University of California C A L N L P Berkeley

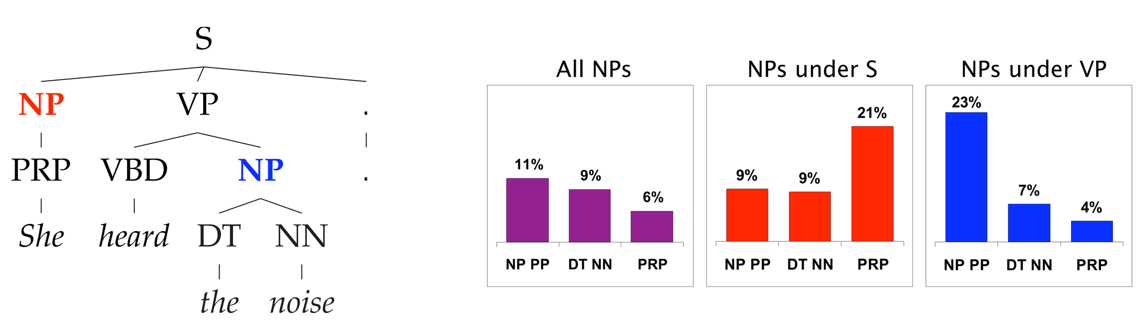
Discriminative Log-Linear Grammars with Latent Variables

Slav Petrov and Dan Klein
University of California, Berkeley

Motivation

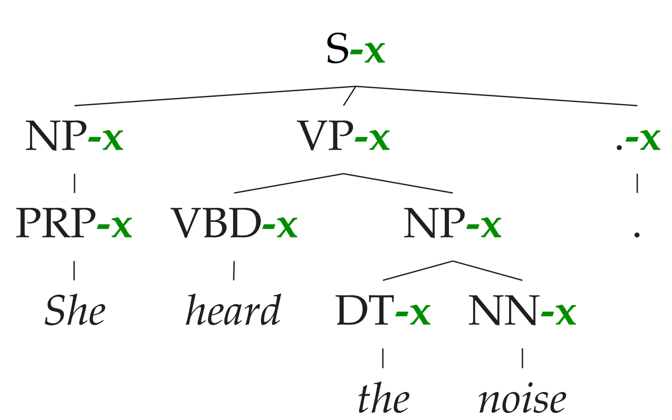
Grammar Learning

The observed treebank categories are too coarse because the rewrite probabilities depend on context.



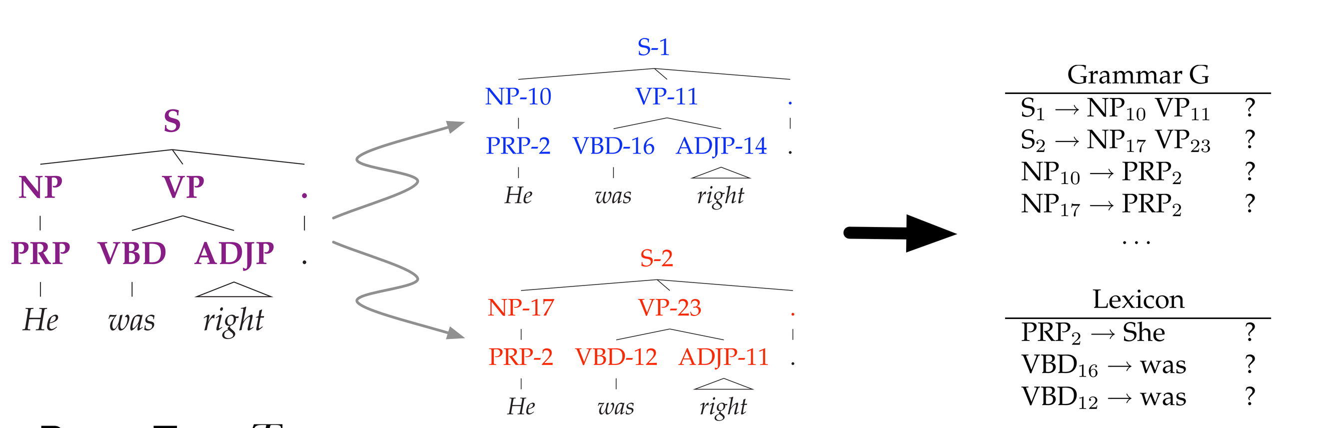
Grammars with Latent Variables

Given a treebank over a set of categories learn an optimally refined grammar for parsing.



