Stochastische Lernalgorithmen

Dozent: Dr. Gerhard Jäger
Referentin: Verena Lyding
Datum: 28. Mai 2003

Empirical Tests of the Gradual Learning Algorithm von Paul Boersma und Bruce Hayes

Thema: automatisches Erlernen von OT-Grammatiken

Der Gradual Learning Algorithm ordnet durch Analyse eines Korpus eine gegebene Menge von Constraints – die geordneten Constraints ergeben eine OT-Grammatik.

I Überblick Optimalitätstheorie

Die Optimalitätstheorie (OT) wurde Anfang der neunziger Jahre neben anderen von Alan Prince, Paul Smolensky, John McCarthy entwickelt.
Funktionsweise:

- Generator entwickelt aus einem Input eine Menge von möglichen Oberflächenformen (Kandidaten)
- geordnete Constraints (Beschränkungen) ermitteln den einzigen wohlgeformten Kandidaten (Wettbewerb)

Wie wird der Gewinner gefunden?

- Der Kandidat, der - von dem höchstgeordnetsten Constraint ausgehend - als letztes ein Constraint verletzt, ist der Gewinner.

Kandidaten	A	B	C	D
K_1	*!		*	
K_2		*!		
→ K_3			*	*

II Intention des Papers:

Empirisches Testen des Gradual Learning Algorithm von Boersma; Vergleich mit Constraint Demotion Algorithmus von Tesar und Smolensky

Testen von Lernalgorithmen

1. theoretisch: Kann der Algorithmus jede Grammatik, die den formalen Eigenschaften des Algorithmus genügt, erlernen?
2. empirisch: Sind die Grammatiken, die der Algorithmus lernen kann, sinnvoll um natürliche Sprache zu analysieren?

III The Gradual Learning Algorithm

Algorithmus zum Lernen von OT-Grammatiken durch automatisches ranken von Constraints
- baut auf dem Constraint Demotion Algorithm von Tesar und Smolensky auf
  > ist auch fehlergesteuert, aber erzeugt eine Stochastische Grammatik

GLA arbeitet mit:
kontinuierlicher Ranking-Skala:

- Constraints sind linear organisiert - ohne obere und untere Grenze
- je höher der Wert des Constraints desto wichtiger ist das Constraint

stochastischer Kandidaten-Evaluation:

- ein Constraint ist keine konkreter Wert, sondern eine Spanne um einen Wert (Wert = *ranking value*)
- zur Bewertungszeit wird zufällig ein Punkt aus der Spanne ausgewählt (*selection point*); der *selection point* ist entscheidend für den Ausgang des Wettbewerbs

! zu beachten
- trotz Annahme von Constraint-Spannen ist das Ranking zur Auswertungszeit immer absolut
- C-Spannen werden als normalverteilte Wahrscheinlichkeitsverteilungen realisiert

IV Vorgehensweise des Algorithmus

Initialzustand:
Constraints werden per Hand „vor“-gerankt oder erhalten alle einen identischen Wert

Schritt 1: Dateneingabe
der Algorithmus erhält ein korrektes Input aus dem Korpus

Schritt 2: Generierung
ausgehend von der zugrundeliegenden Form des Inputs ermittelt die aktuelle Grammatik ein Output (Oberflächenform)

Schritt 3: Vergleich
die Eingabe aus dem Korpus wird mit der Ausgabe der Grammatik verglichen
→ Eingabe und Ausgabe stimmen überein >> Grammatik wird beibehalten
→ Ausgabe weicht von der Eingabe ab >> Grammatik muss angepasst werden

Schritt 4: Verbesserung der Grammatik
Die Abweichung der generierten Form impliziert
a.) Constraints, die nur der Kandidat aus dem Korpus verletzt, sind zu hoch gerankt UND/ODER
b.) Constraints, die nur der generierte Kandidat verletzt, sind zu niedrig gerankt
>> alle unter a.) oder b.) fallenden Constraints werden angepasst
>> als Anpassungsfaktor zählt eine festgelegte, relativ kleine Größe - die Plastizität

