Unlexikalisiertes Parsing

von Dan Klein & Christopher D. Manning

Sybille Reuter
Galina Mihailova

Gliederung

- Motivation
- Aufbau des Experiments
- Annotation
- Vertikale und Horizontale Markovisierung
- Was ist unlexikalisierte Grammatik?
- Externe vs. Interne Annotation
- Tag Splitting
- enthaltene Annotationen in der Treebank
- Head Annotation
- Ergebnisse
- Zusammenfassung

Christopher D. Manning

- Professor der Computerlinguistik an der Standford Universität, USA
- Bekannt geworden durch das Buch „Foundations of Statistical Natural Language Processing“

Dan Klein

- Professor der Informatik an der Berkley Universität,USA
- Ehemaliger Student von Christopher D. Manning

Motivation

- Unlexikalisierte PCFG ist kompakter, verständlicher, leichter zu implementieren als die meisten lexikalisierten Modelle
- Parsing Algorithmus ist einfacher, Laufzeit ist O (n^3)
- Genauer und besser als erste lexikalisierte Modelle (86,36% - LP/LR F_1)

Aufbau des Experiments (1)

- Corpus: Penn Treebank, WSJ

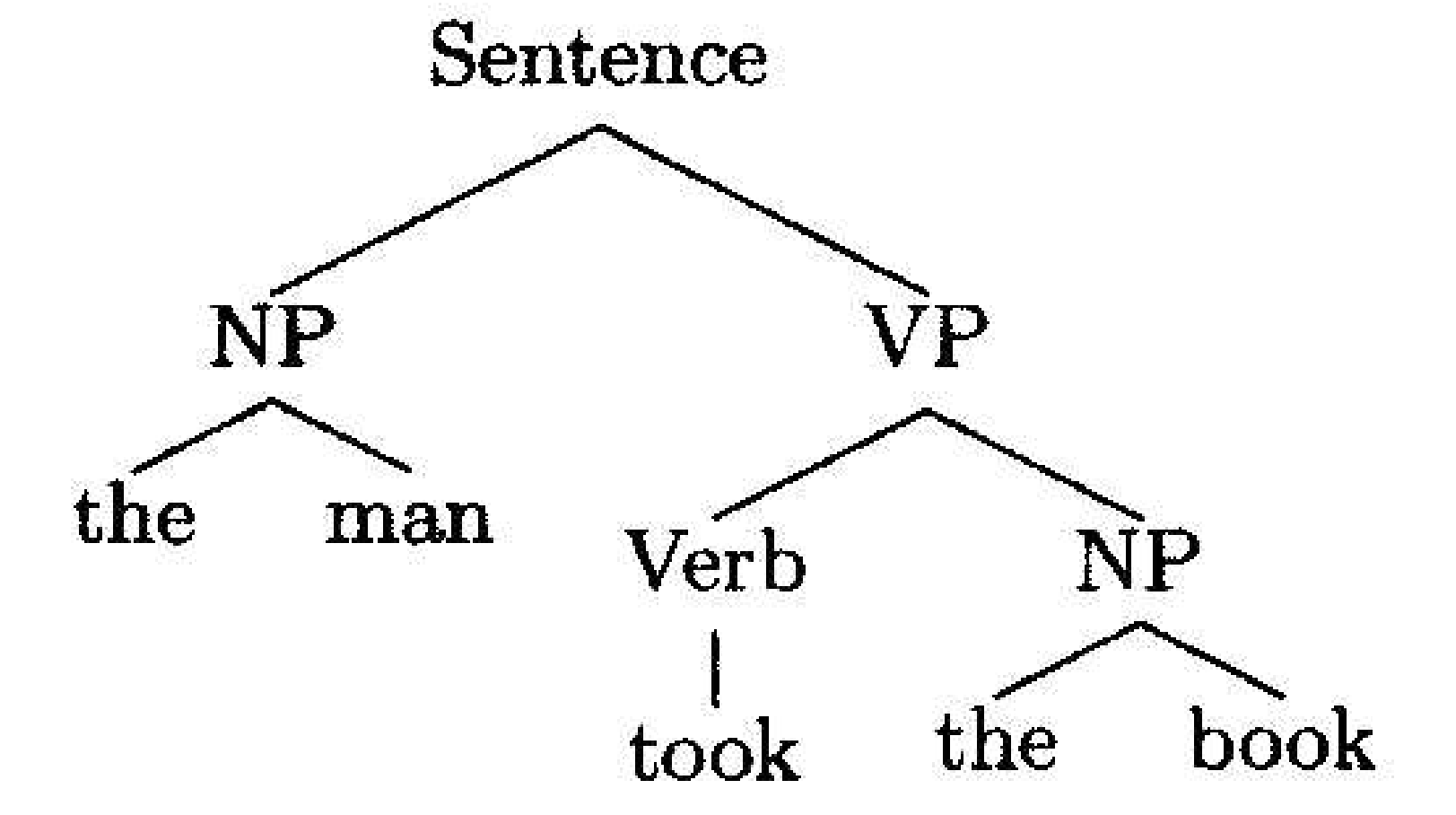
Training des Parsers:	Sektion 02-21
Test-Entwicklung:	Sektion 22
Test-finaler Parser:	Sektion 23

Aufbau des Experiments (2)

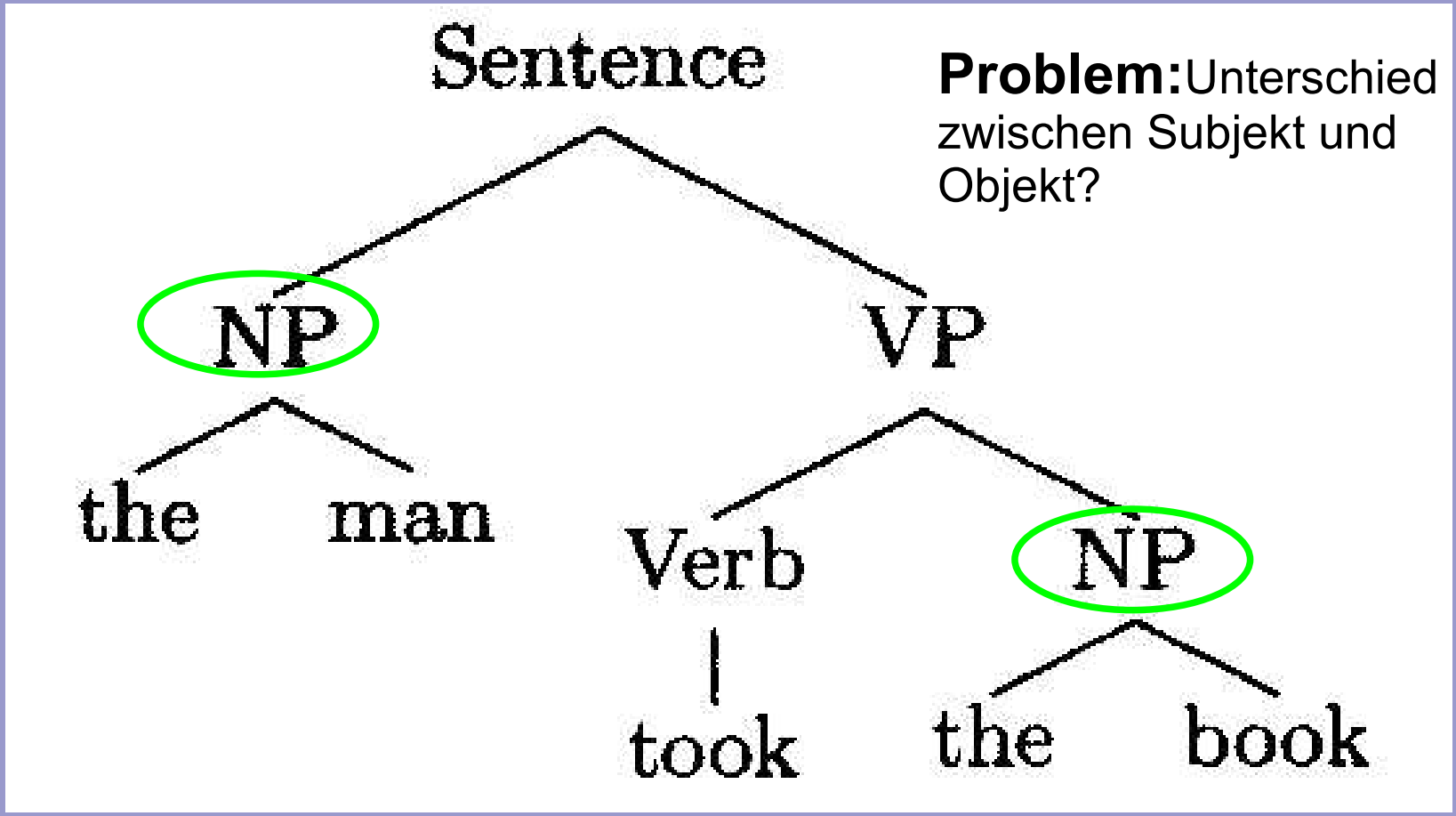
- Einfache Implementierung in Java, basierend auf allgemeinem CKY Parser[1]
- 1 GB Speicher
- Zeit: etwa 3 Sekunden für einen durchschnittlichen Satz

[1]Parser ist verfügbar als Download unter: http://nlp.stanford.edu/downloads/lex-parser.shtml

Annotation

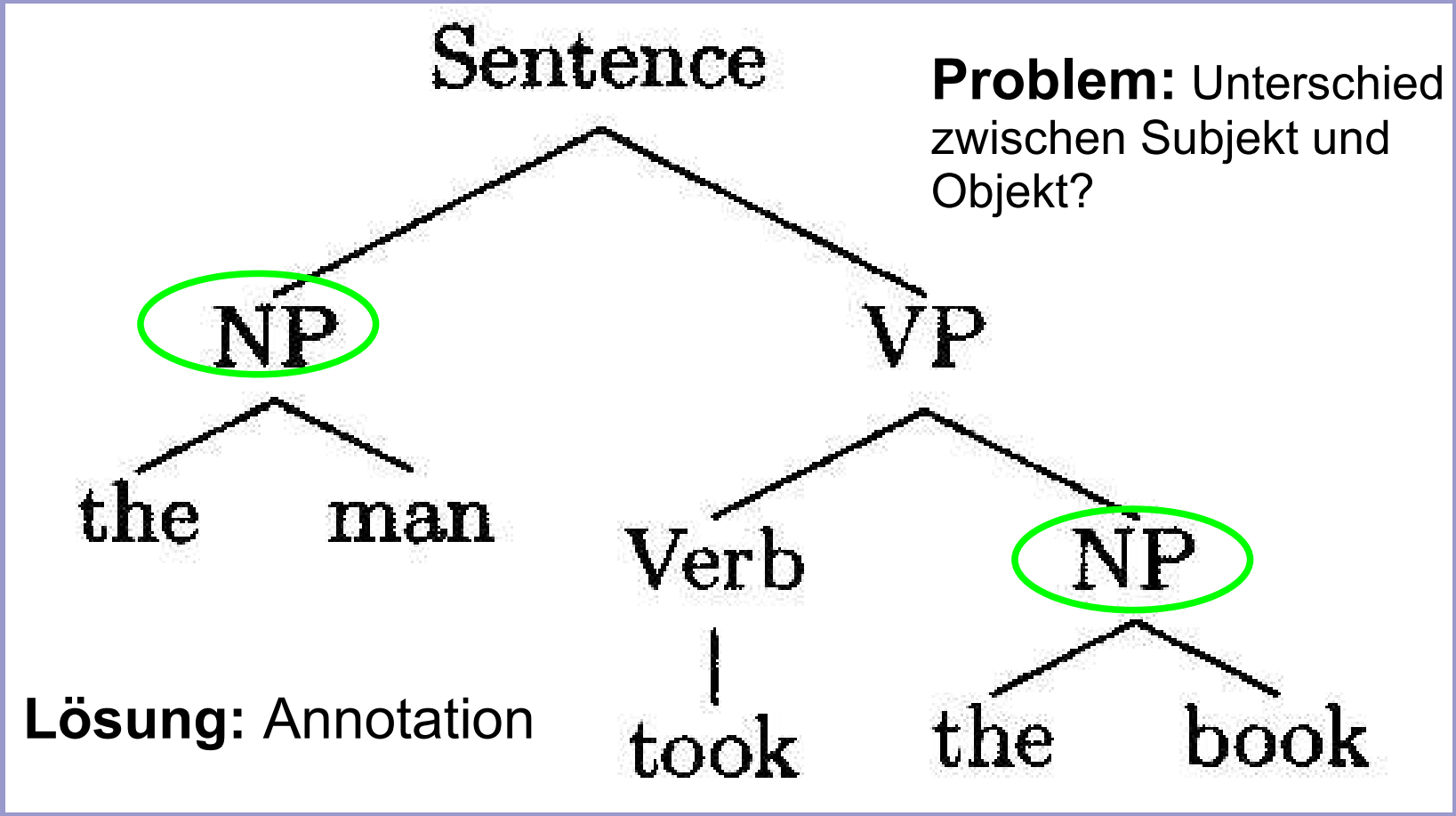


Annotation



Problem: Unterschied zwischen Subjekt und Objekt?

Annotation



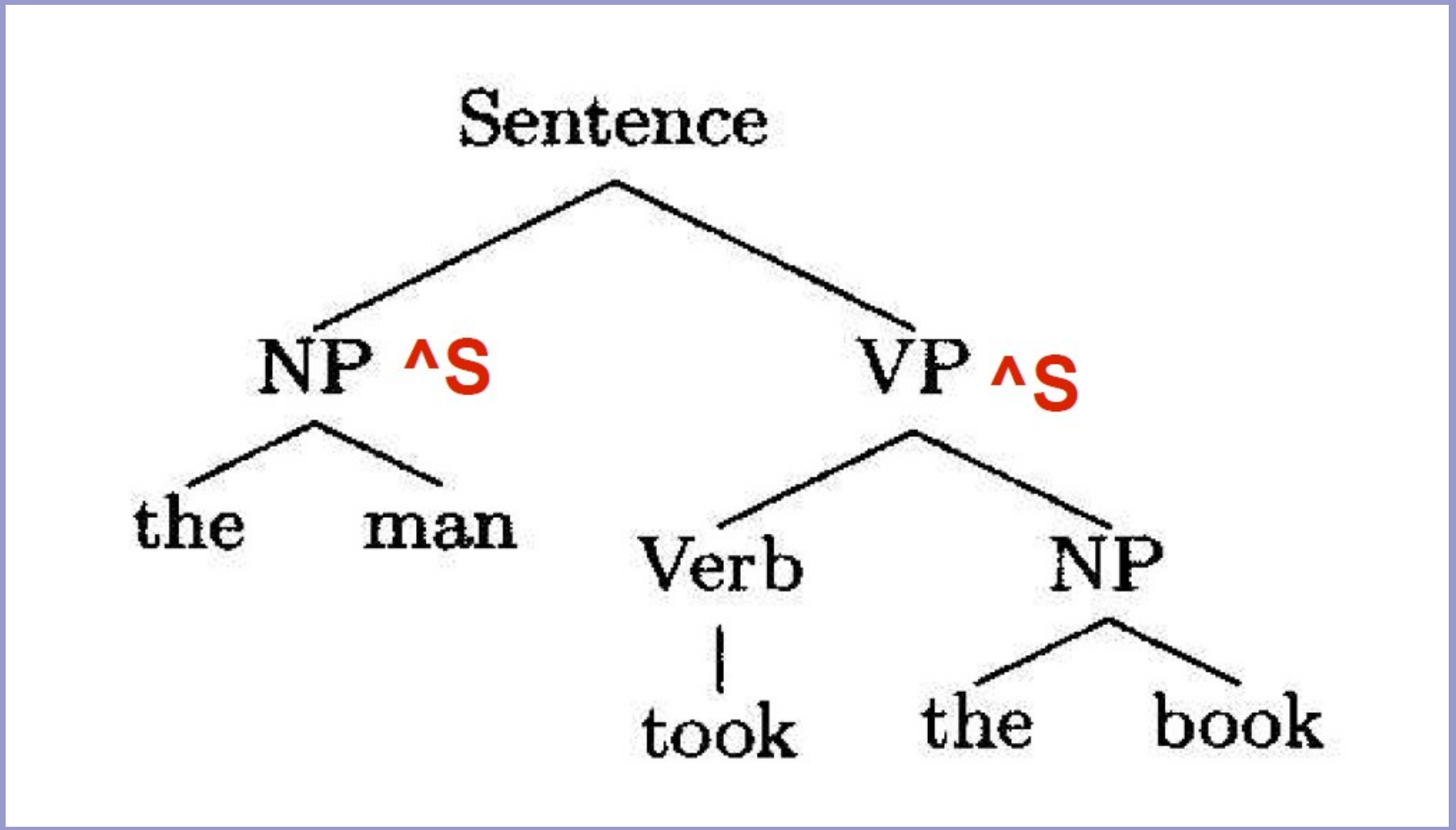
Problem: Unterschied zwischen Subjekt und Objekt?

Lösung: Annotation

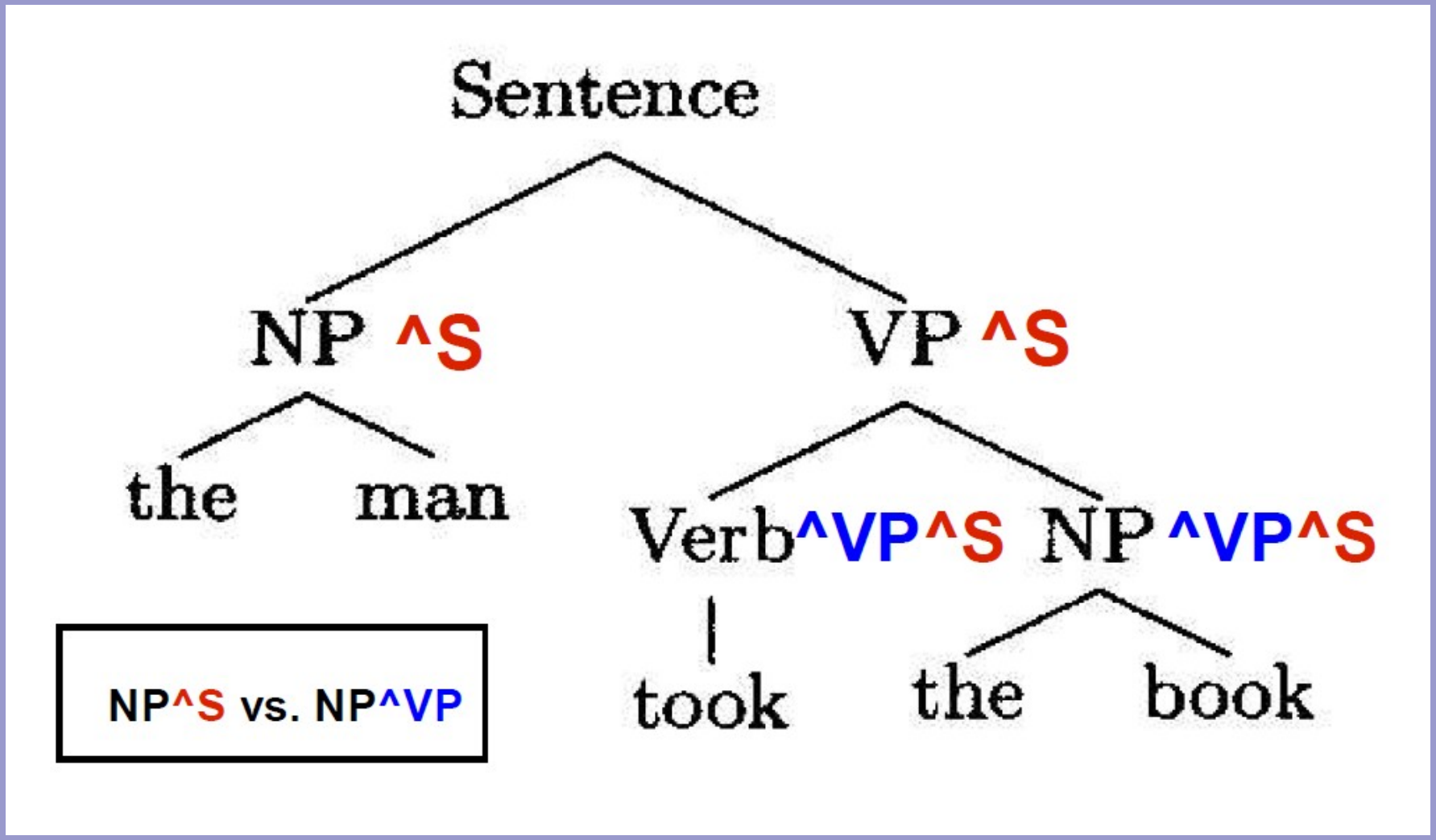
Annotation

- Übernahme der Idee von Johnson (1998)
- Unter Annotation versteht man die Markierung der Knoten
- Die Knoten werden hierbei durch Nichtterminalsymbole markiert

Parent Annotation



Grandparent Annotation



Markovisierung

- Problem: Viele Regeln, die nur einmal oder keinmal im Corpus vorkommen
- Öfteres Vorkommen ähnlicher Regeln im Corpus
- z.B. NP→ DET ADJ N kommt vor!
  Aber: NP→ DET ADJ ADJ N kommt nicht vor!

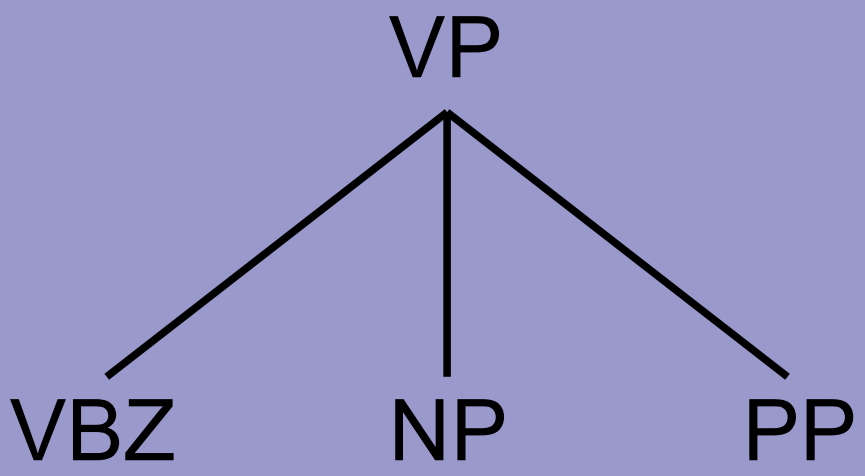
Markovisierung (1)

- Problem: Viele Regeln, die nur einmal oder keinmal im Corpus vorkommen
- Öfteres Vorkommen ähnlicher Regeln im Corpus
- z.B. NP→ DET ADJ N kommt vor!
  Aber: NP→ DET ADJ ADJ N kommt nicht vor!

- Lösung: Markovisierung der Regeln (Collins, 1999)
  Die Regeln werden in Teile aufgebrochen („zerhackt”).
  Als erstes wird der Kopf der Phrase generiert, dann die rechte Seite und danach die linke Seite.

