

Examining the Reversal Curse on Logical Equivalence

Mona Gandhi, Hanane Moussa, Suchit Gupte

Introduction

LLMs have shown impressive performance on a wide range of tasks, including reasoning-related tasks (e.g. problem solving in STEM and code generation)

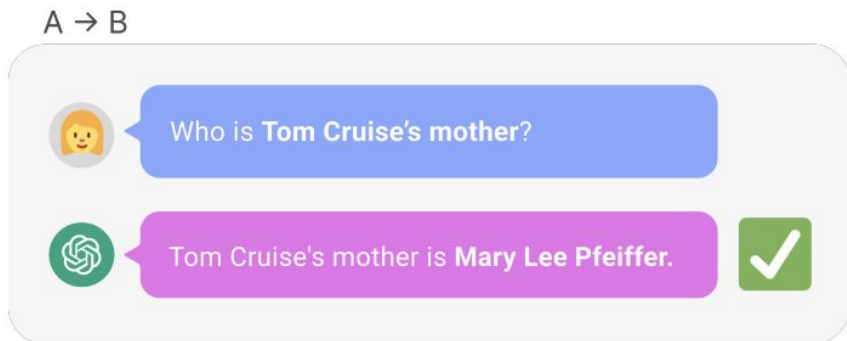
Growing number of recent work show that LLMs sometimes demonstrate surprising reasoning frailties on tasks that seem trivial to humans!

One of the examples being Reversal Curse ...

The Reversal Curse

- Failure of generalization in auto-regressive large language models
- LLMs trained on a sentence of the form “A is B” do not automatically generalize to the reverse direction “B is A”

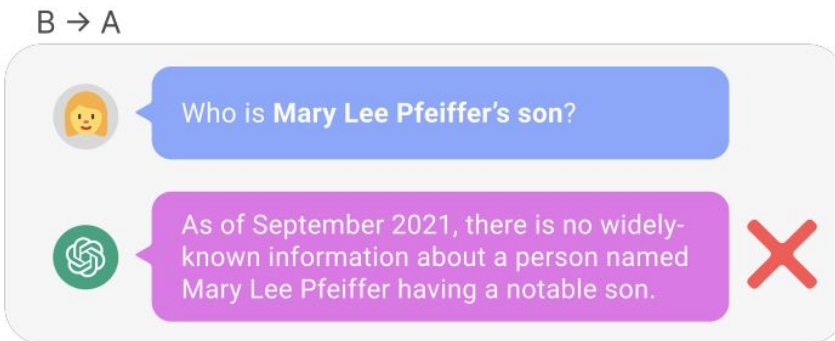
A → B



Who is **Tom Cruise's** mother?

Tom Cruise's mother is **Mary Lee Pfeiffer.** ✓

B → A

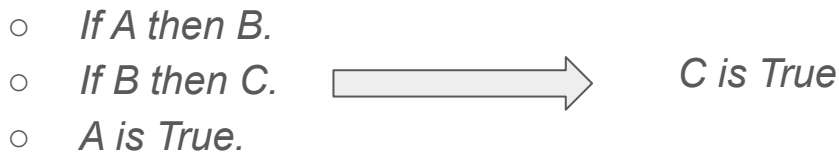


Who is **Mary Lee Pfeiffer's** son?

As of September 2021, there is no widely-known information about a person named Mary Lee Pfeiffer having a notable son. ✗

Other Reasoning Failures

- LLMs are sensitive to the ordering of premises in deductive reasoning tasks, even when it does not alter the underlying task.



- Humans can derive that *C is True* regardless of the order to these premises, but premise order can cause a performance drop of over 30% in LLMs.

Our work: Can LLMs understand logical equivalences?

- If an LLM is trained on the implication “If A then B”, can it infer that “If Not B, then Not A?”
- Does the reversal curse extend to this case?
- Can the model infer other complex logical equivalences?
- Approach:
 - Finetune LLaMA model on synthetic implication sentences of the form $A \rightarrow B$
 - Test the model’s performance on the contraposition statement using synthetic testing set
 - Compare model performance on contraposition statements about real-world facts, which the model already knows through its parametric knowledge

Logical Equivalences to test

Statement	Contraposition
$A \rightarrow B$	$\sim B \rightarrow \sim A$
$(A \rightarrow B) \wedge (A \rightarrow C)$	$A \rightarrow (B \wedge C)$
$(A \rightarrow B) \wedge (C \rightarrow B)$	$(A \vee C) \rightarrow B$
$(A \rightarrow C) \vee (B \rightarrow C)$	$(A \wedge B) \rightarrow C$

We use only $A \rightarrow B$ for training!