Plastizität: kleine Plastizität erwirkt akkuratere Ergebnisse; große Plastizität nähert sich der Zielgrammatik schneller > im Verlauf des Lernens die Plastizität verringern

Endzustand
Schritte 1- 4 werden mit unterschiedlichen Lerndaten so lange wiederholt bis sich die Grammatik nicht weiter verändert > der Lernprozess ist abgeschlossen

! zu beachten
- um statistisches Lernen durchführen zu können braucht man wohlformulierte formale Prinzipien (Menge von Constraints)
- nicht jeder statistische Algorithmus ist ausreichend

V Vorteile des GLA gegenüber anderen Lernalgorithmen

- freie Variation kann erlernt werden (Optionalität)
- Fehler im Trainingskorpus beeinträchtigen die Grammatik wenig (Robustheit)
- gradierte Wohlgeformtheitsurteile können widergespiegelt werden

VI Freie Variation in Ilokano

Freie Variation: verschiedene Varianten eines Ausdrucks sind grammatisch

Behauptung: durch GLA erzeugte Grammatiken können freie Variation erfassen

betrachtete Beispiele:
(1) Metathesen: Konsonantendreher bei verwandten Wörtern
Bsp: brennen, burn, Bernstein
(2) Reduplikation: Verdopplung von anlautenden Silben zum Ausdruck morphosyntaktischer Kategorien (z.B. Pluralbildung)
Bsp: pja.no (Sing), pjan.pja.no (Pl)

(1) Metathesen in Ilokano
Daten: /?w/ wird optional zu /w?/
[w] kommt von dem unterliegenden Element /o/

da?o	**/pag-da?o-an/**	→	**pagda?wan bzw. pagdaw?an**
(Baumart)	(Stelle an der da?o's gepflanzt werden)		

(2) Reduplikation in Ilokano
Daten: Verdopplung der wortinitialen Silben zur Pluralbildung; betrachtet werden Silben, die mit Konsonant und Gleitlaut beginnen

bwa.ja	→	bu:.bwa.ja (Vokalisierung des Gleitlauts)
(Krokodil)		bwaj.bwa.ja (gesamte Silbe + C wird kopiert)
		bub.wa.ja (Vokalisierung des Gleitlauts)

Testen des GLA
1. Constraints wurden per Hand gerankt
2. GLA rankte die Constraints; initialer ranking value war 100 -
   - 21.000 Paare wurden eingelesen; gleichhäufiges Auftreten der Varianten
3. Vergleich von Handranking und GLA-Ranking
4. Testen der ermittelten Grammatik
   - jede Form wurde 1000000 mal einglesen
   >> kaum falsche Formen; Frequenzfehler: 0,46%; absolute Fehler 0,39%

! zu beachten:
- erzeugte Grammatik ist kaum noch graduell, da die meisten Constraints weit auseinander liegen; Gradiertheit ist nur bei freier Variation zu beobachten
- Lernproblem bei Variation wird durch Einlesen großer Datenmengen gelöst

Vergleich mit Constraint Demotion Algorithm
- wurde nicht konzipiert um freie Variation erfassen zu können
Funktionsweise des CDA:
- Constraints werden gerankt indem eine Hierarchie von Strata gebildet wird; jedes Constraint in einem höheren Stratum scheidet jedes Constraint des niedrigeren Stratums aus

Testen des CDA:
Aufbau von drei Strata, das vierte konnte nicht aufgebaut werden (Constraints dominieren sich wegen Variation gegenseitig)

mögliche Lösungen:
1. Algorithmus findet keine Lösung in solchen Fällen
2. alle übrigen Constraints bilden ein viertes Stratum

aus 2 folgt > auch nicht wohlgeformte Wörter werden akzeptiert
>> entweder wird eine falsche Grammatik entwickelt oder gar keine

Andere Versionen des CDA schwankten in dem Ranking und richteten es schließlich nach der letzten Eingabe aus > es wurden ebenfalls nicht-wohlgeformte Wörter generiert

FAZIT: CDA ist nicht zur Behandlung von freier Variation geeignet!