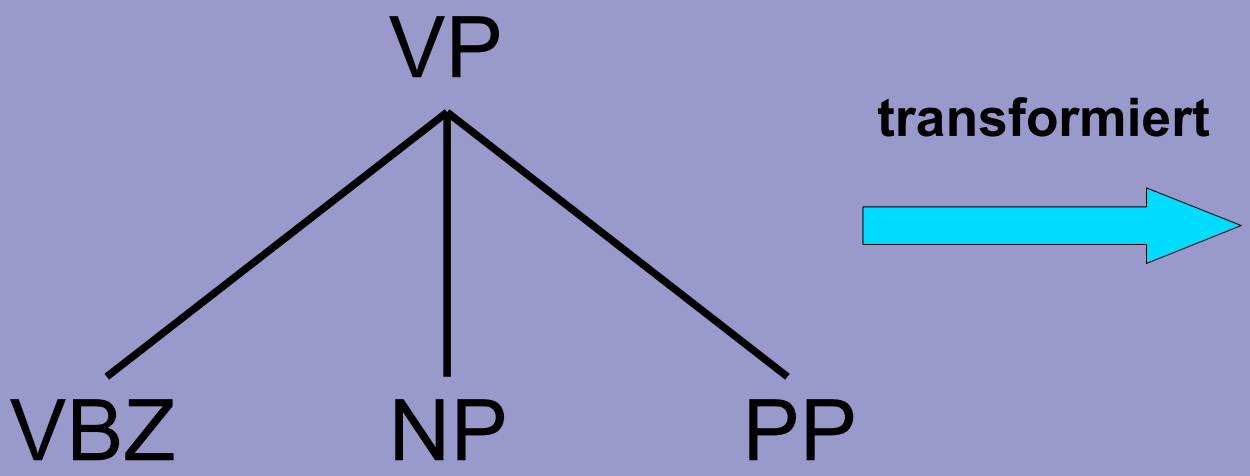
Markovisierung (2)

Transformierung des Parsebaums



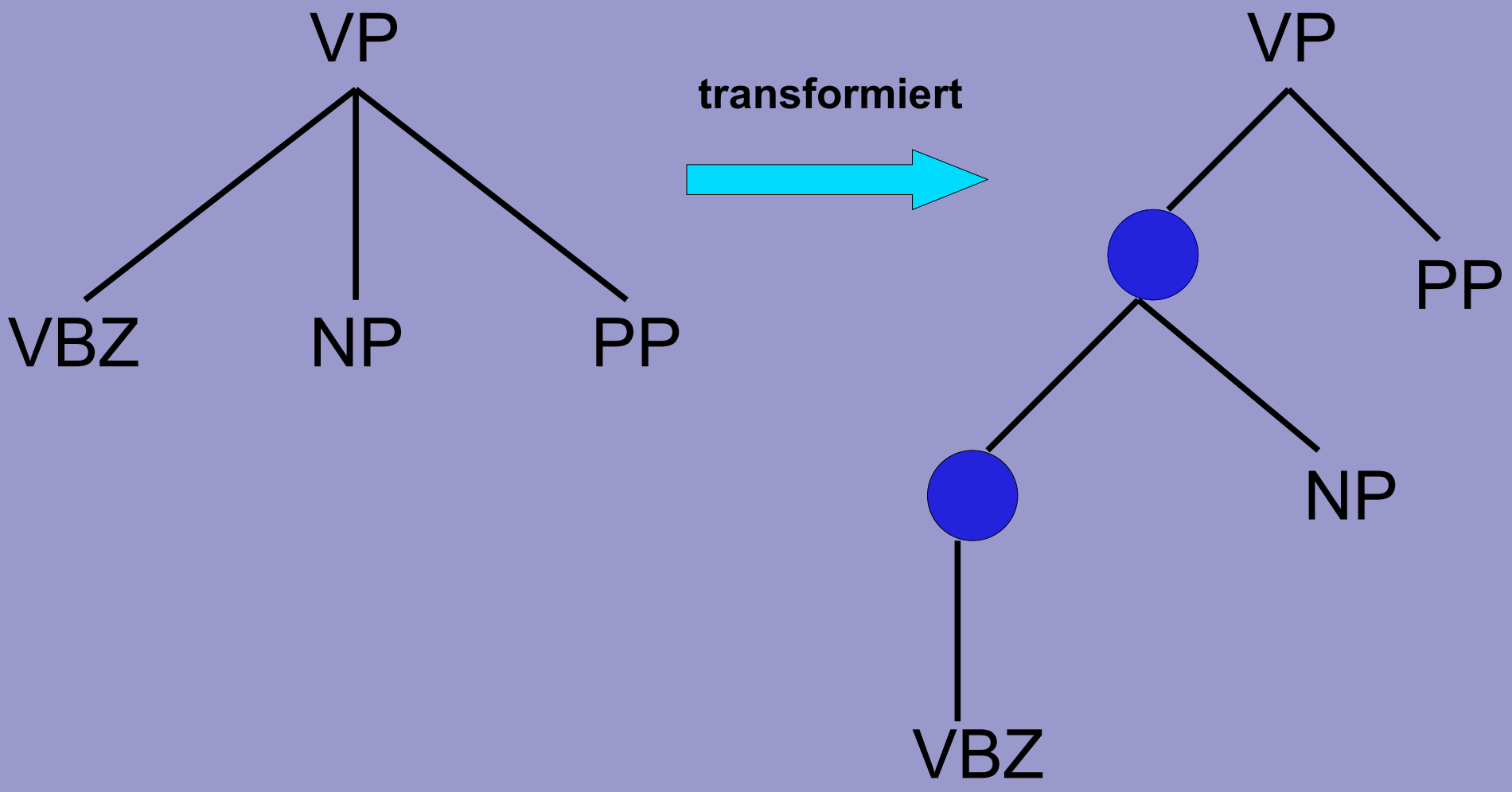
Markovisierung (2)

Transformierung des Parsebaums



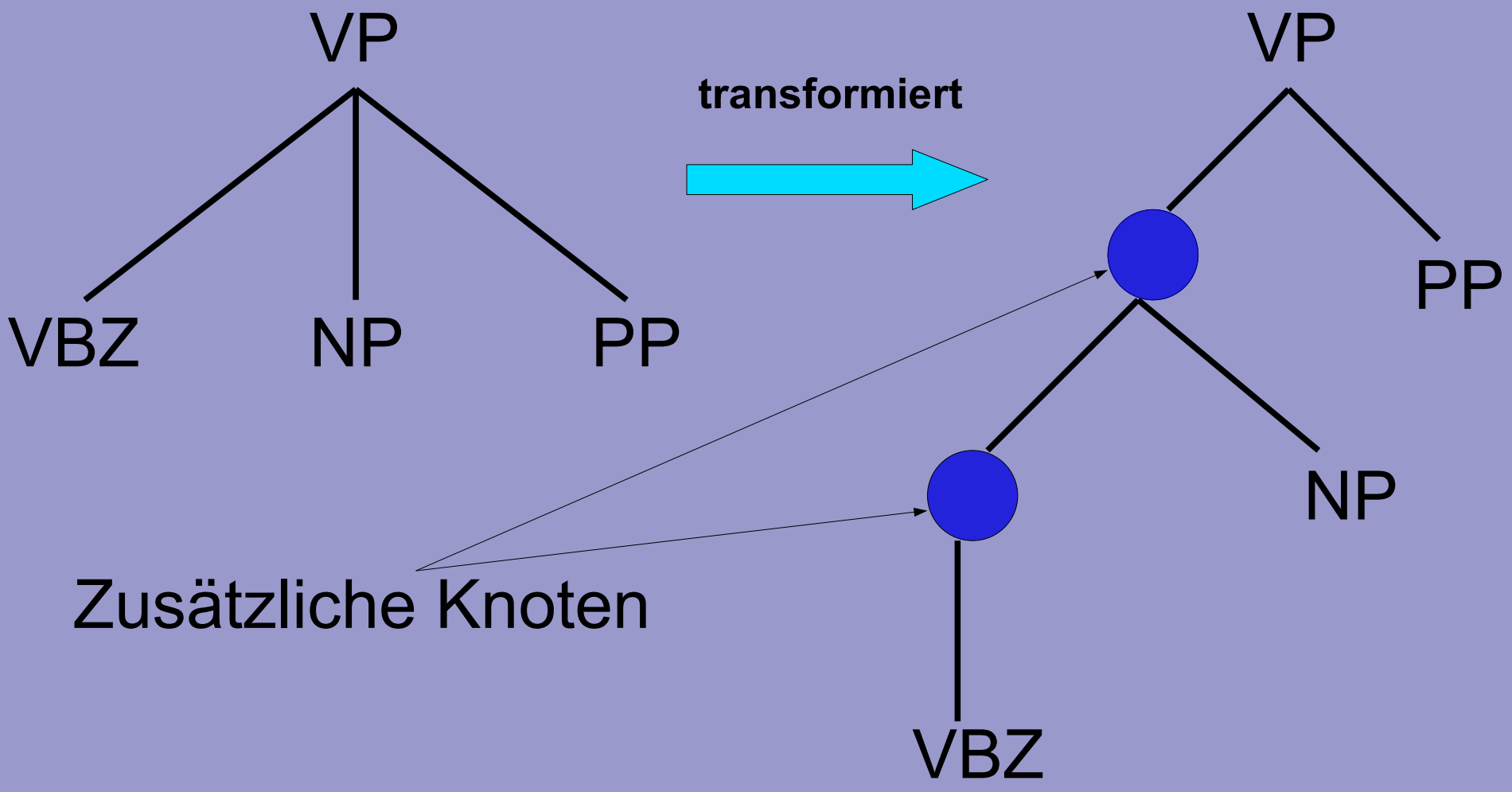
Markovisierung (2)

Transformierung des Parsebaums

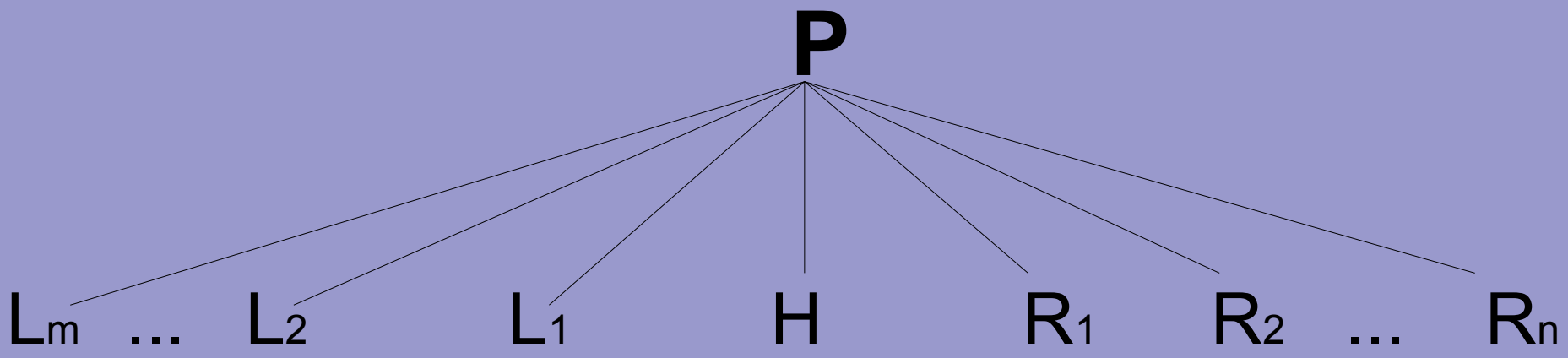


Markovisierung (2)

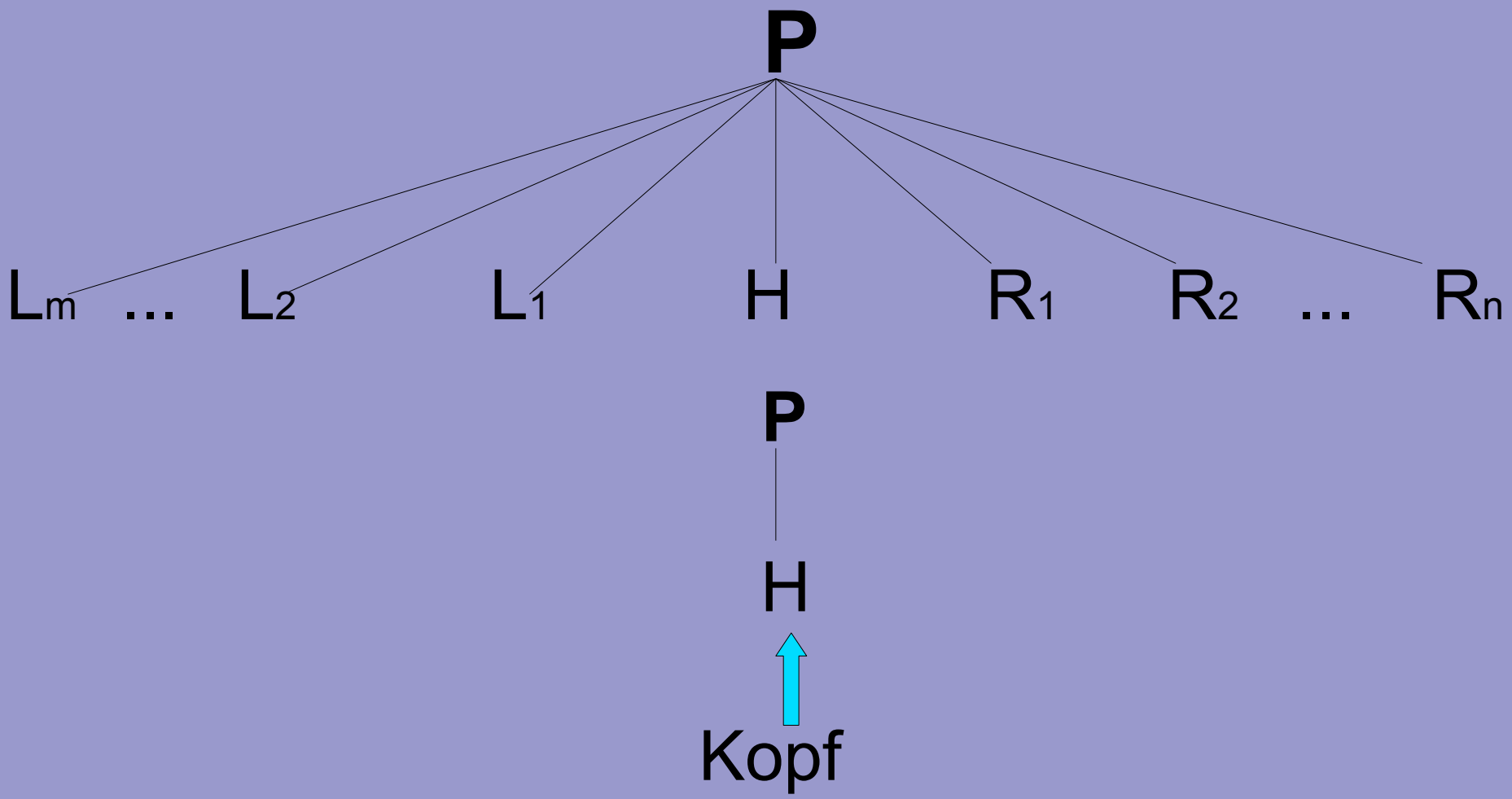
Transformierung des Parsebaums



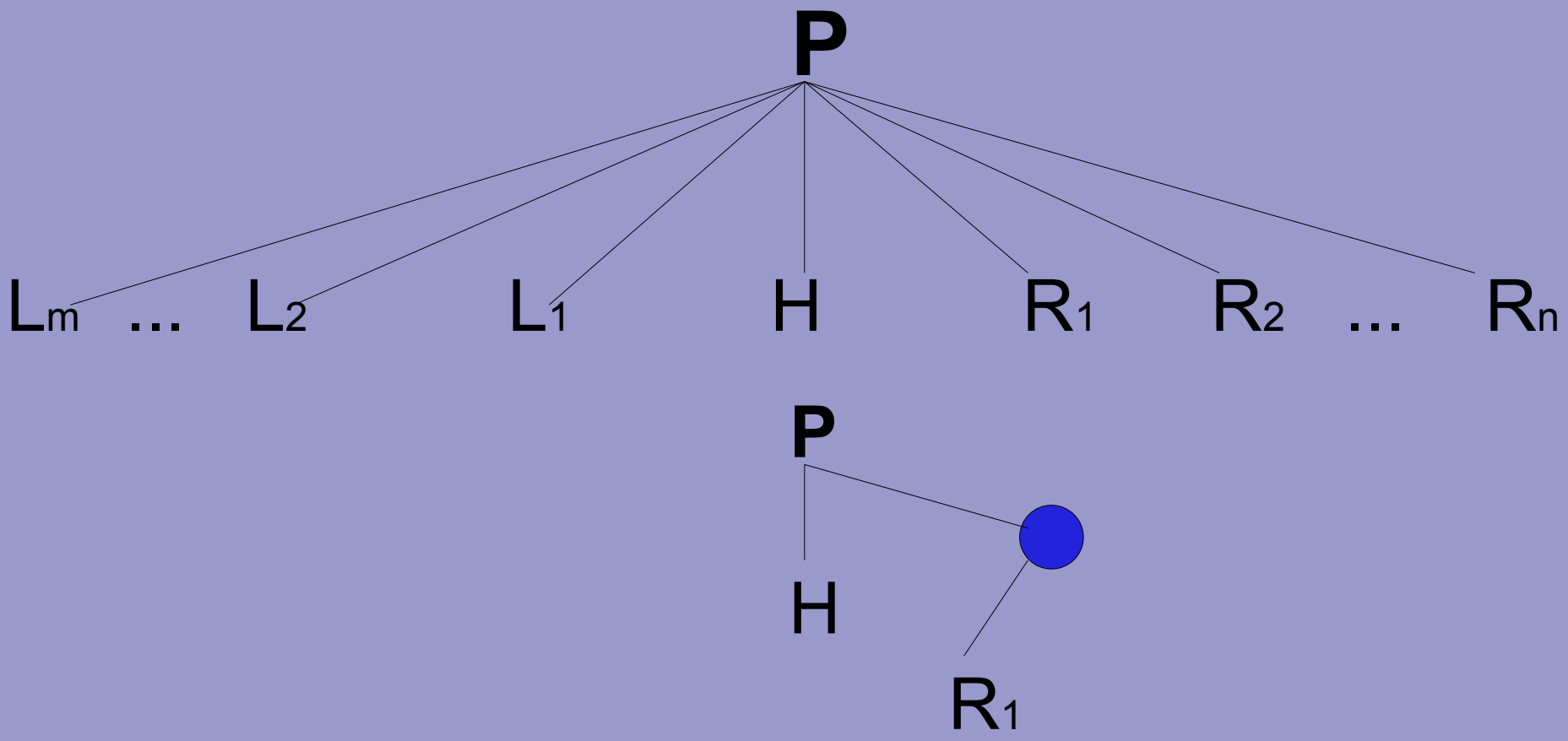
Markovisierung (3)



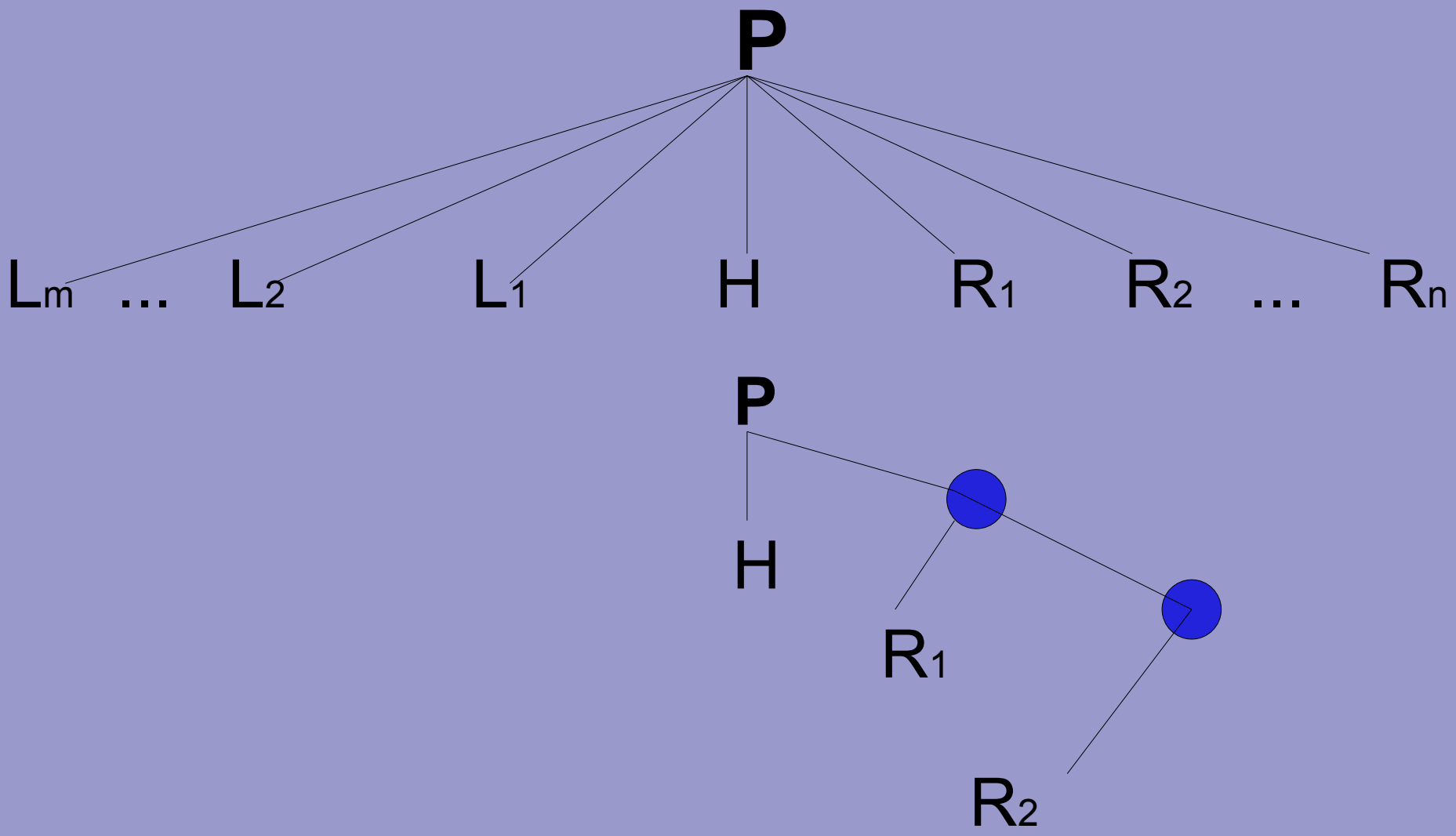
Markovisierung (3)



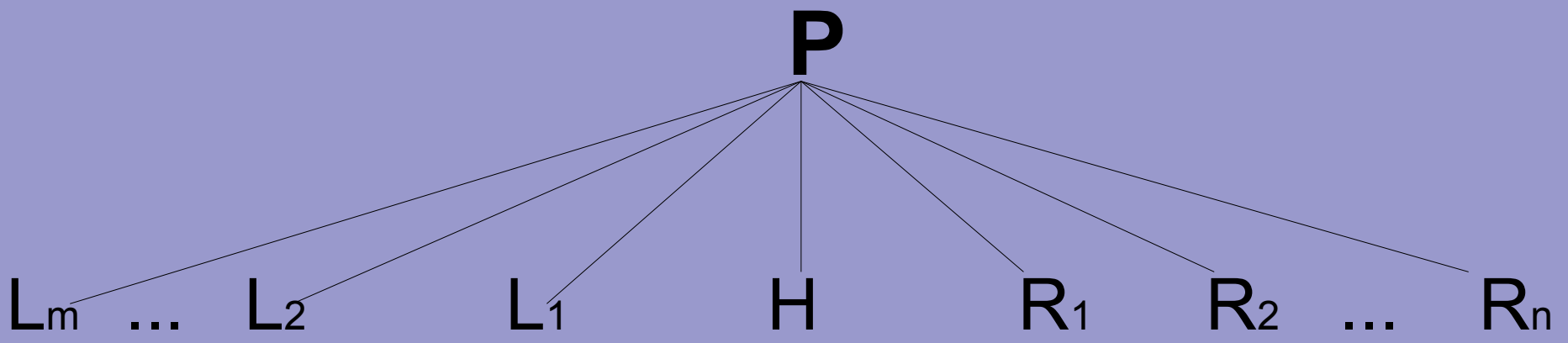
Markovisierung (3)

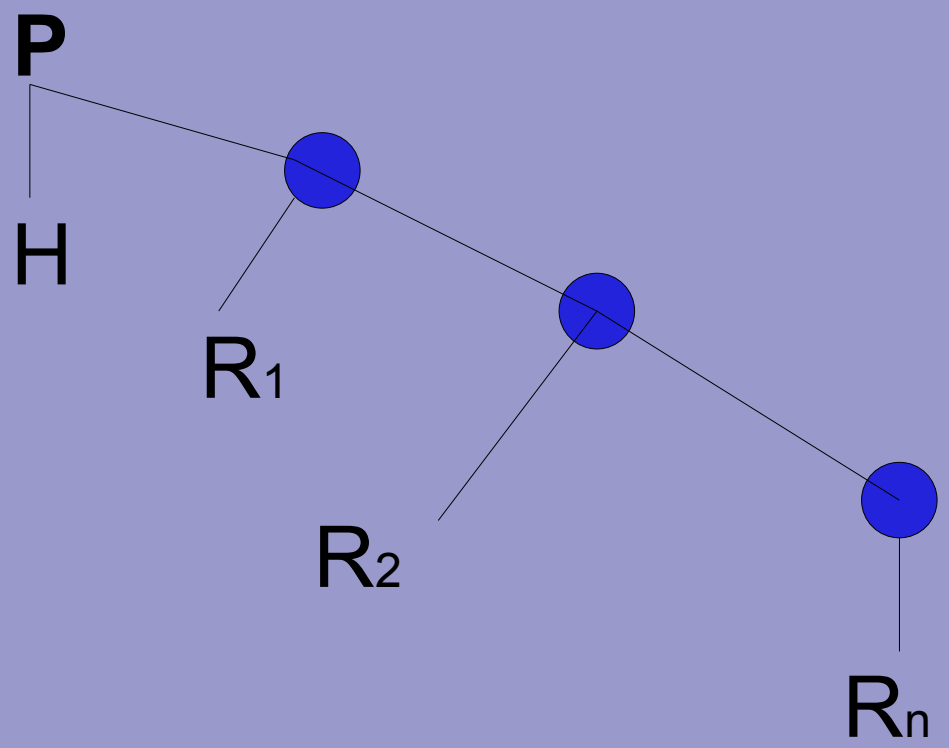


Markovisierung (3)

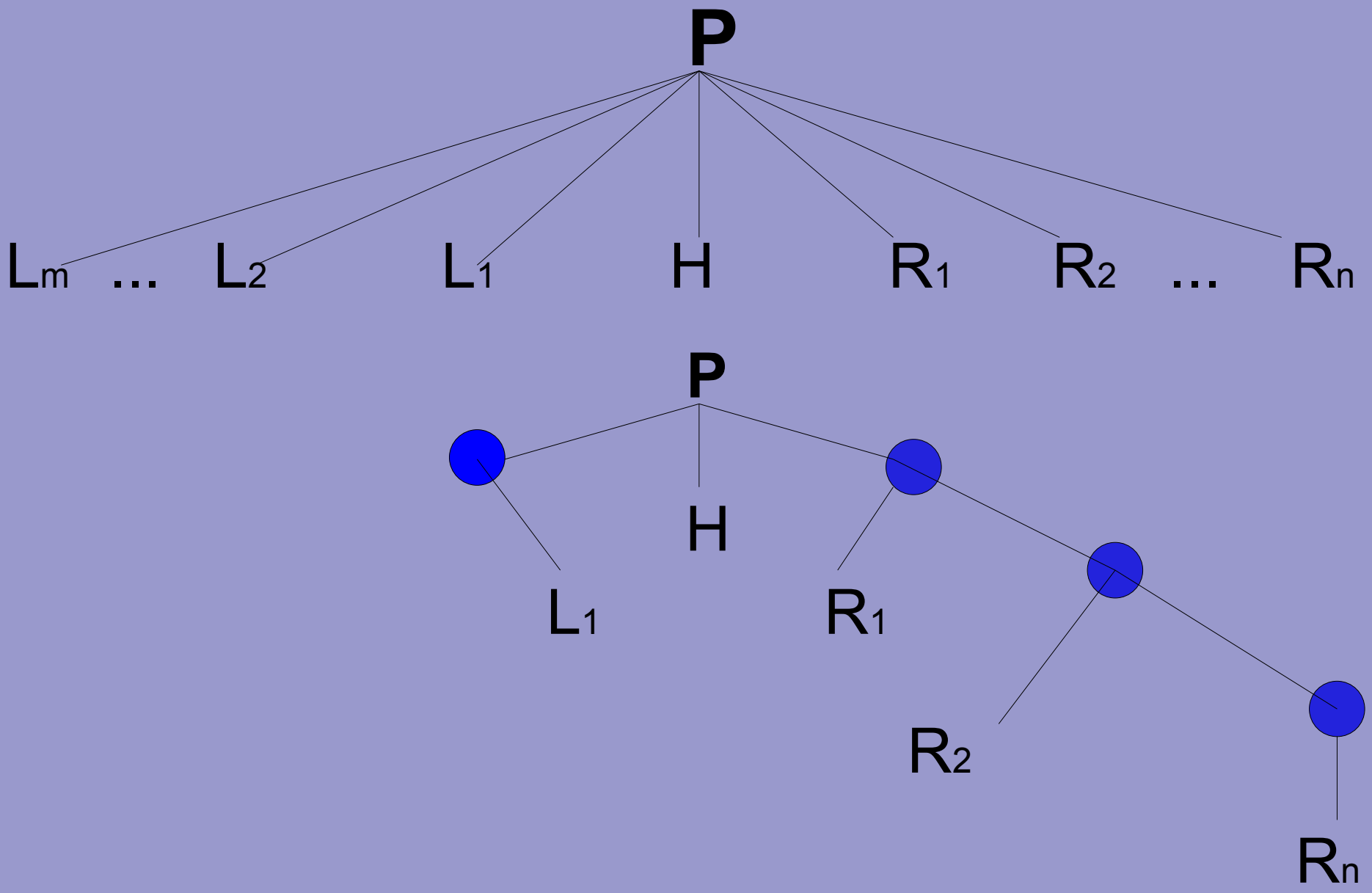


Markovisierung (3)

