Dependency Parsing

L445 / L545

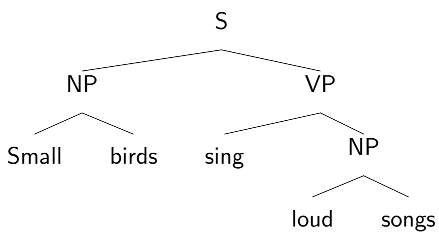
With thanks to Joakim Nivre and Sandra Kübler

Dependency Syntax

- Basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
- In the (translated) words of Lucien Tesnière [Tesnière(1959)]:
  - The sentence is an *organized whole*, the constituent elements of which are *words*. [1.2] Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind perceives *connections*, the totality of which forms the structure of the sentence. [1.3] The structural connections establish *dependency* relations between the words. Each connection in principle unites a *superior* term and an *inferior* term. [2.1] The superior term receives the name *governor*. The inferior term receives the name *subordinate*. Thus, in the sentence *Alfred parle* [. . .], *parle* is the governor and *Alfred* the subordinate. [2.2]

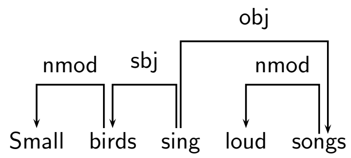
Overview: constituency

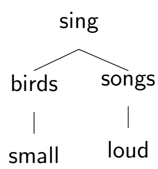
(1) Small birds sing loud songs



Overview: dependency

The corresponding dependency tree representations [Hudson(2000)]:

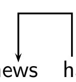




Dependency Structure

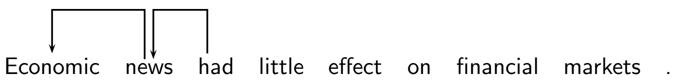
Economic news had little effect on financial markets .

Dependency Structure

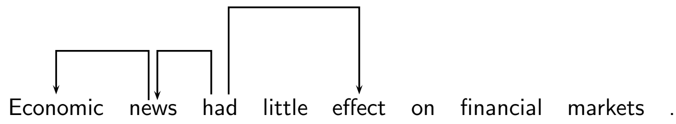


Economic news had little effect on financial markets .

