Introduction to Data-Driven Dependency Parsing

Introductory Course, ESSLLI 2007

Ryan McDonald[1]   Joakim Nivre[2]

[1]Google Inc., New York, USA
E-mail: ryanmcd@google.com

[2]Uppsala University and Växjö University, Sweden
E-mail: nivre@msi.vxu.se

Introduction

- **Dependency parsing:**
  - Syntactic parsing using dependency-based representations.
- **Data-driven models:**
  - Models for dependency parsing based on machine learning.

Overview of the Course

- **Dependency parsing** (Joakim)
- Machine learning methods (Ryan)
- Transition-based models (Joakim)
- Graph-based models (Ryan)
- Loose ends (Joakim, Ryan):
  - Other approaches
  - Empirical results
  - Available software

Lecture 1: Outline

- Dependency syntax:
  - Basic concepts
  - Terminology and notation
  - Dependency graphs
- Dependency parsing:
  - Grammar-driven methods
  - Data-driven methods
- Pros and cons of dependency parsing

Dependency Syntax

- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
- In the words of Lucien Tesnière [Tesnière 1959]:
  - La phrase est un *ensemble organisé* dont les éléments constituants sont les *mots*. [1.2] Tout mot qui fait partie d'une phrase cesse par lui-même d'être isolé comme dans le dictionnaire. Entre lui et ses voisins, l'esprit aperçoit des *connexions*, dont l'ensemble forme la charpente de la phrase. [1.3] Les connexions structurales établissent entre les mots des rapports de *dépendance*. Chaque connexion unit en principe un terme *supérieur* à un terme *inférieur*. [2.1] Le terme supérieur reçoit le nom de *régissant*. Le terme inférieur reçoit le nom de *subordonné*. Ainsi dans la phrase *Alfred parle* [. . .], *parle* est le régissant et *Alfred* le subordonné. [2.2]

Dependency Syntax

- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
- In the 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]

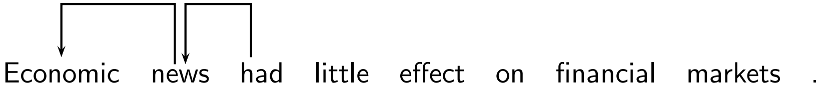
Dependency Structure

Economic news had little effect on financial markets .

Dependency Structure

Economic news had little effect on financial markets .

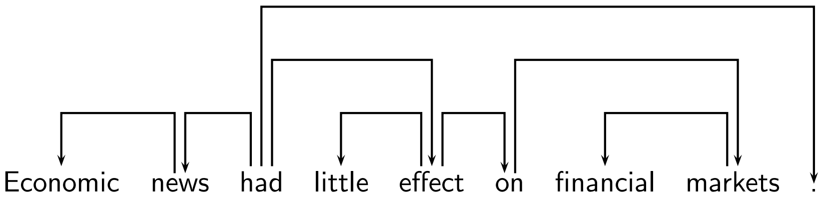
Dependency Structure



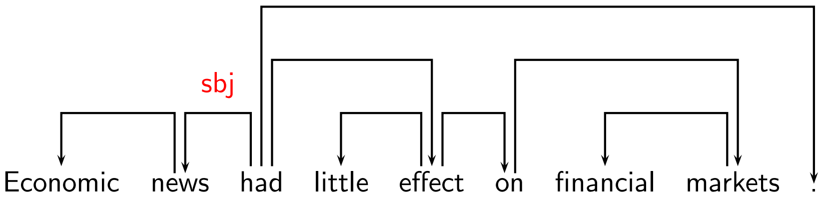
Dependency Structure

Economic news had little effect on financial markets .

