Inhaltsverzeichnis

III Morphologie, Syntax & Parsing

IV Information Retrieval & statistische Ansätze in der Linguistik

Gerhard Heyer & Christian Wolff, Universität Leipzig

Einleitung

In diesem Band sind Vorträge versammelt, die auf der 10. Jahrestagung der *Gesellschaft für Linguistische Datenverarbeitung* gehalten wurden. Sie fand vom 17.-19. März 1997 am Institut für Informatik der Universität Leipzig statt, die erste Jahrestagung der GLDV in den neuen Bundesländern. Unter dem Titel *Linguistik und neue Medien* decken die Beiträge ein breitgefächertes Spektrum ab: Neben klassischen Themen der linguistischen Datenverarbeitung wie Computerlexikographie und Parsing fanden insbesondere aktuelle Problembereiche wie Multimediatechnologie und Anwendungen im WorldWideWeb Berücksichtigung. Diese breite Themenabdeckung steht in Einklang mit der inhaltlichen Ausrichtung der GLDV, die bewußt ein Fachverband sowohl für Computerlinguistik im engeren Sinn als auch für alle Anwendungen der Sprachtechnologie sein will. Die Aufsätze sind insofern für einen breit gestreuten Leserkreis von Interesse. Sie lassen sich in die folgenden Schwerpunkte aufgliedern:

I. Electronic Publishing, Multimedia und Informationsdienstleistungen
II. Computerlexikographie & Terminologiesysteme
III. Morphologie, Syntax & Parsing
IV. Information Retrieval & statistische Ansätze in der Linguistik

Nachfolgend wollen wir einführend eine knappe Übersicht zu den einzelnen Beiträgen geben.

1 Electronic Publishing, Multimedia und Informationsdienstleistungen

Der erste Themenbereich umfaßt ein heterogenes Spektrum von Aufsätzen, die den Bezug zwischen (Computer-)Linguistik und neuen Medien – daher auch das Motto der Tagung – herstellen. Rainer KUHLEN stellt in seinem Beitrag, der auf einen eingeladenen Vortrag zurückgeht, die Organisationsformen und Mehrwertleistungen elektronischer Märkte vor; dabei zeigt er insbesondere auf, welche Rolle linguistische Technologien (z. B. *automatic abstracting*) für solche Dienstleistungen haben können.

Gerhard HEYER & Christian WOLFF erörtern, welchen Beitrag klassische Theorieansätze aus der linguistischen Pragmatik und der künstlichen Intelligenz für die Entwicklung und Strukturierung von Multimediaanwendungen leisten können und spielen eine Operationalisierung der Sprechakttheorie an einem praktischen Beispiel durch.

Mit RECALL präsentiert Tibor KISS ein System, das intelligente Unterstützung im Bereich *computer aided language learning* (*call*) gibt und mit Standard-Webtechnologien realisiert wurde. Die Stärke des Systems liegt in seinem dynamisch anwendbarem linguistischen Wissen, das anders als typische *call*-Programme nicht nur vorgefertigte starre Übungsmuster, sondern in gewissem Umfang auch frei formulierten Text analysieren und dem Lerner Hilfestellungen geben kann.

Angelika STORRER geht auf textlinguistische Probleme bei der Erstellung komplexer Hypertexte ein. Sie erarbeitet dabei eine Methodologie für die verschiedenen Varianten eines solchen Konvertierungsvorgangs. Wie textsortenspezifische Merkmale den Umwandlungsprozeß beeinflussen, zeigt sie am Beispiel von GRAMMIS, der elektronischen Fassung einer am Institut für Deutsche Sprache entwickelten Grammatik des Deutschen.

Abschließend diskutieren Friedrich WENZEL et al. den Einsatz von Multimediatechnologien im Spachlehrunterricht und stellen ein System vor, mit dem sich unter Zusammenwirken verschiedener Multimediatechnologien sprachliche Strukturen visualisieren und manipulieren lassen. Am Beispiel der Fachsprache *technisches Russisch* wird gezeigt, wie interaktive Satzbauanalyse als Multimediaanwendung realisiert werden kann.

2 Computerlexikographie & Terminologiesysteme

Der zweite Themenschwerpunkt enthält Beiträge zur Computerlexikographie und der Verwaltung terminologischer Information. Dabei läßt sich ein guter Überblick zum Entwicklungsstand elektronischer Lexika des Deutschen gewinnen, wobei der Trend zur automatischen Verarbeitung großer Textkorpora beim Aufbau von Lexika offensichtlich ist.

Gregor THURMAIR stellt ein im Rahmen des Projektes AVENTINUS entwickeltes multilinguales Lexikon vor. Er konzentriert sich dabei auf die Wiederverwendbarkeit lexikalischer Ressourcen für Anwendungen wie die maschinelle Übersetzung oder das Information Retrieval. Neben der Erläuterung der Lexikonstruktur wird auch das zugrundegelegte XML-basierte Kodierungsformat vorgestellt.

Lothar LEMNITZER geht auf das Problem der automatischen Wissensextraktion aus großen Textcorpora ein und untersucht statistische Verfahren für die Gewinnung komplexer lexikalischer Einheiten. Im Zentrum seines Beitrags steht der Vergleich von Kollkationsmaßen, die anhand von Beispielen aus einem umfangreichen Textkorpus des Deutschen evaluiert werden.

Das Projekt *Der Deutsche Wortschatz*, vorgestellt von Uwe QUASTHOFF, stellt das Unterfangen dar, aus heterogenen maschinenlesbaren Quellen ein sehr umfangreiches Vollformenlexikon des Deutschen aufzubauen und für Anwendungen wie Rechtschreibprüfung oder maschinelle Übersetzung zu nutzen. Neben dem verteilten Sammlungsprozeß beim Aufbau des Wortschatzkorpus werden Optimierung und Verwaltung des Datenbestands diskutiert.

Ähnlich wie Lothar LEMNITZER und Uwe QUASTHOFF geht auch Jürgen OESTERLE auf das Problem der (semi-)automatischen Extraktion linguistischen Wissens (SELIK) aus großen Textkorpora ein und erörtert dies am Beispiel des CISLEX, eines umfangreichen und hochstrukturierten elektronischen Lexikons. Sein Ansatz verwendet dabei eine *definite clause grammar* für Nominalphrasen, wobei verschiende Anlaysestrategien ausführlich dargelegt werden.

Die beiden letzten Beiträge dieses Themenbereichs behandeln das Problem der Terminologiearbeit. Jan LASS stellt die Terminologiedatenbank T42 vor, die Terminologen bei allen wesentlichen Arbeitsschritten im Umgang mit terminologischem Wissen unterstützt und in andere Anwendungen (Textverarbeitung, maschinelle Übersetzung) eingebunden werden kann.

Abschließend präsentiert Katja KRÜGER das System *TermLand*, das Übersetzer und Terminologen bei der Termerkennung unterstützt und im Sinne eines *translation memory* aus bereits vorliegenden Übersetzungen je geeignete Termkandidaten ermittelt. Durch seine Realisierung mit Hilfe von Standard-Webtechnologien ist das System gut für den verteilten Einsatz z. B. in einem Intranet geeignet.

3 Morphologie, Syntax & Parsing

Im dritten Themenfeld der GLDV-Tagung sind Beiträge versammelt, die sich den klassischen computerlinguistischen Arbeitsgebieten Syntax, Parsing und Morphologie widmen. Dabei reicht das Spektrum von Parsinganwendungen bis hin zu Visualisierungssystemen, die linguistische Theoriemodelle transparent machen wollen. Schwerpunkte der Arbeiten lassen sich vor allem in den Gebieten Aufbau elektronischer Lexika mit Hilfe linguistischer Textanalyse sowie robuste vollautomatische Parsingmodelle für heterogene Texte erkennen.

Markus SCHULZE stellt das System MALAGA vor, das ausgehend von dem Modell linksassoziativer Grammatiken eine linguistische Programmumgebung bzw. ein Grammatikentwicklungswerkzeug bereitstellt. Neben der prinzipiellen Arbeitsweise von MALAGA geht er auf die automatische morphologische Analyse, das Parsing von Nominalphrasen und die semantische Analyse mit Hilfe von MALAGA ein.

Petra MAIER diskutiert im Schnittbereich von Computerlexikographie und Morphologie, wie aufgrund einer Kombination von Suffixanalyse und lokalen Grammatiken eine Defaultzuweisung morpho-syntaktischer Kategorien bei der Extraktion lexikalischer Einheiten erreicht werden kann. Wie auch im Beitrag von Jürgen OESTERLE bildet das maschinenlesbare Lexikon CISLEX den Kontext ihrer Ausführungen.

Lexana, ein System, das lexikalische Analyse auf der Basis von Unifikationsgrammatiken ermöglicht, wird von Anke KÖLZER präsentiert. Ziel des Ansatzes ist die automatische Generierung von Phrasen für ein gegebenes Lexikon. Ihr Beitrag wurde auf der GLDV-Jahrestagung als beste studentische Arbeit ausgezeichnet.

Sebastian GÖSER erörtert in seinem *extended abstract* einen Parsingansatz, der aufbauend auf einer annotierten Konstituentengrammatik sowohl schnelles wie robustes Parsing ermöglicht und so auch für fehlerhafte Texte (z. B. e-mail) und Dokumente mit breiter thematischer Streuung geeignet ist. Auch Andreas MERTENS befaßt sich mit der Problematik des *robusten Parsing*. Sein Ansatz greift das Konzept der Wortagenten auf und stellt ein System vor, das im Rahmen natürlichsprachlicher Dialogsysteme geeignete Analyseinterpretationen für eingegebene Phrasen auf der Basis konkurrierender Agenten ermittelt.

Abschließend zeigt Jean-Yves LALANDE mit dem System VisualGBX, wie sich das linguistische Theoriemodell der generativen Grammatik (insb. das *Minimalist Program*) visualisieren läßt. Dabei findet eine Reinterpretation der linguistischen Theorie mit Begriffen und Modellen der Objektorientierung statt.

4 Information Retrieval & statistische Ansätze in der Linguistik

Der letzte Schwerpunkt stellt Beiträge der angewandten Sprachtechnologie, insbesondere des Information Retrieval zusammen. Bernhard SCHRÖDER stellt mit Pro-SGML ein Prolog-basiertes Retrievalsystem vor, das eine strukturierte Recherche in SGML-kodierten Dokumentenbeständen zuläßt und damit im Vergleich mit traditionellen Textanalysesytemen differenziertere Recherchestrategien ermöglicht. An Hand von Testläufen weist er die Praktikabilität seines Ansatz bei der Verarbeitung umfangreicher Korpora nach.

Die Behandlung strukturierter Dokumente mit logischem Markup greifen auch Christa WOMSER-HACKER und Walter ZETTEL auf, die mit Hilfe klassischer Bewertungsmaße aus dem Information Retrieval die Wirksamkeit einer differenzierten Gewichtung unterschiedlicher Dokumentbestandteile untersuchen. Ihre Evaluierungsergebnisse zeigen deutlich die Vorteile der Volltextindexierung und geben Hinweise auf die Auswahl von Strukturmerkmalen bei einer differenzierten gewichteten Indexierung.

Der Beitrag von Gregor BÜCHEL behandelt ein WWW-basiertes Informationssystem für eine Hochschulbibliothek. Er zeigt auf, wie für den Schlagwortkatalog einer Bibliothek semantische Netze generiert werden können. Anhand zahlreicher Beispielrecherchen wird der Einsatz semantischer Relationen bei der Recherche verdeutlicht.

Abschließend vergleicht Michael MALBURG für Anwendungen aus dem Bereich der automatischen Texterkennung unterschiedliche Ausprägungen von Hidden Markov-Modellen (HMM). Neben der Diskussion der für die Texterkennung geeigneten HMM-Algorithmen stellt er Evaluierungsergebnisse für die Erkennung von Geschäftsbriefen vor.

Rainer Kuhlen, Universität Konstanz

Organisationsformen und Mehrwertleistungen elektronischer Märkte

1 Zusammenfassende Thesen[1]

1. Elektronische Marktplätze sind die Organisationsformen elektronischer Märkte, oder anders: die institutionellen konkreten Vermittlungsformen elektronischer Märkte. Sie entwickeln sich in regionalen und globalem Maßstab.
2. Elektronische Marktplätze dienen nicht nur der Wirtschaft (*electronic shopping*, *business to business*), sondern unterstützen auch öffentliche kommunikative Prozesse, z. B. im Austausch von Bürgern und Verwaltungen.
3. Für elektronische Marktplätze ist das Konzept der Virtualisierung entscheidend. Es gilt nicht nur für ganze Märkte, sondern auch für die Informationsprodukte (virtuelle Bücher) und Organisationsformen in virtuellen Räumen.
4. Vernetzte virtuelle Räume entstehen durch zielorientierte Kooperationsformen von Organisationen, die sich zur Durchführung von temporären oder dauerhaften Aufgaben zusammengeschlossen haben, und von Personen, die ihre "Geschäfte" über Netze abwickeln.
5. Auch auf multimedialen Marktplätzen bleiben die Informationsobjekte mit starken Anteilen sprachvermittelt. Für die linguistische Datenverarbeitung eröffnet sich ein breites Betätigungsfeld, vor allem mit Blick auf Mehrsprachigkeit, Inhaltserschlie-

[1] *Anmerkung zum Text:* Der Vortrag wurde in nicht-linearer Form unter Verwendung der Multimediasoftware *ToolBook* gehalten. Es wurde hier nicht versucht, eine lineare diskursive textuelle Entsprechung zu erstellen. Die wichtigsten Aussagen sind daher im folgenden unter den Hauptthemen weitgehend nur tabellarisch und stichpunktartig wiedergegeben.

ßung, automatische Verknüpfung, Flexibilisierung von Präsentationsformen und semantische Kontrolle.

6. Die Probleme elektronischer Märkte sind weniger technischer oder methodischer Natur, sondern verlangen in vielfacher Hinsicht vertrauensbildende Maßnahmen.
7. Entscheidend für den Erfolg elektronischer Marktplätze ist das Einhalten von Qualitätsstandards und das Erreichen von Akzeptanz der Zielgruppen.

2 Typen und Merkmale elektronischer Marktplätze

2.1 Typen

- Informationsmarkt der Fachkommunikation: Produktion von Wissen in Umgebungen von Wissenschaft und Technik und Nutzer in professionellen Umgebungen
- Märkte der Geschäftskommunikation: Organisationen der Wirtschaft und Verwaltung als Anbieter und professionelle Nutzer in Organisationen der Wirtschaft und Verwaltung
- Märkte der Verwaltungskommunikation: Organisationen der Verwaltung als Anbieter und Nutzer
- Elektronische Publikumsmärkte: Organisationen von Verwaltung und Wirtschaft und individuelle, private Nutzer
- Elektronische Individualmärkte: private Individuen als Anbieter und Nutzer

2.2 Merkmale

- Elektronische Märkte sind mit Hilfe der Telemediatik (Verbindung von *Tele*kommunikation, Multi*medi*a und Inform*atik*) realisierte Marktmechanismen.
- Auf elektronischen Märkten sind die Angebote als ortslose und zeitlose Informationsobjekte für eine räumlich verteilte Käufer- und Verkäuferschaft simultan und virtuell verfügbar.
- Auf elektronischen Märkten werden Güter der Wirtschaft über deren elektronische Substitute gehandelt, aber zunehmend auch immaterielle Informationsobjekte, z. B. aus den Bereichen der Medien, Banken, Versicherung, Verwaltung, Wissenschaft etc.
- *Nachfrager* elektronischer Markplätze sollen auf ihren Endgeräten zu jeder Zeit an jedem Ort eine große Fülle von Angeboten konkurrierender Anbieter leicht aufsuchen können.
- *Anbieter* elektronischer Markplätze sollen über elektronische Marktplätze sehr viele Kunden kostengünstig, variabel und mit leicht aktualisierbaren Produktinformationen erreichen.
- Für *Handelsmittler* eröffnen sich auf elektronischen Marktplätzen neue Möglichkeiten , um Anbietern den Weg zu den Kunden zu eröffnen und Nachfragern integrierte Problemlösungen anzubieten.

- Die Funktionen elektronischer Marktplätze erstrecken sich sowohl auf *Electronic Shopping* als auch auf *Business-to-Business*-Beziehungen. Sie bestehen aus: Information, Präsentation, Kommunikation und Transaktion.
- In methodischer Hinsicht sind elektronische Marktplätze nichts anderes als offene Hypertextsysteme mit Transaktionsfunktionen.

3 Marketingpotential für die Wirtschaft

- Präsentation der Produkte
- Einwerbung neuer Kunden, Kundenbindung
- *Benefitting*, Werbung, *Infotainment*
- Verkauf/Absatz Transaktionen
- Positionierung im Wettbewerb, Marktforschung
- Synergieeffekte, aktive Beteiligung
- Aufbau von *Corporate Image*

4 Leistungen regionaler Markplätze

- Dachfunktion des regionalen Marktplatzes für alle öffentlichen und kommerziellen Dienste und Informationen
- Bereitstellung von Funktionen für *Shopping-Center* und elektronischen Abrechnungsformen (z. B. über SET, vgl. FURCHE & WRIGHTSON 1997), offen und integrierend für alle Anbieter der Wirtschaft, Verwaltung und Gesellschaft
- Bereitstellen von marktplatzüberschreitenden Such-/Navigations- und Orientierungsformen
- Betrieb eines elektronischen Kommunikationsforums (elektronische *Online*-„Zeitung“) zur Erstellung neuer elektronischer regionaler Öffentlichkeit
- Angebot gemeinsamer Datenbankformate und Organisationstemplates zur einheitlichen Darstellung im Marktplatz
- Einbettung der Städte/Kommunen in die Regionalinformation (und die der Welt); Verknüpfung der Angebote im Internet (Globalisierung)
- Dauernde Information: Geographie, Kulturgüter, Institutionen, Geschichte, ...
- Dynamische Information: Veranstaltungskalender, Amtliche Mitteilungen, Aktuelle Angebote, Anzeigen, Hotelnachweis, Verkehrsinformationen, Wahlinformationen, elektronische Abstimmungen (TED), ...
- Mehrwertleistungen, wie Mietspiegel, Recyclingbörsen, Aus- und Fortbildungsangebote, Stellenbörsen, direkter Zugriff zu Verwaltungsfunktionen (Kfz-Anmeldung, Einwohnermeldeamt, interaktive Kontakte zum Finanzamt), ...

5 Virtuelle Regionen

- Elektronische Marktplätze sind Organisationsformen elektronischer Märkte. Sie sind keine Eins-zu-Eins-Abbildung existierenden Wirtschafts- oder Verwaltungs-

geschehens, sondern erzeugen informationelle und kommunikative Mehrwerte (z. B. Interaktivität, Flexibilität, Multimedialität, Virtualisierung).

- Elektronische Märkte ermöglichen zeit- und raumunabhängige Kommunikation (asynchron und dislozierd). Institutionalisierte Kommunikation auf elektronischen Marktplätzen begünstigt reale und virtuelle Verflechtung.
- Ausprägungen von Kommunikation auf Marktplätzen sind zielorientierte Kooperations-, Interaktions- und Transaktionsformen von Personen und Organisationen jeder Art (im Prinzip im globalen Maßstab vernetzt).
- Vernetzung kann gemessen werden als Ausmaß der Verflechtung von heterogenen Akteuren in virtuellen Räumen. Die Extension von Marktplätzen kann so in einem mathematischen Modell berechnet werden. Hohe Vernetzung konstituiert elektronische Marktplätze.
- Hohe Vernetzung in virtuellen Räumen (virtuelle Räume sind Ausprägungen realisierter, aber instabiler Kommunikationsformen) korrespondiert mit hoher Vernetzung in geographischen Räumen (der Kommunen, Regionen, Länder, Staaten, übernationale Verbünde).
- In der Entwicklung elektronischer Netzwerke bilden sich regionale/lokale Kerne und Randbereiche heraus (Vermutung: sie konstituieren neue regionale Identität). Dies sind die regionalen Marktplätze. Regionalität wird zunehmend zur Trumpfkarte im globalen Internet.
- Vernetzung diffundiert an den Rändern. Randbereiche können mehreren Netzen/ Marktplätzen angehören. Marktplätze überlappen sich (globale Regionalisierung – regionalisierte Globalisierung) und sind in nationale und globale Netze eingebettet.

6 Organisations- und Finanzierungsmodelle für universelle Dienste elektronischer Märkte

- *Prinzip der informationellen Grundversorgung*: Kostenlose Nutzung bzw. geringe Pauschale für alle Informationen, die zur Bewältigung des Alltags und der Teilnahme am politischen Leben erforderlich sind; finanziert durch öffentliche Betreiber.
- *Prinzip der Subsidiarität*: Öffentliche Unterstützung, solange die Marktmechanismen noch nicht funktionieren, finanziert durch kommerzielle Betreiber.
- *Prinzip des Marktes*: Nutzung und Finanzierung nach Angebot und Nachfrage auf dem Markt; finanziert durch öffentliche Betreiber mit kommerzieller Mischfinanzierung.

7 Barrieren der Nutzung von elektronischen Mehrwertdiensten und Marktplätzen

Barrieren aus Unternehmenssicht:

- fehlendes *know how*
- unzureichende Daten über Benutzer

- Datensicherheitsprobleme
- rechtliche Unsicherheit
- hohe Netzkosten
- Strukturierungsprobleme
- Unterhaltungskosten

Barrieren aus Nutzersicht:
- Orientierungsprobleme
- zu wenig Interaktion
- nicht individuell genug
- nicht aktuell genug
- nicht wirklich Multimedia
- keine Online-Bedürfnisse

8 Qualitätskriterien elektronischer Markplätze

Mit Bezug auf Informationen:
- *Informationsgehalt*: ausgewogenes Verhältnis zwischen Information und Redundanz
- *Verläßlichkeit*: z. B. gemessen durch das Ausmaß der Referenz auf allgemein akzeptierte Quellen, redaktionelles Überprüfen
- *Aktualität*: z. B. belegt durch *Update*-Raten und durch die Formen der *Maintenance* (automatisch, manuell)
- *Regionalität*: Einschlägigkeit der Information für die Region, z. B. bei der Wetterinformation
- *Einzigartigkeit*: Ausmaß der originalen Darstellungen in dem regionalen Markt (regionales Monopol)
- *Konsistenz der Präsentation*: einheitlicher Stil, einheitliche graphische Metapher der Anwendung

Mit Blick auf Funktionalität
- *Benutzerausrichtung* (*customizing*): Möglichkeit, Darstellungen auf bestimmte Nutzergruppen zurechtzuschneiden (*tailorability*)
- *Interaktionsfunktionen*: Unterstützung aller Formen der Geschäftstransaktion (Ordering, Bezahlen) und Manipulierbarkeit
- *Zusätzliche Leistungen*: für erweiterte Mehrwertfunktionen wie Bestellwesen, Statistik, Durchreichen zu anderen Anbietern, Elektronische Zahlungsformen, Mietspiegel, Verwaltungsvereinfachung durch Interaktivität, elektronische Kommunikationsforen, ...
- *Wissensrepräsentation*: Intensität der inhaltlichen Erschließung als Grundlage für *Retrieval* und *Filtering*
- *Metainformationen:* Bereitstellen übergreifender hypertextgerechter Such- und Navigationsfunktionen; nachvollziehbare, einheitliche Orientierungsformen

Mit Blick auf Vernetzung, Kommunikation:
- *Ausmaß der Offenheit*: Anbindung an andere regionale, nationale und globale Märkte bzw. *Online*-Dienste
- *Verknüpfungsdichte*: Ausmaß der Verknüpfung der Objekte innerhalb des elektronischen Marktes
- *Qualität der Verknüpfung*: Verwendung von typisierten (semantisch kontrollierten) Verknüpfungen, Garantie der Zielinformation
- *Referenzierbarkeit*: Aufnahme in die bestehenden Markt-*Directories*, Search Engines und andere (elektronische und konventionelle) Nachweisformen
- *Ausmaß der Interaktivität*: Kommentare, Annotationen, Kommunikationsforen, Abstimmungen, ...

