Bootstrapping Semantic Parsers from Conversations

Yoav Artzi and Luke Zettlemoyer
University of Washington

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]
Task Oriented Dialog

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
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]
Dialog and Semantics

• Many dialog systems use semantic parsers.
Dialog and Semantics

• Many dialog systems use semantic parsers.
• Supervised learning of such parsers is expensive.
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.
Learning from Conversations

• Human-machine dialogs aim to achieve a **goal**.
• A well-defined goal allows the system to **recover** from failures.
how can i help you?

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]
Complex request
Complex request

System fails
Complex request

↓

System fails

↓

Goal
Complex request

↓

System fails

↓

Goal

←

Specific information
Complex request

System fails

Goal

Specific information

Clarification
Complex request

System fails

Goal

Specific information

Clarification

Rewording
Complex request

System fails

Goal

Specific information
Clarification
Rewording
Additional explanation
Complex request

System fails

Goal

Specific information

Clarification

Rewording

Additional explanation

Complete task
Complex request

System fails

Goal

Specific information

Clarification

Rewording

Additional explanation

Complete task

Learning signal
Noisy signal: premature end, goals change, incomplete recovery etc.
The Problem

• Learn:

\[ f : \text{sentence} \rightarrow \lambda\text{-calculus expression} \]
The Problem

• Learn:

\[ f : \text{sentence} \rightarrow \lambda\text{-calculus expression} \]

I want to go to Seattle on Friday

\[ \lambda x.\text{to}(x, \text{SEA}) \land \text{date}(x, \text{FRI}) \]
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: \( \lambda x. \text{from}(fl, x) \))
USER:  atlanta georgia
SYSTEM:  leaving atlanta. (CONFIRM: \( \text{from}(fl, ATL) \)) going to which city? (ASK: \( \lambda x. \text{to}(fl, x) \))
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: \( \lambda x. \text{from}(fl, x) \))

**USER:** atlanta georgia

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

**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. 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
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: λx.from(fl, x))

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: from(fl, ATL)) going to which city? (ASK: λx.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: \(\lambda x.\text{from}(fl, x)\))**
**USER:** atlanta georgia

**SYSTEM:** leaving atlanta. **(CONFIRM \(\text{from}(fl, ATL)\))** going to which city? **(ASK: \(\lambda x.\text{to}(fl, x)\))**
**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: λx.from(fl, x))
USER: atlanta georgia
SYSTEM: leaving atlanta. (CONFIRM: from(fl, ATL)) going to which city? (ASK: λx.to(fl, x))
USER: london
[conversation continues]

System utterance

<table>
<thead>
<tr>
<th>Natural Language</th>
<th>Speech Act</th>
<th>Logical Meaning</th>
<th>Conversational Objects</th>
</tr>
</thead>
</table>


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: λx.from(fl, x))*
**USER:** atlanta georgia
**SYSTEM:** leaving atlanta. *(CONFIRM: from(fl, ATL))* going to which city? *(ASK: λx.to(fl, x))*
**USER:** london
[conversation continues]

System utterance

- Natural Language
- Speech Act
- Logical Meaning
- Conversational Objects
Training Set

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]
## Training Set

<table>
<thead>
<tr>
<th>SYSTEM:</th>
<th>how can i help you? (OPEN_TASK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER:</td>
<td>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</td>
</tr>
<tr>
<td>SYSTEM:</td>
<td>leaving what city? (ASK: ( \lambda x. \text{from}(fl, x) ))</td>
</tr>
<tr>
<td>USER:</td>
<td>atlanta georgia</td>
</tr>
<tr>
<td>SYSTEM:</td>
<td>leaving atlanta. (CONFIRM: ( \text{from}(fl, ATL) )) going to which city? (ASK: ( \lambda x. \text{to}(fl, x) ))</td>
</tr>
<tr>
<td>USER:</td>
<td>london</td>
</tr>
</tbody>
</table>

[conversation continues]
The Problem

• Learn:

\[ f : \text{sentence} \rightarrow \lambda\text{-calculus expression} \]

• Given logs of conversations
  – Automatically annotated system utterances
Our Approach

Conversational logs → Perceptron-like online learner → Loss function → Context-independent CCG parser

Seed lexicon
Our Approach

Conversational logs → Perceptron-like online learner → Context-independent CCG parser
Seed lexicon → Loss function

No direct evidence about the logical forms paired with user utterances
Related Work

Supervised semantic parsing with:
  - Machine Translation
  - Inducting Logic Programming
  - Prob. CFG Parsing
  - Prob. PDA
  - Support Vector Machines
  - Perceptron-style Learning

