Title

ConceptBridge: Accelerating Formal Proof Discovery through AI-Assisted Conceptual Interpolation

Problem Statement

Existing automated theorem provers often struggle with bridging large conceptual gaps between premises and conclusions, especially in domains requiring creative leaps or interdisciplinary connections. This limitation hinders the discovery of novel proofs and slows down mathematical progress.

Motivation

Current approaches typically rely on exhaustive search or language model generation, which can be inefficient for problems requiring key insights. Human mathematicians often make breakthroughs by identifying crucial intermediate concepts or analogies. An interactive system that proposes such conceptual bridges could dramatically accelerate proof discovery. By combining the strengths of large language models, reinforcement learning, and human expertise, we aim to create a more effective and intuitive proof assistant.

Proposed Method

We propose an interactive proof assistant called ConceptBridge that actively collaborates with human mathematicians. The system uses a large language model (LLM) fine-tuned on mathematical corpora to generate a diverse set of potentially relevant concepts, theorems, and analogies given the current proof state. These are presented to the user, who can select promising directions or provide feedback. The system then uses reinforcement learning to refine its concept proposal strategy based on user interactions. Additionally, we implement a novel 'conceptual interpolation' mechanism that attempts to smoothly connect disparate ideas in the proof by generating a sequence of intermediate bridging concepts. This is achieved using a variational autoencoder (VAE) trained on mathematical concept embeddings, allowing for semantically meaningful interpolation in latent space.

Step-by-Step Experiment Plan

Step 1: Data Preparation

Collect a diverse dataset of mathematical proofs from various fields (e.g., algebra, topology, analysis) from sources like arXiv and mathematical journals. Extract key concepts, theorems, and intermediate steps from these proofs to create a training corpus.

Step 2: LLM Fine-tuning

Fine-tune a pre-trained LLM (e.g., GPT-3 or GPT-4) on the mathematical corpus. Use prompt engineering to teach the model to generate relevant concepts, theorems, and analogies given a partial proof state.

Step 3: Concept Embedding

Train a VAE on the extracted mathematical concepts to create a latent space of concept embeddings. Ensure the latent space allows for meaningful interpolation between concepts.

Step 4: Conceptual Interpolation Mechanism

Implement the interpolation mechanism using the trained VAE. Given two concepts, generate a sequence of intermediate concepts by sampling points along the path in the latent space.

Step 5: User Interface Development

Create an interactive interface where users can input partial proofs, view suggested concepts, and provide feedback. Implement functionality for users to select promising directions and rate the usefulness of suggestions.

Step 6: Reinforcement Learning Integration

Implement a reinforcement learning algorithm (e.g., Proximal Policy Optimization) to optimize the concept proposal strategy based on user feedback. The state space should include the current proof state and user history, while the action space consists of concept suggestions.

Step 7: Baseline Implementation

Implement baseline methods for comparison, including: (a) traditional automated theorem provers, (b) LLM-based proof generation without conceptual interpolation, and (c) human-only proof development.

Step 8: User Study

Recruit 20-30 professional mathematicians for a user study. Provide them with a set of challenging theorems to prove using ConceptBridge and baseline methods. Collect data on time-to-proof, subjective ratings of system usefulness, and qualitative feedback.

Step 9: Benchmark Creation

Develop a benchmark set of 'insight-heavy' theorems from various mathematical fields. These should be theorems where identifying key intermediate concepts is crucial for the proof.

Step 10: Evaluation

Evaluate ConceptBridge and baselines on the benchmark set. Metrics should include success rate, proof length, and time-to-proof. For successful proofs, analyze the importance of suggested concepts in the final proof structure.

Step 11: Ablation Study

Conduct an ablation study to quantify the impact of different components (e.g., conceptual interpolation, reinforcement learning) on system performance.

Step 12: Analysis and Iteration

Analyze results from the user study, benchmark evaluation, and ablation study. Identify strengths and weaknesses of the ConceptBridge system. Iterate on the design based on these insights, potentially refining the LLM fine-tuning, VAE architecture, or reinforcement learning algorithm.

