Natural Language Processing (CSEP 517): Predicate-Argument and Compositional Semantics

Noah Smith

University of Washington
nasmith@cs.washington.edu

May 8, 2017

To-Do List

- Online quiz: due Sunday
- Jurafsky and Martin (2016); (Jurafsky and Martin, 2008, ch. 18), Steedman (1996)
- A4 due May 14 (Sunday)

Semantics vs. Syntax

Syntactic theories and representations focus on the question of which strings in $\mathcal{V}^\dagger$ are in the language.

Semantics is about understanding what a string in $\mathcal{V}^\dagger$ *means*.

Sidestepping a lengthy and philosophical discussion of what "meaning" is, we'll consider two meaning representations:

- Predicate-argument structures, also known as event frames
- Truth conditions represented in first-order logic

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

In this buying/purchasing event/situation, Warren played the role of the buyer, and there was some stock that played the role of the thing purchased.

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

In this buying/purchasing event/situation, Warren played the role of the buyer, and there was some stock that played the role of the thing purchased.

Also, there was presumably a seller, only mentioned in one example.

Motivating Example: Who did What to Who(m)?

- Warren bought the stock.
- They sold the stock to Warren.
- The stock was bought by Warren.
- The purchase of the stock by Warren surprised no one.
- Warren's stock purchase surprised no one.

In this buying/purchasing event/situation, Warren played the role of the buyer, and there was some stock that played the role of the thing purchased.

Also, there was presumably a seller, only mentioned in one example.

In some examples, a separate "event" involving surprise did not occur.

Semantic Roles: Breaking

- Jesse broke the window.
- The window broke.
- Jesse is always breaking things.
- The broken window testified to Jesse's malfeasance.

Semantic Roles: Breaking

- Jesse broke the window.
- The window broke. ?
- Jesse is always breaking things.
- The broken window testified to Jesse's malfeasance.

A breaking event has a BREAKER and a BREAKEE.

Semantic Roles: Eating

- Eat!
- We ate dinner.
- We already ate.
- The pies were eaten up quickly.
- Our gluttony was complete.

Semantic Roles: Eating

- Eat! (you, listener) ?
- We ate dinner.
- We already ate. ?
- The pies were eaten up quickly. ?
- Our gluttony was complete. ?

An eating event has an EATER and FOOD, neither of which needs to be mentioned explicitly.

Abstraction?

$$\textsc{Breaker} \stackrel{?}{=} \textsc{Eater}$$

Abstraction?

$$\textsc{Breaker} \stackrel{?}{=} \textsc{Eater}$$

Both are actors that have some causal responsibility for changes in the world around them.

Abstraction?

$$\textsc{Breaker} \stackrel{?}{=} \textsc{Eater}$$

Both are actors that have some causal responsibility for changes in the world around them.

$$\textsc{Breakee} \stackrel{?}{=} \textsc{Food}$$

Abstraction?

$$\textsc{Breaker} \stackrel{?}{=} \textsc{Eater}$$

Both are actors that have some causal responsibility for changes in the world around them.

$$\textsc{Breakee} \stackrel{?}{=} \textsc{Food}$$

Both are greatly affected by the event, which "happened to" them.

Thematic Roles

(Jurafsky and Martin, 2016, with modifications)

AGENT	[The waiter] spilled the soup.
EXPERIENCER	[John] has a headache.
FORCE	[The wind] blows debris from the mall into our yards.
THEME	Jesse broke [the window]
RESULT	The city built [a regulation-size baseball diamond].
CONTENT	Mona asked, "[You met Mary Ann at a supermarket?]"
INSTRUMENT	He poached catfish, stunning them with [a shocking device].
BENEFICIARY	Ann Callahan makes hotel reservations for [her boss].
SOURCE	I flew in from [Boston].
GOAL	I drove to [Portland].

Verb Alternation Examples: Breaking and Giving

Breaking:

- AGENT/subject; THEME/object; INSTRUMENT/PP_{with}
- INSTRUMENT/subject; THEME/object
- THEME/subject

Giving:

- AGENT/subject; GOAL/object; THEME/second-object
- AGENT/subject; THEME/object; GOAL/PP_{to}

Levin (1993) codified English verbs into classes that share patterns (e.g., verbs of throwing: throw/kick/pass).

Remarks

- Fillmore (1968), among others, argued for semantic roles in linguistics.

Remarks

- Fillmore (1968), among others, argued for semantic roles in linguistics.
- By now, it should be clear that the expressiveness of NL (at least English) makes semantic analysis rather distinct from syntax.

Remarks

- Fillmore (1968), among others, argued for semantic roles in linguistics.
- By now, it should be clear that the expressiveness of NL (at least English) makes semantic analysis rather distinct from syntax.
- General challenges to analyzing semantic roles:
  - What are the predicates/events/frames/situations?
  - What are the roles/participants for each one?
  - What algorithms can accurately identify and label all of them?

Semantic Role Labeling

Input: a sentence x

Output:

- A collection of **predicates**, each consisting of:
  - a label, sometimes called the **frame**
  - a span
  - a set of **arguments**, each consisting of:
    - a label, usually called the **role**
    - a span

In principle, spans might have gaps, though in most conventions they usually do not.

The Importance of Lexicons

Like syntax, any annotated dataset is the product of extensive development of conventions.

Many conventions are specific to particular words, and this information is codified in structured objects called **lexicons**.

You should think of every semantically annotated dataset as both the data and the lexicon.

We consider two examples.

PropBank

(Palmer et al., 2005)

- Frames are verb senses (later extended, though)
- Lexicon maps verb-sense-specific roles onto a small set of abstract roles (e.g., ARG0, ARG1, etc.)
- Annotated on top of the Penn Treebank, so that arguments are always constituents.

fall.01 (move downward)

- ARG1: logical subject, patient, thing falling
- ARG2: extent, amount fallen
- ARG3: starting point
- ARG4: ending point
- ARGM-LOC: medium

- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Palo Alto.

fall.01 (move downward)

- ARG1: logical subject, patient, thing falling
- ARG2: extent, amount fallen
- ARG3: starting point
- ARG4: ending point
- ARGM-LOC: medium

- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Palo Alto.

fall.01 (move downward)

- ARG1: logical subject, patient, thing falling
- ARG2: extent, amount fallen
- ARG3: starting point
- ARG4: ending point
- ARGM-LOC: medium

- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Palo Alto.

fall.01 (move downward)

- ARG1: logical subject, patient, thing falling
- ARG2: extent, amount fallen
- ARG3: starting point
- ARG4: ending point
- ARGM-LOC: medium

- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Palo Alto.

fall.01 (move downward)

- ARG1: logical subject, patient, thing falling
- ARG2: extent, amount fallen
- ARG3: starting point
- ARG4: ending point
- ARGM-LOC: medium

- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Palo Alto.

fall.01 (move downward)

