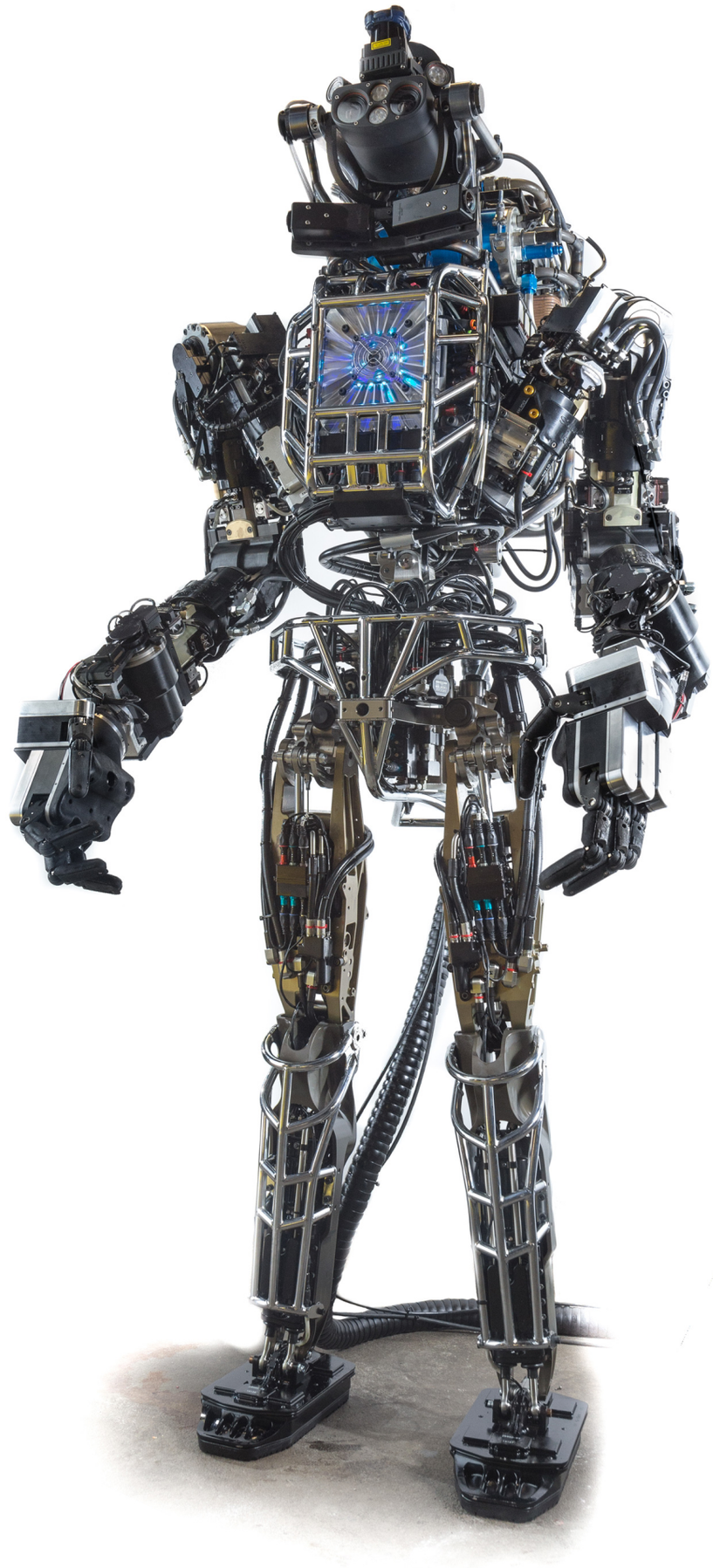
Listen, Attend, and Walk: Neural Mapping of Navigational Instructions to Action Sequences

Motivation

- Command robots using natural language instructions
- Free-form instructions are difficult for robots to interpret due to its ambiguity and complexity
- Previous methods rely on language semantics to parse natural language instructions
- Can robot learn the mapping from instructions to actions directly?

