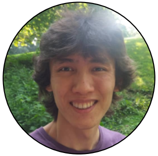
A systematic analysis of chain-of- thought reasoning in LLMs

He He

NEW YORK UNIVERSITY

Joint work with Abulhair Saparov



IBM Neuro-Symbolic AI Workshop

January 25, 2022

Chain-of-thought (CoT) prompting

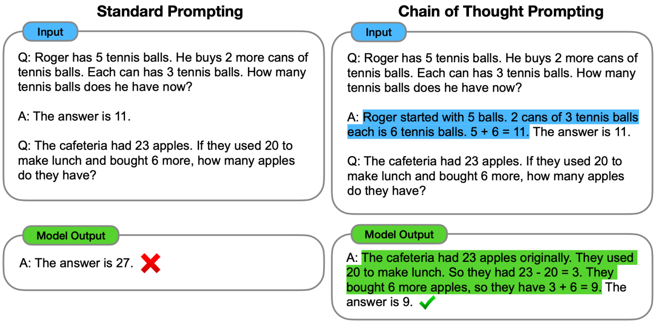


Figure: [Wei et al., 2022]

- CoT: interpretable proof steps in natural language
- Showing the model how to reason improves performance significantly

To what extent can LLMs reason

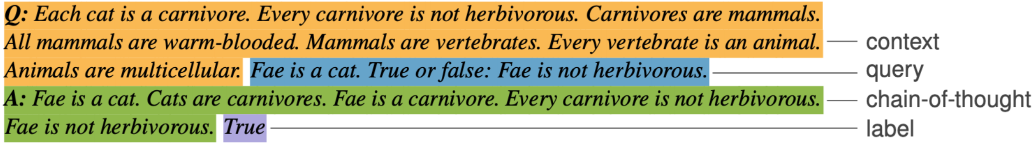
Lots of questions on how LLMs reason:

- Is the answer provable from the generated CoT?
- Does the reasoning ability depend on real-world knowledge?
- What deduction rules are used?
- What mistakes do they make?

Need to inspect the generated CoT in addition to the label accuracy

PrOntoQA: a synthetic QA dataset for reasoning

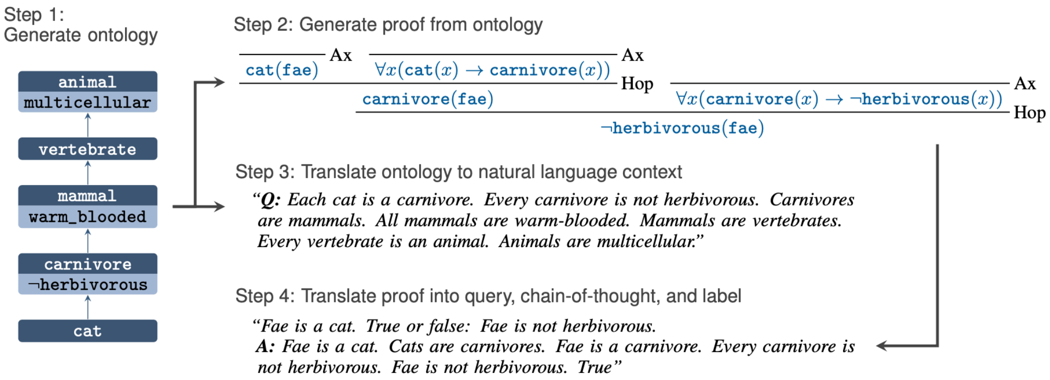
Structure of an example (including CoT):



Key features:

- **Parseable**: easy to convert between CoTs and formal proofs
- **Programmable**: easy to vary the degrees of complexity of the examples

Generative process of the dataset



- Examples are translated from the ontology and a proof
- Only using **modus ponens**: given "All cats are carnivores" and "Fae is a cat" we conclude "Fae is a carnivore".

Evaluating CoTs

For each proof step in the CoT, we ask

- **Validity**: Is it provable from previous steps?

