CS678 Advanced Natural Language Processing

Structured Prediction 2: Semantic Parsing

Antonis Anastasopoulos & Ziyu Yao

https://nlp.cs.gmu.edu/course/cs678-fall22

Acknowledgement: Many slides are taken from Greg Durrett CS388@UT Austin and Antonios Anastasopoulos CS695@GMU

Outline

- Introduction to Semantic Parsing
  - Model Theoretic Semantics
  - Example: CCG Parsing
- (Shallow Parsing) Predicate-Argument Semantics (Eisenstein Ch13)
- Applications of Semantic Parsing

Why Semantic Parsing?

- Syntactic parsing: understand the structural organization of a sentence
- Semantic parsing: understand the underlying meaning of a sentence
  - Answering questions: where is the nearest coffeeshop?
  - Instructing a robot: go to the corner and get the ladder
  - Fact checking, searching the web for contradictory evidence
  - In machine translation, do the source sentence and the translated sentence mean the same?

Eisenstein Ch12

What does Semantic Parsing do?

- Converting natural language into **formal meaning representation**.
- What should the meaning representation be like for being useful?
  - Unambiguity: exactly one meaning per statement;
  - Grounding: providing a way to link language to external knowledge/observations/actions;
  - Computational inference: meanings can be combined to derive additional knowledge;
  - Expressivity: should cover the full range of things that people talk about in natural language

Eisenstein Ch12

Model Theoretic Semantics

- “Model”: a formal construct that can ground a natural language expression to particular states in the world
- Natural language statement S => interpretation of S that models it
  - e.g., *She likes going to that restaurant*
  - Interpretation: defines who *she* and that *restaurant* are, make it able to be concretely evaluated with respect to a world
  - Entailment (statement A implies statement B) reduces to: in all worlds where A is true, B is true
- Our modeling language is *first-order logic*

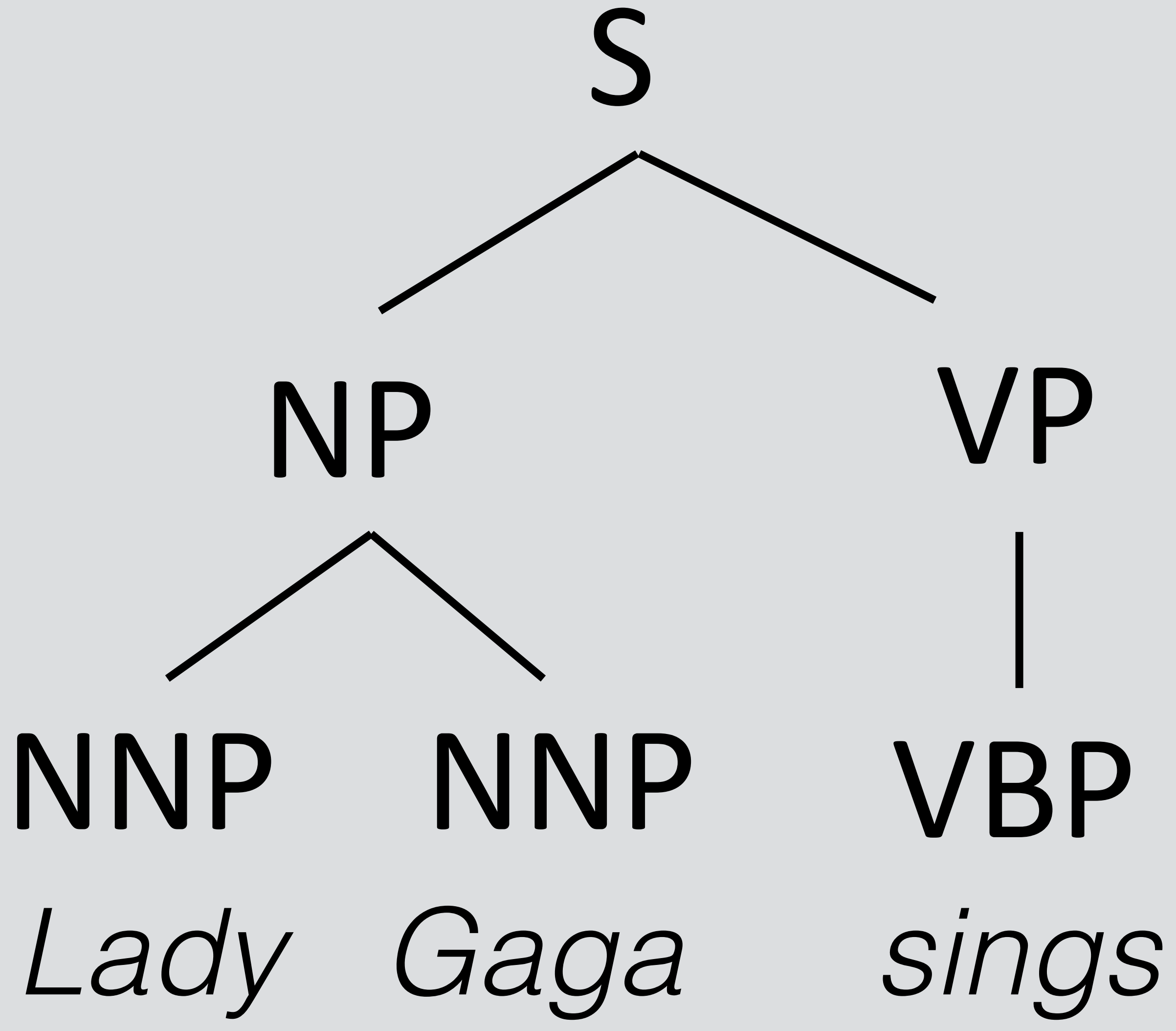
First-order Logic

- Powerful logic formalism including things like entities, relations, and quantifications
  - e.g., *Lady Gaga sings*
  - *sings* is a *predicate* (with one argument), function *f: entity* → *true/false*
  - *sings(Lady Gaga)* = true or false, have to execute this against some database *(world)*
- Quantification: “forall” operator, “there exists” operator

$$\forall x \text{ sings}(x) \lor \text{dances}(x) \Rightarrow \text{performs}(x)$$

“Everyone who sings or dances performs”

Montague Semantics



Id	Name	Alias	Birthdate	Sings?
e470	Stefani Germanotta	Lady Gaga	3/28/1986	T
e728	Marshall Mathers	Eminem	10/17/1972	T

Database containing entities, predicates, etc.

- Sentence expresses something about the world which is either true or false
- *Denotation*: evaluation of some expression against this database

