New Resources and Ideas for Semantic Parsing

Kyle Richardson

Institute for Natural Language Processing (IMS)

School of Computer Science, Electrical Engineering and Information Technology
University of Stuttgart

June 7, 2018

Collaborators: Jonas Kuhn (advisor, Stuttgart) and Jonathan Berant (work on "polyglot semantic parsing", Tel Aviv), last updated 15.10.2018
Main Topic: Semantic Parsing

- **Task**: mapping text to formal (machine-readable) structured meaning representations:

  **Text**: *Find me flights from Boston to New York.*

  \[ \rightarrow \]

  **Logical Form (LF)**:  \[ \lambda x. \text{flight}(x) \land \text{depart}(x, \text{bos}) \land \text{arrive}(x, \text{ny}) \]
Main Topic: Semantic Parsing

▶ **Task:** mapping text to formal (machine-readable) structured meaning representations:

**Text:**  
Find me flights from Boston to New York.

→

**Logical Form (LF):**  
\( \lambda x. \text{flight}(x) \land \text{depart}(x, \text{bos}) \land \text{arrive}(x, \text{ny}) \)

"Machines and programs which attempt to answer English question have existed for only about five years.... Only in recent years have attempts been made to translate mechanically from English into logical formalisms [or LFs]..."

Communications of the ACM
Classical Natural Language Understanding (NLU)

- Conventional **pipeline model**: focus on capturing **deep inference** and **entailment** (ex. Lunar QA system (Woods, 1973)).

```
FOR EVERY X / MAJORELT : T;
    (FOR EVERY Y /
        SAMPLE : (CONTAINS Y X);
        (PRINTOUT Y))
```

List samples that contain every major element

\[
\text{[sem]} = \{\text{S10019, S10059, ...}\}
\]
NLU model is a kind of compiler, involves a **transduction** from NL to a formal (usually logical) language.
Data-driven NLU and Semantic Parsing

Data-driven NLU: Asks an empirical question: Can we learn NLU models from examples?
Data-driven NLU and Semantic Parsing

1. Semantic Parsing
   - List samples that contain every major element

2. Knowledge Representation
   - \[(\text{FOR EVERY } X / \text{ MAJORELT} : T; \text{\ FOR EVERY } Y / \text{SAMPLE} : (\text{CONTAINS Y X}); \text{(PRINTOUT Y)})\]\n   - \([sem] = \{S10019, S10059, \ldots\}\)

3. Reasoning

- **Data-driven NLU**: Asks an empirical question: Can we learn NLU models from examples?
- **Semantic Parser Induction**: Learn semantic parser (i.e., translation to LFs) automatically from example parallel data.
Data-driven Semantic Parsing in a Nutshell

Training

Parallel Training Set
\[ D = \{(x_i, z_i)\}_{i}^{\|D\|} \]

\[ \rightarrow \] Machine Learner

Testing

\[ \text{input} \rightarrow \text{Semantic Parsing} \rightarrow \text{sem} \]

X

\text{decoding} \quad (Finding the best z)

Z

\text{reasoning} \quad \text{world}
Data-driven Semantic Parsing in a Nutshell

Training

Parallel Training Set
\[ D = \left\{ (x_i, z_i) \right\}_{i=1}^{|D|} \]

Machine Learner

Testing

input

Semantic Parsing

decoding
\( (Finding the best z) \)

\( x \)

\( z \)

world

reasoning

model

sem

Evaluation: Correct Sem?
Data-driven Semantic Parsing in a Nutshell

Training

Parallel Training Set
\[ D = \{ (x_i, z_i) \}_{i=1}^{D} \]

Machine Learner

Testing

input

Semantic Parsing

\( x \)

decoding

\( z \)

reasoning

world

model

Evaluation: Correct Sem?
Data-driven Semantic Parsing in a Nutshell

Training

challenge 1: Getting data?

Parallel Training Set

$D = \{(x_i, z_i)\}_{i=1}^{|D|}$

challenge 2: Missing data?

Machine Learner

Testing

input

Semantic Parsing

$\text{decoding}$

(Finding the best $z$)

$X$

$\text{world}$

$\text{reasoning}$

sem

Evaluation: Correct Sem?
Data-driven Semantic Parsing in a Nutshell

**Challenge 1: Getting data?**

**Parallel Training Set**

\[ D = \{(x_i, z_i)\}_{i}^{D} \]

**Challenge 2: Missing data?**

**Challenge 3: Deficient LFs?**

**Training**

**Testing**

**Model**

**Input**

**Semantic Parsing**

**decoding**

\((Finding \ the \ best \ z)\)

\[ x \]

\[ z \]

**Reasoning**

**Evaluation: Correct Sem?**
Thesis Contributions and Talk Outline

challenge 1: Getting data?

