Neural Semantic Parsing

Pengcheng Yin
pcyin@google.com
Google Research

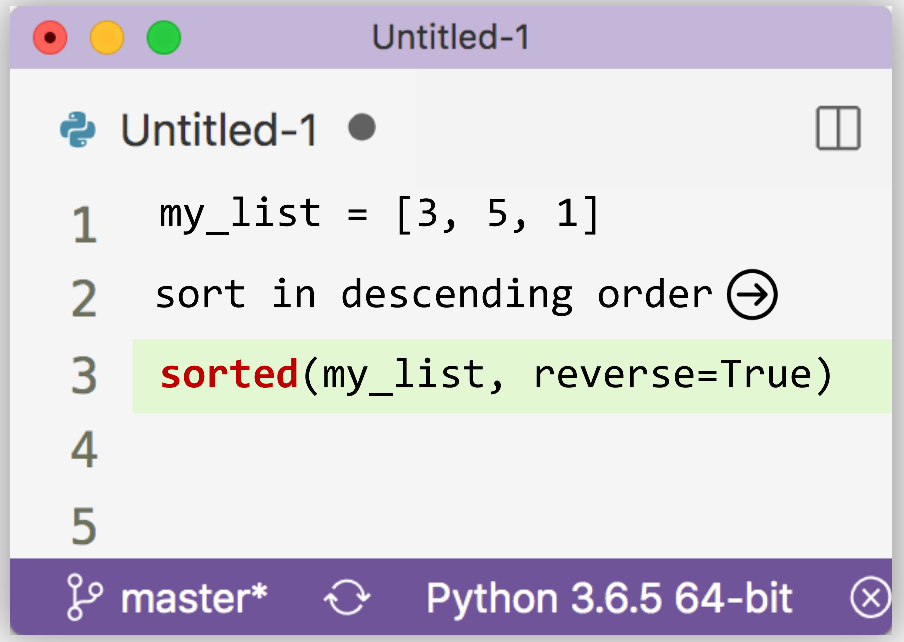
[Some contents are adapted from talks by Graham Neubig at CMU 11-747 NN for NLP]

Semantic Parsers: Natural Language Interfaces to Computers

Cortana

Hey Siri

alexa



Virtual Assistants

- *Set an alarm at 7 AM*
- *Remind me for the meeting at 5pm*
- *Play Jay Chou's latest album*

Natural Language Programming

- *Sort my_list in descending order*
- *Copy my_file to home folder*
- *Dump my_dict as a csv file output.csv*

The Semantic Parsing Task

Parsing natural language utterances into machine-executable meaning representations

Natural Language Utterance

Show me flights from Pittsburgh to Seattle

Meaning Representation

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

Machine-executable Meaning Representations

Translating a user's **natural language utterances** (e.g., queries) into machine-executable **formal meaning representations** (e.g., logical form, SQL, Python code)

Domain-Specific, Task-Oriented Languages (DSLs)

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`

General-Purpose Programming Languages

Sort my_list in descending order

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

`Python code generation`

Clarification about Meaning Representations (MRs)

Machine-executable MRs (our focus today) executable programs to accomplish a task

MRs for Semantic Annotation capture the semantics of natural language sentences

Machine-executable
Meaning Representations

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

Lambda Calculus

Python, SQL, …

Meaning Representations
For Semantic Annotation

The boy wants to go

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

Abstract Meaning Representation (AMR)

Abstract Meaning Representation (AMR),

Combinatory Categorical Grammar (CCG)

Workflow of a Semantic Parser

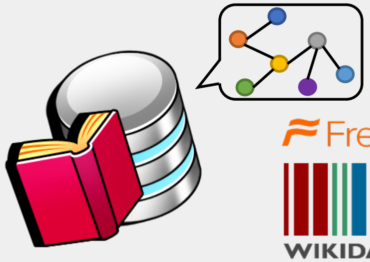
User's Natural Language Query

Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

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

Execute Programs against KBs



Execution Results (Answer)

```
1. Alaska Air 119
2. American 3544 -> Alaska 1101
3. …
```

Semantic Parsing Datasets

Domain-Specific, Task-Oriented Languages (DSLs)

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`

GeoQuery / ATIS / JOBs

WikiSQL / Spider

IFTTT

General-Purpose Programming Languages

Sort my_list in descending order

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

`Python code generation`

Django

HearthStone

CONCODE

CoNaLa

JuICe

GEO Query, ATIS, JOBS

- **GEO Query** 880 queries about US geographical information
- **ATIS** 5410 queries about flight booking and airport transportation
- **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'))))
```

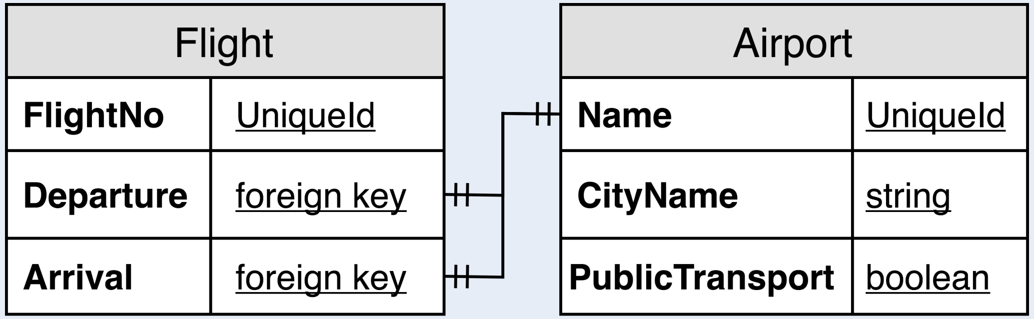
Prolog-style Program

Text-to-SQL Tasks

Natural Language Questions with Database Schema

Input Utterance

Show me flights from Pittsburgh to Seattle



SQL Query

```
SELECT Flight.FlightNo
FROM Flight
JOIN Airport as DepAirport
ON
   Flight.Departure == DepAirport.Name
JOIN Airport as ArvAirport
ON
   Flight.Arrival == ArvAirport.Name
WHERE
    DepAirport.CityName == Pittsburgh
    AND
    ArvAirport.CityName == Seattle
```

[Yu et al., 2018]

Semantic Parsing Datasets

Domain-Specific, Task-Oriented Languages (DSLs)

Show me flights from Pittsburgh to Berkeley

```
lambda $0 e (and (flight $0)
     (from $0 Pittsburgh:ci)
     (to $0 Berkeley:ci))
```

`lambda-calculus logical form`

GeoQuery / ATIS / JOBs

WikiSQL / Spider

IFTTT

General-Purpose Programming Languages

Sort my_list in descending order

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

`Python code generation`

Django

HearthStone

CONCODE

CoNaLa

Generating General-purpose Programs: HearthStone Dataset

Description properties/fields of a HearthStone card

Target code implementation as a Python class from HearthBreaker

Input (Card Description)

```
Name: Divine Favor
Cost: 3
Desc: Draw cards until you have as many as your opponent
```

Target Code (Python class)

```python
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]

Generating General-purpose Programs: CoNaLa

conala-corpus.github.io

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 Manually annotated, high quality natural language queries
- Programs reflect real-world usage of Python programmers
- Code is highly expressive and compositional
- The task is very hard. Exact match accuracy is low. Still evaluated as a text generation problem using fuzzy metrics like BLEU

[YLVN 2018; XJYVN 2020]

Generating General-purpose Programs: SIMPLE PYTHON CODE

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

- **Task:** Fill in the function body given function signature and doc-string
- Examples also includes unit-tests, making them directly executable
- Evaluation is done by checking if the model-predicted code could pass unit tests

[Chen et al., 2021; Austin et al., 2021]

Generating General-purpose Programs: Competition Problems

Problem

H-Index

Given a list of citations counts, where each citation is a nonnegative integer, write a function h_index that outputs the h-index. The h-index is the largest number *h* such that *h* papers have each least *h* citations.

Example:
Input: [3,0,6,1,4]
Output: 3

Generated Code

```
def h_index(counts):
  n = len(counts)
  if n > 0:
    counts.sort()
    counts.reverse()
    h = 0
    while (h < n and
          counts[h]-1>=h):
      h += 1
    return h
  else:
    return 0
```

Test Cases

```
Input:
[1,4,1,4,2,1,3,5,6]

Generated Code Output:
4 ✓

Input:
[1000,500,500,250,100,
100,100,100,100,75,50,
30,20,15,15,10,5,2,1]

Generated Code Output:
15 ✓
```

- Coding competition problems provide a large-scale testbed for code generation

[Example from Hendrycks et al. 2021. More in Alet et al., 2021; Zavershynskyi et al., 2018]

Supervised Learning of Semantic Parsers

User's Natural Language Query

Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

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

Train a neural semantic parser with source natural language utterances and target programs