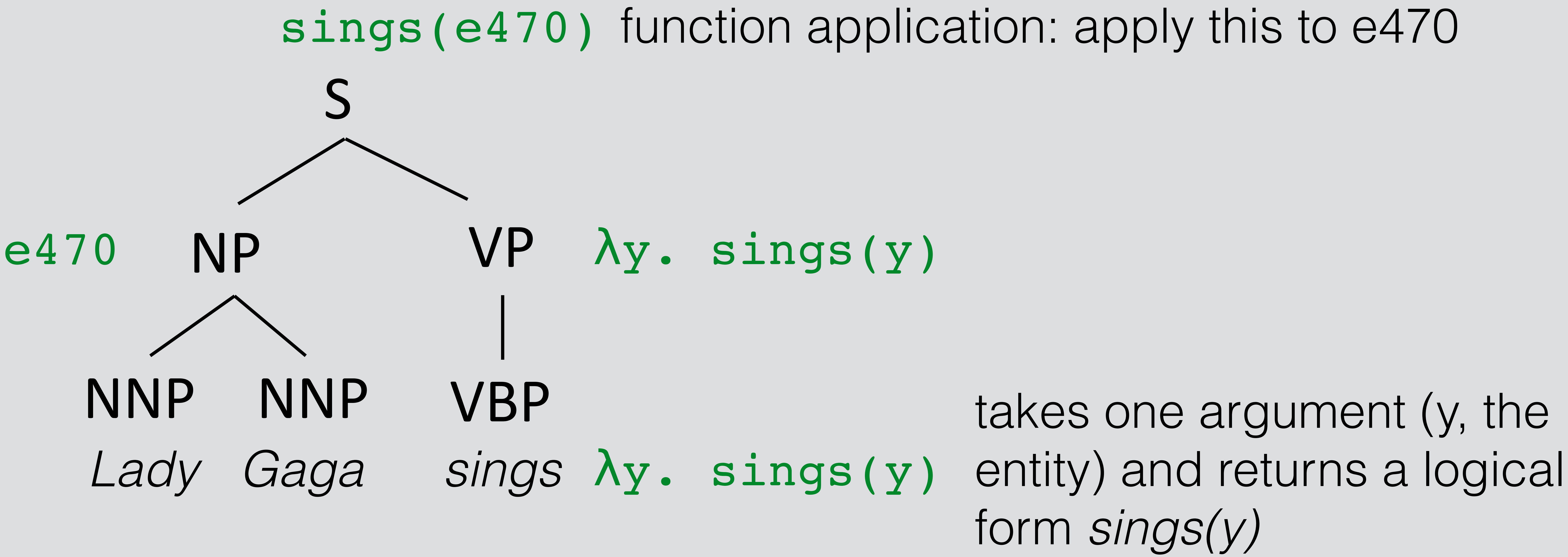
`[[`*Lady Gaga*`]] = e470`

denotation of this string is an entity

`[[sings(e470)]] = True`

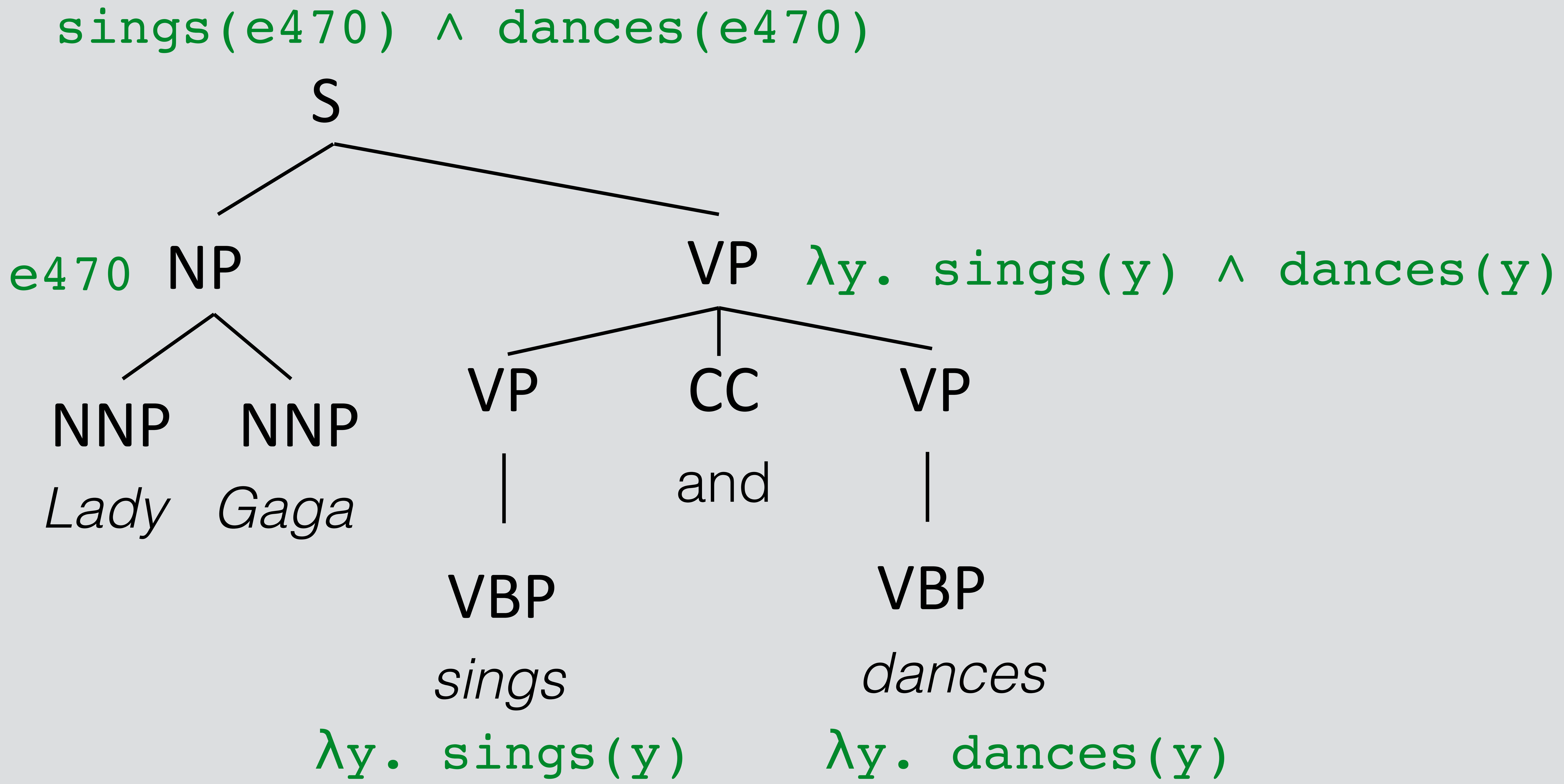
denotation of this expression is T/F

Montague Semantics



- We can use the syntactic parse as a bridge to the **lambda-calculus representation**, build up a **logical form** *compositionally*
- *Lambda notation*: a way to abstract from fully specified FOL formulas

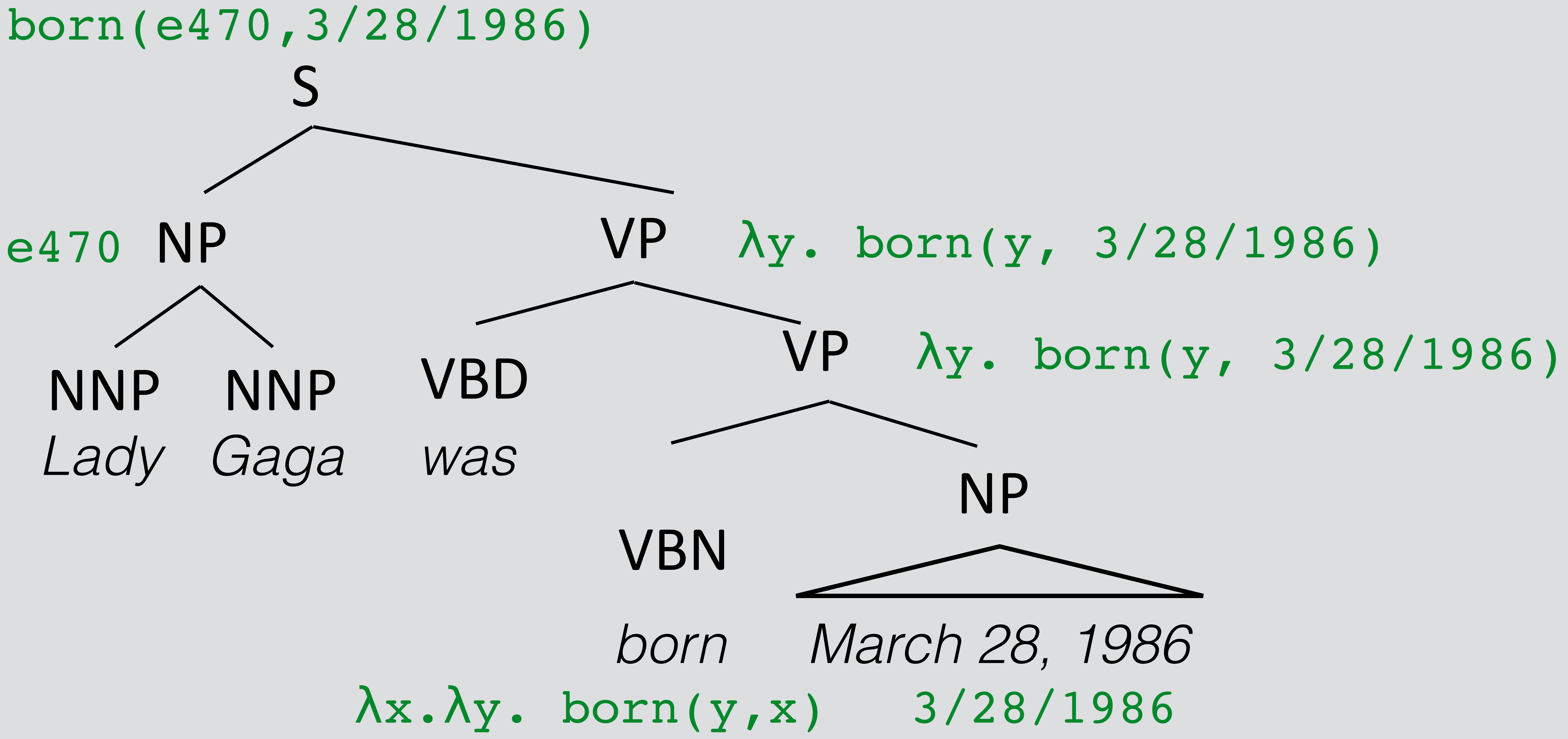
Parses to Logical Forms



- General rules: VP: λy. a(y) ∧ b(y) -> VP: λy. a(y) CC VP: λy. b(y)

  S: f(x) -> NP: x VP: f

Parses to Logical Forms



- Function takes two arguments: first x (date), then y (entity)
- How to handle *tense*: should we indicate that this happened in the past?

Tricky things

- Adverbs/temporality: *Lady Gaga sang well yesterday*

```
sings(Lady Gaga, time=yesterday, manner=well)
```

  - “Neo-Davidsonian” view of events: things with many properties:

```
∃e. type(e,sing) ∧ agent(e,e470) ∧ manner(e,well) ∧ time(e,…)
```

- Quantification: *Everyone is friends with someone*

`∃y ∀x friend(x,y)`	`∀x ∃y friend(x,y)`
(one friend)	(different friends)

  - Same syntactic parse for both! So syntax doesn't resolve all ambiguities

- Indefinite: *Amy ate a waffle* `∃w. waffle(w) ∧ ate(Amy,w)`

- Generic: *Cats eat mice* (all cats eat mice? most cats? some cats?)

Semantic Parsing

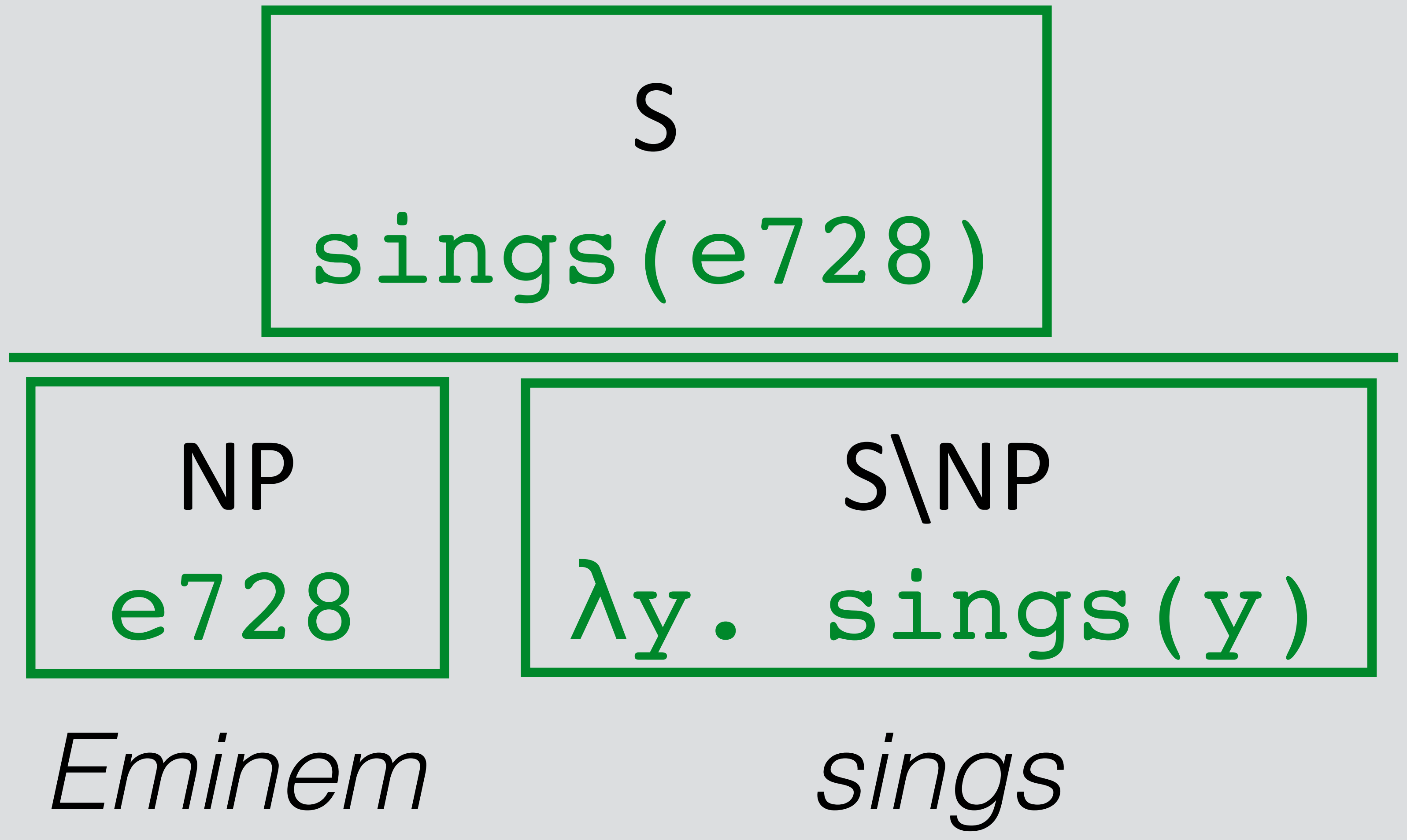
- How is semantic parsing useful?
- e.g., for question answering
  - Syntactic parsing doesn't tell you everything you want to know, but indicates the right structure
  - Semantic parsing gives the meaning representation of the question, which can then be grounded to certain knowledge base/database for answers
  - *More examples (Robot Instructing, Automatic Programming, etc.) later this class!*
- Next: CCG parsing, which produces lambda-calculus expressions that can be executed in contexts

Outline

- Introduction to Semantic Parsing
  - Model Theoretic Semantics
  - Example: CCG Parsing
- (Shallow Parsing) Predicate-Argument Semantics (Eisenstein Ch13)
- Applications of Semantic Parsing

