Introduction to NLP

365.

Semantic Parsing

Semantic Parsing

- Converting natural language to a logical form
  - e.g., executable code for a specific application
- Example:
  - Airline reservations
  - Geographical query systems

Stages of Semantic Parsing

- Input
  - Sentence
- Syntactic Analysis
  - Syntactic structure
- Semantic Analysis
  - Semantic representation

Compositional Semantics

- Add semantic attachments to CFG rules
- Compositional semantics
  - Parse the sentence syntactically
  - Associate some semantics to each word
  - Combine the semantics of words and non-terminals recursively
  - Until the root of the sentence

Example

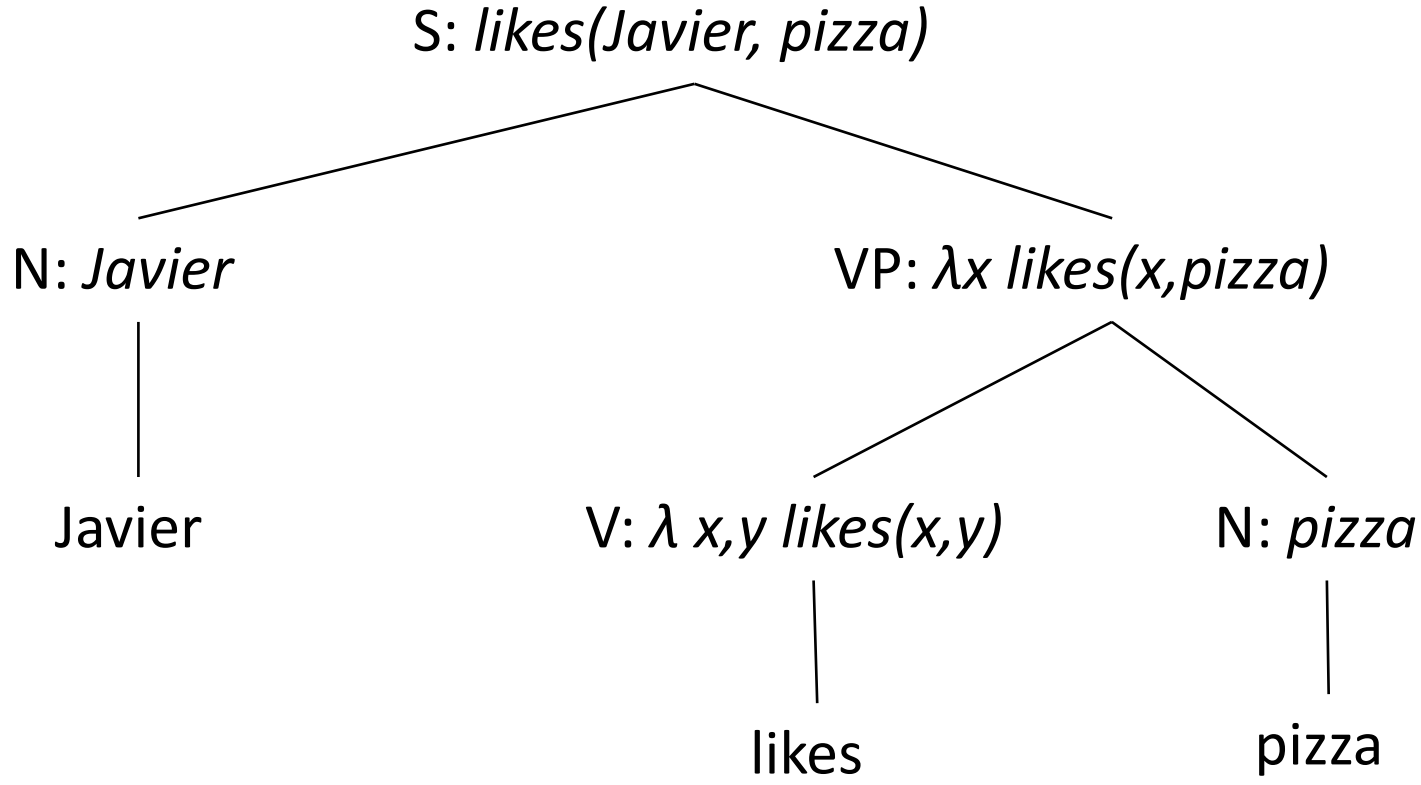
- Input
  - Javier likes pizza
- Output
  - *like(Javier, pizza)*

Example

```
S  -> NP VP    {VP.Sem(NP.Sem)}     t
VP -> V NP     {V.Sem(NP.Sem)}      <e,t>
NP -> N        {N.Sem}              e
V  -> likes    {λ x,y likes(x,y)    <e,<e,t>>
N  -> Javier   {Javier}             e
N  -> pizza    {pizza}              e
```

Semantic Parsing

- Associate a semantic expression with each node



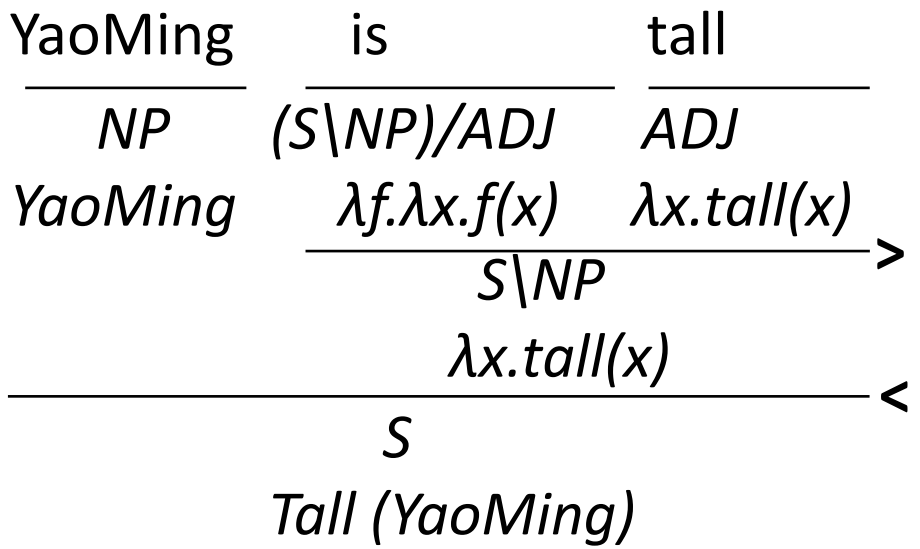
Grammar with Semantic Attachments

Grammar Rule	Semantic Attachment
$S \rightarrow NP\ VP$	$\{NP.sem(VP.sem)\}$
$NP \rightarrow Det\ Nominal$	$\{Det.sem(Nominal.sem)\}$
$NP \rightarrow ProperNoun$	$\{ProperNoun.sem\}$
$Nominal \rightarrow Noun$	$\{Noun.sem\}$
$VP \rightarrow Verb$	$\{Verb.sem\}$
$VP \rightarrow Verb\ NP$	$\{Verb.sem(NP.sem)\}$
$Det \rightarrow every$	$\{\lambda P.\lambda Q.\forall x P(x) \Rightarrow Q(x)\}$
$Det \rightarrow a$	$\{\lambda P.\lambda Q.\exists x P(x) \wedge Q(x)\}$
$Noun \rightarrow restaurant$	$\{\lambda r.Restaurant(r)\}$
$ProperNoun \rightarrow Matthew$	$\{\lambda m.m(Matthew)\}$
$ProperNoun \rightarrow Franco$	$\{\lambda f.f(Franco)\}$
$ProperNoun \rightarrow Frasca$	$\{\lambda f.f(Frasca)\}$
$Verb \rightarrow closed$	$\{\lambda x.\exists e Closed(e) \wedge ClosedThing(e,x)\}$
$Verb \rightarrow opened$	$\{\lambda w.\lambda z.w(\lambda x.\exists e Opened(e) \wedge Opener(e,z) \wedge Opened(e,x))\}$

Example from Jurafsky and Martin

Using CCG (Steedman 1996)

- CCG representations for semantics
  - *ADJ: λx.tall(x)*
  - (S\NP)/ADJ : *λf.λx.f(x)*
  - *NP: YaoMing*



CCG Parsing

- Example:
  - https://bitbucket.org/yoavartzi/spf
- Tutorial by Artzi, FitzGerald, Zettlemoyer
  - http://yoavartzi.com/pub/afz-tutorial.acl.2013.pdf

GeoQuery (Zelle and Mooney 1996)

What is the capital of the state with the largest population?
`answer(C, (capital(S,C), largest(P, (state(S), population(S,P)))))`.

What are the major cities in Kansas?
`answer(C, (major(C), city(C), loc(C,S), equal(S,stateid(kansas))))`.

