Few-shot Object Reasoning for Robot Instruction Following

Yoav Artzi

Workshop on Spatial Language Understanding
EMNLP 2020
Task

- Navigation between landmarks
- Agent: quadcopter drone
- Inputs: poses, raw RGB camera images, and natural language instructions
Task

go straight and stop before reaching the planter
turn left towards the globe and go forward until just before it
Mapping Instructions to Control

• The drone maintains a **configuration** of target velocities

  \[(v, \omega)\]

• Each action updates the configuration or stops

• Goal: learn a mapping from inputs to configuration updates

\[f(go\ straight\ and\ stop\ before\ reaching\ the\ planter, \ turn\ left\ globe) = STOP v_t, \omega_t\]
Modular Approach

- Build/train separate components
- Symbolic meaning representation
- Complex integration

Instruction

Language Understanding

Planning

Control

Perception → Mapping
Single-model Approach (a.k.a. end-to-end)

How to think of extensibility, interpretability, and modularity when packing everything in a single model?
Single-model Approach

- **Extensibility**: extending the model to reason about new object after training
- **Interpretability**: viewing how the model reasons about object grounding and trajectories
- **Modularity**: re-using parts of the model

Within a representation learning framework
Representation: Design vs. Learning

• Systems that use symbolic representations are interpretable and (potentially) extensible

• However: representation design of every possible concept is brittle and hard to scale

• Instead: design the most general concepts and let representation learning fill them with content

• Today, two concepts: objects and trajectories
Today

Few-shot instruction following:

- Few-shot language-conditioned object segmentation
- Object context mapping
- Integration into a visitation-prediction policy for mapping instructions to drone control
Language-conditioned Object Segmentation

• Input: instruction and observation images

• Goal: identify and align objects and references

go straight and stop before reaching the planter turn left towards the globe and go forward until just before it
Few-shot Version

• Input: instruction, observation images, and database

• Goal: identify previously unseen objects and mentions and align them

Database

go straight and stop before reaching the planter turn left towards the globe and go forward until just before it
Alignment via a Database

• Approach: align observations and references through the database

• Adding objects to the database extends the alignment ability

• Requires only adding a few image and language exemplars
Alignment via a Database

• Approach: align observations and references through the database

• Adding objects to the database extends the alignment ability

• Requires only adding a few image and language exemplars
Align straight and stop before reaching the planter. Turn left towards the globe and go forward until just before it.
Alignment Score

go straight and stop before reaching the planter

turn left towards the globe and go forward until just before it

\[
\text{ALIGN}(b, r) = \sum_o \frac{P(o \mid b) P(b) P(o \mid r)}{P(o)}
\]

Database

Object record

\(b\) Bounding box

\(r\) Reference

\(o\) Database object
Alignment Score

\[ \text{ALIGN}(b, r) = \sum_o \frac{P(o \mid b) P(b) P(o \mid r)}{P(o)} \]

- Region proposal network gives bounding boxes and \( P(b) \)
- \( P(o) \) is uniform
Alignment Score

\[
\text{ALIGN}(b, r) = \sum_{o} \frac{P(o \mid b) P(b) P(o \mid r)}{P(o)}
\]

- \(P(o \mid b)\) is computed using visual similarity
- Using Kernel Density Estimation with a symmetric multivariate Gaussian kernel
- \(P(o \mid r)\) is computed similarly using text similarity with pre-trained embeddings
Mask Refinement

• Refine each bounding box with a UNet model
• Gives a tight object mask
• Paired with a bounded alignment score to a reference in the text

go straight and stop before reaching the planter

turn left towards the globe and go forward until just before it

Align = 0.7
Learning

\[ \text{ALIGN}(b, r) = \sum_o \frac{P(o \mid b)P(b)P(o \mid r)}{P(o)} \]

