TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Theoretical Computer Science, Chair of Foundations of Programming

LEARNING PRUNING POLICIES FOR LINEAR CONTEXT-FREE REWRITING SYSTEMS

INF-PM-FPG

Andy Püschel

Dresden, July 20, 2018

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Motivation

Example:

- Weighted Deductive Parsing for LCFRS
- Sentence w = Nun werden sie umworben .
- Parser computes the highest scoring derivation $\hat{d}$

Linear Context-free Rewriting System

Definition

A *linear context-free rewriting system* is a tuple $G = (N, \Sigma, \Xi, P, S)$ where

- N is a finite nonempty $\mathbb{N}$-sorted set (nonterminal symbols),
- Σ is a finite set (terminal symbols) (with $\forall l \in \mathbb{N} : \Sigma \cap N_l = \emptyset$),
- Ξ is a finite nontempty set (variable symbols) (with $\Xi \cap \Sigma = \emptyset$ and $\forall l \in \mathbb{N} : \Xi \cap N_l = \emptyset$),
- P is a *set of production rules* of the form $\rho = \phi \to \psi$ where
  - $\phi = A(\alpha_1, \ldots, \alpha_l)$ (called left-hand side of ρ) where $l \in \mathbb{N}$, $A \in N_l$, $\alpha_1, \ldots, \alpha_l \in (\Sigma \cup \Xi)^*$ and
  - $\psi = B_1(X_1^{(1)}, \ldots, X_{l_1}^{(1)}) \ldots B_m(X_1^{(m)}, \ldots, X_{l_m}^{(m)})$ (called right-hand side of ρ) where $m \in \mathbb{N}$, $B_1 \in N_{l_1}, \ldots, B_m \in N_{l_m}$, $X_j^{(i)} \in \Xi$ for $1 \leq i \leq m, 1 \leq j \leq l_i$

  and for every $X \in \Xi$ occurring in ρ we require that X occurs exactly once in the left-hand side of ρ and exactly once in the right-hand side of ρ, and
- $S \in N_1$ (initial nonterminal symbol).

Example PLCFRS

PLCFRS (G, p) and $G = (N, \Sigma, \Xi, P, S)$ where

- $N = \{VROOT, S, VP, ADV, VAFIN, VAINF, VVINF, PPER, VVPP, \$, \ldots\}$,
- $\Sigma = \{Nun, werden, sie, umworben, ., \ldots\}$ and
- $P = \{\ldots,$

$$
\begin{aligned}
ADV(Nun) &\to \varepsilon\#1, & VAFIN(werden) &\to \varepsilon\#0,5, \\
VAINF(werden) &\to \varepsilon\#0,25, & VVINF(werden) &\to \varepsilon\#0,25, \\
PPER(sie) &\to \varepsilon\#1, & VVPP(umworben) &\to \varepsilon\#1, \\
\$(.) &\to \varepsilon\#1,
\end{aligned}
$$

$, \ldots\}$

Example PLCFRS

PLCFRS (G, p) and $G = (N, \Sigma, \Xi, P, S)$ where

- $N = \{VROOT, S, VP, ADV, VAFIN, VAINF, VVINF, PPER, VVPP, \$, \ldots\}$,
- $\Sigma = \{Nun, werden, sie, umworben, ., \ldots\}$ and
- $P = \{\ldots,$

