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): $\lambda x.\texttt{flight}(x) \land \texttt{depart}(x, \texttt{bos}) \land \texttt{arrive}(x, \texttt{ny})$

Main Topic: Semantic Parsing

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

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

$$\rightarrow$$

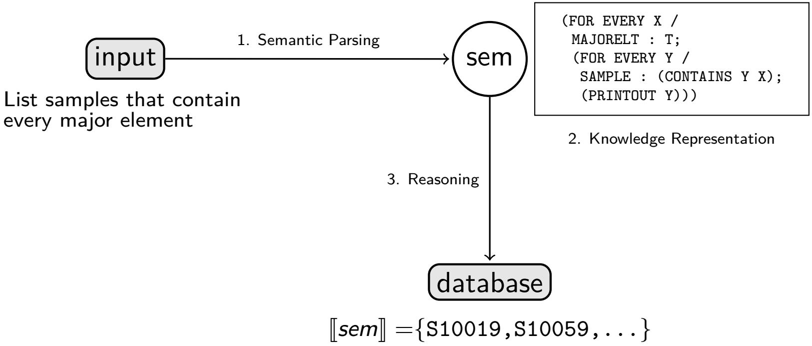
Logical Form (LF): $\lambda x.\texttt{flight}(x) \wedge \texttt{depart}(x, \texttt{bos}) \wedge \texttt{arrive}(x, \texttt{ny})$

> *"**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]**..."*

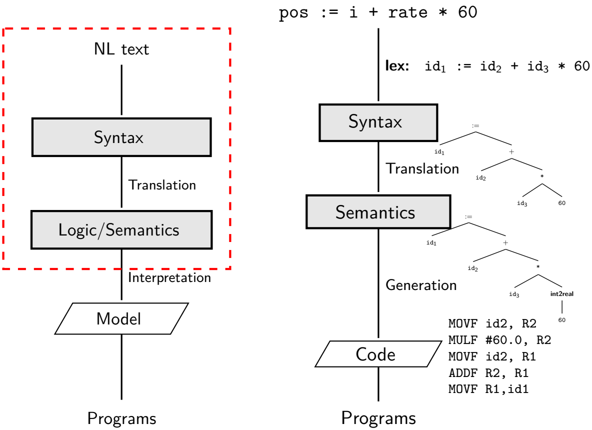
R.F. Simmons. 1965, *Answering English Question by Computer: A Survey*. Communications of the ACM

Classical Natural Language Understanding (NLU)

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

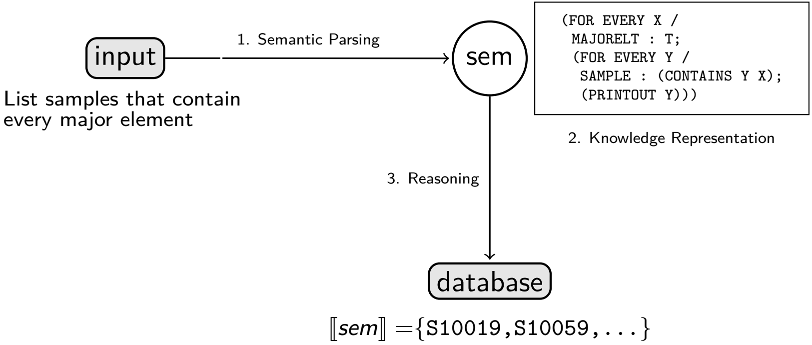


Why and How? Analogy with Compiler Design



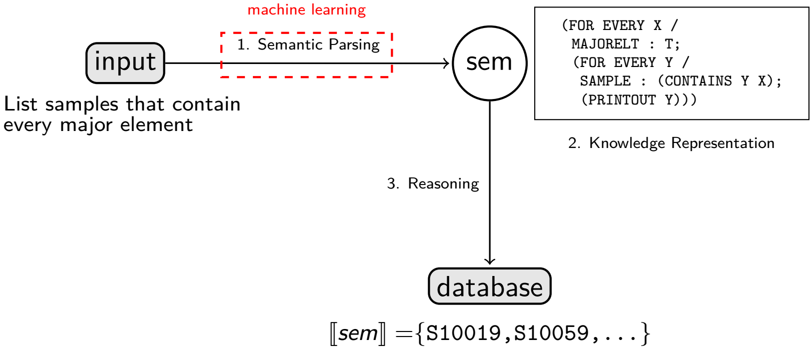
- NLU model is a kind of compiler, involves a **transduction** from NL to a formal (usually logical) language.

Data-driven NLU and Semantic Parsing



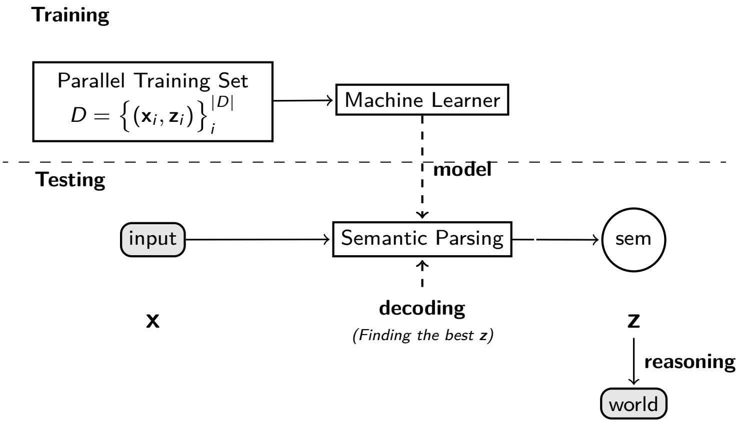
- **Data-driven NLU**: Asks an empirical question: Can we learn NLU models from examples?

Data-driven NLU and Semantic Parsing

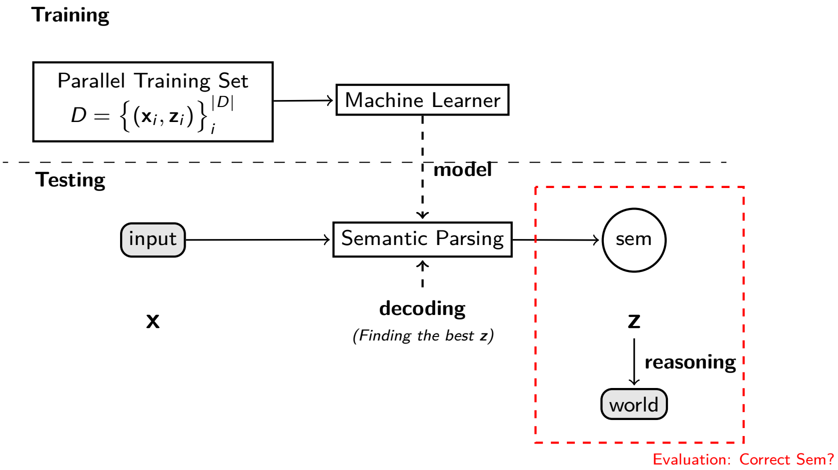


- **Data-driven NLU**: Asks an empirical question: Can we learn NLU models from examples?
  - **Semantic Parser Induction:** Learn semantic parser (i.e., translation to LFs) automatically from example parallel data.

