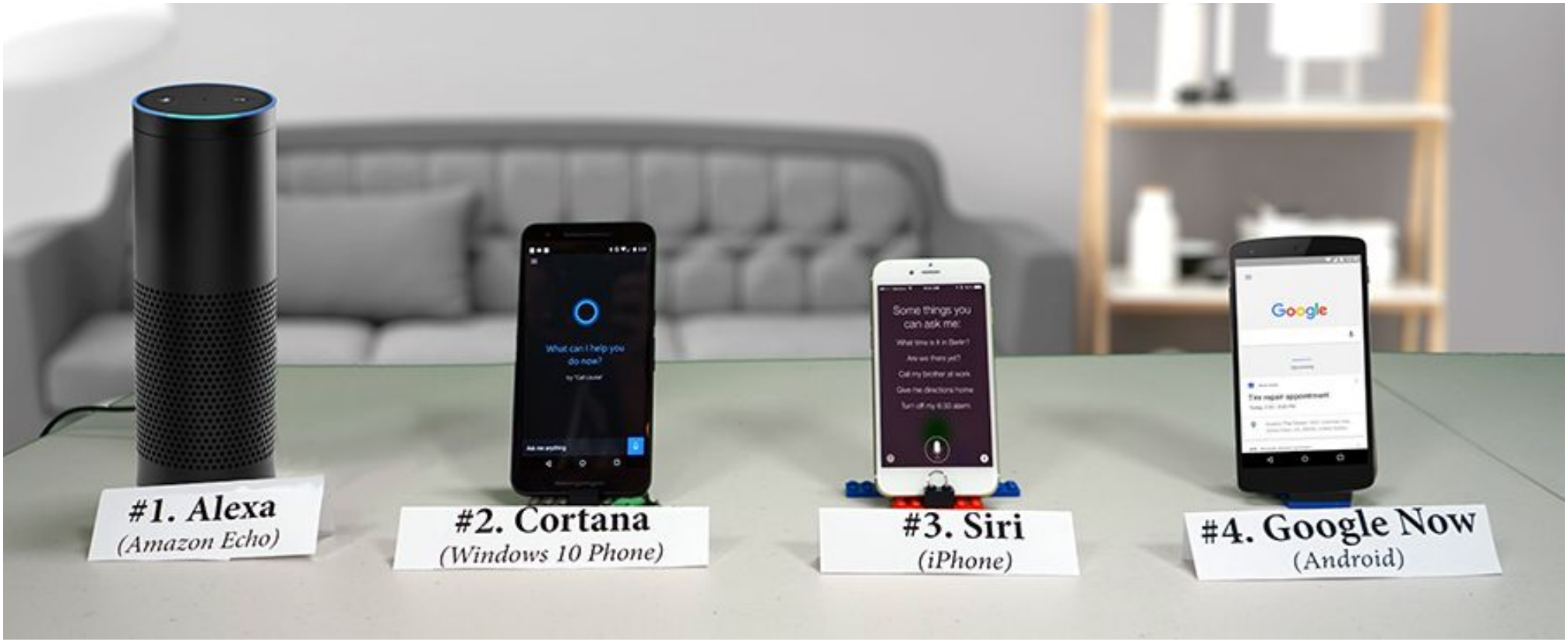
Learning Language Games through Interaction

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

ACL 2016

Natural language interfaces



Natural language interfaces

...

Stephen Colbert: For the love of God, the cameras are on, give me something!

SIRI: What kind of place are you looking for, camera stores or churches?

NLI wishlist

we are stuck when these systems misunderstand us

- **Interactive learning**

  receive feedback from users, improve through use

NLI wishlist

we are stuck when these systems misunderstand us

- **Interactive learning**

  receive feedback from users, improve through use

- **Adapt to users**

  regular weekday alarm, call my children

NLI wishlist

we are stuck when these systems misunderstand us

- **Interactive learning**

  receive feedback from users, improve through use

- **Adapt to users**

  regular weekday alarm, call my children

- **Handle special domains and low resource languages**

  familiar words take on new meaning

  revert to commit 25ad3

  order buy red t5 2

  tayagu-m ada-a

Language game

Wittgenstein. 1953. Philosophical Investigations:

Language derives its meaning from use.

'block' 'pillar' 'slab' 'beam'.

Interactive language game

- Iterated, cooperative game between human and computer

- The human player
  - has a goal, cannot perform actions
  - can use language and provide feedback

- The computer player
  - does not know goal, can perform the actions
  - does not understand language

Interactive language game

- Iterated, cooperative game between human and computer

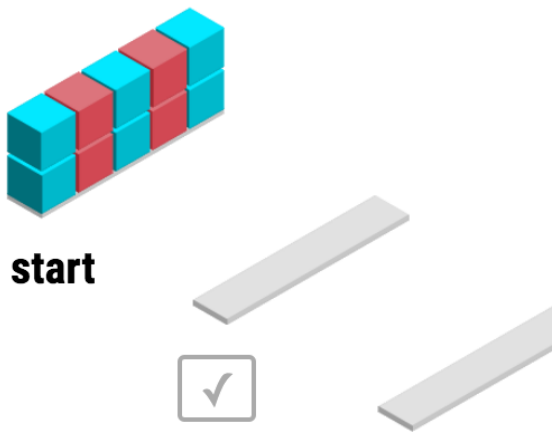
- The human player
  - has a goal, cannot perform actions
  - can use language and provide feedback

must teach the computer a suitable language, and adapt

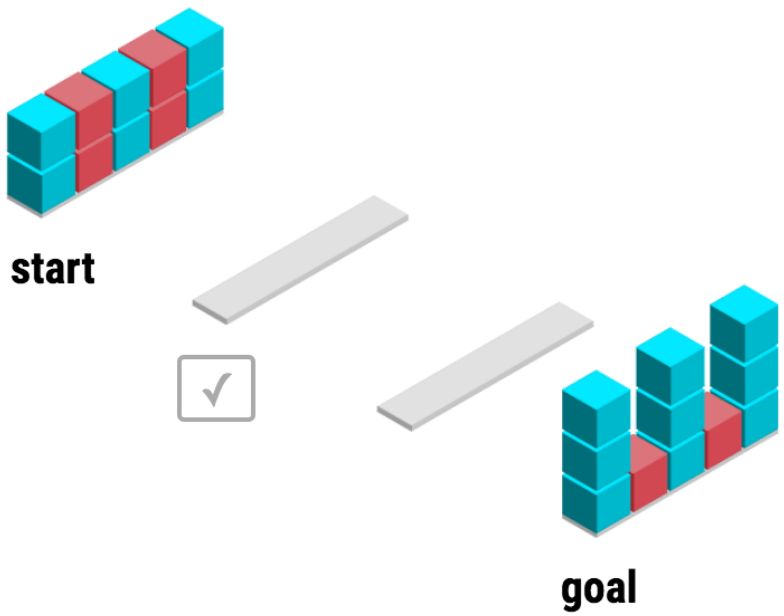
- The computer player
  - does not know goal, can perform the actions
  - does not understand language

must learn language quickly through interaction

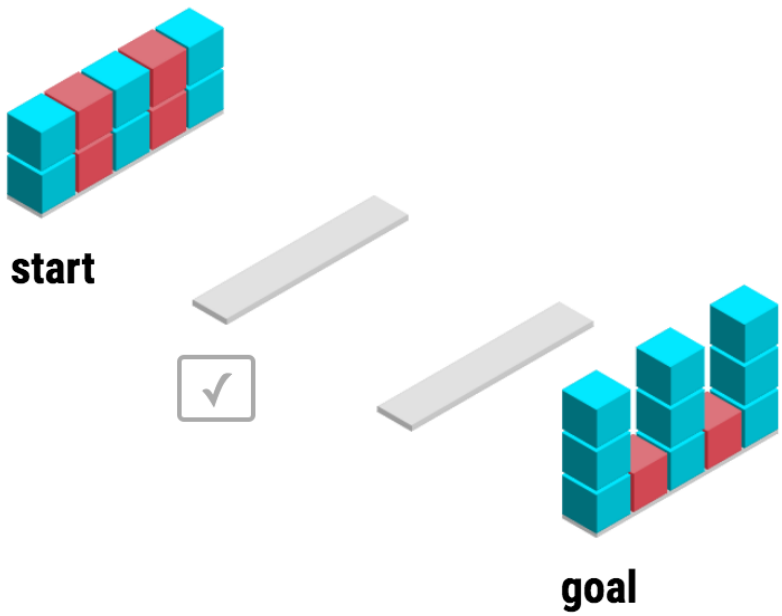
SHRDLURN



SHRDLURN



SHRDLURN



remove red

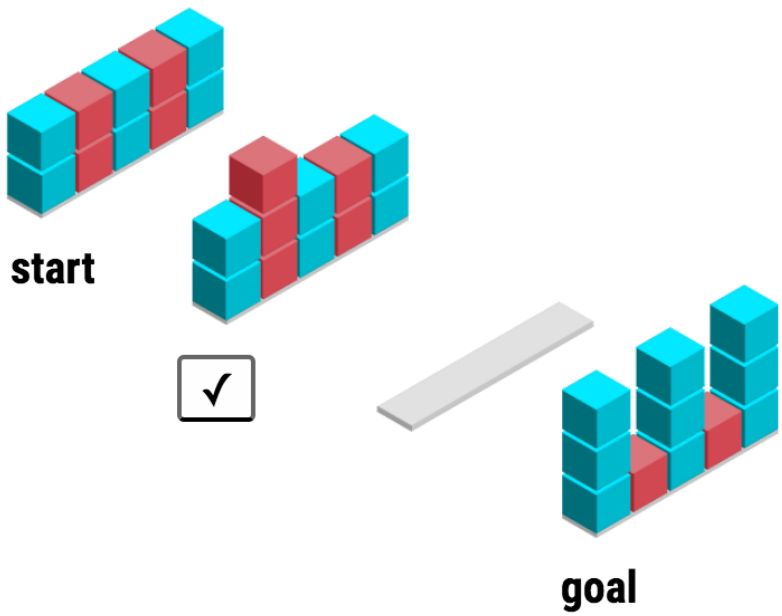
has a goal

has language

performs actions

does not talk

SHRDLURN



remove red

add(leftmost(hascolor(red)),red)

add(red, hascolor(cyan))

remove(hascolor(red))

remove(leftmost(hascolor(red)))

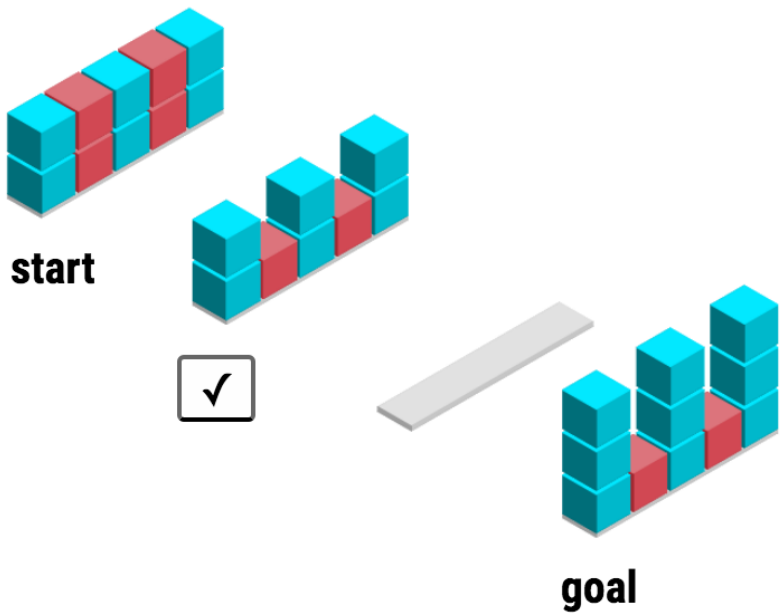
has a goal

has language

performs actions

does not talk

SHRDLURN



remove red

add(leftmost(hascolor(red)),red)

add(red, hascolor(cyan))

remove(hascolor(red))

remove(leftmost(hascolor(red)))

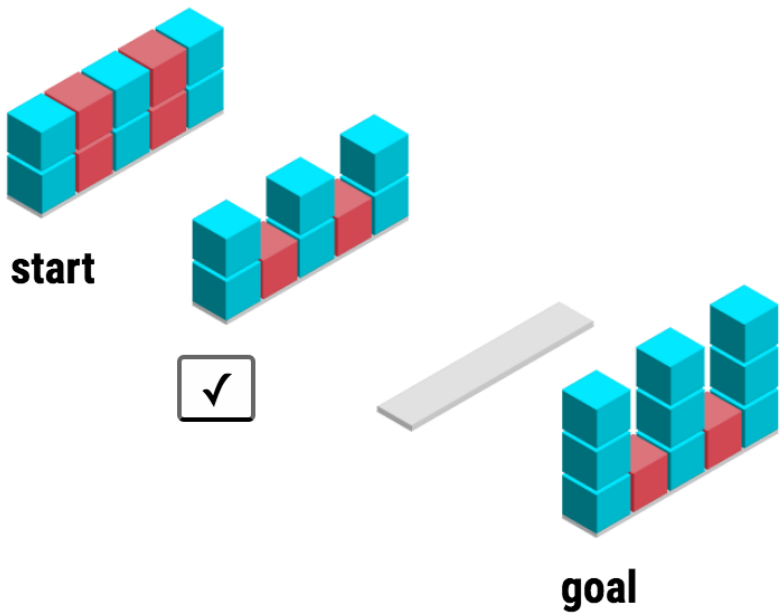
has a goal

has language

performs actions

does not talk

SHRDLURN



remove red

add(leftmost(hascolor(red)),red)

add(red, hascolor(cyan))

remove(hascolor(red))

remove(leftmost(hascolor(red)))