Use source code as resource for building (synthetic) parallel corpora for semantic parsing; introduce 45 new multilingual datasets and models.

*Richardson* and *Kuhn* 2017b. *ACL*

*Richardson* and *Kuhn* 2017a. *EMNLP*

challenge 2: Missing data?

Train semantic parsers on multiple datasets and domains (polyglot modeling), develop a new graph-based decoding framework.

*Richardson*, *Berant* and *Kuhn* 2018. *NAACL*

challenge 3: Deficient LFs?

Train semantic parsers using entailment information; introduce new learning framework: *learning from entailment*.

*Richardson* and *Kuhn* 2016. *TACL*
<Challenge 1>

challenge 1: Getting data?

Training

Parallel Training Set

\[ D = \{(x_i, z_i)\}_{i}^{\mid D \mid} \]

Machine Learner

Testing

model

\[ \downarrow \]

\[ \ldots \]
Learning from LFs: Assumes pairs of text $x$ and full logical forms $z$, goal is to learn $\text{sem}: x \rightarrow z$, evaluate accuracy of translation.

GeoQuery (Zelle and Mooney, 1996): Benchmark dataset, available in four languages, LFs hand annotated by domain experts.
Learning from LFs: Assumes pairs of text $x$ and full logical forms $z$, goal is to learn $\text{sem}: x \rightarrow z$, evaluate accuracy of translation.

GeoQuery (Zelle and Mooney, 1996): Benchmark dataset, available in four languages, LFs hand annotated by domain experts.

Underlying Challenge: Getting pairs of text and full LFs without expensive annotation effort.
Source Code and API Documentation

```
* Returns the greater of two long values
*
* @param a an argument
* @param b another argument
* @return the larger of a and b
* @see java.lang.Long#MAX_VALUE
*/
public static Long max(long a, long b)
```

Source Code Documentation: High-level descriptions of internal software functionality paired with code.
Source Code and API Documentation

```java
public static Long max(long a, long b)

* Returns the greater of two long values
* @param a an argument
* @param b another argument
* @return the larger of a and b
* @see java.lang.Long#MAX_VALUE
*/
```

- **Source Code Documentation:** High-level descriptions of internal software functionality paired with code.

- **Idea:** Treat as a parallel corpus (Allamanis et al., 2015), or **synthetic semantic parsing** dataset.
Source Code as a Parallel Corpus

- Tight coupling between high-level text and code, easy to extract text/code pairs automatically (no annotation).

```
* Returns the greater of two long values
* @param a an argument
* @param b another argument
* @return the larger of a and b
*/
public static Long max(long a, long b)
```

```
(ns ... clojure.core)
(defn random-sample
  "Returns items from coll with random probability of prob (0.0 - 1.0)"
  ([prob] ...) ([prob coll] ...))
```

- **Function signatures:** Header-like representations, have similar predicate-argument structure to atomic predicate logic.

```
Signature ::= lang Math long return name (long a, long b) (named/tipped arguments)
```
New Resources: Stdlib and Py27 Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Pairs</th>
<th>#Symbols</th>
<th>#Words</th>
<th>Vocab.</th>
<th>Example Pairs (x, z)</th>
</tr>
</thead>
</table>
| Java     | 7,183   | 4,072    | 82,696  | 3,721  | x: Compares this Calendar to the specified Object.  
         |         |          |         |        | z: boolean util.Calendar.equals(Object obj)                                         |
| Ruby     | 6,885   | 3,803    | 67,274  | 5,131  | x: Computes the arc tangent given y and x.  
         |         |          |         |        | z: Math.atan2(y,x) → Float                                                            |
| PHP_en   | 6,611   | 8,308    | 68,921  | 4,874  | x: Delete an entry in the archive using its name.  
         |         |          |         |        | z: bool ZipArchive::deleteName(string $name)                                           |
| Python   | 3,085   | 3,991    | 27,012  | 2,768  | x: Remove the specific filter from this handler.  
         |         |          |         |        | z: logging.Filterer.removeFilter(filter)                                               |
| Elisp    | 2,089   | 1,883    | 30,248  | 2,644  | x: Returns the total height of the window.  
         |         |          |         |        | z: (window-total-height window round)                                                  |
| Geoquery | 880     | 167      | 6,663   | 279    | x: What is the tallest mountain in America?  
         |         |          |         |        | z: (highest(mountain(loc_2(countryid usa))))                                           |

