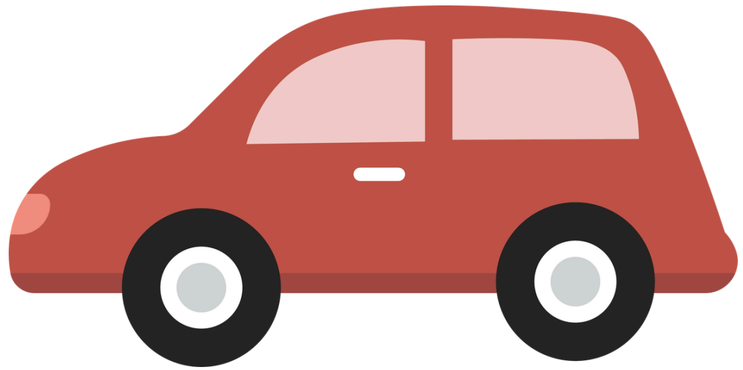
Smith College Capstone | 2026

IN-CONTEXT LEARNING: TEACH, DON'T TRAIN

Kat Brady, Kimberly By Goytia, Yelena Davidson, Hau Phan, Nicole Sanchez Flores, and Lucia Qin

ROAD MAP

- Project Overview
- Literature Review
- Turn-Based
- Parts
- Relationships
- Areas for Future Research
- Conclusion
- Thank You



PROJECT OVERVIEW

Prior research on in-context learning and context engineering suggests that LLMs can adapt from examples in context, but questions remain about how reliably they generalize rules in difficult cases.

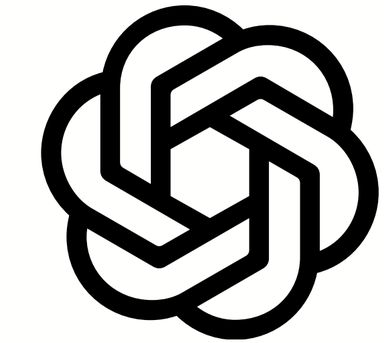
- Examined rule-application failures across GPT-5, GPT-5-mini, and GPT-5-nano.
- Developed three test-case domains: Relationships, Parts, and Turn Based Prompting.
- Evaluated how models performed under ambiguous, messy, or multi-step rule conditions.
- Tracked failure patterns using structured pod-specific metrics and Langfuse-based logging with support from OpenCode.

LITERATURE REVIEW

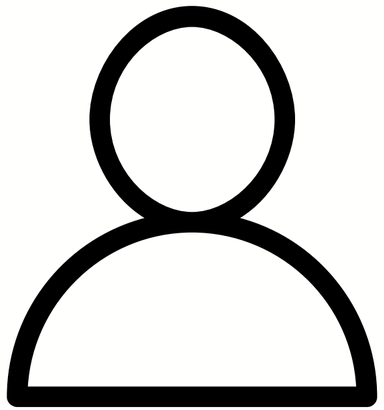
- Large language models can learn from few-shot examples in context rather than retraining.
  - Few-shot learning: giving the model a few examples in context so it can infer the pattern and perform a new task.
- In-context learning may reflect temporary algorithmic learning during inference (mimicry).
- Context engineering improves reliability by organizing prompts, memory, and retrieved information more effectively.
- RAG shows that external retrieval can yield more factual, grounded outputs from tested models.
  - RAG (Retrieval-Augmented Generation): improving responses from models by retrieving relevant outside information before giving output.

TURN-BASED PROMPTING

TURN BASED PROMPTS: THE ISSUE



OpenAI



Main Question: Can an LLM follow turn-based rules over time? If not, what challenges does the model faces? How can we improve it?

METHODS

Why Truth or Dare?

- Simply intuitive, captures the issue, and easy to scale in complexity

Set Up

- Turn-based game (alternating responses)

What we test:

- Turn-taking
- Memory of prior turns
- Rule adherence

Rules

- Only respond when it's your turn
- Maintain correct sequence
- Do not initiate out of turn

EVALUATION

Metrics

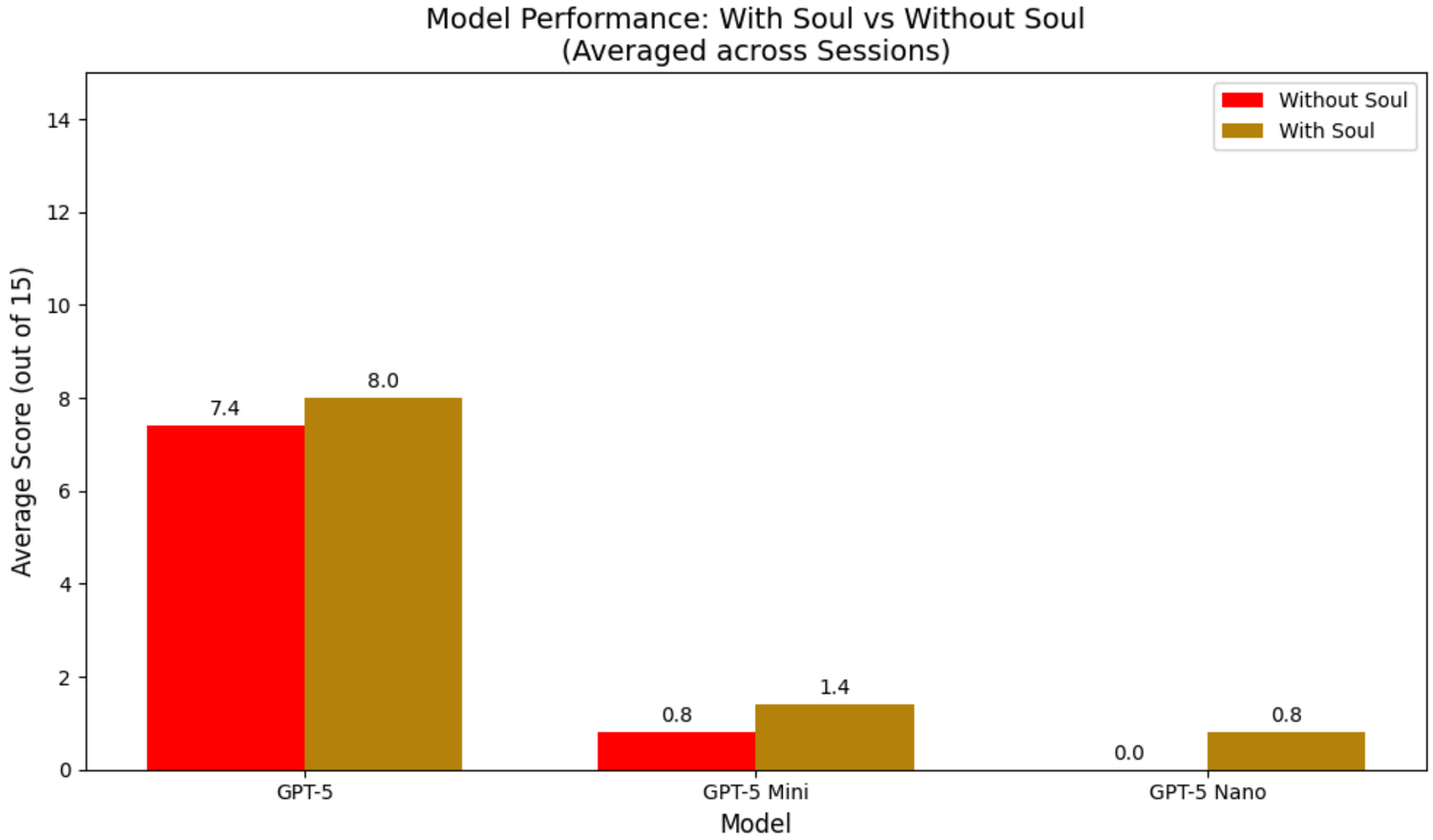
- Accuracy → % correct responses
- Time-to-Failure → How many turns before breakdown