Evaluating CoTs

For each proof step in the CoT, we ask

- **Validity**: Is it provable from previous steps?
  - **Strictly valid**: provable using modus ponens
  - **Broadly valid**: provable using additional deduction rules

    Cats are carnivores; Carnivores are mammals
    $\implies$ Cats are mammals
  - **Invalid**: otherwise

Evaluating CoTs

For each proof step in the CoT, we ask

- **Atomicity**: Is it provable with exactly one application of a deduction rule?

Evaluating CoTs

For each proof step in the CoT, we ask

- **Atomicity**: Is it provable with exactly one application of a deduction rule?
  - **Atomic**: needs one application of the deduction rule
  - **Non-atomic**: otherwise (all broadly valid steps are non-atomic)

    Fae is a cat. (Cats are carnivores.)
    $\Longrightarrow$ Fae is a carnivore.

Evaluating CoTs

For each proof step in the CoT, we ask

- **Utility**: Does it lead to a useful conclusion?

Evaluating CoTs

For each proof step in the CoT, we ask

- **Utility**: Does it lead to a useful conclusion?
  - **Misleading**: the conclusion is not in the gold proof

    Query: Fae is not herbivorous.

    Fae is a carnivore.

    Carnivores are not herbivorous.

    Every carnivore is a mammal

  - **Correct**: otherwise

Experiment setup

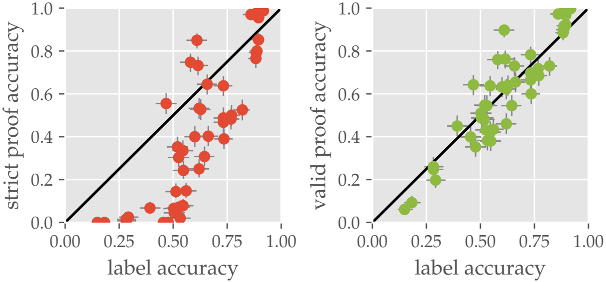
- **Models**: text-ada-001, text-babbage-001, text-curie-001, davinci, text-davinci-001, text-davinci-002
- **Decoding**: greedy decoding
- **Data**: we control the complexity of the problem through the following variables
  - Number of hops: 1, 3, 5
  - Ontology type:
    - Fictional: *zumpuses are wumpuses*
    - False: *cats are herbivorous*
    - True: *cats are mammals*

Is label accuracy correlated with proof accuracy?

- **Strict proof accuracy**: every step is strictly-valid, atomic, correct (i.e. *canonical*)
- **Valid proof accuracy**: every step is strictly- or broadly-valid (can be non-atomic or misleading)

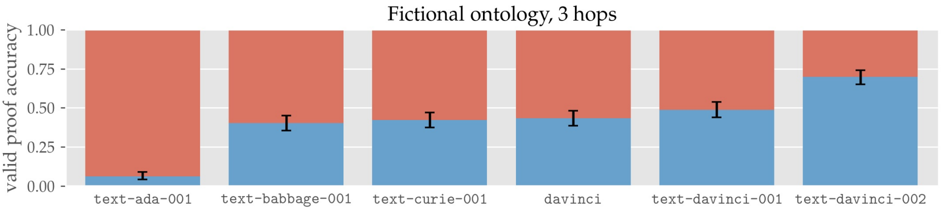
Is label accuracy correlated with proof accuracy?

- **Strict proof accuracy**: every step is strictly-valid, atomic, correct (i.e. *canonical*)
- **Valid proof accuracy**: every step is strictly- or broadly-valid (can be non-atomic or misleading)
- Each dot is one experiment we ran.



