Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions

Yoav Artzi and Luke Zettlemoyer
University of Washington

Semantic Parsing

Show me all papers about semantic parsing

$$\lambda x. \text{paper}(x) \land \text{topic}(x, \text{SEMPAR})$$

Less Supervision
- Answers
- Demonstrations
- Situated examples

More Domains
- Databases
- Large knowledge-bases
- Instructions
- Referring expressions
- Regular expressions

Later this session
Semantic Parsing

Show me all papers about semantic parsing

\[ \lambda x. \text{paper}(x) \land \text{topic}(x, \text{SEMPAR}) \]
Executing Navigation Instructions

place your back against the wall of the t intersection

turn left

go forward along the pink flowered carpet hall two segments to the intersection with the brick hall

•
•
•
Resolve Referents

go to the next sofa
Resolve Referents

go to the next sofa
Disambiguate Word Sense

go to the chair
Disambiguate Word Sense

go to the chair
Identify Executable Actions

walk forward twice
Identify Executable Actions

move, move? I can do that!

walk forward twice
Understand Implicit Requests

at the chair, turn right
Understand Implicit Requests

need to move first

at the chair, turn right
Grounded Learning

move forward? turn?

turn to face the chair
Grounded Learning

move forward?  ❌

turn?

turn to face the chair
Learning Signal

Instruction:
at the chair, move forward three steps past the sofa

Demonstration:
Learning Signal

Instruction:

at the chair, move forward three steps past the sofa

Demonstration:

During learning: validate executions of different interpretations against demonstrations
Learning

Demonstrations

Validation

Execution

Joint Inference

Situated Signals

Logical Form

Parsing

Weighted CCG
Learning

Demonstrations

Validation

Lexical Generation

1 Parsing

2 Logical Form

3 Execution

Situated Signals

Joint Inference

Weighted CCG
Combinatory Categorial Grammars

\[
\begin{align*}
\text{move} & \quad \text{to} & \quad \text{the} & \quad \text{chair} \\
S & \quad \text{AP/NP} & \quad \text{NP/N} & \quad N \\
\lambda a.\text{move}(a) & \quad \lambda x.\lambda a.\text{to}(a, x) & \quad \lambda f.\lambda x.f(x) & \quad \lambda x.\text{chair}(x) \\
\lambda x.\lambda a.\text{to}(a, x) & \quad \text{NP} & \quad \text{NP} & \quad \text{NP} \\
\lambda f.\lambda x.f(x) & \quad \lambda x.\text{chair}(x) & \quad \lambda x.\text{chair}(x) & \quad \lambda x.\text{chair}(x) \\
\lambda f.\lambda x.f(x) & \quad \lambda x.\text{chair}(x) & \quad \lambda x.\text{chair}(x) & \quad \lambda x.\text{chair}(x) \\
\lambda f.\lambda a.f(a) \land \text{to}(a, \lambda x.\text{chair}(x)) & \quad \lambda f.\lambda a.f(a) \land \text{to}(a, \lambda x.\text{chair}(x)) & \quad \lambda f.\lambda a.f(a) \land \text{to}(a, \lambda x.\text{chair}(x)) & \quad \lambda f.\lambda a.f(a) \land \text{to}(a, \lambda x.\text{chair}(x)) \\
\lambda a.\text{move}(a) \land \text{to}(a, \lambda x.\text{chair}(x)) & \quad \lambda a.\text{move}(a) \land \text{to}(a, \lambda x.\text{chair}(x)) & \quad \lambda a.\text{move}(a) \land \text{to}(a, \lambda x.\text{chair}(x)) & \quad \lambda a.\text{move}(a) \land \text{to}(a, \lambda x.\text{chair}(x))
\end{align*}
\]
Combinatory Categorial Grammars

```
move   to  the  chair
S      AP/NP NP/N  N
\la.\text{move}(a)  \la x.\la x.\text{to}(a, x)  \lambda f.\lambda x.\text{f}(x)  \lambda x.\text{chair}(x)
```

```
\lambda f.\lambda a.\text{f}(a) \land \text{to}(a, \lambda x.\text{chair}(x))
```

Lexicon

Combinators
Combinatory Categorial Grammars

\[
\begin{array}{cccc}
S & \lambda a. move(a) & \lambda x. \lambda a. to(a, x) & \lambda f. \lambda x. f(x) \\
AP/NP & \lambda x. NP/N & NP/N & \lambda x. cl \\
ix.chair(x) & \lambda a. to(a, ix.chair(x)) & AP & \lambda a. move(a) \wedge to(a, ix.chair(x)) \\
S \backslash S & \lambda f. \lambda a. f(a) \wedge to(a, ix.chair(x)) & S & \lambda a. move(a) \wedge to(a, ix.chair(x)) \\
\end{array}
\]
Weighted Linear CCGs

• Given a weighted linear model:
  - CCG lexicon $\Lambda$
  - Feature function $f : X \times Y \rightarrow \mathbb{R}^m$
  - Weights $w \in \mathbb{R}^m$

• The best parse is:
  $$y^* = \arg \max_y w \cdot f(x, y)$$

• We consider all possible parses $y$ for sentence $x$ given the lexicon $\Lambda$
Learning

Demonstrations

Validation

3 Execution

Situated Signals

2 Logical Form

✓ Parsing

4 Lexical Generation

Joint Inference

Weighted CCG
Modeling for Semantic Parsing

- **Nouns**: Sets of entities
- **PPs and adjectives**: Constrain sets
- **Noun phrases**: Specific entities
- **Verbs**: Relations between entities
Modeling for Semantic Parsing

<table>
<thead>
<tr>
<th>Nouns</th>
<th>Sets of entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPs and adjectives</td>
<td>Constrain sets</td>
</tr>
<tr>
<td>Noun phrases</td>
<td>Specific entities</td>
</tr>
<tr>
<td>Verbs</td>
<td>Relations between entities</td>
</tr>
</tbody>
</table>

Works well for natural language interfaces for DBs
Modeling for Semantic Parsing

- **Nouns**: Sets of entities
- **PPs and adjectives**: Constrain sets
- **Noun phrases**: Specific entities
- **Verbs**: Relations between entities

Previous work on instructional language adopted procedural representation

[Matuszek et al. 2010; 2012; Chen, Mooney 2011; Chen 2012; Kim, Mooney 2012]
Modeling for Semantic Parsing

Previous work on instructional language adopted procedural representation

How can we use this approach for instructions?
## Modeling for Semantic Parsing

### Name objects

<table>
<thead>
<tr>
<th>Nouns</th>
<th>Sets of entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPs and adjectives</td>
<td>Constrain sets</td>
</tr>
<tr>
<td>Noun phrases</td>
<td>Specific entities</td>
</tr>
</tbody>
</table>

### Instructions to execute

<table>
<thead>
<tr>
<th>Verbs</th>
<th>Davidsonian Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imperatives</td>
<td>Sets of events</td>
</tr>
</tbody>
</table>
Spatial Environment Modeling

- Maps are graphs of connected positions
- Agent can move forward, turn right and turn left
- Agent perceives clusters of positions
- Clusters capture objects
**Spatial Environment Modeling**

- Maps are graphs of connected positions
- Agent can move forward, turn right and turn left
- Agent perceives clusters of positions
- Clusters capture objects
Spatial Environment Modeling

- Maps are graphs of connected positions
- Agent can move forward, turn right and turn left
- Agent perceives clusters of positions
- Clusters capture objects
Spatial Language

- Nouns
- Noun phrases
- Adjectives
- Prepositional phrases
- Spatial relations
Spatial Language

Nouns denote sets of objects

chair

\(\lambda x.\text{chair}(x)\)

