开放域语义解析

韩先培　　陈波

中文信息处理实验室-让机器理解中文
Chinese Information Processing Laboratory

中国科学院软件研究所ISCAS
Institute of Software, Chinese Academy of Sciences

最新PDF：http://www.icip.org.cn/zh/zh_resource/

大纲

- 语义解析简介
- 基于词典-组合文法的语义解析
- 基于语义图的语义解析
- 基于神经网络的语义解析
- 总结和展望

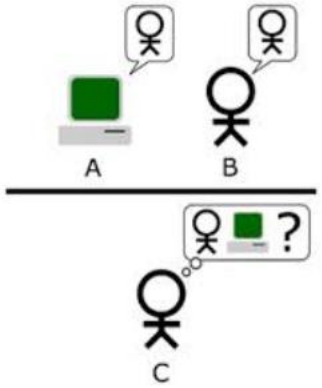
大纲

- 语义解析简介
  - 研究意义
  - 任务介绍
  - 重要组件
  - 发展历程
- 基于词典-组合文法的语义解析
- 基于语义图的语义解析
- 基于神经网络的语义解析
- 总结和展望

自然语言理解

- 让计算机理解自然语言是AI的圣杯之一
- 图灵测试的核心是测试计算机是否理解自然语言

"Can machines think?"



Q: Please write me a sonnet on the subject of the Forth Bridge.

A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.

A: (Pause about 30 seconds and then give as answer) 105621.

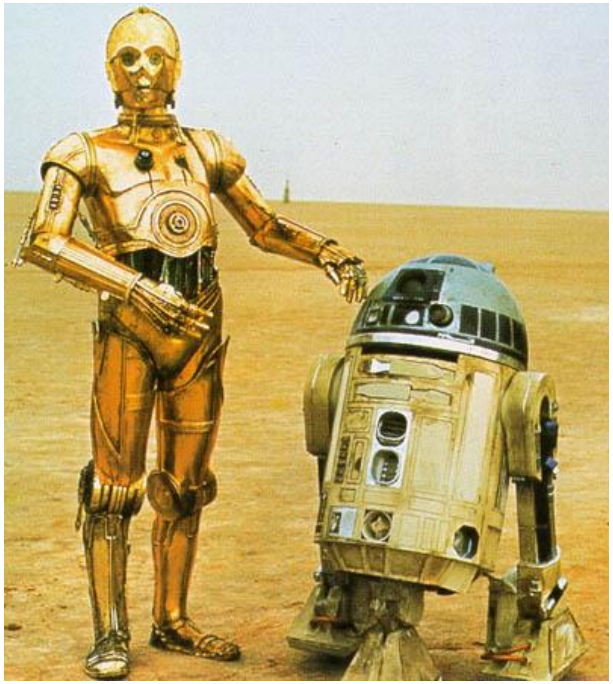
自然语言理解是人工智能的圣杯

网友问伯克利机器学习教授、美国双料院士Michael I. Jordan：”如果你有10亿美金，你怎么花?

Michael I. Jordan: “我会用这10亿美金建造一个NASA级别的自然语言处理研究项目。”

---《AMA: Michael I Jordan》

自然语言理解是AI是否具有智能的重要体现



核心挑战：自然语言和语义表示间的鸿沟

- 自然语言多样，具有歧义，其理解依赖世界知识和情境
- 计算机能理解的语言是统一语义、显式结构、无歧义的

北京的人口
住在北京的人有多少
北京的居民人数
北京人数量
北京人数
家在北京的人有多少万
…

多样性

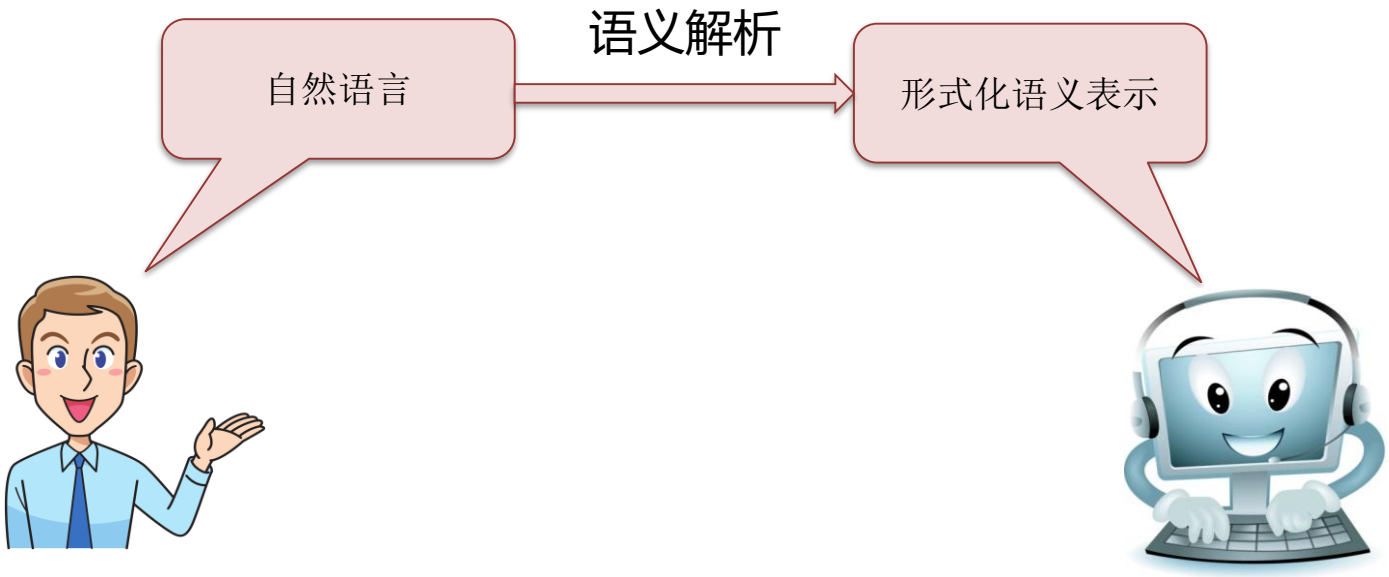
东北的冬天，你能穿多少就穿多少。（多穿点）

武汉的夏天，你能穿多少就穿多少。（少穿点）

歧义性

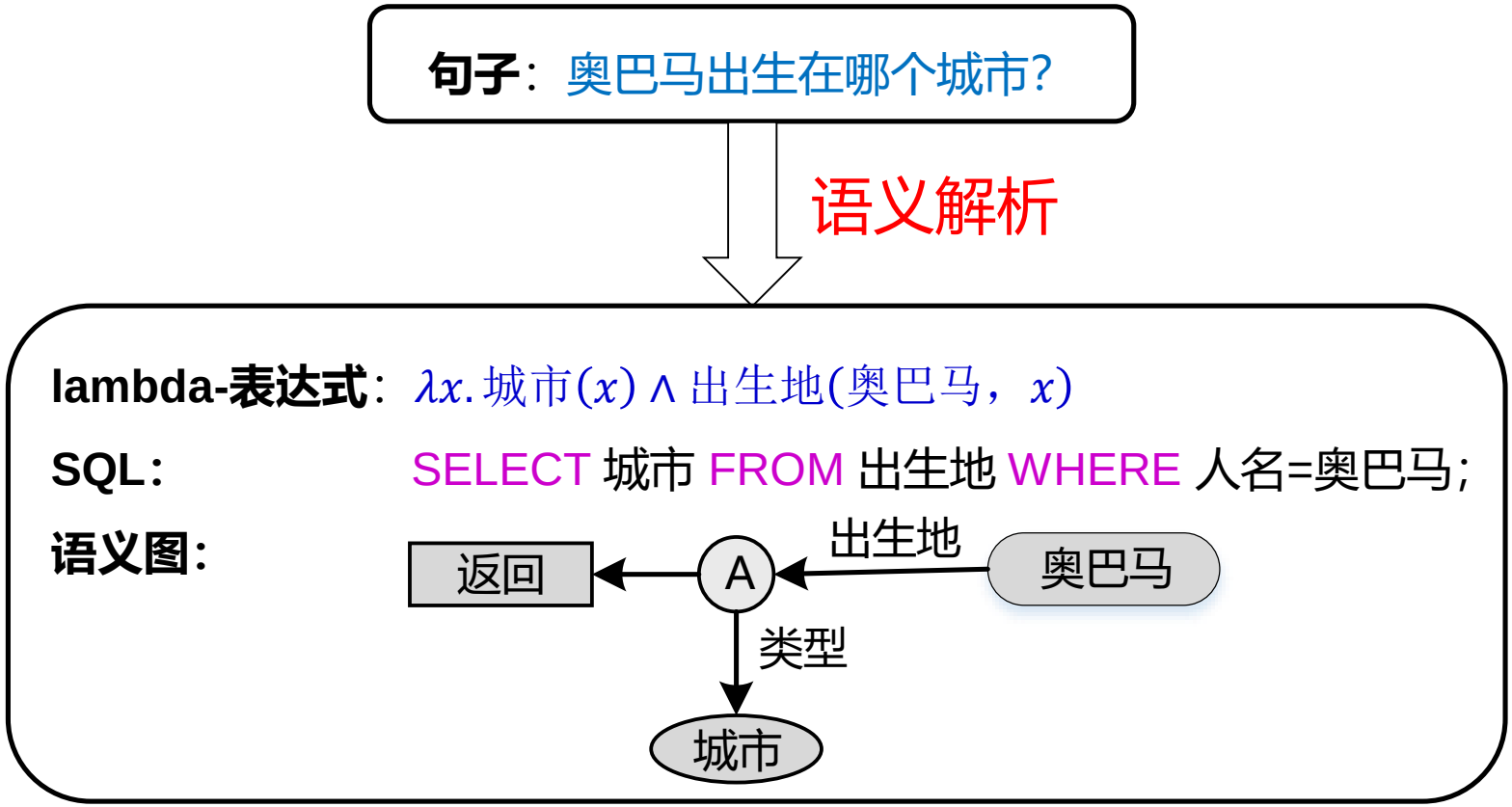
语义解析（Semantic Parsing）

- 语义解析是实现自然语言理解的关键技术之一，目的在于建立自然语言到计算机可以理解的形式化语义表示的映射



语义解析任务定义

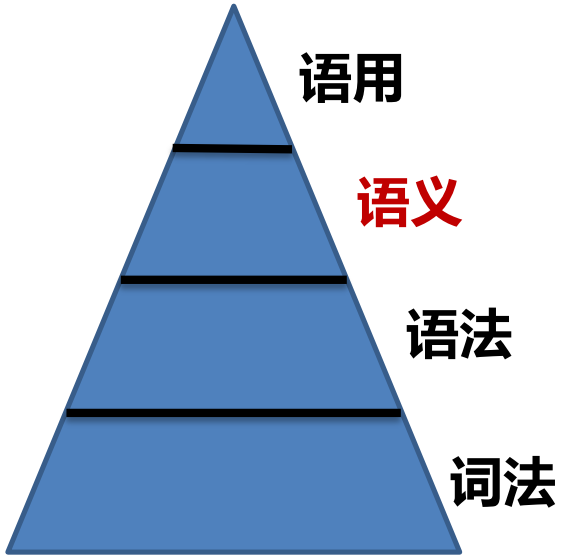
- 将自然语言句子转换成计算机可识别的、可计算的、完全的语义表示，如lambda-表达式、SQL、语义图等



John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. AAAI-1996.

语义解析在自然语言处理中的定位

自然语言处理



语用：句子背后的用意
——今天下午可能会下雨。表示下午的室外活动可能会取消。

语义：句子表达的是什么意思
——今天下午可能会下雨。表示一种可能的状态

语法：句子的语法和结构
——如：句子中哪个是主语，哪个是谓语，哪个是宾语

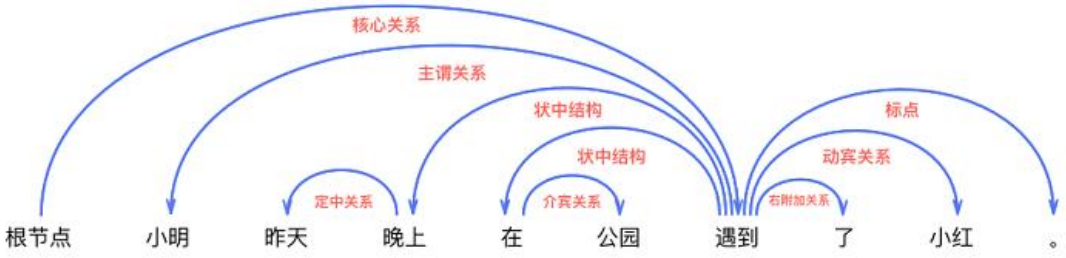
词法：词语的起源，以及词与词之间的关联
——如：“语”字的起源，与“言”的关系

语义解析：从浅层到深层

语言到语义

语义角色标注（SRL）

谁对谁做了什么？在哪里？什么时候？



Vasin Punyakanok, Peter Koomen, Dan Roth, and Wen-tau Yih.
Generalized inference with multiple semantic role labeling systems. CoNLL- 2005.

语义解析：从浅层到深层

语言到语义

框架语义解析（Frame-SP）

一个框架对应一个原型（情境）

小明花了200元从小红那**买**了一辆自行车

交易

买方：小明
卖方：小红
交易物品：自行车
价格：200元

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. Frame-semantic parsing. CL-2014.

语义解析：从浅层到深层

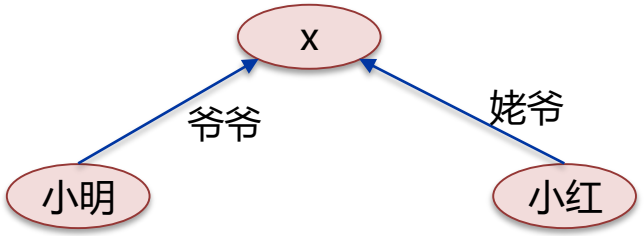
本报告核心

语言到语义

语义解析（SP）

句子的完整意思，不受限于框架的定义

小明的爷爷是小红的姥爷



Raymond J. Mooney. Learning for Semantic Parsing. CICLing-2007.

语义解析的任务场景

- 语言到结构化查询语言（language to query）

北京的理工科大学有哪些? ⟹ $\lambda x.$ 大学(x) ∧ 理工科(x) ∧ 位于(北京)

学校	类型	位置
清华	综合	北京
北大	综合	北京
北航	理工	北京
中科大	理工	合肥
北语	语言	北京
北邮	理工	北京
…	…	…

⟹

北航
北邮
…

Yuk Wah Wong and Raymond J. Mooney.
Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. ACL-2007.

语义解析的任务场景

- 语言到代码（language to code）

输入NL

Adds a scalar to this vector in place.

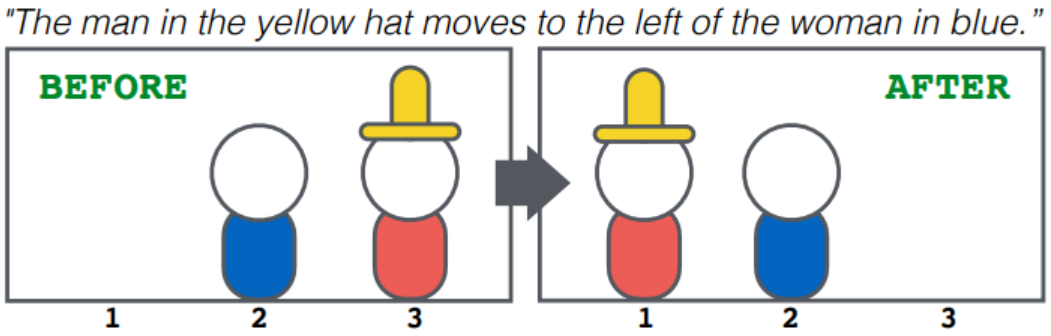
输出代码

```
public void add (final double arg0) {
  for (int loc0 = 0; loc0 < vecElements.length; loc0 ++) {
    vecElements[loc0] += arg0;
    }
}
```

Srinivasan Iyer, Yannis Konstas, Alvin Cheung, Luke Zettlemoyer.
Mapping Language to Code in Programmatic Context. EMNLP-2018.

语义解析的任务场景

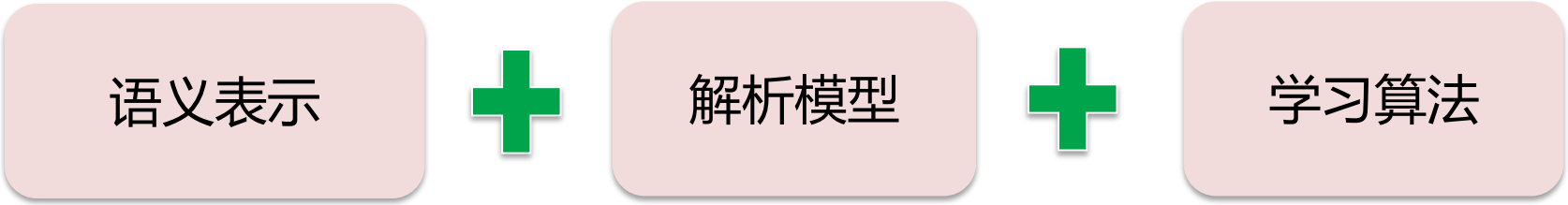
- 语言到机器操作指令 （language to instruction）



move(hasHat(yellow), leftOf(hasShirt(blue)))

Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, Percy Liang.
From language to programs: bridging reinforcement learning and maximum marginal likelihood. ACL-2017.

语义解析中的核心组件



语义表示：如何表示句子的语义

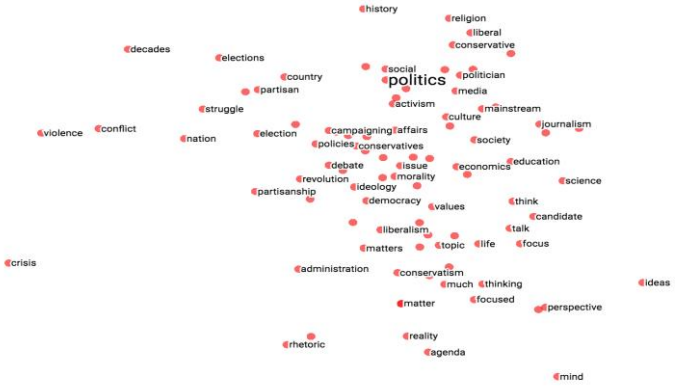
- **框架语义表示**
  - 优势：结构性表示，语义明确
  - 劣势：人工定义框架，覆盖性有限

交易

买方：小明
卖方：小红
交易物品：自行车
价格：200元

业界中常用

- **分布式语义表示**
  - 优势：应用性好，对机器友好
  - 劣势：整体表示，细粒度区分性不好



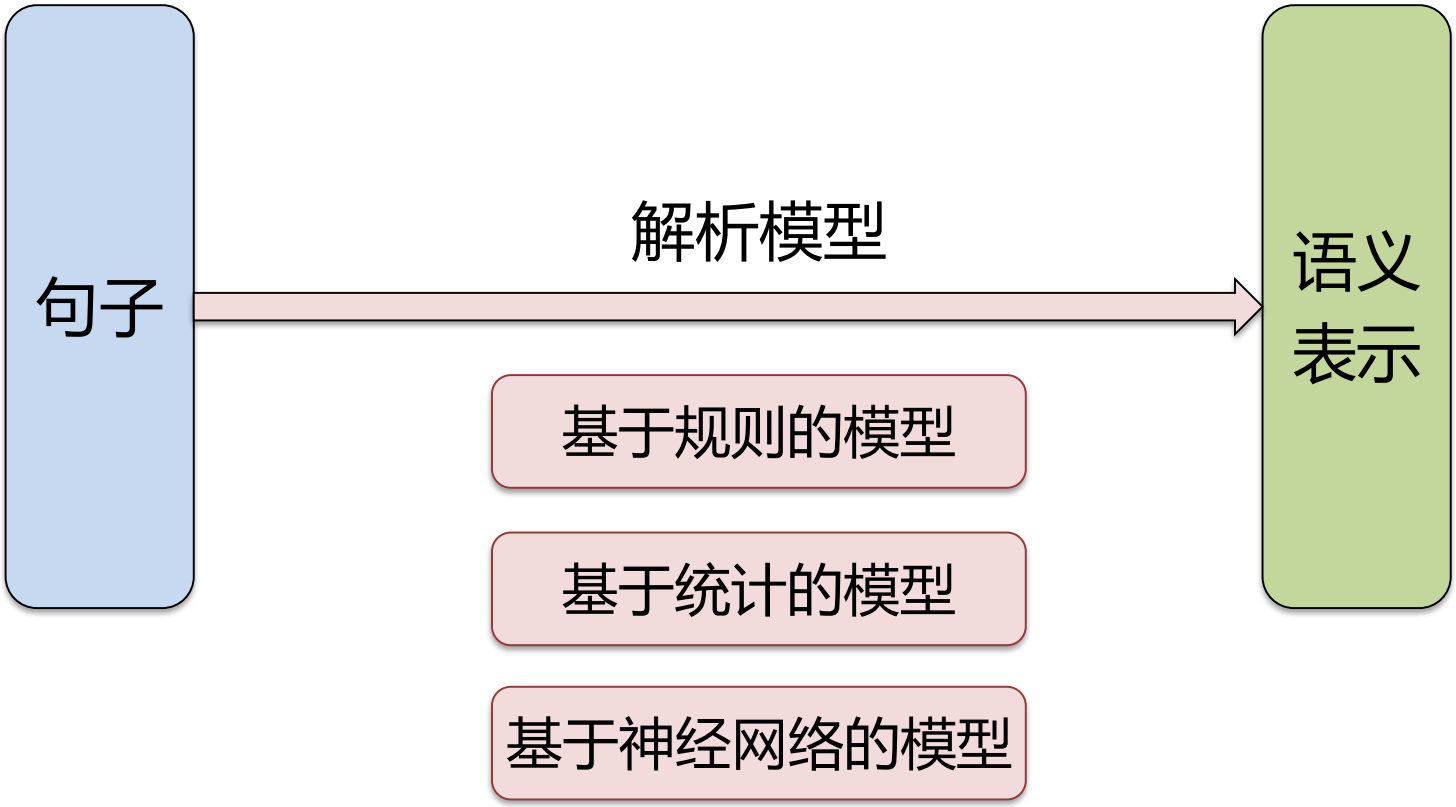
- **模型理论-组合语义表示**
  - 优势：表达能力强，机器可直接利用
  - 劣势：难以生成

逻辑表达式：$\lambda x.\exists y.$ 出生地$(y,x) \wedge$ 作者(平凡的世界，y)

语义图：

学术界中常用

解析模型：句子转换为语义表示的建模



学习算法：如何学习转换过程

- **目的**
  - 让解析模型从一个句子的多种候选语义表示中选择正确的语义表示
  - 让解析模型学习转换过程

Input: What states border Texas?
Output: $\lambda x.state(x) \wedge borders(x,texas)$

Input: What is the largest state?
Output: $argmax(\lambda x.state(x), \lambda x.size(x))$

Input: What states border the largest state?
Output: $\lambda x.state(x) \wedge borders(x, argmax(\lambda y.state(y), \lambda y.size(y)))$

- **监督学习算法**
  - 输入：<句子，语义表示> 标注对
  - 目标：最大似然估计：p(语义表示|句子)

Example Learned Lexical Entries

Words	Category
states	N : $\lambda x.state(x)$
major	N/N : $\lambda g.\lambda x.major(x) \wedge g(x)$
population	N : $\lambda x.population(x)$
cities	N : $\lambda x.city(x)$
river	N : $\lambda x.river(x)$

- **弱监督学习算法**
  - 输入：弱监督信号，如<句子，知识库，答案>，<句子，候选，点击>标注
  - 目标：最大边界似然估计，语义表示当做潜变量

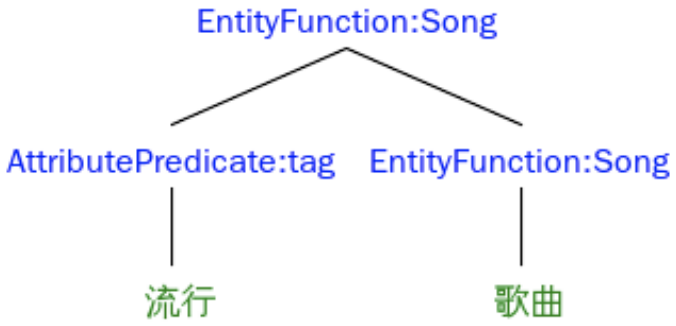
What is the largest state that borders Texas?