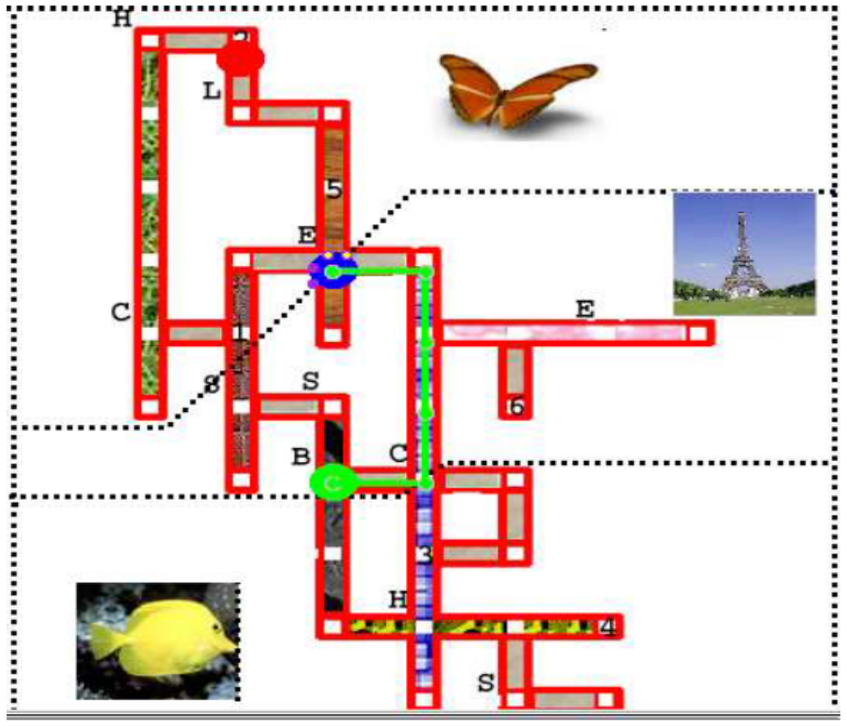


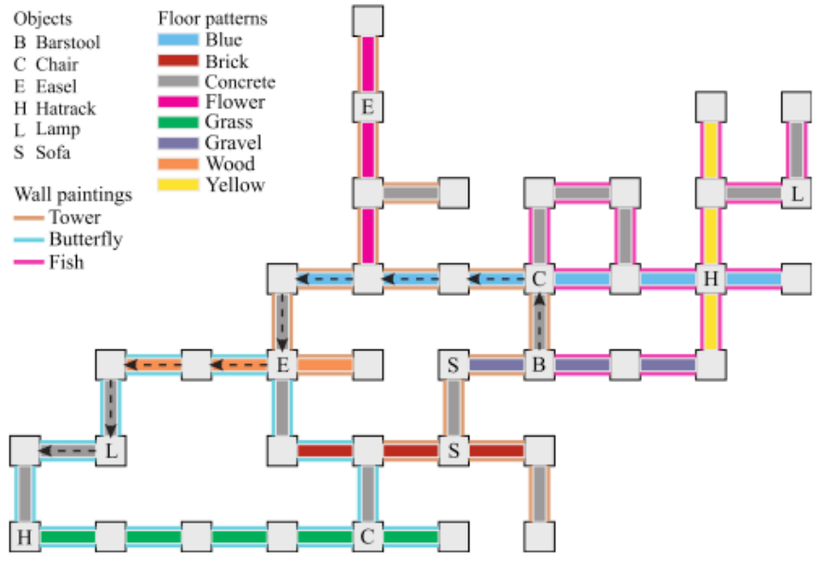
Previous Work

- Symbol grounding problem (Harnad 1990): What is the meaning of words (symbols)?
    - How do the words in our head connects to things they refer to in the real world?
- Manual mapping of words to environment features and actions (MacMahon 2006)
    - Corpus of 786 route instructions from 6 people in 3 large indoor environments
    - Instructions were validated by 36 people with 69% completion rate
    - MACRO:
        - Interpret instructions linguistically to obtain meaning
        - Combine linguistic meaning with spatial knowledge to compose action sequence
        - Infer actions via exploratory actions
        - 61% completion rate

Previous Work

- MACRO: simulated environment for indoor navigation
  - Hallways with pattern on the floor
  - Paintings on the wall
  - Objects at intersections
- This setup and dataset is used in this paper





Previous Work

- Translate instructions into formal language equivalent
  - Learning a parser to handle the mapping
  - Use probabilistic context free grammar to parse free-form instructions into formal actions (Kim and Mooney 2013)
- Mapping instructions to features in the world model
  - Use generative model of the world and learn a model for spatial relations, adverbs and verbs (Kollar 2010)
  - Parse the free-form instructions and and use probability distribution to express the learned relation between words and actions

Problem Statement

- Sequence to sequence learning problem
- Translating navigational instructions to sequence of actions
  - Knowledge of the local environment in the agent's line-of-sight
  - Understand the natural language commands and map words in the instructions to correct actions
    - Instructions may not be completely specified

