The Challenge of Composition in Distributional and Formal Semantics Part II

Ran Tian[1], Koji Mineshima[2] and Pascual Martínez-Gómez[3]

[1]Tohoku University, Japan
[2]Ochanomizu University, Japan
[3]AIST, Japan

IJCNLP 2017, Taipei, Taiwan
November 27, 2017

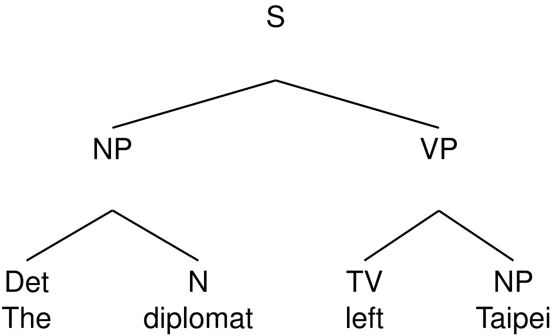
Three Challenges

1. Meaning Representations (MRs): what are proper MRs for natural languages?
2. Compositional Semantics: how to compute the MR of a complex expression from the MRs of its parts?
3. Inference: how can we do inference with MRs?

- We start with Question 2:
  - Combinatory Categorial Grammar (CCG)
  - Lambda Calculus
- And then move on to Question 1 and Question 3
  - First-order and Higher-order Logics
  - A MR is good if it enables correct and efficient inferences

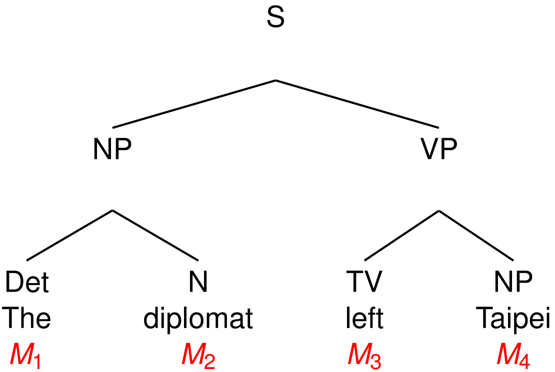
Semantic Composition via Phrase Structure Grammar

S	$\rightarrow$	NP VP
NP	$\rightarrow$	Det N
VP	$\rightarrow$	TV NP
Det	$\rightarrow$	the
N	$\rightarrow$	diplomat
NP	$\rightarrow$	Taipei
TV	$\rightarrow$	left



Semantic Composition via Phrase Structure Grammar

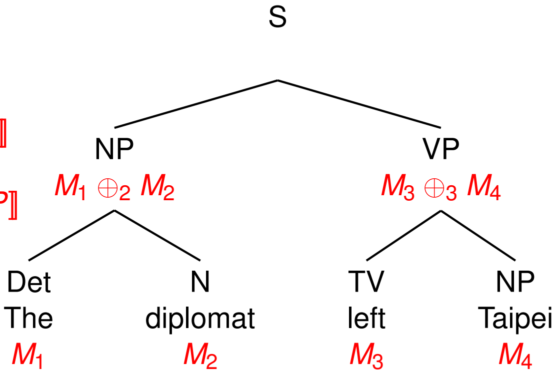
S	$\rightarrow$	NP VP
NP	$\rightarrow$	Det N
VP	$\rightarrow$	TV NP
Det	$\rightarrow$	the
N	$\rightarrow$	diplomat
NP	$\rightarrow$	Taipei
TV	$\rightarrow$	left



- Assign a MR to each leaf node

Semantic Composition via Phrase Structure Grammar

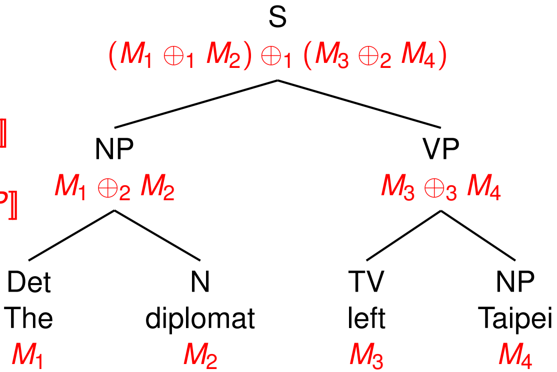
$$
\begin{aligned}
\text{S} &\rightarrow \text{NP VP} \\
\text{NP} &\rightarrow \text{Det N} \\
& \textcolor{red}{[\![NP]\!] = [\![Det]\!] \oplus_2 [\![N]\!]} \\
\text{VP} &\rightarrow \text{TV NP} \\
& \textcolor{red}{[\![VP]\!] = [\![TV]\!] \oplus_3 [\![NP]\!]} \\
\text{Det} &\rightarrow \text{the} \\
\text{N} &\rightarrow \text{diplomat} \\
\text{NP} &\rightarrow \text{Taipei} \\
\text{TV} &\rightarrow \text{left}
\end{aligned}
$$



- Assign a MR to each leaf node
- Compute the MR of each phrase in terms of the MRs of its parts, according to meaning composition rules

Semantic Composition via Phrase Structure Grammar

S $\rightarrow$ NP VP
$[\![S]\!] = [\![NP]\!] \oplus_1 [\![VP]\!]$
NP $\rightarrow$ Det N
$[\![NP]\!] = [\![Det]\!] \oplus_2 [\![N]\!]$
VP $\rightarrow$ TV NP
$[\![VP]\!] = [\![TV]\!] \oplus_3 [\![NP]\!]$
Det $\rightarrow$ the
N $\rightarrow$ diplomat
NP $\rightarrow$ Taipei
TV $\rightarrow$ left



- Assign a MR to each leaf node
- Compute the MR of each phrase in terms of the MRs of its parts, according to meaning composition rules

Semantic Composition via Phrase Structure Grammar

S $\rightarrow$ NP VP

$[\![S]\!] = [\![NP]\!] \oplus_1 [\![VP]\!]$

NP $\rightarrow$ Det N

$[\![NP]\!] = [\![Det]\!] \oplus_2 [\![N]\!]$

VP $\rightarrow$ TV NP

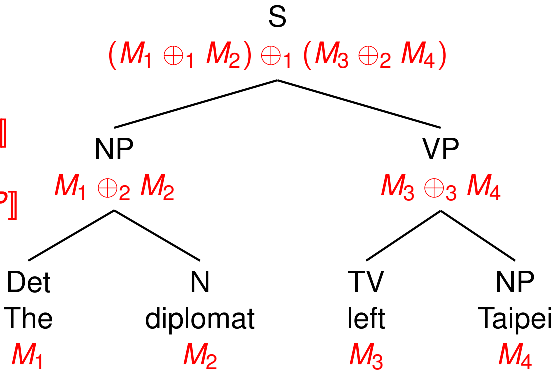
$[\![VP]\!] = [\![TV]\!] \oplus_3 [\![NP]\!]$

Det $\rightarrow$ the

N $\rightarrow$ diplomat

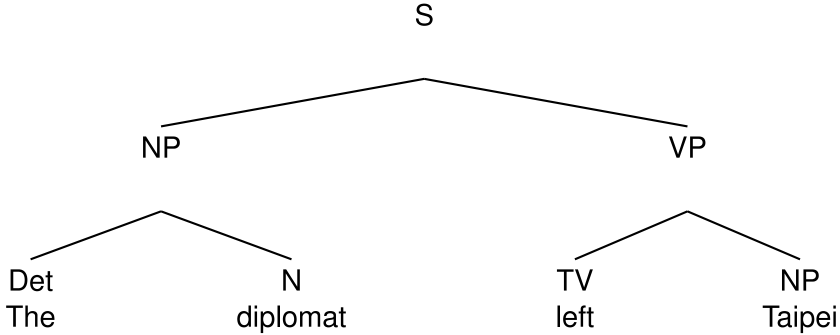
NP $\rightarrow$ Taipei

TV $\rightarrow$ left

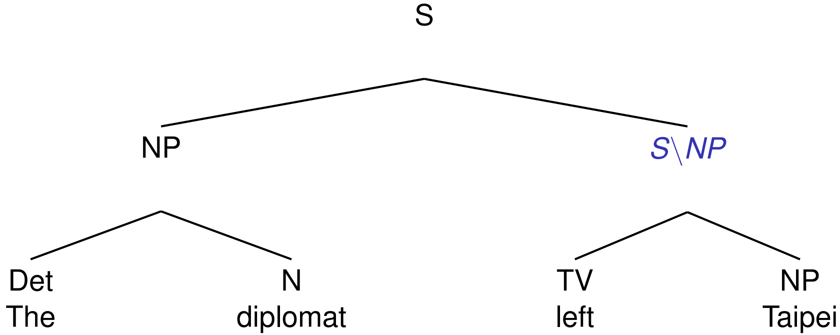


- Assign a MR to each leaf node
- Compute the MR of each phrase in terms of the MRs of its parts, according to meaning composition rules
- Many grammar rules, many composition rules

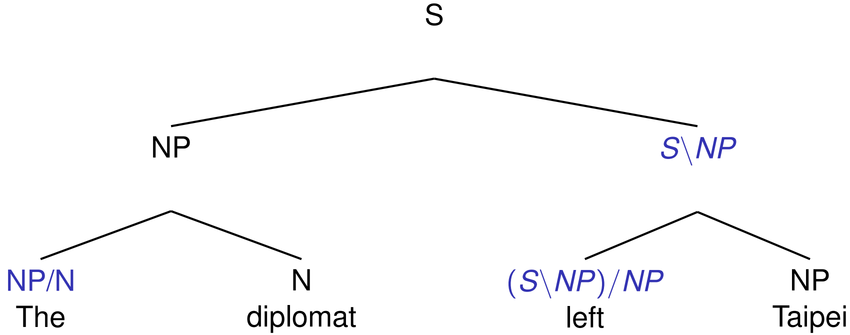
Semantic Composition via Categorial Grammar (CG)



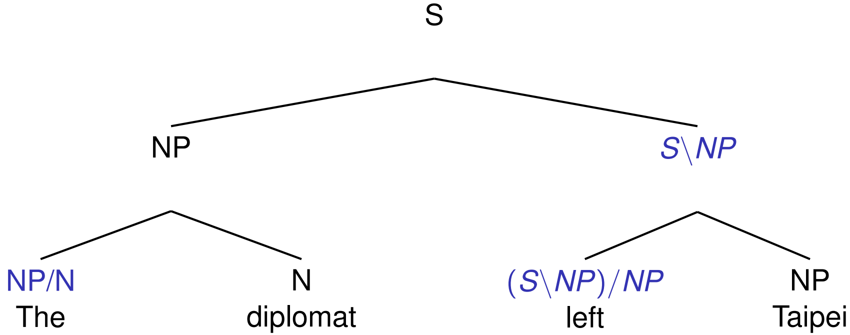
Semantic Composition via Categorial Grammar (CG)



Semantic Composition via Categorial Grammar (CG)

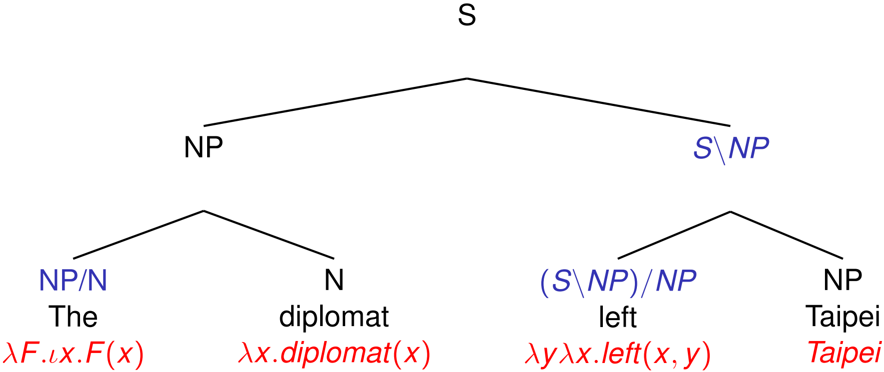


Semantic Composition via Categorial Grammar (CG)



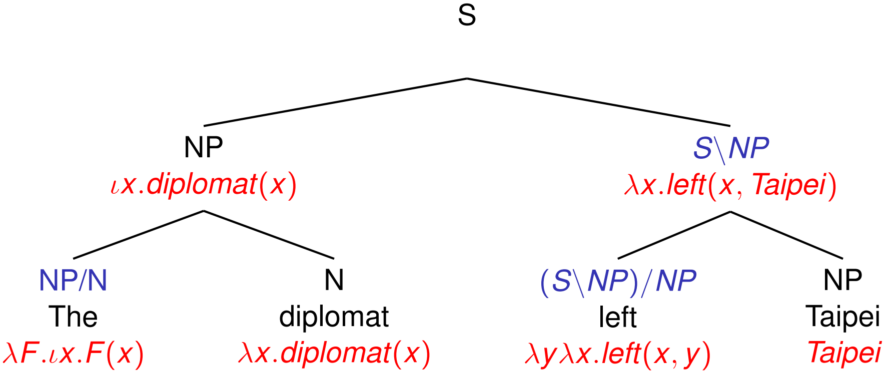
- A small set of basic categories (S, NP, N)
- Each functional category of the form X/Y and $X\backslash Y$ specifies how words combine with each other

Semantic Composition via Categorial Grammar (CG)



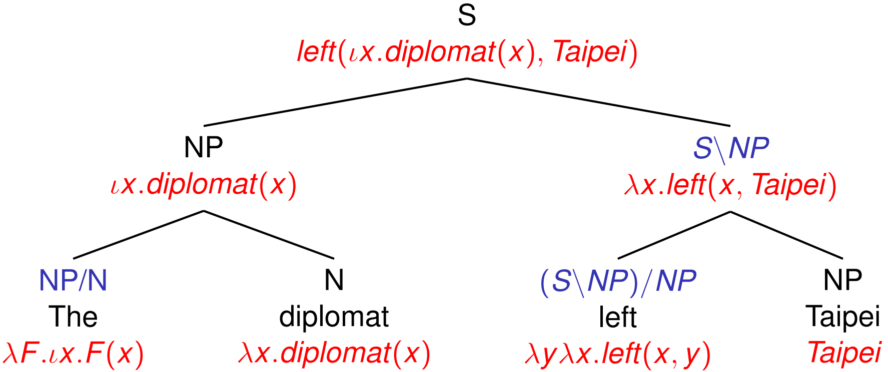
- A small set of basic categories (S, NP, N)
- Each functional category of the form X/Y and $X\backslash Y$ specifies how words combine with each other and, at the same time, how to compute the MR of a phrase node.

Semantic Composition via Categorial Grammar (CG)



- A small set of basic categories (S, NP, N)
- Each functional category of the form X/Y and $X\backslash Y$ specifies how words combine with each other and, at the same time, how to compute the MR of a phrase node.

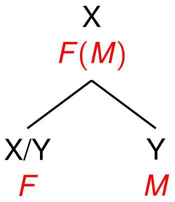
Semantic Composition via Categorial Grammar (CG)



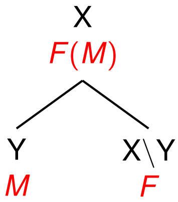
- A small set of basic categories (S, NP, N)
- Each functional category of the form X/Y and $X \backslash Y$ specifies how words combine with each other and, at the same time, how to compute the MR of a phrase node.
- A small set of grammar rules and meaning composition rules

Combinatory Rules

Forward Function Application



Backward Function Application



Derivation trees

- Turn the tree upside down (for a historical reason)
- Derivation trees (proof trees)

$$\dfrac{\dfrac{\texttt{John}}{\begin{array}{c}NP\\ john\end{array}} \qquad \dfrac{\dfrac{\texttt{likes}}{\begin{array}{c}(S\backslash NP)/NP\\ \lambda y\lambda x.like(x,y)\end{array}} \qquad \dfrac{\texttt{Mary}}{\begin{array}{c}NP\\ mary\end{array}}}{\begin{array}{c}S\backslash NP\\ \lambda x.like(x,mary)\end{array}}{>}}{\begin{array}{c}S\\ like(john,mary)\end{array}}{<}$$

- Function Application rules

$$\dfrac{\begin{array}{c}X/Y\\ F\end{array} \qquad \begin{array}{c}Y\\ M\end{array}}{\begin{array}{c}X\\ F(M)\end{array}}{>} \qquad\qquad \dfrac{\begin{array}{c}Y\\ M\end{array} \qquad \begin{array}{c}X\backslash Y\\ F\end{array}}{\begin{array}{c}X\\ F(M)\end{array}}{<}$$

From AB to CCG

- The fragment of categorial grammar consisting of function application rules is called AB grammar (Ajdukiewicz, 1935; Bar-Hillel, 1953)
- Adding more combinatory rules leads to Combinatory Categorial Grammar (CCG) (Steedman, 2000, 2012)

More combinatory rules

Function Composition rules

$$\dfrac{\underset{f}{X/Y} \qquad \underset{g}{Y/Z}}{\underset{\lambda x.f(g(x))}{X/Z}} >\mathbf{B} \qquad\qquad \dfrac{\underset{g}{Y\backslash Z} \qquad \underset{f}{X\backslash Y}}{\underset{\lambda x.f(g(x))}{X\backslash Z}} <\mathbf{B}$$

Crossed Composition rules

$$\dfrac{\underset{f}{X/Y} \qquad \underset{g}{Y\backslash Z}}{\underset{\lambda x.f(g(x))}{X\backslash Z}} >\mathbf{B}_\times \qquad\qquad \dfrac{\underset{g}{Y/Z} \qquad \underset{f}{X\backslash Y}}{\underset{\lambda x.f(g(x))}{X/Z}} <\mathbf{B}_\times$$

A more complicated derivation

John doesn't like Mary

$\neg like(john, mary)$

$$
\begin{array}{cccc}
 & \texttt{doesn't} & \texttt{like} & \\
 & (S\backslash NP)/\boxed{(S\backslash NP)} & \boxed{(S\backslash NP)}/NP & \\
 & \lambda F \lambda x.\neg F(x) & \lambda y \lambda x.like(x, y) & \texttt{Mary} \\
\hline
 & \multicolumn{2}{c}{(S\backslash NP)/NP \quad >\mathbf{B}} & NP \\
\texttt{John} & \multicolumn{2}{c}{\lambda y \lambda x.\neg like(x, y)} & mary \\
\hline
NP & \multicolumn{3}{c}{S\backslash NP \quad >} \\
john & \multicolumn{3}{c}{\lambda x.\neg like(x, mary)} \\
\hline
\multicolumn{4}{c}{S \quad <} \\
\multicolumn{4}{c}{\neg like(john, mary)}
\end{array}
$$

Right node raising shows that "*doesn't like*" can be a constituent:

John [[respects] but [doesn't like]] Mary.

