

Promis*D*

Ein Analyseverfahren zur antizipationsfreien Erkennung und Erklärung von grammatischen Fehlern in Sprachlehrsystemen

Veit A. Reuer

1 Einleitung

In der Dissertation wird der Frage nachgegangen, welche Möglichkeiten es zur automatischen, d.h. maschinellen Analyse von Eingaben mit morphosyntaktischen Fehlern in einem Sprachlernsystem gibt. Besondere Beachtung findet außerdem die Gestaltung des Rahmens, in dem die Eingaben vom Lerner gemacht werden können sowie die Form und der Inhalt der aus der Analyse erzeugten Rückmeldungen.

In einer schriftlichen Arbeit hat ein Lehrer¹ vielfältige Möglichkeiten, morphosyntaktische Fehler anzumerken und zu korrigieren. Dabei können u.a. die Berücksichtigung des textuellen Kontextes, das vermutliche Vorwissen des Lerners sowie das Lernziel der Aufgabe eine Rolle spielen. Davon abhängig ist dann wiederum, mit welcher Ausführlichkeit der Fehler angemerkt wird und ob ein Korrekturvorschlag vom Lehrer gemacht wird.

In einer mündlichen Unterrichtssituation dagegen würde der Lehrer einen morphosyntaktischen Fehler vielleicht komplett ignorieren, wenn er das Gefühl hat, dass die Hörer die Intention des Sprechers verstanden haben und der Kommunikationsfluss weiterlaufen kann. Eine weitere plausible Möglichkeit bestünde in dieser Situation in der Formulierung einer so genannten korrigierten Echofrage, die nur einen Hinweis auf den Fehler beinhaltet, ansonsten aber das Ziel der kommunikativen Gesamtsituation weiterführt.

Es zeigt sich also, dass vielfältige Möglichkeiten zur Analyse einer fehlerhaften Äußerung bestehen, die zudem auf vielfältige Art und Weise erklärt und korrigiert werden kann. Idealerweise sollte die Rückmeldung auf eine fehlerhafte Eingabe in einem Sprachlernsystem² genauso lernerorientiert sein, wie sie im Unterricht möglich ist. Sprachdidaktiker, die Computer im Sprachunterricht einsetzen, bemängeln aber als eine der wesentlichen Unzulänglichkeiten in aktuellen kommerziellen CALL-Systemen genau diesen Aspekt der völlig unzureichenden Fehleranalysen und Rückmeldungen (z.B. Rösler und Tschirner (2002, S. 151), Grüner und Hassert (2000, S. 53) u.a.).

Gegenstand der Arbeit ist zunächst eine Analyse der Anforderungen an Sprachlernsysteme, die sich zum Teil aus der Didaktik des Fremdsprachenunterrichts ergeben. Daraus ergibt sich ein Übungstyp, der vom Lerner eine frei gestaltete Eingabe erfordert und damit insbesondere die kommunikative Kompetenz als eines der wesentlichen didaktischen Ziele fördern kann, der aber auch mit Hilfe computerlinguistischer Methoden realisiert werden kann. Dieser Aspekt wird in Abschnitt 2 erläutert.

¹Wenn es meiner Ansicht nach vertretbar ist, geschlechtlich unspezifizierte Formen oder beide Varianten zu benutzen, so werde ich dies tun. Ansonsten ist bei der Verwendung einer geschlechtlich spezifizierten Form die jeweils andere mitzudenken.

²oder CALL-System (Computer-Assisted Language Learning)

Anschließend wird zur Auswahl einer geeigneten Grammatiktheorie insbesondere die Lexical Functional Grammar (LFG, Kaplan und Bresnan 1982; Bresnan 2001) näher betrachtet (Abschnitt 3). Die Theorie muss sich aus computerlinguistischer Sicht für eine Implementierung im Rahmen eines CALL-Programms eignen und es ist von zusätzlichem Vorteil, wenn die verwendeten Konzepte denen in Lernergrammatiken ähneln, um so die Generierung von Rückmeldungen zu vereinfachen. Im darauf folgenden Abschnitt 4 wird kurz das eigentliche Programm so vorgestellt, wie es sich auch dem Nutzer präsentiert.³

Schließlich wird in Abschnitt 5 ein so genanntes antizipationsfreies Verfahren entwickelt, bei dem weder in der Grammatik noch im Lexikon Informationen zur Identifizierung von Fehlerpositionen und -typen enthalten sind. Die Fehlererkennung soll dabei auf die Bereiche eingeschränkt werden, in denen sich in einem Lernerkorpus häufig Fehler zeigen, um einerseits wesentliche Fehlertypen abzudecken und andererseits eine größere Effizienz bei der Analyse von realen Eingaben zu erreichen. Die Vorstellung des Verfahrens unterteilt sich entsprechend den grundlegenden Struktureinheiten der LFG in zwei Bereiche: die Konstituentenstruktur und die Feature-Struktur. Zum Abschluss erfolgt die Evaluation und es werden die Möglichkeiten zur Gestaltung einer Rückmeldung an der Lerner diskutiert.

2 Didaktik und Computereinsatz im Fremdsprachenunterricht

Mit der rasanten Entwicklung der Computertechnologie, der Telekommunikation und dabei insbesondere des WWW hat sich auch die Nutzung von computergestützten Medien im Sprachunterricht stark ausgebreitet. Interessant ist dabei die Ansicht von Fremdsprachendidaktikern wie z.B. Rüschoff und Wolff (1999), dass sich didaktische Konzepte zur Förderung der kommunikativen Kompetenz, der Sprachbewusstheit (*language awareness*) sowie der Bewusstheit über das Lernen einer Sprache (*language learning awareness*) mit Hilfe des Computers in besonderer Weise umsetzen lassen.

