Learning Dependency-Based Compositional Semantics


Percy Liang

Google/Stanford

joint work with Michael Jordan and Dan Klein
Motivating Problem: Question Answering
Motivating Problem: Question Answering

*What is the largest city in California?*
Motivating Problem: Question Answering

What is the largest city in California?

What is the largest city in a state bordering California?
Semantic Interpretation
Semantic Interpretation

What is the largest city in a state bordering California?

Phoenix
Semantic Interpretation

What is the largest city in a state bordering California?

Phoenix
What is the largest city in a state bordering California?

Phoenix
Semantic Interpretation

What is the largest city in a state bordering California?

\[
\text{city}(c) \land \exists s. \text{state}(s) \land \text{loc}(c, s)
\]

Phoenix
Semantic Interpretation

What is the largest city in a state bordering California?

city(c) \land \exists s. \text{state}(s) \land \text{loc}(c, s) \land \text{border}(s, \text{CA})

Phoenix
Semantic Interpretation

What is the largest city in a state bordering California?

\[
\text{argmax}\{c : \text{city}(c) \land \exists s. \text{state}(s) \land \text{loc}(c, s) \land \text{border}(s, \text{CA})\}, \text{population}
\]

Phoenix
What is the largest city in a state bordering California?

\[
\text{argmax}\{c : \text{city}(c) \land \exists s. \text{state}(s) \land \text{loc}(c, s) \land \text{border}(s, \text{CA})\}, \text{population}
\]

\[
\text{computation}
\]

\[
\text{Phoenix}
\]
Semantic Interpretation

What is the largest city in a state bordering California?

?.

computation

Phoenix
Supervision for Semantic Interpretation
Detailed Supervision (current)  

What is the largest city in California?

\[
\text{argmax}\{c : \text{city}(c) \land \text{loc}(c, \text{CA})\}, \text{population}
\]
Detailed Supervision (current)  What is the largest city in California?

\[
\text{expert} \quad \argmax\{c : \text{city}(c) \land \text{loc}(c, \text{CA})\}, \text{population}
\]
Detailed Supervision (current) - doesn’t scale up

What is the largest city in California?

expert

argmax(\{c : city(c) \land loc(c, CA)\}, population)
Supervision for Semantic Interpretation

**Detailed Supervision (current)**
- doesn’t scale up

**What is the largest city in California?**

\[
\text{expert} \quad \text{argmax}\left(\{c : \text{city}(c) \land \text{loc}(c, \text{CA})\}, \text{population}\right)
\]

**Natural Supervision (new)**

**What is the largest city in California?**

\[
\text{Los Angeles}
\]
Supervision for Semantic Interpretation

**Detailed Supervision (current)**  
- doesn’t scale up

\[ \text{argmax}\{c : \text{city}(c) \land \text{loc}(c, \text{CA})\}, \text{population} \]

**Natural Supervision (new)**

\[ \text{What is the largest city in California?} \]

\[ \text{What is the largest city in California?} \]

\[ \text{Los Angeles} \]
Supervision for Semantic Interpretation

**Detailed Supervision (current)**
- doesn’t scale up

**Natural Supervision (new)**
- scales up

---

**What is the largest city in California?**

**expert**

\[ \text{argmax}\left\{ c: \text{city}(c) \land \text{loc}(c, \text{CA}) \right\}, \text{population} \]

---

**What is the largest city in California?**

**non-expert**

Los Angeles
Supervision for Semantic Interpretation

**Detailed Supervision (current)**
- doesn’t scale up
- representation-dependent

What is the largest city in California?

expert

\[ \text{argmax}\{c : \text{city}(c) \land \text{loc}(c, \text{CA})\}, \text{population} \]

**Natural Supervision (new)**
- scales up

What is the largest city in California?

non-expert

Los Angeles
Supervision for Semantic Interpretation

**Detailed Supervision (current)**
- doesn’t scale up
- representation-dependent

**Natural Supervision (new)**
- scales up
- representation-independent

*What is the largest city in California?*

**expert**
\[
\text{argmax}\{c : \text{city}(c) \land \text{loc}(c, \text{CA})\}, \text{population}\]

**non-expert**
*Los Angeles*
Outline

Representation

Learning

Experiments
Considerations

**Computational:** how to efficiently search exponential space?
Considerations

Computational: how to efficiently search exponential space?

