# Natural Language Interfaces for Semi-Structured Web Pages

University Oral Examination

Ice Pasupat

# Motivation

AliGenie Bixby Clova

# Motivation

environment

...

# Motivation

environment

utterance

Alexa, play Despacito

...

# Motivation

environment

utterance

Alexa, play Despacito

actions

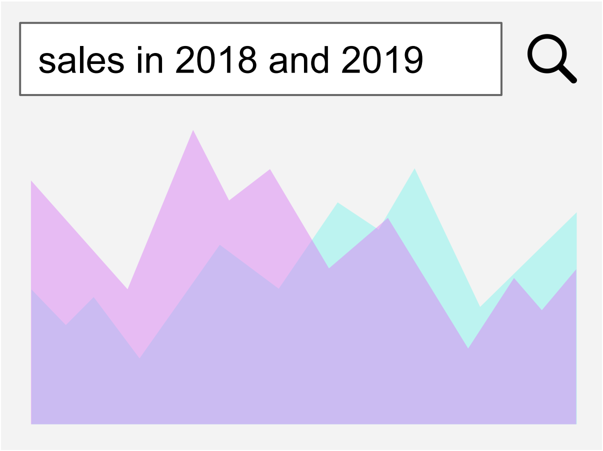
play(track=Despacito)

...

# Motivation


environment
utterance
Alexa, play Despacito
actions
play(track=Despacito)
response
amazon

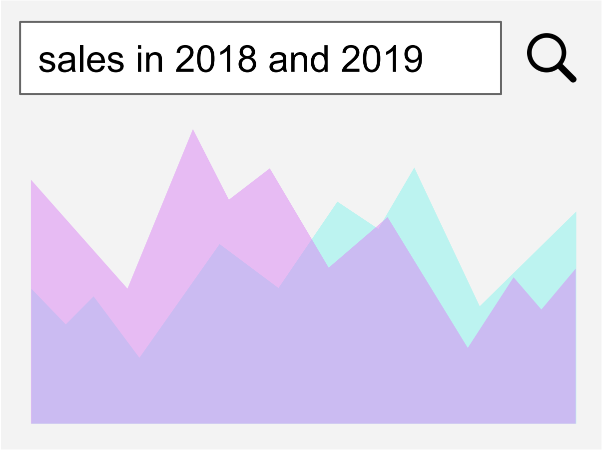
# Motivation




**natural language interface to database**

[Hendrix et al., 1978 / Androutsopoulos et al., 1995 / Popescu et al., 2003 / ...]

# Motivation





**natural language interface to database**

**smart search engines**

[Hendrix et al., 1978 / Androutsopoulos et al., 1995 / Popescu et al., 2003 / ...]

# Motivation

**Goal:** Extend the capability of these systems along two axes:

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)

How much is a Steak Burrito at Ray's Grill?

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
-

How much is a Steak Burrito at Ray's Grill?

?????

# Motivation

semantic parsing on databases

most virtual assistants

**database / apps**
limited schema

Scope of the environment (**breadth**)

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / ...]

# Motivation

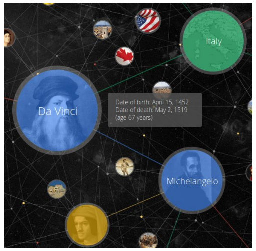
semantic parsing on databases

most virtual assistants

semantic parsing on knowledge bases

**database / apps**
limited schema

**knowledge base**
still limited schema




Freebase

Scope of the environment (**breadth**)

[Cai + Yates, 2013 / Berant et al., 2013 / Kwiatkowski et al., 2013 / Bordes et al., 2015 / ...]

# Motivation

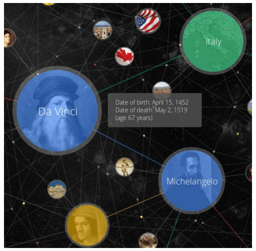
semantic parsing on databases

most virtual assistants

**database / apps**
limited schema

semantic parsing on knowledge bases

**knowledge base**
still limited schema




Freebase

web scraping

**web pages**
open schema


question answering on paragraphs

web search

**any texts**
open schema

The Lacey Act of 1900 was the first federal law that regulated commercial animal markets. It prohibited interstate commerce of animals killed in violation of state game laws, and covered all fish and wildlife and their parts or products, as well as plants. Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society–the species were relatively rare–and little opposition was raised.[9]

Scope of the environment (**breadth**)

[Robertson et al., 19xx / Hearst, 1992 / Richardson et al., 2013 / Rajpurkar et al., 2016 / ...]

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
- 

How much is a Steak Burrito at Ray's Grill?

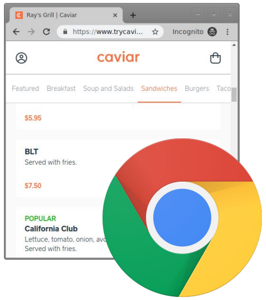
?????

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
-

How much is a Steak Burrito at Ray's Grill?




$8.50

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
- Task complexity (**depth**)

How much is a Steak Burrito at Ray's Grill?

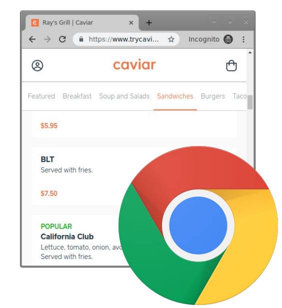

$8.50

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
- Task complexity (**depth**)

What's the cheapest burrito with chicken?




# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
- Task complexity (**depth**)

**Step 1:** Find burritos
**Step 2:** Filter for chicken
**Step 3:** Find the cheapest one

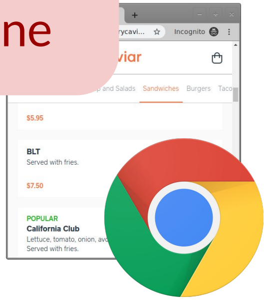
What's the cheapest burrito with chicken?

# Motivation

**Goal:** Extend the capability of these systems along two axes:

- Scope of the environment (**breadth**)
- Task complexity (**depth**)

**Step 1:** Find burritos
**Step 2:** Filter for chicken
**Step 3:** Find the cheapest one




What's the cheapest burrito with chicken?


Grilled Chicken Burrito ($6.95)

# Motivation

Task Complexity (**depth**)

Find pages with "burrito"

**surface form matching**

# Motivation

Task Complexity (**depth**)

How much is a steak burrito?

**1-2 step reasoning**

Find pages with "burrito"

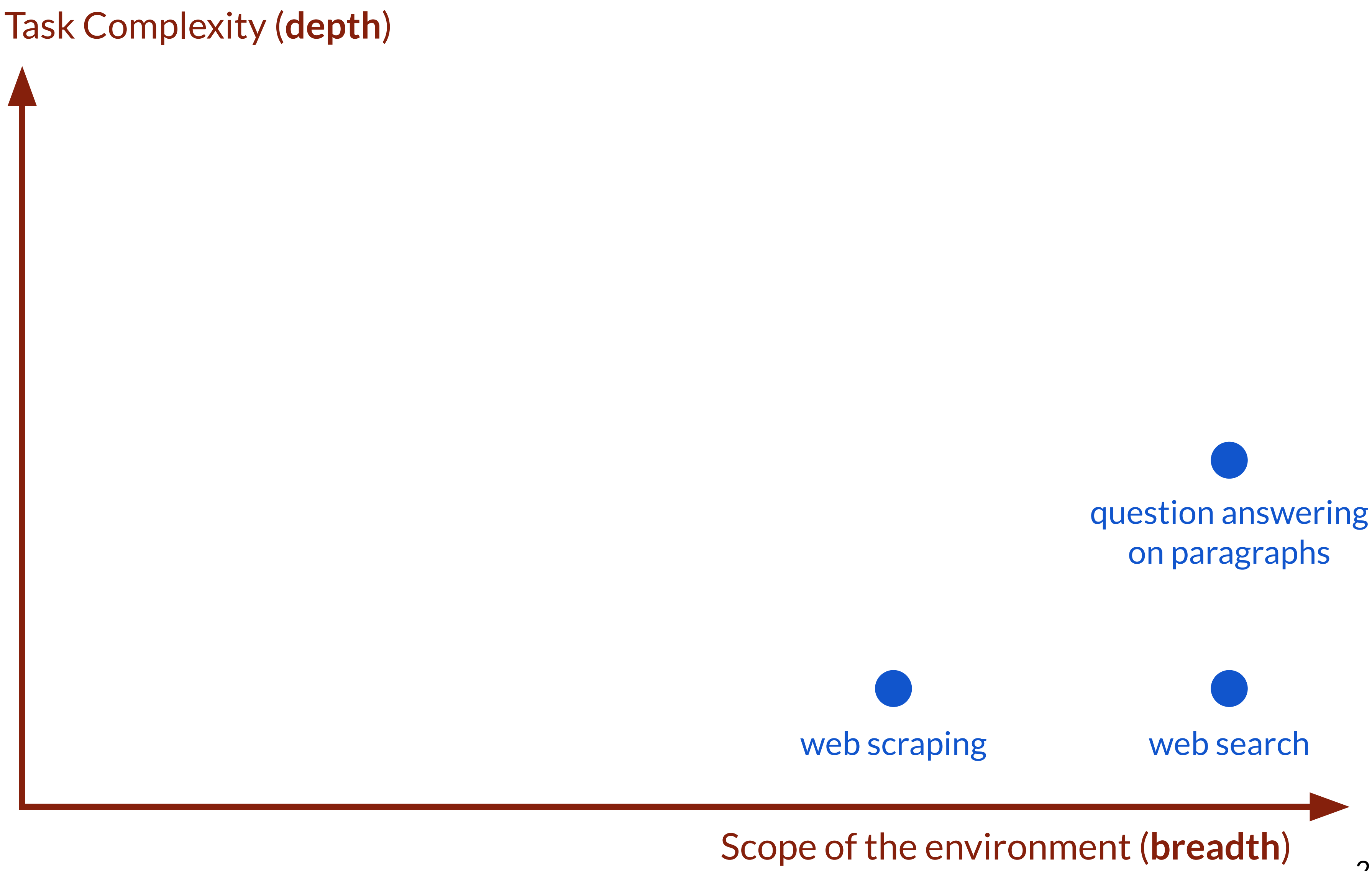
**surface form matching**

# Motivation

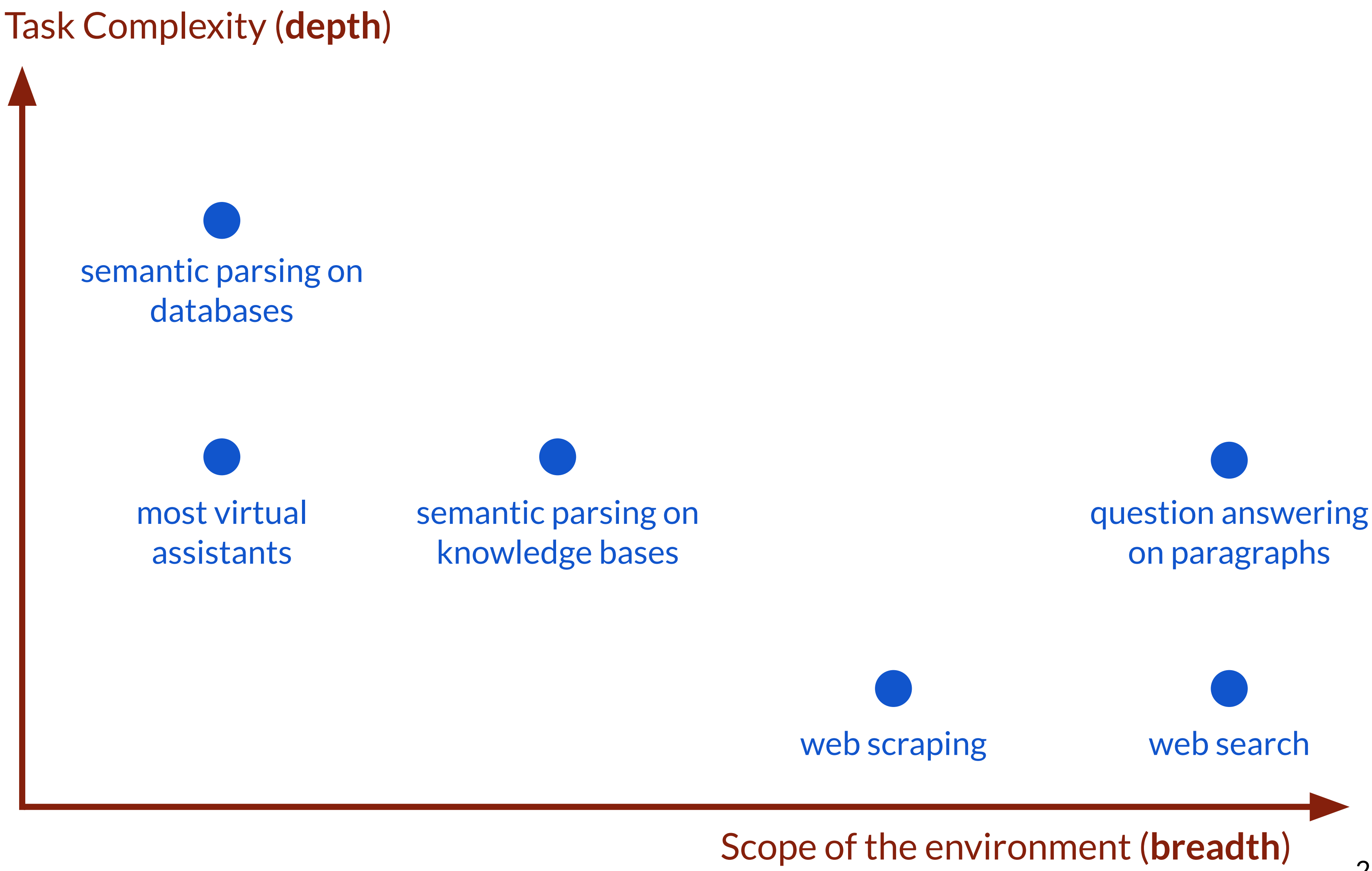
Task Complexity (**depth**)

| Task | Reasoning |
| --- | --- |
| What's the cheapest burrito with chicken? | **multi-step reasoning** |
| How much is a steak burrito? | **1-2 step reasoning** |
| Find pages with "burrito" | **surface form matching** |

# Motivation

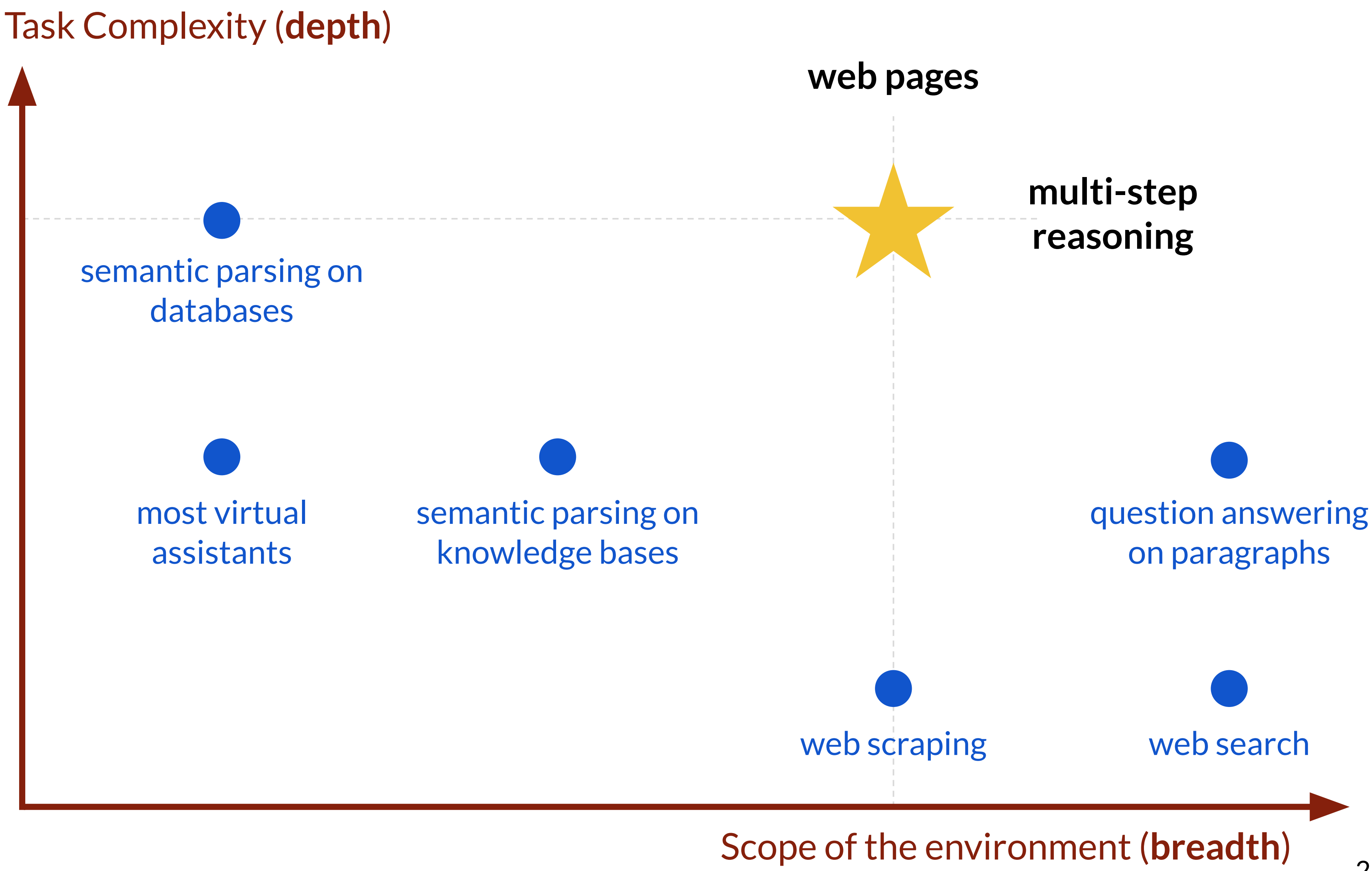

Task Complexity (**depth**)
question answering
on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

# Motivation



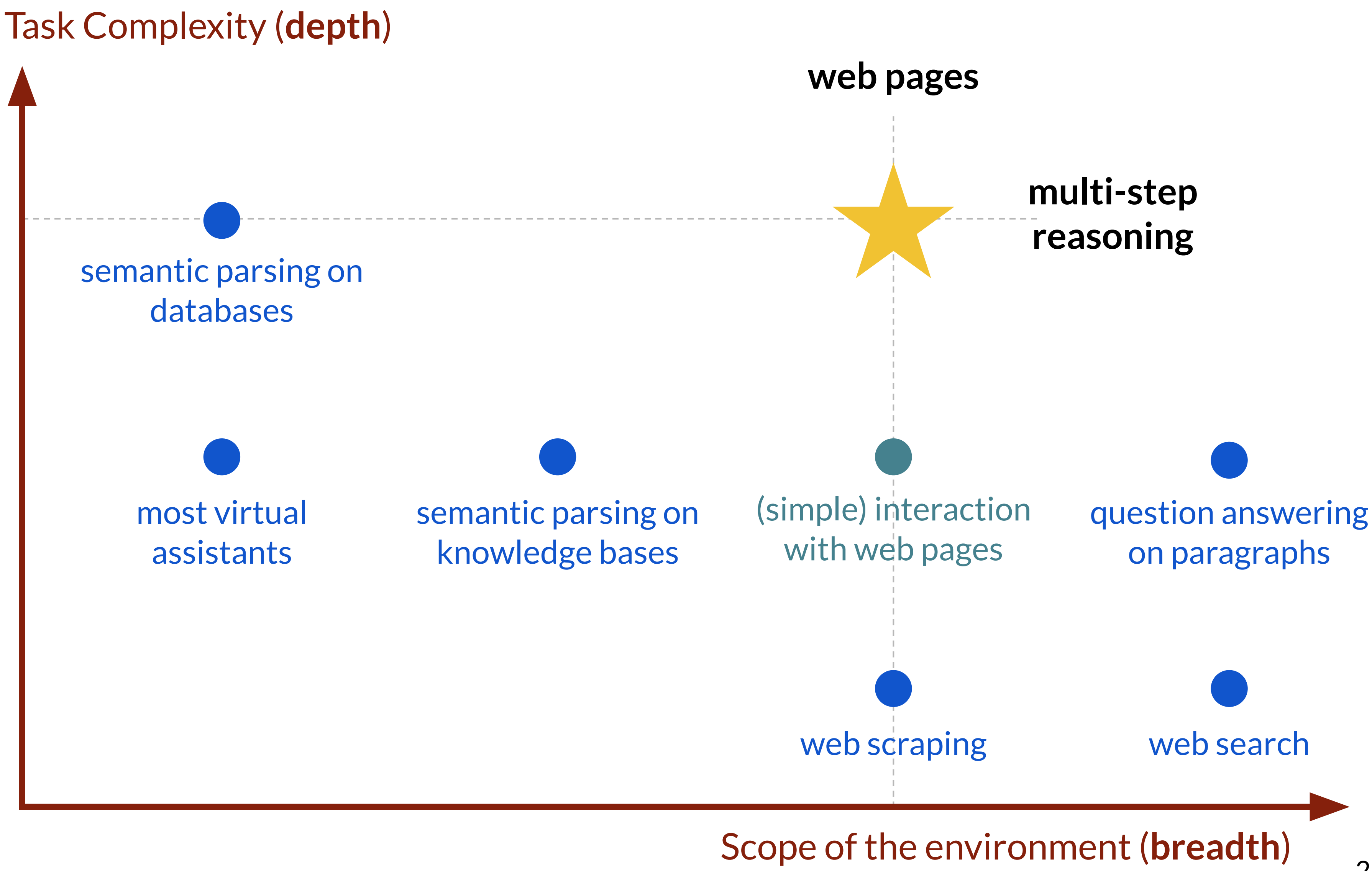
Task Complexity (**depth**)
semantic parsing on databases
most virtual assistants
semantic parsing on knowledge bases
question answering on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

