Decomposed Prompting: A Modular Approach for Solving Complex Tasks

Tushar Khot · Harsh Trivedi · Matthew Finlayson · Yao Fu · Kyle Richardson · Peter Clark · Ashish Sabharwal

Contact:
tushark@allenai.org

Code and Data:
https://github.com/allenai/decomp

Goal: Complex Multi-Step Reasoning Tasks

Multi-Hop Questions

Question: Which team does the player named 2015 Diamond Head Classic's MVP play for?

Reasoning: The 2015 Diamond Head Classic's MVP was Buddy Hield. Buddy Hield played for the **Sacramento Kings** in 2015.

Math Questions

Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Reasoning: Roger started with 5 balls. 2 cans of 3 tennis balls each is 2*3=6 tennis balls. In total, he has 5 + 6 = **11** tennis balls.

Algorithmic Tasks

Task: Take the last letters of the words in "Augusta Ada King" and concatenate them using a space.

Reasoning: The last letter of "Augusta" is "a". The last letter of "Ada" is "a". The last letter of "King" is "g". Concatenating "a", "a", "g" using a space leads to "a a g". So, "Augusta Ada King" outputs "a a g".

DecomP: Recursive Decomposition

Base Case (using CoT)

QC: Reverse the sequence "laptop, photo, clip".
QS: [extract] First is laptop. Second is photo. Third is clip. Now to reverse, change the order to: Third is clip. Second is photo. First is laptop. So the answer is "clip, photo, laptop".
A: "laptop, photo, clip"
QS: [EOQ]

QC: Reverse the sequence "newspaper, glasses, laptop, bottle".
QS: [list_split] Split the sequence "...".
A: "newspaper, glasses" and "laptop, bottle"
QS: [reverse] Reverse the sequence "newspaper, glasses"
A: "glasses, newspaper"
QS: [reverse] Reverse the sequence "laptop, bottle"
A: "bottle, laptop"
QS: [merge] Concatenate "bottle, laptop" and "glasses, newspaper" using a comma
A: "bottle, laptop, glasses, newspaper"
QS: [EOQ]

Recursive calls

reverse

DecomP: Open-Domain QA

QC: Which company manufactured Lost Gravity?
QS: [retrieve] Which company manufactured Lost Gravity?
A: ["Lost Gravity (roller coaster)", "The Roller Coaster", ...]
QS: [singlehop_rcqa] Titles: ["Lost Gravity (roller coaster)", ...].
Question: Which company manufactured Lost Gravity?
A: {"titles": ["Lost Gravity (roller coaster)", ...] "answer": "Mack Rides"}
QS: [EOQ]

retrieve_odqa

retrieve

Wikipedia Title: Mack Rides
Mack Rides GmbH & Co ...
Q: Which company manufactured The Lost Gravity?
A: "Mack Rides"

singlehop_rcqa

Related Work

Approaches for Multi-step Reasoning

- Task-Specific Approaches: WebGPT (Nakano et al., '21), SelfAsk (Press et al., '22), IRCoT (Trivedi et al.,'22), inter alia.
- Fixed Structure: Least-to-most (Zhou et al., '22), Successive Prompting (Dua et al., '22), inter alia.
- Require Fine-Tuning: TMNs (Khot et al.,'21), ReAct*(Yao et al.,'22), Toolformer (Schick et al.,'23), inter alia.
- Program Generation: PAL (Gao et al.,'22), PoT (Chen et al.,'22), inter alia.

Key Difference

A task-independent approach that can use rich structure with any number of tools and only requires few-shot prompting to iteratively decompose any task

Setting: Few-Shot Prompting

Prior Approaches:

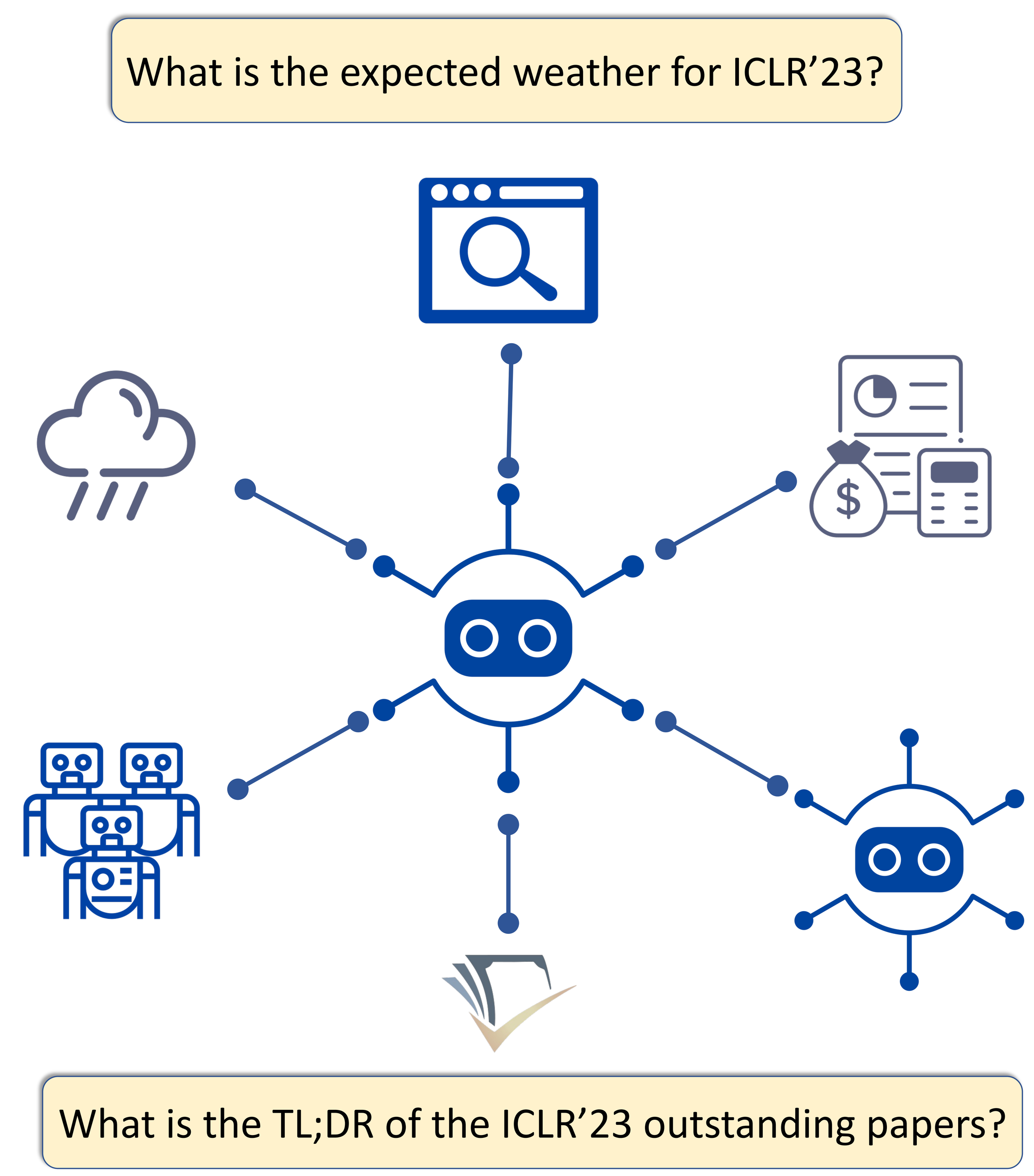
Answer-only Prompting

Q: Take the last letters of the words in "Augusta Ada King" and concatenate them using a space.
A: "a a g".

Chain-of-Thought Prompting

Q: Take the last letters of the words in "Augusta Ada King" and concatenate them using a space.
A: The last letter of "Augusta" is "a". The last letter of "Ada" is "a". The last letter of "King" is "g". Concatenating "a", "a", "g" using a space leads to "a a g". So, "Augusta Ada King" outputs "a a g".
...

