Can LLMs translate SATisfactorily?

Assessing LLMs in Generating and Interpreting Formal Specifications

Rushang Karia, Daksh Dobhal, Daniel Bramblett, Pulkit Verma, Siddharth Srivastava

TL;DR

LLMs cannot translate formal syntax* but we can **now** automatically assess them! *yet

Motivation

- Use of LLMs to translate/interpret formal syntax is increasing
- McKinsey revised 50% automation estimates by a decade

Cognition

Introducing Devin, the first AI software engineer

"Kids shouldn't learn to code"
- Jensen Huang, NVIDIA CEO

WORLD GOVERNMENT SUMMIT, DUBAI, 2024

Widespread adoption: PRINCETON UNIVERSITY cādence Microsoft . . .

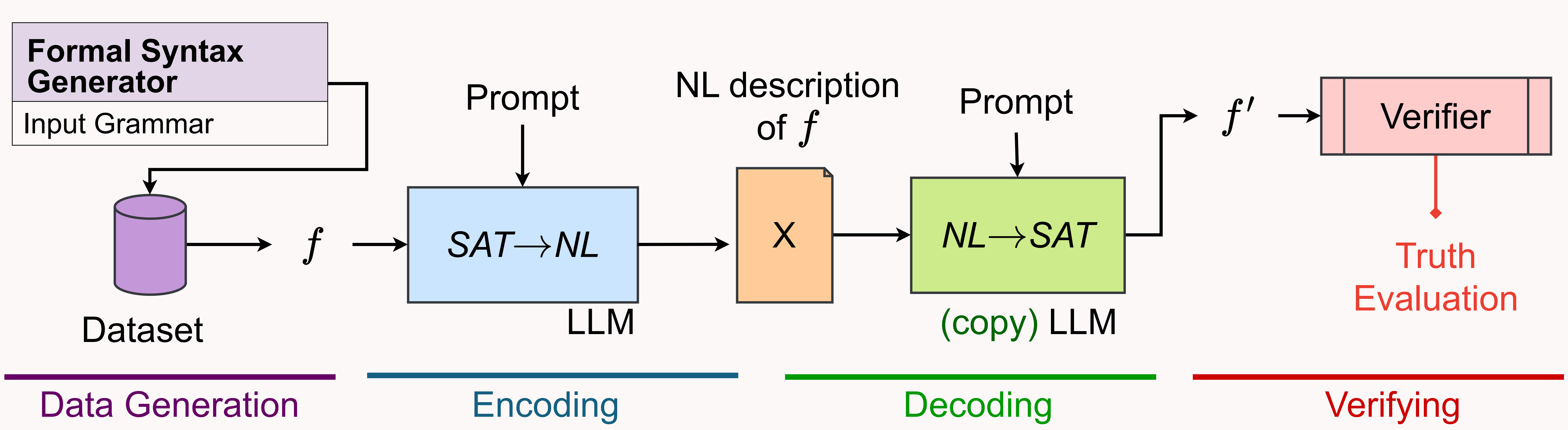
Key Challenges

How do we verifiy the translation capabilities of LLMs?

- How to efficiently generate new OOD data as LLMs evolve?
- How to annotate ground-truth data effectively?
- How to be robust in lieu of LLM hallucinations?

Can we make the pipeline automatic and human-independent?

Our Approach $NL \leftrightarrow SAT$



Is scalable: Uses syntax generators to scale datasets
Is handsfree: Uses two copies of an LLM to perform translation
Is robust: Uses external verifiers to validate the results

