Parsing with CCG

Mark Steedman, Stephen Clark, & Julia Hockenmaier
Informatics, University of Edinburgh

CLSP Summer Workshop, August 2002

www.cogsci.ed.ac.uk/~steedman/papers.html

Introduction

- Combinatory Categorial Grammar (CCG, Steedman 2000b) is a “mildly context-sensitive” grammar formalism.
- That is, it is in a class that can plausibly be claimed to be just expressive enough to capture human language including phenomena like coordination and long range dependency.
- Both are frequent in corpora.
- Most treebank grammars fail to capture these phenomena entirely.
- Many context-free processing techniques generalize to the mildly context sensitive class.
- Making the generalization will give us: **more constrained** and **less under- and over-generalizing** parsers; **better models**; and **semantically interpretable** outputs.

Combinatory Categorial Grammar

(1) $$\begin{array}{lll} S & \rightarrow & NP \quad VP \\ VP & \rightarrow & TV \quad NP \\ TV & \rightarrow & \{proved,\ finds,\ \ldots\} \end{array}$$

(2) proved $:= (S\backslash NP)/NP$

(3) *The functional application rules*

a. $X/Y \quad Y \quad \Rightarrow \quad X$ $\qquad (>)$

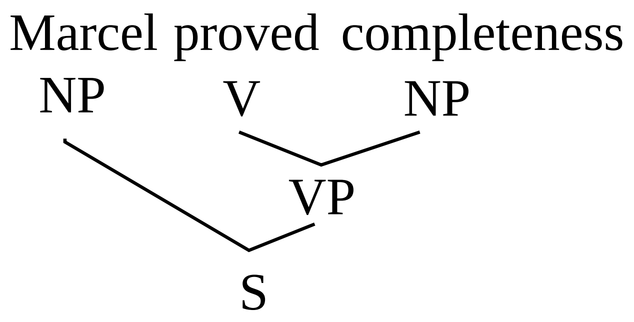
b. $Y \quad X\backslash Y \quad \Rightarrow \quad X$ $\qquad (<)$

A Derivation

(4) a.

$$\begin{array}{ccc}
\text{Marcel} & \text{proved} & \text{completeness} \\
\overline{NP} & \overline{(S\backslash NP)/NP} & \overline{NP} \\
 & \multicolumn{2}{c}{\overline{\hphantom{(S\backslash NP)/NP \quad NP}}{>}} \\
 & \multicolumn{2}{c}{S\backslash NP} \\
\multicolumn{3}{c}{\overline{\hphantom{NP \quad (S\backslash NP)/NP \quad NP}}{<}} \\
\multicolumn{3}{c}{S}
\end{array}$$

b.



Semantics

(5) proved := $(S\backslash NP_{3s})/NP : \lambda x.\lambda y.prove'xy$

(6) *Functional application*

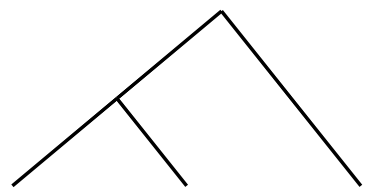
a. $X/Y : f \quad Y : a \quad \Rightarrow \quad X : fa \qquad (>)$

b. $Y : a \quad X\backslash Y : f \quad \Rightarrow \quad X : fa \qquad (<)$

(7)

$$
\begin{array}{ccc}
\text{Marcel} & \text{proved} & \text{completeness} \\
\hline
NP_{3sf} : marcel' & (S\backslash NP_{3s})/NP : \lambda x.\lambda y.prove'xy & NP : completeness' \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP_{3s} : \lambda y.prove'completeness'y \quad >} \\
\hline
\multicolumn{3}{c}{S : prove'completeness'marcel' \quad <}
\end{array}
$$

Notation: Left-Associativity Convention over Logical Forms



(8) a. $(\mathit{prove}'\,\mathit{completeness}')\,\mathit{marcel}'$ b. *prove' completeness' marcel'*

- A nonordered form of the traditional VP is reflected at the level of propositional logical form.
- Such logical forms therefore preserve traditional c-command.

