Compositional Semantic Parsing on Semi-Structured Tables

ACL 2015
Panupong Pasupat and Percy Liang
Stanford University
説明 : 松田耕史 (東北大学)

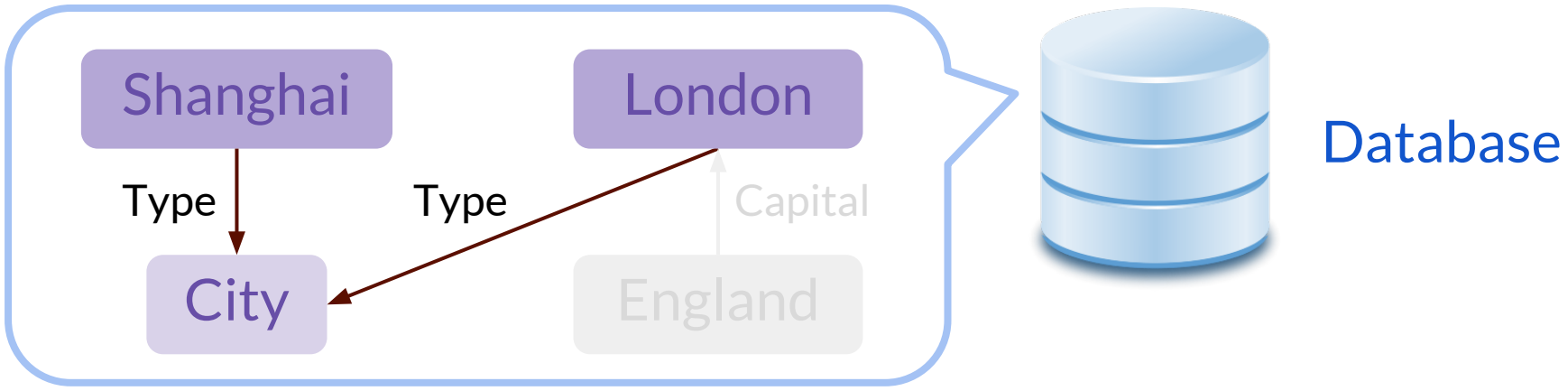
かなりの部分を、著者による発表スライド
http://cs.stanford.edu/~ppasupat/resource/ACL2015-slides.pdf
から使わせて頂いています