# Motivation



Task Complexity (**depth**)
web pages
multi-step reasoning
semantic parsing on databases
most virtual assistants
semantic parsing on knowledge bases
question answering on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

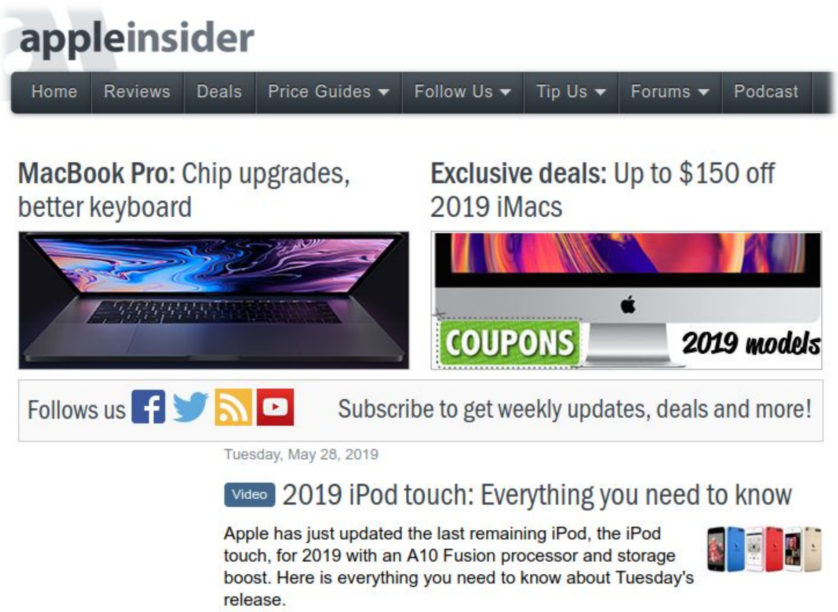
# Motivation


Task Complexity (**depth**)
web pages
multi-step reasoning
semantic parsing on databases
most virtual assistants
semantic parsing on knowledge bases
(simple) interaction with web pages
question answering on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

# Interacting with Web Pages

**interact with HTML elements based on the queries**

follow on facebook

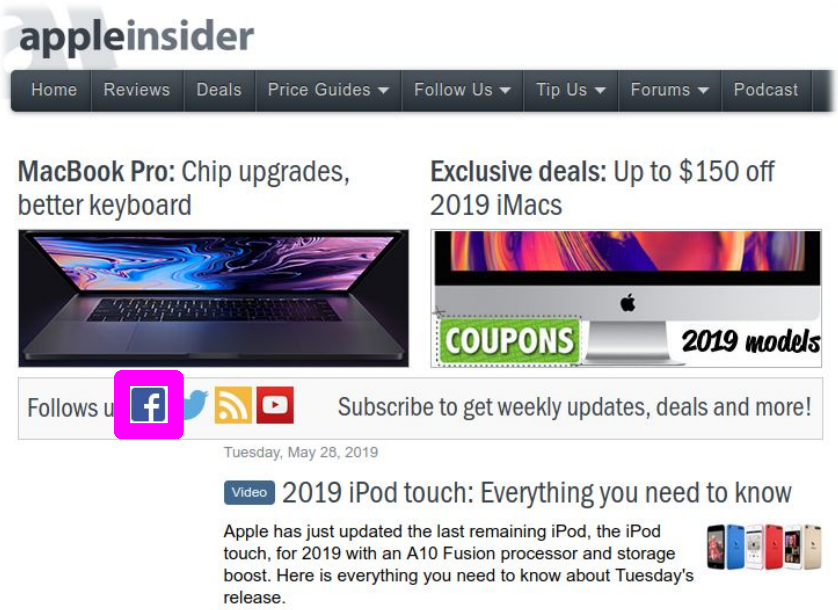



[P., Allan Jiang, Evan Liu, Kelvin Guu, Percy Liang, 2018]

# Interacting with Web Pages

**interact with HTML elements based on the queries**

follow on facebook

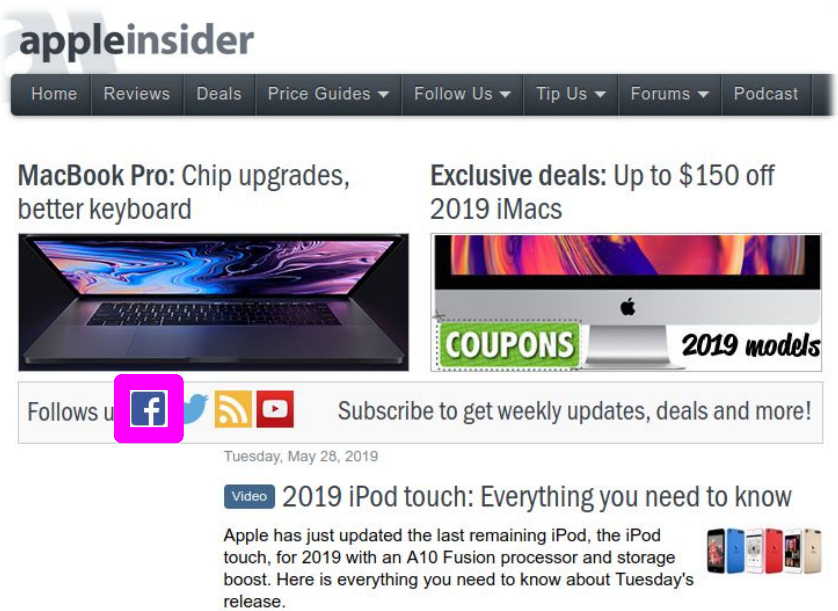



[P., Allan Jiang, Evan Liu, Kelvin Guu, Percy Liang, 2018]

# Interacting with Web Pages

**interact with HTML elements based on the queries**

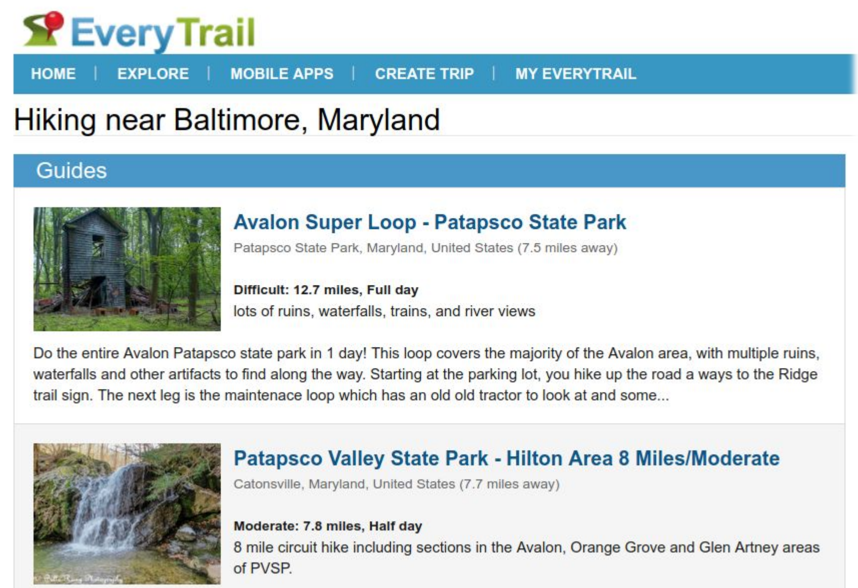
follow on facebook




[P., Allan Jiang, Evan Liu, Kelvin Guu, Percy Liang, 2018]

**extracting a list of entities from the web page**

hiking trails in baltimore

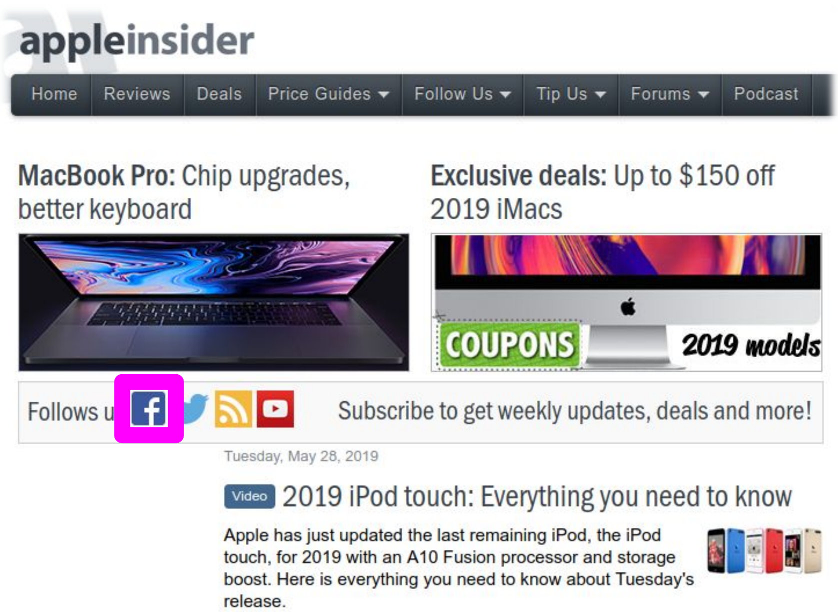



[P. and Liang, 2014]

# Interacting with Web Pages

**interact with HTML elements based on the queries**

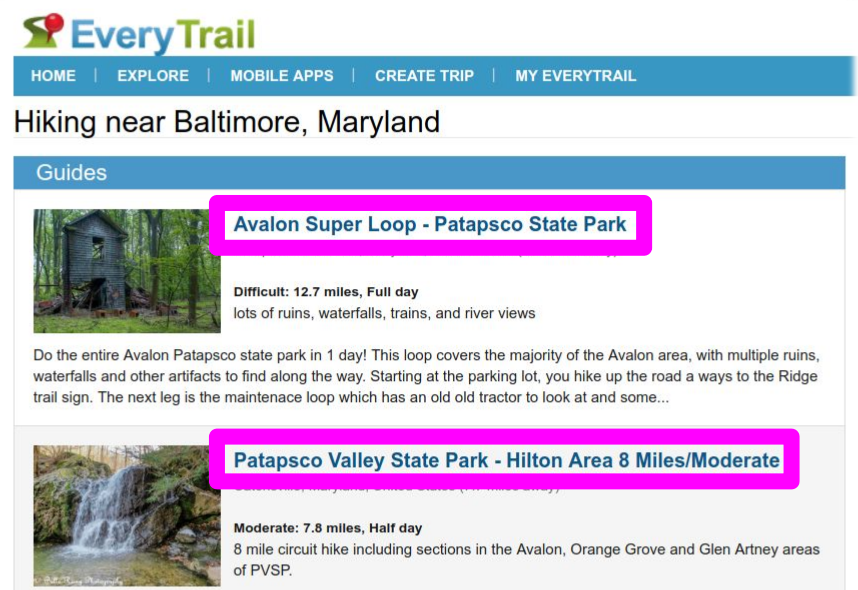
follow on facebook




[P., Allan Jiang, Evan Liu, Kelvin Guu, Percy Liang, 2018]

**extracting a list of entities from the web page**

hiking trails in baltimore




[P. and Liang, 2014]

# Interacting with Web Pages

Structures such as tables have potential for **complex reasoning**!

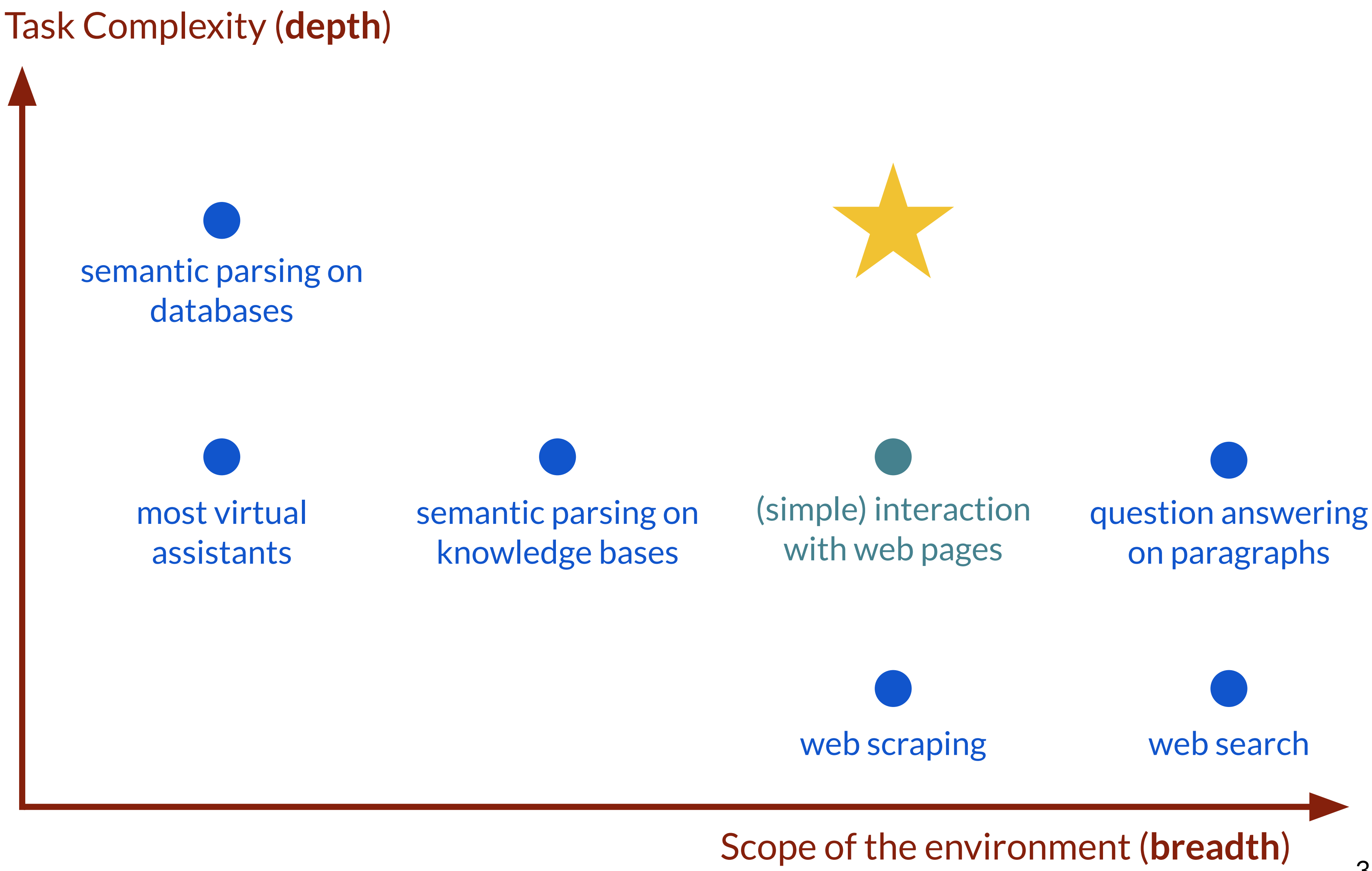
# Interacting with Web Pages

Structures such as tables have potential for **complex reasoning**!

list of hedge funds in new york

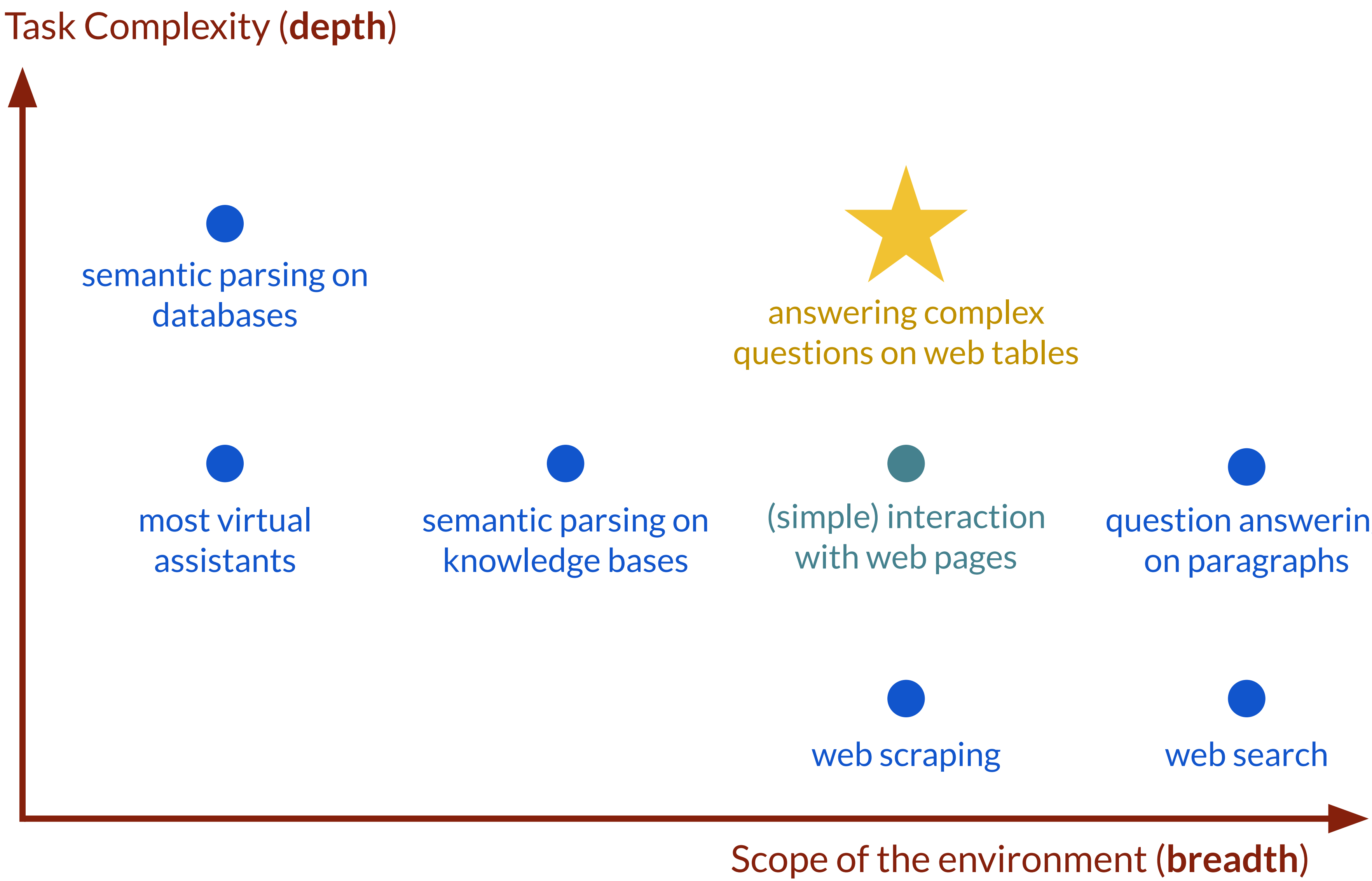
| Rank | Firm | Headquarters |
|---|---|---|
| 1 | Bridgewater Associates | Westport, CT |
| 2 | Man Group | London |
| 3 | J.P. Morgan Asset Management | New York |
| 4 | Brevan Howard Asset Management | London |
| 5 | Och-Ziff Capital Management Group | New York |
| 6 | Paulson & Co. | New York |
| 7 | BlackRock Advisors | New York |

# Motivation



Task Complexity (**depth**)
semantic parsing on databases
most virtual assistants
semantic parsing on knowledge bases
(simple) interaction with web pages
question answering on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

# Motivation


Task Complexity (**depth**)
semantic parsing on databases
answering complex questions on web tables
most virtual assistants
semantic parsing on knowledge bases
(simple) interaction with web pages
question answering on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

# Outline

- Motivation
- **Task: Answering complex questions on web tables**
- Approach: Semantic parsing with distant supervision
- Improvement 1: Make it faster
- Improvement 2: Convert to direct supervision

# Piotr Kędzia

## Competition record

| Year | Competition | Venue | Position | Event | Notes |
|---|---|---|---|---|---|
| Representing Poland | | | | | |
| 2001 | World Youth Championships | Debrecen, Hungary | 2nd | 400 m | 47.12 |
| | | | 1st | Medley relay | 1:50.46 |
| | European Junior Championships | Grosseto, Italy | 1st | 4x400 m relay | 3:06.12 |
| 2003 | European Junior Championships | Tampere, Finland | 3rd | 400 m | 46.69 |
| | | | 2nd | 4x400 m relay | 3:08.62 |
| 2005 | European U23 Championships | Erfurt, Germany | 11th (sf) | 400 m | 46.62 |
| | | | 1st | 4x400 m relay | 3:04.41 |
| | Universiade | Izmir, Turkey | 7th | 400 m | 46.89 |
| | | | 1st | 4x400 m relay | 3:02.57 |
| 2006 | World Indoor Championships | Moscow, Russia | 2nd (h) | 4x400 m relay | 3:06.10 |
| | European Championships | Gothenburg, Sweden | 3rd | 4x400 m relay | 3:01.73 |
| 2007 | European Indoor Championships | Birmingham, United Kingdom | 3rd | 4x400 m relay | 3:08.14 |
| | Universiade | Bangkok, Thailand | 7th | 400 m | 46.85 |
| | | | 1st | 4x400 m relay | 3:02.05 |
| 2008 | World Indoor Championships | Valencia, Spain | 4th | 4x400 m relay | 3:08.76 |
| | Olympic Games | Beijing, China | 7th | 4x400 m relay | 3:00.32 |
| 2009 | Universiade | Belgrade, Serbia | 2nd | 4x400 m relay | 3:05.69 |

[P. and Liang, 2015]

# Piotr Kędzia

## Competition record

| Year | Competition | Venue | Position | Event | Notes |
|---|---|---|---|---|---|
| Representing Poland | | | | | |
| 2001 | World Youth Championships | Debrecen, Hungary | 2nd | 400 m | 47.12 |
| | | | 1st | Medley relay | 1:50.46 |
| | European Junior Championships | Grosseto, Italy | 1st | 4x400 m relay | 3:06.12 |
| 2003 | European Junior Championships | Tampere, Finland | 3rd | 400 m | 46.69 |
| | | | 2nd | 4x400 m relay | 3:08.62 |
| 2005 | European U23 Championships | Erfurt, Germany | 11th (sf) | 400 m | 46.62 |
| | | | 1st | 4x400 m relay | 3:04.41 |
| | Universiade | Izmir, Turkey | 7th | 400 m | 46.89 |
| | | | 1st | 4x400 m relay | 3:02.57 |
| 2006 | World Indoor Championships | Moscow, Russia | 2nd (h) | 4x400 m relay | 3:06.10 |
| | European Championships | Gothenburg, Sweden | 3rd | 4x400 m relay | 3:01.73 |
| 2007 | European Indoor Championships | Birmingham, United Kingdom | 3rd | 4x400 m relay | 3:08.14 |
| | Universiade | Bangkok, Thailand | 7th | 400 m | 46.85 |
| | | | 1st | 4x400 m relay | 3:02.05 |
| 2008 | World Indoor Championships | Valencia, Spain | 4th | 4x400 m relay | 3:08.76 |
| | Olympic Games | Beijing, China | 7th | 4x400 m relay | 3:00.32 |
| 2009 | Universiade | Belgrade, Serbia | 2nd | 4x400 m relay | 3:05.69 |

In what city did Piotr's last 1st place finish occur?