Eine Lerntypologie⁴, die psycholinguistisch und didaktisch sinnvolle Lernformen spezifiziert, erlaubt die Bewertung von Software hinsichtlich ihrer Eignung im Fremdsprachenunterricht. Hieraus ergibt sich, dass vor allem Programme geeignet sind, die entweder einen sehr offenen Charakter haben, d.h. die Informationen zusammengestellt nach unterschiedlichsten Kriterien zur Verfügung stellen können, oder die mit Hilfe des Internets die Bearbeitung authentischer Texte sowie die „authentische“ Kommunikation mit Personen (Email, Chat) erlauben. Diese Software wird z.B. von Rüschoff und Wolff (1999, S.67ff) in 3 Kategorien eingeteilt:

1. Werkzeuganwendungen, die sich mit nahezu beliebigen Inhalten füllen lassen und daher vielfältige Nutzungsmöglichkeiten bieten, 2. Anwendungen, in welchen die Neuen Technologien als Ressourcen für das fremdsprachliche Lernen genutzt werden. Diese grenzen sich insofern von den Werkzeuganwendungen ab, als dass hier in den sogenannten Informations- und Kommunikationstechnologien und den dazugehörigen Werkzeugen eine gesonderte Klasse von Anwendungen gesehen wird und 3. die Telekommunikation, zu der offensichtlich Anwendungen wie z.B. Email oder Chat zählen, die im Gegensatz zum WWW eine direkte Kommunikation zwischen Menschen erlauben.

Aus diesen Betrachtungen ergibt sich, dass insbesondere die Übungsformate in herkömmlichen „tutoriell orientierten Anwendungen“, welche eine weitere Kategorie darstellen, keine relevante Unterstützung für die erwähnten Lerntypen leisten können, wenn auch verschiedene Autoren diesen so genannten „drill&practice“-Übungen einen gewissen Platz einräumen. Zu dieser Gruppe von

³Realisiert wurde die Implementation unter dem Namen *PromisD*, das für „Projekt mediengestütztes interaktives Sprachenlernen - Deutsch“ steht.

⁴Dazu gehören nach Rüschoff und Wolff (1999, S. 66) 1. Inhaltsbezogenes/aufgabenorientiertes Lernen mit authentischen Materialien und Aufgabenstellungen; 2. Projektbasiertes/prozessorientiertes Lernen mit authentischen Interaktionsformen; 3. Kognitiv-konstruktivistisches Lernen mit eigenständigem/eigenverantwortlichem Wissenserwerb und 4. Lernen in einem offenen und multimodalen Umfeld.

Programmen zählen Systeme, die vor allem dazu dienen, Lexeme und grammatische Strukturen zu üben und somit eine stark tutorielle Komponente besitzen. Ein zweiter Gesichtspunkt, der zu dieser Einteilung führt und damit auch PromisD einschließt, ist die Geschlossenheit des Systems; das Design des Programms verhindert eine Erweiterung der Inhalte oder eine Veränderung der Übungstypen.

Die oben angeführten Programme der ersten drei Kategorien haben den Nachteil, dass sie sich effektiv nur in Unterrichtssituationen, in denen ein Lehrer gezielt zur Projektarbeit und zur „Exploration“ einer Sprache anleitet, anwenden lassen. Insbesondere in Selbstlernsituationen sind dagegen tutoriell orientierte Programme notwendig, die „stand-alone“ zum Umgang mit der Sprache anleiten. In diesem Zusammenhang unterscheidet sich der Übungstyp des simulierten Dialogs in PromisD wesentlich von denen herkömmlicher, tutoriell orientierter CALL-Programme, in denen üblicherweise die Interaktion in einer Aufgabenstellung maximal mit der Auswahl aus einer Liste von vorgefertigten Antworten besteht oder in denen die Aufgabenstellung so stark eingeschränkt ist, dass praktisch keine Alternativen sinnvoll sind. Der Übungstyp bietet sich gerade auch in Selbstlernsituationen an, da er die Forderung der Didaktik nach kommunikativen Lernsituationen ergänzen kann. Die Prozessorientiertheit der Bearbeitung einer Dialogaufgabe ergibt sich, da im vorliegenden Rahmen eine Aufgabe genannt werden kann, die über einen längeren Zeitraum hinweg in einem kontextuell geschlossenen Rahmen verfolgt wird.

In Bezug auf Fehlerfeedback, das durch die computerlinguistische Analyse ermöglicht wird, lassen sich drei weitere Anforderungen an das Programm formulieren. Um den Lerner beim Verständnis von fremdsprachlichen Strukturen und bei der Erfassung des eigenen Lernstandes zu unterstützen, sollte ein möglichst ausführliches und für den Lerner verständliches Fehlerfeedback erfolgen. Außerdem wird gefordert, das Fehlerfeedback in Bezug auf den Kommunikationsfluss in der momentanen Aufgabenstellung anzupassen. Schließlich sollte ein Programm in Hinblick auf die inhaltliche Gestaltung der Rückmeldungen insbesondere auch die Möglichkeit der Anpassung an Lernerkenntnisse bieten.

3 Grammatikmodellierung

Drei Anforderungen an eine Grammatiktheorie müssen m.E. erfüllt sein, die in einem ICALL-System, das fehlerhafte Lernereingaben analysieren können soll, zum Einsatz kommt: 1. Die Grammatiktheorie sollte eine gewisse linguistische Adäquatheit besitzen; 2. die Theorie muss formal fundiert sein, damit sie implementierbar ist, und 3. sollte die Theorie Analysen liefern können, die in für Lerner verständliche Meldungen über morphosyntaktische Sachverhalte umgesetzt werden können. Ich möchte in diesem Abschnitt kurz andeuten, warum u.a. die LFG diesen Anforderungen genügen kann. Auf die Möglichkeiten zur Fehlererkennung und zur Erzeugung von Feedback wird in Abschnitt 5 eingegangen.

Die auf transformationellen Ansätzen basierenden Theorien beanspruchen üblicherweise eine gewisse Erklärungsadäquatheit in Bezug auf einen idealen Sprecher/Hörer und in Bezug auf die universalen Eigenschaften der natürlichen Sprachen. Dagegen nimmt die LFG für sich unter anderem in Anspruch, besonders in typologischer Hinsicht weitreichende Beschreibungen auf der Basis der strukturellen Eigenschaften der Theorie liefern zu können.

Ein anderer Gesichtspunkt ist die Modularität der Theorie. Um die Herkunft eines bestimmten linguistischen Phänomens erklären zu können, kann es sich anbieten, eine Theorie in Module bzw. strukturelle Ebenen zu unterteilen und damit eine angemessene Beschreibung zu erreichen.