Parse questions into executable logical forms

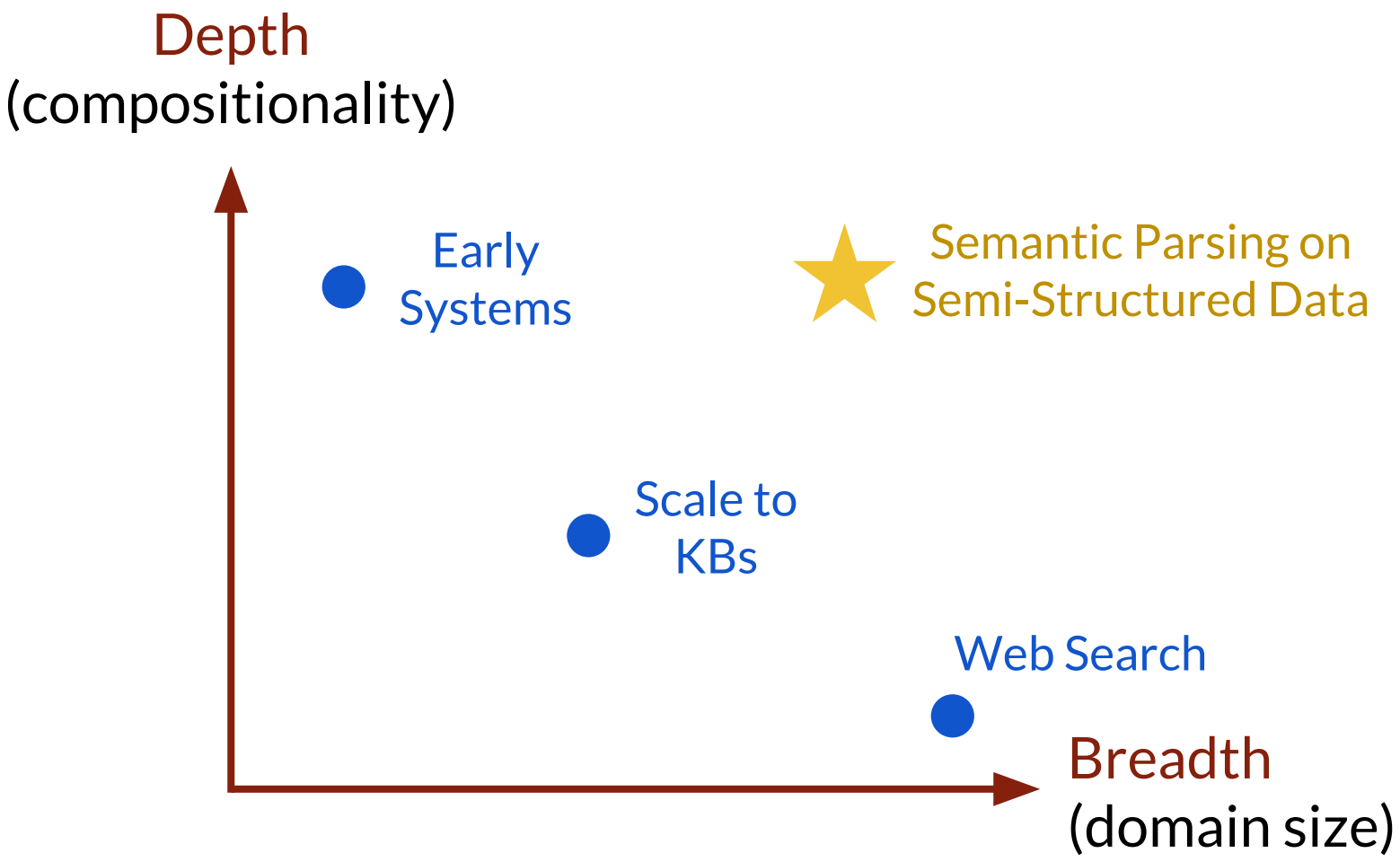
In which city was Ada Lovelace born?

↓

Type.City ⊓ PeopleBornHere.AdaLovelace



Motivation


Depth
(compositionality)
Early
Systems
Semantic Parsing on
Semi-Structured Data
Scale to
KBs
Web Search
Breadth
(domain size)

説明しようとして力尽きました

Percy Liang の ICML2015
チュートリアルスライドが大いに参考になります

http://icml.cc/2015/tutorials/icml2015-nlu-tutorial.pdf

Knowledge Source Compositionality

fixed domain

Early Systems

```
answer(A,
  count(B,
    (river(B), loc(B, C),
     largest(D, (state(C), population(C, D)))),
    A)))
```