- **Stdlib**: Datasets 18 standard libraries, 10 programming languages, 7 natural languages.
- **Py27**: 27 open-source Python projects from GitHub.
# New Resources: Stdlib and Py27 Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Pairs</th>
<th>#Symbols</th>
<th>#Words</th>
<th>Vocab.</th>
<th>Example Pairs (x, z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>7,183</td>
<td>4,072</td>
<td>82,696</td>
<td>3,721</td>
<td>x: Compares this Calendar to the specified Object.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z: boolean util.Calendar.equals(Object obj)</td>
</tr>
<tr>
<td>Ruby</td>
<td>6,885</td>
<td>3,803</td>
<td>67,274</td>
<td>5,131</td>
<td>x: Computes the arc tangent given y and x.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z: Math.atan2(y,x) → Float</td>
</tr>
<tr>
<td>PHP&lt;sub&gt;en&lt;/sub&gt;</td>
<td>6,611</td>
<td>8,308</td>
<td>68,921</td>
<td>4,874</td>
<td>x: Delete an entry in the archive using its name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z: bool ZipArchive::deleteName(string $name)</td>
</tr>
<tr>
<td>Python</td>
<td>3,085</td>
<td>3,991</td>
<td>27,012</td>
<td>2,768</td>
<td>x: Remove the specific filter from this handler.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z: logging.Filterer.removeFilter(filter)</td>
</tr>
<tr>
<td>Elisp</td>
<td>2,089</td>
<td>1,883</td>
<td>30,248</td>
<td>2,644</td>
<td>x: Returns the total height of the window.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z: (window-total-height window round)</td>
</tr>
<tr>
<td>Geoquery</td>
<td>880</td>
<td>167</td>
<td>6,663</td>
<td>279</td>
<td>x: What is the tallest mountain in America?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>z: (highest(mountain(loc_2(countryid usa))))</td>
</tr>
</tbody>
</table>

- **Stdlib**: Datasets 18 standard libraries, 10 programming languages, 7 natural languages.

- **Py27**: 27 open-source Python projects from GitHub.
New Task: Text to Signature Translation

<table>
<thead>
<tr>
<th>text</th>
<th>Returns the greater of two long values</th>
</tr>
</thead>
<tbody>
<tr>
<td>signature</td>
<td>lang.Math long max( long a, long b )</td>
</tr>
</tbody>
</table>

**Task:** Given text/signatures training pairs, learn a *semantic parser*: text $\rightarrow$ signature, predicting within finite signature/translation space.
Text to Signature Translation: How hard is it?

▶ **A First Model**: Use statistical (word-based) machine translation (SMT) (Deng and Chrupała, 2014) and reranking.

### SMT Model
- Gets the total cache size
- `string APCIterator::key(void)`
- `int APCIterator::getTotalHits(void)`
- `int APCIterator::getSize(void)`
- `int APCIterator::getTotalSize(void)`
- `int Memcached::append(string $key)`
- ...

### Constrained Decoder
- `string APCIterator::key(void)`
- `int APCIterator::getTotalHits(void)`
- `int APCIterator::getSize(void)`
- `int APCIterator::getTotalSize(void)`
- `int Memcached::append(string $key)`
- ...

### Discriminative Model

▶ **Decoding**: finding the best output given input, **unconstrained** versus **constrained** (assign probability to wellformed output only).
How does such a simple approach fare on benchmark tasks and our task?
Text to Signature Translation: How Hard Is It?

- How does such a simple approach fare on benchmark tasks and our task?
How does such a simple approach fare on benchmark tasks and our task?
Text to Signature Translation: How Hard Is It?

- How does such a simple approach fare on benchmark tasks and our task?

![Accuracy GeoLang](image)

- Our Constrained SMT Model (IBM Model 1)
- Discriminative Reranker
- Competitor Model/Baseline
- Unconstrained phrase SMT (Moses)
Text to Signature Translation: How Hard Is It?

- How does such a simple approach fare on benchmark tasks and our task?

