

报告主题：对 Search-o1 技术的学习与应用思考

首先需要说明，本报告涉及的主要技术来自 *Search-o1: Agentic Search-Enhanced Large Reasoning Models*[1]这篇文章，其内容主要为介绍 Search-o1 的运作方式及效果。另外还包括了来自 [GitHub - sunlynexus/Search-o1: Search-o1: 代理搜索增强的大型推理模型](https://github.com/sunlynexus/Search-o1)中该论文的作者开源的代码。另外也会包含一部分我在学习过程中因为必要的检索而又衍生出的其他信息。

一、背景介绍

在正式介绍 search-o1 之前，需要先介绍 LRM（Large reasoning models）。在 24 年 9 月份，openAI 发布了 openAI-o1[2]模型，与 GPT 这类以 NLP 为主的产品不同，openAI-o1 注重的是逻辑推理。换句话说，同样的需要逻辑思考才能解决的问题，openAI-o1 的能力是优于 GPT-4o 的（甚至优于人类）。即模型有了正确地解决一部分数学、物理学、生物学、化学、计算机科学等学科的问题的能力。与传统的语言模型不同的是，openAI-o1 在回答问题之前会形成一个思路链，这个思路链是一个逐步推导、逐步分解问题的过程，它模拟了人类思考的方式，使得模型能够更深入地理解问题并给出更准确的答案。这是 LRM 比较重要的成果，也是让这个词汇进入更多人事业的一个契机。

除了 LRM 的特性，还有一项对于大模型应用非常重要的技术，RAG（Retrieval Augmented Generation, RAG），用中文描述可以叫：检索增强生成。它通过从数据源中检索信息来辅助大语言模型（Large Language Model, LLM）生成答案。简而言之，RAG 结合了搜索技术和大语言模型的提示词功能，即向模型提出问题，并以搜索算法找到的信息作为背景上下文，这些查询和检索到的上下文信息都会被整合进发送给大语言模型的提示中。典型的基于 RAG 架构的 LLM 产品有天工 ai（昆仑万维自研的双千亿级大语言模型）、kimi-ai（北京月之暗面科技有限公司推出的大语言模型）。在大部分使用场景下，RAG 获取的参考的提示信息都可以有效的提高模型问题回答正确的概率。同样的，最初被用于 LLM 的 RAG 技术也被应用于了 LRM。大部分 RAG 的解决方案是使用另一个模型对输入内容进行处理，筛选出这个问题可能需要的参考信息，有一部分解决方案是使用知识库来保存这些参

考信息，在模型筛选出需要的参考信息的关键字后由知识库为 LLM 提供提示词和上下文内容，另一部分解决方案是根据提问内容生成一些检索词条通过搜索引擎获取文本在经过筛选后提供给 LLM。

还有一项新的技术，也是在近几日（2025/1/8）发布的，叫做 rStar-Math[3]，简要来说其主要通过代码增强 CoT、蒙特卡洛树搜索等，可以帮助小参数模型在不依赖老师模型蒸馏的情况下，实现多轮自我思维深度进化，极大增强模型的数学推理能力。与 LRM 直接更改推理方式不同，rStar-Math 相当于是给 LLM 打了补丁，成本上来看更低，且对于 LLM 模型的上下文能力和本身文本的处理能力保留的更完备，但能够解决的问题有限。篇幅和学习难度原因此处不多赘述。

二、search-o1 主要特征简介

1. 参考信息获取的动态性

上文提到，RAG 的参考信息筛选和检索一般是放在模型推理之前，这是一种“静态的”处理方式，相对来说不够灵活，且在推理过程中极有可能会产生其他信息，需要新的参考信息，这会使模型推理的结果具有较大的不确定性。而 search-o1 则采用了一种动态的方式进行参考信息的获取，即采用了另一种 DAG 的方案，使模型可以在推理过程中获取额外的参考信息。这使得 LRM 能够在推理过程中主动检索外部知识来弥补知识缺失，并将检索到的知识无缝地整合到推理链中，保持推理过程的连贯性。

2. 对于检索结果的提炼

Search-o1 引入了 Reason-in-Documents 模块，该模块能够对检索到的文档进行深入分析，提炼出与当前推理步骤最相关且最简洁的信息，从而避免直接插入冗长的文档内容而打断原有的推理逻辑。

3. 推理过程的并行性

Search-o1 采用批量推理机制，能够同时高效地处理多个问题，包括并行生成推理步骤和并行提炼知识。原文中这一部分的篇幅较少，经过检索我又搜索到了 vLLM[4]这项批量推理

技术，其中主要包括分区（Partitioning）、虚拟内存（Virtual Memory）、量化（Quantization）、蒸馏（Distillation）、混合精度（Mixed Precision）、并行计算（Parallel Computing）这些概念，解决的是模型并行调度的问题。总之可以确定，这方面的特性并非 search-o1 原创，是业界早已有的方案。

