CSCI 5832
Natural Language Processing

Jim Martin

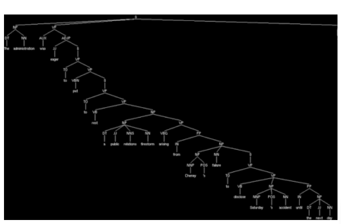
Lecture 19

Today 4/1

- More semantics
  - Dealing with quantifiers
  - Dealing with ambiguity

Example

Even if this is the right tree, what does that tell us about the meaning?



Meaning Representations

- We're going to take the same basic approach to meaning that we took to syntax and morphology
- We're going to create representations of linguistic inputs that capture the meanings of those inputs.
- But unlike parse trees and the like these representations aren't primarily descriptions of the structure of the inputs…

4/10/08 4

Meaning Representations

- In most cases, they're simultaneously descriptions of the meanings of utterances and of some potential state of affairs in some world.

4/10/08 5

Meaning Representations

- What could this mean…
  - representations of linguistic inputs that capture the meanings of those inputs
- For us it means
  - Representations that permit or facilitate semantic processing

4/10/08 6

Representational Schemes

- We're going to make use of First Order Logic (FOL) as our representational framework
  - Not because we think it's perfect
  - Many of the alternatives turn out to be either too limiting or
  - They turn out to be notational variants

4/10/08 7

FOL

- Allows for…
  - The analysis of truth conditions
    - Allows us to answer yes/no questions
  - Supports the use of variables
    - Allows us to answer questions through the use of variable binding
  - Supports inference
    - Allows us to answer questions that go beyond what we know explicitly

4/10/08 8

Example

- *Mary gave a list to John.*
- Giving(Mary, John, List)
- More precisely
  - Gave conveys a three-argument predicate
  - The first arg is the subject
  - The second is the recipient, which is conveyed by the NP in the PP
  - The third argument is the thing given, conveyed by the direct object

4/10/08 9

Better

- Turns out this representation isn't quite as useful as it could be.
  - Giving(Mary, John, List)
- Better would be

[redacted]

Predicates

- The notion of a predicate just got more complicated…
- In this example, think of the verb/VP providing a template like the following

[redacted]

- The semantics of the NPs and the PPs in the sentence plug into the slots provided in the template

Semantic Analysis

- Semantic analysis is the process of taking in some linguistic input and assigning a meaning representation to it.
  - There a lot of different ways to do this that make more or less (or no) use of syntax
  - We're going to start with the idea that syntax does matter
    - The compositional rule-to-rule approach

Compositional Analysis

- Principle of Compositionality
  - The meaning of a whole is derived from the meanings of the parts
- What parts?
  - The constituents of the syntactic parse of the input
- What could it mean for a part to have a meaning?

4/10/08 13

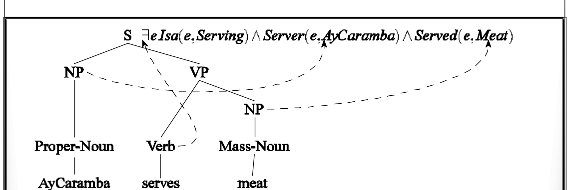
Example

- AyCaramba serves meat

[illegible]

4/10/08 14

Compositional Analysis



4/10/08 15

Augmented Rules

- We'll accomplish this by attaching semantic formation rules to our syntactic CFG rules
- Abstractly

- 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.

Example

- Easy parts…
  - NP -> PropNoun
  - NP -> MassNoun
  - PropNoun -> AyCaramba
  - MassMoun -> meat
- Attachments
  - {PropNoun.sem}
  - {MassNoun.sem}
  - {AyCaramba}
  - {MEAT}

Example

- S -> NP VP
- VP -> Verb NP
- Verb -> serves

- {VP.sem(NP.sem)}
- {Verb.sem(NP.sem)
- ???

Lambda Forms

- A simple addition to FOL
  - Take a FOPC sentence with variables in it that are to be bound.
  - Allow those variables to be bound by treating the lambda form as a function with formal arguments

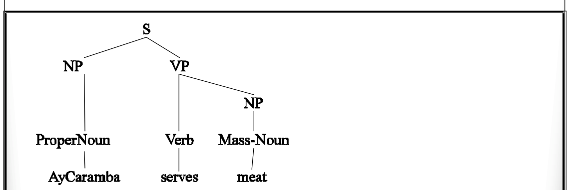
[redacted]

[redacted]

[redacted]

