

# Exemplarbasierte Syntax durch Data-Oriented Parsing

Christian Pietsch

<http://purl.org/net/pietsch>

Exzellenzcluster *Cognitive Interaction Technology* und  
Fakultät für Linguistik und Literaturwissenschaft  
Universität Bielefeld

Vortrag in Prof. Gerhard Jägers Seminar *Exemplarbasierte  
Theorien der Sprachverarbeitung* am 3. Juni 2008  
Literatur: [BOD 2006]

# Übersicht

- 1 Data-oriented parsing**
  - Motivation
- 2 Tree-DOP**
  - DOP: Grundidee
  - Tree-DOP-Operationen
  - Probabilistische Disambiguierung
  - Zusammenfassung
- 3 Ausblick**
  - DOP für artikuliertere Grammatikformalismen

# Kompetenzgrammatik

## Kompetenzgrammatik ...

- erklärt die Produktivität der Sprache bei begrenztem Zeicheninventar (Generative Grammatik)
- Sprachwissen als Beherrschung einer konsistenten Regelmenge
- kategorisiert eine Äußerung als entweder grammatisch oder ungrammatisch
- weist grammatischen Äußerungen eine Bedeutung zu
- erfasst linguistische Generalisierungen
- Grammatik als kleinste, nichtredundante, orthogonale Regelbasis für eine Menge von Äußerungen

# Die empirische Herausforderung

Sprache besteht zu  $\frac{1}{3}$  bis  $\frac{1}{2}$  aus formelhaften Wendungen (i.w.S.)  
[CONKLIN und SCHMITT 2008]

- ⇒ Performanzgrammatik
- ⇒ Usage-based linguistics

# Thesen

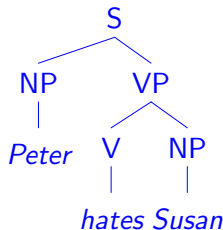
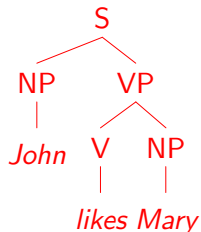
- Exemplarbasierte Syntax ist möglich, denn
- DOP ermöglicht grammatische Produktivität
- Universalrepräsentation statt Universalgrammatik als angeborene Voraussetzung für Sprachverstehen und -produktion

# Voraussetzung

Voraussetzung: syntaktisch (mit Konstituentenstruktur)  
annotiertes Korpus (= Baumbank)

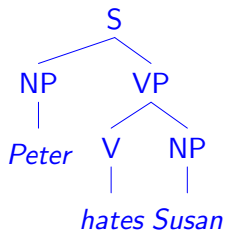
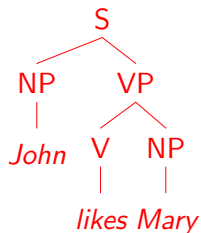
- ⇒ DOP nutzt Analysen nach einer Kompetenzgrammatik, übernimmt aber evt. auch deren Fehler
- ⇒ Es gibt noch kein Korpus mit der Spracherfahrung eines Menschen

# Analysebeispiel



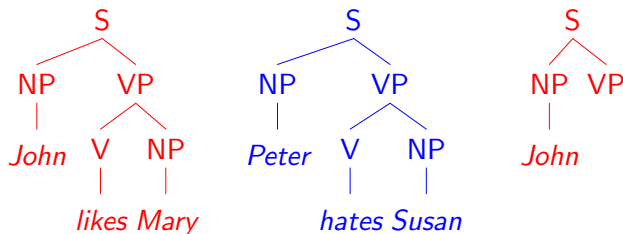
Beispiel: ein kleines annotiertes Korpus (Baumbank)

# Analysebeispiel



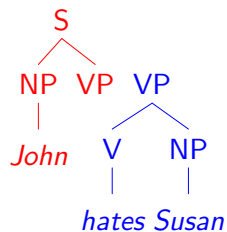
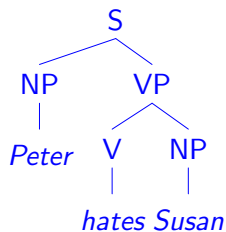
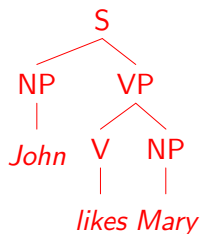
Preisfrage: Was ist die Struktur von *John hates Susan*?

