LCTG—Notes Part 9: Lexicalized Probabilistic Parsing

One might seek to develop a more elaborate relation between statistical and syntactic structure than the simple order of approximation model we have rejected. I would certainly not care to argue that any such relation is unthinkable, but I know of no suggestion to this effect that does not have obvious flaws.

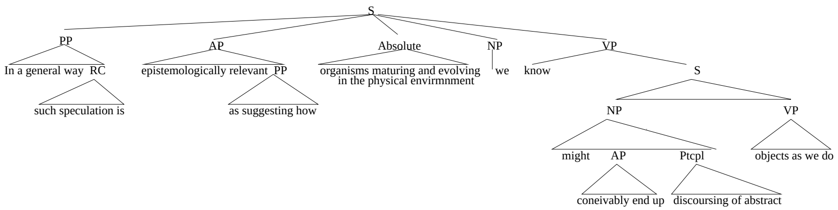
Chomsky, 1957:17, note 4

Why Wide-Coverage Parsing?

- No handwritten grammar ever has the coverage that is needed for e.g. the daily newspaper.
- Statistical parsing types are fond of quoting Sapir at this point, who famously said “All grammars leak.” (This is not in fact what Sapir meant. Sapir was talking about overgeneration, and in his sense, their grammars leak most of all).
- Language is highly ambiguous and it is hard to pick the best parse. Quite ordinary sentences of the kind you read every day turn out to have thousands of parses, albeit mostly 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, cf. Abney 1996).
- —yields the following, among many other horrors:



Three Problems for Wide Coverage Parsers

1. Find best grammar for some language automatically.
2. Select the best parse for a sentence.
3. Efficiently recover the best parse(s).

Probabilistic Background

- Let G be a grammar (from the set of all grammars G^*).
- Let D be a sequence of strings (not necessarily sentences).
- The 'best' grammar might be:

$$G = \text{argmax}_G = \frac{P(G)P(D \mid G)}{\sum_{F \in G^*} P(F)P(D \mid F)}$$

—where $\text{argmax}_X = f(X)$ means "the X that maximizes $f(X)$"

- This is Bayes theorem, which says that the likelihood of a hypothesis G in comparison to every alternative hypothesis is its prior probability $P(G)$ times the probability of the data given that hypothesis—in other words, that to believe a hypothesis it has to *both* be intrinsically probable *and* predict observations.
- $P(G)$ reflects grammar "goodness" (which is going to be a problem).

Generative Parser Models

- Generative models of sentence probabilities are closely related to the grammar itself.
- A sentence s_i will have parses $q_1 \dots q_n$.
- Each parse q_i consists of a sequence of k rule applications.

$$P(q_i) = \prod_{i=1}^{k} P(rule_i : A \rightarrow \alpha \mid A)$$

- Each parse is distinct, so:

$$P(s_i) = \sum_{i=1}^{n} P(q_i)$$

- Conditioning on mother (A) gives a *probabilistic context free grammar* (PCFG).

Probabilistic Background (Contd.)

- If $D = \{s_1, \dots, s_n\}$, and we assume each sentence s_i is independent of each other:

$$P(D \mid G) = P(\{s_1, \dots, s_n\} \mid G) \quad = \quad \prod_{i=1}^{n} P(s_i)$$

- Specifying the prior $P(G)$ is harder.
- Could use linguistic intuitions:
  - Prefer grammars that use headed rules.
  - Prefer grammars that are compact.
  - Prefer CFGs over Turing machines.
  - etc.

Unsupervised Grammar Induction

- One approach:
  - Assume all grammars equally good.
  - Search for grammar that maximises $P(D \mid G)$.
  - This gives bad results (it has been shown that G is invariably linguistically implausible).
- Another approach:
  - Assume compact grammars are better than verbose grammars.
  - Search for grammar that maximises $P(G)P(D \mid G)$ as earlier slide.
  - This gives somewhat better results but still poor.
  - The problem of **unsupervised grammar induction** is a very hard problem, on which no real progress has been made.

