Learning language games through interaction

Sida I. Wang, Percy Liang, Christopher D. Manning (ACL'16)

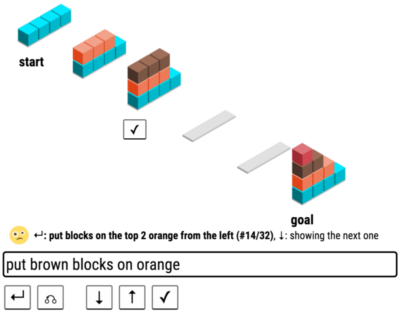
Geelon So, agso@eng.ucsd.edu
CSE291 Continual Learning — April 15, 2021

Introduction

Interactive learning through language games (ILLG) setting:

- two parties need to collaborate to accomplish a goal
- the two parties do not initially speak a common language

SHRDLURN: a collaborative block-building game



Setting

The *objective* of the game is to build the goal state.

- Two players: human and computer
  - only the human can see the goal state
  - only the computer can take actions
- Communication: initially no shared language
  - human provides an utterance to the computer (e.g. types *'put brown blocks on orange'*)
  - computer then proposes list of possible next states that human can select from

Learning problem

- Over multiple rounds of the game, can they achieve *more complex* goals *more efficiently*?

Formal setting

- $\mathcal{Y}$ a set of **states** (stacks of colored block in a line)
- $\mathcal{Z}$ a set of **actions** described by a grammar

Rule	**Semantics**	**Description**
Set	`all()`	all stacks
Color	`cyan|brown|red|orange`	primitive color
Color $\rightarrow$ Set	`with`(c)	stacks whose top block has color c
Set $\rightarrow$ Set	`not`(s)	all stacks except those in s
Set $\rightarrow$ Set	`leftmost|rightmost`(s)	leftmost/rightmost stack in s
Set Color $\rightarrow$ Act	`add`(s, c)	add block with color c on each stack in s
Set $\rightarrow$ Act	`remove`(s)	remove the topmost block of each stack in s

Table 1: The formal grammar defining the compositional action space $\mathcal{Z}$ for SHRDLURN. We use c to denote a Color, and s to denote a Set. For example, one action that we have in SHRDLURN is: *'add an orange block to all but the leftmost brown block'* $\mapsto$ `add(not(leftmost(with(brown))),orange)`.

  - well-formed statements correspond to actions

Formal setting (cont.)

A single round of interaction

1. current state of the game is $s \in \mathcal{Y}$, target state is $t \in \mathcal{Y}$
2. human provides an utterance x to the computer
3. computer maps x to a ranked list $(z_1, \ldots, z_K)$ over logical forms $z \in \mathcal{Z}$
   - applies semantic parsing model to keep top K candidates ($K = 100$)
4. computer executes z_i on s and proposes states $(y_1, \ldots, y_K)$ to human
5. human selects desired state y_i to execute (i.e. state of game advances $s \leftarrow y_i$)
   - computer uses this indirect feedback to update semantic parsing model

Formal setting (cont.)

Language learning

- computer must learn the chosen language from scratch
- computer applies a semantic parsing model, a **log-linear model**,

$$p_\theta(z \mid x) \propto \exp\left(\theta^\top \phi(x, z)\right),$$

  where $\phi(x, z) \in \mathbb{R}^d$ is a feature vector and $\theta \in \mathbb{R}^d$ a parameter vector
  - e.g. $\phi(x, z)$ is a count over features
- model leads to the loss function (take single step of gradient descent per round)

$$\ell(\theta, x, y) = -\log p_\theta(y \mid x, s) + \lambda \|\theta\|_1$$
$$p_\theta(y \mid x, s) = \sum_{z : [\![z]\!]_s = y} p_\theta(z \mid x)$$

  where $[\![z]\!]_s$ is the state generated by executing z on state s

Formal setting (cont.)

The feature map $\phi(x, z) \in \mathbb{R}^d$ counts the underlying features of (x, z).

