



Extended PCFG Parsing

Erhard W. Hinrichs and Sandra Kübler

SfS-CL

Eberhard-Karls-Universität Tübingen



- unlexicalized parsing: Klein and Manning (2003)
- lexicalized parsing, maximum-entropy inspired: Charniak (2000)
- sister-head dependencies: Dubey and Keller (2003)



Why Unlexicalized Parsing?



- lexicalized parsing = attaching head word to mother node
- leads to huge number of different rules \Rightarrow data sparseness problems
- Klein and Manning: use unlexicalized model that outperforms lexicalized ones
- linguistically motivated tree transformations



- training data: sections 2-21
- development data: section 22 (first 20 files)
- test data: section 23
- unsmoothed maximum-likelihood estimates as rule probabilities
- baseline F-score: 72.62

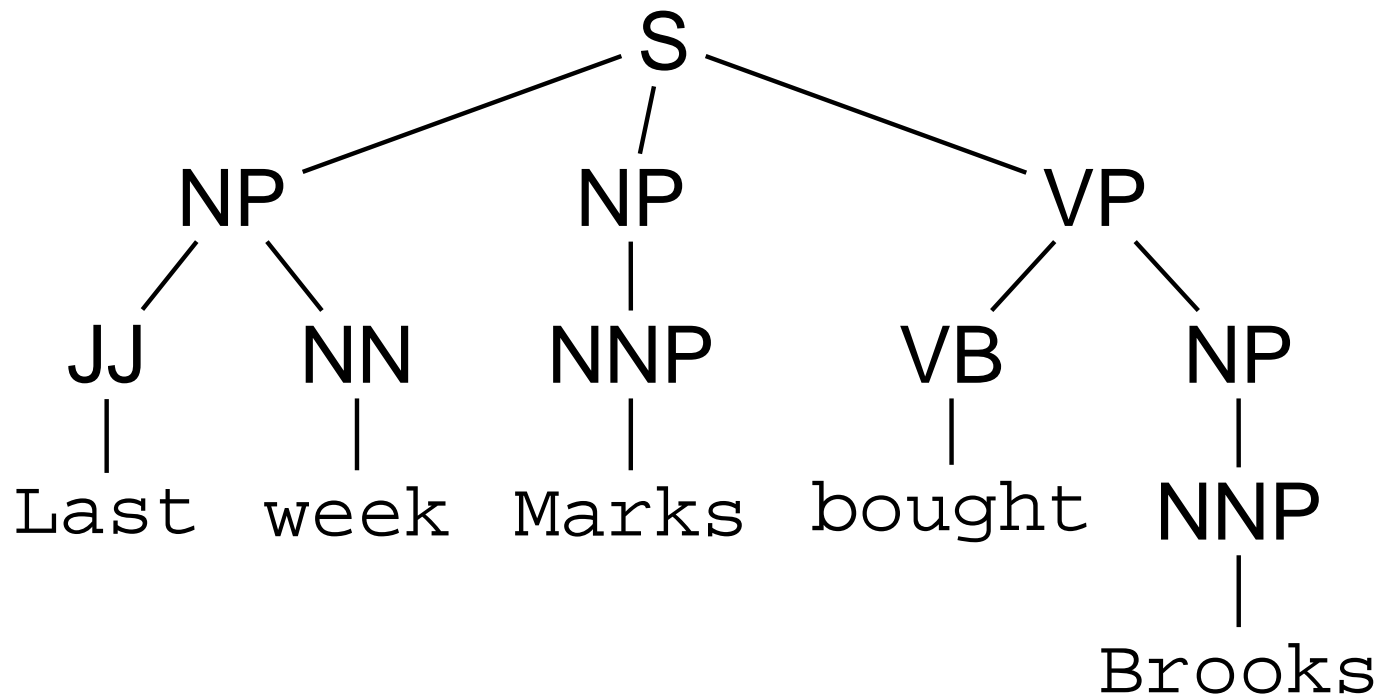


- borrow idea from Johnson
- encode parent / grandparent in node
- adds context from outside the phrase



