Diplomarbeit von Niels Stockfleth - Integration von CDG-Parsing und Fehlersimulation

Jérôme Roy & Karin Colsman

30. November 2001

List of Slides

Slide 1

Fehlerdiagnose mit CDG-Parsing

- CD-Grammatiken
- Bewertete CD-Grammatiken
- Fehlerdiagnose

Slide 2

CD-Grammatiken

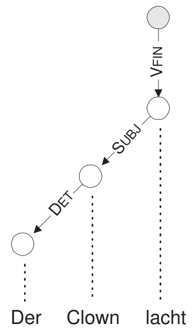
- Dependenzrelation und Dependenzstruktur
- Parsing als Disambiguierungsprozeß
- Formale Definition einer CD-Grammatik
- Strukturbeschreibungen
- Parsing von CD-Grammatiken

Slide 3

Dependenzrelation und Dependenzstruktur

- Binäre Relation
- antisymmetrisch, antireflexiv, antitransitiv
- $dep(x, y) = y$ dominiert x
- z.B. in “Der Clown lacht”:
  x subjekt-von y mit
  $x = Clown$, $y = lacht$
- Dependenzstrukturen sind verbundene, gerichtete Graphen, oft auch nur Baumstrukturen

Slide 4

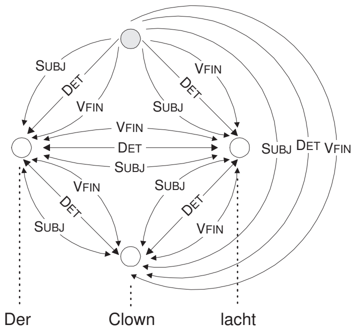


Slide 5

Parsing als Disambiguierungsprozeß

- zuerst werden alle potentiell möglichen Beziehungen ermittelt
- dann Reduktion der potentiellen Strukturen anhand der Grammatikregeln
- Anzahl der möglichen Unterordnungen einer Wortform wächst linear in der Länge der Äußerung
- mehrere Analyseebenen möglich, z.B Syntax, Semantik
- Austausch von Informationen zwischen den Ebenen über wenige Regeln

Slide 6

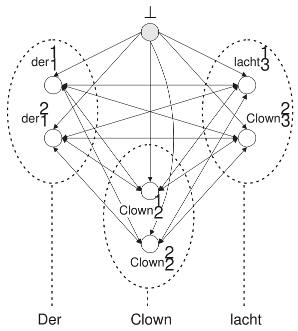


Slide 7

Formale Definition einer CD-Grammatik

- Lexikon
  - Mehrdeutigkeit wird durch mehrere Lexikoneinträge für dieselbe Wortform repräsentiert.
  - Verschiedene Lesarten durch Superskripte markiert
- L-Knoten im Dependenzgraphen der_1^2 = Lesart 2 für "der" an Satzposition 1
- Menge aller potentiellen L-Knoten: $\mathcal{V}$
- zusätzlich zu den L-Knoten ein Wurzelknoten $\perp$

Slide 8



Slide 9

Eine CD-Grammatik besteht aus:

- CD-Lexikon
- $\perp$
- Menge von Analyseebenen r_i
- Menge von Kantenbezeichnungen L_i für jede Analyseebene r_i
- Unäre Bedingungen $c_i^u \subseteq E$
- Binäre Bedingungen $c_i^b \subseteq E \times E$

Slide 10

Beispiel für eine unäre Bedingung:

$det - modifies - noun(X) \Leftrightarrow$
$X.level = SYN \wedge X.label = DET \wedge \neg root(X \uparrow) \rightarrow$
$X \downarrow cat = det \wedge X \uparrow cat = n.$

"Ein Determinierer (z.B.Artikel) modifiziert ein Nomen genau dann, wenn folgendes gilt:
Wenn Kante X auf der Syntax-Analyseebense die Bezeichnung 'DET' trägt und der obere Knoten nicht der Wurzelknoten ist, dann ist die Kategorie des unteren Knotens "determinierer" und die Kategorie des oberen Knotens "nomen".