[Papineni et al. 2007; Ramaswamy, Kleindienst 2000; Wong, Mooney 2006, 2007; Matuszek et al. 2010]
  - [Zelle, Money 1996; Tang, Mooney 2000; Thompson, Mooney 2002]
    - [Miller et al. 2006; Ge, Mooney 2006]
      - [He, Young 2005, 2006]
        - [Kate, Mooney 2006; Nguyen et al. 2006]
          - [Zettlemoyer, Collins 2005, 2007]

Less supervision for database query:
  - Question-Answers Semi-supervision
  - Confidence-driven
  - Unsupervised

[Clarke et al. 2010, Liang et al. 2011]
  - [Goldwasser et al. 2011]
In This Work

• Develop an algorithm for bootstrapping semantic parsers from conversations
  – Loss-driven
  – Requires no labeling effort
• Demonstrate effective learning from conversations
• First step towards a long-term goal of autonomous self-learning dialog systems
Mapping Sentences to $\lambda$-Calculus

- Combinatory Categorial Grammar (CCG)
- Weighted Linear CCGs
## Combinatory Categorial Grammar (CCG)

<table>
<thead>
<tr>
<th>I want to go</th>
<th>to</th>
<th>Boston</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S/N$</td>
<td>$(N \setminus N)/NP$</td>
<td>$NP$</td>
</tr>
<tr>
<td>$\lambda f. f$</td>
<td>$\lambda y. \lambda f. \lambda x. f(x) \land to(x, y)$</td>
<td>$BOS$</td>
</tr>
<tr>
<td></td>
<td>$(N \setminus N)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\lambda f. \lambda x. f(x) \land to(x, BOS)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$N$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\lambda x. to(x, BOS)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\lambda x. to(x, BOS)$</td>
<td></td>
</tr>
</tbody>
</table>
Combinatory Categorial Grammar (CCG)

\[
\begin{array}{c}
\text{I want to go} \\
\frac{S/N}{\lambda f.f} \\
\text{to} \\
\frac{(N \setminus N)/NP}{\lambda y.\lambda f.\lambda x.f(x) \land to(x, y)} \\
\frac{(N \setminus N)\lambda f.\lambda x.f(x) \land to(x, BOS)}{N\lambda x.to(x, BOS)} \\
\frac{N\lambda x.to(x, BOS)}{S\lambda x.to(x, BOS)}
\end{array}
\]

Lexicon
Combinatory Categorial Grammar (CCG)

I want to go
\[ \frac{S/N}{\lambda f. f} \]

\[
\begin{align*}
&\quad \text{to} \\
&\quad \frac{(N \backslash N)/NP}{\lambda y. \lambda f. \lambda x. f(x) \land to(x, y)} \\
&\quad \frac{(N \backslash N)}{\lambda f. \lambda x. f(x) \land to(x, BOS)} \\
&\quad \frac{N}{\lambda x. to(x, BOS)} \\
&\quad \frac{S}{\lambda x. to(x, BOS)}
\end{align*}
\]

Boston
\[ \frac{NP}{BOS} \]

Lexicon

Combinators
Application, composition ...
Weighted Linear CCGs

• Given a log-linear model:
  – CCG lexicon $\Lambda$
  – Feature vector $f$
  – Weights $w$

• 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 $\Lambda$
So Far ...

• General overview of the problem and approach
Next ...

• Indirectly-supervised lexical generation
• Computing loss
• Our learning algorithm
• Evaluation on the DARPA Communicator corpus
Supervised Lexical Induction

Input Training Example

Output Lexicon

Sentence:  i’d like to go to los angeles
Logical form: $\lambda x. \text{to}(x, \text{LA})$

i’d := $S/N : \lambda x.x$
i’d := $(N \setminus N)/NP : \lambda f.\lambda x.\lambda y.f(x) \land \text{to}(x, y)$
go to := $(N \setminus N)/NP : \lambda f.\lambda x.\lambda y.f(x) \land \text{to}(x, y)$
to go to := $(N \setminus N)/NP : \lambda f.\lambda x.\lambda y.f(x) \land \text{to}(x, y)$
to los := $S/N : \lambda x.x$
...