- ARG1: logical subject, patient, thing falling
- ARG2: extent, amount fallen
- ARG3: starting point
- ARG4: ending point
- ARGM-LOC: medium

- Sales fell to $251.2 million from $278.8 million.
- The average junk bond fell by 4.2%.
- The meteor fell through the atmosphere, crashing into Palo Alto.

fall.08 (fall back, rely on in emergency)

- ARG0: thing falling back
- ARG1: thing fallen back on

- World Bank president Paul Wolfowitz has fallen back on his last resort.

fall.08 (fall back, rely on in emergency)

- ARG0: thing falling back
- ARG1: thing fallen back on

- World Bank president Paul Wolfowitz has fallen back on his last resort.

fall.08 (fall back, rely on in emergency)

- ARG0: thing falling back
- ARG1: thing fallen back on

- World Bank president Paul Wolfowitz has fallen back on his last resort.

fall.10 (fall for a trick; be fooled by)

- ARG1: the fool
- ARG2: the trick

- Many people keep falling for the idea that lowering taxes on the rich benefits everyone.

fall.10 (fall for a trick; be fooled by)

- ARG1: the fool
- ARG2: the trick

- Many people keep falling for the idea that lowering taxes on the rich benefits everyone.

fall.10 (fall for a trick; be fooled by)

- ARG1: the fool
- ARG2: the trick

- Many people keep falling for the idea that lowering taxes on the rich benefits everyone.

FrameNet

(Baker et al., 1998)

- Frames can be any content word (verb, noun, adjective, adverb)
- About 1,000 frames, each with its own roles
- Both frames and roles are hierarchically organized
- Annotated without syntax, so that arguments can be anything

`https://framenet.icsi.berkeley.edu`

change_position_on_a_scale

- ITEM: entity that has a position on the scale
- ATTRIBUTE: scalar property that the ITEM possesses
- DIFFERENCE: distance by which an ITEM changes its position
- FINAL_STATE: ITEM's state after the change
- FINAL_VALUE: position on the scale where ITEM ends up
- INITIAL_STATE: ITEM's state before the change
- INITIAL_VALUE: position on the scale from which the ITEM moves
- VALUE_RANGE: portion of the scale along which values of ATTRIBUTE fluctuate
- DURATION: length of time over which the change occurs
- SPEED: rate of change of the value
- GROUP: the group in which an ITEM changes the value of an ATTRIBUTE

FrameNet Example

Attacks on civilians decreased over the last four months

decreased: change_position_on_a_scale

Attacks on civilians: ITEM

over the last four months: DURATION

The ATTRIBUTE is left unfilled but is understood from context (i.e., "frequency").

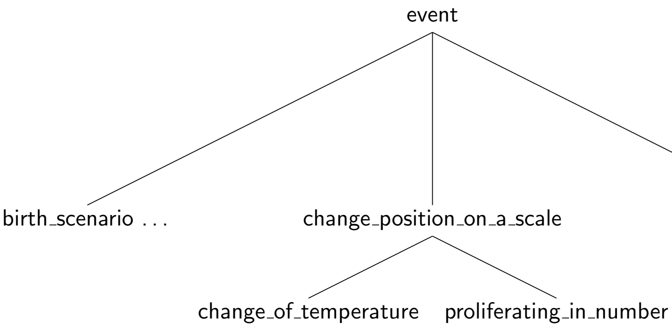
change_position_on_a_scale

Verbs: advance, climb, decline, decrease, diminish, dip, double, drop, dwindle, edge, explode, fall, fluctuate, gain, grow, increase, jump, move, mushroom, plummet, reach, rise, rocket, shift, skyrocket, slide, soar, swell, swing, triple, tumble

Nouns: decline, decrease, escalation, explosion, fall, fluctuation, gain, growth, hike, increase, rise, shift, tumble

Adverb: increasingly

change_position_on_a_scale



(birth_scenario also inherits from sexual_reproduction_scenario.)

Semantic Role Labeling Tasks

The paper that started it all: Gildea and Jurafsky (2002) used FrameNet lexicon (which includes prototypes, not really a corpus).

- When FrameNet started releasing corpora, the task was reformulated. Example open-source system: SEMAFOR (Das et al., 2014).

Semantic Role Labeling Tasks

The paper that started it all: Gildea and Jurafsky (2002) used FrameNet lexicon (which includes prototypes, not really a corpus).

- When FrameNet started releasing corpora, the task was reformulated. Example open-source system: SEMAFOR (Das et al., 2014).

The PropBank corpus is used directly for training/testing.

Semantic Role Labeling Tasks

The paper that started it all: Gildea and Jurafsky (2002) used FrameNet lexicon (which includes prototypes, not really a corpus).

- When FrameNet started releasing corpora, the task was reformulated. Example open-source system: SEMAFOR (Das et al., 2014).

The PropBank corpus is used directly for training/testing.

Conference on Computational Natural Language Learning (CoNLL) shared task in 2004, 2005, 2008, 2009, all PropBank-based.

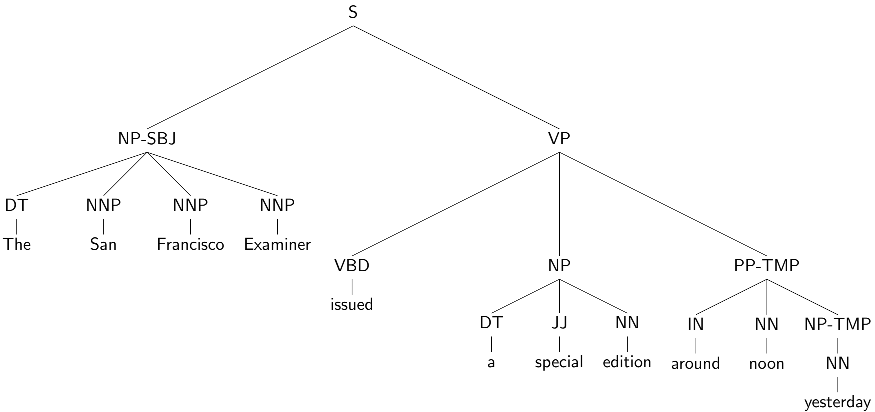
- In 2008 and 2009, the task was cast as a kind of dependency parsing.
- In 2009, seven languages were included in the task.

Methods

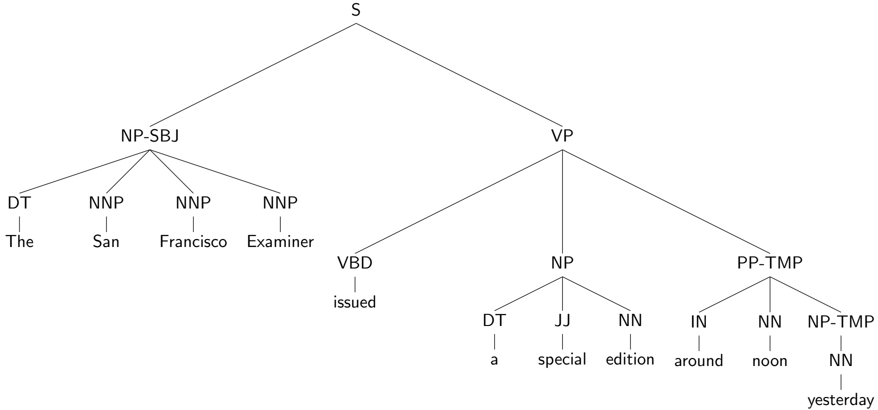
Boils down to labeling spans (with frames and roles).

