From Language Models to Language Agents

从语言模型到语言智能体

Shunyu Yao
Great, but

- Stateless
- Ungrounded
- Limited Knowledge
Part I: what internal mechanisms are needed?

1. ReAct: reasoning
2. Reflexion: learning
3. ToT: planning

Part II: what external environments are needed?

1. WebShop: web
2. InterCode: code
3. Collie: logic
Part I: Internal Mechanisms
Mechanism 1: Reasoning

Published as a conference paper at ICLR 2023

ReACT: Synergizing Reasoning and Acting in Language Models

Shunyu Yao\textsuperscript{*,1}, Jeffrey Zhao\textsuperscript{2}, Dian Yu\textsuperscript{2}, Nan Du\textsuperscript{2}, Izhak Shafran\textsuperscript{2}, Karthik Narasimhan\textsuperscript{1}, Yuan Cao\textsuperscript{2}

\textsuperscript{1}\textit{Department of Computer Science, Princeton University}
\textsuperscript{2}\textit{Google Research, Brain team}
Reasoning helps Acting

Observation $t$: You are cooking a dish and seeing salt is out...

**Reasoning**:

“The dish should be savory, and since salt is out, I should find the soy sauce instead. It is in the cabinet to my right…”

Action $t$: Turn right

Observation $t+1$: You see a cabinet and a table...

Action $t+1$: Open cabinet

Reason how to act

1. Generalization
2. Alignment

“Verbal Reasoning” as Augmented Action Space!
Internal knowledge can be limited / outdated
Hallucination is serious
Acting is key
Reasoning and Acting

- Language models (LM) get better at reasoning and acting, **separately**
- **ReAct**: Let LMs generate both
  - Reasoning traces (to update internal model belief)
  - Actions (to obtain external environment feedback)
ReAct: Overview

- **Tasks**: Question answering, Fact verification, Text game, Web Interaction
- **Learning**: prompting / finetuning
- **Model**: PaLM-540B / GPT-3

- **Synergy**: Reasoning guides acting, acting supports reasoning
- **Benefits of ReAct**
  - Flexibility: diverse reasoning / interactive tasks
  - Generalization: strong few-shot / fine-tuning performances
  - Alignment: the *human* way of problem solving!
ReAct Prompting and Controlled Baselines

ReAct prompt

<table>
<thead>
<tr>
<th>Question</th>
<th>Which magazine was started first Arthur’s Magazine or First for Women?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thought 1</td>
<td>I need to search Arthur’s Magazine and First for Women, and find which was started first.</td>
</tr>
<tr>
<td>Action 1</td>
<td>Search[Arthur’s Magazine]</td>
</tr>
<tr>
<td>Observation 1</td>
<td>Arthur’s Magazine (1844-1846) was an American literary periodical published in Philadelphia…</td>
</tr>
<tr>
<td>Thought 2</td>
<td>Arthur’s Magazine was started in 1844. I need to search First for Women next.</td>
</tr>
<tr>
<td>Action 2</td>
<td>Search[First for Women]</td>
</tr>
<tr>
<td>Observation 2</td>
<td>First for Women is a woman’s magazine published by Bauer Media Group… was started in 1989…</td>
</tr>
<tr>
<td>Thought 3</td>
<td>First for Women was started in 1989. 1844 (Arthur’s Magazine) &lt; 1989 (First for Women), so Arthur’s Magazine was started first.</td>
</tr>
<tr>
<td>Action 3</td>
<td>Finish[Arthur’s Magazine]</td>
</tr>
</tbody>
</table>

Reason-only prompt (Chain-of-thought, CoT)

<table>
<thead>
<tr>
<th>Question</th>
<th>Which magazine was started first Arthur’s Magazine or First for Women?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thought</td>
<td>Arthur’s Magazine was started in 1844. First for Women was started in 1989. 1844 (Arthur’s Magazine) &lt; 1989 (First for Women), so Arthur’s Magazine was started first.</td>
</tr>
<tr>
<td>Answer</td>
<td>Arthur’s Magazine</td>
</tr>
</tbody>
</table>

Act-only prompt (similar to WebGPT)

<table>
<thead>
<tr>
<th>Question</th>
<th>Which magazine was started first Arthur’s Magazine or First for Women?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action 1</td>
<td>Search[Arthur’s Magazine]</td>
</tr>
<tr>
<td>Observation 1</td>
<td>Arthur’s Magazine (1844-1846) was an American literary periodical published in Philadelphia…</td>
</tr>
<tr>
<td>Action 2</td>
<td>Search[First for Women]</td>
</tr>
<tr>
<td>Observation 2</td>
<td>First for Women is a woman’s magazine published by Bauer Media Group… was started in 1989…</td>
</tr>
<tr>
<td>Action 3</td>
<td>Finish[Arthur’s Magazine]</td>
</tr>
</tbody>
</table>

Standard prompt

<table>
<thead>
<tr>
<th>Question</th>
<th>Which magazine was started first Arthur’s Magazine or First for Women?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer</td>
<td>Arthur’s Magazine</td>
</tr>
</tbody>
</table>
Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

Answer: iPod

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch
Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software).
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]
Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software ...

Act 4: Finish[yes]
Shridhar et al. "ALFWorld: Aligning Text and Embodied Environments for Interactive Learning."
You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 1, a coffee machine 1, a countertop 3, a stove burner 1, and a toaster 1.

**Your task is to:** Put some pepper shaker on a drawer.

### Act-Only

**Act 1:** Go to drawer 1  
**Obs 1:** The drawer 1 is closed.

**Act 2:** Open drawer 1  
**Obs 2:** You open the drawer 1. The drawer 1 is open. In it, you see a dishsponge 2, and a spoon 1.

