COMP 786 (Fall 2020)
Natural Language Processing

Week 6: Semantic Parsing (Semantic Role Labeling, Lambda-Calculus, CCG, DCS, etc.)

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Mohit Bansal

(various slides adapted/borrowed from courses by Dan Klein, Artzi-FitzGerald-Zettlemoyer CCG tutorial, JurafskyMartin-SLP3, others)

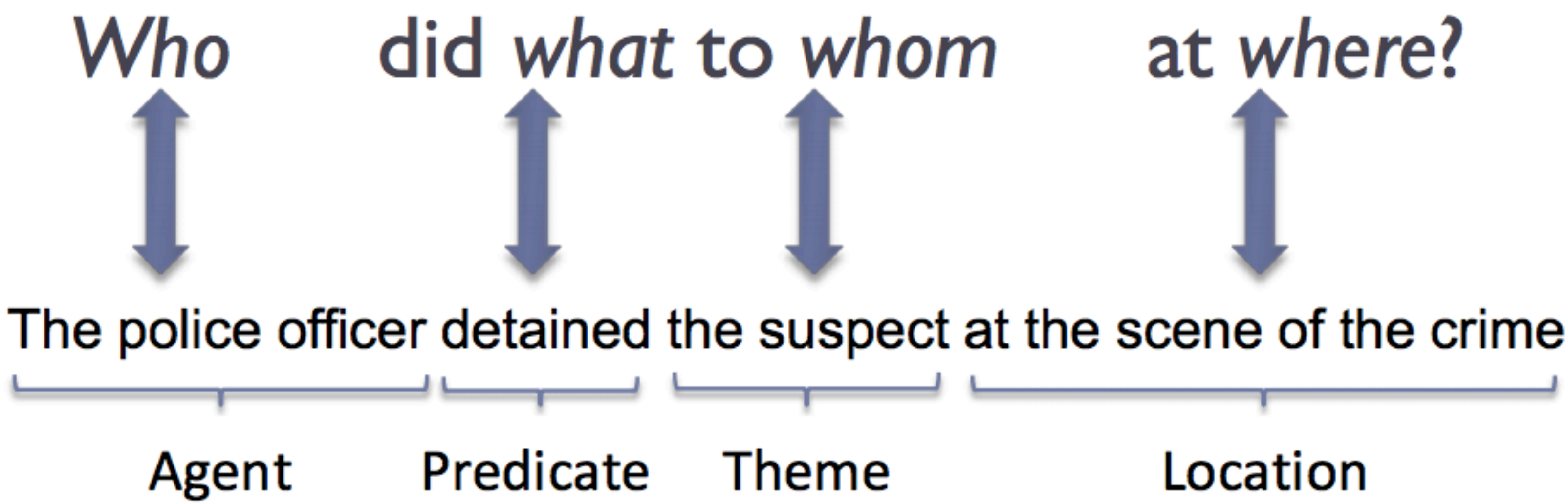
Announcements

- Coding-HW1 (on word vector training+evaluation_ +visualization) has been released (and details emailed) last week – due Sep23 midnight! Stay after class today to get more info/demo.

- Midterm project presentation coming up in 2 weeks (look out for details in email).

Semantic Role Labeling

Semantic Role Labeling (SRL)



[SLP3 book]

Semantic Role Labeling (SRL)

- Role-based relations for the different clauses in the sentence:

[$_{Judge}$ She] **blames** [$_{Evaluee}$ the Government] [$_{Reason}$ for failing to do enough to help] .

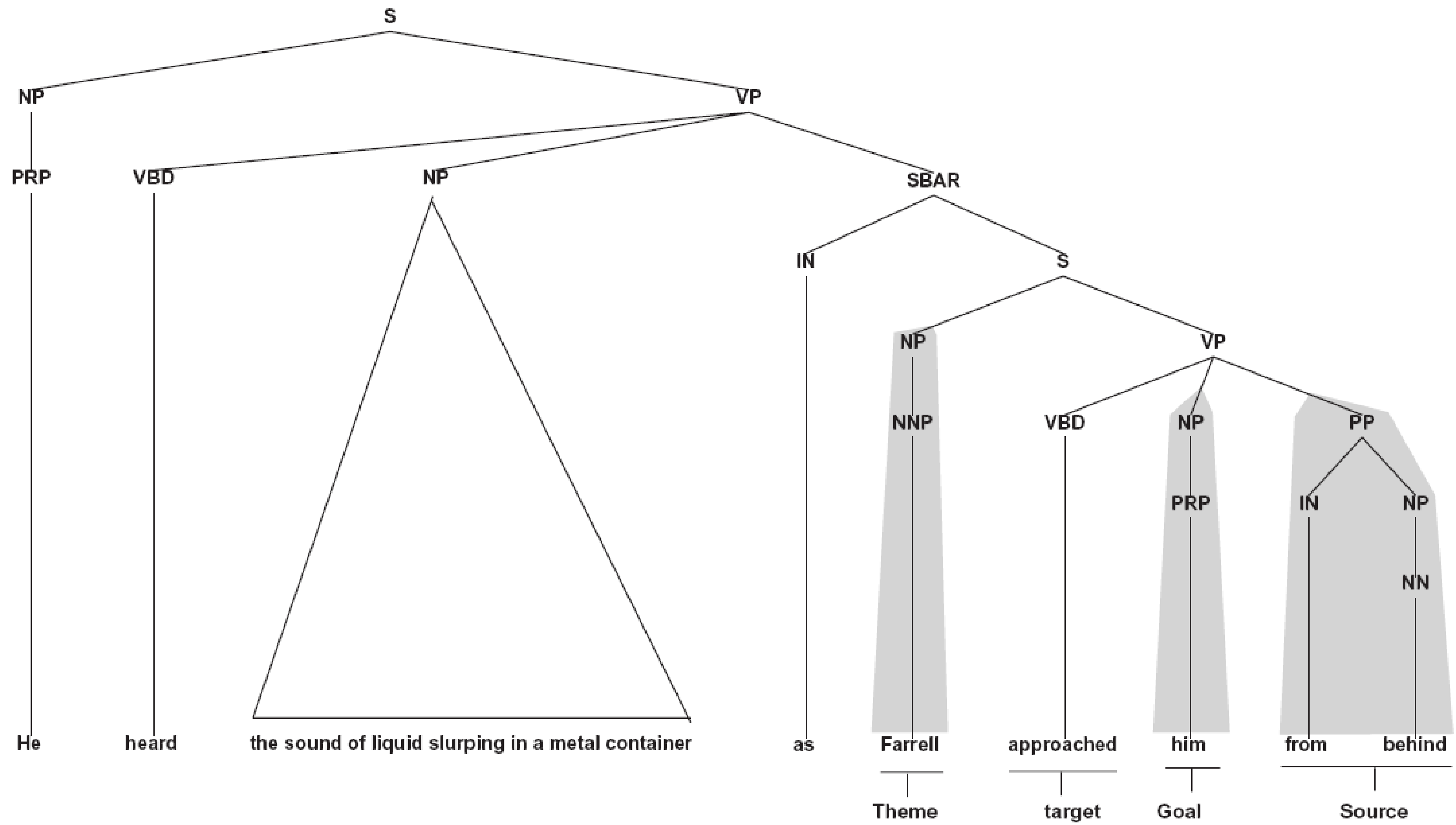
Holman would characterise this as **blaming** [$_{Evaluee}$ the poor] .

The letter quotes Black as saying that [$_{Judge}$ white and Navajo ranchers] misrepresent their livestock losses and **blame** [$_{Reason}$ everything] [$_{Evaluee}$ on coyotes] .

- More semantic relations (e.g., agent, reason, message) than just subject/object style syntactic roles

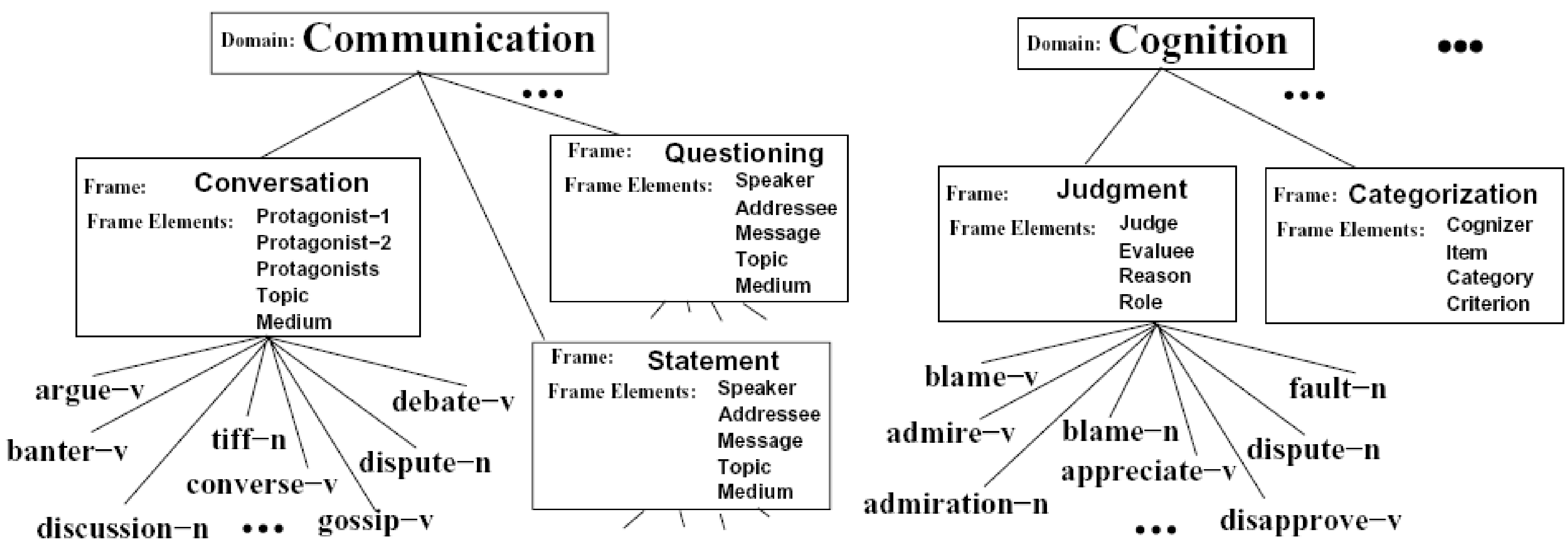
- Typical traditional pipelines involves POS-tagging and parsing, and then features extracted on those (plus NER, etc.)…but then several errors caused by wrong parse!