What is the most populous city in California?

Los Angeles
Considerations

**Computational:** how to efficiently search exponential space?

What is the most populous city in California?

\[ \lambda x. \text{state}(x) \]

Los Angeles
Considerations

**Computational**: how to efficiently search exponential space?

*What is the most populous city in California?*

\[ \lambda x. \text{city}(x) \]

*Los Angeles*
Considerations

Computational: how to efficiently search exponential space?

What is the most populous city in California?

\[ \lambda x.\text{city}(x) \land \text{loc}(x, \text{CA}) \]

Los Angeles
Considerations

**Computational**: how to efficiently search exponential space?

*What is the most populous city in California?*

\[ \lambda x. \text{state}(x) \land \text{border}(x, \text{CA}) \]

*Los Angeles*
Considerations

Computational: how to efficiently search exponential space?

What is the most populous city in California?

population(CA)

Los Angeles
Considerations

Computational: how to efficiently search exponential space?

What is the most populous city in California?

\[ \text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{population}(x)) \]

Los Angeles
Considerations

**Computational**: how to efficiently search exponential space?

*What is the most populous city in California?*

Los Angeles
Considerations

**Computational**: How to efficiently search exponential space?

What is the most populous city in California?

Los Angeles

**Statistical**: How to parametrize mapping from sentence to logical form?

\[
\text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{population}(x))
\]
Dependency-Based Compositional Semantics (DCS)

*What is the most populous city in California?*
What is the most populous city in California?

Los Angeles
Dependency-Based Compositional Semantics (DCS)

What is the most populous city in California?

![Dependency Tree]

Los Angeles

Advantages of DCS: nice computational, statistical, linguistic properties
Where do the answers come from?

What is the most populous city in California?

Los Angeles
Where do the answers come from?

*What is the most populous city in California?*

```
Database  ➔  Los Angeles
```

```
database

city

1 1

population  loc

1 1

c

argmax

C

loc

2 1

CA
```
# Database

<table>
<thead>
<tr>
<th>city</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Francisco</td>
<td>Alabama</td>
</tr>
<tr>
<td>Chicago</td>
<td>Alaska</td>
</tr>
<tr>
<td>Boston</td>
<td>Arizona</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>loc</th>
<th>border</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount Shasta</td>
<td>Washington</td>
</tr>
<tr>
<td>San Francisco</td>
<td>Oregon</td>
</tr>
<tr>
<td>Boston</td>
<td>Idaho</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>California</td>
<td>Washington</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Massachusetts</td>
<td>Oregon</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>California</td>
<td>Washington</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Basic DCS Trees

DCS tree

- city
  - 1
  - loc
    - 2
    - CA
  - 1
- 1
- 1

Database
A DCS tree encodes a \textbf{constraint satisfaction problem} (CSP)
A DCS tree encodes a **constraint satisfaction problem (CSP)**
A DCS tree encodes a **constraint satisfaction problem (CSP)**
Basic DCS Trees

A DCS tree encodes a constraint satisfaction problem (CSP)
A DCS tree encodes a constraint satisfaction problem (CSP)
Basic DCS Trees

A DCS tree encodes a constraint satisfaction problem (CSP)
A DCS tree encodes a constraint satisfaction problem (CSP)
A DCS tree encodes a **constraint satisfaction problem** (CSP)
A DCS tree encodes a constraint satisfaction problem (CSP)
A DCS tree encodes a **constraint satisfaction problem** (CSP)