Type	Form	Example
country	`countryid(Name)`	`countryid(usa)`
city	`cityid(Name, State)`	`cityid(austin,tx)`
state	`stateid(Name)`	`stateid(texas)`
river	`riverid(Name)`	`riverid(colorado)`
place	`placeid(Name)`	`placeid(pacific)`

Form	Predicate
`capital(C)`	`C` is a capital (city).
`city(C)`	`C` is a city.
`major(X)`	`X` is major.
`place(P)`	`P` is a place.
`river(R)`	`R` is a river.
`state(S)`	`S` is a state.
`capital(C)`	`C` is a capital (city).
`area(S,A)`	The area of `S` is `A`.
`capital(S,C)`	The capital of `S` is `C`.
`equal(V,C)`	variable `V` is ground term `C`.
`density(S,D)`	The (population) density of `S` is `P`
`elevation(P,E)`	The elevation of `P` is `E`.
`high_point(S,P)`	The highest point of `S` is `P`.
`higher(P1,P2)`	`P1`'s elevation is greater than `P2`'s.
`loc(X,Y)`	`X` is located in `Y`.
`low_point(S,P)`	The lowest point of `S` is `P`.
`len(R,L)`	The length of `R` is `L`.
`next_to(S1,S2)`	`S1` is next to `S2`.
`size(X,Y)`	The size of `X` is `Y`.
`traverse(R,S)`	`R` traverses `S`.

Zettlemoyer and Collins (2005)

a) What states border Texas
$\lambda x.state(x) \wedge borders(x, texas)$

b) What is the largest state
$\arg\max(\lambda x.state(x), \lambda x.size(x))$

c) What states border the state that borders the most states
$\lambda x.state(x) \wedge borders(x, \arg\max(\lambda y.state(y),$
$\lambda y.count(\lambda z.state(z) \wedge borders(y, z))))$

Utah	:=	NP
Idaho	:=	NP
borders	:=	$(S\backslash NP)/NP$

Utah	:=	$NP : utah$
Idaho	:=	$NP : idaho$
borders	:=	$(S\backslash NP)/NP : \lambda x.\lambda y.borders(y, x)$

a)

Utah	borders	Idaho
NP	$(S\backslash NP)/NP$	NP
utah	$\lambda x.\lambda y.borders(y, x)$	*idaho*
	$(S\backslash NP)$ (>)	
	$\lambda y.borders(y, idaho)$	
	S (<)	
	$borders(utah, idaho)$	

b)

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*
$S/(S\backslash NP)$ (>)		$(S\backslash NP)$ (>)	
$\lambda g.\lambda x.state(x) \wedge g(x)$		$\lambda y.borders(y, texas)$	
S (>)			
$\lambda x.state(x) \wedge borders(x, texas)$			

Zettlemoyer and Collins (2005)

states	:=	$N : \lambda x.state(x)$
major	:=	$N/N : \lambda f.\lambda x.major(x) \wedge f(x)$
population	:=	$N : \lambda x.population(x)$
cities	:=	$N : \lambda x.city(x)$
rivers	:=	$N : \lambda x.river(x)$
run through	:=	$(S\backslash NP)/NP : \lambda x.\lambda y.traverse(y, x)$
the largest	:=	$NP/N : \lambda f.\arg\max(f, \lambda x.size(x))$
river	:=	$N : \lambda x.river(x)$
the highest	:=	$NP/N : \lambda f.\arg\max(f, \lambda x.elev(x))$
the longest	:=	$NP/N : \lambda f.\arg\max(f, \lambda x.len(x))$

Figure 6: Ten learned lexical items that had highest associated parameter values from a randomly chosen development run in the Geo880 domain.

Zettlemoyer and Collins (2005)

- PCCG learning
- Lexicon Λ, parameter vector θ
- GENLEX

Rules		Categories produced from logical form
Input Trigger	Output Category	$\arg\max(\lambda x.state(x) \wedge borders(x, texas), \lambda x.size(x))$
constant c	$NP : c$	$NP : texas$
arity one predicate p_1	$N : \lambda x.p_1(x)$	$N : \lambda x.state(x)$
arity one predicate p_1	$S\backslash NP : \lambda x.p_1(x)$	$S\backslash NP : \lambda x.state(x)$
arity two predicate p_2	$(S\backslash NP)/NP : \lambda x.\lambda y.p_2(y, x)$	$(S\backslash NP)/NP : \lambda x.\lambda y.borders(y, x)$
arity two predicate p_2	$(S\backslash NP)/NP : \lambda x.\lambda y.p_2(x, y)$	$(S\backslash NP)/NP : \lambda x.\lambda y.borders(x, y)$
arity one predicate p_1	$N/N : \lambda g.\lambda x.p_1(x) \wedge g(x)$	$N/N : \lambda g.\lambda x.state(x) \wedge g(x)$
literal with arity two predicate p_2 and constant second argument c	$N/N : \lambda g.\lambda x.p_2(x, c) \wedge g(x)$	$N/N : \lambda g.\lambda x.borders(x, texas) \wedge g(x)$
arity two predicate p_2	$(N\backslash N)/NP : \lambda x.\lambda g.\lambda y.p_2(x, y) \wedge g(x)$	$(N\backslash N)/NP : \lambda g.\lambda x.\lambda y.borders(x, y) \wedge g(x)$
an $\arg\max$ / $\min$ with second argument arity one function f	$NP/N : \lambda g.\arg\max / \min(g, \lambda x.f(x))$	$NP/N : \lambda g.\arg\max(g, \lambda x.size(x))$
an arity one numeric-ranged function f	$S/NP : \lambda x.f(x)$	$S/NP : \lambda x.size(x)$

Figure 3: The rules that define GENLEX. We use the term *predicate* to refer to a function that returns a truth value; *function* to refer to all other functions; and *constant* to refer to constants of type e. Each row represents a rule. The first column lists the triggers that identify some sub-structure within a logical form L, and then generate a category. The second column lists the category that is created. The third column lists example categories that are created when the rule is applied to the logical form at the top of this column.

Dong and Lapata (2016)

JOBS This benchmark dataset contains 640 queries to a database of job listings. Specifically, questions are paired with Prolog-style queries. We used the same training-test split as Zettlemoyer and Collins (2005) which contains 500 training and 140 test instances. Values for the variables company, degree, language, platform, location, job area, and number are identified.

GEO This is a standard semantic parsing benchmark which contains 880 queries to a database of U.S. geography. GEO has 880 instances split into a training set of 680 training examples and 200 test examples (Zettlemoyer and Collins, 2005). We used the same meaning representation based on lambda-calculus as Kwiatkowski et al. (2011). Values for the variables city, state, country, river, and number are identified.

ATIS This dataset has 5,410 queries to a flight booking system. The standard split has 4,480 training instances, 480 development instances, and 450 test instances. Sentences are paired with lambda-calculus expressions. Values for the variables date, time, city, aircraft code, airport, airline, and number are identified.

Dataset	Length	Example
JOBS	9.80 22.90	*what microsoft jobs do not require a bscs?* answer(company(J,'microsoft'),job(J),not((req_deg(J,'bscs'))))
GEO	7.60 19.10	*what is the population of the state with the largest area?* (population:i (argmax $0 (state:t $0) (area:i $0)))
ATIS	11.10 28.10	*dallas to san francisco leaving after 4 in the afternoon please* (lambda $0 e (and (>(departure_time $0) 1600:ti) (from $0 dallas:ci) (to $0 san_francisco:ci)))
IFTTT	6.95 21.80	*Turn on heater when temperature drops below 58 degree* TRIGGER: Weather - Current_temperature_drops_below - ((Temperature (58)) (Degrees_in (f))) ACTION: WeMo_Insight_Switch - Turn_on - ((Which_switch? ("")))

Table 1: Examples of natural language descriptions and their meaning representations from four datasets. The average length of input and output sequences is shown in the second column.

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

Table 2: Evaluation results on JOBS.

