Better Informed Training of Latent Syntactic Features

Markus Dreyer and Jason Eisner
Center for Language and Speech Processing, Johns Hopkins University

Introduction

1. Automatically refine the nonterminals in a treebank, by unsupervised learning
2. NP becomes NP[1], NP[2], ... which behave differently (e.g. subject, object)
3. Orthogonal strategies:
   - Model: Add such features to nonterminals in such a way that they respect patterns of feature passing - each node's nonterminal features are either identical to, or independent of, those of its parent. This new model learned interesting linguistic features, but did not improve parsing results.
   - Training: Split nonterminals selectively only as needed
   - Data: Treebank preprocessing (markovization)
   - Dramatically reduce model size, but maintain high parsing accuracy (compared to Matsuzaki (2005))

Improve nonterminal tagset

Previous model: Constrain EM to learn refined nonterminals

a. Previous approaches had introduced manual nonterminal splits (Collins (1996)) split S into S and SG, Klein and Manning (2003) split several POS tags into finer-grained tags.
   Matsuzaki et al (2005) introduce PCFG-LA model: systematic and automatic split of nonterminals in treebank
   An annotation on each nonterminal token is learned - an unspecified and uninterpreted integer that distinguishes otherwise identical nonterminals: S becomes S[0], S[1], S[2], ...

b. New model: Constrain EM even more

1. Similar to previous model (PCFG-LA, above), but models inheritance of features within the tree
2. A node's feature is either copied from its parent or independent of its parent
3. This linguistic constraint models agreement, reduces runtime and decreases the number of parameters to be learned.
4. Since we have less parameters we can increase the number of splits. The number of parameters we needed for 8 splits in the previous model can here be used to make 80 splits. NP is split into NP[1], NP[2], ... NP[80], and similarly for other nonterminals.
5. Additional parameters control feature passing:
   P(pass to head | rule), P(pass to nonhead | rule), P(pass to both | rule), P(pass to neither | rule), P_{non}(feature | nonterminal)

Don't split everything at once - and don't split everything!

a. Start with simple model (every nonterminal split in two), learn, then selectively make more splits, learn...

b. Analogy to deterministic annealing:
   In clustering by deterministic annealing (DA), number of clusters is gradually increased. Entropy of P(point, cluster) constrained to be high in the beginning, then entropy gradually lowered; clusters, initially uniform, start to move apart.
   Here, we use simplified version (no entropy constraint):
   Use Jensen-Shannon Divergence (a.k.a. KL divergence to the mean) to decide if P(S[1] | S[2]) and P(S[3] | S[4]) have moved apart.
   Main differences to Petrov et al (2006) (which was written and submitted independently at the same time)
   - They split all, learn, merge, split all, learn, merge...
   - We split some, learn, split some, learn,...
   - Different measure used

Results

Results on devset. Basic models are trained on a non-markovized treebank (as in Matsuzaki (2005)); all others trained on a markovized treebank. PCFG-LA is the baseline. The best model (PCFG-LA split some, F1=87.31) has also been decoded on the final test set, reaching F1=88.25. The Inherit model did not help, but markovization and splitting only some nodes did.

L is the number of splits. e.g. L=2 for NP split into [NP[1] and NP[2]]

References: