Sequence-to-Action: End-to-End Semantic Graph Generation for Semantic Parsing

Bo Chen, Le Sun, Xianpei Han
Institute of Software, Chinese Academy of Sciences
Task: Semantic Parsing

- Translate natural language sentences to meaning representations, e.g., logical forms.
Task: Semantic Parsing

- Translate natural language sentences to meaning representations, e.g., logical forms.

Sentence: Which city was Barack Obama born in?
Task: Semantic Parsing

- Translate natural language sentences to meaning representations, e.g., logical forms.

Sentence: Which city was Barack Obama born in?
Task: Semantic Parsing

- Translate natural language sentences to meaning representations, e.g., logical forms.

**Sentence:** Which city was Barack Obama born in?

**Logical form:** $\lambda x. \text{City}(x) \land \text{PlaceOfBirth}(\text{Barack\_Obama}, x)$
Outline

- Motivation
- Sequence-to-Action
- Experiments & Conclusion
Two Lines of Work in Semantic Parsing
Two Lines of Work in Semantic Parsing

Semantic Graph Based
Two Lines of Work in Semantic Parsing

Semantic Graph Based

- Use semantic graphs to represent sentence meanings
Two Lines of Work in Semantic Parsing

Semantic Graph Based

- Use semantic graphs to represent sentence meanings
- Semantic parsing as semantic graph matching or staged semantic query graph generation
Two Lines of Work in Semantic Parsing

- Use semantic graphs to represent sentence meanings
- Semantic parsing as semantic graph matching or staged semantic query graph generation

[Reddy et al., 2014,2016,2017]
[Yih et al., 2015]
[Bast and Haussmann, 2015]
Two Lines of Work in Semantic Parsing

Semantic Graph Based

- Use semantic graphs to represent sentence meanings
- Semantic parsing as semantic graph matching or staged semantic query graph generation

[Reddy et al., 2014,2016,2017]
[Yih et al., 2015]
[Bast and Haussmann, 2015]
Two Lines of Work in Semantic Parsing

Semantic Graph Based

- Use semantic graphs to represent sentence meanings
- Semantic parsing as semantic graph matching or staged semantic query graph generation
  
  [Reddy et al., 2014,2016,2017]
  [Yih et al., 2015]
  [Bast and Haussmann, 2015]

Sequence-to-Sequence Based

- Linearize logical forms
Two Lines of Work in Semantic Parsing

**Semantic Graph Based**
- Use semantic graphs to represent sentence meanings
- Semantic parsing as semantic graph matching or staged semantic query graph generation
  
  [Reddy et al., 2014, 2016, 2017]
  [Yih et al., 2015]
  [Bast and Haussmann, 2015]

**Sequence-to-Sequence Based**
- Linearize logical forms
- Semantic parsing as a sequence-to-sequence problem
Two Lines of Work in Semantic Parsing

Semantic Graph Based
- Use semantic graphs to represent sentence meanings
- Semantic parsing as semantic graph matching or staged semantic query graph generation

[Reddy et al., 2014, 2016, 2017]
[Yih et al., 2015]
[Bast and Haussmann, 2015]

Sequence-to-Sequence Based
- Linearize logical forms
- Semantic parsing as a sequence-to-sequence problem

[Dong and Lapata, 2016]
[Jia and Liang, 2016]
[Xiao et al., 2016]
[Rabinovich et al., 2017]
Two Lines of Work in Semantic Parsing

- Semantic Graph Based
- Sequence-to-Sequence Based
Two Lines of Work in Semantic Parsing

**Strengths**

- use semantic graphs to represent sentence meanings, no need for lexicons and grammars

Semantic Graph Based

Sequence-to-Sequence Based
Two Lines of Work in Semantic Parsing

- **Strengths**
  - use semantic graphs to represent sentence meanings, no need for lexicons and grammars

- **Challenges**
  - Hard to model semantic graph construction process
Two Lines of Work in Semantic Parsing

Semantic Graph Based

- **Strengths**
  - use semantic graphs to represent sentence meanings, no need for lexicons and grammars

- **Challenges**
  - Hard to model semantic graph construction process

Sequence-to-Sequence Based

- **Strengths**
  - End-to-end
  - Powerful prediction ability
Two Lines of Work in Semantic Parsing

