Language to Logical Form with Neural Attention

Li Dong and Mirella Lapata
(ACL 2016)

*Presenter: Tejas Khot*
Semantic Parsing

Natural Language (NL) → Parser → Machine Executable Language
Semantic Parsing - Querying a database

**NL**

What are the capitals of states bordering Texas?

**DB**

$$\lambda x. \text{capital}(y, x) \land \text{state}(y) \land \text{next_to}(y, \text{Texas})$$
Semantic Parsing - Instructing a robot

Natural Language (NL):

at the chair, move forward three steps past the sofa

Logic Formulation (LF):

\( \lambda a. \text{pre}(a, x. \text{chair}(x)) \land \text{move}(a) \land \text{len}(a, 3) \land \text{dir}(a, \text{forward}) \land \text{past}(a, y. \text{sofa}(y)) \)
Semantic Parsing - Question Answering

Who are the male actors in Titanic?

\[ \lambda x. \exists y. \text{gender}(\text{MALE}, x) \land \text{cast}(\text{TITANIC}, x, y) \]
Supervised Approaches

Induce parsers from sentences paired with logical forms

**Question**
Who are the male actors in Titanic?

**Logical Form**
\[ \lambda x. \exists y. \text{gender(MALE, } x) \land \text{cast(TITANIC, } x, y) \]

- ** Parsing** (Ge and Mooney, 2005; Lu et al., 2008; Zhao and Huang, 2015)
- **Inductive logic programming** (Zelle and Mooney, 1996; Tang and Mooney, 2000; Thomson and Mooney, 2003)
- **Machine translation** (Wong and Mooney, 2006; Wong and Mooney, 2007; Andreas et al., 2013)
- **CCG grammar induction** (Zettlemoyer and Collins, 2005; Zettlemoyer and Collins, 2007; Kwiatkowski et al., 2010; Kwiatkowski et al., 2011)
Indirect Supervision

Induce parsers from questions paired with side information

**Question**
Who are the male actors in Titanic?

**Answer**
{DiCaprio, BillyZane ...}

- **Answers to questions** (Clarke et al., 2010; Liang et al., 2013)
- **System demonstrations** (Chen and Mooney, 2011; Goldwasser and Roth, 2011; Artzi and Zettlemoyer, 2013)
- **Distant supervision** (Cai and Yates, 2013; Reddy et al., 2014)
Indirect Supervision

Induce parsers from questions paired with side information

**Question**
Who are the male actors in Titanic?

**Logical Form**
\[ \lambda x. \exists y. \text{gender(MALE,} x) \land \text{cast(TITANIC,} x, y) \]

**Answer**
\{D\text{ICAPRIO, BILLYZANE ...}\}

- **Answers to questions** (Clarke et al., 2010; Liang et al., 2013)
- **System demonstrations** (Chen and Mooney, 2011; Goldwasser and Roth, 2011; Artzi and Zettlemoyer, 2013)
- **Distant supervision** (Cai and Yates, 2013; Reddy et al., 2014)
In general

Developing semantic parsers requires linguistic expertise!

- high-quality lexicons based on underlying grammar formalism
- manually-built templates based on underlying grammar formalism
- grammar-based features pertaining to Logical Form and Natural Language
- Domain- and representation-specific!
Goal: All Purpose Semantic Parsing

**Question**
Who are the male actors in Titanic?

**Logical Form**
\[ \lambda x. \exists y. \text{gender(MALE, } x \text{)} \land \text{cast(TITANIC, } x, y \text{)} \]
Goal: All Purpose Semantic Parsing

Question
Who are the male actors in Titanic?

Logical Form
\[ \lambda x. \exists y. \text{gender}(\text{MALE}, x) \land \text{cast}(\text{TITANIC}, x, y) \]

- Learn from NL descriptions paired with meaning representations
- Use minimal domain (and grammar) knowledge
- Model is general and can be used across meaning representations
Problem formulation

Model maps **natural language input** $q = x_1 \cdots x_{|q|}$ to a **logical form representation** of its meaning $a = y_1 \cdots y_{|a|}$.

$$
    p(a|q) = \prod_{t=1}^{|a|} p(y_t|y_{<t}, q)
$$

where $y_{<t} = y_1 \cdots y_{t-1}$

- **Encoder** encodes natural language input $q$ into a vector representation
- **Decoder** generates $y_1, \cdots, y_{|a|}$ conditioned on the encoding vector.

Encoder Decoder Framework

what microsoft jobs do not require a bscs?

Input Utterance

Sequence Encoder

Sequence/Tree Decoder

Logical Form

answer(J,(company(J,'microsoft'),job(J),not((req_deg(J,'bscs')))))

(Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Karpathy and Fei-Fei, 2015; Vinyals et al., 2015;)
Sequence-to-Sequence (Seq2Seq) Model

$h_t^l = \text{LSTM} \left( h_{t-1}^l, h_{t-1}^{l-1} \right)$

$\begin{align*}
\mathbf{h}_t^0 &= W_q \mathbf{e}(x_t) \\
\mathbf{h}_t^0 &= W_{ae}(y_{t-1}) \\
p(y_t | y_{<t}, q) &= \text{softmax} \left( W_o h_t^l \right)^T \mathbf{e}(y_t)
\end{align*}$
Drawbacks of Seq2Seq Model

- Ignore the hierarchical structure of logical forms
- More long distance dependency during decoding
Sequence-to-Tree (Seq2Tree) Model

Define a “nonterminal” \(<n>\) token to indicate subtrees in decoder

```
lambda $0 e <n>

and <n> <n>

> <n> 1600:ti from $0 dallas:ci

departure_time $0
```
(lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci)))
Seq2Tree Model

(lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci)))
Seq2Tree Model

(lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci)))
(lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci)))
(lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci)))
Seq2Tree Decoder

Diagram showing the flow of lambda $0$, e, and other elements through LSTM units for non-terminal decoding, start decoding, parent feeding, encoder unit, and decoder unit.
A `SEQ2TREE` decoding example for the logical form “A B (C)”

- Hidden vector of the parent nonterminal is concatenated with the inputs and fed to the LSTM.

\[
p(a|q) = p(y_1y_2y_3y_4|q) \cdot p(y_5y_6|y_{\leq 3}, q)
\]
Attention Mechanism - Soft Alignment

(Bahdanau et al., 2015; Luong et al., 2015b; Xu et al., 2015)
Training and Inference

Our goal is to maximize the likelihood of the generated logical forms given natural language utterances as input.

\[ \text{minimize} - \sum_{(q,a) \in D} \log p(a|q) \]

where \( D \) is the set of all natural language-logical form training pairs

At test time, we predict the logical form for an input utterance \( q \) by:

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

- Iterating over all possible \( a' \)’s to obtain the optimal prediction is impractical
- Probability \( p(a|q) \) decomposed so that we can use greedy/beam search.
Argument Identification

- Many LFs contain named entities and numbers aka rare words.
- Does not make sense to replace them with special unknown word symbol.
- Identify entities and numbers in input questions and replace them with their type names and unique IDs.

.jobs with a salary of 40000

job(ANS), salary_greater_than(ANS, 40000, year)

- Pre-processed examples are used as training data.
- After decoding, a post-processing step recovers all \textit{type}_i markers to corresponding logical constants.
Argument Identification

- Many LF's contain named entities and numbers aka rare words.
- Does not make sense to replace them with special unknown word symbol.
- Identify entities and numbers in input questions and replace them with their type names and unique IDs.

\[
\text{jobs with a salary of } \text{num}_0
\]

\[
\text{job}(\text{ANS}), \text{salary}_\text{greater}_\text{than}(\text{ANS}, \text{num}_0, \text{year})
\]

- Pre-processed examples are used as training data.
- After decoding, a post-processing step recovers all type markers to corresponding logical constants.
## Semantic Parsing Datasets

<table>
<thead>
<tr>
<th>Length</th>
<th>JOBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.80</td>
<td><em>what microsoft jobs do not require a bscs?</em></td>
</tr>
<tr>
<td>22.90</td>
<td><code>answer(company(J,'microsoft'),job(J),not((req_deg(J,'bscs'))))</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>GEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.60</td>
<td><em>what is the population of the state with the largest area?</em></td>
</tr>
<tr>
<td>19.10</td>
<td><code>(population:i (argmax $0 (state:t $0) (area:i $0)))</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>ATIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.10</td>
<td><em>dallas to san francisco leaving after 4 in the afternoon please</em></td>
</tr>
<tr>
<td>28.10</td>
<td><code>(lambda $0 e (and (&gt; (departure_time $0) 1600:ti) (from $0 dallas:ci) (to $0 san_francisco:ci)))</code></td>
</tr>
</tbody>
</table>

- **JOBS**: queries to job listings; 500 training, 140 test instances.
- **GEO**: queries US geography database; 680 training, 200 test instances.
- **ATIS**: queries to a flight booking system; 4,480 training, 480 dev, 450 test.
IFTTT

IF-This-Then-That

- turn on my lights when I arrive home
- text me if the door opens
- remind me to drink water if I’ve been at a bar for more than two hours

Archive your missed calls from Android to Google Drive
### Semantic Parsing Datasets

<table>
<thead>
<tr>
<th>Length</th>
<th>IFTTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.95</td>
<td><em>turn on heater when temperature drops below 58 degree</em></td>
</tr>
<tr>
<td>21.80</td>
<td>TRIGGER: Weather - Current_temperature_drops_below - ((Temperature(58)) (Degrees_in(f)))</td>
</tr>
<tr>
<td></td>
<td>ACTION: WeMo_Insight_Switch - Turn_on - ((Which_switch?('')))</td>
</tr>
</tbody>
</table>

- if-this-then-that recipes from the IFTTT website (Quirk et al., 2015)
- Recipes are simple programs with exactly one trigger and one action
- 77,495 training, 5,171 development, and 4,294 test examples
- IFTTT programs are represented as abstract syntax trees; NL descriptions provided by users.
Experimental Results

![Bar Chart showing accuracy for different models in the Jobs task]

- Tang and Mooney, 2001: 79.4
- Popescu et al., 2003: 88
- Zettlemoyer and Collins, 2005: 79.3
- Liang et al., 2013: 90.7
- Zhao and Huang, 2015: 85
- vanilla Seq2Seq: 70.7
- w/argument: 77.9
- w/attention: 87.1
- Seq2Tree: 90
Experimental Results

(Quirk et al., 2015): IFTTT baselines
what is the earliest flight from ci0 to ci1

what is the highest elevation in the co0
Error Analysis

➢ Under-Mapping
  ○ Keeping track of attention history
  ○ Esp. for longer sequences

➢ Argument Identification
  ○ Disambiguate arguments based on context
  ○ Eg: 6’0 clock

➢ Rare Words
  ○ Learn word embeddings on unannotated text data
  ○ Esp. a problem for smaller datasets
What have we learned?

- Encoder-decoder neural network model for mapping natural language descriptions to meaning representations
- TREE2SEQ and attention improves performance
- Model general and could transfer to other tasks/architectures
- Future work: learn a model from question-answer pairs without access to target logical forms.
Thanks!

Q&A

(Q-😊->🔴 - ⬆️→A)