Unlexicalized Parsing and Transition-based Parsing

Niranjan Balasubramanian
Stony Brook University

Aug 31, 2015

Vanilla PCFG Parsing

$$T^* = \arg\max_T Pr(T|S)$$

$$Pr(T|S) = \prod_{\alpha \to \beta \in T} Pr(\alpha \to \beta|\alpha)$$

DERIVATION	RULES USED	PROBABILITY
S	S → NP VP	1.0
NP VP	NP → DT N	0.3
DT N VP	DT → the	1.0
the N VP	N → dog	0.1
the dog VP	VP → VB	0.4
the dog VB	VB → laughs	0.5
the dog laughs		

TOTAL PROBABILITY = $1.0 \times 0.3 \times 1.0 \times 0.1 \times 0.4 \times 0.5$

Context

- A rule application is not necessarily independent of its context:
  - Parents, children, or sibling categories influence choice of rule.

    e.g., NPs under S different from NPs under VP

  - Head word of the current child constituent and the head word of the phrase influence choice of rule.

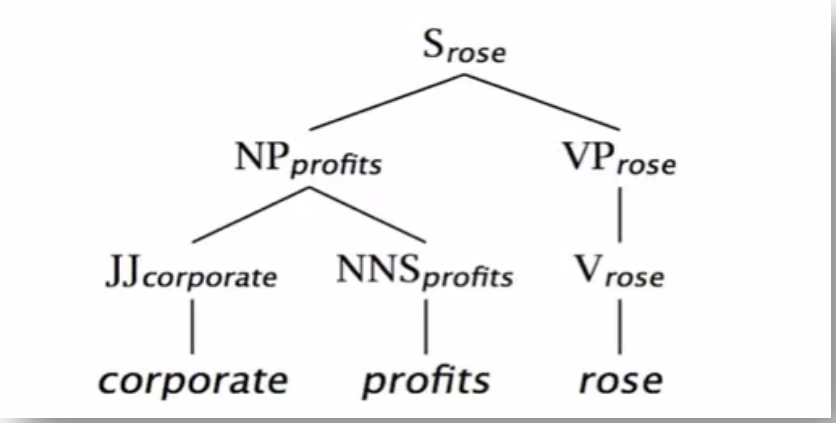
    e.g., Prepositional phrases attachment depend on the head verb.

Lexicalized Charniak Parser

- Key idea is to identify heads of constituents and use them to condition probabilities.
  - There are a handful of rules that specify how to identify heads.
- Probability of lexicalized parse tree is computed using these two quantities.

  P(cur_head = profits | cur_category = NP, parent_head = rose, parent_category = S)

  P(rule = r_i | cur_head = profits, cur_category = NP, parent_category = S)



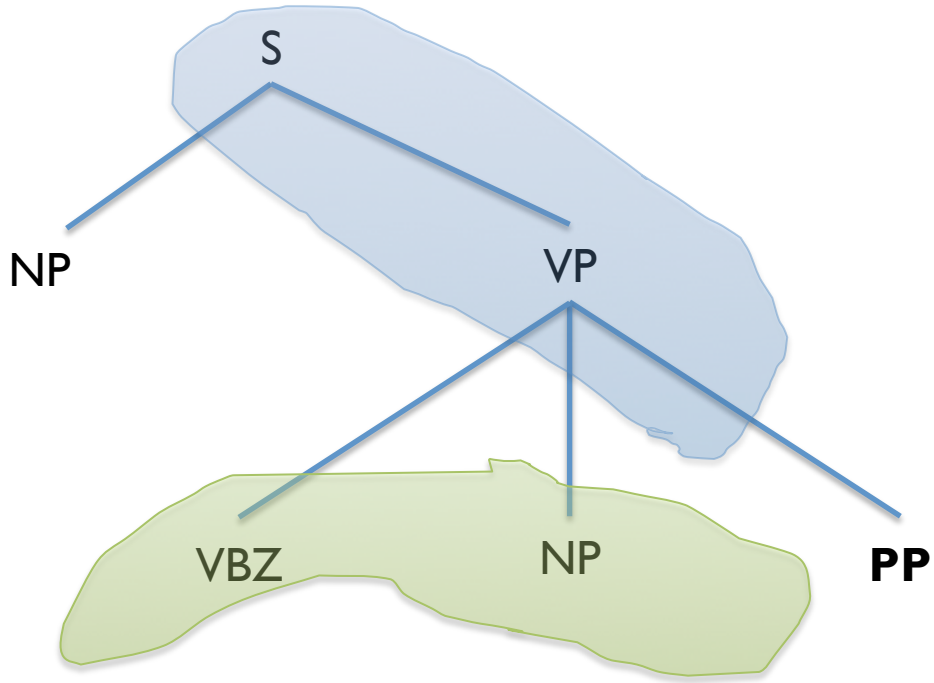
Unlexicalized Parsing
[Klein and Manning, 2003]