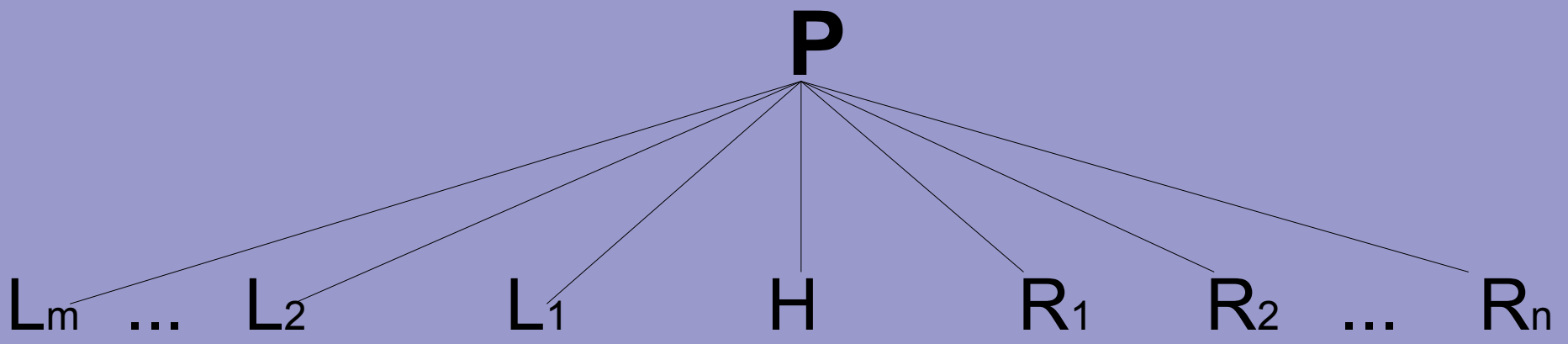


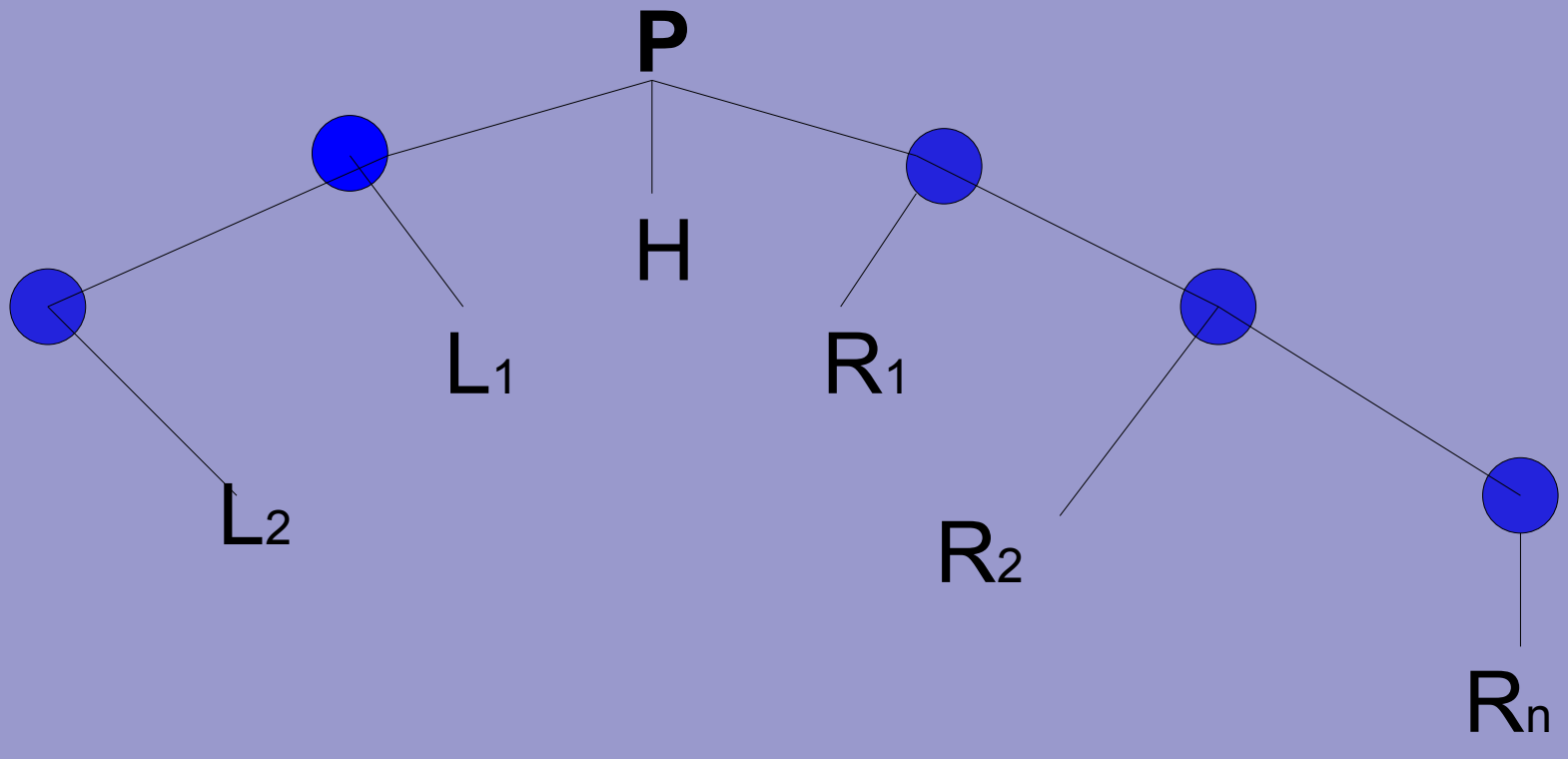


Markovisierung (3)

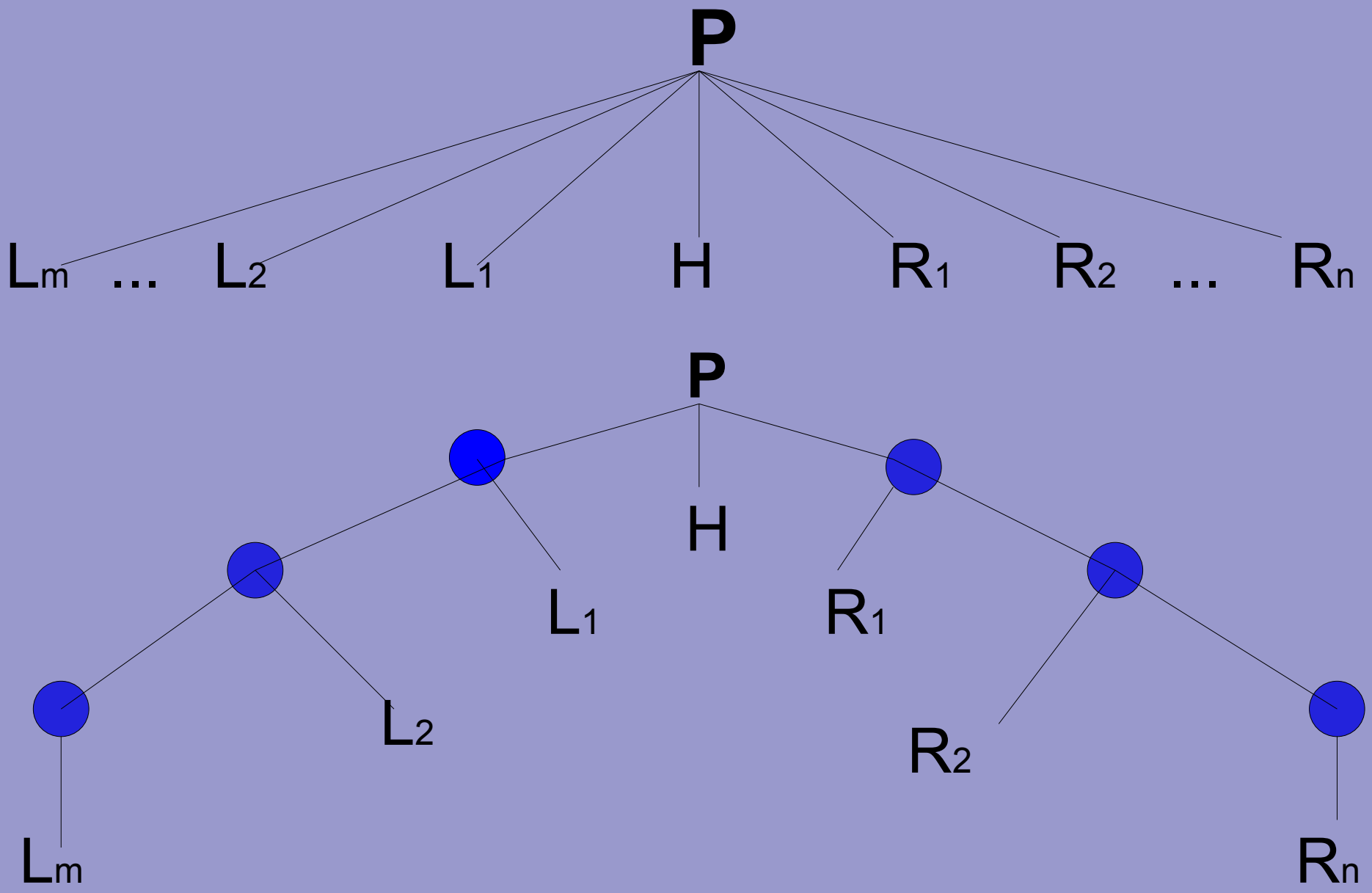


Markovisierung (3)





Markovisierung (3)



Markovisierung (4)

- Grammatiktransformation: Markovisierung der Regel
- Entstehung neuer Grammatik

z.B. $P \rightarrow$  H

 $\rightarrow R_1$

Viele kleine Regeln

Idee: Zuverlässige Kombination für kleine Regelteile

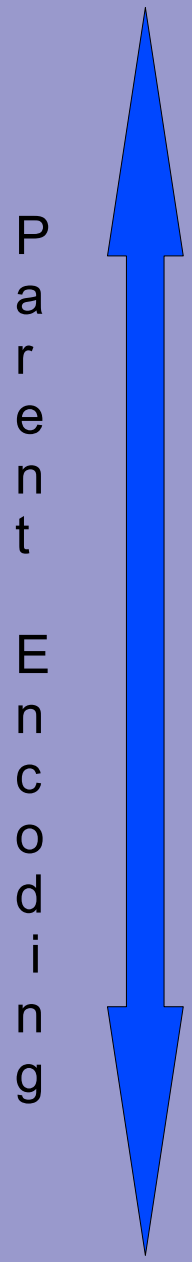
Markovisierung (5)

- **Vertikale Markovisierung**
- **Horizontale Markovisierung**

Markovisierung (5)

- **Vertikale Markovisierung** ist Anzahl der Knoten für„Parent Annotation“
- **Horizontale Markovisierung** ist die Information, die in den zusätzlichen Knoten ist

Vertikale und Horizontale Markovisierung



v = vertical Order

h = horizontal Order

Markovisierung der Regeln

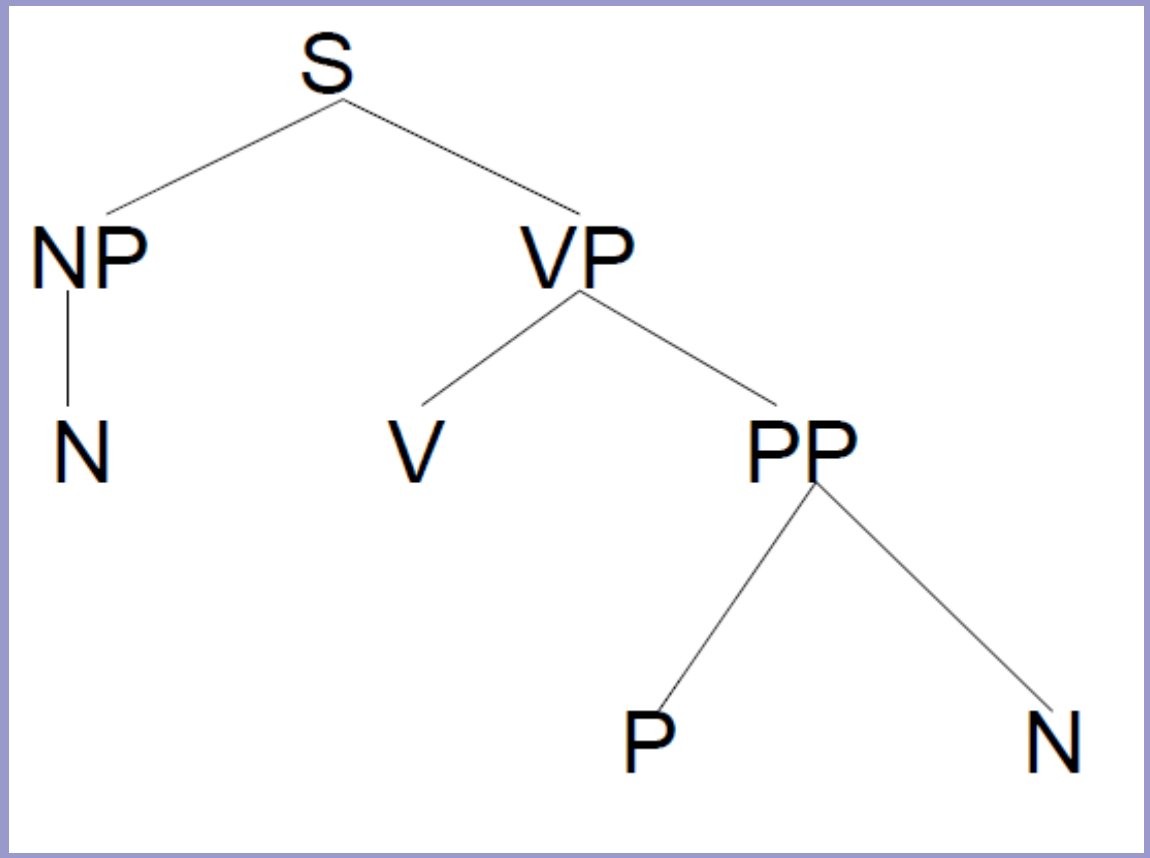
Vertikale Markovisierung (1)

v = 1	Keine Annotation
v = 2	Alle Elternknoten
v ≤ 2	Ausgewählte Elternknoten
v = 3	Alle Großelternknoten
v ≤ 3	Ausgewählte Großelternknoten

Vertikale Markovisierung (2)

v = 1: keine Annotation

Beispielbaum:

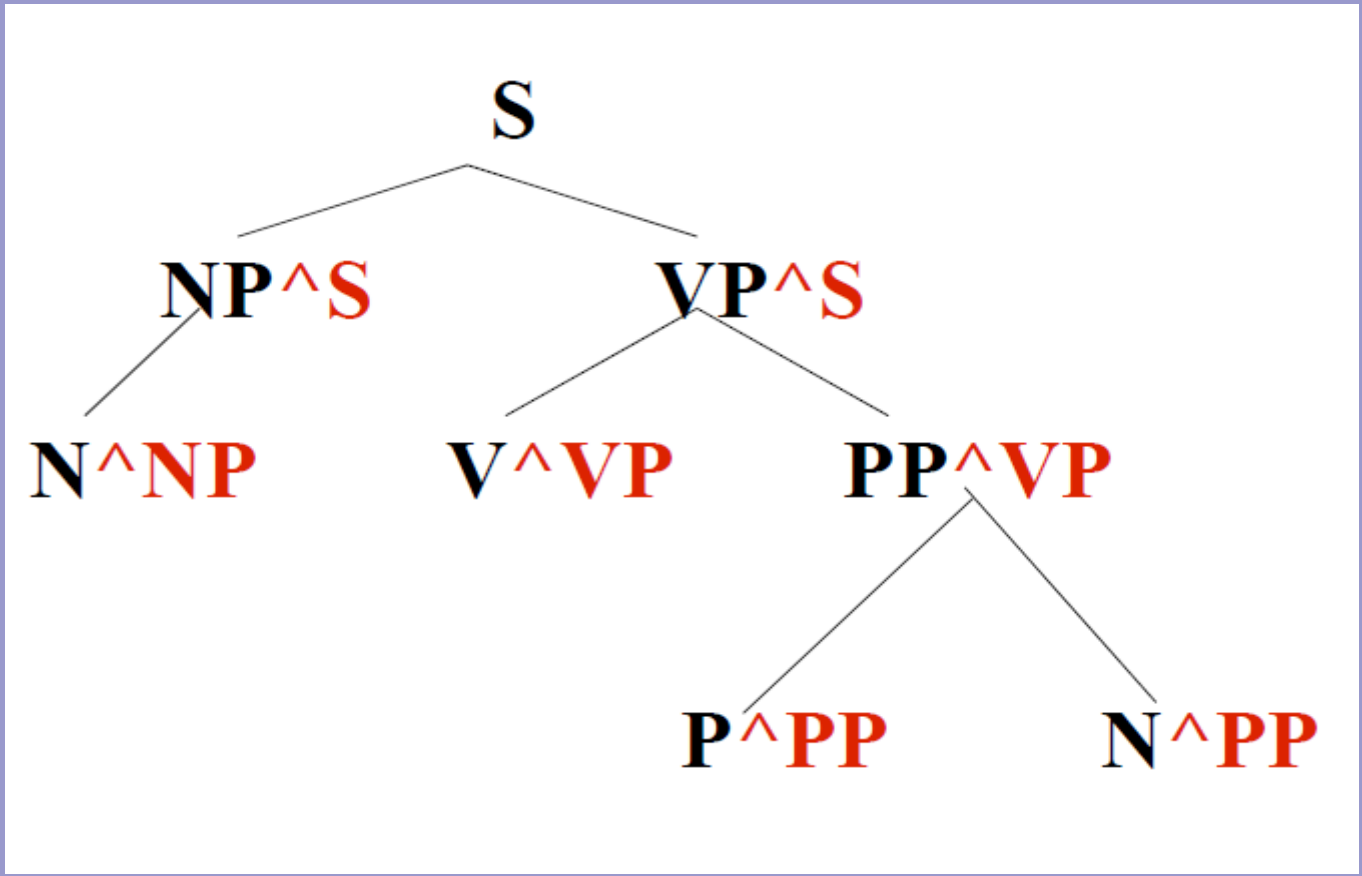


Vertikale Markovisierung (3)

v =2 Alle Elternknoten

man markiert bei allen Kindern, welche ihre Eltern sind

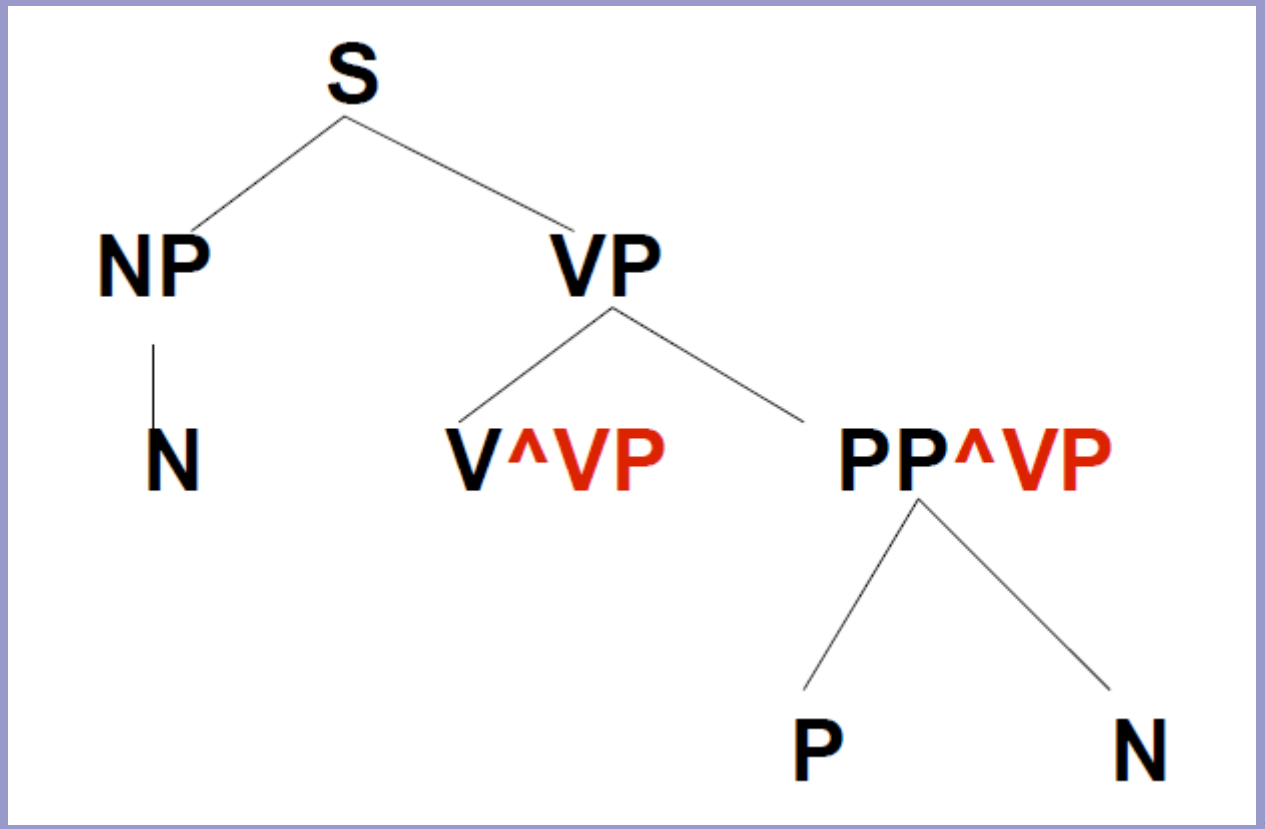
Beispielbaum:



Vertikale Markovisierung (4)

v ≤ 2: Ausgewählte Elternknoten
man markiert alle Kinder, die z.B. ein VP als Elternknoten haben:

Beispielbaum:

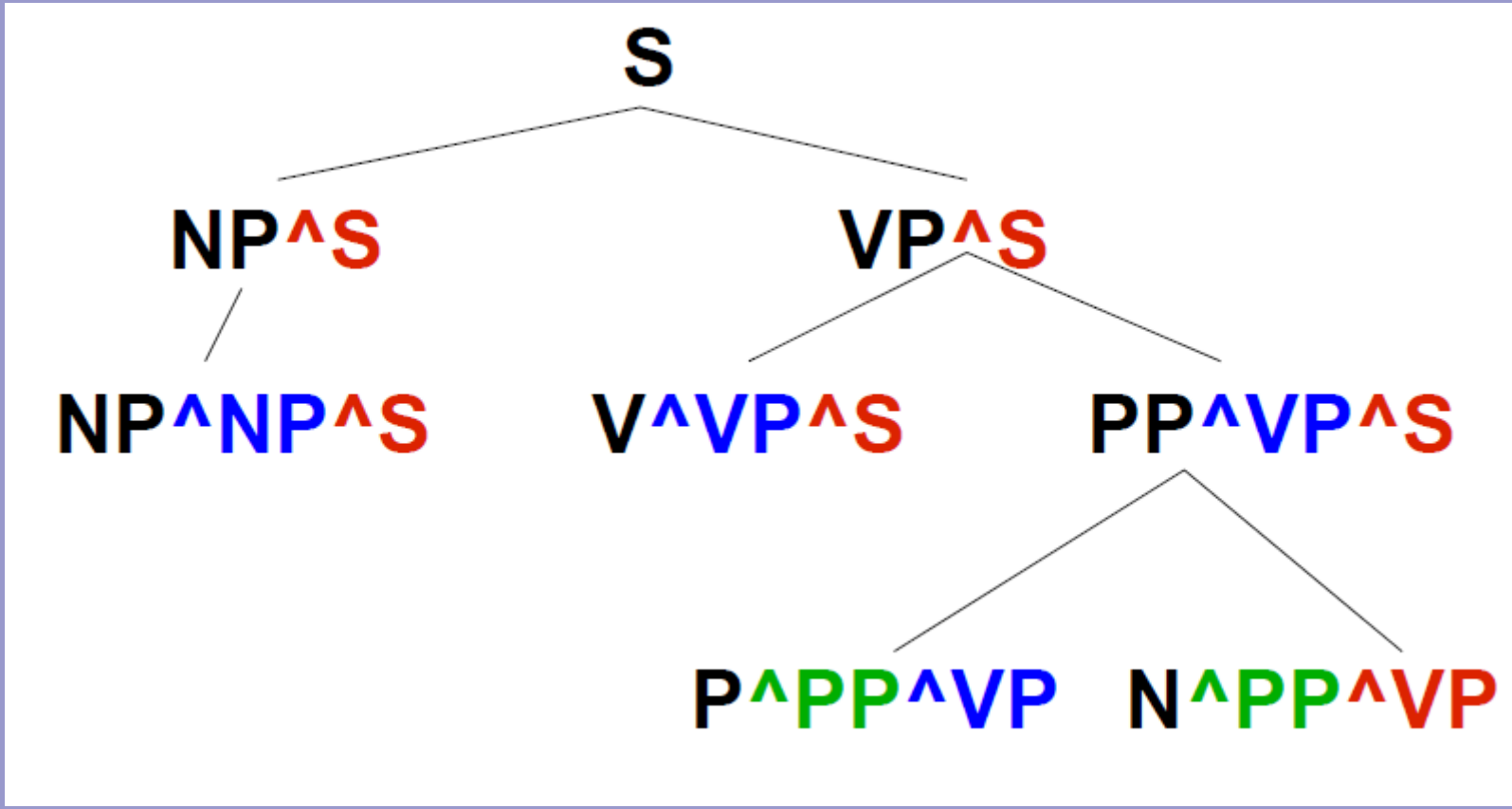


Vertikale Markovisierung (5)

v =3 alle Großelternknoten

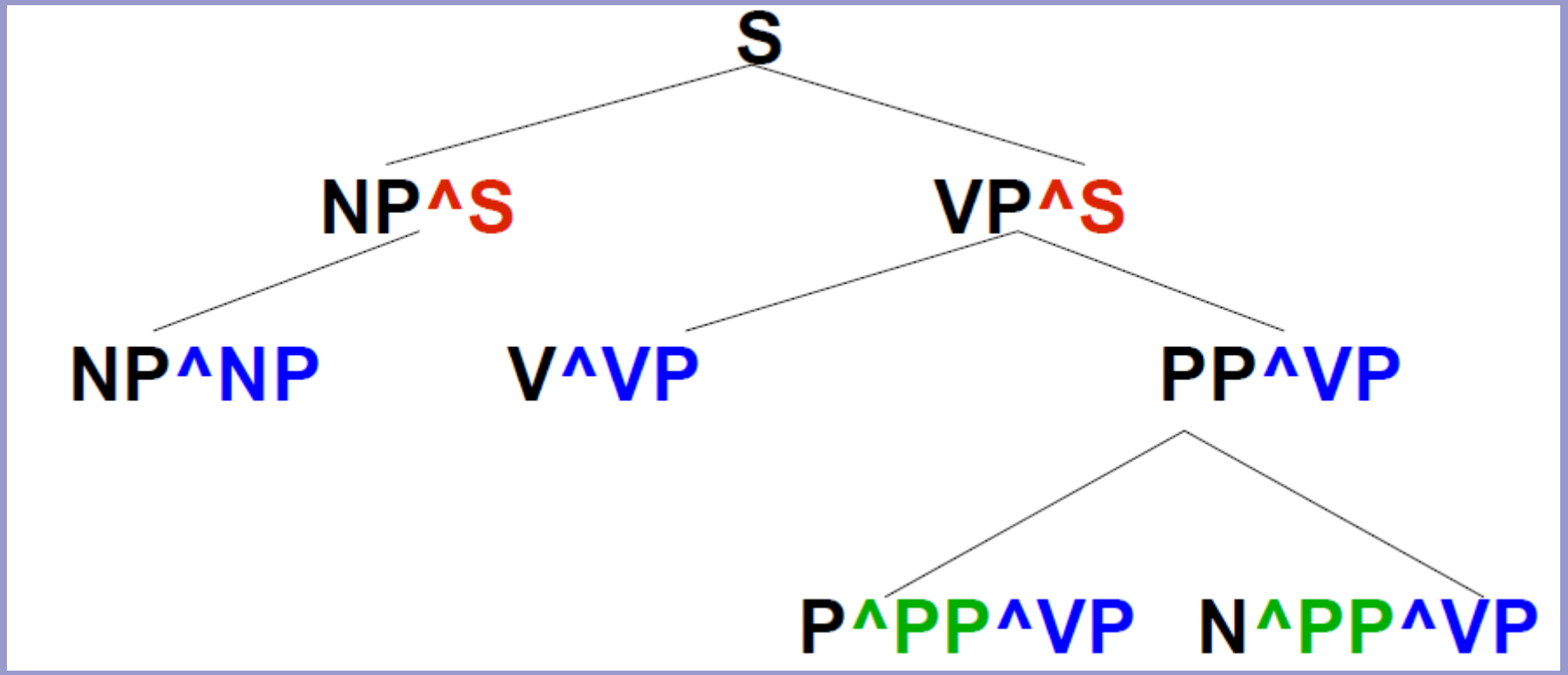
man markiert bei allen Kindern, welche ihre Eltern und Großeltern sind