[Zettlemoyer & Collins 2005]
Supervised Lexical Induction

Sentence:  
I’d like to go to Los Angeles

Logical form:  \( \lambda x. \text{to}(x, LA) \)

i’d := \( S/N : \lambda x.x \)

go to := \( (N\backslash N)/NP : \lambda f.\lambda x.\lambda y.f(x) \land \text{to}(x, y) \)

to go to := \( (N\backslash N)/NP : \lambda f.\lambda x.\lambda y.f(x) \land \text{to}(x, y) \)

to los := \( S/N : \lambda x.x \)

...[Zettlemoyer & Collins 2005]
Indirect-supervised Lexical Induction

Input Training Example

Sentence: I’d like to go to los angeles

Conversation: SYSTEM: how can I help you? (OPEN_TASK)
USER: I’d like to go to los angeles
SYSTEM: Leaving what city? (ASK: \lambda x. from (fl, x))
USER: New york
SYSTEM: Leaving new york. (CONFIRM: from(fl, NYC) going to which city? (ASK: \lambda x. to(fl, x))
USER: Los angeles
SYSTEM: Flying to los angeles. (CONFIRM: to(fl, LA))
[conversation continues]
Indirectly-supervised Lexical Induction

Input Training Example

Sentence:  i’d like to go to los angeles

Conversation:  

SYSTEM:  how can i help you? (OPEN_TASK)

USER:  i’d like to go to los angeles

SYSTEM:  leaving what city? (ASK: λx.from (fI, x))

[conversation continues]

Output Lexicon

i’d := S/N : λx.x

i’d := (N\N)/NP : λf.λx.λy.f(x) \land to(x, y)

i’d := (N\N)/NP : λf.λx.λy.f(x) \land from(x, y)

...
Indirectely-supervised Lexical Induction

Input Training Example

Sentence: i’d like to go to los angeles
Conversation: SYSTEM: how can i help you? (OPEN_TASK)
USER: i’d like to go to los angeles
SYSTEM: leaving what city? (ASK: λx.from (f₁, x))
[conversation continues]

Output Lexicon

• Overgeneralized
• Incomplete
Loss-guided Learning

- Conversational logs
- Seed lexicon
- Context-independent CCG parser
  - Lexicon, Parsing parameters
- Loss function
- Perceptron-like online learner
Supervised Case

i want to fly from new york to los angeles

\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \]
Loss Signal Instead of Labels

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

i want to fly from new york to los angeles
Loss Signal Instead of Labels

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

I want to fly from New York to Los Angeles

\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \]
\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{from}(x, \text{LA}) \]
\[ \lambda x. \text{from}(x, \text{LA}) \]
\[ \lambda x[]. \text{to}(x[0], \text{NYC}) \land \text{from}(x[1], \text{LA}) \]
\[ \ldots \]
Loss Signal Instead of Labels

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

i want to fly from new york to los angeles

-1.8 \( \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \)

2.0 \( \lambda x. \text{from}(x, \text{NYC}) \land \text{from}(x, \text{LA}) \)

-0.9 \( \lambda x. \text{from}(x, \text{LA}) \)

1.5 \( \lambda x[].\text{to}(x[0], \text{NYC}) \land \text{from}(x[1], \text{LA}) \)

\ldots
Loss Signal Instead of Labels

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

i want to fly from new york to los angeles

1 -1.8 \( \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \)

4 2.0 \( \lambda x. \text{from}(x, \text{NYC}) \land \text{from}(x, \text{LA}) \)

2 -0.9 \( \lambda x. \text{from}(x, \text{LA}) \)

3 1.5 \( \lambda x[]. \text{to}(x[0], \text{NYC}) \land \text{from}(x[1], \text{LA}) \)

...
Loss Decomposition

Conversational logs
Seed lexicon

Perceptron-like online learner

Context-independent CCG parser
Lexicon, Parsing parameters

Loss function

Domain constraints + Conversational feedback
Conversation Loss

• Provides rough indication of how well a logical expression represents a user utterance
• Based on system understanding
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: λx.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: λx.to(fl2, x)\from(fl2, LA))

**USER:** back to new york

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

**SYSTEM:** i have a direct flight from new york to los angeles for $480. ok? (OFFER: direct(fl)\from(fl, NYC)\to(fl,LA)\cost(fl, 480))

[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: \( \lambda x.\text{to}(f1, x) \))*

**USER:** los angeles

**SYSTEM:** flying to los angeles. *(CONFIRM: to(fl, LA)) where do you want to go from los angeles? ? *(ASK: \( \lambda x.\text{to}(f2, x) \land \text{from}(f2, LA) \))

**USER:** back to new york

**SYSTEM:** from los angeles to new york *(CONFIRM: from(f2, LA) \land \text{to}(f2, NYC))*

**SYSTEM:** i have a direct flight from new york to los angeles for $480. ok? *(OFFER: direct(f1) \land \text{from}(f1, NYC) \land \text{to}(f1, LA) \land \text{cost}(f1, 480))

