Generative AI with Logical Reasoning

Kalyan Krishnamani
NVIDIA, Santa Clara, CA
kalyans@nvidia.com

Abstract

The proposed system infuses logical reasoning into Generative AI systems. It is extremely valuable in application domains that are regulated - finance, healthcare, etc., where *explainability* is a critical requirement. It involves (a) encoding natural language statements as logical formulas, (b) solving the logical formulas using a solver based on mathematical logic, and (c) encoding the proof results back to natural language. (a) and (c) are done using an accessible LLM. A prototype tool, the `explain` system, implementing the proposed technology, is also presented.

1 Introduction

Trained on massive datasets, LLMs seem to encode all of the knowledge needed to answer user queries, by predicting the next tokens reasonably accurately. However, today's Generative AI systems struggle to reason consistently - *reason* in a logical sense - despite their impressively correct answers.

The inability to reason consistently has serious implications when used in critical applications such as healthcare, finance, critical software development, etc., where one needs to know a logical explanation for the answer. There is an imperative need to have *real* reasoning capabilities, as opposed to *seemingly real* emergent behaviors that stem from training on a massive size and variety of data.

2 Generative AI with Logical Reasoning

The proposed technology infuses *real* logical reasoning capabilities into Generative AI systems and is implemented in our prototype tool - `explain` system. It involves fine-tuning an LLM using a hand-crafted dataset (⓪ in Figure 1) of statements in natural language and the corresponding logical formula pairs. Such a reasonably fine-tuned LLM can consume natural language text along with a user query and generate the corresponding axioms and conjectures (logical formulas) respectively. The user inputs - context text and query are marked ① in Figure 1. These formulas are submitted to a solver that is based on mathematical logic (e.g. theorem provers, SMT solvers). The solver checks if the conjecture can be proved from the set of axioms. The proof, that the conjecture is true or false, is a set of logical formulas - derivations from the axioms applying logical deduction and simplification rules. A second LLM translates this proof to an explanation in natural language, shown by ② in Figure 1. The logical deduction proof steps from the logical solver are saved for review and serves as a proof certificate. Optionally, a Retrieval Augmented Generation (RAG) step, depicted ③ could be used to augment our system. (3a) and (3b) help in generating logical encoding of natural language and natural language description of proofs respectively.

While the system generates formulas in first-order logic, it could be extended to generate higher-order logic formulas if needed.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

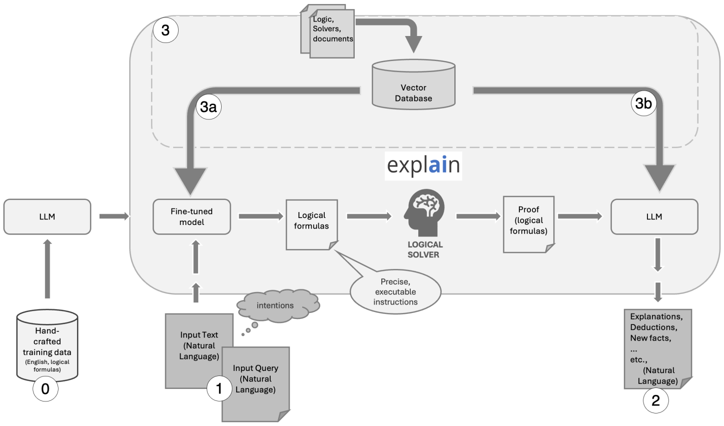

Figure 1: The `explain` System

3 Results

As an example, consider the following puzzle that requires logical reasoning which is not straightforward for the best available LLMs today (GPT-4o, Claude 3.5 Sonnet).

> *Someone who lives in Igloo killed Alice. Alice, Bob, and Charles live in Igloo, 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 Alice hates. Alice hates everyone except Bob. Bob hates everyone not richer than Alice. Bob hates everyone Alice hates. No one hates everyone. Alice is not Bob.*

Query: *Did Alice kill herself?*

`GPT-4o` claims *Alice did not kill herself* while `Claude-3.5-Sonnet` claims *Bob killed Alice*, both of which are misleading. Further, the results are not reproducible consistently. The `explain` system, by using logical formulas and mathematical solving, is able to resolve that *Alice killed herself*.

Links to screenshots of the runs using different LLMs are provided for reference - GPT-4o, Claude-3.5-Sonnet and explain.

4 Challenges and Future Work

There are still some challenges with the `explain` system. First is the *Frame problem* [1], due to which axioms have to be explicitly added even for the facts that are not explicitly stated. A second challenge, related to the first, is the loss of semantic information. For example "Alice *lives* in New York City" and "Alice *spends time* in Big Apple" may be closer concepts in embedding space while in the domain of logical formulas, they would be two different functions each evaluating to a Boolean value. One has to discover these, i.e. add explicit axioms that convey "New York City" and "Big Apple" are the same and so are logical encodings for "lives" and "spends time". An additional layer, possibly looking at semantic distances and adding such axioms, to resolve such ambiguities while moving from the realm of natural language to logical domain is possibly needed.

Knowledge Graphs [2] and ontology matching are being explored to augment the fine tuning phase of the `explain` system to handle these challenges.

5 References

[1] The Frame Problem. `https://plato.stanford.edu/entries/frame-problem/`

[2] Yasunaga, M., Bosselut, A., Ren, H., Zhang, X. Manning, C.D., Liang, P. & Leskovec, J. *Deep Bidirectional Language-Knowledge Graph Pretraining.*, NeurIPS 2022.