- the **set of features** is the cross product of all *utterance* and *logical form* features
  - **utterance features:** the features of an utterance x are its n-grams (with skip-grams)
    e.g. features of *'stack red on orange'* include:

    unigrams: (*'stack'*, $*$, $*$) bigrams: (*'red'*, *'on'*, $*$)

    trigrams: (*'stack'*, *'red'*, *'on'*) skip trigrams: (*'red'*, $*$, *'orange'*)

  - **logical form features:** the features of a logical form h are its tree-gram features
    e.g. features of `remove(all())` include:

    $(\texttt{remove}) \quad (\texttt{all}) \quad (\texttt{remove}, 1, \texttt{all})$

- example features of (*'enlever tout'*, $\texttt{remove}(\texttt{all}())$) include:

  (*'enlever'*, $\texttt{all}$) (*'enlever'*, $\texttt{remove}$) (*'enlever'*, $(\texttt{remove}, 1, \texttt{all})$)

Formal setting (cont.)

Game setting

- human and computer proceed through levels of the game
  - goal states become increasingly complex over levels
- at each round of interaction, human scrolls through potentially $K = 100$ candidates
  - not guaranteed that desired state contained in top K
- good game performance corresponds to low number of scrolls

Game design principle

> “We expect that in the beginning, the computer does not understand what the human is saying. As the computer learns, the two should become more proficient at communicating and playing the game. *Language learning should be necessary for the players to achieve good game performance.*
>
> *Wang et al. (2016)*

Experiment

Setting:

- 100 workers on Amazon Mechanical Turk played the game
- Game consisted of 5 difficulty levels, each with 10 tasks
- Minimal instructions given to players (no example utterances given) to avoid biasing language use

Metrics

- Number of scrolls: the ranking of selected action in the proposed list

Example utterances + player average number of scrolls

	Most successful players (1st–20th)	
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 **(3.01)**	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 **(2.78)**	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 **(2.72)**
	Average players (21th–50th)	
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 **(9.17)**	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 **(8.37)**	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 **(7.18)**
	Least successful players (51th–)	
holdleftmost, holdbrown, holdleftmost, blueonblue, brownonblue1, blueonorange, holdblue, holdorange2, blueonred2 , holdends1, holdrightend, hold2, orangeonorangerightmost **(14.15)**	‘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 **(12.6)**	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 **(14.32)**
	Spam players (∼ 85th–100)	
next, hello happy, how are you, move, gold, build goal blocks, 23,house, gabboli, x, run,,xav, d, j, xcv, dulicate goal (21.7)		
	Most interesting	
usuń brązowe klocki, postaw pomarańczowy klocek na pierwszym klocku, postaw czerwone klocki na pomarańczowych, usuń pomarańczowe klocki w górnym rzędzie	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, rm b, - 1 o, + 2 c,	mBROWN,mBLUE,mORANGE RED+ORANGE^ORANGE, BROWN+BROWNm1+BROWNm3, ORANGE +BROWN +ORANGE^m1+ ORANGE^m3 + BROWN^^2 + BROWN^^4

Human performance

- 22 out of 100 players failed to teach a language (e.g. often typed random phrases)
  - game can be completed by scrolling; they averaged 21.6 scrolls
- remaining players required on average of 7.4 scrolls
- human adaptation: most tended to become more precise and concise
  - *'remove the red ones'* $\rightarrow$ *'remove red'*
  - *'add brown on top of red'* $\rightarrow$ *'add orange on red'*
  - one player used *'the'* in all of the first 20 utterances, and never used it last 75

Computer performance

Define **online accuracy** as *the fraction that the accepted action is ranked highest*,

$$\text{accuracy} = \frac{1}{T} \sum_{j=1}^{T} \mathbf{1}\{ y^{(j)} = [\![z^{(j)}]\!]_{s^{(j)}} \},$$

where $z^{(j)} = \arg\max_z \, p_{\theta^{(j-1)}}(z \mid x^{(j)})$.