**Result**: achieving high accuracy is not easy, not a trivial problem.
How does such a simple approach fare on benchmark tasks and our task?
How does such a simple approach fare on benchmark tasks and our task?
How does such a simple approach fare on benchmark tasks and our task?
Observation: Semantic Parsing is not an unconstrained translation problem, constraining the search is very important.
Introduced 45 new datasets and novel text-to-signature task.

This work is of interest to semantic parsing:

- Reveals the limitations of existing techniques in sparse settings, better benchmark (realistic vocabulary/domain size).
- Requires asking fundamental questions about how decoding and search work.
<Challenge 2>

Training

Parallel Training Set
\[ D = \left\{ (x_i, z_i) \right\}_{i=1}^{|D|} \]

Missing data?

Machine Learner

Testing

Semantic Parsing

decoding
Traditional approaches to semantic parsing train individual models for each available parallel dataset.

**Underlying Challenge:** Datasets tend to be small, hard and unlikely to get certain types of parallel data, e.g., \((es, Java)\).
Polyglot Models: Training on Multiple Datasets

Idea: concatenate all datasets, build a single-model with shared parameters, capture redundancy (Herzig and Berant, 2017).

Polyglot Translator: translates from any input language to any output (programming) language.
Polyglot Models: Training on Multiple Datasets

- Idea: concatenate all datasets, build a single-model with shared parameters, capture redundancy (Herzig and Berant, 2017).

- Polyglot Translator: translates from any input language to any output (programming) language.

1. **Multiple Datasets**: Does this help learn better semantic parsers?

2. **Zero-Short Translation** (Johnson et al., 2016): Can we translate between unobserved language pairs?
Graph-Based Constrained Decoding

▶ Idea: Represent full translation search space as directed graph, add artificial language tokens.
Graph-Based Constrained Decoding

▶ Idea: Represent full translation search space as directed graph, add artificial language tokens.

▶ Decoding/Search (test time): Find a path given an input $x$:

\[ x \text{ : The ceiling of a number} \]

Formulate as weighted shortest-path search (use translation models as dynamic weight functions), defines a general decoding framework.
Shortest Path Decoding in a Nutshell

- **Standard SSSP**: Traverse labeled edges $E$ (label $z$) in order (e.g., sorted or best-first order), and solve for each node $v$ the following recurrence:

$$d[v] = \min_{(u,v,z) \in E} \left\{ d[u] + \text{node score} + \text{incoming node score} + \text{edge score} \right\}$$
Shortest Path Decoding in a Nutshell

- **Standard SSSP**: Traverse labeled edges $E$ (label $z$) in order (e.g., sorted or best-first order), and solve for each node $v$ the following recurrence:

$$d[v] = \min_{(u,v,z) \in E} \left\{ d[u] + \text{TRANS}(x, z) \right\}$$

- Use trained **translation model** to dynamically weight edges, general framework for directly comparing models (Richardson et al., 2018).
Shortest Path Decoding in a Nutshell

- **Standard SSSP**: Traverse labeled edges $E$ (label $z$) in order (e.g., sorted or best-first order), and solve for each node $v$ the following recurrence:

  $$d[v] = \min_{(u,v,z) \in E} \left\{ d[u] + \text{TRANS}(x, z) \right\}$$

  node score  incoming node score  translation

- Use trained **translation model** to dynamically weight edges, general framework for directly comparing models (Richardson et al., 2018).

- **constrained decoding**: ensure that output is well-formed, related efforts: Krishnamurthy et al. (2017); Yin and Neubig (2017).
DAG Decoding for Neural Semantic Parsing (Example)

- **Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016), variants of (Bahdanau et al., 2014), direct decoder model (unconstrained):

\[
p(z \mid x) = \text{CONDITIONALRNNLM}(z) \]

\[
= \prod_{i} p_{\Theta}(z_i \mid z_{<i}, x)
\]

DAGs \( G = (V, E) \), numerically sorted nodes (acyclic), trained decoder.

0: \( d[b] \leftarrow 0 \).

