Lecture 27 Part 3: Grounded Dialogue
Collaborative Construction and Communication in Minecraft

Julia Hockenmaier
University of Illinois
CS447
Blocks World: Winograd’s SHRDLU (1971)

SHRDLU had a symbolic representation of a scene with several different types of blocks (to simulate an *immobile robot with an arm*).

Users could *instruct SHRDLU to move blocks* in this scene (and ask questions about the scene).

But SHRDLU was based entirely on *handwritten symbolic rules and domain knowledge*.

Can modern systems *learn to perform this task without handwritten rules*?
Minecraft as a virtual platform for NLP

Popular multi-player gaming platform where avatars navigate in a 3D world and manipulate block-like materials.

Microsoft’s Project Malmo API makes it possible to use Minecraft for reinforcement learning and other AI research.

We show that this makes Minecraft a great virtual platform to study interactive, situated language generation & understanding.

We can use Minecraft to simulate a Blocks World for embodied agents.
THE MINECRAFT COLLABORATIVE BUILDING TASK

(Narayan-Chen, Jayannavar, Hockenmaier, ACL 2019)
The Architect knows the Target 
observes the Builder

The Builder has to build a copy of the Target

Chat Interface

A: In about the middle build a column five tall
A: then two more to the left of the top to make a 7
A: now a yellow 6
A: the long edge of the 6 aligns with the stem of the 7 and faces right
B: where does the 6 start?
A: behind the 7 from your perspective
How do People Perform This Task?
<Architect> go the middle and place an orange block two spaces to the left.
<Architect> go the middle and place an orange block two spaces to the left.
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a **staircase** with 2 stairs left and 2 right with orange

**Names of Substructures!**
<Architect> go the middle and place an orange block two spaces to the left.
<Architect> now make a staircase with 2 stairs left and 2 right with orange.
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v

Variable-length action sequences


Our Dataset: The Minecraft Dialogue Corpus
Minecraft Dialogue Corpus
(Narayan-Chen, Jayannavar & Hockenmaier, 2019)

- 150 Target structures, split across train/test/dev
- 509 Human-human dialogues & game logs for the Collaborative Building Task
- 15.9k Utterances (11.5k Architect, 4.4k Builder)
- 6.6k Builder action sequences

- Built on top of Microsoft’s Project Malmo
- You can download our data and data collection code
- Caveat: data collection requires users to have our version of Minecraft/Malmo on their machines
How Can We Build Systems that Can Perform This Task?
How can we build agents that can perform this task?

Option 1:
Develop rich linguistic representations for this domain
Annotate the Minecraft Dialogue Corpus
Train generation and parsing models on these annotations
Develop agents that use these models

Option 2:
Train end-to-end neural models on this data
Starting Point for Architect: Utterance Generation

(Narayan-Chen, Jayannavar, Hockenmaier, ACL 2019)
Architect: Tasks and Challenges

Give clear and correct instructions in a changing environment

A. needs to identify next steps for B.
A. needs to align target and build region
A. needs to adapt to B’s current position
A. needs to identify mistakes made by B.

Answer Builder’s questions

Interrupt the Builder to correct mistakes
A. should respond in real time (no turns)
Generate a suitable Architect utterance for a game state in a human-human game when the human Architect said something.

Ignores real-time aspect (when to speak)

Ignores overall task completion (how to maintain a whole conversation)

Allows us to use supervised learning to develop baseline models
Modeling the World State: Align Target and Built structures

**Design decisions:**
No landmarks in the environment. Any placement of the Target structure within the Built region is correct.
Modeling the World State \textit{naively} with Block Counters

**Global Block counters** (one 18-dimensional vector)
For each of the 6 colors: \#blocks to be \textbf{added}, \textbf{added next}, and \textbf{removed}
Averaged over all optimal alignments of built to target.