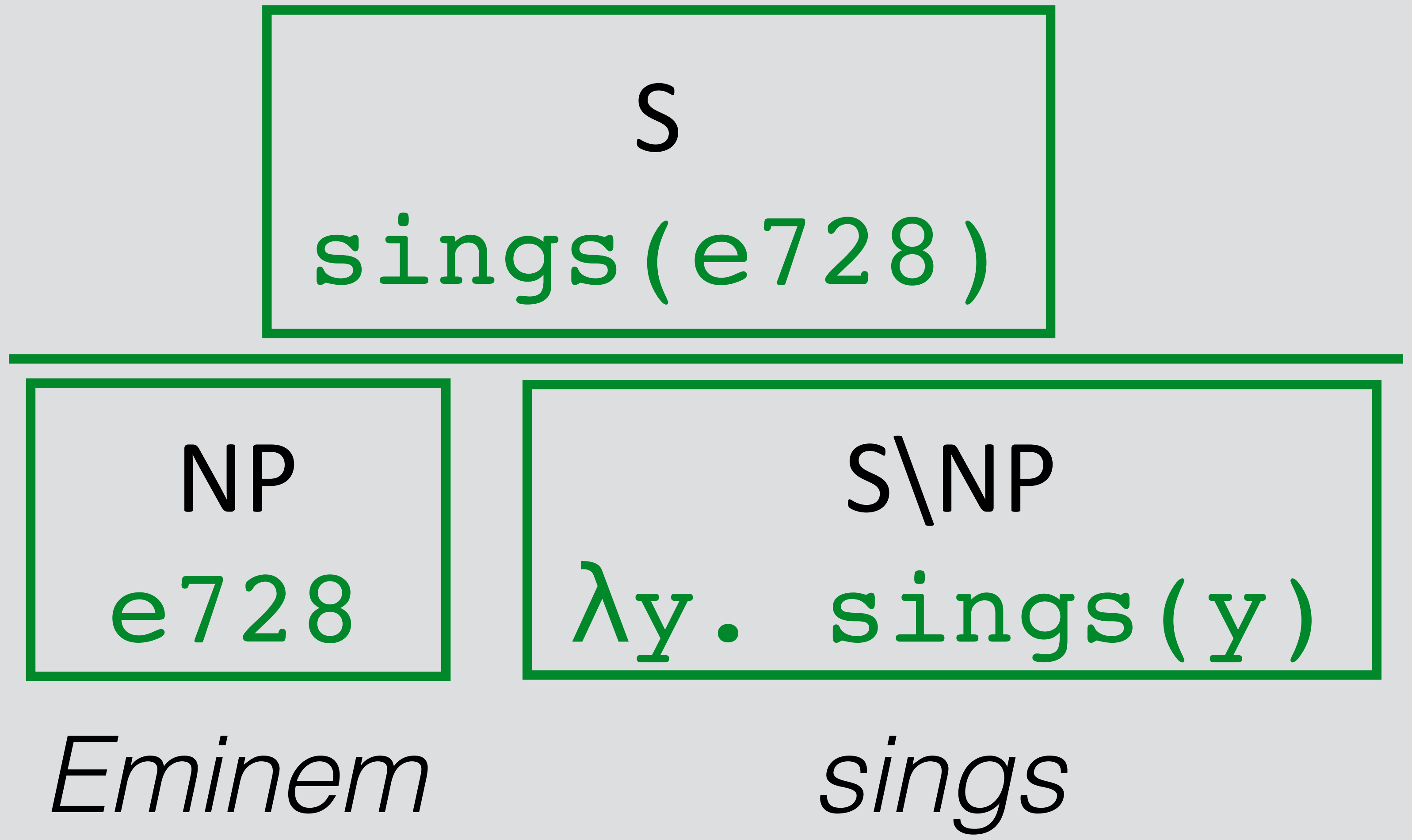
Combinatory Categorial Grammar

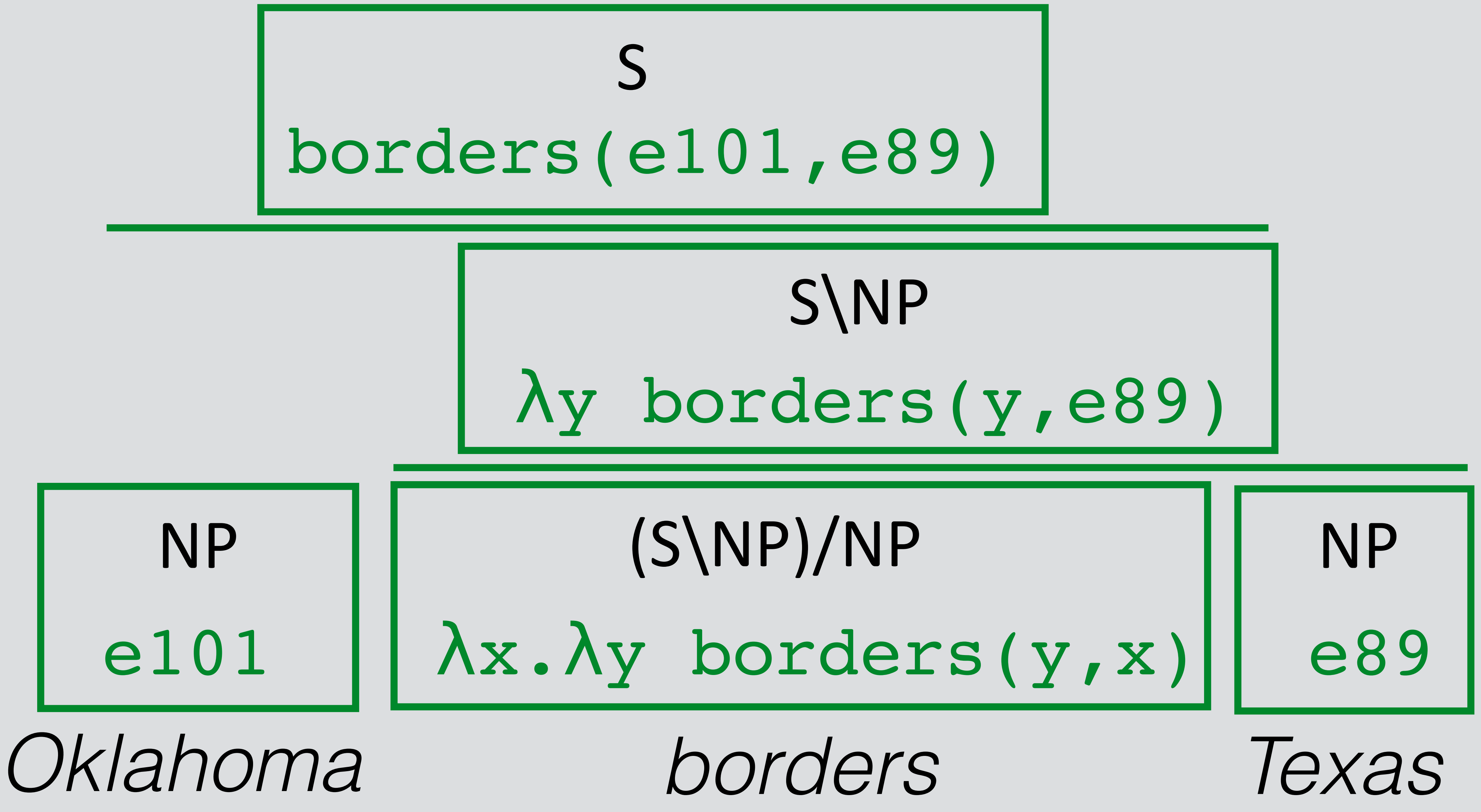
- Steedman+Szabolcsi (1980s): formalism bridging syntax and semantics
- Parallel derivations of syntactic parse and lambda calculus expression
- Syntactic categories (for this lecture): S, NP, "slash" categories
- S\NP: "if I combine with an NP on my left side, I form a sentence" — verb
- When you apply this, there has to be a parallel instance of function application on the semantics side



Combinatory Categorial Grammar

- Steedman+Szabolcsi (1980s): formalism bridging syntax and semantics
- Syntactic categories (for this lecture): S, NP, "slash" categories
  - S\NP: "if I combine with an NP on my left side, I form a sentence" — verb
  - (S\NP)/NP: "I need an NP on my right and then on my left" — verb with a direct object





CCG Parsing

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

$$\xrightarrow{\hspace{6em}>} \quad \begin{array}{c} (S\backslash NP) \\ \lambda y.borders(y, texas) \end{array}$$

- “What” is a ***very*** complex type: needs a noun and needs a S\NP to form a sentence. S\NP is basically a verb phrase (*border Texas*)

Zettlemoyer and Collins (2005)

CCG Parsing

$$
\begin{array}{cccc}
\text{What} & \text{states} & \text{border} & \text{Texas} \\
\hline
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP \\
\lambda f.\lambda g.\lambda x.f(x) \wedge g(x) & \lambda x.state(x) & \lambda x.\lambda y.borders(y,x) & texas
\end{array}
$$

$$
\begin{array}{cc}
\multicolumn{1}{c}{\xrightarrow{\hspace{6em}} >} & \multicolumn{1}{c}{\xrightarrow{\hspace{6em}} >} \\
S/(S\backslash NP) & (S\backslash NP) \\
\lambda g.\lambda x.state(x) \wedge g(x) & \lambda y.borders(y, texas)
\end{array}
$$

$$
\xrightarrow{\hspace{12em}} >
$$

$$
\begin{array}{c}
S \\
\lambda x.state(x) \wedge borders(x, texas)
\end{array}
$$

- "What" is a ***very*** complex type: needs a noun and needs a S\NP to form a sentence. S\NP is basically a verb phrase (*border Texas*)
- Lexicon is highly ambiguous — all the challenge of CCG parsing is in picking the right lexicon entries

Zettlemoyer and Collins (2005)

CCG Parsing

Show me	flights	to	Prague
S/N	N	(N\N)/NP	NP
$\lambda f.f$	$\lambda x.flight(x)$	$\lambda y.\lambda f.\lambda x.f(x) \wedge to(x,y)$	PRG

N\N

$\lambda f.\lambda x.f(x) \wedge to(x,PRG)$

N

$\lambda x.flight(x) \wedge to(x,PRG)$

S

$\lambda x.flight(x) \wedge to(x,PRG)$

- “to” needs an NP (destination) and N (parent)

Slide credit: Dan Klein

How to Learn a Semantic Parser?

- Different approaches in different settings

- For example, if you have a set of *annotated sentences with CCG tags*, then you can learn a “supertagger”, and then run the parser

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

  - Recall: how to learn a sequence tagging model? CRF

How to Learn a Semantic Parser?