**Act 3:** Go to sinkbasin 1  
**Obs 3:** On the sinkbasin 1, you see a dishsponge 3, a spatula 1, and a spoon 2.

**Act 4:** Take peppershaker 1 from sinkbasin 1  
**Obs 4:** Nothing happens.

**Act 5:** Take peppershaker 1 from sinkbasin 1  
**Obs 5:** Nothing happens.
You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 1, a coffee machine 1, a countertop 3, a stove burner 1, and a toaster 1.

Your task is to: Put some pepper shaker on a drawer.

ReAct (Reason + Act)

Act 1: Think[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ... ]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
   (Here the agent goes to cabinet 1, 2, 3, 4, 5, 6, then countertop 1 and 2)
Act 9: Go to countertop 3
Obs 9: On the countertop 3, you see a apple 1, a bread 1, ...... a pepper shaker 1, and a vase 3.
Act 10: Take pepper shaker 1 from countertop 3
Obs 10: You pick up the pepper shaker 1 from the countertop 3.

Act 11: Think[Now I find a pepper shaker 1. Next, I need to put it in/on drawer ].
Act 12: Go to drawer 1
Obs 12: Drawer 1 is closed.

Act 13: Open drawer 1
Obs 13: You open Drawer 1 ...

Act 14: Put pepper shaker 1 in/on drawer 1
Obs 14: You put pepper shaker 1 in/on the drawer 1.

✓
You are in the middle of a room. Looking quickly around you, you see an armchair 1, a cabinet 1, a drawer 1, a dresser 1, a garbage can 1, a safe 1, a shelf 1, a sidetable 1, and a sofa 1.

Your task is to: put two keychain in safe.

(a) ReAct

| Obs 16 | You open the drawer 4. The drawer 4 is open. In it, you see a keychain 3, and a watch 2. |
| Act 17 | Think [Now I find the first keychain (3). Next, I need to take it. I can find the second keychain (2) later in drawer 4.] |
| Act 18 | take keychain 3 from drawer 4 |
| Obs 18 | You pick up the keychain 3 from the drawer 4. |
| Act 22 | put keychain 3 in/on safe 1 |
| Obs 22 | You put the keychain 3 in/on the safe 1. |
| Act 23 | Think [Now I put the first keychain in safe. Next, I need to find the second keychain. I can directly go to drawer 4.] |
| Act 24 | go to drawer 4 |
| Obs 24 | The drawer 4 is open. In it, you see a watch 2. |
| Act 25 | Think [Now I find the second keychain (2). Next, I need to take it.] |
| Act 26 | take keychain 2 from drawer 4 |
| Obs 26 | Nothing happens. |
You are in the middle of a room. Looking quickly around you, you see a armchair 1, a cabinet 4, ... a cabinet 1, a drawer 5, ..., a drawer 1, a dresser 1, a garbagecan 1, a safe 1, a shelf 12, ..., a shelf 1, a sidetable 1, and a sofa 1.

Your task is to: put two keychain in safe.

(b) ReAct (Human Thought edit)

Obs 16: You open the drawer 4. The drawer 4 is open. In it, you see a keychain 3, and a watch 2.
Act 17: Think [Now I find the first keychain (3). Next, I need to take it. I can find the second keychain (2) later in drawer 4.]
Act 18: take keychain 3 from drawer 4
Obs 18: You pick up the keychain 3 from the drawer 4.

Obs 22: You put the keychain 3 in/on safe 1.
Act 23: Think [Now I put the first keychain in safe. Next, I need to find the second keychain, which is more likely to appear in dresser 1, garbagecan 1, safe 1, sidetable 1, sofa 1, shelf 1-12.]
Act 24: go to dresser 1
Act 25: Think [Now I find the second keychain (2). Next, I need to take it.]
Act 26: take keychain 2 from dresser 1
Obs 26: You pick up the keychain 2 from the dresser 1.
Act 27: Think [Now I take the second keychain (2). Next, I need to put it in/on safe 1.]
Act 28: go to safe 1
Act 29: put keychain 2 in/on safe 1
Obs 29: You put the keychain 2 in/on the safe 1.
Finetuning > Prompting

- Prompting only works with LLMs, limited learning support
- Finetuning is promising (initial results on HotpotQA, using prompted trajectories)
  - ReAct finetuned small LMs > ReAct prompted large LMs
  - ReAct finetuning is better than other finetuning formats across model sizes
ReAct: Summary

- Two basic functions of language agents: reason and act
  - Many followups since 2022: LangChain, AutoGPT, ...

- Simple view: **action space**

- Slightly more complex view: **working memory**
  - LM can “talk” to memory (reason) or env (act)
  - Memory short-term, append-only
  - What if memory is long-term, not append-only?
Mechanism 2: Learning

Reflexion: Language Agents with Verbal Reinforcement Learning

Noah Shinn
Northeastern University
noahshinn024@gmail.com

Beck Labash
Northeastern University
labash.b@northeastern.edu

Karthis Narasimhan
Princeton University
karthikn@princeton.edu

Federico Cassano
Northeastern University
cassano.f@northeastern.edu

Ashwin Gopinath
Massachusetts Institute of Technology
agopi@mit.edu

Shunyu Yao
Princeton University
shunyuy@princeton.edu
(Reinforcement) Learning

- Behavior -> Feedback -> Update -> Better Behavior
Learning: Feedback

- Behavior -> **Feedback** -> Update -> Better Behavior

- **Scalar Feedback:** 0 or 1

- **Language Feedback:**
  - Runtime error messages
  - Unit test cases and results
  - Human/self-feedback: “the code did not handle this corner case...”
Learning: Update

- Behavior -> Feedback -> **Update** -> Better Behavior

- **Parameter Update:** PPO, A3C, DQN ...

