# Three Generative, Lexicalised Models for Statistical Parsing

Michael Collins
University of Pennsylvania

# BACKGROUND

- $S$ = a sentence.
- $T$ = a parse tree for the sentence.
- A statistical model estimates $P(T|S)$.
- The best parse under the model is then

$$T_{best} = argmax_T P(T|S)$$

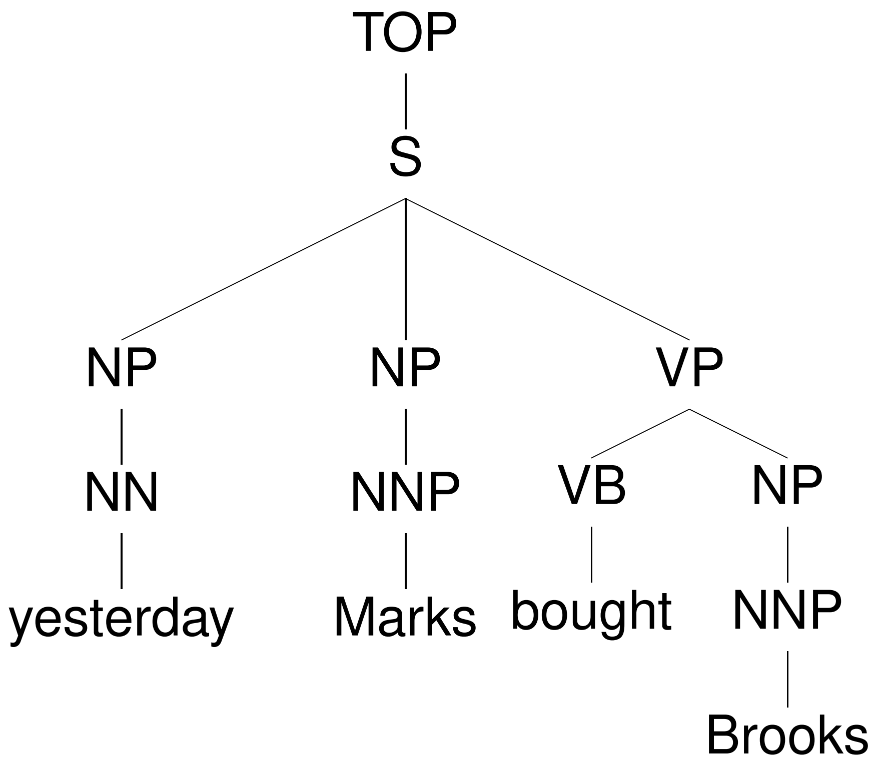
- Parameters trained from an annotated corpus (e.g. the Penn Treebank)

# Probabilistic Context Free Grammars

- $T_{best} = \text{arg max}_T \, \mathcal{P}(T \mid S) = \text{arg max}_T \, \frac{\mathcal{P}(T,S)}{\mathcal{P}(S)} = \text{arg max}_T \, \mathcal{P}(T,S)$
- For a tree with $n$ context-free rules $LHS_i \Rightarrow RHS_i$

$$\mathcal{P}\big(T,S\big) = \prod_{i=1..n} \mathcal{P}\big(RHS_i \mid LHS_i\big)$$

| Rule | Probability |
|---|---|
| VP $\Rightarrow$ VB NP | 0.6 |
| VP $\Rightarrow$ VB NP PP | 0.2 |
| VP $\Rightarrow$ VB NP NP | 0.2 |




$$
\begin{aligned}
\text{PROB} = \quad & P(\text{TOP} \rightarrow \text{S}) \\
& \times P(\text{S} \rightarrow \text{NP NP VP}) && \times P(\text{VP} \rightarrow \text{VB NP}) \\
& \times P(\text{NP} \rightarrow \text{NN}) && \times P(\text{NP} \rightarrow \text{NNP}) \\
& \times P(\text{NP} \rightarrow \text{NNP}) && \times P(\text{NN} \rightarrow yesterday) \\
& \times P(\text{NNP} \rightarrow Marks) && \times P(\text{VB} \rightarrow bought) \\
& \times P(\text{NNP} \rightarrow Brooks)
\end{aligned}
$$