Mit Blick auf soziale Aspekte:
- *Datensicherheit*: Lese-/Schreibrecht, Bestandssicherung
- *Datenschutz*: Erweiterung personenbezogener Daten auf Geschäftsprozesse, auch Fragen des *Copyright*
- *Vertrauensbildende Maßnahmen*: Angebote der Zertifizierung, Identifizierung/Authentifizierung, Weitergabepolitik; *Rating*-/Filter-Verfahren; *Ombudsman*

Mit Blick auf technische Aspekte:
- *Technische Reife*: Ausnutzung der jeweils erreichten Standards (z. B. HTML-, VRML-Versionen), *CGI*-Skripts, *Java*-Anwendungen; Datenbankabsicherung
- *Effizienz der Grafik*: ausgewogenes Verhältnis zwischen graphischer Machbarkeit und Geschwindigkeit der Darstellung des *Browsers*
- *Mediale Innovation*: Verwendung von Animation, Videoclips, ... medialen Verarbeitungstechniken
- *Netzwerkleistung*: Übertragungsleistung, Verfügbarkeit, Zuverlässigkeit
- *Aufbaukosten*: für die jeweiligen Server (einmalige und laufende Kosten)
- *Speicherkosten*: nach Anfall, inkrementell, verborgen
- *Zugriffskosten*: laufende Zugriffskosten für Anbieter und Nachfrager

9 Akzeptanz elektronischer Marktplätze

9.1 Effizienz der Innovation elektronischer Marktplätze für Organisationen

Nachteile mit Blick auf Effizienz:
- zu teuer
- nicht kalkulierbar
- elektronische Präsentation nicht umsatzfördernd
- technischer und monetärer Aufwand zu hoch
- Sorge vor Mehrarbeit

Vorteile mit Blick auf Effizienz:
- schneller, besser
- aktuellere Vorabinformation

- keine Medienbrüche
- gute Werbeträger
- Know how beherrschbar

9.2 Effektivität mit Blick auf Absatz auf elektronischen Marktplätzen

Nachteile mit Blick auf Effektivität:
- keine Kundenkreiserweiterung
- Nachfragemarkt klein
- regionaler Markt kein Vorteil

Vorteile mit Blick auf Effektivität:
- lokale, regionale und weltweite Präsenz und damit Kundenkreiserweiterung
- aktuelle Werbung
- Senkung von Kaufschwellen und Steigerung von Umsatz

9.3 Image der entstehenden elektronischen Marktplätze für Organisationen

Nachteile mit Blick auf Image:
- Mißtrauen gegenüber Technikeinsatz im Handel
- Abbau persönlicher Kontakte
- Aufbau von Scheinwelten
- Akademikerideen
- modisch, unseriös

Vorteile mit Blick auf Image:
- Pioniergeist
- Prestigegewinn
- Teilhabe an zukunftsweisender Technologie
- Image Globalisierung/Internationalisierung

10 Probleme elektronischer Marktplätze

- Dekontextualisierung: Das Hauptproblem virtueller Produkte und Dienstleistungen elektronischer Märkte besteht in der Dekontextualisierung von Wissen. Wissen bzw. daraus erarbeitete (problemangemessene) Information ist aber in hohem Maße von den situativen kontextuellen Rahmenbedingungen abhängig.
- Systeme zur Produktion virtueller Güter auf elektronischen Märkten müssen über entsprechende pragmatische Komponenten zur Rekonstruktion dieser Rahmenbedingungen verfügen.
- Qualität elektronischer Produkte: In elektronischen Informationsmärkten gehen die für unsere abendländische Kultur grundlegende Unterscheidung zwischen episteme und doxa verloren. Information steht orthogonal zur Wissensskala, sagt also nichts über den Wahrheitsgehalt der zugrundeliegenden Daten aus. Informationen sind

jeweils irgendwo in der Wissensskala zwischen „Wahrheit“ und „Lüge“ angesiedelt.

- Wer garantiert die Qualität und die Kohärenz von elektronischen, oft virtuellen Produkten, nicht nur der einzelnen Teile, sondern auch der Gesamtheit ihrer neuen Zusammenstellung ?
- Veränderungen in Autorenbegriff: Virtuelle Informationsprodukte lösen den klassischen, Qualität gewährleistenden oder Qualität einklagbaren Autorenbegriff auf. Individuelle Referenzierbarkeit ist bislang jedoch als Basis für das Vertrauen zu publiziertem Wissen unbestritten wichtig. Wer ist der Autor von elektronischen, virtuellen Produkten? Oder werden wir uns – auch im Kontext fortschreitender Gruppenarbeit (CSCW) – von dem individuellen Autorenbegriff verabschieden können oder müssen?
- Informationelle Selbstbestimmung: Mit fortschreitender Kommerzialisierung von Wissen wird der Zugriff auf Information vermutlich auch für die Wissenschaft immer weniger gebührenfrei sein. Der öffentliche, demokratische Zugriff auf das weltweit produzierte Wissen muß gesichert bleiben. Der vom Bundesverfassungsgericht vorgeschlagene Begriff der informationellen Selbstbestimmung, bislang gemeint als das Recht, über die die eigene Person betreffenden Daten selber verfügen zu können, sollte durch diese Komponente erweitert werden.
- Wie kann Vertrauen in Leistungen elektronischer Marktplätze etabliert werden?

11 Linguistische Probleme

- Die automatische Verknüpfung von Hypertexteinheiten bzw. Web Pages, auch über semantisch kontrollierte getypte Verknüpfungen stellt in großen Hypertextbasen und offenen Systemen, wie sie im WWW üblich sind, wegen des Aufwandes und der Pflege ein gravierendes Editierproblem dar. Die bislang zur Anwendung kommenden assoziativen (referentiellen) Verknüpfungen sind für Navigateure in offenen Hypertexten eine große Quelle der Unsicherheit. Verfahren sind erforderlich, durch die zum einen (vermutlich domänenspezifische) Verknüpfungstypen definiert werden, semantisch kontrollierbar und in der Erstellung automatisierbar gemacht werden können. Getypte Verknüpfungen sind dann ein wesentliche Mittel einer kontrollierten Navigation in komplexen Netzen.
- Das automatisches Indexieren als klassische Anwendungsform der Informationslinguistik im Kontext des Information Retrieval erlebt eine Renaissance durch die Entwicklung elektronischer Mehrwertleistungen und Marktplätze, vor allem deshalb, weil angesichts der immer größer werdenden Menge von Web Sites, sowohl in einzelnen Marktplätzen als auch auf dem offenen Markt insgesamt, Such- und Orientierungsverfahren zur Unterstützung der Selektionsprozesse immer wichtiger werden. Die bislang weitgehend zum Einsatz kommenden Verfahren orientieren sich weitgehend an der einfachen Volltextinvertierung oder an dem durch SALTON et al. entwickelten mathematisch-statistischen Paradigma des *Information Retrieval* bzw. des *Indexing*. Höhere Leistungen sind durch eine Kombination von visualisie-

renden, statistischen, linguistischen und Wissen repräsentierenden Verfahren zu erzielen. Aus linguistischer Sicht sollten neben den klassischen morphologischen Erkennungsverfahren und dem partiellem *Parsing*, weitgehend von Nominalstrukturen, auch diskursive und argumentative Strukturen zunehmend Berücksichtigung finden.

- *Text Summarizing* (sinnvollerweise in Form von strukturierten Abstracts) als Bestandteil einer Übergangsrhetorik; Passagenbrowsing, flexibilisierte kaskadierte Textpräsentation in unterschiedlicher Informationsdichte und Medialität
- Textmakrostrukturen für getypte Hypertextobjekte (zur Strukturierung virtueller Informationsprodukte); Rhetorik für Pfade (*guided tours*); Autorenwerkzeuge; Dossiers (als Realisierung eines virtuellen Buches)
- Multilinguale Zugriffsformen, Flexibilisierung durch individualisierte Sprachreaktion; Ausfiltern sprachfremder oder unerwünschter Dokumente
- Pragmatische Benutzermodellierung (zur Erstellung individualisierter Informationsprodukte) situative Kommunikationsanalyse; Sprecher-, Hörermodelle; sprachgesteuerte Navigation in dreidimensionalen Räumen
- Automatisches, im multimodalen Nutzermodus immer wichtiger werdendes Erkennen gesprochener Eingabe; Identifizierung prosodischer Information (Geschwindigkeit, relative, absolute Tonhöhe, Interaktionsrhythmus); variable multimodale Sprechsynthese Rhythmus

12 Forschungsprobleme auf elektronischen Markplätzen aus informationslinguistischer Sicht

- Vielsprachige Anwendungen: individualisierte Reaktionen bei Anfragen aus unterschiedlichen Kultur-/Sprachbereichen
- Hypertextgerechte Such- und Präsentationsverfahren zur Navigation in komplexen Räumen (anstelle linearer Listen); Entwicklung intelligenter Agenten, Filter
- Entwicklung von Rhetoriken, Navigations- und Orientierungsformen dreidimensionalen und multimedialen Räumen
- Entwicklung variabler/kaskadierter Informationsprodukte (virtuelle Zeitungen, elektronische Dossiers)
- benutzergesteuerte interaktive Produktionsverfahren (virtuelle Produkte)
- Assoziative (statistische und regelbasierte) Vergleichs- und Anreicherungsverfahren zur individualisierten Benutzerführung
- Wissensrepräsentation und Suchverfahren in multimedialem Material (Ton, bewegte Bilder)

Gerhard Heyer & Christian Wolff, Universität Leipzig

Zur Relevanz linguistischer Pragmatik bei der Entwicklung von Multimediaanwendungen

1 Einleitung

In der derzeitigen Diskussion um die Entwicklung von Multimediaanwendungen kann man beobachten, daß auf Kosten der Auseinandersetzung mit der inhaltlichen Strukturierung eine deutliche *Technologiezentriertheit* vorherrscht. Dies zeigt sich etwa in Überblickspublikationen zum Thema Elektronisches Publizieren (SANDKUHL & KINDT 1996) bzw. Multimedia (STEINMETZ 1994): Operationalisierbare Methodologien sind erst in Ansätzen erkennbar (vgl. DEGEN 1996).

Im Bereich der konzeptionellen Umsetzung von Multimediaprojekten bestehen deutliche Defizite, soweit man über den rein technischen Aspekt hinausgeht (z. B. zulässige Medientypen, Interaktionsformen etc.) – auch an nicht-technischen Aspekten von Hyper- und Multimediasystemen orientierte Arbeiten sind zum Thema Methodologie wenig aussagekräftig (vgl. etwa SCHULMEISTER 1996) oder beziehen auf vornehmlich textzentrierte elektronische Publikationen (vgl. DILLON 1994, LANDOW & DELANEY 1993).

Ziel unseres Ansatzes ist es, einen Weg zur Behebung dieses Defizits aufzuzeigen. Dabei greifen wir einerseits auf die linguistische Pragmatik zurück, insbesondere die Spechakttheorie, wie sie von AUSTIN & SEARLE entwickelt worden ist, andererseits verwenden wir klassische KI-Konzepte wie SCHANKS Ansatz der Strukturierung komplexer Situationen und Handlungsabläufe mit Hilfe von *Scripts* und *memory organization packages* (MOPs). Ihre Anwendbarkeit im Kontext einer Multimedia-Entwicklungsmethodologie wollen wir exemplarisch zu untersuchen.

2 Entwicklungsebenen für Multimediasysteme

Das allgemeine Gliederungsschema der Semiotik in Syntax, Semantik und Pragmatik bildet sowohl auf der Mikroebene – Modellierung einzelner Benutzeraktionen (vgl. HERCZEG 1994:10ff) – als auch auf der Makroebene – holistische Modellierung der

Gestaltung von Informationssystemen[1] – eine allgemein akzeptierte Grundlage der Modellierung der Mensch-Maschine-Schnittstelle. Im folgenden wollen wir klären, welchen Beitrag Erkenntnisse der linguistischen Pragmatik für die Gestaltung von Multimediasystemen zu leisten vermögen. Wir unterscheiden vier Entwicklungsebenen der Gestaltung von Multimediasystemen:

1. Ebene der *Kommunikationsziele*, d. h. der vom Gestalter/Designer angestrebten Vermittlung von Inhalten
2. Ebene der *Erwartungsstrukturen* der potentiellen Benutzer an ein Multimediasystem
3. Ebene von *Gestaltung* und *Interaktion* (Mediendesign und Interaktionsdesign nach DEGEN 1996), repräsentiert durch den Gestaltungsentwurf und dessen Konkretisierung durch einzelne Gestaltungselemente (durch je ein bestimmtes Medium repräsentierte Informationseinheiten[2]) sowie durch die Gestaltung des Interaktionsablaufs zwischen Benutzer und System
4. die Ebene der tatsächlichen *technischen Umsetzung*, z. B. mit einem Autorensystem wie z. B. *Asymetrix ToolBook*.

Von der untersten Ebene, d. h. der technischen Realisierung kann in diesem Kontext abstrahiert werden, da sie zwar als einschränkende Randbedingung auf die höheren Ebenen zurückwirken kann (z. B. bei fehlender Systemfunktionalität für einen bestimmten Typus von Animation), sich im Allgemeinen aber vom „eigentlichen" Modellierungsproblem ablösen läßt.

Für die Ebene von *Gestaltung* und *Interaktion* existieren methodische Vorgaben, die unmittelbar zur Operationalisierung führen bzw. bei der Umsetzung eines Kommunikationsziels beachtet werden müssen: Die Auswahl einzelner Gestaltungselemente kann wenigstens partiell durch konkrete Stylevorgaben der Gestaltungsplattform (z. B. ein *GUI style guide*, vgl. EBERLEH 1994, bes. 165ff) analysiert bzw. synthetisiert werden; zumindest lokale Interaktionsabläufe können auf der Basis softwareergonomischer Normen bzw. Evaluierungsverfahren (z. B. GOMS, vgl. HERCZEG 1994:41ff, 219ff.) gestaltet werden.

Von entscheidender Bedeutung ist jedoch die Modellierung der *Kommunikationsziele* (Perspektive des Gestalters), ihre konkrete *multimediale Umsetzung* auf der Gestaltungsebene und ihre Abgleichung mit den *Erwartungshaltungen* des Benutzers. Zur Analyse, unter welchen Bedingungen Kommunikationshandlungen „glücken" bzw. „scheitern" – in unserem Fall also das Glücken oder Scheitern einer über das Medium einer multimedialen Applikation vermittelten Kommunikation – bietet es sich an, Kommunikationsmodelle aus der linguistischen Pragmatik heranzuziehen, wobei von

---

[1] Vgl. STEINMÜLLER 1994:202ff zur Rolle der Semiotik unter besonderer Berücksichtigung einer *sigmatischen* Ebene als Bezug von Information (ausgedrückt durch ein Zeichensystem) zur Ebene real existierender Dinge.

[2] Zu Definition und Begriff der Informationseinheit bzw. der informationellen Einheit vgl. KUHLEN 1991:79ff.

vornherein davon auszugehen ist, daß aufgrund der Eigenheiten der Mensch-Maschine-Interaktion an eine einfache 1:1-Übertragung nicht zu denken ist.[3]

3 Ein Anwendungsbeispiel: Stadtrundgang Leipzig

Als Anwendungsbeispiel für unsere Überlegungen greifen wir auf ein Multimediaprojekt zurück, das seit dem Sommersemester 1996 mehrfach am Institut für Informatik der Universität Leipzig im Rahmen eines einsemestrigen Praktikums *Elektronisches Publizieren* durchgeführt wurde. Dabei entstand eine multimediale CD-ROM mit Informationen über das Studium der Informatik in Leipzig im besonderen sowie Zusatzmaterialien über die Universität, die Stadt etc. (vgl. QUASTHOFF & WOLFF 1997). Die Arbeiten an dieser CD-ROM entwickelten sich zunächst entsprechend der oben skizzierten Technologiezentriertheit. Erst im Laufe des Projekts wuchs das Problembewußtsein für eine systematische Entwicklungsmethodologie. Die im folgenden dargestellten Überlegungen stellen in diesem Sinne also eine Reflexion auf die Erfahrungen aus diesem Projekt dar. Derzeit wird eine ähnliche Anwendung für die Universität Leipzig insgesamt umgesetzt, so daß wir Gelegenheit, die Angemessenheit, Praktikabilität und Nützlichkeit unseres Ansatzes als *Entwicklungsheuristik* zu überprüfen.

Als konkretes Beispiel soll ein multimedialer „Rundgang“ durch die Stadt Leipzig dienen, wie er in Abb. 1a und 1b gezeigt wird. Mit dem elektronischen Stadtplan wird versucht, die räumliche Metapher des Rundgangs auf das elektronische Medium zu übertragen: Der Benutzer kann sich von Ort zu Ort „fortbewegen“ indem er die dargestellten Highlights über einen schematischen Stadtplan direkt-manipulativ auswählen und sich anzeigen lassen kann. Im Unterschied zu einem tatsächlichen Rundgang kann er jedoch von einem Ort zum anderen „springen“ und ist nicht an eine bestimmte (wenn auch nur Kräfte sparende) Reihenfolge der Besichtigung gebunden.

Abb. 1a/b: Ausschnitt aus einem multimedialen Stadtrundgang

[3] Immerhin wird man mit SCHULMEISTER 1995:34f festhalten können, daß in der *Interaktion* zwischen Mensch und Computer die kommunikativen Absichten des Systemdesigners vermittelt werden: „Der Computer ist dadurch nicht mehr bloßes Werkzeug, sondern wird zu einem ‘kulturell situierten Objekt’“ (SCHULMEISTER 1996:35).

4 Korrelation mit Modellbildungen der linguistischen Pragmatik

4.1 Klassische Sprechakttheorie: AUSTIN und SEARLE

AUSTIN 1955, dt. 1979 führt als erster den Begriff des Sprechaktes ein und unterscheidet folgende Aspekte: Wer spricht, vollzieht einen *lokutionären* Akt, indem er sich auf ein Objekt bezieht (*reference*) und über dieses etwas aussagt (*sense*). Wer spricht, vollzieht aber auch einen *illokutionären* Akt, d. h. er tut etwas, indem er etwas sagt (z. B. zu etwas rät oder etwas empfiehlt). Wer spricht, vollzieht schließlich aber auch einen *perlokutionären* Akt, d. h. er tut etwas dadurch, daß er etwas sagt (z. B. daß jemand etwas tut). Die Unterscheidung der Akte geschieht unter dem Gesichtspunkt der charakteristischen Fehler, die man beim Sprechen machen kann: Jemand kann gegen die Regeln der deutschen Grammatik verstoßen (und damit einen Fehler auf lokutionärer Ebene machen); jemand kann eine bestimmte Konvention fehlanwenden (und damit einen Fehler auf illokutionärer Ebene machen); oder jemand kann je nach der Situation, in der er sich befindet, die Wirkung seiner Äußerung fehleinschätzen oder willentlich mißbrauchen (und damit einen Fehler auf perlokutionärer Ebene machen). Übertragen wir AUSTINs Unterscheidung auf unsere oben skizzierten Entwicklungsebenen der Gestaltung von Multimediasystemen, so läßt sich feststellen, daß die Erwartungen eines Benutzers ganz analog auf den Ebenen der Gestaltung (lokutionäre Ebene), der Interaktion (illokutionäre Ebene) und der Kommunikationsziele (perlokutionäre Ebene) *nicht* erfüllt werden können. Der AUSTINsche Ansatz ist für unsere Zwecke jedoch zu wenig differenziert und kaum hilfreich bei der Festlegung von Kriterien für ein „gutes“, d. h. die Benutzererwartungen optimal auf die Kommunikationsziele einstellendes Design einer Multimediaapplikation.

Anders als AUSTIN konzentriert SEARLE seine Aufmerksamkeit nicht auf den in einen institutionellen Rahmen eingefügten Sprechaktvollzug, sondern analysiert die Bedeutung explizit performativer Sätze, d. h. solcher Sätze, deren aufrichtige Äußerung den Vollzug eines bestimmten Sprechaktes darstellt. Von der Voraussetzung ausgehend, daß Sprechen eine Form regelgeleiteten Verhaltens sei, fragt SEARLE danach, welche semantischen Regeln für den Vollzug etwa eines Versprechens bzw. für den Gebrauch von „Ich verspreche, daß ...“ konstitutiv sind. In der Beantwortung dieser Frage ergibt sich ein Regelsystem: Zum Versprechen gehört als *Regel des propositionalen Gehalts* das Prädizieren eines eigenen zukünftigen Aktes, als *wesentliche Regel* die Übernahme der Verpflichtung zur Ausführung dieses Akts, usf. Die Regeln des Versprechens, Aufforderns usw. sind nach SEARLE universal: Wo immer es so etwas wie das Versprechen gibt, gehorcht es den Regeln, die SEARLE herauszuarbeiten versucht hat.

Dieses klassische Schema der Illokutionstypen eines Sprechaktes, wie es bei SEARLE 1983:100ff ausführlich beschrieben ist, läßt sich auf die Modellierung von Kommunikationszielen im Kontext eines Multimediasystems übertragen. Die Schematisierung unterschiedlicher Typen von Kommunikationsakten findet eine Analogie in den Kommunikationszielen eines Multimediasystems. Das zentrale Problem der

Gruppenbildung, das schon in der Sprechakttheorie nicht abschließend gelöst werden konnte bleibt zwar grundsätzlich erhalten, läßt sich aber bei Einführung einer nicht abgeschlossenen Menge unterschiedlicher Kommunikationsziele eines Multimediasystems „pragmatisch umgehen." Exemplarisch soll Tabelle 1 eine Modellierung von Kommunikationszielen in der Begrifflichkeit der Sprechakttheorie am Beispiel „Werben" verdeutlichen:[4]

Propositionaler Gehalt	***eine zukünftige Handlung A von H***
Voraussetzungsbedingung	1. Es ist nicht offenkundig, daß H im weiteren Verlauf der Dinge A tun wird 2. S hat Gründe A zu empfehlen 3. S darf annehmen, daß H seine Gründe A zu empfehlen nicht kennt
Aufrichtigkeitsbedingung	S ist davon überzeugt, daß A gut ist
Wesentliche Bedingung	zählt als Versuch, H zu A zu bewegen

Tabelle 1: Sprechakt/Kommunikationsziel „Werben" (H = Hörer/Nutzer, S = Sprecher/System)

Im Unterschied zur klassischen Theorie der Sprechakte ist bei der Anwendung auf Formen multimedialer Kommunikation allerdings die Erweiterung wesentlich, daß die Einleitungsregeln des Sprechakts (allgemeiner: des kommunikativen Aktes in seiner jeweiligen Repräsentation in einem Multimediasystem) mit den Erwartungshaltungen des Benutzers zu korrelieren sind.

4.2 Modellierung von Erwartungshaltungen: Scripts und MOPs

Ein geeigneter Ansatz zur Modellierung von *Erwartungshaltungen* auf der Ebene des Benutzers oder Anwenders eines Multimediasystems, und damit ihrer Korrelation mit den Kommunikationszielen des Gestalters, findet sich in dem Begriff von *memory organization packets* (MOPs) von SCHANK in Erweiterung seines seit langem als klassisch geltendem Script-Ansatzes (vgl. BARR & FEIGENBAUM 1981:216ff, 307ff; SCHANK 1987:179ff). SCHANK unterscheidet die folgenden unterschiedlichen Gedächtnisebenen (Tab. 2):

Erfahrungen	***Erwartungen***
EM (*event memory*) spezielle Erfahrungen	spezielle Situation
GEM (*generalized event memory*) Verallgemeinerung spezieller Erfahrungen	Verallgemeinerung spezieller Situationen
SM (*situational memory*) Bestätigung spezieller Situationserwartungen	allgemeiner Kontext für spezielle Situationen
IM (*intentional memory*) Bestätigung allgemeiner Problemlösungserwartung	Regeln zur Problemlösung

Tabelle 2: Korrelation Gedächtnisebenen und Erwartungshaltungen nach Schank

[4] An einer Typologie unterschiedlicher Kommunikationsziele, die auch eine hierarchische Strukturierung bzw. Einbettung zuläßt, arbeiten wir derzeit noch.