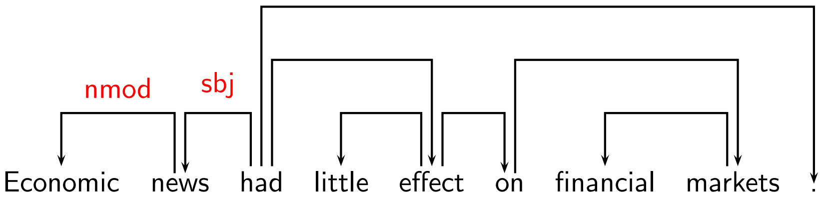
Dependency Structure



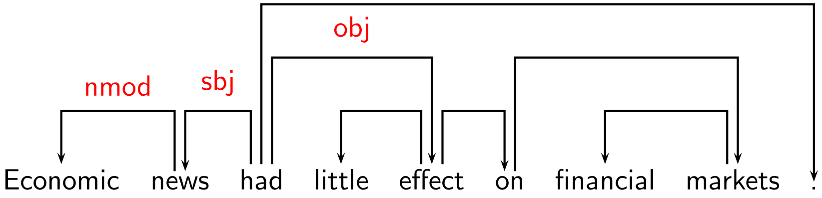
Dependency Structure



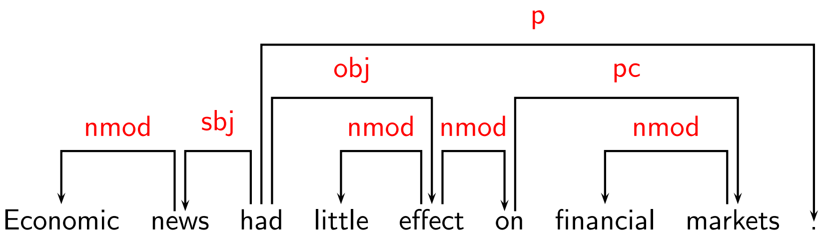
Dependency Structure



Dependency Structure



Dependency Structure



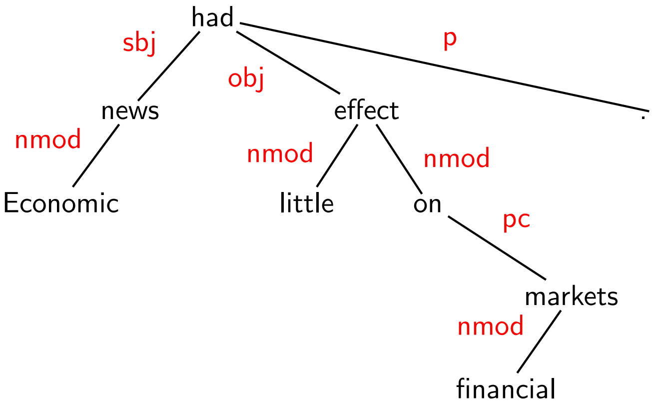
Terminology

Superior	Inferior
Head	Dependent
Governor	Modifier
Regent	Subordinate
⋮	⋮

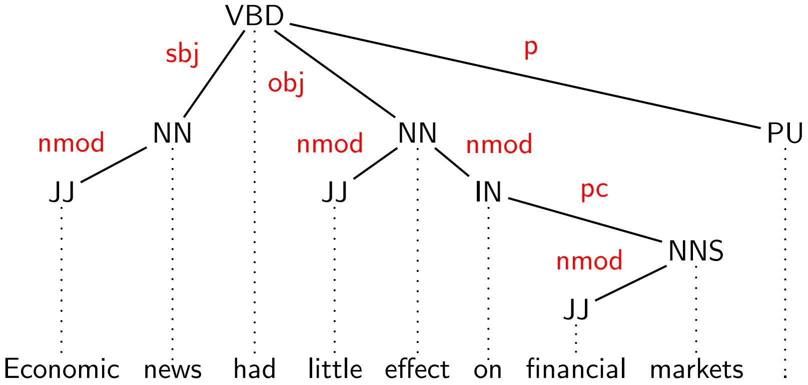
Terminology

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

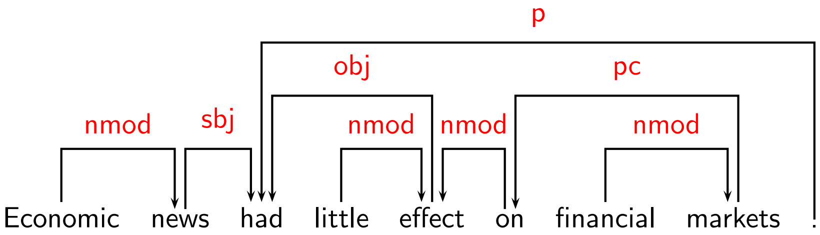
Notational Variants



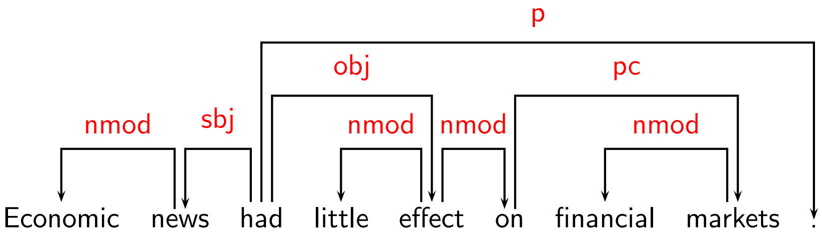
Notational Variants



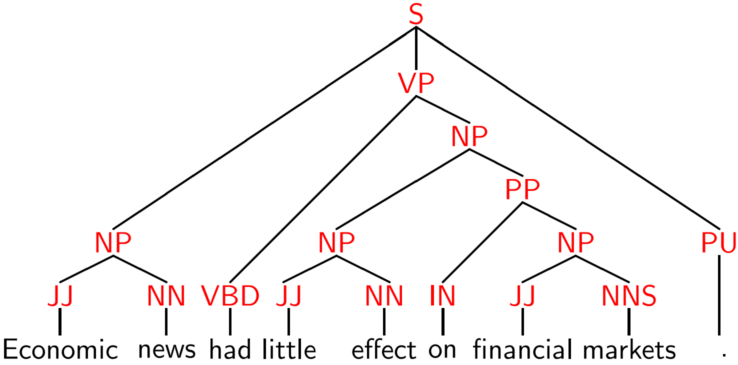
Notational Variants



Notational Variants



Phrase Structure



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]
- 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]

Some Theoretical Issues

- Dependency structure sufficient as well as necessary?
- Mono-stratal or multi-stratal syntactic representations?
- What is the nature of lexical elements (nodes)?
  - Morphemes?
  - Word forms?
  - Multi-word units?
- What is the nature of dependency types (arc labels)?
  - Grammatical functions?
  - Semantic roles?
- What are the criteria for identifying heads and dependents?
- What are the formal properties of dependency structures?

Some Theoretical Issues

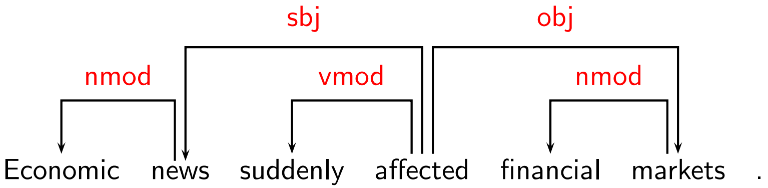
- Dependency structure sufficient as well as necessary?
- Mono-stratal or multi-stratal syntactic representations?
- What is the nature of lexical elements (nodes)?
  - Morphemes?
  - Word forms?
  - Multi-word units?
- What is the nature of dependency types (arc labels)?
  - Grammatical functions?
  - Semantic roles?
- What are the criteria for identifying heads and dependents?
- What are the formal properties of dependency structures?