Adding complexity

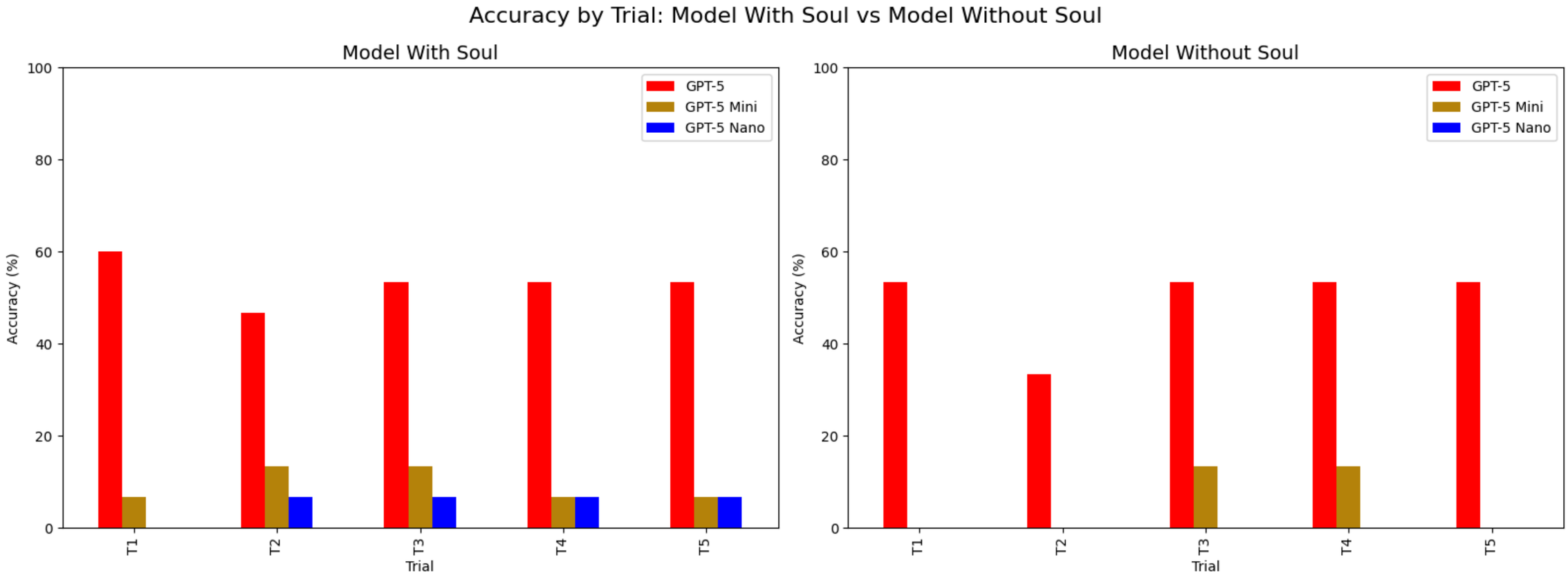
- Soul.md
- Different models (Chat-gpt-mini, nano etc)

RESULTS

- Overall, each models slightly better with soul.md for alternating turns when asked truth or dare.



RESULTS


Accuracy by Trial: Model With Soul vs Model Without Soul
Model With Soul
Model Without Soul
GPT-5
GPT-5 Mini
GPT-5 Nano
Accuracy (%)
Trial
T1
T2
T3
T4
T5
0
20
40
60
80
100

RESULTS SUMMARY

Quantitative Findings

- As expected, GPT-5 performed the best
- The smaller the models we used, the worse the performance.
- Ultimately, all models performed slightly better with soul.md

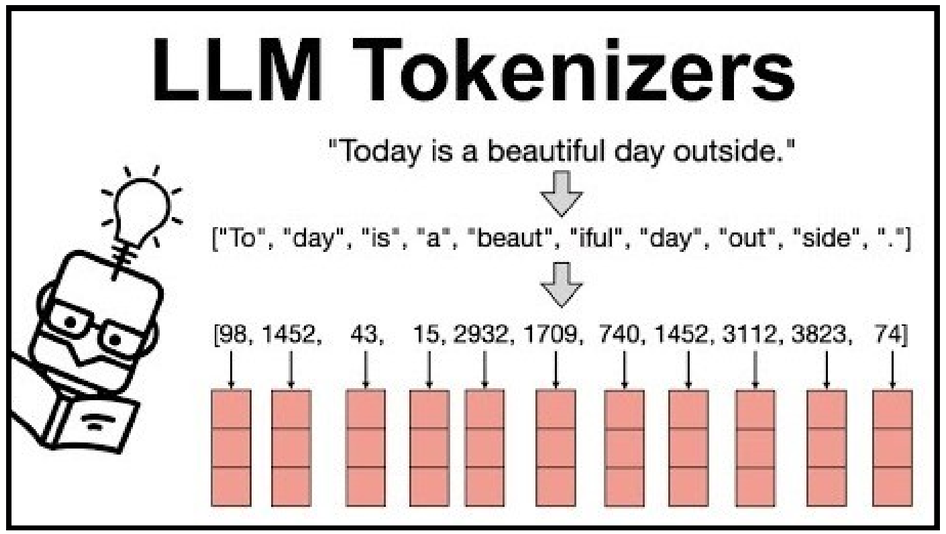
Qualitative Findings

- Models tend to disclose truth or dare questions before asking the user.
- Models forgets whose it is once prompt to answer a truth or dare question or when user chose dare.
- The word "truth" is a trigger word.
- Soul.md gives the model a sense of agency

PARTS

THE TOKENIZATION PROBLEM

LLMs process prompt inputs by breaking them down into tokens. This does not work well when the LLM is asked to specifically look at the **parts** of the prompt.



METHODS

Prompting

Each model was asked to operate as a rule learning system, given examples of sentences before and after the rule was applied, given a sentence to then apply the rule to, and asked to guess the rule and transform the sentence.