Jede Erfahrung läßt sich schematisch einer Gedächtnisebene zuordnen, wobei jede Erfahrung wiederum eine entsprechende Erwartung bedingt. Einzelne Erfahrungen verschiedener Gedächtnisebene werden unter einem *memory organization package* (MOP) gebündelt. Diese MOPs können bei der Verarbeitung neuer Informationen aufgerufen und aufgrund der korrelierten Erwartungen von Erfahrungen für eine weitergehende erwartungsgesteuerte Verarbeitung genutzt werden, wie SCHANK am Beispiel mehrerer natürlichsprachlicher Systeme gezeigt hat.

Für unsere Zwecke bietet sich das Konzept der MOPs vor allem zur Modellierung von ereignisgesteuerten Erwartungen auf seiten des Benutzers an, wie sie bei einer multimedialen Applikation anzutreffen sind. Tabelle 3 zeigt die Korrelation zwischen dem gegebenen Informationsproblem und den Erwartungsstrukturen des Benutzers nach dem Schema von SCHANK 1987 am Beispiel unseres Stadtrundgangs Leipzig:

Gedächtnisebene	***Hörer/Nutzer***	***Sprecher/System***
IM	INFOPROBLEM (Finden) INFOS SUCHEN INFOS AUFRUFEN	INFOPROBLEM (Präsentieren) INFOS ANBIETEN INFOS HERVORHEBEN
SM	STADTPLAN SUCHEN HIGHLIGHTS SUCHEN HIGHLIGHTS UND ZUGEHÖRIGE INFOS AUFRUFEN	STADTPLAN UND HIGHLIGHTS PRÄSENTIEREN HIGHLIGHTS UND ZUGEHÖRIGE INFOS PRÄSENTIEREN
GEM	Stadtplan anschauen Highlights lokalisieren Ausgangspunkt lokalisieren, orientieren Stadtrundgang planen Informationen zu Highlights bereithalten und entsprechend abrufen räumlich-historische Querbezüge herstellen etc.	Stadtplan zeigen Highlights lokalisieren Ausgangspunkt lokalisieren, orientieren Stadtrundgang anbieten Informationen zu Highlights bereithalten und ggf. präsentieren räumlich-historische Querbezüge herstellen etc.
EM	Highlight nicht gefunden verlaufen falsche Information gehabt etc.	Wirkung Mausbewegung, linker und rechter Mausklick virtuelle Bewegung scrolling Präsentation Highlights Präsentation Informationen etc.

Tabelle 3: Erfahrungs- und Erwartungsstrukturen für Leipzig Info

5 Fazit und Ausblick

Als Hilfsmittel für die Modellierung von Multimediasnwendungen wurden die klassische Theorien der linguistischen Pragmatik (AUSTIN, SEARLE) und der künstlichen Intelligenz auf ihre Eignung untersucht. Die oben eingeführten Entwicklungsebenen 1 und 2 für Multimediasysteme korrelieren wir dabei mit den folgenden Theorien der Pragmatik bzw. Ansätzen aus der KI (Tabelle 4).

Entwicklungsebene	***Theoriemodell***	***Operationalisierung***
Kommunikationsziele (Gestalter)	Sprechakttheorie (AUSTIN, SEARLE)	linguistische Pragmatik
Erwartungsstrukturen (Benutzer)	scripts und memory object packages (MOPs) (SCHANK 1987)	Erfahrungs- und Erwartungsstrukturen (MOPs)
Gestaltung und Interaktion	Medien- und Interaktionsdesign (vgl. DEGEN 1996)	Gestaltungsentwurf und Interaktionsablauf
technische Umsetzung	Software Engineering und Software Ergonomie	Autorensystem, Software Engineering Normen

Tabelle 4: Korrelation Gestaltungsebenen & Theoriewissen

Nach unserer Auffassung kann der hier vorgeschlagene Theorietransfer – bei hinreichender, noch zu leistender Präzisierung – in unterschiedlichen Richtungen fruchtbar gemacht werden:

a) Bei der Entwicklung von Analyse- und Bewertungsverfahren für Multimediasysteme bzw.
b) bei der Entwicklung bzw. Entwicklungsmethodologie selbst, wobei hier wiederum zu unterscheiden ist zwischen Leitfäden für den Entwickler und implementierbaren Entwicklungswerkzeugen.

Gleichzeitig sind jedoch auch die Grenzen der theoretischen Basis offenzulegen: Die Theorien beziehen sich auf die Kommunikation zwischen Menschen mittels gesprochener Sprache und berücksichtigen nicht die Besonderheiten der Mensch-Maschine-Interaktion (vgl. HERRMANN 1986). Sie eignen sich zunächst vor allem für die Beurteilung *einzelner* kommunikativer Akte. Wesentlich für die Gestaltung eines Multimediasystems ist aber gerade das Problem der Strukturierung größerer Informationseinheiten. Dazu bedarf es ergänzender (hyper-) textlinguistischer Verfahren (vgl. etwa STORRER 1995:301ff) für die Bearbeitung vorliegenden Materials einerseits, einer Theorie für die hierarchische Strukturierung kommunikativer Akte auf der Basis der Sprechakttheorie andererseits. Es ist darüber hinaus zu fragen, inwieweit zunächst Sprach- bzw. textbezogene theoretische Ansätze zu einer semiotischen Theorie für Multimediasysteme erweitert werden können. Ausgehend von einer Typologie unterschiedlicher kommunikativer Akte (z. B. Belehren, Informieren, Werben, Darstellen etc.) müssen verschiedene, für Multimediasysteme typische Kriterien mit in Betracht gezogen werden. Dabei handelt es sich um

- die Repräsentationsform, d. h. die Frage, durch welches Medium ein Kommunikationsakt vollzogen werden soll (Text, Sprache, Bild, Animation etc.);
- die Strukturierungsform, d. h. wie umfangreich die letztlich einzeln darstellbaren Informationseinheiten sein können und welche Gliederungsform gewählt wird (streng hierarchische Gliederung, Hypertextnetzwerk, Typologie von Verknüpfungsformen etc.); an dieser Stelle können Erkenntnisse sowohl der Textlinguistik wie auch der Rhetorik Hilfestellung geben;

- der Bezug zu unterschiedlichen Bewertungsebenen der Kommunikation: Globale Gliederung, Aufbau einzelner Sektionen, Detailgestaltung abgeschlossener Informationseinheiten.

Die Entwicklung und Gestaltung eines komplexen multimedialen Informationssystems ist ein kreativer Akt, der sich einer vollständigen Formalisierung – auch im Sinne einer Entwurfsmethodologie – entzieht, vergleichbar der Tätigkeit eines Buchautors[5]. Hinsichtlich der Frage, inwiefern dies erlernbar ist, bestehen etwa zwischen dem europäischen und dem angloamerikanischen Raum deutliche Unterschiede, wenn man z. B. die durchaus erfolgreiche angelsächsische Tradition, *creative writing* zu lehren, mit eher am originären Geniegedanken europäischen Haltungen vergleicht.[6] Die Weiterführung dieses Gedankens soll keineswegs zur Sammlung statischer Entwurfsschemata für Multimediaanwendungen führen, wie sie in meist wenig hilfreicher Weise von Autorensystemen angeboten werden; vielmehr ist an einen Modellierungsleitfaden gedacht, der eine Entwurfsheuristik zur Verfügung stellt, die

- die Projektplanung erleichtert,
- die Abbildung von Kommunikationszielen auf die Oberflächenrealisierung erleichtert und
- bei der Definition des Interaktionsdesign behilflich ist.

Der hier vorgestellte Ansatz kann nur ein erster Versuch sein, Theorien der linguistischen Pragmatik für die Entwicklung von Multimediasystemen zu operationalisieren. Es zeigt sich allerdings, daß auch in anderen Disziplinen der Problematik multimedialer Kommunikation Aufmerksamkeit geschenkt wird: Einen ähnlichen Ansatz verfolgt z. B. LAUREL 1993:125ff, wenn sie versucht, für die Interaktion zwischen Mensch und Multimediasystem Gestaltungsmaximen aus der Dramentheorie abzuleiten.

---

[5] Eine Theorie des Schreibprozesses am Computer im Unterschied zur klassischen Vorgehensweise (ein Sonder- oder Unterfall der Gestaltung eines Multimedia-Systems) existiert allenfalls in ersten Ansätzen, vgl. KAPLAN & MOULTHROP 1993.

[6] Als Ausnahme könnte man das *Deutsche Literaurinstitut Leipzig* ansehen, das der Universität Leipzig angegliedert ist und an dem eine "Ausbildung zum Schriftsteller" möglich ist.

Tibor Kiss, IBM Global Services, Institute for Logic and Linguistics, Heidelberg

RECALL – Demonstrating a System Architecture for Repairing Errors in Computer Aided Language Learning

1 Summary

RECALL (Repairing Errors in Computer Aided Language Learning) was a preparatory study carried out to identify future tendencies of intelligent programs for CALL (Computer-Assisted Language Learning). RECALL aimed at providing the learner with individualized error feedback through a user-oriented interface. The project thus involved identification of improved techniques for handling errors as well as recording user feedback on a CALL prototype. On the technical side, RECALL concentrated on the following modules:

- A Diagnosis Module that analyses the students' input by means of knowledge of the language to be learnt represented in syntax and lexicon,
- A Learner Module that stores information about the individual learner, and
- A Tutoring Module that leads the user to exercises suitable to his strengths and weaknesses and gives structured feedback to erroneous input.

RECALL developed an architecture for the interaction of tutor, learner and error recognition modules capable of handling freely written natural language input (cf. KRÜGER et al. 1996, FITZPATRICK & GRIESZL 1996, MURPHY 1997). Parts of this generic architecture were integrated into a small-scale demonstrator that was publicly available for experiments.

The following report briefly describes the functionality of the major modules of the system architecture, relates the system architecture to the functionality of the demonstrator, and describes the underlying pedagogic approach.

The research carried out within RECALL encompassed work in the social sciences, CALL, natural language processing (NLP), as well as artificial intelligence (AI). Besides describing the system components, the report also tries to clarify which further steps could be undertaken at the boundary of CALL, NLP, and AI.[1]

---

[1] The work reported herein was partly supported by the Commission of the European Communities

2 Major Modules of the System Architecture

RECALL's contribution to the system design of intelligent CALL systems revolved around the provision of three modules to be integrated into an existing CALL system, *diagnosis module, a learner module,* and *a tutoring module.* The functionality of these three modules is briefly described in the following subsections.

Diagnosis Module
The Diagnosis Module (DM) concentrated on designing a modular natural language processing approach for error diagnosis. One of the main features of the work was to construct meaningful error descriptions for a range of syntactic and morphological errors as described in KRÜGER & GEURTS 1997. These descriptions are used by the Tutoring Module for the selection of feedback and used to update the learner model.

Learner Module
The Learner Module (LM) maintained individual learner models by monitoring the learner's progression and by gathering information from the Diagnosis and Tutoring Modules. The learner model is continuously updated by a series of implicit acquisition rules working at runtime.

Tutoring Module
The Tutoring Module (TM) concentrated on the design of a hybrid tutoring approach combining communicative and grammatically based exercises. In addition, the TM provides enhanced feedback through the design of a multilevel response strategy. This strategy ensures that the learner is given appropriate help through a variety of learner-centered pathways to the problem solution. Finally, the TM contains an explicit model of the curriculum that is used as the foundation of exercise selection and maintenance of the learner model.

Demonstrator
The demonstrator incorporates a role-playing scenario/game supplemented by a series of grammatically based remedial exercises. The demonstrator incorporates a language tutor that operates at the linguistic level and a game tutor that operates at the story level.

Within the scenario, the learner is allowed to enter almost free input. This input is checked for correctness and breakdowns in competence. Based on a detailed error description from the DM, the TM is able to select appropriate feedback texts, explanations and tutoring responses. The LM maintains a statistical representation of the learner's competence in the language that is used by the DM. This representation is

as part of the TELEMATICS applications programme (LE1-1615), carried out from November 1995 until April 1997 (cf. KISS et al. 1997). I would like to acknowledge the work of the following group of people which led to RECALL's system architecture and RECALL's demonstrator: Ivo DÜNTSCH, Claire FITZPATRICK, Günter GEDIGA, Peter GERSTL, Bart GEURTS, Andrea GRIESZL, Marc HÜSKEN, Anja KRÜGER, Maureen MURPHY. It should be stressed that none of them is responsible for any error made in the present report, which is based on KISS et al. (1997).

used to select the most probable error hypothesis and by the TM to select suitable remedial exercises.

The game tutor focuses the learner on tasks that have to be accomplished to progress in the game. This provides an innovative level of interactivity with the CALL program that serves to motivate the learner and enhances the communicative aspects of a CALL application.

The Demonstrator runs as a 16-bit application from Windows™ 95. The minimal hardware requirements for running the RECALL demonstrator correspond to the requirements of the version of Netscape Navigator™ used. Since the demonstrator makes use of Netscape's frame concept the version of Navigator used must be at least 2.0.

The DM incorporates a C-based chart-parser with a feature based grammar and lexicon making use of an attribute-value-formalism of the PATR-II family. The LM and the TM are both written in PROLOG.

3 The Diagnosis Module

Matching input with anticipated correct and incorrect solutions can be best applied with multiple choice or single-word gap filling exercises. However, this procedure often cannot assess the student's errors. In more complex exercise types, or whenever the TM fails to provide an adequate reaction to the student's input, an analysis based on linguistic methods becomes necessary. The DM thus analyses the learner's input by means of a parser operating on a grammar, error grammar and lexicon. Based on an error typology and classification , the task of the DM is then to describe the error as detailed as possible.

To find an applicable system of classifying errors the work started out from a survey of common mistakes and their classification (cf. KRÜGER & GEURTS 1997). The compiled data showed errors on all linguistic levels (e. g., orthographic, morphological, syntactical). Overgeneralization and interference were identified as a cause of the error in many cases.

Error rules represent erroneous structures, e. g. due to typical syntactical interference from other languages. It was shown that interference chiefly occurs in exercises that allow freely formulated sentences. Another clear outcome was that overgeneralization can be detected most frequently on the morphological level.

A central component of the DM is a chart parser written in C using an algorithm similar to the EARLEY algorithm. The parser is capable of constructing nested feature structures.[2] The use of these structures allows a clear organization of the syntactic information attached to the constituents allowing for a greater readability of both the knowledge bases and the analysis results.

[2] RECALL has made use of a parsing module developed in work conducted by Hagen LANGER at the University of Osnabrück.

The grammar used in the RECALL project is a feature term grammar, with a formalism similar to the PATR II formalism (SHIEBER 1986). By unification of feature structures the parser builds up the syntactic structure of the input sentence.

The linguistic knowledge that is needed for the analysis of the learner's input is contained in several knowledge bases: grammar, error grammar and lexicon. The grammar employed in the demonstrator comprises some 200 rules, half of which are error rules. With the capacity to analyze several types of both *yes-no* and *wh-questions,* interrogatives are the most complex type of clauses in RECALL. Other features of the grammar include handling declarative sentences, genitive attributes, adjectives with a preposition and gerund and infinitive clauses.

4 Learner Model

The learner model was designed to include the following information:

- *Student Profile*: background information on the learner such as name; native language; initial proficiency in the target language; proficiency in other background languages; motivation for learning language level of academic qualifications and date of learner's last session.
- *Student Model*: a dynamic representation of the learner's competence in the domain based on his grasp of grammatical phenomenon and her proneness to commit errors.
- *Cognitive Model*: stable characteristics of the learner. The includes information on the preferred feedback media, the learner's interest in grammar and the learner's preference for the usage of polite form in tutoring feedback.

In addition, a number of update rules ensured that information provided by the TM and the DM were used to maintain an accurate picture of the learner's performance.

Due to the limited nature of the demonstrator phase, the LM was not fully implemented. However, key features were extracted to develop a statistically based learner model that maintains a score of the number of errors made. These errors are classified into error types. The resulting statistics are updated as the learner progresses through the role-playing scenes and remedial exercises. The demonstrator incorporates an implicit assumption that the learner is a pre-intermediate having some knowledge of the domain. Hence there is no initial stereotype instantiation.

5 Tutoring Module

A TM uses the information from the DM and the learner model to calculate an appropriate system reaction. This reaction can either be a direct feedback to the user's input or some appropriate system advice that corresponds to the learner's proficiencies. Thus the TM of an intelligent CALL system is assigned a twofold role:

- To determine when and how to intercede in the lesson and feedback generation, and
- To select and order the exercises for each learner.

The feedback generation of the TM follows a strategy that

- distinguishes between different input qualities
- adapts to the number of attempts for each question, and
- offers a multilevel error-specific help device.

The TM's prior task is the control of the feedback strategies in correspondence to the results of the diagnosis. This role has been implemented as follows:

The *tutor* determines whether the input is a new one before the user input is transmitted to the diagnosis. Any new input will be sent to the DM (cf. KRÜGER et al. 1996). The results are passed back to the tutor module which will react according to a multi-level strategy.

We distinguish eight cases about the quality of the input, depending on whether the input is grammatically and lexically correct, and whether the input includes typographic errors. These cases combined with the number of attempts needed and information that is stored in the LM will lead to different response strategies that are stored in the TM.

If the input is correct, the tutor reacts with a laudative notification of the input's correctness. If the input is correct but contains typographic errors, the typos will be acknowledged, and the user will be offered to correct the error. The treatment of grammatically incorrect user responses depends on whether the input contains a typographic error as well. Typographic errors that have occurred additionally are regarded as less important and will not be acknowledged until the input is grammatically correct. Lexical errors are treated in similar fashion. A lexical error is considered more important than a typographic error.

The same hierarchical strategy applies if the input is completely corrupted. In this case, the correction of a grammatical error is considered more prominent than the correction of a lexical error. A typographic error occupies the lowest position in this hierarchy. We thus decided to consider each error, independent of its importance with respect to the current learning goal. The errors, however, are displayed in a structured manner.

6 Directions for Future Work

6.1 Diagnosis Module

With NLP methods it is possible to provide detailed error feedback for a range of error types in an exercise allowing the input of freely formulated sentences. The range of error types in RECALL was selected according to a hierarchy of most frequent errors developed in KRÜGER & GEURTS 1997. In addition, we focused on errors that were to be expected in the exercise scenarios. Morphological and syntactical errors are most prominent. What is more, they are relatively simple to treat computationally.

Using the same technology for a complete language course would most probably not be feasible, however. If more errors have to be taken into account, the increase in

complexity may lead to contradicting analyses of the input. In RECALL, this problem was overcome by restricting the language fragment and by modularizing the system grammar and the error grammar. Moreover, the interaction of the DM with the learner model allowed a significant restriction of hypotheses, so that most input sentences could be analyzed.

If NLP is to be employed into CALL, even more restricted scenarios are required. Gap texts offer this sort of restriction. Here, the type of sentence the student is expected to enter can be predicted easily. Thus the application of parsing using a modularized grammar and error grammar would be very practical in these exercises.

Free input seems to demand a general parsing and diagnosis procedure to be able to deal with the input. Employing NLP techniques allows at least for a broad and exercise-independent coverage of error-types. One could still investigate whether such results can also be obtained by simpler, and computationally more feasible matching procedures.

6.2 Learner Module

The LM contains a full range of information that can be used to approximate the learner's competency in the domain and her preference when using an intelligent CALL system. The update rules are computationally feasible to ensure that a real-time learner modeling component can be incorporated into a commercial application. It is a simple task to maintain the learner model information as the design employs a modular structure.

It may be difficult, however, to formulate an explicit Domain Model if a large language fragment is chosen for the CALL application. Problems arise due to the need to specify the ordering of the language, i. e. finding the topic pre-requisites and the order in which topics are taught.

6.3 Tutoring Module

Limitations for the TM only arise from time resources and from the input it receives, in particular from the results delivered by the DM. Given sufficient time, it is feasible to create a representative set of exercises that may serve as a considerable amount of material for a hierarchical Domain Model.

The second important task of the TM is exercise selection and sequencing. It has hardly been implemented in the Demonstrator, but can be accomplished almost independently from the performances of the DM.

A welcome extension would be the retrieval of information concerning the user's proficiencies. Such information can be gathered by evaluating *positive evidence.* Such information particularly encompasses topic-specific exercises which are not only as remedial material, but also as main-exercises. If the user has to solve these to progress, every correct answer can be recorded as a positive evidence for a better proficiency in the corresponding topic.

7 Pedagogic Approach of the Tutoring Module

The RECALL Demonstrator is a story-based pre-intermediate English (mini)course, which trains the student to build grammatically correct sentences in given situational contexts. The user is allowed to enter almost free input in these exercises, apart from one restriction: the user has to choose the lexemes from a wordlist, as is illustrated in Figure 1.

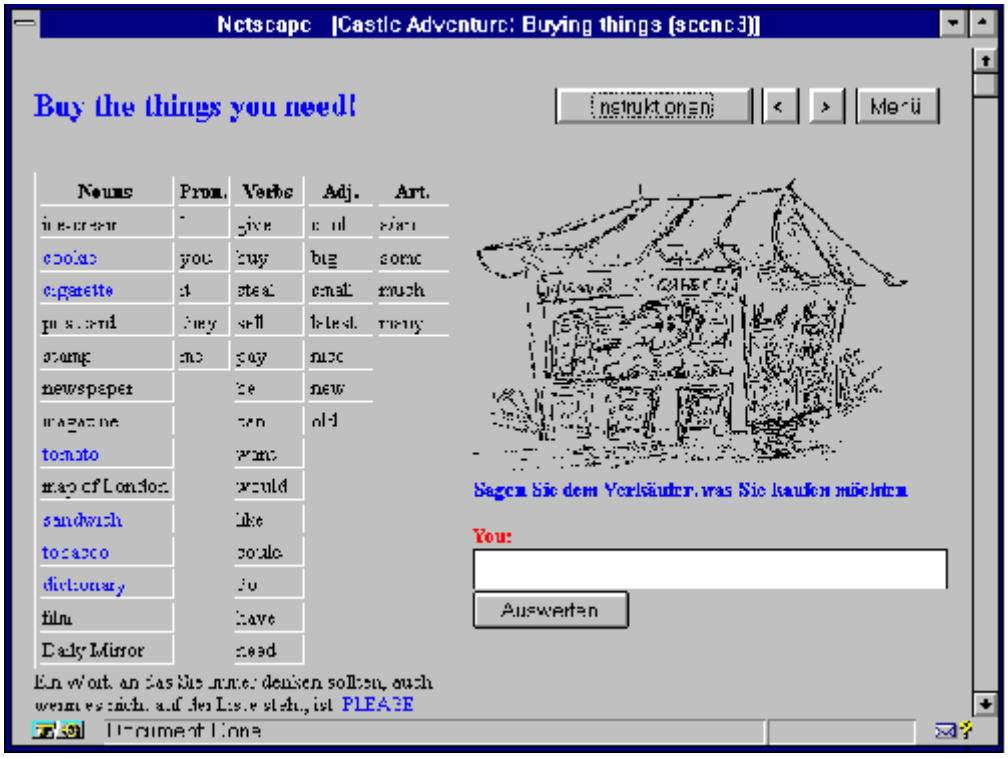


Fig. 1: Input Screen of RECALL Demonstrator (GERSTL & GRIESZL 1997)

The wordlists, however, allow for a wide variety of different sentence types. Additionally, specific grammatical topics may be dealt with more intensively in extra *remedial exercises.* As one alternative, these will be offered to the student after an error-cumulation in a particular error-category has been recognized. As a second alternative, the user may choose them from a menu whenever needed.

RECALL incorporates a *check* and *practice* system that is able to generate feedback and select suitable remedial exercises based on the individual learner requirements (cf. BOWERMAN 1991, KRASHEN 1995, FITZPATRICK & GRIESZL 1996).

A key question to be answered at the planning stage was the distribution of system control: Should the navigation be tutor-controlled or user-controlled or should there be a balance between both possible controlling agents? This question was answered in relation to the user level. The original platform for the demonstrator has considered exclusively tutor-controlled navigation for beginners and a tutor-suggested navigation for advanced learners. These tutor suggestions can be skipped or changed by extra buttons that are only provided on the advanced level. So for the advanced user the Demonstrator's design is a mixed initiative system. It has a tutor-controlled orientation

if the users do not want to make their decisions and does allow learner-controlled exploration if the user wishes to gain control.