- Why bother?
  - Lexicalization increases grammar size
  - Estimation headaches.
  - Domain adaptation is an issue.
  - Asymptotic complexity jumps to $O(N^5)$.
    - Requires clever algorithms to get it down to $O(N^3)$.

    [Eisner and Satta, 1996]

- What is the main idea?
  - Improve vanilla PCFG by adding different forms of context annotations.

Markovization



There is a horizontal context and a vertical context.

Vanilla PCFG

-- Conditions on the immediate vertical context i.e., parent.

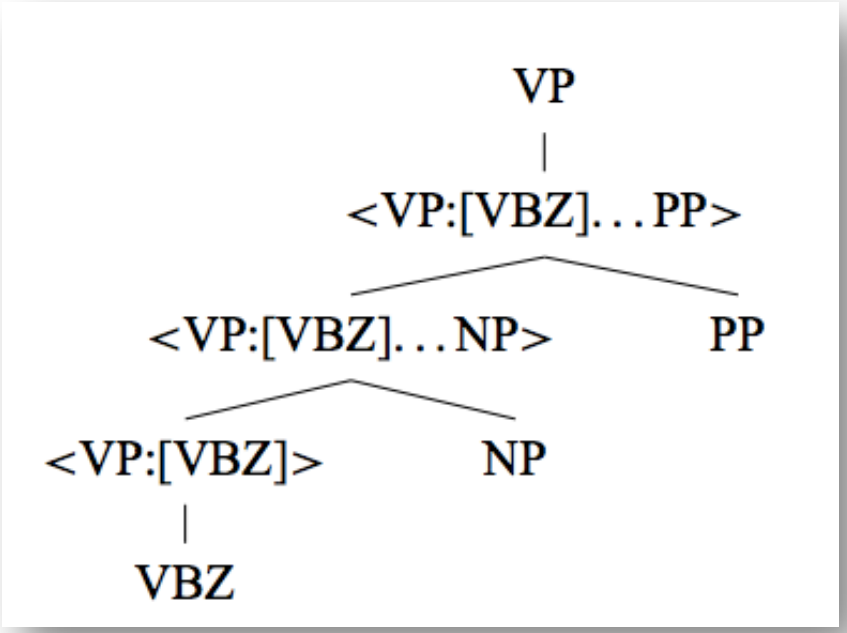
-- Uses all of the horizontal context i.e., expands to all children categories simultaneously.

Horizontal Markovization

Intuition:

Estimating over entire RHS can lead to poor estimates when RHS has many non-terminals.

Break it down by assuming that expansions from **head** are independent given immediate neighbors.



