Coarse-to-Fine Decoding for Neural Semantic Parsing
July 16, 2018

Li Dong and Mirella Lapata
Semantic Parsing

Mapping **natural language** to **structured representations**

Human-friendly -> Computer-friendly

all flights from dallas before 10am

(lambd (0 (and (flight (0) (from (0 dallas:ci)) (< (departure_time (0) 1000:ti))))

Example from ATIS (Kwiatkowski et al., 2011)
**Neural Semantic Parsing**

- **Sequence decoder** (Jia and Liang, 2016; Dong and Lapata, 2016; Ling et al., 2016; Iyer et al., 2017)

- **Syntactically-constrained decoder** (Dong and Lapata, 2016; Xiao et al., 2016; Alvarez-Melis and Jaakkola, 2017; Yin and Neubig, 2017; Cheng et al., 2017; Krishnamurthy et al., 2017; Rabinovich et al., 2017; Xu et al., 2017)
This Work

all flights from dallas before 10am

Meaning Sketch
(lambda#2 (and flight@1 from@2 (< departure time@1 ? ) ) )

&

Low-level Details
(e.g., arguments and variable names)

(lambda $0 e (and (flight $0) (from $0 dallas:ci) (< (departure_time $0) 1000:ti)))
Python code example

```python
if length of bits is lesser than integer 3 or second element of bits is not equal to string 'as',

if len(NAME) < NUMBER or NAME[NUMBER] != STRING:

    if len(bits) < 3 or bits[1] != 'as':
```

SQL example

```
What record company did conductor Mikhail Snitko record for after 1996?

WHERE > AND =

SELECT Record Company WHERE (Year of Recording > 1996)
AND (Conductor = Mikhail Snitko)
```
**Meaning Sketch**

- **Disentangle** high-level from low-level semantics
  - Model meaning at different levels of granularity
- **More compact** meaning representation
  - Length: 21.1 → 9.2 (on ATIS)
- **Explicit sharing** coarse structure
  - For examples that have the same basic meaning
- **Provide global** context to fine meaning decoder
  - Know what the basic meaning of input looks like
Method

Sketch Decoding

Input Encoding

(lambda#2 (and flight@1 (< departure _time@1 ? ) ) ) ) ) ) ) </s>

\( \text{e}_1 \rightarrow \text{e}_2 \rightarrow \text{e}_3 \rightarrow \text{e}_4 \)

all flights before ti0
Method

Sketch Encoding

Sketch Decoding

Input Encoding

\[ \text{all} \rightarrow \text{flights} \rightarrow \text{before} \rightarrow \text{ti0} \]
Method
Sketch constrains the decoding output

- Example 1: one augment is missing
  
  \[ \text{flight@1} \rightarrow \text{(flight ?)} \]

- Example 2: type information
  
  \[ \text{NUMBER} \rightarrow \_ \_ \_ \text{ (a numeric token)} \]
Training and Inference

- \( x \): input, \( a \): sketch, \( y \): meaning representation
- Training: maximize the log likelihood

\[
\max \sum_{(x,a,y) \in \mathcal{D}} \log p(y|x,a) + \log p(a|x)
\]

- Inference: greedy search

\[
\hat{a} = \arg \max_{a'} p(a'|x)
\]

\[
\hat{y} = \arg \max_{y'} p(y'|x, \hat{a})
\]
### Semantic Parsing Tasks

- **Natural language to logical form (Geo/ATIS)**

  what is the population of the state with the largest area?
  \[ \text{argmax} \ 0 \ (\text{and} \ (\text{mountain:}t \ 0) \ (\text{loc:}t \ 0 \ \text{alaska:s})) \ (\text{elevation:}i \ 0)) \]

- **Natural language to source code (Django)**

  if length of bits is lesser than integer 3 or second element of bits is not equal to string ’as’,
  \[
  \text{if} \ \text{len(bits)} < 3 \ \text{or} \ \text{bits}[1] \neq \text{’as’}:
  \]

- **Natural language to SQL (WikiSQL)**

<table>
<thead>
<tr>
<th>Pianist</th>
<th>Conductor</th>
<th>Record Company</th>
<th>Year of Recording</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>What record company did conductor Mikhail Snitko record for after 1996?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ \text{SELECT} \ \text{Record Company} \ \text{WHERE} \ (\text{Year of Recording} &gt; 1996) \ \text{AND} \ (\text{Conductor} = \text{Mikhail Snitko}) ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Zettlemoyer and Collins, 2005; Kwiatkowski et al., 2011; Oda et al., 2015; Zhong et al., 2017)
Natural Language to Logical Form

- "#" Variable information (e.g., lambda, count, and argmax)
- "@" Arguments of predicate or operator
- "?" Partial argument information

```
(lambda#2 (and flight@1 from@2 (< departure time@1 ? )))
```

```
(lambda $0 e (and (flight $0) (from $0 dallas:ci) (< (departure_time $0) 1000:ti)))
```
Substitute tokens with their token types