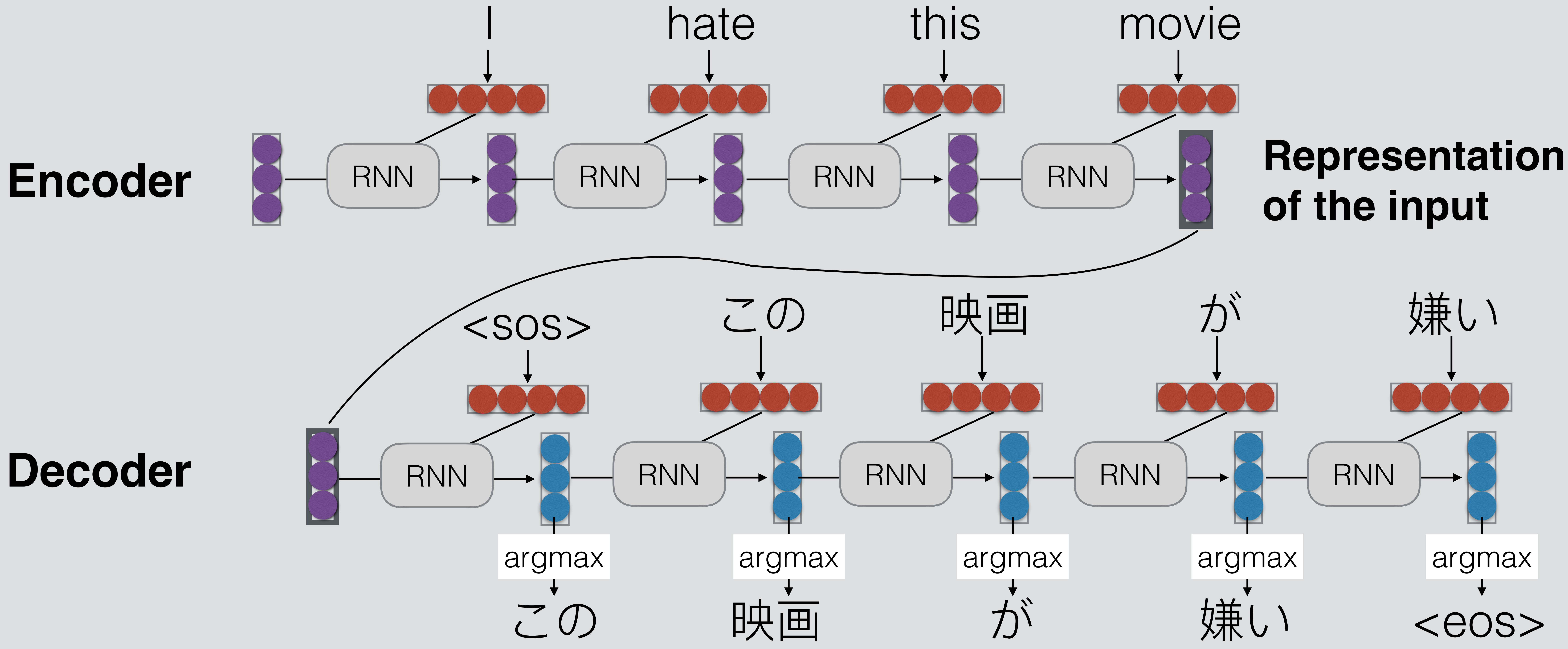
- Different approaches in different settings
- Or, if you know the *derivations* of a set of CCG parse trees, then learn a CCG parser is the same as learning a constituency parser

$$
\begin{array}{cccc}
\text{What} & \text{states} & \text{border} & \text{Texas} \\
\hline
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP \\
\lambda f.\lambda g.\lambda x.f(x) \wedge g(x) & \lambda x.state(x) & \lambda x.\lambda y.borders(y,x) & texas \\
\hline
\multicolumn{2}{c}{S/(S\backslash NP) \quad >} & \multicolumn{2}{c}{(S\backslash NP) \quad >} \\
\multicolumn{2}{c}{\lambda g.\lambda x.state(x) \wedge g(x)} & \multicolumn{2}{c}{\lambda y.borders(y, texas)} \\
\hline
\multicolumn{4}{c}{S \quad >} \\
\multicolumn{4}{c}{\lambda x.state(x) \wedge borders(x, texas)}
\end{array}
$$

How to Learn a Semantic Parser?

- Different approaches in different settings
- More challenging (yet practical) settings:
  - Learning from <NL, logical form> pairs
    - e.g., *What states border Texas —> λx. state(x) ∧ borders(x, e89)*
    - Goal: learn a parser that can generate a lambda expression for any given test sentence
  - Learning from <NL, denotation> pairs
    - Denotation: the evaluation of the target logical form over the world
    - e.g., *What states border Texas —> {Oklahoma, Arkansas, Louisiana, New Mexico}*

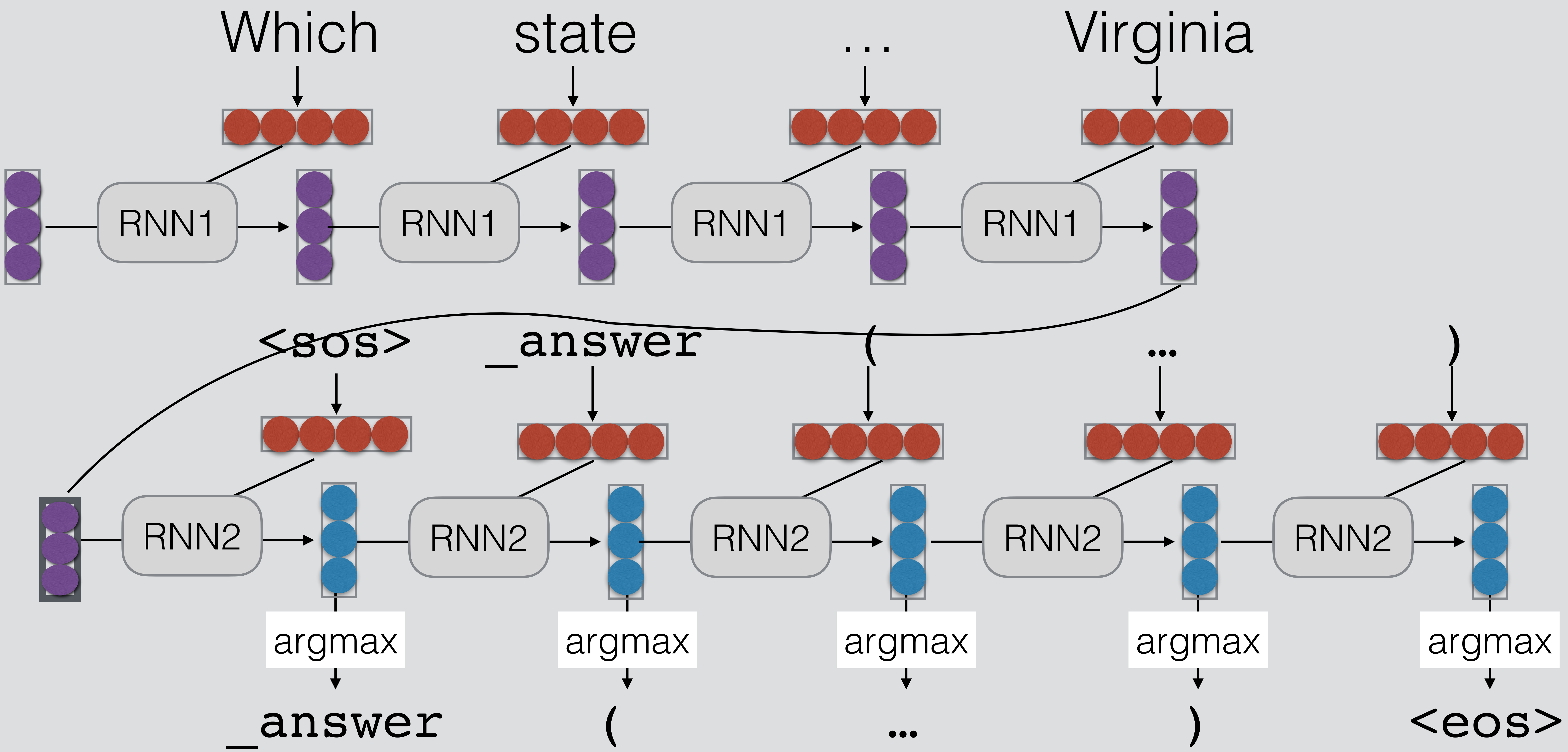
Sequence-to-Sequence Neural Model



Seq2Seq for Semantic Parsing

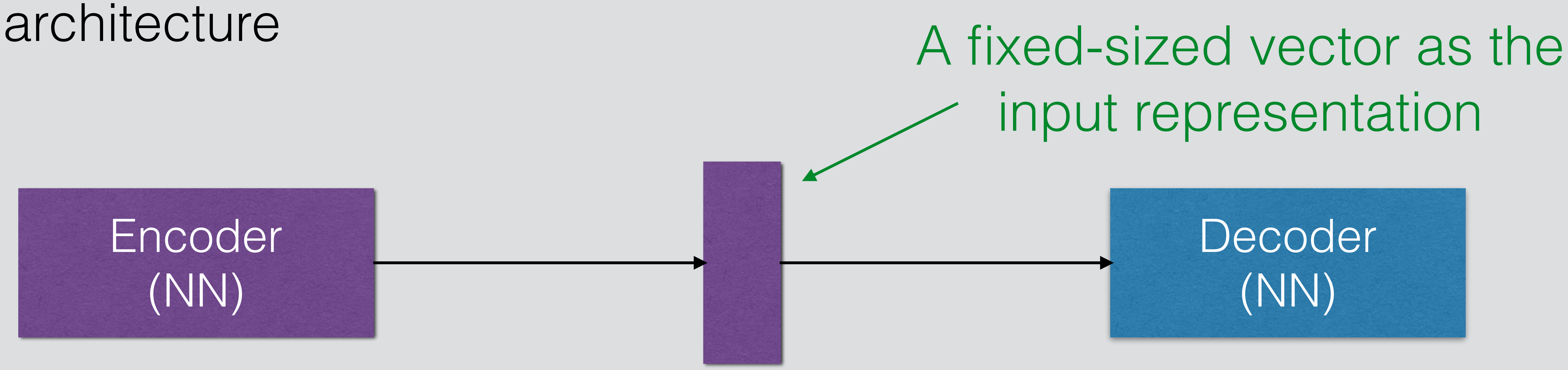
Generating logical form from NL, e.g.,

Which state borders Virginia? —> `_answer ( A , ( _state ( A ) , _next_to ( A , B ) , _const ( B , _stateid ( va ) ) ) )`



Encoder-Decoder Models

- A general model architecture



- Depending on the instantiation of the data types:
  - Sequential Encoder + Sequential Decoder = Seq2Seq
  - Sequential Encoder + Tree-structured Decoder = Seq2Tree
  - Tree-structured Encoder + Sequential Decoder = Tree2Seq
  - Tree-structured Encoder + Tree-structured Decoder = Tree2Tree
- The choice of model units: LSTM, Transformer, etc.