Example:

Pr(VP → VBZ NP PP PP | VP)

= Pr(<VP:[VBZ]> → VBZ | <VP:[VBZ]>) x

Pr(<VP:[VBZ]…NP> → <VP:[VBZ]>NP | <VP:[VBZ]…NP>) x

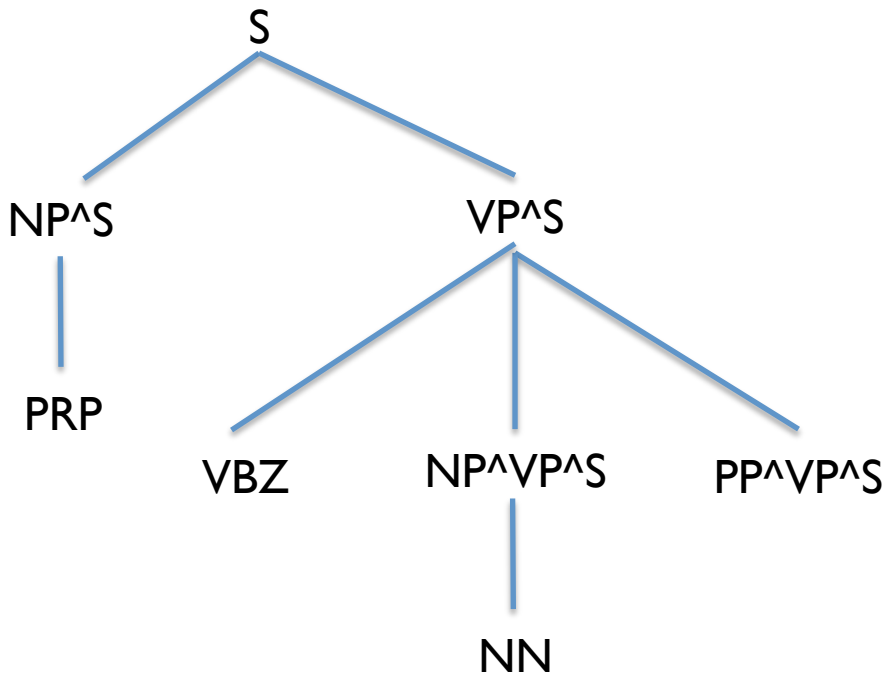
Pr(<VP:[VBZ]…PP> → <VP:[VBZ]>NP PP | <VP:[VBZ]…PP>)

Vertical Markovization

Intuition:

Expanding conditioning on the one parent assumes too little vertical context.

Use all ancestors as vertical context.
Explore various length vertical histories.



Example:

Pr(T|S) = ???