Die LFG zeichnet sich außerdem dadurch aus, dass die zur Sprachbeschreibung verwendeten Grundstrukturen von vornherein formal fundiert konzipiert sind (cf. Dalrymple et al. 1995). Sowohl annotierte PS-Regeln als auch Feature-Strukturen (z.B. Karttunen 1984) sind schon längere Zeit wohldefinierte Mittel, die als formal abgesichert angesehen werden können. Auch die in der LFG-

Theorie eingesetzten Erweiterungen wie z.B. „Functional Uncertainty“ oder Adjunkt-Listen als Werte in Feature-Strukturen sind innerhalb eines formalen Ansatzes entwickelt worden.

In einem exemplarischen Vergleich von Terminologie und Strukturierung in deskriptiven Grammatiken sowie in der LFG hat sich gezeigt, dass in den vorgestellten Bereichen die Modellierung der linguistischen Phänomene in LFG ausreichend Übereinstimmung mit der Beschreibung derselben Phänomene in den Grammatiken existiert, um ein für den Lerner relevantes Feedback erzeugen zu können. Insbesondere für den Bereich des funktionalen Aufbaus und den damit verbundenen Kongruenz- und Rektionsbeziehungen eines Satzes lassen sich präzise Meldungen über die Struktur bzw. über Fehler in der Struktur erzeugen. Hervorzuheben ist schließlich die Betonung des Lexikons als hauptsächliche Datenbasis, aus der u.a. für den Unterricht abgeleitet werden kann, dass das Vokabellernen insbesondere von Verben möglichst auch die spezifischen Subkategorisierungseigenschaften der einzelnen Lexeme mit umfassen sollte (Hubbard 1994, S. 59f).

4 Allgemeine Systembeschreibung

Das Programm ist so konzipiert, dass dem Lerner Dialogsituationen präsentiert werden, die ein selbstständiges Formulieren des Lerners erfordern. In diesen Frage-Antwort-Dialogen wird der Nutzer aufgefordert, in verschiedenen Situationen, auf die Fragen des Systems mit natürlichsprachlichen Sätzen zu antworten. Beispielsweise wird der Lerner in einem Dialog aufgefordert, einen Unfall zu melden, der ihm vorher grafisch präsentiert worden ist.⁵

In *PromisD* wird der Lerner gefordert, sein eigenes Wissen zum Gebrauch der Sprache zu aktivieren. Die Leistung des Lerners liegt einerseits im Verständnis einer konkreten Frage und andererseits in der vollständig freien Produktion eines Antwortsatzes. Das System unterstützt den Lerner dabei, indem ein plausibler Kontext für den Dialog und eine präzise Rückmeldung aufgrund einer ausführlichen computerlinguistischen Analyse geliefert werden kann.

Um die für die Übungsform „Dialog“ im Unterricht bestehende Empfehlung (cf. Kleppin 1998), Unterbrechungen wegen Fehlern nur vorzunehmen, wenn die Kommunikation scheitert bzw. Fehlerhinweise so zu gestalten, dass der Kommunikationsfluss nicht unterbrochen wird, zu simulieren, läuft auch in *PromisD* der Dialog für den Lerner ununterbrochen weiter, soweit das System in der Lage ist, eine Fortsetzung des Dialogs zu bestimmen. Wenn ein morphosyntaktischer Fehler diagnostiziert wird, aktiviert das Programm nur einen Fehlerknopf, der dann durch einen roten Rand markiert ist, weshalb der Lerner selber entscheiden kann, ob er sich den Fehler anschauen oder lieber den Dialog weiterführen möchte. Bei semantischen Fehlern bzw. Fehlern, die durch die Limitierung des Dialoginhalts ausgelöst werden, reagiert das System nicht mit einer „externen“ Fehlermeldung, sondern die Reaktion des virtuellen Gesprächspartners repräsentiert das Scheitern der Interpretation der Eingabe.

Mit der Möglichkeit, für einen morphosyntaktischen Fehler eine ausführliche Rückmeldung zu präsentieren, wird eine zweite didaktische Anforderung erfüllt: Die Meldungen zur Morphosyntax können die geforderte Sprachbewusstheit fördern, wenn der Lerner in geeigneter Weise auf die Meldungen eingeht. Zusätzlich kann sich der Lerner in der Auswertung einen präzisen Überblick über seinen Sprachstand zumindest in Bezug auf die verwendeten Konstruktionen verschaffen.

⁵Anregungen zu ähnlichen Dialogübungen finden sich auch in vielen Sprachlehrwerken (z.B. Themen 1997, S. 93ff).

5 Behandlung von fehlerhaften Eingaben

Das Ziel der Analyse eines fehlerhaften Satzes in diesem Kontext ist es, eine Beschreibung und gegebenenfalls eine Korrektur zu liefern, obwohl das aufgrund der linguistischen Datenbasis, d.h. aufgrund der Grammatik und des Lexikons für korrekte Sätze, nicht möglich ist. Zwei Strategien bieten sich an.

1. Die Grammatik wird derart verändert, dass mit Hilfe von hinzugefügten Fehlerregeln oder so genannten Meta-Regeln eine Beschreibung *antizipierend* generiert werden kann. Wenn eine Grammatikregel verwendet wird, um eine fehlerhafte Form zu beschreiben, besteht ein wesentlicher Nachteil darin, dass möglicherweise eine Generalisierung des Fehlers verloren geht. Die gleiche Form des Fehlers an anderer Position im Satz muss dann erneut mit einer weiteren Regel beschrieben werden. Ein weiterer Nachteil eines solchen Verfahrens besteht in der Notwendigkeit, möglichst viele Fehlertypen antizipieren zu müssen, wenn eine nahezu freie Eingabe durch das System analysiert werden können soll. Damit wird die Grammatik mit vielen Regeln sehr stark „aufgebläht“, was wiederum zu Effizienz- und möglicherweise auch Wartungsproblemen führen kann. Neben den aufgeführten Nachteilen besteht ein Vorteil der Antizipation von Fehlern in der Möglichkeit zur Verwendung von bekanntermaßen effizient arbeitenden Verfahren zur Analyse von natürlicher Sprache. Einzige Bedingung ist, dass sich der Ort der Markierung von Fehlermöglichkeiten eindeutig bestimmen lässt (cf. Krüger-Thielmann 1992, für französische Zahlwörter).

