Formal Grammar and Psycholinguistic Theory

Mark Steedman

[The] development of probabilistic models for the use of language (as opposed to [its] syntactic structure) can be quite interesting. . .

One might seek to develop a more elaborate relation between statistical and syntactic structure than the simple order of approximation model we have rejected.

Chomsky 1957:17n4

Outline

I Introduction: Zipf's Elephant

II Nearly Context-Free Grammar

III Processing: The Strict Competence Hypothesis

IV Wide-Coverage Parsing

V Origins of Grammar in Planned Action

I: Introduction: Zipf's Elephant

- Once upon a time, linguists, psycholinguists, and computational linguists shared a common view of the nature of language.
- This view was based on some results by Chomsky (1957) showing that human language capacity could not be exactly captured using certain simple classes of automaton (finite state machines and simple push-down automata).
- The critical data were to do with *unbounded* constructions, such as relativization and coordination, which seemed to need structure-changing rules of *movement* and *deletion under identity*.
- Chomsky's argument rested on a distinction between "competence", or the nature of the computation, and "performance", the algorithm or mechanism by which the computation was carried out.

Chomsky was careful to leave open the possibility that Markov processes and other approximate models might be important to performance.

The Fall

- This consensus fell apart around 1973.
- In linguistics, there was a fundamental disagreement about the role of semantics in the theory, with Chomsky (1957:chs.9,10, *passim*) insisting on the methodological primacy of syntactic intuition over semantics, and the generative semanticists (e.g. Lakoff 1970b; McCawley 1972) insisting on the logical primacy of semantics over syntax.
- Meanwhile, in mathematical and computational linguistics it had been realized that transformational rules were very expressive—in fact, Turing Machine-complete (Peters and Ritchie 1973)—implying very weak explanatory power.

Paradise Lost

- Linguistics split into two camps, according to two slogans:
  - Take care of the *syntax*, and the semantics will take care of itself!
  - Take care of the *semantics*, and the syntax will take care of itself!
- As a result, very few linguists worked to bring syntax and semantics closer together (but cf. Geach 1972:497 on "breaking Priscian's head").

Both sides disavowed any further interest in constraining the expressivity of syntactic rules (cf. Lakoff 1970a).

The Flaming Sword

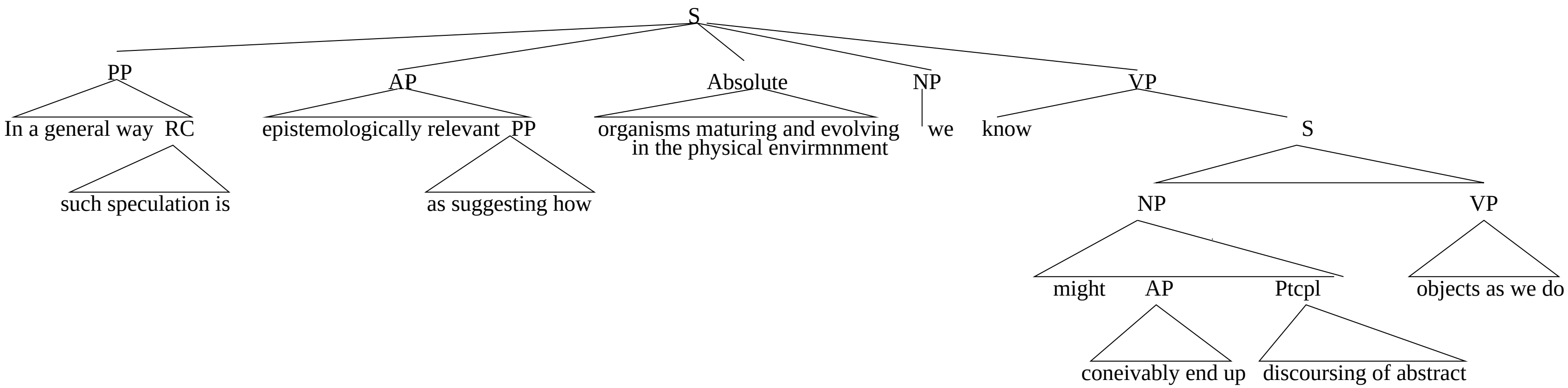
- As a result, experimental psycholinguists have become largely agnostic about both formalist and semanticist linguistic theories (despite a general tendency to empathize with the latter).
  - They asked for bread. We gave them empty categories.
  - Some have even abandoned the Competence/Performance distinction, and turned in desperation to connectionism.
- Meanwhile, the computational linguists did pursue the goal of low (near-context-free) expressive power and polynomial recognition (Gazdar 1981; Ades and Steedman 1982; Joshi 1988).