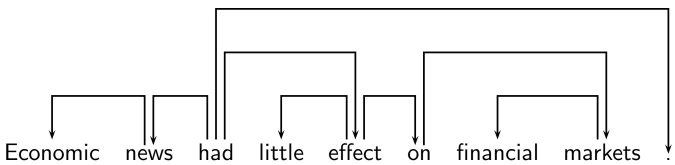
Dependency Structure



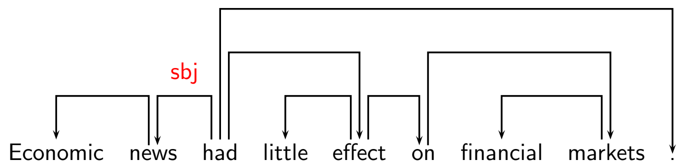
Dependency Structure



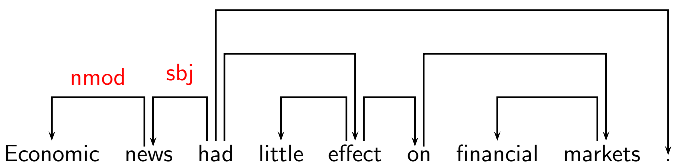
Dependency Structure



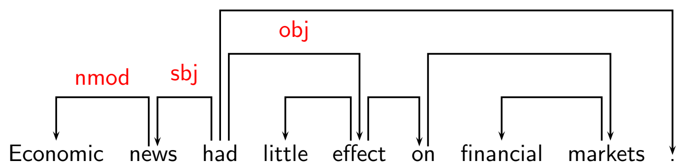
Dependency Structure



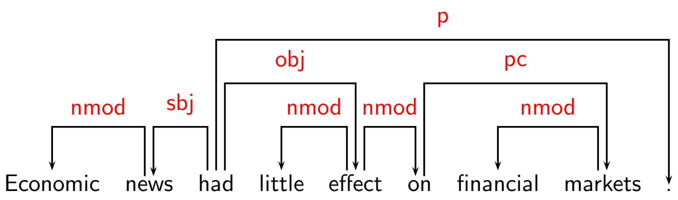
Dependency Structure



Dependency Structure



Dependency Structure



Terminology

Superior	**Inferior**
Head	Dependent
Governor	Modifier
Regent	Subordinate
⋮	⋮

Comparison

- Dependency structures explicitly represent
  - Head-dependent relations (directed arcs)
  - Functional categories (arc labels)
  - Possibly some structural categories (parts-of-speech)
- Phrase structures explicitly represent
  - Phrases (nonterminal nodes)
  - Structural categories (nonterminal labels)
  - Possibly some functional categories (grammatical functions)
- Hybrid representations may combine all elements

Some Theoretical Frameworks

- Word Grammar (WG) [Hudson(1984), Hudson(1990)]
- Functional Generative Description (FGD) [Sgall et al.(1986)Sgall, Hajičová and Panevová]
- Dependency Unification Grammar (DUG) [Hellwig(1986), Hellwig(2003)]
- Meaning-Text Theory (MTT) [Mel'čuk(1988)]
- (Weighted) Constraint Dependency Grammar ([W]CDG) [Maruyama(1990), Harper and Helzerman(1995), Menzel and Schröder(1998), Schröder(2002)]
- Functional Dependency Grammar (FDG) [Tapanainen and Järvinen(1997), Järvinen and Tapanainen(1998)]
- Topological/Extensible Dependency Grammar ([T/X]DG) [Duchier and Debusmann(2001), Debusmann et al.(2004)Debusmann, Duchier and Kruijff]

Dependency Graphs

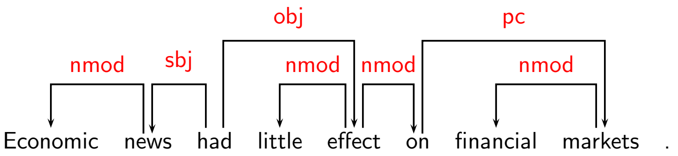
- A dependency structure can be defined as a directed graph G, consisting of
  - A set V of nodes,
  - A set E of arcs (edges),
  - A linear precedence order $<$ on V.
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in E are labeled with dependency types.
- Notational conventions ($i, j \in V$):
  - $i \to j \;\equiv\; (i, j) \in E$
  - $i \to^* j \;\equiv\; i = j \vee \exists k : i \to k, k \to^* j$

Formal Conditions on Dependency Graphs

- G is (weakly) connected:
  - For every node i there is a node j such that $i \to j$ or $j \to i$.
- G is acyclic:
  - If $i \to j$ then not $j \to^* i$.
- G obeys the single-head constraint:
  - If $i \to j$, then not $k \to j$, for any $k \neq i$.
- G is projective:
  - If $i \to j$ then $i \to^* k$, for any k such that $i < k < j$ or $j < k < i$.

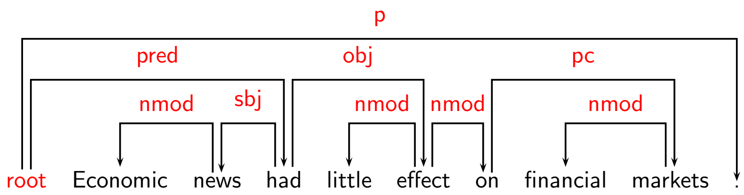
Connectedness, Acyclicity and Single-Head

- Intuitions:
  - Syntactic structure is complete (Connectedness).
  - Syntactic structure is hierarchical (Acyclicity).
  - Every word has at most one syntactic head (Single-Head).
- Connectedness can be enforced by adding a special root node.



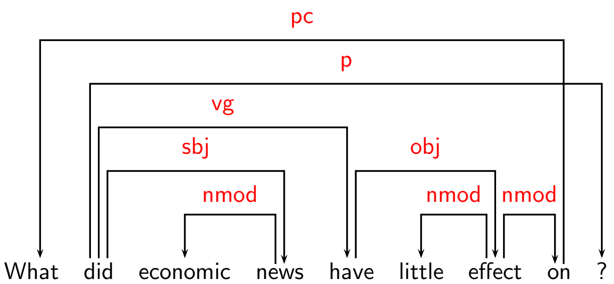
Connectedness, Acyclicity and Single-Head

- Intuitions:
  - Syntactic structure is complete (Connectedness).
  - Syntactic structure is hierarchical (Acyclicity).
  - Every word has at most one syntactic head (Single-Head).
- Connectedness can be enforced by adding a special root node.