1: for vertex \( u \in V \) in topologically sorted order do

2: \( d(v) = \min(u, v, z \in E \{d(u) + w(u, v, z)\}) \)

3: \( s[v] \leftarrow \text{RNN state for min edge and } z 

4: \text{return } \min_{v \in V} \{d(v)\} \]
DAG Decoding for Neural Semantic Parsing (Example)

- **Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016), variants of (Bahdanau et al., 2014), direct decoder model (unconstrained):

\[
p(z \mid x) = \text{CONDITIONALRNNLM}(z) = \prod_i p_{\Theta}(z_i \mid z_{<i}, x)
\]

- **DAGs** $\mathcal{G} = (V, E)$, **numerically sorted nodes** (acyclic), trained decoder.
**DAG Decoding for Neural Semantic Parsing (Example)**

- **Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016), variants of (Bahdanau et al., 2014), direct decoder model (unconstrained):

  \[
  p(z \mid x) = \text{CONDITIONAL RNNLM}(z) \\
  = \prod_{i} p_{\Theta}(z_i \mid z_{<i}, x)
  \]

- **DAGs** $\mathcal{G} = (V, E)$, *numerically sorted nodes* (acyclic), trained decoder.

0: $d[0] \leftarrow 0.0$
1: **for** node $v \in V$ in topologically sorted order
DAG Decoding for Neural Semantic Parsing (Example)

- **Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016), variants of (Bahdanau et al., 2014), direct decoder model (unconstrained):

\[
p(z \mid x) = \text{CONDITIONALRNNLM}(z) = \prod_{i} p_{\Theta}(z_i \mid z_{<i}, x)
\]

- **DAGs**: \(G = (V, E)\), numerically sorted nodes (acyclic), trained decoder.

0: \(d[b] \leftarrow 0.0\)
1: for node \(v \in V\) in topologically sorted order
2: \(d(v) = \min_{(u,v,z_j) \in E} \{d(u) + -\log p_{\Theta}(z_j \mid z_{<j}, x)\}\)
DAG Decoding for Neural Semantic Parsing (Example)

- **Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016), variants of (Bahdanau et al., 2014), direct decoder model (unconstrained):

\[
p(z \mid x) = \text{CONDITIONALRNNLM}(z) = \prod_{i} p_{\Theta}(z_i \mid z_{<i}, x)
\]

- **DAGs** \( G = (V, E) \), **numerically sorted nodes** (acyclic), trained decoder.

0: \( d[b] \leftarrow 0.0 \)
1: for node \( v \in V \) in topologically sorted order
2: do \( d(v) = \min_{(u, v, z_j) \in E} \left\{ d(u) + -\log p_{\Theta}(z_j \mid z_{<j}, x) \right\} \)
3: \( s[v] \leftarrow \text{RNN state for min edge and } z_j \)
DAG Decoding for Neural Semantic Parsing (Example)

▶ **Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016), variants of (Bahdanau et al., 2014), direct decoder model (unconstrained):

\[
p(z \mid x) = \text{CONDITIONALRNNLM}(z) = \prod_{i} p_{\Theta}(z_i \mid z_{<i}, x)
\]

▶ **DAGs** \(G = (V, E)\), **numerically sorted nodes** (acyclic), trained decoder.

0: \(d[b] \leftarrow 0.0\)
1: **for** node \(v \in V\) in topologically sorted order
2: **do** \(d(v) = \min_{(u,v,z_j) \in E} \left\{ d(u) + -\log p_{\Theta}(z_j \mid z_{<j}, x) \right\}\)
3: \(s[v] \leftarrow \text{RNN state for min edge and } z_j\)
4: **return** \(\min_{v \in V} \{ d(v) \}\)
Shortest Path Decoding: Comparing Models

▶ **Shortest Path Decoding Framework**: Directly compare the performance of different semantic parsing models under a single search procedure.

▶ **Neural Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016; Jia and Liang, 2016).
Shortest Path Decoding: Comparing Models

- **Shortest Path Decoding Framework**: Directly compare the performance of different semantic parsing models under a single search procedure.

- **Neural Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016; Jia and Liang, 2016).
Shortest Path Decoding: Comparing Models

▶ **Shortest Path Decoding Framework**: Directly compare the performance of different semantic parsing models under a single search procedure.

▶ **Neural Seq2Seq**: popular in semantic parsing (Dong and Lapata, 2016; Jia and Liang, 2016).
Training on Multiple Datasets: Does this help?

- **Polyglot Models**: Directly compare if training on multiple datasets improves translation.

- **Benchmark Datasets**: Training *polyglot models* on multiple datasets can increase performance, makes learning more robust.
Training on Multiple Datasets: Does this help?

▶ **Polyglot Models:** Directly compare if training on multiple datasets improves translation.

▶ **Code Datasets:** Training *polyglot models* on multiple datasets can increase performance, depending on the model.
Training on Multiple Datasets: Does this help?

- **Polyglot Models**: Directly compare if training on multiple datasets improves translation.

