Tree of Thoughts: Deliberate Problem Solving with Large Language Models

Overview

1. **Input-Output (IO) Prompting:** $y \sim p_\theta^{\text{IO}}(y|x)$
2. **Chain-of-Thought(CoT) Prompting:** introduce a chain of thought $z_1, \cdots, z_n$ where each thought z_i is sequentially sampled $z_i \sim p_\theta^{\text{CoT}}(y|x, z_{1\cdots i-1})$ to serve as a meaningful intermediate step to reach $y \sim p_\theta^{\text{CoT}}(y|x, z_{1\cdots n})$.
3. **Self-consistency with CoT (CoT-SC):** k i.i.d. samples $[z_{1\cdots n}^{(i)}, y^{(i)}] \sim p_\theta^{\text{CoT}}(z_{1\cdots n}, y|x)$, and return most frequent output $\arg\max_y \#\{i|y^{(i)} = y\}$.

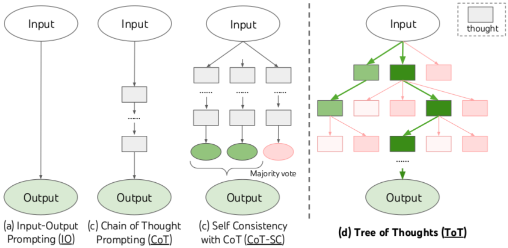

Figure 1: Illustration of various approaches in problem solving

Tree of Thoughts (ToT)

ToT frames a problem as a search over tree, where each node of a tree is a state $s = [x, z_{1 \cdots i}]$ or partial solution to the problem.

Tree of Thoughts (ToT)

1. Thought decomposition

Unlike CoT, which sequentially sample thoughts without explicit decomposition, ToT design problem-specific decomposition.

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

Figure 2: Task overview

Tree of Thoughts (ToT)

2. **Thought generator** $G(p_\theta, s, k)$
   - Sample: k i.i.d. samples $z^{(j)} \sim p_\theta^{\text{CoT}}(z_{i+1}|s)$
     - Good when search space is rich
     - ex. Creative writing
   - Propose: generate thougths sequentially using "propose prompt"
     - Good when search space is constrained
     - ex. Crosswords, Game of 24

Tree of Thoughts (ToT)

3. State evaluator $V(p_\theta, S)$
Use LLM p_θ (mulitple times) to reason about the state s

- Value: $V(p_\theta, S)(s) \sim p_\theta^{\text{value}}(v|s)$
  - score : scalar value or classification (ex. sure/likely/impossible)
  - few lookahead simulation and commonsense
- Vote: $V(p_\theta, S)(s) = \mathbb{1}[s = s^\star]$, where "good" state $s^\star = p_\theta^{\text{vote}}(s^\star|S)$
  - Use when direct evaluation is hard
  - similar to multi-step self-consistency strategy

Tree of Thoughts (ToT)

4. **Search algorithm**
    1. Breadth-first search
    2. Depth-first search

Tree Search Algorithms

Breadth-First Search (BFS)

```
def BFS(G: graph, V0: root, Vt: target):
    Q = Queue()
    Q.append(V0)
    V0.visited = True
    while len(Q) != 0:
        V = Q.dequeue()
        V.visitied = True
        if V == Vt:
            return
        else:
            for v in V.children():
                if not v.visited:
                    Q.enqueue(v)
```

Tree Search Algorithms

Depth-First Search (DFS)

```python
def DFS(G: graph, V: root, Vt: target):
    V.visited = True
    if V == Vt:
        return
    else:
        for v in V.children():
            if not v.visited:
                DFS(G, v, Vt)
```

Tree of Thoughts (ToT)

Algorithm 1 ToT-BFS

Given:

  Input x, LLM p_θ, thought generator G, size limit k, state evaluator V, step limit T, breadth limit b

$S_0 \leftarrow \{x\}$

for $t = 0, 1, \cdots, T$ **do**

  $S'_t \leftarrow \{[s, z] | s \in S_{t-1}, z_t \in G(p_\theta, s, k)\}$ $\triangleright$ $k * b$ candidates

  $V_t \leftarrow V(p_\theta, S'_t)$

  $S_t \leftarrow \arg\max_{S \subset S'_t, |S| = b} \sum_{s \in S} V_t(s)$ $\triangleright$ b candidates

end for

return $G(p_\theta, \arg\max_{s \in S_T} V_T(s), 1)$

Tree of Thoughts (ToT)