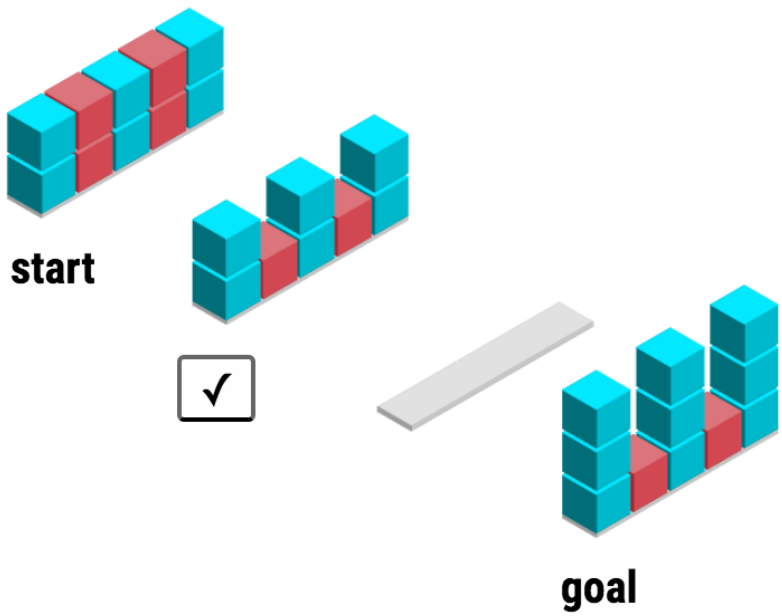
has a goal

has language

performs actions

does not talk

SHRDLURN



把 红的 拿走

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

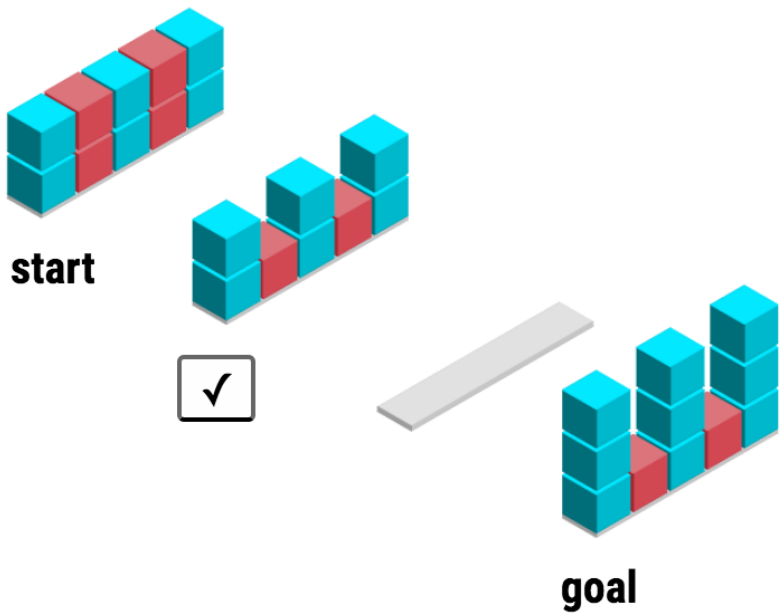
has a goal

has language

performs actions

does not talk

SHRDLURN



emoveray edray

add(leftmost(hascolor(red)),red)

add(red, hascolor(cyan))

remove(hascolor(red))

remove(leftmost(hascolor(red)))

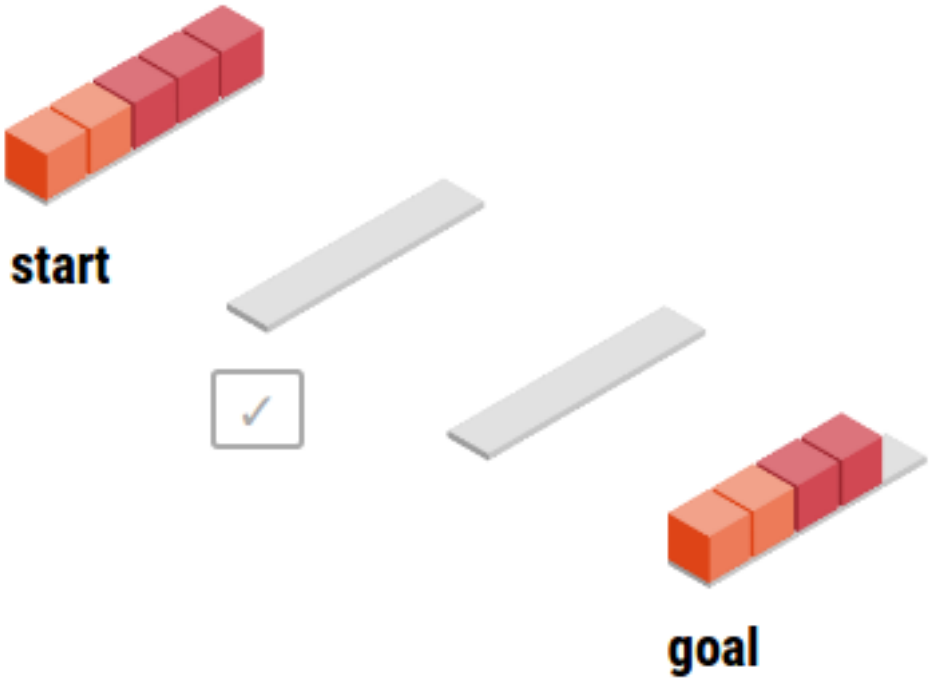
has a goal

has language

performs actions

does not talk

SHRDLURN



enter a command, you did it! solve this puzzle 6 more times to advance.

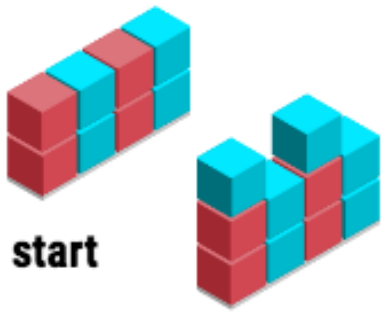
remove right red

Outline

- **Computer: semantic parsing**
- Human: 100 Turkers
- Pragmatics
- Final remarks

Semantic parsing

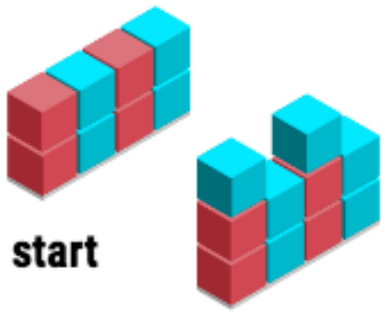
Actions as logical forms:



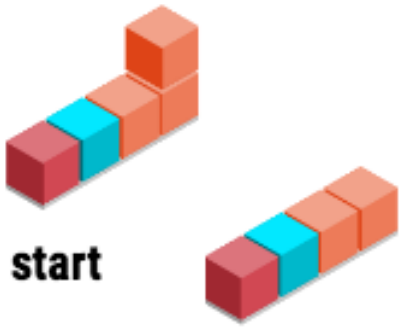
add(hascolor(red), cyan)

Semantic parsing

Actions as logical forms:



add(hascolor(red), cyan)



remove(rightmost(all()))
remove(rightmost(hascolor(orange)))

"Parsing" freely

- Generate logical forms
  - start from the smallest size
  - score them with a model
  - use beam search to find longer high-scoring logical forms
  - like the floating parser [Pasupat and Liang 2015]

brown

hascolor(brown)

leftmost(hascolor(brown))

diff(all(),leftmost(hascolor(brown))

remove(diff(all(),leftmost(hascolor(brown)))

Model

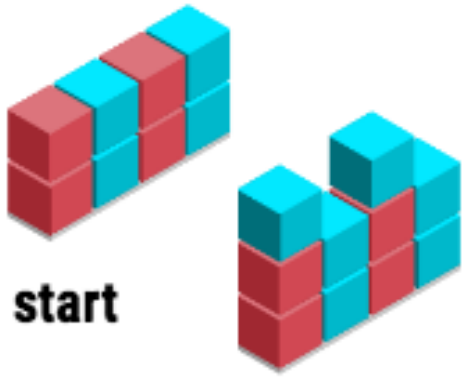
log-linear model with features $\phi(x, z)$:

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

x : *add a cyan block to red blocks*

z : add(hascolor(red), cyan)

y :



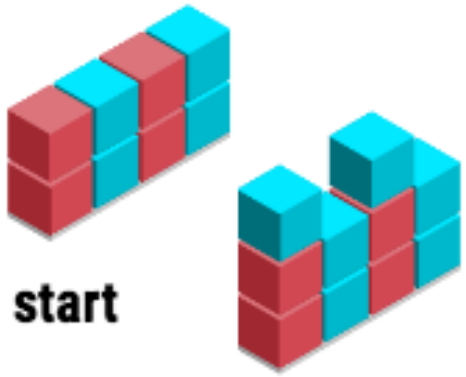
Learning from denotations

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

x : *add a cyan block to red blocks*

z : add(hascolor(red), cyan)

y :



Learning from denotations

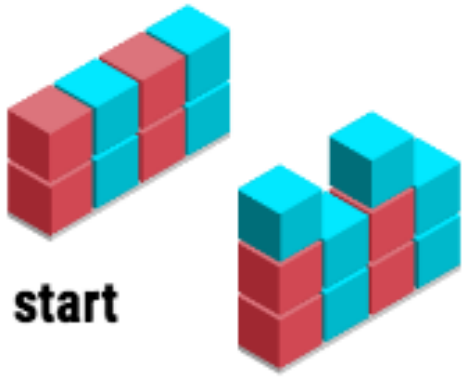
$$p_\theta(z \mid x) \propto \exp(\phi(x, z) \cdot \theta)$$

$$p_\theta(y \mid x) = \sum_{z:\text{Exec}(z)=y} p_\theta(z \mid x)$$

x : *add a cyan block to red blocks*

z : add(hascolor(red), cyan)

y :



