Introdução NLP e IR

Alexandre Rademaker[1]

FGV/EMAp

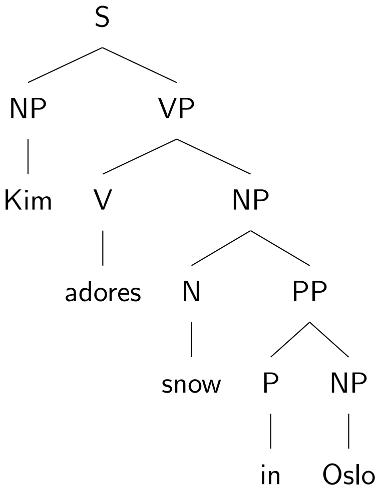
October 13, 2022

[1] based on Olga Zamaraeva slides

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?



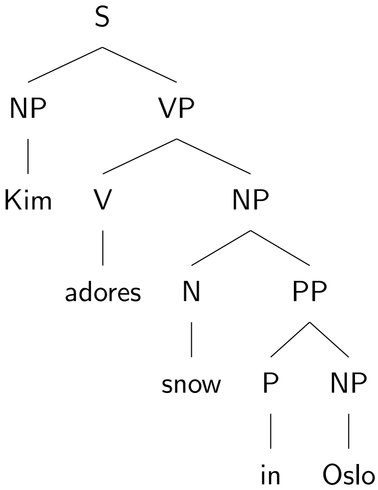
Why represent meaning computationally?

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:

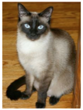
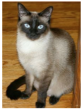
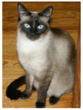
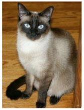
How could we put this tree in correspondence to a model of the world?



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)

- [[Fluffy]] = 

- [[angry]] = { x | x is angry } = {  }

- [[Fluffy is angry]] = 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

- 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
- $[\![\textit{every}]\!]$ $\{ <P,Q> \mid P \subseteq Q\}$
- $[\![$every cat is angry$]\!]$ is True iff $\{ x \mid x$ is a cat $\} \subseteq \{ y \mid y$ is angry $\}$
- $[\![\textit{some}]\!]$ $\{ <P,Q> \mid P \cap Q \neq \emptyset\}$
- $[\![$some cat is angry$]\!]$ is True iff $\{ x \mid x$ is a cat $\} \cap \{ y \mid y$ is angry $\} \neq \emptyset$
- $[\![\textit{many}]\!]$?

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

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

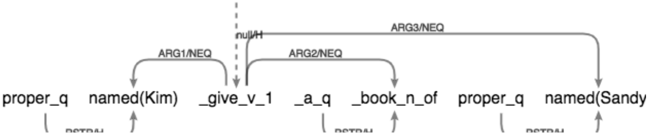
- The meaning of the whole must be directly assembled from its parts
- E.g. *Agent/patient* information comes from the *subject/object constituents*
- The syntactico-semantic formalism must explicitly ensure such connections and assembly

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)



Kim gave a book to Sandy

Minimal Recursion Semantics (MRS)