$respect(john, mary) \wedge \neg like(john, mary)$

Lambda calculus

- A formal system to represent computation
- Simple yet very expressive

function	input	output
$\lambda x.x + 2$	number x	$x + 2$
$\lambda x.walk(x)$	entity x	proposition $walk(x)$

β-conversion (simplification, substitution):

$$\overbrace{(\lambda x.[\dots\ x\ \dots])}^{\text{function}}\ \overbrace{(a)}^{\text{argument}} = [\ \dots\ a\ \dots\ ]$$

Examples:

- $(\lambda x.x + 2)(5) = 5 + 2$
- $(\lambda x.walk(x))(john) = walk(john)$

β-conversion: more examples

β-conversion (simplification):

$$\underbrace{(\lambda x.[\ldots\ x\ \ldots])}_{\text{function}}\ \underbrace{(a)}_{\text{argument}} = [\ \ldots\ a\ \ldots\ ]$$

1. $(\lambda x.like(x, y))(john) = like(john, y)$
2. $(\lambda y.like(x, y))(john) = like(x, john)$
3. $(\lambda x.like(x, x))(john) = like(john, john)$
4. $(\lambda x.like(mary, x) \wedge boy(x))(john) = like(mary, john) \wedge boy(john)$
5. $((\lambda y.\lambda x.like(x, y))(john))(mary) =$
   $(\lambda x.like(x, john))(mary) = like(mary, john)$

α-conversion

α-conversion (renaming):

$$\lambda \boxed{x}.[\ldots \boxed{x} \ldots] = \lambda \boxed{y}.[\ldots \boxed{y} \ldots]$$

Example:

$$\lambda x.boy(x) \wedge love(x)(z) \;=\; \lambda y.boy(y) \wedge love(y)(z)$$

α-conversion

α-conversion (renaming):

$$\lambda \boxed{x} .[\ldots \boxed{x} \ldots] = \lambda \boxed{y} .[\ldots \boxed{y} \ldots]$$

Example:

$$\lambda x.\mathit{boy}(x) \wedge \mathit{love}(x)(z) \;=\; \lambda y.\mathit{boy}(y) \wedge \mathit{love}(y)(z)$$

Lambda calculus vs. Set Theory

Lambda calculus	Set Theory
$\lambda x.Fx$	$\{x \mid Fx\}$
$(\lambda x.Fx)(a)$	$a \in \{x \mid Fx\}$
$(\lambda x.Fx)(a) = Fa$	$a \in \{x \mid Fx\} \Leftrightarrow Fa$

Adding type information

- But is meaning composition via lambda calculus always safe?
- What we need: Type safety
- Type safety lies at the heart of formal compositional semantics

Adding type information

- But is meaning composition via lambda calculus always safe?
- What we need: Type safety
- Type safety lies at the heart of formal compositional semantics

Define simple types:

Type	Meaning
$\mathtt{E}$	Entity
$\mathtt{T}$	Proposition
$X \to Y$	A function from X to Y

Examples:

john, *mary* :	$\mathtt{E}$	entity
$\lambda x.walk(x)$:	$\mathtt{E} \to \mathtt{T}$	function from entities to propositions
$\lambda y.\lambda x.like(x, y)$:	$\mathtt{E} \to (\mathtt{E} \to \mathtt{T})$	function from two entities to propositions

Adding type information

- But is meaning composition via lambda calculus always safe?
- What we need: Type safety
- Type safety lies at the heart of formal compositional semantics

Define simple types:

Type	Meaning
$\mathtt{E}$	Entity
$\mathtt{T}$	Proposition
$X \rightarrow Y$	A function from X to Y

Examples:

john, *mary* :	$\mathtt{E}$	entity
$\lambda x.walk(x)$:	$\mathtt{E} \rightarrow \mathtt{T}$	function from entities to propositions
$\lambda y.\lambda x.like(x, y)$:	$\mathtt{E} \rightarrow (\mathtt{E} \rightarrow \mathtt{T})$	function from two entities to propositions
$walk(john)$:	$\mathtt{T}$	proposition

Adding type information

- But is meaning composition via lambda calculus always safe?
- What we need: Type safety
- Type safety lies at the heart of formal compositional semantics

Define simple types:

Type	Meaning
$\mathtt{E}$	Entity
$\mathtt{T}$	Proposition
$X \to Y$	A function from X to Y

Examples:

$\mathit{john}, \mathit{mary}$:	$\mathtt{E}$	entity
$\lambda x.\mathit{walk}(x)$:	$\mathtt{E} \to \mathtt{T}$	function from entities to propositions
$\lambda y.\lambda x.\mathit{like}(x, y)$:	$\mathtt{E} \to (\mathtt{E} \to \mathtt{T})$	function from two entities to propositions
$\mathit{walk}(\mathit{john})$:	$\mathtt{T}$	proposition
$\mathit{like}(\mathit{john}, \mathit{mary})$:	$\mathtt{T}$	proposition

Adding type information

- But is meaning composition via lambda calculus always safe?
- What we need: Type safety
- Type safety lies at the heart of formal compositional semantics

Define simple types:

Type	Meaning
$\mathtt{E}$	Entity
$\mathtt{T}$	Proposition
$X \to Y$	A function from X to Y

Examples:

$john, mary$:	$\mathtt{E}$	entity
$\lambda x.walk(x)$:	$\mathtt{E} \to \mathtt{T}$	function from entities to propositions
$\lambda y.\lambda x.like(x, y)$:	$\mathtt{E} \to (\mathtt{E} \to \mathtt{T})$	function from two entities to propositions
$walk(john)$:	$\mathtt{T}$	proposition
$like(john, mary)$:	$\mathtt{T}$	proposition
$walk(like)$:	# type-mismatch	

Types control semantic composition

β-conversion (simplification):

Type: $A \rightarrow B$ Type: A Type: B

$(\lambda x.[\ldots\ x\ \ldots])\ (a) = [\ \ldots\ \mathrm{a}\ \ldots\ ]$

Example:

Type: $\mathtt{E} \rightarrow \mathtt{T}$ Type: $\mathtt{E}$ Type: $\mathtt{T}$

$(\lambda x.walk(x))\ (john) = \mathrm{walk(john)}$

CCG-based Compositional Semantics

- Type information is always implicit in CCG-derivation trees

$$
\dfrac{
\dfrac{\mathtt{John}}{\begin{array}{c} NP \\ john \end{array}}
\qquad
\dfrac{
\dfrac{\mathtt{likes}}{\begin{array}{c} (S\backslash NP)/NP \\ \lambda y.\lambda x.like(x, y) \end{array}}
\qquad
\dfrac{\mathtt{Mary}}{\begin{array}{c} NP \\ mary \end{array}}
}{\begin{array}{c} S\backslash NP \\ \lambda x.like(x, mary) \end{array}} >
}{\begin{array}{c} S \\ like(john, mary) \end{array}} <
$$

CCG-based Compositional Semantics

- Type information is always implicit in CCG-derivation trees

$$
\dfrac{\dfrac{\texttt{John}}{\begin{array}{c} NP \\ john : \mathtt{E} \end{array}} \quad \dfrac{\dfrac{\texttt{likes}}{\begin{array}{c} (S\backslash NP)/NP \\ \lambda y.\lambda x.like(x,y) : \mathtt{E} \to (\mathtt{E} \to \mathtt{T}) \end{array}} \quad \dfrac{\texttt{Mary}}{\begin{array}{c} NP \\ mary : \mathtt{E} \end{array}}}{\begin{array}{c} S\backslash NP \\ \lambda x.like(x, mary) : \mathtt{E} \to \mathtt{T} \end{array}} >}{\begin{array}{c} S \\ like(john, mary) : \mathtt{T} \end{array}} <
$$

Syntactic sugar

Special symbols (constants) to represent logical expression:

Logical expression	Type	
$\neg$	$\mathtt{T} \to \mathtt{T}$	negation
$\wedge$	$\mathtt{T} \to \mathtt{T} \to \mathtt{T}$	conjunction
$\vee$	$\mathtt{T} \to \mathtt{T} \to \mathtt{T}$	disjunction
$\to$	$\mathtt{T} \to \mathtt{T} \to \mathtt{T}$	implication
$\forall$	$(\mathtt{E} \to \mathtt{T}) \to \mathtt{T}$	universal quantifier
$\exists$	$(\mathtt{E} \to \mathtt{T}) \to \mathtt{T}$	existential quantifier
ι	$(\mathtt{E} \to \mathtt{T}) \to \mathtt{E}$	iota operator

We can write :

$$\begin{array}{lll} A \wedge B & \text{for} & \wedge(A, B) \\ \forall x F x & \text{for} & \forall(\lambda x.Fx) \\ \exists x F x & \text{for} & \exists(\lambda x.Fx) \end{array}$$

and so on.

- Logics can be encoded in Lambda calculus!

From categories to types

We can define a homomorphism $(\cdot)^\bullet$ from categories to types:

$$\begin{aligned} NP^\bullet &= \mathtt{E} \\ S^\bullet &= \mathtt{T} \\ (X/Y)^\bullet &= (X\backslash Y)^\bullet = X^\bullet \to Y^\bullet \end{aligned}$$

Example:

- $(S\backslash NP)^\bullet = \mathtt{E} \to \mathtt{T}$ (intransitive verbs)
- $((S\backslash NP)/NP)^\bullet = \mathtt{E} \to (\mathtt{E} \to \mathtt{T})$ (transitive verbs)

- As for as the type homomorphism is preserved, there will be no type-clash during meaning composition.

Lexicon: open words and closed words

- For an open word, we can use a template to specify its MR.
- φ is the position in which the lemma of a word appears.

Category	Meaning templates	Type
$S\backslash NP$	$\lambda x.\varphi(x)$	$\mathtt{E} \to \mathtt{T}$
$(S\backslash NP)/NP$	$\lambda y.\lambda x.\varphi(x, y)$	$\mathtt{E} \to (\mathtt{E} \to \mathtt{T})$

- For a closed word, we can directly assign its MR.
- For example, if we are interested in logical expressions, we can use the following lexical entries:

Lemma	Category	MR	Type
some	NP/N	$\lambda F \lambda G.\exists x(Fx \land Gx)$	$(\mathtt{E} \to \mathtt{T}) \to (\mathtt{E} \to \mathtt{T}) \to \mathtt{T}$
every	NP/N	$\lambda F \lambda G.\forall x(Fx \land Gx)$	$(\mathtt{E} \to \mathtt{T}) \to (\mathtt{E} \to \mathtt{T}) \to \mathtt{T}$
no	NP/N	$\lambda F \lambda G.\neg\exists x(Fx \land Gx)$	$(\mathtt{E} \to \mathtt{T}) \to (\mathtt{E} \to \mathtt{T}) \to \mathtt{T}$

Excerpts of Templates from ccg2lambda

CCG category	Meaning Representation
NP	$\lambda NF.\exists x(N(\varphi, x) \wedge F(x))$
$S\backslash NP_{nom}$	$\lambda QK.Q(\lambda I.I, \lambda x.\exists v(K(\varphi, v) \wedge (\mathit{Nom}(v) = x)))$
$S\backslash NP_{nom}/NP_{acc}$	$\lambda Q_2 Q_1 K.Q_1(\lambda I.I, \lambda x_1.Q_2(\lambda I.I, \lambda x_2.\exists v(K(\varphi, v) \wedge (\mathit{Nom}(v) = x_1) \wedge (\mathit{Acc}(v) = x_2))))$
S/S	$\lambda SK.S(\lambda Jv.K(\lambda v'.(J(v') \wedge \varphi(v')), v))$
NP/NP	$\lambda QNF.Q(\lambda Gx.N(\lambda y.(\varphi(y) \wedge G(y)), x), F)$

Types

$$\text{Type} \;::=\; \mathtt{E} \mid \mathtt{Event} \mid \mathtt{T} \mid X \Rightarrow Y$$

Mapping from syntactic categories to semantic types

$$\begin{aligned} NP^{\bullet} &= ((\mathtt{E} \to \mathtt{T}) \to \mathtt{E} \to \mathtt{T}) \to (\mathtt{E} \to \mathtt{T}) \to \mathtt{T} \\ S^{\bullet} &= ((\mathtt{Event} \to \mathtt{T}) \to \mathtt{Event} \to \mathtt{T}) \to \mathtt{T} \\ (C1/C2)^{\bullet} = (C1\backslash C2)^{\bullet} &= C2^{\bullet} \to C1^{\bullet} \end{aligned}$$

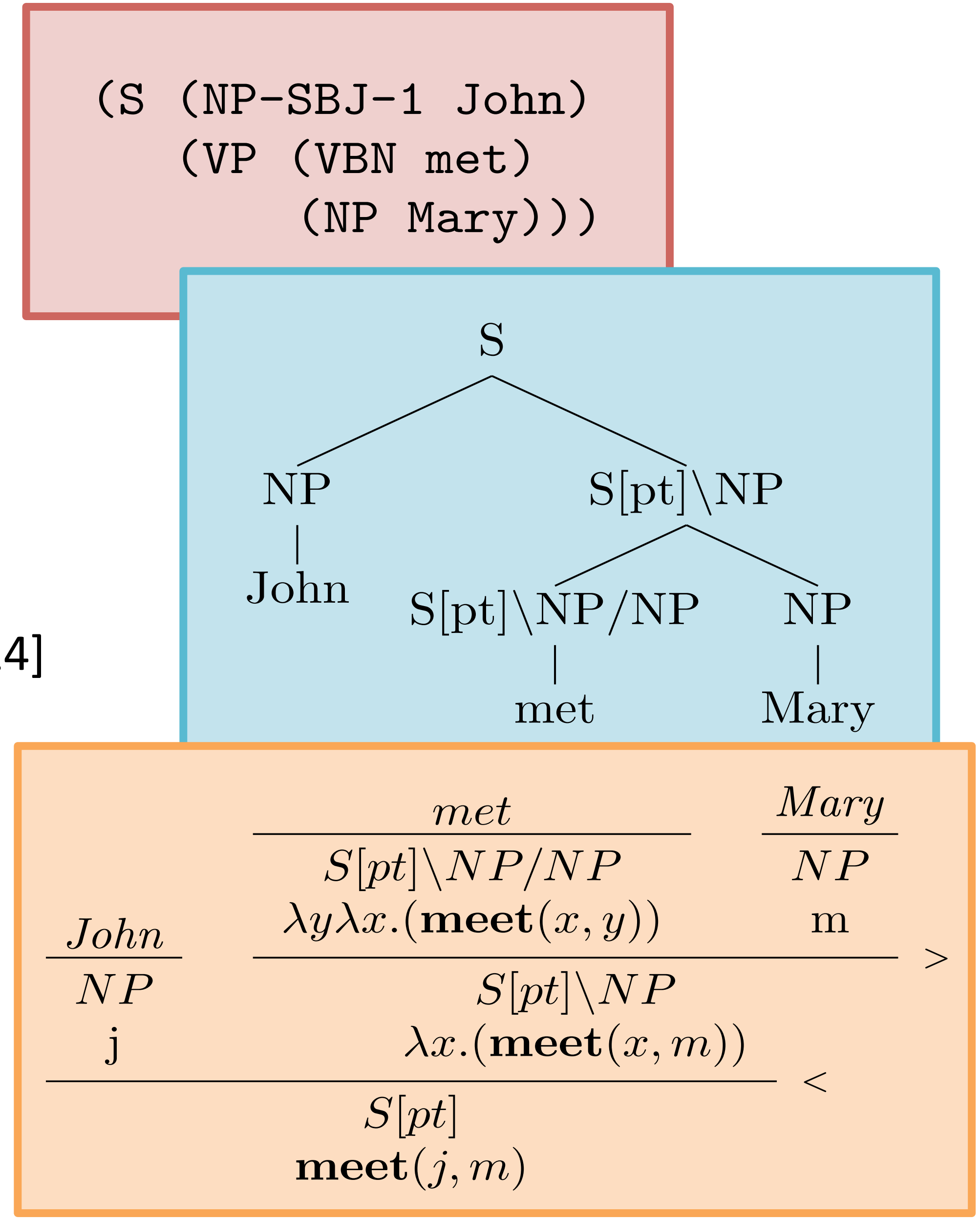
English CCG parser

✓ **Penn Treebank**

✓ **CCGBank**
[Hockenmaier and Steedman 2007]