[P. and Liang, 2015]

# WikiTableQuestions Dataset

A new dataset of Wikipedia tables, complex questions, and answers

- 22K examples
- 2K tables

[P. and Liang, 2015]

# WikiTableQuestions Dataset

## Task Complexity (depth):

| Year | Competition | Venue | Position | Event | Notes |
|---|---|---|---|---|---|
| Representing Poland | | | | | |
| 2001 | World Youth Championships | Debrecen, Hungary | 2nd | 400 m | 47.12 |
| | | | 1st | Medley relay | 1:50.46 |
| | European Junior Championships | Grosseto, Italy | 1st | 4x400 m relay | 3:06.12 |
| 2003 | European Junior Championships | Tampere, Finland | 3rd | 400 m | 46.69 |
| | | | 2nd | 4x400 m relay | 3:08.62 |
| 2005 | European U23 Championships | Erfurt, Germany | 11th (sf) | 400 m | 46.62 |
| | | | 1st | 4x400 m relay | 3:04.41 |
| | Universiade | Izmir, Turkey | 7th | 400 m | 46.89 |
| | | | 1st | 4x400 m relay | 3:02.57 |
| 2006 | World Indoor Championships | Moscow, Russia | 2nd (h) | 4x400 m relay | 3:06.10 |
| | European Championships | Gothenburg, Sweden | 3rd | 4x400 m relay | 3:01.73 |
| 2007 | European Indoor Championships | Birmingham, United Kingdom | 3rd | 4x400 m relay | 3:08.14 |
| | Universiade | Bangkok, Thailand | 7th | 400 m | 46.85 |
| | | | 1st | 4x400 m relay | 3:02.05 |
| 2008 | World Indoor Championships | Valencia, Spain | 4th | 4x400 m relay | 3:08.76 |
| | Olympic Games | Beijing, China | 7th | 4x400 m relay | 3:00.32 |
| 2009 | Universiade | Belgrade, Serbia | 2nd | 4x400 m relay | 3:05.69 |

How long did it take this competitor to finish the 4x400 meter relay at Universiade in 2005?

[P. and Liang, 2015]

# WikiTableQuestions Dataset

## Task Complexity (depth):

| Year | Competition | Venue | Position | Event | Notes |
|---|---|---|---|---|---|
| Representing Poland | | | | | |
| 2001 | World Youth Championships | Debrecen, Hungary | 2nd | 400 m | 47.12 |
| | | | 1st | Medley relay | 1:50.46 |
| | European Junior Championships | Grosseto, Italy | 1st | 4x400 m relay | 3:06.12 |
| 2003 | European Junior Championships | Tampere, Finland | 3rd | 400 m | 46.69 |
| | | | 2nd | 4x400 m relay | 3:08.62 |
| 2005 | European U23 Championships | Erfurt, Germany | 11th (sf) | 400 m | 46.62 |
| | | | 1st | 4x400 m relay | 3:04.41 |
| | Universiade | Izmir, Turkey | 7th | 400 m | 46.89 |
| | | | 1st | 4x400 m relay | 3:02.57 |
| 2006 | World Indoor Championships | Moscow, Russia | 2nd (h) | 4x400 m relay | 3:06.10 |
| | European Championships | Gothenburg, Sweden | 3rd | 4x400 m relay | 3:01.73 |
| 2007 | European Indoor Championships | Birmingham, United Kingdom | 3rd | 4x400 m relay | 3:08.14 |
| | Universiade | Bangkok, Thailand | 7th | 400 m | 46.85 |
| | | | 1st | 4x400 m relay | 3:02.05 |
| 2008 | World Indoor Championships | Valencia, Spain | 4th | 4x400 m relay | 3:08.76 |
| | Olympic Games | Beijing, China | 7th | 4x400 m relay | 3:00.32 |
| 2009 | Universiade | Belgrade, Serbia | 2nd | 4x400 m relay | 3:05.69 |

How long did it take this competitor to finish the 4x400 meter relay at Universiade in 2005?

Where was the competition held immediately before the one in Turkey?

[P. and Liang, 2015]

# WikiTableQuestions Dataset

## Task Complexity (depth):

| Year | Competition | Venue | Position | Event | Notes |
|---|---|---|---|---|---|
| Representing Poland | | | | | |
| 2001 | World Youth Championships | Debrecen, Hungary | 2nd | 400 m | 47.12 |
| | | | 1st | Medley relay | 1:50.46 |
| | European Junior Championships | Grosseto, Italy | 1st | 4x400 m relay | 3:06.12 |
| 2003 | European Junior Championships | Tampere, Finland | 3rd | 400 m | 46.69 |
| | | | 2nd | 4x400 m relay | 3:08.62 |
| 2005 | European U23 Championships | Erfurt, Germany | 11th (sf) | 400 m | 46.62 |
| | | | 1st | 4x400 m relay | 3:04.41 |
| | Universiade | Izmir, Turkey | 7th | 400 m | 46.89 |
| | | | 1st | 4x400 m relay | 3:02.57 |
| 2006 | World Indoor Championships | Moscow, Russia | 2nd (h) | 4x400 m relay | 3:06.10 |
| | European Championships | Gothenburg, Sweden | 3rd | 4x400 m relay | 3:01.73 |
| 2007 | European Indoor Championships | Birmingham, United Kingdom | 3rd | 4x400 m relay | 3:08.14 |
| | Universiade | Bangkok, Thailand | 7th | 400 m | 46.85 |
| | | | 1st | 4x400 m relay | 3:02.05 |
| 2008 | World Indoor Championships | Valencia, Spain | 4th | 4x400 m relay | 3:08.76 |
| | Olympic Games | Beijing, China | 7th | 4x400 m relay | 3:00.32 |
| 2009 | Universiade | Belgrade, Serbia | 2nd | 4x400 m relay | 3:05.69 |

How long did it take this competitor to finish the 4x400 meter relay at Universiade in 2005?

Where was the competition held immediately before the one in Turkey?

How many times has this competitor placed 5th or better in competition?

[P. and Liang, 2015]

# WikiTableQuestions Dataset

## Scope of the environment (breadth):

- Diverse data types: 4K different column headers among 2K tables

| Year | Division | League | Regular Season | Playoffs | Open Cup | Avg. Attendance |
|---|---|---|---|---|---|---|
| 2001 | 2 | USL A-League | 4th, Western | Quarterfinals | *Did not qualify* | 7,169 |
| 2002 | 2 | USL A-League | 2nd, Pacific | 1st Round | *Did not qualify* | 6,260 |
| 2003 | 2 | USL A-League | 3rd, Pacific | *Did not qualify* | *Did not qualify* | 5,871 |
| 2004 | 2 | USL A-League | 1st, Western | Quarterfinals | 4th Round | 5,628 |
| 2005 | 2 | USL First Division | 5th | Quarterfinals | 4th Round | 6,028 |
| 2006 | 2 | USL First Division | 11th | *Did not qualify* | 3rd Round | 5,575 |
| 2007 | 2 | USL First Division | 2nd | Semifinals | 2nd Round | 6,851 |
| 2008 | 2 | USL First Division | 11th | *Did not qualify* | 1st Round | 8,567 |
| 2009 | 2 | USL First Division | 1st | Semifinals | 3rd Round | 9,734 |
| 2010 | 2 | USSF D-2 Pro League | 3rd, USL (3rd) | Quarterfinals | 3rd Round | 10,727 |

| Year | Competition | Venue | Position | Event | Notes |
|---|---|---|---|---|---|
| Representing Poland | | | | | |
| 2001 | World Youth Championships | Debrecen, Hungary | 2nd | 400 m | 47.12 |
| | | | 1st | Medley relay | 1:50.46 |
| | European Junior Championships | Grosseto, Italy | 1st | 4 × 400 m relay | 3:06.12 |
| 2002 | World Junior Championships | Kingston, Jamaica | 4th | 4 × 400 m relay | 3:06.25 |
| 2003 | European Junior Championships | Tampere, Finland | 3rd | 400 m | 46.69 |
| | | | 2nd | 4 × 400 m relay | 3:08.62 |
| 2005 | European U23 Championships | Erfurt, Germany | 11th (sf) | 400 m | 46.62 |
| | | | 1st | 4 × 400 m relay | 3:04.41 |
| | Universiade | Izmir, Turkey | 7th | 400 m | 46.89 |
| | | | 1st | 4 × 400 m relay | 3:02.57 |
| 2006 | World Indoor Championships | Moscow, Russia | 2nd (h) | 4 × 400 m relay | 3:06.10 |
| | European Championships | Gothenburg, Sweden | 3rd | 4 × 400 m relay | 3:01.73 |
| 2007 | European Indoor Championships | Birmingham, United Kingdom | 3rd | 4 × 400 m relay | 3:08.14 |
| | Universiade | Bangkok, Thailand | 7th | 400 m | 46.85 |
| | | | 1st | 4 × 400 m relay | 3:02.05 |
| 2008 | World Indoor Championships | Valencia, Spain | 4th | 4 × 400 m relay | 3:08.76 |
| | Olympic Games | Beijing, China | 6th | 4 × 400 m relay | 3:00.32 |
| 2009 | Universiade | Belgrade, Serbia | 2nd | 4 × 400 m relay | 3:05.69 |

| | Team | County | Wins | Years won |
|---|---|---|---|---|
| 1 | Two Mile House | Kildare | 1 | 2018 |
| | Kilanerin–Ballyfad | Wexford | 1 | 2017 |
| | St. Colmcille's | Meath | 1 | 2016 |
| | Ratoath | Meath | 1 | 2015 |
| | Seán O'Mahonys | Louth | 1 | 2014 |
| | Geraldines | Louth | 1 | 2013 |
| | Monasterevin | Kildare | 1 | 2012 |
| | Éire Óg Greystones | Wicklow | 1 | 2011 |
| | Ballymore Eustace | Kildare | 1 | 2010 |
| | Maynooth | Kildare | 1 | 2009 |
| | Ballyroan Abbey | Laois | 1 | 2008 |
| | Fingal Ravens | Dublin | 1 | 2007 |
| | Confey | Kildare | 1 | 2006 |
| | Crettyard | Laois | 1 | 2005 |
| | Wolfe Tones | Meath | 1 | 2004 |
| Dundalk Gaels | Louth | 1 | 2003 | |

| Rank | City | Passengers | Ranking | Airline |
|---|---|---|---|---|
| 1 | United States, Los Angeles | 15,383 | – | Alaska Airlines |
| 2 | Canada, Calgary | 9,279 | – | WestJet |
| 3 | Canada, Toronto | 2,241 | ▲ 3 | Sunwing Airlines |
| 4 | United States, Houston | 1,730 | ▲ 1 | United Express |
| 5 | United States, Minneapolis | 1,279 | ▲ 1 | Sun Country Airlines |
| 6 | United States, Phoenix-Sky Harbor | 821 | ▲ 2 | American Airlines |

| Date | Opponent | Venue | Result | Attendance | Scorers | Notes |
|---|---|---|---|---|---|---|
| 15 August 1987 | Derby County | Away | 0–1 | 17,204 | — | |
| 18 August 1987 | Coventry City | Home | 0–1 | 9,380 | — | |
| 22 August 1987 | West Ham United | Home | 2–2 | 8,073 | Harford (2) | |
| 29 August 1987 | Chelsea | Away | 0–3 | 16,075 | — | |
| 31 August 1987 | Arsenal | Home | 1–1 | 8,745 | Wilson (pen) | |
| 5 September 1987 | Oxford United | Away | 5–2 | 6,804 | Breacker, Harford, Hill, Nwajiobi, B. Stein | |
| 12 September 1987 | Everton | Home | 2–1 | 8,124 | Hill, B. Stein | |
| 19 September 1987 | Charlton Athletic | Away | 0–1 | 5,002 | — | |
| 26 September 1987 | Queens Park Rangers | Away | 0–2 | 11,175 | — | |
| 3 October 1987 | Manchester United | Home | 1–1 | 9,137 | Harford | |
| 10 October 1987 | Portsmouth | Away | 1–3 | 12,391 | Harford (pen) | |
| 17 October 1987 | Wimbledon | Home | 2–0 | 7,018 | B. Stein, Wilson | |
| 24 October 1987 | Liverpool | Home | 0–1 | 11,997 | — | |
| 7 November 1987 | Newcastle United | Home | 4–0 | 7,638 | Nwajiobi, B. Stein, M. Stein (2) | |
| 14 November 1987 | Sheffield Wednesday | Away | 2–0 | 16,960 | Allinson, M. Stein | |
| 21 November 1987 | Tottenham Hotspur | Home | 2–0 | 10,091 | Allinson (2) | |
| 5 December 1987 | Norwich City | Home | 1–2 | 7,002 | B. Stein | |
| 12 December 1987 | Watford | Away | 1–0 | 12,152 | Foster | |
| 18 December 1987 | Southampton | Home | 2–2 | 6,618 | Harford, McDonough | |
| 26 December 1987 | Everton | Away | 0–2 | 32,128 | — | |
| 28 December 1987 | Charlton Athletic | Home | 1–0 | 7,243 | Wilson | |
| 1 January 1988 | Chelsea | Home | 3–0 | 8,018 | Harford, B. Stein, M. Stein | |
| 2 January 1988 | West Ham United | Away | 1–1 | 16,716 | M. Stein | |
| 16 January 1988 | Derby County | Home | 1–0 | 7,175 | McDonough | |
| 6 February 1988 | Oxford United | Home | 7–4 | 8,063 | Harford (2), McDonough, B.Stein, M.Stein (3) | |
| 13 February 1988 | Arsenal | Away | 1–2 | 22,612 | M.Stein | |
| 5 March 1988 | Wimbledon | Away | 0–2 | 4,854 | — | |
| 15 March 1988 | Coventry City | Away | 0–4 | 13,711 | — | |
| 29 March 1988 | Portsmouth | Home | 4–1 | 6,740 | B.Stein, M.Stein, Wilson, own goal | |

[P. and Liang, 2015]

# WikiTableQuestions Dataset

**Scope of the environment (breadth):**

- Diverse data types: 4K different column headers among 2K tables
- Only ~20% can be answered by Freebase (a large knowledge base)

[P. and Liang, 2015]

# WikiTableQuestions Dataset

**Scope of the environment (breadth):**

- Diverse data types: 4K different column headers among 2K tables
- Only ~20% can be answered by Freebase (a large knowledge base)

**We ensure that tables in the test set do not appear in the training data**

- This tests the model's ability to generalize to unseen data schema

[P. and Liang, 2015]

# Outline

- Motivation
- Task: Answering complex questions on web tables
- **Approach: Semantic parsing with distant supervision**
- Improvement 1: Make it faster
- Improvement 2: Convert to direct supervision

# Semantic Parsing

**Idea:** Parse the utterance into an executable **logical form**

Where did the last 1st place finish occur?

VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Semantic Parsing

**Idea:** Parse the utterance into an executable **logical form**

Where did the last 1st place finish occur?

VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Semantic Parsing

**Idea:** Parse the utterance into an executable **logical form**

Where did the last 1st place finish occur?

VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Semantic Parsing

**Idea:** Parse the utterance into an executable **logical form**

Where did the last 1st place finish occur?

VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Semantic Parsing

**Idea:** Parse the utterance into an executable **logical form**

Where did the last 1st place finish occur?

VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Semantic Parsing

**Idea:** Parse the utterance into an executable **logical form**

Where did the last 1st place finish occur?

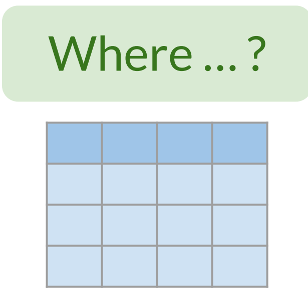
VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

"**denotation**"

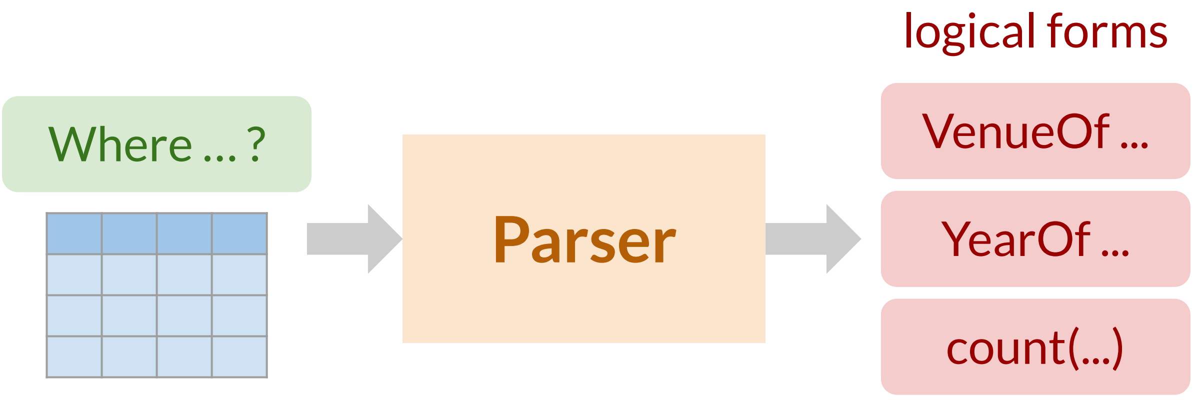
[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Prediction