## The Importance of Lexical Information

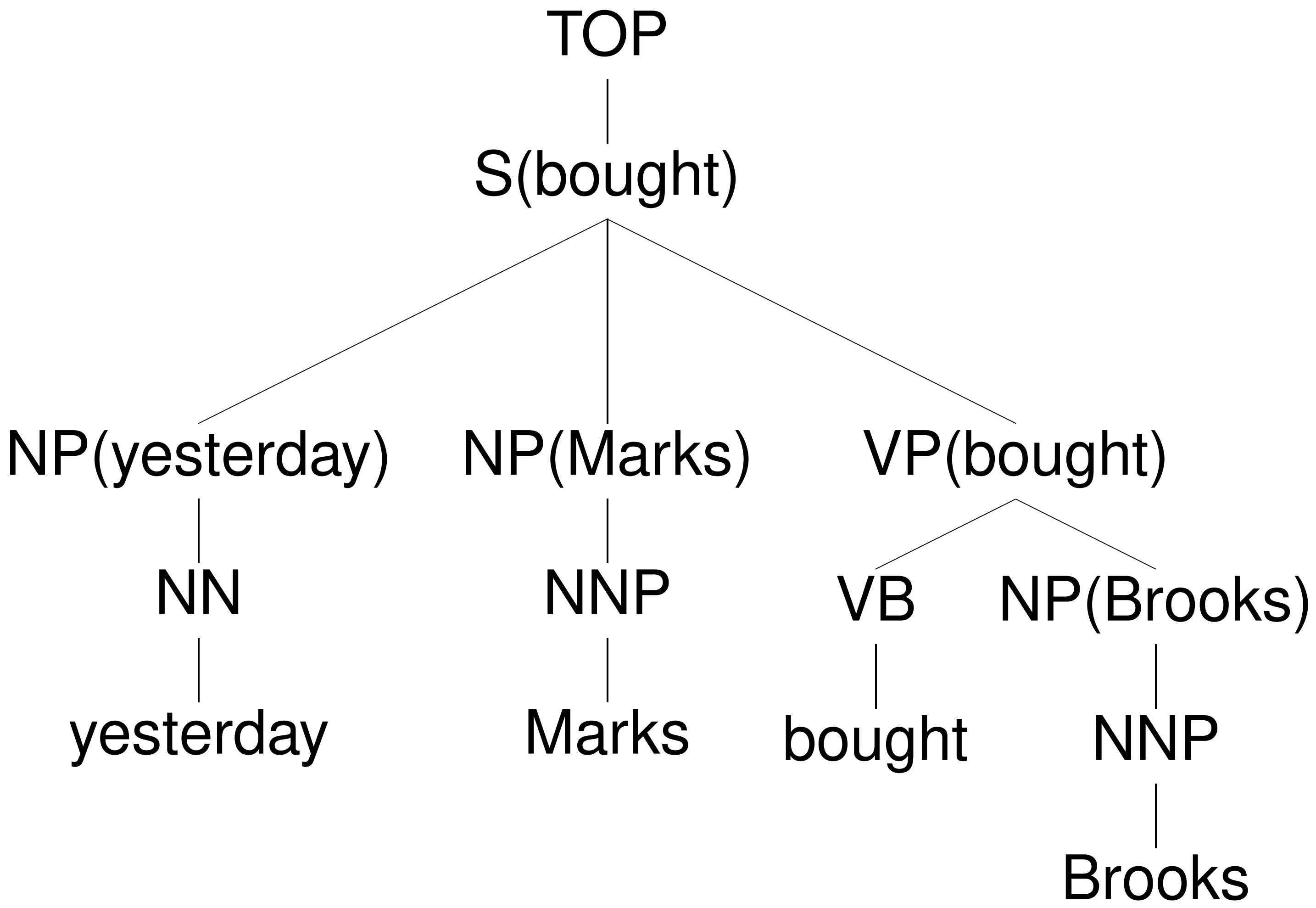
- Hindle and Rooth 92, Prepositional Phrase Attachment

- Magerman 95, Jelinek et al 94, SPATTER. $P(T|S) = \Pi_{i=1..n} \mathcal{P}(d_i|History)$, where $d_1 ... d_n$ are the decisions made by the parser in building the tree.