Parent Annotation

- borrow idea from Johnson
- encode parent / grandparent in node
- adds context from outside the phrase

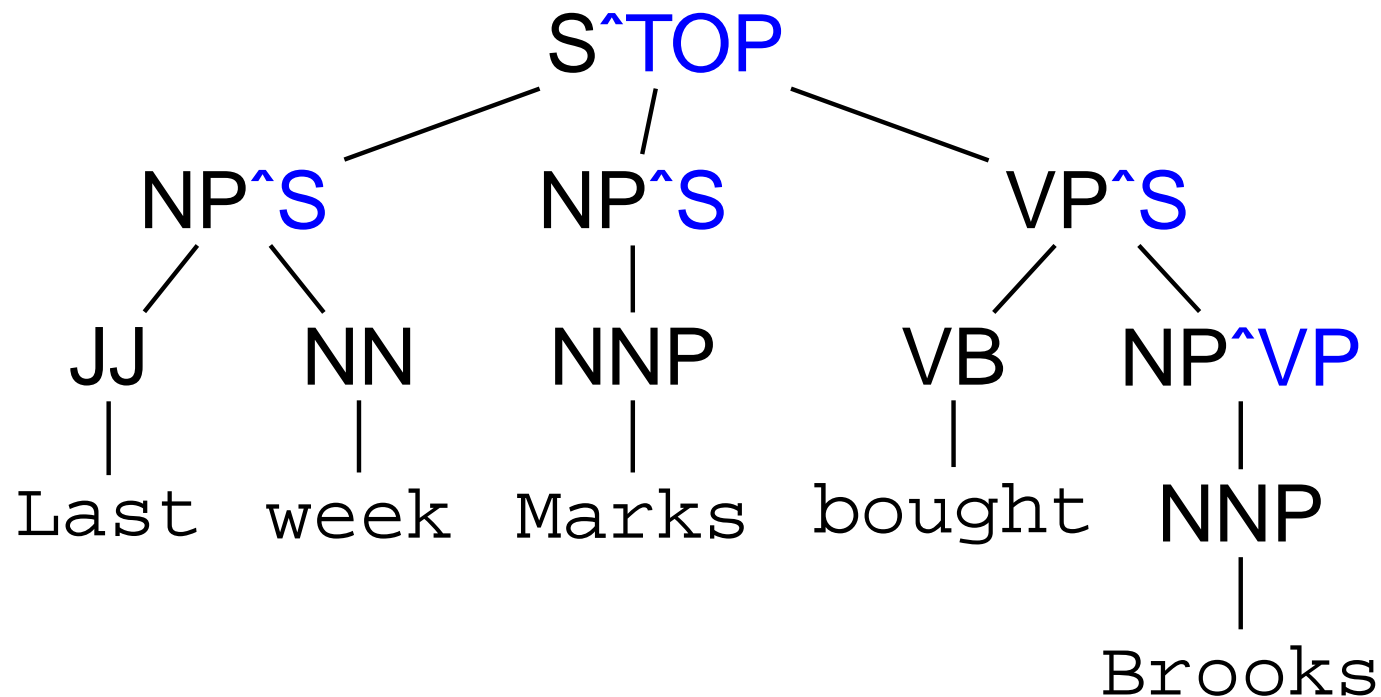




Parent Annotation

- borrow idea from Johnson
- encode parent / grandparent in node
- adds context from outside the phrase

