A Survey of Robotic Language Grounding: Tradeoffs between Symbols and Embeddings

Vanya Cohen, Jason Xinyu Liu, Raymond Mooney, Stefanie Tellex, David Watkins

IJCAI 2024 Survey Track
August 8th, 2024
Robotic Language Grounding

Connect linguistic elements in language to the robot’s perception of and actions in the physical world.
Robotic Language Grounding

Connect linguistic elements in language to the robot’s perception of and actions in the physical world.

1. What grounding representation to use?
Robotic Language Grounding

Connect linguistic elements in language to the robot’s perception of and actions in the physical world.

1. What grounding representation to use?

2. How to ground natural language to the grounding representation of choice?
Robotic Language Grounding
Grounding Language to Symbols

Symbols

- Discrete
- More Structure; More bias
- Unambiguous
- Verifiable
- Interpretable

Grounding Language to Embeddings

Symbols
- Discrete
- More Structure; More bias
- Unambiguous
- Verifiable
- Interpretable

High-dimensional Embeddings
- Continuous
- Less structure; More variance
- Adaptive

Grounding Language to Logic

"Go to the green room."

"Go to the store on Main Street, and always avoid the parking lot."

First Order Logic

Linear Temporal Logic

in(\text{the}(x), \text{the}(\text{agent}(x)), \text{the}(\lambda y. \text{green}(y) \land \text{room}(y)))

\logic{\text{walmart}} \land \neg \logic{\text{parking_lot}}

**Input Command**

**Target Representation**

**Input Command**

- "Go to the green room."
- "Go to the store on Main Street, and always avoid the parking lot."

**Target Representation**

- First Order Logic
  - \text{in(\text{the}(x), \text{the}(\text{agent}(x)), \text{the}(\lambda y. \text{green}(y) \land \text{room}(y))))}
  - \logic{\text{walmart}} \land \neg \logic{\text{parking_lot}}

**Comparison**

- **LEFT**
  - More Structure
  - Hsu et al., 2023

- **Lang2LTL**
  - More Bias
  - Liu et al., 2023b

Grounding Language to Logic

Pros
• Unambiguous semantics
• Verifiable
• Interpretable
• Reduce search space
Grounding Language to Logic

Pros
- Unambiguous semantics
- Verifiable
- Interpretable
- Reduce search space

Cons
- Require manually defined structures
- Difficult to represent low-level control
Grounding Language to Logic: Lang2LTL

Lang2LTL
• Natural language navigation command
• Modular system produces a grounded linear temporal logic (LTL) formula
• Given MDP definition
• Planner outputs a trajectory
Grounding Language to Logic

More Papers

- AutoTAMP: Autoregressive Task and Motion Planning with LLMs as Translators and Checkers [Chen et al. 2024]
- NL2TL: Transforming Natural Languages to Temporal Logics using Large Language Models [Chen et al. 2023]
- NL2LTL: a Python Package for Converting Natural Language (NL) Instructions to Linear Temporal Logic (LTL) Formulas [Fuggitti and Chakraborti 2023]
Grounding Language to PDDL

"Go to the green room."

First Order Logic

in(the(λx.agent(x)),
the(λy.green(y) ∧ room(y)))

"Go to the store on Main Street, and always avoid the parking lot."

Linear Temporal Logic

(walmart) ∧ (∃(parking_lot)

"Move the novel onto the table."

PDDL

Initial:
(ontable laptop)
(on novel laptop)
Goal:
(ontable novel)

Input Command

Target Representation

Discrete
More Structure
More Bias

LEFT
Hsu et al., 2023

Lang2LTL
Liu et al., 2023b

LLM+P
Liu et al., 2023

Grounding Language to PDDL

Pros

• Sound
• Complete
• (Often) Optimal
Grounding Language to PDDL

Pros
- Sound
- Complete
- (Often) Optimal

Cons
- Require manually defined structures
Grounding Language to PDDL: LLM+P

LLM+P

• Natural language description of a planning problem
• LLM translates it to PDDL problem
• Given a PDDL domain description, i.e., action preconditions and effects
• Symbolic planner solves PDDL
Grounding Language to PDDL

More Papers

- Translating Natural Language to Planning Goals with Large-Language Models [Xie et al. 2023]
- Structured, Flexible, and Robust: Benchmarking and Improving Large Language Models Towards More Human-like Behavior in Out-of-distribution Reasoning Tasks [Collins et al. 23]
- Leveraging Pre-trained Large Language Models to Construct and Utilize World Models for Model-based Task Planning [Guan et al. 2023]
- PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change [Valmeekam et al. 2023]
Grounding Language to Code