[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: direct(fl) \wedge from(fl, NYC) \wedge to(fl, LA) \wedge cost (fl, 480))

[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(\textit{fl}, \textit{NYC})) to where? *(ASK: \lambda x. to(\textit{fl}, x))*

**USER:** los angeles

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

**USER:** back to new york

**SYSTEM:** from los angeles to new york *(CONFIRM: from(\textit{fl2}, \textit{LA})\land to(\textit{fl2}, \textit{NYC}))*

**SYSTEM:** I have a direct flight from new york to los angeles for $480. ok? *(OFFER: direct(\textit{fl})\land from(\textit{fl}, \textit{NYC})\land to(\textit{fl}, \textit{LA})\land cost(\textit{fl}, 480))*

[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: λx.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: λx.to(fl2, x)\ from(fl2, LA))*

**USER:** back to new york

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

**SYSTEM:** i have a direct flight from new york to los angeles for $480. ok? *(OFFER: direct(fl)\ from(fl, NYC)\ to(fl, LA)\ cost (fl, 480))*

[conversation continues]
Given a Candidate Logical Expression

\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \]

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>(from, fl, NYC)</td>
<td>-0.5</td>
</tr>
<tr>
<td>(to, fl, LA)</td>
<td>-0.83</td>
</tr>
<tr>
<td>(from, fl2, LA)</td>
<td>-0.16</td>
</tr>
<tr>
<td>(to, fl2, NYC)</td>
<td>-0.16</td>
</tr>
</tbody>
</table>
Align

\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \]

Alignment: \( x \rightarrow fl \)

<table>
<thead>
<tr>
<th>Property</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>(from, fl, NYC)</td>
<td>-0.5</td>
</tr>
<tr>
<td>(to, fl, LA)</td>
<td>-0.83</td>
</tr>
<tr>
<td>(from, fl2, LA)</td>
<td>-0.16</td>
</tr>
<tr>
<td>(to, fl2, NYC)</td>
<td>-0.16</td>
</tr>
</tbody>
</table>

\textbf{USER:} i want to fly from new york to los angeles
Match

**User:** I want to fly from New York to Los Angeles

\[ \lambda x. \text{from}(x, NYC) \land \text{to}(x, LA) \]

Alignment: \( x \rightarrow fl \)

<table>
<thead>
<tr>
<th>Property</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (from, fl, NYC) )</td>
<td>-0.5</td>
</tr>
<tr>
<td>( (to, fl, LA) )</td>
<td>-0.83</td>
</tr>
<tr>
<td>( (from, fl2, LA) )</td>
<td>-0.16</td>
</tr>
<tr>
<td>( (to, fl2, NYC) )</td>
<td>-0.16</td>
</tr>
</tbody>
</table>
Score

**USER:**  \( i \text{ want to fly from new york to los angeles} \)

\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \]

Alignment:  \( x \rightarrow fl \)

**Loss**  \[ -0.5 - 0.83 = -1.33 \]

<table>
<thead>
<tr>
<th>Property</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>((\text{from, fl, NYC}))</td>
<td>-0.5</td>
</tr>
<tr>
<td>((\text{to, fl, LA}))</td>
<td>-0.83</td>
</tr>
<tr>
<td>((\text{from, fl2, LA}))</td>
<td>-0.16</td>
</tr>
<tr>
<td>((\text{to, fl2, NYC}))</td>
<td>-0.16</td>
</tr>
</tbody>
</table>
Align and Score

\[ \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \]
Alignment: \( x \to \text{fl} \)
\[ \text{Loss} = -0.5 - 0.83 = -1.33 \]

<table>
<thead>
<tr>
<th>Property</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>(from, fl, NYC)</td>
<td>-0.5</td>
</tr>
<tr>
<td>(to, fl, LA)</td>
<td>-0.83</td>
</tr>
<tr>
<td>(from, fl2, LA)</td>
<td>-0.16</td>
</tr>
<tr>
<td>(to, fl2, NYC)</td>
<td>-0.16</td>
</tr>
</tbody>
</table>

\[ \lambda x[]. \text{from}(x[0], \text{NYC}) \land \text{from}(x[1], \text{LA}) \]
Alignment: \( x[0] \to \text{fl}, x[1] \to \text{fl2} \)
\[ \text{Loss} = -0.5 - 0.16 = -0.66 \]
Comparing Two Candidates

