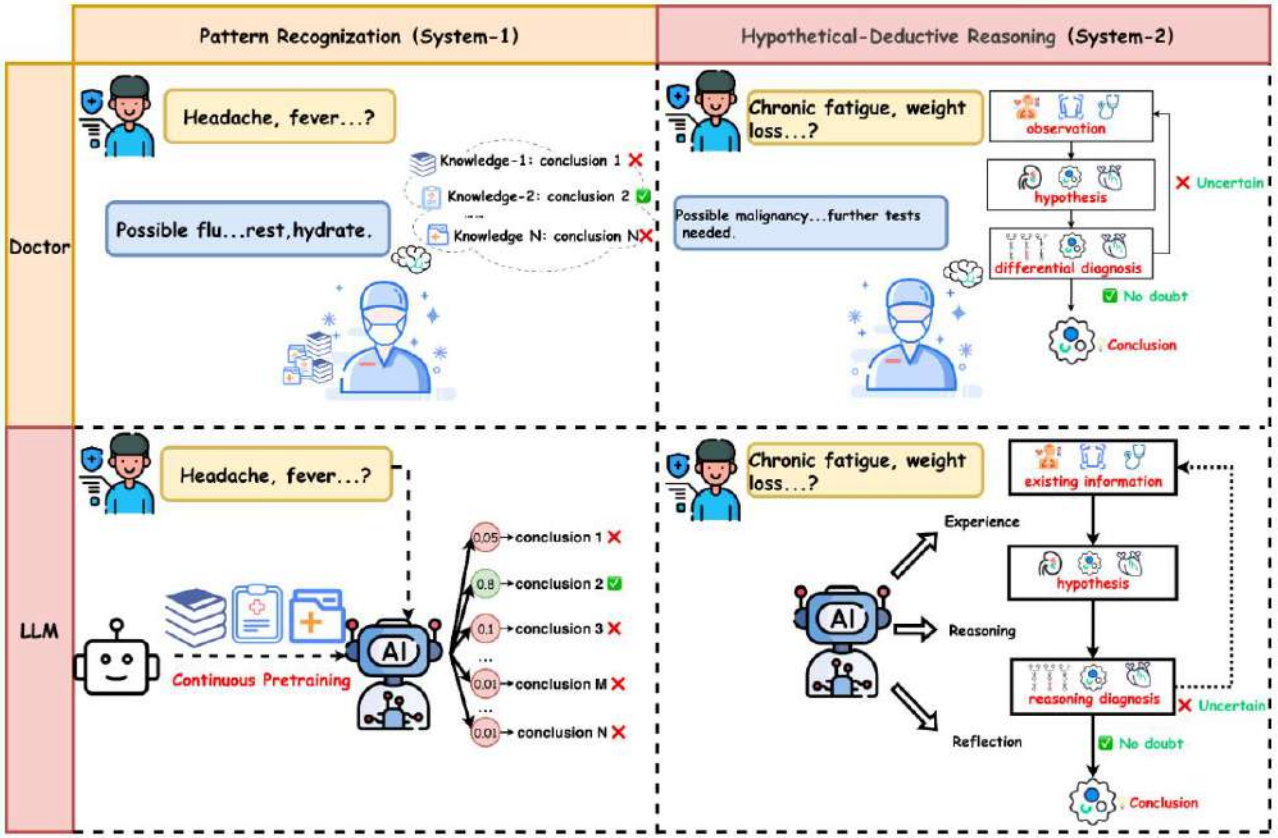
医疗：Citrus 复杂推理任务+强领域知识+低幻觉容忍

模拟专家认知路径：提出结合“假设 - 演绎推理”和“模式识别”的双专家推理框架，模仿医生的临床决策过程。

双专家推理机制：

- 推理专家：生成假设 - 演绎推理链。
- 反思专家：基于真实答案验证推理逻辑，过滤无效步骤。





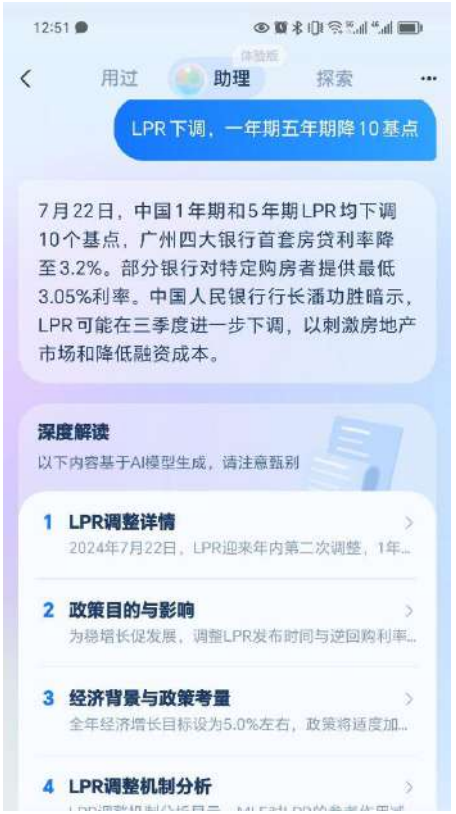
通过SFT和RL的训练，LLMs表现出与医学专家相似的临床决策路径。

Citrus: Leveraging Expert Cognitive Pathways in a Medical Language Model for Advanced Medical Decision Support, arXiv, 2025.