Prompt Types

- Substitutions
- Palindromes

METHODS: SUBSTITUTIONS

Design

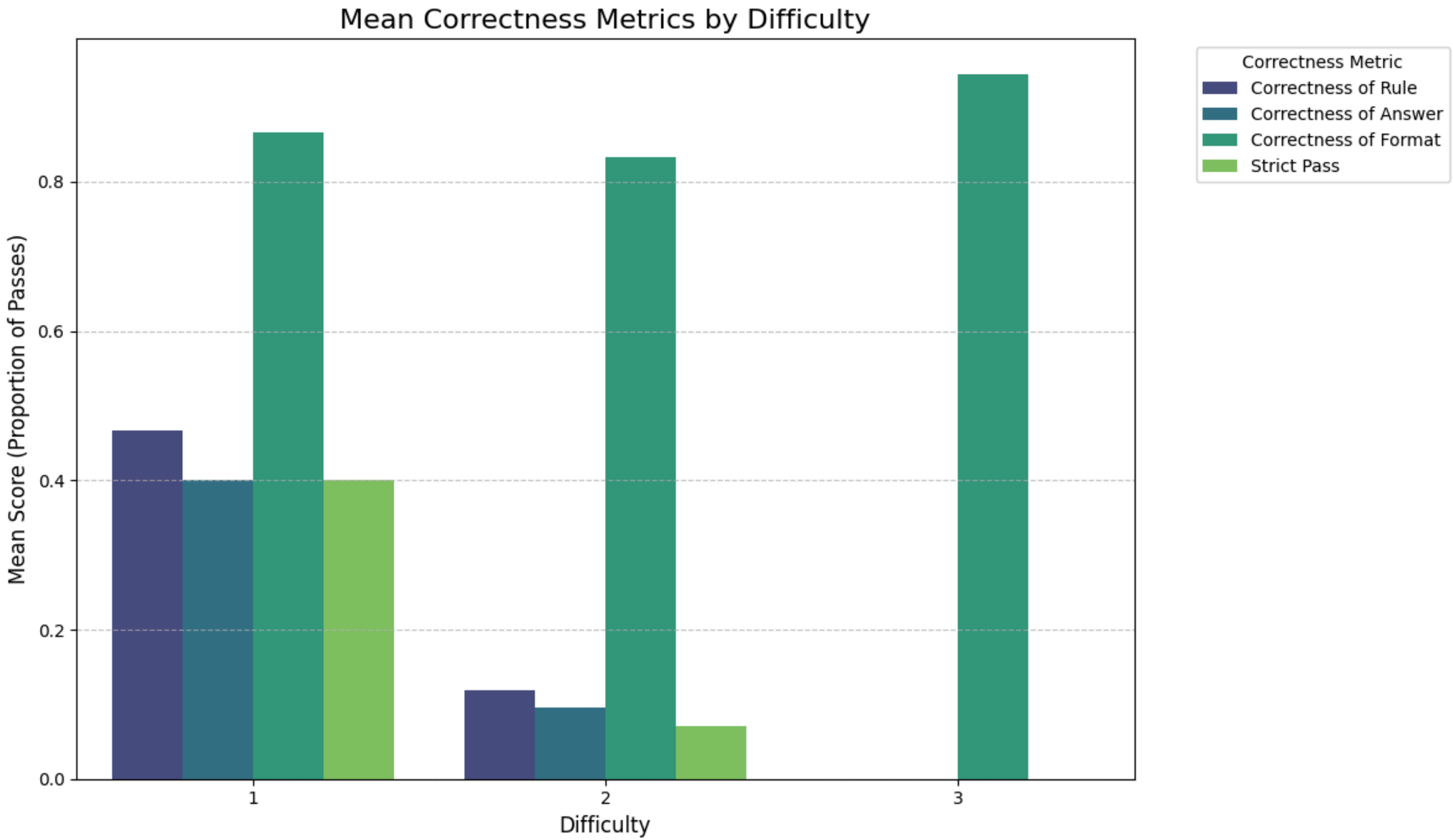
- 3 models each provided 25 prompts containing 5 examples of a rule
- Rules rated on 3 different difficulty levels
  - Level 1: One simple alteration
  - Level 2: Two simple alterations or one more complex alteration
  - Level 3: Two or more complex alterations

Metrics

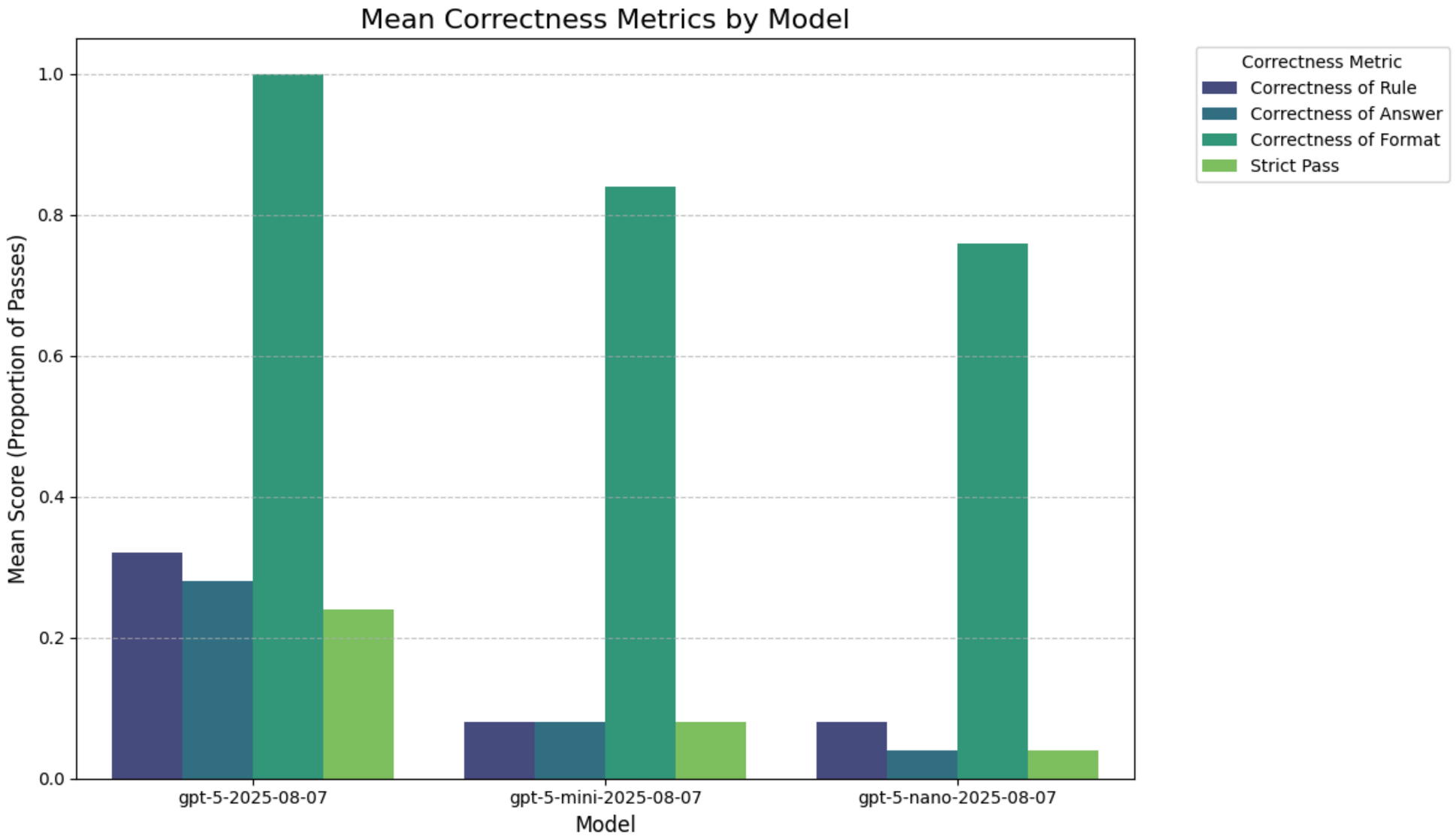
Scored on:

- Correctness of rule
- Correctness of answer
- Correctness of formatting
- Overall correctness
- Strict pass
- Certainty

RESULTS


Mean Correctness Metrics by Difficulty
Mean Score (Proportion of Passes)
0.0
0.2
0.4
0.6
0.8
1
2
3
Difficulty
Correctness Metric
Correctness of Rule
Correctness of Answer
Correctness of Format
Strict Pass

