CS 4650/7650
Natural Language Semantics

Jacob Eisenstein
(some slides by Artzi, FitzGerald, and Zettlemoyer)

October 21, 2014

Why semantics?

Goal is to convert text into structured knowledge representations.
Some motivations:

- Automatically update databases of facts
- Infer new facts and relationships
- Answer complex questions, e.g.,
  what cheese-exporting countries are hereditary monarchies?
- Logic-check written arguments
- ...

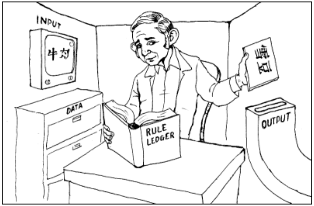
Why semantics?

Semantics is a stumbling block for NLP at all levels:

- I shot an elephant in my pajamas
- How to solve PP attachment question?
- Bilexical probabilities are just a noisy approximation

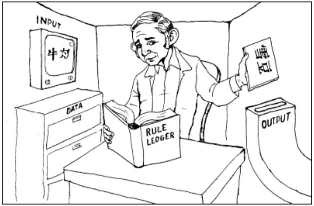
Can your computer ever *really* understand you?

What does it really mean to understand language anyway?



Can your computer ever *really* understand you?

What does it really mean to understand language anyway?



Some functional answers:

- Answer reading comprehension tests
- Determine whether a statement is true or false
- Choose the appropriate action
- Convert text to a *meaning representation*

The semantics roadmap

- **Compositional semantics**
  assemble the meaning of a sentence from its components
- **Shallow semantics**
  identify the predicates and their arguments and adjuncts
- **Distributional semantics**
  vector-space models for the meaning of words and phrases

Language to Meaning

More informative

Language to Meaning

Information
Extraction

Recover information
about pre-specified
relations and entities

More informative

Example Task

Relation Extraction

The New York Times

OBAMA

RACIAL BARRIER FALLS IN DECISIVE VICTORY

Democrats in Congress Strengthen Grip

$$is\_a(OBAMA, PRESIDENT)$$

Language to Meaning

Broad-coverage Semantics

Focus on specific phenomena (e.g., verb-argument matching)

More informative

Example Task

Summarization

Obama wins election. Big party in Chicago. Romney a bit down, asks for some tea.

Language to Meaning

Semantic Parsing

Recover **complete** meaning representation

More informative

Example Task

Database Query

What states border Texas?

Oklahoma
New Mexico
Arkansas
Louisiana

Language to Meaning

Semantic Parsing

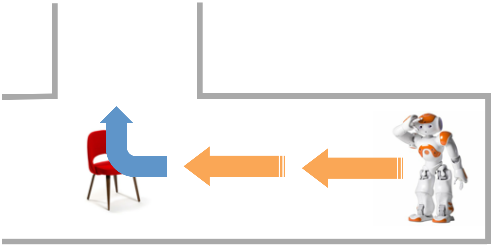
Recover **complete** meaning representation

More informative

Example Task

Instructing a Robot

at the chair,
turn right



Language to Meaning

Semantic Parsing

Recover **complete** meaning representation

More informative

Complete meaning is sufficient to complete the task

- Convert to database query to get the answer
- Allow a robot to do planning

Language to Meaning

at the chair, move forward three steps past the sofa

$$\lambda a.pre(a, \iota x.chair(x)) \wedge move(a) \wedge len(a, 3) \wedge dir(a, forward) \wedge past(a, \iota y.sofa(y))$$

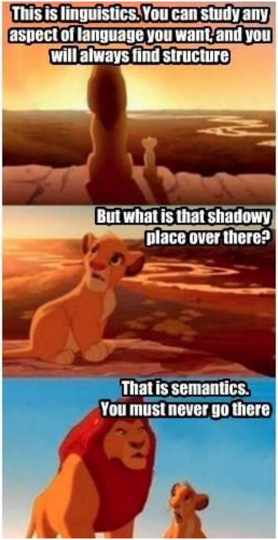
Language to Meaning

at the chair, move forward three steps past the sofa

$$\lambda a.pre(a, \iota x.chair(x)) \land move(a) \land len(a, 3) \land$$
$$dir(a, forward) \land past(a, \iota y.sofa(y))$$

But beware!

