Learning From and About Context in Semantic Parsing

Luke Zettlemoyer
Computer Science & Engineering
University of Washington

joint work with Yoav Artzi, Michael Collins,
Tom Kwiatkowski, Sharon Goldwater, Mark Steedman

Mapping Sentences to Meaning

`Texas borders Kansas.`

next-to(*TEX*,*KAN*)

Mapping Sentences to Meaning

What states border Texas?

$\lambda x.state(x) \wedge next\text{-}to(x, TEX)$

Mapping Sentences to Meaning

What states border Texas?

$\lambda x.state(x) \wedge next\text{-}to(x, TEX)$

Machine Learning Problem

Given: Many input, output pairs

Learn: A function that maps sentences to lambda-calculus expressions

More Examples

Input: What is the largest state?

Output: $argmax(\lambda x.state(x), \lambda y.size(y))$

Input: What states border the largest state?

Output: $\lambda z.state(z) \wedge borders(z, argmax(\lambda x.state(x), \lambda y.size(y)))$

Input: What states border states that border states ... that border Texas?

Output: $\lambda x.state(x) \wedge \exists y.state(y) \wedge \exists z.state(z) \wedge \ldots \wedge borders(x,y) \wedge borders(y,z) \wedge borders(z,texas)$

Related Work

Supervised semantic parsing with:

Machine Translation	[Papineni et al. 2007; Wong, Mooney 2006, 2007; Matuszek et al. 2010]
Inducting Logic Programming	[Zelle, Money 1996; Tang, Mooney 2000; Thompson, Mooney 2002]
Prob. CFG Parsing	[Miller et al. 2006; Ge, Mooney 2006]
Prob. PDA	[He, Young 2005, 2006]
Support Vector Machines	[Kate, Mooney 2006; Nguyen et al. 2006]
Perceptron-style Learning	[Zettlemoyer, Collins 2005, 2007]
Higher-order unification	[Kwiatkowski 2010,2011]

Less supervision:

Question-Answers Semi-supervision	[Clarke et al. 2010, Liang et al. 2011]
Confidence-driven Unsupervised	[Goldwasser et al. 2011]
Learning from Natural Instructions	[Goldwasser and Roth, 2011]
Learning from Conversations	[Artzi and Zettlemoyer, 2011]

Talk Outline

- Representing and recovering meaning with CCG
- Leaning context-dependent semantic analyses
- Learning from conversations with no meaning annotations

Parsing with CCGs

i'd like to go	from	seattle	to	denver
S/N $\lambda f.f$	$(N\backslash N)/NP$ $\lambda y.\lambda f.\lambda x.f(x) \wedge from(x,y)$	NP SEA	$(N\backslash N)/NP$ $\lambda y.\lambda f.\lambda x.f(x) \wedge to(x,y)$	NP DEN

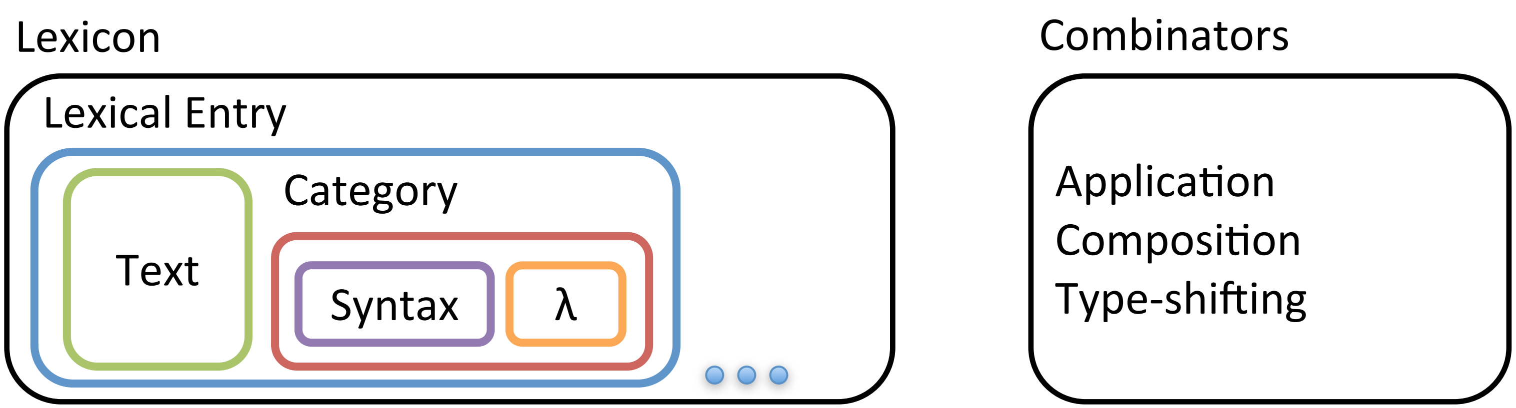
$>$: $(N\backslash N)$ $\quad \lambda f.\lambda x.f(x) \wedge from(x, SEA)$

$>$: $(N\backslash N)$ $\quad \lambda f.\lambda x.f(x) \wedge to(x, DEN)$

$<$**B**: $N\backslash N$ $\quad \lambda f.\lambda x.f(x) \wedge from(x, SEA) \wedge to(x, DEN)$

N $\quad \lambda x.from(x, SEA) \wedge to(x, DEN)$

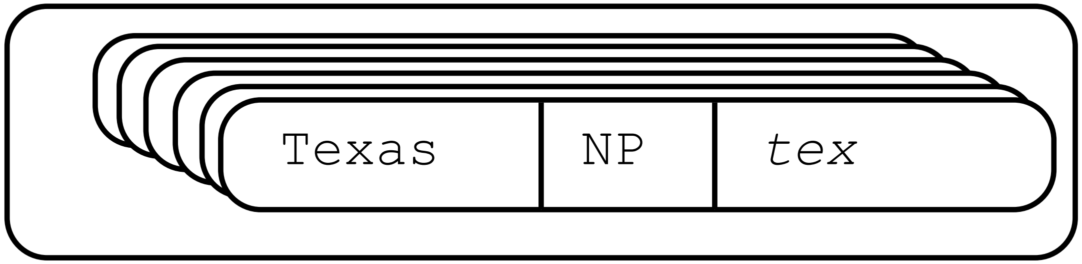
$>$: S $\quad \lambda x.from(x, SEA) \wedge to(x, DEN)$



[Steedman 96,00]

Probabilistic CCGs

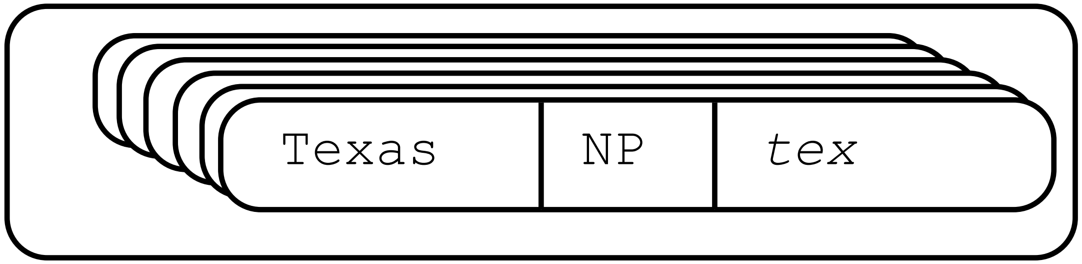
Lexicon: Parameters:

$\Lambda =$ 

$, \quad \theta$

Probabilistic CCGs

Lexicon: $\Lambda =$



Parameters: $, \ \theta$

Probability distribution: sentence x, parse y, logical form z

- Log-linear model:

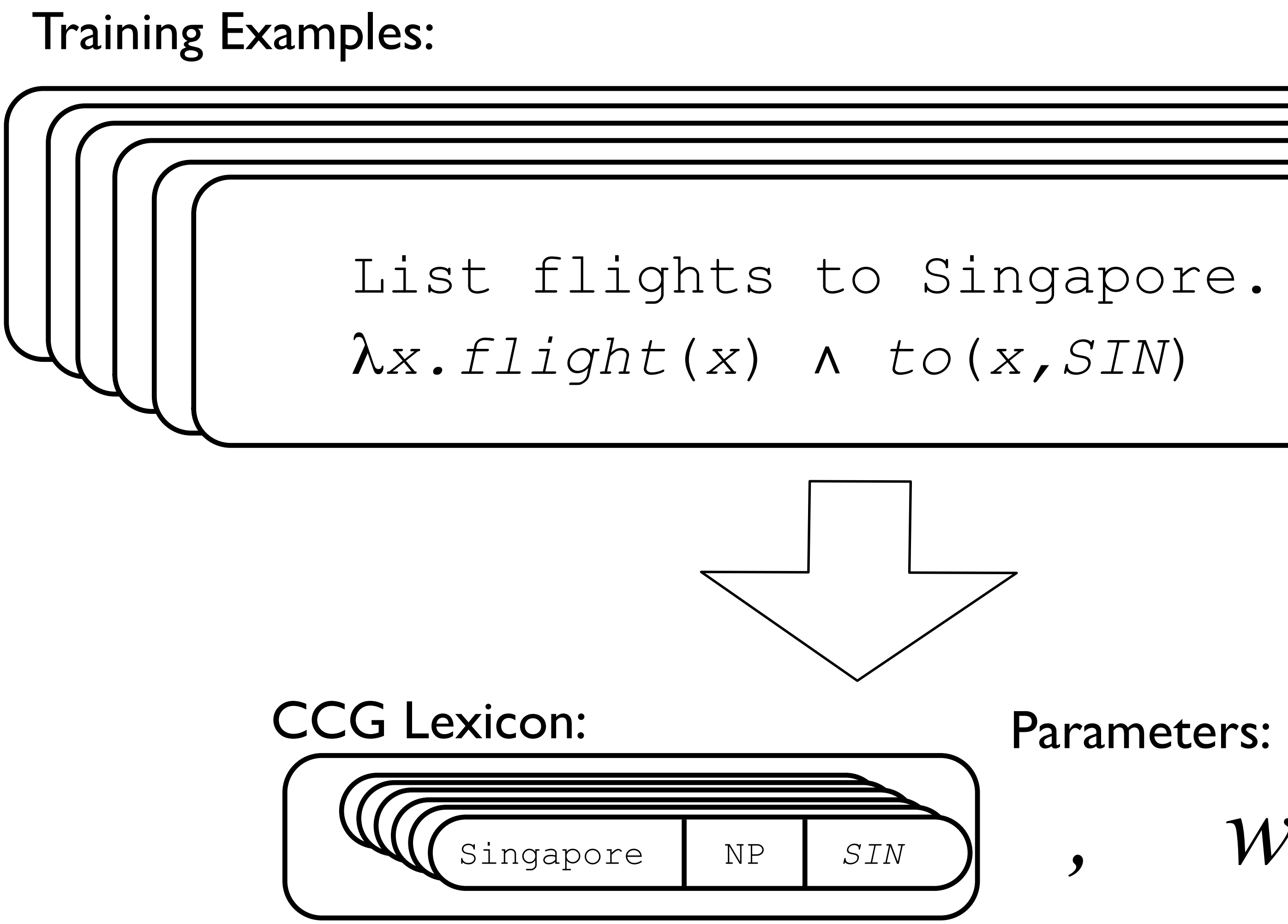
$$P(y, z|x; \theta, \Lambda) = \frac{e^{\theta \cdot \phi(x,y,z)}}{\sum_{(y',z')} e^{\theta \cdot \phi(x,y',z')}}$$

- Parsing:

$$f(x) = \arg\max_z p(z|x; \theta, \Lambda)$$

where $$p(z|x; \theta, \Lambda) = \sum_y p(y, z|x; \theta, \Lambda)$$

Background: Context-independent Learning



A Multilingual Learning Algorithm

Key challenge: learn from data with different natural languages and meaning representations

English, logical-form:

NL: `what states border texas`
MR: $\lambda x.state(x) \wedge next\_to(x, tex)$

Turkish, functional query language:

NL: `texas a siniri olan eyaletler nelerdir`
MR: $answer(state(next\_to\_2(stateid\ \ tex)))$

[Kwiatkowski 2010,2011]

A Multilingual Learning Algorithm

Key challenge: learn from data with different natural languages and meaning representations

English, logical-form:

NL: `what states border texas`
MR: $\lambda x.state(x) \wedge next\_to(x, tex)$

Turkish, functional query language:

NL: `texas a siniri olan eyaletler nelerdir`
MR: $answer(state(next\_to\_2(stateid\ \ tex)))$

Approach: use higher-order unification to recursively split the input LF

[Kwiatkowski 2010,2011]

Example Learned Lexical Entries

English
population of $\vdash NP/NP : \lambda x.population(x)$
smallest $\vdash NP/(S\|NP) : \lambda f.arg\ min(y, f(y), size(y))$
what $\vdash S\|NP/(S\|NP) : \lambda f \lambda x.f(x)$
border $\vdash S\|NP/NP : \lambda x \lambda y.next\_to(y, x)$
state $\vdash S\|NP : \lambda x.state(x)$
most $\vdash NP/(S\|NP)\backslash(S\|NP)\backslash(S\|NP\|NP) :$ $\lambda f \lambda g \lambda h \lambda x.argmax(y, g(y), count(z, f(z, y) \wedge h(z)))$

