Naturalizing a programming language via interactive learning

Sida I. Wang, Samuel Ginn
Percy Liang, Christopher D. Manning

ACL 2017, Vancouver BC
how can they perform more complex actions?
Semantic parsing

natural language $x$ to an executable program $z$

- human produces some utterance
- *add 3 red blocks on this*

Semantic parsing

natural language $x$ to an executable program $z$

- human produces some utterance
- *add 3 red blocks on this*

- converts utterance to a program

Semantic parsing

natural language $x$ to an executable program $z$

- human produces some utterance
  - $add\ 3\ red\ blocks\ on\ this$

- converts utterance to a program
  - $(\text{loop}\ 3\ (\text{add}\ red\ top))$

**Semantic parsing**

natural language $x$ to an executable program $z$

- human produces some utterance
- $add\ 3\ red\ blocks\ on\ this$

- converts utterance to a program
- $(loop\ 3\ (add\ red\ top))$
- execute the program $z$
- produce a result $y$

Semantic parsing

natural language $x$ to an executable program $z$

- human produces some utterance
- *add 3 red blocks on this*

- converts utterance to a program
- *(loop 3 (add red top))*
- execute the program $z$
- produce a result $y$

Collect a static dataset

*increase temperature by 3C*
setTemp(getTemp()+3)

*what is the largest state*
answer(A,largest(A,state(A))))

*people with children born in Vancouver*
Children.PlaceOfBirth.Vancouver

*add 3 red blocks on this*
(loop 3 (add red top))
Collect a static dataset

*increase temperature by 3C*

setTemp(getTemp()+3)

*what is the largest state*

answer(A,largest(A,state(A))))

*people with children born in Vancouver*

Children.PlaceOfBirth.Vancouver

*add 3 red blocks on this*

(loop 3 (add red top))

then fit a model.
Collect a static dataset

increase temperature by 3C
setTemp(getTemp()+3)

what is the largest state
answer(A,largest(A,state(A))))

people with children born in Vancouver
Children.PlaceOfBirth.Vancouver

add 3 red blocks on this
(loop 3 (add red top))

then fit a model.

then deploy the system
With a static system and static dataset
With a static system and static dataset

we are stuck when these systems misunderstand us
With a static system and static dataset we are stuck when these systems misunderstand us.
With a static system and static dataset we are stuck when these systems misunderstand us.

systems with interactive learning ability built-in
With a static system and static dataset

we are stuck when these systems misunderstand us

systems with interactive learning ability built-in

learn from users in real-time
Test ideas in blocks world

blocks world: intuitive, easy to crowdsourced, and captures some unsolved problems
Interactive learning language games

Wittgenstein. 1953. Philosophical Investigations:

Language derives its meaning from use.

'block' 'pillar' 'slab' 'beam'.
Start
start

√

goal
SHRDLURN

has a goal

remove red

performs actions
does not talk

has language
has a goal
has language

performs actions
does not talk

add(leftmost(hascolor(red)),red)
add(red, hascolor(cyan))
remove(hascolor(red))
remove(leftmost(hascolor(red)))
SHRDLURN

- **start**
  - *remove red*
  - *add (leftmost (hascolor (red)), red)*
  - *add (red, hascolor (cyan))*
  - *remove (hascolor (red))*
  - *remove (leftmost (hascolor (red)))*

- **goal**

**has a goal**

**has language**

**performs actions**

**does not talk**

**remove red**

7
Learning signal is weak

- user provides a label via selection
- system can learn from denotation, logical form, or final reward
Learning signal is weak

• user provides a label via selection

• system can learn from denotation, logical form, or final reward

• for any learning to happen using these methods, the correct answer has to fall inside the search space
Learning signal is weak

- user provides a label via selection

- system can learn from denotation, logical form, or final reward

- for any learning to happen using these methods, the correct answer has to fall inside the search space

- works for short programs:
  - \( \text{answer}(A, \text{largest}(A, \text{state}(A)))) \)
  - \( \text{remove}(\text{leftmost}(\text{hascolor}(\text{red})))) \)
Learning signal is weak

- user provides a label via selection

- system can learn from denotation, logical form, or final reward

- for any learning to happen using these methods, the correct answer has to fall inside the search space

- works for short programs:
  - answer(A, largest(A, state(A))))
  - remove(leftmost(hascolor(red))))

- cannot possibly scale to more complex programs
Cannot possibly keep up

**legs of height 3 with 3 spaces apart**

number of programs of this length \(> 10^{100}\)
Cannot possibly keep up legs of height 3 with 3 spaces apart

number of programs of this length $> 10^{100}$

need stronger supervision to produce such programs
demonstrations, instructions, definitions
Naturalizing a programming language

start with a core programming language

• starting point of definitions
Naturalizing a programming language

start with a core programming language
  • starting point of definitions

a community of users interact with the system
  • more definition using core and previous definitions
  • leg of height 3 := brown column of height 3
Naturalizing a programming language

start with a core programming language
  • starting point of definitions

a community of users interact with the system
  • more definition using core and previous definitions
  • leg of height 3 := brown column of height 3

induce grammar rule
Naturalizing a programming language

start with a core programming language
  • starting point of definitions

a community of users interact with the system
  • more definition using core and previous definitions
  • leg of height 3 := brown column of height 3

induce grammar rule
  • ”learning macros from crowd programming”
  • no explicit arguments and variables
Voxelurn

world is a set of objects
with relations

voxels: \((x, y, z, \text{color})\)

relations: left, top, front, etc.

actions: select, add, move
Core language

• controls: if, foreach, repeat, while

• block-structured scoping
  • , [], isolate
Core language

• controls: if, foreach, repeat, while
• block-structured scoping
  • , [], isolate
• lambda DCS for sets

yellow blocks in row 1

\( \lambda \)-DCS: has color yellow and has row 1
Core language

• controls: if, foreach, repeat, while
• block-structured scoping
  • , [], isolate
• lambda DCS for sets
  yellow blocks in row 1
  
\(\lambda\)-DCS: has color yellow and has row 1

• selection as the default argument
  • add red top (to selected)
Lets make some trees

- define new things in terms of what’s already defined
- trace back to the core language

<table>
<thead>
<tr>
<th></th>
<th>initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>add palm tree</td>
</tr>
<tr>
<td>3</td>
<td>select 3 to the left</td>
</tr>
</tbody>
</table>

[Diagram showing steps to add a yellow palm tree]
Palm tree example

define new concepts in terms of what’s already defined
everything trace back to the core language
add palm tree:
  add brown trunk height 3:

   go to top:

   add leaves here:
Palm tree example

define new concepts in terms of what’s already defined
everything trace back to the core language
add palm tree:
    add brown trunk height 3:
        add brown top 3 times:

    go to top:

    add leaves here:
Palm tree example

define new concepts in terms of what’s already defined

everything trace back to the core language

add palm tree:
  
  add brown trunk height 3:
    
    add brown top 3 times:
      
      repeat 3 [add brown top]
    
  go to top:

  add leaves here:
Palm tree example

define new concepts in terms of what’s already defined
everything trace back to the core language
add palm tree:
    add brown trunk height 3:
        add brown top 3 times:
            repeat 3 [add brown top]
    go to top:
        select very top of all
    add leaves here:
Palm tree example

define new concepts in terms of what’s already defined
everything trace back to the core language

add palm tree:

  add brown trunk height 3:

    add brown top 3 times:

      repeat 3 [add brown top]

  go to top:

    select very top of all

add leaves here:

  select left or right or front or back; add green
Inside a definition

unparsable head: *add red left 3 times*
Inside a definition

unparsable head:  *add red left 3 times*

parsable body:  *repeat 3 add red left*

derivation of the body:  *(loop 3 (add red left))*
Inside a definition

unparsable head: \textit{add red left 3 times}
	parsable parts of the head: \textit{red, left, 3, add red left}