Dong and Lapata (2016)

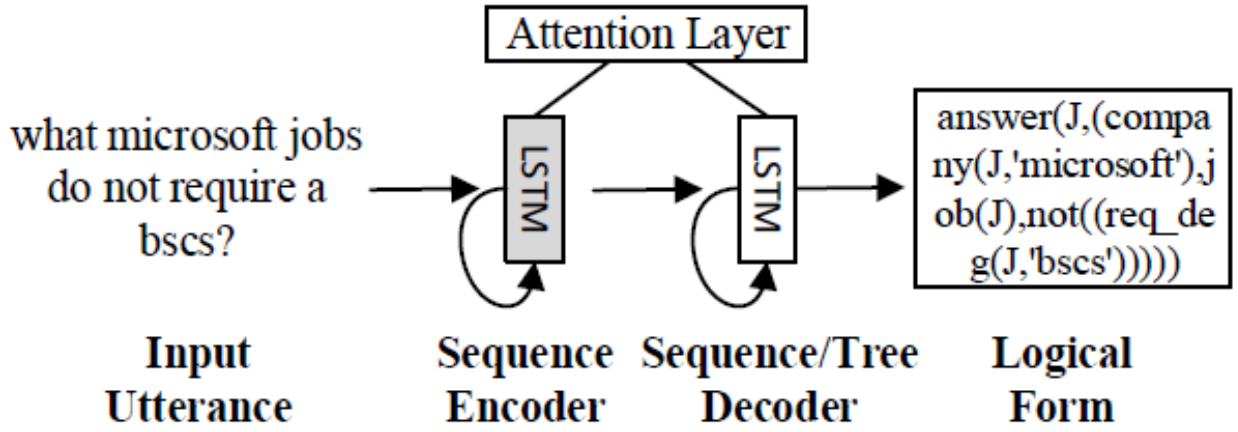


Figure 1: Input utterances and their logical forms are encoded and decoded with neural networks. An attention layer is used to learn soft alignments.

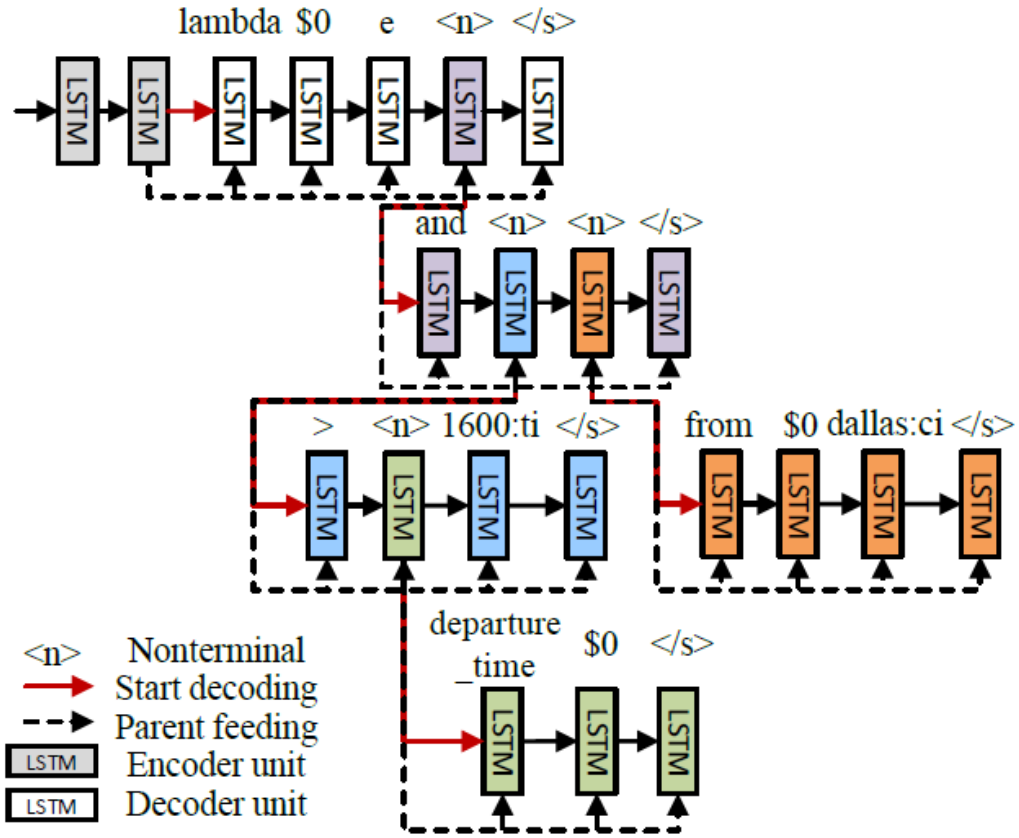
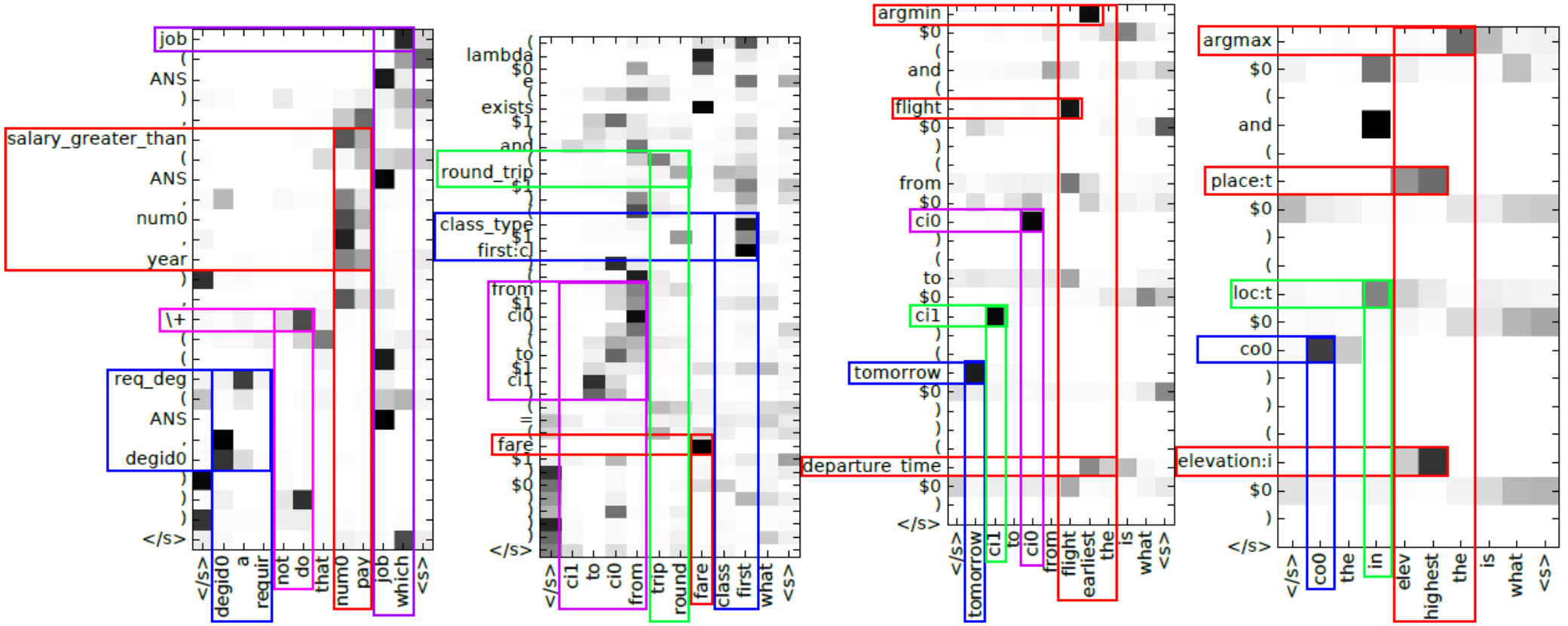


Figure 3: Sequence-to-tree (SEQ2TREE) model with a hierarchical tree decoder.

Dong and Lapata (2016)



(a) which jobs pay num0 that do not require a degid0

(b) what's first class fare round trip from ci0 to ci1

(c) what is the earliest flight from ci0 to ci1 tomorrow

(d) what is the highest elevation in the co0

Figure 6: Alignments (same color rectangles) produced by the attention mechanism (darker color represents higher attention score). Input sentences are reversed and stemmed. Model output is shown for SEQ2SEQ (a, b) and SEQ2TREE (c, d).

Dong and Lapata (2018)

Dataset	Length	Example
GEO	7.6	x : *which state has the most rivers running through it?*
	13.7	y : (argmax $0 (state:t $0) (count $1 (and (river:t $1) (loc:t $1 $0))))
	6.9	a : (argmax#1 state:t@1 (count#1 (and river:t@1 loc:t@2)))
ATIS	11.1	x : *all flights from dallas before 10am*
	21.1	y : (lambda $0 e (and (flight $0) (from $0 dallas:ci) (< (departure_time $0) 1000:ti)))
	9.2	a : (lambda#2 (and flight@1 from@2 (< departure_time@1 ?)))
DJANGO	14.4	x : *if length of bits is lesser than integer 3 or second element of bits is not equal to string 'as' ,*
	8.7	y : if len(bits) < 3 or bits[1] != 'as':
	8.0	a : if len (NAME) < NUMBER or NAME [NUMBER] != STRING :
WIKISQL	17.9	Table schema: ‖*Pianist*‖*Conductor*‖*Record Company*‖*Year of Recording*‖*Format*‖
	13.3	x : *What record company did conductor Mikhail Snitko record for after 1996?*
	13.0	y : SELECT *Record Company* WHERE (*Year of Recording* > *1996*) AND (*Conductor* = *Mikhail Snitko*)
	2.7	a : WHERE > AND =

Table 1: Examples of natural language expressions x, their meaning representations y, and meaning sketches a. The average number of tokens is shown in the second column.

