

# Introduction to Natural Language Processing

Alexandre Rademaker

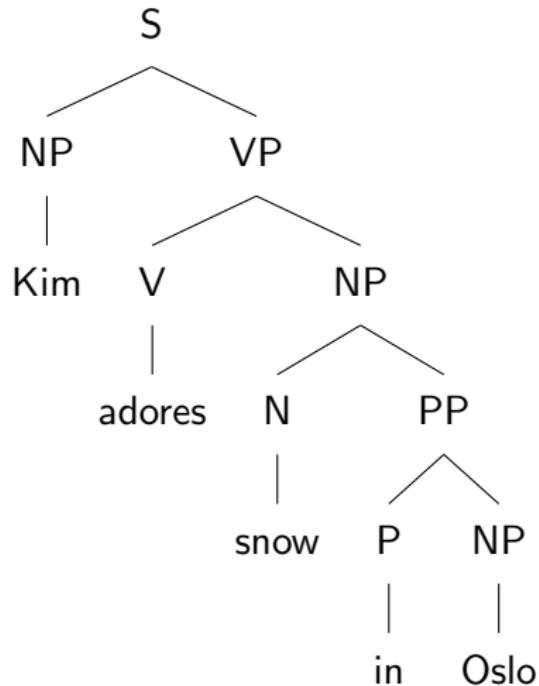
FGV EMAp

October 19, 2021

# Overview

- ▶ Formal semantics, FOL, lambda-calculus
- ▶ Compositional semantics
- ▶ Semantics in computational linguistics
- ▶ Semantics in NLP

Parsing makes explicit inherent structure. So, does this tree represent meaning?



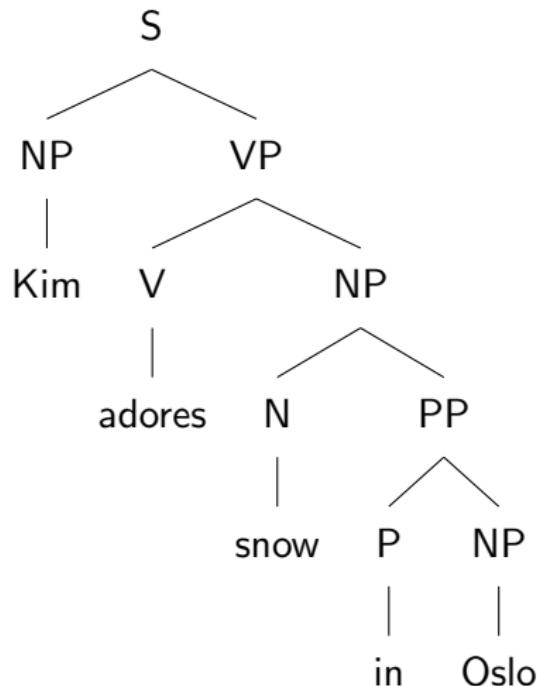
# Why represent meaning computationally?

*I hated this movie!*

- ▶ A Dialog system:
  - ▶ Parser:
    - ▶ Yes, it is grammatical!
    - ▶ Here's the structure!
  - ▶ System: Great, but what am I supposed to DO?

## Formal semantics question:

**How could we put this tree in correspondence to a model of the world?**



# Model theoretic semantics

- ▶ Create a model of the world consisting of elements, sets of elements, and relations
  - ▶ not so much a model of what things **mean** as of **how we reason** about them
- ▶ Create an interpretation function which maps linguistic elements (parts of the semantic structure) to parts of the model
- ▶ Simple propositions are interpreted by checking their truth in the model
- ▶ Define semantics for “logical vocabulary”: and, or, not, if, every, some...