parsable body: \textit{repeat 3 add red left}

derivation of the body: \textbf{(loop 3 (add red left))}
unparsable head: \textit{add red left 3 times}
unparsable head: \textit{add red left 3 times}
Parsable body

unparsable head: *add red left 3 times*

parsable body: *repeat 3 add red left*

\[(\text{loop } 3 \ (\text{add red left}))\]
Substitute matches

unparsable head: \textbf{add red left} 3 times

parsable body: \textbf{repeat 3 add red left}

$$(\text{loop 3 (add red left)})$$

\begin{itemize}
\item \texttt{repeat}\hspace{2cm}Number\hspace{2cm}Action
\item 3\hspace{2cm}3\hspace{2cm}(add red left)
\item add\hspace{2cm}red\hspace{2cm}left
\end{itemize}
Substitute matches

unparsable head: `add red left` 3 times

parsable body: `repeat 3 add red left`

(indloop 3 (add red left))

induced rule: \( A \rightarrow A \ 3 \text{ times} : \lambda A. (\text{loop} \ 3 \ A) \)
Substitute matches

unparsable head: \textbf{add red left} \textbf{3} times

parsable body: \textbf{repeat} \textbf{3} \textbf{add red left}

induced rule: \( A \rightarrow A \ N \ \text{times} : \lambda A \ N.(\text{loop } N \ A) \)
Matches not unique

more abstract

\[
\text{add red left} \begin{array}{c}
3 \\
\end{array}
\text{times}
\]

\[
\text{repeat} \begin{array}{c}
3 \\
\end{array}
\text{add red left}
\]

\[
A \rightarrow A \ N \ \text{times} : \lambda A \ N. (\text{loop } N \ A)
\]

less abstract

\[
\text{add} \begin{array}{c}
\text{red} \\
\text{left} \\
3 \\
\end{array}
\text{times}
\]

\[
\text{repeat} \begin{array}{c}
3 \\
\end{array}
\text{add} \begin{array}{c}
\text{red} \\
\text{left} \\
\end{array}
\]

\[
A \rightarrow \text{add} \ C \ D \ N \ \text{times} : \lambda C \ D \ N. (\text{loop } N \ (\text{add } C \ D))
\]
Take highest scoring ones

- a packing is a set of non-overlapping potential matches
  - maximal packing – no span can be added

add red left 3 times

repeat 3 add red left

Zettlemoyer and Collins 2005, 2007, Kwiatkowski et al., 2010
Take highest scoring ones

- a packing is a set of non-overlapping potential matches
  - maximal packing – no span can be added

add red left 3 times
repeat 3 add red left

Zettlemoyer and Collins 2005, 2007, Kwiatkowski et al., 2010
Take highest scoring ones

- a packing is a set of non-overlapping potential matches
  - maximal packing – no span can be added

Add 3 times

Repeat 3 add red left

- abstract away the highest scoring maximal packing

\[ P_l^* = \arg\max_{P \in \text{packing}(M)} \sum_{d \in P} \text{score}(d). \]

- solve with a dynamic program

Zettlemoyer and Collins 2005, 2007, Kwiatkowski et al., 2010
Model over derivations

log-linear model with features $\phi(d, x, u)$:

$$p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta)$$

$x$: add two chairs 5 spaces apart

$z = \text{formula}(d) : (\text{blk} \ (\text{loop} ...))$
Learning from denotations

\[ p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta) \]

\( x: \text{add two chairs 5 spaces apart} \)

\( z = \text{formula}(d) : (\text{:blk (:loop ...)}) \)
Learning from denotations

\[ p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta) \]

\[ p_\theta(y \mid x, u) = \sum_{d : \text{Exec}(d) = y} p_\theta(d \mid x, y) \]

\( x : \text{add two chairs 5 spaces apart} \)

\( z = \text{formula}(d) : (: \text{blk} (: \text{loop}...)) \)

\( y : \)
Learning from denotations

\[ p_\theta(d \mid x, u) \propto \exp(\phi(d, x, u) \cdot \theta) \]

\[ p_\theta(y \mid x, u) = \sum_{d: \text{Exec}(d) = y} p_\theta(d \mid x, y) \]

\[ x: \text{add two chairs 5 spaces apart} \]

\[ z = \text{formula}(d) : (: \text{blk} (: \text{loop...})) \]

