Combinatory Categorial Grammars for Robust Natural Language Processing

Mark Steedman

With Jason Baldridge, Çem Boszahin. Ruken Çakıcı, Stephen Clark, James Curran, Julia Hockenmaier, Tom Kwiatkowski, Emily Thomforde, Prachya Boonkwan *and many others*

EACL Tutorial **Athens** **April 2009**

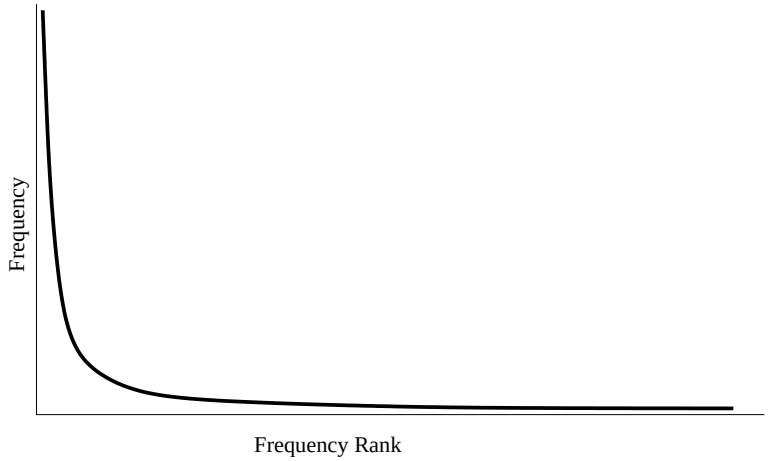
Prospectus

- I: Prologue: Why use CCG for NLP?
- II: Combinatory Categorial Grammar
- III: Wide Coverage Parsing with CCG
- IV: Work in Progress
- V Interim Conclusion and Future Directions
- APPENDIX: The Statistical Problem of Child Language Development

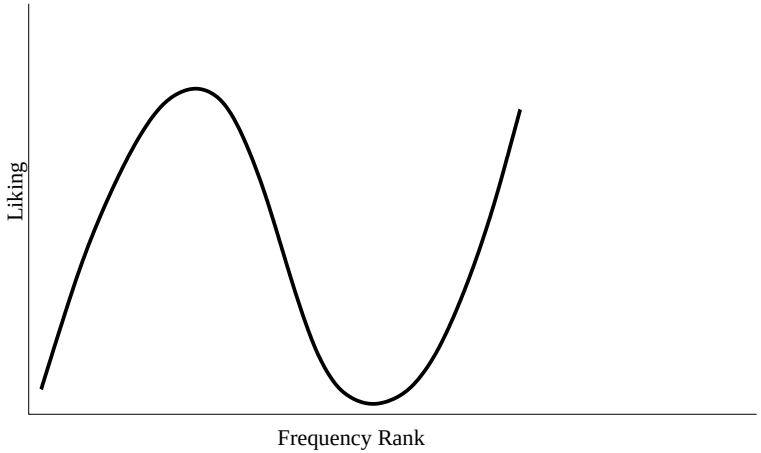
I: Prologue: Why Use CCG for NLP?

Prologue: The Long Tail and the Uncanny Valley

- Zipf's Law:



- The Uncanny Valley:



- Ignoring the long tail can engender the uncanny:

In the Uncanny Valley

- TREC 2005:

Q77.6 Name opponents who Foreman defeated.

Q77.7 Name opponents who defeated Foreman.

- A QA Program (Kor 2005):

Opponents who Foreman defeated:
George Foreman
Joe Frazier
Ken Norton
Sonny
Archie Moore

Opponents who defeated Foreman:
George Foreman
Joe Frazier
Ken Norton
Sonny
Archie Moore

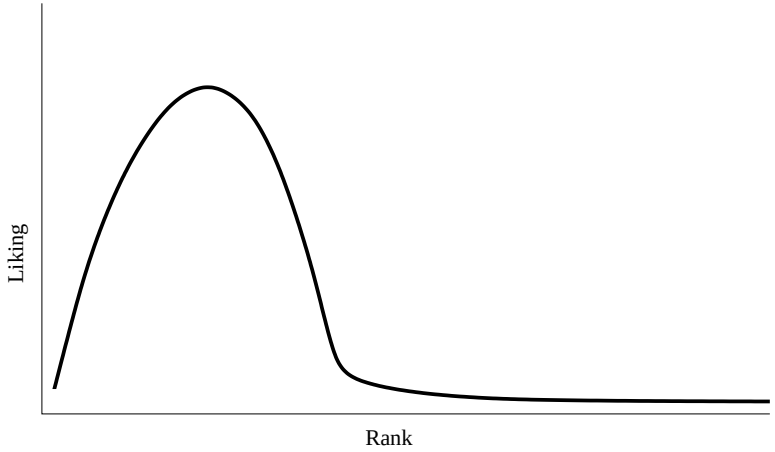
The Problem

- The contribution of certain constructions to determining system acceptability is disproportionate to their low frequency.
- This is bad news.

Machine learning is very bad at acquiring systems for which important information is in rare events.

The Darkling Plain

If the distribution of event types really is a power law curve, then there is no other side to the Uncanny Valley.



- We shall see that, for certain categories of parser error, up to half the error rate is due to unseen grammatical event types (such as lexical entries), and up to half is due to unseen model tokens for seen types (such as head word dependencies).
- So the long tail is already hurting us badly.
- What to do?

What To Do

- The distribution of grammatical event types isn't a true power law, because there is a finite number of them, defined generatively, ultimately by a universal semantics.
- In principle, we can enumerate the types.

But there are more constructions than you can shake a stick at (Goldberg 1995)

- Induce them from labeled data. (Or get linguists to enumerate them).
- If we knew what that semantics was, we might be able to solve the model problem as well.

But none of the existing logicist semantic formalisms will do (MacCartney and Manning 2007).

How To Do It

- We need a readily extensible, construction-based grammar.
- It must be robustly and efficiently parsable with wide coverage
- It must be transparent to a “natural” semantics, supporting cheap inference.

II: Combinatory Categorial Grammar

Categorial Grammar

- Categorial Grammar replaces PS rules by lexical categories and general combinatory rules (**Lexicalization**):

(1) $S \rightarrow NP \quad VP$

$VP \rightarrow TV \quad NP$

$TV \rightarrow \{proved, finds, \ldots\}$

- Categories:

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

(3) think := $(S\backslash NP)/_{\diamond}S$

Categorial Grammar

- **Categorial Grammar** replaces PS rules by lexical categories and general combinatory rules (**Lexicalization**):

(1) $S \rightarrow NP \quad VP$

$VP \rightarrow TV \quad NP$

$TV \rightarrow \{proved, finds, \ldots\}$

- **Categories** with semantic intepretations:

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

(3) think := $(S\backslash NP)/_{\diamond}S : think'$

Applicative Derivation

- **Functional Application**

$$\frac{X/_{\star}Y \qquad Y}{X}\ > \qquad \frac{Y \qquad X\backslash_{\star}Y}{X}\ <$$

- (4)

$$\dfrac{\dfrac{\text{Marcel}}{NP} \qquad \dfrac{\dfrac{\text{proved}}{(S\backslash NP)/NP} \qquad \dfrac{\text{completeness}}{NP}}{S\backslash NP}\,>}{S}\,<$$

(5)

$$\dfrac{\dfrac{\text{I}}{NP} \qquad \dfrac{\dfrac{\text{think}}{(S\backslash NP)/_{\diamond}S} \qquad \dfrac{\dfrac{\text{Marcel}}{NP} \qquad \dfrac{\dfrac{\text{proved}}{(S\backslash NP)/NP} \qquad \dfrac{\text{completeness}}{NP}}{S\backslash NP}\,>}{S}\,<}{S\backslash NP}\,>}{S}\,<$$

Applicative Derivation

- **Functional Application** with semantic intepretations:

$$\frac{X/_{\star}Y : f \quad Y : g}{X : f(g)}\ > \qquad \frac{Y : g \quad X\backslash_{\star}Y : f}{X : f(g)}\ <$$

- (4)

$$\dfrac{\dfrac{\text{Marcel}}{NP : marcel'} \quad \dfrac{\dfrac{\text{proved}}{(S\backslash NP)/NP : prove'} \quad \dfrac{\text{completeness}}{NP : completeness'}}{S\backslash NP : \lambda y.prove'\,completeness'\,y}>}{S : prove'\,completeness'\,marcel'}<$$

(5)

$$\dfrac{\dfrac{\text{I}}{NP : i'} \quad \dfrac{\dfrac{\text{think}}{(S\backslash NP)/_{\diamond}S : think'} \quad \dfrac{\dfrac{\text{Marcel}}{NP : marcel'} \quad \dfrac{\dfrac{\text{proved}}{(S\backslash NP)/NP : prove'} \quad \dfrac{\text{completeness}}{NP : completeness'}}{S\backslash NP : \lambda y.prove'\,completeness'\,y}>}{S : prove'\,completeness'\,marcel'}<}{S\backslash NP : think'\,(prove'\,completeness'\,marcel')}>}{S : think'\,(prove'\,completeness'\,marcel')\,i'}<$$

Combinatory Categorial Grammar (CCG)

- **Combinatory Rules**:

$$\frac{X/_{\star}Y \qquad Y}{X} > \qquad \frac{Y \qquad X\backslash_{\star}Y}{X} <$$

$$\frac{X/_{\diamond}Y \qquad Y/_{\diamond}Z}{X/_{\diamond}Z} >\mathbf{B} \qquad \frac{Y\backslash_{\diamond}Z \qquad X\backslash_{\diamond}Y}{X\backslash_{\diamond}Z} <\mathbf{B}$$

$$\frac{X/_{\times}Y \qquad Y\backslash_{\times}Z}{X\backslash_{\times}Z} >\mathbf{B}_{\times} \qquad \frac{Y/_{\times}Z \qquad X\backslash_{\times}Y}{X/_{\times}Z} <\mathbf{B}_{\times}$$

- All arguments are type-raised via the lexicon:

$$\frac{X}{\mathrm{T}/(\mathrm{T}\backslash X)} >\mathbf{T} \qquad \frac{X}{\mathrm{T}\backslash(\mathrm{T}/X)} <\mathbf{T}$$

Combinatory Categorial Grammar (CCG)

- **Combinatory Rules** with semantic intepretations:

$$\frac{X/_{\star}Y : f \quad Y : g}{X : f(g)} > \qquad \frac{Y : g \quad X\backslash_{\star}Y : f}{X : f(g)} <$$

$$\frac{X/_{\diamond}Y : f \quad Y/_{\diamond}Z : g}{X/_{\diamond}Z : \lambda z.f(g(z))} >\mathbf{B} \qquad \frac{Y\backslash_{\diamond}Z : g \quad X\backslash_{\diamond}Y : f}{X\backslash_{\diamond}Z : \lambda z.f(g(z))} <\mathbf{B}$$

$$\frac{X/_{\times}Y : f \quad Y\backslash_{\times}Z : g}{X\backslash_{\times}Z : \lambda z.f(g(z))} >\mathbf{B}_{\times} \qquad \frac{Y/_{\times}Z : g \quad X\backslash_{\times}Y : f}{X/_{\times}Z : \lambda z.f(g(z))} <\mathbf{B}_{\times}$$

- All arguments are type-raised via the lexicon:

$$\frac{X : x}{\mathrm{T}/(\mathrm{T}\backslash X) : \lambda f.f(x)} >\mathbf{T} \qquad \frac{X : x}{\mathrm{T}\backslash(\mathrm{T}/X) : \lambda f.f(x)} <\mathbf{T}$$

- We omit a further family of rules based on the combinator **S**

Slash Typing

- The features $\star, \diamond, \times$ were introduced by Baldridge 2002 following Hepple (1987)
- They form a lattice