Beispielbaum:



Vertikale Markovisierung (6)

v ≤ 3 : Ausgewählte Großelternknoten
man markiert: 1. die Elternknoten, 2. ausgewählte Großelternknoten, hier z.B. alle VPs,
Beispielbaum:



Horizontale Markovisierung

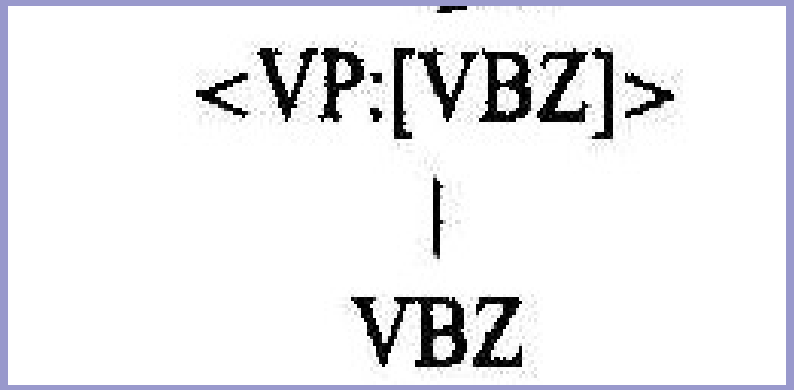
h = 0	Kopf- und Mutter-Kategorie
h = 1	Eine Schwester
h = 2	Zwei Schwestern
h ≤ 2	Eine oder zwei Schwestern
h = ∞	Keine Markovisierung

Vertikale und horizontale Markovisierung

- Z.B. VP→ VBZ NP PP
- Head rule: <VP : [VBZ]> → VBZ
  < >: Zwischensymbol
  [VBZ]: Kopf der Phrase
- Erster rechter Schwesterknoten:
  <VP :[VBZ] ... NP> → <VP : [VBZ]> NP
- Zweiter rechter Schwesterknoten:
  <VP : [VBZ] ... PP> → <VP : [VBZ] ... NP> PP

Vertikale und horizontale Markovisierung

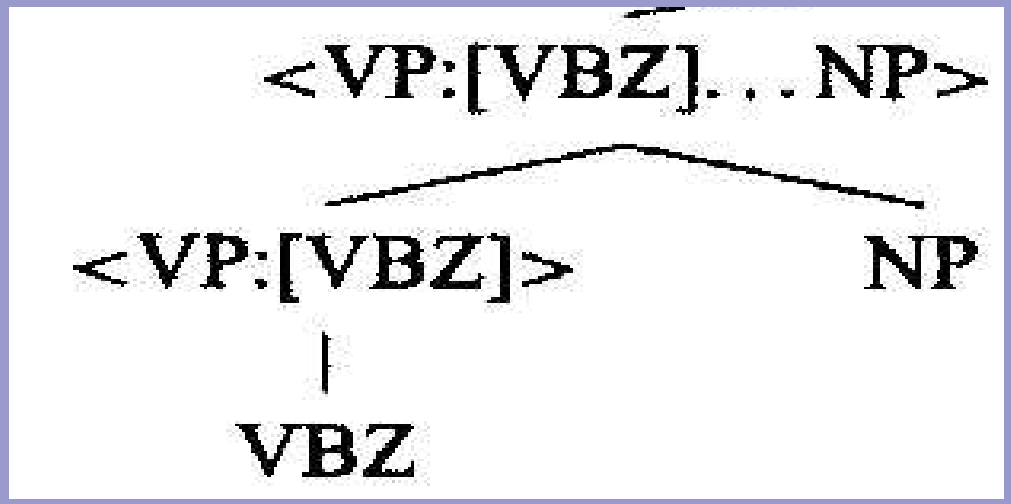
Regel: <VP : [VBZ]> → VBZ



The v = 1, *h* = 1 markovization of VP→ VBZ NP PP.

Vertikale und horizontale Markovisierung

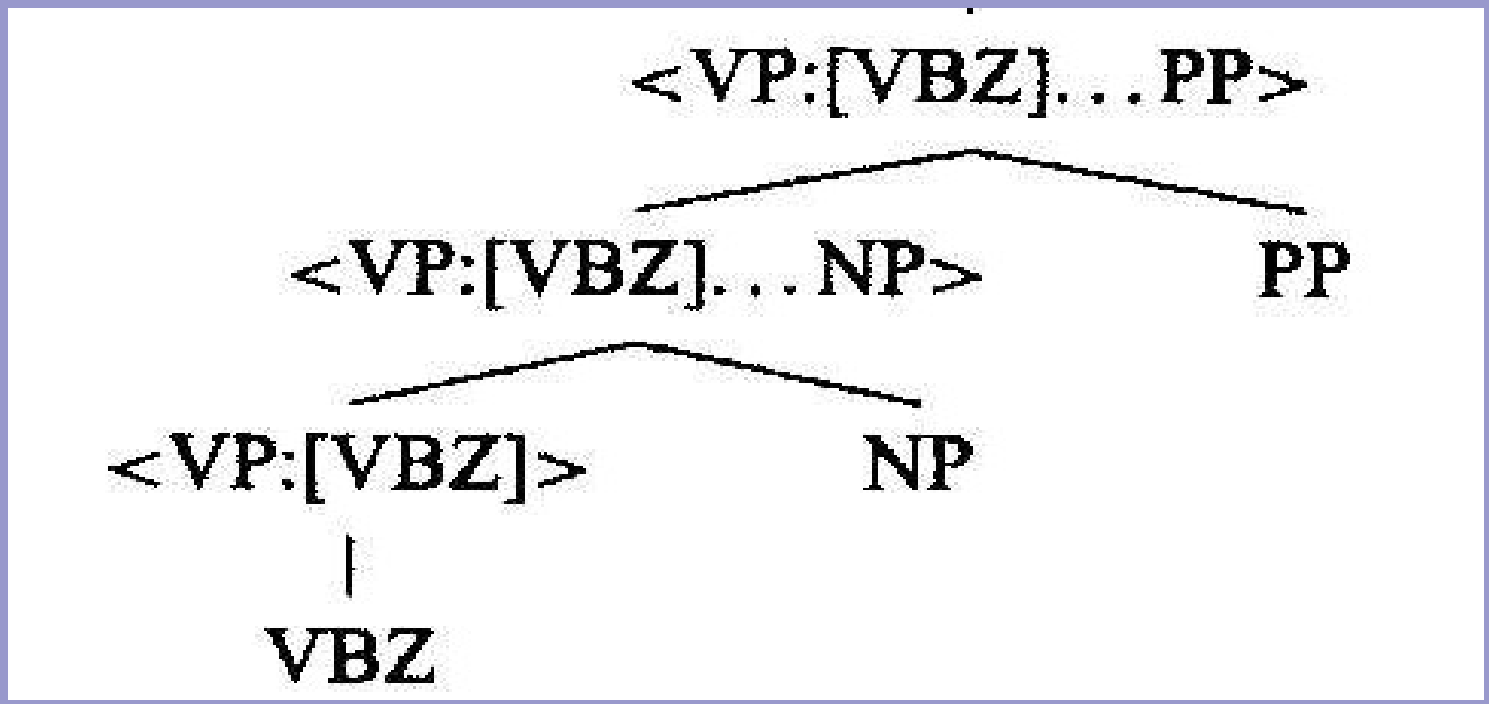
Regel: <VP : [VBZ]> → VBZ ;
<VP :[VBZ] ... NP> → <VP : [VBZ]> NP



The v = 1, *h* = 1 markovization of VP→ VBZ NP PP.

Vertikale und horizontale Markovisierung

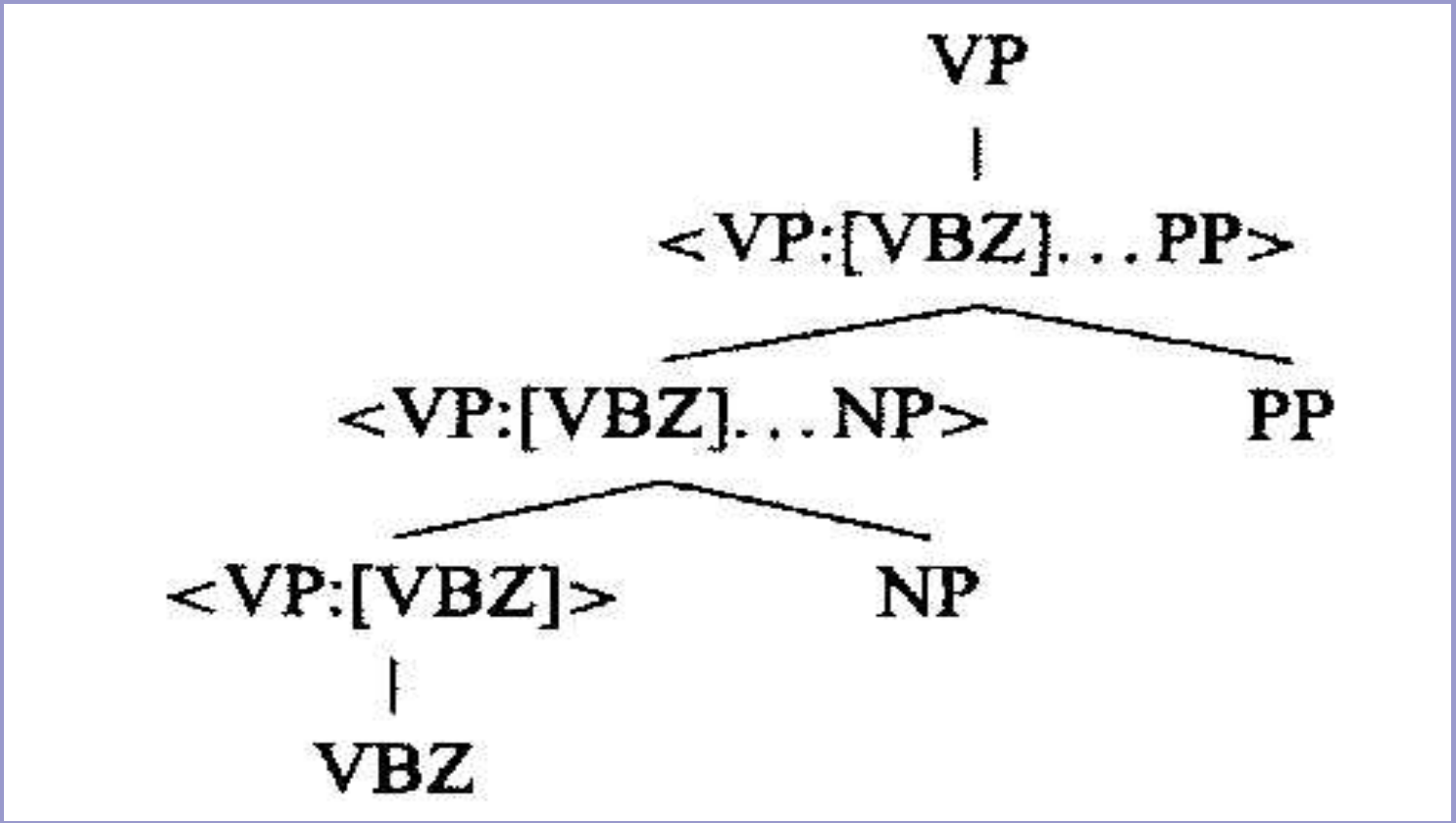
Regel: <VP : [VBZ]> → VBZ ;
<VP : [VBZ] ... NP> → <VP : [VBZ]> NP;
<VP : [VBZ] ... PP> → <VP : [VBZ] ... NP> PP



The v = 1, *h* = 1 markovization of VP→ VBZ NP PP.

Vertikale und horizontale Markovisierung

Regel: <VP : [VBZ]> → VBZ ;
<VP : [VBZ] ... NP> → <VP : [VBZ]> NP;
<VP : [VBZ] ... PP> → <VP : [VBZ] ... NP> PP



The v = 1, *h* = 1 markovization of VP→ VBZ NP PP

Parseval Scores

Precision (LP): Prozentsatz der Klammern im Parse, die mit denen im korrekten Baum übereinstimmen

Recall (LR): Prozentsatz der Klammern des korrekten Baums, die im Parse sind

F_1: harmonisches Mittel (zwischen Precision und Recall)

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}$$

Ergebnisse

Vertical Order	Horizontal Markov Order $h=0$	$h=1$	$h\leq 2$	$h=2$	$h=\infty$
$v=1$ No annotation	71.27	72.5	73.46	72.96	72.62
	(854)	(3119)	(3863)	(6207)	(9657)
$v\leq 2$ Sel. Parents	74.75	77.42	77.77	77.50	76.91
	(2285)	(6564)	(7619)	(11398)	(14247)
$v=2$ All Parents	74.68	77.42	77.81	77.50	76.81
	(2984)	(7312)	(8367)	(12132)	(14666)
$v\leq 3$ Sel. GParents	76.50	78.59	79.07	78.97	78.54
	(4943)	(12374)	(13627)	(19545)	(20123)
$v=3$ All GParents	76.74	79.18	79.74	79.07	78.72
	(7797)	(15740)	(16994)	(22886)	(22002)

Ergebnisse

Vertical Order		Horizontal Markov Order $h=0$	$h=1$	$h\leq 2$	$h=2$	$h=\infty$
$v=1$	No annotation	71.27 (854)	72.5 (3119)	73.46 (3863)	72.96 (6207)	72.62 (9657)
$v\leq 2$	Sel. Parents	74.75 (2285)	77.42 (6564)	77.77 (7619)	77.50 (11398)	76.91 (14247)
$v=2$	All Parents	74.68 (2984)	77.42 (7312)	77.81 (8367)	77.50 (12132)	76.81 (14666)
$v\leq 3$	Sel. GParents	76.50 (4943)	78.59 (12374)	79.07 (13627)	78.97 (19545)	78.54 (20123)
$v=3$	All GParents	76.74 (7797)	79.18 (15740)	79.74 (16994)	79.07 (22886)	78.72 (22002)

Beispiel:

- Row Treebank: v = 1, h = ∞

Ergebnisse

Vertical Order	Horizontal Markov Order $h = 0$	$h = 1$	$h \leq 2$	$h = 2$	$h = \infty$
$v = 1$ No annotation	71.27	72.5	73.46	72.96	72.62
	(854)	(3119)	(3863)	(6207)	(9657)
$v \leq 2$ Sel. Parents	74.75	77.42	77.77	77.50	76.91
	(2285)	(6564)	(7619)	(11398)	(14247)
$v = 2$ All Parents	74.68	77.42	77.81	77.50	76.81
	(2984)	(7312)	(8367)	(12132)	(14666)
$v \leq 3$ Sel. GParents	76.50	78.59	79.07	78.97	78.54
	(4943)	(12374)	(13627)	(19545)	(20123)
$v = 3$ All GParents	76.74	79.18	79.74	79.07	78.72
	(7797)	(15740)	(16994)	(22886)	(22002)

Beispiele:

- Row Treebank: v = 1, h = ∞
- Johnson 98: v = 2, h = ∞

Ergebnisse

Vertical Order	Horizontal Markov Order $h = 0$	$h = 1$	$h \leq 2$	$h = 2$	$h = \infty$
$v = 1$ No annotation	71.27 (854)	72.5 (3119)	73.46 (3863)	72.96 (6207)	72.62 (9657)
$v \leq 2$ Sel. Parents	74.75 (2285)	77.42 (6564)	77.77 (7619)	77.50 (11398)	76.91 (14247)
$v = 2$ All Parents	74.68 (2984)	77.42 (7312)	77.81 (8367)	77.50 (12132)	76.81 (14666)
$v \leq 3$ Sel. GParents	76.50 (4943)	78.59 (12374)	79.07 (13627)	78.97 (19545)	78.54 (20123)
$v = 3$ All GParents	76.74 (7797)	79.18 (15740)	79.74 (16994)	79.07 (22886)	78.72 (22002)

Beispiele:

- Row Treebank: v = 1, h = ∞
- Johnson 98: v = 2, h = ∞
- Collins 99: v = 2, h = 2

Ergebnisse

Vertical Order	Horizontal Markov Order $h = 0$	$h = 1$	$h \leq 2$	$h = 2$	$h = \infty$
$v = 1$ No annotation	71.27 (854)	72.5 (3119)	73.46 (3863)	72.96 (6207)	72.62 (9657)
$v \leq 2$ Sel. Parents	74.75 (2285)	77.42 (6564)	77.77 (7619)	77.50 (11398)	76.91 (14247)
$v = 2$ All Parents	74.68 (2984)	77.42 (7312)	77.81 (8367)	77.50 (12132)	76.81 (14666)
$v \leq 3$ Sel. GParents	76.50 (4943)	78.59 (12374)	79.07 (13627)	78.97 (19545)	78.54 (20123)
$v = 3$ All GParents	76.74 (7797)	79.18 (15740)	79.74 (16994)	79.07 (22886)	78.72 (22002)

Beispiele:

- Row Treebank: v = 1, h = ∞
- Johnson 98: v = 2, h = ∞
- Collins 99: v = 2, h = 2
- Bestes F_1 v = 3, h ≤ 2

„Unlexikalisierte Grammatik“

- Qualitative Unterscheidung zwischen „Lexical Words“ und „Function Words“
- Function Words haben sicheren Einfluss auf die syntaktische Struktur:
  - PP[to] ✓
  - NP[stocks] **x**
- Keine Annotation von Lexical Words, aber von Function Words
- ---> Grammatik ist nicht „echt“ unlexikalisiert

Ergebnisse nach Linguistischer Annotierung

	Cumulative			Indiv.
Annotation	Size	F_1	ΔF_1	ΔF_1
Baseline ($v \leq 2, h \leq 2$)	7619	77.77	–	–
UNARY-INTERNAL	8065	78.32	0.55	0.55
UNARY-DT	8066	78.48	0.71	0.17
UNARY-RB	8069	78.86	1.09	0.43
TAG-PA	8520	80.62	2.85	2.52
SPLIT-IN	8541	81.19	3.42	2.12
SPLIT-AUX	9034	81.66	3.89	0.57
SPLIT-CC	9190	81.69	3.92	0.12
SPLIT-%	9255	81.81	4.04	0.15
TMP-NP	9594	82.25	4.48	1.07
GAPPED-S	9741	82.28	4.51	0.17
POSS-NP	9820	83.06	5.29	0.28
SPLIT-VP	10499	85.72	7.95	1.36
BASE-NP	11660	86.04	8.27	0.73
DOMINATES-V	14097	86.91	9.14	1.42
RIGHT-REC-NP	15276	87.04	9.27	1.94

Linguistische Annotationen-Ablauf(1)

Externe vs. Interne Annotation

Annotation
Baseline ($v \leq 2, h \leq 2$)
UNARY-INTERNAL
UNARY-DT
UNARY-RB
TAG-PA
SPLIT-IN
SPLIT-AUX
SPLIT-CC
SPLIT-%
TMP-NP
GAPPED-S
POSS-NP
SPLIT-VP
BASE-NP
DOMINATES-V
RIGHT-REC-NP

Linguistische Annotationen-Ablauf(2)

Externe vs. Interne Annotation

Tag Splitting

Annotation
Baseline ($v \leq 2, h \leq 2$)
UNARY-INTERNAL
UNARY-DT
UNARY-RB
TAG-PA
SPLIT-IN
SPLIT-AUX
SPLIT-CC
SPLIT-%
TMP-NP
GAPPED-S
POSS-NP
SPLIT-VP
BASE-NP
DOMINATES-V
RIGHT-REC-NP

Linguistische Annotationen-Ablauf(3)

Externe vs. Interne Annotation

Tag Splitting

In der Treebank enthaltenen Annotationen

Annotation
Baseline ($v \leq 2, h \leq 2$)
UNARY-INTERNAL
UNARY-DT
UNARY-RB
TAG-PA
SPLIT-IN
SPLIT-AUX
SPLIT-CC
SPLIT-%
TMP-NP
GAPPED-S
POSS-NP
SPLIT-VP
BASE-NP
DOMINATES-V
RIGHT-REC-NP

Linguistische Annotationen-Ablauf(4)

Externe vs. Interne Annotation

Tag Splitting

In der Treebank enthaltenen Annotationen

Head-Annotation

Annotation
Baseline ($v \leq 2, h \leq 2$)
UNARY-INTERNAL
UNARY-DT
UNARY-RB
TAG-PA
SPLIT-IN
SPLIT-AUX
SPLIT-CC
SPLIT-%
TMP-NP
GAPPED-S
POSS-NP
SPLIT-VP
BASE-NP
DOMINATES-V
RIGHT-REC-NP

Linguistische Annotationen-Ablauf(5)

Externe vs. Interne Annotation

Tag Splitting

In der Treebank enthaltenen Annotationen

Head-Annotation

„Distance“

Annotation
Baseline ($v \leq 2, h \leq 2$)
UNARY-INTERNAL
UNARY-DT
UNARY-RB
TAG-PA
SPLIT-IN
SPLIT-AUX
SPLIT-CC
SPLIT-%
TMP-NP
GAPPED-S
POSS-NP
SPLIT-VP
BASE-NP
DOMINATES-V
RIGHT-REC-NP

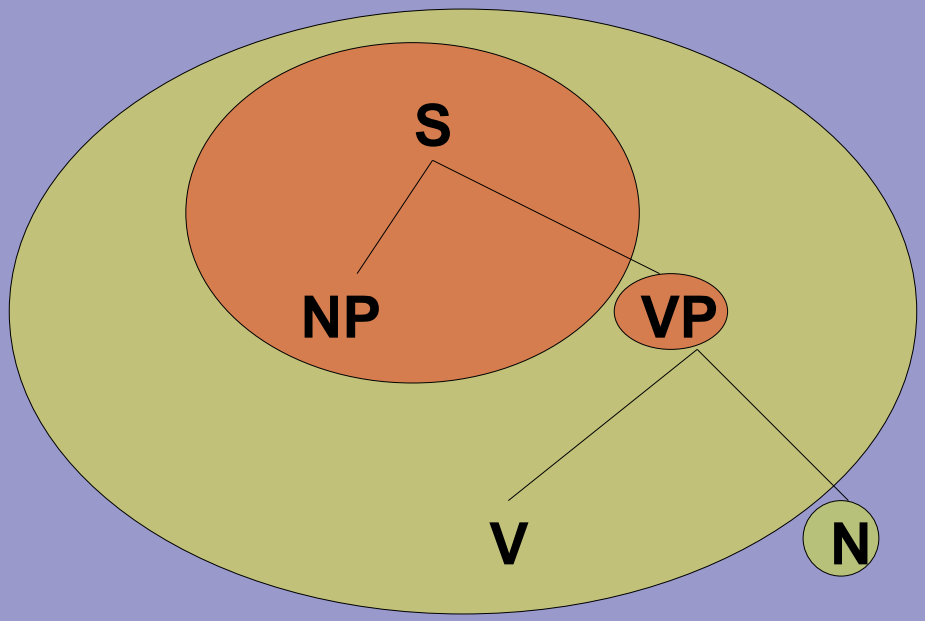
Externe vs. Interne Annotation(1)

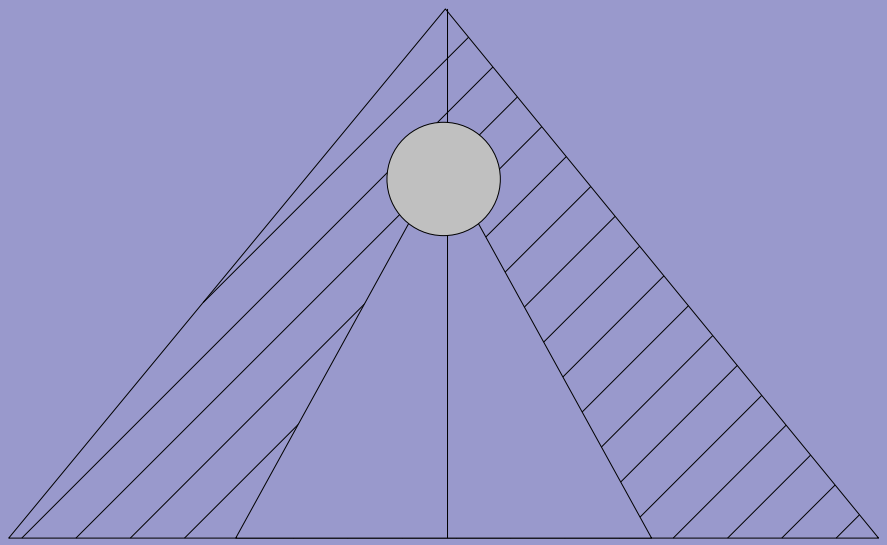
- Wir kennen bereits
  - Parent Annotation und
  - (Head) Lexicalization
- Diese Arten der Annotation können als Subklassen der
  - Externen Annotation (PA) und
  - Internen Annotation (Lex.)

  angesehen werden

Externe vs. Interne Annotation(2)

- Externe Umgebung eines Knotens...