- Collins 96. $P(T|S) = \Pi_{i=1..n} \mathcal{P}(d_i|S)$, where $d_1 ... d_n$ are the dependencies in the tree.

- Alshawi 96, Eisner 96. Probabilistic head automata, statistical dependency parsing.

- Lafferty, Sleator and Temperley. Link Grammars.


TOP
S(bought)
NP(yesterday)
NP(Marks)
VP(bought)
NN
NNP
VB
NP(Brooks)
yesterday
Marks
bought
NNP
Brooks

- Model 1: a lexicalised, generative model
- Model 2: The complement/adjunct distinction and subcategorisation probabilities
- Model 3: Wh-movement
- Results on WSJ text (training and testing on the Penn treebank)
- Conclusions and future Work

# LEXICALISED CONTEXT-FREE GRAMMARS

LHS $\Rightarrow$ RHS

$$P(h) \Rightarrow L_m(l_m) \ldots L_1(l_1)\ H(h)\ R_1(r_1) \ldots R_n(r_n)$$

For example

S(bought) $\Rightarrow$ NP(yesterday) NP(Marks) VP(bought)

$P(h) \Rightarrow L_2(l_2) \quad L_1(l_1) \quad H(h)$

## HOW TO ESTIMATE P(RHS|LHS)?

- Maximum likelihood estimate

$$P(RHS|LHS) = \frac{C(RHS, LHS)}{C(LHS)}$$

e.g.

$$\frac{C(\text{S(bought)} \Rightarrow \text{NP(yesterday) NP(Marks) VP(bought)})}{C(\text{S(bought)})}$$

BUT sparse data – 40,000 sentences from WSJ have around 12,000 rules even without lexical information

# Decomposing P(RHS|LHS)

1. Generate head label, with probability

$$\mathcal{P}_{\mathcal{H}}(H \mid P, h) \qquad \text{e.g. } \mathcal{P}_{\mathcal{H}}(\text{VP} \mid \text{S, bought})$$

2. Generate right modifiers with probability

$$\prod_{i=1..m+1} \mathcal{P}_{\mathcal{R}}(R_i(r_i)|\text{P,h,H}) \quad \text{e.g. } \mathcal{P}_{\mathcal{R}}(\text{NP(Brooks)}|\text{VP, VB, bought})$$

3. Generate left modifiers with probability

$$\prod_{i=1..n+1} \mathcal{P}_{\mathcal{L}}(L_i(l_i)|\text{P,h,H}) \qquad \text{e.g. } \mathcal{P}_{\mathcal{L}}(\text{NP(Marks)}|\text{S,VP,bought})$$

```
 S(bought) ⇒
NP(yesterday) NP(Marks) VP(bought)
```