Turkish
nedir $\vdash S\backslash NP\backslash(NP\|NP) : \lambda f \lambda x.f(x)$
sehir $\vdash S\|NP : \lambda x.city(x)$
nufus yogunlugu $\vdash NP\|NP : \lambda x.density(x)$
siniri$\vdash S\|NP/NP : \lambda x \lambda y.next_to(y, x)$
kac tane $\vdash S\backslash NP/(S\|NP\|NP)\backslash(S\|NP) :$ $\lambda f \lambda g \lambda x.count(y, f(y) \wedge g(y, x))$
ya siniri $\vdash S\|NP\backslash NP : \lambda x \lambda y.next\_to(y, x)$

Japanese
no $\vdash NP\|NP/(NP\|NP) : \lambda f \lambda x.f(x)$
shuu $\vdash S\|NP : \lambda x.state(x)$
nan desu ka $\vdash S\backslash NP\backslash(NP\|NP) : \lambda f \lambda x.f(x)$
wa $\vdash NP\|NP\backslash(NP\|NP) : \lambda f \lambda x.f(x)$
ikutsu $\vdash NP\|(S\|NP)\backslash(S\|NP\|(S\|NP)) :$ $\lambda f \lambda g.count(x, f(g(x)))$
chiiki $\vdash NP\backslash NP : \lambda x.area(x)$

Spanish
en $\vdash S\|NP/NP: \lambda x \lambda y.loc(y, x)$
que es la $\vdash S/NP/(NP\|NP): \lambda f \lambda x.f(x)$
pequena $\vdash NP\backslash(S\|NP)\backslash(NP\|NP) :$ $\lambda g \lambda f.arg\ min(y, f(y), g(y))$
estado $\vdash S\|NP : \lambda x.state(x)$
mas $\vdash S\backslash(S\|NP)/(S\|NP)\backslash(NP\|NP\|(S\|NP)) :$ $\lambda f \lambda g \lambda h.argmax(x, h(x), f(g, x))$
mayores $\vdash S\|NP\backslash(S\|NP) : \lambda f \lambda x.f(x) \wedge major(x)$

Learning Context-dependent Mappings from Sentences to Logical Form

[Zettlemoyer & Collins, ACL 2009]

Context-dependent Analysis

```
Show me flights from New York to Singapore.

Which of those are nonstop?

Show me the cheapest one.

What about connecting?
```

Context-dependent Analysis

Show me flights from New York to Singapore.

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN)$

Which of those are nonstop?

Show me the cheapest one.

What about connecting?

Context-dependent Analysis

Show me flights from New York to Singapore.

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN)$

Which of those are nonstop?

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN) \wedge nonstop(x)$

Show me the cheapest one.

What about connecting?

Context-dependent Analysis

Show me flights from New York to Singapore.

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN)$

Which of those are nonstop?

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN) \wedge nonstop(x)$

Show me the cheapest one.

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN) \wedge nonstop(x),$
$\lambda y.cost(y))$

What about connecting?

Context-dependent Analysis

Show me flights from New York to Singapore.

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN)$

Which of those are nonstop?

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN) \wedge nonstop(x)$

Show me the cheapest one.

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN) \wedge nonstop(x),$
$\lambda y.cost(y))$

What about connecting?

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SIN) \wedge connect(x),$
$\lambda y.cost(y))$

A Supervised Learning Problem

Each training example:
a sequence of sentences and logical forms

Show me flights from New York to Seattle.
$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA)$

List ones from Newark on Friday.
$\lambda x.flight(x) \wedge from(x,NEW) \wedge to(x,SEA) \wedge day(x,FRI)$

Show me the cheapest.
$argmax(\lambda x.flight(x) \wedge from(x,NEW) \wedge to(x,SEA) \wedge day(x,FRI),$
$\lambda y.cost(y))$

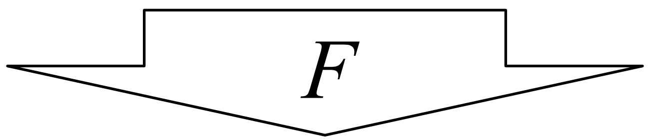
A Supervised Learning Problem

Goal: Find a function F

$\lambda x.flight(x) \wedge from(x,NYC)$
$\wedge to(x,SEA)$

$\lambda x.flight(x) \wedge to(x,SEA)$
$\wedge from(x,NEW) \wedge day(x,FRI)$

Show me the cheapest?



$argmax(\lambda x.flight(x) \wedge from(x,NEW) \wedge to(x,SEA) \wedge day(x,FRI),$
$\lambda y.cost(y))$

[Zettlemoyer & Collins, 2009]

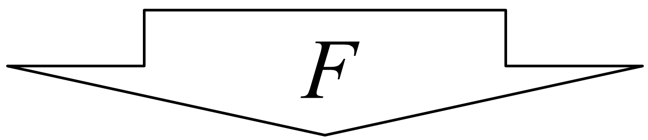
A Supervised Learning Problem

Goal: Find a function F

$\lambda x.flight(x) \land from(x,NYC) \land to(x,SEA)$

$\lambda x.flight(x) \land to(x,SEA) \land from(x,NEW) \land day(x,FRI)$

Show me the cheapest?



$$argmax(\lambda x.flight(x) \land from(x,NEW) \land to(x,SEA) \land day(x,FRI), \lambda y.cost(y))$$

Key Challenges:

- Structured input and output (lambda calculus)
- Hidden variables (only annotate final logical forms)

[Zettlemoyer & Collins, 2009]

An Example Analysis

Show me flights from New York to Seattle.

$\lambda x.flight(x) \land from(x, NYC) \land to(x, SEA)$

List ones from Newark on Friday.

An Example Analysis

Show me flights from New York to Seattle.

$\lambda x.flight(x) \land from(x,NYC) \land to(x,SEA)$

List ones from Newark on Friday.

An Example Analysis

Show me flights from New York to Seattle.

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA)$

List ones from Newark on Friday.

$\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NEW) \wedge day(x,FRI)$

An Example Analysis

Show me flights from New York to Seattle.

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA)$

List ones from Newark on Friday.

$\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NEW) \wedge day(x,FRI)$

An Example Analysis

Context:

$\lambda x.flight(x) \wedge from(x, NYC)$
$\wedge to(x, SEA)$

Current sentence:

List ones from Newark on Friday.

An Example Analysis

Context:

$\lambda x.flight(x) \land from(x, NYC)$
$\land to(x, SEA)$

Current sentence:

`List ones from Newark on Friday.`

Step 1: Context-independent parse

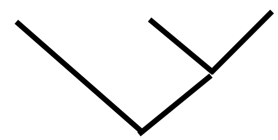
An Example Analysis

Context:

$\lambda x.flight(x) \land from(x,NYC)$
$\land to(x,SEA)$

Current sentence:

`List ones from Newark on Friday.`



$\lambda x.!f(x) \land from(x,NEW) \land day(x,FRI)$

Step 1: Context-independent parse

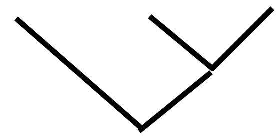
An Example Analysis

Context:

$\lambda x.flight(x) \land from(x,NYC)$
$\land to(x,SEA)$

Current sentence:

`List ones from Newark on Friday.`



$\lambda x.!f(x) \land from(x,NEW) \land day(x,FRI)$

Step 1: Context-independent parse

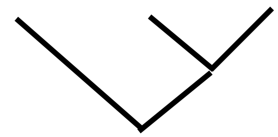
An Example Analysis

Context:

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA)$

Current sentence:

`List ones from Newark on Friday.`



$\lambda x.!f(x) \wedge from(x,NEW) \wedge day(x,FRI)$

Step 1: Context-independent parse

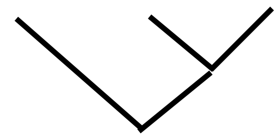
An Example Analysis

Context:

$\lambda x.flight(x) \land from(x, NYC)$
$\land to(x, SEA)$

Current sentence:

`List ones from Newark on Friday.`



$\lambda x.!f(x) \land from(x, NEW) \land day(x, FRI)$

Step 1: Context-independent parse

Step 2: Resolve reference

An Example Analysis

Context:

$\lambda x.flight(x) \wedge from(x,NYC)$
$\wedge to(x,SEA)$

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

Current sentence:

