Recursion of Thought:

A Divide and Conquer Approach to Multi-Context Reasoning with Language Models

Soochan Lee and Gunhee Kim

61 ACL 2023

Short Findings

SEOUL NATIONAL UNIV.
VISION & LEARNING

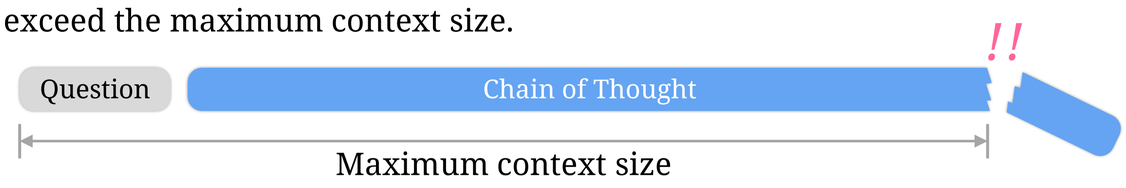
Chain of Thought and the Limited Context of Language Models

- Generating *Chain of Thought* (CoT) is crucial to solving complex reasoning problems.

Question | Answer ✗

Question | Chain of Thought | Answer ✓

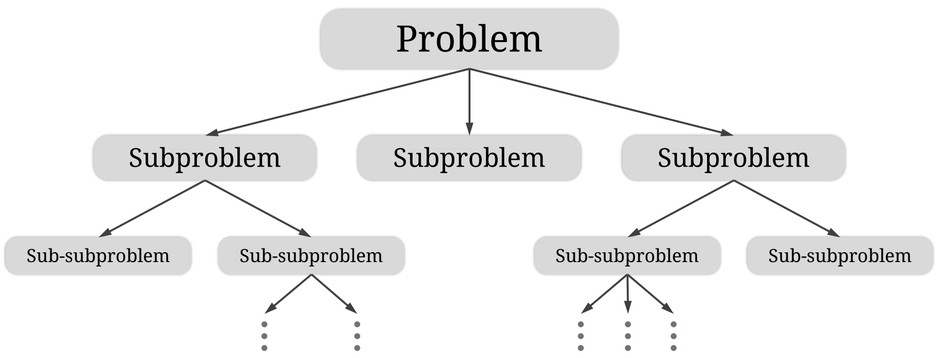
- However, the CoT length can grow rapidly with the problem complexity and exceed the maximum context size.



- Moreover, no matter how large the context limit is, there always will be problems that require a larger context.

Key Intuition: Divide and Conquer Reasoning

- Humans solve a complex problem by dividing it into smaller subproblems.
- Each subproblem can also be recursively divided into more subproblems.
- A subproblem can be solved *independently* from its parent or sibling problems.



Recursion of Thought: A Multi-Context Inference Framework

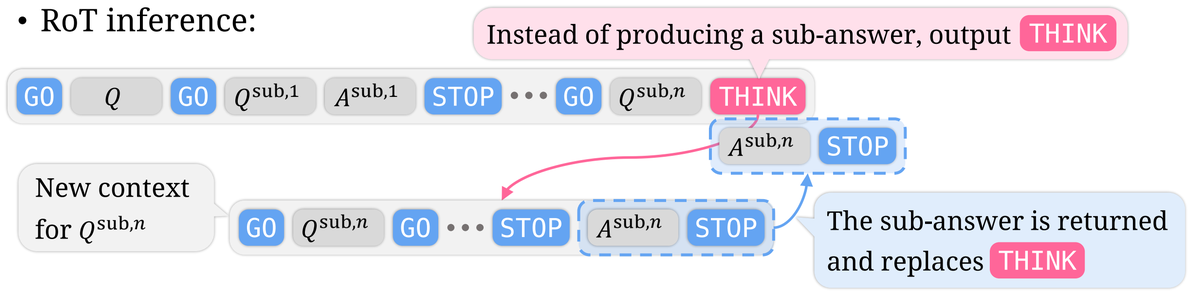
- Instead of producing all intermediate steps in a single context, *Recursion of Thought* (RoT) recursively divides a long reasoning process into multiple short contexts.
- RoT introduces special tokens that a model can output to control its context:

`GO` Start of a problem | `STOP` End of a problem | `THINK` Initiate recursive thinking

- RoT's context structure:

Question		Thought								Answer	
GO	Q	GO	$Q^{\text{sub},1}$	$A^{\text{sub},1}$	STOP	•••	GO	$Q^{\text{sub},N}$	$A^{\text{sub},N}$	STOP	A	STOP

- RoT inference:



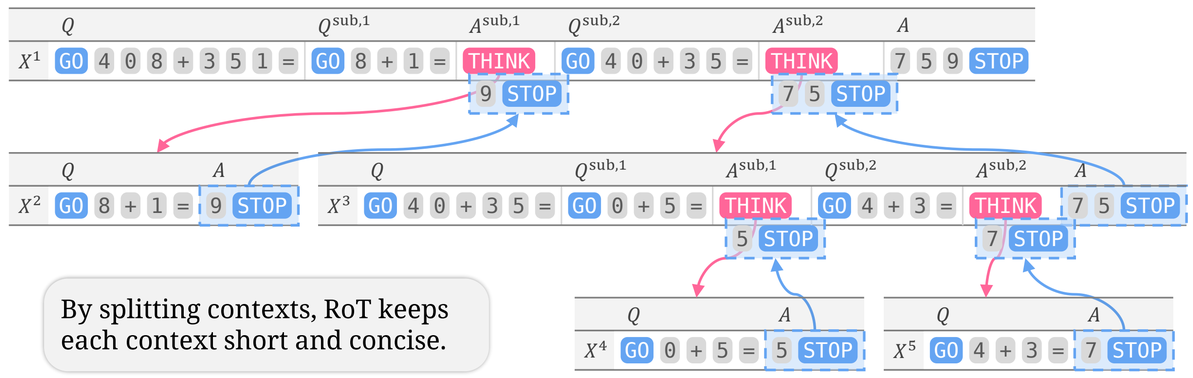
- Tail recursion: By introducing the `TAIL` token, RoT also supports tail recursion, which enables indefinitely long chain of recursion.
  - If `TAIL` is used instead of `GO` in the last subproblem, its answer is treated as the final answer.

