DOP – Data-Oriented Parsing

Galina Sîrcu

Christine Neupert

Gliederung

- Einführung
- Rückblick
- Idee des DOP
- Schwierigkeiten
- Ergebnisse

Kurze Geschichte

- DOP ist älter als PCFG (~ Anfang 90er Jahre)
- Erfunden von Remko Scha (Universität Amsterdam)
- Werbung von Rens Bod (Student von Remko Scha)
- Parser brauchte am Anfang 5 Stunden für einen Satz

→ DOP wurde nicht ernst genommen

DOP – Data-oriented parsing

Kurze Geschichte

Remko Scha

Rens Bod

Khalil Sima'an

Kurze Geschichte

Model	Scope of Statistical Dependencies
Charniak (1996)	context-free rules
Collins (1996), Eisner (1996)	context-free rules, headwords
Charniak (1997)	context-free rules, headwords, grandparent nodes
Collins (2000)	context-free rules, headwords, grandparent nodes/rules, bigrams, two-level rules, two-level bigrams, nonheadwords
Bod(1992)	all fragments within parse trees

1. Rückblick

Bisher: PCFG

- Input: Bäume, Sätze, Grammatik

  Wahrscheinlichkeiten für Regeln

- Output: Wahrscheinlichste Analyse für einen Satz

1. Rückblick

Beispiel:

Satz: John loves Mary.

Grammatik PCFG: $P(S \rightarrow NP\ VP) = 1$

$P(VP \rightarrow V\ NP) = 1$

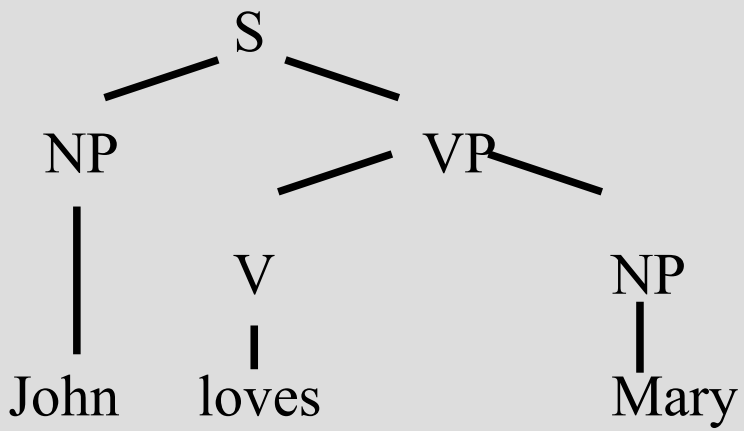
$P(NP \rightarrow \{John\}) = 0{,}5$

$P(NP \rightarrow \{Mary\}) = 0{,}5$

$P(V \rightarrow \{loves\}) = 1$

1. Rückblick

Beispiel:



$$P(\text{Baum}) = \prod (P(\text{Regeln})$$

$$P(\text{Satz}) = \sum (P(\text{Bäumen}))$$

1. Rückblick

- Viterbi-Algorithmus: berechnet den wahrscheinlichsten Parse der Satzanalyse
- Bäume lexikalisch annotiert um Ergebnisse zu verbessern
  - Köpfe
  - Großeltern
  - Bigramme...

2. Idee des DOP

DOP:

- Input: Sätze, Subtrees

  Wahrscheinlichkeiten für Subtrees
- Output: Wahrscheinlichste Analyse für einen Satz

2. Idee des DOP

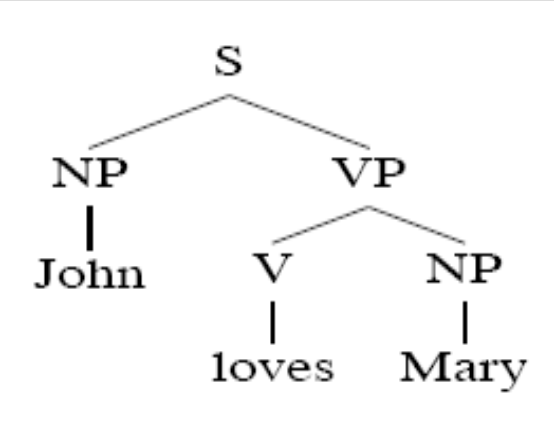
Parsing neuen Sätze:

- Die Zusammensetzung neuer Baumbankfragmente (Subtree) um eine Analyse für den Input-Satz zu finden;
- Die Wahl der wahrscheinlichsten Analyse für den Input-Satz mit Hilfe der Wahrscheinlichkeiten von Fragmenten.

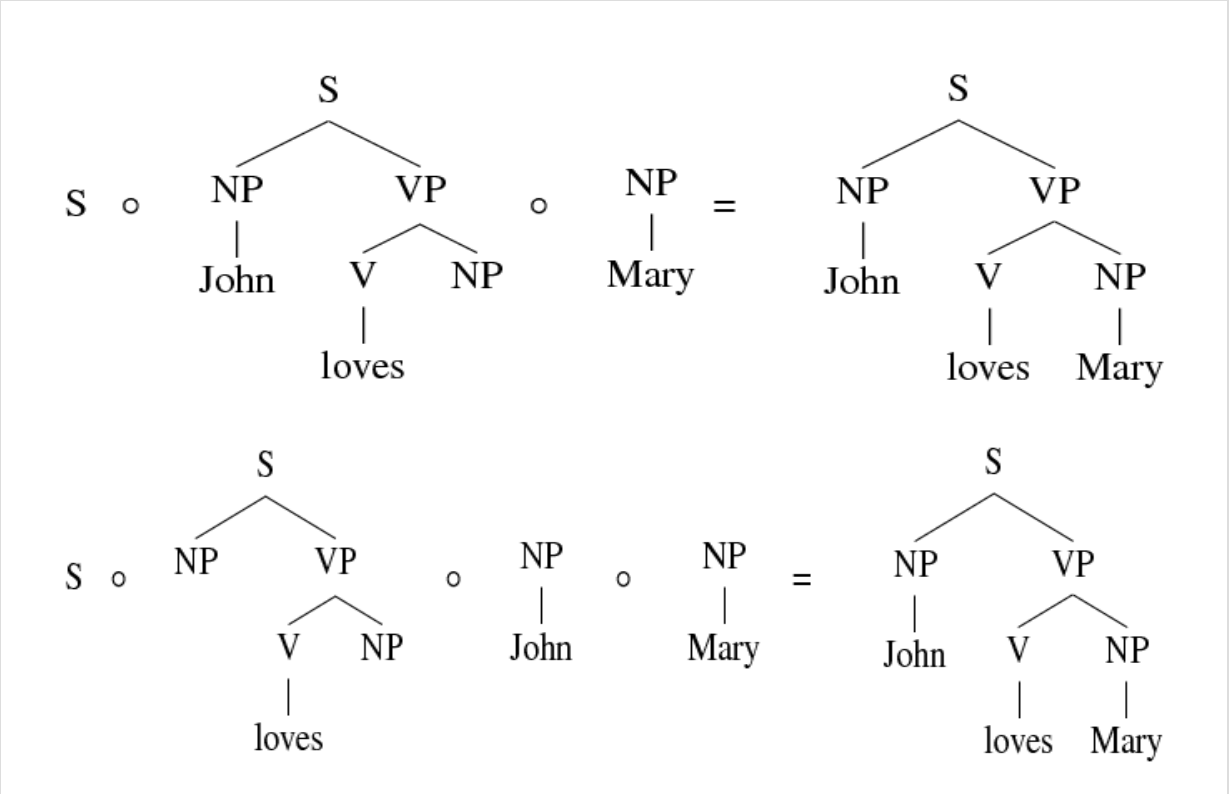
Treebank:

- Fragmente (Subtree) mit ihren Wahrscheinlichkeiten.