$\mathcal{P}_{\mathcal{H}}(\text{VP} \mid \text{S,bought}) \times$

$\mathcal{P}_{\mathcal{L}}(\text{NP(Marks)} \mid \text{S,VP,bought}) \times$

$\mathcal{P}_{\mathcal{L}}(\text{NP(yesterday)} \mid \text{S,VP,bought}) \times$

$\mathcal{P}_{\mathcal{L}}(\text{STOP} \mid \text{S,VP,bought}) \times$

$\mathcal{P}_{\mathcal{R}}(\text{STOP} \mid \text{S,VP,bought})$

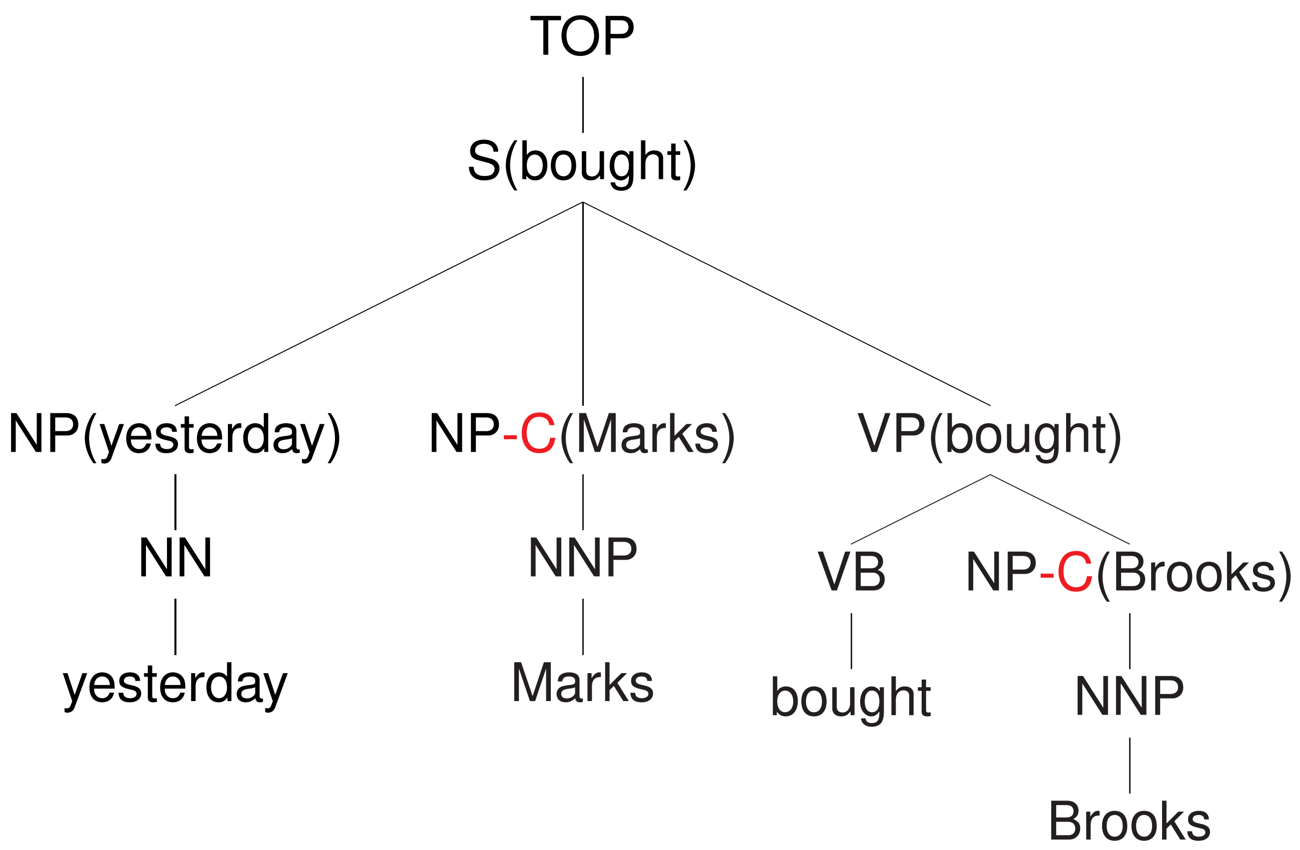
# Other Details in the Paper

- Estimation
- Parsing. CKY-style dynamic programming chart parser.
- POS tagging is part of the statistical model.
- Conditioning on the “distance” from the head (allows preferences for right or left branching structures).