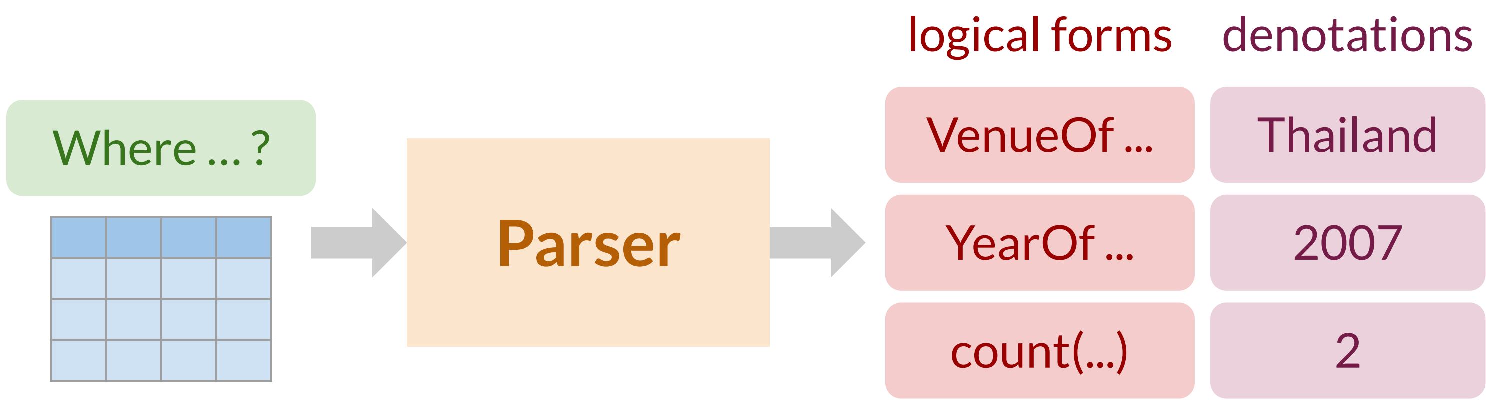

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Prediction



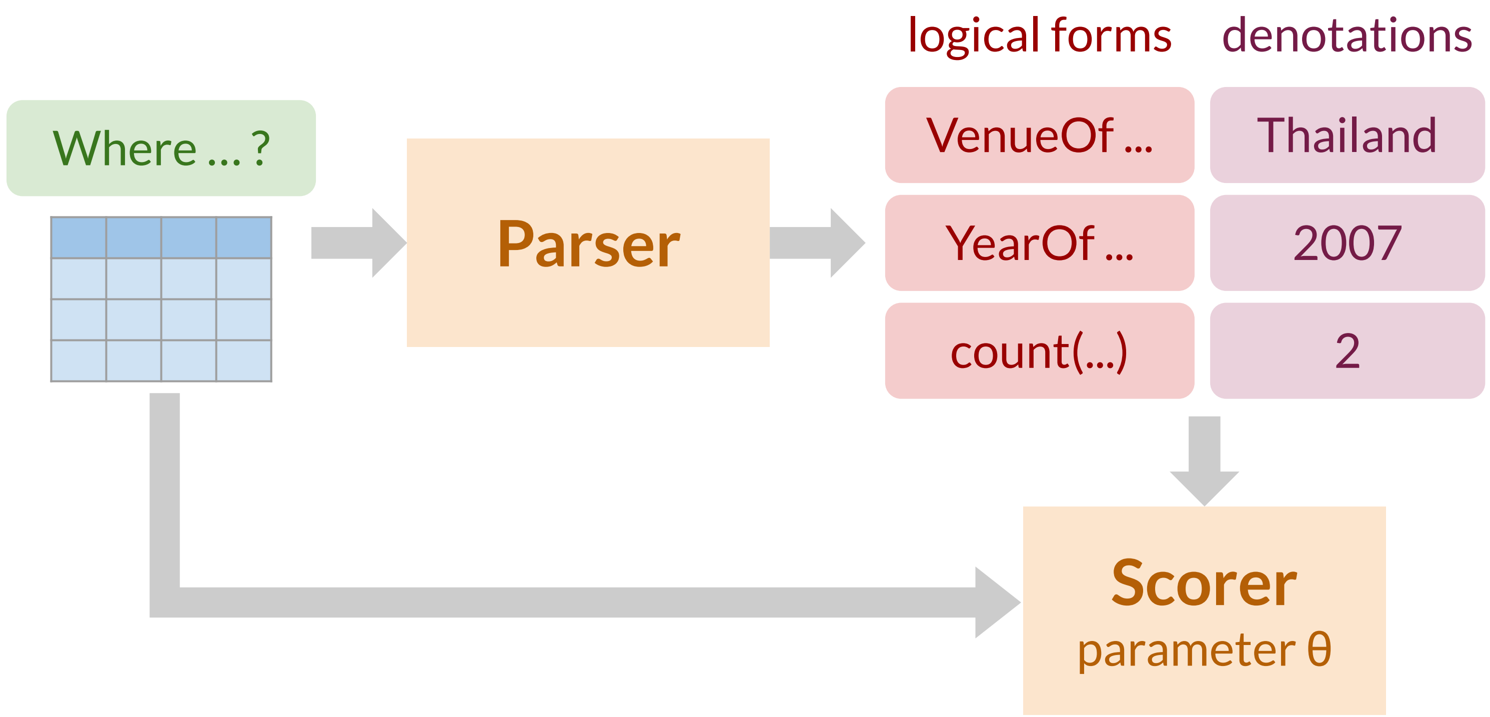

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Prediction



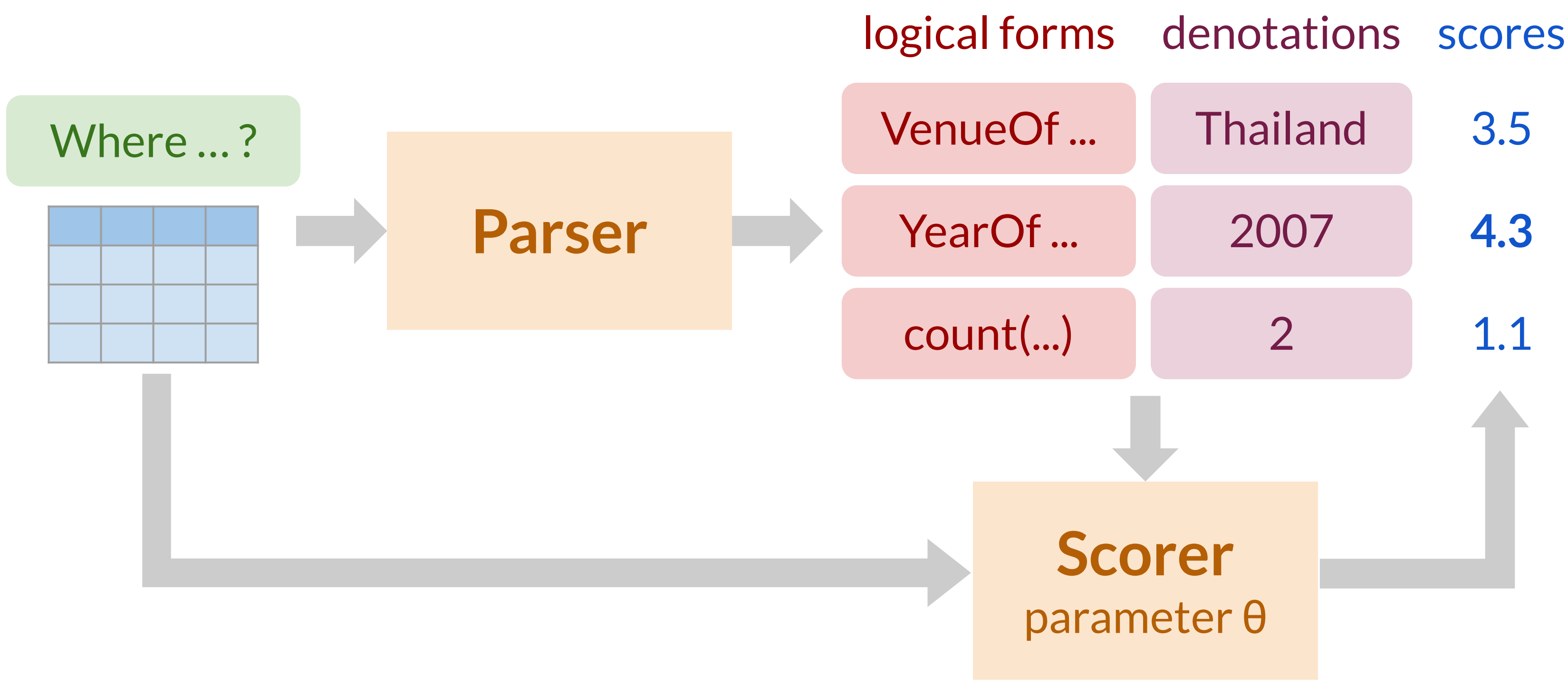

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Prediction



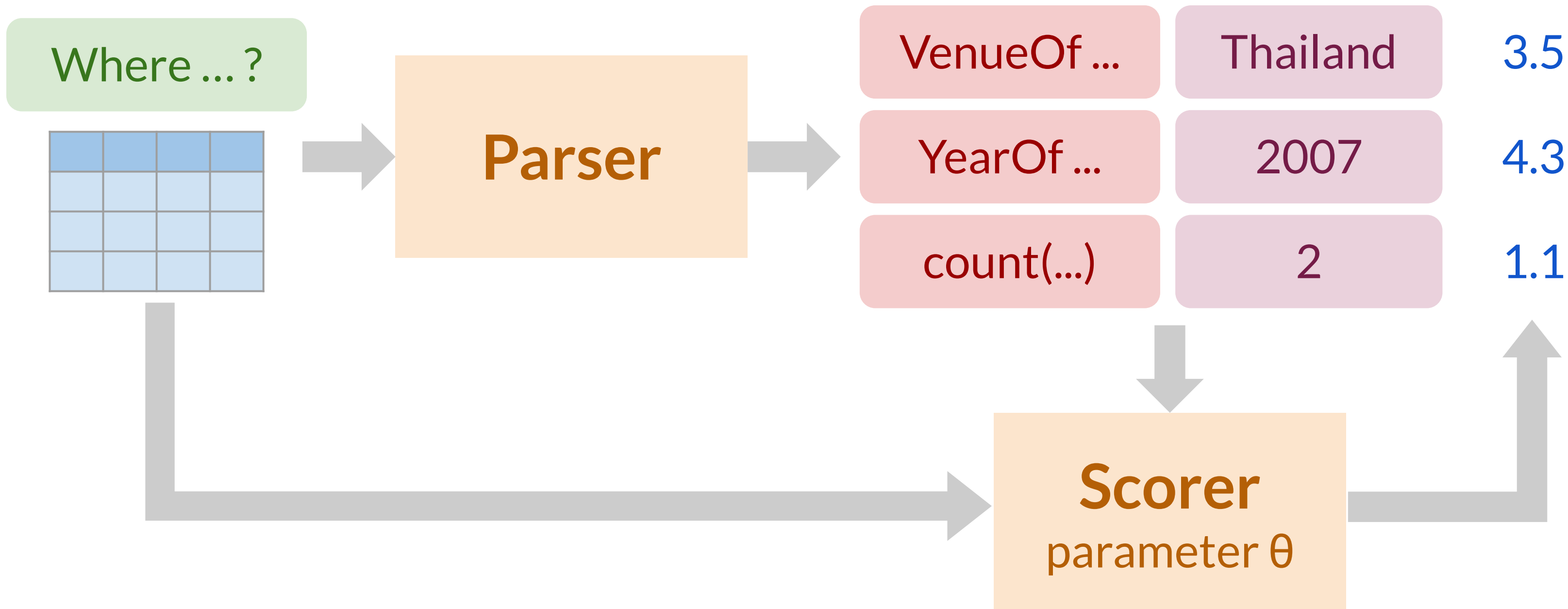

[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Prediction




[Zelle + Mooney, 1996 / Zettlemoyer + Collins, 2007 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Learning from Denotations



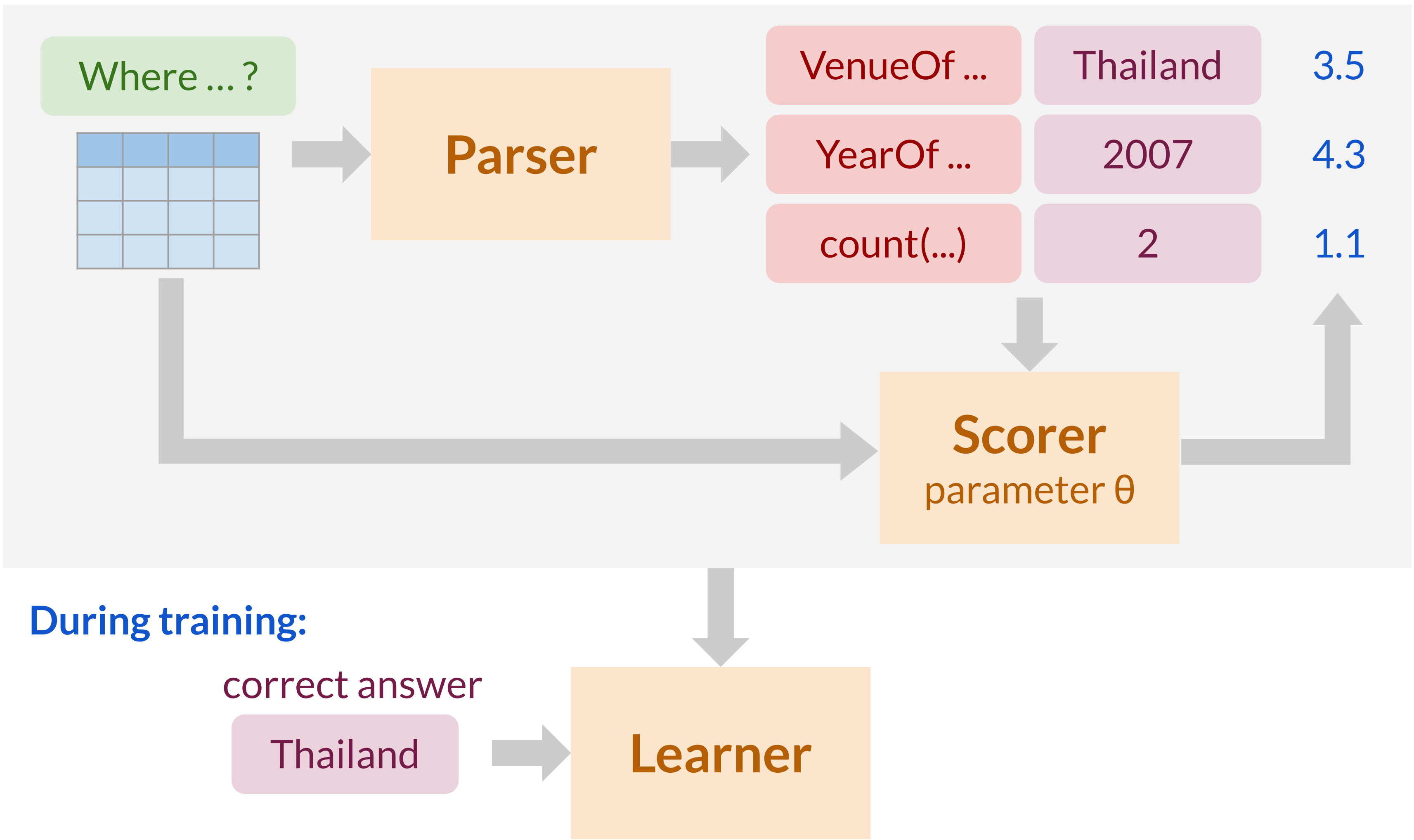

**During training:**

correct answer

Thailand

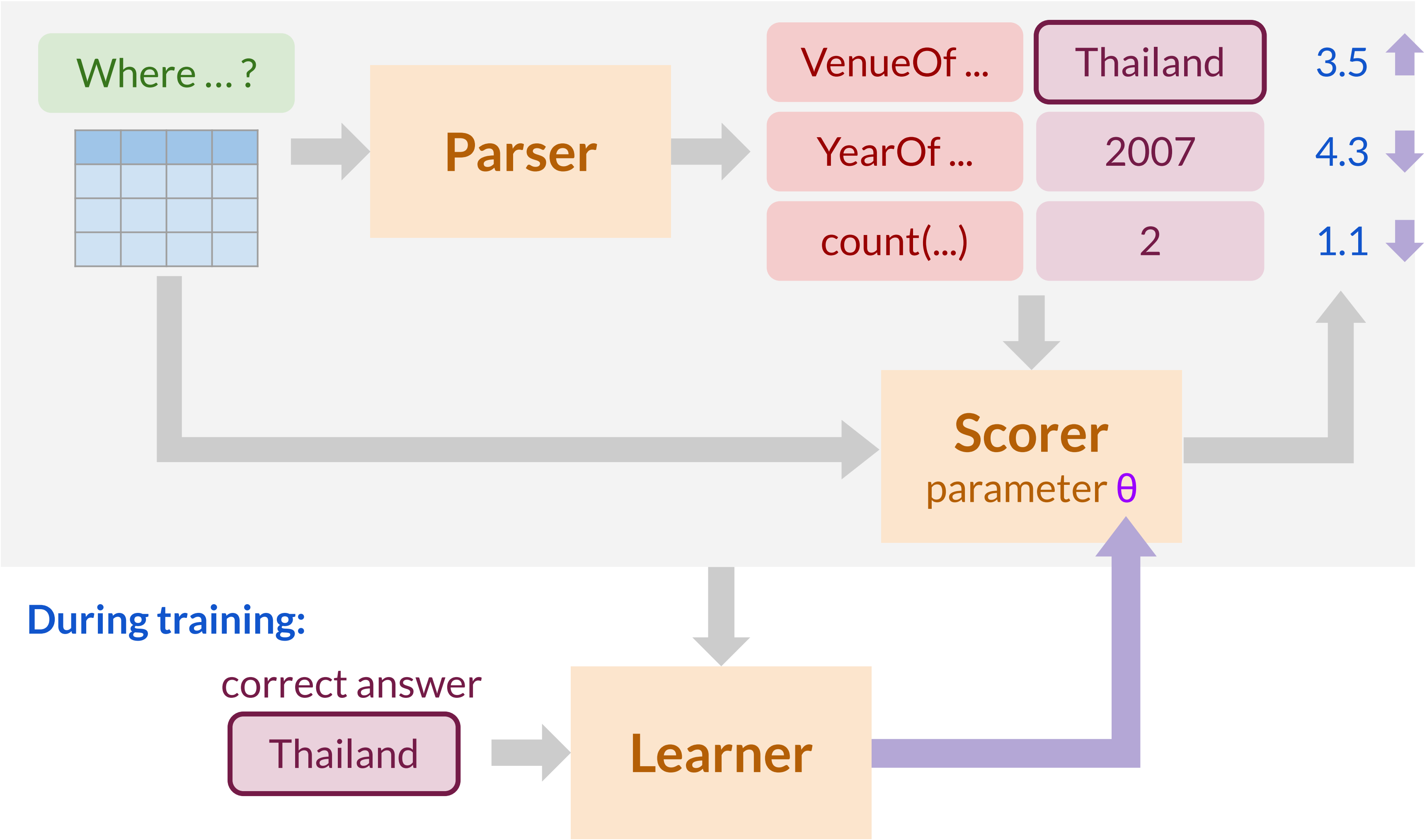
[Clark et al., 2010 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Learning from Denotations



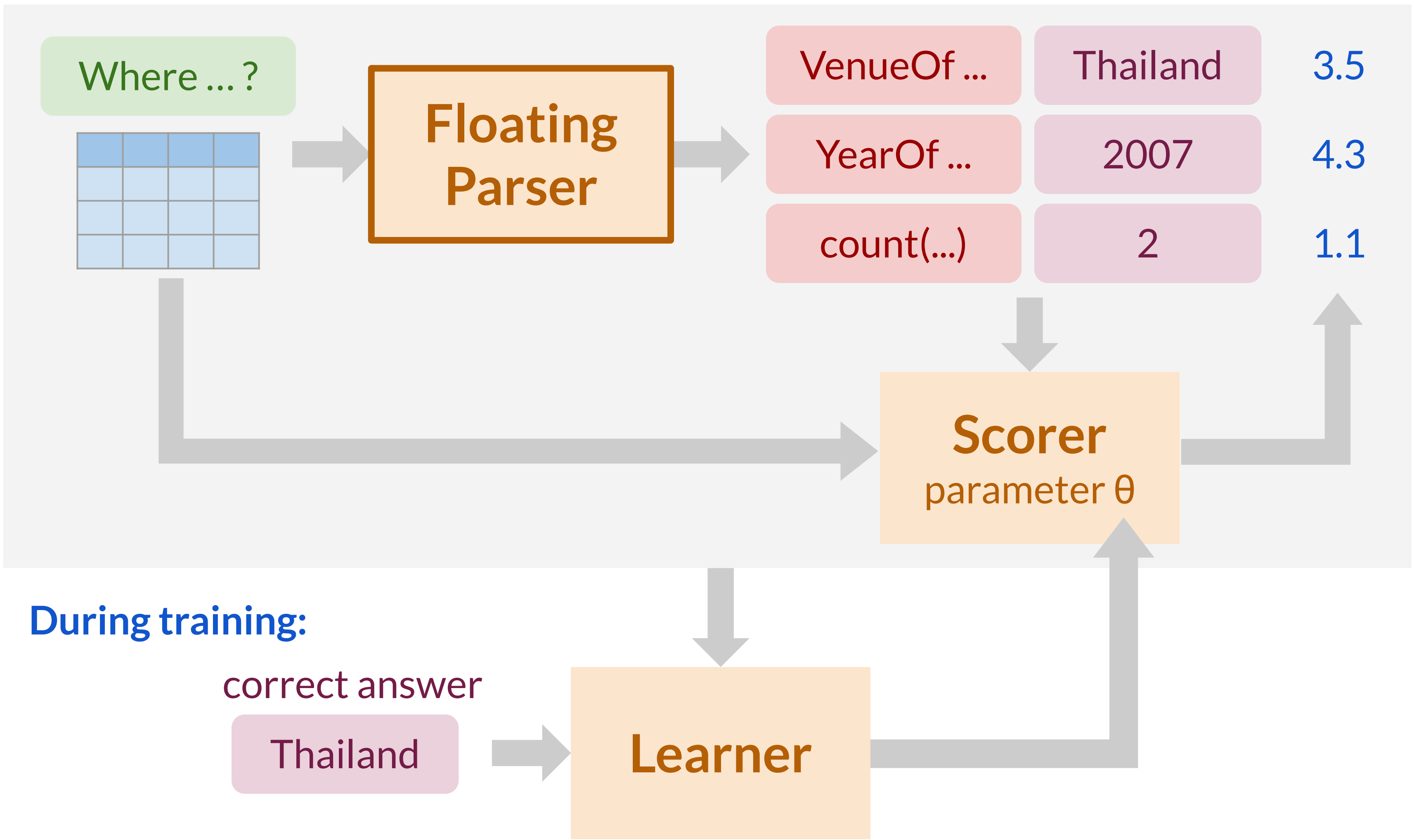

[Clark et al., 2010 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Learning from Denotations




[Clark et al., 2010 / Liang et al., 2011 / Berant et al., 2013 / ...]

