

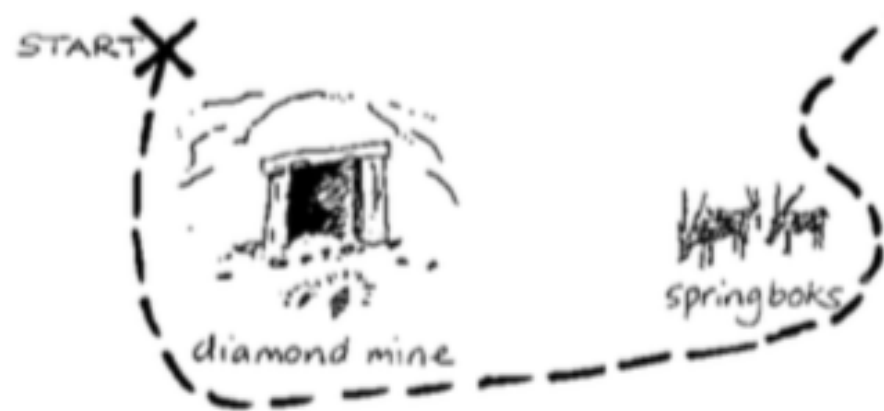
# Learning to Follow Navigational Directions

Adam Vogel and Dan Jurafsky

Presented by Siliang Lu & Rhea Jain

# Goal

- Develop an apprenticeship learning system which learns to imitate human instruction following, without linguistic annotation
- Learn a policy, or mapping from world state to action, which most closely follows the reference route



1. go vertically down until you're underneath eh diamond mine
2. then eh go right until you're
3. you're between springbok and highest view-point

# Dataset

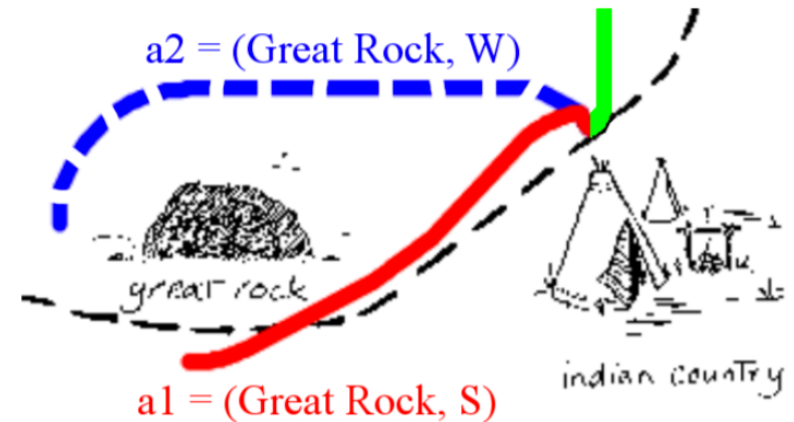
- The Map Task Corpus
  - A set of dialogs between instruction giver and an instruction follower
    - 128 dialogs with 16 different maps
  - Each participant has a map with landmarks
  - The instruction giver:
    - Having a path drawn on the map
    - Must communicate this path to the instruction follower in natural language

# Semantics of spatial language

- Egocentric (speaker-centered frame of reference): “the ball to your left.”
- Allocentric (speaker independent): “the road to the north of the house.”

# Reinforcement Learning

- Goal : Construct Series of moves in the map which most closely map the expert path
- Set  $S$  : States – Intermediate Steps
- Set  $A$ : Actions – Interpretative Steps
- Reward Function  $R$
- Transition Function –  $T(s,a)$
- $D$  – set of Dialogues
- $(l_1, \dots, l_m)$ - Landmarks



# STATE, ACTION & TRANSITION

- State

$$s = (u_i, l, c),$$

- Action

$$a = (l', c')$$

- Transition

$$s' = T(s, a).$$

# Reward

- Reward  $R(s, a)$ : Linear Combination of three features
- Binary Feature indicating if expert would take same path
- Binary Feature indicating the right direction
- Feature which counts number of words similar to the target landmark

## • Policy $\pi(s) = \max_a Q(s, a).$

- Measuring the utility of executing a following policy for the remainder
$$Q^\pi(s, a) = R(s, a) + V^\pi(T(s, a))$$
$$= R(s, a) + Q^\pi(T(s, a), \pi(s))$$

# Features

 $\phi(s, a)$ 

- Mixture of the World Information and linguistic

Information(utterances + landmarks)

above, below, under, underneath, over, bottom, top, up, down, left, right, north, south, east, west, on
---------------------------------------------------------------------------------------------------------------

Components of the Feature Vector

Table 1: The list of given spatial terms.

