Carnegie Mellon University
School of Computer Science

CS11-747 Neural Networks for NLP
Neural Semantic Parsing

Pengcheng Yin
pcyin@cs.cmu.edu
Language Technologies Institute
Carnegie Mellon University

Language Technologies Institute

[Some contents are adapted from talks by Graham Neubig]

The Semantic Parsing Task

Motivation how to represent the meaning of the sentence?

Task Parsing natural language utterances into formal meaning representations (MRs)

Natural Language Utterance

Show me flights from Pittsburgh to Seattle

Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

The Semantic Parsing Task

Task-specific Meaning Representations designed for a specific task (e.g., question answering)

General-purpose Meaning Representations capture the semantics of natural language

Task-Specific
Meaning Representations

Show me flights from Pittsburgh to Seattle

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Task-specific Logical Form

Example: Smart Personal Agent
Question Answering Systems

General-Purpose
Meaning Representations

The boy wants to go

```
(want-01
    :arg0 (b / boy)
    :arg1 (g / go-01))
```

Abstract Meaning Representation (AMR)

Example: AMR, Combinatory Categorical Grammar (CCG)

Workflow of a (Task-specific) Semantic Parser

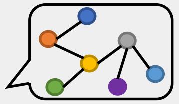
User's Natural Language Query

Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Query Execution



Freebase

WIKIDATA

Execution Results (Answer)

```
1. AS 119
2. AA 3544 -> AS 1101
3. …
```

Build natural language interfaces to computers

Task-specific Semantic Parsing: Datasets

- Domain-specific Meaning Representations and Languages
  - GEO Query, ATIS, JOBS
  - WikiSQL, Spider
  - IFTTT
- General-purpose Programming Languages
  - HearthStone
  - Django
  - CoNaLa

GEO Query, ATIS, JOBS

- **ATIS** 5410 queries about flight booking
- **GEO Query** 880 queries about US geographical information
- **JOBS** 640 queries to a job database

GEO Query

which state has the most rivers running through it?

```
argmax $0
 (state:t $0)
 (count $1 (and
             (river:t $1)
             (loc:t $1 $0)))
```

Lambda Calculus Logical Form

ATIS

Show me flights from Pittsburgh to Seattle

```
lambda $0 e
  (and (flight $0)
  (from $0 pittsburgh:ci)
  (to $0 seattle:ci))
```

Lambda Calculus Logical Form

JOBS

what microsoft jobs do not require a bscs?

```
answer(
  company(J,'microsoft'),
  job(J),
  not((req deg(J,'bscs'))))
```

Prolog-style Program

WikiSQL

Table: CFLDraft

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

Question:

How many CFL teams are from York College?

SQL:

```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = "York"
```

Result:

2

- 80654 examples of Table, Question and Answer
- **Context** a small database table extracted from a Wikipedia article
- **Target** a SQL query

[Zhong *et al.*, 2017]

IFTTT Dataset

- Over 70K user-generated task completion snippets crawled from ifttt.com
- Wide variety of topics: home automation, productivity, etc.
- Domain-Specific Language: IF-THIS-THEN-THAT structure, much simpler grammar

https://ifttt.com/applets/1p-autosave-your-instagram-photos-to-dropbox

IFTTT Natural Language Query and Meaning Representation

Autosave your Instagram photos to Dropbox

```
IF Instagram.AnyNewPhotoByYou
THEN Dropbox.AddFileFromURL
```

Domain-Specific Programming Language

[Quirk *et al.*, 2015]

HearthStone (HS) Card Dataset

- Description: properties/fields of an HearthStone card
- Target code: implementation as a Python class from HearthBreaker

Intent (Card Property)

<name> Divine Favor </name>
<cost> 3 </cost>
<desc> Draw cards until you have as many in hand as your opponent </desc>

Target Code (Python class)

```
class DivineFavor(SpellCard):
  def __init__(self):
    super().__init__("Divine Favor", 3, CHARACTER_CLASS.PALADIN,
                     CARD_RARITY.RARE)
  def use(self, player, game):
    super().use(player, game)
    difference = len(game.other_player.hand) - len(player.hand)
    for i in range(0, difference):
      player.draw()
```


[Ling *et al.*, 2016]

Django Annotation Dataset

- Description: manually annotated descriptions for 10K lines of code
- Target code: one liners
- Covers basic usage of Python like variable definition, function calling, string manipulation and exception handling

Intent *call the function _generator, join the result into a string, return the result*

Target

```
return ''.join(_generator())
```


[Oda *et al.*, 2015]

The CoNaLa Code Generation Dataset

Get a list of words `words` of a file 'myfile'

```
words = open('myfile').read().split()
```

Copy the content of file 'file.txt' to file 'file2.txt'

```
shutil.copy('file.txt’, 'file2.txt')
```

Check if all elements in list `mylist` are the same

```
len(set(mylist)) == 1
```

Create a key `key` if it does not exist in dict `dic` and append element `value` to value

```
dic.setdefault(key, []).append(value)
```

- 2,379 training and 500 test examples
- Manually annotated, high quality natural language queries
- Code is highly expressive and compositional
- Also ship with 600K extra mined examples!

conala-corpus.github.io

[Yin *et al.*, 2018]

Learning Paradigms

Supervised Learning

Utterances with Labeled Meaning Representation

Weakly-supervised Learning

Utterances with Query Execution Results

Semi-supervised Learning

Learning with Labeled and Unlabeled Utterances

Learning Paradigm 1: Supervised Learning

User's Natural Language Query

Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Train a neural semantic parser with source natural language query and target meaning representations

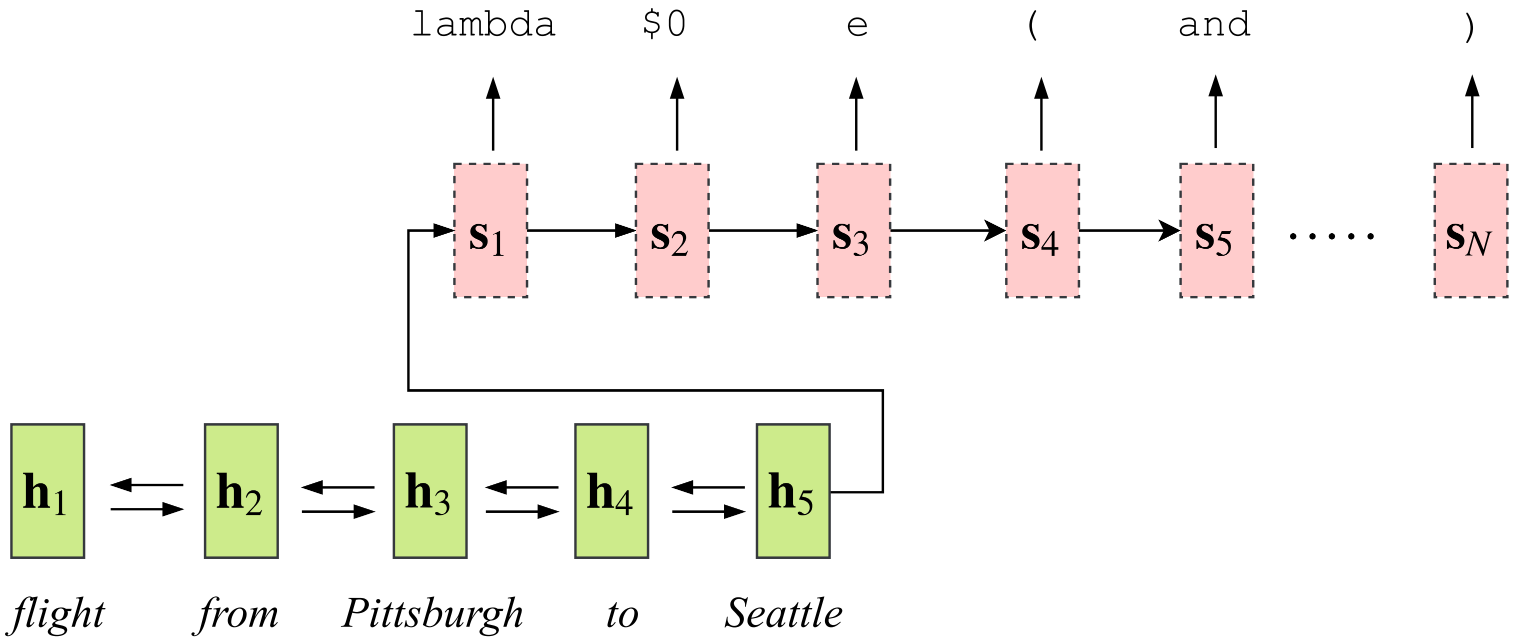
Sequence-to-Sequence Learning with Attention