- Semantics is difficult to model formally.
- The syntax-semantic interface is treacherous.
- Simple sentences can be surprisingly hard to analyze — even for humans.



What do we want from a meaning representation?

- **Verifiability**: can check statements against KB
- **No ambiguity**: one meaning per statement
- **Canonical form**: one statement per meaning
- **Expressiveness**: we can say what we want
- **Inference**

Inference

Some inferences are purely linguistic.

$$\frac{\text{some wugs are wags} \quad \text{all wags are foo}}{\text{some wugs are foo}}$$

Inference

Some inferences are purely linguistic.

$$\frac{\text{some wugs are wags} \quad \text{all wags are foo}}{\text{some wugs are foo}}$$

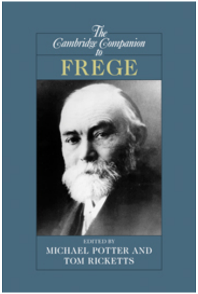
Other inferences require world knowledge.

$$\frac{\text{Christopher Walken was born in Queens}}{\text{Christopher Walken was not born in Atlanta}}$$

Requirements for inference

Linguistic inference requires:

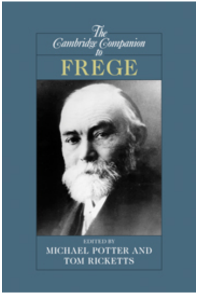
- A mapping from words to logical symbols
- **Compositional** rules that allow us to build logical statements from multi-word units
- Variables
- Inference rules



Requirements for inference

Linguistic inference requires:

- A mapping from words to logical symbols
- **Compositional** rules that allow us to build logical statements from multi-word units
- Variables
- Inference rules



World-knowledge inference requires

- **Grounding** the semantic representation in a **model** of the world.

First-order logic

- **Terms** are the basic building blocks.
  - constants, e.g., $\textsc{Moe's}$, $\textsc{Tin-Drum}$
  - functions of other terms, e.g., $\textsc{LocationOf}(\textsc{Tin-Drum})$
  - variables, e.g. x, y

First-order logic

- **Terms** are the basic building blocks.
  - constants, e.g., $\textsc{Moe's}$, $\textsc{Tin-Drum}$
  - functions of other terms, e.g., $\textsc{LocationOf}(\textsc{Tin-Drum})$
  - variables, e.g. x, y
- **Predicates** describe relations between terms: $\textsc{Serves}(\textsc{Moe's},\textsc{burritos})$

First-order logic

- **Terms** are the basic building blocks.
  - constants, e.g., $\textsc{Moe's}$, $\textsc{Tin-Drum}$
  - functions of other terms, e.g., $\textsc{LocationOf}(\textsc{Tin-Drum})$
  - variables, e.g. x, y
- **Predicates** describe relations between terms: $\textsc{Serves}(\textsc{Moe's},\textsc{burritos})$
- **Formulae** are combinations of predicates, using
  - **connectives**, e.g., $\textsc{Serves}(\textsc{Moe's},\textsc{burritos}) \wedge \neg\textsc{Expensive}(\textsc{Moe's})$
  - **quantifiers**
    - Existential: $\exists x.\textsc{Serves}(\textsc{Moe's}, x)$
    - Universal: $\forall x.\textsc{Serves}(x, \textsc{pizza}) \Rightarrow \textsc{Vegetarian}(x)$

Lambda calculus

Lambda expressions describe "anonymous" functions.

- You may know them from functional programming, e.g.

$$\text{SQUARE} = \lambda x.x * x$$

- Lambda calculus is very useful for composing large formulae out of smaller ones.
  - What vegetarian items are served at Moe's?
  - Things that are vegetarian: $\lambda x.\text{VEGETARIAN}(x)$
  - Things that Moe's serves: $\lambda x.\text{SERVES}(\text{MOE'S}, x)$
  - Vegetarian things that Moe's serves: $\lambda x.\text{SERVES}(\text{MOE'S}, x) \wedge \text{VEGETARIAN}(x)$

Model-theoretic semantics

Model-theoretic semantics is the bridge between a representation and a "model" of the world.