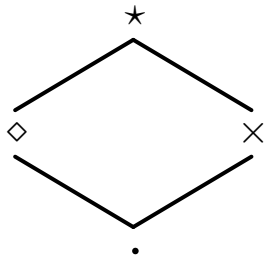
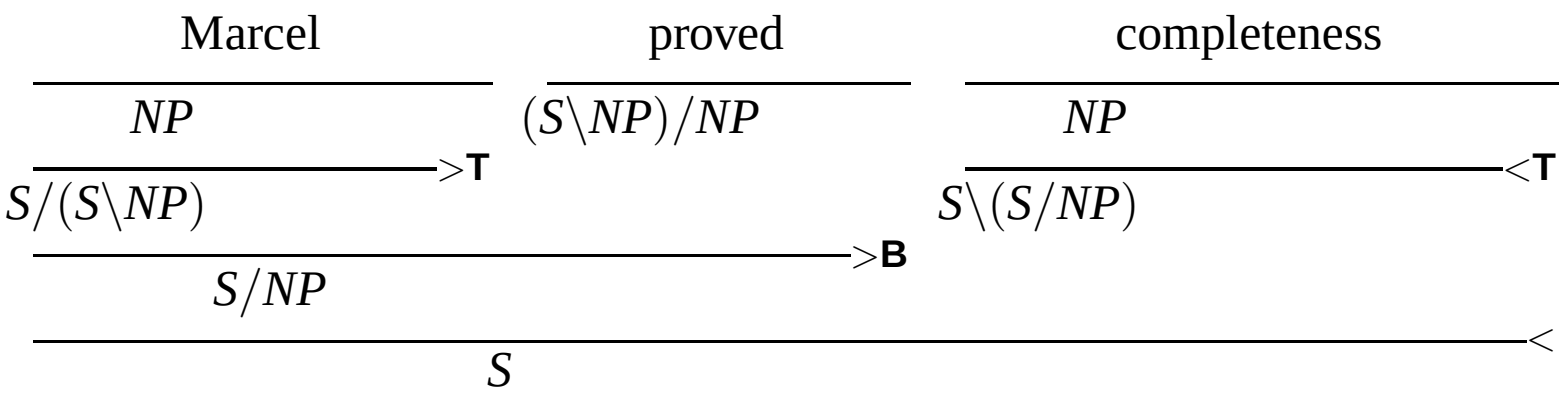


Figure 1: CCG type hierarchy for slash features (Baldridge and Kruijff 2003).

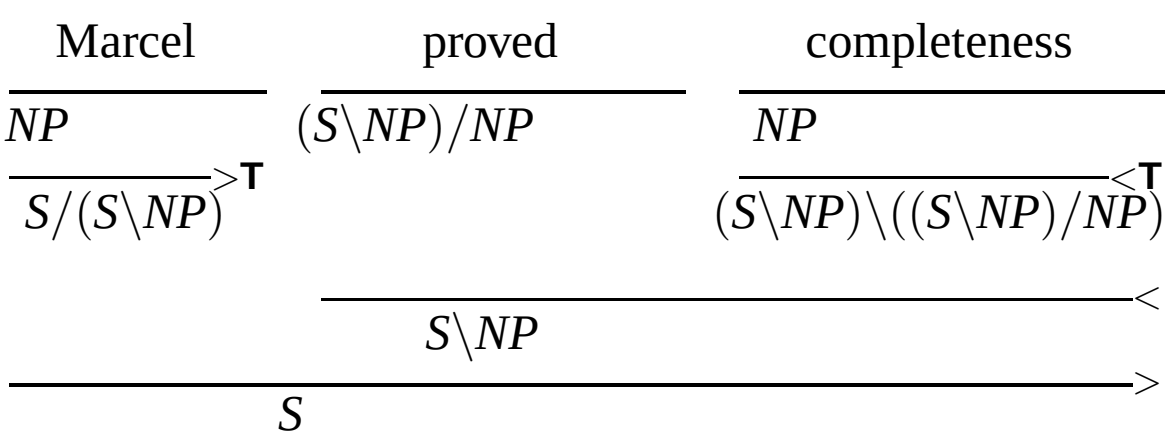
- $\cdot$ type written as bare slash e.g. α/β means any rule can apply
- $\diamond$ type e.g. $\alpha/_{\diamond}\beta$ means any rule except $\times$ can apply.
- $\times$ type e.g. $\alpha/_{\times}\beta$ means any rule except $\diamond$ can apply.
- $\star$ type e.g. $\alpha/_{\star}\beta$ means no rule except $\star$ can apply.

Combinatory Derivation

(6)



(7)



Combinatory Derivation

(6)

$$\dfrac{\dfrac{\dfrac{\text{Marcel}}{NP : marcel'}}{S/(S\backslash NP) : \lambda f.f\ marcel'}{>}\mathbf{T} \quad \dfrac{\dfrac{\text{proved}}{(S\backslash NP)/NP : prove'}}{S/NP : \lambda x.prove'\ x\ marcel'}{>}\mathbf{B} \quad \dfrac{\dfrac{\text{completeness}}{NP : completeness'}}{S\backslash(S/NP) : \lambda p.p\ completeness'}{<}\mathbf{T}}{S : prove'\ completeness'\ marcel'}{<}$$

(7)

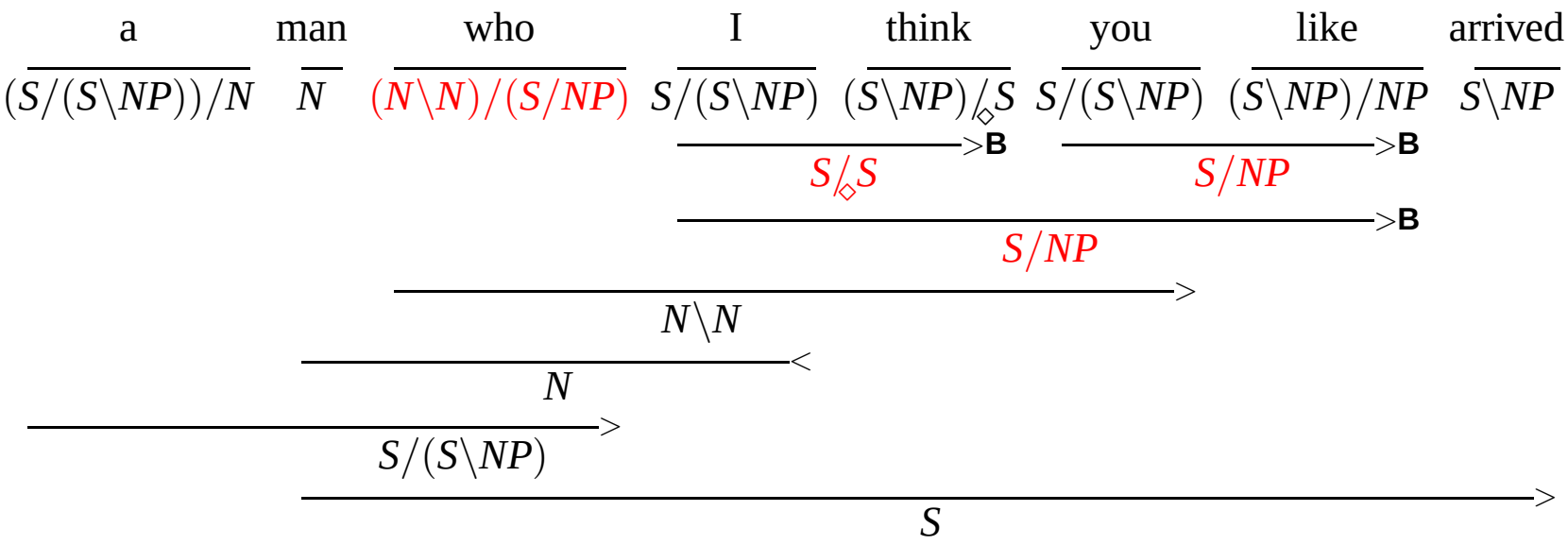
$$\dfrac{\dfrac{\dfrac{\text{Marcel}}{NP : marcel'}}{S/(S\backslash NP) : \lambda f.f\ marcel'}{>}\mathbf{T} \quad \dfrac{\dfrac{\dfrac{\text{proved}}{(S\backslash NP)/NP : prove'} \quad \dfrac{\dfrac{\text{completeness}}{NP : completeness'}}{(S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.p\ completeness'}{<}\mathbf{T}}{S\backslash NP : \lambda y.prove'\ completeness'\ y}{<}}{S : prove'\ completeness'\ marcel'}{>}$$

- Type-raising is simply grammatical *case*, as in Latin/Japanese.
- We need to schematize $T/(T\backslash NP)$, $T\backslash(T/NP)$

Linguistic Predictions: Unbounded "Movement"

- The combination of type-raising and composition allows derivation to project lexical function-argument relations onto "unbounded" constructions such as relative clauses and coordinate structures, without transformational rules:

(8)



Predictions: English Intonation

- A minimal pair of contexts and contours:

(9) Q: I know who proved soundness. But who proved COMPLETENESS?

A: (MarCEL) (proved COMPLETENESS).

H*L L+H* LH%

(10) Q: I know which result Marcel PREDICTED. But which result did Marcel PROVE?

A: (MARcel PROVED) (COMPLETENESS).

L+H* LH% H* LL%

- Crossing contexts and responses yields complete incoherence.

Prosodic Phrases $\subset$ CCG constituents.

Predictions: Argument-Cluster Coordination

- The following construction is predicted on arguments of symmetry.

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{DTV}\quad\dfrac{\dfrac{}{TV\backslash DTV}<\mathbf{T}\quad\dfrac{}{VP\backslash TV}<\mathbf{T}}{VP\backslash DTV}<\mathbf{B}\quad\dfrac{}{(\mathrm{X}\backslash_\star \mathrm{X})/_\star \mathrm{X}}\quad\dfrac{\dfrac{}{TV\backslash DTV}<\mathbf{T}\quad\dfrac{}{VP\backslash TV}<\mathbf{T}}{VP\backslash DTV}<\mathbf{B}}{(VP\backslash DTV)\backslash_\star(VP\backslash DTV)}>}{VP\backslash DTV}<}{VP}<$$

—**where** $VP = S\backslash NP$; $TV = (S\backslash NP)/NP$; $DTV = ((S\backslash NP)/NP)/NP$, and X is a variable over any category up to some low bounded valency.

- A variant like the following cannot occur in an SVO language like English:

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

Syntax = Type-Raising and Composition

- CCGs combination of type-raising and composition yields a “mildly context sensitive” permuting and rebracketing calculus closely tuned to the needs of natural grammar.
- The argument cluster coordination construction (11) is an example of a universal tendency for “deletion under coordination” to respect basic word order: in all languages, if arguments are on the left of the verb then argument clusters coordinate on the left, if arguments are to the right of the verb then argument clusters coordinate to the right of the verb (Ross 1970):

(13) SVO: *SO and SVO SVO and SO

VSO: *SO and VSO VSO and SO

SOV: SO and SOV *SOV and SO

These Things are Out There in the Treebank

- Full Object Relatives (570 in WSJ treebank)
- Reduced Object Relatives (1070 in WSJ treebank)
- Argument Cluster Coordination (230 in WSJ treebank):

```
(S (NP-SBJ It)
   (VP (MD could)
       (VP (VP (\myRed{VB} cost)
               (\myRed{NP-1} taxpayers)
               (\myRed{NP-2} $ 15 million))
           (CC and)
           (VP (\myRed{NP=1} BPC residents)
               (\myRed{NP=2}$ 1 million)))))
```

- It could cost taxpayers 15 million and __ BPC residents 1 million

These Things are Out There (contd.)

- Parasitic Gaps (at least 6 in WSJ treebank):

```
(S (NP-SBJ Hong Kong's uneasy relationship with China)
   (VP (MD will)
       (VP (VP (VB constrain)
               (NP (\myRed{-NONE- *RNR*-1})))
           (PRN (: --)
                (IN though)
                (VP (RB not)
                    (VB inhibit)
                    (NP (\myRed{-NONE- *RNR*-1})))
                (: --))
           (\myRed{NP-1} long-term economic growth))))
```

- Hong Kong's uneasy relationship with China will constrain __, though not inhibit __, long-term growth.

CCG is "Nearly Context-Free"

- CCG and TAG are provably weakly equivalent to Linear Indexed Grammar (LIG) Vijay-Shanker and Weir (1994).
- Hence they are not merely "Mildly Context Sensitive" (Joshi 1988), but rather "Nearly Context Free," or "Type 1.$\dot{9}$" in the Extended Chomsky Hierarchy.

