Einführung in die Dependenzgrammatik

Günter Neumann, LT lab, DFKI

Quelle u. a.: http://www.ryanmcd.com/courses/esslli2007/

Überblick

- Dependenzsyntax - Basiskonzepte
- Dependenzparsing - Hauptansätze
- Dependenzbasierte Baumbanken

Dependenzsyntax

- Die Kernidee:
    - Die syntaktische Struktur besteht aus lexikalischen Elementen, die durch binäre, asymmetrische Beziehungen verlinkt sind. Diese Beziehungen werden als Dependenzen bezeichnet.
- In den Worten von Luciene Tesnière:
    - La phrase est un ensemble organiśe 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]

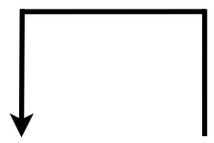
Dependenzsyntax

- Die Kernidee:
    - Die syntaktische Struktur besteht aus lexikalischen Elementen, die durch binäre, asymmetrische Beziehungen verlinkt sind. Diese Beziehungen werden als Dependenzen bezeichnet.
- In den Worten von Luciene Tesnière:
    - Der Satz ist ein organisiertes Ganzes, seine konstituierenden Elemente sind Wörter. [1.2] Jedes zum Satz gehörende Wort endet durch sich selbst isoliert in einem Wörterbuch. Zwischen dem Wort und seinen Nachbarn, erkennt der Verstand Verbindungen, die als Ganzes die Struktur des Satzes formen. [1.3] Diese strukturellen Verbindungen etablieren Abhängigkeitsbeziehungen zwischen den Wörtern. Jede Verbindung kennzeichnet prinzipiell einen übergeordneten und einen untergeordneten Term. [2.1] Der übergeordnete Term wird als regierend bezeichnet. Der untergeordnete Term wird als abhängig bezeichnent. Demnach ist in einem Satz *Alfred spricht*, *spricht* der Regent und *Alfred* der Abhängige. [2.2]

Dependenzstruktur

Economic news had little effect on financial markets .

Dependenzstruktur

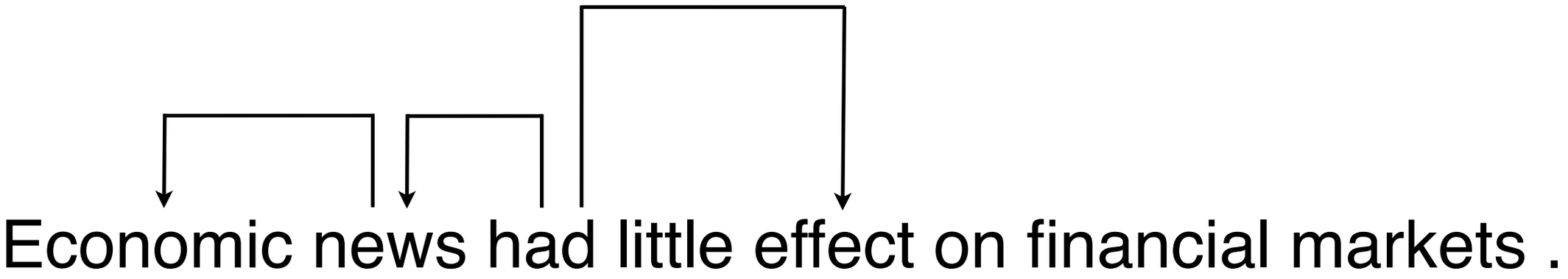


Economic news had little effect on financial markets .

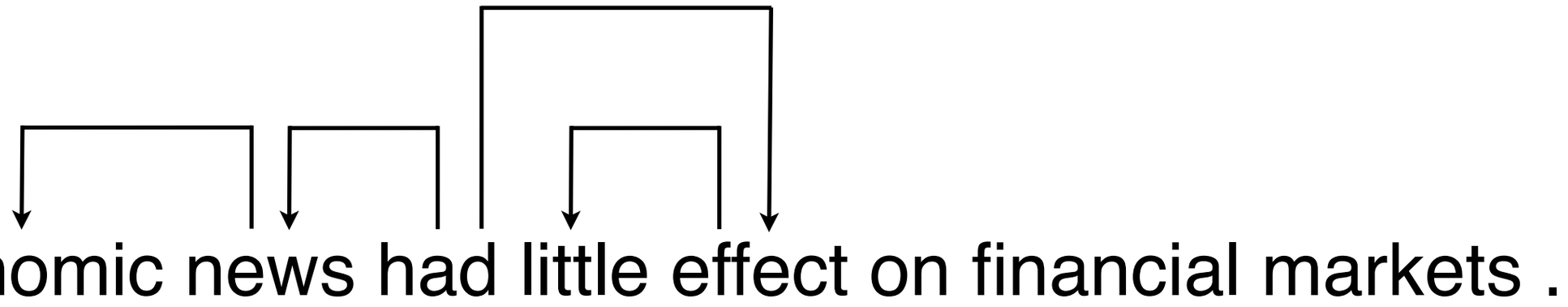
Dependenzstruktur

Economic news had little effect on financial markets .

