Proof Recommendation System for HOL4 Theorem Prover

Nour Dekhil, Adnan Rashid, and Sofiène Tahar

Department of Electrical and Computer Engineering
Concordia University, Montreal, QC, Canada

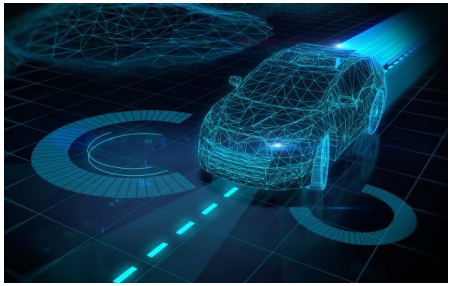
AITP 2024
Sep 5, 2024

Why formal verification is so important?

> Ensuring the correct functionality of complex software systems or critical hardware components is crucial.

Automotive

Healthcare



Aerospace

> **Interactive Theorem Provers** (ITPs) help apply mathematical logic to verify these systems.

Challenges of ITPs

Challenging and labor-intensive !

Smart copilot that guides us through theorem proving.

Our Objective

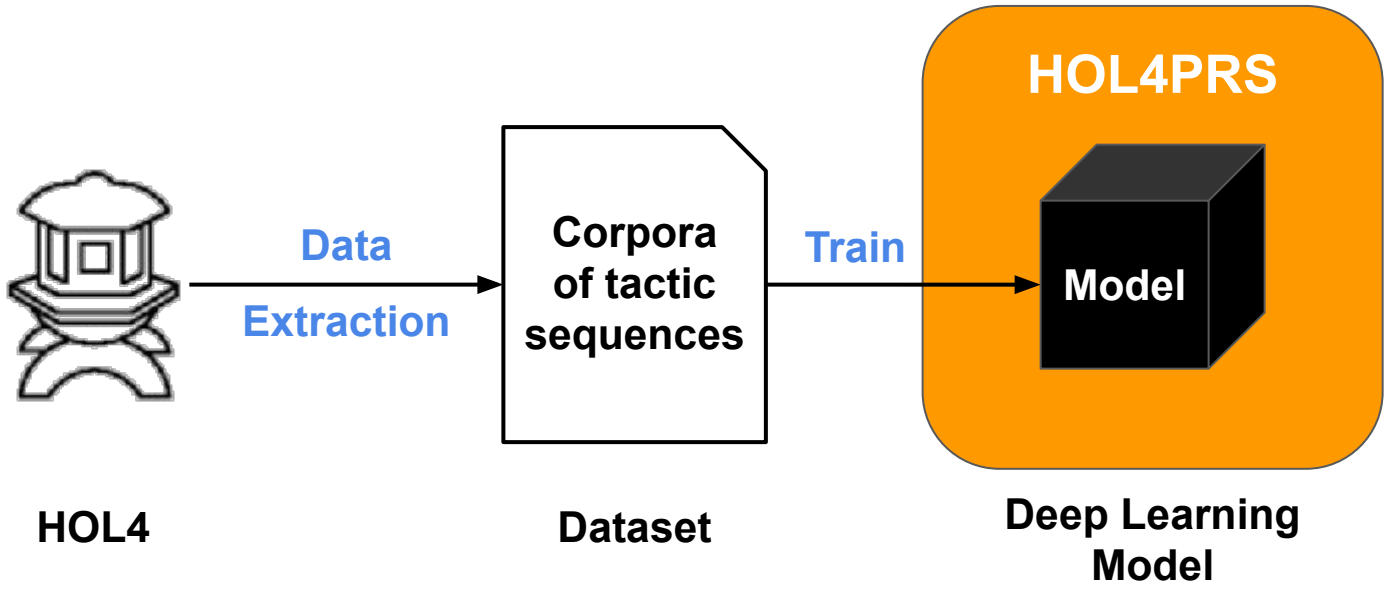
- Leverage Large Language Models (LLMs) capabilities to make proof writing more efficient.

- Recommend the best HOL4 proof step to use.