L1 penalty and update with AdaGrad
Features

- **generic**: ruleId, span
- **rule type**: core? induced? used?
- **social**: authorId, (authorId, userId), self?
- captures user community
Experiments

can crowdworkers provide such strong supervision?

initial users have to learn the core language

following user can build on previous users

*chair legs of height 3*

```scheme
(:s (:s (:blkr (:s (:loop (number 3) (:s (: add brown here) (:for (call adj top this) (: select)))) (:loop (number 3) (:for (call adj bot this) (: select)))) (:loop (number 3) (:for (call adj left this) (: select)))) (:s (:s (:s (:blkr (:s (:loop (number 3) (:s (: add brown here) (:for (call adj top this) (: select)))) (:loop (number 3) (:for (call adj bot this) (: select)))) (:loop (number 3) (:for (call adj back this) (: select)))) (:blkr (:s (:loop (number 3) (:s (: add brown here) (:for (call adj top this) (: select)))) (:loop (number 3) (:for (call adj bot this) (: select)))) (:loop (number 3) (:for (call adj right this) (: select)))) (:blkr (:s (:loop (number 3) (:s (: add brown here) (:for (call adj top this) (: select)))) (:loop (number 3) (:for (call adj bot this) (: select)))))
```
Experiments

- users built great structures?
Experiments

- users built great structures! (Show leaderboard)
Leaderboard

go down 5, go up and back, left 2
add green monster, deer head
Leaderboard

8 upvotes
MT_A #37QW 391 blks
right 6; back 3; add yellow column 3
black 10×10×10 frame, green cube size 4

7 upvotes
MT_A #3X87 993 blks
initial black bottom
barrier blue draw right; barrier blue draw back
right 6; back 3; up 3; power pellet
back 3; repeat 2 [ right 6]; barrier blue point front
left 42 in initial.

7 upvotes
MT_A #3PWW 972 blks
initial select left 6 select front 8 black 10×10×10 frame
black 10×10×10 frame move front 10 move left 9 move bot 8
move front 7 move left 9 move front move front 9
move left 9 move bot 1 black 10×10×10 frame
left 13; 12; 10; move left 7.
Leaderboard

7 upvotes

MT_A #3DBQ 143 blks

select back 4; add brn  
select left 2; add brn  
select front 4; add brn  
select top; select back 4  
add ylw  
select front 4  
add yellow tower 2  
select right 2; select bot 2  
add ylw tower 2; select back 4; select bot 2  
add ylw; select top  
add brown tower 2  
select front 4; select bot  
add tower brn/ylw

7 upvotes

MT_A #3DHE 427 blks

initial  
select front 6; move right 6  
add row blue right 10  
select left; move front 10  
add plate blue 8x10  
select left 8; move bot  
add row blue right 10; select left; move front 10  
add blue right 10; select left; move front 10

add brn, add ylw, add brn tower 2
add plate 8x10
Leaderboard

MT_A #3EJJ 686 blks
initial cover select origin; back corner; back 3 Ladder&board pool&legs select origin left 2; top 1; add white select white select very right of this

MT_A #3WQ3 723 blks
initial blue bottom skip remove back 12 skip remove right 12 left skip remove front 12 back skip remove left 12 right; front skip add green back 12 skip add green right 12 move left skip add green front 12 move back skip add green left 12

skip add green back 12, skip remove back 12
Setup

• qualifier: build a fixed structure

• post-qual: over 3 days build whatever they want

• prizes for best structures
  • day 1: bridge, house, animal
  • day 2: tower, monster(s), flower(s)
  • day 3: ship(s), dancer(s), and castle

• prize for top h-index
  • a rule (and its author) gets a citation whenever it is used
Basic statistics

- 70 workers qualified, 42 participated, 230 structures

- 64,075 utterances, 36,589 accepts

  - each accept leads to a datapoint labeled by derivation(s)

- 2495 definitions combining over 15k commands, 2817 induced rules (¡100 core)
Is naturalization happening

maybe best to use the core language and program...
Is naturalization happening

maybe best to use the core language and program...

