



STaR: Self-Taught Reasoner

Bootstrapping Reasoning With Reasoning

Eric Zelikman, Yuhuai Wu, Jesse Mu, Noah D. Goodman



Why did the machine learning model go to therapy?



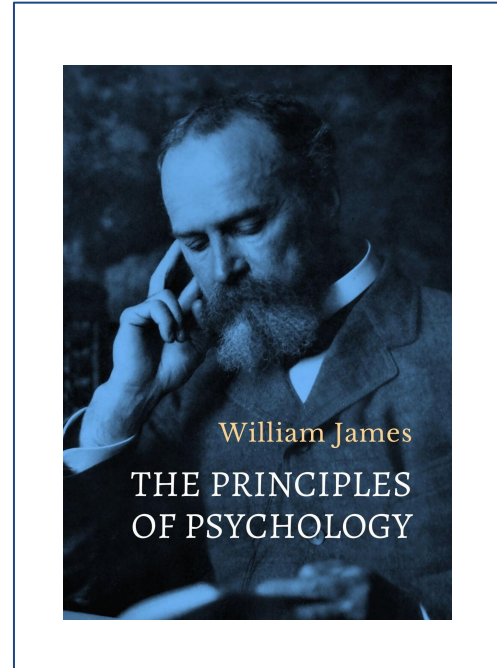
What are LLMs missing?

- Operates through chain of associations
- Gives an answer without considering or breaking it down into sub-problems.



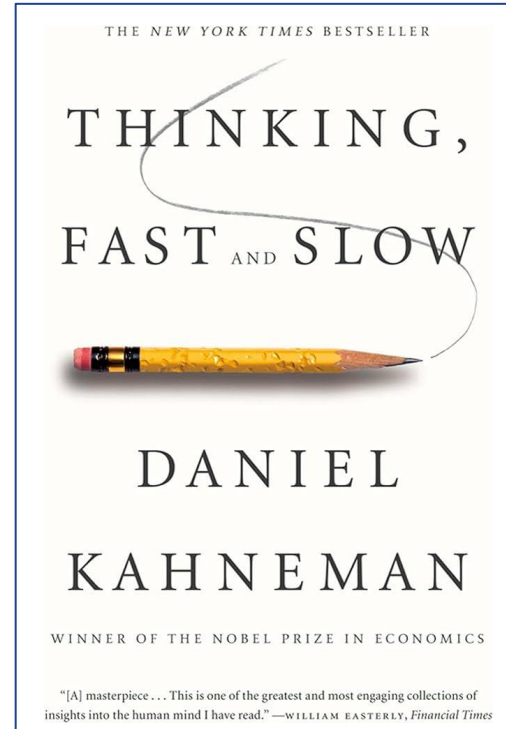
Introduction

- Human thought can be characterized as a flowing stream such that human thought was more so like a distinct chain.
- Human decision making is often made by extended chains of thought.
- Subproblems are processed in detail with multiple cycles in the brain before assembling an answer together.



System 2 Thinking

- Reasoning
- Slow
- Considerable cognitive resources



Rationale in LLMs

The image displays a collage of research papers and their abstracts, illustrating the concept of 'Rationale in LLMs'. The papers are arranged in a grid-like fashion, with some overlapping. The papers shown include:

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models** (arXiv:2201.11903v6 [cs.CL] 10 Jan 2023) by Jesse Wei, Xuezhi Wang, Dale Schuurman, Brian Ichter, Fu Xia, Ed H. Chi, Quoc V. Le. Abstract: We explore how generating a chain of thought—a series of steps—significantly improves the ability of large language models to solve complex reasoning tasks. In particular, we show how such models are able to solve complex reasoning tasks in a way that is similar to human reasoning, where a chain of thought denotes a sequence of steps. Experiments on three large language models show that chain-of-thought prompting significantly improves performance on a range of arithmetic, commonsense reasoning, and symbolic tasks. The empirical gains can be striking. For example, on the GSM8K benchmark of math word problems, chain-of-thought prompting improves performance by 100%.
- Tree of Thoughts: Decomposing and Searching Problem Spaces with Large Language Models** (arXiv:2305.10601v2 [cs.CL] 3 Dec 2023) by Shunyu Yao, Dian Yu, Jeffrey Zhao, Yanning Shen, Yanping Huang, Rui Xue, Fei Huang, Chong Wang, and Yuan Cao. Abstract: Large language models are increasingly being used to solve a wide range of tasks, but their decision-making processes during such tasks that require exploration, strategic planning, and search are often opaque. We propose a novel paradigm, "Tree of Thoughts" (ToT), which decomposes a problem into a search space of intermediate thoughts and explores them iteratively. ToT allows for exploring multiple different paths to solve a problem, and it enables the model to backtrack and refine its thoughts. We evaluate ToT on a variety of tasks, including arithmetic, commonsense reasoning, and symbolic tasks. For instance, on Game of 24, while GPT-4 solves 49% of tasks, our method achieves 100%.
- SHOW YOUR WORK: SCRATCHPADS FOR INTERMEDIATE COMPUTATION WITH LANGUAGE MODELS** (arXiv:2312.00114v1 [cs.LG] 30 Nov 2023) by Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henry Michalewski, Jacob Austin, David Bieber, David Dohan, Alireza Leshkevycz, Maarten Bosma, David Lian, Charles Sutton, and Augustus Odena. Abstract: Large pre-trained language models perform remarkably well on tasks that can be done "in one pass", such as generating text, but struggle on tasks that require intermediate computation with language models. We propose a new paradigm, "Scratchpads", which allows models to perform multi-step computations—even in the few-shot regime—when asked to perform the operation "step by step", showing the results of intermediate computation. In particular, we train Transformers to perform multi-step computations by asking them to emit intermediate computation, step (i) in a "scratchpad". On a suite of 16 reasoning tasks ranging from logic puzzles to the evaluation of arbitrary programs, we show that scratchpads dramatically improve the ability of language models to perform multi-step computations.

37th Conference on Neural Information Processing

Rationale generations and rationalization

Q: What are people in a library likely doing?

Answer Choices:

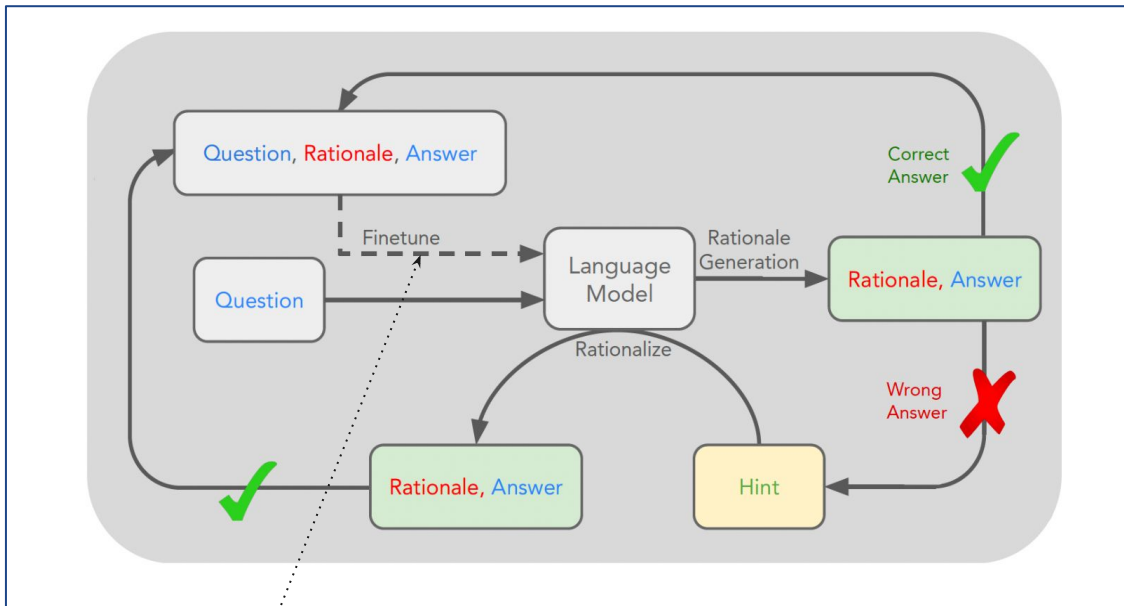
- (a) talk to each other
- (b) board ships
- (c) study books
- (d) suffer hunger
- (e) playing games

A: The answer must be something that people in a library are likely to be doing. People in a library are likely to be studying books. Therefore, the answer is study books (c).

Rationale/Reasoning



What is STaR Method?



Outer-loop Fine-tuning

The **questions** and **ground truth answers** are expected to be present in the dataset, while the **rationales** are generated using STaR.