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

\( \lambda x. \text{from}(x, \text{NYC}) \land \text{to}(x, \text{LA}) \) ✅

Alignment: \( x \rightarrow \text{fl} \)

\( \text{Loss} = -0.5 - 0.83 = -1.33 \)

\( \lambda x [\cdot]. \text{from}(x[0], \text{NYC}) \land \text{from}(x[1], \text{LA}) \) ✗

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

\( \text{Loss} = -0.5 - 0.16 = -0.66 \)
• So far:
  – Showed how to generate lexicon using the conversation
  – Introduced our sources of loss

• Next:
  – Combine everything into a Perceptron-like learning algorithm
Learning Algorithm

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

For T iterations, for each training sample:

• Step 1: Lexical generation
  – Generate a large lexicon from system utterances
  – Parse with the generated lexicon using the model
  – Get the minimal loss parses from the k-best parses
  – Add their lexical items to the lexicon

• Step 2: Update parameters
Learning Algorithm

For T iterations, for each training sample:

• Step 1: Lexical generation
• Step 2: Update parameters
  – Parse using the model
  – Split all parses into two sets using the loss function: optimal and non-optimal
  – Find model score violations between the sets
  – Do a perceptron update using these violations
Evaluation

- Dataset
- Experimental setup
- Evaluation metrics
- Systems and results
DARPA Communicator

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

<table>
<thead>
<tr>
<th></th>
<th>Lucent</th>
<th></th>
<th>BBN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>Conversations</td>
<td>144</td>
<td>70</td>
<td>71</td>
<td>91</td>
</tr>
<tr>
<td>Context Independent</td>
<td>208</td>
<td>96</td>
<td>67</td>
<td>67</td>
</tr>
</tbody>
</table>
Experimental Setup

• Features similar to Zettlemoyer and Collins 2007:
  – Indicators for lexical item use
  – Indicators for parsing operators use
  – Properties of the logical form

• Seed lexicon:
  – Domain independent
    • “and”, “the” ...
  – List of domain-specific noun phrases
    • “new york”, “jfk”, “delta” ...

• Learning parameters optimized on training data
Evaluation Metrics

• Performance against gold standard:
  – Exact match
  – Partial credit

Sentence: i want to return from seattle to boston
Label: $\lambda x. to(x, BOS) \land from(x, SEA) \land return(x)$
Guess: $\lambda x. to(x, BOS) \land from(x, SEA)$  ✗

<table>
<thead>
<tr>
<th>Partial Credit</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2/3</td>
<td>2/2</td>
</tr>
</tbody>
</table>
Results: Ablation Tests

Development Set Results
Exact Match, F1 Measure

- Without Conversational Loss
- Without Domain Loss
- Our Approach

![Bar chart comparing Lucent and BBN results for ablation tests.

Lucent
- Blue: Without Conversational Loss
- Red: Without Domain Loss
- Green: Our Approach

BBN
- Blue: Without Conversational Loss
- Red: Without Domain Loss
- Green: Our Approach]
Results: Evaluation Data

F1 Measure

- Lucent Exact Match
- Lucent Partial Credit
- BBN Exact Match
- BBN Partial Credit

Legend:
- Blue: No Conversation Baseline
- Green: Our Approach
- Red: Supervised
Future Work

• Reason about context-dependent meaning
• Model and recover speech acts
• Combine in a complete self-improving dialog system
Contributions

• Developed a loss-driven learning approach for semantic parsers
• Demonstrated learning of semantic parsers from conversational logs
• Described a learning approach that will allow dialog systems to learn from failures
Contributions

• Developed a loss-driven learning approach for semantic parsers
• Demonstrated learning of semantic parsers from conversational logs
• Described a learning approach that will allow dialog systems to learn from failures

Thanks for listening
Supplementary Slides
Semantic Parsing

Natural Language \( F \) Meaning Representation
## Context-Dependent vs. Context-Independent

| SYSTEM | hello, evaluation user. welcome to the bell labs travel reservation system. i have a tutorial explaining my new features. if you'd like to hear it, just ask me for the tutorial. 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 kansas 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 kansas 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]
Parsing with CCGs

i’d like to go from seattle to denver
Parsing with CCGs

\[
\begin{align*}
\text{i'd like to go} & \quad \frac{S/N}{\lambda f. f} \\
\text{from} & \quad \frac{(N \backslash N)/NP}{\lambda y. \lambda f. \lambda x. f(x) \land from(x, y)} \\
\text{seattle} & \quad \frac{NP}{SEA} \\
\text{to} & \quad \frac{(N \backslash N)/NP}{\lambda y. \lambda f. \lambda x. f(x) \land to(x, y)} \\
\text{denver} & \quad \frac{NP}{DEN}
\end{align*}
\]
Parsing with CCGs