- **Language Update:**
  - “Be sure to handle this corner case…”
Reflexion: “Verbal” RL

- RL: Behavior -> Feedback -> Update -> Better Behavior
  - Traditional RL: scalar feedback, parameter update
  - “Verbal” RL: language feedback, language update

- **Key idea**: LM can self-evaluate/reflect/critic
  - Works with various tasks (game, QA, coding, ...)
  - Works with various models (ReAct, CoT, ...)
  - Works with various feedback sources (env, human, self-generated, ...)
  - **Significantly improves the agent across episodes, using long-term memory update**
Decision Making (ALFWorld)

1. Task
2. Action + observation trajectory
3. Evaluation
4. Self-reflection
5. Repeat

Coding (multi-task, multi-lingual)

<table>
<thead>
<tr>
<th>Benchmark + Language</th>
<th>Prev SOTA Pass@1</th>
<th>SOTA Pass@1</th>
<th>Reflexion Pass@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HumanEval (PY)</td>
<td>65.8 (CodeT [4] + GPT-3.5)</td>
<td>80.1 (GPT-4)</td>
<td>91.0</td>
</tr>
<tr>
<td>HumanEval (RS)</td>
<td>60.0 (GPT-4)</td>
<td>68.0</td>
<td></td>
</tr>
<tr>
<td>MBPP (PY)</td>
<td>67.7 (CodeT [4] + Codex [5])</td>
<td>80.1 (GPT-4)</td>
<td>77.1</td>
</tr>
<tr>
<td>MBPP (RS)</td>
<td>70.9 (GPT-4)</td>
<td>75.4</td>
<td></td>
</tr>
<tr>
<td>Leetcode Hard (PY)</td>
<td>7.5 (GPT-4)</td>
<td>15.0</td>
<td></td>
</tr>
</tbody>
</table>

1. (Instruction)
2. (Function implementation)
3. (Unit test feedback) ← execution
4. (Self-reflection)
5. (Repeat)
Importance of Unit Test

Test generation omission
1. (Instruction)
2. (Function implementation)
3. (Unit test feedback) → execution
4. (Self-reflection)
5. (Repeat)

Self-reflection omission
1. (Instruction)
2. (Function implementation)
3. (Unit test feedback) ← execution
4. (Self-reflection)
5. (Repeat)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Test Generation</th>
<th>Self-reflection</th>
<th>Pass@1 (Acc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base model</td>
<td>False</td>
<td>False</td>
<td>0.60</td>
</tr>
<tr>
<td>Test generation omission</td>
<td>False</td>
<td>True</td>
<td>0.52</td>
</tr>
<tr>
<td>Self-reflection omission</td>
<td>True</td>
<td>False</td>
<td>0.60</td>
</tr>
<tr>
<td>Reflexion</td>
<td>True</td>
<td>True</td>
<td>0.68</td>
</tr>
</tbody>
</table>
Reasoning (HotpotQA)

1. (Question)
2. (Knowledge retrieval)
3. (Answer)
4. (Exact match evaluation)
5. (Self-reflection)
6. (Repeat)

Importance of Reflection

1. (Question)
2. (Knowledge retrieval)
3. (Answer)
4. (Exact match evaluation)
5. (Self-reflection)
6. (Repeat)
Reflexion: Summary

● A verbal / “no-gradient” approach to learning
  ○ Concurrent work: self-refine, self-debugging, ...

● Simple view: append reflection to prompt
● Slightly more complex view: long-term memory

● Limitation?
Mechanism 3: Planning

Tree of Thoughts: Deliberate Problem Solving with Large Language Models

Shunyu Yao
Princeton University

Dian Yu
Google DeepMind

Jeffrey Zhao
Google DeepMind

Izhak Shafran
Google DeepMind

Thomas L. Griffiths
Princeton University

Yuan Cao
Google DeepMind

Karthik Narasimhan
Princeton University
Motivating Task: Game of 24

Use numbers and basic arithmetic operations (+ - * /) to obtain 24. Each step, you are only allowed to choose two of the remaining numbers to obtain a new number.

Input: 4 4 6 8

Steps: 4 + 8 = 12 (left: 4 6 12); 6 - 4 = 2 (left: 2 12); 2 * 12 = 24 (left: 24)

Answer: (6 - 4) * (4 + 8) = 24

{more CoT examples...}

Input: 4 5 6 10

Steps: 5 - 4 = 1 (left: 1 6 10); 10 - 1 = 9 (left: 6 9); 6 * 4 = 24 (left: 24)

Answer: (10 - (5 - 4)) * 6 = 24

Remember: LM was designed to generate text autoregressively, not to act or make decisions!

<table>
<thead>
<tr>
<th>Method</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO prompt</td>
<td>7.3%</td>
</tr>
<tr>
<td>CoT prompt</td>
<td>4.0%</td>
</tr>
<tr>
<td>CoT-SC (k=100)</td>
<td>9.0%</td>
</tr>
<tr>
<td>IO + Refine (k=10)</td>
<td>27%</td>
</tr>
<tr>
<td>ToT (ours) (b=1)</td>
<td>45%</td>
</tr>
<tr>
<td>ToT (ours) (b=5)</td>
<td>74%</td>
</tr>
</tbody>
</table>

Why hard even for GPT-4 + CoT?

No backtrack: First couple tokens ("5 -") wrong, game over... (In fact, 60% CoT samples fail after first step)

Token decision: really hard to tell if first token "5" is good or bad
Tree of Thoughts (ToT)

ToT recipe:

1. What is thought?
2. How to generate it?
3. How to evaluate it?
4. How to search?
0. Why Tree Search?

“A genuine problem-solving process involves the repeated use of available information to initiate exploration, which discloses, in turn, more information until a way to attain the solution is finally discovered.”