- Region proposal network parameters for bounding box proposal
- Image similarity measure for \( P(o \mid b) \)
- UNet parameters for mask refinement
- Text similarity uses pre-trained embeddings
- Challenge: need large-scale heavily annotated visual data

**Annotations**

- \( b \) Bounding box
- \( r \) Reference
- \( o \) Database object
Augmented Reality Training Data

FPV  Overlay  Composite  Mask labels
Augmented Reality Training Data

Large-scale generation with ShapeNet objects

Learned representations generalize beyond specific objects for:

- Region proposal network for bounding boxes
- Image similarity measure for $P(o \mid b)$
- UNet parameters for mask refinement

Composite Mask labels
Today

Few-shot instruction following:

• Few-shot language-conditioned object segmentation

• Object context mapping

• Integration into a visitation-prediction policy for mapping instructions to drone control
Object Context Mapping

Goal: create maps that capture object location and the instruction behavior around objects

1. Identify and align object mentions to observations

2. Compute abstract contextual representations for object references

3. Project and aggregate masks over time

4. Combine aggregated masks with contextual representations to create a map
Object Context Mapping

Step I: Identify and Align

- Bounding box proposals from Region Proposal Network
- Object references from tagger
- Align with language-conditioned segmentation and the database
- To compute: first-person masks aligned to instruction references
Object Context Mapping
Step II: Abstract Contextual Representations

- Replace references with object placeholders
- Compute bi-directional RNN representations for all tokens
- The hidden state for each placeholder is the object context representation

… reaching the planter turn left towards the globe and …

… reaching ObjectA left towards ObjectB and …

Abstract references
Object Context Mapping
Step III: Projection and Aggregation

- Projection from first-person camera masks to third-person environment ground with pinhole camera model
- Deterministic aggregation

Go straight and stop before reaching the planter. Turn left towards the globe and go forward until just before it.
Object Context Mapping

Step III: Projection and Aggregation

- Projection from first-person camera masks to third-person environment ground with pinhole camera model
- Deterministic aggregation
Object Context Mapping

Step III: Projection and Aggregation

- Projection from first-person camera masks to third-person environment ground with pinhole camera model
- Deterministic aggregation

\[
\sum \text{Projected Masks (time } t) \rightarrow \text{Integrator} \rightarrow \text{Masks (time } t-1) \rightarrow \text{Masks (time } t)\]
Step IV: Combine Object Representations

- Each position is a product of a mask value and its aligned object context representation

... reaching ObjectA left towards ObjectB and ...
Object Context Map

- Map information abstracts over reference content stripped from instruction
- Includes for each object the context of its reference in the instruction
- Tells the agent how to behave around the object
- Policy remains blind to the object itself
Today

Few-shot instruction following:

• Few-shot language-conditioned object segmentation

• Object context mapping

• Integration into a visitation-prediction policy for mapping instructions to drone control
Two-stage Policy

1. Map and predict states likely to visit + track observability
2. Generate actions to visit high-probability states and explore
Visitation Distributions

• The state-visitation distribution $d(s; \pi, s_0)$ is the probability of visiting state $s$ following policy $\pi$ from start state $s_0$

• Predicting $d(s; \pi^*, s_0)$ for an expert policy $\pi^*$ tells us the states to visit to complete the task

• We compute two distributions: trajectory-visitation and goal-visitation
Visitation Distributions

• Distributions reflect the agent plan

• Model path and goal observability

• Refined as observing more of the environment
Stage I: Mapping and Plan Generation

- Few-shot language-conditioned segmentation to construct an object context map
- Predict distribution over map positions
Plan Generation

- Cast distribution prediction as image generation
- LingUUnet: an image-to-image encoder-decoder
- Visual reasoning at multiple image scales
- Conditioned on language input at all levels of reasoning using text-based convolutions
LingUNet

- Object Map
- Convolutions
- Instruction
- RNN
- Visitation Distributions
- SoftMax
- Text Kernels
- Deconvolutions