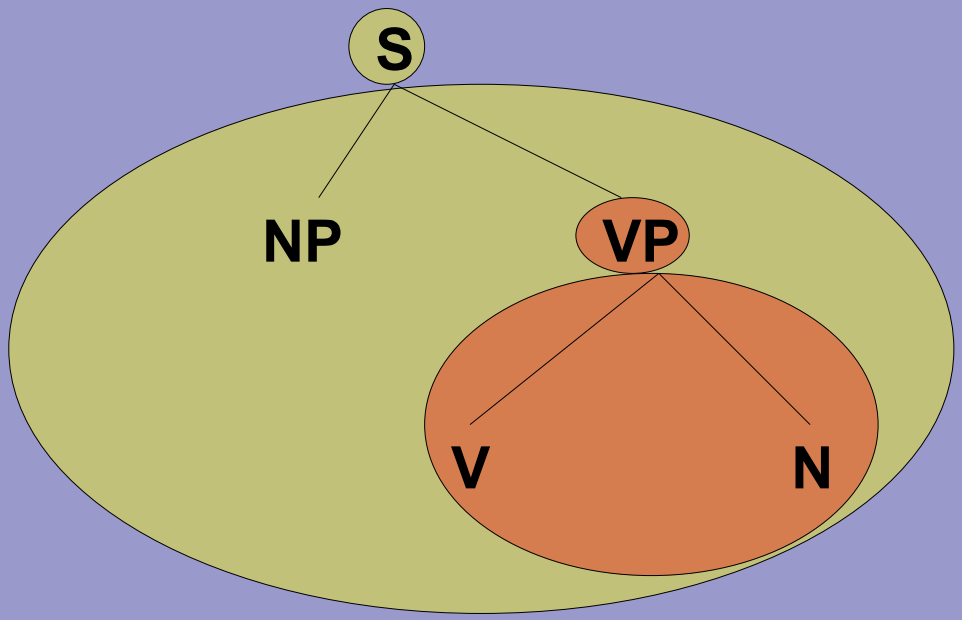


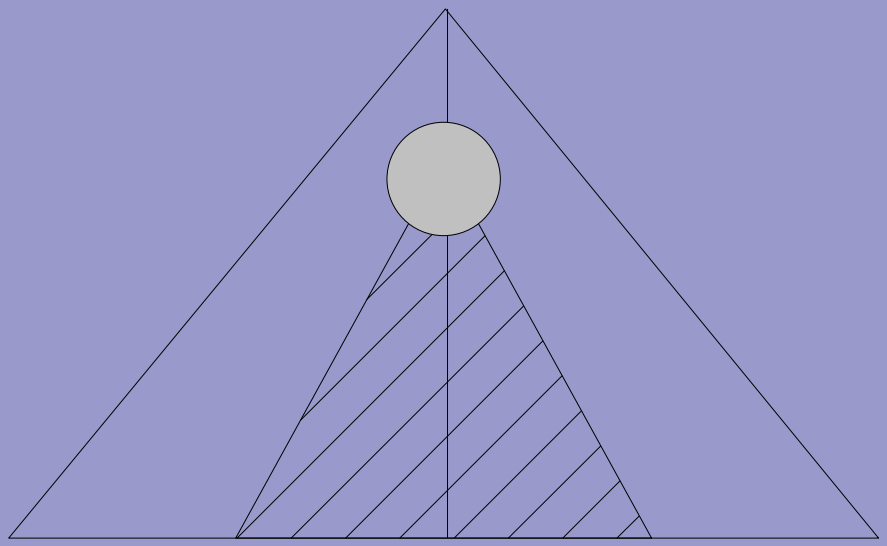


... Wird markiert durch ^x (vgl. Parent Annotation)...

Externe vs. Interne Annotation(3)

- Interne Umgebung eines Knotens....





... Wird markiert durch -x (vgl. Head Lexicalization).

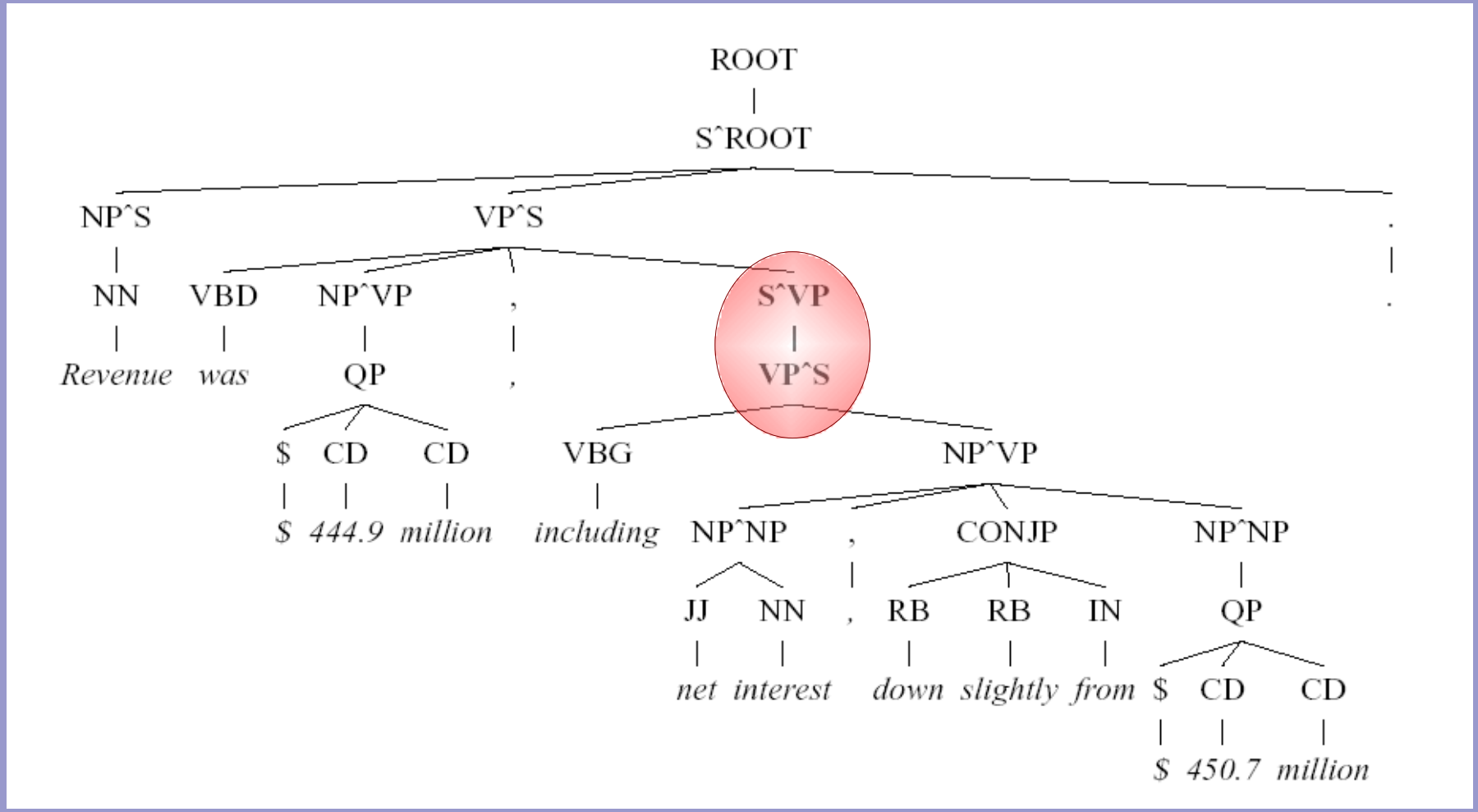
Externe vs. Interne Annotation(4)

- Problemfall: Unaries
  - Regeln der Form
    - X->Y
    - Y->X
  - Kettenbildung:

  - Sind allerdings praktisch sehr selten

Externe vs. Interne Annotation(5)

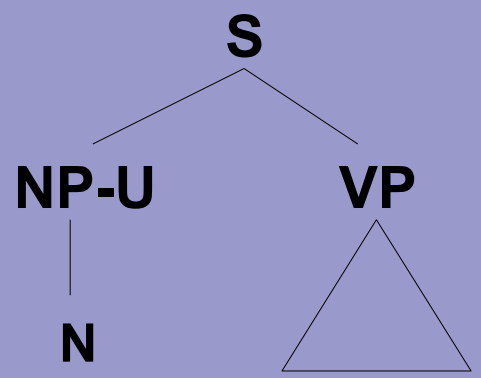
- Beispiel: Fehlerhafter Parse bei Unary-Regel



Externe vs. Interne Annotation: UNARY- INTERNAL

- Mögliche Lösung 1:
- UNARY-INTERNAL: Markierung aller Nonterminale, die nur ein Kind haben, mit

  -U
- Beispiel:

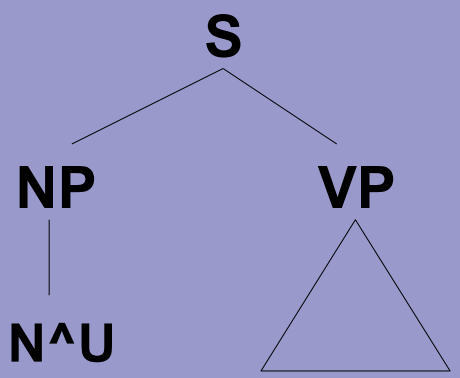


---> Lässt F_1 auf 78,32% steigen

Externe vs. Interne Annotation: „UNARY- EXTERNAL“?

- Mögliche Lösung 2:
- UNARY-EXTERNAL: Markierung aller Knoten, die keine Geschwister haben, mit

  ^U
- Beispiel:



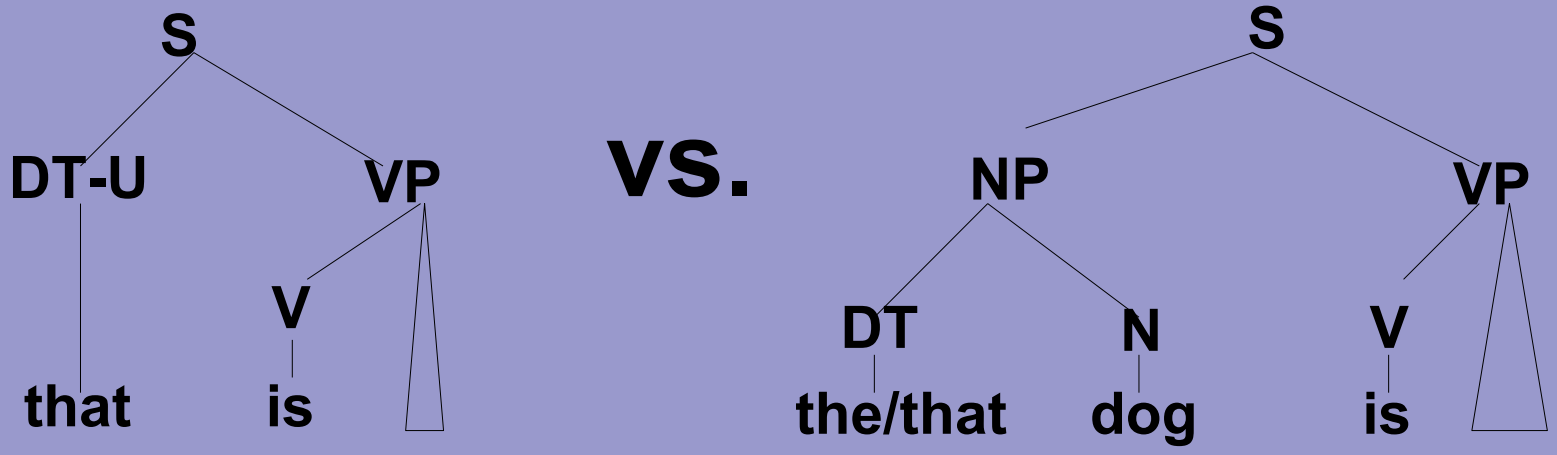
---> Hat alleine ähnlichen Effekt auf F_1 wie UNARY-INTERNAL, aber in Verbindung mit späteren Annotationen schlecht

---> Wird verworfen!

Externe vs. Interne Annotation: UNARY-DT

- Zusätzliche Unary – Annotationen:

- UNARY-DT: Zur Unterscheidung von Demonstrativen (that, those) und „echten“ Determinern (the, a)



Externe vs. Interne Annotation: UNARY-RB

- UNARY-RB: Zur Unterscheidung von Adverbien (z.B. „as well“ (zwei Präterminale!) vs. „also“)

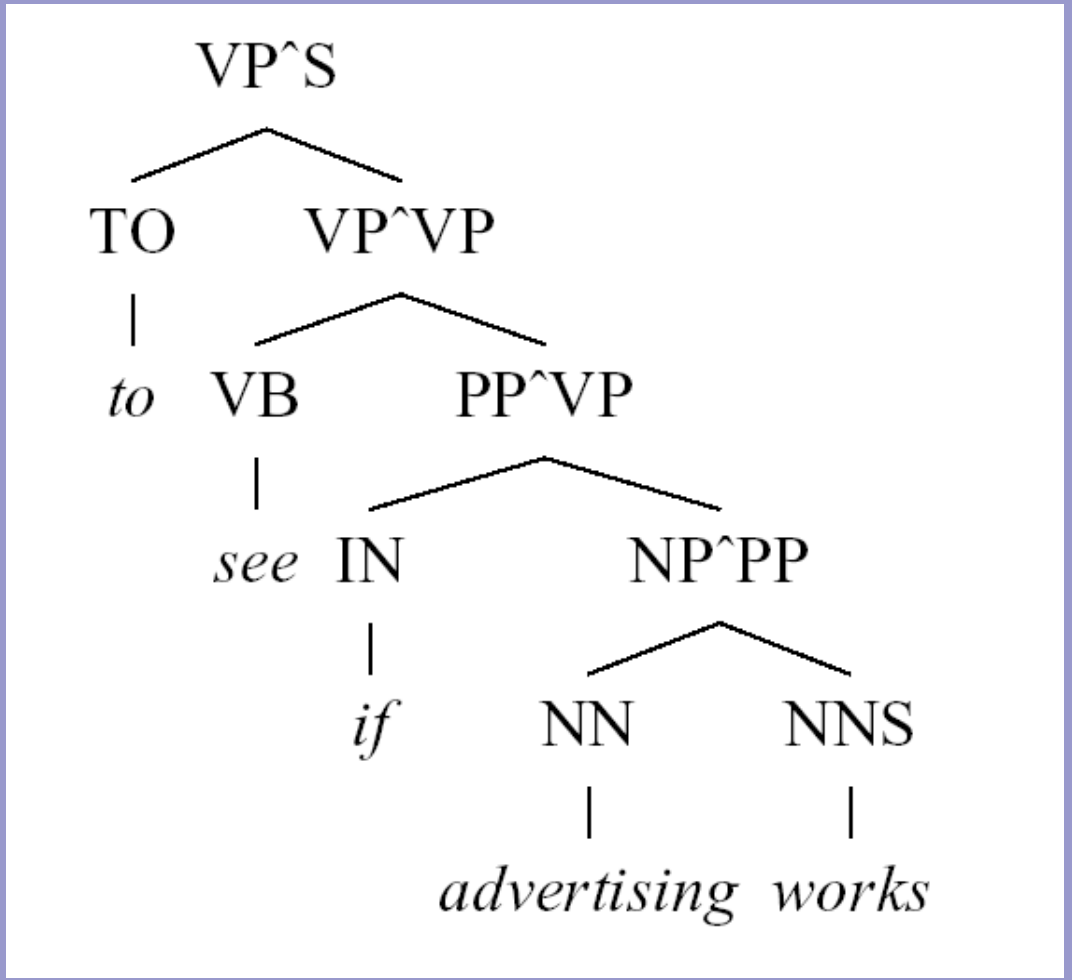
---> Nach UNARY-DT und UNARY-RB steht F_1 bei 78,86%

Tag Splitting: TAG-PA(1)

- Beobachtung: Parent Annotation bei Nonterminalen ist sehr effektiv (vgl. UNARY-DT)
- Annahme: Parent Annotation ist auch bei Tags sinnvoll

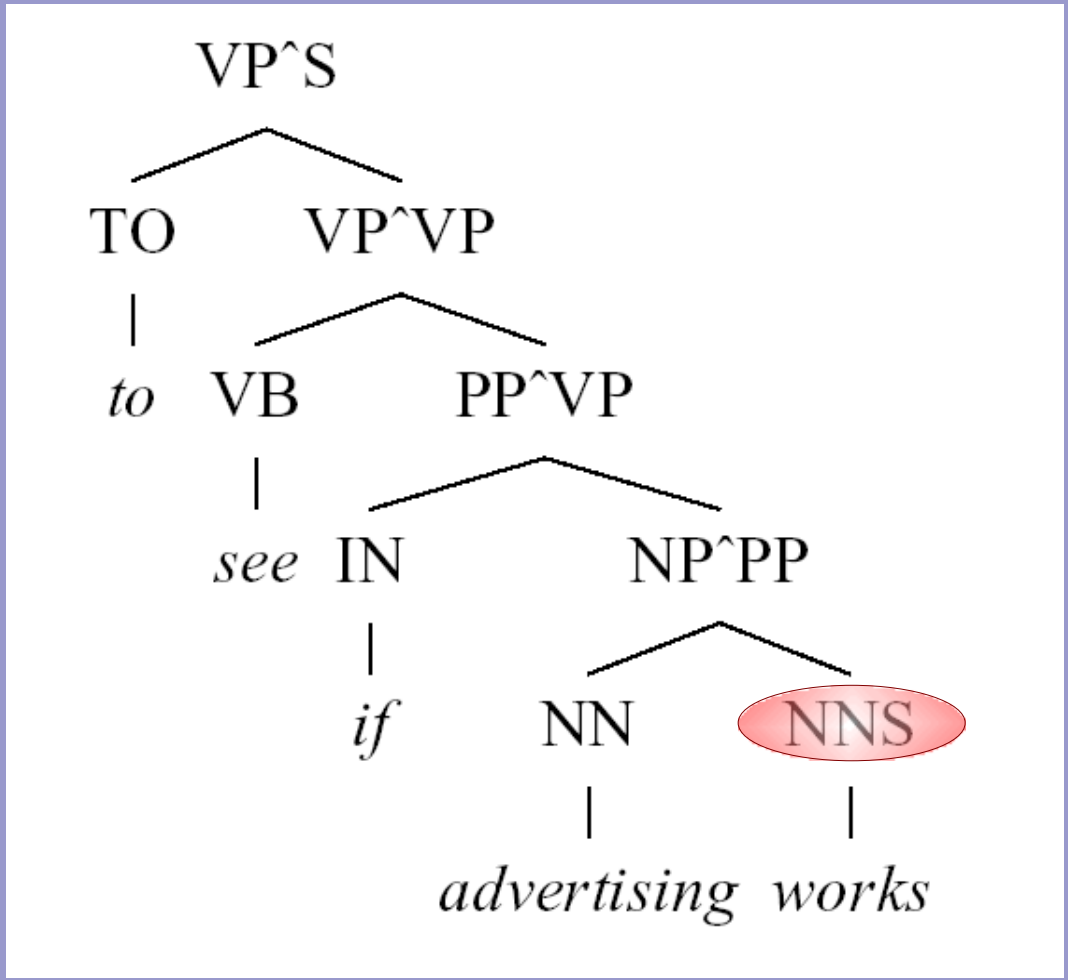
Tag Splitting: TAG-PA(2)

- Fehlerhafter Parse



Tag Splitting: TAG-PA(3)

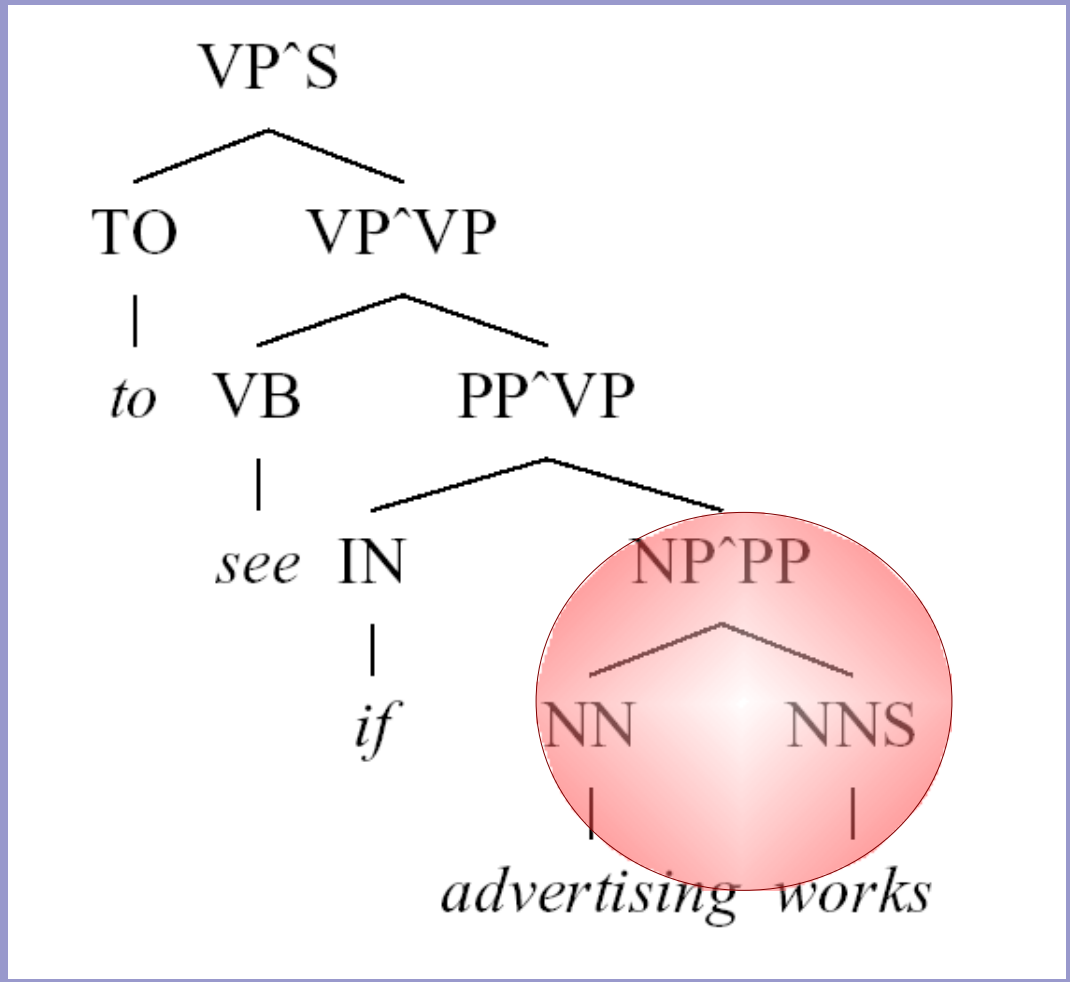
- Fehlerhafter Parse



- Grundsätzlicher Fehler bereits an dieser Stelle:

Tag Splitting: TAG-PA(4)

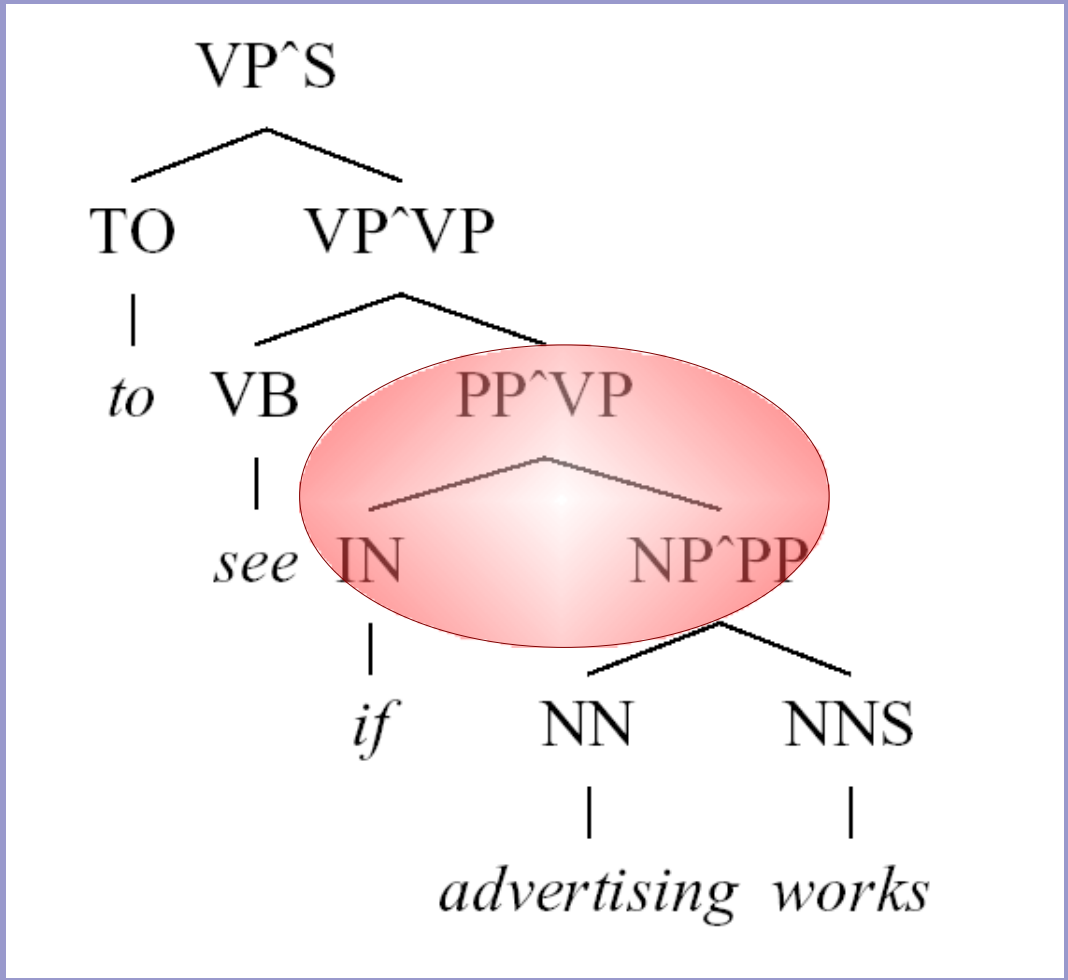
- Fehlerhafter Parse



- Beobachtung: NNS-Tags stehen meistens unter NPs

Tag Splitting: TAG-PA(5)

