Detecting AI Planning Modelling Mistakes - Potential Errors and Benchmark Domains

Kayleigh Sleath and Pascal Bercher

Australian National University

firstname.lastname@anu.edu.au

AI Planning

- Algorithms
- Deterministic & traceable - not reliant on probabilities/large sets of training data

Applications

- Transportation problems
- Chemical reactions
- Robotics
- Urban planning
- Scheduling
- Manufacturing

Domain-Independent Planners

- Domain-specific: expensive, inefficient, limited
- Domain-independent: flexible, easy to use
- If you have a correct domain, can solve any planning problem

DOMAIN: possible actions

PROBLEM: initial and goal state

Listing 1.1. A PDDL action for moving a truck between locations [3].

```
(:action drive
  :parameters (?t - truck ?from ?to - location)
  :precondition (at ?t ?from)
  :effect (and (not (at ?t ?from)) (at ?to)))
```

Domain-Independent Planners

- Domain-specific: expensive, inefficient, limited
- Domain-independent: flexible, easy to use
- If you have a correct domain, can solve any planning problem

IF you have a correct domain

Domains

- Hard & long process
- Encode a real-world situation
- Human input is error-prone
- Software relatively new - underdeveloped error diagnosis

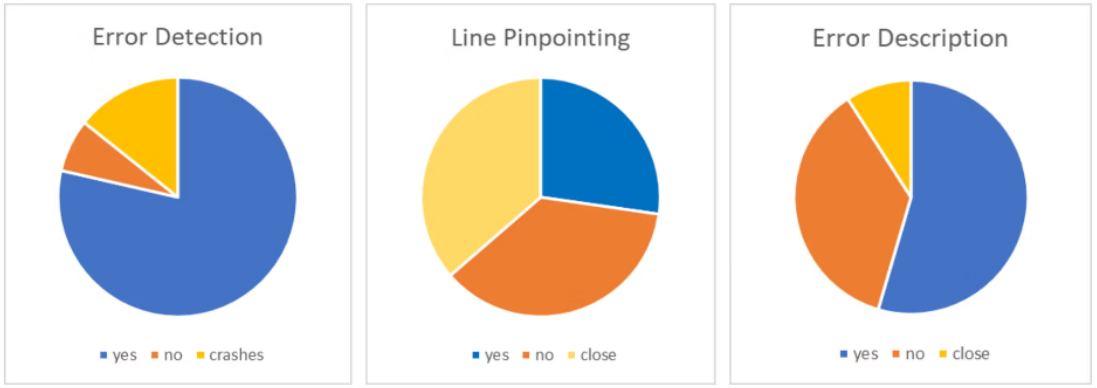
Researchers pioneering AI planning don't have time to think about domain error diagnosis ...

But I Do

- list of potential domain errors
  - syntax (errors)
  - semantics (warnings)
- benchmark repository of flawed domains (classical and hierarchical) - https://github.com/ProfDrChaos/flawedPlanningModels
- evaluation of domain modelling assistance software/planners

Planning.Domains

- Cloud-based accessible interface
- Widely used - 178,000 since 2015
- 79% syntax error detection
- 64% location guidance
- 55% error description

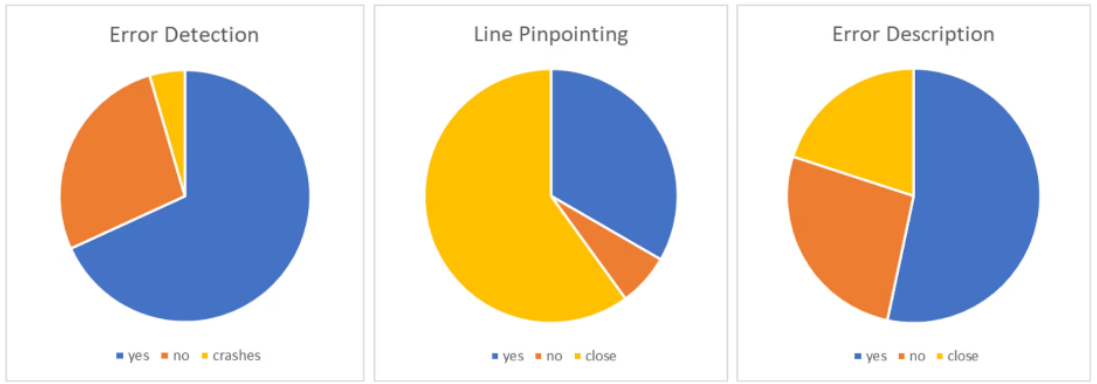


VSCode Plugin

- Syntax support
- Integrated with Planning.Domains
- 100% syntax error detection
- 75% location guidance
- 42% error description
- Semantics error detection: unused predicates

