Better Informed Training of Latent Syntactic Features

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

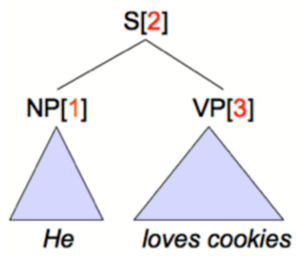
1 Introduction

- Automatically refine the nonterminals in a treebank, by unsupervised learning
- NP becomes NP[1], NP[2], …, which behave differently (e.g. subject, object)
- Orthogonal strategies:
  - Model: Add such features to nonterminals in such a way that they respect patterns of linguistic 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 ▶ Helped parsing accuracy
  - Data: Treebank preprocessing (markovization)
- Dramatically reduce model size, but maintain high parsing accuracy (compared to Matsuzaki (2005))

2 Improve nonterminal tagset

a Previous model: Constrain EM to learn refined nonterminals

- Previous approaches had introduced manual nonterminal splits (Collins (1996) split S split 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], …

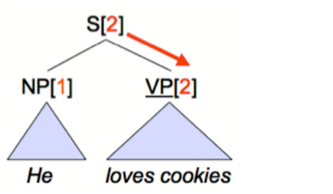


$$P(\text{tree}) = P(\texttt{ROOT} \rightarrow \texttt{S}[2]) \times P(\texttt{S}[2] \rightarrow \texttt{NP}[1]\ \texttt{VP}[3]) \times P(\texttt{NP}[1] \rightarrow^* \texttt{He}) \times P(\texttt{VP}[3] \rightarrow^* \texttt{loves cookies})$$

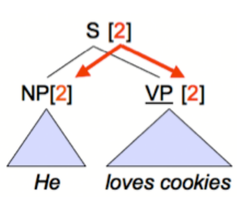
Penn Treebank tree with automatic annotations in red.
The parser sums over all possible annotations.

b New model: Constrain EM even more

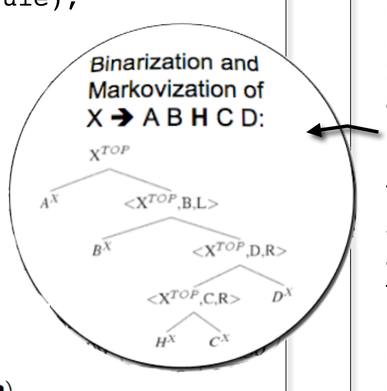
- Similar to previous model (PCFG-LA, above), but models inheritance of features within the tree
- A node's feature is either copied from its parent or independent of its parent
- This linguistic constraint models agreement, reduces runtime and decreases the number of parameters to be learned.
- Since 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.
- 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_{ann}`(feature | nonterminal)`



$P(\texttt{ROOT} \rightarrow \texttt{S}[2]) \times P(\texttt{S}[2] \rightarrow \texttt{NP}\ \underline{\texttt{VP}}) \times P(\textbf{pass to head} \mid \texttt{S} \rightarrow \texttt{NP}\ \underline{\texttt{VP}}) \times P_{ann}(1 \mid \texttt{NP}) \times P(\texttt{NP}[1] \rightarrow^* \texttt{He}) \times P(\texttt{VP}[2] \rightarrow^* \texttt{loves cookies})$



$P(\texttt{ROOT} \rightarrow \texttt{S}[2]) \times P(\texttt{S}[2] \rightarrow \texttt{NP}\ \underline{\texttt{VP}}) \times P(\textbf{pass to both} \mid \texttt{S} \rightarrow \texttt{NP}\ \underline{\texttt{VP}}) \times P(\texttt{NP}[2] \rightarrow^* \texttt{He}) \times P(\texttt{VP}[2] \rightarrow^* \texttt{loves cookies})$