Semantic Graph Based

- **Strengths**
  - Use semantic graphs to represent sentence meanings, no need for lexicons and grammars

- **Challenges**
  - Hard to model semantic graph construction process

Sequence-to-Sequence Based

- **Strengths**
  - End-to-end
  - Powerful prediction ability

- **Challenges**
  - Hard to capture structure information
  - Ignore the relatedness to KB
Seq2Act: synthesizes their advantages
Seq2Act: synthesizes their advantages

- Use semantic graphs to represent sentence meanings
  - tight-coupling with knowledge bases
Seq2Act: synthesizes their advantages

- Use semantic graphs to represent sentence meanings
  - tight-coupling with knowledge bases

- Leverage the powerful prediction ability of RNN models
  - End-to-End
Seq2Act: end-to-end semantic graph generation
Which states border Texas?
Which states border Texas?
Seq2Act: end-to-end semantic graph generation

Which states border Texas?
Which states border Texas?
Which states border Texas?
Which states border Texas?

Seq2Act: end-to-end semantic graph generation

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st

Semantic graph:
- Node A
- Type state
- Next to relationship
- Texas:st

Sentence:
Which states border Texas?
Which states border Texas?
Which states border Texas?
Which states border Texas?

Seq2Act: end-to-end semantic graph generation

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to
Action 5: return
Which states border Texas?
Which states border Texas?
Seq2Act: end-to-end semantic graph generation

Which states border Texas?

Our contribution

Sequence-to-Action

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to
Action 5: return

Semantic graph

Return

Type state

Next_to
texas:st
Outline

- Motivation
- Sequence-to-Action
- Experiments & Conclusion
Overview of Our Method

Sentence -> Sequence-to-Action RNN Model

Which states border Texas?

add_variable: A
add_type: state
arg_node: A
add_entity: texas:st
add_edge: next_to
arg_node: A
arg_node: texas:st
return: A

return

A

KB

Semantic Graph

Constraints

generate

Construct

next_to

type

state

texas:st
Overview of Our Method

Which states border Texas?

Sequence-to-Action RNN Model

Sentence

Generate

Constraints

Construct

Action Sequence

Semantic Graph

KB

add_variable: A
add_type: state
arg_node: A
add_entity: texas:st
add_edge: next_to
arg_node: A
arg_node: texas:st
return: A

return
type
state
texas:st
Overview of Our Method

```
Sequence-to-Action RNN Model
Sentence
Action Sequence
Semantic Graph
Generate
Constraints
Construct
KB

Which states border Texas?

add_variable: A
add_type: state
arg_node: A
add_entity: texas:st
add_edge: next_to
arg_node: A
arg_node: texas:st
return: A
```

```
return A
type state
next_to

texas:st
```
Overview of Our Method

Sequence-to-Action RNN Model

Sentence

<br>

Which states border Texas?

<table>
<thead>
<tr>
<th>Action</th>
<th>Argument</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_variable</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>add_type</td>
<td>state</td>
<td></td>
</tr>
<tr>
<td>arg_node</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>add_entity</td>
<td>texas:st</td>
<td></td>
</tr>
<tr>
<td>add_edge</td>
<td>next_to</td>
<td></td>
</tr>
<tr>
<td>arg_node</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>arg_node</td>
<td>texas:st</td>
<td></td>
</tr>
<tr>
<td>return</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

KB

Semantic Graph

Constraints

Construct

return

A

type

state

texas:st

next_to

Overview of Our Method

Sequence-to-Action RNN Model

Sentence

Which states border Texas?

Add variable: A
Add type: state
Arg node: A
Add entity: texas:st
Add edge: next_to
Arg node: A
Arg node: texas:st
Return: A

Constraints

Generate

Action Sequence

Construct

Semantic Graph

KB
Major components of Our Model

- Sequence-to-Action RNN Model
- Sentence
- Action Sequence
- Semantic Graph
- Generate
- Constraints
- Construct
- KB

Which states border Texas?

add_variable: A
add_type: state
arg_node: A
add_entity: texas:st
add_edge: next_to
arg_node: A
arg_node: texas:st
return: A

Type state
next_to
texas:st
Major components of Our Model (1)

Sequence-to-Action RNN Model

Sentence

Which states border Texas?

add_variable: A
add_type: state
arg_node: A
add_entity: texas:st
add_edge: next_to
arg_node: A
arg_node: texas:st
return: A

Constraints

Generate

Action Sequence

Construct

Semantic Graph

KB

Action set

1

return

A

type

state

texta:st

next_to

A
Major components of Our Model (2)

Sequence-to-Action RNN Model

Sentence → Which states border Texas?

Action Sequence

Generate Constraints

Semantic Graph

KB

add_variable: A
add_type: state
arg_node: A
add_entity: texas:st
add_edge: next_to
arg_node: A
arg_node: texas:st
return: A
Major components of Our Model (3)

1. **Sentence**
   - Which states border Texas?

2. **Sequence-to-Action RNN Model**
   - add_variable: A
   - add_type: state
   - arg_node: A
   - add_entity: texas:st
   - add_edge: next_to
   - arg_node: A
   - arg_node: texas:st
   - return: A

3. **Constraints**
   - Generate
   - action_sequence
     - arg_node: texas:st
     - return: A

4. **Semantic Graph**
   - KB

Action set
Which states border Texas?

Action set:

- **add_variable**: A
- **add_type**: state
- **arg_node**: A
- **add_entity**: texas:st
- **add_edge**: next_to
- **arg_node**: A
- **arg_node**: texas:st
- **return**: A
Action Set

- Define atom actions involved in semantic graph construction
Action Set

- Define atom actions involved in semantic graph construction

Which states border Texas?

Node: A (variable), texas:st (entity), state (type)
Edge: next_to
Return node: A
Action Set

- Add variable node
  - E.g., A
- Add entity node
  - E.g., texas:st
- Add type node
  - E.g., state
- Add edge
  - E.g., next_to
- Operation action
  - E.g., argmax, argmin, count
- Argument action
  - For type node, edge and operation

Sentence: Which river runs through the most states?
Semantic Graph:
Action Sequence:

<table>
<thead>
<tr>
<th>Structure</th>
<th>Semantic</th>
<th>Arg</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_operation</td>
<td>most</td>
<td></td>
</tr>
<tr>
<td>add_variable</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>add_type</td>
<td>river</td>
<td></td>
</tr>
<tr>
<td>add_variable</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>add_type</td>
<td>state</td>
<td></td>
</tr>
<tr>
<td>add_edge</td>
<td>traverse</td>
<td>A, B</td>
</tr>
<tr>
<td>end_operation</td>
<td>most</td>
<td>A, B</td>
</tr>
<tr>
<td>return</td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>
Which states border Texas?

- **Add variable**: A
- **Add type**: state
- **Arg node**: A
- **Add entity**: texas:st
- **Add edge**: next_to
- **Arg node**: A
- **Arg node**: texas:st
- **Return**: A

**Sequence-to-Action RNN Model**

**Action sequence**

**Semantic Graph**

**Encoder-Decoder Model**
Encoder-Decoder Model
Encoder-Decoder Model

Typical encoder-decoder model (bi-LSTM with attention)
Encoder-Decoder Model

Typical encoder-decoder model (bi-LSTM with attention)

Action embedding
add_edge : next_to
add_edge : loc
Action Embedding

Structure part

```
add_edge : next_to
add_edge : loc
```
Action Embedding

Structure part: add_edge : next_to
Semantic part: add_edge : loc
Action Embedding

add_edge : next_to
add_edge : loc

Structure part       Semantic part
Action Embedding

Structure part  Semantic part

Φ (add_edge:next_to) = [Φ (add_edge); Φ (next_to)]
Structure & Semantic Constraints

- **Sequence-to-Action RNN Model**
  - Sentence: *Which states border Texas?*
  - Action Sequence:
    - add_variable: A
    - add_type: state
    - arg_node: A
    - add_entity: texas:st
    - add_edge: next_to
    - arg_node: A
    - arg_node: texas:st
    - return: A

- **Constraints**
  - KB: Semantic Graph

1. **Action set**
2. **Sequence**
3. **Generate**
   - Constraints
5. **Construct**
   - Action Sequence
6. **Semantic Graph**
Structure & Semantic Constraints

- Structure constraints
  - Ensure action sequence will form a connected acyclic graph

- Semantic constraints
  - Ensure the constructed graph must follow the schema of knowledge bases
**Sentence:** Which states border Texas?

