Semantic parsing

BM1 Advanced Natural Language Processing

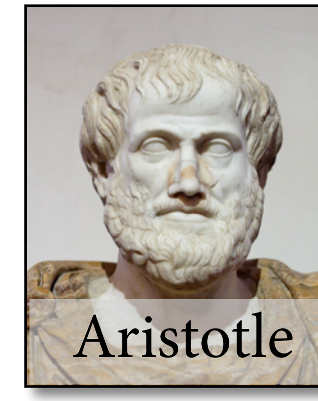
Alexander Koller

06 January 2015

The rest of the course

- Semantics
  - semantic representations and how to compute them
  - corpus-based approaches to semantic knowledge
- Bayesian methods
- Machine translation

Computing with meanings



- Ancient problem: *inference.*
  - How can we tell whether a sentence follows from others?
  - Can we compute this automatically?

All men are mortal.

Socrates is a man.

Therefore, Socrates is mortal.

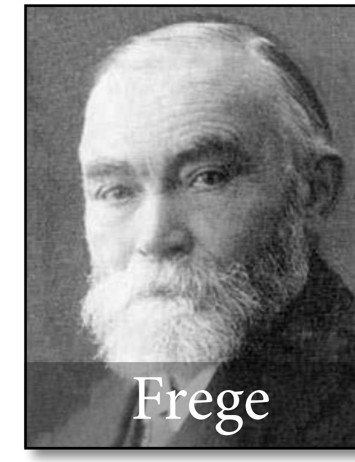
Applications

- Inference is useful in many applications of CL, e.g. question answering and summarization.

Which genetically caused connective tissue disorder has severe symptoms and complications regarding the aorta and skeletal features, and, very characteristically, ophthalmologic subluxation?

Marfan's is created by a defect of the gene that determines the structure of Fibrillin-11. One of the symptoms is displacement of one or both of the eyes' lenses. The most serious complications affect the cardiovascular system, especially heart valves and the aorta.

Formal meaning representations



- Modern approach to natural-language inference:
  - Compute *meaning representation* in some formal language (e.g. predicate logic)
  - so that it captures something relevant about the sentence's meaning (e.g. its *truth conditions*)
  - and then use reasoning tools for the formal language (e.g. a *theorem prover* for predicate logic)

All men are mortal.
Socrates is a man.

Therefore, Socrates is mortal.

$\forall x.\ \text{man}(x) \rightarrow \text{mortal}(x)$
$\text{man}(s)$

$\text{mortal}(s)$

Key questions

- How can we compute formal meaning representations from sentences?
- How can we formalize the necessary world knowledge that we need for inferences?
- How can we carry out the inferences efficiently?

Some challenges

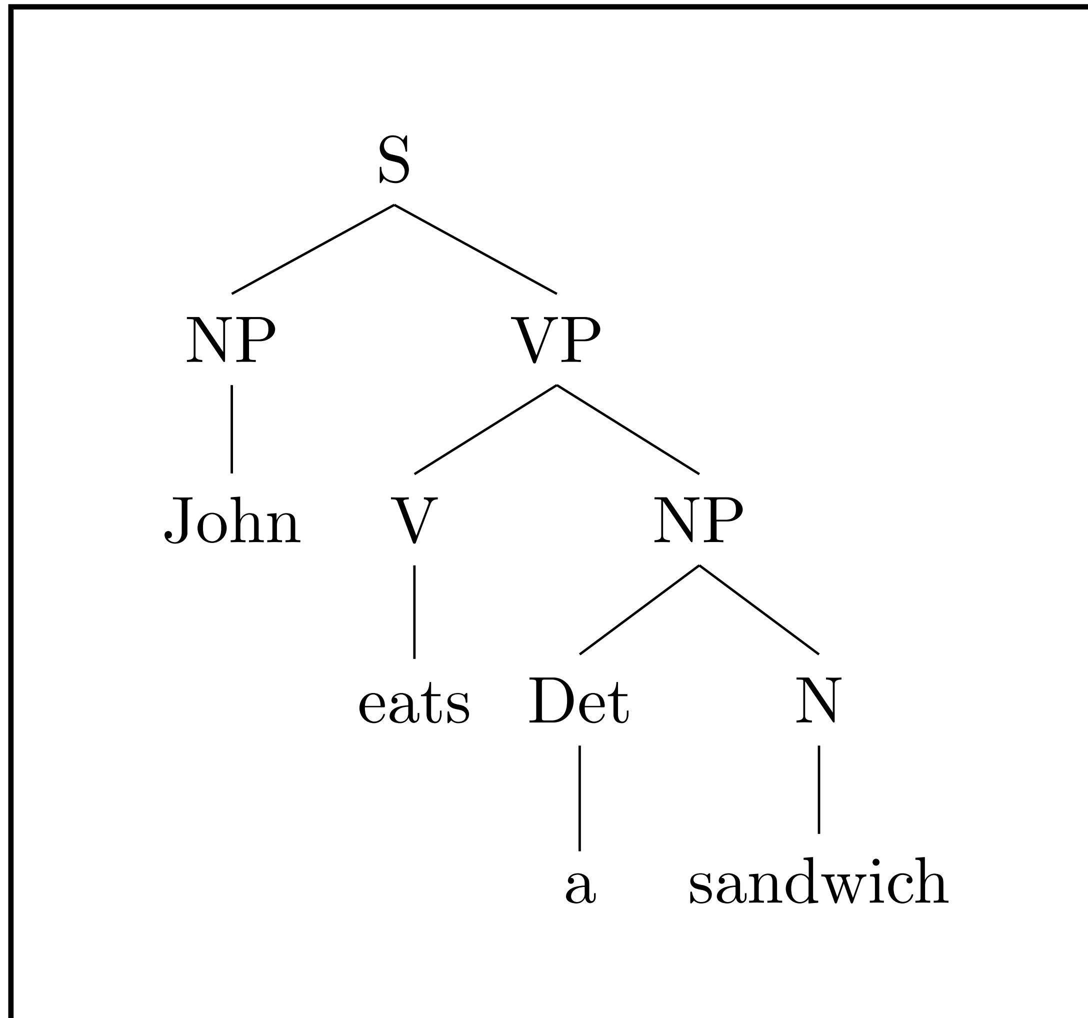
- Predicate-argument structure:
  - “dog bites man” vs. “man bites dog”
- Passive:
  - “the dog bites the man” vs. “the dog is bitten by the man”
- Control and raising:
  - “John wants to sleep”
- Complex logical operators:
  - “Every student did not pass the exam”