Criteria for Heads and Dependents

- Criteria for a syntactic relation between a head H and a dependent D in a construction C [Zwicky 1985, Hudson 1990]:
  1. H determines the syntactic category of C; H can replace C.
  2. H determines the semantic category of C; D specifies H.
  3. H is obligatory; D may be optional.
  4. H selects D and determines whether D is obligatory.
  5. The form of D depends on H (agreement or government).
  6. The linear position of D is specified with reference to H.
- Issues:
  - Syntactic (and morphological) versus semantic criteria
  - Exocentric versus endocentric constructions

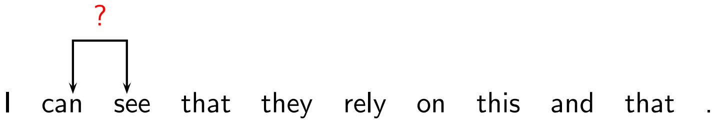
Some Clear Cases

Construction	Head	Dependent
Exocentric	Verb	Subject (sbj)
	Verb	Object (obj)
Endocentric	Verb	Adverbial (vmod)
	Noun	Attribute (nmod)



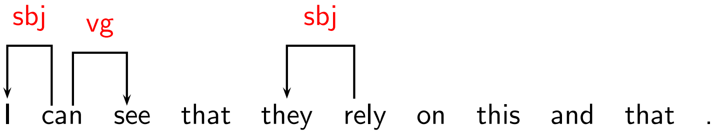
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



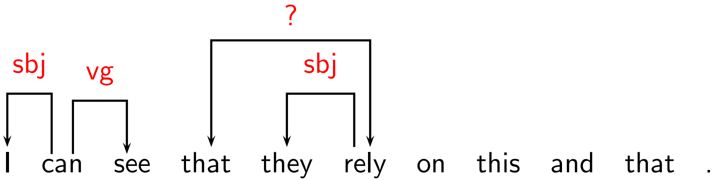
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



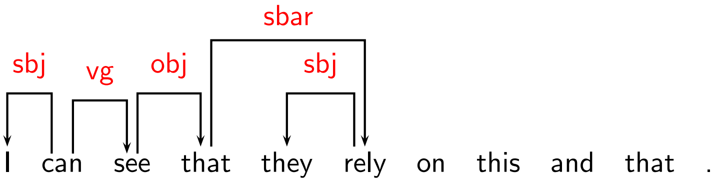
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



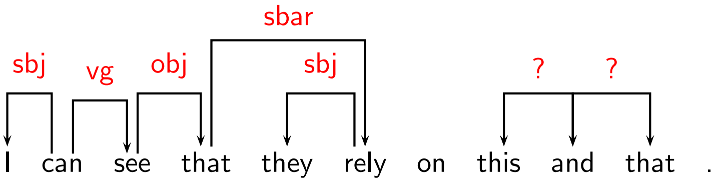
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



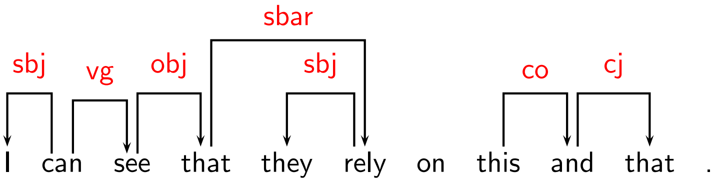
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



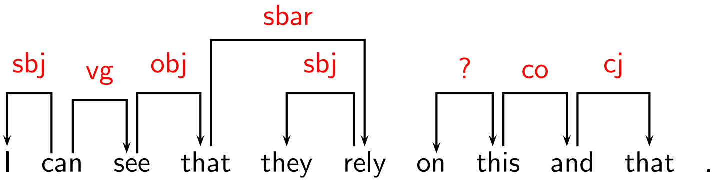
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



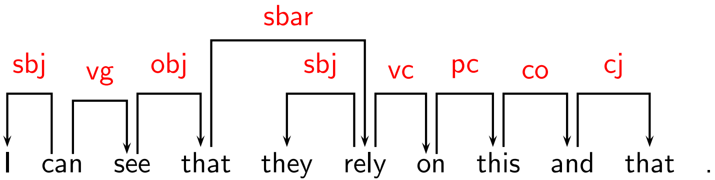
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