Text Convolutions

Convolutions
Two-stage Policy

1. Map and predict states likely to visit + track observability
2. Generate actions to visit high-probability states and explore
Stage II: Action Generation

- Relatively simple control problem without language
- Transform and crop to agent perspective and generate configuration update
Training

Instruction

Few-shot Segmentation

Mapping

RNN

Plan Generation

LingUNet

Control Network

CNN+MLP

Trained separately

Abstract Instruction

Object Context Map

Mask

Visitations
Training in Simulation

- Language-conditioned segmentation trained separately for simulation and real environment
- Policy training does not require access to real world
- After training: swap the segmentation component
- Data: demonstrations and experience

Go between the mushroom and flower chair the tree all the way up to the phone booth
SuReAL
Supervised and Reinforcement Asynchronous Learning

Instruction

Supervised Learning

Reinforcement Learning

Few-shot Segmentation

Trained separately

Mapping

RNN

Plan Generation

LingUNet

Control Network

CNN+MLP

Abstract Instruction

Object Context Map

Mask

Visitations
Supervised Learning

Objective: generate visitation distributions
Data: simulation states paired with visitation predictions

Instruction

Few-shot Segmentation
Trained separately

Mapping

Plan Generation

RNN

LingUNet

Cross-entropy loss

Demonstration Visitations
RL for Control

Instruction

Few-shot Segmentation

Trained separately

Mapping

RNN

Plan Generation

LingUNet

Control Network

CNN+MLP

Abstract Instruction

Object Context Map

Mask

Visitations

Reinforcement Learning

Intrinsic reward
SuReAL
Supervised and Reinforcement Asynchronous Learning

Supervised Learning
- Instruction
- Few-shot Segmentation (Trained separately)
- Mapping (RNN)
- Plan Generation (LingUNet)

Periodic parameter updates

Sampled action sequences

Reinforcement Learning
- Control Network (CNN+MLP)
SuReAL
Supervised and Reinforcement Asynchronous Learning

• **Stage I:** learn to predict visitation distributions based on noisy predicted execution trajectories

• **Stage II:** learn to predict actions using predicted visitation distributions

Periodic parameter updates
Replace gold action sequences with sampled
Experimental Setup

- Intel Aero quadcopter
- Vicon motion capture for pose estimate
- Simulation with Microsoft AirSim
- Drone cage is 4.7x4.7m
- All evaluation with eight new objects
- Database includes five images and five phrases for each object
- Training data: 41k instruction-demonstration pairs in simulation, no demonstration data in the real world
Human Evaluation

- Score path and goal on a 5-point Likert scale for 63 examples

- Our model receives 4-5 path scores 53% of the time, double than PVN2-SEEN, showing effective generalization to unknown objects

- Outperforming PVN2-ALL illustrates the benefit of the object-centric inductive bias
Example

and move to the right side of the firetruck
and looping around it until heading towards the globes
left side
keep the red lego to your right as you to around it to face the beige fence pass the strawberry on your left and move forward to stop just past the globe
fly straight at the red lego up ahead stop just in front of the red lego and then veer left in front of the red lego
Today

Few-shot instruction following:

- Few-shot language-conditioned object segmentation
  Modeling objects and aligning their references and observations + training with augmented reality data

- Object context mapping
  Incorporate contextual text information into spatial map without specific object information

- Integration into a visitation-prediction policy for mapping instructions to drone control
  Generate trajectory plans over object context map + train in simulation only by swapping the segmentation component
Some Open Questions

• How to elicit exemplars to add to the database from human users, potentially within interaction?

• How to generalize from objects to more general objects types?

