Modeling Conceptual Understanding in Image Reference Games

Rodolfo Corona, Stephan Alaniz, Zeynep Akata

Presenter : Serkan Bugur

Authors

Rodolfo Corona

- PhD student at UC Berkeley
- Fullbright Student Research Grant at Amsterdam ML Research Group

Stephan Alaniz

- PhD student at MPI (University of Tubingen)

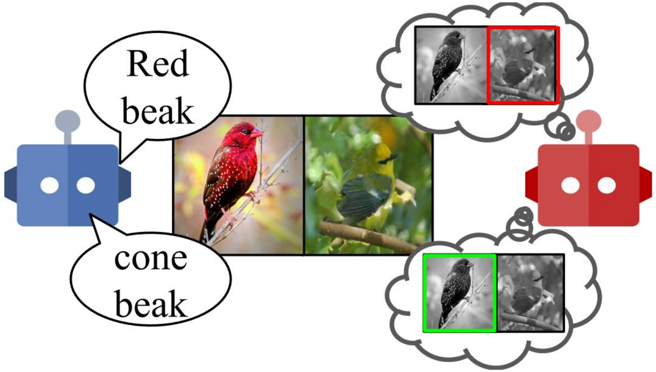
Zeynep Akata

- Professor at University of Tubingen

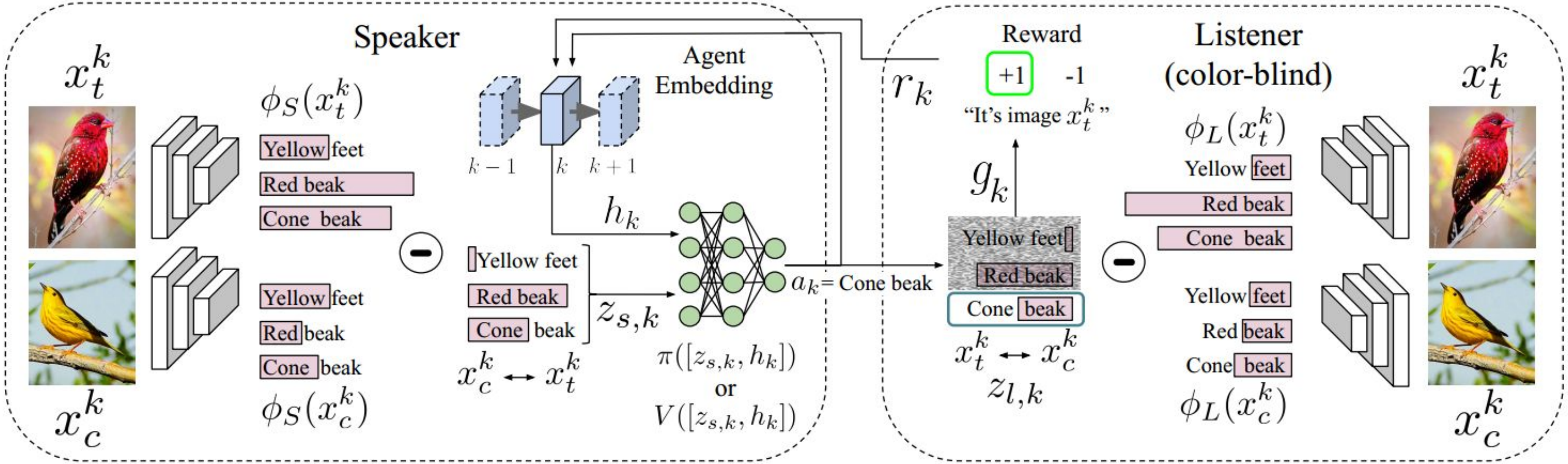
Motivation

- A system interacting with a large population of users.
  - **Variety** of agents' understanding of the world.
  - **Perception** difference, e.g. human & machine.

- An agent capable of forming representations of other agent's understanding;
  - Internal
  - Human-Interpretable