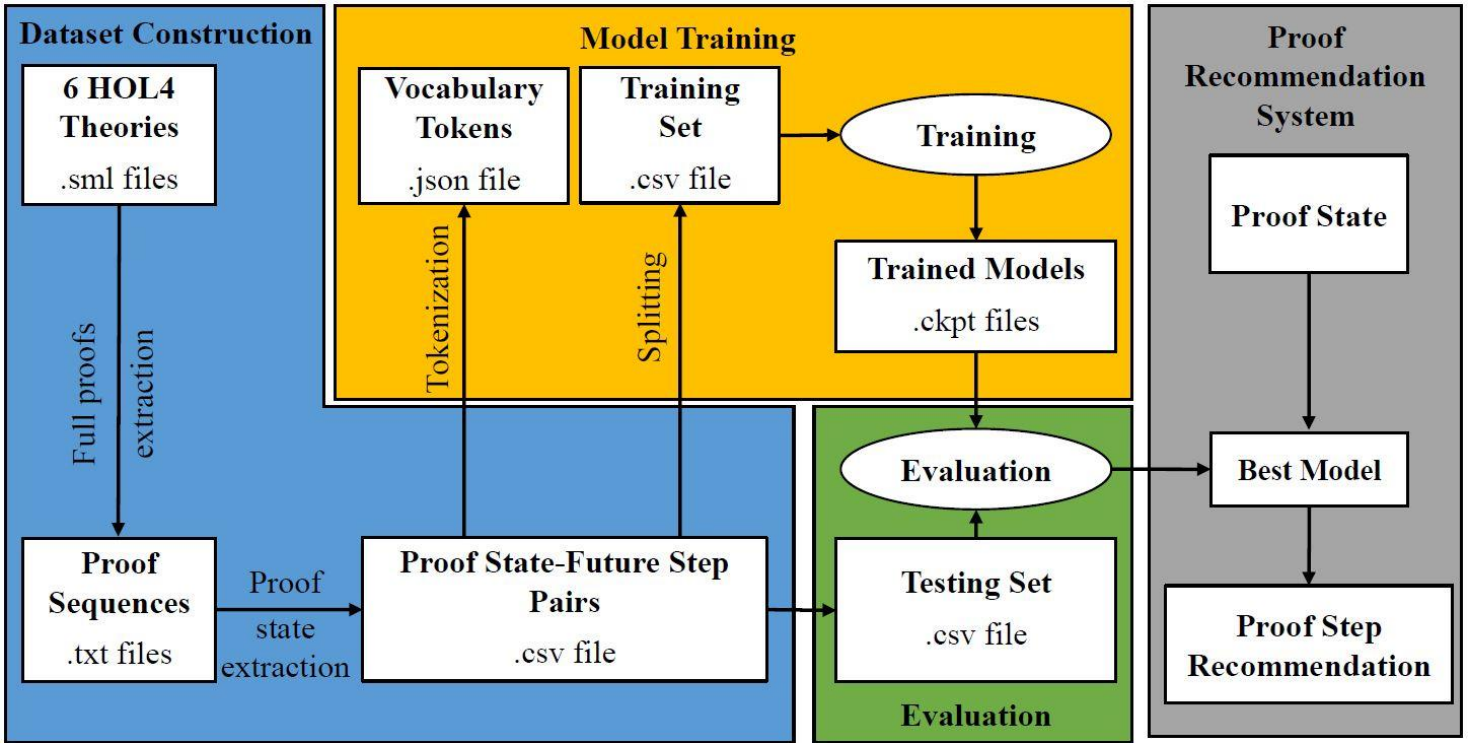
Related Work

Related Work	Approach	Prover	Success Rate
Gauthier et al., 2020	Machine Learning (k-NN)	HOL4	66.4%
Blaauwbroek et al., 2020	Machine Learning	Coq	23.4%
Luan et al., 2021	Deep Learning (LSTM)	Coq	87%
Yeh et al., 2023	Deep Learning (T5)	PVS	70%

HOL4PRS: Proof Recommendation System for HOL4 Theorem Prover



HOL4PRS Tool Structure

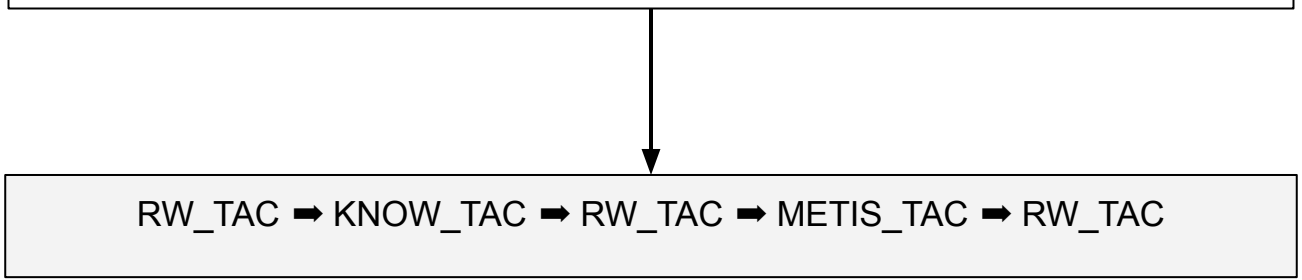


Dataset Construction

.*sml* File

```
val rule_18 = store_thm("rule_18",
  ``!X Y. D_AND X Y = D_OR (P_AND X Y)(P_AND Y X)``,

RW_TAC std_ss[D_AND_def, P_AND_def, D_OR_def]
THEN KNOW_TAC(``  !s.( max (X s) (Y s)) =
    (
         min (if X s <= Y s then Y s else PosInf)
           (if Y s <= X s then X s else PosInf))``)
    THEN1 (RW_TAC std_ss[extreal_max_def, extreal_min_def] THEN
METIS_TAC[extreal_lt_def, extreal_le_def, lt_le, le_lt, le_trans,
lt_trans, lt_infty, le_infty])
THEN RW_TAC std_ss[] ) ;
```