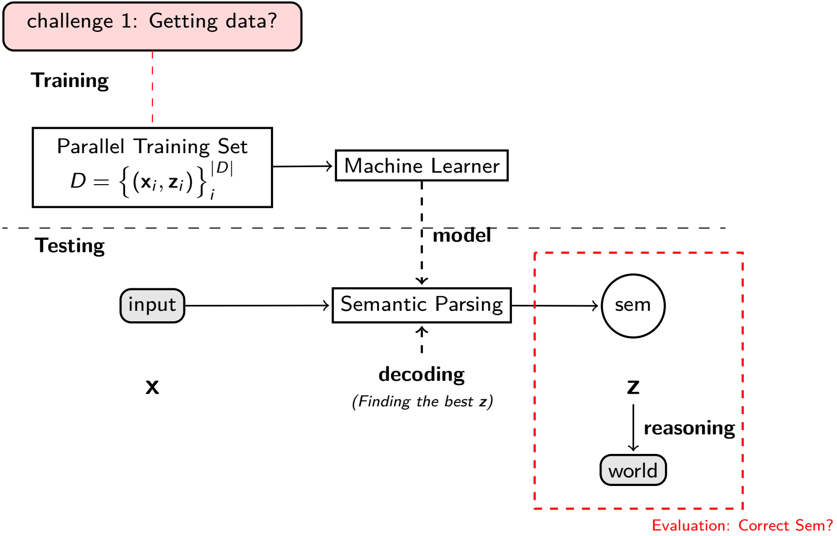
Data-driven Semantic Parsing in a Nutshell



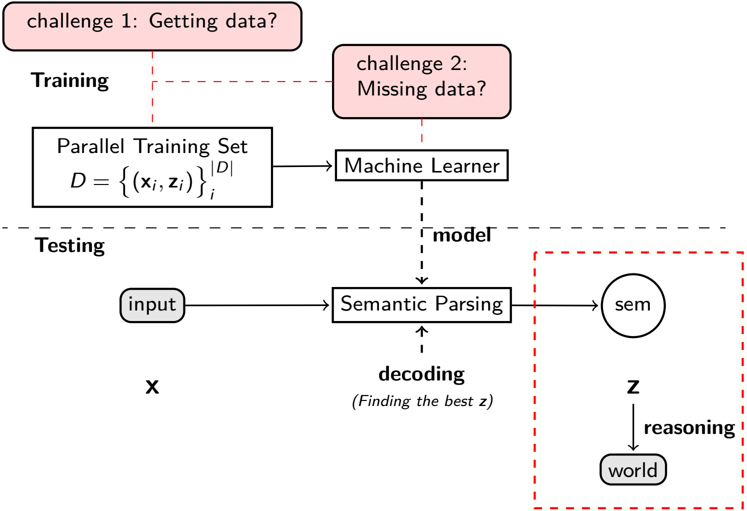
Data-driven Semantic Parsing in a Nutshell



Data-driven Semantic Parsing in a Nutshell

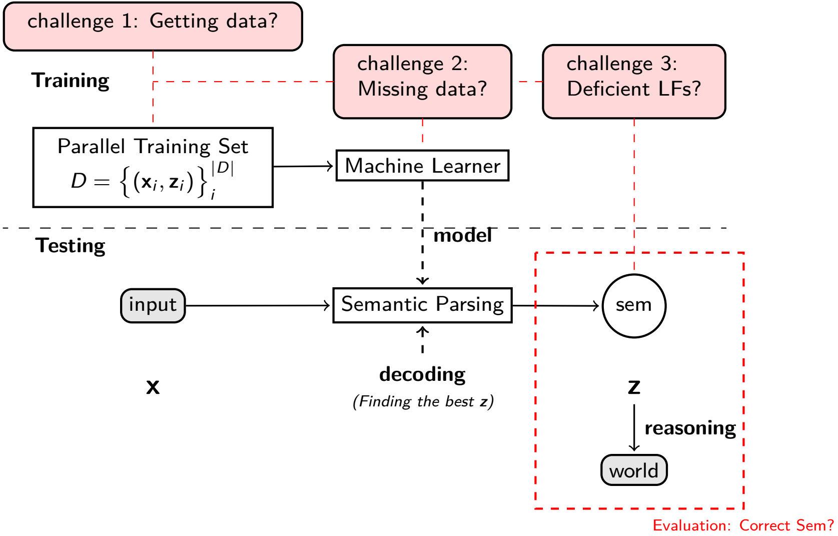


Data-driven Semantic Parsing in a Nutshell



Evaluation: Correct Sem?

Data-driven Semantic Parsing in a Nutshell



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.

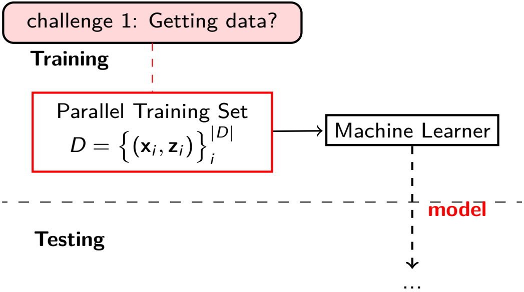
Richardson*, Berant and Kuhn 2018. NAACL*

challenge 3: Deficient LFs?

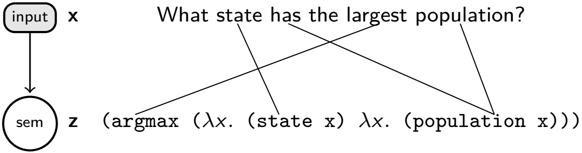
Train semantic parsers using entailment information; introduce new learning framework: *learning from entailment*.

Richardson *and Kuhn 2016. TACL*

<Challenge 1>

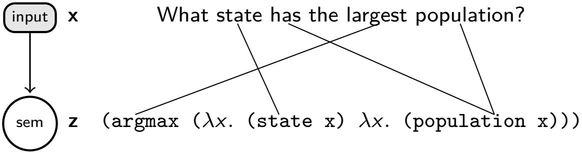

challenge 1: Getting data?
Training
Parallel Training Set
$D = \left\{ (\mathbf{x}_i, \mathbf{z}_i) \right\}_i^{|D|}$
Machine Learner
model
Testing
...

