Grounding Complex Natural Language Commands for Temporal Tasks in Unseen Environments

Jason Liu* (xinyu_liu@brown.edu), Ziyi Yang*, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah

Motivation

Existing language grounding systems
- require retraining in novel environments
- do not provide safety guarantees

Problem Definition

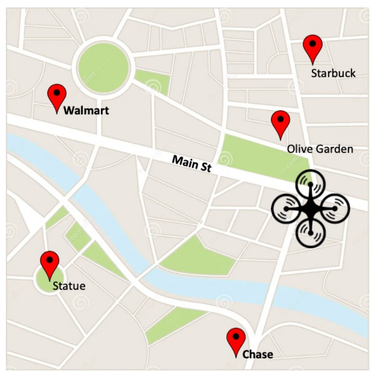
Input
- Natural language navigational command
- Semantic database of the environment

Output
- Linear temporal logic (LTL) task specification whose propositions are grounded to real-world landmarks

Main Contributions

- A modular system Lang2LTL that grounds navigational commands in novel environments without retraining
- Provide safety guarantees
- Generalization tests for language grounding systems

Semantic Database



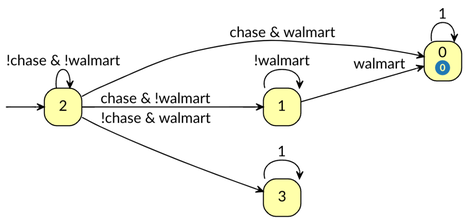

```
proposition: walmart
{
    "Walmart": {
        "addr:housenumber": "692",
        "shop": "supermarket",
        "opening_hours": "Mo-Su 08:00-20:00",
        "phone": "6173389788",
        "addr:postcode": "02111",
        "addr:street": "Main Street"
    },
    ...,
}
```

Task Specification: Linear Temporal Logic (LTL)

- LTL can represent non-Markovian tasks
- LTL = propositional logic + temporal operators
- Syntax

$$\varphi := \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi_1 \mathbf{U} \varphi_2$$

- Labeling function maps MDP states to propositions
- Translate LTL to Büchi automaton to track task progress