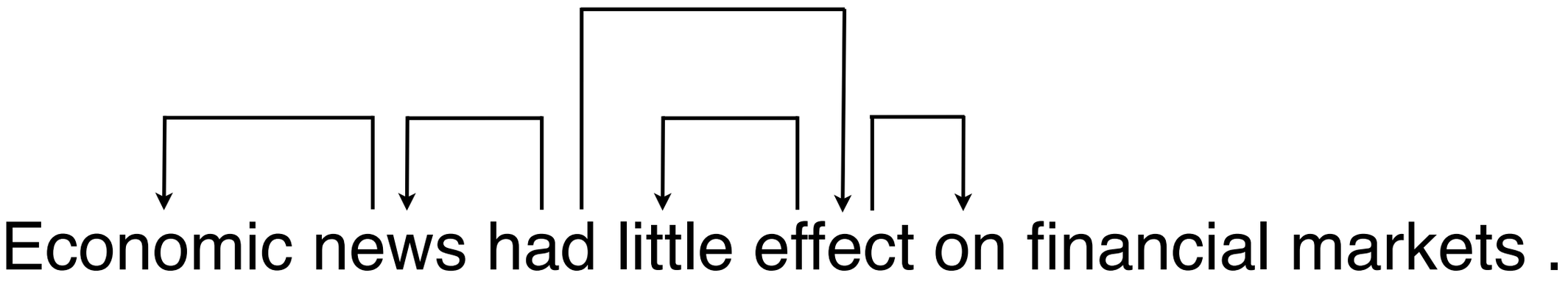
Dependenzstruktur



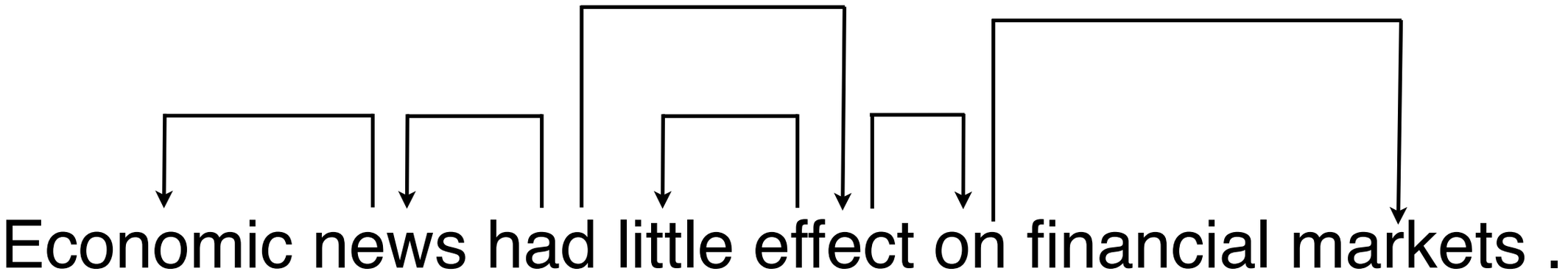
Dependenzstruktur



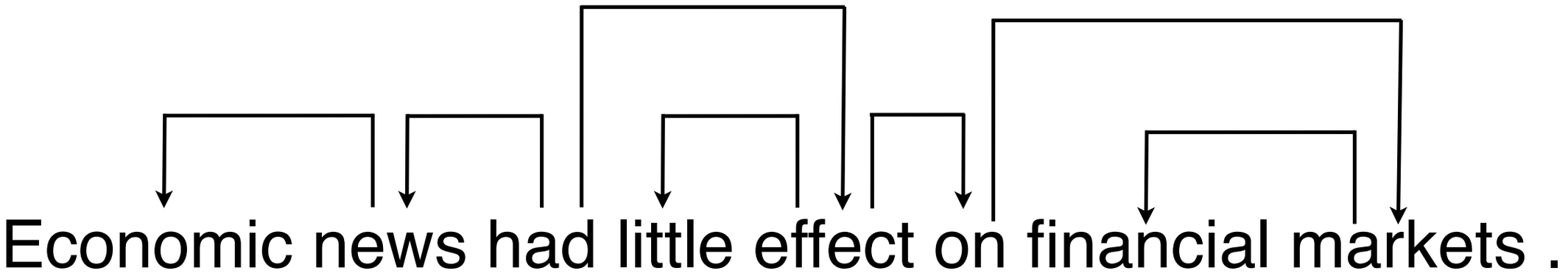
Dependenzstruktur



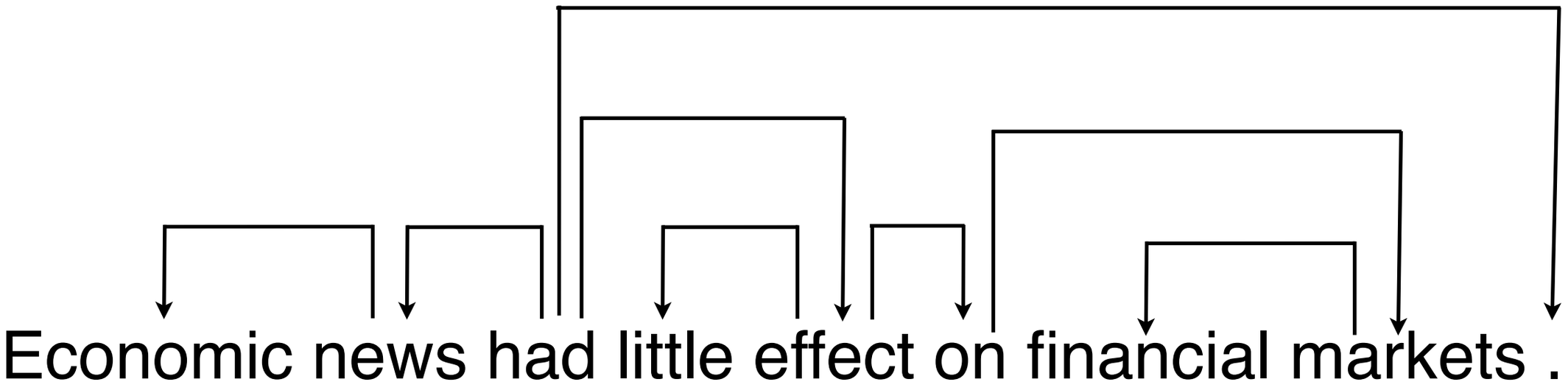
Dependenzstruktur



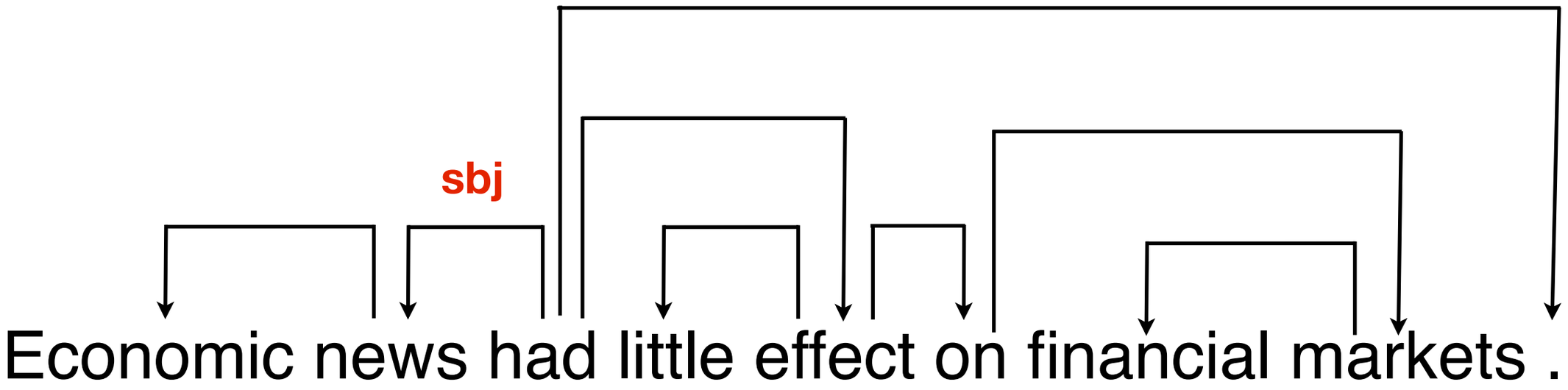
Dependenzstruktur



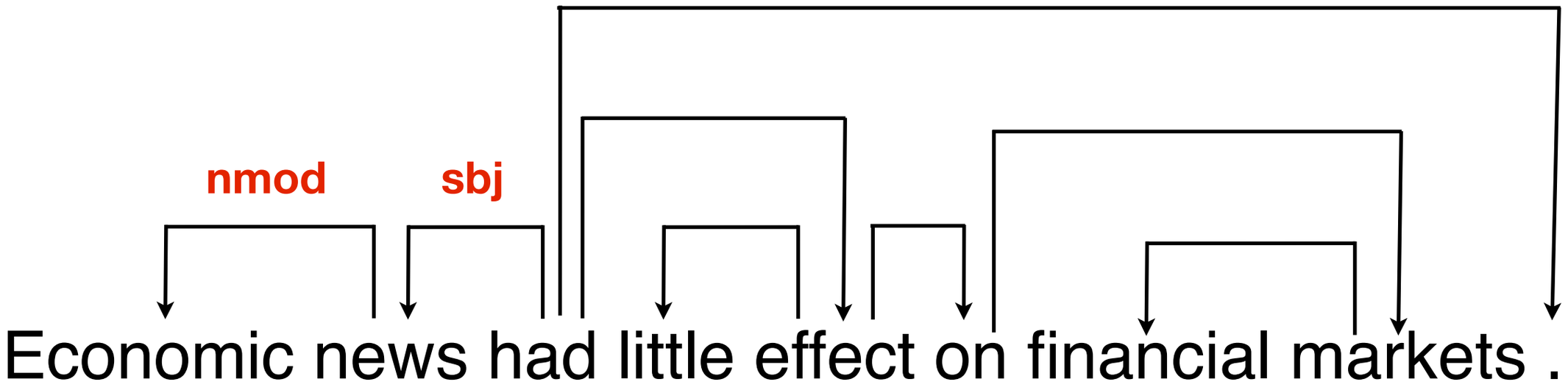
Dependenzstruktur



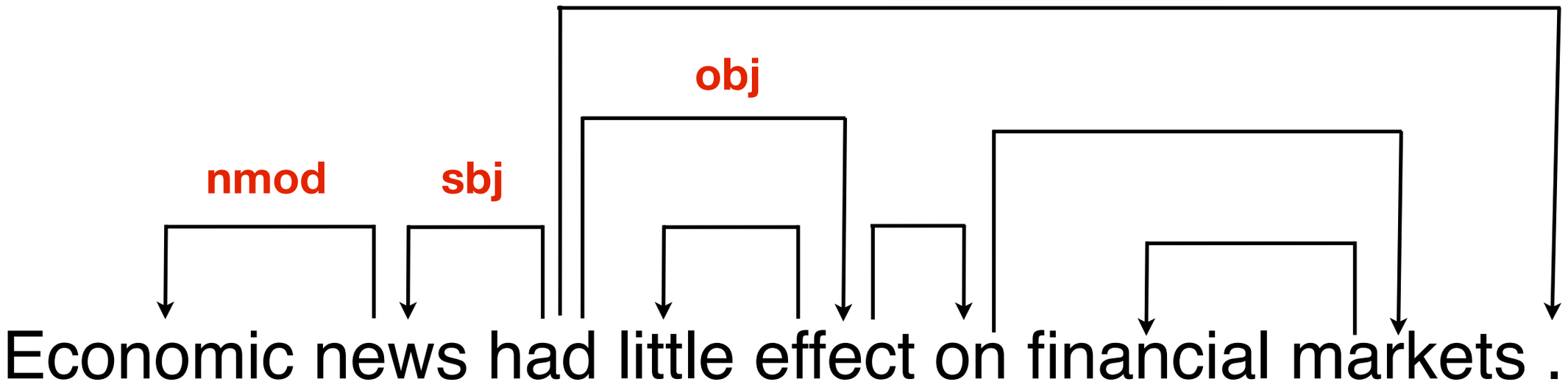
Dependenzstruktur



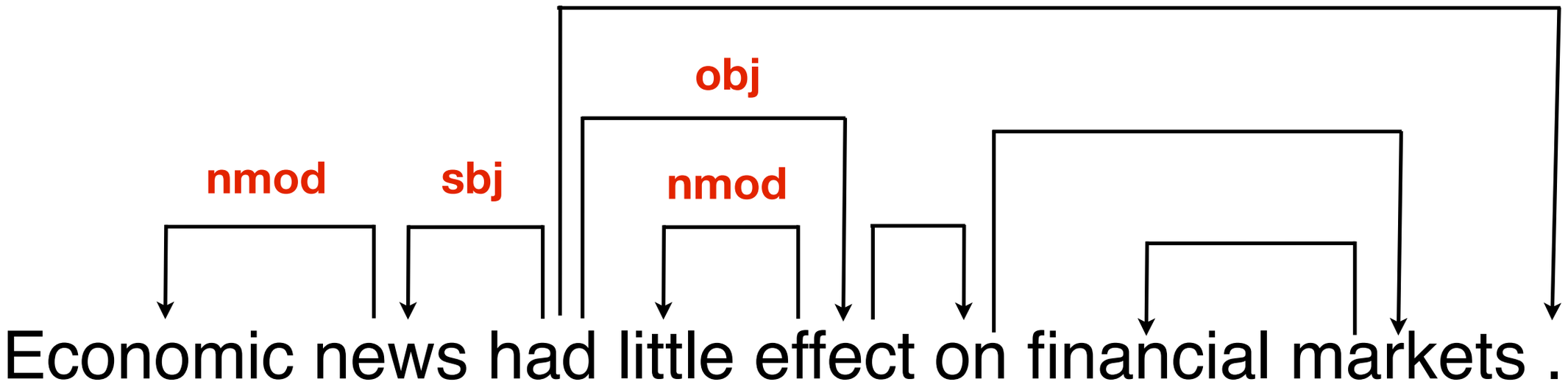
Dependenzstruktur



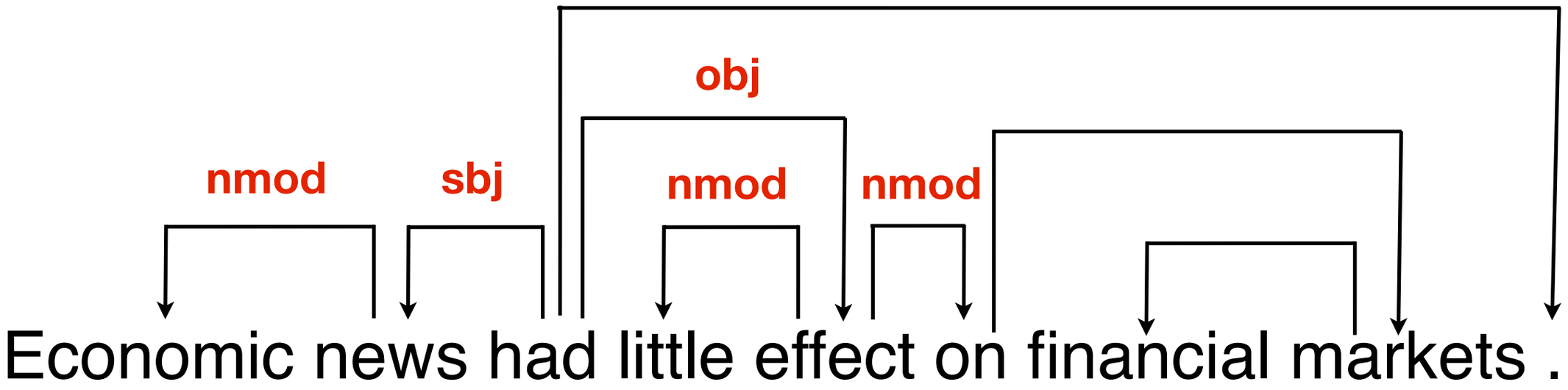
Dependenzstruktur



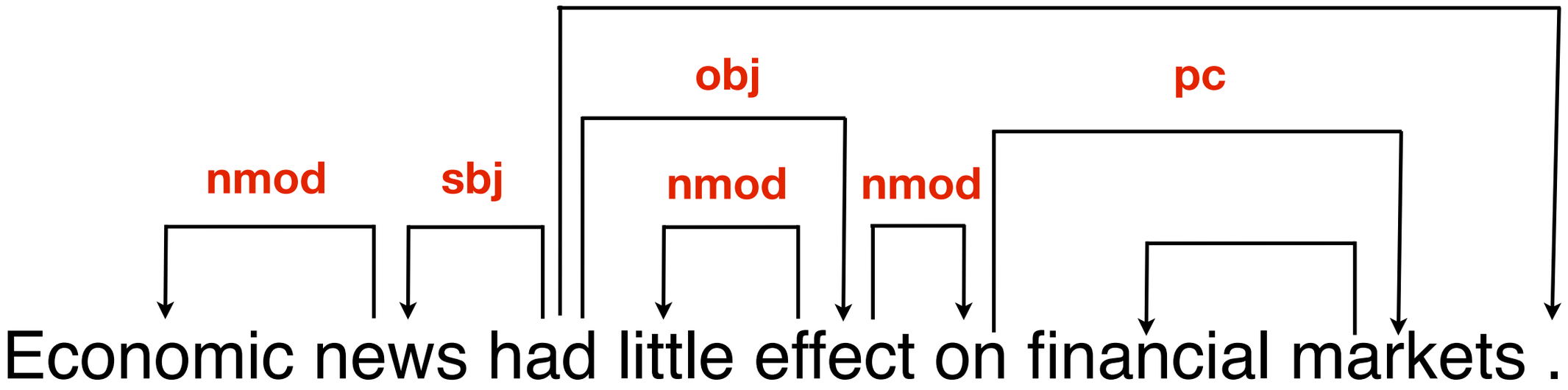
Dependenzstruktur



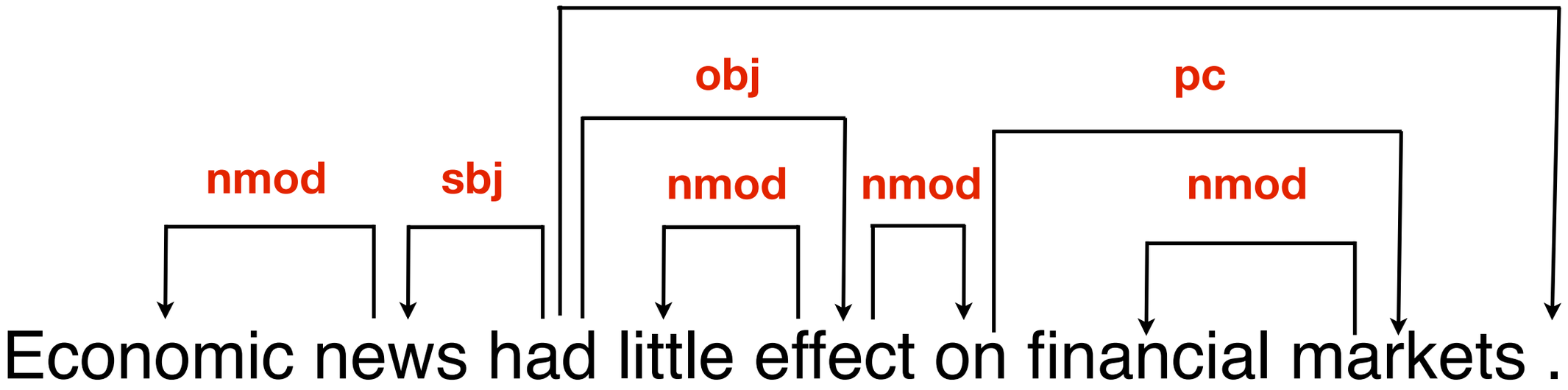
Dependenzstruktur



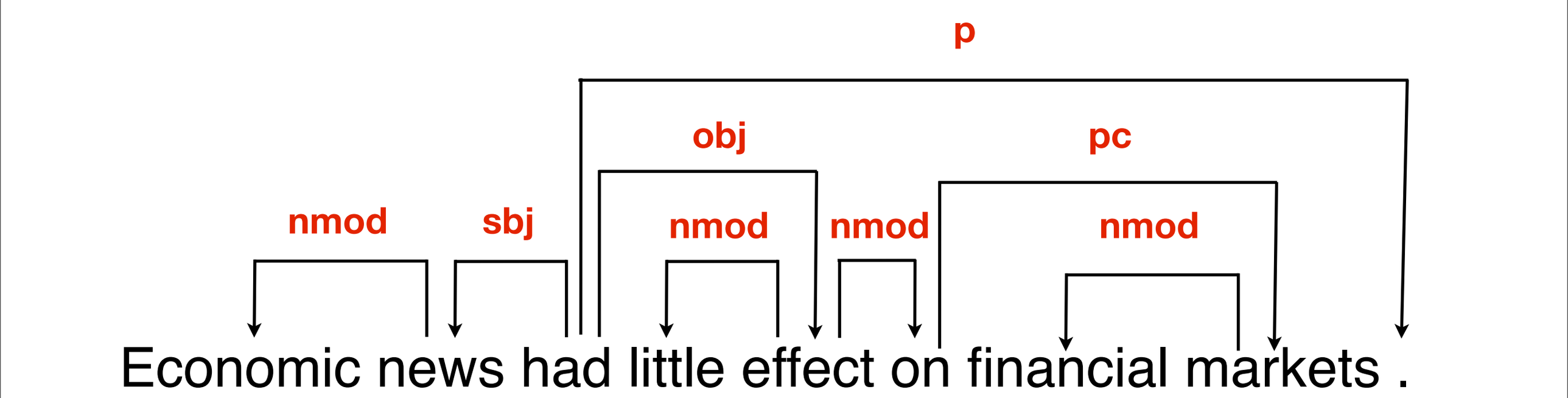
Dependenzstruktur



Dependenzstruktur



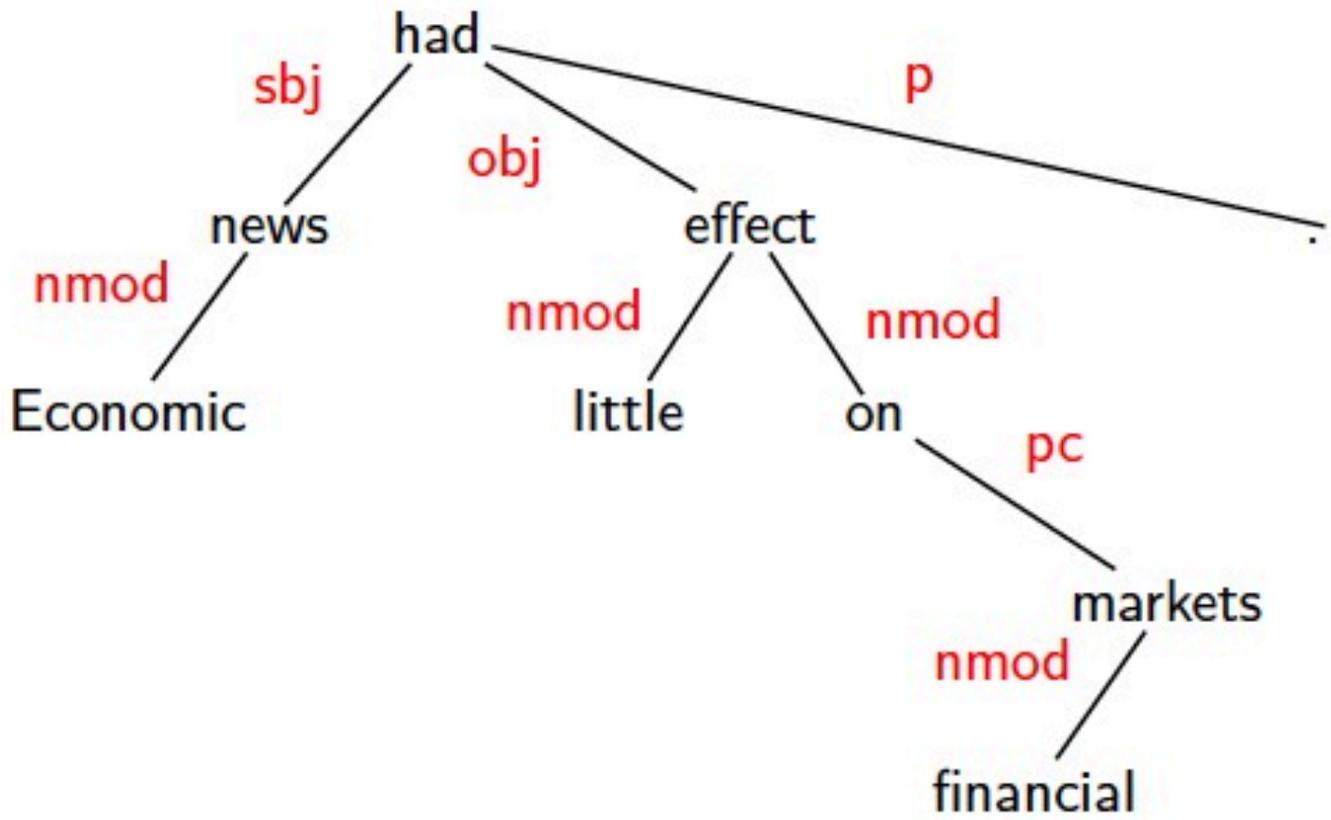
Dependenzstruktur



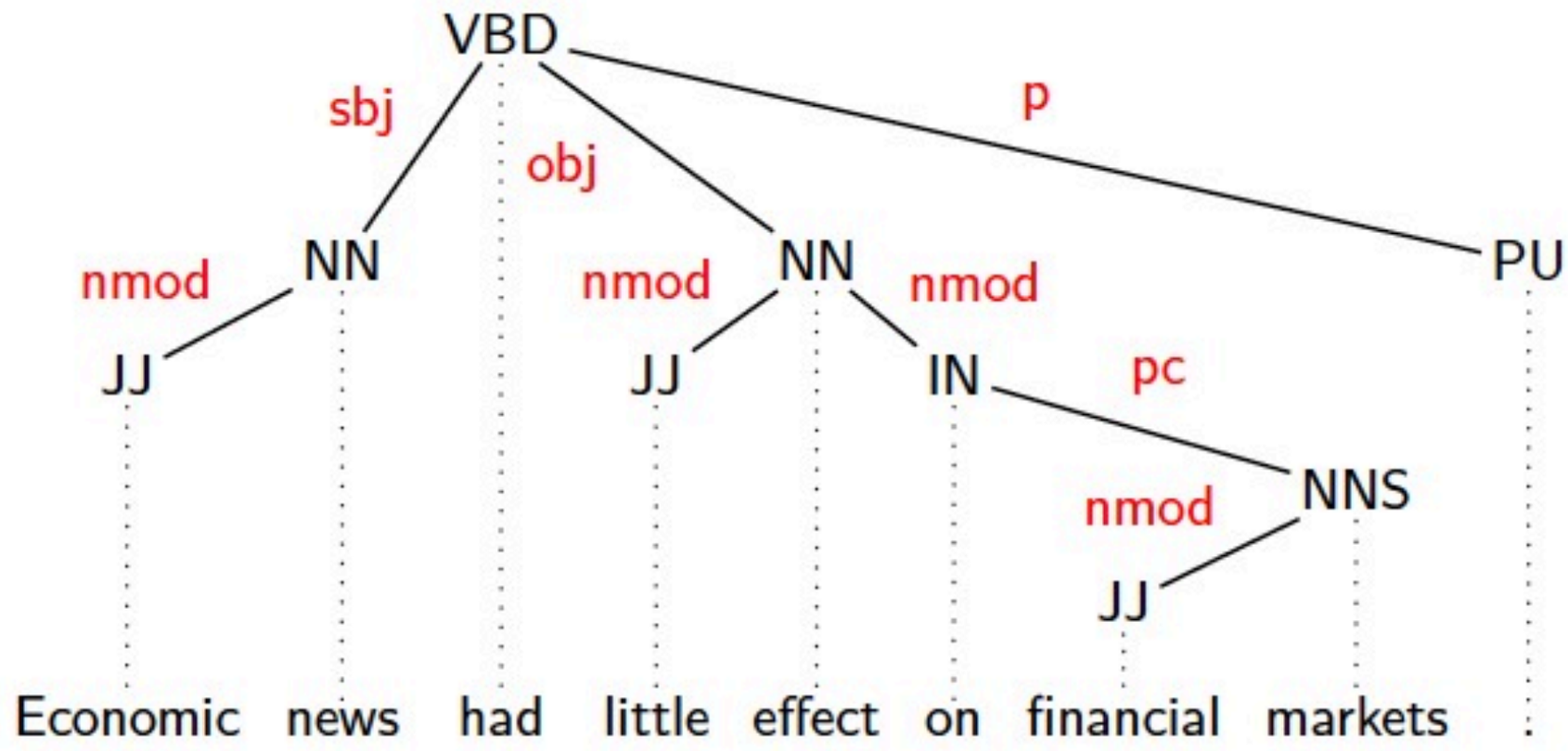
Terminologie

Übergeordnet	Untergeordnet
Kopf	Dependent
Regierender	Modifier
Regent	Abhängiger
...	...