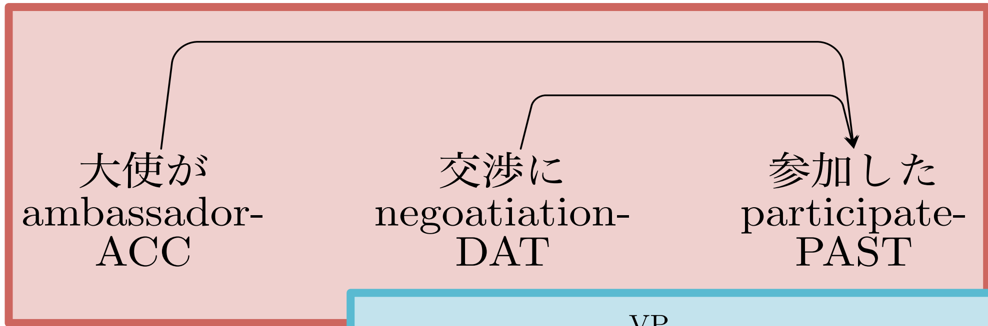
✓ **CCG parser**
- **C&C** [Curran and Clark 2007]
- **EasyCCG** [Lewis and Steedman EMNLP2014]
- **depccg** [Yoshikawa+ ACL2017]

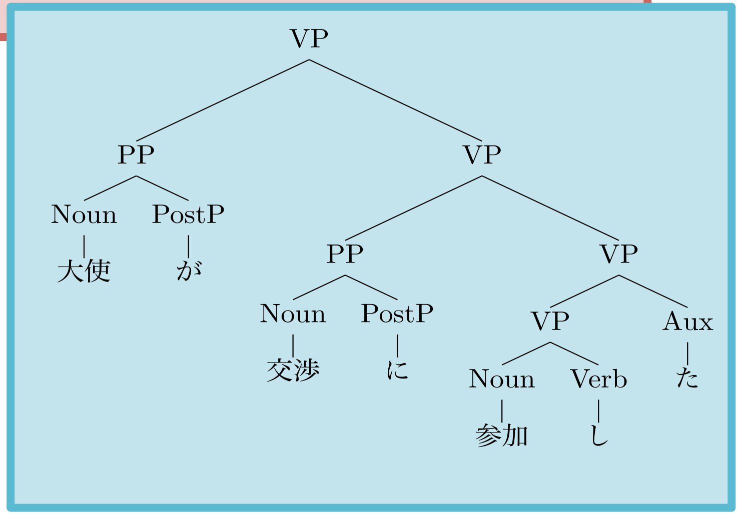
✓ **Semantic Parser**
- **Boxer** [Bos+ 2004]
- **Langpro** [Abzianidze EMNLP2015]
- **ccg2lambda** [Mineshima+ EMNLP2015]



Japanese CCG parser

- ✓ **Kyoto/NAIST Corpus**
- ✓ **Japanese CCGBank**
  [Uematsu+ ACL2013]
- ✓ **CCG parser (Jigg, depccg)**
  - **Jigg** [Noji and Miyao ACL2016]
  - **depccg** [Yoshikawa+ ACL2017]
- ✓ **Semantic parser (ccg2lambda)**
  - **ccg2lambda** [Mineshima+ EMNLP2016]





Three levels of MRs

- (Level 0 : Individual words)
- Level 1 : Predicate-Argument structure
- Level 2 : Basic logical features (negation, disjunction, etc.)
- Level 3 : Higher-order logical features

Level 1: Predicate-Argument Structure

- Who did what, where, when?
- MRs in Event semantics (Parsons, 1990):

Brutus stabbed Caesar on the street at noon.

$$\exists e(stab(e) \wedge (subj(e) = brutus) \wedge (obj(e) = caesar) \wedge (location(e) = street) \wedge (time(e) = noon))$$

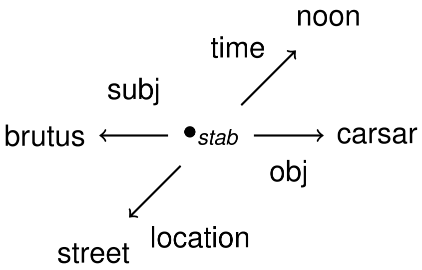
- MRs have a flat structure with:
  - $\exists$ (existential quantifier)
  - $\wedge$ (conjunction)
- Extensional descriptions of scenes or situations

Other notations: DRS and Graph

- Discourse Representation Structure (DRS) (Kamp and Reyle, 1993):

e
stab(e) subj(e) = brutus obj(e) = caesar location(e) = street time(e) = noon

- Graph notation:



- These three notations deliver the same information

The Diamond Inference

(1) Brutus stabbed Caesar on the street at noon.

- $\Rightarrow$ (2) Brutus stabbed Caesar on the street
- $\Rightarrow$ (3) Brutus stabbed Caesar at noon.
- $\Rightarrow$ (4) Brutus stabbed Caesar.

$$\begin{tikzcd} & (1) \arrow[dl] \arrow[dr] & \\ (2) \arrow[dr] & & (3) \arrow[dl] \\ & (4) & \end{tikzcd}$$

e
stab(e) subj(e) = brutus obj(e) = caesar location(e) = street time(e) = noon

The Diamond Inference

(1) Brutus stabbed Caesar on the street at noon.

$\Rightarrow$ (2) Brutus stabbed Caesar on the street

$\Rightarrow$ (3) Brutus stabbed Caesar at noon.

$\Rightarrow$ (4) Brutus stabbed Caesar.

$$
\begin{tikzcd}
 & (1) \arrow[dl] \arrow[dr] & \\
(2) \arrow[dr] & & (3) \arrow[dl] \\
 & (4) &
\end{tikzcd}
$$

e
stab(e) subj(e) = brutus obj(e) = caesar location(e) = street time(e) = noon

The Diamond Inference

(1) Brutus stabbed Caesar on the street at noon.

$\Rightarrow$ (2) Brutus stabbed Caesar on the street

$\Rightarrow$ (3) Brutus stabbed Caesar at noon.

$\Rightarrow$ (4) Brutus stabbed Caesar.

$$
\begin{tikzcd}
 & (1) \arrow[dl] \arrow[dr] & \\
(2) \arrow[dr] & & (3) \arrow[dl] \\
 & (4) &
\end{tikzcd}
$$

e
stab(e)   subj(e) = brutus   obj(e) = caesar   location(e) = street   time(e) = noon

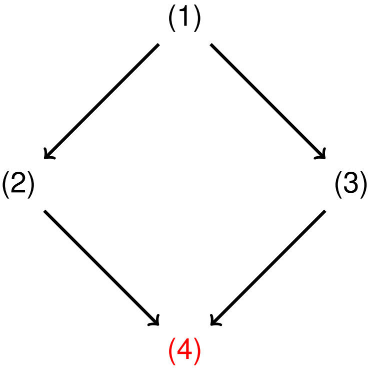
The Diamond Inference

(1) Brutus stabbed Caesar on the street at noon.

⇒ (2) Brutus stabbed Caesar on the street

⇒ (3) Brutus stabbed Caesar at noon.

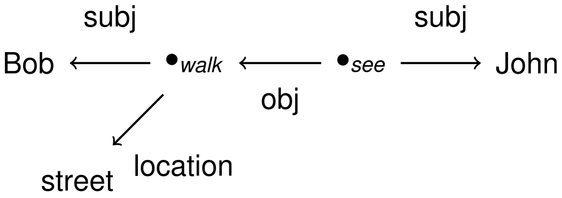
⇒ (4) Brutus stabbed Caesar.



e
stab(e)
subj(e) = brutus
obj(e) = caesar
location(e) = street
time(e) = noon

The Semantics of Voice

- Perceptual report:
  John saw Bob walking on the street.
  ⇒ Bob walked on the street.



- Active-Passive alternation:
  Brutus stabbed Caesar.
  ⇒ Caesar was stabbed by Brutus.
- Causative-inchoative alternation:
  John closed the door.
  ⇒ The door became closed.

Level 2: Basic logical features

- Add basic logical expressions:
  - *not* (negation, $\neg$)
  - *or* (disjunction, $\vee$)
  - *if* (implication, $\rightarrow$)
  - *any* (universal quantification, $\forall$)
- Indeterminate/underspecified description of a situation
- Not easy to visualize (“Draw a picture of *A man is not walking*“)

Monotonicity inference

Basic/general patterns of inferences triggered by logic features

P entails *H*

= There is no situation in which *P* is true but *H* is false.

= The information in *P* already contains the information in *H*.

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence? (Moss, 2014)

P: Some bears danced.

H1. Some animals danced.

H2. Some grizzlies danced.

H3. Some bears moved.

H4. Some bears waltzed.

Monotonicity inference

Basic/general patterns of inferences triggered by logic features

P entails *H*

$=$ There is no situation in which *P* is true but *H* is false.

$=$ The information in *P* already contains the information in *H*.

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence? (Moss, 2014)

P: Some bears danced.

$\Rightarrow$ H1. Some animals danced.

H2. Some grizzlies danced.

H3. Some bears moved.

H4. Some bears waltzed.

Monotonicity inference

Basic/general patterns of inferences triggered by logic features

P entails *H*

$=$ There is no situation in which *P* is true but *H* is false.

$=$ The information in *P* already contains the information in *H*.

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence? (Moss, 2014)

P: Some bears danced.

$\Rightarrow$ H1. Some animals danced.

$\not\Rightarrow$ H2. Some grizzlies danced.

H3. Some bears moved.

H4. Some bears waltzed.

Monotonicity inference

Basic/general patterns of inferences triggered by logic features

P entails *H*

$=$ There is no situation in which *P* is true but *H* is false.

$=$ The information in *P* already contains the information in *H*.

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence? (Moss, 2014)

P: Some bears danced.

$\Rightarrow$ H1. Some animals danced.

$\not\Rightarrow$ H2. Some grizzlies danced.

$\Rightarrow$ H3. Some bears moved.

H4. Some bears waltzed.

Monotonicity inference

Basic/general patterns of inferences triggered by logic features

P entails *H*

$=$ There is no situation in which *P* is true but *H* is false.

$=$ The information in *P* already contains the information in *H*.

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence? (Moss, 2014)

P: Some bears danced.
$\Rightarrow$ H1. Some animals danced.
$\not\Rightarrow$ H2. Some grizzlies danced.
$\Rightarrow$ H3. Some bears moved.
$\not\Rightarrow$ H4. Some bears waltzed.

Monotonicity inference

Basic/general patterns of inferences triggered by logic features

P entails *H*

$=$ There is no situation in which *P* is true but *H* is false.

$=$ The information in *P* already contains the information in *H*.

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence? (Moss, 2014)

P: Some bears danced.
$\Rightarrow$ H1. Some animals danced.
$\not\Rightarrow$ H2. Some grizzlies danced.
$\Rightarrow$ H3. Some bears moved.
$\not\Rightarrow$ H4. Some bears waltzed.

We write: Some bears$^{\uparrow}$ danced$^{\uparrow}$

NP and *VP* in *Some NP VP* are upward monotonic

Monotonicity inference

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence?

P: No bears danced.

H1. No animals danced.

H2. No grizzlies danced.

H3. No bears moved.

H4. No bears waltzed.

Monotonicity inference

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence?

P: No bears danced.
$\not\Rightarrow$ H1. No animals danced.
H2. No grizzlies danced.
H3. No bears moved.
H4. No bears waltzed.

Monotonicity inference

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence?

P: No bears danced.
$\not\Rightarrow$ H1. No animals danced.
$\Rightarrow$ H2. No grizzlies danced.
H3. No bears moved.
H4. No bears waltzed.

Monotonicity inference

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence?

P: No bears danced.
$\not\Rightarrow$ H1. No animals danced.
$\Rightarrow$ H2. No grizzlies danced.
$\not\Rightarrow$ H3. No bears moved.
H4. No bears waltzed.

Monotonicity inference

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence?

P: No bears danced.
$\not\Rightarrow$ H1. No animals danced.
$\Rightarrow$ H2. No grizzlies danced.
$\not\Rightarrow$ H3. No bears moved.
$\Rightarrow$ H4. No bears waltzed.

Monotonicity inference

- *grizzly* $\leq$ *bear* $\leq$ *animal*
- *waltz* $\leq$ *dance* $\leq$ *move*

P entails which sentence?

P: No bears danced.
$\not\Rightarrow$ H1. No animals danced.
$\Rightarrow$ H2. No grizzlies danced.
$\not\Rightarrow$ H3. No bears moved.
$\Rightarrow$ H4. No bears waltzed.

We write: No bears$^{\downarrow}$ danced$^{\downarrow}$
NP and *VP* in *No NP VP* are downward monotonic

- Logical words like *some*, *no*, *every*, *any*, *not*, *if* play a role in determining the upward/downward monotonicity.

Bare NPs

For bare NPs (NPs without determines), predicates play a crucial role.

tigress $\leq$ *tiger* $\leq$ *animal*

Tigers are striped. $\Rightarrow$ Tigresses are striped. $\not\Rightarrow$ Animals are striped.

Tigers are on the lawn. $\not\Rightarrow$ Tigresses are on the lawn. $\Rightarrow$ Animals are o

Tigers$^{\downarrow}$ are striped. (individual-level predicate)
Tigers$^{\uparrow}$ are on the lawn. (stage-level predicate)

- The basic patterns of monotonicity inferences are directly predictable from logic-based MRs.
- Upward/downward monotonicity properties follow from the properties of logical operators.
  $\exists x(\mathit{bear}^{\uparrow}(x) \wedge \mathit{dance}^{\uparrow}(x))$
  $\neg\exists x(\mathit{bear}^{\downarrow}(x) \wedge \mathit{dance}^{\downarrow}(x))$

Level 3: Advanced logic features

There are many linguistic phenomena that allegedly go beyond standard first-order logic.

- Attitudes, modals and aspectual operators.
- Generalized/proportional quantifiers
- Intensional adjectives
- Comparative and superlatives
- Other higher-order predicates

Some features:

- Introducing intensionality (involving speaker's perspectives, mental states, etc.)
- Quantifying over higher-order objects (objects other than entities)
- Not directly formalizable in first-order logics

Attitudes, modals and temporal operators

- Attitude predicates like *know* and *believe* take propositional objects as argument.
- Inferential contrast between factive predicates (eg. know) and non-factive predicate (eg. believe)

- John knows that it is raining.
  $\Rightarrow$ It is raining.
- John does not know that it is raining.
  $\Rightarrow$ It is raining.
- John believes that it is raining.
  $\not\Rightarrow$ It is raining.
- John does not believe that it is raining.
  $\not\Rightarrow$ It is raining.

- modals: *likely*, *probably*, *might*, *must*, *can*. etc.
- aspectual operators: progressives, perfectives, etc.

Generalized quantifiers

- *Most*, *half of*, *70% of* ...

Most students smoked.	$\not\Rightarrow \not\Leftarrow$	Most female student smoked.
Most students smoked.	$\Longleftarrow$	Most student smoked in a building.

- But these quantifiers are known to be not first-orderizable (Barwise and Cooper, 1981)

Adjectives: subsective and non-subsective

Subsective (intersective) adjective

- Dumbo is a small elephant. $\mathit{small}(\mathit{dumbo}) \wedge \mathit{elephant}(\mathit{dumbo})$
  $\Rightarrow$ Dumbo is an elephant. $\mathit{elephant}(\mathit{dumbo})$

Non-subsective adjective

- This is a fake diamond.
  $\not\Rightarrow$ This is a diamond.
  $\Rightarrow$ This is not a diamond.

Comparatives

- Alice is taller than Bob.
  $\not\Rightarrow$ Alice is tall.

- Alice is taller than Bob.
- Bob is tall.
  $\Rightarrow$ Alice is tall.

- Alice is taller than Bob.
- Bob is taller than Carol.
  $\Rightarrow$ Alice is taller than Carol.

Question:

- What are proper MRs for adjective constructions that are suitable to efficient inferences?
- How to give a compositional semantics of predicates *tall* and *taller* (how the meanings of *tall* and *taller* are related to each other?)

Some higher-order predicates

- Higher-order predicates that apply to objects other than entities: *rise*, *change*, *decrease*

- The price of gasoline is rising.
- The price of gasoline is 1,000 dollars.

  $\not\Rightarrow$ 1,000 dollars are rising.