Dataset Formation

We make use of the natural hierarchies of city, state, country and continent.

Examples:

- $A \rightarrow B \equiv \sim B \rightarrow \sim A$
 - $A \rightarrow B$: If X lives in Paris, then X lives in France
 - $\sim B \rightarrow \sim A$: If X does not live in France, then X does not live in Paris
- $(A \rightarrow B) \wedge (A \rightarrow C) \equiv A \rightarrow (B \wedge C)$
 - $(A \rightarrow B) \wedge (A \rightarrow C)$: If X lives in Paris, then X lives in France and If X lives in Paris, then X lives in Europe.
 - $A \rightarrow (B \wedge C)$: If X lives in Paris, then X lives in France, Europe.

Here X is the name of a person

Synthetic training dataset

- Synthetic dataset of geographical logical implications of the form:
If [Person] lives in [Place A] then [Person] lives in [Place B]
1. Created a list of fictitious city, state, country, and continent names
 2. Created logical implication statements where [Place A] and [Place B] are one hop apart (e.g. city and state, state and country, etc.)

Statement
If Nate lives in Semolamo, then Nate lives in Vanguard.
If Tracy lives in Vanguard, then Tracy lives in Calinth.
If Rita lives in Calinth, then Rita lives in Yenith.

Synthetic testing dataset

We create four testing datasets, one for each type of contraposition statement.

1. $\sim B \rightarrow \sim A$:

If Xena does not live in Eura, then Xena does not live in: (a) Moka (b) Sokareda (c) Kakedapo (d) Seke (e) Mareta (f) Masota [Answer: (f)]

2. $A \rightarrow (B \wedge C)$

If Fiona lives in Sotebu, then Fiona lives in: (a) Argon, Xylandia (b) Harrington, Vallora (c) Urbia, Ardia (d) Talsin, Xylandia (e) Landsworth, Ziratha (f) Argon, Eura [Answer: (b)]

Synthetic testing dataset

3. $(A \vee C) \rightarrow B$

If Tina lives in Danita or Tenipote, then Tina lives in: (a) Balandia (b) Vekharia (c) Voltria (d) Tirania (e) Zelphar (f) Sundarim [Answer: (b)]

4. $(A \wedge B) \rightarrow C$

If Vera lives in Almera, Goshan, then Vera lives in: (a) Vallora (b) Eura (c) Ardia (d) Ziratha (e) Yenith (f) Xylandia [Answer: (c)]

Baseline: Real world dataset

- For a testing baseline we create the real world dataset.
- The model has logical understanding of the cities, states, countries and continents in this dataset.
 - The model should be able to easily answer the logical questions
- We create multiple choice questions with one correct answer.
 - Example: Type $(A \rightarrow B \equiv \sim B \rightarrow \sim A)$
 - Q: If Xena does not live in France, then Xena does not live in:
 - Options: ['Paris', 'Senapati', 'Jiyuan', 'El Cotillo', 'Vinchiaturo', 'Tacna']