But once the machines got large enough to try these ideas out on a realistic scale, it became apparent that human grammars are too vast to define by hand, and too ambiguous for even computers to search for parses exhaustively.

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:



This is only one among the thousands of spurious derivations that typically arise from even moderately complicated sentences.

East of Eden

- None of the psychologists "parsing strategies" (Kimball 1973; Frazier 1978 helped at all with this problem.

- Such strategies assume that there are only ever at most two analyses to worry about, but computational linguists know there may be millions.
- As a result, the computational linguists returned to the same simple models that Chomsky rejected, using machine learning to induce grammars and build parsing models for finite-state and context-free systems.

Zipf's Elephant

- Linguists, psycholinguists, and computational are as blind men defining an elephant by feel.
- The elephant is Zipf's Law, which says that everything in language, from word-frequency to frequency of constructions, obeys a doubly exponential power law.
- The computational linguists only feel the fat finite-state and context-free end of the distribution, because that is susceptible to machine learning and accounts for 90% of the variance.
- The linguists only feel the "long tail", because they know that that is where the important information about the nature of the system is to be found.
- The psychologists don't know where to turn.

The Present Danger

- We are already running up against the limitations of incomplete finite state and context-free approximations.
  - Even for apparently susceptible problems like automatic speech recognition (ASR), current linear improvement is due to the machines (and therefore the approximate models) getting exponentially larger.
  - Even if Moore's Law continues to hold, it is not clear that there ever could be enough training data to make such models approach human performance (Lamel *et al*. 2002; Moore 2003).
- Machine learning is very bad indeed at acquiring systems for which important information is in rare events.

What To Do

- We need a readily extensible, construction-based theory of universal grammar.
- It must support a learnable parsing model for robust and efficient wide-coverage parsing.
- It must directly consitute a model for psychological language processing and language acquisition.
- It must be transparent to a “natural” semantics, supporting cheap inference.

II: “Nearly Context-Free” Grammar

Categorial Grammar

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

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

- 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 placeholder 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) Marcel proved completeness

$$\begin{array}{lll}
\text{Marcel} & \text{proved} & \text{completeness} \\
NP & (S\backslash NP)/NP & NP \\
 & \multicolumn{2}{l}{\overline{\qquad S\backslash NP \qquad}^{>}} \\
\multicolumn{3}{l}{\overline{\qquad\qquad S \qquad\qquad}^{<}}
\end{array}$$

(5) I think Marcel proved completeness

$$\begin{array}{lllll}
\text{I} & \text{think} & \text{Marcel} & \text{proved} & \text{completeness} \\
NP & (S\backslash NP)/_{\diamond}S & NP & (S\backslash NP)/NP & NP \\
 & & & \multicolumn{2}{l}{\overline{\qquad S\backslash NP \qquad}^{>}} \\
 & & \multicolumn{3}{l}{\overline{\qquad\qquad S \qquad\qquad}^{<}} \\
 & \multicolumn{4}{l}{\overline{\qquad\qquad S\backslash NP \qquad\qquad}^{>}} \\
\multicolumn{5}{l}{\overline{S \qquad\qquad\qquad}^{<}}
\end{array}$$