$$VP(X_1^{(1)}, X_1^{(2)}) \to ADV(X_1^{(1)})VVP(X_1^{(2)})\#0,5,$$

$$S(X_1^{(1)}X_1^{(2)}X_1^{(3)}) \to VAFIN(X_1^{(1)})PPER(X_1^{(2)})VVPP(X_1^{(3)})\#0,25,$$

$$S(X_1^{(1)}X_1^{(2)}, X_2^{(1)}) \to VP(X_1^{(1)}, X_2^{(1)})VAINF(X_1^{(2)})\#0,25,$$

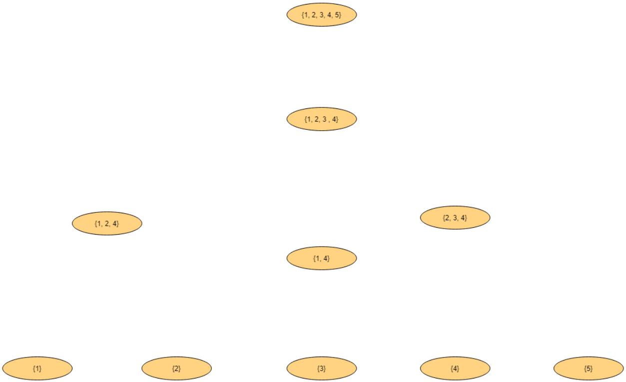
$$S(X_1^{(1)}X_1^{(2)}X_1^{(3)}X_2^{(1)}) \to VP(X_1^{(1)}, X_2^{(1)})VAFIN(X_1^{(2)})PPER(X_1^{(3)})\#0,5,$$

$$S(X_1^{(1)}X_2^{(1)}X_1^{(2)}X_3^{(1)}) \to S(X_1^{(1)}X_2^{(1)}, X_3^{(1)})PPER(X_1^{(2)})\#0,25,$$

$$VROOT(X_1^{(1)}X_2^{(1)}X_3^{(1)}X_4^{(1)}X_1^{(2)}) \to S(X_1^{(1)}X_2^{(1)}X_3^{(1)}X_4^{(1)})\$(X_1^{(2)})\#1$$

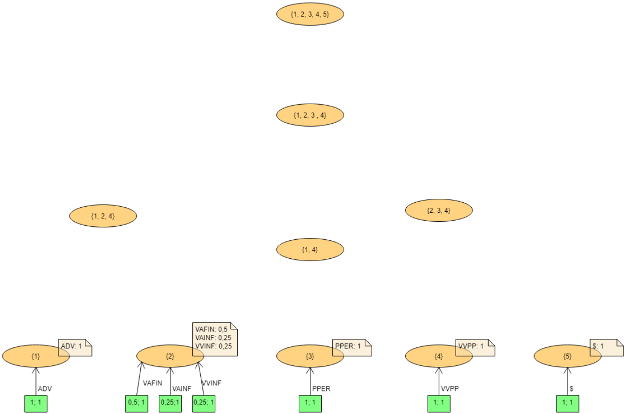
$, \ldots\}$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



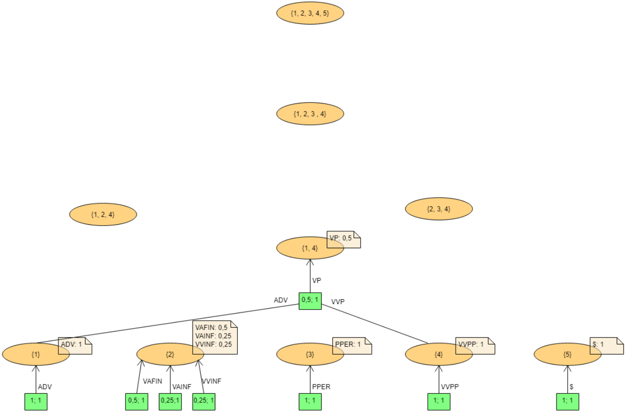
Initialize vertives

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



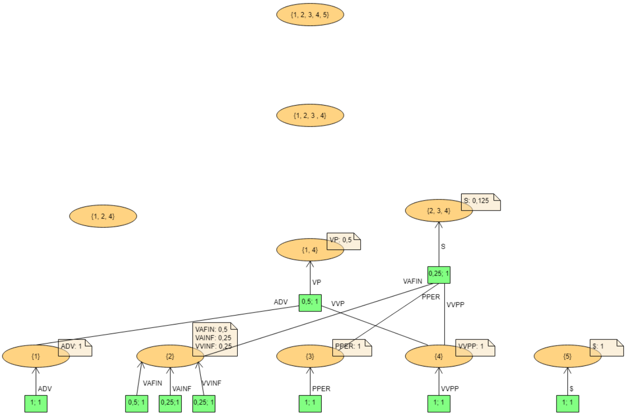
hyperedges for $ADV(Nun) \to \varepsilon\#1, \ldots$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