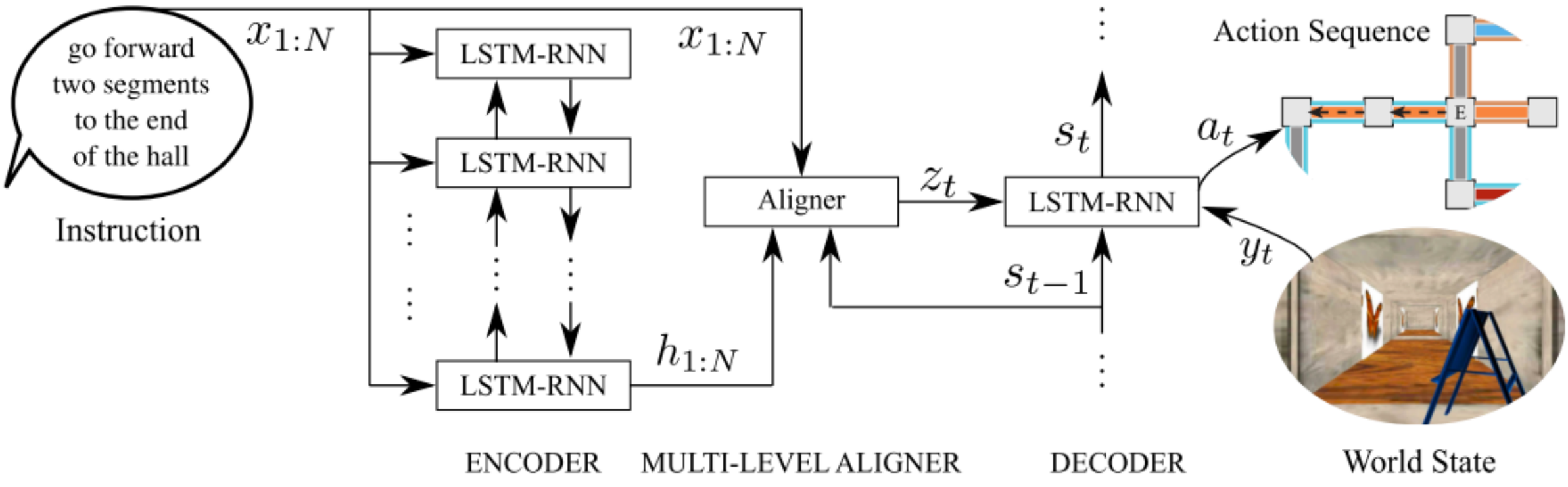
Problem Statement

- Variables
  - $\mathbf{x}^{(i)}$, variable length natural language instructions
  - $\mathbf{y}^{(i)}$, observable environment (world state)
  - $\mathbf{a}^{(i)}$, action sequence
- Mapping instructions to action sequence
  - $\mathbf{a}_{1:T} = \underset{\mathbf{a}_{1:T}}{\arg\max}\, P(\mathbf{a}_{1:T} \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:N})$

Implementation: Encoder

- Encoder-decoder architecture for sequence to sequence mapping
- Encoder: Bidirectional Recurrent Neural Net (BiRNN)
- $\mathbf{h_j = f(x_j, h_{j-1}, h_{j+1})}$, the encoder's hidden state for word j
- Hidden states h are obtained via feeding instructions x to Long Short-Term Memory(LSTM)-RNN
  - h describes the temporal relationships between previous words

Implementation: Overview



Implementation: Encoder

- Why LSTM-RNN?
  - RNN handles variable length input: input sequence of symbols are compressed into the context vector (h)
  - RNN models the sequence probabilistically
  - LSTM is shown to provide better recurrent activation function for RNN: LSTM unit “remembers” previous information better

Implementation: Multi-Level Aligner

- x_j and h_j describes the instruction and the context
- aligner decides which part of input will have higher influence (attention weight) and help the decoder to focus depending on the context
- This paper included x_j in the aligner to improve performance
  - both high-level (h) and low-level (x) representations are considered by the aligner
  - The model can offset information lost in abstraction of the instruction
- $\mathbf{z_t = c(h_1, \ldots, h_N)}$, the context vector to encode instructions at time t - this is for the decoder

Implementation: Decoder

- LSTM-RNN
  - decoder takes world state (y_t) and context of instruction (z_t) as input
  - The output is the conditional probability for the next action

$$P_{a,t} = P(a_t | a_{1:t-1}, y_t, x_{1:N})$$

Implementation: Training

- Objective

- $$a^*_{1:T} = \arg\max_{a_{1:T}} P(a_{1:T}|y_{1:T}, x_{1:N})$$

$$= \arg\max_{a_{1:T}} \prod_{t=1}^{T} P(a_t|a_{1:t-1}, y_t, x_{1:N})$$

- Loss function

- $$L = -\log P(a^*_t|y_t, x_{1:N})$$

- Parameters are learned through back-propagation

Experiment: Setup

- SAIL route instruction dataset (MacMahon, 2006)
- Local environment: features and objects in line-of-slight
- Single-sentence and multi-sentence task
- Training
  - 3 maps for 3-fold cross validation
  - for each map, 90% training and 10% validation

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

- Outperforms state-of-the-art in single sentence task
- Competitive result for multi-sentence task

Results: Ablation Studies and Distance Evaluation

	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

- The encoder-decoder architecture using RNN with multi-level aligner can significantly improve performance

Distance (d)	0	1	2	3
Single-sentence	71.73	86.62	92.86	95.74
Multi-sentence	26.07	42.88	59.54	72.08

- In the failure cases, the model can produce end-points that are close to the destination

Conclusion

- LSTM-RNN with multi-level aligner achieves a new state-of-the-art performance on single sentence navigation task
- This model does not require linguistic knowledge and can be trained end-to-end
- Low-level context (the original input) is shown to improve performance

Discussion

- This problem is very similar to the machine translation problem, with additional environment information for the model to make the decision

- The authors' approach is largely inspired by advances in neural machine translation and encoder-decoder architecture

- The model does not implement exploratory behaviour nor correcting mistakes

- It would be interesting to investigate the effect of error in the instructions in leading to the failed navigation

- Multilevel alignment and the use of BiRNN greatly increase model complexity