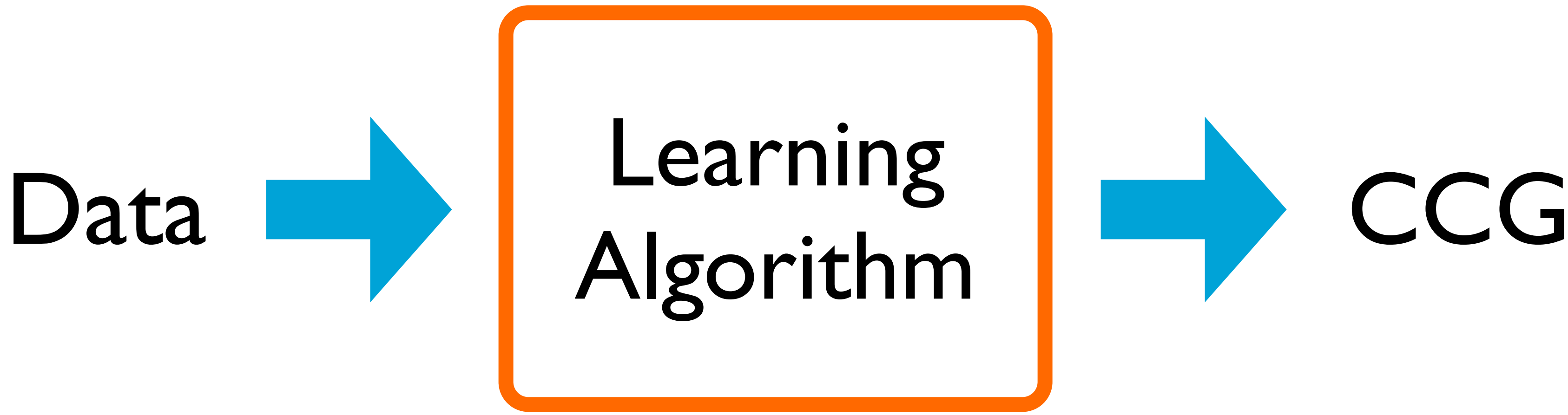
Semantic Parsing with Combinatory Categorial Grammars

Yoav Artzi, Nicholas FitzGerald and Luke Zettlemoyer
University of Washington

ACL 2013 Tutorial
Sofia, Bulgaria

Learning



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

Supervised Data

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

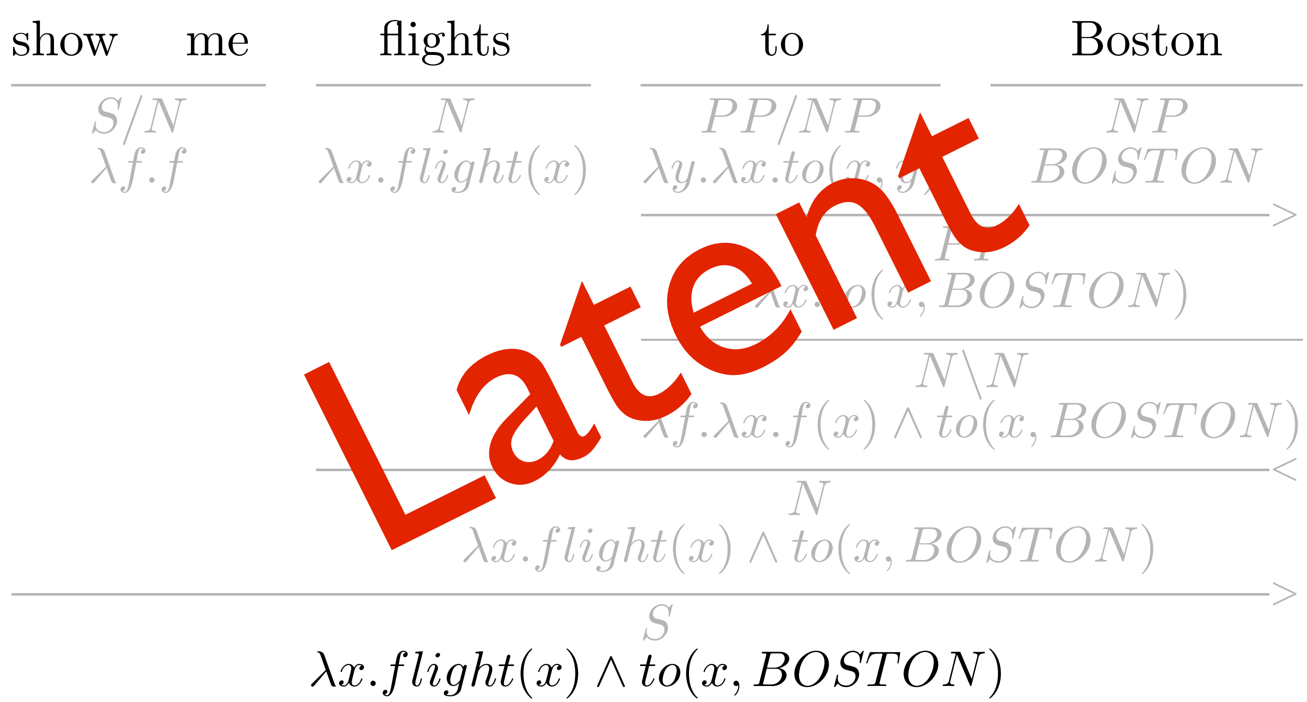
$$\xrightarrow{>} \quad PP \quad \lambda x.to(x, BOSTON)$$