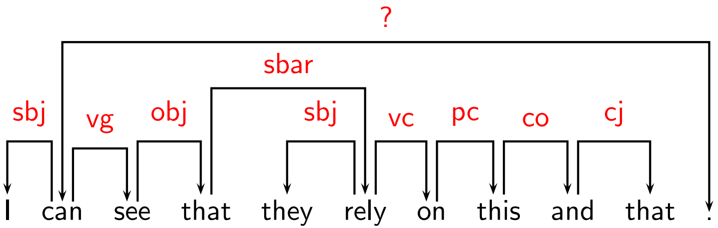
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



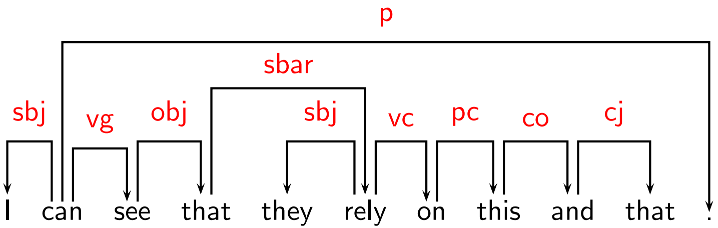
Some Tricky Cases

- Complex verb groups (auxiliary $\leftrightarrow$ main verb)
- Subordinate clauses (complementizer $\leftrightarrow$ verb)
- Coordination (coordinator $\leftrightarrow$ conjuncts)
- Prepositional phrases (preposition $\leftrightarrow$ nominal)
- Punctuation



Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation



Dependency Graphs

- A dependency structure can be defined as a directed graph G, consisting of
  - a set V of nodes (vertices),
  - a set A of arcs (directed edges),
  - a linear precedence order $<$ on V (word order).
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in A are labeled with dependency types:
    - $L = \{l_1, \ldots, l_{|L|}\}$ is the set of permissible arc labels.
    - Every arc in A is a triple (i, j, k), representing a dependency from w_i to w_j with label l_k.

Dependency Graph Notation

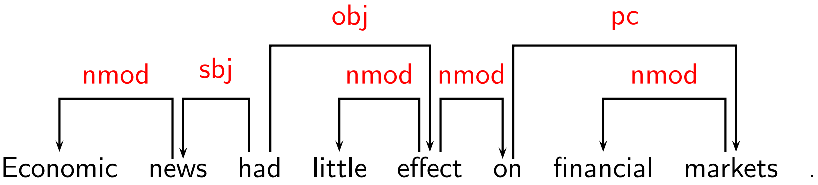
- For a dependency graph $G = (V, A)$
- With label set $L = \{l_1, \ldots, l_{|L|}\}$
  - $i \rightarrow j \quad \equiv \quad \exists k : (i, j, k) \in A$
  - $i \leftrightarrow j \quad \equiv \quad i \rightarrow j \vee j \rightarrow i$
  - $i \rightarrow^* j \quad \equiv \quad i = j \vee \exists i' : i \rightarrow i', \; i' \rightarrow^* j$
  - $i \leftrightarrow^* j \quad \equiv \quad i = j \vee \exists i' : i \leftrightarrow i', \; i' \leftrightarrow^* j$

Formal Conditions on Dependency Graphs

- G is (weakly) connected:
  - If $i, j \in V$, $i \leftrightarrow^* j$.
- G is acyclic:
  - If $i \rightarrow j$, then not $j \rightarrow^* i$.
- G obeys the single-head constraint:
  - If $i \rightarrow j$, then not $i' \rightarrow j$, for any $i' \neq i$.
- G is projective:
  - If $i \rightarrow j$, then $i \rightarrow^* i'$, for any i' such that $i < i' < j$ or $j < i' < i$.