Outline

- Introduction to Semantic Parsing
  - Model Theoretic Semantics
  - Example: CCG Parsing
- (Shallow Parsing) Predicate-Argument Semantics **(Eisenstein Ch13)**
- Applications of Semantic Parsing

Predicate-Argument Semantics

- A “lightweight” semantic representations
  - Discards some aspects of first-order logic
  - But focus on predicate-argument structures

Predicate-Argument Semantics

- A “lightweight” semantic representations
  - Discards some aspects of first-order logic
  - But focus on predicate-argument structures
- Recall: the event semantics
  - e.g., *Lady Gaga sings well yesterday*

```
∃e. type(e,sing) ∧ agent(e,e470) ∧
    manner(e,well) ∧ time(e,…)
```

Semantic Role Labeling

- A relaxed form of semantic parsing
  - “Shallow semantics”
- e.g., *Boyang wants Asha to give him a linguistics book*
  - (PREDICATE : *wants*, WANTER : *Boyang*, DESIRE : *Asha to give him a linguistics book*)
  - (PREDICATE : *give*, GIVER : *Asha*, RECIPIENT : *him*, GIFT : *a linguistics book*)
- “Thematic roles”: generalizing across predicates
  - e.g., *Agent, Patient,* and *Recipient*

Example Systems

	Asha	*gave*	*Boyang*	*a book*
VerbNet	AGENT		RECIPIENT	THEME
PropBank	ARG0: giver		ARG2: entity given to	ARG1: thing given
FrameNet	DONOR		RECIPIENT	THEME
	Asha	*taught*	*Boyang*	*algebra*
VerbNet	AGENT		RECIPIENT	TOPIC
PropBank	ARG0: teacher		ARG2: student	ARG1: subject
FrameNet	TEACHER		STUDENT	SUBJECT

VerbNet

(Kipper-Schuler, 2005)

- A lexicon of verbs, including 30 core thematic roles
  - E.g., Agent, Patient, Recipient, Theme, Topic

	Asha	*gave*	*Boyang*	*a book*
VerbNet	AGENT		RECIPIENT	THEME
PropBank	ARG0: giver		ARG2: entity given to	ARG1: thing given
FrameNet	DONOR		RECIPIENT	THEME
	Asha	*taught*	*Boyang*	*algebra*
VerbNet	AGENT		RECIPIENT	TOPIC
PropBank	ARG0: teacher		ARG2: student	ARG1: subject
FrameNet	TEACHER		STUDENT	SUBJECT

PropBank

(Palmer, 2005)

- The Proposition Bank
- As a middle ground between generic thematic roles and predicate-specific roles
  - ARG0: proto-agent
  - ARG1: proto-patient
  - Others: verb-specific

	Asha	*gave*	*Boyang*	*a book*
VerbNet	Agent		Recipient	Theme
PropBank	Arg0: giver		Arg2: entity given to	Arg1: thing given
FrameNet	Donor		Recipient	Theme
	Asha	*taught*	*Boyang*	*algebra*
VerbNet	Agent		Recipient	Topic
PropBank	Arg0: teacher		Arg2: student	Arg1: subject
FrameNet	Teacher		Student	Subject

FrameNet

(Fillmore and Baker, 2009)

- Semantic *frames*: descriptions of situations or event
  - *Lexical units*: which evoke the frame, e.g., a certain verb
  - *Frame elements*: which describe the situation/event, just like *roles*
- FrameNet: ~1000 frames and a corpus of more than 200,000 exemplar sentences with annotations
  - Grouping verbs to frames
    - e.g., “teach” and “learn”
  - Shared frame elements
    - e.g., between “GIVE” and “GET”

	Asha	*gave*	*Boyang*	*a book*
VerbNet	AGENT		RECIPIENT	THEME
PropBank	ARG0: giver		ARG2: entity given to	ARG1: thing given
FrameNet	DONOR		RECIPIENT	THEME
	Asha	*taught*	*Boyang*	*algebra*
VerbNet	AGENT		RECIPIENT	TOPIC
PropBank	ARG0: teacher		ARG2: student	ARG1: subject
FrameNet	TEACHER		STUDENT	SUBJECT

Semantic Role Labeling

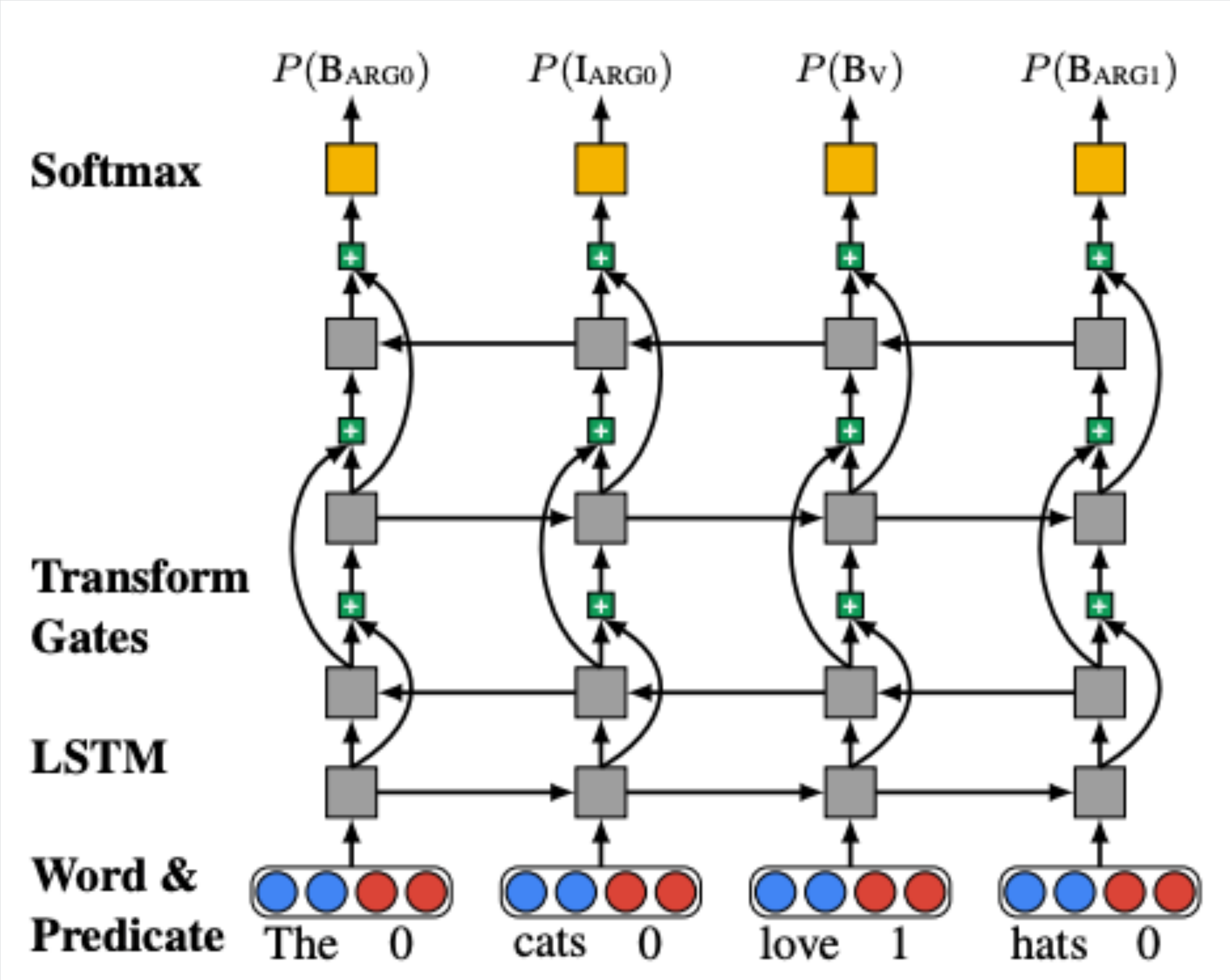
- Can be formulated as a sequence labeling problem
- Recall: Named Entity Recognition

B-PER	I-PER	O	O	O	B-LOC	O	O	O	B-ORG	O	O
Barack	*Obama*	*will*	*travel*	*to*	*Hangzhou*	*today*	*for*	*the*	*G20*	*meeting*	*.*

PERSON (*Barack Obama*), LOC (*Hangzhou*), ORG (*G20*)

Semantic Role Labeling

- Can be formulated as a sequence labeling problem
- Tagging for SRL



(He et al., 2017)

Abstract Meaning Representation (AMR)

- Semantic Role Labeling only extracts the predicate-argument relationship, but does not process “coreference resolution”

- e.g., *<u>The whale</u> wants the captain to pursue <u>him</u>*
  - PropBank SRL:

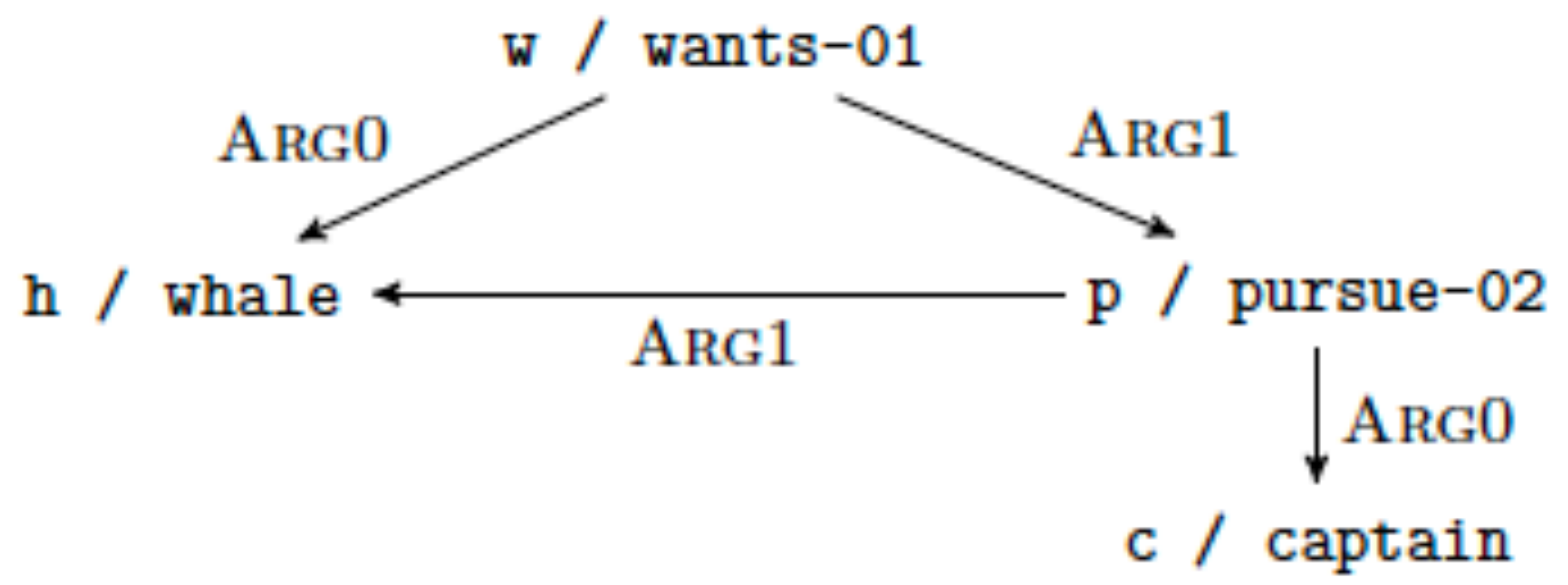
    - (PREDICATE : *wants*, ARG0 : *the whale*, ARG1 : *the captain to pursue him*)
    - (PREDICATE : *pursue*, ARG0 : *the captain*, ARG1 : *him*)

- AMR unifies them into a *graph structure*

Abstract Meaning Representation (AMR)

- e.g., *<u>The whale</u> wants the captain to pursue <u>him</u>*

```
(w / want-01
  :ARG0 (h / whale)
  :ARG1 (p / pursue-02
          :ARG0 (c / captain)
          :ARG1 h))
```