- Can use this metric to evaluate the importance of **compositionality**. Compare with two other models:
  - **memorization:** model memorizes pairs of utterances and logical forms, e.g.

    $$(\textit{`remove all red blocks'}, z_{\texttt{rm-red}})$$

  - **half-compositional:** only utterances are treated as compositional, e.g.

    $$(\textit{`remove'}, z_{\texttt{rm-red}}) \qquad (\textit{`red'}, z_{\texttt{rm-red}})$$

Computer performance (cont.)

	players ranked by # of scrolls			
Method	top 10	top 20	top 50	all 100
memorize	25.4	24.5	22.5	17.6
half model	38.7	38.4	36.0	27.0
half + prag	43.7	42.7	39.7	29.4
full model	48.6	47.8	44.9	33.3
full + prag	52.8	49.8	45.8	33.8

Figure 1: Online accuracy scores (%) to compare (i) non-compositional model, (ii) half-compositional model, and (iii) fully compositional model.

Modeling pragmatics

Phenomenon: when humans learn language, humans tend to use *mutual exclusivity*.

- If I tell you *'rem x'* means `remove(red)`,
  what is *'rem y'* more likely to mean?
  - `remove(red)`
  - `remove(cyan)`

Pragmatics: incorporate this assumption by modeling language as a *cooperative game*.

- Let $S(x \mid z)$ model the speaker's strategy
- Let $L(z \mid x)$ model the listener's strategy

Suppose that the speaker knows the semantic parsing model. Let p be a prior. Set:

$$S(x \mid z) \propto \big(p_\theta(z \mid x)p(x)\big)^\beta \qquad (\beta \geq 1)$$
$$L(z \mid x) \propto S(x \mid z)p(z).$$

Pragmatics example

	$z_{\texttt{rm-red}}$	$z_{\texttt{rm-cyan}}$	$z_3, z_4, \ldots$
		$p_\theta(z \mid x)$	
'remove red'	0.8	0.1	0.1
'remove cyan'	**0.6**	0.2	0.2
		$S(x \mid z)$	
'remove red'	0.57	0.33	0.33
'remove cyan'	0.43	0.67	0.67
		$L(z \mid x)$	
'remove red'	0.46	0.27	0.27
'remove cyan'	0.24	**0.38**	**0.38**

Figure 2: Suppose the computer saw an example *'remove red'* $\mapsto z_{\texttt{rm-red}}$. The literal listener (top) mistakenly chooses $z_{\texttt{rm-red}}$ for *'remove cyan'*. The pragmatic listener (bottom) does not.

Summary

- **Grounded language**: language used in an environment as a means toward a goal.
- Model learns language **completely from scratch**.
- Training is completely **online**, taking a single pass over data.
- Both human and computer **mutually adapt** to each other.
- Here, **compositionality** appears to help.

Discussion

Learning to communicate

What learning took place in SHRDLURN?

- a language is constructed through interaction
- the computer learned to understand the language
- the human learned to speak the language

In other words, the **ability to communicate** was learned.

- learning not the property of the individual, but the system
  - 'learning as a social process'

A continual learning problem

Setting: multiple parties must collaborate to solve stream of varying tasks

- collaboration/communication is required for success
- how should a language be formed? how can it evolve over time?

Language construction and learning

- **A more general setting:** game that simultaneously requires

  $$\textit{learning to communicate} + \textit{learning to perform task.}$$

- **Question:** does being forced to communicate also facilitate task learning?
  - it may force a player to construct a low-precision representation of data
  - language may be a form of memory for the continual learning setting
  - ‘teaching is one of the best way to learn’

Mutual adaptation

SHRDLURN is a setting where the **human also learns and adapts** to the computer.

- It is hard for the human to generate huge amounts of annotation.
- It is easy for the human to learn the inductive biases of the computer.

Question: can we study this setting more formally?

References

Sida I. Wang, Percy Liang, and Christopher D. Manning. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2368–2378, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1224. URL `https://www.aclweb.org/anthology/P16-1224`.