Slide 11

Strukturbeschreibungen

- Eine Strukturbeschreibung ist eine Baumstruktur für jede Analyseebene
  - Alle Kanten einer Analyseebene bilden Baumstruktur mit $\perp$
  - Für jede Wortform gibt es genau einen L-Knoten (Vollständigkeit, Eindeutigkeit)
- Erfüllung aller Regeln nicht unbedingt erforderlich
  - Eine verletzte Bedingung heißt *Konflikt*
  - Eine Kantenmenge heißt *konfliktfrei*, falls keine Kante einen Konflikt verursacht
  - Strukturbeschreibungen sind *wohlgeformt*, falls sie keinen Konflikt verursachen

Slide 12

Dependenzanalyse

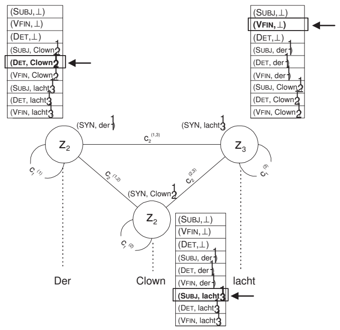
- eine *Dependenzanalyse* ist die Menge aller wohlgeformten Strukturbeschreibungen (bei mehrdeutigen Äußerungen mehr als eine Strukturbeschreibung möglich)

Slide 13

Parsing von CD-Grammatiken

- Ermittlung aller wohlgeformten Strukturbeschreibungen
- Methode: Lösung eines *Constraint Satisfaction Problems*
- Die Constraints sind durch eine CD-Grammatik definiert
- Gesucht ist eine partielle Belegungsfunktion für L-Knoten und Kantenbezeichner
- Partiell, da sich für eine Lesart entschieden werden muß
- Zyklenfreiheit für den Dependenzgraphen gefordert

Slide 14



Slide 15

Bewertete CD-Grammatiken

Menge aller Strukturbeschreibungen mit Hilfe einer Kostenfuktion geordnet

- Bestrafung von Regelverletzungen
- Kosten einer Strukturbeschreibung
- Parsing als Optimierungsproblem

Slide 16

Bestrafung von Regelverletzungen

- Erweiterung der Grammatik
- Jeder Bedingung c wird eine Bewertung $\Phi(c)$ zwischen 0 und 1 zugeordnet
- Je kleiner die Bewertung, desto schwerer die Regelverletzung
- Durch $\Phi(c) = 0$ werden harte Constraints formuliert
- Durch de Einführung von weichen Constraints können
  - ungrammatische Äußerungen geparst und
  - bevorzugte Strukturen honoriert werden

Slide 17

Beispiel (Topikalisierung):

"Das Kind sucht seine Mutter"

Interpretation 1: Subjekt = "das Kind" bevorzugt gegenüber

Interpretation 2: Subjekt = "seine Mutter"

Die für die zweite Interpretation relevante Regel wird mit einer leichten Strafe versehen, z.B. $\Phi(c) = 0.9$

Slide 18

Kosten einer Strukturbeschreibung

- Die Kosten für jede Regelanwendung werden miteinander multipliziert ($= \Phi(E)$, E Kantenmenge)
- Die Kosten von E sind:

$$k(E) = \begin{cases} 0 & \text{wenn } E \text{ konfliktfrei ist.} \\ 1 - \Phi(E) & \text{sonst.} \end{cases}$$

- Optimale Strukturbeschreibungen: Wird keine Bedingung verletzt, sind die Kosten minimal (=0)
- Unzulässige Strukturbeschreibungen: Die Verletzung einer harten Bedingung verursacht immer maximale Kosten (=1)
- Die Kosten wachsen monoton mit der Anzahl der Kanten

Slide 19

Parsing als Optimierungsproblem

- Das CSP ist NP-vollständig
- Branch and Bound mit Speicherung einer begrenzten Anzahl von Suchzuständen in einer Suchagenda
- Ausgehend von einer suboptimalen Lösung die Konfliktstellen schrittweise eliminieren