\{ , \}
Spatial Language

Noun phrases name specific entities
Spatial Language

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Noun phrases name specific entities

the chair

\( \forall x. \text{chair}(x) \)
Spatial Language

Noun phrases name specific entities

the chair

$\forall x.\text{chair}(x)$

Definite determiner depends on agent state
Spatial Language

Noun phrases name specific entities

\( \text{definite determiner} \)

\( \forall x. \text{chair}(x) \)

Definite determiner depends on agent state
Spatial Language

- Nouns
- Noun phrases
- Adjectives
- Prepositional phrases
- Spatial relations
Modeling Instructions

• Sequences of identical actions are events
• Use Neo-Davidsonian event semantics
• Represent imperatives as sets of events
Modeling Instructions

Imperatives are sets of events

\[ \text{move} \lambda a.\text{move}(a) \]

\[ \{ \rightarrow, \rightarrow\rightarrow, \rightarrow\rightarrow\rightarrow \} \]
Modeling Instructions

Imperatives are sets of events

\[\lambda a. \text{move}(a)\]

Disambiguate by preferring shorter sequences
Events can be modified by adverbials

\( \lambda a. \text{move}(a) \wedge \text{to}(a, \_x. \text{chair}(x)) \)
Executing Logical Forms

\[ \lambda a. move(a) \land to(a, \forall x. chair(x)) \]

Consider all variable assignments to find satisfying ones
Executing Logical Forms

\[ \lambda a. \text{move}(a) \land \text{to}(a, \text{chair}(x)) \]

1. go to the chair
Executing Logical Forms

\[ \lambda a. \text{move}(a) \land \text{to}(a, \forall x. \text{chair}(x)) \]

go to the chair

1  2  3  4  5

1  2  3  4  5

\[ \text{go to the chair} \]

\[ \lambda a. \text{move}(a) \land \text{to}(a, \forall x. \text{chair}(x)) \]
move until you reach the chair

\[ \text{move} (a)^{\text{post} (a, \text{intersect}(\text{ix.chair}(x), \text{you}))} \]

World models change during execution
Dynamic Models

World models change during execution

move until you reach the chair

\( \lambda a. move(a) \land \)

\( post(a, \text{intersect}(ix.chair(x), you)) \)
Dynamic Models

World models change during execution

move until you reach the chair

\[ \lambda a. \text{move}(a) \land \text{post}(a, \text{intersect}(\text{\textit{x:chair}(x, you)))) \]

Never intersects
Dynamic Models

World models change during execution

move until you reach the chair

\[
\lambda a.\text{move}(a) \land \\
\text{post}(a, \text{intersect}(\text{ix.chair}(x), \text{you}))
\]

Update model to reflect state change
Dynamic Models

World models change during execution

move until you reach the chair

\[ \lambda a. \text{move}(a) \land \text{post}(a, \text{intersect}(\text{ix.chair}(x), \text{you})) \]

Update model to reflect state change
Implicit Actions

Consider actions with prefixed implicit actions at the chair, turn left

$$\lambda a. \text{turn}(a) \land \text{dir}(a, \text{left}) \land \text{pre}(a, \text{intersect}(\text{ux.chair}(x), \text{you}))$$
Consider actions with prefixed implicit actions at the chair, turn left

\[ \lambda a. \text{turn}(a) \land \text{dir}(a, \text{left}) \land \text{pre}(a, \text{intersect}(ix.\text{chair}(x), you)) \]
Consider actions with prefixed implicit actions at the chair, turn left

$$\lambda a. \text{turn}(a) \land \text{dir}(a, \text{left}) \land \text{pre}(a, \text{intersect}(\text{ux.chair}(x), \text{you}))$$

Implicit actions
Joint Inference

World State

Language Understanding

Actions
• Execution is integrated into parsing
• During parsing, execute logical forms and observe the result
Learning