Semantic Role Labeling (SRL)



PropBank vs. FrameNet

- PropBank has each verb get its own roles, whereas FrameNet shares roles between verbs (e.g., argue and banter in figure below)
- PropBank more convenient w.r.t. being layered over Treebank parses (and hence more coverage)



PropBank Roles

- Based on Dowty, 1991: roles are verb-sense specific in PropBank (role definitions depend on specific verb and relation to other roles)

- Each verb sense has numbered arguments e.g., ARG-0, ARG-1, etc.
  - ARG-0 is usually PROTO-AGENT
  - ARG-1 is usually PROTO-PATIENT
  - ARG-2 is usually benefactive, instrument, attribute
  - ARG-3 is usually start point, benefactive, instrument, attribute
  - ARG-4 is usually end point (e.g., for move or push style verbs)

  (ARG-2,3,4 onwards not very consistent and highly depend on specific verb and its sense in the sentence, hence labeling of PropBank is tricky)

PropBank Example 1

fall.01 sense: move downward

roles:	Arg1:	thing falling
	Arg2:	extent, distance fallen
	Arg3:	start point
	Arg4:	end point

Sales fell to $251.2 million from $278.7 million.

arg1:	Sales
rel:	fell
arg4:	to $251.2 million
arg3:	from $278.7 million

PropBank Example 2

rotate.02		sense: shift from one thing to another	
	roles:	Arg0:	causer of shift
		Arg1:	thing being changed
		Arg2:	old thing
		Arg3:	new thing

Many of Wednesday's winners were losers yesterday as investors quickly took profits and rotated their buying to other issues, traders said. (wsj_1723)

arg0:	investors
rel:	rotated
arg1:	their buying
arg3:	to other issues

PropBank Example 3

aim.01 sense: intend, plan
roles: Arg0: aimer, planner
Arg1: plan, intent

The Central Council of Church Bell Ringers aims *trace* to improve relations with vicars. (wsj_0089)

arg0: The Central Council of Church Bell Ringers
rel: aims
arg1: *trace* to improve relations with vicars

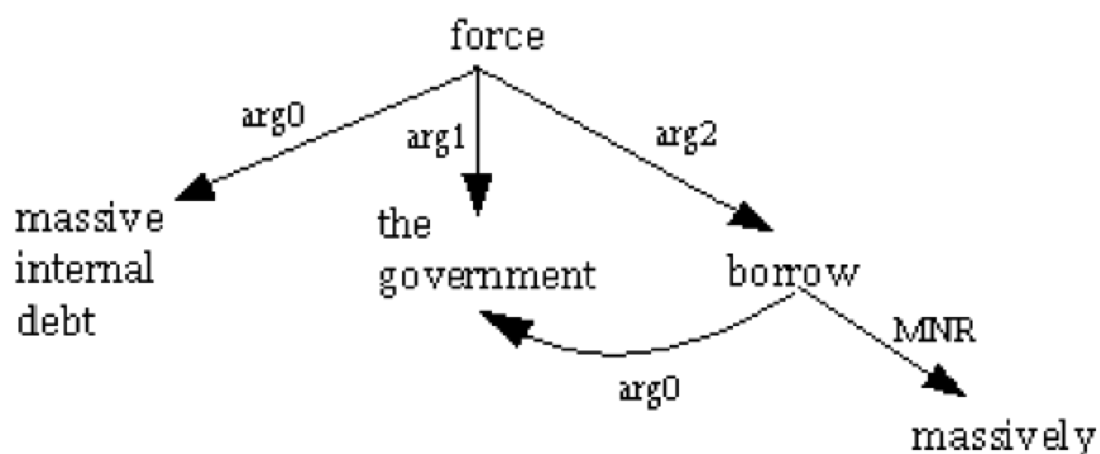
aim.02 sense: point (weapon) at
roles: Arg0: aimer
Arg1: weapon, etc.
Arg2: target

Banks have been aiming packages at the elderly.

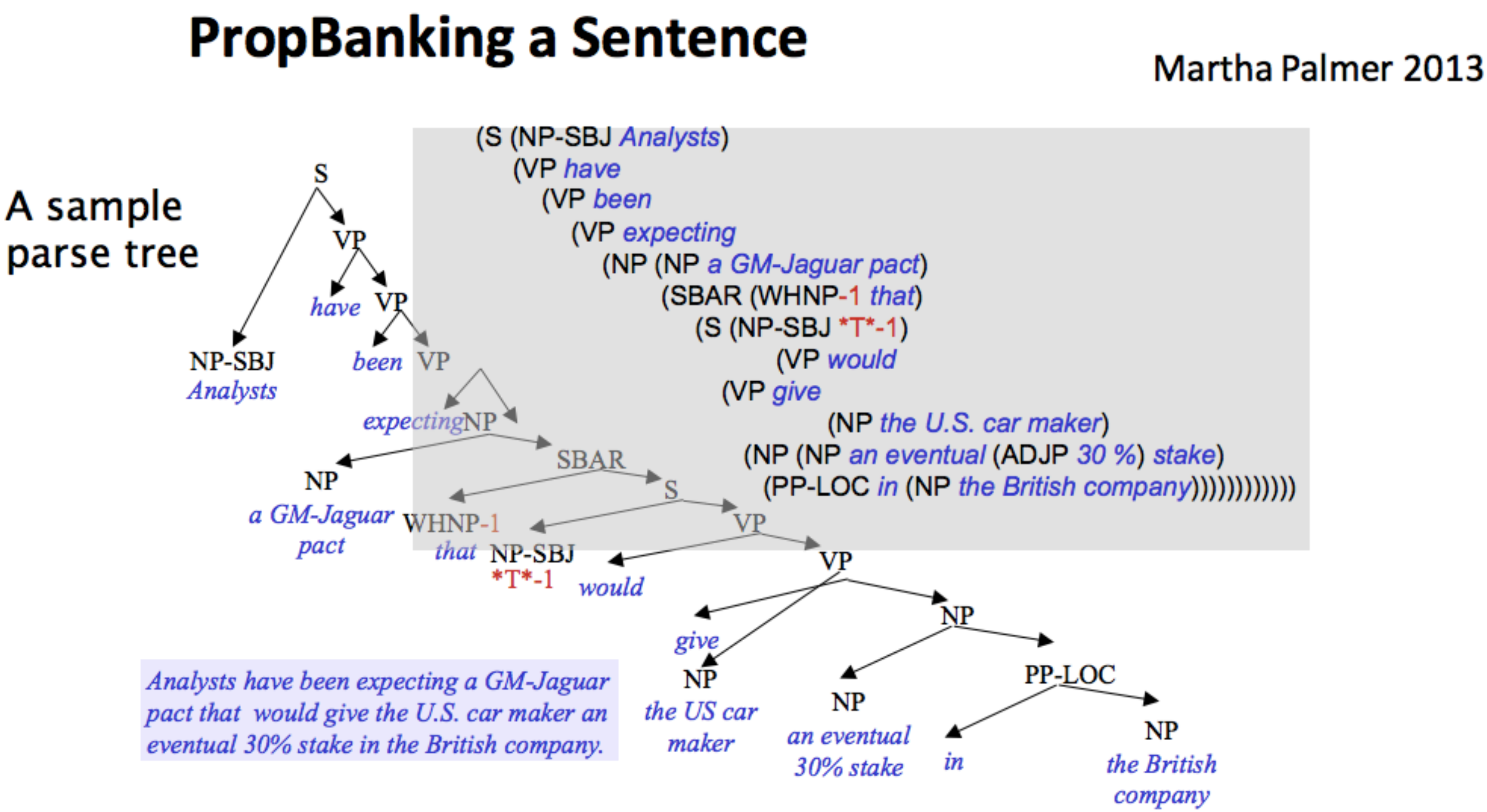
arg0: Banks
rel: aiming
arg1: packages
arg2: at the elderly

Shared Arguments

```
(NP-SBJ (JJ massive) (JJ internal) (NN debt) )
  (VP (VBZ has)
   (VP (VBN forced)
    (S
     (NP-SBJ-1 (DT the) (NN government) )
     (VP
      (VP (TO to)
       (VP (VB borrow)
        (ADVP-MNR (RB massively) )...
```



PropBank Parsing

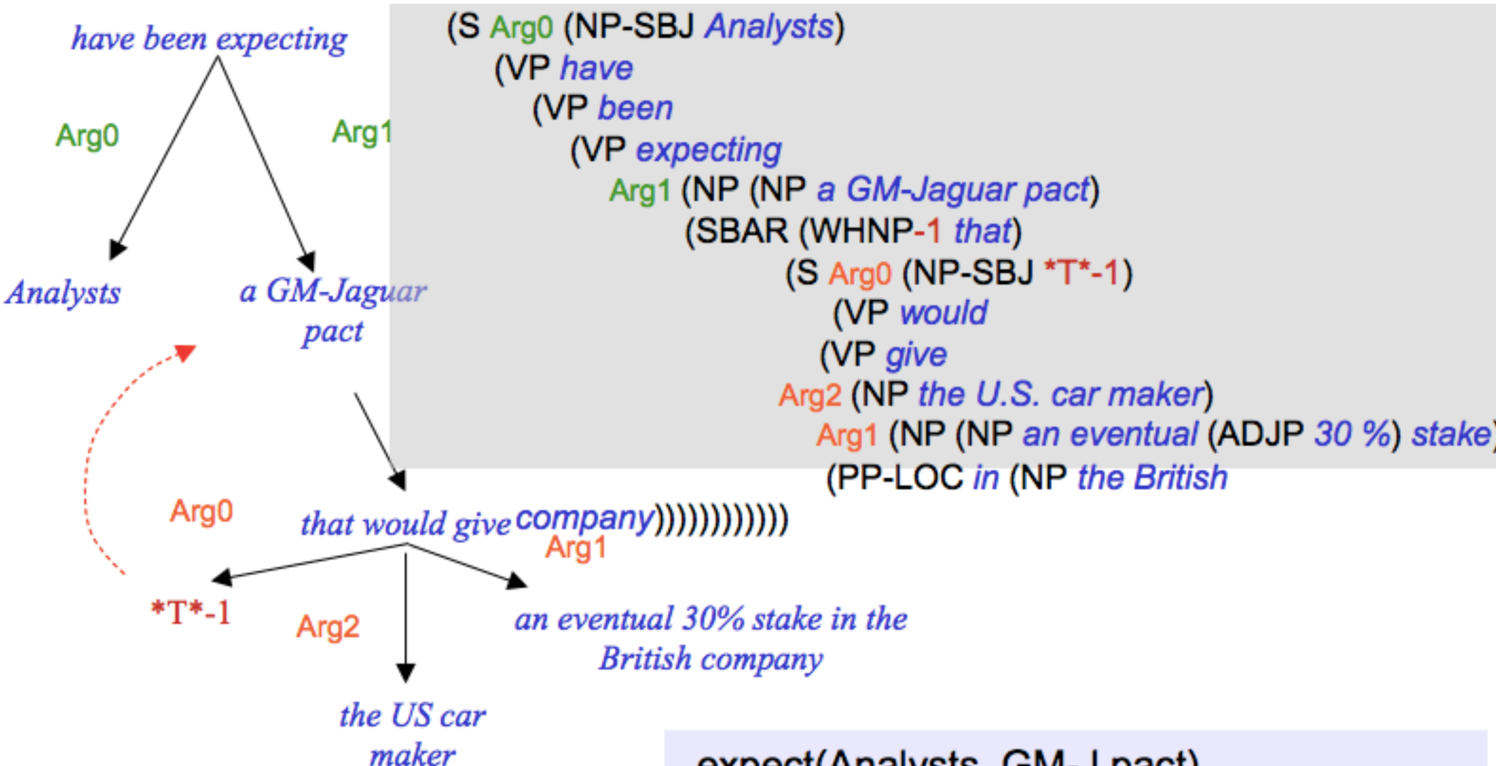


[SLP3 book]

