# Statistical Parsing

## Paper presentation:

Eugene Charniak and Mark Johnson (2005). "Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 173–180. DOI: 10.3115/1219840.1219862. URL: http://dx.doi.org/10.3115/1219840.1219862

Çağrı Çöltekin

University of Tübingen
Seminar für Sprachwissenschaft

December 2016

# The general idea

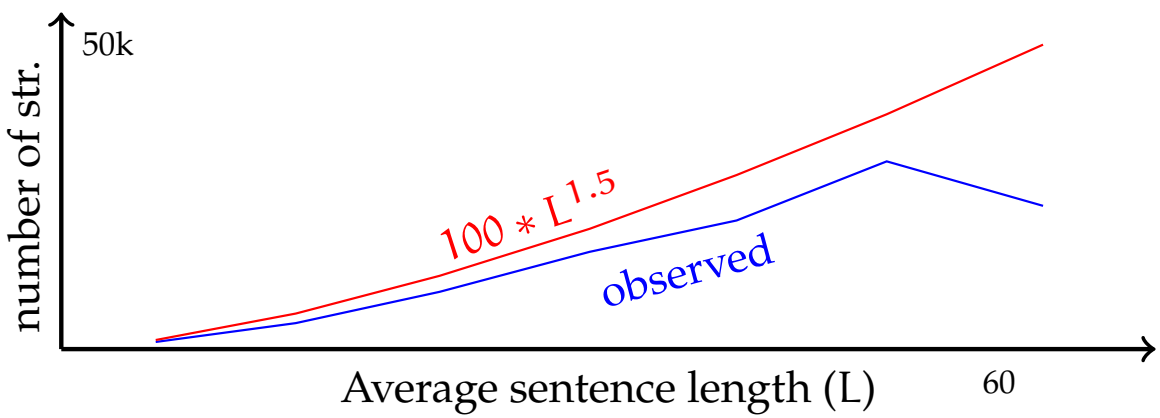
- A two-stage parsing process
  - n-best generative parser with limited/local features
  - discriminative re-ranker with lots of global features
- The problems/issues
  - Efficient n-best parsing is non-trivial
  - The features/methods for re-ranking

# N-best parsing: the problem

- Beam search (n-best parsing) is tricky with dynamic programming:
  - Space complexity becomes an issue, theoretical complexity for bi-lexical grammars: $O(nm^3)$
- Potential solutions:
  - Abandon dynamic programming, use a backtracking parser (slow)
  - Keep dynamic programming with (clever) tricks (potentially resulting in approximate solutions)

# Coarse-to-fine n-best parsing

- First parse with a coarse (non-lexicalized) PCFG
- Prune the parse forest, removing the branches with probability less than a threshold (about $10^{-4}$)
- Lexicalize the pruned parse forest
  - \+ Conditions on information that non-lexicalized PCFG does not have
  - – Increases the number of dynamic programming states. But space complexity seems to stay sub-quadratic (add-hoc calculation: below $100 * L^{1.5}$)



# Getting the n-best parse with dynamic programming

- For each span (CKY chart entry) keep only the n-best non-terminals
- Note: if lists are sorted by probability, combination would not require $n^2$ time
- Space efficiency does not seem to be a problem in practice (only a few MB)
- N-best oracle results:

| n | 1 | 2 | 10 | 25 | 50 |
|---|---|---|---|---|---|
| F-score | 0.897 | 0.914 | 0.948 | 0.960 | 0.968 |

cf. 89.7% F-score of the base parser

# Re-ranking

- Having 50-best parses from the base parser, the idea now is to re-rank them
- Each parse tree is converted a numeric vector of features
- The first feature is the log probability assigned by the base parser
- Other features are assigned based on templates
  - For example, $f_{\text{eat pizza}}(y)$ counts number of times the head of parse tree was 'eat' with complement 'pizza'
  - Note: they distinguish between 'lexical' and 'functional' heads
- After discarding rare features, total number of features is $1\,148\,697$

# Feature templates

| | |
|---|---|
| CoPar | conjunct parallelism |
| CoLenPar | length difference between conjuncts, including a flag indicating final conjuncts |
| RightBranch | number of non-terminals that (do not) lie on the path between root and the rightmost terminal |
| Heavy | categories and their lengths, including whether they are final or they follow a punctuation |
| Neighbors | preterminals before/after the node |
| Rule | whether nodes are annotated with their preterminal heads, their terminal heads and their ancestors' categories |
| NGram | ngrams (bigrams) of the siblings |
| Heads | Head-to-head dependencies |
| LexFunHeads | POS tags of lexical and functional heads |

# Feature templates (cont.)

| | |
|---|---|
| WProj | preterminals with the categories of their closest $\ell$ maximal projection ancestors |
| Word | lexical items with the their closest $\ell$ maximal projection ancestors |
| HeadTree | tree fragments consisting of the local trees consisting of the projections of a preterminal node and the siblings of such projections |
| NGramTree | subtrees rooted in the least common ancestor of $\ell$ contiguous preterminal nodes |

# Results/Conclusions

| | F-score |
|---|---|
| New | 0.9102 |
| Collins | 0.9037 |

- Also better than $0.907$ reported by Bod (2003), but more efficient
- $13\,\%$ error reduction over the base parser (or maybe even $18\,\%$, considering PTB is not perfect)
- The parser is publicly available

# Results/Conclusions

| | F-score |
|---|---|
| New | 0.9102 |
| Collins | 0.9037 |

- Also better than $0.907$ reported by Bod (2003), but more efficient
- $13\,\%$ error reduction over the base parser (or maybe even $18\,\%$, considering PTB is not perfect)
- The parser is publicly available

- State-of-the art parsing of PTB with generative n-best parser, followed by discriminative re-ranking

# Parameter estimation

- They use a maximum-entropy model (=logistic regression) for re-ranking
- Feature weights are calculated by minimizing L2 regularized negative log-likelihood
- A slight divergence: the gold-standard parse is not always in n-best list
  - Pick the tree(s) that are most similar to gold-standard tree (with best F-score)
  - In case of ties (multiple best trees), prefer the solution maximizing the log likelihood of all

# Summary

- Accurate generative parser that breaks down rules
- Does well on 'core' dependencies, adjuncts and coordination are the main sources of error
- Either conditioning on adjacency or subcategorization is needed for good accuracy
- The models work well with flat dependencies
- Breaking down the rules have good properties (can use rules that were not seem in the training)

# Bibliography


Bod, Rens (2003). "An Efficient Implementation of a New DOP Model". In: *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*. EACL '03. Budapest, Hungary: Association for Computational Linguistics, pp. 19–26. ISBN: 1-333-56789-0. DOI: 10.3115/1067807.1067812. URL: http://dx.doi.org/10.3115/1067807.1067812.

Charniak, Eugene and Mark Johnson (2005). "Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 173–180. DOI: 10.3115/1219840.1219862. URL: http://dx.doi.org/10.3115/1219840.1219862.

Collins, Michael and Terry Koo (2005). "Discriminative Reranking for Natural Language Parsing". In: *Computational Linguistics* 31.1, pp. 25–70. ISSN: 0891-2017. DOI: 10.1162/0891201053630273. URL: http://dx.doi.org/10.1162/0891201053630273.