Demonstrations

Validation

Execution

Logical Form

Situated Signals

Joint Inference

Weighted CCG

 Parsing

Lexical Generation

4
Learning

- Based on a small set of seed templates
- New coarse-to-fine parsing algorithm to gradually prune the potential lexical entries
- Conservative approach to introduce new entries to the model
Learning

Demonstrations

Validation

Lexical Generation

Execution ✓
Logical Form ✓
Parsing ✓

Situated Signals

Joint Inference

Weighted CCG
Validation-Driven Learning

- Online
- 2 steps:
  - Lexical generation
  - Parameter update
- Driven by a weak validation signal
Validation-Driven Learning

For $T$ iterations, for each training sample:

- **Step 1: Lexical generation**
  - Generate a large number of potential lexical entries
  - Parse with the generated lexicon using the model
  - Select the best valid parses from the k-best parses
  - Add their lexical items to the lexicon

- **Step 2: Update parameters**
Validation-Driven Learning

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: max scoring valid and invalid
  - Find margin violating pairs between the sets
  - Do a perceptron-style update using these violations
## Related Work

<table>
<thead>
<tr>
<th>Category</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised semantic parsing</td>
<td>[Kate, Mooney 2006; Wong, Mooney 2007; Muresan 2011]</td>
</tr>
<tr>
<td>Supervised semantic parsing with CCGs</td>
<td>[Zettlemoyer, Collins 2005; 2007; Kwiatkowski et al. 2010; 2011]</td>
</tr>
<tr>
<td>Weakly supervised semantic parsing</td>
<td>[Clarke et al. 2010; Goldwasser, Roth 2011; Liang et al. 2011; Kirshnamurthy, Mitchell 2012; Goldwasser et al. 2011]</td>
</tr>
<tr>
<td>Grounded Semantic Analysis</td>
<td>[Liang et al. 2009; Chen et al. 2010; Matuszek et al. 2012]</td>
</tr>
<tr>
<td>Non-joint Execution of Instructions</td>
<td>[Matuszek et al. 2010; 2012; Chen, Mooney 2011; Chen 2012; Kim, Mooney 2012]</td>
</tr>
</tbody>
</table>
Experimental Setup

• Seed lexicon from an annotated randomly selected 12 instruction sequences

• Features: lexical, type-raising usage and repetitions in logical coordinations

• Consider only executable parses as complete
Results

SAIL Corpus - Cross Validation

Chen and Mooney 2011
Chen 2012
Kim and Mooney 2012
Results
SAIL Corpus - Cross Validation

Chen and Mooney 2011
Chen 2012
Kim and Mooney 2012
Final State Validation
Trace Validation
Results
SAIL Corpus - Cross Validation

- Chen and Mooney 2011
- Chen 2012
- Kim and Mooney 2012
- Final State Validation
- Trace Validation
- Kim and Mooney 2013

<table>
<thead>
<tr>
<th></th>
<th>Single Sentence</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen and Mooney 2011</td>
<td>64.25</td>
<td>30.9</td>
</tr>
<tr>
<td>Chen 2012</td>
<td>65.28</td>
<td>31.93</td>
</tr>
<tr>
<td>Kim and Mooney 2012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final State Validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace Validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kim and Mooney 2013</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results
Oracle Corpus - Held-out Set

![Bar chart showing results for Single Sentence and Sequence with categories: Final State Validation and Trace Validation.](image)
Contributions

• Linguistically-driven modeling of instructional language

• Joint inference for interpretation and execution of grounded language

• General weakly-supervised learning approach for semantic parsers
UW SPF

Open source semantic parsing framework

http://yoavartzi.com/spf

Semantic Parser

Flexible High-Order Logic Representation

Learning Algorithms
UW SPF

Open source semantic parsing framework

http://yoavartzi.com/spf

Semantic Parser
Flexible High-Order Logic Representation
Learning Algorithms

Navigation code and data available online

http://yoavartzi.com/navi

Coming up: ACL tutorial
[fin]