- LLM generates a rationale based on the question.
- Q, R, A structure: **Question** (Q), **Rationale** (R), **Answer** (A) is stored in a database.
- If the rationale is correct, it's added directly to the database.
- If incorrect, post-rationalization occurs:
 - The model is provided with the correct answer.
 - It then generates a reasoning that would logically lead to this answer.
- This newly generated rationale is added to the Q, R, A database.

STaR Algorithm

Input, Output

Given some rationale examples
(how we get from x to y)

Algorithm 1 STaR

Input M : a pretrained LLM; dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^D$ (w/ few-shot prompts)

- 1: $M_0 \leftarrow M$ # Copy the original model
 - 2: **for** n **in** $1 \dots N$ **do** # Outer loop
 - 3: $(\hat{r}_i, \hat{y}_i) \leftarrow M_{n-1}(x_i) \quad \forall i \in [1, D]$ # Perform rationale generation
 - 4: $(\hat{r}_i^{\text{rat}}, \hat{y}_i^{\text{rat}}) \leftarrow M_{n-1}(\text{add_hint}(x_i, y_i)) \quad \forall i \in [1, D]$ # Perform rationalization
 - 5: $\mathcal{D}_n \leftarrow \{(x_i, \hat{r}_i, y_i) \mid i \in [1, D] \wedge \hat{y}_i = y_i\}$ # Filter rationales using ground truth answers
 - 6: $\mathcal{D}_n^{\text{rat}} \leftarrow \{(x_i, \hat{r}_i^{\text{rat}}, y_i) \mid i \in [1, D] \wedge \hat{y}_i \neq y_i \wedge \hat{y}_i^{\text{rat}} = y_i\}$ # Filter rationalized rationales
 - 7: $M_n \leftarrow \text{train}(M, \mathcal{D}_n \cup \mathcal{D}_n^{\text{rat}})$ # Finetune the original model on correct solutions - inner loop
 - 8: **end for**
-

**parts in blue corresponding to rationalization.

**GPT-J – base language model

Methodology

◆ Rationale Generation Bootstrapping (STaR Without Rationalization)

- We are given a pre trained LLM M and an initial **dataset** of problems x (including answer choices if applicable) with correct final answers y .
- The technique starts with a small prompt set P of examples with intermediate rationales r . Like standard few-shot prompting, we concatenate this prompt set to each example in D which encourages the model to produce a rationale for input followed by an answer.
- We filter the generated rationales to include only the ones which result in the correct answer.
- We fine-tune the base model on this filtered M dataset, and then restart this process by generating the new rationales with the newly fine-tuned model.
- STaR can be seen as an approximation to an RL-style policy gradient objective.

$$J(M, X, Y) = \sum_i \mathbb{E}_{\hat{r}_i, \hat{y}_i \sim p_M(\cdot | x_i)} \mathbb{1}(\hat{y}_i = y_i),$$
$$\nabla J(M, X, Y) = \sum_i \mathbb{E}_{\hat{r}_i, \hat{y}_i \sim p_M(\cdot | x_i)} [\mathbb{1}(\hat{y}_i = y_i) \cdot \nabla \log p_M(\hat{y}_i, \hat{r}_i | x_i)],$$

Methodology



Rationalization

- Apply rationalization to problems which the model failed to solve with rationale generation.
 - Provide the model with the correct answer as a hint.
 - Ask the model to generate rationales consistent with the previous rationale generation approach.
 - With the answer known, the model can reason backwards to create a rationale that aligns with the correct answer.

Q: Where do you put your grapes just before checking out?

Answer Choices:

- (a) mouth
- (b) grocery cart (CORRECT)
- (c) super market
- (d) fruit basket
- (e) fruit market

A: The answer should be the place where grocery items are placed before checking out. Of the above choices, grocery cart makes the most sense for holding grocery items. Therefore, the answer is grocery cart (b).