# Analysebeispiel



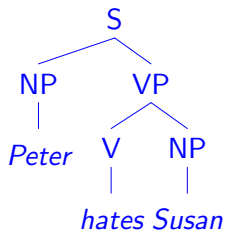
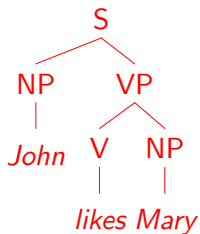
Ableitung 1

# Analysebeispiel

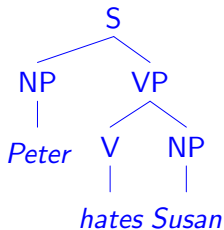
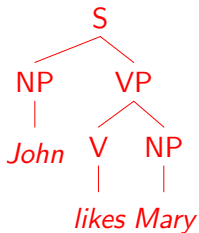


Ableitung 1

# Analysebeispiel

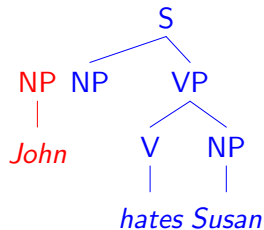
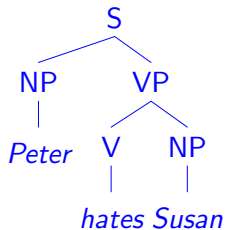
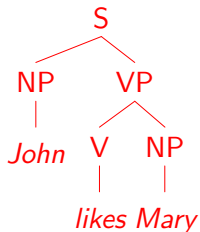


# Analysebeispiel



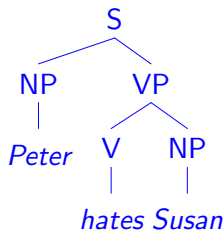
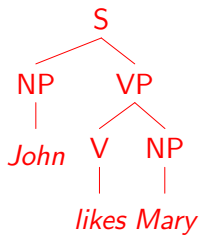
Ableitung 2

# Analysebeispiel

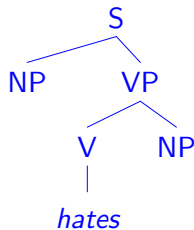
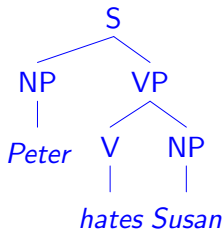
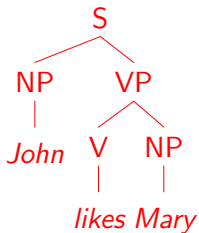


Ableitung 2

# Analysebeispiel

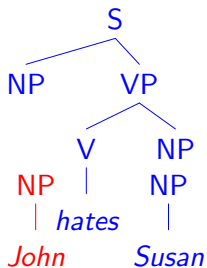
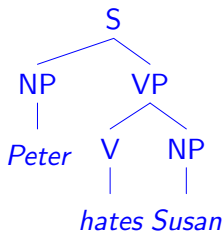
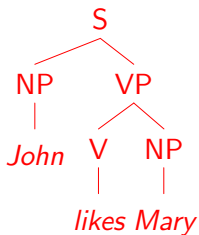


# Analysebeispiel



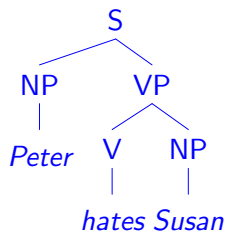
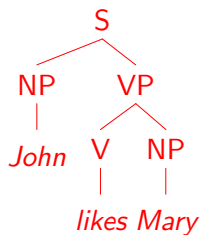
Ableitung 3