It's mostly about features.

Example: Path Features

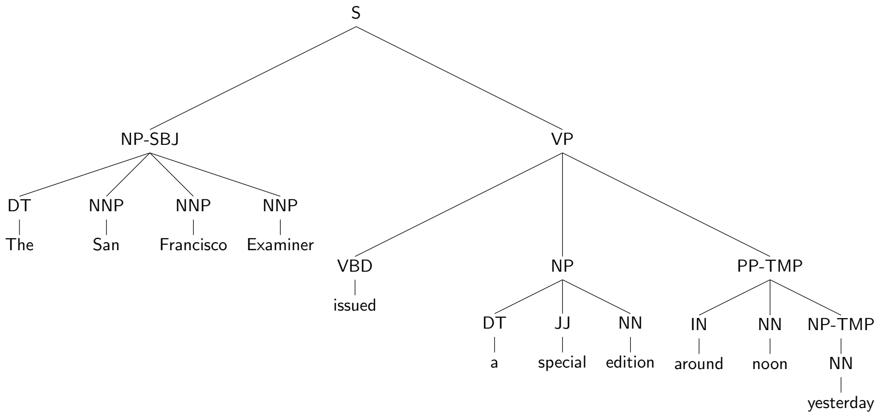


Example: Path Features



Path from NP-SBJ (The San Francisco Examiner) to issued: NP$\uparrow$S$\downarrow$VP$\downarrow$VBD

Example: Path Features



Path from NP (a special edition) to issued: NP↑VP↓VBD

Methods: Beyond Features

The span-labeling decisions interact a lot!

- Presence of a frame increases the expectation of certain roles
- Roles for the same predicate shouldn't overlap
- Some roles are mutually exclusive or require each other (e.g., "resemble")

Methods: Beyond Features

The span-labeling decisions interact a lot!

- Presence of a frame increases the expectation of certain roles
- Roles for the same predicate shouldn't overlap
- Some roles are mutually exclusive or require each other (e.g., "resemble")

Ensuring well-formed outputs:

- Using syntax as a scaffold allows efficient prediction; you're essentially labeling the parse tree (Toutanova et al., 2008).
- Others have formulated the problem as constrained, discrete optimization (Punyakanok et al., 2008).
- Also greedy methods (Björkelund et al., 2010) and joint methods for syntactic and semantic dependencies (Henderson et al., 2013).

Methods: Beyond Features

The span-labeling decisions interact a lot!

- Presence of a frame increases the expectation of certain roles
- Roles for the same predicate shouldn't overlap
- Some roles are mutually exclusive or require each other (e.g., "resemble")

Ensuring well-formed outputs:

- Using syntax as a scaffold allows efficient prediction; you're essentially labeling the parse tree (Toutanova et al., 2008).
- Others have formulated the problem as constrained, discrete optimization (Punyakanok et al., 2008).
- Also greedy methods (Björkelund et al., 2010) and joint methods for syntactic and semantic dependencies (Henderson et al., 2013).

Current work:

- Some recent attempts to merge FrameNet and PropBank have shown promise (FitzGerald et al., 2015; Kshirsagar et al., 2015)

Related Problems in "Relational" Semantics

- **Coreference resolution**: which mentions (within or across texts) refer to the same entity or event?
- **Entity linking**: ground such mentions in a structured knowledge base (e.g., Wikipedia)
- **Relation extraction**: characterize the relation among specific mentions

Information extraction: transform text into a structured knowledge representation

- Classical IE starts with a predefined schema
- "Open" IE includes the automatic construction of the schema; see `http://ai.cs.washington.edu/projects/open-information-extraction`

General Remarks

Criticisms of semantic role labeling:

- Semantic roles are just "syntax++" since they don't allow much in the way of reasoning (e.g., question answering).

General Remarks

Criticisms of semantic role labeling:

- Semantic roles are just “syntax++” since they don’t allow much in the way of reasoning (e.g., question answering).
- Lexicon building is slow and requires expensive expertise. Can we do this for every (sub)language?

General Remarks

Criticisms of semantic role labeling:

- Semantic roles are just "syntax++" since they don't allow much in the way of reasoning (e.g., question answering).
- Lexicon building is slow and requires expensive expertise. Can we do this for every (sub)language?

We've now had a taste of two branches of semantics:

- Lexical semantics (e.g., supersense tagging)
- Relational semantics (e.g., semantic role labeling)

General Remarks

Criticisms of semantic role labeling:

- Semantic roles are just "syntax++" since they don't allow much in the way of reasoning (e.g., question answering).
- Lexicon building is slow and requires expensive expertise. Can we do this for every (sub)language?

We've now had a taste of two branches of semantics:

- Lexical semantics (e.g., supersense tagging)
- Relational semantics (e.g., semantic role labeling)

Next up, a third:

- Compositional semantics

Bridging the Gap between Language and the World

In order to link NL to a knowledge base, we might want to design a formal way to represent meaning.

Desiderata for a **meaning representation language**:

Bridging the Gap between Language and the World

In order to link NL to a knowledge base, we might want to design a formal way to represent meaning.

Desiderata for a **meaning representation language**:

- represent the state of the world, i.e., a knowledge base

Bridging the Gap between Language and the World

In order to link NL to a knowledge base, we might want to design a formal way to represent meaning.
Desiderata for a **meaning representation language**:

- represent the state of the world, i.e., a knowledge base
- query the knowledge base (e.g., verify that a statement is true, or answer a question)

Bridging the Gap between Language and the World

In order to link NL to a knowledge base, we might want to design a formal way to represent meaning.
Desiderata for a **meaning representation language**:

- represent the state of the world, i.e., a knowledge base
- query the knowledge base (e.g., verify that a statement is true, or answer a question)
- handle ambiguity, vagueness, and non-canonical forms
  - "I wanna eat someplace that's close to UW"
  - "something not too spicy"

Bridging the Gap between Language and the World

In order to link NL to a knowledge base, we might want to design a formal way to represent meaning.

Desiderata for a **meaning representation language**:

- represent the state of the world, i.e., a knowledge base
- query the knowledge base (e.g., verify that a statement is true, or answer a question)
- handle ambiguity, vagueness, and non-canonical forms
  - "I wanna eat someplace that's close to UW"
  - "something not too spicy"
- support inference and reasoning
  - "can Karen eat at Schultzy's?"