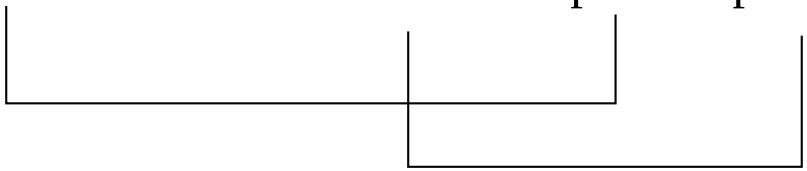
Language Type	**Automaton**	**Rule-types**	**Exemplar**
Type 0: RE	Universal Turing Machine	$\alpha \rightarrow \beta$	
Type 1: CS	Linear Bound Automaton (LBA)	$\phi A \psi \rightarrow \phi \alpha \psi$	$\mathcal{P}(a^n b^n c^n)$ (?)
I	Nested Stack Automaton(NSA)	$A_{[(i),\ldots]} \rightarrow \phi B_{[(i),\ldots]} \psi C_{[(i),\ldots]} \xi$	a^{2^n}
LCFRS (MCS)	ith-order NPDA	$A_{[[(i),\ldots]\ldots]} \rightarrow \phi B_{[[(i),\ldots]\ldots]} \psi$	$a^n b^n c^n \ldots m^n$
"Type 1.$\dot{9}$": LI	Nested PDA (NPDA)	$A_{[(i),\ldots]} \rightarrow \phi B_{[(i),\ldots]} \psi$	$a^n b^n c^n$
Type 2: CF	Push-Down Automaton (PDA)	$A \rightarrow \alpha$	$a^n b^n$
Type 3: FS	Finite-state Automaton (FSA)	$A \rightarrow \begin{cases} \text{a B} \\ \text{a} \end{cases}$	a^n

A Trans-Context Free Natural Language

- CCG can capture unboundedly crossed dependencies in Dutch and Zurich German (examples from Shieber 1985):

... das mer em Hans es huus haelfed aastriiche

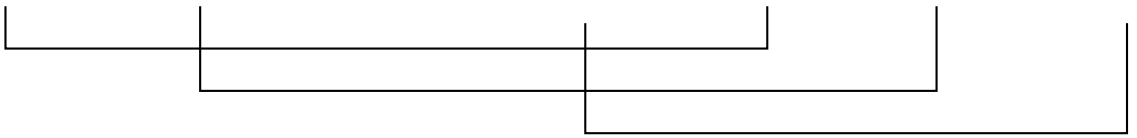
... that we.NOM Hans.DAT the house.ACC helped paint



'... that we helped Hans paint the house.'

... das mer d'chind em Hans es huus loend haelfe aastriiche

... that we.NOM the children.ACC Hans.DAT the house.ACC let help paint



'... that we let the children help Hans paint the house.'

A Trans-Context Free Natural Language

(14)

das	mer	em Hans	es huus	hälfed	aastriiche
that	we−NOM	Hans−DAT	the house−ACC	helped	paint
	$NP^{\uparrow}_{nom}$	$NP^{\uparrow}_{dat}$	$NP^{\uparrow}_{acc}$	$((S_{+SUB}\backslash NP_{nom})\backslash NP_{dat})/_{\times} VP$	$VP\backslash NP_{acc}$
				$((S_{+SUB}\backslash NP_{nom})\backslash NP_{dat})\backslash NP_{acc}$ (>$\mathbf{B}_{\times}$)	
				$(S_{+SUB}\backslash NP_{nom})\backslash NP_{dat}$ (>)	
			$S_{+SUB}\backslash NP_{nom}$ (>)		
		S_{+SUB} (>)			

"that we helped Hans paint the house"

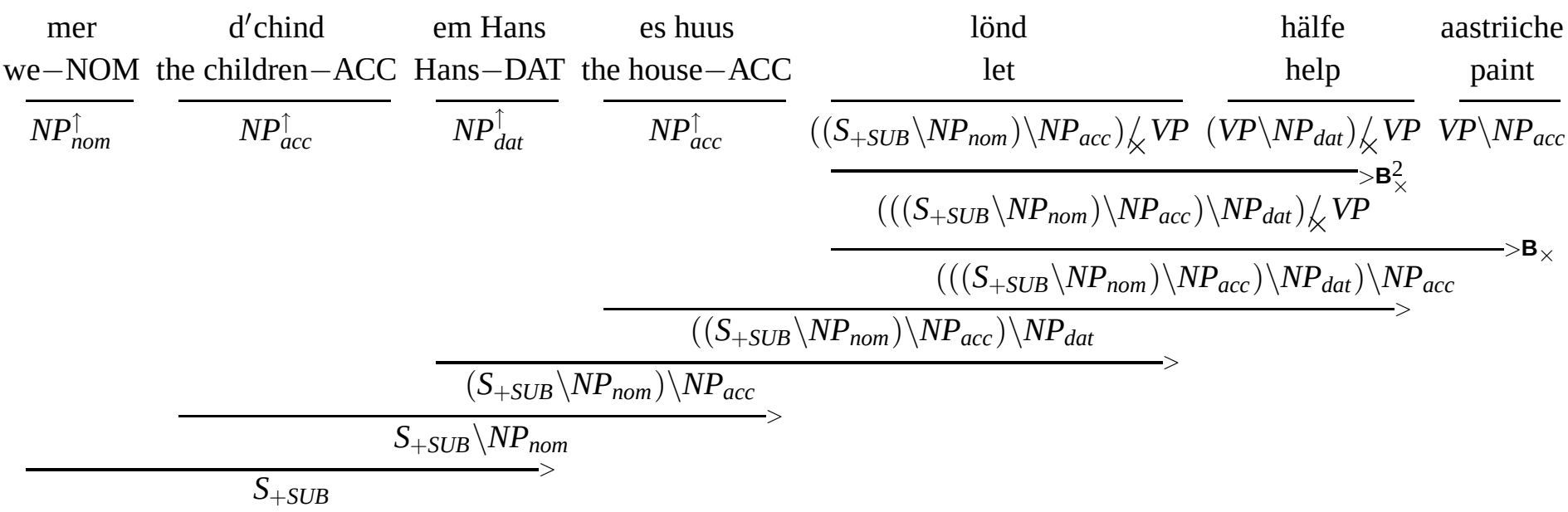
- The following is correctly also allowed:

(15) Das mer em Hans hälfed es huus aastriiche.


The corresponding word order is *disallowed* in the related Dutch construction.

A Trans-Context Free Natural Language

(15)

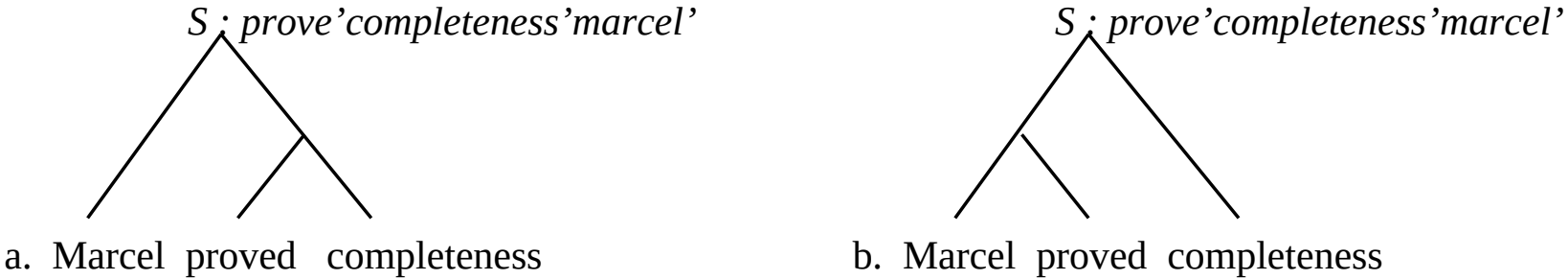


"that we let the children help Hans paint the house"

- Again, other word orders are correctly allowed.

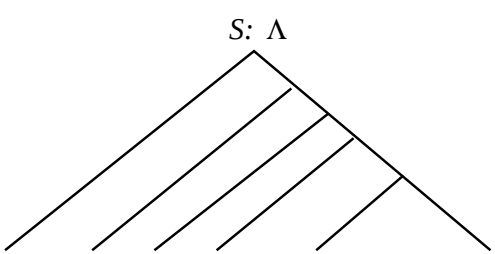
On So-called “Spurious” Ambiguity

- Examples like (10), and (11), embody the claim that fragments like “Marcel proved”, and “a policeman a flower”, 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*:

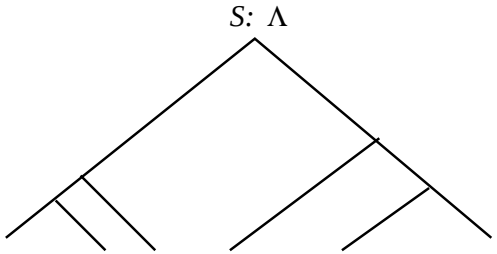


On So-called “Spurious” Ambiguity (Contd.)

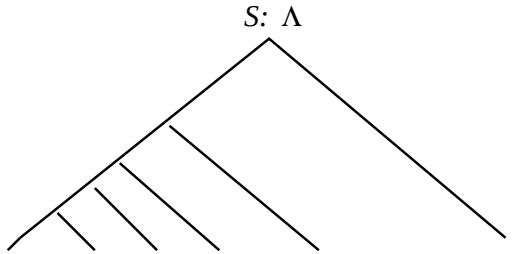
- More complex sentences are multiplely ambiguous:



a. Frankie thinks that Anna married Manny.



b. Frankie thinks that Anna married Manny.



c. Frankie thinks that Anna married Manny.

- 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**.

Parsing in the Face of "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, with packed charts, either:
  - checking identity of **underlying** representation of table entries (Steedman 2000), rather than identity of derivation, or:
  - parsing normal-form derivations (Eisner 1996)

CCG is Nearly Context-Free (contd.)

- It has polynomial parsing complexity (Vijay-Shanker and Weir 1990)
- Hence it has nice “Divide and Conquer” algorithms, like CKY, and Dynamic Programming.
- For real-life sized examples like parsing the newspaper, such algorithms must be statistically optimized.

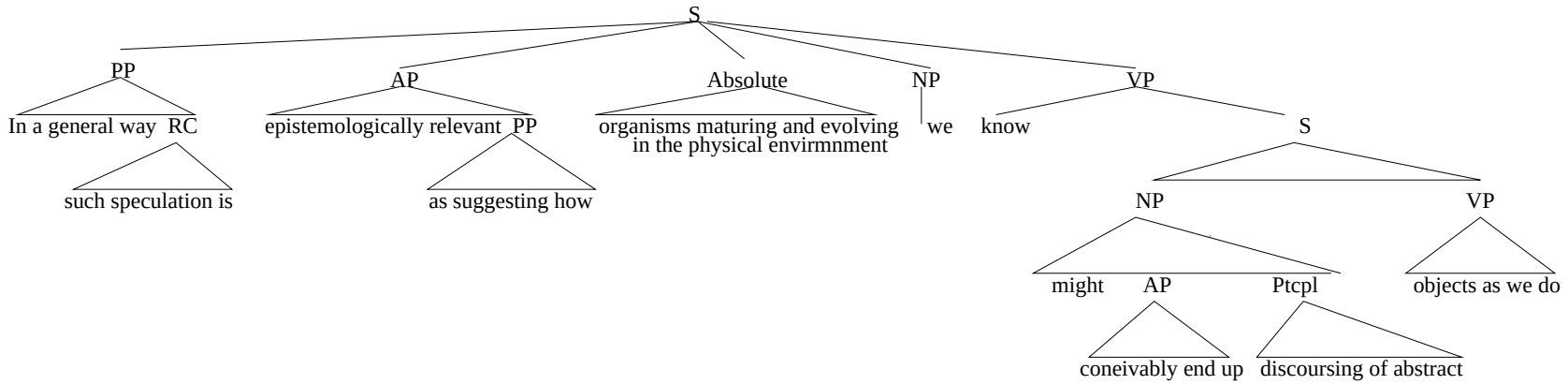
III: Wide-Coverage Parsing with CCG

Human and Computational NLP

- No handwritten grammar ever has the coverage that is needed to read the daily newspaper.
- Language is syntactically highly ambiguous and it is hard to pick the best parse. Quite ordinary sentences of the kind you read every day routinely turn out to have hundreds and on occasion thousands of parses, albeit mostly semantically wildly implausible ones.
- High ambiguity and long sentences break exhaustive parsers.

