NLP CAFÉ @ CTS

21 Sept 2023

Constantin Orasan

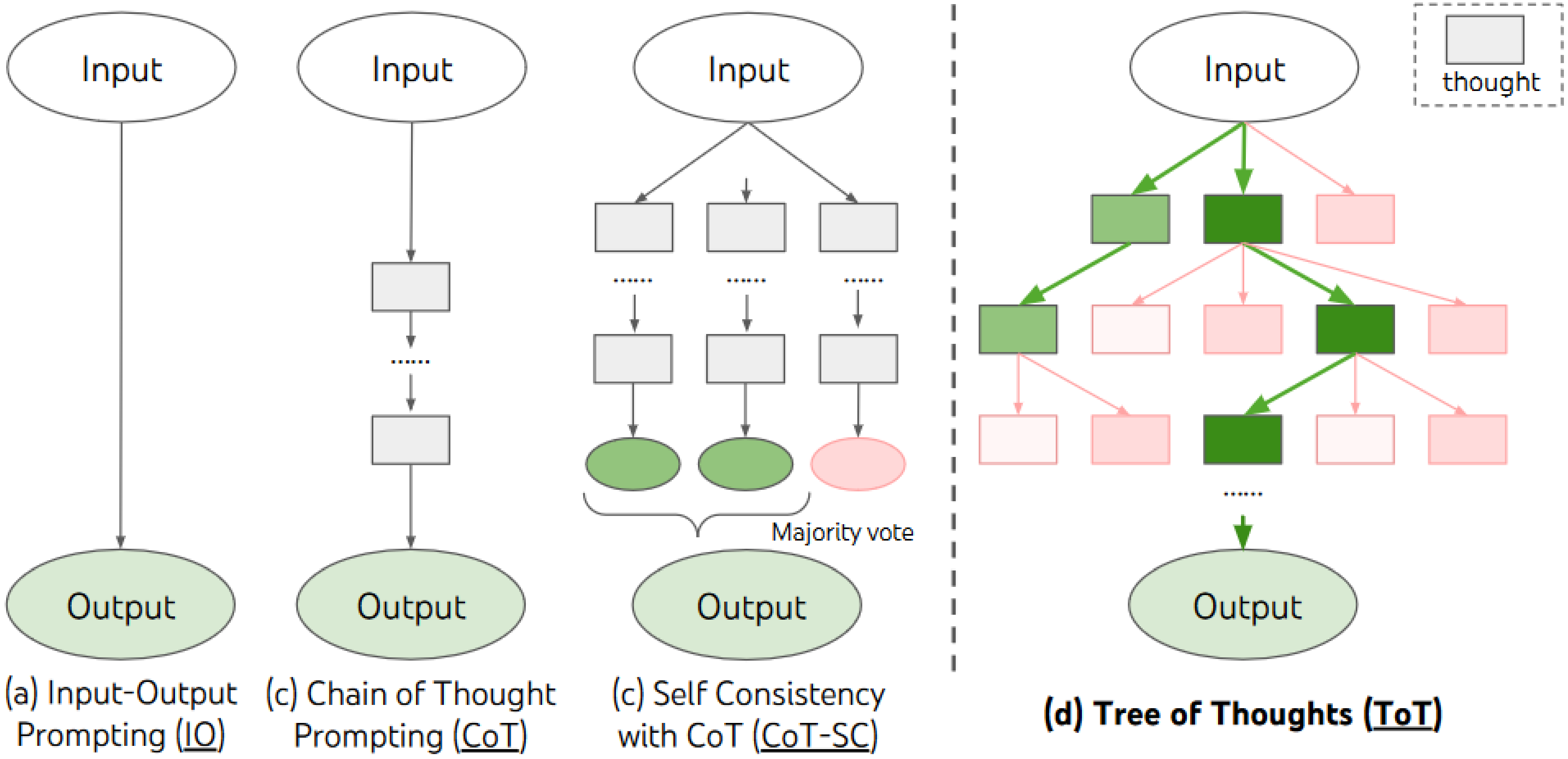
https://dinel.org.uk

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

https://arxiv.org/abs/2305.10601

WHO HASN'T HEARD OF LLMs?