Next, a teaching approach or methodology had to be incorporated into the system. One of the earliest approaches that is still used today is the *grammatical approach*. It has been argued, however, that this approach would undermine the development of *conversational skills*. Thus the Demonstrator has been realized as an eclectic system combining both these approaches.

The conversational or communicative skills are practiced during RECALL's scenario-based level. Story related exercises are offered in a fixed order. The exercises put the user into certain everyday situations. Thus, the system is superficially following a conversational approach. The user is allowed to enter almost free input and is not unnecessarily restricted in the kind of input she might produce. For instance, the learner can decide whether to pose a question or formulate a statement in response to a system question. Nonetheless, the diagnosis component is strictly linguistically oriented as is the selection and appearance of the remedial exercises. From the learner's perspective pure grammar exercises often appear a little boring and unnatural with learner interaction often restricted to multiple choice questions or simple gap filling texts. However, these types of exercises are the most amenable to automatic diagnosis that performs more than just pattern matching. So the decision has been made to wrap the less stimulating focus of the course's main objective, i. e. the production of grammatically correct sentences into a motivating scenario: an adventure game. The communication between user and system is guided by the following steps:

- *Gaining the attention of the learner:* This is achieved by the provision of a coherent story with the user actively prompted to participate in the game.
- *Demonstrating the relevance of the lesson:* This has been omitted in the Demonstrator since the environment was to recapitulate topics already taught in the classroom.
- *Instilling confidence to succeed in the learner:* The learner always has the opportunity to find out the correct answer with or without system assistance. Many systems allow for two guesses and automatically give the correct solution after the third wrong attempt. Other systems offer the next task without giving the solution. The latter kind of reaction can be very frustrating for the learner, more so than those systems that automatically give the solution. In our approach, the user is offered some help after the first attempt and more substantial help after the second attempt to encourage her to repair the error herself. The user may attempt to repair the error as many times as she wishes to allow her to explore a range of possible inputs. Chances to succeed are thus higher than they would be if the user was put under pressure.
- *Giving them satisfaction when they do succeed:* Every correct or successful entry is rewarded by a story-related reply.

Thus, the tutoring strategy is based on the method of leading the student to the solution instead of simply presenting it. This is founded on the belief that the learner is more likely to commit the correct version of the input to memory if she is able to recognize and correct errors on her own. The *act* of realizing and correcting an error after a hint has been given, occupies the brain longer and with more impact than the sole presentation of the solution or a system's correction. The user is always free to accept or reject the help devices. She may also accelerate the path to the solution if she feels over-tutored in a certain context.

The Tutoring Model has been designed to give detailed reactions to any kind of error (that the Diagnosis is able to detect) but *not all at one time.* Here, we follow the results of ALLWRIGHT & BAILEY 1991 who stated that the complexity of responses, both conceptually and practically, is much greater than previously imagined. The response to learner's errors must not be discouraging but still informative. Too much information results in confusion. Even though the TM may be able to provide comments for more than one error that has occurred in the current input it is not wise to display all this information at one time.

Every kind of over-tutoring that might lead to confusion for the user has been suppressed by a defined weighting of relevance. In the Demonstrator for instance a typo is regarded as less important (relevant) than a grammatical error. So in the case that both kinds of errors occur in the user's input only the most relevant one will be mentioned. Of course, this does not mean that spelling errors would be ignored. If a syntactically correct sentence should still contain any typo, the user will not only be notified, but also offered a corrected version which the user may accept or reject.

8 Conclusion

A system to be based on the RECALL architecture will most effectively address a limited range of language learning issues at a given level. It would not yet be feasible to build a system that, by itself as an independent tutoring tool, would take the learner from the beginner's to the advanced level. Until significant technological advances have been made in areas such as spoken dialogue, CALL systems will have to be complemented by more traditional classroom based teaching methods.

Angelika Storrer

Vom Grammatikbuch zur Hypertext-Grammatik

Methodisches Vorgehen bei der Hypertextualisierung nicht-standardisierter Textsorten

1 Einleitung

„Just as the best films are not made by putting a camera in the best seat of a theater, the best hypertexts are not made from text that was originally written for the linear medium." (NIELSEN 1995:323). Die besten und innovativsten Hypertext-Anwendungen sind sicherlich solche, die von Beginn an für das neue Medium konzipiert sind und dessen Mehrwerteigenschaften optimal zur Geltung bringen. Dies gilt sowohl für geschlossene Hypermedia-Anwendungen (z. B. auf CD-ROM publizierte hypermediale Nachschlagewerke, Lernprogramme und Kiosksysteme) als auch für Hyperdokumente, die in ein offenes Hypertextnetzwerk wie das World Wide Web integriert sind. Wie Jakob Nielsen im Nachsatz zum obigen Zitat jedoch selbst anmerkt, liegt so viel wertvolles Wissen in gedruckter Form vor, daß die Überführung dieses Wissens in Hypertext für lange Zeit noch ein Thema bleiben wird. Und – so wie es gute und schlechte Verfilmungen von Theaterstücken und Büchern gibt – kann auch eine solche Konversion zu mehr oder weniger guten Hypertext-Anwendungen führen.

Im folgenden möchte ich einige Überlegungen dazu anstellen, wie man als Hypertext-Autor aus dieser zweitbesten Ausgangslage das Beste machen kann. In Abschnitt 2 werde ich – im Anschluß an KUHLEN 1991 – verschiedene Strategien der Konversion unterscheiden und zeigen, daß der bei der Konversion zu betreibende Aufwand in umgekehrt proportionalem Verhältnis zum Standardisierungsgrad des zu konvertierenden Textes steht. In Abschnitt 3 wird das methodische Vorgehen skizziert werden, das für die Hypertextualisierung eines nicht-standardisierten Textes entwikkelt wurde. Es handelt sich um eine umfassende wissenschaftliche Grammatik des Deutschen, die am Institut für deutsche Sprache in Mannheim erarbeitet wurde und 1997 als Dreibänder in Buchform erschienen ist (ZIFONUN et al. 1997). Im Projekt GRAMMIS (Grundlagen eines grammatischen Informationssystems) wurden ausge-

wählte Themenbereiche dieser Grammatik in eine Hypermedia-Anwendung überführt und mit verschiedenen Testnutzern getestet. [1]

2 Strategien zur Konversion von Texten in Hyperdokumente

Als Hyperdokument bezeichne ich ein Netzwerk von Hypertext-Einheiten mit einem erkennbaren Thema und erkennbarer kommunikativer Funktion. Die Hypertext-Einheiten eines Hyperdokuments sind über Verknüpfungen (Hyperlinks) miteinander verbunden, wobei sowohl die Verwaltung der Einheiten als auch die Verknüpfungen von einer als Hypertextsystem bezeichneten Software verwaltet werden.

Grundsätzlich geht es bei der Konversion darum, einen maschinenlesbar vorliegenden, linear organisierten Ausgangstext in ein solches durch computerisierte Verweisungen verknüpftes Netzwerk von Hypertext-Einheiten zu überführen. Dabei sind im wesentlichen drei Teilaufgaben zu lösen:

a) Die Segmentierung des Ausgangstextes in Textsegmente,
b) die Umorganisation dieser Textsegmente in Hypertext-Einheiten,
c) die Strukturierung des Hyperdokuments durch Verknüpfungen und Verknüpfungsmuster.

Wenn man unter Hyperdokumenten nicht nur rein textuelle Netzwerke versteht, sondern auch die Verknüpfung von unterschiedlichen medialen Objekten (Bild, Ton, Video) mit einbezieht,[2] müssen zusätzliche Probleme gelöst werden: Über welche Sinneskanäle und mit Hilfe welcher Symbolsysteme kann die jeweilige Information am effektivsten vermittelt werden? Wann bietet sich eine Mehrfachkodierung derselben Information an, und wie stützen die verschiedenen medialen Angebote einander? Diese Fragen der Medienintegration erfordern interdisziplinäre Zusammenarbeit, deren bisherige Ergebnisse u. a. in ISSING & KLIMSA 1995, SCHULMEISTER 1996, BÖHLE et al. 1997 dokumentiert sind. Ich möchte mich jedoch auf die Aspekte konzentrieren, zu denen Text- und Computerlinguistik etwas beizutragen haben, nämlich die Segmentierung, Umgestaltung und Neurelationierung komplexer, überwiegend schriftlich fixierter Texte.

2.1 Differenzierung von Konversionsstrategien

Voraussetzung für die eigentliche Konversion ist natürlich, daß der zu konvertierende Ausgangstext, wenngleich für das gedruckte Medium konzipiert, in maschinenlesbarer Form vorliegt. Bei Texten neueren Erscheinungsdatums ist dies meist der Fall, da sie

---

[1] An der Entwicklung der verschiedenen Komponenten waren Eva BREINDL, Roman SCHNEIDER, Angelika STORRER und Bruno STRECKER (Leitung) beteiligt. Die in der Pilotphase entwickelten Prototypen können zu Testzwecken heruntergeladen werden:
http://www.ids-mannheim.de/grammis/download/download.html.

[2] Ich verwende den Terminus „Hypertext“ in diesem Aufsatz für nicht-lineare Informationsdarstellung im weiteren Sinne und schließe damit die oft auch als ”Hypermedia” bezeichnete Verbindung von Text-, Bild-, Ton- und Videoobjekten mit ein.

entweder bereits mit einem Textverarbeitungsprogramm erfaßt sind oder zumindest in der Form von Satzbanddateien vorliegen, die bei der Drucklegung vom Setzer mittels einer Lichtsatzmaschine erzeugt werden. Ältere Texte müssen entweder eingetippt oder gescannt und entsprechend korrigiert werden. Liegt eine maschinenlesbare Textversion vor, dann kann man mit KUHLEN 1991:163f vier Konversionsstrategien unterscheiden, die sich in den Kriterien unterscheiden, die zur Segmentierung und Neurelationierung herangezogen werden:

1) Bei der **einfachen Konversion** wird der Ausgangstext als Ganzes auf eine Hypertext-Einheit abgebildet. Deren Teile können durch intratextuelle Hyperlinks angereichert sein, indem z. B. die Einträge eines vorangestellten Inhaltsverzeichnisses mit den jeweiligen Kapitelanfängen verknüpft sind. Resultate einer solchen Konversionsstrategie sind beispielsweise HTML-Dokumente, die mit einem Textverarbeitungsprogramm erstellt und dann automatisch in HTML konvertiert wurden. Weitere Mehrwerte entstehen dabei allenfalls durch die Kombination mit Strategie 4 (intertextuelle Konversion), indem z. B. extratextuelle Verknüpfungen zu anderen HTML-Dokumenten im WWW angelegt werden. Beispiele für diesen Konversionstyp finden sich im WWW zuhauf, häufig handelt es sich um Vor- oder Parallelpublikationen von Aufsätzen, die in traditionellen wissenschaftlichen Journalen erscheinen.

2) Bei der **Segmentierung und Relationierung nach formalen Texteigenschaften** wird der Ausgangstext in Textsegmente wie Kapitel, Unterkapitel und Paragraphen zerlegt, die an der Textoberfläche gekennzeichnet sind. Diese werden dann in Analogie zur hierarchischen Dokumentenstruktur des Ausgangstextes wieder miteinander verknüpft, so daß die Struktur des Hyperdokuments ein Imitat der hierarchischen Dokumentenstruktur des gedruckten Ausgangstextes ist. Natürlich können auch hier weitere inter-, intra- und extratextuelle Verknüpfungen hinzutreten. Als Beispiel für ein derartiges Konversionsprodukt sei das amtliche Regelwerk zur neuen deutschen Rechtschreibung genannt, das im WWW-Angebot des Instituts für deutsche Sprache (IDS) publiziert ist (RSREFORM O. J.).

3) Erst die **Segmentierung und Relationierung nach Kohärenzkriterien** resultiert in einem wirklich als Netzwerk strukturierten Hyperdokument. Relationierung nach Kohärenzkriterien bedeutet im Idealfall, daß die Verknüpfungen so angelegt werden, daß es Benutzern mit unterschiedlichem Vorwissen und unterschiedlichen Interessen gelingt, bei ihrem individuellen Weg durch das Hyperdokument eine kohärente Wissensstruktur zum dort beschriebenen Gegenstand aufzubauen. Erst bei dieser Art von Konversion entsteht ein Hyperdokument, das die Mehrwerteigenschaften des neuen Mediums voll zum Tragen bringt. Als Beispiel für ein derartiges Konversionsprodukt im WWW kann die von MIT Press publizierte elektronische Biographie zu Noam Chomsky (BARSKY 1997) gelten.

4) Bei der **intertextuellen Konversion** schließlich werden mehrere unabhängig voneinander publizierte Ausgangstexte nach thematischen Kriterien systematisch miteinander verknüpft. Ein Beispiel im WWW ist die bereits erwähnte elektronische Fassung des Regelwerks zur neuen Rechtschreibung gelten, die systematisch mit einem Hyperdokument vernetzt wurde, das auf der Grundlage einer Sonderausgabe der Zeitschrift „Sprachreport" entstanden ist (HELLER 1996) und das die wichtigsten Neuerungen erläutert, die sich durch die Neuregelung ergeben haben.

Es dürfte klar sein, daß der bei der Konversion zu betreibende Aufwand in hohem Maße von der gewählten Konversionsstrategie abhängt: Strategien 1 und 2 orientieren sich stark an Merkmalen der Textoberfläche und lassen sich deshalb relativ einfach automatisieren. Strategien 3 und 4 hingegen erfordern ein inhaltliches Verständnis des Textes und Hypothesen über den Informationsbedarf der potentiellen Rezipienten, um sinnvolle und informative intra- und extratextuelle Verknüpfungen anzulegen. Der zusätzliche Aufwand, der betrieben werden muß, lohnt jedoch die Mühe. Schließlich sollen bei der Konversion nicht einfach die für das gedruckte Medium konzipierten Strukturen im Hyperdokument nachgebildet werden. Die Übernahme von Organisationsprinzipien und Zugriffsstrukturen vom gedruckten Buch ins elektronische Buch können in einer Zeit des Medienwechsels allenfalls als Krücke fungieren, an der sich ungeübte Benutzer zunächst festhalten können. Langfristig sind es aber gerade die im Buch nicht nachbildbaren Zugiffs- und Navigationsangebote, die Hypertext-Anwendungen attraktiv machen und die Nachteile des Bildschirmmediums gegenüber dem Buch, z. B. geringere Portabilität und schlechtere Lesbarkeit, aufwiegen.

Auf medienspezifische Angebote zur Informationserschließung muß bei der Konversion also besonders geachtet werden: Nicht Imitation sondern Rekonstruktion und mediengerechte Umsetzung der Strukturen und Inhalte lautet das von KUHLEN 1991:160 formulierte Desiderat für den Konversionsprozeß. Ein Blick in das Angebot des WWW zeigt jedoch, daß praktische Umsetzungen dieses Desiderats bislang eher selten sind. Dies mag daran liegen, daß viele Dokumente im Zuge einer Dabeisein-ist-alles-Mentalität gar nicht primär zum Lesen am Bildschirm, sondern zum Ausdrucken ins WWW gestellt werden.[3] Es mag aber auch daran liegen, daß erst Erfahrungen gesammelt und Ideen entwickelt werden müssen, welche Textsorten in welcher Weise als Hypertexte aufbereitet werden müssen, um einen größtmöglichen Mehrwert gegenüber dem gedruckten Medium zu schaffen.

Natürlich werden die Konversionsstrategie und der zu betreibende Aufwand im konkreten Fall von finanziellen und zeitlichen Rahmenbedingungen eines Konversionsprojekts und der jeweils zur Verfügung stehenden Konversions- und Hypertextsoftware bestimmt. Unabhängig davon gibt es aber, wie im folgenden gezeigt wird, weitere Parameter, die den Konversions-Aufwand beeinflussen: der Standardisie-

[3] Tatsächlich zeigen Nutzerumfragen, daß das WWW vielfach vor allem zum Stöbern und Suchen nach Dokumenten benutzt wird, die dann jedoch ausgedruckt und auf Papier rezipiert und archiviert werden (vgl. z. B. *http://www.cc.gatech.edu/gvu/user_surveys/*).

rungsgrad der zu konvertierenden Textsorte und bestimmte Textstrukturmerkmale des Ausgangstexts.

2.2 Abhängigkeit des Aufwands vom Standardisierungsgrad der Textsorte

Daß das Wissen um die Bauweise von Textsorten – im weiteren Textmuster genannt – die Produktion und Rezeption von Texten maßgeblich beeinflußt, ist in der textlinguistischen und psycholinguistischen Textrezeptions- und Textproduktionsforschung unumstritten (vgl. z. B. SCHNOTZ 1994, SANDIG 1997). Textsorten unterscheiden sich nun dadurch, inwieweit die für sie charakteristischen Textmuster verbindlich oder variabel sind. Ich möchte drei Kategorien unterscheiden:

A) *Textsorten mit standardisiertem Textmuster:* Prototypische Beispiele sind Bibliographien, Telefonbücher, Wörterbücher, Enzyklopädien, Gesetzeswerke. Die entsprechenden Textexemplare werden meist von einer Gruppe von Autoren geschrieben; diese sind beim Verfassen ihrer Textsegmente an Vorgaben gebunden, die meist sogar schriftlich fixiert vorliegen. Ein Beispiel hierfür sind die in großen Wörterbuchverlagen benutzten Artikelstrukturprogramme, die für den Aufbau komplexer Wörterbuchartikel verbindlich sind. Solche Vorgaben können in modernen SGML-basierten Redaktionssystemen mittlerweile über als DTD repräsentierte Artikelstrukturgrammatiken formuliert und kontrolliert werden; in ihnen sind z. B. die Typen der verwendeten Textsegmente und Anordnung im Text, Konventionen für Abkürzungen, ein festgelegtes Beschreibungsvokabular sowie die Längenbegrenzung der Textsegmente festgelegt.[4]

B) *Textsorten mit konventionell vorgegebenem Textmuster:* Prototypische Beispiele sind Geschäftsbriefe, studentische Seminararbeiten, wissenschaftliche Untersuchungsberichte. Für die Textexemplare dieser Kategorie existiert zwar kein explizit festgelegter Bauplan, ihr Aufbau wird aber durch konventionalisierte Textmuster bestimmt, die sich im Gebrauch dieser Textsorten etabliert haben und Teil des Wissens einer Kulturgemeinschaft sind. Diese Textmuster sind jedoch nicht verbindlich; sie können zur Erzeugung von stilistischen Effekten gemischt werden (vgl. SANDIG 1989), wenn der Autor bereit ist, die sich daraus ergebenden Konsequenzen bei der Rezeption in Kauf zu nehmen.

C) *Textsorten mit variablen Textmustern:* Prototypische Beispiele sind Reiseführer, Ratgeberbücher, Hand- und Lehrbücher. Obwohl sich auch bei diesen Textsorten eine Reihe von möglichen Strukturierungsvarianten herausgebildet hat, ist die Anordnung der Textteile und die Bezüge zwischen ihnen im wesentlichen bestimmt von der Struktur des behandelten Gegenstands, der Textfunktion und von Hypothesen über Interessen und Wissensvoraussetzungen der potentiellen Rezi-

[4] Zu Funktion und Formen der Standardisierung von Wörterbuchtexten vgl. WIEGAND 1987 und WIEGAND 1997.

pienten. Natürlich müssen auch hier bestimmte Strukturerwartungen der Rezipienten berücksichtigt werden, z. B. daß Inhaltsverzeichnis und Einleitung am Anfang, ein Stichwortverzeichnis eher am Ende des Buches zu finden ist, ansonsten bleibt den Autoren solcher Texte viel Spielraum bei der Textgestaltung.

Die Zuordnung von Textsorten zu den Kategorien ist im Einzelfall nicht unproblematisch.[5] Die Kategorien geben aber Anhaltspunkte für die Abschätzung des Aufwandes, der für die Konversion von Textexemplaren betrieben werden muß. Als Faustregel gilt: Je stärker Texte nach einem verbindlich oder zumindest konventionell vorgegeben Baumuster strukturiert sind, wie dies bei Textsorten der Kategorie A und B der Fall ist, desto einfacher kann die funktional-semantische Kategorie von Textsegmenten automatisch anhand ihrer Position im Textganzen bestimmt werden. So sind z. B. sog. Wörterbuchparser in der Lage, standardisierte Wörterbuchartikel anhand einer Artikelstrukturgrammatik in Textsegmente zu zerlegen und diesen Segmenten die richtige funktional-semantische Kategorie (z. B. Grammatikangabe, Beispiel etc.) zuzuweisen (vgl. BLÄSER & WERMKE 1990; HAUSER & STORRER 1994). In dem Maße, in dem der Grad der Standardisierung abnimmt, wächst der intellektuell zu betreibende Aufwand für die Analyse des Ausgangstextes, der für die mehrwerterzeugenden Konversionsstrategien 3 und 4 (s. o. 2.1.) benötigt wird.

Aus diesen Überlegungen folgt, daß für die Hypertextualisierung von Textsorten der Kategorie C mit einem erheblichen intellektuellen Analyse- und Nachbearbeitungsaufwand gerechnet werden muß, wenn man die Möglichkeiten des neuen Mediums wirklich ausreizen möchte. Es lohnt sich also, darüber nachzudenken, ob der in Frage stehende Text überhaupt durch Hypertextualisierung gewinnen kann oder ob er nicht besser auch im elektronischen Medium als fortlaufender linearer Text angeboten werden soll, z. B. als PDF-Dokument[6]. Als lohnenswerte Kandidaten gelten für eine Entlinearisierung gelten:[7]

a) Texte, die eher zum punktuellen Nachschlagen als zum Durcharbeiten gedacht sind,
b) Texte, deren Benutzer in verschiedenen Benutzungssituationen unterschiedliche Interessen verfolgen, bei denen es sich also lohnt, verschiedene, gleichberechtigte Lesewege anzubieten und unterschiedliche Perspektiven auf den behandelten Gegenstand zu werfen,

---

[5] Einerseits gibt es zwischen Textsorten mit konventionellen und mit variablen Textmustern fließende Übergänge; andererseits gibt es innerhalb derselben Textsorte Exemplare, die verschiedenen Kategorien zugeordnet werden müßten. So gibt es neben dem Normalfall des Wörterbuchs mit stark standardisiertem Baumuster Wörterbuchtexte wie das Wörterbuch "Dummdeutsch" (HENSCHEID et al. 1985), bei denen nur die alphabetische Anordnung der Artikel verbindlich ist, die also den Kategorien B oder C zuzurechnen wären.

[6] Das von ADOBE entwickelte *Portable Document Format* (PDF) erleichtert die elektronische Publikation von komplett formatierten Dokumenten im WWW und verbindet die Vorteile der Seitenbeschreibungssprache *PostScript* mit den Hypertext-Eigenschaften von HTML.

[7] Vgl. Kuhlen (1991, 2.4.3) und Gloor 1990.

c) Texte, bei denen die Einbindung von Ton, bewegter Graphik und Video die Wissenskodierung und -vermittlung vereinfacht,
d) Texte, die einen Gegenstand behandeln, der sich rasch verändert, die also häufig aktualisiert werden müssen.

Wissenschaftliche Grammatiken, wie die am Institut für deutsche Sprache entwickelte „Grammatik der deutschen Sprache", erfüllen zumindest die ersten drei Kriterien:

- Der gedruckte Dreibänder wird wohl nur von wenigen Interessierten vollständig gelesen werden; vielmehr ist mit einer Vielfalt von Rezipienten mit höchst unterschiedlichen Interessen und Wissensvoraussetzungen zu rechnen.
- Der Gegenstandsbereich „grammatische Strukturen des Deutschen" kann unter sehr unterschiedlichen Perspektiven und mit verschiedenen „theoretischen Brillen" untersucht und beschrieben werden. Die Komplexität des Gegenstands spiegelt sich in vielen expliziten und impliziten Verweisen zwischen den Textsegmenten der Grammatik, die bei der Hypertextualisierung durch Verknüpfungen nachgebildet werden können.
- Dazu gewinnen Phänomene der gesprochenen Sprache und der Wortstellungsregularitäten an Anschaulichkeit, wenn mit multimedialen Elementen wie Tondateien und mit „animierten" Satzbeispielen gearbeitet werden kann.

