Grounded Semantics

Berkeley

N L P

Jacob Andreas

What does the world look like?

HAL'

$open'$

$\wedge$

$close'$

$Bowman'$

$podBayDoors'$

$\exists$

Today's plan

1. How do we relate language to a richer representation of the world?

2. How do we learn meanings without annotated logical forms?

Today's plan

Open the pod bay doors, HAL

↓

$$\mathrm{open}(\mathrm{HAL}, \mathrm{podBayDoors})$$

↓

Today's plan

Grounded

~~Formal~~ semantics:

How do we learn the relationship between text and ~~logical forms~~?

the world

Three approaches

1. Learning with hardcoded predicates

2. Jointly learning parsers and classifiers

3. Learning a policy directly

Hard-coded predicates

Don't forget:

the λ-calculus is a programming language!

```
final Entity HAL = ...
Entity podBayDoors = ...
void open(Entity opener, Entity opened) {
   ...
}
```

Hard-coded predicates

Given full supervision we can immediately execute output from our semantic parser.

```
final Entity HAL = ...
Entity podBayDoors = ...
void open(Entity opener, Entity opened) {
   ...
}
```

Hard-coded predicates

Open the pod bay doors, HAL

↓

`open(HAL, podBayDoors)`

↓

Distant supervision

Can we use the ability to execute predicted parses to learn with weaker supervision?

Distant supervision

Can we use the ability to execute predicted parses to learn with weaker supervision?

Before:

Open the pod bay doors	observe text
`close(HAL, podBayDoors)`	predict LF
`open(HAL, podBayDoors)`	observe true LF
1.0	incur loss

Distant supervision

Can we use the ability to execute predicted parses to learn with weaker supervision?

Before:

Open the pod bay doors	observe text
`open(HAL, podBayDoors)`	predict LF
`open(HAL, podBayDoors)`	observe true LF
0.0	incur loss

Distant supervision

Can we use the ability to execute predicted parses to learn with weaker supervision?

Now:

Open the pod bay doors	observe text
`close(HAL, podBayDoors)`	predict LF
`doorsClosed = true`	predicted outcome
`doorsClosed = false`	desired outcome
1.0	incur loss

Distant supervision

Recall our previous training procedure.

Structured perceptron update:

$$\theta^{t+1} = \theta^t + \Phi(x, y) - \Phi(x, \hat{y})$$

where

$$\hat{y} = \arg\max_y \ \theta^\top \Phi(x, y)$$

Distant supervision

Now only supervision is an outcome z.

Structured perceptron update:

$$\theta^{t+1} = \theta^t + \Phi(x, \boxed{y^*}) - \Phi(x, \hat{y})$$

where

$$\hat{y} = \arg\max_{y} \ \theta^\top \Phi(x, y)$$

$$y^* = \arg\max_{y:\boxed{\mathrm{exec}(y)=z}} \ \theta^\top \Phi(x, y)$$

Distant supervision

close(HAL, podBayDoors) $\hat{y}$

open(HAL, podBayDoors) y^*

open(HAL, cockpitDoors)

make(HAL, sandwich, Dave)

…

smash(HAL, podBayDoors, filingCabinet) y^*

Distant supervision

Open the pod bay doors, HAL

↓

`open(HAL, podBayDoors)`

↑

What can we do with this?

Learn to answer questions given only (question, answer) pairs and a database of facts

[Liang et al. 2011 & various others]

Learn to follow directions given only (source, pairs) and a model environment

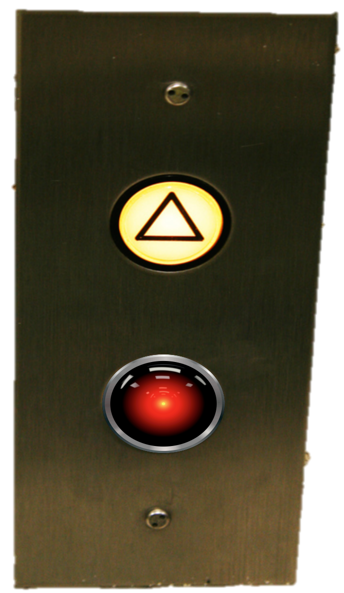
[Chen & Mooney 2011, Artzi & Zettlemoyer 2013]

Joint parsing and perception

What if the world doesn't look like a database underneath?