Learning from denotations

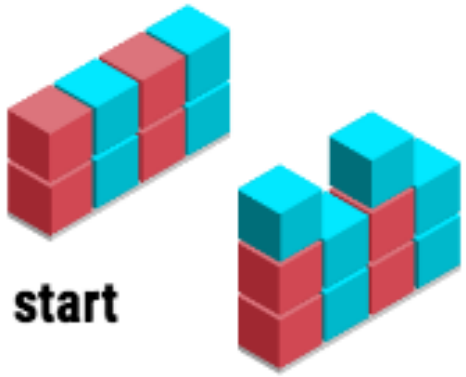
$$p_\theta(z \mid x) \propto \exp(\phi(x, z) \cdot \theta)$$

$$p_\theta(y \mid x) = \sum_{z:\text{Exec}(z)=y} p_\theta(z \mid x)$$

x : *add a cyan block to red blocks*

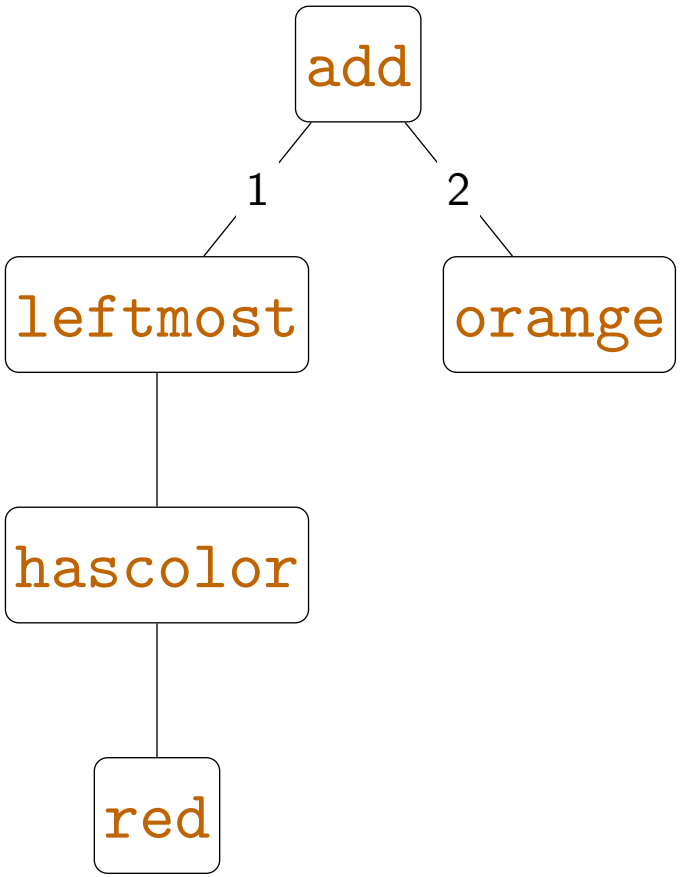
z : add(hascolor(red), cyan)

y :



