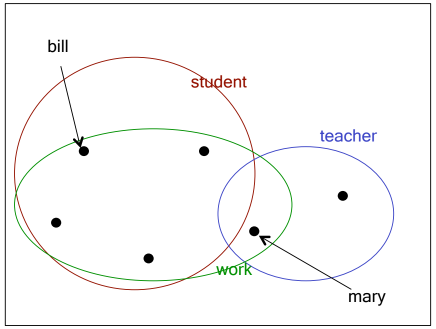
Foundations of Language Science and Technology

Semantics 2

Manfred Pinkal
Saarland University

An example model structure



Interpretation of Atomic Formulae

- An interpretation function $[[\ ]]^{M,g}$ recursively assigns semantic values $[[\ \alpha\ ]]^{M,g}$ to all expressions α with respect to a model structure and a variable assignment g.
- Interpretation of terms:

  $[[\ c\ ]]^{M,g} = V_M(c)$ for all individual constants c

  $[[\ x\ ]]^{M,g} = g(x)$
- Interpretation of atomic expressions:

  $[[\ R(t_1, ..., t_n)\ ]]^{M,g} = 1$ iff $([[t_1]]^{M,g}, ..., [[t_n]]^{M,g}) \in V_M(R)$

  $[[\ t_1 = t_2\ ]]^{M,g} = 1$ iff $[[t_1]]^{M,g} = [[t_2]]^{M,g}$

Interpretation of connectives

$[[\ \neg\varphi\ ]]^{M,g} = 1$ iff $[[\ \varphi\ ]]^{M,g} = 0$

$[[\ \varphi \wedge \psi\ ]]^{M,g} = 1$ iff $[[\ \varphi\ ]]^{M,g} = 1$ and $[[\ \psi\ ]]^{M,g} = 1$

$[[\ \varphi \vee \psi\ ]]^{M,g} = 1$ iff $[[\ \varphi\ ]]^{M,g} = 1$ or $[[\ \psi\ ]]^{M,g} = 1$

$[[\ \varphi \rightarrow \psi\ ]]^{M,g} = 1$ iff $[[\ \varphi\ ]]^{M,g} = 0$ or $[[\ \psi\ ]]^{M,g} = 1$

$[[\ \varphi \leftrightarrow \psi\ ]]^{M,g} = 1$ iff $[[\ \varphi\ ]]^{M,g} = [[\ \psi\ ]]^{M,g}$

- Connectives in predicate logic are truth-functional: Their truth-value is completely determined by the truth-values of their constituent clauses.
- The interpretation of connectives can be represented by truth-tables.

Truth Tables for Connectives

A	¬A
0	1
1	0

A	B	(A∧B)
0	0	0
0	1	0
1	0	0
1	1	1

A	B	(A∨B)
0	0	0
0	1	1
1	0	1
1	1	1

A	B	(A→B)
0	0	1
0	1	1
1	0	0
1	1	1

A	B	(A↔B)
0	0	1
0	1	0
1	0	0
1	1	1

Quantifier Interpretation– Preliminary!

- A preliminary formulation of a general interpretation function for quantified formulae:

  $[[\exists xA]]^{M,g} = 1$ iff

  there is at least one variable assignment g' such that $[[A]]^{M,g'} = 1$

  $[[\forall xA]]^{M,g} = 1$ iff

  $[[A]]^{M,g'} = 1$ for all variable assignments g'.

"*Bill works*": Truth conditions

- Input sentence is: *"Bill works"*
- Semantic construction returns the formula work(bill)
- Predicate logic interpretation gives the truth conditions:

  $[[work(bill)]]^{M,g} = 1$

  iff $[[bill]]^{M,g} \in V_M(work)$

  iff $V_M(bill) \in V_M(work)$
- "work(bill)" is true in a model structure M iff the object denoted by "bill" in M is member of the set denoted by "work" in M.

Note that the computation is valid irrespective of the choice of a special variable assignment function.

