Experiments with Language Models for Isabelle Autoformalization

David Valente, Manuel Eberl, Cezary Kaliszyk, Josef Urban

AITP 2024

Motivation

- Formalization of mathematical theorems is crucial but time-consuming
- Learning-assisted autoformalization offers a promising solution
- Project goal: Finetune Phi-2 model for LaTeX to Isabelle translation
- Explore feedback loop with type-checking and theorem proving
- Consider adding RAG for more accurate use of AFP

Training Data

- Dataset: Pairs of natural language statements and Isabelle lemmas
- LaTeX representations generated using Mistral Large model
- Multiple LaTeX versions per statement for diversity
- Over 100,000 LaTeX-Isabelle lemma pairs in total

Example Data

Natural Language: If a set X is countable, then the cardinality of set X is less than or equal to Aleph null.

LaTeX: If a set X is countable, then $|X| \leq \aleph_0$.

Verbatim LaTeX:

```
If a set X is countable, then $|X| \leq \aleph_0$.
```

Isabelle Lemma:

```
    lemma countable_imp_g_le_Aleph0:
      "countable X \<Longrightarrow> gcard X \<le> \<aleph>0"
```

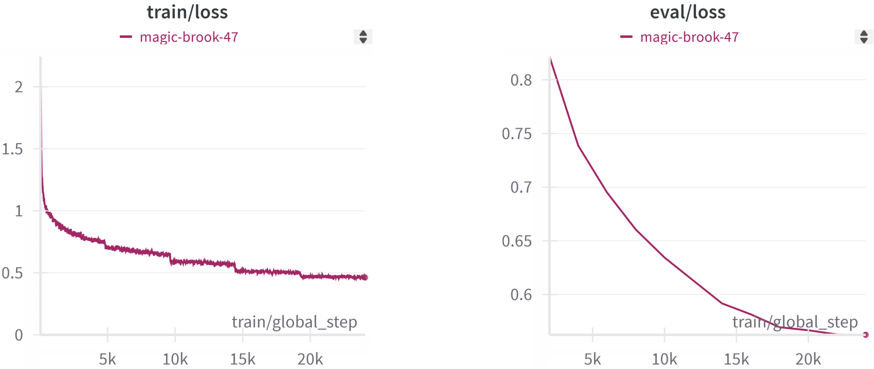
Training Methodology

- Data Preparation: Merged input/output with special tokens
- Model: Pre-trained "microsoft/phi-2"
- Optimizations:
  - 4-bit quantization
  - Flash Attention
  - Quantized Low-Rank Adapters (QLoRA)
- Finetuning: SFTTrainer with PEFT integration

Fine-tuning Parameters:

Number of Training Epochs:	5
Batch Size:	2
Gradient Accumulation Steps:	32
Optimizer:	Paged AdamW 8-bit
Learning Rate:	2e-4
Learning Rate Scheduler Type:	Cosine decay
Warmup Ratio:	0.05
Weight Decay:	0.01

Training and Validation



(a) Training Loss

(b) Validation Loss

Figure: Training and Validation Loss

Initial Evaluation

- Test set: 338 theorems from "Introduction to Analytic Number Theory"
- Results:
  - 152 translations parse and typecheck without errors
  - 16 out of 152 can be automatically proved by Sledgehammer
- Manual classification of 38 results (see next slide)

Manual Classification Results

- 15 nonsense
- 6 true but unrelated to original text
- 4 quite wrong
- 9 partially ok
- 3 quite good
- 1 correct

Conclusion: Results are often grammatically correct but largely semantically incorrect

Sample attempts

Thm	Translation to Isabelle
3.8	lemma mutually_visible_lattice_points_commute: "mutually_visible (a, b) (m, n) mutually_visible (b, a) (n, m)"
1.11	lemma divisor_set_prod_unique: assumes "n > 0" "n = prod_list ps" shows "divis n = set (map (λ p. p^a) ps)",
10.5	lemma residue_primroot: assumes "prime p" "odd p" shows "residue (primroot p) 1"
7.2	lemma prime_elem_4_times_i: "prime (4 * n + 1 :: int)"
7.7	lemma sum_log_p_over_p: assumes "x > 1" shows "sum_log_p_over_p x = (1 - N of_nat (phis k) * log x + O(λ_. 1)",
2.24	lemma multiplicative_function_unique: assumes "multiplicative_function f" "multi tive_function g" shows "f = g"
1.2	lemma common_divisor_dvd: fixes a b :: int shows "common_divisor a b dvd a" " mon_divisor a b dvd b"
6.8	lemma finite_characters: "finite (characters G)"
1.4	lemma gcd_comm_nat: "gcd a b = gcd b a" for a b :: nat
3.13	lemma abs_sum_upto_moebius_mu_over_n: "abs (sum_upto (λ n. moebius_mu n / n) x) ≤ 1"

Sample Translations

- Theorem 3.8: Incorrect translation of mutually visible lattice points
- Theorem 7.7: Close translation, minor issues with notation
- Theorem 1.4: Correct but incomplete translation (only one property out of four)

Challenges and Future Work

- Improve semantic accuracy of translations
- Implement feedback loop with type-checking and theorem proving
- Explore RAG integration for better use of AFP
- Evaluate on books not yet formalized
- Refine model to handle complex mathematical concepts and notations