Open the elevator doors, HAL

What's a door?



Joint parsing and perception

What's a door?

$$f(podBayDoors) = \text{true}$$

door	podBayDoor, elevatorDoor1, cockpitDoor, …
window	bridgeWindow, bathroomWindow, …
isOpen	podBayDoor, bathroomWindow

Joint parsing and perception

What's a door?

$f\Bigg($  $\Bigg) = \text{true}$

Joint parsing and perception

What's a door?

$f\Big($  $\Big) = \text{true}$

Joint parsing and perception

What's a door?

$$f\left(\right) = \text{false}$$

Joint parsing and perception

Fixed inventory of functions

Joint parsing and perception

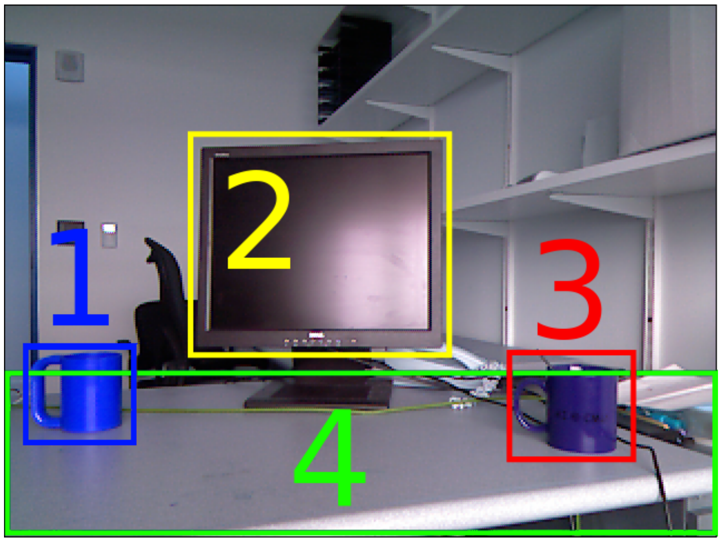
~~Fixed inventory of functions~~

One function per word

door door': Image $\mapsto$ Boolean

in in': (Image, Image) $\mapsto$ Boolean

Joint parsing and perception



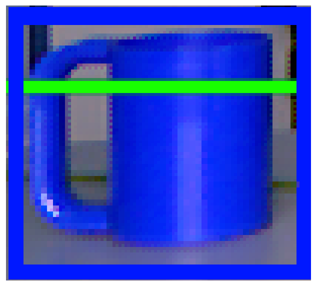
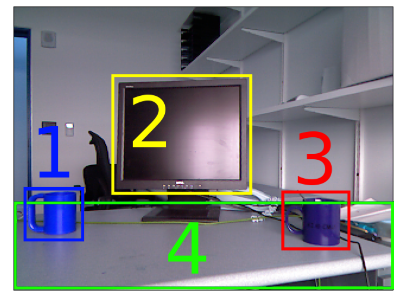
blue	1
mug	1, 3
on	(1,4), (2,4), (3,4)
table	4

blue mug on the table

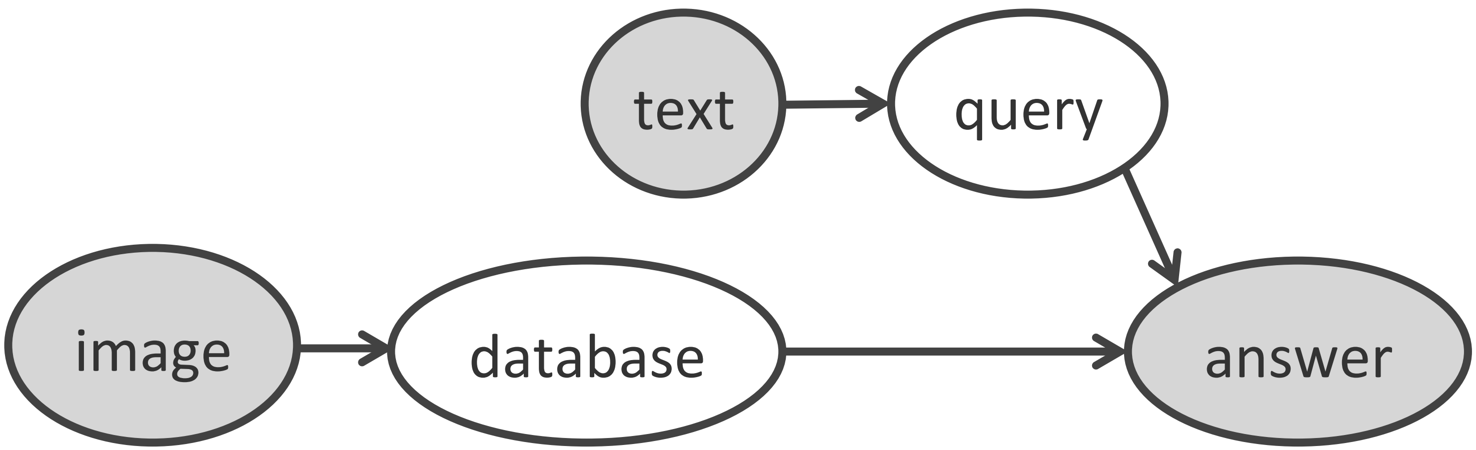
$$\lambda x.\exists y.blue(x) \wedge table(y) \wedge on(x, y)$$