Tactic Sequence

Datasets

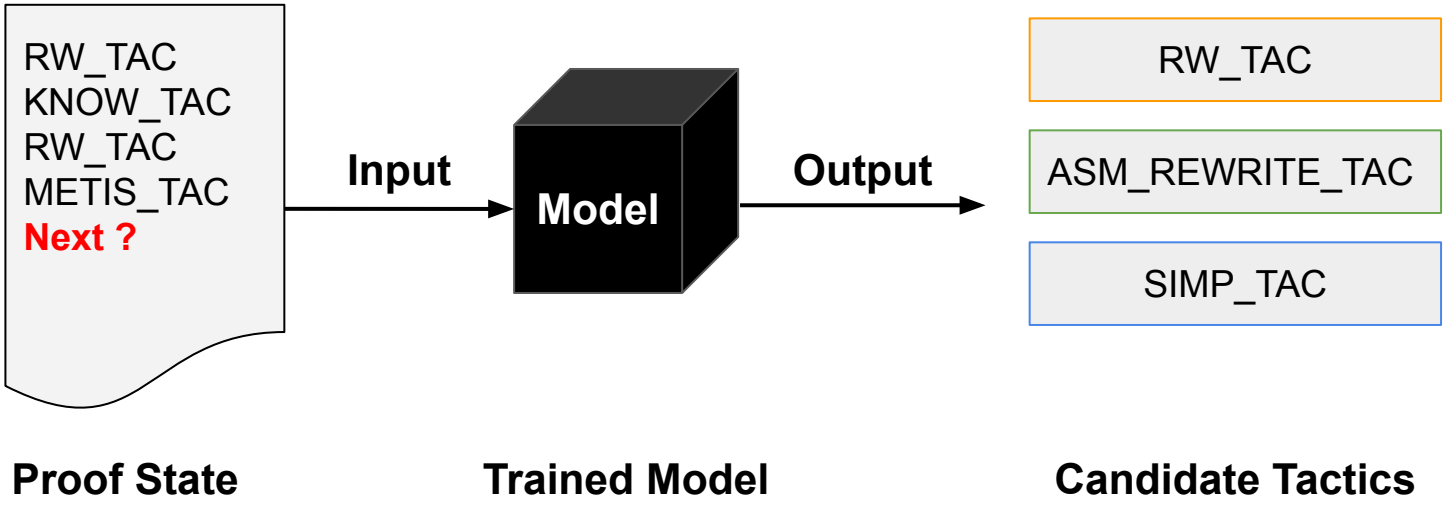
Proof: A sequence of n tactics used to prove a theorem.

Proof States: All possible tactic sequences recorded from a proof.

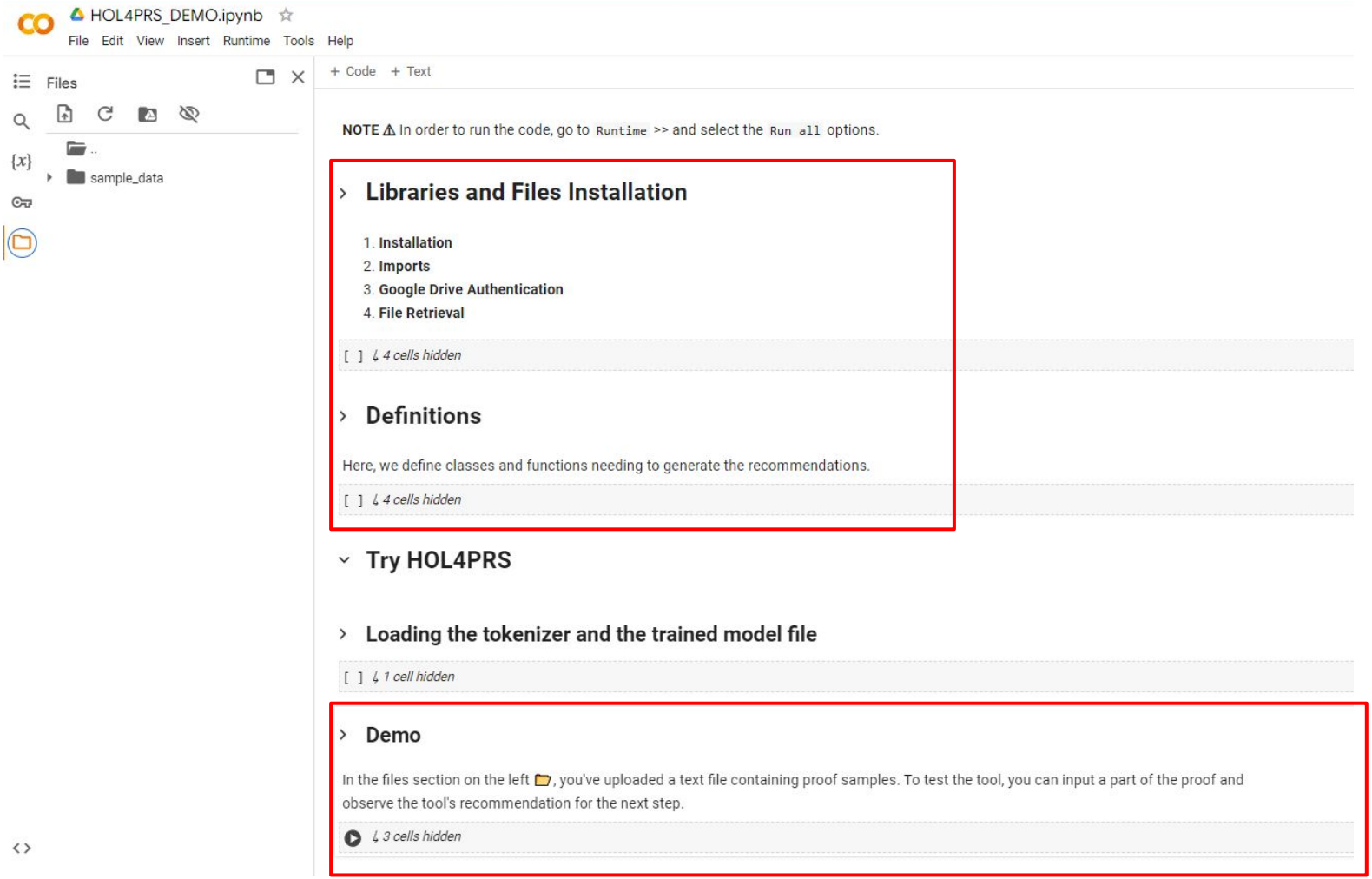
	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7
Proofs	1,873	2,475	153	295	61	279	5,136
Proof States	43,167	57,602	2,973	7,371	1,784	3,259	116,156