Logic-based Meaning Representations

Natural Logic

- formalizes inferences with surface form
- ▲ only allows single premise inferences (mononicity inference)

First-order logic (FOL)

- efficient provers exist
- dominate computational linguistics
- ▲ limited expressive power

Higher-order logic (HOL)

- high expressive power
- dominate formal semantics
- ▲ no general-purpose efficient prover exists

more efficient
less expressive ⟷ less efficient
more expressive

MacCartney (2009)

Boxer (Bos 2008)

Mineshima et al. (2015)

Logic-based Meaning Representations

Natural Logic

- formalizes inferences with surface form
- ▲ only allows single premise inferences (mononicity inference)

First-order logic (FOL)

- efficient provers exist
- dominate computational linguistics
- ▲ limited expressive power

Higher-order logic (HOL)

- high expressive power
- dominate formal semantics
- ▲ no general-purpose efficient prover exists

more efficient
less expressive

less efficient
more expressive

MacCartney (2009)

Boxer (Bos 2008)

Mineshima et al. (2015)

Logic-based Meaning Representations

Natural Logic	First-order logic (FOL)	Higher-order logic (HOL)
• formalizes inferences with surface form ▲ only allows single premise inferences (mononicity inference)	• efficient provers exist • dominate computational linguistics ▲ limited expressive power	• high expressive power • dominate formal semantics ▲ no general-purpose efficient prover exists
more efficient less expressive	⟷	less efficient more expressive
MacCartney (2009)	Boxer (Bos 2008)	Mineshima et al. (2015)

HOL as representation language

Higher-order constructions in natural languages

1. Generalized quantifiers
   Most students work $\rightsquigarrow$ $\mathit{most}(\lambda.\mathit{student}(x), \lambda x.\mathit{work}(x))$
2. Modals
   John might come $\rightsquigarrow$ $\mathit{might}(\mathit{come}(j))$
3. Veridical and anti-veridical predicates
   Someone managed to come $\rightsquigarrow$ $\exists x(\mathit{manage}(x, \mathit{come}(x)))$
   Someone failed to come $\rightsquigarrow$ $\exists x(\mathit{fail}(x, \mathit{come}(x)))$
4. Attitude verbs
   John knows that some student came. $\rightsquigarrow$
   $$\mathit{know}(j, \exists x(\mathit{student}(x) \wedge \mathit{come}(x)))$$

- Higher-order inference system implemented in Coq (Mineshima et al., 2015)
- Alternative: first-order decomposition/reification (Hobbs, 1985)

Natural Language Inference (Recognizing Textual Entailment, RTE)

- Does P entail H?

P Most cities in Japan prohibit smoking in restaurants.
H Some cities in Japan do not allow smoking in public spaces.

Yes (entail)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many applications in NLP
  - Question Answering,
  - Text Summarization
  - Fact validation/checking
  - etc.

Datasets for Recognizing Textual Entailment (RTE)

- English:

Dataset	Size	Crowdsourcing
FraCaS (Cooper et al., 1994)	346	
PASCAL-RTE1–5 (Dagan et al. 2006)	7K	
SICK (Marelli et al., 2014)	10K	$\surd$
SNLI (Bowman et al., 2015)	570K	$\surd$
MultiNLI (Williams et al. 2017)	432K	$\surd$

- Japanese:

Dataset	Size	Crowdsourcing
JSeM	780	
NTCIR RITE 1–2	1,800	
Kyoto RTE dataset	2,471	

FraCaS (Cooper et al. 1996)

- Created by linguists in 1990s.
- Size: 346 problems
- The inferences are divided into nine sections in terms of linguistic phenomena:
  - Generalized quantifier, Plurals, Nominal anaphora, Ellipsis, Adjective, Comparatives, Temporal reference, Verbs, Attitudes
- Contains lots of logical expressions (at Level 2 and Level 3)
- Lexical and world knowledge is mostly excluded
- Contains multiple-premise inferences

# premise	# problem	
1	192	55.5%
2	122	35.3%
3	29	8.4%
4	2	0.6%
5	1	0.3%

FraCaS: Examples

- The XML format was created by Bill MacCartney
  https://nlp.stanford.edu/w̃cmac/downloads/

fracas-038 (Generalized quantifier) label: no (contradiction)
P: No delegate finished the report.
H: Some delegate finished the report on time.

fracas-084 (Plural) label: yes (entailment)
P: Either Smith, Jones or Anderson signed the contract.
H: If Smith and Anderson did not sign the contract, Jones signed the contract.

fracas-134 (Nominal Anaphora) label: yes (entailment)
P1: Every customer who owns a computer has a service contract for it.
P2: MFI is a customer that owns exactly one computer.
H: MFI has a service contract for all its computers.

Japanese Semantics Test Suite (JSeM)

Kawazoe et al. (2015)

`http://researchmap.jp/community-inf/JSeM/`

- Translation of FraCaS (624 problems) and Japanese original ones (166 problems)
- Each problem is tagged with:
  - phenomena type (quantifier, adjective, negation, etc.)
  - inference type (logical entailment, presupposition)
- single-premised (66%) and multi-premised (34%) problems

jsem-id:1	answer: yes	inference type: entailment	phenomena: Generalized Quantifier, conservativity
	linked to: fracas-001	literal translation?: yes	same phenomena?: unknown

P1

script あるイタリア人が世界最高のテノール歌手になった。

English An Italian became the world's greatest tenor.

H

script 世界最高のテノール歌手になったイタリア人がいた。

English There was an Italian who became the world's greatest tenor.

SICK (Sentences Involving Compositional Knowldedge)

SemEval14, Marelli et al. (2013)

- Size: $4,500/500/4,927$ for training, dev. and testing.
- Premise: taken from image captions in Flickr30k Corpus
- Hyphothesis and Label: crowdsourcing and expert-check
- contains only single-premise inferences
- contains logical expressions at Level 2 (negation, disjunction, quantifiers)
- Both word-level and phrase-level paraphrases are required

SICK: Examples

SICK-506 (label: no)
P: A man wearing a dyed black shirt is sitting at the table and laughing.
H: There is no man wearing a shirt dyed black, sitting at the table and laughing.

SICK-718 (label: unknown)
P: A few men in a competition are running outside.
H: A few men are running competitions outside.

SICK-3156 (label: yes)
P: A man is cutting a box.
H: A box is being cut by a man.

SICK-3668 (label: yes)
P: A man is strolling in the rain.
H: A man is walking in the rain.

SNLI

Bowman et al. (2015)

- The Stanford Natural Language Inference (SNLI) Corpus
- P: taken from image captions in Flickr30k Corpus
- H and Label: crowdsourcing
- contains only single-premise inferences
- sentences are confined to descriptions of scenes, not containing logical features (limited to Level 1)
- largely limited to simple lexical inferences

label: entailment
P: A white dog with long hair jumps to catch a red and green toy.
H: An animal is jumping to catch an object.

MultiNLI

Williams et al. (2017)

- The Multi-Genre Natural Language Inference (MultiNLI)

genre: Fiction, answer: entailment
P: He turned and saw Jon sleeping in his half-tent.
H: He saw Jon was asleep.

genre: telephone, answer: contradiction
P: someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny
H: No one noticed and it wasn't funny at all.

- A set of linguistic phenomena tags are automatically assigned to the development set (10K sentences):
  - quantifiers, belief verbs, time terms, conditionals, etc.

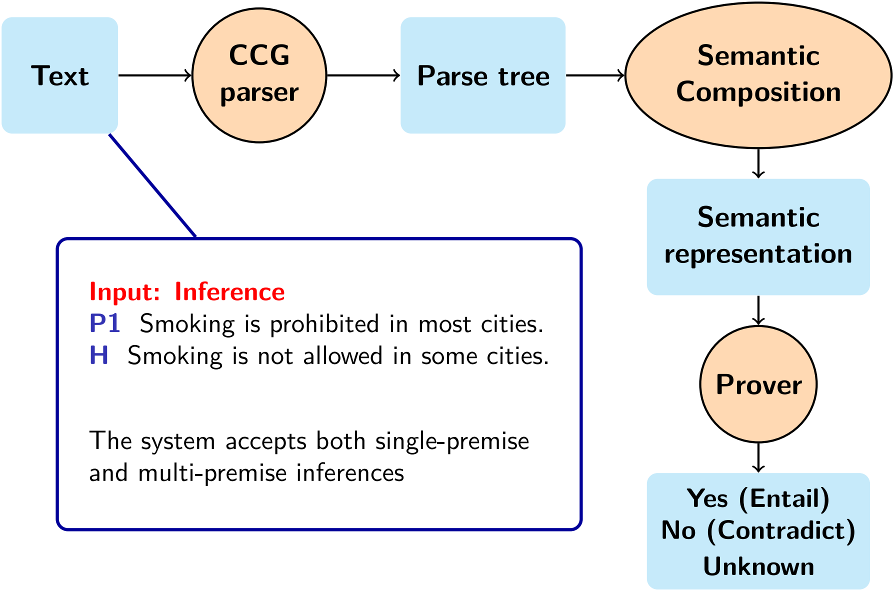
Summary

- Compositional Semantics:
  - Meaning composition via CCG and Lambda Calculus
- Meaning Representations:
  - Three levels of MRs for semantic composition: Predicate-Argument Structure, Basic Logics and beyond
  - Event Semantics, First-order logic, and Higher-order logic
- Inference: RTE datasets

Introduction to ccg2lambda

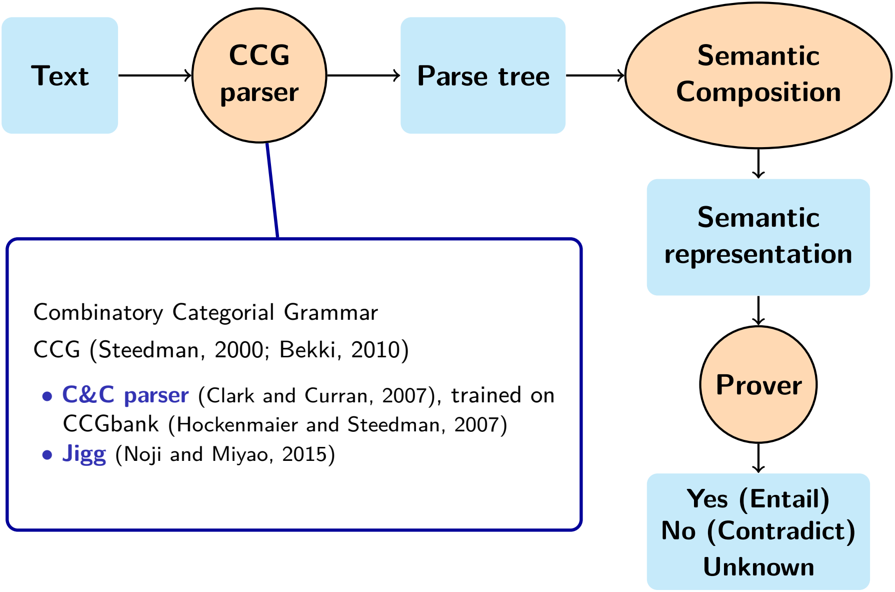
ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



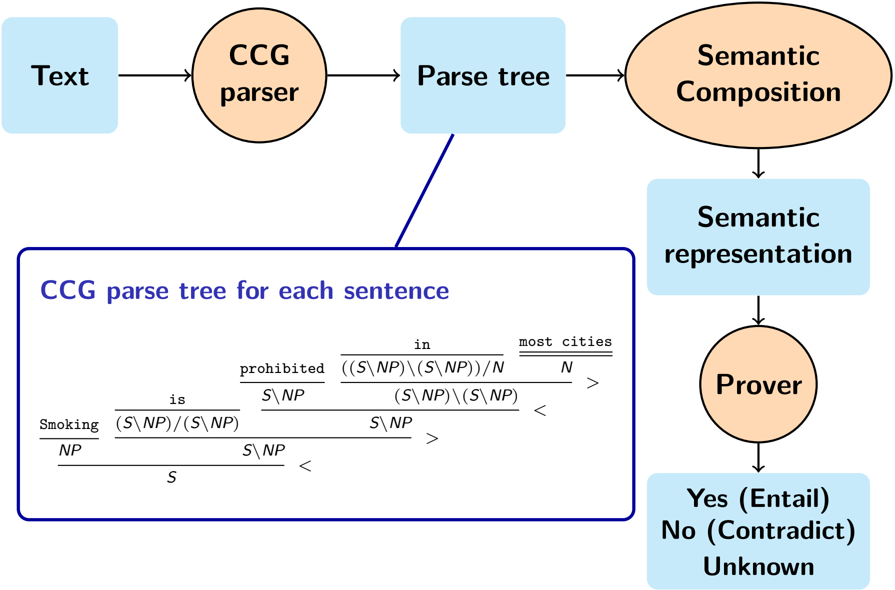
ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



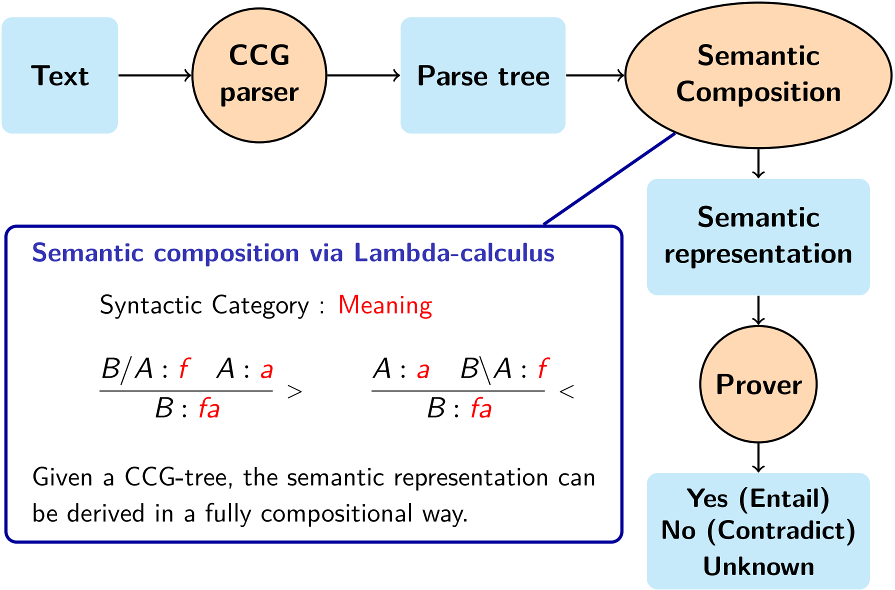
ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



Semantic composition via Lambda-calculus

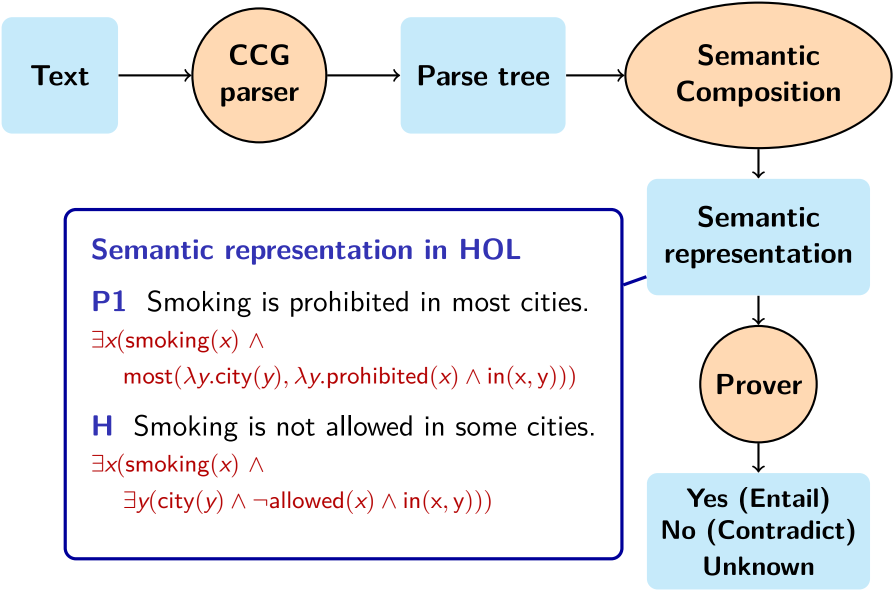
Syntactic Category : Meaning

$$\frac{B/A : f \quad A : a}{B : fa} > \qquad \frac{A : a \quad B\backslash A : f}{B : fa} <$$

Given a CCG-tree, the semantic representation can be derived in a fully compositional way.

ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



Semantic representation in HOL

P1 Smoking is prohibited in most cities.

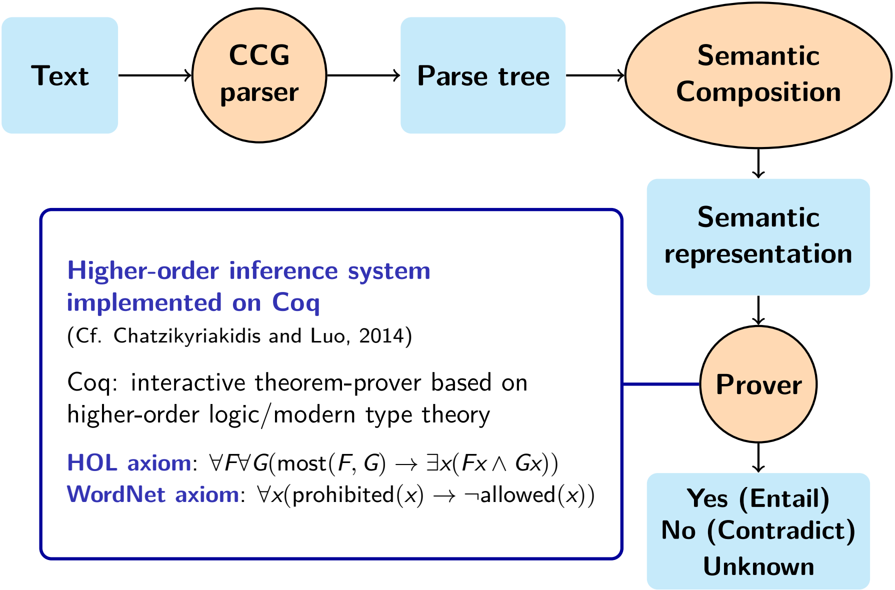
$\exists x(\mathrm{smoking}(x) \wedge$
$\quad \mathrm{most}(\lambda y.\mathrm{city}(y), \lambda y.\mathrm{prohibited}(x) \wedge \mathrm{in}(\mathrm{x}, \mathrm{y})))$

H Smoking is not allowed in some cities.