deep composition

Scale to KBs

R[FirstAppearance].KittyPryde

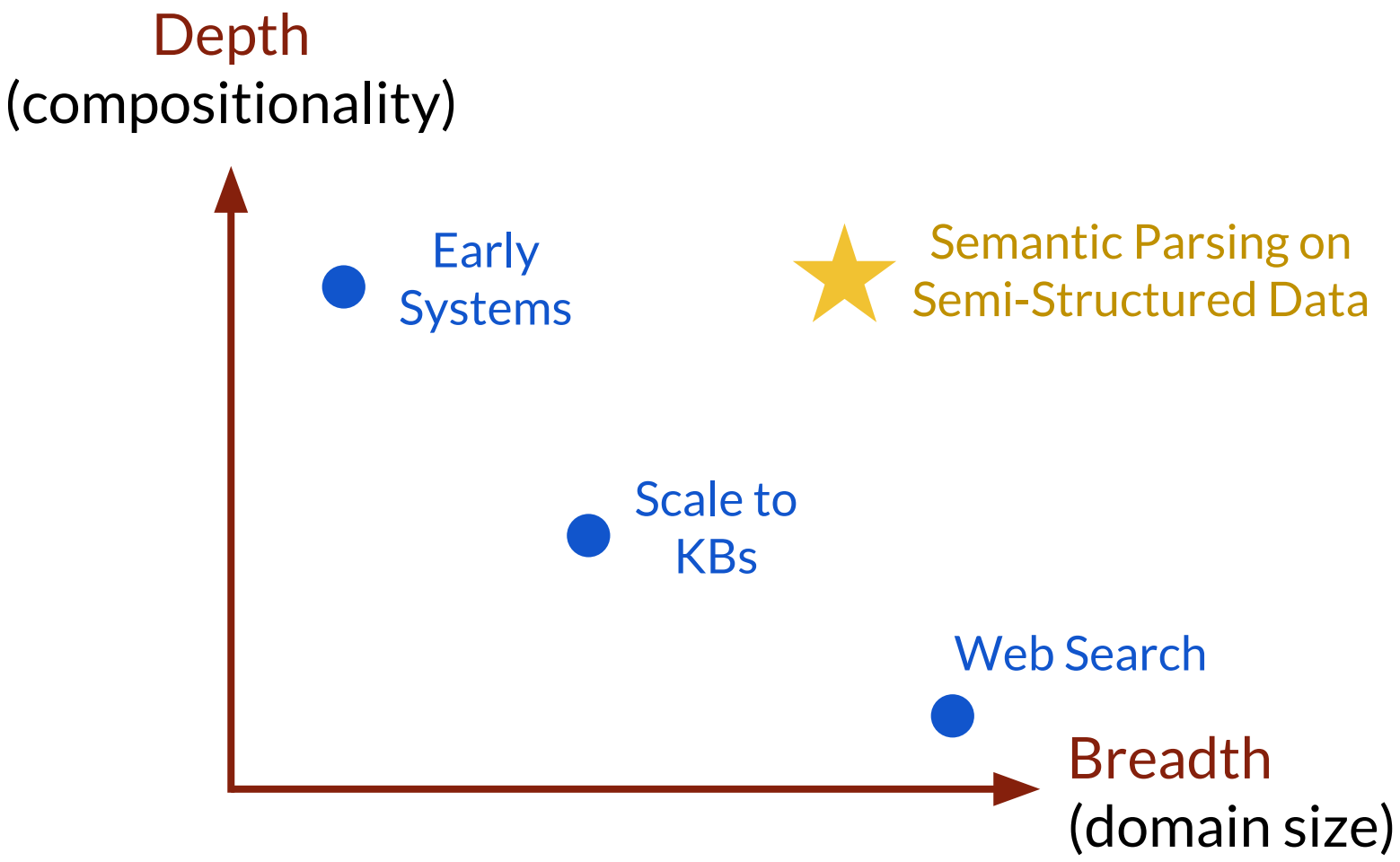
Web Search

単なるキーワード検索

broad domain

no compositionality

Motivation


Depth
(compositionality)
Early
Systems
Semantic Parsing on
Semi-Structured Data
Scale to
KBs
Web Search
Breadth
(domain size)

ポイント

- ウェブ上のテーブルに対する質問応答
- Question – Denotation のペアから論理表現への変換（Semantic Parser) を学習するモデル
  - 全体の枠組みは “Semantic Parsing on Freebase from Question-Answer Pairs” (Berant, 2013) の延長線上
  - サポートされる演算は増えている
- テーブルを「グラフ」で表現する

Task Description

Input: utterance x and HTML table t

Output: answer y

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

x = Greece held its last Summer Olympics in which year?

y = 2004

データセット：Wikipediaから、AMTを使って作りました！

Dataset

WikiTableQuestions dataset:

- Tables t are from Wikipedia
- Questions x and answers y are from Mechanical Turk — Prompts are given to encourage compositionality