Projectivity

- Most theoretical frameworks do not assume projectivity.
- Non-projective structures are needed to account for
  - long-distance dependencies,
  - free word order.



Where we're going

- Dependency parsing:
  - Input: Sentence $x = w_1, \ldots w_n$
  - Output: Dependency graph G
- Focus:
  - Computational methods for dependency parsing
  - Resources for dependency parsing (parsers, treebanks)

Parsing Methods

- Three main traditions:
  - Deterministic parsing (specifically: **Transition-based parsing**)
  - Dynamic programming (specifically: **Graph-based parsing**)
  - Constraint satisfaction (not covered today)
- Special issue:
  - Non-projective dependency parsing

Deterministic Parsing

- Basic idea:
  - Derive a single syntactic representation (dependency graph) through a deterministic sequence of elementary parsing actions
  - Sometimes combined with backtracking or repair
- Motivation:
  - Psycholinguistic modeling
  - Efficiency
  - Simplicity

Covington's Incremental Algorithm

- Deterministic incremental parsing in $O(n^2)$ time by trying to link each new word to each preceding one [Covington(2001)]:

```
PARSE(x = (w_1, ..., w_n))
for i = 1 up to n
    for j = i − 1 down to 1
        LINK(w_i, w_j)
```

$$\textsc{Link}(w_i,\ w_j) = \begin{cases} E \leftarrow E \cup (i,j) & \text{if } w_j \text{ is a dependent of } w_i \\ E \leftarrow E \cup (j,i) & \text{if } w_i \text{ is a dependent of } w_j \\ E \leftarrow E & \text{otherwise} \end{cases}$$

- Different conditions, such as Single-Head and Projectivity, can be incorporated into the LINK operation.

Shift-Reduce Type Algorithms

Transition-based parsing

- Data structures:
  - Stack $[\ldots, w_i]_S$ of partially processed tokens
  - Queue $[w_j, \ldots]_Q$ of remaining input tokens
- Parsing actions built from atomic actions:
  - Adding arcs ($w_i \rightarrow w_j$, $w_i \leftarrow w_j$)
  - Stack and queue operations
- Left-to-right parsing in $O(n)$ time
- Restricted to projective dependency graphs

Yamada's Algorithm

- Three parsing actions:

$$\text{Shift} \quad \frac{[\ldots]_S \quad [w_i, \ldots]_Q}{[\ldots, w_i]_S \quad [\ldots]_Q}$$

$$\text{Left} \quad \frac{[\ldots, w_i, w_j]_S \quad [\ldots]_Q}{[\ldots, w_i]_S \quad [\ldots]_Q \quad w_i \rightarrow w_j}$$

$$\text{Right} \quad \frac{[\ldots, w_i, w_j]_S \quad [\ldots]_Q}{[\ldots, w_j]_S \quad [\ldots]_Q \quad w_i \leftarrow w_j}$$

- Algorithm variants:
  - Originally developed for Japanese (strictly head-final) with only the Shift and Right actions [Kudo and Matsumoto(2002)]
  - Adapted for English (with mixed headedness) by adding the Left action [Yamada and Matsumoto(2003)]
  - Multiple passes over the input give time complexity $O(n^2)$

Nivre's Algorithm

- Four parsing actions:

$$\text{Shift} \quad \frac{[\ldots]_S \quad [w_i, \ldots]_Q}{[\ldots, w_i]_S \quad [\ldots]_Q}$$

$$\text{Reduce} \quad \frac{[\ldots, w_i]_S \quad [\ldots]_Q \quad \exists w_k : w_k \rightarrow w_i}{[\ldots]_S \quad [\ldots]_Q}$$

$$\text{Left-Arc}_r \quad \frac{[\ldots, w_i]_S \quad [w_j, \ldots]_Q \quad \neg\exists w_k : w_k \rightarrow w_i}{[\ldots]_S \quad [w_j, \ldots]_Q \quad w_i \xleftarrow{r} w_j}$$

$$\text{Right-Arc}_r \quad \frac{[\ldots, w_i]_S \quad [w_j, \ldots]_Q \quad \neg\exists w_k : w_k \rightarrow w_j}{[\ldots, w_i, w_j]_S \quad [\ldots]_Q \quad w_i \xrightarrow{r} w_j}$$

