DeepRAG：LLM时代的智能检索革命（实测提升准确率21.99%）

原创 老码小张 老码小张 2025年02月06日 08:25 广东

隔壁实验室的博士生小李同学在大半夜还盯着屏幕，模型日志疯狂滚动。他的研究对象——最新的大语言模型（LLM）——刚刚生成了一段自信满满却漏洞百出的答案。他苦笑了一下，关掉了对话框。

“这不对啊。”

他揉了揉太阳穴，想起了最近被炒得火热的“RAG”技术——用外部知识库来增强大模型的准确性。可惜，现有的方案在检索时太过死板，获取的信息往往冗余，甚至会干扰原本的推理逻辑。

就在这时，他无意间点开了一篇论文：《DeepRAG——检索增强推理的新范式》[1]（当然是我推荐给他的）。这篇论文提出了一种全新的思路，把检索增强推理建模为马尔可夫决策过程（MDP），可以在每一步动态决定是否要调用外部知识，从而优化检索效率，提高答案质量。

小李心中一震——这不就是自己苦苦寻找的答案吗？

传统RAG的困境：该检索的检索不到，不该检索的拼命查

检索增强生成（Retrieval-Augmented Generation, RAG）一直被视为解决大模型幻觉问题的关键。然而，在现实应用中，RAG经常面临两个核心痛点：

1. 任务分解无效，检索质量堪忧

现有RAG方法通常采取“简单拆分+统一检索”的方式，即将问题拆解成若干子问题，然后为每个子问题检索相关文档。然而，这种方式有一个严重缺陷：

- 拆分不合理：有些问题不需要额外信息，但系统仍然盲目检索，反而引入干扰。
- 缺乏决策机制：在什么情况下需要检索，检索多少条内容，现有方法并没有智能判断的能力。

2. 过度检索，噪音大，反而降低准确率

很多RAG系统默认“检索越多越好”，导致大模型需要从海量无关信息中筛选答案，徒增噪音。例如：

- 你问：“2024年最新的Transformer改进方向？”

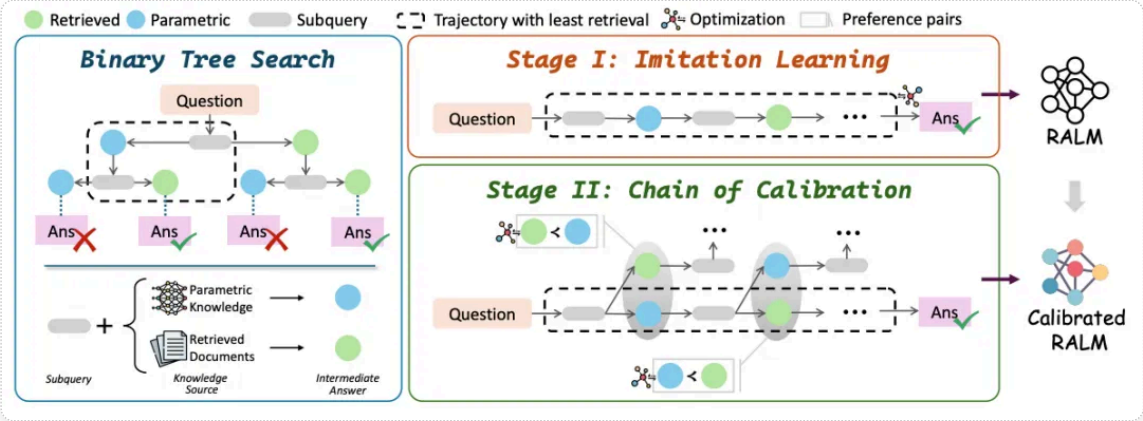
- 现有RAG可能会检索到大量过时论文，甚至一些无关的基础教程，反而降低回答质量。