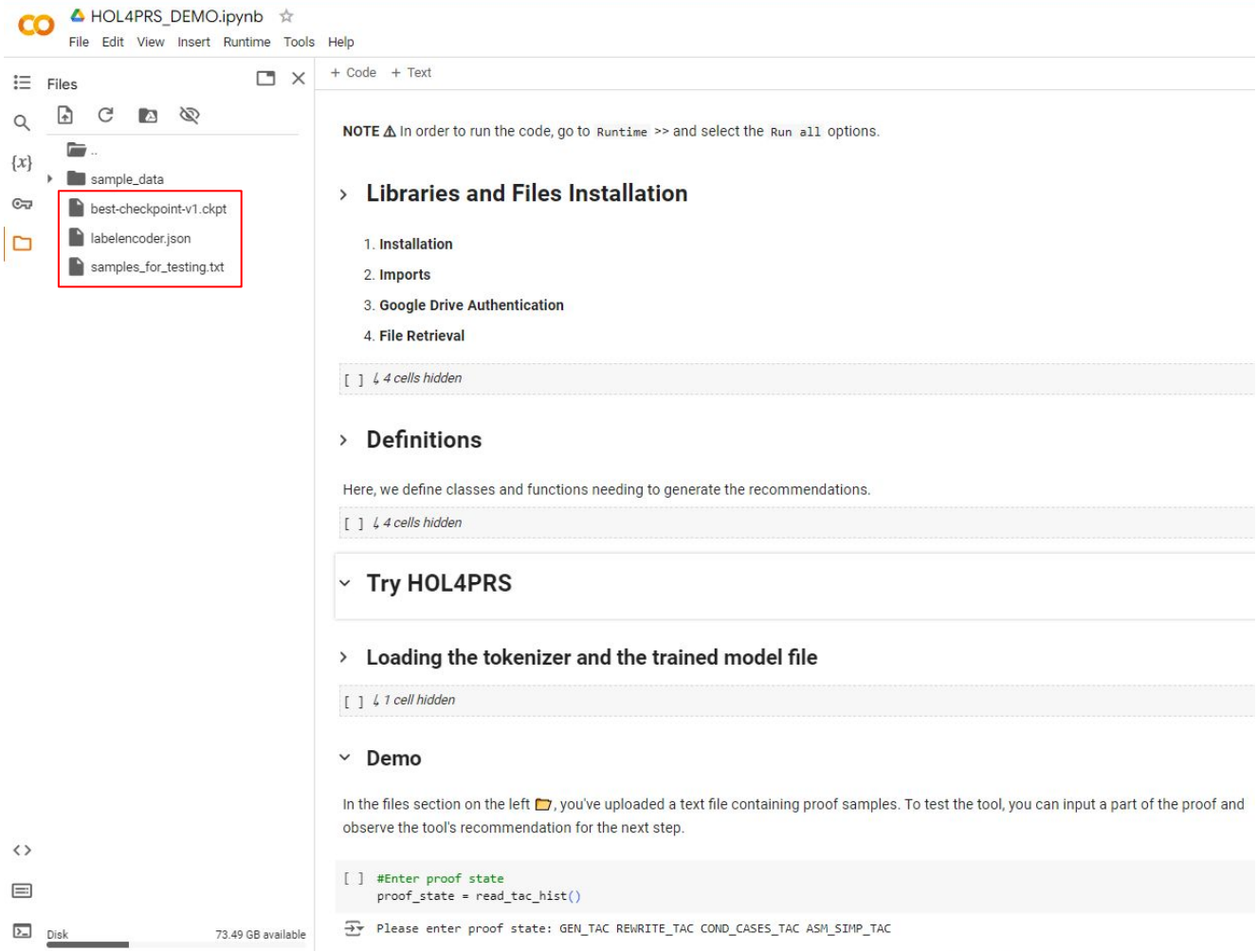
Dataset 7 is a combination of the Datasets 1 to 6.