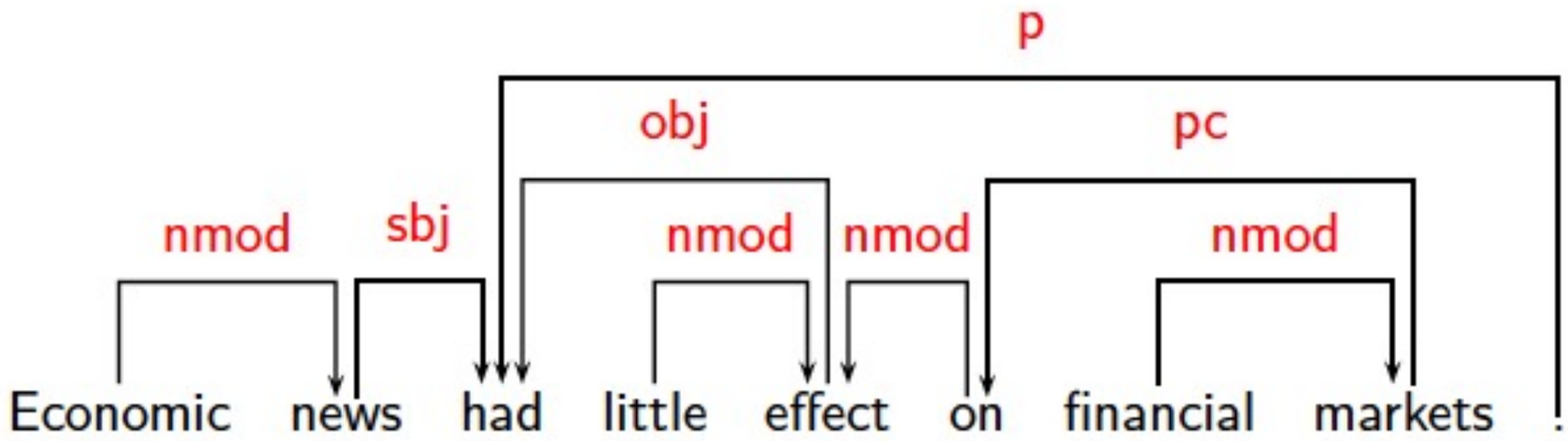
Notationelle Varianten



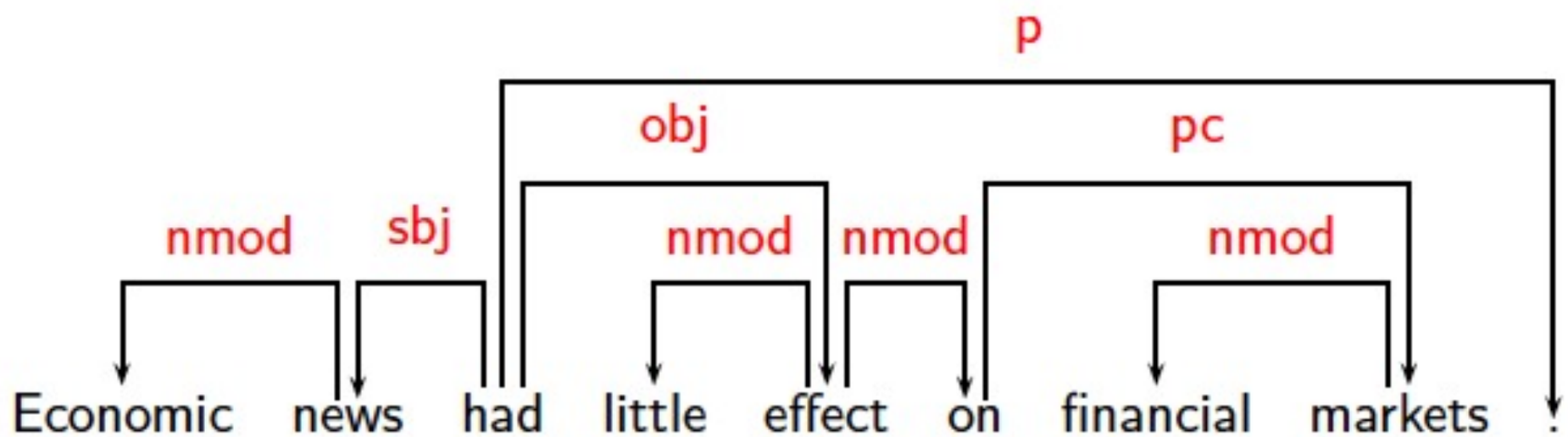
Notationelle Varianten



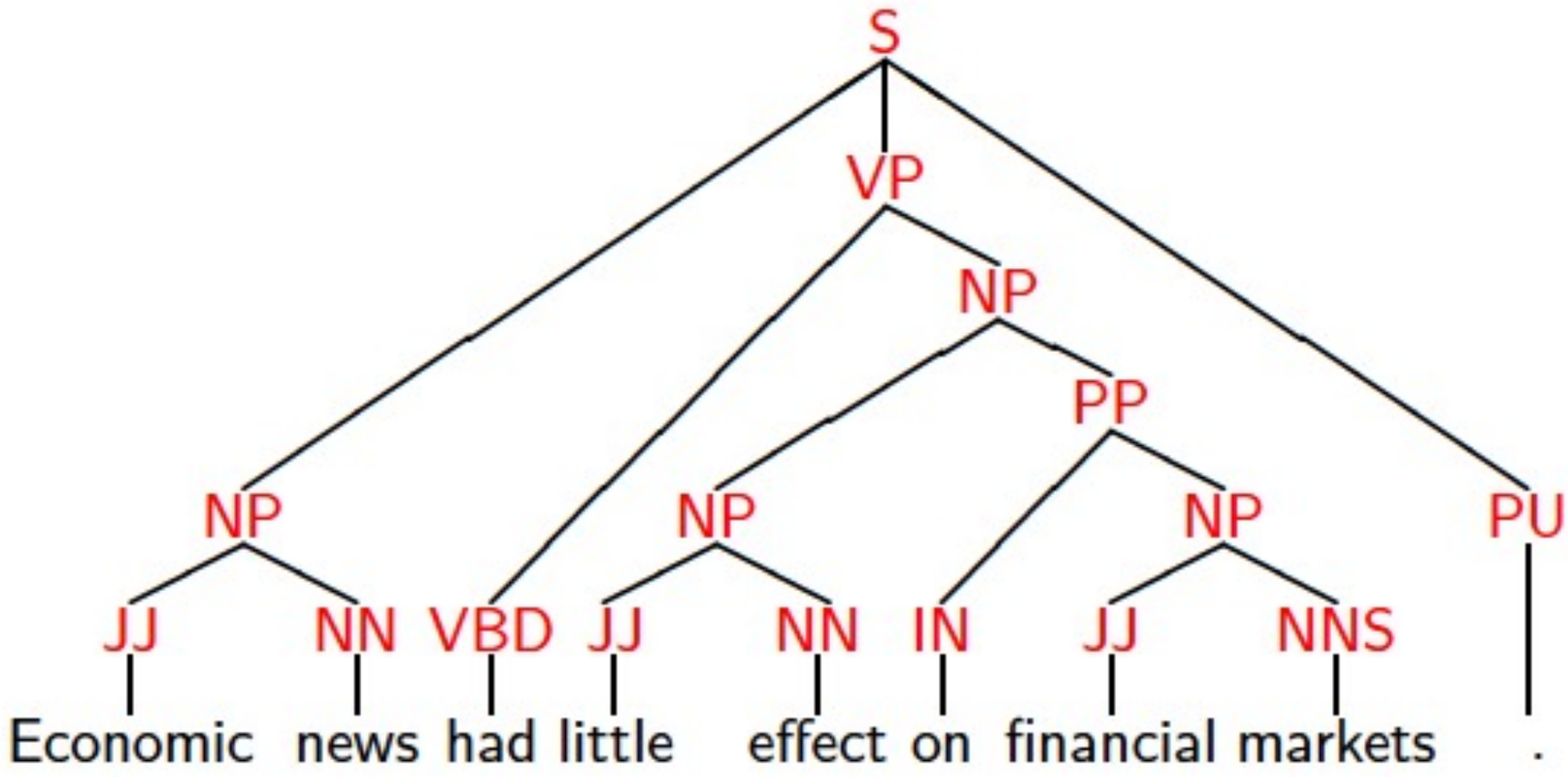
Notationelle Varianten



Notationelle Varianten



Phrasenstruktur



Vergleich

- Dependenzstrukturen repräsentieren explizit:
  - Kopf-Dependent-Beziehungen (gerichtete Kanten)
  - funkionale Kategorien (Kantenmarkierungen)
  - eventuell einige strukturelle Kategorien (Wortart)
- Phrasenstrukturen repräsentieren explizit:
  - Phrasen (nichtterminale Knoten)
  - strukturelle Kategorien (nichtterminale Markierungen)
  - eventuell einige funktionale Kategorien (grammatische Funktionen)
- Hybride Repräsentationen können alle Elemente kombinieren

Einige theoretische 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]

Beispiel: TDG

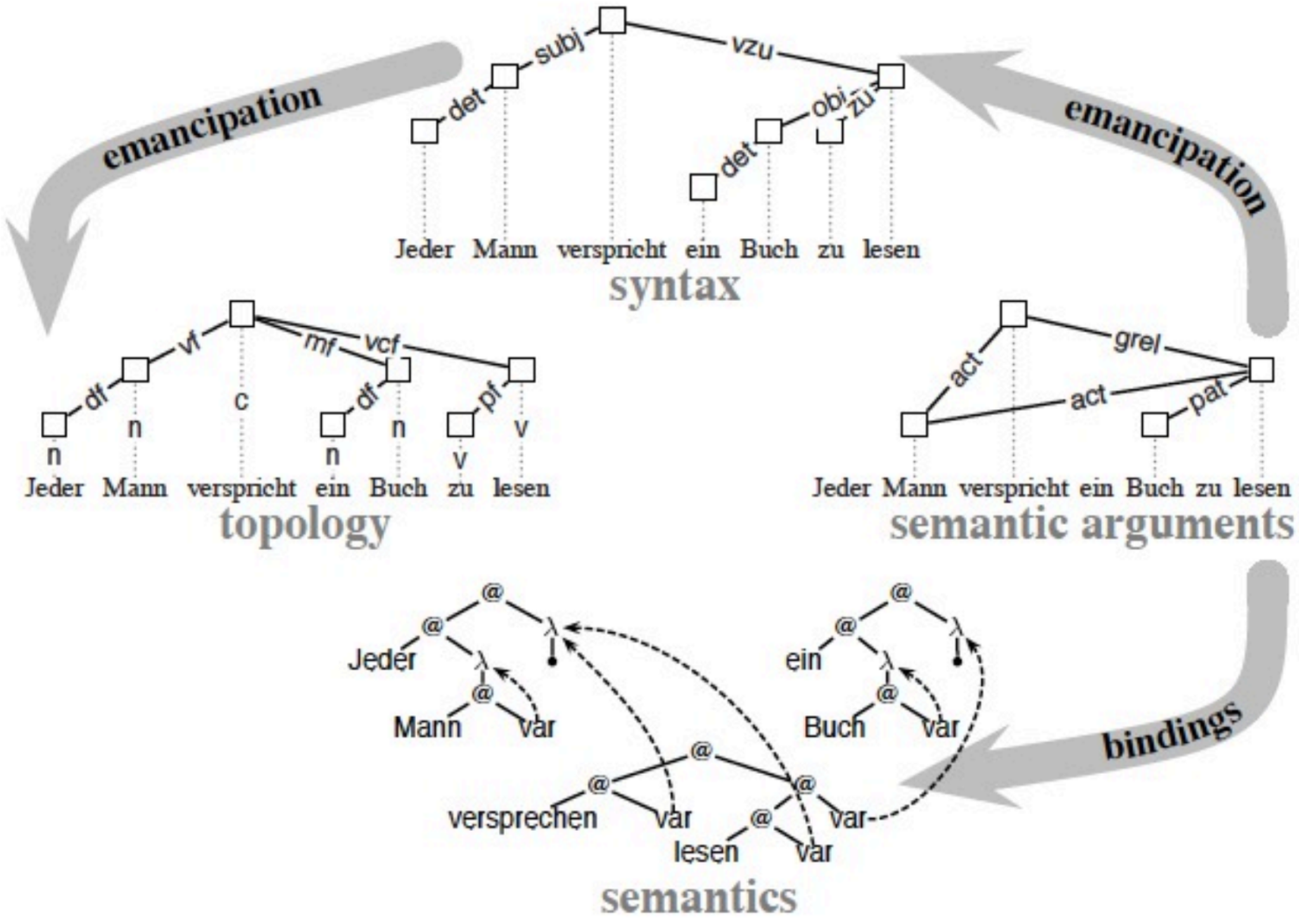


Figure 1: An architectural overview of TDG

Einige theoretische Aspekte

- Sind Dependenzstrukturen ausreichend und notwendig ?
- Mono-stratale oder multi-stratale syntaktische Repräsentationen ?
- Was ist die Natur der lexikalischen Elemente (Knoten) ?
    - Morpheme ? Wortformen ? Multi-Wort Einheiten ?
- Was ist die Natur der Dependenztypen (Kantenlabels) ?
    - Grammatische Funktionen ? Semantische Rollen ?
- Was sind die Kriterien zur Bestimmung von Köpfen und Dependenten ?
- Was sind die formalen Eigenschaftten von Dependenzen ?

Einige theoretische Aspekte

- Sind Dependenzstrukturen ausreichend und notwendig ?
- Mono-stratale oder multi-stratale syntaktische Repräsentationen ?
- Was ist die Natur der lexikalischen Elemente (Knoten) ?
  - Morpheme ? Wortformen ? Multi-Wort Einheiten ?
- Was ist die Natur der Dependenztypen (Kantenlabels) ?
  - Grammatische Funktionen ? Semantische Rollen ?
- Was sind die Kriterien zur Bestimmung von Köpfen und Dependenten ?
- Was sind die formalen Eigenschaftten von Dependenzen ?

Kriterien für Kopf und Dependent

- Kritierien für eine syntaktische Beziehung zwischen einem Kopf H und einem Dependenten D in einer Konstruktion C [Zwicky 1985, Hudson 1990]:

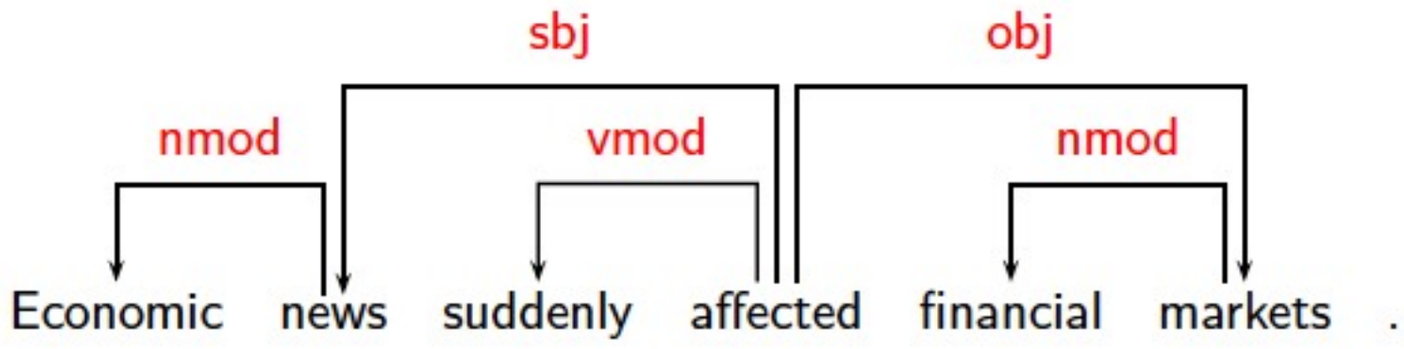
    1. H bestimmt die syntaktische Kategorie von C; H kann C ersetzen .
    2. H bestimmt die semantische Kategorie von C; D spezifiziert H .
    3. H ist obligatorisch; D kann optional sein .
    4. H selektiert D und bestimmt, ob D obligatorisch ist .
    5. Die Form von D hängt von H ab (Agreement oder Subkategorisierung) .
    6. Die lineare Position von D wird mit bezug auf H spezifiziert .