- "Go to the green room."
  - First Order Logic:
    - \( \text{in}(\text{the}(\lambda x. \text{agent}(x)), \text{the}(\lambda y. \text{green}(y) \land \text{room}(y))) \)
- "Go to the store on Main Street, and always avoid the parking lot."
  - Linear Temporal Logic
- "Move the novel onto the table."
  - PDDL
  - Initial:
    - (on table laptop)
    - (on novel laptop)
  - Goal:
    - (on table novel)
- "Pick up the largest block."
  - Code: Python, JS...
  - \( \text{big\_block} = \max(\text{blocks}, \lambda x: \text{key}(\lambda x: \text{x.volume}()) \lambda x: \text{pick\_up}(\text{big\_block})) \)

Discrete
- LEFT
  - Hsu et al., 2023
More Structure
- Lang2LTL
  - Liu et al., 2023b
More Bias
- LLM+P
  - Liu et al., 2023
Code as Policies
- Liang et al., 2022

Grounding Language to Code

Pros

• Flexible
• High-level plan and low-level control
Grounding Language to Code

Pros
- Flexible
- High-level plan and low-level control

Cons
- Require predefined perception and control models in specific domains
Grounding Language to Code: Code as Policies

Code as Policies

• Natural language command
• Given predefined perception and control models
• Code-writing LLM outputs executable code
Grounding Language to Code

More Papers

- Embodied AI with Two Arms: Zero-shot Learning, Safety and Modularity [Varley et al. 2024]
- ProgPrompt: Generating Situated Robot Task Plans using Large Language Models [Singh et al. 2023]
- Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language [Zeng et al. 2023]
- ITP: Interactive Task Planning with Language Models [Li et al. 2023]
- Voyager: An Open-ended Embodied Agent with Large Language Models [Wang et al. 2023]
Grounding Language to Predefined Skills

**Input Command**
- "Go to the green room."
- "Go to the store on Main Street, and always avoid the parking lot."
- "Move the novel onto the table."
- "Pick up the largest block."
- "I spilled my drink, can you help?"

**Target Representation**
- First Order Logic
- Linear Temporal Logic
- PDDL
- Code: Python, JS...
- Predefined Skills

**Discrete**
- LEFT
- More Structure
- More Bias
- Hsu et al., 2023
- Lang2LTL
- Liu et al., 2023b
- LLM+P
- Liu et al., 2023
- Code as Policies
- Liang et al., 2022
- SayCan
- Ahn et al. 2022

**Examples**
- **LEFT**
  - in(the(λx.agent(x)), the(λy.green(y) ∧ room(y)))

- **Lang2LTL**
  - EV(walmart) ∧ ¬(parking_lot)

- **LLM+P**
  - Initial: (on novel laptop) (on laptop)
  - Goal: (ontable novel)

- **Predefined Skills**
  - big_block = max(blocks, key=lambda x: x.volume())
  - pick_up(big_block)

  **Steps**
  1. find a sponge
  2. pick up the sponge
  3. bring it to you
Grounding Language to Predefined Skills

Pros
• Adaptive
Grounding Language to Predefined Skills

Pros
- Adaptive

Cons
- Require predefined skills
- Possibly incorrect plans
Grounding Language to Predefined Skills: SayCan

SayCan
• Natural language command
• LLM proposes candidate skills every step
• Pretrained value functions to rank available skills
• Language-conditioned policies execute the top skill
Grounding Language to Predefined Skills

More Papers

- CAPE: Planning with Large Language Models via Corrective Re-prompting [Raman et al. 2024]
- Inner Monologue: Embodied Reasoning through Planning with Language Models [Huang et al. 2022]
- Language Models as Zero-shot Planners: Extracting Actionable Knowledge for Embodied Agent [Huang et al. 2022]
Grounding Language to Subgoals

Grounding Language to Subgoals

Pros

• Adaptive
Grounding Language to Subgoals

Pros
- Adaptive

Cons
- Require predefined skills
- Possibly incorrect plans
Grounding Language to Subgoals: VLP

Video Language Planning (VLP)
- Tree search
- VLM proposes language subgoals
- Video model conditioned on text generates image subgoals
- Policy conditioned on image executes the plan
Grounding Language to Subgoals

More Papers

- Zero-Shot Robotic Manipulation with Pretrained Image-Editing Diffusion Models [Black et al. 2023]
- UniSim: A Neural Closed-Loop Sensor Simulator [Yang et al. 2023]
- GAIA-1: A Generative World Model for Autonomous Driving [Hu et al. 2023]
Grounding Language to Embeddings
Grounding Language to Embeddings

