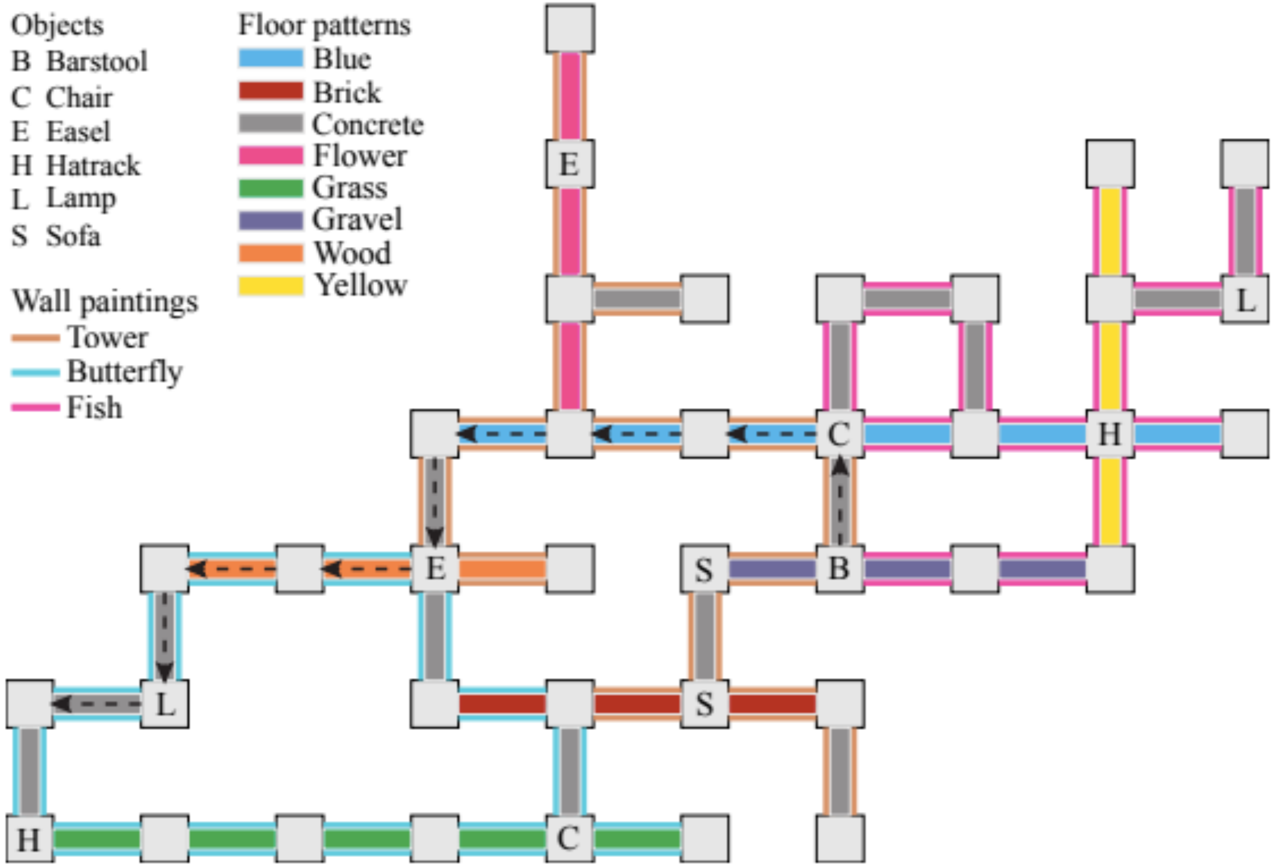
Listen, Attend, and Walk: Neural Mapping of Navigational Instructions to Action Sequences

Hongyuan Mei, Mohit Bansal, Matthew R. Walter

Toyota Technological Institute, Chicago

Introduction

- Neural sequence-to-sequence model for direction following



Place your back against the wall of the “T” intersection. Go forward one segment to the intersection with the blue-tiled hall. This interesction [sic] contains a chair. Turn left. Go forward to the end of the hall. Turn left. Go forward one segment to the intersection with the wooden-floored hall. This intersection conatains [sic] an easel. Turn right. Go forward two segments to the end of the hall. Turn left. Go forward one segment to the intersection containing the lamp. Turn right. Go forward one segment to the empty corner.