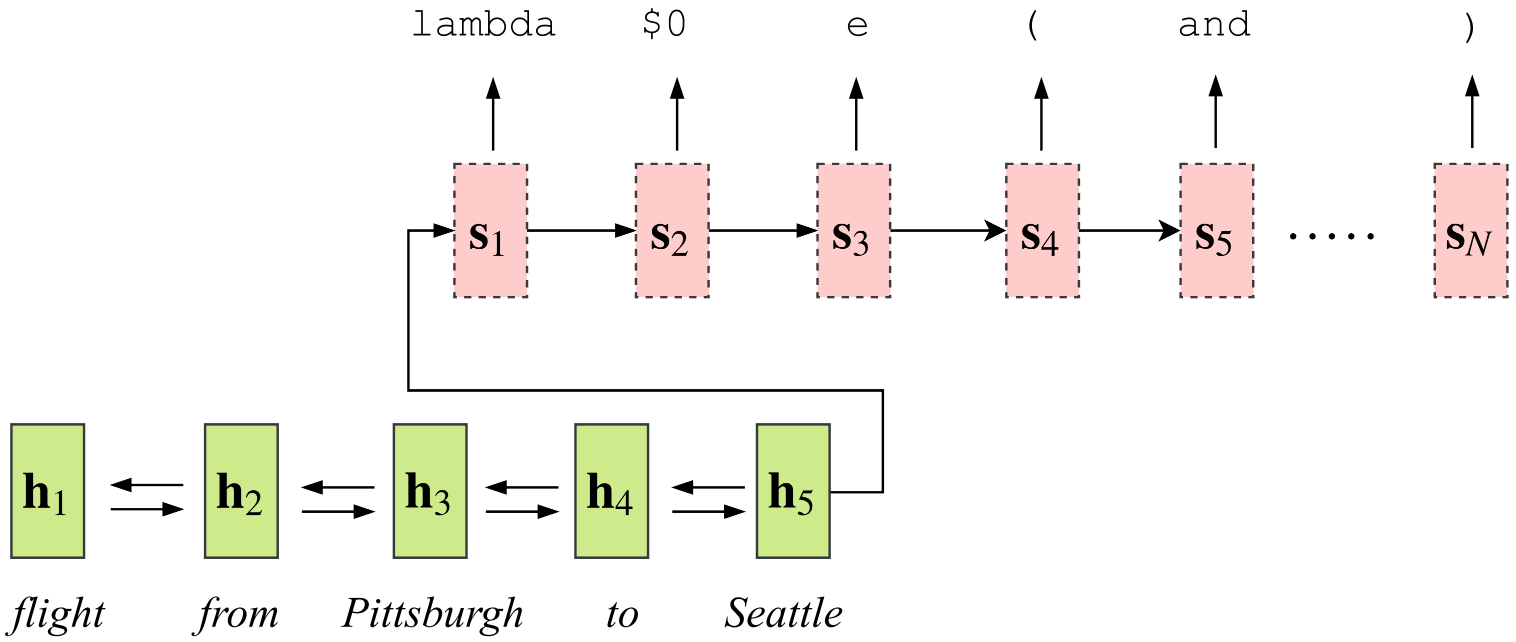
Semantic Parsing as Sequence-to-Sequence Transduction

Task-Specific Meaning Representations

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



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

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

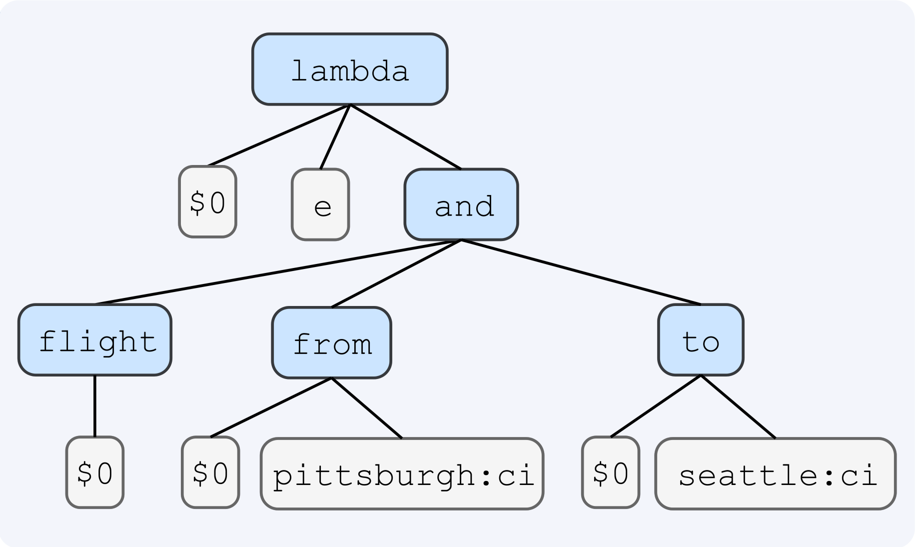
Meaning Representations have Strong Structures

Semantic Parsing

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



Tree-structured Representation

[Dong and Lapata, 2016]

Issues with Predicting Linearized Programs

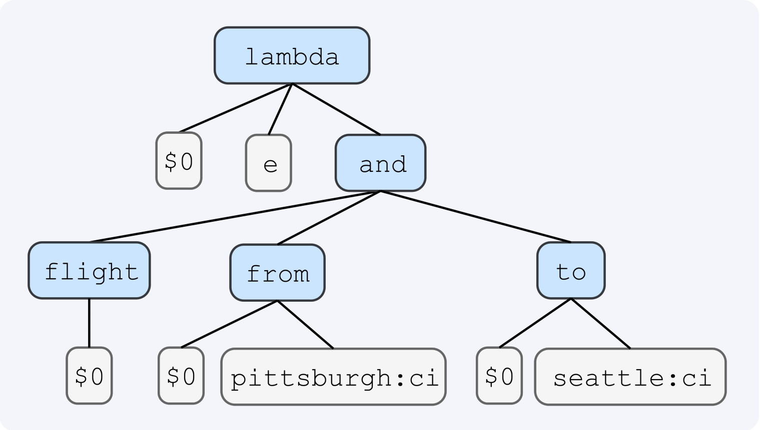
- **Meaning Representations** (e.g., a database query) have strong underlying structures.
- **Issue** Using vanilla seq2seq models ignore the rich structures of meaning representations, and could generate invalid outputs that are not trees

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

[Xiao et al., 2016; Jia and Liang, 2016; Dong and Lapata, 2016]

Core Research Question for Better Models

How to add inductive biases to networks to better capture the **program structures**?