i'd like to go
\[ S/N \]
\[ \lambda f.f \]

from
\[ (N\backslash N)/NP \]
\[ \lambda y.\lambda f.\lambda x.f(x) \land from(x, y) \]

seattle
\[ NP \]
\[ SEA \]

\[\lambda y.\lambda f.\lambda x.f(x) \land to(x, y)\]

denver
\[ NP \]
\[ DEN \]
# Parsing with CCGs

<table>
<thead>
<tr>
<th>i'd like to go</th>
<th>from</th>
<th>seattle</th>
<th>to</th>
<th>denver</th>
</tr>
</thead>
<tbody>
<tr>
<td>S/N (\lambda \ell. \ell)</td>
<td>(\frac{(N\setminus N)}{NP}) (\lambda y. \lambda f. \lambda x. \ell(x) \land from(x, y))</td>
<td>NP (\text{SEA})</td>
<td>(\frac{(N\setminus N)}{NP}) (\lambda y. \lambda f. \lambda x. \ell(x) \land to(x, y))</td>
<td>NP (\text{DEN})</td>
</tr>
</tbody>
</table>

## Lexicon

### Lexical Entry

- **Text**
## Parsing with CCGs

<table>
<thead>
<tr>
<th>i’d like to go</th>
<th>from</th>
<th>seattle</th>
<th>to</th>
<th>denver</th>
</tr>
</thead>
<tbody>
<tr>
<td>[S/N] [\lambda f. f]</td>
<td>((N \backslash N) / NP) [\lambda y. \lambda f. \lambda x. f(x) \land \text{from}(x, y)]</td>
<td>[NP] [SEA]</td>
<td>((N \backslash N) / NP) [\lambda y. \lambda f. \lambda x. f(x) \land \text{to}(x, y)]</td>
<td>[NP] [DEN]</td>
</tr>
</tbody>
</table>

### Lexicon

- **Lexical Entry**
  - **Text**
  - **Category**
Parsing with CCGs
Parsing with CCGs

\[
\begin{array}{cccc}
\text{i'd like to go} & \text{from} & \text{seattle} & \text{to} \\
S/N & (N \setminus N)/NP & NP & (N \setminus N)/NP \\
\lambda f. f & \lambda y. \lambda f. \lambda x. f(x) \land \text{from}(x, y) & \text{SEA} & \lambda y. \lambda f. \lambda x. f(x) \land \text{to}(x, y) \\
\end{array}
\]
# Parsing with CCGs

<table>
<thead>
<tr>
<th>i’d like to go</th>
<th>from</th>
<th>seattle</th>
<th>to</th>
<th>denver</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S/N )</td>
<td>( (N/N)/NP )</td>
<td>( NP )</td>
<td>( (N/N)/NP )</td>
<td>( NP )</td>
</tr>
<tr>
<td>( \lambda f . f )</td>
<td>( \lambda y . \lambda f . \lambda x . f(x) \land \text{from}(x, y) )</td>
<td>( \text{SEA} )</td>
<td>( \lambda y . \lambda f . \lambda x . f(x) \land \text{to}(x, y) )</td>
<td>( \text{DEN} )</td>
</tr>
</tbody>
</table>

**Lexicon**

**Lexical Entry**

- **Text**
- **Category**
- **Syntax**
- **λ**
Parsing with CCGs
Parsing with CCGs

Lexicon

Lexical Entry

Text

Category

Syntax

λ

Combinators

Application
Parsing with CCGs

\[
\frac{\text{i'd like to go}}{S/N \lambda f.f} \quad \frac{\text{from}}{(N\backslash N)/NP \lambda y.\lambda f.\lambda x.f(x) \land \text{from}(x, y)} \quad \frac{\text{seattle}}{NP \text{SEA}} \quad \frac{\text{to}}{(N\backslash N)/NP \lambda y.\lambda f.\lambda x.f(x) \land \text{to}(x, y)} \quad \frac{\text{denver}}{NP \text{DEN}}
\]

\[
\frac{(N\backslash N)}{\lambda f.\lambda x.f(x) \land \text{from}(x, \text{SEA})} \quad \frac{(N\backslash N)}{\lambda f.\lambda x.f(x) \land \text{to}(x, \text{DEN})}
\]

\[
\frac{N\backslash N}{\lambda f.\lambda x.f(x) \land \text{from}(x, \text{SEA}) \land \text{to}(x, \text{DEN})}
\]

