Computational Semantics

CS224N 2007
Christopher Manning

(Borrows some slides from Mary Dalrymple, Jason Eisner, and Jim Martin)
Why study computational semantics?

• Because everyone has been wanting me to talk about this all course!?

• Obvious high-level applications
  • Summarization
  • Translation
  • Question answering
  • Information access
  • Talking to your pet robot
  • Speech user interfaces

• The next generation of intelligent applications need deeper semantics than we have seen so far
  • Often you must understand well to be able to act
We can do more than one might have thought without deep linguistic analysis
   • This is *the* lesson of the last decade

But we can’t do everything we would like:
   • Not all tasks can ignore higher structure
   • Unsuitable if new text must be generated
   • Unsuitable if machine must act rather than relying on user to interpret material written by the author of the document

You get what you pay for:
   • Cheap, fast, low-level techniques are appropriate in domains where speed and volume are more important than accuracy
   • More computationally expensive, higher-level techniques are appropriate when high-quality results are required
MSN Search: Which is the largest African country?

Answer: Sudan (country), republic in northeastern Africa, the largest country of the African continent. It is bounded on the north by Egypt, on the east by the...
Live Search: Which is the largest African country?

African facts and figures
Sudan, Republic of, republic in north-eastern Africa, the largest country of the African continent. Sudan has a total area of 2,505,800 sq km (967,490 sq mi).
www.africaguide.com/facts.htm - Cached page

South Africa - Wikipedia, the free encyclopedia
At 470,979 mi² (1,239,912 km²), South Africa is the world's 25th-largest country (after Mali) ... It is the only African country to have successfully developed nuclear weapons.
en.wikipedia.org/wiki/South_Africa - Cached page

Central African Republic - Wikipedia, the free encyclopedia
At 240,519 mi² (622,984 km²), the Central African Republic is the world's 43rd-largest country (after Somalia). It is comparable in size to Ukraine, and is somewhat smaller than the US state of Texas.
en.wikipedia.org/wiki/Central_African_Republic - Cached page

Southern African Development Community Country Analysis Brief
On January 1, 2004, Angola became the 37th sub-Saharan African country eligible for tariff ... Songo Songo, the country's largest known field, is located in the Indian Ocean southeast of the ...
www.eia.doe.gov/emeu/cabs/sadc.html - Cached page
Live Search: What is the capital of Sudan?

Sudan Capital: Khartoum

Juba, Sudan - Wikipedia, the free encyclopedia
Juba is the regional capital of Southern Sudan and the capital of the Sudanese state of Central Equatoria. Population In 2005 its population was 163,442. Development of Population: History British ...

en.wikipedia.org/wiki/Juba%2C_Sudan - Cached page

Sudan - Wikipedia, the free encyclopedia
... the height of its power in the second and third centuries BC, Merneptah extended over a region from the third cataract in the north to Sawba, near present-day Khartoum (the modern day capital of Sudan)

en.wikipedia.org/wiki/Sudan - Cached page

Show more results from en.wikipedia.org

Map of Sudan and its Capital
Maps [ Map of Sudan | Map of Greater Khartoum ] Additional detailed maps of the various campuses will be added. Map of the Republic of Sudan. This map shows the location of the Capital of Sudan.

www.sudan.net/uk/maps.htm - Cached page
Which countries does the Danube flow through?

Danube River

... and the only major European river to flow from West to East. It takes its source ... the Romanian coast. Along its way, the Danube flows through nine countries (Germany, Austria, Slovakia, Hungary ...

www.public.asu.edu/~goutam/gcu325/danube.htm Cached page

Danube River

... down this river. The Danube, through its rich history, remains ... of Yugoslavia. On the Danube, different countries have built dams and ... level. After all, the Danube does provide a major ...

www2.intop.net/~jholliis/danube.htm Cached page

Danube River

... only major European river to flow from west to east. It rises ... where the Danube Delta is. The Danube is an important international waterway. It flows through ten countries (Austria, Bulgaria, Croatia ...
MSN Search: What are the capitals of the countries bordering the Baltic?
What we say to dogs

Okay, Ginger! I've had it! You stay out of the garbage! Understand, Ginger? Stay out of the garbage, or else!
what they hear

blah blah GINGER blah
blah blah blah blah blah
blah blah GINGER blah
blah blah blah blah blah...
Precise semantics. An early example: Chat-80

- Developed between 1979 and 1982 by Fernando Pereira and David Warren; became Pereira’s dissertation
- Proof-of-concept natural language interface to database system
- Used in projects: e.g. Shoptalk (Cohen et al. 1989), a natural language and graphical interface for decision support in manufacturing
- Even used in an AppliedNLP-2000 conference paper! [Asking about train routes and schedules]
- Available in cs224n src directory
  - Need sicstus prolog: /usr/sweet/bin/sicstus
The CHAT-80 Database

% Facts about countries.
% country(Country,Region,Latitude,Longitude,
%       Area (sqmiles), Population, Capital,Currency)
country(andorra,southern_europe,42,-1,179,
       25000,andorra_la_villa,franc_peseta).
country(angola,southern_africa,-12,-18,481351,
       5810000,luanda,?).
country(argentina,south_america,-35,66, 1072067,
       23920000,buenos_aires,peso).

capital(C,Cap) :- country(C,_,_,_,_,_,Cap,_,_).
Question: What is the capital of Australia?