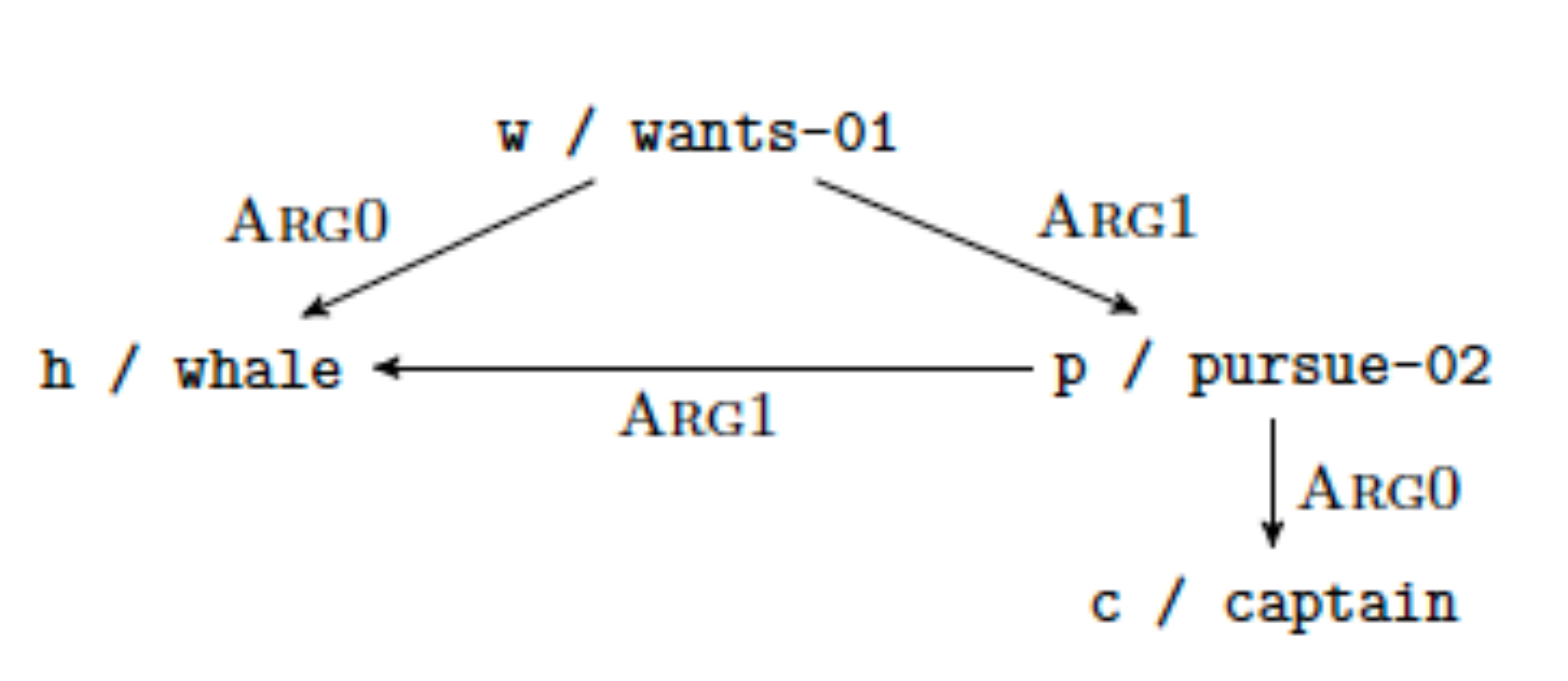


- Nodes are *variables;* edges indicate *concepts*

AMR Parsing

- e.g., graph-based methods, similar to doing dependency parsing
  - Scoring the edges: should there be a link from “wants” to “whale”?

https://github.com/nschneid/amr-tutorial/tree/master/slides



Outline

- Introduction to Semantic Parsing
  - Model Theoretic Semantics
  - Example: CCG Parsing
- (Shallow Parsing) Predicate-Argument Semantics (Eisenstein Ch13)
- Applications of Semantic Parsing

Materials adapted from ACL 2018 tutorials on neural semantic parsing
https://github.com/allenai/acl2018-semantic-parsing-tutorial

Semantic Parsing

Sentence

↓

Semantic Parser

↓

Meaning Representation

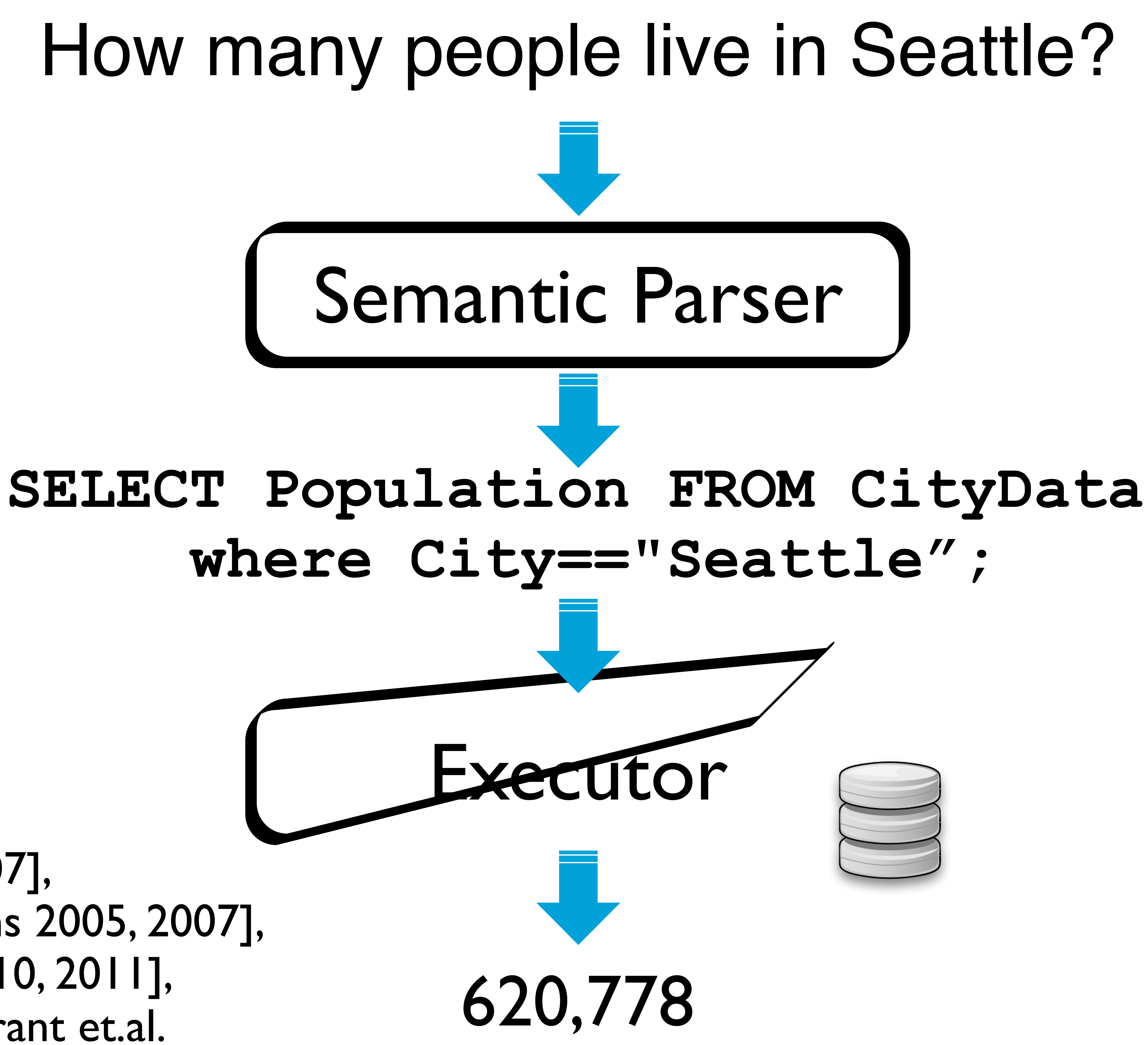
↓

Executor

↓

Response

Semantic Parsing: QA



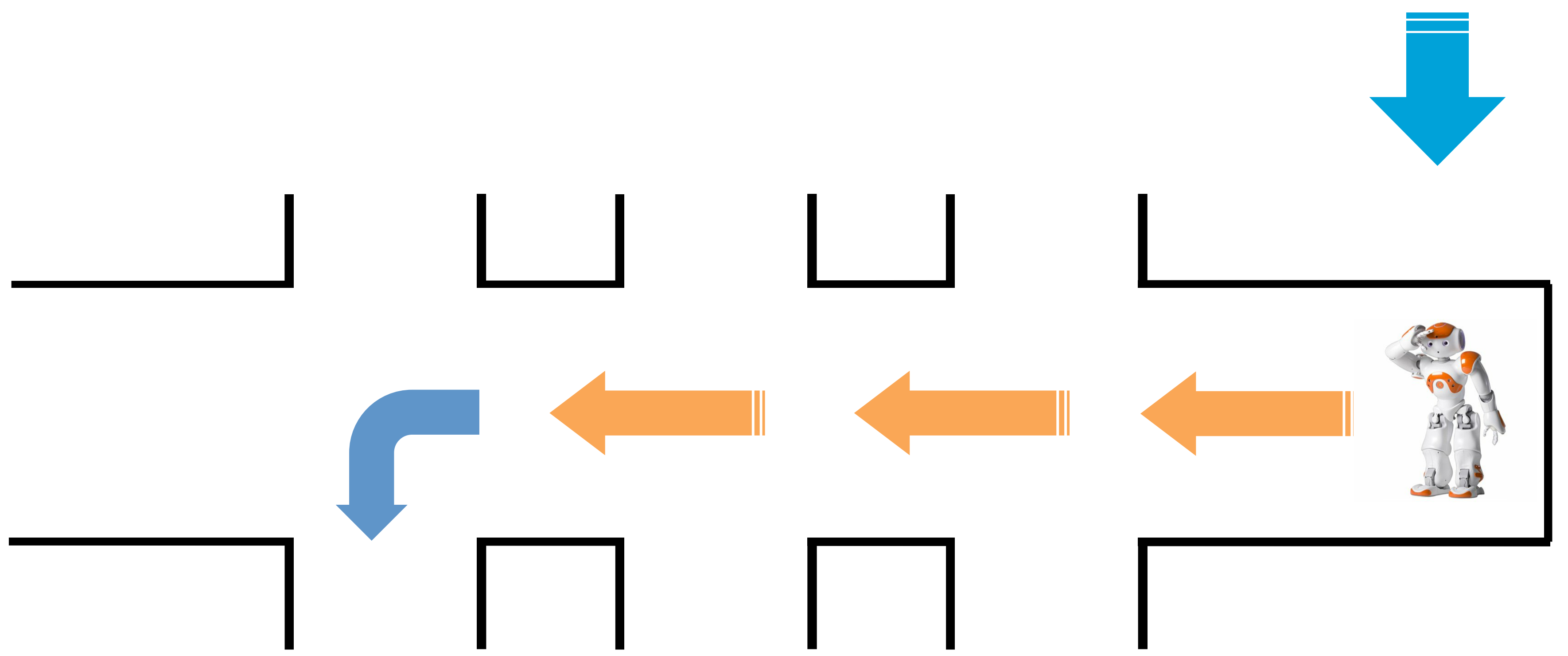
[Wong & Mooney 2007],
[Zettlemoyer & Collins 2005, 2007],
[Kwiatkowski et.al 2010, 2011],
[Liang et.al. 2011],[Berant et.al. 2013,2014],[Reddy et.al, 2014,2016],
[Dong and Lapata, 2016] .....

Semantic Parsing: Instructions

Go to the third junction and take a left

Semantic Parser

```
(do-seq(do-n-times 3
 (move-to forward-loc
  (do-until
   (junction current-loc
    (move-to forward-loc))))
 (turn-right))
```



[Chen & Mooney 2011]
[Matuszek et al 2012]
[Artzi & Zettlemoyer 2013]
[Mei et.al. 2015][Andreas et al, 2015]
[Fried at al, 2018] ....

Language to Meaning

More informative

Language to Meaning

Information Extraction

Recover information about pre-specified relations and entities

More informative

Example Task

Relation Extraction


The New York Times
OBAMA
RACIAL BARRIER FALLS IN DECISIVE VICTORY
Democrats in Congress Strengthen Grip

$$is\_a(OBAMA, PRESIDENT)$$

Language to Meaning

Broad-coverage
Semantics

Focus on specific
phenomena (e.g., AMR)

More informative

Example Task

Summarization

The New York Times

OBAMA

RACIAL BARRIER FALLS IN DECISIVE VICTORY

Democrats in Congress
Strengthen Grip

Obama wins
election. Big party
in Chicago.
Romney a bit
down, asks for
some tea.

Language to Meaning

Semantic Parsing

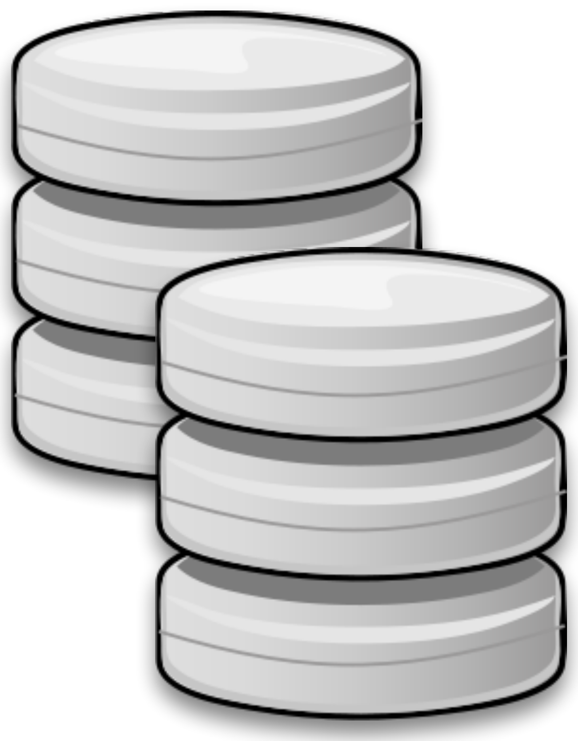
Recover **complete** meaning representation

More informative

Example Task

Database Query

What states border Texas?



Oklahoma
New Mexico
Arkansas
Louisiana

Language to Meaning

Semantic Parsing

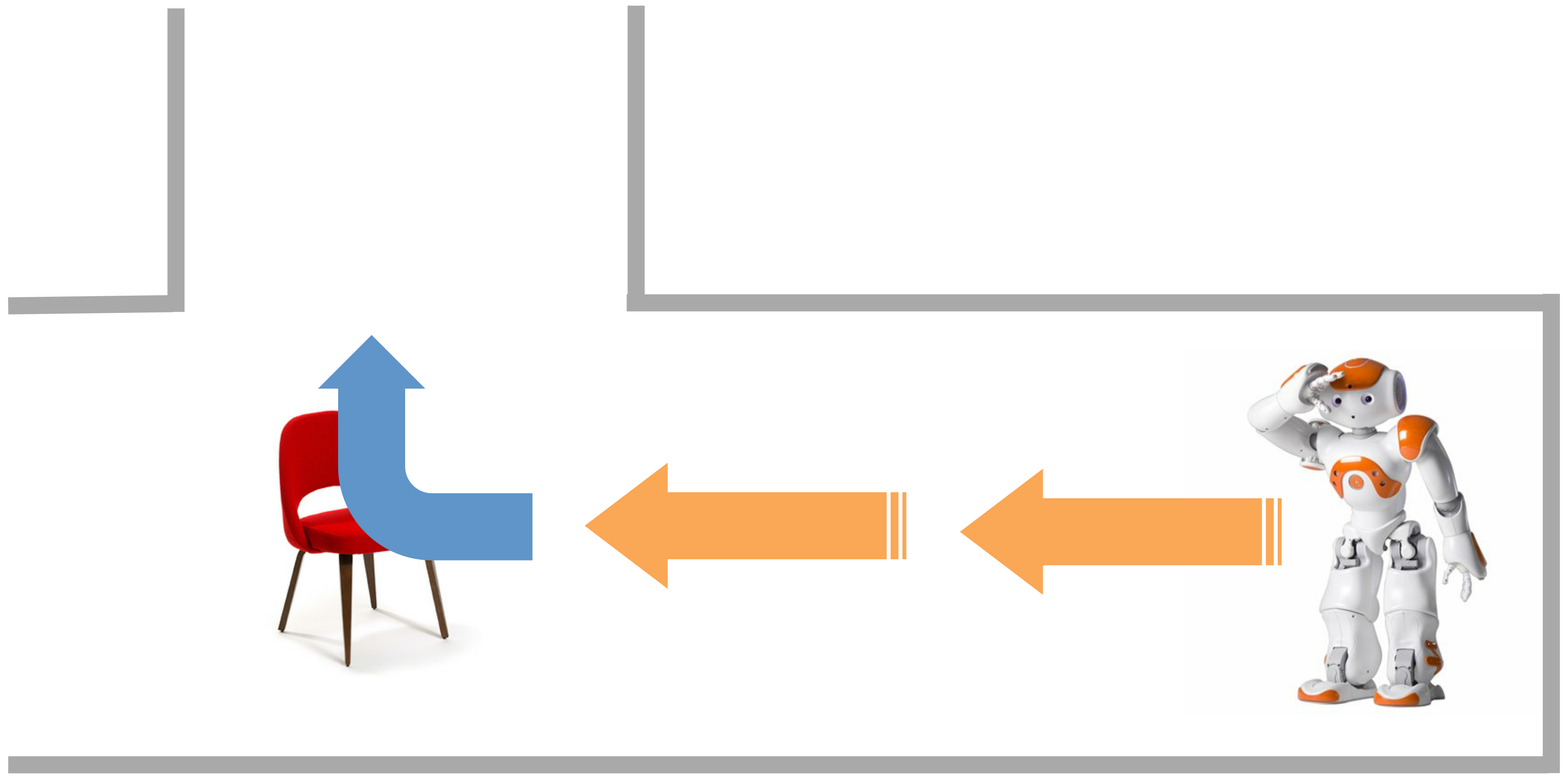
Recover **complete** meaning representation

More informative

Example Task

Instructing a Robot

at the chair,
turn right



Language to Meaning

Complete meaning is sufficient to complete the task

- Convert to database query to get the answer
- Allow a robot to do planning

Language to Meaning

at the chair, move forward three steps past the sofa

$$\lambda a.pre(a, \iota x.chair(x)) \wedge move(a) \wedge len(a, 3) \wedge$$
$$dir(a, forward) \wedge past(a, \iota y.sofa(y))$$

Language to Meaning

Semantic Parsing

Recover **complete** meaning representation

More informative

at the chair, move forward three steps past the sofa

$$\lambda a.pre(a, \iota x.chair(x)) \wedge move(a) \wedge len(a, 3) \wedge dir(a, forward) \wedge past(a, \iota y.sofa(y))$$

Language to Meaning

at the chair, move forward three steps past the sofa

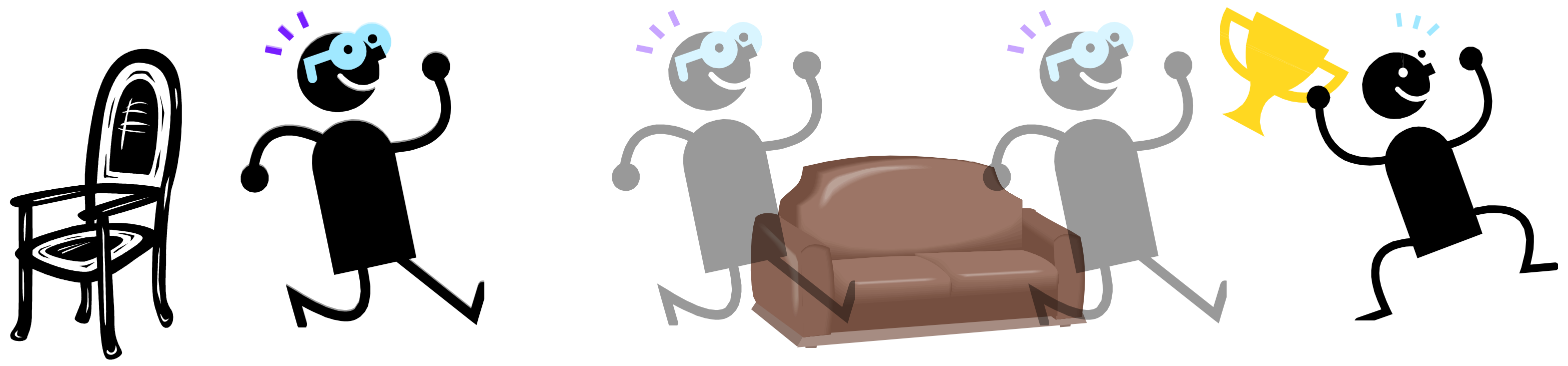
$$\lambda a.pre(a, \iota x.chair(x)) \land move(a) \land len(a, 3) \land$$
$$dir(a, forward) \land past(a, \iota y.sofa(y))$$

Learn

$$f : \text{sentence} \rightarrow \text{logical form}$$

Language to Meaning

at the chair, move forward three steps past the sofa



Learn

$$f : \text{sentence} \rightarrow \text{logical form}$$

Research Focuses

- Generating executable representations
- Understanding in a situated environment
- Generalizing to broad domains
- Sequential language understanding

Executable Representations

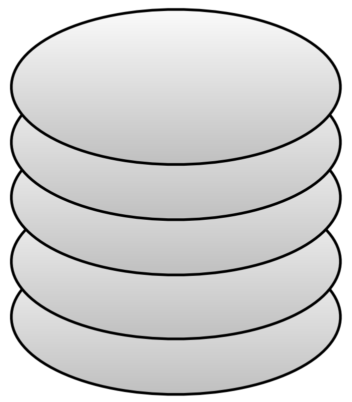
- **Goal:** generate compositional, executable, formal representation, e.g. code
- Formal representation requires following strict constraints:
  - Syntax (e.g., correct number of parentheses)
  - Semantics (e.g., calling function with the right type of arguments)

Request:

Show me flights from Seattle to Boston next Monday

SQL query:

```
(SELECT DISTINCT flight.flight_id FROM flight WHERE (flight.from_airport IN (SELECT
airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code
FROM city WHERE city.city_name = 'SEATTLE'))) AND (flight.to_airport IN (SELECT airport_service.airport_code
FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM city WHERE
city.city_name = 'BOSTON'))) AND (flight.flight_days IN (SELECT days.days_code FROM days WHERE
days.day_name IN (SELECT date_day.day_name FROM date_day WHERE date_day.year = 1993 AND
date_day.month_number = 2 AND date_day.day_number = 8))));
```



Database

Result:

31 flights available

[Hemphill et al. 1990; Dahl et al. 1994]

Request:

Copy the content of file ‘file.txt’ to file ‘file2.txt’

↓

```
shutil.copy(‘file.txt’, ‘file2.txt’)
```

Request:

Check if all elements in list ‘mylist’ are the same

↓

```
len(set(mylist)) == 1
```

[Yin et al. 2018]