Semantic Parsing and Parallel Data



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

Semantic Parsing and Parallel Data



- **Learning from LFs**: Assumes pairs of text **x** and full logical forms **z**, goal is to learn $\texttt{sem} : \mathbf{x} \rightarrow \mathbf{z}$, 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

```
* 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.
- **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
* @see java.lang.Long#MAX_VALUE
*/
 public static Long max(long a, long b)
```

$\Downarrow$ extraction

text	*Returns the greater...*
code	lang.Math long max(long...)

```
(ns ... clojure.core)

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

$\Downarrow$ extraction

text	*Returns items from coll...*
code	(core.random-sample prob...)

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

$$\text{Signature} ::= \underbrace{\texttt{lang}}_{\texttt{namespace}}\ \underbrace{\texttt{Math}}_{\texttt{class}}\ \underbrace{\texttt{long}}_{\texttt{return}}\ \underbrace{\texttt{max}}_{\texttt{name}}\left(\underbrace{\texttt{long a, long b}}_{\texttt{named/typed arguments}}\right)$$

New Resources: Stdlib and Py27 Datasets

Dataset	#Pairs	#Symbols	#Words	Vocab.	Example Pairs (**x**, **z**)
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)` $\to$ `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

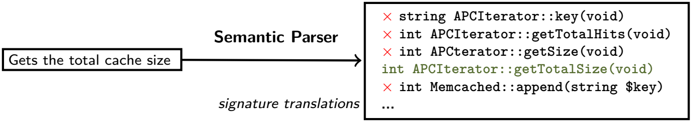
Dataset	#Pairs	#Symbols	#Words	Vocab.	Example Pairs (**x**, **z**)
Java	7,183	4,072	82,696	3,721	x : Compares this Calendar to the specified Object. z : `boolean util.Calendar.equals(Object obj)`
Ruby	6,885	3,803	67,274	5,131	x : Computes the arc tangent given y and x. z : `Math.atan2(y,x)` $\rightarrow$ `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 Task: Text to Signature Translation

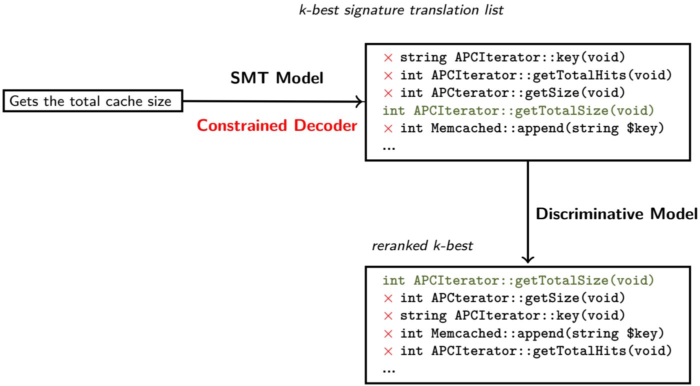
text	*Returns the greater of two long values*
signature	`lang.Math long max( long a, long b )`

- **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?

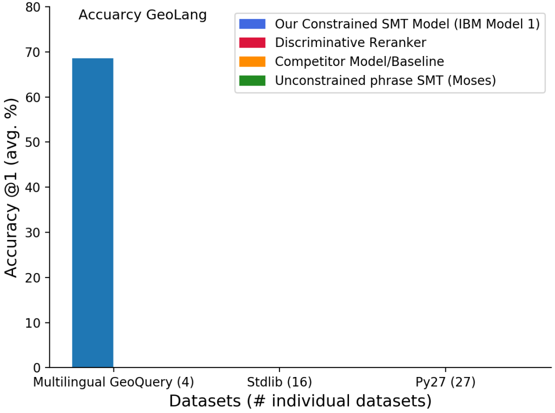
- **A First Model**: Use statistical (word-based) machine translation (SMT) (Deng and Chrupała, 2014) and reranking.



- **Decoding**: finding the best output given input, **unconstrained** versus **constrained** (assign probability to wellformed output only).

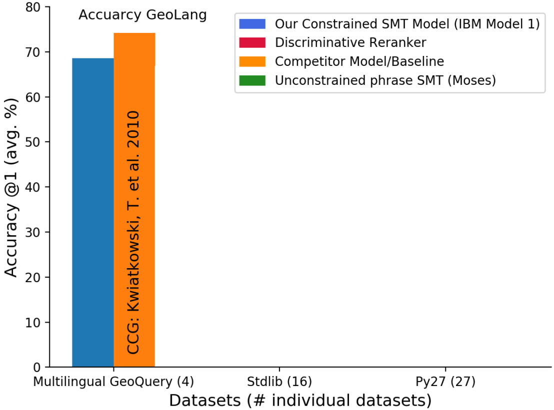
Text to Signature Translation: How Hard Is It?

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



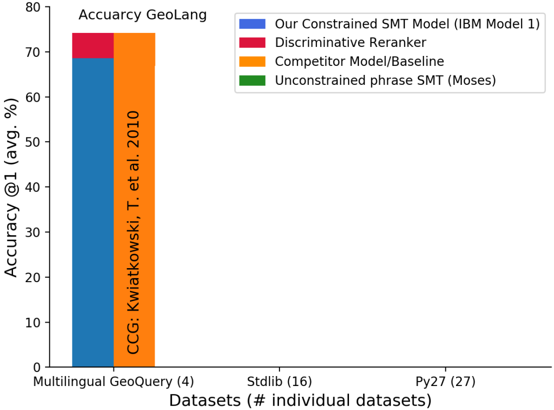
Text to Signature Translation: How Hard Is It?

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



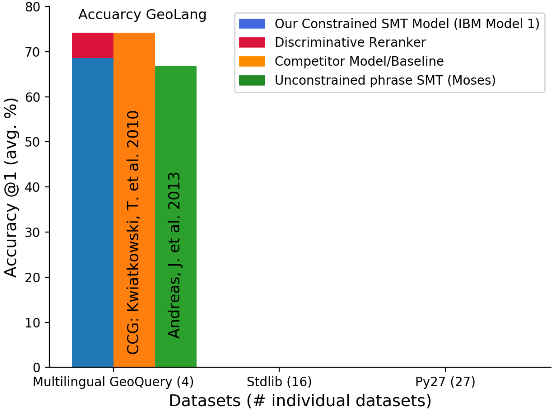
Text to Signature Translation: How Hard Is It?

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



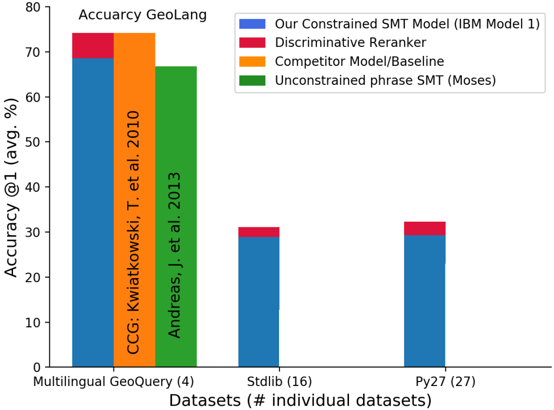
Text to Signature Translation: How Hard Is It?

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



Text to Signature Translation: How Hard Is It?

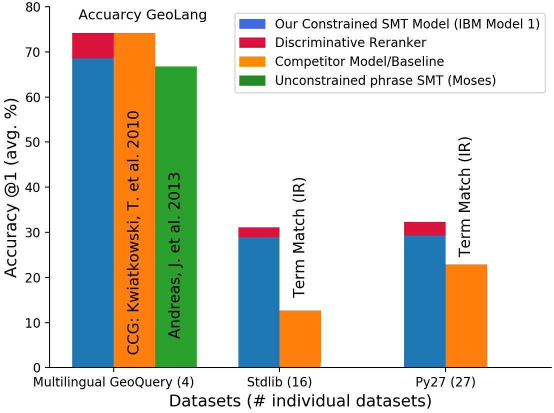
- How does such a simple approach fare on benchmark tasks and our task?



- **Result**: achieving high accuracy is not easy, not a trivial problem.

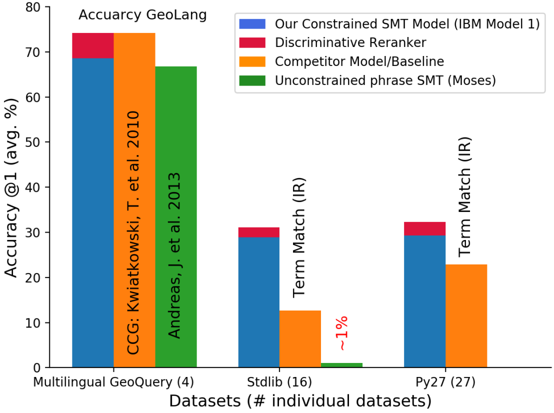
Text to Signature Translation: How Hard Is It?

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



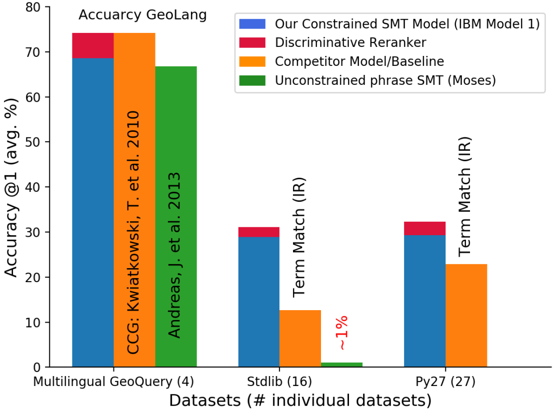
Text to Signature Translation: How Hard Is It?

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



Text to Signature Translation: How Hard Is It?

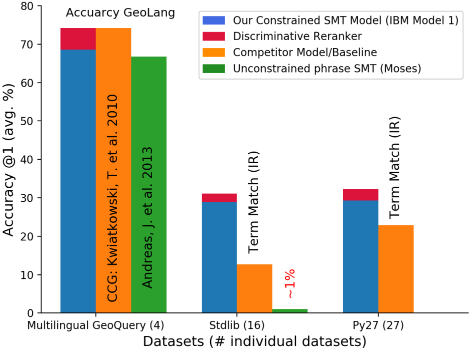
- How does such a simple approach fare on benchmark tasks and our task?



Text to Signature Translation: How Hard Is It?