For Example:

- "In a general way such speculation is epistemologically relevant, as suggesting how organisms maturing and evolving in the physical environment we know might conceivably end up discoursing of abstract objects as we do." (Quine 1960:123).
- —yields the following (from Abney 1996), among many other horrors:



The Anatomy of a Parser

- Every parser can be identified by three elements:
  - A Grammar (Regular, Context Free, Linear Indexed, etc.) and an associated automaton (Finite state, Push-Down, Nested Push-Down, etc.);
  - A search Algorithm characterized as left-to-right (etc.), bottom-up (etc.), and the associated working memories (etc.);
  - An Oracle, to resolve ambiguity.
- The oracle can be used in two ways, either to actively limit the search space, or in the case of an “all paths” parser, to rank the results.
- In wide coverage parsing, we mostly have to use it in the former way.

Competence and Performance

- Linguists (Chomsky 1957, *passim*), have always insisted on the methodological independence of “Competence” (the grammar that linguists study) and “Performance” (the mechanisms of language use).
- This makes sense: there are many more parsers than there are grammars.
- Nevertheless, Competence and Performance must have evolved as a single package, for what evolutionary edge does a parser without a grammar have, or a grammar without a parser?

Any theory that does not allow a one-to-one relation between the grammatical and derivational constituency has some explaining to do.

Human Sentence Processing

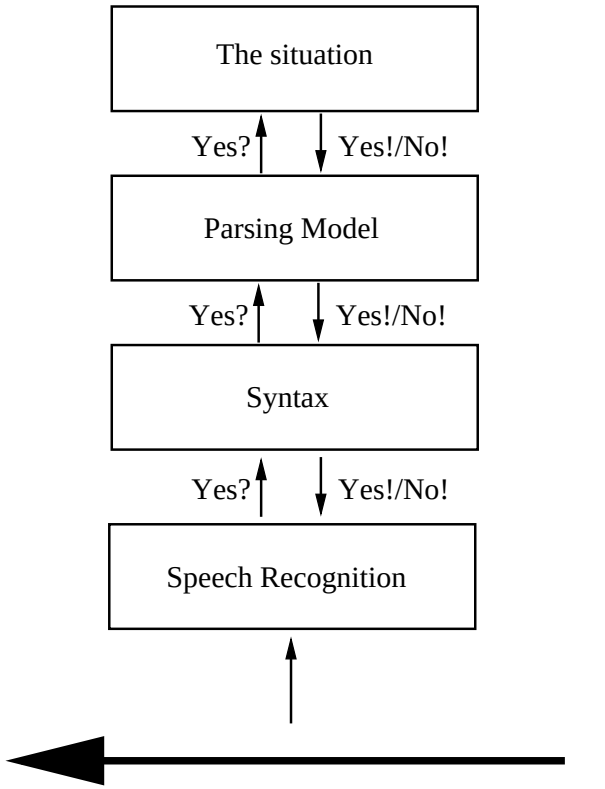
- “Garden path” sentences are sentences which are grammatical, but which naive subjects fail to parse.
- Example (16a) is a garden path sentence, because the ambiguous word “sent” is analysed as a tensed verb:

(16) a. # The doctor sent for the patient died.
b. The flowers sent for the patient died.

- However (16b) is not a garden path.
- So garden path effects are sensitive to world knowledge (Bever 1970).
- They are even sensitive to referential context: (Altmann and Steedman 1988) showed that (simplifying somewhat) if a context is established with two doctors, one of whom was sent for a patient, then the garden path effect is reversed.

The Architecture of the Human Sentence Processor

- This requires a “cascade” architecture:



Grammar and Incrementality

- Most left prefix substrings of sentences are typable constituents in CCG, for which alternative analyses can be compared using the parsing model
- The fact that (17a,b) involve the nonstandard constituent [The doctor sent for]$_{S/NP}$, means that constituent is also available for (17c,d)

(17) a. The patient that [the doctor sent for]$_{S/NP}$ died.

b. [The doctor sent for]$_{S/NP}$ and [The nurse attended]$_{S/NP}$ the patient who had complained of a pain.

c. #[The doctor sent for] $\left\{ \begin{array}{l} S/NP \\ (S/(S\backslash NP))/N \quad N \quad (N\backslash N)/NP \end{array} \right\}$ [the patient]$_{NP}$ died$_{S\backslash NP}$.

d. [The flowers sent for] $\left\{ \begin{array}{l} \#S/NP \\ (S/(S\backslash NP))/N \quad N \quad (N\backslash N)/NP \end{array} \right\}$ [the patient]$_{NP}$ died$_{S\backslash NP}$.

- (18) a. #[The doctor sent for the patient] $_{S}$ died$_{S\backslash NP}$.

b. [The flowers sent for the patient died$_{S}$.

The Strict Competence Hypothesis

- Since the spurious constitutent [#The flowers sent for]$_{S/NP}$ is available in the chart, so that its low probability in comparison with the probabilities of the unreduced components can be detected (according to some "figure of merit" (Charniak *et al.* 1998) discounting the future), the garden path in (16b) is avoided, even under the following very strong assumption about the parser:
  - The Strict Competence Hypothesis: the parser only builds structures that are licensed by the Competence Grammar as typable *constituents*.
- This is an attractive hypothesis, because it allows the Competence Grammar and the Performance Parser/Generator to evolve as a package deal, with parsing completely transparent to grammar, as in standard bottom-up algorithms.
- But is such a simple parser possible? We need to look at some real-life parsing programs.

Wide Coverage Parsing: the State of the Art

- Early attempts to model parse probability by attaching probabilities to rules of CFG performed poorly.
- Great progress as measured by the ParsEval measure has been made by combining statistical models of headword dependencies with CF grammar-based parsing (Collins 1997; Charniak 2000; McCloskey *et al.* 2006)
- However, the ParsEval measure is very forgiving. Such parsers have until now been based on highly overgenerating context-free covering grammars. Analyses depart in important respects from interpretable structures.
- In particular, they fail to represent the long-range "deep" semantic dependencies that are involved in relative and coordinate constructions, as in *A company$_i$ that$_i$ the Wall Street Journal says expects$_i$ to have revenue of $10M*, and *You can buy$_i$ and sell$_i$ all items$_i$ and services$_i$ on this easy to use site*.

Head-dependencies as Oracle

- Head-dependency-Based Statistical Parser Optimization works because it approximates an oracle using real-world knowledge.
- In fact, the knowledge- and context- based psychological oracle may be much more like a probabilistic relational model augmented with associative epistemological tools such as typologies and thesauri and associated with a dynamic context model than like traditional logicist semantics and inferential systems.
- Many context-free processing techniques generalize to the “mildly context sensitive” grammars.
- The “nearly context free” grammars such as LTAG and CCG—the least expressive generalization of CFG known—have been treated by Xia (1999), Hockenmaier and Steedman (2002a), and Clark and Curran (2004).

Nearly Context-Free Grammar

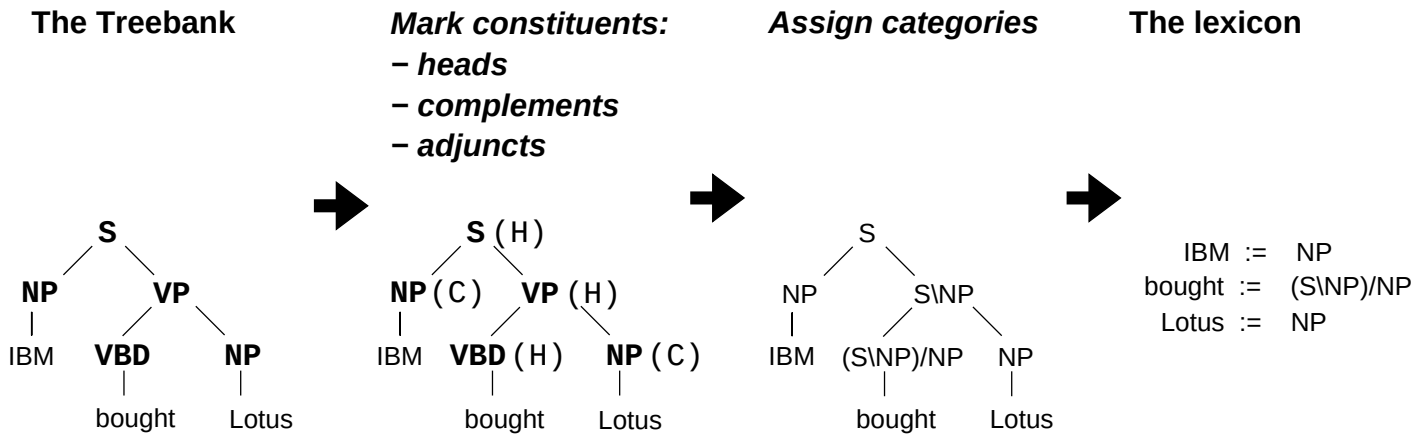
- Such Grammars capture the deep dependencies associated with coordination and long range dependency.
- Both phenomena are frequent in corpora, and are explicitly annotated in the Penn WSJ corpus.
- Standard treebank grammars ignore this information and fail to capture these phenomena entirely.

Zipf's law says using it won't give us much better overall numbers. (aropund 3% of sentences in WSJ include long-range object dependencies, but LRoDs are only a small proportion of the dependencies in those sentences.)

- But there is a big difference between getting a perfect eval-b score on a sentence including an object relative clause and interpreting it!

Supervised CCG Induction by Machine

- Extract a CCG lexicon from the Penn Treebank: Hockenmaier and Steedman (2002a), Hockenmaier (2003) (cf. Buszkowski and Penn 1990; Xia 1999).



- This trades lexical types (500 against 48) for rules (around 3000 instantiated binary combinatory rule types against around 12000 PS rule types) with standard Treebank grammars.

The trees in the CCG-bank are CCG derivations, and in cases like Argument Cluster Coordination and Relativisation they depart radically from Penn Treebank structures.

Supervised CCG Induction: Full Algorithm

- ```
  foreach tree T:
  preprocessTree(T);
  preprocessArgumentCluster(T);
  determineConstituentType(T);
  makeBinary(T);
  percolateTraces(T);
  assignCategories(T);
  treatArgumentClusters(T);
  cutTracesAndUnaryRules(T);
  ```