L1 penalty and update with AdaGrad

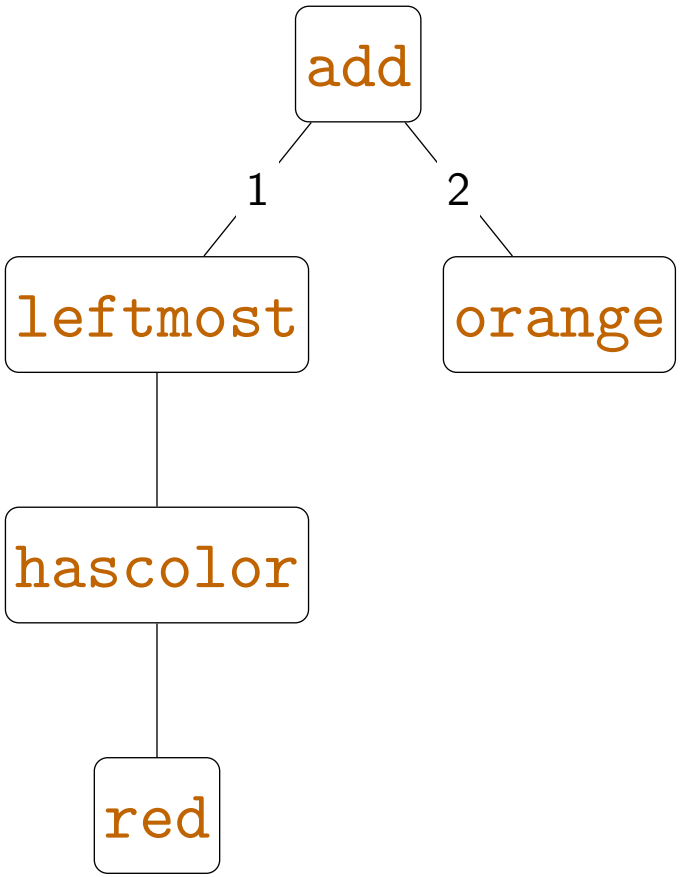
Features



⇕

put orange on the very left red block

Features



uni-, bi-, skip- grams

put, orange, on, the

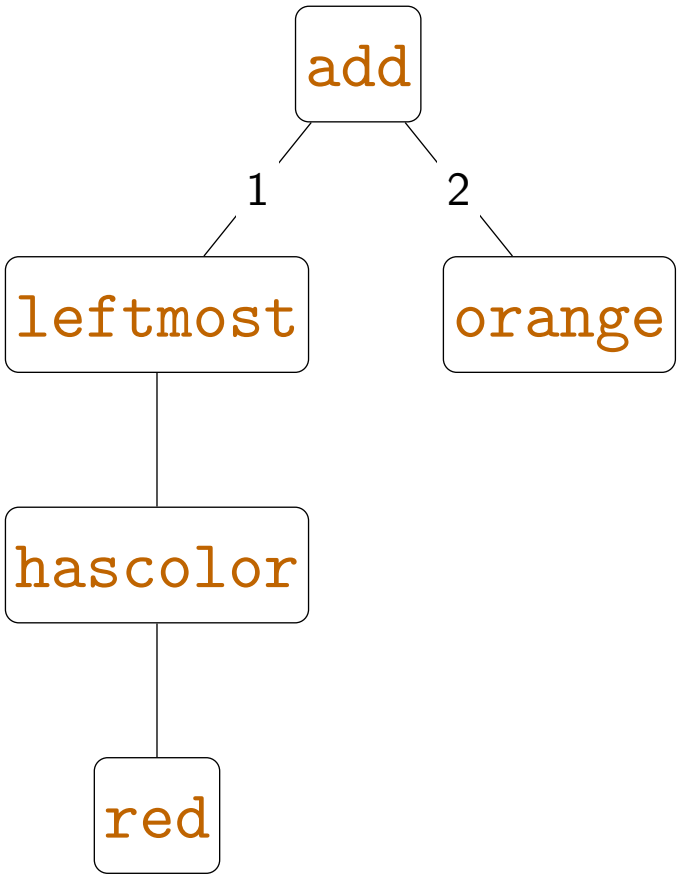
put orange, orange on, ...,

*put * on, orange * the, ...,*

⇕

put orange on the very left red block

Features



⇕

put orange on the very left red block

uni-, bi-, skip- grams

put, orange, on, the

put orange, orange on, ...,

*put * on, orange * the, ...,*

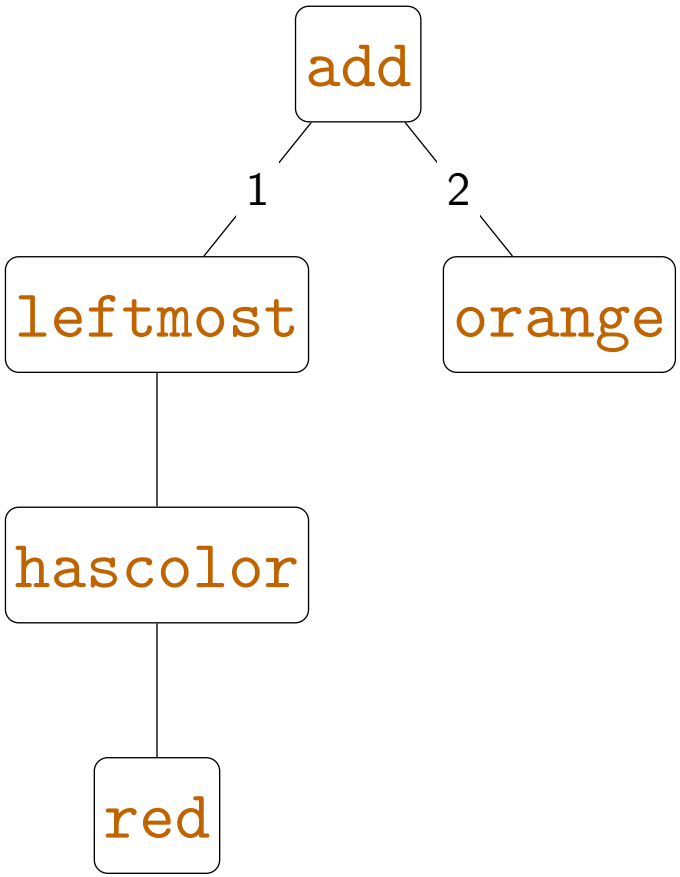
tree-grams

`add(leftmost(*), orange)`

`leftmost(hascolor(*))`

$\lambda c.$`(hascolor(c))`

Features



⇕

put orange on the very left red block

uni-, bi-, skip- grams

put, orange, on, the

put orange, orange on, ...,

*put * on, orange * the, ...,*

tree-grams

`add(leftmost(*), orange)`

`leftmost(hascolor(*))`

$\lambda c.$`(hascolor(c))`

cross product features

(*put*,`add(*,*)`)

(*put orange*,`add(*,orange)`)

(*put*,`orange`)

Outline

- Computer: semantic parsing
- **Human: 100 Turkers**
- Pragmatics
- Final remarks

Experiments

- 100 Turkers played SHRDLURN
  - Got 10223 utterances in total (6 hrs to complete)

Experiments

- 100 Turkers played SHRDLURN
  - Got 10223 utterances in total (6 hrs to complete)
- Minimal instructions
  - no examples provided to avoid bias
  - instructed to use any language

Experiments

- 100 Turkers played SHRDLURN
  - Got 10223 utterances in total (6 hrs to complete)
- Minimal instructions
  - no examples provided to avoid bias
  - instructed to use any language
- Some players liked the game
  - *"That was probably the most fun thing I have ever done on mTurk."*
  - *"This is SO SO cool. I wish there were a way I could better contribute because this research seems to be just insanely interesting and worthwhile."*

Experiments

- 100 Turkers played SHRDLURN
  - Got 10223 utterances in total (6 hrs to complete)
- Minimal instructions
  - no examples provided to avoid bias
  - instructed to use any language
- Some players liked the game
  - *"That was probably the most fun thing I have ever done on mTurk."*
  - *"This is SO SO cool. I wish there were a way I could better contribute because this research seems to be just insanely interesting and worthwhile."*
- performance is measured by the amount of scrolling needed

Results: top players (rank 1-20)

precise and consistent:

 (3.01)

rem cy pos 1
stack or blk pos 4
rem blk pos 2 thru 5
rem blk pos 2 thru 4
stack bn blk pos 1 thru 2
fill bn blk
stack or blk pos 2 thru 6
rem cy blk pos 2 fill rd blk

 (2.72)

Remove the center block
Remove the red block
Remove all red blocks
Remove the first orange block
Put a brown block on the first brown block
Add blue block on first blue block

 (2.78)

remove the brown block
remove all orange blocks
put brown block on orange blocks
put orange blocks on all blocks
put blue block on leftmost blue block in top row

Results: average players (rank 21-50)

inconsistent or mismatches computer capability:

 (9.17)

reinsert pink
take brown
put in pink
remove two pink from second layer
Add two red to second layer in odd intervals
Add five pink to second layer
Remove one blue and one brown from bottom layer

 (7.18)

move second cube
double red with blue
double first red with red
triple second and fourth with orange
add red
remove orange on row two
add blue to column two
add brown on first and third

 (8.37)

remove red
remove 1 red
remove 2 4 orange
add 2 red
add 1 2 3 4 blue
emove 1 3 5 orange
add 2 4 orange
add 2 orange
remove 2 3 brown
add 1 2 3 4 5 red
remove 2 3 4 5 6
remove 2
add 1 2 3 4 6 red

Results: worst players (rank 51-100)

spammy, vague, did not tokenize:

 (12.6)

'add red cubes on center left
center right
far left and far right'
'remove blue blocks on row two column two
row two column four'
remove red blocks in center left and center right on second row

 (14.32)

laugh with me
red blocks with one aqua
aqua red alternate
brown red red orange aqua orange
red brown red brown red brown
space red orange red
second level red space red space red space

 (14.15)

holdleftmost
holdbrown
holdleftmost
blueonblue
brownonblue1
blueonorange
holdblue
holdorange2
blueonred2
holdends1
holdrightend
hold2
orangeonorangerightmost

Results: interesting players

 (Polish)

usuń brązowe klocki
usuń niebieski klocek
usuń pomarańczowe klocki
usuń czerwony klocek
postaw brązowy klocek na pierwszym klocku
postaw czerwony klocek na pierwszym klocku
postaw pomarańczowe klocki na brązowych
postaw czerwone klocki
usuń ostatni brązowy klocek
usuń wszystkie klocki oprócz ostatniego
postaw niebieski klocek na czerwonym
postaw brązowy klocek na pierwszym klocku

 (Polish notation)