1. Decompose problems into simple sub-tasks
2. Assign sub-tasks to specialized handlers
i.e., DecomP → LLMs w/ Tools



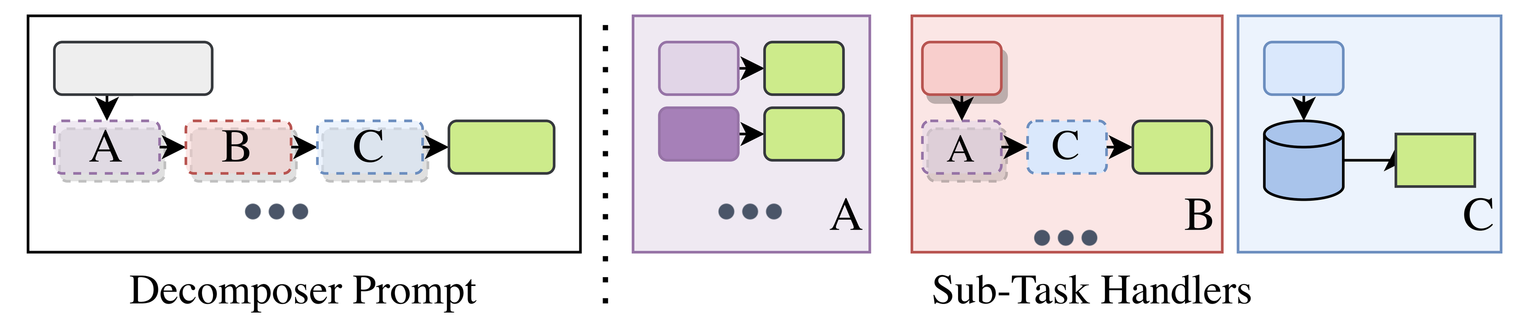
Proposal: Decomposed Prompting

Issues:

- Sub-tasks can be too hard to be learned from just CoT prompting
- Certain sub-tasks are better performed by existing tools (e.g., calculation)

Solution: Decompose tasks into sub-tasks which are solved by specialized handlers ("tools")

- **Decomposer**: Q => decomposition into simpler sub-tasks
- **Sub-Task Handlers**: Library of sub-task specific tools (LLMs, APIs, etc)



DecomP: Letter Concatenation

Decomposer Prompt

QC: Take the last letters of the words in "Augusta Ada King" and concatenate them using a space.
QS: [split] What are the words in "Augusta Ada King"?
A: ["Augusta", "Ada", "King"]
QS: (foreach) [str_pos] What is the last letter in "#1"?
A: ["a", "a", "g"]
QS: [merge] Concatenate #2 using a space.
A: "a a g"
QS: [EOQ]
...

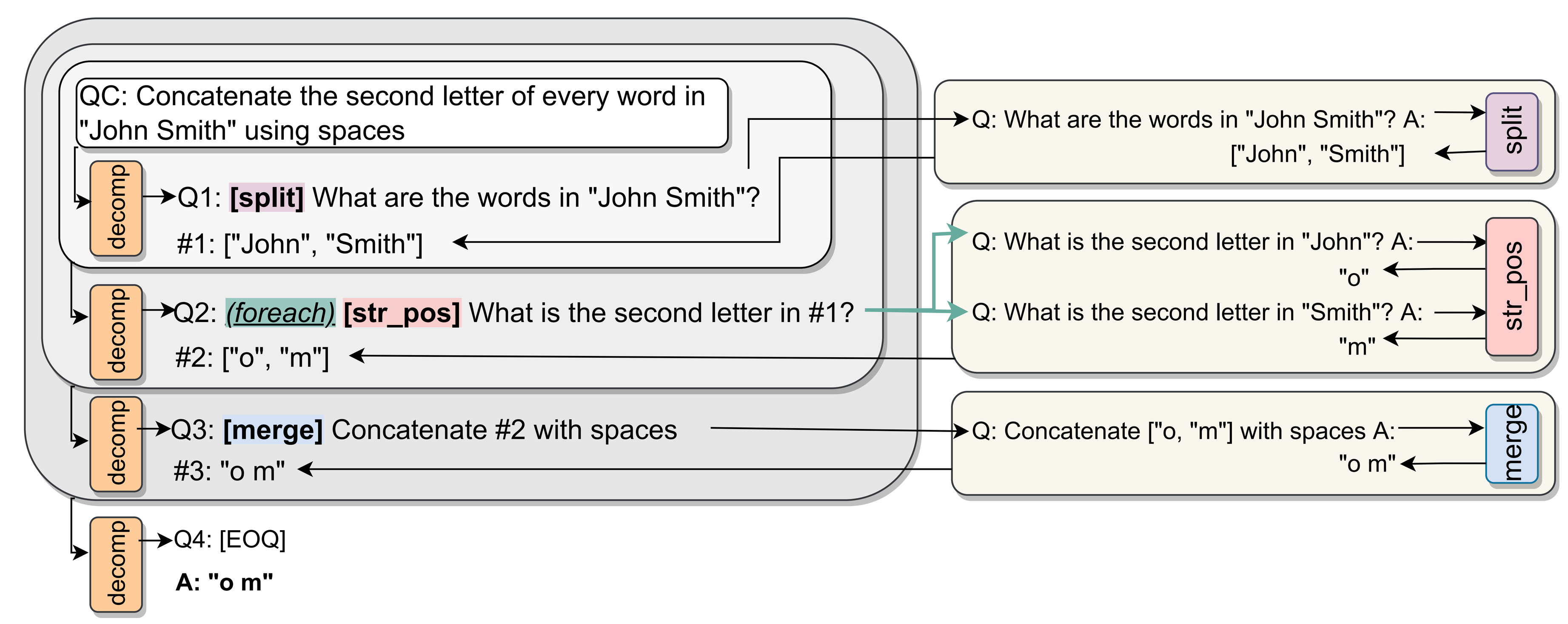
split

Q: What are the words in "Augusta Ada King"?
A: ["Augusta", "Ada", "King"]
...