Except

- Delimiters (e.g., “[” and “:]”)
- Operators (e.g., “+” and “*”)
- Built-in keywords (e.g., “True”, and “while”)

if NAME [ : NUMBER ] . NAME ( ) == STRING :

if s [ : 4 ] . lower ( ) == 'http':

https://docs.python.org/3/library/tokenize.html
WikiSQL (Zhong et al., 2017)

```
SELECT agg_operator agg_column
WHERE (cond_column cond_operator cond_value)
AND ...
```

```
SELECT Record Company
WHERE (Year of Recording > 1996) AND (Conductor = Mikhail Snitko)
```

WHERE > AND =
Decoding is table-aware

**How many presidents are graduated from A?**

<table>
<thead>
<tr>
<th>President</th>
<th>College</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```sql
SELECT COUNT(President) WHERE (College = A)
```

<table>
<thead>
<tr>
<th>College</th>
<th>Number of Presidents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```sql
SELECT Number of Presidents WHERE (College = A)
```
Table-aware input encoder

Natural Language to SQL

Input Question

Column 1

Column 2

LSTM units
Vectors
--- Attention

Question-to-Table Attention

\[ \tilde{e}_1 \rightarrow \tilde{e}_2 \rightarrow \tilde{e}_3 \rightarrow \tilde{e}_4 \]

\[ c_e^1 \rightarrow c_e^2 \rightarrow c_e^3 \rightarrow c_e^4 \]

\[ x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \]
**Natural Language to SQL**

**SELECT clause**

```
SELECT agg_operator agg_column
WHERE (cond_column cond_operator cond_value)
AND ...
```

Diagram:
- **Softmax Classifier**
  - Corresponds to the `agg_column` selection
  - Represents the possible aggregation operators (e.g., COUNT, MIN, MAX, SUM, AVG)

- **Column Pointer**
  - Points to the specific columns (Column 1 and Column 2)
  - Details include:
    - `college`
    - `number of presidents`

**Question Vector**
- Reflects the context or question being asked

**Diagram Connections**
- Arrows indicate the flow of information from the question vector through the classifier to the column pointer, illustrating the selection process.
WHERE Clause

```
SELECT agg_operator agg_column
WHERE (cond_column cond_operator cond_value)
AND ...
```

<table>
<thead>
<tr>
<th>Pianist</th>
<th>Conductor</th>
<th>Record Company</th>
<th>Year of Recording</th>
<th>Format</th>
</tr>
</thead>
</table>

What record company did conductor Mikhail Snitko record for after 1996?
WHERE Clause

Sketch-Guided
WHERE Decoding
Sketch Encoding
Sketch Classification

SELECT agg_operator agg_column
WHERE (cond_column cond_operator cond_value) AND ...

What record company did conductor Mikhail Snitko record for after 1996?
Natural Language to SQL

**WHERE Clause**

```
SELECT agg_operator agg_column
WHERE (cond_column cond_operator cond_value) AND ...
```

What record company did conductor Mikhail Snitko record for after 1996?

<table>
<thead>
<tr>
<th>Pianist</th>
<th>Conductor</th>
<th>Record Company</th>
<th>Year of Recording</th>
<th>Format</th>
</tr>
</thead>
</table>

Point to a table column

**Sketch-Guided**

**WHERE Decoding**

**Sketch Encoding**

**Sketch Classification**
Experimental Results

NL->Code (Django)

Accuracy

62.3
69.5
74.1

(Ling et al., 2016) (Yin and Neubig, 2017) OneStage (w/o sketch) Coarse2Fine
Experimental Results

Baseline: (Dong and Lapata, 2016; Rabinovich et al., 2017)
Experimental Results

NL->SQL (WikiSQL)

- Aug Pointer Network (Zhong et al., 2017): 53.3
- (Zhong et al., 2017): 59.4
- (Xu et al., 2017): 68
- OneStage (w/o sketch): 75.9
- Coarse2Fine: 78.5
Sketch Accuracy

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OneStage</th>
<th>Coarse2Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geo</td>
<td>85.4</td>
<td>89.3</td>
</tr>
<tr>
<td>ATIS</td>
<td>85.9</td>
<td>88</td>
</tr>
<tr>
<td>Django</td>
<td>73.2</td>
<td>77.4</td>
</tr>
<tr>
<td>WikiSQL</td>
<td>95.4</td>
<td>95.9</td>
</tr>
</tbody>
</table>
Oracle Meaning Sketch

Accuracy

Geo

ATIS

Django

WikiSQL

Coarse2Fine

+ Oracle Sketch
Future Work

- Alternative ways of defining meaning sketches
  - Different levels of granularity
- Weakly supervised setting
  - Meaning sketch reduces search space
- Partial annotation
  - Only annotate meaning sketches for some examples
Thanks!

Q&A

Code Available:
http://homepages.inf.ed.ac.uk/s1478528