“Many kinds of information can aid in solving problems: information may suggest the order in which possible solutions should be examined; it may rule out a whole class of solutions ... All these kinds of information are heuristics --- things that aid discovery.”

- At the root of AI, tree search was invented as a general problem solving process
- Hand-written heuristics limited the generality of symbolic AI
- Marry the tree search’s power as algorithm + LLM’s power as heuristics
1. What is thought?

**Bandit of outputs**
- Easy to evaluate
- Hard to generate

**Tree of tokens**
- Easy to generate
- Hard to evaluate

**Tree of thoughts**
A nice tradeoff game between generation vs evaluation

(a) \[\begin{align*}4/9/10/13 & \\
4+9=13 & 9-4=5 \\
13-10=3 & 10/5=2 \\
3*13=39 & 2*13=26 \\
\end{align*}\]

(b) \[\begin{align*}4/9/10/13 & \\
9 & 13 \\
\end{align*}\]

(c) \[\begin{align*}4/9/10/13 & \\
4+9=13 & 13-9=4 \\
4+4=8 & 10-4=6 \\
4*6=24 & 4+6=10 \\
\end{align*}\]
2. How to generate thoughts?

Option A. Sample i.i.d. Thoughts
- Good for long thoughts
- Might generate repetitive short thoughts

Use numbers and basic arithmetic operations (+, -, *, /) to obtain 24. Each step, you are only allowed to choose two of the remaining numbers to obtain a new number.
Input: 4 4 6 8
Steps:
4 + 8 = 12 (left: 4 6 12)
6 - 4 = 2 (left: 2 12)
2 * 12 = 24 (left: 24)
Answer: (6 - 4) * (4 + 8) = 24
Input: 2 9 10 12
Steps:
4 runs
GPT-4
(stop=\n)

4 * 6 = 24 (left: 5 10 24)
5 * 4 = 20 (left: 6 10 20)
4 * 5 = 20 (left: 6 10 20)
5 * 4 = 20 (left: 6 10 20)
2. How to generate thoughts?

**Option A. Sample i.i.d. Thoughts**
- Good for long thoughts
- Might generate repetitive short thoughts

**Option B. Propose sequential thoughts**
- Make sure thoughts are unique in the same context

---

**Input:** 2 8 8 14  
**Possible next steps:**
- \(2 + 8 = 10\) (left: 8 10 14)
- \(8 / 2 = 4\) (left: 4 8 14)
- \(14 + 2 = 16\) (left: 8 8 16)
- \(2 \times 8 = 16\) (left: 8 14 16)
- \(8 - 2 = 6\) (left: 6 8 14)
- \(14 - 8 = 6\) (left: 2 6 8)
- \(14 / 2 = 7\) (left: 7 8 8)
- \(14 - 2 = 12\) (left: 8 8 12)

**Input:** 4 5 6 10  
**Possible next steps:**
- \(4 + 5 = 9\) (left: 6 9 10)
- \(5 + 6 = 11\) (left: 4 10 11)
- \(10 - 4 = 6\) (left: 5 6 6)
- \(4 \times 5 = 20\) (left: 6 10 20)
- \(10 / 5 = 2\) (left: 2 4 6)
- \(6 - 4 = 2\) (left: 5 2 10)
- \(10 - 6 = 4\) (left: 4 4 5)
- \(6 / 4 = 1.5\) (left: 5 1.5 10)
3. How to evaluate thoughts?

**Option A. Value independently**
- When final “success” is tractable
- More priors injectable
  - E.g. Promote with simulation success
  - E.g. Reject with commonsense

Input: 4 9 10 13

(b) Value Prompt
Evaluate if given numbers can reach 24 (sure/likely/impossible)
10 14: 10 + 14 = 24. sure
10 13 13

Thought Evaluation
(13 - 10) * 13 = 3 * 13 = 39
10 + 13 + 13 = 36
There is no way to obtain 24 with these big numbers. impossible
*3. How to evaluate thoughts?

**Option A. Value independently**
- When final “success” is tractable
- More priors injectable
  - E.g. Promote with simulation success
  - E.g. Reject with commonsense

**Option B. Vote across**
- When “success” is open-ended
- Stepwise self-consistency / democracy
  - Can be zero-shot

Given an instruction and several choices, decide which choice is most promising. Analyze each choice in detail, then conclude in the last line "The best choice is {s}", where s the integer id of the choice.
4. How to search?

- Breadth-first search (BFS)
  - Maintain a set of promising thoughts per step
  - Branch and prune per step
  - **Good for:** problems with shallow trees (e.g. <=3 steps), but hard to get initial choices right

- Depth-first search (DFS)
  - Always explore the most promising thought
  - Until self-evaluation is “bad”, then backtrack
  - **Good for:** problems with deep trees (e.g. >= 10 steps), but possible to get initial choices right
## Tasks

<table>
<thead>
<tr>
<th></th>
<th><strong>Game of 24</strong></th>
<th><strong>Creative Writing</strong></th>
<th><strong>5x5 Crosswords</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>4 numbers (4 9 10 13)</td>
<td>4 random sentences</td>
<td>10 clues (h1. presented;..)</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>An equation to reach 24</td>
<td>A passage of 4 paragraphs ending in the 4 sentences</td>
<td>5x5 letters: SHOWN; WIRRA; AVAIL; ...</td>
</tr>
<tr>
<td><strong>Thoughts</strong></td>
<td>3 intermediate equations</td>
<td>A short writing plan</td>
<td>Words to fill in for clues: (h1. shown; v5. naled; ...)</td>
</tr>
<tr>
<td></td>
<td>(13-9=4 (left 4,4,10); 10-4=6 (left 4,6); 4*6=24)</td>
<td>(1. Introduce a book that connects...)</td>
<td></td>
</tr>
<tr>
<td><strong>#ToT steps</strong></td>
<td>3</td>
<td>1</td>
<td>5-10 (variable)</td>
</tr>
</tbody>
</table>
Game of 24

