



# Bare-Bones Dependency Parsing

A Case for Occam's Razor?

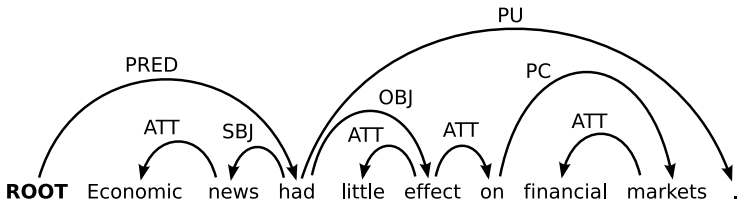
Joakim Nivre

Uppsala University  
Department of Linguistics and Philology  
[joakim.nivre@lingfil.uu.se](mailto:joakim.nivre@lingfil.uu.se)



# Introduction

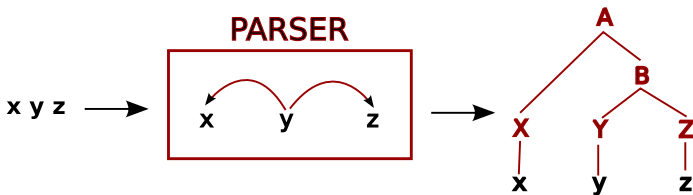
- ▶ Syntactic parsing of natural language:
  - ▶ Who does what to whom?
- ▶ Dependency-based syntactic representations
  - ▶ Binary, asymmetric relations between words
  - ▶ Long tradition in descriptive linguistics
  - ▶ Increasingly popular in computational linguistics





## Varieties of Dependency Parsing

- ▶ Dependencies as internal representations (for parsers)
  - ▶ Dependency relations useful for disambiguation
  - ▶ Incorporated into head-lexicalized grammars

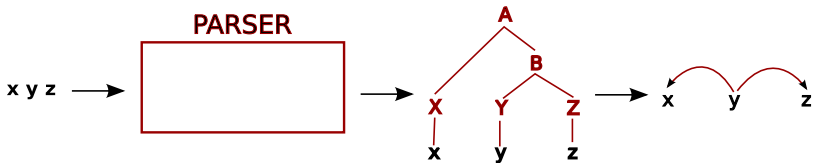


Example: The Collins Parser [Collins 1997]



## Varieties of Dependency Parsing

- ▶ Dependencies as final representations (for applications)
  - ▶ Information extraction [Culotta and Sorensen 2004]
  - ▶ Question answering [Bouma et al. 2005]
  - ▶ Machine translation [Ding and Palmer 2004]

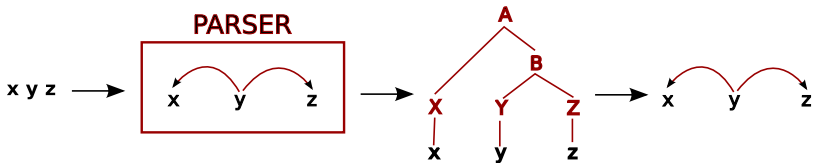


**Example:** The Stanford Parser [Klein and Manning 2003]



## Varieties of Dependency Parsing

- ▶ Dependencies as final representations (for applications)
  - ▶ Information extraction [Culotta and Sorensen 2004]
  - ▶ Question answering [Bouma et al. 2005]
  - ▶ Machine translation [Ding and Palmer 2004]

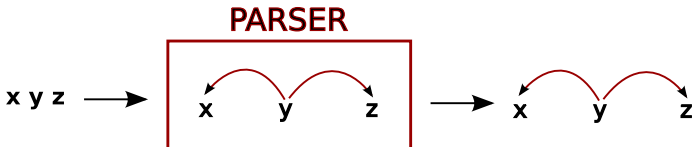


**Example:** The Stanford Parser [Klein and Manning 2003]



## Varieties of Dependency Parsing

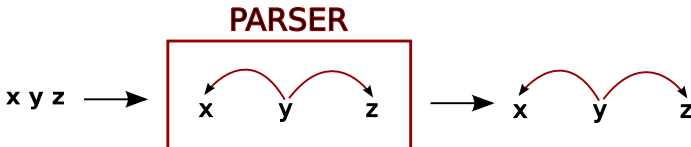
- ▶ Dependencies as the **one and only** representation
  - ▶ If we only want a dependency tree, why do more?
  - ▶ Bare-bones dependency parsing [Eisner 1996]