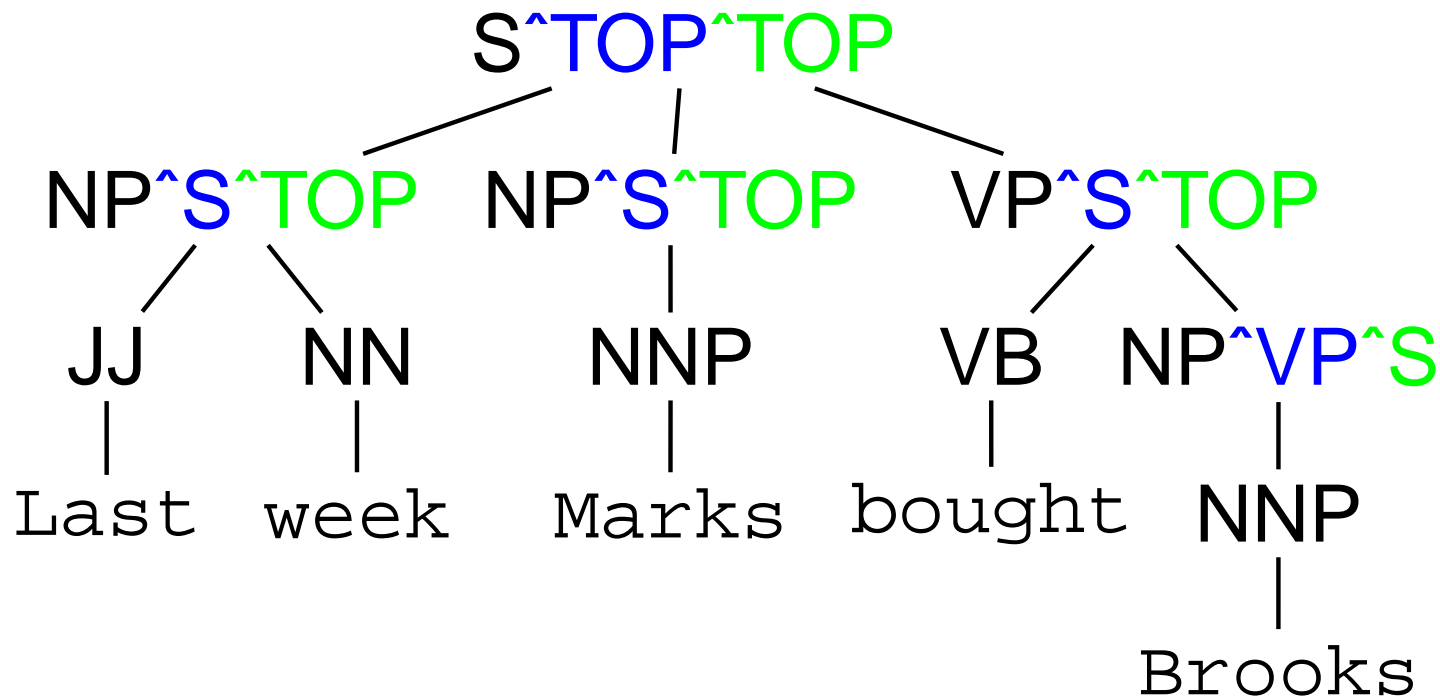
parent:





Parent Annotation

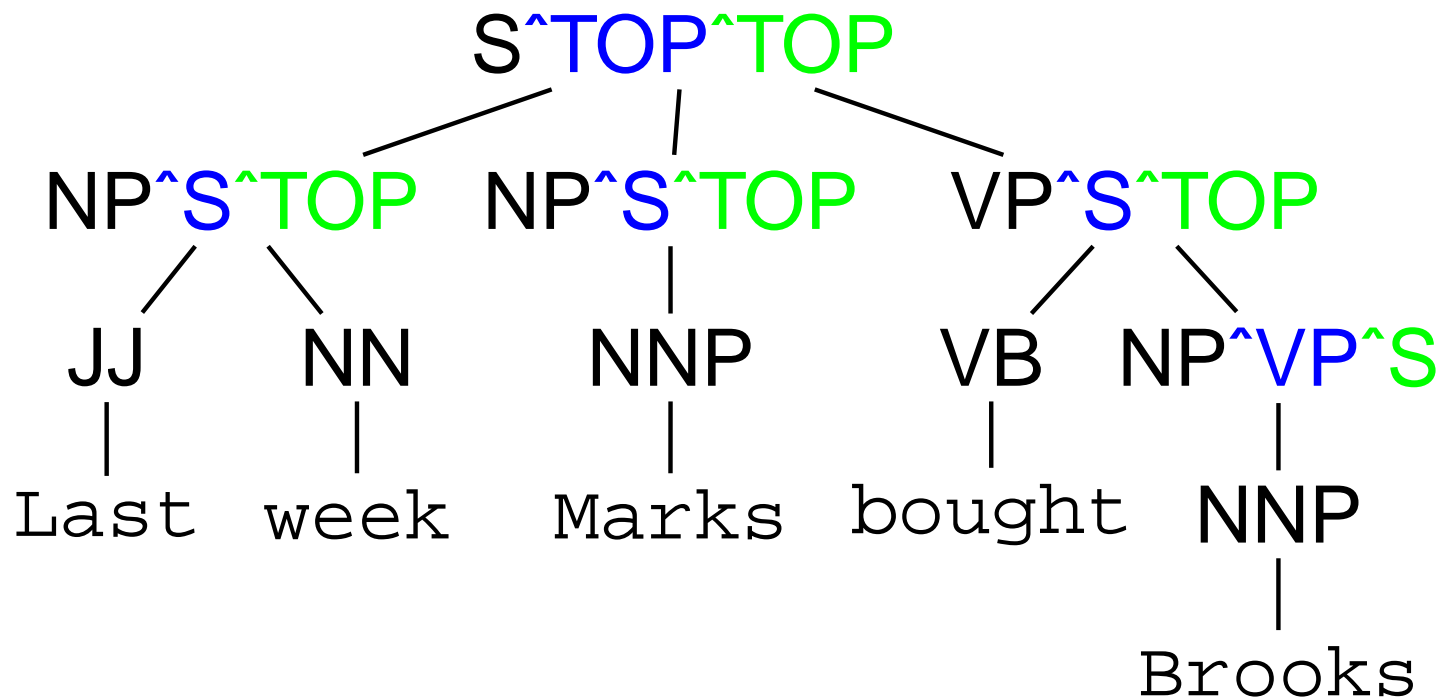
- borrow idea from Johnson
- encode parent / grandparent in node
- adds context from outside the phrase





Parent Annotation

- borrow idea from Johnson
- encode parent / grandparent in node
- adds context from outside the phrase



- adding all parents: increases F-score from 72.62 to 78.72



- many rules which occur only once or not at all in training data
- often very similar rules are in training data
- e.g. NP \rightarrow DET ADJ N occurs
but NP \rightarrow DET ADJ ADJ N does not



- many rules which occur only once or not at all in training data
- often very similar rules are in training data
- e.g. $NP \rightarrow DET\ ADJ\ N$ occurs
but $NP \rightarrow DET\ ADJ\ ADJ\ N$ does not
- solution: markovization: split rule into parts,
generate head child first, then right modifiers,
then left modifiers
- e.g. $VP \rightarrow VB\ NP\ PP$
- head rule: $\langle VP : [VBZ] \rangle \rightarrow VBZ$
 $\langle \rangle$: not complete yet, intermediate symbol
 $[VBZ]$: head of the rule