RAG企业级应用场景

分析写作

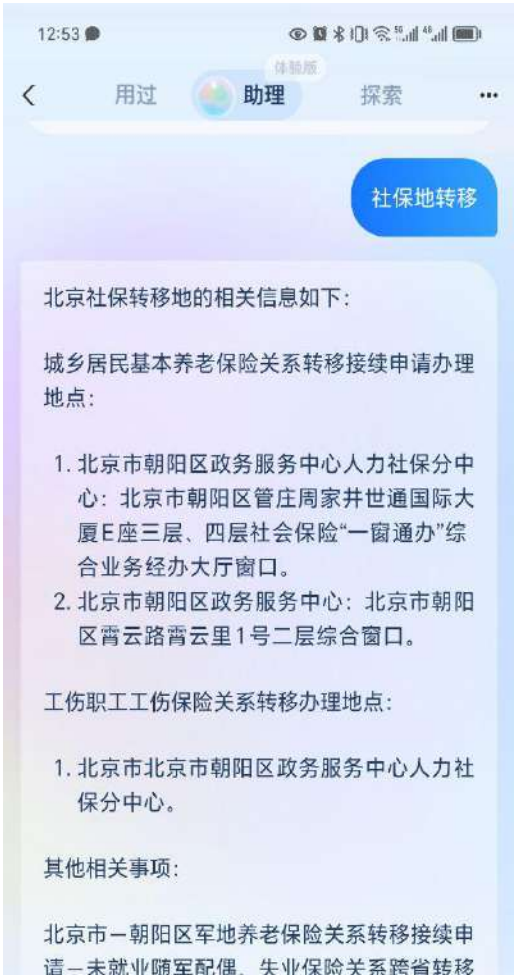
热点事件解读



银行风险分析

知识问答

政务办事问答



医疗健康问答

RAG企业级应用场景

保定市提取公积金需要哪些材料

购买自住住房提取住房公积金

不全，提取公积金的12种不同情况

生育险在哪里看

生育津贴支付

检索错误，参保人员参保信息查询

	有果 准确率	召回率
基本RAG	0.55	0.37
知识增强RAG	0.91	0.71

舟山市怎么查房产证

未找到相关信息

遗漏，不动产权属证明网上查询

社保月缴费多少

职工参保登记

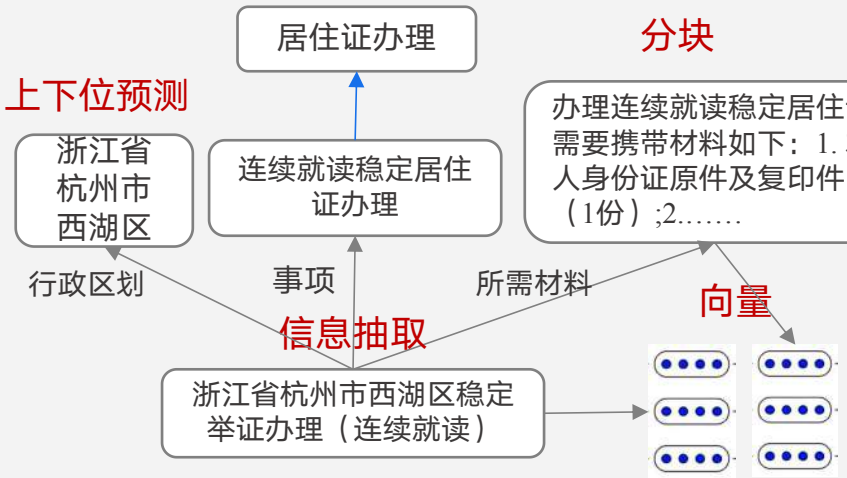
检索错误，没有事项

我买房申请了500万公积金贷款，计划20年还清，当前月利率是0.02，每个月应该还多少钱

已知公积金贷款月供计算公式为[贷款本金×月利率×(1+月利率)^还款月数]÷[(1+月利率)^还款月数-1]
算式：500万×0.02×（1+0.02）^(20×12)÷[(1+0.02)^(20×12)-1

RAG企业级应用场景

离线索引



UIE+概念推理+向量化 「建立互索引」

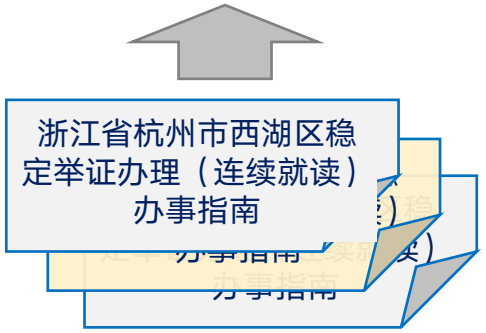
在线1: 查询改写

用户查询

在读博士申请居住证是否需要学校开具在读证明

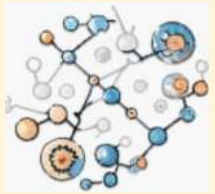
严谨规划：将问题拆解为多跳符号化的逻辑表式
逻辑形式 = F(查询 + 图模式)

1. 查找博士办理居住证的办事指南
get_spo(s=s1: 在读博士申请居住证|政务办事,p=p1: 材料证明，o=o1: 材料证明)
2. 判断材料中是否需要在读证明
verify(left_expr=o1, right_expr=在读证明，op=包含)