1. Thought: each intermediate equation
2. Generate thoughts: **propose**
3. Evaluate thoughts: **value**
4. Search algorithm: **BFS** (depth=4, breadth<=5)
Game of 24: Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO prompt</td>
<td>7.3%</td>
</tr>
<tr>
<td>CoT prompt</td>
<td>4.0%</td>
</tr>
<tr>
<td>CoT-SC (k=100)</td>
<td>9.0%</td>
</tr>
<tr>
<td>ToT (ours) (b=1)</td>
<td>45%</td>
</tr>
<tr>
<td>ToT (ours) (b=5)</td>
<td>74%</td>
</tr>
<tr>
<td>IO + Refine (k=10)</td>
<td>27%</td>
</tr>
<tr>
<td>IO (best of 100)</td>
<td>33%</td>
</tr>
<tr>
<td>CoT (best of 100)</td>
<td>49%</td>
</tr>
</tbody>
</table>

Table 2: Game of 24 Results.

Figure 3: Game of 24 (a) scale analysis & (b) error analysis.
Creative Writing

Write a coherent passage of 4 short paragraphs. The end sentence of each paragraph must be:
1. It isn't difficult to do a handstand if you just stand on your hands.
2. It caught him off guard that space smelled of seared steak.
3. When she didn’t like a guy who was trying to pick her up, she started using sign language.
4. Each person who knows you has a different perception of who you are.

Plan 1
1. Introduce and explain the technique of doing a handstand
2. Switch to a story about an astronaut's first time in space
3. Describe a situation where a woman uses sign language to avoid unwanted attention
4. The final paragraph explains how everyone has different perceptions of others

Plan 2
1. Introduction to an unusual self-help book, mentioning a handstand as a metaphor for embracing challenges.
2. Discuss the unexpected things learned from astronauts, including the smell of space.
3. Describe a woman's clever tactic for avoiding unwanted attention at a bar.
4. Contemplate how different perceptions of oneself can shape one's identity.
1. Thought: just one intermediate plan
2. Generate thoughts: **sample**
3. Evaluate thoughts: **vote**
4. Search algorithm: **BFS** (depth=2, breadth=5)

---

Write a coherent passage of 4 short paragraphs. The end sentence of each paragraph must be: 1. It isn’t difficult to do a handstand if you just stand on your hands. 2. It caught him off guard that space smelled of seared steak. 3. When she didn’t like a guy who was trying to pick her up, she started using sign language. 4. Each person who knows you has a different perception of who you are.

Plan 1: 1. Introduce and explain the technique of doing a handstand. 2. Switch to a story about an astronaut’s first time in space. 3. Describe a situation where a woman uses sign language to avoid unwanted attention. 4. The final paragraph explains how everyone has different perceptions of others.

Plan 2: 1. Introduction to an unusual self-help book, mentioning a handstand as a metaphor for embracing challenges. 2. Discuss the unexpected things learned from astronauts, including the smell of space. 3. Describe a woman’s clever tactic for avoiding unwanted attention at a bar. 4. Contemplate how different perceptions of oneself can shape one’s identity.

Analyzing each choice in detail: Choice 1, while incorporating the required end sentences, seems to lack a clear connection between the paragraphs. Choice 2 offers an interesting perspective by using the required end sentences to present a self-help book’s content. It connects the paragraphs with the theme of self-improvement and embracing challenges, making for a coherent passage. The best choice is 2.
Creative Writing: Results

(a) GPT-4 coherency scores

(b) Human coherency comparison
Mini Crosswords

In order to indicate an answer, you must click on the number associated with its definition.

**Horizontal**

1. An agendum; something to be done
2. An engine
3. Pretentious; flowery
4. A salon; a hall
5. To mock; to sneer

**Vertical**

1. To heap
2. An Indian antelope
3. To intend; to plan; to devise; a nettle; to guess
4. A nozzle
5. Desiccator; more dry

https://www.goobix.com/crosswords/0505/1/
1. Thought: each clue word
2. Generate thoughts: propose
3. Evaluate thoughts: value
4. Search algorithm: DFS (depth=5-10)

Importance of ...
- Backtrack
- Subtree pruning
ToT: Summary

- LM + tree search complement each other
  - LM provides flexible and versatile **heuristics** for search
  - Search augments **control structure** for LLM

- **But how is this related to language agents?**
We have found some piecewise insightful ideas:
- Language as universal interface across mechanisms

<table>
<thead>
<tr>
<th>Action space</th>
<th>RL Agent</th>
<th>Language Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
<td>Parameter update</td>
<td>Language update</td>
</tr>
<tr>
<td>Planning</td>
<td>Learned values</td>
<td>LM reasoning for both generation and evaluation</td>
</tr>
</tbody>
</table>

A more systematic framework to synthesize these ideas?
Part II: External Environments
Environment: how to be cheap, fast, yet useful?

**Physical World / Humans**
- 😊 **Practical**: robots / chatbots
- 😞 **Not Scalable**: expensive and slow to collect data

**Simulations / Games**
- 😞 **Not Practical**: sim-to-real is hard
- 😊 **Scalable**: free, unlimited interactions

**Digital World (Internet, code, software, …)**
- 😊 **Practical**: important tasks to automate
- 😊 **Scalable**: huge scale, rich complexity, free and fast
Evaluation: how to be cheap, fast, yet high-quality?

