arxiv.org/abs/2305.11738

CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing

开源代码：http://github.com/microsoft/ProphetNet/tree/master/CRITIC

Zhibin Gou[1][2]*, Zhihong Shao[1][2]*, Yeyun Gong[2], Yelong Shen[3],
Yujiu Yang[1]†, Nan Duan[2], Weizhu Chen[3]
[1]Tsinghua University
[2]Microsoft Research Asia, [3]Microsoft Azure AI
{gzb22,szh19}@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn
{yegong, yeshe, nanduan, wzchen}@microsoft.com

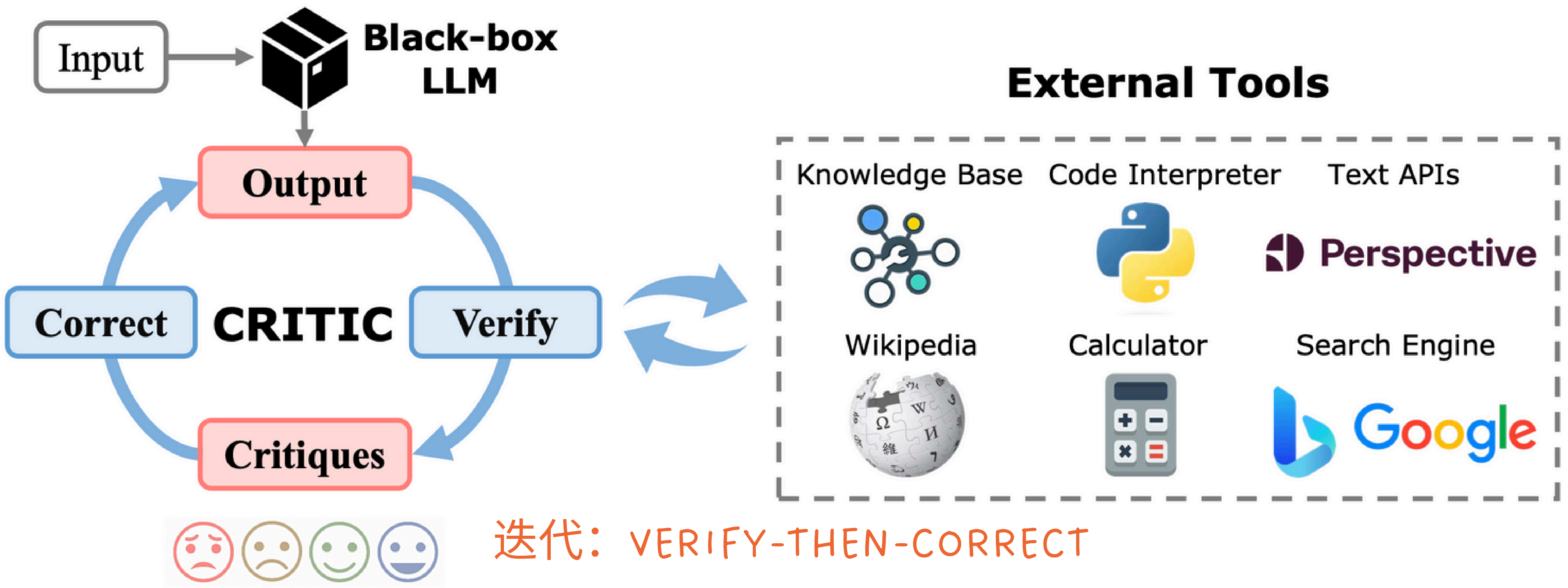
简介

本文提出 CRITIC (Self-Correcting with Tool-Interactive Critiquing) 框架，让LLM对自己生成的response先纠错再改进，核心创新是CRITIC引入了外部工具(比如搜索引擎、代码解释器、毒性检测API)提供更加客观的错误反馈。CRITIC使用迭代式“verify-then-correct” workflow，先对上一轮的response进行纠错，然后将错误信息追加到response，再让LLM生成更好的response。这个过程反复进行，直到生成高质量的response。整个和LLM交互都依赖few-shot prompt。

背景

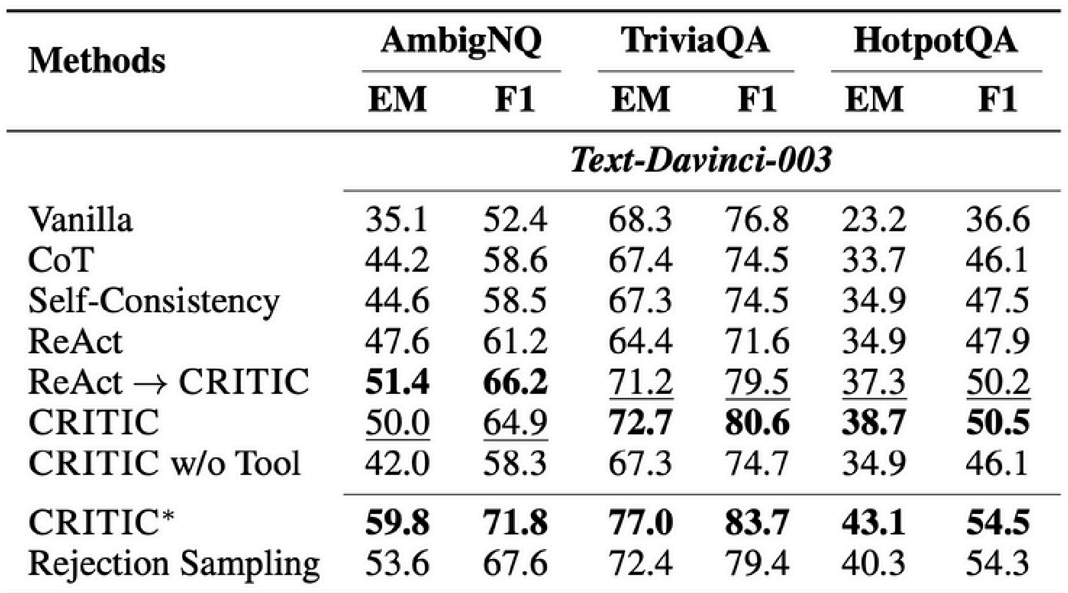
LLM生成的内容仍然可能存在错误。我们人类在遇到不懂的问题时，可以借助Google/百度搜索，补充知识或者纠正大脑中错误的知识，能不能让LLM也像人一样，借助外部tool提升response质量呢？当然是可以的，本文只聚焦基于prompt的方法，提出了CRITIC框架。