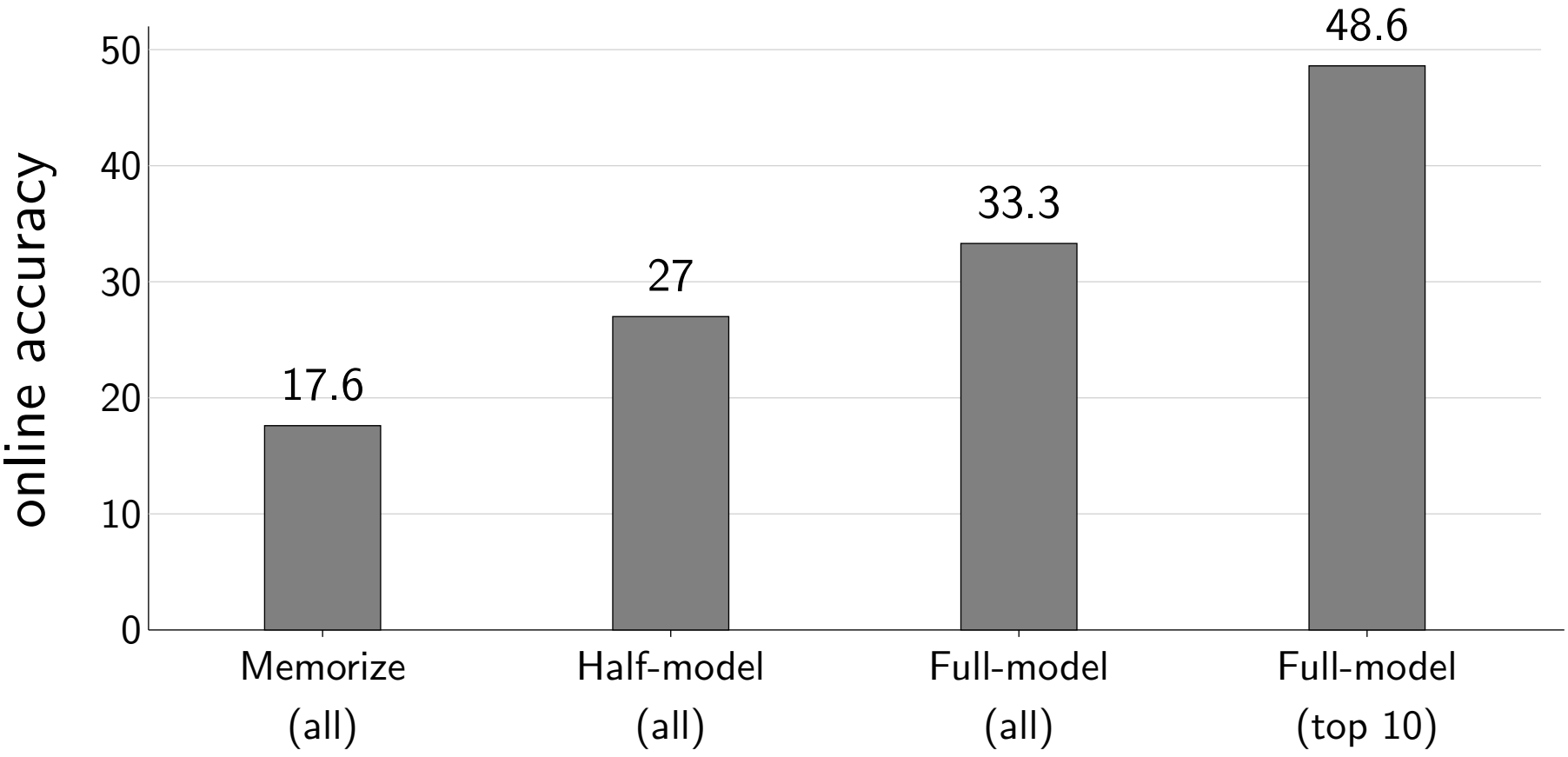
rm scat + 1 c
+ 1 c
rm sh
+ 1 2 4 sh
+ 1 c
- 4 o
rm 1 r
+ 1 3 o
full fill c
rm o
full fill sh
- 1 3
full fill sh
rm sh
rm r
+ 2 3 r
rm o
+ 3 sh
+ 2 3 sh

Players adapt

- More consistent
  - *remove, delete* $\rightarrow$ *remove*

- More concise
  - *Remove the red ones* $\rightarrow$ *Remove red*
  - *add brown on top of red* $\rightarrow$ *add orange on red*
  - *the, a* $\rightarrow \epsilon$

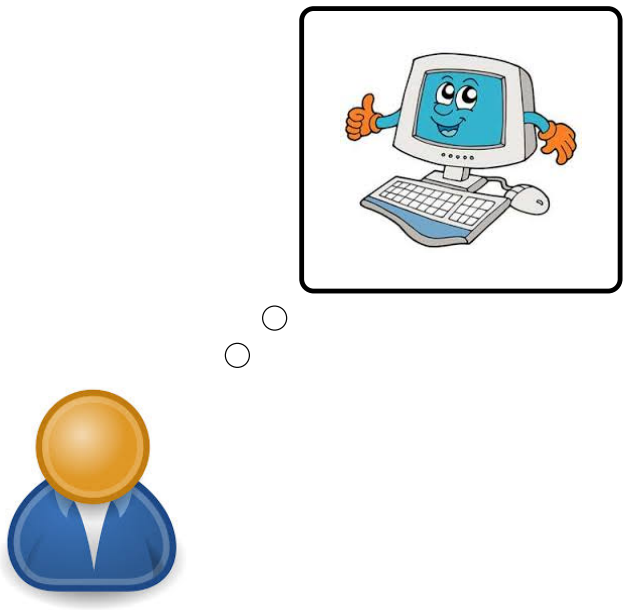
Quantitative results



Learning works fairly well, especially for top players

Outline

- Computer: semantic parsing
- Human: 100 Turkers
- **Pragmatics**
- Final remarks



Pragmatics: motivation

delete cardinal

```
remove(hascolor(red))
```

Pragmatics: motivation

delete cardinal

```
remove(hascolor(red))
```

delete cyan

Pragmatics: motivation

delete cardinal

```
remove(hascolor(red))
```

delete cyan

```
remove(hascolor(red))

remove(hascolor(cyan))

remove(hascolor(brown))
```

Pragmatics: motivation

delete cardinal

```
remove(hascolor(red))
```

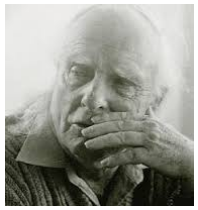
delete cyan

```
remove(hascolor(red))
```

```
remove(hascolor(cyan))
```

```
remove(hascolor(brown))
```

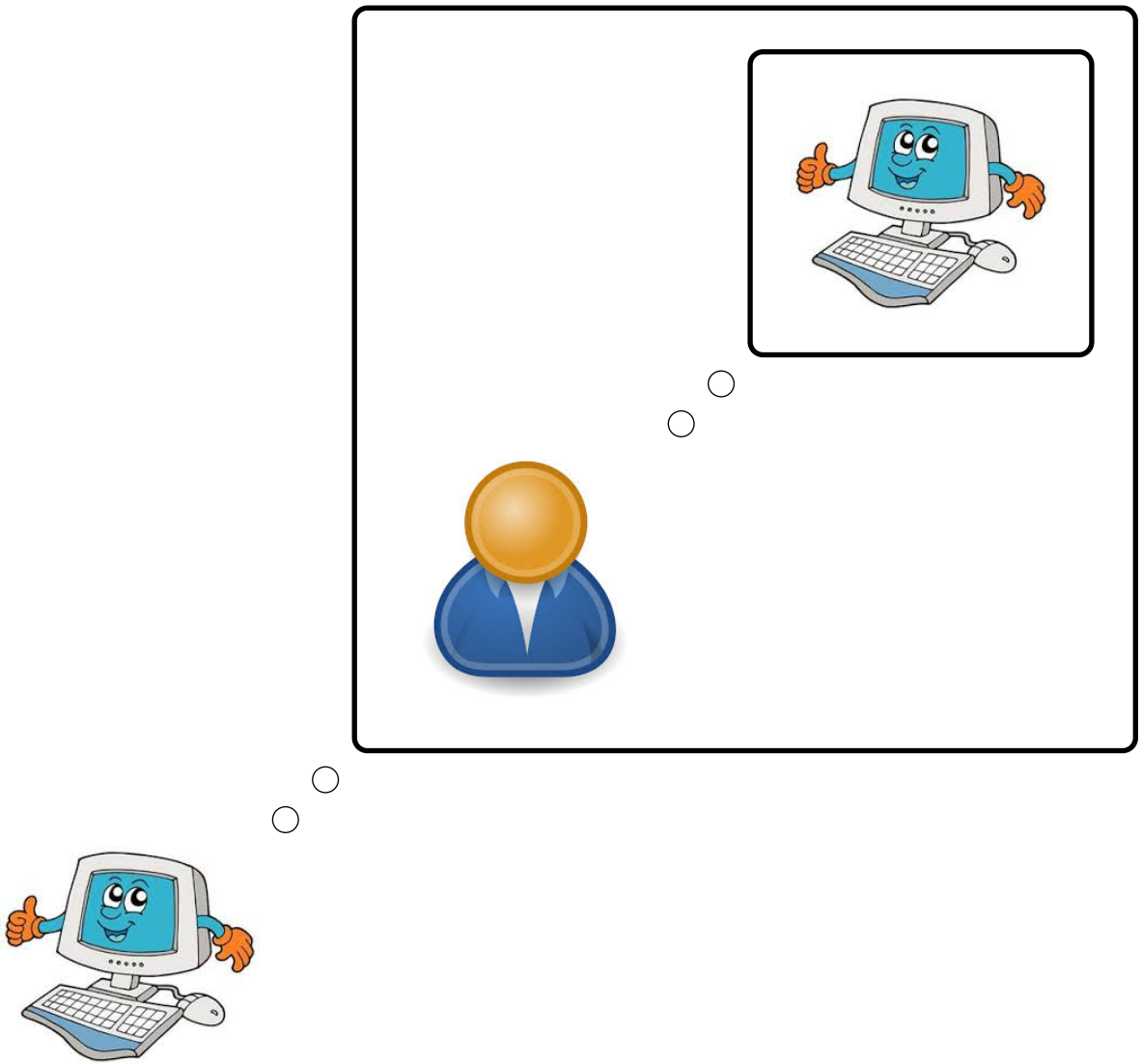
Intuition: cooperative communication



Paul Grice

[Golland et al. 2010; Frank/Goodman, 2012]