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)

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

(5)

$$\begin{array}{ccccc}
\text{I} & \text{think} & \text{Marcel} & \text{proved} & \text{completeness} \\
\hline
NP : i' & (S\backslash NP)/_{\diamond}S : think' & NP : marcel' & (S\backslash NP)/NP : prove' & NP : completeness' \\
 & & & \multicolumn{2}{c}{\overline{\quad S\backslash NP : \lambda y.prove'completeness'y \quad}>} \\
 & & \multicolumn{3}{c}{\overline{\quad S : prove'completeness'marcel' \quad}<} \\
 & \multicolumn{4}{c}{\overline{\quad S\backslash NP : think'(prove'completeness'marcel') \quad}>} \\
\multicolumn{5}{c}{\overline{\quad S : think'(prove'completeness'marcel')i' \quad}<}
\end{array}$$

Combinatory Categorial Grammar (CCG)

- Steedman (2000)
- **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)

- Steedman (2000)
- **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**

Combinatory Derivation

(6)

$$
\begin{array}{ccc}
\text{Marcel} & \text{proved} & \text{completeness} \\
\hline NP & \overline{(S\backslash NP)/NP} & \overline{NP} \\
\overline{S/(S\backslash NP)}^{>\mathbf{T}} & & \overline{S\backslash(S/NP)}^{<\mathbf{T}} \\
\multicolumn{2}{c}{\overline{S/NP}^{>\mathbf{B}}} & \\
\multicolumn{3}{c}{\overline{S}^{<}}
\end{array}
$$

(7)

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

Combinatory Derivation

(6) Marcel proved completeness

$$\text{Marcel}: NP : marcel' \qquad \text{proved}: (S\backslash NP)/NP : prove' \qquad \text{completeness}: NP : completeness'$$

$$\xrightarrow{>\mathbf{T}} S/(S\backslash NP) : \lambda f.f\ marcel' \qquad \xrightarrow{<\mathbf{T}} S\backslash(S/NP) : \lambda p.p\ completeness'$$

$$\xrightarrow{>\mathbf{B}} S/NP : \lambda x.prove'x\ marcel'$$

$$\xrightarrow{<} S : prove'completeness'marcel'$$

(7) Marcel proved completeness

$$\text{Marcel}: NP : marcel' \qquad \text{proved}: (S\backslash NP)/NP : prove' \qquad \text{completeness}: NP : completeness'$$

$$\xrightarrow{>\mathbf{T}} S/(S\backslash NP) : \lambda f.f\ marcel' \qquad \xrightarrow{<\mathbf{T}} (S\backslash NP)\backslash((S\backslash NP)/NP) : \lambda p.p\ completeness'$$

$$\xrightarrow{<} S\backslash NP : \lambda y.prove'completeness'y$$

$$\xrightarrow{>} S : prove'completeness'marcel'$$

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

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)

a	man	who	I	think	you	like	arrived
$(S/(S\backslash NP))/N$	N	$(N\backslash N)/(S/NP)$	$S/(S\backslash NP)$	$(S\backslash NP)/_{\diamond}S$	$S/(S\backslash NP)$	$(S\backslash NP)/NP$	$S\backslash NP$

- I think: $>\mathbf{B}$ $S/_{\diamond}S$
- you like: $>\mathbf{B}$ S/NP
- I think you like: $>\mathbf{B}$ S/NP
- who I think you like: $>$ $N\backslash N$
- man who I think you like: $<$ N
- a man who I think you like: $>$ $S/(S\backslash NP)$
- a man who I think you like arrived: $>$ S

Predictions: Argument-Cluster Coordination

- The following construction is predicted on arguments of symmetry.

