

ReAct: Synergizing Reasoning and Acting in Language Models

Yao, Zhao, Yu, Du, Shafran, Narasimhan, Cao · Princeton / Google Research · ICLR 2023 · arXiv:2210.03629

위자야 히즈키아

CONTRIBUTION

This paper proposes interleaving verbal “thoughts” with real actions so both can guide each other, solving the problem of LLMs being forced to either reason (Chain-of-Thought) or act on an environment.

- Reasoning + Acting
- Reasoning and acting were studied separately.

The gap between reasoning and acting

R Reasoning-only

Thinks step-by-step using only its own frozen internal knowledge — like a closed-book exam.

- Never checks itself against the real world
- Prone to hallucination: states false “facts” confidently
- One wrong claim poisons every later step (error propagation)

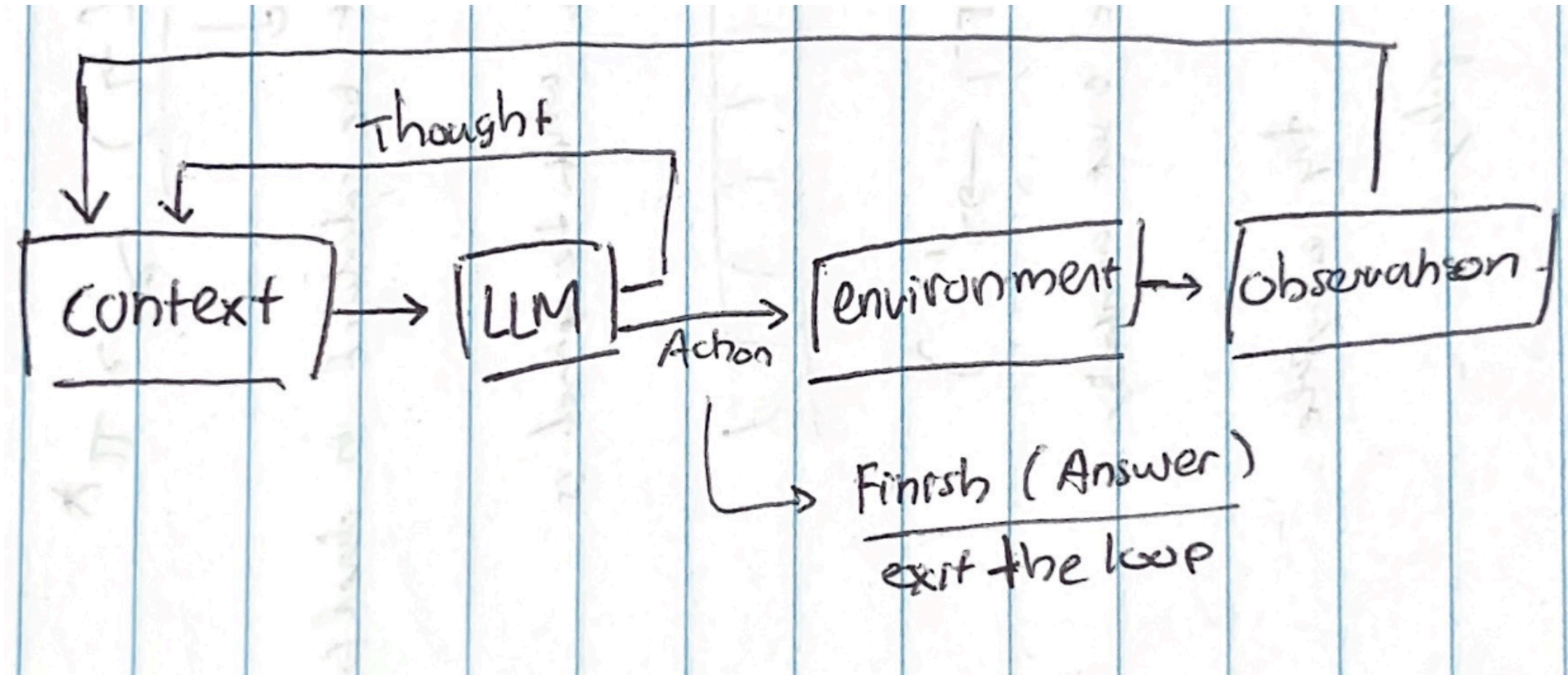
A Acting-only

Takes actions (search, click, move) straight from patterns — no verbal reasoning attached.

- No working memory or high-level plan
- Loses track of the goal mid-task
- Can't handle exceptions or adjust course

Neither alone lets a model reason about what to look up, and look things up to correct its reasoning.

CORE MECHANISM



Thought

Free-form English reasoning. Never touches the outside world — pure planning / interpreting.

Action

One command from a small fixed menu: Search[x], Lookup[x], Finish[x] (or go to / take / open, etc.).

Observation

Whatever the environment hands back — search text, or a game's response.

Context feeds the LLM. The LLM outputs either a Thought or an Action. Actions go out to the Environment and come back as an Observation, which also gets appended to context. This repeats until the Action is Finish, which exits the loop with the answer.

KEY RESULTS

PaLM-540B prompting results

Prompt Method	HotpotQA (EM)	FEVER (Acc)
Standard	28.7	57.1
CoT	29.4	56.3
CoT-SC	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	64.6
ReAct → CoT-SC	35.1	62.0

- Reasoning helps guide acting.
- Combining ReAct with self-consistency (CoT-SC) beats every single method alone.

Human-labeled error analysis

	Type	ReAct	CoT
Success	True positive	94%	86%
	False positive	6%	14%
Failure	Reasoning error	47%	16%
	Search result error	23%	-
	Hallucination	0%	56%
	Label ambiguity	29%	28%

- ReAct works with real search results, it can't hallucinate*
- ReAct is prone to reasoning errors.

*given there exist something to validate to

Assumption: a clean action space + reliable retrieval

- **Rigid action space.** Researchers pre-defined exactly 3–4 actions (Search / Lookup / Finish). Open-ended real-world tasks have no obvious small menu to hand it. rigidity costs flexibility: ReAct's reasoning-error rate is 3× CoT's (47% vs. 16%)
- **Retrieval quality is critical.** 23% of ReAct's own failures come from non-informative search results derailing the whole reasoning chain. This puts a caveat on the 0% hallucination result.
- **Cost is unmeasured.** ReAct needs many more LLM calls per task than a single-shot answer. Every thought and action is a separate generation step. No reports on token count, latency, etc.

CONNECTION

- The Thought→Action→Observation loop here is the direct ancestor of **modern LLM agents**
- ReAct's Search action, gated by reasoning, is the conceptual ancestor of what's now marketed as "**agentic RAG**" systems.

Thank You!