PANDA

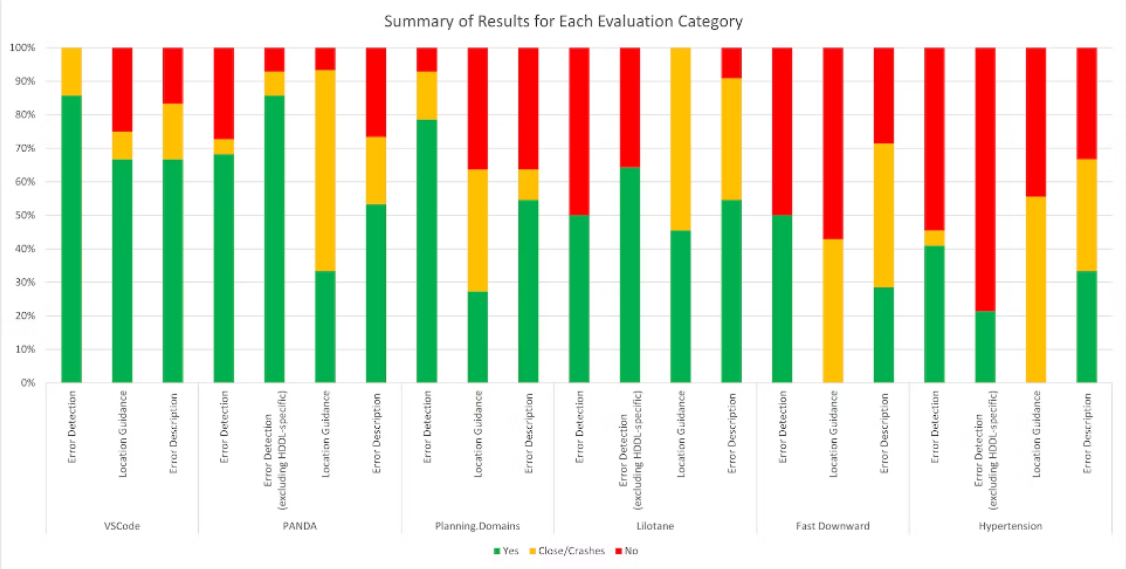
- Planning and Acting in a Network Decomposition Architecture
- 68% syntax error detection rate (86% of non-hierarchical errors)
- 93% location guidance
- 53% error description



more in the paper :)

Conclusion

- More robust error checking required
  - Average detection rate of syntax errors 65%
  - Almost no detection of potential semantics errors
- More helpful error diagnosis required
  - Average 78% location guidance, 48% clear error messages



Future Work

- parser code to catch undetected errors (PANDA developers have already fixed 3)
- fix unhelpful error descriptions
- improve test domains
- cloud-based hierarchical planner
- repeat for problem errors

Acknowledgements

I would like to thank ...

- Pascal Bercher
- Bernd Schattenberg
- Thomas Willingham and Gregor Behnke
- Yikai Ge and Yuchen Fang
- Dan Macdonald

Questions

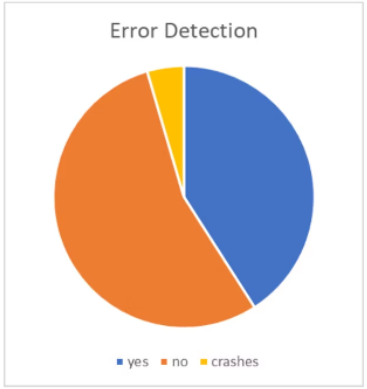
Appendix

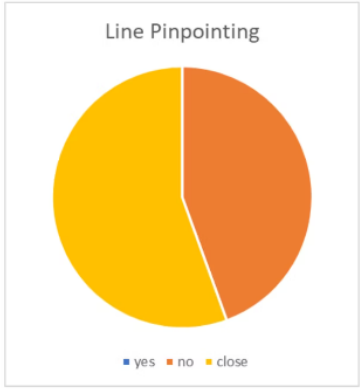
Lilotane

- Runner up of IPC 2020 total order track
- 50% syntax error detection
- 100% location guidance
- 55% error description

Hypertension

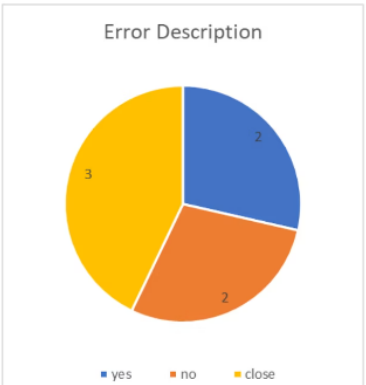
- Winner of IPC 2020 total order track
- 41% syntax error detection
- 56% location guidance
- 33% error description





FastDownward

- Only 50% syntax errors
- 27% location guidance
- 18% helpful error message



Citations

Images generated by *Tome,* 13 May 2023, https://tome.app/