The Infinite Tree

Finkel et al.

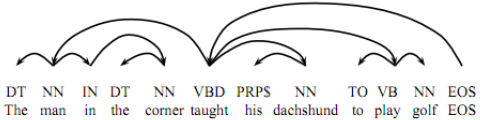
presented by Vazheh Moussavi

Overview

- Introduction
- Finite Trees
- Dirichlet Process & HDP
- Infinite Trees
- Inference
- Experiments
- Conclusion

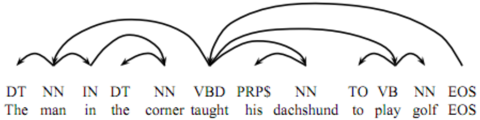
Introduction/Motivation

- want to model syntactic dependency tree:



- states: "tags", or word categories (noun, plural noun, etc.)
- observations: words
- structure: tag-pair dependencies (syntax)
- problem: word categories are too coarse, don't give enough discriminative power for automatic parsers
- fix #1: give tags their actual lexical form
- fix #2: manually split tagset
- fix #3: learn splits with heuristics

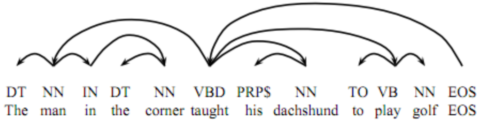
- want to model syntactic dependency tree:



- states: "tags", or word categories (noun, plural noun, etc.)
- observations: words
- structure: tag-pair dependencies (syntax)
- problem: word categories are too coarse, don't give enough discriminative power for automatic parsers
- fix #1: give tags their actual lexical form
- fix #2: manually split tagset
- fix #3: learn splits with heuristics

Introduction/Motivation

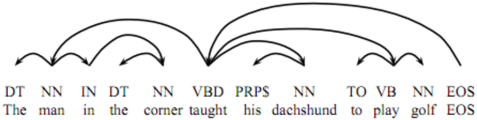
- want to model syntactic dependency tree:



- states: “tags”, or word categories (noun, plural noun, etc.)
- observations: words
- structure: tag-pair dependencies (syntax)
- problem: word categories are too coarse, don’t give enough discriminative power for automatic parsers
- fix #1: give tags their actual lexical form
- fix #2: manually split tagset
- fix #3: learn splits with heuristics

Introduction/Motivation

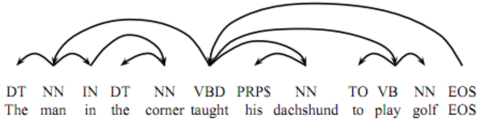
- want to model syntactic dependency tree:



- states: “tags”, or word categories (noun, plural noun, etc.)
- observations: words
- structure: tag-pair dependencies (syntax)
- problem: word categories are too coarse, don’t give enough discriminative power for automatic parsers
- fix #1: give tags their actual lexical form
- fix #2: manually split tagset
- fix #3: learn splits with heuristics

Introduction/Motivation

- want to model syntactic dependency tree:



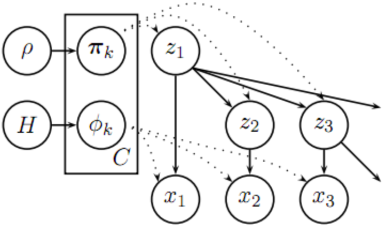
- states: "tags", or word categories (noun, plural noun, etc.)
- observations: words
- structure: tag-pair dependencies (syntax)
- problem: word categories are too coarse, don't give enough discriminative power for automatic parsers
- fix #1: give tags their actual lexical form
- fix #2: manually split tagset
- fix #3: learn splits with heuristics

Proposed Solution

- "Infinite Tree": recursive branching structure, potentially infinite states
- three different children dependency forms
- some notation:

$$
\begin{aligned}
t &: \text{tree and root node} \\
c(t) &: \text{list of } t\text{'s children} \\
c_i(t) &: i^{th} \text{ child of } t \\
p(t) &: \text{parent of } t \\
z_t \in \{1, \ldots, C\} &: \text{state of } t \\
x_t &: \text{observation (word) at } t
\end{aligned}
$$

Finite Trees



- (conjugate) dirichlet prior H on observation parameters: $\phi_k \mid H \sim H$
- emissions parameterized by F: $x_t \mid z_t \sim F(\phi_{z_t})$
- state z'_t for child of t depends on z_t, multinomial: $z'_t \mid z_t \sim \text{Multinomial}(\pi_{z_t})$
- uniform prior on multinomial state parameters: $\pi_k \mid \rho \sim \text{Dir}(\rho, \ldots, \rho)$

Finite Trees, cont.