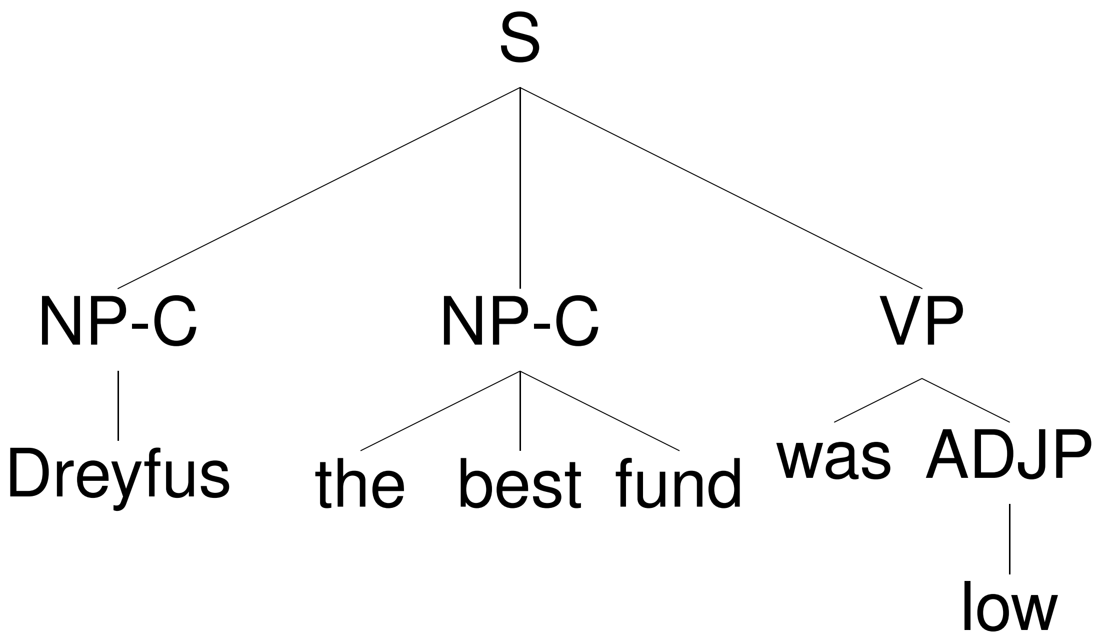
## MODEL 2: THE COMPLEMENT/ADJUNCT DISTINCTION

- Needed for extraction of predicate-argument structure.
- Needed to improve parse accuracy.

# The Complement/Adjunct Distinction and Predicate-Argument Structure



# THE COMPLEMENT/ADJUNCT DISTINCTION AND BAD INDEPENDENCE ASSUMPTIONS




$$\mathcal{P}(\text{NP-C(Dreyfus), NP-C(fund)} \mid \text{S,VP,was}) =$$
$$\mathcal{P}(\text{NP-C(Dreyfus)} \mid \text{S,VP,was})$$
$$\times\mathcal{P}(\text{NP-C(fund)} \mid \text{S,VP,was})$$

# Model 2: Introducing Subcategorisation Probabilities

1. Generate head label, with probability

$$\mathcal{P}_{\mathcal{H}}(H \mid P, h)$$

2. Choose left and right subcat frames with probabilities

$$\mathcal{P}_{LC}(LC \mid P, h, H) \quad \mathcal{P}_{RC}(RC \mid P, h, H)$$

e.g. $\mathcal{P}_{RC}(\{\text{NP-C,NP-C}\} \mid \text{VP, gave, VB})$

3. Generate right modifiers with probability

$$\prod_{i=1..m+1} \mathcal{P}_{\mathcal{R}}(R_i(r_i) \mid \text{P, h, H, distance, RC})$$

4. Generate left modifiers with probability

$$\prod_{i=1..n+1} \mathcal{P}_{\mathcal{L}}(L_i(l_i) \mid \text{P, h, H, distance, LC})$$

# Some Important Properties

- $\mathcal{P}_{\mathcal{L}}(\text{STOP} \mid \text{....LC}) = 0$ if $LC$ is non-empty.
- $\mathcal{P}_{\mathcal{L}}(\text{X-C} \mid \text{....LC}) = 0$ if X-C $\notin LC$
- Independence assumptions are improved. For example $\mathcal{P}_{LC}(\{\text{NP-C, NP-C}\} \mid \text{S, VP, bought})$ is very low.