- 1.Coherence – Similar words between utterance and landmark
- 2.Landmark Locality – check if landmark l is closest
- 3.Direction Locality – Check if cardinal direction closest to the target landmark
- 4.Null Action – Checks if target is null
- 5.Allocentric Spatial – co-joins side c we pass the landmark on with each spatial term
- 6.Egocentric Spatial- co-joins cardinal direction we move in with spatial term

# Approximate Dynamic Programming

- SARSA Algorithm
- Boltzmann Exploration
- Actions with weighted probability

$$\Pr(a_t|s_t; \theta) = \frac{\exp(\frac{1}{\tau}\theta^T \phi(s_t, a_t))}{\sum_{a'} \exp(\frac{1}{\tau}\theta^T \phi(s_t, a'))}$$

- Bellman Equation  $Q(s_t, a_t) = R(s_t, a_t) + \max_{a'} Q(s_{t+1}, a')$
- Minimize temporal difference

$$\theta = \theta + \alpha_t \phi(s_t, a_t) (R(s_t, a_t) + \theta^T \phi(s_{t+1}, a_{t+1}) - \theta^T \phi(s_t, a_t))$$



**Input:** Dialog set  $D$   
Reward function  $R$   
Feature function  $\phi$   
Transition function  $T$   
Learning rate  $\alpha_t$

**Output:** Feature weights  $\theta$

```
1 Initialize  $\theta$  to small random values
2 until  $\theta$  converges do
3   foreach  $Dialog\ d \in D$  do
4     Initialize  $s_0 = (l_1, u_1, \emptyset)$ ,  
      $a_0 \sim \text{Pr}(a_0|s_0; \theta)$ 
5     for  $t = 0; s_t$  non-terminal;  $t++$  do
6       Act:  $s_{t+1} = T(s_t, a_t)$ 
7       Decide:  $a_{t+1} \sim \text{Pr}(a_{t+1}|s_{t+1}; \theta)$ 
8       Update:
9          $\Delta \leftarrow R(s_t, a_t) + \theta^T \phi(s_{t+1}, a_{t+1})$   

10           $\quad - \theta^T \phi(s_t, a_t)$ 
11          $\theta \leftarrow \theta + \alpha_t \phi(s_t, a_t) \Delta$ 
12     end
13   end
14 end
15 return  $\theta$ 
```

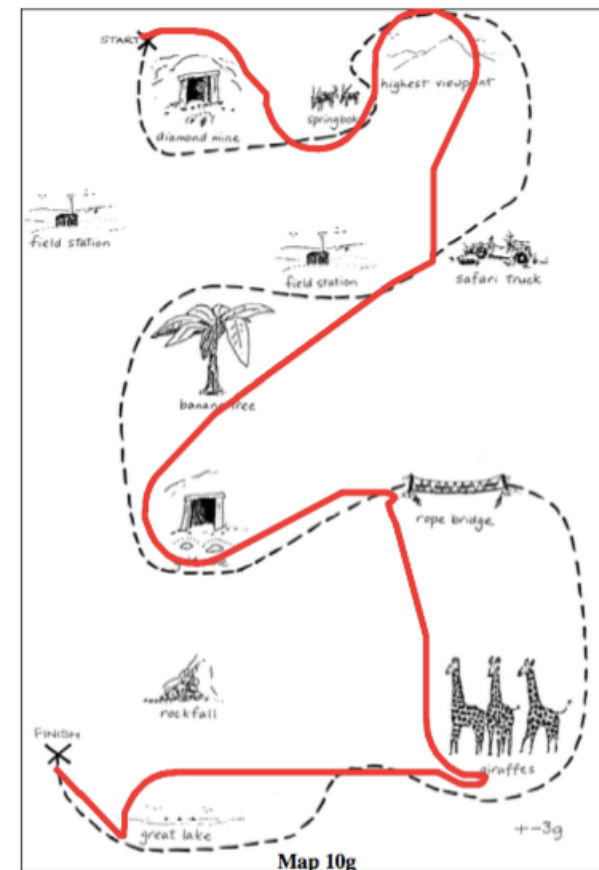
**Algorithm 1:** The SARSA learning algorithm.