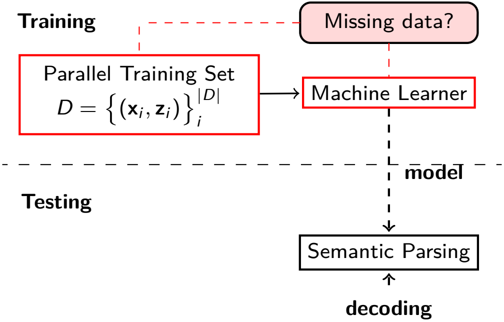
text	*Returns the index of the first occurrrence of char in the string*
Moses	(`start end occurrence lambda char string string string`)

- **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
Missing data?
Parallel Training Set
$D = \left\{(\mathbf{x}_i, \mathbf{z}_i)\right\}_i^{|D|}$
Machine Learner
model
Testing
Semantic Parsing
decoding

Challenge 2: Insufficient and Missing Data

(en,PHP) → $\theta_{\texttt{en}} \rightarrow$ `Java`

(en,Lisp) → $\theta_{\texttt{en}} \rightarrow$ `Lisp`

(es, PHP) → $\theta_{\texttt{es}} \rightarrow$ `PHP`

(ja, Python) → $\theta_{\texttt{ja}} \rightarrow$ `Python`

(en, Haskell) → $\theta_{\texttt{en}} \rightarrow$ `Haskell`

...

?$\theta_{\texttt{es}} \rightarrow$ `Java`?

- 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

(en,`Java`)
(en,`Lisp`)
(es, `PHP`)
(ja, `Python`)
(en, `Haskell`)
...

$\longrightarrow \theta_{\texttt{polyglot}}$

- **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.

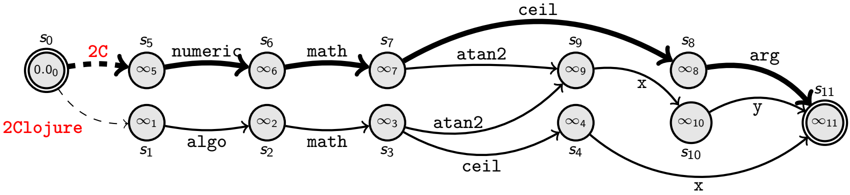
Polyglot Models: Training on Multiple Datasets

(en,`Java`)
(en,`Lisp`)
(es, `PHP`)
(ja, `Python`)
(en, `Haskell`)
...

$\longrightarrow \theta_{\texttt{polyglot}}$

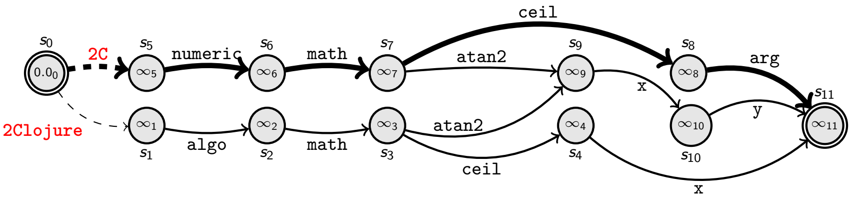
- **Idea**: concatenate all datasets, build a single-model with shared parameters, capture redundancy (Herzig and Berant, 2017).
- **Polyglot Translator:** translates from any input language to any output (programming) language.
  1. **Multiple Datasets:** Does this help learn better semantic parsers?
  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 *artifical language tokens*.

Graph-Based Constrained Decoding



- **Idea**: Represent full translation search space as directed graph, add *artifical language tokens*.

- **Decoding/Search** (test time): Find a path given an input $\mathbf{x}$:

$$\mathbf{x}: \text{ The ceiling of a number}$$

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:

$$\underbrace{d[v]}_{\substack{\uparrow \\ \text{node score}}} = \min_{(u,v,z)\in E} \Big\{ \underbrace{d[u]}_{\substack{\uparrow \\ \text{incoming node score}}} + \underbrace{w(u,v,z)}_{\substack{\uparrow \\ \text{edge score}}} \Big\}$$

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:

$$\underbrace{d[v]}_{\substack{\uparrow \\ \text{node score}}} = \min_{(u,v,z)\in E} \Big\{ \underbrace{d[u]}_{\substack{\uparrow \\ \text{incoming node score}}} + \underbrace{\textcolor{red}{\text{TRANS}}(\mathbf{x}, z)}_{\substack{\uparrow \\ \textcolor{red}{\text{translation}}}} \Big\}$$

  - 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:

$$\underbrace{d[v]}_{\substack{\uparrow \\ \text{node score}}} = \min_{(u,v,z)\in E} \left\{ \underbrace{d[u]}_{\substack{\uparrow \\ \text{incoming node score}}} + \underbrace{\textrm{TRANS}(\mathbf{x}, z)}_{\substack{\uparrow \\ \text{translation}}} \right\}$$

  - 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):

$$\begin{aligned} p(\mathbf{z} \mid \mathbf{x}) &= \textsc{ConditionalRNNLM}(\mathbf{z}) \\ &= \prod_{i}^{|\mathbf{z}|} p_{\Theta}(z_i \mid z_{<i}, \mathbf{x}) \end{aligned}$$

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):

$$
\begin{aligned}
p(\mathbf{z} \mid \mathbf{x}) &= \textsc{ConditionalRNNLM}(\mathbf{z}) \\
&= \prod_{i}^{|\mathbf{z}|} p_{\Theta}(z_i \mid z_{<i}, \mathbf{x})
\end{aligned}
$$

- **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):

$$p(\mathbf{z} \mid \mathbf{x}) = \textsc{ConditionalRNNLM}(\mathbf{z})$$
$$= \prod_{i}^{|\mathbf{z}|} p_{\Theta}(z_i \mid z_{<i}, \mathbf{x})$$

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