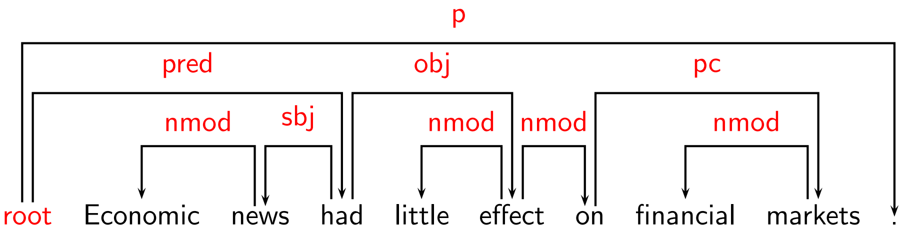
Connectedness, Acyclicity and Single-Head

- 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.



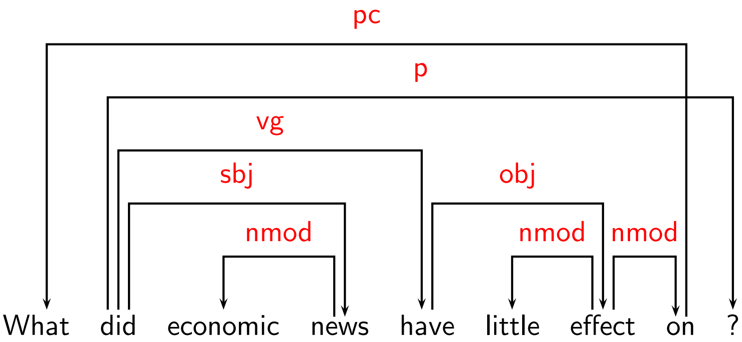
Connectedness, Acyclicity and Single-Head

- 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.



Dependency Parsing

- The problem:
  - Input: Sentence $x = w_0, w_1, \ldots, w_n$ with $w_0 = root$
  - Output: Dependency graph $G = (V, A)$ for x where:
    - $V = \{0, 1, \ldots, n\}$ is the vertex set,
    - A is the arc set, i.e., $(i, j, k) \in A$ represents a dependency from w_i to w_j with label $l_k \in L$
- Two main approaches:
  - Grammar-based parsing
    - Context-free dependency grammar
    - Constraint dependency grammar
  - Data-driven parsing
    - Transition-based models
    - Graph-based models

Context-Free Dependency Grammar

- Dependency grammar as lexicalized context-free grammar:
  - $H \longrightarrow L_1 \cdots L_m \, h \, R_1 \cdots R_n$
  - $H \in V_N; h \in V_T; L_1 \cdots L_m, R_1 \cdots R_n \in V_N^*$
- Standard context-free parsing algorithms (CKY, Earley, etc.)
- Projective, unlabeled dependency trees only
- Weakly equivalent to (arbitrary) context-free grammars [Hays 1964, Gaifman 1965]
- Recent developments:
  - Link Grammar [Sleator and Temperley 1991]
  - Earley-style parser with left-corner filtering [Lombardo and Lesmo 1996]
  - Bilexical grammars [Eisner 1996, Eisner 2000]

Constraint Dependency Grammar

- Parsing as constraint satisfaction [Maruyama 1990]:
  - Grammar consists of a set of boolean constraints, i.e. logical formulas that describe well-formed dependency graphs.
  - Constraint propagation removes candidate graphs that contradict constraints (eliminative parsing).
- Handles non-projective labeled dependency graphs
- Parsing intractable in the general case
- Recent developments:
  - Weighted Constraint Dependency Grammar [Menzel and Schröder 1998, Foth et al. 2004]
  - Probabilistic Constraint Dependency Grammar [Harper and Helzerman 1995, Wang and Harper 2004]
  - Topological/Extensible Dependency Grammar [Duchier and Debusmann 2001, Debusmann et al. 2004]

Transition-Based Models

- Basic idea:
  - Define a transition system (state machine) for mapping a sentence to its dependency graph.
  - Learning: Induce a model for predicting the next state transition, given the transition history.
  - Parsing: Construct the optimal transition sequence, given the induced model.
- Characteristics:
  - Local training of a model for optimal transitions
  - Greedy search/inference