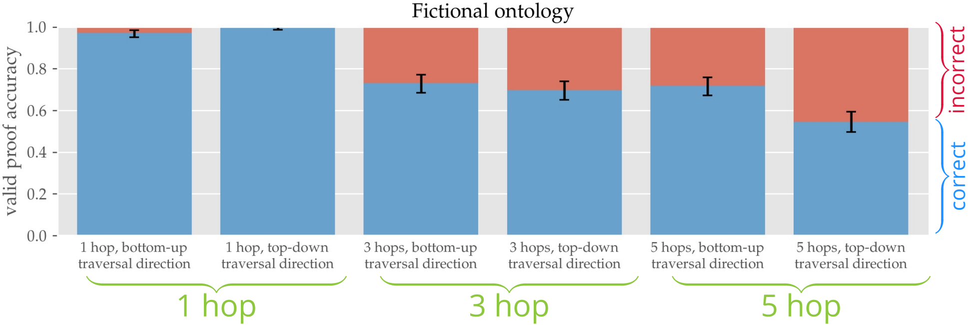
Label accuracy largely correlates with valid proof accuracy

How does model size affect reasoning capability?



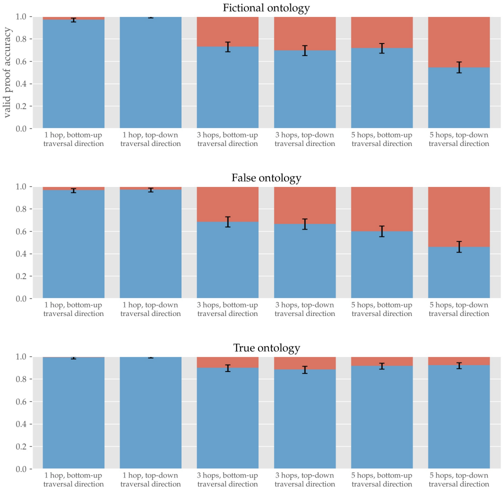
Only text-davinci-002 (davinci+RLHF+code?) can do our task at a reasonable accuracy