Parse: 0.0sec.

```
whq
  $VAR
    1
    s
      np
        3+sin
        wh(B)
        []
        verb(be, active, pres+fin, [], pos)
      arg
        dir
        np
          3+sin
```

Semantics: 0.0sec.

```
answer([B]) :-
  capital(australia,B)
```

canberra.
What is meaning of $3+5\times6$?
- First parse it into $3+(5\times6)$
- Now give a meaning to each node in the tree (bottom-up)
More complex meanings

- How about $3 + 5 \times x$?
- Don’t know $x$ at compile time
- “Meaning” at a node is a piece of code, not a number

- Form is “rule-to-rule” translation
  - We provide a way to form the semantics of each parent in terms of the semantics of the children
What Counts as Understanding?

• A somewhat difficult philosophical question
• We understand if we can respond appropriately
  • “throw axe at dwarf”
• We understand statement if we can determine its truth
• We understand statement if we can use it to answer questions
  [similar to above – requires reasoning]
  • Easy: John ate pizza. What was eaten by John?
• Understanding is the ability to translate
  • English to Chinese? requires deep understanding?? String transduction!
  • English to logic? deepest - the definition we’ll use!
  • all humans are mortal = ∀x [human(x) ⇒ mortal(x)]
• We assume we have logic-manipulating rules to tell us how to act, draw conclusions, answer questions ...
Lecture Plan

• Today:
  • Look at some sentences and phrases
  • What would be reasonable logical representations for them?
  • Get some idea of compositional semantics
  • An alternative semantic approach
    • Semantic grammars
• Next wednesday:
  • How can we build those representations?
• Another course (somewhere in AI, hopefully):
  • How can we reason with those representations?
• Last week of lectures:
  • Lexical semantics
  • Question answering/semantic search/textual entailment
Three major kinds of objects

1. Booleans (Bool)
   - Roughly, the semantic values of sentences

2. Individuals/Entities (Ind)
   - Values of NPs, i.e., objects
   - Maybe also other types of entities, like times

3. Functions of various types
   - A function returning a boolean is called a “predicate”
     - e.g., frog(x), green(x)
   - A predicate defines a set of individuals that satisfy it
   - A one argument predicate is called a “property”
   - More complex functions return other functions!
   - Some functions take other functions as arguments!
     - (Higher order functions.)
Logic: Lambda Terms

- Lambda terms:
  - A way of writing “anonymous functions”
    - No function header or function name
    - But defines the key thing: behavior of the function
    - Just as we can talk about 3 without naming it “x”
  - Let square = λp. p*p
  - Equivalent to int square(p) { return p*p; }
  - But we can talk about λp p*p without naming it
  - Format of a lambda term: λ variable . expression
Logic: Lambda Terms

- Lambda terms:
  - Let `square = \lambda p \ p*p`
  - Then `square(3) = (\lambda p \ p*p)(3) = 3*3`
  - Note: `square(x)` isn’t a function! It’s just the value `x*x`.
  - But `\lambda x \ square(x) = \lambda x \ x*x = \lambda p \ p*p = square`
    (proving that these functions are equal – and indeed they are, as they act the same on all arguments: what is `(\lambda x \ square(x))(y)`?)

- Let `even = \lambda p \ (p \ mod \ 2 == 0)` a predicate: returns true/false
- `even(x)` is true if `x` is even
- How about `even(square(x))`?
- `\lambda x \ even(square(x))` is true of numbers with even squares
  - Just apply rules to get `\lambda x \ (even(x*x)) = \lambda x \ (x*x \ mod \ 2 == 0)`
  - This happens to denote the same predicate as `even` does
Logic: Multiple Arguments

• All lambda terms have one argument
• But we can fake multiple arguments ...