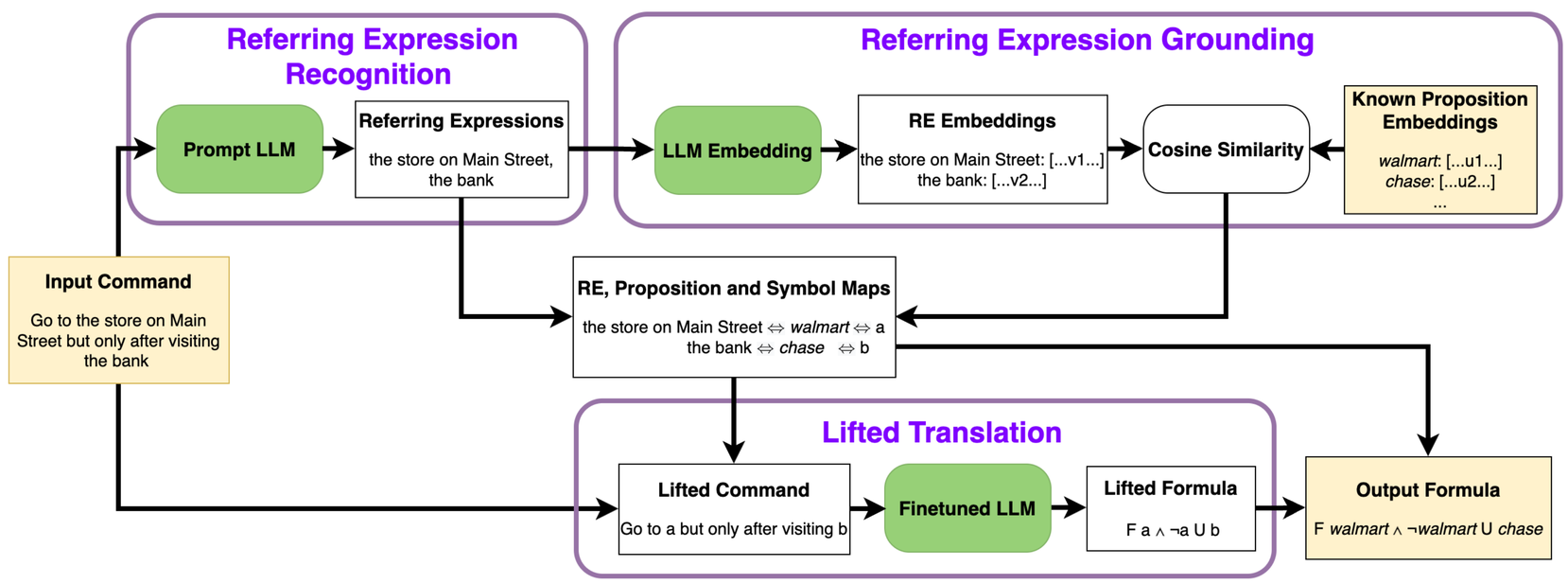


Example LTL Tasks

Type	Command	LTL
Visit	go to a and b in any order	*F*a ∧ *F*b
Sequenced Visit	move to a then b	*F*(a ∧ *F*b)
Ordered Visit	visit b but only after a	¬b *U* a ∧ *F*b
Patrolling	keep visiting a and b	*GF*a ∧ *GF*b
Global Avoidance	never visit a or b	*G* ¬a ∧ *G* ¬b
Lower Restricted Avoidance	visit a at least twice	*F*(a ∧ (a *U* (¬a ∧ (¬a *U* *F*a))))

https://lang2ltl.github.io/

Modular System: Lang2LTL



Generalization Tests

Robustness to Paraphrasing
- Go to *chase* Visit *chase*
- *F chase*

Robustness to Substitutions
- Training: *F* chase; *G* ¬ walmart
- Test: *F* walmart

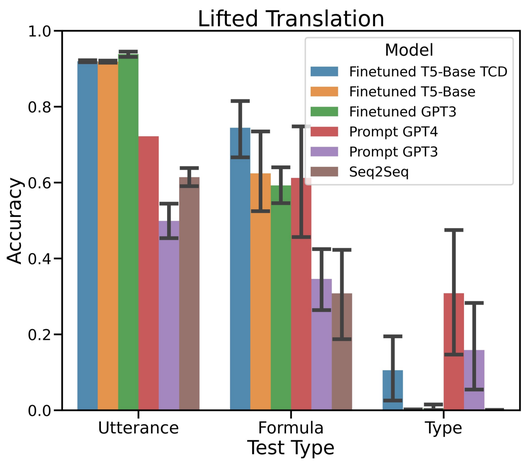
Robustness to Vocabulary Shift
- Training: *F* chase
- Test: *F* walmart

Robustness to Unseen Formulas
- Training: *F* chase; *F* walmart
- Test: *F* walmart ∧ *F* chase

Robustness to Unseen Templates
- Training: *F* chase; *F* walmart ∧ *F* chase
- Test: *F* chase ∧ (¬ walmart *U* chase)

Module-Wise Evaluation

Component	Accuracy
RE Recognition	$98.01 \pm 2.08\%$
RE Grounding	$98.20 \pm 2.30\%$

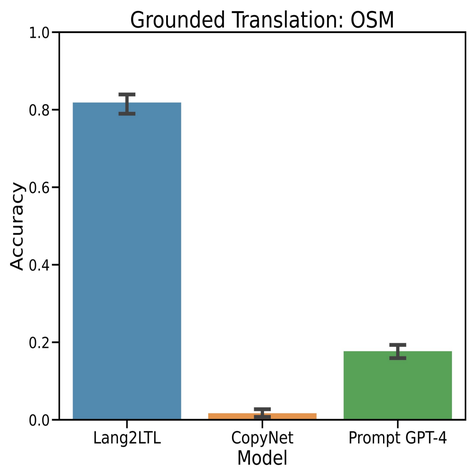


Cross-Domain Generalization

	OSM [44]	*CleanUp* [3]
Lang2LTL	$49.40 \pm 15.49\%$*	$78.28 \pm 1.73\%$*
CopyNet [44]	$45.91 \pm 12.70\%$	2.57%*
RNN-Attn [3]	NA*	$95.51 \pm 0.11\%$

OpenStreetMap

- Lang2LTL generalizes to 21 novel cities
- Grounding accuracy: 81.83% ± 8.22%



Robot Demonstration

Navigational Command	Lang2LTL Result
go to brown bookshelf, metal desk, wooden desk, kitchen counter, and the blue couch in any order	Success
move to grey door, then bookshelf, then brown desk, then counter, then white desk	Success
visit brown wooden desk but only after bookshelf	Success
visit bookshelf at most three times	Success
go to wooden desk exactly three times	Success
go to doorway exactly two times, in addition always avoid the table	Success
Go to the counter, but never visit the counter	Abort Correctly (contradiction)
find the kitchen counter, in addition avoid the doorway	Abort Correctly (environment)
visit counter at least six times	Abort Correctly (incorrect grounding)