Linguistic scaffolds for policy learning

Jacob Andreas
Berkeley → Microsoft Semantic Machines → MIT
Linguistic scaffolds for policy learning

Work on language!

Jacob Andreas

Berkeley → Microsoft Semantic Machines → MIT
What RL can do for language

What language can do for RL

replace the last letter of the word
drop head
change the final letter to t i
add a z if the last character is a
every vowel becomes y
change only the first consonant to
first & last 3 letters
delete every vowel
replace all n s with c
What RL can do for language

Daniel Fried

Ronghang Hu

Volkan Cirik

w/ Anja Rohrbach, L.P. Morency, Taylor Berg-Kirkpatrick, Trevor Darrell and Dan Klein
Turn right and walk through the kitchen. Go right into the living room and stop by the rug.

[Anderson et al. 18]
A reference game
“glasses"
“glasses”
“glasses"
“glasses"
The rational speech acts model

\[ L_o(. \mid \text{glasses}) \]
\[ L_o(. \mid \text{hat}) \]

1/2
0

1/2
1

[Frank & Goodman 12]
The rational speech acts model

\[ L_0(. \mid \text{glasses}) \]
\[ L_0(. \mid \text{hat}) \]
\[ S_1(\text{glasses} \mid .) \propto L_0(. \mid \text{glasses}) \]
\[ S_1(\text{hat} \mid .) \]

<table>
<thead>
<tr>
<th></th>
<th>1/2</th>
<th>1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_0(. \mid \text{glasses}) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_0(. \mid \text{hat}) )</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( S_1(\text{glasses} \mid .) \propto L_0(. \mid \text{glasses}) )</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>( S_1(\text{hat} \mid .) )</td>
<td>0</td>
<td>2/3</td>
</tr>
</tbody>
</table>

[Frank & Goodman 12]
The rational speech acts model

\[ L_1( . | \text{glasses} ) \propto S_1( \text{glasses} | . ) \]

\[ L_1( . | \text{hat} ) \]  
\[ S_1( \text{glasses} | . ) \propto L_0( . | \text{glasses} ) \]

\[ S_1( \text{hat} | . ) \]  

[Frank & Goodman 12]
Q: Do you know what time it is?
Q: Do you know what time it is?
A: Yes
Pragmatics

Q: Do you know what time it is?
A: Yes

I find his cooking very interesting.

[Grice 70]
RSA game tree

Speaker

Hat

Glasses
RSA game tree: as speaker

- **Speaker**
  - **Hat**
  - **Glasses**

- **Listener**
  - **Hat**
  - **Glasses**

Values:
- +1
- -1
RSA game tree: as speaker

speaker

hat

hat

(listener)

+1

-1

+1

-1

glasses

glasses
RSA game tree: as speaker

speaker

hat

(glasses)

(listener)

hat

+1

-1

+1

-1

(glasses)
RSA game tree: as listener

(speaker) listener

glasses

?
RSA game tree: as listener

(speaker) listener

Language use is gameplay!

glasses
A recipe for pragmatic text generation

1. Train a base **listener** model
A recipe for pragmatic text generation

1. Train a base **listener** model

2. Train a reasoning **speaker** to win when playing with the listener
Application: image captioning

1. Train an image retrieval / gen model

*a snake is slithering away from Jenny*
2. **Describe** images using the listener model for search at inference time

[A & Klein 16, Vedantam et al. 17]
2. **Describe** images using the listener model as a training-time reward ("self-play")

[Yu et al. 16, Mao et al. 16]
Descriptive captions [Vedantam et al. 17]

**seq2seq** captioner: *this bird has a yellow breast with a short pointy bill*

**pragmatic** captioner: *a small yellow bird with black stripes on its body and black stripe on the wings.*
Contrastive captions without contrastive data!

Mike is holding a baseball bat.

The snake is slithering away from Mike & Jenny.

[A & Klein 16]
Application: instruction generation

1. Train a base **instruction following** model

2. Train an **instruction generation** model to get the follower to goal states
Application: instruction generation

**seq2seq:** Walk past the dining room table and chairs and wait there.

**speaker-listener:** Walk past the dining room table and chairs and take a right into the living room. Stop once you are on the rug.

**human:** Turn right and walk through the kitchen. Go right into the living room and stop by the rug.

[Fried, Hu, Cirik et al. 18]
Listener mode

**human:** Go through the door on the right and continue straight. Stop in the next room in front of the bed.

- **orange**: trajectory without pragmatic inference
- **green**: trajectory with pragmatic inference

(seq-to-seq) speaker-listener

[Fried, Hu, Cirik et al. 18]
The rules of the game

glasses

+1
The rules of the game

hat

+1
Killer robots [Lewis et al. 17]

Bob: *i can i i everything else ................
Alice: *balls have zero to me to me to me to me to me to me to me to me to me to
Bob: *you i everything else ..............
Alice: *balls have a ball to me to me to me to me to me to me to me to me
Killer robots [Lewis et al. 17]

Bob: *i can i i everything else ..............
Alice: *balls have zero to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me to me t
Problems to work on

How do we use tools like self-play and tree search while remaining within the rules of natural language?

How do we do efficient search in string-valued action spaces?
Problems to work on

How do we use tools like self-play and tree search while remaining within the rules of natural language?

