Practical Experiments in Parsing using Tree Adjoining Grammars

Anoop Sarkar
Dept. of Computer and Information Sciences
University of Pennsylvania
anoop@linc.cis.upenn.edu

Parsing Efficiency

- How does the real world relate to the theoretical analysis of parsers.

- Parsing efficiency: time complexity for producing all parses.

- Parsing accuracy: statistical parsing and pruning.

An example: Quicksort

- Input: $\{9, 4, 2, 3, 1, 5, 8, 6, 7, 10\}$

- Output: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

- If input is in random order, time complexity $= \Theta(n \times log(n))$
  but if input is already almost sorted, time complexity $= \Theta(n^2)$

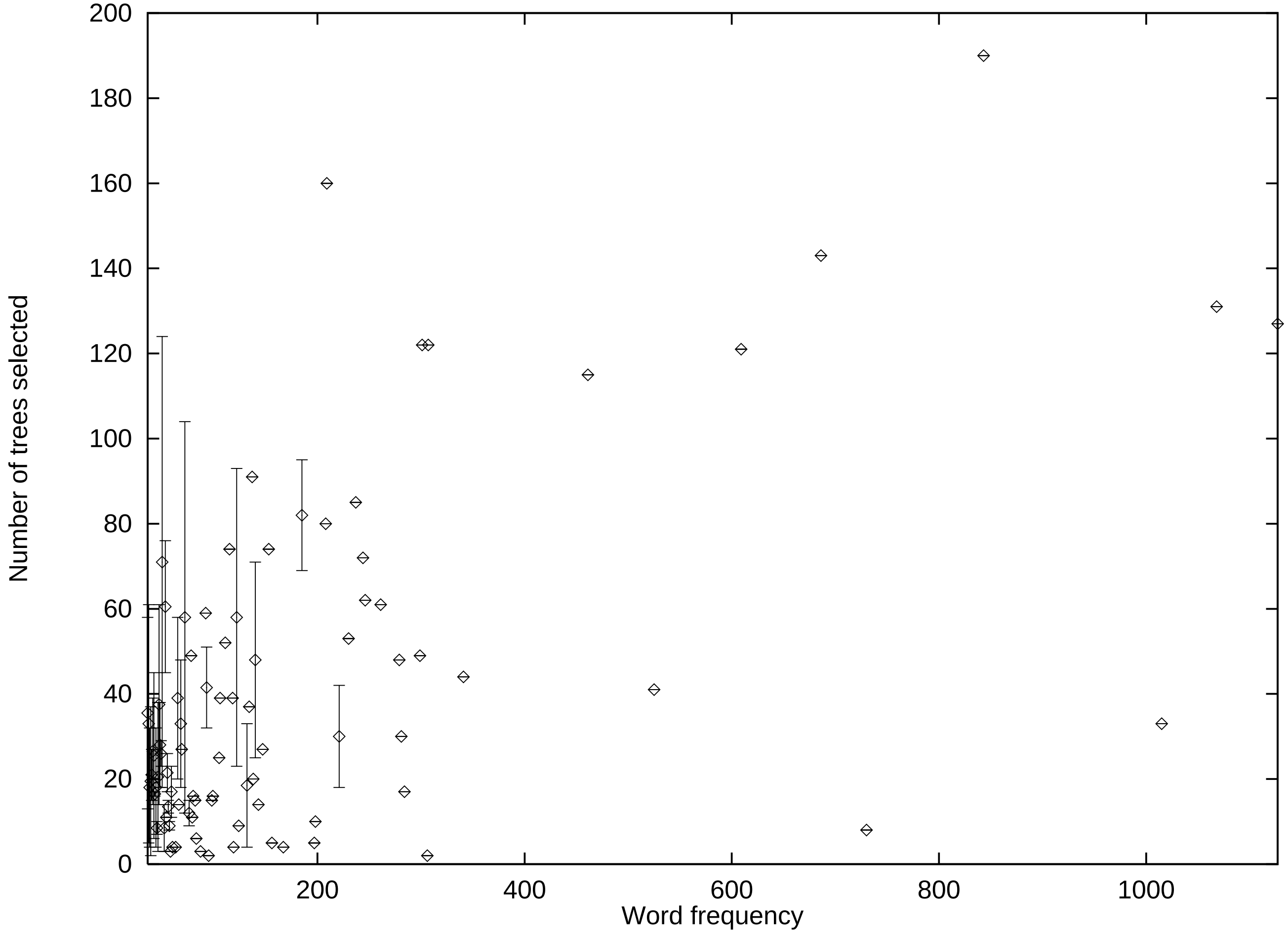
- Are there similar criteria for parsing algorithms . . .
  . . . especially for lexicalized grammar formalisms.

