Introduction to Natural Language Processing

Chapter 15

Today's Outline

Review Semantics

Semantic Interpretation

- Syntax-Driven Compositional Analysis
- Idioms
- More Robust Syntax-Driven Methods

Review: Meaning Representations

The approach to meaning that we'll present is based on the systematic creation of meaning representations–representations that bridge that gap from linguistic forms to knowledge of the world.

Semantic Analysis

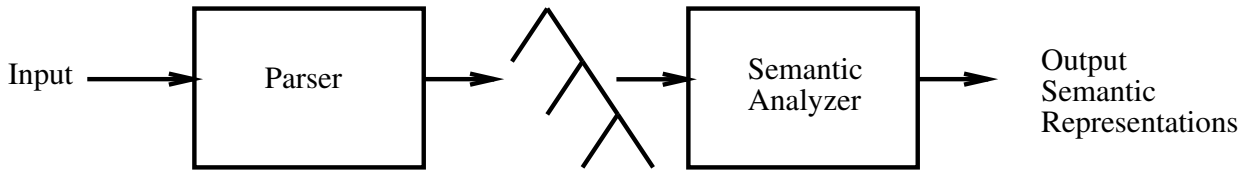
Semantic Analysis is the process of taking in some linguistic form and producing a meaning representation for it.

There are many ways of doing this, ranging from completely ad hoc domain specific methods to more theoretically founded but not quite useful methods.

Most methods rely in some way on a prior or concurrent syntactic analysis (parse).

We'll outline a compositional rule-to-rule approach.

A Simple Pipeline



Compositional Semantics

At the core of most methods is the principle of compositionality which states that the meaning of the whole is based on the meaning of parts.

What are the parts?

Compositional Parts

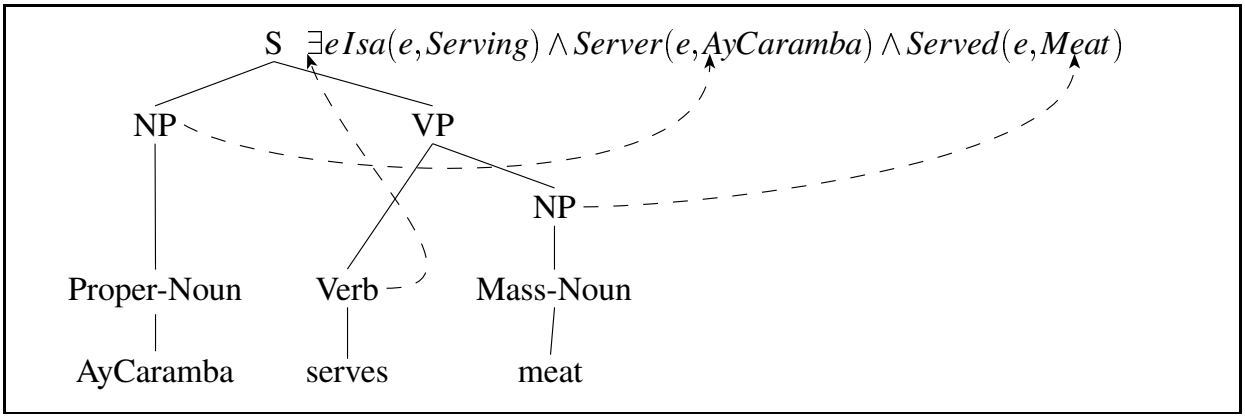
What are the parts?

- words and syntactic constituents

What does it mean for a part (word or constituents) to have a meaning?

Example

AyCaramba serves meat.



Augmented Rules

We'll accomplish this by attaching semantic rules to our CFG rules just as we did with unification.

Abstractly ...

- $A \rightarrow \alpha_1 \ldots \alpha_n \; \{f(\alpha_j.sem, \ldots, \alpha_k.sem)\}$

Augmented Rules

Back to our example . . .