# Parser




[P. and Liang, 2015]

# Parser

The parser **generates logical forms** from the utterance and table

VenueOf.argmax(HasPosition.1st, Index)

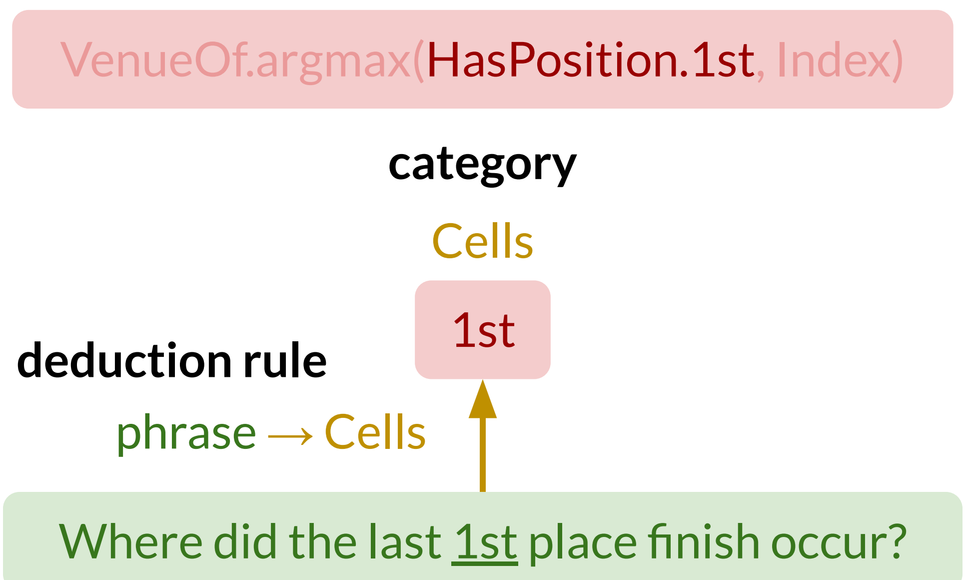
| Year | Venue | Position | Time |
| --- | --- | --- | --- |
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

Where did the last 1st place finish occur?

[P. and Liang, 2015]

# Parser

The parser **generates logical forms** from the utterance and table




| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2015]

# Parser

The parser **generates logical forms** from the utterance and table

VenueOf.argmax(HasPosition.1st, Index)

| Cells | Col |
|---|---|
| 1st | Position |

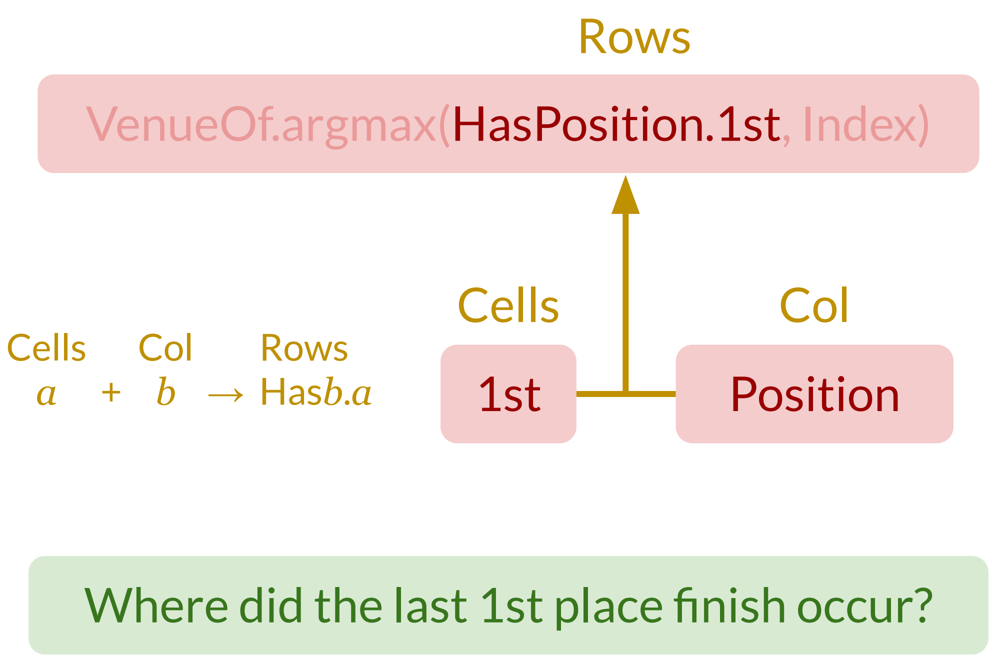
Where did the last 1st place finish occur?

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2015]

# Parser

The parser **generates logical forms** from the utterance and table

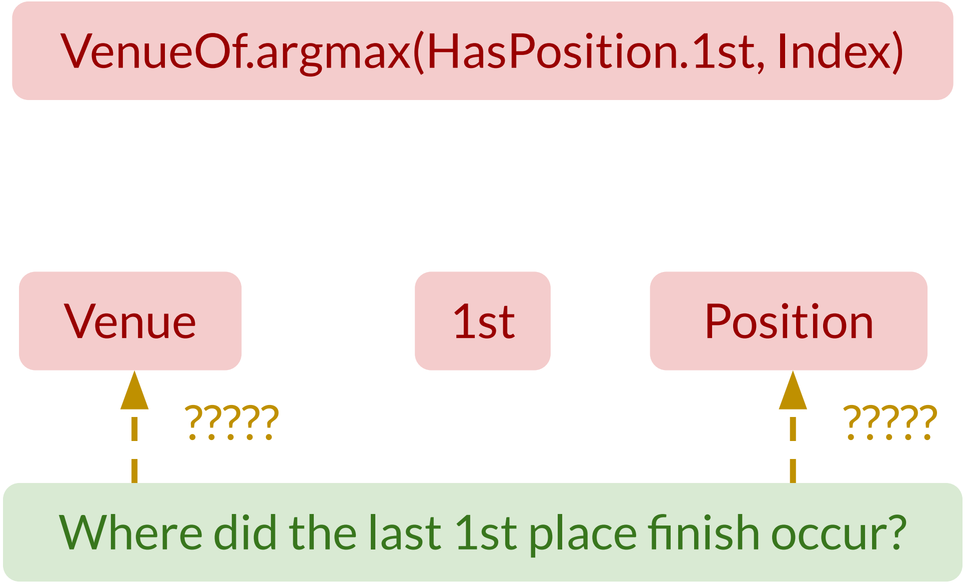



| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2015]

# Parser

The parser **generates logical forms** from the utterance and table

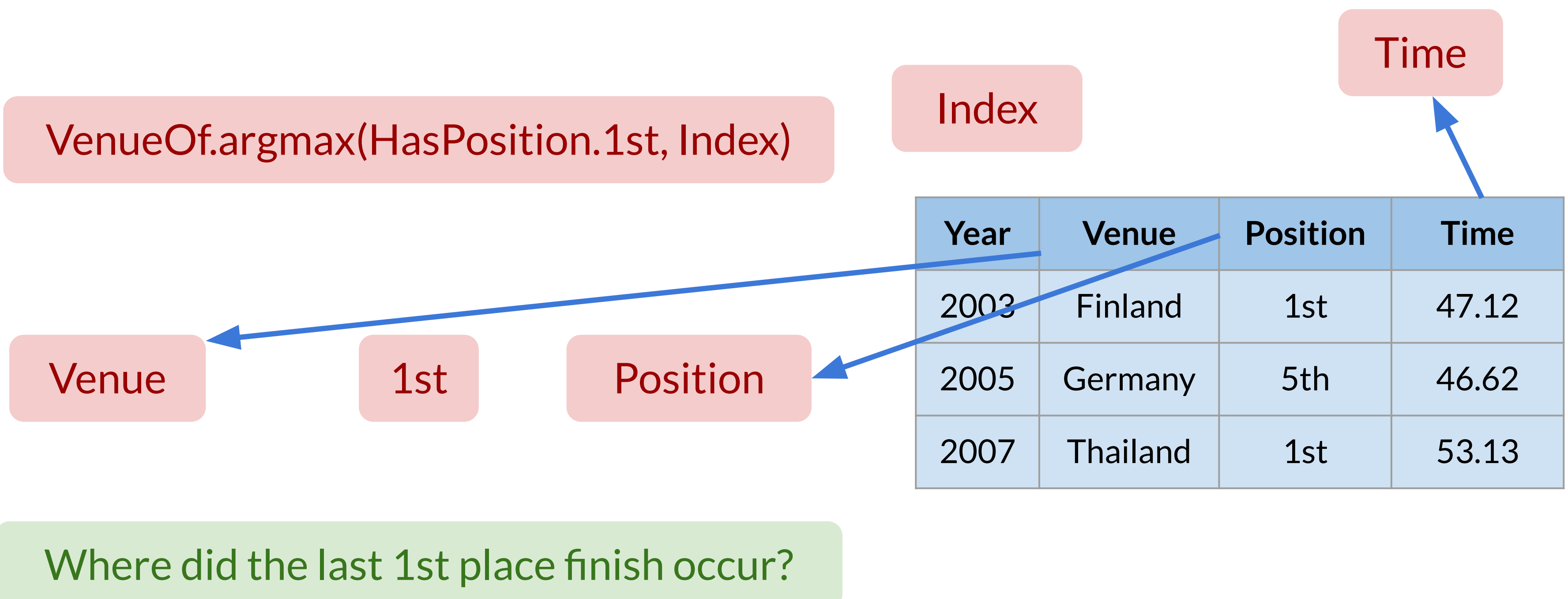



| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2015]

# Floating Parser

Allows LF parts to be **generated from other sources** than the utterance



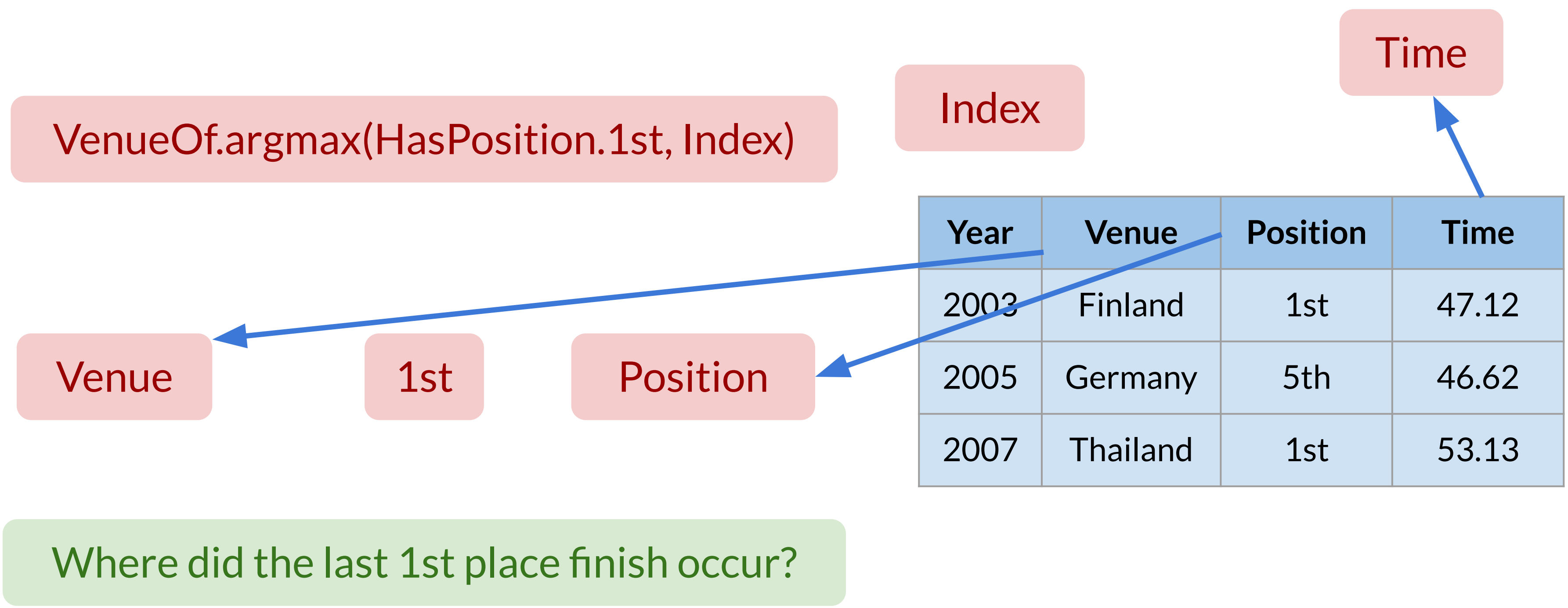

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2015]

# Floating Parser

Allows LF parts to be **generated from other sources** than the utterance

The **scorer** is responsible for capturing the relationship between "Venue" and words in the utterance

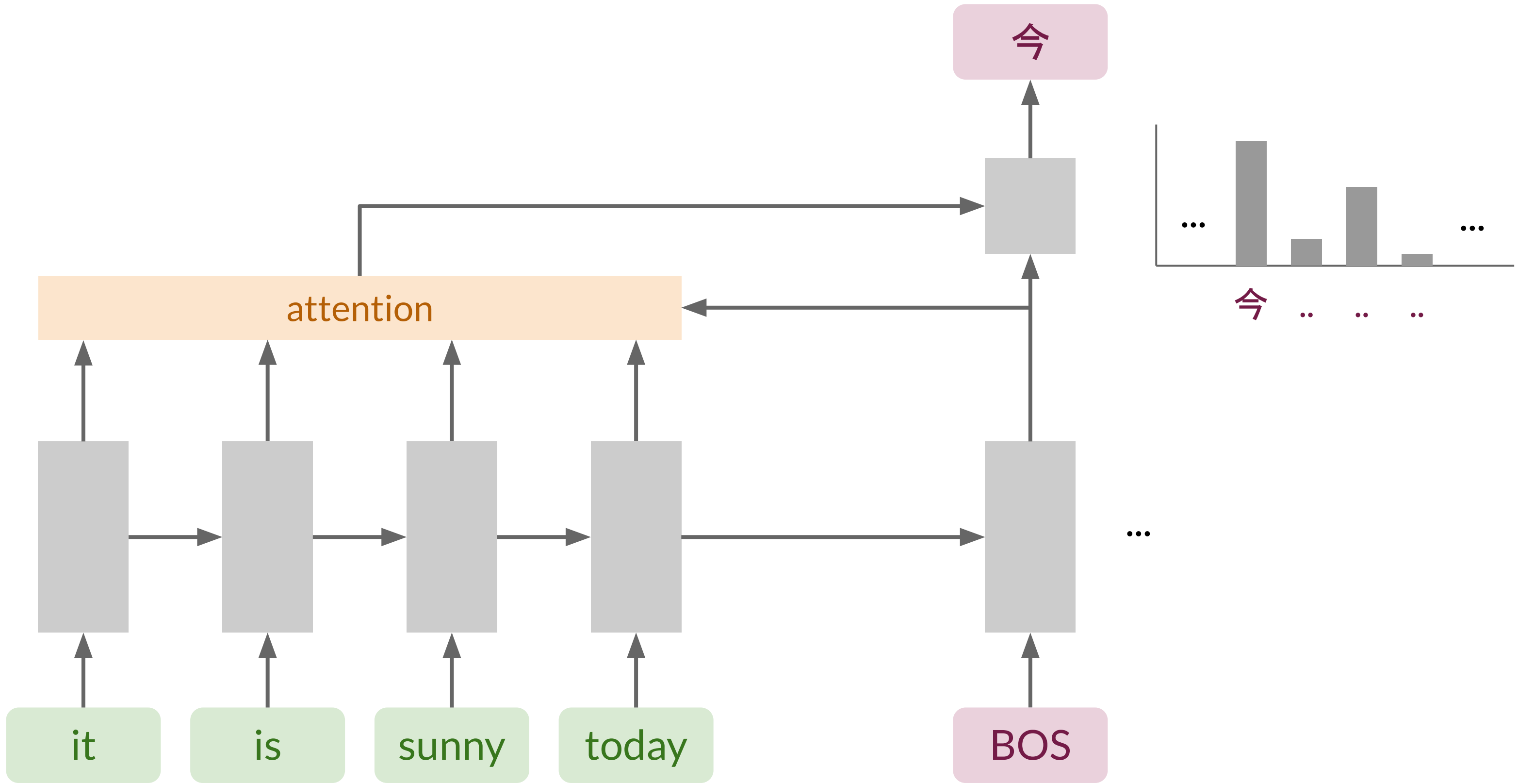



| Year | Venue | Position | Time |
| --- | --- | --- | --- |
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2015]

# Floating Parser

Most neural decoders today generate tokens in a floating fashion
(with a soft guidance from attention)



# Results

| | Test Accuracy |
|---|---|
| Tiny rule set<br>(cell lookup + counting) | 24.3 |
| Small rule set<br>(sum, argmax, next/prev row, subtraction, etc.) | 37.1 |
| Large rule set<br>(fuzzy string matching, advanced argmax, etc.) | 42.7 |

[P. and Liang, 2015]

# Results

| | Test Accuracy |
|---|---|
| Tiny rule set<br>(cell lookup + counting) | 24.3 |
| Small rule set<br>(sum, argmax, next/prev row, subtraction, etc.) | 37.1 |
| Large rule set<br>(fuzzy string matching, advanced argmax, etc.) | 42.7 |
| Upper bound<br>(Ice writes a logical form) | (84.0) |

[P. and Liang, 2015]

# Results

| | Test Accuracy | Runtime |
|---|---|---|
| Tiny rule set<br>(cell lookup + counting) | 24.3 | |
| Small rule set<br>(sum, argmax, next/prev row, subtraction, etc.) | 37.1 | ~ 4 hours |
| Large rule set<br>(fuzzy string matching, advanced argmax, etc.) | 42.7 | ~ 11 hours |
| Upper bound<br>(Ice writes a logical form) | (84.0) | |

Slow runtime prevents us from increasing coverage!

[P. and Liang, 2015]

# Outline

- Motivation
- Task: Answering complex questions on web tables
- Approach: Semantic parsing with distant supervision
- **Improvement 1: Make it faster**
- Improvement 2: Convert to direct supervision

# Using macros to make search faster

For each training example:

Who took office right after Uriah Forrest?

[Yuchen Zhang, P. and Liang, 2017]

# Using macros to make search faster

For each training example:

- Fetch previously processed examples with **similar utterances** (Levenshtein distance)

Who took office right after Uriah Forrest?

Who ranked right after Turkey?

NationOf.NextOf.HasNation.Turkey

[Yuchen Zhang, P. and Liang, 2017]

# Using macros to make search faster

For each training example:

- Fetch previously processed examples with **similar utterances** (Levenshtein distance)

Who took office right after Uriah Forrest?

**macro**

Column Of.NextOf.Has Column . Cell

[Yuchen Zhang, P. and Liang, 2017]

# Using macros to make search faster

For each training example:

- Fetch previously processed examples with **similar utterances**
- During search, try creating logical forms from **macros** found in those examples first

Who took office right after Uriah Forrest?

**macro**

Column Of.NextOf.Has Column . Cell

NameOf.NextOf.HasName.UriahForrest

BossOf.NextOf.HasBoss.UriahForrest

[Yuchen Zhang, P. and Liang, 2017]

# Using macros to make search faster

| | Dev Accuracy | Time (ms/example) | |
|---|---|---|---|
| | | Train | Predict |
| Large rule set | **40.6** | 1117 | 1150 |
| + Macros | 40.4 | **99** | **70** |

11× speedup! 16× speedup!