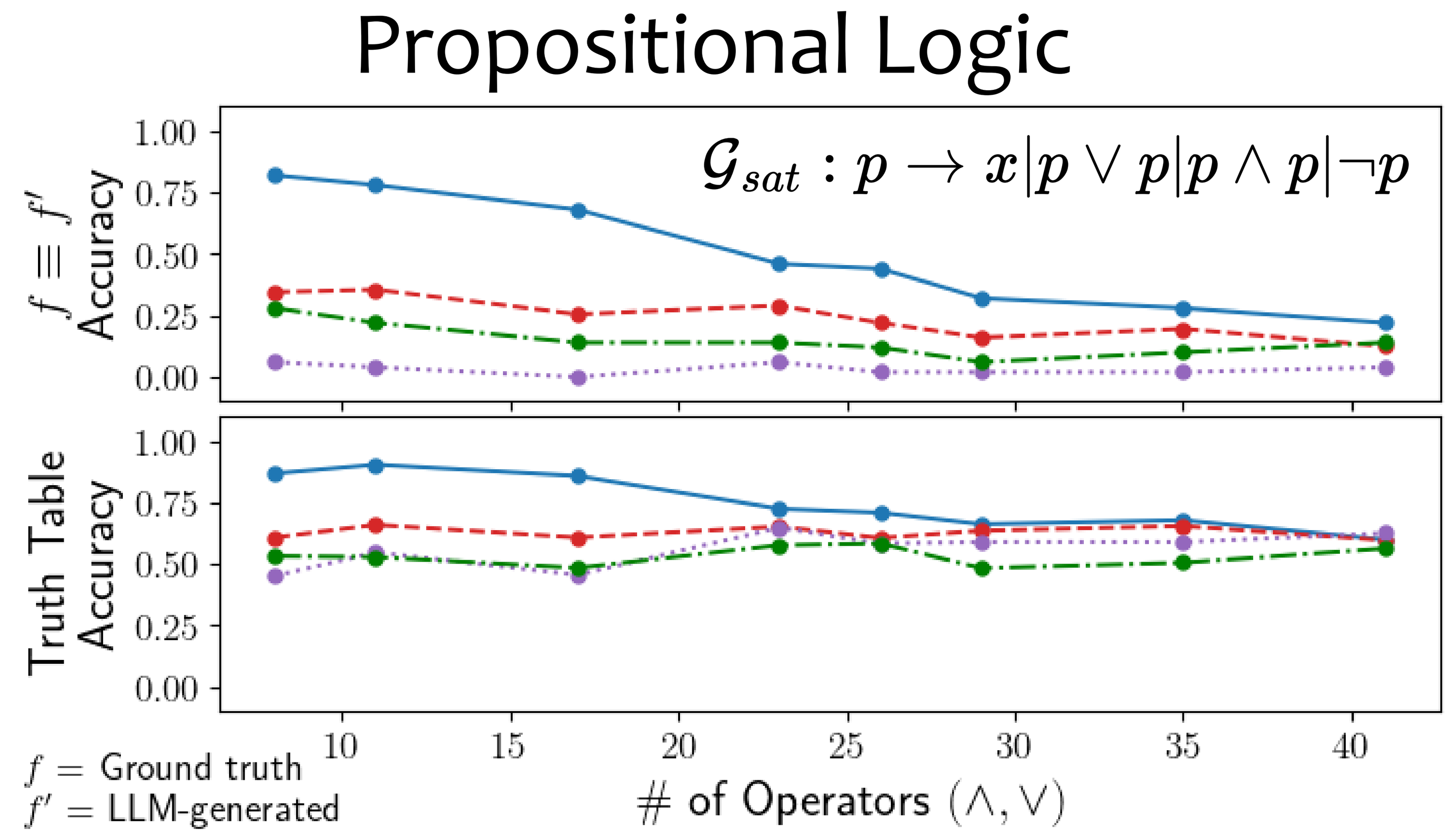
Results

Higher values better

Legend: GPT-4, GPT-3.5-turbo, Mistral, Gemini

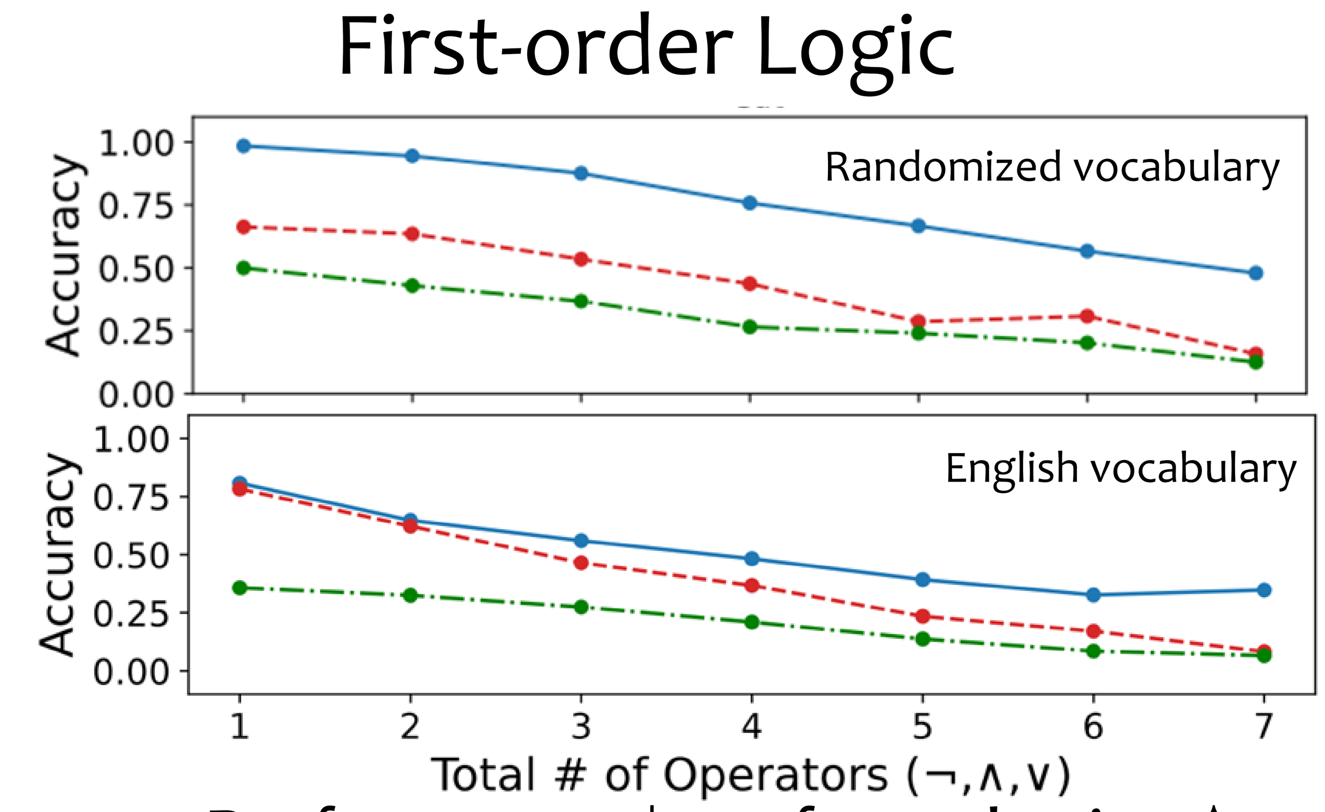
Prompt tuning: $\geq$ 95% accuracy on k-SAT

Propositional Logic



- Performance ↓ as formula size ↑
- LLMs often misplace parentheses
- LLMs often hallucinate propositions

First-order Logic



- Performance ↓ as formula size ↑
- Worse than $\mathcal{G}_{sat}$ on smaller inputs
  - Similar failures as $\mathcal{G}_{sat}$
  - Quantifiers often misplaced
- Using English is worse

Work performed at: AAIR Autonomous Agents and Intelligent Robots ASU | Grant Ack: NSF | AAAI-24 Summer Symposium