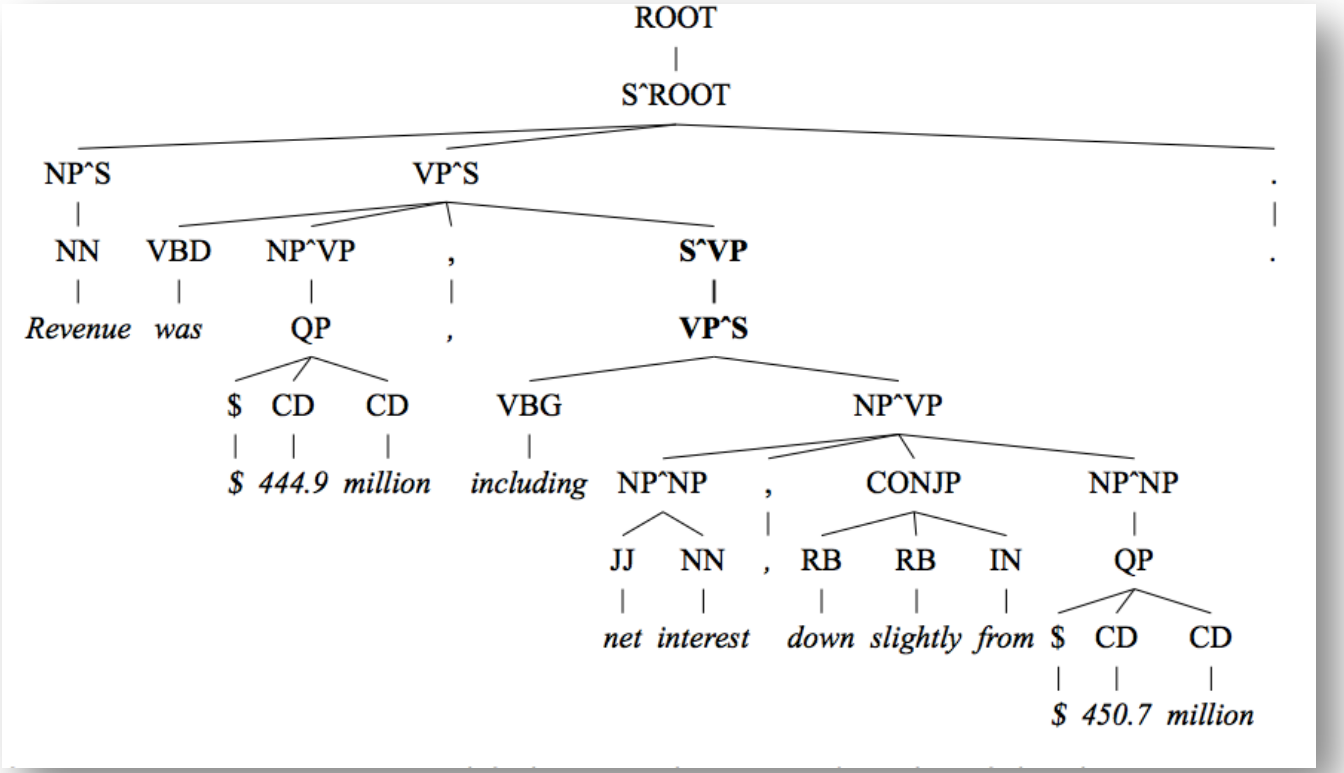
Markovization Results

Vertical Order		Horizontal Markov Order $h=0$	$h=1$	$h\leq 2$	$h=2$	$h=\infty$
$v=1$	No annotation	71.27	72.5	73.46	72.96	72.62
		(854)	(3119)	(3863)	(6207)	(9657)
$v\leq 2$	Sel. Parents	74.75	77.42	77.77	77.50	76.91
		(2285)	(6564)	(7619)	(11398)	(14247)
$v=2$	All Parents	74.68	77.42	77.81	77.50	76.81
		(2984)	(7312)	(8367)	(12132)	(14666)
$v\leq 3$	Sel. GParents	76.50	78.59	79.07	78.97	78.54
		(4943)	(12374)	(13627)	(19545)	(20123)
$v=3$	All GParents	76.74	79.18	79.74	79.07	78.72
		(7797)	(15740)	(16994)	(22886)	(22002)

Summary:

Adding vertical context helps.
Restricting horizontal context to depend on fewer ancestors helps.
Trade-off between grammar size and utility of history.

Internal Annotations



Unary production S^VP→VP^S is incorrectly used here.

The content of the expanded VP precludes the use of this rule.

Internal S nodes are often complements of communication verbs.

e.g., She thought that John was killed in the accident.

Mark the non-terminal in unary productions to say it has only one child.

External Annotation

Mark nodes that have no siblings.

Useful in the case of pre-terminals where internal annotation is meaningless

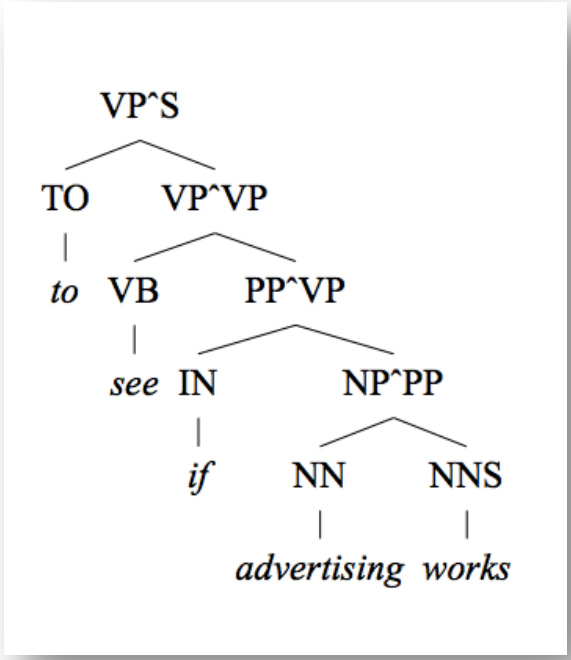
-- All pre-terminal to terminal expansions are unary!