- The *domain* of the model is a set of objects: *a,b,c,d,e,...*
  Each non-logical symbol has a *denotation* as an object:
  $[\![\textsc{Tin-Drum}]\!] = a$, $[\![\textsc{noodles}]\!] = e$
- *Relations* are sets of tuples: $\textsc{Serves} = \{\langle a,e\rangle, \langle b,e\rangle, \langle b,f\rangle, \ldots\}$
- *Properties* are relations between *objects* and *booleans*.
  Equivalently, properties are just sets of objects.
  $\textsc{cheap} = \{a, b\}$, $\textsc{noisy} = \{a, c\}$
- Many different logical statements have the same denotation:

$$[\![\textsc{brother-of}(\textsc{lisa})]\!] = [\![\textsc{son-of}(\textsc{homer})]\!]$$
$$= [\![\textsc{son-of}(\textsc{marge})]\!] = [\![\textsc{bart}]\!]$$

Event semantics

Descriptions of events can become complex:

- Jones buttered the toast
- Jones buttered the toast with the knife
- Jones buttered the toast with the knife in the bathroom
- Jones buttered the toast with the knife in the bathroom at midnight

Event semantics

Descriptions of events can become complex:

- Jones buttered the toast
- Jones buttered the toast with the knife
- Jones buttered the toast with the knife in the bathroom
- Jones buttered the toast with the knife in the bathroom at midnight

Davidsonian event semantics handles this by **reifying** the event:

$$\exists e \textsc{butter}(e, \textsc{Jones}, \textsc{the toast}) \\ \& \textsc{with}(e, \textsc{the knife}) \\ \& \textsc{in}(e, \textsc{the bathroom}) \\ \& \textsc{at}(e, \textsc{midnight})$$

Neo-Davidsonian event semantics

Jones buttered the toast with the knife in the bathroom at midnight

$$
\begin{aligned}
\exists e\ & \textsc{butter}(e) \\
& \& \textsc{Agent}(e, \textsc{Jones}) \\
& \& \textsc{Patient}(e, \textsc{the toast}) \\
& \& \textsc{Instrument}(E, \textsc{the knife}) \\
& \& \textsc{In}(E, \textsc{the bathroom}) \\
& \& \textsc{At}(E, \textsc{midnight})
\end{aligned}
$$

- Predicates (e.g., butter) label events
- Events have **thematic roles** that are shared across many predicates

Constructing Lambda Calculus Expressions

at the chair, move forward three steps past the sofa

?

$$\lambda a.pre(a, \iota x.chair(x)) \land move(a) \land len(a, 3) \land dir(a, forward) \land past(a, \iota y.sofa(y))$$

Between CCGs and CFGs

	CFGs	CCGs
Combination operations	Many	Few
Parse tree nodes	Non-terminals	Categories
Syntactic symbols	Few dozen	Handful, but can combine
Paired with words	POS tags	Categories

Examples from Manning

red car in Atlanta	spurious ambiguity
Kathy runs in Atlanta	event semantics
the red car in Atlanta	quantificiation
every student runs	quantificiation, argument raising
some kid broke every toy	ambiguity, argument raising
what does Kathy like?	gap threading

“The” semantics in SQL

```
select Cars.obj from Cars, Locations, Red
where Cars.obj=Locations.obj
AND Locations.place = 'Atlanta'
AND Cars.obj = Red.obj
HAVING count(*) = 1
```

Parsing with CCGs

square blue or round yellow pillow

Parsing with CCGs

$$\dfrac{\text{square}}{\begin{array}{c} ADJ \\ \lambda x.square(x) \end{array}} \quad \dfrac{\text{blue}}{\begin{array}{c} ADJ \\ \lambda x.blue(x) \end{array}} \quad \dfrac{\text{or}}{\begin{array}{c} C \\ disj \end{array}} \quad \dfrac{\text{round}}{\begin{array}{c} ADJ \\ \lambda x.round(x) \end{array}} \quad \dfrac{\text{yellow}}{\begin{array}{c} ADJ \\ \lambda x.yellow(x) \end{array}} \quad \dfrac{\text{pillow}}{\begin{array}{c} N \\ \lambda x.pillow(x) \end{array}}$$