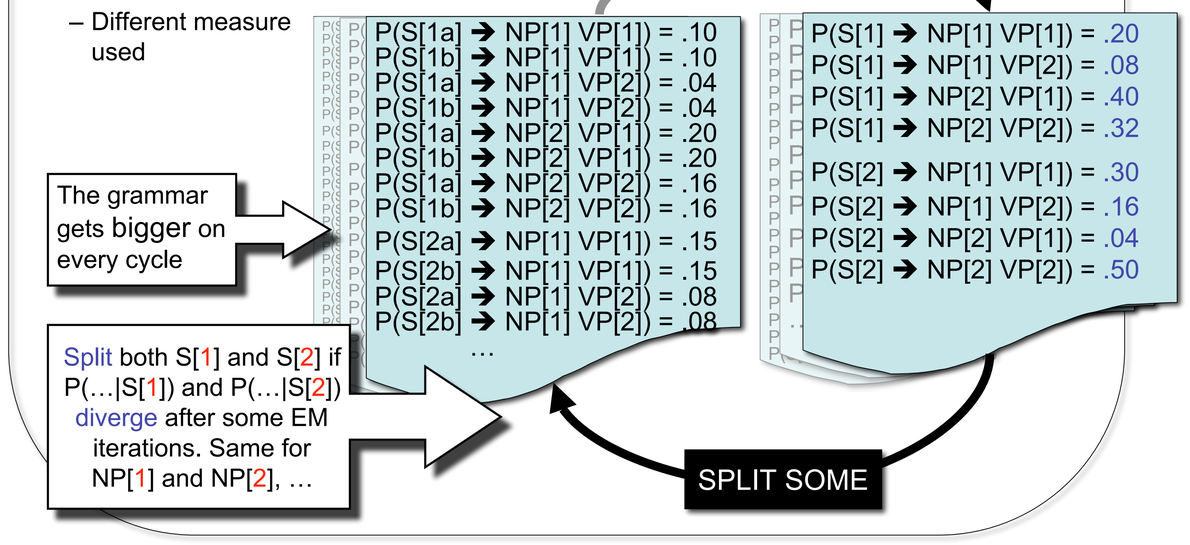
Two trees following different passpatterns and their probabilities. In the left tree, S passes its feature to the head child (underlined, given by Collins' rules) while the other child is free to pick its own feature according to P_{ann}(feature | nonterminal). In the right tree, the feature is passed to both children.

c What was learned?

- Plural/Singular: NP[2] picks up more plural nouns than NP[1]. This effect is stronger in our more constrained INHERIT model, which is also more likely to pass the plural/singular feature to both children: Det. and noun must agree.
- A tensed auxiliary feature is learned: This feature on a VP makes it expand as `V_Aux VP`. It is passed to the head V_Aux, causing it to expand as a form of be, have, or do.
- Subordinate conjunctions (*while, if*) in IN[1], prepositions (*under, after*) in IN[2]
- Upper-case conjunctions at beginning of sentence (*And, But*) vs mid-sentence conjunctions

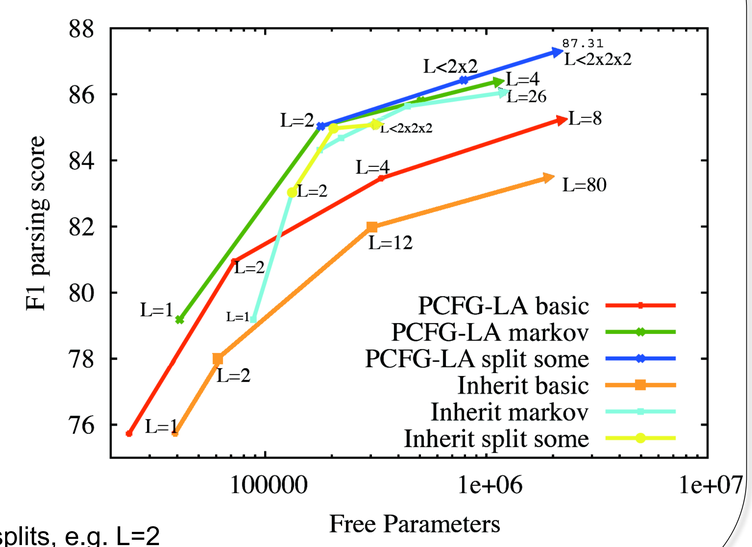
3 Don't split everything at once - and don't split everything!

- Start with simple model (every nonterminal split in two), learn, then selectively make more splits, learn, …
- 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):
  If two distributions, e.g. P(…|S[1]) and P(…|S[2]) move apart during EM learning, then split them further into P(…|S[1a]), P(…|S[1b]), P(…|S[2a]), P(…|S[2b]).
- Use Jensen-Shannon Divergence (a.k.a. KL divergence to the mean) to decide if P(…|S[1]) and P(…|S[2]) 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



4 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*, F_1=87.31) has also been decoded on the final test set, reaching F_1=86.25. 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: Collins (1996), Goodman (1997), Johnson (1998), Klein and Manning (2003), Matsuzaki et al (2005), Petrov et al (2006), Prescher (2005), Rose (1998), …