Semantics and Computational Semantics

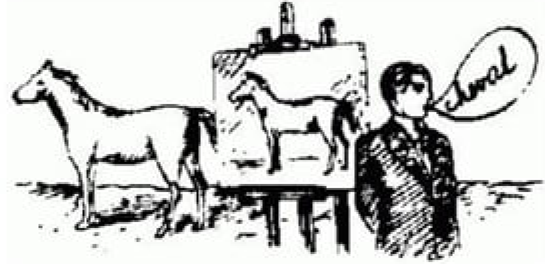
Outline

- Representation and Semantics
- Computation and Compositional Semantics
  - Logic
  - Unification
- Conclusions

What is Semantics?

Generally speaking…

semantics is the study of meaning that is used for understanding human expressions through language



- *Semantics* is the study of meaning:
  accounts for links between words and the world.

- Semantics is the study of how meaning is conveyed through signs and language.
  - Denotations are the literal or primary meanings of words.
  - Connotations are ideas or feelings that a word invokes for a person in addition to its literal or primary meaning.
- Semantics focuses on the relation between signifiers (like words, phrases, signs, and symbols) and what they stand for, ie their denotations (concepts) and connotations (semantic analysis).

Comput(er)-ational Semantics

- **Semantics** is the study of meaning in language.
- **Computer science** is the study of precise descriptions of finite processes;
- **Thus**, computational semantics embraces any project that approaches the phenomenon of meaning by way of tasks that can be performed by following definite sets of mechanical instructions.

We can start by saying that…

”The aim of computational semantics is to find techniques for automatically *constructing semantic representations* for expressions of human language, representations that be used to perform *inference*.”

Blackburn and Bos (2003).

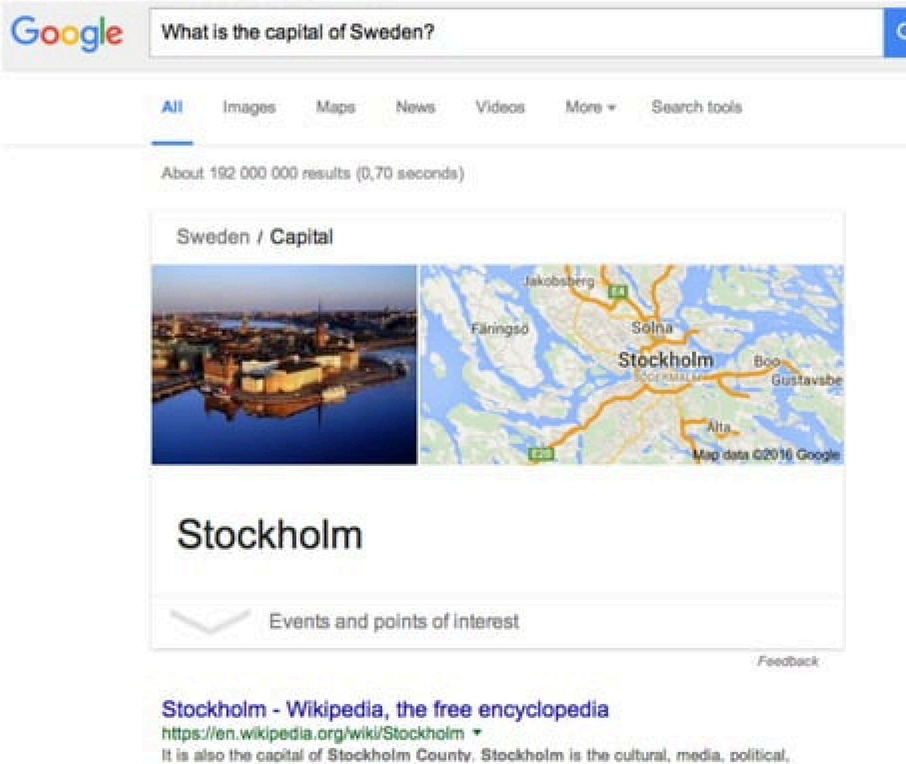
Inference = reasoning

Why do we need ”meaning” in Language Technology?