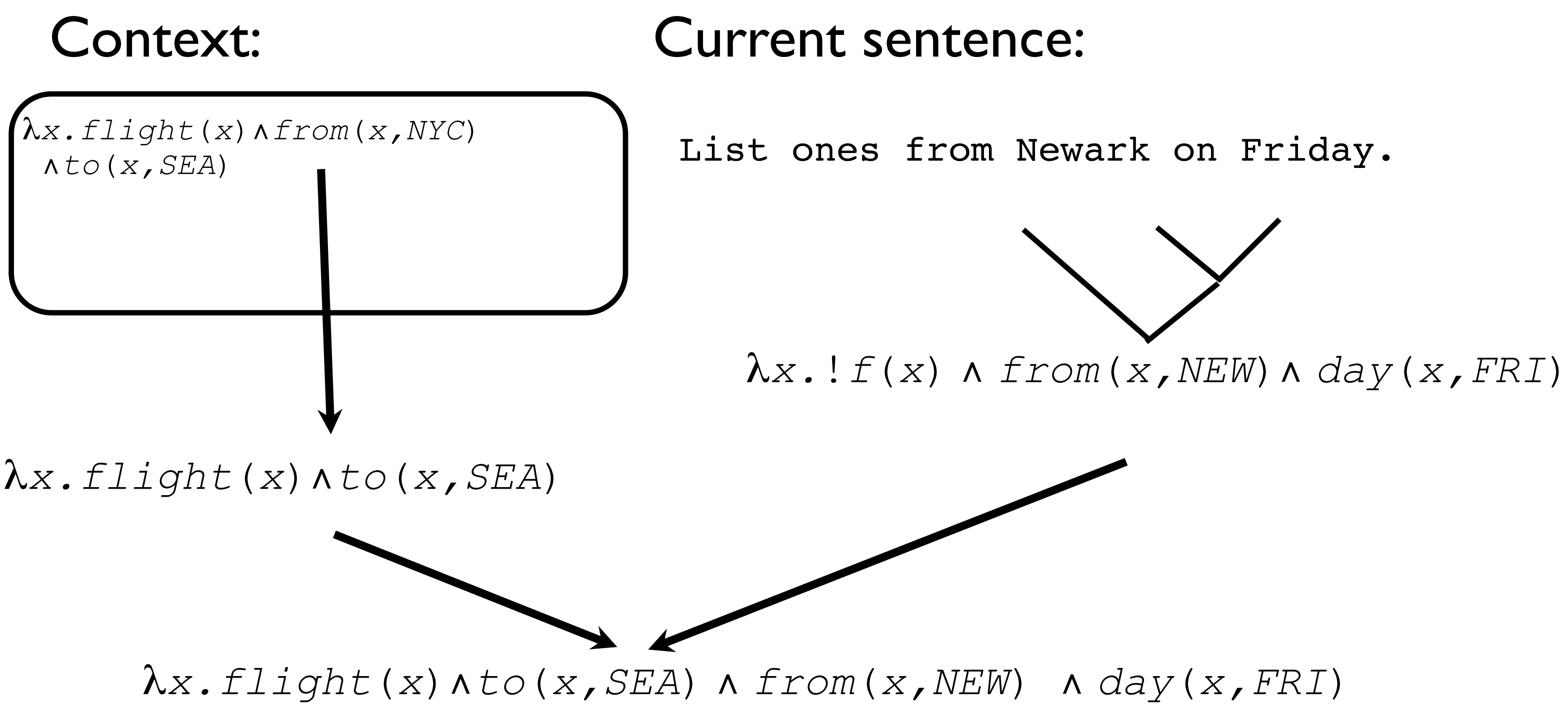
`List ones from Newark on Friday.`

$\lambda x.!f(x) \wedge from(x,NEW) \wedge day(x,FRI)$

Step 1: Context-independent parse

Step 2: Resolve reference

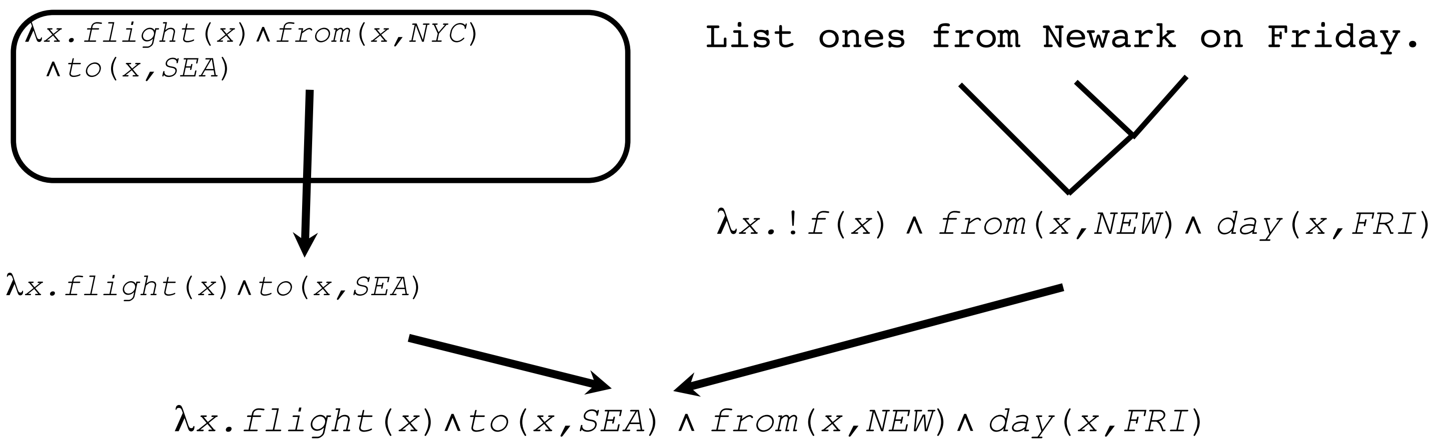
An Example Analysis



Step 1: Context-independent parse

Step 2: Resolve reference

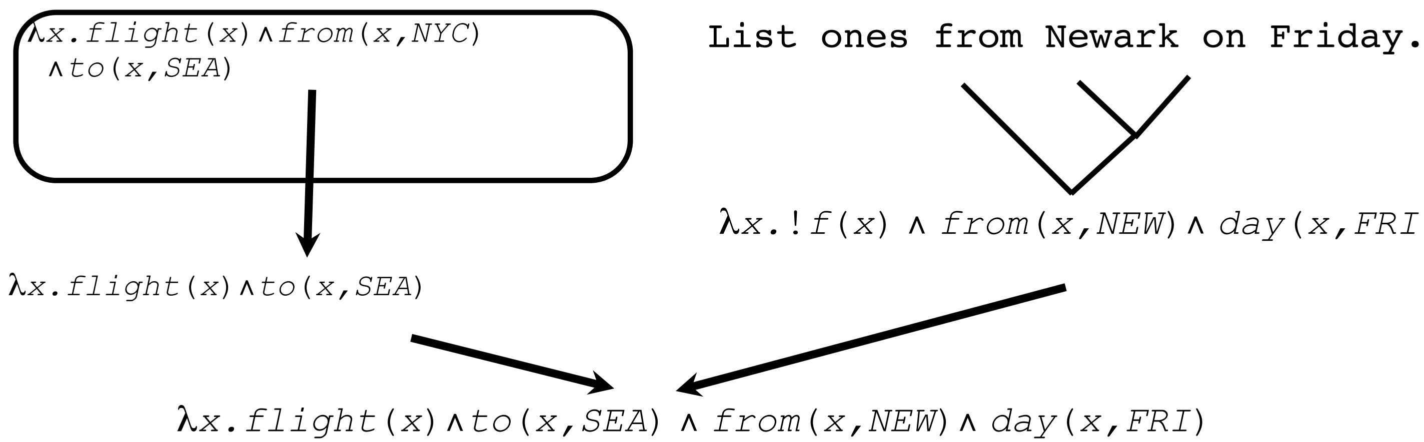
Derivations



Three step process:

- Step 1: Context-independent parsing
- Step 2: Resolve all references
- Step 3: Optionally, perform an elaboration

Derivations

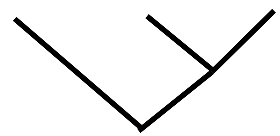


Three step process:

- Step 1: Context-independent parsing
- Step 2: Resolve all references
- Step 3: Optionally, perform an elaboration

Step 1: Referential lexical items

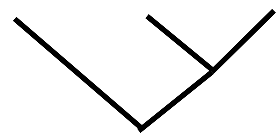
`List ones from Newark on Friday.`



$\lambda x.!f(x) \wedge from(x,NEW) \wedge day(x,FRI)$

Step 1: Referential lexical items

`List ones from Newark on Friday.`



$\lambda x.!f(x) \land from(x,NEW) \land day(x,FRI)$

First extension:

Add referential lexical items

ones	N	$\lambda x.!f(x)$
it	NP	$!e$
...		

Step 1: Type-shifting operations

Second extension:

Add type-shifting operators for elliptical expressions

`the cheapest`

$$\frac{\texttt{the cheapest}}{\begin{array}{c}\mathbf{NP/N} \\ \lambda g.\mathit{argmin}(g, \lambda y.\mathit{cost}(y))\end{array}}$$

Step 1: Type-shifting operations

Second extension:

Add type-shifting operators for elliptical expressions

the cheapest

$$\frac{\qquad\qquad}{\begin{array}{c}\mathbf{NP/N} \\ \lambda g.\mathit{argmin}(g, \lambda y.\mathit{cost}(y))\end{array}}$$

$$\frac{\qquad\qquad}{\begin{array}{c}\mathbf{NP} \\ \mathit{argmin}(\lambda x.!f(x), \lambda y.\mathit{cost}(y))\end{array}}$$

Step 1: Type-shifting operations

Second extension:

Add type-shifting operators for elliptical expressions

the cheapest

NP/N

λg.argmin(g,λy.cost(y))

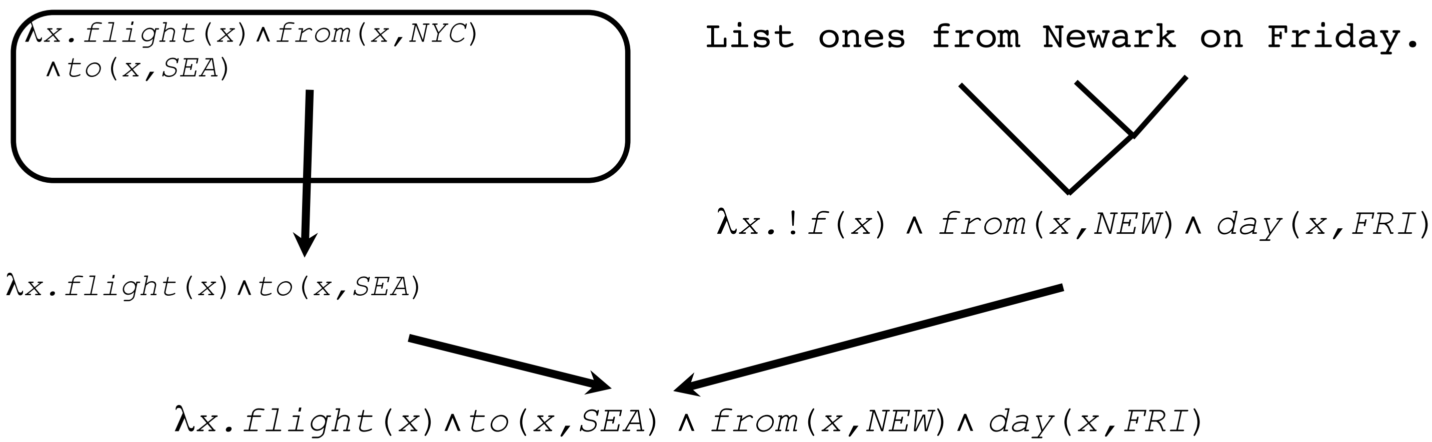
NP

argmin(λx.!f(x),λy.cost(y))

A/B : g => A : g(λ*x*.!*f*(*x*))

where g is a function with input type <e,t>

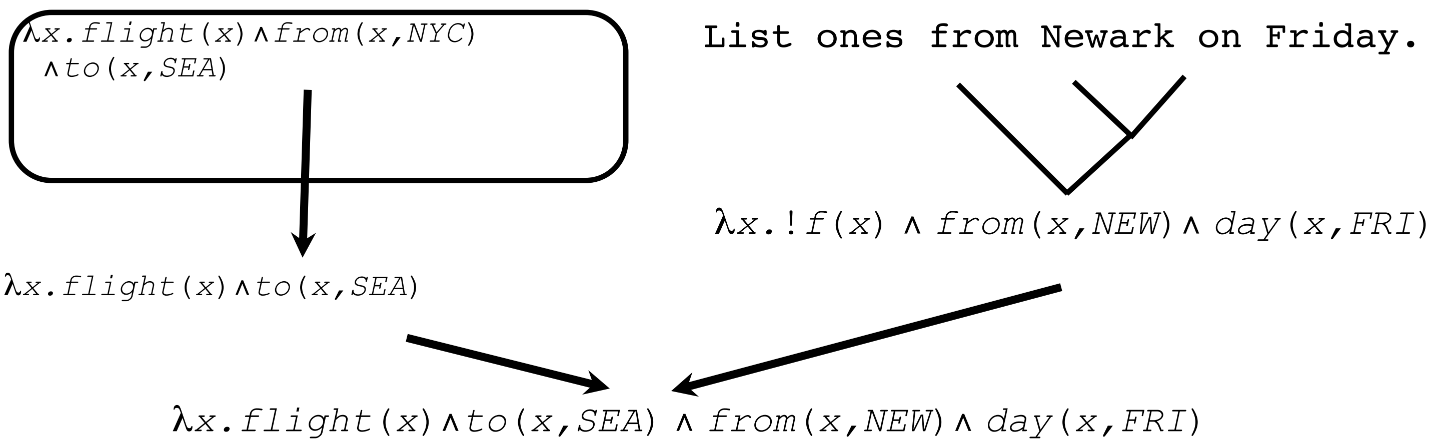
Derivations



Three step process:

- Step 1: Context-independent parsing
- Step 2: Resolve all references
- Step 3: Optionally, perform an elaboration

Step 2: Resolving References



For each reference:

1. Select a (sub)expression from the context
2. Substitute into current analysis

Step 2: Selecting from Context

For each logical form in context,
enumerate `e` and `<e,t>` type subexpressions:

Context:

$\lambda x.flight(x) \wedge from(x,NYC)$
$\wedge to(x,SEA)$

$\lambda x.flight(x) \wedge to(x,SEA)$
$\wedge from(x,NEW) \wedge day(x,FRI)$

$argmax(\lambda x.flight(x) \wedge to(x,SEA)$
$\wedge from(x,BOS),$
$\lambda y.depart(y))$

Step 2: Selecting from Context

For each logical form in context,
enumerate `e` and `<e,t>` type subexpressions:

Context:

$\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA)$ $\rightarrow SEA$

$\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NEW) \wedge day(x,FRI)$

$argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,BOS), \lambda y.depart(y))$

Step 2: Selecting from Context

For each logical form in context,
enumerate e and <e, t> type subexpressions:

Context:

$\lambda x.flight(x) \wedge from(x, NYC) \wedge to(x, SEA)$

$\lambda x.flight(x) \wedge to(x, SEA) \wedge from(x, NEW) \wedge day(x, FRI)$

$argmax(\lambda x.flight(x) \wedge to(x, SEA) \wedge from(x, BOS), \lambda y.depart(y))$

→ SEA

→ NYC

Step 2: Selecting from Context

For each logical form in context,
enumerate `e` and `<e, t>` type subexpressions:

Context:

$\lambda x.flight(x) \wedge from(x, NYC)$
$\wedge\, to(x, SEA)$

$\lambda x.flight(x) \wedge to(x, SEA)$
$\wedge\, from(x, NEW) \wedge day(x, FRI)$

$argmax(\lambda x.flight(x) \wedge to(x, SEA)$
$\wedge\, from(x, BOS),$
$\lambda y.depart(y))$

→ SEA

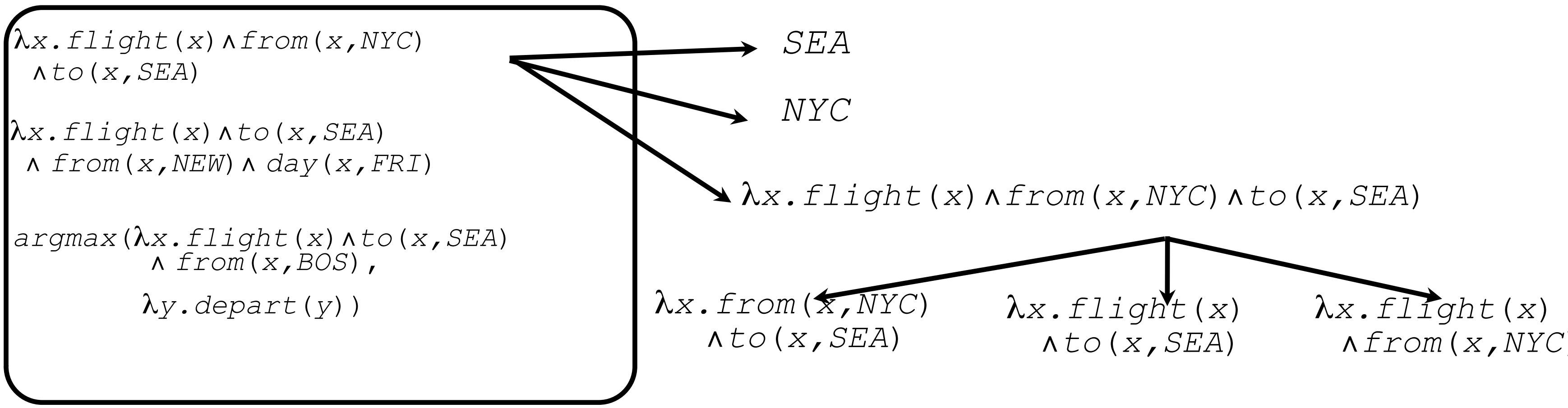
→ NYC

→ $\lambda x.flight(x) \wedge from(x, NYC) \wedge to(x, SEA)$

Step 2: Selecting from Context

For each logical form in context,
enumerate `e` and `<e,t>` type subexpressions:

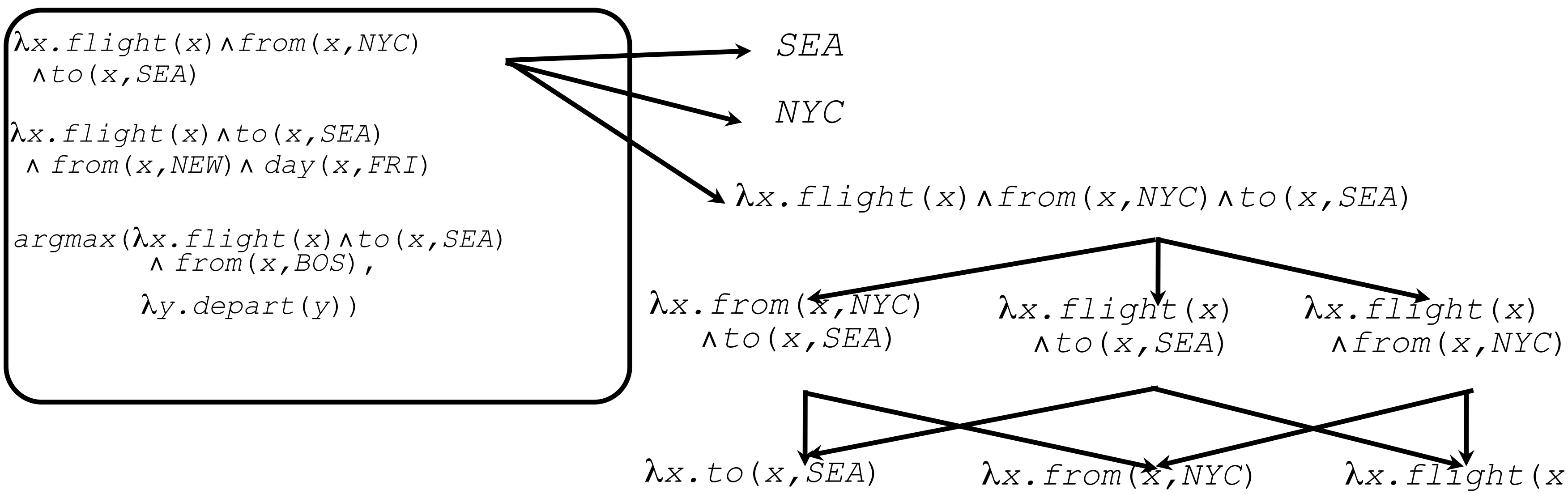
Context:



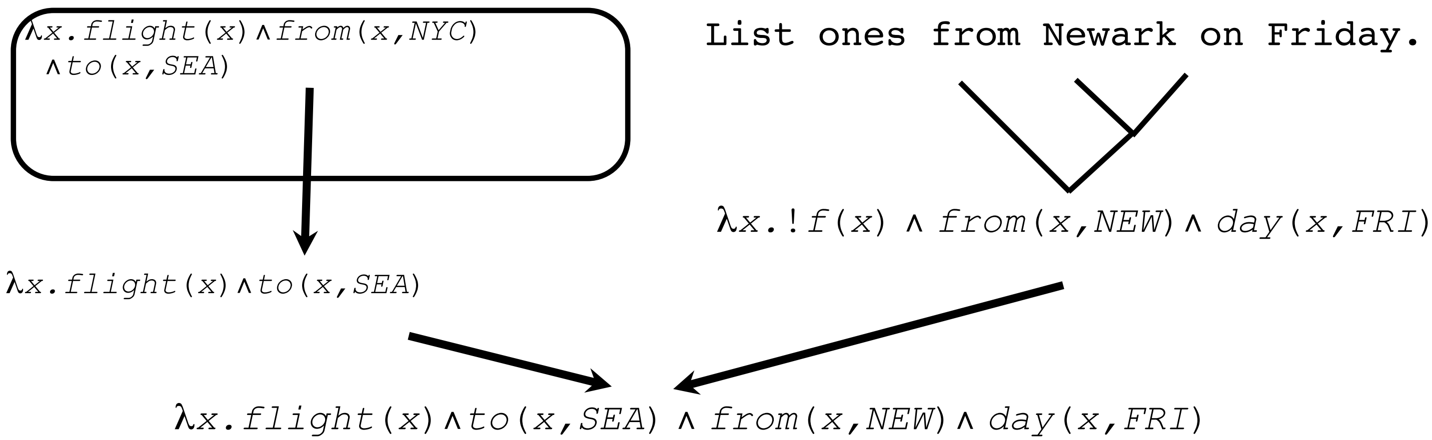
Step 2: Selecting from Context

For each logical form in context,
enumerate `e` and `<e,t>` type subexpressions:

Context:



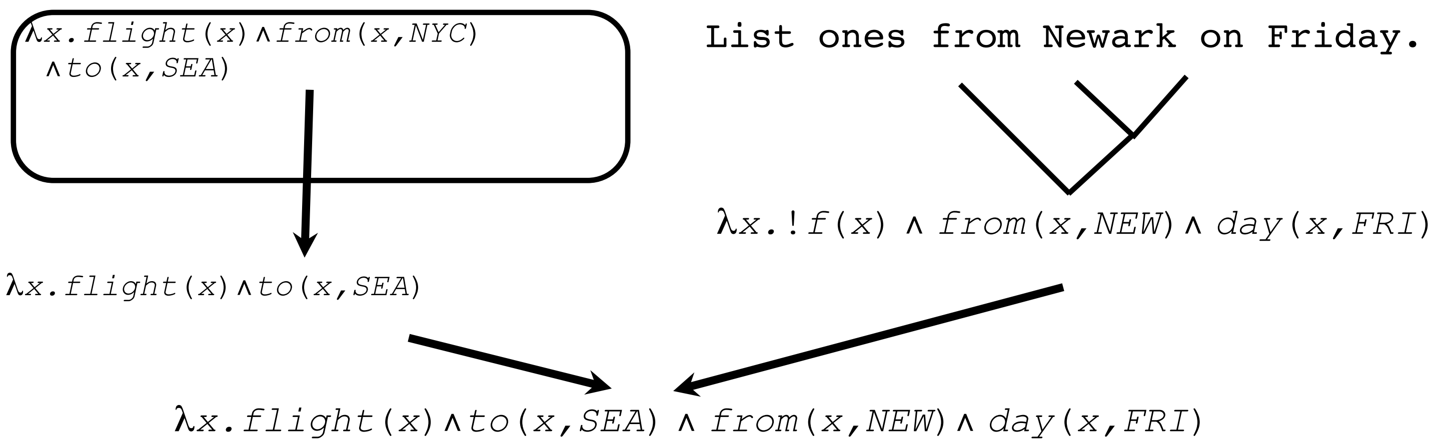
Step 2: Resolving References



For each reference:

- Select an expression from the context
- Substitute into current analysis

Derivations



Three step process:

- Step 1: Context-independent parsing
- Step 2: Resolve all references
- Step 3: Optionally, perform an elaboration

Step 3: Elaboration operations

Show me the latest flight from New York to Seattle.

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA),$
$\quad \lambda y.time(y))$

on Friday

Step 3: Elaboration operations

Show me the latest flight from New York to Seattle.

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA),$
$\lambda y.time(y))$

on Friday

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA) \wedge day(x,FRI),$
$\lambda y.time(y))$

Step 3: Elaboration operations

$argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NYC), \lambda y.time(y))$

on Friday

↓

$\lambda x.day(x,FRI)$

Step 3: Elaboration operations

$$argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NYC), \lambda y.time(y))$$

$$\lambda f.argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NYC) \wedge f(x), \lambda y.time(y))$$

on Friday

$$\lambda x.day(x,FRI)$$

Step 3: Elaboration operations

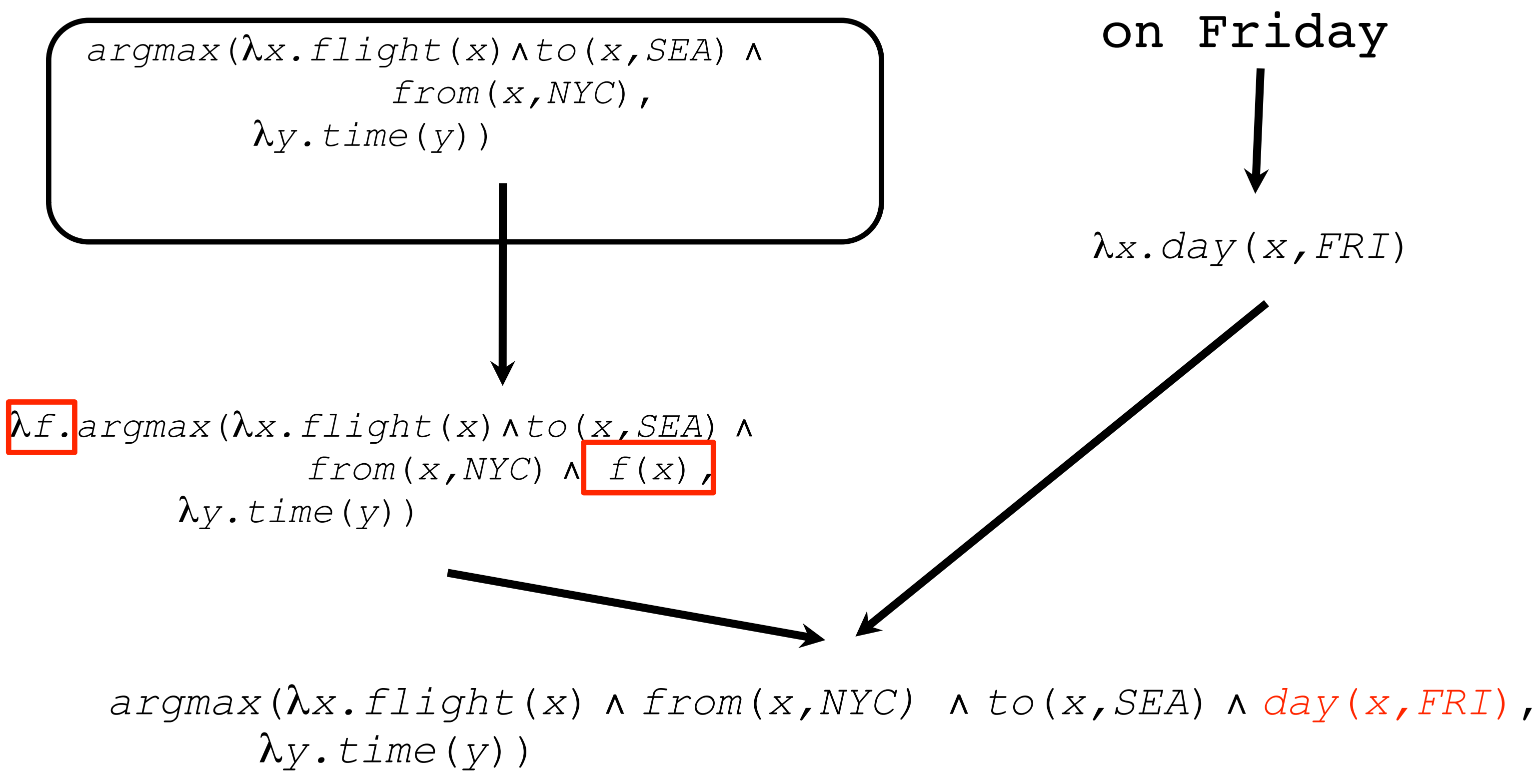
$argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge$
$from(x,NYC),$
$\lambda y.time(y))$

$\lambda f.argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge$
$from(x,NYC) \wedge f(x),$
$\lambda y.time(y))$

on Friday

$\lambda x.day(x,FRI)$

Step 3: Elaboration operations



Step 3: Elaboration operations

$argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NYC), \lambda y.time(y))$

on Friday

$\lambda x.day(x,FRI)$

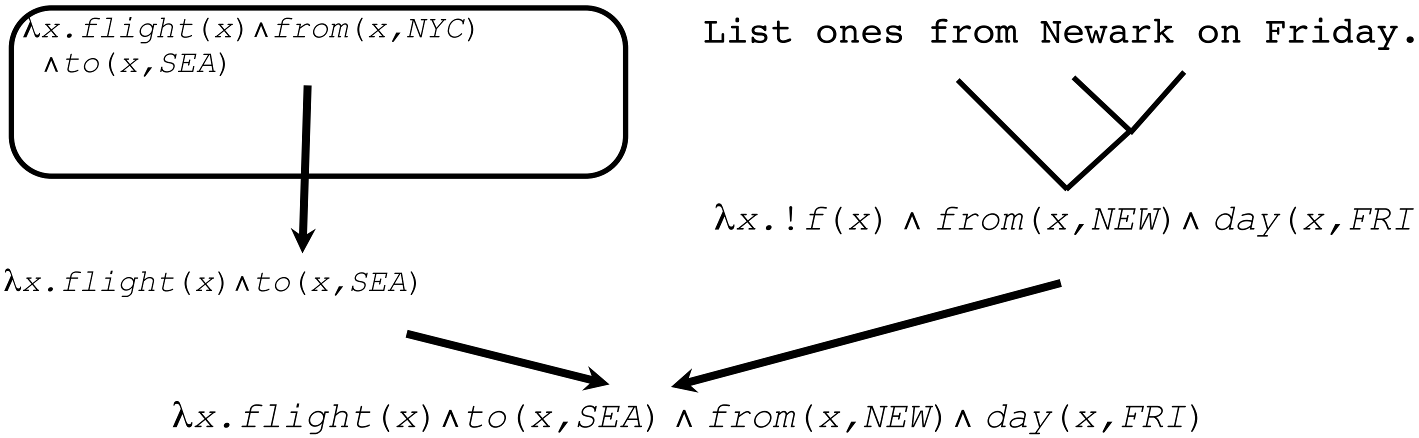
$\lambda f.argmax(\lambda x.flight(x) \wedge to(x,SEA) \wedge from(x,NYC) \wedge f(x), \lambda y.time(y))$