PTB conflates demonstratives (those, that) from true determiners (a, the).

-- Adding UNARY-DT distinguishes between these two cases.

e.g., The apples that were good vs. Those apples were good.

Tag Splitting



Many POS tags in PTB conflate distinct categories with different attachment preferences.

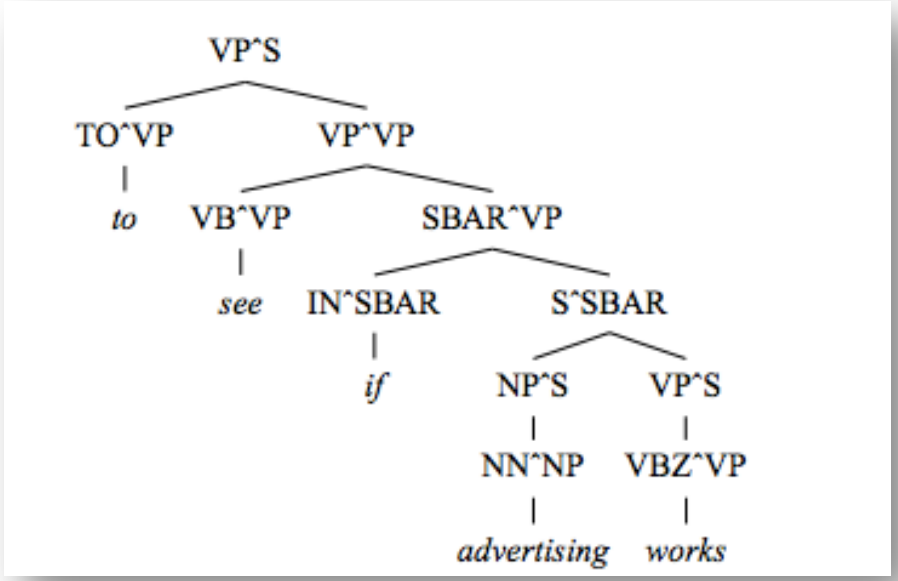
e.g. Preposition tag IN can be sub-ordinating conjunctions or regular prepositions.

Sub-ordinating conjunctions typically associate with a sentence category (S) to form SBARs.

1) Mark pre-terminals with parent information when they occur in non-canonical categories.