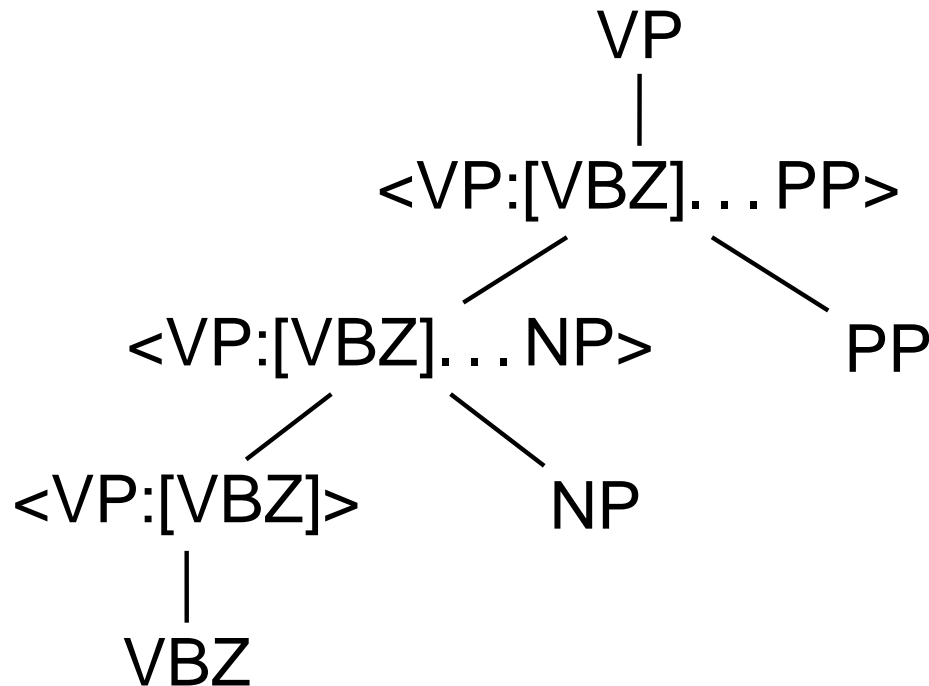
Horizontal Markovization (2)



- head rule: $\langle VP : [VBZ] \rangle \rightarrow VBZ$
- first right sibling:
 $\langle VP : [VBZ] \dots NP \rangle \rightarrow \langle VP : [VBZ] \rangle NP$
- second right sibling:
 $\langle VP : [VBZ] \dots PP \rangle \rightarrow \langle VP : [VBZ] \dots NP \rangle PP$



- head rule: $\langle VP : [VBZ] \rangle \rightarrow VBZ$
- first right sibling:
 $\langle VP : [VBZ] \dots NP \rangle \rightarrow \langle VP : [VBZ] \rangle NP$
- second right sibling:
 $\langle VP : [VBZ] \dots PP \rangle \rightarrow \langle VP : [VBZ] \dots NP \rangle PP$
- covers rules:
 $VP \rightarrow VBZ$; $VP \rightarrow VBZ NP$; $VP \rightarrow VBZ NP PP$
- tree:





Calculating Probabilities:

- 0th order Markov processes:
condition only on the head
- 1st order Markov processes:
condition on the head and on the next inner
sibling; exception: first left sibling is conditioned
on the last right sibling
- ...



