MEDIATE: A Modular Environment for Development and Insights in Automated Theorem Experiments

Written by AAAI Press Staff[1]*
AAAI Style Contributions by Pater Patel Schneider, Sunil Issar,
J. Scott Penberthy, George Ferguson, Hans Guesgen, Francisco Cruz†**, Marc Pujol-Gonzalez**†

[1]Association for the Advancement of Artificial Intelligence
1900 Embarcadero Road, Suite 101
Palo Alto, California 94303-3310 USA
proceedings-questions@aaai.org

Abstract

Recent interest in Artificial Intelligence for Theorem Proving (AITP) has given rise to a plethora of benchmarks and methodologies, particularly in the area of Interactive Theorem Proving (ITP). Research in the area is fragmented, with a diverse set of approaches being spread across several ITP systems. This presents a significant challenge to the comparison of methods, which are often complex and difficult to replicate. To address this, we present MEDIATE: A Modular Environment for Insights and Development in Automated Theorem Experiments. By separating the learning approach from the data and ITP environment, MEDIATE allows for the fair and efficient comparison of approaches between systems. MEDIATE is designed to seamlessly incorporate new systems and benchmarks, currently integrating the HOList, HOLStep, MIZAR40, LeanStep, LeanGym and TacticZero benchmarks. We demonstrate the utility of MEDIATE through a study of embedding architectures in the context of the above systems. These experiments lay the foundations for future work in evaluation of performance of formula embedding, while simultaneously demonstrating the capability of the framework. We complement these with a qualitative analysis of embeddings within and between systems, illustrating that improved performance was associated with more semantically aware embeddings. By streamlining the implementation and comparison of Machine Learning algorithms in the ITP context, we anticipate MEDIATE will be a springboard for future research.

Introduction

Interactive Theorem Proving (ITP) is a key paradigm of formal verification. With a human providing high level proof guidance, ITP systems have been used to develop verified compilers (cite), formalise mathematical conjectures (cite) and develop provably correct microkernels (cite). Successful proof guidance requires proficiency in both the formal system and the application domain. This has limited the scale and widespread adoption of formal methods, with e.g... The nascent field of Artificial Intelligence for Theorem Proving (AITP) has the potential to address this, with several strong results in automating human ITP guidance. AITP research encompasses several mathematical reasoning tasks, such as math word problems (cite) and autoformalisation (cite). The focus of this paper is specifially AI for Interactive Theorem Proving, which we refer to as AI-ITP.

Current datasets and environments for AI-ITP are divided across several distinct ITP systems. Although this provides a variety of tasks for benchmarking, it poses a challenge to the fair and efficient comparison of the automation approaches. This is further complicated by the broad range of methods which have been applied in the area, which are generally tested only in the context of a single ITP system (cite). (examples, HOList over the HOL Light prover, LeanStep/LeanGym over Lean, TacticZero and TacticToe,.. with HOL4).

A central component of AI-ITP systems is their embedding model. Vector embeddings of the ITP logical expressions are required for the application of learning algorithms. These are used for tasks such as premise selection, goal selection which are essential for proof guidance.

Expressions are either treated as a natural language sequence, or as a directed graph derived from their abstract syntax tree representation. It has been argued that a graph representation is more appropriate, with a large body of work in AI-ITP using Graph Neural Network (GNN) as the embedding model to achieve strong results in several tasks.

However, Transformer models applied to the sequence representation have also demonstrated strong performance by leveraging large models pre-trained on NLP datasets.

Both architectures have fundamental limitations, as GNNs suffer from poor integration of global information, and vanlla Transformers ignore the structural information of the expression. Recent work has shown that combining both architectures leads to improved performance on several graph learning tasks. In the context of theorem proving however, there has been no thorough comparison between approaches.

We address these two issues by ... our contributions can be summarised as ...

*With help from the AAAI Publications Committee.

†These authors contributed equally.

Related Work

Unified toolkit proposed for MWP, MWPToolkit.

- However, AI for ITP has several additional considerations: - Fundamentally different learning approaches. Range from E2E RL, direct supervised learning, pretrained LLMs .. - Makes it difficult to be fully modular as there are some unavoidable dependencies between e.g. environment and model (e.g. HOL4 environment for TacticZero) - Interaction with an environment - Several axes of variation in algorithm design. Learning paradigm, proof search, embedding arch., action selection arch.

Note we restrict ourselves to the problem of automating the human interaction with ITP, through the setup in Figure x. Other approaches such as ML for Hammers, also promising

AIITP benchmarks mentioned, limited in single system. Other unified frameworks, e.g. NLP, CV, GNN (graph benchmarks), which have provided strong value to their respective areas

Background

ITP

Approaches

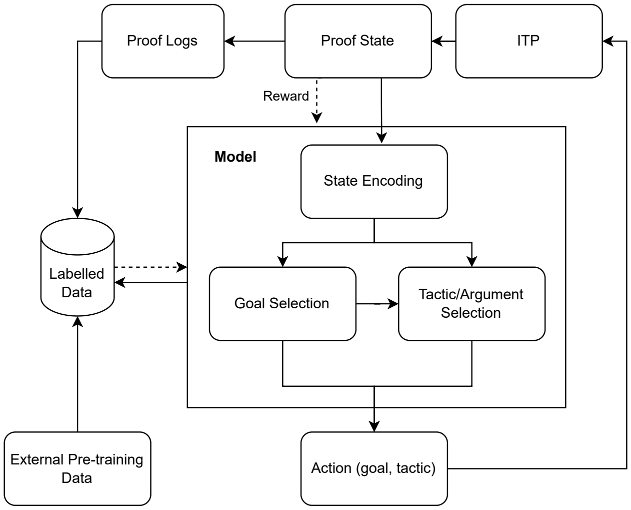

Figure 1: Caption for the figure.

Learning Approach RL (Policy Gradient) end to end, Supervised Training over labelled proof logs (tactic, premise, goal), Pre-training over proof data (PACT) Pre-training over NLP corpus

Proof Search BFS, Fringe, HyperTree MCTS

State Embedding Transformer, GNN, SAT, Directed SAT, Autoencoder

Tactic Selection Fixed set of tactics/arguments (TacticZero, HOList), generative model (GPT-f, Curriculum learning, lean)

Datasets and Benchmarks

Figure (Table): Benchmark vs Details (size, system, interactive) Highlight in bold what is currently included Include HOL4 premise selection HOList, LeanStep, HOLStep, MIZAR40, LeanGym, TacticZero, (CoqGym, IsarStep etc from survey)

As outlined in Figure x, benchmarks consider each system in isolation, and generally focus on a small set of learning and proof search approaches. Given the many differences between ITPs, this is unsurprising, as there is significant effort required to set up these systems for automated proving.

Embedding Architecture

Figure (Table): Interactive Approaches vs Details Proof search BFS, MCTS, Fringe Embedding architecture GNN, Transformer, Autoencoder Tactic/Argument selection Generative (lean, hypertree), Fixed set (HOL4, HOList) Learning approach RL (TacticZero), Supervised proof logs (HOList, Lean, Hypertree?)

MEDIATE

Diagram and explanation of system architecture

Open Source additions: LeanGym/LeanStep output s-expressions HOList using non-google tools, and independent of TF1 and ML framework

Curated, large dataset of varied ITPs

Embedding Experiments

Supervised

MIZAR, HOLStep, HOL4 pretrain, HOList Pretrain, Lean?

End to End

TacticZero, HOList

Qualitative Analysis

Discussion

Conclusion