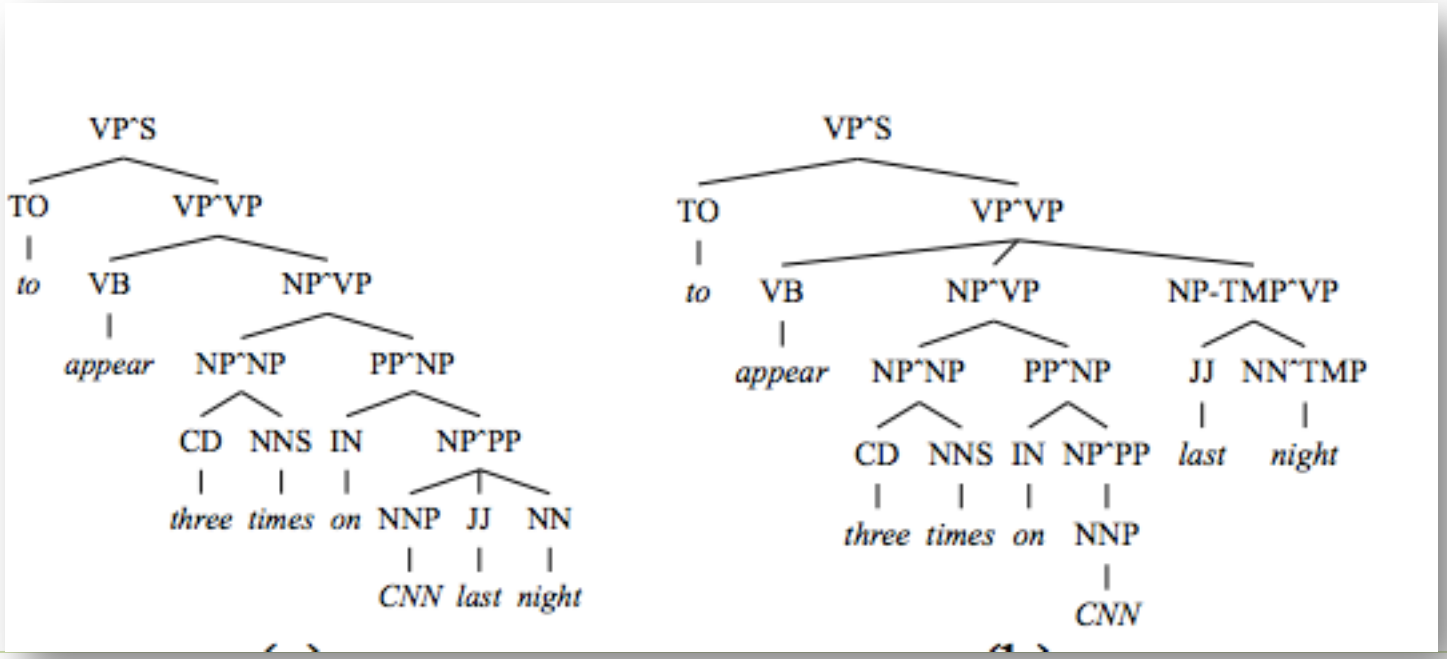
Tags associating with non-canonical categories have a specific distribution.

2) SPLIT-IN to mark prep categories.