- Aspekte:

    - Syntaktische (und morphologische) versus semantische Kriterien
    - Exozentrische versus endozentrische Konstruktionen (Head-Modifier-Konstruktionen)

Einige klare Fälle

Konstruktion	**Kopf**	**Dependent**
Exozentrisch	Verb	Subject (sbj)
	Verb	Object (obj)
Endozentrisch	Verb	Adverbial (vmod)
	Noun	Attribute (nmod)

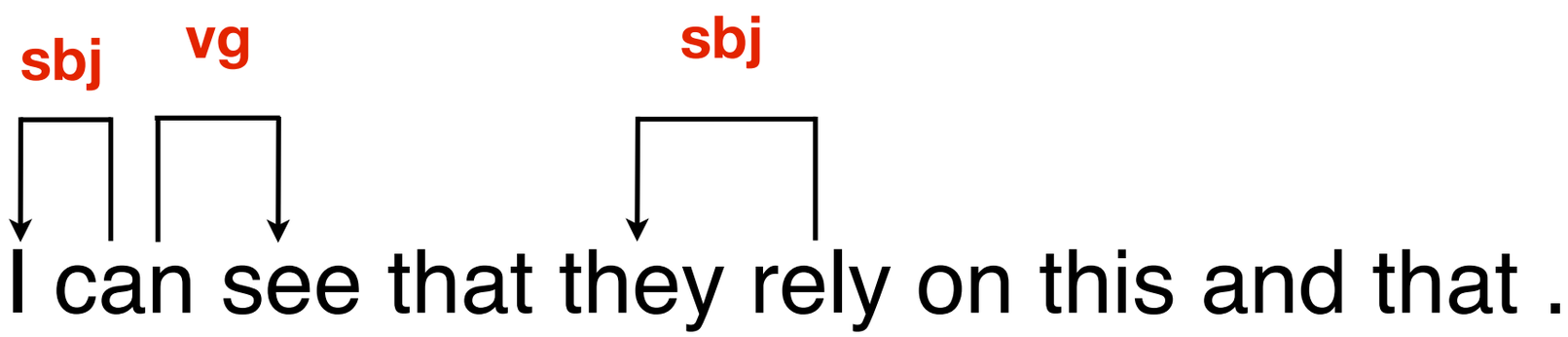


Einige Trickreiche Fälle

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

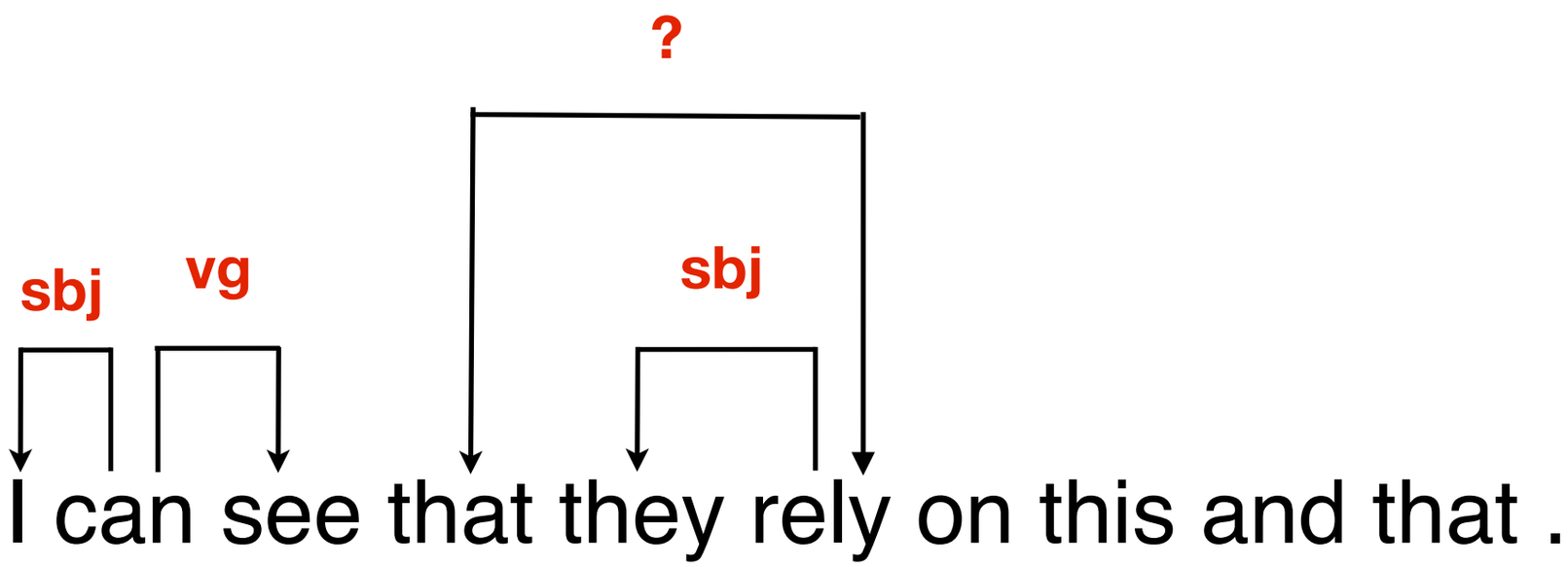
Einige Trickreiche Fälle

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



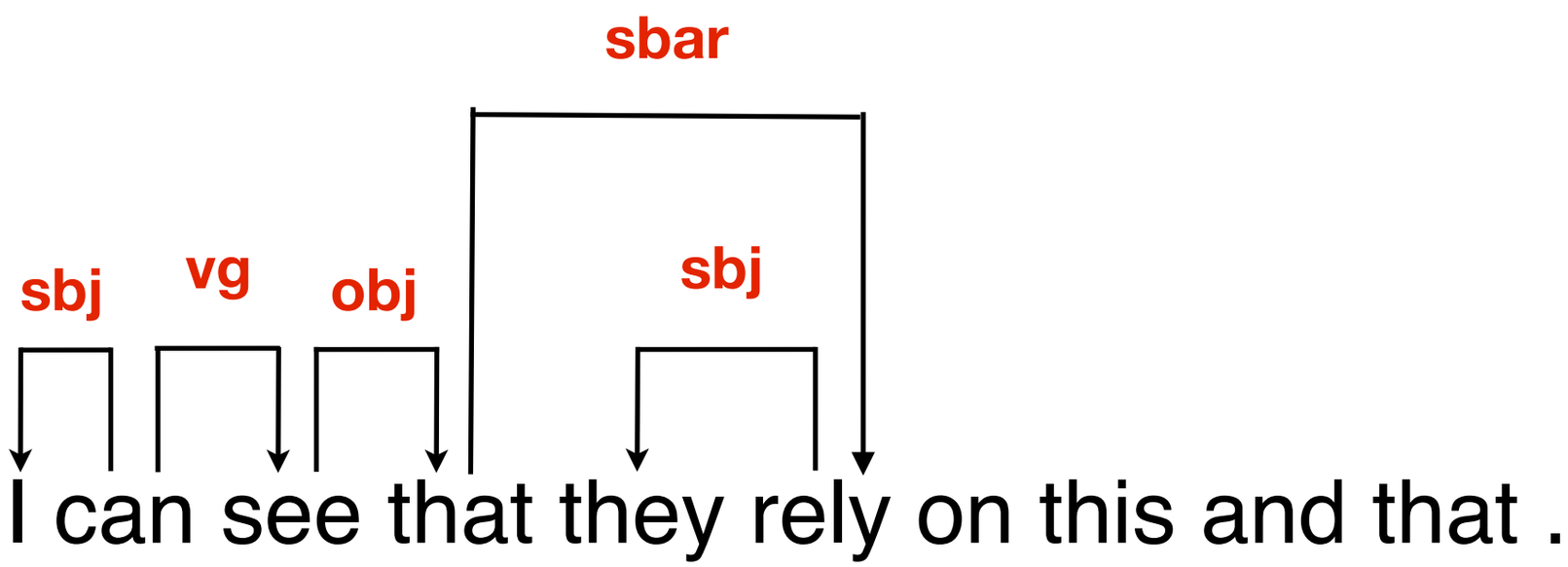
Einige Trickreiche Fälle

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



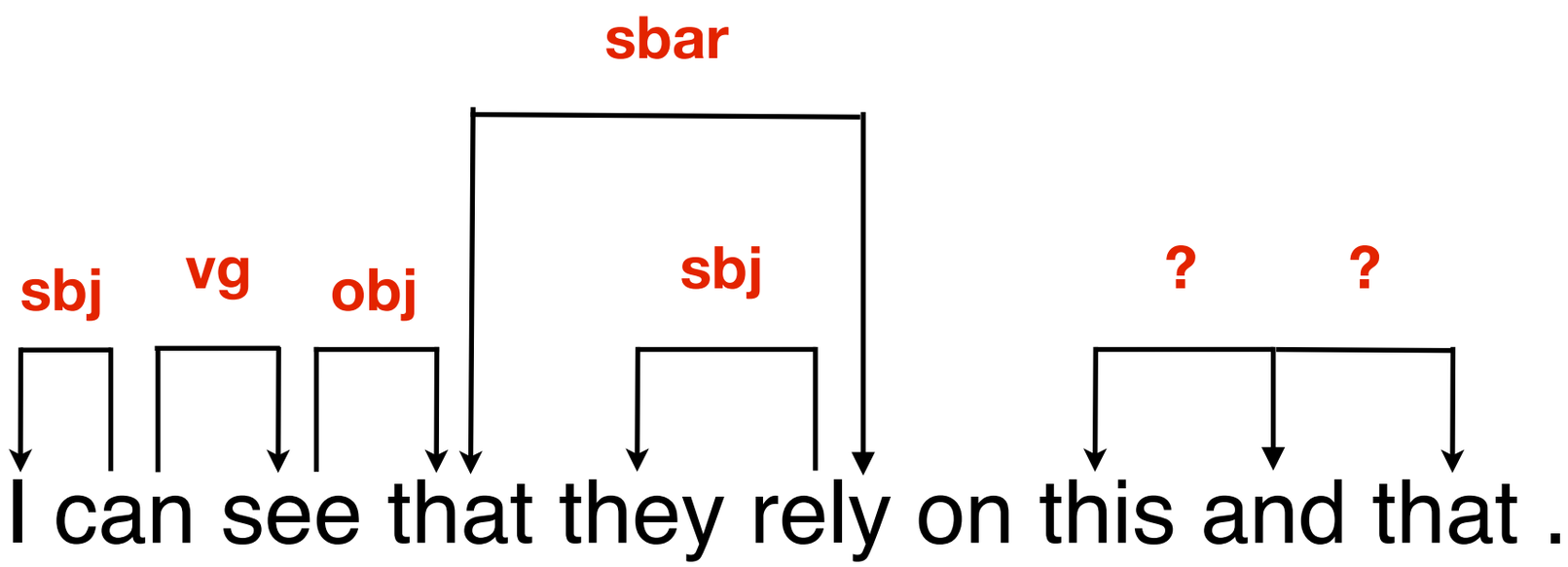
Einige Trickreiche Fälle

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



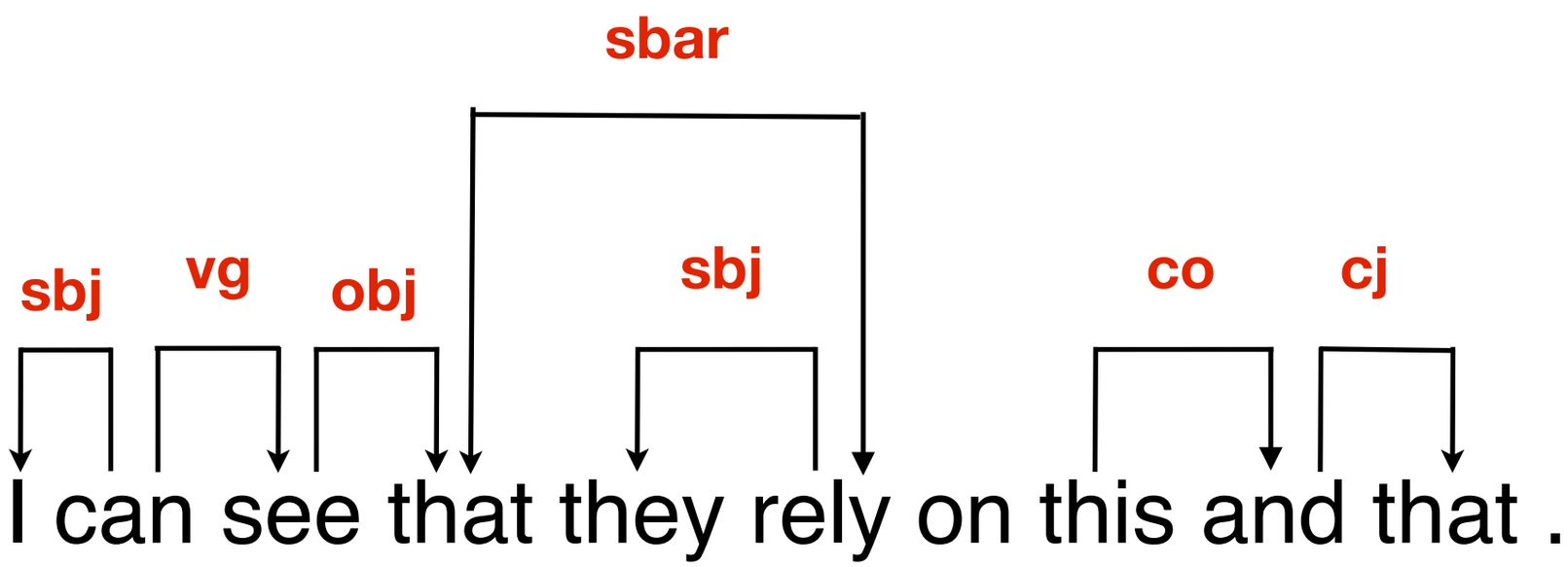
Einige Trickreiche Fälle

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



Einige Trickreiche Fälle

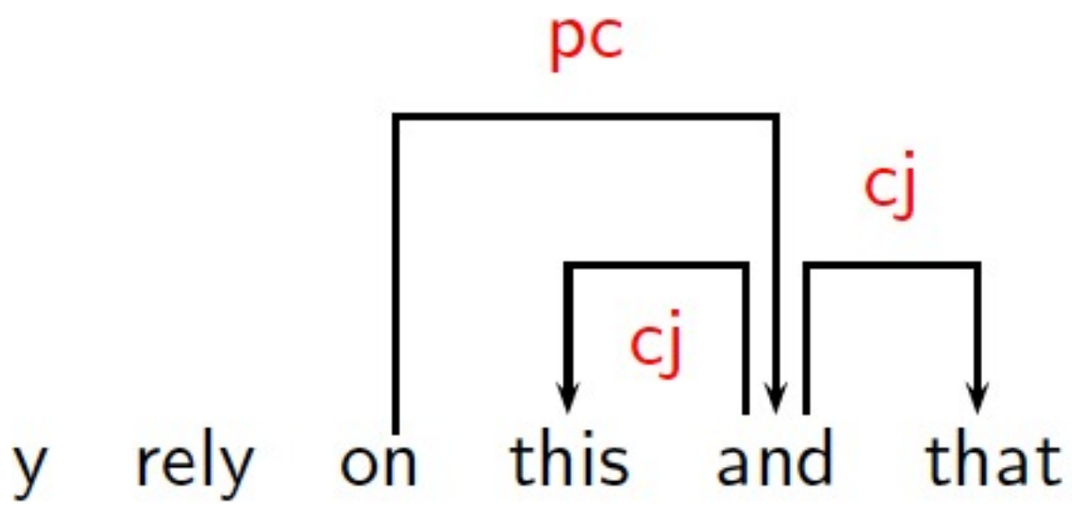
- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation

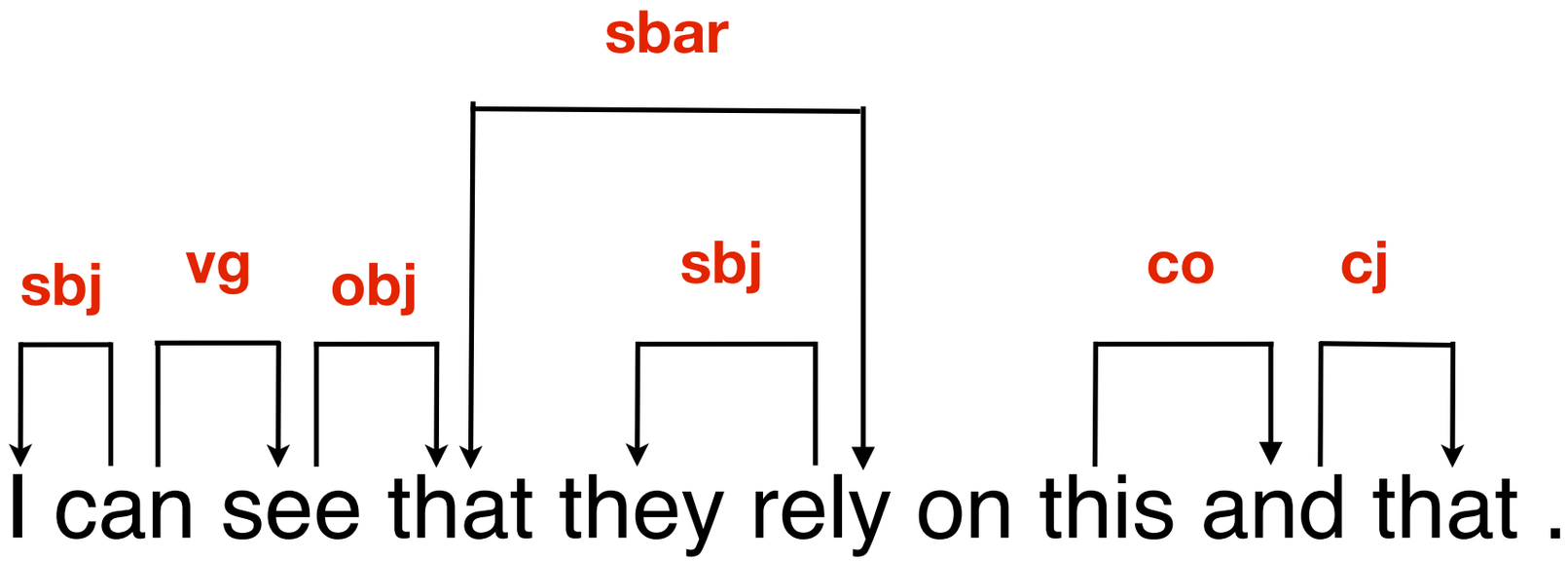


Einige Trickreiche Fälle

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

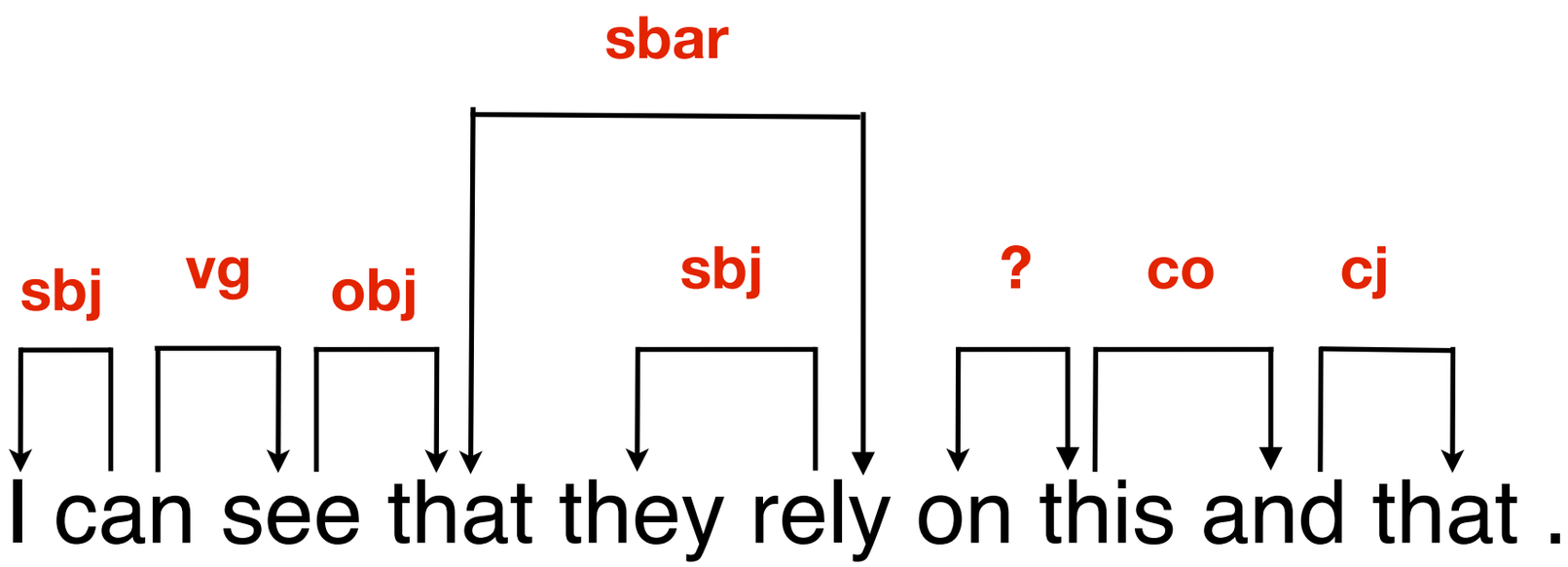
Alternative Darstellung





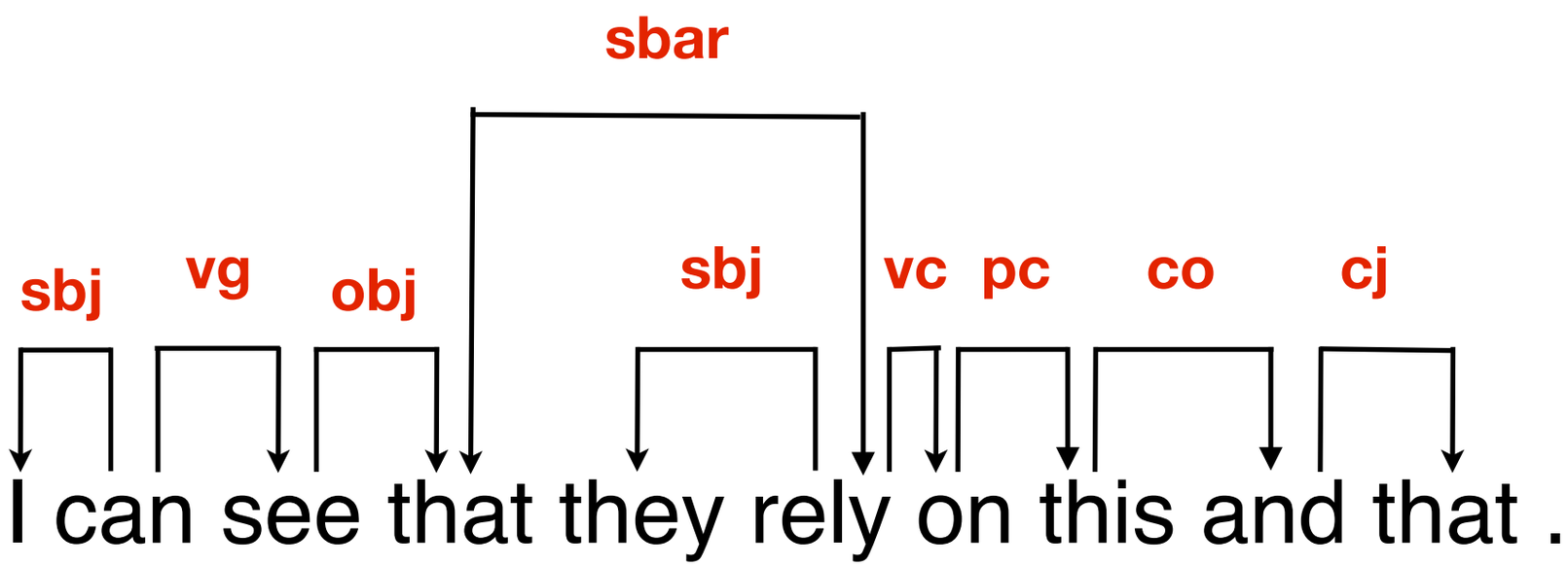
Einige Trickreiche Fälle

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



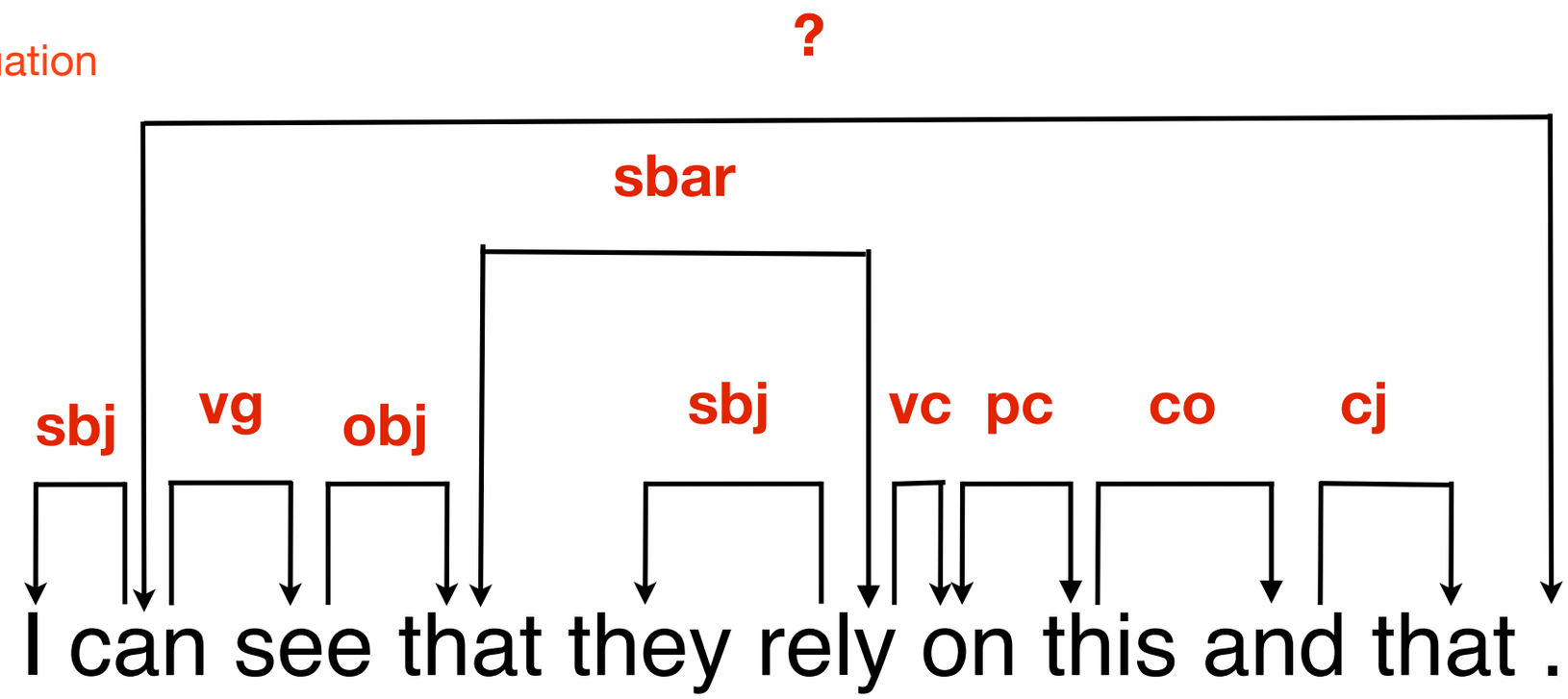
Einige Trickreiche Fälle

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



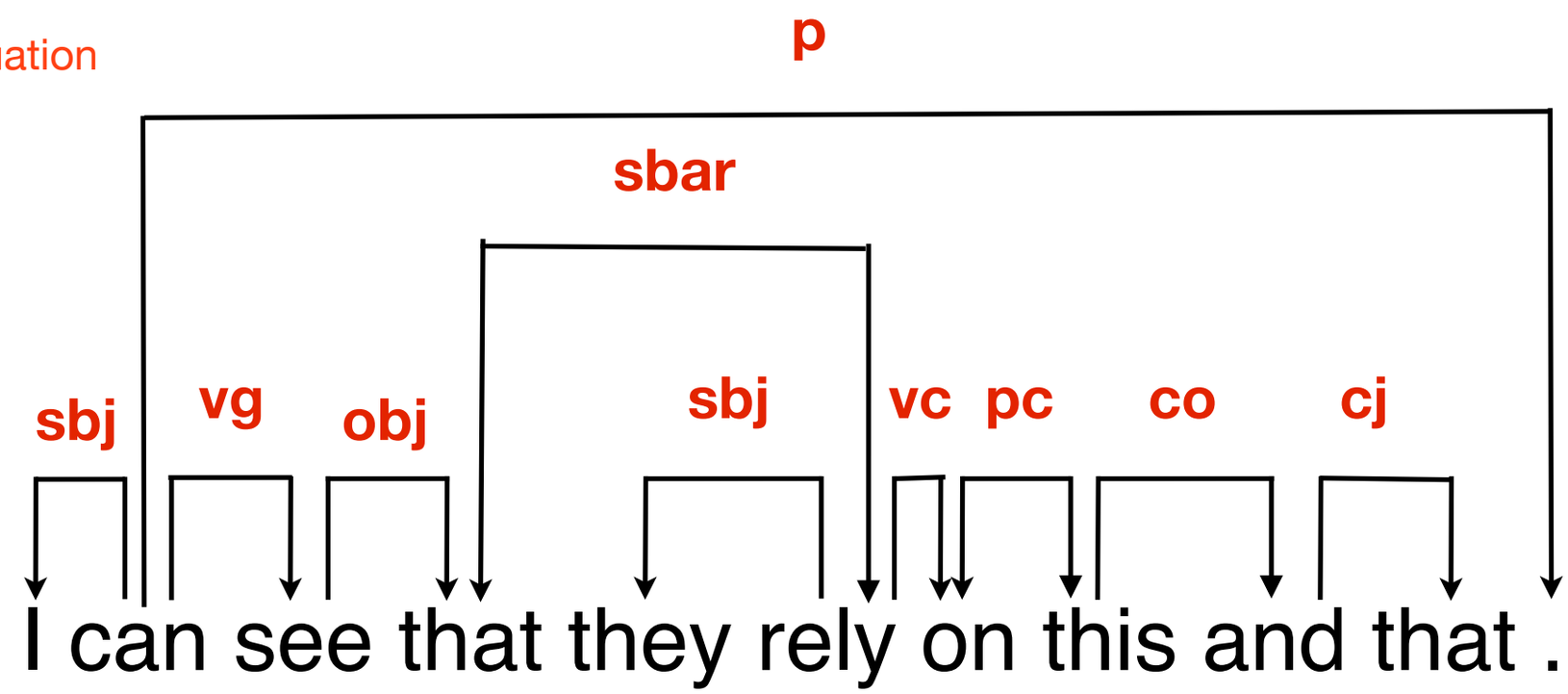
Einige Trickreiche Fälle

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



Einige Trickreiche Fälle

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



Valenz und Grammatikalität

- Ein wichtiges Konzept in vielen Varianten der DG ist das der **Valenz** = die Fähigkeit eines Wortes Argumente zu nehmen

- Ein Lexikon kann bsp. so aussehen [Hajič et al.(2003)]:

	$Slot_1$	$Slot_2$	$Slot_3$
$sink_1$	ACT(nom)	PAT(acc)	
$sink_2$	PAT(nom)		
give	ACT(nom)	PAT(acc)	ADDR(dat)

- Um Grammatikalität zu überprüfen (grob) ...

  - Wörter haben Valenzbedingungen, die überprüft werden müssen

  - Wende generelle Regeln auf die Valenzen an, um zu sehen, ob ein Satz valide ist.

Beispiel: Subkategorisierung

- „Weil Scherben nur für Fachleute lesbar sind, ergänzte sie die Fundstücke mit Zeichnungen.“
    - „lesbar<(Subj: NP-Nom) (Obl-für: P(für)>“ (vgl. Rohrer und Heid, 2002)
- Verb „schreiben“:
    - (:np . :nom) (:np . :akk) (:pp :akk "an")
    - (:np . :nom) (:np . :akk)
    - (:np . :nom) (:np . :dat) (:np . :akk)
    - (:np . :nom) (:np . :dat) (:subclause . "dass")

Dependenzgraphen

- Eine Dependenzstruktur kann definiert werden als gerichteter Graph G bestehend aus:
  - einer Menge V von Knoten (Eckpunkte),
  - einer Menge A von Bögen (gerichtete Kanten)
  - einer lineare Ordnung < über V (Wortordnung)
- Markierte Graphen:
  - Knoten in V sind mit Wortformen (und Annotationen) markiert
  - Kanten in A sind mit Dependenztypen markiert:
    - $L = \{l_1, \ldots, l_{|L|}\}$ ist die Menge der erlaubten Kantenlabels.
    - Jede Kante in A ist ein Tripel (i, j, k) und repräsentiert eine Dependenz von w_i zu w_j mit Label l_k .