$argmax(\lambda x.flight(x) \wedge from(x,NYC) \wedge to(x,SEA) \wedge day(x,FRI), \lambda y.time(y))$

Possible elaborations:

- Potentially expand any embedded variable
- Can do deletions on elaboration function

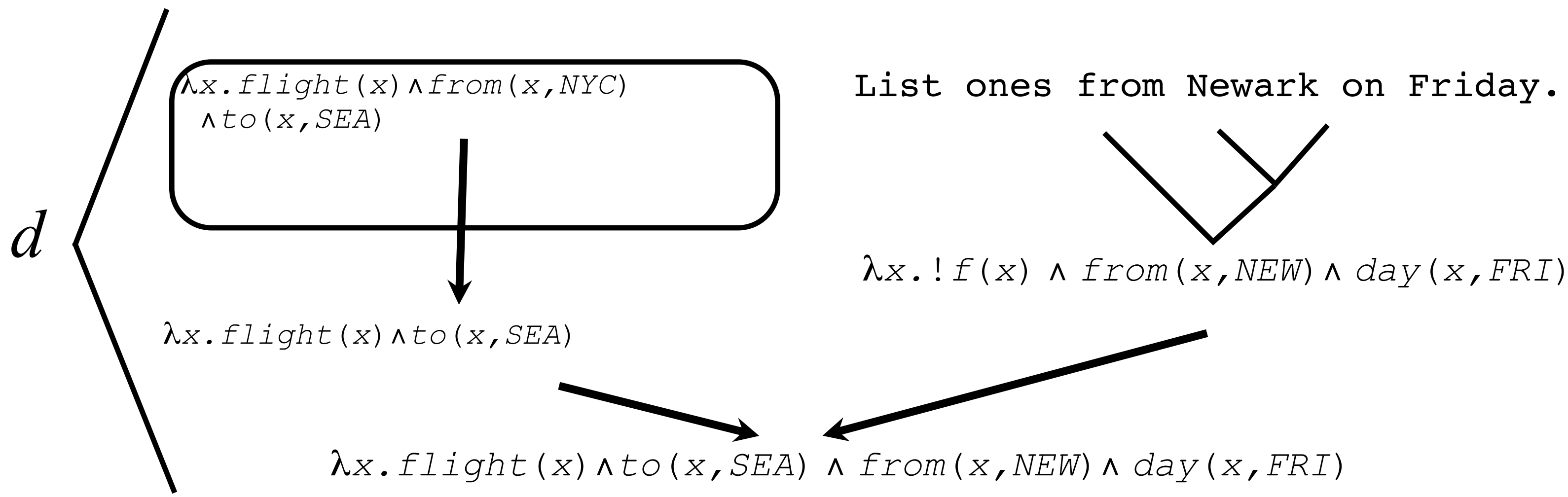
Derivations



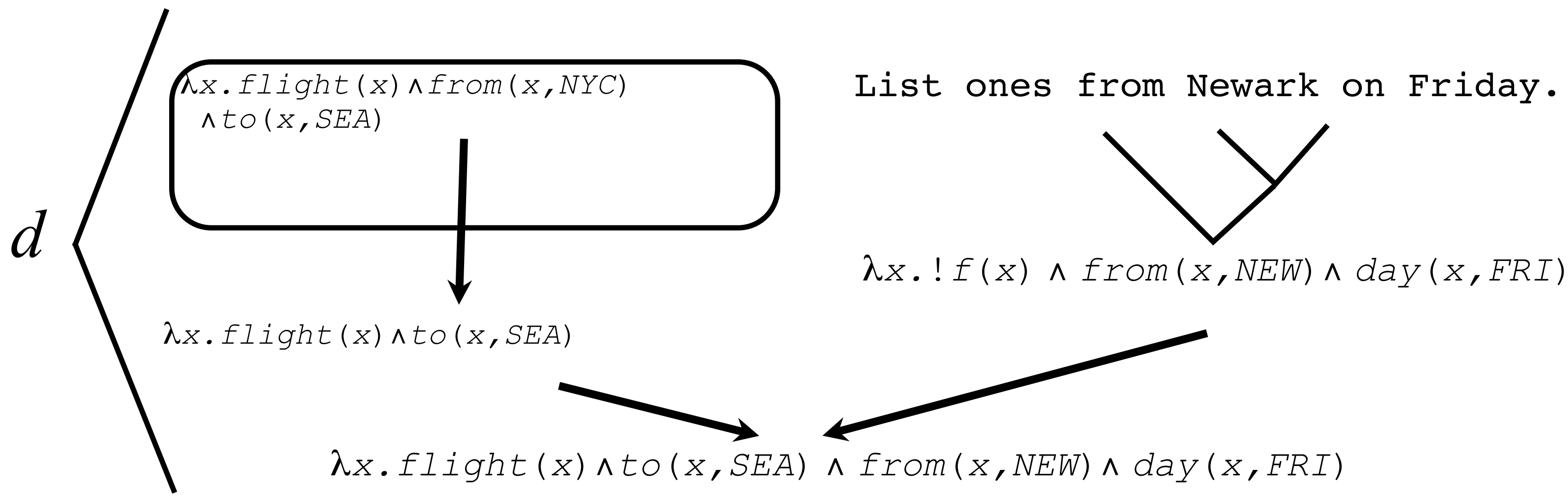
Three step process:

- Step 1: Context-independent parsing
- Step 2: Resolve all references
- Step 3: Optionally, perform an elaboration

Scoring Derivations



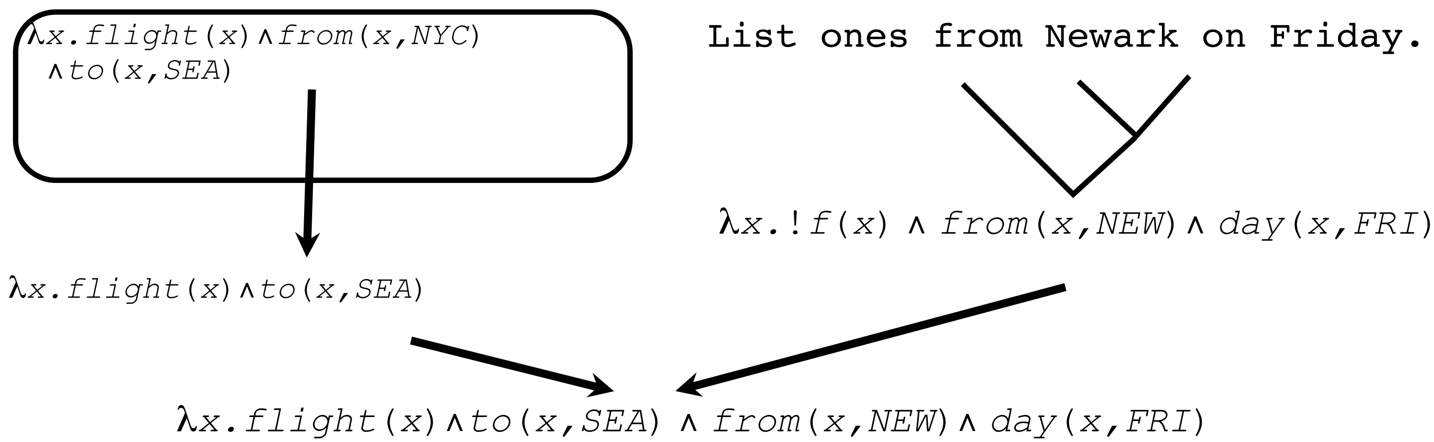
Scoring Derivations



Weighted linear model:

- Introduce features: $f(d)$
- Compute scores for derivations: $w \cdot f(d)$

Features for Derivations: $f(d)$



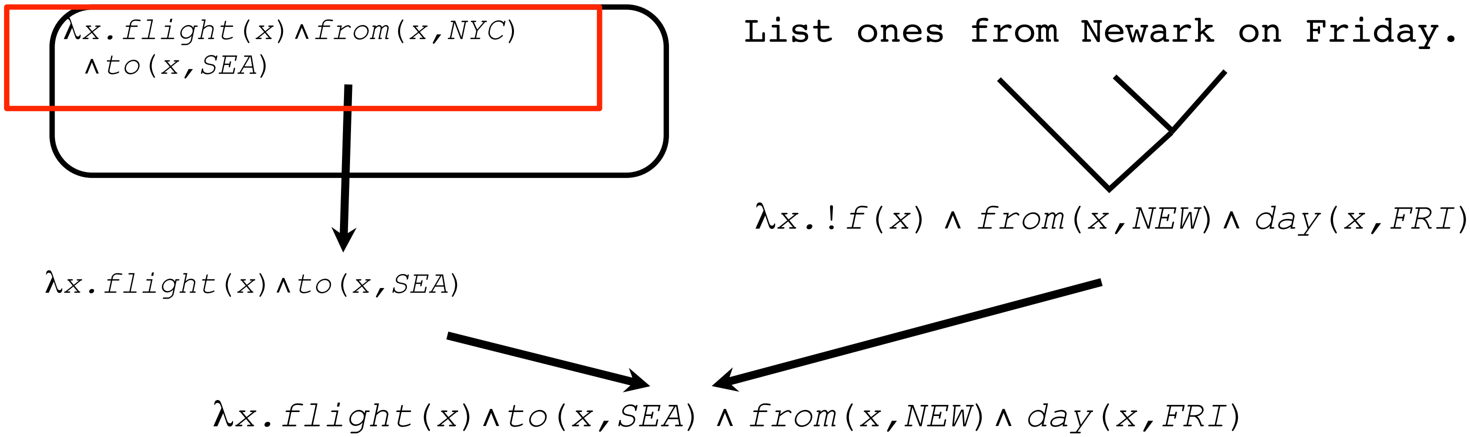
Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {*flight*, *from*, *to*, ...}
- Deletion indicators, for all pairs of predicates

{(*from*, *flight*), (*from*, *from*), (*from*, *to*), ...}

Features for Derivations: $f(d)$



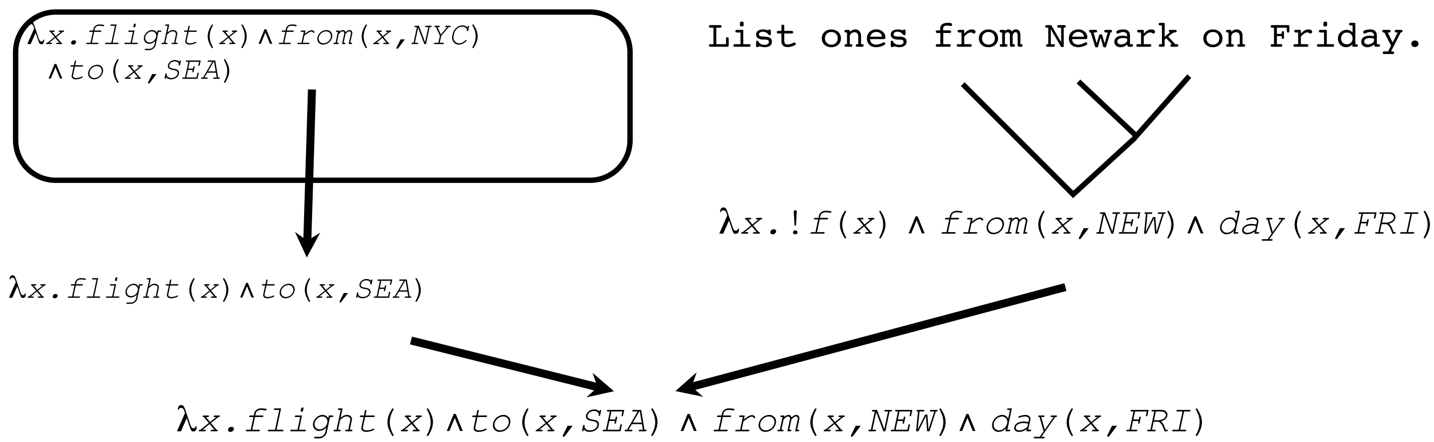
Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {*flight*, *from*, *to*, ...}
- Deletion indicators, for all pairs of predicates