Bridging the Gap between Language and the World

In order to link NL to a knowledge base, we might want to design a formal way to represent meaning.

Desiderata for a **meaning representation language**:

- represent the state of the world, i.e., a knowledge base
- query the knowledge base (e.g., verify that a statement is true, or answer a question)
- handle ambiguity, vagueness, and non-canonical forms
  - "I wanna eat someplace that's close to UW"
  - "something not too spicy"
- support inference and reasoning
  - "can Karen eat at Schultzy's?"

Eventually (but not today):

- deal with non-literal meanings
- expressiveness across a wide range of subject matter

A (Tiny) World Model

- **Domain:** Adrian, Brook, Chris, Donald, Schultzy's Sausage, Din Tai Fung, Banana Leaf, American, Chinese, Thai
- **Property:** Din Tai Fung has a long wait, Schultzy's is noisy; Alice, Bob, and Charles are human
- **Relations:** Schultzy's serves American, Din Tai Fung serves Chinese, and Banana Leaf serves Thai

Simple questions are easy:

- Is Schultzy's noisy?
- Does Din Tai Fung serve Thai?

A (Tiny) World Model

- **Domain:** Adrian, Brook, Chris, Donald, Schultzy's Sausage, Din Tai Fung, Banana Leaf, American, Chinese, Thai
  $a, b, c, d, ss, dtf, bl, am, ch, th$
- **Property:** Din Tai Fung has a long wait, Schultzy's is noisy; Alice, Bob, and Charles are human
  $Longwait = \{dtf\}, Noisy = \{ss\}, Human = \{a, b, c\}$
- **Relations:** Schultzy's serves American, Din Tai Fung serves Chinese, and Banana Leaf serves Thai
  $Serves = \{(ss, am), (dtf, ch), (bl, th)\}, Likes = \{(a, ss), (a, dtf), \ldots\}$

Simple questions are easy:

- Is Schultzy's noisy?
- Does Din Tai Fung serve Thai?

A Quick Tour of First-Order Logic

- **Term:** a constant (ss) or a variable
- **Formula:** defined inductively ...
  - If R is an n-ary relation and $t_1, \ldots, t_n$ are terms, then $R(t_1, \ldots, t_n)$ is a formula.
  - If ϕ is a formula, then its negation, $\neg\phi$, is a formula.
  - If ϕ and ψ are formulas, then binary logical connectives can be used to create formulas:
    - $\phi \wedge \psi$
    - $\phi \vee \psi$
    - $\phi \Rightarrow \psi$
    - $\phi \oplus \psi$
  - If ϕ is a formula and v is a variable, then quantifiers can be used to create formulas:
    - Universal quantifier: $\forall v, \phi$
    - Existential quantifier: $\exists v, \phi$

Note: Leaving out functions, because we don't need them in a single lecture on FOL for NL.

Translating Between FOL and NL

1. Schultzy's is not loud
2. Some human likes Chinese
3. If a person likes Thai, then they aren't friends with Donald
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \vee \neg Likes(a, x))$
5. $\forall x, \exists y, \neg Likes(x, y)$
6. $\exists y, \forall x, \neg Likes(x, y)$

Translating Between FOL and NL

1. Schultzy's is not loud $\neg Noisy(ss)$
2. Some human likes Chinese
3. If a person likes Thai, then they aren't friends with Donald
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \vee \neg Likes(a, x))$
5. $\forall x, \exists y, \neg Likes(x, y)$
6. $\exists y, \forall x, \neg Likes(x, y)$

Translating Between FOL and NL

1. Schultzy's is not loud $\neg Noisy(ss)$
2. Some human likes Chinese $\exists x, Human(x) \wedge Likes(x, ch)$
3. If a person likes Thai, then they aren't friends with Donald
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \vee \neg Likes(a, x))$
5. $\forall x, \exists y, \neg Likes(x, y)$
6. $\exists y, \forall x, \neg Likes(x, y)$

Translating Between FOL and NL

1. Schultzy's is not loud $\neg Noisy(ss)$
2. Some human likes Chinese $\exists x, Human(x) \land Likes(x, ch)$
3. If a person likes Thai, then they aren't friends with Donald
   $\forall x, Human(x) \land Likes(x, th) \Rightarrow \neg Friends(x, d)$
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \lor \neg Likes(a, x))$
5. $\forall x, \exists y, \neg Likes(x, y)$
6. $\exists y, \forall x, \neg Likes(x, y)$

Translating Between FOL and NL

1. Schultzy's is not loud $\neg Noisy(ss)$
2. Some human likes Chinese $\exists x, Human(x) \wedge Likes(x, ch)$
3. If a person likes Thai, then they aren't friends with Donald
   $\forall x, Human(x) \wedge Likes(x, th) \Rightarrow \neg Friends(x, d)$
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \vee \neg Likes(a, x))$
   Every restaurant has a long wait or is disliked by Adrian.
5. $\forall x, \exists y, \neg Likes(x, y)$
6. $\exists y, \forall x, \neg Likes(x, y)$

Translating Between FOL and NL

1. Schultzy's is not loud $\neg Noisy(ss)$
2. Some human likes Chinese $\exists x, Human(x) \land Likes(x, ch)$
3. If a person likes Thai, then they aren't friends with Donald
   $\forall x, Human(x) \land Likes(x, th) \Rightarrow \neg Friends(x, d)$
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \lor \neg Likes(a, x))$
   Every restaurant has a long wait or is disliked by Adrian.
5. $\forall x, \exists y, \neg Likes(x, y)$
   Everybody has something they don't like.
6. $\exists y, \forall x, \neg Likes(x, y)$

Translating Between FOL and NL

1. Schultzy's is not loud $\neg Noisy(ss)$
2. Some human likes Chinese $\exists x, Human(x) \wedge Likes(x, ch)$
3. If a person likes Thai, then they aren't friends with Donald
   $\forall x, Human(x) \wedge Likes(x, th) \Rightarrow \neg Friends(x, d)$
4. $\forall x, Restaurant(x) \Rightarrow (Longwait(x) \vee \neg Likes(a, x))$
   Every restaurant has a long wait or is disliked by Adrian.
5. $\forall x, \exists y, \neg Likes(x, y)$
   Everybody has something they don't like.
6. $\exists y, \forall x, \neg Likes(x, y)$
   There exists something that nobody likes.

Logical Semantics

(Montague, 1970)

The denotation of a NL sentence is the set of conditions that must hold in the (model) world for the sentence to be true.

Every restaurant has a long wait or Adrian doesn't like it.

is true if and only if

$$\forall x, Restaurant(x) \Rightarrow (Longwait(x) \vee \neg Likes(a, x))$$

is true.

This is sometimes called the **logical form** of the NL sentence.

The Principle of Compositionality

The meaning of a NL phrase is determined by the meanings of its sub-phrases.