Dependenzgraphen Notation

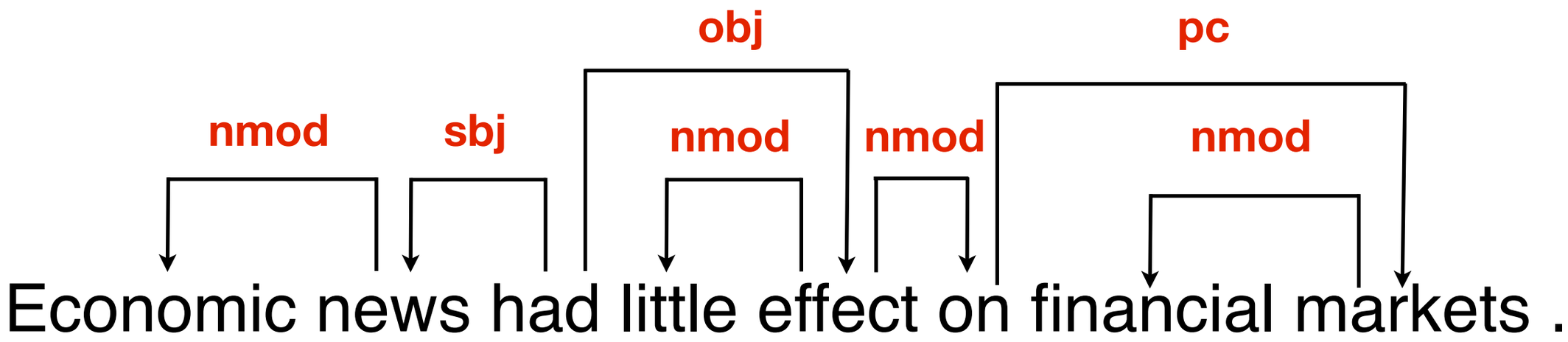
- Für einen Dependenzgraphen $G = (V, A)$
- Mit der Labelmenge $L = \{l_1, \ldots, l_{|L|}\}$
  - $i \rightarrow j \equiv \exists\, k : (i, j, k) \in A$
  - $i \leftrightarrow j \equiv i \rightarrow j \vee j \rightarrow i$
  - $i \rightarrow^* j \equiv i = j \vee \exists\, i' : i \rightarrow i',\ i' \rightarrow^* j$
  - $i \leftrightarrow^* j \equiv i = j \vee \exists\, i' : i \leftrightarrow i',\ i' \leftrightarrow^* j$

Formale Bedingungen von Dependenzgraphen

- G ist (schwach) zusammenhängend:
  - Wenn $i, j \in V$, $i \leftrightarrow^* j$.
- G ist azyklisch:
  - Wenn $i \rightarrow j$, dann nicht $j \rightarrow^* i$.
- G gehorcht der Einzel-Kopf Bedingung
  - Wenn $i \rightarrow j$, dann nicht $i' \rightarrow j$, für jedes $i' \neq i$.
- G ist projektiv
  - Wenn $i \rightarrow j$, dann $i \rightarrow^* i'$, für jedes i' sodass $i < i' < j$ oder $j < i' < i$.

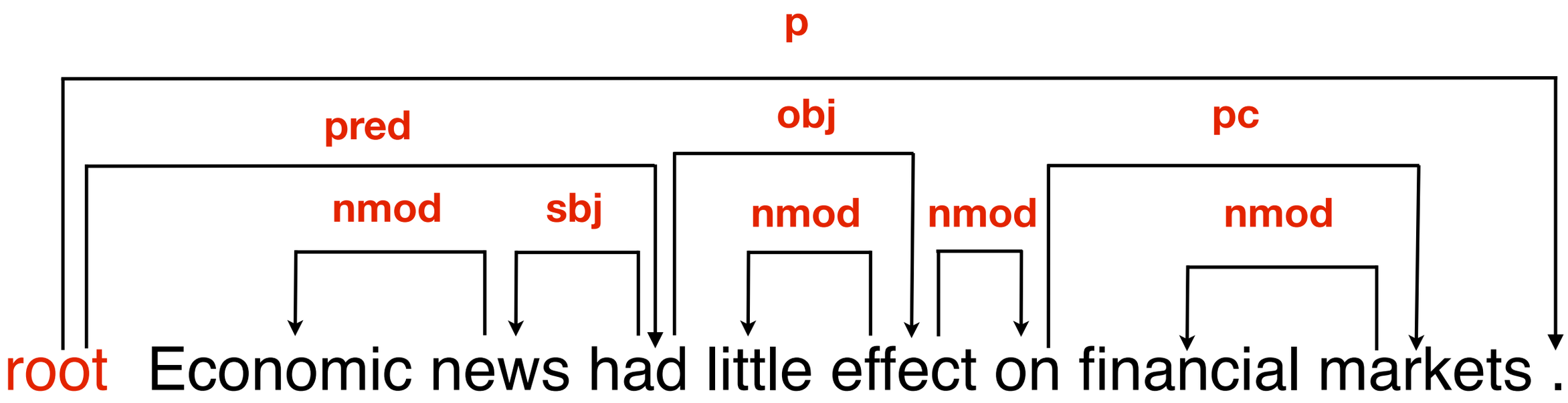
Zusammenhängend, Azyklisch, Einzel-Kopf

- Intuition
  - Syntaktische Struktur ist vollständig (zusammenhängend)
  - Syntaktische Struktur ist hierarchisch (azyklisch)
  - Jedes Wort hat maximal einen Kopf (Einzel-Kopf)
- Zusammenhang kann durch Hinzufügen eines speziellen Wurzel-Knotens erzwungen werden



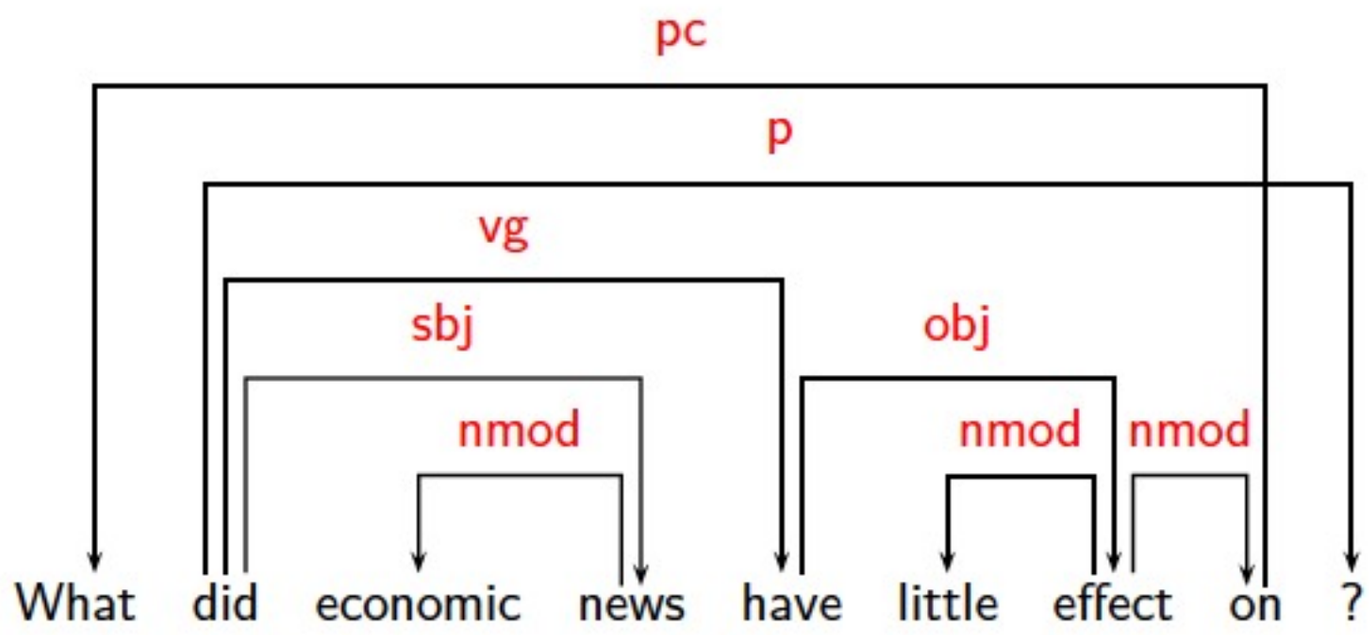
Zusammenhängend, Azyklisch, Einzel-Kopf

- Intuition
- Syntaktische Struktur ist vollständig (zusammenhängend)
- Syntaktische Struktur ist hierarchisch (azyklisch)
- Jeds Wort hat maximal einen Kopf (Einzel-Kopf)
- Zusammenhang kann durch hinzufügen eines speziellen Wurzel-Knotens erzwungen werden



Projektivität

- Die meisten theoretischen Frameworks nehmen Projektivität nicht an.
- Nicht-projektive Strukturen werden benötigt zur Betrachtung von
  - Fernabhängigkeiten
  - Freie Wortstellung



Projektivität (cf. Kuhlmann, PhD, 2007)

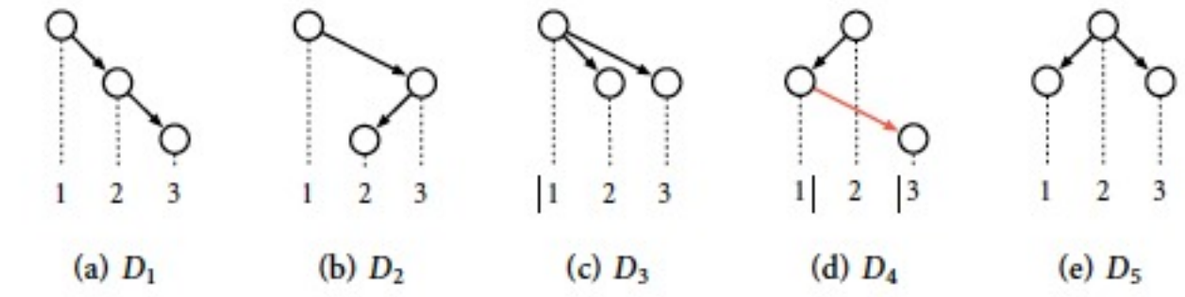


Figure 3.1: Five (of nine) dependency structures with three nodes

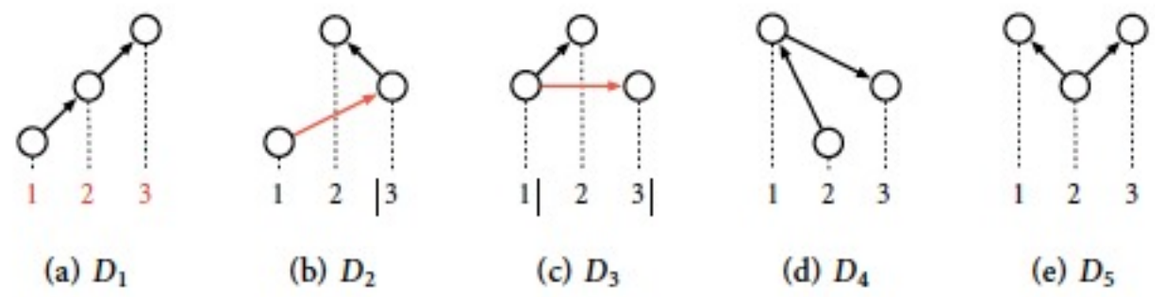
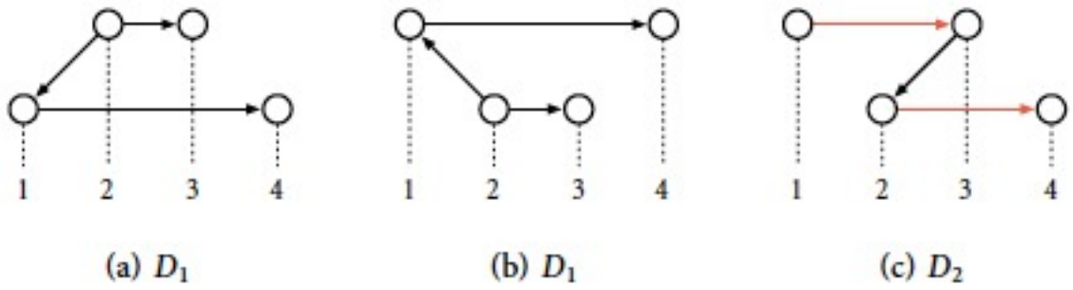


Figure 3.2: Alternative pictures for the dependency structures from Figure 3.1

Schwache nicht-projektive Dependenzstrukturen



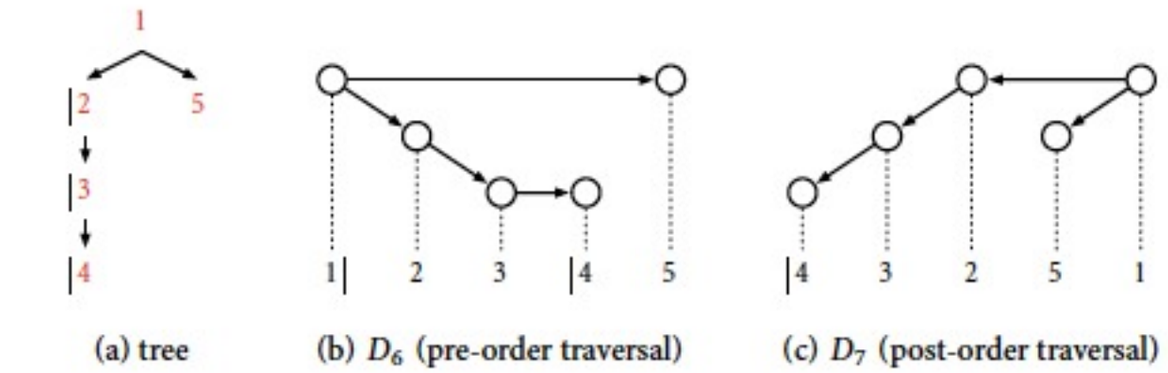
(a) D_1 (b) D_1 (c) D_2

Figure 5.1: Three pictures of non-projective dependency structures. The structure shown in pictures (a) and (b) is weakly non-projective; the structure shown in picture (c) contains overlapping edges.

(a) D_1 (b) D_2

Figure 5.2: Alternative pictures for the dependency structures from Figure 5.1

Linearisierung



(a) tree (b) D_6 (pre-order traversal) (c) D_7 (post-order traversal)

Figure 3.4: Dependency structures obtained by tree traversals of a children-ordered tree

Dependenzparsing

- Das Problem
  - Eingabe: Satz x = $w_0, w_1, \ldots, w_n$ with w_0 = root
  - Ausgabe: Dependenzgraph G = (V, A) für x wobei:
    - V = {0, 1, . . . , n} die Knotenmenge ist
    - A ist die Kantenmenge, d.h., $(i, j, k) \in A$ repräsentiert eine Dependenz von w_i zu w_j mit dem Label $l_k \in L$

- Zwei Hauptansätze
  - Grammatikbasiertes Parsing
    - Kontextfreie Dependenzgrammatik
    - Constraint-Dependenzgrammatik
  - Daten-gesteuertes Parsing
    - Übergangsbasierte Modelle
    - Graphbasierte Modelle

Kontextfreie Dependenzgrammatik

- Dependenzgrammatik als lexikalisierte kontextfreie Grammatik
  - $H \rightarrow L_1 \ldots L_m \; h \; R_1 \ldots R_n$
  - $H \in V_N;\; h \in V_T;\; L_1 \ldots L_m,\; R_1 \ldots R_n \in V^*_N$
- Standard kontextfreie Parsingalgorithmen (CKY, Early)

- Nur projektive, unmarkierte Dependenzbäume
- Schwach äquivalent zu (beliebigen) kontextfreien Grammatiken [Hays 1964, Gaifman 1965]
- Aktuelle Entwicklungen
  - Link Grammar [Sleator and Temperley 1991]
  - Early-basierter Parser mit left-corner Filterung [Lombardo and Lesmo 1996]
  - Bilexikalische Grammatiken [Eisner 1996, Eisner 2000]

Constraint-Dependenzgrammatik

- Parsing als Constraint-Erfüllung [Maruyama 1990]:
  - Grammatik besteht aus einer Menge von booleaschen Constraints, d.s. logischen Formeln, die die Wohlgeformtheit von Dependenzgraphen bschreiben.
  - Constraint-Propagation entfernt Kandidaten-Graphen, die die Constraints verletzen (eliminierendes Parsing)
- Behandlung von nicht-projektiven Dependenzgraphen
- Parsing für den allgemeinen Fall nicht lösbar (NP-vollständig)
- Aktuelle Entwicklungen
  - 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]

Übergangsbasierte Modelle

- Kernidee
  - Definiere ein Übergangssystem (Zustandsmaschine) zur Abbildung eines Satzes auf seinen Dependenzgraphen.
  - Lernen: Induziere eine Modell zur Voraussage des nächsten Zustandsübergangs unter Berücksichtigung früherer Übergänge.
  - Parsing: Konstruiere die optimale Übergangssequenz unter Berücksichtigung des induzierten Modells.
- Charakterisierungen:
  - Lokales Trainieren eines Models für optimale Übergänge
  - Greedy Suche/Inferenz