2.1. Beispiel



Eine einfache Treebank

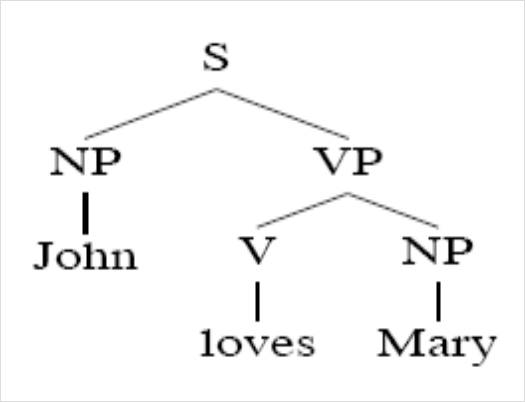


Die Zusammensetzung von Fragmenten

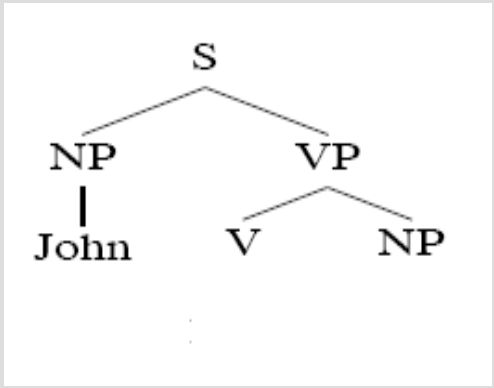
2.2. Subtree (Definition)

- ein abgeschlossener sub-Graph einer Baum aus der Treebank hat am mindestens eine Regel;
- jede innere Knoten dominiert entweder alle seine Kinder oder keine von ihnen.

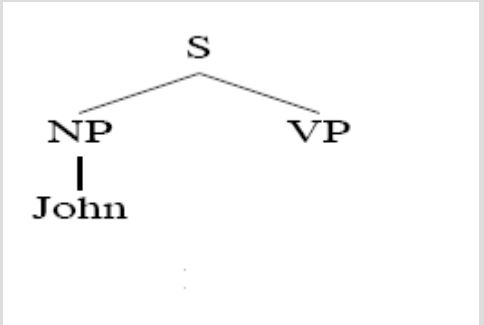
2.3. Subtree (Beispiel)



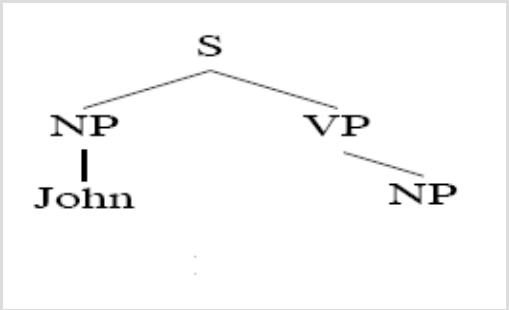
Ein einfache Treebank



Subtree

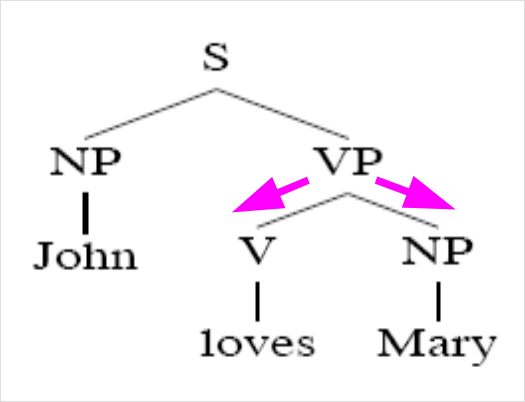


Subtree

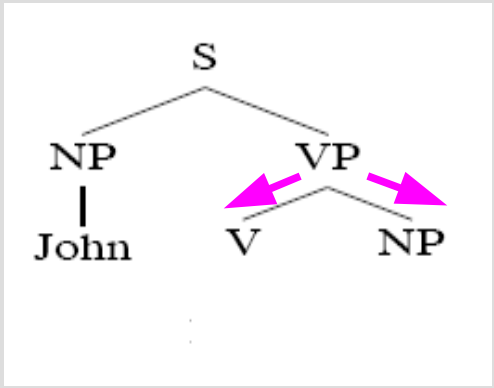


Kein Subtree

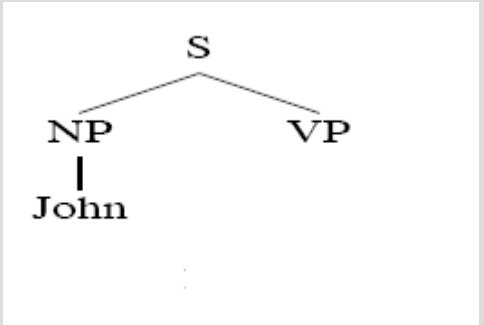
2.3. Subtree (Beispiel)



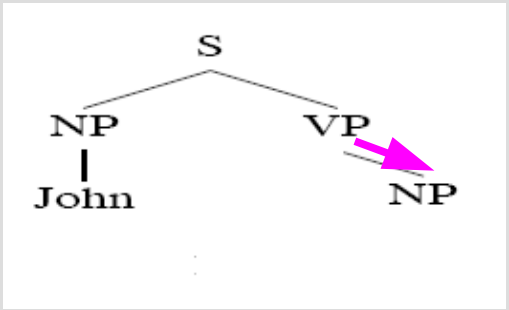
Ein einfache Treebank



Subtree



Subtree



Keine Subtree

2.4. Stochastic Tree-Substitution Grammar (STSG)

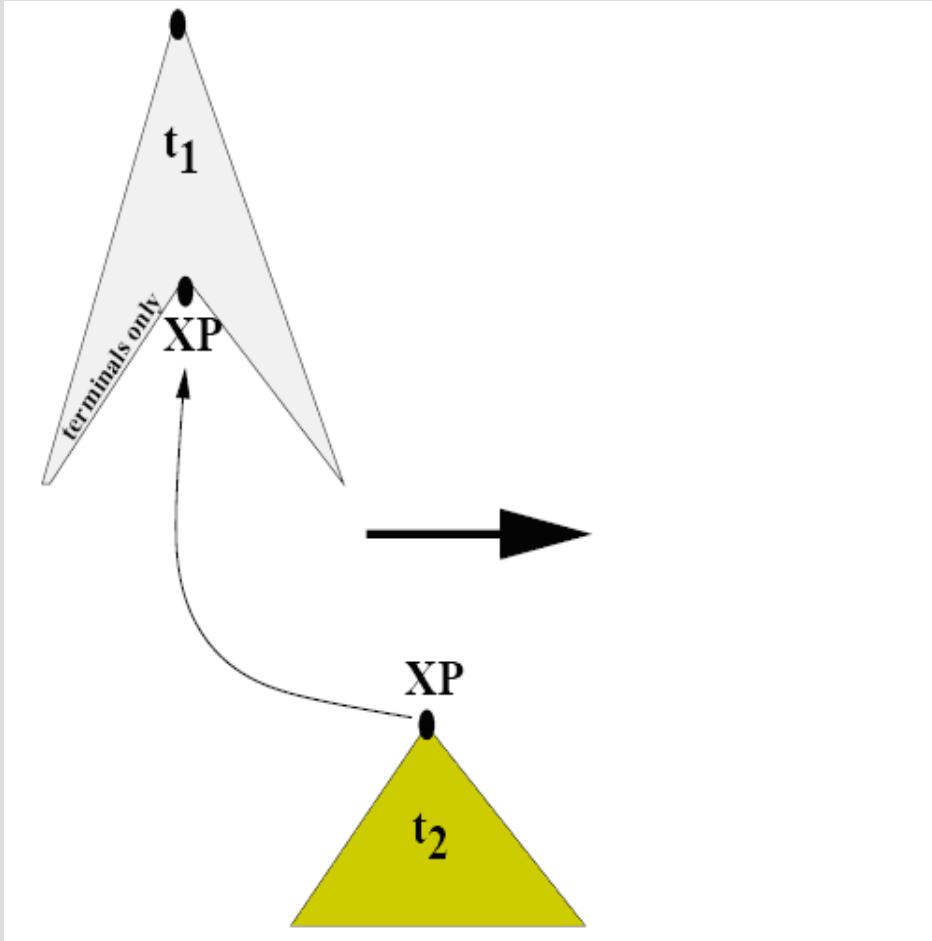
Eine STSG ist das 5-Tupel:

- Terminalsymbole (V_T);
- Nonterminalsymbole (V_N);
- Startsymbol (S);
- Regeln (R – die Menge von alle Subtree);
- Wahrscheinlichkeiten: ($P : R \rightarrow (0,1]$, sodass für alle $A \in V_N$

$$\sum_{t_i \in R: root(t_i) == A} P(t_i \mid A) = 1.0)$$

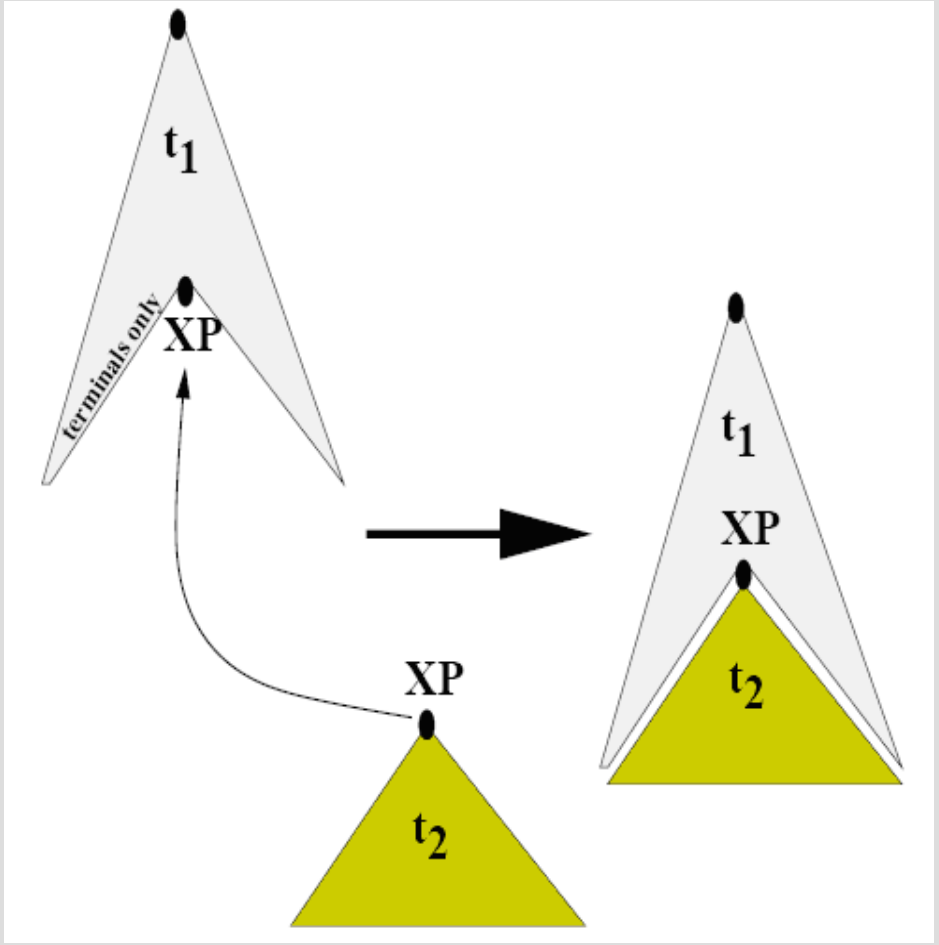
2.5. Substitution in STSG (Definition)

- Operator: ○
- $t_1 \circ t_2$:
  - t_2 – Subtree;
  - t_1 - Subtree oder die Analyse nach der früheren Substitutionen;
  - XP ist das Startsymbol für t_2 und der erste Nonterminal-Symbol links für t_1

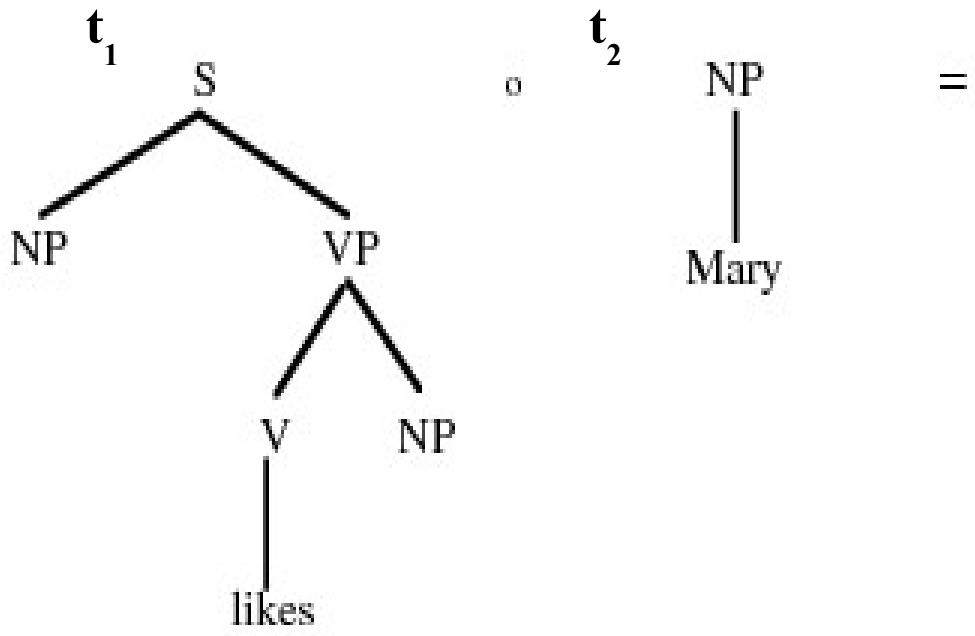


2.5. Substitution in STSG (Definition)

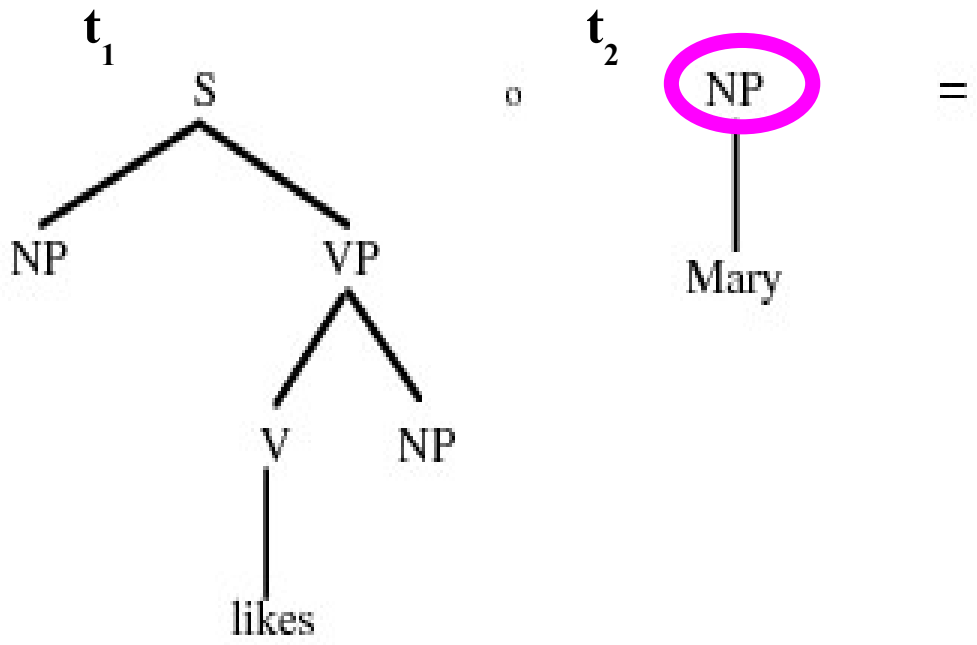
- Das Ergebnis $t_1 \circ t_2$:
  - Die neue Analyse, die mit Hilfe der Substitution XP durch t_2 bekommen wird.



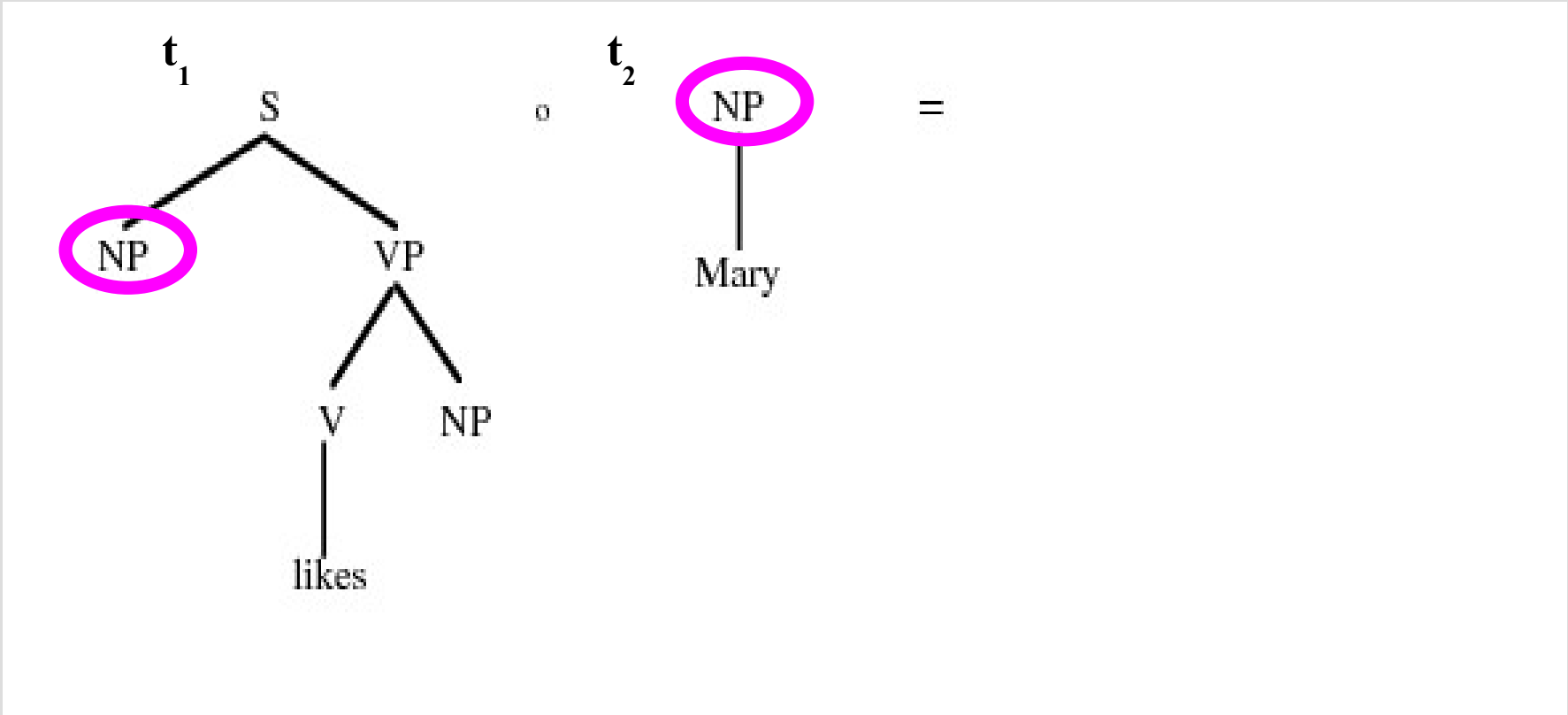
2.6. Substitution in STSG (Bespiel)



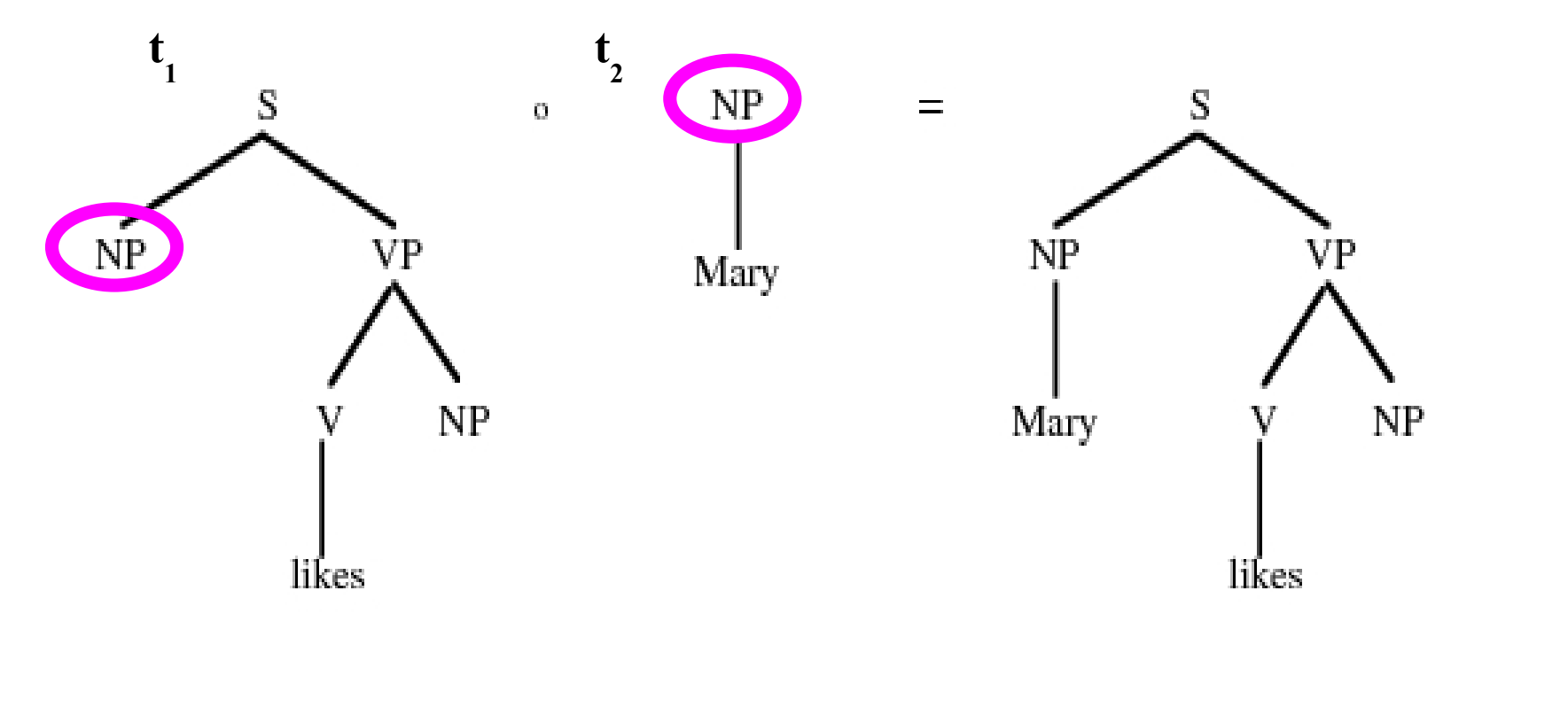
2.6. Substitution in STSG (Bespiel)