**Computation:** dynamic programming $\Rightarrow$ time $= O(\# \text{ nodes})$
Properties of DCS Trees

```
city
  \-- loc (1)
  \   |  \___ traverses (2)
  \   |       \___ state (1)
  \   |           |  \___ border (2)
  \   |           |      \___ CA (1)
  \   \           \___ traverse (1)
  \___ river (1)
      |  \___ major (1)
      |      \___ traverse (2)
      |          \___ AZ (1)
```
Properties of DCS Trees

Trees
Properties of DCS Trees

Linguistics

syntactic locality

Trees
Properties of DCS Trees

Linguistics

syntactic locality

Trees

Computation
efficient interpretation
Divergence between Syntactic and Semantic Scope

most populous city in California
Divergence between Syntactic and Semantic Scope

most populous city in California

Syntax

```
    city
     \---
      \--
       \- populous
         \- in
           \- most
             \- California
```
Divergence between Syntactic and Semantic Scope

*most populous city in California*

**Syntax**

```
most populous in city
```

**Semantics**

```
argmax(\(\lambda x.\text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x.\text{population}(x)\))
```
Divergence between Syntactic and Semantic Scope

\[ \text{most populous city in California} \]

**Syntax**

```
most populous in city
```

**Semantics**

\[ \text{argmax} (\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{population}(x)) \]
Divergence between Syntactic and Semantic Scope

*most populous city in California*

**Syntax**

```
(populous)
    
(city)
    
    (in)
    
    (most)
```

**Semantics**

```
\text{argmax}(\lambda x.\text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x.\text{population}(x))
```

**Problem:** syntactic scope is lower than semantic scope
Divergence between Syntactic and Semantic Scope

*most populous city in California*

**Syntax**

```
  city
    populous
      most
    in
      California
```

**Semantics**

```
argmax(\lambda x. city(x) \land loc(x, CA), \lambda x. population(x))
```

**Problem:** syntactic scope is lower than semantic scope

If DCS trees look like syntax, how do we get correct semantics?
Solution: Mark-Execute

*most populous city in California*

Superlatives

![Diagram showing the logical structure for finding the most populous city in California.](image-url)
Solution: Mark-Execute

**most populous city in California**

Mark at syntactic scope

![Diagram showing the argument max with city and population connected to the main node, and loc with CA connected to the main node.]
Solution: Mark-Execute

*most populous city in California*

**Execute** at semantic scope

**Mark** at syntactic scope
Solution: Mark-Execute

Alaska borders no states.

Execute at semantic scope

Mark at syntactic scope
Solution: Mark-Execute

*Some river traverses every city.*

**Execute** at semantic scope

**Mark** at syntactic scope
Some river traverses every city.

**Execute** at semantic scope

**Mark** at syntactic scope
Solution: Mark-Execute

Some river traverses every city.

Quantification (wide)

Execute at semantic scope

Mark at syntactic scope

Analogy: Montague’s quantifying in, Carpenter’s scoping constructor
Outline

Representation

Learning

Experiments
Graphical Model

database

$w$
Graphical Model

database

\[ z \]

\[ \text{capital} \]

\[ \text{CA} \]

\[ y \quad \text{Sacramento} \]

\[ w \]
Graphical Model

Interpretation: $p(y \mid z, w)$
(deterministic)
**Graphical Model**

![Graphical Model Diagram]

- **x**: capital of California?
- **z**: capital
  - Value: CA
- **w**: database
- **y**: Sacramento

**Interpretation:** $p(y | z, w)$
(deterministic)
Graphical Model

\[ p(y \mid z, w) \]
(deterministic)

Interpretation: \( p(y \mid z, w) \)
Graphical Model

Semantic Parsing: $p(z \mid x, \theta)$
(probabilistic)

Interpretation: $p(y \mid z, w)$
(deterministic)
Plan

- What’s possible? \( z \in \mathcal{Z}(x) \)
- What’s probable? \( p(z | x, \theta) \)
- Learning \( \theta \) from \((x, y)\) data
Words to Predicates (Lexical Semantics)

What is the most populous city in CA?
Words to Predicates (Lexical Semantics)

What is the most populous city in CA?