Thought = "a coherent language sequence that serves as an intermediate step towards problem solving"

WHY?

- Tries to combat the usual "token-level decisions one by one and in a left-to-right fashion."
- Justification in human cognition: two modes of taking decisions
  - fast automatic unconscious mode
  - slow deliberate conscious mode

- Thought are seen as high-level semantic units that allow the LLM to self-evaluate the progress towards a solution
- Search heuristics via the LLM, rather than hard coded/learnt from data

NICE FORMALISATION OF PROMPTING

We first formalize some existing methods that use large language models for problem-solving, which our approach is inspired by and later compared with. We use p_θ to denote a pre-trained LM with parameters θ, and **lowercase letters** $x, y, z, s, \cdots$ **to denote a language sequence**, i.e. $x = (x[1], \cdots, x[n])$ where each $x[i]$ is a token, so that $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x[i]|x[1...i])$. We use uppercase letters $S, \cdots$ to denote a collection of language sequences.

Input-output (IO) prompting is the most common way to turn a problem input x into output y with LM: $y \sim p_\theta(y|\texttt{prompt}_{IO}(x))$, where $\texttt{prompt}_{IO}(x)$ wraps input x with task instructions and/or few-shot input-output examples. For simplicity, let us denote $p_\theta^{\text{prompt}}(\texttt{output} \mid \texttt{input}) = p_\theta(\texttt{output} \mid \texttt{prompt}(\texttt{input}))$, so that IO prompting can be formulated as $y \sim p_\theta^{IO}(y|x)$.

NICE FORMALISATION OF PROMPTING

We first formalize some existing methods that use large language models for problem-solving, which our approach is inspired by and later compared with. We use p_θ to denote a pre-trained LM with parameters θ, and **lowercase letters** $x, y, z, s, \cdots$ **to denote a language sequence**, i.e. $x = (x[1], \cdots, x[n])$ where each $x[i]$ is a token, so that $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x[i]|x[1...i])$. We use uppercase letters $S, \cdots$ to denote a collection of language sequences.

Chain-of-thought (CoT) prompting [35] was proposed to address cases where the mapping of input x to output y is non-trivial (e.g. when x is a math question and y is the final numerical answer). The key idea is to introduce a chain of *thoughts* $z_1, \cdots, z_n$ to bridge x and y, where each z_i is a coherent language sequence that serves as a meaningful intermediate step toward problem solving (e.g. z_i could be an intermediate equation for math QA). To solve problems with CoT, each thought $z_i \sim p_\theta^{CoT}(z_i \mid x, z_{1\cdots i-1})$ is sampled sequentially, then the output $y \sim p_\theta^{CoT}(y|x, z_{1\cdots n})$. In practice, $[z_{1\cdots n}, y] \sim p_\theta^{CoT}(z_{1\cdots n}, y|x)$ is sampled as a continuous language sequence, and the **decomposition** of thoughts (e.g. is each z_i a phrase, a sentence, or a paragraph) is left ambiguous.

NICE FORMALISATION OF PROMPTING

We first formalize some existing methods that use large language models for problem-solving, which our approach is inspired by and later compared with. We use p_θ to denote a pre-trained LM with parameters θ, and **lowercase letters** $x, y, z, s, \cdots$ **to denote a language sequence**, i.e. $x = (x[1], \cdots, x[n])$ where each $x[i]$ is a token, so that $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x[i]|x[1...i])$. We use uppercase letters $S, \cdots$ to denote a collection of language sequences.

Self-consistency with CoT (CoT-SC) [33] is an ensemble approach that samples k i.i.d. chains of thought: $[z_{1\cdots n}^{(i)}, y^{(i)}] \sim p_\theta^{CoT}(z_{1\cdots n}, y|x)$ $(i = 1 \cdots k)$, then returns the most frequent output: $\arg\max_y \#\{i \mid y^{(i)} = y\}$. CoT-SC improves upon CoT, because there are generally different thought processes for the same problem (e.g. different ways to prove the same theorem), and the output decision can be more faithful by exploring a richer set of thoughts. However, within each chain there is no local exploration of different thought steps, and the "most frequent" heuristic only applies when the output space is limited (e.g. multi-choice QA).