- **Code Datasets**: Training *polyglot models* on multiple datasets can increase performance, depending on the model.
Training on Multiple Datasets: Does this help?

- **Polyglot Models**: Directly compare if training on multiple datasets improves translation.

- **Code Datasets**: Training *polyglot models* on multiple datasets can increase performance, depending on the model.
Advantages: Any/Mixed Language Decoding

▶ Any Language Decoding: translating between multiple APIs, letting the decoder decide output language, zero-shot translation.

<table>
<thead>
<tr>
<th>Source API (stdlib): (es, PHP)</th>
<th>Input: Devuelve el mensaje asociado al objeto lanzado.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language: PHP</td>
<td>Translation: public string Throwable::getMessage ( void )</td>
</tr>
<tr>
<td>Language: Java</td>
<td>Translation: public String lang.getMessage( void )</td>
</tr>
<tr>
<td>Language: Clojure</td>
<td>Translation: (tools.logging.fatal throwable message &amp; more)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source API (stdlib): (ru, PHP)</th>
<th>Input: конвертирует строку из формата UTF-32 в формат UTF-16.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language: PHP</td>
<td>Translation: string PDF_utf32_to_utf16 ( ... )</td>
</tr>
<tr>
<td>Language: Ruby</td>
<td>Translation: String#toutf16 =&gt; string</td>
</tr>
<tr>
<td>Language: Haskell</td>
<td>Translation: Encoding.encodeUtf16LE :: Text -&gt; ByteString</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source API (py): (en, stats)</th>
<th>Input: Compute the Moore-Penrose pseudo-inverse of a matrix.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: sympy</td>
<td>Translation: matrices.matrix.base.pinv_solve( B, ... )</td>
</tr>
<tr>
<td>Project: sklearn</td>
<td>Translation: utils.pinvh( a, cond=None,rcond=None,... )</td>
</tr>
<tr>
<td>Project: stats</td>
<td>Translation: tools.pinv2( a,cond=None,rcond=None )</td>
</tr>
</tbody>
</table>
Advantages: Any/Mixed Language Decoding

- **Mixed Language Decoding**: translating from input with NPs from multiple languages, introduced a new mixed GeoQuery test set.

<table>
<thead>
<tr>
<th>Mixed Lang.</th>
<th>Input: Wie hoch liegt der höchstgelegene punkt in Αλαμπάμα?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF: answer(elevation_1(highest(place(loc_2(stateid('alabama')))))))</td>
<td></td>
</tr>
</tbody>
</table>
- Polyglot modeling: training on multiple datasets, helps to make models more robust and learn across domains.

- Developed a graph-based constrained decoding framework:
  - Supports polyglot and mixed language decoding.
  - Allows for directly comparing models using a single search protocol.
<Challenge 3>

Training

Parallel Training Set

\[ D = \left\{ (x_i, z_i) \right\}_{i=1}^{|D|} \]

Testing

input \rightarrow \text{Semantic Parsing} \rightarrow \text{sem} \rightarrow \text{world}

Deficient LFs?

Machine Learner

Complex Evaluation?
Entailment: One of the basic aims of semantics (Montague, 1970).  

- t. All samples that contain a major element
  →
- h. Some sample that contains a major element

database

\[
\text{[sem]} = \{S10019, S10059, \ldots \} \supseteq \{S10019\}
\]

1Recognizing Textual Entailment (RTE): would a person reading t usually infer h? (Dagan et al., 2005), answers: { Entail (yes), Contradict (no), Unknown (possible) }
Question: What happens if we unit test our semantic parsers using an RTE test?

Sportscaster: \( \approx 1,800 \) soccer descriptions paired with logical forms (LFs) (Chen and Mooney, 2008).

<table>
<thead>
<tr>
<th>sentence</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>t    Pink 3 passes to Pink 7</td>
<td>pass(pink3,pink7)</td>
</tr>
<tr>
<td>h    Pink 3 quickly kicks to Pink 7</td>
<td>pass(pink3,pink7)</td>
</tr>
</tbody>
</table>

| inference (human) t \( \rightarrow \) h | Unknown (RTE) |
| inference (LF match) t \( \rightarrow \) h | Entail (RTE) |
Question: What happens if we unit test our semantic parsers using an RTE test?

Sportscaster: ≈1,800 soccer descriptions paired with logical forms (LFs) (Chen and Mooney, 2008).