New Mexico

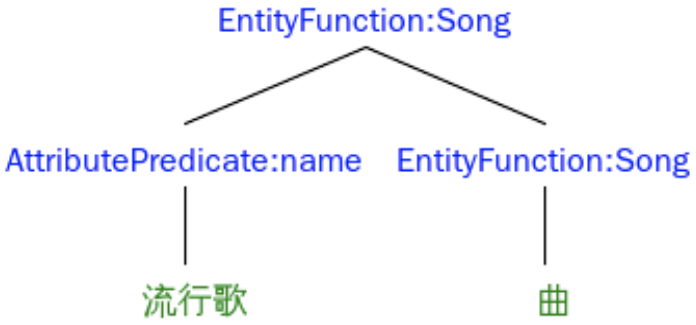
$argmax(\lambda x.state(x) \wedge border(x, TX), \lambda y.size(y))$ → *New Mexico* ✓

$argmax(\lambda x.river(x) \wedge in(x, TX), \lambda y.size(y))$ → *Rio Grande* ✗

模型训练之后如何选择正确的语义表示



Song → Tag Song : 1.0
Tag → 流行: 1.2
Song → 歌曲: 1.2



Song → Name Song : 0.2

Name → 流行歌: 0.3
Song → 曲: 1.2

相关资源：语义解析数据集概览

数据集	开始使用时间	数据总数	Single-domain/ Open-domain	Domain是否封闭
Geo	1996	880	Single	封闭
Jobs	2001	640	Single	封闭
ATIS	2005	5418	Single	封闭
Webquestions	2013	5800	Open	封闭
Free917	2013	917	Open	封闭
Wikitable	2015	22033	Open	开放
Overnight	2015	13682	Multi	开放
Graphquestions	2016	5166	Open	开放
Spider	2018	10181	Open	开放

相关资源：语义解析工具包

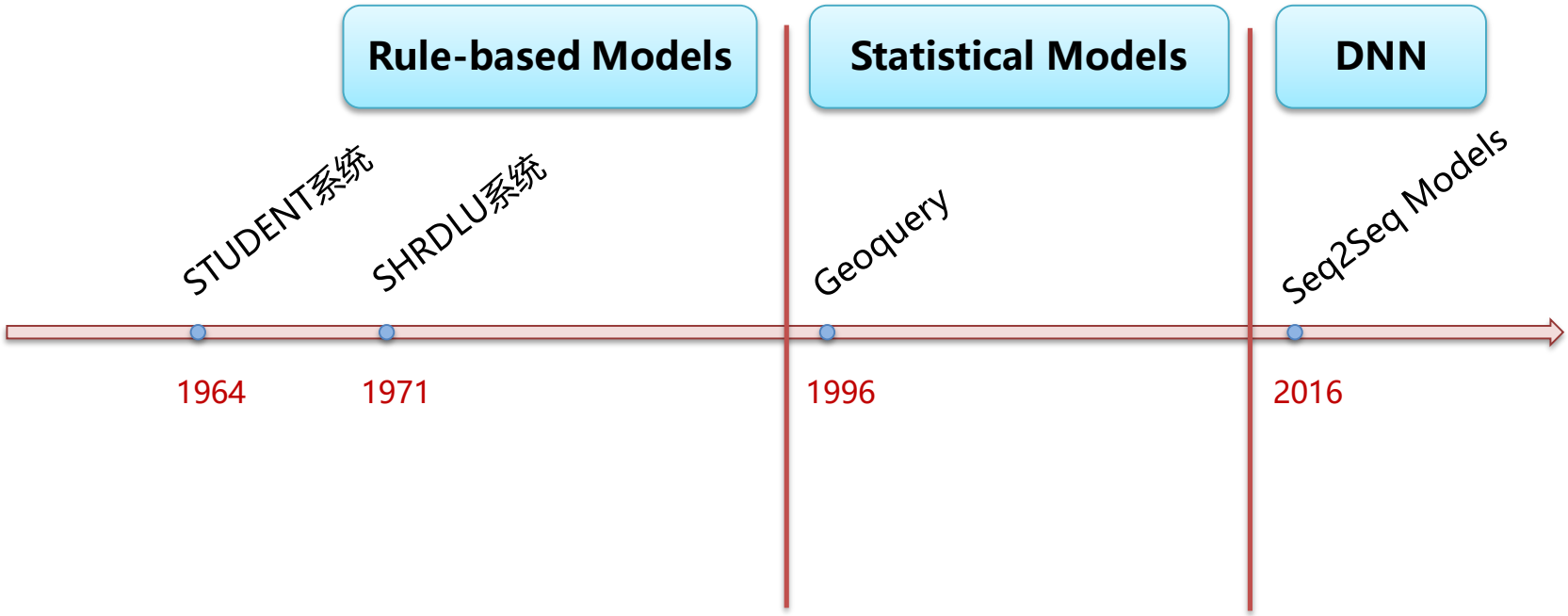
- Sempre（https://github.com/percyliang/sempre）
  - Stanford, Percy Liang’ s group
  - Java
  - DCS, WikiTable

- Cornell SPF（https://github.com/lil-lab/spf）
  - Cornell University and University of Washington
  - Yoav Artzi, Luke Zettlemoyer , Tom Kwiatkowski, Kenton Lee
  - Java
  - CCG

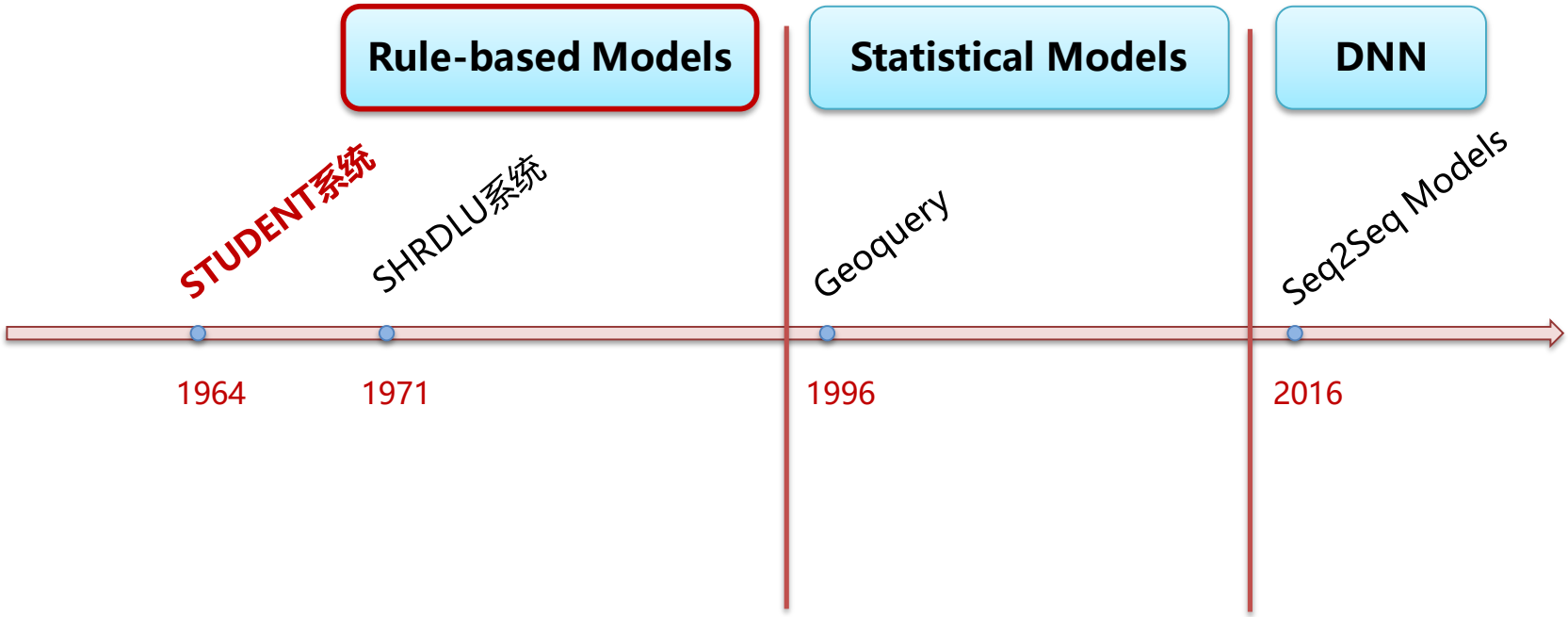
- AllenNLP（https://github.com/allenai/allennlp）
  - Allen AI group
  - Pytorch
  - Neural

语义解析的发展历程

语义分析的发展历程


Rule-based Models
Statistical Models
DNN
STUDENT系统
SHRDLU系统
Geoquery
Seq2Seq Models
1964
1971
1996
2016

语义分析的发展历程

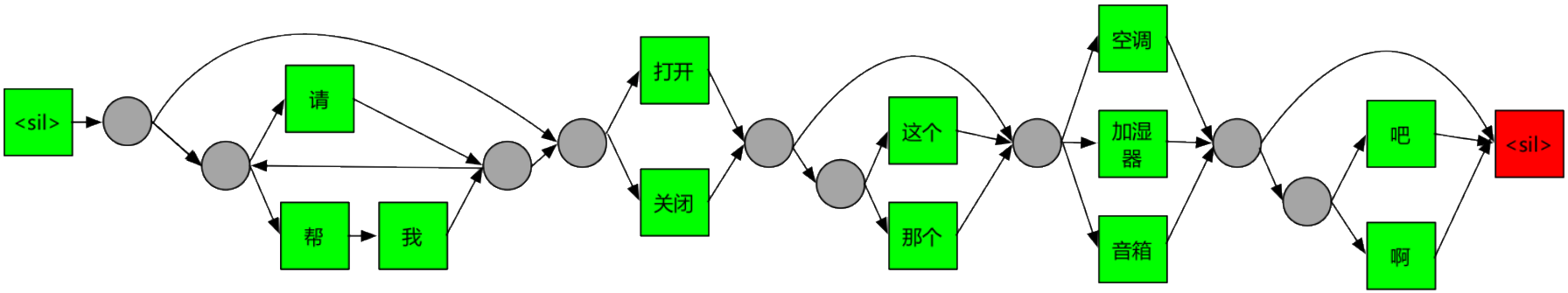

Rule-based Models
Statistical Models
DNN
STUDENT系统
SHRDLU系统
Geoquery
Seq2Seq Models
1964
1971
1996
2016

基于规则的语义解析方法

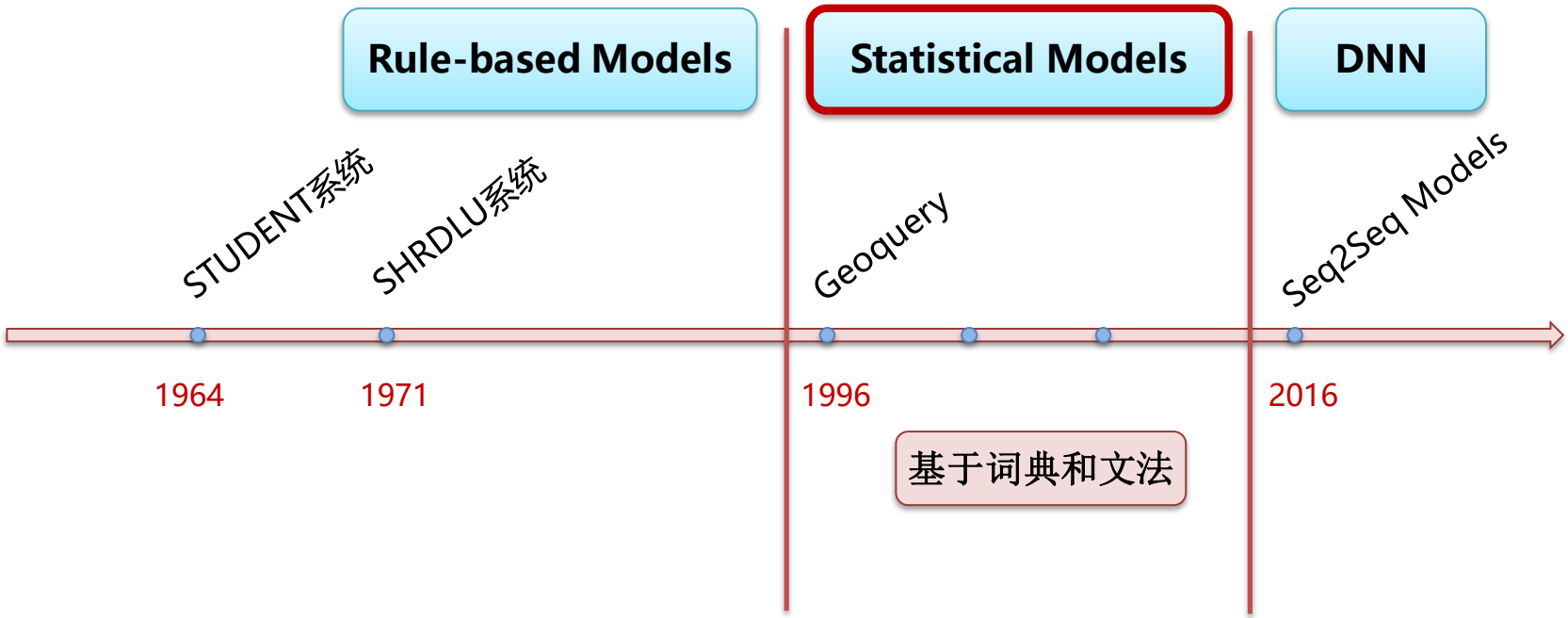
- STUDENT系统 [Bobrow, 1964]：基于规则的线性代数求解
- 规则系统：错误可控可溯源、起步容易/自助构建、增量式模型

```
public <controlDevice> = <startPolite> <command> <endPolite>;

<command> = <action> <object>;
<action> = (打开|关闭);
<object> = [这个|那个](空调|加湿器|音箱){device};
<startPolite> = (请|帮 我) *;
<endPolite> = [啊|吧];
```

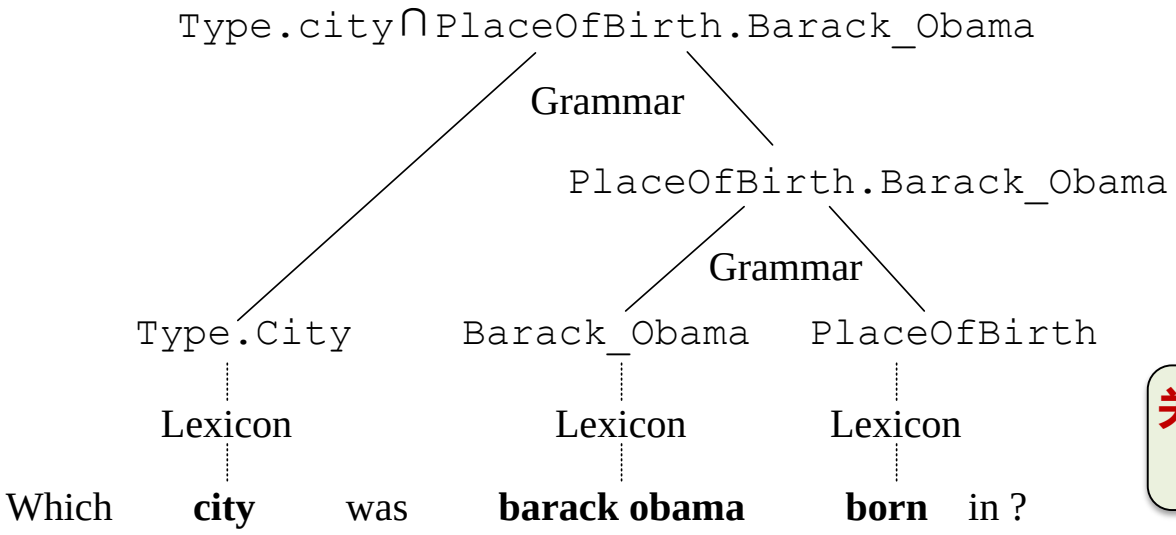


语义分析的发展历程


Rule-based Models
Statistical Models
DNN
STUDENT系统
SHRDLU系统
Geoquery
Seq2Seq Models
1964
1971
1996
2016
基于词典和文法

基于统计的语义解析方法

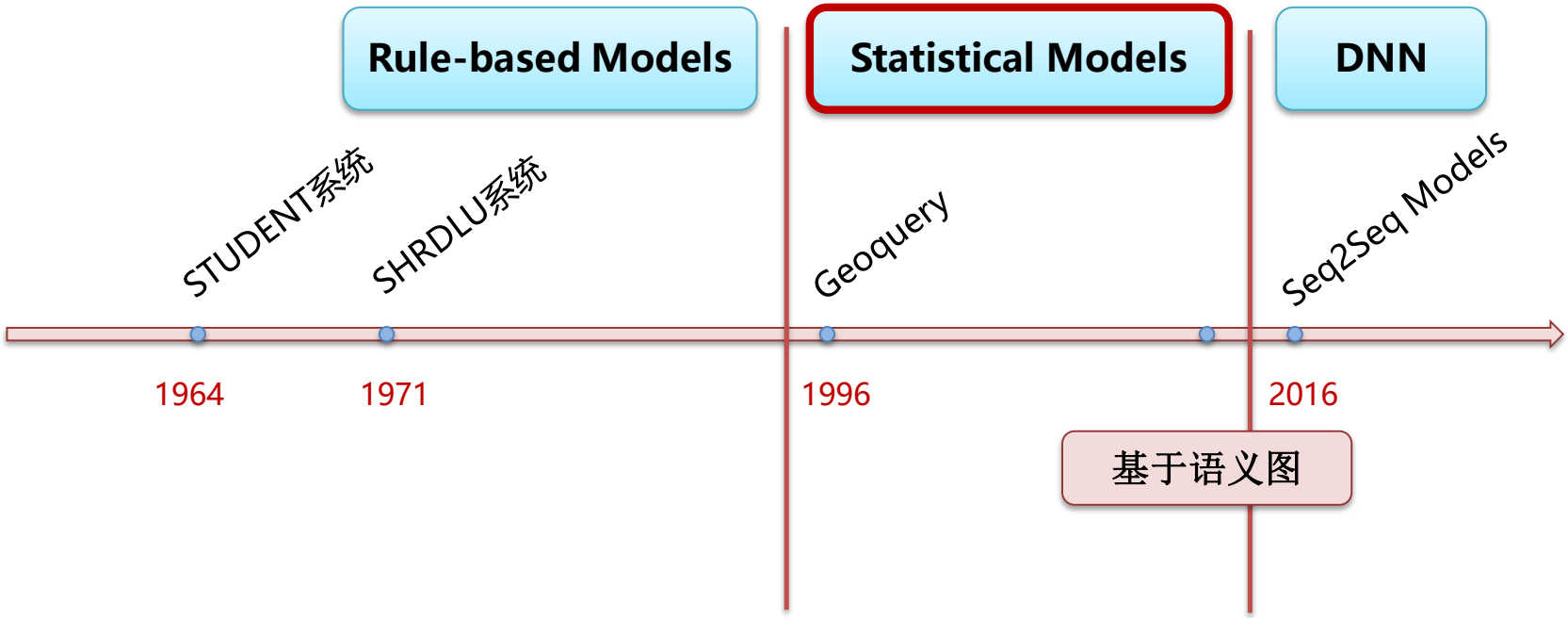
- 重要引发点：统计方法的兴起，并逐步替代规则方法
- 代表性方法：基于词典-组合文法的语义解析
  - CCG和DCS
  - 核心组件：词典（lexicon）、组合文法（grammar）、概率模型



关键：词典学习、组合模型

Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang.
Semantic parsing on Freebase from question-answer pairs. EMNLP-2013.

语义分析的发展历程


Rule-based Models
Statistical Models
DNN
STUDENT系统
SHRDLU系统
Geoquery
Seq2Seq Models
1964
1971
1996
2016
基于语义图

基于语义图的语义解析方法

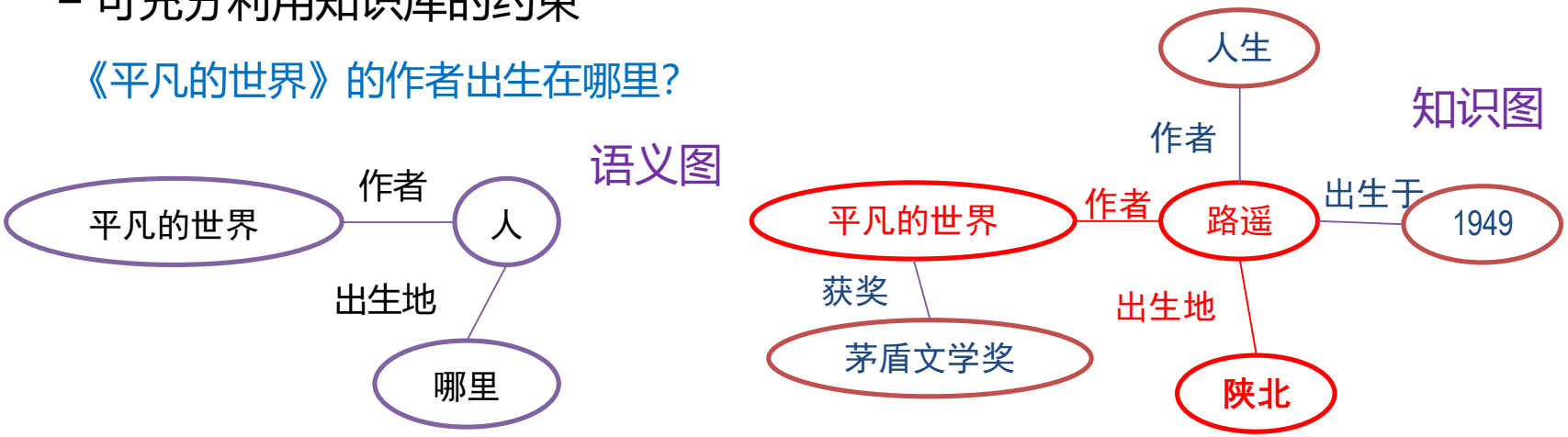
- 重要引发点：大规模开放知识图谱（Freebase等）的出现

- 用语义图来表示句子的语义
- 语义解析转换为语义图构建/生成的问题
- 带来的优势：
  - 不需要提前学习词典
  - 不需要定义组合文法
  - 可充分利用知识库的约束

关键：如何构建/生成语义图

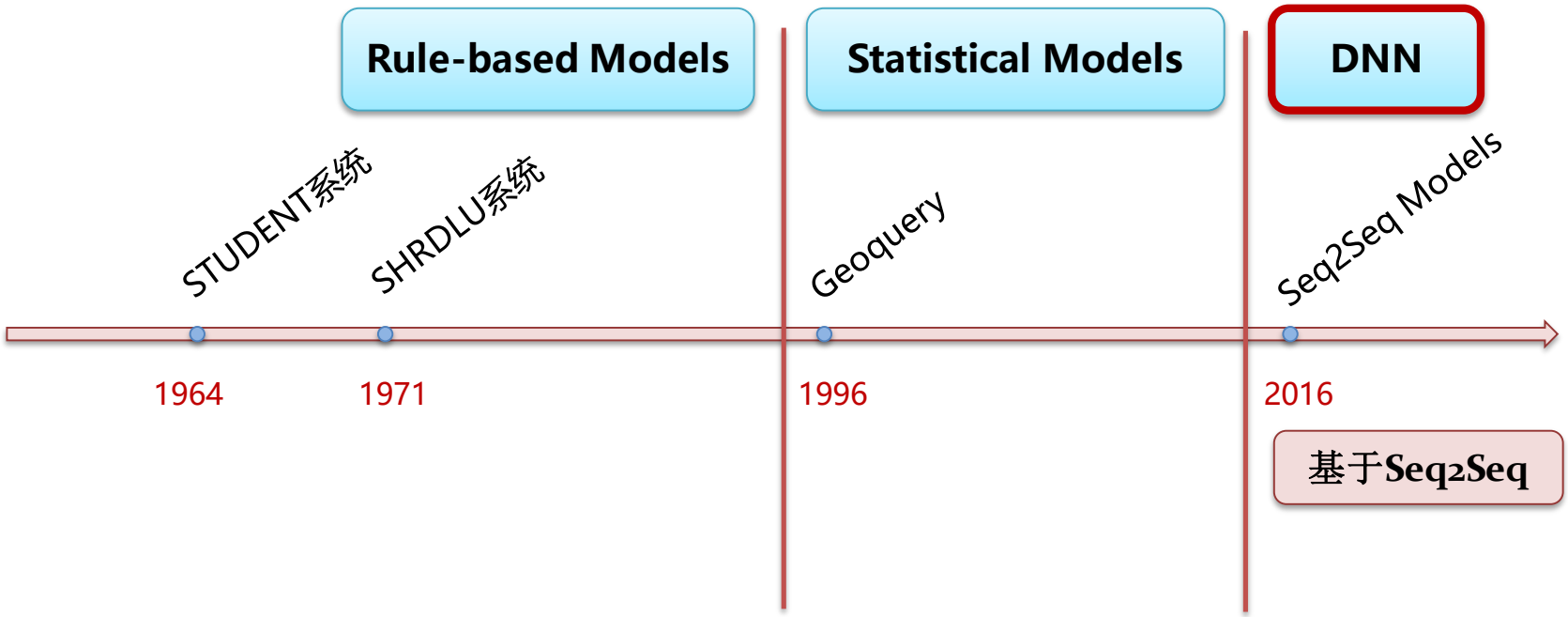
依存分析（dependency）
模板（template）
分步构建

《平凡的世界》的作者出生在哪里？



Siva Reddy, Mirella Lapata, Mark Steedman. Large-scale Semantic Parsing without Question-Answer Pairs. TACL-2014.

语义分析的发展历程

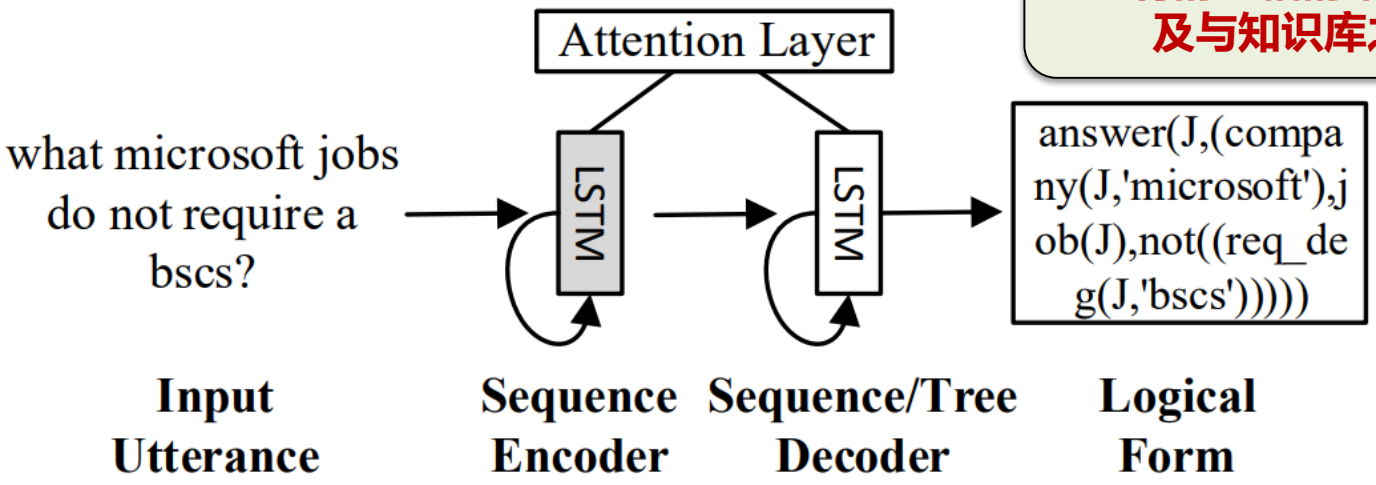

Rule-based Models
Statistical Models
DNN
STUDENT系统
SHRDLU系统
Geoquery
Seq2Seq Models
1964
1971
1996
2016
基于Seq2Seq

基于神经网络的语义解析方法

- 重要引发点：神经网络模型（特别是Seq2Seq模型）在自然语言处理其他任务上的应用及取得的成功

- 将逻辑表达式序列化
- 语义解析转换为词语序列到逻辑表达式序列的翻译过程（Seq2Seq）

关键：如何利用语义表示所具有的严格的结构约束，以及与知识库之间的联系



Li Dong, Mirella Lapata. Language to Logical Form with Neural Attention. ACL-2016.

小结：语义解析简介

- **语义解析**：从自然语言到计算机语义表示
- **任务设置**：语言到查询、语言到代码、语言到指令
- **关键组件**：语义表示+解析模型+学习算法
- **发展历程**：
  - 基于规则的方法（60s-90s）
  - 基于统计的方法（90s-10s）
  - 基于语义图的方法(10s至今)
  - 基于NN的方法(10s至今)

大纲

- 语义解析简介
- 基于词典-组合文法的语义解析
  - 组合语义的基本准则
  - 代表性方法（CCG和DCS）
  - 从限定域到开放域
- 基于语义图的语义解析
- 基于神经网络的语义解析
- 总结和展望

组合语义的基本准则

- Principle of compositionality [Frege, 1965]
  - the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.
  - 复杂表达式的意义由其子表达式的意义以及意义如何组合的规则共同决定

组合语义的基本准则

- Principle of compositionality [Frege, 1965]
  - the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.
  - 复杂表达式的意义由其子表达式的意义以及意义如何组合的规则共同决定

关键组件：词典和组合文法

负责审议的唐代中央机构是什么？

短语拆分：

负责 | 审议 | 唐代中央机构

短语语义：

pred:职能 | value:审议 | type:唐代中央机构

语义组合：

λ *x* ISA(*x*, 唐代中央机构) ∧ 职能(*x*, 审议)

Input Trigger	Output Category
constant c	NP : c
arity one predicate p	N : λx.p(x)
arity one predicate p	S \ NP : λx.p(x)
arity two predicate p	(S \ NP) / NP : λx.λy.p(y, x)
arity two predicate p	(S \ NP) / NP : λx.λy.p(x, y)
arity one predicate p	N / N : λg.λx.p(x) ∧ g(x)
arity two predicate p and constant c	N / N : λg.λx.p(x, c) ∧ g(x)
arity two predicate p	(N \ N) / NP : λx.λg.λy.p(y, x) ∧ g(x)
arity one function f	NP / N : λg.argmax/min(g(x), λx.f(x))
arity one function f	S / NP : λx.f(x)

词典

组合文法

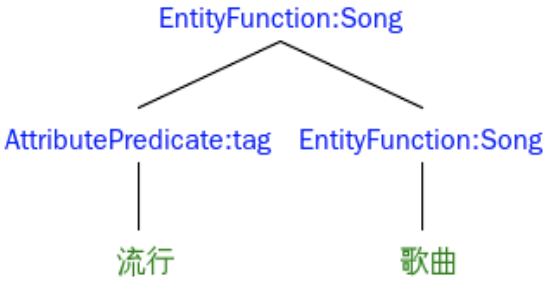
语义表示

关键组件：词典和组合文法

- **词典**
  - 存储自然语言词语到词语语义的映射
  - 语义组合的基本单元

词语	词语语义
生于	PlaceOfBirth
歌	Song
首都	CapitalOf
…	…

- **组合文法**
  - 将小语义单元组合成更大的语义单元
  - 组合语义基本准则中的combining rule



- **代表性方法**
  - CCG: Combinatory Category Grammar, 组合范畴文法
    [Zettlemoyer & Collins, 2005]
  - DCS: Dependency-Compositional Semantics, 依存组合语义文法
    [Liang et al., 2011]

基于CCG的语义解析

基于CCG的语义解析

- **CCG (Combinatory Categorial Grammar),由ACL终身成就奖获得者Steedman教授提出**
  - Steedman, M. (1996). *Surface Structure and Interpretation*. The MIT Press.
  - Steedman, M. (2000). *The Syntactic Process*. The MIT Press.
- **早期用于句法分析，后期由Zettlemoyer & Collins(05)应用于语义解析**
  - 可以捕捉句子长距离依赖
- **核心组件**
  - 词典： 使用CCG Category来表示词汇语义
  - 组合文法：CCG组合算子(combinators)
  - 句子语义：使用lambda表达式表示句子语义

Luke Zettlemoyer and Michael Collins.
Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. UAI-2015.

