Learning Dependency-Based Compositional Semantics

ACL 2011 – Portland, OR

Percy Liang  Michael I. Jordan  Dan Klein

Berkeley

N  L  P
The Big Picture

What is the most populous city in California?

System

Los Angeles
The Big Picture

What is the most populous city in California?

Database \rightarrow System

Los Angeles
The Big Picture

What is the most populous city in California?

Database → System

Los Angeles

Expensive: logical forms

[Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005]
[Wong & Mooney, 2007; Kwiatkowski et al., 2010]

What is the most populous city in California?
⇒ \( \text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{pop.}(x)) \)

How many states border Oregon?
⇒ \( \text{count}(\lambda x. \text{state}(x) \land \text{border}(x, \text{OR}) \)

...
What is the most populous city in California?
⇒ Los Angeles

How many states border Oregon?
⇒ 3

Expensive: logical forms
[Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005]
[Wong & Mooney, 2007; Kwiatkowski et al., 2010]

Cheap: answers
[Clarke et al., 2010]
[this work]

What is the most populous city in California?
⇒ \( \text{argmax}(\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}), \lambda x. \text{pop.}(x)) \)

How many states border Oregon?
⇒ count(\( \lambda x. \text{state}(x) \land \text{border}(x, \text{OR}) \))
...

What is the most populous city in California?
⇒ Los Angeles
How many states border Oregon?
⇒ 3
...

Challenges

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

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

What is the most populous city in California?

*Los Angeles*
Challenges

Computational: how to efficiently search exponential space?

What is the most populous city in California?

$\lambda x.\text{state}(x)$

Los Angeles
Challenges

Computational: how to efficiently search exponential space?

What is the most populous city in California?

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

Los Angeles
Challenges

**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*
Challenges

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
Challenges

Computational: how to efficiently search exponential space?

What is the most populous city in California?

population(CA)

Los Angeles
Challenges

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
Challenges

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

What is the most populous city in California?

Los Angeles
Challenges

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

Statistical: how to parametrize mapping from sentence to logical form?

What is the most populous city in California?

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

*most populous city in California*
New: Dependency-Based Compositional Semantics (DCS)

most populous city in California

most populous California

most

California
most populous city in California

```
<city>
  <population>
    <argmax>
      <CA>
    </argmax>
  </population>
  <loc>
    <CA>
  </loc>
```

New: Dependency-Based Compositional Semantics (DCS)
New: Dependency-Based Compositional Semantics (DCS)

most populous city in California

Diagram:

```
city
  | 1 1
  /   /
/     /
|     |
population  loc
  | 1 1
  /   /
/     /
|     |
argmax  CA
       | 2
       | 1
```
New: Dependency-Based Compositional Semantics (DCS)

most populous city in California

Los Angeles
New: Dependency-Based Compositional Semantics (DCS)

Advantages of DCS: nice computational, statistical, linguistic properties
Outline

Representation

Learning

Experiments
Basic DCS Trees

DCS tree

city
1
1
loc
2
1
CA
Database
6
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)
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)

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

```
Properties of DCS Trees
1
2
1
1
2
1
1
1
2
1
CA
border
state
loc
1
1
1
1
1
1
major
2
1
AZ
traverse
river
traverse
city
```
Properties of DCS Trees

Linguistics

syntactic locality

------------------------

Trees
Properties of DCS Trees

Linguistics

syntactic locality

Trees

efficient interpretation

Computation
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
  (most
   )
  (California
   ))
 (in
   ))
```
Divergence between Syntactic and Semantic Scope

*most populous city in California*

Syntax