# Evaluation

- Visit Order:

- The order in which we visit landmarks
- The minimum distance from  $P_e$  to each landmark
- order precision =  $N / |P|$
- order recall =  $N / |P_e|$

	Visit Order			Side		
	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>
Baseline	28.4	37.2	32.2	46.1	60.3	52.2
PG	31.1	43.9	36.4	49.5	<b>69.9</b>	57.9
SARSA	<b>45.7</b>	<b>51.0</b>	<b>48.2</b>	<b>58.0</b>	64.7	<b>61.2</b>



# Discussion

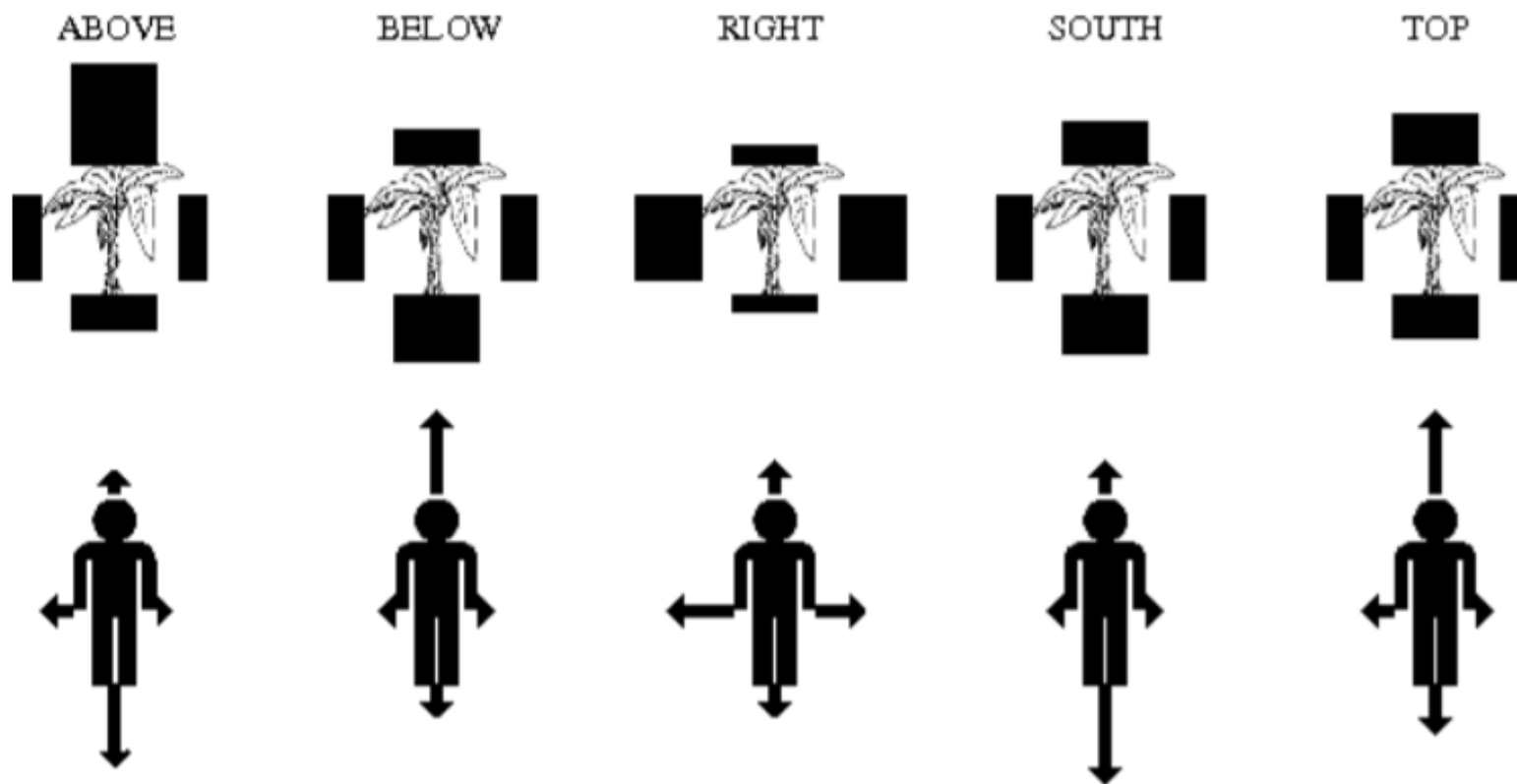


Figure 5: This figure shows the relative weights of spatial features organized by spatial word. The top row shows the weights of allocentric (landmark-centered) features. For example, the top left figure shows that when the word *above* occurs, our policy prefers to go to the north of the target landmark. The bottom row shows the weights of egocentric (absolute) spatial features. The bottom left figure shows that given the word *above*, our policy prefers to move in a southerly cardinal direction.