RESULTS


Mean Correctness Metrics by Model
Correctness Metric
Correctness of Rule
Correctness of Answer
Correctness of Format
Strict Pass
Mean Score (Proportion of Passes)
1.0
0.8
0.6
0.4
0.2
0.0
gpt-5-2025-08-07
gpt-5-mini-2025-08-07
gpt-5-nano-2025-08-07
Model

METHODS: PALINDROME

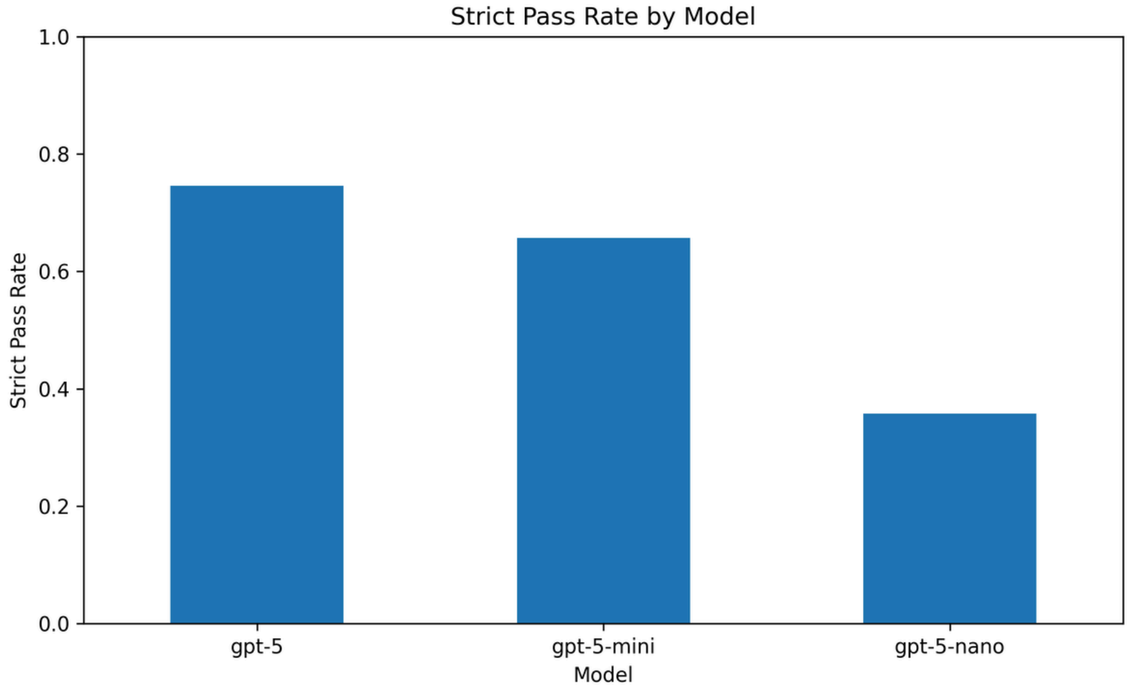
Experimental Design

- Input: demonstrations (input → output) + test input
- Output: RULE:... + ANSWER:... OR "Cannot be determined."
- 10 prompt templates → 2 regimes (Certain VS. Ambiguous)
- 201 total trials (67 per model)

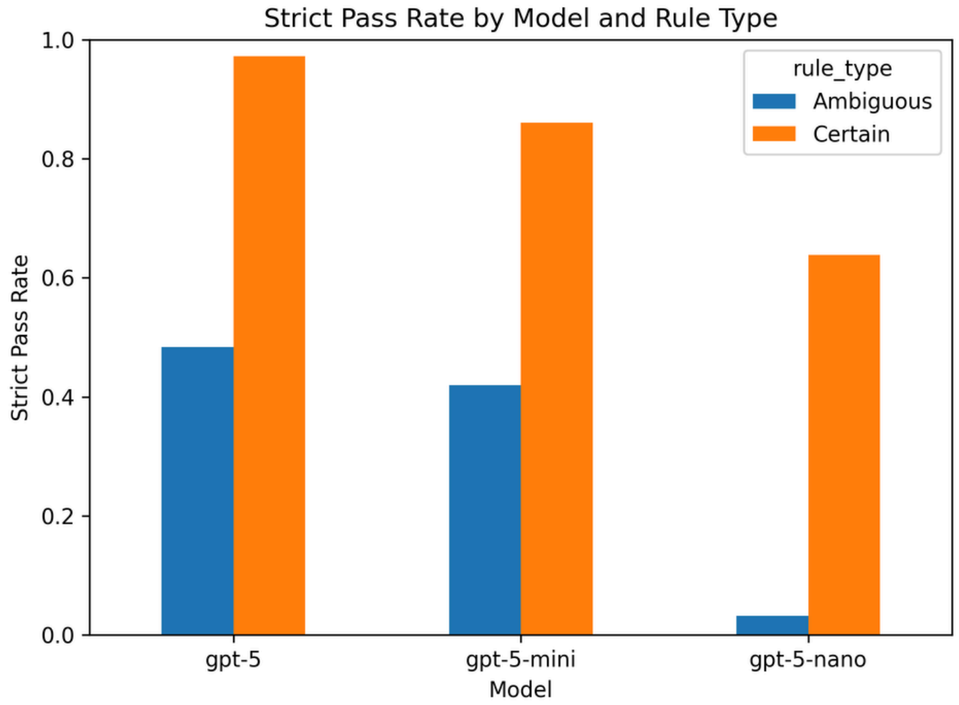
Metrics

- Format correctness: valid structure
- Decision correctness: correct choice (rule vs ambiguity)
- Rule correctness (certain only): identifies reversal
- Answer correctness: exact string match

RESULTS: PALINDROMES

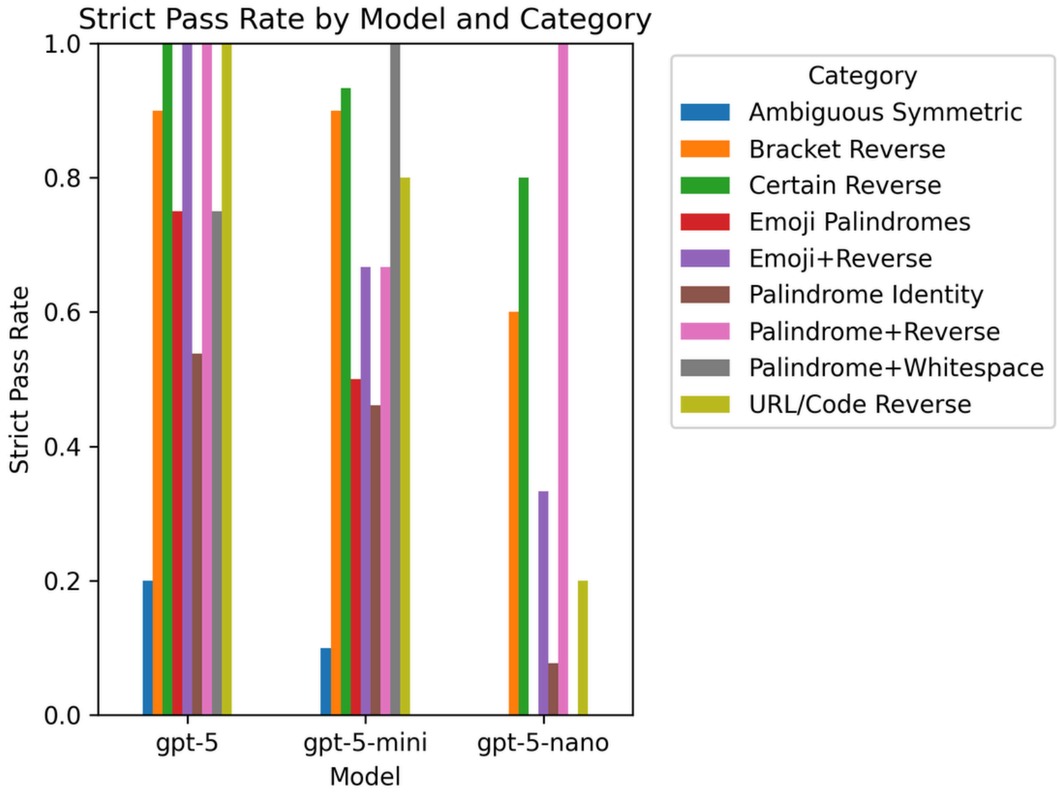


- GPT-5: 75% (50/67)
- GPT-5-mini: 66% (44/67)
- GPT-5-nano: 36% (24/67)