Supervised Grammar Induction

- Children do not learn grammar like this from speech or text. They have access to **meaning representations**.
- Apart from dealing with ambiguity (which possible meaning representation) and noise (exactly what string) this reduces the problem to working out which word(s) are paired with which elements of meaning, and what order they combine in. (In CCG terms, this comes down to identifying the directionality of the slashes.)
- Much greater success with grammar induction has been obtained by putting the machine in the same position, using **supervised** learning.

Supervised Grammar Induction and Parsing

- A grammar and frequency counts are derived from a hand-annotated parsed corpus (eg Wall Street Journal).
- WSJ is around a million words, somewhat inaccurately and inconsistently annotated, divided into 24 sections of which sections 0, 1, and 22 are reserved for development, sections 2-21 for training material and sections 23 and 24 reserved as unseen material for testing.
- Estimate PCFG rule probabilities using maximum likelihood estimator:

$$P(A \rightarrow \alpha \mid A) \approx \frac{freq(A \rightarrow \alpha)}{\sum_{\gamma} freq(A \rightarrow \gamma)}$$

—where $A \rightarrow \alpha$ is a rule expanding A as a string α of terminals and non-terminals, and $freq(X)$ is the frequency of X.

A Naive PCFG

- A naive SCFG to get us started:

Rule	Prob	Rule	Prob
$S \rightarrow NP\ VP$	1.0	$NP \rightarrow Det\ N$	0.6
$NP \rightarrow PN$	0.4	$PN \rightarrow$ John	0.5
$PN \rightarrow$ Mary	0.5	$N \rightarrow$ cat	0.6
$VP \rightarrow V\ NP$	1.0	$V \rightarrow$ loves	0.6
$V \rightarrow$ saw	0.4	$Det \rightarrow$ a	1.0
$N \rightarrow$ saw	0.4		

- The grammar has been chosen so that all NPs and verbs are singular. A more realistic grammar would have to distinguish number to ensure a sound model (Abney 1997).

Probabilistic DCG

- We can translate the naive PCFG into a (P) DCG, passing probabilities with variables P0 etc. and using the “curly brackets” augmentation to compute probabilities on the side, obtaining the product of all conditional rule probabilities via recursion:

```
s(P0, [s,[NP, VP]]) -->                          %Rule r1
   np(P1, NP),  vp(P2, VP),
   {p(r1, P), P0 is P*P1*P2}.

p(r1, 1.0).

np(P0, [np,[PN]]) -->                             %Rule r2
   pn(P1, PN),
   {p(r2,P), P0 is P*P1}.

p(r2, 0.3).
```

```
np(P0, [np,[Det, N]]) -->                                    %Rule r3
   det(P1, Det), n(P2, N),
   {p(r3,P), P0 is P*P1*P2}.

p(r3, 0.7).

vp(P0, [vp,[V, NP]]) -->                                     %Rule r5
   v(P1, V), np(P2, NP),
   {p(r5,P), P0 is P*P1*P2}.

p(r5, 0.3).
```


Probabilistic DCG (Contd.)

- The lexicon:

```
pn(0.5, pn[john]) --> [john].
pn(0.5, pn[mary]) --> [mary].
v(0.4, [v,[saw]]) --> [saw].
v(0.6, [v,[loves]]) --> [loves].
det(1.0, [det,[a]]) --> [a].
n(0.6, [n,[cat]]) --> [cat].
n(0.4, [n,[saw]]) --> [saw].
```