[Yuchen Zhang, P. and Liang, 2017]

# Using macros to make search faster

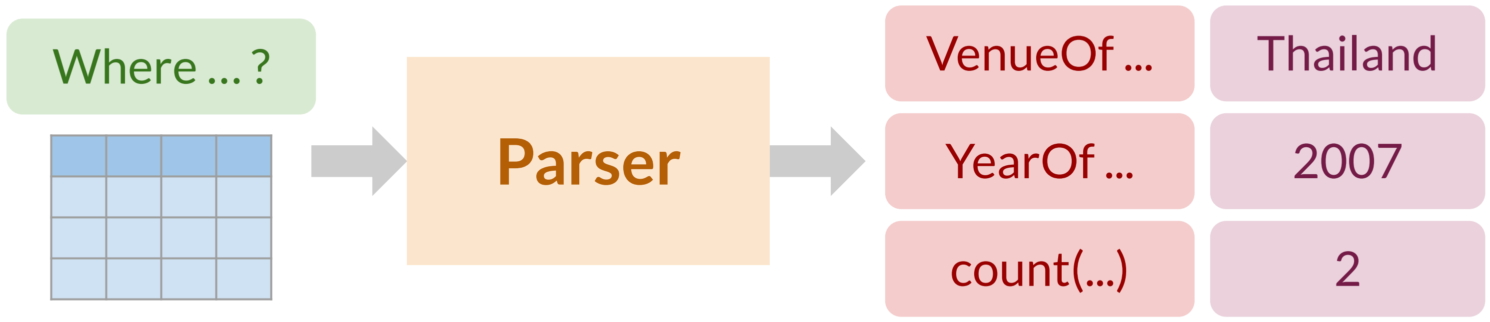
| | Test Acc. | +Ensemble |
|---|---|---|
| Small rule set | 37.1 | - |
| Neural Programmer<br>(Neelakantan et al., 2016) | 34.2 | 37.7 |
| Neural Multi-Step Reasoning<br>(Haug et al., 2017) | 34.8 | 38.7 |
| Large rule set | 42.7 | - |
| + Macros | **43.7** | - |
| Upper bound<br>(Ice writes a logical form) | (84.0) | - |

[Yuchen Zhang, P. and Liang, 2017]

# Outline

- Motivation
- Task: Answering complex questions on web tables
- Approach: Semantic parsing with distant supervision
- Improvement 1: Make it faster
- **Improvement 2: Convert to direct supervision**

# Learning from Denotations

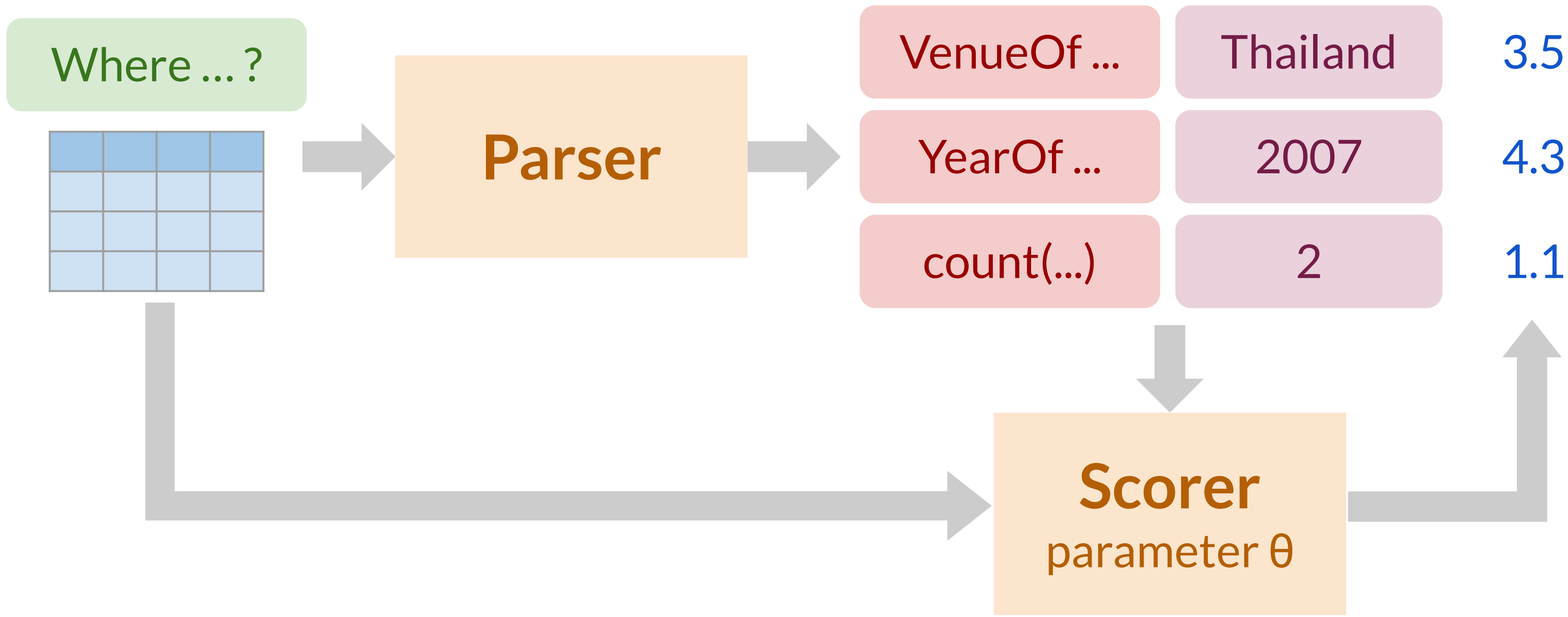



**During training:**

correct answer

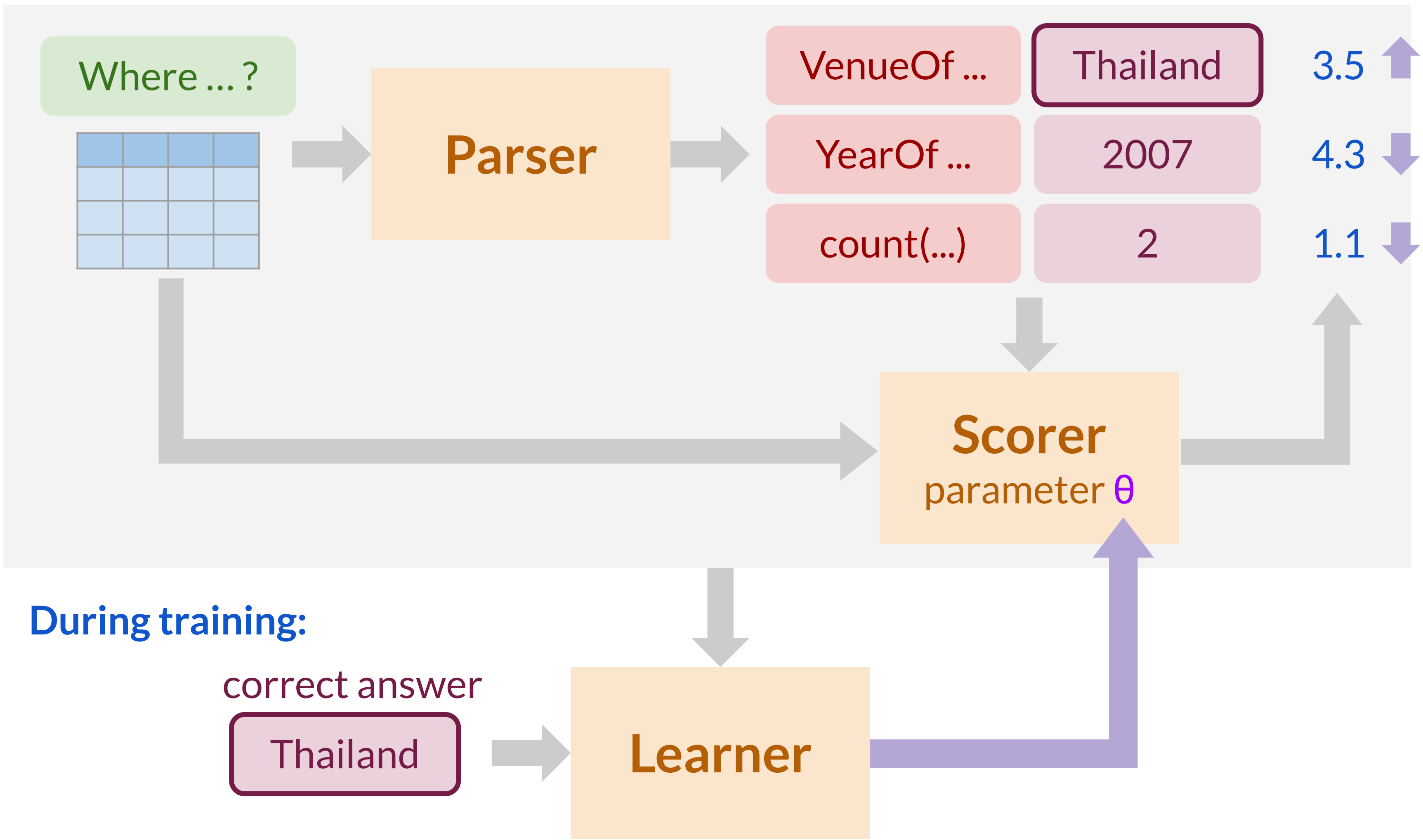
Thailand

# Learning from Denotations




**During training:**

# Learning from Denotations



# Learning from Denotations

**Pros:**

- Collecting answers is easier than collecting LFs for each question
- The dataset is not tied to a specific LF formalism

# Learning from Denotations

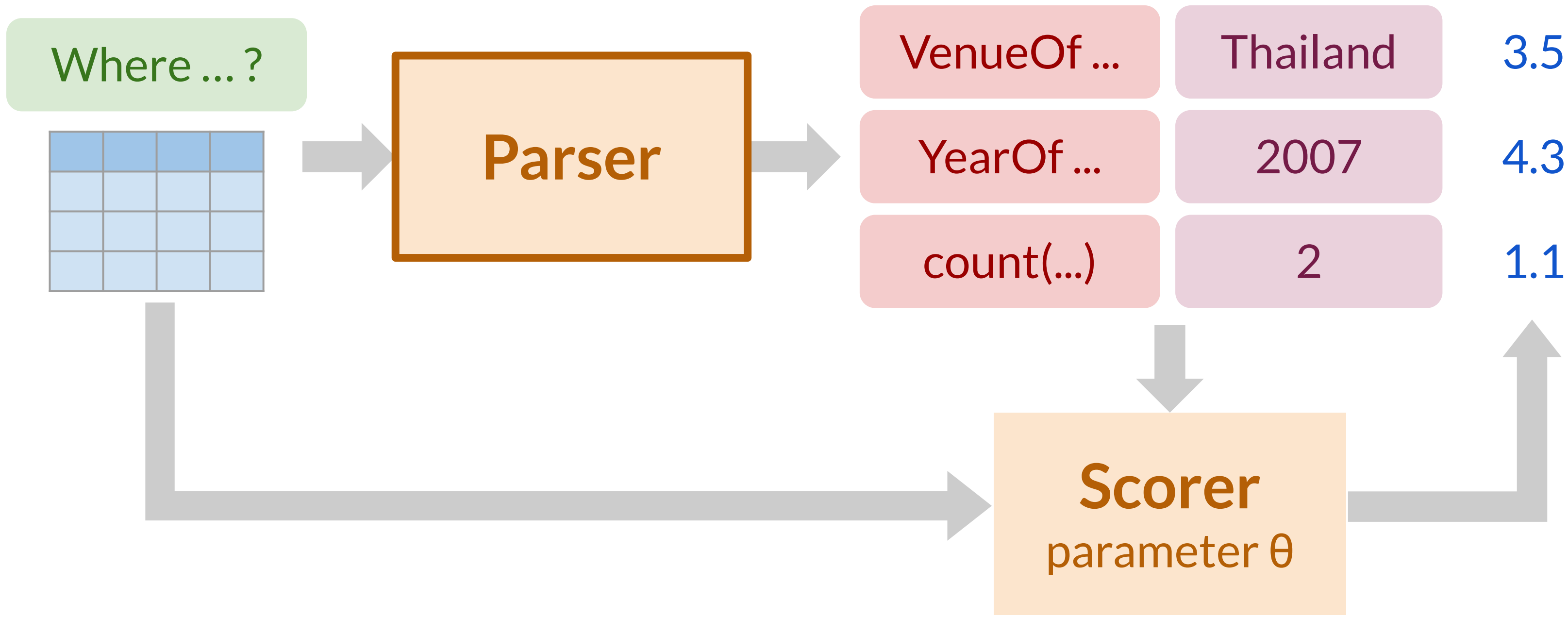
**Pros:**

- Collecting answers is easier than collecting LFs for each question
- The dataset is not tied to a specific LF formalism

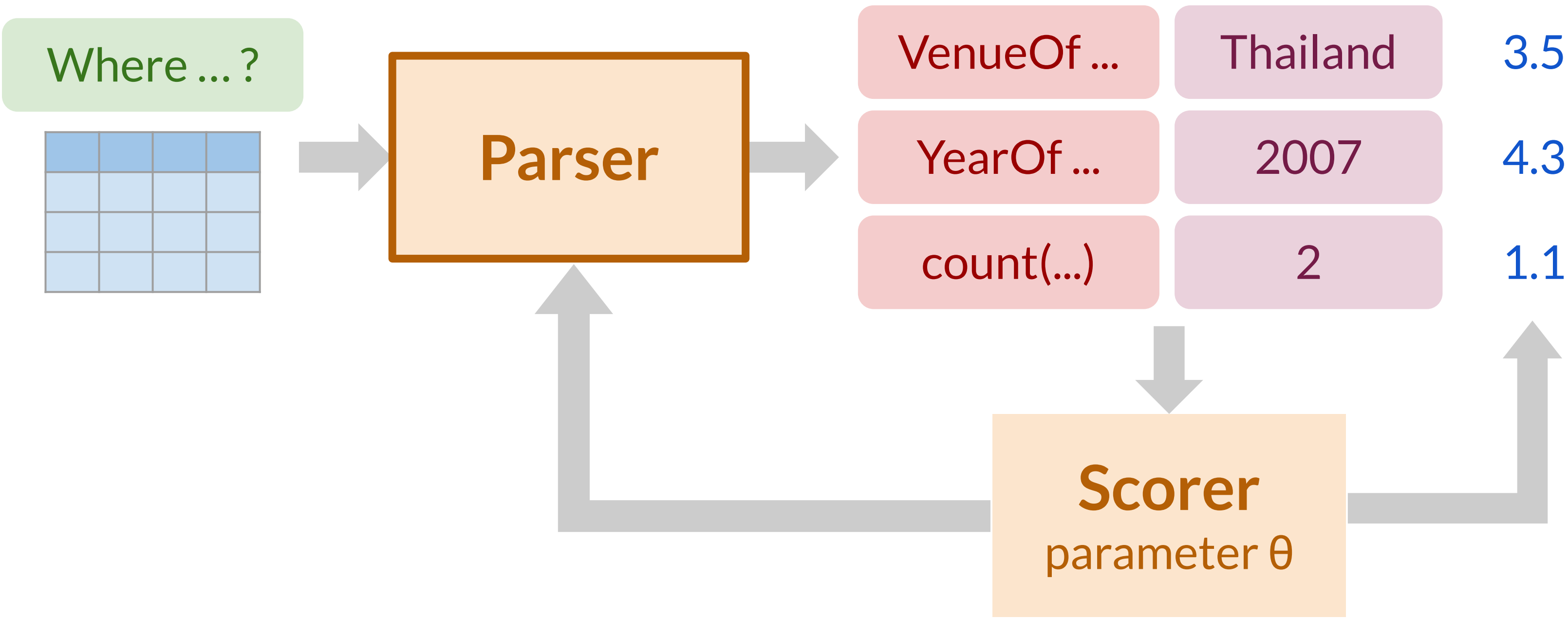
**Cons:**

- Need to do **search** (both at training and test time)
  - Slow (macros helped a bit)
  - **Two more problems during training:** ...

# Search is Hard



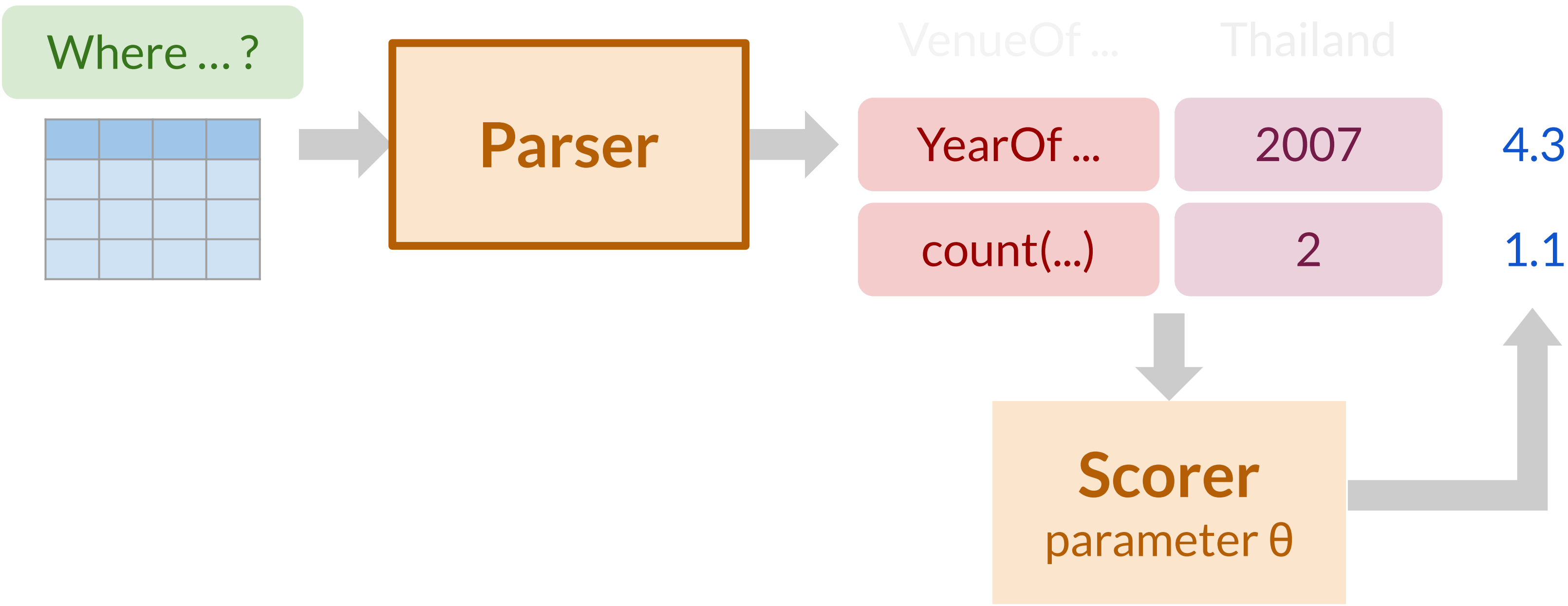
# Search is Hard



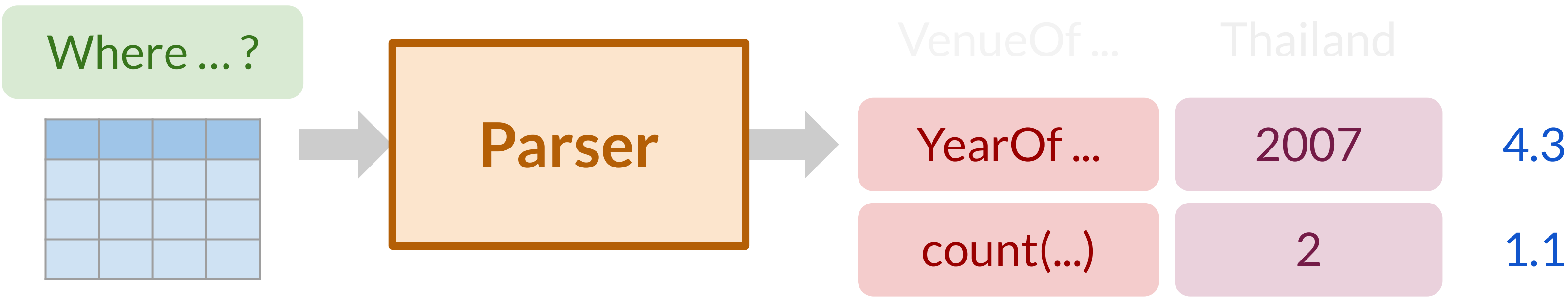

Parser uses **beam search** to make search tractible

- For each parsing state, only keep up to B = 100 highest-scoring partial LFs and **discard the rest**

# Search is Hard



# Search is Hard

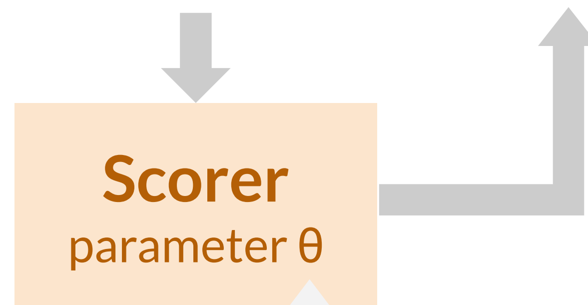


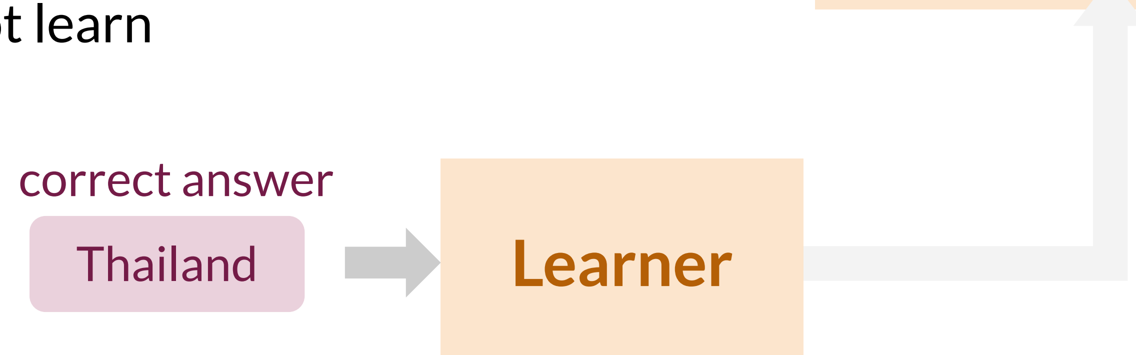

**Problem 1:**

If the parser cannot find a **consistent** LF

(consistent: denotation matches the correct answer)

→ cannot learn



# Search is Hard

Some LFs are **spurious** (right for wrong reasons)

Where did the last 1st place finish occur?

VenueOf.argmax(HasPosition.1st, Index)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

# Search is Hard

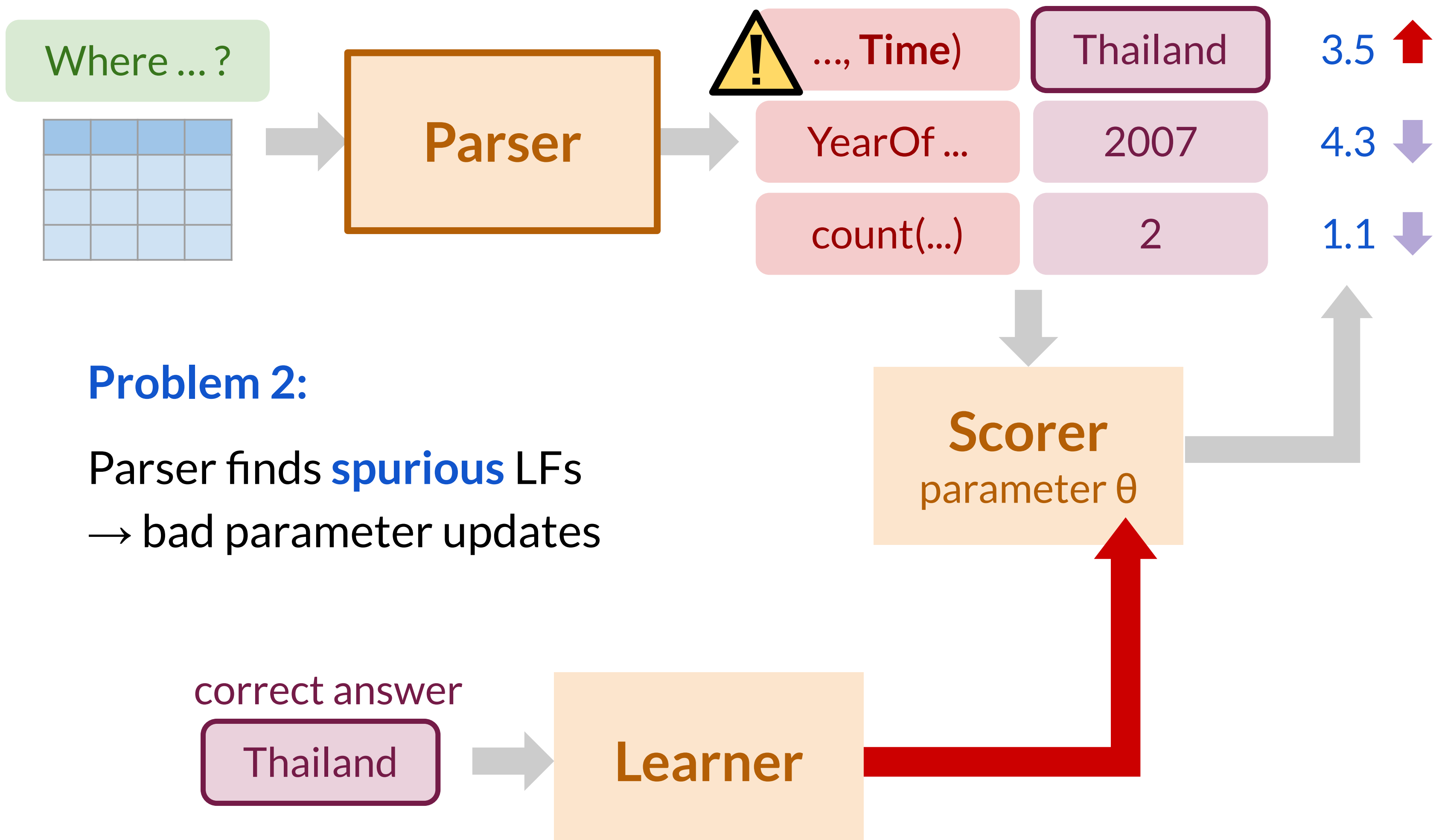
Some LFs are **spurious** (right for wrong reasons)

Where did the last 1st place finish occur?

