Carnegie Mellon University
School of Computer Science

Learning Executable Semantic Parsers for Natural Language Understanding

10-808: Language Grounding to Vision and Control

Sarah Mallepalle

5 September 2017

Introduction: Some Definitions

Semantic Parsers

- A map from natural language to semantic representations.
- Input: An utterance
- Output: A logical form

Logical Form

- Think of it as a “program” that can be executed to yield an answer or behavior from an utterance.

Introduction: Some Definitions

Example

- Context: Simple mathematics
- Utterance: What is the largest prime less than 10?
- Logical Form: max(primes ∩ (-∞, 10))
- Action: 7

Introduction: Statistical Semantic Parsers

- Semantic parsers started off as very rule-based.
  - Can only handle narrow domains.
  - Hard to generalize.

- Modern semantic parsers tend more towards machine learning algorithms:
  - Supervised learning.
  - Need a labeled dataset of natural language utterances paired with annotated logical forms.

Introduction: Statistical Semantic Parsers

- Semantic parsers started off as very rule-based.
  - Can only handle narrow domains.
  - Hard to generalize.

- Modern semantic parsers tend more towards machine learning algorithms:
  - Supervised learning.
  - Need a labeled dataset of natural language utterances paired with annotated logical forms.

Ideas for Statistical Semantic Parsers

- Goal 1: Reduce the amount of supervision from annotated logical forms to answers.
  - We are not given the actual logical form - we learn it!

- Before (Rules-Based):
  - Utterance -> Logical Form -> Action

- After (Machine Learning):
  - Utterance -> Action

- Benefit: Easier to obtain data through crowdsourcing.

Ideas for Statistical Semantic Parsers

- Goal 2: Scale semantic parsers to bigger domains
  - Working with larger amounts of data
  - Tackling a bigger set of applications
    - Question-Answer
    - Robot Navigation
    - Computer Vision

The Big Picture of Semantic Parsing

Linguistically:
Represent and construct semantics from natural language.

Statistical:
Learn semantic parsers from (weak) supervision and generalize it well to new data.

Computational:
How can we do this learning, and do it efficiently?

The Big Picture of Semantic Parsing

Purpose

- Given an utterance, *x*, in a context, *c*, output the desired action *y*.
- Learn the best intermediate logical form, *z*, that connects *x* and *y*.

Example

- c: Simple mathematics
- x: What is the largest prime less than 10?
- z: max(primes ∩ (-∞, 10))
- y: 7

Framework of a Statistical Semantic Parser

1. Executor
2. Grammar
3. Model
4. Parser
5. Learner

Example: What is the largest prime less than 10?

Framework: Grammar

- Grammar connects an input utterance to a set of possible derivations of logical forms, *D(x, c)*.

- Set of rules, a ⇒ b:
  - a: some span[i:j] of the utterance (0 ≤ i, j ≤ len(x))
  - b: syntactic category & logical form

Framework: Grammar

Example: What is the largest prime less than 10?

(1)	prime	⇒	NP[primes]	[4]
(2)	10	⇒	NP[10]	[7]
(3)	less than NP[z]	⇒	QP[(−∞, z)]	[5:7]
(4)	NP[z1] QP[z2]	⇒	NP[z1 ∩ z2]	[4:7]
(5)	largest NP[z]	⇒	NP[max(z)]	[3]
(6)	largest NP[z]	⇒	NP[min(z)]	[3]
(7)	What is the NP[z]?	⇒	ROOT[z]	[0:7]

Framework: Grammar

Example: What is the largest prime less than 10?

Rules:

(1)	prime	⇒	NP[primes]	[4]
(2)	10	⇒	NP[10]	[7]
(3)	less than NP[z]	⇒	QP[(−∞, z)]	[5:7]
(4)	NP[z1] QP[z2]	⇒	NP[z1 ∩ z2]	[4:7]
(5)	largest NP[z]	⇒	NP[max(z)]	[3]
(6)	**largest NP[z]**	**⇒**	**NP[min(z)]**	**[3]**
(7)	What is the NP[z]?	⇒	ROOT[z]	[0:7]

Framework: Grammar

Example: What is the largest prime less than 10?