Lexical Triggers:
1. String match $CA \Rightarrow CA$
Words to Predicates (Lexical Semantics)

\[
\text{argmax} \quad \text{What is the most populous city in CA?}
\]

Lexical Triggers:

1. String match \( CA \Rightarrow CA \)
2. Function words (20 words) \( most \Rightarrow \text{argmax} \)
Words to Predicates (Lexical Semantics)

What is the most populous city in CA?

Lexical Triggers:

1. String match
   
   CA \Rightarrow CA

2. Function words (20 words)
   
   most \Rightarrow argmax

3. Nouns/adjectives
   
   city \Rightarrow city state river population
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]
Predicates to DCS Trees (Compositional Semantics)

$C_{i,j} = \text{set of DCS trees for span } [i, j]$
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]

\[ \arg \max \text{ population} \]

most populous

\[ C_{i,k} \]

\[ C_{k,j} \]

city in California

The image includes a diagram with nodes labeled as follows:

- Population
- City
- Location
- CA

The diagram illustrates the concepts of predicting the most populous city in California using DCS trees.
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]
Predicates to DCS Trees (Compositional Semantics)

$C_{i,j} = \text{set of DCS trees for span } [i, j]$
Predicates to DCS Trees (Compositional Semantics)

\(C_{i,j} = \text{set of DCS trees for span } [i, j]\)

\[\text{most populous city in California}\]

\[\text{city in California}\]
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]

- Most populous city in California
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span} \ [i, j] \]

\[ C_{i,k} \rightarrow \text{argmax} \rightarrow \text{population} \rightarrow \text{city} \rightarrow \text{loc} \to \text{CA} \]

\[ C_{k,j} \rightarrow \text{city} \rightarrow \text{loc} \to \text{CA} \]
Plan

What’s possible? $z \in \mathcal{Z}(x)$

What’s probable? $p(z \mid x, \theta)$

Learning $\theta$ from $(x, y)$ data
Log-linear Model

$z$: city $\rightarrow$ loc $\rightarrow$ CA

$x$: city $\rightarrow$ in $\rightarrow$ California
Log-linear Model

features(\(x, z\)) = \begin{pmatrix}
\end{pmatrix} \in \mathbb{R}^d
Log-linear Model

\[ z: \quad \text{city} \quad \text{loc} \quad \text{CA} \]

\[ x: \quad \text{city} \quad \text{in} \quad \text{California} \]

\[
\text{features}(x, z) = \begin{pmatrix}
\text{in} & \ldots & \text{loc} & : 1
\end{pmatrix} \in \mathbb{R}^d
\]
Log-linear Model

\[ \text{features}(x, z) = \begin{pmatrix} \text{in} & \cdots & \text{loc} & : & 1 \\ \text{city} & \text{1-1-loc} & : & 1 \end{pmatrix} \in \mathbb{R}^d \]
Log-linear Model

\[ \text{features}(x, z) = \begin{pmatrix} \text{in} & \cdots & \text{loc} & : & 1 \\ \text{city} & -1 & \text{loc} & : & 1 \\ \cdots \end{pmatrix} \in \mathbb{R}^d \]
Log-linear Model

\[ z: \text{city} \rightarrow \text{loc} \rightarrow \text{CA} \]

\[ x: \text{city} \rightarrow \text{in} \rightarrow \text{California} \]

\[ \text{features}(x, z) = \begin{pmatrix}
\text{in} & \cdots & \text{loc} & : & 1 \\
\text{city} & -1 & -1 & \text{loc} & : & 1 \\
\cdots & & & & \\
\end{pmatrix} \in \mathbb{R}^d \]

\[ \text{score}(x, z) = \text{features}(x, z) \cdot \theta \]
Log-linear Model

\[ \text{features}(x, z) = \left( \begin{array}{c} \text{in} \cdots \text{loc} : 1 \\ \text{city} \cdots \text{loc} : 1 \\ \vdots \end{array} \right) \in \mathbb{R}^d \]

\[ \text{score}(x, z) = \text{features}(x, z) \cdot \theta \]

\[ p(z \mid x, \theta) = \frac{e^{\text{score}(x, z)}}{\sum_{z' \in \mathcal{Z}(x)} e^{\text{score}(x, z')}} \]
Plan