Introduction

- Learn correspondences between instruction and actions using an alignment-based LSTM
- End-to-end differentiable sequence-to-sequence model

Model architecture

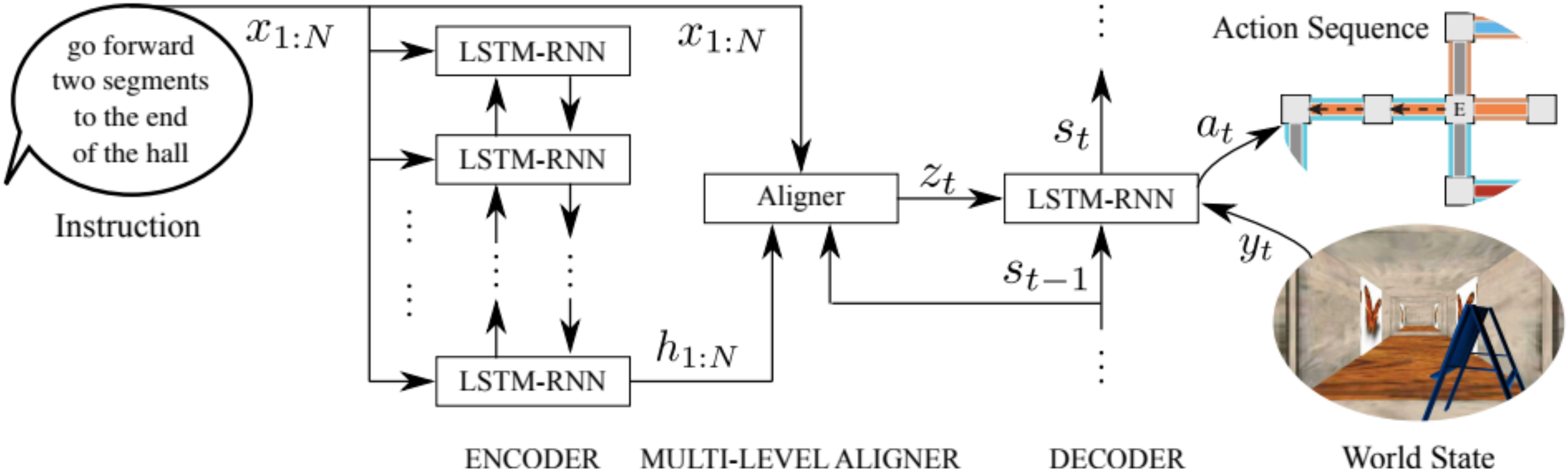


Figure 2: Our encoder-aligner-decoder model with multi-level alignment

Model architecture

- Inference over a probabilistic model

$$
\begin{aligned}
a^*_{1:T} &= \underset{a_{1:T}}{\arg\max}\, P(a_{1:T}|y_{1:T}, x_{1:N}) \\
&= \underset{a_{1:T}}{\arg\max} \prod_{t=1}^{T} P(a_t|a_{1:t-1}, y_t, x_{1:N})
\end{aligned}
$$

- Neural encoder decoder model with attention

Model architecture

- Bidirectional LSTM to encode instruction

$$\begin{pmatrix} i_j^{\mathrm{e}} \\ f_j^{\mathrm{e}} \\ o_j^{\mathrm{e}} \\ g_j^{\mathrm{e}} \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T^{\mathrm{e}} \begin{pmatrix} x_j \\ h_{j-1} \end{pmatrix}$$

$$c_j^e = f_j^e \odot c_{j-1}^e + i_j^e \odot g_j^e$$

$$h_j = o_j^e \odot \tanh(c_j^e)$$

$$h_j = (\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top)^\top$$

Model architecture

- Multi level aligner: High level (hidden states of LSTM) + low level (input words)
- One layer neural perceptron

$$z_t = \sum_j \alpha_{tj} \begin{pmatrix} x_j \\ h_j \end{pmatrix}$$

$$\alpha_{tj} = \exp(\beta_{tj}) / \sum_j \exp(\beta_{tj}),$$

$$\beta_{tj} = v^\top \tanh(W s_{t-1} + U x_j + V h_j)$$

- Intuitively, better match the salient words in input sentence (e.g., “easel”) directly to corresponding landmarks in the current world state y(t) used in decoder

Model architecture

- LSTM decoder

$$\begin{pmatrix} i_t^d \\ f_t^d \\ o_t^d \\ g_t^d \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T^d \begin{pmatrix} Ey_t \\ s_{t-1} \\ z_t \end{pmatrix}$$

$$c_t^d = f_t^d \odot c_{t-1}^d + i_t^d \odot g_t^d$$

$$s_t = o_t^d \odot \tanh(c_t^d)$$

$$q_t = L_0(Ey_t + L_s s_t + L_z z_t)$$

$$P_{a,t} = \mathrm{softmax}\left(q_t\right)$$

- Output P is the conditional probability distribution over actions
- E is an embedding matrix

- Trained using negative log likelihood of demonstrated action

Experiments

- SAIL route instructor dataset
- World state ($y(t)$) encodes local observable world at time t, encoded as a concatenation of a bag-of-words vector for each direction (forward, left, and right).

