Phrases and Sentences

1. Language models
2. Chunking
3. Structural descriptions
4. Parsing with phrase structure grammars
5. Probabilistic parsers
6. Parsing with dependency grammars
7. Principles and Parameters
8. Unification-based grammars
9. Semantics construction

Phrases and Sentences

1. Language models
2. Chunking
3. Structural descriptions
4. Parsing with phrase structure grammars
5. Probabilistic parsers
6. Parsing with dependency grammars
7. Principles and Parameters
8. Unification-based grammars
9. Semantics construction

Semantics construction

- The standard model
- Learning the syntax-semantics mapping

slides designed after KOLLER (2015)

Semantics construction

- target structure: logical form
- usually a (restricted) first order logic
- compromise between expressiveness and computational efficiency

A man rides a bicycle.

$$man'(x) \wedge bicycle'(y) \wedge riding'(x, y)$$

$$\exists x \exists y . man'(x) \wedge bicycle'(y) \wedge riding'(x, y)$$

$$\exists x \exists y \exists z . type(x, man) \wedge type(y, bicycle)$$
$$\wedge event(z) \wedge type(z, riding) \wedge agent(z, x) \wedge theme(z, y)$$

Semantics construction

- compositional semantics construction
- compute the meaning of an utterance from the meaning of its sub-expressions, guided by the syntactic structure of the utterance

- What are the most elementary meaning representations, i.e. the lexical entries?
- How can partial meaning representations be combined?

Lambda calculus

- λ-expression: application of a function to its argument values
- incomplete semantic expressions understood as functions of yet to be filled in information pieces (arguments)

$\langle X \textit{ rides } Y \rangle$

$$\lambda P\, \lambda Q \;\; \exists x\, \exists y \;.\; P(x) \wedge Q(y) \wedge \textit{riding}'(x, y)$$

is a function of two arguments P and Q

Lambda calculus

- application by means of β reduction (illustrated with a simplified treatment of quantification)

  X rides a bicycle

  $$\lambda P\, \lambda Q\ \ \exists x\, \exists y\ .\ Q(x) \wedge P(y) \wedge \mathit{riding}'(x, y)\ (\mathit{bicycle}')$$

  $$\rightarrow_\beta\ \lambda Q\ \ \exists x\, \exists y\ .\ Q(x) \wedge \mathit{bicycle}'(y) \wedge \mathit{riding}'(x, y)$$

  A man rides a bicycle

  $$\lambda Q\ \ \exists x\, \exists y\ .\ Q(x) \wedge \mathit{bicycle}'(y) \wedge \mathit{riding}'(x, y)\ (\mathit{man}')$$

  $$\rightarrow_\beta\ \ \exists x\, \exists y\ .\ \mathit{man}'(x) \wedge \mathit{bicycle}'(y) \wedge \mathit{riding}'(x, y)$$

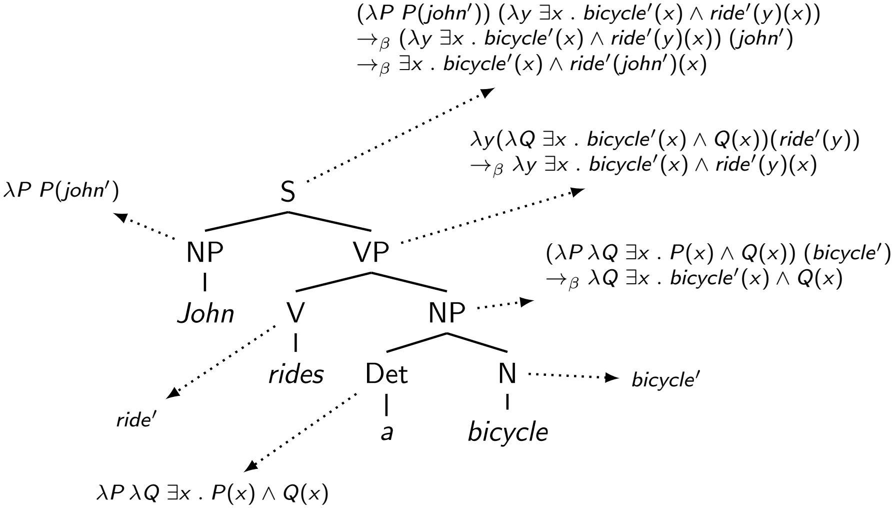
- construction of n-ary function symbols from simpler ones