Deshalb wird am Institut für deutsche Sprache seit 1993 das Projekt GRAMMIS durchgeführt, in dem auf der Grundlage der gedruckten Grammatik ein multimediales grammatisches Informationssystem aufgebaut wird. Die im Laufe der 1997 abgeschlossenen Pilotphase entwickelten Prototypen behandeln ausgewählte Themenbereiche und gaben ersten Aufschluß über Aufwand und Nutzen einer solchen Konversion.[8] Da sich ein derart komplexer Text sich jedoch nicht ohne methodische Systematik hypertextualisieren läßt, wurde für den Prototyp GRAMMIS-1 das im folgenden Abschnitt beschriebene methodische Vorgehen entwickelt. Da es in seinen Grundzügen textsortenübergreifend gehalten ist, gibt es generell Aufschluß über die verschiedenen Arbeitsschritte, die bei der Konversion von nicht-standardisierten Textsorten nach Strategie 3 (vgl. 2.1.) nötig sind. Bei den Prototypen handelt es sich um geschlossene Hypertext-Anwendungen, die mit dem Autorensystem *Toolbook* (*Asymetrix*, Version 4.0) und dem Datenbanksystem *Paradox* (*Borland*, Version 4.5) entwickelt wurden. Aspekte der extratextuellen Verknüpfung mit Hyperdokumenten anderer Autoren, wie sie bei der Diskussion von Strategie 4 in 2.1. angesprochen wurden, sind deshalb in dieser Methode unterrepräsentiert.

[8] Vgl. STORRER 1997 und die GRAMMIS-spezifischen Beiträge in STORRER & HARRIEHAUSEN 1998.

3 Methodisches Vorgehen bei der Konversion einer Print-Grammatik in Hyperdokumente

Die Methode, die der Hypertextualisierung von ausgewählten Kapiteln der „Grammatik der deutschen Sprache" zugrundelag, unterscheidet zwei Schritte: die funktional-holistische Analyse des Ausgangstextes und daran anschließend die Strukturierung des Hyperdokuments.

3.1 Schritt 1: funktional-holistische Textanalyse

Texte werden von Produzenten und Rezipienten nicht losgelöst von einer übergreifenden Kommunikationssituation wahrgenommen; vielmehr werden sowohl Textganzes als auch die einzelnen Textsegmente systematisch im Hinblick auf ihre Funktion in einem textübergreifenden Zusammenhang interpretiert. Entsprechend müssen bei der Analyse neben den eigentlichen Strukturmerkmalen des Ausgangstextes auch Autoren- und Lesermerkmale mit einbezogen werden.

a) Autorenmerkmale berücksichtigen die globalen und spezifischen Ziele, die den Autor bei der Produktion des Textes geleitet haben, sein Vorwissen sowie seine Hypothesen über das Vorwissen und die Kommunikationsziele der Rezipienten. Autorenmerkmale sind für die (recht häufigen) Fälle wichtig, in denen der Autor des Hypertextes nicht mit dem Autor des Ausgangstextes identisch ist. In dieser Situation, die NIELSEN 1995:326 recht zutreffend mit der nachträglichen Kolorierung eines Schwarz-Weißfilms vergleicht, basieren Segmentierung und Neurelationierung auf Hypothesen des Hypertextautors über die Absichten und Zielsetzungen des ursprünglichen Autors.

b) Lesermerkmale beziehen sich auf das Vorwissen der Leser über den im Text behandelten Gegenstandsbereich, auf deren Erwartungen bezüglich Textmuster, Zugriffsstrukturen usw. Weiterhin wichtig sind die Interessen und Zielsetzungen, mit denen der Text üblicherweise gelesen wird, sowie die Rezeptionsformen, d. h. Suchen, Durchlesen oder selektives Informationslesen, Lesen in Lernkontexten etc. Da es zu vielen Textsorten – Grammatiken gehören auch dazu – wenig bis keine Lese- und Benutzungsforschung gibt, können viele Lesermerkmale lediglich aufgrund eigener Lesegewohnheiten antizipiert werden.

c) Textstrukturmerkmale schließlich beziehen sich auf die Art und Weise, wie das im Ausgangstext externalisierte Wissen portioniert, sequenziert und relationiert ist. Für die Konversion müssen insbesondere die nicht-linearen Beziehungen zwischen Textsegmenten herausgearbeitet werden, wie sie auch für linear organisierte Texte charakteristisch sind:

1. Zwischen Textsegmenten gibt es Teil-Ganzes-Beziehungen (z. B. Unterkapitel-zu-Kapitel, Paragraph-zu-Unterkapitel), durch die die *hierarchische Dokumentenstruktur* des Textes festgelegt wird.

2. Zugriffsstrukturen wie Inhaltsverzeichnis, Stichwortverzeichnis, Wortregister erleichtern den gezielten Zugriff auf Textsegmente meist mit Blick auf verschiedene Situationen der Benutzung.
3. Innerhalb und zwischen den Textsegmenten gibt es explizite Verweise mit unterschiedlicher Funktion. Als explizite Verweise zählen Textsequenzen, die über ein Verweissymbol (z. B. einen Pfeil) oder einen Verweisausdruck (z. B. „siehe“, „vgl.“) und eine Verweiszielangabe („Kap.1.4“, „Kuhlen 1991“) bestehen. Wichtig ist die Differenzierung zwischen makrostrukturellen Verweisen auf andere Textsegmente im fortlaufenden Text, mikrostrukturellen Verweisen innerhalb desselben Textsegments (z. B. innerhalb eines Wörterbuchartikels) und intertextuellen Verweisen auf andere Texte (z. B. auf bibliographische Angaben, Belegstellen etc.). Die Gesamtheit der verschiedenen Verweisrelationen konstituiert die *Verweisstruktur* des Textes.
4. Innerhalb und zwischen den Textsegmenten gibt es (meist implizit) rhetorische Beziehungen (z. B. Regel-Beispiel; Regel-Regelausnahme-Ausnahmebeispiel), die wesentlich zur Herstellung lokaler und globaler Kohärenz beitragen. Diese können mit einem textsortenspezifisch zu entwickelnden Inventar rhetorischer Relationen beschrieben werden, aufbauend auf dem Inventar, das in der *Rhetorical Structure Theory* (MANN & THOMPSON 1988) vorgeschlagen worden ist.

Anhand der Analyse der hierarchischen Dokumentenstruktur wird der Ausgangstext in elementare Textsegmente unterteilt, die im wesentlich auch den kleinsten formal erkennbaren Gliederungseinheiten, den Paragraphen, entsprechen. Die dabei entstehenden Textsegmente wurden weiter im Hinblick auf Verweisstruktur und rhetorische Relationen analysiert. In der Pilotphase von GRAMMIS geschah dies sehr traditionell unter Verwendung einer gedruckten Vorlage, Bleistift und Buntstiften. Die bei der Analyse gewonnenen Segmente wurden dann relativ zügig aus dem Textverarbeitungsprogramm in die entsprechenden Themen-Einheiten (s. u.) der Prototypen integriert und dann weiterverarbeitet. Dies hatte zwar den Vorteil, allmählich ein Gefühl für die Charakteristika der Textsorte zu entwickeln. Für die künftige Entwicklung wäre nun aber wünschenswert, Werkzeuge zur automatischen Segmentierung in Anlehnung die hierarchische Dokumentenstruktur einzusetzen. Wenn man die Analyseergebnisse der Verweisbeziehungen und der rhetorischen Relationen mit einer Dokumentbeschreibungssprache, z. B. SGML, direkt im maschinenlesbaren Ausgangstext festhalten könnte, ließen sich viele der im zweiten Schritt (s. u.) zu bewältigenden Aufgaben automatisieren.

Eine DTD für wissenschaftliche Grammatiken zu entwickeln ist aber sicherlich nicht trivial. Problematisch, auch schon für die Drucklegung einer solchen Grammatik, ist der ständige Wechsel zwischen Objektsprache (der grammatisch beschriebene Einzelsprache) der formalen Metasprache, mit der die Regularitäten der Objektsprache formal beschrieben sind, und der natürlichen Metasprache (der Sprache des Grammatiktextes). Die Ebenentrennung ist zwar typographisch gekennzeichnet, die Kenn-

zeichnung ist jedoch abhängig vom Typ des Textsegments: Im fortlaufenden Text erscheinen z. B. objektsprachliche Ausdrücke kursiv, in den Beispielen recte. Die formale Metasprache umfaßt eine Vielzahl von Notationskonventionen für die Darstellung syntaktischer Strukturen und semantisch-logischer Beziehungen. Weiterhin gibt es Konventionen für die Markierung von Akzentsetzung und Intonationsverläufen; die Beispiele der gesprochenen Sprache umfassen Transkriptionen in der dafür üblichen „Partiturschreibweise", also viele ungewöhnliche Textpassagen, die dazu noch auf vielfache Weise thematisch miteinander verknüpft sind.

3.2 Schritt 2: Strukturierung des Hyperdokuments

Die in Schritt 1 gewonnenen Textsegmente werden im zweiten Schritt hypertextgerecht umgearbeitet und durch Verknüpfungen miteinander vernetzt. Dabei lassen sich die bei der Analyse in Schritt 1 gewonnenen Strukturen nutzen, auch wenn es bei der Konversion im Sinne des o. g. Desiderats vor allem darum gehen muß, über die im Buch vorhandenen Strukturen hinauszugehen und Nichtlinearität zum dominierenden Gestaltungsprinzip zu machen. Schritt 2 kann weiter in vier Teilschritte untergliedert werden:

Im ersten Teilschritt werden verschiedene funktionale Typen von Hypertext-Einheiten unterschieden. Im grammatischen Informationssystem sind dies beispielsweise

- Hypertext-Einheiten mit Angaben zu grammatischen Themen (Themen-Einheiten), im wesentlichen die Paragraphen des Ausgangstextes.
- Hypertext-Einheiten mit Definitionen und Erläuterungen zu grammatischen Termini (Glossar-Einheiten) werden aus entsprechenden Textsegmenten extrahiert und genereller formuliert.
- Hypertext-Einheiten mit grammatischen Informationen zu einzelnen Lexemen (Lexembeschreibungs-Einheiten) werden anhand der Lexembeschreibungen in der Grammatik bzw. im Sinne der dort skizzierten Kriterien aufgebaut (z. B. eine Datenbank der Funktionswörter des Deutschen).
- Hypertext-Einheiten mit Überblicksdarstellungen über die thematische Strukturierung eines grammatischen Teilgebiets (Überblicks-Einheiten).
- Hypertext-Einheiten mit verschiedene Arten von Metainformationen zum System (Struktur, Bedienung, Autorenschaft etc.).

Für jeden Typ muß ein charakteristisches Aufbauschema festgelegt werden. Die Themen-Einheiten des Wortartenbuchs in GRAMMIS-1 (vgl. z. B. Abb. 1) sind folgendermaßen aufgebaut: Aus der linken Seite wird das behandelte Teilthema textuell-abstrakt abgehandelt; auf der rechten Seite finden sich verschiedenen Typen von Beispielen, Graphiken und Animationen, die als zusätzliche Informationsangebote gedacht sind. Die Glossar-Einträge (vgl. z. B. Abb. 2) enthalten neben einer kurzen Definition mehrere Verwendungsbeispiele; weiterhin wurde versucht, Bezüge zu entsprechenden Termini anderer Grammatiken, insbesondere der Schulgrammatik, herzustellen und

auf die Besonderheiten der Terminologie der „Grammatik der deutschen Sprache“ hinzuweisen. Die Lexembeschreibungs-Einheiten sind nach der Art einer Karteikarte aufgebaut (vgl. Abb. 3); die primitive Datenbankfunktion von Toolbook erlaubt dabei die Suche in den verschiedenen Datensatzfeldern.

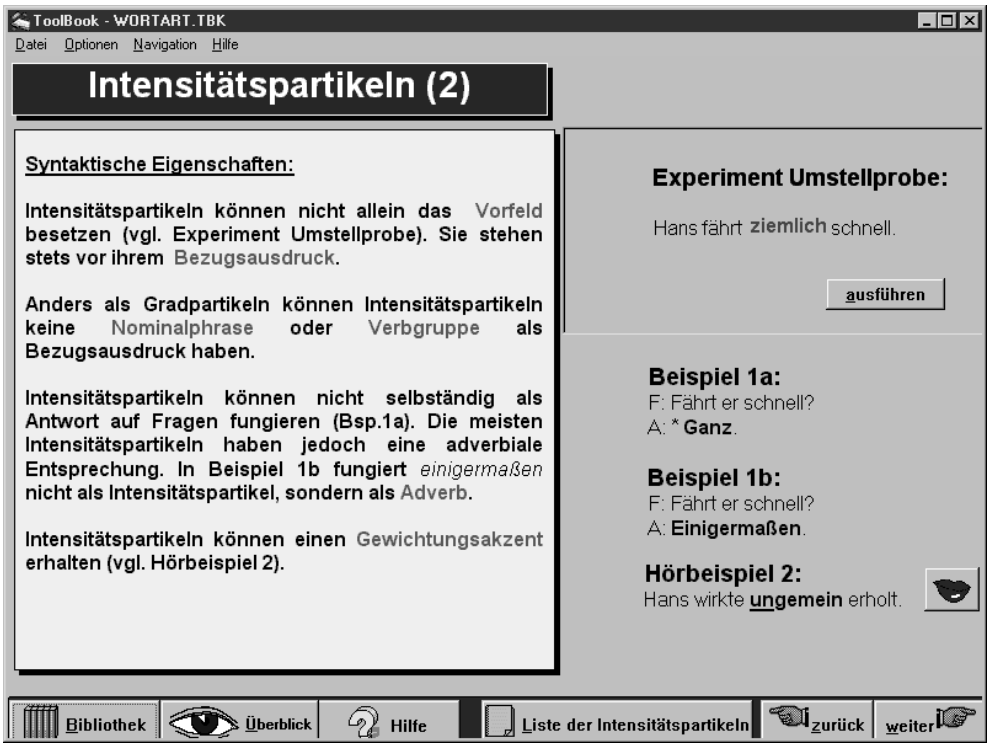


Abb. 1: Themen-Einheit zu den Intensitätspartikeln in GRAMMIS-1

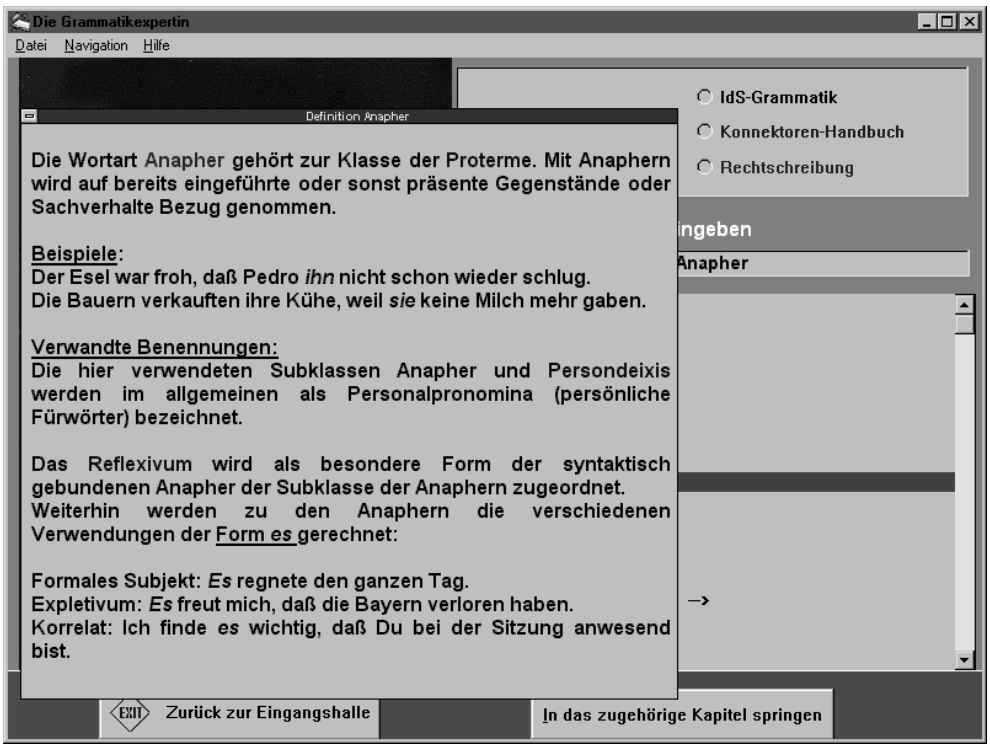


Abb. 2: Glossar-Einheit zum Terminus „Anapher“

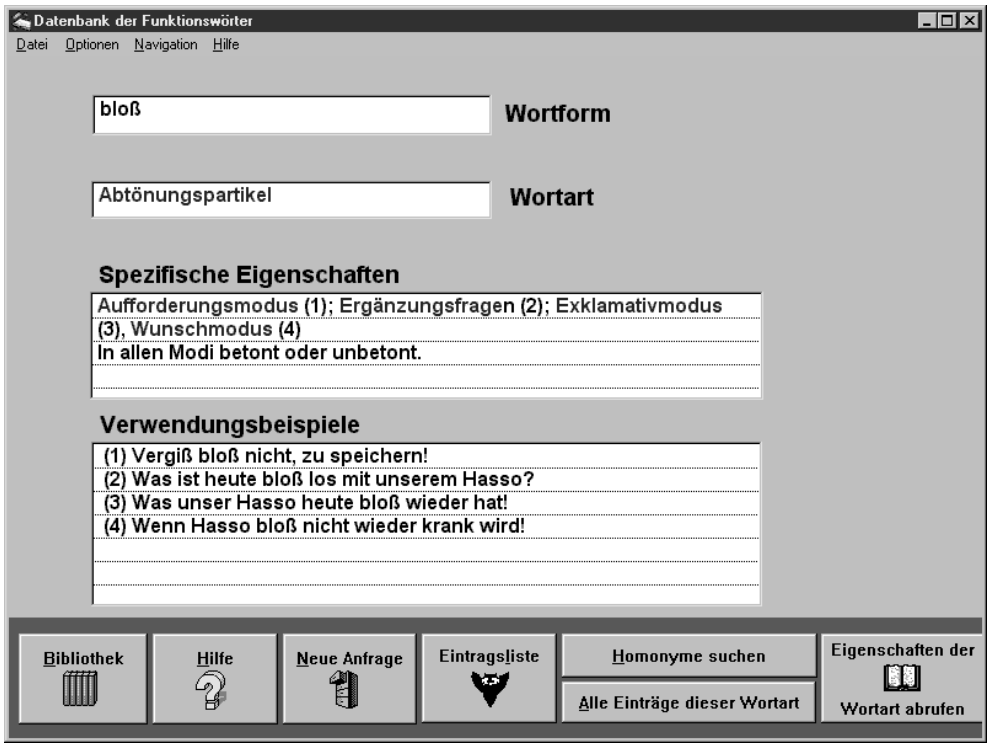


Abb. 3: Lexembeschreibungs-Einheit zur Abtönungspartikel „bloß“

Im zweiten Teilschritt werden die Textsegmente des Ausgangstextes in Hypertext-Einheiten umgestaltet, wobei sich wiederum drei Aufgaben ergeben: Mediale Anreicherung, Restrukturierung und Reformulierung.

a) *Mediale Anreicherung:* Texteinheiten können um multimediale Elemente – Tonbeispiele, Graphik, Animationen – angereichert werden, sofern dies sinnvoll ist. So kann in dem in Abb. 1 gezeigten Beispiel ein Verschiebeprobe-Experiment ausgeführt werden, daß die theoretische Aussage „Intensitätspartikeln können nicht alleine das Vorfeld besetzen“ am Beispiel beweist. Die Aussagen zur Akzentuierbarkeit von Intensitätspartikeln werden an einem Hörbeispiel exemplifiziert.

b) *Restrukturierung* kann notwendig sein, um das Textsegment dem Aufbauschema der Hypertext-Einheit anzupassen. So werden die im Ausgangstext enthaltenen Beispiele dem für die Themen-Einheiten gewählten Aufbauschema gemäß in den rechten Textblock des Bildschirms ausgelagert und durch eine entsprechende Verweisangabe mit dem linken Textteil verbunden. Viele Probleme bereitete der im ersten Prototyp gewählte Ansatz, keine Rollbalken anzubieten, sondern die Hypertext-Einheiten so anzulegen, daß sie auf einer Bildschirmseite darstellbar sind. Diese Vorgabe führte zu teilweise kontraintuitiven Segmentierungen und wurde deshalb in späteren Prototypen wieder aufgegeben, auch wenn das Grundprinzip einer kartenorientierten Modellierung weiterhin beibehalten wurde. Ebenfalls verzichtet wurde auf die platzintensive Aufteilung der Themeneinheiten in textuelle Kernaussage und zusätzliche Beispiele. Erläuternde Beispiele und Zusatzinformationen werden in späteren Komponenten nach Bedarf in eingebetteten Fenstern parallel angezeigt.

c) *Reformulierung* wird notwendig, wenn Kohärenzbildungshilfen verwendet werden, die auf die lineare Abfolge der Textsegmente im gedruckten Medium rekurrieren. Explizite Verweiszielangaben müssen in Verknüpfungen umgewandelt werden, da die für gedruckte Texte typischen Verweisziele (Seitenzahlen, Kapitelnumerierung) im Hyperdokument nicht mehr in der Form angelegt sind. Anaphorische Ausdrücke, die über die Hypertext-Einheit hinausweisen, müssen durch ihr Antezedens ersetzt, Konnektoren und konnektive Floskeln eliminiert und die darin ausgedrückten Bezüge auf andere Art realisiert werden. Erst diese Umarbeitung führt zu den „kohäsiv geschlossenen" (KUHLEN 1991:34 u.87) Hypertext-Einheiten, die Voraussetzung dafür sind, daß das Hyperdokument tatsächlich auf verschiedenen Lesewegen rezipiert werden kann.

Der Aufbau der GRAMMIS-Prototypen hat deutlich gemacht, daß der Paragraph als kleinstes formal erkennbares Textsegment nicht zwangsläufig eine funktional und kohäsiv abgeschlossene Einheit ist. Während man paragraphenübergreifende Kohäsionsmittel im einführenden Kapitel zur Wortartenklassifikation eher selten findet, sind sie in weiterführenden Kapiteln eher die Regel als die Ausnahme. Ein Blick z. B. in die einführenden Bemerkungen zum Thema „Verfahren zur Klassifizierung der Komplemente" (ZIFONUN, HOFMANN & STRECKER 1997; E 2.2:1070ff) zeigt, daß die Mehrzahl der Paragraphenanfänge entweder explizite Anaphern[9] enthält oder zumindest implizit[10] an den Vortext anknüpft. Der Reformulierungsaufwand stieg also proportional zum Grad der Detailliertheit, mit der ein grammatisches Phänomen beschrieben und auf dem Hintergrund eines umfassenden Forschungskontextes erörtert wurde. Wir haben schließlich diese Häufung von kohäsiven Bezügen zwischen Paragraphen als Indiz gewertet, daß die entsprechenden Textsegmente auch im Hyperdokument am günstigsten in der vorgesehenen Abfolge zu rezipieren sind. In GRAMMIS-2 wurde also auf eine Umarbeitung in kohäsiv geschlossene Einheiten verzichtet, mit allen Konsequenzen, die dies für den Hypertext-Rezipienten mit sich bringt, und die vorgegebene Abfolge durch entsprechende Pfade nachgebildet.

Im dritten Teilschritt werden die umgestalteten Hypertext-Einheiten nach verschiedenen Prinzipien neu relationiert:

a) *Relationierung in Anlehnung an die hierarchische Dokumentenstruktur:* Die Verknüpfung der Hypertext-Einheiten analog zur hierarchischen Gliederung in Kapitel, Unterkapitel und Paragraphen kann als Strukturskelett für das Hyperdokument übernommen werden. Wichtig ist es jedoch, einen Strukturüberblick über die Dokumentenstruktur anzubieten, der von jeder Hypertext-Einheit aus aktiviert werden kann und von dem aus zum Beginn eines jeden Unterkapitels gesprungen werden kann. Erst dadurch kann sich ein Benutzer wirklich von einer vorgegeben Abfolge lösen und seinen

---

[9] Pronomina („Auf der Grundlage *dieser* Restriktionen (...)") oder anaphorische Nominalphrasen („*Das Problem* entpuppt sich aber als nur scheinbar (...)").