- Independent Children: (given parent)

$$P_{tr}(t) = P(x_t \mid z_t) \prod_{t' \in c(t)} P(z'_t \mid z_t) P_{tr}(t')$$

- Simultaneous Children: (no independence assumed at all)

$$P_{tr}(t) = P(x_t \mid z_t) P\left(\left(z'_t\right)_{t' \in c(t)} \mid z_t\right) \prod_{t' \in c(t)} P_{tr}(t')$$

- $c_t \sim \lambda_k$ (multinomial, learned)
- Markov Children:

$$P_{tr}(t) = P(x_t \mid z_t) \prod_{i=1}^{|c(t)|} P\left(z_{c_i(t)} \mid z_{c_{i-1}(t)}, z_t\right) P_{tr}(t')$$

Finite Trees, cont.

- Independent Children: (given parent)

$$P_{tr}(t) = P(x_t \mid z_t) \prod_{t' \in c(t)} P\left(z'_t \mid z_t\right) P_{tr}\left(t'\right)$$

- Simultaneous Children: (no independence assumed at all)

$$P_{tr}(t) = P(x_t \mid z_t) P\left(\left(z'_t\right)_{t' \in c(t)} \mid z_t\right) \prod_{t' \in c(t)} P_{tr}\left(t'\right)$$

- $c_t \sim \lambda_k$ (multinomial, learned)
- Markov Children:

$$P_{tr}(t) = P(x_t \mid z_t) \prod_{i=1}^{|c(t)|} P\left(z_{c_i(t)} \mid z_{c_{i-1}(t)}, z_t\right) P_{tr}\left(t'\right)$$

Finite Trees, cont.

- Independent Children: (given parent)

$$P_{tr}(t) = P(x_t \mid z_t) \prod_{t' \in c(t)} P\left(z'_t \mid z_t\right) P_{tr}\left(t'\right)$$

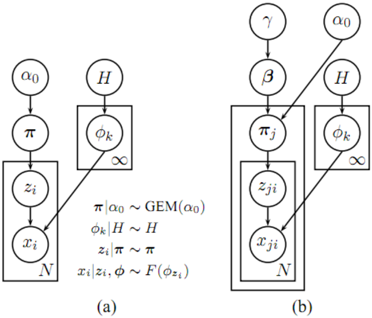
- Simultaneous Children: (no independence assumed at all)

$$P_{tr}(t) = P(x_t \mid z_t) P\left(\left(z'_t\right)_{t' \in c(t)} \mid z_t\right) \prod_{t' \in c(t)} P_{tr}\left(t'\right)$$

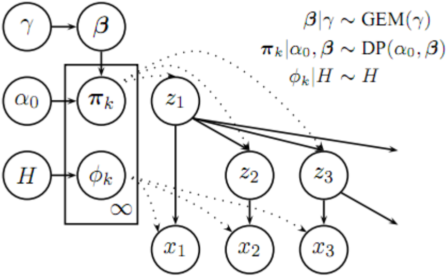
- $c_t \sim \lambda_k$ (multinomial, learned)
- Markov Children:

$$P_{tr}(t) = P(x_t \mid z_t) \prod_{i=1}^{|c(t)|} P\left(z_{c_i(t)} \mid z_{c_{i-1}(t)}, z_t\right) P_{tr}\left(t'\right)$$

Dirichlet Process & HDP



Infinite Trees



- mostly same, now have dirichlet process as prior

Infinite Trees, cont.

- Independent Children:
- mostly same, now infinite states, $\pi_k \sim DP(\alpha_0, \beta)$
- HDP-HMM is same, but with only one child
- Simultaneous Children:
- sparsity exacerbated even further

$$P\left(\left(z_t'\right)_{t' \in c(t)} \mid \pi\right) = \prod_{t' \in c(t)} P(z_{t'} \mid \pi) = \prod_{t' \in c(t)} \pi_{z_t'}$$

- $\lambda_k \mid \zeta, L_k \sim DP(\zeta, L_k)$, L_k is a deterministic function of π_k, acts as base measure
- Markov Children:
- same, now $\pi_{ki} \sim DP(\alpha_0, \beta)$ (from i to k)

Infinite Trees, cont.