<table>
<thead>
<tr>
<th>Add</th>
<th>Add</th>
<th>Remove</th>
<th>Add</th>
<th>Add</th>
<th>Remove</th>
<th>Add</th>
<th>Add</th>
<th>Remove</th>
<th>Add</th>
<th>Add</th>
<th>Remove</th>
<th>Add</th>
<th>Add</th>
<th>Remove</th>
<th>Add</th>
<th>Add</th>
<th>Remove</th>
</tr>
</thead>
</table>

**Local Block Counters** (concatenate 27 block counters)
Separate counters for each cell in the $33 \times 3$ \textbf{cube around the last cell} the Builder touched.
To capture the Builders’ current perspective, the order of cells depends on the Builder’s current position, pitch and yaw.
Our Model

**Discourse Context encoder:**
biGRU over previous dialogue with Glove embeddings

**World Context Encoder:**
$W_{global}$: Global Block counters
$W_{local}$: Local Block counters

**Output utterance decoder:**
Reads block counter embeddings (and last token) at each time step
## Automatic Evaluation

<table>
<thead>
<tr>
<th>Automatic Evaluation</th>
<th>BLEU-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq2seq</td>
<td>15.3</td>
</tr>
<tr>
<td>Block Counter</td>
<td>15.7</td>
</tr>
</tbody>
</table>

Block Counter model gives a *minor improvement* in BLEU-1.
## Automatic Evaluation

<table>
<thead>
<tr>
<th>Automatic Evaluation</th>
<th>BLEU-1</th>
<th>Spatial P/R</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq2seq</td>
<td>15.3</td>
<td>9.3/8.6</td>
</tr>
<tr>
<td>Block Counter</td>
<td>15.7</td>
<td>8.7/8.7</td>
</tr>
</tbody>
</table>

Block Counter model gives a **minor improvement in BLEU-1.** Block Counter model has **slightly lower performance on spatial terms.**
## Automatic Evaluation

<table>
<thead>
<tr>
<th>Automatic Evaluation</th>
<th>BLEU-1</th>
<th>Spatial P/R</th>
<th>Color P/R</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq2seq</td>
<td>15.3</td>
<td>9.3/8.6</td>
<td>8.1/17.0</td>
</tr>
<tr>
<td>Block Counter</td>
<td>15.7</td>
<td>8.7/8.7</td>
<td>14.9/28.7</td>
</tr>
</tbody>
</table>

Block Counter model gives a **minor improvement in BLEU-1**. Block Counter model has **slightly lower performance on spatial terms**. Block Counter model has **much better precision and recall of color terms**.
Human Evaluation

How correct are the generated utterances (wrt. current game state and target)? Correct utterances are more likely to lead to task completion.

<table>
<thead>
<tr>
<th></th>
<th>Fully correct</th>
<th>Partially correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human (ceiling)</td>
<td>89.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Most human utterances are fully correct (remainder: correctness can’t be assessed, e.g. in chit-chat)
Human Evaluation

How correct are the generated utterances (wrt. current game state and target)? Correct utterances are more likely to lead to task completion.

<table>
<thead>
<tr>
<th></th>
<th>Fully correct</th>
<th>Partially correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human (ceiling)</td>
<td>89.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>seq2seq (baseline)</td>
<td>14.0%</td>
<td>28.0%</td>
<td>48.0%</td>
</tr>
</tbody>
</table>

Almost half of the baseline model’s utterances are incorrect.
Human Evaluation

How **correct** are the generated utterances (wrt. **current game state and target**)?
Correct utterances are more likely to lead to **task completion**.

<table>
<thead>
<tr>
<th></th>
<th>Fully correct</th>
<th>Partially correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human (ceiling)</td>
<td>89.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>seq2seq (baseline)</td>
<td>14.0%</td>
<td>28.0%</td>
<td>48.0%</td>
</tr>
<tr>
<td>Block Counters</td>
<td>25.0%</td>
<td>36.0%</td>
<td>32.0%</td>
</tr>
</tbody>
</table>

The **Block Counter** Model produces **significantly more fully/partially correct utterances** and **significantly fewer incorrect ones** than the baseline (even if it is still pretty far from human performance)
What can the neural Architect do?

Builder has just placed the red block in the top right corner

A: “perfect! now place a red block to the left of that”

The neural architect gives natural, fluent block-by-block instructions that contain color terms and spatial relations
Model A: okay, we'll start with a row of three red blocks, place a red block in front of you.
Model A: okay, we'll start with a row of three red blocks, place a red block in front of you.

