Dependency Parsing with Bounded Block Degree and Well-nestedness via Lagrangian Relaxation and Branch-and-Bound

Caio Corro, Joseph Le Roux, Mathieu Lacroix, Antoine Rozenknop and Roberto Wolfler-Calvo

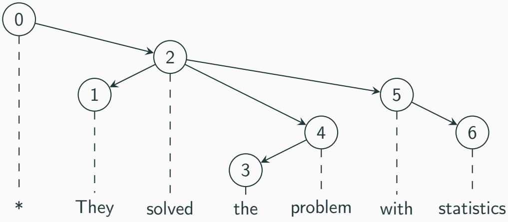
August 7-12

Université Paris 13 – LIPN

This work is supported by a public grant overseen by the French National Research Agency (ANR) as part of the Investissements d'Avenir program (ANR-10-LABX-0083).

Dependency trees

- Association of each word of sentence with a vertex
- Dependency tree: spanning tree rooted at 0

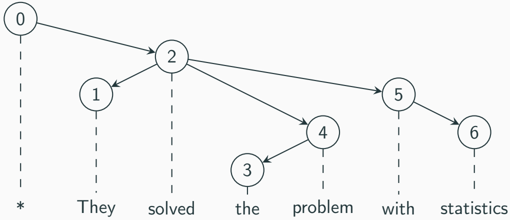


Dependency parsing

- Set of valid dependency trees for sentence x: Y_x
- Arc factored model: $score(y) = \sum_{a \in Y} score(a)$
- Dependency parsing: $\hat{y}_x = \arg\max_{y \in Y_x} score(y)$

Dependency trees

- Association of each word of sentence with a vertex
- Dependency tree: spanning tree rooted at 0

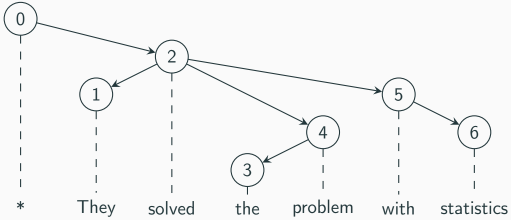


Structural properties **[Bodirsky et al. 2009; Kuhlmann 2010]**

Non-projective ⟷ **Projective**

Dependency trees

- Association of each word of sentence with a vertex
- Dependency tree: spanning tree rooted at 0



Structural properties **[Bodirsky et al. 2009; Kuhlmann 2010]**

Non-projective ⟷ **k-Bounded Block Degree** ⟷ Projective

Non-projective ⟷ **Well-nested** ⟷ Projective

Distribution of dependency tree characteristics

	English (PTB/LTH)		German (SPMRL)		Dutch (UD)	
	WN	~~WN~~	WN	~~WN~~	WN	~~WN~~
BD 1	92.26		67.60		69.13	
BD 2	7.58	0.12	27.12	0.79	28.50	0.08
BD 3	0.12	0.01	3.86	0.30	2.24	0.01
BD 4	0.00	0.00	0.19	$<$0.01	0.04	0.00
BD $>$ 4	0.00	0.00	0.11	$<$0.01	0.00	0.00

	Spanish (UD)		Portuguese (UD)	
	WN	~~WN~~	WN	~~WN~~
BD 1	93.95		81.56	
BD 2	5.99	0.04	13.92	0.05
BD 3	0.02	0.00	3.76	0.02
BD 4	0.00	0.00	0.54	0.00
BD $>$ 4	0.00	0.00	0.14	0.00

Distribution of dependency tree characteristics

	English (PTB/LTH)		German (SPMRL)		Dutch (UD)	
	WN	~~WN~~	WN	~~WN~~	WN	~~WN~~
BD 1	**92.26**		**67.60**		**69.13**	
BD 2	7.58	0.12	27.12	0.79	28.50	0.08
BD 3	0.12	0.01	3.86	0.30	2.24	0.01
BD 4	0.00	0.00	0.19	$<$0.01	0.04	0.00
BD $>$ 4	0.00	0.00	0.11	$<$0.01	0.00	0.00

	Spanish (UD)		Portuguese (UD)	
	WN	~~WN~~	WN	~~WN~~
BD 1	**93.95**		**81.56**	
BD 2	5.99	0.04	13.92	0.05
BD 3	0.02	0.00	3.76	0.02
BD 4	0.00	0.00	0.54	0.00
BD $>$ 4	0.00	0.00	0.14	0.00

- **Blue**: Projective dependency trees

Distribution of dependency tree characteristics

	English (PTB/LTH)		German (SPMRL)		Dutch (UD)	
	WN	~~WN~~	WN	~~WN~~	WN	~~WN~~
BD 1	**92.26**		**67.60**		**69.13**	
BD 2	**7.58**	0.12	**27.12**	**0.79**	**28.50**	0.08
BD 3	0.12	0.01	**3.86**	**0.30**	**2.24**	0.01
BD 4	0.00	0.00	0.19	$<$0.01	0.04	0.00
BD $>$ 4	0.00	0.00	0.11	$<$0.01	0.00	0.00

