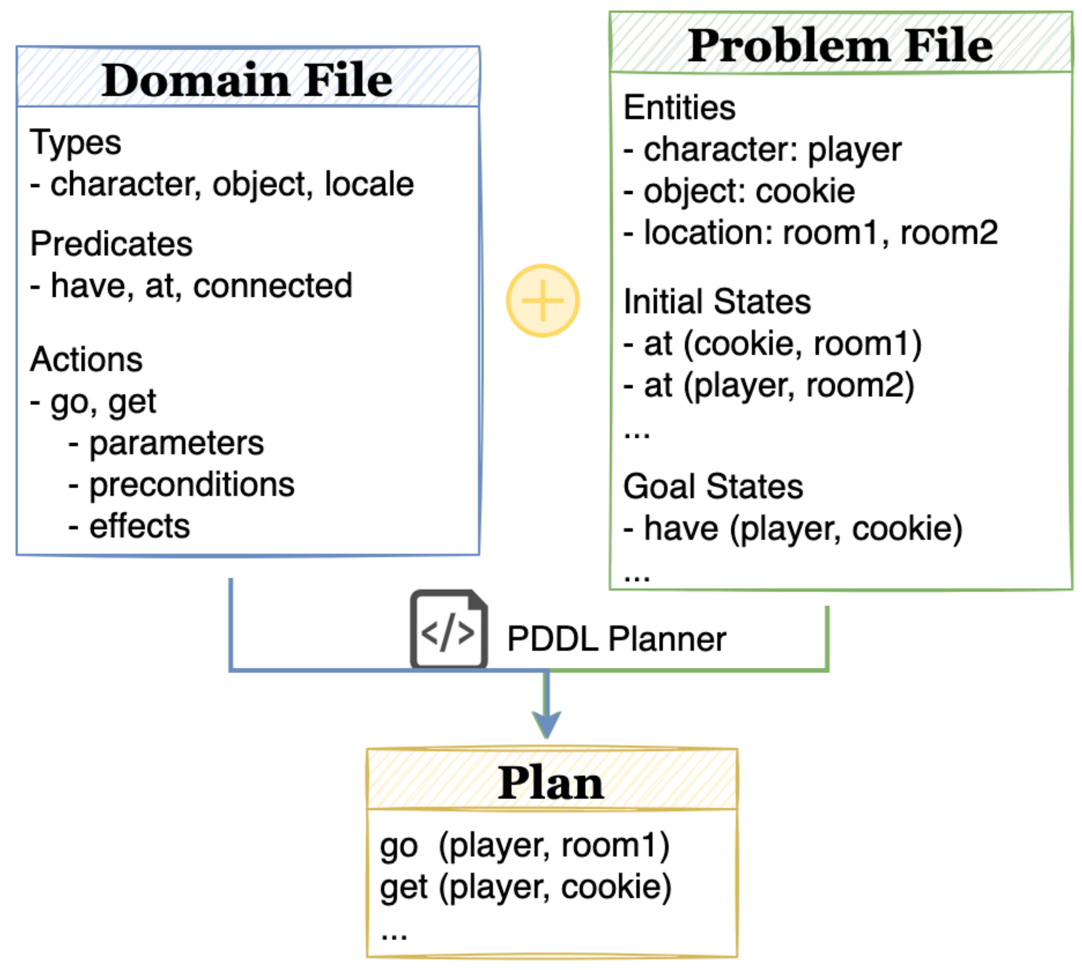
PROC2PDDL: Open-Domain Planning Representations from Texts

Tianyi Zhang* Li Zhang* Zhaoyi Hou Ziyu Wang Yuling Gu
Peter Clark Chris Callison-Burch Niket Tandon

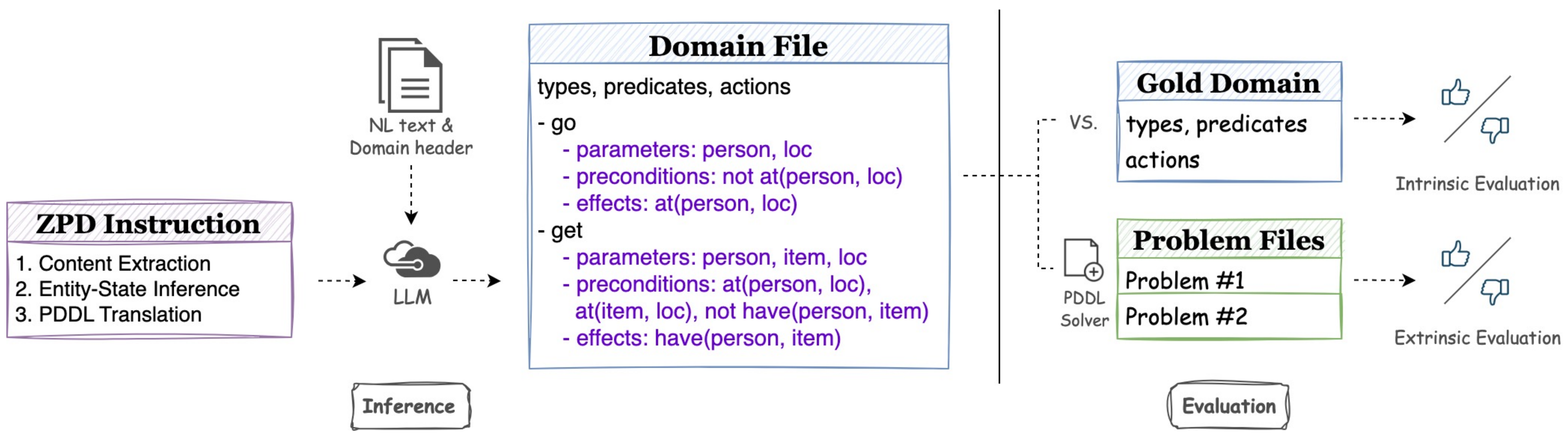
University of Pennsylvania Allen Institute for Artificial Intelligence