Task-Specific
Meaning Representations

Show me flights from Pittsburgh to Seattle

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Task specific logical form



- Treat the target meaning representation as a sequence of surface tokens
- Reduce the task as another sequence-to-sequence learning problem

Language Technologies Institute

[Jia and Liang, 2016; Dong and Lapata, 2016]

Sequence-to-Sequence Learning with Attention

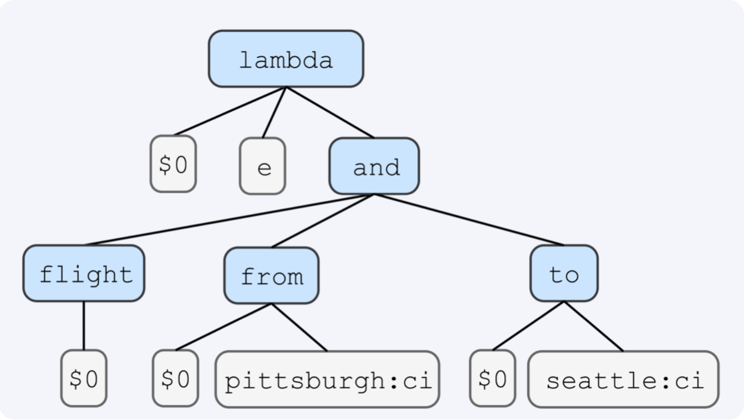
- **Meaning Representations** (e.g., a database query) have strong underlying structures!
- **Issue** Using vanilla seq2seq models ignore the rich structures of meaning representations

Task-Specific Meaning Representations

Show me flights from Pittsburgh to Seattle

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Task specific logical form

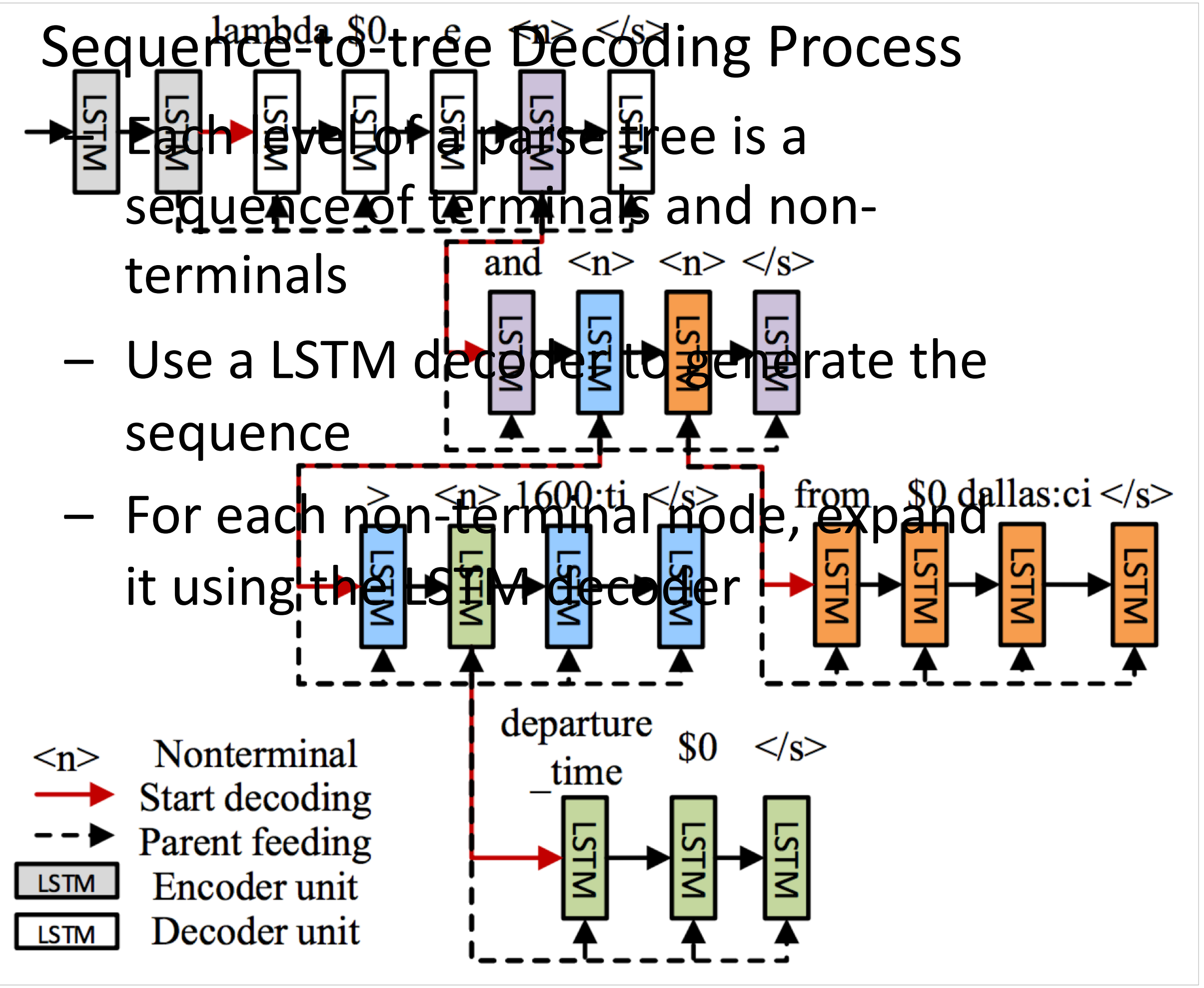


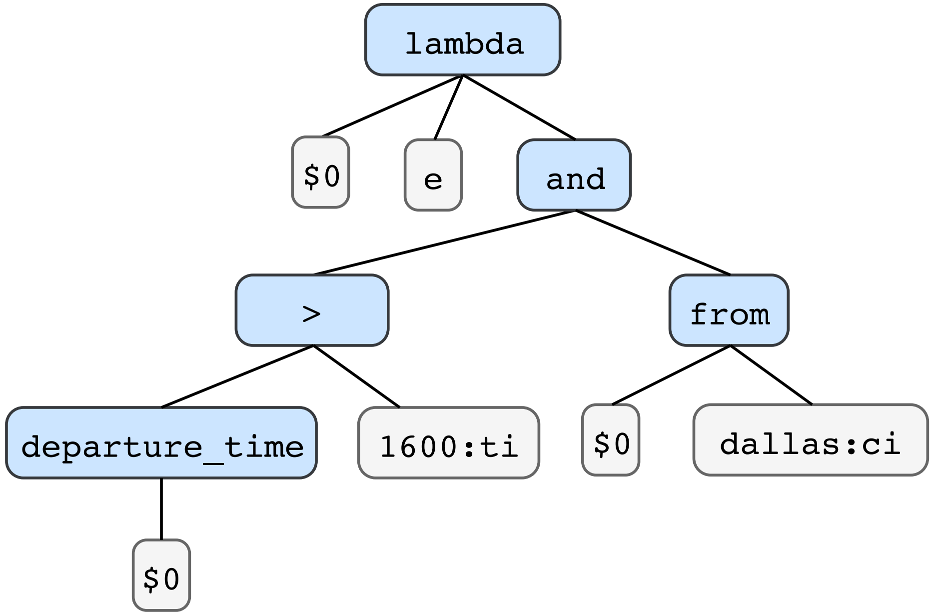
Tree-structured Representation

[Jia and Liang, 2016; Dong and Lapata, 2016]

Structure-aware Decoding for Semantic Parsing

- **Motivation** utilize the rich syntactic structure of target meaning representations
- **Seq2Tree** Generate from top-down using hierarchical sequence-to-sequence model
- Sequence-to-tree Decoding Process
  - Each level of a parse tree is a sequence of terminals and non-terminals
  - Use a LSTM decoder to generate the sequence
  - For each non-terminal node, expand it using the LSTM decoder