Pragmatics: model



Pragmatics: example

Listener (computer):

$p_\theta(z \mid x)$: semantic parsing model

	`remove(red)`	`remove(cyan)`	others
delete cardinal	0.8	0.1	0.1
delete cyan	**0.6**	0.2	0.2

Pragmatics: example

Speaker (human):

$$S(x \mid z) \propto p_\theta(z \mid x)p(x)$$

(assume $p(x)$ uniform)

	`remove(red)`	`remove(cyan)`	others
delete cardinal	0.57	0.33	0.33
delete cyan	0.43	0.67	0.67

Pragmatics: example

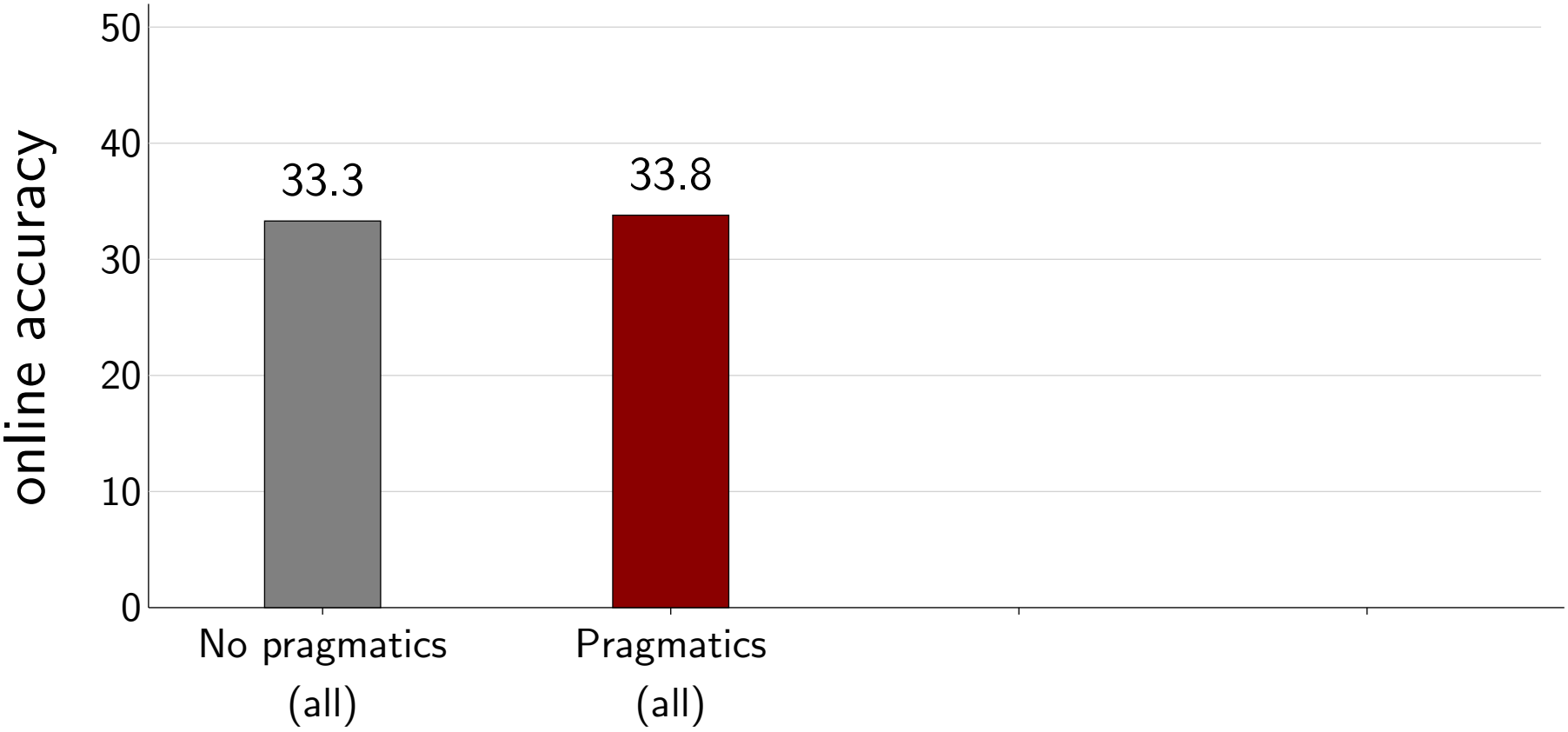
Listener (computer):

$$L(z \mid x) \propto S(x \mid z)p(z)$$

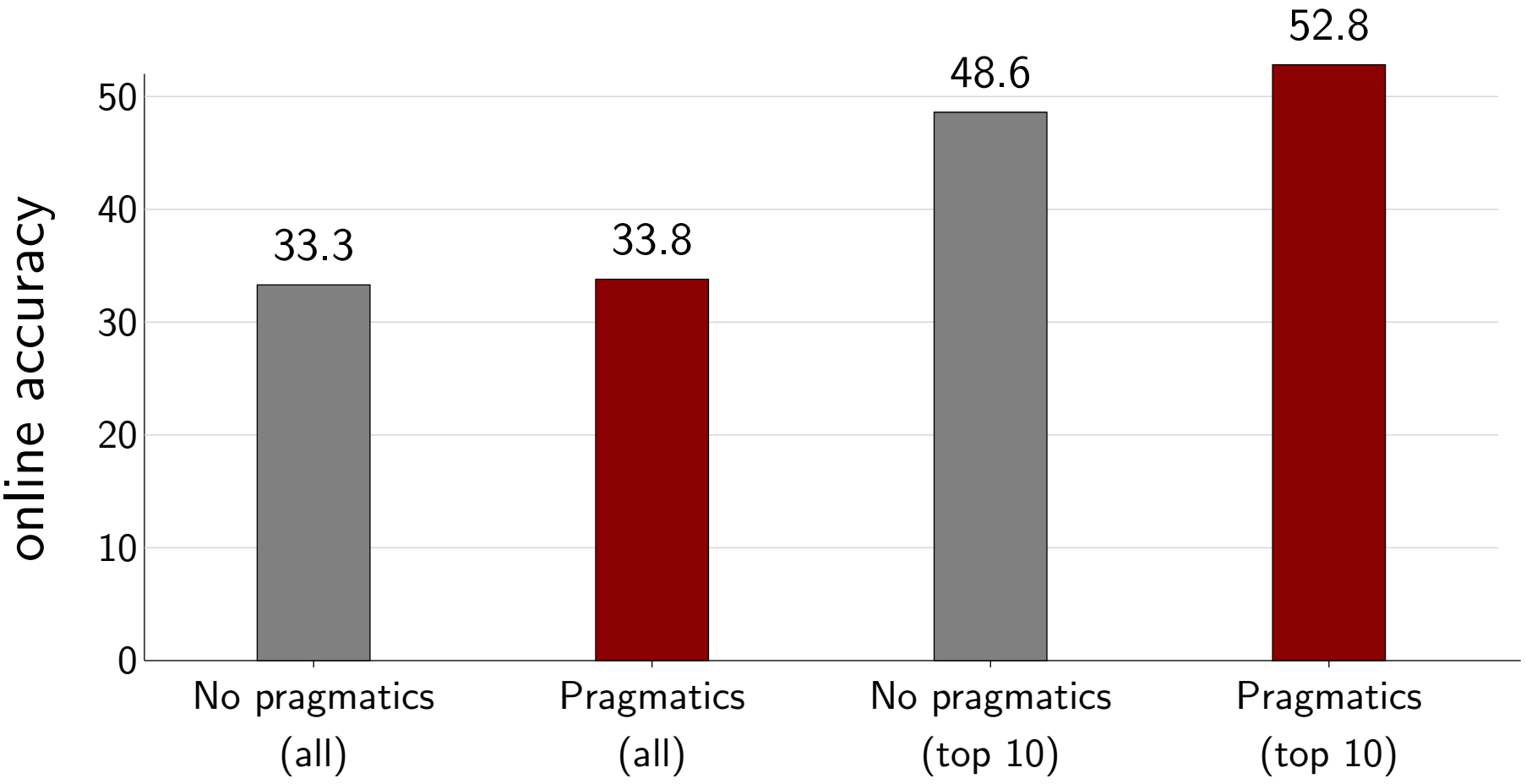
(assume $p(z)$ uniform)