## Varieties of Dependency Parsing

- ▶ Dependencies as the **one and only** representation
  - ▶ If we only want a dependency tree, why do more?
  - ▶ Bare-bones dependency parsing [Eisner 1996]



Occam's razor: pluralitas non est ponenda sine necessitate





# Outline

- ▶ Basic concepts of dependency parsing
  - ▶ Representations, metrics, benchmarks
- ▶ Parsing methods for bare-bones dependency parsing
  - ▶ Chart parsing techniques
  - ▶ Parsing as constraint satisfaction
  - ▶ Transition-based parsing
  - ▶ Hybrid methods
- ▶ Comparative evaluation
  - ▶ Different types of parsers evaluated on dependency output
  - ▶ Can we really appeal to Occam's razor?





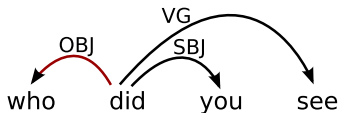
## Dependency Graphs

- ▶ A **dependency graph** for a sentence  $S = w_1, \dots, w_n$  is a directed graph  $G = (V, A)$ , where:
  - ▶  $V = \{1, \dots, n\}$  is the set of **nodes**, representing tokens,
  - ▶  $A \subseteq V \times V$  is the set of **arcs**, representing dependencies.
- ▶ Note:
  - ▶ Arc  $i \rightarrow j$  is a dependency with head  $w_i$  and dependent  $w_j$
  - ▶ Arc  $i \rightarrow j$  may be labeled with a dependency type  $r \in R$



## Constraints on Dependency Graphs

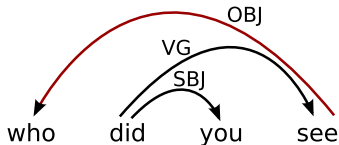
- ▶  $G$  must be a projective **tree**
  - ▶ All subtrees have a contiguous yield
  - ▶ Simple conversion from/to phrase structure trees
  - ▶ Hard to represent long-distance dependencies





## Constraints on Dependency Graphs

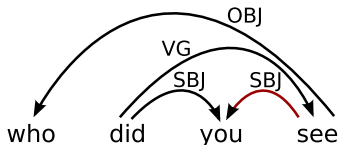
- ▶  $G$  must be a **tree**
  - ▶ Subtrees may have a discontinuous yield
  - ▶ Allows non-projective arcs for long-distance dependencies
  - ▶ Prague Dependency Treebank [Hajič et al. 2001] (25% trees)





## Constraints on Dependency Graphs

- ▶  $G$  must be connected and acyclic (**DAG**)
  - ▶ A node may have more than one incoming arc
  - ▶ Allows multiple heads for deep syntactic relations
  - ▶ Danish Dependency Trebank [Kromann 2003]





## Parsing Problem

- ▶ **Input:**  $S = w_1, \dots, w_n$
- ▶ **Output:**  $G^* = \operatorname{argmax}_{G \in \mathcal{G}(S)} \mathcal{F}(S, G)$
- ▶ **Note:**
  - ▶  $\mathcal{F}(S, G)$  is the score of  $G$  for  $S$
  - ▶  $\mathcal{G}(S)$  is the space of possible dependency graphs for  $S$
  - ▶ Nodes given by input, only arcs need to be found
  - ▶ With tree constraint, assignment of head  $h_i$  and relation  $r_i$