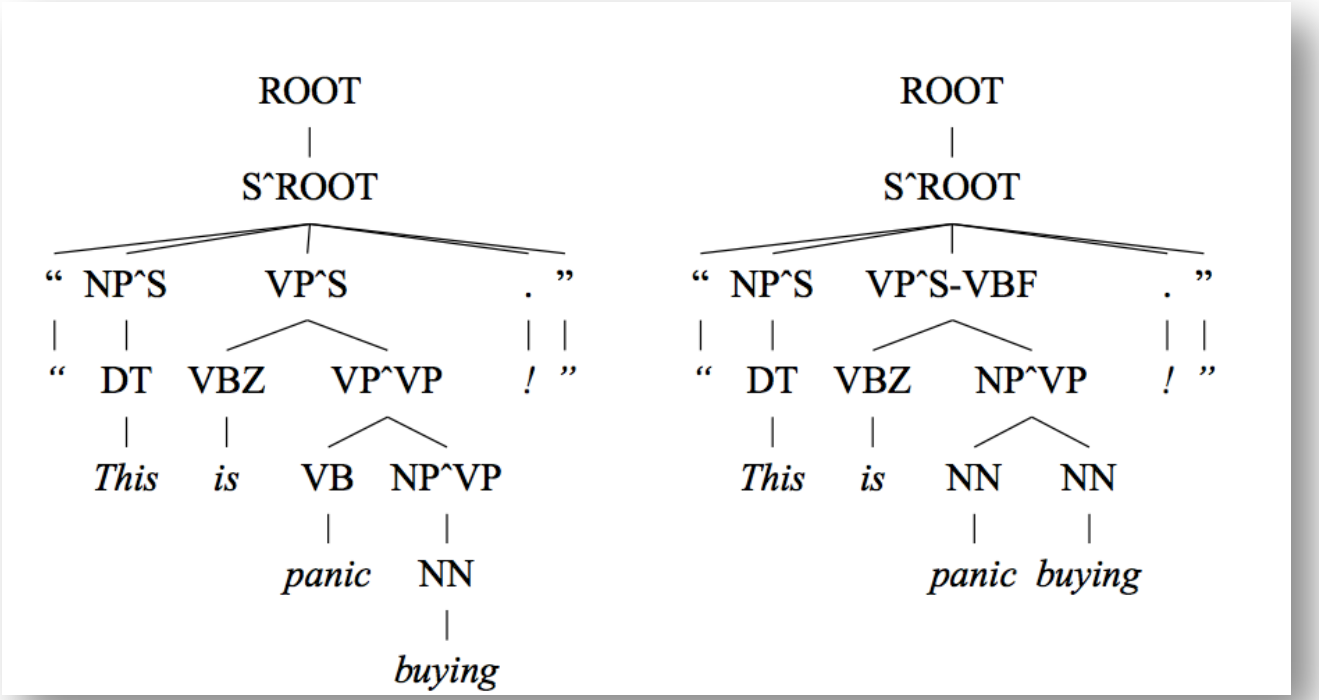


Other Modifications

- SPLIT-CC, SPLIT-VP etc.
- Mark S nodes with empty subjects (GAPPED-S).
  - She was planning [_____ to apply for the position.]
- TMP (temporal tags) are helpful.
  - TMP tags on NPs are propagated down from heads.



Head Annotation



Present tense verbs do not usually take infinitival verb complements

-- “is” doesn’t take the infinitive “panic”, rather panic is a modifier on buying.

-- Marking non-infinitival verbs as VBFs fixes the problem.

Similarly marking POSS-NP constructions also helps.

-- In PTB annotations, NP → NP * is used only for possessives.

e.g., John’s pizza as opposed to Long Island beaches

How far can an unlexicalized grammar go?

Pretty far! Nearly a 10% absolute improvement in F1.

	Cumulative			Indiv.
Annotation	Size	F_1	ΔF_1	ΔF_1
Baseline ($v \leq 2, h \leq 2$)	7619	77.77	–	–
UNARY-INTERNAL	8065	78.32	0.55	0.55
UNARY-DT	8066	78.48	0.71	0.17
UNARY-RB	8069	78.86	1.09	0.43
TAG-PA	8520	80.62	2.85	2.52
SPLIT-IN	8541	81.19	3.42	2.12
SPLIT-AUX	9034	81.66	3.89	0.57
SPLIT-CC	9190	81.69	3.92	0.12
SPLIT-%	9255	81.81	4.04	0.15
TMP-NP	9594	82.25	4.48	1.07
GAPPED-S	9741	82.28	4.51	0.17
POSS-NP	9820	83.06	5.29	0.28
SPLIT-VP	10499	85.72	7.95	1.36
BASE-NP	11660	86.04	8.27	0.73
DOMINATES-V	14097	86.91	9.14	1.42
RIGHT-REC-NP	15276	87.04	9.27	1.94

So what is being done here?

- Linguistic insights i.e., knowledge about language constructions are being used.

- What is problematic about this approach?
  - Well, we need people knowledgeable about language (aka linguists).
  - Adapting to new domains requires manual intervention.

- Not sure if we've captured all relevant insights!

Summary

- One can think of many of these annotations as introducing sub-categories.
- By sub-categorizing are we effectively doing what lexicalization does?
  - The sub-categorization or state-splitting can never quite get close to the sparsity issues of open-class words.
  - Care taken to not back-off to uncategorized rules.
- Is there a way to automatically induce these sub-categories rather than manually specifying them?
  - Latent variable models w/ EM [Petrov and Klein, 2007]
- Essentially there is a trade-off in grammar size vs. effectiveness.
  - What are the implications for how much data we need?
  - Can un-lexicalized grammars be learnt sooner?