Use lexicon to match words and phrases with their categories

Parsing with CCGs

square	blue	or	round	yellow	pillow
ADJ $\lambda x.square(x)$	ADJ $\lambda x.blue(x)$	C $disj$	ADJ $\lambda x.round(x)$	ADJ $\lambda x.yellow(x)$	N $\lambda x.pillow(x)$
N/N $\lambda f.\lambda x.f(x) \wedge square(x)$					

Shift adjectives to combine

$$ADJ : \lambda x.g(x) \Rightarrow N/N : \lambda f.\lambda x.f(x) \wedge g(x)$$

Parsing with CCGs

square	blue	or	round	yellow	pillow
ADJ $\lambda x.square(x)$	ADJ $\lambda x.blue(x)$	C $disj$	ADJ $\lambda x.round(x)$	ADJ $\lambda x.yellow(x)$	N $\lambda x.pillow(x)$
N/N $\lambda f.\lambda x.f(x) \wedge square(x)$	N/N $\lambda f.\lambda x.f(x) \wedge blue(x)$		N/N $\lambda f.\lambda x.f(x) \wedge round(x)$	N/N $\lambda f.\lambda x.f(x) \wedge yellow(x)$	

Shift adjectives to combine

$$ADJ : \lambda x.g(x) \Rightarrow N/N : \lambda f.\lambda x.f(x) \wedge g(x)$$

Parsing with CCGs

square	blue	or	round	yellow	pillow
ADJ $\lambda x.square(x)$	ADJ $\lambda x.blue(x)$	C $disj$	ADJ $\lambda x.round(x)$	ADJ $\lambda x.yellow(x)$	N $\lambda x.pillow(x)$
N/N $\lambda f.\lambda x.f(x) \land square(x)$	N/N $\lambda f.\lambda x.f(x) \land blue(x)$		N/N $\lambda f.\lambda x.f(x) \land round(x)$	N/N $\lambda f.\lambda x.f(x) \land yellow(x)$	
$>\mathbf{B}$ N/N $\lambda f.\lambda x.f(x) \land square(x) \land blue(x)$			$>\mathbf{B}$ N/N $\lambda f.\lambda x.f(x) \land round(x) \land yellow(x)$		

Compose pairs of adjectives

$$A/B : f \quad B/C : g \Rightarrow A/C : \lambda x.f(g(x)) \quad (> B)$$

Parsing with CCGs

square	blue	or	round	yellow	pillow
ADJ $\lambda x.square(x)$	ADJ $\lambda x.blue(x)$	C $disj$	ADJ $\lambda x.round(x)$	ADJ $\lambda x.yellow(x)$	N $\lambda x.pillow(x)$
N/N $\lambda f.\lambda x.f(x) \wedge square(x)$	N/N $\lambda f.\lambda x.f(x) \wedge blue(x)$		N/N $\lambda f.\lambda x.f(x) \wedge round(x)$	N/N $\lambda f.\lambda x.f(x) \wedge yellow(x)$	

$$\frac{N/N \quad N/N}{N/N \;\; \lambda f.\lambda x.f(x) \wedge square(x) \wedge blue(x)} >\mathbf{B} \qquad \frac{N/N \quad N/N}{N/N \;\; \lambda f.\lambda x.f(x) \wedge round(x) \wedge yellow(x)} >\mathbf{B}$$

$$\frac{}{N/N \;\; \lambda f.\lambda x.f(x) \wedge ((square(x) \wedge blue(x)) \vee (round(x) \wedge yellow(x)))} <\Phi>$$