<table>
<thead>
<tr>
<th>sentence</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>The pink goalie passes to pink 7</td>
<td>pass(pink1,pink7)</td>
</tr>
<tr>
<td>Pink 1 kicks the ball</td>
<td>kick(pink1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>inference (human) t → h</th>
<th>Entail (RTE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>inference (LF match) t → h</td>
<td>Contradict (RTE)</td>
</tr>
</tbody>
</table>
 Semantic Parsing and Entailment

- **Question**: What happens if we *unit test* our semantic parsers using an RTE test?

- **Sportscaster**: $\approx 1,800$ soccer descriptions paired with logical forms (LFs) (Chen and Mooney, 2008).

<table>
<thead>
<tr>
<th>Inference Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>33.1%</td>
</tr>
<tr>
<td>RTE Classifier</td>
<td>52.4%</td>
</tr>
<tr>
<td>LF Matching</td>
<td>59.6%</td>
</tr>
</tbody>
</table>

- **Challenge 3**: Model cannot solve RTE, can we teach our model to reason logically about entailment?
Learning from Entailment: Illustration

- Add pairs of sentences with entailment judgements to training, jointly train model to reason logically about entailment and soccer.

```
input: (t,h)
t  pink3  λ  passes to  pink1
 h  pink3  quickly  kicks  λ

Correct Logical Reasoning
```

```
z  Uncertain
```

world
Add pairs of sentences with entailment judgements to training, jointly train model to reason logically about entailment and soccer.
Learning from Entailment: Illustration

Add pairs of sentences with entailment judgements to training, jointly train model to reason logically about entailment and soccer.

Correct Logical Reasoning

- passes to pink1 ⇒ kicks
- passes to pink1 ? quickly kicks

Uncertain
Learning from Entailment: Illustration

- Add pairs of sentences with entailment judgements to training, jointly train model to reason logically about entailment and soccer.

```
input: (t,h)
```

```
t = pink3, \lambda passes to pink1
h = pink3, quickly kicks, \lambda
```

---

**Correct Logical Reasoning**

- \( \checkmark \) passes to pink 1 \( \Rightarrow \) kicks
- \( \times \) passes to pink1 \( ? \) quickly kicks
- \( \times \) pink 3 passes to pink1 \( ? \) pink3 quickly kicks

\( ? = \text{Uncertain} \)

---

```
EI = Uncertain
```

---

```
world
```
Learning from Entailment: Illustration

- Add pairs of sentences with entailment judgements to training, jointly train model to reason logically about entailment and soccer.

\[
\text{input: } (t,h) \quad \text{t} \quad \text{pink3} \quad \lambda \quad \text{passes to} \quad \text{pink1} \\
\text{h} \quad \text{pink3} \quad \text{quickly} \quad \text{kicks} \quad \lambda
\]

**Incorrect Reasoning**

- ✔ passes to pink1 ⇒ kicks
- ✔ passes to pink1 ⇒ quickly kicks
- ✔ pink3 passes to pink1 ⇒ pink3 quickly kicks

⇒ = Entail

**z Uncertain**
Grammar Approach: Sentences to Logical Form

- Translation rules as probabilistic grammar rewrites, constructed from target representations using templates (Börschinger et al. (2011))

\[(x: \text{purple 10 quickly kicks}, z: \{\text{kick(purple10), block(purple7),...}\})\]

\[\downarrow \text{(rule extraction)}\]
Grammar Approach: Sentences to Logical Form

- Translation rules as probabilistic grammar rewrites, constructed from target representations using templates (Börschinger et al. (2011))

\[(x: \text{purple 10 quickly kicks}, z:\{\text{kick(purple10)}, \text{block(purple7)},...\})\]
Translation rules as probabilistic grammar rewrites, constructed from target representations using templates (Börschinger et al. (2011))

\[(x: \text{purple 10 quickly kicks}, z: \{\text{kick(purple10), block(purple7)},\ldots\}\)]

\[\Rightarrow \text{(rule extraction)}\]

- kick(purple10)
- kick(purple10)
- block(purple7)
- block(purple9)
Modeling Entailments as Structured Proofs

- Define a novel probabilistic language and logic based on the natural logic calculus (MacCartney and Manning, 2009).
- Rules decompose to probabilistic rewrites, allows for joint training with ordinary semantic parser using single generative model.