	`remove(red)`	`remove(cyan)`	others
delete cardinal	0.46	0.27	0.27
delete cyan	0.24	**0.38**	**0.38**

Pragmatics: results



Pragmatics: results

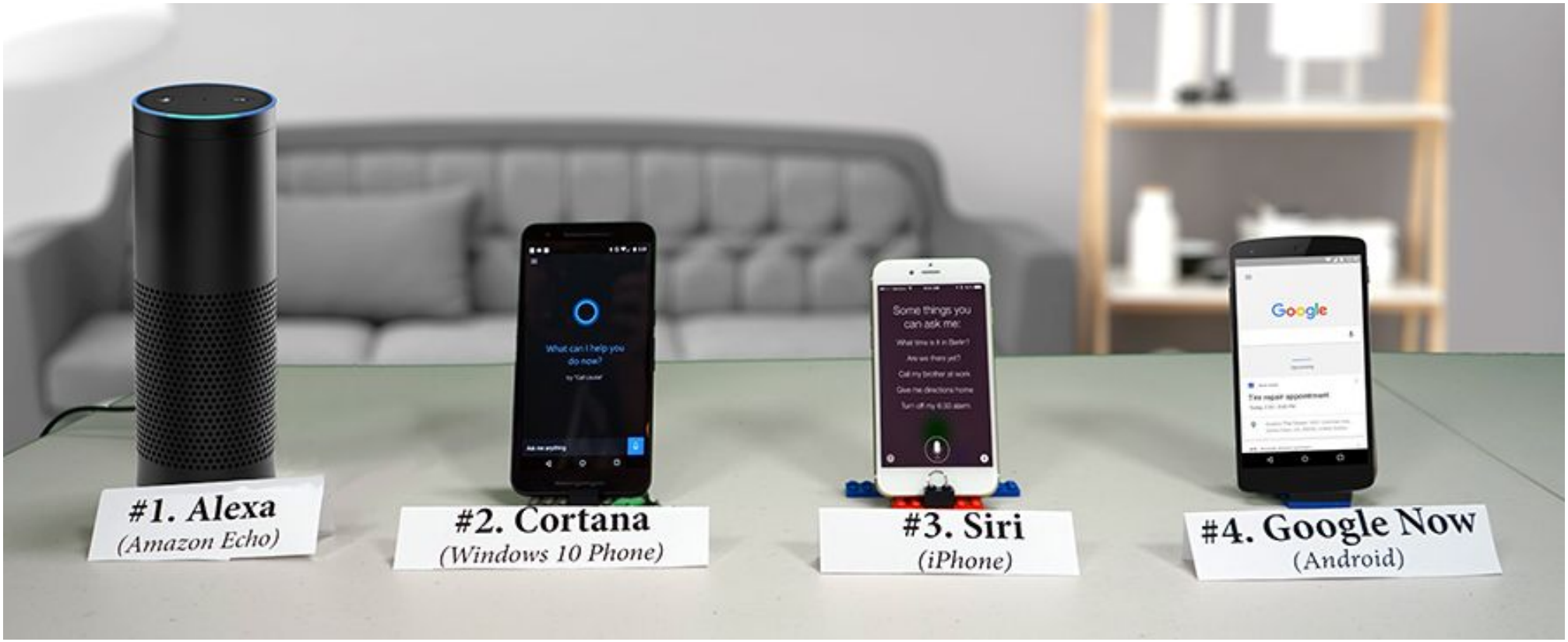


pragmatics helps top (cooperative, rational) players

Outline

- Computer: semantic parsing
- Human: 100 Turkers
- Pragmatics
- **Final remarks**

Natural language interfaces



Remark: better usability

- more capable and responsive to users
  - the computer never understands X
  - user teaches the computer what X means!

- feedback mechanism $\rightarrow$ less likely to be stuck
  - user is stuck, although the system is almost right
  - use the built-in feedback mechanics to move on (i.e. by picking from a list)

Remark: better usability

- more capable and responsive to users
  - the computer never understands X
  - user teaches the computer what X means!

- feedback mechanism $\rightarrow$ less likely to be stuck
  - user is stuck, although the system is almost right
  - use the built-in feedback mechanics to move on (i.e. by picking from a list)

integrate learning with normal usage

Remark: better machine learning

- good for low-resource languages and new domains
  - write rules, collect data, pay for engineering
  - deploy the system and learn from scratch

- learn from the actual distribution!
  - crawl the web, pay crowdworkers
  - deploy the system and get real data

Remark: better machine learning

- good for low-resource languages and new domains
  - write rules, collect data, pay for engineering
  - deploy the system and learn from scratch

- learn from the actual distribution!
  - crawl the web, pay crowdworkers
  - deploy the system and get real data

less distinction between training and testing

The real data

- Data from June 2016 - May 2017
  - 26k+ labeled examples, 1599 games

(NLPers?)

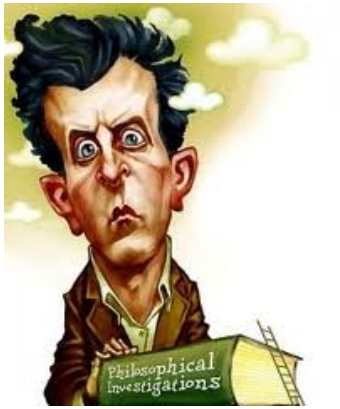
add brown on the top unless the rightmost not(red)
pick up blue blocks
+ 1 2 3 4 5 r
Not the brown block!
The orange block!
છોડો વાદળી 0 1
બધા વાદળી દૂર
છોડો નારંગી 1 4
add blo 1 bro
rem ora blo
add blo 6 pin
add blo 134 bl
去掉最后一个块
在蓝色块上面加一层橙色块
smaz 1 a 3 jednou
retire les blocs bleus

(NLPers?)

move all blocks but middle
- 1 br - 4 br - 6 br
一番奥にオレンジを置く
一番右の赤を消す
add red one on the first
lift 1 3 5
add one orange block on top of each orange
去掉 蓝色 方块
smaz 1 a 2 a 3 a 5
quita el bloque marrón
quita el primer bloque por la derecha
drop orange not left not right
add brown on all blue in line 2 in line 3
Add x x o x o x red block
只保留桔黄色的方块
quitar cubo rojo
quitar ultimo cubo rojo

3

Improve through use!



Wittgenstein: language derives its meaning through use

Code, experiments, demo: shrdlurn.sidaw.xyz

CodaLab