TREE OF THOUGHTS (ToT)

- Human problem-solving: people search through a combinatorial problem-space – a tree where nodes represent partial solutions and branches correspond to opeators that modify the solutions
- Existing uses of LLMs do not
  - Explore alternative solutions (branches)
  - Incorporate any type of planning, lookahead or backtracking

A specific instantiation of ToT specifies:

1. How to **decompose** the intermediate process into thought steps;
2. How to **generate** potential thoughts from each state;
3. How to heuristically **evaluate** states;
4. What **search** algorithm to use

TREE OF THOUGHTS (ToT)

1. **Thought decomposition**: explicitly requires to decompose the problem into intermediate thought steps

A thought should be "small" enough so that LLMs can generate promising and diverse samples, yet big enough so that it can be evaluated towards problem solving

	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)

Table 1: Task overview. Input, output, thought examples are in blue.

TREE OF THOUGHTS (ToT)

2. **Thought generator**: generate k candidates for the next thought step
   - Sample: if the space is rich (generate some text)
   - Propose: what's next in the sequence given some constraints

3. **State evaluator**: assesses whether there is progress towards the solution with the help of the LLM (no rules or ML)
   - **Value** each state independently: give a score to each state (0-10, sure/likely/impossible), promote good states, eliminate bad states
   - **Vote** across states: compare different solutions and vote for the most promising one

4. **Search algorithm**: how to explore the search space
   - **Breadth-first search**
   - **Depth-first search**

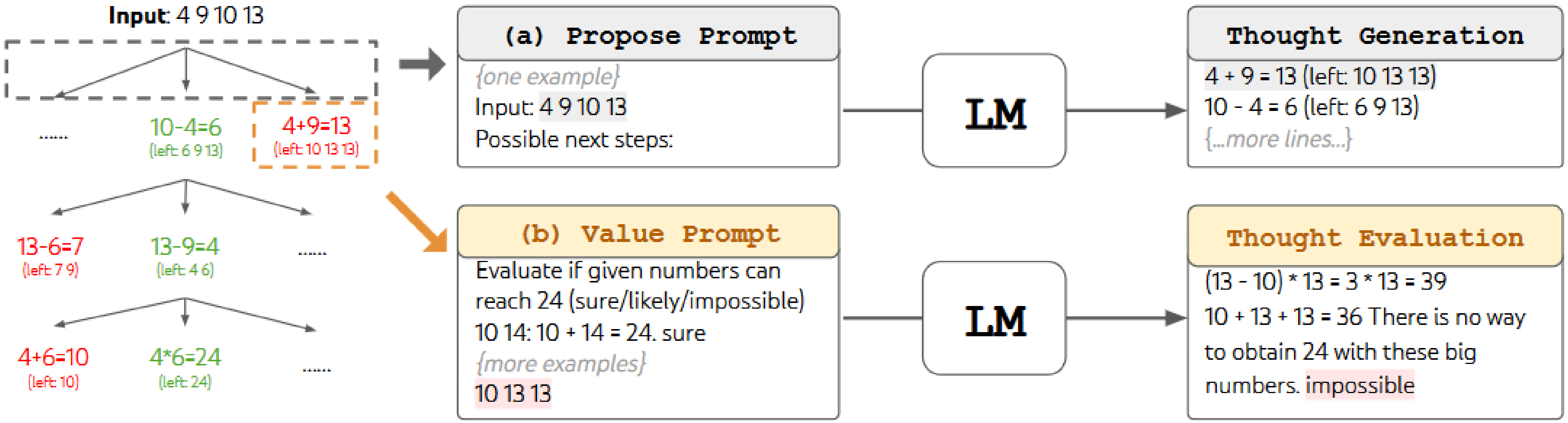


Figure 2: ToT in a game of 24. The LM is prompted for (a) thought generation and (b) valuation.