Experiment: The Parser

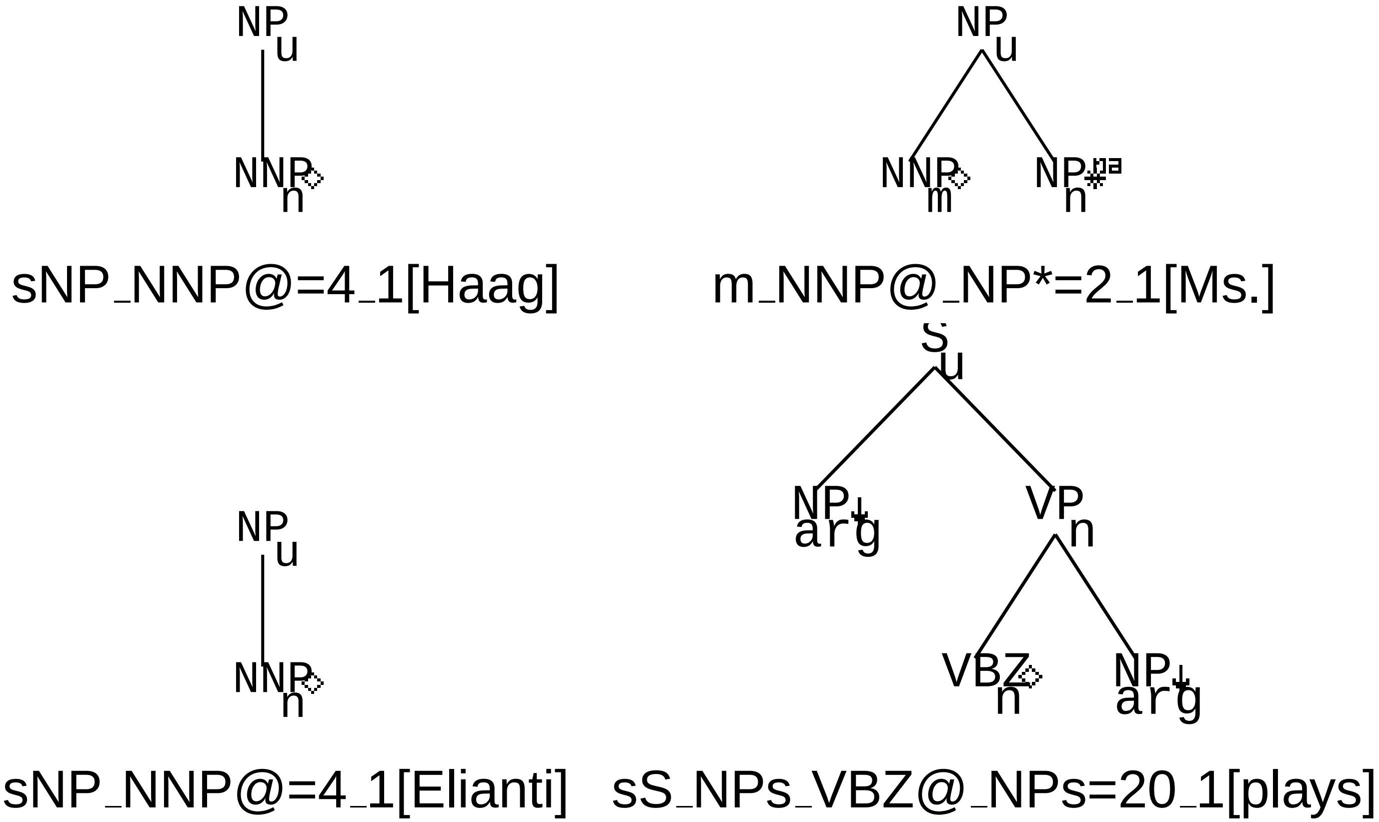
- Implementation of head-corner chart-based parser.
- It is bi-directional – van Noord style.
- Works with the full XTAG English grammar.
- Produces a derivation forest as output.
- Written in ANSI C: α-version available at `ftp://ftp.cis.upenn.edu/xtag/pub/lem`.