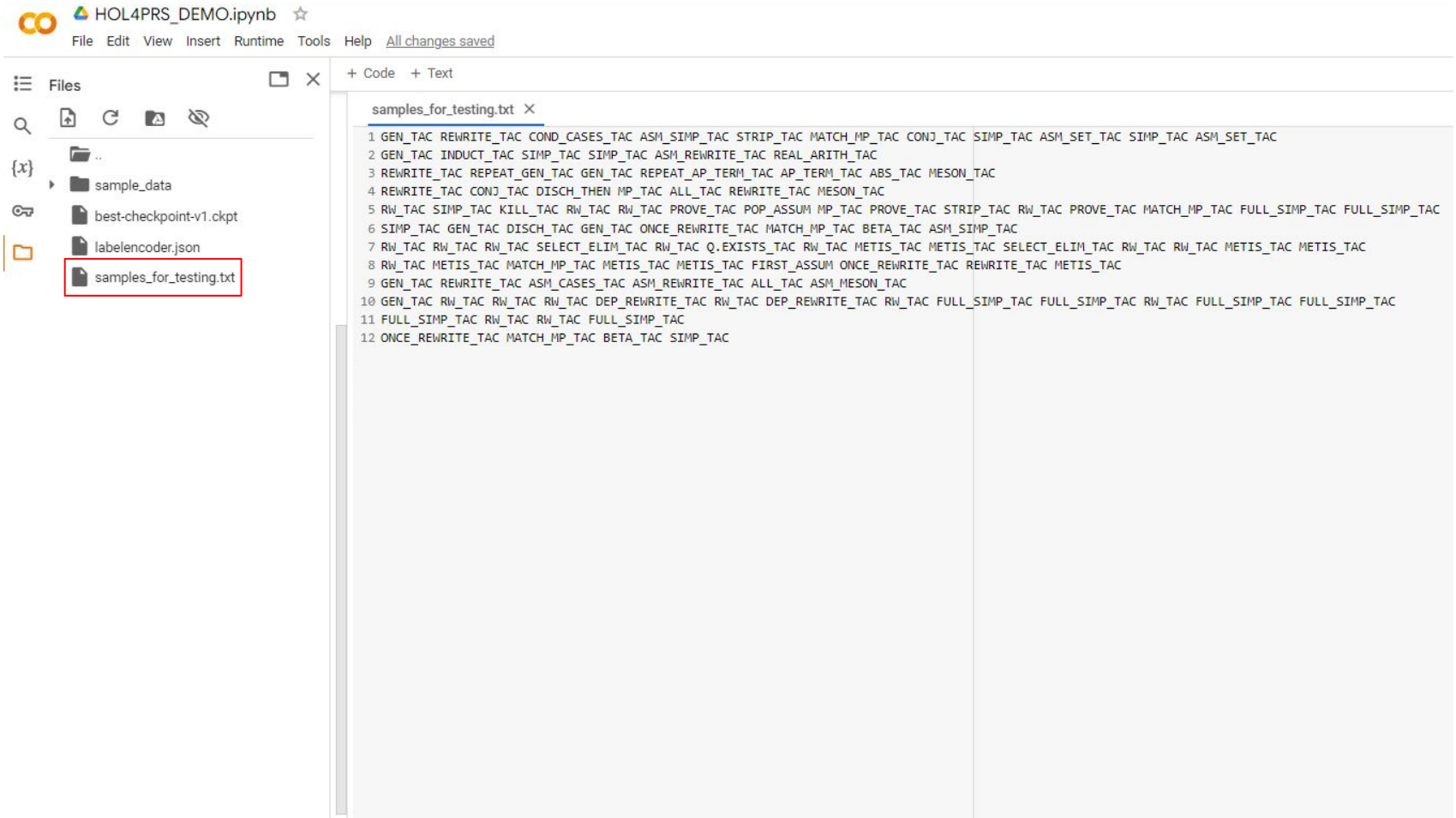
Train Model


RW_TAC
KNOW_TAC
RW_TAC
METIS_TAC
Next ?
Input
Model
Output
RW_TAC
ASM_REWRITE_TAC
SIMP_TAC
Proof State
Trained Model
Candidate Tactics

Demo


HOL4PRS_DEMO.ipynb
File Edit View Insert Runtime Tools Help
Files
sample_data
+ Code + Text
NOTE ⚠ In order to run the code, go to Runtime >> and select the Run all options.
Libraries and Files Installation
1. Installation
2. Imports
3. Google Drive Authentication
4. File Retrieval
[ ] 4 cells hidden
Definitions
Here, we define classes and functions needing to generate the recommendations.
[ ] 4 cells hidden
Try HOL4PRS
Loading the tokenizer and the trained model file
[ ] 1 cell hidden
Demo
In the files section on the left, you've uploaded a text file containing proof samples. To test the tool, you can input a part of the proof and observe the tool's recommendation for the next step.
3 cells hidden