Encode Utterance and In-Domain Knowledge Schema

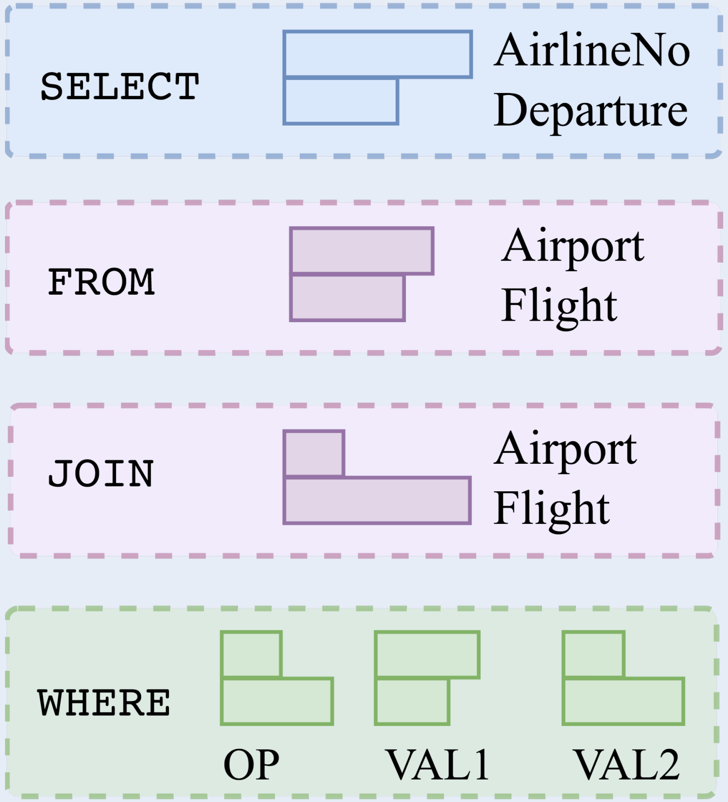
Input Utterance

Show me flights from Pittsburgh to Berkeley

Flight	
FlightNo	UniqueId
Departure	foreign key
Arrival	foreign key

Airport	
Name	UniqueId
CityName	string
PublicTransport	boolean

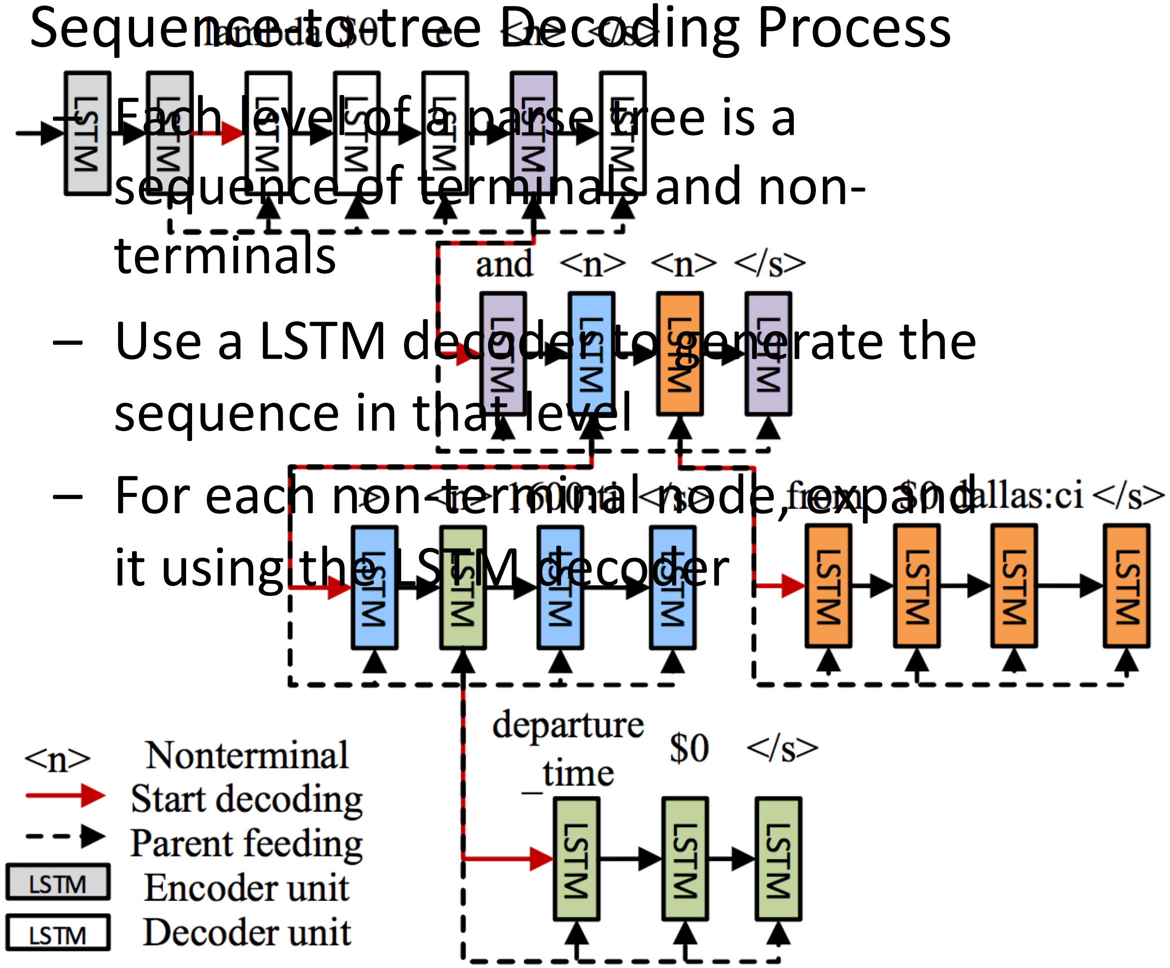
Predict Programs Following Task-Specific Program Structures

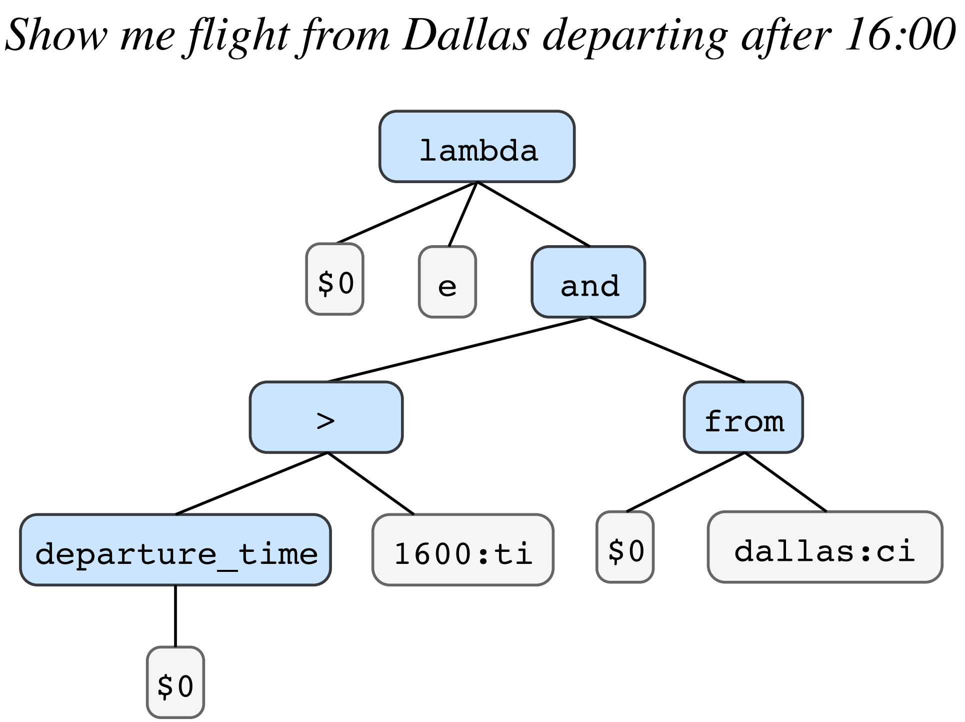


[Xu et al., 2017; Yu et al., 2018]

Structure-aware Decoding for Semantic Parsing

- **Seq2Tree** Generate the parse tree of a program using a hierarchy of recurrent neural decoders following the tree structure
- 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 in that level
  - For each non-terminal node, expand it using the LSTM decoder

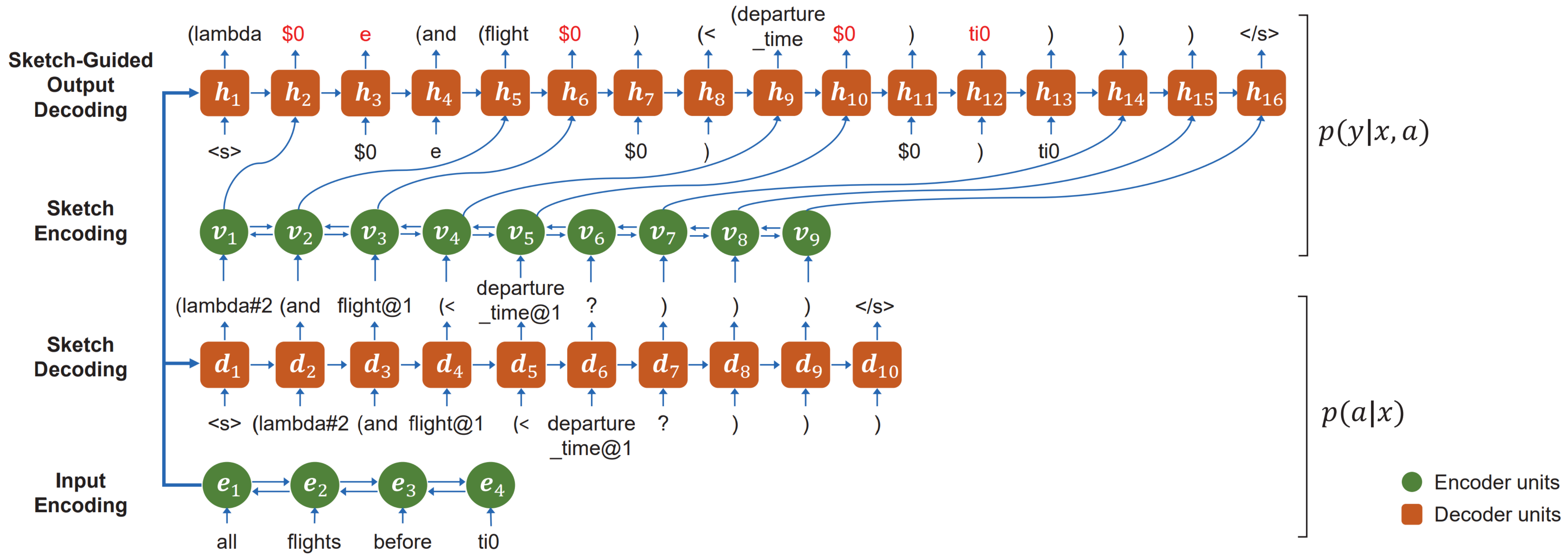




[Dong and Lapata, 2016]

Structure-aware Decoding: Model Decomposition in Programming

- **Coarse-to-Fine Decoding** decodes 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 a **coarse global structure** of the logical form, and use it to guide the generation of the **fine-grained structure**



[Dong and Lapata, 2018]

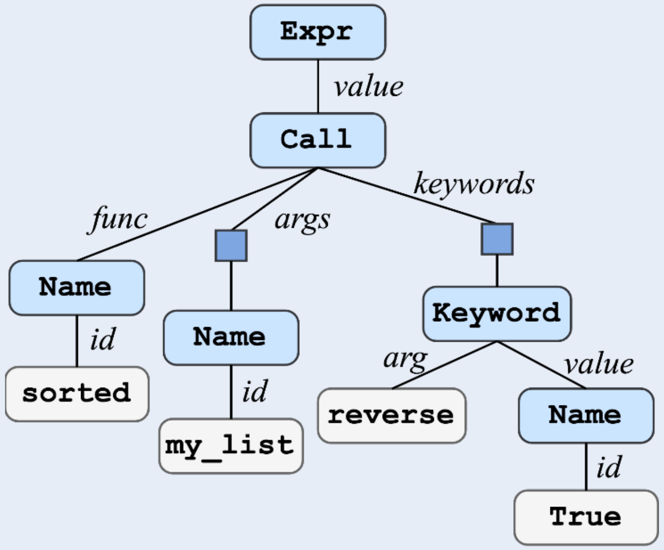
Grammar-constrained Decoding

- Previously introduced methods could generate tree-structured representations, but cannot guarantee they are grammatically correct.
- Meaning representations (e.g., Python) have strong underlying grammar/syntax
- How can we explicitly leverage the grammar of programs for better generation?