Coordination

(9) *Simplified coordination rule* $(<\Phi>)$

$$X \quad CONJ \quad X' \Rightarrow \quad X''$$

(10) Marcel conjectured and proved completeness

$$\begin{array}{ccccc} \text{Marcel} & \text{conjectured} & \text{and} & \text{proved} & \text{completeness} \\ \hline NP & (S\backslash NP)/NP & CONJ & (S\backslash NP)/NP & NP \end{array}$$

$$(S\backslash NP)/NP \ \ (S\backslash NP)/NP \ \ CONJ \ \ (S\backslash NP)/NP \Rightarrow_{<\Phi>} (S\backslash NP)/NP$$

$$(S\backslash NP)/NP \ \ NP \Rightarrow_{>} S\backslash NP$$

$$NP \ \ S\backslash NP \Rightarrow_{<} S$$

Composition

(11) *Forward composition* ($>\mathbf{B}$)

$$X/Y : f \quad Y/Z : g \quad \Rightarrow_{\mathbf{B}} \quad X/Z : \lambda x.f(gx)$$

(12)

$$\begin{array}{cccccc}
\text{Marcel} & \text{conjectured} & \text{and} & \text{might} & \text{prove} & \text{completeness}\\
\hline
NP & (S\backslash NP)/NP & CONJ & (S\backslash NP)/VP & VP/NP & NP \\
: marcel' & : conjecture' & : and' & : might' & : prove' & : completeness'
\end{array}$$

$$\cfrac{(S\backslash NP)/VP : might' \qquad VP/NP : prove'}{(S\backslash NP)/NP : \lambda x.\lambda y.might'(prove'x)y} {>}\mathbf{B}$$

$$\cfrac{(S\backslash NP)/NP : conjecture' \qquad CONJ : and' \qquad (S\backslash NP)/NP : \lambda x.\lambda y.might'(prove'x)y}{(S\backslash NP)/NP : \lambda x.\lambda y.and'(might'(prove'x)y)(conjecture'xy)} {<}\Phi{>}$$

$$\cfrac{(S\backslash NP)/NP : \lambda x.\lambda y.and'(might'(prove'x)y)(conjecture'xy) \qquad NP : completeness'}{S\backslash NP : \lambda y.and'(might'(prove'completeness')y)(conjecture'completeness'y)} {>}$$

$$\cfrac{NP : marcel' \qquad S\backslash NP : \lambda y.and'(might'(prove'completeness')y)(conjecture'completeness'y)}{S : and'(might'(prove'completeness')marcel')(conjecture'completeness'marcel')} {<}$$

Type-Raising

(13) *Forward type-raising* ($>\mathbf{T}$)

$$NP : a \quad \Rightarrow_{\mathbf{T}} \quad \mathrm{T}/(\mathrm{T}\backslash NP) : \lambda f.fa$$

(14)

Marcel	proved	and	I	disproved	completeness
NP	$(S\backslash NP)/NP$	$CONJ$	NP	$(S\backslash NP)/NP$	NP
$S/(S\backslash NP)$ $>\mathbf{T}$			$S/(S\backslash NP)$ $>\mathbf{T}$		
S/NP $>\mathbf{B}$			S/NP $>\mathbf{B}$		
S/NP $<\Phi>$					
		S $>$			

- Type raising is restricted to primitive argument categories, NP, PP etc., and over primitive functors like verbs, resembling the traditional notion of *case*.

Many Linguistic Predictions—For Example:

- The following construction is predicted on arguments of symmetry.

(15) give a teacher an apple and a policeman a flower

$$
\begin{array}{cccccc}
\text{give} & \text{a teacher} & \text{an apple} & \text{and} & \text{a policeman} & \text{a flower} \\
\overline{DTV} & \overline{TV\backslash DTV}^{<\mathbf{T}} & \overline{VP\backslash TV}^{<\mathbf{T}} & \overline{CONJ} & \overline{TV\backslash DTV}^{<\mathbf{T}} & \overline{VP\backslash TV}^{<\mathbf{T}} \\
 & \multicolumn{2}{c}{\overline{\quad VP\backslash DTV \quad}^{<\mathbf{B}}} & & \multicolumn{2}{c}{\overline{\quad VP\backslash DTV \quad}^{<\mathbf{B}}} \\
 & \multicolumn{5}{c}{\overline{\qquad\qquad VP\backslash DTV \qquad\qquad}^{<\Phi>}} \\
\multicolumn{6}{c}{\overline{\qquad\qquad\qquad VP \qquad\qquad\qquad}^{<}}
\end{array}
$$

- $DTV = (VP/NP)/NP \quad TV = VP/NP$

- In accord with observations of Ross (1970) concerning the relation of "verb gapping" and word order, CCG examples like the following cannot occur in an SVO language like English:

(16) *A policeman a flower and give a teacher an apple.