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

$$\underline{DTV}\quad \underline{TV\backslash DTV}_{<\mathbf{T}}\quad \underline{VP\backslash TV}_{<\mathbf{T}}\quad \underline{(X\backslash_{\star}X)/_{\star}X}\quad \underline{TV\backslash DTV}_{<\mathbf{T}}\quad \underline{VP\backslash TV}_{<\mathbf{T}}$$

$$\underline{\qquad VP\backslash DTV \qquad}_{<\mathbf{B}} \qquad\qquad \underline{\qquad VP\backslash DTV \qquad}_{<\mathbf{B}}$$

$$\underline{(VP\backslash DTV)\backslash_{\star}(VP\backslash DTV)}_{>}$$

$$\underline{\qquad VP\backslash DTV \qquad}_{<}$$

$$\underline{\qquad VP \qquad}_{<}$$

—**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:

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

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.

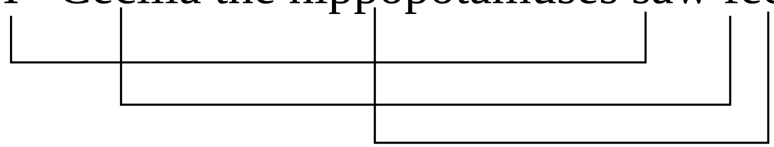
Language Type	**Automaton**	**Rule-types**	**Exemplar**
Type 0: RE	Universal Turing Machine	$\alpha \rightarrow \beta$	
Type 1: CS	Linear Bound Automaton (LBA)	$\phi A \psi \rightarrow \phi \alpha \psi$	$\mathcal{P}(a^n b^n c^n)$ (?)
I	Nested Stack Automaton(NSA)	$A_{[(i),...]} \rightarrow \phi B_{[(i),...]}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

A Trans-Context Free Natural Language

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

... omdat ik Cecilia de nijlpaarden zag voeren.

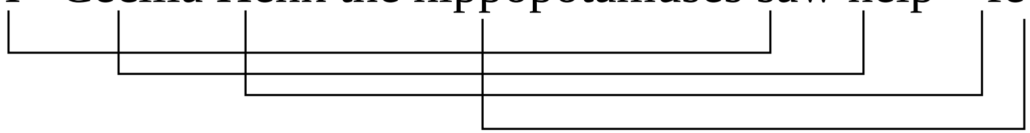
... because I Cecilia the hippopotamuses saw feed



'... because I saw Cecilia feed the hippopotamuses.'

... omdat ik Cecilia Henk de nijlpaarden zag helpen voeren.

... because I Cecilia Henk the hippopotamuses saw help feed



'... because I saw Cecilia help Henk feed the hippopotamuses.'

III: Parsing: The Strict Competence Hypothesis

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.

Human Sentence Processing

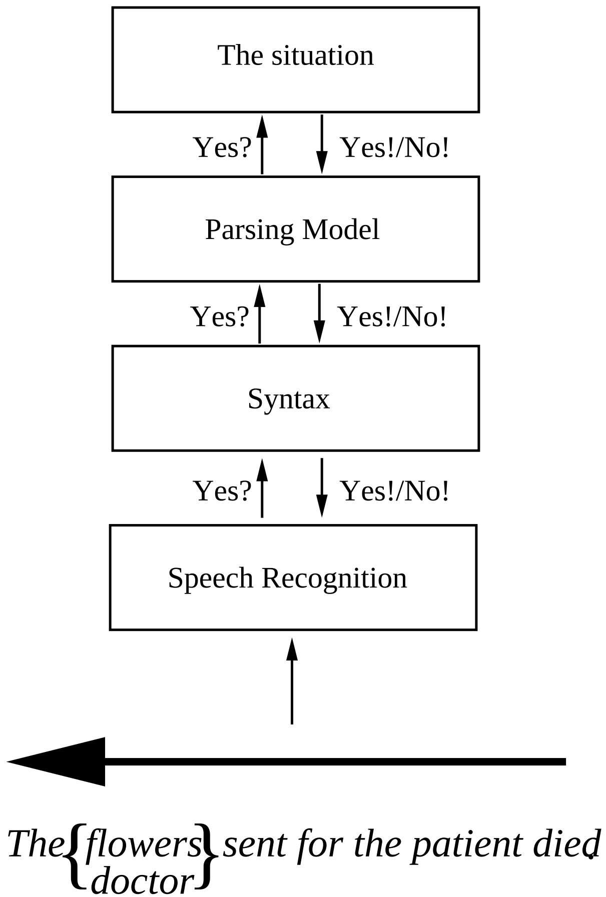
- "Garden path" sentences are sentences which are grammatical, but which naive subjects fail to parse.
- Example (11a) is a garden path sentence, because the ambiguous word "sent" is analysed as a tensed verb:

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

- However (11b) 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 (12a,b) involve the nonstandard constituent [The doctor sent for]$_{S/NP}$, means that constituent is also available for (12c,d)

(12) 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] $\left\{\begin{array}{lll} S/NP & & \\ (S/(S\backslash NP))/N & N & (N\backslash N)/NP \end{array}\right\}$ [the patient]$_{NP}$ died$_{S\backslash NP}$.

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

- (13) 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 (11b) 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.

A Problem for Strict Competence

- Sturt and Lombardo (2005) suggest that CCG is not incremental *enough* for Strict Competence to hold:

  (14) The pilot insulted the stewardess and put #herself in an embarrassing position.

- Is such a simple parser possible? And is it correct? We need to look at some real-life parsing programs.

IV: Wide Coverage Parsing

Wide Coverage Parsing

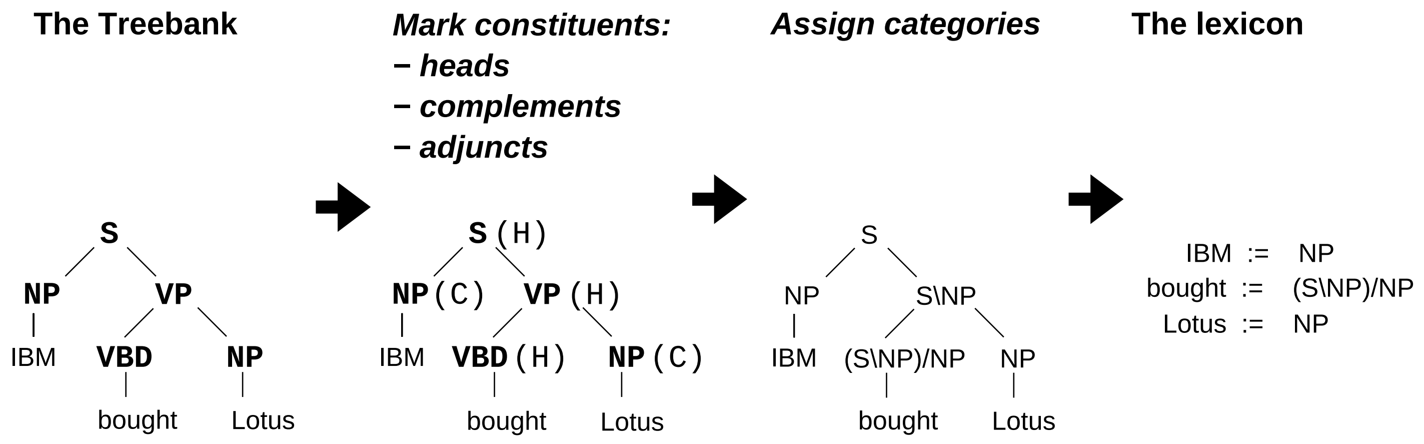
- Early attempts to model parse probability by attaching probabilities to rules of CFG performed poorly.
- Great progress as measured by the ParsEval measure has been made by combining statistical models of headword dependencies with CF grammar-based parsing (Collins 1997; Charniak 2000; McCloskey *et 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 (2002), and Clark and Curran (2004).

