Natural language interfaces

#1. Alexa
(Amazon Echo)

#2. Cortana
(Windows 10 Phone)

#3. Siri
(iPhone)

#4. Google Now
(Android)
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?
we are stuck when these systems misunderstand us

- Interactive learning
  receive feedback from users, improve through use
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

start
has a goal
has language

remove red

performs actions
does not talk
SHRDLURN

start

has a goal

has language

remove red

perform actions

does not talk

add(leftmost(hascolor(red)),red)
add(red, hascolor(cyan))
remove(hascolor(red))
remove(leftmost(hascolor(red)))
has a goal

removes red

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

performs actions

has language
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
has a goal

performs actions

has language
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)))

6
has a goal
has language

emoveray edray

perform actions
does not talk

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

\[ \text{add(hascolor(red), cyan)} \]
Semantic parsing

Actions as logical forms:

```
add(hascolor(red), cyan)
remove(rightmost(all()))
remove(rightmost(hascolor(orange)))
```

```
start
```

```
start
```

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]
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)*

\( y \):

\[ \text{start} \]
Learning from denotations

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

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

\( x: \text{add a cyan block to red blocks} \)
\( z: \text{add(hascolor(red), cyan)} \)

\( y: \text{start} \)
Learning from denotations

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

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

\( x : \text{add a cyan block to red blocks} \)

\( z : \text{add(hascolor(red), cyan)} \)

\( y : \text{L1 penalty and update with AdaGrad} \)
Features

```
<table>
<thead>
<tr>
<th>add</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

leftmost

hascolor

red

\[\Leftrightarrow\]

*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

\[
\begin{align*}
\text{uni-, bi-, skip- grams} & : \text{put, orange, on, the} \\
& \text{put orange, orange on, ...} \\
& \text{put * on, orange * the, ...} \\
\text{tree-grams} & : \text{add(leftmost(*), orange)} \\
& \text{leftmost(hascolor(*))} \\
& \lambda c. (\text{hascolor}(c))
\end{align*}
\]

\text{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, ....
tree-grams
add(leftmost(*), orange)
leftmost(hascolor(*))
\( \lambda c. (\text{hascolor}(c)) \)
cross product features
\((\text{put}, \text{add}(\ast, \ast))\)
\((\text{put orange}, \text{add}(\ast, \text{orange}))\)
\((\text{put}, \text{orange})\)

put orange on the very left red block
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

(8.37)
- remove red
- remove 1 red
- remove 2 4 orange
- add 2 red
- add 1 2 3 4 blue
- remove 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

(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
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 → remove*

- More concise
  - *Remove the red ones → Remove red*
  - *add brown on top of red → add orange on red*
  - *the, a → ε*
Learning works fairly well, especially for top players.
Outline

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

\textit{delete cardinal}

\textit{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

\textit{delete cardinal}

\texttt{remove(hascolor(red))}

\textit{delete cyan}

\texttt{remove(hascolor(red))}
\texttt{remove(hascolor(cyan))}
\texttt{remove(hascolor(brown))}

\textbf{Intuition: cooperative communication}
Pragmatics: model

[Paul Grice]

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

Listener (computer):

\[ p_\theta(z \mid x): \text{semantic parsing model} \]

- **remove(red)**
  - other: 0.8

- **remove(cyan)**
  - other: 0.6

- **delete cardinal**
  - other: 0.1

- **delete cyan**
  - other: 0.2
Pragmatics: example

Speaker (human):

\[ S(x \mid z) \propto p_\theta(z \mid x)p(x) \]

(assume \( p(x) \) uniform)

<table>
<thead>
<tr>
<th>remove(red)</th>
<th>remove(cyan)</th>
<th>others</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{delete cardinal}</td>
<td>0.57</td>
<td>0.33</td>
</tr>
<tr>
<td>\textit{delete cyan}</td>
<td>0.43</td>
<td>0.67</td>
</tr>
</tbody>
</table>
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

![Bar graph showing online accuracy for no pragmatics and pragmatics (all)].

- No pragmatics (all): 33.3%
- Pragmatics (all): 33.8%
Pragmatics helps top (cooperative, rational) players
Outline

- Computer: semantic parsing
- Human: 100 Turkers
- Pragmatics
- Final remarks
Natural language interfaces

#1. Alexa
(Amazon Echo)

#2. Cortana
(Windows 10 Phone)

#3. Siri
(iPhone)

#4. Google Now
(Android)
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!
- 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
Improve through use!

Wittgenstein: language derives its meaning through use

Code, experiments, demo: shrdlurn.sidaw.xyz