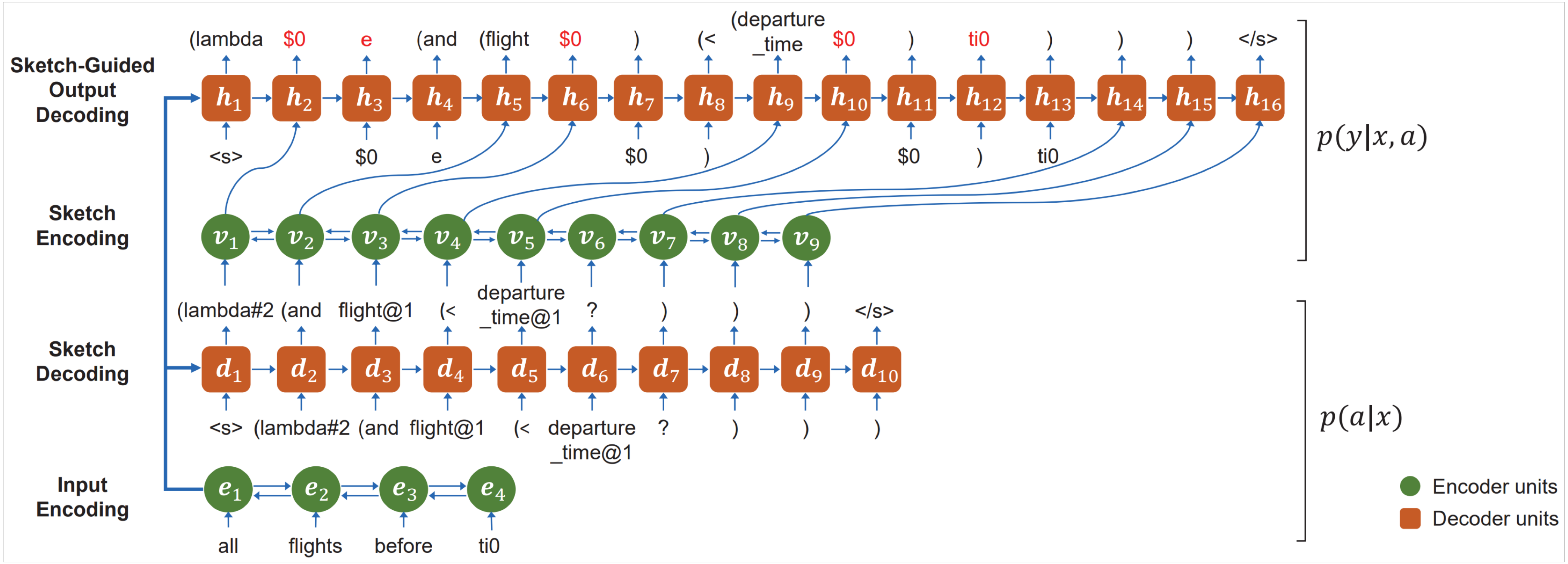


Show me flight from Dallas departing after 16:00

[Dong and Lapata, 2016]

Structure-aware Decoding (Cont'd)

- **Coarse-to-Fine Decoding** decode a coarse sketch of the target logical form first and then decode the full logical form conditioned on both the input query and the sketch
- Explicitly model the coarse global structure of the logical form, and use it to guide the parsing process



[Dong and Lapata, 2018]

Grammar/Syntax-driven Semantic Parsing

- Previously introduced methods only added structured components to the decoding model
- Meaning representations (e.g., Python) have strong underlying syntax
- How can we **explicitly** model the underlying syntax/grammar of the target meaning representations in the decoding process?

Python Abstract Grammar

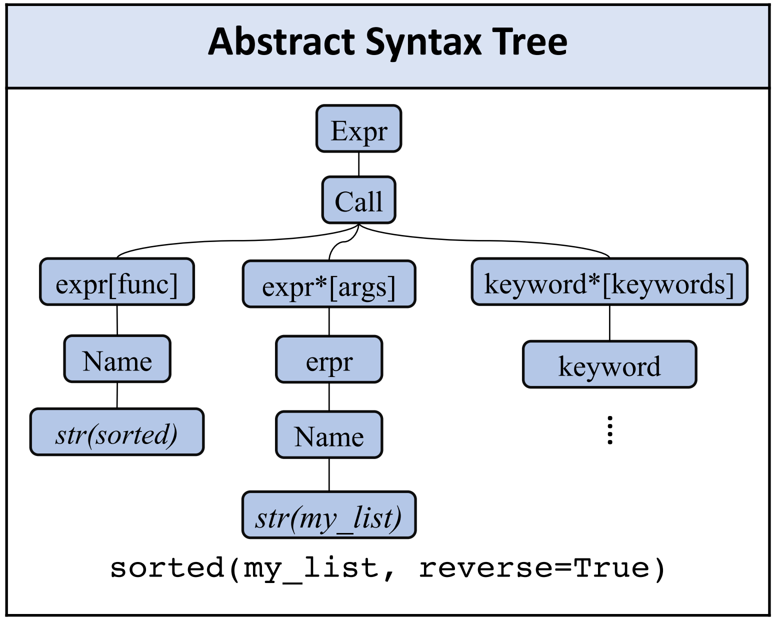
expr $\longmapsto$ Name | Call

Call $\longmapsto$ expr[*func*] expr*[*args*] keyword*[*keywords*]

If $\longmapsto$ expr[*test*] stmt*[*body*] stmt*[*orelse*]

For $\longmapsto$ expr[*target*] expr*[*iter*] stmt*[*body*] stmt*[*orelse*]

FunctionDef $\longmapsto$ identifier[*name*] expr*[*iter*] stmt*[*body*] stmt*[*orelse*]



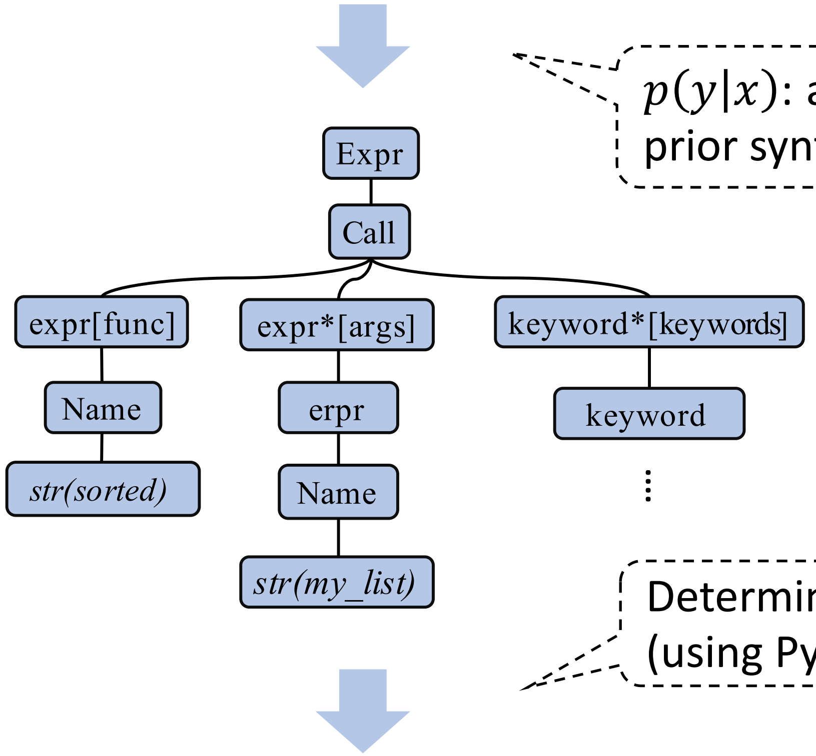
[Yin and Neubig, 2017; Rabinovich *et al.*, 2017]

Grammar/Syntax-driven Semantic Parsing

- Key idea: use the grammar of the target meaning representation (Python AST) as prior knowledge in a neural sequence-to-sequence model

Input Intent (x) *sort my_list in descending order*

Generated AST (y)