Semantics construction

- *Semantics construction* is the problem of deriving a formal meaning representation from a syntax tree.

- Mostly agreed that semantics construction should be *compositional*.
    - Essentially: meaning of a larger constituent is determined by meanings of its parts and the grammar rule.
    - This corresponds to bottom-up evaluation of syntax tree, so is really convenient in practice.
    - Some things hard to analyze compositionally, e.g. coreference/anaphora.

An example

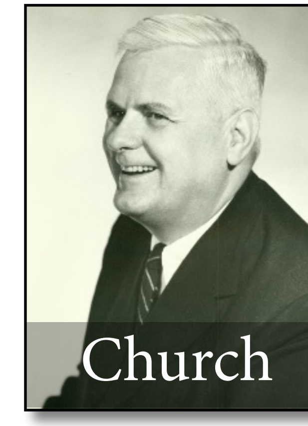


$\Big\downarrow$?

target representation:

$\exists x.\ \text{sandwich}(x) \wedge \text{eat}(x)(j)$

Lambda calculus



- Term notation for functions:

$$\lambda P\ \lambda x.\ P(x) \wedge \mathrm{sleep}(x)$$

two arguments function body

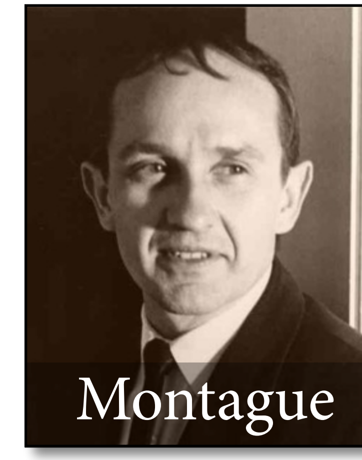
- Apply one lambda term to another and simplify with *beta reduction:*

$$(\lambda P\ \lambda x.\ P(x) \wedge \mathrm{sleep}(x))\ (\mathrm{boy}) \quad \rightarrow_\beta \quad \lambda x.\ \mathrm{boy}(x) \wedge \mathrm{sleep}(x)$$

$$(\lambda x.\ \mathrm{boy}(x) \wedge \mathrm{sleep}(x))\ (a) \quad \rightarrow_\beta \quad \mathrm{boy}(a) \wedge \mathrm{sleep}(x)$$

(full lambda calculus also has α and η equivalence)

Syntax-semantics interface



S → NP VP	$\langle S \rangle = \langle NP \rangle(\langle VP \rangle)$
VP → V NP	$\langle VP \rangle = \lambda y\ \langle NP \rangle(\langle V \rangle(y))$
NP → Det N	$\langle NP \rangle = \langle Det \rangle(\langle N \rangle)$
NP → John	$\langle NP \rangle = \lambda P\ P(j')$
V → eats	$\langle V \rangle = eat'$
Det → a	$\langle Det \rangle = \lambda P \lambda Q \exists x\ P(x) \wedge Q(x)$
N → sandwich	$\langle N \rangle = sw'$