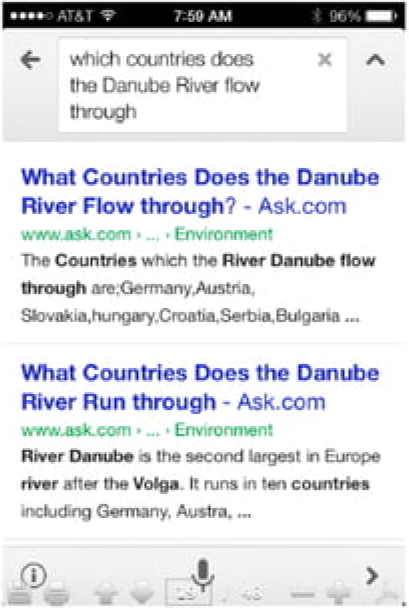
- The short answer is: because ”intelligent” applications must ”understand” in order to act.

Ex 1: Google - What is the capital of Sweden?

- Answer:



Ex 2: Siri - Which countries does the Danube river flows through?



Ex 3: Facebook Graph Search

Facebook Graph Search was a **semantic search engine** that was introduced by Facebook in March 2013.

It was designed to give answers to user natural language queries rather than a list of links."

Examples:

"Friends who Like Star Wars and Harry Potter"

"*photos of my friends taken at National Parks*"

https://en.wikipedia.org/wiki/Facebook_Graph_Search

Understanding Meaning

If an agent (eg. a robot) hears a sentence and act accordingly, the agent is said to understand the sentence:

Example: *Leave the book on the table*

Understanding may involve **inference**: Which book? Which table?

So, understanding may involve some kind of **reasoning**...

NLP/LT and Semantics

- Not all LT-based applications require semantics.
  - Ex: taggers, parsers, etc.

- Much can be done with shallow semantics.
  - Ex: word sense disambiguation, etc.

- For more complex tasks (more "intelligent" tasks") semantic reasoning is needed.
  - Ex: QAnswering, Info Extraction, Ontology Creation, etc.

Traditionally…

…reasoning is done by using some kind of logic…

A traditional approach is to use FOL (first order logic) to ”formalize” meaning (see Formal Semantics Theories).

… before start studying how semantics is used in Language Technology-based applications, let’s summarise how semantics has been brought into computer science and artificial intelligence.

REPRESENTATION AND SEMANTICS

Properties of Meaning in Logic

- Verifiability
  - Can a statement be verified against a knowledge base (KB)
  - Example: does my cat Martin have whiskers?
- Unambiguousness
  - Give me the book
  - Which book?
- Canonical form (standard/abstract form of an expression)
- Expressiveness
  - Can the formalism express temporal relations, beliefs, …?
  - Is it domain-independent?
- Inference

Representing Meaning

- A traditional approach to meaning is to
  - use propositional logic
  - predicate/first order logic (FOL)
  - use theorem proving (inference) to determine whether a statement entails another

Propositional Logic : propositions are represented by letters

- In propositional logic, we use letters to symbolize entire propositions.
- Propositions are statements of the form "x is y" where x is a subject and y is a predicate.
- For example, "Socrates is a Man" is a proposition and might be represented in propositional logic as "S".

True and False

- Propositional Logic studies propositions, whether they are true or false.
- A sentence might contain several propositions that are linked together by logical connectives.

- Logical connectives are found in natural languages. In English for example, some examples are "and" (conjunction), "or" (disjunction), "not" (negation) and "if" (but only when used to denote material conditional).
- Ex (wikipedia): simple inference within the scope of propositional logic:
- Premise 1: If it's raining then it's cloudy. Premise 1: $P \to Q$
- Premise 2: It's raining. Premise 2: P
- Conclusion: It's cloudy. Conclusion: Q

Both premises and the conclusion are propositions. The premises are taken for granted and then with the application of modus ponens (an inference rule) the conclusion follows.

Syntax of Propositional Logic

- The simplest type of logic

– If S is a sentence, ¬S is a sentence (negation)
– If S1 and S2 are sentences, S1 ∧ S2 is a sentence (conjunction)
– If S1 and S2 are sentences, S1 ∨ S2 is a sentence (disjunction)
– If S1 and S2 are sentences, S1 ⇒ S2 is a sentence (implication)
– If S1 and S2 are sentences, S1 ⇔ S2 is a sentence(biconditional)

Propositional Logic and Natural Language

Operator Precedence

$\neg$ (highest) = negation

$\wedge$ = conjunction

$\vee$ = disjunction

$\Rightarrow$ = implication

$\Leftrightarrow$ (lowest) = biconditional*

*A biconditional statement is defined to be true whenever both parts have the same truth value.