\[
(\text{\emph{t}: pink 1 kicks, \emph{h}: pink 1 quickly passes to pink 2}, z: \text{Uncertain})
\]

\[
\downarrow \text{(inference rules)}
\]
We refer to weakly-supervised semantic parsing (Liang et al., 2013; Berant et al., 2013), treat as partially-observed random process (Guu et al., 2017).

\[ x = (t, h), z \in \{\text{Entail, Contradict, Unknown}\} \]
Joint Entailment Modeling and Reasoning

Weakly-supervised semantic parsing (Liang et al., 2013; Berant et al., 2013), treat as partially-observed random process (Guu et al., 2017).

\[ x = (t, h), \ z \in \{\text{Entail}, \text{Contradict}, \text{Unknown}\} \]

\[ p(z \mid x) = \sum_{y \in \mathcal{Y}_x} p(z \mid y) \times \left( p(y \mid x) \right) \]

- \( p(z \mid y) \): 1 if proof derives correct entailment, 0 otherwise
- \( p(y \mid x) \): Model proof structures and rules as PCFG, use variant of natural logic calculus (MacCartney and Manning, 2009).

Results in an interesting probabilistic logic, efficient proof search via reduction to (P)CFG search.
Joint Entailment Modeling and Reasoning

- Weakly-supervised semantic parsing (Liang et al., 2013; Berant et al., 2013), treat as partially-observed random process (Guu et al., 2017).

\[
x = (t, h), \ z \in \{\text{Entail, Contradict, Unknown}\}
\]

\[
p(z \mid x) = \sum_{y \in \mathcal{Y}_x} p(z \mid y) \times p_\theta(y \mid x)
\]

- \(p(z \mid y)\): 1 if proof derives correct entailment, 0 otherwise
Joint Entailment Modeling and Reasoning

- Weakly-supervised semantic parsing (Liang et al., 2013; Berant et al., 2013), treat as partially-observed random process (Guu et al., 2017).

\[ x = (t, h), \ z \in \{\text{Entail, Contradict, Unknown}\} \]

\[ p(z \mid x) = \sum_{y \in \mathcal{Y}_x} p(z \mid y) \times p_\theta(y \mid x) \]

- \( p(z \mid y) \): 1 if proof derives correct entailment, 0 otherwise

- \( p_\theta(y \mid x) \): Model proof structures and rules as PCFG, use variant of natural logic calculus (MacCartney and Manning, 2009).
Joint Entailment Modeling and Reasoning

- Weakly-supervised semantic parsing (Liang et al., 2013; Berant et al., 2013), treat as partially-observed random process (Guu et al., 2017).

\[ x = (t, h), \ z \in \{\text{Entail, Contradict, Unknown}\} \]

\[ p(z | x) = \sum_{y \in Y_x} p(z | y) \times p(y | x) \]

- \( p(z | y) \): 1 if proof derives correct entailment, 0 otherwise

- \( p_\theta(y | x) \): Model proof structures and rules as PCFG, use variant of natural logic calculus (MacCartney and Manning, 2009).

  - Results in an interesting probabilistic logic, efficient proof search via reduction to (P)CFG search.

<table>
<thead>
<tr>
<th>sentence</th>
<th>analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>Pink 3 passes to Pink 7</td>
</tr>
<tr>
<td>h</td>
<td>Pink 3 quickly kicks to Pink 7</td>
</tr>
</tbody>
</table>

| inference (human) $t \rightarrow h$ | Unknown (RTE) |
| inference (LF match) $t \rightarrow h$ | Entail (RTE) |

<table>
<thead>
<tr>
<th>Inference Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>33.1%</td>
</tr>
<tr>
<td>LF Matching</td>
<td>59.6%</td>
</tr>
<tr>
<td><strong>Logical Inference Model</strong></td>
<td>73.4%</td>
</tr>
</tbody>
</table>
Jointly training semantic parsers to reason about entailment.

Created a novel semantic parsing model that supports joint probabilistic symbolic reasoning:

- We achieve state-of-the-art performance on the original semantic parsing task.
- Allows for evaluating semantic parsers on entailment tasks, perform domain-specific reasoning.
<Conclusions>
Introduced several new algorithmic/learning techniques, tasks and resources for helping making semantic parsing easier.

- 45 new multilingual datasets in the software domain, and a novel text-to-signature task and set of models.
- A new graph decoding framework, which allows for polyglot modeling, new mixed language dataset and task, improve results on code datasets.
- A new learning framework and dataset for entailment modeling and semantic parsing, state-of-the-art results on original task.
Thank You
References