Lexicon
- **Lexical Entry**
  - **Text**
  - **Category**
    - **Syntax**
    - \(\lambda\)

Combinators
- **Application Composition**
Parsing with CCGs

\[
\begin{align*}
S/N & \quad \lambda f.f \\
\text{i'd like to go} & \quad \text{from} & \quad \text{seattle} & \quad \text{to} & \quad \text{denver} \\
\frac{(N\backslash N)/NP}{\lambda y.\lambda f.\lambda x.f(x) \land from(x, y)} & \quad \frac{NP}{\lambda y.\lambda f.\lambda x.f(x) \land to(x, y)} & \quad \frac{(N\backslash N)/NP}{\lambda y.\lambda f.\lambda x.f(x) \land to(x, y)} & \quad \frac{NP}{\lambda y.\lambda f.\lambda x.f(x) \land to(x, y)} \\
\lambda f.\lambda x.f(x) \land from(x, \text{SEA}) & \quad \frac{(N\backslash N)}{\lambda f.\lambda x.f(x) \land to(x, \text{DEN})} & \quad \frac{(N\backslash N)}{\lambda f.\lambda x.f(x) \land to(x, \text{DEN})} & \quad \frac{(N\backslash N)}{\lambda f.\lambda x.f(x) \land to(x, \text{DEN})} \\
\frac{N\backslash N}{\lambda f.\lambda x.f(x) \land from(x, \text{SEA}) \land to(x, \text{DEN})} & \quad \frac{N}{\lambda x.\text{from}(x, \text{SEA}) \land to(x, \text{DEN})} \\
\lambda f.\lambda x.f(x) \land from(x, \text{SEA}) \land to(x, \text{DEN}) & \quad \lambda x.\text{from}(x, \text{SEA}) \land to(x, \text{DEN})
\end{align*}
\]
Parsing with CCGs

\[
\begin{align*}
\text{Lexicon} & \quad \text{Combinators} \\
\text{Lexical Entry} & \quad \text{Application} \\
\text{Text} & \quad \text{Composition} \\
\text{Syntax} & \quad \text{Type-shifting} \\
\lambda & \quad \lambda
\end{align*}
\]
Parsing with CCGs

\[
\begin{align*}
S/N & \lambda f.f \\
\text{i'd like to go} & \frac{(N \setminus N) / NP}{\lambda y. \lambda f. \lambda x. f(x) \land from(x, y)} & \text{seattle} & \frac{(N \setminus N) / NP}{\lambda y. \lambda f. \lambda x. f(x) \land to(x, y)} & \text{denver} & \frac{NP}{NP} \\
\text{NP} & \text{SEA} & \text{DEN} & \text{DEN} & <B > \\
\frac{(N \setminus N)}{\lambda f. \lambda x. f(x) \land from(x, SEA)} & \frac{N \setminus N}{\lambda f. \lambda x. f(x) \land to(x, DEN)} & \frac{N}{\lambda x. \text{from}(x, \text{SEA}) \land to(x, \text{DEN})} & \frac{S}{\lambda x. \text{from}(x, \text{SEA}) \land to(x, \text{DEN})}
\end{align*}
\]
Supervised Lexical Learning with GENLEX

| Input Training Example | Sentence: i’d like to go to los angeles | Logical form: $\lambda x. to(x, LA)$ |

[Zettlemoyer & Collins 2005]
Supervised Lexical Learning with GENLEX

Sentence:  i’d like to go to los angeles
Logical form: \( \lambda x. to(x, LA) \)
Supervised Lexical Learning with GENLEX

Input Training Example

Output Lexicon

Sentence: i’d like to go to los angeles
Logical form: \( \lambda x.\text{to}(x,\text{LA}) \)

All possible sub-strings:
i’d
like
i’d like
to go to
to los
...

Categories created by rules triggered from the logical form:

S/N : \( \lambda x.x \)
N/N/NP : \( \lambda x.\lambda y.\lambda z.y(z)\Lambda to(z,x) \)
...

[Zettlemoyer & Collins 2005]
Supervised Lexical Learning with GENLEX

Input
Training
Example

Sentence:  i’d like to go to los angeles
Logical form: $\lambda x. to(x, LA)$

Output
Lexicon

All possible sub-strings:

i’d
like
i’d like
to go to
to los
...

Categories created by rules triggered from the logical form:

S/N : $\lambda x.x$

N\N/NP : $\lambda x. \lambda y. \lambda z.y(z)\land to(z,x)$
...