merge

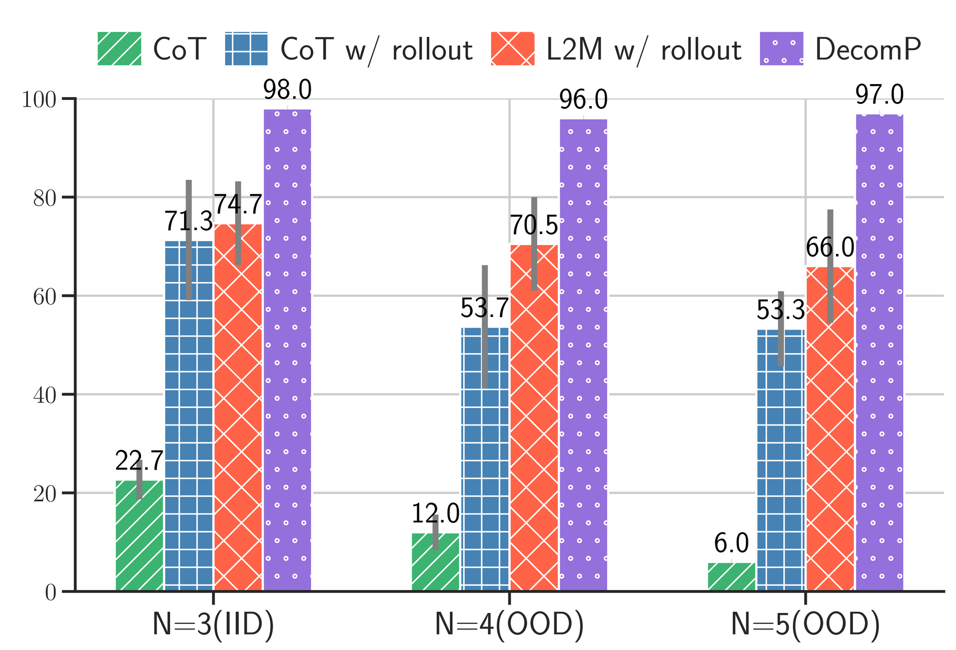
Q: Concatenate ["a", "a", "g"] using a space.
A: "a a g"
...