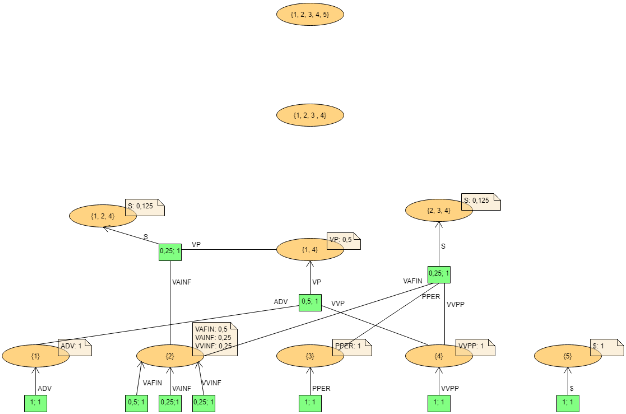
hyperedge for $VP(X_1^{(1)}, X_1^{(2)}) \rightarrow ADV(X_1^{(1)})VVP(X_1^{(2)})\#0,5$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



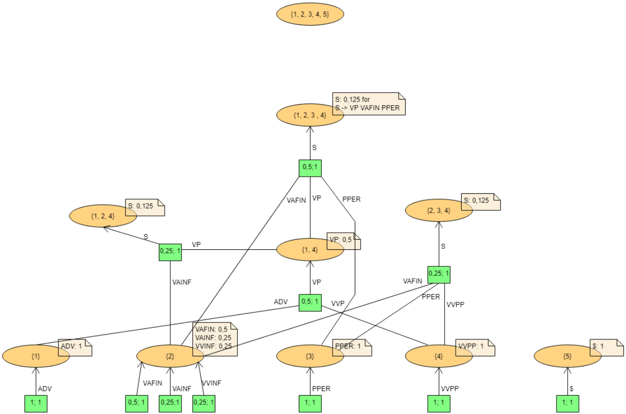
hyperedge for $S(X_1^{(1)}X_1^{(2)}X_1^{(3)}) \rightarrow VAFIN(X_1^{(1)})PPER(X_1^{(2)})VVPP(X_1^{(3)})\#0,25$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



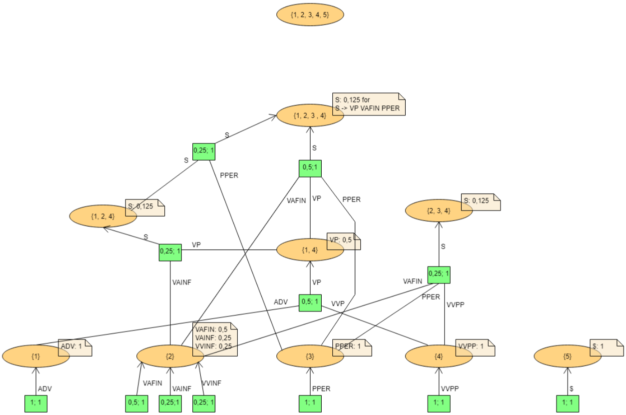
hyperedge for $S(X_1^{(1)} X_1^{(2)}, X_2^{(1)}) \rightarrow VP(X_1^{(1)}, X_2^{(1)}) VAINF(X_1^{(2)}) \#0,25$

Parse - Weighted Deductive Parsing: Nun werden sie umworben .



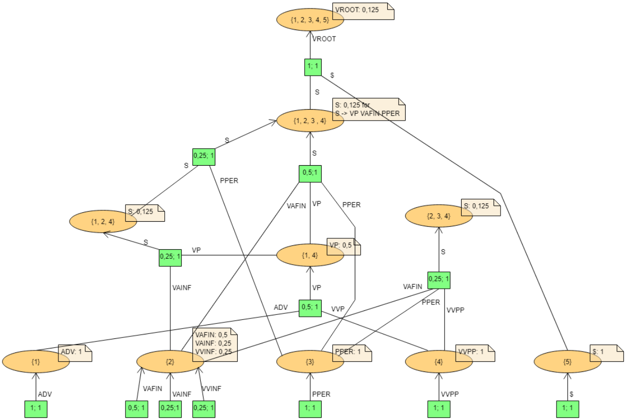
hyperedge for $S(X_1^{(1)}X_1^{(2)}X_1^{(3)}X_2^{(1)}) \rightarrow VP(X_1^{(1)}, X_2^{(1)})\, VAFIN(X_1^{(2)})\, PPER(X_1^{(3)})\#0,5$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



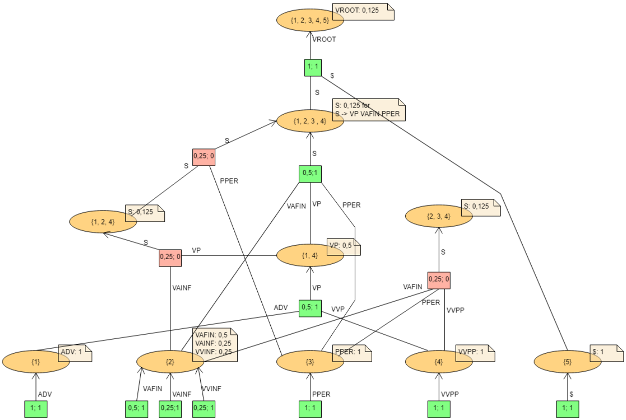
hyperedge for $S(X_1^{(1)} X_2^{(1)} X_1^{(2)} X_3^{(1)}) \rightarrow S(X_1^{(1)} X_2^{(1)}, X_3^{(1)}) PPER(X_1^{(2)}) \#0,25$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



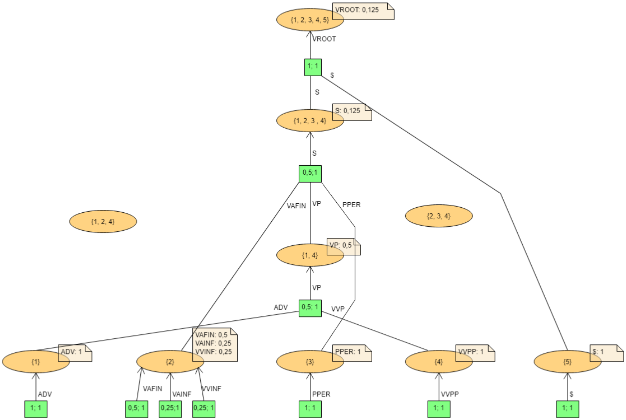
hyperedge for $VROOT(X_1^{(1)} X_2^{(1)} X_3^{(1)} X_4^{(1)} X_1^{(2)}) \rightarrow S(X_1^{(1)} X_2^{(1)} X_3^{(1)} X_4^{(1)})\$(X_1^{(2)})\#1$

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



Undesired hyperedges

PARSE - Weighted Deductive Parsing: Nun werden sie umworben .



Prune

Motivation

- How to reduce the parse time for a sentence?

Motivation

- How to reduce the parse time for a sentence?
- What is a good pruning method?

Motivation

- How to reduce the parse time for a sentence?
- What is a good pruning method?
- How to train such a pruning method?

Overview

- Motivation
- Preliminaries
- LOLS
- Change Propagation
- Dynamic Programming
- Results

Preliminaries

$$
\begin{aligned}
H = (V, E) \in \mathcal{H}_{(G,p)}(w) &: \text{derivation graph from } \textsc{Parse} \\
c \subset \Sigma^* \times T_N(\Sigma) &: X \times Y \text{ -- corpus} \\
s &: \text{state of the derivation graph} \\
a \in \{\textit{keep}, \textit{prune}\} &: \text{action} \\
\tau = s_0 a_0 s_1 a_1 \ldots s_T &: \text{trajectory}
\end{aligned}
$$

Preliminaries

pruning policy π : inputs a hyperedge and a sub sentence w'
outputs a pruning decision $a \in \{keep, prune\}$

How to evaluate π?

Preliminaries

pruning policy π : inputs a hyperedge and a sub sentence w'
outputs a pruning decision $a \in \{keep, prune\}$

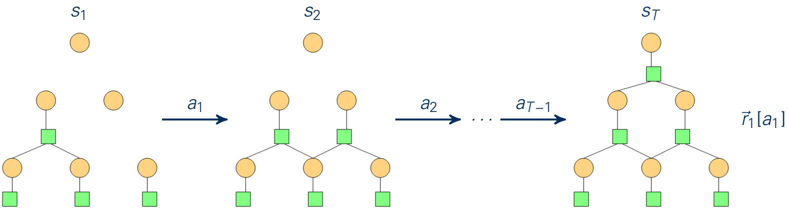
How to evaluate π?

reward function $r : \mathcal{H}_{(G,p)}(w) \times T_N(\Sigma) \to \mathbb{R}$
schematically $r = accuracy - \lambda \cdot runtime$
where $accuracy : T_N(\Sigma) \times T_N(\Sigma) \to \mathbb{R}$
and $runtime : \mathcal{H}_{(G,p)}(w) \to \mathbb{R}$
$\lambda \in \mathbb{R}$: trade-off factor

empirical value of π : $\mathcal{R}(\pi) = \frac{1}{|c|} \sum_{(w,\xi) \in c} r(\textsc{Parse}(G, w, \pi), \xi) \cdot c(w, \xi)$

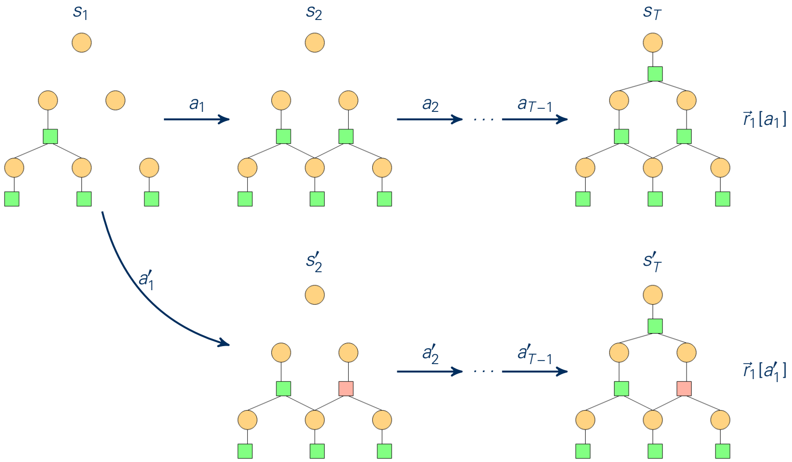
Preliminaries

trajectory: $s_0 a_0 s_1 a_1 \dots s_T$



Preliminaries

trajectory: $s_0 a_0 s_1 a_1 \ldots s_T$, (intervention at state s_1)



LOLS

Locally Optimal Learning to Search

Algorithm 1 Locally Optimal Learning to Search algorithm by [VE17] and [Cha+15]

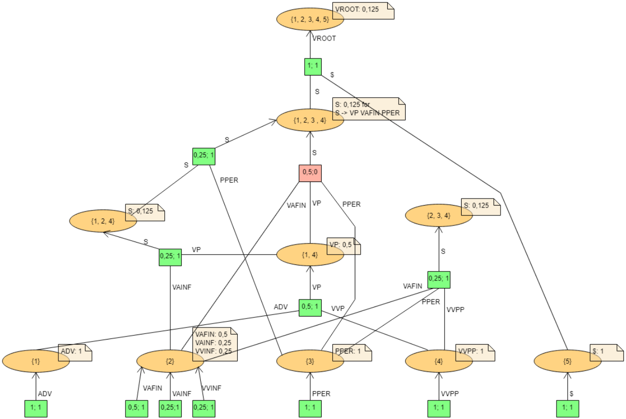
Input: PLCFRS (G, p) with $G = (N, \Sigma, \Xi, P, S)$,
$X \times Y$-corpus c such that $X \subset \Sigma^*$ and $Y \subset T_N(\Sigma)$
Output: pruning policy π

function $\textsc{Lols}((G, p), c)$
  $\pi_1 := \textsc{InitializePolicy}(\ldots)$
  for $i := 1$ to n **do** ▷ n : number of iterations
    $Q_i := \emptyset$ ▷ Q_i : set of state-reward tuples
    for $(w, \xi) \in c$ **do** ▷ w : sentence
      $\tau := \textsc{Roll-In}((G, p), w, \pi_i, \xi)$ ▷ $\tau = s_0 a_0 s_1 a_1 \ldots s_T$: trajectory
      for $t := 0$ to $|\tau| - 1$ **do**
        for $\bar{a}_t \in \{keep, prune\}$ **do** ▷ intervention
          $\vec{r}_t[a'_t] := \textsc{Roll-Out}\ (\pi_i, s_t, a'_t, \xi)$
        end for
        $Q_i := Q_i \cup \{(s_t, \vec{r}_t)\}$
      end for
    end for
    $\pi_{i+1} := \textsc{Train}(\bigcup_{k=1}^{i} Q_k)$ ▷ dataset aggregation
  end for
  return $\text{argmax}_{\pi_j : 1 \leq j \leq n} \mathcal{R}(\pi_j)$
end function