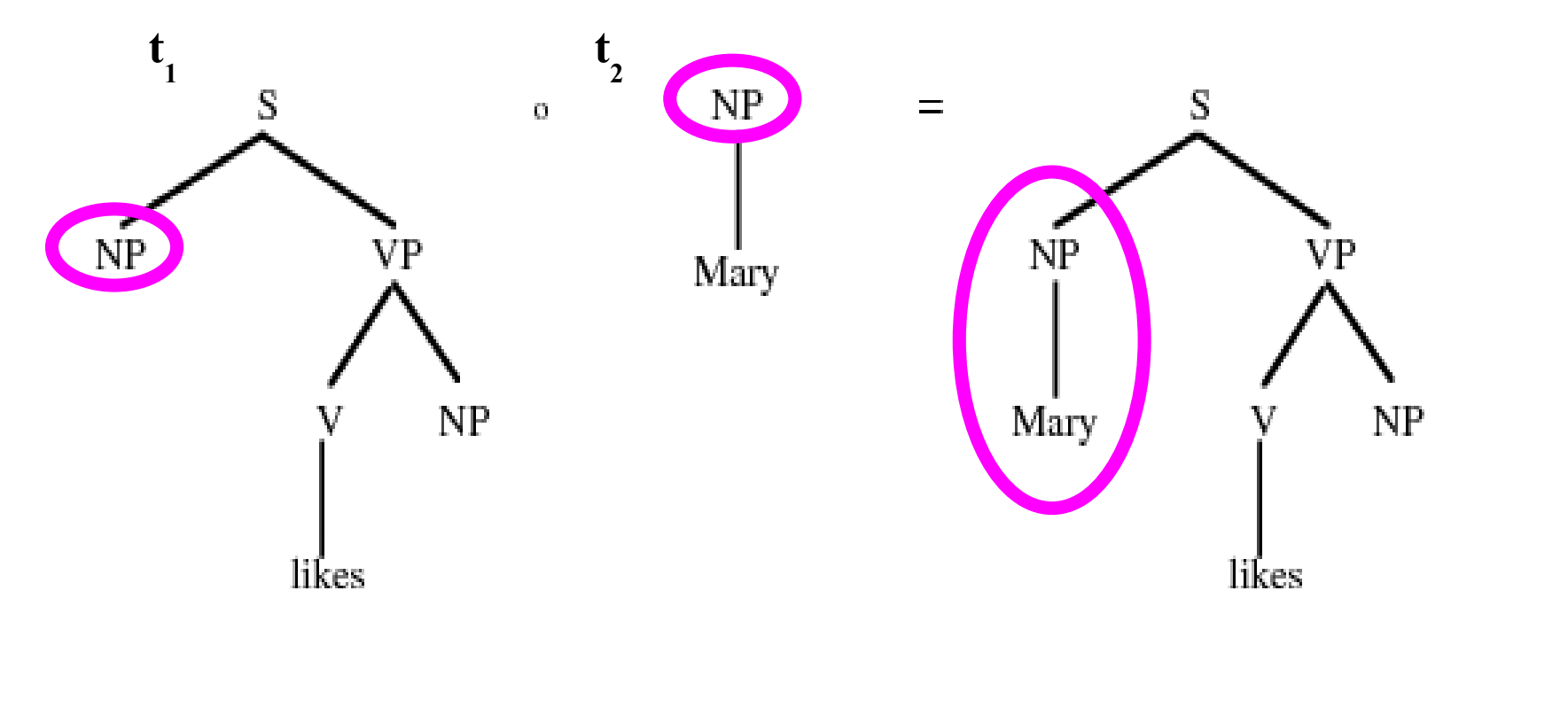
2.6. Substitution in STSG (Bespiel)



2.6. Substitution in STSG (Bespiel)



2.6. Substitution in STSG (Bespiel)



2.7. Derivation

- Derivation - Die Reihenfolge von einer oder mehreren Substitutionen:

$$t_1 \circ t_2 \circ \ldots \circ t_n$$

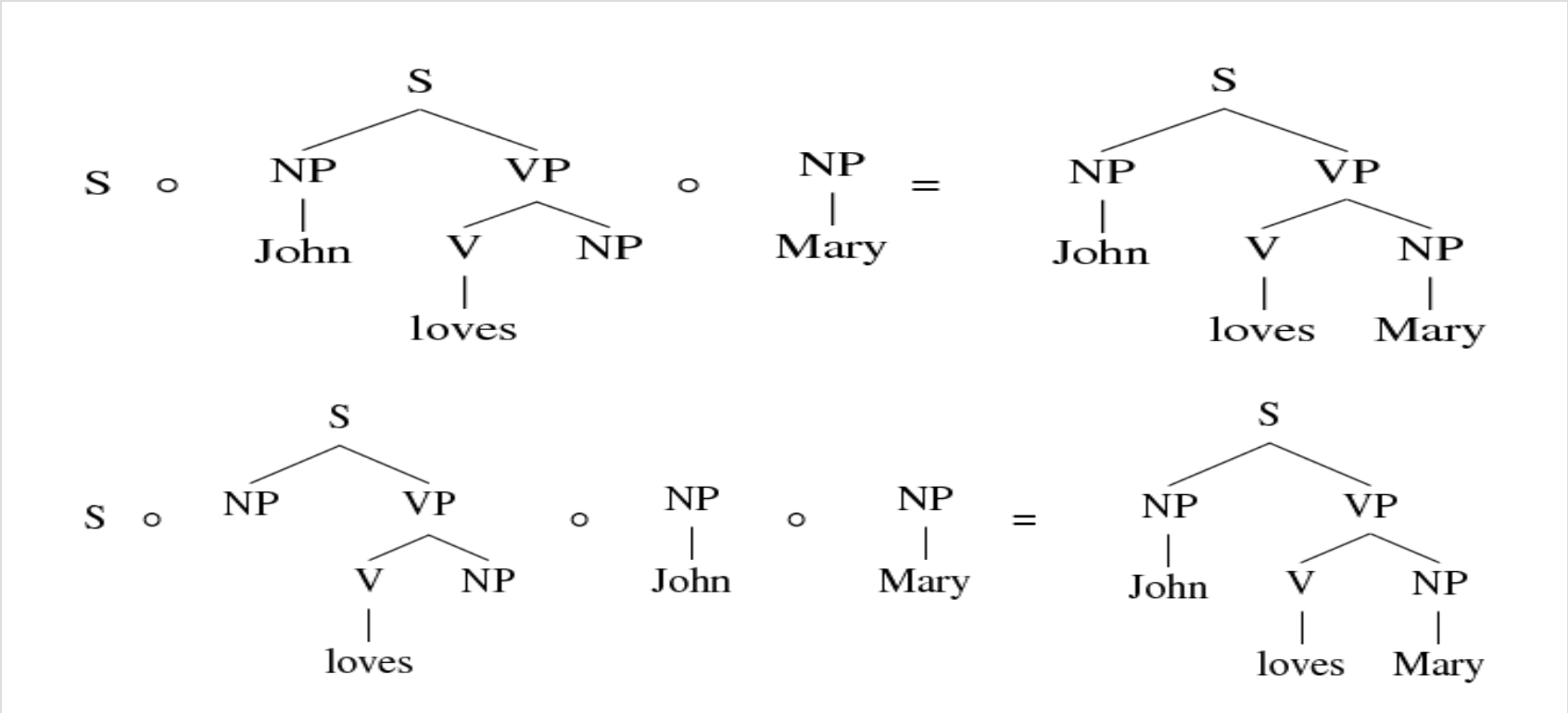
ist

$$(\ldots (t_1 \circ t_2) \circ t_3) \ldots \circ t_{i-1}) \ldots)$$

- Analyse - Der Baum, dessen Struktur aus der Derivation gewonnen wird.

2.8. Bemerkung

- Eine Analyse kann mit verschiedenen Derivationen generiert werden.