政府官网办事指南

在线2: 检索



向量检索

连续就读

语义检索

在读博士

文本检索

S: 在读博士申请居住证

P: 材料证明

语义推理增强检索：查询和知识库分布差异

在线3: 生成

数学：Count/sum/sort/calculate/……

自然语言推断：has_property/isa/ entailment

Generator

答案生成

执行 「执行器+生成器」提升答案可信度

08 RAG 应用实践

8-1. RAG 预定义流程 | 8-2. 动态流程 | 8-3. 隐性成本与实践指南

成本演进：从LLM到RAG再到RAG+推理

基础LLM

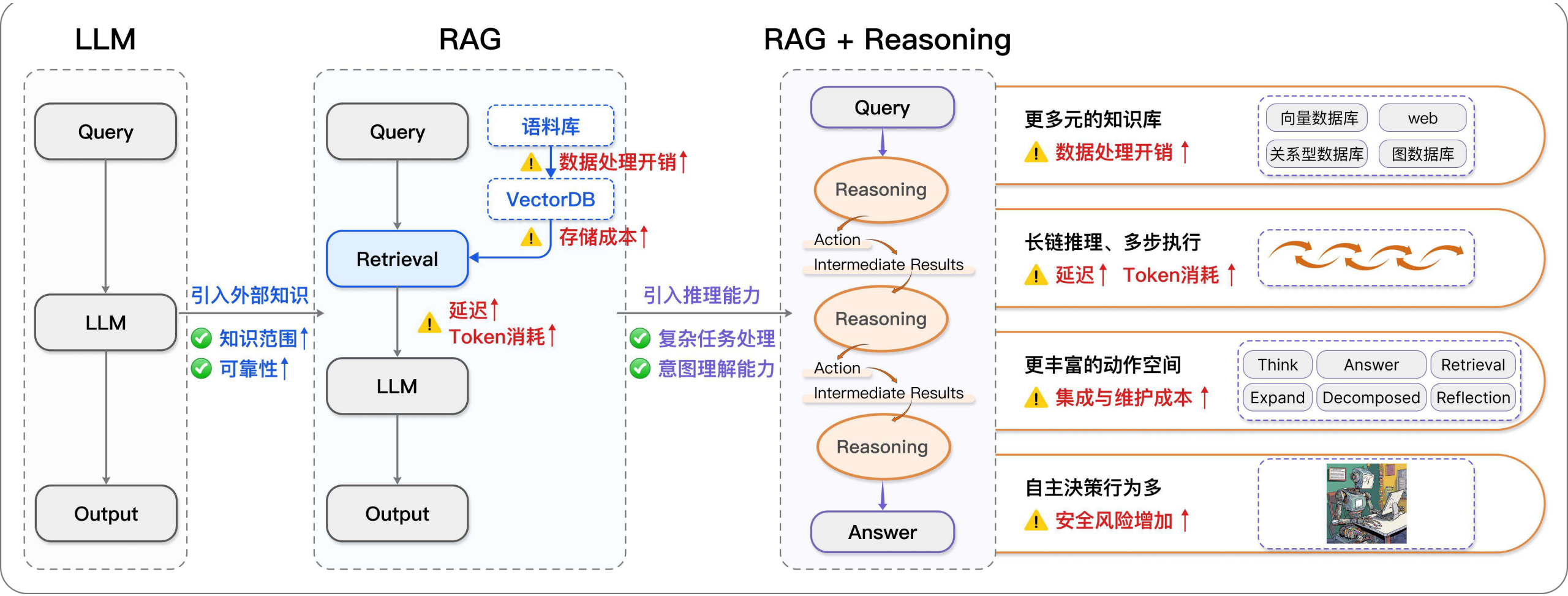
• 直接、高效 • 低延迟 • 低Token消耗 • 仅限预训练知识

RAG

• 引入外部知识检索 • 扩展响应范围 • 增加数据处理开销 • 提高延迟和Token成本

RAG+推理

• 多步骤推理能力 • 复杂任务处理 • 计算需求大幅增加 • 系统复杂性提升



RAG + 推理的成本权衡

计算资源的非线性增长

多步推理复杂性

执行多次LLM生成和检索，远超基准模型复杂度

资源扩展挑战

随模型规模、推理链长度和任务复杂度呈超线性增长

隐性Token膨胀

Token膨胀来源

- 思考链生成
- 检索文档处理
- 多轮验证反馈
- 候选路径探索

检索效率的边际衰减

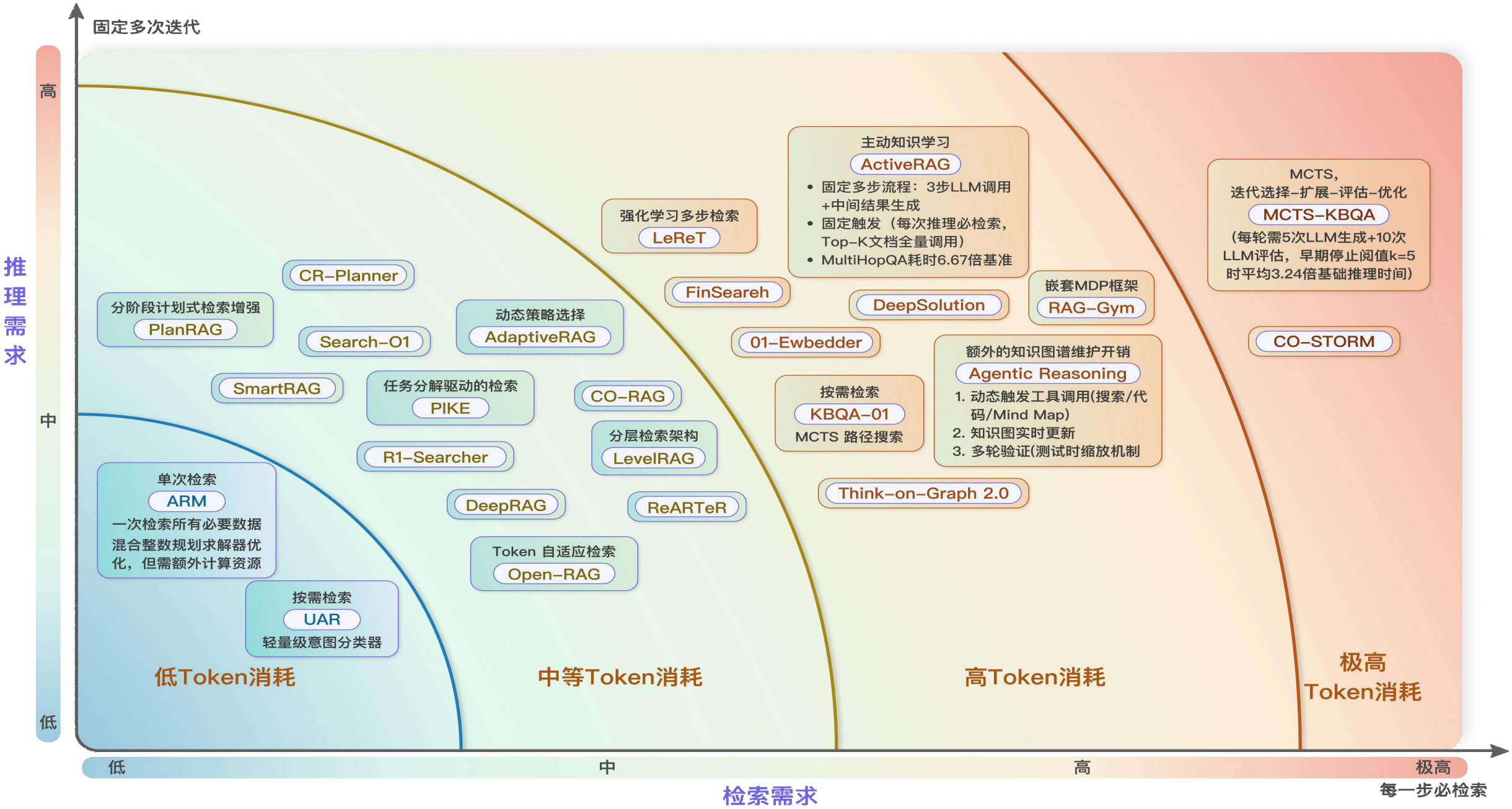
动态检索的权衡

提高知识精度，但随任务复杂度降低效率

效率天花板

高精度检索方法带来巨大计算和时间成本

RAG + 推理的成本权衡矩阵



未来还需要什么？

精细成本模型

量化计算资源与性能间的真实权衡

平衡有效性

在复杂任务中控制推理开销

目标导向

限制推理范围，提高系统效率

RAG+推理：性能提升与成本控制的精细平衡

实践指南：RAG+推理系统中的领域特征分析

领域特征维度

查询阶段特征

意图理解复杂性

语义保留与推理深度

不同行业查询抽象程度差异

检索阶段特征

知识源动态适应性

多域数据整合能力

频繁更新与碎片化知识挑战

生成阶段特征

输出质量与可解释性

幻觉控制与追溯性

不同领域的延迟要求

时间维度

知识时效性验证

信息更新周期

时间敏感信息处理

计算效率

推理深度与计算开销

资源分配策略

避免过度推理

典型领域分析

金融领域

- 投资决策分析
- 实时市场数据
- 严格的追溯性
- 高频更新知识

医疗领域

- 复杂医学语义解析
- 多学科推理
- 极低的幻觉容忍度
- 跨模态知识整合

法律领域

- 精确法律术语解析
- 案例与法规引用
- 高于95%的可追溯性
- 严格的文档标准

个人助理

- 多样化用户需求
- 实时上下文感知
- 动态知识整合
- 灵活的延迟控制

工业领域

- 设备手册解析
- 多参数关联推理
- 关键参数精确提取
- 复杂流程建模

实用指南：如何根据场景特点选择合适的方法

Do's (推荐做法)

- 结构化推理场景应用思维链(CoT)分解
- 针对动态需求使用提示工程动态适配
- 在确定性决策场景建立多级保证系统
- 使用知识图谱约束推理空间
- 动态关联碎片化知识单元
- 设计动态剪枝阈值函数

Don'ts (避免做法)