Blue: Model Architect
Red: Mistakes

The block counters model has no access to complex shapes (rows vs. towers/columns).
Model A: okay, we'll start with a row of three red blocks, place a red block in front of you

Human A: hello builder, i will tell you this. it appears we are creating a belltower. but first i will start with step by step instructions. we will start with green blocks

Humans give much more complex instructions. ("belltower")
Model A: okay, we'll start with a row of three red blocks, place a red block in front of you

Human A: hello builder, i will tell you this. it appears we are creating a belltower. but first i will start with step by step instructions. we will start with green blocks

Humans give much more complex instructions. (“belltower”)

There may be different possible next actions (making automatic evaluation difficult)
**Human B:** is this good?

**Human A:** yes, one moment

**Model A:** perfect! now put a purple block on top of that

**Human A:** on the 8th block to the left of the green block, please create the same structure with red blocks

There may be different possible next actions (making automatic evaluation difficult)
**Human A:** excellent builder

**Model A:** now place an orange block on top of that

**Human A:** now we must create the bell. please start by extending 4 orange blocks down from the middle purple block, as if it were hanging
STARTING POINT FOR BUILDER: ACTION PREDICTION

(Jayannavar, Narayan-Chen, Hockenmaier, ACL 2020)
Builder: Task and Challenges

Understand and execute instructions
B. needs to understand **descriptions of structures**
B. needs to understand **spatial relations**
B. needs to understand utterances in the **current context**

*Execution:* **place and remove blocks** in the $11 \times 9 \times 11$ build region

Ask clarification questions as needed
B. needs to know what information is **missing or unclear**
B. needs to know when instructions **can’t be executed**

Future work: Requires execution model
The Builder Action Prediction (BAP) Task

Predict the **sequence of actions** (block placements and/or removals) that a Builder performed at a particular point in a human-human game.

<Architect> go the middle and place an orange block two spaces to the left
<Architect> now make a staircase with 2 stairs left and 2 right with orange
<Architect> so it will look like a v
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to \( t = 0 \)
- World state grid \( W^{(0)} \)

Predicts:
- Sequence of \( B \) actions \( a^{(0)} \ldots a^{(t+1)} \) with \( a^{(0)} = \text{START} \)
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of $B$ actions $a^{(0)} ... a^{(t+1)}$ with $a^{(0)} = \text{START}$
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to \( t = 0 \)
- World state grid \( W^{(0)} \)

Predicts:
- Sequence of \( B \) actions \( a^{(0)} \) \( \ldots \) \( a^{(t+1)} \) with \( a^{(0)} = \text{START} \)
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of $B$ actions $a^{(0)} ... a^{(t+1)}$ with $a^{(0)} = \text{START}$
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to \( t = 0 \)
- World state grid \( W^{(0)} \)

Predicts:
- Sequence of \( B \) actions \( a^{(0)} \ldots a^{(t+1)} \) with \( a^{(0)} = \text{START} \)
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of $B$ actions $a^{(0)} \ldots a^{(t+1)}$ with $a^{(0)} = \text{START}$
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of B actions $a^{(0)} ... a^{(t+1)}$ with $a^{(0)} = \text{START}$
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of $B$ actions $a^{(0)} \ldots a^{(t+1)}$ with $a^{(0)} = \text{START}$
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of $B$ actions $a^{(0)} ... a^{(t+1)}$ with $a^{(0)} = \text{START}$
Our Model

Encoder-decoder network with GRU backbone

Inputs:
- Game history up to $t = 0$
- World state grid $W^{(0)}$

Predicts:
- Sequence of $B$ actions $a^{(0)} \ldots a^{(t+1)}$ with $a^{(0)} = \text{START}$
  and $a^{(t+1)} = \text{STOP}$
Game History
GRU Encoder

Encodes game history as flat sequence of tokens

\(<A>\) place a red block on the ground \(</A>\) \(<B>\) like this ? \(</B>\)
Different amounts of history

H3

\(<A>\) on the same plane facing you, leave a space and then put 2 red blocks down in a row