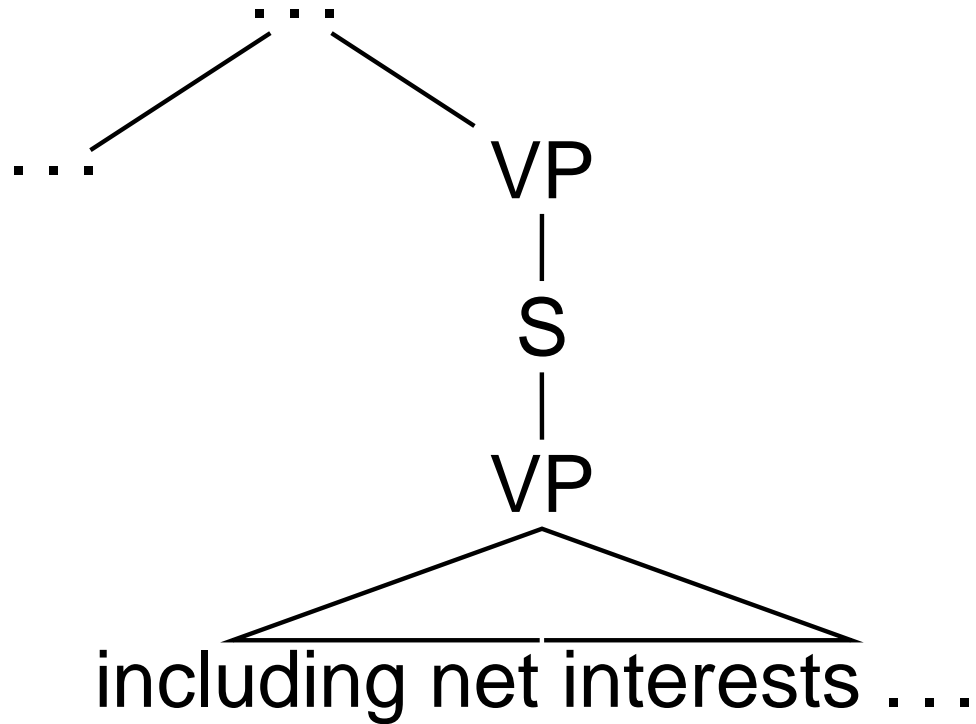
Calculating Probabilities:

- 0th order Markov processes:
condition only on the head
- 1st order Markov processes:
condition on the head and on the next inner
sibling; exception: first left sibling is conditioned
on the last right sibling
- ...
- without parenting: increases F-score from 72.62
to 73.46
with parenting: increases F-score from 78.72 to
79.74



Marking Unaries

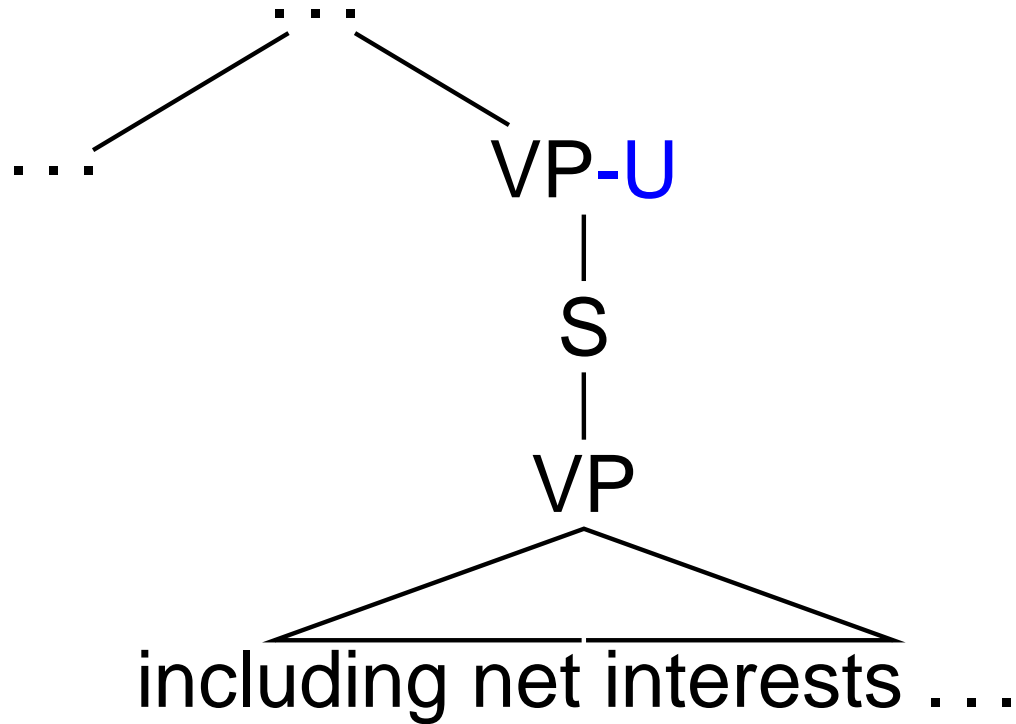
- mark nodes which have only one child
- e.g. from