  $$(\lambda Q\ Q(x))\ (p(y)) \rightarrow_\beta p(y)(x) \qquad\qquad [\,p(y)(x) \equiv p(y, x)\,]$$

Lambda calculus

S $\rightarrow$ NP VP	$\langle \mathrm{S} \rangle = \langle \mathrm{NP} \rangle(\langle \mathrm{VP} \rangle)$
VP $\rightarrow$ V NP	$\langle \mathrm{VP} \rangle = \lambda x \langle \mathrm{NP} \rangle(\langle \mathrm{V} \rangle(x))$
NP $\rightarrow$ Det N	$\langle \mathrm{NP} \rangle = \langle \mathrm{Det} \rangle(\langle \mathrm{N} \rangle)$
NP $\rightarrow$ *John*	$\langle \mathrm{NP} \rangle = \lambda P\, P(\mathrm{john'})$
V $\rightarrow$ *rides*	$\langle \mathrm{V} \rangle = \mathrm{riding'}$
Det $\rightarrow$ *a*	$\langle \mathrm{Det} \rangle = \lambda P\, \lambda Q\, \exists x\, .\, P(x) \wedge Q(x)$
N $\rightarrow$ *bicycle*	$\langle \mathrm{N} \rangle = \mathrm{bicycle'}$

Lambda calculus



Semantic Ambiguity

- prototypical problem: scope ambiguity

  Every man loves a woman

  wide scope: $\forall x.man'(x) \rightarrow (\exists y.woman'(y) \land love'(x,y))$

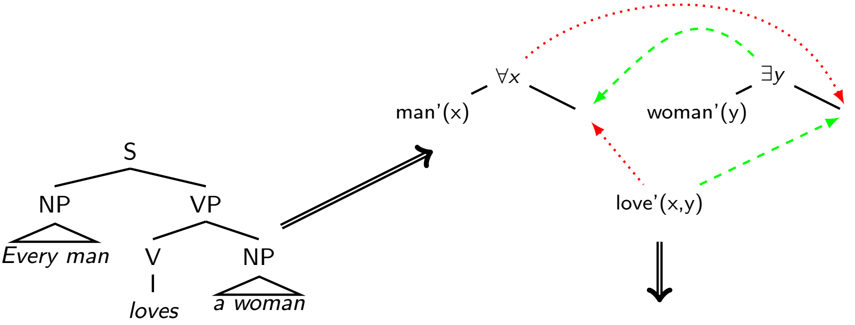
  narrow scope: $\exists y.woman'(y) \land (\forall x.man'(x) \rightarrow love'(x,y))$

Semantic Ambiguity

three approaches to deal with semantic ambiguity

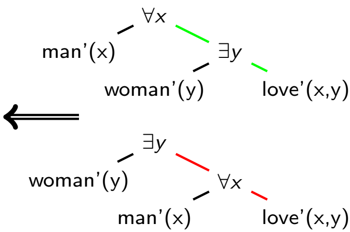
- Montague Grammar: quantifying in
  - raising the NP
  - scope ambiguity becomes a syntactic one
- Cooper storage: keep quantifiers temporarily on a storage to process it later
  - makes semantics construction a non-deterministic procedure
- underspecification: create a description from which all the individual interpretations can be recovered on demand
  - leaves the ambiguity implicit
  - enumeration of the interpretations only when really necessary

Underspecification



$$\forall x.man'(x) \rightarrow (\exists y.woman'(y) \wedge love'(x,y))$$

$$\exists y.woman'(y) \wedge (\forall x.man'(x) \rightarrow love(x,y))$$



Underspecification

- underspecification allows the parser
  - to delay the enumeration of the different interpretations
    - until perhaps one or all of them can be eliminated anyhow
  - to combine alternative, but logically equivalent descriptions (redundancy elimination)
    - e.g. if twice the same quantifier is used

      A man loves a women.