The Principle of Compositionality

The meaning of a NL phrase is determined by the meanings of its sub-phrases.

I.e., semantics is derived from syntax.

The Principle of Compositionality

The meaning of a NL phrase is determined by the meanings of its sub-phrases.

I.e., semantics is derived from syntax.

We need a way to express semantics of phrases, and compose them together!

λ-Calculus

(Much more powerful than what we'll see today; ask your PL professor!)

Informally, two extensions:

- λ-**abstraction** is another way to "scope" variables.
  - If ϕ is a FOL formula and v is a variable, then $\lambda v.\phi$ is a λ-term, meaning: an unnamed function from values (of v) to formulas (usually involving v)
- **application** of such functions: if we have $\lambda v.\phi$ and ψ, then $[\lambda v.\phi](\psi)$ is a formula.
  - It can be **reduced** by substituting ψ in for every instance of v in ϕ.

λ-Calculus

(Much more powerful than what we'll see today; ask your PL professor!)

Informally, two extensions:

- λ-**abstraction** is another way to "scope" variables.
  - If ϕ is a FOL formula and v is a variable, then $\lambda v.\phi$ is a λ-term, meaning: an unnamed function from values (of v) to formulas (usually involving v)
- **application** of such functions: if we have $\lambda v.\phi$ and ψ, then $[\lambda v.\phi](\psi)$ is a formula.
  - It can be **reduced** by substituting ψ in for every instance of v in ϕ.

Example:
$\lambda x.Likes(x, dtf)$ maps things to statements that they like Din Tai Fung

λ-Calculus

(Much more powerful than what we'll see today; ask your PL professor!)

Informally, two extensions:

- λ-**abstraction** is another way to "scope" variables.
  - If ϕ is a FOL formula and v is a variable, then $\lambda v.\phi$ is a λ-term, meaning: an unnamed function from values (of v) to formulas (usually involving v)
- **application** of such functions: if we have $\lambda v.\phi$ and ψ, then $[\lambda v.\phi](\psi)$ is a formula.
  - It can be **reduced** by substituting ψ in for every instance of v in ϕ.

Example:
$[\lambda x.Likes(x, dtf)](c)$ reduces to $Likes(c, dtf)$

λ-Calculus

(Much more powerful than what we'll see today; ask your PL professor!)

Informally, two extensions:

- λ-**abstraction** is another way to "scope" variables.
  - If ϕ is a FOL formula and v is a variable, then $\lambda v.\phi$ is a λ-term, meaning: an unnamed function from values (of v) to formulas (usually involving v)
- **application** of such functions: if we have $\lambda v.\phi$ and ψ, then $[\lambda v.\phi](\psi)$ is a formula.
  - It can be **reduced** by substituting ψ in for every instance of v in ϕ.

Example:
$\lambda x.\lambda y.Friends(x, y)$ maps things x to maps of things y to statements that x and y are friends

λ-Calculus

(Much more powerful than what we'll see today; ask your PL professor!)

Informally, two extensions:

- λ-**abstraction** is another way to "scope" variables.
  - If ϕ is a FOL formula and v is a variable, then $\lambda v.\phi$ is a λ-term, meaning: an unnamed function from values (of v) to formulas (usually involving v)
- **application** of such functions: if we have $\lambda v.\phi$ and ψ, then $[\lambda v.\phi](\psi)$ is a formula.
  - It can be **reduced** by substituting ψ in for every instance of v in ϕ.

Example:
$[\lambda x.\lambda y.Friends(x, y)](b)$ reduces to $\lambda y.Friends(b, y)$

λ-Calculus

(Much more powerful than what we'll see today; ask your PL professor!)

Informally, two extensions:

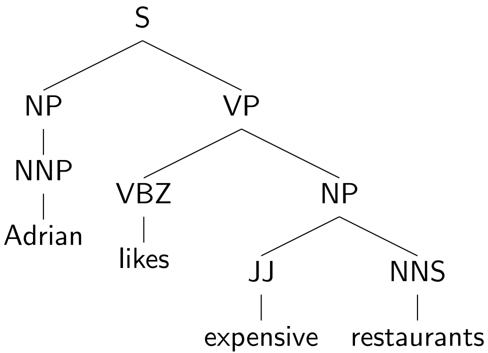
- λ-**abstraction** is another way to "scope" variables.
  - If ϕ is a FOL formula and v is a variable, then $\lambda v.\phi$ is a λ-term, meaning: an unnamed function from values (of v) to formulas (usually involving v)
- **application** of such functions: if we have $\lambda v.\phi$ and ψ, then $[\lambda v.\phi](\psi)$ is a formula.
  - It can be **reduced** by substituting ψ in for every instance of v in ϕ.

Example:
$[[\lambda x.\lambda y.Friends(x, y)](b)](a)$ reduces to $[\lambda y.Friends(b, y)](a)$, which reduces to $Friends(b, a)$

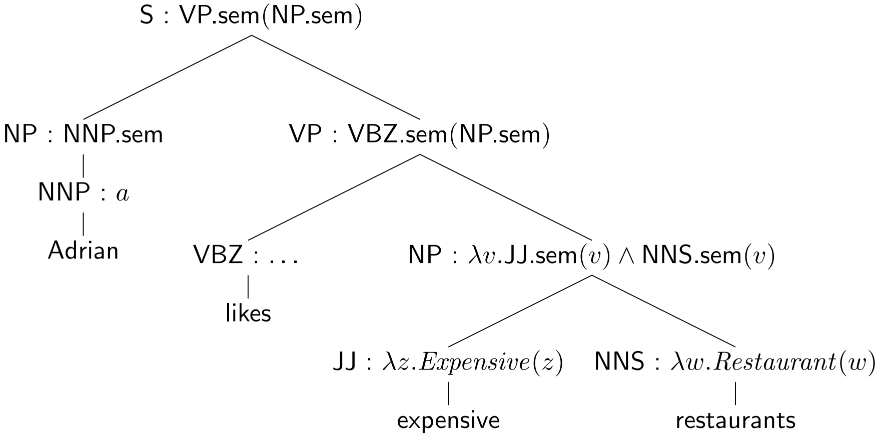
Semantic Attachments to CFG

- NNP $\rightarrow$ Adrian $\{a\}$
- VBZ $\rightarrow$ likes $\{\lambda f.\lambda y.\forall x f(x) \Rightarrow Likes(y,x)\}$
- JJ $\rightarrow$ expensive $\{\lambda x.Expensive(x)\}$
- NNS $\rightarrow$ restaurants $\{\lambda x.Restaurant(x)\}$
- NP $\rightarrow$ NNP {NNP.sem}
- NP $\rightarrow$ JJ NNS $\{\lambda x.\text{JJ.sem}(x) \wedge \text{NNS.sem}(x)\}$
- VP $\rightarrow$ VBZ NP {VBZ.sem(NP.sem)}
- S $\rightarrow$ NP VP {VP.sem(NP.sem)}