Experiment: Input Grammar

- Fei Xia's Treebank Grammar
- extracted from Sections 02–21 WSJ Penn Treebank
- 6789 tree templates, 123039 lexicalized trees
- number of word types in the lexicon is 44215
- average number of trees per word is 2.78



Number of trees selected by the 150 most frequent words



sNP_NNP@=4_1[Haag]

m_NNP@_NP*=2_1[Ms.]

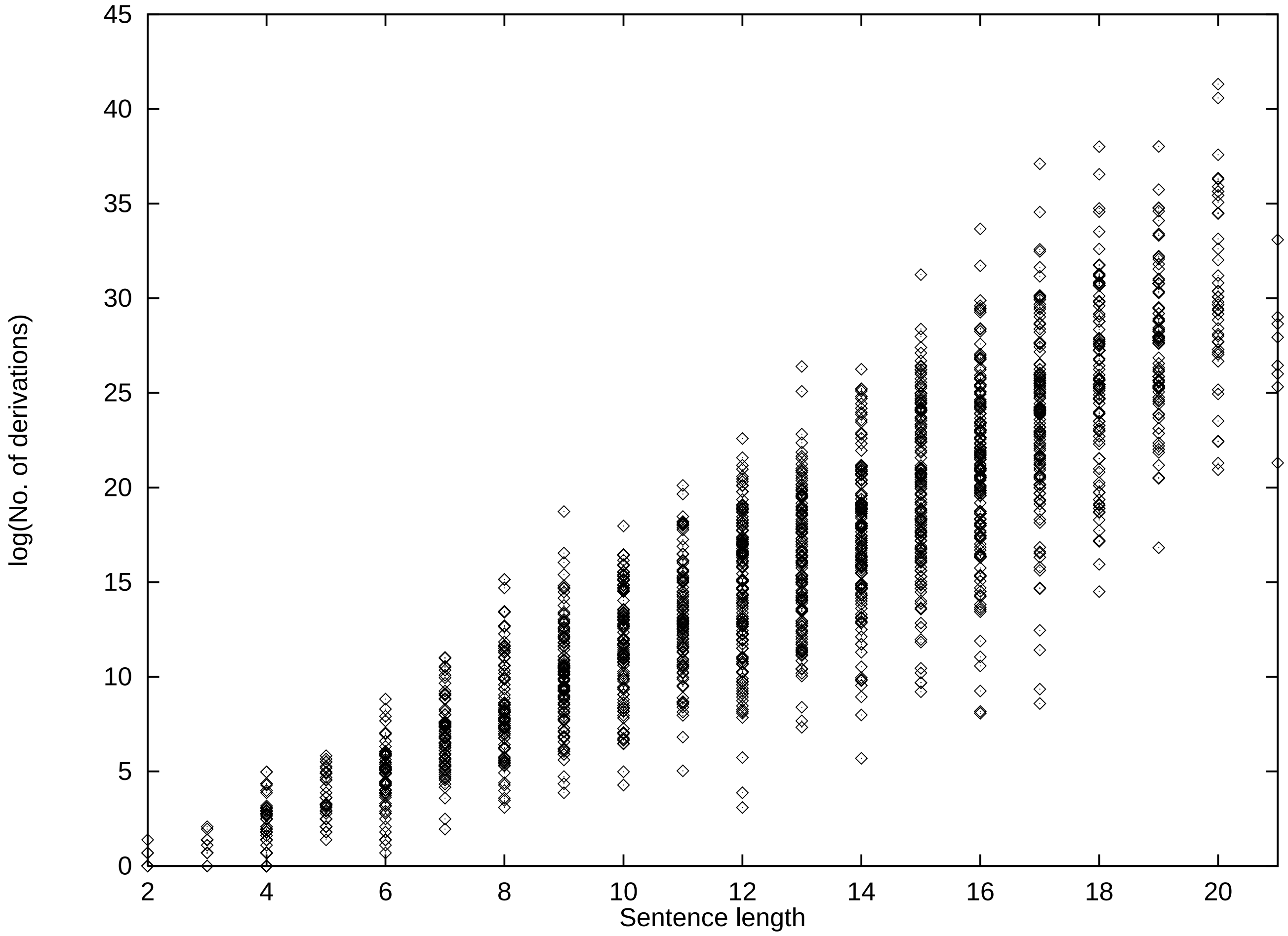
sNP_NNP@=4_1[Elianti]