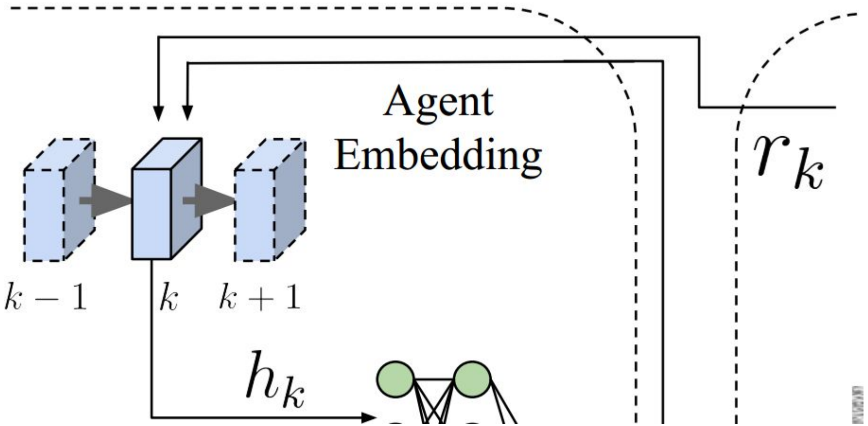
Model



- **Goal:** given image pair (x_t, x_c), speaker communicates attribute a that allows listener to select x_t
- **Episode k:**
  - Speaker and listener encode the image pair using perceptual modules ϕ_S, ϕ_L
  - Speaker selects a target image x_t^k and attribute a_k with its policy conditioned on agent embedding h_k and image features $z_{s,k} = \phi(x_t^k) - \phi(x_c^k)$
  - Given a_k , the listener guesses the target image rationally $g = \arg\max_{x \in \{x_t^k, x_c^k\}} \phi_L^a(x)$
  - Speaker updates embedding h_{k+1} given the reward r_k received for using a_k in that game
- Play N episodes to learn about listener, then play M episodes on which reward is evaluated

Agent Embedding Module

- LSTM's hidden state is used for agent embedding.
- After each episode,
  - Attribute a_k is selected, r_k is received.
  - Generate $o_k \rightarrow$ one-hot vector.
  - Update $\rightarrow h_k = LSTM(h_{k-1}, o_k)$



Attribute Selection Mechanism

- Use parameterized value function $V(s_k, a_k)$ and policy function $\pi_S(s_k, a_k)$ with $s_k = [z_{s,k}; h_k]$

$$a_k = \arg\max_{a \in A} V(s_k, a)$$

- Value-function Loss: $\mathcal{L}_V = \frac{1}{N+M} \sum_{N+M} \mathrm{MSE}(V(s_k, a_k), r_k)$
- **Epsilon Greedy:** random with probability ϵ, otherwise $a_k = \arg\max_{a \in A} V(s_k, a)$
- **Active Policy:** $\mathcal{L}_a = \frac{1}{N} \sum_N -R \log \pi_S(s_t, a_t)$ with $R = -\frac{1}{M} \sum_M \mathrm{MSE}(V(s_k, a_k), r_k)$

- **Random Sampling :** Random during practice, greedy during evaluation
- **Reactive Policy :** Use same action, change on negative reward, remember ‘bad’
- **Random Agent :** Always random

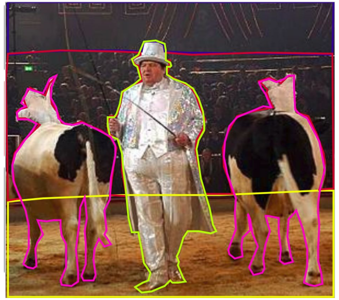
Datasets

- AwA2 [Xian et al., 2018]
  - 85 attributes

polar bear

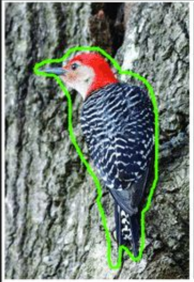
black:	no
white:	yes
brown:	no
stripes:	no
water:	yes
eats fish:	yes

- SUN Attribute [Patterson et al., 2014]
  - 102 attributes for each object



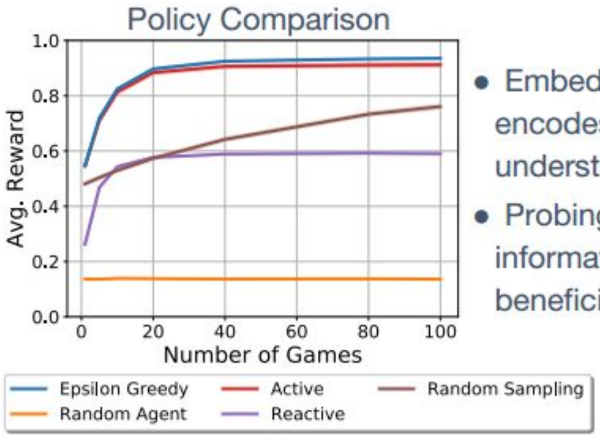
- CUB [Wah et al., 2011]
  - > 200 attributes

forehead_color	black	black	black
breast_pattern	solid	solid	solid
breast_color	white	white	white
head_pattern	plain	capped	plain
back_color	white	white	black
wing_color	grey/white	grey	white
leg_color	orange	orange	orange
size	medium	large	medium
bill_shape	needle	dagger	dagger
wing_shape	pointed	tapered	long
...	...	...	...
primary_color	white	white	white

