Computational Semantics

CS224N
Christopher Manning
(Borrows some slides from Mary Dalrymple, Jason Eisner, and Jim Martin)

Why study computational semantics?
• Because everyone has been wanting to hear about this all course?!
• Obvious high-level applications
  • Question answering
  • Information access
  • Talking to your pet robot
  • Speech user interfaces
  • Summarization
  • Translation
• 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

Shallow vs. deep semantics

We can do more than people thought without deep linguistic analysis

This is the lesson of the last decade

Shallow vs. deep semantics

But we can’t do everything we would like

Shallow vs. deep semantics

We can’t do everything we would like:
• Not all tasks can ignore language structure/meaning
• 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 when speed and volume are more important than accuracy
• Computationally expensive, higher-level techniques are appropriate when high-quality results are required

SHRDLU
http://hci.stanford.edu/winograd/shrdlu/
Google: What is the capital of Algeria?

Google: Which countries does the Danube flow through?

Google: What are the capitals of the countries bordering the Baltic Sea?

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

Apple Knowledge Navigator video (1987)
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
- Used in an AppliedNLP-2000 conference paper!
[Asking about train routes and schedules – still live]
- UNFORTUNATELY THERE DOESN’T SEEM TO BE ANY PROLOG INTERPRETER ON ITS MACHINES NOW 😞

Things you could ask...

- What is the total area of countries south of the Equator and not in Australasia?
- What is the average area of the countries in each continent?
- Is there more than one country in each continent?
- What are the countries from which a river flows into the Black_Sea?
- Which country bordering the Mediterranean borders a country that is bordered by a country whose population exceeds the population of India?

Chat-80 trace (small)

Semantics: 0.0sec.

answer(B) :-
capital(australia,B)
canberra.

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, 2392000,buenos_aires,peso).
capital(C,Cap) :- country(C,_,_,_,_,_,Cap,_).

The CHAT-80 grammar

Fast-forward to 2015

- Siri
- Google Now
- Cortana
- Amazon Echo/Alexa
- Nuance Nina
- Facebook M

Speech is fast becoming the preferred mobile UI

- Satya Nadella (MSFT), NXT Economy conference, Nov ’15: “speaking and texting will be ‘the new U.I.’ for software.” Natural Language Understanding becomes necessary
The largest country in South America is Brazil. It is mentioned in the results below.

What countries does the Danube River flow through? - Ask.com
www.ask.com/...

The Danube River flows through Austria, Germany, Hungary, and Serbia. Brazil is also mentioned in the results.

What countries does the Danube River border? - Ask.com
www.ask.com/...

The Danube River borders Austria, Germany, and Hungary. Brazil is also mentioned in the results.

I found some tables for four around 7 pm. Tap the one you'd like to reserve:

2 Restaurants

- **Reposado**
  - 0.7 mi
  - 306 Hamilton Ave
  - Mexican, Vegetarian, $$$
  - Open today: 8:00 PM - 7:30 PM
  - 4 stars and 553 reviews

- **Joya**
  - 0.8 mi
  - 309 University Ave
  - Latin American, Mexican, $$$
Google’s latest – Nov 16, 2015

The Google app can now handle superlatives

“Who are the tallest Mavericks players?”
Second, we now understand questions with dates:

“What songs did Taylor Swift record in 2014?”
Finally, we’re starting to understand some complex combinations:

“What are some of Seth Gabel’s father-in-law’s movies?”
“What was the U.S. population when Bernie Sanders was born?”

More complex meanings

• How about 3 + 5 * 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

(3+5*6) = 3 + (5*6)

Logic: Some Preliminaries

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”
   • A predicate defines a set of individuals that satisfy it
   • A function returning a set 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:
  • Let square = λp p*p
  • Then square(3) = (λp p*p) 3 = 3*3
  • Note: square(x) isn’t a function! It’s just the value x*x.
  • But λx square(x) = λx x*x = λx p p = square
  • proving these functions are equal - and indeed they are, as they act the same on all arguments: what is the square(x*y)?
  • Let even = λp (p mod 2 == 0)
  • even(x) is true if x is even
  • How about even(square(x))?
  • λx even(square(x)) is true of numbers with even squares
  • just apply rules to get λp (p mod 2 == 0) = λp (p mod 2 == 0)
  • This happens to denote the same predicate as even does

Programming Language Interpreter

• What is meaning of 3 + 5*6?
• First parse it into 3 + (5*6)
• Now give a meaning to each node in the tree (bottom-up)