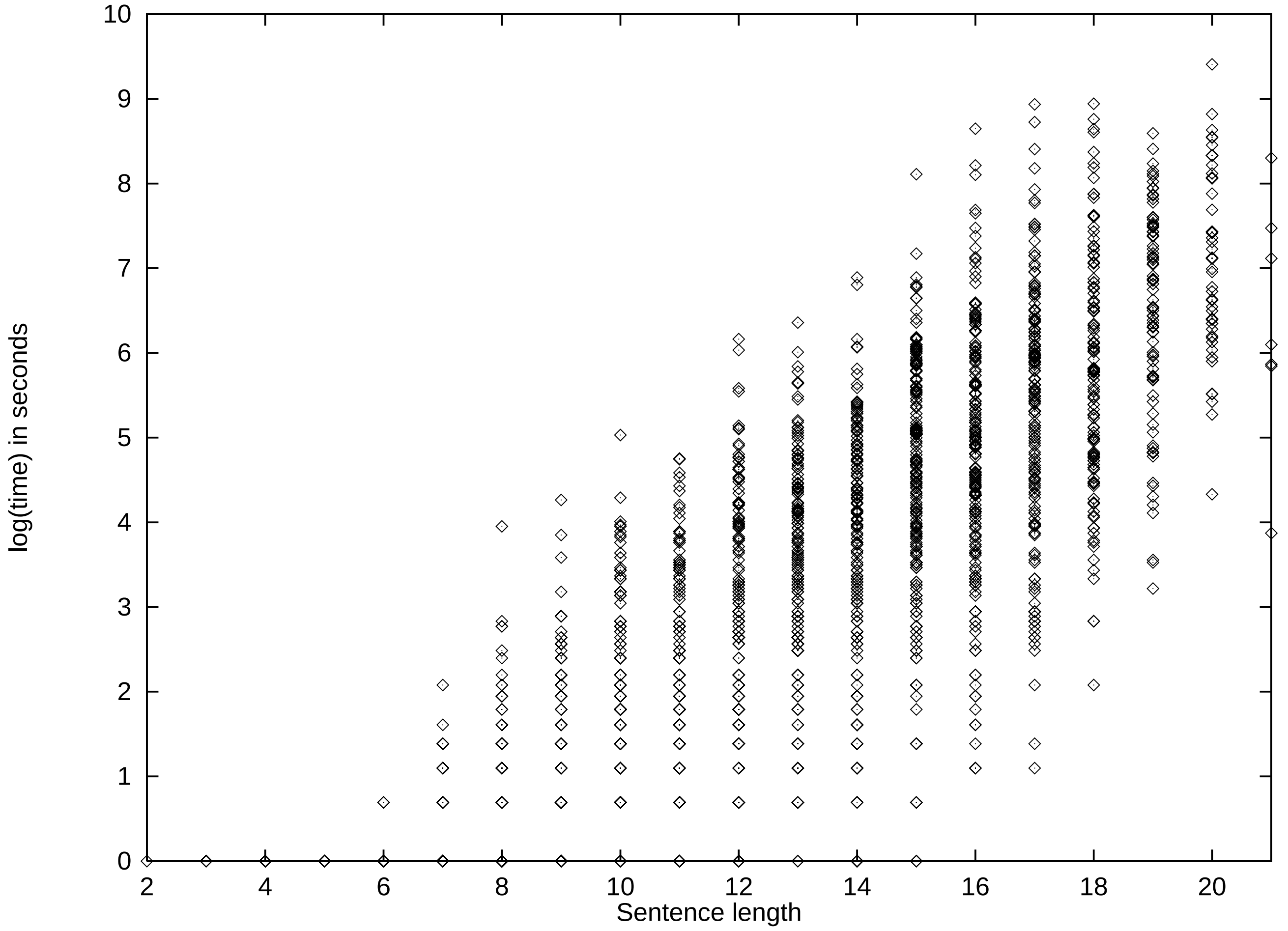
sS_NPs_VBZ@_NPs=20_1[plays]

Example lexicalized elementary trees from the Treebank Grammar. These trees can be combined to parse the sentence *Ms. Haag plays Elianti*.