- Certain
  - GPT-5: 97%
  - GPT-5-mini: 86%
  - GPT-5-nano: 64%
- Ambiguous
  - GPT-5: 48%
  - GPT-5-mini: 42%
  - GPT-5-nano: 3%

RESULTS: PALINDROMES

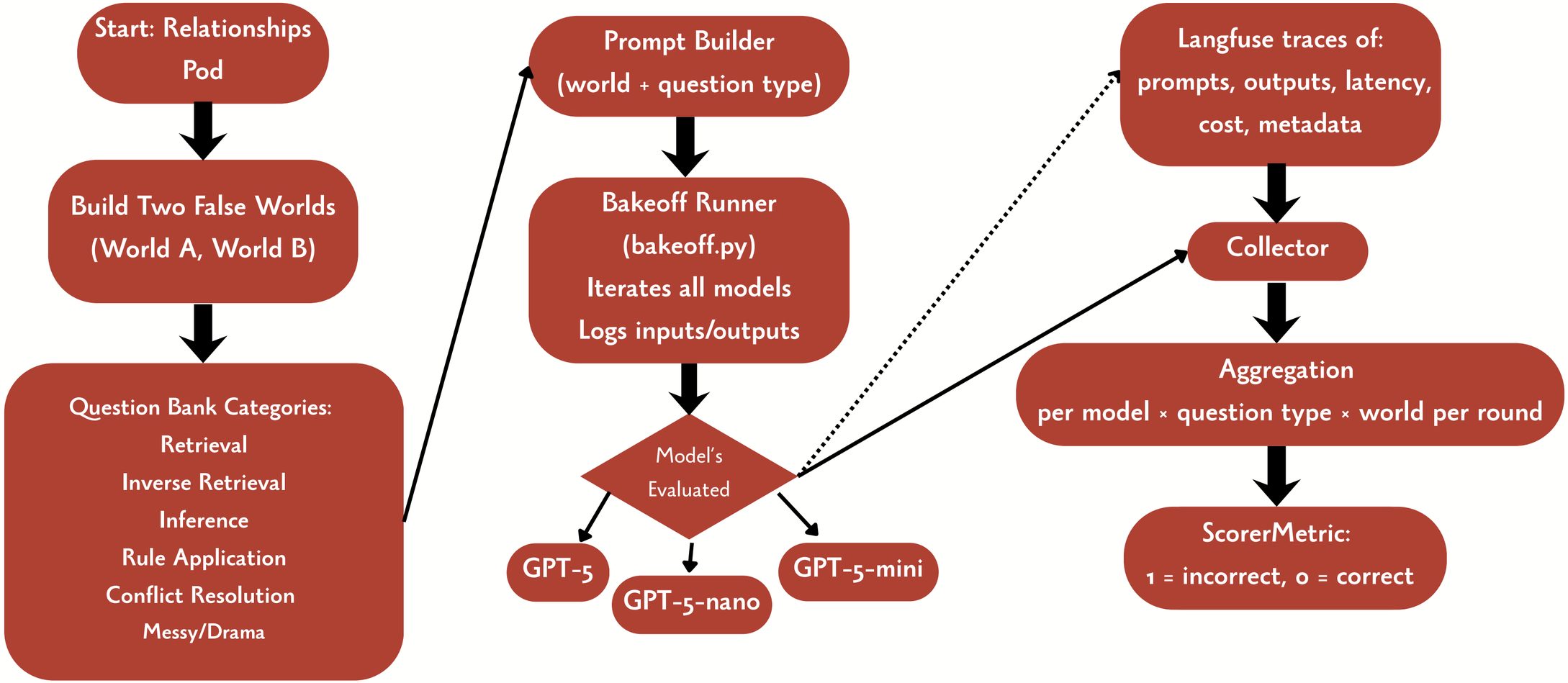


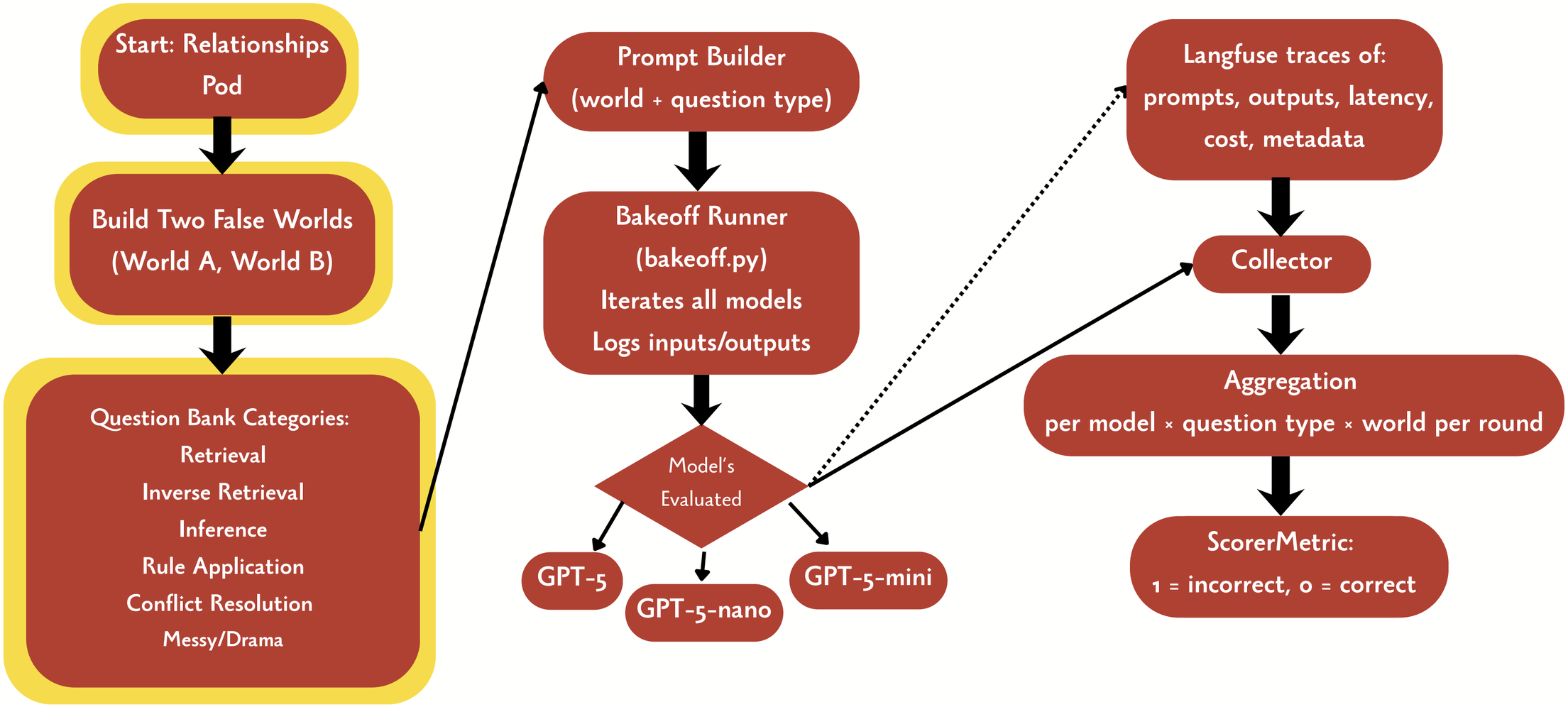
- More commitment when: Inputs look like variables / code
- More abstention when:
  - Inputs give mixed cues
  - Or are unfamiliar (emojis!)