```
d[b] ← 0.0
for node v ∈ 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):

$$p(\mathbf{z} \mid \mathbf{x}) = \textsc{ConditionalRNNLM}(\mathbf{z}) = \prod_{i}^{|\mathbf{z}|} p_\Theta(z_i \mid z_{<i}, \mathbf{x})$$

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

0: $d[b] \leftarrow 0.0$
1: **for** node $v \in V$ in topologically sorted order
2: **do** $d(v) = \min_{(u,v,z_j) \in E} \left\{ d(u) + -\log p_\Theta(z_j \mid z_{<j}, \mathbf{x}) \right\}$

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):

$$p(\mathbf{z} \mid \mathbf{x}) = \textsc{ConditionalRNNLM}(\mathbf{z})$$
$$= \prod_{i}^{|\mathbf{z}|} p_{\Theta}(z_i \mid z_{<i}, \mathbf{x})$$

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

$d[b] \leftarrow 0.0$
for node $v \in V$ in topologically sorted order
    do $d(v) = \min_{(u,v,z_j) \in E} \Big\{ d(u) + - \log p_{\Theta}(z_j \mid z_{<j}, \mathbf{x}) \Big\}$
    $s[v] \leftarrow$ `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):

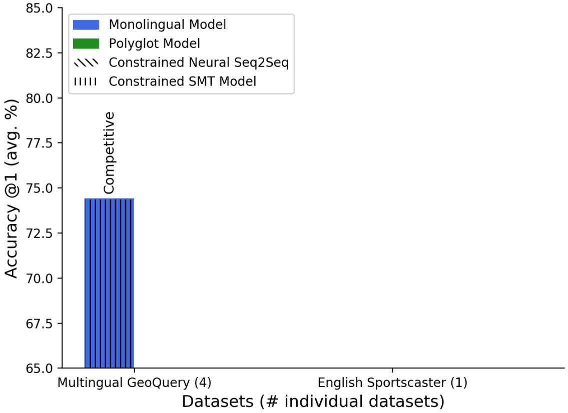
$$p(\mathbf{z} \mid \mathbf{x}) = \textsc{ConditionalRNNLM}(\mathbf{z})$$
$$= \prod_{i}^{|\mathbf{z}|} p_{\Theta}(z_i \mid z_{<i}, \mathbf{x})$$

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

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

Shortest Path Decoding: Comparing Models

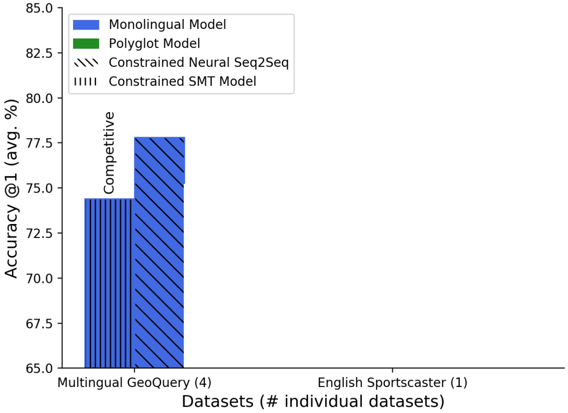
- **Shortest Path Decoding Framework**: Directly compare the performance of different semantic parsing models under a single search procedure.



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

Shortest Path Decoding: Comparing Models

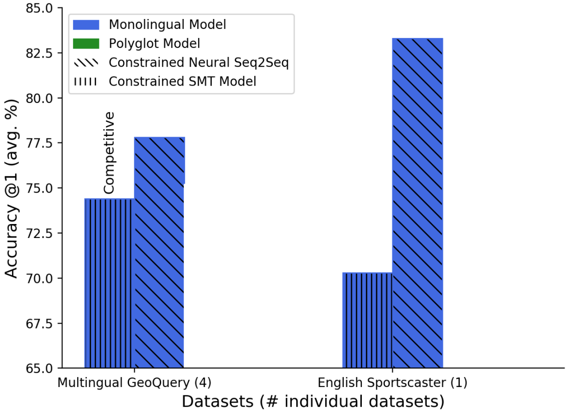
- **Shortest Path Decoding Framework**: Directly compare the performance of different semantic parsing models under a single search procedure.



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

Shortest Path Decoding: Comparing Models

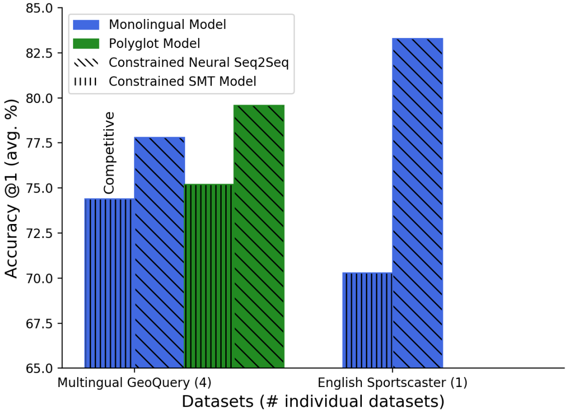
- **Shortest Path Decoding Framework**: Directly compare the performance of different semantic parsing models under a single search procedure.



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

Training on Multiple Datasets: Does this help?

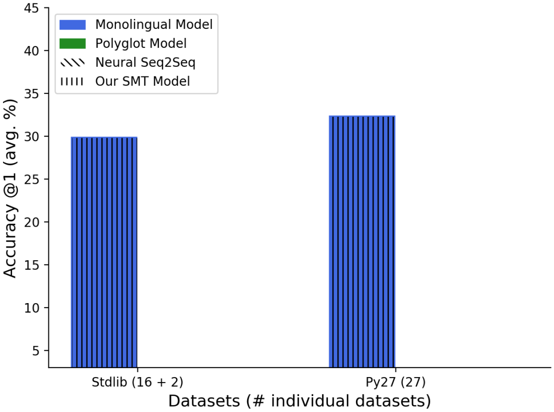
- **Polyglot Models**: Directly compare if training on multiple datasets improves translation.



- **Benchmark Datasets**: Training *polyglot models* on multiple datasets can increase performance, makes learning more robust

Training on Multiple Datasets: Does this help?