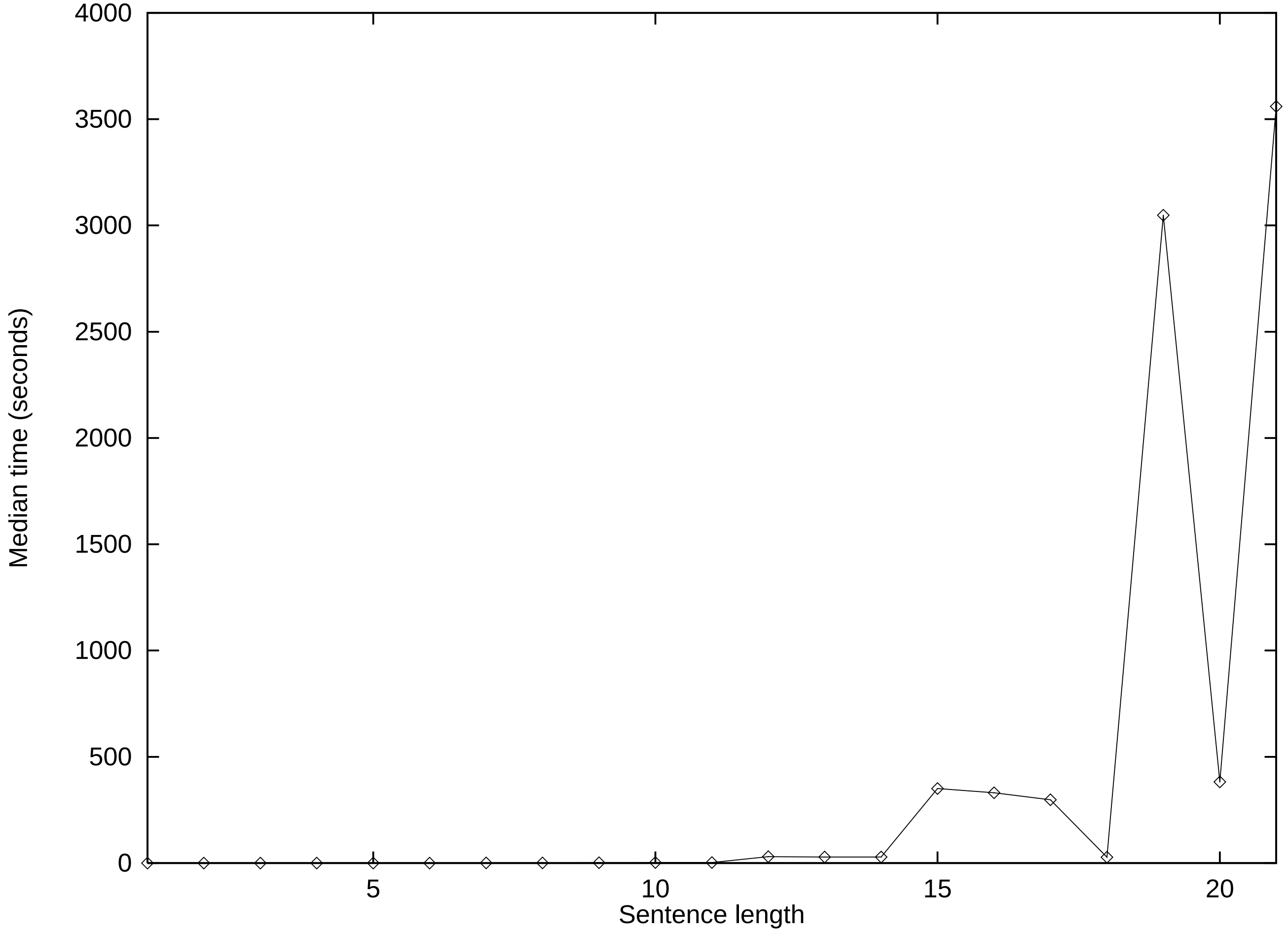
Experiment: Test Corpus

- input was a set of 2250 sentences
- each sentence was 21 words or less
- avg. sentence length was 12.3
- number of tokens = 27715
- output: shared forest of parses


log(No. of derivations)
0
5
10
15
20
25
30
35
40
45
2
4
6
8
10
12
14
16
18
20
Sentence length