# Parsing Problem

- ▶ **Input:**  $S = w_1, \dots, w_n$
- ▶ **Output:**  $G^* = \operatorname{argmax}_{G \in \mathcal{G}(S)} \mathcal{F}(S, G)$
- ▶ **Note:**
  - ▶  $\mathcal{F}(S, G)$  is the score of  $G$  for  $S$
  - ▶  $\mathcal{G}(S)$  is the space of possible dependency graphs for  $S$
  - ▶ Nodes given by input, only arcs need to be found
  - ▶ With tree constraint, assignment of head  $h_i$  and relation  $r_i$

	Relation $r_i \in R$	OBJ	ROOT	SBJ	VG
<b>Output</b>	Head $h_i \in V \cup \{0\}$	4	0	2	2
<b>Input</b>	Node $i \in V$	1	2	3	4
	Word $w_i \in S$	who	did	you	see
	PoS tag	WP	VBD	PRP	VB



## Evaluation Metrics

- ▶ Accuracy on individual arcs:

$$\text{Recall (R)} = \frac{|\text{PARSED} \cap \text{GOLD}|}{|\text{GOLD}|}$$

$$\text{Precision (P)} = \frac{|\text{PARSED} \cap \text{GOLD}|}{|\text{PARSED}|}$$

$$\text{Attachment score (AS)} = P = R \text{ (only for trees)}$$

- ▶ All metrics can be labeled (L) or unlabeled (U)



## Benchmark Data Sets

- ▶ Penn Treebank (**PTB**) [Marcus et al. 1993]:
  - ▶ Phrase structure annotation converted to dependencies
  - ▶ **Penn2Malt** – projective trees [Nivre 2006]
  - ▶ **Stanford** – projective trees or graphs [de Marneffe et al. 2006]
- ▶ Prague Dependency Treebank (**PDT**) [Hajič et al. 2001]:
  - ▶ Native dependency annotation – non-projective trees
- ▶ CoNLL Shared Tasks [Buchholz and Marsi 2006, Nivre et al. 2007]:
  - ▶ **CoNLL-06**: 13 languages (trees, mostly non-projective)
  - ▶ **CoNLL-07**: 10 languages (trees, mostly non-projective)





# Parsing Methods

- ▶ Parsing methods for bare-bones dependency parsing
  - ▶ Chart parsing techniques
  - ▶ Parsing as constraint satisfaction
  - ▶ Transition-based parsing
  - ▶ Hybrid methods



# Chart Parsing Techniques

- ▶ Context-free dependency grammar:

$$H \rightarrow L_1 \cdots L_m h R_1 \cdots R_n$$

- ▶ Parsing methods:
  - ▶ Standard chart parsing techniques (CKY, Earley, etc.)
  - ▶ Goes back to the 1960s [Hays 1964, Gaifman 1965]
  - ▶ Grammar can be augmented/replaced with statistical model
  - ▶ Efficiency gains thanks to dependency tree constraints

# Eisner's Algorithm

- ▶ In standard CKY style parsing, chart items are trees
- ▶ Eisner's algorithm [Eisner 1996, Eisner 2000]:
  - ▶ Split head representation
  - ▶ Chart items are (complete or incomplete) half-trees

CKY	Eisner
$C[i, h, l, h', j] \Rightarrow O(n^5)$	$C[h, h', j] \Rightarrow O(n^3)$






## Statistical Models

- ▶ Chart parsing requires factorized scoring function  $\mathcal{F}$ :

$$T^* = \operatorname{argmax}_{T \in \mathcal{T}(S)} \mathcal{F}(S, T)$$

$$\mathcal{F}(S, T) = \sum_{g \in T} f(S, g)$$

- ▶ Size of subgraph  $g$  determines model complexity

Model	Subgraph	TC	PTB	Reference
1st-order		$O(n^3)$	90.9	[McDonald et al. 2005a]
2nd-order		$O(n^3)$	91.5	[McDonald and Pereira 2006]
3rd-order		$O(n^4)$	93.0	[Koo and Collins 2010]



