

基于 SELF-RAG 算法的反思机制

摘要

本篇文章首先对 RAG 的研究背景以及相关工作进行了初步介绍，然后指出了本文的研究所基于的方法-SELF-RAG 算法，接着说明了本文研究方法的工作流程，即介绍了如何将 SELF-RAG 算法和反思机制相结合，然后将本文方法和原本的 SELF-RAG 算法进行一定的对比并对实验结果进行了一定的分析，最后是总结了本次复现所存在的一些问题并给出了未来的一些研究方向。

关键词：自然语言处理；检索增强生成；反思机制

1 引言

随着自然语言处理（Natural Language Processing）的快速发展，基于大规模预训练语言模型（Pre-trained Language Models）的生成技术在文本生成、问答系统等任务中表现出强大的能力。然而，由于参数量的限制和训练数据覆盖范围的不足，传统生成模型在面对事实性问题时容易产生幻觉现象，即容易导致生成内容在回答的准确性和一致性上存在一定缺陷。Retrieval-Augmented Generation (RAG) 技术在这一背景下应运而生，它通过将检索模块（retrieval module）与生成模块（generation module）相结合，在一定程度上弥补了生成模型的缺陷。RAG 的核心思想是利用外部知识库或文档集合，在生成任务中引入实时的事实性支持，从而显著提升生成结果的准确性和可靠性。这一技术被广泛应用于开放领域问题（Open-Domain Question Answering）、知识增强生成（Knowledge Enhanced Generation）等领域，并展示出优异的性能。虽然 RAG 技术取得了较好的效果，但由于缺乏对答案的评估，有可能无法筛选出最佳的结果输出，因此本文基于 SELF-RAG 算法进行研究，并在此基础上加入了反思机制，旨在能够对语言模型生成的结果进行进一步的完善以获得更好的生成结果。

2 相关工作

许多学者对于 RAG 进行了研究。例如，Patrick Lewis 等人在 2020 年的研究 [1] 奠定了 RAG 方法的理论和实践基础。他们的研究表明，通过将检索信息整合到生成模型中，可以显著提高知识密集型任务的准确性。另外，Chen 等人在 2022 年提出了一种多模态检索增强生成（Multimodal-RAG）模型 [2]，专门针对视觉问答任务进行了优化。多模态 RAG 技术通过将文本和图像等异构信息结合，扩展了传统 RAG 在视觉问答和图像描述生成等任务中的应用能力。通过引入多模态知识库和跨模态 Transformer 结构，实现了更高效的信息检索和生成性能，在视觉问答等任务中显著提升了回答的准确性和上下文理解能力。而本文的研究则

主要基于 Akari Asai 等人提出的 SELF-RAG 算法 [3]，该算法使用了一系列特殊 token 来对生成结果进行评估从而筛选出最佳的答案从而显著提升了生成结果的质量。

3 本文方法

3.1 本文方法概述

本文首先训练了批判模型用于为生成结果生成特殊 token 来进行评估，然后根据输入检索相应的文档去辅助生成初步结果，然后使用批判模型对初步结果进行评估之后根据评估结果决定是否进行进一步的反思去对生成的结果进行改善从而提高生成结果的质量。

3.2 批判模型训练

1、基础模型：使用了 llama-2-7b 模型作为用于微调的基本模型

2、训练数据：从公开任务数据集（如 Open-Instruct 和知识密集型任务数据集）中随机抽样，使用一个高级语言模型（如 GPT-4）为这些数据打上标签，即用于评价生成结果的特殊 token。

3、批判模型所生成的特殊 token：

(1) Relevance：分别有 [Relevant] 和 [Irrelevant] 两个值，用于评估检索文档与输入问题的相关性。

(2) Support：分别有 [Fully support] 和 [Partially support] 和 [No support] 三个值，用于评估结果是否支撑搜索文档。

(3) Utility：用于评估答案对问题的帮助程度，范围为 1(最低)-5(最高)。

3.3 推理过程

1、文档检索：根据输入的 query 使用 Contriever 模型从维基百科中检索相关的文本段落，并使用批判模型评估检索文本和 query 的相关性，即生成 Relevance 指标，若不相关则继续搜寻新的文本段落。

2、初步结果生成：将检索到的文档加入到 prompt 当中输入到生成模型中得到初步的结果，生成模型使用的是通用语言模型 llama-2-7b

3、初步结果评估：使用批判模型对初步结果的 Support 和 Utility 指标进行判断，若评分未达到要求，则需要对结果进行进一步的改进。

4、反思与结果改进

(1) 反思：若使用 Support 和 Utility 指标评估答案后发现未满足要求，则使用批判模型去对答案进行反思生成 criticism。

(2) 结果改进：将 criticism 结合 query 和 answer 输入到生成模型当中重新生成新的答案，若新的答案满足要求，则将其作为结果输出，否则则需要重复上述过程去对答案继续进行改进。需要注意设置了 max-critic 参数限制模型进行反思的最大次数。

4 复现细节

4.1 与已有开源代码对比

本文引用了 SELF-RAG 算法训练批判模型有关的代码和检索文档的代码，SELF-RAG 算法中是并行生成多个答案然后对结果进行 rank 从而筛选出最佳的答案，这种方式虽然可以得到质量相对更好的结果，但若生成的一批答案中的质量都较差的话，有可能导致最后筛选出来的结果的质量也不能达到预期，因此本文的工作主要在于使用了反思的机制来对答案进行改进，通过指出答案中存在的问题与缺陷，并基于问题去对答案进行多次的改进或完善直到答案达到要求，从而获得质量更好的生成结果作为最终输出。这样的方式有利于提高生成结果的稳定性，可以有效避免生成结果出现或好或坏的情况。