Coeff of determination $R^2 = 0.65$


4000
3500
3000
2500
2000
1500
1000
500
0
Median time (seconds)
5
10
15
20
Sentence length

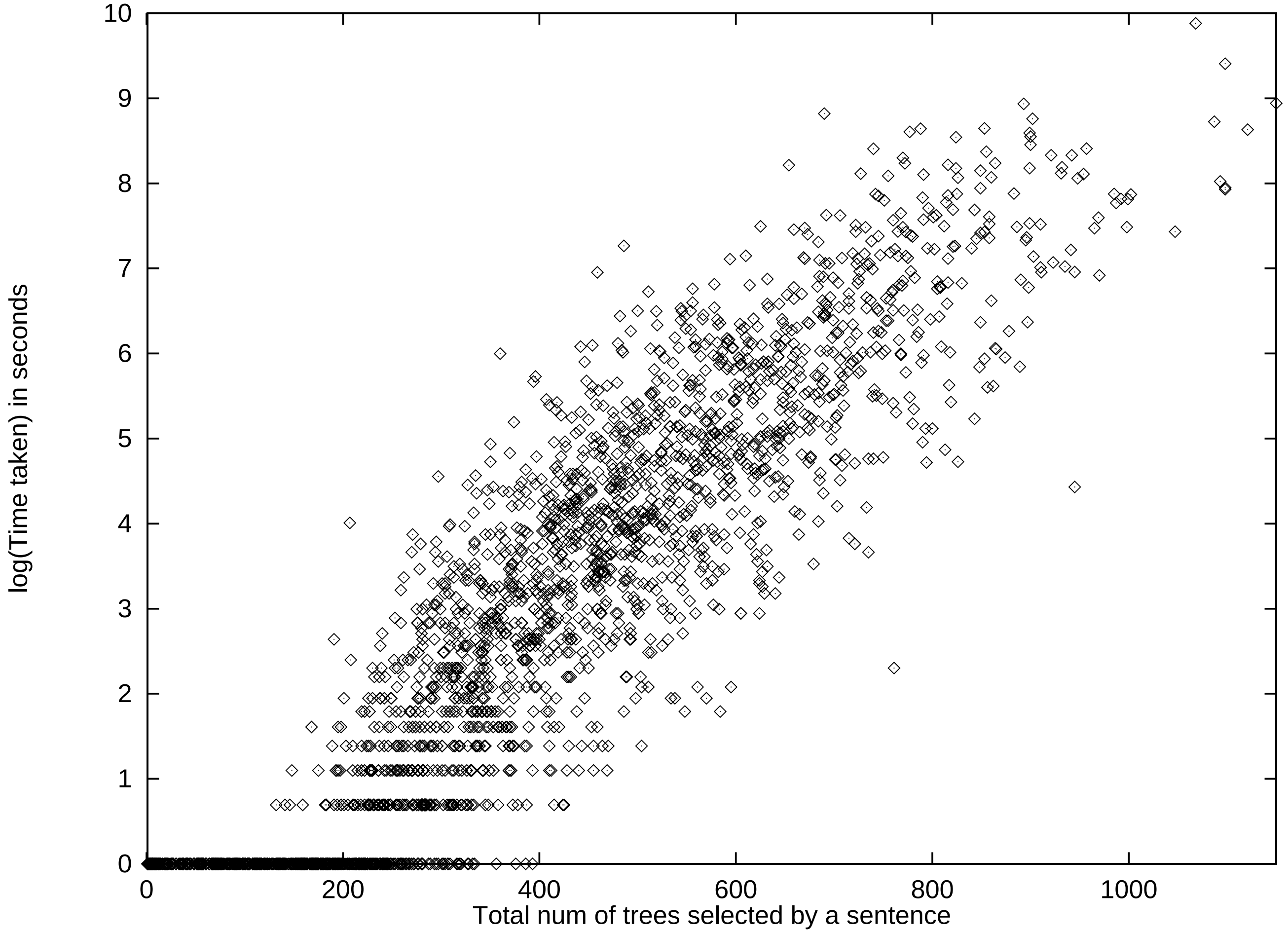
Observations

- LTAG Parsing is shown to be more time and space efficient than the older XTAG Common Lisp parser.

- There are no large hidden constants.

- But there is a large variability in parse times.

- Sentence length is not a reliable indicator of parsing time.