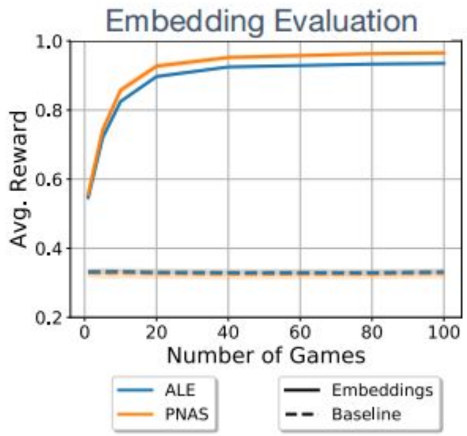


forehead_color	red	red	red
breast_pattern	multi-colored	solid	solid
breast_color	white	white/red	white
head_pattern	capped	capped	capped
back_color	white/black	white/black	white/black
wing_color	white/black	white/black	white/black
leg_color	buff	black	black
size	small	medium	medium
bill_shape	all-purpose	dagger	all-purpose
wing_shape	pointed	tapered	pointed
...	...	...	...
primary_color	black, red	white, black	white, black

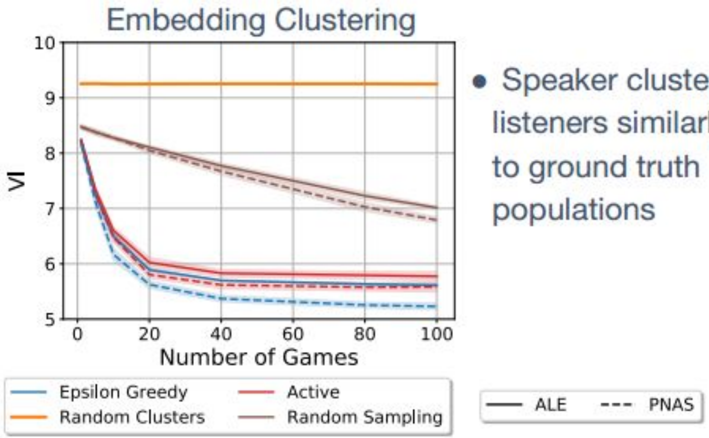
Results



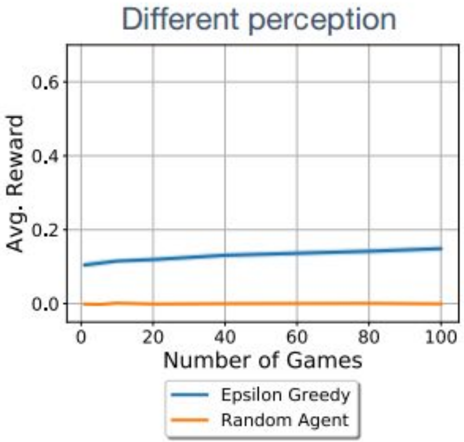
- Embedding encodes concept understanding
- Probing concept information beneficial