      $\exists x.man'(x) \land (\exists y.woman'(y) \land love'(x,y))$

      $\exists y.woman'(y) \land (\exists x.man'(x) \land love'(x,y))$

Learning the syntax-semantics mapping

- inducing the mapping from annotated corpus data
  - given: a collection of sentences with their syntactic structures and their semantic representations
- often using simplified semantic representations for special purposes
  - e.g. controlling a robot or quering a database
- e.g. Geoquery corpus
  - 880 questions to a geographic data base

    What is the smallest state by area?

    `answer(x1, smallest(x2, state(x1), area(x1, x2)))`
  - all variables are universally quantified
- more general representations: AMR corpus

Rule extraction

- general procedure based on the idea of compositional semantics
  - training: decompose the meaning representations into elementary building blocks that can be assigned to individual lexical items
    - guided by the syntactic structure
    - and (possibly) by an alignment between the lexical items in the input and the predicate symbols of the meaning representation
  - parsing: combine the elementary building blocks into a complete meaning representation for the whole utterance

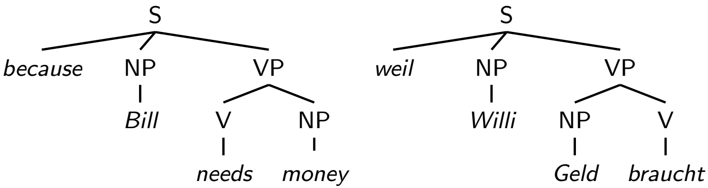
Rule extraction

- most serious problem: many different logical formulas can express an identical meaning
  - e.g. $p(a,b) \wedge q(a,c) \equiv q(a,c) \wedge p(a,b)$
  - results in a huge artificial (lexical) ambiguity
    - that creates an intractably large search space for the parser
    - and leads to poorly trained models because of data sparseness

Rule extraction

- early approach (Wong and Mooney 2007)
    - based on synchroneous context-free grammars (SCFG)
- SCFGs have been originally introduced for machine translation

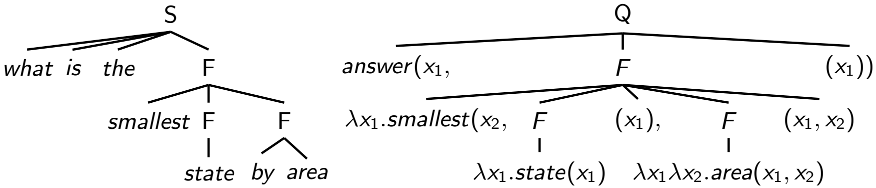
S $\rightarrow$ because NP VP	S $\rightarrow$ weil NP VP
NP $\rightarrow$ Bill | money	NP $\rightarrow$ Willi | Geld
VP $\rightarrow$ V NP	VP $\rightarrow$ NP V
V $\rightarrow$ needs	V $\rightarrow$ braucht



Rule extraction

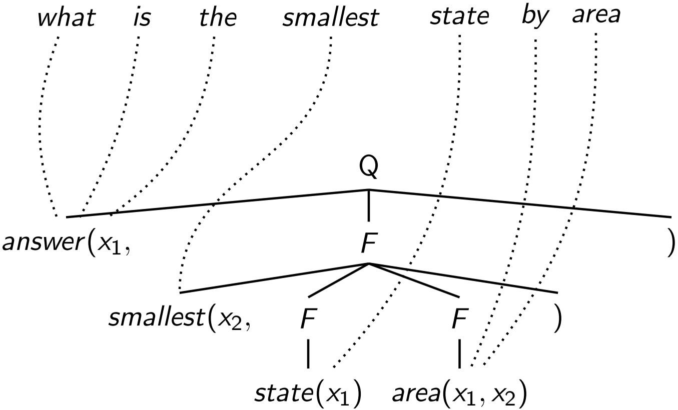
- λ-SCFG: synchronous derivation of a syntax tree and a λ-expression