Similar Argument Clusters are Prefixes In Dutch/German

(17) dat Jan Cecilia Henk de nijlpaarden zag helpen voeren

$$
\begin{array}{lllll}
\text{Jan} & \text{Cecilia} & \text{Henk} & \text{de nijlpaarden} & \text{zag helpen voeren} \\
\hline
\mathrm{T}/(\mathrm{T}\backslash NP_1) & \mathrm{T}/(\mathrm{T}\backslash NP_2) & \mathrm{T}/(\mathrm{T}\backslash NP_3) & \mathrm{T}/(\mathrm{T}\backslash NP_4) & (((S_{+SUB}\backslash NP_1)\backslash NP_2)\backslash NP_3)\backslash NP_4
\end{array}
$$

$$\mathrm{T}/(\mathrm{T}\backslash NP_1) \quad \mathrm{T}/(\mathrm{T}\backslash NP_2) \Rightarrow_{>\mathbf{B}} \mathrm{T}/((\mathrm{T}\backslash NP_1)\backslash NP_2)$$

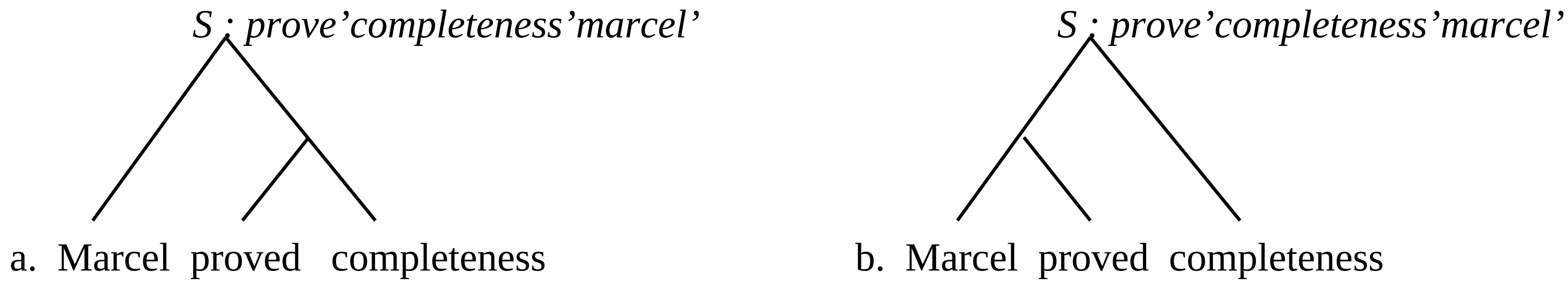
$$\Rightarrow_{>\mathbf{B}} \mathrm{T}/(((\mathrm{T}\backslash NP_1)\backslash NP_2)\backslash NP_3)$$

$$\Rightarrow_{>\mathbf{B}} \mathrm{T}/((((\mathrm{T}\backslash NP_1)\backslash NP_2)\backslash NP_3)\backslash NP_4)$$

$$\Rightarrow_{>} S_{+SUB}$$

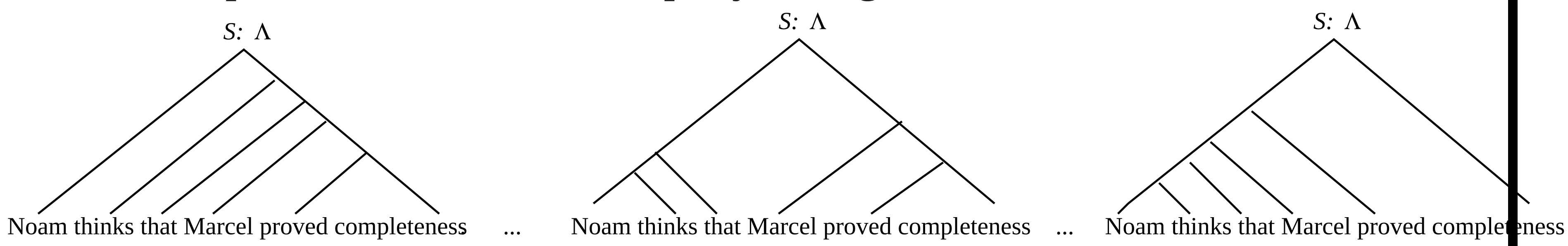
On So-called “Spurious” Ambiguity

- Examples like (14), (15), and (17) embody the claim that fragments like “Marcel proved”, “a policeman a flower”, and “Jan Cecilia Henk de nijlpaarden” are *constituents*, comparable to “proved completeness”.
- If “Marcel proved” can be constituent in right node raising, then it can be a constituent of a canonical transitive sentence.
- Even such simple sentences are *derivationally ambiguous*:



On So-called “Spurious” Ambiguity (Contd.)