Python Abstract Grammar

```
Expr ⟼ Name | Call

Call ⟼ expr[func] expr*[args]
keyword*[kwargs]

  If ⟼ expr[test] stmt*[body] stmt*[orelse]

 For ⟼ expr[target] expr*[iter]
                stmt*[body] stmt*[orelse]

FunctionDef ⟼ identifier[name] expr*[iter]
              stmt*[body]  stmt*[orelse]
```

Abstract Syntax Tree (AST)



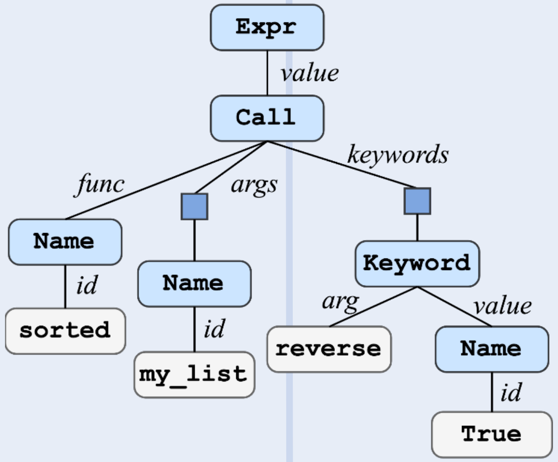
`sorted(my_list, reverse=True)`

General-purpose Syntax-driven Program Generation

Natural Language Intent

Sort my_list in descending order

Abstract Syntax Tree (AST)



Python Source Code

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

– Use Abstract Syntax Trees as general-purpose intermediate meaning representations

– $p_\theta(\ \ |\ \ )$ is a seq-to-tree model using program grammar as prior syntactic knowledge to guide neural decoding

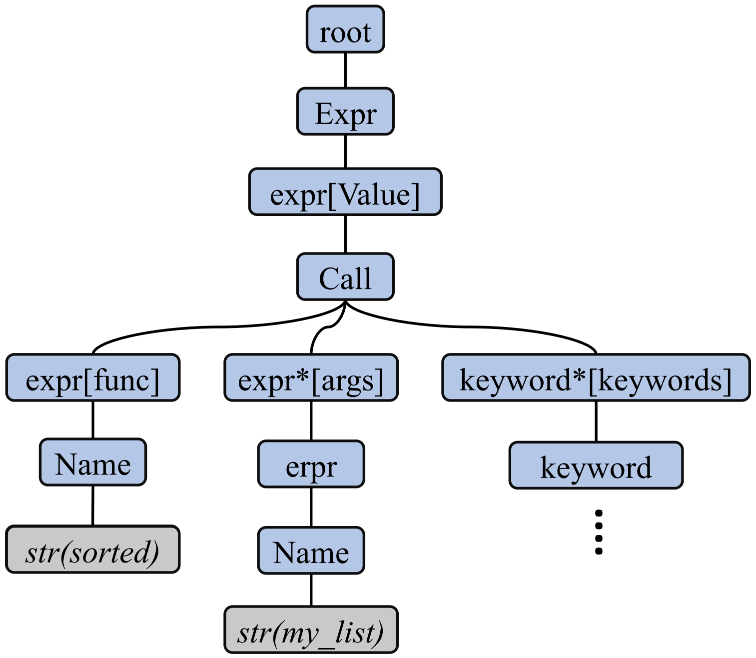
– Deterministic transformation to source code

[Yin and Neubig 2017]

$p_\theta(\;|\;)$: AST Generation using Auto-regressive Models

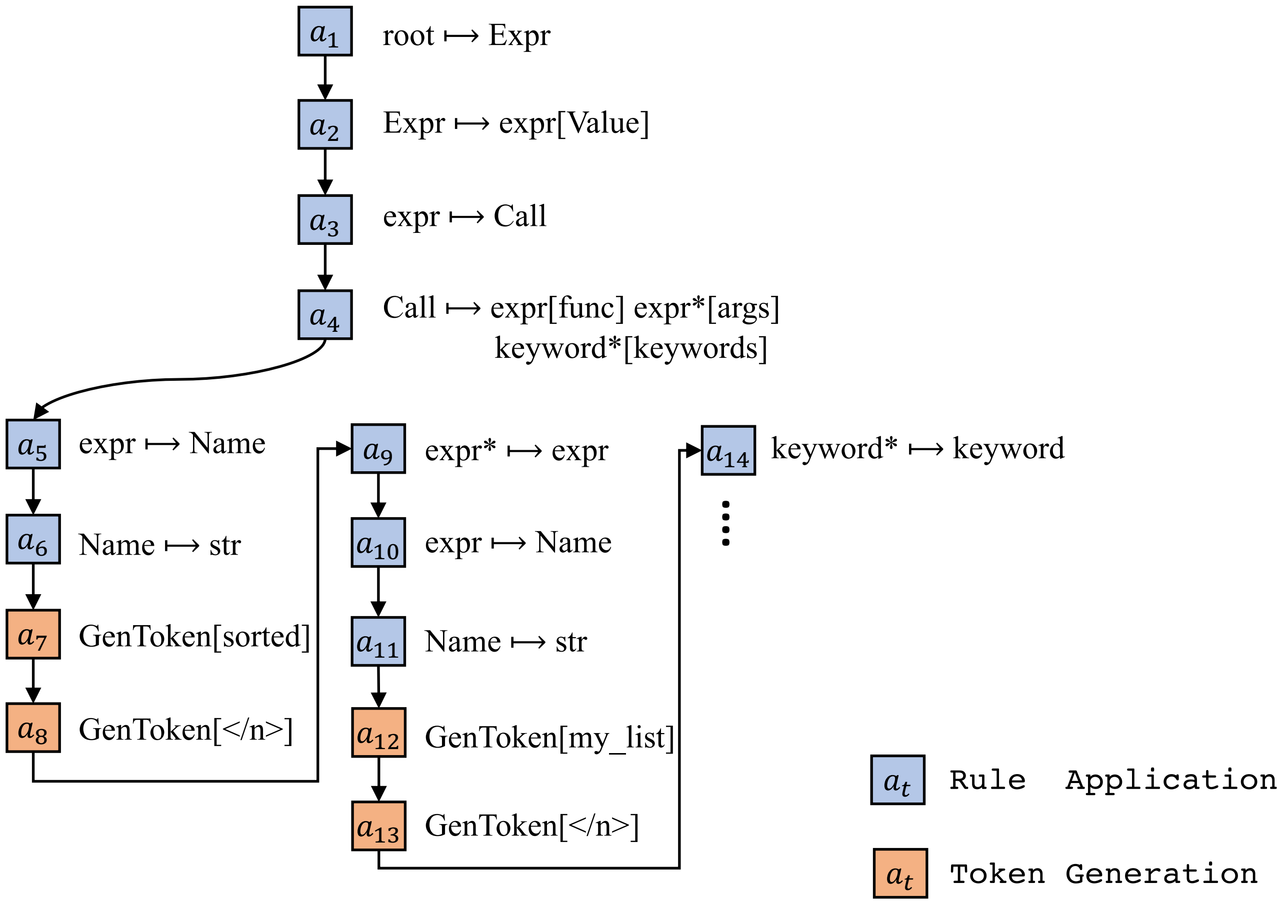
Factorize the generation story of an AST into sequential application of *actions* $\{a_t\}$:

Derivation Abstract Syntax Tree




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

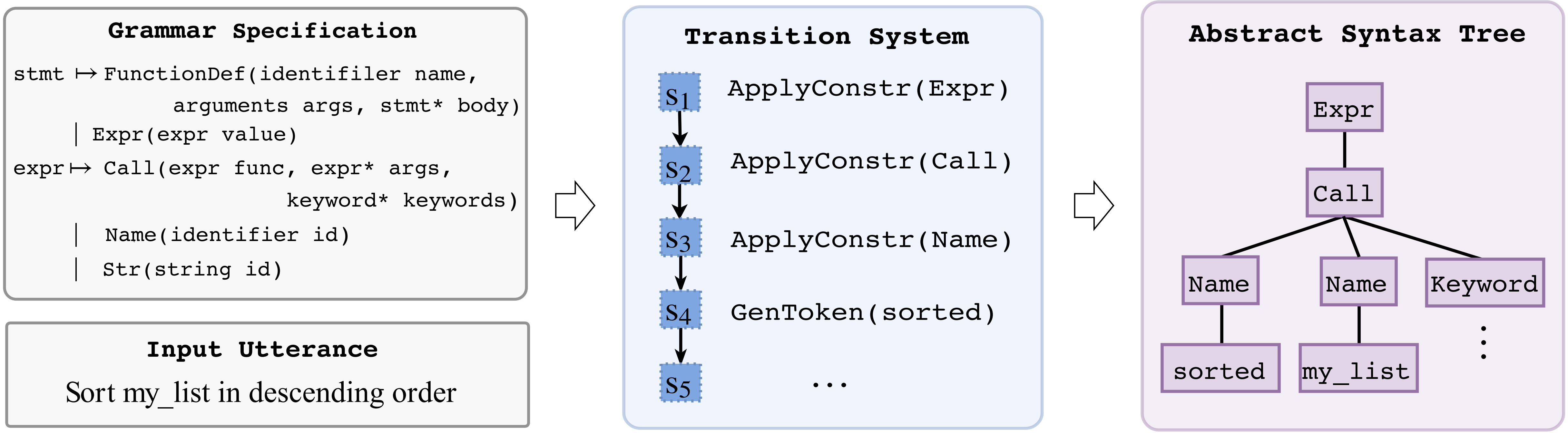
Action Sequence



[Yin and Neubig 2017]