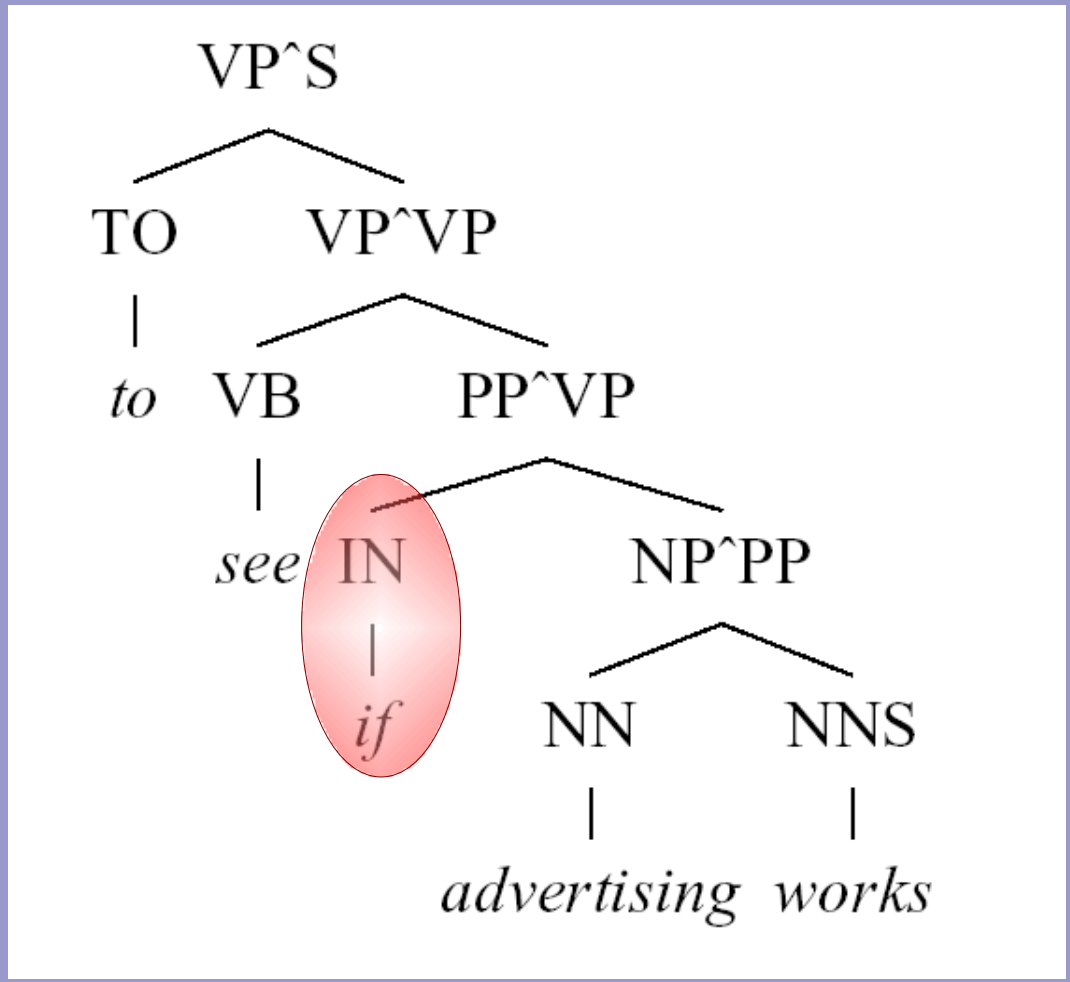
- Fehlerhafter Parse



- ... PP->IN NP am Wahrscheinlichsten

Tag Splitting: TAG-PA(6)

- Fehlerhafter Parse

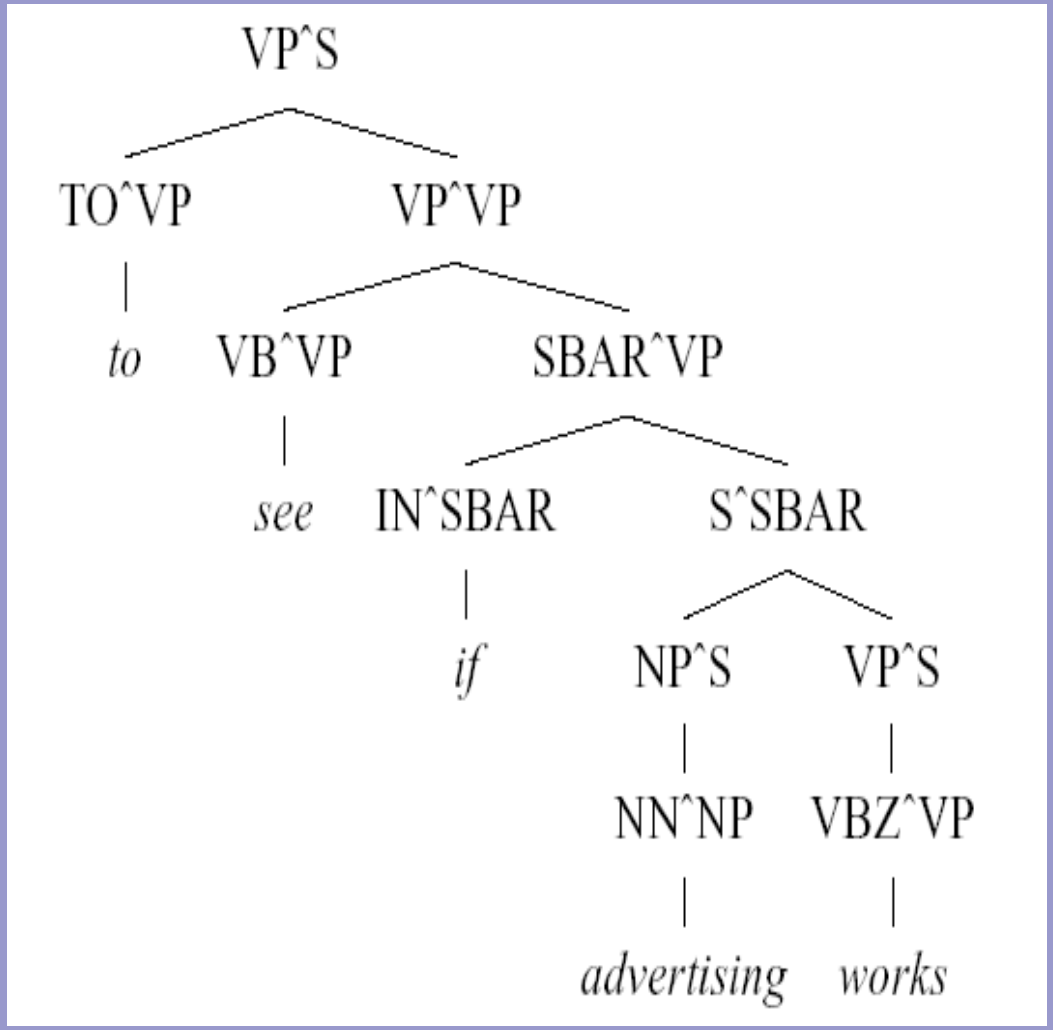


- *Lösung:*

  If muss auch S-Komplemente bevorzugen können!

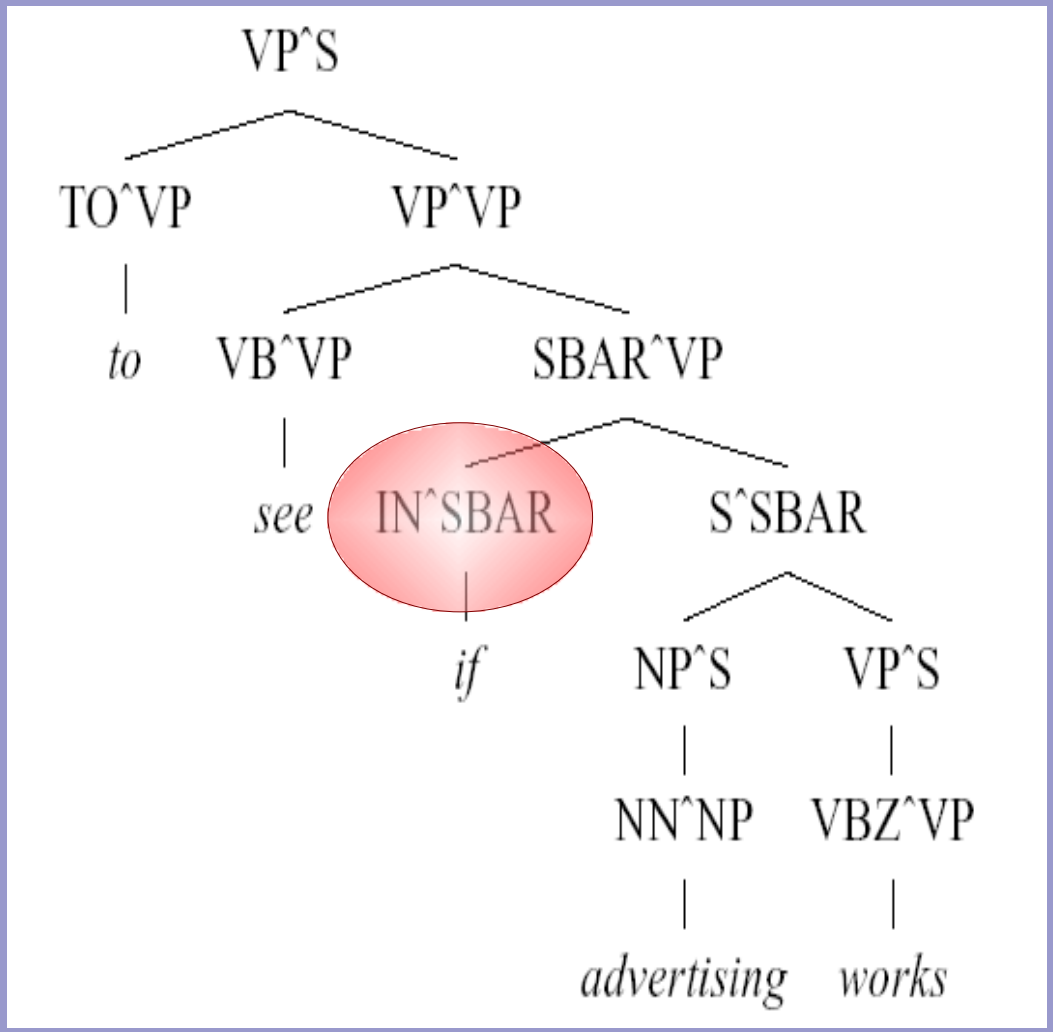
Tag Splitting: TAG-PA(7)

- Lösung durch Parent Annotation der Tags



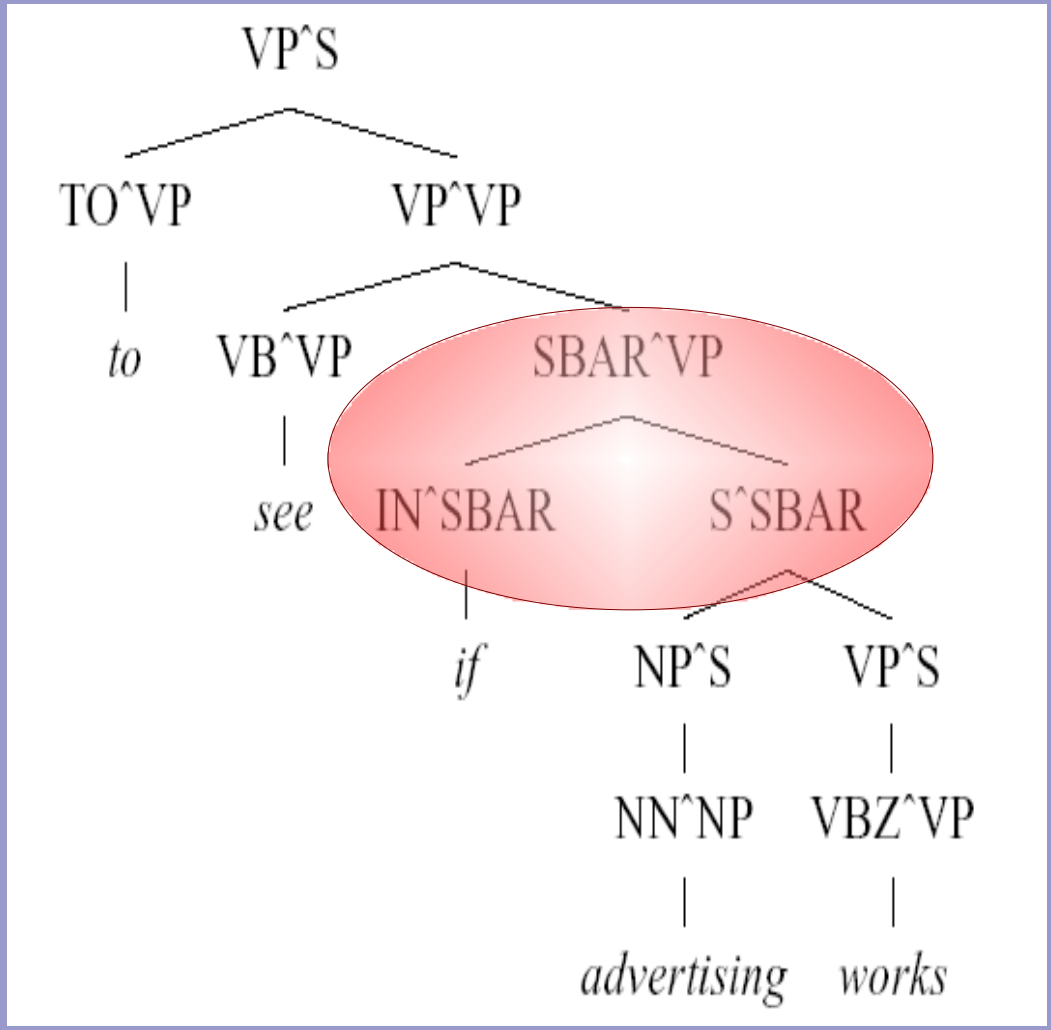
Tag Splitting: TAG-PA(8)

- Lösung durch Parent Annotation der Tags
- *If* bevorzugt jetzt SBARs...



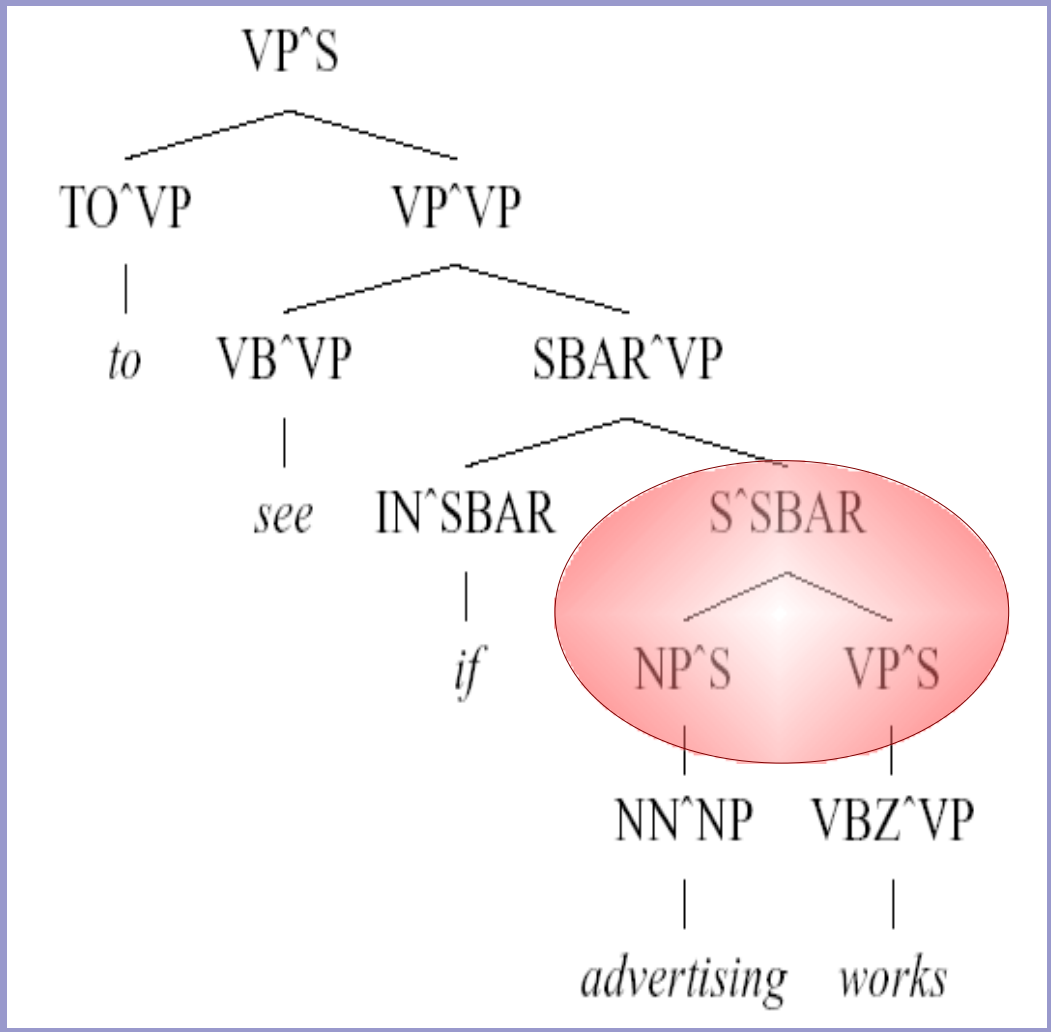
Tag Splitting: TAG-PA(9)

- Lösung durch Parent Annotation der Tags
- SBAR-> IN S am Wahrscheinlichsten...



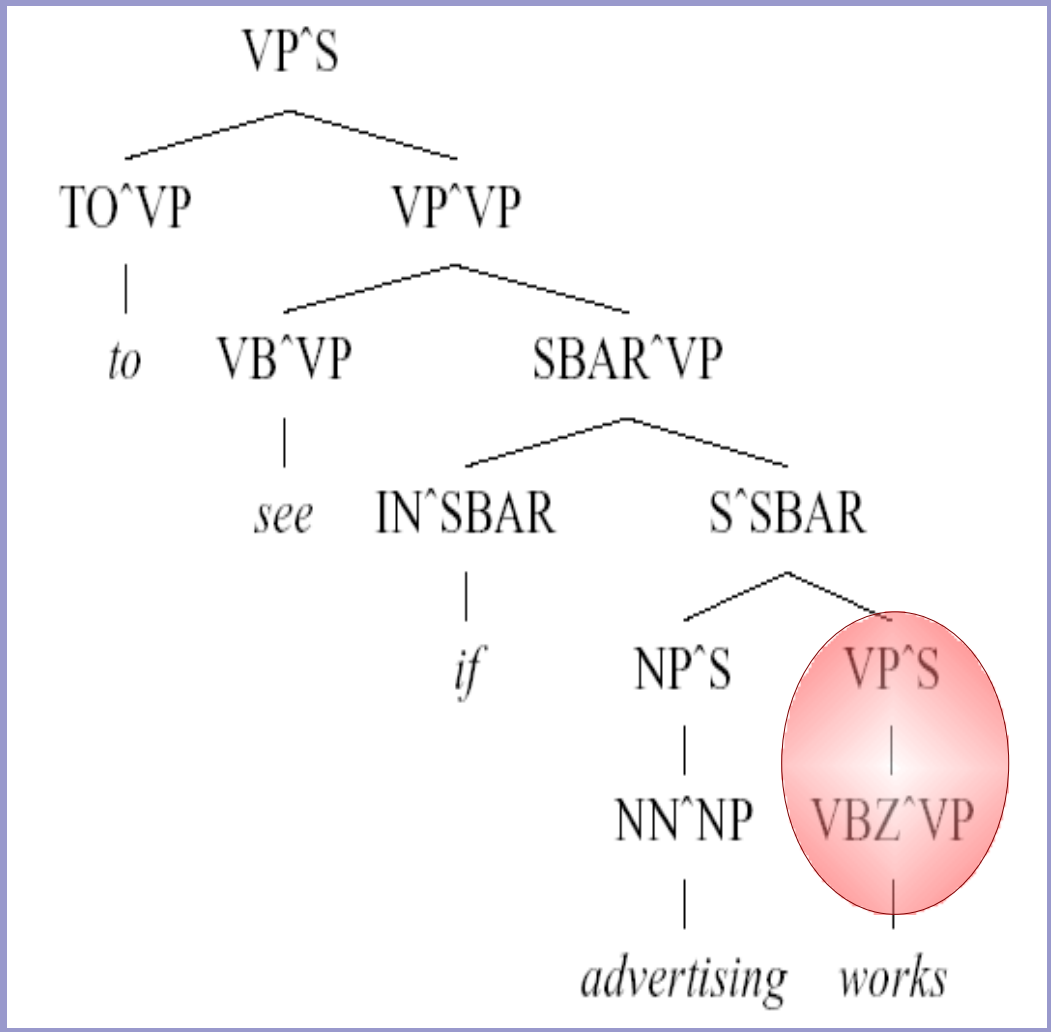
Tag Splitting: TAG-PA(10)

- Lösung durch Parent Annotation der Tags
- S-> NP VP am Wahrscheinlichsten...



Tag Splitting: TAG-PA(11)

- Lösung durch Parent Annotation der Tags



- ... *works* wird erkannt als VBZ (=Verb, 3. Pers. Präsens Singular)

--->F_1 steigt nach TAG-PA auf 80,62%

Tag Splitting: SPLIT-IN(1)

- Das Penn Treebank POS Tagset enthält nicht jede Kategorie, die unterschieden werden kann.
  - Hier: Tag „IN“ steht für
    - Subordinierende Konjunktionen (while, as, if)
    - Complementizer (that, for)
    - Präpositionen (of, in, from)
    - Ausschließlich Nomen/Verb-definierende Präpositionen (of/as)
    - ...
- Das Wissen über diese Kategorien wird wegen der unterschiedlichen Distributionen der entsprechenden Nonterminale aber benötigt

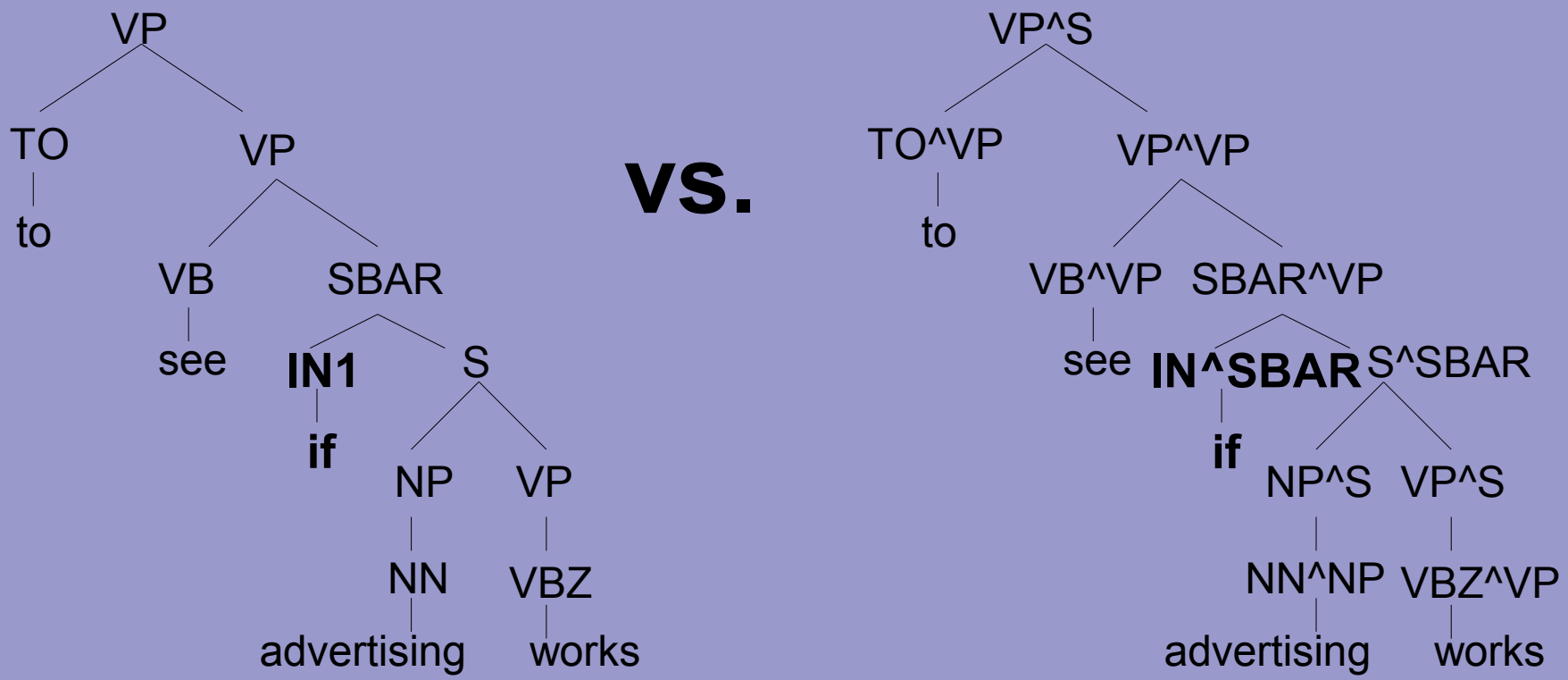
Tag Splitting: SPLIT-IN(2)

- Lösung:
- Der IN-Tag wird in insgesamt 6 Teile zerlegt
    - Etwa IN1, IN2, IN3, IN4, IN5, IN6
- ... entsprechend der Kategorie mit unterschiedlicher Distribution (z.B. IN1 für subordinierende Konjunktionen, IN2 für Complementizer etc.)

