Multilevel Coarse-to-Fine PCFG Parsing

Eugene Charniak, Mark Johnson, Micha Elsner, Joseph Austerweil, David Ellis, Isaac Haxton, Catherine Hill, Shrivaths Iyengar, Jeremy Moore, Michael Pozar, and Theresa Vu

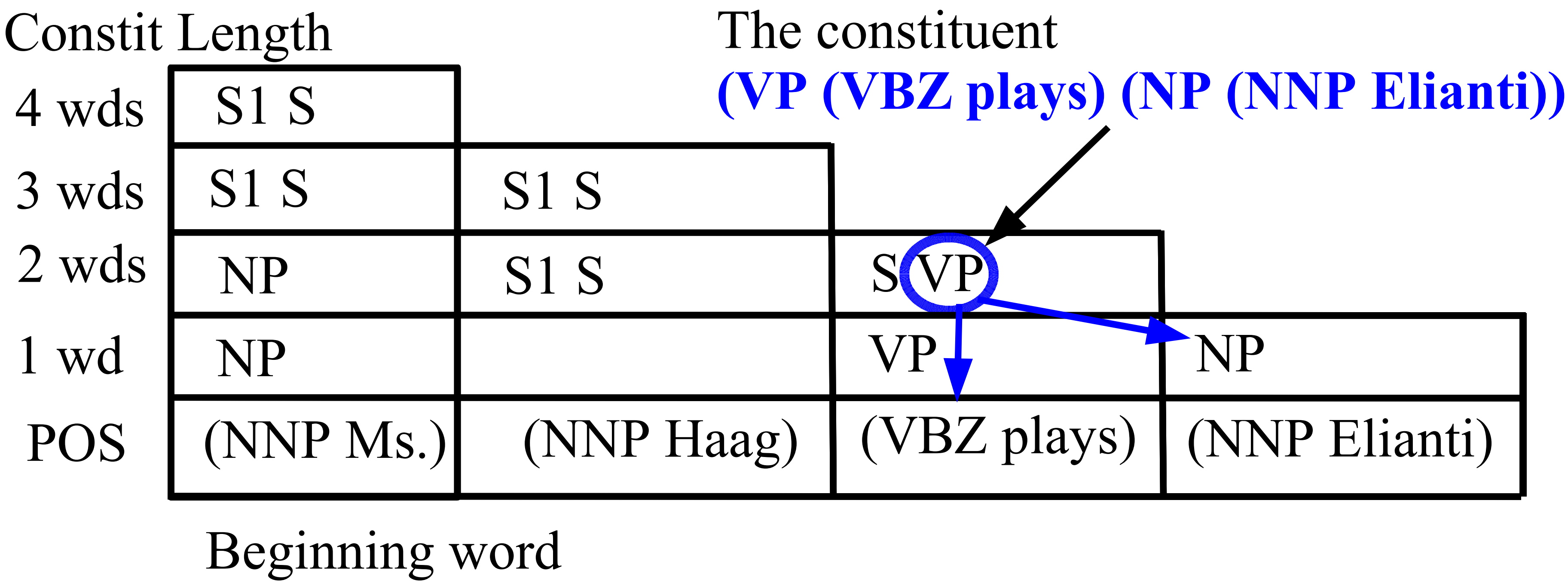
Brown Laboratory for Linguistic Information Processing (BLLIP)

BROWN

Statistical Parsing Speed

- Lexicalized statistical parsing can be slow.
  - Charniak: 0.7 seconds per sentence.
- Real applications demand more speed!
  - Large corpora, eg. NANTC (McClosky, Charniak and Johnson 2006)
  - More words to consider-- lattices from speech recognition (Hall and Johnson 2004)
  - Costly second stage such as question answering.

Bottom-up Parsing I



- Standard probabilistic CKY chart parsing.
  - Computes the inside probability β for each constituent.

Bottom-up Parsing II

Constit Length				
4 wds	**S1 S**			
3 wds	S1 S	S1 S		
2 wds	**NP**	S1 S	S **VP**	
1 wd	NP		VP	**NP**
POS	(NNP Ms.)	(NNP Haag)	(VBZ plays)	(NNP Elianti)

Beginning word

- Some constituents are **gold constituents** (parts of correct parse).
  - These may not be part of the highest probability (Viterbi) parse.
  - We can use a reranker to try to pick them out later on.