$$N\backslash N \quad \lambda f.\lambda x.f(x) \wedge to(x, BOSTON)$$

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

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

Supervised Data



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?

$$New\ Mexico$$

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

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

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

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

$$New\ Mexico$$

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

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

$\rightarrow New\ Mexico$

$\rightarrow Rio\ Grande$

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

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

$$New\ Mexico$$

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

$\rightarrow$

$\rightarrow New\ Mexico$ ✓

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

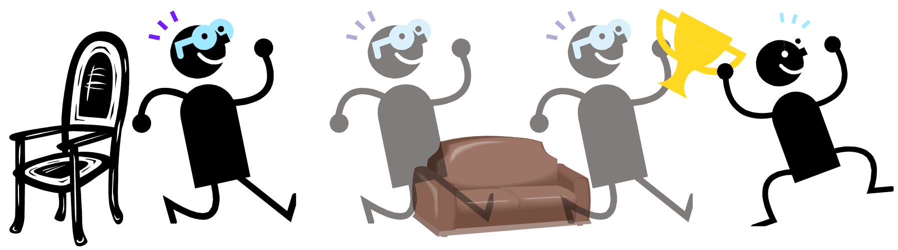
$\rightarrow$ $\rightarrow Rio\ Grande$ ✗

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

Weak Supervision

Learning from Demonstrations

at the chair, move forward three steps past the sofa

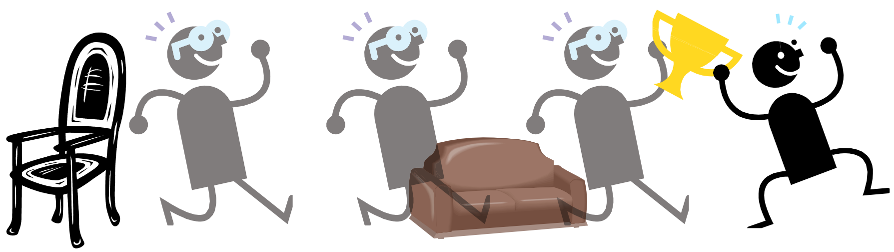


[Chen and Mooney 2011; Kim and Mooney 2012; Artzi and Zettlemoyer 2013b]

Weak Supervision

Learning from Demonstrations

at the chair, move forward three steps past the sofa



Some examples from other domains:

- Sentences and labeled game states [Goldwasser and Roth 2011]
- Sentences and sets of physical objects [Matuszek et al. 2012]

[Chen and Mooney 2011; Kim and Mooney 2012; Artzi and Zettlemoyer 2013b]

Parsing | Learning | Modeling

- Structured perceptron
- A unified learning algorithm
- Supervised learning
- Weak supervision

Structured Perceptron

- Simple additive updates
  - Only requires efficient decoding (argmax)
  - Closely related to maxent and other feature rich models
  - Provably finds linear separator in finite updates, if one exists
- Challenge: learning with hidden variables

Structured Perceptron

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

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

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

$y^* \leftarrow \arg\max_y \langle \theta, \Phi(x_i, y) \rangle$ [predict]