CRITIC



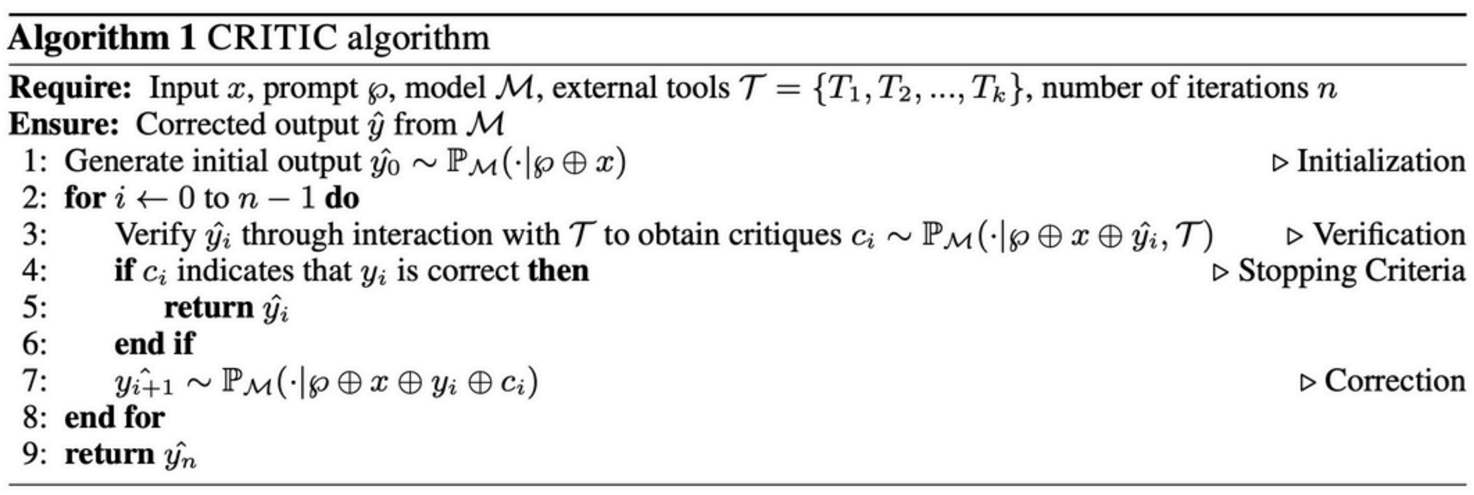
迭代：VERIFY-THEN-CORRECT

部分实验结果



Methods	AmbigNQ		TriviaQA		HotpotQA	
	EM	F1	EM	F1	EM	F1
	Text-Davinci-003					
Vanilla	35.1	52.4	68.3	76.8	23.2	36.6
CoT	44.2	58.6	67.4	74.5	33.7	46.1
Self-Consistency	44.6	58.5	67.3	74.5	34.9	47.5
ReAct	47.6	61.2	64.4	71.6	34.9	47.9
ReAct → CRITIC	**51.4**	**66.2**	71.2	79.5	37.3	50.2
CRITIC	50.0	64.9	**72.7**	**80.6**	**38.7**	**50.5**
CRITIC w/o Tool	42.0	58.3	67.3	74.7	34.9	46.1
CRITIC*	**59.8**	**71.8**	**77.0**	**83.7**	**43.1**	**54.5**
Rejection Sampling	53.6	67.6	72.4	79.4	40.3	54.3

流程



```
Algorithm 1 CRITIC algorithm
Require: Input $x$, prompt $\wp$, model $\mathcal{M}$, external tools $\mathcal{T} = \{T_1, T_2, ..., T_k\}$, number of iterations $n$
Ensure: Corrected output $\hat{y}$ from $\mathcal{M}$
Generate initial output $\hat{y_0} \sim \mathbb{P}_{\mathcal{M}}(\cdot|\wp \oplus x)$                ▷ Initialization
for $i \leftarrow 0$ to $n-1$ do
    Verify $\hat{y_i}$ through interaction with $\mathcal{T}$ to obtain critiques $c_i \sim \mathbb{P}_{\mathcal{M}}(\cdot|\wp \oplus x \oplus \hat{y_i}, \mathcal{T})$    ▷ Verification
    if $c_i$ indicates that $y_i$ is correct then                ▷ Stopping Criteria
        return $\hat{y_i}$
    end if
    $\hat{y_{i+1}} \sim \mathbb{P}_{\mathcal{M}}(\cdot|\wp \oplus x \oplus y_i \oplus c_i)$                ▷ Correction
end for
return $\hat{y_n}$
```

思考

如果去掉tool，还用相同的workflow: verify-then-correct让LLM自我提升，效果如何呢？对应实验表格中的“CRITIC w/o Tool”，答案是不太行。

@机器爱学习