$\exists x(\mathrm{smoking}(x) \wedge$
$\quad \exists y(\mathrm{city}(y) \wedge \neg \mathrm{allowed}(x) \wedge \mathrm{in}(\mathrm{x}, \mathrm{y})))$

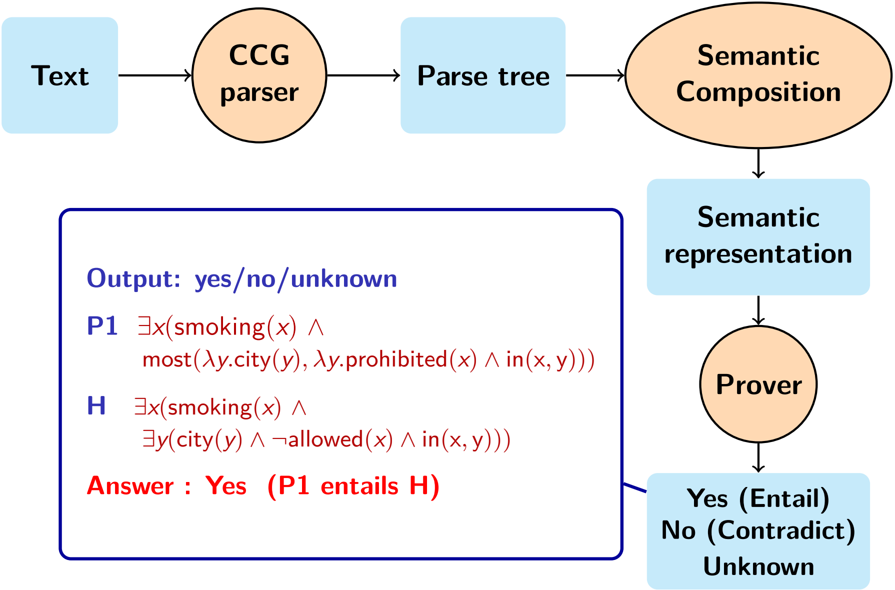
ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



ccg2lambda: Semantic Parser and Inference System

https://github.com/mynlp/ccg2lambda



Semantic composition on CCG tree

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\text{Every}\\ NP/N\end{matrix} \quad \begin{matrix}\text{student}\\ N\end{matrix}}{NP}> \quad \dfrac{\begin{matrix}\text{who}\\ (NP\backslash NP)/(S\backslash NP)\end{matrix} \quad \begin{matrix}\text{works}\\ S\backslash NP\end{matrix}}{NP\backslash NP}>}{NP}< \quad \begin{matrix}\text{comes}\\ S\backslash NP\end{matrix}}{S}$$

- Syntactic categories and rules indicate composition.

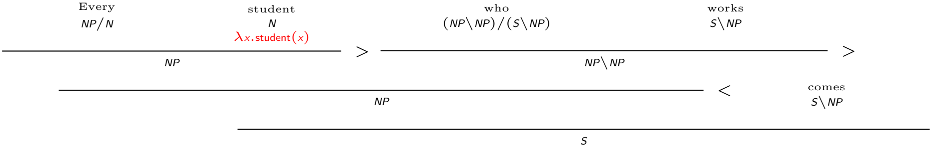
Semantic composition on CCG tree

$$
\begin{array}{cccc}
\text{Every} & \text{student} & \text{who} & \text{works} \\
NP/N & N & (NP\backslash NP)/(S\backslash NP) & S\backslash NP \\
 & \color{red}{(\lambda Ex.E(x))(\mathsf{student})} & & \\
\hline
\multicolumn{2}{c}{NP \quad >} & \multicolumn{2}{c}{NP\backslash NP \quad >} \\
\hline
\multicolumn{4}{c}{NP \quad <} \\
\end{array}
\qquad
\begin{array}{c}
\text{comes} \\
S\backslash NP
\end{array}
$$

$$
\overline{\qquad\qquad\qquad S \qquad\qquad\qquad}
$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.

Semantic composition on CCG tree



- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.

Semantic composition on CCG tree

$$
\begin{array}{cccc}
\text{Every} & \text{student} & \text{who} & \text{works} \\
NP/N & N & (NP\backslash NP)/(S\backslash NP) & S\backslash NP \\
 & \lambda x.\mathsf{student}(x) & & \lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{work}(x)) \\
\hline
\multicolumn{2}{c}{NP \quad >} & \multicolumn{2}{c}{NP\backslash NP \quad >}
\end{array}
$$

$$
\begin{array}{cc}
 & \text{comes} \\
 & S\backslash NP \\
NP \quad < & \lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{come}(x)) \\
\hline
\multicolumn{2}{c}{S}
\end{array}
$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.

Semantic composition on CCG tree

$$
\begin{array}{cccc}
\text{Every} & \text{student} & \text{who} & \text{works} \\
NP/N & N & (NP\backslash NP)/(S\backslash NP) & S\backslash NP \\
\lambda FGH.\forall x(Fx \wedge Gx \rightarrow Hx) & \lambda x.\mathsf{student}(x) & \lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \wedge Fx)) & \lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{work}(x)) \\
\hline
\multicolumn{2}{c}{NP \quad >} & \multicolumn{2}{c}{NP\backslash NP \quad >}
\end{array}
$$

$$
\begin{array}{cc}
\begin{array}{c} NP \quad NP\backslash NP \\ \hline NP \quad < \end{array} &
\begin{array}{c} \text{comes} \\ S\backslash NP \\ \lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{come}(x)) \end{array} \\
\hline
\multicolumn{2}{c}{S}
\end{array}
$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.

Semantic composition on CCG tree

Every: NP/N : $\lambda FGH.\forall x(Fx \wedge Gx \rightarrow Hx)$

student: N : $\lambda x.\mathsf{student}(x)$

$$\dfrac{NP/N \quad N}{NP : \lambda GH.\forall x(\mathsf{student}(x) \wedge Gx \rightarrow Hx)}>$$

who: $(NP\backslash NP)/(S\backslash NP)$: $\lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \wedge Fx))$

works: $S\backslash NP$: $\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{work}(x))$

$$\dfrac{(NP\backslash NP)/(S\backslash NP) \quad S\backslash NP}{NP\backslash NP}>$$

$$\dfrac{NP \quad NP\backslash NP}{NP}<$$

comes: $S\backslash NP$: $\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{come}(x))$

$$\dfrac{NP \quad S\backslash NP}{S}$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.
- Semantic composition from leaves to root.

Semantic composition on CCG tree

$$
\dfrac{
  \dfrac{
    \dfrac{\text{Every} \atop {NP/N \atop \lambda FGH.\forall x(Fx \land Gx \rightarrow Hx)} \quad \text{student} \atop {N \atop \lambda x.\mathrm{student}(x)}}{NP \atop \lambda GH.\forall x(\mathrm{student}(x) \land Gx \rightarrow Hx)} >
    \quad
    \dfrac{\text{who} \atop {(NP\backslash NP)/(S\backslash NP) \atop \lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \land Fx))} \quad \text{works} \atop {S\backslash NP \atop \lambda Q.Q(\lambda x.\mathrm{True})(\lambda x.\mathrm{work}(x))}}{NP\backslash NP \atop \lambda QF.Q(\lambda x.(\mathrm{work}(x) \land Fx))} >
  }{NP \atop \lambda FH.\forall x(\mathrm{student}(x) \land \mathrm{work}(x) \land Fx \rightarrow Hx)} <
  \quad
  \text{comes} \atop {S\backslash NP \atop \lambda Q.Q(\lambda x.\mathrm{True})(\lambda x.\mathrm{come}(x))}
}{S \atop \forall x(\mathrm{student}(x) \land \mathrm{work}(x) \land \mathrm{True} \rightarrow \mathrm{come}(x))}
$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.
- Semantic composition from leaves to root.
- Logical meaning representation of the sentence at the root.

Lexical entries

1. For **closed words**: lexical entries directly assigned to surface form (a limited number of grammatical and logical expressions): 80 entries

Example

- **category**: NP/N
- **semantics**: $\lambda F \lambda G \lambda H. \forall x(Fx \wedge Gx \rightarrow H)$
- **surf**: every

2. For **open words**: schematic lexical entry (semantic templates) assigned to syntactic categories: 57 entries

Example

- **category**: N
- **semantics**: $\lambda E \lambda x.\mathrm{E}(x)$

"E" is a position in which a particular lexical item appears.

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

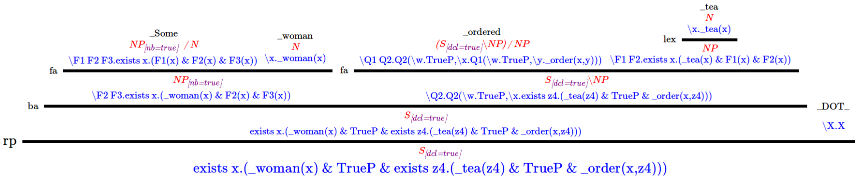
- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`

```xml
<?xml version='1.0' encoding='utf-8'?>
<root>
  <document>
    <sentences>
      <sentence>
        <tokens>
          <token id="t0_0" pos="DT"  cat="NP[nb]/N"     surf="Some"    base="some"/>
          <token id="t0_1" pos="NN"  cat="N"            surf="woman"   base="woman"/>
          <.../>
        </tokens>
        <ccg root="s0_sp0" id="s0_ccg0">
          <span id="s0_sp0" child="s0_sp1 s0_sp9" category="S[dcl=true]"      rule="rp"/>
          <span id="s0_sp1" child="s0_sp2 s0_sp5" category="S[dcl=true]"      rule="ba"/>
          <...>
        </ccg>
        <semantics status="success" root="s0_sp0">
          <span id="s0_sp0" child="s0_sp1 s0_sp9"
                sem="exists x.(_woman(x) & TrueP & exists z1.(_tea(z1) & TrueP & _order(x,z1)))"/>
          <span id="s0_sp4" type="_woman : Entity -> Prop"
                sem="\x._woman(x)"/>
          <...>
        </semantics>
      </sentence>
    </sentences>
  </document>
</root>
```

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`



ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`
  - `# python prove.py semantics.xml`

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`
  - `# python prove.py semantics.xml`
- Easy to extend (declarative).

$$\begin{array}{l} -\ \texttt{semantics} : \lambda\text{-formula} \\ \quad \texttt{category} : \mathrm{syntactic\_category} \\ \quad \texttt{cond}_2 : \mathrm{value}_2 \\ \quad \texttt{cond}_\mathrm{i} : \mathrm{value}_\mathrm{i} \end{array}$$

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`
  - `# python prove.py semantics.xml`
- Easy to extend (declarative).
- Easy to process (XML output).

Recognizing Textual Entailment

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many application areas (Question Answering, Machine Translation, etc.)

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many application areas (Question Answering, Machine Translation, etc.)

- relevant factors:
  1. syntax
  2. logical words: *most, not, some, every*

Logical/ Compositional semantics

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many application areas (Question Answering, Machine Translation, etc.)

- relevant factors:

1. syntax
2. logical words: *most*, *not*, *some*, *every*

Logical/ Compositional semantics

3. content words:

restaurant → *public_space*
prohibited → ¬ *allowed*

Lexical Knowledge

Introducing Lexical Knowledge

Introduction

Logic sometimes is not enough

T: *men are sawing logs.*

$\exists x.(\mathrm{man}(x) \wedge \exists y.(\mathrm{log}(y) \wedge \mathrm{saw}(x, y))$

H: *men are cutting wood.*

$\exists x.(\mathrm{man}(x) \wedge \exists y.(\mathrm{wood}(y) \wedge \mathrm{cut}(x, y))$

Introduction

Logic sometimes is not enough

T: *men are sawing logs.*

$$\exists x.(\mathrm{man}(x) \wedge \exists y.(\mathrm{log}(y) \wedge \mathrm{saw}(x,y))$$

H: *men are cutting wood.*

$$\exists x.(\mathrm{man}(x) \wedge \exists y.(\mathrm{wood}(y) \wedge \mathrm{cut}(x,y))$$

Method: to inject lexical knowledge into the proof.

- Word relations can be found in ontologies (e.g. WordNet, etc.)

$$\forall x \forall y.\mathrm{saw}(x,y) \rightarrow \mathrm{cut}(x,y)$$

$$\forall x.\mathrm{log}(x) \rightarrow \mathrm{wood}(x)$$

Naïve injection of lexical knowlege

Running example:

$$\exists x_1 v_1(\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$$

T: *A black and white dog naps .*

H: *A black and white dog sleeps .*

$$\exists x_2 v_2(\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$$

- Obtain semantic representation.

Naïve injection of lexical knowlege

Running example:

$$\exists x_1 v_1(\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$$

T: *A black and white dog naps.*

H: *A black and white dog sleeps.*

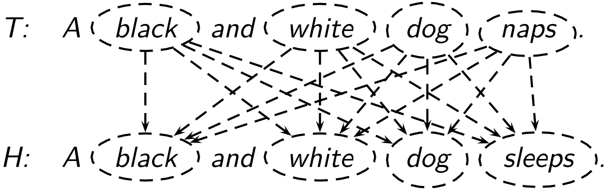
$$\exists x_2 v_2(\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$$

- Identify content/interesting words.

Naïve injection of lexical knowlege

Running example:

$$\exists x_1 v_1(\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$$



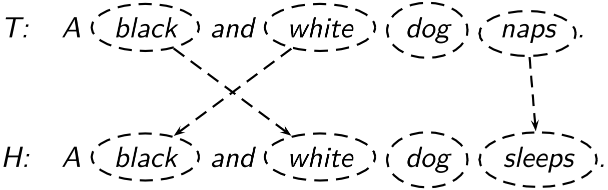
$$\exists x_2 v_2(\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$$

- Enumerate possible relations.

Naïve injection of lexical knowlege

Running example:

$$\exists x_1 v_1(\text{dog}(x_1) \wedge \text{white}(x_1) \wedge \text{black}(x_1) \wedge \text{nap}(v_1) \wedge \text{Subj}(v_1) = x_1)$$



$$\exists x_2 v_2(\text{dog}(x_2) \wedge \text{white}(x_2) \wedge \text{black}(x_2) \wedge \text{sleep}(v_2) \wedge \text{Subj}(v_2) = x_2)$$

- Select/predict relations according to ontology or classifier:
  - $\forall x.\text{black}(x) \rightarrow \neg\ \text{white}(x)$
  - $\forall x.\text{white}(x) \rightarrow \neg\ \text{black}(x)$
  - $\forall v.\text{nap}(v) \rightarrow \text{sleep}(v)$

Naïve injection of lexical knowlege

Running example:

$$\exists x_1 v_1 (\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$$

T: *A black and white dog naps.*

H: *A black and white dog sleeps.*

$$\exists x_2 v_2 (\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$$

- Insert knowledge, run proof.
  - ... *and possibly get the wrong answer.*
  - This problem is aggravated for longer sentences.

Proving strategy and Axiom construction

T : $\exists x_1 v_1(\text{dog}(x_1) \wedge \text{white}(x_1) \wedge \text{black}(x_1) \wedge \text{nap}(v_1) \wedge \text{Subj}(v_1) = x_1)$
H : $\exists x_2 v_2(\text{dog}(x_2) \wedge \text{white}(x_2) \wedge \text{black}(x_2) \wedge \text{sleep}(v_2) \wedge \text{Subj}(v_2) = x_2)$

step 0

p_1: $\text{dog}(x_1)$
p_2: $\text{white}(x_1)$
p_3: $\text{black}(x_1)$
p_4: $\text{Subj}(v_1) = x_1$
p_5: $\text{nap}(v_1)$

g_1: $\text{dog}(x_2)$
g_2: $\text{white}(x_2)$
g_3: $\text{black}(x_2)$
g_4: $\text{Subj}(v_2) = x_2$
g_5: $\text{sleep}(v_2)$

- Decompose T and H into:
  - Pool of logical premises P.
  - List of sub-goals G.

Proving strategy and Axiom construction

$T : \exists x_1 v_1(\text{dog}(x_1) \wedge \text{white}(x_1) \wedge \text{black}(x_1) \wedge \text{nap}(v_1) \wedge \text{Subj}(v_1) = x_1)$

$H : \exists x_2 v_2(\text{dog}(x_2) \wedge \text{white}(x_2) \wedge \text{black}(x_2) \wedge \text{sleep}(v_2) \wedge \text{Subj}(v_2) = x_2)$

step 1

p_1: dog(x_1)
p_2: white(x_1)
p_3: black(x_1)
p_4: Subj$(v_1) = x_1$
p_5: nap(v_1)

~~g_1: dog(x_1)~~
~~g_2: white(x_1)~~
~~g_3: black(x_1)~~
g_4: Subj$(v_2) = x_1$
g_5: sleep(v_2)

- Decompose T and H into:
  - Pool of logical premises P.
  - List of sub-goals G.
- Variable unification $x_2 := x_1$.
  - Prove g_1, g_2 and g_3 ...
  - ... using p_1, p_2 and p_3.