Human Evaluation

😊 High-quality (esp. if you’re OpenAI): RLHF

😢 Not Scalable (if you’re not OpenAI): expensive and slow to collect data

LM Evaluation

😢 Not High-quality (yet): hallucinations

😊 Scalable (somewhat): unlimited, but with costs

“Rule-based” Evaluation

😊 High-quality: if we leverage domain priors!

😊 Scalable: unlimited, free, fast
## Overview

<table>
<thead>
<tr>
<th></th>
<th>WebShop</th>
<th>InterCode</th>
<th>Collie</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task</strong></td>
<td>“Find a t-shirt that…”</td>
<td>“Remove the file that…”</td>
<td>“Write a paragraph that…”</td>
</tr>
<tr>
<td><strong>Action</strong></td>
<td>Web actions</td>
<td>Code as action</td>
<td>Text as action</td>
</tr>
<tr>
<td><strong>Observation feedback</strong></td>
<td>Webpage</td>
<td>Code execution result</td>
<td>Synthetic constraint satisfaction feedback</td>
</tr>
<tr>
<td><strong>Reward feedback</strong></td>
<td>Attribute rule-based</td>
<td>Unit test or file diff</td>
<td>Grammar rule-based</td>
</tr>
</tbody>
</table>
Web Interaction

How can I train the crows in my neighborhood to bring me gifts?

This question does not make sense
This question should not be answered

Search results for: how to train crows to bring you gifts

How to Make Friends With Crows - PetHelpful
If you did this a few times, your crows would learn your new place, but as I said, I’m not sure if they will follow or visit you there since it’s probably not in their territory. The other option is simply to make new crow friends with the crows that live in your new neighborhood.

Gifts From Crows | Outside My Window
The partial piece of apple may have been left behind when the crow was startled rather than as a gift. If the crows bring bright objects you’ll know for sure that it’s a gift because it’s not something they eat. Brandi Williams says: May 28, 2020 at 7:19 am.

Quotes

[(i) Gifts From Crows | Outside My Window (www.birdoutsideMyWindow.org)]
Many animals give gifts to members of their own species but crows and other corvids are the only ones known to give gifts to humans.

Past actions
Search how to train crows to bring you gifts
Click Gifts From Crows | Outside My Window www.birdoutsideMyWindow.org
Quote
Title
Search results for: how to train crows to bring you gifts

Scroller: 0 - 11
Text:
(i) How to Make Friends With Crows - PetHelpfulpethelpful.com
If you did this a few times, your crows would learn your new place, but as I said, I’m not sure if they will follow or visit you there since it’s probably not in their territory. The other option is simply to make new crow friends with the crows that live in your new neighborhood.

Gifts From Crows | Outside My Window www.birdoutsideMyWindow.org
The partial piece of apple may have been left behind when the crow was startled rather than as a gift. If the crows bring bright objects you’ll know for sure that it’s a gift because it’s not something they eat. Brandi Williams says: May 28, 2020 at 7:19 am.

Actions left: 96
Next action

Nogueira et al. End-to-End Goal-Driven Web Navigation
Nakano et al. WebGPT: Browser-assisted question-answering with human feedback

Mini WWikiNav: WebGPT: RLHF games”mes”
Why Shopping?

- Scalable **environment**: rich dynamics, scalable items
- Scalable **task**: well-defined problem, possible for automatic reward synthesis
- Interesting **challenges**: multi-modal understanding, decision making, etc.
Scrape 1.18M products from amazon.com, text mine attributes

- Optical zoom
- LCD screen
- Memory card
- Digital camera
Construction

- Scrape 1.18M products from amazon.com, text mine attributes
- Build synthetic website with aligned text interface
Construction

- Scrape 1.18M products from amazon.com, text mine attributes
- Build synthetic website with aligned text interface
- **Human** instructions & **Automatic** reward via product attributes/options/prices/types

Goal product (U)

Fujifilm X-T1 16 MP Mirrorless Digital Camera with 3.0-inch LCD (Body Only) (Graphite Silver & Weather Resistant) (Renewed)

Attributes:
- certified refurbished
- water resistant
- high performance

Goal Instruction

"I'd like to find a digital camera that's water resistant. The color needs to be graphite silver and I want the configuration to be the international version, and price under 1000 dollars"

- Goal Attributes: [water resistant]
- Goal Options: [graphite silver, international version]
Construction

- Scrape 1.18M products from amazon.com, text mine attributes
- Build synthetic website with aligned text interface
- **Human** instructions & **Automatic** reward via product attributes/options/prices/types

---

**Goal Instruction**

"I'd like to find a digital camera that's water resistant. The color needs to be graphite silver and I want the configuration to be the international version, and price under 1000 dollars"

- Goal Attributes: [water resistant]
- Goal Options: [graphite silver, international version]

---

**Automatic Reward Calculation**

\[
r = r_{type} \cdot \frac{|U_{att} \cap Y_{att}| + |U_{opt} \cap Y_{opt}| + 1[y_{price} \leq u_{price}]}{|U_{att}| + |U_{opt}| + 1}
\]

\[
r_{type} \in \{0,0.1,0.5,1\}
\]

1. attribute matching = 0/1
2. option matching = 2/2
3. price matching = 1/1
4. Type matching = 1.0

\[
r = 1.0 \cdot \frac{0 + 2 + 1}{1 + 2 + 1} = 0.75
\]
Sim-to-real transfer (Amazon/eBay)

Learned policy generalizes to different search engine and products
WebShop: Summary

- First large-scale, realistic web interaction benchmark
  - Trending in 2023: WebArena, Mind2Web, ...
- **Synthetic website** provides controlled development, transfers to real websites
- **Task priors** (self/model-supervision) provides scalable reward

(Individual) Human reward finetuning on top of synthetic reward pre-training?
Code Interaction

- Static NL2Code benchmarks: HumanEval, Spider, NL2Bash, ...
- But humans code in a fundamentally interactive manner!
- Some interactive/execution-based methods, but no standard benchmark
InterCode Setup

- Standard RL env:
  - **Environment**: Docker-based Python/SQL/bash terminals
  - **Action**: code command
  - **Observation**: execution result

- Benefits
  - Safe and reproducible
  - Unlock new tasks (e.g. CTF)
  - Unlock new evaluations (e.g. Bash)
  - Unlock new methods (e.g. Plan-and-solve)
InterCode: new methods

- Interactive >> seq2seq
- Different interactive methods have different tradeoffs
- Large room for improvement
Future: Coding -> Software engineering?