- $ProperNoun \rightarrow AyCaramba \; \{AyCaramba\}$
- $MassNoun \rightarrow meat \; \{Meat\}$
- $NP \rightarrow ProperNoun \; \{ProperNoun.sem\}$
- $NP \rightarrow MassNoun \; \{MassNoun.sem\}$
- $Verb \rightarrow serves \; \{\exists e, x, y Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, y)\}$

Lambda Forms

We've been speaking informally for a while now as if Verbs, and similar objects, were functions. The arguments to these functions are the various constituents in the tree. Therefore the semantics of the Verb should encode the incorporation of its arguments. We can use lambda forms to do that:

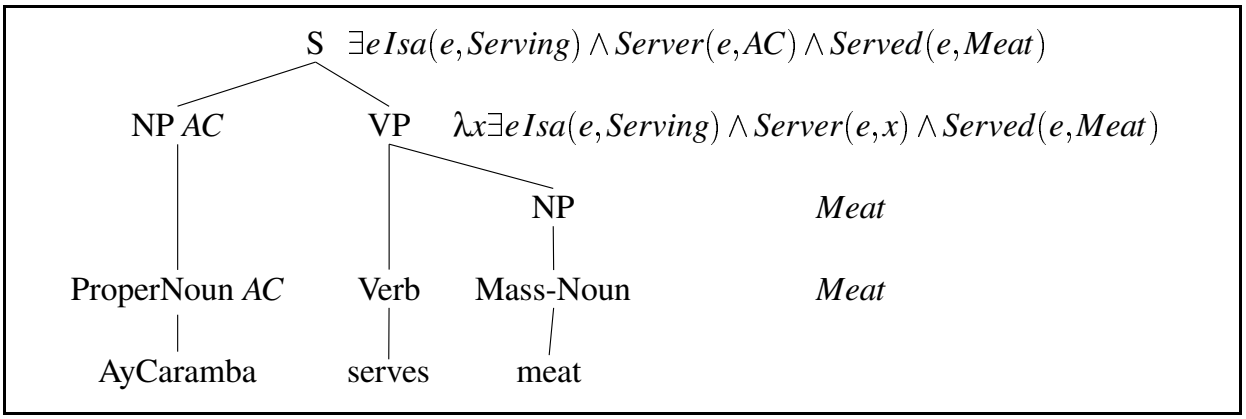
- $\lambda x P(x)$
- $\lambda x P(x)(A)$
- $P(A)$

Lambda Forms (cont.)

- $Verb \rightarrow serves \{\lambda x \lambda y \exists e Isa(e, Serving) \wedge Server(e, y) \wedge Served(e, x)\}$
- $S \rightarrow NPVP\{VP.sem(NP.sem)\}$
- $VP \rightarrow VerbNP\{Verb.sem(NP.sem)\}$

Lambda Forms (cont.)

AyCaramba serves meat.



Another Example

How about . . .

- *A restaurant serves meat.*

Should be . . .

- $\exists e, x Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Meat) \wedge Isa(x, Restaurant)$

Another Example (cont.)

A restaurant serves meat.

Representation of "a restaurant":

- $\exists x Isa(x, Restaurant)$

Representation of "serves meat":

- $\lambda y \exists e Isa(e, Serving) \wedge Server(e, y) \wedge Served(e, Meat)$

Putting them together:

- $\exists e Isa(e, Serving) \wedge Server(e, \exists x Isa(x, Restaurant)) \wedge Served(e, Meat)$

Then how can we get this into the goal expression:

- $\exists e, x Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Meat) \wedge Isa(x, Restaurant)$

Another Example (cont.)

The λ application in the S rule does not do it correctly. It produces something like ...

- $\exists e Isa(e, Serving) \wedge Server(e, \exists x Isa(x, Restaurant)) \wedge Served(e, Meat)$

The problem is that the semantic representation of the subject NP *a restaurant* has parts that need to be distributed or incorporated in various ways into the representation of the VP and hence the S.

Complex Terms

The solution is make the representation of the NP into a kind of object with parts that are accessible.