Q $\rightarrow$ *what is the* F	$Q \rightarrow answer(x_1, F(x_1))$
F $\rightarrow$ *smallest* F F	$F \rightarrow \lambda x_1 \,.\, smallest(x_2, F(x_1), F(x_1, x_2))$
F $\rightarrow$ *state*	$F \rightarrow \lambda x_1 \,.\, state(x_1)$
F $\rightarrow$ *by area*	$F \rightarrow \lambda x_1 \lambda x_2 \,.\, F(x_1, x_2)$



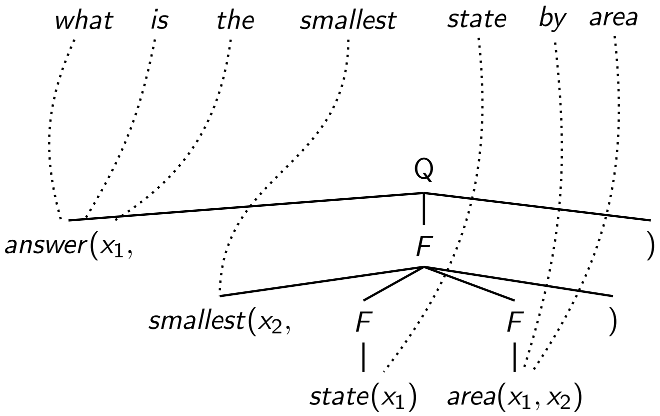
Rule extraction

- GeoQuery: representations without quantifiers
- alignment with the input word forms