- Characteristics:
  - Integrated labeled dependency parsing
  - Arc-eager processing of right-dependents
  - Single pass over the input gives time complexity $O(n)$

Example

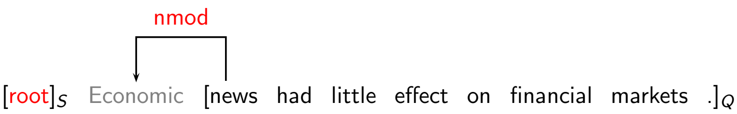
$[\text{root}]_S$ [Economic news had little effect on financial markets $.]_Q$

Example

$[\text{root Economic}]_S$ [news had little effect on financial markets $.]_Q$

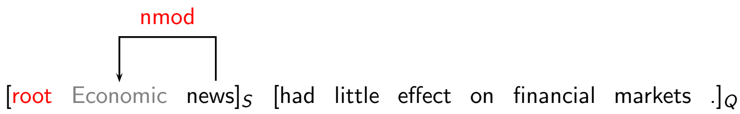
Shift

Example



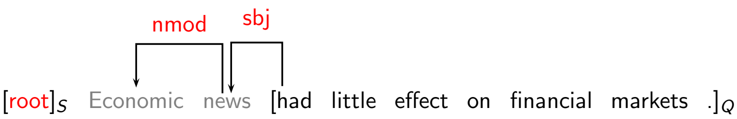
Left-Arc$_{nmod}$

Example



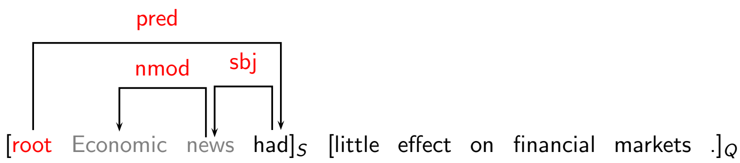
Shift

Example



Left-Arc$_{sbj}$

Example

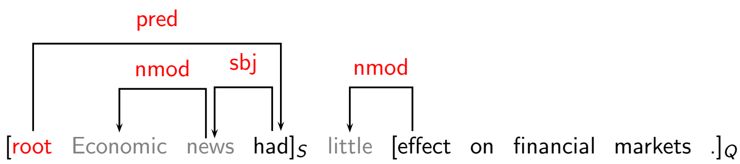


Right-Arc$_{pred}$

Example

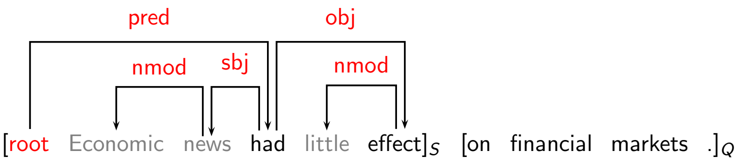
Shift

Example



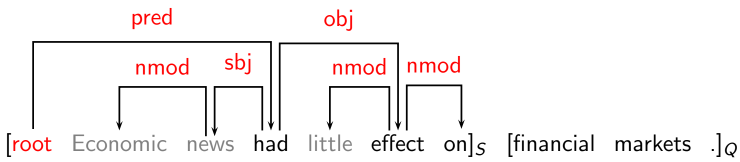
Left-Arc$_{nmod}$

Example



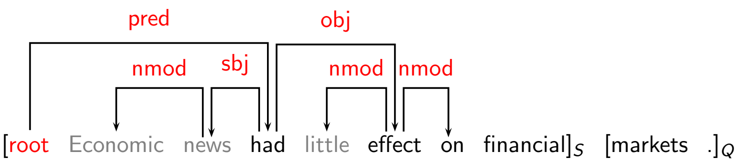
Right-Arc$_{obj}$

Example



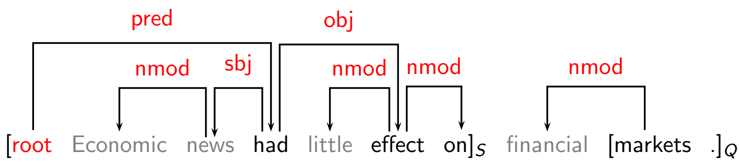
Right-Arc$_{nmod}$

Example



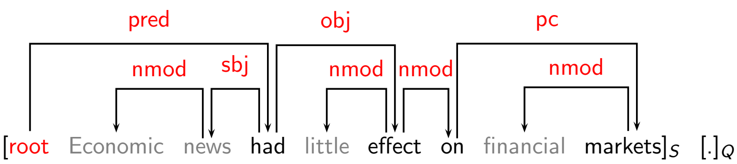
Shift

Example



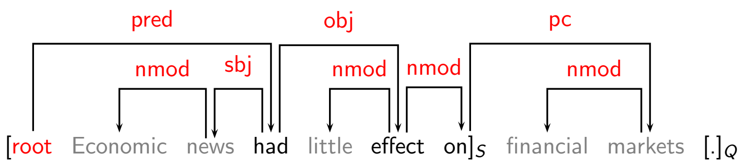
Left-Arc$_{nmod}$

Example



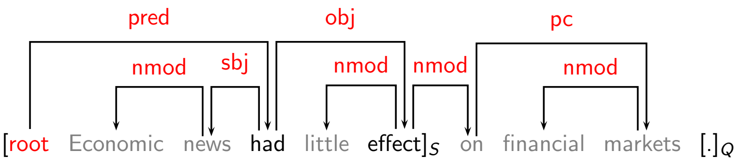
Right-Arc$_{pc}$

Example



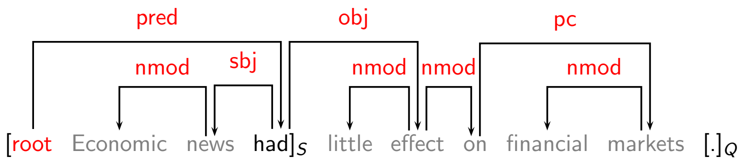
Reduce

Example



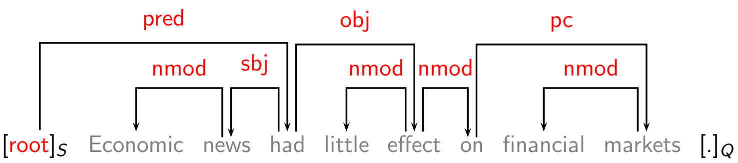
Reduce

Example