	Spanish (UD)		Portuguese (UD)	
	WN	~~WN~~	WN	~~WN~~
BD 1	**93.95**		**81.56**	
BD 2	**5.99**	0.04	**13.92**	0.05
BD 3	0.02	0.00	**3.76**	0.02
BD 4	0.00	0.00	0.54	0.00
BD $>$ 4	0.00	0.00	0.14	0.00

- **Blue**: Projective dependency trees
- **Blue** + **Purple**: $\approx$ 99% of the dependency trees

Distribution of dependency tree characteristics

	English (PTB/LTH)		German (SPMRL)		Dutch (UD)	
	WN	~~WN~~	WN	~~WN~~	WN	~~WN~~
BD 1	**92.26**		**67.60**		**69.13**	
BD 2	**7.58**	**0.12**	**27.12**	**0.79**	**28.50**	**0.08**
BD 3	**0.12**	**0.01**	**3.86**	**0.30**	**2.24**	**0.01**
BD 4	**0.00**	**0.00**	**0.19**	**$<$0.01**	**0.04**	**0.00**
BD $>$ 4	**0.00**	**0.00**	**0.11**	**$<$0.01**	**0.00**	**0.00**

	Spanish (UD)		Portuguese (UD)	
	WN	~~WN~~	WN	~~WN~~
BD 1	**93.95**		**81.56**	
BD 2	**5.99**	**0.04**	**13.92**	**0.05**
BD 3	**0.02**	**0.00**	**3.76**	**0.02**
BD 4	**0.00**	**0.00**	**0.54**	**0.00**
BD $>$ 4	**0.00**	**0.00**	**0.14**	**0.00**

- **Blue**: Projective dependency trees
- **Blue** + **Purple**: $\approx$ 99% of the dependency trees
- **Blue** + **Purple** + **Red**: Non-projective dependency trees

Motivations

Observation

- Projective parsing: does not correctly cover datasets
- Non-projective parsing: produce invalid structures

Problem

- WN and k-BBD parsing: no tractable algorithm

Contribution

- First efficient parsing algorithm based on Lagrangian Relaxation

Outline

1.Introduction

2. Dependency tree characterization

3. Existing parsing algorithms

4. Novel characterization based on arc-sets

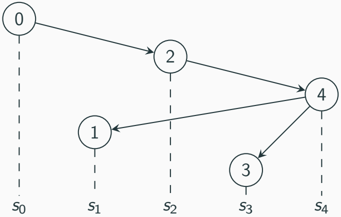
5. Efficient parsing with fine-grained constraints

6. Experiments

7. Conclusion

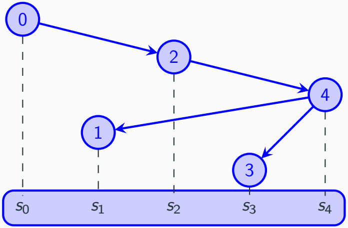
Yield

Yield of a node v: set of all nodes reachable from v



Yield

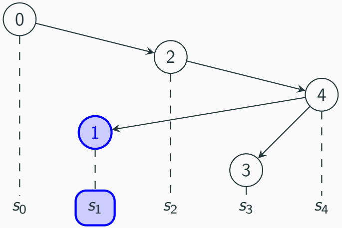
Yield of a node v: set of all nodes reachable from v