$$
\left[\begin{array}{ll}
\text{TOP} & h0 \\
\text{INDEX} & e2 \\
\text{RELS} & \left\langle
\begin{bmatrix} \multicolumn{2}{l}{\mathit{proper\_q}\langle 0{:}3\rangle} \\ \text{LBL} & h4 \\ \text{RSTR} & h5 \\ \text{ARG0} & x3 \\ \text{BODY} & h6 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{named}\langle 0{:}3\rangle} \\ \text{LBL} & h7 \\ \text{ARG0} & x3 \\ \text{CARG} & \mathit{Kim} \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{\_give\_v\_1}\langle 4{:}8\rangle} \\ \text{LBL} & h1 \\ \text{ARG3} & x10 \\ \text{ARG2} & x9 \\ \text{ARG1} & x3 \\ \text{ARG0} & e2 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{\_a\_q}\langle 9{:}10\rangle} \\ \text{LBL} & h11 \\ \text{RSTR} & h12 \\ \text{ARG0} & x9 \\ \text{BODY} & h13 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{\_book\_n\_of}\langle 11{:}15\rangle} \\ \text{LBL} & h14 \\ \text{ARG1} & i15 \\ \text{ARG0} & x9 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{proper\_q}\langle 19{:}24\rangle} \\ \text{LBL} & h16 \\ \text{RSTR} & h17 \\ \text{ARG0} & x10 \\ \text{BODY} & h18 \end{bmatrix}
\right. \\
 & \left.
\begin{bmatrix} \multicolumn{2}{l}{\mathit{named}\langle 19{:}24\rangle} \\ \text{LBL} & h19 \\ \text{ARG0} & x10 \\ \text{CARG} & \mathit{Sandy} \end{bmatrix}
\right\rangle \\
\text{HCONS} & \left\langle
\begin{bmatrix} \multicolumn{2}{l}{\mathit{qeq}} \\ \text{HARG} & h17 \\ \text{LARG} & h19 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{qeq}} \\ \text{HARG} & h12 \\ \text{LARG} & h14 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{qeq}} \\ \text{HARG} & h0 \\ \text{LARG} & h1 \end{bmatrix},
\begin{bmatrix} \multicolumn{2}{l}{\mathit{qeq}} \\ \text{HARG} & h5 \\ \text{LARG} & h7 \end{bmatrix}
\right\rangle
\end{array}\right.
$$

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.

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. $\lambda x[\text{cat}(x) \wedge (\text{black}(x) \wedge \text{fierce}(x))]$

- $\mathit{fierce}(x) \wedge \mathit{black}(x) \wedge \mathit{cat}(x)$ – 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

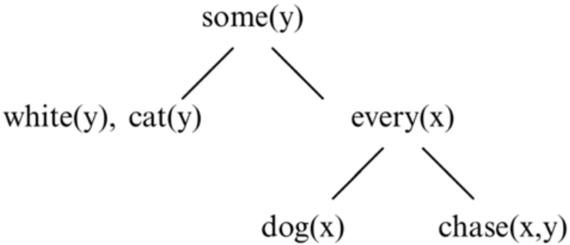
- *Every white horse is old*
- every (x, white (x) $\wedge$ horse (x), old (x))
- Flat: every(x), horse(x), old(x), white(x)
- problem?
- *Every old horse is white*?

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.



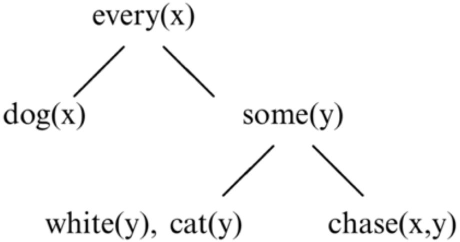
c. $h1\colon \text{every}(x, h3, h4), h3\colon \text{dog}(x), h7\colon \text{white}(y), h7\colon \text{cat}(y),$
$h5\colon \text{some}(y, h7, h1), h4\colon \text{chase}(x, y)$

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.

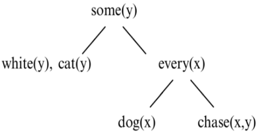


c. $h1: \text{every}(x, h3, h5), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y),$
$h5: \text{some}(y, h7, h4), h4: \text{chase}(x, y)$

Scope underspecification

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

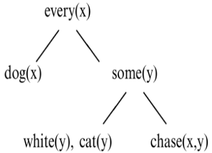
b.



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

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

b.



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

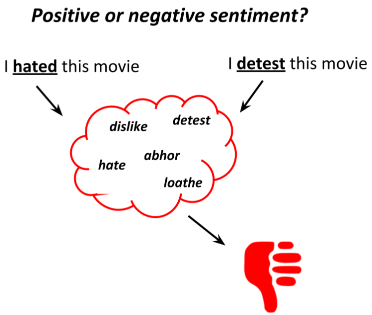
$h1$: $\mathrm{every}(x, h3, h8)$, $h3$: $\mathrm{dog}(x)$, $h7$: $\mathrm{white}(y)$, $h7$: $\mathrm{cat}(y)$, $h5$: $\mathrm{some}(y, h7, h9)$, $h4$: $\mathrm{chase}(x, y)$