CCG-语义表示：Lambda表达式

- 用于函数定义、函数应用和递归的一种形式化表示
- 常量Constants/变量Var
  - 实体、数字、变量
  - New York, 12, 北京, …
- 谓词Predicate
  - ProvinceOf, ISA...
- 逻辑连接符
  - 与^,或v,非(¬),蕴含（→）
- 量词
  - ∃，∀
- 额外的函数
  - Count，argmax, average, …

Lambda表达式表示—例子

- 与北京相邻的省有哪些?
  - λx. 省份(x) ∧ 相邻(北京，x)
- 与北京相邻的省有多少个?
  - count(x, 省份(x) ∧ 相邻(北京，x))
- 黄山在安徽的什么地方?
  - λx. 地方(x) ∧ 位于(x，安徽)
- 我国海拔最高的盆地在哪个省份?
  - λx.省份(x) ∧ 位于(argmax(y, 盆地(y), 海拔(y))，x)

CCG-词典

- **Lexicon**: 词语到范畴（category）的映射（范畴同时包含句法信息和语义信息）

词语	范畴（category）	
	句法部分	语义部分
Texas	NP	*texas*
borders	(S\NP)/NP	$\lambda x.\lambda y.border(y,x)$
Kansas	NP	*kansas*
Kansas city	NP	*kansas_city_mo*

其中*S*，*NP*是CCG里面的基本类型（还有*N*)
- *S*表示一个完整的句子
- *NP*表示一个专有名词
- *N*表示一个普通名词

*(S\NP)/NP*是一种复合类型
- 右向的slash(/)表示右侧可以接受一个参数（ *NP* ），接受之后得到左侧的类型（ *S\NP* ）
- 左向的slash(\)表示左侧可以接收一个参数（ *NP* ），接受之后得到左侧的类型（*S*)

CCG-连接符

- 组合文法
  - Application combinators
  - Composition combinators (**B**)
  - Type-raising combinators (**T**)
  - Substitution combinators (**S**)
  - Coordination combinators (Φ)

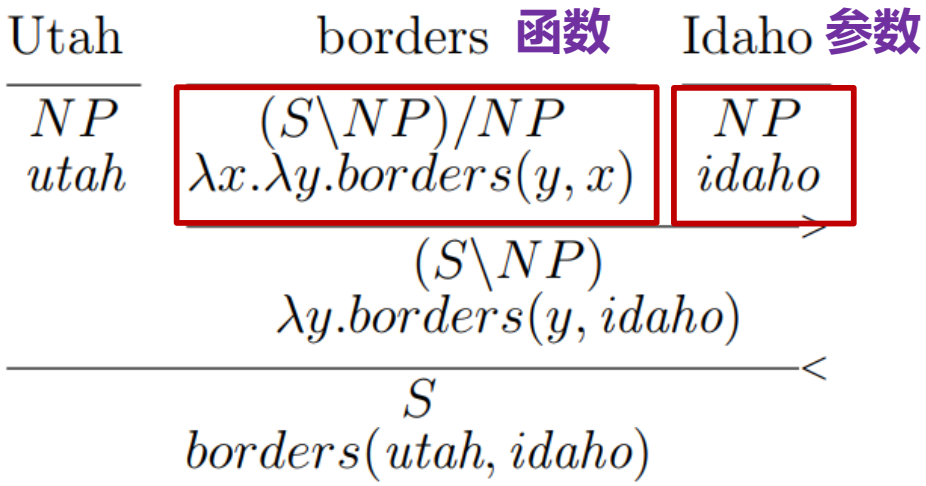
核心连接符

CCG-连接符-Application Combinators

- **应用**连接符（前向和后向）
  - 类似于函数的应用，如$f(x) = x + 1$，$f(2) = 2 + 1$

$$A/B : f \quad B : g \quad \Rightarrow \quad A : f(g) \qquad (>)$$

$$B : g \quad A\backslash B : f \quad \Rightarrow \quad A : f(g) \qquad (<)$$

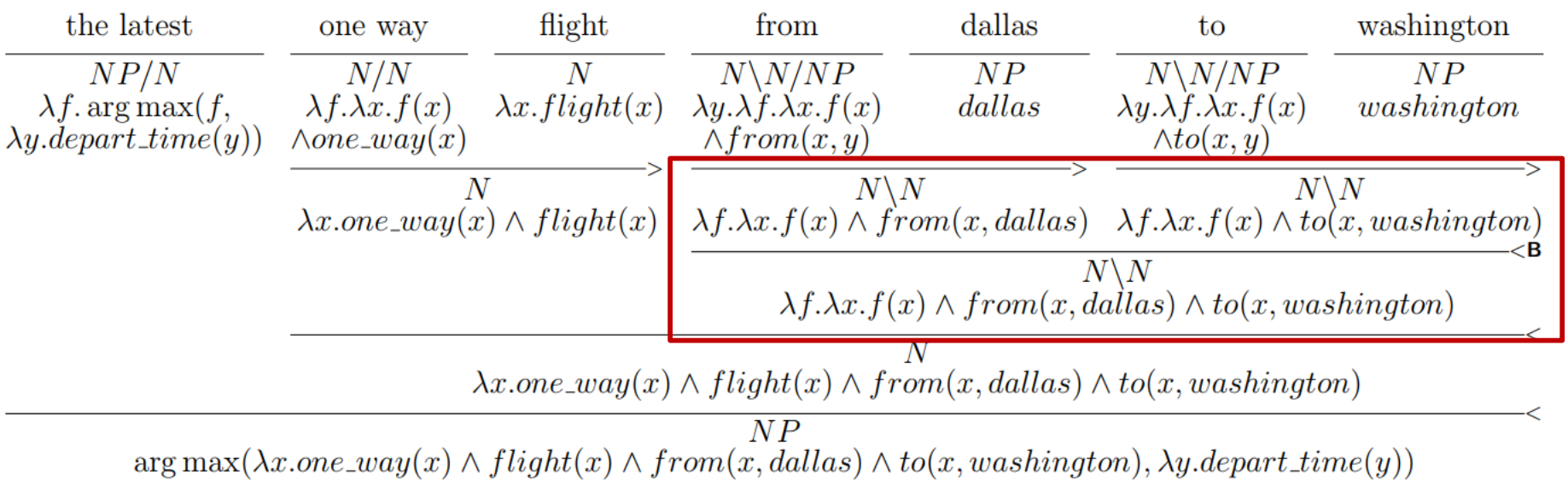


CCG-连接符-Composition combinators

- **组合**连接符（前向和后向）
  - 类似于组合两个函数，得到一个新函数同时保留函数的功能，多用于串联两个条件短语
  - 母亲 兄弟(x) → 舅舅(x)： 姚明的母亲的兄弟

$$A/B : f \quad B/C : g \quad \Rightarrow \quad A/C : \lambda x.f(g(x)) \quad (>\mathbf{B})$$

$$B\backslash C : g \quad A\backslash B : f \quad \Rightarrow \quad A\backslash C : \lambda x.f(g(x)) \quad (<\mathbf{B})$$

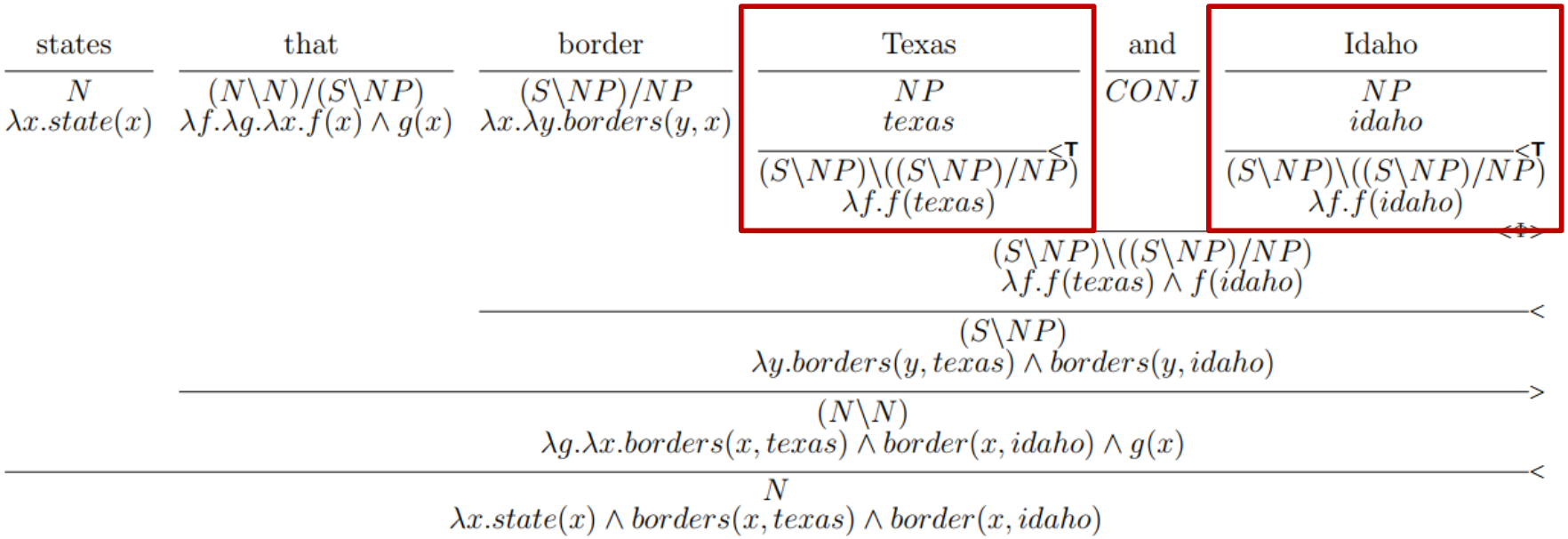


CCG-连接符- Type-raising combinators

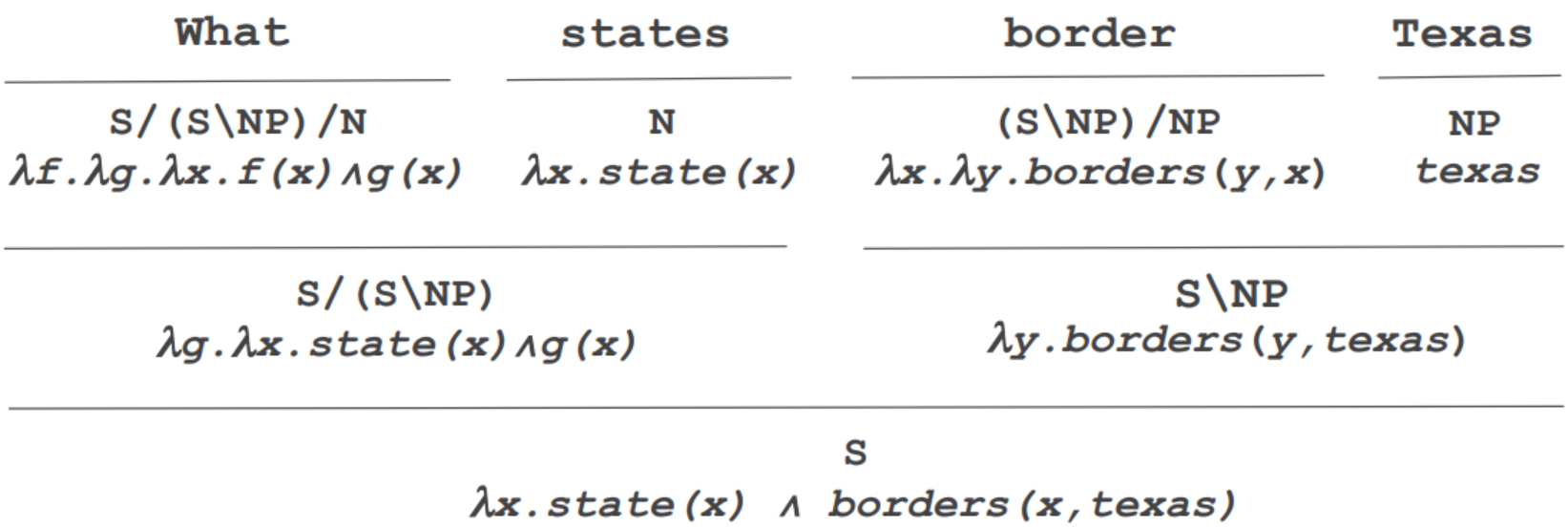
- **类型提升**连接符 (**T**) （前向和后向）
  - 类似于参数变为函数，函数的功能取决了新增的 T
  - *并列： 刘德华和陈奕迅的歌曲*
  - *省略： 播放刘德华（需要把刘德华提升为Song类型）*

$$A : g \quad \Rightarrow \quad T/(T\backslash A) : \lambda f.f(g) \quad (>\mathbf{T})$$

$$A : g \quad \Rightarrow \quad T\backslash(T/A) : \lambda f.f(g) \quad (<\mathbf{T})$$



基于CCG的解析例子


What
states
border
Texas
S/(S\NP)/N
λf.λg.λx.f(x)∧g(x)
N
λx.state(x)
(S\NP)/NP
λx.λy.borders(y,x)
NP
texas
S/(S\NP)
λg.λx.state(x)∧g(x)
S\NP
λy.borders(y,texas)
S
λx.state(x) ∧ borders(x,texas)

基于CCG的解析例子

What	states	border	Texas
S/(S\NP)/N $\lambda f.\lambda g.\lambda x.f(x) \wedge g(x)$	N $\lambda x.state(x)$	(S\NP)/NP $\lambda x.\lambda y.borders(y,x)$	NP *texas*

词汇

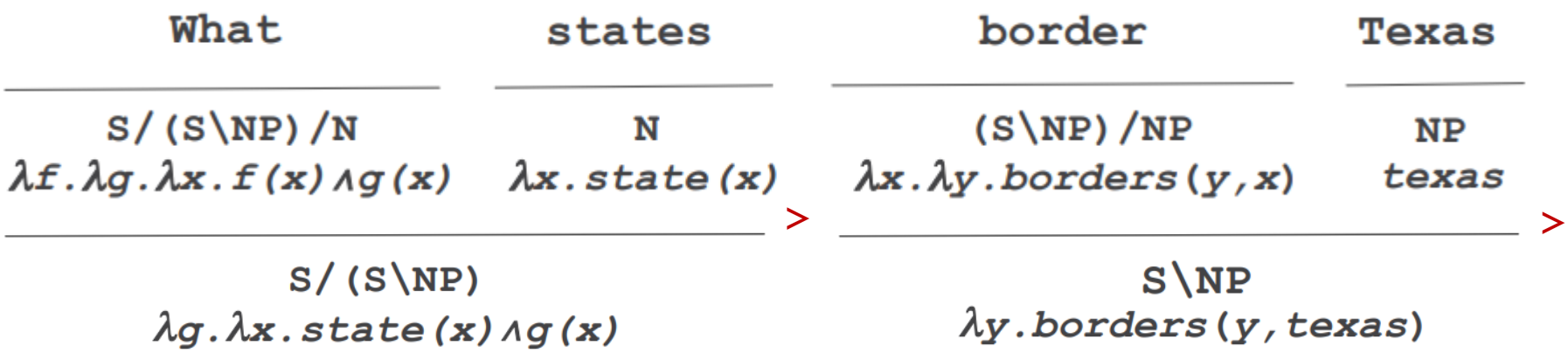
what: S/(S\NP)/N:$\lambda f.\lambda g.\lambda x.f(x) \wedge g(x)$

states: N:$\lambda x.state(x)$ 类型词汇

border: (S\NP)/NP:$\lambda x.\lambda y.borders(y,x)$ 二元谓词词汇

Texas: NP:*texas* 实体词汇

基于CCG的解析例子

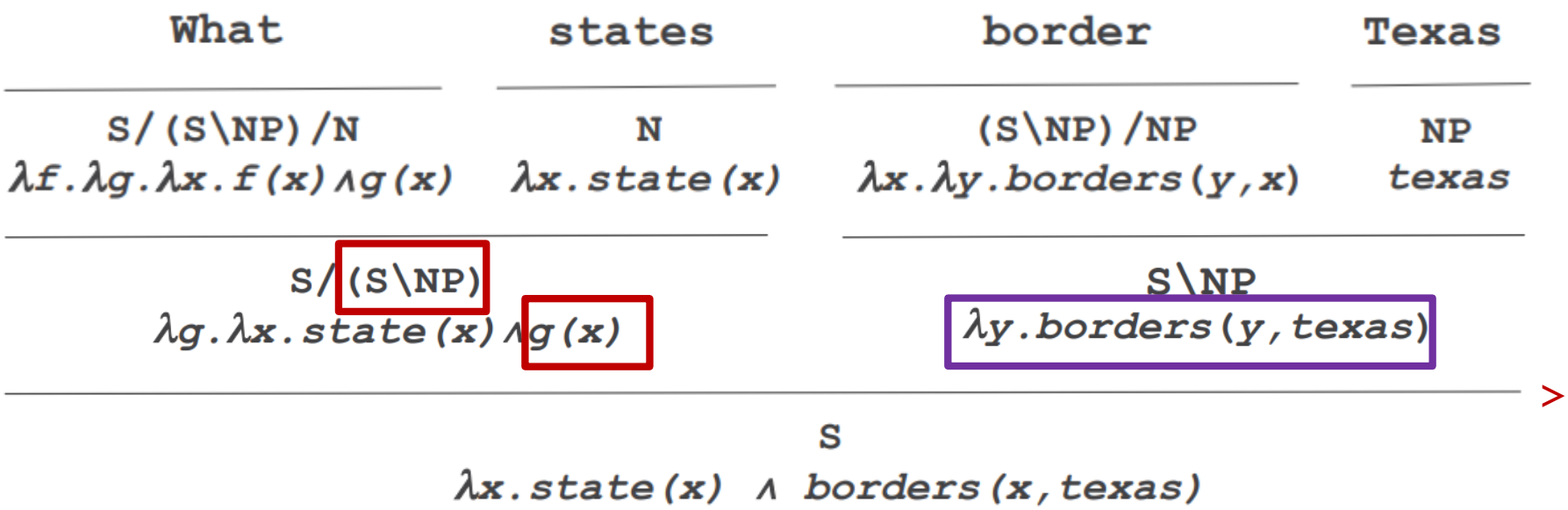


前向应用连接符 >

- X/Y : f　　Y : a　　=>　　X : f(a)

(S\NP)/NP	NP	S\NP
λx.λy.borders(y,x)	texas	λy.borders(y,texas)

基于CCG的解析例子


What
states
border
Texas
S/(S\NP)/N
λf.λg.λx.f(x)∧g(x)
N
λx.state(x)
(S\NP)/NP
λx.λy.borders(y,x)
NP
texas
S/(S\NP)
λg.λx.state(x)∧g(x)
S\NP
λy.borders(y,texas)
>
S
λx.state(x) ∧ borders(x,texas)

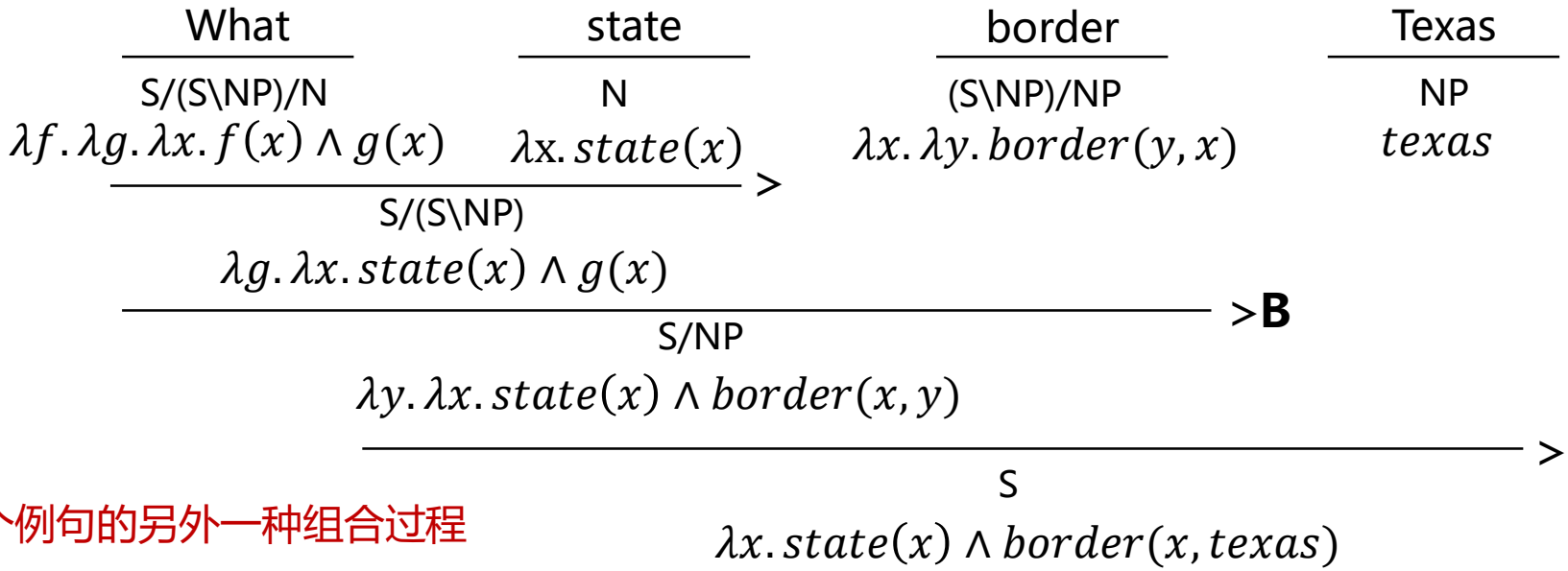
解析过程中的歧义

- **Lexicon层（主要）**
  - 一个词语可以映射到多个的category

    *Which rivers run through the states bordering **Mississippi**?*

    其中的Mississippi可映射为 **state: Mississippi**, or **river: Mississippi**
  - 初始的时候，没有标注的映射，任一词语可以映射到任一category

- **组合层（可忽略）**
  - 存在多种组合的可能性，但往往会得到相同的lambda-表达式



*上个例句的另外一种组合过程

PCCG：估计每一个lambda-表达式的概率

- **对数线性模型**：S为句子，L为lambda-表达式，T为解析的过程，θ为模型参数，f(L, T, S)为该解析的特征函数，一个句子被解析为L的概率为：

$$P(L \mid S;\theta) = \sum_T P(L,T \mid S;\theta)$$

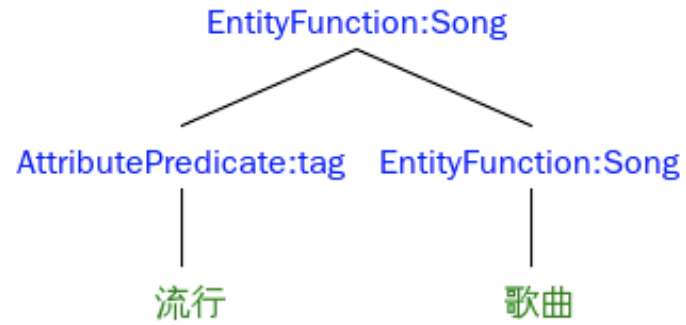
其中 $$P(L,T \mid S;\theta) = \frac{e^{f(L,T,S)\cdot\theta}}{\sum_{(L,T)} e^{f(L,T,S)\cdot\theta}}$$

- 重要特征：
  - 使用的词汇
  - 使用的组合文法
  - ……

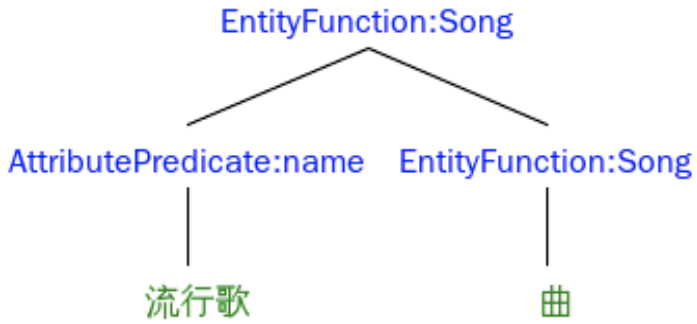
- 参数学习：
  - 基于标注好的<S，L>对
  - 最大化似然

PCCG例子

- 构建每一种结果的特征，根据特征的可信度来进行相应的打分



Song → Tag Song : 1.0
Tag → 流行: 1.2
Song → 歌曲: 1.2
总分：3.4



Song → Name Song : 0.2
Name → 流行歌: 0.3
Song → 曲: 1.2
总分：1.7

CCG的性能

- 数据集：GEO和Jobs

	Geo 880		Jobs 640	
	Precision	Recall	Precision	Recall
Our Method	96.25	79.29	97.36	79.29
COCKTAIL	89.92	79.40	93.25	79.84

CCG的扩展工作（1）：松弛的CCG文法

- 问题：CCG的文法过于严格，使得模型比较脆弱，特别是当面向口语化的文本的时候，会有大量词序错乱，省略的现象。

The lexical entries that work for:

Show me	the latest	flight	from Boston	to Prague	on Friday
S/NP …	NP/N …	N …	N\N …	N\N …	N\N …

Will not parse:

Boston	to Prague	the latest	on Friday
NP …	N\N …	NP/N …	N\N …

Luke Zettlemoyer and Michael Collins.
Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. EMNLP-2017.

CCG的扩展工作（1）：松弛的CCG文法

- 问题：CCG的文法过于严格，使得模型比较脆弱，特别是当面向口语化的文本的时候，会有大量词序错乱，省略的现象。

- 解决方法：使用松弛的CCG文法
  - Add application and composition rules that relax word order
  - Add type shifting rules to recover missing words

Luke Zettlemoyer and Michael Collins.
Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. EMNLP-2017.

松弛的CCG

- Relaxed Application（允许词序的变化）

```
X/Y : f     Y : a    =>   X : f(a)
Y : a     X\Y : f    =>   X : f(a)
```

```
X\Y : f     Y : a    =>   X : f(a)
Y : a     X/Y : f    =>   X : f(a)
```

flights	one way
N	**N/N**
$\lambda x.flight(x)$	$\lambda f.\lambda x.f(x) \wedge one\_way(x)$
N	
$\lambda x.flight(x) \wedge one\_way(x)$	

松弛的CCG

- Relaxed Composition（允许短语的前置或者后置）

X/Y : f Y/Z : g => X/Z : λx.f(g(x))
Y\Z : g X\Y : f => X\Z : λx.f(g(x))

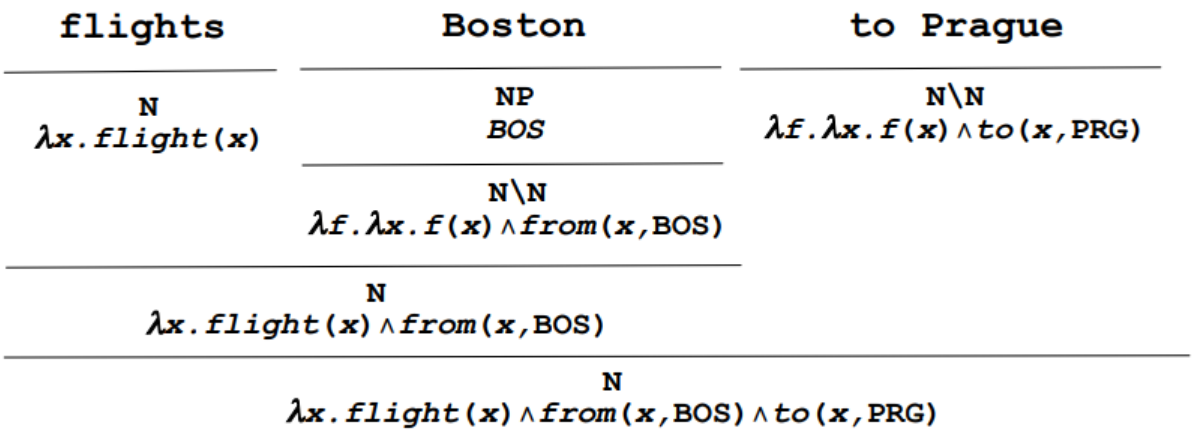
X\Y : f Y/Z : g => X/Z : λx.f(g(x))
Y\Z : g X/Y : f => X\Z : λx.f(g(x))

to Prague	the latest	flight
N\N λf.λx.f(x)∧to(x,PRG)	NP/N λf.argmax(λx.f(x),λx.time(x))	N λx.flight(x)
NP\N λf.argmax(λx.f(x)∧to(x,PRG), λx.time(x))		
N argmax(λx.flight(x)∧to(x,PRG), λx.time(x))		

松弛的CCG

- Type-shifting（允许省略）

- NP : c => N\N : λf.λx.f(x) ∧ p(x,c)



松弛的CCG的效果

ATIS数据集	Precision	Recall	F1
Single-Pass	90.61	81.92	**86.05**
Two-Pass	85.75	84.60	85.16

GEO数据集	Precision	Recall	F1
Single-Pass	95.49	83.20	**88.93**
Two-Pass	91.63	86.07	88.76
Zettlemoyer & Collins 2005	96.25	79.29	86.95
Wong & Money 2007	93.72	80.00	86.31

与ZC05相比，ZC07的召回率大大提升，准确率稍有下降

松弛的CCG的效果

ATIS数据集

	Precision	Recall	F1
Single-Pass	90.61	81.92	**86.05**
Two-Pass	85.75	84.60	85.16

GEO数据集

	Precision	Recall	F1
Single-Pass	95.49	83.20	**88.93**
Two-Pass	91.63	86.07	88.76
Zettlemoyer & Collins 2005	96.25	79.29	86.95
Wong & Money 2007	93.72	80.00	86.31

Two-Pass: 当single-Pass无法解析时，允许在second-pass中skip词语（加惩罚）

Two-pass与Single-Pass相比，同样召回率上升，准确率下降

CCG的扩展工作（2）：因子化词汇

- 问题：词典在CCG parsing中至关重要，而词典稀疏，难以学习
- 观察：一个词语可以映射到多个类型，一个类型又会在多个词汇中共享

(1) $flight \vdash N : \lambda x.flight(x)$
(2) $flight \vdash N/(S|NP) : \lambda f \lambda x.flight(x) \wedge f(x)$
(3) $flight \vdash N\backslash N : \lambda f \lambda x.flight(x) \wedge f(x)$
(4) $fare \vdash N : \lambda x.cost(x)$
(5) $fare \vdash N/(S|NP) : \lambda f \lambda x.cost(x) \wedge f(x)$
(6) $fare \vdash N\backslash N : \lambda f \lambda x.cost(x) \wedge f(x)$
(7) $Boston \vdash NP : bos$
(8) $Boston \vdash N\backslash N : \lambda f \lambda x.from(x, bos) \wedge f(x)$
(9) $New\ York \vdash NP : nyc$
(10) $New\ York \vdash N\backslash N : \lambda f \lambda x.from(x, nyc) \wedge f(x)$

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical generalization in ccg grammar induction for semantic parsing. EMNLP-2011.