Coordinate composed adjectives

Parsing with CCGs

$$\begin{array}{cccccc}
\text{square} & \text{blue} & \text{or} & \text{round} & \text{yellow} & \text{pillow} \\
\hline
ADJ & ADJ & C & ADJ & ADJ & N \\
\lambda x.square(x) & \lambda x.blue(x) & disj & \lambda x.round(x) & \lambda x.yellow(x) & \lambda x.pillow(x) \\
\hline
N/N & N/N & & N/N & N/N & \\
\lambda f.\lambda x.f(x) \land square(x) & \lambda f.\lambda x.f(x) \land blue(x) & & \lambda f.\lambda x.f(x) \land round(x) & \lambda f.\lambda x.f(x) \land yellow(x) & \\
\hline
\multicolumn{2}{c}{>\mathbf{B}} & & \multicolumn{2}{c}{>\mathbf{B}} & \\
\multicolumn{2}{c}{N/N} & & \multicolumn{2}{c}{N/N} & \\
\multicolumn{2}{c}{\lambda f.\lambda x.f(x) \land square(x) \land blue(x)} & & \multicolumn{2}{c}{\lambda f.\lambda x.f(x) \land round(x) \land yellow(x)} & \\
\hline
\multicolumn{5}{c}{<\Phi>} & \\
\multicolumn{5}{c}{N/N} & \\
\multicolumn{5}{c}{\lambda f.\lambda x.f(x) \land ((square(x) \land blue(x)) \lor (round(x) \land yellow(x)))} & \\
\hline
\multicolumn{6}{c}{>} \\
\multicolumn{6}{c}{N} \\
\multicolumn{6}{c}{\lambda x.pillow(x) \land ((square(x) \land blue(x)) \lor (round(x) \land yellow(x)))}
\end{array}$$

Apply coordinated adjectives to noun

$$A/B : f \quad B : g \Rightarrow A : f(g) \quad (>)$$

Parsing with CCGs

$$
\begin{array}{ll}
x & \begin{array}{ccc} \text{CCG} & \text{is} & \text{fun} \end{array} \\[1em]
y & \left\{ \begin{array}{ccc}
\hline
NP & S\backslash NP/ADJ & ADJ \\
CCG & \lambda f.\lambda x.f(x) & \lambda x.fun(x) \\
\cline{2-3}
 & \multicolumn{2}{r}{>} \\
 & \multicolumn{2}{c}{S\backslash NP} \\
 & \multicolumn{2}{c}{\lambda x.fun(x)} \\
\hline
\multicolumn{3}{r}{<} \\
\multicolumn{3}{c}{S} \\
\multicolumn{3}{c}{z \quad fun(CCG)}
\end{array} \right.
\end{array}
$$

Lexical Ambiguity + Many parsing decisions → Many potential trees and LFs

Weighted Linear CCGs

- Given a weighted linear model:
  - CCG lexicon Λ
  - Feature function $f : X \times Y \to \mathbb{R}^m$
  - Weights $w \in \mathbb{R}^m$
- The best parse is:

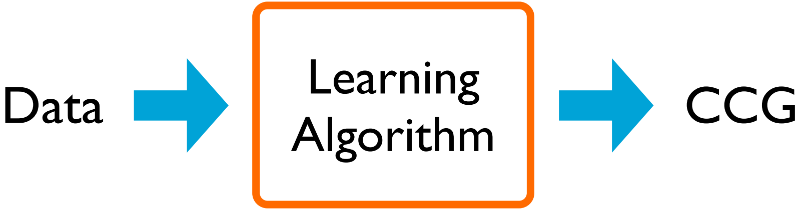
$$y^* = \arg\max_y w \cdot f(x, y)$$

- We consider all possible parses y for sentence x given the lexicon Λ

Parsing Algorithms

- Syntax-only CCG parsing has polynomial time CKY-style algorithms
- Parsing with semantics requires entire category as chart signature
  - e.g., $ADJ : \lambda x.fun(x)$
- In practice, prune to top-N for each span
  - Approximate, but polynomial time

Learning



- What kind of data/supervision we can use?
- What do we need to learn?