{(*from*, *flight*), (*from*, *from*), (*from*, *to*), ...}

Features for Derivations: $f(d)$



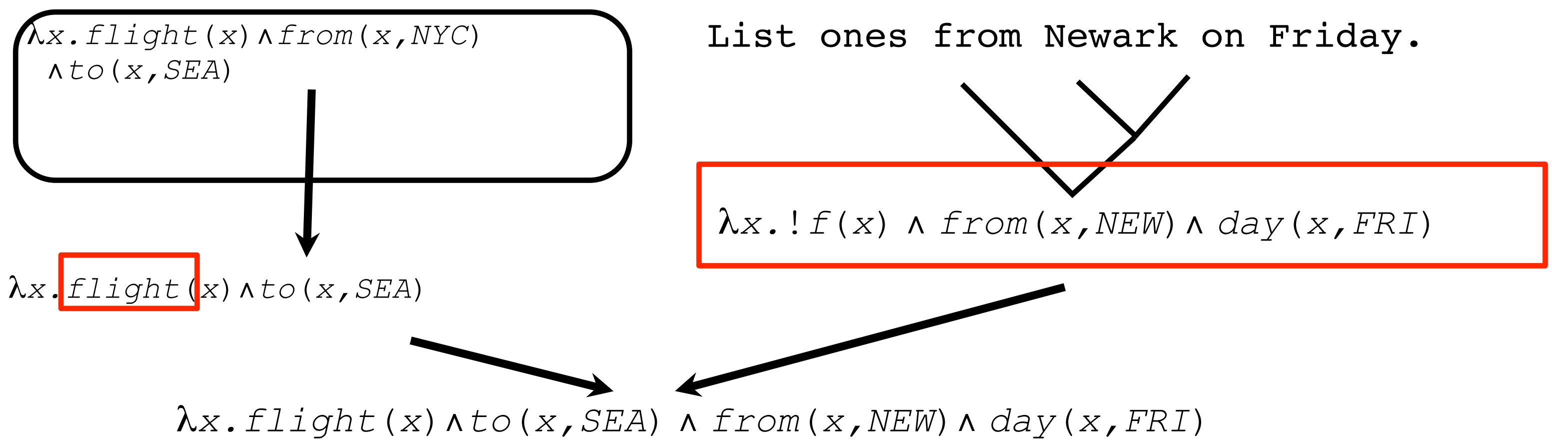
Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {flight, from, to, ...}
- Deletion indicators, for all pairs of predicates

{(from, flight), (from, from), (from, to), ...}

Features for Derivations: $f(d)$



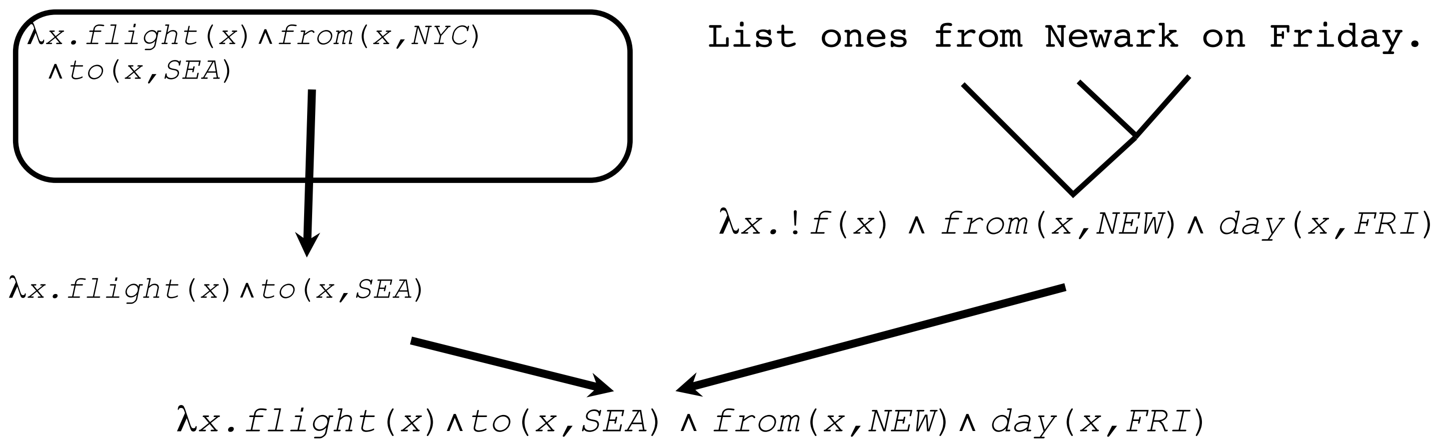
Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {`flight`, `from`, `to`, ...}
- Deletion indicators, for all pairs of predicates

{(`from`, `flight`), (`from`, `from`), (`from`, `to`), ...}

Features for Derivations: $f(d)$



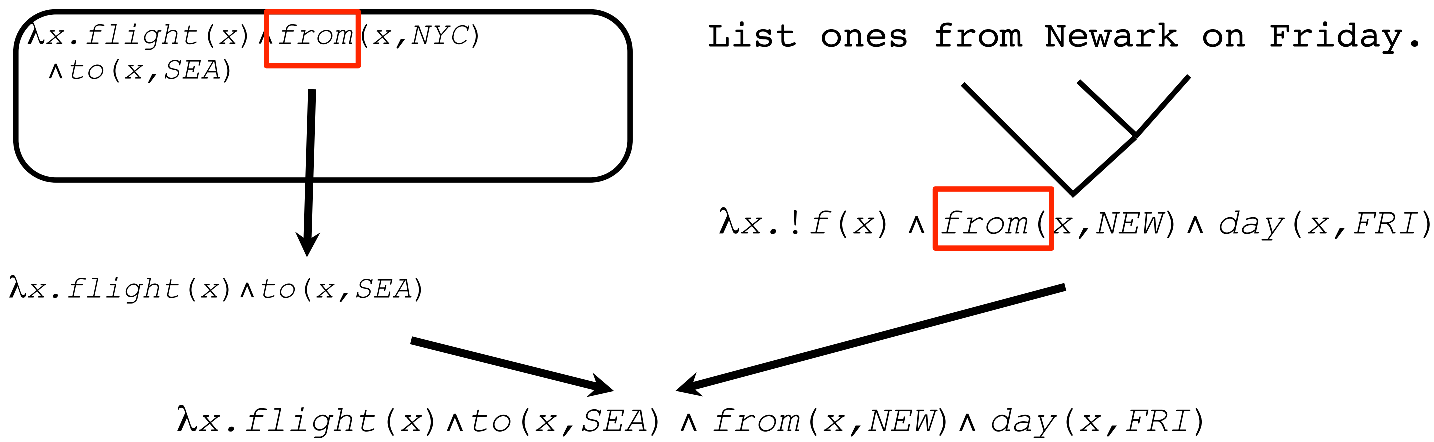
Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {flight, from, to, ...}
- Deletion indicators, for all pairs of predicates

{(from, flight), (from, from), (from, to), ...}

Features for Derivations: $f(d)$



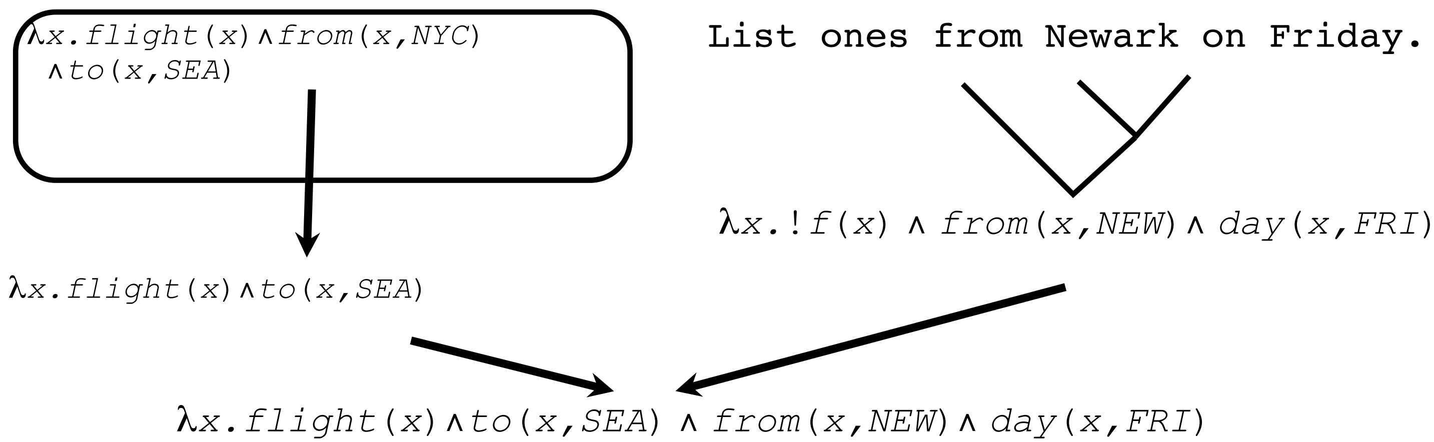
Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {*flight*, *from*, *to*, ...}
- Deletion indicators, for all pairs of predicates

{(*from*, *flight*), (*from*, *from*), (*from*, *to*), ...}

Features for Derivations: $f(d)$



Parsing features: set from Zettlemoyer and Collins (2007)

Context features:

- Distance indicators, for integers (0,1,2,...)
- Copy indicators, for all predicates {`flight`, `from`, `to`, ...}
- Deletion indicators, for all pairs of predicates

{(`from`, `flight`), (`from`, `from`), (`from`, `to`), ...}

Inference and Learning

Two computations:

- Best derivation:

$$d^* = \arg\max_{d} w \cdot f(d)$$

- Best derivation with final logical form z :

$$d' = \arg\max_{d \ s.t.\ L(d)=z} w \cdot f(d)$$

We use a beam search algorithm.

Inference and Learning

Two computations:

- Best derivation:

$$d^* = \arg\max_d \; w \cdot f(d)$$

- Best derivation with final logical form z :

$$d' = \arg\max_{d \; s.t. \; L(d)=z} w \cdot f(d)$$

We use a beam search algorithm.

Learning:

- Hidden variable version of the structured perceptron algorithm [Liang et al., 2006] [Zettlemoyer & Collins, 2007]

Inputs: Training set $\{I_i \mid i = 1 \ldots n\}$ of interactions. Each interaction $I = \{(w_{i,j}, z_{i,j}) \mid j = 1 \ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Output: Parameters w.

Inputs: Training set $\{I_i \mid i = 1 \ldots n\}$ of interactions. Each interaction $I = \{(w_{i,j}, z_{i,j}) \mid j = 1 \ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Computation:

For $t = 1 \ldots T$, $i = 1 \ldots n$: (Iterate interactions)

Set $C = \{\}$ (Reset Context)

For $j = 1 \ldots n_i$: (Iterate training examples)

Output: Parameters w.

Inputs: Training set $\{I_i \mid i = 1 \ldots n\}$ of interactions. Each interaction $I = \{(w_{i,j}, z_{i,j}) \mid j = 1 \ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Computation:

For $t = 1 \ldots T,\ i = 1 \ldots n$: (Iterate interactions)

Set $C = \{\}$ (Reset Context)

For $j = 1 \ldots n_i$: (Iterate training examples)

Step 3: Update context: Append $z_{i,j}$ to C

Output: Parameters w.

Inputs: Training set $\{I_i \mid i = 1 \ldots n\}$ of interactions. Each interaction $I = \{(w_{i,j}, z_{i,j}) \mid j = 1 \ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Computation:

For $t = 1 \ldots T, i = 1 \ldots n$: (Iterate interactions)

Set $C = \{\}$ (Reset Context)

For $j = 1 \ldots n_i$: (Iterate training examples)

Step 1: Check Correctness

- Find best analysis: $d^* = \arg\max_d w \cdot f(d)$
- If correct: $L(d^*) == z_{i,j}$, go to the Step 3.

Step 3: Update context: Append $z_{i,j}$ to C

Output: Parameters w.

Inputs: Training set $\{I_i \mid i = 1 \ldots n\}$ of interactions. Each interaction $I = \{(w_{i,j}, z_{i,j}) \mid j = 1 \ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Computation:

For $t = 1 \ldots T$, $i = 1 \ldots n$: (Iterate interactions)

Set $C = \{\}$ (Reset Context)

For $j = 1 \ldots n_i$: (Iterate training examples)

Step 1: Check Correctness

- Find best analysis: $d^* = \arg\max_d w \cdot f(d)$
- If correct: $L(d^*) == z_{i,j}$, go to the Step 3.

Step 2: Update Parameters

- Find best correct analysis: $d' = \arg \max_{d \text{ s.t. } L(d) = z_{i,j}} w \cdot f(d)$
- Update parameters: $w = w + f(d') - f(d^*)$

Step 3: Update context: Append $z_{i,j}$ to C

Output: Parameters w.

Inputs: Training set $\{I_i \mid i=1\ldots n\}$ of interactions. Each interaction $I=\{(w_{i,j}, z_{i,j}) \mid j=1\ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Computation:

For $t=1\ldots T,\ i=1\ldots n$: (Iterate interactions)

Set $C=\{\}$ (Reset Context)

For $j=1\ldots n_i$: (Iterate training examples)

Step 1: Check Correctness

- Find best analysis: $d^* = \arg\max_d w \cdot f(d)$
- If correct: $L(d^*) == z_{i,j}$, go to the Step 3.

Step 2: Update Parameters

- Find best correct analysis: $d' = \arg\max_{d \text{ s.t. } L(d)=z_{i,j}} w \cdot f(d)$
- Update parameters: $w = w + f(d') - f(d^*)$

Step 3: Update context: Append $z_{i,j}$ to C

Output: Parameters w.

Inputs: Training set $\{I_i \mid i = 1 \ldots n\}$ of interactions. Each interaction $I = \{(w_{i,j}, z_{i,j}) \mid j = 1 \ldots n_i\}$ is a sequence of sentences and logical forms. Initial parameters w. Number of iterations T.

Computation:

For $t = 1 \ldots T, i = 1 \ldots n$: (Iterate interactions)

Set $C = \{\}$ (Reset Context)

For $j = 1 \ldots n_i$: (Iterate training examples)

Step 1: Check Correctness

- Find best analysis: $d^* = \arg\max_d w \cdot f(d)$
- If correct: $L(d^*) == z_{i,j}$, go to the Step 3.

Step 2: Update Parameters

- Find best correct analysis: $d' = \arg \max_{d \text{ s.t. } L(d) = z_{i,j}} w \cdot f(d)$
- Update parameters: $w = w + f(d') - f(d^*)$

Step 3: Update context: Append $z_{i,j}$ to C

Output: Parameters w.