"*Bill works*": Truth values

$[[work(bill)]]^{M,g} = 1$ iff $V_M(bill) \in V_M(work)$

Let M=M1:

$V_{M1}(bill) \in V_{M1}(work)$,

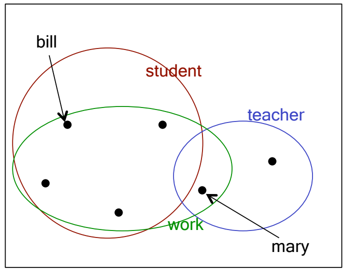
so $[[work(bill)]]^{M1,g} = 1$

Let M=M2:

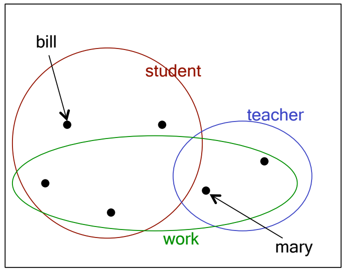
$V_{M2}(bill) \notin V_{M2}(work)$,

so $[[work(bill)]]^{M2,g} = 0$

M1



M2



"*Every student works*": Truth conditions

- *"Every student works"* $\Rightarrow \forall x(\text{student}(x) \rightarrow \text{work}(x))$
- $[[\forall x(\text{student}(x) \rightarrow \text{work}(x))]]^{M,g} = 1$

  iff $[[\text{student}(x) \rightarrow \text{work}(x)]]^{M,g'} = 1$ for every variable assignment g'

  $[[\text{student}(x) \rightarrow \text{work}(x)]]^{M,g'} = 1$

  iff $[[\text{student}(x)]]^{M,g'} = 0$ or $[[\text{work}(x)]]^{M,g'} = 1$

  iff $[[x]]^{M,g'} \notin V_M(\text{student})$ or $[[x]]^{M,g'} \in V_M(\text{work})$

  iff $g'(x) \notin V_M(\text{student})$ or $g'(x) \in V_M(\text{work})$
- $\forall x(\text{student}(x) \rightarrow \text{work}(x))$ is true in M iff for every variable assignment g': $g'(x) \notin V_M(\text{student})$ or $g'(x) \in V_M(\text{work})$

  Note that again the computation is valid irrespective of the choice of a special variable assignment function g: This time because all variables occuring in the original formula are bound by the universal quantifier, the assignment g is overwritten (by g') during interpretation.

"*Every student works*": Truth conditions continued

- $\forall x(\text{student}(x) \rightarrow \text{work}(x))$ is true in M iff for every variable assignment g': $g'(x) \notin V_M(\text{student})$ or $g'(x) \in V_M(\text{work})$
- In other words: for every variable assignment g':

  If $g'(x) \in V_M(\text{student})$, then $g'(x) \in V_M(\text{work})$
- Which is equivalent to saying that: for every $a \in U_M$:

  If $a \in V_M(\text{student})$, then $a \in V_M(\text{work})$
- Which again is equivalent to: $V_M(\text{student}) \subseteq V_M(\text{work})$

Thus:

- $\forall x(\text{student}(x) \rightarrow \text{work}(x))$ is true in M iff $V_M(\text{student}) \subseteq V_M(\text{work})$

"*Every student works*": Truth values

$[[\forall x(\text{student}(x) \rightarrow \text{work}(x))]]^{M,g} = 1$ iff $V_M(\text{student}) \subseteq V_M(\text{work})$

Let M=M1:

$V_{M1}(\text{student}) \subseteq V_{M1}(\text{work})$,

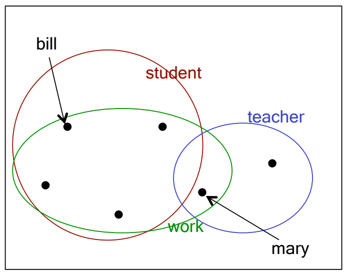
so $[[\forall x(\text{student}(x) \rightarrow \text{work}(x))]]^{M1,g} = 1$

Let M=M2:

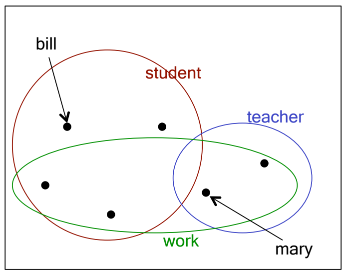
$V_{M2}(\text{student}) \not\subseteq V_{M2}(\text{work})$,

so $[[\forall x(\text{student}(x) \rightarrow \text{work}(x))]]^{M2,g} = 0$

M1



M2



Truth, Satisfaction, Entailment

- A formula A is true in model structure M

  iff $[[A]]^{M,g} = 1$ for every variable assignment g.
- A formula A is valid ($\models A$)
  - iff A is true in every model structure.
- A set of formulas Γ entails formula A ($\Gamma \models A$) iff A is true in in every model structure M in which all $A \in \Gamma$ are true.

  If all $A \in \Gamma$ are true in a model structure M, we also say that M satisfies (or: simultaneously satisfies) Γ.