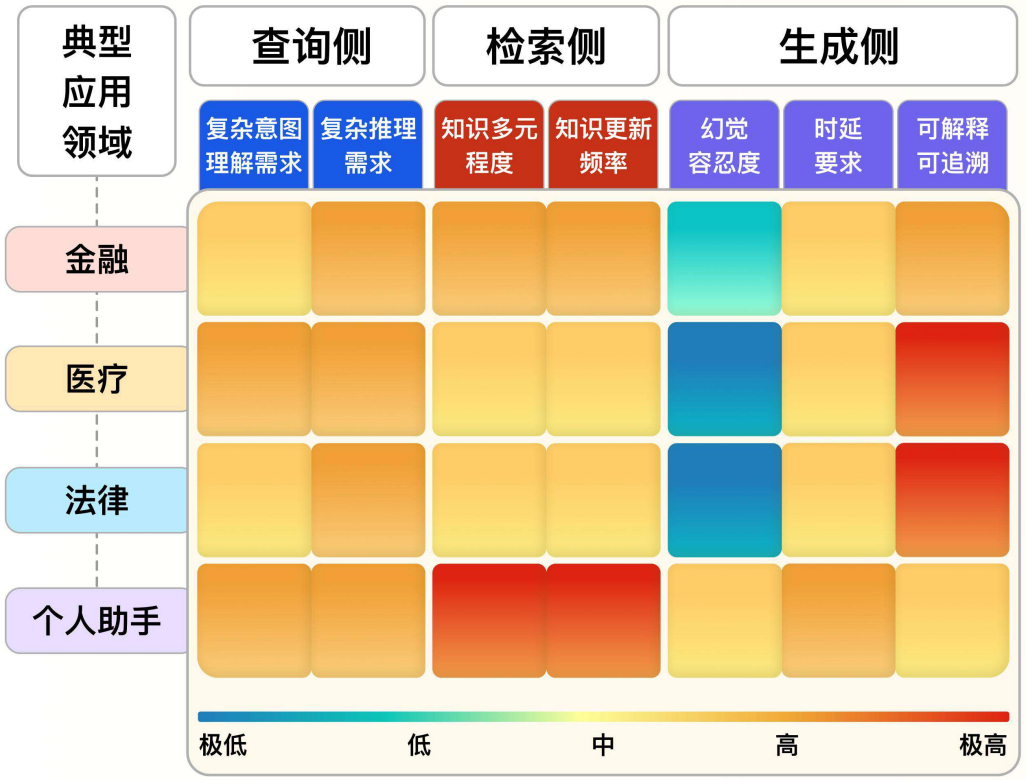
- 避免模型过度推理和计算浪费
- 不要盲目追求高复杂度推理
- 避免在检索–推理循环中放大错误
- 不要忽视不同领域的特定约束
- 避免使用穷举搜索所有潜在路径
- 不要在实时场景使用耗时的模型微调

机遇点 (Opportunities)

- 冷热分层知识索引
- 跨机构知识库构建
- 动态上下文管理
- 符号推理与知识图谱
- 多模态知识融合
- 任务规划图谱技术
- 工具使用与管理图谱

实用指南：如何根据场景特点选择合适的方法

Practical Guide on RAG + Reasoning



场景特点	✅ Do's	❌ Don'ts
结构化推理场景 • 需分解多步骤逻辑 • 依赖结构化知识库 • 结果可审计	✅ CoT任务分解+Graph推理（知识图谱引导） ✅ 校验输出审查（时效性/逻辑闭环验证） ✅ 特殊Token预测（定向触发知识源）	❌ 无检索验证的知识利用 ❌ 完全依赖LLM自由推理
动态需求响应场景 • 需求频繁变化 • 需快速适配新条件 • 用户偏好敏感	✅ Prompt Base（低成本动态调整） ✅ Search（碎片知识关联+启发式规则剪枝） ✅ 灵活动态适配机制	❌ 频繁微调更新模型 ❌ 基于强化学习的策略（RLHF/DPO）
确定性决策场景 • 需唯一结论 • 结果可靠性优先 • 过程可解释	✅ 时效性校验层 ✅ 领域敏感关联检索（触发确定规则） ✅ 知识图谱约束路径	❌ 概率式探索策略 ❌ 外部分类模型决策
高时效场景 • 响应延迟敏感 • 状态空间可控	✅ 启发式规则剪枝（高频访问优先） ✅ 定向检索扩展	❌ MCTS类算法（采样耗时长） ❌ 复杂求解器（Solver）
风险敏感场景 • 需规避系统性风险 • 防止幻觉	✅ 行动前审核机制 ✅ 知识可靠性验证层	❌ 无约束RL策略 ❌ 直接执行决策Action
复杂路径探索场景 • 多分支可能性 • 需平衡深度/广度	✅ 权重排序搜索 ✅ 知识图谱路径引导	❌ 蒙特卡洛树搜索 ❌ 无剪枝的暴力搜索 ❌ 避免特殊token触发检索

机会点

数据与索引
- 冷热分层索引
- 动态上下文管理
- 跨机构知识库共建
- 精细化分层与置信度分级

模型与方法
- 事件驱动的主动检索
- 时空感知和关联的检索
- 推理过程中的多模态融合
- 动态风险传导建模与推理

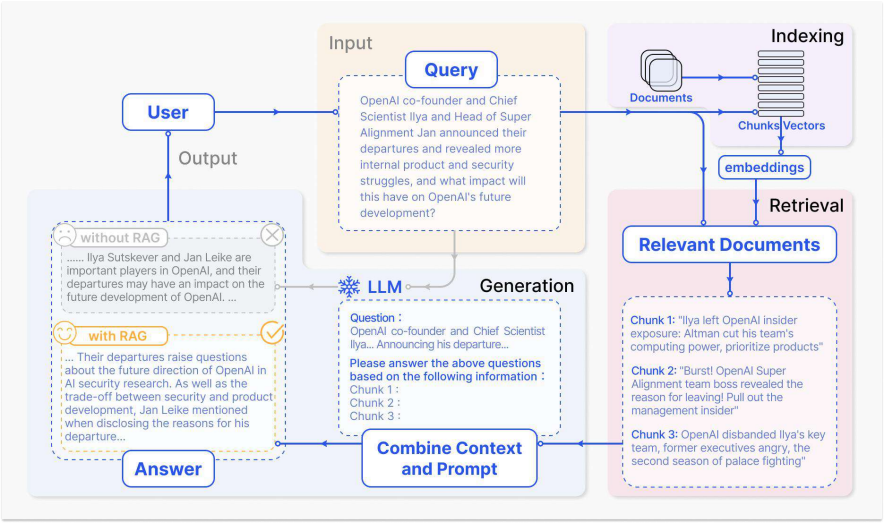
服务与应用
- 逻辑链完整性验证
- 过程中的可干预式生成
- 风险决策拦截防火墙
- 边缘–云端协同检索推理