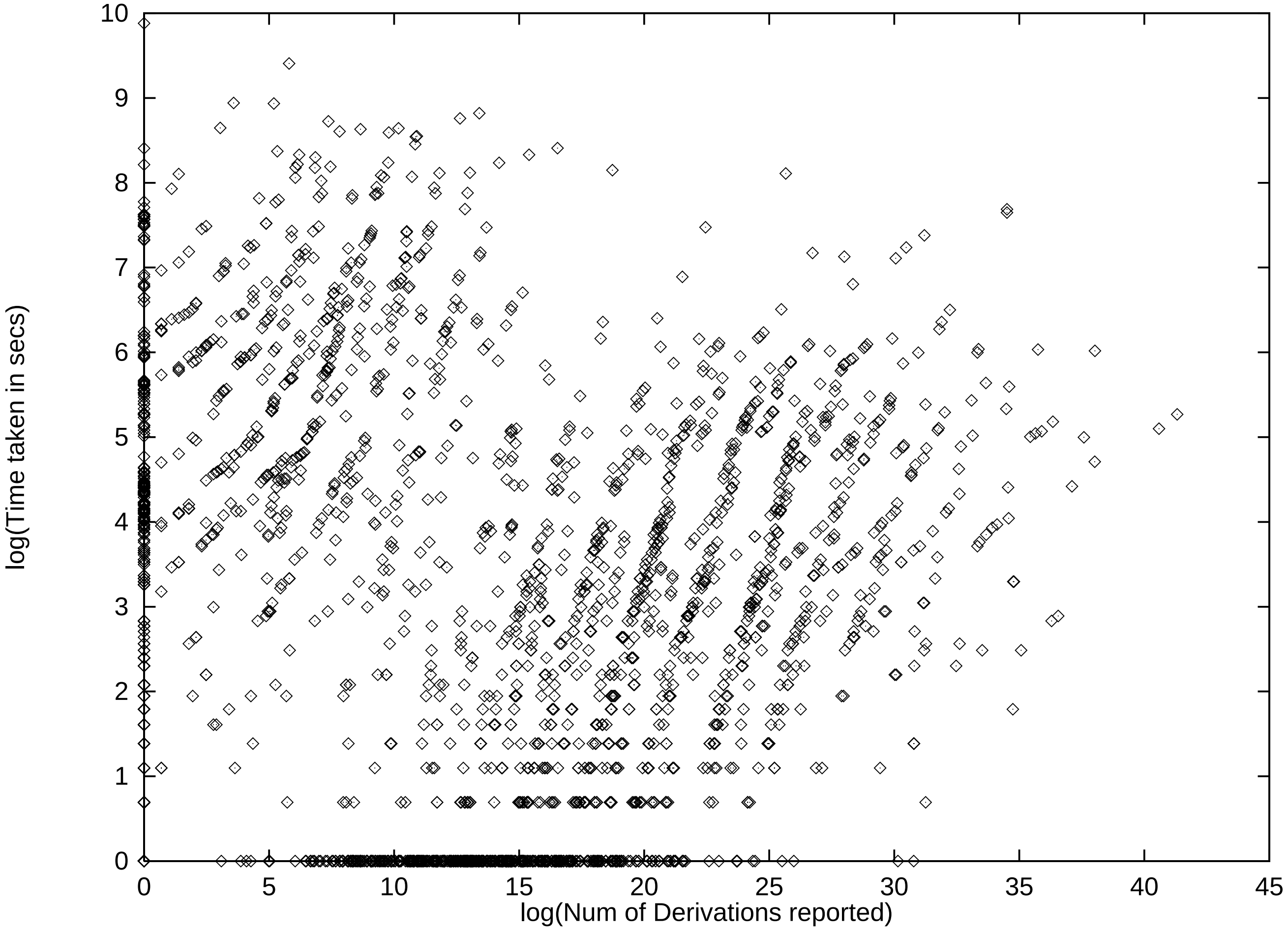
The impact of *syntactic* lexical ambiguity on parsing times. $R^2 = 0.82$