4. 推理结果的正确性

根据原文提供的比较结果，在复杂的科学、数学和编程推理任务以及多种开放域问答基准测试中,Search-o1 显著优于直接推理的原生 openai-o1 和标准 RAG 方法,甚至超越了人类专家水平,展现出更可靠和通用的推理能力。下图为原文给出的开放域问题回答的结果表格：

Method	GPQA (PhD-Level Science QA)				Math Benchmarks			LiveCodeBench			
	Physics	Chemistry	Biology	Overall	MATH500	AMC23	AIME24	Easy	Medium	Hard	Overall
<i>Direct Reasoning (w/o Retrieval)</i>											
Qwen2.5-32B	57.0	33.3	52.6	45.5	75.8	57.5	23.3	42.3	18.9	14.3	22.3
Qwen2.5-Coder-32B	37.2	25.8	57.9	33.8	71.2	67.5	20.0	61.5	16.2	12.2	25.0
QwQ-32B	75.6	39.8	68.4	58.1	83.2	82.5	53.3	61.5	29.7	20.4	33.0
Qwen2.5-72B	57.0	37.6	68.4	49.0	79.4	67.5	20.0	53.8	29.7	24.5	33.0
Llama3.3-70B	54.7	31.2	52.6	43.4	70.8	47.5	36.7	57.7	32.4	24.5	34.8
DeepSeek-R1-Lite†	-	-	-	58.5	91.6	-	52.5	-	-	-	51.6
GPT-4o†	59.5	40.2	61.6	50.6	60.3	-	9.3	-	-	-	33.4
o1-preview†	89.4	59.9	65.9	73.3	85.5	-	44.6	-	-	-	53.6
<i>Retrieval-augmented Reasoning</i>											
RAG-Qwen2.5-32B	57.0	37.6	52.6	47.5	82.6	72.5	30.0	61.5	24.3	8.2	25.9
RAG-QwQ-32B	76.7	38.7	73.7	58.6	84.8	82.5	50.0	57.7	16.2	12.2	24.1
RAgent-Qwen2.5-32B	58.1	33.3	63.2	47.0	74.8	65.0	20.0	57.7	24.3	6.1	24.1
RAgent-QwQ-32B	76.7	46.2	68.4	61.6	85.0	85.0	56.7	65.4	18.9	12.2	26.8
<i>Retrieval-augmented Reasoning with Reason-in-Documents</i>											
Search-o1 (Ours)	77.9	47.3	78.9	63.6	86.4	85.0	56.7	57.7	32.4	20.4	33.0

三、search-o1 的工作流及部署

Search-o1 的工作方式遵循以下步骤：

接受问题->LRM 开始思考->对于 LRM 正在推理的步骤，寻找有效信息->通过 reason-in-documents 获取简洁准确的提示信息并继续推理步骤->重复这两步->得到结果

（以上内容参照 github 上提供的信息总结）

下图为该流程的伪代码。

Algorithm 1 Search-o1 Inference

Require: Reasoning Model \mathcal{M} , Search function Search

```
1: Input: Questions  $\mathcal{Q}$ , Task instruction  $I$ , Reason-in-documents instruction  $I_{\text{docs}}$ 
2: Initialize set of unfinished sequences  $\mathcal{S} \leftarrow \{I \oplus q \mid q \in \mathcal{Q}\}$ 
3: Initialize set of finished sequences  $\mathcal{F} \leftarrow \{\}$ 
4: while  $\mathcal{S} \neq \emptyset$  do
5:   Generate all sequences in  $\mathcal{S}$  until EOS or <end_search_query>:  $\mathcal{T} \leftarrow \mathcal{M}(\mathcal{S})$   $\triangleright$  Batch Generate
6:   Initialize empty set  $\mathcal{S}_r \leftarrow \{\}$   $\triangleright$  Reason-in-documents Inputs
7:   for each sequence  $\text{Seq} \in \mathcal{T}$  do
8:     if  $\text{Seq}$  ends with <end_search_query> then
9:       Extract search query:  $q_{\text{search}} \leftarrow \text{Extract}(\text{Seq}, \text{code}<begin\_search\_query>, <end\_search\_query>)
10:      Retrieve documents:  $\mathcal{D} \leftarrow \text{Search}(q_{\text{search}})$   $\triangleright$  Retrieval
11:      Construct input for Reason-in-documents:  $I_{\mathcal{D}} \leftarrow I_{\text{docs}} \oplus q_{\text{search}} \oplus \text{Seq}$ 
12:      Append the tuple  $(I_{\mathcal{D}}, \text{Seq})$  to  $\mathcal{S}_r$ 
13:     else if  $\text{Seq}$  ends with EOS then
14:       Remove  $\text{Seq}$  from  $\mathcal{S}$ , add  $\text{Seq}$  to  $\mathcal{F}$   $\triangleright$  Sequence Finished
15:   if  $\mathcal{S}_r \neq \emptyset$  then
16:     Prepare batch inputs:  $\mathcal{I}_r \leftarrow \{I_{\mathcal{D}} \mid (I_{\mathcal{D}}, \text{Seq}) \in \mathcal{S}_r\}$ 
17:     Reason-in-documents:  $\mathcal{T}_r \leftarrow \mathcal{M}(\mathcal{I}_r)$   $\triangleright$  Batch Generate
18:     for  $i \leftarrow \{1, \dots, |\mathcal{T}_r|\}$  do
19:       Let  $r \leftarrow \mathcal{T}_r[i]$ ,  $\text{Seq} \leftarrow \mathcal{S}_r[i].\text{Seq}$ 
20:       Extract knowledge-injected reasoning step:  $r_{\text{final}} \leftarrow \text{Extract}(r)$ 
21:       Update sequence in  $\mathcal{S}$ :  $\text{Seq} \leftarrow \text{Insert}(\text{code}<begin\_search\_result>,  $r_{\text{final}}$ , <end\_search\_result>)
22: Output: Finished Sequences  $\mathcal{F}$$$ 
```