• Suppose we want to write \( \text{times}(5,6) \)
• Remember: \( \text{square} \) can be written as \( \lambda x.\text{square}(x) \)
• Similarly, \( \text{times} \) is equivalent to \( \lambda x.[\lambda y.\text{times}(x,y)] \)

• Claim that \( \text{times}(5)(6) \) means same as \( \text{times}(5,6) \)
  • \( \text{times}(5) = (\lambda x.\lambda y.\text{times}(x,y))(5) = \lambda y.\text{times}(5,y) \)
    • If this function weren’t anonymous, what would we call it?
  • \( \text{times}(5)(6) = (\lambda y \text{times}(5,y))(6) = \text{times}(5,6) \)
• Referred to as “currying”
Logic: Interesting Constants

- We have “constants” that name some of the entities and functions (e.g., times):
  - GeorgeWBush - an entity
  - red – a predicate on entities
    - holds of just the red entities: red(x) is true if x is red!
  - loves – a predicate on 2 entities
    - loves(GeorgeWBush, LauraBush)
    - Question: What does loves(LauraBush) denote?

- Constants used to define meanings of words
- Meanings of phrases will be built from the constants
Logic: Interesting Constants

• **Generalized Quantifiers**

• **most** – a predicate on 2 predicates on entities
  - **most**(pig, big) = “most pigs are big”
  - Equivalently, **most**(\(\lambda x\) pig(x), \(\lambda x\) big(x))
  - returns true if most of the things satisfying the first predicate also satisfy the second predicate

• similarly for other quantifiers
  - **all**(pig, big) (equivalent to \(\forall x\) pig(x) \(\Rightarrow\) big(x))
  - **exists**(pig, big) (equivalent to \(\exists x\) pig(x) AND big(x))
  - can even build complex quantifiers from English phrases:
    - “between 12 and 75”; “a majority of”; “all but the smallest 2”
Quantifier Order

• Groucho Marx celebrates quantifier order ambiguity:
  • In this country a woman gives birth every 15 min. Our job is to find that woman and stop her.
  • $\exists$woman $(\forall 15\text{min gives-birth-during}(\text{woman}, 15\text{min}))$
  • $\forall 15\text{min } (\exists \text{woman gives-birth-during}(15\text{min}, \text{woman}))$
  • Surprisingly, both are possible in natural language!
  • Which is the joke meaning?
    • (where it’s always the same woman)
We’ve discussed what semantic representations should look like.

But how do we get them from sentences???

First - parse to get a syntax tree.

Second - look up the semantics for each word.

Third - build the semantics for each constituent
  - Work from the bottom up
  - The syntax tree is a “recipe” for how to do it

Principle of Compositionality
  - The meaning of a whole is derived from the meanings of the parts, via composition rules
A simple grammar of English
(in Definite Clause Grammar, DCG, form – as in Prolog)

sentence --> noun_phrase, verb_phrase.
noun_phrase --> proper_noun.
noun_phrase --> determiner, noun.
verb_phrase --> verb, noun_phrase.

Proper_noun --> [John]  verb --> [ate]
Proper_noun --> [Mary]   verb --> [kissed]
determiner --> [the]     noun --> [cake]
determiner --> [a]       noun --> [lion]
Extending the grammar to check number agreement between subjects and verbs

S --> NP(Num), VP(Num).
NP(Num) --> Proper_noun(Num).
NP(Num) --> det(Num), noun(Num).
VP(Num) --> verb(Num), noun_phrase(_).

Proper_noun(s) --> [Mary].
noun(s) --> [lion].
det(s) --> [the].
noun(p) --> [lions].
det(p) --> [the].
verb(s) --> [eats].
verb(p) --> [eat].
A simple DCG grammar with semantics

\[
\begin{align*}
\text{sentence}(\text{SMeaning}) & \rightarrow \text{noun_phrase}(\text{NPMeaning}), \\
& \text{verb_phrase}(\text{VPMeaning}), \{\text{combine}\ (\text{NPMeaning}, \ 	ext{VPMeaning}, \text{SMeaning})\}\}.
\text{verb_phrase}(\text{VPMeaning}) & \rightarrow \text{verb}(\text{Vmeaning}), \\
& \text{noun_phrase}(\text{NPMeaning}), \{\text{combine}\ (\text{NPMeaning}, \ 	ext{VMeaning}, \text{VPMeaning})\}\}.
\text{noun_phrase}\ (\text{NPMeaning}) & \rightarrow \text{name}(\text{NPMeaning}).
\text{name}(\text{john}) & \rightarrow [\text{john}]. \quad \text{verb}(\lambda x. \text{jumps}(x)) \rightarrow [\text{jumps}]
\text{name}(\text{mary}) & \rightarrow [\text{mary}]. \quad \text{verb}(\lambda y. \lambda x. \text{loves}(x,y)) \rightarrow [\text{loves}]
\text{Combine}(X, Y, Z) & \rightarrow \text{apply}(Y, X, Z)
\end{align*}
\]
Parse tree with associated semantics

Sentence
loves(john,mary)

Noun Phrase
john
  Name
  “John”
  john

Verb Phrase
λx. loves(x,mary)
  Verb
  λy. λx. loves(x,y)
  “loves”
  λy. λx. loves(x,y)

Noun Phrase
  Name
  mary
  “Mary”
  mary
Augmented CFG Rules

• We can also accomplish this just by attaching semantic formation rules to our syntactic CFG rules

\[ A \rightarrow \alpha_1...\alpha_n \quad \{ f(\alpha_1.sem,...,\alpha_n.sem) \} \]

• This should be read as the semantics we attach to A can be computed from some function applied to the semantics of A’s parts.