[10] „Erschwerend *kommt hinzu* (...)"; „*Trotz* der Vielfalt der Leitformen (...)".

eigenen Weg durch den Hypertext wählen; die auf den Buchdruck hin ausgerichtete Kapitelnumerierung wird dabei überflüssig.

b) *Relationierung in Anlehnung an die Verweisstruktur:* Eine wichtige Rolle für die Relationierung spielen die verschiedenen Typen von expliziten Verweisen. Diese können durch Hyperlinks nachgebildet werden, wobei die Unterscheidung in makro-, mikro und intertextuelle Verweise jeweils als inter-, intra- oder extratextuelle Verknüpfungen nachzubilden sind. Vor allem bei im WWW publizierten Hyperdokumenten ist es wichtig, extratextuelle Verknüpfungen, die ja aus dem eigentlichen Hyperdokument herausführen, als solche zu kennzeichnen. Weiterhin muß jeweils entschieden werden, ob das Verweisziel (die Hypertext-Einheit, auf die die Verknüpfung verweist) parallel z. B. in einem darüberliegenden Fenster angezeigt wird oder ob das Verweisziel die ursprüngliche Hypertext-Einheit ersetzt.[11] Die Entscheidung richtet sich dabei nach funktionalen Kriterien: Wird die Information des Verweisziels vermutlich zur Ergänzung und dem bessern Verständnis der aktuell rezipierten Einheit benötigt, bietet sich eine parallele oder eingebettete Anzeige an (z. B. für Glossar-Einträge, Beispiele, Literaturangaben). Eine ersetzende Anzeige ist dann angebracht, wenn im Verweisziel ein neuer, weiterführender Aspekt zum aktuellen Thema behandelt wird (Themenvertiefung, Themenwechsel).

c) *Hinzufügung zusätzlicher Verknüpfungen nach Kohärenz- und Relevanzkriterien*: Während sich die Verknüpfungen des Typs a) und b) relativ schematisch anlegen lassen und von daher auch recht gut automatisierbar sind[12], erfordert die Verknüpfung nach Kohärenz- und Relevanzkriterien zunächst ein Verständnis des Textinhalts und Hypothesen über den Informationsbedarf der Rezipienten in verschiedenen Benutzungssituationen, die in Schritt 1 als Lesermerkmale bezeichnet wurden. Daraus können dann relevante Verknüpfungstypen abgeleitet werden, die sich dann wieder (teil)automatisieren lassen. Im grammatischen Informationssystem sind z. B. alle in den Hypertext-Einheiten vorkommenden grammatischen Termini mit den entsprechenden Glossar-Einheiten verknüpft, die durch Mausklick in einem eigenen Fenster angezeigt werden. Ab dem zweiten Prototyp werden Verknüpfungen zu bibliographischen Angaben angeboten, die in einer Literaturdatenbank verwaltet werden. Weitere Verknüpfungen erfordern dann jedoch ein tiefergehendes Verständnis des behandelten grammatischen Gegenstandsbereichs und gute Hypothesen über potentielle Verstehensschwierigkeiten der Benutzer: Z. B. ist es sinnvoll, die Themen-Einheiten, in denen die Subklassen der Pronomina (z. B. Possessivum) behandelt werden, mit den Themen-Einheiten zu verknüpfen, in denen die entsprechenden Artikelwörter (possessives Determinativ) abgehandelt sind. Sinnvoll ist auch die Vernetzung von Wortarten

---

[11] Vgl. die Unterscheidung in ersetzende, parallele oder eingebettete Anzeige in KUHLEN (1991, S.16). Eine Wahl kann natürlich nur getroffen werden, wenn das benutzte Hypertext-System überhaupt alle Optionen anbietet; im WWW dominierte lange die ersetzende Anzeige, allenfalls ergänzt um die parallele Anzeige mittels der problematischen Frame-Technik.

[12] Vgl. z. B. REARICK 1991, RINER 1991, SARRE/GÜNTZER 1990, HAMMWÖHNER 1990.

mit syntaktischen Kategorien (Adverb vs. Adverbiale), von satzsemantisch motivierten Kombinationskategorien mit morpho-syntaktisch motivierten Konstruktionskategorien etc. Solche Verknüpfungen in der Hypertext-Grammatik führen zu erheblichen Mehrwerten gegenüber gedruckten Grammatiken, in denen meist häufig hin- und hergeblättert werden muß, bis die gesuchte Information gefunden und verstanden wird (vgl. STORRER 1998). Auch wenn die Kompetenz, die für die Herstellung solcher Verknüpfungen benötigt wird, auf absehbare Zeit wohl nicht mit dem Computer simuliert werden kann, liegt in ihnen genau der Vorteil des neuen Mediums für die Verfasser und die Benutzer von Grammatiken (STRECKER 1998).

Neben den vom Autor angelegten Verknüpfungen sollte ein gutes Hypertextsystem auch den Benutzern die Möglichkeit geben, eigene Verknüpfungen nach Kohärenz- und Relevanzgesichtspunkten anzulegen. Im Prototyp GRAMMIS-1 können Hypertext-Einheiten mit etikettierten Lesezeichen markiert werden, die beim Verlassen des Systems unter dem beim Systemaufruf eingegeben Benutzernamen gespeichert bleiben. Mit dem Annotationseditor können Anmerkungen zu Textteilen eingefügt und farblich gekennzeichnet werden. Im Prototyp GRAMMIS-2 (vgl. SCHNEIDER 1997) können auch eigene Verknüpfungen angelegt und verwaltet werden.

Im vierten Teilschritt schließlich muß die Benutzeroberfläche entworfen und mit entsprechenden Navigationshilfen und Suchwerkzeugen ausgestattet werden.
Bei der Gestaltung der *Benutzeroberfläche* werden gerne räumliche und funktionale Metaphern herangezogen, die die Bedienbarkeit des Systems vereinfachen. Das GRAMMIS-Pilotsystem bietet drei zentrale Zugriffsarten an, die über die Metapher eines ”virtuellen Grammatikinstituts” vermittelt werden:

- Der Zugriff auf grammatische Teilthemen geschieht durch die Auswahl *grammatischer Bücher* im Bücherregal, das sich in der grammatischen Bibliothek befindet. Die interne Struktur der Bücher orientiert sich an der hierarchischen Dokumentenstruktur der gedruckten Grammatik, legt aber keine bestimmte Leseabfolge bei den Teilthemen nahe.
- Ebenfalls in der Bibliothek befinden sich die *lexikalischen Datenbanken*; sie eignen sich für Benutzungssituationen, in denen grammatische Eigenschaften eines bestimmten Wortes gesucht werden, und verfügen über verschiedene Suchoptionen. Die interne Struktur der Datenbanken wird über die Karteikastenmetapher vermittelt.
- Das Büro der *Grammatikexpertin* ist der ideale Einstieg für Benutzer, die sich gezielt über die Bedeutung eines grammatischen Terminus informieren wollen. Die Grammatikexpertin kann verschiedene Terminologien parallel verwalten: Der Benutzer kann sich entweder alle Termini anzeigen lassen, wobei die Zugehörigkeit durch Verwendung farblich gekennzeichnet ist, oder eine bestimmte Terminologie auswählen. Die Termini sind mit den Hypertext-Einheiten der

grammatischen Bücher verknüpft, die weitergehende Informationen zum Terminus enthalten.

Diese über die Benutzermetapher vermittelten Einstiegswege werden durch Zugriffswerkzeuge ergänzt, die an vom Printmedium her Bekanntes anknüpfen und dieses medienspezifisch umsetzen. Hierzu zählen dynamische Inhaltsverzeichnisse, die dem Benutzer den aktuellen Standort in der hierarchischen Kapitelstruktur angeben, Farbleitsysteme und das von der Grammatikexpertin verwaltete elektronische Glossar.

Komponentenspezifische Navigations- und Suchwerkzeuge komplettieren die Möglichkeiten des Benutzers, sich die im System angebotenen Informationen auf individuellen Lesewegen zu erschließen:

- Der im Wortartenbuch verwendete *Browser* erlaubt es, von jeder Seite aus auf den Anfang eines beliebigen Unterkapitels zu springen.
- GRAMMIS-2 bietet eine *Übersichtsgraphik* an, die die aktuell besuchte Hypertext-Einheit durch einen Wegweiser kennzeichnet. Nach dem Prinzip der Fischaugenlinse (*fisheye view*) werden die mit der aktuellen Einheit verbundenen Knoten detailliert angezeigt, während von den übrigen Knoten nur noch die zentralen Verknüpfungspunkte aufgeführt sind.
- Die sog. *Chronik* (engl. *history-list*) bietet die Möglichkeit, alle in der Sitzung bereits besuchten Knoten direkt wieder anzuspringen.
- Die *Volltextsuche* bietet die Möglichkeit, die Hypertextbasis oder Teile davon nach textuellen Suchmustern zu durchsuchen.

Neben diesen Werkzeugen zur freien Navigation führen zusätzlich *verzweigende und sequentielle Pfade* durch die grammatischen Bücher, die sich an der hierarchischen Kapitelstruktur des gedruckten Buches orientieren. Auf diese Weise wurde versucht, gerade ungeübten Hypertext-Nutzern einen Einstieg auf vorgebahnten, linearen Pfaden zu ermöglichen, die sie bei weiterer Vertrautheit mit dem System wieder verlassen können.

4 Fazit

Die einzelnen methodischen Schritte geben Aufschluß über Interdependenzen zwischen den Textstrukturmerkmale des Ausgangstextes und dem bei der Konversion zu betreibenden Aufwand.

a) Bereits in 2.3 wurde gezeigt, daß die Textanalyse in Schritt 1 umso einfacher ist, je stärker die Texte nach einem standardisierten Baumuster aufgebaut sind. Auch die Segmentierung in Textsegmente und deren funktional-semantische Charakterisierung ist umso einfacher, je expliziter die Textstrukturmerkmale an der Textoberfläche sichtbar sind. Bei Texten, deren Struktur von Beginn an mit einer Dokumentenbeschreibungssprache wie SGML ausgezeichnet ist, kann der Analyseschritt 1 gänzlich überflüssig werden, wenn entsprechende funktional-semantische Kategorien in die Textauszeichnung integriert sind.

b) Die Umgestaltung der Textsegmente in Hypertext-Einheiten ist dann relativ aufwendig, wenn die in Schritt 1 gewonnenen Segmente stark von dem geplanten Aufbauschema der Hypertext-Einheiten abweichen. Der Aufwand steigt, wenn die entstehenden Textsegmente viele, über das Segment hinausweisende Kohäsionsmittel enthalten. Hier gilt: je komplexer die im Ausgangstext behandelte Materie, umso höher der Umarbeitungsaufwand. Das einführende Wortartenkapitel – das der GRAMMIS-Komponente zu den Wortarten zugrundeliegt – war also wesentlich einfacher zu bearbeiten als die Komponenten zur Valenz, zur funktionalen Grammatik oder zu den Konnektoren (BREINDL 1998). Schwierig für die Hypertextualisierung waren insbesondere argumentative und diskursive Textpassagen, bei denen die Bauteile der Argumentation über verschiedene Textsegmente hinweg verteilt sind. Um einen optimalen Mehrwert zu erzielen, müßten im Grunde spezifische Verknüpfungsmuster und Zugriffsformen verwendet werden, z. B. eine „timeline" für die wissenschaftshistorische Diskussion eines grammatischen Phänomens (Wortarteneinteilung) oder einer Begriffsgeschichte („Prädikat", „Thema") oder Argumentationsschemata (vgl. STREITZ & HERMANN 1990) zur Darstellung von Kontroversen, z. B. um verschiedene Prinzipien zur Erklärung der Wortstellungsregularitäten im Deutschen.

Es dürfte klar sein, daß die beschriebene Methode relativ aufwendig ist, wenn alle Schritte manuell-intellektuell durchgeführt werden müssen. Hier sollten künftig die vorhandenen Möglichkeiten der Automatisierung besser genutzt werden. Insgesamt hat die Fragebogen-Aktion, die bei den Testnutzern des ersten Prototypen GRAMMIS-1 erhoben wurde, die Erwartung bestätigt, daß sich der bei der Konversion zu betreibende Aufwand lohnt. Es sind dabei gerade die im Buch nicht nachbildbaren interaktiven und multimedialen Eigenschaften und die schnellen und vielfältigen Zugriffsmöglichkeiten, die am meisten geschätzt wurden. Vermißt wurde die Anbindung der im Informationssystem verwendeten Termini an die Terminologie der generativen Grammatik (von Fachkollegen) und an die Termini der Schulgrammatik (vor allem von Seiten der Lehrenden an Gymnasien und im Bereich Deutsch als Fremdsprache). Letztere wünschten sich auch mehr Beispiele sowie eine anschaulichere und einfachere Sprache. Die Perspektiven für den künftigen Ausbau des grammatischen Informationssystems liegen also darin, mehr Multimedia und mehr Interaktionsmöglichkeiten anzubieten und die bislang geschlossene Anwendung zu einem offenen, im WWW recherchierbaren Informationssystem ausbauen, das über externe Verknüpfungen mit grammatischen Beschreibungen verschiedener theoretischer Provenienz vernetzt ist.

Friedrich Wenzel, Thomas Bahn, Luise Regier, Lydia Winschel

Visualisierung russischer fachsprachlicher Grammatik in einem interaktiven multimedialen System

1 „Stratigraphie“ der didaktischen Visualisierungsmittel

Unter Stratigraphie versteht der Geologe eine Einteilung der Schichtenfolgen von Sedimentgesteinen nach den Erdzeitaltern. Dieses Bild auf die Didaktik und die von ihr verwendeten Visualisierungsmittel angewendet, haben wir es zunächst mit den Ablagerungen von einigen Jahrhunderten Kreide zu tun. Und wenn wir durch die Hörsäle streifen, so können wir an einer ganz frischen Schicht von Kreidestaub feststellen, daß in einigen Gebieten die didaktische Kreidezeit noch gar nicht zu Ende ist.[1] In anderen Gebieten hat sich seit den 60-er Jahren unseres höchstpersönlichen Lebenszeitalters, gleichsam im didaktischen Quartär, bereits ein mächtiges Massiv von Folienschichten gebildet. Erst in den jüngsten Jahren des didaktischen Alluviums, mischt sich hier und da etwas anderes ein, das keine Schichten mehr hinterläßt, sondern nur noch elektronische Spuren: das – ich beziehe mich auf den Hörsaal – aus dem Rechner projizierte Bild.

Betrachten wir die Abfolge der Zeitalter der didaktischen Visualisierungsmittel noch einmal unter dem Gesichtspunkt der spezifischen Leistungen des jeweiligen Mediums, so müssen wir feststellen, daß die Folie an Visualisierung zunächst einmal nichts anderes bot, als die Tafel. Und niemand, der seine Folienstifte liebt, bilde sich ein, daß jedenfalls die Farbe neu war, denn die kannte das Tafelbild schon, bevor die Folie aufkam. Die entscheidende Innovation war, daß das mit Mühe angelegte Bild nicht mehr gleich darauf dem Tafelschwamm zum Opfer fiel, sondern wiederverwendbar wurde. Die Verfechter der Folie werden anmahnen in Erwähnung zu bringen, daß die Folie durch Hinterlegung mit farbigen Folienstücken die Möglichkeit der Hervorhebung bietet. Ferner, daß durch das schrittweise Übereinanderlegen von mehreren Folien und Masken eine Abfolge und Entwicklung visualisiert werden könne. Das ist richtig und bedeutete für die Visualisierungstechnik sicherlich einen wichtigen Entwicklungsschritt. Aber wer in der Praxis, außer Referendaren und einigen Enthusia-

[1] Da, wo man die Kreidetafel auch heute noch ganz bewußt einsetzt, wie etwa in der Mathematik, möge man den Autoren die Ironie nachsehen, die in der geologischen Metaphorik liegt. Jedes didaktische Mittel hat da seine Berechtigung, wo es den didaktischen Zwecken entspricht, das gilt natürlich auch für die Tafel.

sten, nimmt schon die Mühe auf sich, seine Folien derart zu präparieren. Von dem Innovativen bleibt in der Praxis die Wiederverwendbarkeit und, daß man seine Folien mit einem Blatt Papier abdecken und das Dargestellte mit dem Gange des Vortrages Stück für Stück freigeben kann. Das machen wir alle.

Der Übergang zum Rechner war zunächst auch nichts anderes als wiederum die Übertragung ein und desselben Bildes von einem Medium in das andere. Die Entwicklung kam gesetzmäßig. Ist man das „Folienpinseln“ leid und mit der Lesbarkeit seiner Handschrift auf glatter Folie nicht mehr zufrieden – ich überspringe den Zwischenschritt, Schreibmaschinentext auf Folie zu xerokopieren –, so liegt der Schritt nahe, die Entwurfsarbeit in den Rechner zu verlagern, zumal, wenn man entdeckt, daß sich Farbgraphik (und damit natürlich auch farbige Hinterlegungen) sehr gut auf Folie drucken lassen. Soweit sind wir immer noch beim statischen Bild, nur mit einem anderen Werkzeug hergestellt.

Daran ändert sich immer noch nichts, wenn wir statt auf die Folie nun auf den Bildschirm des Rechners gehen, das Bild bleibt statisch. Erst das Faktum, daß im Falle des Rechners nicht nur das Bild als Ganzes wiederverwendbar ist, sondern daß Wiederverwendbarkeit für jedes einzelne Bild- und Textelement gilt, schafft den Ansatz für die Entwicklung. Indem im Rechner jedes graphische und textliche Element kopierbar und in Form, Farbe, Größe und dem Bezug auf seinen relativen Ort veränderbar ist, ist der Ansatz zu einer schrittweisen Veränderung des Gesamtbildes gegeben. Der eigentliche technologische Sprung im Sinne der didaktischen Visualisierung kommt dann mit der Automatik dieser Veränderungen durch Programmierung und der Geschwindigkeit der Bildfolge, die der Rechner ermöglicht: In das Bild kommt Bewegung, Dynamik.

An dieser Stelle ist ein Stichwort nicht zu vermeiden, obwohl wir es gern täten, denn es handelt sich um ein Modewort: Multimedia. Befreit man den Gegenstand von den Blähungen der Werbesprache, so kommt man zu etwas ganz einfachem, die Zusammenführung verschiedener Medien. Bereits das Sprachlabor der 60-er Jahre machte die verschiedensten Medien – Schallplatte, Dia, Overheadfolie und später auch das Videoband – an einem Ort verfügbar; allerdings apparativ getrennt. Multimediatechnik ermöglicht zunächst einmal, diese verschiedenen Bild-, Text-, Toninformationen auf einem einzigen Speichermedium zu lagern. Die eigentliche Leistung besteht jedoch in der unmittelbaren Verfügbarkeit all dieser Information (wie es heißt „auf Mausklick“) und zwar nicht nur getrennt nacheinander, sondern auch nebeneinander oder auch miteinander – Bild, Text, Ton aus ursprünglich verschiedenen Quellen nun integriert und zeitgleich. Für den Betrachter passiert nichts Spektakuläres. Doch der, der erlebt hat, wie mühselig es z. B. im Phonetikunterricht ist, ein Tonband hin- und herzuspulen und auf den Anfang einer Textpassage zu bringen, kann die Bedeutung einer Technik ermessen, die es ermöglicht, in einem projizierten Text eine Passage zu markieren und eben diese Passage mit einem einfachen Signal in ihrer Aussprache hörbar zu machen. Einen komplexen Sachverhalt verbal darzustellen kann sehr aufwendig sein. Eine eingespielte Videosequenz, ausgelöst durch Schalten auf einem

Filmsymbol in einem projizierten Wörterbucheintrag, kann diesen Sachverhalt visualisieren und dadurch unter Umständen in wenigen Sekunden anschaulich machen.

Das, was das neue Medium, der Bildschirm des Rechners zur Stratigraphie der didaktischen Visualisierung beiträgt, ist: es bringt Bewegung und Ton ins Bild. Was am Anfang der Entwicklung der didaktischen Visualisierung einmal das statische Tafelbild war, wird nun auf dem Bildschirm oder – nach Projektion – auf der Leinwand lebendig.

Für den didaktischen Effekt von entscheidender Bedeutung ist allerdings noch ein anderes, das sich mit dieser Visualisierungstechnik unter Rechnereinsatz verbindet: die Interaktivität. Der Betrachter kann mit dem Bild kommunizieren. Er kann in das Bild eingreifen. So kann er z. B. ein Textstück an einen anderen Ort bewegen und kann damit eine Aktion auslösen, durch die etwa im Rechner geprüft wird, ob die Zuordnung zu einem graphischen syntaktischen Schema korrekt ist oder nicht. Das Ergebnis wird visuell und akustisch an den Betrachter zurückgegeben.

2 Ziele und Adressaten

Vorgestellt und in seiner Funktionsweise gezeigt werden soll ein System zur Visualisierung russischer fachsprachlicher Grammatik. Daß es sich hier um das Russische und um Fachsprache handelt, trifft den historischen Tatbestand. Das System wäre jedoch auch auf andere Sprachen und Sachbereiche übertragbar. Mit der Entwicklung dieses Systems wird eine didaktische Zielsetzung verfolgt. Es steht im Dienste der Aufgabenstellung, Naturwissenschaftlern und Ingenieuren die Fähigkeiten zu vermitteln, fremdsprachige, im gegebenen Falle russische, Fachtexte zu lesen und möglichst exakt inhaltlich zu verstehen.

Das System wurde entworfen im Hinblick auf die Denkstruktur und die Wahrnehmungs- und Mitteilungsgewohnheiten der Adressaten. Naturwissenschaftler und Ingenieure denken in den Kategorien von Elementen, Funktionen und Strukturen. Wollen sie etwas mitteilen, so greifen sie lieber zum Zeichenstift als zu Worten. Aus Symbolen aufgebaute Formeln, graphische Schemata, Diagramme sind das von ihnen bevorzugte Mittel der Darstellung.

Es liegt nahe, diesen Adressaten nun auch Sprache darzustellen als ein System, in dem Elemente (Wörter, Morpheme) sich zu Klassen ordnen lassen, bestimmte Funktionen tragen und sich in Strukturen (Nominalgruppen, Sätze) anordnen, und dieses dann, soweit es nur geht, zu visualisieren, d. h. unter Zuhilfenahme graphischer Mittel darzustellen.

3 Das technische Instrumentarium

Bei dem vorzustellenden System handelt es sich um ein multimediales interaktives System mit Steuerung des Rechners an der Leinwand mittels Laserstrahl.

Das technische Instrumentarium besteht, abgesehen von einem transportablen multimediafähigen Rechner, aus einem Projektionsbildschirm, Handgeräten zur Er-

zeugung des Laserstrahls und einer optoelektronischen Vorrichtung, mittels der der durch den Laserstrahl erzeugte Bildpunkt dem entsprechenden Punkt auf der Bildschirmmatrix zugeordnet wird. Damit läßt sich nun durch Ein- und Ausschalten des Laserstrahls derselbe Effekt erzielen, wie durch das Klicken mit der Maus. Durch Führen des eingeschalteten Laserstrahls läßt sich ein Objekt – das könnte auch ein Stück Text sein – auf der Leinwand bewegen, wie man es mit der Maus im Falle des Bildschirmes machen würde (Abb. 1).