**Partial Semantic Graph:**

```
A  type  state
   \next_to
  texas:st
```

<table>
<thead>
<tr>
<th>Generated Actions</th>
<th>Structure</th>
<th>Semantic</th>
<th>Arg</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_variable</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>add_type</td>
<td>state</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>add_entity</td>
<td>texas:st</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Structure</th>
<th>Semantic</th>
<th>Arg</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_type</td>
<td>city</td>
<td>texas:st</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>add_edge</td>
<td>loc</td>
<td>A, texas:st</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>add_edge</td>
<td>next_to</td>
<td>A, A</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>add_edge</td>
<td>next_to</td>
<td>A, texas:st</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Action 1: violate type conflict
Action 2: violate selectional preference constraint
Action 3: structure constraint
Action 4: YES
Outline

- Motivation
- Sequence-to-Action
- Experiments & Conclusion
Experiments

- Datasets: GEO [Zelle and Mooney, 1996], ATIS [He and Young, 2005], OVERNIGHT [Wang et al., 2015b]

- We generate the action sequences from logical forms automatically.

what is the population of illinois?

```plaintext
add_node:-::B add_node:-::A add_edge:-::_population arg_node:-::B arg_node:-::A add_entity_node:-::illinois:=:state arg_node:-::B return:-::A
```
Baselines

- **Traditional Methods**
  - Zettlemoyer and Collins, 2005
  - Zettlemoyer and Collins, 2007
  - Liang et al., 2011
  - Zhao et al., 2015
  - Wang et al., 2015

- **Sequence-to-Sequence Models**
  - Dong and Lapata, 2016
  - Jia and Liang, 2016
  - Xiao et al., 2016
  - Rabinovich et al., 2017
## Competitive performance on three datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SOTA</th>
<th>SOTA without extra resources</th>
<th>Our full model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>91.1 [Liang et al., 2011]</td>
<td>89.9 [zhao et al., 2015]</td>
<td>89.9</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.5</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>79.0</td>
</tr>
</tbody>
</table>
### Competitive performance on three datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SOTA</th>
<th>SOTA without extra resources</th>
<th>Our full model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>91.1[^1]</td>
<td>89.9[^2]</td>
<td>89.9</td>
</tr>
<tr>
<td></td>
<td>[Liang et al., 2011]</td>
<td>[zhao et al., 2015]</td>
<td></td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9[^3]</td>
<td>85.9</td>
<td>85.5</td>
</tr>
<tr>
<td></td>
<td>[Rabinovich et al., 2017]</td>
<td>[Rabinovich et al., 2017]</td>
<td></td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5[^4]</td>
<td>75.8</td>
<td>79.0</td>
</tr>
<tr>
<td></td>
<td>[Jia and Liang, 2016]</td>
<td>[Jia and Liang, 2016]</td>
<td></td>
</tr>
</tbody>
</table>

[^1]: Liang et al., 2011
[^2]: zhao et al., 2015
[^3]: Rabinovich et al., 2017
[^4]: Jia and Liang, 2016
# Competitive performance on three datasets

<table>
<thead>
<tr>
<th></th>
<th>SOTA</th>
<th>SOTA without extra resources</th>
<th>Our full model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>91.1 [Liang et al., 2011]</td>
<td>89.9 [zhao et al., 2015]</td>
<td>89.9</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.5</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>79.0</td>
</tr>
</tbody>
</table>
Competitive performance on three datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SOTA without extra resources</th>
<th>Our full model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>89.9 (Liang et al., 2011)</td>
<td>89.9</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 (Rabinovich et al., 2017)</td>
<td>85.5</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 (Jia and Liang, 2016)</td>
<td>79.0</td>
</tr>
</tbody>
</table>

Need to design complex grammars
## Competitive performance on three datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SOTA</th>
<th>SOTA without extra resources</th>
<th>Our full model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>91.1 [Liang et al., 2011]</td>
<td>89.9 [zhao et al., 2015]</td>
<td>89.9</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.5</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>79.0</td>
</tr>
</tbody>
</table>
## Seq2Act outperforms Seq2Seq

<table>
<thead>
<tr>
<th></th>
<th>Seq2Seq SOTA</th>
<th>Seq2Seq SOTA without extra resources</th>
<th>Seq2Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>89.3 [Jia and Liang, 2016]</td>
<td>87.1 [Dong and Lapata, 2016]</td>
<td>87.5</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>84.6</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>78.0</td>
</tr>
</tbody>
</table>
Seq2Act outperforms Seq2Seq