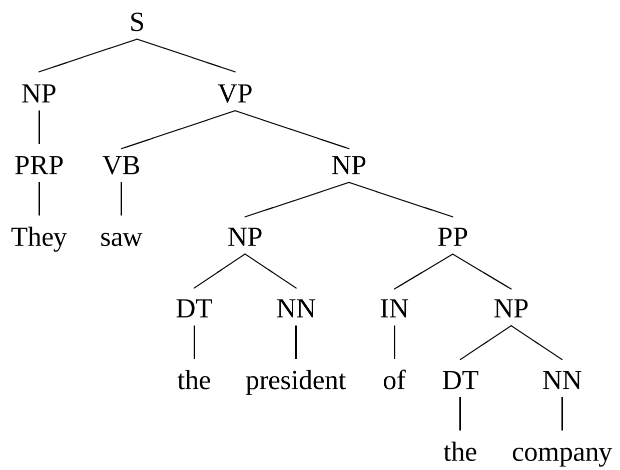
- It assigns probabilities to parses:

```
%| ?- s(P, T, [mary, saw, john],[]).
%
%P = 0.0026999999999999997,
%T = [s,[[np,[[pn,[mary]]]],[vp,[[v,[saw]],[np,[[pn,[john]]]]]]]] ? ;
```

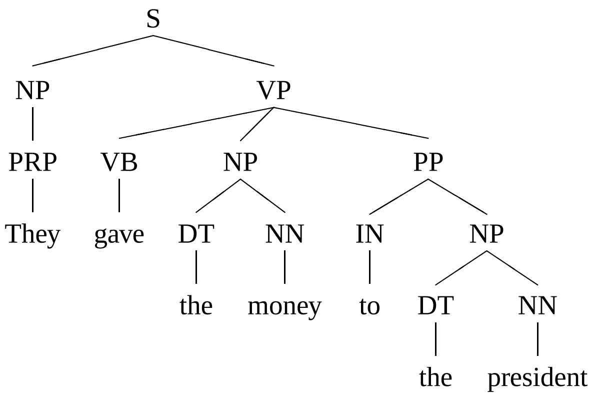

Selecting the best parse

- Basic PCFGs (for parse selection) have a number of problems:
  - They ignore lexical information (such as the likelihood of the verb *find* occuring as an intransitive verbphrase).
  - Are biased towards short flat parses.
  - Make unwarranted independence assumptions.
  - Counts are usually low, so it is important to smooth probabilities.

Selecting the best parse

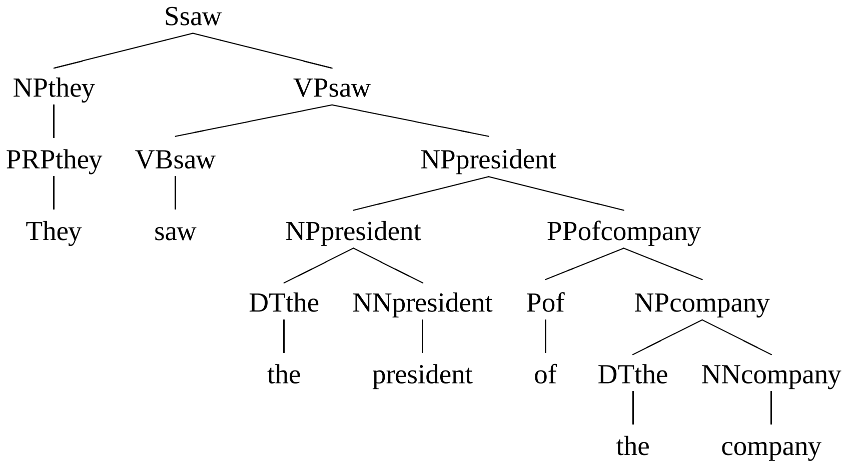


Selecting the best parse



Selecting the best parse

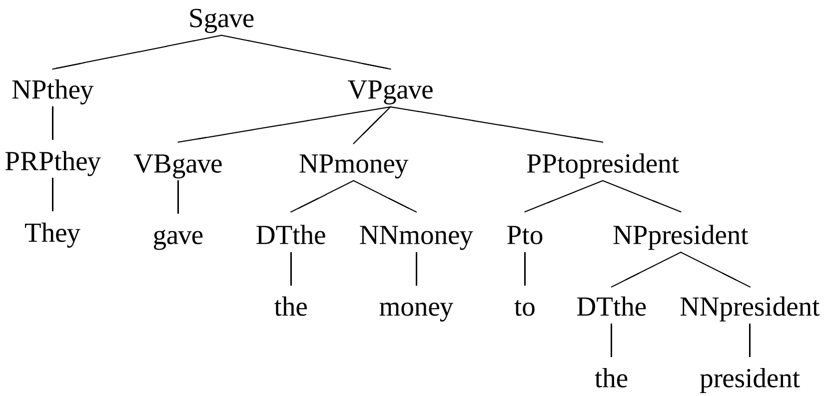
Lexicalisation:



(Magerman 1995, Collins 1997, Charniak 1997):

Selecting the best parse

Lexicalisation:



Why Lexicalize the Model?

- To assign probabilities to such trees, we need to be more careful of our independence assumptions
- A treebank:

```
[S(grows),[NP, grass],[VP,grows]]
[S(grows),[NP, grass],[VP,grows]]
[S(grows),[NP, rice],[VP,grows]]
[S(grow),[NP, bananas],[VP,grow]]
```

Why Lexicalize the Model? (Contd.)

- A naive PCFG is unsound!

	Rule $A \to \alpha$				Count	P ≈ Rel.Frequency|A
r1	S	$\to$	NP	VP	4	1
r2	NP	$\to$	rice		1	1/4
r3	NP	$\to$	grass		2	1/2
r4	NP	$\to$	bananas		1	1/4
r5	VP	$\to$	grows		3	3/4
r6	VP	$\to$	grow		1	1/4

- P(S[NP[grass]VP[grows]]) = 1/2*3/4*1 = 3/8
- P(S[NP[rice]VP[grows]]) = 1/4*3/4*1 = 3/16
- P(S[NP[bananas]VP[grow]]) = 1/4*1/4*1 = 1/16 total Z = 5/8 $\neq$ 1

Why Lexicalize the Model? (Contd.)

- Normalization with respect to Z doesn't help:
- P(S[NP[grass]VP[grows]]) = 1/2*3/4*1*8/5 = 3/5
- P(S[NP[rice]VP[grows]]) = 1/4*3/4*1*8/5 = 3/10
- P(S[NP[bananas]VP[grow]]) = 1/4*1/4*1*8/5 = 1/10 total = 1
- The probabilities are still wrong by inspection
- The problem is the independence assumption: there are non-local dependencies.

Why Lexicalize the Model? (Contd.)

- To build such trees, we separate the grammar into rules which say how heads are passed, and tables of dependency probabilities:
- A lexicalized PCFG:

	Rule $A \to \alpha$				Count	P ≈ Rel.Frequency|A
r0	START	$\to$	S(grows)		3	3/4
r0	START	$\to$	S(grow)		1	1/4
r1	S(H2)	$\to$	NP(H1)	VP(H2)	4	1
r2	NP(rice)	$\to$	rice		1	1
r3	NP(grass)	$\to$	grass		2	1
r4	NP(bananas)	$\to$	bananas		1	1
r5	VP(grows)	$\to$	grows		3	1
r6	VP(grow)	$\to$	grow		1	1

Why Lexicalize the Model? (Contd.)

- $P([S,[NP,grass][VP,grows]]) =$
  $P(S(grows)|START) * P(r1|S(grows)) * P(VP(grows)|S(grows),r1,2) *$
  $P(NP(grass)|S(grows),r1,1)$
  $= 3/4*1*1*2/3 \qquad = 1/2$
- $P([S,[NP,rice][VP,grows]]) =$
  $P(S(grows)|START) * P(r1|S(grows)) * P(VP(grows)|S(grows),r1,2) *$
  $P(NP(rice)|S(grows),r1,1)$
  $= 3/4*1*1*1/3 \qquad = 1/4$
- $P([S,[NP,bananas][VP,grow]]) =$
  $P(S(grow)|START) * P(r1|S(grow)) * P(VP(grow)|S(grow),r1,2) *$
  $P(NP(bananas)|S(grow),r1,1)$
  $= 1/4*1*1*1 \qquad = 1/4$
- These probabilities are correct by observation. (In general they need to be normalized by length of derivation.)
- Head dependencies are also getting us the effect of number agreement.