<u>Umgang mit Fehlern im Korpus:</u>
GLA:
- ➢ einzelne Fehler wirken sich unwesentlich aus
- ➢ Reihen von gleichen Fehlern spiegeln die Auftretensfrequenzen der Fehler wider

CDA:
- ➢ Fehler wirken fatal (Widersprüche beim Aufbauen von Strata)
- ➢ Fehler machen Grammatik vorübergehend inkorrekt

VII Textfrequenzen am Beispiel von Finnischen Genitiven im Plural

<u>Behauptung</u>: durch GLA erzeugte Grammatiken können die relativen Häufigkeiten von Formen des Korpus widerspiegeln

<u>Vergleich mit Anttilas Modell zu Genitiv-Pluralen im Finnischen</u>
<u>Anttilas Modell</u>
- Organisation der Constraints in Strata; Constraints aus höheren Strata dominieren Constraints aus niedrigeren Strata; innerhalb der Strata sind alle Rankings erlaubt (≠ CDA)
- ein Kandidat gilt als wohlgeformt, wenn er unter einem Ranking, was die Strata beachtet wohlgeformt ist
- Frequenzen werden aus den Anzahlen der Rankings, die einen Kandidaten erzeugen hergeleitet; die Summe dieser Rankings sei proportional zu den Auftretenshäufigkeiten des Kandidaten

Daten:
Genitiv-Plurale können mit Endung /-iden/ und Endung /-jen/ gebildet werden
Bsp: ka.me.roi.den bzw. ka.me.ro.jen (Kameras)

<u>Testen von Antillas Modell:</u>
- Constraints wurden in fünf Strata organisiert
- aus Anzahlen der möglichen Rankings werden die Frequenzen der Kandidaten errechnet

<u>Testen des GLA:</u>
- ermitteln der Constraint-Hierarchie durch den GLA
- Testen der durch 100 000maligen Durchlauf jeden unterliegenden Form

FAZIT: Für Fälle in denen eine Variante viel wahrscheinlicher ist scheint der GLA deutlich besser geeignet als Anttilas Modell.

VIII Mittlere Wohlgeformtheitsurteile am Beispiel von hellem und dunklem /l/ im Englischen

Mittlere Wohlgeformtheit: manche Formen sind weder besonders gut noch ungrammatisch

<u>Behauptung</u>: Urteile für mittlere Wohlgeformtheit kommen von uneindeutigen Frequenzen im Korpus
- selten auftauchende Formen werden als Versprecher eingeordnet
- häufig auftretende Formen werden als grammatisch verstanden
- seltene Auftreten, die aber häufiger sind als Sprachfehler können nicht richtig eingeordnet werden und führen zu mittleren Wohlgeformtheitsurteilen

>> durch Lernen von Frequenzen, kann GLA auch mittlere Wohlgeformtheitsurteile lernen

<u>Ansatz von Hayes und McEachern</u>
- Constraints können unterschiedliche Bandbreiten haben
- Teile der Constraint-Spanne werden als Ränder definiert
- mittlere Wohlgeformtheit wird angenommen, wenn ein *selection point* im Rand liegt

> Analyse muss sprachspezifisch erfolgen

<u>Daten:</u> helles und dunkles /l/
- helles /l/ in initialer Position (Bsp: light)
- dunkles /l/ in finaler Position (Bsp: bell) und vor Konsonanten (Bsp: help)
- in mittlerer pre-atonic Position beides möglich >> gradierte Wohlgeformtheit (Bsp: mailer)

<u>Vorgehen des GLA</u>
Brücke spannen zwischen Frequenzen und Wohlgeformtheitsurteilen > Hypothese in Form von Gleichsetzungen
1. empirisch eingeholte graduelle Wohlgeformtheitsurteile werden in Frequenzen umgewandelt
2. ermittelte Frequenzen werden in Wohlgeformtheitsurteile umgerechnet

empirische Wohlgeformtheitsurteile → errechnete Frequenzen → GLA → Grammatik (vorhergesagte Frequenzen) → errechnete Wohlgeformtheitsurteile
>> Vergleich der Urteile

FAZIT: Versuchsergebnisse bestätigen die Behauptung!