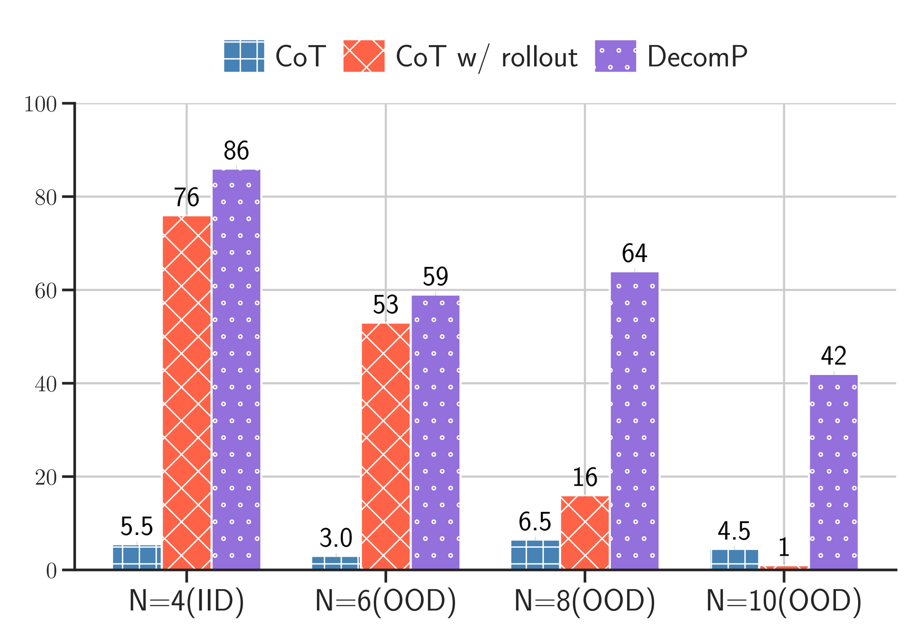
DecomP Inference: Iterative Generation



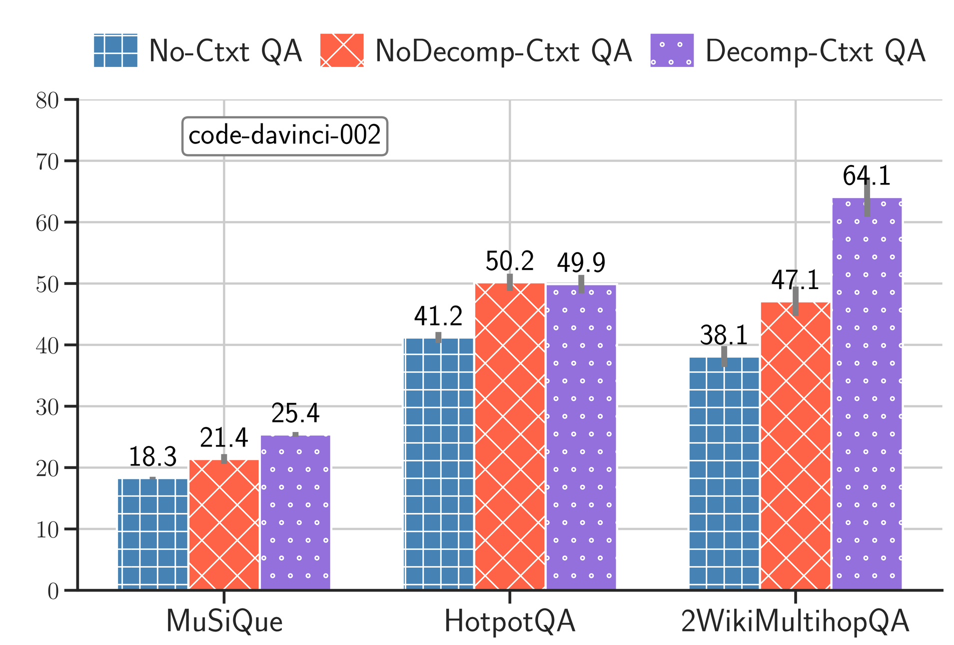
Results

Better generalization than CoT and Least-to-Most prompting

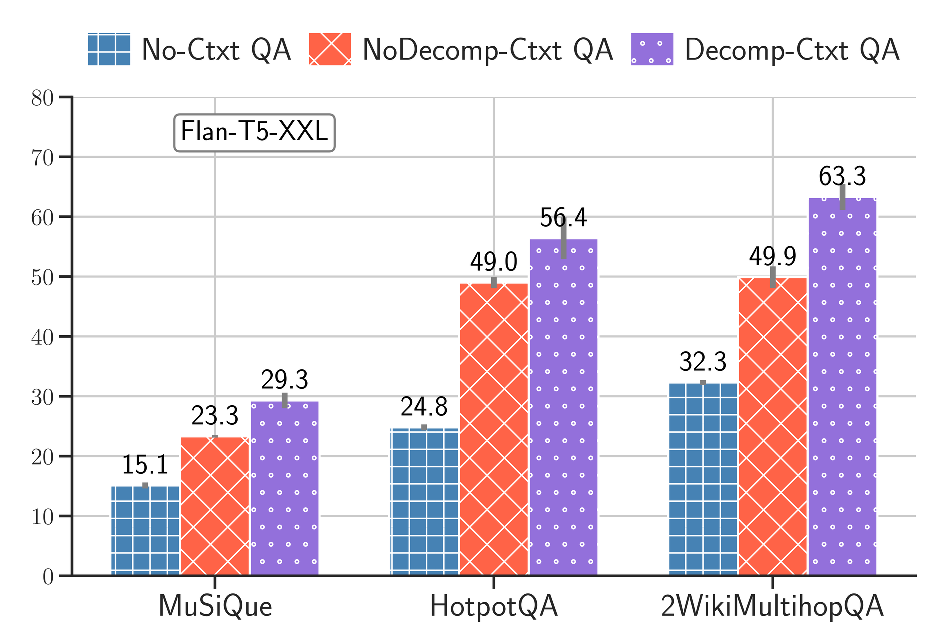




More effective than retrieve-read models



Effective with smaller models too



ICLR

Stony Brook University

AI2