Graphbasierte Modelle

- Kernidee
  - Definiere einen Suchraum für mögliche Dependenzgraphen für einen Satz.
  - Lernen: Induziere ein Modell zur Bewertung eines vollständigen Dependenzgraphen für einen Satz.
  - Parsing: Finde den am höchsten bewerteten Dependenzgraphen unter Berücksichtigung des induzierten Modells.
- Charakterisierungen:
  - Globales Lernen eines Modells für optiomale Dependenzgraphen
  - Umfassende Suche/Inferenz

Vor- und Nachteile von Dependenzparsing

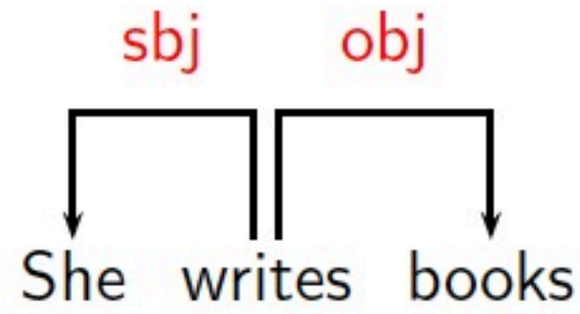
- Was sind die Vorteile der dependenzbasierten Methoden?
- Was ihre Nachteile?
- Vier Typen von Betrachtungen
  - Komplexität
  - Transparenz
  - Wortordnung
  - Ausdrucksstärke

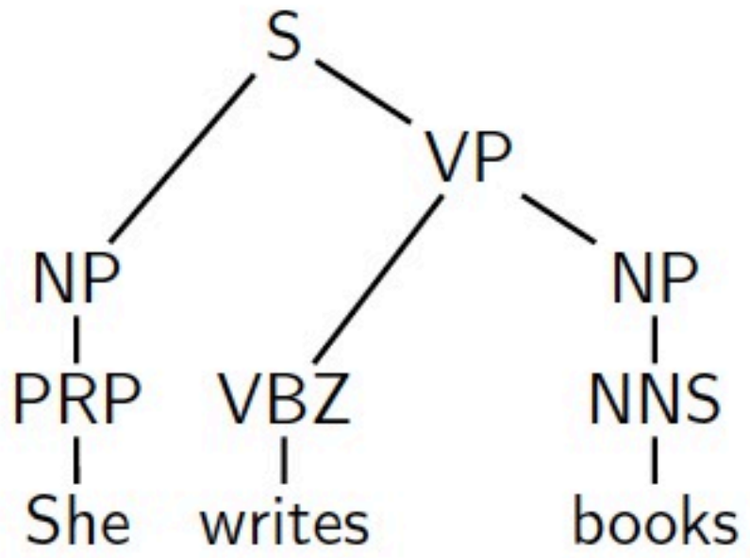
Komplexität

- Praktische Komplexität
  - Gegeben die Einzel-Head-Bedingung, dann kann das Parsing eines Satzes $x = w_1, ..., w_n$ reduziert werden auf das Markieren jeden Wortes w_i mit:
    - einem Kopf-Wort h_i
    - einem Dependenztypen d_i
- Theoretische Komplexität
  - Wenn man die besonderen Eigenschaften von Dependenzgraphen ausnutzt, ist es möglich, manchmal die Worst-Case Komplexität von konstitutentenbasierten Parsing zu verbessern
  - lexikalisches Parsing in $O(n^3)$ Zeit [Eisner 1996]

Transparenz

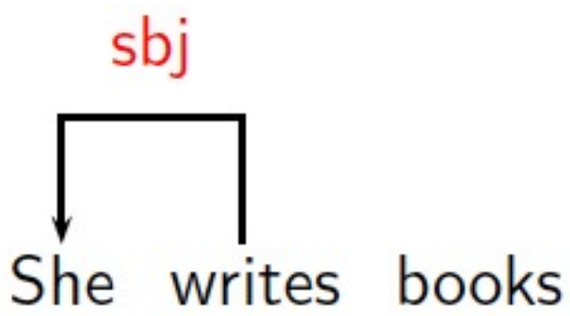
- Unmittelbare Kodierung von Prädikat-Argument-Strukturen





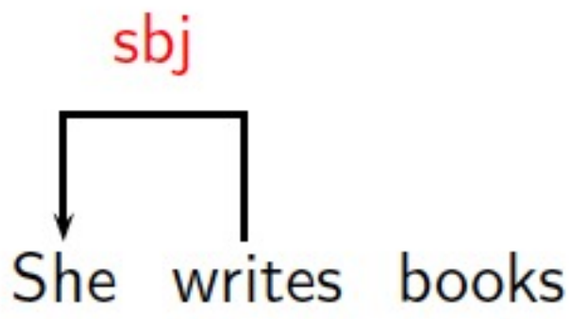
Transparenz

- Unmittelbare Kodierung von Predikat-Argument-Strukturen
- Fragmente können direkt interpretiert werden



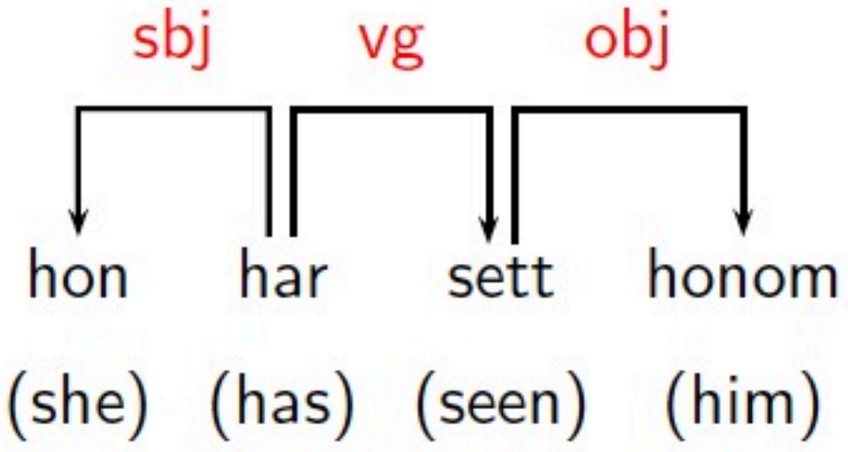
Transparenz

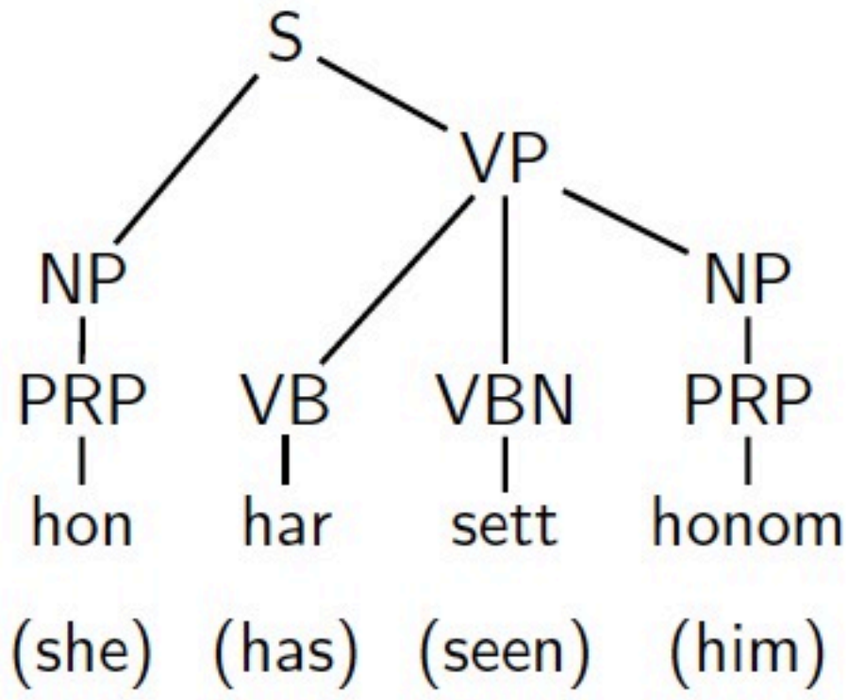
- Unmittelbare Kodierung von Predikat-Argument-Strukturen
- Fragmente können direkt interpretiert werden
- Aber sinnvoll nur mit markierten Dependenzgraphen



Wortordnung

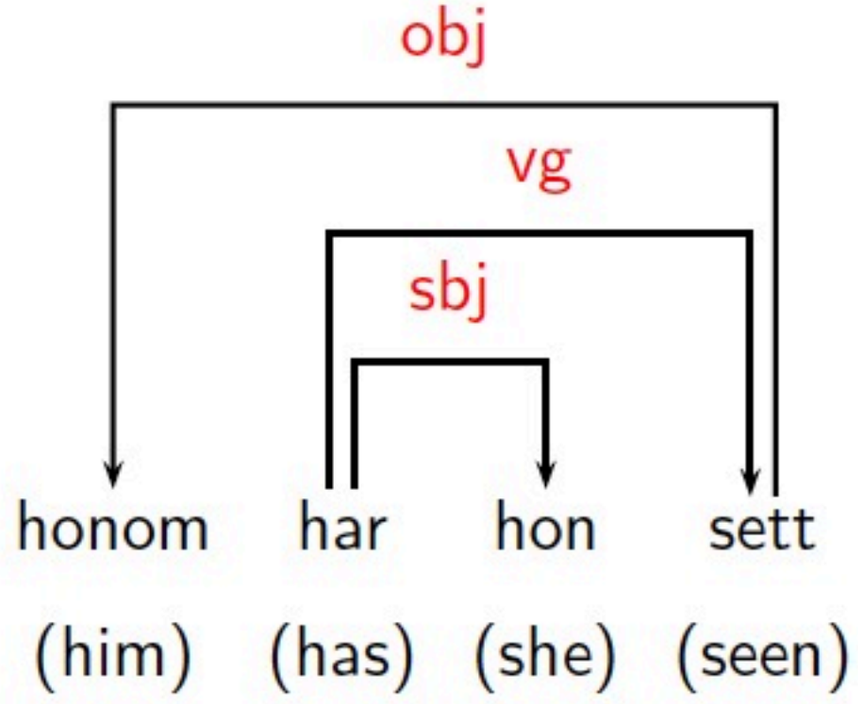
- Dependenzstrukturen sind unabhängig von der Wortordnung
- Daher geeignet für Sprachen mit freier Wortstellung

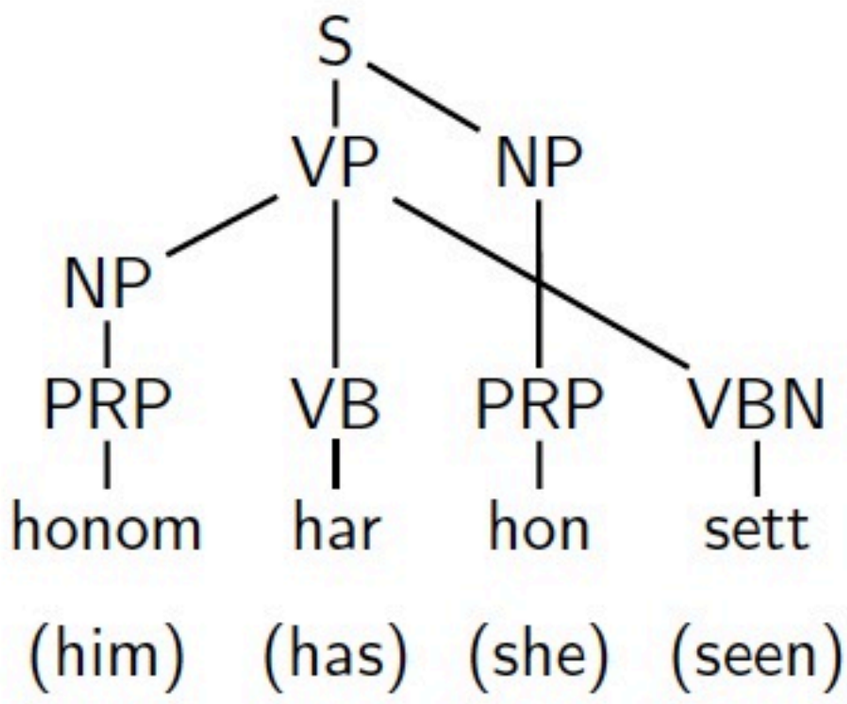




Wortordnung

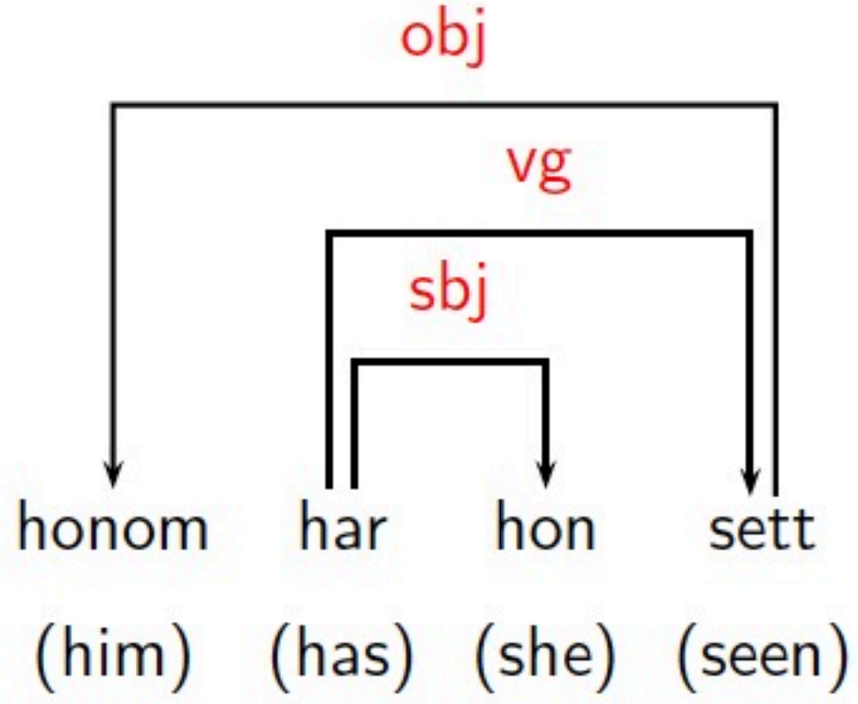
- Dependenzstrukturen sind unabhängig von der Wortordnung
- Daher geeignet für Sprachen mit freier Wortstellung

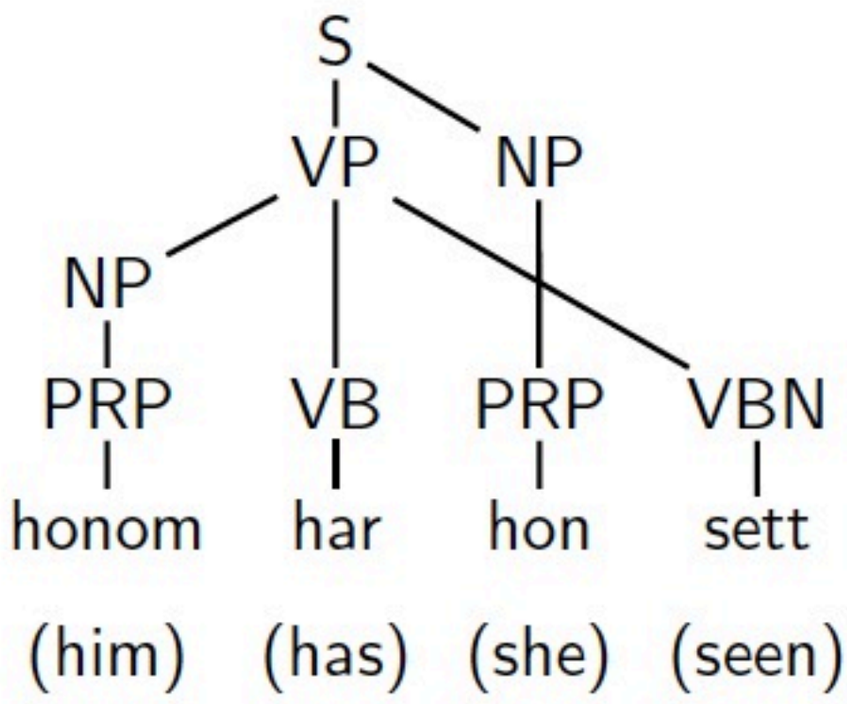




Wortordnung

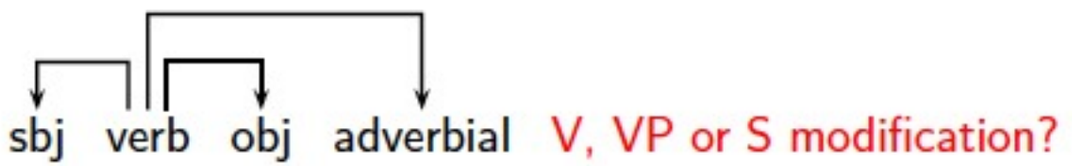
- Dependenzstrukturen sind unabhängig von der Wortordnung
- Daher geeignet für Sprachen mit freier Wortstellung
- Aber nur mit nicht-projektiven Dependenzgraphen





Ausdruckskraft

- Eingeschränkte Ausdruckskraft
  - Jede projektive Dependenzgrammatik hat eine stark äquivalente kontextfreie Grammatik, aber nicht vice versa [Gaifman 1965].
  - Unmöglich, phrasale Modifikation und Kopf-Modifikation in unmarkierten Dependenzstrukturen zu unterscheiden [Mel'čuk 1988].