Automatic Grammar Refinement

Refine the observed trees with latent variables and learn subcategories.



Parse Tree T Sentence w Parse Derivations $t_i : T$ Model Parameters θ

$$P_\theta(t|w) = \frac{1}{Z(\theta,w)} \prod_{X\to\gamma\in t} e^{\theta_{X\to\gamma}} = \frac{1}{Z(\theta,w)} e^{\theta^\top f(t)}$$

Training

Generative Training

Maximize the joint likelihood:

$$\mathcal{L}_{joint}(\theta) = \log \prod_i P_\theta(T_i, w_i) = \log \prod_i \sum_{t:T_i} P_\theta(t, w_i)$$

$$\theta^* = \operatorname*{argmax}_\theta \left(\log \prod_i \sum_{t:T_i} P_\theta(t, w_i) \right)$$

The parameters can be learned with an Expectation Maximization algorithm. The E-Step involves computing expectations over derivations corresponding to the observed trees. These expectations are normalized in the M-Step to update the rewrite probabilities:

$$\phi_{X\to\gamma} = \frac{\sum_T \mathbb{E}_\theta[f_{X\to\gamma}(t)|T]}{\sum_{\gamma'} \sum_T \mathbb{E}_\theta[f_{X\to\gamma'}(t)|T]}$$

Computing expectations over derivations corresponding to the observed trees can be done in linear time (in the number of words).

Discriminative Training

Maximize the conditional likelihood:

$$\mathcal{L}_{cond}(\theta) = \log \prod_i P_\theta(T_i|w_i) = \log \prod_i \sum_{t:T_i} P_\theta(t|w_i)$$

$$\theta^* = \operatorname*{argmax}_\theta \left(\log \prod_i \sum_{t:T_i} P_\theta(t|w_i) \right)$$

The parameters can be learned with a numerical gradient based method (e.g. L-BFGS). Computing the gradient involves calculating expectations over derivations corresponding to the observed trees, as well as over all possible trees:

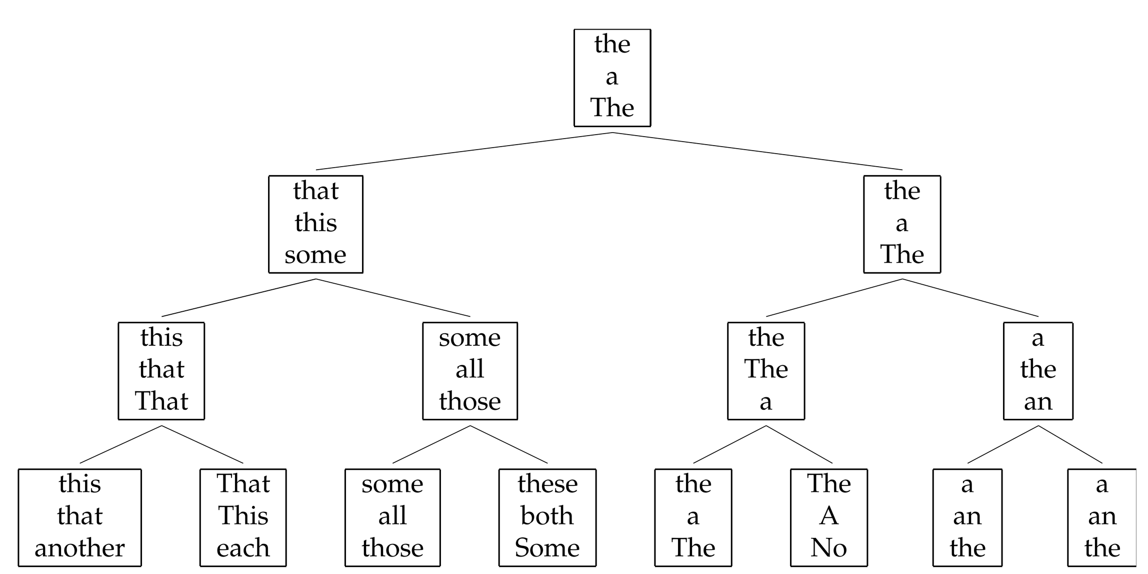
$$\frac{\partial \mathcal{L}_{cond}(\theta)}{\partial \theta_{X\to\gamma}} = \sum_i \left(\mathbb{E}_\theta\left[f_{X\to\gamma}(t)|T_i\right] - \mathbb{E}_\theta\left[f_{X\to\gamma}(t)|w_i\right] \right)$$

Computing expectations over derivations corresponding to all possible trees involves parsing the training corpus, which requires cubic time (in the number of words).