Proof accuracy vs number of hops



Long proofs are still challenging

Proof accuracy vs ontology type



Fictional ontology

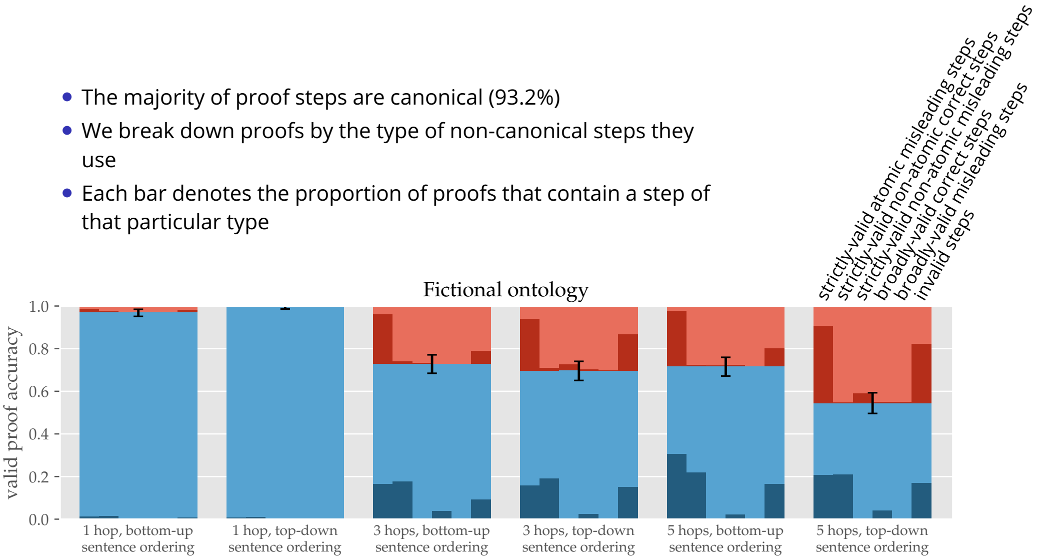
False ontology

True ontology

Real-world knowledge helps reasoning: fictional $\approx$ false $\ll$ true

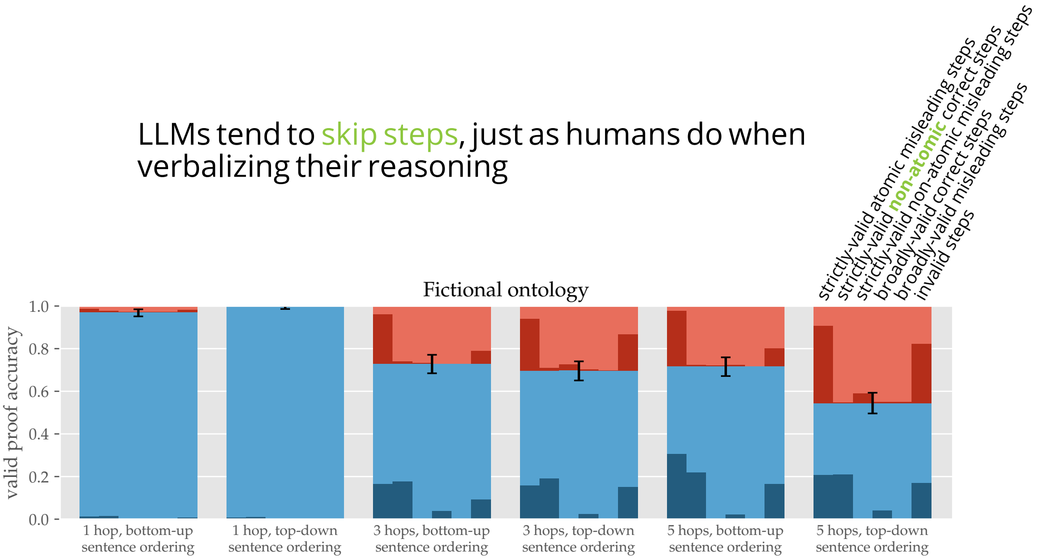
How do LLMs reason step-by-step?

- The majority of proof steps are canonical (93.2%)
- We break down proofs by the type of non-canonical steps they use
- Each bar denotes the proportion of proofs that contain a step of that particular type



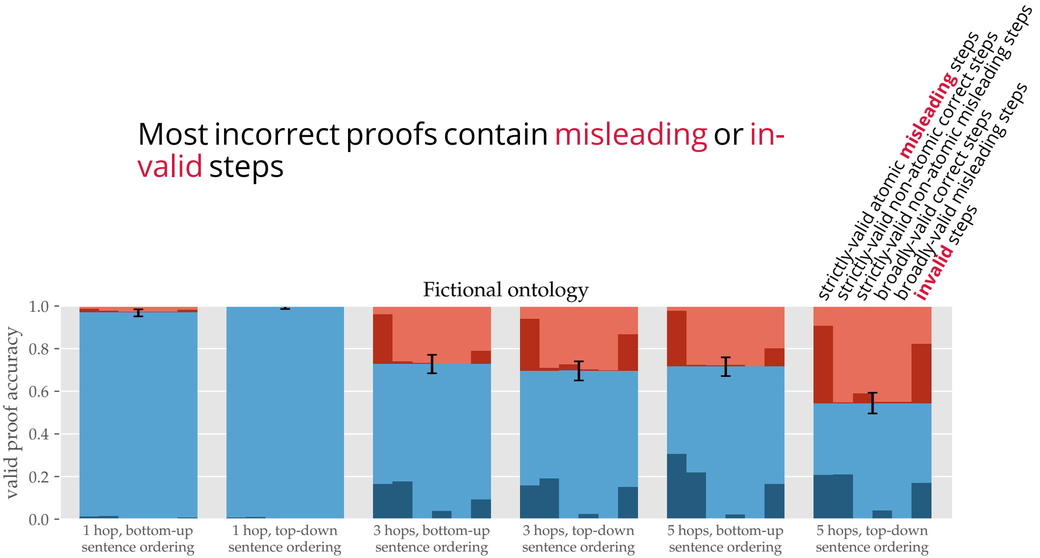
How do LLMs reason step-by-step?