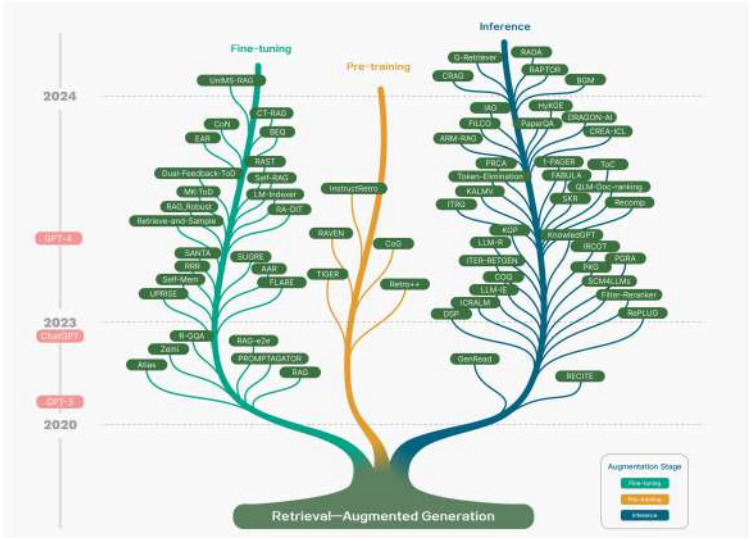
09 总结与展望

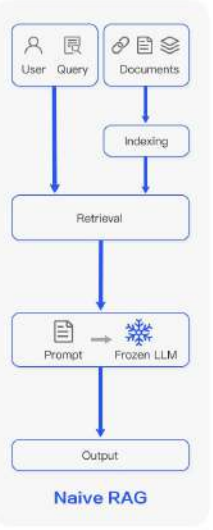
9-1. RAG发展总结 | 9-2. 未来的展望

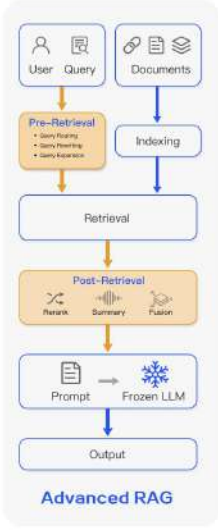
兴起：大模型时代的RAG崛起

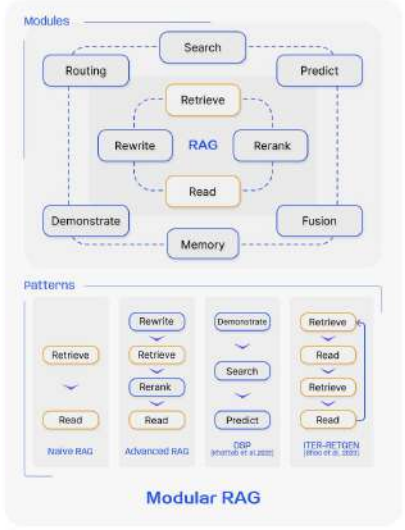
2023年12月《Retrieval-Augmented Generation for Large Language Models: A Survey》（arXiv:2312.10997）