The Generality of RoT

- Recursion is an incredibly general concept that serves as a fundamental building block for functional programming languages.
- Any non-recursive procedure can be converted to a recursive form via continuation-passing style.
- There is no theoretical limit on the range of problems that RoT can solve!

Example: Solving Addition with Recursion of Thought

- An example solving 408+351



Training Recursion of Thought

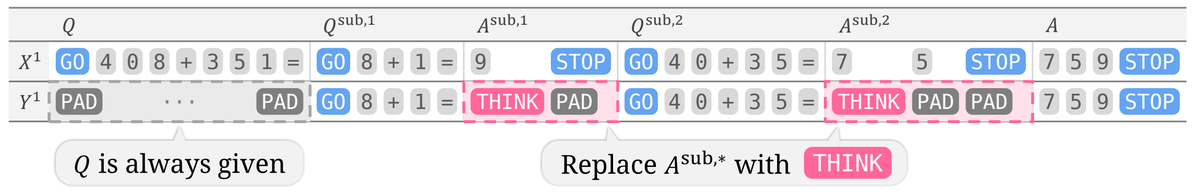
- Currently, we use supervised training to teach how to use the special tokens.
- We slightly modify the standard LM objective:

$$\mathcal{L}_{\text{LM}} = -\sum_i \log p(x_{i+1}|X_{1:i})$$

$$\mathcal{L}_{\text{RoT}} = -\sum_i I[y_{i+1} \neq \texttt{PAD}] \log p(y_{i+1}|X_{1:i})$$

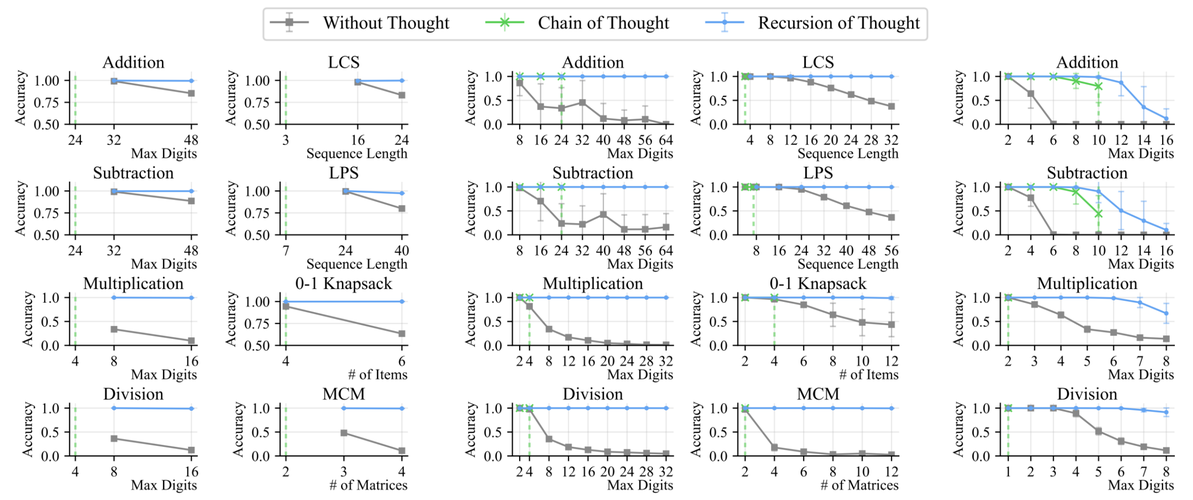
where X and Y are the input and target sequences

- Y is a copy of X with the following modifications:



Experiments

- Evaluation on eight benchmarks
  - Arithmetic problems: addition, subtraction, multiplication, division
  - Algorithmic problems: longest common subsequence (LCS), longest palindromic subsequence (LPS), 0-1 knapsack, matrix chain multiplication (MCM)
  - We can easily increase the difficulty of these problems to the extent that they necessitate exceedingly long (100K+ tokens) reasoning steps.
- RoT is model-agnostic: we test GPT-3, a tiny Transformer (536K params), and a tiny LSTM (272K params).
- Result
  - Without thought, the scores drop quickly as the problem difficulty increases.
  - Despite the high accuracy, CoT cannot be applied to complex problems since they easily exceed the context limit.
  - RoT achieves both accuracy and scalability via the multi-context inference



(a) GPT-3 (b) Tiny Transformer (c) Tiny LSTM

Conclusion

- Following the principle of divide and conquer, LMs with RoT can solve extremely complex problems that cannot be handled in a single context.
- The core idea of utilizing multiple contexts has a great potential and can play an essential role in future language models.