<table>
<thead>
<tr>
<th></th>
<th>Seq2Seq SOTA</th>
<th>Seq2Seq SOTA without extra resources</th>
<th>Seq2Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>89.3 [Jia and Liang, 2016]</td>
<td>87.1 [Dong and Lapata, 2016]</td>
<td>87.5</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>84.6</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>78.0</td>
</tr>
</tbody>
</table>
Seq2Act outperforms Seq2Seq

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Seq2Seq SOTA</th>
<th>Seq2Seq SOTA without extra resources</th>
<th>Seq2Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>89.3 [Jia and Liang, 2016]</td>
<td>87.1 [Dong and Lapata, 2016]</td>
<td>87.5</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>84.6</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>78.0</td>
</tr>
</tbody>
</table>
Seq2Act outperforms Seq2Seq

<table>
<thead>
<tr>
<th></th>
<th>Seq2Seq SOTA</th>
<th>Seq2Seq SOTA without extra resources</th>
<th>Seq2Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>89.3 [Jia and Liang, 2016]</td>
<td>87.5 [Dong and Lapata, 2016]</td>
<td>87.5</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>85.9 [Rabinovich et al., 2017]</td>
<td>84.6</td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5 [Jia and Liang, 2016]</td>
<td>75.8 [Jia and Liang, 2016]</td>
<td>78.0</td>
</tr>
</tbody>
</table>

*Need to design complex grammars*
Seq2Act outperforms Seq2Seq

<table>
<thead>
<tr>
<th></th>
<th>Seq2Seq SOTA</th>
<th>Seq2Seq SOTA without extra resources</th>
<th>Seq2Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEO</td>
<td>89.3</td>
<td>87.1</td>
<td>87.5</td>
</tr>
<tr>
<td></td>
<td>[Jia and Liang, 2016]</td>
<td>[Dong and Lapata, 2016]</td>
<td></td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9</td>
<td>85.9</td>
<td>84.6</td>
</tr>
<tr>
<td></td>
<td>[Rabinovich et al., 2017]</td>
<td>[Rabinovich et al., 2017]</td>
<td></td>
</tr>
<tr>
<td>OVERNIGHT</td>
<td>77.5</td>
<td>75.8</td>
<td>78.0</td>
</tr>
<tr>
<td></td>
<td>[Jia and Liang, 2016]</td>
<td>[Jia and Liang, 2016]</td>
<td></td>
</tr>
</tbody>
</table>
Seq2Act+C1 outperforms Seq2Act

- **GEO**: Seq2Act 87.5, Seq2Act+C1 88.2
- **ATIS**: Seq2Act 84.6, Seq2Act+C1 85
- **OVERNIGHT**: Seq2Act 78, Seq2Act+C1 78.4

C1: Structure Constraints
Seq2Act+C1+C2 outperforms Seq2Act+C1

C1: Structure Constraints
C2: Semantic Constraints
Average Length of Logical Forms and Action Sequences

- **GEO**
  - Average len of logical forms: 28.2
  - Average len of action sequences: 18.2
  - Reduction: 35.5%

- **ATIS**
  - Average len of logical forms: 28.4
  - Average len of action sequences: 25.8
  - Reduction: 9.2%

- **OVERNIGHT**
  - Average len of logical forms: 46.6
  - Average len of action sequences: 33.3
  - Reduction: 28.5%
Error Analysis

- **Un-covered Sentence Structure**
  - *Iowa borders how many states?* (Formal Form: *How many states does Iowa border?*)

- **Under Mapping**
  - *Please show me first class flights from indianapolis to memphis one way leaving before 10am*
Conclusion

- Sequence-to-Action: End-to-End Semantic Graph Generation
  - Representation ability of semantic graphs
  - Sequence prediction ability of RNN models

- Achieve competitive results on GEO, ATIS and OVERNIGHT
Future work

- Weak supervised learning algorithm for Seq2Act
  - So our method can be applied to (q, a) pair datasets such as WebQuestions

- Apply Seq2Act model to other parsing tasks (e.g., AMR parsing)
Thanks!

Data and code available: https://github.com/dongpobeyond/Seq2Act

Email: chenbo42424@gmail.com