Proving strategy and Axiom construction

$T : \exists x_1 v_1 (\text{dog}(x_1) \land \text{white}(x_1) \land \text{black}(x_1) \land \text{nap}(v_1) \land \text{Subj}(v_1) = x_1)$

$H : \exists x_2 v_2 (\text{dog}(x_2) \land \text{white}(x_2) \land \text{black}(x_2) \land \text{sleep}(v_2) \land \text{Subj}(v_2) = x_2)$

step 2

p_1: dog(x_1)
p_2: white(x_1)
p_3: black(x_1)
p_4: Subj$(v_1) = x_1$
p_5: nap(v_1)

~~g_1: dog(x_1)~~
~~g_2: white(x_1)~~
~~g_3: black(x_1)~~
~~g_4: Subj$(v_1) = x_1$~~
g_5: sleep(v_1)

- Decompose T and H into:
  - Pool of logical premises P.
  - List of sub-goals G.
- Variable unification $x_2 := x_1$.
  - Prove g_1, g_2 and g_3 ...
  - ... using p_1, p_2 and p_3.
- Variable unification $v_2 := v_1$.
  - Prove g_4 using p_4.

Proving strategy and Axiom construction

$T : \exists x_1 v_1 (\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$
$H : \exists x_2 v_2 (\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$

step 3

p_1: dog(x_1)
p_2: white(x_1)
p_3: black(x_1)
p_4: Subj$(v_1) = x_1$
p_5: nap(v_1)

~~g_1: dog(x_1)~~
~~g_2: white(x_1)~~
~~g_3: black(x_1)~~
~~g_4: Subj$(v_1) = x_1$~~
~~g_5: sleep(v_1)~~

- Decompose T and H into:
  - Pool of logical premises P.
  - List of sub-goals G.
- Variable unification $x_2 := x_1$.
  - Prove g_1, g_2 and g_3 ...
  - ... using p_1, p_2 and p_3.
- Variable unification $v_2 := v_1$.
  - Prove g_4 using p_4.
- Inject axiom $\forall v.\mathsf{nap}(v) \rightarrow \mathsf{sleep}(v)$.
  - nap(v_1) and sleep(v_1) share variable.
  - nap-sleep $\in$ WordNet.
  - Continue proof.

Proving strategy and Axiom construction

$T: \exists x_1 v_1 (\mathsf{dog}(x_1) \wedge \mathsf{white}(x_1) \wedge \mathsf{black}(x_1) \wedge \mathsf{nap}(v_1) \wedge \mathsf{Subj}(v_1) = x_1)$

$H: \exists x_2 v_2 (\mathsf{dog}(x_2) \wedge \mathsf{white}(x_2) \wedge \mathsf{black}(x_2) \wedge \mathsf{sleep}(v_2) \wedge \mathsf{Subj}(v_2) = x_2)$

step 3

p_1: dog(x_1)
p_2: white(x_1)
p_3: black(x_1)
p_4: Subj$(v_1) = x_1$
p_5: nap(v_1)

~~g_1: dog(x_1)~~
~~g_2: white(x_1)~~
~~g_3: black(x_1)~~
~~g_4: Subj$(v_1) = x_1$~~
~~g_5: sleep(v_1)~~

- Decompose T and H into:
  - Pool of logical premises P.
  - List of sub-goals G.
- Variable unification $x_2 := x_1$.
  - Prove g_1, g_2 and g_3 ...
  - ... using p_1, p_2 and p_3.
- Variable unification $v_2 := v_1$.
  - Prove g_4 using p_4.
- Inject axiom $\forall v.\mathsf{nap}(v) \rightarrow \mathsf{sleep}(v)$.
  - nap(v_1) and sleep(v_1) share variable.
  - nap-sleep $\in$ WordNet.
  - Continue proof.

Proving strategy and Axiom construction

$T : \exists x_1 v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge nap(v_1) \wedge Subj(v_1) = x_1)$

$H : \exists x_1 v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge sleep(v_1) \wedge Subj(v_1) = x_1)$

Proving strategy and Axiom construction

$T : \exists x_1\, v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge nap(v_1) \wedge Subj(v_1) = x_1)$

$H : \exists x_1\, v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge sleep(v_1) \wedge Subj(v_1) = x_1)$

- Variable unification from proof...

Proving strategy and Axiom construction

$$T : \exists x_1 v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge nap(v_1) \wedge Subj(v_1) = x_1)$$

$$H : \exists x_1 v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge sleep(v_1) \wedge Subj(v_1) = x_1)$$

- Variable unification from proof...
  - Defines an alignment between logic predicates.
  - Most theorem provers perform backtracking in the search of best alignment.

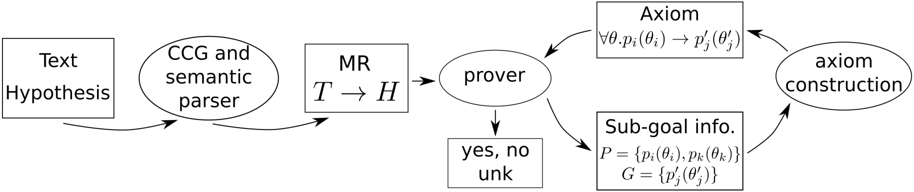
Proving strategy and Axiom construction

$T : \exists x_1 v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge nap(v_1) \wedge Subj(v_1) = x_1)$

$H : \exists x_1 v_1 (dog(x_1) \wedge white(x_1) \wedge black(x_1) \wedge sleep(v_1) \wedge Subj(v_1) = x_1)$

- Variable unification from proof...
  - Defines an alignment between logic predicates.
  - Most theorem provers perform backtracking in the search of best alignment.
- Better identify logic/textual relations:
  - $\forall v.\mathrm{nap}(v) \rightarrow \mathrm{sleep}(v)$.

System



1. Tokenize T and H.
2. Syntactic parsing with C&C and EasyCCG.
3. Obtain Meaning Representations with ccg2lambda.
4. Monitor proof and inject axioms on-demand:
   - synonymy (e.g. house $\to$ home),
   - hypernymy (e.g. sea $\to$ water),
   - adjectival similarity (e.g. huge $\to$ big),
   - derivationally related forms (e.g. accommodating $\to$ accommodation),
   - inflection relations (e.g. wooded $\to$ wood),
   - antonymy relations (e.g. big $\to$ $\neg$small).

Evaluation

SICK dataset

- Size: $4,500/500/4,927$ for training, dev. and testing.
- Label distribution: $.29/.15/.56$ for yes/no/unk.
- About $212,000$ running words.
- Average premise and conclusion length: 10.6.
- No parameter estimation.

Evaluation

SICK dataset

- Size: $4,500/500/4,927$ for training, dev. and testing.
- Label distribution: $.29/.15/.56$ for yes/no/unk.
- About $212,000$ running words.
- Average premise and conclusion length: 10.6.
- No parameter estimation.

Examples:

Problem ID	T-H pairs	Entailment
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes
4114	T: *There is no man eating food .* H: *A man is eating a pizza .*	No
718	T: *A few men in a competition are running outside .* H: *A few men are running competitions outside .*	Unknown

Evaluation

Results:

System	Prec.	Rec.	Acc.
Baseline (majority)	–	–	56.69

Evaluation

Results:

System	Prec.	Rec.	Acc.
Baseline (majority)	–	–	56.69
MLN	–	–	73.40
Nutcracker	–	–	74.30
Nutcracker-WN	–	–	77.50
Nutcracker-WN-PPDB	–	–	78.60
MLN-WN-PPDB	–	–	80.40
LangPro Hybrid-800	97.95	58.11	81.35
The Meaning Factory	93.63	60.64	81.60

Evaluation

Results:

System	Prec.	Rec.	Acc.
Baseline (majority)	–	–	56.69
MLN	–	–	73.40
Nutcracker	–	–	74.30
Nutcracker-WN	–	–	77.50
Nutcracker-WN-PPDB	–	–	78.60
MLN-WN-PPDB	–	–	80.40
LangPro Hybrid-800	97.95	58.11	81.35
The Meaning Factory	93.63	60.64	81.60
No axioms	98.90	46.48	76.65
Naïve	92.99	59.70	80.98
SPSA,WN,VO	97.04	63.64	**83.13**

Evaluation

Results:

System	Prec.	Rec.	Acc.
Baseline (majority)	–	–	56.69
MLN	–	–	73.40
Nutcracker	–	–	74.30
Nutcracker-WN	–	–	77.50
Nutcracker-WN-PPDB	–	–	78.60
MLN-WN-PPDB	–	–	80.40
LangPro Hybrid-800	97.95	58.11	81.35
The Meaning Factory	93.63	60.64	81.60
No axioms	98.90	46.48	76.65
Naïve	92.99	59.70	80.98
SPSA,WN,VO	97.04	63.64	**83.13**
SemantiKLUE	85.40	69.63	82.32
UNAL-NLP	81.99	76.80	83.05
ECNU	84.37	74.37	83.64
Illinois-LH	81.56	81.87	84.57
MLN-eclassif (CL2016)	–	–	85.10
Yin-Schutze (EACL2017)	–	–	**87.10**

Error analysis

(more complex examples in back-up slide)

Prob. ID	T-H pairs	Gold	System	Axioms needed
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathsf{saw}(v) \rightarrow \mathsf{cut}(v)$ $\forall x.\mathsf{log}(x) \rightarrow \mathsf{wood}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathsf{slice}(v) \rightarrow \mathsf{cut}(v)$
2895	T: *The man isn't lifting weights .* H: *The man is lifting barbells .*	No	No	$\forall x.\mathsf{weight}(x) \rightarrow \mathsf{barbell}(x)$

Error analysis

(more complex examples in back-up slide)

Prob. ID	T-H pairs	Gold	System	Axioms needed
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathsf{saw}(v) \rightarrow \mathsf{cut}(v)$ $\forall x.\mathsf{log}(x) \rightarrow \mathsf{wood}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathsf{slice}(v) \rightarrow \mathsf{cut}(v)$
2895	T: *The man isn't lifting weights .* H: *The man is lifting barbells .*	No	No	$\forall x.\mathsf{weight}(x) \rightarrow \mathsf{barbell}(x)$
530	T: *A biker is wearing gear which is black .* H: *A biker wearing black is breaking the gears .*	Unk	Yes	

Error analysis

(more complex examples in back-up slide)

Prob. ID	T-H pairs	Gold	System	Axioms needed
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\text{saw}(v) \rightarrow \text{cut}(v)$ $\forall x.\text{log}(x) \rightarrow \text{wood}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\text{slice}(v) \rightarrow \text{cut}(v)$
2895	T: *The man isn't lifting weights .* H: *The man is lifting barbells .*	No	No	$\forall x.\text{weight}(x) \rightarrow \text{barbell}(x)$
530	T: *A biker is wearing gear which is black .* H: *A biker wearing black is breaking the gears .*	Unk	Yes	
1495	T: *A man is playing a guitar .* H: *A man is strumming a guitar .*	Yes	Unk	$\forall v.\text{play}(v) \rightarrow \text{strum}(v)$
1266	T: *A band is playing on a stage .* H: *A band is playing onstage .*	Yes	Unk	"on a stage" $\rightarrow$ "onstage"
2166	T: *A woman is sewing with a machine .* H: *A woman is using a machine made for sewing .*	Yes	Unk	"sewing with a machine" $\rightarrow$ "using a machine made for sewing"
384	T: *A white and tan dog is running through the tall and green grass .* H: *A white and tan dog is running through a field .*	Yes	Unk	"tall and green grass" $\rightarrow$ "field"

Phrasal Entailments with Visual Denotations

Phrasal Entailments with Visual Denotations

Recognizing phrase entailments is also necessary!

T: *men walk in the tall and green grass.*

$$\exists x.(\mathrm{man}(x) \wedge \exists y.(\mathrm{tall}(y) \wedge \mathrm{green}(y) \wedge \mathrm{grass}(y) \wedge \mathrm{walk}(x, y))$$

H: *men walk in the field.*

$$\exists x.(\mathrm{man}(x) \wedge \exists y.(\mathrm{field}(y) \wedge \mathrm{walk}(x, y))$$

Problem:

- Such knowledge can not be found in databases (e.g. WordNet, PPDB).
- Semantic relatedness $\neq$ semantic entailment.
- Distributional approaches (e.g. word2vec) are not effective:
  - piano $\not\Rightarrow$ guitar, cat $\not\Rightarrow$ dog

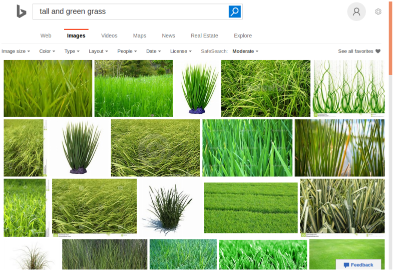
Phrasal Entailments with Visual Denotations

Get visual denotations of phrases and compare images.

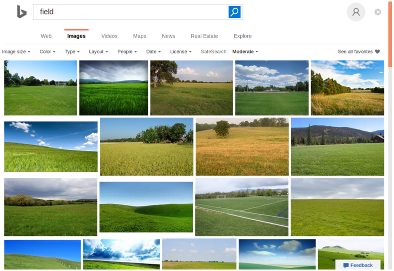
T: *men walk in the tall and green grass.*

H: *men walk in the field.*

tall and green grass



field



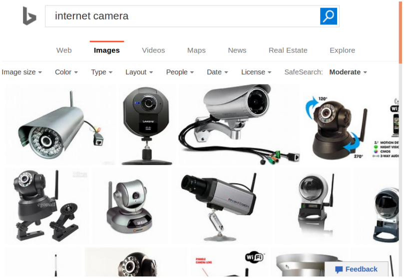
Phrasal Entailments with Visual Denotations

Get visual denotations of phrases and compare images.

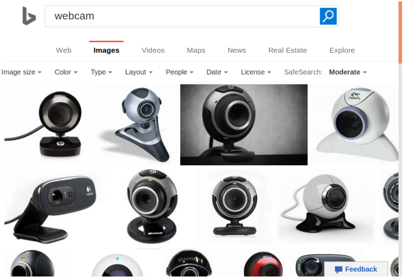
T: *He chats with his wife via internet camera.*

H: *He chats with his wife via webcam.*

internet camera

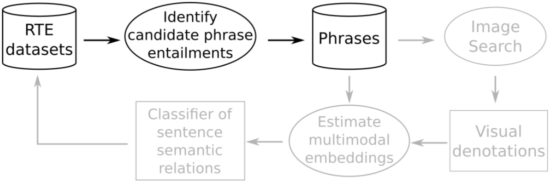


webcam

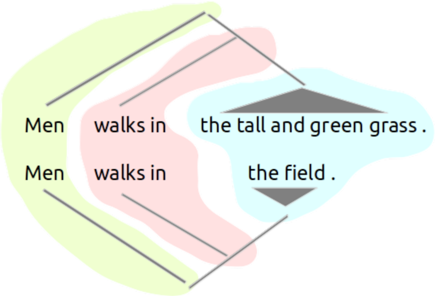


Phrasal Entailments with Visual Denotations

Step 1: phrase pair identification

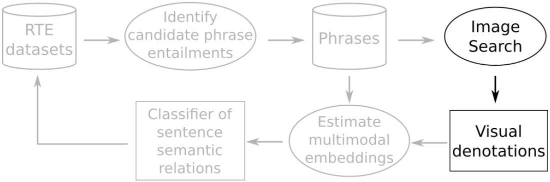


- Identify examples of phrase equivalences.

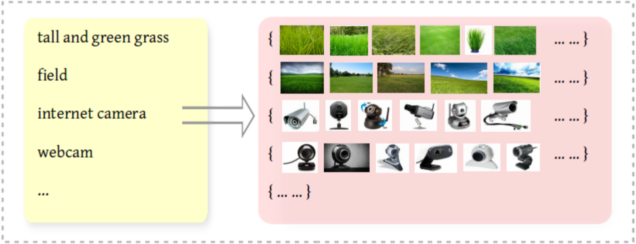


Phrasal Entailments with Visual Denotations

Step 2: obtain visual denotations

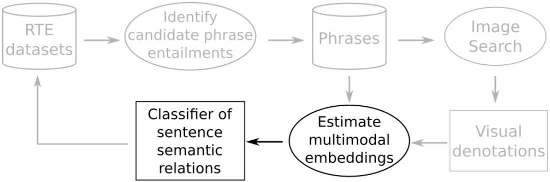


- Query images using phrases.

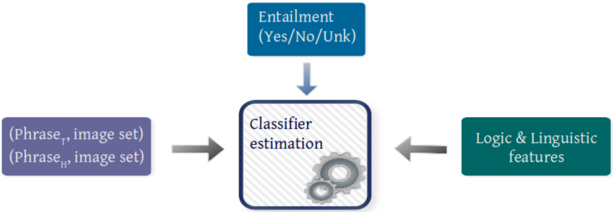


Phrasal Entailments with Visual Denotations

Step 3: Learn RTE Classifier

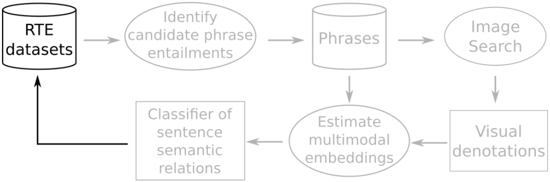


- Learn parameters of RTE classifier.