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

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

[Collins 2002]

One Derivation of the Perceptron

Log-linear model: $p(y|x) = \frac{e^{w \cdot f(x,y)}}{\sum_{y'} e^{w \cdot f(x,y')}}$

Step 1: Differentiate, to maximize data log-likelihood

$$update = \sum_i f(x_i, y_i) - E_{p(y|x_i)} f(x_i, y)$$

Step 2: Use online, stochastic gradient updates, for example i:

$$update_i = f(x_i, y_i) - E_{p(y|x_i)} f(x_i, y)$$

Step 3: Replace expectations with maxes (Viterbi approx.)

$$update_i = f(x_i, y_i) - f(x_i, y^*) \text{ where } y^* = \arg\max_y w \cdot f(x_i, y)$$

The Perceptron with Hidden Variables

Log-linear model: $p(y|x) = \sum_h p(y,h|x)$ $\quad p(y,h|x) = \frac{e^{w \cdot f(x,h,y)}}{\sum_{y',h'} e^{w \cdot f(x,h',y')}}$

Step 1: Differentiate marginal, to maximize data log-likelihood

$$update = \sum_i E_{p(h|y_i,x_i)}[f(x_i,h,y_i)] - E_{p(y,h|x_i)}[f(x_i,h,y)]$$

Step 2: Use online, stochastic gradient updates, for example i:

$$update_i = E_{p(y_i,h|x_i)}[f(x_i,h,y_i)] - E_{p(y,h|x_i)}[f(x_i,h,y)]$$

Step 3: Replace expectations with maxes (Viterbi approx.)

$$update_i = f(x_i,h',y_i) - f(x_i,h^*,y^*)$$ where

$y^*, h^* = \arg\max_{y,h} w \cdot f(x_i,h,y)$ and $h' = \arg\max_h w \cdot f(x_i,h,y_i)$

Hidden Variable Perceptron

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

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

For $i = 1 \ldots n$: [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]

Hidden Variable Perceptron

- No known convergence guarantees
  - Log-linear version is non-convex
- Simple and easy to implement
  - Works well with careful initialization
- Modifications for semantic parsing
  - Lots of different hidden information
  - Can add a margin constraint, do probabilistic version, etc.

Learning Choices

Validation Function

$$\mathcal{V} : \mathcal{Y} \rightarrow \{t, f\}$$

- Indicates correctness of a parse y
- Varying $\mathcal{V}$ allows for differing forms of supervision

Lexical Generation Procedure

$$GENLEX(x, \mathcal{V}; \Lambda, \theta)$$

- Given:
  - sentence x
  - validation function $\mathcal{V}$
  - lexicon Λ
  - parameters θ
- Produce a overly general set of lexical entries

Initialize θ using Λ_0 , $\Lambda \leftarrow \Lambda_0$

For $t = 1 \ldots T, i = 1 \ldots n$:

Step 1: (Lexical generation)

a. Set $\lambda_G \leftarrow GENLEX(x_i, \mathcal{V}_i; \Lambda, \theta)$,
$\lambda \leftarrow \Lambda \cup \lambda_G$

b. Let Y be the k highest scoring parses from $GEN(x_i; \lambda)$

c. Select lexical entries from the highest scoring valid parses:
$\lambda_i \leftarrow \bigcup_{y \in MAXV_i(Y;\theta)} LEX(y)$

d. Update lexicon: $\Lambda \leftarrow \Lambda \cup \lambda_i$

Step 2: (Update parameters)

Output: Parameters θ and lexicon Λ

Unification-based $GENLEX(x, z; \Lambda, \theta)$

I want a flight to Boston

$\lambda x.flight(x) \wedge to(x, BOS)$

1. Find highest scoring correct parse
2. Find splits that most increases score
3. **Return new lexical entries**

Iteration 2

to
$$\frac{\text{to}}{(S|NP)/NP \quad \lambda y.\lambda x.to(x,y)}$$

Boston
$$\frac{\text{Boston}}{NP \quad BOS}$$

$$\dfrac{\dfrac{\text{I want a flight}}{S/(S|NP) \quad \lambda f.\lambda x.flight(x) \wedge f(x)} \qquad \dfrac{\text{to Boston}}{S|NP \quad \lambda x.to(x, BOS)}}{S \quad \lambda x.flight(x) \wedge to(x, BOS)} >$$