Test Case Examples

Baseline Prompt Input (Traditional Automated Theorem Prover)

Prove: For any continuous function f on [a,b], if f(a) < 0 and f(b) > 0, then there exists a c in (a,b) such that f(c) = 0.

Baseline Prompt Expected Output (Traditional Automated Theorem Prover)

- Assume f is continuous on [a,b]
- Assume f(a) < 0
- Assume f(b) > 0
- Define g(x) = f(x) for x in [a,b]
- g is continuous on [a,b]
- g(a) < 0 and g(b) > 0
- By the Intermediate Value Theorem, there exists c in (a,b) such that g(c) = 0
- Therefore, f(c) = 0
- QED

Proposed Prompt Input (ConceptBridge; Step 1: Concept Generation)

Given the theorem 'For any continuous function f on [a,b], if f(a) < 0 and f(b) > 0, then there exists a c in (a,b) such that f(c) = 0.', suggest relevant mathematical concepts, theorems, or analogies that might be useful for proving this statement.

Proposed Prompt Expected Output (ConceptBridge; Step 1: Concept Generation)

- Intermediate Value Theorem
- Continuity of functions
- Bolzano's Theorem
- Mean Value Theorem
- Analogy: Crossing a river (function crossing the x-axis)
- Concept of sign change
- Rolle's Theorem
- Extreme Value Theorem
- Bisection method (numerical analysis)
- Topology of the real line

Proposed Prompt Input (ConceptBridge; Step 2: Conceptual Interpolation)

Generate a sequence of intermediate concepts connecting 'continuity of functions' and 'Intermediate Value Theorem'.

Proposed Prompt Expected Output (ConceptBridge; Step 2: Conceptual Interpolation)

- Continuity of functions
- Preservation of limits under continuous functions
- Connectedness of intervals
- Image of connected sets under continuous functions
- Continuous image of $[a,b]$ is $[f(a), f(b)]$ or $[f(b), f(a)]$
- Existence of all intermediate values
- Intermediate Value Theorem

Proposed Prompt Input (ConceptBridge; Step 3: Proof Construction)

Using the generated concepts and interpolation, construct a proof for the theorem.

Proposed Prompt Output (ConceptBridge; Step 3: Proof Construction)

Proof:
- We are given that f is continuous on $[a,b]$, $f(a) < 0$, and $f(b) > 0$.
- By the concept of continuity, f preserves connectedness of intervals.
- The interval $[a,b]$ is connected.
- Therefore, the image of $[a,b]$ under f is also connected.
- Since $f(a) < 0$ and $f(b) > 0$, the image of $[a,b]$ under f must contain the interval $[f(a), f(b)]$.
- 0 is in the interval $[f(a), f(b)]$ because $f(a) < 0 < f(b)$.
- By the Intermediate Value Theorem, which states that a continuous function on a closed interval takes on all values between its values at the endpoints, there must exist a c in (a,b) such that $f(c) = 0$.
- Therefore, the theorem is proved. QED

explanation

ConceptBridge generates relevant concepts and performs conceptual interpolation, guiding the user towards key insights like the Intermediate Value Theorem. This approach provides a more intuitive and structured path to the proof compared to the traditional automated theorem prover, which jumps directly to applying the theorem without explaining the underlying concepts.

Fallback Plan

If ConceptBridge doesn't significantly outperform baselines, we can pivot to an analysis paper exploring why AI-assisted conceptual interpolation may not be as effective as hypothesized for formal proof discovery. We could investigate: (1) the quality and relevance of generated concepts, (2) the effectiveness of the conceptual interpolation in bridging gaps in mathematical reasoning, (3) how different types of mathematical problems benefit (or don't) from this approach, and (4) the cognitive load on users when interacting with the system. We could also explore alternative approaches, such as combining ConceptBridge with more traditional automated reasoning techniques or developing a hybrid system that leverages both AI-generated concepts and human-curated mathematical knowledge bases. Additionally, we could conduct a more in-depth study on how human mathematicians utilize conceptual analogies and interpolations in their reasoning process, which could inform future iterations of AI-assisted proof systems.

Ranking Score: 6