• The functions/operations permitted in the semantic rules are restricted, falling into two classes
  • Pass the semantics of a daughter up unchanged to the mother
  • Apply (as a function) the semantics of one of the daughters of a node to the semantics of the other daughters
Six sculptures – C, D, E, F, G, H – are to be exhibited in rooms 1, 2, and 3 of an art gallery.

- Sculptures C and E may not be exhibited in the same room.
- Sculptures D and G must be exhibited in the same room.
- If sculptures E and F are exhibited in the same room, no other sculpture may be exhibited in that room.
- At least one sculpture must be exhibited in each room, and no more than three sculptures may be exhibited in any room.

If sculpture D is exhibited in room 3 and sculptures E and F are exhibited in room 1, which of the following may be true?

1. Sculpture C is exhibited in room 1.
2. Sculpture H is exhibited in room 1.
3. Sculpture G is exhibited in room 2.
4. Sculptures C and H are exhibited in the same room.
5. Sculptures G and F are exhibited in the same room.
At least one sculpture must be exhibited in each room.
The same sculpture in each room?
No more than three sculptures may be exhibited in any room.

Reading 1: For every room, there are no more than three sculptures exhibited in it.

Reading 2: Only three or less sculptures are exhibited (the rest are not shown).

Reading 3: Only a certain set of three or less sculptures may be exhibited in any room (for the other sculptures there are restrictions in allowable rooms).

- Some readings will be ruled out by being uninformative or by contradicting other statements
- Otherwise we must be content with distributions over scope-resolved semantic forms
Semantic Grammars

• A problem with traditional linguistic grammars is that they don’t necessarily reflect the semantics in a straightforward way
• You can deal with this by...
  • Fighting with the grammar
    • Complex lambdas and complex terms, etc.
  • Rewriting the grammar to reflect the semantics
    • And in the process give up on some syntactic niceties
    • known as “Semantic grammars”
      • Simple idea, dumb name
The term semantic grammar refers to the motivation for the grammar rules:
- The technology (plain CFG rules with a set of terminals) is the same as we’ve been using.
- The good thing about them is that you get exactly the semantic rules you need.
- The bad thing is that you need to develop a new grammar for each new domain.

Typically used in conversational agents in constrained domains:
- Limited vocabulary
- Limited grammatical complexity
- Syntactic parsing can often produce all that’s needed for semantic interpretation even in the face of “ungrammatical” input – write fragment rules.
Lifer Semantic Grammars

- Example domain—access to DB of US Navy ships
  \[ S \rightarrow \text{<present>} \text{ the } \text{<attribute>} \text{ of } \text{<ship>} \]
  \[ \text{<present>} \rightarrow \text{what is } | \text{[can you] tell me} \]
  \[ \text{<attribute>} \rightarrow \text{length } | \text{beam } | \text{class} \]
  \[ \text{<ship>} \rightarrow \text{the } \text{<shipname>} \]
  \[ \text{<shipname>} \rightarrow \text{kennedy } | \text{enterprise} \]
  \[ \text{<ship>} \rightarrow \text{<classname>} \text{ class ships} \]
  \[ \text{<classname>} \rightarrow \text{kitty hawk } | \text{lafayette} \]

- Example inputs recognized by above grammar:
  * can you tell me the class of the Enterprise
  * what is the length of Kitty Hawk class ships

  - Many categories are not "true" syntactic categories
  - Words are recognized by their context rather than category (e.g. class)
  - Recognition is strongly directed
  - Strong direction useful for error detection and correction
Semantic Grammars Summary

- Advantages:
  - Efficient recognition of limited domain input
  - Absence of overall grammar allows pattern-matching possibilities for idioms, etc.
  - No separate interpretation phase
  - Strength of top-down constraints allows powerful ellipsis mechanisms

  *What is the length of the Kennedy? The Kittyhawk?*

- Disadvantages:
  - Different grammar required for each new domain
  - Lack of overall syntax can lead to "spotty" grammar coverage
    - E.g. fronting possessive in "<attribute> of <ship>“ to <ship> ’s <attribute> doesn't imply fronting in "<rank> of <officer>"
  - Difficult to develop grammars past a certain size
  - Suffers from fragility