MT Task 1 : 質問を作ってもらう

Wikipediaのテーブルを見せて、質問を作ってもらう。(36種類のプロンプト)
例）「最後の」を含めた質問を作れ

MT Task 2 : 答えをつけてもらう

Wikipediaのテーブルと、 Task 1 で作った質問を見せて、答えをつけてもらう

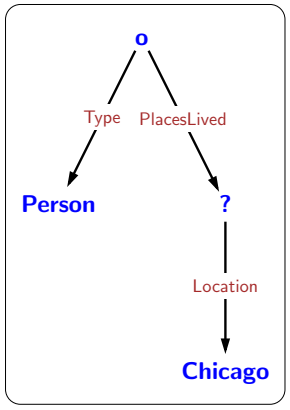
⇒ 22033 Question-Answer Pair on 2108 Tables

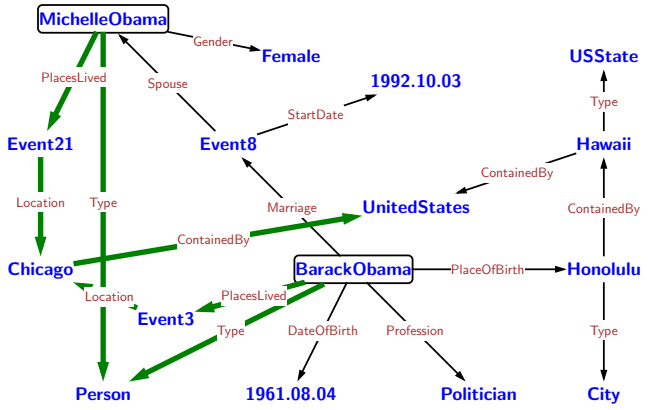
lambda DCS

- (Liang, 2013) : Model-theoretic compisitional semantics のための論理表現形式

lambda-DCS 表現を自然言語からいかに得るか

Type.Person ⊓ PlacesLived.Location.Chicago





サポートされる演算

Entity
Chicago

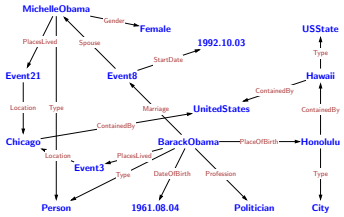
Join
PlaceOfBirth.Chicago

Intersect
Type.Person⊓PlaceOfBirth.Chicago

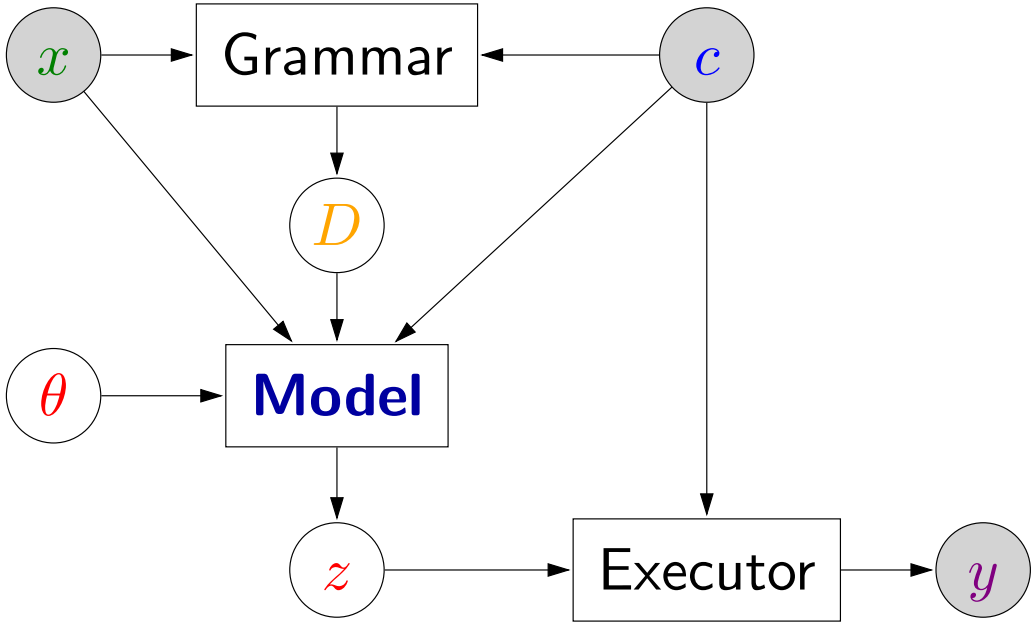
Aggregation
count(Type.Person ⊓ PlaceOfBirth.Chicago)