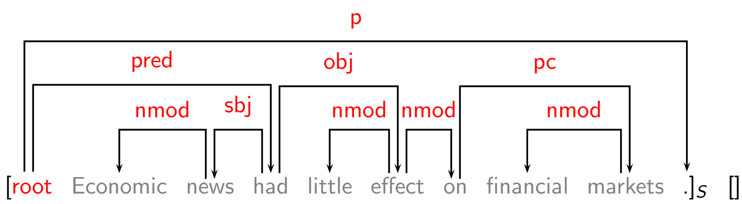
Reduce

Example



Reduce

Example



Right-Arc$_p$

Classifier-Based Parsing

- Data-driven deterministic parsing:
  - Deterministic parsing requires an oracle.
  - An oracle can be approximated by a classifier.
  - A classifier can be trained using treebank data.
- Learning methods:
  - Support vector machines (SVM)
    [Kudo and Matsumoto(2002), Yamada and Matsumoto(2003), Isozaki et al.(2004)Isozaki, Kazawa and Hirao, Cheng et al.(2004)Cheng, Asahara and Matsumoto, Nivre et al.(2006)Nivre, Hall, Nilsson, Eryiğit and Marinov]
  - Memory-based learning (MBL)
    [Nivre et al.(2004)Nivre, Hall and Nilsson, Nivre and Scholz(2004)]
  - Maximum entropy modeling (MaxEnt)
    [Cheng et al.(2005)Cheng, Asahara and Matsumoto]

Feature Models

- Learning problem:
  - Approximate a function from parser states, represented by feature vectors to parser actions, given a training set of gold standard derivations.
- Typical features:
  - Tokens:
    - Target tokens
    - Linear context (neighbors in S and Q)
    - Structural context (parents, children, siblings in G)
  - Attributes:
    - Word form (and lemma)
    - Part-of-speech (and morpho-syntactic features)
    - Dependency type (if labeled)
    - Distance (between target tokens)

