Proof Recommendation System for the HOL4 Theorem Prover

Nour Dekhil, Adnan Rashid, and Sofiène Tahar

Department of Electrical and Computer Engineering
Concordia University, Montreal, QC, Canada
{n_dekhil,rashid,tahar}@ece.concordia.ca

We introduce a proof recommender system for the HOL4 theorem prover [1]. Our tool is built upon a transformer-based model [2] designed specifically to provide proof assistance in HOL4. The model is trained to discern theorem proving patterns from extensive libraries of HOL4 containing proofs of theorems. Consequently, it can accurately predict the next tactic(s) (proof step(s)) based on the history of previously employed tactics. The tool operates by reading a given sequence of tactics already used in a proof process (in our case, it contains at least three tactics), referred to as the current proof state, and provides recommendations for the next optimal proof step(s).

Figure 1 depicts the major steps taken to develop the proof recommendation tool. The initial block (highlighted in blue color) refers to the construction of a HOL4 proofs dataset. In the dataset construction phase, we are abstracting the proof scripts to only include the tactics used to prove a theorem or a lemma. This process involves systematically parsing each `sml` file, which contains the proof scripts written in HOL4. Within each file, we identify all theorems and lemmas that are subject to proof. Once these target points are identified, the next task is to extract the specific tactics that were used to prove each theorem or lemma. This involves traversing the proof script to capture only those commands that directly contribute to the proof, omitting extraneous elements that do not influence the proof's logical flow.

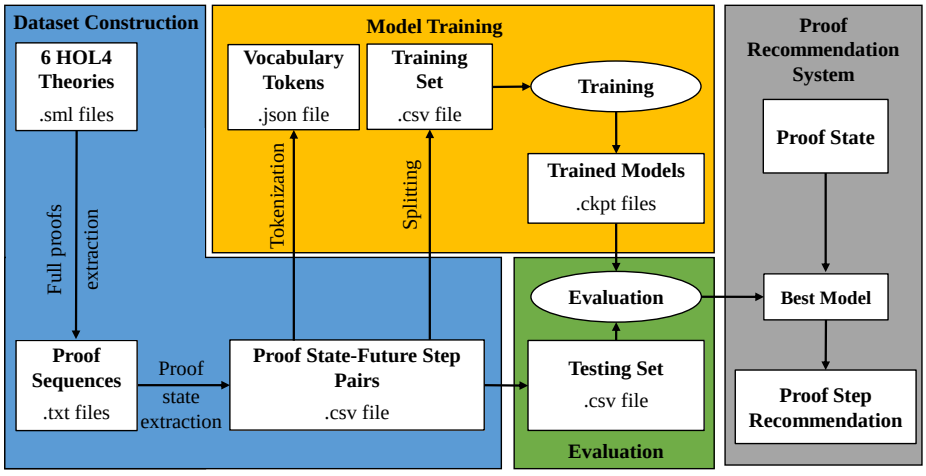

Figure 1: Proof Recommendation System

We created large proof sequences datasets (Datasets 1-5) from five HOL4 theories [3–7] developed by the Hardware Verification Group (HVG) of Concordia University alongside an already available dataset created using the real arithmetic theory of HOL4 (Dataset 6) [8]. For experimental purposes, we combined all datasets into Dataset 7. Our objective is to predict

the subsequent tactic from a sequence of previously employed tactics. To accomplish this, we approach this challenge as a multi-label classification task using language models. To facilitate this, we restructure the dataset into pairs of current proof states and possible future tactics. More details on the datasets used for classification are given in Table 1.

Table 1: Summary of the used Datasets

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7
Distinct Tactics	115	132	26	44	32	89	162
Proofs	1,873	2,475	153	295	61	279	5,136
Proof States	43,167	57,602	2,973	7,371	1,784	3,259	116,156

Our primary objective is to predict the subsequent tactic in a sequence of previously applied tactics during a proof. To address this, we framed the problem as a multi-label classification task, which is particularly suitable for scenarios where multiple correct outcomes are possible. We restructured the original dataset into pairs, with each pair consisting of a current proof state (a sequence of tactics that have already been applied) and the corresponding possible future tactics that could logically follow. This restructuring allows the model to learn the relationships between different proof states and their subsequent steps, enabling it to make informed predictions about the next optimal tactic.

In our experimental phase, we explored various transformer-based language models, including BERT [9], RoBERTa [10], and T5 [11]. These models are well-known for their ability to capture intricate patterns in sequential data, making them ideal for our task of proof recommendation. Each model was trained on the restructured datasets, which were split into a 90-10 ratio for training and testing purposes (block of Figure 1 highlighted in orange color). This split ensures that the models are exposed to a broad range of examples during training while still having a significant portion of data reserved for testing.

To optimize the performance of each model, we employed a grid search of hyperparameters, a method that systematically evaluates a combination of parameters to identify the configuration that yields the best results (block of Figure 1 highlighted in green color). This process was critical in fine-tuning the models, ensuring they were not only accurate but also efficient in their predictions. Given the multitude of possible tactics at each proof state, we decided to generate multiple recommendations for the next proof step, rather than a single prediction. This approach acknowledges the inherent complexity and variability of theorem proving, where several tactics could be appropriate in advancing a proof.