# Model theoretic semantics (example)

- Entities: Joey:



Fluffy:



Tiger:



- Properties: calm: {



- }; angry: {



}

- Relations: knows: { <



, >, <



> }

# Model theoretic semantics (denotations)

- $[[\text{Fluffy}]] =$



- $[[\text{angry}]] = \{ x \mid x \text{ is angry} \} = \{$



}

- $[[\text{Fluffy is angry}]] = \text{True iff}$  the entity denoted by *Fluffy* is in the set denoted by *angry*
- Compositionality: The process of determining the truth conditions of *Fluffy is angry* based on the denotations of its parts and its syntactic structure

## Logical vocabulary gets special treatment

- ▶ “First Class and Smile Members here“ (airport)
- ▶ Fluffy is angry and Joey is not angry.
  - ▶ What does *and* mean?
  - ▶ What does *not* mean?
- ▶ Every cat is angry.
  - ▶ What does *cat* mean?
  - ▶ What does *every* mean?
- ▶ Is the division into logical and non-logical vocabulary an inherent property of language or an artifact of the system of meaning representation?

# Quantifiers

- ▶ The semantic type of a quantifier is a relation between sets, called the restriction and body (or scope) of the quantifier
- ▶  $\llbracket \text{every} \rrbracket \{ \langle P, Q \rangle \mid P \subseteq Q \}$
- ▶  $\llbracket \text{every cat is angry} \rrbracket$  is True iff  
 $\{ x \mid x \text{ is a cat} \} \subseteq \{ y \mid y \text{ is angry} \}$
- ▶  $\llbracket \text{some} \rrbracket \{ \langle P, Q \rangle \mid P \cap Q \neq \emptyset \}$
- ▶  $\llbracket \text{some cat is angry} \rrbracket$  is True iff  
 $\{ x \mid x \text{ is a cat} \} \cap \{ y \mid y \text{ is angry} \} \neq \emptyset$
- ▶  $\llbracket \text{many} \rrbracket ?$

## Partial evaluation for FOL: Lambda calculus

- ▶ Basic idea: pass around partially evaluated functions
- ▶ feed them to other functions as arguments
- ▶ e.g.  $f : y = x + 2$
- ▶ plug in  $x = 3$ , evaluate to 5
- ▶ or:  $f : z = y * (x + 2)$
- ▶ plug in  $x = 3$ , evaluate to  $f : z = 3y$
- ▶ then can plug in  $y = 2$  and evaluate to 6

# Lambda calculus for semantics

- ▶ Used to evaluate FOL expressions in a compositional manner
- ▶ e.g. constituent by constituent
- ▶ A constituent does not necessarily have a truth value:

$$\lambda x. \text{gave}(Kim, book, x)$$

- ▶ need to hold on to a partially evaluated constituent
- ▶ Converting multi-argument predicates to sequences of single-argument predicates
- ▶ Incrementally accumulates multiple arguments spread over different parts of the tree

# Lambda calculus

- ▶ Form:  $\lambda$  + Variable + FOL expression
- ▶  $\lambda x.P(x)$  (evaluating the expression with respect to  $x$ )
- ▶  $\lambda x.P(x)(A) \rightarrow P(A)$  ( $\lambda$ -reduction; binding a formal parameter to a concrete term)
- ▶  $\lambda x.\lambda y.Near(x, y)$
- ▶  $\lambda x.\lambda y.Near(x, y)(Moscow)$
- ▶  $\lambda y.Near(Moscow, y)$
- ▶  $\lambda y.Near(Moscow, y)(Center)$
- ▶  $Near(Moscow, Center)$

# Lambda calculus and composition

- ▶ One semantic composition rule per syntax rule.
- ▶  $S \rightarrow NP\ VP$   
 $VP'(NP')$
- ▶ Rover barks:  
 $VP$  *bark* is  $\lambda x[\text{bark}'(x)]$   
 $NP$  *Rover* is  $r$   
 $\lambda x[\text{bark}'(x)](r) = \text{bark}'(r)$

# Transitive verbs

Kitty chases Rover

- ▶ Transitive verbs: two arguments (NOTE the order)  
 $\lambda x[\lambda y[\text{chase}'(y, x)]]$
- ▶ VP → Vtrans NP  
 $\text{Vtrans}'(\text{NP}')$
- ▶  $\lambda x\lambda y[\text{chase}'(y, x)](r) = \lambda y[\text{chase}'(y, r)]$
- ▶ S → NP VP  
 $\text{VP}'(\text{NP}')$
- ▶  $\lambda y[\text{chase}'(y, r)](k) = \text{chase}'(k, r)$