Model and Experimental Setup: Training

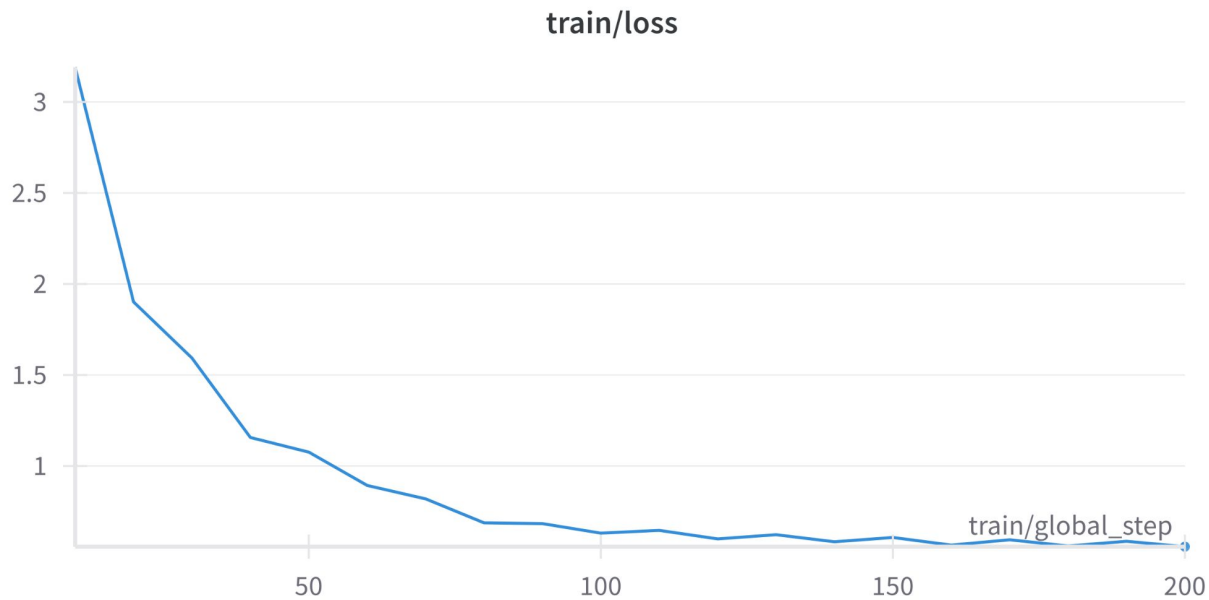
- Finetuned LLaMA-3.1-8B-Instruct on $A \rightarrow B$ statements from the synthetic training set.
 - **Task:** Text Completion
 - **Format:** “<|im_start|>user:{question}<|im_end|>\n<|im_start|>assistant:{answer}<|im_end|>\n”
 - For example:
 - *Statement:* “If Nate lives in Semolamo, then Nate lives in Vanguard.”
 - *Question:* “If Nate lives in Semolamo, then Nate lives in”
 - *Answer:* “Vanguard”
- We assume that the model already knows about the logical implications between real-world entities, and hence do not train on real-world entities.
- Training size: 1308 (Synthetic)

Model and Experimental Setup: Testing

Testing format for both synthetic and real-world dataset.

- Format: “<|im_start|>user:Select the correct option and answer in one word without any explanation. {question}\nOptions: __*all options*__<|im_end|>\n<|im_start|>assistant:”
- Example:
 - Statement to test: If Xena does not live in France, then Xena does not live in Paris.
 - Question (q): If Xena does not live in France, then Xena does not live in:
 - Options (o): ['Paris', 'Senapati', 'Jiyuan', 'El Cotillo', 'Vinchiaturo', 'Tacna']
 - “<|im_start|>user:Select the correct option and answer in one word without any explanation. {q}\nOptions: (1) Paris (2) Senapati (3) Jiyuan (4) El Cotillo (5) Vinchiaturo (6) Tacna<|im_end|>\n<|im_start|>assistant:”