Computing Entailment

- Does Γ = {student(bill), ∀x(student(x) → work(x))} entail work(bill) ?
- student(bill), ∀x(student(x) → work(x)) ⊨ work(bill) ?

- For every M :
  student(bill) is true in M iff V_M(bill) ∈ V_M(student)
  ∀x(student(x) → work(x)) is true in M iff V_M(student) ⊆ V_M(work)
- From V_M(bill) ∈ V_M(student) and V_M(student) ⊆ V_M(work), it follows that V_M(bill) ∈ V_M(work) (basic set-theoretic law)
- Now, V_M(bill) ∈ V_M(work) is just the truth condition for work(bill).
- Therefore: In every model structure M satisfying student(bill) and ∀x(student(x)→work(x)), the formula work(bill) is true:
  Entailment holds.

Entailment and Deduction

- We just have proved:
  student(bill), ∀x(student(x) → work(x)) ⊨ work(bill)
- We did this independent of the choice of non-logical constants. Thus, the result can be generalized to arbitrary individual constants b and one-place predicates F, G:
  F(b), ∀x(F(x) → G(x)) ⊨ G(b)

- On the basis of this result, we can safely use the deduction rule:
  F(b), ∀x(F(x) → G(x)) ⊢ G(b) (one of the Aristotelian syllogisms).
- Whenever we see formulas of the form F(b) and ∀x(F(x) → G(x)) in a database, we can apply the rule and add G(b), without doing any semantic interpretation.

Deduction Calculi

- Computing entailment and other logical concepts through semantic interpretation is inefficient and in many cases infeasible.
- Deduction calculi (or proof theoretic systems) provide a strictly syntactic way of checking entailment, through rewrite of logical formulae.
- Formula A is derivable (deducible) from a set of formulas Γ ($\Gamma \vdash$ A) in a given deduction system, iff one can obtain A starting from Γ, by using deduction rules and possibly axioms of that deduction system.

Soundness and Completeness

- Truth-conditional interpretation of the logical formalism enable us to determine whether some given deduction system is
  - sound, i.e., derives only those formula A from a set of premisses Γ which are entailed by Γ.
  - complete, i.e., allows to derive all formulae entailed by Γ.
- In short:
  - Soundness: If $\Gamma \vdash$ A, then $\Gamma \models$ A.
  - Completeness: If $\Gamma \models$ A, then $\Gamma \vdash$ A.
- Sound and complete deduction systems derive all and only the truth-conditionally correct entailments.

A simple Deduction Example

- Actually, rules used in the standard deduction calculi (and proved to be sound or correct) are even more general.
- The following simple derivation uses the rules of Universal Instantiation and Modus Ponens, which in combination have the same effect as the syllogism above.

(1) ∀d (student(d)→work(d))	Premise
(2) student(bill)→work(bill)	Universal Instantiation: $\forall x A \vdash A\ [x/a]$
(3) student(bill)	Premise
(4) work(bill)	Modus Ponens: $A, A \rightarrow B \vdash B$ (2), (3)

Theorem Provers

- The problem of FOL entailment checking is very hard: It is even undecidable.
- However, there are automated deduction systems available (called theorem provers, because the original motivation was mathematical theorem proving), which have been optimized through the decades, and have become very efficient.
- So, efficiency is not the problem ...

The Story

- Modelling natural-language inference as deduction in a framework of truth-conditionally interpreted logic appears intuitive and straightforward.
- But: Logical methods are expensive and lack robustness and coverage.

Problems with logic-based inference

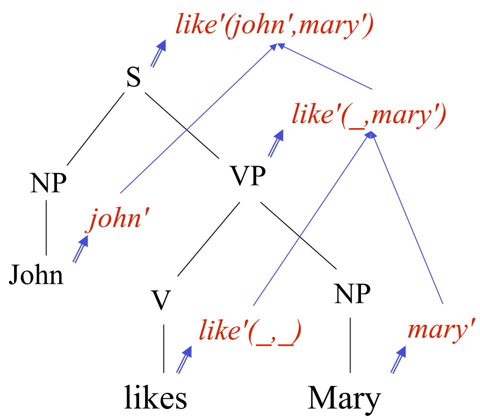
Modelling of inference with logical deduction

- requires full translation of natural-language text into precise first-order representations
- requires additional axioms encoding lexical-semantic and common-sense knowledge
- requires special techniques to process common-sense knowledge.