PropBank Parsing

The same parse tree PropBanked

Martha Palmer 2013



expect(Analysts, GM-J pact)
give(GM-J pact, US car maker, 30% stake)

[SLP3 book]

Data Size

Annotated PropBank Data

- Penn English TreeBank,
  OntoNotes 5.0.
  - Total ~2 million words
- Penn Chinese TreeBank
- Hindi/Urdu PropBank
- Arabic PropBank

2013 Verb Frames Coverage
Count of word sense (lexical units)

Language	*Final Count*
English	**10,615***
Chinese	**24, 642**
Arabic	**7,015**

From Martha Palmer 2013 Tutorial

[SLP3 book]

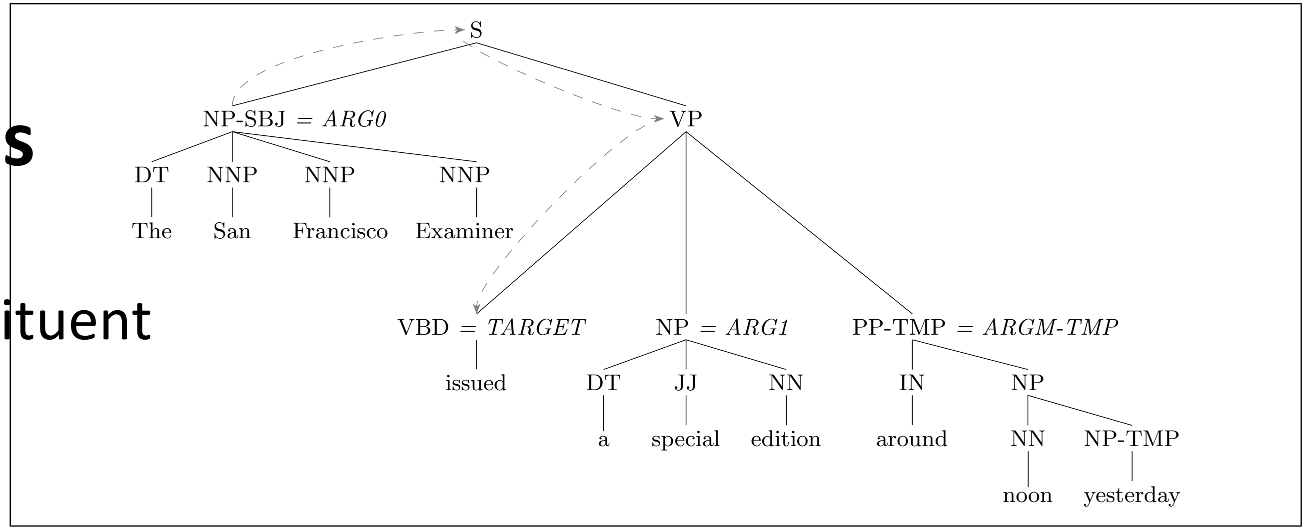
Simple SRL Algo

function SEMANTICROLELABEL(*words*) **returns** labeled tree

parse $\leftarrow$ PARSE(*words*)
for each *predicate* **in** *parse* **do**
  for each *node* **in** *parse* **do**
    featurevector $\leftarrow$ EXTRACTFEATURES(*node*, *predicate*, *parse*)
    CLASSIFYNODE(*node*, *featurevector*, *parse*)

SRL Features

Features



Headword of constituent

Examiner

Headword POS

NNP

Voice of the clause

Active

Subcategorization of pred

VP -> VBD NP PP

Named Entity type of constit

ORGANIZATION

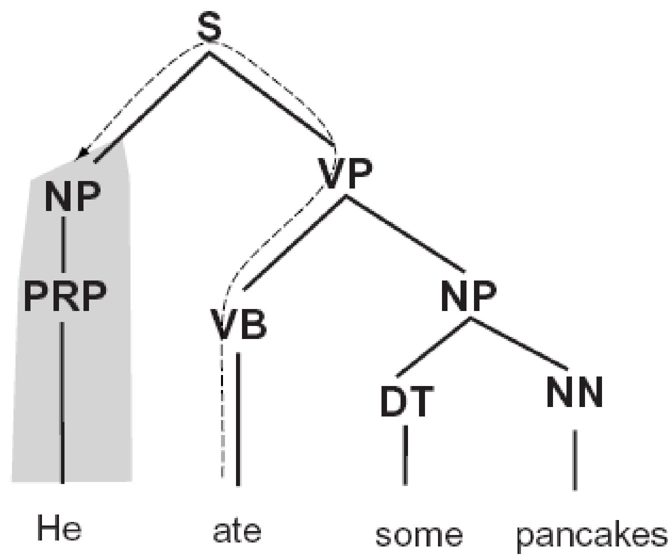
First and last words of constit

The, Examiner

Linear position,clause re: predicate

before

Path-based Features for SRL



Path	*Description*
VB↑VP↓PP	PP argument/adjunct
VB↑VP↑S↓NP	subject
VB↑VP↓NP	object
VB↑VP↑VP↑S↓NP	subject (embedded VP)
VB↑VP↓ADVP	adverbial adjunct
NN↑NP↑NP↓PP	prepositional complement of noun

Some SRL Results

- So major feature categories in traditional feature-based SRL models were:
  - Headword, syntactic type, case, etc. of candidate node/ constituent
  - Linear and tree path from predicate target to node
  - Active vs. passive voice
  - Second order and higher order features

- Accuracy for such feature-based SRL models then highly depends on accuracy of underlying parse tree!
  - So quite high SRL results when using ground-truth parses
  - Much lower results with automatically-predicted parses!

CORE		ARGM	
F1	Acc.	F1	Acc.
92.2	80.7	89.9	71.8

CORE		ARGM	
F1	Acc.	F1	Acc.
84.1	66.5	81.4	55.6

Neural SRL

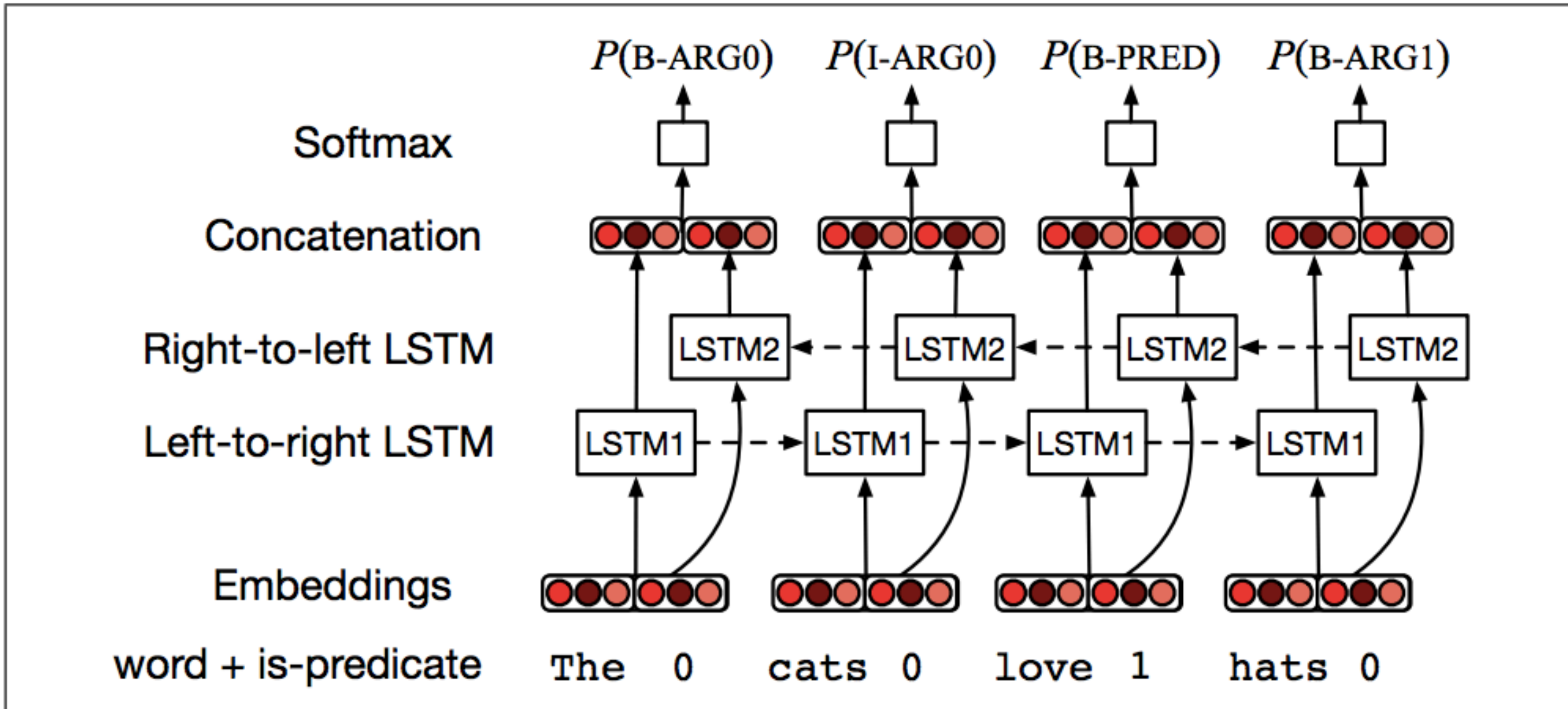
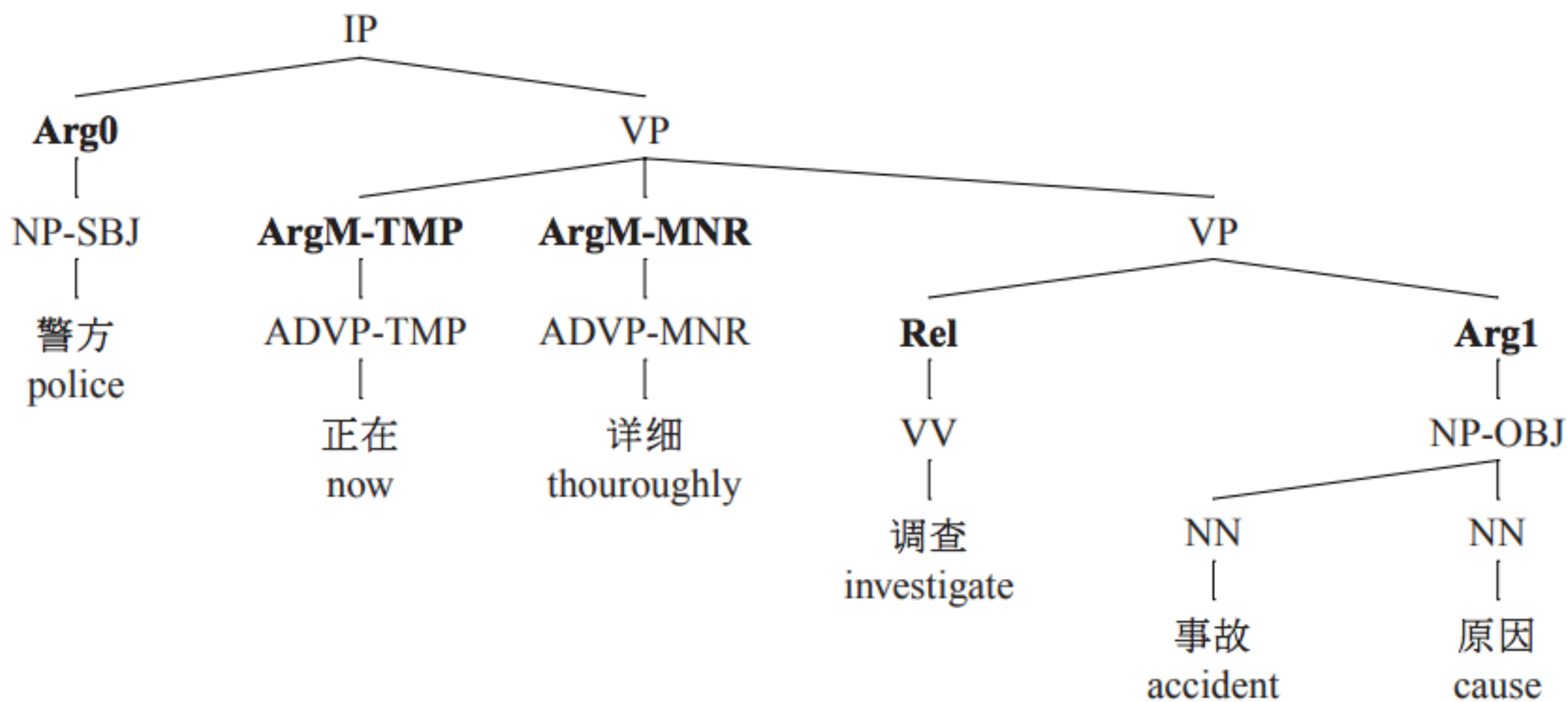