HOL4PRS_DEMO.ipynb
File Edit View Insert Runtime Tools Help
Files
sample_data
best-checkpoint-v1.ckpt
labelencoder.json
samples_for_testing.txt
Disk 73.49 GB available
+ Code + Text
NOTE ⚠ In order to run the code, go to Runtime >> and select the Run all options.
Libraries and Files Installation
1. Installation
2. Imports
3. Google Drive Authentication
4. File Retrieval
[ ] 4 cells hidden
Definitions
Here, we define classes and functions needing to generate the recommendations.
[ ] 4 cells hidden
Try HOL4PRS
Loading the tokenizer and the trained model file
[ ] 1 cell hidden
Demo
In the files section on the left, you've uploaded a text file containing proof samples. To test the tool, you can input a part of the proof and observe the tool's recommendation for the next step.
[ ] #Enter proof state
proof_state = read_tac_hist()
Please enter proof state: GEN_TAC REWRITE_TAC COND_CASES_TAC ASM_SIMP_TAC


HOL4PRS_DEMO.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
+ Code + Text
sample_data
best-checkpoint-v1.ckpt
labelencoder.json
samples_for_testing.txt
samples_for_testing.txt
1 GEN_TAC REWRITE_TAC COND_CASES_TAC ASM_SIMP_TAC STRIP_TAC MATCH_MP_TAC CONJ_TAC SIMP_TAC ASM_SET_TAC SIMP_TAC ASM_SET_TAC
2 GEN_TAC INDUCT_TAC SIMP_TAC SIMP_TAC ASM_REWRITE_TAC REAL_ARITH_TAC
3 REWRITE_TAC REPEAT_GEN_TAC GEN_TAC REPEAT_AP_TERM_TAC AP_TERM_TAC ABS_TAC MESON_TAC
4 REWRITE_TAC CONJ_TAC DISCH_THEN MP_TAC ALL_TAC REWRITE_TAC MESON_TAC
5 RW_TAC SIMP_TAC KILL_TAC RW_TAC RW_TAC PROVE_TAC POP_ASSUM MP_TAC PROVE_TAC STRIP_TAC RW_TAC PROVE_TAC MATCH_MP_TAC FULL_SIMP_TAC FULL_SIMP_TAC
6 SIMP_TAC GEN_TAC DISCH_TAC GEN_TAC ONCE_REWRITE_TAC MATCH_MP_TAC BETA_TAC ASM_SIMP_TAC
7 RW_TAC RW_TAC RW_TAC SELECT_ELIM_TAC RW_TAC Q.EXISTS_TAC RW_TAC METIS_TAC METIS_TAC SELECT_ELIM_TAC RW_TAC RW_TAC METIS_TAC METIS_TAC
8 RW_TAC METIS_TAC MATCH_MP_TAC METIS_TAC METIS_TAC FIRST_ASSUM ONCE_REWRITE_TAC REWRITE_TAC METIS_TAC
9 GEN_TAC REWRITE_TAC ASM_CASES_TAC ASM_REWRITE_TAC ALL_TAC ASM_MESON_TAC
10 GEN_TAC RW_TAC RW_TAC RW_TAC DEP_REWRITE_TAC RW_TAC DEP_REWRITE_TAC RW_TAC FULL_SIMP_TAC FULL_SIMP_TAC RW_TAC FULL_SIMP_TAC FULL_SIMP_TAC
11 FULL_SIMP_TAC RW_TAC RW_TAC FULL_SIMP_TAC
12 ONCE_REWRITE_TAC MATCH_MP_TAC BETA_TAC SIMP_TAC

Demo

In the files section on the left 📂, you've uploaded a text file containing proof samples. To test the tool, you can input a part of the proof and observe the tool's recommendation for the next step.

```
#Enter proof state
proof_state = read_tac_hist()
```

```
Please enter proof state:
```

```
[ ] #Generate recommendation
    get_recom(proof_state, roberta_model, tokenizer)
```

samples_for_testing.txt

```
1 GEN_TAC REWRITE_TAC COND_CASES_TAC ASM_SIMP_TAC STRIP_TAC MATCH_MP_TAC CONJ_TAC SIMP_TAC ASM_SET_TAC SIMP_TAC ASM_SET_TAC
2 GEN_TAC INDUCT_TAC SIMP_TAC SIMP_TAC ASM_REWRITE_TAC REAL_ARITH_TAC
3 REWRITE_TAC REPEAT_GEN_TAC GEN_TAC REPEAT_AP_TERM_TAC AP_TERM_TAC ABS_TAC MESON_TAC
4 REWRITE_TAC CONJ_TAC DISCH_THEN MP_TAC ALL_TAC REWRITE_TAC MESON_TAC
5 RW_TAC SIMP_TAC KILL_TAC RW_TAC RW_TAC PROVE_TAC POP_ASSUM MP_TAC PROVE_TAC STRIP_TAC RW_TAC PROVE_TAC MATCH_MP_TAC FULL_SIMP_TAC FULL_SIMP_TAC
6 SIMP_TAC GEN_TAC DISCH_TAC GEN_TAC ONCE_REWRITE_TAC MATCH_MP_TAC BETA_TAC ASM_SIMP_TAC
7 RW_TAC RW_TAC RW_TAC SELECT_ELIM_TAC RW_TAC Q.EXISTS_TAC RW_TAC METIS_TAC METIS_TAC SELECT_ELIM_TAC RW_TAC RW_TAC METIS_TAC METIS_TAC
```