EXPERIMENTS

PROMPTS

```
# 5-shot
standard_prompt = '''Use numbers and basic arithmetic operations (+ - * /) to obtain 24.
Input: 4 4 6 8
Answer: (4 + 8) * (6 - 4) = 24
Input: 2 9 10 12
Answer: 2 * 12 * (10 - 9) = 24
Input: 4 9 10 13
Answer: (13 - 9) * (10 - 4) = 24
Input: 1 4 8 8
Answer: (8 / 4 + 1) * 8 = 24
Input: 5 5 5 9
Answer: 5 + 5 + 5 + 9 = 24
Input: {input}
'''
```

```
cot_prompt = '''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:

12 * 2 = 24 (left: 9 10 24)

10 - 9 = 1 (left: 1 24)

24 * 1 = 24 (left: 24)

Answer: (12 * 2) * (10 - 9) = 24

Input: 4 9 10 13

Steps:

13 - 10 = 3 (left: 3 4 9)

9 - 3 = 6 (left: 4 6)

4 * 6 = 24 (left: 24)

Answer: 4 * (9 - (13 - 10)) = 24

Input: 1 4 8 8

Steps:

8 / 4 = 2 (left: 1 2 8)

1 + 2 = 3 (left: 3 8)

3 * 8 = 24 (left: 24)

Answer: (1 + 8 / 4) * 8 = 24

Input: 5 5 5 9

Steps:

5 + 5 = 10 (left: 5 9 10)

10 + 5 = 15 (left: 9 15)

15 + 9 = 24 (left: 24)

Answer: ((5 + 5) + 5) + 9 = 24

Input: {input}
'''
```

```
propose_prompt = '''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 * 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: {input}
Possible next steps:
'''
```

```
value_prompt = '''Evaluate if given numbers can reach 24 (sure/likely/impossible)

10 14

10 + 14 = 24

sure

11 12

11 + 12 = 23

12 - 11 = 1

11 * 12 = 132

11 / 12 = 0.91

impossible

4 4 10

4 + 4 + 10 = 8 + 10 = 18

4 * 10 - 4 = 40 - 4 = 36

(10 - 4) * 4 = 6 * 4 = 24

sure

4 9 11

9 + 11 + 4 = 20 + 4 = 24

sure

5 7 8

5 + 7 + 8 = 12 + 8 = 20

(8 - 5) * 7 = 3 * 7 = 21

I cannot obtain 24 now, but numbers are within a reasonable range

likely

5 6 6

5 + 6 + 6 = 17

(6 - 5) * 6 = 1 * 6 = 6

I cannot obtain 24 now, but numbers are within a reasonable range

likely

10 10 11

10 + 10 + 11 = 31

(11 - 10) * 10 = 10

10 10 10 are all too big

impossible

1 3 3

1 * 3 * 3 = 9

(1 + 3) * 3 = 12

1 3 3 are all too small

impossible

{input}

'''
```

```
value_last_step_prompt = '''Use numbers and basic arithmetic operations (+ - * /) to obtain 24. Given an input and an answer, give a judgement (sure/impossible) if the answer is correct, i.e. it uses each input exactly once and no other numbers, and reach 24.

Input: 4 4 6 8

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

Judge:

sure

Input: 2 9 10 12

Answer: 2 * 12 * (10 - 9) = 24

Judge:

sure

Input: 4 9 10 13

Answer: (13 - 9) * (10 - 4) = 24

Judge:

sure

Input: 4 4 6 8

Answer: (4 + 8) * (6 - 4) + 1 = 25

Judge:

impossible

Input: 2 9 10 12

Answer: 2 * (12 - 10) = 24

Judge:

impossible

Input: 4 9 10 13

Answer: (13 - 4) * (10 - 9) = 24

Judge:

impossible

Input: {input}

Answer: {answer}

Judge:'''
```

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%

Table 2: Game of 24 Results.