Translating text to FOL – Good News

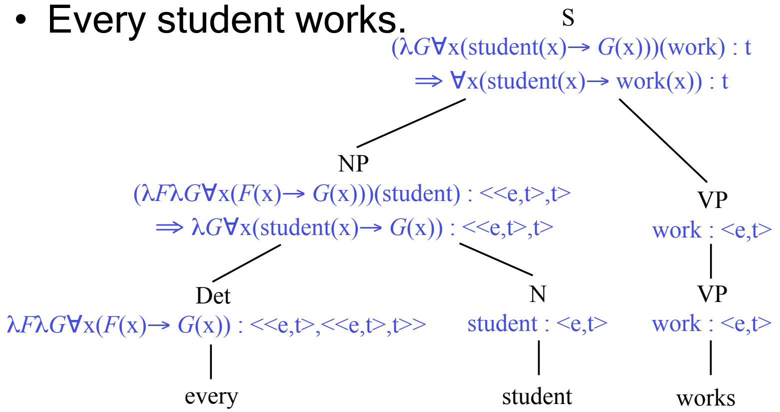
- Principle of compositionality ("Frege's Principle"): The meaning of a complex expression (a sentence) is determined by the meanings of its parts (its words) plus syntactic information.
- The basic task of semantic composition or semantic construction has been solved by type-theoretic semantics (typed lambda-calculus, Montague grammar).
- Wide-coverage grammars are available, which provide deep syntactic information (e.g., in the HPSG or LFG framework).
- These grammar frameworks also provide efficient implementations of the semantic composition process, in part using first-order unification.

Basic Semantic Construction



Semantic Costruction using λ–conversion

- Every student works.



A more complex example

Premise:

- *Several airlines polled saw costs grow more than expected, even after adjusting for inflation*

Hypothesis:

- *Some companies reported cost increases.*

A more complex example

Several airlines polled saw costs grow more than expected.

$\exists^{several}x(airline(x) \wedge \exists y(poll(y,x) \wedge see(x, \exists z \exists d(costs(z) \wedge degree(d) \wedge grow(z,d) \wedge \exists d'(degree(d') \wedge expect(x, grow(z,d')) \wedge d'>d)))$

$$\begin{aligned}&\exists^{several}x(airline(x) \wedge\\ &\quad \exists y(poll(y,x) \wedge\\ &\quad\quad see(x, \exists z \exists d(costs(z) \wedge\\ &\quad\quad\quad degree(d) \wedge\\ &\quad\quad\quad grow(z,d) \wedge\\ &\quad\quad\quad \exists d'(degree(d') \wedge\\ &\quad\quad\quad\quad expect(x, grow(z,d')) \wedge\\ &\quad\quad\quad\quad d'>d)))))\end{aligned}$$

Davidsonian Event Semantics

John kicked Bill

- Standard Predicate-Logic Representation, 1 argument position per syntactic complement:
  kick(john, bill)
- Davidsonian Representation, 1 additional existentially bound event argument :
  $\exists e$ kick(e, john, bill)
- Neo-Davidsonian Representation: Event Verbs are represented by 1-place predicates + thematic role information encoded in 2-place relations:
  $\exists e\ (kick(e) \wedge agent(e, john) \wedge patient(e, bill))$

A more complex example

Several airlines polled saw costs grow more than expected.

$$\begin{aligned}&\exists^{several} \exists y\ \exists z \exists e_1 \exists e_2 \exists e_3 \exists e_4 \exists e_5 \exists d \exists d'\ (airline(x)\\ &\quad \wedge poll(e_1) \wedge agent(e_1,y) \wedge patient(e_1,x)\\ &\quad \wedge see(e_2) \wedge agent(e_2,x) \wedge patient(e_2, e_3)\\ &\quad \wedge grow(e_3) \wedge costs(z) \wedge agent(e_3,z) \wedge degree(e_3,d)\\ &\quad \wedge expect(e_4) \wedge agent(e_4,x) \wedge patient(e_4, e_5)\\ &\quad \wedge grow(e_5) \wedge agent(e_5,z) \wedge degree(e_5,d')\\ &\quad \wedge d'>d)\end{aligned}$$

Translating text to FOL – Bad News

- Full translation of natural-language text into precise logical representations is tremendously difficult.
- The basic semantic construction task is a challenge, but doable. Two related tasks are really hard:
- Semantic Resolution: Finding the contextually appropriate reading for natural-language sentence, which typically comes with a vast number of possible readings.
- Extending (and interpreting!) the representation framework to cover all "ontologically difficult" cases (which abound in every ordinary newspaper text).