\(<\text{builder\_putdown\_red}>\)
\(<\text{builder\_putdown\_red}>\)
\(<\text{builder\_putdown\_red}>\)
\(<\text{builder\_pickup\_red}>\)

\(<A>\) and the same thing on the other side
World State CNN Encoder

Encodes the world state at each time step

Input: 11×9×11 3D grid
Each grid cell is represented as a 7-dim 1-hot vector of its block color (or empty)
Encoding Action History

Actions that follow each other often affect adjacent grid cells

Action history weights
Concatenate an action history weight $\alpha \in \{0,1,2,3,4,5\}$ to each cell’s vector representation

The last five actions get weights from 1 through 5 (least to most recent)
All other actions are weighted 0
Encoding the Builder’s Perspective

**Spatial relations** in instructions (e.g. “left”) often depend on B’s perspective (current position and orientation).

Encode B’s perspective using **perspective coordinates**.

Given: a cell $c$ and the absolute coordinates of the cell $\langle x_c, y_c, z_c \rangle$.

Compute relative coordinates of the cell $\langle x'_c, y'_c, z'_c \rangle$ wrt B’s current position $\langle x_B, y_B, z_B \rangle$ and orientation (pitch and yaw angles).
Action Sequence Decoder

GRU backbone input:

11-dim vector \( \mathbf{a} \) representing action taken at last timestep

<table>
<thead>
<tr>
<th>Action type</th>
<th>Block color (all 0s if removal)</th>
<th>Absolute coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>R</td>
<td>x</td>
</tr>
<tr>
<td>r</td>
<td>B</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>z</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

Action Sequence Decoder

Game History

- GRU
- \( W_c \)
- \( \mathbf{a} \)
- column

World State

- \( W^{(0)} \)
- \( k \times k \times k \) conv
- ReLU
- \( 1 \times 1 \times 1 \) conv
- ReLU
- \( k \times k \times k \) conv
- ReLU

Action Sequence Decoder

- GRU
- GRU
- GRU
- \( W_c \)
- \( W_r \)
- \( W_e \)
- \( \mathbf{a}^{(1)} \)
- \( \mathbf{a}^{(2)} \)

START
Action Sequence Decoder

CNN action predictor
Action Sequence Decoder

CNN action predictor

STOP token predictor:

Conditioned on action predictor representation

Predicts the likelihood of ending the action sequence

Final prediction:

Distribution over all possible actions in the grid + STOP probability
Data Augmentation

<\text{A}> now take a \textcolor{red}{red} block
<\text{A}> place it in the square diagonally to the right of the \textcolor{purple}{purple} block
<\text{A}> nice
<\text{B}> thank you
<\text{A}> now do the same thing on the other side but with an \textcolor{orange}{orange} block
<\text{B}> other side of the \textcolor{purple}{purple} or \textcolor{yellow}{yellow}?
<\text{A}> \textcolor{yellow}{yellow}

<\text{A}> now take a \textcolor{purple}{purple} brick
<\text{A}> place it in the square diagonally to the right of the \textcolor{green}{green} block
<\text{A}> alright
<\text{B}> thank you
<\text{A}> now do the same thing on the other side however with an \textcolor{blue}{blue} block
<\text{B}> other side of the \textcolor{green}{green} or \textcolor{red}{red}?
<\text{A}> \textcolor{red}{red}
Evaluation: Net Actions F1

Net actions ignore the order of actions and blocks that were placed and removed in the same sequence.
**Evaluation: Net Actions F1**

*Net actions* ignore the order of actions and blocks that were placed and removed in the same sequence.
Evaluation: Net Actions F1

We compute a **micro-averaged F1** between net actions in the **ground truth (human)** sequence $A_h$ and in the model’s **predicted** sequence $A_m$. 
Quantitative Results

3,709 train / 1,616 test / 1,331 dev B action sequences (splits across target structures)

Supervised training to minimize cross entropy loss; greedy decoding

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAP-base</td>
<td>11.8</td>
<td>12.4</td>
<td>14.6</td>
</tr>
</tbody>
</table>

Richer game history helps increase performance
Quantitative Results

3,709 train / 1,616 test / 1,331 dev B action sequences (splits across target structures)