Demo

In the files section on the left 📂, you've uploaded a text file containing proof samples. To test the tool, you can input a part of the proof and observe the tool's recommendation for the next step.

```
[10] #Enter proof state
     proof_state = read_tac_hist()

Please enter proof state: REWRITE_TAC CONJ_TAC DISCH_THEN MP_TAC ALL_TAC REWRITE_TAC
```

```
#Generate recommendation
get_recom(proof_state, roberta_model, tokenizer)

HOL4PRS reccommendations are:  ['MAPEVERY', 'MESON_TAC', 'REALARITH_TAC', 'REPEAT GEN_TAC', 'REPEAT STRIP_TAC', 'REWRITE_TAC', 'SET_TAC']
```

samples_for_testing.txt

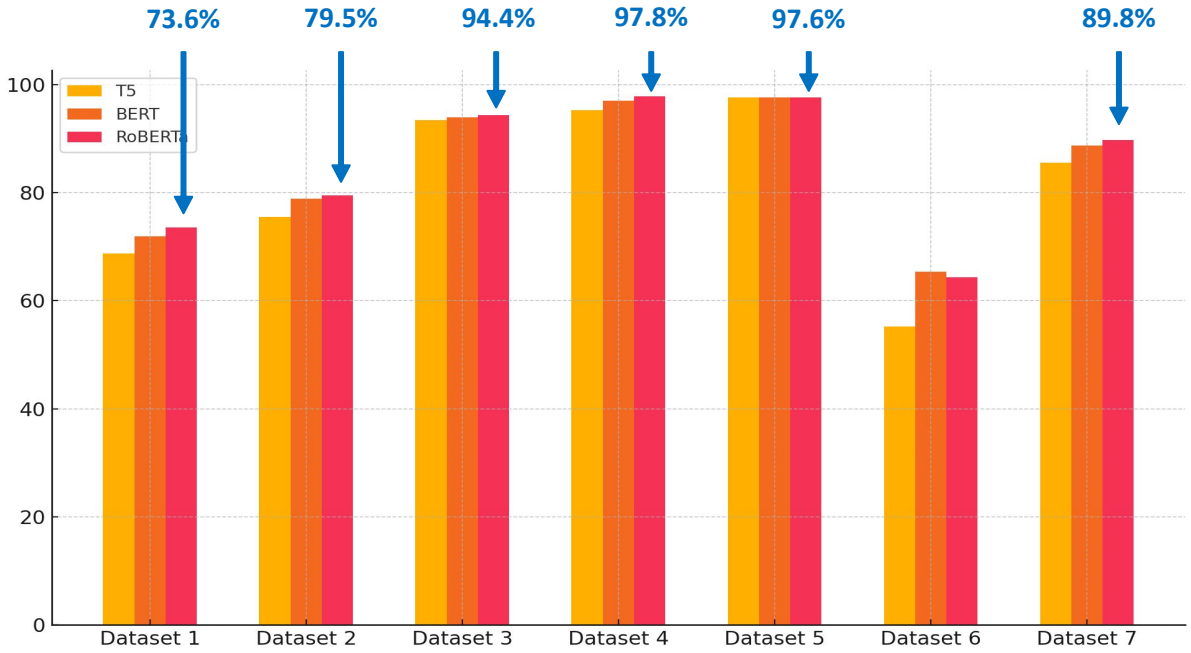
```
1 GEN_TAC REWRITE_TAC COND_CASES_TAC ASM_SIMP_TAC STRIP_TAC MATCH_MP_TAC CONJ_TAC SIMP_TAC ASM_SET_TAC SIMP_TAC ASM_SET_TAC
2 GEN_TAC INDUCT_TAC SIMP_TAC SIMP_TAC ASM_REWRITE_TAC REAL_ARITH_TAC
3 REWRITE_TAC REPEAT_GEN_TAC GEN_TAC REPEAT_AP_TERM_TAC AP_TERM_TAC ABS_TAC MESON_TAC
4 REWRITE_TAC CONJ_TAC DISCH_THEN MP_TAC ALL_TAC REWRITE_TAC MESON_TAC
5 RW_TAC SIMP_TAC KILL_TAC RW_TAC RW_TAC PROVE_TAC POP_ASSUM MP_TAC PROVE_TAC STRIP_TAC RW_TAC PROVE_TAC MATCH_MP_TAC FULL_SIMP_TAC FULL_SIMP_TAC
6 SIMP_TAC GEN_TAC DISCH_TAC GEN_TAC ONCE_REWRITE_TAC MATCH_MP_TAC BETA_TAC ASM_SIMP_TAC
7 RW_TAC RW_TAC RW_TAC SELECT_ELIM_TAC RW_TAC Q.EXISTS_TAC RW_TAC METIS_TAC METIS_TAC SELECT_ELIM_TAC RW_TAC RW_TAC METIS_TAC METIS_TAC
```

Experimental Results

N-Correctness Rate: Probability that the correct tactic is among the top N recommendations.