Slide 20

Fehlerdiagnose

- Definition des Diagnosebegriffs
- Modellierung
- Fehlererklärung

Slide 21

Definition des Diagnosebegriffs

- Aufteilung der Regeln in zwei disjunkte Mengen
  1. Harte und präferenzbasierte weiche Regeln
  2. Für die Diagnose relevante weiche Regeln
- Die in einem Konflikt beteiligten Kanten für Differenzierung von Lesarten und Generierung von Fehlererklärungen
- Eine Äußerung ist dann korrekt, wenn keine der Diagnoseregeln (2.) verwendet wurde
- Verschiedene Diagnosen für dieselbe Äußerung möglich

Slide 22

Modellierung

- Verletzte *Präferenzbedingungen* sind keine Fehler, aber die Einführung von Präferenzbedingungen kann die Analyse beschleunigen
  Beispiel: "Ein Artikel befindet sich *möglichst in der Nähe* des modifizierten Substantivs"
- *Partielles Parsing*: Einführung der Kantenbeschreibung IGNORE für überflüssige Konstituenten. IGNORE erhält eine hohe Strafe
  Beispiel: * "Ein der Clown lacht"

Slide 23

- *Existenzforderungen*: Notwendig, da einige Konstituenten obligatorische Komplemente fordern, z.B. braucht ein Substantiv einen Artikel
  - Da alle Bedingungen allquantifiziert sind, muß pro Existenzforderung eine Hilfsebene eingeführt werden
  - Die Regel auf der Hilfsebene wird durch eine weitere Regel mit der Syntaxebene verbunden

Slide 24

Fehlererklärung

- *Fehlerverbalisierung*: Für ein Aktivierungsmuster wird ein Textmuster erzeugt.
  Das Aktivierungsmuster basiert auf
  - einer Strukturbeschreibung und
  - dem zugehörigen Diagnoseergebnis

Beispiel: $violated(det - n - agr, X) \Rightarrow$ "Der Artikel $X \downarrow$ stimmt nicht mit dem Substantiv $X \uparrow$ im Kasus, Numerus oder Genus überein."

Slide 25

- *Fehlerselektion*: Ordnung auf der Diagnosemenge und Reduktion der Diagnosemenge
  - Unterschiedliche Strafen für alle weichen Regeln: Unwahrscheinlich, daß verschiedene Regelverletzungen gleiche Kosten verursachen
  - Gleiche Strafen für bestimmte Regelverletzungen ermöglichen alternative Erklärungen

Slide 26

Beispiel: * “Das Buch liegt auf den Tisch”

Mögliche Erklärungen:

- “Das Verb *liegen* fordert eine lokale Ortsangabe.”
- “Der Artikel *den* und das Substantiv *Tisch* kongruieren nicht im Kasus.”
- “Der Artikel steht im Akkusativ, während das Substantiv im Dativ steht.”
- “Der Artikel und das Substantiv stimmen nicht im Numerus überein.”

Slide 27

Integration von CDG-Parsing und Fehlersimulation

- Motivation
- Verarbeitungsmodell
- Evaluation
- Ein Entwicklungswerkzeug

Slide 28

Motivation

- Modellierungstechniken zur Suchraumreduktion
- Transitivität und Merkmalstransport
- Beschränkung auf binäre Regeln
- Potential des integrativen Ansatzes

Slide 29

Analyseebenen

8 Ebenen (Syntaxebene & 7 Hilfsebenen):

SYN	Wohlgeformtheitsbedingungen
SYN-DET	Substantive benötigen einen Artikel
SYN-PN	Präpositionen benötigen ein Substantiv
SYN-SUBJ	Finite Verben benötigen ein Subjekt
SYN-IOBJ	Bestimmte Verben benötigen ein indirektes Objekt
SYN-DOBJ	Bestimmte Verben benötigen ein direktes Objekt
SYN-PMOD	Bestimmte Verben benötigen eine Präpositionalgruppe
SYN-VPAR	Das Hilfsverb "haben" benötigt ein Partizip

Slide 30

Getestete Grammatiken

Beispielsatz: * "Ich habe mich auf meinem kleinen Stuhl gesetzt"

- G1:
  - keine harten Regeln
  - Laufzeit: über 20 min, wurde abgebrochen
- G2:
  - gleiche Regelbewertungen für einige Regeln
  - Laufzeit: 160 s
- G3:
  - unterschiedliche Regelbewertungen
  - 8 zusätzliche Präferenzregeln bevorzugen bestimmte Lesarten
  - Laufzeit: 100 s