该框架的部署也是非常简单的，只需要将 github 上的文件克隆到本地，在根据其中提供的 requirements.txt 配置 conda 环境即可。可惜的是我家里的网络环境并不能支持我获取到 openai-o1 的服务，故不在本地进行测试了。

四、Search-o1 的应用前景

Search-o1 实质上是一个推理准确性更高的 LRM，因此它的应用场景也会不超出原有的 LRM 涵盖的业务范围。即主要包阔以下几个范围：

科学研究:在化学、物理和生物学等科学领域，动态检索和整合知识，解决复杂的科学问题，如化学反应分析、物理问题求解和生物学问题解答。

数学教育:在数学问题解决和竞赛辅导中，检索数学公式、定理和解题技巧，帮助学生逐步推导出复杂数学问题的解决方案，提高解题能力。

编程开发:在编程任务中，检索编程语言语法、库函数和代码优化技巧，生成正确的代码并优化现有代码的性能，提高开发效率

开放域问答:在单跳和多跳问答任务中，检索相关事实和信息，进行多步推理，准确回答各种复杂问题，提供全面的知识支持。

医疗健康:在疾病诊断和治疗方案推荐中，检索症状、疾病信息、诊断方法和最新治疗

指南，辅助医生进行准确诊断和提供最佳治疗建议，提升医疗决策的科学性。

但是这种动态 RAG 方案对于推理结果准确性的提升是有价值的，可以预见，在未来 LRM 或许真的可以投入到现实问题的解决上，为未来的科学研究和工业生产提升效率（或许还包括让学生们写理科作业的速度提高）。

顺便一提的是，在 search-o1 的讨论区有测试过的用户提到模型推理速度较慢，对于响应速度有要求的话是没法用的。

五、学习收获

最初选择这篇论文中的技术作为本篇报告的主题是来源于巧合。nvidia 在前几天发布了 dlss4 技术的演示视频，让我很感兴趣，所以原本我是想找到 dlss4 技术的介绍的文档然后以这项技术作为报告内容的。但可惜我找遍了我能找的地方也没找到除了演示视频以外的信息。但也是在我找论文的平台，我看到了 search-o1 这篇论文。论文标题和摘要表示这个框架可以赋予模型自主动态进行信息检索的能力，因为我本身用过不少静态 RAG 方案的产品，所以产生了兴趣，于是就有了这篇报告。

在研读这篇论文过程中，我因为需要搜索一些名词的解释，还有一些别的我不知道的技术的解释，所以我链式地学习了很多其他的东西。而在这其中，search-o1 和 rStar-Math 这两项技术甚至是相差很短的时间发布的，而且也都是我在动笔写这篇报告的前几天，这些巧合给我的学习过程带来了不少惊喜。也算是第一次追了人工智能领域的前沿成果，收获颇丰啊。

参考文献

- [1] Search-o1: Agentic Search-Enhanced Large Reasoning Models,Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, Zhicheng Dou,arXiv:2501.05366
<https://doi.org/10.48550/arXiv.2501.05366>
- [2] OpenAI o1 System Card,OpenAI: Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya et al. (162 additional authors not shown),arXiv:2412.16720,<https://doi.org/10.48550/arXiv.2412.16720>
- [3] rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking,Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, Mao Yang,arXiv:2501.04519,<https://doi.org/10.48550/arXiv.2501.04519>
- [4] Visual Large Language Models for Generalized and Specialized Applications,Yifan Li, Zhixin Lai, Wentao Bao, Zhen Tan, Anh Dao, Kewei Sui, Jiayi Shen, Dong Liu, Huan Liu, Yu Kong,arXiv:2501.02765,<https://doi.org/10.48550/arXiv.2501.02765>