- **What’s possible?** $z \in \mathcal{Z}(x)$
- **What’s probable?** $p(z \mid x, \theta)$
- **Learning** $\theta$ from $(x, y)$ data
Learning

Objective Function:

\[ p(y \mid z, w) \ p(z \mid x, \theta) \]

Interpretation: Semantic parsing
Learning

Objective Function:

$$\max_{\theta} \quad p(y \mid z, w) \ p(z \mid x, \theta)$$

Interpretation  Semantic parsing
Learning

Objective Function:

$$\max_{\theta} \sum_z p(y \mid z, w) p(z \mid x, \theta)$$

Interpretation Semantic parsing
Learning

Objective Function:

$$\max_{\theta} \sum_z p(y \mid z, w) p(z \mid x, \theta)$$

Interpretation Semantic parsing

EM-like Algorithm:

parameters $\theta$

$$(0, 0, \ldots, 0)$$
Learning

Objective Function:

$$\max_{\theta} \sum_z p(y \mid z, w) p(z \mid x, \theta)$$

Interpretation Semantic parsing

EM-like Algorithm:

parameters $\theta$

enumerate/score DCS trees

$(0, 0, \ldots, 0)$
Learning

Objective Function:

$$\max_{\theta} \sum_z p(y | z, w) p(z | x, \theta)$$

Interpretation: Semantic parsing

EM-like Algorithm:

parameters $\theta$

$$(0, 0, \ldots, 0)$$

enumerate/score DCS trees

$k$-best list:

- tree1 ✗
- tree2 ✗
- tree3 ✓
- tree4 ✗
- tree5 ✗
Learning

Objective Function:

\[
\max_\theta \sum_z p(y \mid z, w) p(z \mid x, \theta)
\]

Interpretation: Semantic parsing

EM-like Algorithm:

- parameters \( \theta \)
- enumerate/score DCS trees
- numerical optimization (L-BFGS)

\( \kappa \)-best list
- tree1 ×
- tree2 ×
- tree3 ✓
- tree4 ×
- tree5 ×
Learning

Objective Function:

$$\max_{\theta} \sum_z p(y \mid z, w) p(z \mid x, \theta)$$

Interpretation Semantic parsing

EM-like Algorithm:

parameters $\theta$

enumerate/score DCS trees

numerical optimization (L-BFGS)

$k$-best list

- tree3 ✔
- tree8 ✔
- tree6 ✗
- tree2 ✗
- tree4 ✗
Learning

Objective Function:

$$\max_{\theta} \sum_{z} p(y \mid z, w) p(z \mid x, \theta)$$

Interpretation Semantic parsing

EM-like Algorithm:

parameters $\theta$

(0.3, -1.4, ..., 0.6)

enumerate/score DCS trees

numerical optimization (L-BFGS)

$k$-best list

- tree3 ✓
- tree8 ✓
- tree6 ✗
- tree2 ✗
- tree4 ✗
Learning

Objective Function:

$$\max_{\theta} \sum_z p(y \mid z, w) p(z \mid x, \theta)$$

Interpretation  Semantic parsing

EM-like Algorithm:

parameters $\theta$

(0.3, −1.4, . . . , 0.6)

enumerate/score DCS trees

numerical optimization (L-BFGS)

$k$-best list

tree3 ✓
tree8 ✓
tree2 ✗
tree4 ✗
tree9 ✗
Outline

Representation

Learning

Experiments
US Geography Benchmark

Standard semantic parsing benchmark since 1990s
600 training examples, 280 test examples
US Geography Benchmark

Standard semantic parsing benchmark since 1990s
   600 training examples, 280 test examples

What is the highest point in Florida?

How many states have a city called Rochester?

What is the longest river that runs through a state that borders Tennessee?