Conclusions

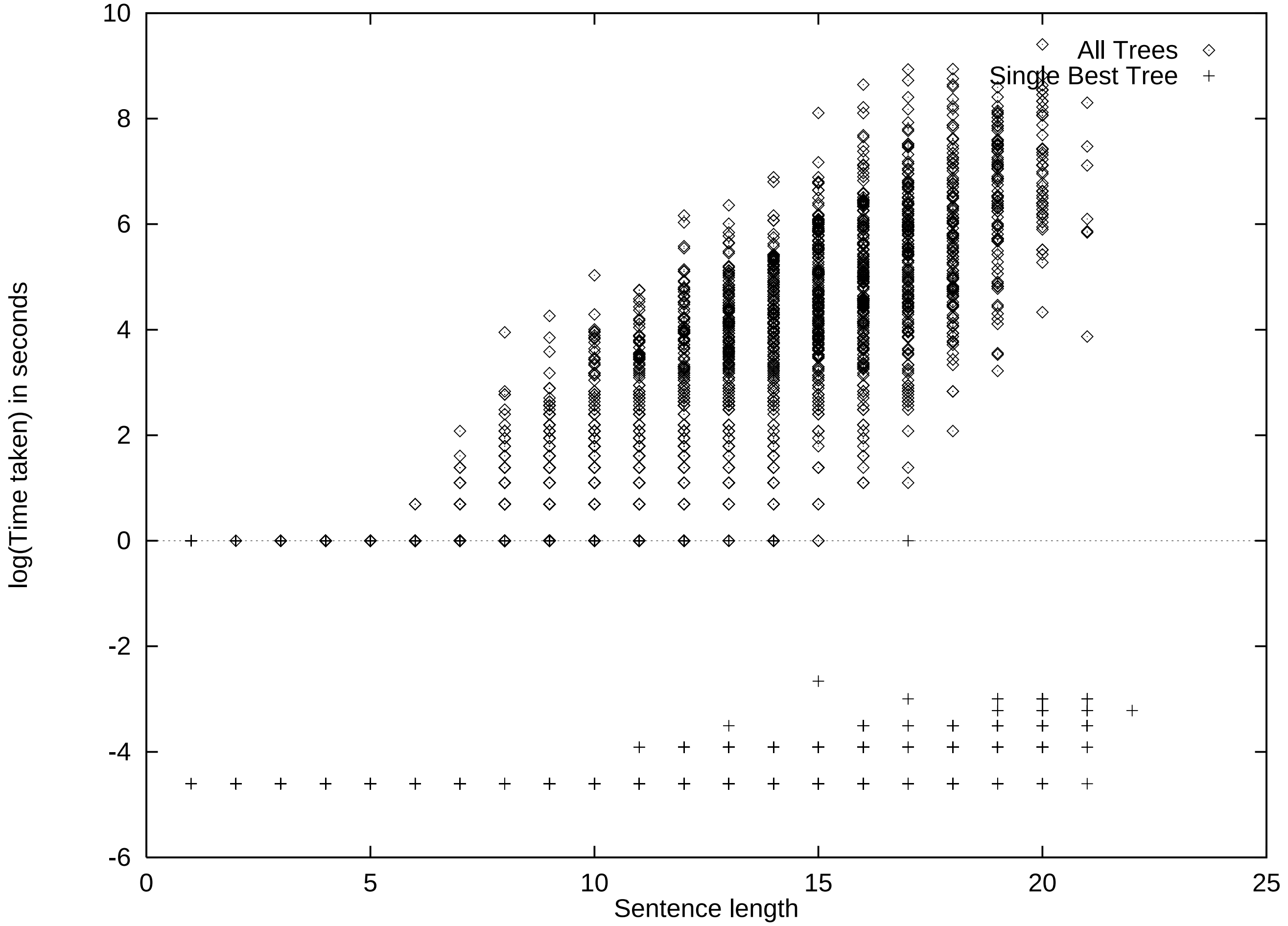
- Length of input sentence is not the only factor in the time complexity for parsing of lexicalized TAGs.

- Syntactic lexical ambiguity (number of trees selected by the sentence) is more significant.

- Theoretical parsing papers should pay attention to the large constants in parsing realistic lexicalized grammars.



Number of derivations plotted against parsing time.



Parsing with correct elementary tree assigned to each word.

Improving Parsing Efficiency: Conjectures

- estimate priors for lexicalized trees using techniques like n-best SuperTagging (but cf: Chen et al. 1999).

- pruning techniques such as (Goodman 1997, Caraballo and Charniak 1998) extended to LTAG parsers (see: Poller and Becker 1998).

- More experiments . . .