Ein weiterer Vorteil eines antizipationsbasierten Ansatzes gegenüber der Nicht-Antizipation ist die Möglichkeit zur Unterscheidung von fehlerhaften und nicht von der Grammatik abgedeckten Eingaben, wodurch ein identifizierter Fehler mit der entsprechenden Sicherheit an den Lerner weitergegeben werden kann.

2. Der Analysealgorithmus wird derart verändert, dass er trotz der Verwendung einer Grammatik und eines Lexikon, die keine Informationen über mögliche Fehlertypen und -positionen enthalten, eine Beschreibung *antizipationsfrei* liefern kann. Ein erster Vorteil aus dem Einsatz von antizipationsfreien Verfahren besteht in der Möglichkeit, bestimmte Fehlertypen, wenn sie generell vom System erkannt werden, an potentiell beliebigen Positionen identifizieren zu können. Beispielsweise wird eine Inkongruenz zwischen Subjekt und Verb im Deutschen nicht nur erkannt, wenn das Subjekt vor dem finiten Verb steht, sondern auch, wenn der Satz ein topikalisierendes Element enthält und das Subjekt erst nach dem finiten Verb folgt und damit eine andere PS-Regel angewendet wird. Mit Hilfe sehr genereller Verfahren zum Parsen mit einfachen PS-Regeln können auch Wortfolgefehler an *beliebigen* Positionen erkannt und gegebenenfalls korrigiert werden.⁶ Als Beispiel sei Kato (1994) genannt, mit dessen Verfahren Einfügungen, Auslassungen und Ersetzungen erkannt werden können.

Ein weiterer wichtiger Vorteil besteht in der Möglichkeit zur unabhängigen Entwicklung einer Grammatik und eines Lexikons. Die Grammatik und das Lexikon sind dann so konstruiert, dass sie tatsächlich nur korrekte Sätze beschreiben. Hieraus ergibt sich die Möglichkeit zur Integration von „fremden“ Daten, die zum Beispiel in anderen Zusammenhängen entwickelt wurden. Im Rahmen dieser Arbeit wird daher eine *antizipationsfreie* Methode zur Fehlererkennung entwickelt.

Während bei antizipationsbasierten Systemen der Aufwand zur Erkennung von fehlerhaften Eingaben in der Grammatik und den entsprechenden Fehlerregeln liegt, muss in einem antizipationsfreien System dieser Aufwand in das Erkennungsverfahren übertragen werden. Aufgrund der komplexen Verfahren, die relativ viele Hypothesen produzieren und damit die Systeme potentiell ineffizient machen, stellt die Eingrenzung des Suchraums einen besonders schwierigen Aspekt bei der antizipationsfreien Analyse dar. Da solche Systeme den Suchraum an (fast) beliebigen Positionen für

⁶Zu *Wortfolgefehlern* zähle ich nicht nur Wortstellungsfehler im engeren Sinne, d.h. Fehler, bei denen die Reihenfolge vertauscht ist, aber das gesamte lexikalische Material vorhanden ist. Zusätzlich möchte ich zu diesem Fehlertyp auch Auslassungs-, Einfügungs- und Ersetzungsfehler zählen.

potentielle Fehler öffnen müssen, sollten geeignete Verfahren entwickelt werden, die den Suchraum von vornherein in möglichst restriktiver Weise einschränken. Gefunden werden soll ja im Allgemeinen eine möglichst global minimale Fehlerhypothese.

Im Folgenden wird in zwei getrennten Abschnitten auf die beiden Bereiche eingegangen, die in der LFG für den Aufbau der Beschreibungen genutzt werden: Phrasen-Struktur-Parsing und Feature-Unifikation. Schließlich zeige ich, wie aus den Analysen der fehlerhaften Eingaben die Informationen für das Feedback an den Lerner gewonnen werden können.

5.1 Phrasenstruktur-Parsing: Fehlererkennung in der K-Struktur

Der vermutlich bekannteste Ansatz zur Fehlererkennung mit Hilfe von Phrasenstruktur-Regeln wurde in Mellish (1989) vorgestellt. Darauf aufbauend wurde in Kato (1994) ein verbessertes Verfahren vorgestellt, das einige Schwierigkeiten von Mellishs Ansatz vermeidet. Der wesentliche Vorteil eines solchen generellen Verfahrens ist, dass auf diese Weise nahezu beliebige Fehler, wie Einfügungen, Auslassungen und Ersetzungen erkannt werden können. Ein Nachteil besteht in der Anforderung des doppelten Parses. Allerdings werden korrekte Sätze ohne zusätzlichen Aufwand geparkt. Schwerer noch wiegt der Nachteil, dass das Parsing-Verfahren nicht in der Lage ist, verschobene Elemente zu erkennen und an geeigneter Stelle wieder einzufügen. Diese Art von Fehler hat sich aber in den von mir untersuchten Belegen als relativ häufig erwiesen.

Die von mir entwickelte Methode zur Erkennung von Fehlern ähnelt der von Lyon (1974). In beiden Ansätzen wird eine Änderung der *Scan*-Prozedur des Chart-basierten Earley-Algorithmus (Earley 1970) vorgenommen. Bevor die *Scan*-Prozedur aufgerufen wird, werden in diesem Algorithmus sämtliche Hypothesen, d.h. so genannte aktive Kanten zur Chart hinzugefügt, die durch die Grammatik lizenziert sind. Im Anschluss daran wird normalerweise mit Hilfe der *Scan*-Prozedur eine passive Kante für das folgende Wort in die Chart eingefügt und der Versuch unternommen, so viele Hypothesen wie möglich zu erfüllen. In der modifizierten Form werden zusätzlich passive Kanten zur Chart hinzugefügt, die zwar nicht durch die verwendete Grammatik lizenziert sind, die aber eine mögliche, lokale Fehlerquelle beschreiben.