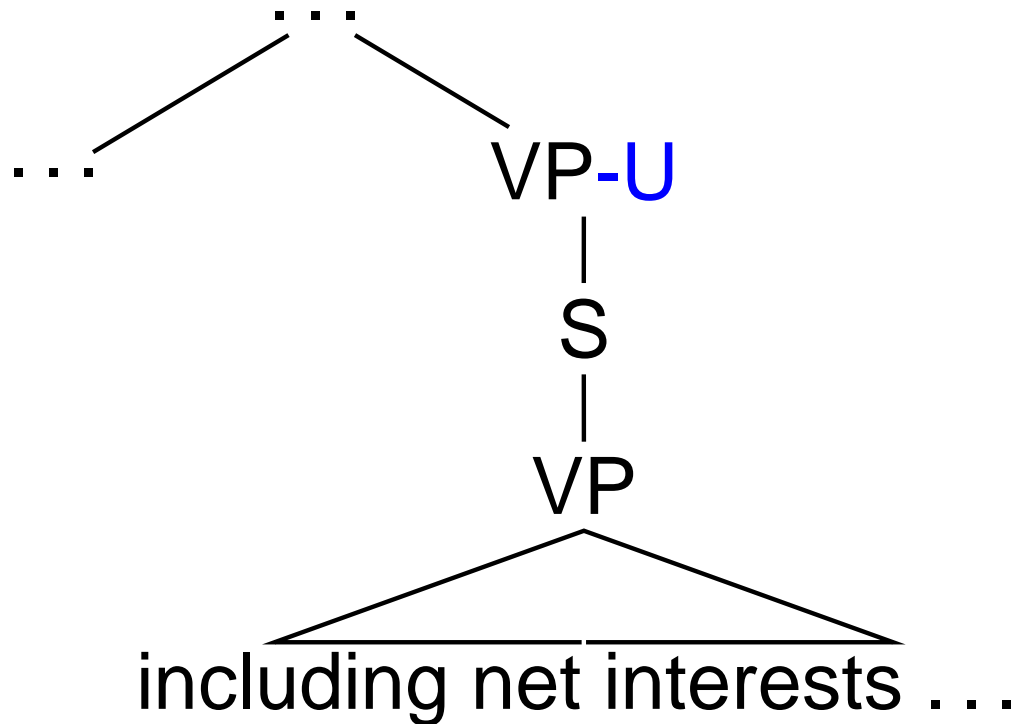
Marking Unaries

- mark nodes which have only one child
- e.g. to





- mark nodes which have only one child
- e.g. to



- increases F-score from 77.77 (vertical ≤ 2 , horizontal ≤ 2) to 78.32



- mark POS tags whether they have siblings
- helps distinguish DT determiners (siblings) and DT demonstratives (no siblings)
- helps distinguish adverbs (as well vs. also)
- increases F-score from 78.32 to 78.86



- extend POS tag with parent node
- helps distinguish IN prepositions (e.g. `of`, `in`, `from`), IN subordinating conjunctions (e.g. `while`, `as`, `if`), and IN complementizers (e.g. `that`, `for`)
- attach word lemma
- helps with auxiliaries and conjunctions
- increases F-score from 78.86 to 81.81



- adding all functional labels (e.g. location PPs, temporal adverbial phrases) has detrimental effects
- adding selected functions such as temporal NPs and gapped clauses helps
- marking possessive NPs and finiteness in VPs helps
- adding distance information implicitly helps: base NPs, in nodes dominating a VP, and NPs with another NP in right periphery
- final F-score: 87.04



- PCFGs are too restricted, need more context about outside structure, less inside structure
- adding linguistic insights helps
- most extensions apart from parenting and markovization are dependent on language and annotation scheme!
- absolute gain: 14.42