Results

Method	Single-sent	Multi-sent
Chen and Mooney (2011)	54.40	16.18
Chen (2012)	57.28	19.18
Kim and Mooney (2012)	57.22	20.17
Kim and Mooney (2013)	62.81	26.57
Artzi and Zettlemoyer (2013)	65.28	31.93
Artzi, Das, and Petrov (2014)	64.36	**35.44**
Andreas and Klein (2015)	59.60	–
Our model (vDev)	69.98	26.07
Our model (vTest)	**71.05**	30.34

Ablation results

Table 2: Model components ablations

	Full Model	High-level Aligner	No Aligner	Unidirectional	No Encoder
Single-sentence	69.98	68.09	68.05	67.44	61.63
Multi-sentence	26.07	24.79	25.04	24.50	16.67

Visualization

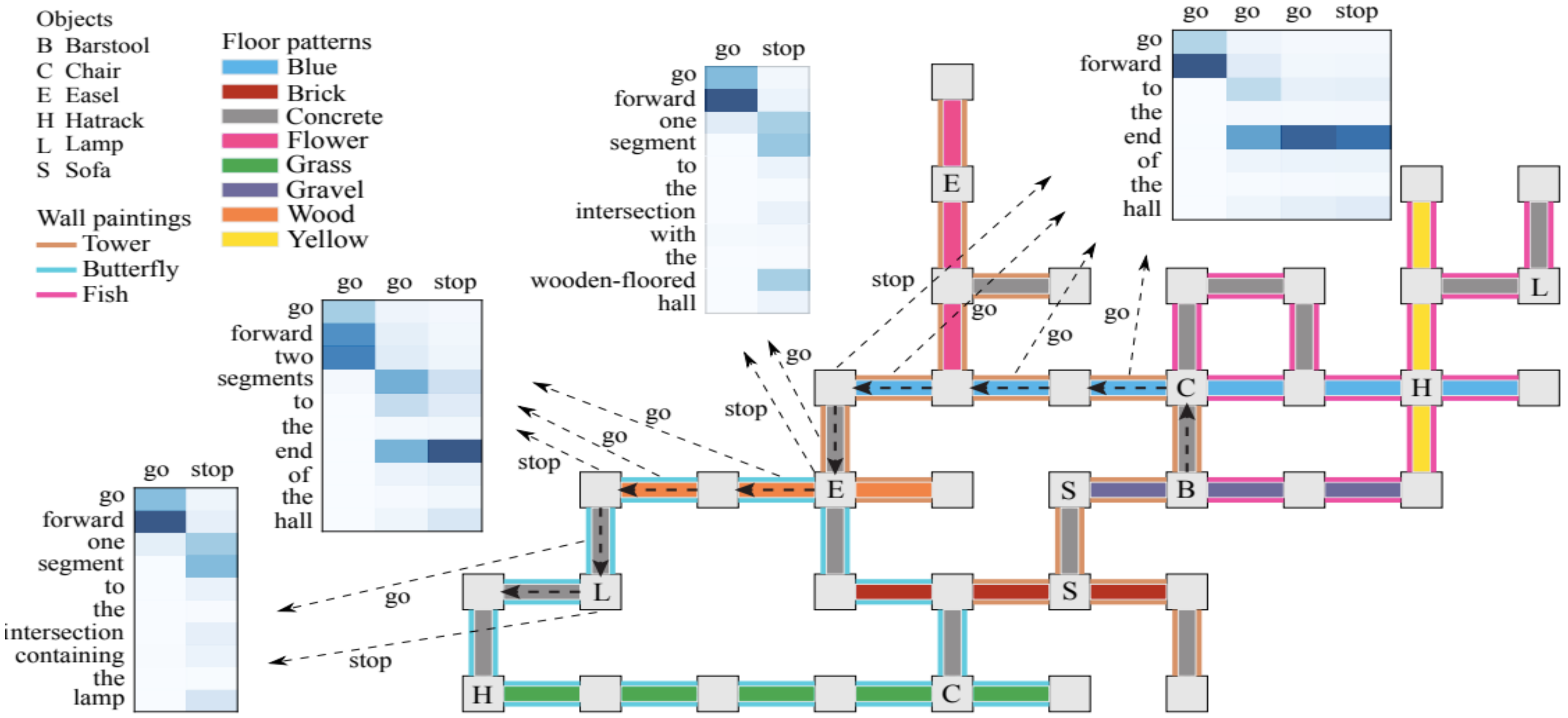


Figure 4: Visualization of the alignment between words to actions in a map for a multi-sentence instruction.