Figure 20.6 A bi-LSTM approach to semantic role labeling. Most actual networks are much deeper than shown in this figure; 3 to 4 bi-LSTM layers (6 to 8 total LSTMs) are common. The input is a concatenation of an embedding for the input word and an embedding of a binary variable which is 1 for the predicate to 0 for all other words. After He et al. (2017).

Semantic Roles Beyond English

Not just English



"The police are thoroughly investigating the cause of the accident."

Semantic Roles Beyond Verbs

Not just verbs: NomBank

Meyers et al. 2004

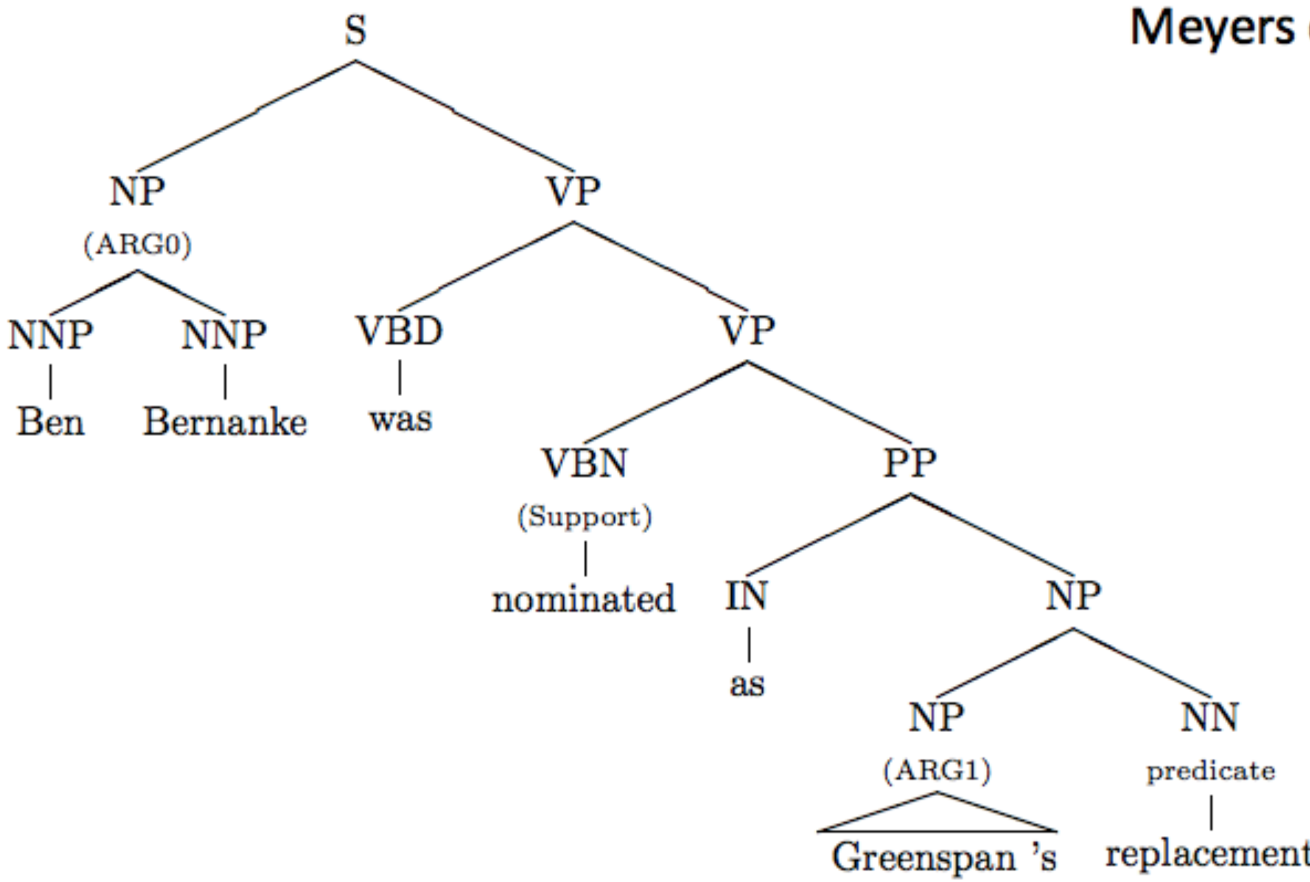


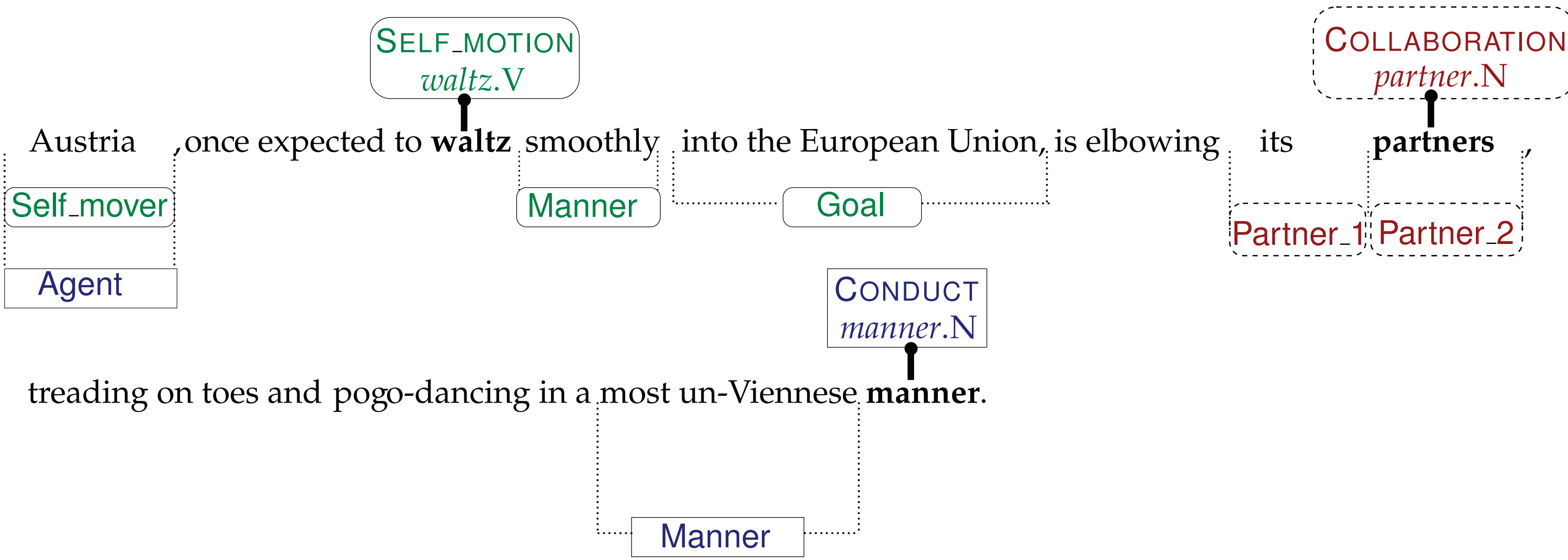
Figure from Jiang and Ng 2006

Semantic Roles Beyond Verbs

Additional Issues for nouns

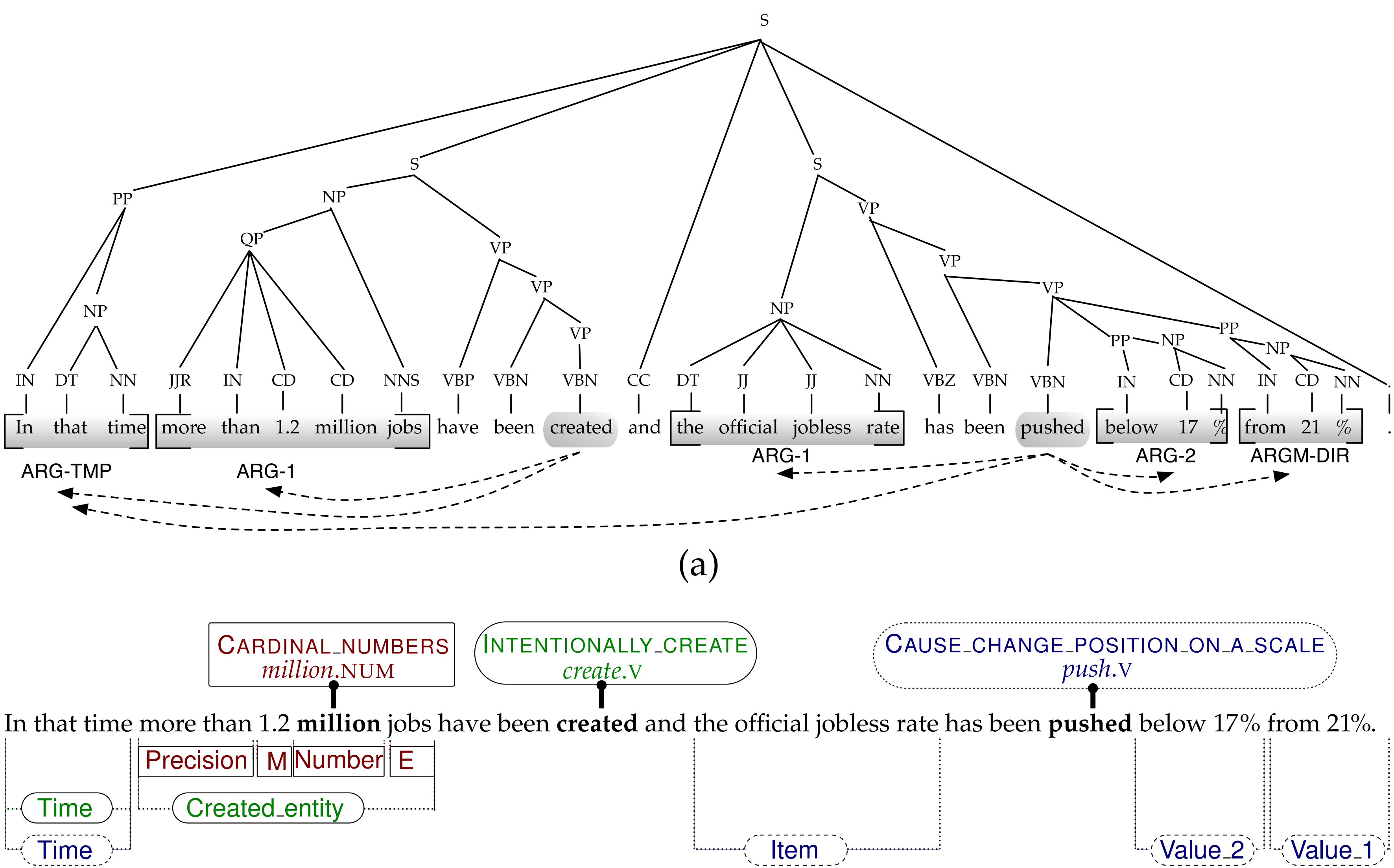
- Features:
  - Nominalization lexicon (employment→ employ)
  - Morphological stem
    - Healthcare, Medicate → care
- Different positions
  - Most arguments of nominal predicates occur inside the NP
  - Others are introduced by support verbs
  - Especially light verbs “X made an argument”, “Y took a nap”

