Monte Carlo Syntax Marginals for Exploring and Using Dependency Parses (NAACL'18)

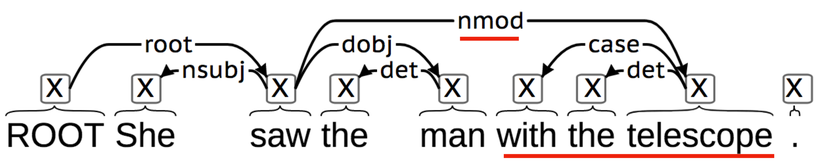
Katherine A. Keith, Su Lin Blodgett, and Brendan O'Connor
College of Information and Computer Sciences, University of Massachusetts Amherst

Summary

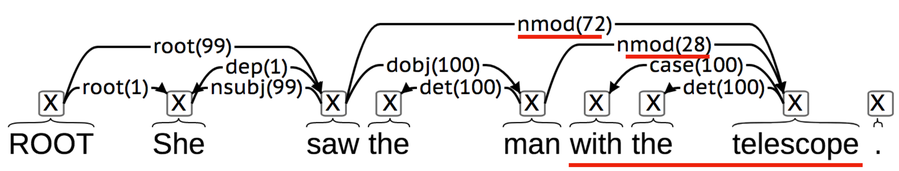
- **Goal:** Communicate inherent syntactic ambiguity
- **Key idea:** Replace greedy algorithm with fast and simple method—*transition sampling* from the full joint distribution of parse trees
- **New task:** Introduce *dependency path prediction,* predicting a set of length-*d* dependency paths for a sentence
- **Evaluation:** Our method (1) dominates the greedy algorithm on above task, (2) provides better error analysis, (3) improves performance on two downstream tasks

Example of inherent ambiguity

Greedy



100-sample marginals



Transition-sampling for dependency parsing

To get one parse tree sample, $y(a_1, \cdots, a_n)$

- Initialize state S_0
- For $n = 1, 2, \ldots$:
  - (A) Choose a_n
  - (B) $S_n := \text{Update}(S_{n-1}, a_n)$
  - (C) Break if $\text{InEndState}(S_n)$

Greedy

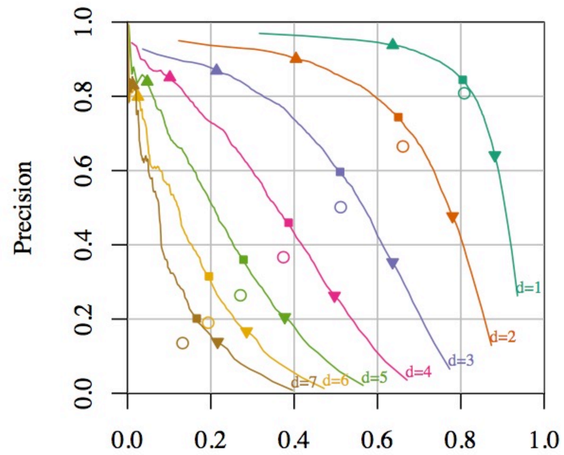
$$a_n := \arg\max_{a_n} p(a_n \mid S_{n-1})$$

Transition sampling

$$a_n \sim p(a_n \mid S_{n-1})$$

Dependency *path* prediction task

Previous work only evaluated length-1 paths (LAS)



Path Len.	Greedy F1	Marginal Max F1	Marginal Thresh.	MC-MAP F1
1	0.808	0.824	0.45	0.807
2	0.663	0.694	0.44	0.660
3	0.506	0.550	0.34	0.501
4	0.370	0.420	0.28	0.363
5	0.268	0.314	0.25	0.262
6	0.192	0.241	0.25	0.188
7	0.134	0.182	0.17	0.131

▲ Conf≥0.9 ▼ Conf≥0.1

F1-scores for retrieved set of all length-*d* paths with 100 MC samples against gold (UD 1.3 dev set) parse tree

Error analysis via entropy calculations

Sentence	Domain Size	Top 3 Freq.	Entropy
In Ramadi , there was a big demonstration .	3	[98, 1, 1]	0.112
US troops there clashed with guerrillas in a fight that left one Iraqi dead .	40	[33, 11, 6]	2.865
The sheikh in wheel - chair has been attacked with a F - 16 - launched bomb .	98	[2, 2, 1]	4.577

Whole-tree entropy

$$H(p) = -\sum_{y \in \mathcal{Y}(x)} p(y \mid x) \log p(y \mid x)$$

$$\approx H(\tilde{p}) = -\sum_{y \in \tilde{\mathcal{Y}}(x)} \frac{c(y)}{S} \log \frac{c(y)}{S}.$$

Improving downstream applications

(1) Classifying names of persons killed by police (Keith et al. 2017)

Method	F1
RPI-JIE	0.170
Greedy	0.215
1 samp.	0.212
10 samp.	0.219
100 samp.	**0.222**

(2) Semantic role assignment

English OntoNotes v5.0

Method	Accuracy
Baseline (most common arg)	0.393
Greedy	0.496
100-sample	**0.529**