## Beyond Projective Trees

- ▶ Context-free techniques are limited to projective trees
- ▶ Extension to mildly non-projective trees:
  - ▶ Well-nested trees with gap degree 1 in  $O(n^7)$  time  
[Kuhlmann and Satta 2009, Gómez-Rodríguez et al. 2009]
- ▶ Post-processing techniques:
  - ▶ 2nd-order model + hill-climbing [McDonald and Pereira 2006]
  - ▶ Can handle non-projective arcs as well as multiple heads
  - ▶ Top-scoring model in CoNLL-06 [**MSTParser**]



## Parsing as Constraint Satisfaction

- ▶ Constraint dependency grammar [Maruyama 1990]:
  - ▶ Boolean variables  $h_1, \dots, h_n$  with domain  $\{0, 1, \dots, n\}$
  - ▶ Grammar  $\mathcal{G}$  = set of boolean constraints
  - ▶ Parsing = search for tree in  $\{T \in \mathcal{T}(S) \mid \forall c \in \mathcal{G} : c(S, T)\}$
- ▶ Adding soft weighted constraints [Menzel and Schröder 1998]:

$$T^* = \operatorname{argmax}_{T \in \mathcal{T}(S)} \prod_{c: \neg c(S, T)} f(c)$$

- ▶ Characteristics:
  - ▶ Non-projective trees easily accommodated
  - ▶ Constraints not inherently restricted to local subgraphs
  - ▶ Exact inference intractable except in restricted cases



## Approaches to Inference

- ▶ Maximum spanning tree parsing [McDonald et al. 2005b]:
  - ▶ First-order model: constraints restricted to single arcs
  - ▶  $T^*$  = maximum spanning tree in complete graph
  - ▶ Exact parsing with non-projective trees in  $O(n^2)$  time
  - ▶ “An island of tractability” (D. Smith)
- ▶ Approximate inference for higher-order models:
  - ▶ Transformational search [Foth et al. 2004]
  - ▶ Gibbs sampling [Nakagawa 2007]
  - ▶ Loopy belief propagation [Smith and Eisner 2008]
  - ▶ Linear programming [Riedel and Clarke 2006, Martins et al. 2009]



## Transition-Based Approaches

- ▶ Transition-based dependency parsing:
  - ▶ Define a transition system for dependency parsing
  - ▶ Train a classifier for predicting the next transition
  - ▶ Use the classifier to do deterministic parsing
- ▶ Open source implementation:
  - ▶ **MaltParser** [Nivre et al. 2006]  
<http://maltparser.org>
- ▶ Characteristics:
  - ▶ Highly efficient – linear time complexity for projective trees
  - ▶ History-based feature models with unrestricted scope
  - ▶ Sensitive to local prediction errors and error propagation