$Yield(0) = \{0, 1, 2, 3, 4\}$

Yield

Yield of a node v: set of all nodes reachable from v

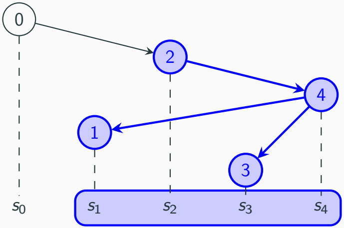


$Yield(0) = \{0, 1, 2, 3, 4\}$

$Yield(1) = \{1\}$

Yield

Yield of a node v: set of all nodes reachable from v



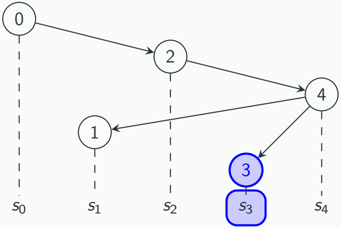
$Yield(0) = \{0, 1, 2, 3, 4\}$

$Yield(1) = \{1\}$

$Yield(2) = \{1, 2, 3, 4\}$

Yield

Yield of a node v: set of all nodes reachable from v



$Yield(0) = \{0, 1, 2, 3, 4\}$

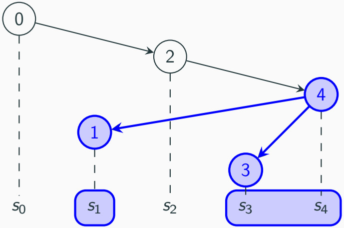
$Yield(1) = \{1\}$

$Yield(2) = \{1, 2, 3, 4\}$

$Yield(3) = \{3\}$

Yield

Yield of a node v: set of all nodes reachable from v



$Yield(0) = \{0, 1, 2, 3, 4\}$

$Yield(1) = \{1\}$

$Yield(2) = \{1, 2, 3, 4\}$

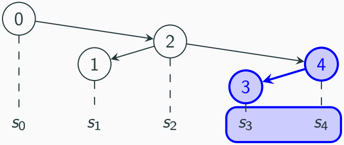
$Yield(3) = \{3\}$

$Yield(4) = \{3, 4\}$

Structural properties of dependencies

Projective dependency trees

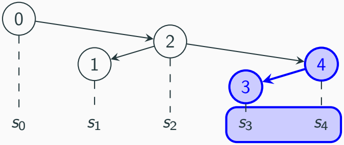
⇒ Trees with contiguous yields only



Structural properties of dependencies

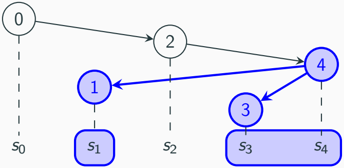
Projective dependency trees

⇒ Trees with contiguous yields only



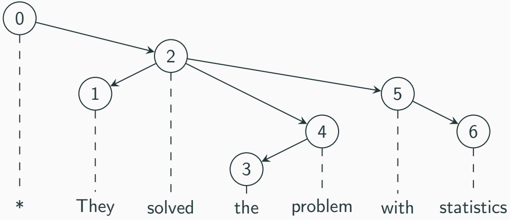
Non-projective dependency trees

⇒ Unconstrained trees

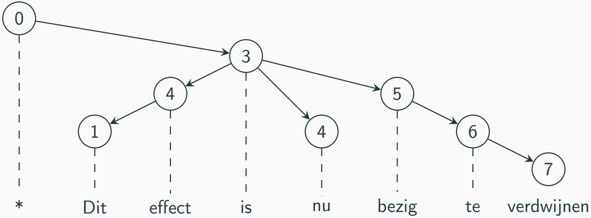


Example: Projective dependency trees

- English

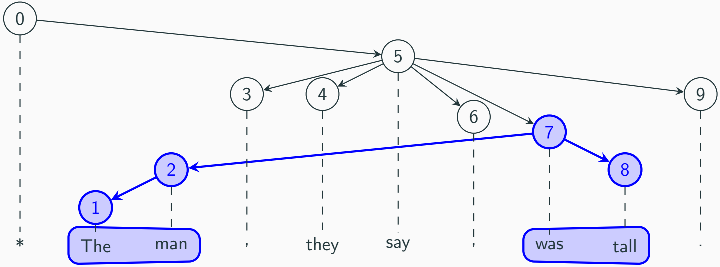


- Dutch

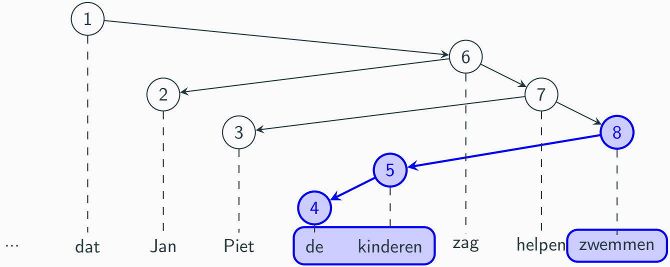


Example: Non-projective dependency trees

- English: surrounding argument



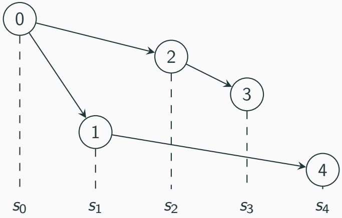
- Dutch: cross-serial dependencies



Structural properties (1/2): k-BBD

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k

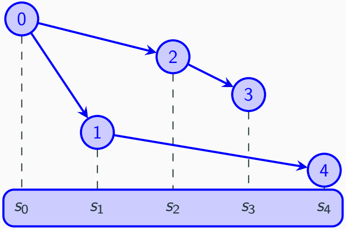


Tree of block degree 2

Structural properties (1/2): k-BBD

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k

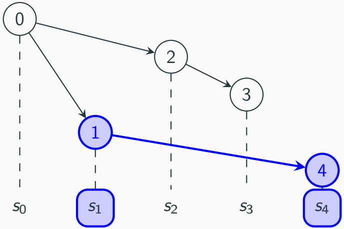


$Yield(0) = [0 \ldots 4]$ $BD(0) = 1$

Tree of block degree 2

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k



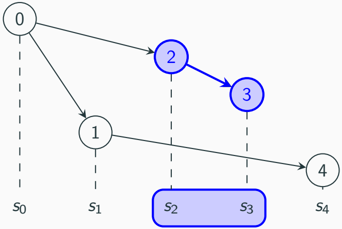
$Yield(0) = [0 \ldots 4]$ $\quad BD(0) = 1$