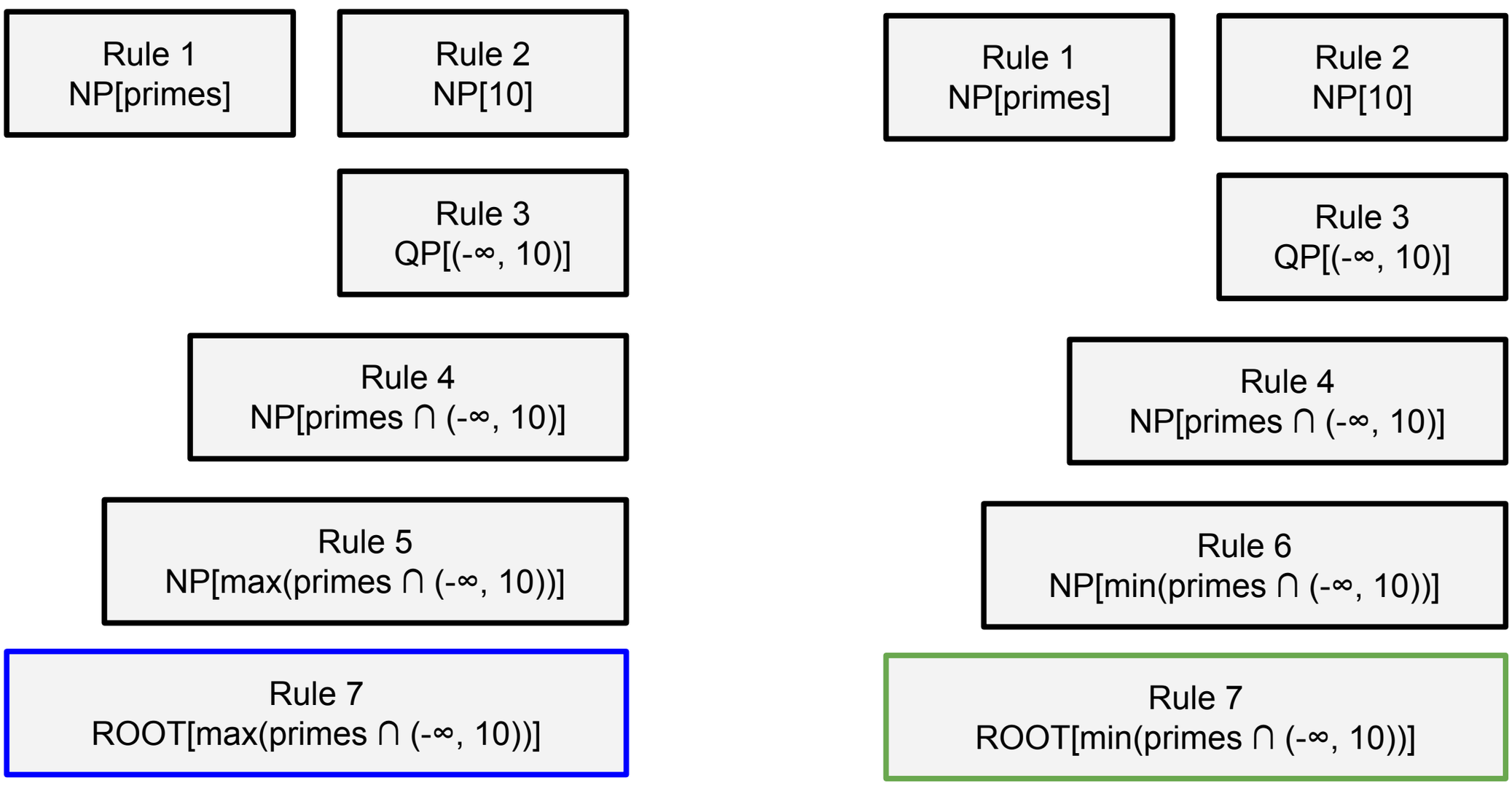
Rules:

(1)	prime	⇒	NP[primes]	[4]
(2)	10	⇒	NP[10]	[7]
(3)	less than NP[z]	⇒	QP[(−∞, z)]	[5:7]
(4)	NP[z1] QP[z2]	⇒	NP[z1 ∩ z2]	[4:7]
(5)	largest NP[z]	⇒	NP[max(z)]	[3]
(6)	largest NP[z]	⇒	NP[min(z)]	[3]
(7)	**What is the NP[z]?**	**⇒**	**ROOT[z]**	**[0:7]**

Framework: Grammar

Example: What is the largest prime less than 10?

Possible Derivations: D(x, c) = {d1, d2}



Framework: Model

- A model specifies a distribution, $p_\theta(d \mid x, c)$, that scores the each candidate derivation *d* in set D.

- Most common distribution for semantic parsers is a log-linear model.

Framework: Model

Example: What is the largest prime less than 10?

Log-Linear Model: Define Feature Vector

$$\phi(x, c, d) \in \mathbb{R}^F$$

- Let each feature be binary, corresponding to whether a grammar rule is used or not. F = 7.
- $\phi(x, c, d_1)$ = [1, 1, 1, 1, 1, 0, 1]
- $\phi(x, c, d_2)$ = [1, 1, 1, 1, 0, 1, 1]

Framework: Model

Example: What is the largest prime less than 10?

Log-Linear Model: Define Parameter Vector that weights each feature.

- θ = [0, 0, 0, 0, +1, -1, 0]
- Weighted combination: score(x, c, d) = $\phi * \theta$

Framework: Model

Example: What is the largest prime less than 10?

Log-Linear Model: Define distribution $p_\theta(d \mid x, c)$

$$p_\theta(d \mid x, c) = \frac{\exp(\text{score}(x, c, d))}{\sum_{d' \in D(x,c)} \exp(\text{score}(x, c, d'))}$$

Framework: Parser

The parser computes the highest probability derivation for the span of an utterance *x*.
Final, most probable derivations are collected in the ROOT category (span[0:len(x)])

Chart Parser: Recursively builds derivations for each span[i:j] of the utterance.

Beam Search: Only keep the derivations with the highest probability under model pθ.

Framework: Parser

Example: What is the largest prime less than 10?

(1)	prime	⇒	NP[primes]	[4]
(2)	10	⇒	NP[10]	[7]
(3)	less than NP[z]	⇒	QP[(−∞, z)]	[5:7]
(4)	NP[z1] QP[z2]	⇒	NP[z1 ∩ z2]	[4:7]
(5)	largest NP[z]	⇒	NP[max(z)]	[3]
(6)	largest NP[z]	⇒	NP[min(z)]	[3]
(7)	What is the NP[z]?	⇒	ROOT[z]	[0:7]

Framework: Learner

Learner: Estimates parameters theta given training examples {(x, c, y)}

Compute the likelihood of each action yi

Consider all derivations d whose logical form z results in action yi.

Optimizing an objective function, i.e. log-likelihood of observed action yi with respect to each d in D with SGD.

Framework: Learner

Example: What is the largest prime less than 10?

Answer, yi: 7

Possible interpretations of logical forms:

1. What is the largest prime less than 10?
2. How many days are there in a week?
3. What is Mickey Mantle’s jersey number?

Framework: Executor

The executor computes the denotation (action) y given a logical form z and context c.
Defines our semantic representation.
Concerned purely with what we want to express, regardless of how it would be expressed in natural language - forget utterance, x.