2.9. Die Berechnung der Wahrscheinlichkeiten

- Die Wahrscheinlichkeit einer Subtree:

$$P(t \mid root(t)) = \frac{freq(t)}{\sum_{t_i : root(t_i) = = root(t)} freq(t_i)}$$

- Die Wahrscheinlichkeit einer Derivation $D = S \circ t_1 \circ t_2 \circ \ldots \circ t_n$:

$$P(D \mid S) = \prod_{i=1}^{n} P(t_i \mid R_{t_i})$$

2.9. Die Berechnung der Wahrscheinlichkeiten

- Die Wahrscheinlichkeit einer Analyse T:

$$P_G(T) = \sum_{der \in DERS(G)} P(T, der)$$

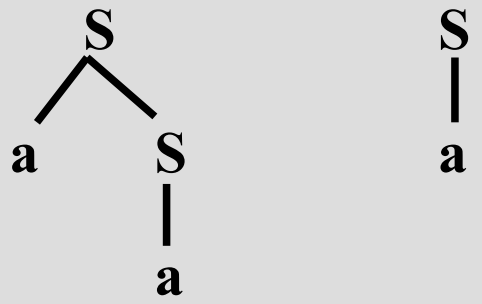
- Die Wahrscheinlichkeit eines Satzes U:

$$P(U) = \sum_{der \in DERS(G)} P(U, der)$$

2.10. Die Berechnung der Wahrscheinlichkeiten (Beispiel)

Sei gegeben:

- Tree-bank:



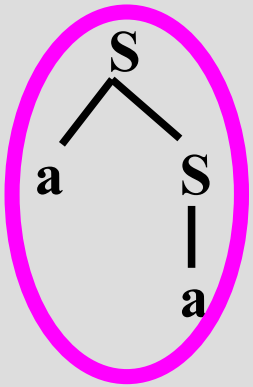
T_1 T_2

$n=3$ $m=7$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

Sei gegeben:

- Tree-bank:



T_1

$n=3$

T_2

$m=7$

- Subtree und Wahrscheinlichkeiten

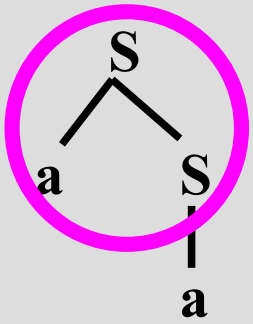
t_1

$n_1=3$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

Sei gegeben:

- Tree-bank:
- Subtree und Wahrscheinlichkeiten



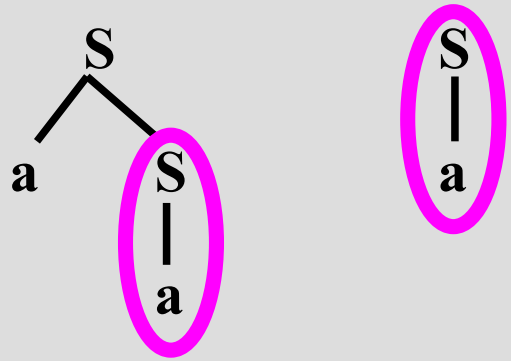
T_1 T_2 t_1 t_2

$n=3$ $m=7$ $n_1=3$ $n_2=3$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

Sei gegeben:

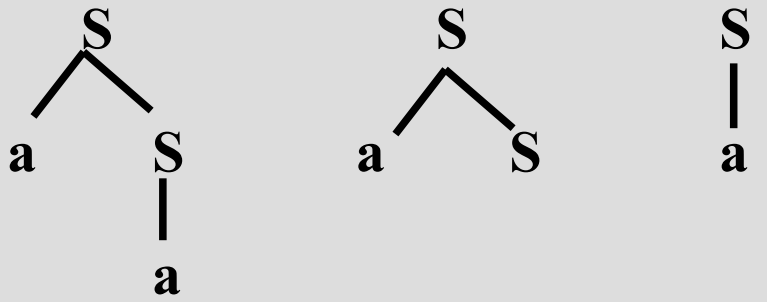
- Tree-bank:



T_1 T_2

$n=3$ $m=7$

- Subtree und Wahrscheinlichkeiten



t_1 t_2 t_3

$n_1=3$ $n_2=3$ $n_3=10$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

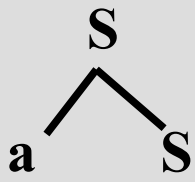
Sei gegeben:

- Tree-bank:

T_1 T_2

$n=3$ $m=7$

- Subtree und Wahrscheinlichkeiten



t_1 t_2 t_3

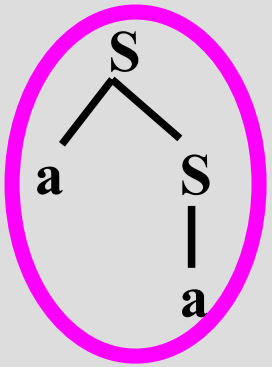
$n_1=3$ $n_2=3$ $n_3=10$

$P(t_1)=3/16$ $P(t_2)=3/16$ $P(t_3)=10/16$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

Sei gegeben:

- Tree-bank:

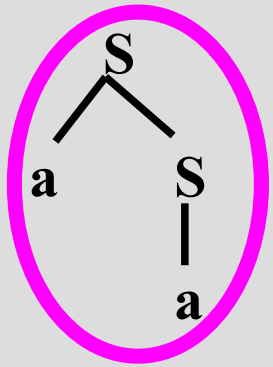


T_1

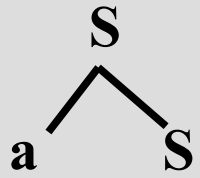
T_2

$n=3$ $m=7$

- Subtree und Wahrscheinlichkeiten



t_1



t_2

t_3

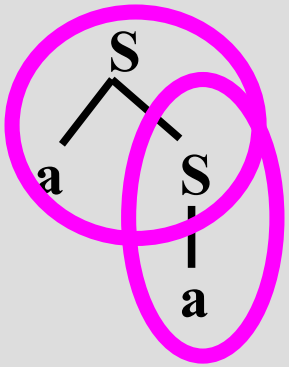
$P(t_1)=3/16$ $P(t_2)=3/16$ $P(t_3)=10/16$

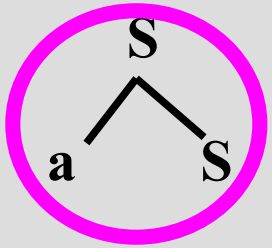
$P(T_1)=P(t_1)+...$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

Sei gegeben:

- Tree-bank:
- Subtree und Wahrscheinlichkeiten





T_1 T_2 t_1 t_2 t_3

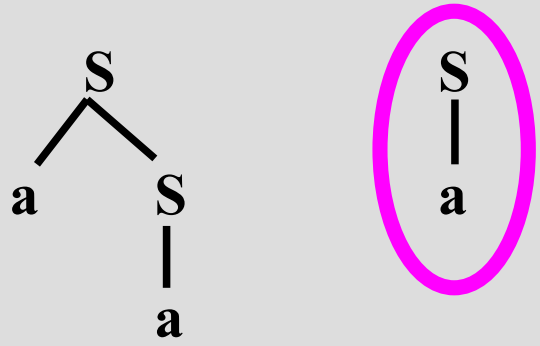
$n=3$ $m=7$ $P(t_1)=3/16$ $P(t_2)=3/16$ $P(t_3)=10/16$

$$P(T_1)=P(t_1)+P(t_2)*P(t_3)=0.3046875$$

2.10. Die Berechnung der Wahrscheinlichkeit (Beispiel)

Sei gegeben:

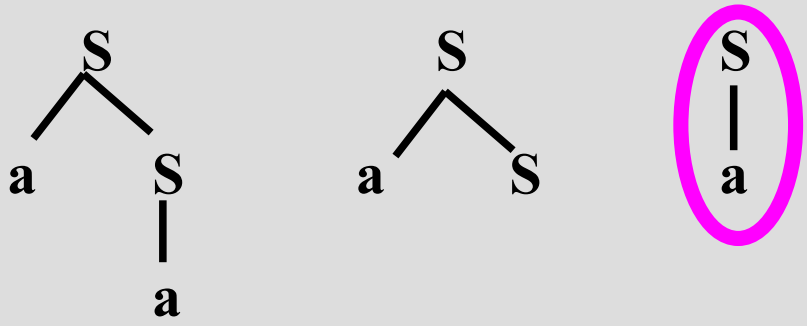
- Tree-bank:



T_1 T_2

$n=3$ $m=7$

- Subtree und Wahrscheinlichkeiten



t_1 t_2 t_3

$P(t_1)=3/16$ $P(t_2)=3/16$ $P(t_3)=10/16$

$$P(T_1)=P(t_1)+P(t_2)*P(t_3)=0.3046875$$

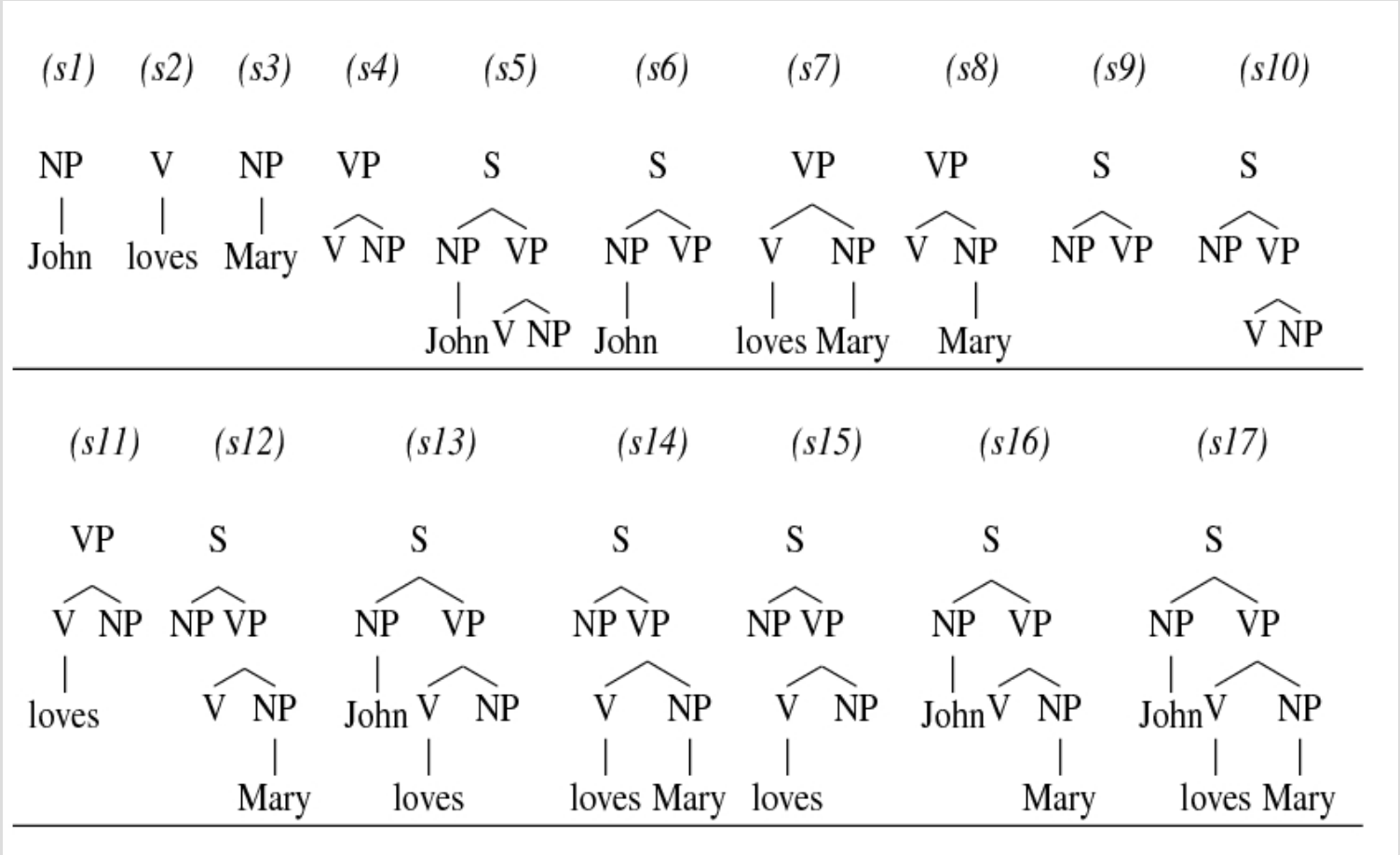
$$P(T_2)=P(t_3)=0.625$$

3. Schwierigkeiten

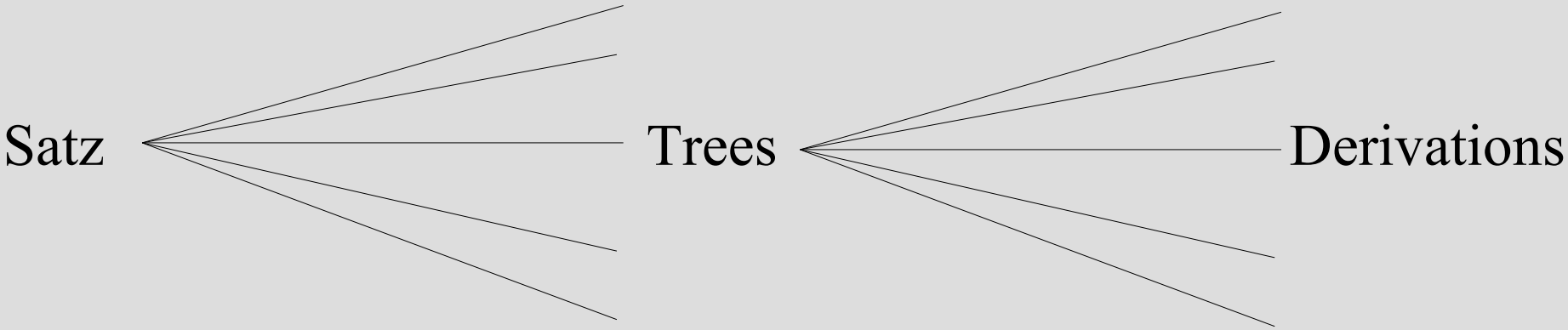
	PCFG	DOP
Symbolisches Parsing	Einfach	Einfach
Statistisches Parsing	Einfach	**Schwierig!**

3. Schwierigkeiten

- Grund: viele Möglichkeiten für Zerlegung von Bäumen



3. Schwierigkeiten



3. Schwierigkeiten

- **Bester Parse**: $P(Tree) = \sum_d \prod_i P(t_{id})$
- Problem **NP-schwer** durch Summe mal Produkt

  → lange Rechenzeiten, keine Lösung garantiert

- Beweis für NP-Completeness von Khalil Sima'an – ebenfalls Student von Remko Scha

3. Schwierigkeiten

- Lösung: statt bester Parse

  beste Derivation (most probable derivation MPD)

  oder

  symbolisch kürzeste Derivation

  (Substitution in möglichst wenig Schritten)

3. Schwierigkeiten

- **Most Probable Derivation – zu schwach**

  wahrscheinlichste Derivation muss nicht zum wahrscheinlichsten Baum gehören

  ähnlich wie im PCFG parser

- **Symbolisch kürzeste Derivation**

  ebenso verfälschte Ergebnisse

3. Schwierigkeiten

- **DOP1** nutzt weak lexicalisation – schwache Lexikalisierung
  - bringt kaum Verbesserung
  - schwierig zu realisieren
  - z.B. Subkategorisierung von Verben nicht realisierbar
  - vollständige Lexikalisierung verbraucht zu viel Zeit und Speicher

3. Schwierigkeiten

- anderer Ansatz: MPP zu teuer
  - Lösung: Beschränkung der Menge der Derivationen
    - **N-best Derivationen** aller Analysen eines Satzes
    - Gruppierung der Derivationen nach Analysen
    - Berechnung der Wahrscheinlichkeit der Analysen

      $\sum$ P(zur Analyse gehörenden Derivationen)
    - Output: wahrscheinlichste Analyse

3. Schwierigkeiten

- Bsp: 1000 wahrscheinlichste Derivationen für einen Satz
- Grouping: 230 Derivationen für Analyse 1

  450 Derivationen für Analyse 2

  320 Derivationen für Analyse 3
- Berechnung der Wahrscheinlichkeiten für Analysen 1, 2 und 3
- Analyse 3 ist die wahrscheinlichste

4.1 Experiment

- Training: WSJ (wie früher bei Collins, 1997, 1999; Charniak, 1997, 2000; Ratnaparkhi, 1999)

≈ 40,000 Sätze

- Testing: WSJ

≈ 2416 Sätze (≤100 Wörter)

4.1 Experiment

- Subtree:
  - ohne semantische Tags, Koreferenz, die Anführungszeichen;
  - für jede Tiefe zwischen >1 and ≤14:

Die Vergleichslinienmenge von Subtrees:

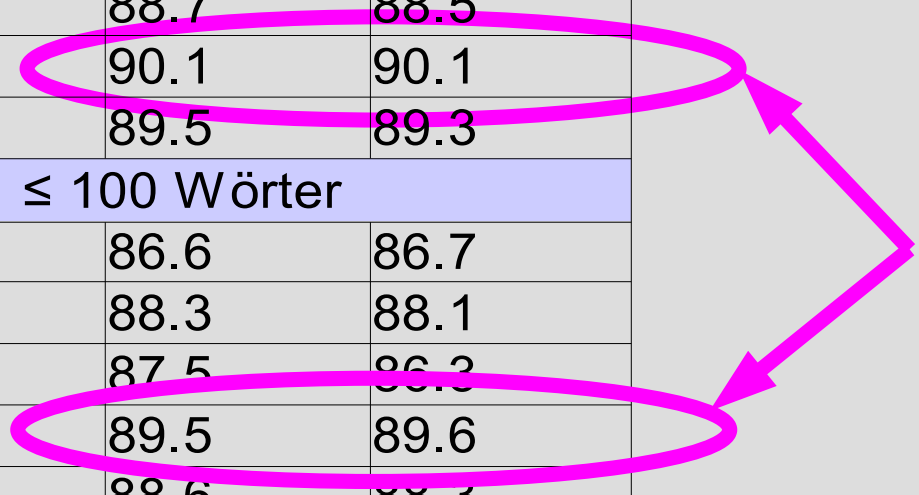
≈ 5,217,529 Subtrees

4.2 Vergleich von Ergebnissen

Parser	LP	LR
≤ 40 Wörter		
Char97	87.4	87.5
Coll99	88.7	88.5
Char00	90.1	90.1
Bod00	89.5	89.3
≤ 100 Wörter		
Char97	86.6	86.7
Coll99	88.3	88.1
Ratna99	87.5	86.3
Char00	89.5	89.6
Bod00	88.6	88.3

4.2 Vergleich der Ergebnisse

Parser	LP	LR
≤ 40 Wörter		
Char97	87.4	87.5
Coll99	88.7	88.5
Char00	90.1	90.1
Bod00	89.5	89.3
≤ 100 Wörter		
Char97	86.6	86.7
Coll99	88.3	88.1
Ratna99	87.5	86.3
Char00	89.5	89.6
Bod00	88.6	88.3



4.2 Vergleich der Ergebnisse

Parser	LP	LR
≤ 40 Wörter		
Char97	87.4	87.5
Coll99	88.7	88.5
Char00	90.1	90.1
Bod00	89.5	89.3
≤ 100 Wörter		
Char97	86.6	86.7
Coll99	88.3	88.1
Ratna99	87.5	86.3
Char00	89.5	89.6
Bod00	88.6	88.3

4.3 Der Einfluss der Subtrees

Tiefe des Subtrees	LP	LR
1	76.0	71.8
≤2	80.1	76.5
≤3	82.8	80.9
≤4	84.7	84.1
≤5	85.5	84.9
≤6	86.2	86.0
≤8	87.9	87.1
≤10	88.6	88.0
≤12	89.1	88.8
≤14	89.5	89.3

4.3 Der Einfluss der Tiefe der Subtrees

Tiefe des Subtrees	LP	LR
1	76.0	71.8
≤2	80.1	76.5
≤3	82.8	80.9
≤4	84.7	84.1
≤5	85.5	84.9
≤6	86.2	86.0
≤8	87.9	87.1
≤10	88.6	88.0
≤12	89.1	88.8
≤14	89.5	89.3

Charniak's Treebank Grammar (1997)

Charniak's Treebank Grammar (2000)

4.4 Der Einfluss des lexikalischen Kontexts

# Wörter im Subtree	LP	LR
1	84.4	84.0
≤2	85.2	84.9
≤3	86.6	86.3
≤4	87.6	87.4
≤6	88.0	87.9
≤8	89.2	89.1
≤10	90.2	90.1
≤11	90.8	90.4
≤12	90.8	90.5
≤13	90.4	90.3
≤14	90.3	90.3
≤16	89.9	89.8
unbeschränkt	89.5	89.3

4.4 Der Einfluss des lexikalischen Kontexts

# Wörter im Subtree	LP	LR
1	84.4	84.0
≤2	85.2	84.9
≤3	86.6	86.3
≤4	87.6	87.4
≤6	88.0	87.9
≤8	89.2	89.1
≤10	90.2	90.1
≤11	90.8	90.4
≤12	90.8	90.5
≤13	90.4	90.3
≤14	90.3	90.3
≤16	89.9	89.8
unbeschränkt	89.5	89.3

Besser

als

Charniak (2000)

4.4 Der Einfluss des lexikalischen Kontexts

# Wörter im Subtree	LP	LR
1	84.4	84.0
≤2	85.2	84.9
≤3	86.6	86.3
≤4	87.6	87.4
≤6	88.0	87.9
≤8	89.2	89.1
≤10	90.2	90.1
≤11	90.8	90.4
≤12	90.8	90.5
≤13	90.4	90.3
≤14	90.3	90.3
≤16	89.9	89.8
unbeschränkt	89.5	89.3

4.5 Der Einfluss des strukturellen Kontexts

Tiefe von unlexikalisierte Subtree	LP	LR
≥1	79.9	77.7
≥2	86.4	86.1
≥3	89.9	89.5
≥4	90.6	90.2
≥5	90.7	90.6
≥6	90.8	90.6
≥7	90.8	90.5
≥8	90.8	90.5
≥10	90.8	90.5
≥12	90.8	90.5

4.5 Der Einfluss des strukturellen Kontexts

Tiefe von unlexikalisierte Subtree	LP	LR
≥1	79.9	77.7
≥2	86.4	86.1
≥3	89.9	89.5
≥4	90.6	90.2
≥5	90.7	90.6
≥6	90.8	90.6
≥7	90.8	90.5
≥8	90.8	90.5
≥10	90.8	90.5
≥12	90.8	90.5

4.6 Der Einfluss von non-Kopfwörter

# non-Kopfwörter im Subtree	LP	LR
0	89.6	89.6
≤1	90.2	90.1
≤2	90.4	90.2
≤3	90.3	90.2
≤4	90.6	90.4
≤5	90.6	90.6
≤6	90.6	90.5
≤7	90.7	90.7
≤8	90.8	90.6
unbeschränkt	90.8	90.6

4.6 Der Einfluss von non-Kopfwörtern

# non-Kopfwörter im Subtree	LP	LR
0	89.6	89.6
≤1	90.2	90.1
≤2	90.4	90.2
≤3	90.3	90.2
≤4	90.6	90.4
≤5	90.6	90.6
≤6	90.6	90.5
≤7	90.7	90.7
≤8	90.8	90.6
unbeschränkt	90.8	90.6

ΔLP=1.2%

ΔLR=1.0%

4.7 Die finalen Ergebnisse

- Beschränkungen für einen Subtree:
  - bis 12 Wörter;
  - Tiefe der unlexikalisierte Subtrees ≤6.
- Ergebnisse für alle Sätze (≤100 Wörter):
  - LP = 89.7%;
  - LR = 89.7%.

4.7 Die finale Ergebnisse

Parser	LP	LR
≤ 100 Wörter		
Char97	86.6	86.7
Coll99	88.3	88.1
Ratna99	87.5	86.3
Char00	89.5	89.6
Bod00	88.6	88.3
Bod01	89.7	89.7

4.7 Die finale Ergebnisse

Parser	LP	LR
≤ 100 Wörter		
Char97	86.6	86.7
Coll99	88.3	88.1
Ratna99	87.5	86.3
Char00	89.5	89.6
Bod00	88.6	88.3
Bod01	89.7	89.7

Bibliography

- Rens Bod, Remko Scha (1996), Data-Oriented Language Processing – An Overview
- Rens Bod (2001), What is the Minimal Set of Fragments that Achieves Maximal Parse Accuracy?, ACL 2001
- Khalil Sima'an (2003), A Short Introduction to the DOP Model
- Khalil Sima'an (2003), Lecture on Data-Oriented Parsing, ESSLII 2003