- More complex sentences are multiplely ambiguous:



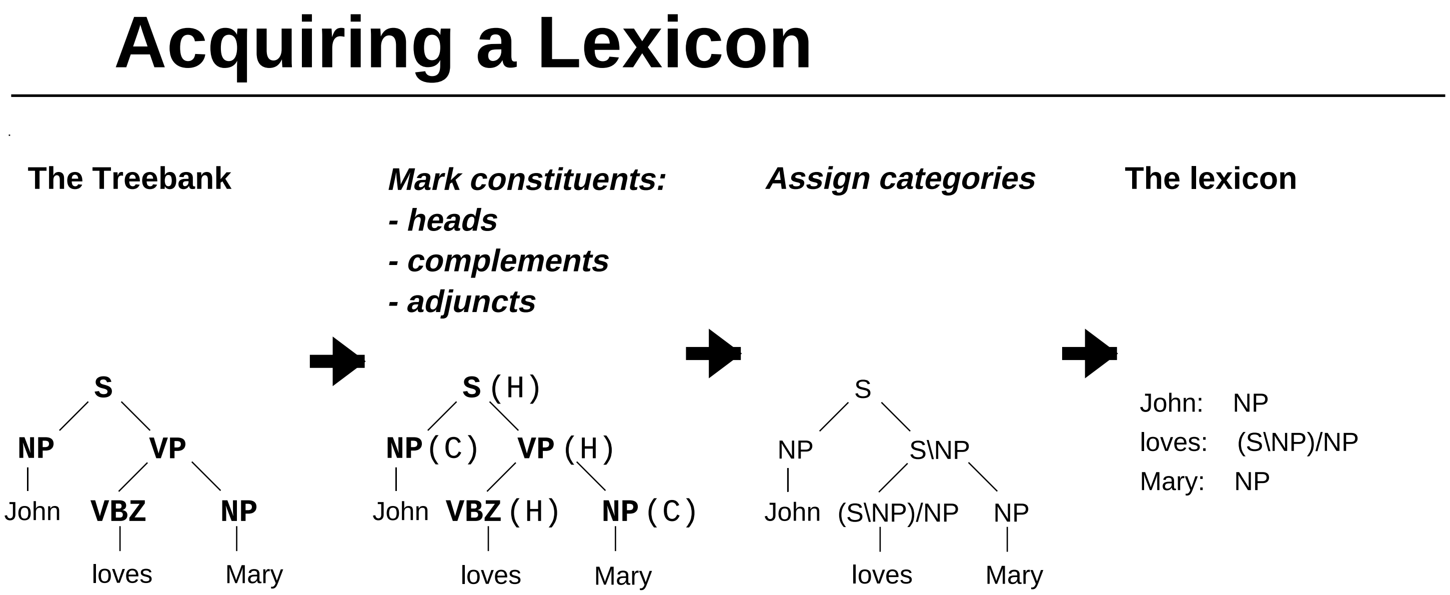
- This has been referred to (misleadingly) as “Spurious” ambiguity, since all the derivations have the same interpretation Λ.
- Interestingly, so called “spurious” constituents include most **left prefixes**.

How to Parse with a Grammar that Has "Spurious Ambiguity"

- **All** grammars exhibit derivational ambiguity—even CFG.
- **Any** grammar that captures coordination at all will have the **same** derivational ambiguity as CCG.
- Use standard table-driven parsing methods such as CKY, checking identity of **underlying** representation of table entries, rather than derivation: Karttunen (1989); Komagata (1997, 1999); Hockenmaier *et al.* (2002).
- Vijay-Shanker and Weir 1994 show how to make this worst-case polynomial, although for realistic grammars exponential parsers seem to be average-case cubic (see Komagata 1999 for English and Japanese).