How do we do efficient search in string-valued action spaces?
What language can do for RL

w/ Dan Klein and Sergey Levine
A crafting game

make planks

make sticks
Learning with sketches

- Get wood
- Use saw
- Get wood
- Use axe
The options framework

[Sutton et al. 99]
Unsupervised option learning

[Bacon & Precup 16]
Learning with intermediate rewards

[Kearns & Singh 02, Kulkarni et al. 16]
Segmenting demonstrations

[Stolle & Precup 02, Fox & Krishnan et al. 16]
Learning from sketches

get wood \rightarrow use saw

[A, Klein & Levine 17]
Modular policies

- **\( \pi_1 \)**: get wood
- **\( \pi_2 \)**: use saw
- **\( \pi_3 \)**: get wood, use axe
Modular policies

- get wood
- use saw
- $\pi_1$
- $\pi_2$
- get wood
- use axe
- $\pi_3$
Modular policies

π_1

TURN LEFT

get wood
Results: crafting game
Results: crafting game

- Reward

- Sketches: modular
  - Unsupervised

- Sketches: joint

x 10^6 episodes
Results: locomotion
Results: locomotion

Reward

Sketches: modular

Sketches: joint

Unsupervised

x $10^8$ timesteps
What if I don’t get a sketch at test time?
What if I don’t get a sketch at test time?
Moral

A little bit of (structured) language goes a long way!
Beyond structured sketches

Language learning

itch → itctch

first & last 3 letters

Learning from demonstrations

emboldens → emboldecs
dogtrot → dogtrot
loneliness → locelicess
vein → ???

[A, Klein & Levine 17]
Beyond structured sketches

Language learning

itch ➔ itctch

first & last 3 letters

Learning from demonstrations

emboldens ➔ emboldecs
dogtrot ➔ dogtrot
loneliness ➔ loceliciess
vein ➔ ???
Pretraining via language learning

\[ f([\text{wonderful}; \eta,]) \rightarrow \text{wonful} \]

[first & last 3 letters]

[Branavan et al., 09]
Concept learning

\[ L(f(\cdot; \eta, \bigcirc)) \]
every vowel becomes i
Concept learning

$L(f(n))$

every vowel becomes i

128.6

every vowel becomes i

emboldens

vein

loneliness

emboldens

vein

loneliness

emboldens

vein

loneliness

emboldecs

veic

locelicess
Concept learning

\[ L(f(\cdot; \eta, \cdot)) \]

- emboldens
- vein
- loneliness

- emboldecs
- veic
- locelicens

- every vowel becomes i

- change consonants to c

128.6
52.3
Concept learning

$L(f(; \eta, \quad ))$

- replace n with c: 8.3
- change consonants to c: 52.3
- every vowel becomes i: 128.6

emboldens
vein
loneliness

emboldecs
veic
locelicious
Prediction

\[ f(\cdot; \eta, \cdot) \]

replace \( n \) with \( c \)
loonies $f(\cdot; \eta, \cdot)$

replace n with c
loonies $f(\ ; \eta, \ )$ \hspace{1cm} \rightarrow \ loocies

replace n with c
As multitask learning

**Pretraining data**
- wonderful
- glabrous
- itch

**Training data**
- emboldens
- vein
- loneliness

$\arg\min_{\eta} L(f(\text{itch} | \text{itch}; \eta, \theta))$

$\arg\min_{\eta} L(f(\text{veic} | \text{vein}; \eta, \theta))$

---

[Caruana 97]
As inverse reinforcement learning

\[
\text{cost function}
\]

\[
\arg \min \hat{E}_{\tau \sim \pi} [L(f(\tau | \text{vein}; \eta, c))]
\]

replace n with c
As a language game...

arg min

vein $\rightarrow$ replace n with c $\rightarrow$ veic

$\arg\min$ ???

$f \circ L$

speaker model

listener loss
Results: string editing accuracy

- Identity: 18
- Multitask: 50
- Meta: 62
- This Work: 76
### Results

<table>
<thead>
<tr>
<th>examples</th>
<th>true description</th>
<th>true output</th>
</tr>
</thead>
<tbody>
<tr>
<td>emboldens</td>
<td>replace all n s with c</td>
<td>loocies</td>
</tr>
<tr>
<td>kisses</td>
<td></td>
<td>loonies</td>
</tr>
<tr>
<td>loneliness</td>
<td>change any n to a c</td>
<td>loocies</td>
</tr>
<tr>
<td>vein</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dogtrot</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 6: Example predictions for string editing.
Problems

How do we bootstrap from (unannotated) exploration of the environment alone?

How good are inferred descriptions as explanations?
Problems

How do we bootstrap from (unannotated) exploration of the environment alone?

How good are inferred descriptions as explanations?
Conclusions
Conclusions

**Use RL in NLP** by formulating language generation / understanding as reward maximization rather than supervised learning.
Use language in (I)RL as a scaffold for learning options, goal representations, cost functions.

Languages encode 100k years of accumulated knowledge about which abstractions are useful — take advantage of it!
Thanks!

also...

Looking for NLP jobs? Ask me about Microsoft!
Applying to PhD programs? Ask me about MIT!
jaandrea@microsoft.com