Complex term

- $\langle\ Quantifier\ variable\ body\ \rangle$
- $\exists x Isa(x, Restaurant)$

Back to the Example

- $\exists e Isa(e, Serving) \wedge Server(e, \langle \exists x Isa(x, Restaurant) \rangle) \wedge Served(e, Meat)$

How do we get from this to a correct FOL form ...

- P($\langle$ *Quantifier variable body* $\rangle$) $\Rightarrow$ *Quantifier Variable body Connective P(variable)*
- $Server(e, \langle \exists x, Isa(x, Restaurant) \rangle)$

$\Rightarrow \exists x Isa(x, Restaurant) \wedge Server(e, x)$

Connectives:

- $\wedge$ for $\exists$ ("there exists")
- $\Rightarrow$ for $\forall$ ("for all")

Goal: $\exists e, x Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Meat) \wedge Isa(x, Restaurant)$

Doing Compositional Semantics

Incorporating semantics into a grammar involves two separate but intertwined processes:

1. Figuring out the right representation for a single constituent based on the parts of that constituent.
2. Figuring out the right representation for a category of constituents based on the other grammar rules that make use of that constituent.

Semantic Analysis

Ok. So we now have a bunch of function-like semantic attachments incorporated into the grammar. How do we do semantic analysis?

There are two possibilities:

1. Just as we did with unification–alter the Early parser so that when completed constituents (dot at the end of the rule) are created, the semantic function attached to the rule is applied and a meaning representation is created and stored with that state.
2. Let the parser run to completion and walk the resulting tree running the semantic attachments from the bottom-up.

Semantic Analysis (cont.)

- $S \rightarrow NP\ VP\ \{VP.sem(NP.sem)\}$

The VP.sem is stored in the state representing the VP, and the NP.sem is stored with the state for the NP. When this rule is completed you go get the VP.sem (it better be a lambda), go get the NP.sem, and apply the VP.sem to the Np.sem and store the result in the S.sem.

Integrated Semantic Analysis

In the integrated style, semantic analysis is fully integrated into the parser. As fragments of the input are parsed semantic fragments are created. This can be used to block ambiguous representations.

- *I want to eat someplace near campus.*

Integrated Semantic Analysis (cont.)

But . . . it also means that you're performing semantic analysis on orphaned constituents that play no role in a final parse. Hence a pipelined semantic analysis might be better.

Non-Compositional Language

Unfortunately, there are lots of examples of non-compositional inputs – inputs where the final meaning is not derived from the meanings of the component parts alone.

- jokes
- irony
- indirect requests
- metaphor
- idioms
- . . .

Kick the Bucket

$VP \rightarrow kick\ the\ bucket\ \{\lambda x Die(x)\}$

Tip of the iceberg

Coupons are just *the tip of the iceberg.*

The SEC's allegations are only *the tip of the iceberg.*

Coronary bypass surgery, hip replacement and intensive-care units are but *the tip of the iceberg.*