(Formal/Compositional)
Computational Semantics

• Sentences: “John smokes.”
  “Everyone who smokes snores.”
• Syntactic Analyses:
  S
  NP
  VP
  John
  smokes
• Semantics Construction:
  smoke(j)
• Logic as meaning representation language
• Inference:
  ∀x.smoke(x) → snore(x), smoke(j) => snore(j)
Logic: Multiple Arguments

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

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

• Claim that `times(5)(6)` means same as `times(5,6)`
  
  \[ \text{times}(5) = (\lambda y.\text{times}(5,y)) \]

• 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`):
  
  - `GeorgeW Bush` – 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(GeorgeW Bush, LauraBush)`
    
    - Question: What does `loves(LauraBush)` denote?
  
  • Constants used to define meanings of words
  
  • Meanings of phrases will be built from the constants

Compositional Semantics

• 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 DCG grammar with semantics

\[ \text{sentence} \rightarrow \text{noun_phrase}, \text{verb_phrase}. \]
\[ \text{noun_phrase} \rightarrow \text{proper_noun}. \]
\[ \text{proper_noun} \rightarrow [\text{John}]. \]
\[ \text{verb} \rightarrow [\text{ate}]. \]
\[ \text{noun} \rightarrow [\text{cake}]. \]
\[ \text{determiner} \rightarrow [\text{the}]. \]
\[ \text{verb} \rightarrow [\text{eat}]. \]

Extending the grammar to check number agreement between subjects and verbs

\[ S \rightarrow \text{NP}(\text{Num}), \text{VP}(\text{Num}). \]
\[ \text{NP}(\text{Num}) \rightarrow \text{proper_noun}(\text{Num}). \]
\[ \text{proper_noun}(\text{Num}) \rightarrow [\text{Mary}]. \]
\[ \text{verb} \rightarrow [\text{jumps}]. \]
\[ \text{noun} \rightarrow [\text{lion}]. \]
\[ \text{determiner} \rightarrow [\text{the}]. \]
\[ \text{verb} \rightarrow [\text{eats}]. \]
\[ \text{determiner} \rightarrow [\text{a}]. \]

Combine(X, Y, Z) \rightarrow apply(Y, X, Z)
Parse tree with associated semantics

In detail: Beta-Reduction

Formal Compositional Semantics ...

Augmented CFG Rules

Logic: Interesting Constants

How do things get more complex? (The former) GRE analytic section
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.
  - ∃ woman ( ∀ 15min gives-birth-during(woman, 15min) )
  - ∀ 15min ( ∃ woman gives-birth-during(15min, woman) )
  - Surprisingly, both are possible in natural language!
  - Which is the joke meaning?
    - (where it’s always the same woman)

Scope Needs to be Resolved!

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

An alternative:
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 lambda 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

Lifer Semantic Grammars

- Example domain—access to DB of US Navy ships
  S → <present> the <attribute> of <ship>
  <present> → what is | [can you] tell me
  <attribute> → length | beam | class
  <ship> → the <shipname> | enterprise
  <shipname> → kenne d y | enterprise
  <classname> → k i tty hawk | 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 Grammar

- 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

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 ships" to "ship"’s attribute doesn’t imply fronting in "rank of officers"
  - Difficult to develop grammars past a certain size
  - Suffers from fragility
Facebook Graph Search

- Uses a weighted context-free grammar (WCFG) to represent the Graph Search query language:

- \([\text{start}] \rightarrow [\text{users}]\) $1$
- \([\text{users}] \rightarrow \text{my friend(s)}\) friends(me)
- \([\text{users}] \rightarrow \text{friends of [users]}\) friends($1$)
- \([\text{users}] \rightarrow \text{[user]}\) $1$
- \([\text{users}] \rightarrow \text{[user] - filter}\) intersect($1$, $2$)
- \([\text{start}] \rightarrow [\text{photos}]\) $1$
- \([\text{photos}] \rightarrow \text{photos of [users]}\) photos($1$)

- A terminal symbol can be an entity, e.g., \{user\}, \{city\}, \{employer\}, \{group\}; it can also be a word/phrase, e.g., friends, live in, work at, members, etc. A parse tree starts from [start] and expands via rules to terminal symbols.

The parse tree, semantic and entity ID used in the above example are for illustration only; they do not represent real information used in Graph Search Beta.