Supervised training to minimize cross entropy loss; greedy decoding

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAP-base</td>
<td>11.8</td>
<td>12.4</td>
<td>14.6</td>
</tr>
<tr>
<td>+ action history</td>
<td>14.6</td>
<td>18.2</td>
<td><strong>19.7</strong></td>
</tr>
</tbody>
</table>

Richer world state representations help increase performance
Quantitative Results

3,709 train / 1,616 test / 1,331 dev B action sequences (splits across target structures)

Supervised training to minimize cross entropy loss; greedy decoding

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAP-base</td>
<td>11.8</td>
<td>12.4</td>
<td>14.6</td>
</tr>
<tr>
<td>+ action history</td>
<td>14.6</td>
<td>18.2</td>
<td>19.7</td>
</tr>
<tr>
<td>+ perspective</td>
<td>15.7</td>
<td>18.7</td>
<td>18.8</td>
</tr>
</tbody>
</table>

Richer world state representations help increase performance
Quantitative Results

3,709 train / 1,616 test / 1,331 dev B action sequences (splits across target structures)

Supervised training to minimize cross entropy loss; greedy decoding

Data augmentation helps increase performance. Now we get the best results with the full world state representation.

<table>
<thead>
<tr>
<th></th>
<th>Trained on original data</th>
<th>On augmented dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H1</td>
<td>H2</td>
</tr>
<tr>
<td>BAP-base</td>
<td>11.8</td>
<td>12.4</td>
</tr>
<tr>
<td>+ action history</td>
<td>14.6</td>
<td>18.2</td>
</tr>
<tr>
<td>+ perspective</td>
<td>15.7</td>
<td>18.7</td>
</tr>
</tbody>
</table>

Only 21.2 F1? That's pretty bad, right?
What can the Neural Builder do?

Perfect interpretation of "do it one more time"
("it" = "place two blue blocks on top of the edges of the line")

Architect: now place two blue blocks on top of the edges of the line

Human-Human Game History

Last Built State

Builder’s Actions

Architect: do it one more time

Builder’s Actions

Penultimate Built State
Human-Human Game History

What can the Neural Builder do?

Human Builder’s Actions

Plausible interpretation of “and do the same on the other side”

"the same" = “the next two blocks" "the other side" =??

Neural Builder’s Actions

Human-Human Game History

"the next two blocks will be off the corners of each of those, in the direction of the last yellow block."

Architect: like that, or somewhere else?

Builder: add one more block to the end of that on your side?

Architect: the next two blocks will be off the corners of each of those, in the direction of the last yellow block.

Penultimate Built State

Last Built State

Architect: the next two blocks will be off the corners of each of those, in the direction of the last yellow block.

Architect: and do the same on the other side

Builder: like that, or somewhere else?
What can the Neural Builder do?

Correct interpretation of "add one red block on top of that"
.... but the human anticipated the next steps

**Human-Human Game History**

**Penultimate Built State**

**Last Built State**

**Architect:**
same on the other side

**Neural Builder's Actions**

**Builder's Actions**

Architect:
add one red block on top of that
What Have We Accomplished?

Supervised training on relatively small amounts of data with simple end-to-end neural models and no linguistic annotation yields (surprisingly?) decent baseline models:

– The Architect gives block-by-block instructions that are fluent and often (but far from always) correct
– The Builder executes instructions in ways that are often correct or plausible
– The Builder shows some understanding of complex concepts and context (row, middle, gap, the same, other side)
What Remains To Be Done?

We haven’t yet *solved the tasks* we started working on
– We need *higher accuracy* of instructions and executions
– We want the Architect to generate *richer, more diverse utterances*

This requires *richer models*, possibly *more data*,
and other *training regimes*
– What’s the role of *explicit domain knowledge*?
– Naively using *3D CNNs* as world state representations for the architect
doesn’t seem to work, because there is not enough supervision.
What Remains To Be Done?

Fully interactive agents require further capabilities:
– Both systems need to be trained for task completion
– The Builder needs to speak, but this requires knowing what to ask
– Both agents need to know when to speak