BUT

And that's but *the tip of Mrs. Ford's iceberg.*

These comments describe only *the tip of a 1,000-page iceberg.*

The 10 employees represent *the merest tip of the iceberg.*

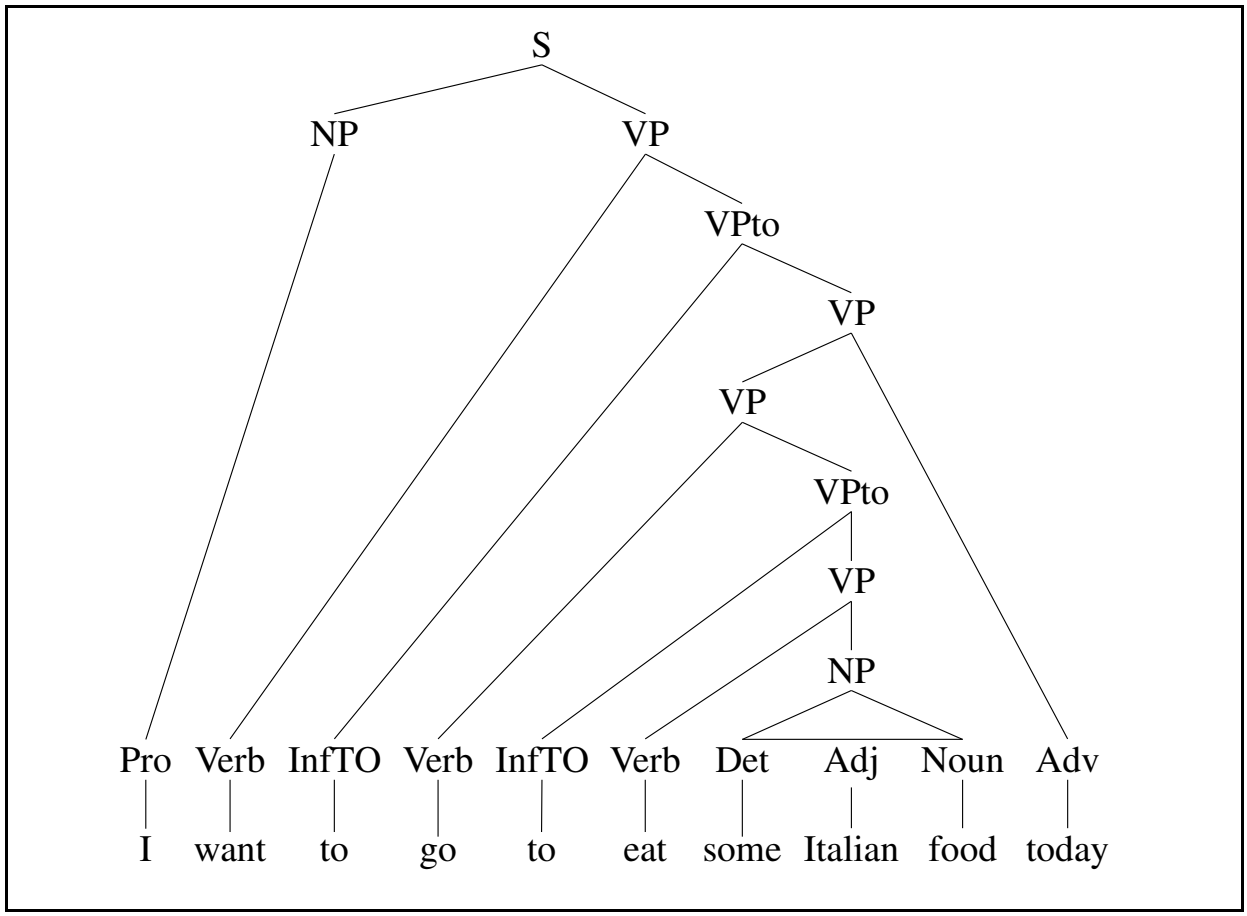
Syntactic Grammar Problems

1. Key semantic elements often widely distributed across parse trees.

2. Parse trees often contain syntactically motivated constituents that play no role in semantic processing.

3. The general nature of syntactic constituents results in nearly vacuous meanings.

Problems 1 and 2



Problem 3

I want to go to eat some Italian food today.

Using the representations we've seen so far to represent the NP *Italian food*, all we know is that the nominal is modified by the adjective.

- $Nom \rightarrow AdjNom \{\lambda xNom.sem(x) \wedge AM(x, Adj.sem)\}$
- $\exists x ISA(x, Food) \wedge AM(x, Italian)$

So *Italian food*, *Italian restaurant*, ... would all get the same (vague) interpretation.

Robust Variants

Semantic Grammars

Information Extraction and Finite-State Methods

Semantic Grammars

As we've seen, a problem with syntactically motivated grammars is that they often don't served semantic purposes very well. One solution is to fight to make them work (as in the first part of Chapter 15).

An alternative is to work with grammars that closely match the semantics. Such grammars often go by the name of semantic grammars.

- key semantic components occur together within single rules
- rules are no more general than needed

Example

I want to go to eat Italian food.

We can dump the notion of sentence and parse looking for something closer to what users say in this domain.