How to Induce a CCG

- Extract a CCG lexicon from the Penn Treebank: Hockenmaier and Steedman (2002a)



- **This trades lexical types (1200 against 48) for rules (3000 instantiations of combinatory rules against 12000) with standard Treebank grammars.**

How to Statistically Optimize the Parser

- Use standard Maximum Entropy techniques to train a FSM “supertagger” Clark (2002), **multitagging at over 98% accuracy**
- Then **either**:
  - Parse using the Komagata technique, building and modeling *deep* or semantic dependency structures: Clark *et al.* (2002).
  - **or** use Normal-form parsing (Wittenburg and Wall (1991)), building and modeling Collins-style *surface* dependencies : Hockenmaier and Steedman (2002b).

A Problem

- Standard generative models for the local dependencies characteristic of CFGs do not immediately generalize to the **reentrant dependencies** generated by these more expressive grammars (Abney 1997—see Johnson lecture in this series).
- The model of Clark et al. 2002 is, technically, unsound. The generative model of Hockenmaier et al. only models local dependencies.
- Log Linear models offer one (rather desperate) kind of solution, but have known disadvantages for e.g. language modeling.
- We conjecture that a sound full generative model is as possible for mildly context sensitive grammars as it is for CFG (Hockenmaier, in preparation).

Performance on Overall Dependency Recovery

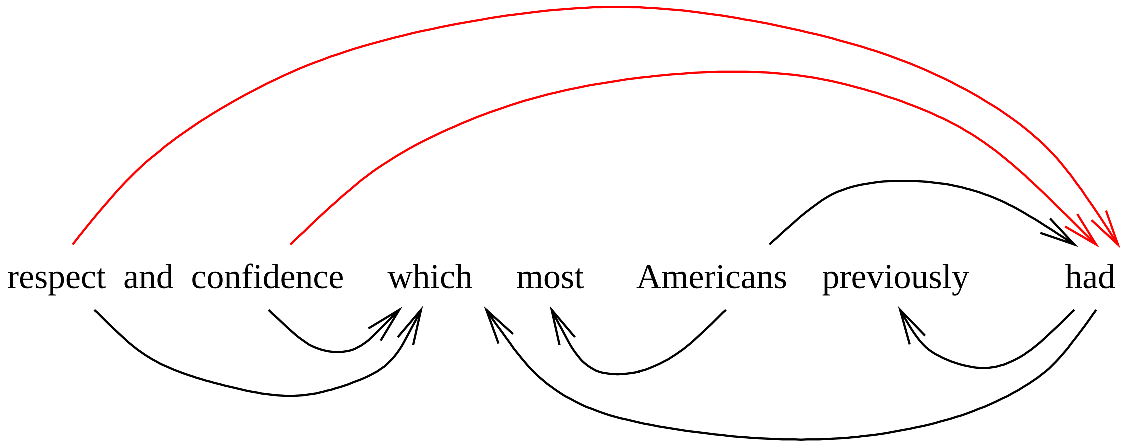
- Hockenmaier and Steedman (2002b)

Model	LexCat	Parseval LP	LR	BP	BR	Surface dependencies $\langle PHS \rangle$	$\langle\rangle$
Baseline	87.7	72.8	72.4	78.3	77.9	81.1	84.3
HWDep	92.0	81.6	81.9	85.5	85.9	84.0	90.1

- Collins (1999) reports 90.9% for “surface” dependencies.
- **CCG benefits greatly from word-word dependencies**.
  (in contrast to Gildea (2001)’s observations for Collins’ Model 1)
- Compare on Clark *et al.* (2002)’s (different) “deep” dependencies:

	LP	LR	UP	UR
Clark et al. ’02	81.9%	81.8%	89.1%	90.1%
Hockenmaier	83.7%	84.2%	90.5%	91.1%

Deep Dependencies: Clark *et al.* (2002)



lexical_item	category	slot	head_of_arg
which	$(NP_X\backslash NP_{X,1})/(S[dcl]_2/\ cf NP_X)$	2	*had*
which	$(NP_X\backslash NP_{X,1})/(S[dcl]_2/\ cf NP_X)$	1	*confidence*
which	$(NP_X\backslash NP_{X,1})/(S[dcl]_2/NP_X)$	1	*respect*
had	$(S[dcl]_{had}\backslash NP_1)/NP_2)$	2	*confidence*
had	$(S[dcl]_{had}\backslash NP_1)/NP_2)$	2	*respect*