```
city
down

most

populous

in

California
```

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

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**
- **City**
  - **Popular**
  - **In**
    - **Most**
    - **California**

**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**

![Syntax Tree]

- most
- California
- in
- city

**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

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

most populous city in California

Superlatives

- \( \text{argmax} \) population
- \( \text{loc} \) CA
- \( x_1 \) city
most populous city in California

Mark at syntactic scope
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
Solution: Mark-Execute

Some river traverses every city.

Execute at semantic scope

Mark at syntactic scope

Quantification (wide)
Some river traverses every city.

Execute at semantic scope

Mark at syntactic scope

Quantification (wide)

Analogy: Montague's quantifying in, Carpenter's scoping constructor
Graphical Model

database

w
Graphical Model

- database
- z
- capital
- CA
- Sacramento
- w
- y
**Graphical Model**

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

capital of California?

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

Parameters $\theta$

Database $w$

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

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

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

What's possible? $z \in \Omega$?

What's probable? $p(z | x, \theta)$

Learning $\theta$ to data

---

$\mathbf{x}$ capital of California?

parameters

$\theta$

$z$

capital

database

$w$

$y$

Sacramento
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)

\[ \text{argmax} \quad \text{CA} \]

*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)
   \( most \Rightarrow \text{argmax} \)

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

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

\( i \) \quad most \ populous \quad \quad \quad \quad \quad city \ in \ California \quad \quad \quad \quad \quad 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)

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

\[ C_{i,k} \subset C_{k,j} \]
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)

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

most populous city in California

\[ C_{i,k} \]

\[ C_{k,j} \]
Predicates to DCS Trees (Compositional Semantics)

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

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

\[ C_{i,k} \]

\[ C_{k,j} \]

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

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

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

\[
\text{most populous city in California} \quad \text{city} \quad \text{loc} \quad \text{population} \quad \text{argmax} \quad \text{CA} \\
\text{border} \quad \text{argmax} \quad \text{CA} \quad \text{city} \quad \text{loc} \quad \text{population} \\
\]

\[
C_{i,k} \quad C_{k,j} \\
\text{most populous} \quad \text{city in California} \\
i \quad k \quad j
\]
Predicates to DCS Trees (Compositional Semantics)

\[ C_{i,j} = \text{set of DCS trees for span } [i, j] \]
<table>
<thead>
<tr>
<th>Montague semantics</th>
<th>DCS</th>
</tr>
</thead>
</table>

## Comparison

<table>
<thead>
<tr>
<th>Montague semantics</th>
<th>Logical form</th>
<th>DCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda x.\text{city}(x) \land \text{loc}(x,\text{CA})$</td>
<td>DCS trees</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="DCS Tree" /></td>
</tr>
</tbody>
</table>

### Logical Form

- **Montague semantics**: $\lambda x.\text{city}(x) \land \text{loc}(x,\text{CA})$
- **DCS Tree**: ![DCS Tree](image)
## Comparison

<table>
<thead>
<tr>
<th>Montague semantics</th>
<th>DCS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical form</strong></td>
<td><strong>DCS trees</strong></td>
</tr>
<tr>
<td>$\lambda x. \text{city}(x) \land \text{loc}(x, \text{CA})$</td>
<td>$\text{city} - 1 - \text{loc} - 2 - 1 - \text{CA}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Lexicon</strong></th>
<th><strong>predicates</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>major</strong></td>
<td><strong>major</strong></td>
</tr>
</tbody>
</table>

Lexicon categories + lambda calculus predicates

$\text{NP/\text{NP}} : \lambda p.\lambda x.p(x) \land \text{major}(x)$
## Comparison

<table>
<thead>
<tr>
<th>Montague semantics</th>
<th>DCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical form</td>
<td></td>
</tr>
<tr>
<td>lambda calculus formulae</td>
<td>DCS trees</td>
</tr>
<tr>
<td>$\lambda x.\text{city}(x) \land \text{loc}(x, \text{CA})$</td>
<td></td>
</tr>
<tr>
<td>Lexicon</td>
<td></td>
</tr>
<tr>
<td>categories + lambda calculus</td>
<td>predicates</td>
</tr>
<tr>
<td>$\text{np/np} : \lambda p.\lambda x.p(x) \land \text{major}(x)$</td>
<td>$\text{major}$</td>
</tr>
<tr>
<td>Construction</td>
<td></td>
</tr>
<tr>
<td>combinator rules</td>
<td>dependency parsing</td>
</tr>
<tr>
<td>$f:a/b \quad g:b$</td>
<td>$f \quad i \quad j \quad g$</td>
</tr>
<tr>
<td>$\frac{}{f(g):a}$</td>