Superlative
argmin(Type.Person ⊓ PlaceOfBirth.Chicago, DateOfBirth)

Components of a semantic parser



people who have lived in Chicago



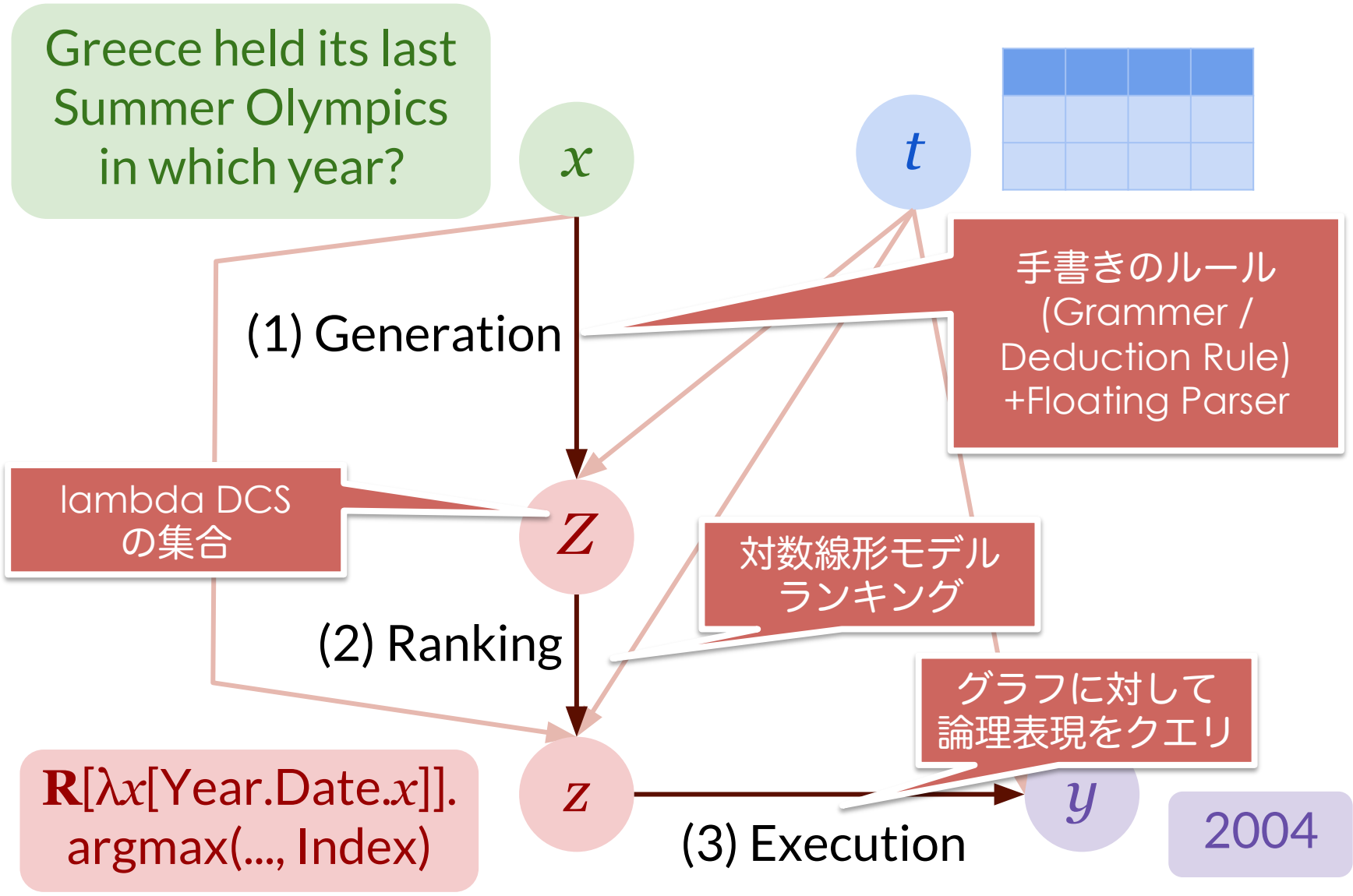
`Type.Person` $\sqcap$ `PlacesLived.Location.Chicago`