# Analysebeispiel

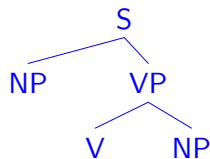
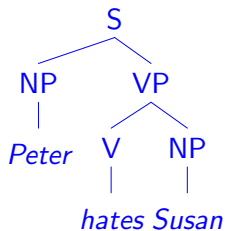
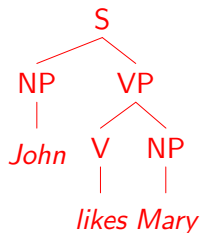


Ableitung 3

# Analysebeispiel

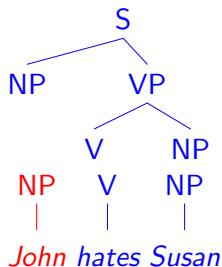
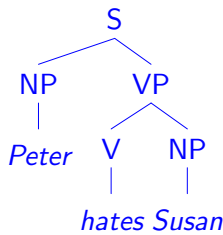
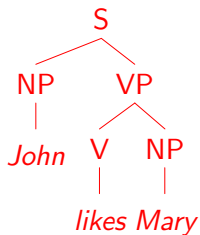


# Analysebeispiel



Ableitung 4

# Analysebeispiel



Ableitung 4

# Dekomposition

In Tree-DOP wird ein Baum durch die wiederholte Anwendung zweier Operationen in Fragmente zerlegt:

## Definition (Root operation)

Die Root-Operation wählt einen Knoten eines Baums als Wurzel des neuen Unterbaums und löscht alle Knoten außer dem ausgewählten und allen Knoten, die er dominiert.

## Definition (Frontier operation)

Die Frontier-Operation wählt dann eine (evt. leere) Menge von Knoten (aber nicht die Wurzel) in dem neuen Unterbaum und löscht alle Unterbäume, die von diesen Knoten dominiert werden.

Alle resultierenden Fragmente (auch Duplikate) werden gespeichert (Multimenge).

# Kombination

## Definition (Label substitution)

In Tree-DOP werden zwei Fragment-Bäume kombiniert, indem der Wurzelknoten des einen Baums den am weitesten links stehenden Nichtterminalknoten des anderen Baums ersetzt, sofern die Kategorien übereinstimmen. Schreibweise des Operators: ○

# Fragmentwahrscheinlichkeit

## Definition (Fragmentwahrscheinlichkeit)

$$P(t) = \frac{|t|}{\sum_{t': r(t')=r(t)} |t'|}$$

$|t|$  ... Häufigkeit des Baums  $t$  in der Multimenge der Fragmente  
 $r(t)$  ... Kategorie des Wurzelknotens von  $t$

# Ableitungswahrscheinlichkeit

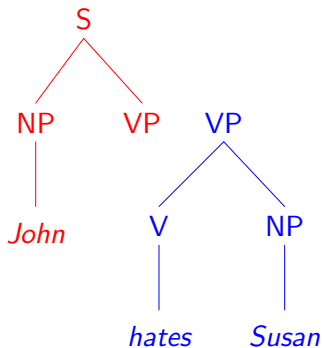
## Definition (Ableitungswahrscheinlichkeit)

$$P(D) = P(t_1 \circ \dots \circ t_n) = \prod_i P(t_i)$$

Die Fragmentwahrscheinlichkeiten  $P(t_i)$  aus der vorigen Folie sind statistisch unabhängig voneinander (nimmt Bod an). Daher ergibt sich die Wahrscheinlichkeit eines gemeinsamen Auftretens (joint probability) durch Multiplikation der Fragmentwahrscheinlichkeiten.

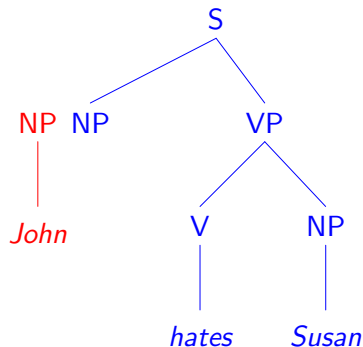
# Ableitung 1

(Siehe Liste aller Fragmente in Abb. 8 auf S. 11 im Artikel!)