Efficient Estimation

Hierarchical Splitting

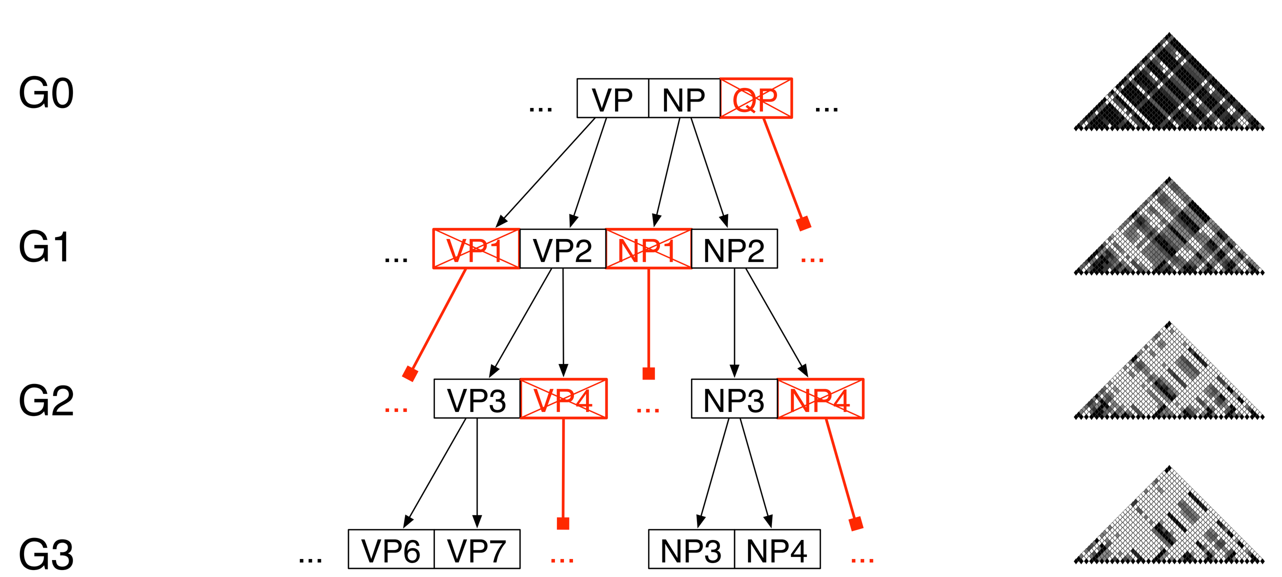
Repeatedly split each category in two and retrain the grammar, initializing with the previous grammar.



Linguistically interpretable subcategories emerge in the course of hierarchical refinement.

Feature Count Approximation

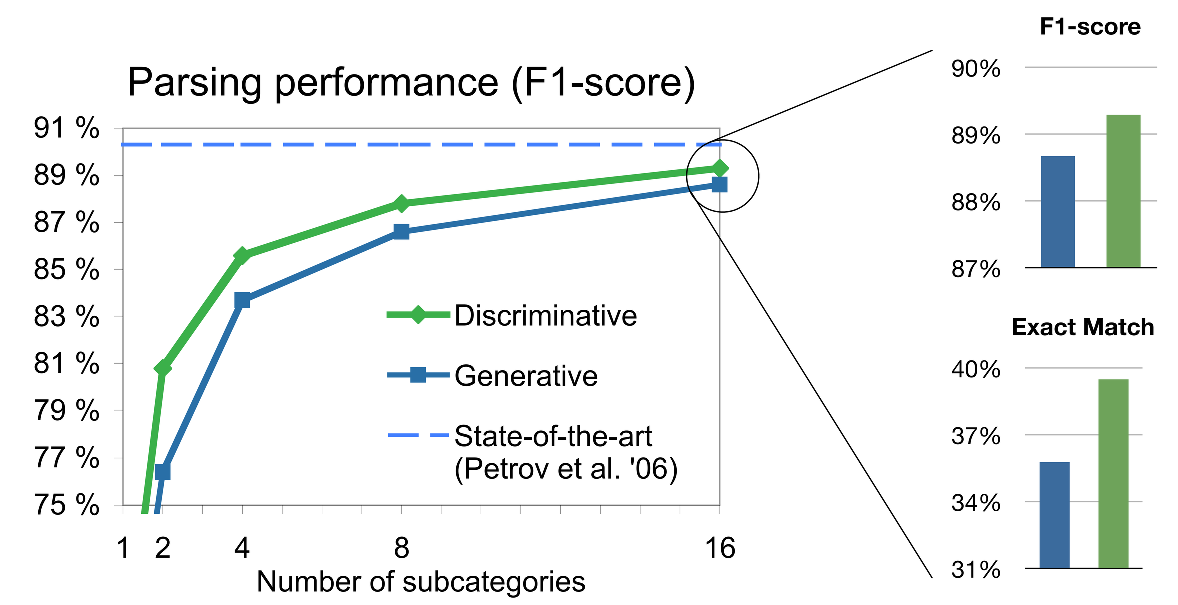
Use predictions from hierarchical coarse-to-fine parsing to prune unlikely chart items, setting their expectations to zero.