Logic and Language

A = Today is a holiday.

B = We are going to the park.

$A \Rightarrow B$

$A \wedge \neg B$

$\neg A \Rightarrow B$

$\neg B \Rightarrow A$

$B \Rightarrow A$

Truth table

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Propositional Logic and Semantics

Translating propostions to Eng

A = Today is a holiday.
B = We are going to the park.

$A \Rightarrow B$
If today is a holiday, we are going to the park.
$A \wedge \neg B$
Today is a holiday and we are not going to the park.
$\neg A \Rightarrow \neg B$
If today is not a holiday, then we are not going to the park.
$\neg B \Rightarrow \neg A$
If we are not going to the park, then today is not a holiday.
$B \Rightarrow A$
If we are going to the park, then today is a holiday.

Semantics of Propositional Logic

$\neg S$ is true iff S is false
$S_1 \wedge S_2$ is true iff S_1 is true and S_2 is true
$S_1 \vee S_2$ is true iff S_1 is true or S_2 is true
$S_1 \Rightarrow S_2$ is true iff S_1 is false or S_2 is true
i.e., is false iff S_1 is true and S_2 is false
$S_1 \Leftrightarrow S_2$ is true iff $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true
Recursively, one can compute the truth value of longer formulas

Propositional logic vs FOL/Predicate Logic

- Propositional logic (also called sentential logic) is logic that includes sentence letters (A,B,C …) and logical connectives, but not quantifiers. The semantics of propositional logic uses truth assignments to the letters to determine whether a compound propositional sentence is true.

- Predicate logic is usually used as a synonym for first-order logic. Syntactically, first-order logic has the same connectives as propositional logic, but it also has variables for individual objects, quantifiers, symbols for functions, and symbols for relations.

- Its semantics includes a domain of discourse for the variables and quantifiers to range over, along with interpretations of the relation and function symbols.

Predicate Logic: more expressive than Propositional Logic

- In predicate logic, we symbolize subject and predicate separately.

- The important difference is that you can use predicate logic to say something about a set of objects.
- By introducing the universal quantifier (" ∀ "), the existential quantifier (" ∃ ") and variables ("x", "y" or "z"), we can use predicate logic to represent thing like:
  - "Everything is green" as " ∀ Gx" "
  - Something is blue" as " ∃ Bx"

Used to represent

- Objects – Martin the cat
- Relations – Martin and Moses are brothers
- Functions – Martin's age

FOL elements (short version)

- Constant
- Variables
- Predicates
- Boolean connectives
- Quantifiers
- Brackets and comma to group the symbols together

- Ex: A woman crosses Sunset Boulevard

$$\exists x(\text{WOMAN}(x) \wedge \exists y(y = \text{SUNSET-BOULEVARD} \wedge \text{CROSS}(x,y)))$$

FOL elements (long version)

Formula → AtomicFormula or Formula Connective Formula or Quantifier Variable Formula or ¬ Formula

(Formula):

AtomicFormula → Predicate (Term…)

Term → Function (Term…) | Constant | Variable

Connective → $\wedge$ | $\vee$ | $\Rightarrow$

Quantifier → $\forall$ | $\exists$

Constant → M | Martin

Variable → x | y | …

Predicate → Likes | Eats | …

Function → AgeOf | ColorOf | …

FOL's most important feature: quantifiers → generalization

¬ *studied_for(John, test)*	John did <u>not</u> study for the test.
studied_for(John, test) ⟹ *aced(John, test)*	<u>If</u> John studied for the test, <u>then</u> he aced it.
$[\exists_x : student(x) \wedge aced(x, test)]$	There exists at least one student who aced the test.
$[\forall_x : student(x) \Longrightarrow studied\_for(x, test)]$	Every student studied for the test.

To ace a test = to get a very high score on a test

Properties of Predicate Logic

- Pros
  - Compositional
  - Declarative
- Cons
  - Limited expressive power
  - Represents facts

Lambda calculus → Variable substitution

Useful for expressing computation by way of variable binding and substitution.
Variables are just names that are bound as arguments.

Lambda calculus is important in programming language theory.