Lexicalizing the PCFG

- As before:

Rule A → α			Prob(Rule—A)
$START$	$\rightarrow$	$S(X)$	$P(S(X)\|START)$
$S(X)$	$\rightarrow$	$NP(Y)$ $VP(X)$	$P(r1\|S(X)) * P(NP(Y)\|S(X), r1, 1) * P(VP(X)\|S(X), r1, 2)$
$NP(X)$	$\rightarrow$	$Det(Y)$ $N(X)$	$P(r2\|NP(X)) * P(Det(Y)\|NP(X), r2, 1) * P(N(X)\|NP(X), r2, 2)$
$NP(X)$	$\rightarrow$	$PN(X)$	$P(r3\|NP(X))$
$VP(X)$	$\rightarrow$	$V(X)$	$P(r4\|VP(X))$
$VP(X)$	$\rightarrow$	$V(X)$ $NP(Y)$	$P(r5\|VP(X)) * P(V(X)\|VP(X), r5, 1) * P(NP(Y)\|VP(X), r5, 2)$

Lexicalizing the PCFG

- Or in DCG notation:

```
start(P,T) --> s(P0, H, T), {p(start,H,P1),P is P1*P0}.
s(P0, H2, [s,[NP, VP]]) -->                              %Rule r1
   np(P1, H1, NP),
   vp(P2, H2, VP),
   {p1(r1,H2, Prulegivenhead),
    p2(r1, H2, 1, H1, Pargumentheadgivenhead),
    P0 is Prulegivenhead*Pargumentheadgivenheadandrule*P1*P2}.
```

- As before, `*P1*P2` to recursively compute the probability of the entire derivation.

Lexicalizing the PCFG

- `p1/3` is a lookup like `p/2` in the PCFG that computes the probability of a rule $R\ A \rightarrow \alpha$ applying given a parent A headed by H. This is

$$\frac{freq(R|H,A)}{\sum_{R'} freq(R'|H,A)}$$

- `p2/5` is a lookup like `p/2` in the PCFG that computes the probability of non-head daughter at position i in a rule $R\ A \rightarrow \alpha$ being headed by I given a parent A headed by H. This is

$$\frac{freq(I|R,i,H,A)}{\sum_{I'} freq(I'|R,i,H,A)}$$

Lexicalized Parsers and Lexicalized Grammars

- These probabilities can be estimated on the basis of frequency counts in annotated corpora such as the Penn treebank.
- But we clearly have a massive sparse data problem: many combinations of VP head word and subject headword will not have been encountered in a mere million words—or in fact in a lifetime. (Notice that the probability of “love your saw” may well be zero but that of “saw your saw” probably isn’t.)
- Various kinds of “backing off” are required. The crudest is backing off to syntactic type. Notice that in frameworks like CCG/TAG this is already implicit in the lexical categories.
- Other tactics are to back off to some categorization of e.g. nouns as e.g. animate/inanimate.

Selecting the best parse

- Possible to incorporate other information:
  - Grandparents.
  - Subcategorization frames.
  - Left/right branching preferences.
  - Punctuation.
  - etc.

Selecting the best parse

- Dealing with bias:
  - Convert grammar to just binary rules. (CCG already is binary).
  - Normalise parse probabilities by length.
  - Combine parse probabilities using geometric mean.
- None satisfactory.

Selecting the best parse

- Unwarranted independence assumptions:
  - Use larger tree fragments.
  - Use something other than PCFG.
- Focus of current research.

Selecting the best parse

- Lexicalisation / grandparents etc all improve over basic PCFG.
- Still not that good. (Collins 1997 recovers 88% of lexical dependencies.)
- Much current research, many approaches, all very similar in performance.
- Part of the problem is that a million words of hand annotated quite errorful WSJ data may not be enough.