$p(y|x)$: a seq2seq model with prior syntactic information

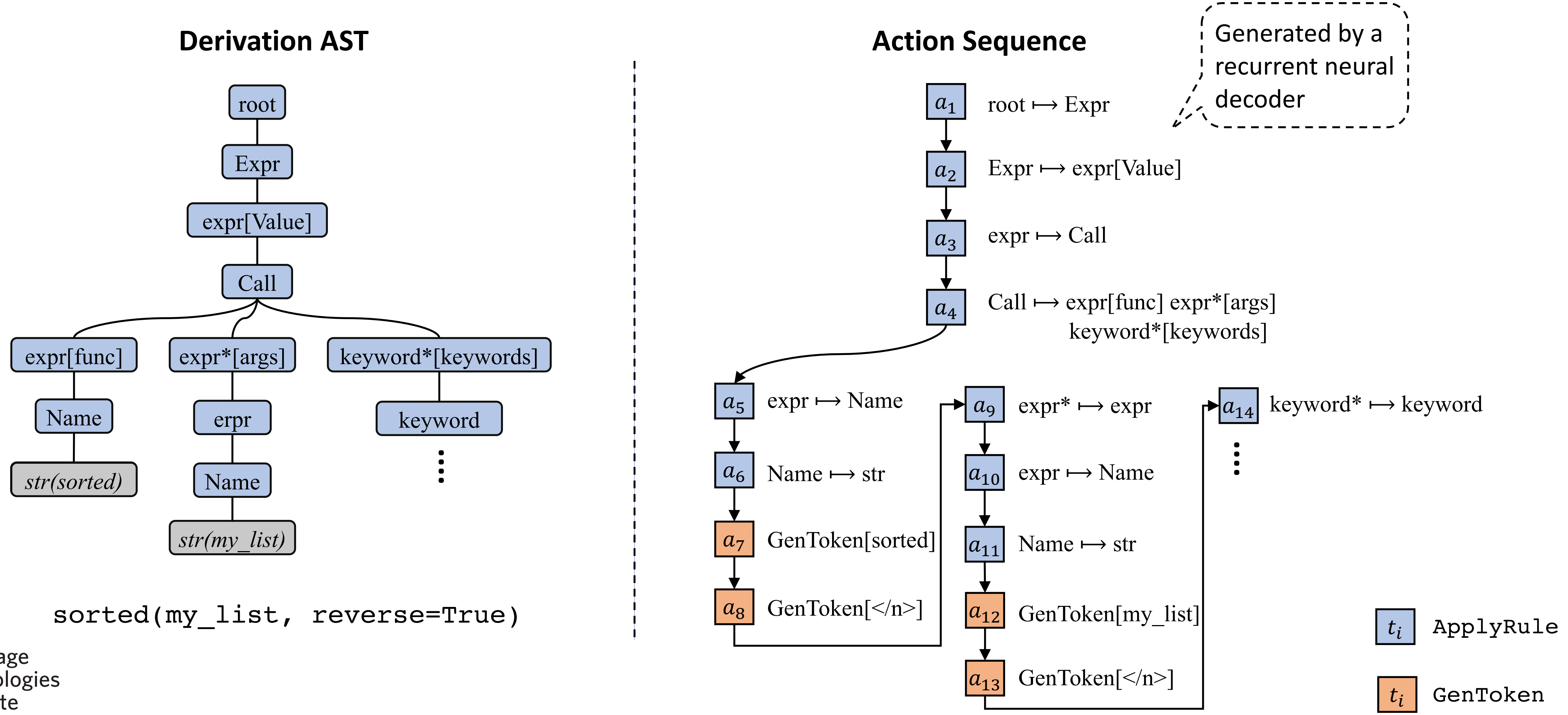
Deterministic transformation (using Python `astor` library)

Surface Code (c) `sorted(my_list, reverse=True)`

[Yin and Neubig, 2017; Rabinovich *et al.*, 2017]

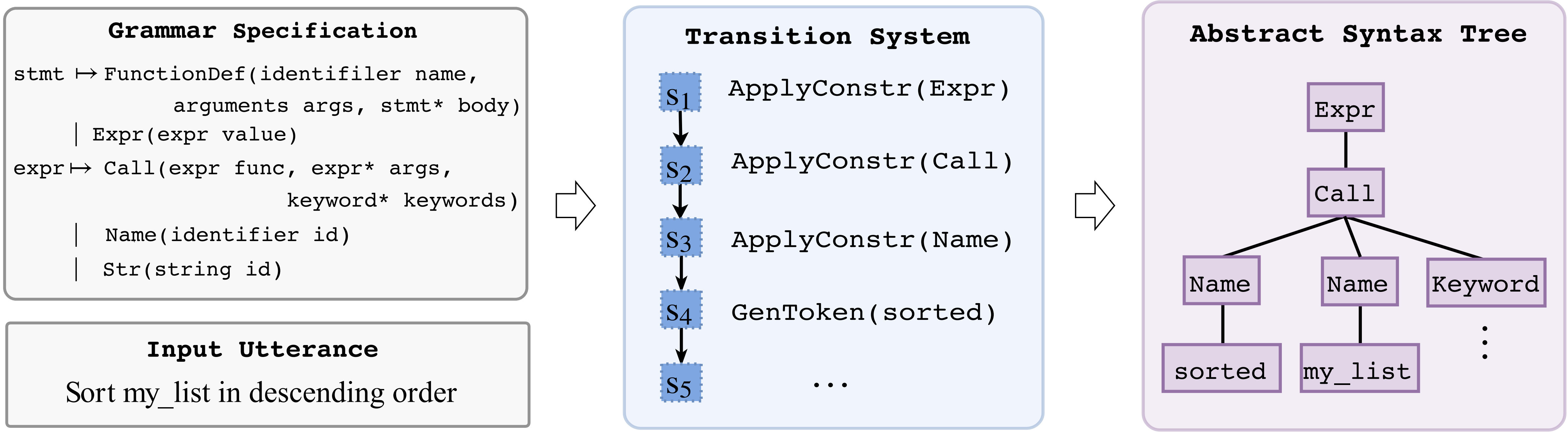
Grammar/Syntax-driven Semantic Parsing

- Factorize the generation story of an AST into sequential application of *actions* $\{a_t\}$:
  - `ApplyRule[r]`: apply a production rule r to the frontier node in the derivation
  - `GenToken[v]`: append a token v (e.g., variable names, string literals) to a terminal



TranX: a General-Purpose Syntax-Driven Semantic Parser

- Support five different meaning representations: Python 2 & 3, SQL, lambda-calculus, prolog

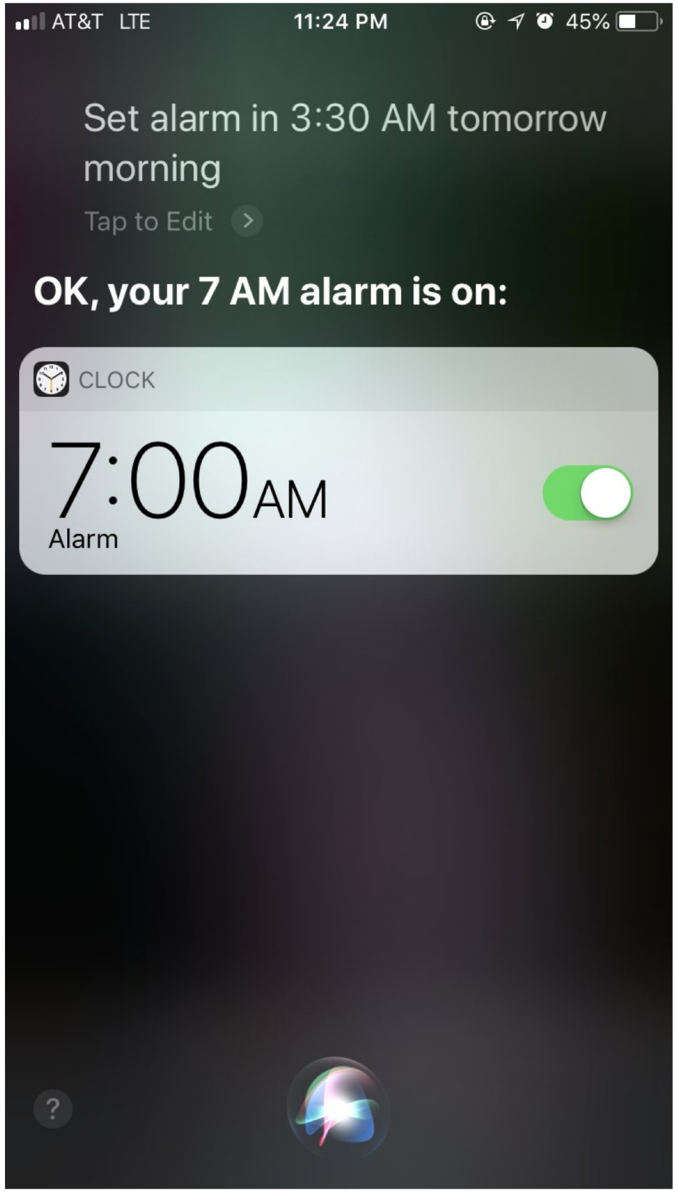


Open sourced at https://pcyin.me/tranX

[Yin and Neubig, 2018]