Graph-Based Models

- Basic idea:
  - Define a space of candidate dependency graphs for a sentence.
  - Learning: Induce a model for scoring an entire dependency graph for a sentence.
  - Parsing: Find the highest-scoring dependency graph, given the induced model.
- Characteristics:
  - Global training of a model for optimal dependency graphs
  - Exhaustive search/inference

Pros and Cons of Dependency Parsing

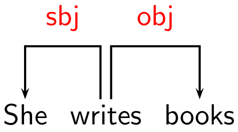
- What are the advantages of dependency-based methods?
- What are the disadvantages?
- Four types of considerations:
  - Complexity
  - Transparency
  - Word order
  - Expressivity

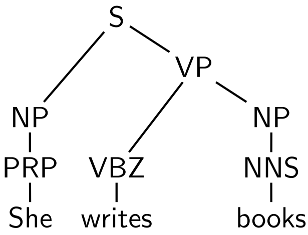
Complexity

- Practical complexity:
  - Given the Single-Head constraint, parsing a sentence $x = w_1, \ldots, w_n$ can be reduced to labeling each token w_i with:
    - a head word h_i,
    - a dependency type d_i.
- Theoretical complexity:
  - By exploiting the special properties of dependency graphs, it is sometimes possible to improve worst-case complexity compared to constituency-based parsing:
    - Lexicalized parsing in $O(n^3)$ time [Eisner 1996]

Transparency

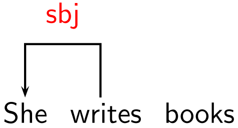
- Direct encoding of predicate-argument structure





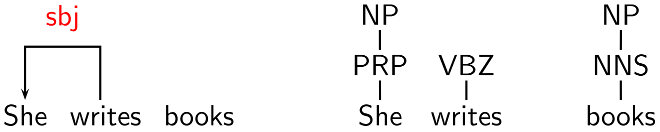
Transparency

- Direct encoding of predicate-argument structure
- Fragments directly interpretable



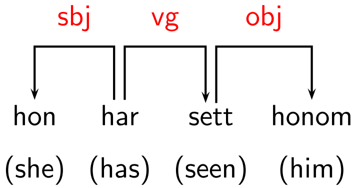
Transparency

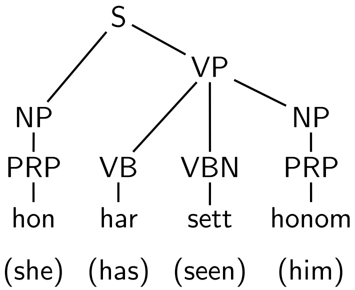
- Direct encoding of predicate-argument structure
- Fragments directly interpretable
- But only with labeled dependency graphs



Word Order

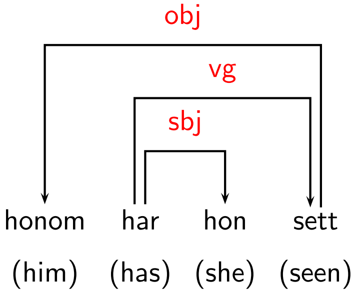
- Dependency structure independent of word order
- Suitable for free word order languages

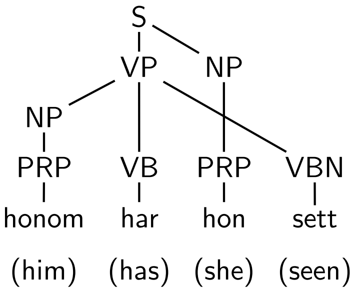




Word Order

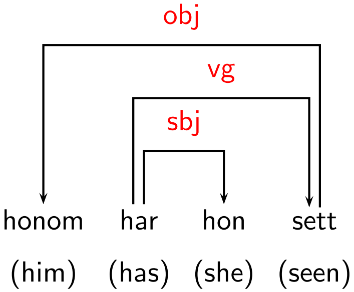
- Dependency structure independent of word order
- Suitable for free word order languages