$Yield(1) = [1] \cup [4]$ $\quad BD(1) = 2$

Tree of block degree 2

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k



$$Yield(0) = [0 \dots 4] \qquad BD(0) = 1$$

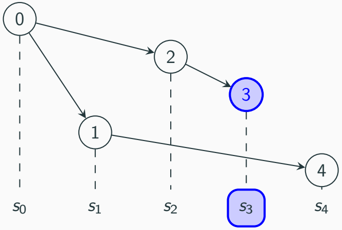
$$Yield(1) = [1] \cup [4] \qquad BD(1) = 2$$

$$Yield(2) = [2 \dots 3] \qquad BD(2) = 1$$

Tree of block degree 2

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k



$Yield(0) = [0 \ldots 4]$ $\quad BD(0) = 1$

$Yield(1) = [1] \cup [4]$ $\quad BD(1) = 2$

$Yield(2) = [2 \ldots 3]$ $\quad BD(2) = 1$

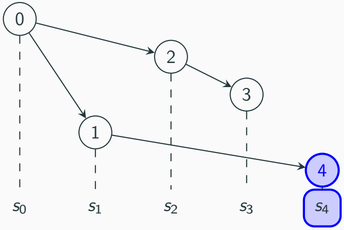
$Yield(3) = [3]$ $\quad BD(3) = 1$

Tree of block degree 2

Structural properties (1/2): k-BBD

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k

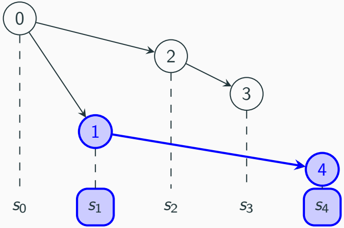


$Yield(0) = [0 \dots 4]$	$BD(0) = 1$
$Yield(1) = [1] \cup [4]$	$BD(1) = 2$
$Yield(2) = [2 \dots 3]$	$BD(2) = 1$
$Yield(3) = [3]$	$BD(3) = 1$
$Yield(4) = [4]$	$BD(4) = 1$

Tree of block degree 2

k-Bounded Block Degree (k-BBD)

- BD of a vertex: number of contiguous intervals described by its yield
- BD of a tree: the maximal block degree of its vertices
- k-BBD tree: tree with a BD less or equal to k

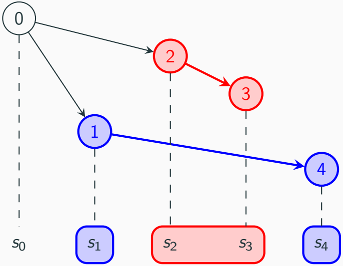


$Yield(0) = [0 \dots 4]$	$BD(0) = 1$
$Yield(1) = [1] \cup [4]$	$BD(1) = 2$
$Yield(2) = [2 \dots 3]$	$BD(2) = 1$
$Yield(3) = [3]$	$BD(3) = 1$
$Yield(4) = [4]$	$BD(4) = 1$

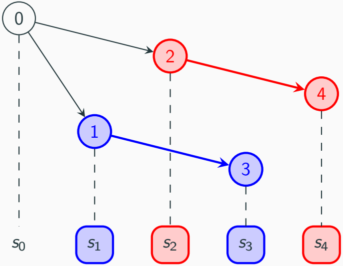
Tree of block degree 2

Well-nestedness (WN)

- Interleaving sets I_1, I_2: there exist $i, j \in I_1$ and $k, l \in I_2$ such that $i < k < j < l$
- *Well-nested* tree: does not contain two vertices whose yields are disjoint and interleave



Well-nested tree



Not well-nested tree

Complexity (arc-factored)

Non-projective	$O(n^2)$	[McDonald et al. 2005]
Projective	$O(n^3)$	[Eisner 2000]
WN + 2-BBD	$O(n^7)$	[Gómez-Rodríguez et al. 2009]
WN + k-BBD, $k \geq 2$	$O(n^{5+2(k-1)})$	[Gómez-Rodríguez et al. 2009]

Remark

Projective $\Leftrightarrow$ 1-BBD and WN

Tractability

- Non-projective and projective: tractable
- WN + k-BBD: not tractable

Non-projective dependency parsing

Integer Linear Program for non-projective parsing

$z \in R^A$: incidence vector such that $z_a = 1$ iff arc a is in the tree.