Slide 31

Modellierungstechniken zur Suchraumreduktion

- Harte unäre Regeln schränken Wertebereiche vor der eigentlichen Suche ein
- Reduktion der lexikalischen Mehrdeutigkeit durch unterspezifizierte Merkmale im Lexikon
  - Die Grammatikregeln müssen angepaßt werden
  - Sonderfälle in den Regeln können zu Modellierungsfehlern verleiten
  - Transitivität der Regeln nicht mehr gewährleistet

Slide 32

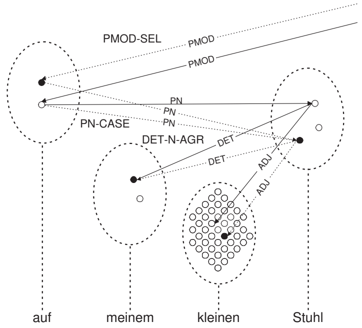
Transitivität und Merkmalstransport

Beispielsatz: * "Ich habe mich auf meinem kleinen Stuhl gesetzt"

- Die Präposition bestimmt den Kasus des Substantivs
- Artikel und Adjektiv kongruieren mit dem Substantiv
- Transitivität der Regeln nicht mehr gewährleistet, wenn unterspezifizierte Merkmale verwendet werden
  → Der Kasusfehler wird nicht erkannt
  → Zusätzliche binäre Regel erforderlich
- Bei mehr als 2 involvierten Kanten und Beschränkung auf binäre Regeln Modellierung der Transitivität nicht mehr möglich

Slide 33

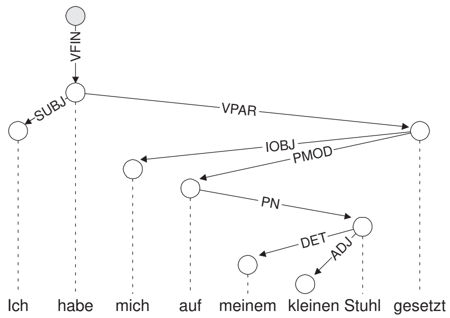
Beispiel: G2 ohne unterspezifizierte Merkmale; verschiedene Lesarten



Slide 34

Reflexivpronomen steht nicht in direkter Beziehung zum Subjekt

Partizip realisiert Person und Numerus nicht



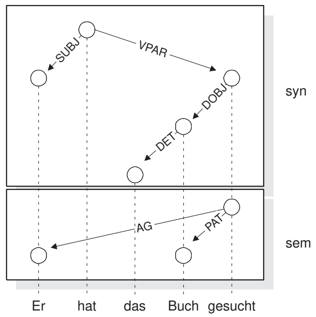
Slide 35

Beschränkung auf binäre Regeln

- Aus Effizienzgründen
- Stellt eine Modellierungsgrenze dar
- Weitere Modellierungsgrenze: Eingeschränkte Verbindung mehrerer Analyseebenen

Slide 36

Identität von Subjekt und Agens erfordert Beziehung zwischen AG, VPAR und SUBJ



Slide 37

Potential des integrativen Ansatzes

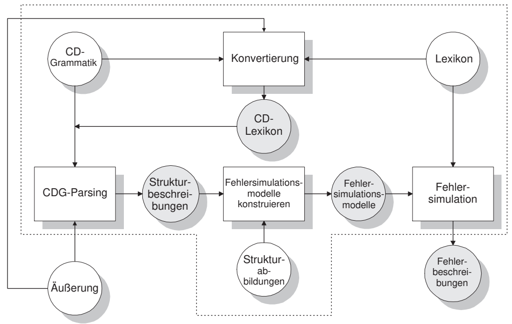
- Fehlersimulation
  - Effiziente Realisierung von transitiven Bedingungsketten
  - Insbesondere für Modellierung von Kongruenzforderungen geeignet
  - Reduziert morphologische Mehrdeutigkeit
    → Entlastung der CDG-Analyse
- CDG-Parsing
  - Strukturbeschreibungen für fehlerhafte Äußerungen
  - Erklärungen für strukturelle Fehler, z.B. fehlende Komplemente oder überflüssige Wortformen