⚠ VenueOf.argmax(HasPosition.1st, **Time**)

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | **53.13** |

# Search is Hard




**Problem 2:**

Parser finds **spurious** LFs
→ bad parameter updates

# Learning from Denotations

**Pros:**

- Collecting answers is easier than collecting LFs for each question
- The dataset is not tied to a specific LF formalism

**Cons:**

- Need to do **search**
  - Slow
  - Cannot find consistent LFs $\rightarrow$ cannot learn
  - Find spurious LFs $\rightarrow$ bad updates

# Offline search

Let's avoid search during training altogether!

[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, **answer**), augment it with LFs

- Use (utterance, table, **LFs**) for **supervised training**

[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with LFs
  - ▷
  - ▷
- Use (utterance, table, LFs) for supervised training

[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, **answer**), augment it with LFs
  - **Enumerate** all LFs consistent with the **correct answer**
  -
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with LFs
  - **Enumerate** all LFs consistent with the correct answer
  - 
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Enumerating consistent LFs

Beam search controls the search space, but can **discard crucial LF parts** $\rightarrow$ low coverage!

If we have access to the **correct answer**, is there a better way to control the search space?

[P. and Liang, 2016]

# Dynamic Programming on Denotations

| Year | Venue | Position | Time |
| --- | --- | --- | --- |
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2016]

# Dynamic Programming on Denotations

HasIndex.2

HasPosition.5th

HasVenue.Germany

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

$r_2$

[P. and Liang, 2016]

# Dynamic Programming on Denotations

TimeOf. HasIndex.2

TimeOf. HasPosition.5th

TimeOf. HasVenue.Germany

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2016]

# Dynamic Programming on Denotations

Group LFs with the same denotation together during search

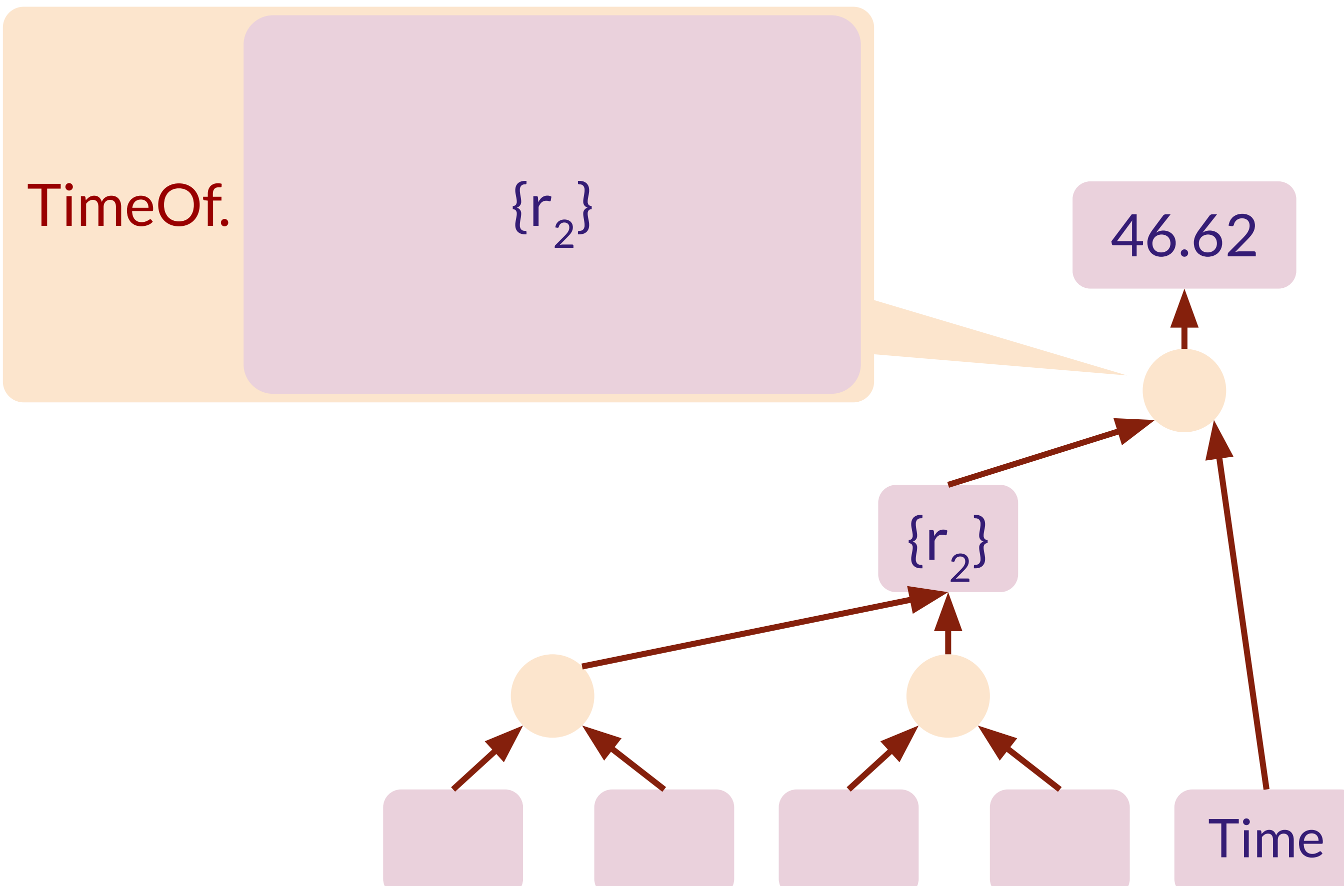
TimeOf. $\{r_2\}$

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

[P. and Liang, 2016]

# Dynamic Programming on Denotations

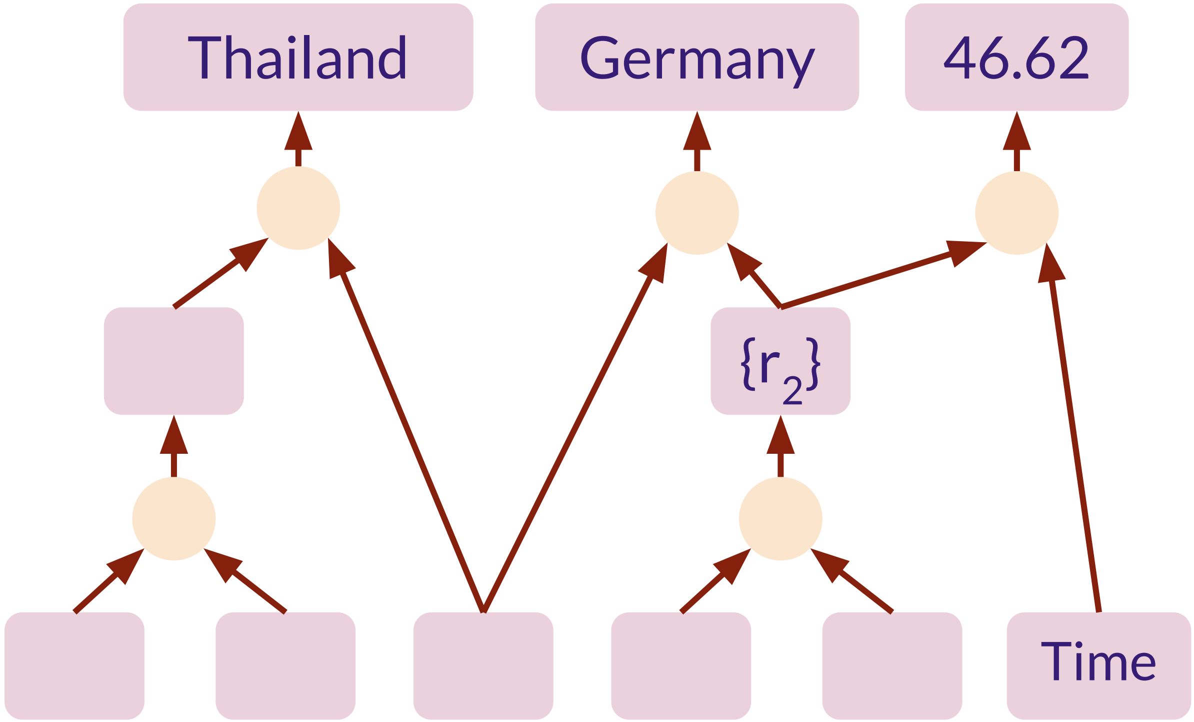
**Phase 1:** Group LFs with the same denotation together during search




[P. and Liang, 2016]

# Dynamic Programming on Denotations

**Phase 1:** Group LFs with the same denotation together during search

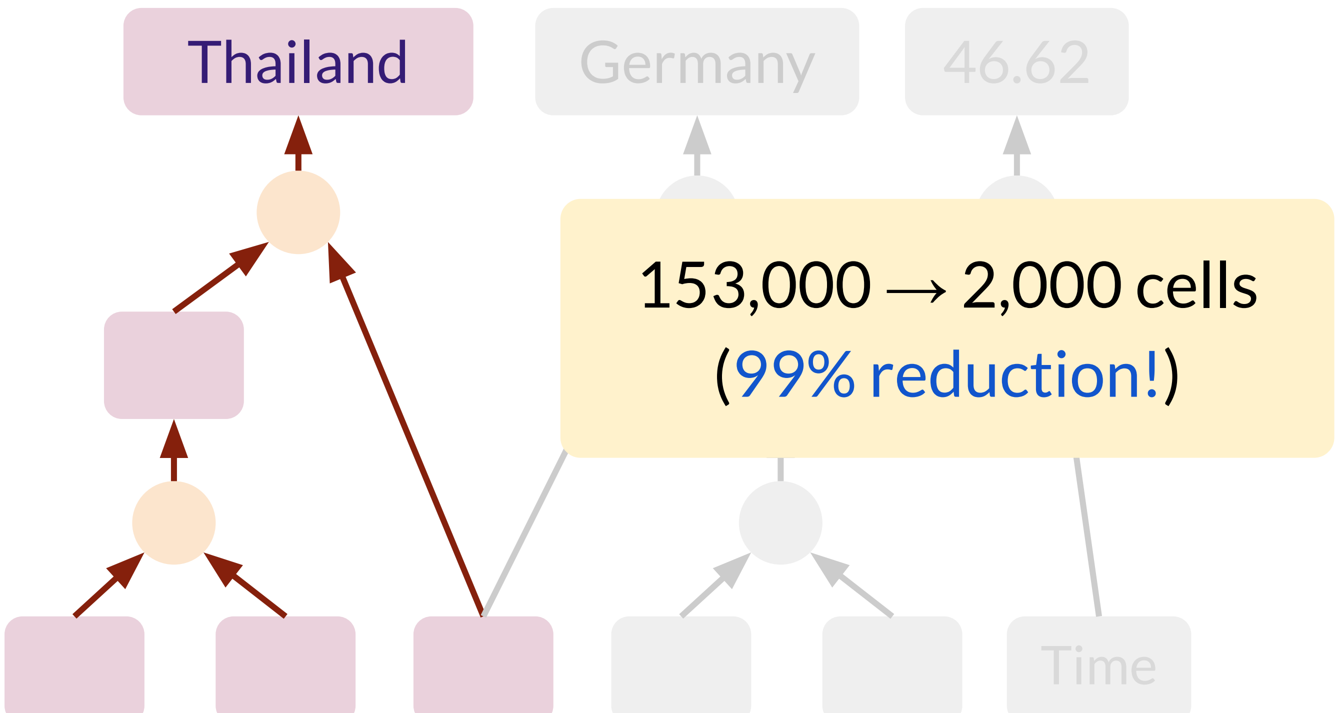



[P. and Liang, 2016]

# Dynamic Programming on Denotations

**Phase 1:** Group LFs with the same denotation together during search

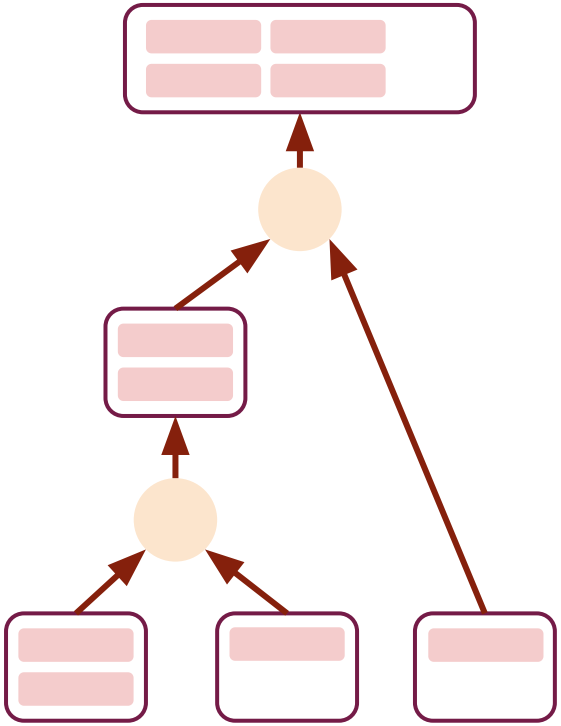
- Remove paths that do not give the **correct answer**




[P. and Liang, 2016]

# Dynamic Programming on Denotations

**Phase 2:** Search over LFs, but only on the paths found in Phase 1

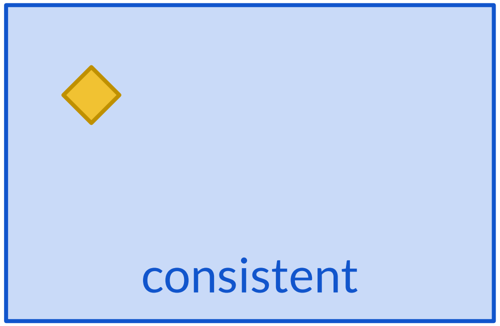


[P. and Liang, 2016]

# Results

## Testing coverage:

- Annotate each example with a semantically correct LF
- Test whether the algorithm can generate the annotated LF



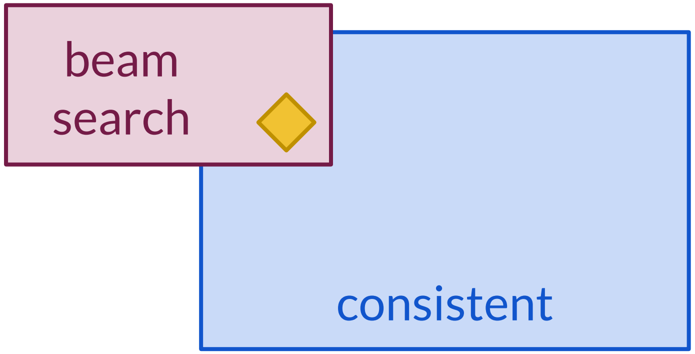

**Note:** Maximum success rate = 84.0% (% of examples with annotated LFs)

[P. and Liang, 2016]

# Results

**Testing coverage:**

- Annotate each example with a semantically correct LF
- Test whether the algorithm can generate the annotated LF




Success rate: 53.7%

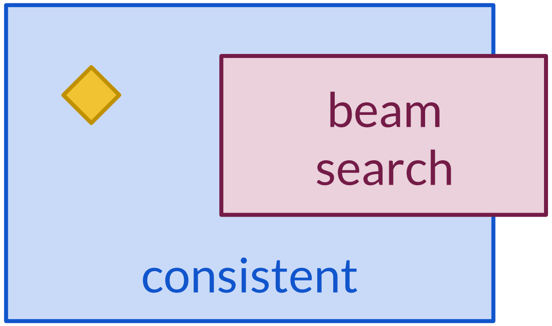
**Note:** Maximum success rate = 84.0% (% of examples with annotated LFs)

[P. and Liang, 2016]

# Results

## Testing coverage:

- Annotate each example with a semantically correct LF
- Test whether the algorithm can generate the annotated LF




Success rate: 53.7%

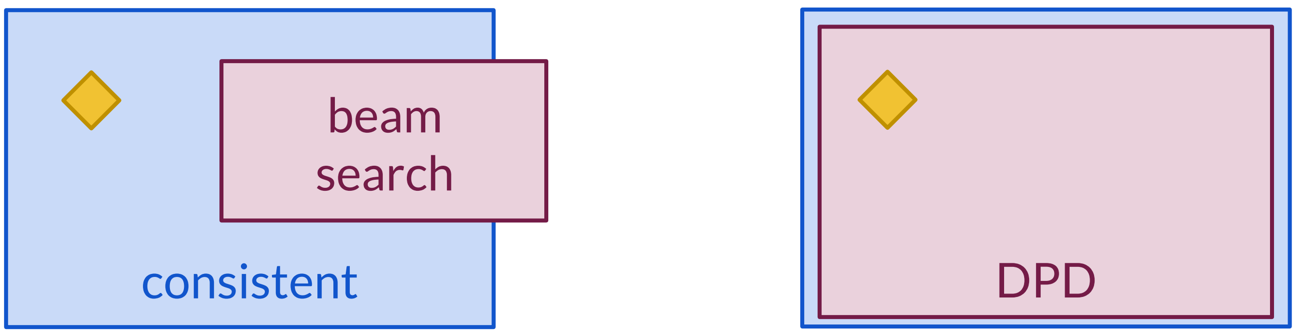
**Note:** Maximum success rate = 84.0% (% of examples with annotated LFs)

[P. and Liang, 2016]

# Results

**Testing coverage:**

- Annotate each example with a semantically correct LF
- Test whether the algorithm can generate the annotated LF




Success rate: 53.7%

Success rate: 76.0%

**Note:** Maximum success rate = 84.0% (% of examples with annotated LFs)

[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Filtering out Spurious LFs

| Year | Venue | Position | Time |
| --- | --- | --- | --- |
| 2003 | Finland | 1st | 47.12 |
| 2005 | Germany | 5th | 46.62 |
| 2007 | Thailand | 1st | 53.13 |

Where did the last 1st place finish occur?

| | |
| --- | --- |
| Human | Thailand |
| Correct LF | Thailand |
| Spurious LF | Thailand |

[P. and Liang, 2016]

# Filtering out Spurious LFs

| Year | Venue | Position | Time |
|---|---|---|---|
| 2003 | Germany | 1st | 53.13 |
| 2005 | Finland | 1st | 47.12 |
| 2007 | Thailand | 5th | 46.62 |

Randomize cells in each column

Where did the last 1st place finish occur?

| | |
|---|---|
| Human | Finland |
| Correct LF | Finland |
| Spurious LF | Germany |

[P. and Liang, 2016]

# Fictitious Tables

Generate **fictitious tables** and execute the logical forms on them

| | Original Table |
|---|---|
| LF 1 | Thailand |
| LF 2 | Thailand |
| LF 3 | Thailand |
| LF 4 | Thailand |
| LF 5 | Thailand |
| Human | Thailand |

[P. and Liang, 2016]

# Fictitious Tables

Generate **fictitious tables** and execute the logical forms on them

| | Original Table | Fictitious Table 1 |
|---|---|---|
| LF 1 | Thailand | Finland |
| LF 2 | Thailand | Thailand |
| LF 3 | Thailand | Finland |
| LF 4 | Thailand | Finland |
| LF 5 | Thailand | Germany |
| Human | Thailand | Finland |

[P. and Liang, 2016]

# Fictitious Tables

Generate **fictitious tables** and execute the logical forms on them

| | Original Table | Fictitious Table 1 |
|---|---|---|
| LF 1 | Thailand | Finland |
| ⚠ ✖ LF 2 | Thailand | Thailand |
| LF 3 | Thailand | Finland |
| LF 4 | Thailand | Finland |
| ⚠ ✖ LF 5 | Thailand | Germany |
| Human | Thailand | Finland |

[P. and Liang, 2016]

# Fictitious Tables

Generate **fictitious tables** and execute the logical forms on them

We also propose a way to select the **most informative** fictitious tables

| | Original Table | Fictitious Table 1 | Fictitious Table 2 |
|---|---|---|---|
| LF 1 | Thailand | Finland | Germany |
| ⚠ ✖ LF 2 | Thailand | Thailand | Germany |
| ⚠ ✖ LF 3 | Thailand | Finland | Thailand |
| LF 4 | Thailand | Finland | Germany |
| ⚠ ✖ LF 5 | Thailand | Germany | Finland |
| Human | Thailand | Finland | Germany |

[P. and Liang, 2016]

# Fictitious Tables

**Results:**

- Accidentally pruned correct LFs in 20% of the examples
  - because randomizing cells can create nonsensical tables
- But for the remaining examples, pruned out 92.1% of spurious LFs

[P. and Liang, 2016]

# Fictitious Tables

**Relax assumption:** Only filter LFs disagreeing with humans > once

- Accidentally pruned correct LFs in ~~20%~~ 3% of the examples
  - because randomizing cells can create nonsensical tables
- But for the remaining examples, pruned out ~~92.1%~~ 78% of spurious LFs

[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Offline search

Let's avoid search during training altogether!