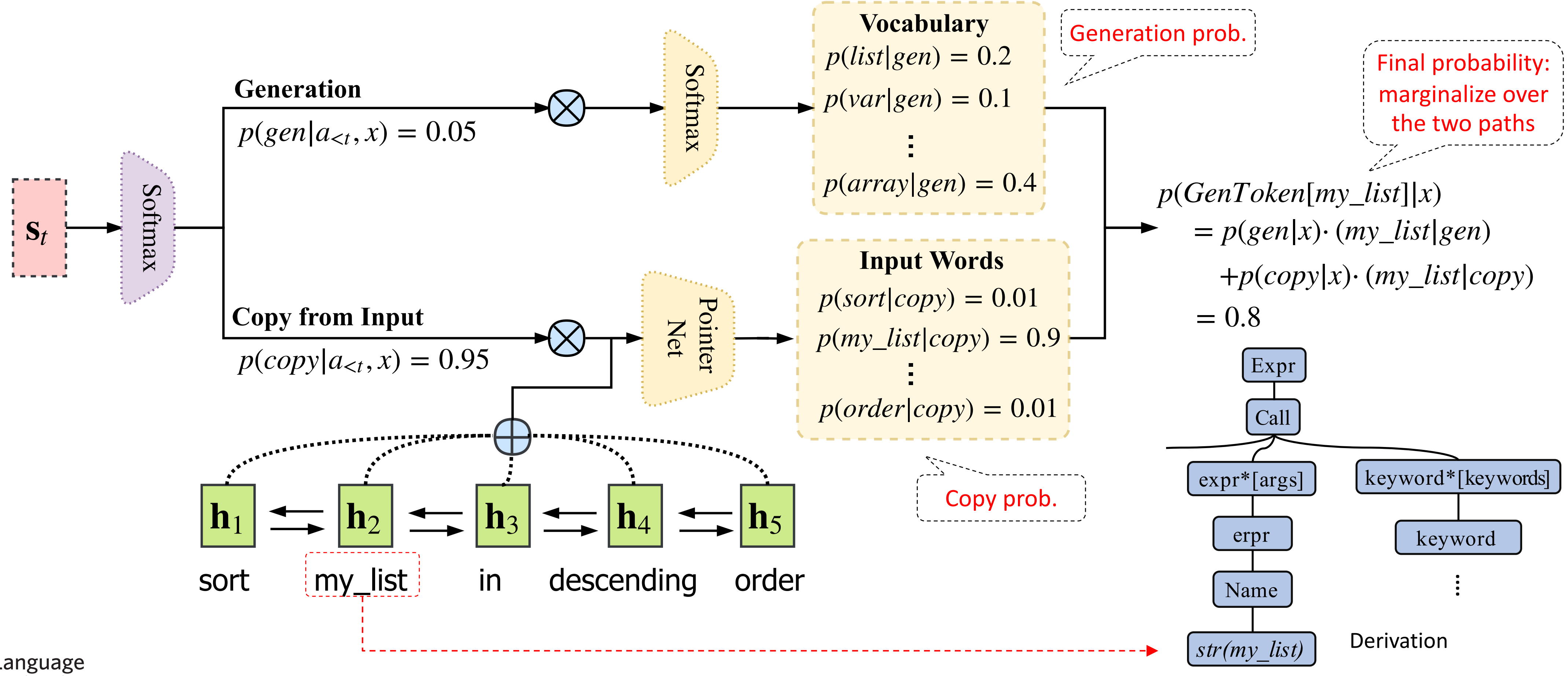
Side Note: Importance of Modeling Copying

- Modeling copying is very important for neural semantic parsers!
- Out-of-vocabulary entities (e.g., city names, date time) often appear in the input query
- Neural networks like to hallucinate entities not included in the input query ☺



Side Note: Importance of Modeling Copying

- Given a token ***v***, marginalize over the probability of copying ***v*** from the input and generating ***v*** from the close vocabulary



[Gu et al, 2016]

Importance of Modeling Copying: Examples

Intent *join app_config.path and string 'locale' into a file path, substitute it for localedir.*

Pred.
```
localedir = os.path.join(app_config.path, 'locale') ✓
```

Intent *self.plural is an lambda function with an argument n, which returns result of boolean expression n not equal to integer 1*

Pred.
```
self.plural = lambda n: len(n) ✗
```

Ref.
```
self.plural = lambda n: int(n!=1)
```

Intent *<name> Burly Rockjaw Trogg </name> <cost> 5 </cost> <attack> 3 </attack> <defense> 5 </defense> <desc> Whenever your opponent casts a spell, gain 2 Attack. </desc> <rarity> Common </rarity> ...*

Ref.
```
class BurlyRockjawTrogg(MinionCard):
    def __init__(self):
        super().__init__('Burly Rockjaw Trogg', 4, CHARACTER_CLASS.ALL, CARD_RARITY.COMMON)
    def create_minion(self, player):
        return Minion(3, 5, effects=[Effect(SpellCast(player=EnemyPlayer()),
            ActionTag(Give(ChangeAttack(2)), SelfSelector()))]) ✓
```

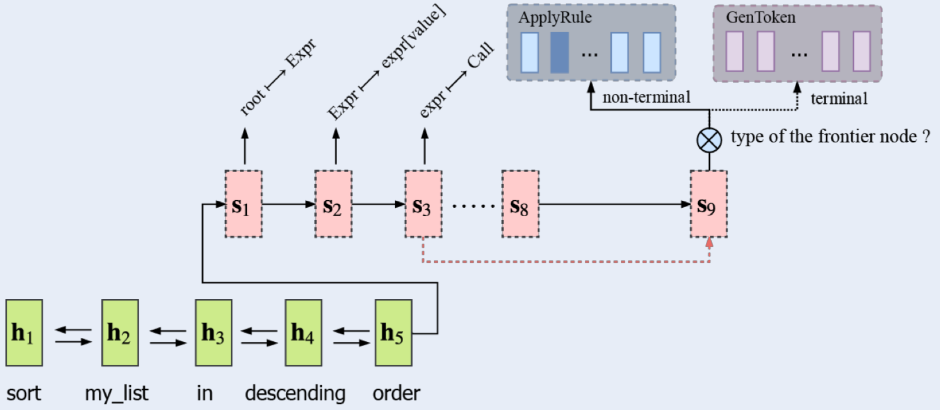
Language Technologies Institute

tokens copied from input

[Yin and Neubig, 2017]

Supervised Learning: the Data Inefficiency Issue

Supervised Parsers are Data Hungry



Purely supervised neural semantic parsing models require large amounts of training data

Data Collection is Costly

Copy the content of file 'file.txt' to file 'file2.txt'

```
shutil.copy('file.txt','file2.txt')
```

Get a list of words `words` of a file 'myfile'

```
words = open('myfile').read().split()
```

Check if all elements in list `mylist` are the same

```
len(set(mylist)) == 1
```

Collecting parallel training data costs and

*Examples from `conala-corpus.github.io` [Yin et al., 2018]

1700 USD for <3K Python code generation examples

Learning Paradigm 2: Weakly-supervised Learning

User's Natural Language Query

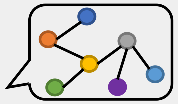
Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

As unobserved latent variable

Query Execution



Freebase

WIKIDATA

Execution Results (Answer)

```
1. AS 119
2. AA 3544 -> AS 1101
3. …
```

Weak supervision signal

Train a semantic parser using natural language query and the execution results (a.k.a. Semantic Parsing **with Execution**)

[Clarke *et al.*, 2010; Liang *et al.*, 2011]

Weakly-supervised Parsing as Reinforcement Learning

NL question

What is the most populous city in United States?

Semantic Parsing

Sampled Logical From
(Lambda DCS, Liang 2011)

z_1 **argmax(λx.city(x)∧located(x,US), λx.population(x))** ✔

z_2 argmax(λx.city(x), λx.population(x)) ✘

z_3 argmax(λx.city(x)∧loc(x,US), λx.GDP(x)) ✔

⋮

Query Execution

Answer
(with rewards)

y_1 New York ✔

y_2 Tokyo ✘

y_3 New York ✔

Optimize Objective

$$p(\boldsymbol{y}^* = \text{New York}) = p(\boldsymbol{y}_1|\boldsymbol{x}) + p(\boldsymbol{y}_3|\boldsymbol{x})$$

Gradient Updates

Learning Objective: Marginalizing Over Candidate Queries

What is the most populous city in United States?

Semantic Parsing

Reward

z_1 **argmax(λx.city(x)∧located(x,US), λx.population(x))** ✓

z_3 argmax(λx.city(x)∧loc(x,US), λx.GDP(x)) ✓

$$\nabla \log p_\theta(\underbrace{\boldsymbol{y}^*}_{\text{Gold Answer}}|\boldsymbol{x}) = \sum_{\boldsymbol{z}:\text{answer}(\boldsymbol{z})=\boldsymbol{y}^*} w(\boldsymbol{z},\boldsymbol{x}) \cdot \nabla \log p_\theta(\underbrace{\boldsymbol{z}}_{\text{Candidate Logical Form}}|\boldsymbol{x})$$

where $$w(\boldsymbol{z},\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{\sum_{\boldsymbol{z}':\text{answer}(\boldsymbol{z}'=\boldsymbol{y}^*)} p_\theta(\boldsymbol{z}'|\boldsymbol{x})}$$

- Intuitively, the gradient from each candidate logical form is weighted by its normalized probability. The more likely the query is, the higher its weight

Weakly-supervised Learning Issue 1: Spurious Logical Forms

- Spurious Queries: queries that have the correct execution result, but are semantically wrong

What is the most populous city in United States?