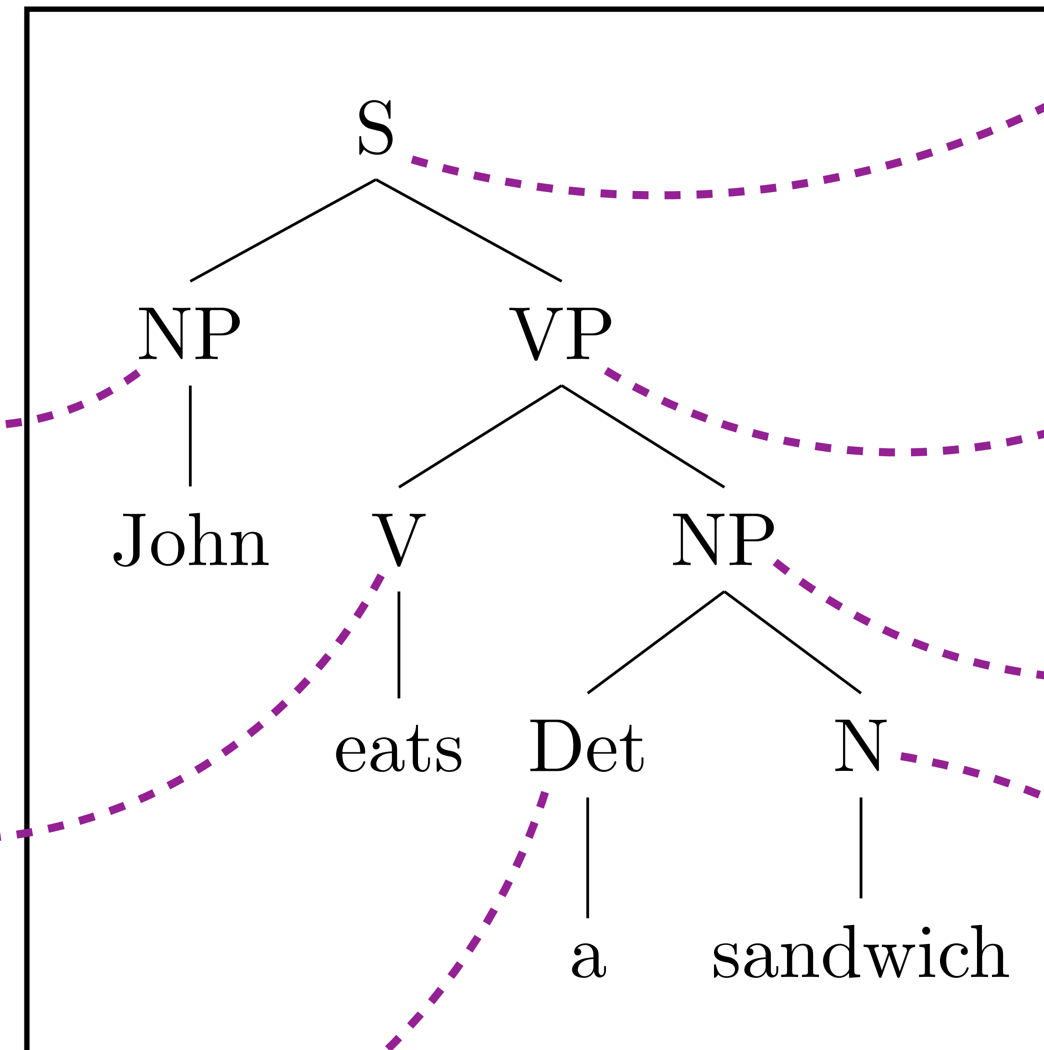
when you apply this syntax rule …

… construct λ-term for parent from λ-terms for children like this

Example

$$(\lambda P\, P(j'))\, (\lambda y\, \exists x\, sw'(x) \wedge eat'(y)(x))$$
$$\to_\beta (\lambda y\, \exists x\, sw'(x) \wedge eat'(y)(x))(j')$$
$$\to_\beta \exists x\, sw'(x) \wedge eat'(j')(x)$$

$\lambda P\, P(j')$



$$\lambda y\ (\lambda Q \exists x\, sw'(x) \wedge Q(x))(eat'(y))$$
$$\to_\beta \lambda y\, \exists x\, sw'(x) \wedge eat'(y)(x)$$

$$(\lambda P \lambda Q \exists x\, P(x) \wedge Q(x))(sw')$$
$$\to_\beta \lambda Q \exists x\, sw'(x) \wedge Q(x)$$

eat'

sw'

$\lambda P \lambda Q \exists x\, P(x) \wedge Q(x)$

Further topics

- More complex semantic phenomena.
  - e.g. intensional operators: "John seeks a unicorn."
- Dealing correctly with quantifiers and negation.
  - "Every man loves a woman."
  - Montague has hacky solution, Cooper storage improves.
- Underspecification approaches.
  - avoid expensive enumeration of semantic ambiguities
  - standard approach for large grammars

Semantic parsing

- Open issue in classical semantics construction: Where do we get large grammar that supports it?

- Current trend in CL is *semantic parsing*: learn mapping from sentence to formal meaning representation using statistical methods.

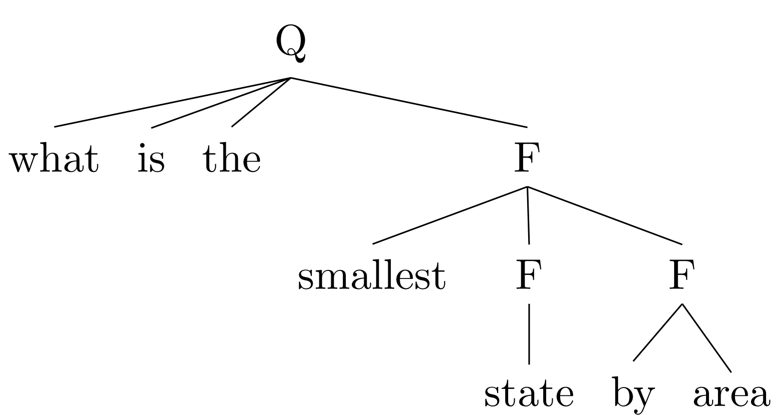
- E.g. from Geoquery corpus (880 sentences):

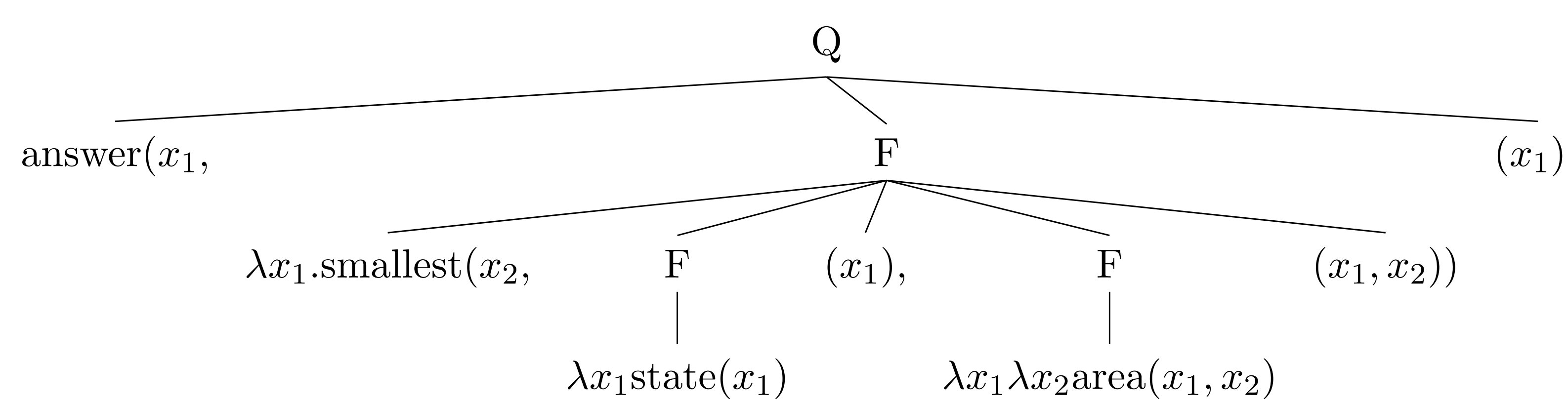
> What is the smallest state by area?
>
> answer(x_1, smallest(x_2, state(x_1), area(x_1, x_2)))

