Vision-Language Navigation
Vision-Language Navigation

Task of navigating inside real environments following natural language instructions

Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.
Aside: Reinforcement Learning

- **Agent** observes the **environment**, identifies its current **state** $s_t \in S$, takes an **action** $a_t \in A$, and gets a **reward** (may be $= 0$).

- **Goal 1**: from each possible state find a sequence of actions that will maximize total reward.

- **Goal 2**: come up with a **policy** $\pi: S \rightarrow A$.

- **Markov Decision Process (MDP)**
Aside: Reinforcement Learning

- States $S = \{s_0 \ldots s_8\}$
- Actions $A(S_t) = \{U, D, R, L\}$
- Transition F-n $T: S, A \rightarrow S'$
- Reward F-n $R(S, A)$
  - $r_t = r(s_t, a_t)$
- Policy $\pi: S \rightarrow A$
- Discounted Cumulative Reward:
  $$V^\pi(s_t) = r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \ldots$$

- Say $\gamma = .9$
  $$V^\pi(s_0) = 0 + 0 \cdot .9 + 0 \cdot .9^2 + 100 \cdot .9^3 = 73$$
- $\pi(S = s_0) = R, \pi(S = s_1) = R,$
  $\pi(S = s_2) = D, \pi(S = s_5) = D$
Challenges

- Reasoning over visual images & natural language instructions is difficult
  - Ground an instruction in the local scene
  - Match the instruction to the visual trajectory over time
- Ill-posed feedback
- Generalization problem
Summary

- Novel Reinforced Cross-Modal Matching approach
  - Reasoning Navigator (based on where I am right now / what I’m seeing):
    - What should I be doing?
    - Which sub-instruction should execute?
  - Matching critic:
    - What is the probability of reconstructing the instruction based on executed trajectory
    - Penalize the trajectories that don’t match instructions
Matching Critic

Enforces the agent to follow the instruction by rewarding trajectories that match the instruction
Summary

- Use Reinforcement Learning with two reward functions
- Use Self-Supervised Imitation learning to deal with unseen environments
- Establishes new state-of-the-art performance on R2R dataset
- Proposes a new evaluation setting for VLN:
  - Exploring unseen environments prior to testing is allowed
Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front of the toilet.

RCM Model

Close to the target?
Follows the instruction?
Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front of the toilet.
turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dining room, … …
turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …

Navigator

split by different viewpoints

CNN v1

CNN v2

CNN vm
Navigator

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, ... ...

Language Encoder

\{w_i\}_{i=1}^n

Attention

\{v_{t,j}\}_{j=1}^m

Panoramic Features

Attention

\{h_t\}

Trajectory Encoder

Action Predictor

\text{c}^{\text{text}}_t

\text{c}^{\text{visual}}_t

LSTM

[w1 \ w2 \ w3 \ ... \ wn]
LSTM

- Good for representation learning of sequential data
Navigator

"Turn" "completely" "around" "until" … …

LSTM

[w1 w2 w3 … wn]
Cross modal reasoning

Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …
Cross modal reasoning

- Learns history based on previous action and visual input
- Learns how to focus on textual instruction based on history
- Learns how to focus on visual input based on textual instruction

---

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, ...
Cross modal reasoning

Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …
Cross modal reasoning

Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction
Cross modal reasoning

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …

Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction
Cross modal reasoning

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dining room, … …

Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction
Attention (generic) Depending on the state, which input should be focused on (weighted more)?

\[
\hat{x} = \text{attention}(s_t, \{x_i\}_{i=1}^{n}) = \\
\sum_{i=1}^{n} \text{softmax}(s_t W_s (x_i W_x)^T) x_i
\]
Attention (generic) Depending on the state, which input should be focused on (weighted more)?

\[ \hat{x} = \text{attention}(s_t, \{x_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \text{softmax}(s_t W_s (x_i W_x)^T) x_i \]

State at time \( t \) Set of inputs Weight of \( x_i \)
Attention (generic)

Depending on the state, which input should be focused on (weighted more)?

\[ \hat{x} = \text{attention}(s_t, \{x_i\}_{i=1}^n) = \sum_{i=1}^{n} \text{softmax}(s_t W_s (x_i W_x)^T) x_i \]

State at time \( t \)  
Set of inputs  
Weight of \( x_i \)  
How state contributes
**Attention** (generic)

Depending on the state, which input should be focused on (weighted more)?

\[
\hat{x} = \text{attention}(s_t, \{x_i\}_{i=1}^n) = \\
\sum_{i=1}^n \text{softmax}(s_t W_s (x_i W_x)^T) x_i
\]