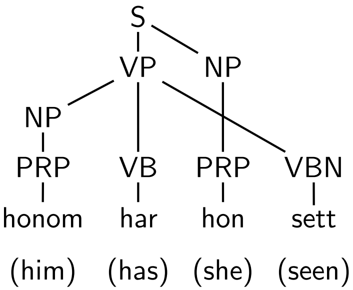




Word Order

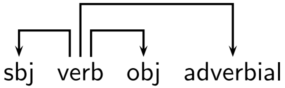
- Dependency structure independent of word order
- Suitable for free word order languages
- But only with non-projective dependency graphs





Expressivity

- Limited expressivity:
  - Every projective dependency grammar has a strongly equivalent context-free grammar, but not vice versa [Gaifman 1965].
  - Impossible to distinguish between phrase modification and head modification in unlabeled dependency structure [Mel'čuk 1988].



V, VP or S modification?

- What about labeled non-projective dependency structures?

Summary

- Dependency syntax – basic concepts
- Dependency parsing – main approaches
- Pros and cons

References and Further Reading

- Ralph Debusmann, 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*, pages 78–85.
- Denys Duchier 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)*, pages 180–187.
- 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.

- Mary P. Harper and R. A. Helzerman. 1995.
  Extensions to constraint dependency parsing for spoken language processing. *Computer Speech and Language*, 9:187–234.

- David G. Hays. 1964.
  Dependency theory: A formalism and some observations. *Language*, 40:511–525.

- Peter Hellwig. 1986.
  Dependency unification grammar. In *Proceedings of the 11th International Conference on Computational Linguistics (COLING)*, pages 195–198.

- Peter Hellwig. 2003.
  Dependency unification grammar. In Vilmos Agel, Ludwig M. Eichinger, Hans-Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, and Hening Lobin, editors, *Dependency and Valency*, pages 593–635. Walter de Gruyter.

- Richard A. Hudson. 1984.
  Word Grammar. Blackwell.

- Richard A. Hudson. 1990.
  English Word Grammar. Blackwell.

- Timo Järvinen and Pasi Tapanainen. 1998.

Towards an implementable dependency grammar. In Sylvain Kahane and Alain Polguère, editors, *Proceedings of the Workshop on Processing of Dependency-Based Grammars*, pages 1–10.

- Vincenzio Lombardo and Leonardo Lesmo. 1996.
  An Earley-type recognizer for dependency grammar. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 723–728.
- Hiroshi Maruyama. 1990.
  Structural disambiguation with constraint propagation. In *Proceedings of the 28th Meeting of the Association for Computational Linguistics (ACL)*, pages 31–38.
- Igor Mel'čuk. 1988.
  Dependency Syntax: Theory and Practice. State University of New York Press.
- Wolfgang Menzel and Ingo Schröder. 1998.
  Decision procedures for dependency parsing using graded constraints. In Sylvain Kahane and Alain Polguère, editors, *Proceedings of the Workshop on Processing of Dependency-Based Grammars*, pages 78–87.
- Ingo Schröder. 2002.
  Natural Language Parsing with Graded Constraints. Ph.D. thesis, Hamburg University.
- Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986.
  The Meaning of the Sentence in Its Pragmatic Aspects. Reidel.

- Daniel Sleator and Davy Temperley. 1991.
  Parsing English with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University, Computer Science.
- Pasi Tapanainen and Timo Järvinen. 1997.
  A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71.
- Lucien Tesnière. 1959.
  Éléments de syntaxe structurale. Editions Klincksieck.
- Wen Wang and Mary P. Harper. 2004.
  A statistical constraint dependency grammar (CDG) parser. In Frank Keller, Stephen Clark, Matthew Crocker, and Mark Steedman, editors, *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL)*, pages 42–29.
- A. M. Zwicky. 1985.
  Heads. *Journal of Linguistics*, 21:1–29.