**BACKGROUND**

- Standard perceptron (Liang et al ’06)
- MERT (2003)
- MIRA (2007–12)
- PRO (2011)
- HOELS (2013)

Our violation-fixing perceptron with truly sparse features

- Training set (>100k sentences)
- Test set (~1k sentences)

**VIOLATION FIXING PERCEPTRON**

- Violation-Fixing Perceptron (Huang+ ’12) is tailored for inexact search
  1. Violation: incorrect prefix scores higher than gold-standard prefix
  2. Guaranteed to converge if each update is valid (i.e., on a violation)
  3. Examples: early update (Collins+Roark ’04) and max-violation (Huang+ ’12)
  4. Standard update does not converge with many invalid updates

- Extend violation-fixing perceptron to handle latent variables (derivations):
  1. Early update: update when no correct derivations survive
  2. Max-violation: update at the bin where the violation is maximum
  4. Local update (also from Liang et al): update towards the derivation with highest sentence-level BLEU in the n-best list

- Liang et al attribute their failure to gold derivations from “bad” rules. But we give a theoretically sound explanation: search errors cause invalid updates.

**FEATURE DESIGN**

1. Dense features: 11 standard phrase-based features from Moses
2. Sparse Features
   - rule-identification features (unique id for each rule)
   - word-edges: lexicalized local translation context within a rule
   - non-local features: dependency between consecutive rules
3. Feature Backoff: Brown clusters, POS tags, Chinese chars/types, etc.

**EXPERIMENTAL RESULTS**

- Comparison of Update Methods
- Feature performance breakdown
- Feature Counts & Contributions
- Minibatch Parallelization
- Results on Large CH-EN (FBIS)
- Results on SP-EN (with 1-ref)

**KEY REFERENCES**