With synchronous grammars

- A *synchronous grammar* simultaneously derives two structures; e.g., a syntax tree and a meaning rep.

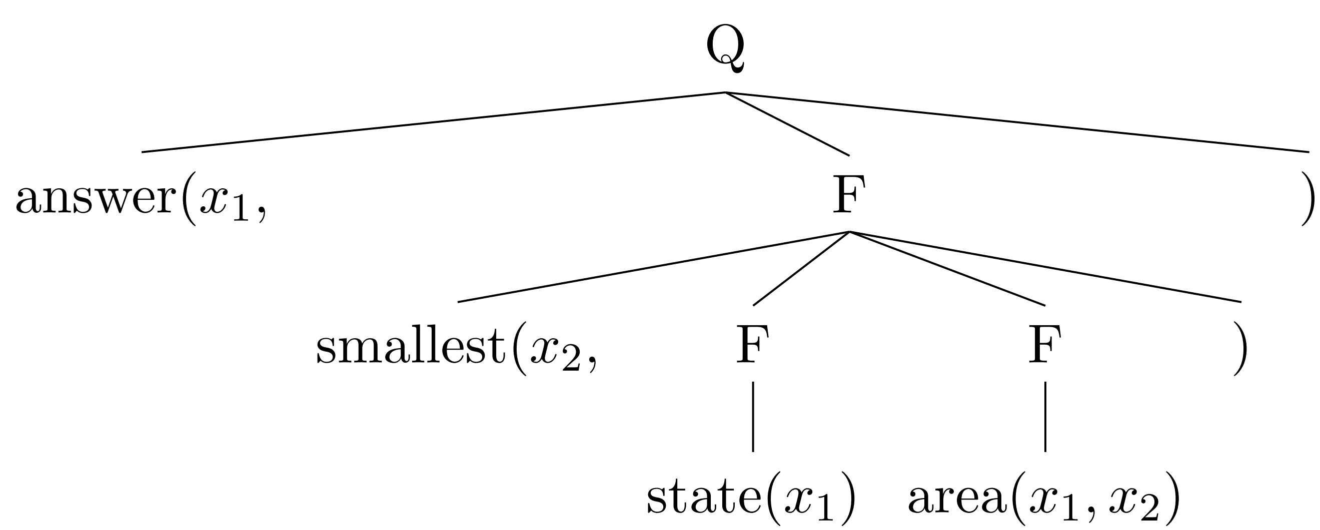
Q → what is the F	Q → answer(x_1, F(x_1))
F → smallest F F	F → λx_1 smallest(x_2, F(x_1), F(x_1, x_2))
F → state	F → λx_1 state(x_1)
F → by area	F → λx_1 λx_2 area(x_1, x_2)





