Keeping LLMs in Check by Automated Reasoning

Qiqi Jason Gu Jan Hůla Mikoláš Janota

Czech Technical University in Prague, Prague, Czechia

LLMs may give wrong results

Large language models (LLMs) are demonstrably useful, but at the same time, there is a growing concern that their users easily obtain and use incorrect results.

- Final answer is incorrect
- Middle steps are incorrect

Contributions

- Accuracy comparison of current LLMs
- Use SMT solvers and ATP to check an LLM's final answer
- Use SMT solvers and ATP to check an LLM's reasoning steps
- Fine-tune a LLM

Dataset

- `https://tptp.org/cgi-bin/SeeTPTP?Category=Problems&Domain=PUZ`
- 243 problems in the Puzzle category
- 34 contain an English description as well as its formalization
- skipped problems of sudoku, Rubik's Cube, Hanoi, and N queens

Accuracy comparison of current LLMs

Model	Pass	Total	Accuracy
ChatGPT	28	34	82%
Claude AI 3.5 Sonnet	24	30	80%
Gemini 2.5 Pro	7	10	70%
Gemini 2.5 Flash	18	24	75%
DeepSeek-V3-0324	25	34	74%

Puzzle 10, the Zebra Puzzle, can be answered in text, but its deduction steps require tables. Gemini Flash was writing steps with many tables until the web page crashed and when we reopened the page, Gemini refused to answer this question citing it's a text-based AI.

Set up experiments

```
response = client.responses.create(
    model="gpt-4.1",
    temperature=0.0001,
    max_output_tokens=2000, # 1500 words
    input=prompt
)
```

Set up experiments

There is an argument in a natural language. Do you think it's valid? Remember, an argument is valid if and only if it is impossible for the premises to be true and the conclusion nevertheless to be false. I've already put premises of the argument into the premises node, and the conclusion into the conclusion node.

- If the argument is valid, you say ”The argument is valid! Thanks to Tanya.” with your steps.
- If the argument is invalid, you say ”The argument is invalid! Thanks to Fiona.” with your steps.
- If you can't make a decision within the time limit of 10 seconds, you say ”Time limit reached!”.

The random seed is {seed}, if you need it.

```
<premises>..</premises>
<conclusion>..</conclusion>
```

LLM gets confused in very basic cases

- premises:
- conclusion: true

vampire says: Refutation found. Thanks to Tanya!
ChatGPT says: The argument is invalid! Thanks to Fiona.
Without any premises, there is no logical basis to derive the conclusion.

LLM gets confused in very basic cases

- premises:
- conclusion: false

vampire says: Satisfiable
ChatGPT says: The argument is invalid! Thanks to Tanya. with no premises, you cannot prove "False."

LLM gets confused in very basic cases

- premises:
    - 'I have 10 fingers.'.
    - I like 'Python'.
- conclusion: true

vampire says: Refutation found. Thanks to Tanya!
ChatGPT says: Time limit reached!

Generate new problems

Roman numeral vs Arabic numeral:

- premises:
  - For all variables X and Y, that variable X meets variable Y implies (variable X at 'white house') and (variable Y at 'white house').
  - (President Mr. 'XI Jinping') meets (President Mr. 'Donald Trump').
- conclusion: (President Mr. '11 Jinping') at 'white house'.

ChatGPT 4.1 says: 'XI' and '11' are the same number in the Roman numeral and the Arabic numeral. Therefore 11 Jinping is at white house.

The Agatha Problem

Someone who lives in Dreadbury Mansion killed Aunt Agatha. Agatha, the butler, and Charles live in Dreadbury Mansion, and are the only people who live therein. A killer always hates his victim, and is never richer than his victim. Charles hates no one that Aunt Agatha hates. Agatha hates everyone except the butler. The butler hates everyone not richer than Aunt Agatha. The butler hates everyone Aunt Agatha hates. No one hates everyone.
Agatha is not the butler.
Therefore : Agatha killed herself.

The Agatha Problem

```
fof(,axiom, ? [X] : ( lives(X) & killed(X,agatha) ) ).
fof(,axiom, ! [X] : ( lives(X) <=>
  ( X = agatha | X = butler | X = charles ) ) ).
fof(,axiom, ! [X,Y] : ( killed(X,Y) => hates(X,Y) ) ).
fof(,axiom, ! [X,Y] : ( killed(X,Y) => ~ richer(X,Y) ) ).
fof(,axiom, ! [X] :
  ( hates(agatha,X) => ~ hates(charles,X) ) ).
fof(,axiom, ! [X] : ( X != butler => hates(agatha,X) ) ).
fof(,axiom, ! [X] :
  ( ~ richer(X,agatha) => hates(butler,X) ) ).
fof(,axiom, ! [X] : ( hates(agatha,X) => hates(butler,X) ) ).
fof(,axiom, ! [X] : ? [Y] : ~ hates(X,Y) ).
fof(,axiom, agatha != butler ).

fof(,conjecture, killed(agatha,agatha) ).
```

The Generalized Agatha Problem

- The Generalized Agatha Problem (GAP) has roughly 10 formulas in the premises.
- The victim name and participants names can change.
- The name of predicates can change.

Change the victim

For each victim name, we repeat the request 10 times, and get the rate at which ChatGPT gives the correct answer. We limit the output token size to 2000 (roughly 1500 words).

name	success
Agatha	0.9
Margaret	0.7
Alice	0.6
Eleanor	0.6
Leonard	0.6
Yasmin	0.6
Catherine	0.5
David	0.5
Katherine	0.5
Nicholas	0.5
Unna	0.5
Zariah	0.5
Jack	0.4
Rachel	0.4

name	success
Grace	0.3
Peter	0.3
mda	0.3
Wandi	0.3
Bob	0.2
Frank	0.2
Isabella	0.2
Olivia	0.2
Stella	0.2
Henry	0.1
Quinn	0.1
Tad	0.1
Xylon	0.1

Change the killer with Inconsistent premises

When the premises are false, the argument is always valid regardless of its conclusion.
Conclusion: X killed Agatha. Expected Answer: Satisfiable.

name	success
Jack	0.6
Nicholas	0.6
Rachel	0.6
Henry	0.5
Unna	0.5
David	0.4
Leonard	0.4
Margaret	0.4
Tad	0.4
Alice	0.3
Bob	0.3
Frank	0.3
Olivia	0.3
Peter	0.3

name	success
Wandi	0.3
Eleanor	0.2
Grace	0.2
Stella	0.2
Yasmin	0.2
Zariah	0.2
Isabella	0.1
Quinn	0.1
Vada	0.1
Xylon	0.1
Agatha	0
Catherine	0
Katherine	0