Im Gegensatz zu Lyon, in dessen Ansatz unrestringiert Kanten, die Fehlerhypothesen darstellen, zur Chart hinzugefügt werden, wird in meinem Konzept zunächst die Chart auf eventuelle Hinweise auf Fehler untersucht. Bevor nun der nächste *Scan*-Schritt folgt, wird die Chart auf die in ihr enthaltenen Typen von aktiven Kanten untersucht. Kanten, die Fehlerhypothesen repräsentieren, werden daher nicht in jedem Fall zur Chart hinzugefügt, sondern nur, wenn es Grund zur Annahme (eine bestimmte „Kantenkonstellation“) eines Fehlers gibt. Die folgenden Kantenkonstellationen können als Hinweise auf Fehler in Betracht kommen:

Erläuterung: eine Chart C enthält Element der Form $\text{item}(_begin, _end, _lhs, _closed, _open)$; e bedeutet leer; ein Wort w_j des Eingabesatzes befindet sich an der Position $1 \leq j \leq n$ mit der lexikalischen Kategorie B .

- Das aktuelle Lexem konnte nicht in die Chart integriert werden. D.h., es gibt kein aktives Element in der Chart, das ausgehend vom aktuellen Lexem eine offene Hypothese enthält.
 $(\text{item}(_j, _-, _-, \beta) \text{ mit } \beta \neq e) \notin C$
- Es gibt aktive Elemente, die das aktuelle Lexem überspannen, aber keine Hypothese lässt sich mit dem folgenden Wort verbinden.
 $\text{not}((\text{item}(_j, _-, _-, \beta) \text{ mit } \beta \neq e) \in C, \langle \text{LINK}(\beta, w_{j+1}) \rangle)$
- Es gibt aktive Elemente, die das aktuelle Lexem überspannen, aber keine aktiven Elemente, die nach dem aktuellen Wort beginnen.
 $(\text{item}(j, _-, _-, \beta) \text{ mit } \beta \neq e) \notin C$

Wenn einer dieser Tests positiv ist, wird eine passive Kante $\text{item}(j, j, \text{Cat}, w, e)$ zur Chart hinzugefügt, die eine Auslassungsfehlerhypothese repräsentiert und es wird versucht, mit Hilfe dieser Kante Hypothesen zu erfüllen, d.h. diese Kante in die Chart zu integrieren. Anzumerken ist, dass nur bestimmte Wortarten verwendet werden, deren Auswahl durch im Korpus häufige Auslassungsfehler

bestimmt wird.

Wenn der erste Test positiv ist, wird außerdem ein Verschiebungsfehler angenommen. Ab der aktuellen Position $j - 1$ wird ein zusätzlicher Parsevorgang begonnen, um die Länge k der Phrase zu bestimmen, die von der aktuellen Position ausgehend verschoben werden soll. Wenn die Länge bestimmt ist, werden die Lexeme dieser Kette in einen Speicher gelegt und die „Lücke“ gefüllt, indem das dem aktuellen Wort vorhergehende Element „verlängert“ wird $\text{item}(j-2,k,B_{j-1},w_{j-1},e)$. Danach muss offensichtlich ein Update der Chart erfolgen, indem diese neue Kante zur Erfüllung von Hypothesen verwendet wird. Falls sich Lexeme im Speicher befinden, versucht der Parser diese zu einem späteren Zeitpunkt in die Analyse zu integrieren. Beide Modifikationen der Analyse führen zu einer Erhöhung des mit der Kante assoziierten Fehlerwertes, der bei der Auswahl eines Ergebnisses am Ende des Parsevorgangs genutzt wird.

Es gibt allerdings Wortfolgefehler im weiteren Sinne, die auf diese Weise nicht erkannt werden können. Dabei handelt es sich um Auslassungen von Aktanten –also eher Valenzfehler–, wobei aber die tatsächliche Satzstruktur nicht den PS-Regeln widerspricht. Valenz wird in der LFG mit Hilfe von zwei wichtigen Prinzipien zur Form von F-Strukturen geregelt: dem Vollständigkeits- und dem Kohärenzprinzip (Kaplan und Bresnan 1982, S. 211f). In dem hier implementierten System werden sowohl die Vollständigkeit als auch die Kohärenz erst am Ende des Parsevorgangs getestet. Auf diese Weise kann das System über eventuell überflüssige Argumente Auskunft geben und auf eine mögliche Fehlerquelle hinweisen.

5.2 Unifikation: Fehlererkennung in der F-Struktur

In diesem Abschnitt soll es um die Erkennung von Fehlern in der Kongruenz, Rektion und anderen Bereichen der Verbeigenshaften wie zum Beispiel der Finitheit gehen. Aus der strukturellen Perspektive geht es um Fehler, die in der F-Struktur auftreten. Es soll ein Verfahren gezeigt werden, bei dem sich widersprechende Werte von Attributen in der F-Struktur bewahrt werden, damit sie am Ende des Parse-Vorgangs zur Erzeugung einer präzisen Fehlermeldung genutzt werden können. Es hat verschiedene Vorschläge zur Behandlung von sich widersprechenden Werten in Feature-Strukturen gegeben, wie z.B. Schwind (1988), Carpenter (1993) oder Vogel und Cooper (1995). In diesen Ansätzen werden allerdings Verfahren beschrieben, die entweder zu inkonsistenten Strukturen führen, nicht monotone Erweiterungen des Unifikationsalgorithmus darstellen oder die Informationen zu den sich widersprechenden Werten nicht in die Feature-Struktur integrieren und damit nicht für ein späteres Feedback speichern. In dem hier vorgestellten Konzept dagegen kann jede komplexe Feature-Struktur mit einer Markierung, d.h. mit einem speziellen Fehler-Attribut versehen werden, das einen Teil der konfigrierenden Attribute und deren atomare Werte enthält. Mit Hilfe dieses speziellen Attributs werden die Informationen des Clashes in der Feature-Struktur gespeichert und können für ein Feedback zu den aufgetretenen Fehler genutzt werden.