Integer Linear Program for non-projective parsing

$z \in R^A$: incidence vector such that $z_a = 1$ iff arc a is in the tree.

$$
\begin{aligned}
\max_z \sum_{a \in A} score(a) \times z_a & & & \textit{Arc-factored model} \quad (1)\\
\text{s.t.} \sum_{a \in \delta^{\text{in}}(v)} z_a = 1 & \quad \forall v \in V^+ & & \textit{One head/word} \quad (2)\\
\sum_{a \in \delta^{\text{in}}(W)} z_a \geq 1 & \quad \forall W \subseteq V^+ & & \textit{Connectedness} \quad (3)\\
z \in \{0,1\}^A & & & \textit{Integrality} \quad (4)
\end{aligned}
$$

Non-projective dependency parsing

Integer Linear Program for non-projective parsing

$z \in R^A$: incidence vector such that $z_a = 1$ iff arc a is in the tree.

$$
\begin{aligned}
&\max_z \sum_{a \in A} score(a) \times z_a && && \textit{Arc-factored model} && (1)\\
&\text{s.t.} \sum_{a \in \delta^{\mathrm{in}}(v)} z_a = 1 && \forall v \in V^+ && \textit{One head/word} && (2)\\
&\sum_{a \in \delta^{\mathrm{in}}(W)} z_a \geq 1 && \forall W \subseteq V^+ && \textit{Connectedness} && (3)\\
&z \in \{0, 1\}^A && && \textit{Integrality} && (4)
\end{aligned}
$$

Efficient decoding

In practice: (directed) *Maximum Spanning Tree* (MST) algorithm
[Schrijver 2003; McDonald et al. 2005]

Non-projective dependency parsing

Integer Linear Program for non-projective parsing

$z \in R^A$: incidence vector such that $z_a = 1$ iff arc a is in the tree.

$$
\begin{aligned}
&\max_z \sum_{a \in A} score(a) \times z_a && && \textit{Arc-factored model} && (1)\\
\text{s.t.} &\sum_{a \in \delta^{\mathrm{in}}(v)} z_a = 1 && \forall v \in V^+ && \textit{One head/word} && (2)\\
&\sum_{a \in \delta^{\mathrm{in}}(W)} z_a \geq 1 && \forall W \subseteq V^+ && \textit{Connectedness} && (3)\\
&z \in \{0,1\}^A && && \textit{Integrality} && (4)
\end{aligned}
$$

Problem enhancement

$\Rightarrow$ Integrating fine-grained structural constraints ?

k-Bounded Block Degree Constraint

Definition

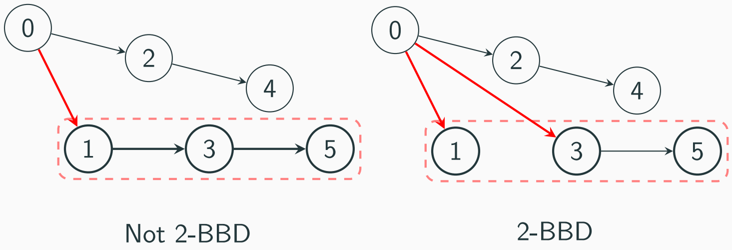
$\mathcal{W}^{k+1}$: vertex subsets describing at least $k+1$ non-adjacent intervals

k-Bounded Block Degree Constraint

Definition

$\mathcal{W}^{k+1}$: vertex subsets describing at least $k+1$ non-adjacent intervals

Example with $k = 2$ **and** $\{1,3,5\} \in \mathcal{W}^3$



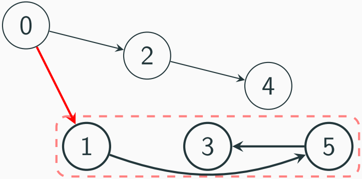
$\color{red}{\rightarrow}$: arcs adjacent to the vertex subset $\{1,3,5\}$

k-Bounded Block Degree Constraint

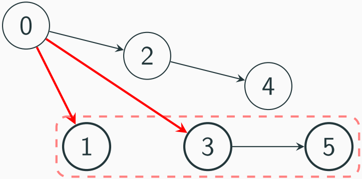
Definition

$\mathcal{W}^{k+1}$: vertex subsets describing at least $k+1$ non-adjacent intervals