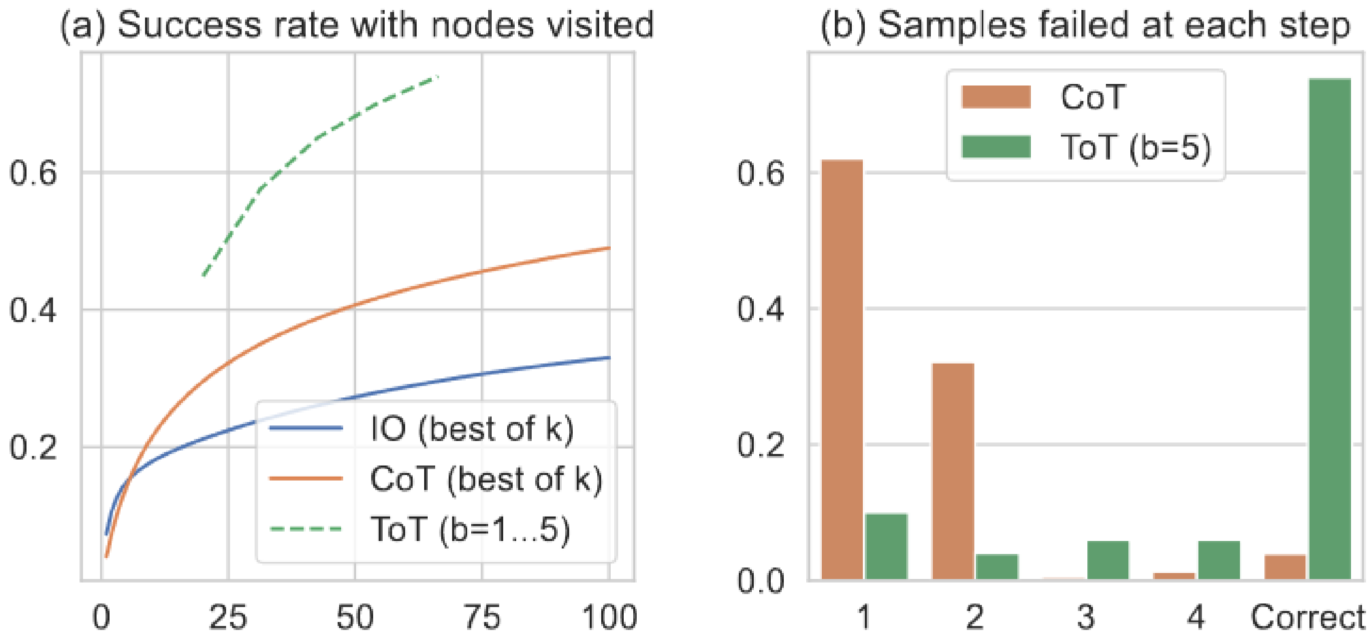


Figure 3: Game of 24 (a) scale analysis & (b) error analysis.