		Semantic Parsing	Reward
Correct	z_1	**argmax(λx.city(x)∧located(x,US), λx.population(x))**	✓
Spurious	z_3	argmax(λx.city(x)∧loc(x,US), λx.GDP(x))	✓

- Solutions:
  - Encourage diversity in gradient updates by updating different hypotheses with roughly equal weights (Guu *et al.*, 2017)
  - Use prior lexical knowledge to promote promising hypotheses. E.g., *populous* has strong association with `λx.population(x)` (Misra *et al.*, 2018)

Weakly-supervised Learning Issue 2: Search Space

- The space of possible logical forms with correct answers is exponentially large
- **Key Issue** logical forms are symbolic and indifferentiable
- How to search candidate logical forms more efficiently?

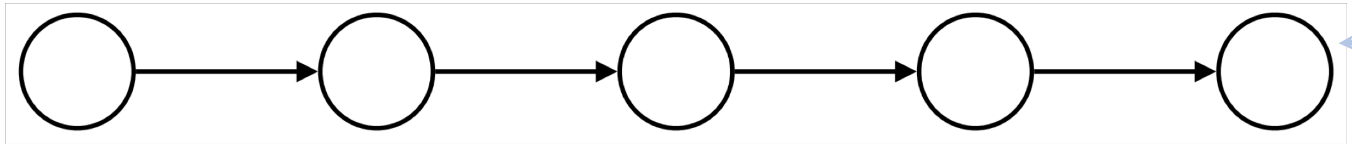
$$\nabla \log p_\theta(\boldsymbol{y}^*|\boldsymbol{x}) = \sum_{\boldsymbol{z}:\text{answer}(\boldsymbol{z})=\boldsymbol{y}^*} w(\boldsymbol{z}, \boldsymbol{x}) \cdot \nabla \log p_\theta(\boldsymbol{z}|\boldsymbol{x})$$

Prohibitively Large
Search Space

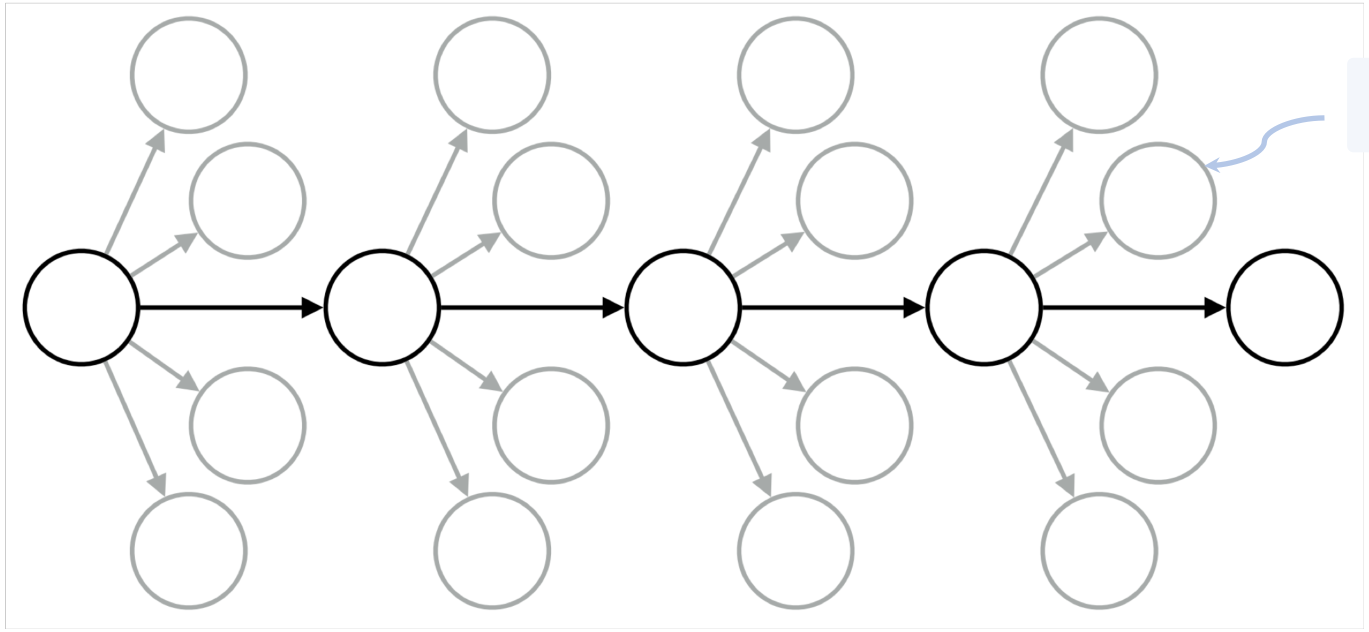
Efficient Search: Single Step Reward Observation

What is the most populous city in United States?

Reward=0



argmax λx.city(x) ∧ located(x,China) λx.population(x)

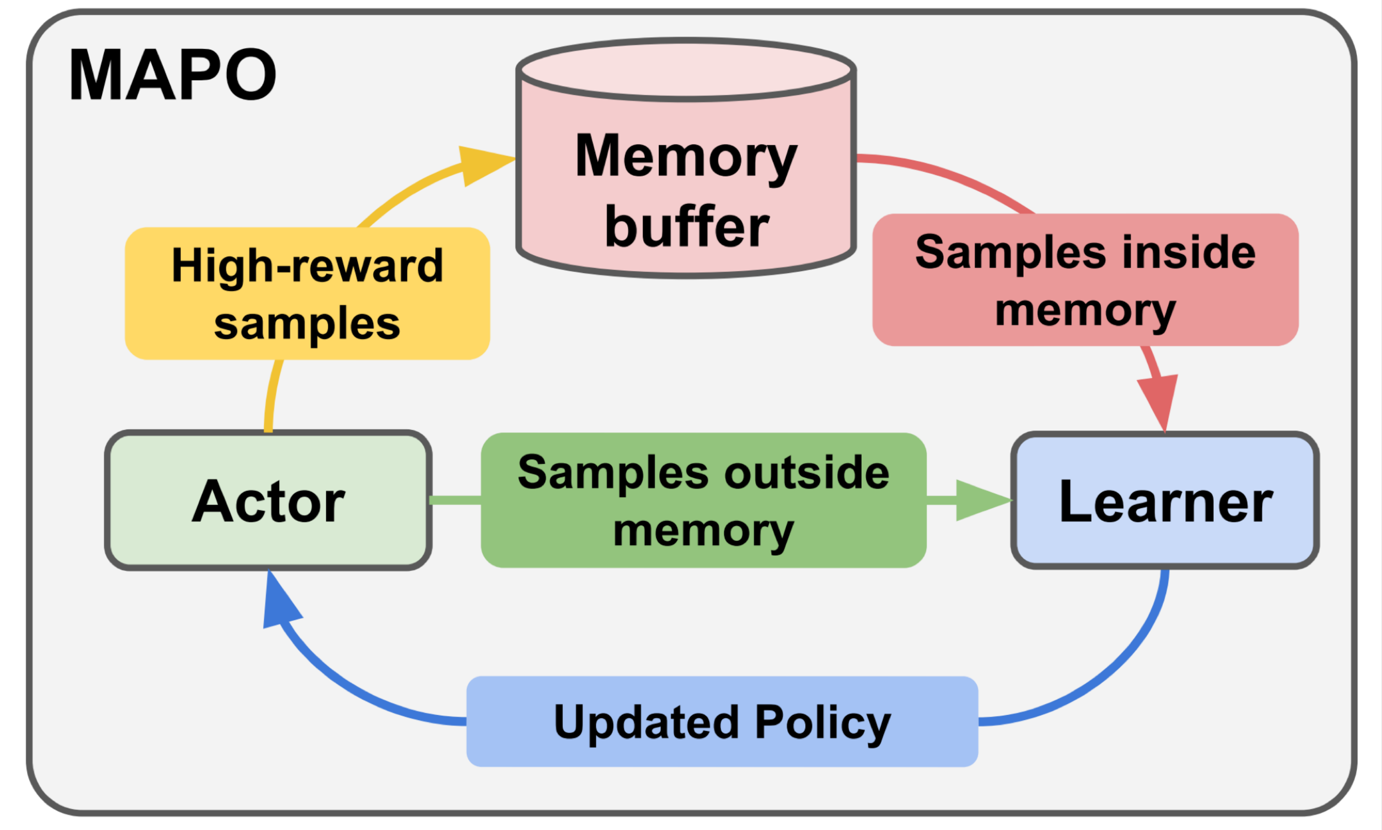


Reward=0

Factorize the reward into each single time step (a.k.a., reward shaping)