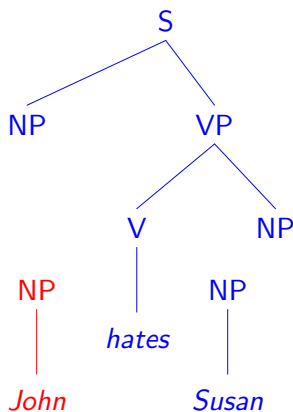
$$P(\text{Ableitung 1}) = 1/20 \times 1/8 = 0,00625$$

# Ableitung 2



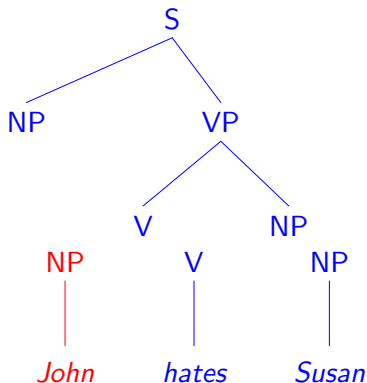
$$P(\text{Ableitung 2}) = 1/4 \times 1/20 = 0,0125$$

# Ableitung 3



$$P(\text{Ableitung 3}) = 1/20 \times 1/4 \times 1/4 = 0,003125$$

# Ableitung 4



$$P(\text{Ableitung 4}) = 2/20 \times 1/4 \times 1/2 \times 1/4 = 0,00625$$

# Analysewahrscheinlichkeit

## Definition (Analysewahrscheinlichkeit)

$$P(T) = \sum_{D \text{ derives } T} P(D)$$

$T$  ... Analyse (parse tree)

Die Wahrscheinlichkeit einer Analyse ist die Wahrscheinlichkeit, dass irgend eine ihrer Ableitungen sie hervorbringt (disjunkte Ereignisse / disjoint probability). Die Ableitungswahrscheinlichkeiten werden daher addiert.

# Äußerungswahrscheinlichkeit

## Definition (Äußerungswahrscheinlichkeit)

$$P(W) = \sum_{T \text{ yields } W} P(T)$$

$W$  ... Äußerung (word string)

Die Wahrscheinlichkeit einer Äußerung ist die Wahrscheinlichkeit, dass irgend eine ihrer Analysen sie hervorbringt. Wieder handelt es sich um eine disjunkte Ereignisse, deren Wahrscheinlichkeiten addiert werden.

# Die wahrscheinlichste Analyse

Unseres Beispiel liefert nur 1 Analyse. Das ist nicht die Regel.

$$P(T|T \text{ yields } W) = \frac{P(T)}{P(W)}$$

Diese bedingte Wahrscheinlichkeit wird genutzt, um konkurrierende Analyse nach ihrer Wahrscheinlichkeit zu ordnen.

# Eigenschaften von Tree-DOP

- Tree-DOP bevorzugt die Analysen, für die es möglichst viele Ableitungen findet
- ⇒ Performanzorientierung (usage-based linguistics)
- Tree-DOP bevorzugt die Analysen, die aus möglichst großen Fragmenten bestehen
- ⇒ Wahrscheinlichkeit als durchschnittliche Ähnlichkeit zwischen einer Äußerung und den Exemplaren im Korpus
- ⇒ Analogiebildung ohne Einschränkung der generativen Kapazität

# DOP-Rezept

Um den DOP-Ansatz auf einen neuen Bereich zu übertragen, müssen 4 Fragen beantwortet werden:

- 1 Wie werden Äußerungen repräsentiert?
- 2 Wie werden Repräsentationen dekomponiert?
- 3 Wie werden Fragmente kombiniert?
- 4 Wie werden die Analyse-Kandidaten disambiguiert?

[ARNOLD und LINARDAKI 2007]

# Warum Tree-DOP nicht reicht

Kontextfreie Konstituentenstrukturgrammatiken haben wohlbekannte Grenzen:

- sie übergenerieren
- ihre generative Kapazität ist für manche Sprachen nicht ausreichend
- viele linguistisch relevante Abhängigkeiten und Generalisierungen lassen sich darin nicht ausdrücken
- sie eignen sich nicht zur Semantikonstruktion oder Sprachgenerierung
- sie eignen sich daher auch nur sehr eingeschränkt zur automatischen Sprachverarbeitung

Computerlinguisten bevorzugen Grammatikformalismen, die mit Merkmalsstrukturen angereichert sind. Wichtigste Vertreter sind die Unifikationsgrammatiken LFG und HPSG.