[Zettlemoyer & Collins 2005]
Indirect supervision Lexical Learning with GENLEX

Input Training Example

Sentence: i’d like to go to los angeles

Conversation:

SYSTEM: how can i help you? (OPEN_TASK)

USER: i’d like to go to los angeles

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

USER: new york

SYSTEM: leaving new york. (CONFIRM: from(fl, NYC)) going to which city? (ASK: \(\lambda x. to(fl, x)\))

USER: los angeles

SYSTEM: flying to los angeles. (CONFIRM: to(fl, LA))

[conversation continues]
Indirectly-supervised Lexical Learning with GENLEX

Input Training Example
Sentence:  
User: i’d like to go to los angeles
Conversation:  
System: how can i help you? (OPEN_TASK)
User: i’d like to go to los angeles
System: leaving what city? (ASK: \lambda x . from (fl, x))
User: new york
System: leaving new york. (CONFIRM: from(fl, NYC)) going to which city?
(ASK: \lambda x . to(fl, x))
User: los angeles
System: flying to los angeles. (CONFIRM: to(fl, LA))
[conversation continues]

Use logical constants from system utterances as triggers
Indirectely-supervised Lexical Learning with GENLEX

Input Training Example

Sentence: i’d like to go to los angeles

Conversation:

SYSTEM: how can i help you? (OPEN_TASK)

USER: i’d like to go to los angeles

SYSTEM: leaving what city? (ASK: λx from (fl, x))

[conversation continues]

Output Lexicon

All possible sub-strings:

i’d
like
i’d like
to go to
to los
...

Categories created by rules triggered from the logical form:

S/N : λx.x
N\N/NP : λx. λy. λz.y(z)∧to(z,x)
N\N/NP : λx. λy. λz.y(z)∧from(z,x)
...

X
Indirectly-supervised Lexical Learning with GENLEX

Generated Lexicon is:
• Overgeneralized
• Incomplete

<table>
<thead>
<tr>
<th>Output Lexicon</th>
<th>All possible sub-strings:</th>
</tr>
</thead>
<tbody>
<tr>
<td>i’d</td>
<td></td>
</tr>
<tr>
<td>like</td>
<td></td>
</tr>
<tr>
<td>i’d like</td>
<td></td>
</tr>
<tr>
<td>to go to</td>
<td></td>
</tr>
<tr>
<td>to los</td>
<td></td>
</tr>
</tbody>
</table>

Categories created by rules triggered from the logical form:

<table>
<thead>
<tr>
<th>S/N : ( \lambda x.x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/N/NP : ( \lambda x. \lambda y. \lambda z.y(z) \land to(z,x) )</td>
</tr>
<tr>
<td>N/N/NP : ( \lambda x. \lambda y. \lambda z.y(z) \land from(z,x) )</td>
</tr>
</tbody>
</table>

...
# Results: Ablation Tests

## Development Set Results

<table>
<thead>
<tr>
<th>Exact Match Metric</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without conversational loss</td>
<td>0.35</td>
<td>0.34</td>
<td>0.35</td>
<td>0.66</td>
<td>0.54</td>
<td>0.59</td>
</tr>
<tr>
<td>Without domain loss</td>
<td>0.42</td>
<td>0.42</td>
<td>0.42</td>
<td>0.69</td>
<td>0.56</td>
<td>0.61</td>
</tr>
<tr>
<td>Our approach</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
<td>0.77</td>
<td>0.64</td>
<td>0.69</td>
</tr>
</tbody>
</table>
## Results: Evaluation Data

<table>
<thead>
<tr>
<th>Exact Match Metric</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No conversations baseline</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.16</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Our approach</td>
<td>0.58</td>
<td>0.55</td>
<td>0.56</td>
<td>0.85</td>
<td>0.75</td>
<td>0.79</td>
</tr>
<tr>
<td>Supervised method</td>
<td>0.7</td>
<td>0.68</td>
<td>0.69</td>
<td>0.87</td>
<td>0.78</td>
<td>0.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partial Credit Metric</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No conversations baseline</td>
<td>0.26</td>
<td>0.35</td>
<td>0.29</td>
<td>0.26</td>
<td>0.33</td>
<td>0.29</td>
</tr>
<tr>
<td>Our approach</td>
<td>0.68</td>
<td>0.63</td>
<td>0.65</td>
<td>0.97</td>
<td>0.57</td>
<td>0.72</td>
</tr>
<tr>
<td>Supervised method</td>
<td>0.75</td>
<td>0.68</td>
<td>0.72</td>
<td>0.96</td>
<td>0.68</td>
<td>0.79</td>
</tr>
</tbody>
</table>