**core**: utterance parsable with the initial core grammar

**induced**: parsable with induced grammar by not by core

**unparsable**: not parsable at all
Is naturalization happening

- 67% of all at the end (up from 0 in the beginning)
- 72.9% of all accepted, and 85.9% of the last 10k accepted
Expressive power

- cumulative average of string.length in program / utterance
- $\text{len}(z)/\text{len}(z)$ is very stable at 2 for core language
- varies greatly by user
Modes of naturalization

short forms:

left, l, mov left, go left, ¡, sel left

br, blk, blu, brn, orangeright, left3

add row brn left 5
:= add row brown left 5
Modes of naturalization

syntactic:

\[
\begin{align*}
go \text{ down and right} & \quad := \quad \text{go down; go right} \\
\text{select orange} & \quad := \quad \text{select has color orange} \\
\text{add red top 4 times} & \quad := \quad \text{repeat 4 [add red top]} \\
\text{l white} & \quad := \quad \text{go left and add white} \\
\text{mov up 2} & \quad := \quad \text{repeat 2 [select up]} \\
\text{go up 3} & \quad := \quad \text{go up 2; go up}
\end{align*}
\]
Modes of naturalization

higher level:

\[
\text{add black block width 2 length 2 height 3}
\]
\[
:= \{\text{repeat 3 [add black platform width 2...}
\]

\[
\text{flower petals}
\]
\[
:= \text{flower petal; back; flower petals}
\]

\[
\text{red cube size 5, add green plate 2 x 4, 5 x 5 open green square, brownbase}
\]
Citations

an induced rule gets a citation whenever it is used in a structure
# Citations

**USER: MT_A37RS | IMPACT: 23 | SCORE: 3599**

- move left 6: 780 pts
- add brown tower 2: 721 pts
- mov up: 577 pts
- mov back: 357 pts
- move down 3: 327 pts
- mov right 3: 291 pts
- front 11: 154 pts
- add 6 brown front: 138 pts
- move up 2: 129 pts
- down: 125 pts

**USER: MT_A7HDY | IMPACT: 22 | SCORE: 2927**

- add b: 1705 pts
- add row brown right 9: 236 pts
- add tower brown 10: 234 pts
- add blu: 172 pts
- add blk: 154 pts
- add brn: 144 pts
- add row brn left 5: 80 pts
- add row brown left 3: 75 pts
- t: 65 pts
- l: 62 pts

**USER: MT_A25AX | IMPACT: 22 | SCORE: 1068**

- add brown col: 243 pts
- red line: 149 pts
- add grass: 116 pts
- add water: 103 pts
- 5 x 5 open green square: 88 pts
- front 12: 86 pts
- short blue line: 76 pts
- open red 3 x 3 square: 72 pts
- add blue wall: 68 pts
- remove line: 67 pts
basic statistics: 1113 cited rules, median 3, mean 46

left 3 : 5820 (self:459)
go up 3 : 712 (self:208)
right : 2879 (self:207)
add brown tower 2 : 721 (self:63)
r white : 175 (self:174)
add red top 4 times : 309
go back and right : 272
select orange : 256
add white plate 6 x 7 : 232
add brown row 3 : 203
mov right 3 : 178
Naturalization via interactive learning

- Community of users can build on each other to naturalize the core language
Naturalization via interactive learning

• community of users can build on each other to naturalize the core language

• crowdworkers are able to provide strong supervision using definitions
Naturalization via interactive learning

- community of users can build on each other to naturalize the core language
- crowdworkers are able to provide strong supervision using definitions
- from core language the users are forced to use, to a language users want to use
Thanks for listening
Questions?
Issues

• predictability and interpretability: 2k+ rules

• after 60k utterances, some simple utterances still not covered

• generalization of higher level concepts

• chair $\rightarrow$ chair with red legs