Schematic of Frame Semantics (FrameNet)



[Das et al., 2014]

PropBank vs. FrameNet Representations



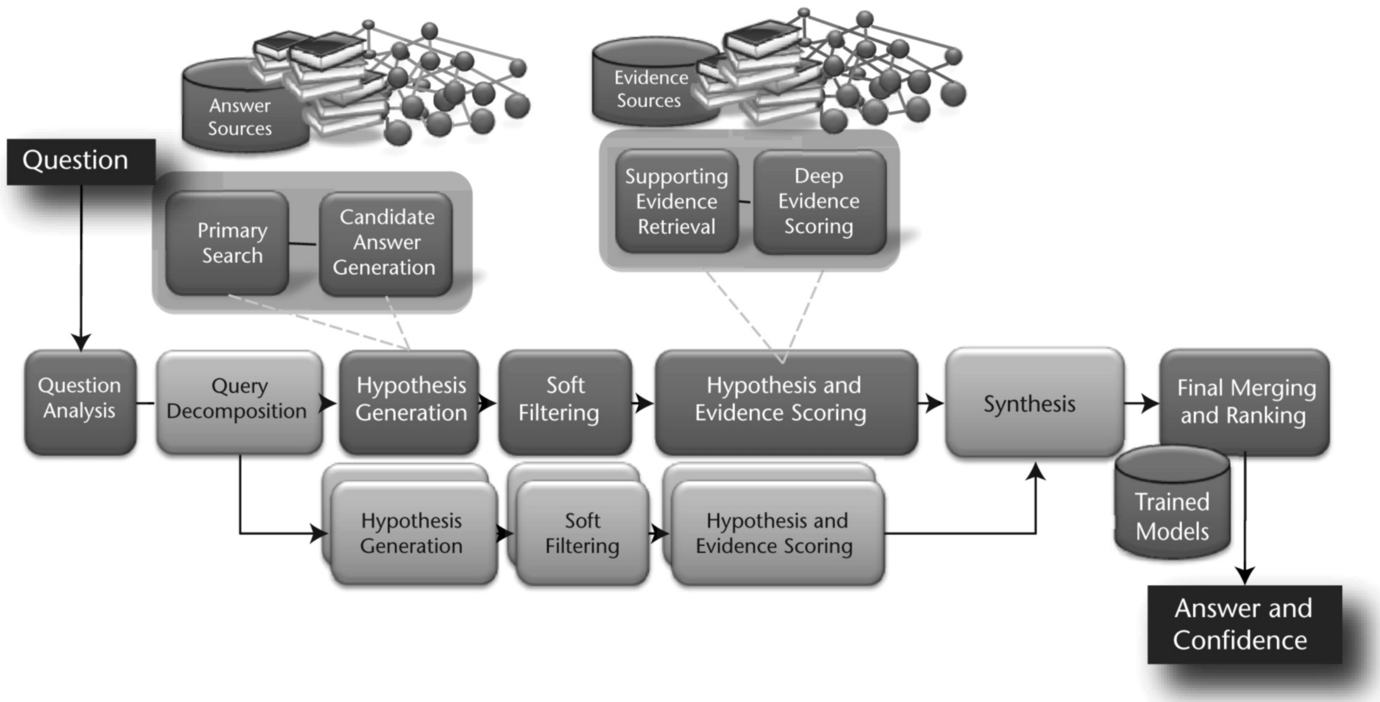
Compositional Semantics

Compositional Semantics I: Logic form

- Logic-form based (lambda calculus), Semantic Parsing
- Useful for Q&A, IE, grounding, comprehension tasks (summarization, reading tasks)
- A lot of focus has been on KB-based Question Answering in this direction (where input-output training data is question-answer pairs, and latent intermediate representation is the question's semantic parse, which is 'executed' against the KB to get the answer)

Question Answering

- Initial approaches to Q&A: pattern matching, pattern learning, query rewriting, information extraction
- Next came a large-scale, open-domain IE system like IBM Watson



[Ferrucci et al., 2010]

Deep Q&A: Semantic Parsing

- Complex, free-form, multi-clause questions

Google | what countries border the black sea

Web Maps Images News Shopping More Search tools

About 2,560,000 results (0.57 seconds)

The Black Sea is an inland sea located between far-southeastern Europe and the far-western edges of the continent of Asia and the country of **Turkey**. It's bordered by **Turkey**, and by the countries of **Bulgaria**, **Romania**, **Ukraine**, Russia and **Georgia**.

Black Sea - World Atlas
www.worldatlas.com/aatlas/infopage/**blacksea**.htm

Deep Q&A: Semantic Parsing

- Complex, free-form, multi-clause questions

Google

what are the capitals of the countries that border the baltic sea

Web Maps Images News Shopping More Search tools

About 467,000 results (0.46 seconds)

Important cities along the Baltic include:

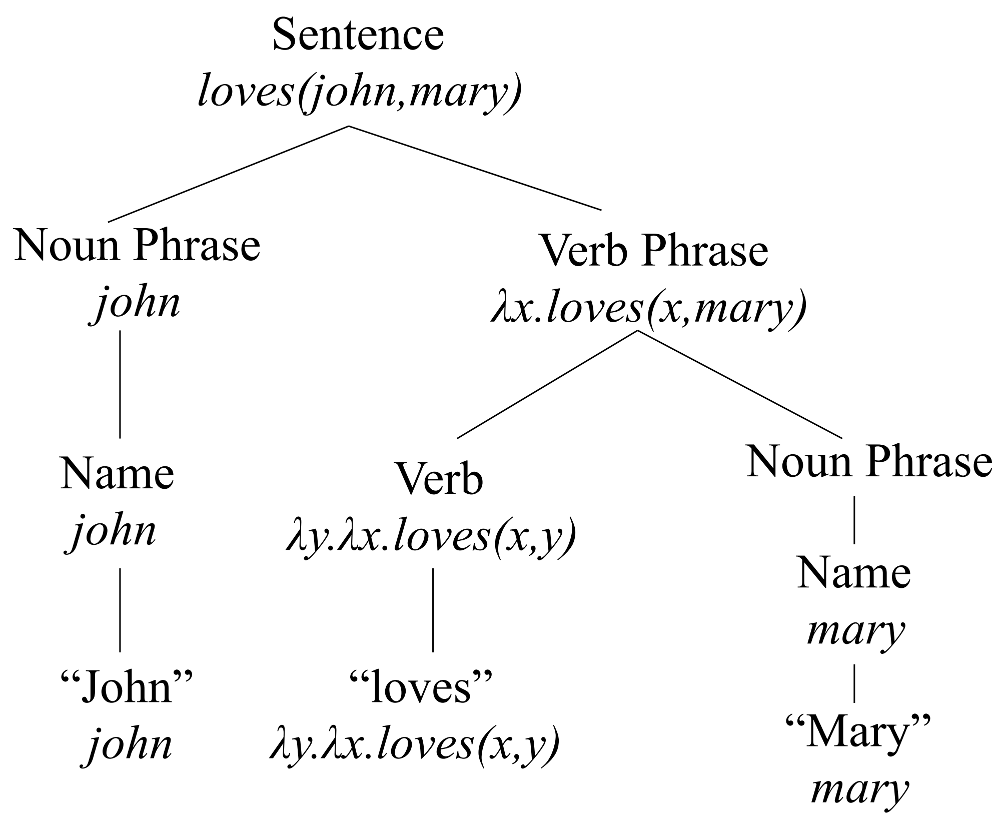
- The Russian cities of St. Petersburg and Kaliningrad.
- Stockholm, capital of Sweden.
- Copenhagen, capital of Denmark.
- Oslo, capital of Norway.
- Helsinki, capital of Finland.
- Tallinn, capital of Estonia.
- Riga, capital of Latvia.

Baltic Sea - Encyclopedia of Earth

www.eoearth.org/article/**Baltic_Sea**

Semantic Parsing: Logic forms

- Parsing with logic (booleans, individuals, functions) and lambda forms

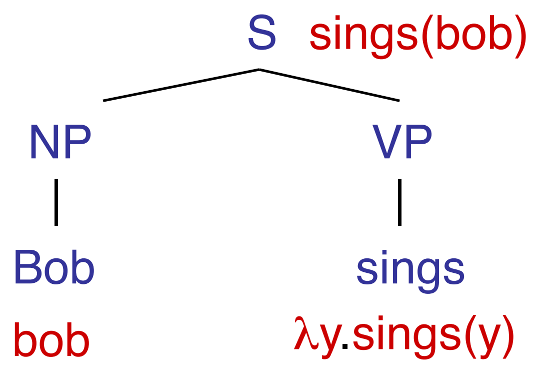


[Wong and Mooney, 2007; Zettlemoyer and Collins, 2007; Poon and Domingos, 2009; Artzi and Zettlemoyer, 2011, 2013; Kwiatkowski et al., 2013; Cai and Yates, 2013; Berant et al., 2013; Poon 2013; Berant and Liang, 2014; Iyyer et al., 2014]

Truth-Conditional Semantics

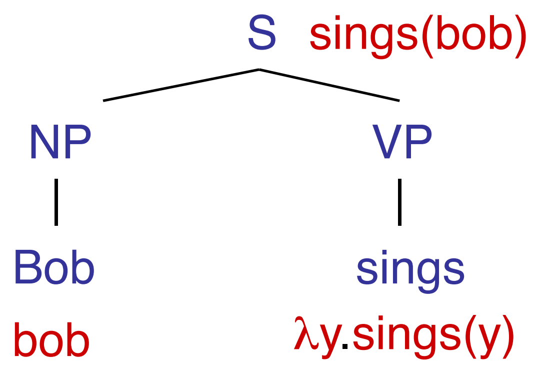
- Examples like "Bob sings"
- Logical translation of this will be something like: sings(bob)
- Types on these translations are entities (e) and truth-values (t), e.g.:

  bob: e
  sings(bob): t



Truth-Conditional Semantics

- For verbs (and verb phrases), sings combines with bob to produce sings(bob)