$\{$`BarackObama`, ...$\}$

Parser

Learner

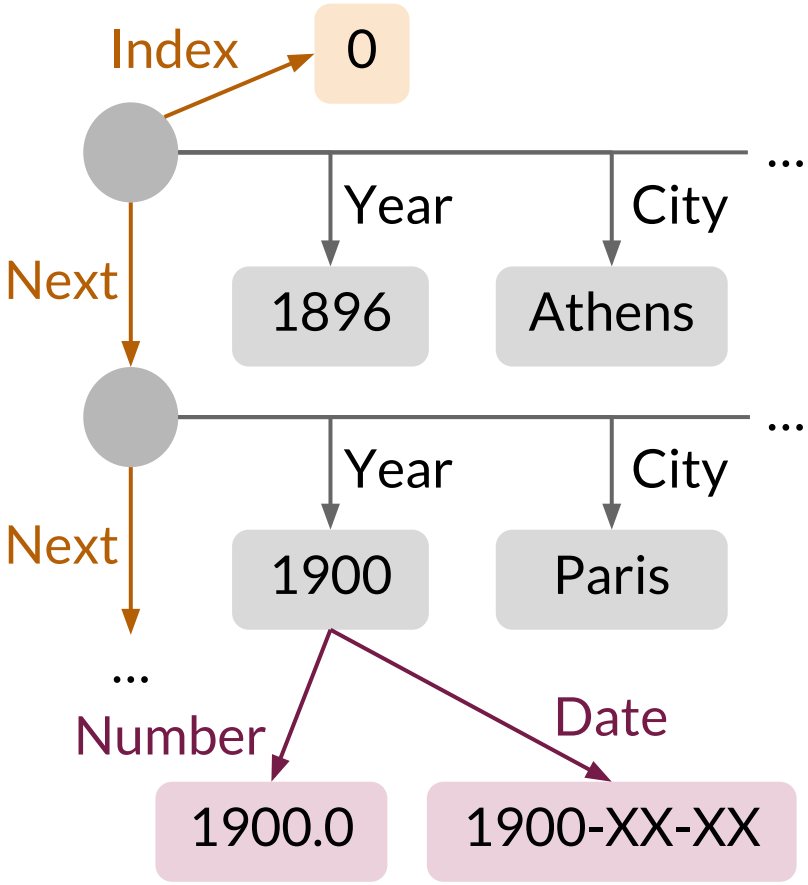
Approach


Greece held its last Summer Olympics in which year?
x
t
(1) Generation
手書きのルール (Grammer / Deduction Rule) +Floating Parser
lambda DCS の集合
Z
対数線形モデル ランキング
(2) Ranking
グラフに対して論理表現をクエリ
R[λx[Year.Date.x]]. argmax(..., Index)
z
y
(3) Execution
2004

