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.

Generative Training
Maximize the joint likelihood:
\[
\mathcal{L}_{\text{joint}}(\theta) = \log \prod_{i} P_b(T_i, w_i) = \log \sum_{t \in T_i} P_g(t, w_i)
\]

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:

\[
\phi_{X \rightarrow Y} = \frac{\sum_i P_g(t, w_i)}{\sum_j \sum_i P_g(t, w_i)}
\]
Computing expectations over derivations corresponding to the observed trees can be done in linear 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.

Discriminative Training
Maximize the conditional likelihood:
\[
\mathcal{L}_{\text{cond}}(\theta) = \log \prod_{i} P_b(T_i|w_i) = \log \sum_{t \in T_i} P_g(t|w_i)
\]

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:

\[
\theta^{*} = \arg\max_{\theta} \left( \log \prod_{t \in T_i} P_g(t|w_i) \right)
\]

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

Automatic Grammar Refinement
Refine the observed trees with latent variables and learn subcategories.

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

Grammar with Latent Variables
Given a treebank over a set of categories learn an optimally refined grammar for parsing.

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.

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.

Coarse-to-Fine Pruning significantly decreases the parsing performance.