Overview

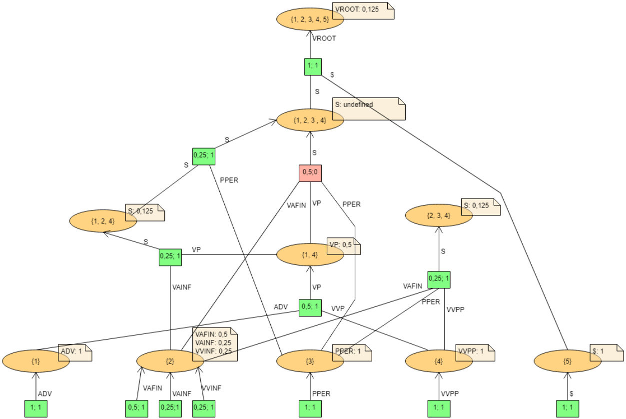
- Motivation
- Preliminaries
- LOLS
- Change Propagation
- Results

Change Propagation



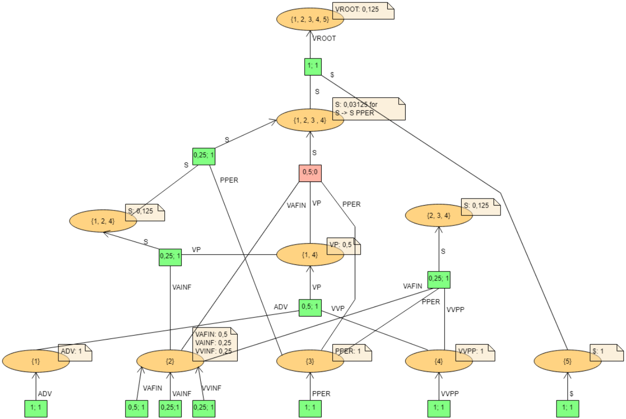
Change pruning bit

Change Propagation



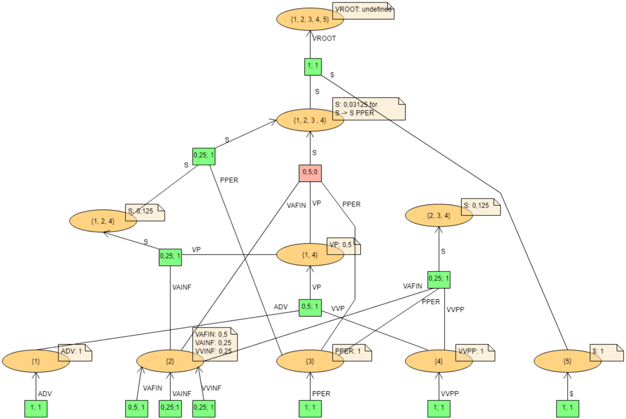
Delete witness for {1, 2, 3, 4} and S

Change Propagation



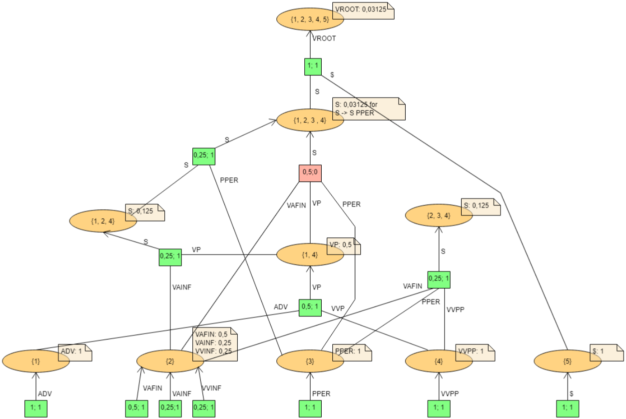
Find new witness for {1, 2, 3, 4} and S

Change Propagation



Repeat for affected vertices

Change Propagation

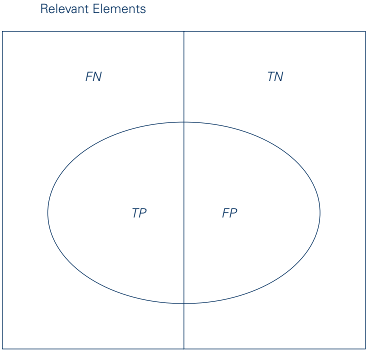


Done

Overview

- Motivation
- Preliminaries
- LOLS
- Change Propagation
- Results

Accuracy Measure

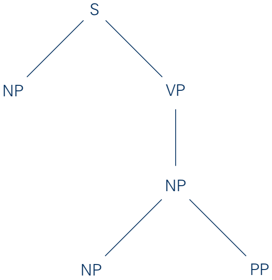


$$\text{precision} = \frac{|TP|}{|TP| + |FP|} \qquad \text{recall} = \frac{|TP|}{|TP| + |FN|}$$

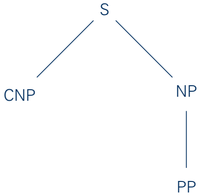
$$\mathfrak{p}(\xi) = \underline{\qquad} \qquad \mathfrak{r}(\xi) = \underline{\qquad}$$

Accuracy Measure

derivation tree by parsing



derivation tree by gold standard

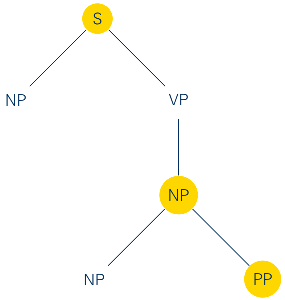


$$\text{precision} = \frac{|TP|}{|TP| + |FP|} \qquad \text{recall} = \frac{|TP|}{|TP| + |FN|}$$

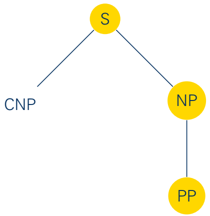
$$\mathfrak{p}(\xi) = \underline{\qquad} \qquad \mathfrak{r}(\xi) = \underline{\qquad}$$