---> Bringt F_1 auf 81,19%

Tag Splitting: TAG-PA vs. SPLIT-IN(1)

Der fehlerhafte Parse aus dem Beispiel für TAG-PA kann auch durch SPLIT-IN gelöst werden:



Tag Splitting: TAG-PA vs. SPLIT-IN(2)

- TAG-PA und SPLIT-IN kollidieren
- Siehe Beispiel:
    - IN^SBAR **vs.** IN1
- Lösung:
    - Parser prüft nach TAG-PA, ob ein Tag-Parent „SBAR“ besteht
    - Für diese Fälle ist SPLIT-IN nicht zuständig

Tag-Splitting: SPLIT-AUX

- Subkategorisierung von Tags für spezifische Lexeme – 1. SPLIT-AUX
- „Partial Auxilliary Annotation“ gibt es auch bei Charniak (1997)
  - Alle Hilfsverben werden mit AUX markiert
  - Alle Gerund-Formen (Auxiliaries und VPs) werden zusätzlich mit -G markiert
- Klein & Manning hängen
  - ^BE an alle Formen von *be*
  - ^HAVE an alle Formen von *have*

---> F_1 steigt auf 81,66%

Tag-Splitting: SPLIT-CC

- Subkategorisierung von Tags für spezifische Lexeme – 2. SPLIT-CC
- Beobachtung: Die Strings „[Bb]ut“ und „&“ haben andere Distribution als andere Konjunktionen
    - „[Bb]ut“ vor allem auf Satzebene, „&“ in Firmennamen

---> F_1 steigt auf 81,69%

Tag-Splitting: SPLIT-%

- Subkategorisierung von Tags für spezifische Lexeme – 3. SPLIT-%
- Das Prozentzeichen „%“ erhält seinen eigenen Tag (Das Dollarzeichen „$“ besitzt bereits einen eigenen Tag im Penn Tag Set)

---> F_1 steigt auf 81,81%

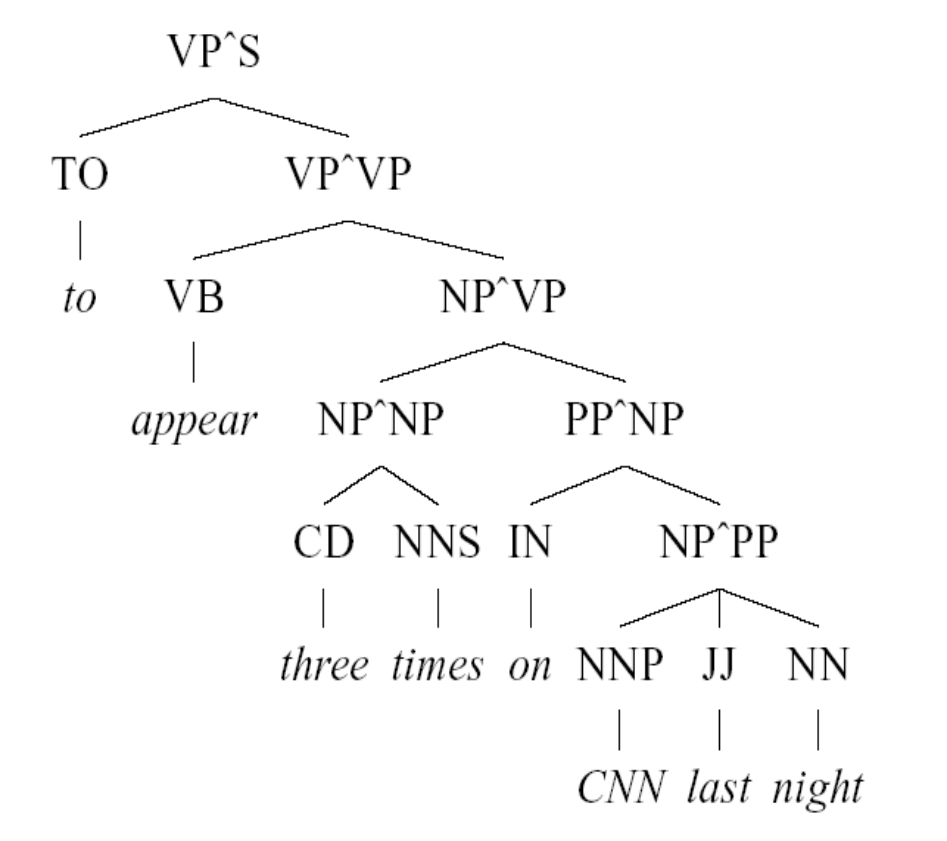
In der Treebank enthaltene Annotationen(1)

- In der Penn Treebank sind bereits Functional Tags enthalten
- Klein&Manning:
  - Entfernen die Treebank-eigenen Annotationen
  - Fügen Annotationen der gleichen Art manuell ein
- Warum?

In der Treebank enthaltene Annotationen(2)

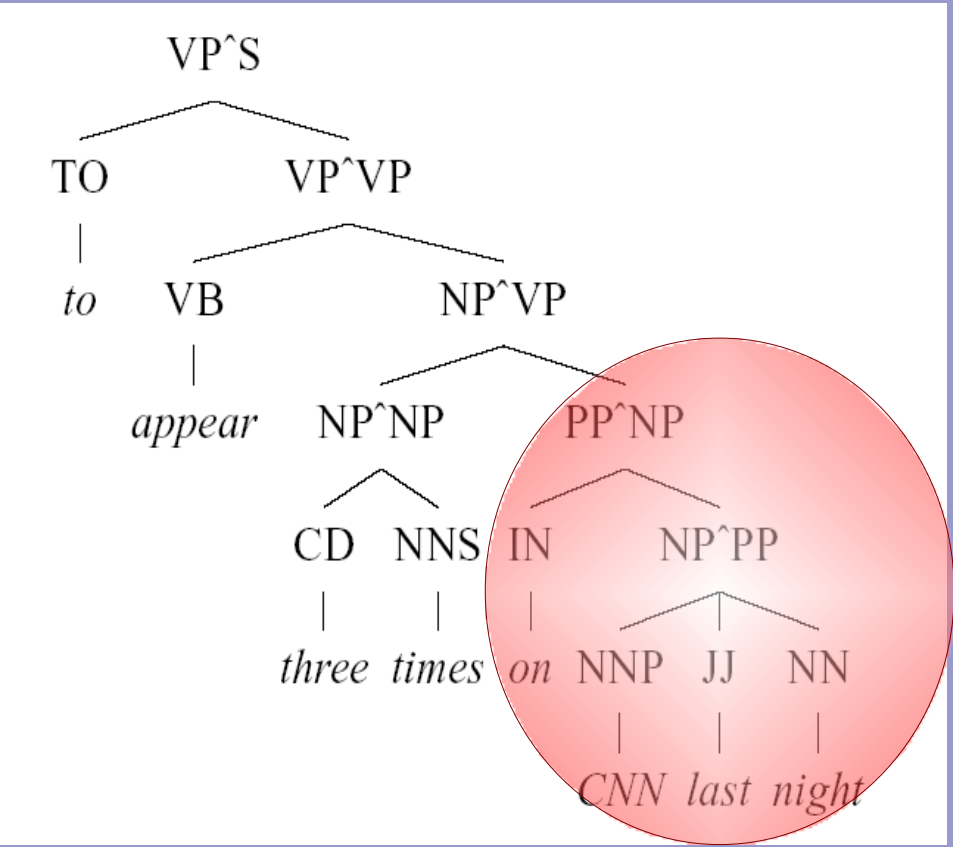
- Großteil der Treebank-eigenen Functional Tags verschlechtert die Performanz des Parsers
    - Beispiele:
        - PP-LOC; ADV-TMP
- Ausgangswert F_1=72,62% sinkt auf 71,49%
- Ausgangswert (mit h ≤2,v ≤1) von F_1=77,77% sinkt sogar auf 72,87%
- ---> Manche Treebank-Tags dennoch sinnvoll:

In der Treebank enthaltene Annotationen: TMP-NP(1)



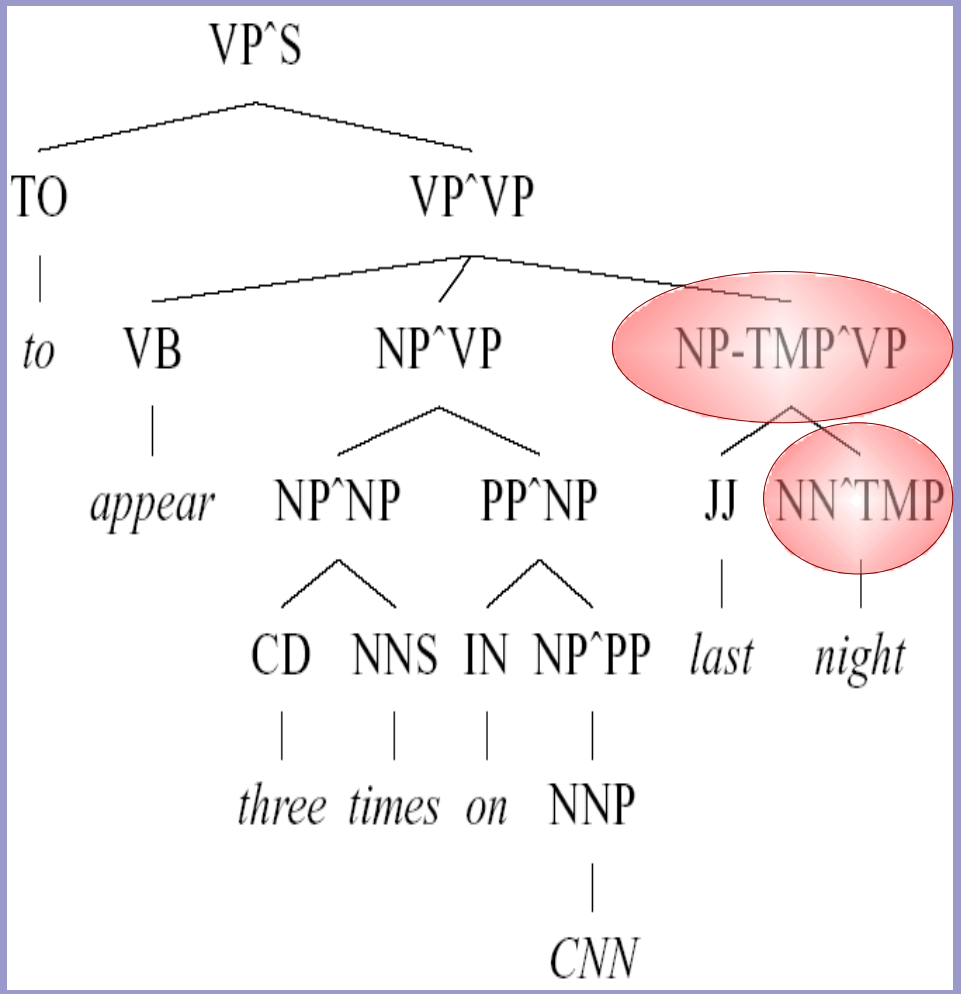
- Fehlerhafter Parse

In der Treebank enthaltene Annotationen: TMP-NP(2)



- Fehlerhafter Parse
- Disambiguierung der Regel kann wieder durch linguistisches Wissen erfolgen: Temporale NPs haben andere Distribution als andere NPs

In der Treebank enthaltene Annotationen: TMP-NP(3)



- Korrektur durch Annotation von temporalen Nomen und NPs

---> TMP-NP Lässt F_1 auf 82,25% steigen

In der Treebank enthaltene Annotationen: GAPPED-S

- Nach Collins(1999)
- Markieren von S- Knoten, die leeres Subjekt haben

--->Lässt F_1 auf 82,28% steigen

Head-Annotation

- Annotation des Kopf-Wortes ist bereits sinnvoll...
- ...Annotation des Kopf-Tags ist besser:
    - Kopf-Tag sagt viel über das Verhalten der Konstituente aus

Head-Annotation: POSS-NP

- Possesive NPs haben von anderen NPs abweichende Distribution
- Beispiel: Regeln wie NP->NPα treten nur bei possessiven NPs auf

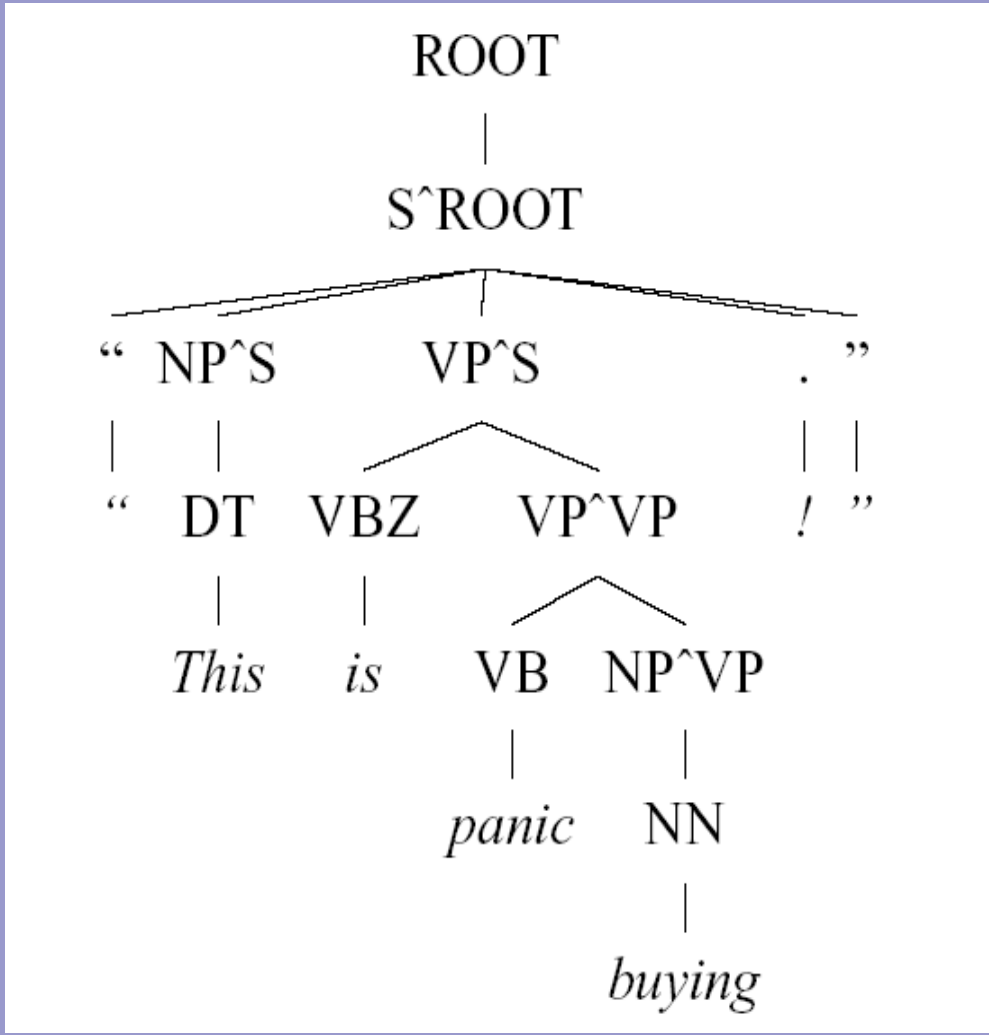
---> Lässt F_1 auf 83,06% steigen

Head-Annotation: SPLIT-VP(1)

- In der Penn Treebank wird nicht zwischen finiten und non-finiten VPs unterschieden
- Linguistisches Wissen: Verben im Präsens nehmen keinen nackten Infinitiv-VP-Komplemente

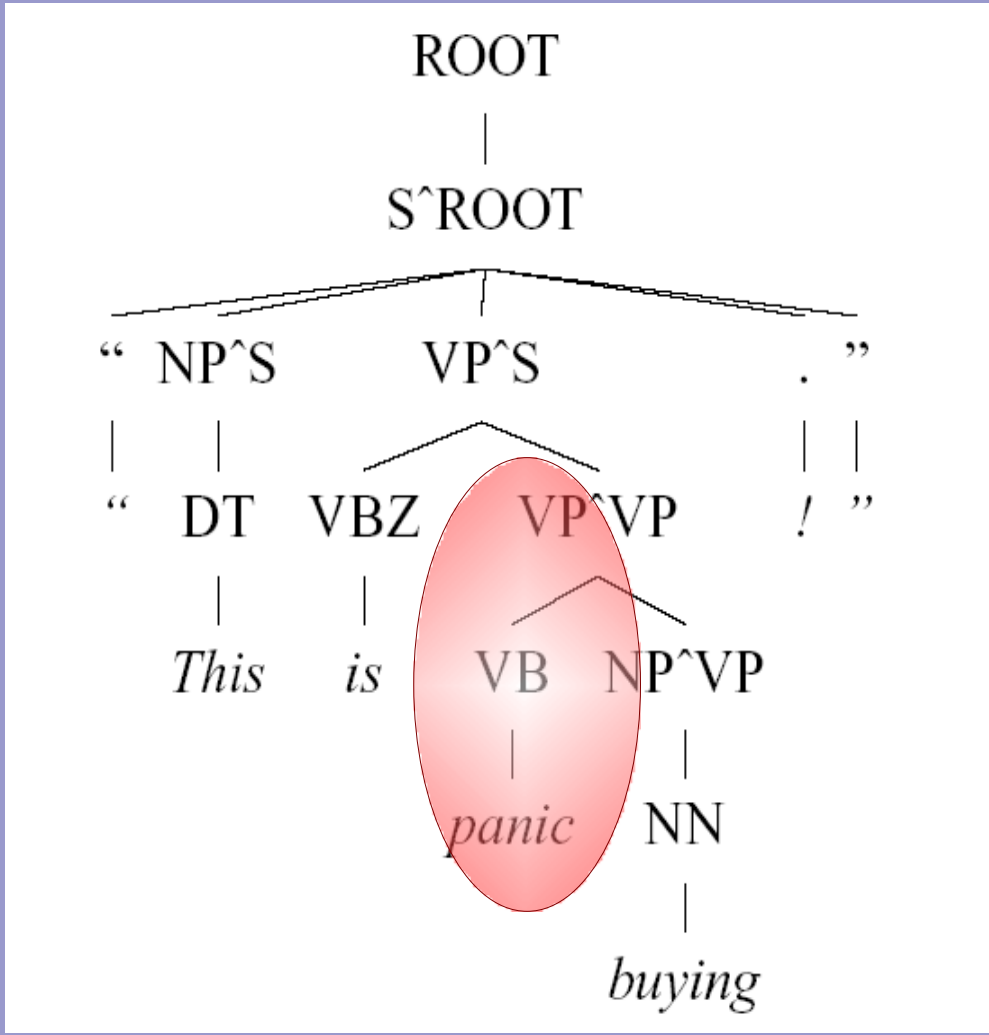
- SPLIT-VP: Alle VP-Knoten werden mit ihrem Kopf annotiert;
  -VBF markiert alle finiten Formen

Head-Annotation: SPLIT-VP(2)



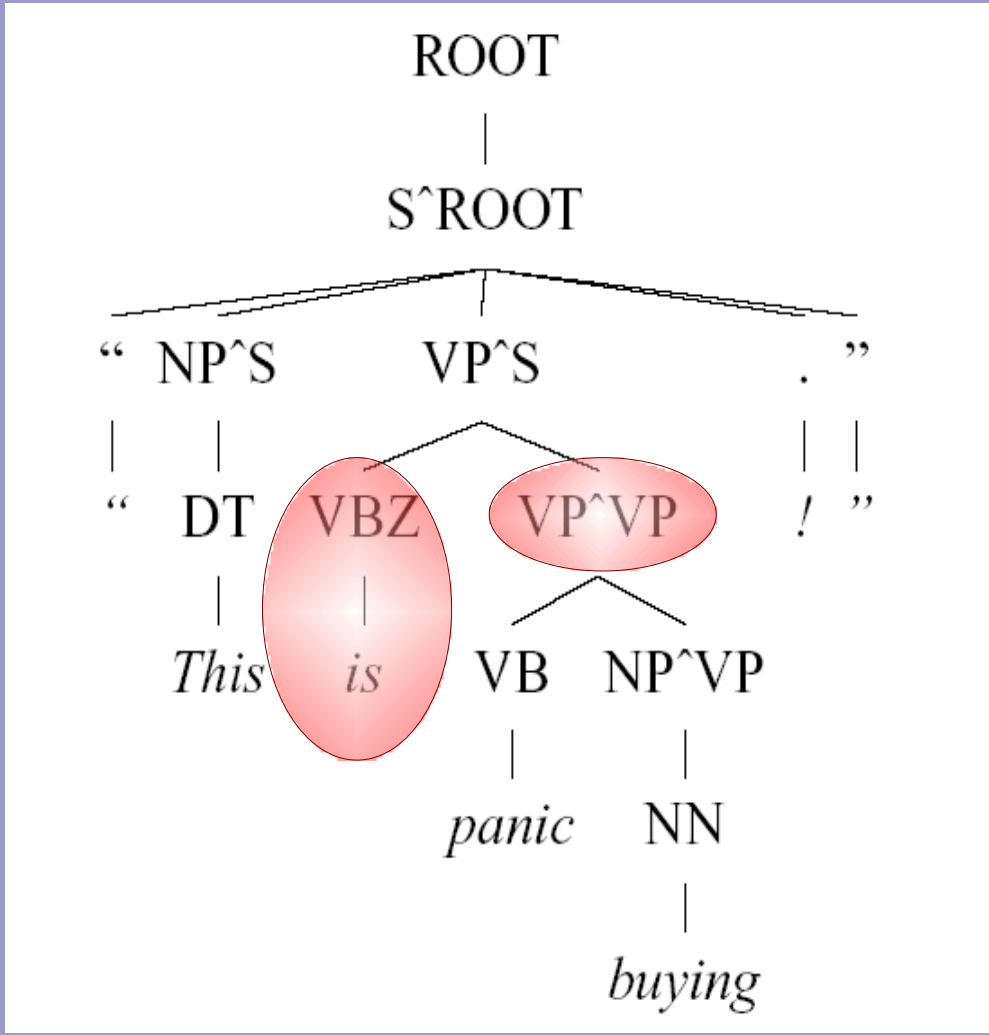
- Fehlerhafter Parse

Head-Annotation: SPLIT-VP(3)



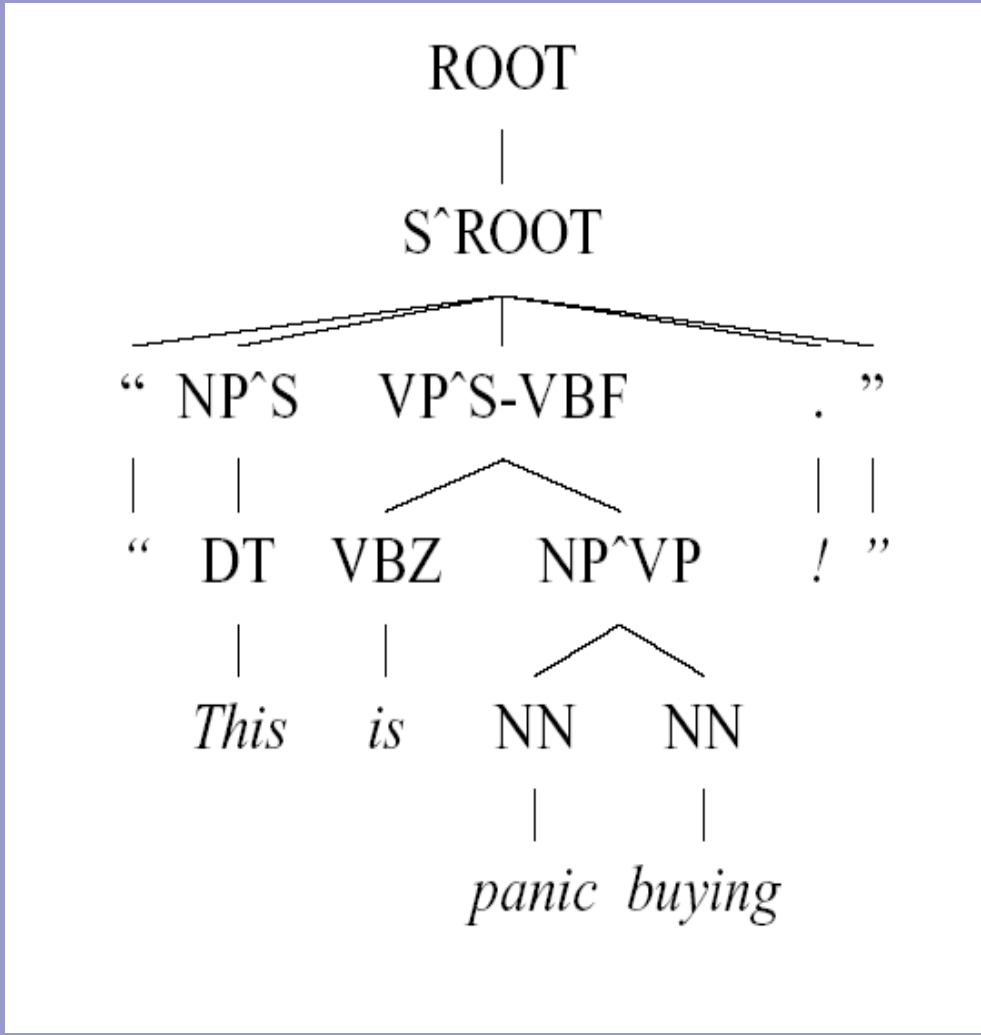
- Fehlerhafter Parse
- *Panic* entsprechend höchster Wahrscheinlichkeit (mit NN als Geschwister) als VP erkannt

Head-Annotation: SPLIT-VP(4)