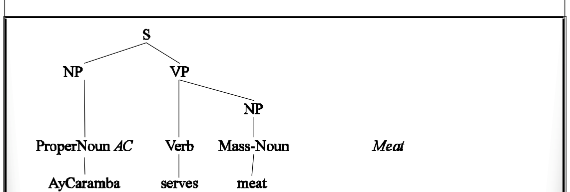
4/10/08 19

Example



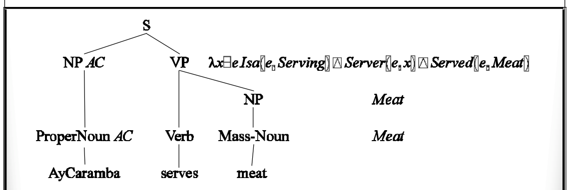
4/10/08 20

Example

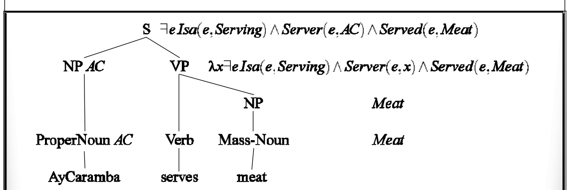


4/10/08 21

Example



Example



Integration

- Two basic approaches
  - Integrate semantic analysis into the parser (assign meaning representations as constituents are completed)
  - Pipeline… assign meaning representations to complete trees only after they're completed

Example

- From BERP
  - I want to eat someplace near campus
- Two parse trees, two meanings

4/10/08 25

Pros and Cons

- If you integrate semantic analysis into the parser as its running…
  - You can use semantic constraints to cut off parses that make no sense
  - You assign meaning representations to constituents that don't take part in the correct (most probable) parse

4/10/08 26

Break

- New schedule is up.
  - Finish 18 today.
  - Next time WSD (secs 20.1 through 20.5)
  - Next week Chapter 22
- Quiz
  - Average was 43 (out of 55)
  - I'll go over it next time.
- Next quiz
  - 4/17
    - Covers 17, 18, 20, 21, 22

4/10/08 27

Quantifiers

- Unfortunately, things get a bit more complicated when we start looking at more complicated NPs.
  - The previous examples simplified things by only dealing with constants (FOL Terms). That is things that can be plugged into FOL predicates. What about...
    - *A menu*
    - *Every restaurant etc*
    - *Not every waiter*

4/10/08 28

Quantifers

- Every restaurant closed.

$$\forall x\, Restaurant(x) \Rightarrow \exists e\, Closed(e) \wedge ClosedThing(e,x)$$

4/10/08 29

Quantifiers

Roughly "every" in an NP like this is used to stipulate something about every member of the class. The NP is specifying the class. And the VP is specifying the thing stipulate.... So the NP is a template like.

$$\forall x\, Restaurant(x) \Rightarrow Q(x)$$

4/10/08 30

Quantifiers

- But that's not combinable with anything so wrap a lambda around it...

$$\lambda Q.\forall x\, Restaurant(x) \Rightarrow Q(x)$$

4/10/08 31

Rules

$$NP \rightarrow Det\, Nominal \qquad \{Det.Sem(Nominal.Sem)\}$$

$$Det \rightarrow every \qquad \{\lambda P.\lambda Q.\forall x\, P(x) \Rightarrow Q(x)\}$$

$$Nominal \rightarrow Noun \qquad \{Noun.sem\}$$

$$Noun \rightarrow restaurant \qquad \{\lambda x.Restaurant(x)\}$$

4/10/08 32

Example

$$\lambda P.\lambda Q.\forall x\, P(x) \Rightarrow Q(x)(\lambda x.Restaurant(x))$$

$$\lambda Q.\forall x\, \lambda x.Restaurant(x)(x) \Rightarrow Q(x)$$

$$\lambda Q.\forall x\, Restaurant(x) \Rightarrow Q(x)$$

4/10/08 33

Every Restaurant Closed

$$\forall x\, Restaurant(x) \Rightarrow \exists e\, Closed(e) \wedge ClosedThing(e,x)$$

$$\lambda Q.\forall x\, Restaurant(x) \Rightarrow Q(x) \qquad \lambda x.\exists e\, Closed(e) \wedge ClosedThing(e,x)$$

closed

$$\lambda P.\lambda Q.\forall x P(x) \Rightarrow Q(x) \qquad \lambda x.Restaurant(x)$$

Every *restaurant*

4/10/08 34

Problem

- Every restaurant has a menu.

$$\forall x\, Restaurant(x) \Rightarrow \exists y\, (Menu(y) \wedge \exists e\, (Having(e) \wedge Haver(e,x) \wedge Had(e,y)))$$

$$\exists y\, Menu(y) \wedge \forall x\, (Restaurant(x) \Rightarrow \exists e\, (Having(e) \wedge Haver(e,x) \wedge Had(e,y)))$$

4/10/08 35

Next Time

- Underspecification

4/10/08 36