Of the states washed by the Mississippi river which has the lowest point?

...
US Geography Benchmark

Standard semantic parsing benchmark since 1990s
600 training examples, 280 test examples

What is the highest point in Florida?
⇒ answer(A,highest(A,(place(A),loc(A,B),const(B,stateid(florida))))))

How many states have a city called Rochester?
⇒ answer(A,count(B,(state(B),loc(C,B),const(C,cityid(rochester,_))),A))

What is the longest river that runs through a state that borders Tennessee?
⇒ answer(A,longest(A,(river(A),traverse(A,B),state(B),next_to(B,C),const(C,stateid(tennessee))))))

Of the states washed by the Mississippi river which has the lowest point?
⇒ answer(A,lowest(B,(state(A),traverse(C,A),const(C,riverid(mississippi)),loc(B,A),place(B))))

...
US Geography Benchmark

Standard semantic parsing benchmark since 1990s
600 training examples, 280 test examples

What is the highest point in Florida?
⇒ Walton County

How many states have a city called Rochester?
⇒ 2

What is the longest river that runs through a state that borders Tennessee?
⇒ Missouri

Of the states washed by the Mississippi river which has the lowest point?
⇒ Louisiana

...

Supervision in past work: question + program
Supervision in this work: question + answer
### Input to Learning Algorithm

**Training data** (600 examples)

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the highest point in Florida?</td>
<td>Walton County</td>
</tr>
<tr>
<td>How many states have a city called Rochester?</td>
<td>2</td>
</tr>
<tr>
<td>What is the longest river that runs through a state that borders Tennessee?</td>
<td>Missouri</td>
</tr>
<tr>
<td>Of the states washed by the Mississippi river which has the lowest point?</td>
<td>Louisiana</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## Input to Learning Algorithm

### Training data (600 examples)

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the highest point in Florida?</td>
<td>Walton County</td>
</tr>
<tr>
<td>How many states have a city called Rochester?</td>
<td>2</td>
</tr>
<tr>
<td>What is the longest river that runs through a state that borders Tennessee?</td>
<td>Missouri</td>
</tr>
<tr>
<td>Of the states washed by the Mississippi river which has the lowest point?</td>
<td>Louisiana</td>
</tr>
</tbody>
</table>

### Lexicon (75 words)

<table>
<thead>
<tr>
<th>Term</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>city</td>
<td>city</td>
</tr>
<tr>
<td>state</td>
<td>state</td>
</tr>
<tr>
<td>mountain</td>
<td>mountain, peak</td>
</tr>
</tbody>
</table>

...
### Input to Learning Algorithm

**Training data** (600 examples)

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the highest point in Florida?</td>
<td>Walton County</td>
</tr>
<tr>
<td>How many states have a city called Rochester?</td>
<td>2</td>
</tr>
<tr>
<td>What is the longest river that runs through a state that borders Tennessee?</td>
<td>Missouri</td>
</tr>
<tr>
<td>Of the states washed by the Mississippi river which has the lowest point?</td>
<td>Louisiana</td>
</tr>
</tbody>
</table>

---

**Lexicon** (75 words)

<table>
<thead>
<tr>
<th>Word</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>city</td>
<td>city</td>
</tr>
<tr>
<td>state</td>
<td>state</td>
</tr>
<tr>
<td>mountain</td>
<td>mountain, peak</td>
</tr>
</tbody>
</table>

---

**Database**

<table>
<thead>
<tr>
<th>city</th>
<th>state</th>
<th>loc</th>
<th>border</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Francisco</td>
<td>California</td>
<td>Mount Shasta</td>
<td>Washington</td>
</tr>
<tr>
<td>Chicago</td>
<td>Boston</td>
<td>San Francisco</td>
<td>Oregon</td>
</tr>
<tr>
<td>Boston</td>
<td>Massachusetts</td>
<td>Boston</td>
<td>Idaho</td>
</tr>
</tbody>
</table>