Dong and Lapata 2018

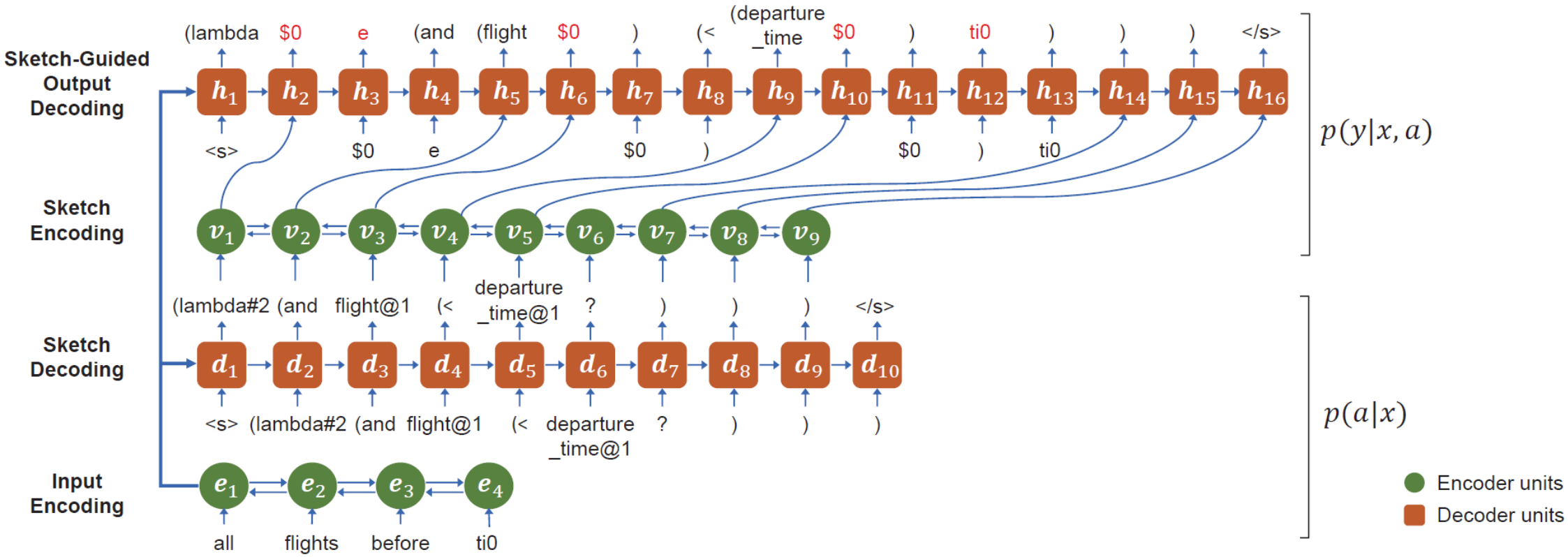


Figure 1: We first generate the meaning sketch a for natural language input x. Then, a fine meaning decoder fills in the missing details (shown in red) of meaning representation y. The coarse structure a is used to guide and constrain the output decoding.

Dong and Lapata 2018

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

Table 2: Accuracies on GEO and ATIS.

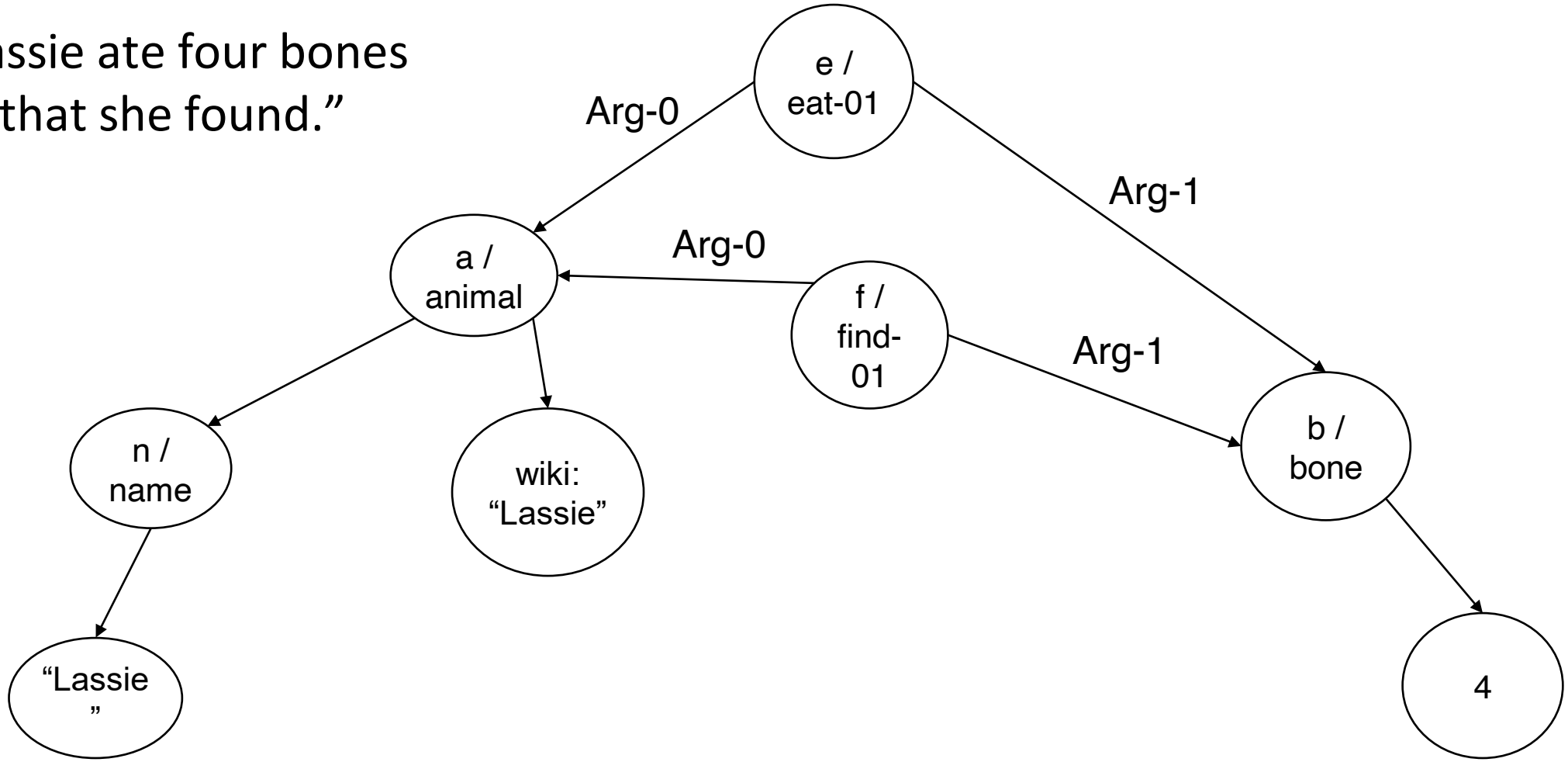
Abstract Meaning Representation (AMR)

- http://amr.isi.edu/
- Single structure that includes:
  - Predicate-Argument Structure
  - Named Entity Recognition
  - Coreference Resolution
  - Wikification

[slide from Jonathan Kummerfeld]

Example

"Lassie ate four bones that she found."