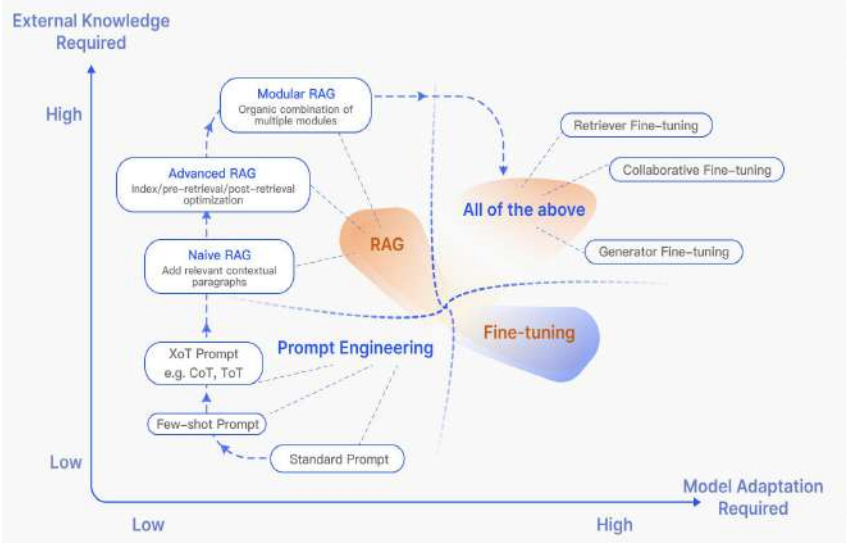


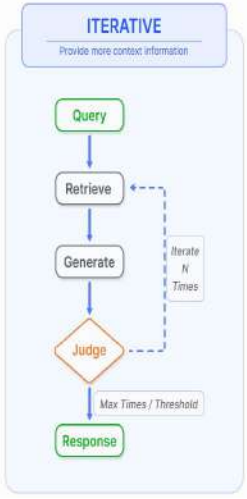


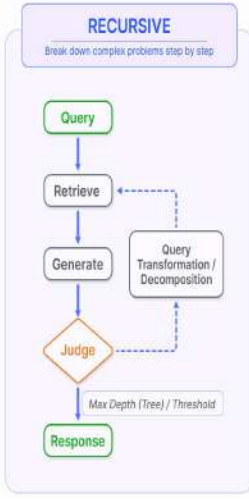


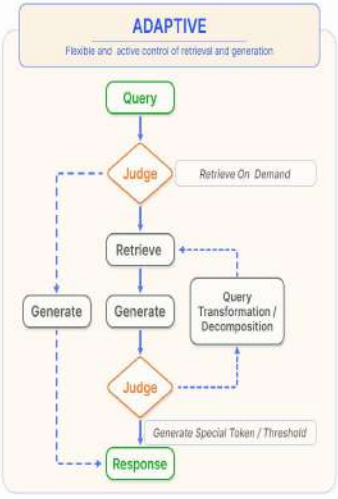


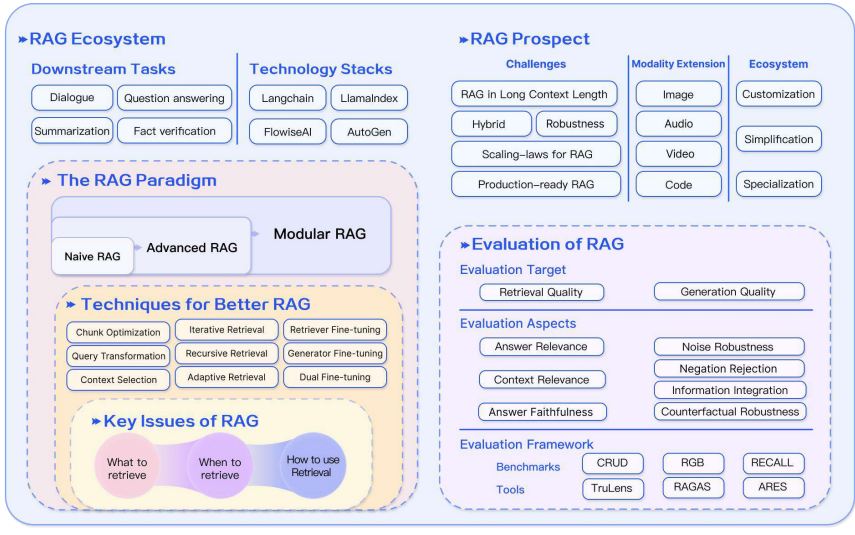






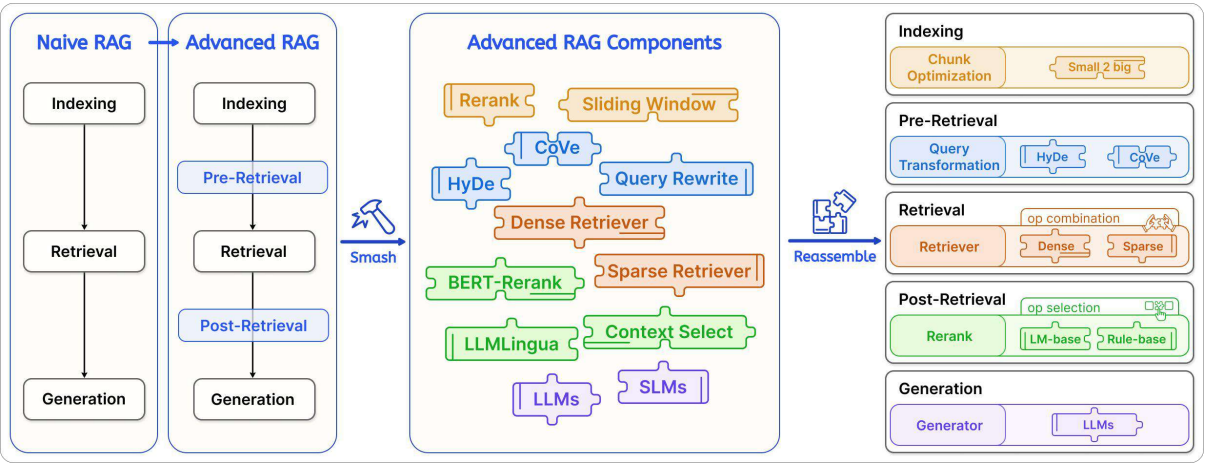


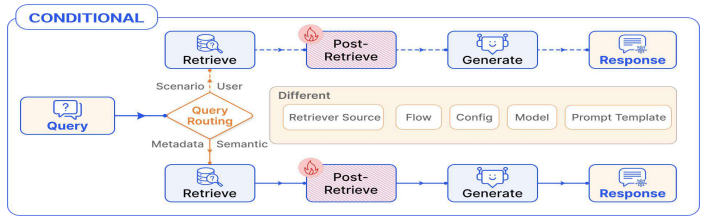


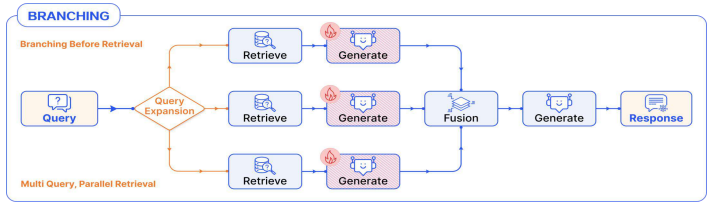


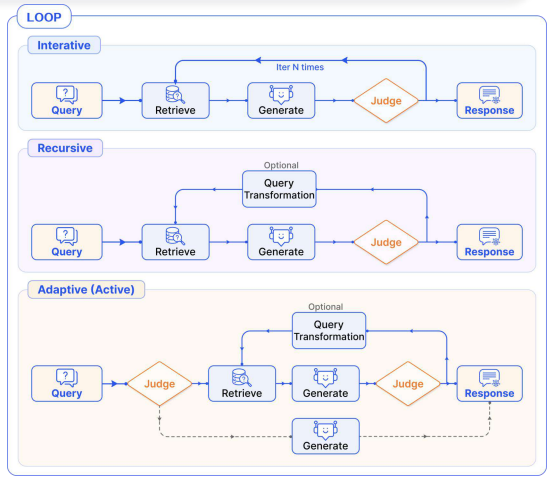
成型：模块化RAG范式

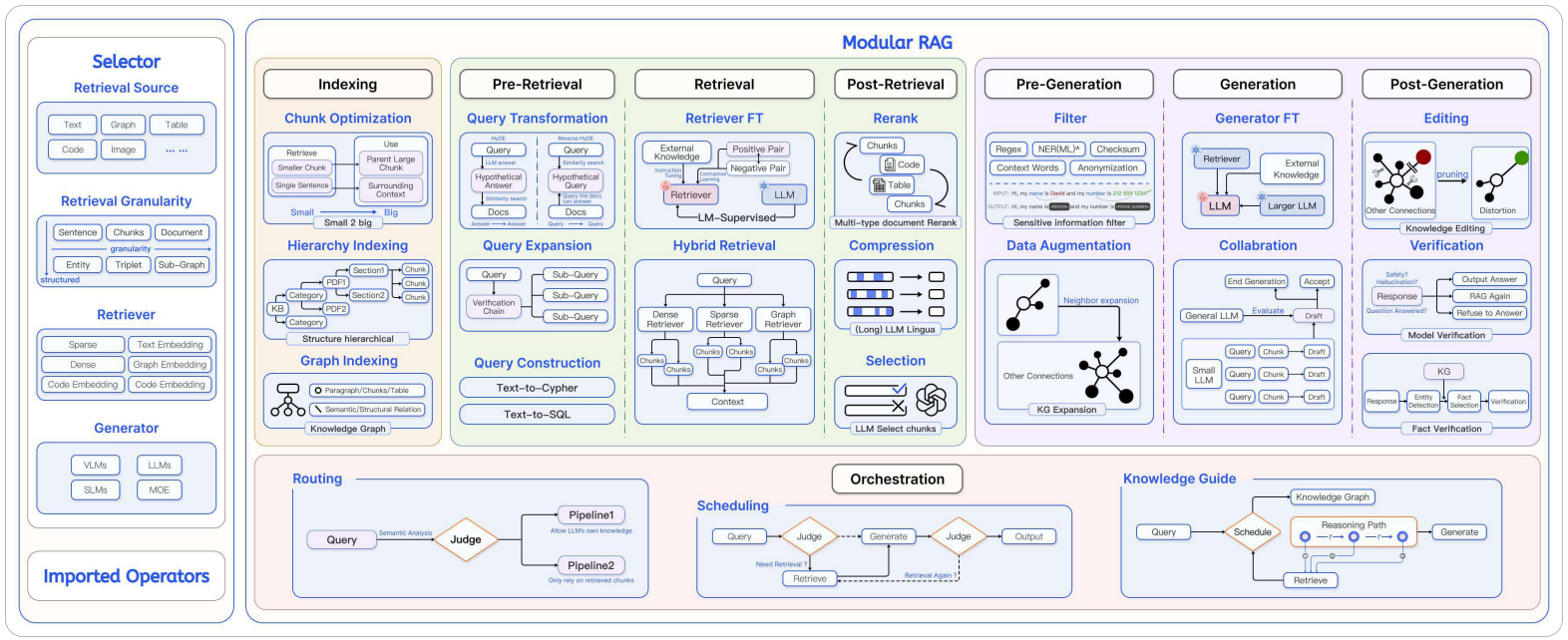
2024年7月《Modular RAG: Transforming RAG Systems into Lego-like Reconfigurable Frameworks》（arXiv:2407.21059）