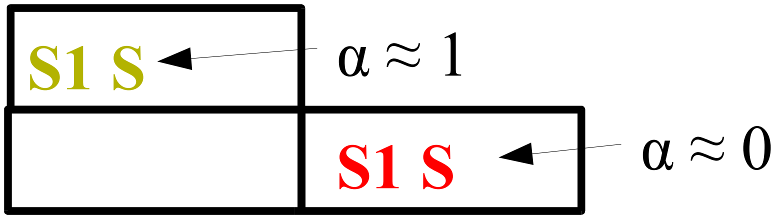
Pruning

- We want to dispose of the incorrect constituents and retain the **gold**.
- Initial idea: prune **constituents with low probability** (~ outside α times inside β).
  - $$p(n^k_{i,j} \mid s) = \frac{\alpha(n^k_{i,j})\beta(n^k_{i,j})}{p(s)}$$

4 wds	**S1 S**			
3 wds	**S1 S**	**S1** S		
2 wds	**NP**	**S1 S**	**S VP**	
1 wd	NP		VP	**NP**
POS	(NNP Ms.)	(NNP Haag)	(VBZ plays)	(NNP Elianti)

Outside Probabilities

- We need the full parse of the sentence to get outside probability α.
  - Estimates how well the constituent contributes to spanning parses for the sentence.



- Caraballo and Charniak (1998): agenda reordering method-- proper pruning needs an approximation of α.
  - Approximated α using ngrams at constituent boundaries.

Coarse-to-Fine Parsing

- Parse quickly with a smaller grammar.

4 wds	**S1 P**			
3 wds	S1 P	S1 P		
2 wds	**P**	S1 P	**P**	
1 wd	P		P	**P**
POS	(NNP Ms.)	(NNP Haag)	(VBZ plays)	(NNP Elianti)

- Now calculate α using the full chart.

4 wds	**S1 P**			
3 wds	**S1** P	**S1** P		
2 wds	**P**	**S1 P**	**P**	
1 wd	P		P	**P**
POS	(NNP Ms.)	(NNP Haag)	(VBZ plays)	(NNP Elianti)

Coarse-to-Fine Parsing II

- Prune the chart, then reparse with a more specific grammar.

4 wds	**S1 S_**			
3 wds	~~**S1**~~ S	~~**S1**~~ S		
2 wds	**N_**	~~**S1 P**~~	S_ **V_**	
1 wd	N_		V_	**N_**
POS	(NNP Ms.)	(NNP Haag)	(VBZ plays)	(NNP Elianti)

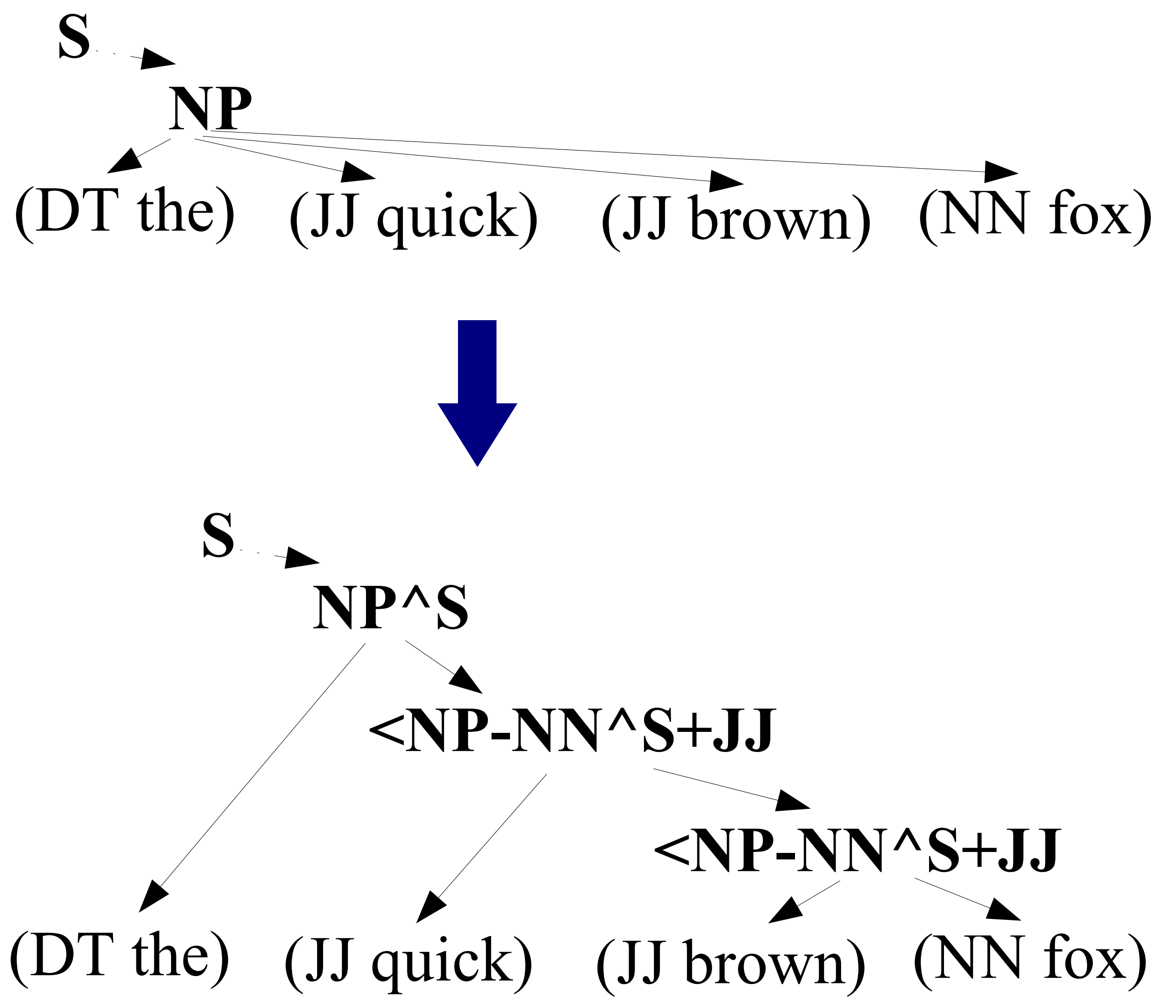
- Repeat the process until the final grammar is reached.
- Reduces the cost of a high grammar constant.