- uses markovization, maximum-entropy inspired probabilistic model
- generative model:

$$p(\pi) = \prod_{c \in \pi} p(t|l, H) \times p(h|t, l, H) \times p(e|l, t, h, H)$$

π = parse tree

t = preterminal of constituent

l = label of constituent, e.g. NP, VP

H = history (to be determined)

h = lexical head of constituent

e = expansion, i.e. further constituents



- markovization: how to calculate e
- for each expansion, one head constituent M is determined out of right-hand side
- remaining sisters are added separately, dependent on previously generated sisters
- expansion e :

$$l \rightarrow \Delta L_m \dots L_1 M R_1 \dots R_n$$

- second order Markov grammar:
 $p(L_2|L_1, M, l, t, h, H)$



- problem: which information to use for conditioning the probabilities
- maximum entropy assumption: model only what you know, keep everything else uniform
- need to decide on set of “features”
- Charniak’s assumption: features are binary
- advantages:
 - probability factored out into features: easy to change
 - ME does not assume independence of features



- standard Penn treebank setup: sections 2-21 for training, section 24 for development, section 23 for testing
- condition t on label and preterminal of the parent, on the label of the preceding sister, and the label and lexical head of the grandparent
- labeled precision: 90.1%, labeled recall: 90.1%
- basic system: (Charniak 1997) without soft clustering and unsupervised learning: treebank grammar, guesses only lexical head



improvements:

- guess preterminal first: +2%
- mark noun phrase and verb phrase coordination: +0.6%
- standard interpolation smoothing: worse results
- maximum-entropy inspired, adding grandparent and left sibling label: +0.4%
- first order Markov: worse results
- second order Markov: +0.4%
- third order Markov: +0.3%



- parse German
- treebank: Negra; German newspaper texts, ca. 20.000 sentences
- differences between German and English: has more morphology and freer word order
- word order: position of verbs is fixed, all other phrases are ordered based on preferences rather than on rules
- Negra annotation: extremely flat phrases: e.g. no NP in PP
- Negra in Penn treebank format: contains empty categories for traces to describe long-distance relationships



German Word Order Example



canonical word order:

Er hat das Buch gestern gelesen.

He has the book yesterday read.

'He has read the book yesterday.'



German Word Order Example



canonical word order:

Er hat das Buch gestern gelesen.

He has the book yesterday read.

'He has read the book yesterday.'

other possibilities:

Er hat gestern das Buch gelesen.

Gestern hat er das Buch gelesen.

Das Buch hat er gestern gelesen.



- test whether lexicalization improves results
- models: standard unlexicalized (lopar), standard lexicalized (Carroll and Rooth 1998), Collins parser
- setup: 18.000 sentences for training, 1.000 sentences for development, 1.000 sentences for testing
- experiments with automatically tagged text and with treebank POS tags
- for Collins parser: empty categories removed from training data
- modified unlexicalized and lexicalized parsing: add grammatical functions



Results for Treebank POS Tags



	lab. recall	lab. precision	coverage
unlexicalized	72.99	70.00	95.25
unlex. + GF	81.14	78.37	65.39
lexicalized	70.79	63.38	95.25
lex. + pool	71.74	64.73	95.25
lex. + GF	81.17	76.83	65.39
Collins	68.63	66.94	96.23



- hypothesis: lexicalization does not cope well with flat structures
- test: modify Collins parser to condition on sister node instead of on the head
- old P_r :

$$P_r(R_i, t(R_i), l(R_i) | P, H, t(H), l(H), d(i))$$

- new P_r :

$$P_r(R_i, t(R_i), l(R_i) | P, R_{i-1}, t(R_{i-1}), l(R_{i-1}), d(i))$$



Results for Treebank POS Tags



	lab. recall	lab. precision	coverage
unlexicalized	72.99	70.00	95.25
Collins	68.63	66.94	96.23
Sister-Head	73.93	74.24	95.21