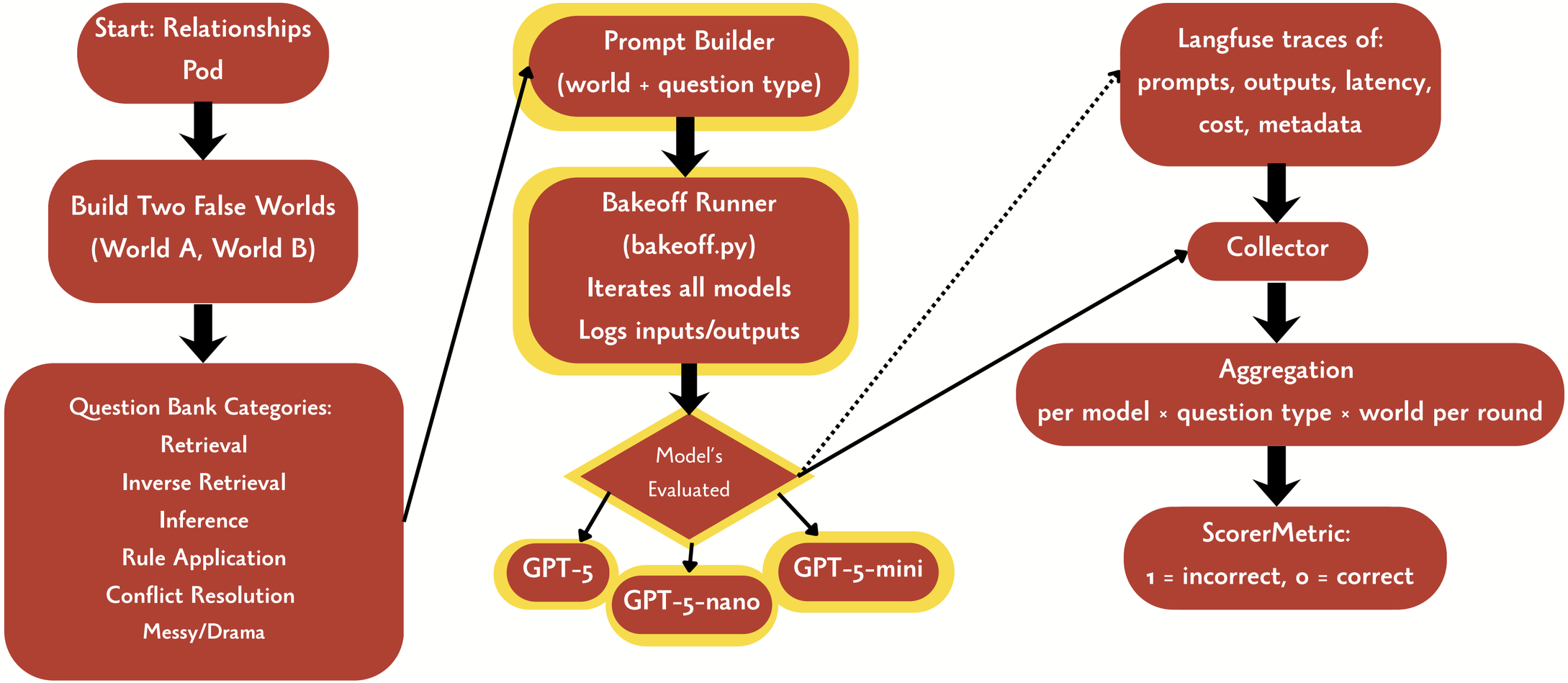
CONCLUSIONS

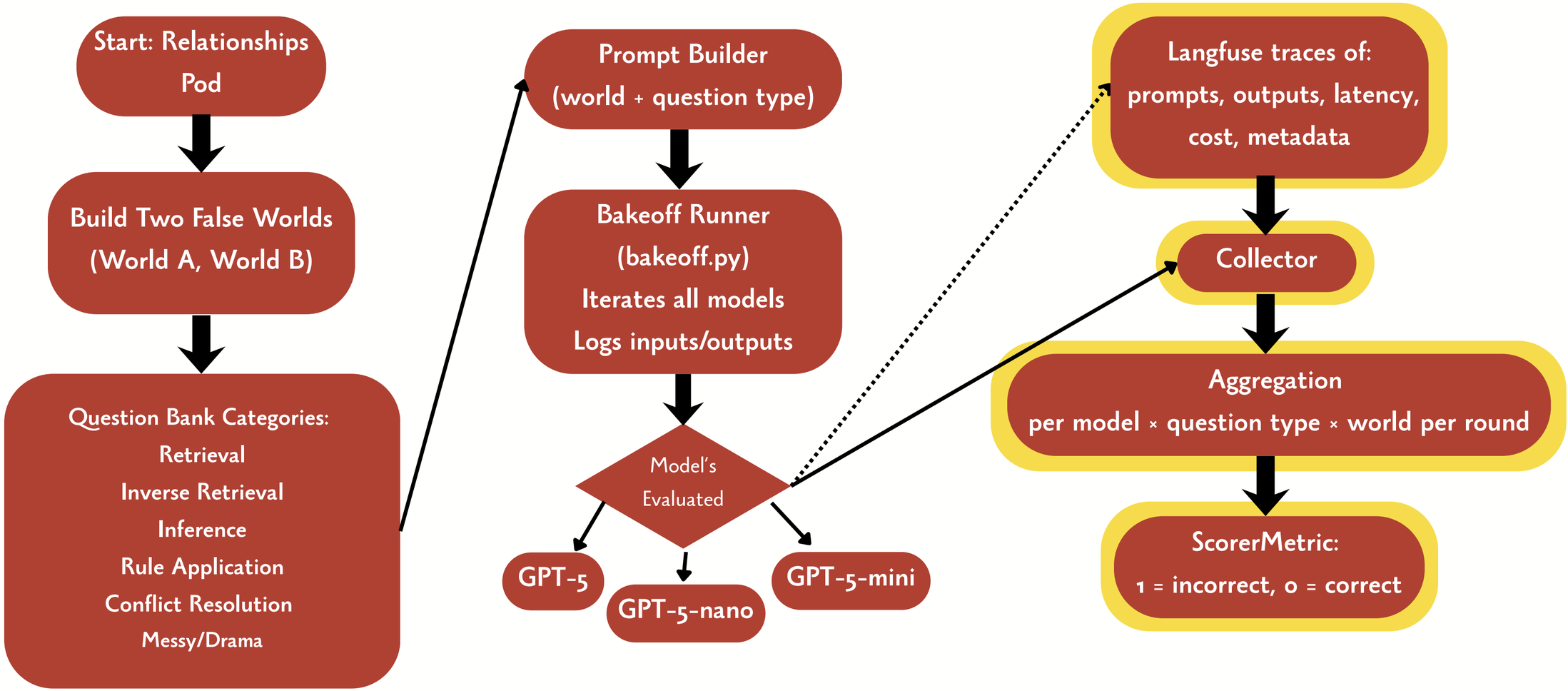
Across both types of experiments, GPT-5 was found to perform the best. All models struggled with ambiguous or high difficulty questions. This suggests something is different about how GPT-5 is able to tokenize and process prompts.