Definition 5.1 (Informationserhaltende Subsumption \sqsubseteq_{err}) *Eine Feature-Struktur F subsumiert eine Feature-Struktur F' genau dann, wenn gilt:*

- F und F' sind atomar und $F = F'$ oder
- F und F' sind komplex und es gilt:
 - jeder Pfad der Länge 1 in F ist auch ein Pfad in F' und entweder
 - * der Wert des Pfades in F ist gleich dem Wert des korrespondierenden Pfades in F'oder
 - * der Wert des Pfades in F ist ungleich dem Wert des korrespondierenden Pfades in F' und das Attribut-Wert-Paar ist in der angefügten Fehlerliste enthalten
 - Jeder Pfad der Länge > 1 in F ist auch ein Pfad in F' . Der Wert eines Pfades in F subsumiert den Wert des mit ihm in F' korrespondierenden Pfades.
 - Alle Pfade, die sich in F einen gemeinsamen Wert teilen, teilen sich auch in F' einen Wert.

Diese Definition von Subsumption unterscheidet sich im Wesentlichen nur in einem Punkt von den Definitionen zum Beispiel in Naumann und Langer (1994): Hier wird zwischen Attributen mit atomaren Werten und Attributen mit komplexen Werten unterschieden. Die Definition der Subsumption hat daher die gewünschten Eigenschaften wie Reflexivität und Transitivität. Allerdings ist die Relation nicht vollständig antisymmetrisch. Zwei Feature-Strukturen können sich gegenseitig subsumieren und sind unter Umständen doch nur äquivalent und nicht identisch.

Die Standarddefinition der Unifikation muss nicht geändert werden.

Definition 5.2 (Unifikation \sqcup) Das Ergebnis der Unifikation von zwei Feature-Strukturen F und F' ist definiert als die kleinste obere Schranke in der Menge der Feature-Strukturen geordnet nach der Subsumptionsrelation.

Mit diesen Definitionen bleibt die Monotizität der Unifikation erhalten, was nicht nur eine erwünschte Eigenschaft der Unifikation ist, sondern auch in der LFG-Theorie angestrebt wird. Die Unifikation entspricht im fehlerfreien Fall der Standard-Unifikation. Im Falle eines Fehlers ergibt sich aus der Unifikation formal eine Äquivalenzklasse an Ergebnissen, wobei in der praktischen Implementierung aber nur eine Struktur ausgewählt wird, da sich die Mitglieder der Klasse gegenseitig subsumieren. Auch hier ist die Fehlermöglichkeit auf bestimmte Features, die sich als häufige Fehlerquellen gezeigt haben, beschränkt. Das folgende Beispiel kann das Verfahren verdeutlichen.

$$(1) \quad \left[\text{subj: } \left[\text{num: pl} \right] \right] \sqcup \left[\begin{array}{l} \text{pred: PRED}(\uparrow \text{subj}) \\ \text{subj: } \left[\text{num: sg} \right] \end{array} \right] = \left[\begin{array}{l} \text{subj: } \left[\begin{array}{l} \text{num: pl} \\ \text{err: } \left[\text{num:sg} \right] \end{array} \right] \\ \text{pred: PRED}(\uparrow \text{subj}) \end{array} \right]$$

Die F-Struktur des Subjekts mit Numerus Plural wird unifiziert mit der F-Struktur des Verbs, das aber ein Subjekt im Singular fordert. Als Konsequenz wird ein entsprechender Eintrag in der neuen F-Struktur erzeugt. Zur Bestimmung einer optimalen Lösung wird wiederum der Fehlerwert der Kante genutzt, der sich mit jedem widersprechenden Wert erhöht. Die Kante mit dem geringsten Fehlerwert stellt dann am Ende des Parsevorgangs das Ergebnis dar, das für die Reaktion des Systems genutzt wird.

Auf einen wesentlichen Unterschied zum Parsen mit PS-Regeln sei hier hingewiesen. In den F-Strukturen wird zwar ein Werte-Mismatch festgehalten, wie eine Korrektur der Lexeme aussieht, wird aber nicht ermittelt. Die Korrektur muss in einem anschließenden Schritt ausgeführt werden. Im Gegensatz dazu wird beim Parsen mit PS-Regeln eine Korrektur sofort durchgeführt; in der vollständigen Beschreibung der Eingabe befindet sich kein Hinweis mehr auf die ursprüngliche Form. Damit trotzdem ein Feedback gegeben werden kann, wird die Kante wie erwähnt so erweitert, dass sie auch Informationen über die Korrektur enthält.

5.3 Evaluation

Zur Evaluation wurden die Grammatik und das Lexikon so erweitert, dass die korrekten Versionen der Sätze im Evaluationskorpus⁷ abgedeckt wurden, d.h., dass der Parser mit abgeschalteter Fehlererkennung für jeden korrekten Satz mindestens eine Analyse generieren konnte. Daraus folgt, dass in dieser Untersuchung keine so genannten False-Positives auftreten konnten. Im Anschluss wurden die fehlerhaften Sätze geparkt. Die folgende Tabelle 1 bietet einen Überblick über die erzielten Ergebnisse.

⁷Das Evaluationskorpus besteht einerseits aus fehlerhaften Sätzen, die von DaF-Lernern der HU und der FU Berlin bei der Benutzung dieses Systems eingegeben worden sind und andererseits aus Sätzen, die sich in einem Fehlerkorpus befinden, das von H.J. Heringer aufgebaut wurde, um Deutschlernern mit Hilfe von fehlerhaften Sätzen die Vermeidung von häufigen Fehlern zu vermitteln (Heringer 1995). Aus diesem Korpus mit insgesamt 7107 Sätzen wurden zufällig 52 Sätze mit morphosyntaktischen Fehlern ausgewählt.