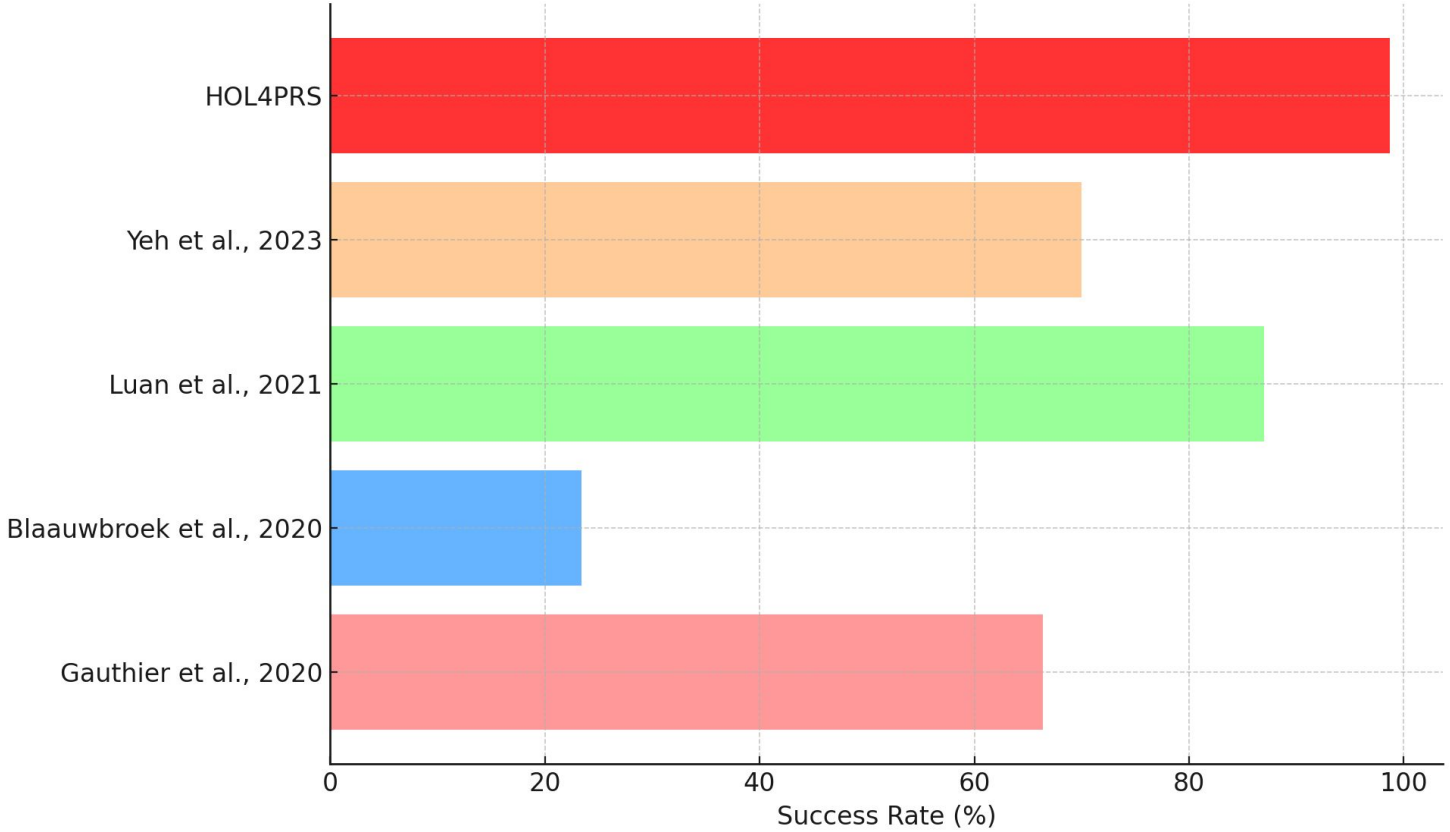
Datasets	T5			BERT			RoBERTa		
	Top 3	Top 7	Top 10	Top 3	Top 7	Top 10	Top 3	Top 7	Top 10
Dateset 1	51.3%	68.7%	76.4%	52.7%	71.9%	79.9%	54.5%	73.6%	93.7%
Dateset 2	60.4%	75.5%	80.5%	60.5%	78.9%	86.3%	59.7%	79.5%	85.8%
Dateset 3	69.8%	93.4%	95.4%	76.1%	93.9%	97%	78.4%	94.4%	97.5%
Dateset 4	77.3%	95.3%	97.2%	87.3%	97.0%	98.5%	89.5%	97.8%	98.8%
Dateset 5	76.6%	97.6%	98.2%	76.6%	97.6%	98.2%	76.6%	97.6%	97.6%
Dateset 6	39.9%	55.2%	61.9%	45.1%	65.4%	72.7%	43.4%	64.3%	73.8%
Dataset 7	**72.9%**	**85.6%**	**87.8%**	**75.4%**	**88.7%**	**92.3%**	**77.3%**	**89.8%**	**93.7%**

Experimental Results



HOL4PRS deploys RoBERTA for Top-7 recommendations

Results Comparaison



Future Work

- Integrate more contextual information with the input to improve the accuracy and relevance of proof step recommendations.
- Expand HOL4PRS to include more HOL4 theories.
- Extend the system to support other interactive theorem provers (ITPs) beyond HOL4.
- Explore premise selection approaches.
- Autonomously generate complete proofs without human intervention.

Proof Recommendation System for HOL4 Theorem Prover

For more information visit

https://hvg.ece.concordia.ca/projects/fvai/pr1/

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

Thank you!
Q&A

UNIVERSITÉ Concordia UNIVERSITY