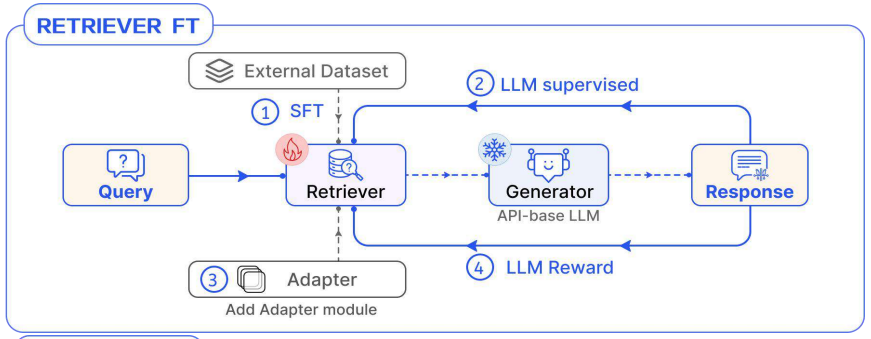


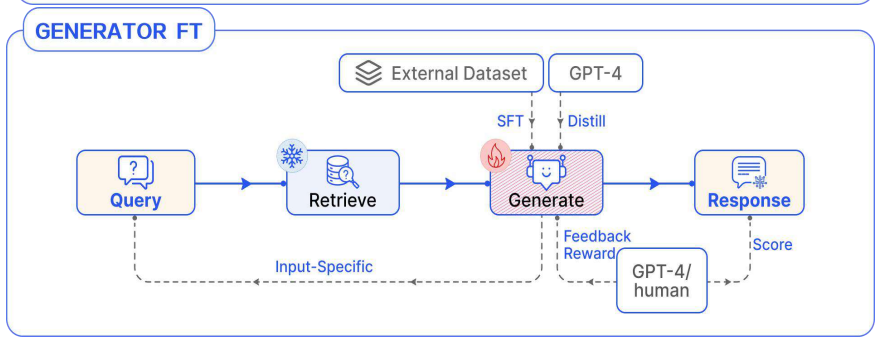






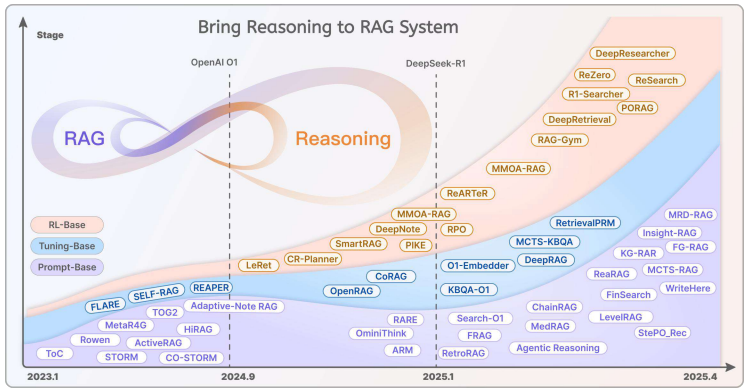


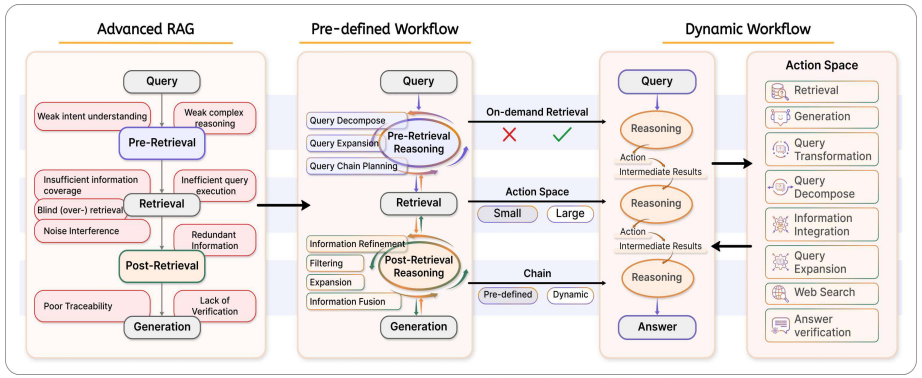


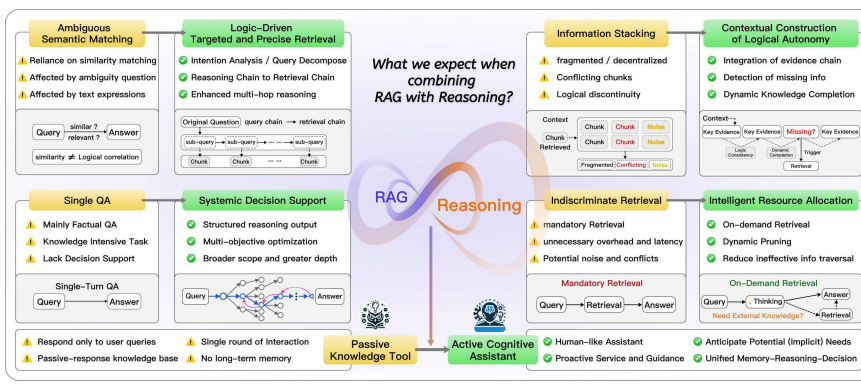


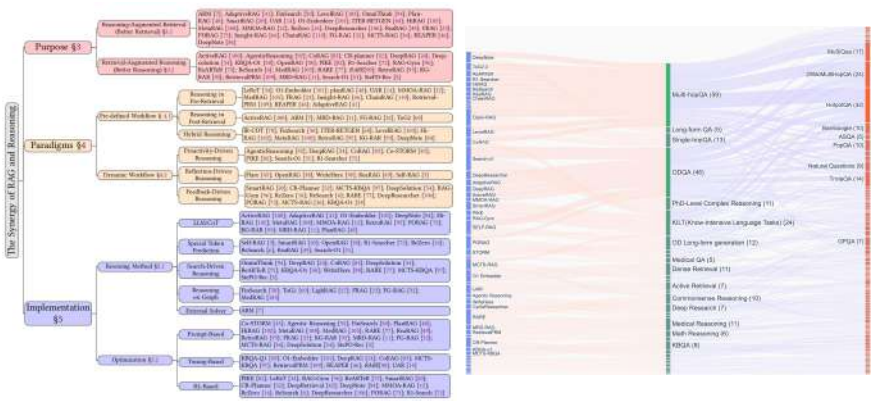
进化：RAG与推理协同的AgenticRAG

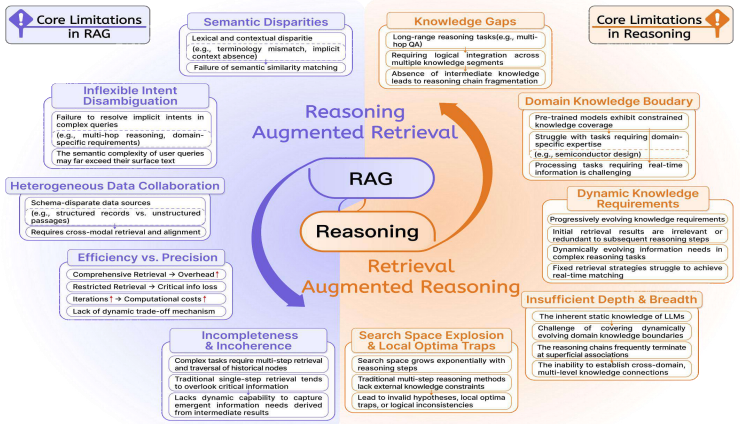
2025年4月 《Synergizing RAG and Reasoning: A Systematic Review》 （arXiv:2504.15909）