Supervised CCG Induction by Machine

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

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

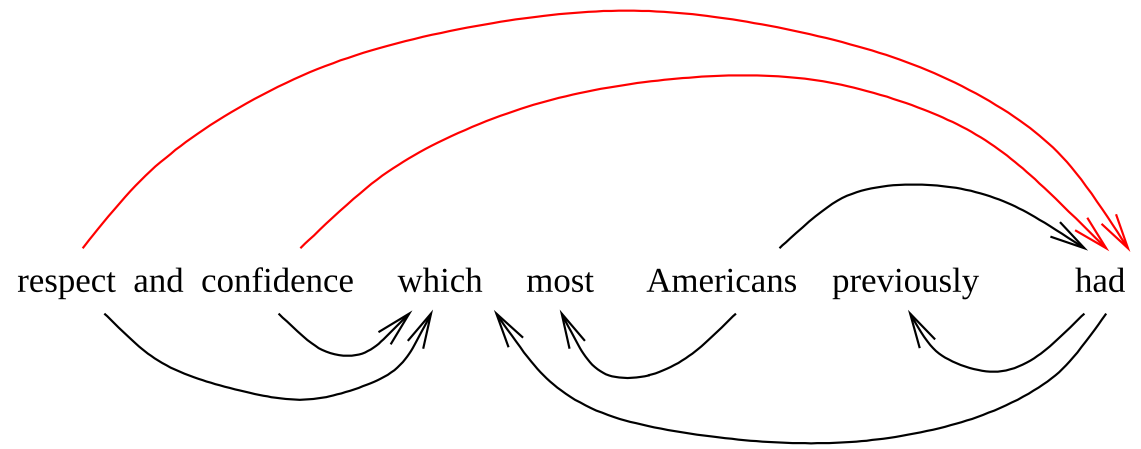
- To maintain comparability to Collins, Hockenmaier (2003) did not use a Supertagger, and was forced to use beam-search. With a Supertagger front-end, the Generative model might well do as well as the Log-Linear model. We have yet to try this experiment.

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*.

$\surd$ 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.

$\surd$ 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.

$\surd$ 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*.

$\surd$ 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.

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

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

$\surd$ 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 $\surd$ are cases where the parser correctly recovers all object dependencies

Sturt and Lombardo's Example

- The head dependency parsing model is entirely consistent with Strict Competence.

- The use of a Morkovian/Perceptron-like supertagger front-end in the C&C parser means that the bottom-up CKY parser can predict a locative PP after "put herself"

(15) The pilot insulted the stewardess and put #herself in an embarrassing position.

- So Strict Competence survives: the parser never builds anything the grammar doesn't countenance.

V: Grammar and Planned Action

The Statistical Problem of Language Acquisition

- The syntax, the semantics and the operations of the CCG processor are essentially isomorphic.

- The tight coupling between syntactic and semantic combination means that the “logical” problem of child language acquisition reduces to the problem of learning a parsing model for all the options that universal grammar would allow, on the basis of exposure to strings of the language paired with possibly ambiguous, erroneous, and noisy meaning representations (Kwiatkowski and Steedman 2009)

But the possible lexical types and the individual combinatory rule types must be given.

Combinators, Planning, And Affordance

- The possible lexical types are determined by the conceptual base.
- But where do the combinators come from?
  - If **actions** are functions from situations to situations, then composing actions to form novel actiions or **plans** is function composition **B**.
  - If the **affordances** of tools are functions from the actions that they afford to the results of those actions, also to be used in planning, then our concepts of a tool is type-raised **T**.

Planning And Affordance

- Some animals can make quite complex plans involving tools (Köhler 1925).