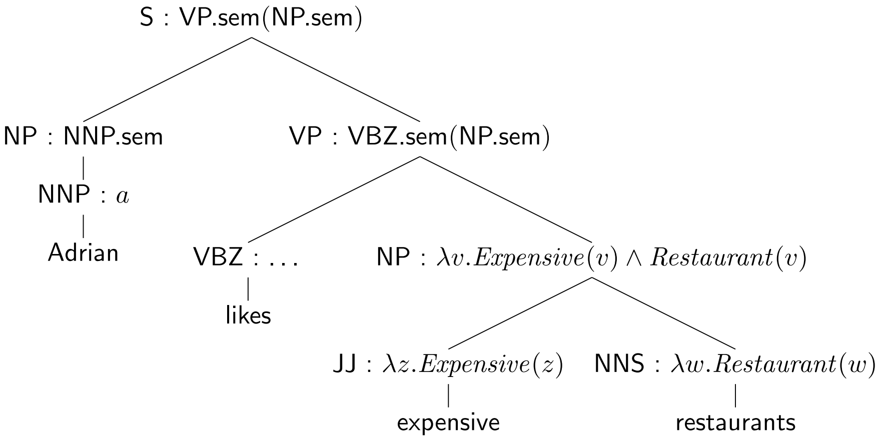
Example



Example

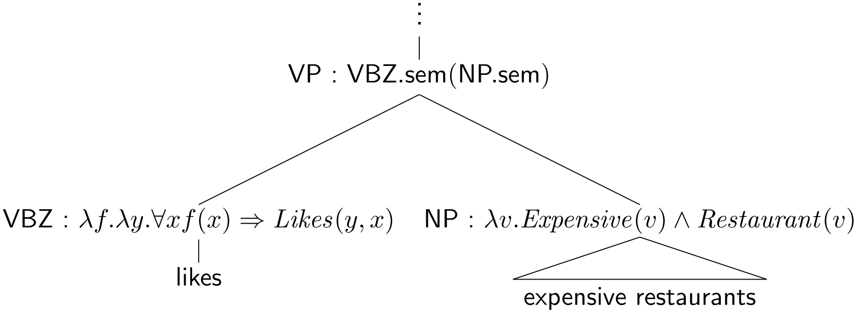

S : VP.sem(NP.sem)
NP : NNP.sem
VP : VBZ.sem(NP.sem)
NNP : a
Adrian
VBZ : . . .
likes
NP : $\lambda v.$JJ.sem$(v) \wedge$ NNS.sem(v)
JJ : $\lambda z.Expensive(z)$
expensive
NNS : $\lambda w.Restaurant(w)$
restaurants

Example

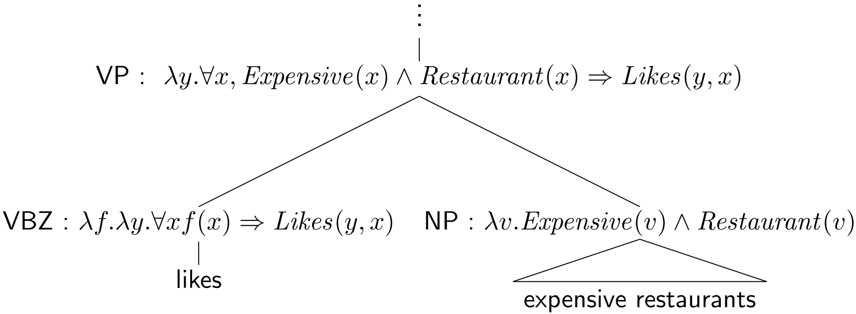


$$\lambda v.\left[\underbrace{\lambda z.Expensive(z)}_{\text{JJ.sem}}\right](v) \wedge \left[\underbrace{\lambda w.Restaurant(w)}_{\text{NNS.sem}}\right](v)$$

Example


⋮
VP : VBZ.sem(NP.sem)
VBZ : $\lambda f.\lambda y.\forall x\, f(x) \Rightarrow Likes(y, x)$
likes
NP : $\lambda v.Expensive(v) \wedge Restaurant(v)$
expensive restaurants

Example

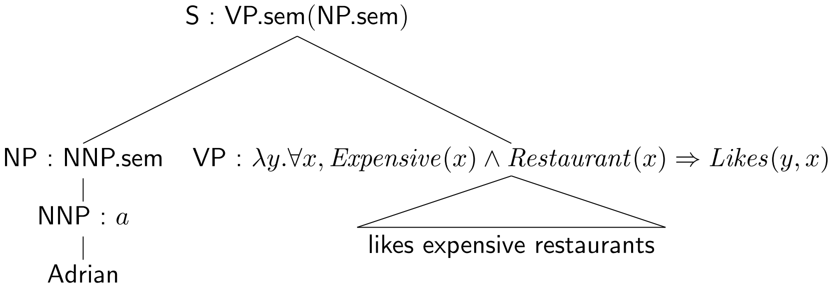


$$\Bigg[\underbrace{\lambda f.\lambda y.\forall x f(x) \Rightarrow Likes(y, x)}_{\text{VBZ.sem}}\Bigg]\Bigg(\underbrace{\lambda v.Expensive(v) \wedge Restaurant(v)}_{\text{NP.sem}}\Bigg)$$

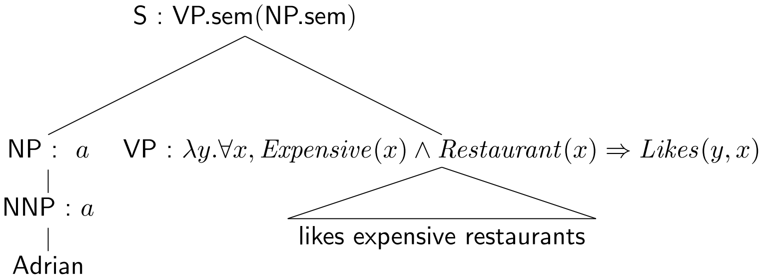
$$\lambda y.\forall x\, [\lambda v.Expensive(v) \wedge Restaurant(v)]\,(x) \Rightarrow Likes(y, x)$$

$$\lambda y.\forall x, Expensive(x) \wedge Restaurant(x) \Rightarrow Likes(y, x)$$

Example

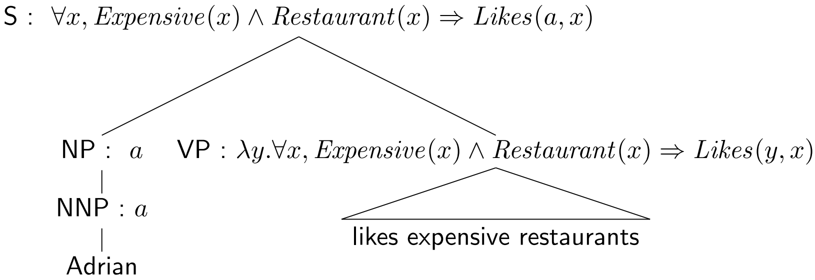

S : VP.sem(NP.sem)
NP : NNP.sem
VP : $\lambda y.\forall x, Expensive(x) \wedge Restaurant(x) \Rightarrow Likes(y, x)$
NNP : a
Adrian
likes expensive restaurants