EVALUATION FOR GAME OF 24

CREATIVE TEXT WRITING

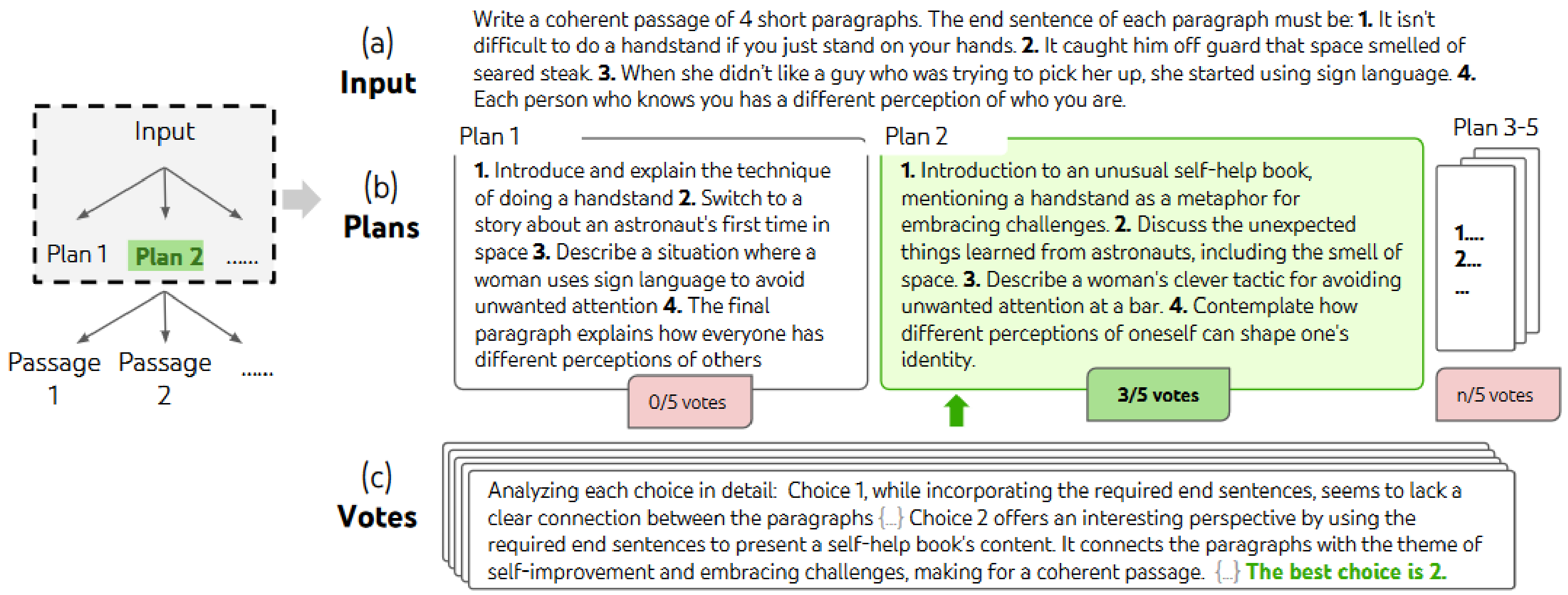


Figure 4: A step of deliberate search in a randomly picked Creative Writing task. Given the input, the LM samples 5 different plans, then votes 5 times to decide which plan is best. The majority choice is used to consequently write the output passage with the same sample-vote procedure.

```
standard_prompt = '''
Write a coherent passage of 4 short paragraphs. The end sentence of each paragraph must be: {input}
'''
```

```
cot_prompt = '''
Write a coherent passage of 4 short paragraphs. The end sentence of each paragraph must be: {input}

Make a plan then write. Your output should be of the following format:

Plan:
Your plan here.

Passage:
Your passage here.
'''
```

```
vote_prompt = '''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.
'''


compare_prompt = '''Briefly analyze the coherency of the
following two passages. Conclude in the last line "The more
coherent passage is 1", "The more coherent passage is 2", or
"The two passages are similarly coherent".
'''


score_prompt = '''Analyze the following passage, then at the
last line conclude "Thus the coherency score is {s}", where
s is an integer from 1 to 10.
'''
```