RELATIONSHIPS

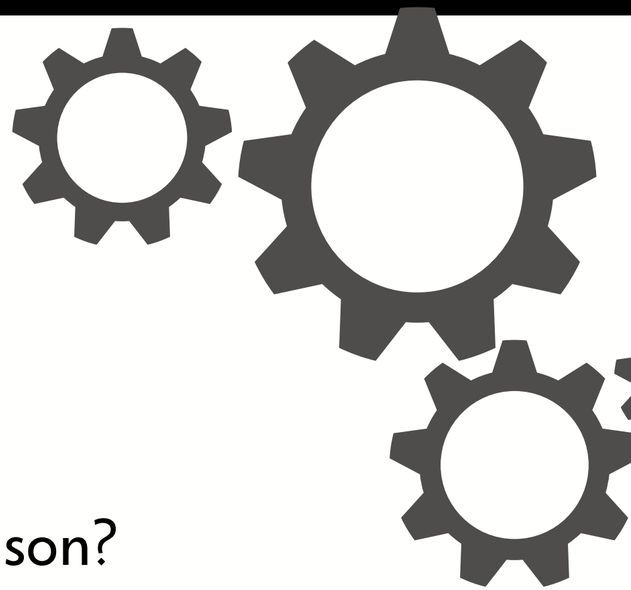

Start: Relationships Pod
Build Two False Worlds (World A, World B)
Question Bank Categories:
Retrieval
Inverse Retrieval
Inference
Rule Application
Conflict Resolution
Messy/Drama
Prompt Builder (world + question type)
Bakeoff Runner (bakeoff.py)
Iterates all models
Logs inputs/outputs
Model's Evaluated
GPT-5
GPT-5-nano
GPT-5-mini
Langfuse traces of: prompts, outputs, latency, cost, metadata
Collector
Aggregation
per model × question type × world per round
ScorerMetric:
1 = incorrect, 0 = correct


Start: Relationships Pod
Build Two False Worlds (World A, World B)
Question Bank Categories:
Retrieval
Inverse Retrieval
Inference
Rule Application
Conflict Resolution
Messy/Drama
Prompt Builder (world + question type)
Bakeoff Runner (bakeoff.py)
Iterates all models
Logs inputs/outputs
Model's Evaluated
GPT-5
GPT-5-nano
GPT-5-mini
Langfuse traces of: prompts, outputs, latency, cost, metadata
Collector
Aggregation
per model × question type × world per round
ScorerMetric:
1 = incorrect, 0 = correct


Start: Relationships Pod
Build Two False Worlds (World A, World B)
Question Bank Categories:
Retrieval
Inverse Retrieval
Inference
Rule Application
Conflict Resolution
Messy/Drama
Prompt Builder (world + question type)
Bakeoff Runner (bakeoff.py)
Iterates all models
Logs inputs/outputs
Model's Evaluated
GPT-5
GPT-5-nano
GPT-5-mini
Langfuse traces of: prompts, outputs, latency, cost, metadata
Collector
Aggregation
per model × question type × world per round
ScorerMetric:
1 = incorrect, 0 = correct


Start: Relationships Pod
Build Two False Worlds (World A, World B)
Question Bank Categories:
Retrieval
Inverse Retrieval
Inference
Rule Application
Conflict Resolution
Messy/Drama
Prompt Builder (world + question type)
Bakeoff Runner (bakeoff.py)
Iterates all models
Logs inputs/outputs
Model's Evaluated
GPT-5
GPT-5-nano
GPT-5-mini
Langfuse traces of: prompts, outputs, latency, cost, metadata
Collector
Aggregation
per model × question type × world per round
ScorerMetric:
1 = incorrect, 0 = correct

MAIN PROBLEM

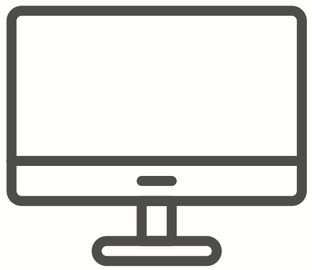


- LLMs struggle to understand human relational chains
  - Who is Tom Cruise's mother? vs. Who is Mary Lee Pfeiffer's son?
- LLMs struggle to understand messy or conflicting human relationship dynamics
  - A LLM could not clear your email inbox because it does not understand your realtionship to people who you are emailing

METHODOLOGY

5x5x3 Prompting Syle

- Asked questions relating to Scenerio 1 or Scenerio 2
- Asked 5 questions of the same type per round for five rounds