Evaluation

- Domain: ATIS travel database queries
  - 399 training interactions (3813 sentences)
  - 127 test interactions (826 sentences)
- Comparison: previous state-of-the-art [Miller et al. 1996]
  - requires full annotation of all syntactic, semantic, and context-resolution decisions
  - decision tree learning

Miller et al. [1996]

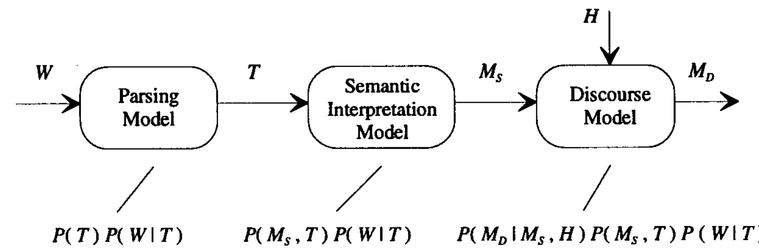


Figure 1: Overview of statistical processing.

Step 1: Semantic parsing

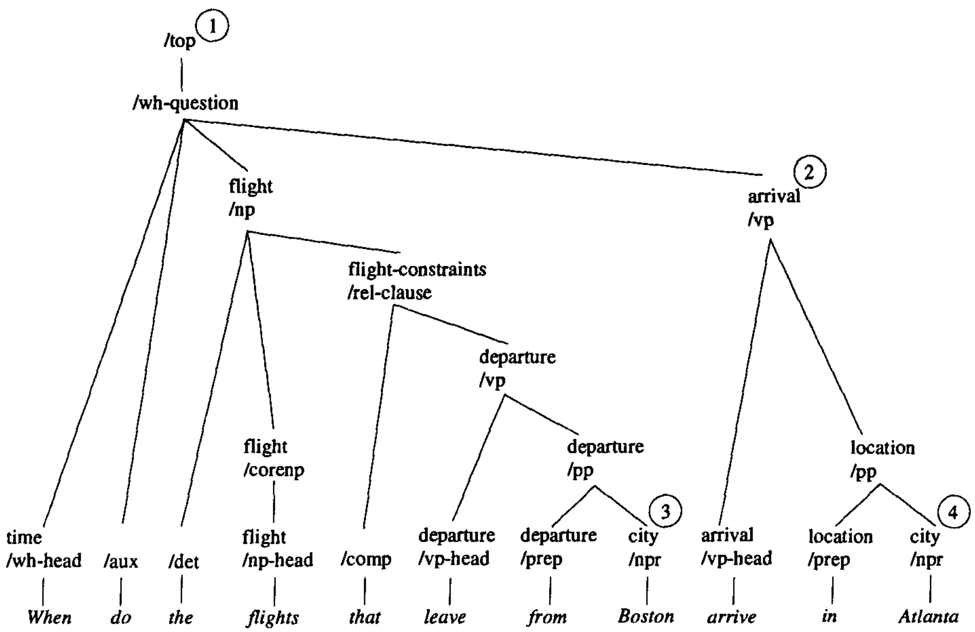


Figure 2: A sample parse tree.

Step 2: Select frame and fill slot values

Air-Transportation

Show: (Arrival-Time)

Origin: (City "Boston")

Destination: (City "Atlanta")

Figure 3: A sample semantic frame.

Step 3: Optionally copy slot values from previous frames

Evaluation

- Domain: ATIS travel database queries
  - 399 training interactions (3813 sentences)
  - 127 test interactions (826 sentences)
- Comparison: previous state-of-the-art [Miller et al. 1996]
- Metric: accuracy recovering fully correct meanings

Evaluation

- Domain: ATIS travel database queries
  - 399 training interactions (3813 sentences)
  - 127 test interactions (826 sentences)
- Comparison: previous state-of-the-art [Miller et al. 1996]
- Metric: accuracy recovering fully correct meanings
- Results:
  - improved accuracy: 78.4% => 83.7%
  - less engineering effort: only annotated final meanings

Varying the Length of a Context Window M

ATIS Development Set:

Context Length	Accuracy
M=0	45.4
M=1	79.8
M=2	81.0
M=3	**82.1**
M=4	81.6
M=10	81.4

Example Learned Feature Weights

Negative weights:

- Distance features: (1,2,3,...)

Positive weights:

- Copy features: `flight, from, to`
- Deletion features: (`from, from`), (`nonstop, connect`), (`during-day, time`)

Summary

$\lambda x.flight(x) \wedge from(x, NYC) \wedge to(x, SEA)$

$\lambda x.flight(x) \wedge to(x, SEA) \wedge from(x, NEW) \wedge day(x, FRI)$

Show me the cheapest?

F

$argmax(\lambda x.flight(x) \wedge from(x, NEW) \wedge to(x, SEA) \wedge day(x, FRI), \lambda y.cost(y))$

Key challenges:

- Structured input and output, hidden structure not annotated

Solution:

- Analysis: two-stage approach
- Learn: how to incorporate meaning from the context

Bootstrapping Semantic Parsers from Conversations

[Artzi & Zettlemoyer, EMNLP 2011]

Task Oriented Dialog

SYSTEM	how can i help you ?
USER	i ' d like to go from richmond to atlantic city on june seventh
SYSTEM	leaving richmond on thursday, june seventh arriving in atlantic city leaving at what time ?
USER	eight a. m.
SYSTEM	leaving in the morning what airline would you prefer ?
USER	u. s. air
SYSTEM	i currently have you leaving richmond on thursday, june seventh in the morning arriving in atlantic city flying on u. s. air . is this correct ?
USER	yes
SYSTEM	o . k . then . where would you like to go next ?
USER	richmond

[conversation continues]

Dialog and Semantics

- Many dialog systems use semantic parsers.
- Supervised learning of such parsers is expensive.
- Can computers learn through raw conversations?

Learning from Conversations

- Human-machine dialogs aim to achieve a **goal**.
- A well-defined goal allows the system to **recover** from failures.

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

✗ System fails

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? *System fails*

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city?

USER: atlanta georgia

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city?

USER: atlanta georgia

SYSTEM: leaving atlanta. going to which city?

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city?

USER: atlanta georgia

SYSTEM: leaving atlanta. going to which city?

USER: london

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city?

USER: atlanta georgia

SYSTEM: leaving atlanta. going to which city?

USER: london

SYSTEM: arriving in london england. what date would you like to depart atlanta?

System Recovery

SYSTEM: how can i help you?

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city?

USER: atlanta georgia

SYSTEM: leaving atlanta. going to which city?

USER: london

SYSTEM: arriving in london england. what date would you like to depart atlanta?

USER: september twenty fourth in the early evening

[conversation continues]

Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]

Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]

System utterance

Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.\mathit{from}(\mathit{fl}, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $\mathit{from}(\mathit{fl}, \mathit{ATL})$) going to which city? (ASK: $\lambda x.\mathit{to}(\mathit{fl}, x)$)

USER: london

[conversation continues]

System utterance

Natural Language

Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]

System utterance

Natural Language

Speech Act

Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]

System utterance

Natural Language | Speech Act | Logical Meaning

Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

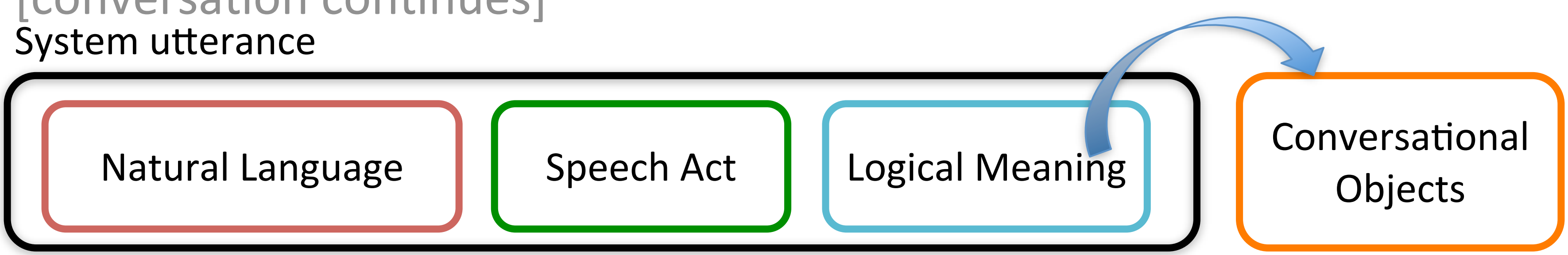
SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]



Automatically Annotated Logs

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

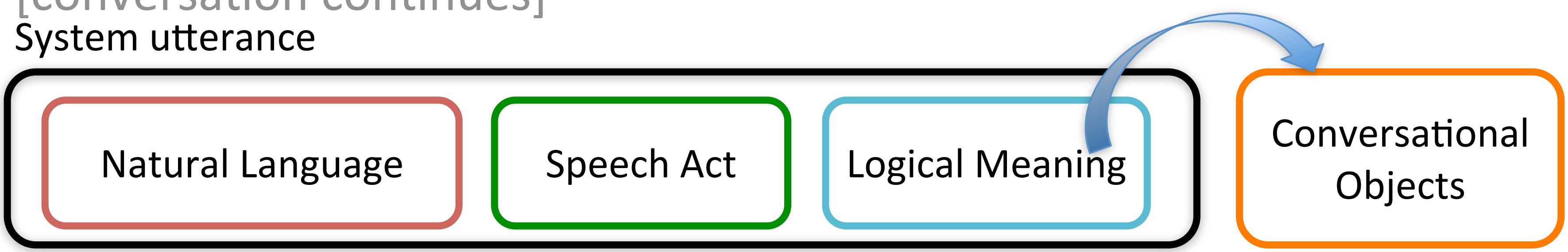
SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]



One Training Example

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]

Supervised Case

i want to fly from new york to los angeles

$$\lambda x.from(x, NYC) \wedge to(x, LA)$$

Loss Signal Instead of Labels

$$\mathcal{L} : \langle z, \mathcal{C} \rangle \rightarrow \mathbb{R}$$

i want to fly from new york to los angeles

Loss Signal Instead of Labels

$\mathcal{L} : \langle z, \mathcal{C} \rangle \to \mathbb{R}$

i want to fly from new york to los angeles

$$\lambda x.from(x, NYC) \land to(x, LA)$$

$$\lambda x.from(x, NYC) \land from(x, LA)$$

$$\lambda x.from(x, LA)$$

$$\lambda x[].to(x[0], NYC) \land from(x[1], LA)$$

$$\ldots$$

Loss Signal Instead of Labels

$$\mathcal{L} : \langle z, \mathcal{C} \rangle \rightarrow \mathbb{R}$$

i want to fly from new york to los angeles

1 -1.8 $\lambda x.from(x, NYC) \wedge to(x, LA)$

4 2.0 $\lambda x.from(x, NYC) \wedge from(x, LA)$

2 -0.9 $\lambda x.from(x, LA)$

3 1.5 $\lambda x[].to(x[0], NYC) \wedge from(x[1], LA)$

$\ldots$

Computing Conversation Loss

1. Select the relevant subset of system utterances
2. Align the logical expression to content from selected system utterances
3. Score the alignment

Conversation Loss

SYSTEM: how can i help you (OPEN_TASK)

USER: i want to fly from new york to los angeles

SYSTEM: from new york. (CONFIRM: $from(fl, NYC)$) to where? (ASK: $\lambda x.to(fl, x)$)

USER: los angeles

SYSTEM: flying to los angeles. (CONFIRM: $to(fl, LA)$) where do you want to go from los angeles? ? (ASK: $\lambda x.to(fl2, x) \wedge from(fl2, LA)$)

USER: back to new york

SYSTEM: from los angeles to new york (CONFIRM: $from(fl2, LA) \wedge to(fl2, NYC)$)

SYSTEM: i have a direct flight from new york to los angeles for $480. ok? (OFFER:

[conversation continues]

Segmenting

SYSTEM: how can i help you (OPEN_TASK)

USER: i want to fly from new york to los angeles

SYSTEM: from new york. (CONFIRM: $from(fl, NYC)$) to where? (ASK: $\lambda x.to(fl, x)$)

USER: los angeles

SYSTEM: flying to los angeles. (CONFIRM: $to(fl, LA)$) where do you want to go from los angeles? ? (ASK: $\lambda x.to(fl2, x) \wedge from(fl2, LA)$)

USER: back to new york

SYSTEM: from los angeles to new york (CONFIRM: $from(fl2, LA) \wedge to(fl2, NYC)$)

SYSTEM: i have a direct flight from new york to los angeles for $480. ok? (OFFER:

[conversation continues]

Extract Properties

SYSTEM: how can i help you (OPEN_TASK)

USER: i want to fly from new york to los angeles

SYSTEM: from new york. (CONFIRM: *from*(*fl*, *NYC*)) to where? (ASK: $\lambda x.to(fl, x)$)

USER: los angeles

SYSTEM: flying to los angeles. (CONFIRM: *to*(*fl*, *LA*)) where do you want to go from los angeles? ? (ASK: $\lambda x.to(fl2, x) \wedge from(fl2, LA)$)

USER: back to new york

SYSTEM: from los angeles to new york (CONFIRM: $from(fl2, LA) \wedge to(fl2, NYC)$)

SYSTEM: i have a direct flight from new york to los angeles for $480. ok? (OFFER:

[conversation continues]

Property

(*from*, *fl*, *NYC*)

(*to*, *fl*, *LA*)

(*from*, *fl2*, *LA*)

(*to*, *fl2*, *NYC*)

Extract Properties

SYSTEM: how can i help you (OPEN_TASK)

USER: i want to fly from new york to los angeles

SYSTEM: from new york. (CONFIRM: *from(fl, NYC)*) to where? (ASK: $\lambda x.to(fl, x)$)

USER: los angeles

SYSTEM: flying to los angeles. (CONFIRM: *to(fl, LA)*) where do you want to go from los angeles? ? (ASK: $\lambda x.to(fl2, x) \wedge from(fl2, LA)$)

USER: back to new york

SYSTEM: from los angeles to new york (CONFIRM: $from(fl2, LA) \wedge to(fl2, NYC)$)

SYSTEM: i have a direct flight from new york to los angeles for $480. ok? (OFFER:

[conversation continues]

Property	Loss
(*from*, *fl*, *NYC*)	-0.5
(*to*, *fl*, *LA*)	-0.83
(*from*, *fl2*, *LA*)	-0.16
(*to*, *fl2*, *NYC*)	-0.16

Comparing Two Candidates

USER:	i want to fly from new york to los angeles

$$\lambda x.from(x, NYC) \wedge to(x, LA)$$

Alignment: $x \rightarrow fl$

$$Loss = -0.5 - 0.83 = -1.33$$

Property	Loss
(*from, fl, NYC*)	-0.5
(*to, fl, LA*)	-0.83
(*from, fl2, LA*)	-0.16
(*to, fl2, NYC*)	-0.16

$$\lambda x[].from(x[0], NYC) \wedge from(x[1], LA)$$

Alignment: $x[0] \rightarrow fl, x[1] \rightarrow fl2$

$$Loss = -0.5 - 0.16 = -0.66$$

Comparing Two Candidates

USER: i want to fly from new york to los angeles

Property	Loss
(from, fl, NYC)	-0.5
(to, fl, LA)	-0.83
(from, fl2, LA)	-0.16
(to, fl2, NYC)	-0.16

$$\lambda x.from(x, NYC) \wedge to(x, LA)$$

$$\text{Alignment: } x \rightarrow fl$$

$$Loss = -0.5 - 0.83 = -1.33$$

$$\lambda x[].from(x[0], NYC) \wedge from(x[1], LA)$$

$$\text{Alignment: } x[0] \rightarrow fl, x[1] \rightarrow fl2$$

$$Loss = -0.5 - 0.16 = -0.66$$

Learning Algorithm

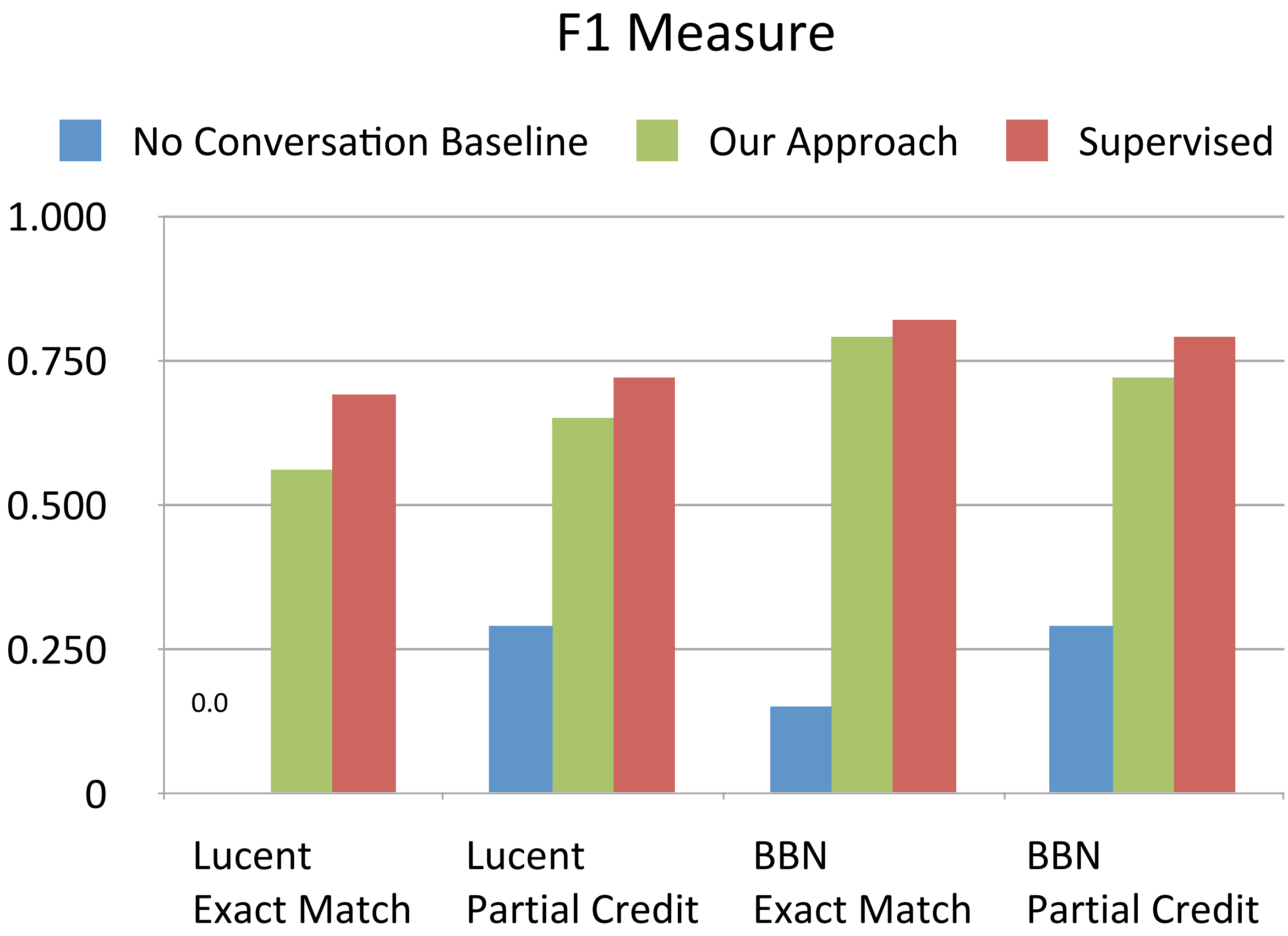
- Online
- Loss-driven
- 2 steps:
  - Lexical generation
  - Parameter update [Singh-Miller and Collins 2007]

DARPA Communicator

- Raw conversational logs [Walker et al. 2002]
- Annotated system utterances

	Lucent		BBN	
	Train	**Test**	**Train**	**Test**
Conversations	144	70	71	91
Context Independent Utterances	208	96	67	67

Results: Evaluation Data



Contributions

- Learning to do context-dependent semantic parsing
  - Complex, hidden variable problem
  - Learn distribution over derivations
- Learning from conversation logs
  - Loss driven, requiring no annotations
  - First step towards autonomous, self-improving dialog systems

Current Work: Self-learning dialog systems

Goal: learn full dialog semantic parser the produces context-dependent analyses

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening I would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? *(ASK:$\lambda x.from(fl, x)$)*

USER: atlanta georgia.

SYSTEM: leaving atlanta. *(CONF:$from(fl,ATL)$)* going to which city ? *(ASK:$\lambda x.to(fl, x)$)*

USER london

SYSTEM: arriving in *(CONF:$to(fl, LON)$)* what date would to depart Atlanta? *(ASK:$\lambda x.from(fl, ATL) \wedge depart\text{-} date(fl, x)$)*

USER september twenty fourth in the early evening

[conversation continues]

Current Work: Mapping Speech to Logical From

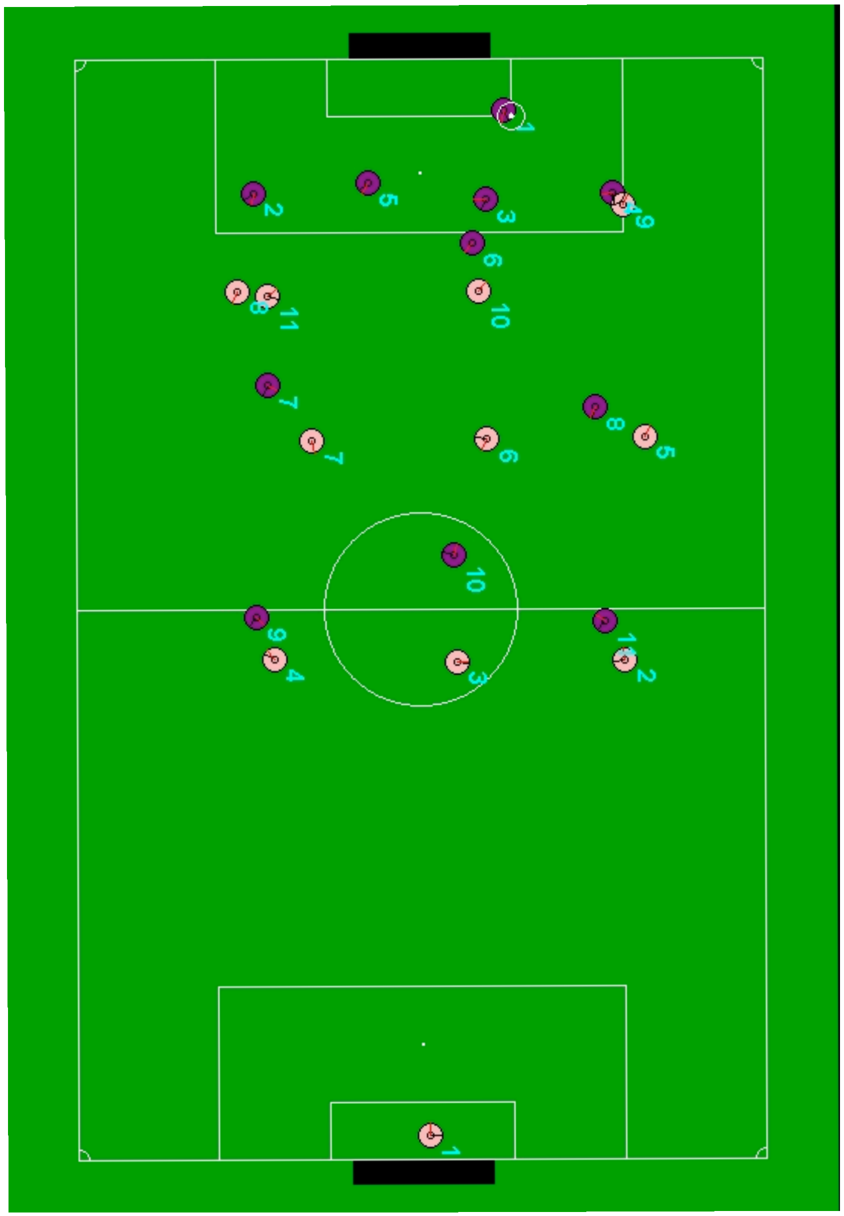
`Uh, flights New York to Singapore, sure`

ACCEPT: $\lambda x.\mathit{flight}(x) \wedge \mathit{from}(x, NYC) \wedge \mathit{to}(x, SIN)$

Current Work: Leaning Grounded Language

Challenge: Learn to sportscast, given only text and the game log

```
Purple10 is rushing down the
    field with only three
    defenders
Purple10 passes out front to
    Purple9 near the side
Purple9 passes back to Purple10
    in the middle
Purple10 again has a good chance
    to score a goal here
Purple10 dribbles toward the goal
Pink3 tries to stay in front of
    Purple10
Purple10 passes to Purple9 on the
    side while getting open
....
```



A Joint Model for Naming Objects

These are the ones that are not blue

Semantic Parsing

$\lambda x. \neg color(x, blue)$

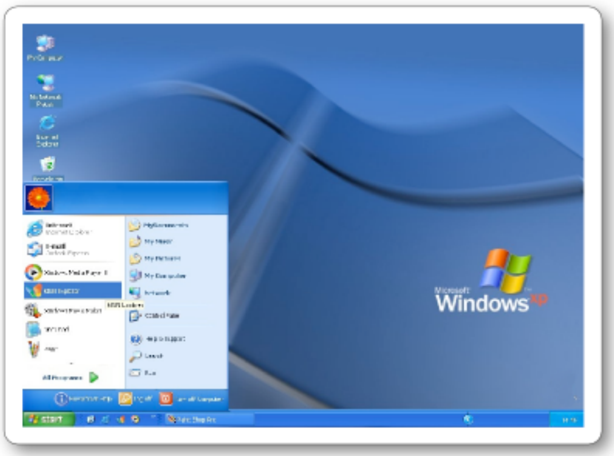
Grounded Query

Attribute Classification

Blue Green Round Broccoli

Future: General language use in grounded settings

Conversational interaction in simulated environments:



- Can gather user input: *Which printer do you want to use?*
- Can help with learning: *Can you show me how to X?*

Learning through explanation in robotic environments:

Can we teach the robot to play?

- *This is a pawn.*
- *Pawns can move forward one square at a time.*
- *unless it is the first move, then they can ...*

Learning About and From Context in Semantic Parsing

special thanks to

Yoav Artzi, Tom Kwiatkowski, Sharon Goldwater, Mark Steedman, Adrienne Wang, Mark Yatskar

for more info:

http://www.cs.washington.edu/homes/lsz/