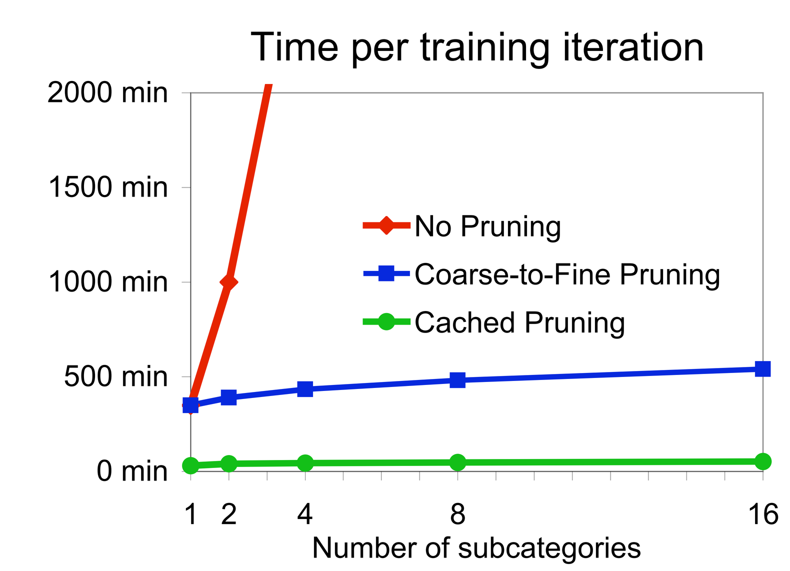
Coarse-to-fine pruning leads to few brackets with non-zero feature counts. In cached pruning only the finest level of the hierarchy is updated.

Results

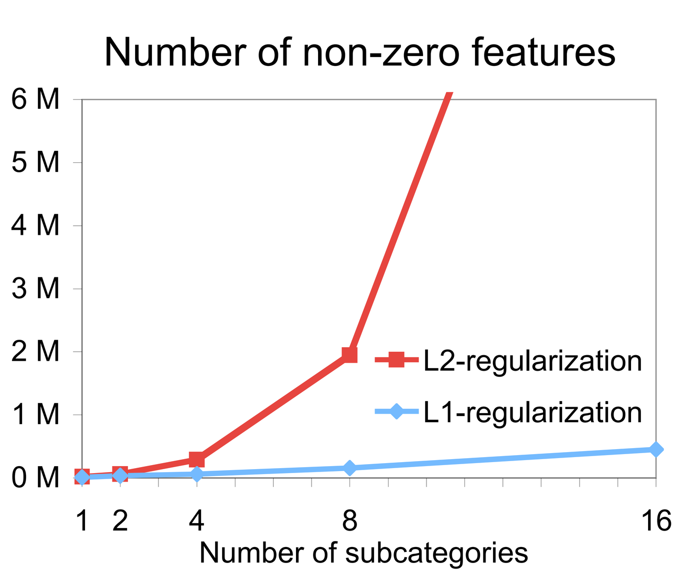
Grammars were trained on the Wall Street Journal section of the Penn Treebank using the standard splits. The training set contains roughly 1M words in 40K sentences.



Discriminative training is superior to generative training in terms of F1-score and exact match.



Coarse-to-Fine pruning gives a tremendous speed-up over exhaustive parsing, but only cached pruning makes large scale training of discriminative grammars practically feasible.



L1-regularization leads to extremely sparse grammars without decreasing the parsing performance.