S(bought) $\Rightarrow$ NP(yesterday) NP-C(Marks) VP(bought)

$\mathcal{P}_{\mathcal{H}}$(VP | S,bought)$\times$

$\mathcal{P}_{LC}$($\{$NP-C$\}$ | S,bought,VP)$\times$

$\mathcal{P}_{RC}$($\{\}$ | S,bought,VP)$\times$

$\mathcal{P}_{\mathcal{L}}$(NP-C(Marks) | S, VP, bought, $\{$NP-C$\}$)$\times$

$\mathcal{P}_{\mathcal{L}}$(NP(yesterday) | S, VP, bought, $\{\}$)$\times$

$\mathcal{P}_{\mathcal{L}}$(STOP | S, VP, bought, $\{\}$)$\times$

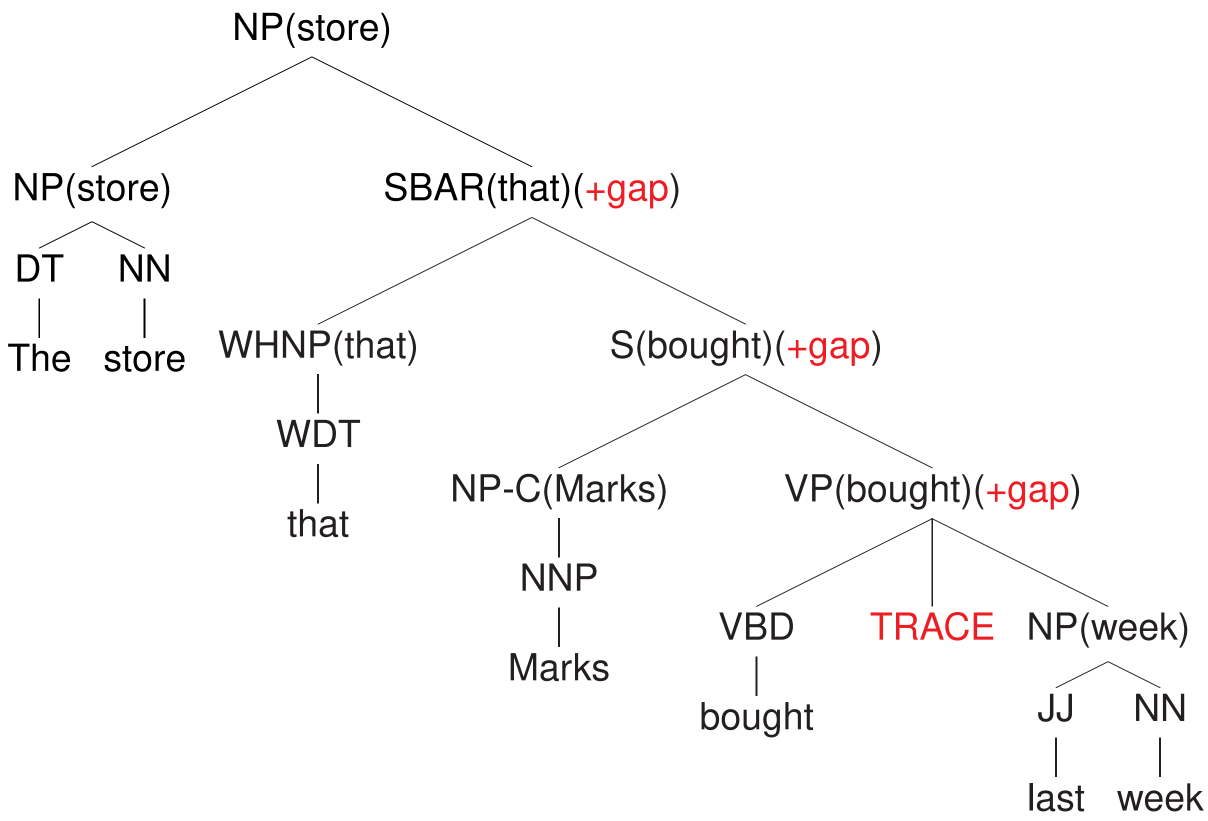
$\mathcal{P}_{\mathcal{R}}$(STOP | S, VP, bought, $\{\}$)

# MODEL 3: TRACES AND WH-MOVEMENT

Typical examples:

- The store (SBAR that TRACE bought Brooks Brothers)
- The store (SBAR that Marks bought TRACE)
- The store (SBAR that Marks bought Brooks Brothers from TRACE)