Example with $k = 2$ **and** $\{1, 3, 5\} \in \mathcal{W}^3$



Not 2-BBD



2-BBD

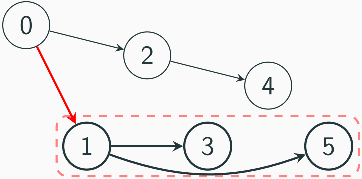
$\color{red}{\rightarrow}$: arcs adjacent to the vertex subset $\{1, 3, 5\}$

k-Bounded Block Degree Constraint

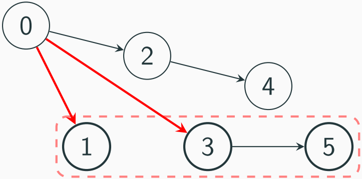
Definition

$\mathcal{W}^{k+1}$: vertex subsets describing at least $k+1$ non-adjacent intervals

Example with $k = 2$ and $\{1, 3, 5\} \in \mathcal{W}^3$



Not 2-BBD



2-BBD

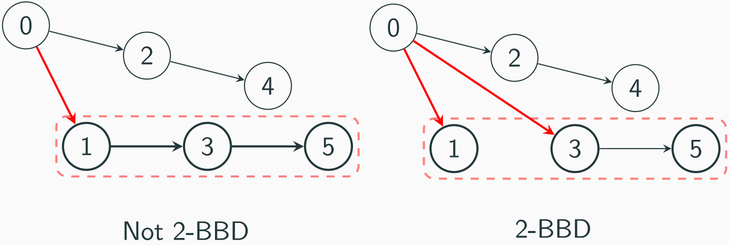
$\rightarrow$: arcs adjacent to the vertex subset $\{1, 3, 5\}$

k-Bounded Block Degree Constraint

Definition

$\mathcal{W}^{k+1}$: vertex subsets describing at least $k+1$ non-adjacent intervals

Example with $k = 2$ **and** $\{1, 3, 5\} \in \mathcal{W}^3$



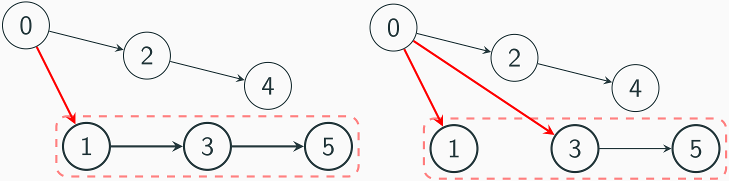
$\color{red}{\rightarrow}$: arcs adjacent to the vertex subset $\{1, 3, 5\}$

k-Bounded Block Degree Constraint

Definition

$\mathcal{W}^{k+1}$: vertex subsets describing at least $k+1$ non-adjacent intervals

Example with $k = 2$ **and** $\{1, 3, 5\} \in \mathcal{W}^3$



Not 2-BBD   2-BBD

$\rightarrow$: arcs adjacent to the vertex subset $\{1, 3, 5\}$

Constraint

For each vertex subset $W \in \mathcal{W}^{\geq k+1} \Rightarrow$ At least two adjacent arcs

Well-nestedness constraint

Definition

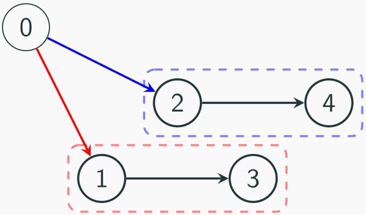
$\mathcal{I}$: family of couples of disjoint interleaving vertex subsets

Well-nestedness constraint

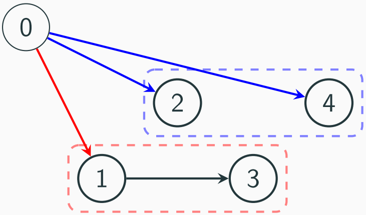
Definition

$\mathcal{I}$: family of couples of disjoint interleaving vertex subsets

Example with $(\{1,3\},\{2,4\}) \in \mathcal{I}$



Not Well-nested



Well-nested

$Yield(1) = \{1, 3\}$
$Yield(2) = \{2, 4\}$
$1 < 2 < 3 < 4$

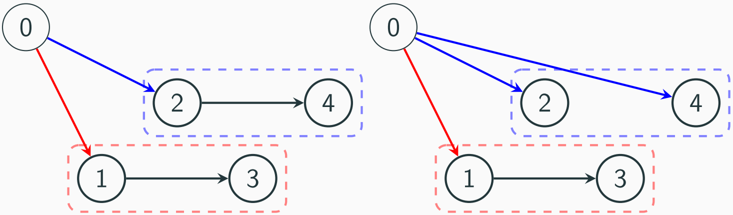
$Yield(1) = \{1, 3\}$
$Yield(2) = \{2\}$
$Yield(4) = \{4\}$

Well-nestedness constraint

Definition

$\mathcal{I}$: family of couples of disjoint interleaving vertex subsets

Example with $(\{1,3\},\{2,4\}) \in \mathcal{I}$



Not Well-nested

Well-nested

Constraint

For each couple $(I_1, I_2) \in \mathcal{I} \Rightarrow$ At least two adjacent arcs for I_1 or I_2