Figure 1: Chimpanzee plans (from Köhler 1925)

Planning And Affordance

- Such planning seems to be *reactive* to the presence of the tool and *forward-chaining* (working from tool to goal), rather than backward-chaining (working from goal to tool).
- This seems a good way for an animal to plan, and implies that actions are accessed via perception of the objects that mediate them—in other words that actions are represented as the *Affordances* of objects, in Gibson's terms.
- So it is reasonable to suppose that prelinguistic animal planning provides the cognitive substrate for syntactic composition and type-raising.

Origins of Recursive Syntax

- *So why don't Sultan, Washoe, Kanzi, and the rest of them use language like you and me?*
- There is only one place where there is room for any difference.
- That is the lexicon, and the concepts that it lexicalizes.
- Concepts like "knows" are truly recursive.
- If the ape concept of other minds does not support thie same concept, then their syntax may not be recursive.
- This suggests a *semantic* origin for the claim of Hauser *et al*. (2002) that recursion is distinctive.

Moral

- It's possible to have a fully formal theory of grammar that is fully transparent to psycholinguistic processing under the Strict Competence Hypothesis.

- The grammar is also compatible with a semantics cognitively grounded in action in the world.

- There are many open problems in defining such a semantics, but we should insist that any hypotheses concerning it are surface compositional, consistent with derivation in a near-context free syntax without non-monotonic structure-changing rules of movement or deletion.

References

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

Ades, Anthony and Steedman, Mark, 1982. "On the Order of Words." *Linguistics and Philosophy* 4:517–558.

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

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.

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 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, 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.

Davidson, Donald and Harman, Gilbert (eds.), 1972. *Semantics of Natural Language*. Dordrecht: Reidel.

Frazier, Lyn, 1978. *On Comprehending Sentences*. Ph.D. thesis, University of Connecticut.

Gazdar, Gerald, 1981. “Unbounded Dependencies and Coordinate Structure.” *Linguistic Inquiry* 12:155–184.

Geach, Peter, 1972. “A Program for Syntax.” In Donald Davidson and Gilbert Harman (eds.), *Semantics of Natural Language*, Dordrecht: Reidel. 483–497.

Hauser, Marc, Chomsky, Noam, and Fitch, W. Tecumseh, 2002. “The Faculty of Language: What Is It, Who Has It, and How did it Evolve?” *Science* 298:1569–1579.

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

Hockenmaier, Julia and Steedman, Mark, 2002. “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.

Joshi, Aravind, 1988. “Tree Adjoining Grammars.” In David Dowty, Lauri

Karttunen, and Arnold Zwicky (eds.), *Natural Language Parsing*, Cambridge: Cambridge University Press. 206–250.

Kimball, John, 1973. “Seven Principles of Surface Structure Parsing in Natural Language.” *Cognition* 2:15–47.

Köhler, Wolfgang, 1925. *The Mentality of Apes*. New York: Harcourt Brace and World.

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

Lakoff, George, 1970a. “Global Rules.” *Language* 46:24–50.

Lakoff, George, 1970b. “Linguistics and Natural Logic.” *Synthése* 22:151–271. reprinted in Davidson and Harman 1972:545-665.

Lamel, Lori, Gauvain, J-L., and Adda, G., 2002. “Unsupervised Acoustic Model Training.” In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*. IEEE, 877–880.

McCawley, James, 1972. “A Program for Semantics.” In Donald Davidson and Gilbert Harman (eds.), *Semantics of Natural Language*, Dordrecht: Reidel. 498–544.

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.

Moore, Roger, 2003. “A Comparison of the Data Requirements of Automatic Speech Recognition Systems and Human Listeners.” In *Proceedings of Eurospeech Conference*. 2582–2585.

Peters, Stanley and Ritchie, Robert, 1973. “On the Generative Power of Transformational Grammars.” *Information Science* 6:49–83.

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

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

Sturt, Patrick and Lombardo, Vincenzo, 2005. “Processing Coordinated Structures: Incrementality and Connectedness.” *Cognitive Science* 29:291–305.

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

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