Performance on Full Object Relatives: Clark *et al.* (2002)

- 24 cases of extracted objects in Section 00 associated with object relative pronoun category $(NP_X\backslash NP_X)/(S[dcl]/NP_X)$
- 10 (41.7%) recovered with all dependencies correct
  - so both noun attachment and rel_pronoun-verb dependency correct
  - 10 incorrect because wrong category assigned to relative pronoun
    - complementizer *that* has a high prior probability
  - 3 incorrect only because relative clause attached to the wrong noun
  - 1 incorrect only because wrong category assigned to predicate
- Much better performance can be expected with a better model.
- Other varieties of deep dependency discussed in Clark *et al.* (2002).

Hockenmaier and Steedman (2002b)

- **Extraction:**
  - Lexical cat. for **subject relative pronoun**
    $(\mathbf{NP}\backslash\mathbf{NP})/(\mathbf{S}[\mathbf{dcl}]\backslash\mathbf{NP})$: 97.1%P, 94.3%R
  - Lexical cat. for **embedded subject extraction** (Steedman '96)
    $((\mathbf{S}[\mathbf{dcl}]\backslash\mathbf{NP})/\mathbf{NP})/(\mathbf{S}[\mathbf{dcl}]\backslash\mathbf{NP})$: 100.0%P, 83.3%R
  - Lexical cat. for **object relative pronoun**
    $(\mathbf{NP}\backslash\mathbf{NP})/(\mathbf{S}[\mathbf{dcl}]/\mathbf{NP})$: 84.2%P, 76.2%R
- **Coordination:**
  - VP coordination (coordination of $\mathbf{S}[.]\backslash\mathbf{NP}$): 67.3%P, 67.0%R
  - Right-node-raising (coordination of $(\mathbf{S}[.]\backslash\mathbf{NP})/\mathbf{NP}$): 73.1%P, 79.2%R

CCG Parsers as Language Models

- Standard technique/baseline is Trigram modeling.
- Strict left-to-right parsing interpolated with trigram model does better: Chelba and Jelinek (1998); Roark (2001).
- Immediate-Head parser modeling alone does even better, even with a non-left-to-right algorithm: Charniak (2001).
- CCG type-raising treats head and complement as **dual**: In some sense, it makes **all** constructions head first.
- Hence most left prefixes are constituents, even in Dutch/German/Japanese.
- While any grammar can in principle be mapped onto a prefix grammar and a corresponding generative model, CCG makes this trivial: the grammar *is* a prefix grammar and probabilities for prefix dependency structures can be derived from the standard dependency model.

CCG Parsers as Language Models

- For example, in Dutch the prefix *dat Cecilia een hond een knok* ... ("that Cecilia a dog a bone ...") has a category $S/(((S\backslash NP)\backslash NP)\backslash NP)$.
- The type of this constituents tells you how to *invert* the dependency model to obtain a left-to-right prediction.
- It predicts a ditransitive verbgroup and tells you all you need to know to estimate its Arg Max from verbs of that class. (For example, the "give" stem is going to come out ahead of the "sell" stem.)
- *dat een hond een knok Cecilia* ... is going to make a quite different predictions.
- So are some of the alternative derivations of *dat Cecilia een hond een knok* ...
- CCG similarly offers a direct way to use prosodic information (Steedman 2000a).

The Need for More Training Data

- Apart from weaknesses in statistical models that can be fixed, the chief limitation on the CCG parsers (and any parser with a comparably large category set and/or constrained grammar) is **known words which have not been seen with the necessary category**.
- This is a problem of the amount of training data: a million words is not enough for effective supervised learning for **any** grammar (Gildea 2001).
- **Either:**
  - Generalize the lexicon by clustering, or sources like WordNet.
  - Used semi supervised techniques to generate labelled dat automatically

Weakly Supervised Training for Statistical Models

- Charniak showed that running his parser over 30M words of unlabelled data and retraining slightly improved performance.
- We also know voting among parsers improves performance Hendriks (1998): 92.09% LP / 90.1% LR.
- Co-training (Blum and Mitchell 1998): bootstrapping from small labeled corpus using multiple “views” from different parsers with different models and levels of confidence: Sarkar (2001)
- This is rather like what children do when their partial knowledge of the language tells them that a new word must be a verb, even though they don’t yet know the associated concept.
- We are pursuing this for a number of grammars/parsers at the CSLP Summer Workshop.