<td></td>
</tr>
</tbody>
</table>
### Comparison

<table>
<thead>
<tr>
<th>Montague semantics</th>
<th>DCS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical form</strong></td>
<td>lambda calculus formulae</td>
</tr>
<tr>
<td>( \lambda x.\text{city}(x) \land \text{loc}(x, \text{CA}) )</td>
<td></td>
</tr>
<tr>
<td><strong>Lexicon</strong></td>
<td>categories + lambda calculus</td>
</tr>
<tr>
<td>( \text{major} )</td>
<td>( \lambda p.\lambda x.p(x) \land \text{major}(x) )</td>
</tr>
<tr>
<td><strong>Construction</strong></td>
<td>combinator rules</td>
</tr>
</tbody>
</table>
|                     | \[
\begin{array}{c}
\text{f}:a/b \\
\text{g}:b \\
\hline
\text{f(g)}:a
\end{array}
\] | \( \text{f}i\text{jg} \) |
| **Goal**           | generate exact set | generate overapproximation |
## Comparison

<table>
<thead>
<tr>
<th>Montague semantics</th>
<th>DCS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical form</strong></td>
<td>lambda calculus formulae</td>
</tr>
<tr>
<td>( \lambda x. \text{city}(x) \land \text{loc}(x, \text{CA}) )</td>
<td>( \text{city} - 1 \rightarrow 1 \rightarrow \text{loc} - 2 \rightarrow 1 \rightarrow \text{CA} )</td>
</tr>
<tr>
<td><strong>Lexicon</strong></td>
<td>categories + lambda calculus predicates</td>
</tr>
<tr>
<td><strong>major</strong></td>
<td>NP/NP : ( \lambda p. \lambda x. p(x) \land \text{major}(x) ) major</td>
</tr>
<tr>
<td><strong>Construction</strong></td>
<td>combinator rules</td>
</tr>
<tr>
<td>( f : a/b ) \hspace{1cm} ( g : b ) \hspace{1cm} ( f(g) : a )</td>
<td>( f \rightarrow i \rightarrow j \rightarrow g )</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>generate exact set</td>
</tr>
</tbody>
</table>

DCS allows more flexibility \( \Rightarrow \) simpler
Plan

- What's possible? $z \in \Omega$?
- What's probable? $p(z | x, \theta)$
- Learning $\theta$ to data

parameters

$\theta$

$z$

capital

$w$

database

$y$

Sacramento

$x$

capital of California?
Log-linear Model

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

\[ x: \text{city} \rightarrow \text{in} \rightarrow \text{California} \]
Log-linear Model

\[
\text{features}(x, z) = \begin{pmatrix}
\end{pmatrix}
\]
Log-linear Model

\[
\text{features}(x, z) = \begin{pmatrix} \text{in} \quad \cdots \quad \text{loc} \end{pmatrix}
\]
Log-linear Model

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

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

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

\[
\begin{align*}
  z: & \quad \text{city} \quad \text{loc} \quad \text{CA} \\
  \quad & \quad 1 \quad 1 \quad 2 \quad 1 \\
  x: & \quad \text{city} \quad \text{in} \quad \text{California} \\

\text{features}(x, z) &= \left( \begin{array}{c}
  \text{in} \\
  \cdots \\
  \text{city} \\
  \text{loc}
\end{array} \right) \\

\text{score}(x, z) &= \text{features}(x, z) \cdot \theta
\end{align*}
\]
Log-linear Model

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

\[ \text{features}(x, z) = \begin{pmatrix}
\text{in} & \ldots & \text{loc} \\
\text{city} & 1 & 1 & \text{loc} \\
\ldots & & & \\
\end{pmatrix} \]

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

\[ p(z \mid x, \theta) \propto e^{\text{score}(x, z)} \]
Plan

- What's possible? $z \in \Omega$?
- What's probable? $p(z | x, \theta)$
- **Learning** $\theta$ to data

**Parameters**

\[
\theta \rightarrow z \rightarrow \text{capital} \rightarrow y \rightarrow \text{Sacramento}
\]

**Database**

$w \rightarrow y$
Learning

Objective:

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

Interpretation Semantic parsing
Learning

Objective:

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

Interpretation Semantic parsing
Learning

Objective:

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

Interpretation Semantic parsing
Learning

Objective:

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

Interpretation Semantic parsing

EM-like Algorithm:

parameters $\theta$

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

Objective:

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

Interpretation: Semantic parsing

EM-like Algorithm:

parameters \( \theta \)

enumerate/score DCS trees

(0, 0, ..., 0)
Learning

Objective:

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

Interpretation Semantic parsing

EM-like Algorithm:

parameters $\theta$

(0, 0, ..., 0)

enumerate/score DCS trees

tree1

tree2

tree3

tree4

tree5

$k$-best list

$X$

$\checkmark$
Learning

Objective:

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

Interpretation Semantic parsing

EM-like Algorithm:

parameters \( \theta \) (0.2, -1.3, \ldots, 0.7)

enumerate/score DCS trees

numerical optimization (L-BFGS)

\( k \)-best list

tree1 \( \times \)
tree2 \( \times \)
tree3 \( \checkmark \)
tree4 \( \times \)
tree5 \( \times \)
Learning

Objective:

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

Interpretation  Semantic parsing

EM-like Algorithm:

parameters $\theta$

(0.2, −1.3, ..., 0.7)

enumerate/score DCS trees

numerical optimization (L-BFGS)

$k$-best list

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

Objective:

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

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:

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

Interpretation  Semantic parsing

EM-like Algorithm:

parameters \( \theta \) k-best list

(0.3, -1.4, \ldots, 0.6)

enumerate/score DCS trees

numerical optimization (L-BFGS)

\[
\begin{array}{c}
\text{tree3} \checkmark \\
\text{tree8} \checkmark \\
\text{tree2} \times \\
\text{tree4} \times \\
\text{tree9} \times \\
\end{array}
\]
Outline

Representation

Learning

Experiments
Experimental Setup

Benchmarks:

US Geography (GEO): *What is the capital of California?* ⇒ *Sacramento*

Job Search (JOBS): *Show me jobs that require C++* ⇒ *Job1, Job5, Job9*
Experimental Setup

Benchmarks:

US Geography (Geo): What is the capital of California? ⇒ Sacramento

Job Search (Jobs): Show me jobs that require C++ ⇒ Job1, Job5, Job9

Evaluation: answer accuracy
Experimental Setup

Benchmarks:

US Geography (Geo): What is the capital of California? $\Rightarrow$ Sacramento
Job Search (Jobs): Show me jobs that require C++ $\Rightarrow$ Job1, Job5, Job9

Evaluation: answer accuracy

Possible types of supervision:

Answers: Los Angeles
Experimental Setup

Benchmarks:

- US Geography (Geo): *What is the capital of California?* ⇒ *Sacramento*
- Job Search (Jobs): *Show me jobs that require C++* ⇒ *Job1, Job5, Job9*

Evaluation: answer accuracy

Possible types of supervision:

- Answers: *Los Angeles*
- General lexicon (domain-independent): *no* ⇒ *not*
- Specific lexicon (domain-dependent): *city* ⇒ *city*
Experimental Setup

Benchmarks:

US Geography (Geo): What is the capital of California? ⇒ Sacramento

Job Search (Jobs): Show me jobs that require C++ ⇒ Job1, Job5, Job9

Evaluation: answer accuracy

Possible types of supervision:

Answers: Los Angeles

General lexicon (domain-independent): no ⇒ not

Specific lexicon (domain-dependent): city ⇒ city

Logical forms: argmax(λx.city(x) ∧ loc(x,CA), λx.population(x))
Experiment 1

On 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</td>
<td>✓ ✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
</tr>
<tr>
<td>95</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>75</td>
</tr>
</tbody>
</table>

- **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>✓ ✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

![Test Accuracy Chart]

- CGCR10: 73.2%
- DCS: 78.9%
### 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>
<tr>
<td>DCS</td>
<td>our system</td>
<td>✓ X</td>
<td>X</td>
</tr>
<tr>
<td>DCS⁺</td>
<td>our system</td>
<td>✓ ✓</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>test accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGCR10</td>
<td>73.2%</td>
</tr>
<tr>
<td>DCS</td>
<td>78.9%</td>
</tr>
<tr>
<td>DCS⁺</td>
<td>87.2%</td>
</tr>
</tbody>
</table>
Experiment 2

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

On Geo, 600 training examples, 280 test examples

System Description

Lexicon Logical forms

Test accuracy
Experiment 2

On Geo, 600 training examples, 280 test examples

System Description Lexicon Logical forms
zc05 CCG [Zettlemoyer & Collins, 2005]  ✗ ✗ ✔

79.3% 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>
<tr>
<td>zc07</td>
<td>relaxed CCG [Zettlemoyer &amp; Collins, 2007]</td>
<td>❌ ❌</td>
<td>✓ ✓</td>
</tr>
</tbody>
</table>

![Bar chart showing test accuracy](chart.png)

- zc05: 79.3%
- zc07: 86.1%
## 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>xx</td>
<td>✓</td>
</tr>
<tr>
<td>zc07</td>
<td>relaxed CCG [Zettlemoyer &amp; Collins, 2007]</td>
<td>xx</td>
<td>✓</td>
</tr>
<tr>
<td>KZGS10</td>
<td>CCG w/unification [Kwiatkowski et al., 2010]</td>
<td>xx</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Test Accuracy

<table>
<thead>
<tr>
<th>System</th>
<th>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>

<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>
<tr>
<td>DCS</td>
<td>88.6%</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>
<tr>
<td>DCS⁺</td>
<td>our system</td>
<td>✓ ✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

![Graph showing test accuracy](image)
Some Intuition on Learning
Some Intuition on Learning

parameters $\theta$

(1) search DCS trees (hard!)

(2) numerical optimization

$k$-best lists
Some Intuition on Learning

Parameters $\theta$ → (1) search DCS trees (hard!) → $k$-best lists

(2) numerical optimization

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

parameters $\theta$

(1) search DCS trees (hard!)

(2) numerical optimization

$k$-best lists

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

(1) search DCS trees (hard!)

parameters $\theta$ \quad \longleftrightarrow \quad k\text{-}best lists

(2) numerical optimization

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

Effect: natural bootstrapping, learning improves search
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) > existing systems (with logical forms)
Conclusion

Goal: learn to answer questions from question/answer pairs

Empirical result:
DCS (no logical forms) > 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) > 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