TranX: Transition-based Abstract SyntaX Parser

- Convenient interface to specify task-dependent grammar in plain text
- Customizable conversion from abstract syntax trees to domain-specific programs
- Built-in support for many languages: Python, SQL, Lambda Calculus, Prolog…

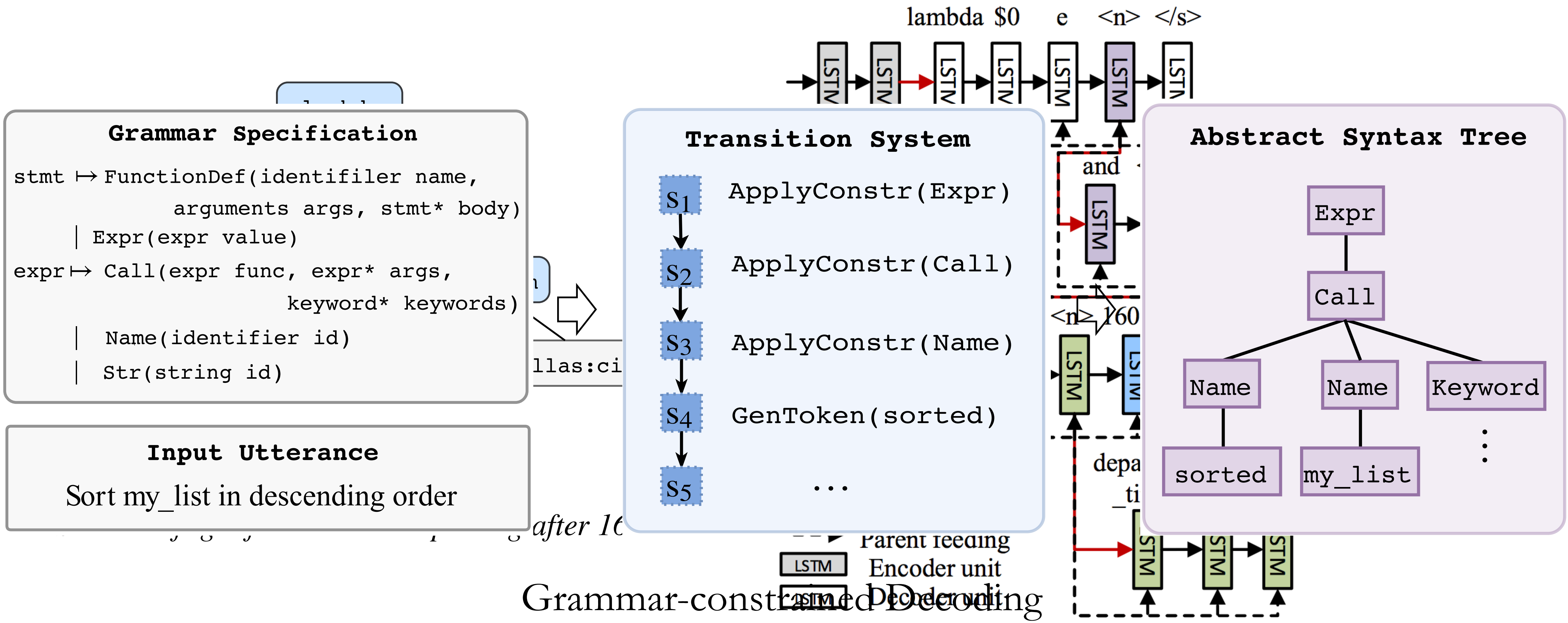


github.com/pcyin/tranX

[Yin and Neubig 2018, Yin and Neubig 2019]

Summary: Supervised Learning of Semantic Parsers

Key Research Question: constrain the output space following the structure of programs

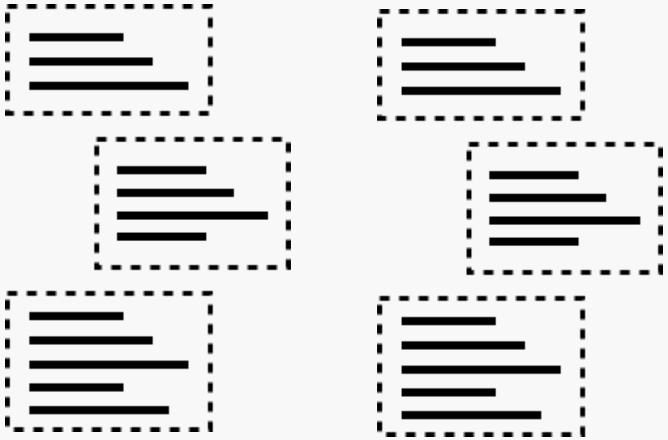


Grammar-constrained Decoding

Structure-aware Decoding

Wait…. Do We Really Need Program Structures?

Large-scale Code Repos from GitHub



Open AI Codex (Code-finetuned GPT-3)

Code Generation as a Prompting Task

Prompt:

```
def find_k_largest(arr, k):
  """"return the k largest elements
  in the input array.""""
```

Model Prediction:

```
  result = sorted(
            arr, reverse=True)[:k]
  return result
```

- **Open AI Codex**: Large-scale transformer language model pre-trained on 170+ GB of GitHub code data

[Chen et al., 2021; Austin et al., 2021]

Impressive Performance of Large Code Language Models

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Solves nearly 50% such simple coding programs using only 10 samples

Do We Really Need Program Structures?

- Perhaps…. not? Large language models trained on code seldom make syntax errors
- However, language models treat code as plain text without modeling its rich semantics (e.g., variables defined in the context).
- We hope to language models could capture program semantics with large-scale data, but this is not very sample efficient.

```
int foo() {
  x.write();  Undefined variable.
  x.flush();
}
```

```
int foo() {
  String x = “Hello”;
  return x;  Return type violation
}
```

Large language models are still prune to such semantic errors.

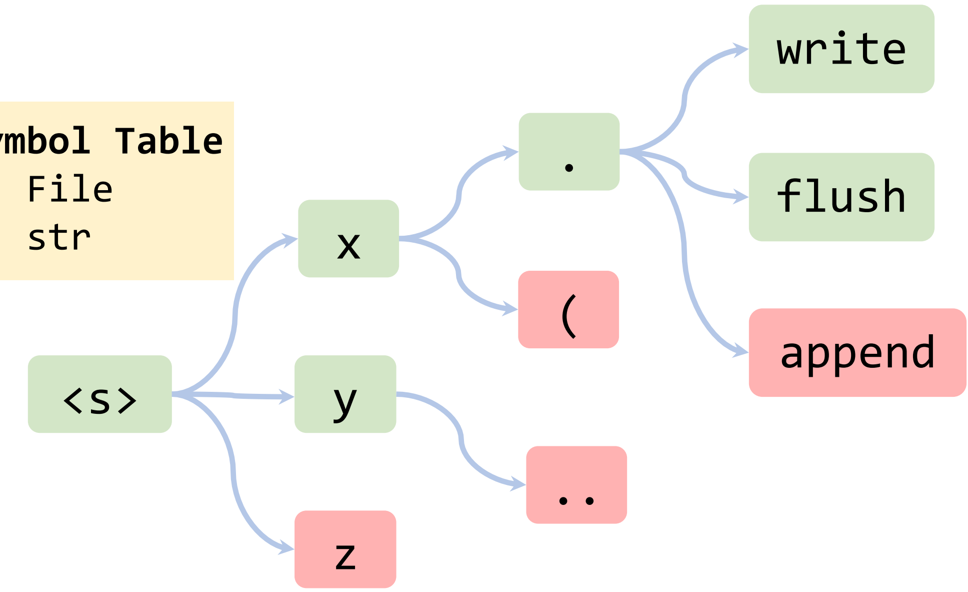
[Mukherjee et al., 2021]

How Constrained-decoding can be Useful?