**State at time t**

**Set of inputs**

**Weight of xi**

**How input contributes**
History Context

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …

\[ v_t = \text{attention}(h_{t-1}, \{v_{t,j}\}_{j=1}^m) = \sum_j \text{softmax}(h_{t-1}W_h(v_{t,j}W_v)^T)v_{t,j} \]
turn completely around until you face an open door with a window to the left and a patio to the right, walk forward through the door and into a dining room, … …

$$v_t = \text{attention}(h_{t-1}, \{v_{t,j}\}_{j=1}^m)$$
$$= \sum_j \text{softmax}(h_{t-1}W_h(v_{t,j}W_v)^T)v_{t,j}$$
Visually conditioned textual context

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dining room, … ….

\[ c_t^{\text{text}} = \text{attention}(h_t, \{w_i\}_{i=1}^{n}) \]
Textually conditioned visual context

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, ... ...

\[ c_t^{visual} = \text{attention}(c_t^{text}, \{v_j\}_{j=1}^m) \]
turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …

$$p_k = \text{softmax}([h_t, c_t^{\text{ext}}, c_t^{\text{visual}}]W_c(u_kW_u)^T)$$
Action prediction

Each viewpoint corresponds to a next-step reachable location

\[ p_k = \text{softmax}( [ h_t, c_t^{text}, c_t^{visual} ] W_c (u_k W_u)^T ) \]
Action prediction

\[ p_k = \text{softmax}( [h_t, c_t^{text}, c_t^{visual}] W_c (u_k W_u)^T ) \]

- image patch from one viewpoint
- heading
- elevation
- action vector
- Context vector
Action prediction

\[ p_k = \text{softmax}( [ h_t, c_t^{text}, c_t^{visual} ] W_c(u_k W_u)^T ) \]

- image patch from one viewpoint
- heading
- elevation
- action vector

- Context vector
- Trainable weights
Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front of the toilet.
Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front of the toilet.
Matching Critic (intrinsic reward)

Enforces the agent to follow the instruction by rewarding trajectories that match the instruction and to determine matching...
Matching Critic (intrinsic reward)

Try to reconstruct the instruction from the trajectory and see how it matches the original one.
Sequence to sequence

Encoder

Decoder
Matching Critic (intrinsic reward)

- Try to reconstruct the instruction from the trajectory and see how it matches the original one.

\[ V_\beta(\chi, \tau) = p_\beta(\chi|\tau) \]
Learning

● Goal: learn the policy $\pi: S \rightarrow A$

● Approximate the policy first!
  ○ Behavior cloning
  ○ Supervised Learning
  ○ Use state-action data from seen environments
  ○ $a^*_t$ - demonstration action

\[ L_{sl} = -\mathbb{E}[\log(\pi_\theta(a^*_t | s_t))] \]
Learning

- **Problem:** limited generalizability
  - Unpredictable behavior in unseen states

- **Solution:** switch over to Reinforcement Learning!
Learning - Extrinsic Reward

- Two metrics:
  - Immediate reward:
    - **Distance** from the current state \( s_t \) to the target state \( s_{\text{target}} \):
      \[
      D_{\text{target}}(s_t) = D_{\text{target}}(s_t) - D_{\text{target}}(s_{t+1})
      \]
  - Delayed reward:
    - When the agent reaches its destination:
      \[
      r(s_T, a_T) = \mathbb{1}(D_{\text{target}}(s_T) \leq d)
      \]
Learning - Extrinsic Reward

\[ R_{extr}(s_t, a_t) = r(s_t, a_t) + \sum_{t'=t+1}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'}) \]

- immediate reward
- discounted future reward
Learning - Extrinsic & Intrinsic Rewards

- Matching critic calculates intrinsic reward $R_{intr}$
- Based on alignment between the trajectory and the original instruction
- Final Reward function:
  
  \[ L_{rl} = -\mathbb{E}_{a_t \sim \pi_\theta} [A_t] \text{ where } A_t = R_{extr} + \delta R_{intr} \]

  - $\delta$ is a hyperparameter defining importance of $R_{intr}$
  - Gradient of the loss function:
    \[ \nabla_\theta L_{rl} = -A_t \nabla_\theta \log \pi_\theta(a_t|s_t) \]
Self-Supervised Imitation Learning

- **Original setting**: train on *seen* environments and test on *unseen* ones
- **New setting**: allow exploration on *unseen* environments as well
  - Lifelong learning & adaptation
  - No ground-truth demonstrations
  - Don’t know where the target location is => no extrinsic reward
Self-Supervised Imitation Learning

1. Get an instruction $\mathcal{X}$

2. Navigator produces possible trajectories

3. Matching Critic $V_\beta$ finds the best trajectory: $\hat{\tau} = \arg \max_\tau V_\beta(\mathcal{X}, \tau)$

4. Navigator stores it into a replay buffer

5. The agent uses the saved trajectory $\hat{\tau}$ as a ground-truth

6. Self-supervised learning loss: $L_{sil} = -\mathbb{E}[\log(\pi_\theta(\hat{a}_t | s_t))]$
Experimental setup:

• **R2R dataset**: This dataset covers most of the visual diversity and instructions. It consists of training, seen validation, unseen validation and test sets.