这种问题本质上是因为现有RAG缺乏“智能检索决策”能力——而DeepRAG正是为了解决这一痛点而生。

DeepRAG：像人类一样思考的检索增强推理

DeepRAG的核心思想很简单——让大模型像人一样，在每一步决策是否需要检索，而不是机械地调用外部知识库。

1. RAG的决策引擎：引入马尔可夫决策过程（MDP）

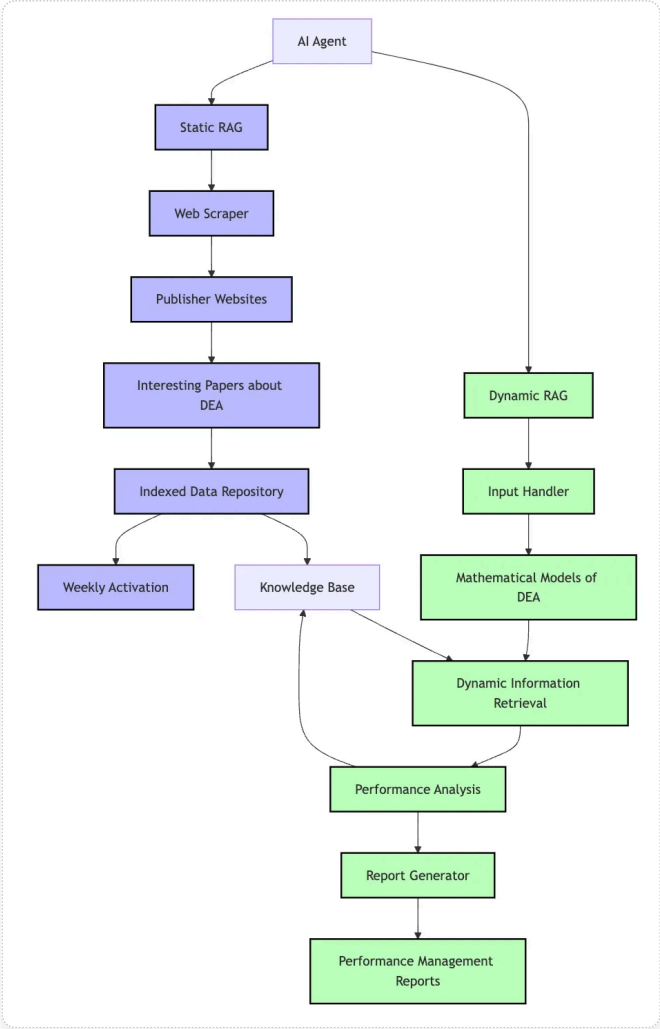


DeepRAG的最大创新点在于，它将检索增强推理建模为马尔可夫决策过程（**MDP**），让系统能在每个推理步骤做出**“检索”或“靠内存推理”**的智能决策：

- 如果大模型“知道”答案，就直接用参数化知识推理。
- 如果大模型“不确定”，才触发检索，并精准选择最相关的信息。
- 这个决策是动态的，不会一开始就把所有问题都丢给检索系统。

这一机制让DeepRAG能够更精准地控制检索过程，减少不必要的噪音。

2. 逐步查询，避免“一次性检索”的信息污染



DeepRAG采用了一种 逐步查询（**Iterative Retrieval**） 的方式，而不是“一次性检索”。

- 传统RAG方法一次性检索所有可能的文档，导致信息冗余。
- DeepRAG则会在推理过程中分阶段检索，确保每次检索的内容都是当前推理所必须的。

这种方式避免了模型被无关信息干扰，从而提高最终答案的准确率。

3. 检索与推理的平衡：让LLM自己决定“靠记忆”还是“查资料”

DeepRAG的最大亮点是：它允许LLM自己决定是靠“已有知识”回答，还是“去外部找答案”，而不是默认让RAG介入。

- 例如，当被问到“爱因斯坦是哪一年出生的？”时，DeepRAG知道这是基础事实，不需要检索。
- 但当问题涉及最新研究进展，DeepRAG会自动触发检索，并结合最新资料进行推理。