Related Work

- Two-stage parsers:
  - Maxwell and Kaplan (1993); automatically extracted first stage
  - Goodman (1997); first stage uses regular expressions
  - Charniak (2000); first stage is unlexicalized
- Agenda reordering:
  - Klein and Manning (2003); A* search for the best parse using an upper bound on α.
  - Tsuruoka and Tsujii (2004); iterative deepening.

Parser Details

- Binarized grammar based on Klein and Manning (2003)
  - Head annotation.
  - Vertical (parent) and horizontal (sibling) Markov context.



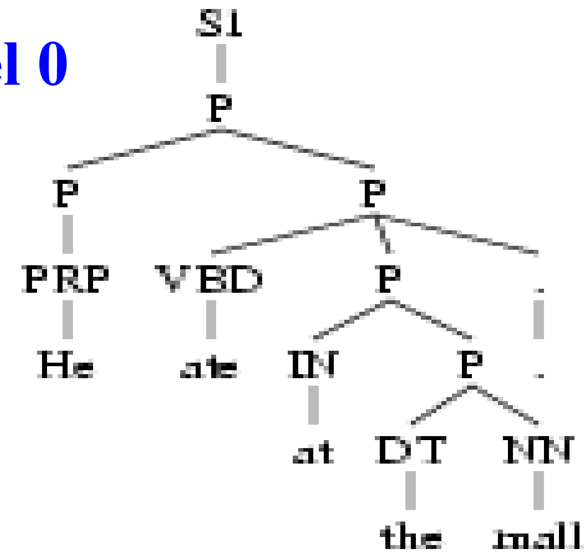
Coarse-to-Fine Scheme

S1	P				**Level 0**
S1	HP		MP		**Level 1**
					Level 2
S1	S_	N_	A_	P_	
S1	S VP UCP SQ SBAR SBARQ	NP NAC NX LST X UCP FRAG	ADJP QP CONJP ADVP INTJ PRN PRT	PP PRT RRC WHADJP WHADVP WHNP WHPP	