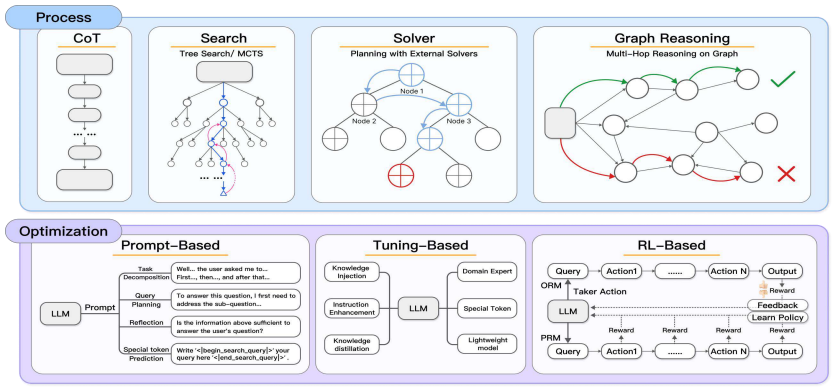


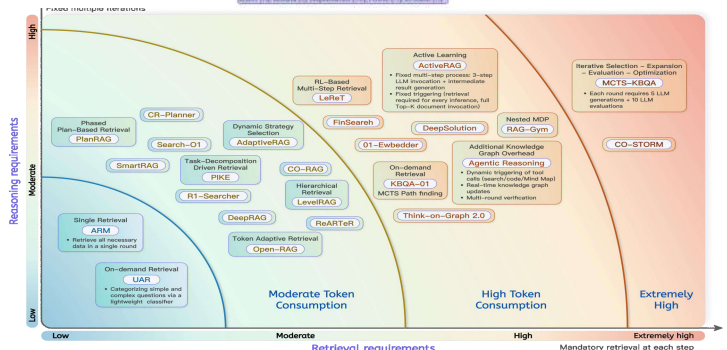


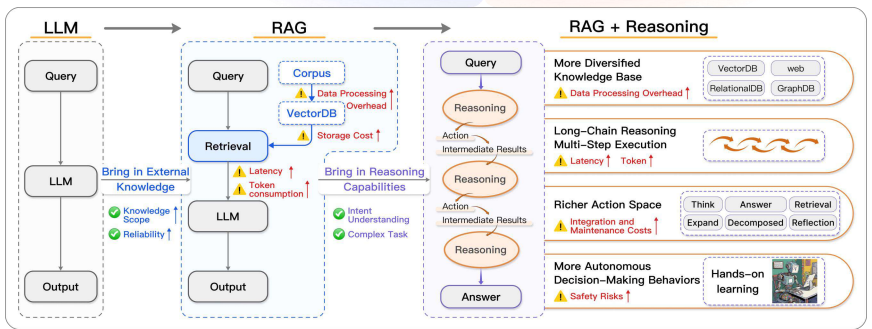


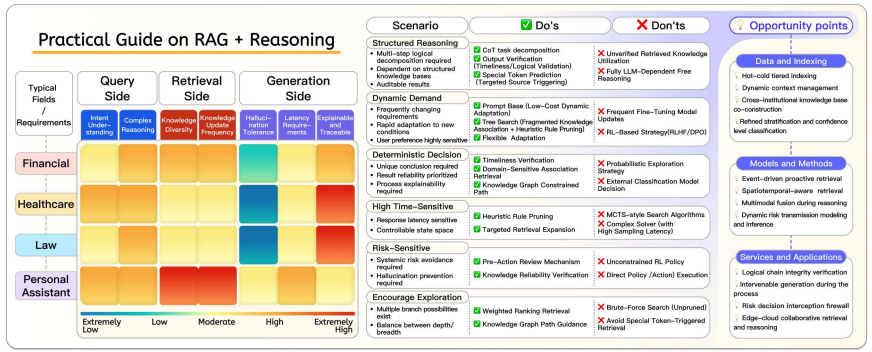




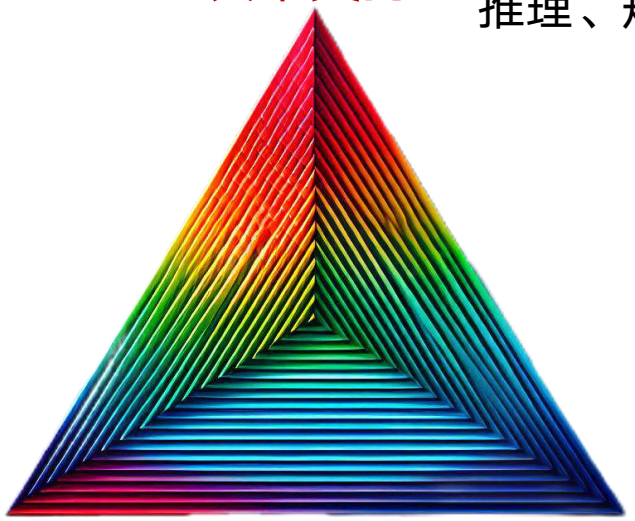








新一代RAG呼之欲出



决策支持

Reasoning:
推理、规划、溯源、校验

知识库

结构化组织（Graph化）

业务流程

服务算子编排（Bussiness）

New RAG = Reasoning+ Graph +Business ？

开放问题：

1. 趋势：三者的组合是否能够构成下一代的RAG系统？

2. 技术：下一代RAG有什么挑战？

3. 场景：率先大规模落地的杀手级应用会是？

致谢与团队介绍

感谢团队成员对本次汇报的卓越贡献

指导教授

研究员

王昊奋

同济大学·设计创意学院

研究方向为知识图谱、大模型和检索增强生成；全球最大的中文开放知识图谱联盟OpenKG轮值主席；发表100余篇AI领域高水平论文，累计引用近6500次

教授

熊贇

复旦大学·计算机科学与技术学院

研究方向为数据科学、数据挖掘和大数据处理；累计主持多项国家级科研项目，出版专著4本，获中国计算机学会科学技术发明奖等奖项。

副教授

王萌

同济大学·设计创意学院

研究方向为多模态大模型、智能座舱、交互设计；主持国家自然科学基金2项，是CCF-腾讯犀牛鸟基金、CCF-百度松果基金的获得者，在SCI期刊和顶级会议发表论文50余篇，参与出版专著2本。

副教授

李博涵

南京航空航天大学·计算机科学与技术学院

研究方向为数据库、大模型、推荐系统；CCF杰出会员；NDBC2017、2020，MLICOMM2019，ADMA2019，NDBC2020优秀学生论文，WSDM2023（提名） 会议最佳论文获得者

研究团队

博士

高云帆

同济大学·上海自主智能无人系统科学中心

研究方向：检索增强生成（RAG）；负责内容：汇报内容的整体制作

硕士

毕羽西

同济大学·设计创意学院

研究方向：人工智能驱动的时尚创新；负责内容：负责研究内容可视化设计

博士

黄欣怡

复旦大学·计算机科学与技术学院

研究方向：数据科学及大数据应用；负责内容：RAG框架与工具新趋势部分。

博士

王宏润

同济大学·设计创意学院

研究方向：多媒体与人机交互；负责内容：应用实践、个人知识助手和领域应用。

ขอบคุณ 谢谢

感谢 Thanks

Terima kasih ありがとう

Github

OpenRAG

OpenKG