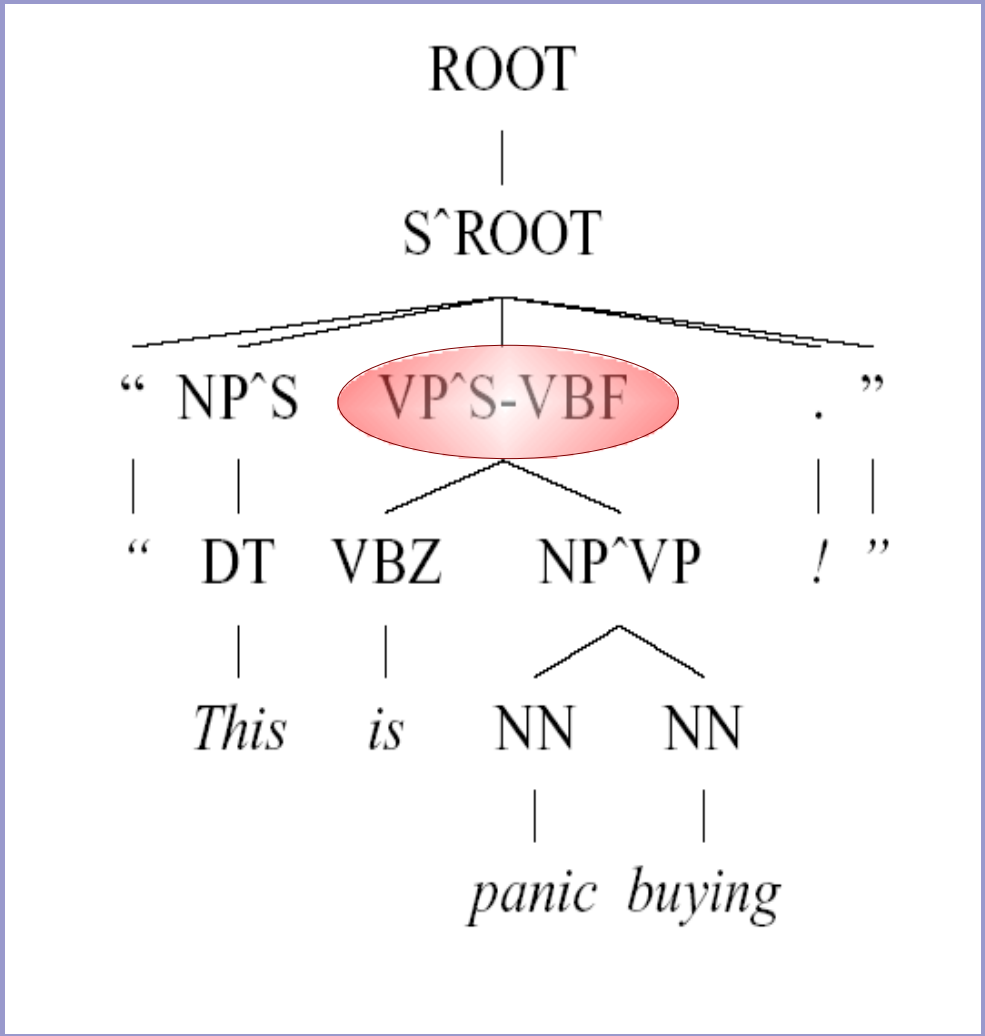
- Fehlerhafter Parse
- Zugrundeliegender Fehler: Finites Verb kann kein VP-Infinitiv-Komplement haben

Head-Annotation: SPLIT-VP(5)



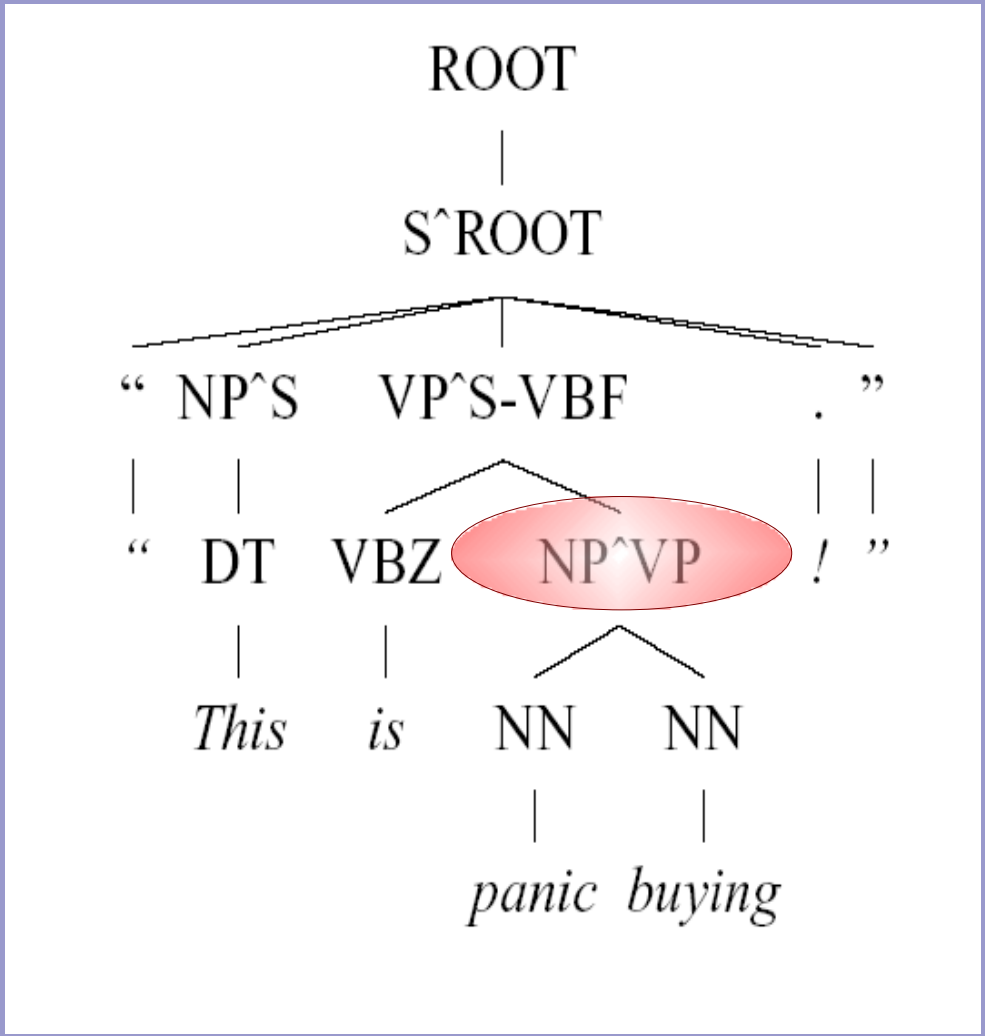
- Lösung durch SPLIT-VP

Head-Annotation: SPLIT-VP(6)



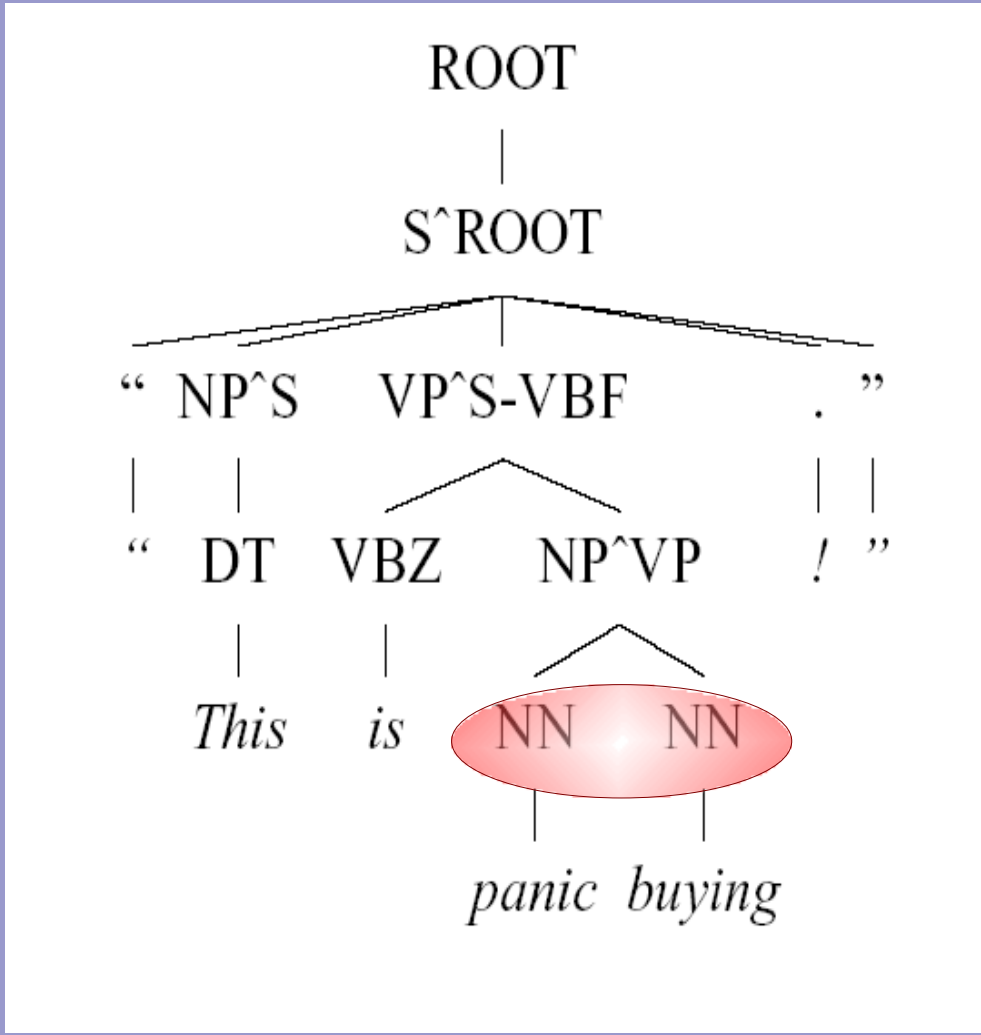
- Lösung durch SPLIT-VP
- Annotierung des VP-Knotens mit der Kategorie seines Kopfes (=finites Verb)

Head-Annotation: SPLIT-VP(7)



- Lösung durch SPLIT-VP
- Durch linguistisches Wissen wird jetzt VP->VBZ VP unmöglich

Head-Annotation: SPLIT-VP(8)



- Lösung durch SPLIT-VP
- NP->NN NN
  (NP->VB NN ist jetzt unmöglich)

--->SPLIT-VP bringt F_1 auf 85,72%

„Distance“

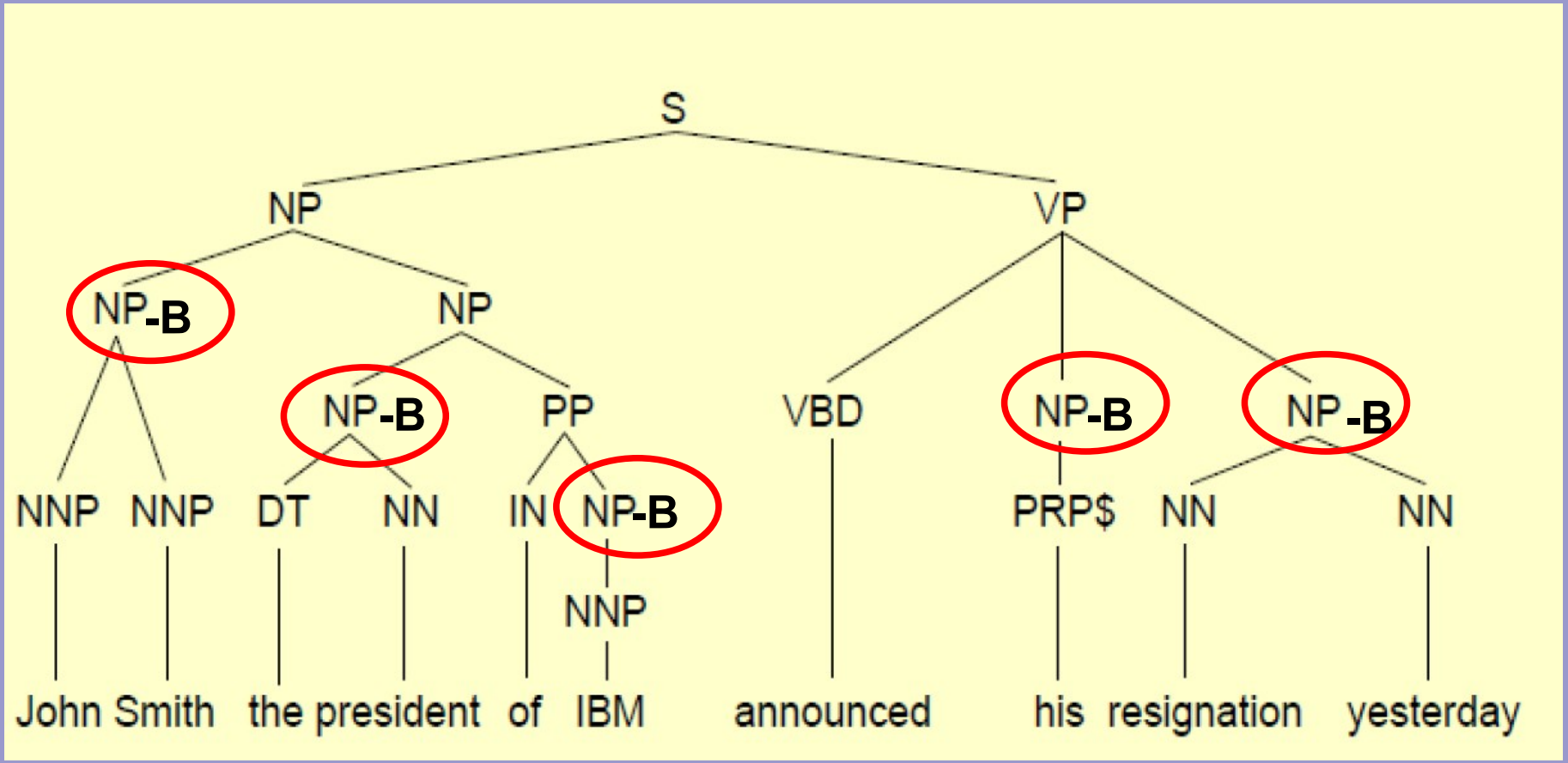
- Fehleranalyse an diesem Punkt zeigt, dass die meisten Fehler beim Attachment Level und Skopus der Konjunktionen liegen
- Collins verwendet dazu Distance
- Distance ist für eine PCFG schwer zu encoden (jeden Knoten mit seinen Argumenten zu markieren ist sehr speicheraufwändig)
- -> Klein&Manning verwenden drei indirekte Annotationen: BASE-NP, DOMINATES-V und RIGHT-REC-NP

„Distance“: BASE-NP(1)

- Vgl. Collins'(1999) base NP
- Markierung von nicht-rekursiven NPs mit -B
- Achtung: Unaries werden hier nicht markiert

„Distance“: BASE-NP(2)

Beispiel (Vgl. Collins (1999)):



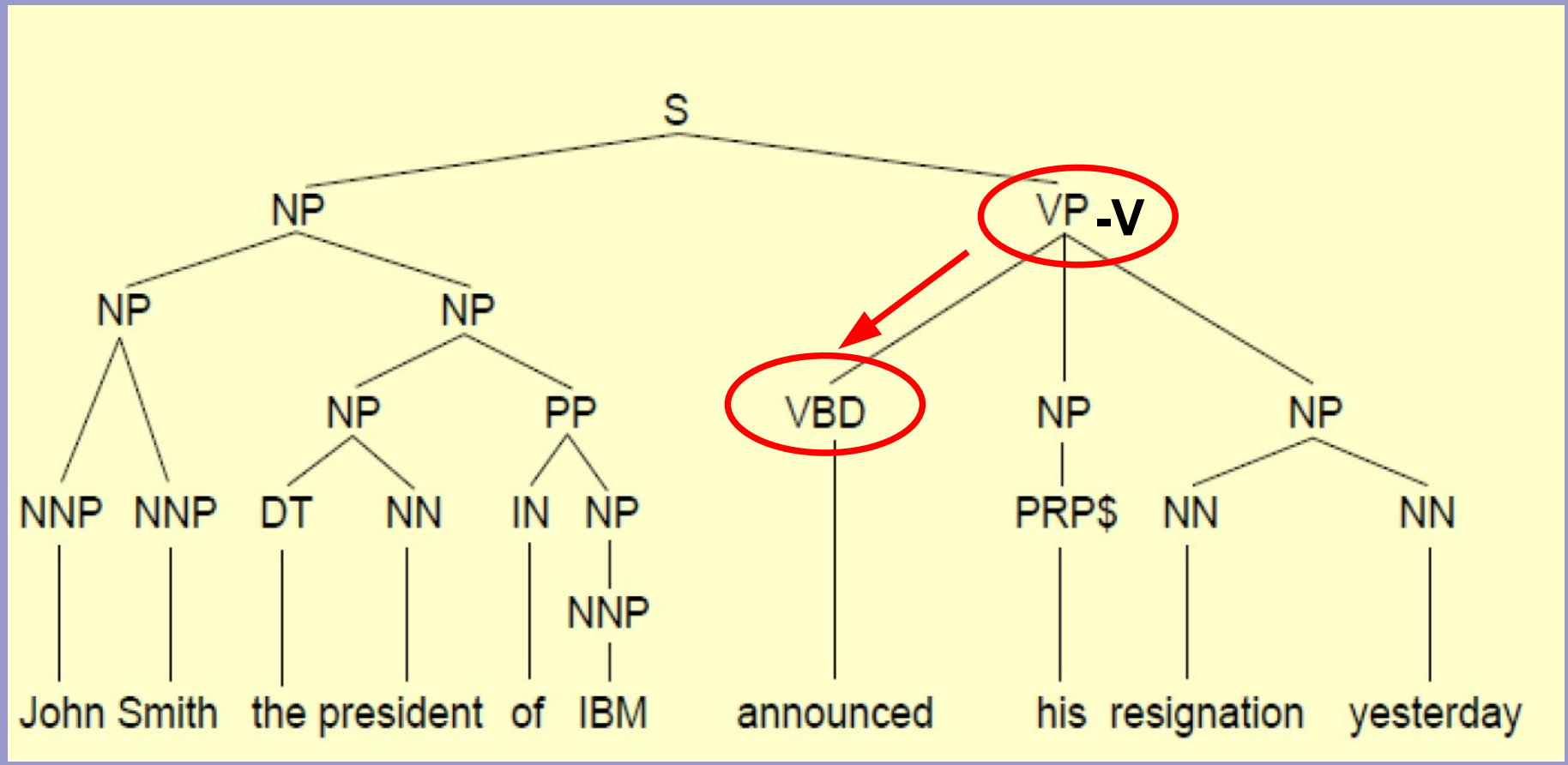
---> F_1 steigt auf 86,04%

„Distance“: DOMINATES-V(1)

- Markierung aller Knoten, die einen Verbknoten (V*, MD) dominieren, mit -V
- Auch bei Collins(1999), als „Verbal Distance“ umgesetzt

„Distance“: DOMINATES-V(2)

Beispiel:



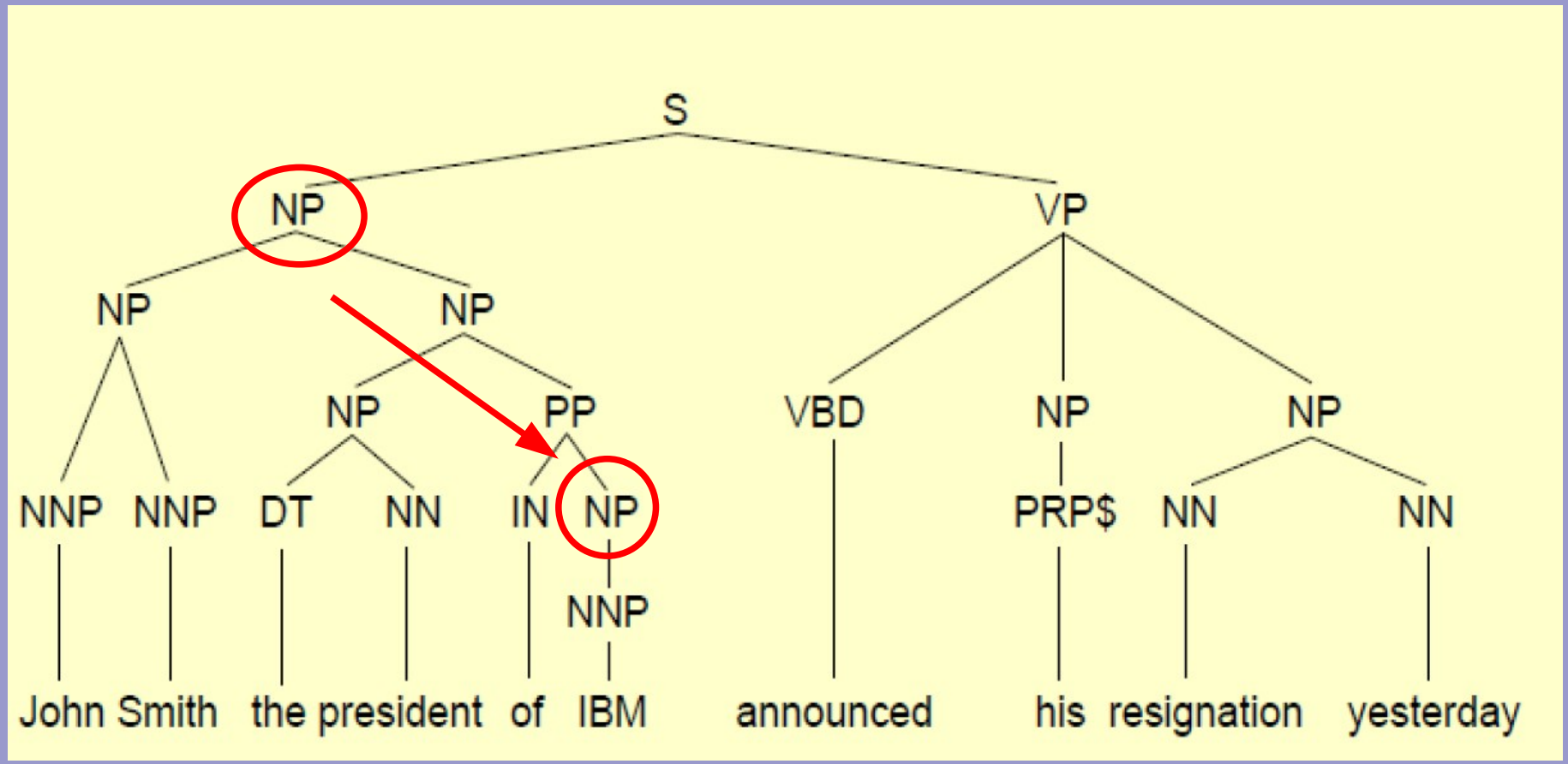
---> F_1 steigt auf 86,91%

„Distance“: RIGHT-REC-NP(1)

- Markierung aller NPs, die eine NP in ihrer rechten Peripherie (= als rechtestes Kind) haben

„Distance“: RIGHT-REC-NP(2)

Beispiel:



---> F_1 steigt auf 87,04%

Ergebnisse

Wiederholung: Ergebnisse nach Linguistischer Annotation(1)

	Cumulative			Indiv.
Annotation	Size	F_1	ΔF_1	ΔF_1
Baseline ($v \leq 2, h \leq 2$)	7619	77.77	–	–
UNARY-INTERNAL	8065	78.32	0.55	0.55
UNARY-DT	8066	78.48	0.71	0.17
UNARY-RB	8069	78.86	1.09	0.43
TAG-PA	8520	80.62	2.85	2.52
SPLIT-IN	8541	81.19	3.42	2.12
SPLIT-AUX	9034	81.66	3.89	0.57
SPLIT-CC	9190	81.69	3.92	0.12
SPLIT-%	9255	81.81	4.04	0.15
TMP-NP	9594	82.25	4.48	1.07
GAPPED-S	9741	82.28	4.51	0.17
POSS-NP	9820	83.06	5.29	0.28
SPLIT-VP	10499	85.72	7.95	1.36
BASE-NP	11660	86.04	8.27	0.73
DOMINATES-V	14097	86.91	9.14	1.42
RIGHT-REC-NP	15276	87.04	9.27	1.94

Wiederholung: Ergebnisse nach Linguistischer Annotation(2)

	Cumulative			Indiv.
Annotation	Size	F_1	ΔF_1	ΔF_1
Baseline ($v \leq 2, h \leq 2$)	7619	77.77	–	–
UNARY-INTERNAL	8065	78.32	0.55	0.55
UNARY-DT	8066	78.48	0.71	0.17
UNARY-RB	8069	78.86	1.09	0.43
TAG-PA				
SPLIT-IN				
SPLIT-AUX				
SPLIT-CC				
SPLIT-%				
TMP-NP				
GAPPED-S				
POSS-NP				
SPLIT-VP				
BASE-NP				
DOMINATES-V	14097	86.91	9.14	1.42
RIGHT-REC-NP	15276	87.04	9.27	1.94

Fast 10% Verbesserung von F_1!

Verhältnis zu anderen Parsern(1)

Test mit Sektion 23 der Penn Treebank:

Length $\leq$ 40	LP	LR	F_1	Exact	CB	0 CB
Magerman (1995)	84.9	84.6			1.26	56.6
Collins (1996)	86.3	85.8			1.14	59.9
this paper	86.9	85.7	86.3	30.9	1.10	60.3
Charniak (1997)	87.4	87.5			1.00	62.1
Collins (1999)	88.7	88.6			0.90	67.1

Length $\leq$ 100	LP	LR	F_1	Exact	CB	0 CB
this paper	86.3	85.1	85.7	28.8	1.31	57.2

Verhältnis zu anderen Parsern(2)

Test mit Sektion 23 der Penn Treebank:

Length $\leq$ 40	LP	LR	F_1	Exact	CB	0 CB
Magerman (1995)	84.9	84.6			1.26	56.6
Collins (1996)	86.3	85.8			1.14	59.9
this paper	86.9	85.7	86.3	30.9	1.10	60.3
Charniak (1997)	87.4	87.5			1.00	62.1
Collins (1999)	88.7	88.6			0.90	67.1

Length $\leq$ 100	LP	LR	F_1	Exact	CB	0 CB
this paper	86.3	85.1	85.7	28.8	1.31	57.2

Überraschend: Die Ergebnisse liegen im Mittelfeld zwischen den lexikalisierten Parsern!

Fazit(1)

- Klein&Manning haben gezeigt, dass linguistisch annotierte unlexikalisierte Grammatiken vergleichbare Ergebnisse wie (frühe) lexikalisierte Parser liefern
- Vorteile einer unlexikalisierten Grammatik:
  - Leicht einzuschätzen
  - Leicht zu parsen
  - Zeiteffizient
  - Platzeffizient

Fazit(2)

- Die linguistischen Annotationen verbessern Parsing im Allgemeinen:
- Ein Parser muss nicht lexikalisiert sein...
- ... Aber auch lexikalisierte Parser können von Klein&Mannings linguistischen Annotationen profitieren!

Quelle

Dan Klein, Christopher D. Manning. Accurate Unlexicalised Parsing, ACL 2003.

http://www.cs.berkeley.edu/~klein/

http://www.nlp.stanford.edu/~manning/

Michael John Collins. A New Statistical Parser Based On Bigram Lexical Dependencies, ACL 1996.