- The resulting treebank is somewhat cleaner and more consistent, and is offered for use in inducing grammars in other expressive formalisms. It was released in June 2005 by the Linguistic Data Consortium with documentation and can be searched using t-grep.
  ```

Statistical Models for Wide-Coverage Parsers

- There are two kinds of statistical models:
  - Generative models directly represent the probabilities of the rules of the grammar, such as the probability of the word *eat* being transitive, or of it taking a nounphrase headed by the word *integer* as object.
  - Discriminative models compute probability for whole parses as a function of the product of a number of weighted features, like a Perceptron. These features typically include those of generative models, but can be anything.
- Both have been applied to CCG parsing

Generative Models (Hockenmaier)

- 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).
- The generative model of Hockenmaier and Steedman 2002b only models probability for Collins-style local dependencies (although it can *recover* long range dependencies).
- It uses “Normal-form modeling”, where the derivations modeled are those in which type-raising and composition are only used when there is no alternative.
- Hockenmaier (2003) showed that a sound full generative model is as possible for mildly context sensitive grammars as it is for CFG.
- Log Linear models offer another solution (Clark and Curran 2003, 2004, and see below)

Hockenmaier 2002/2003: Overall Dependency Recovery

- Hockenmaier and Steedman (2002b)

		Parseval				Surface dependencies	
Model	LexCat	LP	LR	BP	BR	$\langle PHS \rangle$	$\langle\rangle$
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 unlabeled $\langle\rangle$ "surface" dependencies.
- **CCG benefits greatly from word-word dependencies**.
  (in contrast to Gildea (2001)'s observations for Collins' Model 1)
- This parser is available on the project webpage.

Recovery of Long Range Dependencies

Hockenmaier (2003)

- **Extraction:**
  - Dependencies involving **subject relative pronoun** (**NP****NP**)/(**S**[**dcl**]**NP**): 98.5%LP, 95.4%LR (99.6%UP, 98.2%UR)
  - Lexical cat. for **embedded subject extraction** (Steedman '96) ((**S**[**dcl**]**NP**)/**NP**)/(**S**[**dcl**]**NP**): 100.0%P, 83.3%R
  - Dependencies involving **object relative pronoun (including ES)** (**NP****NP**)/(**S**[**dcl**]/**NP**): 66.7%LP, 58.3%LR (76.2%UP, 58.3%UR)
- **Coordination:**
  - VP coordination (coordination of **S**[.]**NP**): 67.3%P, 67.0%R
  - Right-node-raising (coordination of (**S**[.]**NP**)/**NP**): 73.1%P, 79.2%R
- A direct comparison with Johnson (2002) postprocessing method is not immediately possible.

Log-Linear Conditional CCG Parsing Models

- Features f_i encode evidence indicating good/bad parses
- (19) $p(d|S) = \frac{1}{Z(S)} e^{\Sigma_i \lambda_i f_i(d,S)}$
- Use standard Maximum Entropy techniques to train a FSM "supertagger" Clark (2002) to assign CCG categories, **multitagging ($n \approx 3$) at over 98% accuracy** (Clark and Curran 2003, 2004).
- Clark and Curran use a conditional log-linear model such as Maximum Entropy of **either**:
  - The derived structure or parse yield;
  - All derivations;
  - All derivations with Eisner Normal Form constraints.

Conditional CCG Parsing Models (Contd.)

- Discriminative estimation via the limited-memory BFGS algorithm is used to set feature weights
- Estimation is computationally expensive, particularly for “all derivations”:
  - Beowulf cluster allows complete Penn Treebank to be used for estimation.
  - The fact that the supertagger is very accurate makes this possible.

Overall Dependency Recovery

	LP	LR	UP	UR	cat
Clark et al. 2002	81.9	81.8	90.1	89.9	90.3
Hockenmaier 2003	84.3	84.6	91.8	92.2	92.2
Clark and Curran 2004	**86.6**	**86.3**	**92.5**	**92.1**	**93.6**
Hockenmaier (POS)	83.1	83.5	91.1	91.5	91.5
C&C (POS)	**84.8**	**84.5**	**91.4**	**91.0**	**92.5**

Table 1: Dependency evaluation on Section 00 of the Penn Treebank

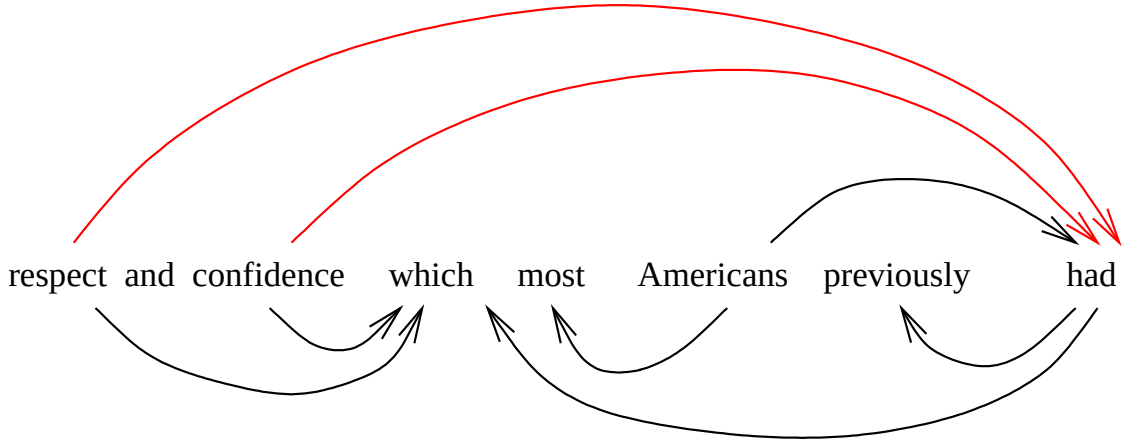
- To maintain comparability to Collins, Hockenmaier (2003) did not use a Supertagger, and was forced to use beam-search. With a Supertagger front-end, the Generative model might well do as well as the Log-Linear model. We have yet to try this experiment.

Log-Linear Overall Dependency Recovery

- The C&C parser has state-of-the-art dependency recovery.
- The C&C parser is very fast ($\approx$ 30 sentences per second)
- The speed comes from highly accurate supertagging which is used in an aggressive "Best-First increasing" mode (Clark and Curran 2004), and behaves as an "almost parser" (Bangalore and Joshi 1999
- Clark and Curran 2006 show that CCG all-paths almost-parsing with supertagger-assigned categories loses only 1.3% dependency-recovery F-score against parsing with a full dependency model
- C&C has been ported to the TREC QA task (Clark *et al.* 2004) using a hand-supertagged question corpus, and applied to the entailment QA task (Bos *et al.* 2004), using automatically built logical forms.

Recovering Deep or Semantic Dependencies

Clark *et al.* (2004)



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

Full Object Relatives in Section 00

- 431 sentences in WSJ 2-21, 20 sentences (24 object dependencies) in Section 00. 1. Commonwealth Edison now faces an additional court-ordered *refund* on its summerwinter rate differential collections *that* the Illinois Appellate Court has *estimated* at DOLLARS.

  2. Mrs. Hills said many of the 25 *countries that* she *placed* under varying degrees of scrutiny have made genuine progress on this touchy issue.

  √ 3. It's the petulant complaint of an impudent *American whom* Sony *hosted* for a year while he was on a Luce Fellowship in Tokyo – to the regret of both parties.

  √ 4. It said the *man, whom* it did not *name,* had been found to have the disease after hospital tests.

  5. Democratic Lt. Gov. Douglas Wilder opened his gubernatorial battle with Republican Marshall Coleman with an abortion *commercial* produced by Frank Greer *that* analysts of every political persuasion *agree* was a tour de force.

  6. Against a shot of Monticello superimposed on an American flag, an announcer talks about the strong *tradition* of freedom and individual liberty *that* Virginians have *nurtured* for generations.

  √ 7. Interviews with analysts and business people in the U.S. suggest that Japanese capital may produce the economic *cooperation that* Southeast Asian politicians have *pursued* in fits and starts for decades.

  8. Another was Nancy Yeargin, who came to Greenville in 1985, full of the *energy* and *ambitions that* reformers wanted to *reward*.

  9. Mostly, she says, she wanted to prevent the *damage* to self-esteem *that* her low-ability students would *suffer* from doing badly on the test.

  √ 10. Mrs. Ward says that when the cheating was discovered, she wanted to avoid the morale-damaging public *disclosure that* a trial would *bring*.

  √ 11. In CAT sections where students' knowledge of two-letter consonant sounds is tested, the authors noted that

Scoring High concentrated on the same *sounds that* the test *does* – to the exclusion of other *sounds that* fifth graders should *know*.

√ 12. Interpublic Group said its television programming *operations – which* it *expanded* earlier this year – agreed to supply more than 4,000 hours of original programming across Europe in 1990.

13. Interpublic is providing the programming in return for advertising *time, which* it *said* will be valued at more than DOLLARS in 1990 and DOLLARS in 1991.

√ 14. Mr. Sherwood speculated that the *leeway that* Sea Containers *has* means that Temple would have to substantially increase their bid if they're going to top us.

√ 15. The Japanese companies bankroll many small U.S. companies with promising products or ideas, frequently putting their money behind *projects that* commercial banks won't *touch*.

√ 16. In investing on the basis of future transactions, a role often performed by merchant banks, trading companies can cut through the *logjam that* small-company owners often *face* with their local commercial banks.

17. A high-balance *customer that* banks *pine for*, she didn't give much thought to the rates she was receiving, nor to the fees she was paying.

√ 18. The events of April through June damaged the *respect* and *confidence which* most Americans previously *had* for the leaders of China.

√ 19. He described the situation as an escrow *problem*, a timing *issue, which* he *said* was rapidly rectified, with no losses to customers.

√ 20. But Rep. Marge Roukema (R., N.J.) instead praised the House's acceptance of a new youth training wage, a *subminimum that* GOP administrations have *sought* for many years.

Cases of object extraction from a relative clause in 00; the extracted object, relative pronoun and verb are in italics; sentences marked with a √ are cases where the parser correctly recovers all object dependencies

Clark *et al.* (2004): Full Object Relatives

- 24 cases of extracted object in Section 00 associated with object relative pronoun category $(NP_X \backslash NP_X)/(S[dcl]/NP_X)$
- 15/24 (62.5%) recovered with all dependencies correct (15/20 (75%) precision)
  - That is, with both noun attachment and rel_pronoun-verb dependency correct—comparable to 58.3%/67% labelled recall/precision by Hockenmaier 2003 and significantly better than Clark *et al.* (2002) 42% recall
  - 1 sentence (1) failed to parse at all (necessary category for seen verb *estimated* unseen in 2-21).
  - 5 were incorrect because wrong category assigned to relative pronoun, of which: in two (5, 9) this was only because again the necessary category for a seen verb was unseen in 2-21, and one (17) was incorrect because the POS tagger used for back-off labeled the entirely unseen verb incorrectly
  - 3 incorrect only because relative clause attached to the wrong noun

Clark *et al.* (2004): Free Relatives

- 14/17 (82%) recall 14/15 (93%) precision for the single dependency.
- Better performance on long-range dependencies can be expected with more features such as regular expressions for Max Ent to work on.
- Other varieties of deep dependency (Control, subject relatives, reduced relatives) discussed in Hockenmaier (2003); Clark *et al.* (2002, 2004).
- It looks as though about half the errors arise because the lexicon is too small, and about half because the head-dependency model is too weak.


1M words of treebank is nothing like enough data

Experiments with Porting the Parser

- As with all treebank grammars, almost any practical application involves porting the parser to a different grammar and model.
- For example, in ongoing experiments with open domain question answering, we would like to use the parser for parsing the questions.
- However, all treebank grammars including this one do appallingly badly on the TREC question database, because WSJ contains almost no direct questions, and none at all of some common patterns.
- Hand-labelling data for retraining is usually not possible.
- However, semi-automatically hand-supertagging a few thousand sentences and retraining the supertagger with those included is quite practical.
- We did the 1,171 *What* questions from TREC in a week

Porting to Questions: Results

- 171 *What*-question development set. 1000 for training (and testing using tenfold cross-validation), average length 8.6 words.
- Since the gold standard question data is only labelled to the level of lexical category we can only evaluate to that level.