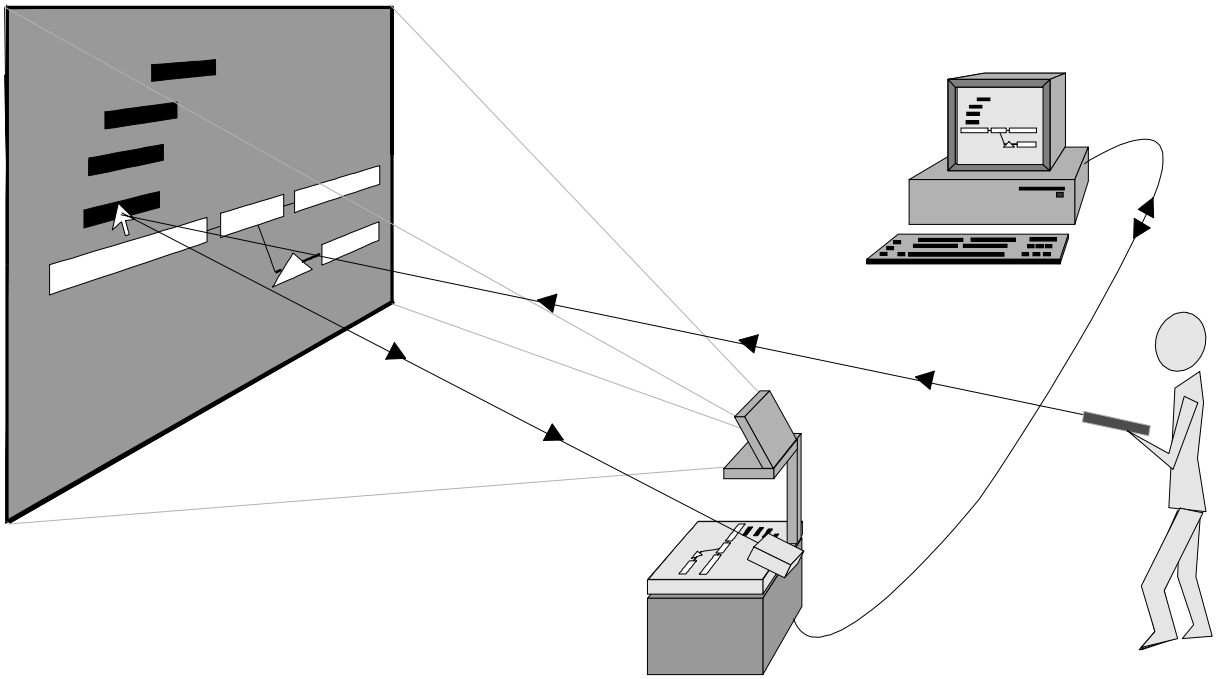


Abb. 1:Technisches Instrumentarium – interaktive Arbeitsweise

Die Software des Visualisierungssystems wird am Fachgebiet Naturwissenschaftliches und Technisches Russisch der Universität Hannover entwickelt, und zwar innerhalb des Programmiersystems HyperCard in der Sprache HyperTalk mit Erweiterungen in C++. Im Hintergrund läuft ein Datenbanksystem (4^{th} Dimension), das für die einzelnen Module des Systems die erforderlichen Texte, grammatischen und lexikalischen Informationen, Ton und Videosequenzen zur Verfügung stellt und entsprechend strukturiert und programmiert wurde.

Die Hardwarekomponenten sind weitgehend am Markt vorhanden. Eine kleine Laserpistole wurde für die Zwecke der Rechnersteuerung aus dem Publikum heraus modifiziert. Das Instrumentarium ist angelegt auf die Arbeit im Hörsaal.

4 Aufbau und Arbeitsweise des Systems

4.1 Grundlegender Aufbau und allgemeine Funktionen

Das System gliedert sich in vier Arbeitsbereiche. Der erste („Syntax“) dient der Untersuchung der syntaktischen Struktur von Sätzen und Nominalgruppen, der zweite („Formbildung“) der Bestimmung von Wortformen im Kontext des Formbildungs- und Flexionssystems der verschiedenen Wortklassen, der dritte („Wortbildung“) dient

zur Untersuchung der Struktur derivierter und komponierter Wörter und der vierte zur lexikalischen/terminologischen/semantischen Bearbeitung von Wörtern und auch größeren terminologischen Einheiten. Einen Überblick über den Aufbau des Systems gibt Abb. 2, in der auch das Zusammenspiel zwischen den einzelnen Arbeitsbereichen und Modulen angedeutet ist.

Das gesamte Verfahren ist modular aufgebaut. So gibt es z. B. innerhalb des Arbeitsbereiches „Syntax“ ein Modul zur Visualisierung der Struktur von Nominalgruppen, oder innerhalb des Arbeitsbereiches „Formenbildung“ ein Modul „Substantiv“, in dem das Flexionsschema der Substantive dargestellt wird, innerhalb dessen der Ort (Numerus, Kasus) eines konkreten Textwortes aufgesucht werden kann. Die einzelnen Module kommunizieren untereinander. Jedes Modul kann aber auch unabhängig von den anderen für sich verwendet werden.

Texteinheiten können auf sehr verschiedene Weise in ein Modul gebracht werden bzw. kommen. Jedes Modul bietet die Möglichkeit der Texteingabe über die Tastatur oder zum Einlesen von Text aus der Datenbank. Texteinheiten können aber auch von einem Modul an ein anderes übergeben werden. So kann z. B. eine bei der syntaktischen Analyse eines Satzes als Satzglied anfallende Nominalgruppe zur Untersuchung ihrer Binnenstruktur in das entsprechende Modul übertragen werden.

Für Darstellung eines gesamten Fachtextes in seiner typographischen Gestaltung und zum Arbeiten mit diesem Text dient das Modul „Text“. Von hier können einzelne Texteinheiten, Sätze, isolierte Nominalgruppen aus Überschriften und Abbildungsbeschriftungen oder einzelne Wörter zur Bearbeitung in die entsprechenden Module übertragen werden.

Der theoretische Analysegang von größeren Einheiten zu immer kleineren – also etwa: Text, Satzgefüge, einfacher Satz, Nominalgruppe, (abgeleitetes) Wort, Morphem – läßt sich mit dem Visualisierungssystem in dieser Abfolge vollziehen. Der modulare Aufbau des Systems gestattet es jedoch, je nach Zielsetzung und Informationsbedarf zwischen einzelnen Modulen hin und her zu springen. Eine markierte Texteinheit wird dabei automatisch in das gewählte Modul übertragen.

Wird Text in die Rahmen oder Felder eines graphischen Schemas gebracht, so wird deren Größe automatisch angepaßt.

Allgemein für das gesamte System ist, daß die von Hand vorgenommene Zuordnung von Elementen zu Einheiten des jeweiligen Strukturschemas oder auch die Abtrennung von Endungen und anderen Morphemen im Wort vom System automatisch auf Richtigkeit geprüft wird.

Allgemein für das gesamte System ist auch, daß derartige Zuordnungen auch automatisch durchgeführt werden können, wobei die einzelnen Analyseschritte vor den Augen des Betrachters gleichsam wie ein Film auf der Leinwand ablaufen.

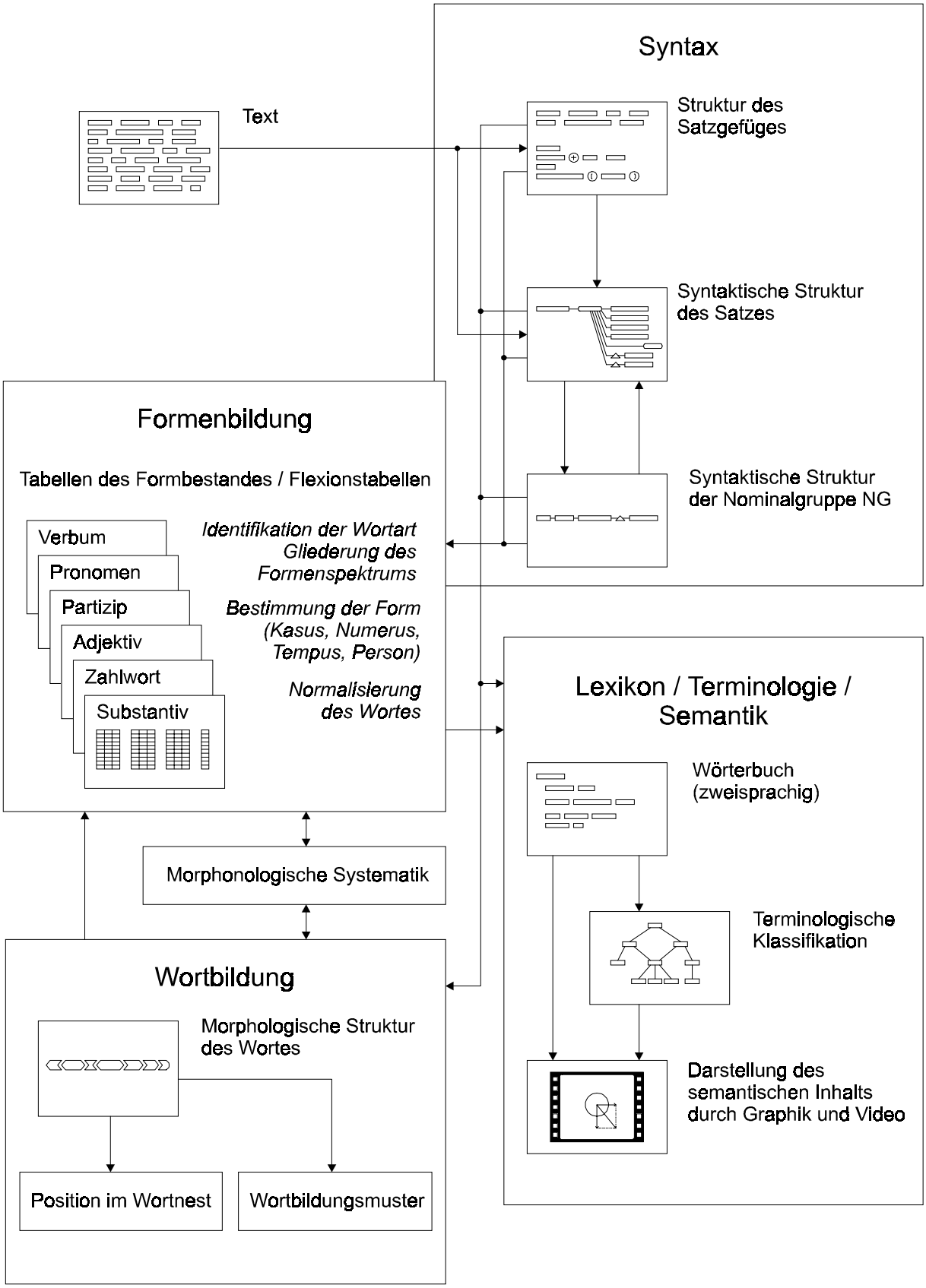


Abb. 2: Visualisierungssystem – schematische Übersicht

4.2 Arbeiten mit Text

Um die Funktionsweise des Systems zu veranschaulichen, gehen wir von einer gängigen Situation im Lehrbetrieb aus, der Arbeit mit einem Text. Der Text „Передача электрической энергии“ [Übermittlung von elektrischer Energie] wird aus der Datenbank in das Modul „Text“ geholt und im Layoutmodus dargestellt, der die wichtigsten Merkmale der typographischen Gestaltung wiedergibt (Abb. 3). (Die Schrift ist aus Gründen der Lesbarkeit in der Projektion durch eine Bildschirmschrift ersetzt. Abbildungen sind, um eine große Darstellung des Textes zu ermöglichen, in die Datenbank ausgelagert) Die aus dem Fachtext selbst stammende Abbildung wird durch Schalten auf einem Bildsymbol, das in der zur Steuerung des Systems dienenden Symbolleiste befindet, in den Text eingeblendet (Abb. 3). Sie gibt einen ersten Einblick in den im Text verhandelten Sachverhalt.

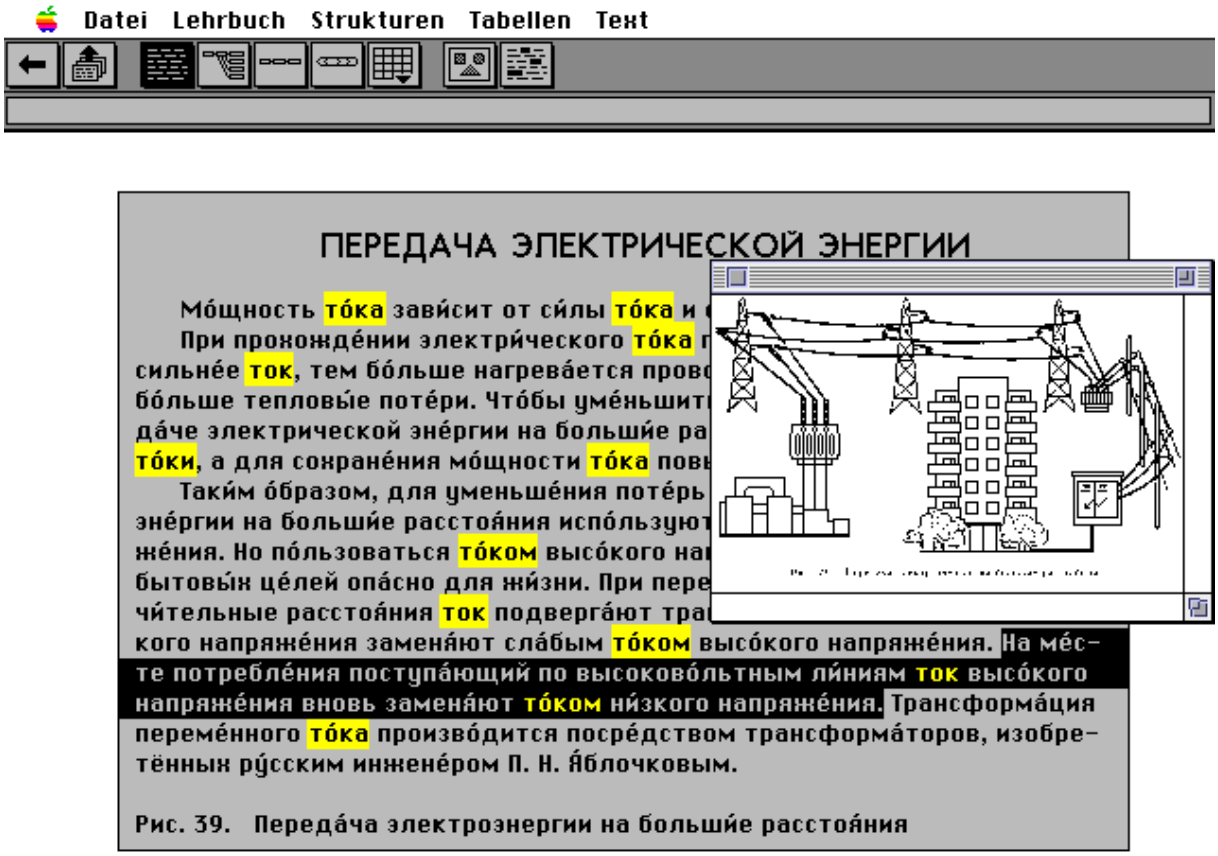


Abb. 3: Arbeiten mit Text

Geht man von dem tragenden Wort des im Titel des Textes enthaltenden Terminus „передача“ [Grundbedeutung: Übergabe] aus, könnte in Zweifel stehen, worum es sich in dem Text eigentlich handelt. Das Markieren dieses Wortes bringt nach Aufrufen des Wörterbuches über das Menü den entsprechenden Wörterbuchartikel auf die Leinwand, der neben morphologischen Angaben und der Grundbedeutung des Wortes kontextabhängige Bedeutungen sehr unterschiedlicher Art umfaßt. Darunter findet sich auch der Kontext des Titels und wir kommen zu der Bedeutung „Übermittlung/Trans-

port von elektrischen Energie“ (Abb. 4). Ist damit immer noch nicht sicher, was darunter zu verstehen ist, so macht das Schalten auf einem Filmsymbol eine Videosequenz sicht- und hörbar. Anhand des Filmes können wir in diesem Fall verfolgen, wie, von dem Hochspannungsverteilsystem eines Umspannwerkes ausgehend, sich eine Ferntrasse über das Land zieht.

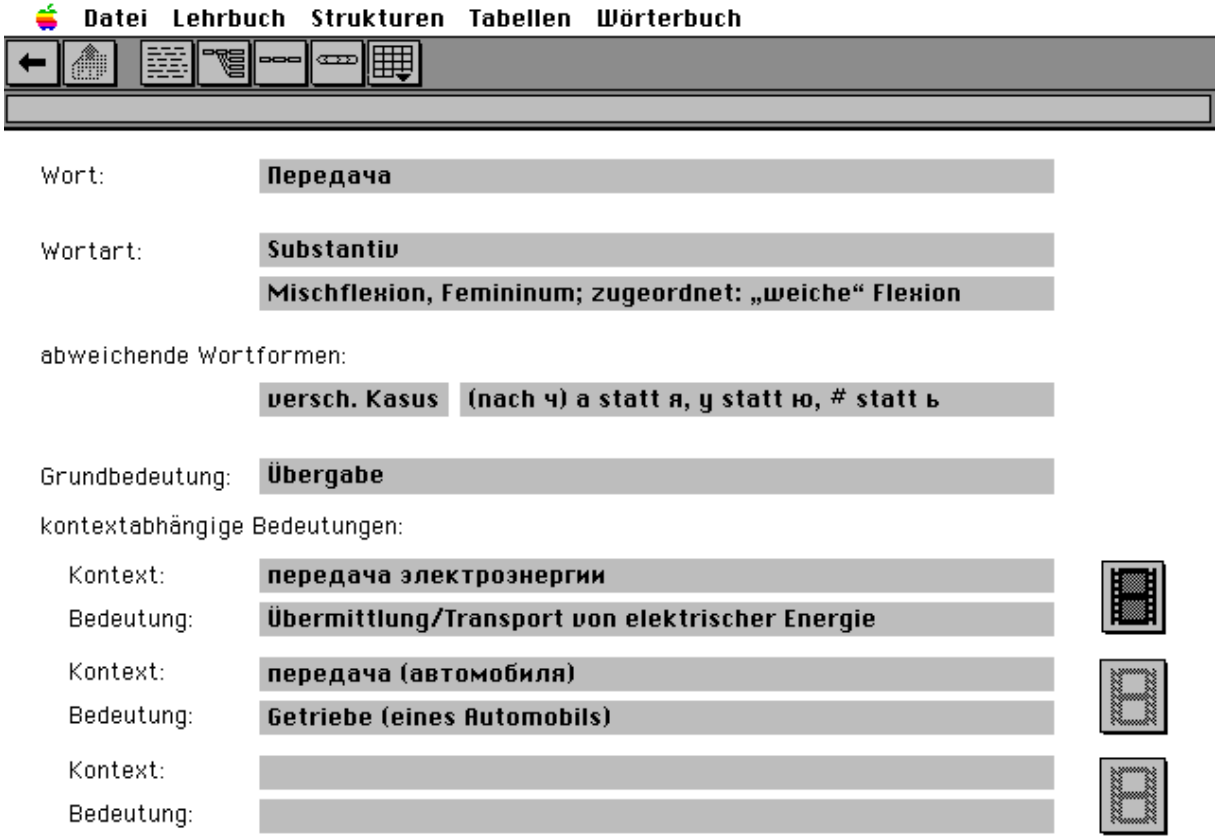


Abb. 4: Wörterbucheintrag

Wenden wir uns nun in dem Text einem Satz zu, der im Hinblick auf Struktur und Wortbestand interessant genug ist, daß sich mit ihm eine größere Zahl weiterer Funktionen des Systems demonstrieren läßt:

> „На месте потребления поступающий по высоковольтным линиям ток высокого напряжения вновь заменяют током низкого напряжения.“
>
> [Textnahe Übersetzung: Am Verbrauchsort tauscht man den über Hochspannungsleitungen eintreffenden Strom hoher Spannung erneut aus [und zwar] durch einen Strom niedriger Spannung. (Im Deutschen würde man fachsprachlich korrekt sagen: man formt ihn um in...; „hoher Spannung“ würde man zur Vermeidung einer Tautologie weglassen.)]

Abb. 3 zeigt den Satz bereits markiert. Schaltet („klickt“) man erneut auf dem markierten Satz, so wird dadurch das Modul zur Analyse der syntaktischen Struktur von

Sätzen aufgerufen und der Satz in ein Textfeld oberhalb des Syntaxschemas übertragen (Abb. 5).[2]

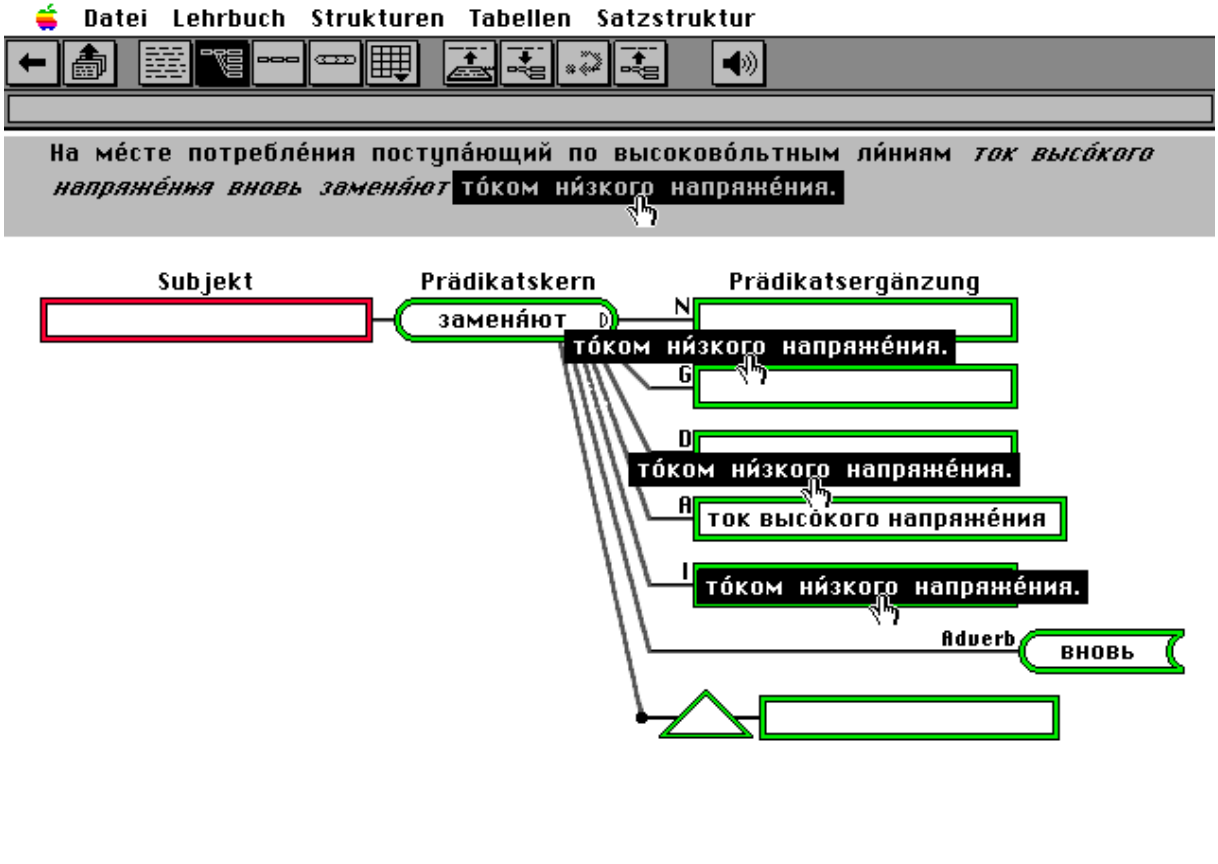


Abb. 5: Strukturschema für den einfachen erweiterten Satz

Schaltet man auf dem Lautsprechersymbol in der Funktionsleiste, so wird der Satz in seiner muttersprachlichen Intonation hörbar.

4.3 Untersuchung der Satzstruktur

Die Aufgabe ist nun, die Wörter des Satzes einzeln oder satzgliedweise aus dem Textfeld herauszubewegen und auf das Strukturschema zu verteilen. Wird das Schema erstmalig eingeführt, so wird das der Lehrende in Form einer Präsentation tun. Im allgemeinen wird er jedoch die Lernenden arbeiten lassen und nur hier und da Hilfestellung geben.