---
Experiment 1

On $\text{GEO}$, 250 training examples, 250 test examples
Experiment 1

On Geo, 250 training examples, 250 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon (gen./spec.)</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGCR10</td>
<td>FunQL [Clarke et al., 2010]</td>
<td>✓ ✓</td>
<td>X</td>
</tr>
</tbody>
</table>

Test accuracy: 73.2%
Experiment 1

On Geo, 250 training examples, 250 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon (gen./spec.)</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGCR10</td>
<td>FunQL [Clarke et al., 2010]</td>
<td>✓ ✓</td>
<td>✓</td>
</tr>
<tr>
<td>DCS</td>
<td>our system</td>
<td>✓ x</td>
<td>✓</td>
</tr>
</tbody>
</table>

![Bar chart showing test accuracy]

73.2% for CGCR10
78.9% for DCS
## Experiment 1

On Geo, 250 training examples, 250 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon (gen./spec.)</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGCR10</td>
<td>FunQL [Clarke et al., 2010]</td>
<td>✓ ✓</td>
<td>×</td>
</tr>
<tr>
<td>DCS</td>
<td>our system</td>
<td>✓ ×</td>
<td>×</td>
</tr>
<tr>
<td>DCS⁺</td>
<td>our system</td>
<td>✓ ✓</td>
<td>×</td>
</tr>
</tbody>
</table>

### Test Accuracy

- CGCR10: 73.2%
- DCS: 78.9%
- DCS⁺: 87.2%
Experiment 2

On \texttt{GEO}, 600 training examples, 280 test examples
Experiment 2

On GEO, 600 training examples, 280 test examples

System Description  Lexicon  Logical forms

![Graph of test accuracy]
Experiment 2

On Geo, 600 training examples, 280 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>zC05</td>
<td>CCG [Zettlemoyer &amp; Collins, 2005]</td>
<td>✗ ✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Test accuracy

![Graph showing test accuracy for zC05 system with 79.3% accuracy]
Experiment 2

On Geo, 600 training examples, 280 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>zc05</td>
<td>CCG [Zettlemoyer &amp; Collins, 2005]</td>
<td>✗ ✗</td>
<td>✓ ✗</td>
</tr>
<tr>
<td>zc07</td>
<td>relaxed CCG [Zettlemoyer &amp; Collins, 2007]</td>
<td>✗ ✗</td>
<td>✓ ✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>79.3%</td>
</tr>
<tr>
<td>86.1%</td>
</tr>
</tbody>
</table>

Graph showing test accuracy for zc05 (79.3%) and zc07 (86.1%)
**Experiment 2**

On \textit{Geo}, 600 training examples, 280 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>zc05</td>
<td>CCG [Zettlemoyer &amp; Collins, 2005]</td>
<td>(\times)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>zc07</td>
<td>relaxed CCG [Zettlemoyer &amp; Collins, 2007]</td>
<td>(\times)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>KZGS10</td>
<td>CCG w/unification [Kwiatkowski et al., 2010]</td>
<td>(\times)</td>
<td>(\checkmark)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>zc05</td>
<td>79.3%</td>
</tr>
<tr>
<td>zc07</td>
<td>86.1%</td>
</tr>
<tr>
<td>KZGS10</td>
<td>88.9%</td>
</tr>
</tbody>
</table>
**Experiment 2**

On **Geo**, 600 training examples, 280 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>zc05</td>
<td>CCG [Zettlemoyer &amp; Collins, 2005]</td>
<td></td>
<td>✗ ✗</td>
</tr>
<tr>
<td>zc07</td>
<td>relaxed CCG [Zettlemoyer &amp; Collins, 2007]</td>
<td>✗ ✗</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>KZGS10</td>
<td>CCG w/unification [Kwiatkowski et al., 2010]</td>
<td>✗ ✗</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>DCS</td>
<td>our system</td>
<td>✓ ✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

![Bar Chart]

- **79.3%** for zc05
- **86.1%** for zc07
- **88.9%** for KZGS10
- **88.6%** for DCS
Experiment 2