## Arc-Eager Shift-Reduce Parsing [Nivre 2003]

**Start state:**  $([], [1, \dots, n], \{ \})$

**Final state:**  $(S, [], A)$

**Shift:**  $(S, i|B, A) \Rightarrow (S|i, B, A)$

**Reduce:**  $(S|i, B, A) \Rightarrow (S, B, A)$

**Right-Arc:**  $(S|i, j|B, A) \Rightarrow (S|i|j, B, A \cup \{i \rightarrow j\})$

**Left-Arc:**  $(S|i, j|B, A) \Rightarrow (S, j|B, A \cup \{i \leftarrow j\})$



## Parsing Example

**Stack**

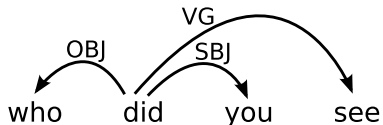
[ ]<sub>s</sub>

**Buffer**

[who, did, you, see]<sub>B</sub>

**Arcs**

{ }





## Parsing Example

**Stack**

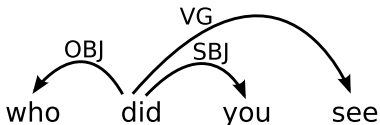
[who]<sub>S</sub>

**Buffer**

[did, you, see]<sub>B</sub>

**Arcs**

{ }





## Parsing Example

**Stack**

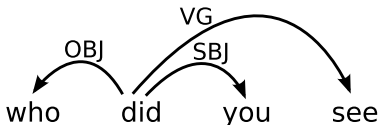
[ ]<sub>S</sub>

**Buffer**

[did, you, see]<sub>B</sub>

**Arcs**

{ who  $\xleftarrow{\text{OBJ}}$  did }





## Parsing Example

**Stack**

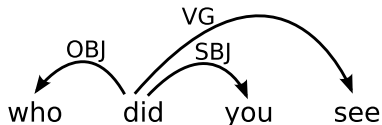
[did]<sub>S</sub>

**Buffer**

[you, see]<sub>B</sub>

**Arcs**

{ who  $\xleftarrow{\text{OBJ}}$  did }





## Parsing Example

**Stack**

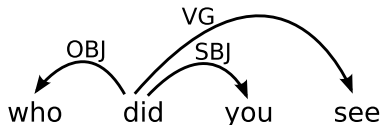
[did, you]<sub>S</sub>

**Buffer**

[see]<sub>B</sub>

**Arcs**

{ who  $\xleftarrow{\text{OBJ}}$  did,  
did  $\xrightarrow{\text{SBJ}}$  you }





## Parsing Example

**Stack**

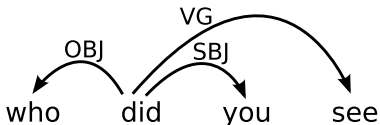
[did]<sub>S</sub>

**Buffer**

[see]<sub>B</sub>

**Arcs**

{ who  $\xleftarrow{\text{OBJ}}$  did,  
did  $\xrightarrow{\text{SBJ}}$  you }





# Parsing Example

**Stack**

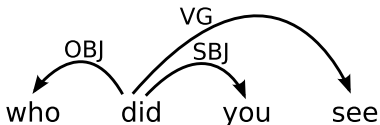
[did, see]<sub>S</sub>

**Buffer**

[ ]<sub>B</sub>

**Arcs**

{ who  $\xleftarrow{\text{OBJ}}$  did,  
did  $\xrightarrow{\text{SBJ}}$  you,  
did  $\xrightarrow{\text{VG}}$  see }







## Statistical Models

- ▶ Parse defined by transition sequence  $C = c_0, c_1, \dots, c_n$
- ▶ Local learning [Yamada and Matsumoto 2003, Nivre et al. 2004]:
  - ▶ Maximize accuracy of local prediction  $f(c_i, c_{i+1})$
  - ▶ Deterministic parsing with 1-best configuration
  - ▶ Top-scoring model in CoNLL-06 [**MaltParser**]
- ▶ Global learning [Titov and Henderson 2007, Zhang and Clark 2008]:
  - ▶ Maximize accuracy over entire sequence  $\sum_{i=0}^{n-1} f(c_i, c_{i+1})$
  - ▶ Beam search with k-best configurations
  - ▶ State of the art on PTB: 82.9 UAS [Zhang and Nivre 2011]



## Beyond Projective Trees

- ▶ Directed acyclic graphs in linear time [Sagae and Tsujii 2008]:

**Right-Arc:**  $(S|i, j|B, A) \Rightarrow (S|i, j|B, A \cup \{i \rightarrow j\})$

**Left-Arc:**  $(S|i, j|B, A) \Rightarrow (S|i, j|B, A \cup \{i \leftarrow j\})$

- ▶ Subset of non-projective trees in linear time [Attardi 2006]:

**Right-Arc2:**  $(S|i|k, j|B, A) \Rightarrow (S|i|k, B, A \cup \{i \rightarrow j\})$

**Left-Arc2:**  $(S|i|k, j|B, A) \Rightarrow (S|k, j|B, A \cup \{i \leftarrow j\})$

- ▶ All non-projective trees in linear expected time [Nivre 2009]:

**Swap:**  $(S|i|k, j|B, A) \Rightarrow (S|i, j|k|B, A)$



## Hybrid Methods

- ▶ Parser combination by voting:
  - ▶ Majority vote for  $h_i$  and  $r_i$  [Zeman and Žabokrtský 2005]
  - ▶ Vote for  $f(S, g)$  in MST parsing [Sagae and Lavie 2006]
  - ▶ Top-ranked system in CoNLL-07 [Hall et al. 2007]
- ▶ Parser combination by stacking:
  - ▶ Let P2 learn from output of P1 [Nivre and McDonald 2008]
  - ▶ Substantial improvement for best systems in CoNLL-06 [Nivre and McDonald 2008, Torres Martins et al. 2008]
- ▶ Parser combination by dual decomposition:
  - ▶ Optimize joint score  $\mathcal{F}_1(T) + \mathcal{F}_2(T)$
  - ▶ 1st-order MST + 3rd-order non-projective chart parsing
  - ▶ State of the art for PDT and CoNLL-06 [Koo et al. 2010]



# Comparative Evaluation

- ▶ Bare-bones dependency parsers against the world
  - ▶ Do we need phrase structure to derive dependency trees?
  - ▶ How do different parsers compare in terms of efficiency?
  - ▶ Do we have a case for Occam's razor?



## English: PTB → Penn2Malt

		<b>UAS</b>
[Yamada and Matsumoto 2003]	<b>Trans-Local</b>	90.3
[Collins 1999]*	<b>PCFG</b>	91.5
[Charniak 2000]*	<b>PCFG</b>	92.1

\* Result not in original paper



## English: PTB → Penn2Malt

		<b>UAS</b>
[Yamada and Matsumoto 2003]	Trans-Local	90.3
[McDonald et al. 2005a]	Chart-1st	90.9
[Collins 1999]*	PCFG	91.5
[McDonald and Pereira 2006]	Chart-2nd	91.5
[Charniak 2000]*	PCFG	92.1

\* Result not in original paper



## English: PTB → Penn2Malt

		<b>UAS</b>
[Yamada and Matsumoto 2003]	Trans-Local	90.3
[McDonald et al. 2005a]	Chart-1st	90.9
[Collins 1999]*	PCFG	91.5
[McDonald and Pereira 2006]	Chart-2nd	91.5
[Charniak 2000]*	PCFG	92.1
[Koo et al. 2010]	Hybrid-Dual	92.5
[Sagae and Lavie 2006]	Hybrid-MST	92.7

\* Result not in original paper



## English: PTB → Penn2Malt

		<b>UAS</b>
[Yamada and Matsumoto 2003]	Trans-Local	90.3
[McDonald et al. 2005a]	Chart-1st	90.9
[Collins 1999]*	PCFG	91.5
[McDonald and Pereira 2006]	Chart-2nd	91.5
[Charniak 2000]*	PCFG	92.1
[Koo et al. 2010]	Hybrid-Dual	92.5
[Sagae and Lavie 2006]	Hybrid-MST	92.7
[Petrov et al. 2006]*	PCFG-Latent	92.8
[Charniak and Johnson 2005]*	PCFG+Rank	93.7

\* Result not in original paper





## English: PTB → Penn2Malt

		<b>UAS</b>
[Yamada and Matsumoto 2003]	Trans-Local	90.3
[McDonald et al. 2005a]	Chart-1st	90.9
[Collins 1999]*	PCFG	91.5
[McDonald and Pereira 2006]	Chart-2nd	91.5
[Charniak 2000]*	PCFG	92.1
[Koo et al. 2010]	Hybrid-Dual	92.5
[Sagae and Lavie 2006]	Hybrid-MST	92.7
[Petrov et al. 2006]*	PCFG-Latent	92.8
[Zhang and Nivre 2011]	Trans-Global	92.9
[Koo and Collins 2010]	Chart-3rd	93.0
[Charniak and Johnson 2005]*	PCFG+Rank	93.7