- However, supertagger accuracy and sentence accuracy correlate very highly with dependency and category recall by the parser, and we know we need around 97% per word and 60% per sentence for the original WSJ performance

MODEL	1 CAT ACC	SENT ACC	1.5 cats /word	SENT ACC
CCGbank	72.0	1.8	84.8	11.1
Qs	92.3	66.7	96.6	80.7
Qs+CCGbank	93.1	61.4	98.1	86.5

Table 2: Accuracy of Supertagger on Development set Question Data

Porting to Questions: Results

Supertagging/ parsing method	CAT ACC	SENT ACC	WHAT ACC
Increasing av. cats	94.6	81.8	91.2
Decreasing av. cats	89.7	65.3	80.0
Increasing cats (rand)	93.4	79.4	88.2
Decreasing cats (rand)	64.0	9.4	21.2
Baseline	68.5	0.0	60.6

Table 3: Category accuracy of parser on dev question data

- For the *What* object questions, per word/sentence accuracies were 90%/71%, suggesting that they are harder than the average question.
- Object dependency recall by the parser for these questions was 78%.

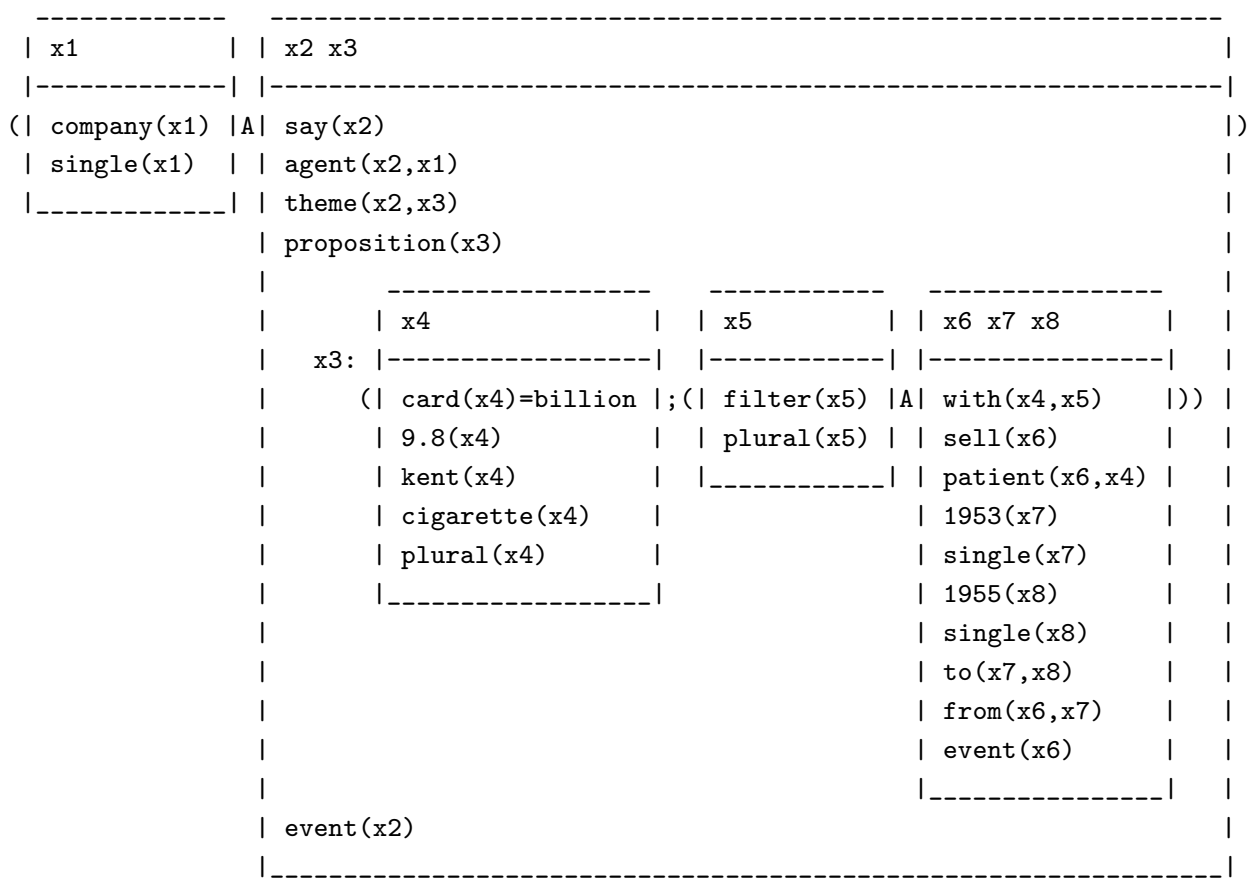
IV: Work in Progress

Work in Progress: Building Interpretations

- The interpretation of the combinatory rules as type raising and composition guarantees "surface compositionality" with any compositional semantic representation.
- This in turn means that the process of interpretation building can be built into the categories and combinatory rules, and can be done in parallel to derivation, as in (4)
- To make such a semantics wide-coverage involves specifying a semantics or a morphological stem-based semantic schema for the 400-500 most frequent category types (Hockenmaier *et al*. 2004; Bos et al. 2004)
- Generalize non-terminal categories containing unseen words.
- We use first order logics such as FOPL or DRT, using the lambda calculus as a "glue language".

Bos et al. 2004

From 1953 to 1955 , 9.8 billion Kent cigarettes with the filters were
sold , the company said .



The Poverty of Logicism

- Parsing with C&C 2004, and feeding such logical forms to a battery of FOL theorem provers, Bos and Markert (2005) attained quite high precision of 76% on the 2nd PASCAL RTE Challenge Problems.

However, recall was only 4%, due to the overwhelming search costs of FOL theorem proving.

- MacCartney and Manning (2007) argue that entailment must be computed much more directly, from the surface form of sentences, or from the strings themselves.

Work in Progress: Polarity

- It is well-known that explicit and implicit *negation* systematically switches the "upward" or "downward direction of entailment of sentences with respect to ontology-based inference:

(20) Egon walks $\vdash$ Egon moves

$\nvdash$ Egon walks quickly

Egon doesn't walk $\vdash$ Egon doesn't walk quickly

$\nvdash$ Egon doesn't move

- Sanchez Valencia (1991) and Dowty (1994) point out that polarity can be computed surface-compositionally using CG.

Polarity and Directional Entailment

- (21) doesn't$^{\circ}$:= $(S^{\circ}\backslash NP)/(S^{\bullet}_{inf}\backslash NP) : \lambda p.\bullet p$
- $\circ$ stands for the polarity of the syntactic/semantic environment, and $\bullet$ stands for $-\circ$, its inverse.
- Crucially, this category inverts the polarity of the predicate alone.

Polarity and Directional Entailment

- (22)

Enoch	doesn′t	walk
$\text{Enoch}^{+} := S^{\circ}/(S^{\circ}\backslash NP^{+}) : \lambda p.p\ {+}enoch'$	$\text{doesn}'\text{t}^{\circ} := (S^{\circ}\backslash NP)/(S^{\bullet}_{inf}\backslash NP) : \lambda p\lambda x.{\bullet}p\ {\circ}x$	$\text{walk}^{\circ} := S^{\circ}_{inf}\backslash NP : {\circ}walk'$
	$\text{doesn}'\text{t}^{\circ}\text{walk}^{\bullet} := S^{\circ}\backslash NP : {\bullet}walk'$ (>)	
$\text{Enoch}^{+}\text{doesn}'\text{t}^{+}\text{walk}^{-} := S^{+} : {-}walk'{+}enoch'$ (>)		

Work in Progress: Building Interpretations

- Quantifier scope alternation appears at first glance not to be surface compositional in the CCG sense, and is currently assigned by command-based default.
- Rather than generalizing the notion of surface derivation via further type-changing rules, we propose translating existentials as underspecified Skolem terms, integrating specification with derivation as an “anytime” operation (Steedman 2000).
- Dynamic phenomena such as anaphora (notably including tense) not yet covered at all.

V: Interim Conclusion

Where do we Go from Here?

- This performance is still bad by human standards.
- The main obstacle is that 1M words of annotated training data is not nearly enough,
- There are lots of words that never occur at all in the TreeBank at all.
  - This is a problem that the supertagger can help with. (In fact the front-end supertagger is already crucial to performance.)
- But a worse problem is words that *have* been seen, but *not with the necessary category*.
- The only answer to this problem is to generalize the grammar and the model, using
  - Active learning over unreliable parser output from unlabeled data, or
  - High precision low recall methods over web-scale amounts of data.

Moral

- You can have the linguistic expressivity that is needed to build interpretable structure *and* parse efficiently with wide coverage—with an automatically induced CCG lexicon and a statistical head-dependency model

Appendix: Child Language Aquisition

Child and Computer Language Development

- The child's problem is similar to the problem of inducing a treebank grammar, but a little harder.
  - They have unordered logical forms, not language-specific ordered derivation trees.
  - So they have to work out which word(s) go with which element(s) of logical form, as well as the directionality of the syntactic categories (which are otherwise universally determined by the semantic types of the latter).

Child and Computer Language Development

- Children do not seem to have to deal with a greater amount of error than the Penn WSJ treebank has (McWhinnie 2005).
  - But they may need to deal with situations which support a number of logical forms.
  - And they need to be able to recover from temporary wrong lexical assignments.
  - And they need to be able to handle lexical ambiguity.

Computational Accounts

- Siskind (1995, 1996), Villavicencio (2002), and Zettlemoyer and Collins (2005) offer computational models of this process.
- Both theories make strong assumptions about the association of words with elements of logical form.
- Both make strong assumptions about universally available parametrically specified rule- or category- types, the latter in the form of a type hierarchy
- Both deal with noise and homonymy probabilistically.

Computational Accounts: Zettlemoyer and Collins

- Zettlemoyer and Collins' algorithm (UAI 2005) allows any contiguous substring of the sentence to be a lexical item. For a given logical form, the learner has to search the cross-product of the substring powerset of the string with the set of pairs of legal categories with elements of the substructure powerset of the logical form for categories that yield combinatory derivations that yield the correct logical form.
- Learning is via a log-linear model using lexical entries (only) as features and gradient descent on their weights, iterating over successive sentences of a corpus of sentence-logical form pairs.
- We can improve on this by
  - Directly generating the parses that UG supports for the sentence-meaning pair.
  - Building a full parsing model (necessary if we are to scale).

Zettlemoyer and Collins (Contd.)

- The algorithm as presented in 2005 learns only a very small rather unambiguous fragment of English, hand-labeled with uniquely identified database queries as logical forms, and an English specific inventory of possible syntactic category types in lieu of Universal Grammar.
- CCG almost-parsing is why Zettlemoyer and Collins do so well on parser induction for a small not very ambiguous corpus without having a parser model at all.
- However, Siskind's and Villavicencio's results already tell us that the algorithm should work with multiple candidate logical forms.
- Similarly, their results suggest that a universal set of category types can be used without overwhelming the learner.

Zettlemoyer and Collins (Contd.)

- All of these models depend on availability to the learner of short sentences paired with logical forms, since complexity is determined by a cross-product of powersets both of which are exponential in sentence length.
- A number of techniques are available to make search efficient including use of a head-dependency parsing model.

The Generative Model

- We will assume that $P(D, I, S)$ is a generative model for an (exhaustive) parser, rather than the discriminative model of Zettlemoyer *et al.*.
- One advantage of generative models besides their closeness to competence grammar is that we can invert the parsing model to define the probability of an utterance given a meaning.

However, another difference between the child and standard treebank grammar-induction programs is that the child learns grammar *incrementally*, utterance-by-utterance.

Recomputing the model over the entire corpus so far, as each new sentence is encountered, is not only psychologically absurd, but computationally exponential.

Example

- The child thinks: $more' dog'$
- The Adult says: “More doggies!”
- Given the string “more dogs” paired with the logical form $more' dogs'$, and a mapping from semantic types onto syntactic type like S, NP, $S\backslash NP$ etc., the child can use the universal **BT**-based combinatory rules of CCG to generate
  - all possible syntactic derivations, pairing
  - all possible decompositions of the logical form with
  - all possible word candidates
- Learning a language is just learning its lexicon and a parsing model.

The Derivations

- CCG permits just three derivations for the new utterance “More doggies” , as follows:

(23) a.
$$\frac{\textsc{More} \qquad \textsc{doggies}}{\dfrac{NP/N : more'_{((e,t),e)} \quad N : dogs'_{(e,t)}}{NP : more'\,dogs'_e}\,>}\ !$$

b.
$$\frac{\textsc{More} \qquad \textsc{doggies}}{\dfrac{N : dogs'_{(e,t)} \quad NP\backslash N : more'_{((e,t),e)}}{NP : more'\,dogs'_e}\,<}\ !$$

c.
$$\frac{\textsc{More doggies}}{NP : more'\,dogs'_e}\ !$$

The Child's First Lexicon

- (24) The child's lexical candidates:

  more:= $\mathbf{NP/N} : \mathbf{more}'_{((\mathbf{e},\mathbf{t}),\mathbf{e})}$

  $N : dogs'_{(e,t)}$

  doggies:= $NP\backslash N : more'_{((e,t),e)}$

  $\mathbf{N} : \mathbf{dogs}'_{(\mathbf{e},\mathbf{t})}$

  more doggies:= $NP : (more'\, dogs')_e$

- A statistical model for these hypotheses can be learned using an incremental variant of the semi-supervised inside-outside (EM) algorithm (Pereira and Schabes 1992; Neal and Hinton 1999). We begin with a simplified model, representing probabilities as expected frequencies, then define the model we actually use.

Learning the Model for English

- In order to obtain an incremental algorithm, we represent the model as a vector of expected frequencies for each production p, defined as

(25) $fexp(p) = \sum_{s \in S} \sum_{i \in I} P(i|s) \sum_{d \in D} P(d|s,i).count(p,d)$,
where $P(d|s,i) = \frac{P(d)}{\sum_{d \in D} P(d)}$

The primary requirement for such a model is that learned information about seen events in a derivation should influence the probabilities assigned to unseen events.

- Thus, if the language only consists of sentences of the form "More X", and the hundredth sentence is "More erasers", where "erasers" is a previously unseen word, this sentence should not only make the learner a little more certain that "more" is a determiner meaning $more'$.