Slide 38

Verarbeitungsmodell

- Strikte Trennung in der Wissensmodellierung zwischen CD-Grammatik und Fehlersimulation
  → Höhere Effizienz
  → Einige Präferenzregeln können nicht modelliert werden wegen Verlagerung in die Fehlersimulation
- Regeln der Fehlersimulation zum Teil schon in der CD-Grammatik enthalten
  → Niedrigere Effizienz
  → Alle Präferenzregeln können modelliert werden

Slide 39



Slide 40

Evaluation

- Grammatik G4
  - Strikte Trennung zwischen CD-Grammatik und Fehlersimulation
  - Keine Modellierung von Kasus, Numerus, Genus und Person in der CDG
- Grammatik G5
  - Keine strikte Trennung zwischen CD-Grammatik und Fehlersimulation
  - Grammatik G3 enthält unterschiedliche Regelbewertungen und zusätzliche Präferenzregeln
  - Strukturabbildungen aus G4

Slide 41

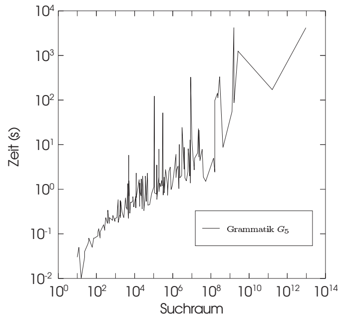
- 84 korrekte Äußerungen mit möglichst allen linguistischen Phänomenen
  - Verben: Präsens, Perfekt, Topikalisierung, Komplemente
  - Substantiv- und Präpositionalgruppen: Adjektive, Artikel, Pronomina, Genitivergänzungen, adverbiale Ergänzungen
- 81 fehlerhafte Äußerungen mit möglichst vielen Fehlerkonstellationen
  - Fehlende obligatorische Konstituenten
  - Überflüssige Konsituenten
  - Anordnung
  - Semantische Sortenverletzungen
  - Kongruenzfehler, Verbflexion, reflexive Verben, Pronomina, Rektion

Slide 42

- Für fehlerhafte Äußerungen produziert G4 in einigen Fällen mehr Strukturen als G5
- In vereinzelten Fällen Fehldiagnosen
  Beispiel: * "Er legt den Ball den Tisch"
  "den Tisch" wird als Genitivergänzung von "Ball" interpretiert anstatt als überflüssige Konsituente

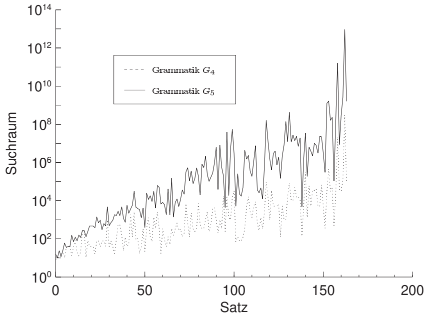
Slide 43

Beziehung zwischen Suchraumgröße und Analysezeit



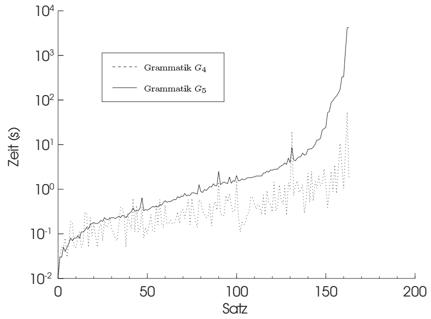
Slide 44

Vergleich der Suchraumgröße



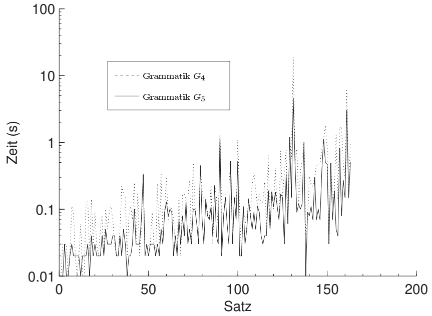
Slide 45

Vergleich der Analysezeiten



Slide 46

Vergleich der Analysezeiten der Fehlersimulation



Slide 47

Ein Entwicklungswerkzeug