Task
Challenge Description:
You have access to the file `unpackme.flag.py` that contains the hidden flag. Find the flag's value.
Flag: picoCTF{175_chr157m45_85f5d0ac}

#1 Bash Skill
Check `unpackme.flag.py` content
Agent
`/bin/bash -c "cat unpackme.flag.py"`
Bash Env
```
import base64; from cryptography.fernet import Fernet
payload = b'...'
key_str = 'correctts place...'
key_base64 = base64.b64encode(...)
```

#2 Python Skill
Recognize it's python code and run it
Agent
`python3 -c "import base64; ..."`
Bash Env
Traceback (most recent call last):
  File "...", line 1, in ...
ModuleNotFoundError: 'cryptography'

#3 Error Correction
Know how to resolve missing crypto lib
Agent
`pip3 install cryptography`
Bash Env
Collecting cryptography, Downloading cryptography-40.0.2 ...
Successfully installed ...

#4 Cryptography
Recognize picoCTF output is the flag
Agent
`python3 -c "import base64; ..."` (Repeat #2)
Bash Env
```
if input('>') == '...':
    print('picoCTF{175_chr157m45_85f5d0ac}')
```

...
Constrained Text Generation

- A traditional and important NLP (seq2seq) task: constraints -> text
- **Prior benchmarks**: fixed constraint type and too simple for LLMs
  - “Generate a sentence with dog, catch, happy.”
- **Collie’s goal**:
  - Diverse and arbitrarily hard constraints for LLMs, yet guaranteed to be solvable
  - Automatic task construction + evaluation, without human efforts!
  - Challenge language understanding/generation, semantic planning, logical/arithmetic reasoning, ...
- **Core idea**: leverage the infinite expressivity of grammar
Collie: Grammar

\[ S \rightarrow (\text{level}(\xi) = \ell) \land M \]  
\[ M \rightarrow C \mid C \land M \mid C \lor M \]  
\[ C \rightarrow \text{count}(T, \ell, v_{\text{str}} \mid \ell') \oplus v_{\text{num}} \mid \text{pos}(T, \ell, v_{\text{num}}) \circ v_{\text{str}} \]  
\[ T \rightarrow \xi \mid \text{pos}(T, \ell, v_{\text{num}}) \]  
\[ \ell \rightarrow \text{char} \mid \text{word} \mid \text{sentence} \mid \text{paragraph} \mid \text{passage} \]  
\[ \circ \rightarrow = \mid \neq \mid > \mid < \mid \leq \mid \geq \]  
\[ v_{\text{str}} \in \Sigma^* \quad v_{\text{num}} \in \mathbb{Z} \]

1. (constraint specification)  
2. (multi-constraint)  
3. (base-constraint)  
4. (text)  
5. (level)  
6. (relation)  
7. (value)

- **Few core concepts:** count, pos, level
- But compositionality yields power
- Easily extensible (e.g. POS, sentiment, topic, ...)

\[ \theta \rightarrow = \mid \neq \mid > \mid < \mid \leq \mid \geq \]
Collie-v1: 2,080 Constraints across 13 Types