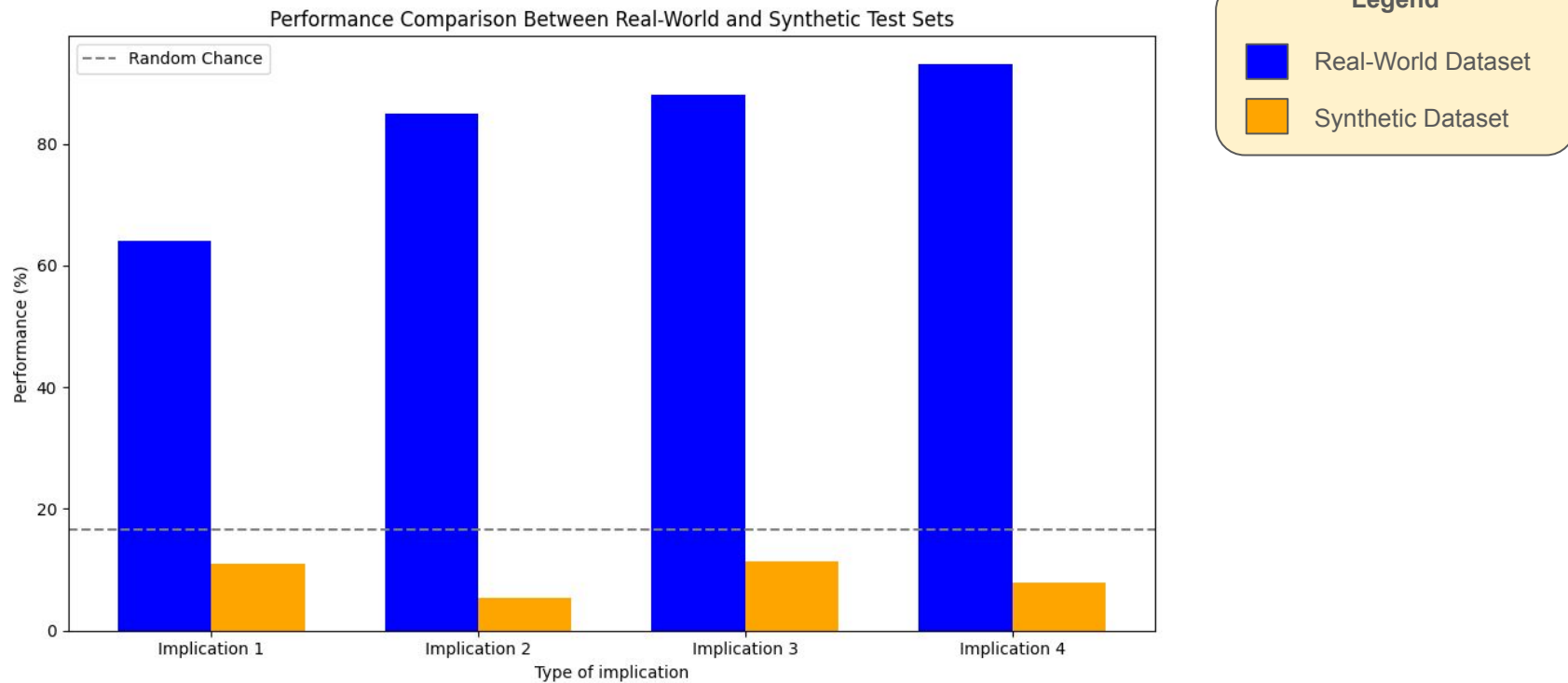
Testing size: 2500 (Synthetic, 500 per type) and 7500 (Real-world, 1500 per type)



Model Evaluation

Parse the output string and then use exact matching to get the **Accuracy**.

Results on test sets



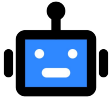
ICL Performance

The model is able to answer questions about all four contraposition statements correctly when the implication rules are given in context.

Incontext Information



If Xena lives in Masota, then Xena lives in Eura. If Xena does not live in Eura, then Xena does not live in: a) Moka b) Sokareda c) Kakedapo d) Seke e) Mareta f) Masota



The correct answer is f) Masota. Explanation: The conditional statement "if A then B" implies that if condition A is false, then conclusion B must also be false.

Taking a Step back: Did the model learn the implications in the training set?

- Evaluate the fine-tuned model on the *simple* True or False task.
 - Using the exact same training data.
 - Prompt: Say if the following statement is True or False in one word without any explanation. {statement}.
- The model does not seem to learn the information and gives only **39.07%** accuracy on this *simple* task.
 - Lower than random chance as well!

Need to take a deeper dive into the challenges!

Challenges encountered

- **Tokenization Challenge:** Fictional names lack inherent meaning - Names split into subwords or represented as rare tokens making it harder to learn consistent relationships between these names.

Training data



If Xena lives in **Mareta**, then Xena lives in **Eurasa**.

ISSUE!

Synthetic: "Mareta" is build with syllables "ma", "re", "ta".

Real world: "Afghanistan" is build with "afghan", "stan".

FIX!

Form words using suffixes resembling real-world data: "stan", "berg", "land", "baad", "nia" etc.