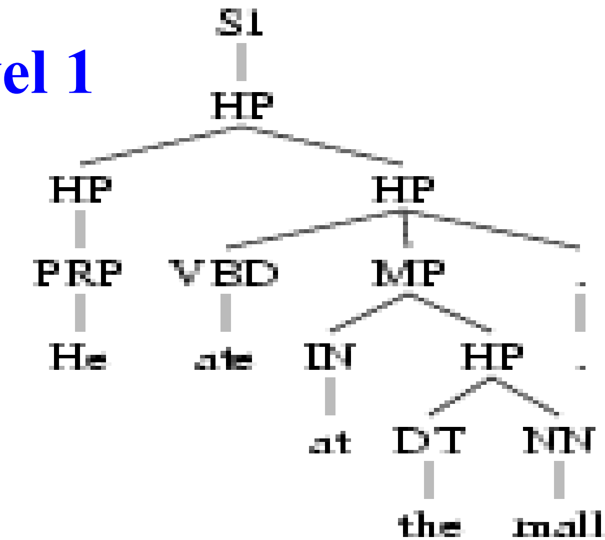
Level 3: Full Treebank Grammar

Examples

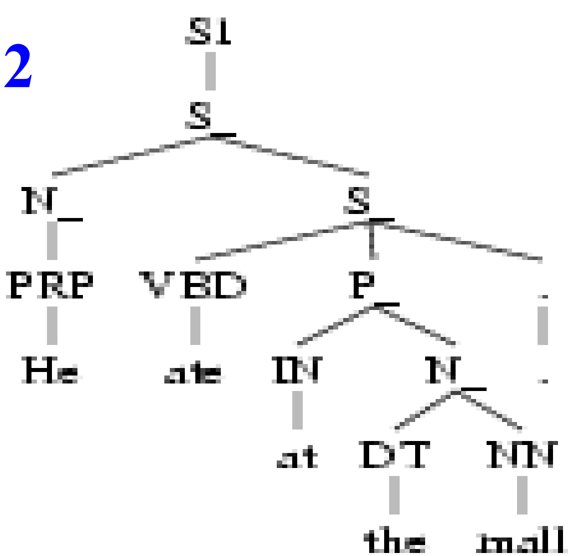
Level 0



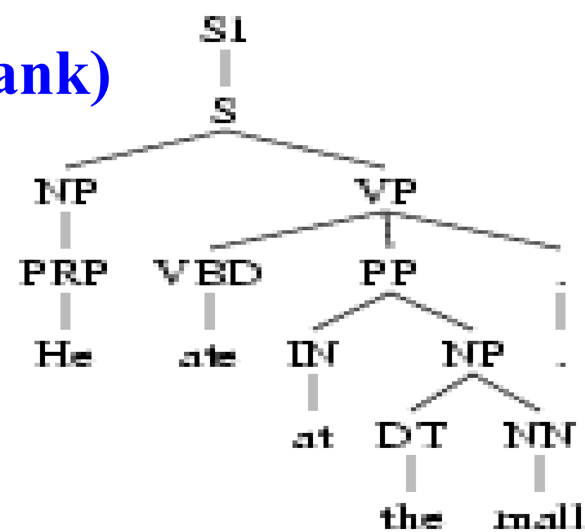
Level 1



Level 2



Level 3 (Treebank)



Coarse-to-Fine Probabilities

Heuristic probabilities:

$P(N\_ \rightarrow N\_\ P\_) =$
weighted-avg(
$P(NP \rightarrow NP\ PP)$
$P(NP \rightarrow NP\ PRT)$
...
$P(NP \rightarrow NAC\ PP)$
$P(NP \rightarrow NAC\ PRT)$
...
$P(NAC \rightarrow NP\ PP)$
...)

Using **max** instead of **avg** computes an exact upper bound instead of a heuristic (Geman and Kochanek 2001).

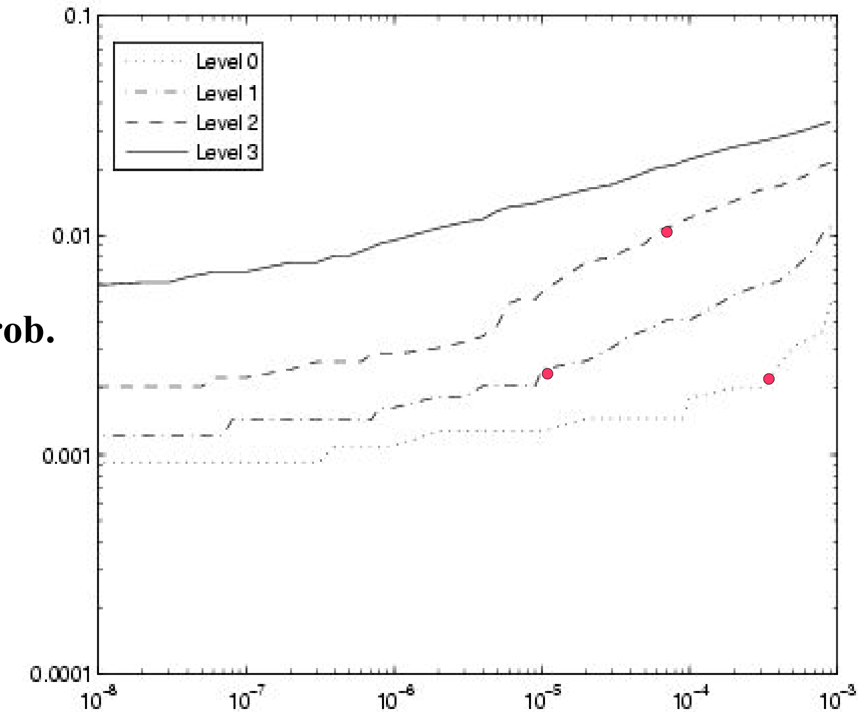
No smoothing needed.