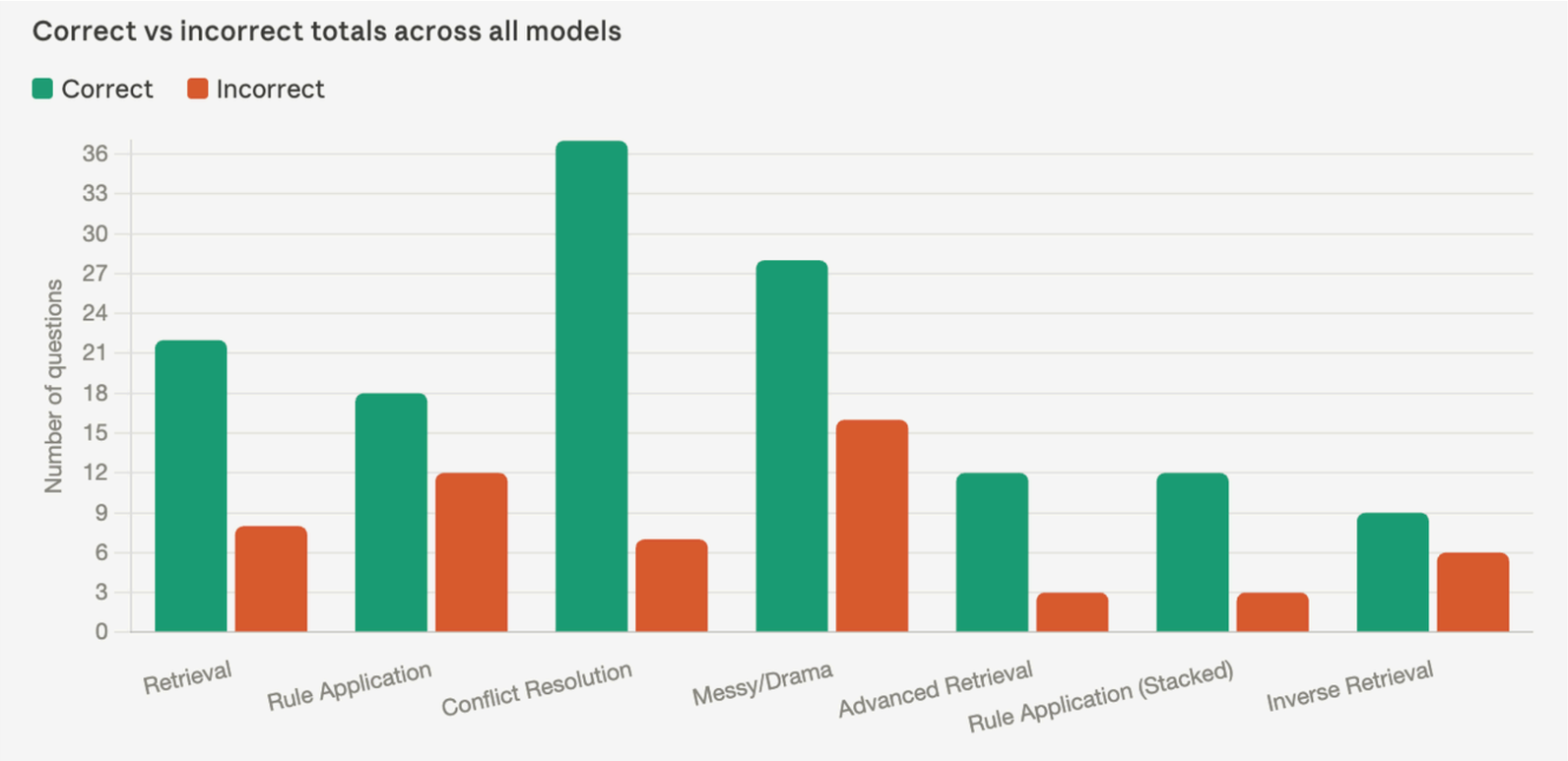
Metrics

- Quantitative:
  - Correct (0/1)
  - Question Type
- Qualitative:
  - Common reasoning errors or misunderstandings

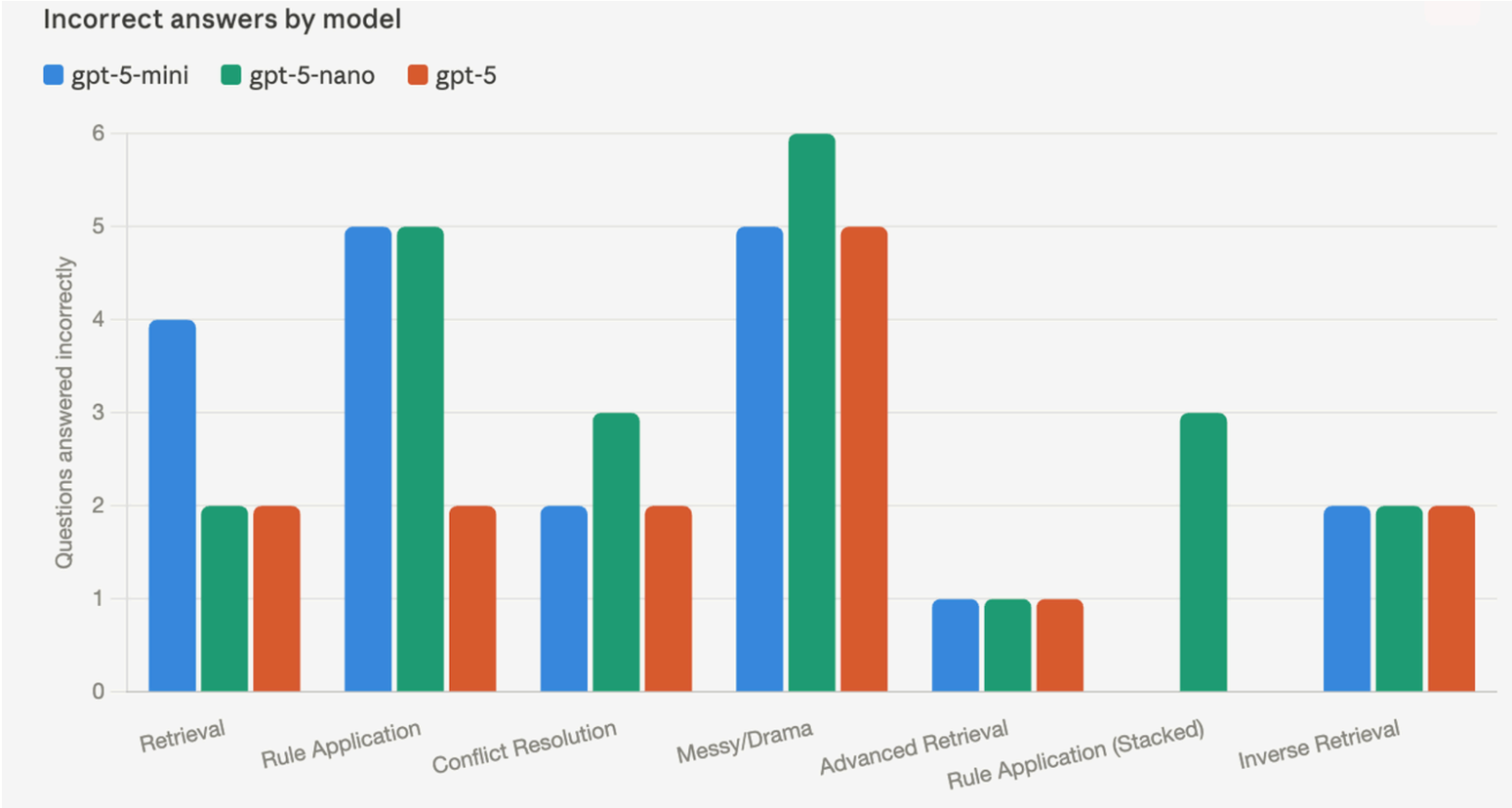
Question Type

- Retrieval
- Inverse Retrieval
- Inference/Relational Chain
- Rule Application
- Conflict Resolution
- Messy/Drama Scenerios

RESULTS: ACROSS MODELS



RESULTS: BETWEEN MODELS



RESULTS: SUMMARY

Overall all models preformed well when asked basic questions that align struturally with the fact that are provided. Model success rates for retrieval question in Scenario 1 was 93%. GPT-5 was the best performing model.

Quantitative Findings

- GPT-5 had the highest accuracy for more challenging questions
- Average model success rate of:
  - 67% for all models
  - 79% for Scenario 1
  - 75% for Scenario 2

Qualitative Findings

- Main Gaps in Model reasoning
  - Sequential logic
  - Making judgement calls when conflicting rules
  - Assigning blame to multiple characters
  - Past vs. Present in relationships

OVERALL NEXT STEPS

Turn-Based

- Tracking → Counting how many truths/dares
- Avoiding the word "truth"

Relationships

- Sequential Hierarchy
- RAG → Organizing information for each person

Parts

- Focusing on specific characters, and somehow change tokenization

REFERENCES

- Brown et al. (2020), Language Models are Few-Shot Learners
- Akyürek et al. (2023), What Learning Algorithm is In-Context Learning? Investigations with Linear Models
- Mäkinen (2025), Context Engineering for AI-Assisted Software Development
- Lewis et al. (2020), Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
- Dherin, B., Munn, M., Mazzawi, H., Wunder, M., & Gonzalvo, J. (2025). Learning without training: The implicit dynamics of in-context learning. arXiv.

THANK YOU

FOR YOUR ATTENTION