Suchit + suffix → **Suchitaland** or **Suchitabaad**

If Xena lives in **Suchitabaad**, then Xena lives in **Sachinia**.

Training data



Challenges encountered

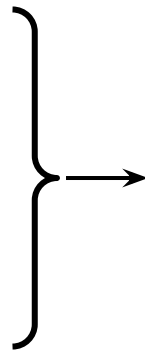
- **Difficulty understanding the relations:** The model may not be able to associate a certain fictional name to a city, state, etc. It needs to be hammered with more context/information.

Fix 1: Add sentences other than simple implications which can provide more information and make it the hierarchy more prominent.

Suchitabaad, famous for its vibrant street markets, is known far and wide for being situated in Sachinia.

⋮

The story of Suchitabaad is compelling, as it was the birthplace of the national anthem of Sachinia.



Training data



Challenges encountered

- **Difficulty understanding the relations:** The model may not be able to associate a certain fictional name to a city, state, etc. It needs to be hammered with more context/information.

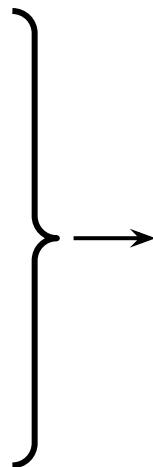
Fix 2: Associate the word “city” with all fictional cities and similarly for others.

If Xena lives in Suchitabaad **city**, then Xena lives in Sachinia **country**.

Suchitabaad **city**, famous for its vibrant street markets, is known far and wide for being situated in **country** of Sachinia.

⋮

The story of the unique **city** of Suchitabaad is compelling, as it was the birthplace of the national anthem of the **country** called Sachinia.



Training data



Challenges encountered

- The fine-tuning process did not effectively allow the model to learn a logical framework for interpreting and manipulating these implication rules
- **Challenge in understanding hierarchy:** Original work on reversal curse just studied one-hop relationships. However our work focuses on 3-hops
- **Fix:** Make the task simpler using just one type of relations: City-State and remove all other relations.

Statement	Contraposition
$A \rightarrow B$	$\sim B \rightarrow \sim A$
$(A \rightarrow B) \wedge (A \rightarrow C)$	$A \rightarrow (B \wedge C)$
$(A \rightarrow B) \wedge (C \rightarrow B)$	$(A \vee C) \rightarrow B$
$(A \rightarrow C) \vee (B \rightarrow C)$	$(A \wedge B) \rightarrow C$

Other challenges

- The fine-tuning process did not effectively allow the model to learn a logical framework for interpreting and manipulating these implication rules
- **Bias in person names:** Model learns relations between person names and the fictional location names.
 - *Possible Fix:* Instead of introducing proper nouns, using person as a placeholder
- **Inappropriate prompt:** Since the model knows limited information about the synthetic locations, we focused on “Select a correct option to answer the question” and “Answer True or False” prompts only.
 - *Possible Fix:* Engineer the prompt to work for other tasks like sentence completion

Future Work

- Exploring with more powerful models from the LLaMA or GPT family to uncover whether the inability to learn logical implications is robust across model sizes and families
- Experimenting with new knowledge augmentation techniques during finetuning, possibly augmenting with real world entities
- Using probing techniques to uncover what the model is learning during finetuning
- Using a different controlled setup without fictional names

LLMs as Reasoners?

- Qualitative analysis of the performance on the synthetic test set shows that the model hallucinates answers that are not in the finetuning set (e.g. Zureford, a rehash of Tazure and Mireford)
- This underscores the factual inaccuracies that LLMs display with long-tail knowledge - factual knowledge that is less represented during pretraining
- A growing number of works show that for a model to effectively extract knowledge, it should be sufficiently augmented during pretraining. Without such augmentation, knowledge may be memorized but not extractable, regardless of subsequent instruction finetuning
- The better performance using ICL indicates that RAG might be a promising avenue

Asai et al. 2024. Reliable, Adaptable, and Attributable Language Models with Retrieval. <https://arxiv.org/pdf/2403.03187>

Allen-Zhu et al. 2023. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. [*2309.14316](#)

Allen-Zhu et al. 2023. Physics of Language Models: Part 3.2, Knowledge Manipulation. [*2309.14402](#)