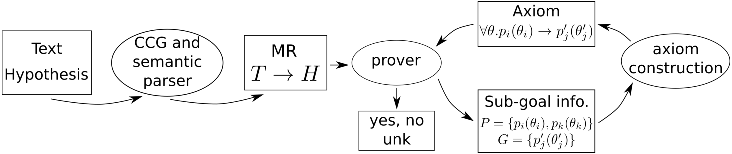


Phrasal Entailments with Visual Denotations

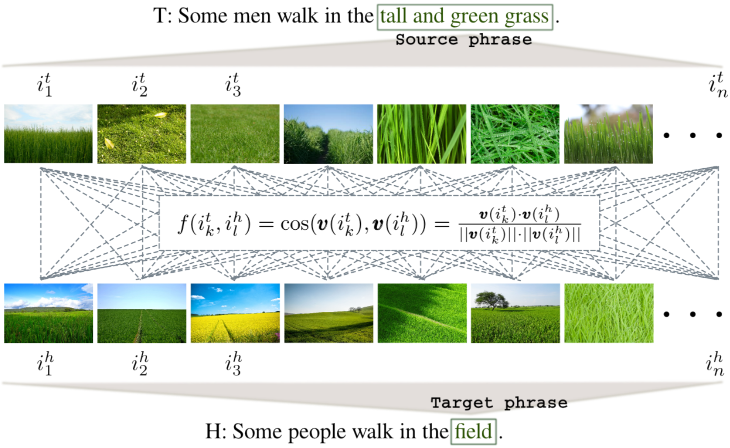
Step 4: Integrate into RTE pipeline



- Integrate on RTE pipeline and evaluate.



Phrasal Entailments with Visual Denotations



- Select best and worst phrase pair according to:

$$\mathrm{score}(t, h) = \frac{1}{|I_h|} \sum_{i^h_l \in I_h} \max_{i^t_k \in I_t} f(i^t_k, i^h_l)$$

Phrasal Entailments with Visual Denotations

Results when using visual denotations

System	Prec.	Rec.	Acc.
`ccg2lambda` + images	90.24	71.08	$\mathbf{84.29}$
`ccg2lambda`, only text	96.95	62.65	83.13
L&H, text + images	–	–	82.70
L&H, only text	–	–	81.50
Baseline (majority)	–	–	56.69

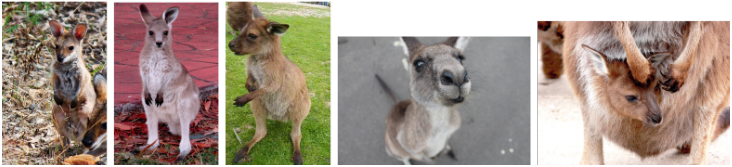
Phrasal Entailments with Visual Denotations

Examples

True positive:

T: The woman is picking up a kangaroo that is little.

H: The woman is picking up a baby kangaroo.



Phrasal Entailments with Visual Denotations

Examples

False positive:

T: A monkey is wading through a marsh.

H: A monkey is wading through a river.

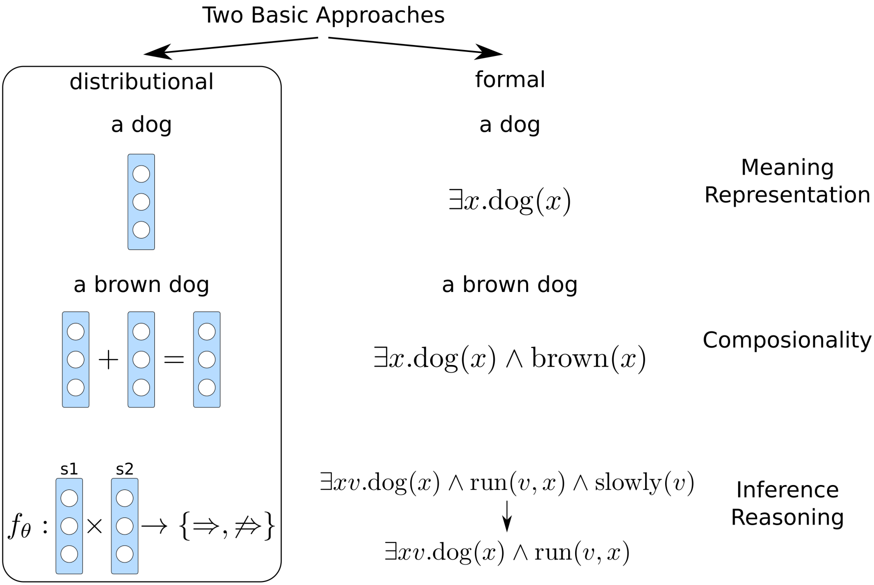
Phrasal Entailments with Visual Denotations

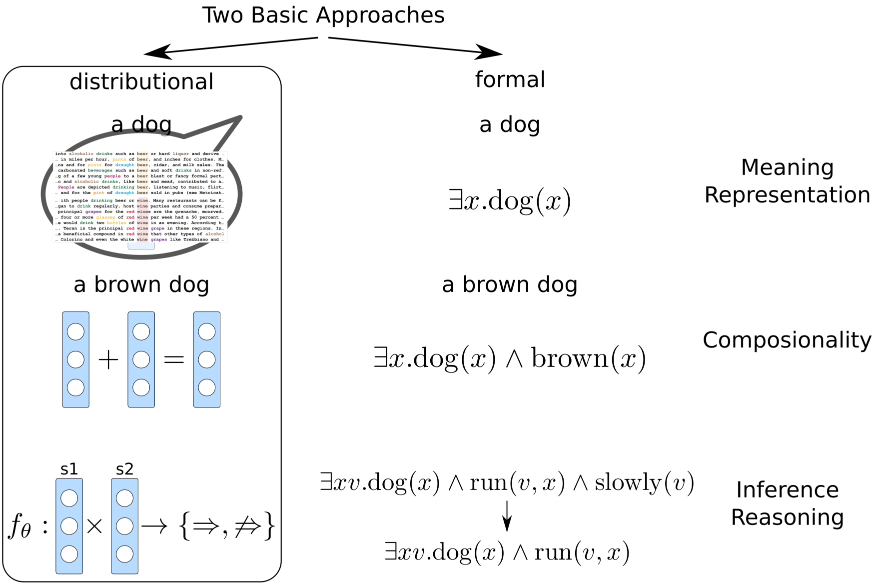
Examples

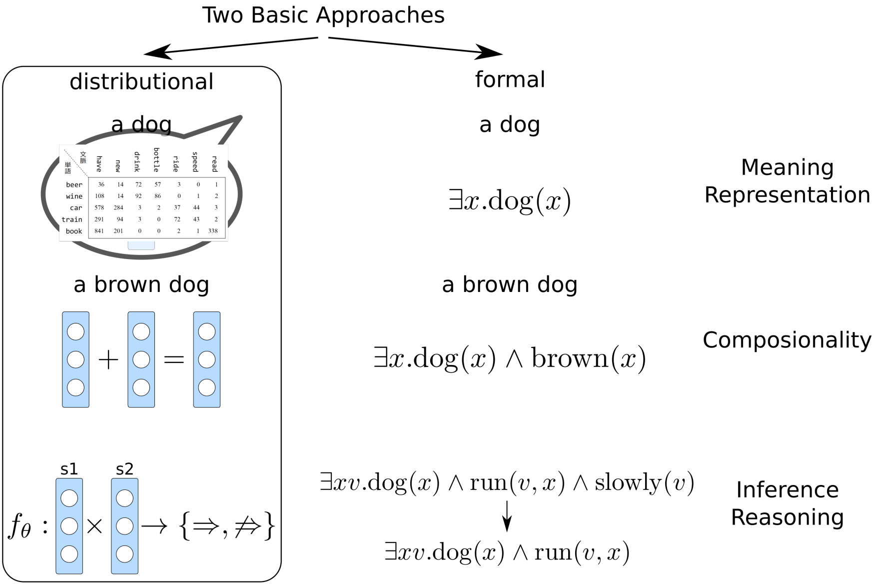
False negative:

T: A boy is spanking a man with a $\boxed{\text{plastic sword}}$.

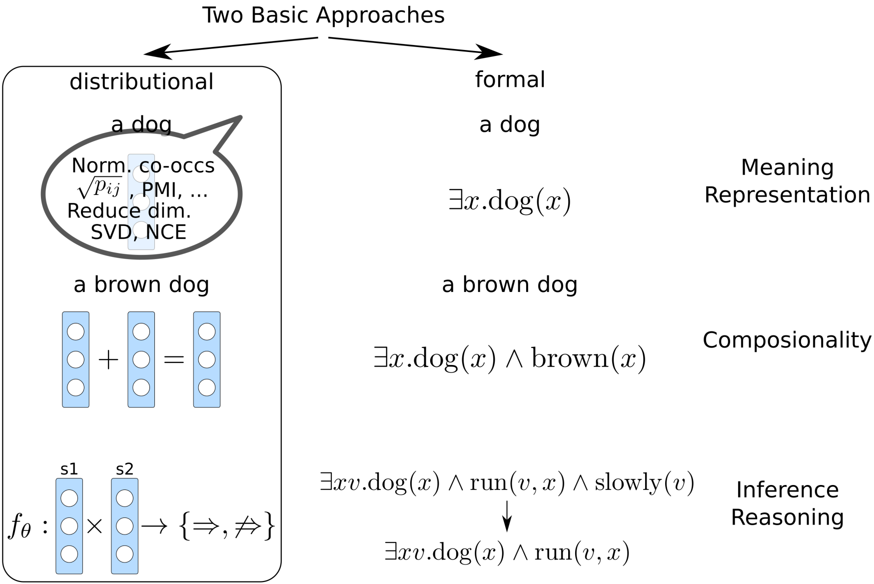
H: A boy is spanking a man with a $\boxed{\text{toy weapon}}$.


Two Basic Approaches
distributional
formal
a dog
a dog
$\exists x.\mathrm{dog}(x)$
Meaning Representation
a brown dog
a brown dog
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
Composionality
s1
s2
$f_\theta :$ $\times$ $\rightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Inference Reasoning

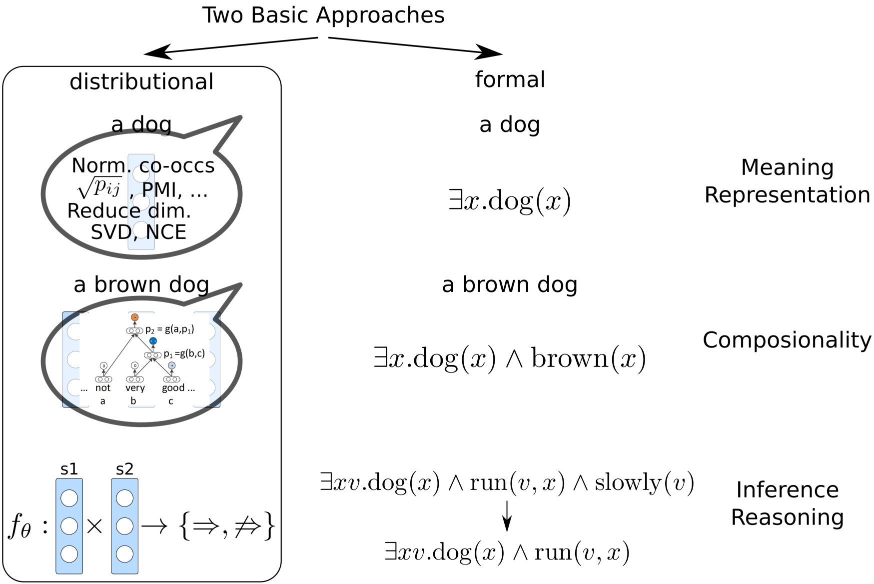

Two Basic Approaches
distributional
formal
a dog
a dog
Meaning Representation
$\exists x.\mathrm{dog}(x)$
a brown dog
a brown dog
Composionality
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
s1
s2
$f_\theta : \times \rightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Inference Reasoning



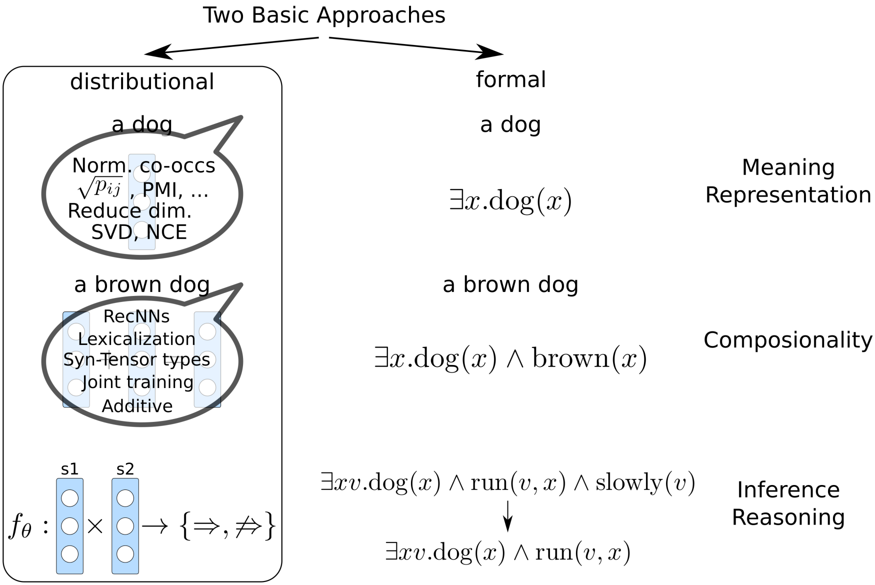
Two Basic Approaches
distributional
formal
a dog
文脈
単語
have new drink bottle ride speed read
beer 36 14 72 57 3 0 1
wine 108 14 92 86 0 1 2
car 578 284 3 2 37 44 3
train 291 94 3 0 72 43 2
book 841 201 0 0 2 1 338
a dog
$\exists x.\mathrm{dog}(x)$
Meaning Representation
a brown dog
a brown dog
$+$ $=$
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
Composionality
s1
s2
$f_\theta :$ $\times$ $\rightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v, x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v, x)$
Inference Reasoning



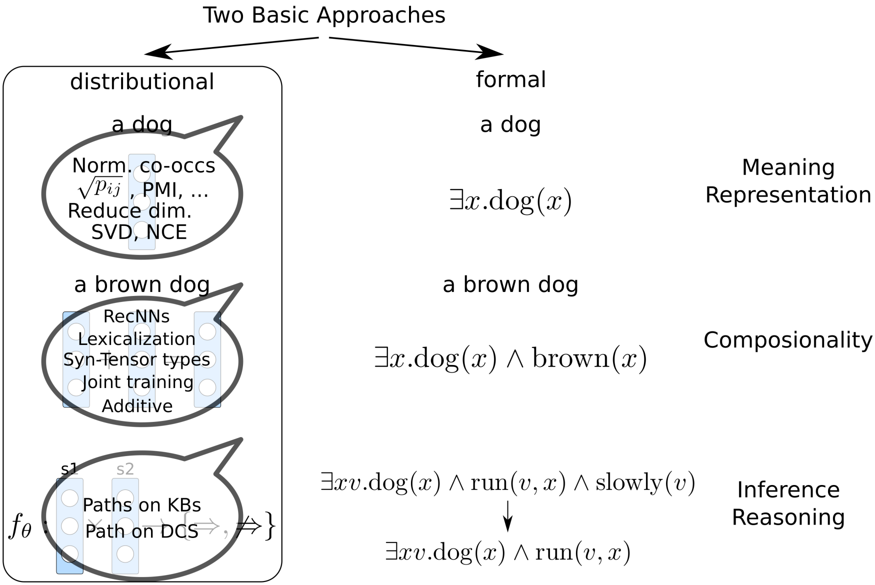
Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a dog
$\exists x.\mathrm{dog}(x)$
Meaning Representation
a brown dog
a brown dog
$+$ $=$
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
Composionality
s1
s2
$f_\theta :$ $\times$ $\rightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Inference Reasoning



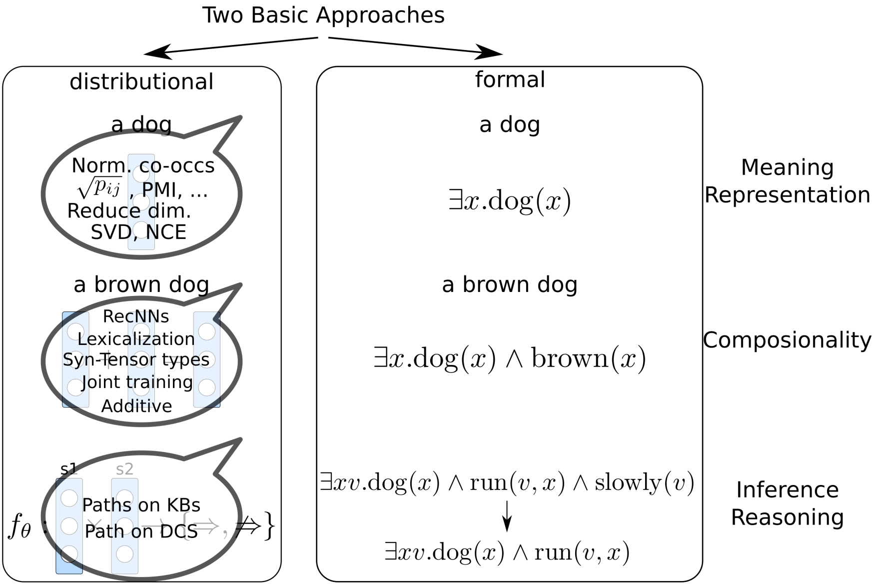
Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a dog
$\exists x.\mathrm{dog}(x)$
Meaning Representation
a brown dog
$p_2 = g(a,p_1)$
$p_1 = g(b,c)$
... not very good ...
a b c
a brown dog
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
Composionality
s1
s2
$f_\theta : \times \rightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Inference Reasoning