• What other object properties should we model? Such as permanence and reference consistency
The Papers

- **Few-shot Object Grounding for Mapping Natural Language Instructions to Robot Control**
  Valts Blukis, Ross A. Knepper, and Yoav Artzi
  CoRL, 2020

- **Learning to Map Natural Language Instructions to Physical Quadcopter Control Using Simulated Flight**
  Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A. Knepper, and Yoav Artzi
  CoRL, 2019

- **Mapping Navigation Instructions to Continuous Control Actions with Position Visitation Prediction**
  Valts Blukis, Dipendra Misra, Ross A. Knepper, and Yoav Artzi
  CoRL, 2018

- **Following High-level Navigation Instructions on a Simulated Quadcopter with Imitation Learning**
  Valts Blukis, Nataly Brukhim, Andrew Bennett, Ross A. Knepper, and Yoav Artzi
  RSS, 2018.
go straight and stop before reaching the planter turn
left towards the globe and go forward until just before it

And collaborators: Dipendra Misra, Eyvind Niklasson, Nataly Brukhim, Andrew Bennett, and Ross Knepper

Valts Blukis

https://github.com/lil-lab/drif

Thank you! Questions?
[fin]
Object Database
The object database used during development in the physical environment.
The object database used **during testing**, containing previously unseen physical objects.
Visitation Distributions
Visitation Distribution

- Given a Markov Decisions Process:

  - The state-visitation distribution $d(s; \pi, s_0)$ is the probability of visiting state $s$ following policy $\pi$ from start state $s_0$

  - Predicting $d(s; \pi^*, s_0)$ for an expert policy $\pi^*$ tells us the states to visit to complete the task

  - Can learn from demonstrations, but prediction generally impossible: $S$ is very large!
Approximating Visitation Distributions

• Solution: approximate the state space

• Use an approximate state space $\tilde{S}$ and a mapping between the state spaces $\phi : S \rightarrow \tilde{S}$

• For a well chosen $\phi$, a policy $\pi$ with a state-visitation distribution close to $d(\tilde{s}; \pi^*, \tilde{s}_0)$ has bounded sub-optimality
Visitation Distribution for Navigation

- $\tilde{S}$ is a set of discrete positions in the world
- We compute two distributions: trajectory-visitation and goal-visitation

**MDP**

$S$ States $A$ Actions $R$ Reward $H$ Horizon

- Planning with Position Visitation Prediction
- Action Generation

**Stage I**
- Trajectory Probability
- Goal Probability

**Instruction**

**Stage II**
- Action

**Diagram:**
- Instruction flow leading to planning with position visitation.
- Transition from planning to action generation.
Drone Related Work
(Somewhat outdated)
Related Work: Task

- Mapping instructions to actions with robotic agents

- Mapping instruction to actions in software and simulated environments
  MacMahon et al. 2006; Branavan et al. 2010; Matuszek et al. 2010, 2012; Artzi et al. 2013, 2014; Misra et al. 2017, 2018; Anderson et al. 2017; Suhr and Artzi 2018

- Learning visuomotor policies for robotic agents
Related Work: Method

- Mapping and planning in neural networks
  Bhatti et al. 2016; Gupta et al. 2017; Khan et al. 2018; Savinov et al. 2018; Srinivas et al. 2018

- Model and learning decomposition

- Learning to explore
  Knepper et al. 2015; Nyga et al. 2018
Drone Data Collection
Data

• Crowdsourced with a simplified environment and agent

• Two-step data collection: writing and validation(segmentation)

Go towards the pink flowers and pass them on your left, between them and the ladder. Go left around the flower until you're pointed towards the bush, going between the gorilla and the traffic cone. Go around the bush, and go in between it and the apple, with the apple on your right. Turn right and go around the apple.
Data

- Crowdsourced with a simplified environment and agent
- Two-step data collection: writing and validation/segmentation

Go towards the pink flowers and pass them on your left, between them and the ladder. Go left around the flower until you're pointed towards the bush, going between the gorilla and the traffic cone. Go around the bush, and go in between it and the apple, with the apple on your right. Turn right and go around the apple.
CoRL 2018
Experiments
Experimental Setup