- Clear advantage of modeling conceptual understanding

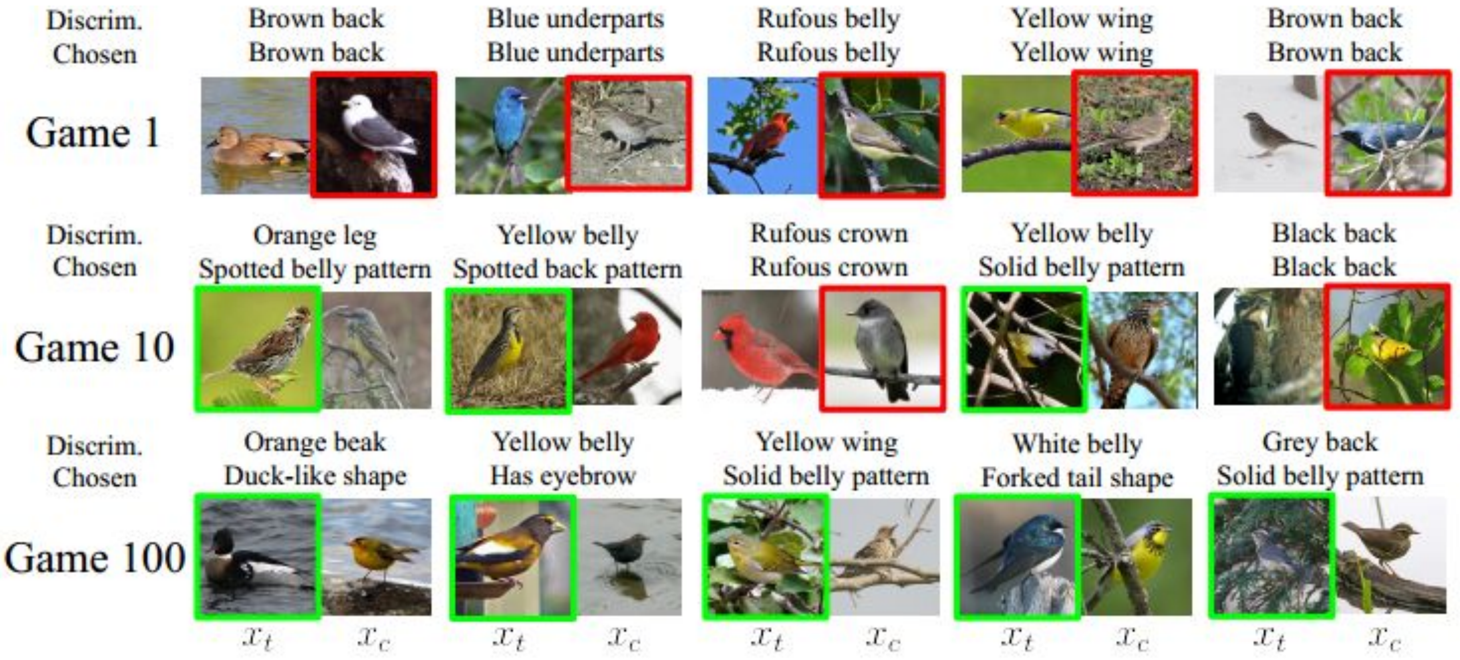


- Speaker clusters listeners similarly to ground truth populations



- Difficult when perception varies

Results -- Continued

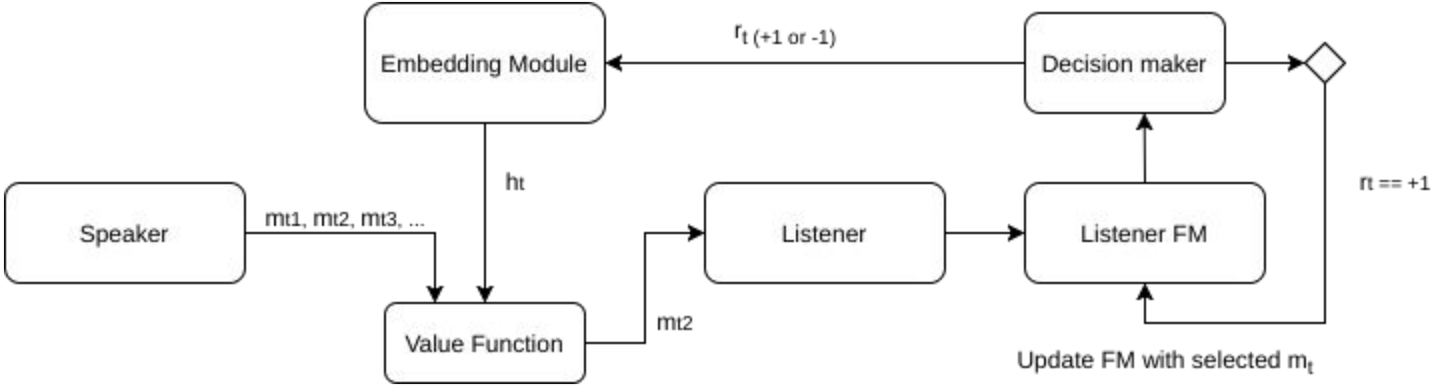


- Color-blind listener
- Epsilon Greedy agent
- Red -> incorrect pick
  Green -> correct pick
- G1: discriminative color attributes chosen
- G100: shape/pattern attributes preferred

Idea

- **Speaker** → A perfect tutor (agent) that already learned to apply several actions on several objects perfectly.
- **Listeners** → Partial knowledge about the actions and states.
- Speaker should generate **motor commands** (including state and action information) which the listener doesn't know how to perform perfectly.
- **Possible actions :** push (left, right, front, etc.), grasp, flip, etc.
- **Possible states:** different objects on different locations, multiple object setups, etc.

Initial Diagram



Initial Setup

- Initialize motor commands as following: 001 → push left, 000 → push right, 101 → grasp left, etc.
- For the sake of simplicity let’s say that an agent learns an action after it collects **N** samples for that action.