On Geo, 600 training examples, 280 test examples

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Lexicon</th>
<th>Logical forms</th>
</tr>
</thead>
<tbody>
<tr>
<td>zc05</td>
<td>CCG [Zettlemoyer &amp; Collins, 2005]</td>
<td>✗ ✗</td>
<td>✓</td>
</tr>
<tr>
<td>zc07</td>
<td>relaxed CCG [Zettlemoyer &amp; Collins, 2007]</td>
<td>✗ ✗</td>
<td>✓</td>
</tr>
<tr>
<td>KZGS10</td>
<td>CCG w/unification [Kwiatkowski et al., 2010]</td>
<td>✗ ✗</td>
<td>✓</td>
</tr>
<tr>
<td>DCS</td>
<td>our system</td>
<td>✓ ✗</td>
<td>✗</td>
</tr>
<tr>
<td>DCS⁺</td>
<td>our system</td>
<td>✓ ✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

![Test accuracy chart]

- zc05: 79.3%
- zc07: 86.1%
- KZGS10: 88.9%
- DCS: 88.6%
- DCS⁺: 91.1%
Some Intuition on Learning
Some Intuition on Learning

1. search DCS trees (hard!)
2. numerical optimization

parameters $\theta$ \quad $k$-best lists
Some Intuition on Learning

(1) search DCS trees (hard!)

parameters $\theta$ $\rightarrow$ $k$-best lists

(2) numerical optimization

If no DCS tree on $k$-best list is correct, skip example in (2)
Some Intuition on Learning

(1) search DCS trees (hard!)

parameters $\theta$ \rightarrow k-best lists

(2) numerical optimization

If no DCS tree on $k$-best list is correct, skip example in (2)
Some Intuition on Learning

(1) search DCS trees (hard!)

parameters $\theta$  \rightarrow  $k$-best lists

(2) numerical optimization

If no DCS tree on $k$-best list is correct, skip example in (2)

Effect: automatic curriculum learning, learning improves search
Current Limitations
Current Limitations

Only using forward information

Execute program to get answer, but want to invert
Current Limitations

Only using forward information

Execute program to get answer, but want to invert

Non-identifiability of program

If all cities in database are in US, then

can’t distinguish \( \{ c : \text{city}(c) \} \) and \( \{ c : \text{city}(c) \land \text{loc}(c, \text{US}) \} \)
Current Limitations

Only using forward information
   Execute program to get answer, but want to invert

Non-identifiability of program
   If all cities in database are in US, then
      can’t distinguish \{c : \text{city}(c)\} and \{c : \text{city}(c) \land \text{loc}(c, \text{US})\}

Unknown facts: *How far is Los Angeles from Boston?*
   Database has no distance information
Current Limitations

Only using forward information
Execute program to get answer, but want to invert

Non-identifiability of program
If all cities in database are in US, then
can’t distinguish \{c : \text{city}(c)\} and \{c : \text{city}(c) \land \text{loc}(c, \text{US})\}

Unknown facts: *How far is Los Angeles from Boston?*
Database has no distance information

Unknown concepts: *What states are landlocked?*
Need to induce database view for \text{landlocked}(x) = \neg \text{border}(x, \text{ocean})
Conclusion

Goal: learn to answer questions from question/answer pairs
Conclusion

Goal: learn to answer questions from question/answer pairs

Empirical result:

DCS (no logical forms) \cong \text{existing systems (with logical forms)}
Conclusion

Goal: learn to answer questions from question/answer pairs

Empirical result:

DCS (no logical forms) \( \cong \) existing systems (with logical forms)

Conceptual contribution: DCS trees

- Trees: connects dependency syntax with efficient evaluation
Conclusion

Goal: learn to answer questions from question/answer pairs

Empirical result:
DCS (no logical forms) \( \approx \) existing systems (with logical forms)

Conceptual contribution: DCS trees
- Trees: connects dependency syntax with efficient evaluation
- Mark-Execute: unifying framework for handling scope
thank

2

1

you