CCG的扩展工作（2）：因子化词汇

- 问题：词典在CCG parsing中至关重要，而**词典稀疏**难以学习
- 观察：一个词语可以映射到多个类型，一个类型又会在多个词汇中共享

- 方法：因子化词汇
  - Lexeme: flight: [flight]
  - Lexical template: flight

    $\lambda(\omega,v).[\omega \vdash N : \lambda x.v_1(x)]$

    $\lambda(\omega,v).[\omega \vdash N/(S|NP):\lambda\, f\, \lambda x.v_1(x) \wedge f(x)]$

(1) $flight \vdash N : \lambda x.flight(x)$
(2) $flight \vdash N/(S|NP) : \lambda f \lambda x.flight(x) \wedge f(x)$
(3) $flight \vdash N\backslash N : \lambda f \lambda x.flight(x) \wedge f(x)$
(4) $fare \vdash N : \lambda x.cost(x)$
(5) $fare \vdash N/(S|NP) : \lambda f \lambda x.cost(x) \wedge f(x)$
(6) $fare \vdash N\backslash N : \lambda f \lambda x.cost(x) \wedge f(x)$
(7) $Boston \vdash NP : bos$
(8) $Boston \vdash N\backslash N : \lambda f \lambda x.from(x, bos) \wedge f(x)$
(9) $New\ York \vdash NP : nyc$
(10) $New\ York \vdash N\backslash N : \lambda f \lambda x.from(x, nyc) \wedge f(x)$

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical generalization in ccg grammar induction for semantic parsing. EMNLP-2011.

基于DCS的语义解析

DCS (Dependency-based compositional semantics)

- 由Percy Liang于2011年提出（博士论文仅这一个工作），后于2013年在此基础上提出λ –DCS，使其表示能力更强。
- DCS是一种语义表示体系，同时包含了从句子到语义表示所需的词典和组合文法
- 与CCG类似，基于组合语义的思想，依赖于**词典**和**组合文法**

Percy Liang, Michael I. Jordan, Dan Klein. Learning dependency-based compositional semantics. ACL-2011.

DCS的特点

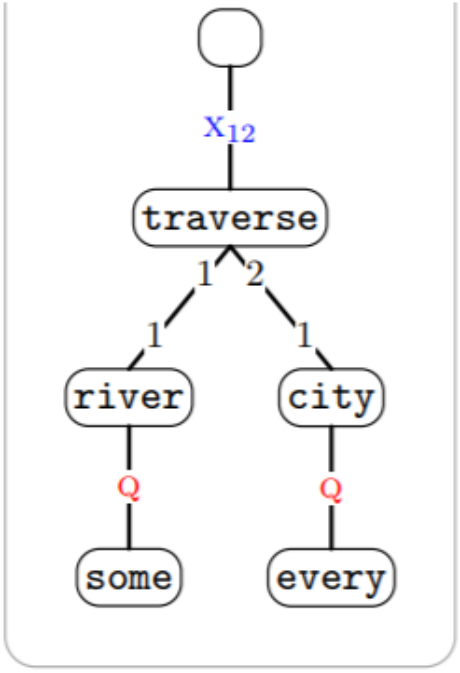
- **表示能力强**：需要与lambda-表达式具备等价的语义表示能力
- **简洁**：与lambda-表达式相比，DCS的表示更加简洁
- **结构清晰**：保有与自然语言类似的结构

DCS的特点——表示能力强

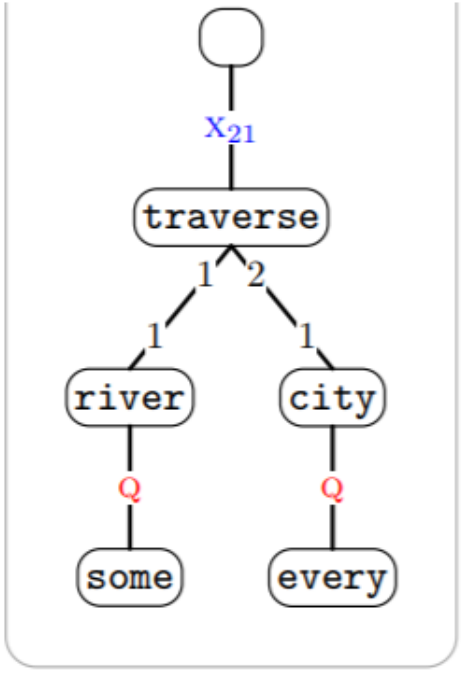
- 示例：量词（some, every, one, etc.）辖域问题

Some river traverses every city

Quantification (narrow)



Quantification (wide)



Some (river) traverses every city

Some (river traverses every city)

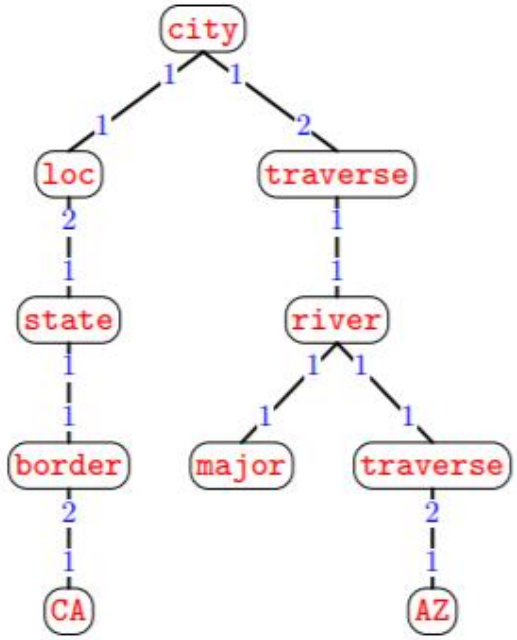
DCS的特点——简洁

Utterance: *"people who have lived in Seattle"*

Logical form (lambda calculus): $\lambda x.\exists e.\texttt{PlacesLived}(x, e) \wedge \texttt{Location}(e, \texttt{Seattle})$

Logical form (lambda DCS): `PlacesLived.Location.Seattle`

DCS的特点——结构清晰（与自然语言的结构类似）



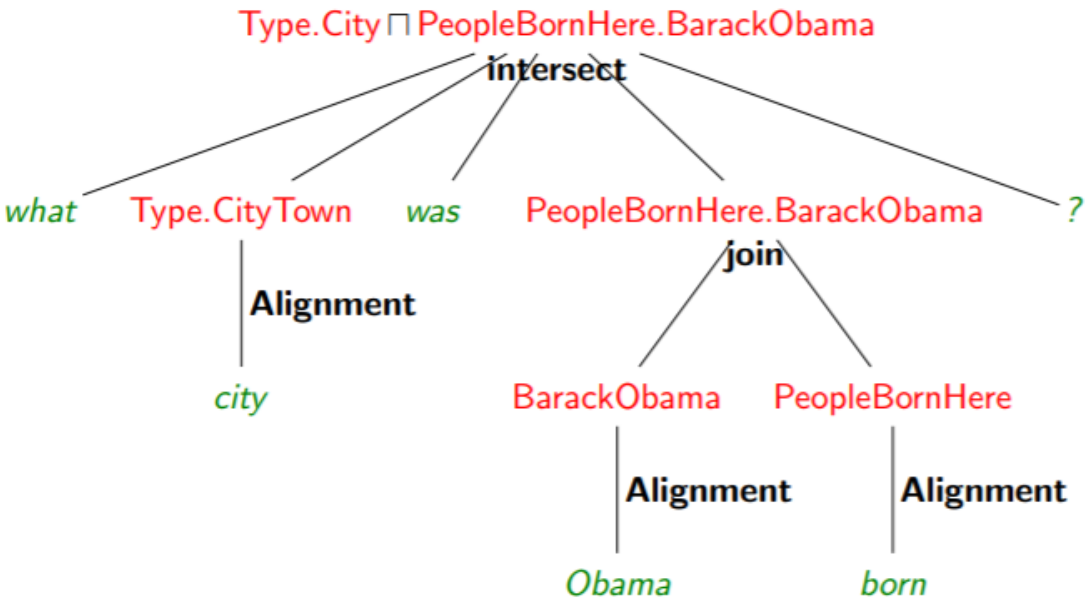
Linguistics ..... Trees ..... Computation

syntactic locality

efficient interpretation

DCS-语义

- 集合论：一个语义表示的语义是符合其语义约束的所有值的集合
  - 山东省 → {山东省} (山东)
  - PlaceOfBirth → {<莫言，山东>，<秦琼，山东>，<居里，法国>, ...}
  - Profession→ {<莫言，作家>， <秦琼，武将>，<居里，科学家>}
  - PlaceOfBirth.山东→ {莫言，秦琼, ...} (山东出生)
  - PlaceOfBirth.山东 ^ Profession.作家 → {莫言...} (山东出生的作家)



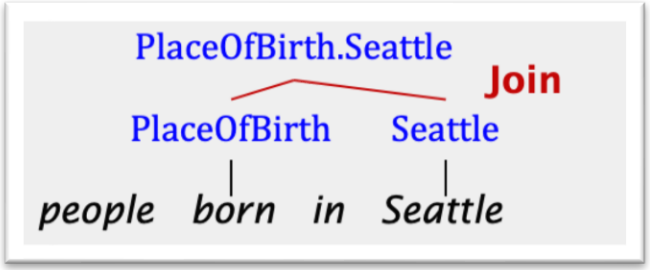
DCS-词典

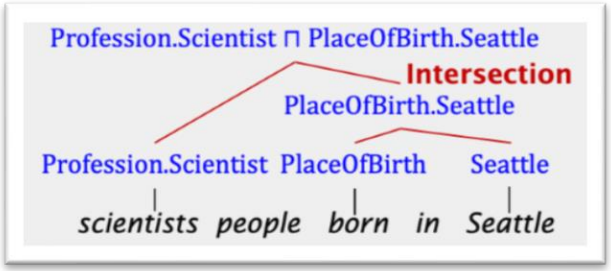
- Unary case (**u**)：其语义为实体集合
  - 实体:“Texas” → texas {texas}
  - 类型：“state” → state {纽约州、华盛顿州、密西西比州...}
  - ...

- Binary case (**b**): 其语义为符合关系的元组集合
  - 二元谓词（predicate）：“born” → PlaceOfBirth {<莫言，山东>，<秦琼，山东>，<居里，法国>, ...}

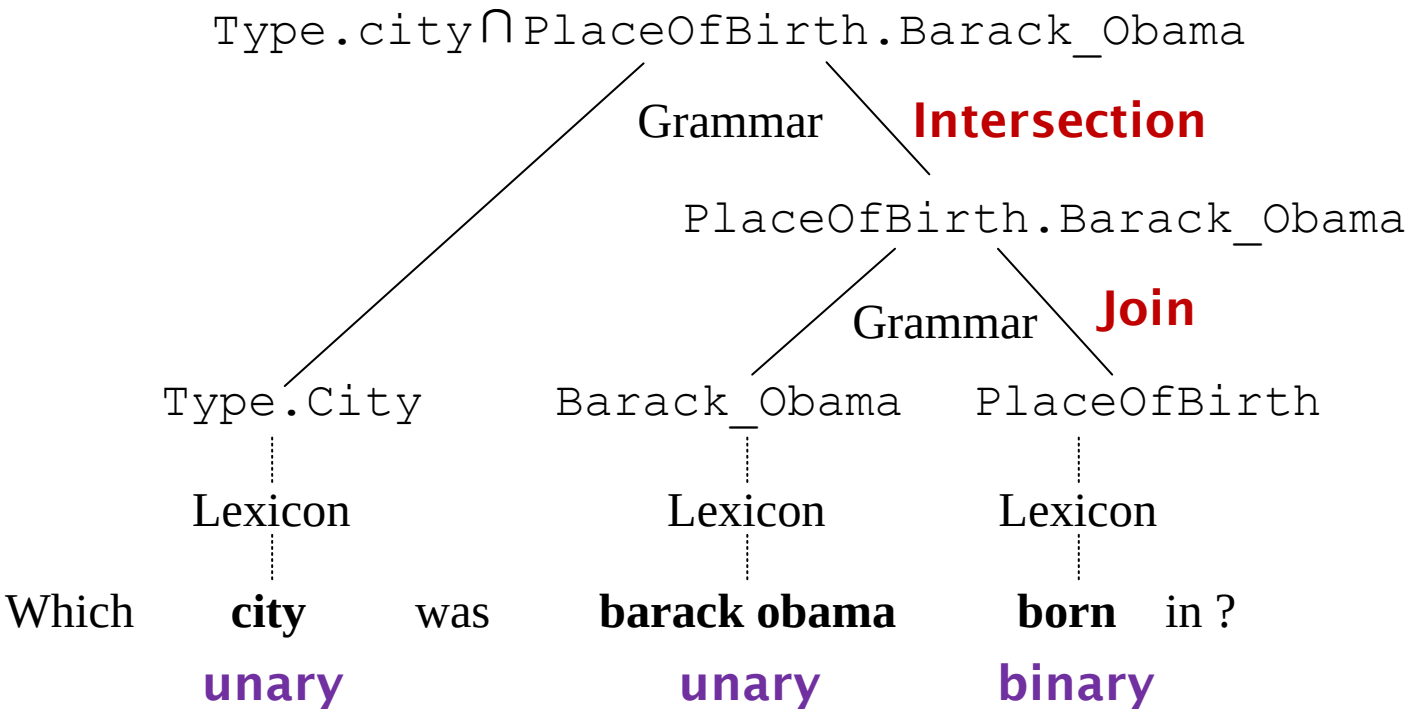
DCS-组合文法-集合和集合之间的组合和转换

- **Join（合并）**
  - binary (Join) unary → unary
  - unary (Join) binary → unary
  - 山东 出生 → 出生.山东
  - {山东} (Join) {<莫言，山东>，<秦琼，山东>，<居里，法国>, ...} → {莫言，秦琼， ...}
- **Intersection（交集）**
  - unary (Intersection) unary → unary
  - 流行歌手 → 流行 AND 歌手
- **Union (并集)**
  - unary (Union) unary → unary
  - 周杰伦或刘德华 → 周杰伦 OR 刘德华
- **高阶函数**
  - Count (计数)：Count unary → unary
  - argmax, argmin, max, min等
  - 周杰伦的歌曲 数目 → Count(周杰伦的歌曲)





DCS-解析例子



Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang.
Semantic parsing on Freebase from question-answer pairs. EMNLP-2013.

基于词典-组合文法的语义解析

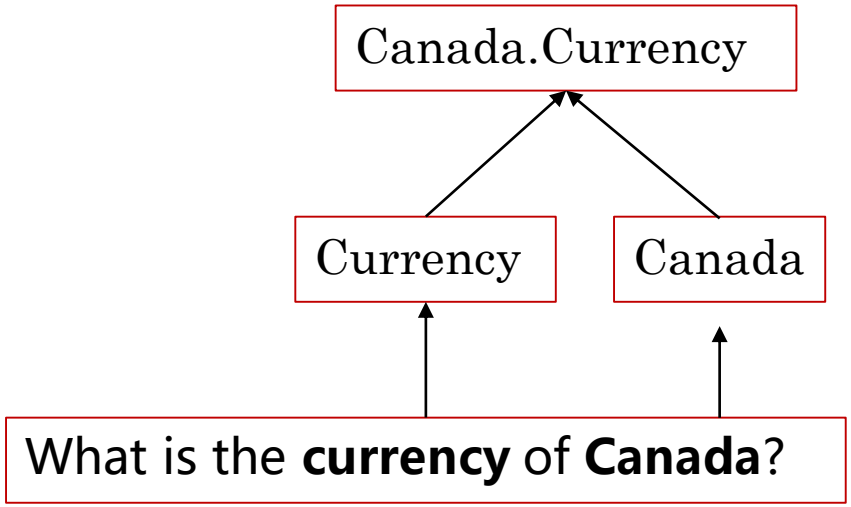
从限定域到开放域

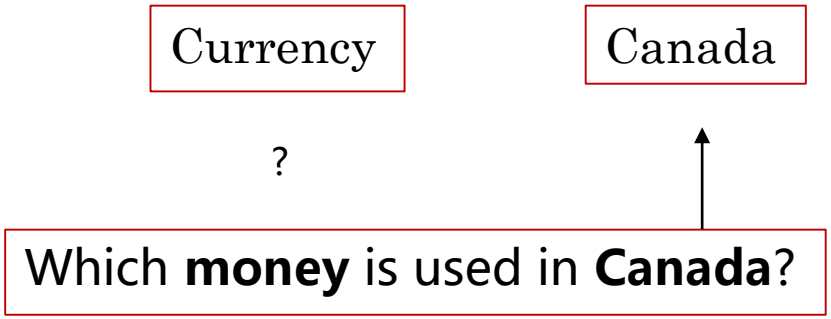
从限定域到开放域：关键问题

词典覆盖度不足	→	词典扩充学习
结构不匹配	→	复述/重写
标注语料成本过高	→	弱监督学习

词典扩充学习

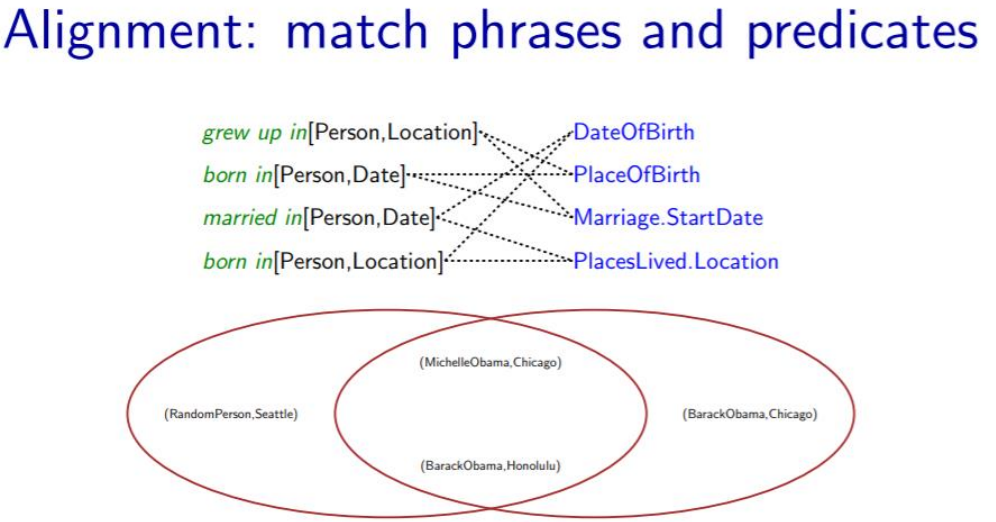
- 词典的覆盖度不足：基于有监督语料构建的词典只能覆盖有限的词语



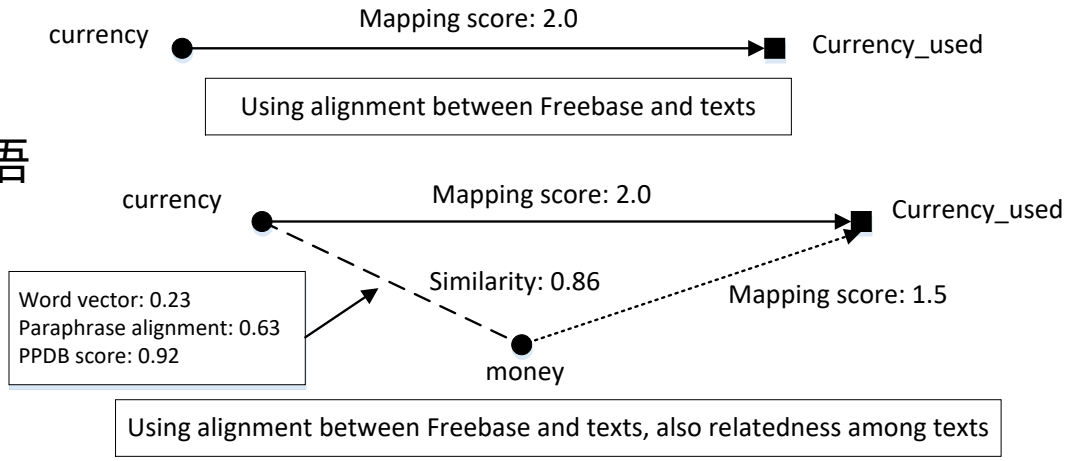


词典扩充学习代表性方法

- 使用实体对共现(Reference)
  - 基本假设：文本中的三元组与知识库中的三元组若有多个共同的实体对，则文本中的关系（词语）映射到知识库中的关系（词语语义）



- 使用标签传播算法
  - 基本假设：意思相近的词语应该映射到相同的语义



Qingqing Cai and Alexander Yates. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. ACL-2013.
Bo Chen, Bo An, Le Sun, Xianpei Han. Semi-Supervised Lexicon Learning for Wide-Coverage Semantic Parsing. Coling-2018.

复述/重写

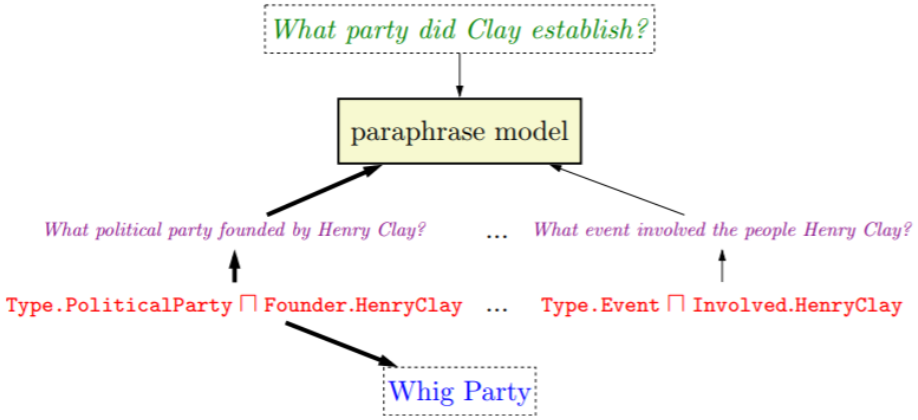
- 由自然语言多样性引起的本体不匹配问题

s_1: *What is the population of Berlin?*
s_2: *How many people live in Berlin?*
lf_1: `λx.population(Germany,x)`
lf2: `count(λx.person(x)∧live(x,Berlin))`

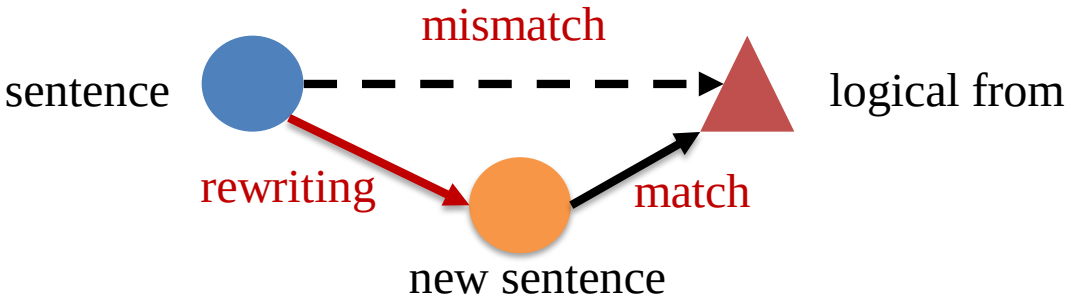
难以把s_1映射到lf_2，把s_2映射到lf_1

复述/重写

- 复述
  - 原理：将语义解析反过来，先得到句子的语义表示，然后生成新的句子，利用复述模型判断新句子与原句子是否构成复述关系



- 重写
  - 原理：一个句子存在另外一种表达，该表达与原句子具有相同的语义，并且其结构与目标语义表示的结构一致



Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. EMNLP-2014.
Bo Chen, Le Sun, Xianpei Han, Bo An. Sentence Rewriting for Semantic Parsing. ACL-2016.

弱监督学习

- 语义表示难以标注

句子: what is the highest point in florida ?
Lambda-表达式: _answer (A , _highest (A , (_place (A) , _loc (A , B) , _const (B , _stateid (florida)))))

句子: what are the high points of states surrounding mississippi ?
Lambda-表达式: _answer (A , (_high_point (B , A) , _loc (A , B) , _state (B) , _next_to (B , C) , _const (C , _stateid (mississippi))))

句子: what state has the shortest river ?
Lambda-表达式: _answer (A , (_state (A) , _loc (B , A) , _shortest (B , _river (B))))

弱监督学习

- 利用语义表示的执行结果（如问题的答案）来监督模型的学习
  - 利用冗余的弱监督信息也能指导模型的准确学习

Where did Mozart tupress?

~~PlaceOfBirth.WolfgangMozart ⇒ Salzburg~~

PlaceOfDeath.WolfgangMozart ⇒ Vienna

PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

Where did Hogarth tupress?

PlaceOfBirth.WilliamHogarth ⇒ London

PlaceOfDeath.WilliamHogarth ⇒ London

~~PlaceOfMarriage.WilliamHogarth ⇒ Paddington~~

London

Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang.
Semantic parsing on Freebase from question-answer pairs. EMNLP-2013.