- In general, we use lambda-calculus or λ-calculus, i.e., a notation for functions whose arguments have not yet been filled/resolved/satisfied

- $\lambda x.\text{sings}(x)$

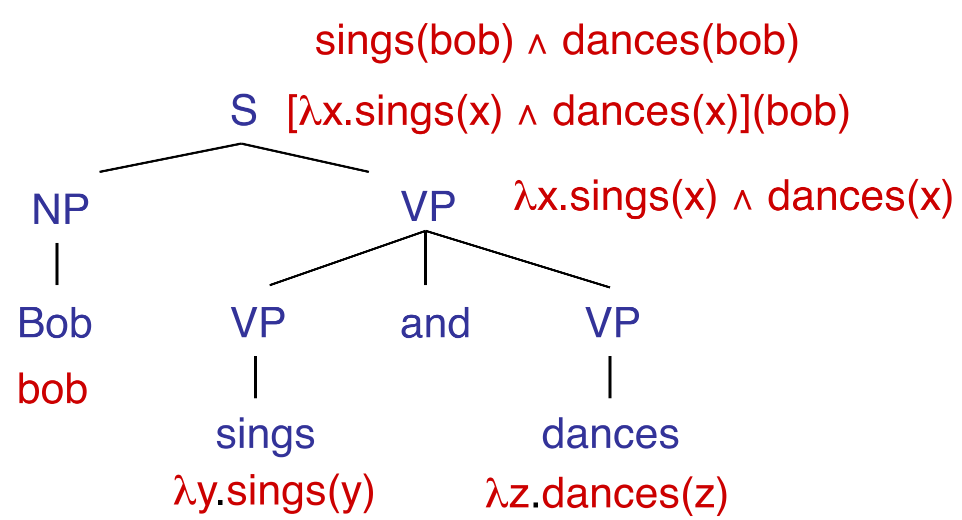
- This is a 'predicate', i.e., a function which take an entity (type e) and produces a truth value (type t), denoted as $e \rightarrow t$

Compositional Semantics

- Now after we have these meanings for words, we want to combine them into meaning for phrases and sentences
- For this, we associate a combination rule with each grammar rule of the parse tree, e.g.:

  S: $\beta(\alpha)$ → NP: α VP: β (*function application*)

  VP: $\lambda x \,.\, \alpha(x) \wedge \beta(x)$ → VP: α and: $\varnothing$ VP: β (*intersection*)

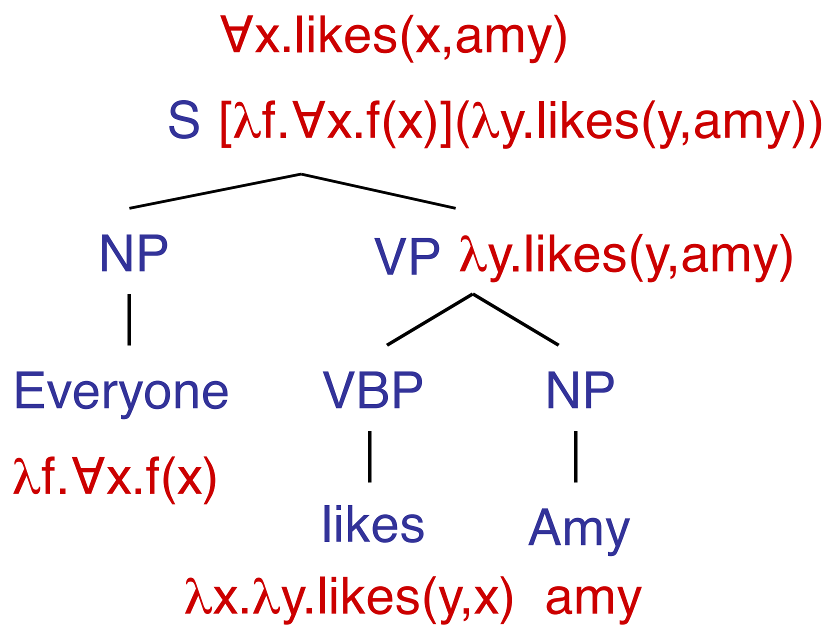


Transitive Verbs & Quantifiers

- Transitive verbs example is ‘like’ predicate:
- $\lambda x.\lambda y.likes(y,x)$
- These are two-place predicates hence $e \rightarrow (e \rightarrow t)$
- Whereas ‘likes Amy’ = $\lambda y.likes(y,amy)$ is just a one-place predicate because x has been satisfied/resolved

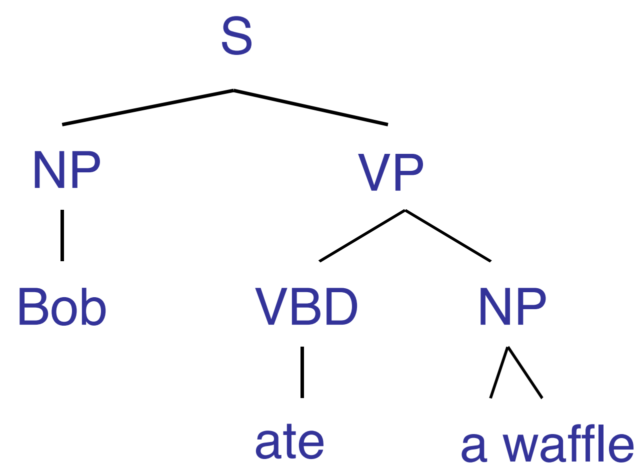
Transitive Verbs & Quantifiers

- What about the 'everyone' quantifier, e.g., "Everyone likes Amy"?
- Everyone = $\lambda f.\forall x.f(x)$
- See example figure on how this works →
- Gets tricky for examples like: "Amy likes everyone" and "Everyone likes someone"



Indefinites

- If we say “Bob ate a waffle” and “Amy ate a waffle”, then using:
  ate(bob, waffle)
  ate(amy, waffle)
- Doesn’t seem correct for ‘a waffle’
- More correct seems to use ‘there exists’ operator:
- $\exists x$: waffle(x) $\wedge$ ate(bob, x)
- And what about ‘the’ and ‘every’?



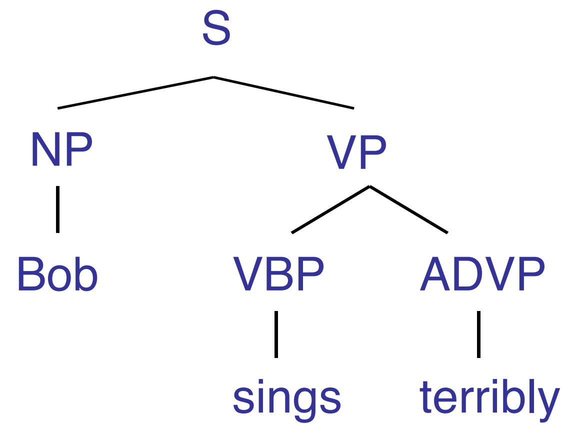
Tense and Events

- We need **event** variables because just verbs don't get us far!
- Example: "Bob sang"
- sang(bob)?
- $\exists$ e: singing(e) $\land$ agent(e, bob) $\land$ (time(e) < now)

- Hence, these event variable e help us represent complex tense and aspect structures:
- Example: "Bob had been singing when Mary coughed"
- $\exists$ e, e': singing(e) $\land$ agent(e, bob) $\land$
  coughing(e') $\land$ agent(e', mary) $\land$
  (start(e) < start(e') $\land$ end(e) = end(e')) $\land$
  (time(e') < now)

Adverbs

- Example: “Bob sings terribly”
- terribly(sings(bob))?
- (terribly(sings))(bob)?
- ∃ e: present(e) ∧ type(e, singing) ∧ agent(e, bob) ∧ manner(e, terrible)?
- Gets tricky pretty quickly…



CCG Parsing

- Combinatory Categorial Grammars:
- Each category encodes an argument sequence (fwd/bwd slashes specify argument order/direction)
- Closely related to lambda calculus
- Captures both syntactic and semantic info
- Naturally allows meaning representation and semantic parsing

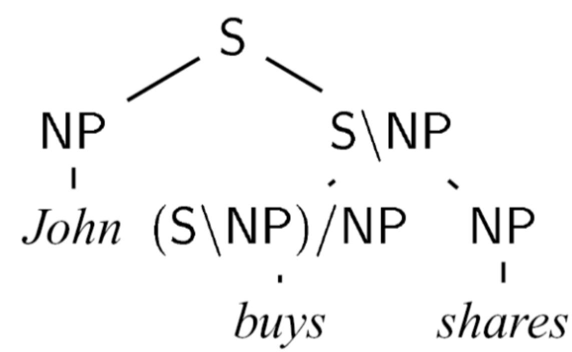
$$\textit{John} \vdash \mathsf{NP}$$

$$\textit{shares} \vdash \mathsf{NP}$$

$$\textit{buys} \vdash (\mathsf{S}\backslash\mathsf{NP})/\mathsf{NP}$$

$$\textit{sleeps} \vdash \mathsf{S}\backslash\mathsf{NP}$$

$$\textit{well} \vdash (\mathsf{S}\backslash\mathsf{NP})\backslash(\mathsf{S}\backslash\mathsf{NP})$$



CCG Parsing

- Given training examples with paired sentences/questions and their logical-form lambda calculus,

  Input: `List one way flights to Prague.`

  Output: $\lambda x.flight(x) \wedge one\_way(x) \wedge to(x,PRG)$

- This is a tricky learning problem because the derivations are not annotated, so we learn lexicon and parameters for a weighted CCG (e.g., based on [Zettlemoyer and Collins, 2005])

CCG Lexicon

Words	Category
flights	N : $\lambda x.flight(x)$
to	(N\N)/NP : $\lambda x.\lambda f.\lambda y.f(x) \wedge to(y,x)$
Prague	NP : PRG
New York city	NP : NYC
...	...