[slide from Jonathan Kummerfeld]

Example

About 14,000 people fled their homes at the weekend after a local tsunami warning was issued, the UN said on its Web site

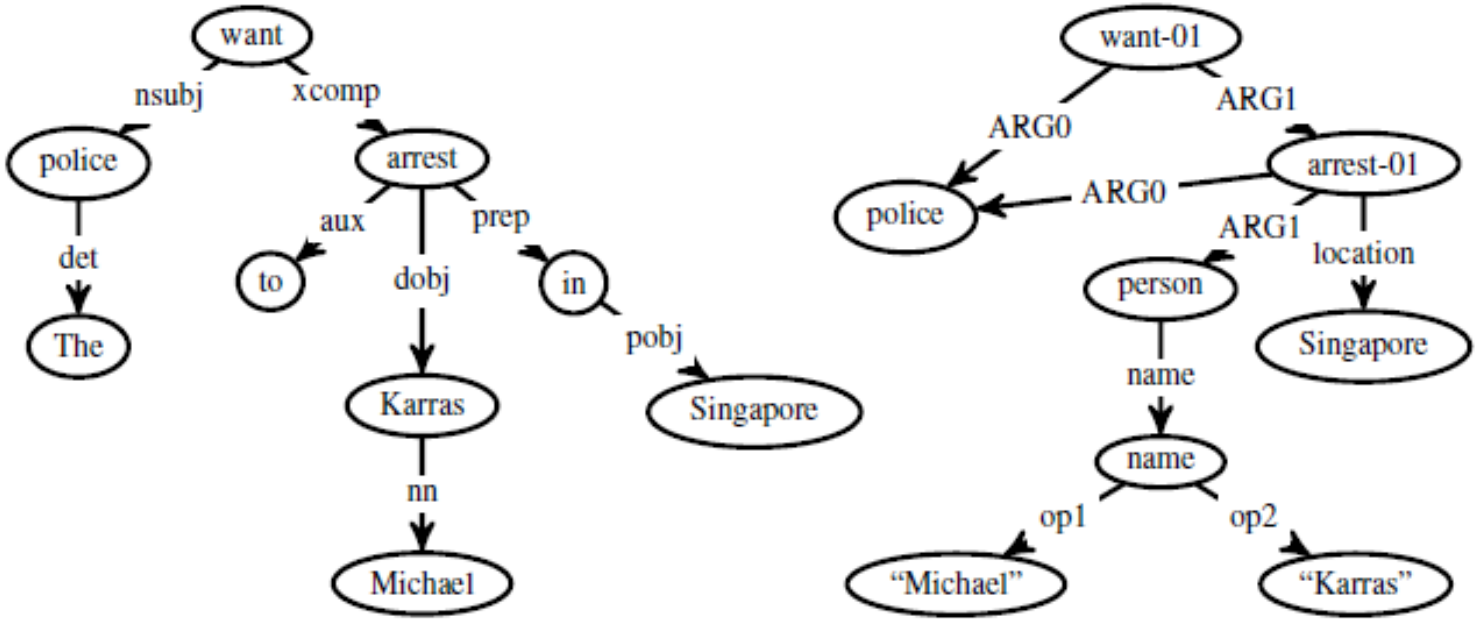
```
(s / say-01
  :ARG0 (g / organization
          :name (n / name
                  :op1 "UN"))
  :ARG1 (f / flee-01
          :ARG0 (p / person
                  :quant (a / about
                           :op1 14000))
          :ARG1 (h / home :poss p)
          :time (w / weekend)
          :time (a2 / after
                  :op1 (w2 / warn-01
                           :ARG1 (t / tsunami)
                           :location (l / local))))
  :medium (s2 / site
            :poss g
            :mod (w3 / web)))
```

Status of AMR

- AMR currently lacks
  - Multilingual consideration
  - Quantifier scope
  - Co-references across sentences
  - Grammatical number, tense, aspect, quotation marks
  - Many noun-noun or noun-adjective relations
  - Many detailed frames, e.g. Earthquake (with roles for magnitude, epicenter, casualties, etc)

[slide from Jonathan Kummerfeld]

AMR Parsing (Wang et al. 2015,16)



(a) Dependency tree (b) AMR graph

Figure 1: Dependency tree and AMR graph for the sentence, "The police want to arrest Micheal Karras in Singapore."

AMR Parsing (Wang et al. 2015,16)

Action	Current state ⇒ Result state	Assign labels	Precondition
NEXT EDGE-l_r	$(\sigma_0\|\sigma', \beta_0\|\beta', G) \Rightarrow (\sigma_0\|\sigma', \beta', G')$	$\delta[(\sigma_0, \beta_0) \rightarrow l_r]$	β is not empty
SWAP-l_r	$(\sigma_0\|\sigma', \beta_0\|\beta', G) \Rightarrow (\sigma_0\|\beta_0\|\sigma', \beta', G')$	$\delta[(\beta_0, \sigma_0) \rightarrow l_r]$	
REATTACH$_k$-l_r	$(\sigma_0\|\sigma', \beta_0\|\beta', G) \Rightarrow (\sigma_0\|\sigma', \beta', G')$	$\delta[(k, \beta_0) \rightarrow l_r]$	
REPLACE HEAD	$(\sigma_0\|\sigma', \beta_0\|\beta', G) \Rightarrow (\beta_0\|\sigma', \beta = CH(\beta_0, G'), G')$	NONE	
REENTRANCE$_k$-l_r	$(\sigma_0\|\sigma', \beta_0\|\beta', G) \Rightarrow (\sigma_0\|\sigma', \beta_0\|\beta', G')$	$\delta[(k, \beta_0) \rightarrow l_r]$	
MERGE	$(\sigma_0\|\sigma', \beta_0\|\beta', G) \Rightarrow (\tilde{\sigma}\|\sigma', \beta', G')$	NONE	
NEXT NODE-l_c	$(\sigma_0\|\sigma_1\|\sigma', [], G) \Rightarrow (\sigma_1\|\sigma', \beta = CH(\sigma_1, G'), G')$	$\gamma[\sigma_0 \rightarrow l_c]$	β is empty
DELETE NODE	$(\sigma_0\|\sigma_1\|\sigma', [], G) \Rightarrow (\sigma_1\|\sigma', \beta = CH(\sigma_1, G'), G')$	NONE	

Table 1: Transitions designed in our parser. $CH(x, y)$ means getting all node x's children in graph y.

AMR Parsing (Wang et al. 2015,16)