Parsing as Structure Prediction

$$
\begin{array}{ccccc}
\text{show} \quad \text{me} & \text{flights} & \text{to} & \text{Boston} & \\
\hline
S/N & N & PP/NP & NP & \\
\lambda f.f & \lambda x.flight(x) & \lambda y.\lambda x.to(x,y) & BOSTON & \\
\cline{3-4}
 & & \multicolumn{2}{c}{PP} & > \\
 & & \multicolumn{2}{c}{\lambda x.to(x,BOSTON)} & \\
\cline{3-4}
 & & \multicolumn{2}{c}{N\backslash N} & \\
 & & \multicolumn{2}{c}{\lambda f.\lambda x.f(x) \wedge to(x,BOSTON)} & \\
\cline{2-4}
 & \multicolumn{3}{c}{N} & < \\
 & \multicolumn{3}{c}{\lambda x.flight(x) \wedge to(x,BOSTON)} & \\
\cline{1-4}
\multicolumn{4}{c}{S} & > \\
\multicolumn{4}{c}{\lambda x.flight(x) \wedge to(x,BOSTON)} &
\end{array}
$$

Learning CCG

$$
\begin{array}{cccc}
\text{show} \quad \text{me} & \text{flights} & \text{to} & \text{Boston} \\
\hline
S/N & N & PP/NP & NP \\
\lambda f.f & \lambda x.flight(x) & \lambda y.\lambda x.to(x, y) & BOSTON \\
\cline{3-4}
 & & \multicolumn{2}{c}{PP \quad >} \\
 & & \multicolumn{2}{c}{\lambda x.to(x, BOSTON)} \\
\cline{3-4}
 & & \multicolumn{2}{c}{N\backslash N} \\
 & & \multicolumn{2}{c}{\lambda f.\lambda x.f(x) \land to(x, BOSTON)} \\
\cline{2-4}
 & \multicolumn{3}{c}{N \quad <} \\
 & \multicolumn{3}{c}{\lambda x.flight(x) \land to(x, BOSTON)} \\
\hline
\multicolumn{4}{c}{S \quad >} \\
\multicolumn{4}{c}{\lambda x.flight(x) \land to(x, BOSTON)}
\end{array}
$$

Lexicon

Combinators

Learning CCG

show me	flights	to	Boston
S/N	N	PP/NP	NP
$\lambda f.f$	$\lambda x.flight(x)$	$\lambda y.\lambda x.to(x,y)$	$BOSTON$

$$\frac{PP/NP \quad NP}{PP \quad \lambda x.to(x, BOSTON)} >$$

$$\frac{PP \quad \lambda x.to(x, BOSTON)}{N\backslash N \quad \lambda f.\lambda x.f(x) \wedge to(x, BOSTON)}$$

$$\frac{N \quad N\backslash N}{N \quad \lambda x.flight(x) \wedge to(x, BOSTON)} <$$

$$\frac{S/N \quad N}{S \quad \lambda x.flight(x) \wedge to(x, BOSTON)} >$$

Lexicon

Combinators

Predefined

Learning CCG

$$
\begin{array}{c}
\text{show me} \quad \text{flights} \quad \text{to} \quad \text{Boston} \\
\hline
\begin{array}{cccc}
S/N & N & PP/NP & NP \\
\lambda f.f & \lambda x.flight(x) & \lambda y.\lambda x.to(x,y) & BOSTON
\end{array} \\
\hline
PP \quad (>) \\
\lambda x.to(x, BOSTON) \\
\hline
N\backslash N \\
\lambda f.\lambda x.f(x) \wedge to(x, BOSTON) \\
\hline
N \quad (<) \\
\lambda x.flight(x) \wedge to(x, BOSTON) \\
\hline
S \quad (>) \\
\lambda x.flight(x) \wedge to(x, BOSTON)
\end{array}
$$

Lexicon

Combinators

Predefined

w

Supervised Data

$$
\begin{array}{ccccc}
\text{show} \quad \text{me} & \text{flights} & \text{to} & & \text{Boston} \\
\hline
S/N & N & PP/NP & & NP \\
\lambda f.f & \lambda x.flight(x) & \lambda y.\lambda x.to(x,y) & & BOSTON \\
\cline{3-5}
 & & \multicolumn{3}{c}{PP \quad >} \\
 & & \multicolumn{3}{c}{\lambda x.to(x, BOSTON)} \\
\cline{3-5}
 & & \multicolumn{3}{c}{N \backslash N} \\
 & & \multicolumn{3}{c}{\lambda f.\lambda x.f(x) \wedge to(x, BOSTON)} \\
\cline{2-5}
 & \multicolumn{4}{c}{N \quad <} \\
 & \multicolumn{4}{c}{\lambda x.flight(x) \wedge to(x, BOSTON)} \\
\hline
\multicolumn{5}{c}{S \quad >} \\
\multicolumn{5}{c}{\lambda x.flight(x) \wedge to(x, BOSTON)}
\end{array}
$$