The accuracy of our model's recommendations was assessed using the n-correctness rate, an evaluation metric that measures the probability that a correct tactic from the testing dataset is included among the top-n recommended tactics. This metric is particularly useful in scenarios where multiple recommendations are provided, as it quantifies the likelihood of the correct tactic being present within a certain range of suggestions. Through extensive testing, we found out that RoBERTa demonstrated a superior performance across most cases for $n = 7$. As a result, we deploy it into our proof recommendation tool (block of Figure 1 highlighted in grey color).

With the aim of efficiently predicting the next tactic ($k = 1$, where k represents the number of future tactics to predict) for the majority of theory datasets, we also challenged our tool by attempting to predict two future tactics. Table 2 provides further details of the experimental results for RoBERTa in predicting one future tactic ($k = 1$) and two future tactics ($k = 2$). After examining the performance results across different datasets, it seems that the variations arise from the diversity and patterns unique to each dataset, as well as the range of tactics employed. Specifically, Datasets 1-5 exhibit a uniformity in their proof structures, originating

Table 2: Correctness Rates of RoBERTa Considering Top-7 Recommendations

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7
k = 1	73.6%	79.5%	94.4%	**97.8%**	97.6%	64.3%	89.8%
k = 2	54.3%	58.6%	88.1%	**96.8%**	92.2%	29.4%	80.3%

from one application project written by a single person, thus making the proofs more homogeneous and consistent in style. However, Dataset 6, came from HOL4 libraries containing a diverse range of theorems regarding different mathematical concepts, presents proofs with heterogeneous patterns, making them challenging to predict. Additionally, we observed a decrease in performance when attempting to predict two future tactics, which may be attributed to the expansive space of possibilities and resulting in increased uncertainty.

In the recent past, several studies have integrated artificial intelligence into theorem prover tools (e.g., PVS and Coq), particularly for predicting future-proof steps. For instance, in the study reported in [12], accuracies ranging from 50% to 70% were achieved for the top 3-5 recommendations, while the work in [13] achieved 87% accuracy for the top 3, and the one in [14] reported 54.3% accuracy for the top 10. In comparison, our tool surpasses results reported in these studies, achieving accuracies of 77.3%, 89.88%, and 93.7% for the top 3, 7, and 10 next tactic recommendations, respectively, measured on the combined Dataset 7. The current tool version is available to try online [15]. In the future, we plan to expand it to include more HOL4 theories and enhance its interfacing with HOL4. In addition, we are investigating its potential to automatically generate complete proofs, considering the need for optimization given the exponential growth in combination possibilities with the proof sequence length. To address this, we plan to use some advanced tree search algorithms.

References

[1] HOL4. https://hol-theorem-prover.org/, 2024.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, page 6000–6010. Curran Associates Inc., 2017.

[3] Dataset 1: Formal Dynamic Dependability Analysis using HOL Theorem Proving. https://hvg.ece.concordia.ca/projects/prob-it/pr9.php, 2024.

[4] Dataset 2: Formal Probabilistic Analysis of Wireless Sensor Networks. https://hvg.ece.concordia.ca/projects/prob-it/wsn.php, 2024.

[5] Dataset 3: Formal Probabilistic Risk Assessment using Theorem Proving. https://hvg.ece.concordia.ca/projects/prob-it/pr10/index.php, 2024.

[6] Dataset 4: Formal Analysis of Information Flow Using Min-Entropy and Belief Min-Entropy. https://hvg.ece.concordia.ca/projects/prob-it/pr5.php, 2024.

[7] Dataset 5: Formalization of Normal Random Variables. https://hvg.ece.concordia.ca/projects/prob-it/pr7.html, 2024.

[8] Dataset 6: Proof searching in HOL4 with Genetic Algorithm. https://dl.acm.org/doi/10.1145/3341105.3373917, 2024.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. Association for Computational Linguistics, 2019.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining

approach. *CoRR*, abs/1907.11692, 2019.

[11] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. volume 21, pages 1–67, 2019.

[12] Eric Yeh, Briland Hitaj, Sam Owre, Maena Quemener, and Natarajan Shankar. CoProver: A Recommender System for Proof Construction. In *Intelligent Computer Mathematics*, volume 14101 of *LNAI*, pages 237–251. Springer, 2023.

[13] Lasse Blaauwbroek, Josef Urban, and Herman Geuvers. Tactic Learning and Proving for the Coq Proof Assistant. *arXiv preprint arXiv:2003.09140*, 2020.

[14] Xiaokun Luan, Xiyue Zhang, and Meng Sun. Using LSTM to Predict Tactics in Coq. In *Software Engineering and Knowledge Engineering*, pages 132–137, 2021.

[15] HOL4PRS: Proof Recommendation System for the HOL4 Theorem Prover. https://github.com/DkNour/HOL4PRS-Proof-Recommendation-System-for-the-HOL4-Theorem-Prover.git.