Task: Natural Language to Symbolic Language (PDDL) Translation

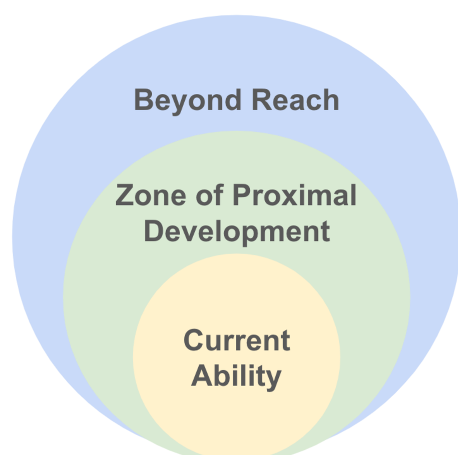
- Comparison of Language Reasoning Approaches:
  - Natural language reasoning using black-box LMs is often unreliable.
  - In contrast, symbolic reasoning with executable code is reliable.
- Motivation:
  - Abundant action descriptions in NL vs. Limited domain actions in PDDL
  - LMs' strong common knowledge + faithful planning ability of PDDL solver
- Previous work:
  - Robotics: infer the domain actions from obtained action-state sequences
  - NLP: generate partial problem states by conditioning on natural language text
- Our work:
  - Automatically generate domain actions from open-domain natural language procedure



Methods: Zone of Proximal Development Scaffolding on Task Skills



- Approach:
  - zone of proximal development **(ZPD)** scaffolding – dissect the **skills**:
    entity-state extraction → inference → PDDL translation
  - Chain-of-thought **(CoT)** prompting – dissect the **components**:
    parameters → precondition → effect



- Evaluation:
  - Intrinsic: comparison with gold domain actions
  - Extrinsic: applying predicted domain actions to solve gold problem files

Example – action 'get_water'

Input – wikiHow text

```
Find a fresh water source. Go inland and try
to find a water source from a stream or
waterfall on the island.
```

Intermediate step – entity-state extraction and inference

```
player:
  before: Is searching for water.
  after: Has water in the inventory.
water:
  before: At a location with a water source.
  after: Collected by the player.
location:
  before: Location with a water source.
  after: Unchanged.
```

Output - PDDL

```
(:action get_water
    :parameters (?player - player ?loc -
location ?water - water)
    :precondition (has_water_source ?loc) (at
?player ?loc)
    :effect (inventory ?player ?water)
```

Evaluations: Evident Improvement through Our ZPD Method

Model %	Intrinsic action acc.	Extrinsic PF solve
gpt-3.5	0.2	1.0
gpt-4	15.9	33.7
+ CoT	9.3	21.1
+ ZPD	**18.1**	**35.8**
+ ZPD, 3 shot	11.9	23.2
gpt-4o	18.2	37.9
+ CoT	19.5	33.7
+ ZPD	**21.4**	**45.3**
+ ZPD, 3 shot	20.3	40.0
gold	100	100

Model %	Parameter	Precondition	Effect
gpt-4	36.7	31.1	**53.0**
+ CoT	29.7	25	54.7
+ ZPD	**42.2**	29.7	48.1
gpt-4o	45.1	31.1	**62.5**
+ CoT	52.4	34.2	54.1
+ ZPD	**53.5**	**40.1**	53.5

- Prompt Instruction:
  - **ZPD** is superior to **CoT** both intrinsically and extrinsically
  - **Few-shot** is ineffective due to our task requirements
- Action Generation:
  - Entity-state extraction and inference occasionally miss **entities**, e.g. implicit tools
  - Translation of **predicates** is inaccurate sometime
    - Wrong matches for equivalent semantics: e.g. (has_fire ?loc) = (at ?loc ?fire)
    - Inconstant expression of variables: e.g. ?f - fruit (variable) , fruit (constant)
  - Precondition is harder to predict than effect (more complex and less obvious predicates)