## Grammar fragment using lambda calculus

S -> NP VP	VP'(NP')
VP -> Vtrans NP	Vtrans'(NP')
VP -> Vintrans	Vintrans'
Vtrans -> chases	$\lambda x \lambda y [\text{chase}'(y, x)]$
Vintrans -> barks	$\lambda z [\text{bark}'(z)]$
Vintrans -> sleeps	$\lambda w [\text{sleep}'(w)]$
NP -> Kitty	k

## Beyond toy examples . . .

- ▶ Use first order logic where possible (e.g., event variables, next slide).
- ▶ However, First Order Predicate Calculus (FOPC) is sometimes inadequate: e.g., *most*, *may*, *believe*.
- ▶ Quantifier scoping multiplies analyses:  
*Every cat chased some dog:*  
 $\forall x[\text{cat}'(x) \implies \exists y[\text{dog}'(y) \wedge \text{chase}'(x, y)]]$   
 $\exists y[\text{dog}'(y) \wedge \forall x[\text{cat}'(x) \implies \text{chase}'(x, y)]]$
- ▶ Often no straightforward logical analysis  
e.g., Bare plurals such as *Ducks lay eggs*.
- ▶ Non-compositional phrases (multiword expressions): e.g., *red tape* meaning bureaucracy.

## Event variables

- ▶ Allow first order treatment of adverbs and PPs modifying verbs by **reifying** the event.
- ▶ Rover barked
- ▶ instead of  $\text{bark}'(r)$  we have  $\exists e[\text{bark}'(e, r)]$
- ▶ Rover barked loudly
- ▶  $\exists e[\text{bark}'(e, r) \wedge \text{loud}'(e)]$
- ▶ There was an event of Rover barking and that event was loud.

## Computational semantics desiderata (J&M)

- ▶ Verifiability: We must be able to compare the representation to a knowledge base
- ▶ Lack of ambiguity: A semantic representation should have just one interpretation
- ▶ Canonical form: A given interpretation should have just one representation
- ▶ Expressiveness: Must be able to adequately represent a wide range of expressions

## Computational semantics desiderata (Copestake)

- ▶ Expressive Adequacy: The framework must allow linguistic meanings to be expressed correctly
- ▶ Grammatical Compatibility: clear link to other kinds of grammatical information (most notably syntax)
- ▶ Computational Tractability: Process meanings, check semantic equivalence, express relationships between semantic representations straightforwardly
- ▶ Underspecifiability: Allow resolution of partial semantic representations

# Computational semantics

- ▶ Semantic parsing: mapping surface sentence to a semantic representation
- ▶ Should this representation be a structure?
  - ▶ Sentence meaning: probably yes
  - ▶ Speaker meaning: unclear

(But sentence meaning is usually directly involved in speaker meaning)

## Sentence vs. Speaker meaning (Grice 1968)

- ▶ Through experience within our speech communities, we learn (and help create) shared linguistic conventions.
- ▶ These conventions support fairly consistent calculation of **sentence meaning** by different speakers in the same community.
- ▶ The sentence meaning of an utterance (together with its form) serves as a clue which a listener can use to construct his/her representation of the speaker's **speaker meaning**

## Sentence vs. Speaker meaning

*Could you pass me the salt? – No, I couldn't pass you the salt!*

- ▶ Sentence meaning, but not speaker meaning, is compositional
- ▶ Systems attempting to understand speaker meaning directly from surface: resolve the same problems around grammatical structure for each task unlikely to scale