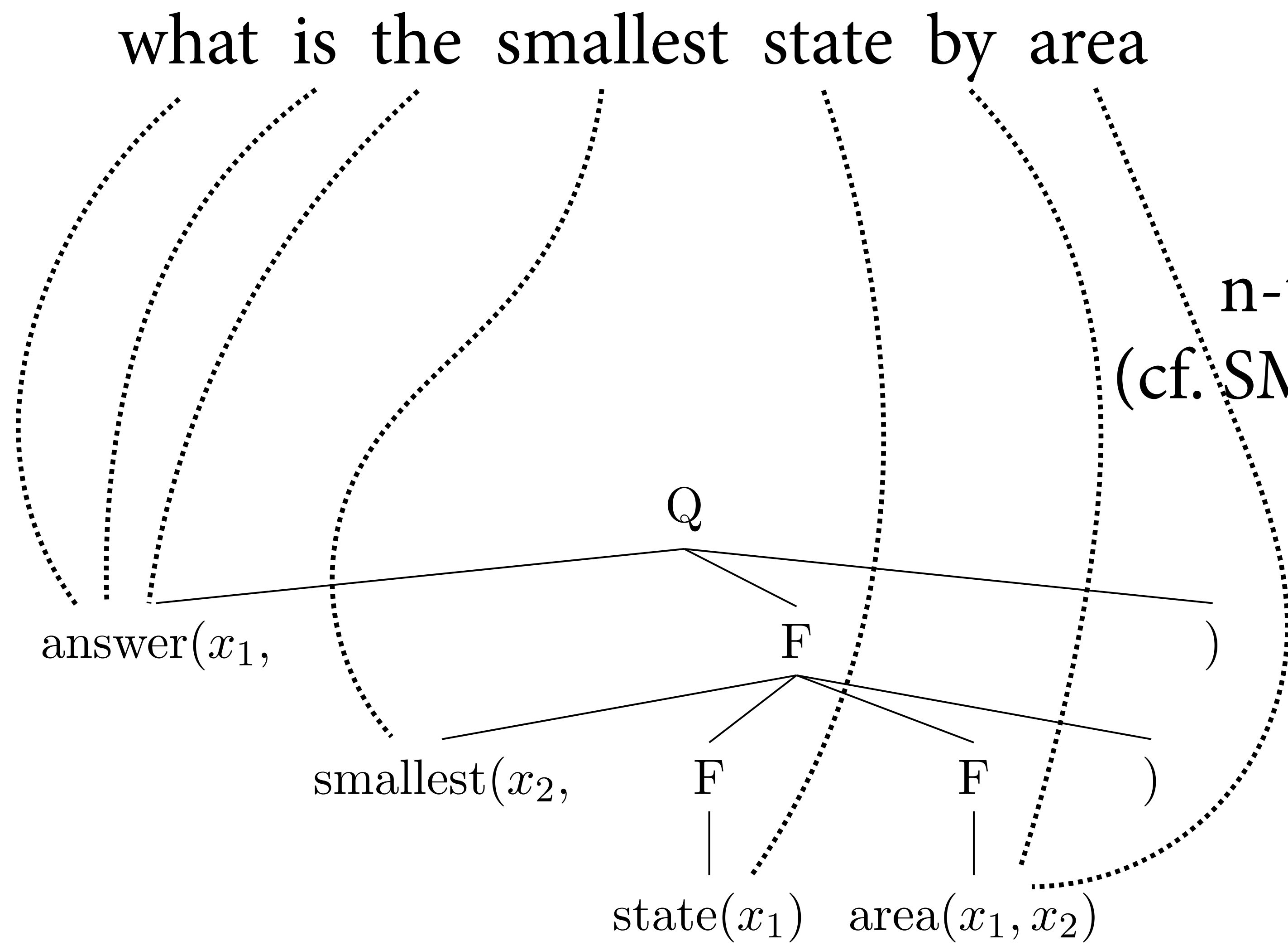
Wong & Mooney

what is the smallest state by area



Assumptions:
- n-to-1 alignment between words and nodes
- unambiguous structure of meaning representation

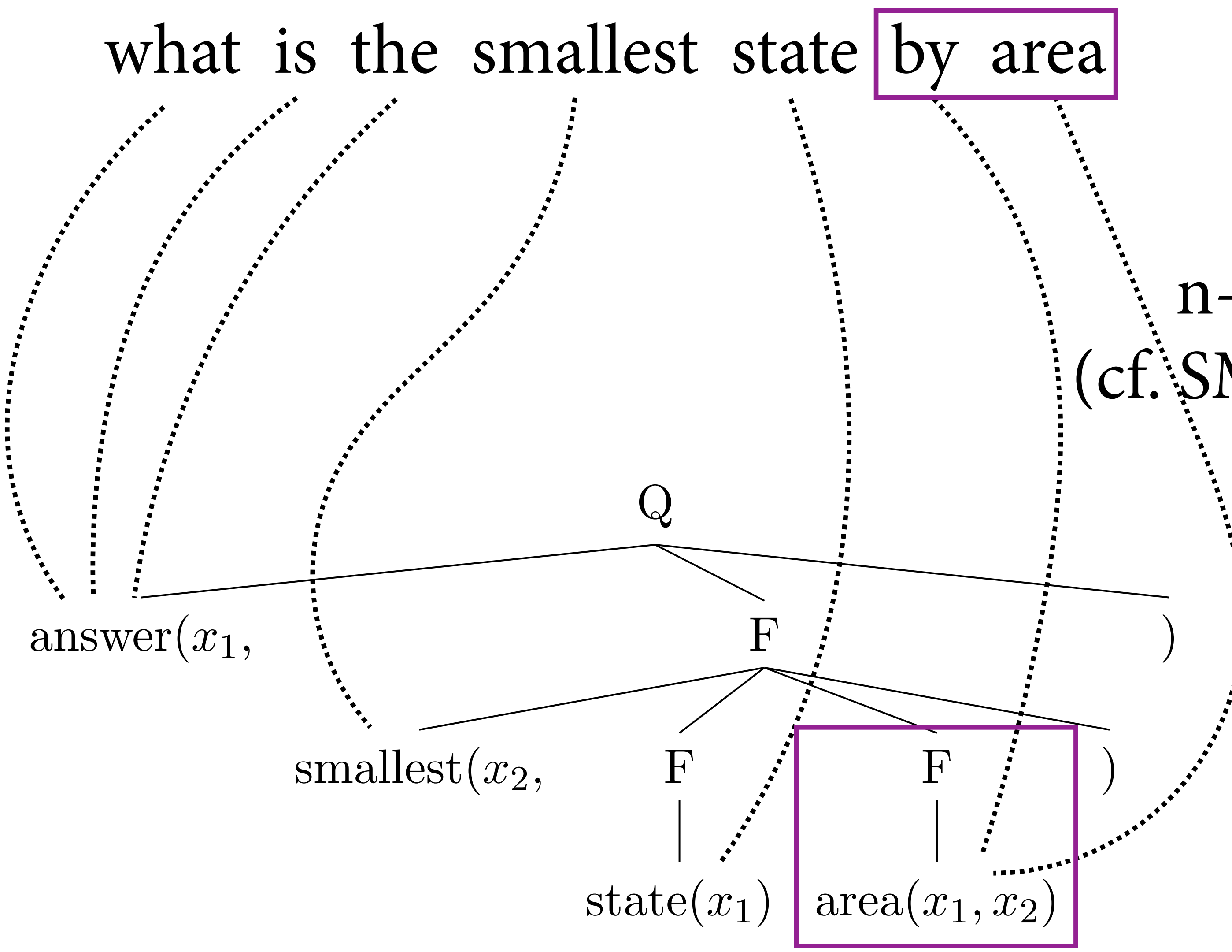
Wong & Mooney



n-to-1 alignment
(cf. SMT in a few weeks)