- For each training example (utterance, table, answer), augment it with **semantically correct** LFs
  - **Enumerate** all LFs consistent with the correct answer
  - **Filter out** spurious LFs
- Use (utterance, table, LFs) for supervised training


[P. and Liang, 2016]

# Using macros to make search faster

| | Test Acc. | +Ensemble |
|---|---|---|
| Small rule set | 37.1 | - |
| Neural Programmer (Neelakantan et al., 2016) | 34.2 | 37.7 |
| Neural Multi-Step Reasoning (Haug et al., 2017) | 34.8 | 38.7 |
| Large rule set | 42.7 | - |
| + Macros | 43.7 | - |

Krishnamurthy et al., 2017 uses 100 shortest LFs we generated to train a neural parser (top-down tree generation)

# Using macros to make search faster

| | Test Acc. | +Ensemble |
|---|---|---|
| Small rule set | 37.1 | - |
| Neural Programmer (Neelakantan et al., 2016) | 34.2 | 37.7 |
| Neural Multi-Step Reasoning (Haug et al., 2017) | 34.8 | 38.7 |
| Large rule set | 42.7 | - |
| + Macros | 43.7 | - |
| Neural Parser trained on **consistent** LFs (DPD; no filtering) | 36.3 | |

Krishnamurthy et al., 2017 uses 100 shortest LFs we generated to train a neural parser (top-down tree generation)

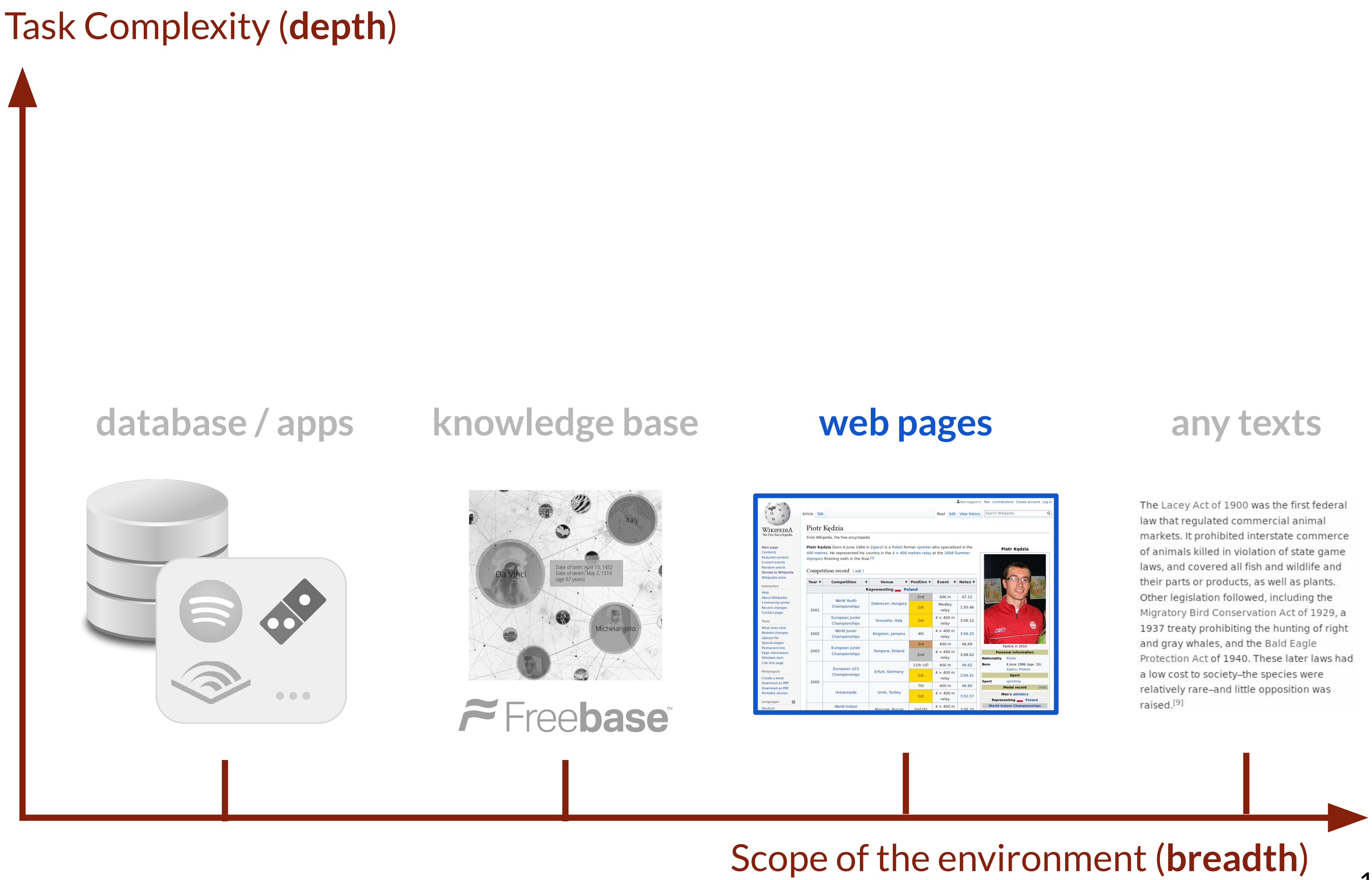
[Mudrakarta et al., 2018]

# Using macros to make search faster

| | Test Acc. | +Ensemble |
|---|---|---|
| Small rule set | 37.1 | - |
| Neural Programmer (Neelakantan et al., 2016) | 34.2 | 37.7 |
| Neural Multi-Step Reasoning (Haug et al., 2017) | 34.8 | 38.7 |
| Large rule set | 42.7 | - |
| + Macros | 43.7 | - |
| Neural Parser trained on **consistent** LFs (DPD; no filtering) | 36.3 | |
| Neural Parser trained on **correct** LFs (DPD + filtering) | 43.3 | 45.9 |

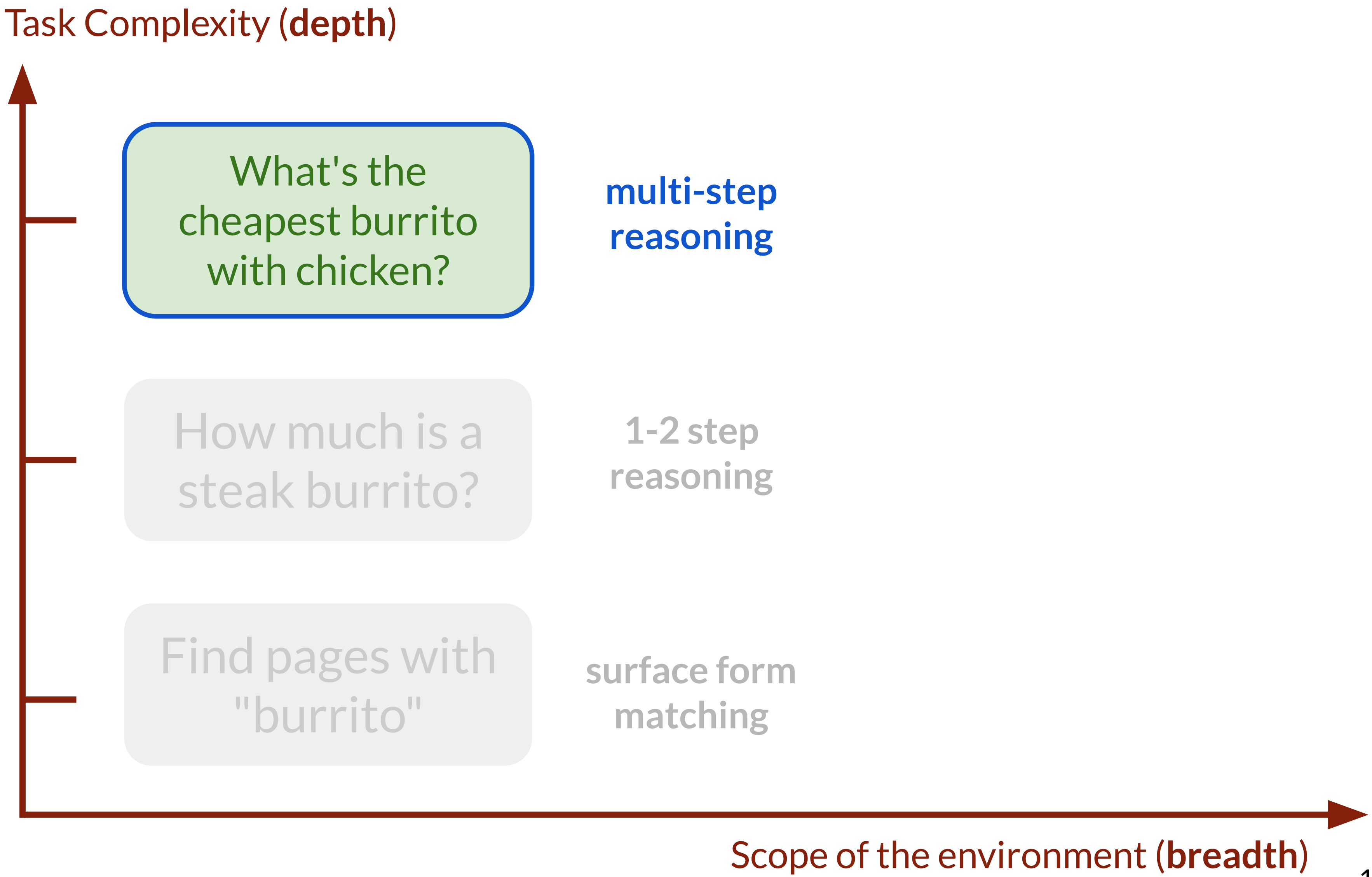
Krishnamurthy et al., 2017 uses 100 shortest LFs we generated to train a neural parser (top-down tree generation)

# Summary

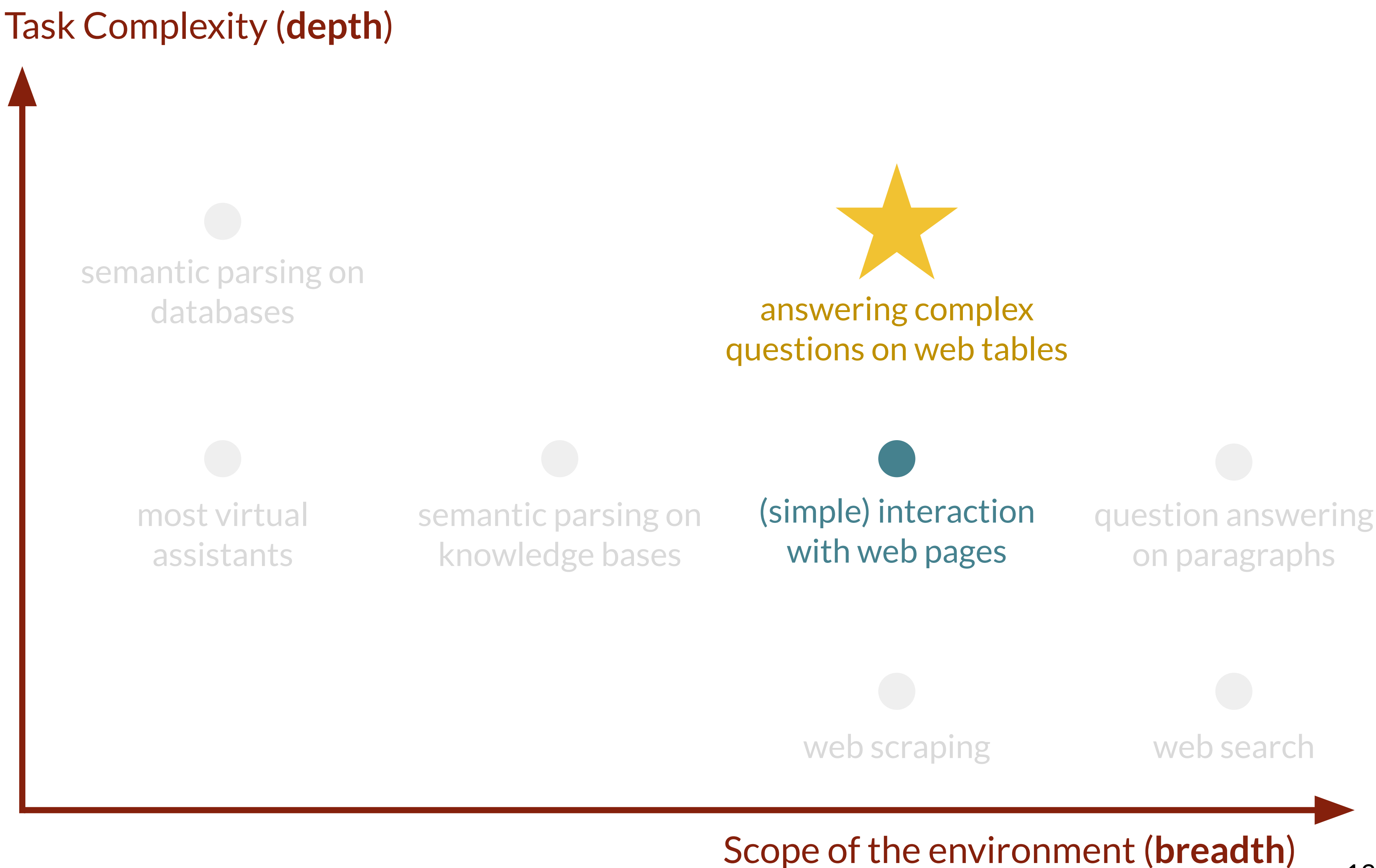


Task Complexity (depth)
database / apps
knowledge base
web pages
any texts
Freebase
The Lacey Act of 1900 was the first federal law that regulated commercial animal markets. It prohibited interstate commerce of animals killed in violation of state game laws, and covered all fish and wildlife and their parts or products, as well as plants. Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society–the species were relatively rare–and little opposition was raised.[9]
Scope of the environment (breadth)

# Summary

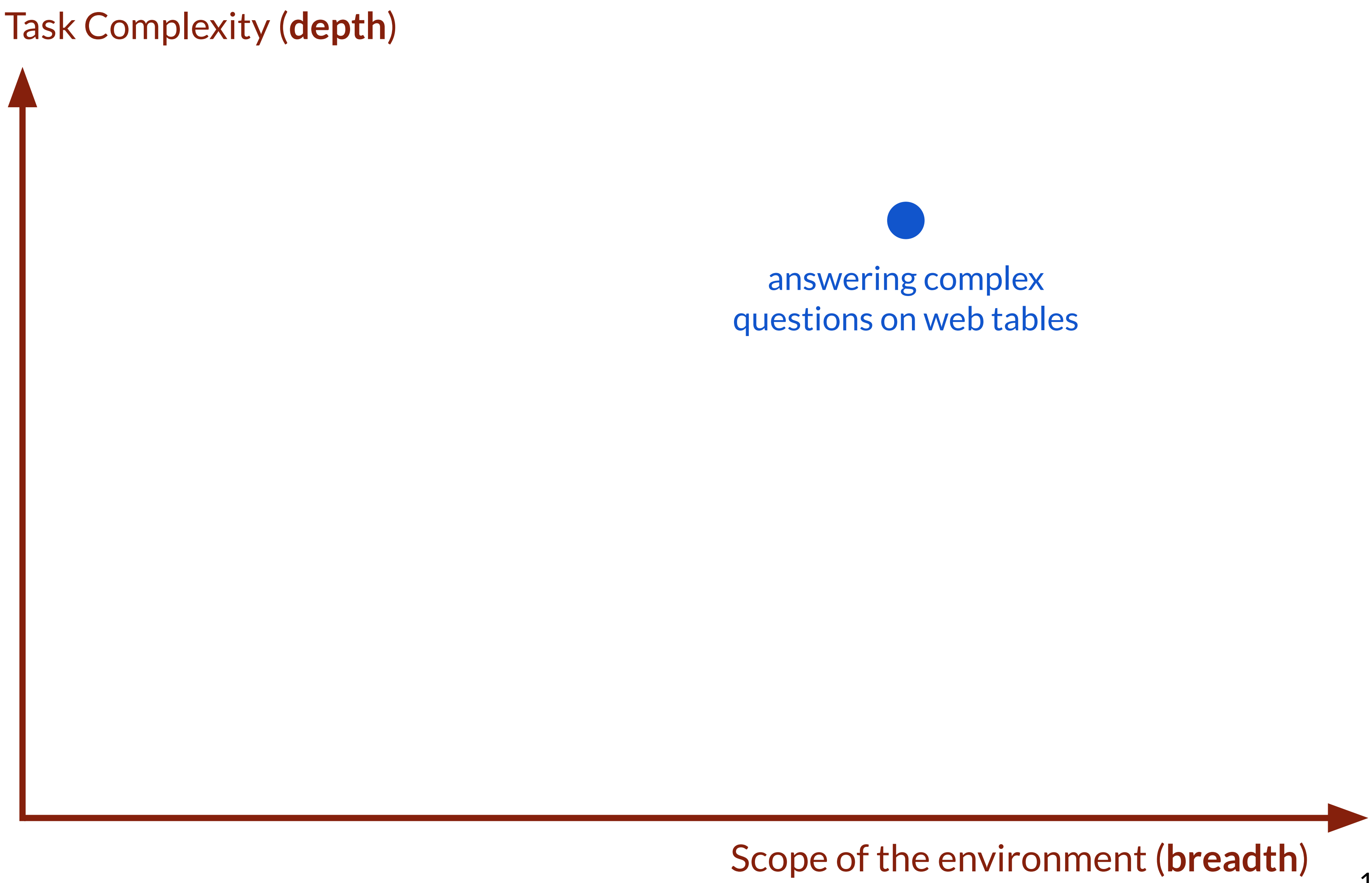

Task Complexity (**depth**)
What's the cheapest burrito with chicken?
**multi-step reasoning**
How much is a steak burrito?
**1-2 step reasoning**
Find pages with "burrito"
**surface form matching**
Scope of the environment (**breadth**)

# Summary



Task Complexity (**depth**)
semantic parsing on databases
answering complex questions on web tables
most virtual assistants
semantic parsing on knowledge bases
(simple) interaction with web pages
question answering on paragraphs
web scraping
web search
Scope of the environment (**breadth**)

# Future Directions



# Future Directions

Task Complexity (**depth**)

- Learn implicit columns
  ("What is ….?" → which column?)
- Detect if the question cannot be answered by the table
- More complex reasoning
  (e.g., "consecutive")
- Better table understanding
  (╯°□°)╯︵ ┻━┻
- ...

answering complex questions on web tables

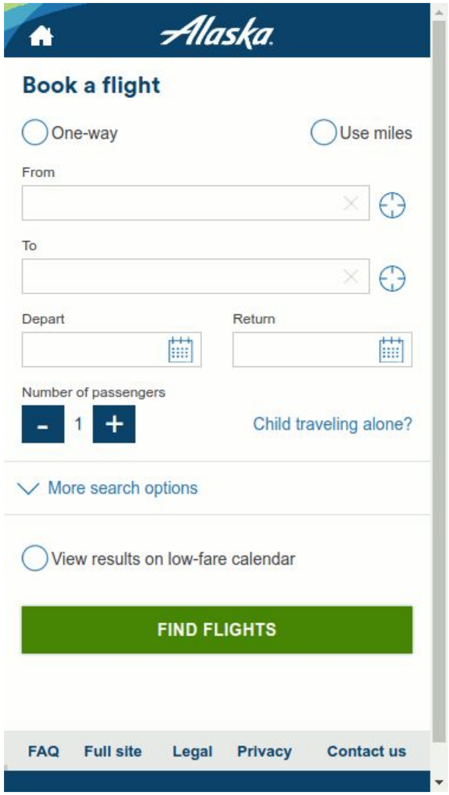
Scope of the environment (**breadth**)

[WikiSQL (Zhong et al., 2017) / SQUAD 2.0 (Rajpurkar et al., 2018) / Dong and Lapata, 2018 / ...]

# Future Directions

Task Complexity (**depth**)

multi-step web interaction




answering complex questions on web tables

Scope of the environment (**breadth**)

[PLOW (Allen et al., 2007) / Branavan et al., 2010 / World of Bits (Shi et al., 2017 / Liu et al., 2018 / Gur et al., 2019) / ...]

# Future Directions

Task Complexity (**depth**)

multi-step web interaction

combine information from multiple sources

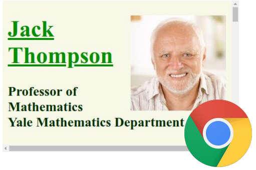
answering complex questions on web tables

gas station near Jack's office

which Jack?

office?

gas station?




Scope of the environment (**breadth**)


[QALD shared tasks / Spider (Yu et al., 2018) / HotpotQA (Yang et al., 2018) / DROP (Dua et al., 2019) / ...]

# Thank you!