<table>
<thead>
<tr>
<th>ID</th>
<th>Example instruction</th>
<th>Multi-constraint $\mathcal{M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>word01</td>
<td>Generate a word with at least 15 letters.</td>
<td>$\text{count}(\xi, \text{char}, \text{word}) \geq 15$</td>
</tr>
<tr>
<td>word02</td>
<td>Generate a word with 10 letters, where letter 1 is 's', letter 3 is 'r', letter 9 is 'e'.</td>
<td>$\text{count}(\xi, \text{char}, \text{word}) = 10 \land \text{pos}(\xi, \text{char}, 1) = 's' \land \text{pos}(\xi, \text{char}, 3) = 'r' \land \text{pos}(\xi, \text{char}, 9) = 'e'$</td>
</tr>
<tr>
<td>word03</td>
<td>Generate a word with at most 10 letters and ends with 'r'.</td>
<td>$\text{count}(\xi, \text{char}, \text{word}) \leq 10 \land \text{pos}(\xi, \text{char}, -1) = 'r'$</td>
</tr>
<tr>
<td>sent01</td>
<td>Please generate a sentence with exactly 82 characters. Include whitespace into your character count.</td>
<td>$\text{count}(\xi, \text{char}, \text{sentence}) = 82$</td>
</tr>
<tr>
<td>sent02</td>
<td>Generate a sentence with 10 words, where word 3 is &quot;soft&quot; and word 7 is &quot;beach&quot; and word 10 is &quot;math&quot;.</td>
<td>$\text{count}(\xi, \text{word}, \text{sentence}) = 10 \land \text{pos}(\xi, \text{word}, 3) = \text{&quot;soft&quot;} \land \text{pos}(\xi, \text{word}, 7) = \text{&quot;beach&quot;} \land \text{pos}(\xi, \text{word}, 10) = \text{&quot;math&quot;}$</td>
</tr>
<tr>
<td>sent03</td>
<td>Generate a sentence with at least 20 words, and each word less than six characters.</td>
<td>$\text{count}(\xi, \text{word}, \text{sentence}) \geq 20 \land \text{count}(\xi, \text{char}, \text{word}) \leq 6$</td>
</tr>
<tr>
<td>sent04</td>
<td>Generate a sentence but be sure to include the words &quot;soft&quot;, &quot;beach&quot; and &quot;math&quot;.</td>
<td>$\text{count}(\xi, \text{word}, \text{sentence}) \geq 20 \land \text{count}(\xi, \text{word, &quot;soft&quot;)} &gt; 0 \land \text{count}(\xi, \text{word, &quot;beach&quot;)} &gt; 0 \land \text{count}(\xi, \text{word, &quot;math&quot;)} &gt; 0$</td>
</tr>
<tr>
<td>para01</td>
<td>Generate a paragraph where each sentence begins with the word &quot;soft&quot;.</td>
<td>$\text{pos}(\text{pos}(\xi, \text{sentence, 1}), \text{word, 1}) = \text{&quot;soft&quot;) \land \text{pos}(\text{pos}(\xi, \text{sentence, 2}), \text{word, 1}) = \text{&quot;soft&quot;) \land ...}$</td>
</tr>
<tr>
<td>para02</td>
<td>Generate a paragraph with at least 4 sentences, but do not use the words &quot;the&quot;, &quot;and&quot; or &quot;of&quot;.</td>
<td>$\text{count}(\xi, \text{sentence, paragraph}) \geq 4 \land \text{count}(\xi, \text{word, &quot;the&quot;)} = 0 \land \text{count}(\xi, \text{word, &quot;and&quot;)} = 0 \land \text{count}(\xi, \text{word, &quot;of&quot;)} = 0$</td>
</tr>
<tr>
<td>para03</td>
<td>Generate a paragraph with exactly 4 sentences, each with between 10 and 15 words.</td>
<td>$\text{count}(\xi, \text{sentence, paragraph}) = 4 \land \text{count}(\xi, \text{word, sentence}) \geq 10 \land \text{count}(\xi, \text{word, sentence}) \leq 15$</td>
</tr>
<tr>
<td>para04</td>
<td>Generate a paragraph with at least 3 sentences, each with at least 15 words.</td>
<td>$\text{count}(\xi, \text{sentence, paragraph}) \geq 3 \land \text{count}(\xi, \text{word, sentence}) \geq 15$</td>
</tr>
<tr>
<td>para05</td>
<td>Generate a paragraph with 2 sentences that end in &quot;math&quot; and &quot;rock&quot; respectively.</td>
<td>$\text{count}(\xi, \text{sentence, paragraph}) = 2 \land \text{pos}(\text{pos}(\xi, \text{sentence, 1}), \text{word, -1}) = \text{&quot;math&quot;) \land \text{pos}(\text{pos}(\xi, \text{sentence, 2}), \text{word, -1}) = \text{&quot;rock&quot;)}$</td>
</tr>
<tr>
<td>pass01</td>
<td>Generate a passage with 2 paragraphs, each ending in &quot;I sit,&quot; and &quot;I cry,&quot; respectively.</td>
<td>$\text{count}(\xi, \text{paragraph, passage}) = 2 \land \text{pos}(\text{pos}(\xi, \text{paragraph, 1}), \text{sentence, -1}) = \text{&quot;I sit&quot;) \land \text{pos}(\text{pos}(\xi, \text{paragraph, 2}), \text{sentence, -1}) = \text{&quot;I cry&quot;)}$</td>
</tr>
</tbody>
</table>

### Zero-shot performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-4</td>
<td>97.2%</td>
</tr>
<tr>
<td>GPT-3.5</td>
<td>96.7%</td>
</tr>
<tr>
<td>PaLM</td>
<td>48.5%</td>
</tr>
<tr>
<td>Vicuna-7B</td>
<td>49.6%</td>
</tr>
<tr>
<td>Alpaca-7B</td>
<td>49.9%</td>
</tr>
</tbody>
</table>

### Constraint satisfaction rate - GPT-4

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>word01</td>
<td>0.9%</td>
</tr>
<tr>
<td>word02</td>
<td>9.5%</td>
</tr>
<tr>
<td>word03</td>
<td>0.4%</td>
</tr>
<tr>
<td>sent01</td>
<td>48.5%</td>
</tr>
<tr>
<td>sent02</td>
<td>49.6%</td>
</tr>
<tr>
<td>sent03</td>
<td>49.9%</td>
</tr>
<tr>
<td>sent04</td>
<td>95.1%</td>
</tr>
<tr>
<td>para01</td>
<td>96.7%</td>
</tr>
<tr>
<td>para02</td>
<td>93.3%</td>
</tr>
<tr>
<td>para03</td>
<td>38.5%</td>
</tr>
<tr>
<td>para04</td>
<td>49.6%</td>
</tr>
<tr>
<td>para05</td>
<td>49.9%</td>
</tr>
<tr>
<td>pass01</td>
<td>41.6%</td>
</tr>
</tbody>
</table>
Task Construction is Fully Automatic

- Human just specify constraint types
- Collie automatically extracts constraint “values” from corpora
- Rule-based instruction rendering and text evaluation (extensible)
Feedback helps!

Many other cool findings in the paper!

- Generate a word with 10 letters, where letter 1 is "s", letter 3 is "r", letter 9 is "e"

- Sure, here's a word that meets your criteria: "strenuous".

- Your word only has 9 letters, and letter 9 is "s", not "e".

- I apologize for the mistake. Here's a correct word that meets your criteria: "strategies".
Internal mechanisms:
Leverage language and reasoning, go beyond RL

External environments:
Digital domains, scalable and faithful evaluation

1. ReAct: reasoning
2. Reflexion: learning
3. ToT: planning

1. WebShop: web
2. InterCode: code
3. Collie: logic