小结：基于词典-组合文法的语义解析方法

- **核心组件：**词典+组合文法+学习算法
- **代表性方法：**CCG，DCS
- **从限定域到开放域**：词典扩充学习、复述/句子重写、弱监督学习

- 优点：
  - 基于组合语义的思想
  - 有词典和组合规则，整个解析过程可见，可解释性强

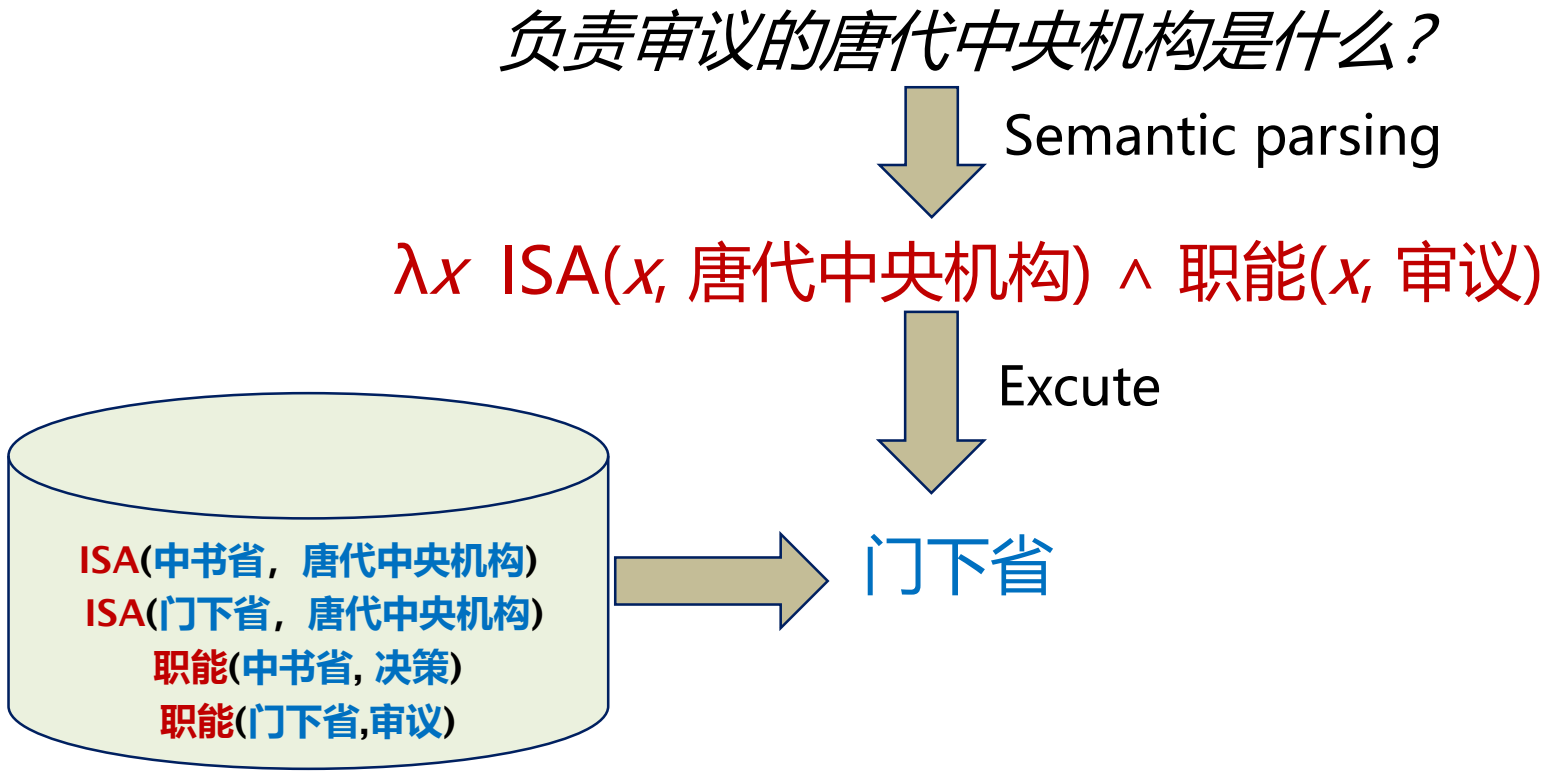
- 缺点：
  - 太依赖于词典，若有词典未覆盖的词语，就无法解析
  - 脆弱，依赖于组合文法，有些复杂的语言现象，文法也无法覆盖

大纲

- 语义解析任务介绍
- 基于词典-文法的语义解析
- 基于语义图的语义解析
- 基于神经网络的语义解析
- 总结和展望

基于词典-组合文法语义解析的问题

- 需要提前学习词典和定义文法
- 语义表示与知识库联系不紧密，无法在解析过程中利用知识约束



基于语义图的语义解析

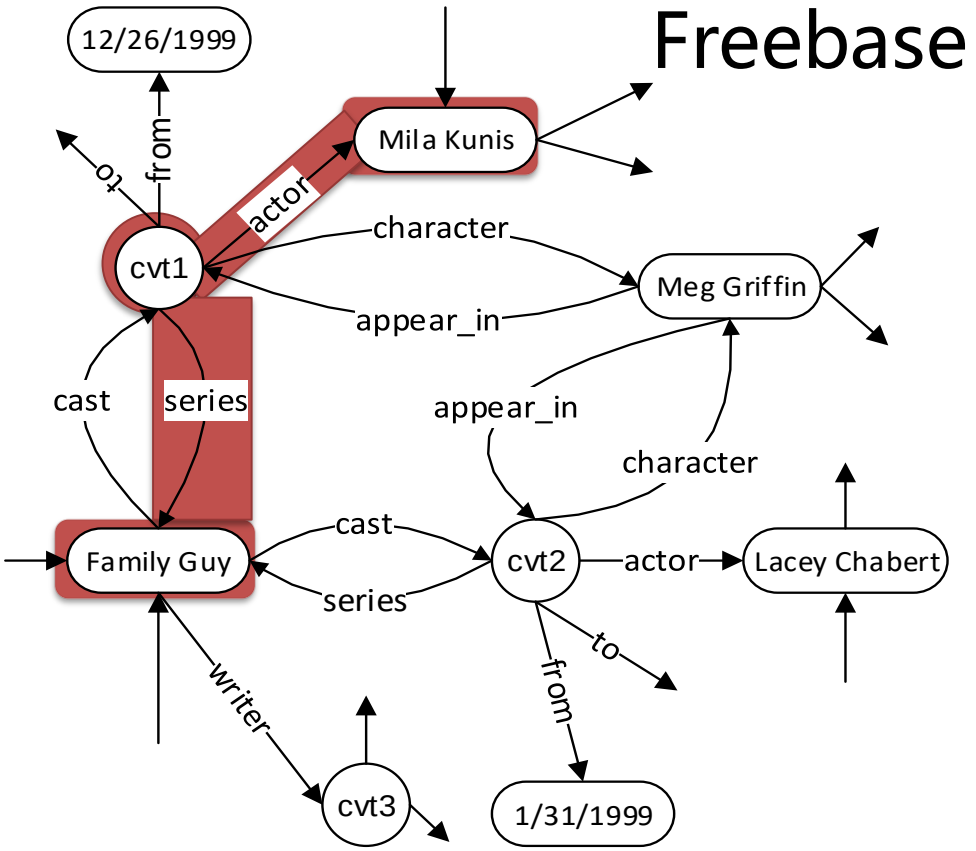
- 关键组件：
  - 语义图表示
  - 语义图构建
- 使用语义图作为目标语义表示
  - 优点：语义图的结构与自然语言句子的结构具有类似性
- 语义解析过程由组合文法转换为语义图构建
  - 优点：可以充分使用知识库知识的约束，从而有效减少搜索空间

什么是语义图

- 语义图可以看做是知识库的一个子图

- 语义图的基本组成为：
  - 节点，如实体、变量或者类型
  - 边，表示关系
  - 操作符，如count, argmax等

Mila Kunis is the actor of Family Guy



语义图

语义图构建代表性方法

- DepLambda [Reddy et al, 2016]：依存树到语义图转换文法
- STAGG [Yih, et al., 2015]: 分步骤构建方法
- Template [Bast & Haussmann, 2015]: 基于模板的语义图生成

DEPLAMBDA：依存树到语义图转换

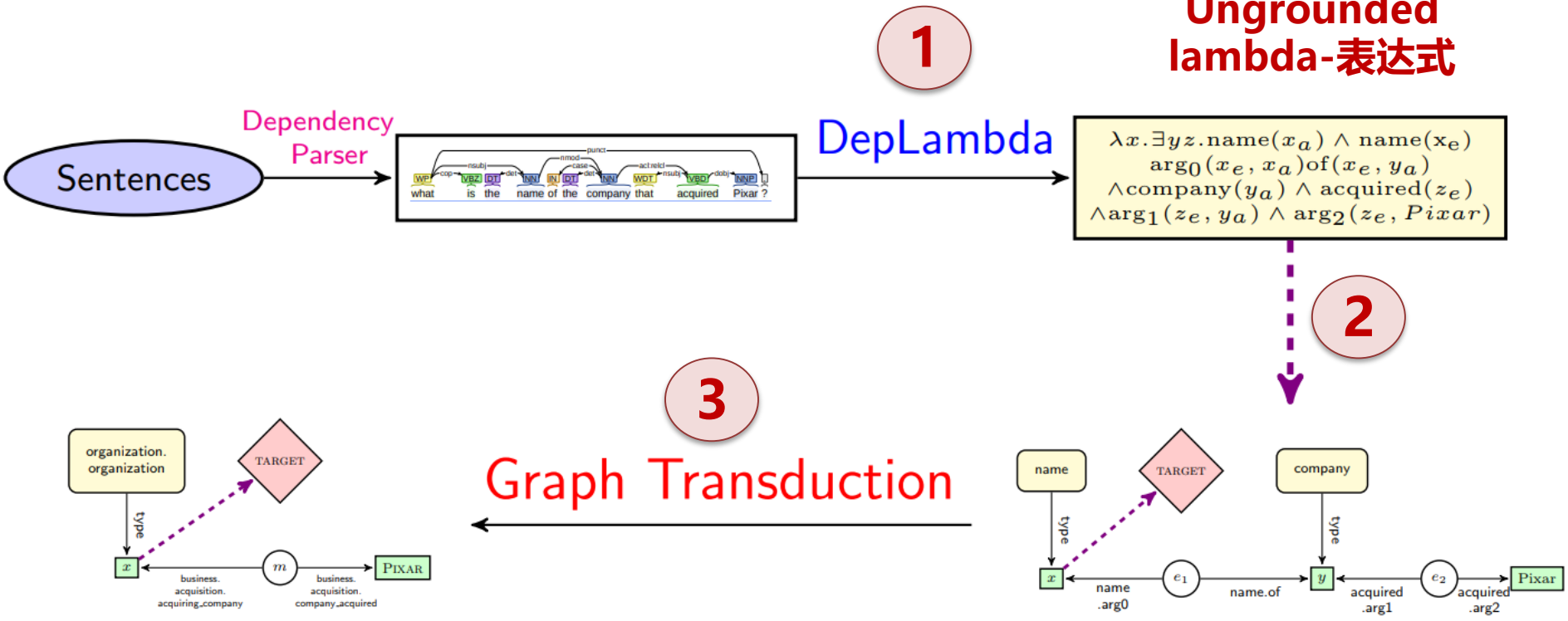
Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, Mirella Lapata. Transforming Dependency Structures to Logical Forms for Semantic Parsing. TACL-2016.

DepLambda

- 原理：依存树与目标逻辑表达式具有类似的结构

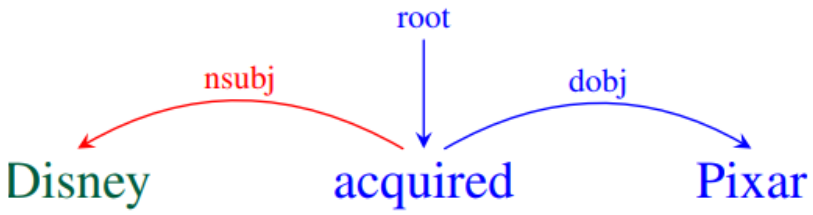
- 步骤：
  - 句子→依存树
  - 依存树→deplambda
  - Deplambda→ungrounded graph
  - Ungrounded graph → grounded semantic graph

DepLambda解析步骤



1 Dependencies to Logical Forms

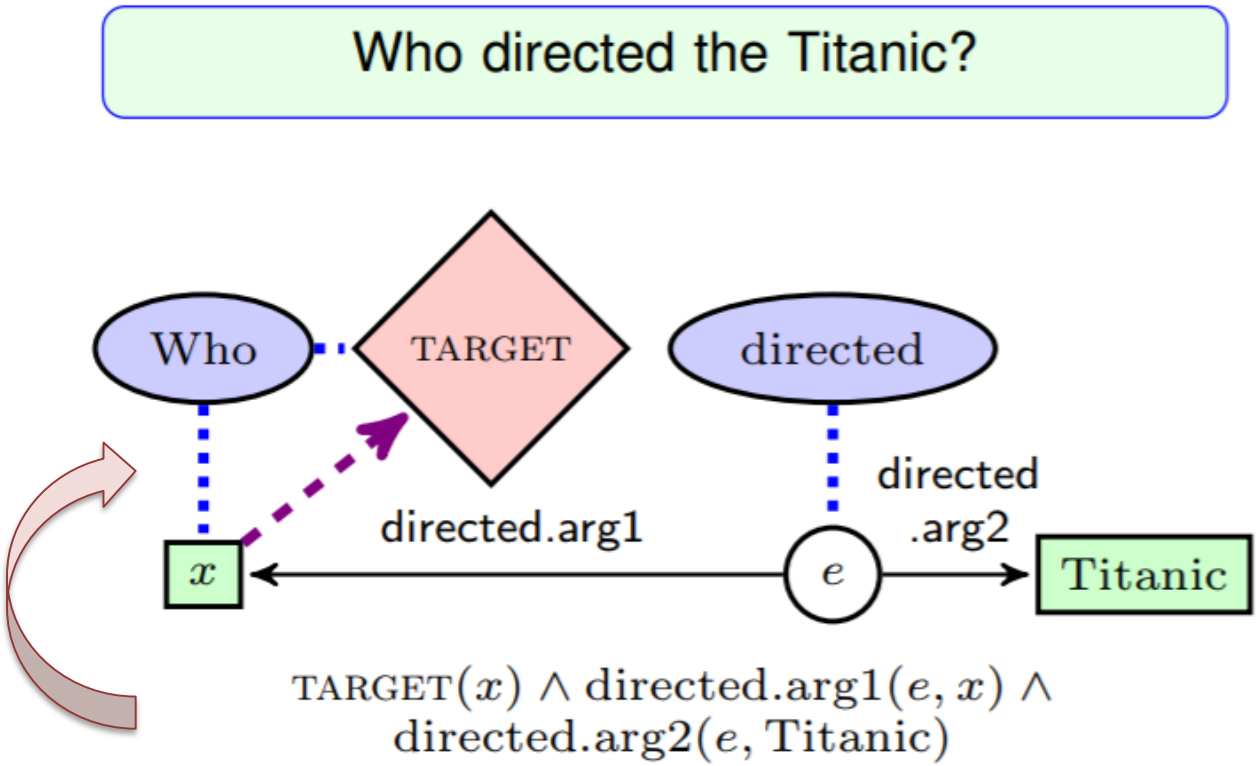
- 利用依存关系来组合



$$\frac{\dfrac{\dfrac{\textbf{(nsubj}}{\lambda f g z.\ \exists x.\ f(z) \wedge g(x) \wedge \arg_1(z_e, x_a)} \quad \frac{\textbf{(dobj acquired Pixar)}}{\lambda z.\ \exists y.\ \text{acquired}(z_e) \wedge \text{Pixar}(y_a) \wedge \arg_2(z_e, y_a)}}{\lambda g z. \exists x y. \text{acquired}(z_e) \wedge \text{Pixar}(y_a) \wedge \text{g}(x) \wedge \arg_1(z_e, x_a) \wedge \arg_2(z_e, y_a)} \quad \frac{\textbf{Disney)}}{\lambda x. \text{Disney}(x_a)}}{\lambda z. \exists x y. \text{acquired}(z_e) \wedge \text{Pixar}(y_a) \wedge \text{Disney}(x_a) \wedge \arg_1(z_e, x_a) \wedge \arg_2(z_e, y_a)}$$

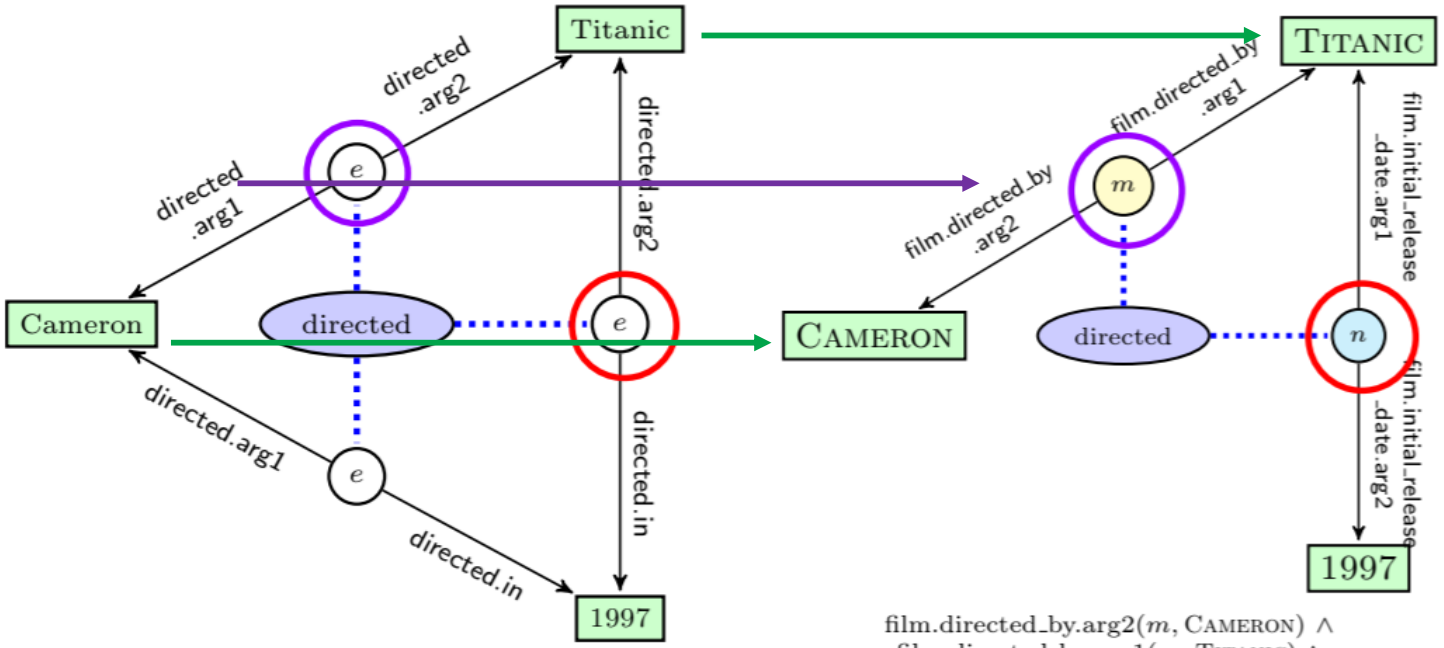
ungrounded的lambda-表达式

2 Map Logical Form to Ungrounded Graph



3 Map Ungrounded Graph to Semantic Graph

- Grounding
  - 实体词 链接到知识库中的实体，如*Titanic* → TITANIC
  - 关系词 链接到知识库中的关系，如*directed* → film.directed_by



$\text{directed.arg1}(e, \text{Cameron}) \wedge$
$\text{directed.arg2}(e, \text{Titanic}) \wedge \text{directed.in}(e, 1997)$

$\text{film.directed\_by.arg2}(m, \text{CAMERON}) \wedge$
$\text{film.directed\_by.arg1}(m, \text{TITANIC}) \wedge$
$\text{film.initial\_release\_date.arg1}(n, \text{TITANIC}) \wedge$
$\text{film.initial\_release\_date.arg2}(n, 1997)$

Ungrounded Graph

Grounded Graph

STAGG：分步构建语义图

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, Jianfeng Gao.
Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. ACL-2015.

STAGG

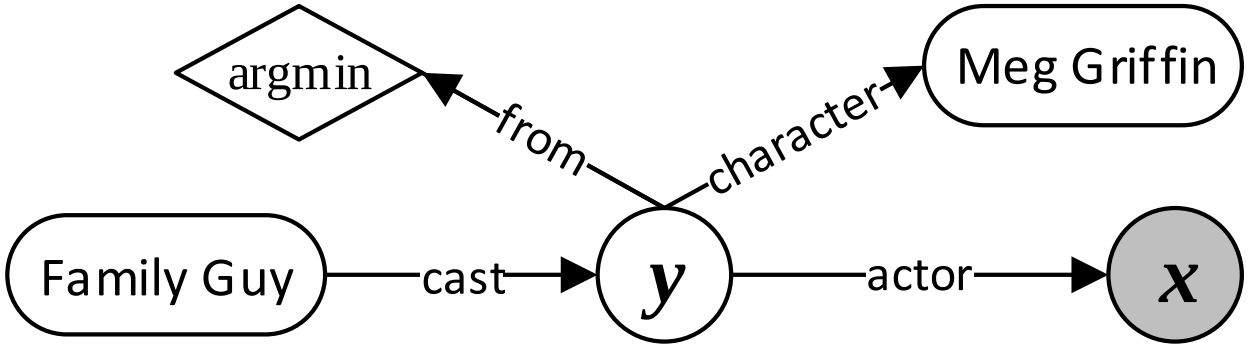
- 主要可分为三步：
  - 中心实体链接
  - 关系链匹配
  - 添加约束条件

- 分步构建语义图的好处
  - 每一步都可以使用最好的技术
  - 每进行一步都可以利用知识库的约束来减少搜索空间

STAGG

Who first voiced Meg on Family Guy?

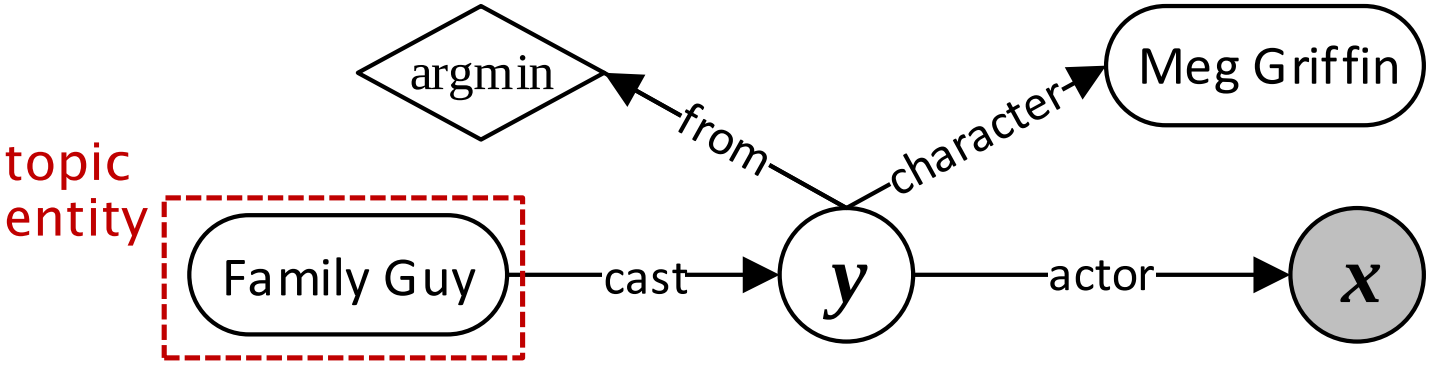
$\lambda x.\, \exists y.\, \text{cast}(\text{FamilyGuy}, y) \wedge \text{actor}(y, x) \wedge \text{character}(y, \text{MegGriffin})$



STAGG

Who first voiced Meg on **Family Guy**?

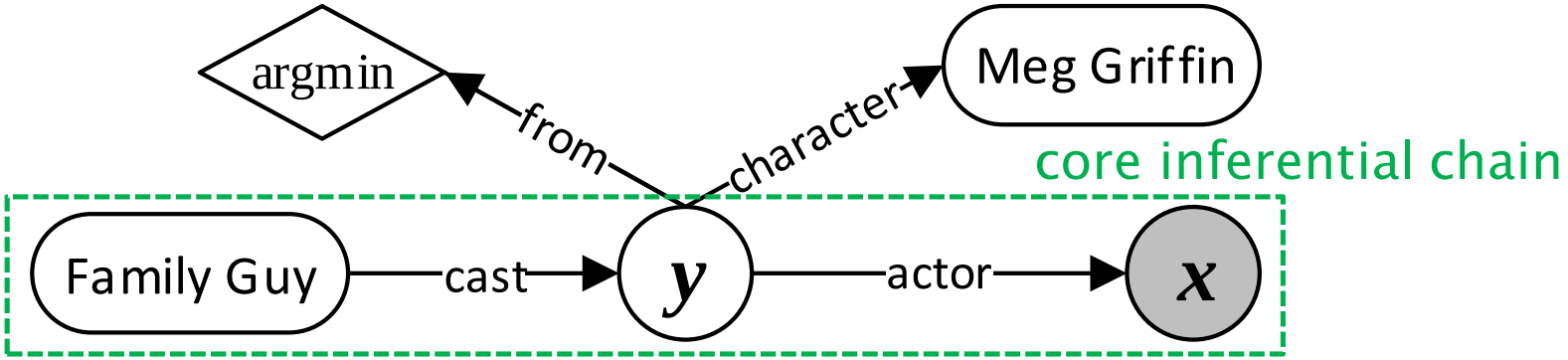
$$\lambda x.\, \exists y.\, \text{cast}(\text{FamilyGuy}, y) \wedge \text{actor}(y, x) \wedge \text{character}(y, \text{MegGriffin})$$



STAGG

Who first **voiced** Meg on Family Guy?

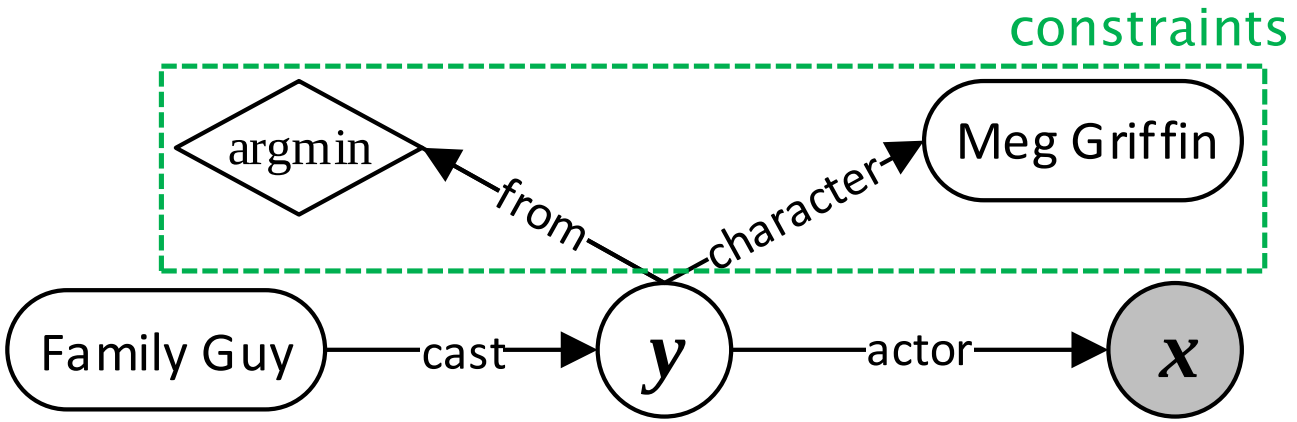
$$\lambda x.\, \exists y.\, \text{cast}(\text{FamilyGuy}, y) \wedge \text{actor}(y, x) \wedge \text{character}(y, \text{MegGriffin})$$



STAGG

Who first voiced Meg on Family Guy?

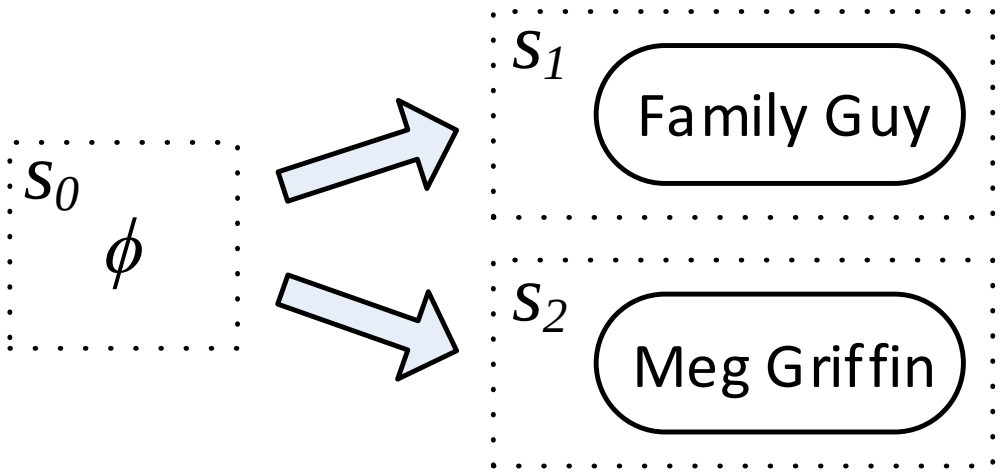
$\lambda x.\, \exists y.\, \text{cast}(\text{FamilyGuy}, y) \wedge \text{actor}(y, x) \wedge \text{character}(y, \text{MegGriffin})$



STAGG：（1）中心实体链接

- Entity linking technology

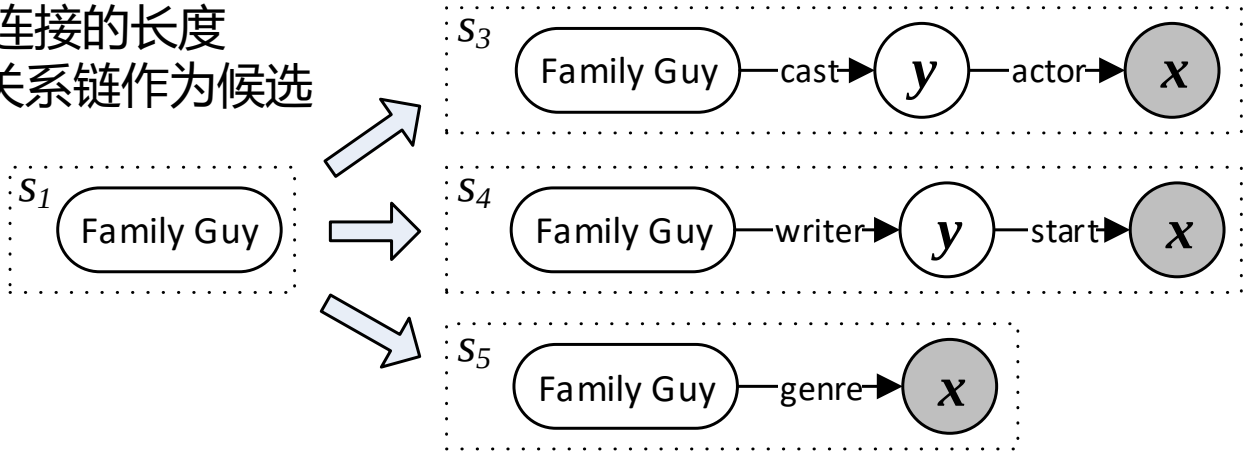
Who first voiced Meg on Family Guy?