[Krishnamurthy et al. 2013]

Joint parsing and perception

$p\Bigg($  $\Bigg|$ blue mug on the table , 

$\Bigg)$



Joint parsing and perception

Can even learn to compose these grounding functions:

a blue eye

a dark blue eye

a dark pastel blue eye

[Andreas et al. 2013]

The picture so far

Open the pod bay doors	observe text
`close(HAL, podBayDoors)`	predict LF
`doorsClosed = true`	predicted outcome
`doorsClosed = false`	desired outcome
1.0	incur loss

The picture so far

Open the pod bay doors	observe text
`doorsClosed = true`	predicted outcome
`doorsClosed = false`	desired outcome
1.0	incur loss

Learning a conditional policy

~~Learn an intermediate meaning representation~~

$$\cancel{p(\mathrm{result}|\mathrm{text}) = \sum_{\mathrm{MR}} p(\mathrm{result}|\mathrm{MR})\; p(\mathrm{MR}|\mathrm{text})}$$

Learn $p(\mathrm{result}|\mathrm{text})$ directly

MDP refresher

- Set S of states
- Set A of actions
- Transition function $T : (S \times A) \rightarrow S$
- Reward function $R : (S \times A) \rightarrow \mathbb{R}$

Lots of algorithms for *learning* a policy
$\pi : S \rightarrow A$ given only black-box interaction

Reading as an MDP

Idea: augment base MDP state space with position in document.

Open the pod bay doors after making me a sandwich

```
{sandwich=true, doorOpen=true},
{sandwich=true, doorOpen=false},
...
```

{ sandwich=true, doorOpen=false
text=Open the pod bay doors ~~after making me a sandwich~~ }

Reading as an MDP

Now just want to pick

$f\left(\begin{array}{l}\texttt{sandwich=true, doorOpen=false} \\ \texttt{text=}\underline{\text{Open the pod bay doors }}\sout{\text{after making me a sandwich}}\end{array}\right) \in$

$$\{a_1, a_2, \ldots\}$$

maximizing reward.

Use your favorite policy learning technique!

[Vogel & Jurafsky 2010, Branavan et al. numerously]

Reading as an MDP

We get pragmatics for free: easy to learn that

Open the pod bay doors

I want you to open the pod bay doors

I'm ready to come inside now

prefer destination states with {doorOpen = true}

Reading as an MDP

But less clear how to handle composition (syntactic or semantic) in this framework:

Open the red door located between two small doors.

Need some way of handling structured action spaces that don't correspond to syntax.

What else is hard?

Event compositionality and coreference:

1. Before disassembling your iPhone, be sure it is powered off
2. Remove the two 3.6mm Pentalobe or Phillips #000 screws next to the dock connector
   …
27. Use the clear plastic pull tab to gently lift the battery out of the iPhone

…

59. De-route the digitizer and LCD cables through the steel inner frame, and remove the display from the iPhone
60. To reassemble your device, follow these instructions in reverse order.

Summary

- Grounding relates language to a model environment with more (or different) structure than formal calculus

- Lots of tools for using environment models to learn semantics without annotated logical forms

Question time