Assumptions:
- n-to-1 alignment between words and nodes
- unambiguous structure of meaning representation

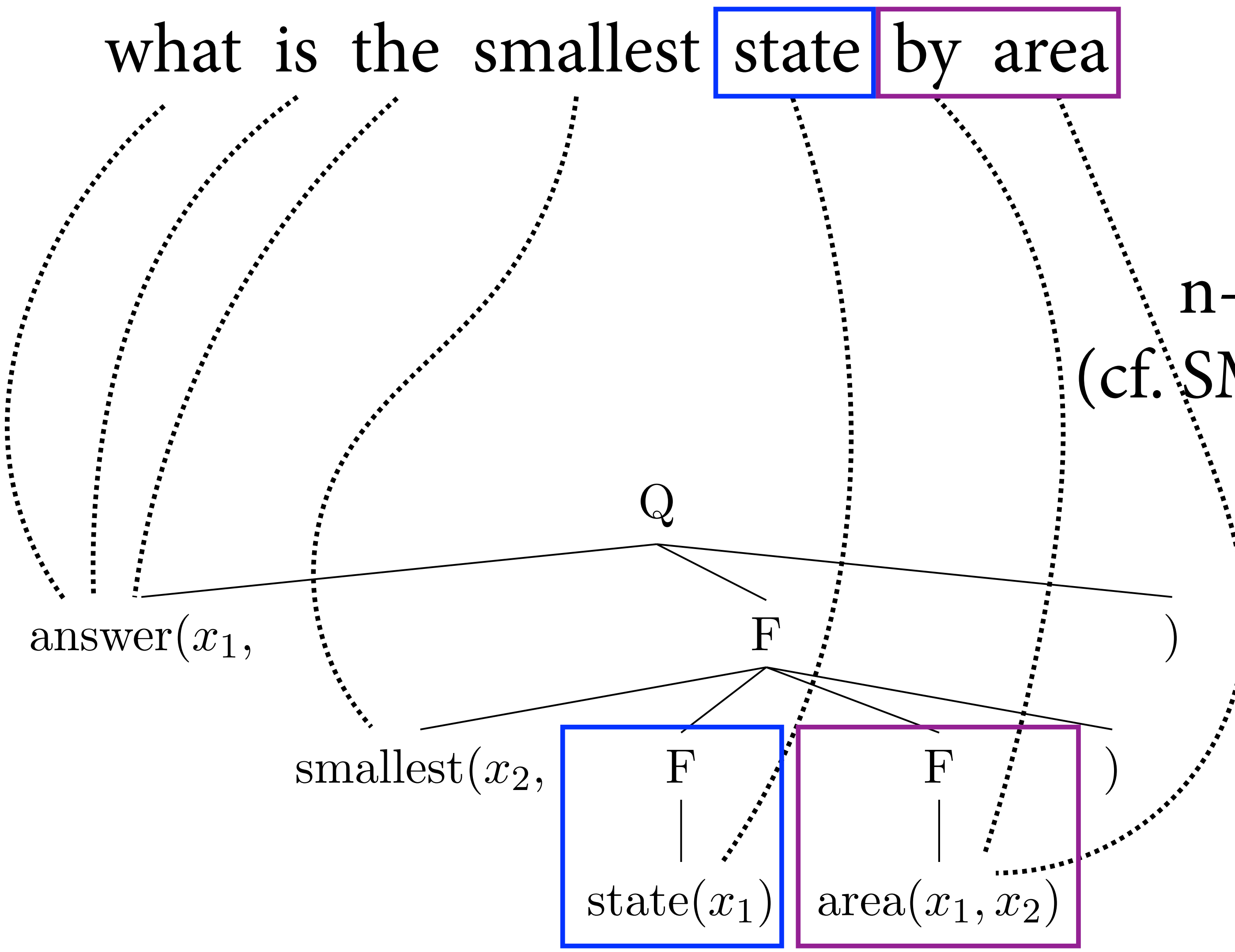
Wong & Mooney



n-to-1 alignment
(cf. SMT in a few weeks)

Assumptions:
- n-to-1 alignment between words and nodes
- unambiguous structure of meaning representation