The Knowledge Bottleneck

- Inference typically requires additional axioms encoding lexical-semantic and common-sense knowledge.
- Examples:

Premise: *Several airlines polled saw costs grow more than expected, even after adjusting for inflation*
Hypothesis: *Some companies reported cost increases.*

The Knowledge Bottleneck

- Inference typically requires additional axioms encoding lexical-semantic and common-sense knowledge.
- Examples:

Premise: *Several airlines polled saw costs grow more than expected, even after adjusting for inflation*
Hypothesis: *Some companies reported cost increases.*

The Knowledge Bottleneck

- Inference typically requires additional axioms encoding lexical-semantic and common-sense knowledge.
- Examples:

Premise: *Several airlines polled saw costs grow more than expected, even after adjusting for inflation*
Hypothesis: *Some companies reported cost increases.*

Premise: *Security authorities have declared a state of maximum emergency in Guatemala, which is located directly in the path of the hurricane.*
Hypothesis: *There is a state of maximum emergency in Guatemala because of the hurricane.*

The Knowledge Bottleneck

- Inference typically requires additional axioms encoding lexical-semantic and common-sense knowledge.
- Large lexical-semantic resources are available:
  - WordNet provides knowledge about semantic relations, in particular the hyponymy or sub-concept relation like *airline - company*
  - FrameNet provides knowledge of the kind that relates the constructions *"costs grow"* and *"cost increases"*
- Wide-coverage common-sense knowledge bases are missing.

Default Knowledge

- Common-sense knowledge is typically uncertain, only valid by default. An example:
- From *Tweety is a bird*
  we infer *Tweety can fly* as a typical property of birds.
  But: Inference holds only in the absence of more specific contradictory knowledge like *Tweety is a chicken.*
  Learning that Tweety is a chicken has the effect that the inference must be removed
- Logical inference is generally monotonic. Knowledge can be added, but never removed.
- To model inference with uncertain knowledge, we have to use special systems for non-monotonic reasoning.

Problems with logic-based inference

- No success stories from language technology applications
- Poor performance in modeling natural-language inference mechanisms (robustness, coverage)
- Is logic an appropriate basis to represent meaning and model inference, after all?

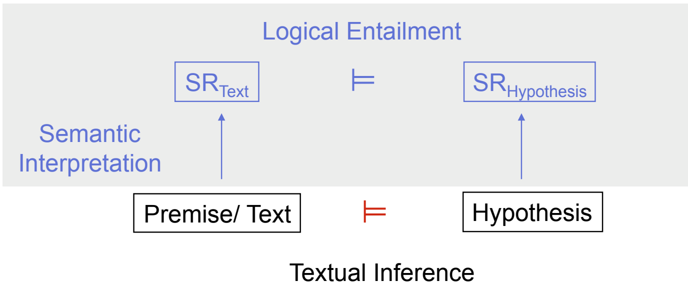
Textual Inference

... also called “textual entailment”, as an alternative to the logic-based inference concept.

„We say that *T entails H* if the meaning of *H* can be inferred from the meaning of *T*, as would typically be interpreted by people. This somewhat informal definition is based on (and assumes) *common human understanding of language* as well as *common background knowledge.”*

Dagan, Glickmann, Magnini, *RTE 2004 Workshop Proceedings*

Textual Inference



Shallow Inference Checking

Logical Entailment: ✓

∀d (dolphin(d)→mammal(d) ∧¬fish(d)) ⊨ ∀d (dolphin(d)→mammal(d))

Semantic Interpretation

Dolphins are mammals, not fish. They are warm blooded like man, and give birth to one baby called a calf at a time. At birth a bottlenose dolphin calf ... ⊨ Dolphins are mammals.

The Story

- Modelling natural-language inference as deduction in a framework of truth-conditionally interpreted logic appears intuitive and straightforward.
- But: Logical methods are expensive and lack robustness and coverage.
- Corpus-based statistical methods for modelling inference are inexpensive and have no coverage problem.

Basic idea: Approximating inference by similarity between H and P.

Word Overlap

String match:

P: *Dolphins are mammals, not fish.*

H: *Dolphins are mammals.*

Word Overlap:

P: *William H. Seward served as Secretary of State under President Abraham Lincoln.*

H: *William H. Seward was Lincoln's Secretary of State*

Word Overlap

String match:

P: *Dolphins are mammals, not fish.*

H: *Dolphins are mammals.*

Word Overlap:

P: *William H. Seward served as Secretary of State under President Abraham Lincoln.*

H: *William H. Seward was Lincoln's Secretary of State*

P-H-relatedness: $\frac{\text{\# of words in H occurring in P}}{\text{length of H}}$