Algorithm 2 ToT-DFS

1: **Given:**
Current state s, step t, LLM p_θ, thought generator G, size limit k, state evaluator V, step limit T, threshold v_{th}
2: **if** $t > T$ **then** record $G(p_\theta, s, 1)$
3: **end if**
4: **for** $s' \in G(p_\theta, s, k)$ **do**
5: **if** $V_\theta, \{s'\})(s) > v_{\text{th}}$ **then** DFS(s', t+1) ▷ check plausibility
6: **end if**
7: **end for**

Tree of Thoughts (ToT)

Benefits

- Generality: previous methods are special case of ToT
- Modularity: each modularized compartment (thought decomp, thought gen, state eval, search alg) can be modified individually
- Adaptability: different problem settings, LMs, resource constraint can be used
- Convenience: no extra training

Experiments

Game of 24: use given 4 numbers to obtain 24 with basic arithmetic operations (ex. $(10-4)*(13-9)=24$)

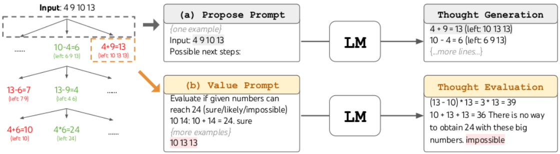

Figure 3: Game of 24

- Thought decomposition: 3 steps
- Thought generator: "propose" prompt
- State evaluator: LLM evaluate each thought as sure/maybe/impossible
- Search algorithm: BFS

Experiments

Game of 24

Method	Success
IO prompt	7.3%
CoT prompt	4.0%
CoT-SC (k=100)	9.0%
ToT (ours) (b=1)	45%
ToT (ours) (b=5)	**74%**
IO + Refine (k=10)	27%
IO (best of 100)	33%
CoT (best of 100)	49%

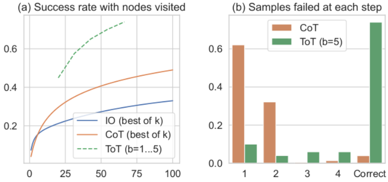

Figure 4: Game of 24 results

Experiments

Creative writing: write 4 short paragraphs where each ends with one of the given 4 sentences.

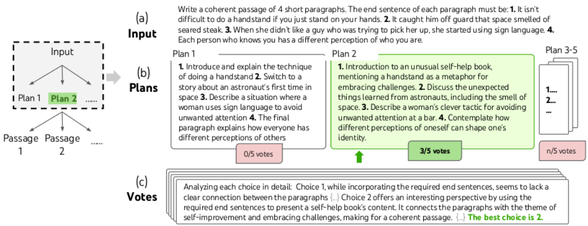

Figure 5: Creative writing

- Thought decomposition: 2 steps
- Thought generator: sample k plans
- State evaluator: LLM voting
- Search algorithm: BFS

Experiments

Creative writing

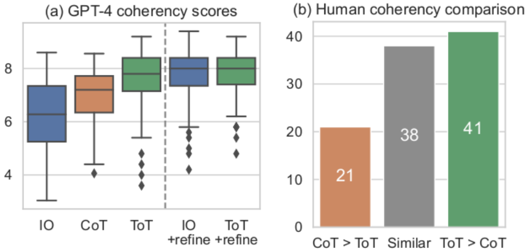

Figure 6: Creative writing results

Evaluation

- GPT-4: make GPT-4 to give 5 independent 1$\sim$10 scalar scores, and report average
- Human: empoly authors to judge between two outputs

Refine: iteratively ask LLM to refine the passage if it is not perfectly coherent.

Experiments

Mini crosswords: 5×5 mini crosswords

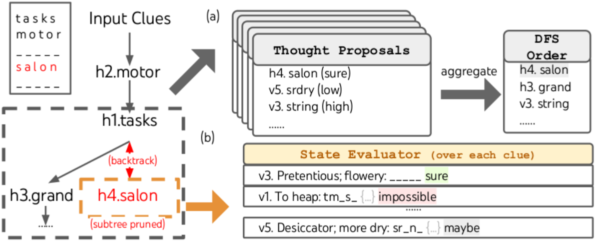

Figure 7: Mini crosswords

- Thought decomposition: at most 10 steps
- Thought generator: "propose" prompt to come up with candidates and its confidence level
- State evaluator: LLM evaluate whether each proposal or not
- Search algorithm: DFS

Experiments

Mini crosswords

Method	Success Rate (%)		
	Letter	Word	Game
IO	38.7	14	0
CoT	40.6	15.6	1
ToT (ours)	**78**	**60**	**20**
+best state	82.4	67.5	35
-prune	65.4	41.5	5
-backtrack	54.6	20	5

Figure 8: Mini crosswords results

Thank You

Q & A