- Independent Children:
- mostly same, now infinite states, $\pi_k \sim DP(\alpha_0, \beta)$
- HDP-HMM is same, but with only one child
- Simultaneous Children:
- sparsity exacerbated even further

$$P\left((z'_t)_{t' \in c(t)} \mid \pi \right) = \prod_{t' \in c(t)} P(z_{t'} \mid \pi) = \prod_{t' \in c(t)} \pi_{z'_t}$$

- $\lambda_k \mid \zeta, L_k \sim DP(\zeta, L_k)$, L_k is a deterministic function of π_k, acts as base measure
- Markov Children:
- same, now $\pi_{ki} \sim DP(\alpha_0, \beta)$ (from i to k)

Infinite Trees, cont.

- Independent Children:
- mostly same, now infinite states, $\pi_k \sim DP(\alpha_0, \beta)$
- HDP-HMM is same, but with only one child
- Simultaneous Children:
- sparsity exacerbated even further

$$P\left(\left(z'_t \right)_{t' \in c(t)} \mid \pi \right) = \prod_{t' \in c(t)} P(z_{t'} \mid \pi) = \prod_{t' \in c(t)} \pi_{z'_t}$$

- $\lambda_k \mid \zeta,\ L_k \sim DP(\zeta, L_k)$, L_k is a deterministic function of π_k, acts as base measure
- Markov Children:
- same, now $\pi_{ki} \sim DP(\alpha_0, \beta)$ (from i to k)

Inference

- sampler: (1) sample state assignments z, (2) sample counts m, (3) sample global stick β
- m_{jk}: number of elements from π_k corresponding to β_j
- n_{jk}: number of observations with state k and parent state j
- marginal counts represented with dot $(\cdot)$

Inference, cont., 1

- sampling z

$$P\left(z_t = k \mid z^{-t}, \beta\right) \propto P\left(z_t = k, \left(z_{t'}\right)_{t' \in s(t)} \mid z_p(t)\right)$$
$$\cdot P\left(\left(z_{t'}\right)_{t' \in s(t)} \mid z_t = k\right) \cdot f_k^{-x_t}\left(x_t\right)$$

- $$f_k^{-x(t)}\left(x_t\right) = \frac{n^{\cdot}_{x_t k} + \rho}{n^{\cdot}_k + N\rho}$$

Inference, cont., 2

SAMPLEM(j,k)

if $n_{jk} = 0$
  then $m_{jk} = 0$
  else $m_{jk} = 1$
    for $i \leftarrow 2$ **to** n_{jk}
      do if rand() $< \frac{\alpha_0}{\alpha_0 + i - 1}$
        then $m_{jk} = m_{jk} + 1$
return m_{jk}

- sample β:

$$(\beta_1, \ldots, \beta_K, \beta_u) \sim Dir\left(m_{\cdot 1}, \ldots, m_{\cdot K}, \alpha_0\right)$$

Experiments

- only have structure, not tags:

Experiments, cont.

- results: learning and splitting tags

Model	ρ	# Classes	Acc.	MI	F1
Indep.	0.01	943	67.89	2.00	48.29
	0.001	1744	73.61	2.23	40.80
	0.0001	2437	74.64	2.27	39.47
Simul.	0.01	183	21.36	0.31	21.57
	0.001	430	15.77	0.09	13.80
	0.0001	549	16.68	0.12	14.29
Markov	0.01	613	68.53	2.12	49.82
	0.001	894	75.34	2.31	48.73

Table 1: Results of part unsupervised POS tagging on the different models, using a greedy accuracy measure.

Model	ρ	Accuracy
Baseline	–	85.11
Independent	0.01	86.18
	0.001	85.88
Markov	0.01	87.15
	0.001	87.35

Table 2: Results of untyped, directed dependency parsing, where the POS tags in the training data have been split according to the various models. At test time, the POS tagging and parsing are done simultaneously by the parser.