Combinator Rules

- Application Unary Rules:
  - `X/Y : f      Y : a   =>   X : f(a)`
  - `Y : a      X\Y : f   =>   X : f(a)`
- Composition Rules:
  - `X/Y : f   Y/Z : g   =>   X/Z : λx.f(g(x))`
  - `Y\Z : f   X\Y : g   =>   X\Z : λx.f(g(x))`
- Type Raising
- Crossed Composition

CCG Parsing Example

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

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

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

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

Weighted CCG

- Given a log-linear model with a CCG lexicon L, a feature vector f , and weights w , the best parse is

$$y^* = \underset{y}{\operatorname{argmax}} \; w \cdot f(x, y)$$

- Where y is the set of all parses for sentence x based on lexicon L

Lexicon Problem and Factored Lexicons

- Lexicon is key component of CCG
- But same word often paired with many different categories
- Difficult to learn with limited sentence-logicform data
- Factored Lexicons is one solution: lexical entries share info; decomposition leads to more compact lexicons

the house dog

house $\vdash ADJ : \lambda x.of(x, \iota y.house(y))$

the dog of the house

house $\vdash N : \lambda x.house(x)$

$$\iota x.dog(x) \wedge of(x, \iota y.house(y))$$

the garden dog

garden $\vdash ADJ : \lambda x.of(x, \iota y.garden(y))$

$$\iota x.dog(x) \wedge of(x, \iota y.garden(y))$$

[Artzi-FitzGerald-Zettlemoyer CCG tutorial]

Weak Supervision

- Instead of relying on sentence-logicform pairs as training data, we can learn from query-answer pairs

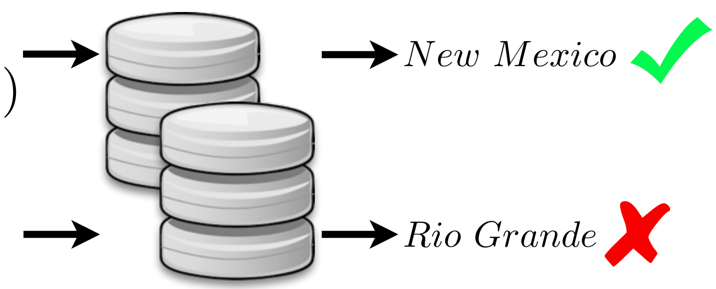
- Logical forms are latent, and we can check which one gets the correct answer on being executed against a knowledge base (KB)

What is the largest state that borders Texas?

$$New\ Mexico$$

$$argmax(\lambda x.state(x) \wedge border(x, TX), \lambda y.size(y))$$

$$argmax(\lambda x.river(x) \wedge in(x, TX), \lambda y.size(y))$$



[Clarke et al. 2010; Liang et al. 2011]

Weak Supervision

- Learning from Instruction-Demonstration Pairs

at the chair, move forward three steps past the sofa

Some examples from other domains:

- Sentences and labeled game states [Goldwasser and Roth 2011]
- Sentences and sets of physical objects [Matuszek et al. 2012]

[Chen and Mooney 2011; Kim and Mooney 2012;Artzi and Zettlemoyer 2013b]

Weak Supervision

- Learning from Conversation Logs

SYSTEM how can I help you ? (OPEN_TASK)

USER i ' d like to fly to new york

SYSTEM flying to new york . (CONFIRM: $from(fl,\ ATL)$) leaving what city ? (ASK: $\lambda x.from(fl,x)$)

USER from boston on june seven with american airlines

SYSTEM flying to new york . (CONFIRM: $to(fl,\ NYC)$) what date would you like to depart boston ? (ASK: $\lambda x.date(fl,x) \wedge to(fl,\ BOS)$)

USER june seventh

[CONVERSATION CONTINUES]

[Artzi and Zettlemoyer 2011]

Other Semantic Parsing Ideas

- Various recent ideas/extensions:
  - Dependency-based compositional semantics (DCS)
  - Bootstrapping w/ conversations
  - On-the-fly ontology matching
  - Question answering on Freebase
  - Paraphrasing
  - Unsupervised SP (clustering lambda forms)
  - Grounded USP (via databases)

[Wong and Mooney, 2007; Zettlemoyer and Collins, 2007; Poon and Domingos, 2009;
Artzi and Zettlemoyer, 2011, 2013; Kwiatkowski et al., 2013; Cai and Yates, 2013;
Berant et al., 2013; Poon 2013; Berant and Liang, 2014; Iyyer et al., 2014; Yao and Van Durne, 2014]

Dependency-based Compositional Semantics (DCS)

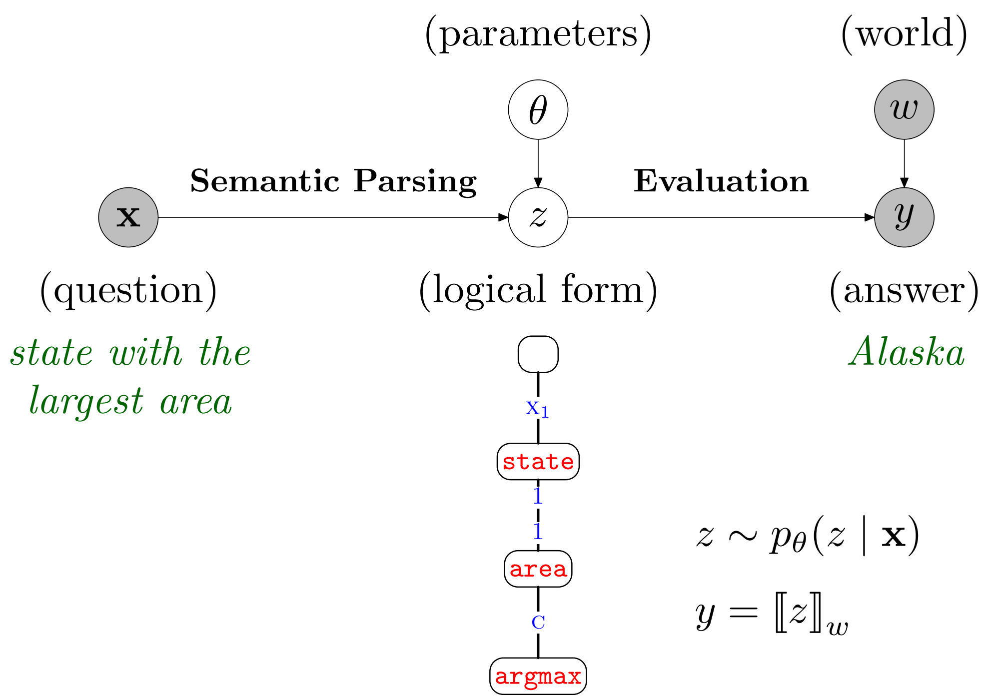


Figure 1: Our probabilistic model: a question $\mathbf{x}$ is mapped to a latent logical form z, which is then evaluated with respect to a world w (database of facts), producing an answer y. We represent logical forms z as labeled trees, induced automatically from $(\mathbf{x}, y)$ pairs.

[Liang et al., 2013]

Dependency-based Compositional Semantics (DCS)

Relations $\mathcal{R}$

$\begin{smallmatrix} j \\ j' \end{smallmatrix}$	(join)	E	(extract)
Σ	(aggregate)	Q	(quantify)
$\text{X}_{\mathbf{i}}$	(execute)	C	(compare)

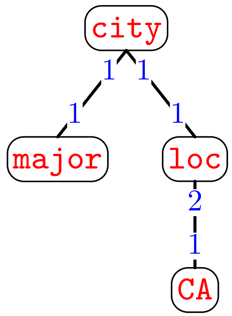
Table 1: Possible relations appearing on the edges of a DCS tree. Here, $j, j' \in \{1, 2, \ldots\}$ and $\mathbf{i} \in \{1, 2, \ldots\}^*$.

[Liang et al., 2013]

Dependency-based Compositional Semantics (DCS)

Example: *major city in California*

$z = \langle \texttt{city}; \tfrac{1}{1}{:}\langle \texttt{major} \rangle ; \tfrac{1}{1}{:}\langle \texttt{loc}; \tfrac{2}{1}{:}\langle \texttt{CA} \rangle \rangle \rangle$



$$\begin{aligned} &\lambda c\, \exists m\, \exists \ell\, \exists s\, . \\ &\quad \texttt{city}(c) \wedge \texttt{major}(m) \wedge \\ &\quad \texttt{loc}(\ell) \wedge \texttt{CA}(s) \wedge \\ &\quad c_1 = m_1 \wedge c_1 = \ell_1 \wedge \ell_2 = s_1 \end{aligned}$$

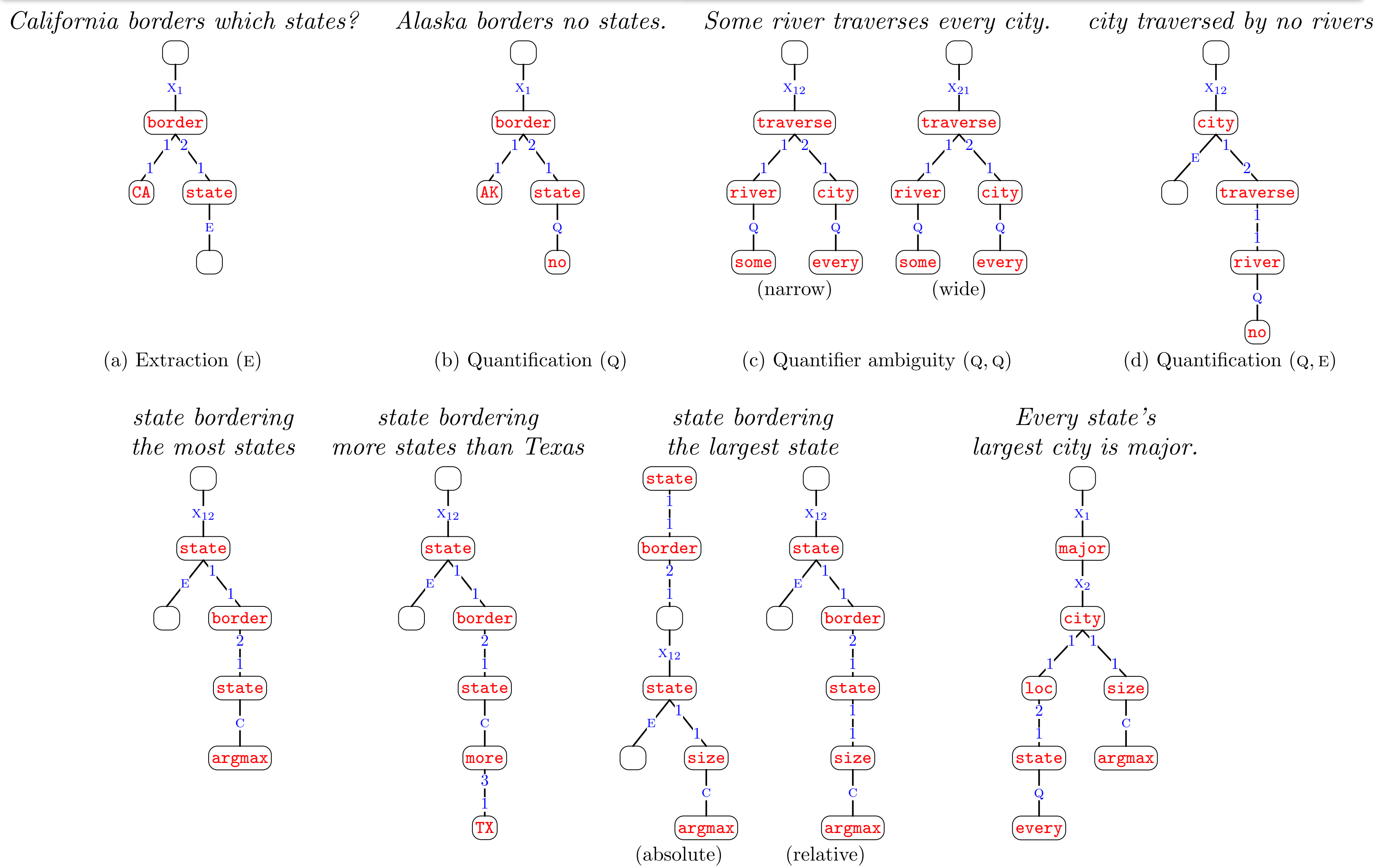
(a) DCS tree (b) Lambda calculus formula

(c) Denotation: $[\![z]\!]_w = \{\texttt{SF}, \texttt{LA}, \ldots\}$

Figure 2: (a) An example of a DCS tree (written in both the mathematical and graphical notation). Each node is labeled with a predicate, and each edge is labeled with a relation. (b) A DCS tree z with only join relations encodes a constraint satisfaction problem. (c) The denotation of z is the set of consistent values for the root node.