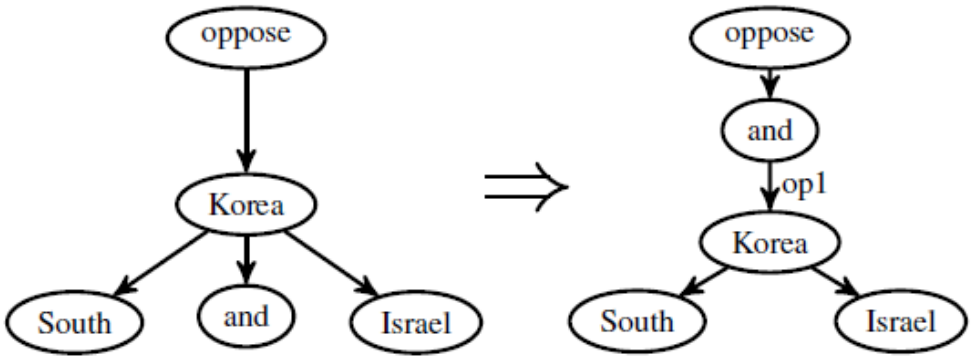


Figure 4: SWAP action

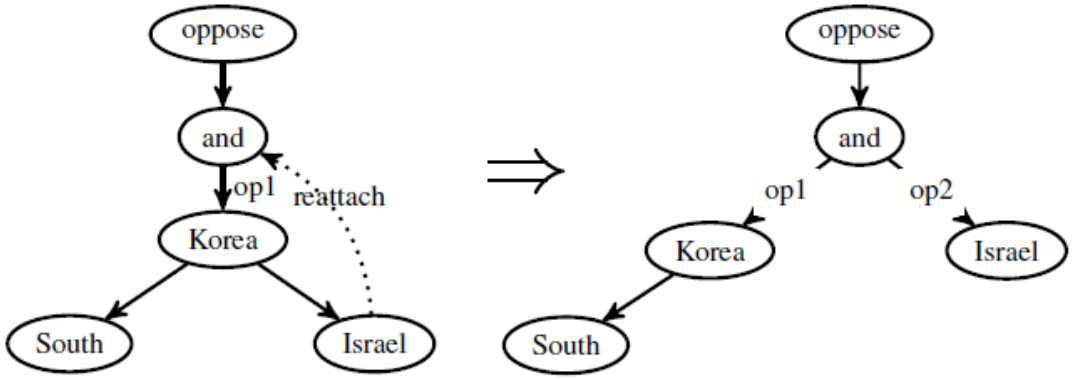


Figure 5: REATTACH action

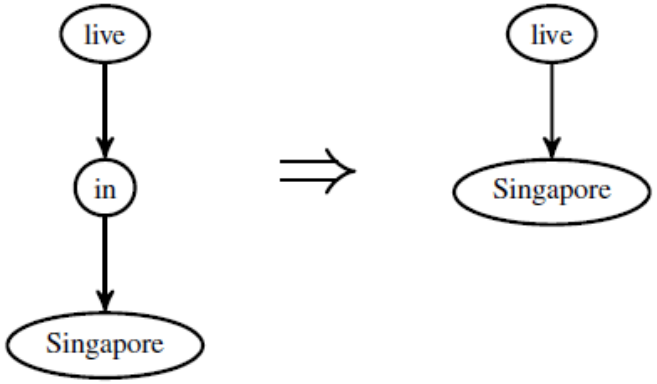


Figure 6: REPLACE-HEAD action

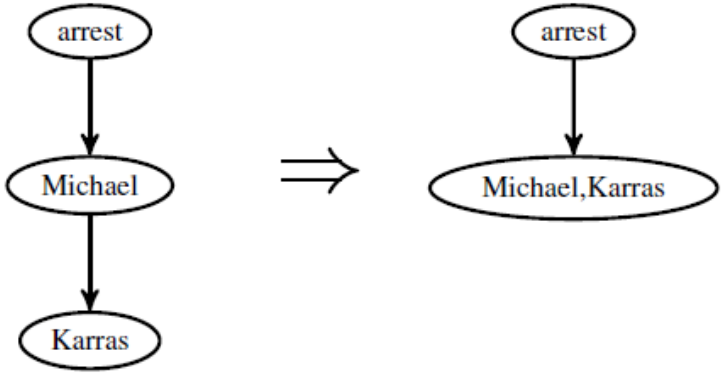
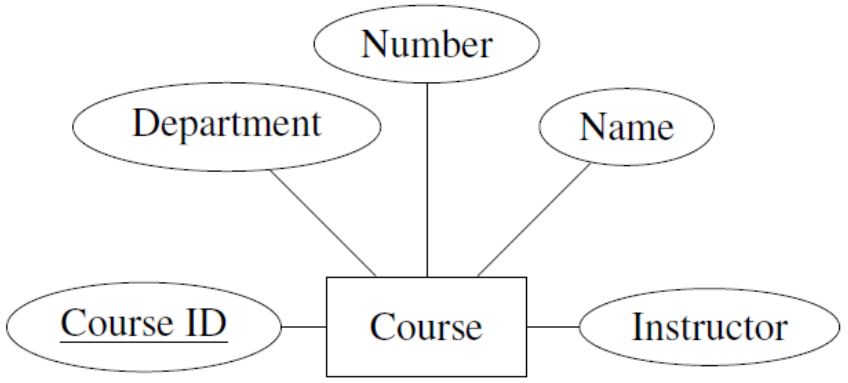


Figure 8: MERGE action

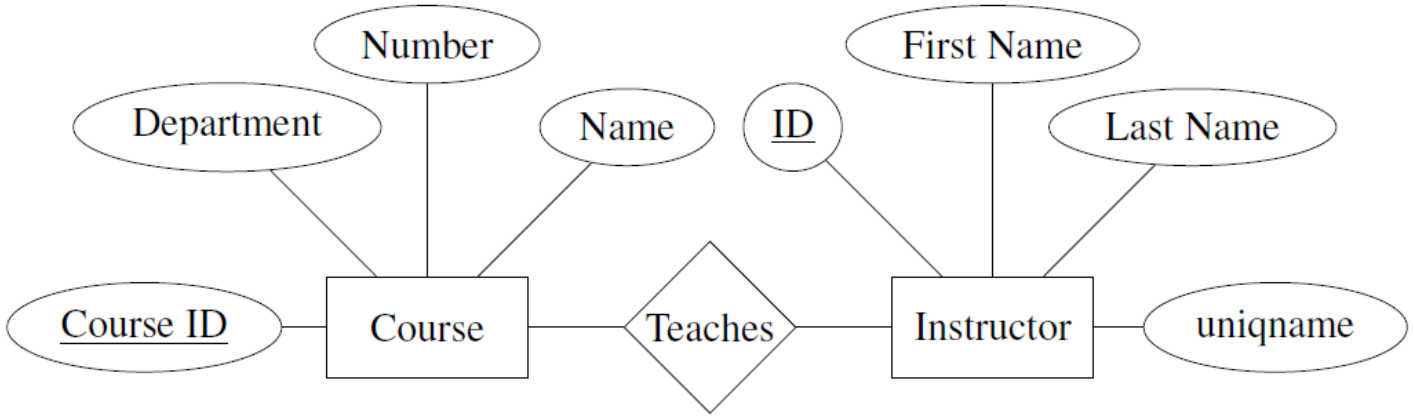
Introduction to NLP

Natural Language to SQL

NL to SQL



(a) Single Table Schema

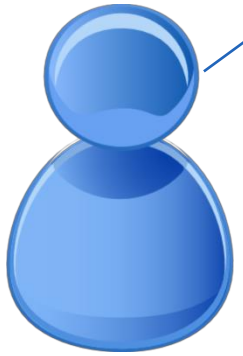


(b) Multi-Table Schema

Figure 1: Two possible schemas for a database that could answer, "Who teaches Discrete Mathematics?"

[Finegan-Dollak et al]

Example: Text-to-SQL



"Who teaches NLP?"

```
SELECT I.NAME
FROM INSTRUCTOR AS I,
OFFERING_INSTRUCTOR AS OI,
COURSE_OFFERING AS O, SEMESTER AS S,
COURSE AS C
WHERE OI.INSTRUCTOR_ID=I.INSTRUCTOR_ID
AND O.OFFERING_ID=OI.OFFERING_ID
AND O.SEMESTER=S.SEMESTER_ID
AND O.COURSE_ID=C.COURSE_ID
AND C.NAME="NLP"
AND S.YEAR=2016 AND S.SEMESTER="FA"
```

All About SQL

Course

Course ID	Dept.	Number	Name	Credits
1	EECS	203	Discrete Math	4
2	LING	137	Epic Grammar Fails	3
3	EECS	595	NLP	4

Instructor

Instructor ID	First Name	Last Name
1	Dragomir	Radev
2	Walter	Lasecki
3	Ezra	Keshet
4	Rada	Mihalcea

Course Offering

Course ID	Instructor ID	Year	Semester
3	1	2016	Fall
3	4	2017	Fall
2	3	2018	Winter

How many credits is NLP?

```
SELECT C.CREDITS
FROM COURSE AS C
WHERE C.NAME = "NLP";
```

All About SQL

Course

Course ID	Dept.	Number	Name	Credits
1	EECS	203	Discrete Math	4
2	LING	137	Epic Grammar Fails	3
3	EECS	595	NLP	4

Instructor

Instructor ID	First Name	Last Name
1	Dragomir	Radev
2	Walter	Lasecki
3	Ezra	Keshet
4	Rada	Mihalcea

Course Offering

Course ID	Instructor ID	Year	Semester
3	1	2016	Fall
3	4	2017	Fall
2	3	2018	Winter

Who teaches NLP?

```
SELECT I.FIRST_NAME, I.LAST_NAME
FROM INSTRUCTOR AS I,
   COURSE AS C,
   COURSE_OFFERING AS CO
WHERE C.NAME = "NLP"
AND C.COURSE_ID = CO.COURSE_ID
AND CO.INSTRUCTOR_ID =
   I.INSTRUCTOR_ID;
```

All About SQL

Course

Course ID	Dept.	Number	Name	Credits
1	EECS	203	Discrete Math	4
2	LING	137	Epic Grammar Fails	3
3	EECS	595	NLP	4

Instructor

Instructor ID	First Name	Last Name
1	Dragomir	Radev
2	Walter	Lasecki
3	Ezra	Keshet
4	Rada	Mihalcea

Course Offering

Course ID	Instructor ID	Year	Semester
3	1	2016	Fall
3	4	2017	Fall
2	3	2018	Winter

What course is worth the most credits?

```
SELECT C1.NAME
FROM COURSE AS C1
WHERE C1.CREDITS =
    (SELECT MAX C2.CREDITS
    FROM COURSE AS C2);
```

More Complicated SQL

Which countries in Europe have at least 3 car manufacturers?

```
SELECT T1.country_name
FROM countries AS T1 JOIN continents
AS T2 ON T1.continent = T2.cont_id
JOIN car_makers AS T3 ON
T1.country_id = T3.country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) >= 3
```

What is the average life expectancy in the countries where English is not the official language?

```
SELECT AVG(life_expectancy)
FROM country
WHERE name NOT IN
    (SELECT T1.name
     FROM country AS T1 JOIN
     country_language AS T2
     ON T1.code = T2.country_code
     WHERE T2.language = "English"
       AND T2.is_official = "T")
```

Seq2SQL datasets are scarce

- Compared to other large datasets such as ImageNet for object recognition, building a decent seq2SQL dataset is even more time-consuming
- Hard to find many databases with multiple tables online
- Annotation requires very specific knowledge in databases

Traditional Seq2SQL datasets

Traditional 9 seq2SQL datasets: ATIS, Geo, Scholar, etc. + Advising

- Pros
  - SQL queries cover complex SQL structures and components
- Cons
  - The number of labeled queries is small (< 500)
    - Paraphrase about 4-10 natural language questions for each SQL query.
    - The total # of question-SQL pairs: ~500 -> ~5,000
  - Each of datasets contains SQL queries only to a single database

GEO Query

which state has the most rivers running through it?

```
argmax $0
 (state:t $0)
 (count $1 (and
            (river:t $1)
            (loc:t $1 $0)))
```

Lambda Calculus Logical Form

ATIS

Show me flights from Pittsburgh to Seattle

```
lambda $0 e
  (and (flight $0)
  (from $0 pittsburgh:ci)
  (to $0 seattle:ci))
```

Lambda Calculus Logical Form

JOBS

what microsoft jobs do not require a bscs?

```
answer(
   company(J,'microsoft'),
   job(J),
   not((req deg(J,'bscs'))))
```

Prolog-style Program

Credit to Pengcheng Yin

ATIS (Price, 1990; Dahl et al., 1994) User questions for a flight-booking task, manually annotated. We use the modified SQL from Iyer et al. (2017), which follows the data split from the logical form version (Zettlemoyer and Collins, 2007).