# A GPSG-Style Treatment

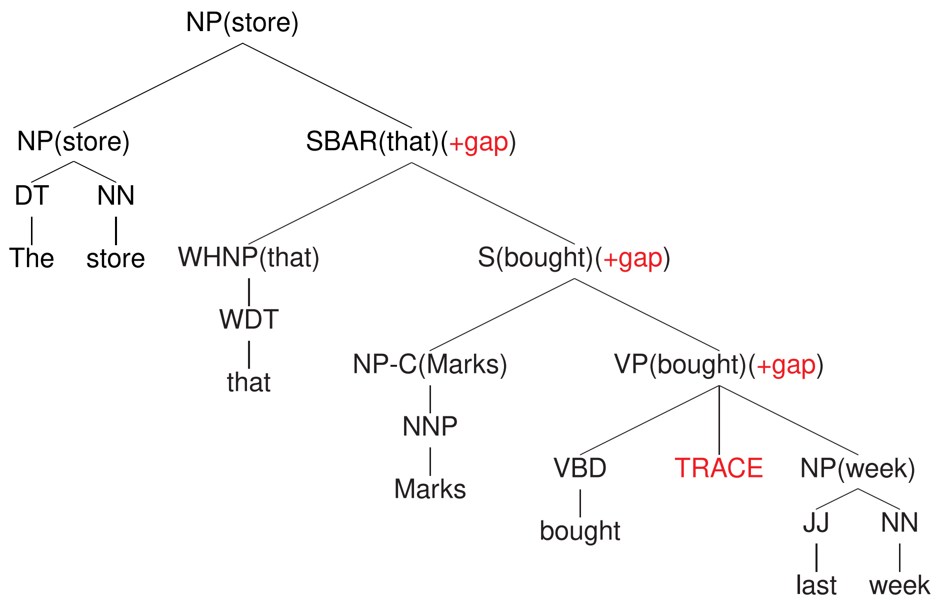


# Three Ways a Gap can Move Down the Tree

**Head** The gap is passed to the head of the phrase.

**Left** The gap is passed on recursively to one of the left modifiers of the head, or is discharged as a $trace$ argument to the left of the head.

**Right** The gap is passed on recursively to one of the right modifiers of the head, or is discharged as a $trace$ argument to the right of the head.




| | | | | |
|---|---|---|---|---|
| NP | -> | NP | SBAR(+gap) | |
| SBAR(+gap) | -> | WHNP | S-C(+gap) | RIGHT |
| S(+gap) | -> | NP-C | VP(+gap) | HEAD |
| VP(+gap) | -> | VB | TRACE NP | RIGHT |

1. Generate head label, with probability

$$\mathcal{P}_{\mathcal{H}}(H \mid P, h)$$

2. Choose direction for the gap,

$$\mathcal{P}_{\mathcal{G}}(\text{Direction} \mid P, h)$$

3. Choose left and right subcat frames with probabilities

$$\mathcal{P}_{LC}(LC \mid P, h, H) \quad \mathcal{P}_{RC}(RC \mid P, h, H)$$

4. Generate left and right modifiers.

# HOW THESE AFFECT THE MODEL

**Head** Left/Right modifers generated as normal. The head passes the gap further down the tree.

**Left** $+gap$ is added to the *LC*.

**Right** $+gap$ is added to the *RC*.

# AN EXAMPLE

SBAR(that)(+gap) -> WHNP(that) S-C(bought)(+gap)

$$\mathcal{P}_{\mathcal{H}}(\texttt{WHNP} \mid \texttt{SBAR,that}) \times$$
$$\mathcal{P}_{G}(\texttt{Right} \mid \texttt{SBAR,WHNP,that}) \times$$
$$\mathcal{P}_{LC}(\{\} \mid \texttt{SBAR,WHNP,that}) \times$$
$$\mathcal{P}_{RC}(\{\texttt{S-C}\} \mid \texttt{SBAR,WHNP,that}) \times$$
$$\mathcal{P}_{\mathcal{R}}(\texttt{S-C(bought)(+gap)} \mid \texttt{SBAR,WHNP,that}, \{\texttt{S-C,+gap}\}) \times$$
$$\mathcal{P}_{\mathcal{R}}(\texttt{STOP} \mid \texttt{SBAR,WHNP,that}, \{\}) \times$$
$$\mathcal{P}_{\mathcal{L}}(\texttt{STOP} \mid \texttt{SBAR,WHNP,that}, \{\})$$