- *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
- Extract meaning representations from linguistic input
- Match input to world knowledge
- Produce replies/take action on the basis of the results

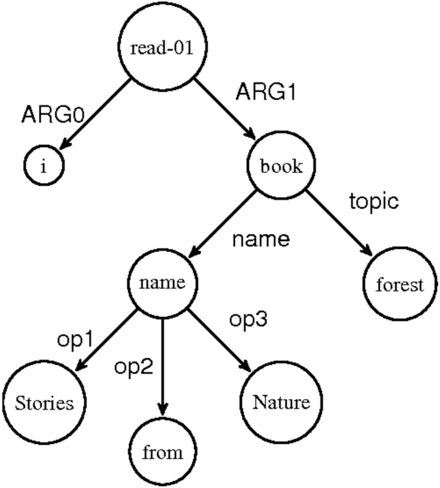
Semantics in NLP



- 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; Banarescu et al., 2013)
- Note similarities with dependency parse



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)

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**

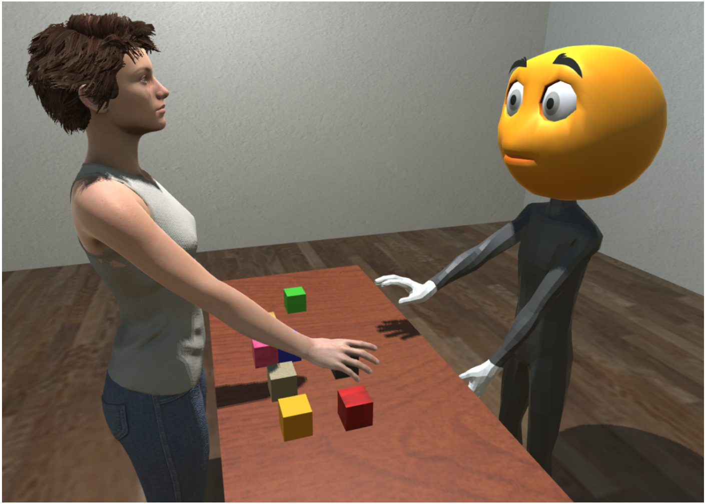
Arg0: Agreer
Arg1: Proposition
Arg2: Other entity agreeing

Ex1: [$_{\text{Arg0}}$ The group] *agreed* [$_{\text{Arg1}}$ it wouldn't make an offer].
Ex2: [$_{\text{ArgM-TMP}}$ Usually] [$_{\text{Arg0}}$ John] *agrees* [$_{\text{Arg2}}$ with Mary] [$_{\text{Arg1}}$ on everything].

(22.12) **fall.01**

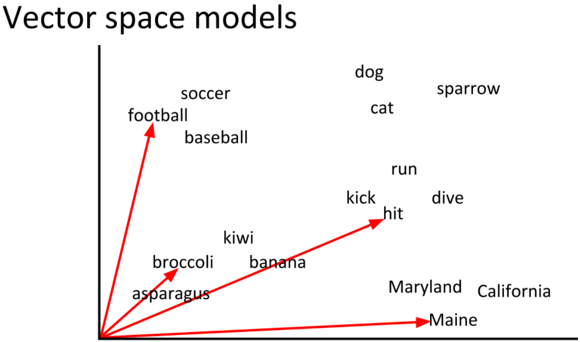
Arg1: Logical subject, patient, thing falling
Arg2: Extent, amount fallen
Arg3: start point
Arg4: end point, end state of arg1
Ex1: [$_{\text{Arg1}}$ Sales] *fell* [$_{\text{Arg4}}$ to \$25 million] [$_{\text{Arg3}}$ from \$27 million].
Ex2: [$_{\text{Arg1}}$ The average junk bond] *fell* [$_{\text{Arg2}}$ by 4.2%].

AI, robotics and grounded reasoning



- *There is exactly one yellow object touching the wall*
- (object-count-equals (yellow (touch-wall all-objects)) 1)
- *natural* language?..

Vector space semantics (next lecture)



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