Accuracy Measure

derivation tree by parsing



derivation tree by gold standard



$$\text{precision} = \frac{|TP|}{|TP| + |FP|}$$

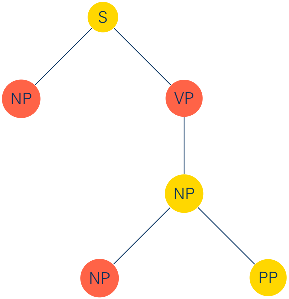
$$\mathfrak{p}(\xi) = \frac{3}{3}$$

$$\text{recall} = \frac{|TP|}{|TP| + |FN|}$$

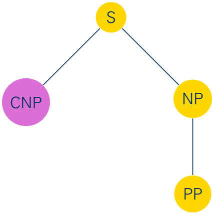
$$\mathfrak{r}(\xi) = \frac{3}{3}$$

Accuracy Measure

derivation tree by parsing



derivation tree by gold standard



$$\text{precision} = \frac{|TP|}{|TP| + |FP|}$$

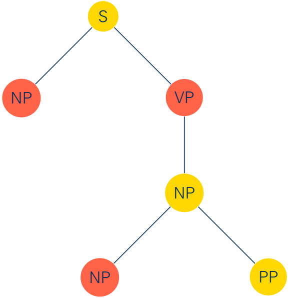
$$\mathfrak{p}(\xi) = \frac{3}{3 + 3}$$

$$\text{recall} = \frac{|TP|}{|TP| + |FN|}$$

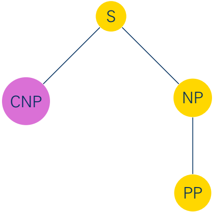
$$\mathfrak{r}(\xi) = \frac{3}{3 + 1}$$

Accuracy Measure

derivation tree by parsing



derivation tree by gold standard



$$\text{precision} = \frac{|TP|}{|TP| + |FP|}$$

$$\mathfrak{p}(\xi) = \frac{3}{3+3} = 0,5$$

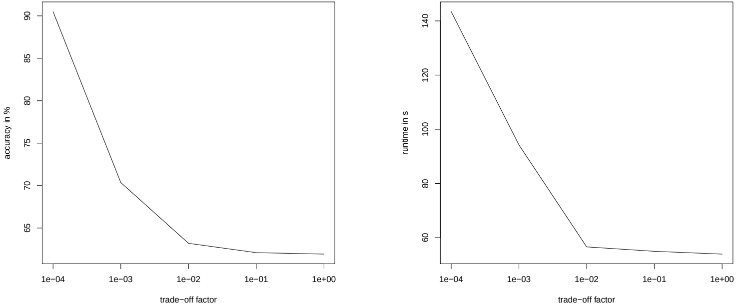
$$\text{recall} = \frac{|TP|}{|TP| + |FN|}$$

$$\mathfrak{r}(\xi) = \frac{3}{3+1} = 0,75$$

Setup

$$\begin{aligned} accuracy(\xi, \zeta) &= 2 \cdot \frac{\mathfrak{p}(\xi, \zeta) \cdot \mathfrak{r}(\xi, \zeta)}{\mathfrak{p}(\xi, \zeta) + \mathfrak{r}(\xi, \zeta)} && \text{F1-Measure,} \\ runtime(H) &= |E| && \text{for } H = (V, E) \\ \lambda &\in [0, 1] \end{aligned}$$

Results



(a) accuracy for λ

(b) runtime for λ

Figure : runtime and accuracy for given *lambda*

References I

Umut A. Acar and Ruy Ley-Wild. "Self-adjusting Computation with Delta ML". In: *Advanced Functional Programming: 6th International School, AFP 2008, Heijen, The Netherlands, May 2008, Revised Lectures*. Ed. by Pieter Koopman, Rinus Plasmeijer, and Doaitse Swierstra. Springer Berlin Heidelberg, 2009, pp. 1–38. ISBN: 978-3-642-04652-0. DOI: 10.1007/978-3-642-04652-0_1.

References II

Kai-Wei Chang et al. "Learning to Search Better than Your Teacher". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. Ed. by David Blei and Francis Bach. JMLR Workshop and Conference Proceedings, 2015, pp. 2058–2066.