STAGG：（2）关系链匹配

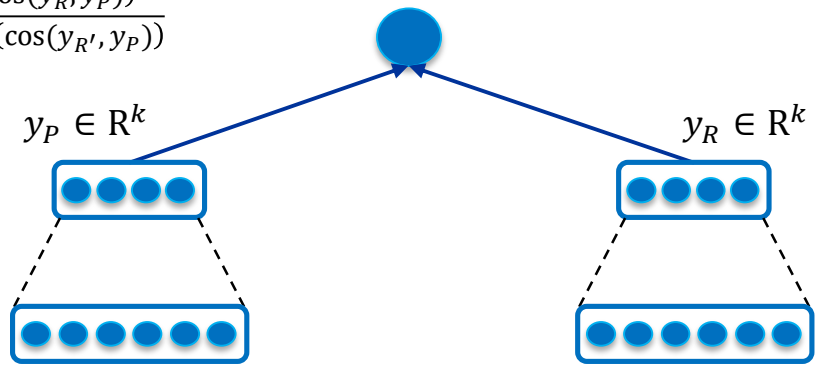
Who first voiced Meg on Family Guy?

与中心实体连接的长度
不超过2的所有关系链作为候选



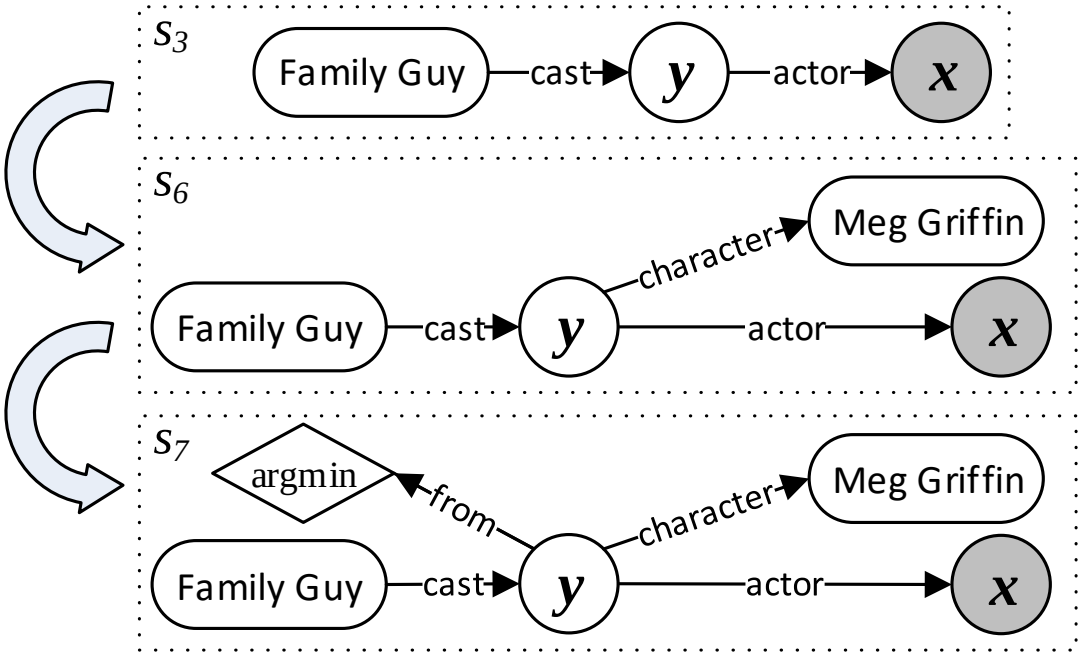
$$P(R|P) = \frac{\exp(\cos(y_R, y_P))}{\sum_{R'} \exp(\cos(y_{R'}, y_P))}$$

Deep Convolutional
Neural Networks



STAGG：（3）添加约束条件

Who first voiced Meg on Family Guy?



STAGG的性能（WebQuestions数据集）

- 特点：高召回率
  - 并不需要词典，也不受限于词典的覆盖度
  - 神经网络模型来建模软匹配过程，能处理模糊匹配的情况，解决OOV问题

Method	Prec.	Rec.	F_1
(Berant et al., 2013)	48.0	41.3	35.7
(Bordes et al., 2014b)	-	-	29.7
(Yao and Van Durme, 2014)	-	-	33.0
(Berant and Liang, 2014)	40.5	46.6	39.9
(Bao et al., 2014)	-	-	37.5
(Bordes et al., 2014a)	-	-	39.2
(Yang et al., 2014)	-	-	41.3
(Wang et al., 2014)	-	-	45.3
Our approach – STAGG	52.8	60.7	**52.5**

TEMPLATE：基于模板的语义图构建

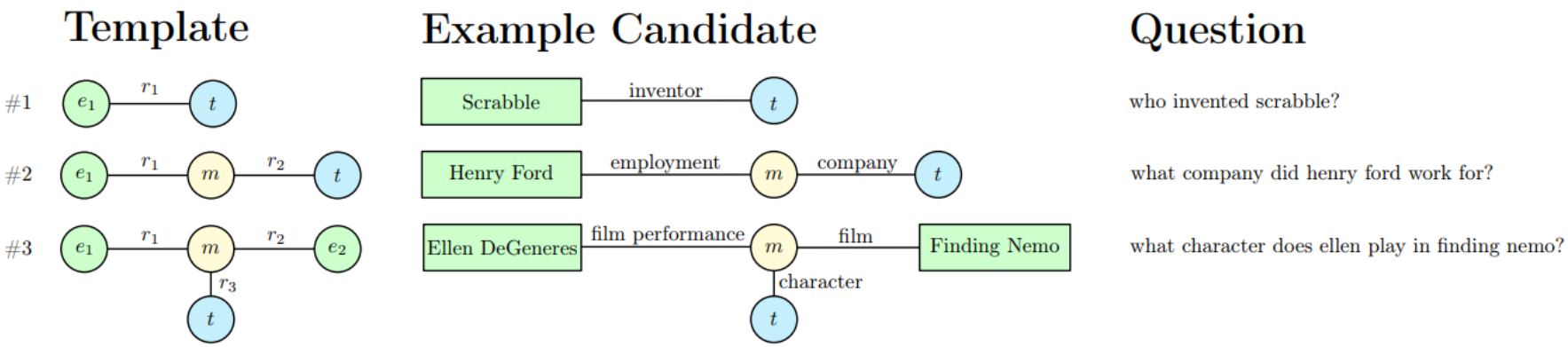
Hannah Bast and Elmar Haussmann. More Accurate Question Answering on Freebase. CIKM-2015.

Template

- 查询往往都具有固定的若干个模板（template）
  - 可通过设计有限的模板来覆盖绝大部分的查询
- 基于template的方法可以把语义图构建转换为模板匹配问题，能够大大简化此问题
- 关键：
  - 模板的定义
  - 模板匹配

Template：模板的定义

- 3个模板覆盖Webquestions和Free917数据集中绝大部分查询



模板	适应问题形式	问题例子
#1	一个关系 + 一个实体	姚明的爸爸是谁?
#2	两个关系 + 一个实体（外加一个虚实体）	科比效力于哪个球队?
#3	三个关系 + 两个实体（外加一个虚实体）	吴京在流浪地球中扮演什么角色?

Template

- 模板匹配
  - 实体链接

what character does ellen play in finding nemo?

Ellen DeGeneres　　Finding Nemo

  - 匹配模板

模板1：<Ellen DeGeneres> <parents> <T>

模板2：<Ellen Page> <performance> <M>
<M> <film> <T>

模板3：<Ellen DeGeneres> <performance> <M>
<M> <film> <Finding Nemo>
<M> <character> <T>

  - 关系匹配

<Ellen DeGeneres> <performance> <M>
<M> <film> <Finding Nemo>
<M> <character> <T>

Template的性能: Webquestions数据集

- 高召回率
  - 模板能够覆盖大部分的问题，简化了语义图构建的过程

model	Precision	Recall	Avg F1
Berant et al., 2013	0.48	0.413	0.357
Berant and Liang, 2014	0.405	0.466	0.399
Yao and Van Durme, 2014	0.517	0.458	0.330
Wang et al., 2014	0.447	0.525	0.483
Yao, 2015	0.526	0.545	0.443
Template	0.498	**0.604**	0.494

小结：基于语义图的语义解析方法

- **核心：**语义图的语义表示、语义图的构建
- **代表性方法：**Dependency转换、Template转换、分步生成

- 优点：
  - 不需要词典和组合文法
  - 语义图的结构与句子的依存结构具有相似性
  - 与知识库紧密联系，可充分利用知识库知识的约束

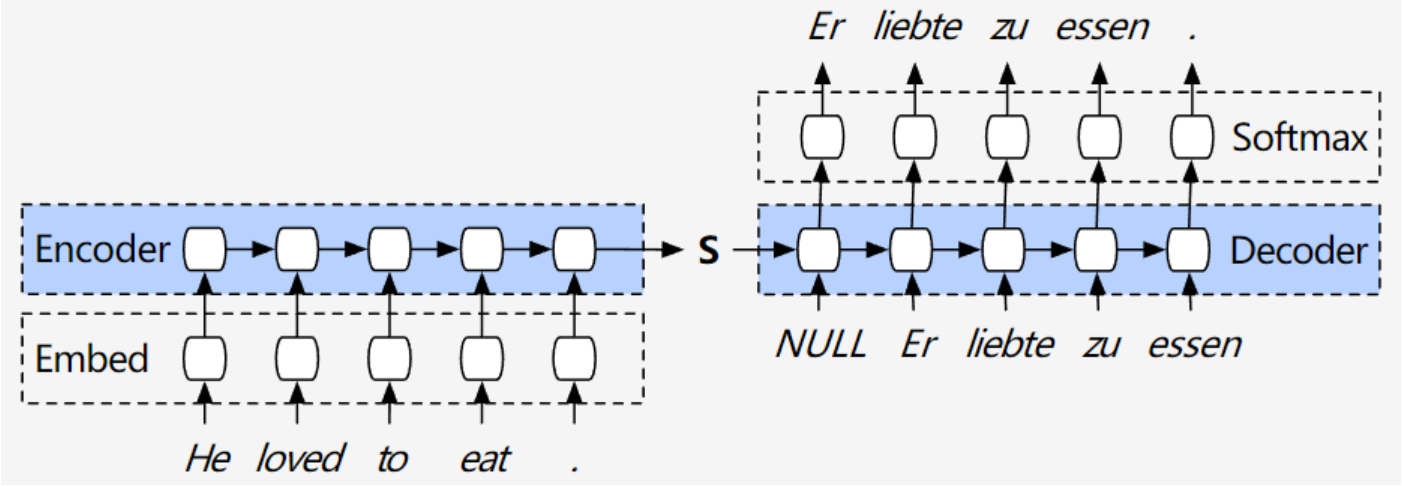
- 缺点：
  - 往往依赖于一些启发式的方法来构建语义图

大纲

- 语义解析任务介绍
- 基于词典-文法的语义解析
- 基于语义图构建的语义解析
- 基于神经网络的语义解析
- 前沿方向
- 总结和展望

基于神经网络的语义解析兴起前的背景

- 之前的模型都很复杂，模块很多
  - 基于词典-组合文法方法中的词典学习，基于语义图方法中的语义图构建
- 人工定义特征
- 神经网络模型在NLP其他任务上取得了成功
  - 基于encoder-decoder框架的神经机器翻译



代表性方法

- **Seq2Seq** [Dong & Lapata, 2016; Jia & Liang, 2016], **Seq2Tree** [Dong & Lapata, 2016]

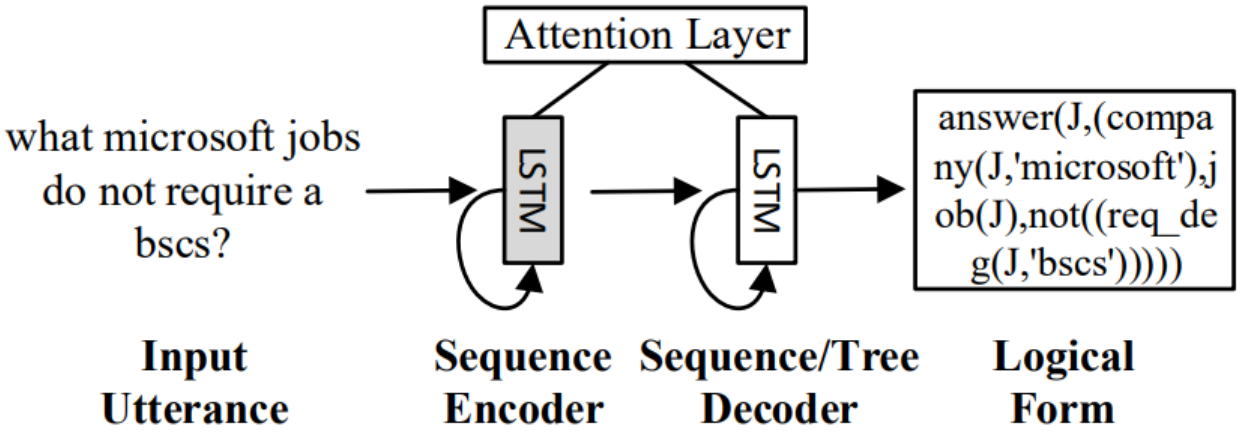
- **Seq2Act** [Chen et al., 2018]

SEQ2SEQ AND SEQ2TREE

Li Dong, Mirella Lapata. Language to Logical Form with Neural Attention. ACL-2016.
Robin Jia and Percy Liang. Data recombination for neural semantic parsing. ACL-2016.

Seq2Seq：把语义解析看做是机器翻译问题

- 将逻辑表达式序列化，看作一系列token，从而将语义解析转化为Seq2Seq问题



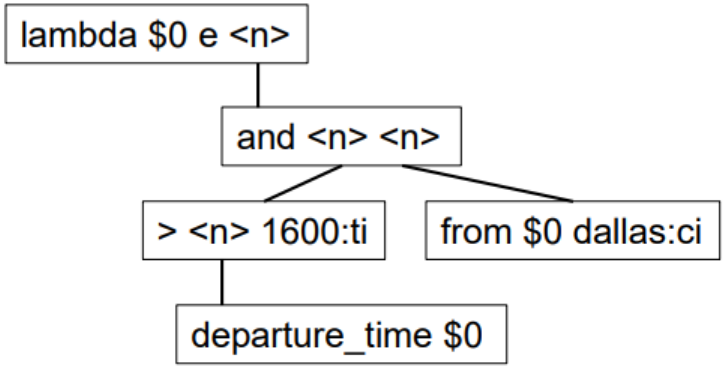
Li Dong, Mirella Lapata. Language to Logical Form with Neural Attention. ACL-2016.

Robin Jia and Percy Liang. Data recombination for neural semantic parsing. ACL-2016.

Seq2Seq中的问题

- 语义解析中的目标语言（语义表示）具有层次结构，而Seq2Seq模型仅把语义表示扁平序列化，从而忽略了层次结构信息

结构化表示



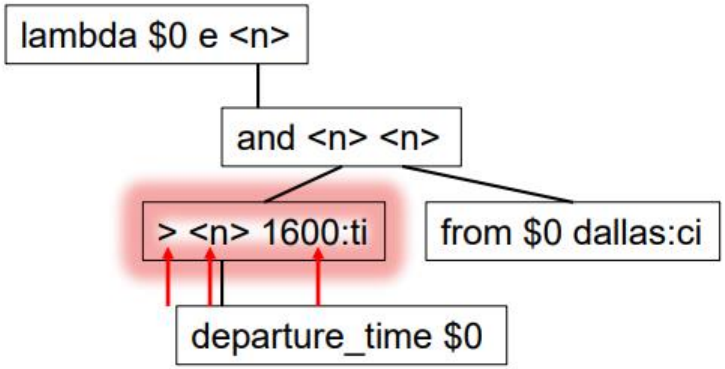
扁平化表示 lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci))

Seq2Seq中的问题

- 语义解析中的目标语言（语义表示）具有层次结构，而Seq2Seq模型仅把语义表示扁平序列化，从而忽略了层次结构信息

- 导致在解码过程中需要考虑更多的长距离依赖

结构化表示



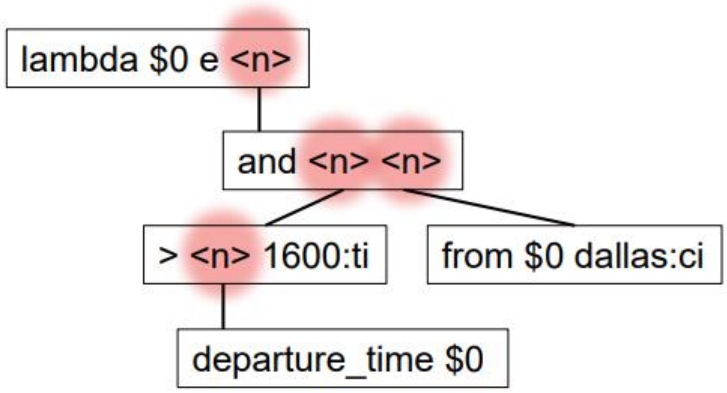
扁平化表示

lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci))

句子 dallas to san francisco leaving after 4 in the afternoon please

Seq2Tree

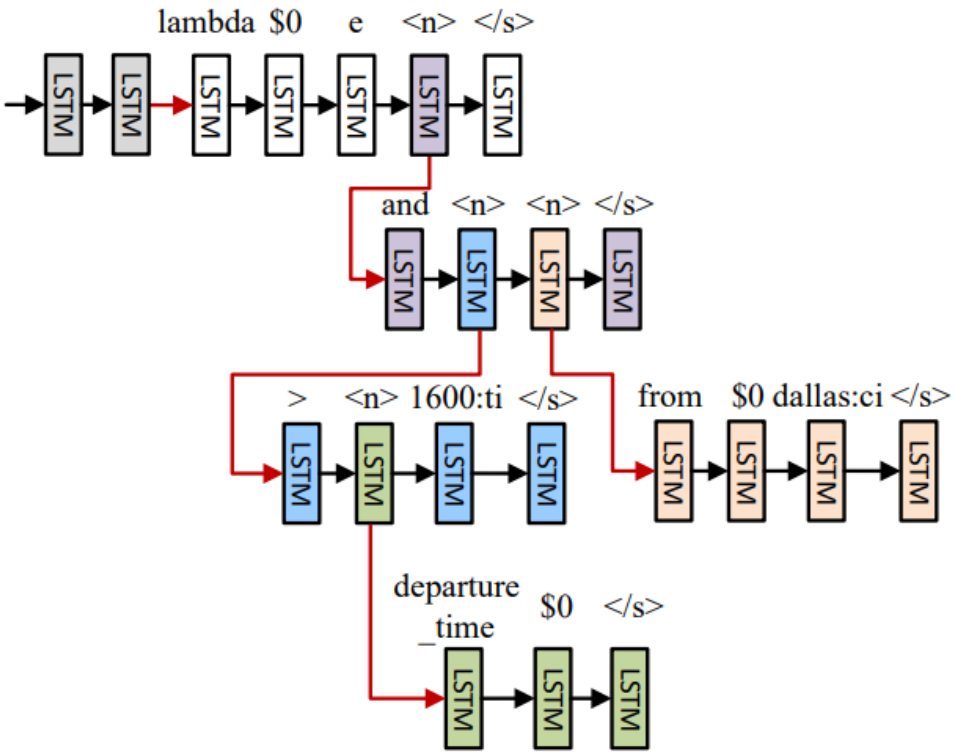
- 层次化的decoder，不生成扁平化的语义表示序列，而是生成层次结构化的语义表示（tree）
- 用<n>来表示树结构中的非终结符



Li Dong, Mirella Lapata. Language to Logical Form with Neural Attention. ACL-2016.

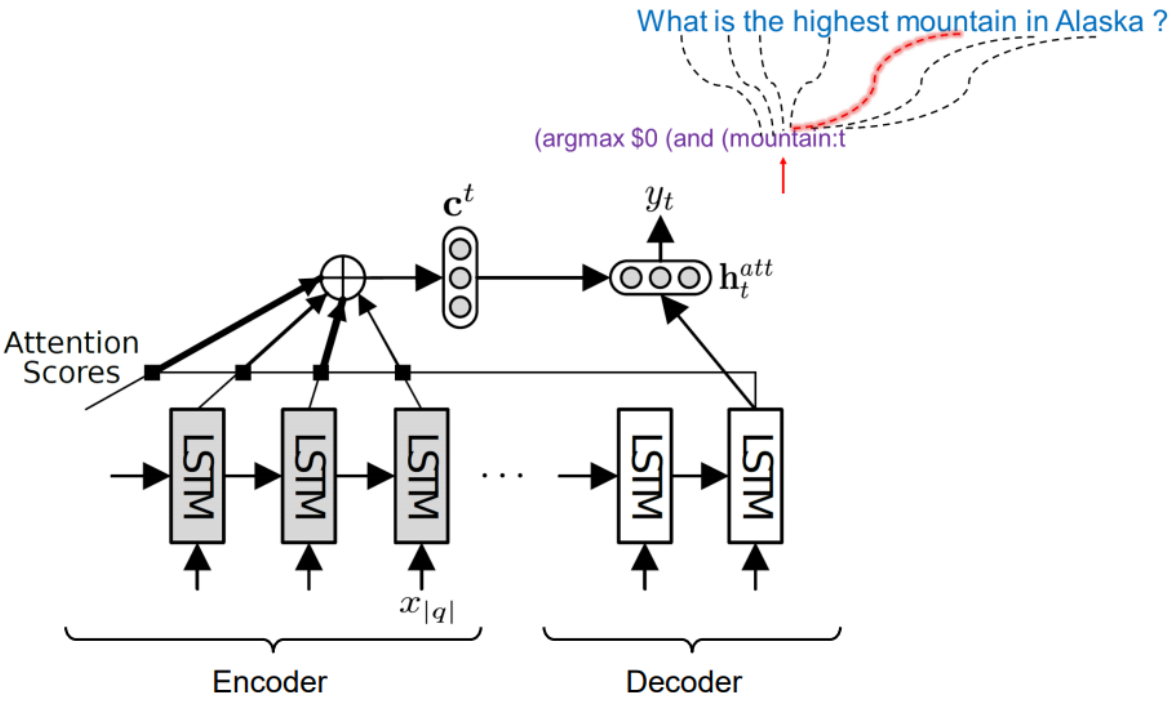
Seq2Tree

- 层次化的decoder，不生成扁平化的语义表示序列，而是生成层次结构化的语义表示（tree）
  - 占位符<n>都存放在一个队列里，直到队列为空才终止。



Seq2Seq/Seq2Tree中的注意力机制

- 注意力机制相当于让模型学习了词语到词语的语义表示之间的软对齐



Seq2Seq和Seq2Tree的性能

JOBS

Method	Accuracy
COCKTAIL (Tang and Mooney, 2001)	79.4
PRECISE (Popescu et al., 2003)	88.0
ZC05 (Zettlemoyer and Collins, 2005)	79.3
DCS+L (Liang et al., 2013)	90.7
TISP (Zhao and Huang, 2015)	85.0
SEQ2SEQ	87.1
– attention	77.9
– argument	70.7
SEQ2TREE	90.0
– attention	83.6

ATIS

Method	Accuracy
ZC07 (Zettlemoyer and Collins, 2007)	84.6
UBL (Kwiatkowski et al., 2010)	71.4
FUBL (Kwiatkowski et al., 2011)	82.8
GUSP-FULL (Poon, 2013)	74.8
GUSP++ (Poon, 2013)	83.5
TISP (Zhao and Huang, 2015)	84.2
SEQ2SEQ	84.2
– attention	75.7
– argument	72.3
SEQ2TREE	84.6
– attention	77.5

GEO

Method	Accuracy
SCISSOR (Ge and Mooney, 2005)	72.3
KRISP (Kate and Mooney, 2006)	71.7
WASP (Wong and Mooney, 2006)	74.8
λ-WASP (Wong and Mooney, 2007)	86.6
LNLZ08 (Lu et al., 2008)	81.8
ZC05 (Zettlemoyer and Collins, 2005)	79.3
ZC07 (Zettlemoyer and Collins, 2007)	86.1
UBL (Kwiatkowski et al., 2010)	87.9
FUBL (Kwiatkowski et al., 2011)	88.6
KCAZ13 (Kwiatkowski et al., 2013)	89.0
DCS+L (Liang et al., 2013)	87.9
TISP (Zhao and Huang, 2015)	88.9
SEQ2SEQ	84.6
– attention	72.9
– argument	68.6
SEQ2TREE	87.1
– attention	76.8

1. 层次结构的decoder使得模型学习到了目标语义表示的结构，这些结构信息能够帮助生成整个语义表示
2. Attention机制让模型学习了词语到词语的语义表示之间的软对齐，起到了类似于词典的效果

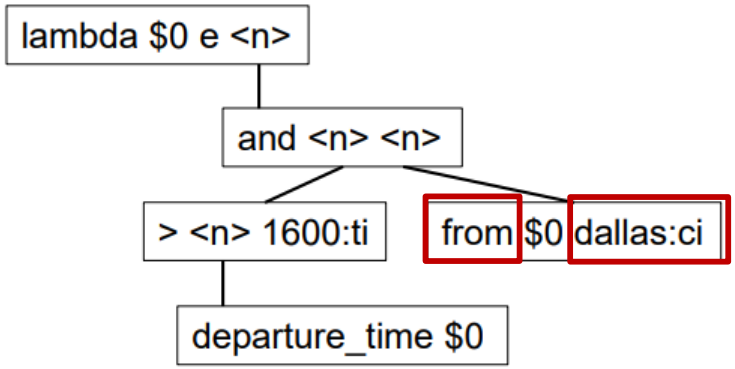
SEQ2ACT

Bo Chen, Le Sun, Xianpei Han.

Sequence-to-Action: End-to-End Semantic Graph Generation for Semantic Parsing. ACL-2018.

Seq2Act

- 之前的方法忽略生成的语义表示token之间的联系
  - 如from只能带两个参数
  - 如from的第二个参数只能是城市或者机场



- 用语义图表示语义
  - 与知识库建立紧密联系，充分利用知识库的知识约束

- 利用RNN模型的强表示能力和序列预测能力
  - 端到端

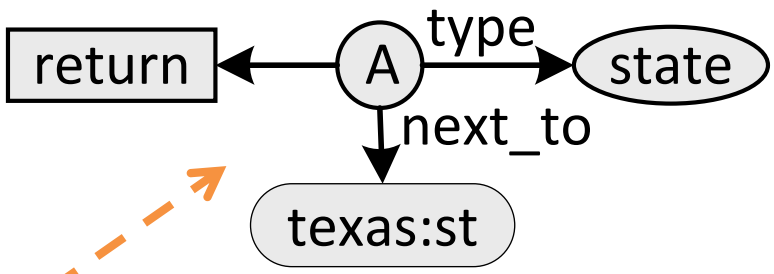
Bo Chen, Le Sun, Xianpei Han.
Sequence-to-Action: End-to-End Semantic Graph Generation for Semantic Parsing. ACL-2018.