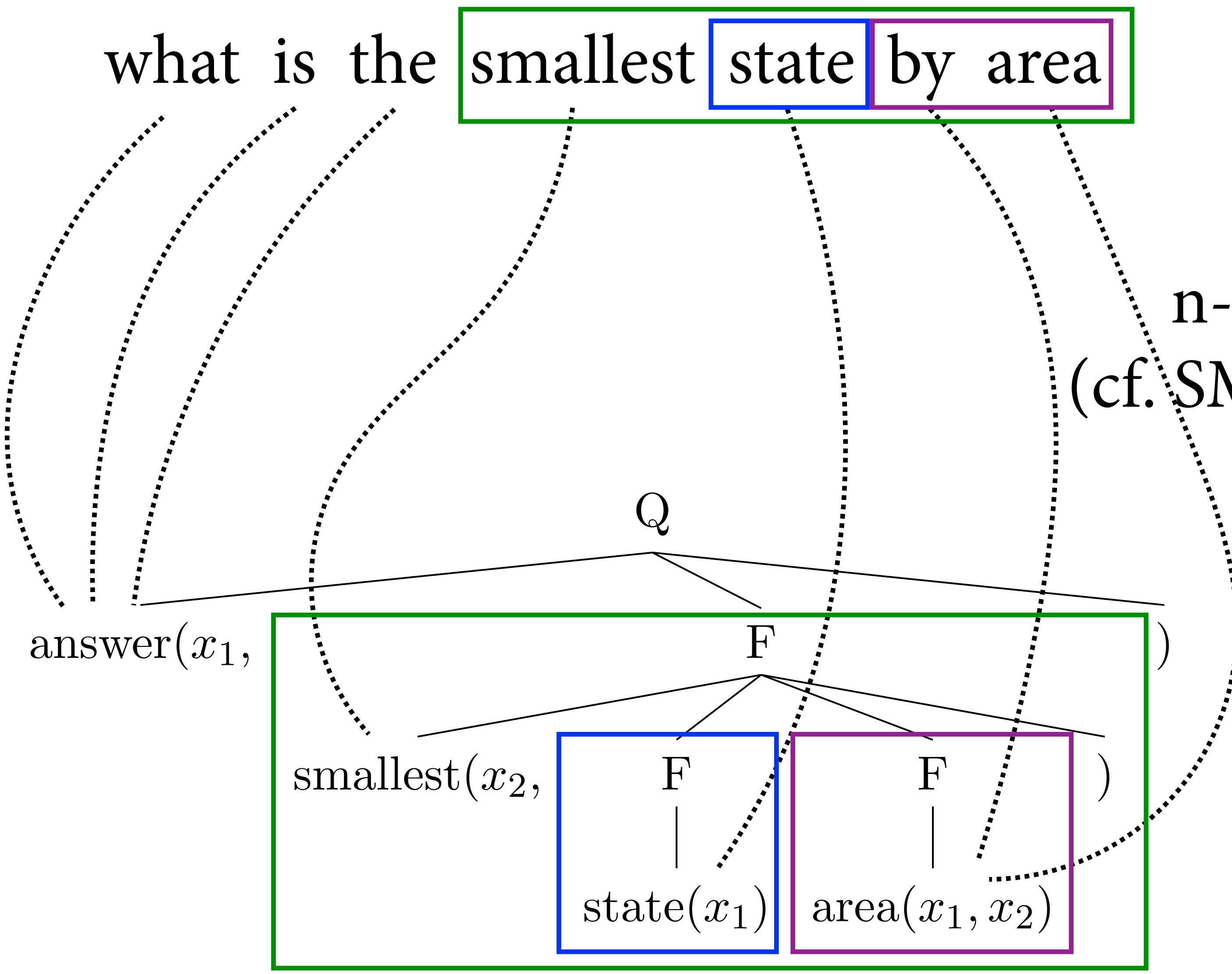
Wong & Mooney



n-to-1 alignment
(cf. SMT in a few weeks)

Assumptions:
- n-to-1 alignment between words and nodes
- unambiguous structure of meaning representation