\* Result not in original paper



## Czech: PDT

		<b>UAS</b>
[Collins 1999]*	PCFG	82.2
[Charniak 2000]*	PCFG	84.3

\* Result not in original paper



## Czech: PDT

		<b>UAS</b>
[Collins 1999]*	PCFG	82.2
[McDonald et al. 2005a]	Chart-1st	83.3
[Charniak 2000]*	PCFG	84.3
[McDonald et al. 2005b]	MST	84.4

\* Result not in original paper



## Czech: PDT

		<b>UAS</b>
[Collins 1999]*	PCFG	82.2
[McDonald et al. 2005a]	Chart-1st	83.3
[Charniak 2000]*	PCFG	84.3
[McDonald et al. 2005b]	MST	84.4
[Hall and Novák 2005]	PCFG+Post	85.0
[McDonald and Pereira 2006]	Chart-2nd+Post	85.2

\* Result not in original paper



## Czech: PDT

		<b>UAS</b>
[Collins 1999]*	PCFG	82.2
[McDonald et al. 2005a]	Chart-1st	83.3
[Charniak 2000]*	PCFG	84.3
[McDonald et al. 2005b]	MST	84.4
[Hall and Novák 2005]	PCFG+Post	85.0
[McDonald and Pereira 2006]	Chart-2nd+Post	85.2
[Zeman and Žabokrtský 2005]	Hybrid-Greedy	86.3
[Koo et al. 2010]	Hybrid-Dual	87.3

\* Result not in original paper



## Czech: PDT

		<b>UAS</b>
[Collins 1999]*	PCFG	82.2
[McDonald et al. 2005a]	Chart-1st	83.3
[Charniak 2000]*	PCFG	84.3
[McDonald et al. 2005b]	MST	84.4
[Hall and Novák 2005]	PCFG+Post	85.0
[McDonald and Pereira 2006]	Chart-2nd+Post	85.2
[Nivre 2009]*	Trans-Local	86.2
[Zeman and Žabokrtský 2005]	Hybrid-Greedy	86.3
[Koo et al. 2010]	Hybrid-Dual	87.3

\* Result not in original paper



## English: PTB → Stanford Dependencies

		LF1	UF1	PTime
MSTParser	Chart-2nd	78.8	82.6	6:01
MaltParser	Trans-Local	81.1	84.8	3:23
Stanford	PCFG	84.2	87.2	11:05
Bikel	PCFG	85.3	88.7	29:57
Charniak	PCFG	87.8	90.5	12:10
Berkeley	PCFG-Latent	87.9	90.5	10:14
Charniak & Johnson	PCFG+Rerank	89.1	91.7	11:18

Cer, D., de Marneffe, M.-C., Jurafsky, D. and Manning, C. (2010) Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy. In *Proceedings of LREC 2010*.

- ▶ Evaluation on collapsed dependencies (lossy conversion)
- ▶ Dependency parsers with default settings (unoptimized)



## French: FTB → Dependencies

		<b>LAS</b>	<b>UAS</b>	<b>PTime</b>
Berkeley	PCFG-Latent	85.6	89.6	12:46
MaltParser	Trans-Local	86.7	89.3	1:25
MSTParser	Chart-2nd	87.6	90.3	14:39

Candito, M. Nivre, J. Denis, P. and Henestroza Anguiano, E. (2010) Benchmarking of Statistical Dependency Parsers for French. In *Coling 2010: Posters*, pp. 108–116.

- ▶ Berkeley most accurate PCFG parser [Seddah et al. 2009]
- ▶ Very similar accuracy across parsers
- ▶ Transition-based parser ten times faster than the others





## Conclusion

- ▶ Bare-bones dependency parsing:
  - ▶ Competitive in terms of parsing accuracy
  - ▶ Often superior in terms of run-time efficiency
  - ▶ Still a field in very rapid development ...
- ▶ Occam's razor?
  - ▶ The jury is still out ...
  - ▶ But if all you want is a dependency tree ...