## Semantic compositionality

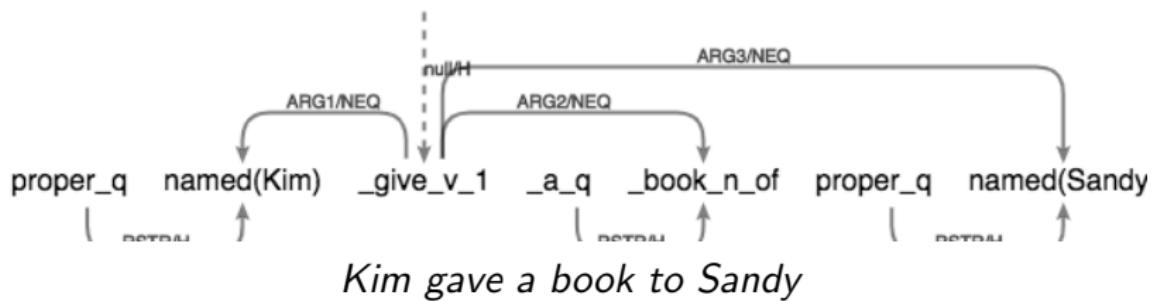
- ▶ The meaning of the whole must be directly assembled from its parts
- ▶ E.g. *Agent/patient* information comes from the *subject/object constituents*
- ▶ The syntactico-semantic formalism must explicitly ensure such connections and assembly

# Compositional layer and syntax-semantics interface

- ▶ Predicate-argument structure
- ▶ Scope of negation and other operators
- ▶ Restriction of quantifiers
- ▶ Modality
- ▶ Tense/aspect/mood
- ▶ Information structure
- ▶ Discourse status of referents of NPs
- ▶ Politeness

# Minimal Recursion Semantics (MRS)

- ▶ An example of a compositional computational semantics approach
- ▶ Copestake et al. (2005)
- ▶ A semantic formalism (not a semantic theory)



# Minimal Recursion Semantics (MRS)

TOP INDEX	$h0$ $e2$
RELS	$\left\{ \begin{bmatrix} proper\_q(0:3) \\ LBL & h4 \\ RSTR & h5 \\ ARG0 & x3 \\ BODY & h6 \end{bmatrix}, \begin{bmatrix} named(0:3) \\ LBL & h7 \\ ARG0 & x3 \\ CARG & Kim \end{bmatrix}, \begin{bmatrix} give\_v\_1(4:8) \\ LBL & h1 \\ ARG3 & x10 \\ ARG2 & x9 \\ ARG1 & x3 \\ ARG0 & e2 \end{bmatrix}, \begin{bmatrix} a\_q(9:10) \\ LBL & h11 \\ RSTR & h12 \\ ARG0 & x9 \\ BODY & h13 \end{bmatrix}, \begin{bmatrix} book\_n\_of(11:15) \\ LBL & h14 \\ ARG1 & i15 \\ ARG0 & x9 \end{bmatrix}, \begin{bmatrix} proper\_q(19:24) \\ LBL & h16 \\ RSTR & h17 \\ ARG0 & x10 \\ BODY & h18 \end{bmatrix} \right\}$ $\left[ \begin{bmatrix} named(19:24) \\ LBL & h19 \\ ARG0 & x10 \\ CARG & Sandy \end{bmatrix} \right]$
HCONS	$\left\{ \begin{bmatrix} qeq \\ HARG & h17 \\ LARG & h19 \end{bmatrix}, \begin{bmatrix} qeq \\ HARG & h12 \\ LARG & h14 \end{bmatrix}, \begin{bmatrix} qeq \\ HARG & h0 \\ LARG & h1 \end{bmatrix}, \begin{bmatrix} qeq \\ HARG & h5 \\ LARG & h7 \end{bmatrix} \right\}$

*Kim gave a book to Sandy*

## Machine translation by transfer

- ▶ Assuming a canonical form for semantic structure, we can generate sentences in one language given a semantic structure which was obtained by parsing a sentence in another language
- ▶ A **symbolic** approach to MT
- ▶ Requires **grammars** for both languages
- ▶ Ensures **precision and grammaticality** of the translations
- ▶ Disadvantage: lack of **robustness**: not every sentence will be translated.

```
ace -g grammar.dat -Tf1 | ace -g grammar.dat -e
```

## MRS: MINIMAL recursion semantics

- ▶ Syntactic structure may sometimes be irrelevant to the truth conditions
- ▶ *fierce black cat* vs *gato negro y feroz*
- ▶ with syntax insufficiently abstracted away, hard to do transfer
- ▶ the LFs produced by the two grammars will look different:
  - a.  $\lambda x[\text{fierce}(x) \wedge (\text{black}(x) \wedge \text{cat}(x))]$
  - b.  $\lambda x[\text{gato}(x) \wedge (\text{negro}(x) \wedge \text{feroz}(x))]$
  - c.  $\lambda x[\text{cat}(x) \wedge (\text{black}(x) \wedge \text{fierce}(x))]$
- ▶  $\text{fierce}(x) \wedge \text{black}(x) \wedge \text{cat}(x)$  – solution?