- It should also make them pretty sure that "erasers" is a noun, and *not* a determiner meaning $more'$.

Two Estimators for Expected Frequency

- We define two estimators for *fexp*.
- $Fexp_E$ is the expected frequency based on the present sentence and the possibilities of universal grammar alone. For simplicity we will assume the latter to be uniformly distributed, so that (25) reduces to the following, where $|D|$ is the number of derivations:

  (26) $fexp_E(p) = \frac{\sum_{d \in D} count(p,d)}{|D|}$

- $Fexp_M$ for a given interpretation i for sentence s is defined as follows, where P is the model estimated so far.

  (27) $fexp_M(p) = \sum_{i \in I} P(i|s) \sum_{d \in D} P(d|s,i).count(p,d)$

The Algorithm

- The model can be learned using the following incremental variant of the semi-supervised inside-outside (EM) algorithm (Pereira and Schabes 1992; Neal and Hinton 1999).
- Every new sentence s_n provides a set D_n of derivations parallel to (23), which defines the following:
  a. A (possibly empty) set of previously unseen productions involved in some derivation in D_i, including those involving novel lexical entries, that must be added to the model with cumulative *fexp* temporarily initialized to zero.
  b. (E-step): The set of all productions including those in a, whose cumulative *fexp* must be multiplied by $n-1$, incremented by $fexp_E$, and divided by n.
  c. (M-step):A further increment of $\frac{fexp_M - fexp_E}{n}$ (which may be negative) to the cumulative *fexp* for all productions involved in some derivation in D_i. I.e., replace the earlier estimate based on $fexp_E$.

The Algorithm

- Step b defines new values for the conditional probabilities for the rules in question, defining an intermediate model for calculating the a posteriori probabilities in step c.
- The further update c to the model defines the expected frequencies for the next cycle. The lexical probabilities for the relevant words in the lexicon given the new sentence can then be calculated using the model and definition (25), where $P(d|I,S)$ is the product of the probabilities of the productions it involves.
- (28) $P(d|I,S) = \prod_{p \in d} P(p|parent) \prod_{LEX(p) \in d} P(\phi, \sigma|\mu)$

This is just a probabilistic context-free grammar parser (PCFG). We actually use a head-dependency model (Collins 2003)

Normalizing Probabilities of Derivations

- The possibility of lexicalizing more than one element of the logical form in a single word means that the alternative derivations for a single logical form such as those in (23) for our running example and the first sentence "More doggies" may be of different lengths.
- Since generative models of the kind outlined above, based on the products of probabilities of rules, assign undue weight to short derivations, we must normalize the probabilities of lexical productions over the complexity of their logical forms.

Thus, the probability $P(\phi.\sigma|\mu)$ of the lexical productions in (28) is

(29) $P(\phi.\sigma|\mu) = \prod_{m \subset \mu} P(\phi,\sigma|m)$

- For example, the probability of derivation (23c) is not a third, but is the conditional probability of "more dogs" given *more*′*dogs*′ times that of "more dogs" given *more*′, times that of "more dogs" given *dogs*′—that is, $\frac{1}{3} \times \frac{1}{3} \times \frac{1}{3}$.

Probabilities of the Derivations

- Thus on the basis of the intermediate value $\frac{(0)fexp(0)+fexp_E}{1}$, the relative conditional probabilities $P(D|I,S)$ of the three derivations (23) are as follows:

(30) a $P(A|I,S) = P(r0|START) \times P(r1|NP : fa)) \times P_{lex}(more, NP/N|more') \times P_{lex}(doggies, N|dogs') = \frac{1\times 0.\dot{3}\times 0.\dot{3}\times 0.\dot{3}}{\sum_d P(d|I,S)}$

b $P(B|I,S) = P(r0|START) \times P(r2|NP : fa)) \times P_{lex}(doggies, NP\backslash N|more') \times P_{lex}(more, N|dogs') = \frac{1\times 0.\dot{3}\times 0.\dot{3}\times 0.\dot{3}}{\sum_d P(d|I,S)}$

c $P(C|I,S) = P(r0|START) \times P_{lex}(more\ \ doggies, NP|more') \times P_{lex}(more\ doggies, NP|dogs') = \frac{1\times 0.\dot{3}\times 0.3\times 0.\dot{3}}{\sum_d P(d|I,S)}$

$P(A|I,S) = P(B|I,S) = P(C|I,S) = 0.\dot{3}$

Child's First Parsing Model (Simplified)

- This means that the initial model can be calculated as follows:

(31)

Rule	$fexp(n-1)$	$\frac{(n-1)fexp(n-1)+fexp_E}{n}$	$fexp(n)$
r0. $START \rightarrow NP : fa$	0	1.0	1.0
r1. $NP : fa \rightarrow NP/N : f \quad N : a$	0	$0.\dot{3}$	$0.\dot{3}$
r2. $NP : fa \rightarrow N : a \quad NP\backslash N : f$	0	$0.\dot{3}$	$0.\dot{3}$
l1. $NP/N : more' \rightarrow$ more	0	$0.\dot{3}$	$0.\dot{3}$
l2. $NP\backslash N : more' \rightarrow$ doggies	0	$0.\dot{3}$	$0.\dot{3}$
l3. $N : dogs' \rightarrow$ doggies	0	$0.\dot{3}$	$0.\dot{3}$
l4. $N : dogs' \rightarrow$ more	0	$0.\dot{3}$	$0.\dot{3}$
l5. $NP : more' dogs' \rightarrow$ more doggies	0	$0.\dot{3}$	$0.\dot{3}$

The Child's First Lexicon

- Thus, we have the following updated probabilistic lexicon:

(32)

ϕ	σ, μ	f_{exp}	$P_{lex}(\sigma, \mu \mid \phi)$	$P_{lex}(\phi \mid \mu)$
more:=	$\mathbf{NP/N} : \mathbf{more}'_{((\mathbf{e},\mathbf{t}),\mathbf{e})}$	$0.\dot{3}$	$0.\dot{3}$	$0.\dot{3}$
	$N : dogs'_{(e,t)}$	$0.\dot{3}$	$0.\dot{3}$	$0.\dot{3}$
doggies:=	$NP\backslash N : more'_{((e,t),e)}$	$0.\dot{3}$	$0.\dot{3}$	$0.\dot{3}$
	$\mathbf{N} : \mathbf{dogs}'_{(\mathbf{e},\mathbf{t})}$	$0.\dot{3}$	$0.\dot{3}$	$0.\dot{3}$
more doggies:=	$NP : (more' dogs')_e$	$0.\dot{3}$	$0.\dot{3}$	$0.\dot{3}$

Early Overgeneration

- Since the word counts and conditional probabilities for “more” and “doggies” with them meaning $more'_{((e,t),e)}$ are all equal at this stage, the child may well make errors of overgeneration, using some approximation to “doggies” to mean “more”.
- However, even on the basis of this very underspecified lexicon, the child will not overgenerate “*doggies more”.
- Moreover, further observations, with further updates to frequency counts, will rapidly lower the estimated conditional probability of the spurious hypotheses concerning categories and substrings in comparison to the correct ones, indicated in bold type, as follows:

The Child's Second Sentence

- Let us suppose that the second utterance the child hears is "More cookies". There are again three derivations parallel to (23). The child can derive a new parsing model by adding new rules, updating expected frequencies for all rules in the new set of derivations, and recalculating a posteriori expected frequencies as described:

Prior Probabilities for the Three Possible Derivations

- On the basis of the intermediate value $\frac{(1)fexp(1)+fexp_E}{2}$,the length-weighted relative conditional probabilities $P(d|I,S)$ of the three derivations for "More cookies" parallel to (23) are as follows:

(33) a $P(A|I,S) = P(r0|START) \times P(r1|NP : fa)) \times P_{lex}(more, NP/N|more') \times P_{lex}(cookies, N|cookies') = \frac{1.0 \times 0.\dot{3} \times 0.\dot{3} \times 0.1\dot{6}}{\sum_d P(d|I,S)} = 0.42$

b $P(B|I,S) = P(r0|START) \times P(r2|NP : fa)) \times P_{lex}(cookies, NP\backslash N|more') \times P_{lex}(more, N|cookies') = \frac{1 \times 0.\dot{3} \times 0.1\dot{6} \times 0.1\dot{6}}{\sum_d P(d|I,S)} = 0.23$

c $P(C|I,S) = P(r0|START) \times P_{lex}(more\ cookies, NP|more') \times P_{lex}(more\ cookies, NP|cookies') = \frac{1 \times 0.\dot{3} \times 0.01\dot{6} \times 0.25}{\sum_d P(d|I,S)} = .35$


$P(A|I,S) \neq P(B|I,S) \neq P(C|I,S) \neq 0.\dot{3}$

The Child's 2nd Parsing Model (Simplified)

- (34)

Rule	$fexp(n-1)$	$\frac{(n-1)fexp(n-1)+fexp_E}{n}$	$fexp(n)$
r0. $START \rightarrow NP : fa$	1.0	1.0	1.0
r1. $NP : fa \rightarrow NP/N : f \quad N : a$	$0.\dot{3}$	$0.\dot{3}$	0.34
r2. $NP : fa \rightarrow N : a \quad NP\backslash N : f$	$0.\dot{3}$	$0.\dot{3}$	0.25
l1. $NP/N : more' \rightarrow$ more	$0.\dot{3}$	$0.\dot{3}$	0.34
l2. $NP\backslash N : more' \rightarrow$ doggies	$0.\dot{3}$	$0.1\dot{6}$	$0.1\dot{6}$
l3. $N : dogs' \rightarrow$ doggies	$0.\dot{3}$	$0.\dot{1}6$	$0.1\dot{6}$
l4. $N : dogs' \rightarrow$ more	$0.\dot{3}$	$0.1\dot{6}$	$0.1\dot{6}$
l5. $NP : more' dogs') \rightarrow$ more doggies	$0.\dot{1}$	$0.1\dot{6}$	$0.1\dot{6}$
l6. $NP : more' cookies' \rightarrow$ more cookies	0	$0.1\dot{6}$	0.17
l7. $NP\backslash N : more' \rightarrow$ cookies	0	$0.1\dot{6}$	0.11
r8. $N($cookies$) : cookies' \rightarrow$ cookies	0	$0.1\dot{6}$	0.24
l9. $N($more$) : cookies' \rightarrow$ more	0	$0.1\dot{6}$	0.11

The Child's Second Lexicon

- Thus, we have the following updated probabilistic lexicon:

(35)

ϕ	σ, μ	$fexp_{lex}(n)$	$P(\sigma,\mu\|\phi)$	$P(\phi\|\sigma,\mu)$
more:=	$\mathbf{NP/N : more'_{((e,t),e)}}$	**0.34**	**0.57895**	**0.57895**
	$N : dogs'_{(e,t)}$	$0.1\dot{6}$	0.26318	0.5
	$N : cookies'_{(e,t)}$	0.11	0.15789	$0.\dot{3}$
doggies:=	$NP\backslash N : more'_{((e,t),e)}$	$0.1\dot{6}$	0.5	0.385
	$\mathbf{N : dogs'_{(e,t)}}$	$0.1\dot{6}$	0.5	0.50
cookies:=	$NP\backslash N : more'_{((e,t),e)}$	0.11	$0.\dot{3}$	0.15789
	$\mathbf{N : cookies'_{(e,t)}}$	**0.24**	$\mathbf{0.\dot{6}}$	$\mathbf{0.\dot{6}}$
more doggies:=	$NP : (more' dogs')_e$	$0.1\dot{6}$	$0.\dot{3}$	$0.\dot{3}$
more cookies:=	$NP : (more' cookies')_e$	0.17	$0.\dot{3}$	$0.\dot{3}$

The Child's Second Lexicon

Notice that the expected frequencies in this table are not quite the same as those that would be obtained by recomputing f_{exp} over the entire corpus, as in standard batch EM.

- Nevertheless, at this point, the child is exponentially less likely to generate "doggie" when she means "more".
- Experimental sampling by elicitation of child utterances during such exponential extinction may well give the appearance of all-or-none setting of parameters like NEG-placement and *pro*-drop claimed by Thornton and Tesan (2006).
- This effect is related to the "winner-take-all" effect observed in Steels' 2004 game-based account of the very similar process of establishing a shared vocabulary among agents who have no preexisting language.

An Aside: A Statistically Sound Model

- We actually need a generative model that explicitly states the probabilities of the productions that are used in producing $\langle S, I, D \rangle$.
  - We model the probability of the syntactic derivation $P(D|START)$ using the PCFG type productions described before.
  - Each derivation gives a set of syntactic components $\underline{\sigma}$
- Kwiatkowski and Steedman (2009)

An Aside: A Statistically Sound Model

- We can now approximate the conditional probability of the associated semantics as:
  - $P(\lambda_{lex}|\sigma_i, \Lambda) \approx \frac{1}{Z} P(\lambda_{lex}|\Lambda) * t(\tau_\sigma, \tau_\lambda)$
  - t is a binary function that checks that the types of the syntax and semantics are compatible.
  - Λ is a model of the semantics available to the system. We break the lexical probability up as follows:
  - $P(\lambda_{lex}|\Lambda) = \prod_{\lambda_c \in \lambda_{lex}} P(\lambda_{lex}|\lambda_c) \times P(\lambda_c|\Lambda)$
  - The $P(\lambda_{lex}|\lambda_c)$ terms allows us to penalise complex semantics that appear in the lexicon.
  - The $P(\lambda_c|\Lambda)$ terms allow us to penalise rare semantics.

An Aside: A Statistically Sound Model

- The probability of $\langle S, I, D \rangle$ is calculated as:

$$P(\langle S, I, D \rangle | START, \Lambda) = P(D|START) \times \prod_i P(\phi_i|\sigma_i, \lambda_i) P(\lambda_i|\sigma_i, \Lambda)$$

- The grammar must model the production probabilities $P(p|parent)$
- The lexicon must model $P(\lambda_{lex}|\lambda_c)$, $P(\lambda_c|\Lambda)$, $P(\phi|\sigma, \lambda)$