Representation

Convert table t to knowledge graph w

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

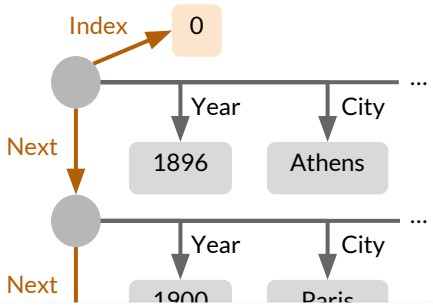


テーブルをグラフで表現する

- 利点
  - 異なる正規化形をノードとして表現できる
  - グラフのトラバースとしていくつかの操作を表現できる
    - 例)「次の・・・」 => Nextポインタをたどる
  - lambda DCSで直接問い合わせできる

Approach

Greece held its last Summer Olympics in which year?



x

w

(1) Generation

手書きのルール
(Grammer / Deduction Rule)
+Floating Parser

lambda DCS
の集合

z

対数線形モデル
ランキング

(2) Ranking

グラフに対して
論理表現をクエリ

$\mathbf{R}[\lambda x[\text{Year.Date.}x]].$
argmax(..., Index)

z

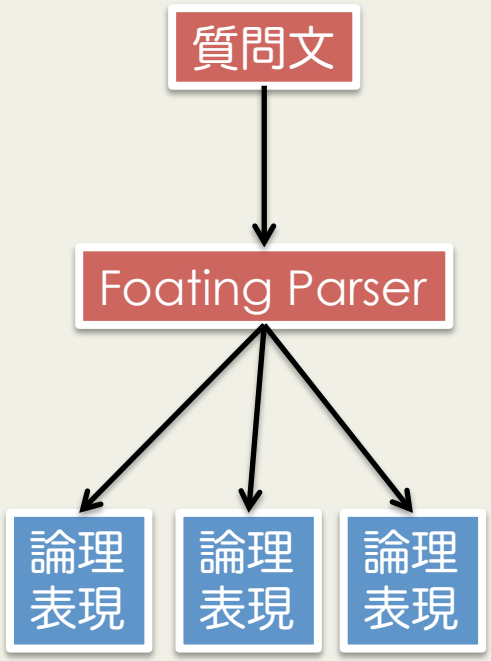
(3) Execution

y

2004

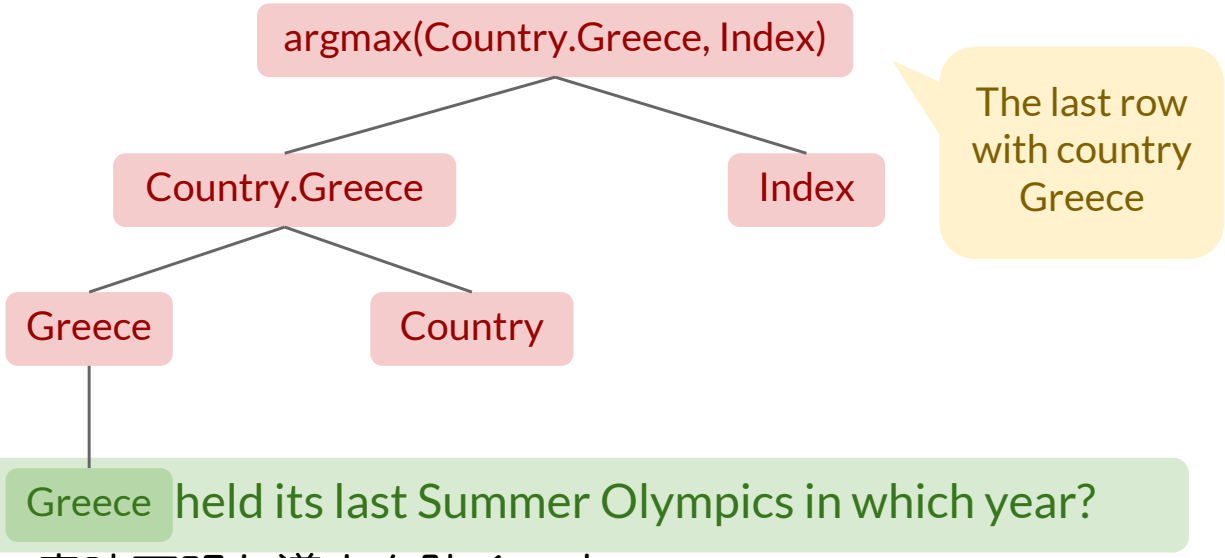
論理表現の生成

基本的には、 ボトムアップパーサー(文法は Table 2, Table 3)
空文字列からnon-terminalを出す仕組み “Floating” を導入



あとでランキング

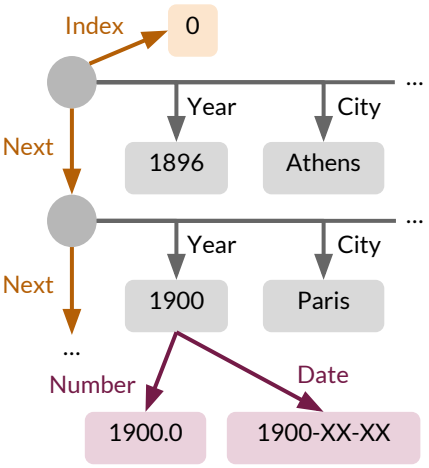
Connection between floating predicates and phrases in the question are made during **ranking**



意味不明な導出を防ぐ工夫:
Pruning, Beam Search, Strong Type Constraint

Approach

Greece held its last Summer Olympics in which year?



x

w

(1) Generation

z

(2) Ranking

z

y

(3) Execution

R[λx[Year.Date.x]].
argmax(..., Index)

2004

Ranking

Given a set Z of candidate formulas z, define a log-linear distribution:

$$p_\theta(z \mid x, w) \propto \exp\{\theta^\top \varphi(x, w, z)\}$$

where

- θ = parameter vector
- $\varphi(x, w, z)$ = feature vector

Learning

質問文　テーブル　答え

Given training example (x, w, y), define