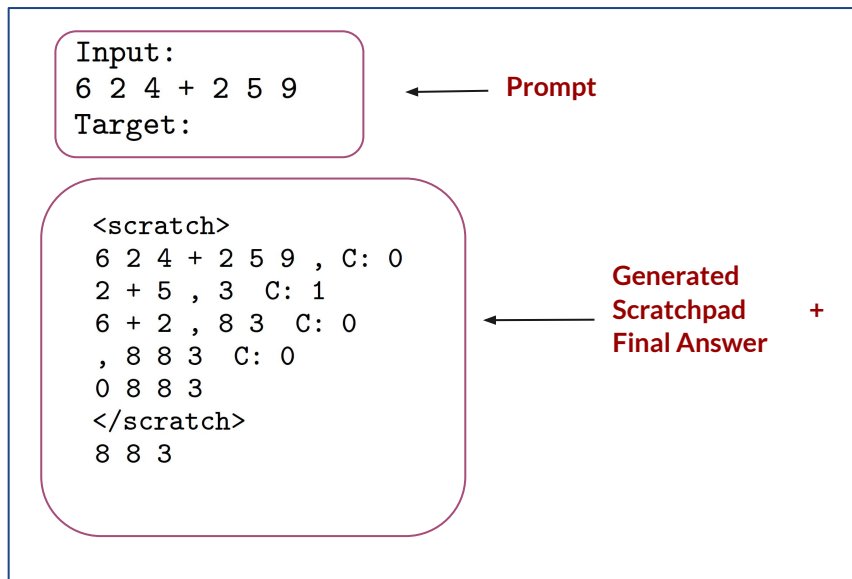
About STaR



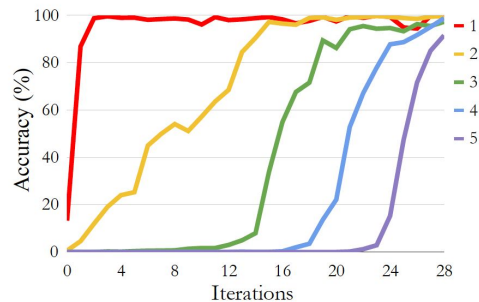
- ❖ This work suggests that **generating explicit rationales** before giving a final answer is valuable for LLMs across diverse tasks including *mathematical reasoning, commonsense reasoning, code evaluation, social bias inference, and natural language inference*.
- ❖ This is a **synergistic process**, *where improvements in rationale generation improve the training data, and improvements in training data further improve rationale generation*.

Datasets & Results

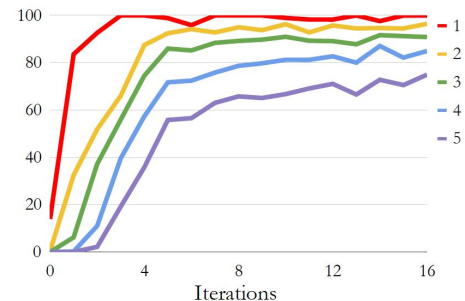
Arithmetic:



During rationalization, correct answer is included after Target.



(a) Without rationalization



(b) With rationalization

Accuracy of n-digit summation

Datasets & Results

CommonsenseQA:

- ★ The multiple-choice commonsense reasoning task.
- ★ CQA contains a diverse set of questions which require commonsense reasoning ability building off of standard world knowledge, where human performance is 89%.

	CQA Dev Set Accuracy (%)	Train Data Used (%)
<i>GPT-3 Direct Finetuned</i> [32]	73.0	100
Few-shot Direct GPT-J	20.9	~0
Few-shot CoT GPT-J ⁴	36.6	~0
Few-shot CoT LaMDA 137B [6]	55.6	~0
GPT-J Direct Finetuned	60.0	100
STaR without rationalization	68.8	69.7
STaR with rationalization	72.5	86.7

Datasets & Results

Grade School Math (GSM8K):

- ★ Grade-school-level word problems
- ★ Require two to eight calculation steps to arrive at a final answer.
- ★ For rationalization, the final answer is included in parentheses immediately after the question as a hint.

	GSM8K Test Accuracy (%)	Train Data Used (%)
Few-shot Direct GPT-J	3.0	~0
Few-shot CoT GPT-J	3.1	~0
GPT-J Direct Finetuned	5.8	100
STaR without rationalization	10.1	25.0
STaR with rationalization	10.7	30.3

Discussion



Why Rationalization?

- Allows a model to reverse-engineer a solution, or provides a heuristic for identifying whether each step makes the conclusion more likely.
- It increases the size of the dataset by adding rationales for previously incorrect answers.
- Introduces alternative reasoning paths by conditioning on the correct answer.

Challenges



- The model can generate correct answers with flawed or irrelevant reasoning.
- Pre-existing biases in the dataset could be amplified through rationalization.
- Generated rationales might not always faithfully represent the model's internal reasoning process.
- No guarantee that our results would generalize to larger models.



Why did the machine learning model go to therapy?

Because it had too many unresolved issues... but don't worry, with some self-taught reasoning, it's learning to work through them!





thank you!