## Flat semantics: quantifier problem

- ▶ *Every white horse is old*
- ▶ every (x, white (x)  $\wedge$  horse (x), old (x))
- ▶ Flat: every(x), horse(x), old(x), white(x)
- ▶ problem?

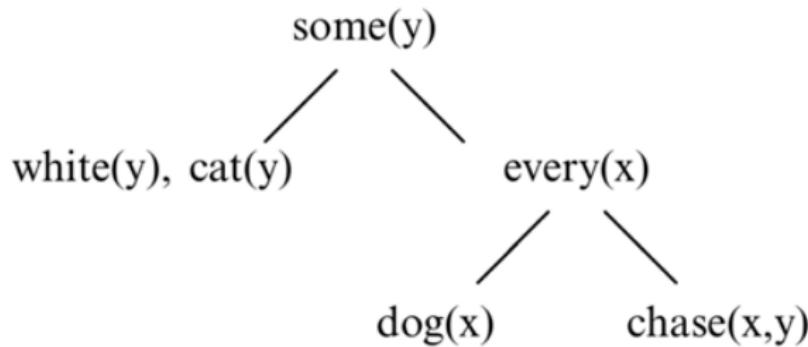
## Flat semantics: quantifier problem

- ▶ *Every white horse is old*
- ▶ every (x, white (x)  $\wedge$  horse (x), old (x))
- ▶ Flat: every(x), horse(x), old(x), white(x)
- ▶ problem?
- ▶ *Every old horse is white?*

## Quantifier scope

*Every dog chases some white cat*

- a.  $\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(x, y)))$
- b.

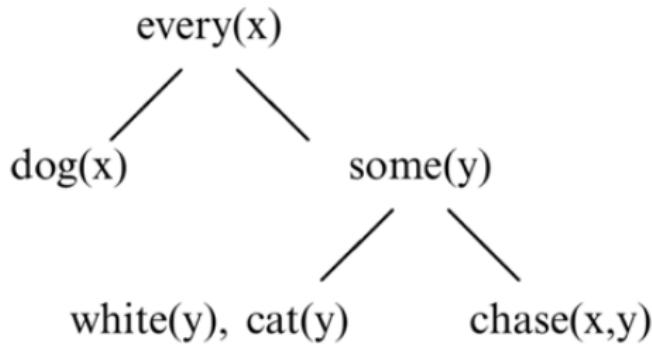


- c.  $h1: \text{every}(x, h3, h4), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y), h5: \text{some}(y, h7, h1), h4: \text{chase}(x, y)$

## Quantifier scope

*Every dog chases some white cat*

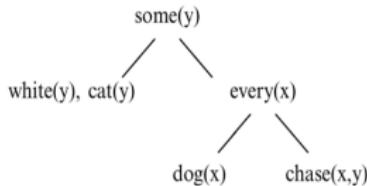
- a.  $\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{chase}(x, y)))$
- b.



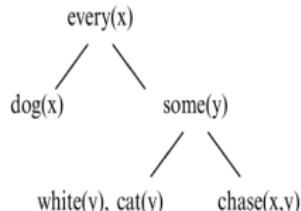
- c.  $h1: \text{every}(x, h3, h5), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y), h5: \text{some}(y, h7, h4), h4: \text{chase}(x, y)$

# Scope underspecification

- a.  $\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(x, y)))$   
b.



- a.  $\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{chase}(x, y)))$   
b.