[Suhr and Artzi, 2018]

Efficient Search: Cache High-rewarding Queries



- Use a memory buffer to cache high-rewarding queries sampled so far
- During training, bias towards high-rewarding queries in the memory buffer

[Liang *et al.*, 2018]

Learning Paradigm 3: Semi-supervised Learning

Natural Language Query

Show me flights from Pittsburgh to Seattle

Labeled Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Unlabeled Natural Language Query

Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

As unobserved latent variable

Learning with
- Limited amounts of labeled natural language query and meaning representation
- Relatively large amounts of unlabeled natural language query

Learning with Labeled and Unlabeled Utterances

Limited Amount of Labeled Data

Sort my_list in descending order

```
sorted(my_list, reverse=True)
```

Copy the content of file 'file.txt' to file 'file2.txt'

```
shutil.copy('file.txt',
            'file2.txt')
```

Check if all elements in list `mylist` are the same

```
len(set(mylist)) == 1
```

+

Extra Unlabeled Utterances*

Get a list of words `words` of a file 'myfile'

Convert a list of integers into a single integer

Format a datetime object `when` to extract date only

Swap values in a tuple/list in list `mylist`

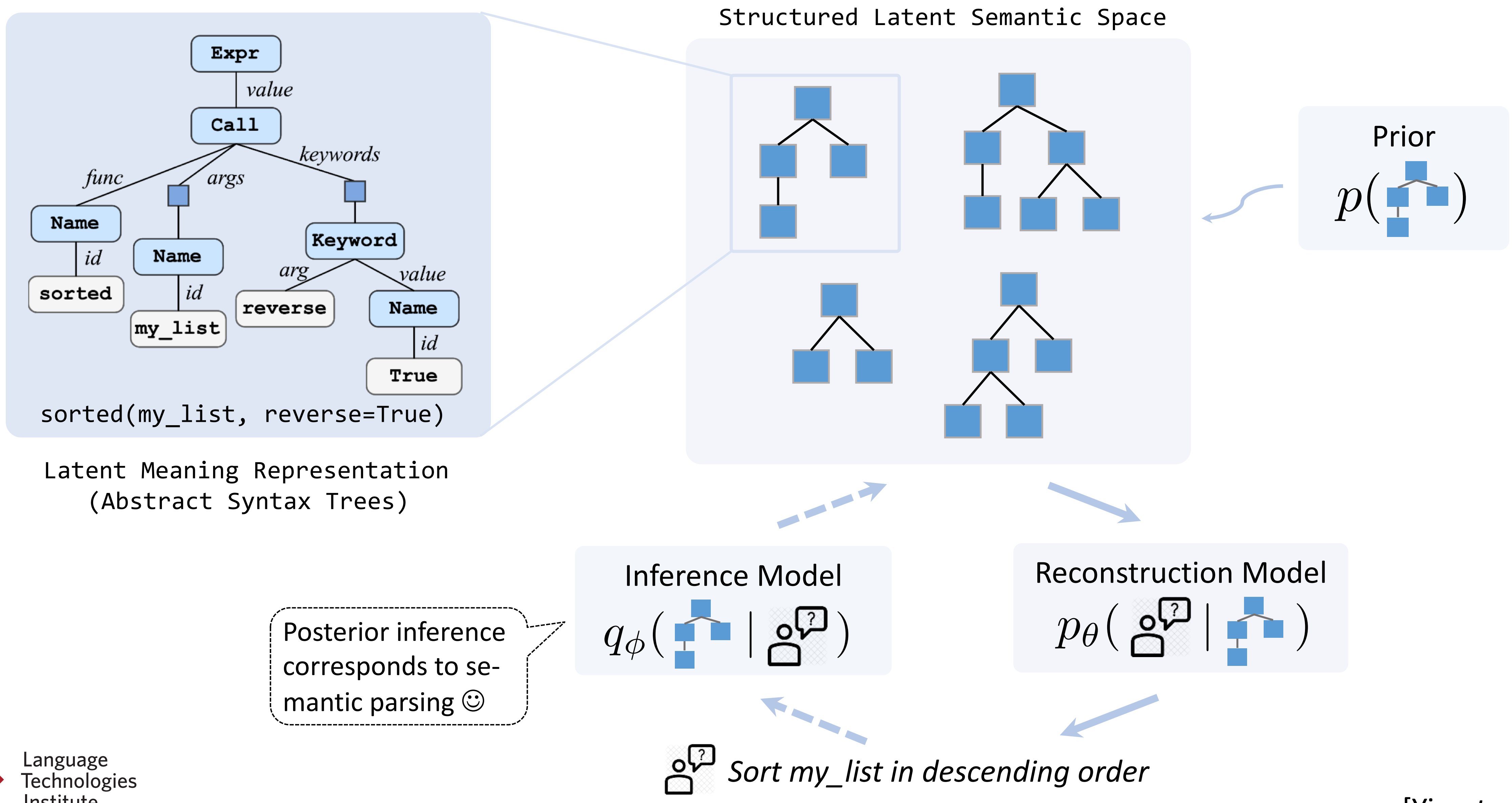
BeautifulSoup search string 'Elsie' inside tag 'a'

Convert string to lowercase

***Examples from** `conala-corpus.github.io`

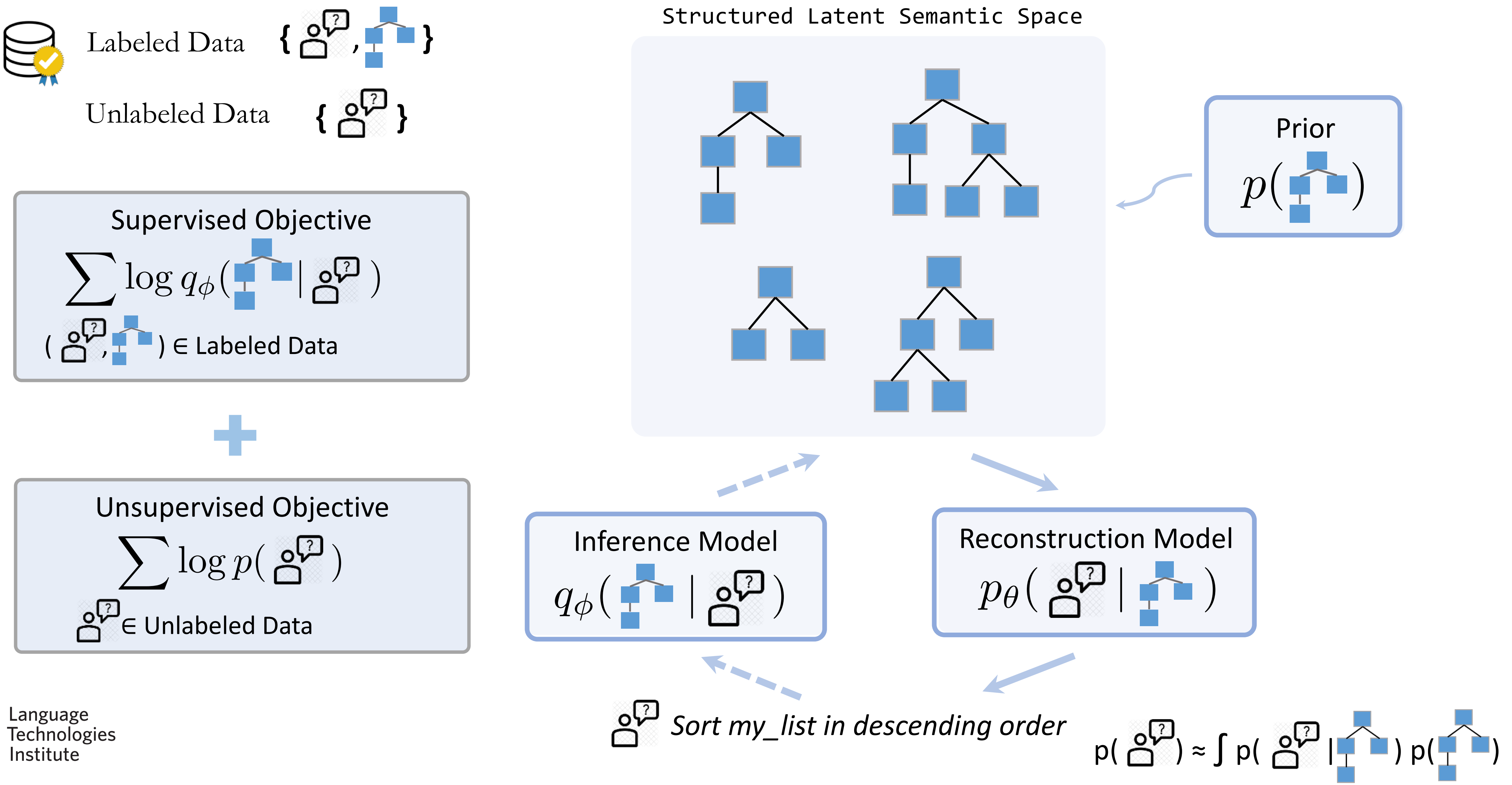
[Kočiský et al., 2016]