Example
– inc(x) = $\lambda x\ x+1$
– then inc(4) = $(\lambda x\ x+1)(4) = 5$

Example
– add(x,y) = $\lambda x,\lambda y(x+y)$
– then add(3,4) = $(\lambda x,\lambda y(x+y))(3)(4)= (\lambda y\ 3+y)(4) =3+4 = 12$

Stages of Semantic Parsing

- Input
  - Sentence
- Syntactic Analysis
  - Syntactic structure
- Semantic Analysis
  - Semantic representation

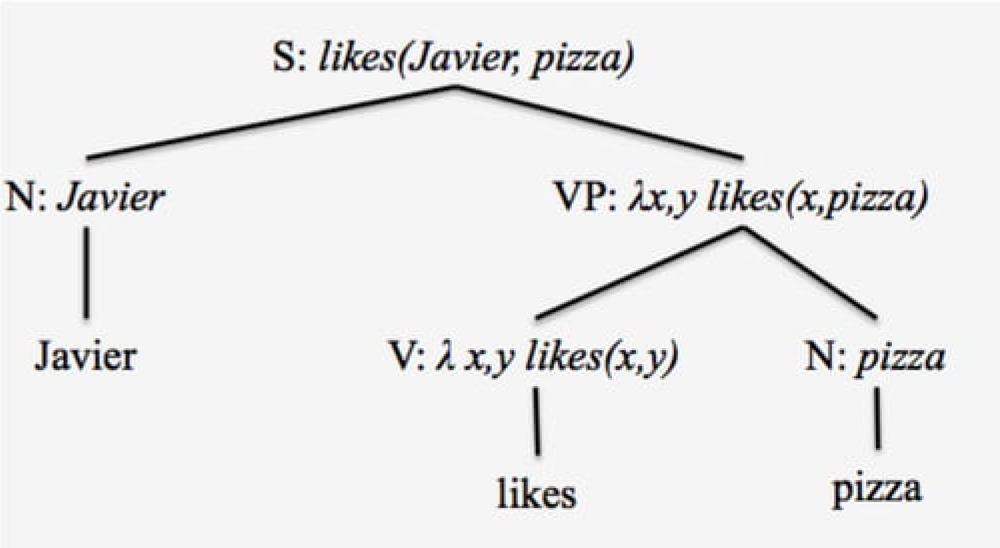
Compositional Semantics

- Add semantic attachments to syntactic rules
- Compositional semantics
  - Parse the sentence syntactically
  - Associate some semantics to each word
  - Combine the semantics of words and non terminals recursively
  - Until the root of the sentence

Example

- Input
  - Javier likes pizza
- Output
  - like(Javier, pizza)

Associate a semantic expression with each node



Semantic Parsing

- Converting natural language to a logical form
  - e.g., executable code for a specific application
  - Example:
    - Airline reservations
    - Geographical query systems

What about ”kick the bucket”?

- What happens when the meaning of single words does not lead to the meaning of the whole?

COMPUTATION AND COMPOSITIONAL SEMANTICS

Unification Grammar

- Unification grammar is a formalism that derives semantic representations by accumulating **constraints** in tandem with a syntactic derivation.
- The constraints take the form of equations between terms which may contain variables.
- The formalism gets its name from the process, **unification**, which determines values for variables to solve the equations.

Example: ”everybody walks”

The meaning representation of $\forall x.walks(x)$ is assembled as the solution to a **set of constraints** and not through the function–argument maninupulation of the lambda calculus.

More flexibility

- Lambda-calculus techniques expect each element in a meaning representation to be specified exactly once.
- Unification: we model sentence meaning as the solution to a set of equations → we might find elements of meaning that are simultaneously determined by multiple equations, and elements of meaning that are left unconstrained and must be resolved by context.
- Constraint techniques offer new ways to analyze idioms and multi-word expressions. Each constituent can not only specify its own meaning but can impose a constraint on the meaning of the whole.

Limitations

- Compositional semantics (any formalisms) gives us representations that capture aspects of the logical structure of meaning.

- But: what specific entities and concepts does a sentence describe, and what specific claim does it make about them?

CONCLUSIONS

Basically…

- We need to tie together:
  - Grammar (grammar and compositional meaning is not enough)
  - Speaker's kowledge about the wolrd
  - **Patterns of use → unique insights given by large corpora of linguistic evidence**

In this course…

- We review research about different aspects of word meaning that are needed for LT-based applications.

How is *meaning* handled in Semantic-Based LT-Applications

- Semantic Role Labelling/Predicate-Argument Structure (???)
- Sentiment Analysis (???)
- Word sense disambiguation (???)
- Information extraction (???)
- Question Answering (???)
- Ontologies (???)

The End