Project Description: Experiments with Language Models for Isabelle Autoformalization

David Valente[1], Manuel Eberl[2], Cezary Kaliszyk[2], and Josef Urban[3]

[1] Instituto Superior Tecnico, Universidade de Lisboa
[2] University of Innsbruck
[3] Czech Technical University

1 Motivation

The formalization of mathematical theorems and their proofs stands as a cornerstone in modern mathematics and computer science. Manual formalization, although precise, is prone to errors and can consume significant time and effort.

Learning-assisted autoformalization [5] may offer a promising path to this challenge. It operates as a subset of machine translation tasks [8] in which (large) language models (LMs/LLMs) have shown to have remarkable performance, albeit with the added complexity of adhering to rigid and intricate grammatical structures inherent in formal logic systems.

In this recently started project, we experiment with the capabilities of LMs to tackle the autoformalization task. Specifically, our objective is to finetune the Phi-2 model on the task of translating LaTeX, a widely used typesetting system for mathematical documents, into Isabelle [9], a formal proof assistant. Furthermore we plan on exploring the benefits of building a feedback loop that adds type-checking and theorem proving to continuously improve the learner [7] and possibly adding RAG [6] to the pipeline for more accurate use of the AFP.

2 Training Data Description

Our training data consists of a curated dataset containing pairs of natural language statements and corresponding Isabelle lemmas. To generate LaTeX representations, we used an existing dataset of natural language-Isabelle lemma pairs [3], prompting the Mistral Large model [4] to generate the corresponding LaTeX. Notably, multiple LaTeX representations were generated for each natural language statement, ensuring diversity and coverage. In total, our dataset comprises over 100,000 pairs of LaTeX-Isabelle lemma pairs.

2.1 Example Data

Natural Language Statement: If a set X is countable, then the cardinality of set X is less than or equal to Aleph null (the smallest infinite cardinal number).

Corresponding LaTeX Representation:

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

Corresponding Isabelle Lemma:

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

3 Training Methodology

Data Preparation:
We preprocessed the data by merging input and output sequences while incorporating special tokens to delineate the beginning and end of LaTeX and Isabelle sections.

Model Configuration and Fine-tuning:
For model configuration, we loaded the pre-trained "microsoft/phi-2" model, ensuring its compatibility with the autoformalization task. Various optimizations were employed during model loading, including quantization with 4-bit configuration (BitsAndBytesConfig) and utilization of Flash Attention. Additionally, the model underwent further optimization using Quantized Low-Rank Adapters (QLoRA), focusing on key weight matrices (Wqkv) and fully-connected layers (fc1, fc2). Finetuning was then done through SFTTrainer to integrate PEFT and improve data and resource efficiency. See Appendix A for the details of the training.

4 Initial Evaluation

Our initial evaluation is done on the book "Introduction to Analytic Number Theory" [1] formalized in Isabelle by the second author [2]. We run the trained model on the LaTeX versions of the 338 main theorems (only statements, no proofs) and lemmas in that book. Note that in principle we are evaluating on data that are in various ways related to the training set, because the Isabelle formalization has been very likely seen by the various LMs used for producing our training data. If our results were very good, we would switch to books that are not formalized yet, however (as will be seen below), this is far from being the case yet.

We then created Isabelle/HOL scripts that (to some extent) allow us to classify the results automatically, and we also classify some of the results manually.

From the 338 translations, 152 result in Isabelle texts that parse and typecheck without producing errors. The remaining translations trigger various parsing and typechecking issues when processed by Isabelle. Only 16 of the 152 parsable ones can be automatically proved by Sledgehammer. An example of such an automatically provable statement is `"gcd a b = gcd b a"`, which is however only a truncated translation of Theorem 1.4 in [1].[1]

Our manual classification of 38 of the results is shown in Appendix B, along with some sample translations. Despite being often grammatically correct, these results are so far largely semantically incorrect. Their summary statistics is as follows: 15 nonsense; 6 true but unrelated to the original text; 4 quite wrong; 9 partially ok; 3 quite good; 1 correct.

[1] More precisely, the theorem there is a conjunction of four properties, and the trained LM only produced one of them.

References

[1] T. M. Apostol. *Introduction to analytic number theory.* Springer Science & Business Media, 2013.

[2] M. Eberl. Nine chapters of analytic number theory in Isabelle/HOL. In *ITP*, volume 141 of *LIPIcs*, pages 16:1–16:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[3] A. Q. Jiang, W. Li, and M. Jamnik. Multilingual mathematical autoformalization. *CoRR*, abs/2311.03755, 2023.

[4] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de Las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024.

[5] C. Kaliszyk, J. Urban, J. Vyskocil, and H. Geuvers. Developing corpus-based translation methods between informal and formal mathematics: Project description. In *CICM*, volume 8543 of *Lecture Notes in Computer Science*, pages 435–439. Springer, 2014.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020.

[7] Q. Wang, C. E. Brown, C. Kaliszyk, and J. Urban. Exploration of neural machine translation in autoformalization of mathematics in mizar. In *CPP*, pages 85–98. ACM, 2020.

[8] Q. Wang, C. Kaliszyk, and J. Urban. First experiments with neural translation of informal to formal mathematics. In *CICM*, volume 11006 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2018.