Full ILP: parsing with k-BBD and WN constraints

$$\max_z \sum_{a \in A} score(a) \times z_a \qquad \textit{Arc-factored} \qquad (5)$$

$$\text{s.t.} z \in Z \qquad \textit{Non-projective} \qquad (6)$$

$$\sum_{a \in \delta(W)} z_a \geq 2 \qquad \forall\ W \in \mathcal{W}^{\geq k+1} \qquad \textit{k-BBD} \qquad (7)$$

$$\sum_{a \in \delta(I_1)} z_a + \sum_{a \in \delta(I_2)} z_a \geq 3 \qquad \forall (I_1, I_2) \in \mathcal{I} \qquad \textit{WN} \qquad (8)$$

Problem

- MST: k-BBD and WN constraints can not be integrated
- Generic solver: exponential number of constraints
- Polynomial algorithm: intractable [Gómez-Rodríguez et al. 2009]

Solving the ILP

$\Rightarrow$ Lagrangian Relaxation applied on constraints (7)-(8)

Lagrangian Relaxation

Lagrangian Dual Problem

$$\min_{u \geq 0} \max_{z \in Z} L(z, u)$$

Efficient minimization of the dual

- Min: Subgradient descent
- Max: Maximum Spanning Tree
- Many relaxed constraints: Non Delayed Relax-and-Cut

Efficient maximization of the primal

- Branch-and-Bound
- Problem reduction (exact pruning technique)

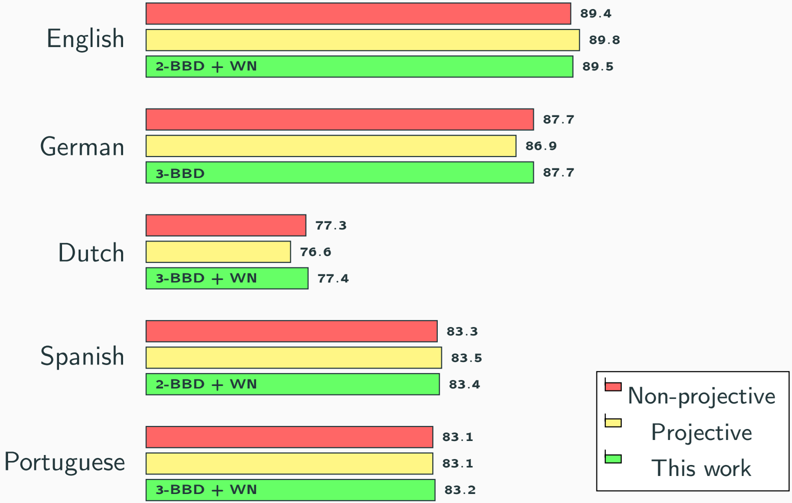
Distribution of dependency tree characteristics

	English (PTB/LTH)		German (SPMRL)		Dutch (UD)	
	WN	~~WN~~	WN	~~WN~~	WN	~~WN~~
BD 1	**92.26**		**67.60**		**69.13**	
BD 2	**7.58**	0.12	**27.12**	**0.79**	**28.50**	0.08
BD 3	0.12	0.01	**3.86**	**0.30**	**2.24**	0.01
BD 4	0.00	0.00	0.19	$<$0.01	0.04	0.00
BD $>$ 4	0.00	0.00	0.11	$<$0.01	0.00	0.00

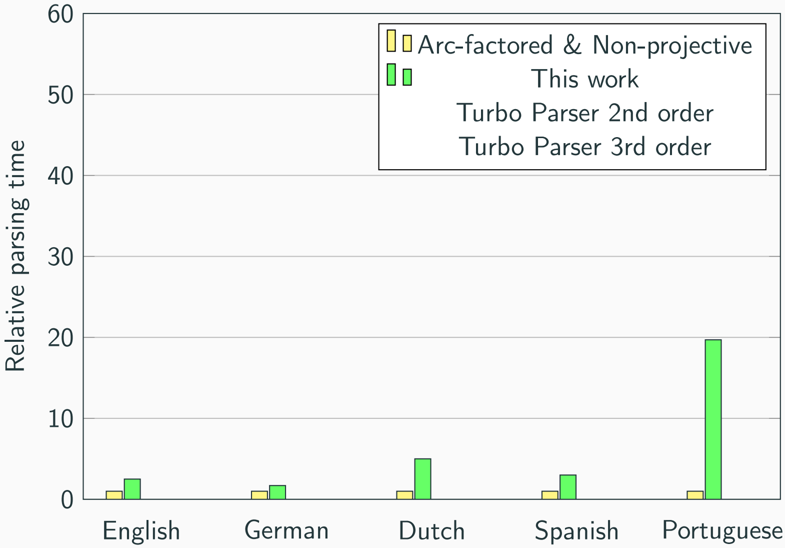
	Spanish (UD)		Portuguese (UD)	
	WN	~~WN~~	WN	~~WN~~
BD 1	**93.95**		**81.56**	
BD 2	**5.99**	0.04	**13.92**	0.05
BD 3	0.02	0.00	**3.76**	0.02
BD 4	0.00	0.00	0.54	0.00
BD $>$ 4	0.00	0.00	0.14	0.00