• **Testing scenarios**:
  - **Standard**: Train in seen environments and test in unseen environments without prior exploration.
  - **Lifelong learning**: The agent is encouraged to explore the environment and learn from the feedback.
Evaluation metrics:

- **PL**: Path length
- **NE**: Navigation error
- **OSR**: Oracle success rate
- **SR**: Success rate
- **SPL**: Success rate weighted by inverse Path Length
<table>
<thead>
<tr>
<th>Model</th>
<th>PL ↓</th>
<th>NE ↓</th>
<th>OSR ↑</th>
<th>SR ↑</th>
<th>SPL ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>9.89</td>
<td>9.79</td>
<td>18.3</td>
<td>13.2</td>
<td>12</td>
</tr>
<tr>
<td>seq2seq [2]</td>
<td>8.13</td>
<td>7.85</td>
<td>26.6</td>
<td>20.4</td>
<td>18</td>
</tr>
<tr>
<td>RPA [48]</td>
<td>9.15</td>
<td>7.53</td>
<td>32.5</td>
<td>25.3</td>
<td>23</td>
</tr>
<tr>
<td>Speaker-Follower [11]</td>
<td>14.82</td>
<td>6.62</td>
<td>44.0</td>
<td>35.0</td>
<td>28</td>
</tr>
<tr>
<td>+ beam search</td>
<td>1257.38</td>
<td>4.87</td>
<td>96.0</td>
<td>53.5</td>
<td>1</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCM</td>
<td>15.22</td>
<td>6.01</td>
<td>50.8</td>
<td>43.1</td>
<td>35</td>
</tr>
<tr>
<td>RCM + SIL (train)</td>
<td>11.97</td>
<td>6.12</td>
<td>49.5</td>
<td>43.0</td>
<td>38</td>
</tr>
<tr>
<td>RCM + SIL (unseen)³</td>
<td>9.48</td>
<td>4.21</td>
<td>66.8</td>
<td>60.5</td>
<td>59</td>
</tr>
</tbody>
</table>

Table 1: Comparison between RCM, SIL and the state-of-the-art
<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>Seen Validation</th>
<th>Unseen Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PL</td>
<td>NE</td>
</tr>
<tr>
<td>0</td>
<td>Speaker-Follower (no beam search) [11]</td>
<td>-</td>
<td>3.36</td>
</tr>
<tr>
<td>1</td>
<td>RCM + SIL (train)</td>
<td>10.65</td>
<td>3.53</td>
</tr>
<tr>
<td>2</td>
<td>RCM</td>
<td>11.92</td>
<td>3.37</td>
</tr>
<tr>
<td>3</td>
<td>– intrinsic reward</td>
<td>12.08</td>
<td>3.25</td>
</tr>
<tr>
<td>4</td>
<td>– extrinsic reward = pure SL</td>
<td>11.99</td>
<td>3.22</td>
</tr>
<tr>
<td>5</td>
<td>– cross-modal reasoning</td>
<td>11.88</td>
<td><strong>3.18</strong></td>
</tr>
<tr>
<td>6</td>
<td>RCM + SIL (unseen)</td>
<td><strong>10.13</strong></td>
<td><strong>2.78</strong></td>
</tr>
</tbody>
</table>

Table 2: Ablation study on seen and unseen environments
**Instruction:** Exit the door and turn left towards the staircase. Walk all the way up the stairs, and stop at the top of the stairs.

Intrinsic Reward: 0.53  
Result: Success (error = 0m)

---

**Instruction:** Turn right and go down the stairs. Turn left and go straight until you get to the laundry room. Wait there.

Intrinsic Reward: 0.54  
Result: Failure (error = 5.5m)

---

Above steps are all good, but it stops at a wrong place in the end.

---

(a) A successful case  
(b) A failure case

Figure 6: Qualitative examples from the unseen validation set.
Strength and weakness:

Strengths:

- Reinforcement learning leads to better generalizability
- Self-Imitation learning facilitates exploring on unseen environments
- Cross modal grounding effectively enhances the model’s ability to capture context information
Strength and weakness:

Weakness:

- Limited in dataset diversity (Only on R2R)
- Model implementation heavily depends on simulator specifics
- Not using depth data
Conclusion:

This paper presents two novel approaches, RCM and SIL, which combine reinforcement learning and self-supervised imitation learning.

The experiment results show that these methods are effective and efficient in both the standard testing scenario and lifelong learning scenario.

SIL help achieve strong generalizability in unseen environments.
Questions