	HU/FU-Korpus	Heringer-Sätze
korrekt identifiziert	42 (53,8%)	35 (67,2%)
u.a. korrekt identifiziert	6 (7,7%)	4 (7,7%)
falsch identifiziert	12 (15,4%)	5 (9,6%)
kein Ergebnis	18 (23,0%)	8 (15,4%)
Gesamt	78	52

Tabelle 1: Evaluationsergebnisse

Die erste Zeile der Tabelle gibt die Anzahl der korrekt identifizierten Fehlertypen an. Entweder konnte der Parser nur ein Ergebnis ermitteln oder es konnte ein einziges Ergebnis mit dem besten Fehlerwert bestimmt werden. Für die Heringer-Sätze wurde die vorhandene Annotation zur Bestimmung der „korrekten“ Fehleranalyse verwendet. Für das HU/FU-Korpus dagegen habe ich den meiner Meinung nach wahrscheinlichsten Fehlertyp festgelegt. Bei einigen Sätzen ergaben sich auf Grund der Bewertung eines Satzes mit Hilfe des Fehlerwertes mehrere Fehlertypen, die einen identischen „besten“ Fehlerwert besitzen. Diese Fälle wurden in der Tabelle 1 unter „u.a. korrekt identifiziert“ zusammengefasst und treffen nur für ca. 8 % der Sätze zu. Die dritte Zeile der Tabelle zeigt die Anzahl der Sätze, bei denen der Parser eine „falsche“ Fehleranalyse produziert hat. Das heißt nicht, dass der erwartete Fehler nicht in der Ergebnismenge lag, sondern nur, dass die bevorzugte Analyse mit dem geringsten Fehlerwert einen unplausiblen Fehler beinhaltete. Schließlich gab es einige Sätze, die in ihrer fehlerhaften Form nicht geparkt werden konnten.

Im Folgenden soll nun etwas allgemeiner auf die Fehler eingegangen werden, bei denen keine Identifizierung möglich ist. Als problematisch müssen die folgenden Phänomene eingestuft werden.

- Eine erste Klasse von Fehlern zeichnet sich dadurch aus, dass in den Sätzen Phrasen zwar lokal korrekt positioniert sind, aber aus globaler Sicht an eine andere Position verschoben werden müssten. Ein Beispiel für einen solchen Satz ist der folgende aus dem HU/FU-Korpus: „Ein Auto in Ampel schlägt“. Da das Parsingverfahren nur einmal von links nach rechts den Satz analysiert, kann die Fehlstellung der PP, die an das Ende des Satzes gerückt werden sollte, nicht erkannt werden. Zu dieser Gruppe zählen auch Stellungsfehler, bei denen ein Wort beziehungsweise eine Phrase nach „vorne“ verschoben werden müsste. Auch wenn der Parser das fehlplatzierte Element erkennt, kann der Fehler aus den gleichen Gründen nicht zufriedenstellend aufgelöst werden.

- Zwei weitere Fehlertypen, die die Wortfolge betreffen, sind überzählige Wörter und Ersetzungsfehler im Satz. Diese Fehler ließen sich mit der von Lyon (1974) unrestringierten Methode erkennen, indem jedes Wort auch vom Parser ignoriert bzw. durch eine andere Kategorie ersetzt werden kann. Mir ist nicht klar, wie für diesen Fall eine Einschränkung des Suchraums aussehen könnte.

- Ein letzter Fehlertyp, der angesprochen werden soll, besteht aus idiomatischen beziehungsweise kollokativen Wendungen, bei denen unklar ist, ob eine vorliegende Beschränkung zum Bereich der Syntax oder bereits zur Semantik gezählt werden muss. Diese Schwierigkeit spiegelt sich in der ausführlichen Diskussion in der Linguistik wieder, bei der versucht wird, Regularitäten zum Auftreten von Kollokationen zu entwickeln. Die Diskussion ist auch ein Hinweis darauf, warum es einerseits so schwierig für Sprachenlerner ist, sich diesen Bereich anzueignen und andererseits die Frage der Modellierung in der Grammatik völlig ungeklärt ist.

Im Gegensatz zu den überwiegend auf der Worstellung basierenden Fehlern, für die nur in beschränktem Umfang Korrekturen vorgenommen werden können, lassen sich wie gezeigt, sich widersprechende Features mit Hilfe eines sehr generellen Verfahrens in die F-Struktur integrieren und so für das Feedback verfügbar machen.

5.4 Rückmeldungen an den Lerner

Die Meldungen für Fehler in der F-Struktur bestehen aus einer Angabe der möglicherweise be-

teiligten funktionalen Argumente, der sich widersprechenden Werte und deren Attribut sowie der möglichen Korrekturen der betroffenen Lexeme, wobei, wie erwähnt, die Korrektur nur mit Hilfe des Ableitungsbaumes und den darin enthaltenen Lexemen generiert werden kann. Bei Wortfolgefehlern sind diese Informationen eingeschränkt, da in dem hier realisierten Verfahren nur eine Korrektur präsentiert wird, die entweder besagt, welche Wortart an welcher Position eingefügt oder, welche Lexeme an welche neue Position verschoben werden mussten.

6 Fazit

Die Entwicklung eines Verfahrens zur antizipationsfreien Erkennung und Erklärung von grammatischen Fehlern, eingebettet in ein tutorielles CALL-System, war das angestrebte Ziel dieser Arbeit. Drei wesentliche Aspekte wurden dabei berücksichtigt.

1. Übungstyp und Fehlermeldungen

Wie in Abschnitt 2 ausgeführt, sehen Didaktiker neben den vor allem methodisch-didaktisch begrenzten Übungstypen die völlig unzureichenden Rückmeldungen kommerzieller Programme auf Lernereingaben als einen der wesentlichen verbesserungswürdigen Bereiche an. Im Gegensatz dazu kann das hier vorgestellte System im begrenzten Umfang auf frei formulierte Sätze reagieren und stellt mit dem Übungstyp des simulierten Dialogs eine entscheidende Erweiterung für den Bereich der tutoriellen Sprachlernsysteme dar. Außerdem ist der in *PromisD* integrierte Parser in der Lage, auf der Basis der LFG insbesondere für Kongruenz- und Rektionsfehler präzise und umfangreiche Rückmeldungen zu liefern. Dabei ist die Art und Weise der Rückmeldung, soweit das möglich ist, an den kommunikativen Übungstyp angepasst.