[Liang et al., 2013]

Dependency-based Compositional Semantics (DCS)



(a) Extraction (E)

(b) Quantification (Q)

(c) Quantifier ambiguity (Q, Q)

(d) Quantification (Q, E)

(e) Superlative (C)

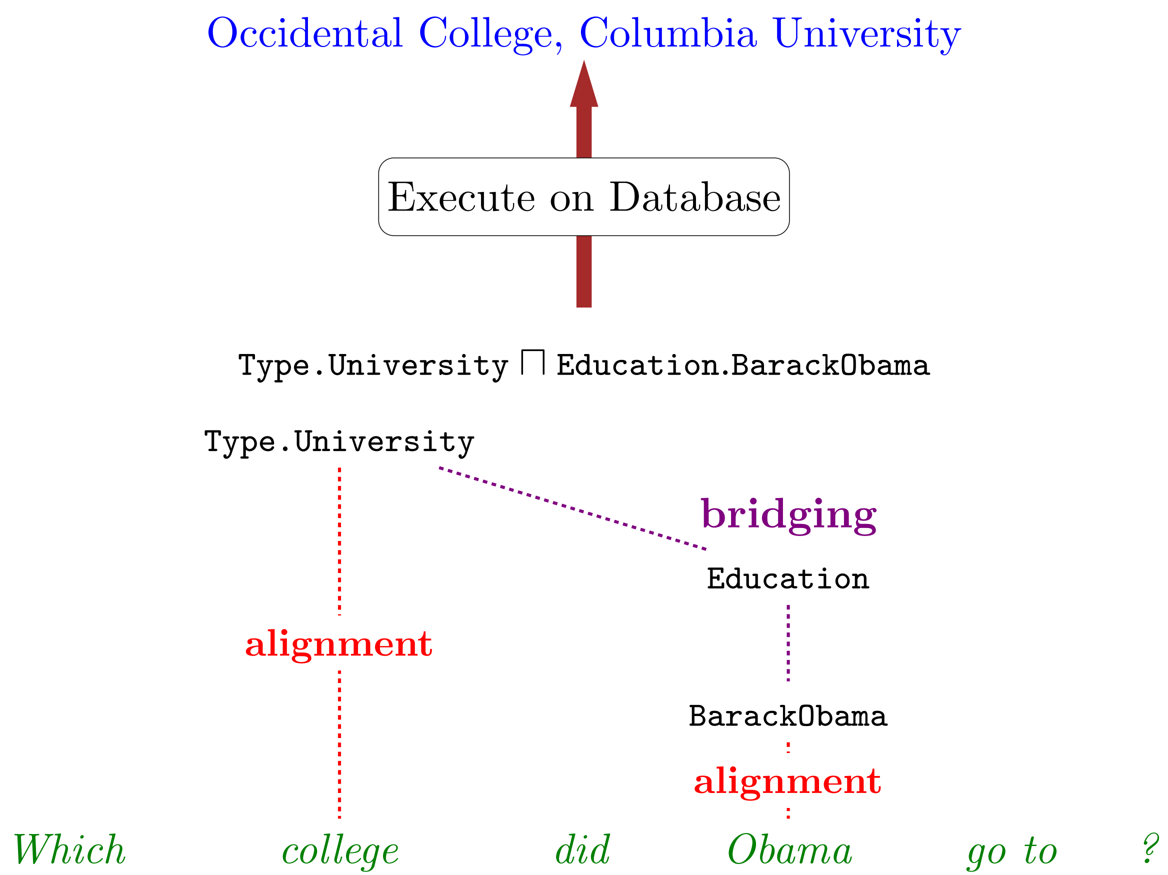
(f) Comparative (C)

(g) Superlative ambiguity (C)

(h) Quantification+Superlative (Q, C)

[Liang et al., 2013]

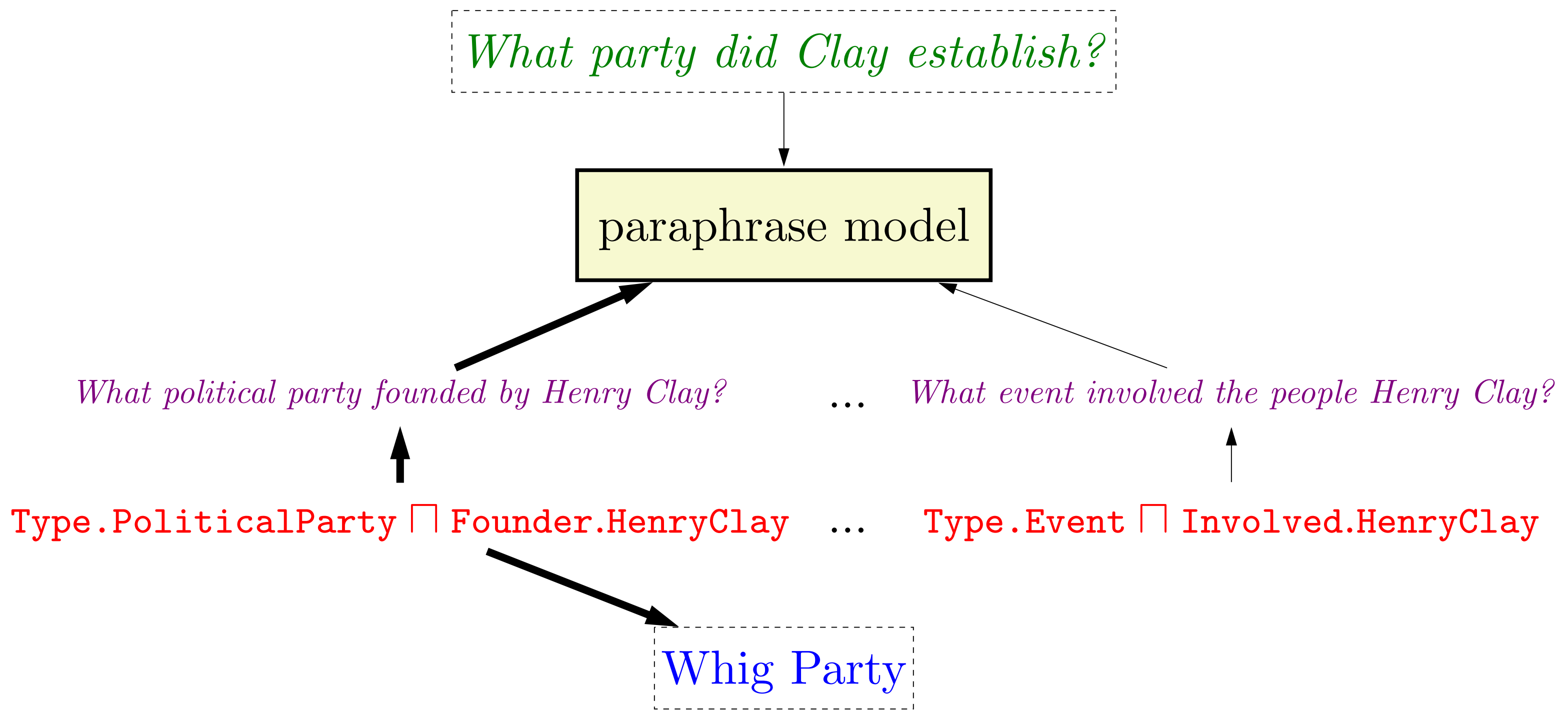
Semantic Parsing on Freebase



Mapping questions to answers via latent logical forms. To narrow down the logical predicate space, they use a (i) coarse *alignment* based on Freebase and a text corpus and (ii) a *bridging* operation that generates predicates compatible with neighboring predicates.

[Berant et al., 2013]

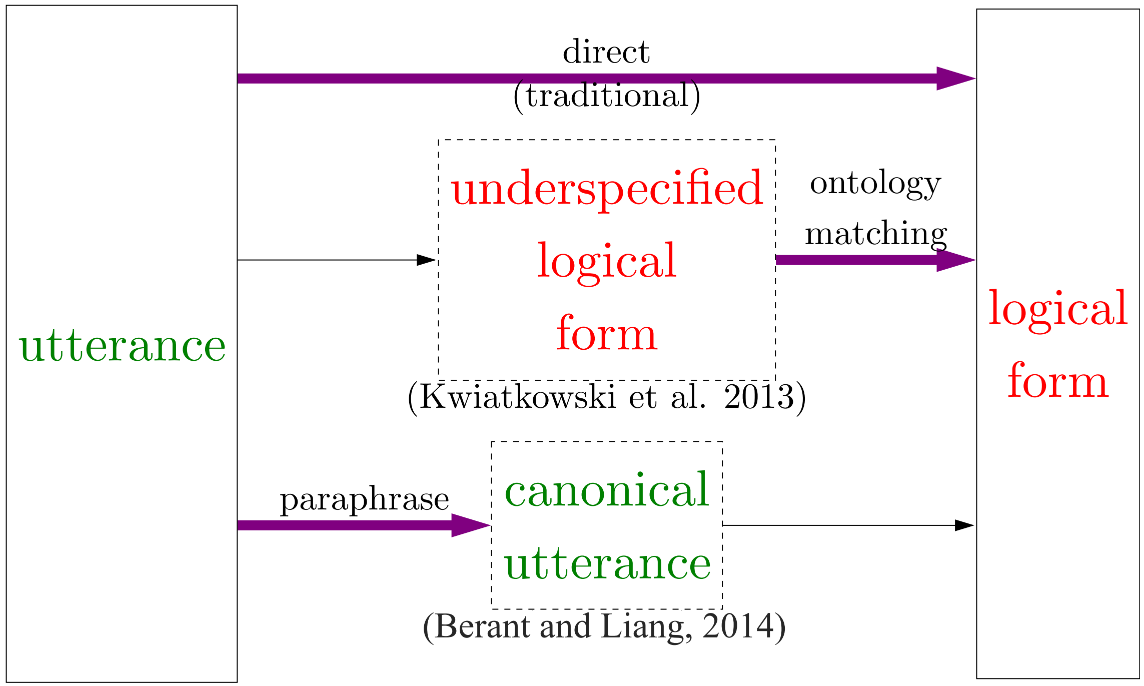
Semantic Parsing via Paraphrasing



For each candidate logical form (red), they generate canonical utterances (purple). The model is trained to paraphrase the input utterance (green) into the canonical utterances associated with the correct denotation (blue).

[Berant and Liang, 2014]

Semantic Parsing via Ontology Matching



The main challenge in semantic parsing is the mismatch between language and the knowledge base. (a) Traditional: map utterances directly to logical forms, (b) Kwiatkowski et al. (2013): map utterance to intermediate, underspecified logical form, then perform ontology matching to handle the mismatch, (c) Berant and Liang (2014): generate intermediate, canonical text utterances for logical forms, then use paraphrase models.

[Kwiatkowski et al., 2013; Berant and Liang, 2014]