Seq2Act: 端到端语义图生成

Which states border Texas?

sentence

Seq2Act: 端到端语义图生成

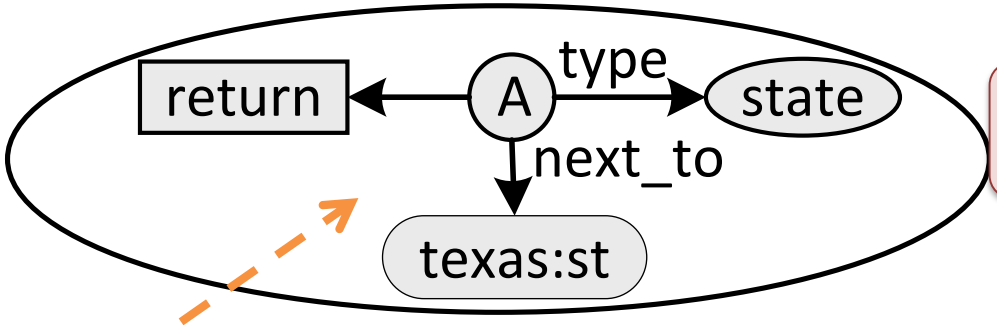


语义图

Which states border Texas?

句子

Seq2Act: 端到端语义图生成

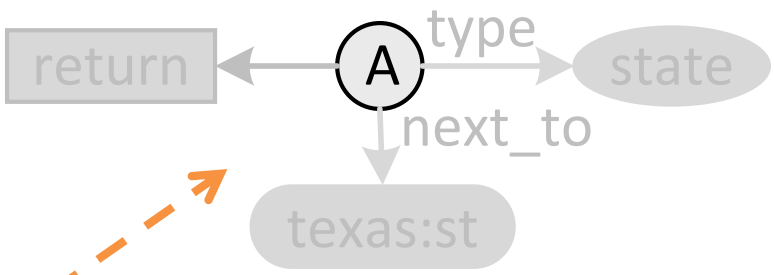


语义图

Which states border Texas?

句子

Seq2Act: 端到端语义图生成



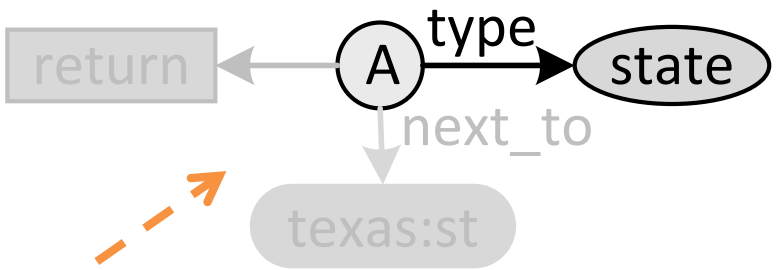
语义图

Action 1: add node A

Which states border Texas?

句子

Seq2Act: 端到端语义图生成



语义图

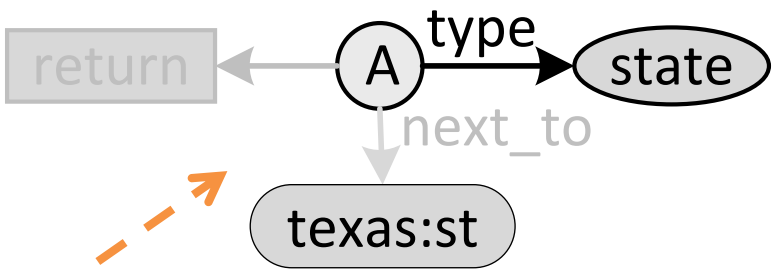
Action 1: add node A

Action 2: add type state

Which states border Texas?

句子

Seq2Act: 端到端语义图生成



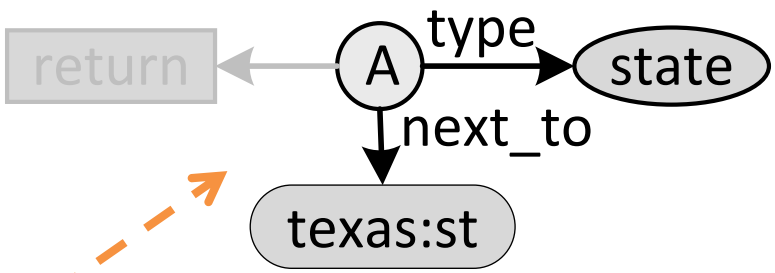
语义图

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st

Which states border Texas?

句子

Seq2Act: 端到端语义图生成



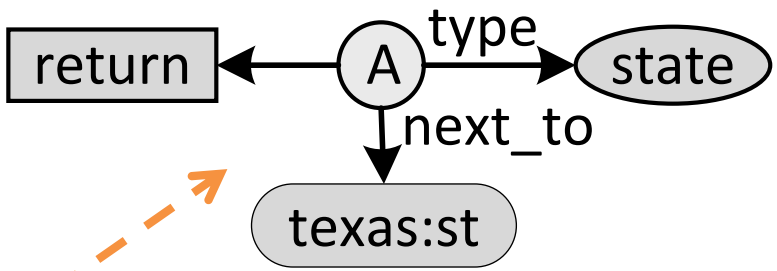
语义图

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to

Which states border Texas?

句子

Seq2Act: 端到端语义图生成



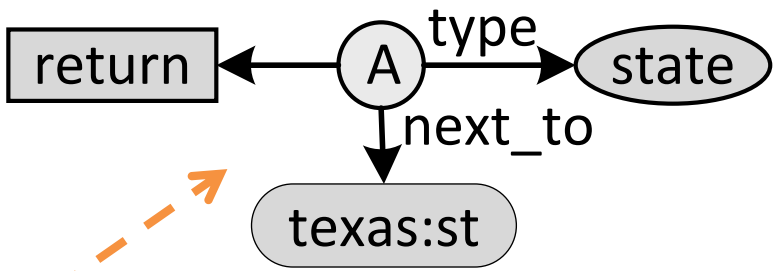
语义图

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to
Action 5: return

Which states border Texas?

句子

Seq2Act: 端到端语义图生成



语义图

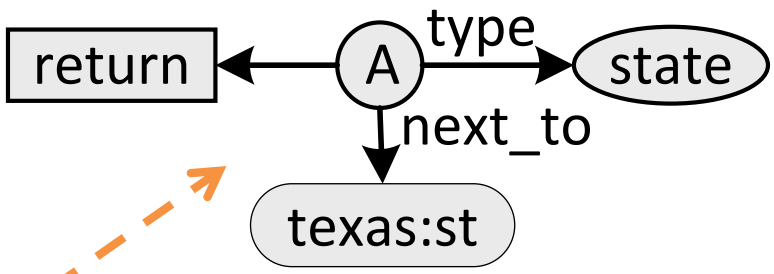
Which states border Texas?

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to
Action 5: return

句子

动作序列

Seq2Act: 端到端语义图生成



语义图

translate

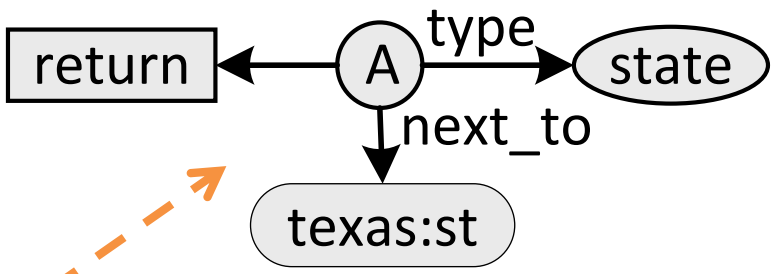
Which states border Texas?

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to
Action 5: return

句子

动作序列

Seq2Act: 端到端语义图生成



语义图

Which states border Texas?

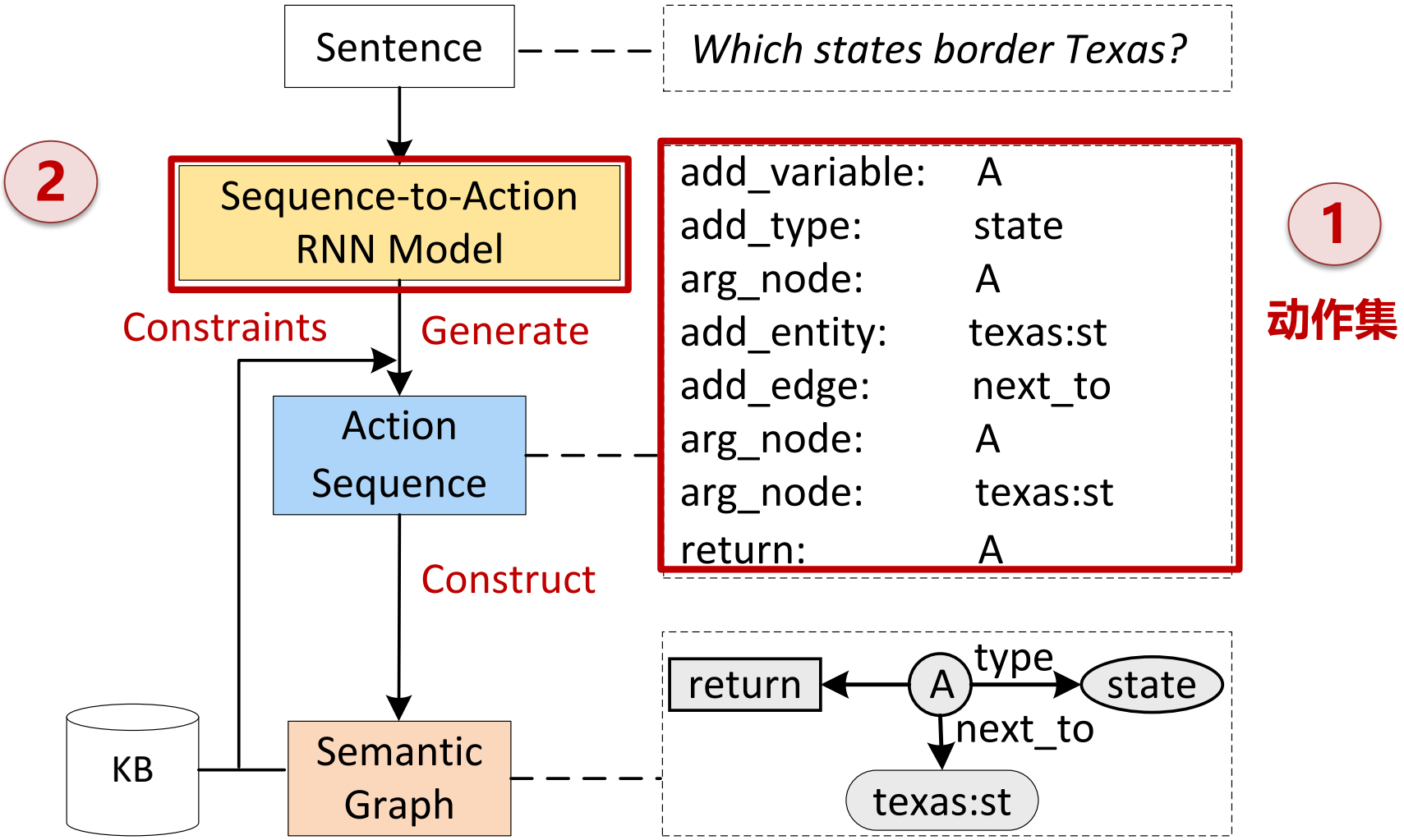
Sequence-to-Action

Action 1: add node A
Action 2: add type state
Action 3: add node texas:st
Action 4: add edge next_to
Action 5: return

句子

动作序列

Seq2Act: 模型框架

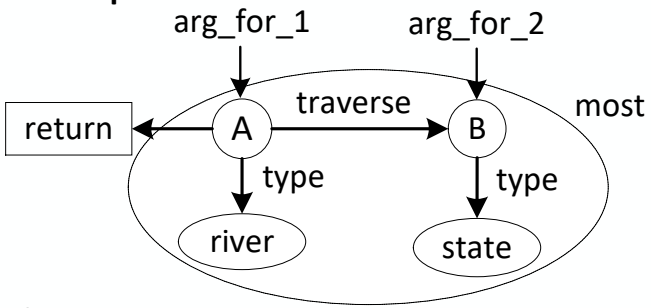


Seq2Act: 动作集

- Add variable node
  - E.g., A
- Add entity node
  - E.g., texas:st
- Add type node
  - E.g., state
- Add edge
  - E.g., next_to
- Operation action
  - E.g., argmax, argmin, count
- Argument action
  - For type node, edge and operation

Sentence: *Which river runs through the most states?*

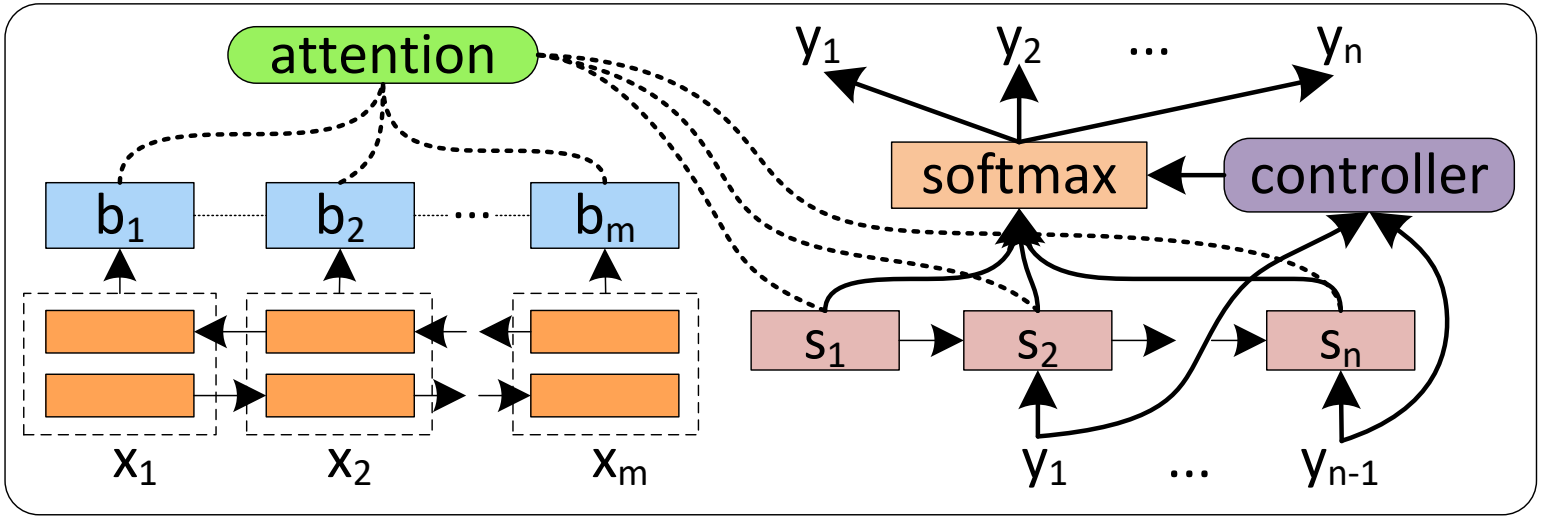
Semantic Graph:



Action Sequence:

Structure	Semantic	Arg
add_operation	most	
add_variable	A	
add_type	river	A
add_variable	B	
add_type	state	B
add_edge	traverse	A, B
end_operation	most	A, B
return	A	

Seq2Act: encoder-decoder model

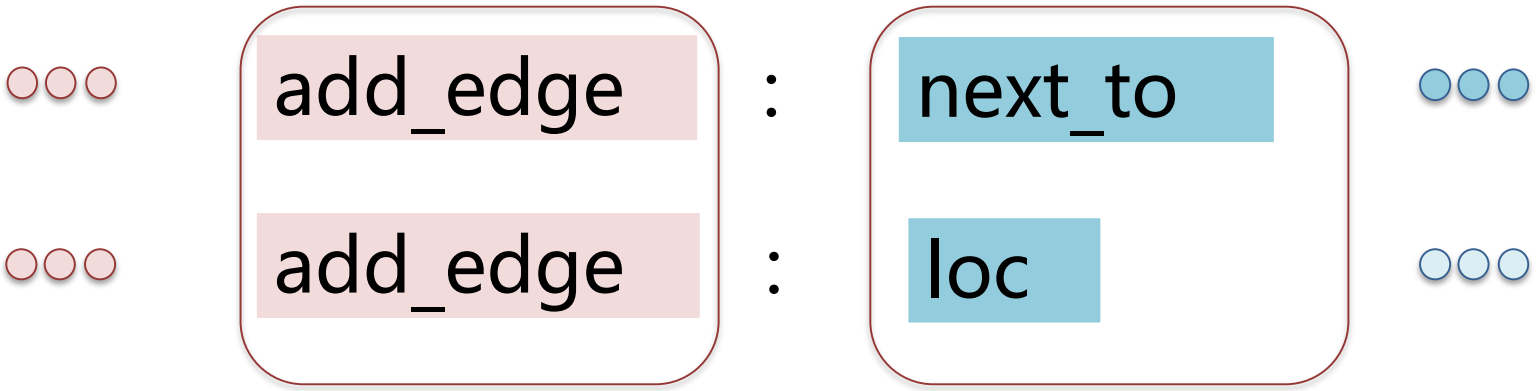


Typical encoder-decoder model (bi-LSTM with attention)

Action embedding

Seq2Act: action embedding

- 包含句法部分和结构部分
  - 分别进行编码，可以一定程度缓解动作的稀疏性（思想与因子化词汇类似）



Φ (add_edge:next to) = [Φ (add_edge); Φ (next_to)]

Seq2Act的性能

- 动作编码更加的紧凑，一定程度解决长距离依赖的问题
- 更容易在解码过程中加入知识库的约束信息，来保证解码时生成符合句法、符合语义的语义表示

	GEO	ATIS
Previous Work		
Zettlemoyer and Collins (2005)	79.3	–
Zettlemoyer and Collins (2007)	86.1	84.6
Kwiatkowksi et al. (2010)	88.9	–
Kwiatkowski et al. (2011)	88.6	82.8
Liang et al. (2011)* (+lexicon)	**91.1**	–
Poon (2013)	–	83.5
Zhao et al. (2015)	88.9	84.2
Rabinovich et al. (2017)	87.1	**85.9**
Seq2Seq Models		
Jia and Liang (2016)	85.0	76.3
Jia and Liang (2016)* (+data)	89.3	83.3
Dong and Lapata (2016): 2Seq	84.6	84.2
Dong and Lapata (2016): 2Tree	87.1	84.6
Our Models		
Seq2Act	87.5	84.6
Seq2Act (+C1)	88.2	85.0
Seq2Act (+C1+C2)	88.9	85.5

	Logical Form	Action Sequence
GEO	28.2	18.2
ATIS	28.4	25.8
OVERNIGHT	46.6	33.3

序列化逻辑表达式的长度与动作序列长度对比

方法小结

- Seq2Seq: 直接把目标语义表示序列化
- Seq2Tree: 考虑到目标语义表示的结构性
- Seq2Act: 用语义图表示语义，用动作序列编码语义图的构建

基于神经网络模型的扩展

- Constrained decoder：跟知识结合
- Coarse-to-fine：“通用结构”+“微调”

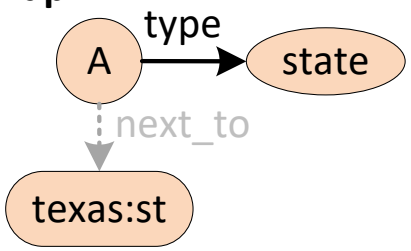
Constrained decoder

- 由于目标语言是形式化语言，需符合严格的条件约束，因此对比机器翻译里面的decoder，语义解析中的decoder可以使用严格的约束条件
- 句法条件和语义条件
  - 句法条件：如”(lambda $0 e (and (flight $0) (from $0 “下一个token应该是一个参数（实体或者变量）。
  - 语义条件：“(lambda $0 e (and (flight $0) (from $0 ”下一个token应该是一个城市或者机场
- 避免生成明显错误的token，从而提高最终生成准确语义表示的概率
- 在所有的基于神经网络的语义解析方法中都可以使用。

Constrained decoder in Seq2Act

Sentence: *Which states border Texas?*

Partial Semantic Graph:



	Structure	Semantic	Arg	Validity
Generated Actions	add_variable	A		
	add_type	state	A	
	add_entity	texas:st		
Candidate Next Action	add_type	city	texas:st	✗
	add_edge	loc	A, texas:st	✗
	add_edge	next_to	A, A	✗
	add_edge	next_to	A, texas:st	✓
	⋮	⋮	⋮	⋮

Action 1: violate type conflict

Action 2: violate selectional preference constraint

Action 3: structure constraint

Action 4: YES

Grammar constrained decoder

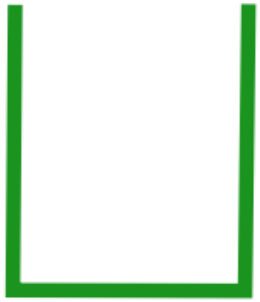
Which athlete was from South Korea after the year 2010?

Generated Actions

START → c
c→(<r,c> r)
<r,c>→(<<c,r>,<r,c>> <c,r>)
<<c,r>,<r,c>>→**reverse**
<c,r>→**athlete**
r→(<r,<r,r>> r r)
<r,<r,r>>→**and**
r→(<c,r> c)
<c,r>→**nation**
c→**south_korea**
r→(<c,r> c)
<c,r>→**year**
c→(<d,c> d)
<d,c>→(<<c,d>,<d,c>> <c,d>)
<<c,d>,<d,c>>→**reverse**
<c,d>→**date**
d→(**>=** d)
d→**2010.mm.dd**

Logical Form

```
((reverse athlete)
 (and (nation south_korea)
      (year ((reverse date)
             (>= 2010-mm-dd)))
```

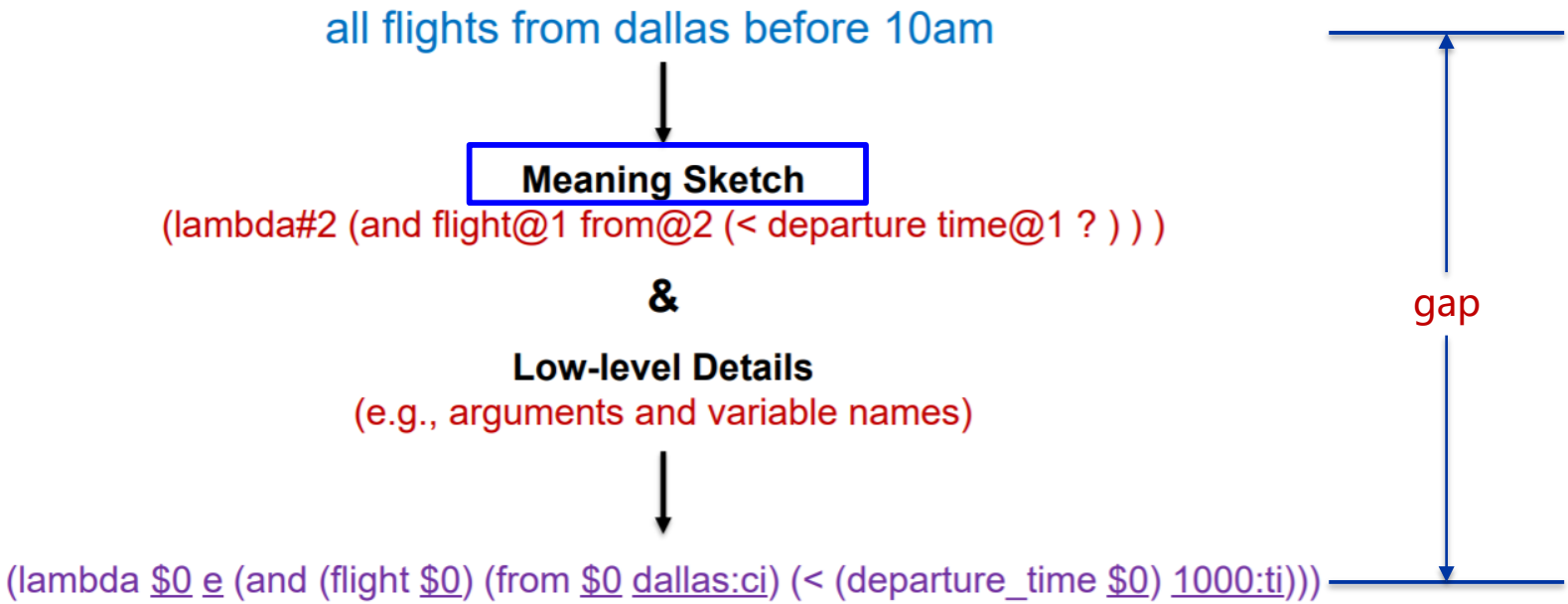


Non-terminal Stack

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner.
Neural semantic parsing with type constraints for semi-structured tables. EMNLP-2017.

Coarse-to-fine

- 自然语言与语义表示之间的gap太大
- 方案：先生成sketch，再进行grounding



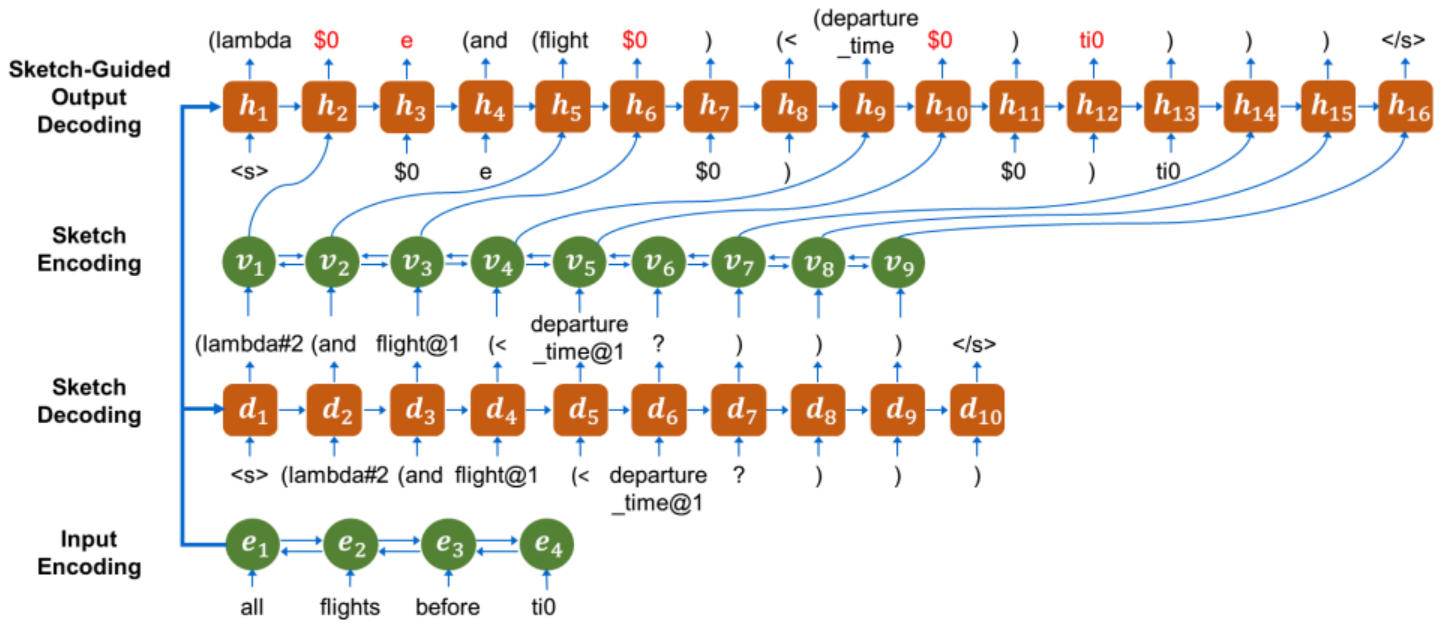
Li Dong and Mirella Lapata. Coarse-to-Fine Decoding for Neural Semantic Parsing. ACL-2018.

Coarse-to-fine

- 先生成sketch的好处?
  - 相比完整的语义表示，先生成sketch要容易很多
  - Sketch具有一定的泛化性，很多句子具有相同的sketch
  - 使用sketch之后，生成的序列长度大大缩减
  - Sketch可以为后面的decoding提供帮助

Coarse-to-fine

- 建模coarse-to-fine的过程



Coarse-to-fine

- Sketch如何指导后面的decoding?

  - Example 1: one augment is missing

    flight@1 ⟶ (flight ?)