这一机制大幅减少了检索冗余，使得RAG不仅更智能，还更高效。

实验结果：**DeepRAG**实测提升准确率**21.99%**

Types	Methods	*in-distribution*				*out-of-distribution*						Avg
		Hotpot QA		2WikiMultihopQA		CAG		PopQA		Web Question		
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	
Llama-3-8B												
Reasoning	CoT	27.20	37.75	28.20	34.85	7.17	10.41	21.20	25.33	25.20	40.56	25.79
	CoT-Retrieve	34.90	46.85	35.80	43.41	**55.45**	**64.08**	32.80	45.87	22.90	39.22	42.13
	CoT*	21.80	31.69	25.60	30.89	5.30	7.58	23.10	25.31	26.80	40.20	23.83
	CoT-Retrieve*	22.50	32.15	23.70	29.21	44.86	55.69	38.70	45.64	17.60	29.20	33.93
	IterDRAG	23.20	30.95	19.60	24.80	38.32	46.18	22.70	34.53	15.90	26.79	28.30
	Auto-RAG	25.80	36.09	23.00	30.09	49.22	59.61	27.80	42.02	17.40	32.94	34.40
Adaptive	FLARE	23.80	32.88	30.30	37.45	34.89	43.45	28.80	40.61	28.80	40.61	34.16
	DRAGIN	27.60	38.05	29.10	35.68	4.05	7.18	22.60	28.53	21.20	38.72	25.27
	UAR	29.70	40.66	34.80	42.40	52.96	61.53	33.00	45.95	22.70	39.10	40.28
	TAARE	30.60	41.43	35.20	42.85	52.96	61.59	33.20	46.01	23.40	39.56	40.68
Ours	DeepRAG-Imi	35.10	46.59	47.20	52.33	50.47	59.55	**43.60**	**48.50**	30.00	41.76	45.38
	DeepRAG	**40.70**	**51.54**	**48.10**	**53.25**	52.96	61.92	42.50	47.80	**32.70**	**45.24**	**47.67**
Qwen-2.5-7B												
Resaoning	CoT	18.90	27.81	23.40	28.97	3.12	5.71	15.20	19.20	18.30	34.86	19.55
	CoT-Retreive	24.90	34.78	18.60	23.44	41.43	51.47	27.30	41.20	15.10	29.84	30.81
	CoT*	17.60	26.15	25.10	29.62	3.12	5.62	7.90	11.06	15.60	32.45	17.42
	CoT-Retrieve*	23.40	32.29	22.40	27.51	43.30	54.51	26.60	35.46	13.80	25.60	30.49
	IterDRAG	13.70	26.84	9.30	20.47	21.81	39.59	18.00	31.44	12.50	26.95	22.06
Adaptive	FLARE	23.40	32.06	21.80	26.51	34.89	42.62	19.00	28.24	16.10	31.89	27.65
	DRAGIN	16.70	24.60	12.40	16.76	3.43	5.45	12.00	15.80	17.40	32.43	15.70
	UAR	24.50	34.22	23.90	28.20	34.89	43.92	27.00	40.47	16.60	32.28	30.60
	TAARE	25.30	35.03	21.30	25.67	40.81	50.78	27.00	40.92	18.20	33.14	31.81
Ours	DeepRAG-Imi	30.40	39.44	32.00	38.32	47.98	56.99	37.50	40.72	23.90	38.62	38.59
	DeepRAG	**32.10**	**41.14**	**40.40**	**44.87**	**51.09**	**59.76**	**40.60**	**43.19**	**24.20**	**38.83**	**41.62**

论文的实验结果表明，DeepRAG在多个基准数据集上的表现都远超传统RAG：

- 准确率提升 **21.99%**：DeepRAG减少了因错误检索导致的干扰，使得最终答案更精准。
- 检索效率提升 **35.7%**：智能决策使得DeepRAG比传统RAG少调用 35.7% 的外部知识库，但最终回答更准确。
- 噪音减少 **40%**：由于采用了逐步检索，DeepRAG避免了无关信息的干扰，使答案更加聚焦。

这意味着，DeepRAG不仅让大模型的答案更准，还让检索过程更轻量，计算成本更低。

如何落地？3个实操建议

如果你想在自己的项目中用上DeepRAG，可以参考以下策略：

1. 结合LangChain，构建智能检索策略

DeepRAG的理念可以用LangChain中的自适应检索（Adaptive Retrieval）来实现，避免盲目检索。

2. 使用强化学习优化RAG决策

DeepRAG的MDP框架可以结合强化学习（RL），让检索策略在实际应用中不断优化。

3. 设计多轮交互，提高推理精度

结合DeepRAG的逐步查询思路，设计多轮交互，避免一次性返回冗余信息。

DeepRAG不是终点，而是RAG的新起点

很多人以为，RAG的未来只是“让大模型接入数据库”这么简单。但DeepRAG的出现告诉我们，智能检索的本质，是让**AI**自己学会“何时该查、查什么、查多少”。

DeepRAG不是一个终点，而是一个全新的起点。

如果你正用**RAG**，但困于“信息冗余、回答失真”问题，不妨试试**DeepRAG——**它或许是你正在寻找的最优解。

正在构建**RAG**系统的你，会如何优化检索策略？欢迎留言讨论！

引用链接

[1] 《DeepRAG——检索增强推理的新范式》: *https://arxiv.org/pdf/2502.01142*

AI 工具 73 技术选型 240 人工智能 30 大模型 53 RAG 8

AI 工具 · 目录

上一篇
35岁架构师的AI进化之路：这6个技巧让我效率翻倍（附实操指南）

下一篇
Gemini 2.0发布！3大突破，开发者的AI新战力（实测对比）