4.2 训练数据示例

Instructions

Given an instruction and an output, rate whether the response appears to be a helpful and informative answer to the query, from 1 (lowest) - 5 (highest). We call this score perceived utility. The detailed criterion is as follows: 5: The response provides a complete, highly detailed, and informative response to the query, fully satisfying the information needs. 4: The response mostly fulfills the need in the query, while there can be some minor improvements such as discussing more detailed information, having better structure of the response, or improving coherence. 3: The response is acceptable, but some major additions or improvements are needed to satisfy users' needs. 2: The response still addresses the main request, but it is not complete or not relevant to the query. 1: The response is barely on-topic or completely irrelevant.

Instruction Who is the current prime minister of the UK as of 2023?

Output Boris Johnson was the prime minister of the UK from 2019 - 2022.

Perceived utility 2

图 1. 训练数据示意

5 实验结果分析

LM	Short-form		Closed-set		Long-form generations (with citations)					
	PopQA (acc)	TQA (acc)	Pub (acc)	ARC (acc)	Bio (FS)	(em)	(rg)	ASQA (mau)	(pre)	(rec)
<i>LMs with proprietary data</i>										
Llama2-c _{13B}	20.0	59.3	49.4	38.4	55.9	22.4	29.6	28.6	—	—
Ret-Llama2-c _{13B}	51.8	59.8	52.1	37.9	79.9	32.8	34.8	43.8	19.8	36.1
ChatGPT	29.3	74.3	70.1	75.3	71.8	35.3	36.2	68.8	—	—
Ret-ChatGPT	50.8	65.7	54.7	75.3	—	40.7	39.9	79.7	65.1	76.6
Perplexity.ai	—	—	—	—	71.2	—	—	—	—	—
<i>Baselines without retrieval</i>										
Llama2 _{7B}	14.7	30.5	34.2	21.8	44.5	7.9	15.3	19.0	—	—
Alpaca _{7B}	23.6	54.5	49.8	45.0	45.8	18.8	29.4	61.7	—	—
Llama2 _{13B}	14.7	38.5	29.4	29.4	53.4	7.2	12.4	16.0	—	—
Alpaca _{13B}	24.4	61.3	55.5	54.9	50.2	22.9	32.0	70.6	—	—
CoVE _{65B} *	—	—	—	—	71.2	—	—	—	—	—
<i>Baselines with retrieval</i>										
Toolformer* _{6B}	—	48.8	—	—	—	—	—	—	—	—
Llama2 _{7B}	38.2	42.5	30.0	48.0	78.0	15.2	22.1	32.0	2.9	4.0
Alpaca _{7B}	46.7	64.1	40.2	48.0	76.6	30.9	33.3	57.9	5.5	7.2
Llama2-FT _{7B}	48.7	57.3	64.3	65.8	78.2	31.0	35.8	51.2	5.0	7.5
SAIL* _{7B}	—	—	69.2	48.4	—	—	—	—	—	—
Llama2 _{13B}	45.7	47.0	30.2	26.0	77.5	16.3	20.5	24.7	2.3	3.6
Alpaca _{13B}	46.1	66.9	51.1	57.6	77.7	34.8	36.7	56.6	2.0	3.8
Our SELF-RAG _{7B}	54.9	66.4	72.4	67.3	81.2	30.0	35.7	74.3	66.9	67.8
Our SELF-RAG _{13B}	55.8	69.3	74.5	73.1	80.2	31.7	37.0	71.6	70.3	71.3
My result	58.6		57.3							

图 3. 实验结果对比

如图 3 所示：本次复现进行测试时使用了 ARC 和 TQA 两个数据集，这两个数据集的问题类型都是选择题，即从给定的选项中选出正确答案，在进行评估时将模型的生成结果和正确答案进行比较，然后对整体的正确率求平均值作为最后的测试结果。从实验结果可以看出，本次复现使用反思技术对模型的初始生成结果进行改进后可以在一定程度上提高生成结果的正确率，但相比于原论文中的 Self-Rag 算法的正确率仍有一定差距，这主要的原因可能是本次复现中使用的 prompt 相比于原论文中的 prompt 效果较差，输入到生成模型中后得到的结果没有达到预期从而造成了正确率的下降。

6 总结与展望

本文首先介绍了 RAG 的研究背景并对相关工作进行了说明，在介绍相关工作时指出了本文复现所基于的方法 SELF-RAG 算法，然后详细说明了本文复现方法的工作流程并将本文工作和 SELF-RAG 算法进行了一定的对比，最后是将本文的实验结果和其他方法的实验结果进行了对比并进行了分析。本次复现在 prompt 设计以及数据的测试上还存在着一些问题，针对本次复现中存在的一些问题，之后的研究方向主要包括三方面，首先是对 prompt 进行改进并进行对比从而筛选出效果更好的 prompt 去帮助模型生成更加符合预期的结果，然后就是对比不同生成模型的生成效果去筛选出生成效果更好的模型作为该方法的生成模型，最

后就是去对其他更多类型的数据集去进行测试从而评估该方法在不同场景下的表现，进而得出该方法适用于哪些领域。

参考文献

- [1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Proc. Conf. on Neural Information Processing Systems.
- [2] Wenhui Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*. Proc. Int. Conf. on Learning Representations, 2024.