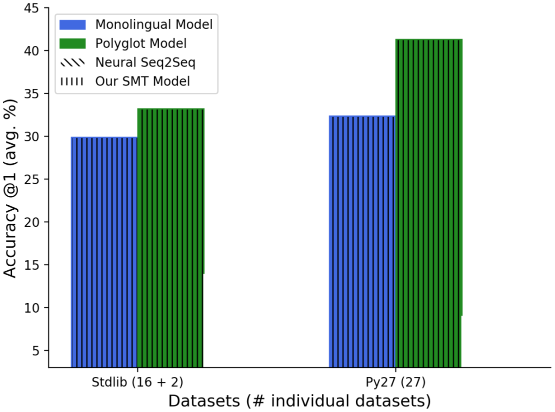
- **Polyglot Models**: Directly compare if training on multiple datasets improves translation.



- **Code Datasets**: Training *polyglot models* on multiple datasets can increase performance, depending on the model.

Training on Multiple Datasets: Does this help?

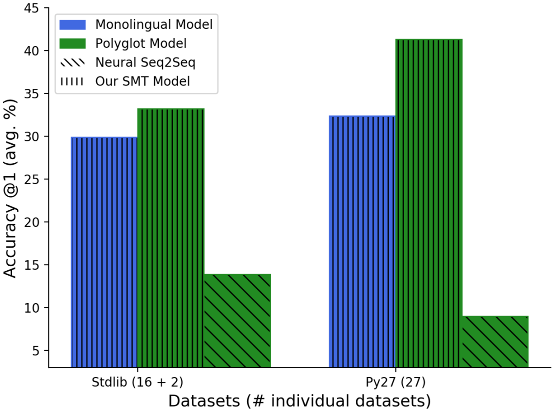
- **Polyglot Models**: Directly compare if training on multiple datasets improves translation.



- **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.

1.	Source API (stdlib): (***es*, PHP**)	**Input**: Devuelve el mensaje asociado al objeto lanzado.
Output	Language: **PHP**	Translation: `public string Throwable::getMessage ( void )`
	Language: **Java**	Translation: `public String lang.getMessage( void )`
	Language: **Clojure**	Translation: `(tools.logging.fatal throwable message & more)`
2.	Source API (stdlib): (***ru*, PHP**)	**Input**: конвертирует строку из формата UTF-32 в формат UTF-16.
Output	Language: **PHP**	Translation: `string PDF_utf32_to_utf16 ( ... )`
	Language: **Ruby**	Translation: `String#toutf16 => string`
	Language: **Haskell**	Translation: `Encoding.encodeUtf16LE :: Text -> ByteString`
3.	Source API (py): (***en*, stats**)	**Input:** Compute the Moore-Penrose pseudo-inverse of a matrix.
Output	Project: **sympy**	Translation: `matrices.matrix.base.pinv_solve( B, ... )`
	Project: **sklearn**	Translation: `utils.pinvh( a, cond=None,rcond=None,... )`
	Project: **stats**	Translation: `tools.pinv2( a,cond=None,rcond=None )`

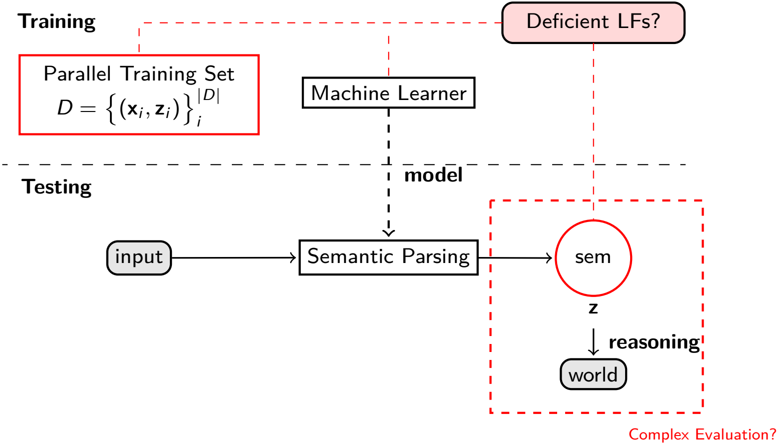
Advantages: Any/Mixed Language Decoding

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

Mixed Lang.	**Input**: Wie hoch liegt der höchstgelegene punkt in Αλαμπάμα?
LF: `answer(elevation_1(highest(place(loc_2(stateid('alabama'))))))`	

- 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.

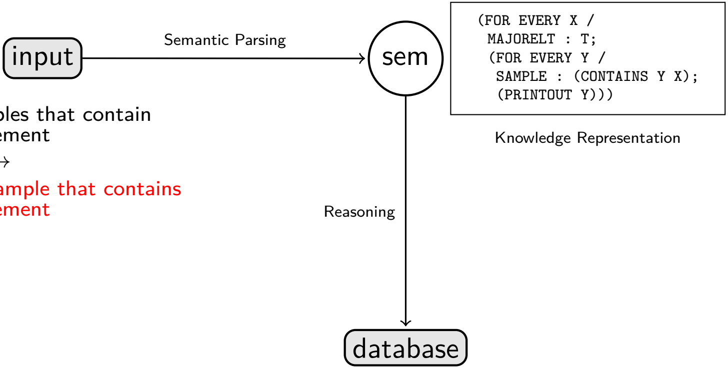
<Challenge 3>



Training
Deficient LFs?
Parallel Training Set
$D = \left\{(\mathbf{x}_i, \mathbf{z}_i)\right\}_i^{|D|}$
Machine Learner
model
Testing
input
Semantic Parsing
sem
z
reasoning
world
Complex Evaluation?

Semantic Parsing and Entailment

- **Entailment:** One of the *basic aims* of semantics (Montague, 1970)[1].



t. All samples that contain
a major element

$\rightarrow$

h. Some sample that contains
a major element

$[\![\textit{sem}]\!] = \{\texttt{S10019}, \texttt{S10059}, \ldots\} \supseteq \{S10019\}$

[1] Recognizing Textual Entailment (RTE): would a person reading `t` usually infer `h`? (Dagan et al., 2005), answers: { `Entail` (yes), `Contradict` (no), `Unknown` (possible) }

Semantic Parsing and Entailment

- **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).

	sentence	LF
t	*Pink 3 passes to Pink 7*	`pass(pink3,pink7)`
h	*Pink 3 quickly kicks to Pink 7*	`pass(pink3,pink7)`
inference (human) `t` → `h`		`Unknown` (RTE)
inference (LF match) `t` → `h`		`Entail` (RTE)

Semantic Parsing and Entailment

- **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).

	sentence	**LF**
t	*The pink goalie passes to pink 7*	`pass(pink1,pink7)`
h	*Pink 1 kicks the ball*	`kick(pink1)`
inference (human) $\texttt{t} \rightarrow \texttt{h}$		`Entail` (RTE)
inference (LF match) $\texttt{t} \rightarrow \texttt{h}$		`Contradict` (RTE)