Efficiently finding the best parse

- Use a Viterbi-like approach (Dynamic Programming).
- Can't use straight Viterbi as local ambiguities cause a need to backtrack.
- Exhaustive parsers produce near exponential growth.
- Possible to get linear performance with respect to sentence length with such techniques as beam-search.

Summary

- Stochastic methods give a principled treatment of problems associated with parsing real-world language, especially when combined with linguistically adequate grammar formalisms, as Chomsky anticipated (1957, p.17).
- The problem can be stated as modelling, estimation and approximation.
- Wide coverage parsing is still not that good and there is much room for improvement.
- One of the most promising lines of investigation involves the use of grammar formalisms which are (in a different sense from that used in reference to parsers) *lexicalized* and which capture *long-range dependencies*, like those discussed in this course.

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
Log-linear	**86.6**	**86.3**	**92.5**	**92.1**	**93.6**
Hockenmaier (POS)	83.1	83.5	91.1	91.5	91.5
Log-linear (POS)	**84.8**	**84.5**	**91.4**	**91.0**	**92.5**

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

- 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.
- Collins 1999 reports 90.9% for unlabeled $\langle\rangle$ "surface" dependencies.

Recovery of Long Range Dependencies

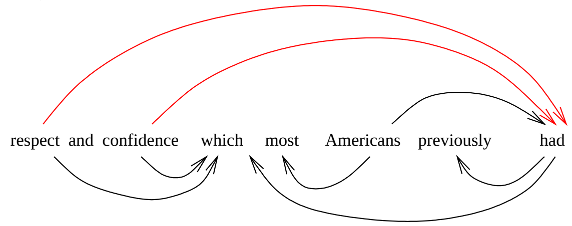
- **Extraction:**
  - Dependencies involving **subject relative pronoun**
    $(\mathbf{NP}\backslash\mathbf{NP})/(\mathbf{S}[\mathbf{dcl}]\backslash\mathbf{NP})$: 98.5%LP, 95.4%LR (99.6%UP, 98.2%UR)
  - Lexical cat. for **embedded subject extraction** (Steedman '96)
    $((\mathbf{S}[\mathbf{dcl}]\backslash\mathbf{NP})/\mathbf{NP})/(\mathbf{S}[\mathbf{dcl}]\bac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

Log-Linear Overall Dependency Recovery

- The C&C parser has state-of-the-art dependency recovery.
- The C&C parser is very fast (≈ 30 sentences per second)
- The speed comes from highly accurate supertagging which is used in a "Best-First increasing" mode (Clark and Curran 2004), and behaves as an "almost parser" (Bangalore and Joshi 1999).
- CCG almost-parsing is why Zettlemoyer and Collins do so well on a small not very ambiguous corpus without having a parser model at all.
- It has been ported to the TREC QA task (Clark *et al.* 2004), and applied to the entailment QA task (Bos *et al.* 2004), using automatically built logical forms.

Recovering Deep or Semantic Dependencies

Clark *et al.* (2002)



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

Is This The Way People Parse Language?

- The existence of garden path sentences might suggest that it is:

(1) a. The horse raced past the barn fell.
    b. The flowers sent for the patient died.

- However, such garden paths can be overridden or induced by *context*:

(2) a. Harry and his brothers were racing their horses all over the farm to see which was fastest.
    b. The horse raced past the barn fell.

- The short term effect of context cannot be captured in global higher-order statistics over corpora. It requires the involvement of knowledge representations and inference.

- It remains possible, even likely, that such knowledge processes are probabilistic in nature.

Competence and Performance Revisited

- Theories of this kind are entirely modular, and the grammar can be identified independently of the statistical model that assigns probabilities to strings and the algorithm that searches for parses.

- Their principles generalize quite readily to more powerful grammars like CCG, TAG, HPSG, LFG, with one important qualification concerning the multiple dependencies captured in these grammars, which are not easy to deal with in generative models (Abney 1997).

- However, in the end Competence and Performance are a package deal: Grammars must not only be linguistically adequate, but must also support parsing mechanisms that capture human performance.

References

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

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.

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, 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.

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