- *InfoRequest* $\rightarrow$ *User want to go to eat Foodtype*
- *Foodtype* $\rightarrow$ *Italian food*

Drawback: almost no reuse

- grammar is domain specific
- more rules needed (*Italian food*, *Italian restaurant*, ...)

Information Extraction

A different kind of work-around is called for with what are called information extraction tasks.

In these tasks, a system is asked to deal with arbitrary running written text and to form shallow attribute-value meaning representations suitable for insertion into a database.

Information Extraction Examples

Scanning newspapers/newswires for financial events of interest

Scanning websites for price/sales information of interest

Scanning radio/tv broadcasts

These tasks may involve extended discourse and typically involve text with long sentences, complex syntax, large vocabularies, multiple writer/speakers, ...

Robustness

Unfortunately, there are practical problems with using syntactically motivated grammars here as well...

Large broad coverage context grammars for English are hard to come by.

The degree of ambiguity in them makes them nearly unusable for arbitrary text.

When you don't get 50 parses, you end up with none.

Parsing is too slow for many current commercial applications.

Robustness (cont.)

Unfortunately, the semantic grammar trick does not scale well to the information extraction domain. A small domain specific grammar can not be used to parse/analyze text from an unrestricted domain.

So we can stay with a syntax-driven approach but fall back from context-free methods to finite state methods.

Finite-State Methods

One approach is to go a cascaded finite-state approach.

In this approach, a series of finite-state transducers are applied to the input. At each stage, some small aspect of syntax/semantics is captured and enclosed so it is available for the next stage.

The end result is typically a simple database scheme with slots to be filled in for the information of interest. The slots can be filled either with strings from the input text or symbols from some small alphabet.

FASTUS Example

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and "metal wood" clubs a month.

FASTUS Example (cont.)

TIE-UP-1:	
Relationship:	TIE-UP
Entities:	"Bridgestone Sports Co."
	"a local concern"
	"a Japanese trading house"
Joint Venture Company	"Bridgestone Sports Taiwan Co."
Activity	ACTIVITY-1
Amount	NT$20000000
ACTIVITY-1:	
Company	"Bridgestone Sports Taiwan Co."
Product	"iron and "metal wood" clubs"
Start Date	DURING: January 1990

Cascades

Work from the input up, doing the easy things first (like tagging).

No.	Step	Description
1	**Tokens:**	Transfer an input stream of characters into a token sequence.
2	**Complex Words:**	Recognize multi-word phrases, numbers, and proper names.
3	**Basic phrases:**	Segment sentences into noun groups, verb groups, and particles.
4	**Complex phrases:**	Identify complex noun groups and complex verb groups.
5	**Semantic Patterns:**	Identify semantic entities and events and insert into templates.
6	**Merging:**	Merge references to the same entity or event from different parts of the text.

Key Point

What about material that is not of interest or can not be processed?

Skipped. Just isn't written out the tape for the next level of processing.

Some systems even use a filter to identify sentences that are not relevant to the task and remove them to begin with.

FASTUS: Basic Phrase Extractor

Company	Bridgestone Sports Co.
Verb Group	said
Noun Group	Friday
Noun Group	it
Verb Group	had set up
Noun Group	a joint venture
Preposition	in
Location	Taiwan
Preposition	with
Noun Group	a local concern
Conjunction	and
Noun Group	a Japanese trading house
Verb Group	to produce
Noun Group	golf clubs
Verb Group	to be shipped
Preposition	to
Location	Japan

FASTUS: Semantic Templates

(1)	Relationship: Entities:	TIE-UP “Bridgestone Sports Co.” “a local concern” “a Japanese trading house”
(2)	Activity Product	PRODUCTION “golf clubs”
(3)	Relationship: Joint Venture Company Amount	TIE-UP “Bridgestone Sports Taiwan Co.” NT$20000000
(4)	Activity Company Start Date	PRODUCTION “Bridgestone Sports Taiwan Co.” DURING: January 1990
(5)	Activity Product	PRODUCTION “iron and “metal wood” clubs”

For Next Time

Chapter 16