Andreas van Cranenburgh, Remko Scha, and Rens Bod. "Data-Oriented Parsing with discontinuous constituents and function tags". In: *Journal of Language Modelling* 4.1 (2016), pp. 57–111. URL: `http://dx.doi.org/10.15398/jlm.v4i1.100`.

References III

Laura Kallmeyer. *Parsing Beyond Context-Free Grammars*. Springer Publishing Company, Incorporated, 2012. ISBN: 3642264530, 9783642264535.

Laura Kallmeyer and Wolfgang Maier. "Data-driven Parsing with Probabilistic Linear Context-free Rewriting Systems". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. COLING '10. Beijing, China: Association for Computational Linguistics, 2010, pp. 537–545. URL: http://dl.acm.org/citation.cfm?id=1873781.1873842.

References IV

Yuki Kato, Hiroyuki Seki, and Tadao Kasami. "Stochastic Multiple Context-free Grammar for RNA Pseudoknot Modeling". In: *Proceedings of the Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms*. TAGRF '06. Sydney, Australia: Association for Computational Linguistics, 2006, pp. 57–64. ISBN: 1-932432-85-X. URL: `http://dl.acm.org/citation.cfm?id=1654690.1654698`.

Mark-Jan Nederhof. "Weighted deductive parsing and Knuth's algorithm". In: *Computational Linguistics* 29.1 (2003), pp. 135–143.

References V

David M. W. Powers. "Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation". In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63. ISSN: 2229-3981 & 2229-399X.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. "No-Regret Reductions for Imitation Learning and Structured Prediction". In: *CoRR* abs/1011.0686 (2010).

Tim Vieira and Jason Eisner. "Learning to Prune: Exploring the Frontier of Fast and Accurate Pasring". In: *Transactions of the Association for Computational Linguistics (TACL)* 5 (Feb. 2017).

References VI

K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. “Characterizing Structural Descriptions Produced by Various Grammatical Formalisms”. In: *Proceedings of the 25th Annual Meeting on Association for Computational Linguistics*. ACL ’87. Stanford, California: Association for Computational Linguistics, 1987, pp. 104–111. DOI: 10.3115/981175.981190. URL: https://doi.org/10.3115/981175.981190.