Supervised Data

show me	flights	to	Boston
S/N	N	PP/NP	NP
$\lambda f.f$	$\lambda x.flight(x)$	$\lambda y.\lambda x.to(x,y)$	$BOSTON$

$$\frac{PP/NP \quad NP}{PP : \lambda x.to(x, BOSTON)}>$$

$$N\backslash N : \lambda f.\lambda x.f(x) \land to(x, BOSTON)$$

$$N : \lambda x.flight(x) \land to(x, BOSTON) \;<$$

$$S \;>$$

$$\lambda x.flight(x) \land to(x, BOSTON)$$

Latent

Supervised Data

Supervised learning is done from pairs
of sentences and logical forms

Show me flights to Boston
$\lambda x.flight(x) \wedge to(x, BOSTON)$

I need a flight from baltimore to seattle
$\lambda x.flight(x) \wedge from(x, BALTIMORE) \wedge to(x, SEATTLE)$

what ground transportation is available in san francisco
$\lambda x.ground\_transport(x) \wedge to\_city(x, SF)$

[Zettlemoyer and Collins 2005; 2007]

Weak Supervision

- Logical form is latent
- “Labeling” requires less expertise
- Labels don’t uniquely determine correct logical forms
- Learning requires executing logical forms within a system and evaluating the result

Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

$$New\ Mexico$$

[Clarke et al. 2010; Liang et al. 2011]

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

$$New\ Mexico$$

$$argmax(\lambda x.state(x) \\ \wedge border(x, TX), \lambda y.size(y))$$

$$argmax(\lambda x.river(x) \\ \wedge in(x, TX), \lambda y.size(y))$$

[Clarke et al. 2010; Liang et al. 2011]

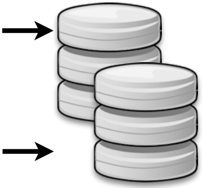
Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

$$New\ Mexico$$

$$argmax(\lambda x.state(x) \\ \quad \wedge border(x, TX), \lambda y.size(y))$$

$$argmax(\lambda x.river(x) \\ \quad \wedge in(x, TX), \lambda y.size(y))$$



$\rightarrow New\ Mexico$

$\rightarrow Rio\ Grande$

[Clarke et al. 2010; Liang et al. 2011]

Weak Supervision

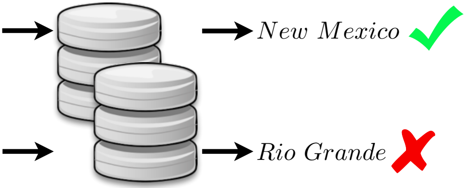
Learning from Query Answers

What is the largest state that borders Texas?

$$New\ Mexico$$

$$argmax(\lambda x.state(x) \\ \quad \wedge border(x, TX), \lambda y.size(y))$$

$$argmax(\lambda x.river(x) \\ \quad \wedge in(x, TX), \lambda y.size(y))$$



[Clarke et al. 2010; Liang et al. 2011]

Hidden Variable Perceptron

Data: $\{(x_i, y_i) : i = 1 \dots n\}$

For $t = 1 \dots T$: [iterate epochs]

For $i = 1 \dots n$: [iterate examples]

$y^*, h^* \leftarrow \arg\max_{y,h} \langle \theta, \Phi(x_i, h, y) \rangle$ [predict]

If $y^* \neq y_i$: [check]

$h' \leftarrow \arg\max_h \langle \theta, \Phi(x_i, h, y_i) \rangle$ [predict hidden]

$\theta \leftarrow \theta + \Phi(x_i, h', y_i) - \Phi(x_i, h^*, y^*)$ [update]

[Liang et al. 2006; Zettlemoyer and Collins 2007]