  - Example 2: type information

    NUMBER ⟶ ? (a numeric token)

Coarse-to-fine的性能

- 在多个数据集上都取得了性能的提升

Method	GEO	ATIS
ZC07 (Zettlemoyer and Collins, 2007)	86.1	84.6
UBL (Kwiatkowksi et al., 2010)	87.9	71.4
FUBL (Kwiatkowski et al., 2011)	88.6	82.8
GUSP++ (Poon, 2013)	—	83.5
KCAZ13 (Kwiatkowski et al., 2013)	89.0	—
DCS+L (Liang et al., 2013)	87.9	—
TISP (Zhao and Huang, 2015)	88.9	84.2
SEQ2SEQ (Dong and Lapata, 2016)	84.6	84.2
SEQ2TREE (Dong and Lapata, 2016)	87.1	84.6
ASN (Rabinovich et al., 2017)	85.7	85.3
ASN+SUPATT (Rabinovich et al., 2017)	87.1	85.9
ONESTAGE	85.0	85.3
COARSE2FINE	88.2	87.7
− sketch encoder	87.1	86.9
+ oracle sketch	93.9	95.1

小结：基于神经网络的语义解析方法

- **主流：**基于encoder-decoder的基本框架生成语义表示
- **核心：**选择语义表示，定义生成语义表示的action、grammar或者推导规则。
- **代表性方法：**Seq2Seq、Seq2Tree、Seq2Action等
- **扩展：**受约束的decoding，coarse-to-fine

- 优点：
  - 模型相对简单、端到端

- 缺点：
  - 可解释性差

大纲

- 语义解析任务介绍
- 基于词典-文法的语义解析
- 基于语义图构建的语义解析
- 基于神经网络的语义解析
- 总结和展望

现有方法总结

方法	核心	优势	劣势
基于词典-组合文法的方法	词典 组合文法	基于组合语义思想，解析过程清晰可见，可解释性强	需要学习词典和定义组合文法，且容易受限于词典的覆盖度
基于语义图的方法	语义图表示 语义图构建	语义图与自然语言句子具有类似的结构；与知识库联系紧密，可充分利用知识库的知识来指导语义图的构建	往往依赖于特定的手段来构建语义图，缺乏一定的通用性
基于神经网络的方法	Decoding中的token形式 Decoding中的constraints	端到端，充分利用神经网络模型的表示能力和拟合能力	可解释性差

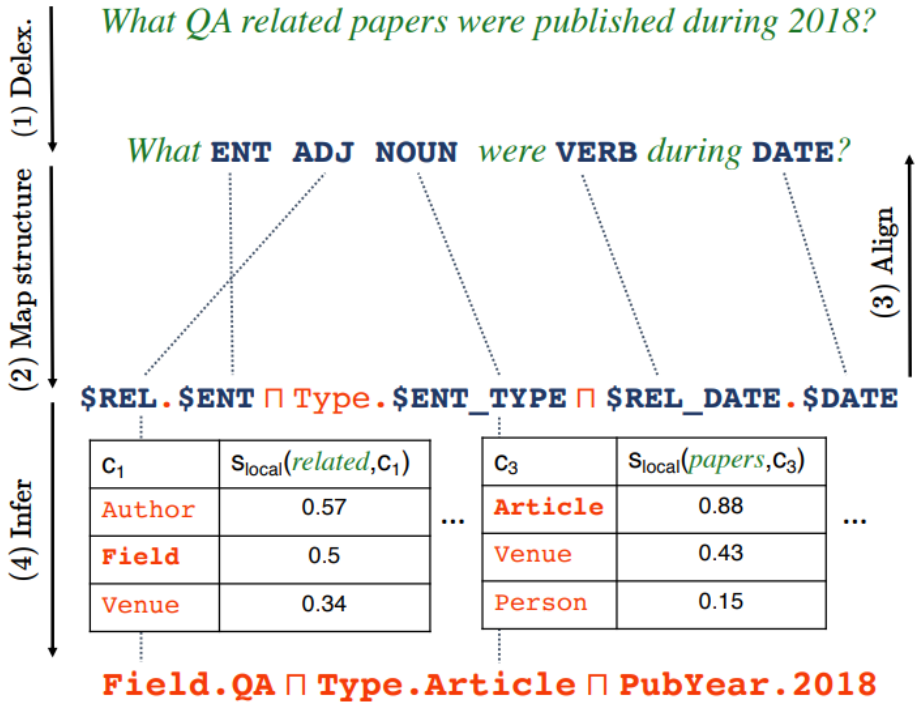
展望（1）：低成本的模型构建

- **问题：语义解析当前的方法需要大量标注语料，不利于语义解析技术的推广和应用，如何进行低成本的模型构建将成为研究热点。**

- **思路：**
  - Transfer
  - 弱监督/无监督
  - 预训练

展望（1）：低成本的模型构建

- **Transfer: 领域迁移**
  - **解耦合结构和词典**
    - ✓**假设：结构在不同领域下具有通用性**



Jonathan Herzig and Jonathan Berant. Decoupling structure and lexicon for zero-shot semantic parsing. EMNLP-2018.

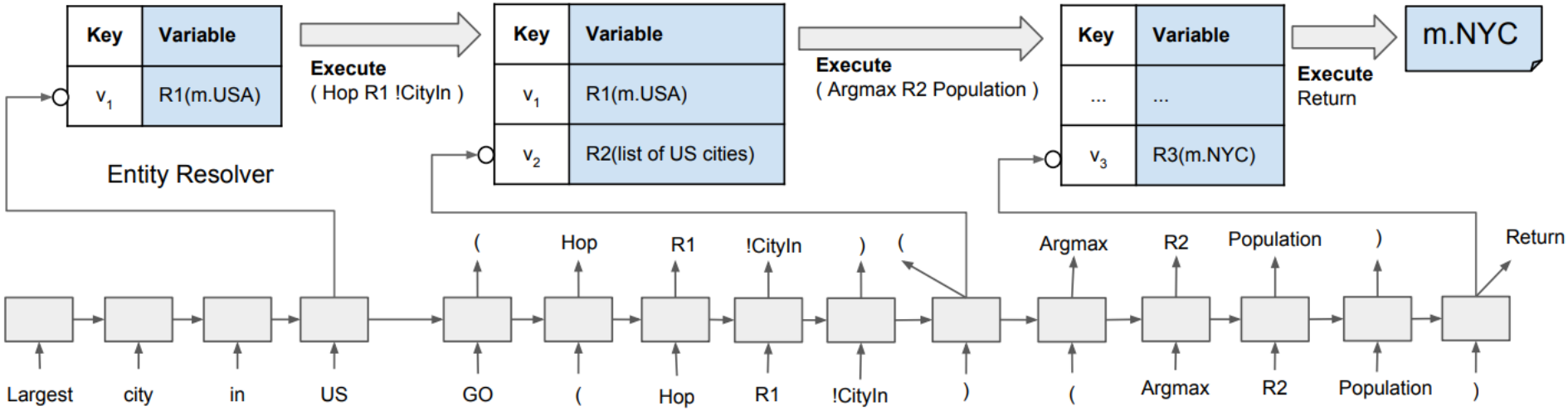
展望（2）：符号与神经网络融合的语义解析

- **现象：两类方法各有优势，基于符号的方法长于推理和利用知识，神经网络的方法长于表示和计算，如何在语义解析中有效融合两类方法也将成为研究者探索的点。**

- **挑战：符号在神经网络中的表示形式，以及计算方式（往往不可导）**

展望（2）：符号与神经网络融合的语义解析

- **当前方法：Neural Symbolic Machines (NSM)**
  - 以memory的形式在neural network中引入symbolic representation。
  - 利用强化学习算法来训练模型（symbolic部分不可导）



Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, Ni Lao.
Neural symbolic machines: learning semantic parsers on Freebase with weak supervision. ACL-2017.

展望（3）：上下文感知的语义解析

- **问题：当前句子解析需要利用之前句子的信息**
  - 场景：对话理解、代码生成

Example #1:
(a) show me the flights from boston to philly
$\lambda x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
(b) show me the ones that leave in the morning
$\lambda x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
$\wedge during(x, morning)$
(c) what kind of plane is used on these flights
$\lambda y.\exists x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
$\wedge during(x, morning) \wedge aircraft(x) = y$

Example #2:
(a) show me flights from milwaukee to orlando
$\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl)$
(b) cheapest
$argmin(\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl),$
$\lambda y.fare(y))$
(c) departing wednesday after 5 o'clock
$argmin(\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl)$
$\wedge day(x, wed) \wedge depart(x) > 1700,$
$\lambda y.fare(y))$

展望（3）：上下文感知的语义解析

- **关键挑战：指代、省略、如何复用信息**

Example #1:
(a) show me the flights from boston to philly
$\lambda x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
(b) show me the ones that leave in the morning
$\lambda x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
$\wedge during(x, morning)$
(c) what kind of plane is used on these flights
$\lambda y.\exists x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
$\wedge during(x, morning) \wedge aircraft(x) = y$

指代

Example #2:
(a) show me flights from milwaukee to orlando
$\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl)$
(b) cheapest
$argmin(\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl),$
$\lambda y.fare(y))$
(c) departing wednesday after 5 o'clock
$argmin(\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl)$
$\wedge day(x, wed) \wedge depart(x) > 1700,$
$\lambda y.fare(y))$

省略

展望（3）：上下文感知的语义解析

- 当前解决方法：
  - 基于CCG的方法：增加新词汇和文法来处理指代和省略等现象
  - 基于神经网络的方法：turn-level encoder + query segment copying

展望（3）：上下文感知的语义解析

- 基于CCG的方法：针对指代

新词汇　　ones　:=　$N : \lambda x.!\langle e,t\rangle(x)$

$\lambda x.!\langle e,t\rangle(x) \wedge during(x, morning)$

Example #1:

(a) show me the flights from boston to philly

$\lambda x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$

替换 $!\langle e,t\rangle$

(b) show me the ones that leave in the morning

$\lambda x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
$\wedge during(x, morning)$

(c) what kind of plane is used on these flights

$\lambda y.\exists x.flight(x) \wedge from(x, bos) \wedge to(x, phi)$
$\wedge during(x, morning) \wedge aircraft(x) = y$

Luke Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. ACL-2009.

展望（3）：上下文感知的语义解析

- 基于CCG的方法：针对省略

Type-shifting

$$A/B : f \Rightarrow A : f(\lambda x.!\langle e,t\rangle(x))$$
$$A\backslash B : f \Rightarrow A : f(\lambda x.!\langle e,t\rangle(x))$$

Example #2:
(a) show me flights from milwaukee to orlando
$\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl)$
(b) cheapest
$argmin(\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl), \lambda y.fare(y))$
(c) departing wednesday after 5 o'clock
$argmin(\lambda x.flight(x) \wedge from(x, mil) \wedge to(x, orl) \wedge day(x, wed) \wedge depart(x) > 1700\,, \lambda y.fare(y))$

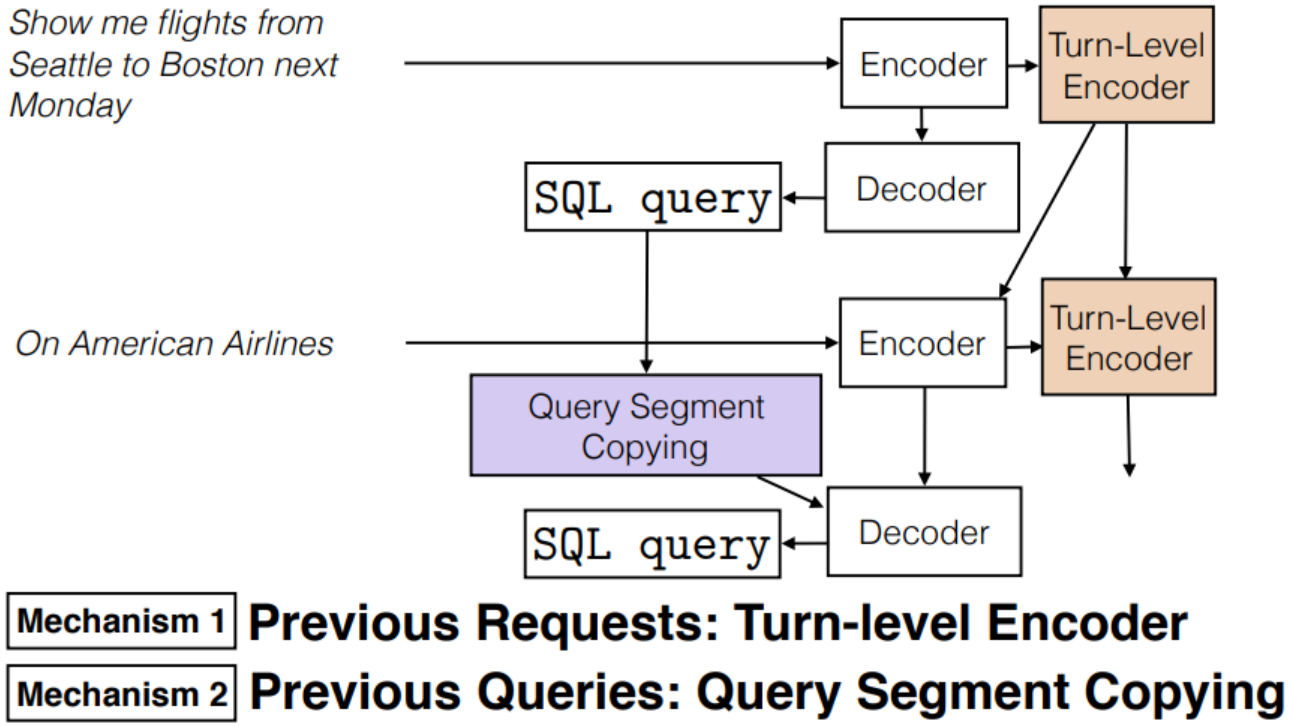
省略了ones

$$\frac{\text{cheapest}}{\dfrac{NP/N}{\lambda g.argmin(\lambda x.g(x), \lambda y.fare(y))}\;\;\dfrac{}{NP}}$$
$$argmin(\lambda x.!\langle e,t\rangle(x), \lambda y.fare(y))$$

Luke Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. ACL-2009.

展望（3）：上下文感知的语义解析

- Neural方法
  - –Turn-level encoder：编码上文信息
  - –Query Segment Copying：用copy机制来复用信息



Alane Suhr, Srinivasan Iyer, Yoav Artzi.

Learning to Map Context-Dependent Sentences to Executable Formal Queries. NAACL-2018.

展望（4）：与物理世界交互的语义解析

- **问题：解析执行一系列指令，状态随着指令的执行而发生变化**
  - 场景：机器人执行指令

Empty out the leftmost beaker of purple chemical

Then, add the contents of the first beaker to the second

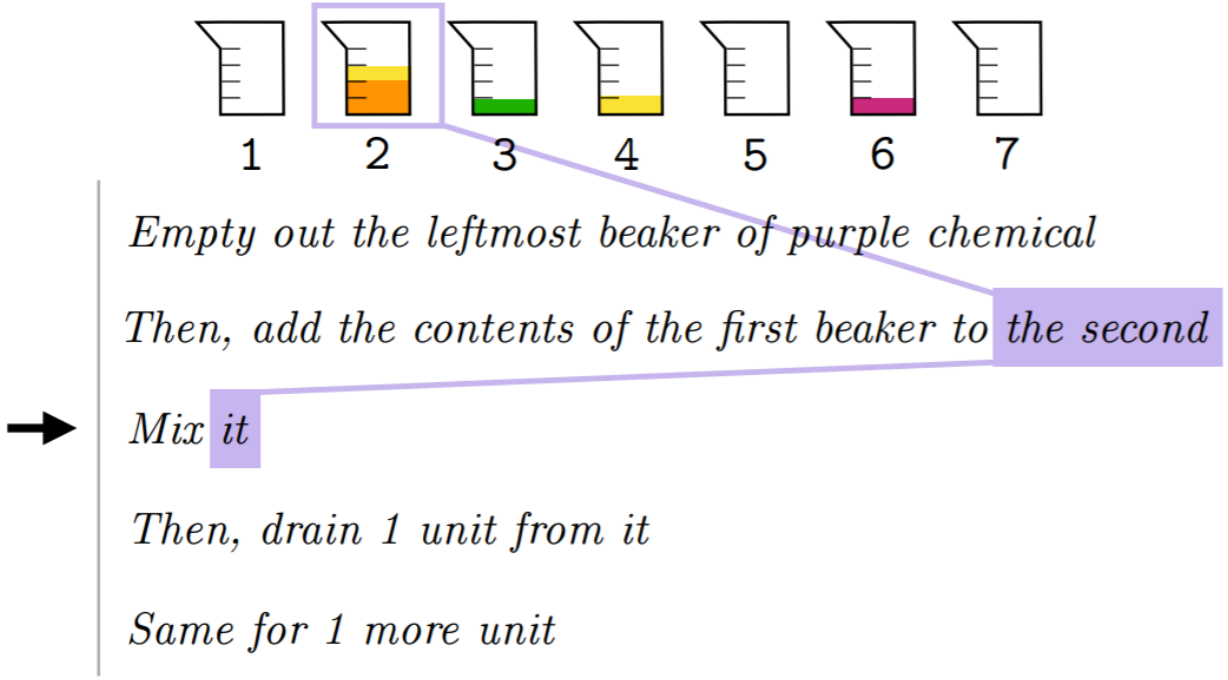
Mix it

Then, drain 1 unit from it

Same for 1 more unit

展望（4）：与物理世界交互的语义解析

- 关键挑战：
  - 如何建模之前的指令（同样有指代和省略的现象）
  - 如何建模当前环境的状态变化



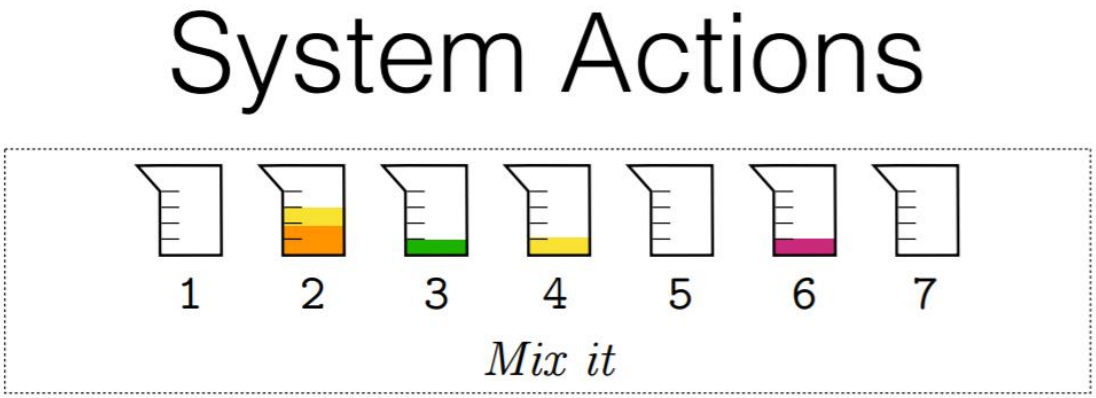
展望（4）：与物理世界交互的语义解析

- 当前解决方法：
  - –基于注意力机制的encoder-decoder模型
    - ✓定义System Action
    - ✓Encoder中建模之前命令以及当前的状态

Alane Suhr and Yoav Artzi.
Situated Mapping of Sequential Instructions to Actions with Single-step Reward Observation. ACL-2018.

展望（4）：与物理世界交互的语义解析

- 根据特定的任务设定system action



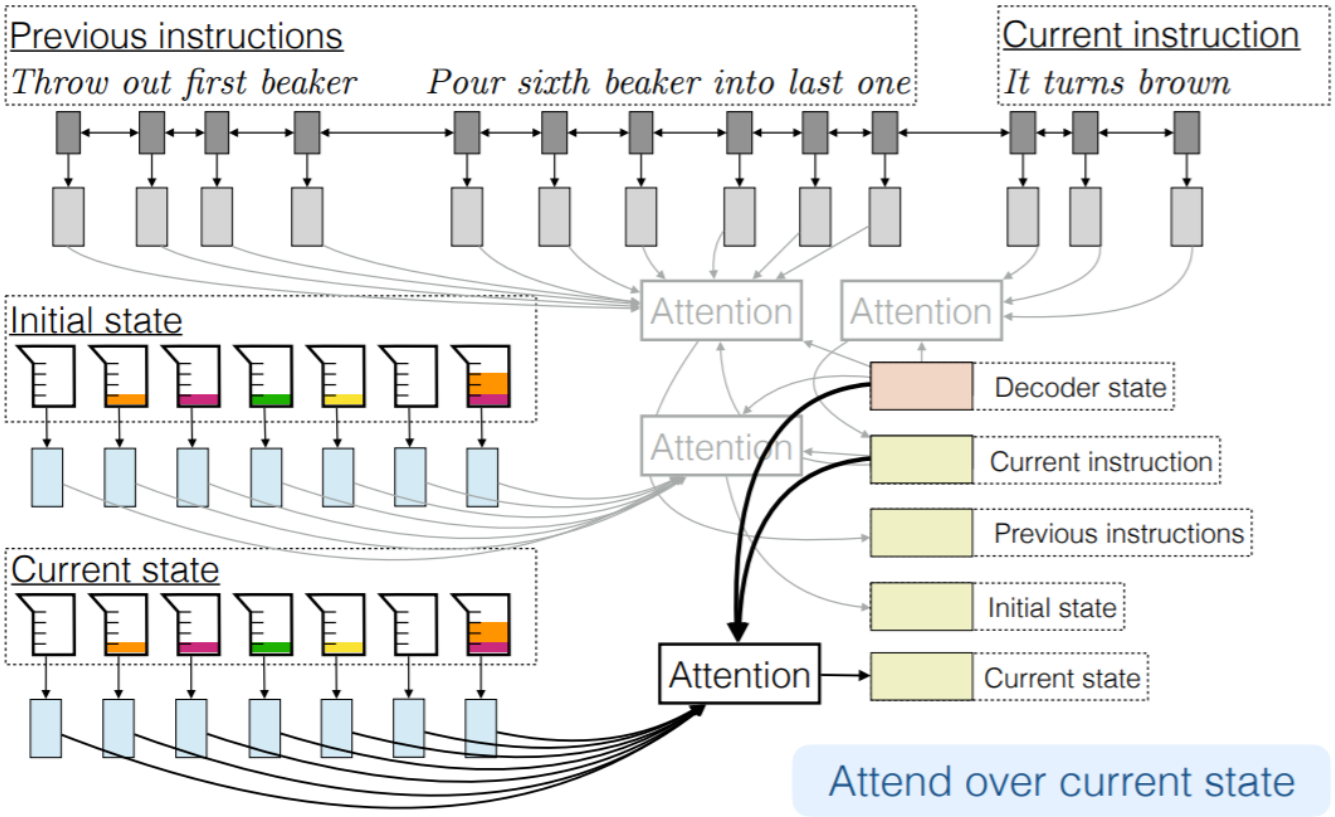
- Each beaker is a stack
- Actions are **pop** and **push**

```
pop  2;
pop  2;
pop  2;
push 2 brown;
push 2 brown;
push 2 brown;
```

Alane Suhr and Yoav Artzi.
Situated Mapping of Sequential Instructions to Actions with Single-step Reward Observation. ACL-2018.

展望（4）：与物理世界交互的语义解析

- Encoder中同时建模当前命令，之前命令，初始状态，当前状态



Alane Suhr and Yoav Artzi.

Situated Mapping of Sequential Instructions to Actions with Single-step Reward Observation. ACL-2018.

总结：未来展望

- 低成本的模型构建
  - Transfer
  - 弱监督/无监督
  - 预训练
- 符号与神经网络融合的语义解析
  - Memory的形式引入符号
  - 强化学习来训练模型
- 上下文感知的语义解析
  - 融合上文信息的encoder
  - 基于copy机制的信息复用方法
- 与物理世界交互的语义解析
  - 融合多信息的encoder（当前命令、之前的命令，当前的状态等）

语义解析资源

- Github: https://github.com/cipnlu/Semantic-Parsing
- 实验室网址： http://www.icip.org.cn/zh/zh_resource/
  - 包含语义解析相关的Tutorial（ACL10，ACL13，ACL18等）

Tutorials about Semantic Parsing

(ordered by year)

1. **开放域语义解析**
   [slides]
   Xianpei Han, Bo Chen. CIPS Summer School, 2019.

2. **Neural Semantic Parsing**
   [slides]
   Pradeep Dasigi, Srini Iyer, Alane Suhr, Matt Gardner, Luke Zettlemoyer. ACL-2018.

3. **Question Answering with Knowledge Base, Web and Beyond**
   [slides]
   Scott Wen-tau Yih. NAACL-2016.

4. **Natural Language Understanding: Foundations and State-of-the-Art**
   [slides]
   Percy Liang. ICML-2015.

5. **Semantic Parsing with Combinatory Categorial Grammars**
   [slides] [website]
   Yoav Artzi, Nicholas FitzGerald and Luke Zettlemoyer. ACL-2013, EMNLP-2014, AAAI-2015.

6. **Semantic Parsing: The Task, the State of the Art and the Future**
   [slides]
   Rohit J. Kate and Yuk Wah Wong. ACL-2010.

语义解析资源

- Github: https://github.com/cipnlu/Semantic-Parsing
- 实验室网址： http://www.icip.org.cn/zh/zh_resource/
  - 还有博士论文（Luke Zettlemoyer, Percy Liang, Tom Kwiatkowski, Jayant Krishnamurthy, Yoav Artzi, Li Dong等）

PhD theses about Semantic Parsing

(ordered by year)

1. **The Lifecycle of Neural Semantic Parsing**
   [thesis]
   Jianpeng Cheng, advisor: Mirella Lapata and Adam Lopez, University of Edinburgh, 2019.

2. **Learning Natural Language Interfaces with Neural Models**
   [thesis]
   Li Dong, advisor: Mirella Lapata, University of Edinburgh, 2018.

3. **基于词典学习和结构映射的语义解析技术研究 (Semantic Parsing based on Lexicon Learning and Structure Mapping)**
   [thesis]
   Bo Chen, advisor: Le Sun, Institute of Software, Chinese Academy of Sciences, 2018.

4. **Integrating Machine Learning and Symbolic Reasoning: Learning to Generate Symbolic Representations from Weak Supervision**
   [thesis]
   Chen Liang, advisor: Ken Forbus and Ni Lao, Northwestern University, 2018.

5. **Syntax-Mediated Semantic Parsing**
   [thesis]
   Siva Reddy, advisor: Mirella Lapata and Mark Steedman, University of Edinburgh, 2017.

语义解析资源

- Github: https://github.com/cipnlu/Semantic-Parsing
- 实验室网址： http://www.icip.org.cn/zh/zh_resource/
  - 工具包（SEMPRE, Cornell SPF, AllenNLP等）

Frameworks or Tools for Semantic Parsing

1. SEMPRE: Semantic Parsing with Execution
   [website]
2. Cornell SPF - Cornell Semantic Parsing Framework
   [website]
3. Allennlp - An open-source NLP research library
   [website]
4. SLING - A natural language frame semantics parser
   [website]
5. Graph-Parser
   [website]
6. OpenNMT (for neural sequence modeling)
   [website]

语义解析资源

- Github: https://github.com/cipnlu/Semantic-Parsing
- 实验室网址： http://www.icip.org.cn/zh/zh_resource/
  - 数据集（GEO，ATIS，Webquestions, Spider等）

Datasets for Semantic Parsing

1. **Geoquery**
   [website] [execution]
   cite: John Zelle and Raymond Mooney. 1996. Learning to parse database queries using inductive logic programming. In AAAI.

2. **Jobs**
   [website] [execution]
   cite: Lappoon Tang and Raymond Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In ECML.

3. **ATIS**
   [context-independent] [context-dependent]
   cite: Charles Hemphill, John Godfrey, and George Doddington. 1990. The ATIS spoken language pilot corpus. In DARPA Speech & Natural Language Workshop.
   Deborah Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In HLT.

4. **Webquestions**
   [website]
   cite: Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In EMNLP.

语义解析资源

- Github: https://github.com/cipnlu/Semantic-Parsing
- 实验室网址： http://www.icip.org.cn/zh/zh_resource/
  – 论文（ACL、EMNLP、NAACL、TACL、CL等）

Papers about Semantic Parsing

(ordered by year)

2019

1. **Learning an Executable Neural Semantic Parser**
   [paper]
   Jianpeng Cheng, Siva Reddy, Vijay Saraswat, Mirella Lapata. CL-2019.

2. **Iterative Search for Weakly Supervised Semantic Parsing**
   [paper]
   Pradeep Dasigi, Matt Gardner, Shikhar Murty, Luke Zettlemoyer, Eduard Hovy. NAACL-2019.

3. **Context-Dependent Semantic Parsing over Temporally Structured Data**
   [paper]
   Charles Chen and Razvan Bunescu. NAACL-2019.

2018

1. **Semantic Parsing with Syntax- and Table-Aware SQL Generation**
   [paper]
   Yibo Sun, Duyu Tang, Nan Duan, Jianshu Ji, Guihong Cao, Xiaocheng Feng, Bing Qin, Ting Liu, Ming Zhou. ACL-2018.

2. **Coarse-to-Fine Decoding for Neural Semantic Parsing**
   [paper] [slides] [code]
   Li Dong and Mirella Lapata. ACL-2018.

敬请大家批评指正！

韩先培、陈波

实验室链接：http://www.icip.org.cn