2. Grammatiktheorie

Es wurde eine Grammatiktheorie integriert, die aus folgenden Gründen für den Einsatz in einem ICALL-System geeignet ist. Die Theorie ist formal fundiert und damit vom Grundsatz her in Kombination mit einem Parser einsetzbar. Die Theorie hat sich außerdem zur Beschreibung von zahlreichen Sprachen als geeignet erwiesen, was sie aus einem bestimmten Blickwinkel zu einer linguistisch adäquaten Theorie macht. Schließlich hat sich die LFG im Vergleich mit anderen gängigen Grammatiktheorien als relativ ähnlich zu deskriptiven Grammatiken gezeigt, die im Sprachunterricht eingesetzt werden. Damit lässt sich die Fehlerursache direkt aus der formalen Beschreibung, der F-Struktur, generieren, wobei Korrekturen nur mit Hilfe des Ableitungsbaumes generiert werden können. Für Wortfolgefehler kann in der hier vorgestellten Form nur eine Korrektur angegeben werden.

3. Antizipation und Effizienz

Zur Identifikation von Fehlern wurden zwei Verfahren entwickelt, die im Prinzip ohne die Antizipation von Fehlertypen und -positionen in den linguistischen Wissensbasen die Beschreibung für fehlerhafte Eingaben liefern können. Damit ist es möglich, eine Grammatik und ein Lexikon ohne Bezug zur Sprachlernsituation zu entwickeln und zur fehlersensitiven Analyse von freien Lernereingaben einzusetzen.

Um eine Möglichkeit zur Optimierung der Verfahren zu erhalten, wurden allerdings sowohl für Fehler in der F-Struktur als auch in der K-Struktur Beschränkungen integriert, die den Suchraum limitieren.⁸ Obwohl der Rechenaufwand zur Bestimmung einer minimalen Lösung nicht unerheblich ist, wurde gezeigt, wie eine generelle antizipationsfreie Methode zur Analyse von fehlerhaften Eingaben in Kombination mit eindeutigen Beschränkungen u.a. auf häufige Fehlertypen in ein intelligentes Sprachlernsystem integriert werden kann.

⁸Menzel (1992, S. 225) weist darauf hin, dass für die Realisierung eines praktischen Systems „entscheidend [ist], wie man eine Kopplung antizipationsfreier Techniken mit antizipationsbasierten Ansätzen so organisiert, daß eine Symbiose der spezifischen Vorteile beider Ansätze wirksam werden kann.“

Literatur

- Bresnan, Joan: *Lexical-Functional Syntax*, Blackwell, Oxford, 2001.
- Carpenter, Bob: *Skeptical and Credulous Default Unification with Applications to Templates and Inheritance*, in: Briscoe, Ted; Copestake, Ann und Paiva, Valeria de (Hgg.), *Inheritance, Defaults and the Lexicon*, 13–37, Cambridge University Press, Cambridge, 1993.
- Dalrymple, Mary; Kaplan, Ronald M.; Maxwell, John T. und Zaenen, Annie (Hgg.): *Formal Issues in Lexical-Functional Grammar*, CSLI, Stanford, 1995.
- Earley, Jay: *An efficient context-free parsing algorithm*, Communications of the ACM, 13 (1970): 94–102.
- Grüner, Margit und Hassert, Timm: *Computer im Deutschunterricht*, Langenscheidt, München, 2000.
- Heringer, Hans Jürgen: *Aus Fehlern lernen*, Augsburg 1995, CD-ROM für Win9x/NT.
- Hubbard, Philip L.: *Non-transformational Theories of Grammar: Implications for Language Teaching*, in: Odlin, Terence (Hg.), *Perspectives on Pedagogical Grammar*, 49–71, Cambridge University Press, Cambridge, 1994.
- Kaplan, Ronald M. und Bresnan, Joan: *Lexical-Functional Grammar: A Formal System for Grammatical Representation*, in: Bresnan, Joan (Hg.), *The mental representation of grammatical relations*, 173–281, MIT Press, Cambridge, MA, 1982.
- Karttunen, Lauri: *Features and Values*, in: *Proc. 10th Int. Conference on Computational Linguistics (COLING)*, 28–33, Stanford, 1984.
- Kato, Tsuneaki: *Yet another Chart-Based Technique for Parsing Ill-Formed Input*, in: *Proc. 4th Conference on Applied Natural Language Processing (ANLP)*, 107–112, Stuttgart, 1994.
- Kleppin, Karin: *Fehler und Fehlerkorrektur*, Langenscheidt, Berlin, 1998.
- Krüger-Thielmann, Karin: *Wissensbasierte Sprachlehrsysteme*, Narr, Tübingen, 1992.
- Lyon, Gordon: *Syntax-Directed Least-Error Analysis for Context-Free Languages: A Practical Approach*, Communications of the ACM, 17(1) (1974): 3–14.
- Mellish, Chris S.: *Some Chart-based Techniques for Parsing Ill-Formed Input*, in: *Proc. 27th Conference of the Association for Computational Linguistics (ACL)*, 102–109, 1989.
- Menzel, Wolfgang: *Modellbasierte Fehlerdiagnose in Sprachlehrsystemen*, Niemeyer, Tübingen, 1992.
- Naumann, Sven und Langer, Hagen: *Parsing*, Teubner, Stuttgart, 1994.
- Rösler, Dietmar und Tschirner, Erwin: *Neue Medien und Deutsch als Fremdsprache - Viele Fragen und ein Aufruf zur Diskussion*, Deutsch als Fremdsprache, 3 (2002): 144–155.
- Rüschhoff, Bernd und Wolff, Dieter: *Fremdsprachenlernen in der Wissensgesellschaft: zum Einsatz der Neuen Technologien in Schule und Unterricht*, Hueber, Ismaning, 1999.
- Schwind, Camilla: *Sensitive Parsing: Error Analysis and Explanation in an Intelligent Language Tutoring System*, in: *Proc. 12th Int. Conference on Computational Linguistics (COLING)*, 608–613, 1988.

Themen (1997): *Themen neu 1: Lehrwerk für Deutsch als Fremdsprache*, Ismaning, 2. Ausgabe 1997.

Vogel, Carl und Cooper, Robin: *Robust Chart Parsing with Mildly Inconsistent Feature Structures*, in: Schöter, Andreas und Vogel, Carl (Hgg.), *Nonclassical Feature Systems*, Band 10 von *Edinburgh Working Papers in Cognitive Science*, 197–216, Edinburgh University, 1995.