# LFG-DOP

- entwickelt von Rens Bod und Ronald Kaplan [BOD 2006]
  - führt eine dritte Dekompositionsoption ein: *Discard*
  - diese erlaubt das Löschen von unverknüpften Merkmalen (außer P<sub>RED</sub>)
- ⇒ die dadurch generalisierten F-Struktur-Fragmente führen zu Übergenerierung, der mit metagrammatischen Urteilen entgegengewirkt werden muss: *Ein Satz ist grammatisch in Hinsicht auf ein Korpus gdw. er mindestens eine gültige Repräsentation hat, die keine generalisierten Fragmente enthält.* Das kann bei Datenknappheit aber dazu führen, dass grammatische Sätze als ungrammatisch beurteilt werden. Vorteil: graduelle Grammatikalitätsurteile.
- *Discard* führt außerdem zu einem sog. "Auslaufen von Wahrscheinlichkeitsmasse"
- ⇒ ist probabilistisch nicht mehr korrekt

# LFG-DOP

- [WAY 1999] hat für seine LFG-DOP-Implementation im Bereich des maschinellen Übersetzens eine pragmatische Lösung entwickelt: Bestimmte Merkmale werden von *Discard* ausgenommen.
- [ARNOLD 2007] präsentiert einen alternativen Ansatz, der mit linguistisch motivierten zusätzlichen LFG-Regeln arbeitet. Dieser leidet nicht am “Auslaufen von Wahrscheinlichkeitsmasse”.

# HPSG-DOP

- erster Ansatz [NEUMANN 2003] war eher eine Technik zur Extraktion einer spezialisierten SLTG-Grammatik aus einer großen HPSG-Grammatik anhand eines Korpus. Diese konnte mit einem schnelleren Parser verarbeitet werden. Anschließend werden HPSG-Repräsentationen wieder “ausgepackt” (vgl. HPSG-nach-TAG-Kompilation).
- zweiter Ansatz [ARNOLD und LINARDAKI 2007] arbeitet tatsächlich mit HPSG-Repräsentationen und macht sich das hierarchische Typsystem moderner HPSG-Grammatiken zunutze. Zentral ist die neue Operation *Type expansion*. Probabilistisch sauber, aber noch nicht empirisch evaluiert.



ARNOLD, DOUG (2007).

*DOP-based models for richer grammatical frameworks.*

In: RADFORD, ANDREW, Hrsg.: *Essex Research Reports in Linguistics*, Bd. 53, S. 17–41. Dept. of Language and Linguistics, University of Essex.



ARNOLD, DOUG und E. LINARDAKI (2007).

*HPSG-DOP: Towards exemplar-based HPSG.*

In: *ESSLLI 2007, 6–17, August 2007.*

Presented at the Workshop 'Exemplar-Based Models of Language Acquisition and Use'.



BOD, RENS (2006).

*Exemplar-Based Syntax: How to Get Productivity from Examples.*

*The Linguistic Review*, 23.

Special Issue on Exemplar-Based Models in Linguistics.



CONKLIN, KATHY und N. SCHMITT (2008).

*Formulaic Sequences: Are They Processed More Quickly than Nonformulaic Language by Native and Nonnative Speakers?.*  
*Applied Linguistics*, 29(1):72–89.



NEUMANN, GÜNTER (2003).

*A Data-driven Approach to Head-driven Phrase Structure Grammar.*

In: BOD, RENS, R. SCHA und K. SIMA'AN, Hrsg.:  
*Introduction to Data-Oriented Parsing.* CSLI.



WAY, ANDY (1999).

*A Hybrid Architecture for Robust MT using LFG-DOP.*

*Journal of Experimental and Theoretical Artificial Intelligence*,  
11(4).

Special Issue on Memory-Based Language Processing.