- Was ist mit markierten nicht-projektierten Dependenzstrukturen?

Verfügbare Dependenzparser

- Stanford Parser, Uni. Stanford (Manning et al.)
  [http://nlp.stanford.edu/software/lex-parser.shtml]
    - PCFG, Transformation in Dependenzgraphen; volle Pipeline
- MaltParser, Uni. Uppsala (Nivre et al.)
  [http://www.maltparser.org/intro.html]
    - Übergangsbasierter Parser; keine volle Pipeline
- MSTParser, Uni. Penn (Baldrige und McDonald)
  http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html]
    - Graphbasierte Parser; keine volle Pipeline
- MDParser, DFKI LT lab (Volokh und Neumann)
  [http://mdparser.sb.dfki.de/]
    - Übergangsbasierter Parser; volle Pipeline

Performanz*

	UAS	LAS
Stanford Parser[6]	89.27	86.39
MST Parser	89.6	87.55
MaltParser(LibSVM)	92.1	90.4
MaltParser(LibLinear)	90.9	89.3
Ensemble	90.2	88.49
Mate-Tools	90.24	89.88
ClearParser	91.18	89.15
MDParser	89.7	87.7

Table 2: Attachment scores for dependency parse

	Accuracy
MiniPar	45/66
Stanford Parser	50/66
MaltParser	51/66
MDParser	50/66

Table 3: Parser Comparison for the PETE Development Data

	Accuracy
MaltParser	196/301
MDParser	197/301

Table 4: Parser Comparison for the PETE Test Data

MDParser	0.0008 seconds / sentence
MSTParser	0.268 seconds / sentence
MaltParser (LibSVM)	0.265 seconds / sentence
MaltParser (LibLinear)	0.0025 seconds / sentence
Ensemble	0.01 seconds / sentence
Mate Tools	0.077 seconds / sentence
Stanford Parser	0.37 seconds / sentence
ClearParser	0.0029 seconds / sentence

Table 8: Efficiency Evaluation

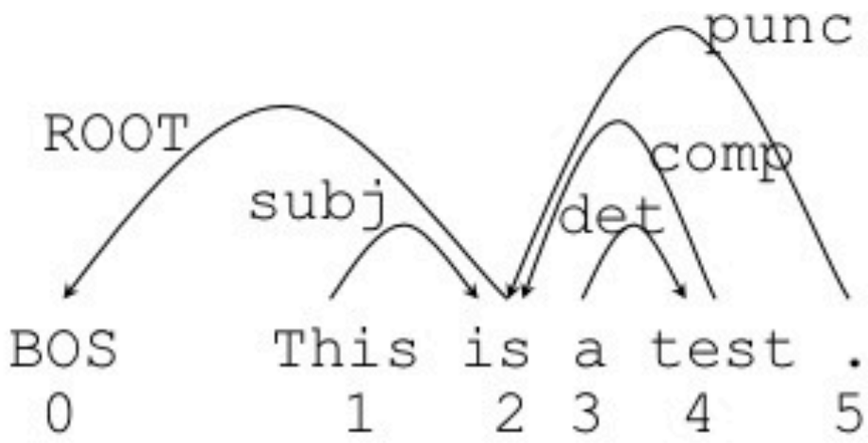
*Alexander Volokh, PhD, 2013

Wie werden Dependenzparser evaluiert ?

- CoNLL-X Shared Task: Multi-lingual Dependency Parsing, Tenth Conference on Computational Natural Language Learning - New York City, June 8-9, 2006. (Buchholz et al.)
- Idee:
  - Definiere ein uniformes Annotationsschema für Dependenzstrukturen
  - Entwickle entsprechende Treebanks
    - Training
    - Development
    - Gold-Standard
- Testen: Parser erhalten unmarkierte Sätze des Gold-Standards als Eingabe und berechnen eine „eigene“ Treebank. Diese wird mit der Gold-Standard-Treebank wortweise verglichen.

CoNLL Format

Data format



ID	FORM	LEMMA	CPOS TAG	POS TAG	FEATS	HEAD	DEPREL
1	This	this	pronoun	demon	sg	2	subj
2	is	be	v	v-fin	3|sg|pres	0	ROOT
3	a	a	art	art	indef	4	det
4	test	test	n	nc	sg	2	comp
5	.	.	punc	punc	_	2	punc

CoNLL Format - Details

Field number:	Field name:	Description:
1	ID	Token counter, starting at 1 for each new sentence.
2	FORM	Word form or punctuation symbol.
3	LEMMA	Lemma or stem (depending on particular data set) of word form, or an underscore if not available.
4	CPOSTAG	Coarse-grained part-of-speech tag, where tagset depends on the language.
5	POSTAG	Fine-grained part-of-speech tag, where the tagset depends on the language, or identical to the coarse-grained part-of-speech tag if not available.
6	FEATS	Unordered set of syntactic and/or morphological features (depending on the particular language), separated by a vertical bar (|), or an underscore if not available.
7	HEAD	Head of the current token, which is either a value of ID or zero ('0'). Note that depending on the original treebank annotation, there may be multiple tokens with an ID of zero.
8	DEPREL	Dependency relation to the HEAD. The set of dependency relations depends on the particular language. Note that depending on the original treebank annotation, the dependency relation may be meaningfull or simply 'ROOT'.
9	PHEAD	Projective head of current token, which is either a value of ID or zero ('0'), or an underscore if not available. Note that depending on the original treebank annotation, there may be multiple tokens an with ID of zero. The dependency structure resulting from the PHEAD column is guaranteed to be projective (but is not available for all languages), whereas the structures resulting from the HEAD column will be non-projective for some sentences of some languages (but is always available).
10	PDEPREL	Dependency relation to the PHEAD, or an underscore if not available. The set of dependency relations depends on the particular language. Note that depending on the original treebank annotation, the dependency relation may be meaningfull or simply 'ROOT'.

CoNLL - Treebanks

Treebanks used

- Czech: Prague Dependency Treebank (PDT)
- Arabic: Prague Arabic Dependency Treebank (PADT)
- Slovene: Slovene Dependency Treebank (SDT)
- Danish: Danish Dependency Treebank (DDT)
- Swedish: Talbanken05
- Turkish: Metu-Sabancı treebank

} Depen-dency format

- German: TIGER treebank
- Japanese: Japanese Verbmobil treebank
- Portuguese: The Bosque part of the Floresta sintá(c)tica
- Dutch: Alpino treebank
- Chinese: Sinica treebank

} Consti-tuents and functions

- Spanish: Cast3LB
- Bulgarian: BulTreeBank

} Constituents and some functions

CoNLL Format - Beispiel Holländische TreeBank

```
1   Cathy          Cathy     N      N      eigen|ev|neut                    2   su      _  _
2   zag            zie       V      V      trans|ovt|1of2of3|ev             0   ROOT    _  _
3   hen            hen       Pron   Pron   per|3|mv|datofacc                2   obj1    _  _
4   wild           wild      Adj    Adj    attr|stell|onverv                5   mod     _  _
5   zwaaien        zwaai     N      N      soort|mv|neut                    2   vc      _  _
6   .              .         Punc   Punc   punt                            5   punct   _  _

1   Ze             ze        Pron   Pron   per|3|evofmv|nom                 2   su      _  _
2   had            heb       V      V      trans|ovt|1of2of3|ev             0   ROOT    _  _
3   met            met       Prep   Prep   voor                             8   mod     _  _
4   haar           haar      Pron   Pron   bez|3|ev|neut|attr               5   det     _  _
5   moeder         moeder    N      N      soort|ev|neut                    3   obj1    _  _
6   kunnen         kan       V      V      hulp|ott|1of2of3|mv              2   vc      _  _
7   gaan           ga        V      V      hulp|inf                         6   vc      _  _
8   winkelen       winkel    V      V      intrans|inf                      11  cnj     _  _
9   ,              ,         Punc   Punc   komma                           8   punct   _  _
10  zwemmen        zwem      V      V      intrans|inf                      11  cnj     _  _
11  of             of        Conj   Conj   neven                            7   vc      _  _
12  terrassen      terras    N      N      soort|mv|neut                    11  cnj     _  _
13  .              .         Punc   Punc   punt                            12  punct   _  _
```

CoNLL Format - Beispiel Tiger Treebank

1	Zwei	_	CARD	CARD	_	2	NK	2	NK	
2	Themen	_	NN	NN	_	14	SB	14	SB	
3	,	_	$,	$,	_	2	PUNC	2	PUNC	
4	die	_	PRELS	PRELS	_	8	OA	8	OA	
5	Perot	_	NE	NE	_	8	SB	8	SB	
6	immer	_	ADV	ADV	_	7	MO	7	MO	
7	wieder	_	ADV	ADV	_	8	MO	8	MO	
8	anspricht		_	VVFIN	VVFIN	_	2	RC	2	RC
9	,	_	$,	$,	_	2	PUNC	2	PUNC	
10	Rezession		_	NN	NN	_	2	APP	2	APP
11	und	_	KON	KON	_	10	CD	10	CD	
12	Bürokratie		_	NN	NN	_	10	CJ	10	CJ
13	,	_	$,	$,	_	14	PUNC	14	PUNC	
14	machen	_	VVFIN	VVFIN	_	0	ROOT	0	ROOT	
15	ihnen	_	PPER	PPER	_	18	DA	18	DA	
16	besonders		_	ADV	ADV	_	18	MO	18	MO
17	zu	_	PTKZU	PTKZU	_	18	PM	18	PM	
18	schaffen		_	VVINF	VVINF	_	14	OC	14	OC
19	.	_	$.	$.	_	14	PUNC	14	PUNC	
1	Statt	_	KOUI	KOUI	_	4	CP	4	CP	
2	Details	_	NN	NN	_	4	OA	4	OA	
3	zu	_	PTKZU	PTKZU	_	4	PM	4	PM	
4	nennen	_	VVINF	VVINF	_	6	MO	6	MO	
5	,	_	$,	$,	_	6	PUNC	6	PUNC	
6	wiederholt		_	VVFIN	VVFIN	_	0	ROOT	0	ROOT
7	er	_	PPER	PPER	_	6	SB	6	SB	
8	unverdrossen		_	ADJD	ADJD	_	6	MO	6	MO
9	die	_	ART	ART	_	11	NK	11	NK	
10	``	_	$(	$(	_	11	PUNC	11	PUNC	
11	Erfolgsformel		_	NN	NN	_	6	OA	6	OA
12	''	_	$(	$(	_	6	PUNC	6	PUNC	
13	:	_	$.	$.	_	6	PUNC	6	PUNC	

Evaluation der Genauigkeit von Parsern

- Vergleich von „Gold-Standard“ Treebank mit durch Parser berechnete Treebank.

- Bewertung:

  - labelled attachment score (LAS)): Der Anteil von Wörtern, die verglichen mit der Gold-Treebank den korrekten Kopf und den korrekten Dependenztypen zugewiesen bekamen. (Interpunktion wird nicht betrachtet): LAS = (correct heads + correct types)/total words

  - unlabelled attachment score (UAS): Es wird nur die korrekte Zuweisung des Kopfes betrachtet: UAS = correct heads/total words

Ergebnisse: CoNLL 2007 - LAS

Team	Average	Arabic	Basque	Catalan	Chinese	Czech	English	Greek	Hungarian	Italian	Turkish
Nilsson	80.32(1)	76.52(1)	76.94(1)	88.70(1)	75.82(15)	77.98(3)	88.11(5)	74.65(2)	80.27(1)	84.40(1)	79.79(2)
Nakagawa	80.29(2)	75.08(2)	72.56(7)	87.90(3)	83.84(2)	80.19(1)	88.41(3)	76.31(1)	76.74(8)	83.61(3)	78.22(5)
Titov	79.90(3)	74.12(6)	75.49(3)	87.40(6)	82.14(7)	77.94(4)	88.39(4)	73.52(10)	77.94(4)	82.26(6)	79.81(1)
Sagae	79.90(4)	74.71(4)	74.64(6)	88.16(2)	84.69(1)	74.83(8)	89.01(2)	73.58(8)	79.53(2)	83.91(2)	75.91(10)
Hall, J.	79.80(5)*	74.75(3)	74.99(5)	87.74(4)	83.51(3)	77.22(6)	85.81(12)	74.21(6)	78.09(3)	82.48(5)	79.24(3)
Carreras	79.09(6)*	70.20(11)	75.75(2)	87.60(5)	80.86(10)	78.60(2)	89.61(1)	73.56(9)	75.42(9)	83.46(4)	75.85(11)
Attardi	78.27(7)	72.66(8)	69.48(12)	86.86(7)	81.50(8)	77.37(5)	85.85(10)	73.92(7)	76.81(7)	81.34(8)	76.87(7)
Chen	78.06(8)	74.65(5)	72.39(8)	86.66(8)	81.24(9)	73.69(10)	83.81(13)	74.42(3)	75.34(10)	82.04(7)	76.31(9)
Duan (1)	77.70(9)*	69.91(13)	71.26(9)	84.95(10)	82.58(6)	75.34(7)	85.83(11)	74.29(4)	77.06(5)	80.75(9)	75.03(12)
Hall, K.	76.91(10)*	73.40(7)	69.81(11)	82.38(14)	82.77(4)	72.27(12)	81.93(15)	74.21(5)	74.20(11)	80.69(10)	77.42(6)
Schiehlen	76.18(11)	70.08(12)	66.77(14)	85.75(9)	80.04(11)	73.86(9)	86.21(9)	72.29(12)	73.90(12)	80.46(11)	72.48(15)
Johansson	75.78(12)*	71.76(9)	75.08(4)	83.33(12)	76.30(14)	70.98(13)	80.29(17)	72.77(11)	71.31(13)	77.55(14)	78.46(4)
Mannem	74.54(13)*	71.55(10)	65.64(15)	84.47(11)	73.76(17)	70.68(14)	81.55(16)	71.69(13)	70.94(14)	78.67(13)	76.42(8)
Wu	73.02(14)*	66.16(14)	70.71(10)	81.44(15)	74.69(16)	66.72(16)	79.49(18)	70.63(14)	69.08(15)	78.79(12)	72.52(14)
Nguyen	72.53(15)*	63.58(16)	58.18(17)	83.23(13)	79.77(12)	72.54(11)	86.73(6)	70.42(15)	68.12(17)	75.06(16)	67.63(17)
Maes	70.66(16)*	65.12(15)	69.05(13)	79.21(16)	70.97(18)	67.38(15)	69.68(21)	68.59(16)	68.93(16)	73.63(18)	74.03(13)
Canisius	66.99(17)*	59.13(18)	63.17(16)	75.44(17)	70.45(19)	56.14(17)	77.27(19)	60.35(18)	64.31(19)	75.57(15)	68.09(16)
Jia	63.00(18)*	63.37(17)	57.61(18)	23.35(20)	76.36(13)	54.95(18)	82.93(14)	65.45(17)	66.61(18)	74.65(17)	64.68(18)
Zeman	54.87(19)	46.06(20)	50.61(20)	62.94(19)	54.49(20)	50.21(20)	53.59(22)	55.29(19)	55.24(20)	62.13(19)	58.10(19)
Marinov	54.55(20)*	54.00(19)	51.24(19)	69.42(18)	49.87(21)	53.47(19)	52.11(23)	54.33(20)	44.47(21)	59.75(20)	56.88(20)
Duan (2)	24.62(21)*				82.64(5)		86.69(7)		76.89(6)		
Nash	8.65(22)*						86.49(8)				
Shimizu	7.20(23)						72.02(20)				

Table 2: Labeled attachment score (LAS) for the multilingual track in the CoNLL 2007 shared task. Teams are denoted by the last name of their first member, with italics indicating that there is no corresponding paper in the proceedings. The number in parentheses next to each score gives the rank. A star next to a score in the Average column indicates a statistically significant difference with the next lower rank.

Zusammenfassung

- Dependenzgrammatik - Basiskonzepte
- Dependenzparsing - Hauptansätze
- Dependenz-Treebanks

Aufgaben

- Dependenzbaum erstellen
- Nichtprojetive Bäume umordnen, dass sie projektiv sind
- Bäume überprüfen
- falsche Parsetree korrigieren