- Crowdsourced instructions and demonstrations
- 19,758/4,135/4,072 train/dev/test examples
- Each environment includes 6-13 landmarks
- Quadcopter simulation with AirSim
- Metric: task-completion accuracy
Test Results

- Explicit mapping helps performance
- Explicit planning further improves performance
Synthetic vs. Natural Language

- Synthetically generated instructions with templates
- Evaluated with explicit mapping (Blukis et al. 2018)
- Using natural language is significantly more challenging
- Not only a language problem, trajectories become more complex

![Success Rate Chart](chart.png)

*Synthetic Language: 79.2%  
Natural Language: 24.36%*
Ablations
Development Results

- The language is being used effectively
- Auxiliary objectives help with credit assignment

Success Rate

- Our Approach
- w/o imitation learning
- w/o goal distribution
- w/o auxiliary objectives
- w/o language
Analysis

Development Results

- Better control can improve performance
- Observing the environment, potentially through exploration, remains a challenge

<table>
<thead>
<tr>
<th>Success Rate</th>
<th>Our Approach</th>
<th>Ideal Actions</th>
<th>Fully Observable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40.44</td>
<td>45.7</td>
<td>60.59</td>
</tr>
</tbody>
</table>
CoRL 2019
Experiments
Environment

• Drone cage is 4.7x4.7m
• Created in reality and simulation
• 15 possible landmarks, 5-8 in each environment
• Also: larger 50x50m simulation-only environment with 6-13 landmarks out of possible 63
Data

- Real environment training data includes 100 instruction paragraphs, segmented to 402 instructions
- Evaluation with 20 paragraphs
- Evaluate on concatenated consecutive segments
- Oracle trajectories from a simple carrot planner
- Much more data in simulation, including for a larger 50x50m environment
Evaluation

- Two automated metrics
  - SR: success rate
  - EMD: path earth’s move distance
- Human evaluation: score path and goal on a 5-point Likert scale
Human Evaluation

- Score path and goal on a 5-point Likert scale for 73 examples
- Our model receives five-point path scores 37.8% of the time, 24.8% improvement over PVN2-BC
- Improvements over PVN2-BC illustrates the benefit of SuReAL and the exploration reward
Observability

- Big benefit when goal is not immediately observed

- However, complexity comes at small performance cost on easier examples
## Test Results

<table>
<thead>
<tr>
<th>Success Rate</th>
<th>Average</th>
<th>PVN-BC</th>
<th>PVN2-BC</th>
<th>Our Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>16.7</td>
<td>20.8</td>
<td>29.2</td>
</tr>
<tr>
<td>EMD</td>
<td></td>
<td>0.71</td>
<td>0.61</td>
<td>0.59</td>
</tr>
</tbody>
</table>

- SR often too strict: 30.6% compared to 39.7% five-points on goal
- EMD performance generally more reliable, but still fails to account for semantic correctness
**Simple vs. Complex Instructions**

- Performance on easier single-segment instructions is much higher.
- Instructions are shorter and trajectories simpler.

<table>
<thead>
<tr>
<th></th>
<th>1-segment Instructions</th>
<th>2-segment Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success Rate</td>
<td>56.5</td>
<td>30.6</td>
</tr>
<tr>
<td>EMD</td>
<td>0.34</td>
<td>0.52</td>
</tr>
</tbody>
</table>
Transfer Effects

- Visual and flight dynamics transfer challenges remain
- Even Oracle shows a drop in performance from 0.17 EMD in the simulation to 0.23 in the real environment
CoRL 2019 Examples
Cool Example

once near the rear of the gorilla turn right and head towards the rock stopping once near it
Failure

head towards the area just to the left of the mushroom and then loop around it
CoRL 2019 Sim-real Shift Examples
Sim-real Control Shift

when you reach the right of the palm tree take a sharp right when you see a blue box head toward it
Sim-real Control Shift

make a right at the rock and head towards the banana