Comparing Algorithms

- Parsing algorithm:
  - Nivre's algorithm gives higher accuracy than Yamada's algorithm for parsing the Chinese CKIP treebank [Cheng et al.(2004)Cheng, Asahara and Matsumoto].
- Learning algorithm:
  - SVM gives higher accuracy than MaxEnt for parsing the Chinese CKIP treebank [Cheng et al.(2004)Cheng, Asahara and Matsumoto].
  - SVM gives higher accuracy than MBL with lexicalized feature models for three languages [Hall et al.(2006)Hall, Nivre and Nilsson]:
    - Chinese (Penn)
    - English (Penn)
    - Swedish (Talbanken)

Graph-Based Parsing

From here, we'll look at some slides from NASSLLI 2010 ...

- Buchholz, Sabine and Erwin Marsi (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing.
  In *Proceedings of the Tenth Conference on Computational Natural Language Learning*.
- Cheng, Yuchang, Masayuki Asahara and Yuji Matsumoto (2004). Determinstic Dependency Structure Analyzer for Chinese.
  In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP)*. pp. 500–508.
- Cheng, Yuchang, Masayuki Asahara and Yuji Matsumoto (2005). Machine Learning-Based Dependency Analyzer for Chinese.
  In *Proceedings of International Conference on Chinese Computing (ICCC)*. pp. ?–?
- Chu, Y. J. and T. J. Liu (1965). On the Shortest Arborescence of a Directed Graph.
  Science Sinica 14, 1396–1400.
- Collins, Michael (1999). Head-Driven Statistical Models for Natural Language Parsing.
  Ph.D. thesis, University of Pennsylvania.
- Covington, Michael A. (2001). A Fundamental Algorithm for Dependency Parsing.
  In *Proceedings of the 39th Annual ACM Southeast Conference*. pp. 95–102.

- Debusmann, Ralph, Denys Duchier and Geert-Jan M. Kruijff (2004). Extensible Dependency Grammar: A New Methodology.
  In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*. pp. 78–85.
- Duchier, Denys and Ralph Debusmann (2001). Topological Dependency Trees: A Constraint-based Account of Linear Precedence.
  In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 180–187.
- Edmonds, J. (1967). Optimum Branchings.
  Journal of Research of the National Bureau of Standards 71B, 233–240.
- Eisner, Jason M. (1996a). *An empirical comparison of probability models for dependency grammar*.
  Tech. Rep. IRCS-96-11, Institute for Research in Cognitive Science, University of Pennsylvania.
- Eisner, Jason M. (1996b). Three new probabilistic models for dependency parsing: An exploration.
  In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*. pp. 340–345.
- Eisner, Jason M. (2000). Bilexical grammars and their cubic-time parsing algorithms.

In Harry Bunt and Anton Nijholt (eds.), *Advances in Probabilistic and Other Parsing Technologies*, Kluwer, pp. 29–62.
- Foth, Kilian, Michael Daum and Wolfgang Menzel (2004). A Broad-Coverage Parser for German Based on Defeasible Constraints.
  In *Proceedings of KONVENS 2004*. pp. 45–52.
- Gaifman, Haim (1965). Dependency systems and phrase-structure systems.
  Information and Control 8, 304–337.
- Hall, Johan, Joakim Nivre and Jens Nilsson (2006). Discriminative Classifiers for Deterministic Dependency Parsing.
  In *Proceedings of COLING-ACL*.
- Hall, Keith and Vaclav Novák (2005). Corrective modeling for non-projective dependency parsing.
  In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*. pp. 42–52.
- Harper, Mary P. and R. A. Helzerman (1995). Extensions to constraint dependency parsing for spoken language processing.
  Computer Speech and Language 9, 187–234.
- Hays, David G. (1964). Dependency Theory: A Formalism and Some Observations.
  Language 40, 511–525.