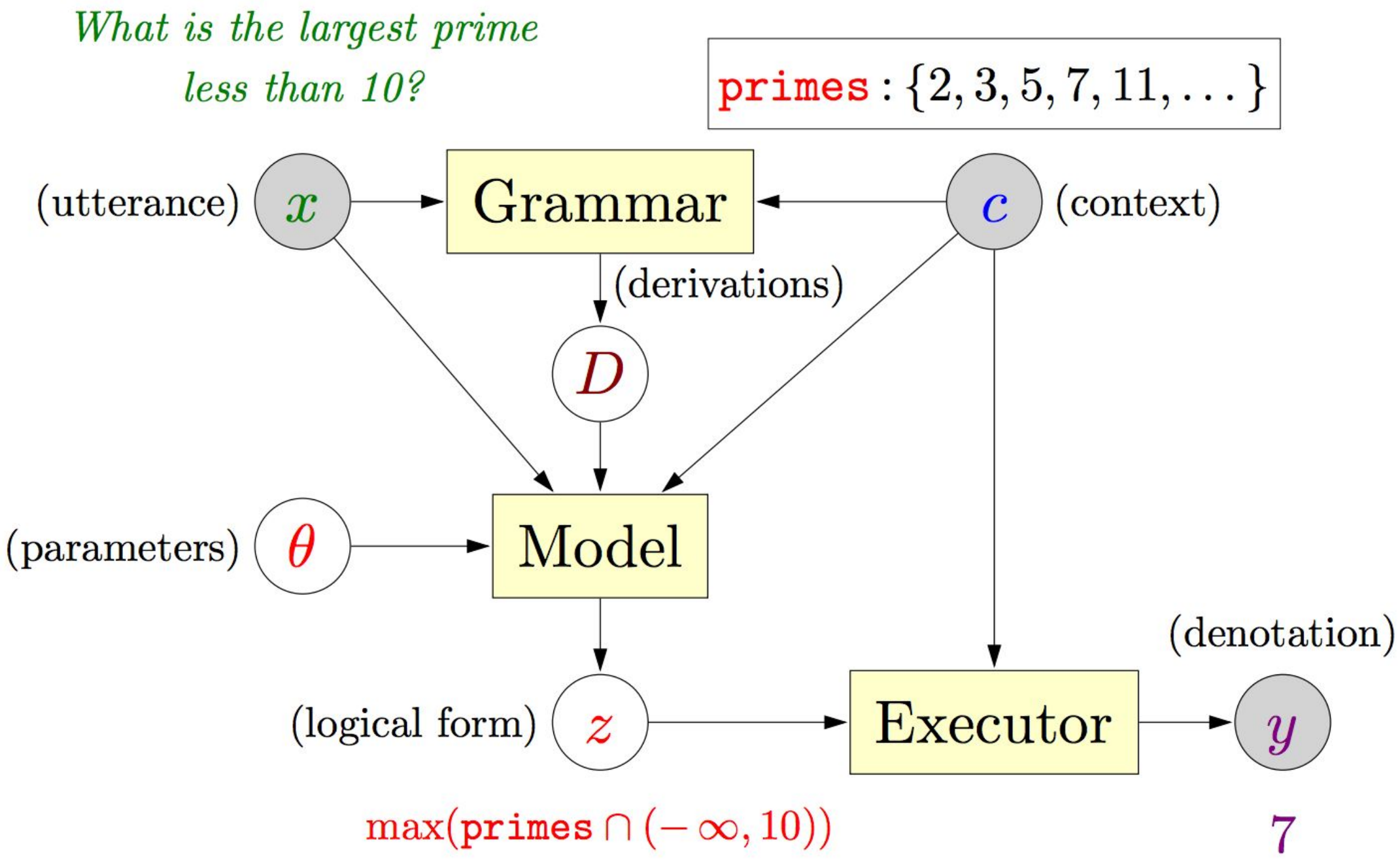
Example

- Context, c: Simple mathematics
- Logical Form, z: max(primes ∩ (-∞, 10))
- Action, y: 7

Framework: Summary

- Executor: What we want to express as a result.
- Grammar: Describes how candidate logical forms are constructed from an input.
- Model: Defines features and makes predictions on which derivation is most accurate.
- Parser: Makes logical forms from sub-phrases of input.
- Learner: Decides what logical form corresponds most accurately to a given action.

Framework: Summary



Questions about Semantic Parsing

- Best way to represent the semantics of a language?
  - First order logic
  - Lambda functions
- What is the best type of supervision to use?