- c.  $h1: \text{every}(x, h3, h4), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y), h5: \text{some}(y, h7, h1), h4: \text{chase}(x, y)$     c.  $h1: \text{every}(x, h3, h5), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y), h5: \text{some}(y, h7, h4), h4: \text{chase}(x, y)$

$h1: \text{every}(x, h3, h8), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y), h5: \text{some}(y, h7, h9), h4: \text{chase}(x, y)$

- ▶ *Every dog chases some white cat*
- ▶ Can say EITHER  $h9 = h1$  or  $h8 = h5$

# NLP business with semantics

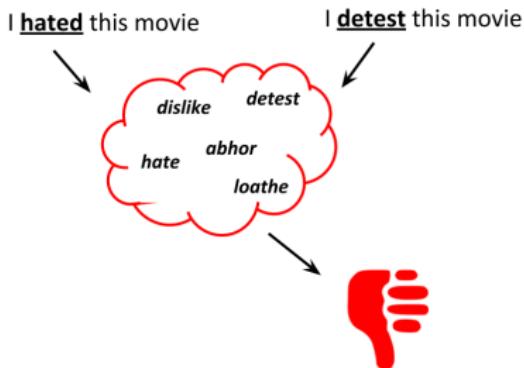
- ▶ Construct knowledge base or model of the world
  - ▶ wikidata, Dbpedia, ConceptNet
  - ▶ TBOX vs ABOX
- ▶ Extract meaning representations from linguistic input
- ▶ Match input to world knowledge
- ▶ Produce replies/take action (or queries) on the basis of the results

# Semantics in NLP

- ▶ The hottest area in NLP right now
- ▶ all kinds of NLU tasks are associated with semantic annotation and semantic parsing
  - ▶ sentiment analysis
  - ▶ dialog systems
  - ▶ news/ads ranking etc.
- ▶ NLP semantics today is usually task-oriented and domain-tailored e.g. a grammar/semantic system for cooking recipes often have nothing to do with theoretical semantics ambitions

# Semantics in NLP

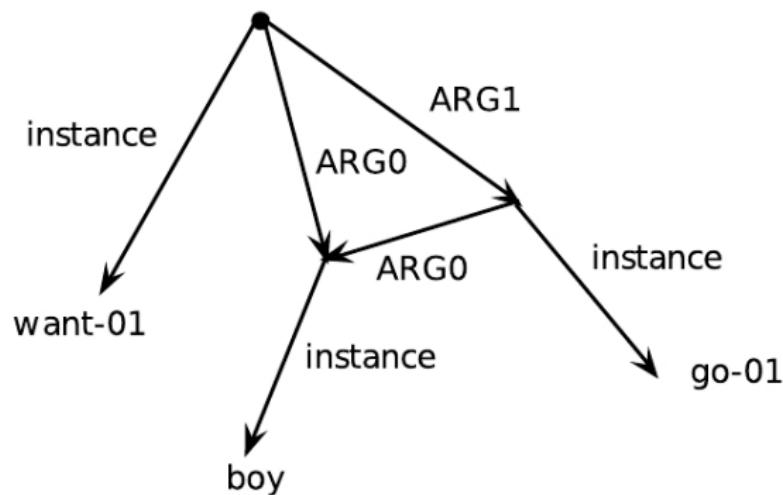
*Positive or negative sentiment?*



- ▶ Linguistic models, syntactic or semantic (or morphological...) tend to be too unwieldy for today's NLP
- ▶ NLP goals: perform well on a task, not necessarily precisely and not necessarily providing explanations
- ▶ Tacit expectation to map directly from *surface* to *speaker meaning*

## (pre-vector space) semantics in NLP

- ▶ e.g. Abstract Meaning Representation (AMR; Banerjee et al., 2013)
- ▶ Note similarities with dependency parse
- ▶ <https://amr.isi.edu>



## AMR: a widely adopted formalism

*"We describe Abstract Meaning Representation (AMR), a semantic representation language in which we are writing down the meanings of thousands of English sentences. We hope that a sembank of simple, whole-sentence semantic structures will spur new work in statistical natural language understanding and generation, like the Penn Treebank encouraged work on statistical parsing."* (Banarescu et al., 2013)