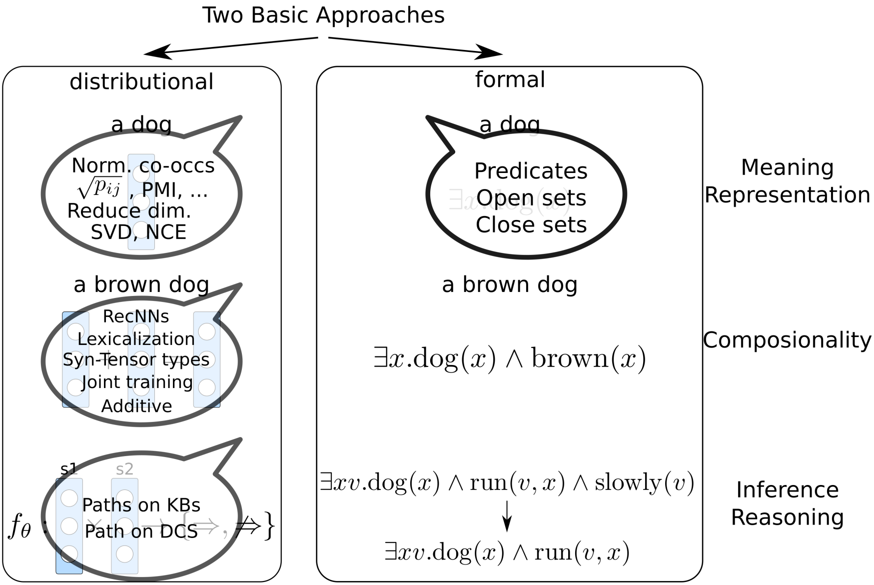


Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a dog
$\exists x.\mathrm{dog}(x)$
Meaning Representation
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
a brown dog
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
Composionality
s1
s2
$f_\theta : \times \rightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Inference Reasoning

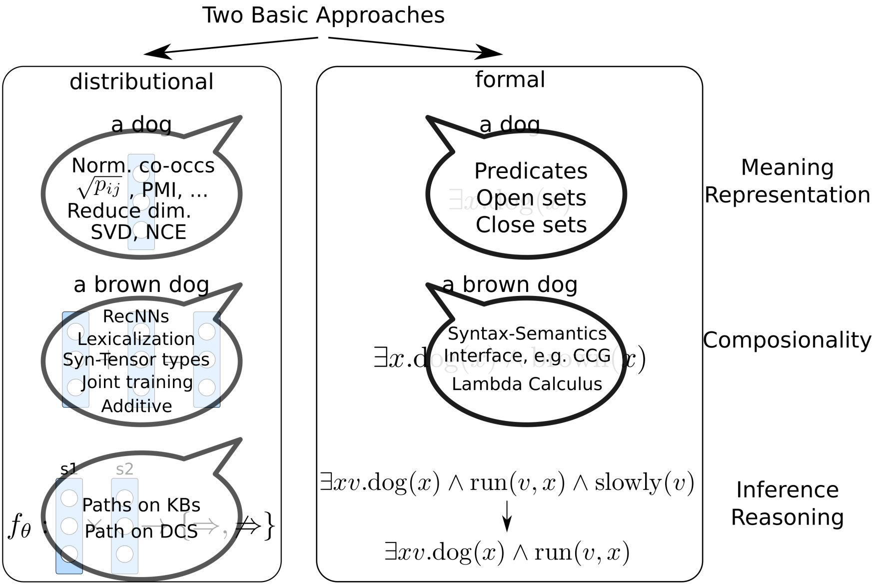


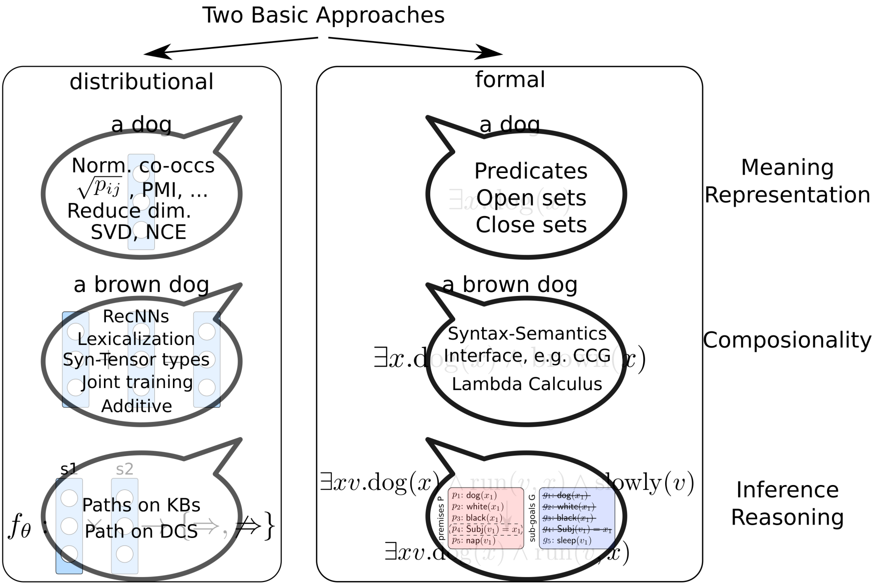
Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a dog
$\exists x.\mathrm{dog}(x)$
Meaning Representation
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
a brown dog
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
Composionality
s1
s2
Paths on KBs
Path on DCS
$f_\theta : \times \longrightarrow \{\Rightarrow, \not\Rightarrow\}$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v, x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v, x)$
Inference Reasoning


Two Basic Approaches
distributional
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
s1
s2
Paths on KBs
Path on DCS
$f_\theta : \times \to \{\Rightarrow, \not\Rightarrow\}$
formal
a dog
$\exists x.\mathrm{dog}(x)$
a brown dog
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Meaning Representation
Composionality
Inference Reasoning

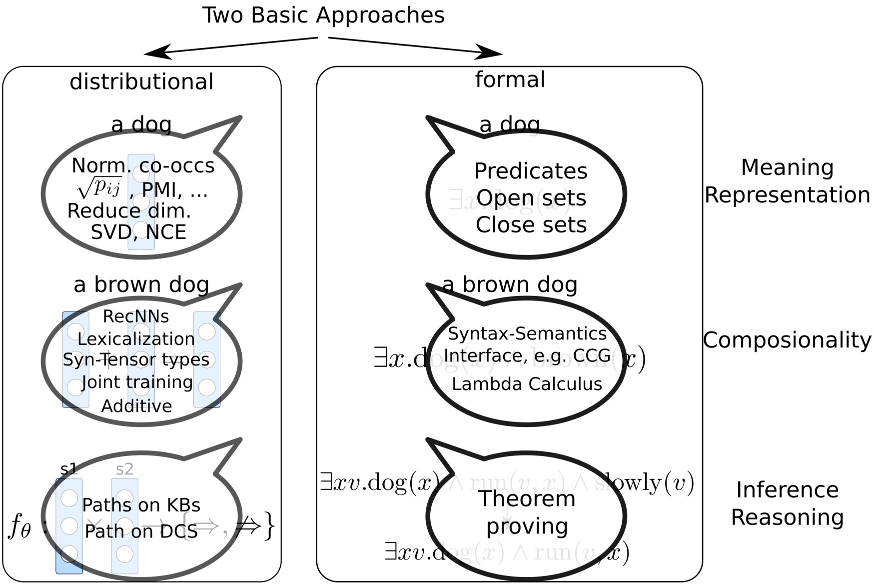


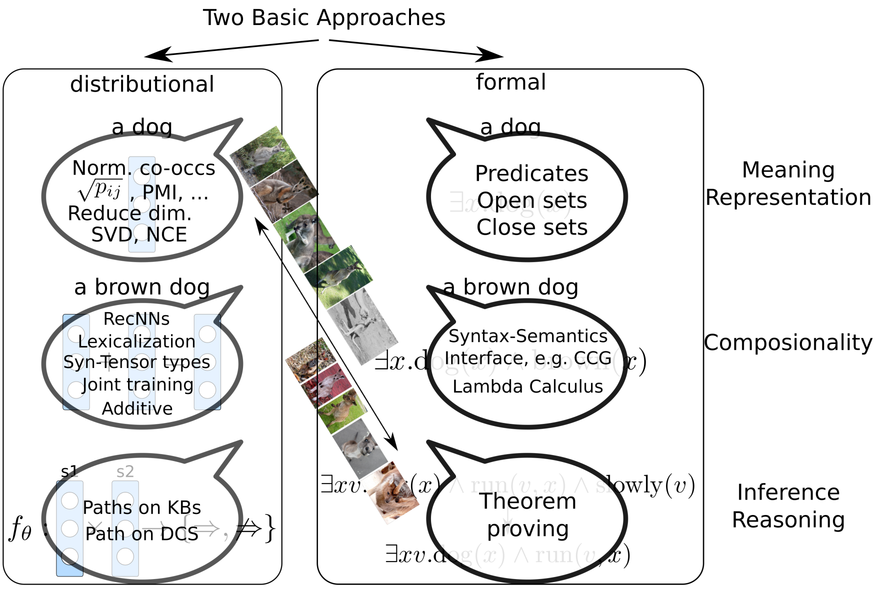
Two Basic Approaches
distributional
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
s1
s2
Paths on KBs
Path on DCS
$f_\theta :$
formal
a dog
Predicates
Open sets
Close sets
a brown dog
$\exists x.\mathrm{dog}(x) \wedge \mathrm{brown}(x)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Meaning Representation
Composionality
Inference Reasoning

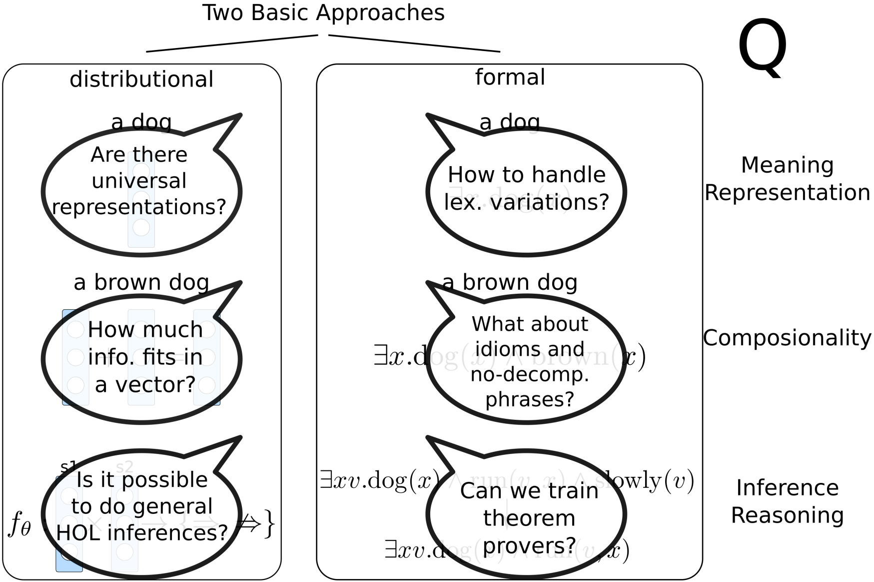

Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
s1
s2
Paths on KBs
Path on DCS
a dog
Predicates
Open sets
Close sets
a brown dog
Syntax-Semantics
Interface, e.g. CCG
Lambda Calculus
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x) \wedge \mathrm{slowly}(v)$
$\exists xv.\mathrm{dog}(x) \wedge \mathrm{run}(v,x)$
Meaning
Representation
Composionality
Inference
Reasoning



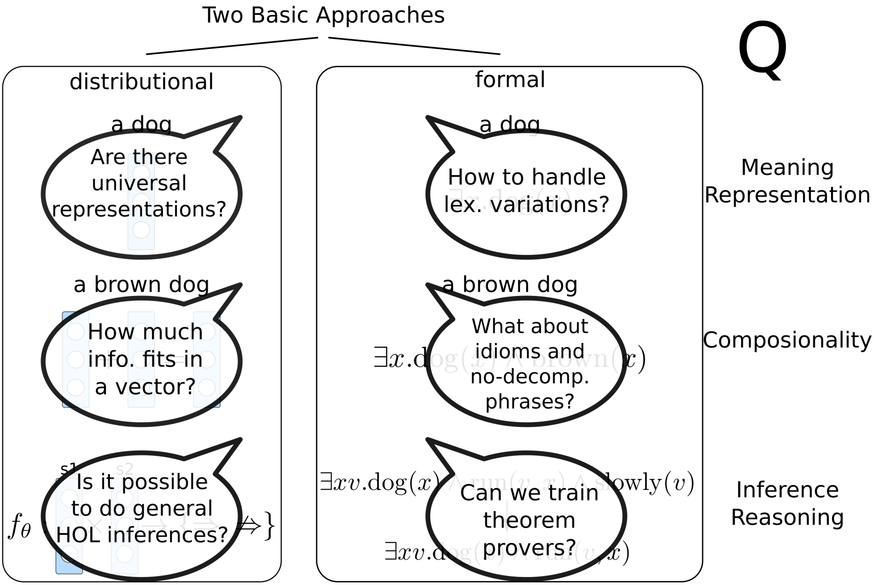
Two Basic Approaches
distributional
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
s1
s2
Paths on KBs
Path on DCS
formal
a dog
Predicates
Open sets
Close sets
a brown dog
Syntax-Semantics
Interface, e.g. CCG
Lambda Calculus
premises P
p_1: dog(x_1)
p_2: white(x_1)
p_3: black(x_1)
p_4: Subj(v_1) = x_1
p_5: nap(v_1)
sub-goals G
g_5: sleep(v_1)
Meaning Representation
Composionality
Inference Reasoning


Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a dog
Predicates
Open sets
Close sets
Meaning
Representation
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
a brown dog
Syntax-Semantics
Interface, e.g. CCG
Lambda Calculus
Composionality
s1
s2
Paths on KBs
Path on DCS
Theorem
proving
Inference
Reasoning


Two Basic Approaches
distributional
formal
a dog
Norm. co-occs
$\sqrt{p_{ij}}$, PMI, ...
Reduce dim.
SVD, NCE
a dog
Predicates
Open sets
Close sets
Meaning
Representation
a brown dog
RecNNs
Lexicalization
Syn-Tensor types
Joint training
Additive
a brown dog
Syntax-Semantics
Interface, e.g. CCG
Lambda Calculus
Composionality
s1
s2
Paths on KBs
Path on DCS
Theorem
proving
Inference
Reasoning



Two Basic Approaches
Q
distributional
formal
a dog
Are there universal representations?
a dog
How to handle lex. variations?
Meaning Representation
a brown dog
How much info. fits in a vector?
a brown dog
What about idioms and no-decomp. phrases?
Composionality
Is it possible to do general HOL inferences?
Can we train theorem provers?
Inference Reasoning



Thank you! Ran Tian, Koji Mineshima, Pascual Martínez-Gómez.

Reference I

Ajdukiewicz, K. (1935) “Die syntaktische Konnexit/”at”, *Studia Philosophica* **1**, pp.1–27.

Bar-Hillel, Y. (1953) “A quasi-arithmetical notation for syntactic description”, *Language* **29**(1), pp.47–58.

Barwise, J. and R. Cooper. (1981) “Generalized quantifiers and natural language”, *Linguistics and Philosophy* **4**(2), pp.159–219.

Bowman, S. R., G. Angeli, C. Potts, and C. D. Manning. (2015) “A large annotated corpus for learning natural language inference”, In the Proceedings of *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Cooper, R., R. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspers, H. Kamp, M. Pinkal, M. Poesio, S. Pulman, et al. (1994) “FraCaS – A Framework for Computational Semantics”, *Deliverable* **D16**.

Hobbs, J. R. (1985) “Ontological promiscuity”, In the Proceedings of *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*. pp.60–69.

Reference II

Kamp, H. and U. Reyle. (1993) *From discourse to logic: An introduction to modeltheoretic semantics of natural language, formal logic and DRT*. Kluwer, Dordrecht.

Kawazoe, A., R. Tanaka, K. Mineshima, and D. Bekki. (2015) "An Inference Problem Set for Evaluating Semantic Theories and Semantic Processing Systems for Japanese", In the Proceedings of *Proceedings of LENLS12*. pp.67–73.

Marelli, M., S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. (2014) "A SICK cure for the evaluation of compositional distributional semantic models.", In the Proceedings of *Proceedings of LREC2014*. pp.216–223.

Mineshima, K., P. Martínez-Gómez, Y. Miyao, and D. Bekki. (2015) "Higher-order logical inference with compositional semantics", In the Proceedings of *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp.2055–2061.

Moss, L. (2014) "Natural Logic". NASSLLI 2014, Lecture slides.

Parsons, T. (1990) *Events in the Semantics of English*. MIT Press.

Reference III

Steedman, M. (2000) *The Syntactic Process*. MIT Press.

Steedman, M. (2012) *Taking scope: The natural semantics of quantifiers*. Mit Press.