- Instead of constraining the generation based on the syntactic grammar, we constrain generation using the semantic properties of the programmatic context
  - Semantic properties can be inferred using a static program analyzer
  - We can augment the **neural** model with information from the **symbolic** analyzer

```
int foo() {
  x.write();  Undefined variable.
  x.flush();
}
```

```
Symbol Table
x: File
y: str
```

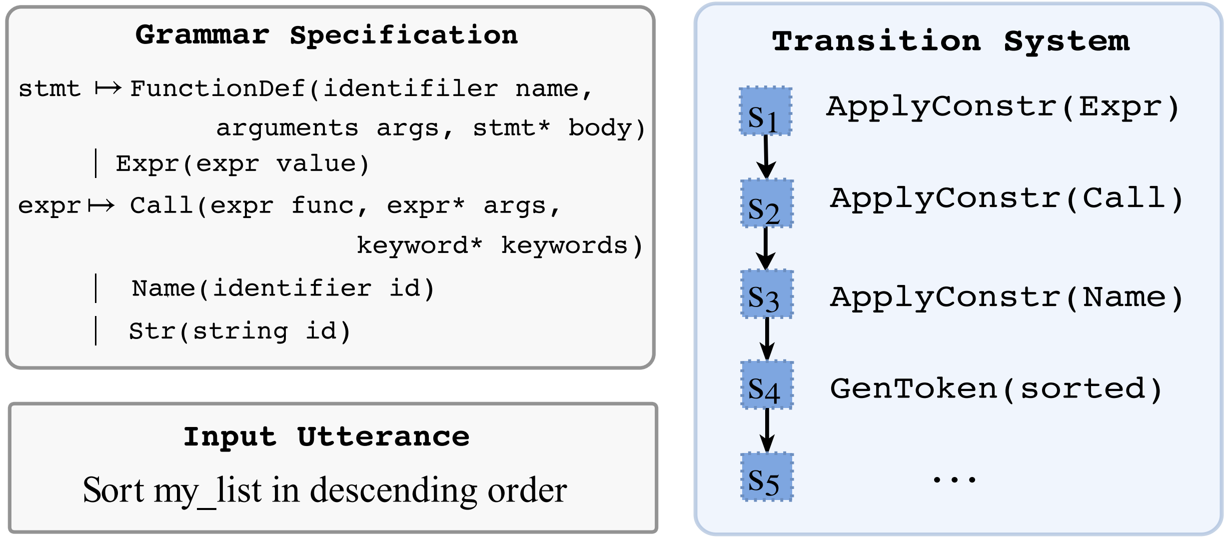


Rule out invalid predictions using typing information in symbol table

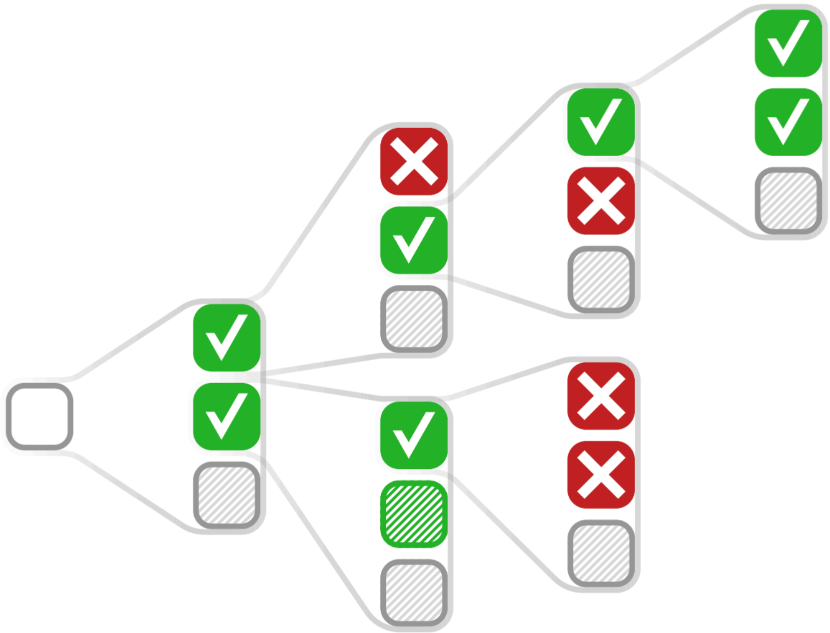
[Examples adapted from Mukherjee et al., 2021]

Brief Summary

- **Grammar-constrained decoding** was quite popular for semantic parsing and code generation before pre-trained language models.
- With **pre-trained language models** on code, constrained decoding using semantic information of programs could still be useful to control and decoding process and prune invalid predictions.



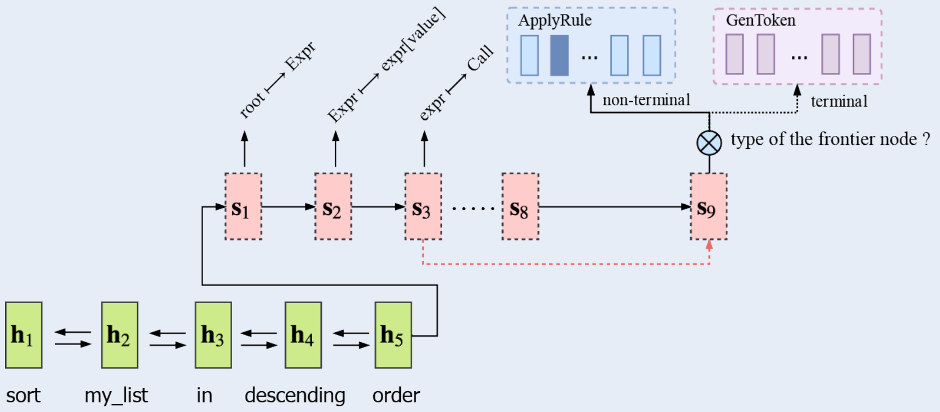
Grammar-constrained Decoding



Generalized-constrained Decoding
[Scholak et al., 2021; Mukherjee et al., 2021]

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

Weakly-supervised Learning of Semantic Parsers

User's Natural Language Query

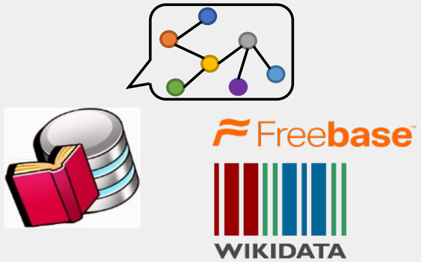
Show me flights from Pittsburgh to Seattle

Parsing to Meaning Representation

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

As unobserved latent variable

Query Execution



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

Weakly Supervised Semantic Parsing

What is the most populous city in United States?

City	Country	Population	GDP
New York	USA	8.62M	1275B
Hong Kong	China	7.39M	341.4B
Tokyo	Japan	9.27M	1800B
London	UK	8.78M	650B
Los Angeles	USA	4.00M	941B

`Answer: New York`

Hypothesized Programs

```
City.OrderBy(Population)
    .First() => Result: Tokyo
```

```
City.Filter(Country=='USA')
    .OrderBy(Population)
    .First() => Result: New York
```

```
City.Filter(Country=='USA')
    .OrderBy(GDP)
    .First() => Result: New York
```

Weakly-supervised Learning -- Challenges

Hypothesized Programs

```
City.OrderBy(Population)
     .First() => Result: Tokyo
```

```
City.Filter(Country==‘USA’)
     .OrderBy(Population)
     .First() => Result: New York
```

```
City.Filter(Country==‘USA’)
     .OrderBy(GDP)
     .First() => Result: New York
```

Large Search Space

Exponentially large search space w.r.t. the size of programs

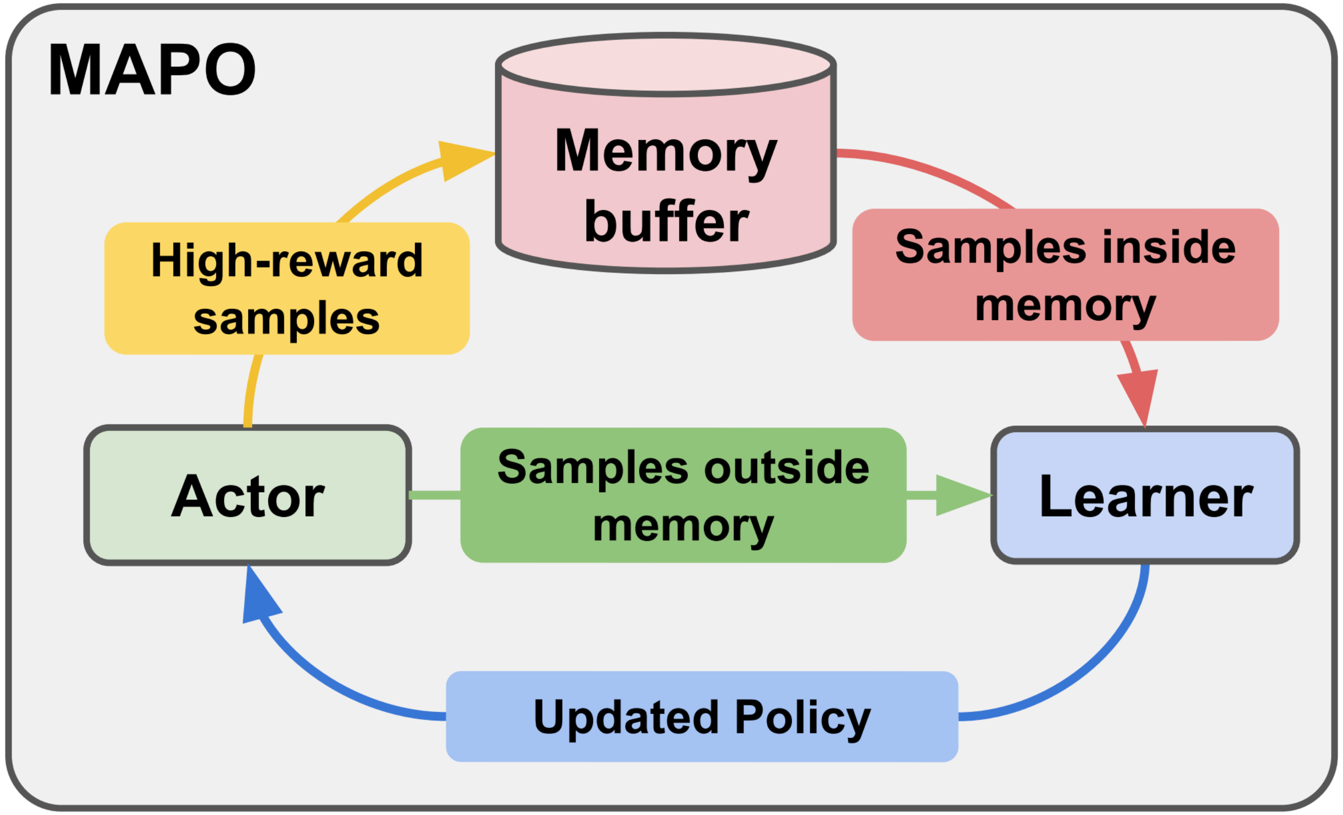
Very Sparse Rewards

Only very few programs are actually correct

Spurious Programs

Spurious programs could also hit the correct answer, leading to noisy reward signals.