**this came true!!** we look at it as an example of a widely adopted formalism!

## AMR: Pros and Cons

- + very simple representation
- + yields nice immediate results
- + wide paraphrase sets
- Plateaus: like WJS, AMR-based research keeps reusing the same dataset and ends up overfitting to it
- Inconsistent and task-dependent annotation

# Thematic Roles

- ▶ describe semantic roles of verbal arguments
- ▶ capture commonality across verbs
- ▶ e.g. subject *break/open* is AGENT (volitional cause).  
THEME (things affected by action)
- ▶ enables generalization over surface order of arguments
  - ▶ John<sub>agent</sub> broke the window<sub>theme</sub>
  - ▶ The rock<sub>instrument</sub> broke the window<sub>theme</sub>
  - ▶ The window<sub>theme</sub> was broken by John<sub>agent</sub>

# Thematic Role Issues

Hard to produce:

- ▶ standard set of roles. Fragmentation: often need to make more specific. E.g. INSTRUMENT can be subject?
- ▶ standard definition of roles: most AGENTs (animals, volitional, sentient, causal...) but not all.

strategies:

- ▶ generalized semantic roles (proto-agent, proto-patient etc)
- ▶ defined heuristically (propbank)
- ▶ define roles specific to verbs/nouns: [FrameNet](#)

Even if we come up with a standard set of some sort, we will have low inter-annotator agreement. Hard to expect that people will interpret the roles exactly the same

## Sembanks, Propbanks...

- ▶ Representations like AMR can be stored in “sembanks”
- ▶ Compare to treebanks
- ▶ Challenge: **interannotator agreement**
  - ▶ ...is a problem with treebanks, too, unless a grammar is used
  - ▶ is even a bigger problem in sembanks
  - ▶ role-labeling is more vague than syntactic structure
  - ▶ e.g. what kind of granularity?
- ▶ Familiar issues with overfitting

# PropBank

## (22.11) **agree.01**

Arg0: Agreeer

Arg1: Proposition

Arg2: Other entity agreeing

Ex1: [Arg0 The group] *agreed* [Arg1 it wouldn't make an offer].

Ex2: [ArgM-TMP Usually] [Arg0 John] *agrees* [Arg2 with Mary]  
[Arg1 on everything].

## (22.12) **fall.01**

Arg1: Logical subject, patient, thing falling

Arg2: Extent, amount fallen

Arg3: start point

Arg4: end point, end state of arg1

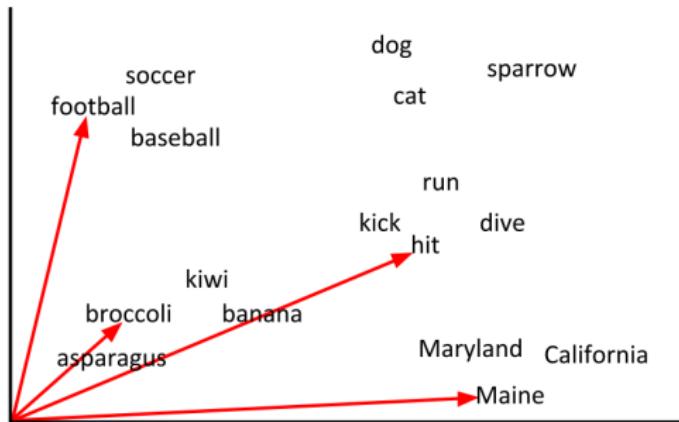
Ex1: [Arg1 Sales] *fell* [Arg4 to \$25 million] [Arg3 from \$27 million].

Ex2: [Arg1 The average junk bond] *fell* [Arg2 by 4.2%].

<http://propbank.github.io>

# Vector space semantics

## Vector space models



- ▶ The core of today's NLP
- ▶ Are word vectors semantic representations?
- ▶ Yes, but not necessarily compositional

## Word senses and Word sense disambiguation

- ▶ A problem from which word vectors arise
- ▶ How close/distinct are the senses of two words?
- ▶ How to determine this computationally?

## Word senses and Word sense disambiguation

- ▶ A problem from which word vectors arise
- ▶ How close/distinct are the senses of two words?
- ▶ How to determine this computationally?
  - ▶ See which contexts they appear in: N-grams! (or, these days, word vectors) → Compute distance → Closer distance = closer meaning?
- ▶ What is one problem with this?