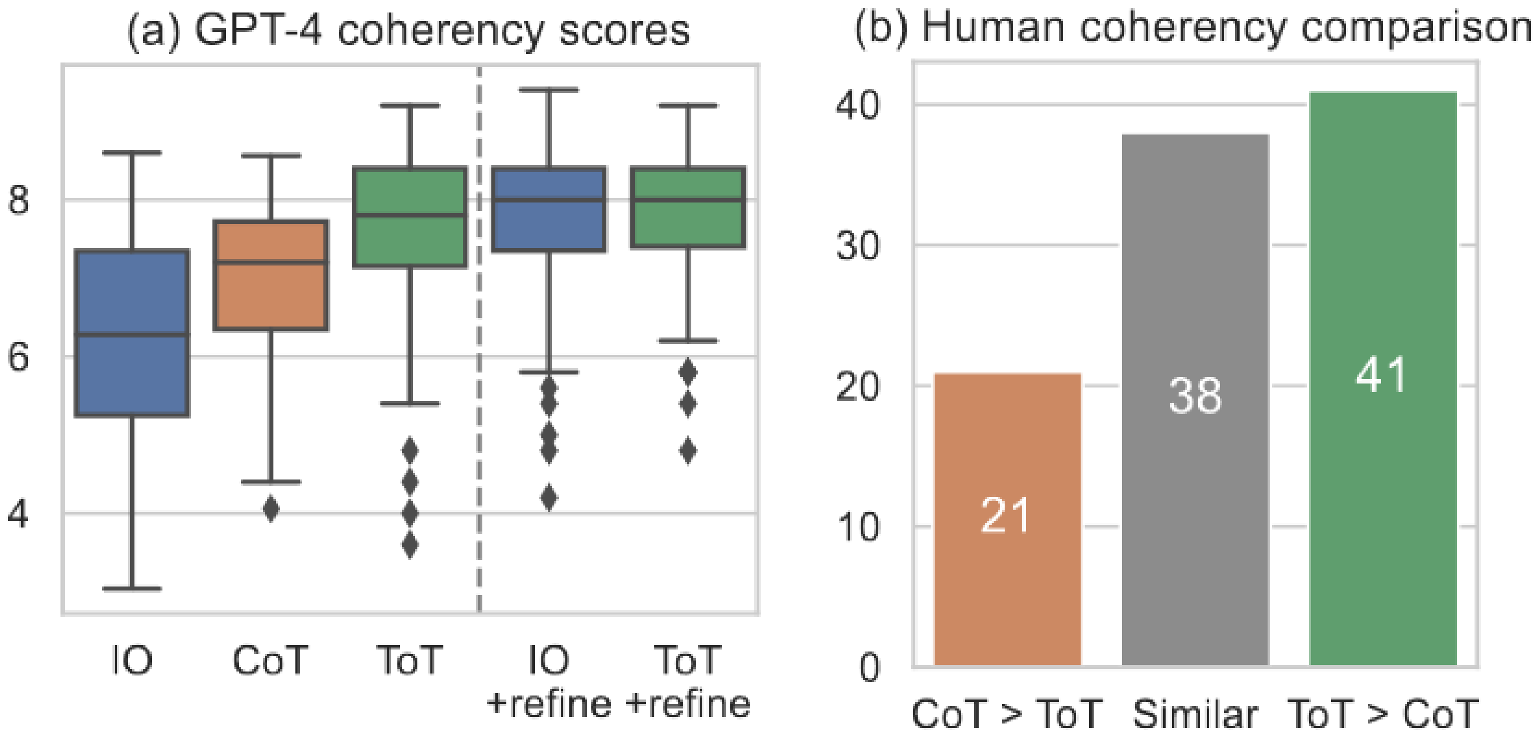


Figure 5: Creative Writing results.

EVALUATION OF CREATIVE WRITING

ToT "ON THE CHEAP"

Imagine three different experts are answering this question.

They will brainstorm the answer step by step reasoning carefully and taking all facts into consideration

All experts will write down 1 step of their thinking, then share it with the group.

They will each critique their response, and the all the responses of others

They will check their answer based on science and the laws of physics

Then all experts will go on to the next step and write down this step of their thinking.

They will keep going through steps until they reach their conclusion taking into account the thoughts of the other experts

If at any time they realise that there is a flaw in their logic they will backtrack to where that flaw occurred

If any expert realises they're wrong at any point then they acknowledges this and start another train of thought

Each expert will assign a likelihood of their current assertion being correct

Continue until the experts agree on the single most likely location

The question is ...

ToT "ON THE CHEAP"

1. Carlos is at the swimming pool.
2. He walks to the locker room, carrying a towel.
3. He puts his watch in the towel and carries the towel tightly to a lounger at the poolside.
4. At the lounger he opens and vigorously shakes the towel, then walks to the snack bar.
5. He leaves the towel at the snack bar, then walks to the diving board.
6. Later Carlos realises he has has lost his watch. Where is the single most likely location of the watch?

See: https://youtu.be/2lnW1PSB2_g?si=DmL6beEdU-0qQVdP

THOUGHTS/ToT?