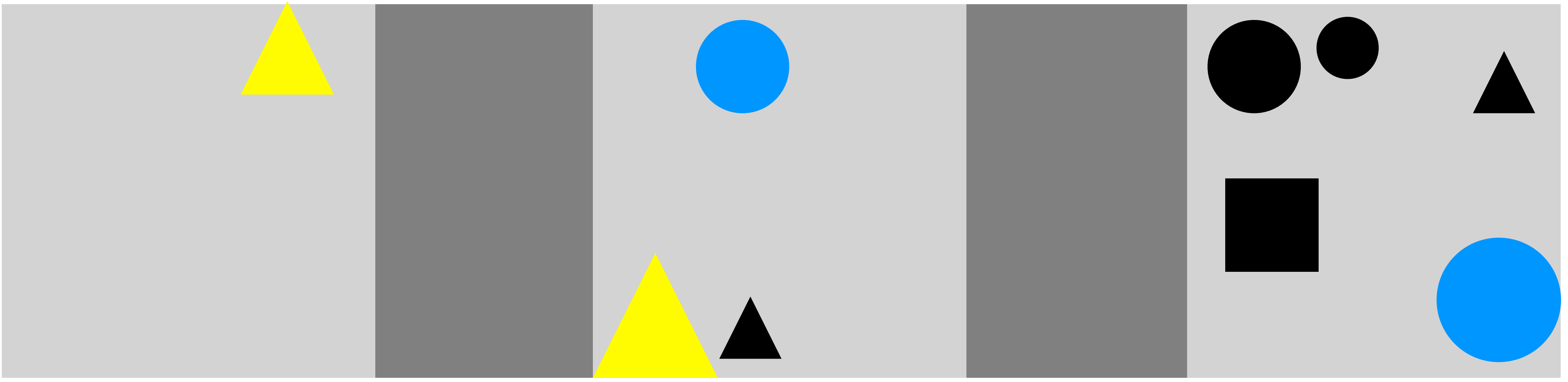
Research Focuses

- Generating executable representations
- Understanding in a situated environment
- Generalizing to broad domains
- Sequential language understanding

NLVR

Task

Image:

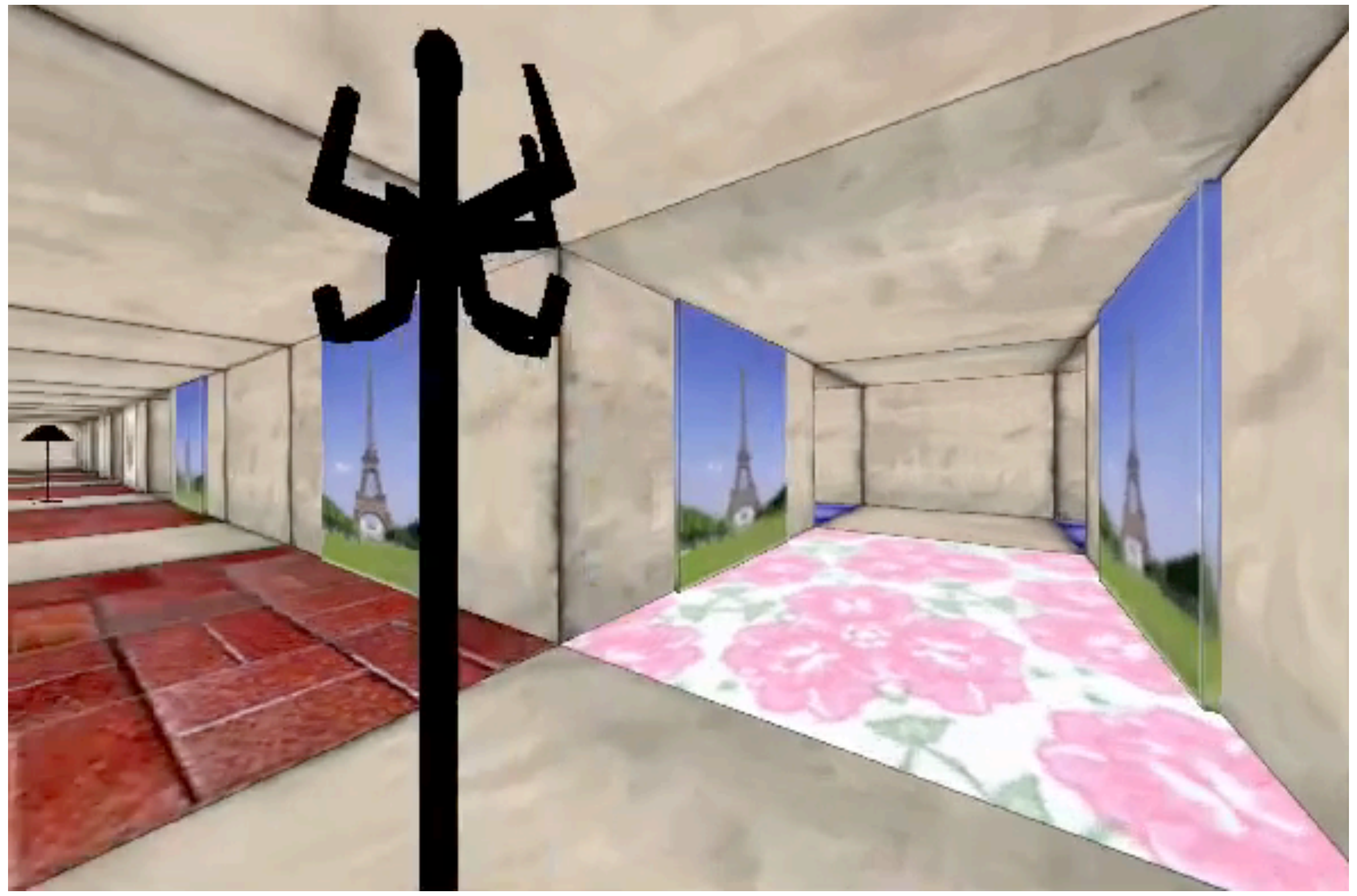


There is a box with 3 items of all 3 different colors.

TRUE or False?

$$\lambda x.\ \lambda y.\ \lambda z.\ \lambda w.\ box(x) \wedge count(x, 3) \wedge object(y) \wedge object(z) \wedge object(w) \wedge in(y, x) \wedge in(z, x) \wedge in(w, x) \wedge \neg(color(y) == color(z)) \wedge \neg(color(y) == color(w)) \wedge \neg(color(z) == color(w))$$

[Suhr et al. 2017]



Instructions:

Place your back against the wall of the T intersection

Turn left

Go forward along the pink flowered carpet hall two segments to the intersection with the brick hall

Example from Artzi and Zettlemoyer 2013

[MacMahon et al. 2006; Chen and Mooney 2011]

Research Focuses

- Generating executable representations
- Understanding in a situated environment
- Generalizing to broad domains
- Sequential language understanding

Semantic Parsing for Machine Reading Comprehension

- Especially useful for numerical reasoning-required tasks

Passage	Question & Answer
Sorting	
...Jaguars kicker **Josh Scobee** managed to get a 48-yard field goal...with kicker Nate Kaeding getting a 23-yard field goal...	**Question:** Who kicked the longest field goal? **Program:** ARGMAX(KV(PASSAGE_SPAN(50,53),VALUE(9)), KV(PASSAGE_SPAN(92,94),VALUE(11))) **Result:** ARGMAX(KV('Josh Scobee', 48), KV('Nate Kaeding', 23)) = 'Josh Scobee'

Neuro-Symbolic Reader (Chen et al., 2020)

Table-based Fact Checking

United States House of Representatives Elections, 1972

District	Incumbent	Party	Result	Candidates
California 3	John E. Moss	democratic	re-elected	John E. Moss (d) 69.9% John Rakus (r) 30.1%
California 5	Phillip Burton	democratic	re-elected	Phillip Burton (d) 81.8% Edlo E. Powell (r) 18.2%
California 8	George Paul Miller	democratic	lost renomination democratic hold	Pete Stark (d) 52.9% Lew M. Warden , Jr. (r) 47.1%
California 14	Jerome R. Waldie	republican	re-elected	Jerome R. Waldie (d) 77.6% Floyd E. Sims (r) 22.4%
California 15	John J. Mcfall	republican	re-elected	John J. Mcfall (d) unopposed

There are five candidates in total, two of them are democrats and three of them are republicans.

How to verify whether this statement is correct or incorrect?

```
Eq(Count(T),5)^Eq(Count(Filter(T,
party='democratic'),2))^Eq(Count(Filter(T,
party='republican'),3)
```

TabFact (Chen et al., 2020)

Research Focuses

- Generating executable representations
- Understanding in a situated environment
- Generalizing to broad domains
- Sequential language understanding

ATIS

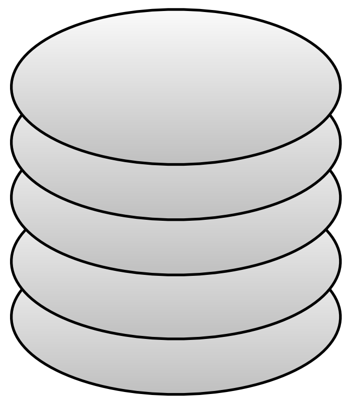
Task

Request:

Show me flights from Seattle to Boston next Monday

SQL query:

```
(SELECT DISTINCT flight.flight_id FROM flight WHERE (flight.from_airport IN (SELECT
airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code
FROM city WHERE city.city_name = 'SEATTLE'))) AND (flight.to_airport IN (SELECT airport_service.airport_code
FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM city WHERE
city.city_name = 'BOSTON'))) AND (flight.flight_days IN (SELECT days.days_code FROM days WHERE
days.day_name IN (SELECT date_day.day_name FROM date_day WHERE date_day.year = 1993 AND
date_day.month_number = 2 AND date_day.day_number = 8))));
```



Database

Result:

31 flights available

[Hemphill et al. 1990; Dahl et al. 1994]

ATIS

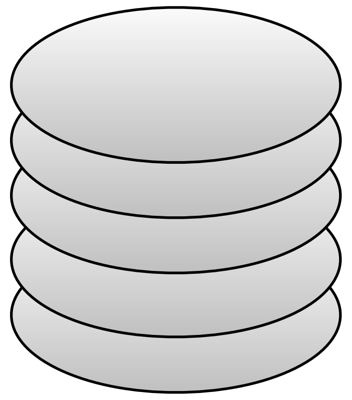
Task

Previous request:
Show me flights from Seattle to Boston next Monday

Request:
On American Airlines

SQL query:

```
(SELECT DISTINCT flight.flight_id FROM flight WHERE (flight.from_airport IN (SELECT
airport_service.airport_code FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code
FROM city WHERE city.city_name = 'SEATTLE'))) AND (flight.to_airport IN (SELECT airport_service.airport_code
FROM airport_service WHERE airport_service.city_code IN (SELECT city.city_code FROM city WHERE
city.city_name = 'BOSTON'))) AND (flight.flight_days IN (SELECT days.days_code FROM days WHERE
days.day_name IN (SELECT date_day.day_name FROM date_day WHERE date_day.year = 1993 AND
date_day.month_number = 2 AND date_day.day_number = 8))) AND flight.airline_code = 'AA');
```



Database

Result:
5 flights available

[Hemphill et al. 1990; Dahl et al. 1994]

Summary

- Introduction to Semantic Parsing
  - Model Theoretic Semantics
  - Example: CCG Parsing
- (Shallow Parsing) Predicate-Argument Semantics (Eisenstein Ch13)
- Applications of Semantic Parsing