# Word senses and Word sense disambiguation

- ▶ A problem from which word vectors arise
- ▶ How close/distinct are the senses of two words?
- ▶ How to determine this computationally?
  - ▶ See which contexts they appear in: N-grams! (or, these days, word vectors) → Compute distance → Closer distance = closer meaning?
- ▶ What is one problem with this?
  - ▶ This place is very **loud**
  - ▶ This place is very **quiet**
- ▶ Antonyms tend to occur in the same context

# Natural language inference

- ▶ Inference on a knowledge base: convert natural language expression to KB expression, valid inference according to KB.
  - + Precise
  - + Formally verifiable
  - + Disambiguation using KB state
  - Limited domain, requires KB to be formally encodable
- ▶ Language-based inference: does one utterance follow from another?
  - + Unlimited domain
  - +/- Human judgement
  - /+ Approximate/imprecise
- ▶ Both approaches may use logical form of utterance.

## Lexical meaning and meaning postulates

- ▶ Some inferences validated on logical representation directly, most require lexical meaning.
- ▶ meaning postulates: e.g.,

$$\forall x[\text{bachelor}'(x) \rightarrow \text{man}'(x) \wedge \text{unmarried}'(x)]$$

- ▶ usable with compositional semantics and theorem provers
- ▶ e.g. from 'Kim is a bachelor', we can construct the LF  $\text{bachelor}'(\text{Kim})$  and then deduce  $\text{unmarried}'(\text{Kim})$
- ▶ Problematic in general, OK for narrow domains or micro-worlds.

## Recognising Textual Entailment (RTE) shared tasks

T: The girl was found in Drummondville earlier this month.

H: The girl was discovered in Drummondville.

- ▶ DATA: pairs of text (T) and hypothesis (H). H may or may not follow from T.
- ▶ TASK: label TRUE (if follows) or FALSE (if doesn't follow), according to human judgements.

## Many approaches

- ▶ <https://aclanthology.org/S14-2125/> and  
<https://aclanthology.org/W15-2205/>
- ▶ Natural Logic <https://aclanthology.org/W09-3714/>
- ▶ Combining symbolic and ML/DL  
<https://aclanthology.org/2020.coling-demos.9/> and  
<https://aclanthology.org/P16-2079/>

## RTE using logical forms

- ▶ T sentence has logical form  $T'$ , H sentence has logical form  $H'$
- ▶ If  $T' \implies H'$  conclude TRUE, otherwise conclude FALSE.

**T** The girl was found in Drummondville earlier this month.

**T'**  $\exists x, u, e[\text{girl}'(x) \wedge \text{find}'(e, u, x) \wedge \text{in}'(e, \text{Drummondville}) \wedge \text{earlier-this-month}'(e)]$

**H** The girl was discovered in Drummondville.

**H'**  $\exists x, u, e[\text{girl}'(x) \wedge \text{discover}'(e, u, x) \wedge \text{in}'(e, \text{Drummondville})]$

**MP**  $[\text{find}'(x, y, z) \implies \text{discover}'(x, y, z)]$

- ▶ So  $T' \implies H'$  and we conclude TRUE

## More complex examples

- T:** Four Venezuelan firefighters who were traveling to a training course in Texas were killed when their sport utility vehicle drifted onto the shoulder of a highway and struck a parked truck.
- H:** Four firefighters were killed in a car accident.

Systems using logical inference are not robust to missing information: simpler techniques can be effective (partly because of choice of hypotheses in RTE).

## More examples

### Ex 1

T: Clinton's book is not a big seller here.

H: Clinton's book is a big seller.

### Ex 2

T: After the war the city was briefly occupied by the Allies and then was returned to the Dutch.

H: After the war, the city was returned to the Dutch.

### Ex 3

T: Lyon is actually the gastronomic capital of France.

H: Lyon is the capital of France.

### Ex 4

T: The young boys are playing outdoors and the man is smiling nearby

H: The kids are playing outdoors near a man with a smile