- **Blue**: Projective dependency trees
- **Blue** + **Purple**: $\approx$ 99% of the dependency trees

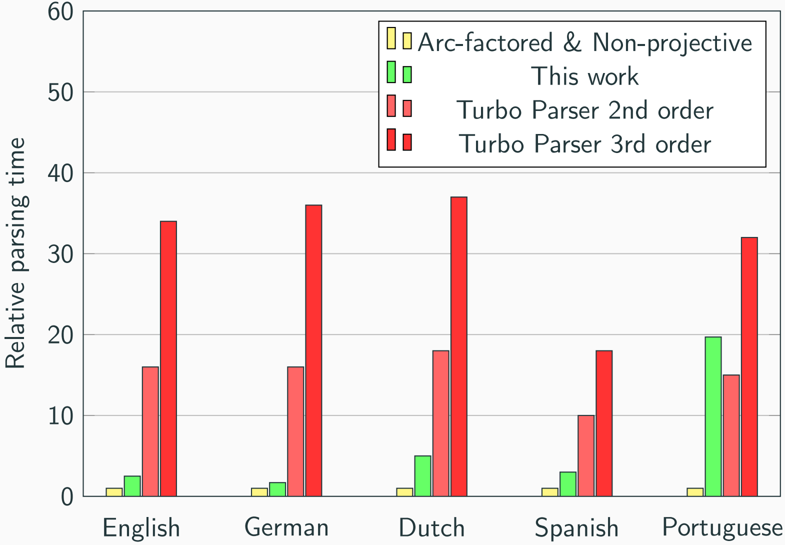
UAS (Ratio of correct arcs)



Efficiency: Relative parsing time



Efficiency: Relative parsing time



Conclusion: k-BBD and WN dependency parsing

Our contribution

- Novel characterization based on arc sets only
- The first efficient and flexible algorithm:
  - k-BBD with arbitrary k
  - WN optional

  Tunable for different languages/properties
- First experimental results with K-BBD and WN parsing

Surprising observation

- Does not improve UAS under an arc-factored model

Perspectives

- LTAG derivation parsing (2-BBD and WN)
- Parsing lexicalized mildly context sensitive languages

References I

Bodirsky, Manuel, Marco Kuhlmann, and Mathias Möhl (2009). "Well-nested drawings as models of syntactic structure". In: *Tenth Conference on Formal Grammar and Ninth Meeting on Mathematics of Language*, pp. 195–203.

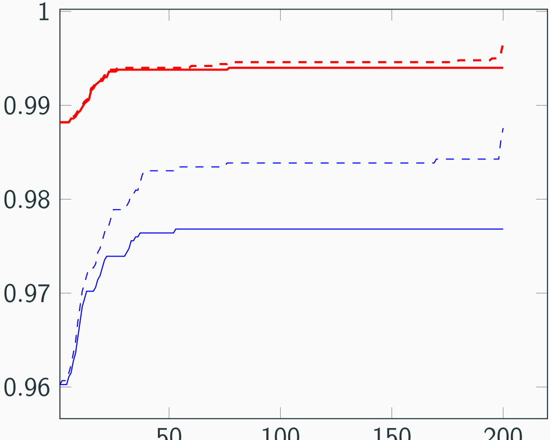
Eisner, Jason (2000). "Bilexical grammars and their cubic-time parsing algorithms". In: *Advances in probabilistic and other parsing technologies*. Springer, pp. 29–61.

Gómez-Rodríguez, Carlos, David Weir, and John Carroll (2009). "Parsing mildly non-projective dependency structures". In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 291–299.

References II

- Koo, Terry et al. (2010). "Dual decomposition for parsing with non-projective head automata". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1288–1298.
- Kuhlmann, Marco (2010). *Dependency Structures and Lexicalized Grammars: An Algebraic Approach*. Vol. 6270. Springer.
- Lemaréchal, Claude (2001). "Lagrangian relaxation". In: *Computational combinatorial optimization*. Springer, pp. 112–156.
- McDonald, Ryan et al. (2005). "Non-projective dependency parsing using spanning tree algorithms". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 523–530.
- Schrijver, A. (2003). *Combinatorial Optimization - Polyhedra and Efficiency*. Springer.

Lagrangian Relaxation: Optimality Rate



(y-axis) Optimality rate

(blue) English

(solid) With certificate

(x-avis) Number of iterations

(red) German

(dashed) Without