GeoQuery (Zelle and Mooney, 1996) User questions about US geography, manually annotated with Prolog. We use the SQL version (Popescu et al., 2003; Giordani and Moschitti, 2012; Iyer et al., 2017), which follows the logical form data split (Zettlemoyer and Collins, 2005).

Restaurants (Tang and Mooney, 2000; Popescu et al., 2003) User questions about restaurants, their food types, and locations.

Scholar (Iyer et al., 2017) User questions about academic publications, with automatically generated SQL that was checked by asking the user if the output was correct.

Academic (Li and Jagadish, 2014) Questions about the Microsoft Academic Search (MAS) database, derived by enumerating every logical query that could be expressed using the search page of the MAS website and writing sentences to match them. The domain is similar to that of Scholar, but their schemas differ.

Yelp and IMDB (Yaghmazadeh et al., 2017) Questions about the Yelp website and the Internet Movie Database, collected from colleagues of the authors who knew the type of information in each database, but not their schemas.

WikiSQL (Zhong et al., 2017) A large collection of automatically generated questions about individual tables from Wikipedia, paraphrased by crowd workers to be fluent English.

Advising (This Work) Our dataset of questions over a database of course information at the University of Michigan, but with fictional student records. Some questions were collected from the EECS department Facebook page and others were written by CS students with knowledge of the database who were instructed to write questions they might ask in an academic advising appointment.

Dialog2SQL Data Creation

Our Complex and Cross-Domain Text-to-SQL Dataset: Spider

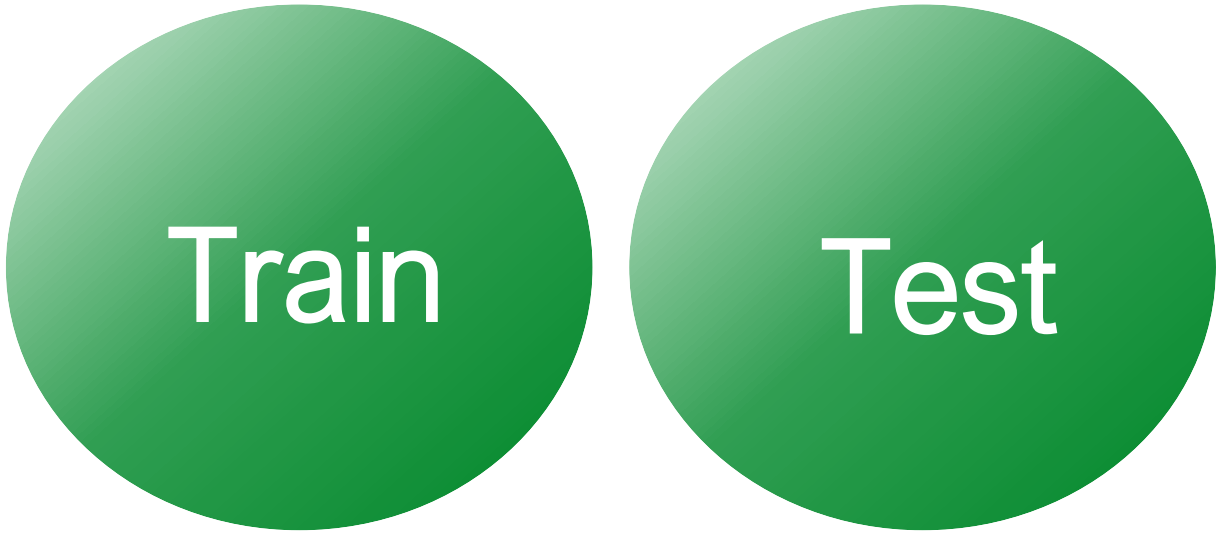
Dataset	# Q	# SQL	# DB	# Table /DB	ORDER BY	GROUP BY	NESTED	HAVING	LIMIT
ATIS	5,280	947	1	32	0	5	315	0	0
GeoQuery	877	247	1	6	20	46	167	9	20
Scholar	817	193	1	7	75	100	7	20	1
Academic	196	185	1	15	23	40	7	18	23
IMDB	131	89	1	16	10	6	1	0	10
Yelp	128	110	1	7	18	21	0	4	18
Advising	3,898	208	1	10	15	9	22	0	11
Restaurants	378	378	1	3	0	0	4	0	0
WikiSQL	80,654	77,840	26,521	1	0	0	0	0	0
Spider (original)	10,181	5,693	200	5.1	1335	1491	844	388	903

Figure: Comparisons of text-to-SQL datasets

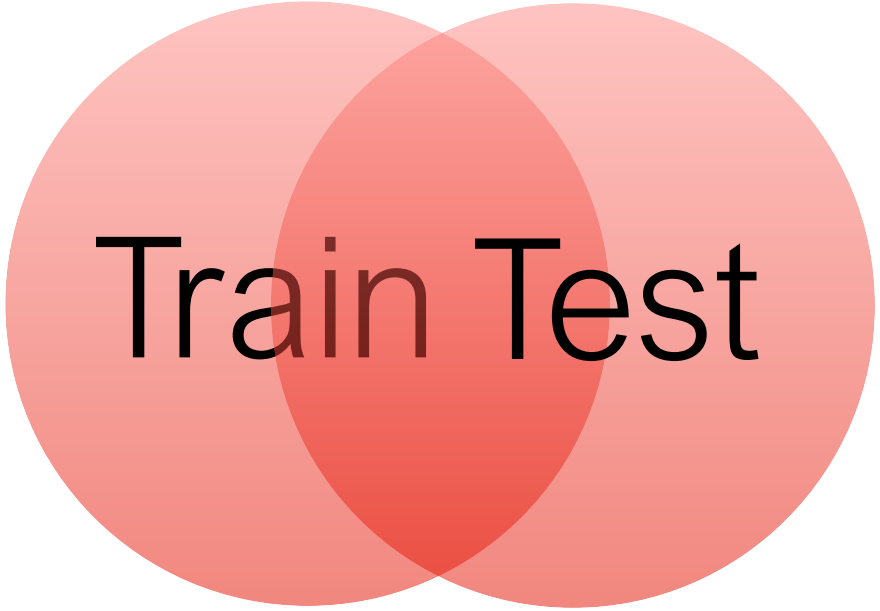
Standard Practice in ML

- Split dataset into train set, test set, optional development (dev) set.
- No training example can also appear in test set.