- ▶ Giuseppe Attardi. 2006. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 166–170.
- ▶ Gosse Bouma, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2005. Question answering for dutch using dependency relations. In *Working Notes of the 6th Workshop of the Cross-Language Evaluation Forum (CLEF 2005)*.
- ▶ Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164.
- ▶ Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine  $n$ -best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–180.
- ▶ Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 132–139.
- ▶ Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 16–23.
- ▶ Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- ▶ Aron Culotta and Jeffery Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–429.
- ▶ Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*.



- ▶ Yuan Ding and Martha Palmer. 2004. Synchronous dependency insertion grammars: A grammar formalism for syntax based statistical MT. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*, pages 90–97.
- ▶ Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 340–345.
- ▶ Jason M. Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In Harry Bunt and Anton Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*, pages 29–62. Kluwer.
- ▶ Kilian Foth, Michael Daum, and Wolfgang Menzel. 2004. A broad-coverage parser for German based on defeasible constraints. In *Proceedings of KONVENS 2004*, pages 45–52.
- ▶ Haim Gaifman. 1965. Dependency systems and phrase-structure systems. *Information and Control*, 8:304–337.
- ▶ Carlos Gómez-Rodríguez, 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 (EACL)*, pages 291–299.
- ▶ Jan Hajič, Barbora Vidova Hladka, Jarmila Panevová, Eva Hajičová, Petr Sgall, and Petr Pajas. 2001. Prague Dependency Treebank 1.0. LDC, 2001T10.
- ▶ Keith Hall and Vaclav Novák. 2005. Corrective modeling for non-projective dependency parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, pages 42–52.
- ▶ Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryiğit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? A study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 933–939.
- ▶ David G. Hays. 1964. Dependency theory: A formalism and some observations. *Language*, 40:511–525.
- ▶ Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430.



- ▶ Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–11.
- ▶ Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1288–1298.
- ▶ Matthias Trautner Kromann. 2003. The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 217–220.
- ▶ Marco Kuhlmann and Giorgio Satta. 2009. Treebank grammar techniques for non-projective dependency parsing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 478–486.
- ▶ Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.
- ▶ Andre Martins, Noah Smith, and Eric Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pages 342–350.
- ▶ Hiroshi Maruyama. 1990. Structural disambiguation with constraint propagation. In *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL)*, pages 31–38.
- ▶ Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 81–88.
- ▶ Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 91–98.



- ▶ Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530.
- ▶ Wolfgang Menzel and Ingo Schröder. 1998. Decision procedures for dependency parsing using graded constraints. In *Proceedings of the Workshop on Processing of Dependency-Based Grammars (ACL-COLING)*, pages 78–87.
- ▶ Tetsuji Nakagawa. 2007. Multilingual dependency parsing using global features. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 952–956.
- ▶ Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 950–958.
- ▶ Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL)*, pages 49–56.
- ▶ Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 2216–2219.
- ▶ Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 915–932.
- ▶ Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.
- ▶ Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer.



- ▶ Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pages 351–359.
- ▶ Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.
- ▶ Sebastian Riedel and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 129–137.
- ▶ Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 129–132.
- ▶ Kenji Sagae and Jun'ichi Tsujii. 2008. Shift-reduce dependency DAG parsing. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 753–760.
- ▶ Djamel Seddah, Marie Candito, and Benoît Crabbé. 2009. Cross parser evaluation : a french treebanks study. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 150–161.
- ▶ David Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 145–156.
- ▶ Ivan Titov and James Henderson. 2007. A latent variable model for generative dependency parsing. In *Proceedings of the 10th International Conference on Parsing Technologies (IWPT)*, pages 144–155.
- ▶ André Filipe Torres Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 157–166.
- ▶ Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 195–206.



- ▶ Daniel Zeman and Zdeněk Žabokrtský. 2005. Improving parsing accuracy by combining diverse dependency parsers. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, pages 171–178.
- ▶ Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 562–571.
- ▶ Yue Zhang and Joakim Nivre. 2011. Transition-based parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.