Programs as Tree-structured Latent Variables



[Yin *et al.*, 2018]

Semi-supervised Learning with STRUCTVAE



Conclusion 1: Pipeline of a Semantic Parser

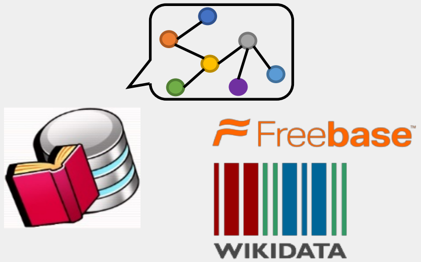
User's Natural Language Query

Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

Query Execution



Execution Results (Answer)

```
1. AS 119
2. AA 3544 -> AS 1101
3. …
```

Conclusion 2: Three Learning Paradigms

Supervised Learning

Utterances with Labeled Meaning Representation

Weakly-supervised Learning

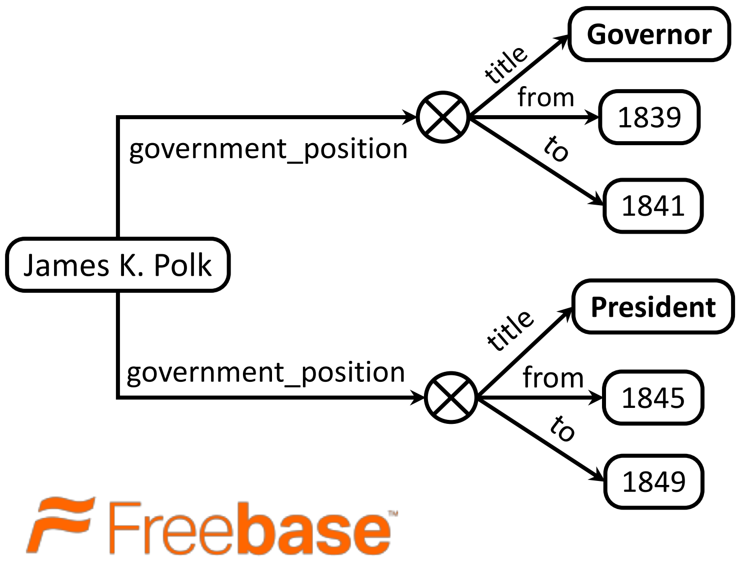
Utterances with Query Execution Results

Semi-supervised Learning

Learning with Labeled and Unlabeled Utterances

Challenge: Natural Language is Highly Compositional

Q: *what was James K. Polk before he was president?*




```
SELECT ?job_title.
FROM Freebase
WHERE{
    James K. Polk   government_position   ?job.
    ?job            title           ?job_title.

    ?job            to                ?to_date.

    FILTER(?to_date < (
        SELECT ?start_date.
        WHERE{
            James K. Polk government_position ?job1.
            ?job1         title              President.
            ?job1         from             ?start_date.
        }
    ))
}
```

Meaning Representation in SPARQL Query

- Sometimes even a short NL phrase/clause has complex structured grounding

[Yin et al., 2015]

Challenge: Scale to Open-domain Knowledge

- Most existing works focus on parsing natural language to queries to structured, curated knowledge bases
- Most of the world's knowledge has unstructured, textual form!
  - Machine Reading Comprehension tasks (e.g., SQUAD) use textual knowledge

User's Natural Language Query

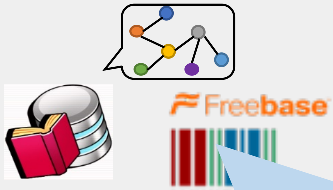
Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

```
lambda $0 e (and (flight $0)
     (from $0 san_Francisco:ci)
     (to $0 seattle:ci))
```

How to design MRs that can be used to query textual knowledge?

Query Execution

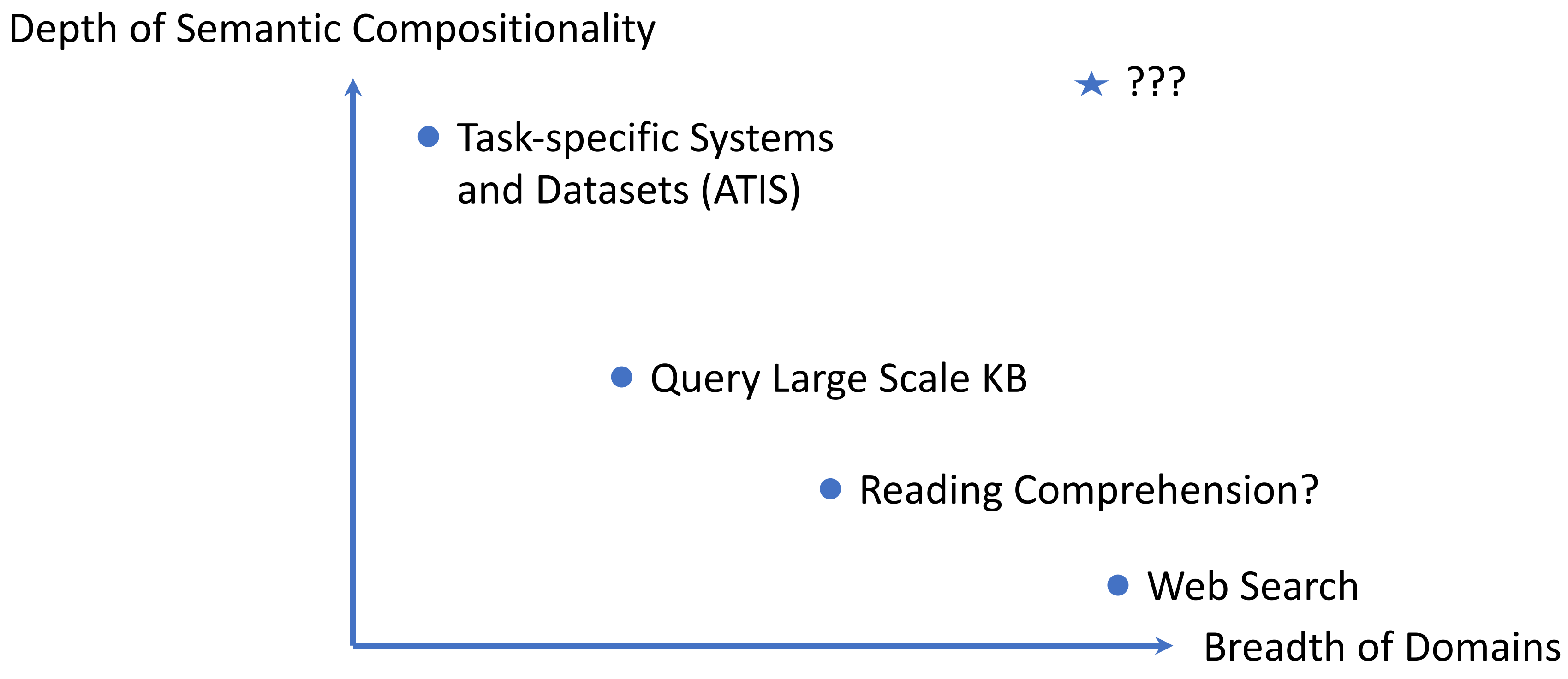


Textual Knowledge (e.g., Wikipedia Articles)

Execution Results (Answer)

```
1. AS 119
2. AA 3544 -> AS 1101
3. …
```

Final Notes: Challenges



(Figure taken from Pasupat and Liang, 2015)