# Island Constraints

- Extraction allowed from complements, not adjuncts:

$$\mathcal{P}_{\mathcal{L}}(\text{SBAR-C(that)(+gap)} \mid \text{VP, VB, said}, \{\text{SBAR-C, +gap}\}) \quad \text{HIGH}$$

$$\mathcal{P}_{\mathcal{L}}(\text{SBAR(because)(+gap)} \mid \text{VP, VB, gave}, \{\text{+gap}\}) \quad \text{LOW}$$

- No extraction from subject NPs:

$$\mathcal{P}_{\mathcal{L}}(\text{TRACE} \mid \text{S, VP, bought}, \{\text{NP-C, +gap}\}) \quad \text{HIGH}$$

$$\mathcal{P}_{\mathcal{L}}(\text{NP-C(+gap)} \mid \text{S, VP, bought}, \{\text{NP-C, +gap}\}) \quad \text{LOW}$$

- Extraction from PP adjuncts? "the man he took the picture of TRACE", "the day she arrived on TRACE":

$$\mathcal{P}_{\mathcal{L}}(\text{PP(of)(+gap)} \mid \text{NP, NN, picture}, \{+gap\})$$

$$\mathcal{P}_{\mathcal{L}}(\text{PP(on)(+gap)} \mid \text{VP, VB, arrived}, \{+gap\})$$

# RESULTS

| MODEL | ≤ 40 Words (2245 sentences) | | | | | ≤ 100 Words (2416 sentences) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LR | LP | CBs | 0 CBs | ≤ 2 CBs | LR | LP | CBs | 0 CBs | ≤ 2 CBs |
| Magerman 95 | 84.6% | 84.9% | 1.26 | 56.6% | 81.4% | 84.0% | 84.3% | 1.46 | 54.0% | 78.8% |
| Collins 96 | 85.8% | 86.3% | 1.14 | 59.9% | 83.6% | 85.3% | 85.7% | 1.32 | 57.2% | 80.8% |
| Model 1 | 87.4% | 88.1% | 0.96 | 65.7% | 86.3% | 86.8% | 87.6% | 1.11 | 63.1% | 84.1% |
| Model 2 | 88.1% | 88.6% | 0.91 | 66.5% | 86.9% | 87.5% | 88.1% | 1.07 | 63.9% | 84.6% |
| Model 3 | 88.1% | 88.6% | 0.91 | 66.4% | 86.9% | 87.5% | 88.1% | 1.07 | 63.9% | 84.6% |

- Model 1: Simple generative with independent children
- Model 2: Model 1 + complement/adjunct distinction
- Model 3: Model 2 + Wh-movement

# RESULTS FOR TRACES

For a trace to be “correct”:

1. It must be an argument to the correct head.
2. It must be dominated by the correct non-terminal.
3. It must be correctly preceding or following the head.

e.g. (bought, VP, following)

Results:

- 93.3% precision, 90.1% precision on 436 cases
- 97.1/98.2% for string-vacuous extraction from subject position (342 cases)
- 76/60.6% for longer distance movement (94 cases)

# Future Work

- Conditioning on more context $\Rightarrow$ more sophisticated estimation methods.

- Syntactic/semantic word class information.

- Unsupervised training?

- Applications? Information retrieval and extraction. Speech synthesis and recognition. Machine translation.

# CONCLUSIONS

- 2.3% improvement over last year's results (16% relative reduction in the error rate over Collins 96, 22% over Magerman 95).

- Probabilistic, lexicalised treatment of subcategorisation, adding the complement-adjunct distinction to the parser's output.

- Probabilistic treatment of wh-movement, adding trace annotations to the parser's output.