- Incremental updates are made to these probability distributions by calculating likelihoods given each new sentence (as before) and using Bayes's rule to update the posterior belief, which is then stored.
- In order to make this simple, the grammar rules are modelled using a Dirichlet prior and the lexical probabilities are modelled using Dirichlet Processes.
  - In both cases the likelihood is conjugate to the prior, so the updates are easy to perform

Later Development

- This effect is also all that is needed to explain the phenomenon of “syntactic bootstrapping” (Gleitman (1990)), where at a later stage of development, the child can learn lexical entries for words for which the corresponding concept is not salient, or is even entirely lacking to the child.
- In this connection it is important that the expected frequency of the non-English rule r2 is already dropping in comparison to r1.

Discussion

- Syntax is learned on the basis of preexisting semantic interpretations afforded by the situation of adult utterance, using a statistical model over a universal set of grammatical possibilities.
- The existence of the model itself helps the child to rapidly acquire a correct grammar even in the face of competing ambiguous semantics and error, without requiring the (empirically questionable) subset principle.
- The fact that the onset of syntactically productive language at the end of the Piagetian sensory-motor develomental phase is accompanied by an explosion of advances in qualitatively different “operational” cognitive abilities suggests that the availability of the statistical model has a feedback effect, allowing “Syntactic bootstrapping” of concepts to which the child would not otherwise gain access.

Parameters and Triggers Unnecessary

- The theory presented here somewhat resembles the proposal of Fodor 1998 as developed in Sakas and Fodor (2001) and Niyogi (2006) in treating the acquisition of grammar as in some sense parsing with a universal "supergrammar". As in that proposal, both parameters and triggers are simply properties of the language-specific grammar itself—in their case, rules over independently learned parts of speech, in present terms, lexical categories.
- Rather than learning rules in an all or none fashion on the basis of unambiguous sentences that admit of only one analysis, the present theory adjusts probabilities in a model of all elements of the grammar for which there is positive evidence for *all* processable utterances.

Against "Parameter Setting"

- In this respect, it resembles the proposal of Yang (2002). However it differs in eliminating explicit parameters.
- If the parameters are implicit in the rules or categories themselves, and you can learn the rules or categories directly, why should the child (or a truly Minimal theory) bother with parameters at all?
- For the child, all-or-none parameter-setting is counterproductive, as it will make it hard to learn the many languages which have inconsistent settings of parameters across lexical types and exceptional lexical items, as in German and Dutch head finality.
- Or consider English expressions like the following:

(36) Doggies galore!

"Galore" is the only phrase-final determiner in E. (stolen from Irish).

References

Abney, Steven, 1996. "Statistical Methods and Linguistics." In Judith Klavans and Philip Resnik (eds.), *The Balancing Act*, Cambridge MA: MIT Press. 1–26.

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

Altmann, Gerry and Steedman, Mark, 1988. "Interaction with Context During Human Sentence Processing." *Cognition* 30:191–238.

Baldridge, Jason, 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Baldridge, Jason and Kruijff, Geert-Jan, 2003. "Multi-Modal Combinatory Categorial Grammar." In *Proceedings of 11th Annual Meeting of the European Association for Computational Linguistics*. Budapest, 211–218.

Bangalore, Srinivas and Joshi, Aravind, 1999. "Supertagging: An Approach to Almost Parsing." *Computational Linguistics* 25:237–265.

Bever, Thomas, 1970. "The Cognitive Basis for Linguistic Structures." In John Hayes (ed.), *Cognition and the Development of Language*, New York: Wiley. 279–362.

Bos, Johan, Clark, Stephen, Steedman, Mark, Curran, James R., and Hockenmaier, Julia, 2004. "Wide-Coverage Semantic Representations from a CCG Parser." In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04), Geneva*. ACL, 1240–1246.

Bos, Johan and Markert, Katya, 2005. "Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment." In *Proceedings of the First Challenge Workshop, Recognizing Textual Entailment*. `http://www.pascal-network.org/Challenges/RTE/`: Pascal, 65–68.

Buszkowski, Wojciech and Penn, Gerald, 1990. "Categorial Grammars Determined from Linguistic Data by Unification." *Studia Logica* 49:431–454.

Charniak, Eugene, 2000. "A Maximum-Entropy-Inspired Parser." In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*. Seattle, WA, 132–139.

Charniak, Eugene, Goldwater, Sharon, and Johnson, Mark, 1998. "Edge-Based Best-First Chart Parsing." In *Proceedings of the 6th Workshop on Very Large Corpora, Montreal, August*. 127–133.

Chomsky, Noam, 1957. *Syntactic Structures*. The Hague: Mouton.

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

Clark, Stephen and Curran, James R., 2003. "Log-Linear Models for Wide-Coverage CCG Parsing." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan, 97–104.

Clark, Stephen and Curran, James R., 2004. "Parsing the WSJ using CCG and Log-Linear Models." In *Proceedings of the 42nd Meeting of the ACL*. Barcelona, Spain, 104–111.

Clark, Stephen and Curran, James R., 2006. "Partial Training for a Lexicalized Grammar Parser." In *Proceedings of the Human Language Technology*

Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '06). New York.

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*. Philadelphia, PA, 327–334.

Clark, Stephen, Steedman, Mark, and Curran, James R., 2004. "Object-Extraction and Question-Parsing Using CCG." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain, 111–118.

Collins, Michael, 1997. "Three Generative Lexicalized Models for Statistical Parsing." In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid*. San Francisco, CA: Morgan Kaufmann, 16–23.

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

Collins, Michael, 2003. “Head-Driven Statistical Models for Natural Language Parsing.” *Computational Linguistics* 29:589–637.

Dowty, David, 1994. “The Role of Negative Polarity and Concord Marking in Natural Language Reasoning.” In *Proceedings of the Fourth Conference on Semantics and Theoretical Linguistics (SALT IV), Rochester, May*. Ithaca: CLC Publications, Cornell University.

Eisner, Jason, 1996. “Efficient Normal-Form Parsing for Combinatory Categorial Grammar.” In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA*. San Francisco, CA: Morgan Kaufmann.

Fodor, Janet Dean, 1998. “Unambiguous Triggers.” *Linguistic Inquiry* 29:1–36.

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.

Gleitman, Lila, 1990. “The Structural Sources of Verb Meanings.” *Language Acquisition* 1:1–55.

Goldberg, Adèle, 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. Chicago, IL: Chicago University Press.

Hepple, Mark, 1987. *Methods for Parsing Combinatory Grammars and the Spurious Ambiguity Problem*. Master’s thesis, University of Edinburgh.

Hockenmaier, Julia, 2003. *Data and models for statistical parsing with CCG*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Hockenmaier, Julia, Bierner, Gann, and Baldridge, Jason, 2004. “Extending the Coverage of a CCG System.” *Journal of Logic and Computation* 2:165–208.

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*. Las Palmas, Spain, 1974–1981.

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

Johnson, Mark, 2002. "A simple pattern-matching algorithm for recovering empty nodes and their antecedents." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA*. San Francisco, CA: Morgan Kaufmann, 136–143.

Joshi, Aravind, 1988. "Tree Adjoining Grammars." In David Dowty, Lauri Karttunen, and Arnold Zwicky (eds.), *Natural Language Parsing*, Cambridge: Cambridge University Press. 206–250.

Kor, Kian Wei, 2005. *Improving Answer Precision and Recall of List Questions*. Ph.D. thesis, Edinburgh.

Kwiatkowski, Tom and Steedman, Mark, 2009. "Computational Grammar Acquisition from CHILDES Data Using a Probabilistic Parsing Model." In *Submitted*.

MacCartney, Bill and Manning, Christopher D., 2007. "Natural Logic for Textual Inference." In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague: Association for Computational Linguistics, 193–200.

McCloskey, David, Charniak, Eugene, and Johnson, Mark, 2006. "Effective Self-Training for Parsing." In *Proceedings of the Human Language Technology Conference of the North American Chapter of ACL*. ACL, 152–159.

McWhinnie, Brian, 2005. "Item Based Constructions and the Logical Problem." In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition. CoNNL-9*. New Brunswick: ACL, 53–68.

Neal, Radford and Hinton, Geoffrey, 1999. "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants." In Michael Jordan (ed.), *Learning in Graphical Models*, Cambridge, MA: MIT Press. 355–368.

Niyogi, Partha, 2006. *Computational Nature of Language Learning and Evolution*. Cambridge MA: MIT Press.

Pereira, Fernando and Schabes, Yves, 1992. "Inside-Outside Reestimation from Partially Bracketed Corpora." In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*. ACL, 128–135.

Quine, Willard van Ormond, 1960. *Word and Object*. Cambridge MA: MIT Press.

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.

Sakas, William and Fodor, Janet Dean, 2001. "The Structural Triggers Learner." In S. Bertolo (ed.), *Language Acquisition and Learnability*, Cambridge: Cambridge University Press. 172–233.

Sanchez Valencia, Victor, 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, Universiteit van Amsterdam.

Siskind, Jeffrey, 1995. "Grounding Language in Perception." *Artificial Intelligence Review* 8:371–391.

Siskind, Jeffrey, 1996. “A Computational Study of Cross-Situational Techniques for Learning Word-to-Meaning Mappings.” *Cognition* 61:39–91.

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

Steels, Luc, 2004. “Constructivist Development of Grounded Construction Grammars.” In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona, 9–14.

Thornton, Rosalind and Tesan, Graciela, 2006. “Categorical Acquisition: Parameter Setting in Universal Grammar.” *Submitted* .

Vijay-Shanker, K. and Weir, David, 1990. “Polynomial Time Parsing of Combinatory Categorial Grammars.” In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh*. San Francisco, CA: Morgan Kaufmann, 1–8.

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

Villavicencio, Aline, 2002. *The Acquisition of a Unification-Based Generalised Categorial Grammar*. Ph.D. thesis, University of Cambridge.

Xia, Fei, 1999. "Extracting Tree Adjoining Grammars from Bracketed Corpora." In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium(NLPRS-99)*.

Yang, Charles, 2002. *Knowledge and Learning in Natural Language*. Oxford: Oxford University Press.

Zettlemoyer, Luke and Collins, Michael, 2005. "Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars." In *Proceedings of the 21st Conference on Uncertainty in AI (UAI)*. ACL, 658–666.

Zettlemoyer, Luke S., Pasula, Hanna M., and Kaelbling, Leslie Pack, 2005. "Learning Planning Rules in Noisy Stochastic Worlds." In *National Conference on Artificial Intelligence (AAAI),*. AAAI.