Efficient Search: Cache High-reward Programs



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

[Liang et al., 2018]

Few-shot Learning: use a few annotated examples

```
Translate questions into programs:

Question 1: What is the capital of UK.
Program 1: Country.Filter(Name=‘UK’).Select(Capital)

Question 2: What is the most populous city in United States?
Program 2: City.Filter(Country==‘USA’).OrderBy(Population).First()

Question 3: Show me countries with the highest GDP?
Program 3: Country.OrderBy(GDP).First()     Language model prediction
```

- Semantic Parsing as a prompting task using large language models.
- Use a handful amount of demonstration examples in the prompt, ask the model to generate a continuation.

[Shin et al., 2021; Shin et al., 2022]

Few-shot Learning: Represent Programs as Canonical Sentences

```
Translate questions into programs:

Question 1: What is the capital of UK.
Solution: From Country, get rows whose name is ‘UK’, select Capital

Question 2: What is the most populous city in United States?
Solution: From City, get rows whose country is USA, rank by
Population, get first row

Question 3: Show me countries with the highest GDP?
Solution: From Country, rank by GDP, get first row.
```

- Better prompt a language model using pseudo-English sentences.
- We can perform constrained decoding on those canonical utterances.

[Wang et al., 2015; Shin et al., 2021; Shin et al., 2022]

Zero-shot Learning: Learning using only synthetic Data

Synchronous Context Free Grammar based on Domain Schema

$Entities ↦ $property of $Entity   GetProperty($Entity , $property)

$PrepNP ↦ $prep $Entity $Entities ↦ $EntType ($PrepNP | $Compl.)

$Compl. ↦ that has the largest $property $Entities ↦ $Entities and $PrepNP

Synthetic Canonical Examples

Citation count of Alan Turing

`GetProperty(alan_turing, citation_num)`

Most recent paper in deep learning

Paper that has the largest publication year and in deep learning

Iterative Paraphrasing by Paraphrase Generation Model

Allan Turing's citations

- *The citations of Allan Turing*
- *How many citations did Allan Turing get?*

How many citations does Allan Turing have?

Most recent research in deep learning

- *Recent research on deep learning*
- *What's the latest deep learning paper?*

What is the latest deep learning study?

- *Latest studies on deep learning*

What's new in deep learning?

Paper in deep learning and the biggest year of publication

What is the biggest year for publishing deep learning?

[Yin *et al.*, 2021; Wang et al. 2015; Xu et al., 2019; Xu et al., 2020]

Conclusion: Workflow of a Semantic Parser

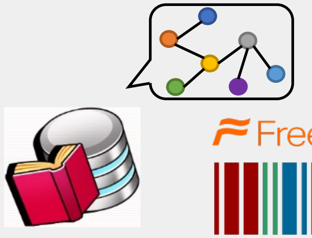
User's Natural Language Query

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



Freebase

WIKIDATA

Execution Results (Answer)

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

Conclusion: Learning Paradigms

Supervised Semantic Parsing

What is the most populous city in United States?

```
City.Filter(Country==‘USA’)
    .OrderBy(Population)
    .First() => Result: New York
```

Weakly Supervised Semantic Parsing

What is the most populous city in United States?

City	Country	Population	GDP
New York	USA	8.62M	1275B
Hong Kong	China	7.39M	341.4B
Tokyo	Japan	9.27M	1800B

`Answer: New York`

Tree-based Decoding

Grammar-constrained Decoding

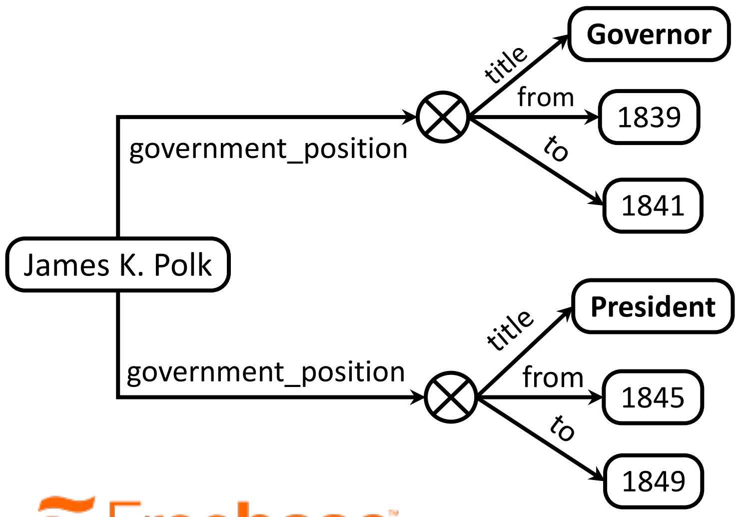
Few/zero-shot Training

Efficient Exploration over Large Search Space

Tackle Spurious Programs

Challenge: Natural Language is Highly Compositional

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



Freebase

```
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

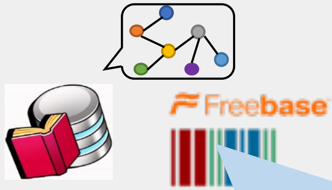
Show me flights from Pittsburgh to Seattle

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

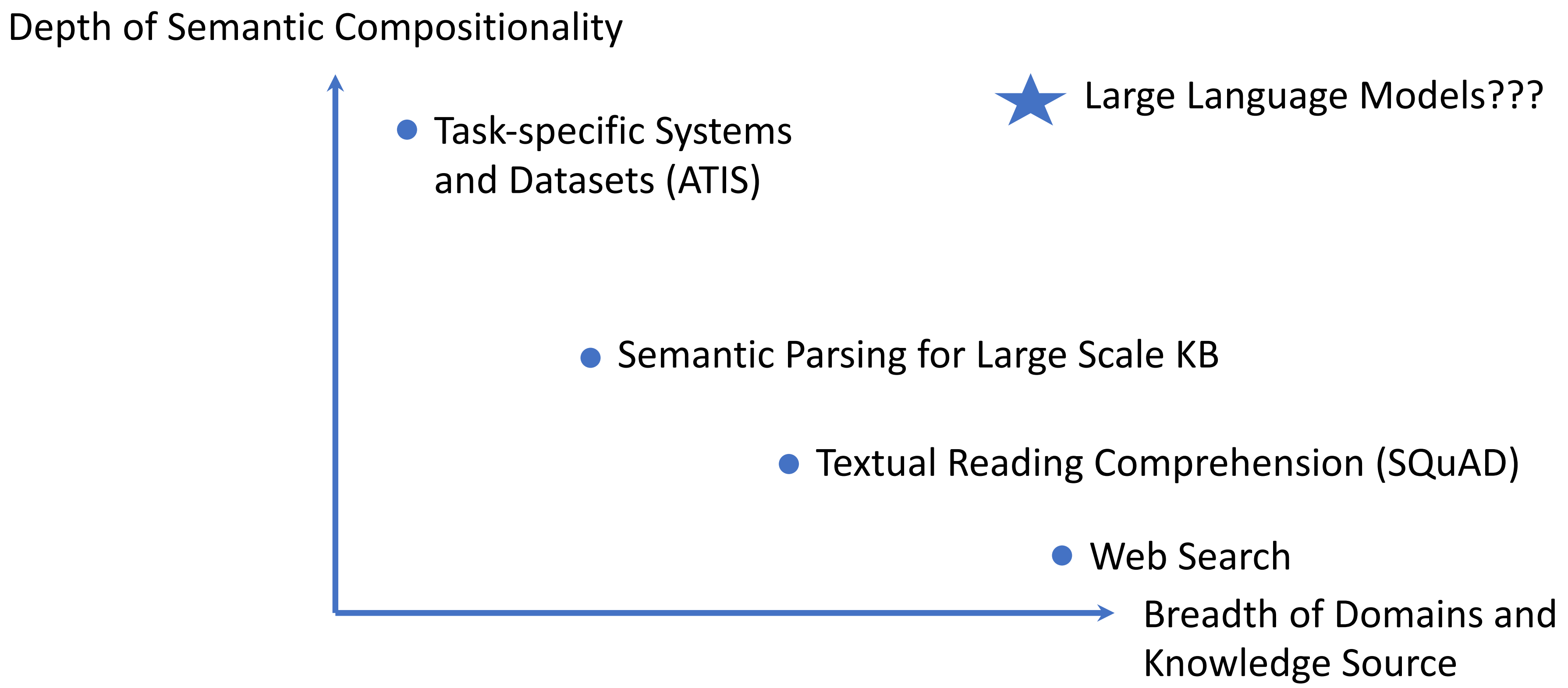


Textual Knowledge (e.g., Wikipedia Articles)

Execution Results (Answer)

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

Final Notes: Challenges



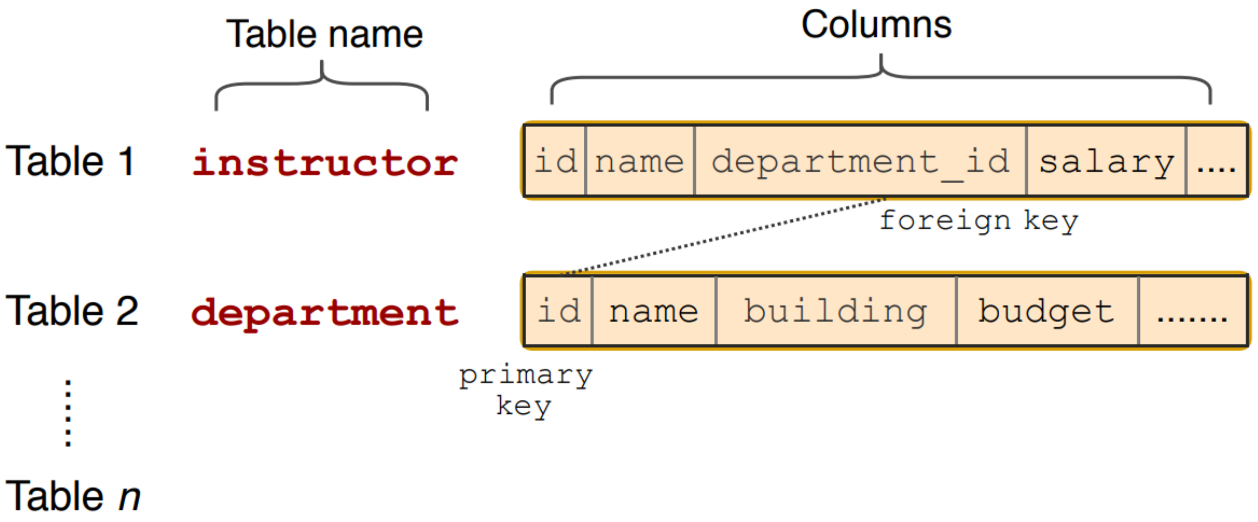
[Figure adapted from Pasupat and Liang, 2015]

Supplementary Slides

More Semantic Parsing Datasets

Spider

Annotators check database schema (e.g., database: `college`)



Annotators create:

Complex question What are the name and budget of the departments with average instructor salary greater than the overall average?

Complex SQL