- Hellwig, Peter (1986). Dependency Unification Grammar.
  In *Proceedings of the 11th International Conference on Computational Linguistics (COLING)*. pp. 195–198.
- Hellwig, Peter (2003). Dependency Unification Grammar.
  In Vilmos Agel, Ludwig M. Eichinger, Hans-Werner Eroms, Peter Hellwig, Hans Jürgen Heringer and Hening Lobin (eds.), *Dependency and Valency*, Walter de Gruyter, pp. 593–635.
- Hudson, Richard A. (1984). *Word Grammar*.
  Blackwell.
- Hudson, Richard A. (1990). *English Word Grammar*.
  Blackwell.
- Hudson, Richard A. (2000). Dependency Grammar Course Notes.
  `http://www.cs.bham.ac.uk/research/conferences/esslli/notes/hudson.\html`.
- Isozaki, Hideki, Hideto Kazawa and Tsutomu Hirao (2004). A Deterministic Word Dependency Analyzer Enhanced with Preference Learning.
  In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*. pp. 275–281.
- Järvinen, Timo and Pasi Tapanainen (1998). Towards an Implementable Dependency Grammar.

In Sylvain Kahane and Alain Polguère (eds.), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*. pp. 1–10.

- Kromann, Matthias Trautner (2005). *Discontinuous Grammar: A Dependency-Based Model of Human Parsing and Language Learning*. Doctoral Dissertation, Copenhagen Business School.
- Kudo, Taku and Yuji Matsumoto (2002). Japanese Dependency Analysis Using Cascaded Chunking. In *Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL)*. pp. 63–69.
- Lombardo, Vincenzio and Leonardo Lesmo (1996). An Earley-type Recognizer for Dependency Grammar. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*. pp. 723–728.
- Maruyama, Hiroshi (1990). Structural Disambiguation with Constraint Propagation. In *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL)*. pp. 31–38.
- McDonald, Ryan, 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)*. pp. 91–98.

- McDonald, Ryan, Kevin Lerman and Fernando Pereira (2006). Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*.
- McDonald, Ryan 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)*. pp. 81–88.
- McDonald, Ryan, 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)*. pp. 523–530.
- Mel'čuk, Igor (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press.
- Menzel, Wolfgang and Ingo Schröder (1998). Decision Procedures for Dependency Parsing Using Graded Constraints. In Sylvain Kahane and Alain Polguère (eds.), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*. pp. 78–87.
- Neuhaus, Peter and Norbert Bröker (1997). The Complexity of Recognition of Linguistically Adequate Dependency Grammars.

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)*. pp. 337–343.

- Nilsson, Jens, Joakim Nivre and Johan Hall (2006). Graph Transformations in Data-Driven Dependency Parsing. In *Proceedings of COLING-ACL*.
- Nivre, Joakim (2003). An Efficient Algorithm for Projective Dependency Parsing. In Gertjan Van Noord (ed.), *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. pp. 149–160.
- Nivre, Joakim (2006). Constraints on Non-Projective Dependency Graphs. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. pp. 73–80.
- Nivre, Joakim, Johan Hall and Jens Nilsson (2004). Memory-Based Dependency Parsing. In Hwee Tou Ng and Ellen Riloff (eds.), *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL)*. pp. 49–56.
- Nivre, Joakim, Johan Hall, Jens Nilsson, Gülsen Eryiğit and Svetoslav Marinov (2006). Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines.

In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*.

- Nivre, Joakim and Jens Nilsson (2005). Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 99–106.
- Nivre, Joakim and Mario Scholz (2004). Deterministic Dependency Parsing of English Text. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*. pp. 64–70.
- Schröder, Ingo (2002). Natural Language Parsing with Graded Constraints. Ph.D. thesis, Hamburg University.
- Sgall, Petr, Eva Hajičová and Jarmila Panevová (1986). *The Meaning of the Sentence in Its Pragmatic Aspects*. Reidel.
- Sleator, Daniel and Davy Temperley (1991). *Parsing English with a Link Grammar*. Tech. Rep. CMU-CS-91-196, Carnegie Mellon University, Computer Science.
- Tapanainen, Pasi and Timo Järvinen (1997). A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. pp. 64–71.
- Tesnière, Lucien (1959). *Éléments de syntaxe structurale*.

Editions Klincksieck.

- Yamada, Hiroyasu and Yuji Matsumoto (2003). Statistical Dependency Analysis with Support Vector Machines. In Gertjan Van Noord (ed.), *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. pp. 195–206.