Pros
- Adaptive
Grounding Language to Embeddings

Pros
- Adaptive

Cons
- Large training set and compute
- Possibly incorrect actions
Grounding Language to Embeddings: VIMA

VIMA
- Tokenize multimodal input
- Transformer architecture
- Output end-effector poses
Grounding Language to Embeddings

More Papers

- Octo: An Open-Source Generalist Robot Policy [Octo Model Team 2024]
- RT-1: Robotics Transformer for Real-World Control at Scale [Brohan et al. 2023]
- PaLME: an Embodied Multimodal Language Model [Driess et al. 2023]
- Vision-Language Foundation Models as Effective Robot Imitators [Li et al. 2023]
- GATO: A Generalist Agent [Reed et al. 2022]
- Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation [Shridhar et al. 2022]
- Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos [Baker et al. 2022]
Language Grounding for Robots

Language Grounding for Robots

Discrete Symbols
- Logic
- Planning domain definition language (PDDL)
- Code
- Descriptions of predefined skills

Language Grounding for Robots

Discrete Symbols
- Logic
- Planning domain definition language (PDDL)
- Code
- Descriptions of predefined skills

High-dimensional Embeddings
- Language and image subgoals
- Neural embeddings
Open Problems and Future Directions
Open Problems and Future Directions

- **Neuro-symbolic Approach**
  - POMDP and PDDL planners
  - Deep learning models with generalizable representations
  - E.g., Jointly learn symbols in the embedding space and skills
Open Problems and Future Directions

• Neuro-symbolic Approach
  • POMDP and PDDL planners
  • Deep learning models with generalizable representations
  • E.g., Jointly learn symbols in the embedding space and skills

• Multimodal Dataset
  • E.g., text, audio, RGB images, point clouds, voxels, videos, demonstrations
  • Semantically diverse
Open Problems and Future Directions

• Neuro-symbolic Approach
  • POMDP and PDDL planners
  • Deep learning models with generalizable representations
  • E.g., Jointly learn symbols in the embedding space and skills

• Multimodal Dataset
  • E.g., text, audio, RGB images, point clouds, voxels, videos, demonstrations
  • Semantically diverse

• Modular Approach
  • Existing robot modules
  • E.g., SLAM, motion planning and object detection
Open Problems and Future Directions

• Neuro-symbolic Approach
  • POMDP and PDDL planners
  • Deep learning models with generalizable representations
  • E.g., Jointly learn symbols in the embedding space and skills

• Multimodal Dataset
  • E.g., text, audio, RGB images, point clouds, voxels, videos, demonstrations
  • Semantically diverse

• Modular Approach
  • Existing robot modules
  • E.g., SLAM, motion planning and object detection

• Verification and Safety
  • Formal methods
## Conclusion

### A Survey of Robotic Language Grounding: Tradeoffs between Symbols and Embeddings

<table>
<thead>
<tr>
<th>Input Command</th>
<th>Target Representation</th>
<th>Method</th>
<th>Code as Policies</th>
<th>Predefined Skills</th>
<th>Natural Language</th>
<th>Vision</th>
<th>Vector</th>
<th>Model Activations</th>
</tr>
</thead>
</table>
| "Go to the green room." | in(the( Ax.agent(x)), the(Ay.green(y) ∨ room(y))) | First Order Logic | Li et al., 2023 | 1. find a sponge, 2. pick up the sponge, 3. bring it to you | Steps: 
Step 1: chop tree block 
Step 2: craft wooden stick... | Images | Models | Activations |
| "Go to the store on Main Street, and always avoid the parking lot." | (walmart) ∧ ¬(parking_lot) | Linear Temporal Logic | Liu et al., 2023b | big_block = max(blocks, key= lambda x: x.volume()) pick_up(big_block) | | | |
| "Move the novel onto the table." | PDDL | PDDL | Li et al., 2023 | Goal: (ontable novel) | | | |
| "Pick up the largest block." | Code: Python, JS... | Code as Policies | Liang et al., 2022 | | | | |
| "I spilled my drink, can you help?" | Predefined Skills | SayCan | Ahn et al. 2022 | 1. find a sponge, 2. pick up the sponge, 3. bring it to you | | | |
| "Craft a diamond pick-axe." | Natural Language | VLP | Du et al., 2023 | | | | |
| | Vision | VIMA | Jiang et al., 2023 | | | | |

**Poster Location:**

E15

Jason Xinyu Liu

https://jasonxylie.github.io/xinyu_liu@brown.edu

https://arxiv.org/abs/2405.13245