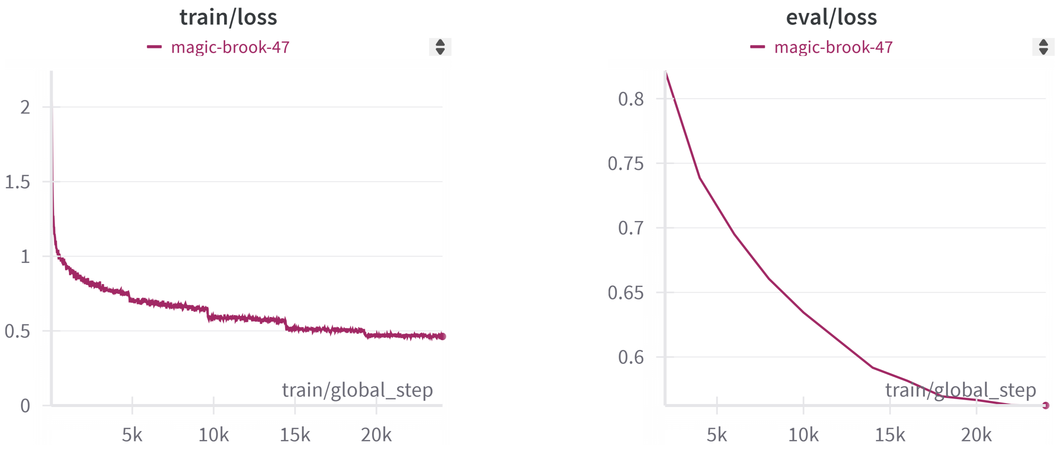
[9] M. Wenzel, L. C. Paulson, and T. Nipkow. The Isabelle framework. In O. A. Mohamed, C. A. Muñoz, and S. Tahar, editors, *Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August 18-21, 2008. Proceedings*, volume 5170 of *Lecture Notes in Computer Science*, pages 33–38. Springer, 2008.

A Training

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

B First impressions

Theorem	Comment
3.8.1	Definition of mutually visible lattice points. Completely wrong; it instead stated some kind of invariance under reflection.
1.11.1	It did not grasp that the ”ps” have to be prime numbers. Multiplicities are ignored completely, as is the fact that it must be possible to vary the multiplicities of each prime. All that aside, I would not phrase this with lists (it's pretty unwieldy in practice).
1.11.2	This is truncated.
10.5.1	Complete nonsense
5.25.1	Complete nonsense
6.14	All nonsense
2.11.1	Does not type check and does not seem to make sense either
10.1.1	Complete nonsense
7.2.1	Well at least it correctly translated ”4n+1” to ”4 * n + 1”, but the statement is still horribly wrong.
7.7.1	Looks pretty good. The ”sum_log_p_over_p” should be expanded to something more explicit, of course, and the ”+ O(λ_. 1)” does not quite typecheck (it should be something like ”+o O(λ_. 1)”), but close enough. It doesn't define the ”N”, but then neither does the LaTeX code you gave it.
7.7.2	Looks syntactically equivalent to the one above
8.12.1	Some good stuff there, but it completely dropped the ”G” and the quantification and the condition on the ”a” in the end is missing entirely. Also, it did not get that (x, y) is ”gcd x y” and not literally the tuple ”(x, y)”.
8.12.2	Same issue
2.24.1	The whole assumption is missing; rest is okay
7.6.1	It uses this ”sum_moebius_over_n”, which is not defined anywhere. Also it turned a ”O(1)” into a ”Θ(1)”. Otherwise okay.
9.12	All nonsense
2.27.1	Complete nonsense. No idea where it got this ”selberg_prob_density” from. Selberg has nothing to do with probabilities.
1.2.1	Well, part of it is there and correct. But most of it is missing.
6.6	It seems to have hallucinated some notation here that I've never seen before. It seems relatively reasonable though, if one were to replace the notation with the right concept. Most of the stuff seems to be missing though.
6.8.1	The order n is missing, as is the assumption that G is a finite abelian group
8.18	Nonsense
1.4.1	Well it got commutativity right, but that's only one of many properties
1.5.2	Wrong. The ”b = Suc 0” should be a ”gcd a b = Suc 0”.
2.26.1	True, but has nothing to do with what was being translated.
2.3.1	True, but has nothing to do with what was being translated.
2.4.1	True, but has nothing to do with what was being translated.
3.13.1	This looks correct.
5.16.1	Nonsensical
5.28	Nonsensical
5.2.2	True but trivial and not what was intended
5.32.1	Nonsense
5.6.1	True, but a completely different property
5.8.1	It basically proved $A \Longrightarrow A$. Wrong translation.
6.1.1	Very much not the property that was to be translated
9.11.1	Nonsense. The ”P = 1” etc. is way too strong an assumption. Apparently, it just doesn't understand what ”(P, Q) = 1” means. The conclusion is also completely wrong.
9.5.1	Nonsense
9.5.2	Nonsense



(a) Training Loss

(b) Validation Loss

Figure 1: Training and Validation Loss

Theorem	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 ”divisor_set n = set (map (λ p. p^a) ps)”,
10.5	lemma residue_primroot: assumes ”prime p” ”odd p” shows ”residue (primroot p) 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 k) / of_nat (phis k) * log x + O(λ_. 1)”,
8.12	lemma (in dcharacter) dcharacter_mod_1_imp_dvd: assumes ”χ mod k ≠ 0” and ”(n, k) > 1” shows ”∃ d. d dvd k d < k χ a = 1”,
2.24	lemma multiplicative_function_unique: assumes ”multiplicative_function f” ”multiplicative_function g” shows ”f = g”
1.2	lemma common_divisor_dvd: fixes a b :: int shows ”common_divisor a b dvd a” ”common_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 / real n) x) ≤ 1”