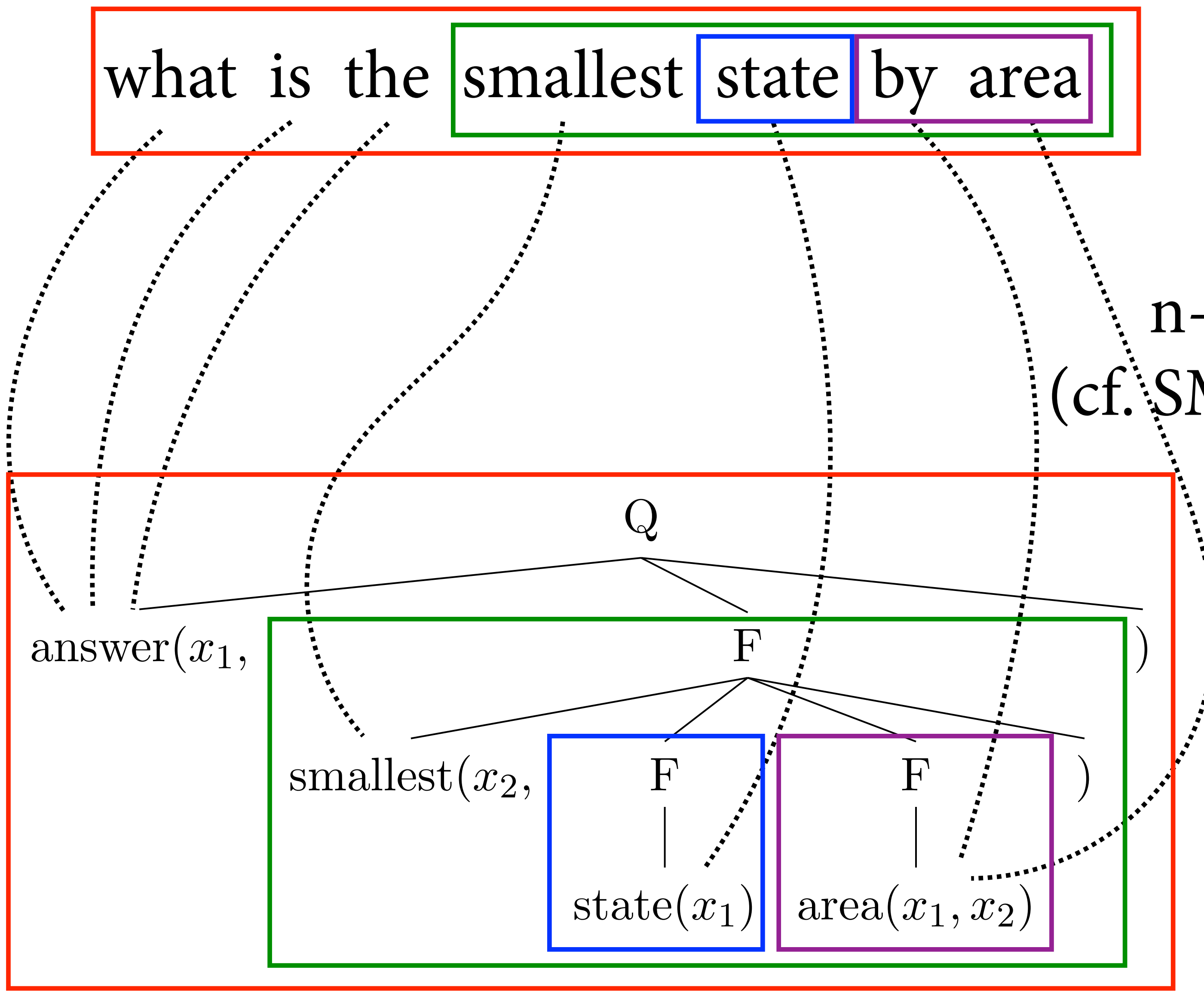
Wong & Mooney



n-to-1 alignment
(cf. SMT in a few weeks)

Assumptions:
- n-to-1 alignment between words and nodes
- unambiguous structure of meaning representation

Wong & Mooney



n-to-1 alignment
(cf. SMT in a few weeks)

Assumptions:
- n-to-1 alignment between words and nodes
- unambiguous structure of meaning representation

Wong & Mooney

- Extract synchronous grammar from corpus.
- Log-linear model on synchronous grammar with local features:
  - rule counts
  - parent annotations
- Parsing on string side easy because features are all local.

Combinatory categorial grammar


Steedman

$$
\begin{array}{ccc}
\text{John} & \text{eats} & \text{cookies} \\
\hline
\mathrm{NP} & (\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP} & \mathrm{NP} \\
\cline{2-3}
 & \multicolumn{2}{c}{\mathrm{S}\backslash\mathrm{NP}} \quad > \\
\hline
\multicolumn{3}{c}{\mathrm{S}} \quad <
\end{array}
$$

Leaves justified by *lexicon entries,* e.g. “John $\vdash$ NP”.
> and < are forward and backward *application*
Full CCG has more rules, which can make it non-context-free.

CCG with semantics


Steedman

John	eats	cookies
NP	(S\NP)/NP	NP
j'	*λxλy eat'(y)(x)*	*cookies'*

$\overline{}$ >

S\NP

λy eat'(y)(cookies')

$\overline{}$ <

S

eat'(j')(cookies')

Lexicon entries specify semantics: John ⊢ NP : *j'*

Forward and backward application also manipulate MRs

Learning semantic CCG grammars

- For every sentence w in training data with MR M, start with initial lexicon entry: w $\vdash$ S : *M*

- Iteratively find all ways to split lexicon entry using *higher-order unification*.

New York borders Vermont $\vdash$ S : *next_to(ny, vt)*

New York $\vdash$ NP : *ny*   borders Vermont $\vdash$ S \ NP : λx *next_to(x, vt)*

- Select best split and retrain model.

Some details

- HOU is really powerful and needs to be restricted to linguistically appropriate splits.

- Log-linear model with local syntactic and semantic features.

- Model makes fewer assumptions than Mooney:
  - use word alignments only to initialize feature weights
  - HOU more powerful in decomposing lambda terms

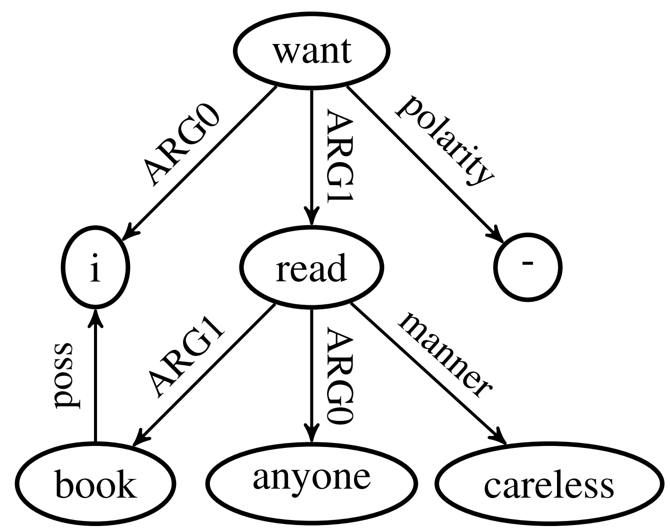
Evaluation results

System	Variable Free			Lambda Calculus		
	Rec.	Pre.	F1	Rec.	Pre.	F1
Cross Validation Results						
KRISP	71.7	**93.3**	81.1	–	–	–
WASP	74.8	87.2	80.5	–	–	–
Lu08	81.5	89.3	**85.2**	–	–	–
λ-WASP	–	–	–	86.6	92.0	89.2
Independent Test Set						
ZC05	–	–	–	79.3	**96.3**	87.0
ZC07	–	–	–	86.1	91.6	88.8
UBL	81.4	89.4	**85.2**	85.0	94.1	**89.3**
UBL-s	**84.3**	85.2	84.7	**87.9**	88.5	88.2

(on Geoquery 880 corpus)

Current Trends

- Brand-new large-scale corpora with semantic annotations: AMR-Bank with *graphs*.



- Becoming hot topic in CL right now. Lots of research on semantic parsing with graphs.

- Come to PM1 "Semantic Parsing" next semester!

Conclusion

- Inference with natural language requires computing (some sort of) formal meaning representations.
- Compositional semantics construction.
- Some methods for learning syn-sem interface:
  - using synchronous grammars
  - using CCG and higher-order unification