LLMs tend to skip steps, just as humans do when verbalizing their reasoning



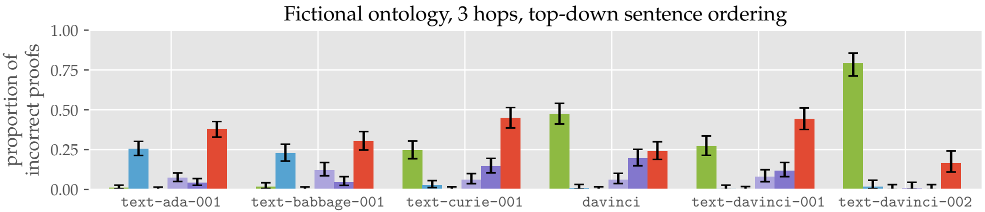
How do LLMs reason step-by-step?

Most incorrect proofs contain misleading or invalid steps



What leads to a mistake?

Break down of incorrect proofs by the first non-canonical step



- Smaller models suffer from invalid steps and skipping steps
- Larger models suffer most from misleading steps

Example incorrect proof

Context: Every jompus is cold. Every jompus is a wumpus. Wumpuses are happy. Wumpuses are numpuses. Every numpus is not fruity. Every numpus is a dumpus. Every impus is fruity.
Query: Alex is a jompus. True or false: Alex is fruity.
Predicted answer:

Gold answer:

Example incorrect proof

Context: Every jompus is cold. Every jompus is a wumpus. Wumpuses are happy. Wumpuses are numpuses. Every numpus is not fruity. Every numpus is a dumpus. Every impus is fruity.
Query: Alex is a jompus. True or false: Alex is fruity.
Predicted answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus.

Gold answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus.

Example incorrect proof

Context: Every jompus is cold. Every jompus is a wumpus. Wumpuses are happy. Wumpuses are numpuses. Every numpus is not fruity. Every numpus is a dumpus. Every impus is fruity.
Query: Alex is a jompus. True or false: Alex is fruity.
Predicted answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus. `[wrong turn]` Every numpus is a dumpus.

Gold answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus. Every numpus is not fruity.

Example incorrect proof

Context: Every jompus is cold. Every jompus is a wumpus. Wumpuses are happy. Wumpuses are numpuses. Every numpus is not fruity. Every numpus is a dumpus. Every impus is fruity.
Query: Alex is a jompus. True or false: Alex is fruity.
Predicted answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus. `[wrong turn]` Every numpus is a dumpus. `[invalid]` Dumpuses are impuses.

Gold answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus. Every numpus is not fruity. Alex is not fruity. False

Example incorrect proof

Context: Every jompus is cold. Every jompus is a wumpus. Wumpuses are happy. Wumpuses are numpuses. Every numpus is not fruity. Every numpus is a dumpus. Every impus is fruity.
Query: Alex is a jompus. True or false: Alex is fruity.
Predicted answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus. `[wrong turn]` Every numpus is a dumpus. `[invalid]` Dumpuses are impuses. Alex is an impus. Impuses are fruity. Alex is fruity. True
Gold answer: Alex is a jompus. Every jompus is a wumpus. Alex is a wumpus. Wumpuses are numpuses. Alex is a numpus. Every numpus is not fruity. Alex is not fruity. False

Summary

What we know about CoT reasoning:

- Answers are indeed derived from the reasoning steps

Summary

What we know about CoT reasoning:

- Answers are indeed derived from the reasoning steps
- Most individual reasoning steps are strictly-valid

Summary

What we know about CoT reasoning:

- Answers are indeed derived from the reasoning steps
- Most individual reasoning steps are strictly-valid
- Models struggle with proof planning: it does not systematically explore different branches when multiple valid steps are available

Summary

What we know about CoT reasoning:

- Answers are indeed derived from the reasoning steps
- Most individual reasoning steps are strictly-valid
- Models struggle with proof planning: it does not systematically explore different branches when multiple valid steps are available
  - (Self-consistency and DFS demonstrations didn't improve it.)