$$p_\theta(y \mid x, w) = \sum_{z \in Z} p_\theta(z \mid x, w)\, \mathbf{I}(y = [\![z]\!]_w)$$

答えの確率を最大に

θを使ってDCSを導出

As usual, we choose θ to maximize the (L1 regularized) expectation of $\log p_\theta(y \mid x, w)$ over training data

評価

- 評価指標
  - Acc: 生成された(最も高いランクの)zがyを得た割合
  - Oracle: 生成された z のうち最低1つ 正しい y が得られる割合
- ベースライン
  - IR-inspired : テーブルセルの上のsoftmax
  - WQ : Berant and Liang (2014)
    - 差分：superlative(argmin, argmax) , union, intersection等

Semantic Parsing導入による改善

	accuracy	oracle
IR-inspired	12.7	70.6
WQ	24.3	35.6
This work	**37.1**	76.6

ルール追加による改善

http://cs.stanford.edu/~ppasupat/resource/ACL2015-slides.pdf
に、面白いPositive Exampleがいくつかあります。

本論文の貢献まとめ

- ウェブ上のテーブルを用いてセマンティックパーサーを訓練する
  - 基本アイディア：テーブルをグラフ表現 + Lambda DCSで問い合わせ
  - Lambda DCSの生成には、ボトムアップのパーサーと、機械学習に基づくランキングを使う
  - データはWikipediaからクラウドソーシングで作っている

- ちょっとずるい点：
  - <テーブル集合> が与えられたもとでの QA ではなく、 <テーブル> が与えられたもとでの QA
  - どのテーブルに答えがあるか、は分かっている状況

おまけ：データセットの特徴

- 22033 Q-A Pair, 2108 Tables, 3929 Unique column headers, 13396 columns
- Only 20% of questions can answered using Freebase (WikiTableQuestions have broad coverage)
- Logical Operation Coverage :
- Compositionality :

Operation	Amount
join (table lookup)	13.5%
+ join with `Next`	+ 5.5%
+ aggregate (`count`, `sum`, `max`, ...)	+ 15.0%
+ superlative (`argmax`, `argmin`)	+ 24.5%
+ arithmetic, ⊓, ⊔	+ 20.5%
+ other phenomena	+ 21.0%