Example



S : VP.sem(NP.sem)
NP : a
VP : $\lambda y.\forall x, Expensive(x) \wedge Restaurant(x) \Rightarrow Likes(y, x)$
NNP : a
Adrian
likes expensive restaurants

Example

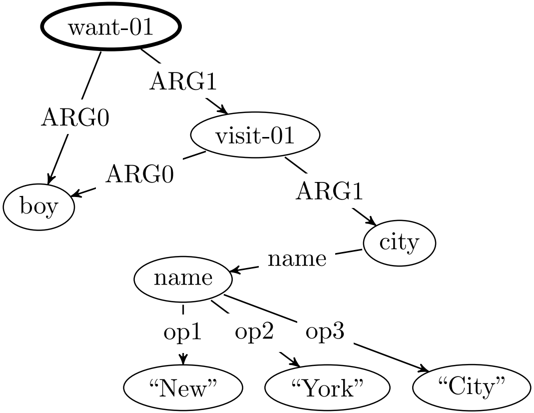


$$\Big[\underbrace{\lambda y.\forall x, Expensive(x) \wedge Restaurant(x) \Rightarrow Likes(y, x)}_{\text{VP.sem}}\Big]\Big(\underbrace{a}_{\text{NP.sem}}\Big)$$

$$\forall x, Expensive(x) \wedge Restaurant(x) \Rightarrow Likes(a, x)$$

Graph-Based Representations

Abstract Meaning Representation (Banarescu et al., 2013)



"The boy wants to visit New York City."

Designed for (1) annotation-ability and (2) eventual use in machine translation.

Combinatory Categorial Grammar

(Steedman, 2000)

CCG is a grammatical formalism that is well-suited for tying together syntax and semantics.

Formally, it is more powerful than CFG—it can represent some of the context-*sensitive* languages (which we do not have time to define formally).

CCG Types

Instead of the "$\mathcal{N}$" of CFGs, CCGs can have an infinitely large set of structured categories (called **types**).

- Primitive types: typically S, NP, N, and maybe more
- Complex types, built with "slashes," for example:
  - S/NP is "an S, except that it lacks an NP to the right"
  - S\NP is "an S, except that it lacks an NP to its left"
  - (S\NP)/NP is "an S, except that it lacks an NP to its right, and its left"

You can think of complex types as functions, e.g., S/NP maps NPs to Ss.

CCG Combinators

Instead of the production rules of CFGs, CCGs have a very small set of generic **combinators** that tell us how we can put types together.

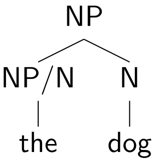
Convention writes the rule differently from CFG: $X \quad Y \Rightarrow Z$ means that X and Y combine to form a Z (the "parent" in the tree).

Application Combinator

Forward $(X/Y \quad Y \Rightarrow X)$ and backward $(Y \quad X\backslash Y \Rightarrow X)$

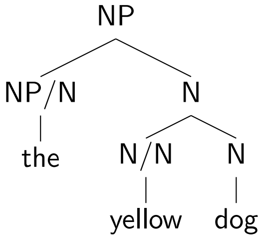
Application Combinator

Forward $(X/Y \quad Y \Rightarrow X)$ and backward $(Y \quad X\backslash Y \Rightarrow X)$



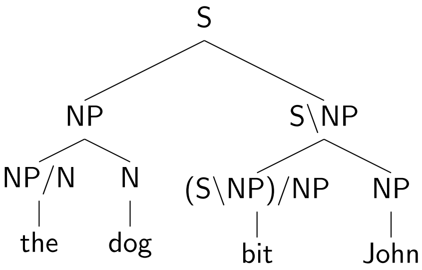
Application Combinator

Forward $(X/Y \quad Y \Rightarrow X)$ and backward $(Y \quad X\backslash Y \Rightarrow X)$



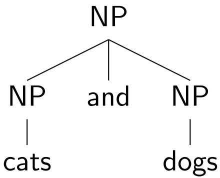
Application Combinator

Forward $(X/Y \quad Y \Rightarrow X)$ and backward $(Y \quad X\backslash Y \Rightarrow X)$