Semantic Parsing and Entailment

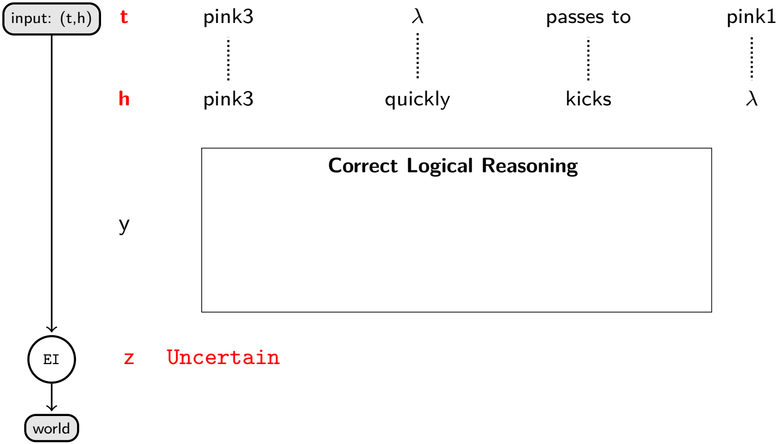
- **Question**: What happens if we *unit test* our semantic parsers using an RTE test?
- **Sportscaster**: ≈1,800 soccer descriptions paired with logical forms (LFs) (Chen and Mooney, 2008).

Inference Model	**Accuracy**
Majority Baseline	33.1%
RTE Classifier	52.4%
LF Matching	59.6%

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

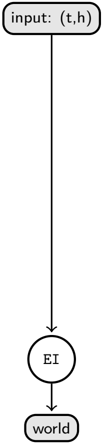
Learning from Entailment: Illustration

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



Learning from Entailment: Illustration

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



t	pink3	λ	passes to	pink1
h	pink3	quickly	kicks	λ

y

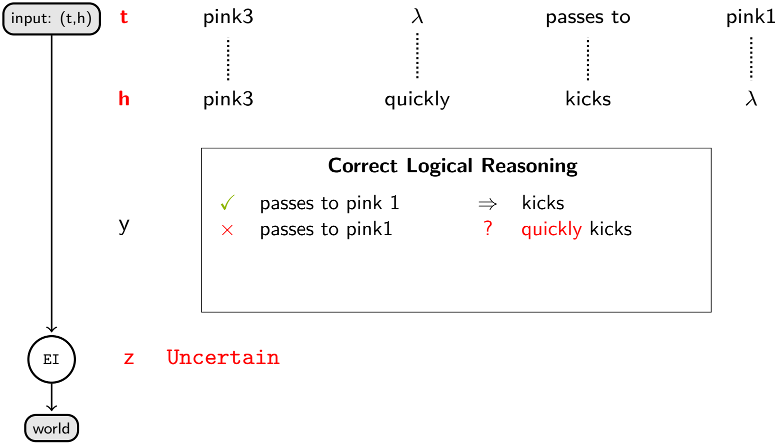
Correct Logical Reasoning

✓ passes to pink 1 $\Rightarrow$ kicks

z `Uncertain`

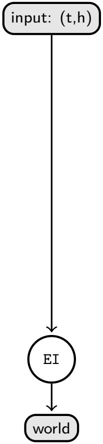
Learning from Entailment: Illustration

- 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.



t	pink3	λ	passes to	pink1
h	pink3	quickly	kicks	λ

y

Correct Logical Reasoning

✓	passes to pink 1	$\Rightarrow$	kicks
×	passes to pink1	?	quickly kicks
×	pink 3 passes to pink1	?	pink3 quickly kicks

? = Uncertain

z 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	λ	passes to	pink1
h	pink3	quickly	kicks	λ

y

Incorrect Reasoning

✓	passes to pink 1	$\Rightarrow$	kicks
✓	passes to pink1	$\Rightarrow$	quickly kicks
✓	pink 3 passes to pink1	$\Rightarrow$	pink3 quickly kicks

$\Rightarrow$ = `Entail`

EI

z `Uncertain`

world

Grammar Approach: Sentences to Logical Form

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

$(x$: purple 10 quickly kicks, z : $\{\texttt{kick(purple10)}$, block(purple7),...$\})$

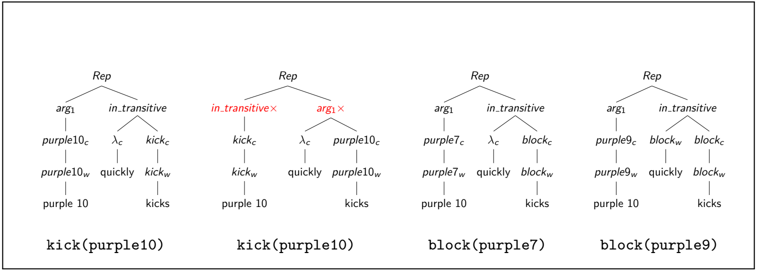
$\downarrow$ (rule extraction)

Grammar Approach: Sentences to Logical Form

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

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

$\downarrow$ (rule extraction)

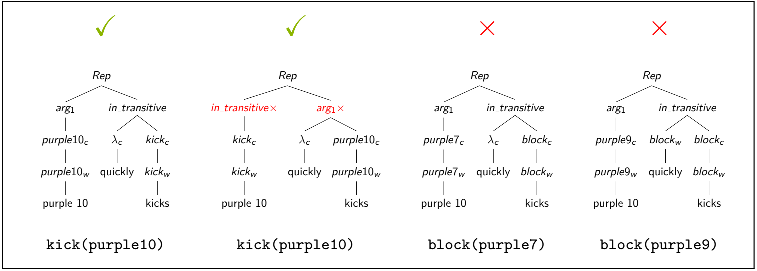


Grammar Approach: Sentences to Logical Form

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

$(x$: purple 10 quickly kicks, z : $\{$`kick(purple10)`, block(purple7),...$\})$

$\downarrow$ (rule extraction)

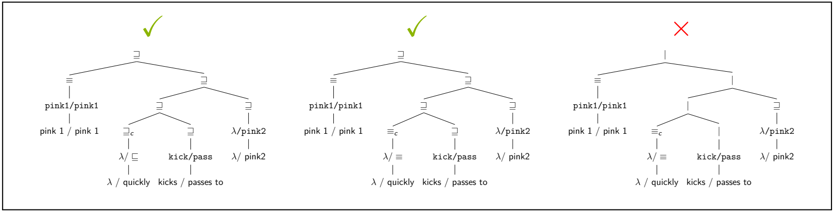


Modeling Entailments as Structured Proofs

- 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.

$$\Big(\big(\textbf{t}\text{: pink 1 kicks}, \textbf{h}\text{: pink 1 quickly passes to pink 2}\big)\text{ ,z: }\texttt{Uncertain}\Big)$$

$$\Big\downarrow \text{(inference rules)}$$



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).

$$\mathbf{x} = (\mathtt{t}, \mathtt{h}),\ \mathbf{z} \in \{\mathtt{Entail}, \mathtt{Contradict}, \mathtt{Unknown}\}$$

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).