Moral

- You can have your linguistic cake and eat it—with a an automatically induced lexicon and a statistical model

References

Abney, Steven, 1997. "Stochastic Attribute-Value Grammars." *Computational Linguistics* 23:597–618.

Blum, Avram and Mitchell, John, 1998. "Combining Labeled and Unlabeled Data with Co-training." In *COLT: Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann.

Charniak, Eugene, 2001. "Immediate-Head Parsing for Language Models." In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse*. San Francisco, California: Morgan Kaufmann Publishers, 116–123.

Chelba, Ciprian and Jelinek, Frederick, 1998. "Exploiting Syntactic Structure for Language Modeling." In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*. San Francisco, California: Morgan Kaufmann Publishers, 225–231.

Clark, Stephen, 2002. "A Supertagger for Combinatory Categorial Grammar." In *Proceedings of the TAG+ Workshop*. Venice, 19–24.

Clark, Stephen, Hockenmaier, Julia, and Steedman, Mark, 2002. "Building Deep Dependency Structures with a Wide-Coverage CCG Parser." In *Proceedings of the 40th Meeting of the ACL (to appear)*. Philadelphia, PA, 327–334.

Collins, Michael, 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Gildea, Dan, 2001. "Corpus Variation and Parser Performance." In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*. Pittsburgh, PA, 167–202.

Hendriks, H., 1998. "Links without Locations." In F. Hamm and E. Zimmermann (eds.), *Linguistische Berichte, Sonderheft 9/1998: SEMANTIK*, Opladen: Westdeutscher Verlag.

Hockenmaier, Julia, Bierner, Gann, and Baldridge, Jason, 2002. "Extending the Coverage of a CCG System." *Journal of Logic and Computation* (forthcoming).

Hockenmaier, Julia and Steedman, Mark, 2002a. “Acquiring Compact Lexicalized Grammars from a Cleaner Treebank.” In *Proceedings of the Third International Conference on Language Resources and Evaluation (to appear)*. Las Palmas, Spain.

Hockenmaier, Julia and Steedman, Mark, 2002b. “Generative Models for Statistical Parsing with Combinatory Categorial Grammar.” In *Proceedings of the 40th Meeting of the ACL (to appear)*. Philadelphia, PA, 335–342.

Karttunen, Lauri, 1989. “Radical Lexicalism.” In Mark Baltin and Anthony Kroch (eds.), *Alternative Conceptions of Phrase Structure*, Chicago: University of Chicago Press.

Komagata, Nobo, 1997. “Efficient Parsing for CCGs with Generalized Type-Raised Categories.” In *Proceedings of the 5th International Workshop on Parsing Technologies, Boston MA*. ACL/SIGPARSE, 135–146.

Komagata, Nobo, 1999. *Information Structure in Texts: A Computational Analysis of Contextual Appropriateness in English and Japanese*. Ph.D. thesis, University of Pennsylvania.

Roark, Brian, 2001. “Probabilistic top-down parsing and language modeling.” *Computational Linguistics* 27:249–276.

Ross, John Robert, 1970. “Gapping and the Order of Constituents.” In Manfred Bierwisch and Karl Heidolph (eds.), *Progress in Linguistics*, The Hague: Mouton. 249–259.

Sarkar, Anoop, 2001. “Applying Co-Training Methods to Statistical Parsing.” In *Proceedings of NAACL 2001. Pittsburgh, PA, June*. Morgan Kaufmann.

Steedman, Mark, 2000a. “Information Structure and the Syntax-Phonology Interface.” *Linguistic Inquiry* 34:649–689.

Steedman, Mark, 2000b. *The Syntactic Process*. Cambridge, MA: MIT Press.

Vijay-Shanker, K. and Weir, David, 1994. “The Equivalence of Four Extensions of Context-Free Grammar.” *Mathematical Systems Theory* 27:511–546.

Wittenburg, Kent and Wall, Robert, 1991. “Parsing with Categorial Grammar in Predictive Normal Form.” In Masaru Tomita (ed.), *Current Issues in Parsing Technology*, Dordrecht: Kluwer. 65–83. Revised selected papers from International Workshop on Parsing Technology (IWPT) 1989, Carnegie Mellon University.