```
SELECT T2.name, T2.budget
FROM instructor as T1 JOIN department as
T2 ON T1.department_id = T2.id
GROUP BY T1.department_id
HAVING avg(T1.salary) >
    (SELECT avg(salary) FROM instructor)
```

- Examples from 200 databases
- Target SQL queries involve joining fields over multiple tables
- Non-trivial Compositionality
  - Nested queries
  - Set Union
  - …

https://yale-lily.github.io

[Yu et al., 2018]

WikiSQL Dataset

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

- 80,654 examples of Table, Question, SQL Query and Answer
- **Context** a small, single database table extracted from a Wikipedia article
- **Target** an SQL query

[Zhong et al., 2017]

HearthStone (HS) Card Dataset

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

Utterance (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]

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

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]

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

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

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

[Oda *et al.*, 2015]

Notes for Weakly Supervised Parsing

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

Probability of Gold Answer

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

Gradient Updates

Maximum Marginal Likelihood Training Objective

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)) ✓

Marginalization over all (sampled) hypotheses

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

Gold Answer

Candidate Logical Form (Latent Variable)

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 logical form is, the higher the weight of its gradient

Weakly-supervised Learning Issue 1: Spurious Logical Forms

- **Spurious Logical Forms** 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 gradient 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)

Tackle Spurious Programs using Heuristics

What is the most populous city in United States?

```
City.Filter(Country==‘USA’)
    .OrderBy(Population)
    .First() => Result: New York
```

```
City.Filter(Country==‘USA’)
    .OrderBy(GDP)
    .First() => Result: New York
```

Similarity(‘populous’, population)

Back-translation-score $p(\ |\ )$

Similarity(‘populous’, GDP)

Back-translation-score $p(\ |\ )$

[Guu et al., 2017; Misra et al., 2018; Cheng et al., 2018]

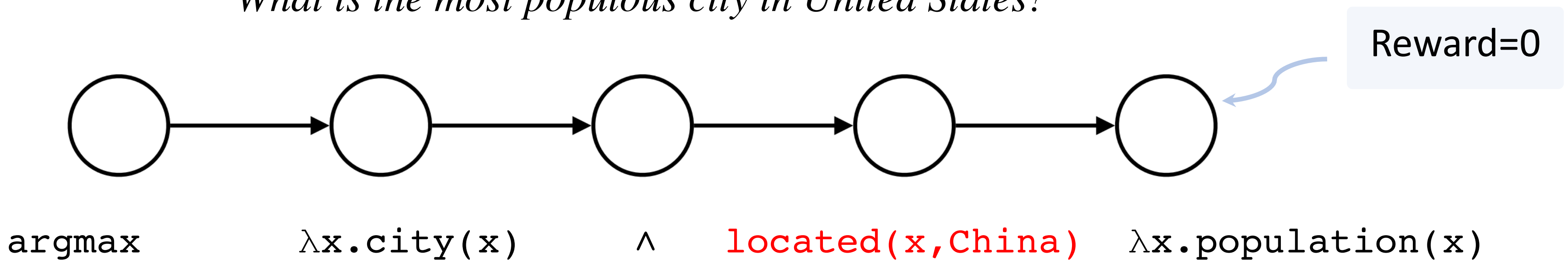
Weakly-supervised Learning Issue 2: Search Space

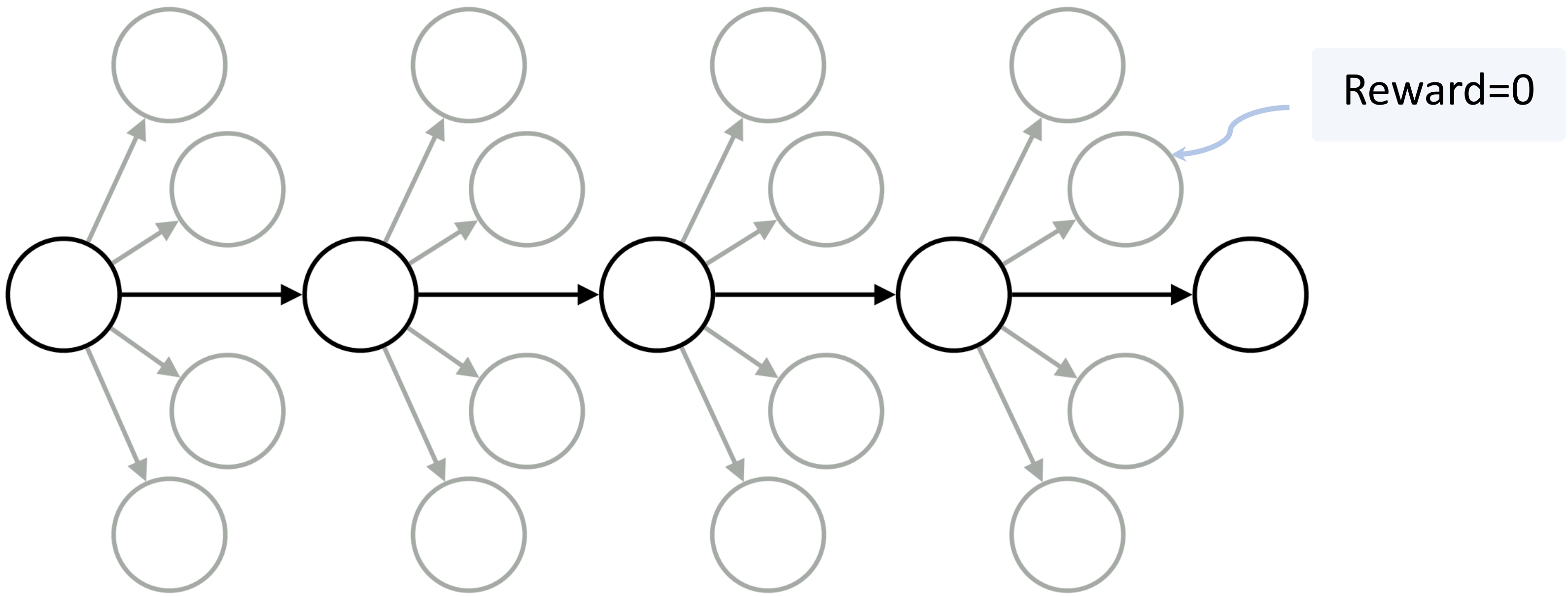
- The space of possible logical forms with correct answers is exponentially large
- How to search candidate logical forms more efficiently?

$$\nabla \log p_\theta(\boldsymbol{y}^*|\boldsymbol{x}) = \sum_{\underbrace{\boldsymbol{z}:\mathrm{answer}(\boldsymbol{z})=\boldsymbol{y}^*}_{\text{Prohibitively Large Search Space}}} w(\boldsymbol{z},\boldsymbol{x}) \cdot \nabla \log p_\theta(\boldsymbol{z}|\boldsymbol{x})$$

Efficient Search: Single Step Reward Observation

What is the most populous city in United States?





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

[Suhr and Artzi, 2018]