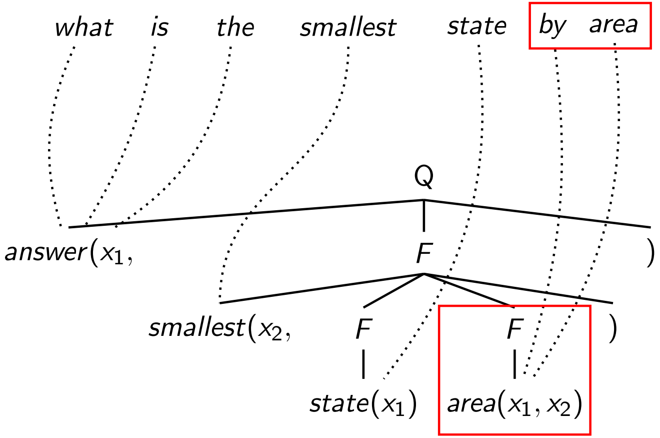
Rule extraction

- extraction of the mapping rules for the constituents



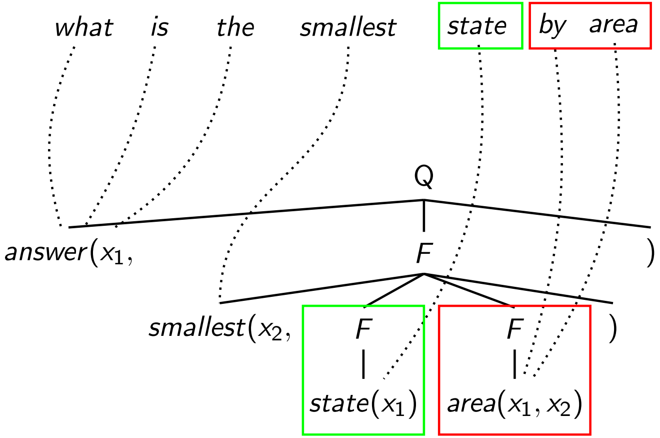
Rule extraction

- extraction of the mapping rules for the constituents



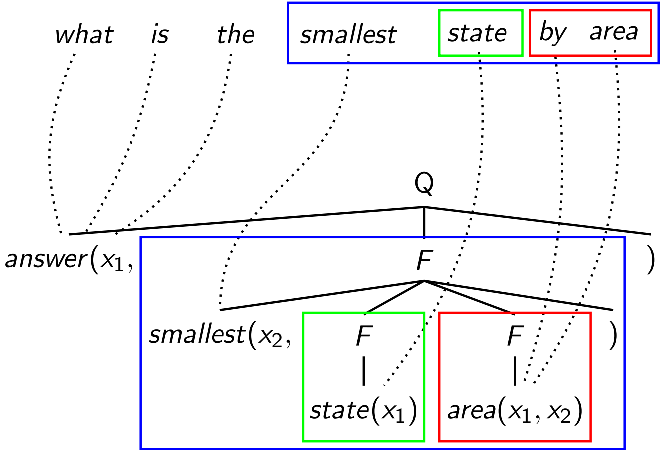
Rule extraction

- extraction of the mapping rules for the constituents



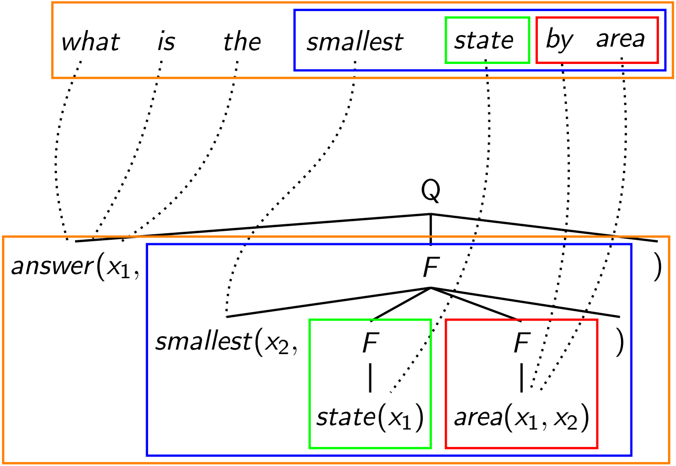
Rule extraction

- extraction of the mapping rules for the constituents



Rule extraction

- extraction of the mapping rules for the constituents



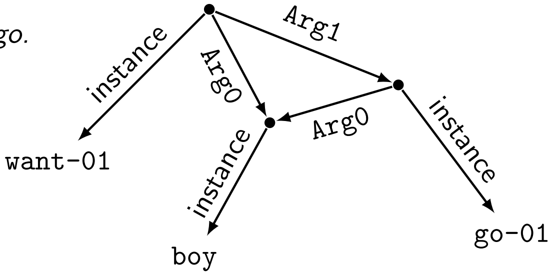
Evaluation

- abstract meaning representations consist of
  - predicate symbols (frames) taken from PropBank, e.g.

wants	`wants-01(Arg0,Arg1)`
go	`go-1(Arg0)`

  - entities, e.g. `boy`
- a semantic representation can be depicted as a directed graph

The boy wants to go.



Evaluation

- graph can be represented as a conjunction of AMR triples (propositions)

```
instance(a,want-01)
instance(b,boy)
instance(c,go-1)
Arg0(a,b)
Arg1(a,c)
Arg0(c,b)
```

- similarity of two graphs is measured as the propositional overlap between them
- precision, recall, and f-score can be computed

Evaluation

- problem: node names (variables) are not shared between different graphs
- multiple alternatives to map variables to each other
  - might result in different propositional overlaps
- the mapping with the maximum overlap has to be determined
  - finding the optimum is NP complete
  - optimal solution can be determined using e.g. Integer Linear Programming ...
  - ... or approximated by heuristic search

Evaluation

- state of the art (Artzi, Lee and Zettlemoyer 2015)
  - rule extraction based on Combinatory Categorial Grammar
  - without an alignment with the input sentence
  - special mechanism for non-compositional aspects of meaning
    - i.e. sentence-internal coreference relationships
  - 66.2 Smatch F1 score on the AMR bank