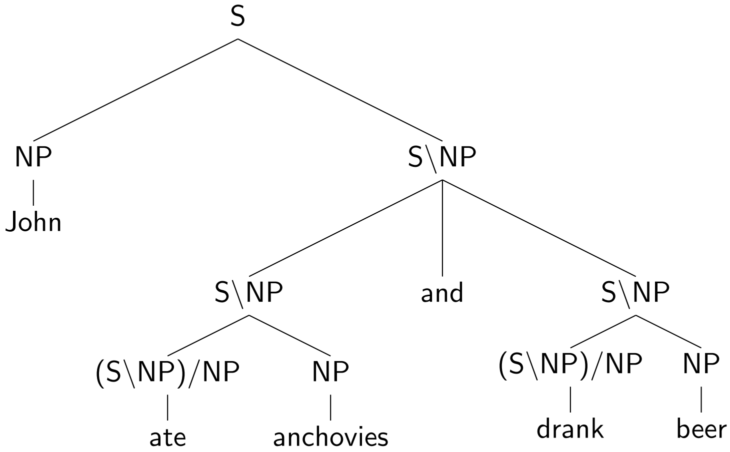
Conjunction Combinator

$$X \text{ and } X \Rightarrow X$$



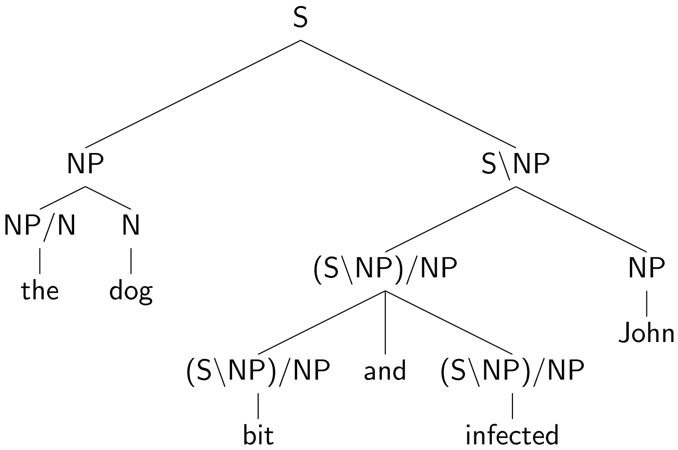
Conjunction Combinator

X and $X \Rightarrow X$



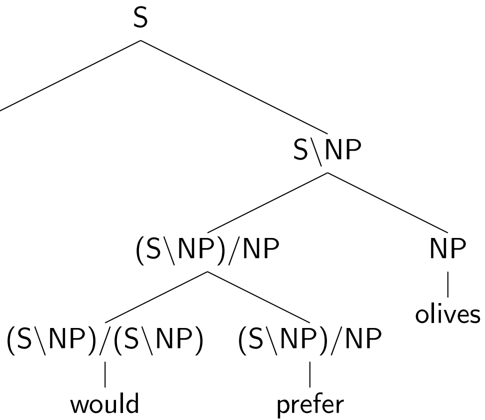
Conjunction Combinator

X and $X \Rightarrow X$



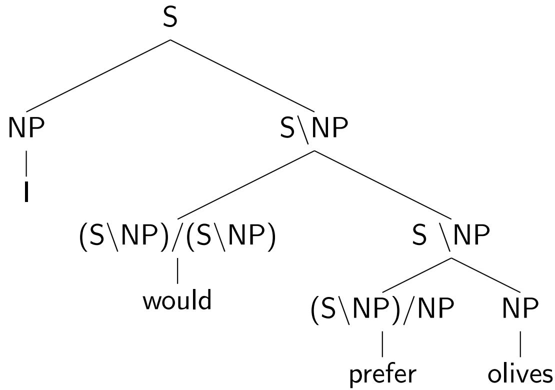
Composition Combinator

Forward $(X/Y \quad Y/Z \Rightarrow X/Z)$ and backward $(Y\backslash Z \quad X\backslash Y \Rightarrow X\backslash Z)$



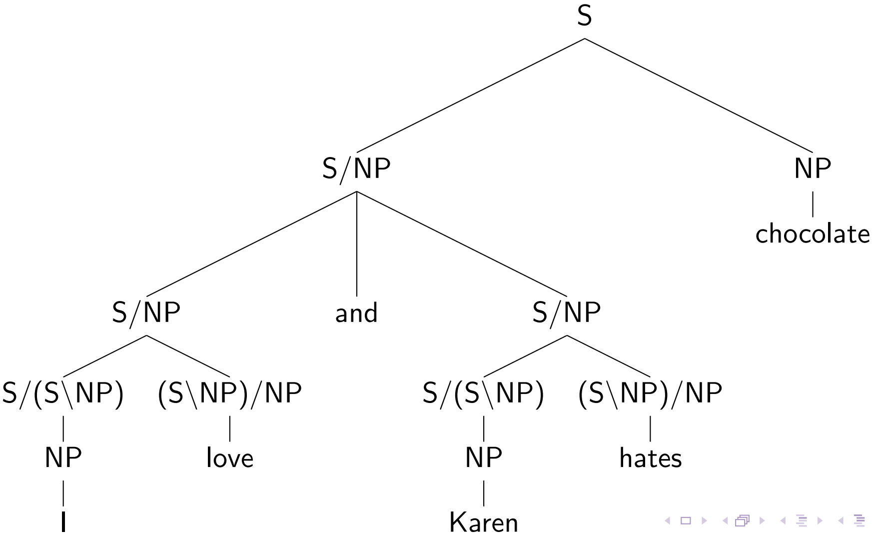
Composition Combinator

Forward $(X/Y \quad Y/Z \Rightarrow X/Z)$ and backward $(Y\backslash Z \quad X\backslash Y \Rightarrow X\backslash Z)$



Type-Raising Combinator

Forward ($X \Rightarrow Y/(Y\backslash X)$) and backward ($X \Rightarrow Y\backslash(Y/X)$)



Back to Semantics

Each combinator also tells us what to do with the semantic attachments.

- Forward application: $X/Y : f \quad Y : g \Rightarrow X : f(g)$
- Forward composition: $X/Y : f \quad Y/Z : g \Rightarrow X/Z : \lambda x.f(g(x))$
- Forward type-raising: $X : g \Rightarrow Y/(Y\backslash X) : \lambda f.f(g)$

CCG Lexicon

Most of the work is done in the lexicon!

Syntactic and semantic information is much more formal here.

- Slash categories define where all the syntactic arguments are expected to be
- λ-expressions define how the expected arguments get "used" to build up a FOL expression

Extensive discussion: Carpenter (1997)

Some Topics We Don't Have Time For

- Tasks, evaluations, annotated datasets (e.g., CCGbank, Hockenmaier and Steedman, 2007)
- Learning for semantic parsing (Zettlemoyer and Collins, 2005) and CCG parsing (Clark and Curran, 2004a)
- Using CCG to represent other kinds of semantics (e.g., predicate-argument structures; Lewis and Steedman, 2014)
- Integrating continuous representations in semantic parsing (Lewis and Steedman, 2013; Krishnamurthy and Mitchell, 2013)
- Supertagging (Clark and Curran, 2004b) and making semantic parsing efficient (Lewis and Steedman, 2014)
- *Grounding* meaning in visual (or other perceptual) experience

References I

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *Proc. of ACL-COLING*, 1998.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proc. of the Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. A high-performance syntactic and semantic dependency parser. In *Proc. of COLING*, 2010.

Bob Carpenter. *Type-logical semantics*. MIT Press, 1997.

Stephen Clark and James R. Curran. Parsing the WSJ using CCG and log-linear models. In *Proc. of ACL*, 2004a.

Stephen Clark and James R. Curran. The importance of supertagging for wide-coverage CCG parsing. In *Proc. of COLING*, 2004b.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56, 2014.

Charles J. Fillmore. The case for case. In Bach and Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart, and Winston, 1968.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. Semantic role labeling with neural network factors. In *Proc. of EMNLP*, 2015.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 24(3): 245–288, 2002.

References II

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4):949–998, 2013.

Julia Hockenmaier and Mark Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, 2007.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, second edition, 2008.

Daniel Jurafsky and James H. Martin. Semantic role labeling and argument structure (draft chapter), 2016. URL `https://web.stanford.edu/~jurafsky/slp3/22.pdf`.

Jayant Krishnamurthy and Tom Mitchell. Vector space semantic parsing: A framework for compositional vector space models. 2013.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime Carbonell, Noah A. Smith, and Chris Dyer. Frame-semantic role labeling with heterogeneous annotations. In *Proc. of ACL*, 2015.

Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago Press, 1993.

Mike Lewis and Mark Steedman. Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192, 2013.

Mike Lewis and Mark Steedman. A* CCG parsing with a supertag-factored model. In *Proc. of EMNLP*, 2014.

Richard Montague. Universal grammar. *Theoria*, 36:373–398, 1970.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.

References III

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287, 2008.

Mark Steedman. A very short introduction to CCG, 1996. URL http://www.inf.ed.ac.uk/teaching/courses/nlg/readings/ccgintro.pdf.

Mark Steedman. *The Syntactic Process*. MIT Press, 2000.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191, 2008.

Luke Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of UAI*, 2005.