Pruning Thresholds

Pruning threshold vs. probability of pruning a **gold** constituent

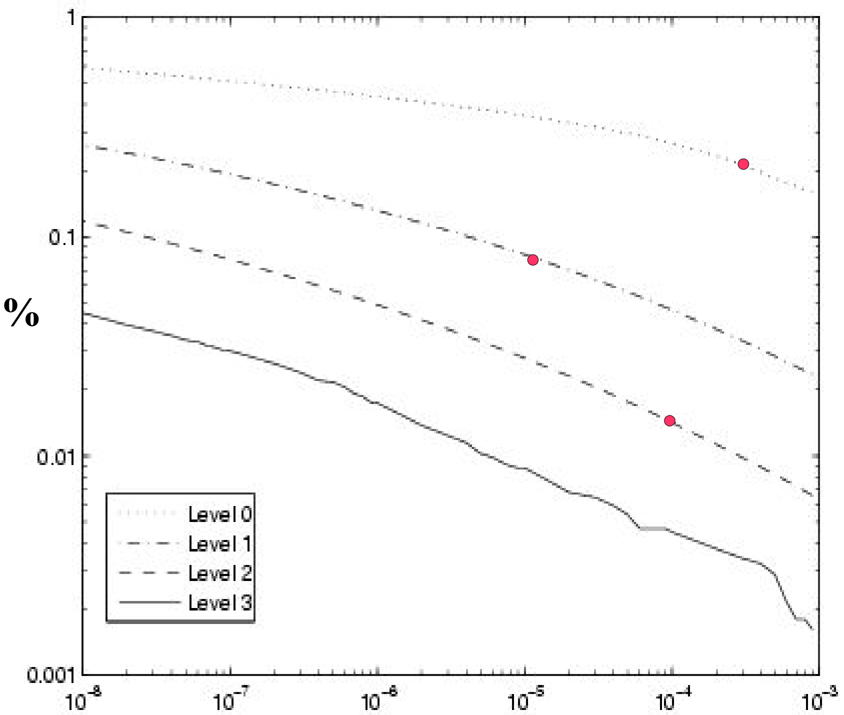
Threshold vs. fraction of incorrect constituents remaining.

Prob.



Pruning threshold

%



Pruning threshold

Pruning Statistics

	Constits Produced (millions)	Constits Pruned (millions)	% Pruned
Level 0	8.82	7.55	86.5
Level 1	9.18	6.51	70.8
Level 2	11.2	9.48	84.4
Level 3	11.8	0	0
Total	**40.4**	-	-
Level 3 only	**392**	0	0

Timing Statistics

	Time At Level	Cumulative Time	F-score
Level 0	1598	1598	
Level 1	2570	4164	
Level 2	4303	8471	
Level 3	1527	**9998**	77.9
Level 3 only	**114654**	-	77.9

10x speed increase from pruning.

Discussion

- No loss in f-score from pruning.
- Each pruning level is useful.
  - Prunes ~80% of constituents produced.
- Pruning at level 0 (only two nonterminals, S1 / P)
  - Preterminals are still useful.
  - Probability of **P-IN** → **NN IN**
    (a constituent ending with a preposition)
    will be very low.

Conclusion

- Multi-level coarse-to-fine parsing allows bottom-up parsing to use top-down information.
  - Deciding on good parent labels.
  - Using the string boundary.
- Can be combined with agenda reordering methods.
  - Use coarser levels to estimate outside probability.
- More stages of parsing can be added.
  - Lexicalization.

Future Work

- The coarse-to-fine scheme we use is hand-generated.
- A coarse-to-fine scheme is just a hierarchical clustering of constituent labels.
  - Hierarchical clustering is a well-understood task.
  - Should be possible to define an objective function and search for the best scheme.
  - Could be used to automatically find useful annotations/lexicalizations.

Acknowledgements

- Class project for CS 241 at Brown University
- Funded by:
  - Darpa GALE
  - Brown University fellowships
  - Parents of undergraduates
- Our thanks to all!