Good Split



Bad Split



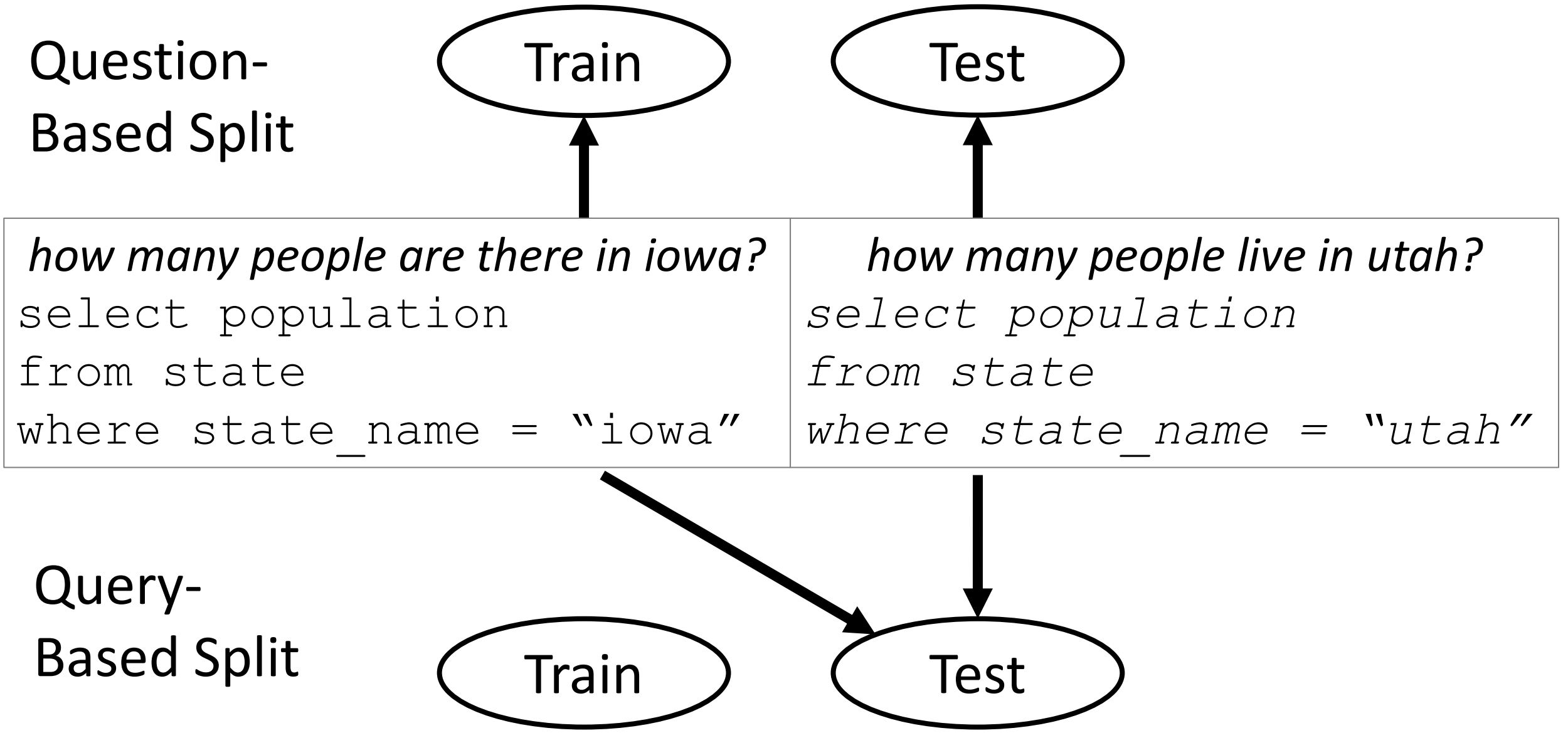
How Do We Define an Example?

how many people are there in iowa?

```
select population
from state
where state_name = “iowa”
```

how many people live in utah?

```
select population
from state
where state_name = “utah”
```


Question-
Based Split
Train
Test
how many people are there in iowa?
select population
from state
where state_name = "iowa"
how many people live in utah?
select population
from state
where state_name = "utah"
Query-
Based Split
Train
Test

Seq2SQL data – WikiSQL (Salesforce)

- The first realistic seq2SQL task definition on the top of WikiSQL makes it the most popular seq2SQL dataset
- Databases in the test set do not appear in the train/dev set, which requires model to generalize to new databases
- https://github.com/salesforce/WikiSQL

WikiSQL Animated GIF

https://beta.techcrunch.com/wp-content/uploads/2017/08/unnamed2.gif

WikiSQL Example

- https://github.com/salesforce/WikiSQL

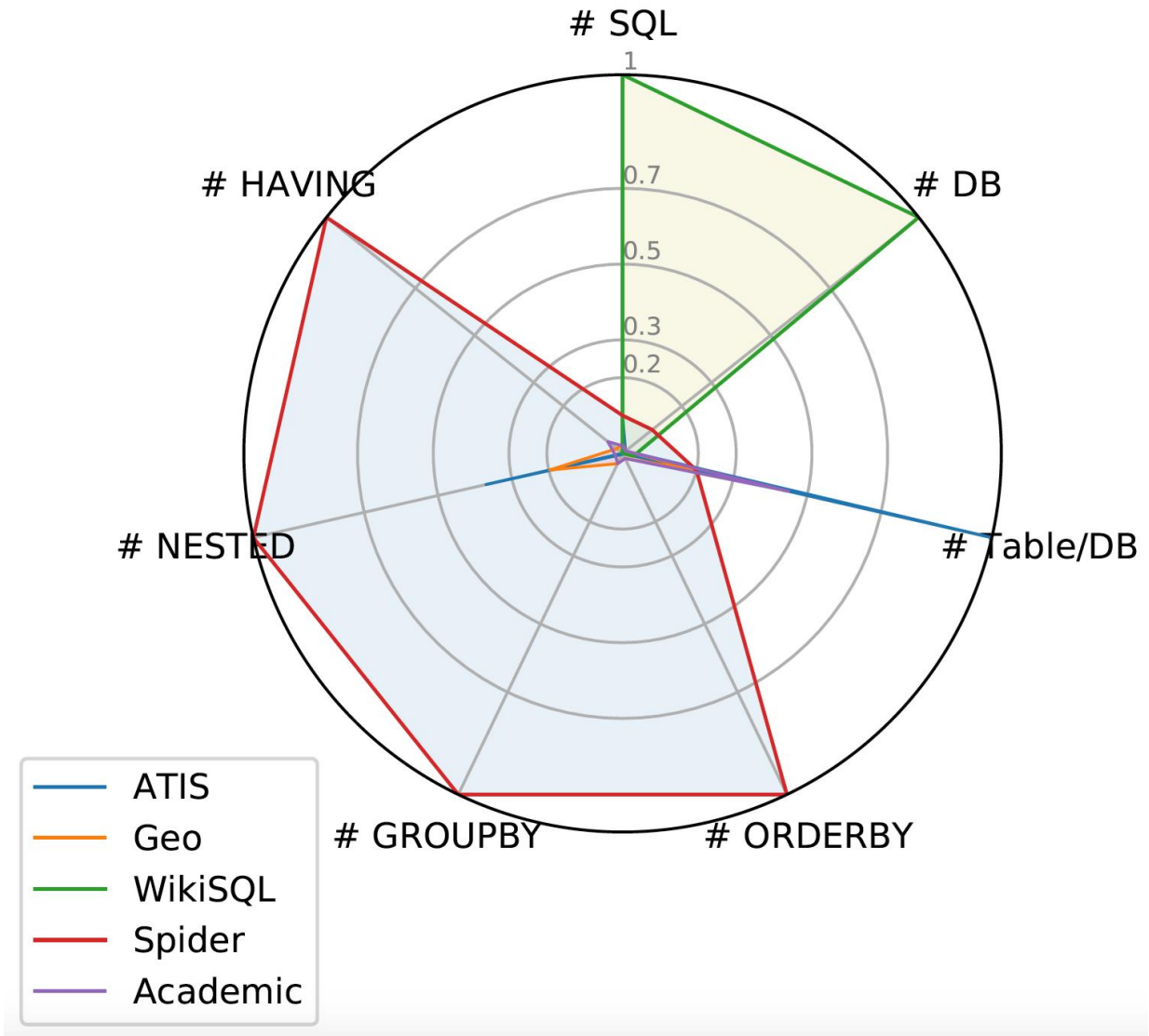
Seq2SQL data – WikiSQL (Salesforce paper)

- WikiSQL - Pros
  - The number of SQL queries and databases is huge (>20,000)
  - Databases in the test set do not appear in the train/dev set, which requires model to generalize to new databases
- WikiSQL - Cons
  - SQL queries are generated by templates and paraphrased by Turkers
  - All databases have only one table - not a full relational database
  - SQL only contains SELECT and WHERE. No GROUP BY/Nested queries etc.

Seq2SQL data - Spider

- WikiSQL is great. But it has limited SQL coverage and a very simple schema, which makes the task simple and less interesting

[Yu et al. 2018]



Seq2SQL data - Yale Spider

- SQL labels cover almost all important SQL components
- Each database has multiple tables and several foreign keys
- It is currently the only large-scale complex and cross-domain semantic parsing and text-to-SQL dataset!
- Check it out!!!
- Our Blog
  - Project Page: https://yale-lily.github.io/spider
  - Github Page: https://github.com/taoyds/spider

Query Difficulty

Easy

What is the number of cars with more than 4 cylinders'

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Meidum

For each stadium, how many concerts are there?

```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2
ON T1.stadium_id = T2.stadium_id
GROUP BY T1.stadium_id
```

Hard

Which countries in Europe have at least 3 car manufacturers?

```
SELECT T1.country_name
FROM countries AS T1 JOIN continents
AS T2 ON T1.continent = T2.cont_id
JOIN car_makers AS T3 ON
T1.country_id = T3.country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) >= 3
```

Extra Hard

What is the average life expectancy in the countries where English is not the official language?

```
SELECT AVG(life_expectancy)
FROM country
WHERE name NOT IN
   (SELECT T1.name
    FROM country AS T1 JOIN
    country_language AS T2
    ON T1.code = T2.country_code
    WHERE T2.language = "English"
      AND T2.is_official = "T")
```