$$\mathbf{x} = (\mathtt{t}, \mathtt{h}),\ \mathbf{z} \in \{\mathtt{Entail}, \mathtt{Contradict}, \mathtt{Unknown}\}$$

$$p(\mathbf{z} \mid \mathbf{x}) = \sum_{\underbrace{y \in \mathcal{Y}_{\mathbf{x}}}_{\text{proofs}}} \underbrace{p(\mathbf{z} \mid y)}_{\substack{\uparrow \\ \text{valid inference?}}} \times \underbrace{p_\theta(y \mid \mathbf{x})}_{\substack{\uparrow \\ \text{proof score}}}$$

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).

$$\mathbf{x} = (\mathtt{t}, \mathtt{h}),\ \mathbf{z} \in \{\mathtt{Entail}, \mathtt{Contradict}, \mathtt{Unknown}\}$$

$$p(\mathbf{z} \mid \mathbf{x}) = \underbrace{\sum_{y \in \mathcal{Y}_{\mathbf{x}}}}_{\text{proofs}} \underbrace{p(\mathbf{z} \mid y)}_{\substack{\uparrow \\ \text{valid inference?}}} \times \underbrace{p_{\theta}(y \mid \mathbf{x})}_{\substack{\uparrow \\ \text{proof score}}}$$

  - $p(\mathbf{z} \mid y)$: 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).

$$\mathbf{x} = (\mathtt{t}, \mathtt{h}),\ \mathbf{z} \in \{\mathtt{Entail}, \mathtt{Contradict}, \mathtt{Unknown}\}$$

$$p(\mathbf{z} \mid \mathbf{x}) = \underbrace{\sum_{y \in \mathcal{Y}_{\mathbf{x}}}}_{\text{proofs}} \underbrace{p(\mathbf{z} \mid y)}_{\substack{\uparrow \\ \text{valid inference?}}} \times \underbrace{p_\theta(y \mid \mathbf{x})}_{\substack{\uparrow \\ \text{proof score}}}$$

  - $p(\mathbf{z} \mid y)$: 1 if proof derives correct entailment, 0 otherwise
  - $p_\theta(y \mid \mathbf{x})$: 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).

$$\mathbf{x} = (\mathtt{t}, \mathtt{h}),\ \mathbf{z} \in \{\mathtt{Entail}, \mathtt{Contradict}, \mathtt{Unknown}\}$$

$$p(\mathbf{z} \mid \mathbf{x}) = \underbrace{\sum_{y \in \mathcal{Y}_\mathbf{x}}}_{\text{proofs}} \underbrace{p(\mathbf{z} \mid y)}_{\substack{\uparrow \\ \text{valid inference?}}} \times \underbrace{p_\theta(y \mid \mathbf{x})}_{\substack{\uparrow \\ \text{proof score}}}$$

  - $p(\mathbf{z} \mid y)$: 1 if proof derives correct entailment, 0 otherwise
  - $p_\theta(y \mid \mathbf{x})$: 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.

Improved Semantic Parsing and RTE Testing

- **New Evaluation**: Can my semantic parser solve RTE tasks? New Sportscaster inference corpus, ≈460 RTE pairs.

	sentence	analysis
t	*Pink 3 passes to Pink 7*	`pass(pink3,pink7)`
h	*Pink 3 quickly kicks to Pink 7*	`pass(pink3,pink7)`
inference (human) `t` → `h`		`Unknown` (RTE)
inference (`LF` match) `t` → `h`		`Entail` (RTE)

Inference Model	Accuracy
Majority Baseline	33.1%
LF Matching	59.6%
Logical Inference Model	73.4%

- 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 I

Allamanis, M., Tarlow, D., Gordon, A., and Wei, Y. (2015). Bimodal modelling of source code and natural language. In *International Conference on Machine Learning*, pages 2123–2132.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

Berant, J., Chou, A., Frostig, R., and Liang, P. (2013). Semantic Parsing on Freebase from Question-Answer Pairs. In *in Proceedings of EMNLP-2013*, pages 1533–1544.

Börschinger, B., Jones, B. K., and Johnson, M. (2011). Reducing grounded learning tasks to grammatical inference. In *Proceedings of EMNLP-2011*, pages 1416–1425.

Chen, D. L. and Mooney, R. J. (2008). Learning to sportscast: A test of grounded language acquisition. In *Proceedings of ICML-2008*, pages 128–135.

Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognizing textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*.

Deng, H. and Chrupała, G. (2014). Semantic approaches to software component retrieval with English queries. In *Proceedings of LREC-14*, pages 441–450.

Dong, L. and Lapata, M. (2016). Language to Logical Form with Neural Attention. *arXiv preprint arXiv:1601.01280*.

Guu, K., Pasupat, P., Liu, E., and Liang, P. (2017). From Language to Programs: Bridging Reinforcement Learning and Maximum Marginal Likelihood. In *Proceedings of ACL*.

References II

Herzig, J. and Berant, J. (2017). Neural semantic parsing over multiple knowledge-bases. In *Proceedings of ACL*.

Jia, R. and Liang, P. (2016). Data Recombination For Neural Semantic Parsing. *arXiv preprint arXiv:1606.03622*.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2016). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv preprint arXiv:1611.04558*.

Krishnamurthy, J., Dasigi, P., and Gardner, M. (2017). Neural Semantic Parsing with Type Constraints for Semi-Structured Tables. In *Proceedings of EMNLP*.

Liang, P., Jordan, M. I., and Klein, D. (2013). Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

MacCartney, B. and Manning, C. D. (2009). An extended model of natural logic. In *Proceedings of the eighth International Conference on Computational Semantics*, pages 140–156.

Montague, R. (1970). Universal grammar. *Theoria*, 36(3):373–398.

Richardson, K. (2018). A Language for Function Signature Representations. *arXiv preprint arXiv:1804.00987*.

Richardson, K., Berant, J., and Kuhn, J. (2018). Polyglot Semantic Parsing in APIs. In *Proceedings of NAACL*.

References III

Richardson, K. and Kuhn, J. (2012). Light Textual Inference for Semantic Parsing. In *Proceedings of COLING-12*.

Richardson, K. and Kuhn, J. (2014). Unixman corpus: A resource for language learning in the unix domain. In *Proceedings of LREC-2014*.

Richardson, K. and Kuhn, J. (2016). Learning to Make Inferences in a Semantic Parsing Task. *Transactions of the Association for Computational Linguistics*, 4:155–168.

Richardson, K. and Kuhn, J. (2017a). Function Assistant: A Tool for NL Querying of APIs. In *Proceedings of EMNLP*.

Richardson, K. and Kuhn, J. (2017b). Learning Semantic Correspondences in Technical Documentation. In *Proceedings of ACL*.

Richardson, K., Zarrieß, S., and Kuhn, J. (2017). The Code2Text Challenge: Text Generation in Source Code Libraries. In *Proceedings of INLG*.

Woods, W. A. (1973). Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, pages 441–450.

Yin, P. and Neubig, G. (2017). A Syntactic Neural Model for General-Purpose Code Generation. *arXiv preprint arXiv:1704.01696*.

Zarrieß, S. and Richardson, K. (2013). An Automatic Method for Building a Data-to-Text Generator. In *Proceedings of the 14th European Workshop on Natural Language Generation*.

Zelle, J. M. and Mooney, R. J. (1996). Learning Parse Database Queries using Inductive Logic Programming. In *Proceedings of AAAI-1996*, pages 1050–1055.