Stellen wir uns also vor, daß ein Student aus dem Publikum heraus mittels des Laserstrahls Wörter einzeln oder in Gruppen in die Rahmen des Strukturschemas hineinführt (Abb. 5 deutet das an). Er könnte versucht sein, das Wort „ток“ [Strom], da es lexikographische Zitierform hat, deshalb nach Nominativ (Sg.) aussieht, in den Subjektrahmen zu bringen. Tut er das, so reagiert das System mit einem schwarzweißen

[2] In der Projektion aus dem Rechner sind alle Schemata farbig angelegt, Farben und Formen von Rahmen und Feldern bilden zusammen ein eindeutiges System zur Kennzeichnung linguistischer Einheiten: Im Schwarzweißdruck gehen unausweichlich Information und Anschaulichkeit verloren.

Flackern des Subjektfeldes und mit einem ebenso unangenehmen Ton – und das Wort springt in seine Ausgangsstellung zurück. Die Erinnerung daran, daß Substantive dieser Art im Nominativ und Akkusativ formgleich sind, führt dazu, das Wort „ток“ in die Akkusativ-, d. h. Objektebene der Ergänzung im Prädikat zu bringen, wo es akzeptiert wird. Die richtige Zuordnung wird zugleich akustisch positiv quittiert.

Geschickter geht vor, wer erst einmal den Prädikatskern identifiziert. Das Verbum ist leicht zu erkennen. Ebenso, daß es sich um eine Pluralform handelt. Die Suche nach einem zugehörigen Nomen im Nominativ Plural bleibt ergebnislos; es folgt also der Schluß, daß es sich um einen unpersönlichen Satz handelt, die Struktur also subjektlos ist. Zur Kennzeichnung wird am Ende, wenn alle Wörter des Satzes der Struktur richtig zugeordnet sind, automatisch das Zeichen „#“ in den Subjektrahmen gesetzt, als Zeichen für ein leeres Glied.

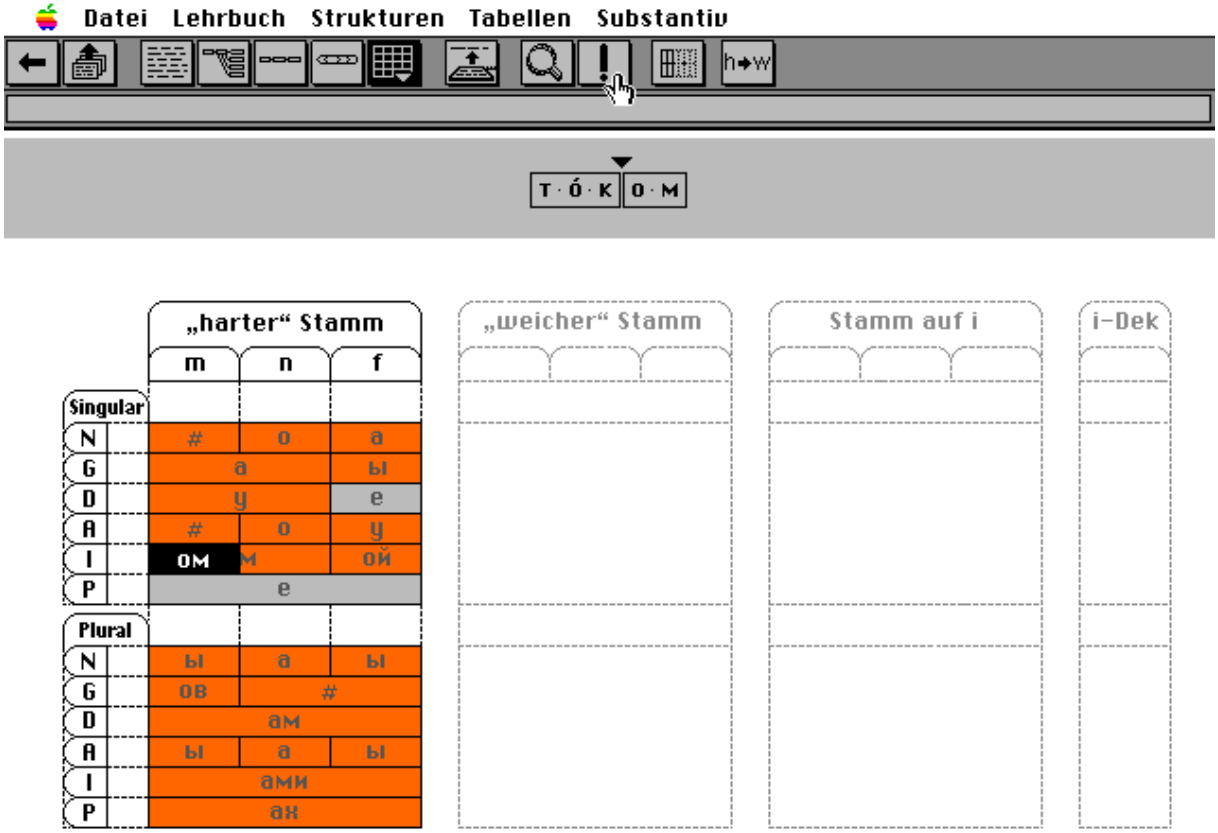


Abb. 6: Flexion des Substantivs – Kontexgebundene Bestimmung von Kasus, Numerus, Genus

Die Zuordnung des Wortes „током“ könnte Schwierigkeiten machen, da der Instrumental, um den es sich hier handelt, dem Deutschen zunächst fremd ist, und es einige Zeit braucht, bis sich der Lernende in seine Verwendung und Funktion eingewöhnt. Wird der Kasus nicht sicher erkannt, so wird das Wort markiert und durch Schalten in der Symbolleiste an die Flexionstabelle der Substantive übergeben (Abb. 6). Initialisiert man den Suchauftrag, so trennt das System die Endung ab und markiert in der Flexionstabelle durch Invertierung im schwarzen Feld die Stelle, an der die Endung

steht. Damit sind zugleich die morphologischen Bestimmungsgrößen Kasus, Numerus, Genus gegeben. In unserem Falle handelt es sich also um einen Instrumental.

Nach Rücksprung in die Satzsyntax kann das Wort „током“ bzw. auch gleich die auf dieses Wort aufbauende Nominalgruppe („током низкого напряжения“ [Strom niedriger Spannung]) in die Instrumentalebene der Prädikatsergänzung gebracht werden. Schalten auf dem Symbol „I“ (für Instrumental) neben dem Rahmen öffnet ein Menü, in dem die verschiedenen Funktionen des Instrumentals im Russischen angezeigt werden. Markiert man hier „Modal (wie?)“ oder auch „Instrument/Mittel“, so quittiert das System dieses als zutreffend und übernimmt diese Funktionsbezeichnung in das Strukturschema (Abb. 7).

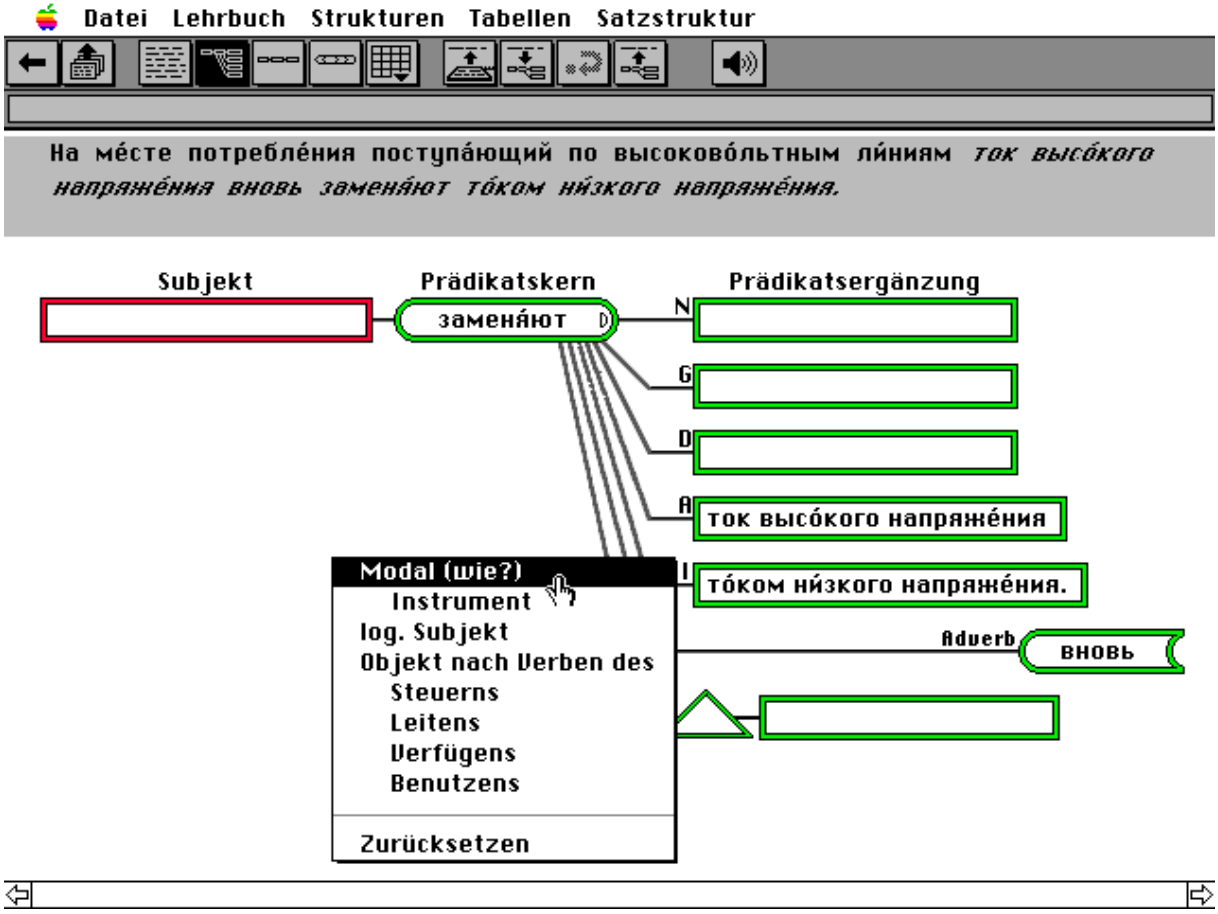


Abb. 7: Satzanalyse – Bestimmung der Funktion des instrumentalen Gliedes

4.4 Arbeiten mit Flexionstabellen

Noch einmal zurück zur Formbestimmung des Wortes „током“. Das System bietet zwei Arten an, im Flexionssystem zu suchen:

(1) Aufsuchen einer Endung unter Berücksichtigung der Flexionsklasse des jeweiligen Wortes und der morphologischen Eigenschaften (Numerus und Kasus, beim Adjektiv/Partizip auch Genus), die im gegebenen Kontext gefordert sind. In diesem Falle wird in der Flexionstabelle der betrachteten Wortklasse nur eine einzige En-

dung angezeigt und damit auch die vom Kontext geforderten morphologischen Parameter.

(2) Aufsuchen der Endung, wo auch immer sie im Flexionssystem der betrachteten Wortklasse vorkommt. D. h. tritt die gleiche Endung mehrfach auf, werden alle Auftretensfälle angezeigt.

Für orientierendes und lernendes Arbeiten ist das zweite Verfahren zu bevorzugen.

Da dem Wort „ток“ [Strom] für den vorliegenden Text eine besondere Bedeutung zukommt, wollen wir ihm weiter nachgehen. Gehen wir noch einmal in den Text zurück und lassen uns über die Funktion „Markierung“ alle Wörter anzeigen, die Formen des Wortes „ток“ sind (Abb. 3 zeigt den Text bereits in der markierten Form). Das Wort taucht in dem Text 14 mal in verschiedenen Formen auf. Darunter auch in der Form „токи“. In der Flexionsklasse, in der wir zuvor den Instrumental „током“ nachgeschlagen haben, tritt die Endung „и“ aber gar nicht auf. Um den Fall zu klären, markieren wir im Text das Wort „токи“ und gehen noch mal in die Flexion der Substantive, wohin wir das Wort „токи“ übertragen vorfinden. Schalten wir auf dem Symbol für die Funktion „Endung suchen“, dann werden in der Tabelle in invertierter Form auf schwarzem Untergrund alle Fälle angezeigt, in denen die Endung „и“ auftritt. An den Stellen jedoch, an denen normalerweise nicht die Endung „и“, sondern die Endung „ы“ auftritt, alternieren jetzt im ständigen Wechsel die Endung „ы“ und die Endung „и“. Unter der Flexionstabelle erscheint eine morphonologische Regel, die besagt, daß bei Stämmen auf „г, к, х“ in der nachfolgenden Endung „ы“ durch „и“ ersetzt wird (Abb. 8). Das System prüft bei Aufruf der Suchfunktion automatisch, ob eine derartige Stamm-Endungskombination vorliegt und steuert die Ausgabe auf den Bildschirm entsprechend.

Durch einige wenige morphonologische Regeln reduziert sich die Vielzahl der Flexionsklassen des Substantivs, die gewöhnlich in Grammatiken aufgeführt werden, erheblich. Zu dem morphonologischen Regelapparat, dessen lernökonomische Wirkung einem Naturwissenschaftler oder Ingenieur durchaus verständlich ist, gehört auch eine Vokaltabelle („Endungsvarianten“, siehe Abb. 8). Bei der Präsentation des Flexionssystem der Substantive läßt sich diese Tabelle in den Vordergrund holen und mit ihrer Hilfe demonstrieren, daß nur der orange hinterlegte Teil der Flexionstabelle (unter „harter“ Stamm) gelernt zu werden braucht. Der übrige (blaue) Teil der Flexionsendungen ergibt sich bis auf wenige Ausnahmen, die farbig anders hinterlegt sind, aus der Vokaltabelle.

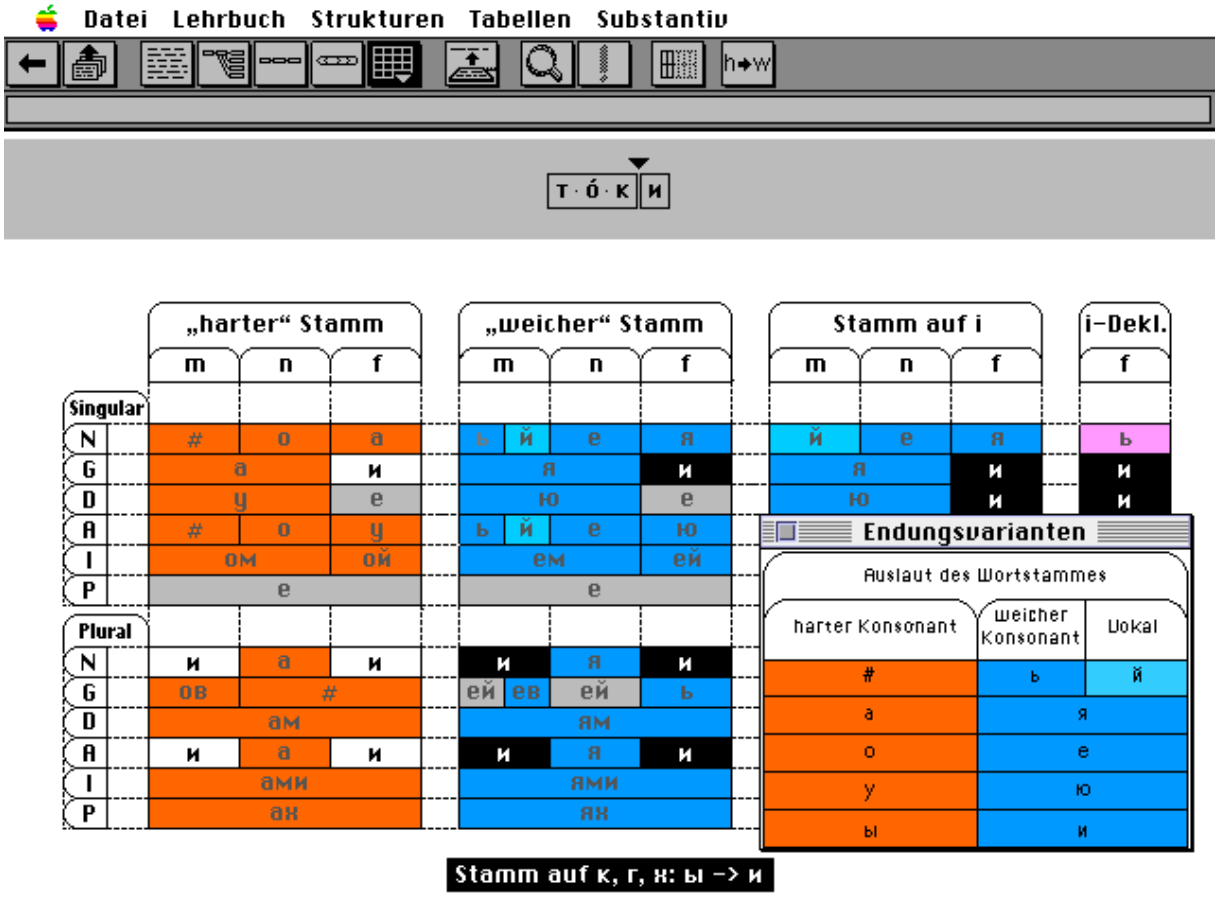


Abb. 8: Aufsuchen einer Endung im Flexionssystem des Substantivs – Berücksichtigung morphologischer Gegebenheiten

4.5 Untersuchung der Nominalgruppenstruktur

In den Fachsprachen von Naturwissenschaft und Technik liegen die Schwierigkeiten oftmals nicht in der Syntax des Satzes. Schwierigkeiten bereiten vielmehr die voluminösen und häufig auch strukturell komplizierten Nominalgruppen. Als Beispiel dafür kann man die in dem noch nicht zu Ende bearbeiteten Satz enthaltene erste Nominalgruppe nehmen. Die Schwierigkeit besteht bereits darin, den Umfang dieser Nominalgruppe richtig zu bestimmen. Man könnte, geleitet von der Struktur der Nominalgruppe im Deutschen, annehmen, daß die initiale präpositionale Gruppe zum Bestand der Nominalgruppe gehören. Also fälschlich: „На месте потребления поступающий…ток…“ − entsprechend dem Deutschen: „Der am Verbrauchsort eintreffende Strom...“. Markiert man nun die Nominalgruppe unter Einbezug der initialen präpositionale Gruppe und versucht sie in den Strukturschema für Nominalgruppen zu übertragen, so wird dieses zurückgewiesen mit der Fehlermeldung: „Entspricht nicht der Struktur von Nominalgruppen“. Damit bleibt für die präpositionale Gruppe nur noch der Ort der modalen Ergänzungen im Prädikat (Gruppierung aus dreieckigem und rechteckigem Rahmen unten im Strukturschema des Satzes, Abb. 5). Der Rest des Satzes läßt sich problemlos in das Textfeld des Strukturschemas für Nominalgruppen übertragen, wird also von dem System als Nominalgruppe akzeptiert.

Das Strukturschema zeigt zunächst einmal nur den prinzipiellen Aufbau einer Nominalgruppe, indem links und rechts von ihrem Kern in blasser Farbstellung die am häufigsten hier auftretenden Attribute angegeben werden. Man muß das Schema noch konkretisieren. „Klickt“ man auf einen der Verbindungsknoten, öffnet man damit ein Menü, in dem die möglichen Attribute angegeben werden. Auf der linken Seite stehen dazu Adjektiv und Partizip zur Auswahl. Die Entscheidung ist von Bedeutung, da sich die beiden Wortarten unterschiedlich erweitern lassen. Ist man sich nicht sicher, ob es sich bei dem Wort „поступающий“ um ein Partizip handelt, und wenn ja, um welches, so übergibt man das Wort an das Formbildungsschema für Partizipien. Überläßt man dem System das Aufsuchen der Form, so werden zunächst die Endung und das den Partizipialstamm bildende Suffix abgetrennt und daraufhin in dem Schema die zutreffende Form (Partizip Präsens aktiv) angezeigt (Abb. 9). Schaltet man nun auf dem automatisch markierten stammbildenden Suffix, so öffnet man damit ein Fenster, in dem aus der Datenbank weitere Beispiele für dieses Partizip eingelesen werden. Schaltet man auf der Flexionsendung, so öffnet man damit ein Fenster mit der verkleinerten Flexionstabelle für Adjektive/Partizipien. Dort werden die von den Adjektivendungen abweichenden Partizipialendungen im Sekundenwechsel angezeigt (Abb. 9).

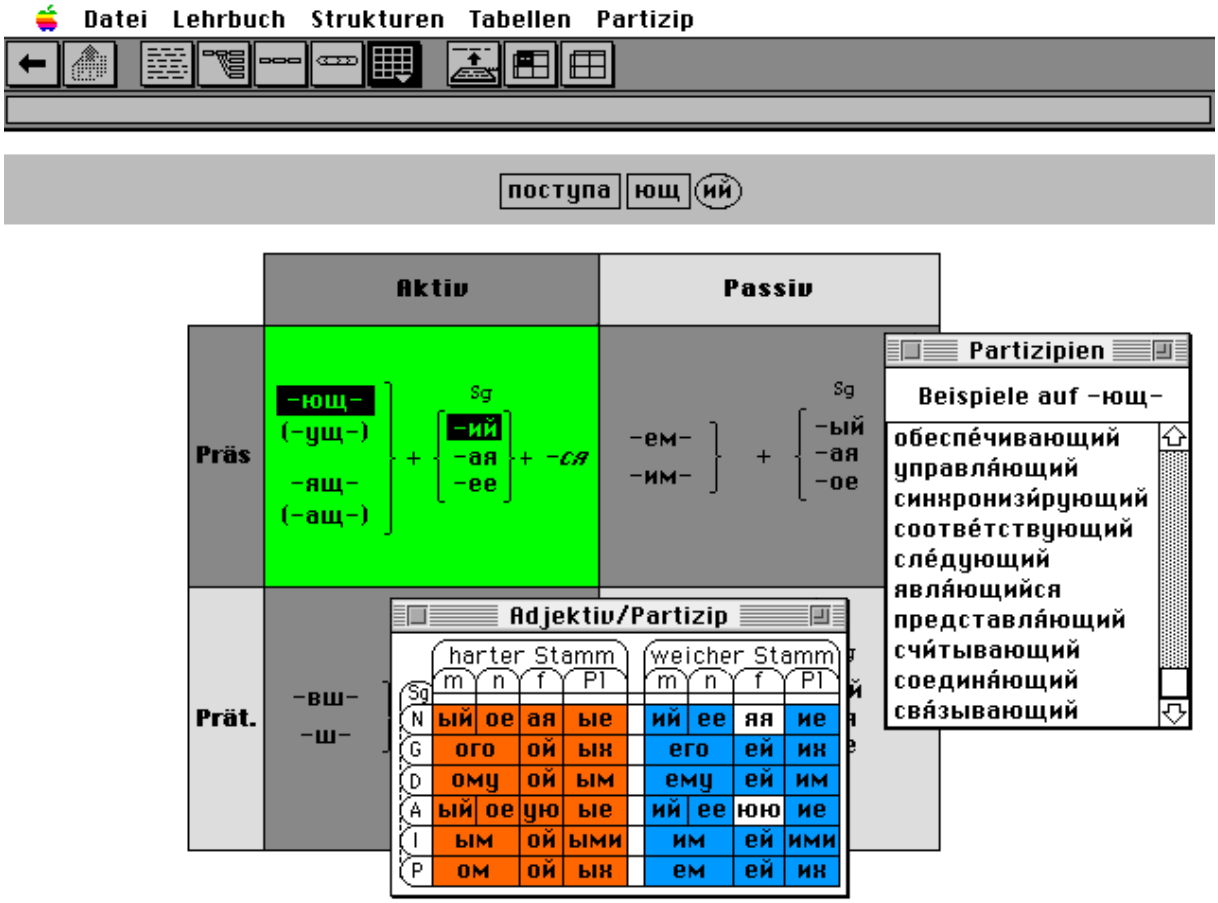


Abb. 9: Bestimmung der Form eines Partizips

Im Strukturschema für Nominalgruppen definiert man nun den linken Attributsrahmen und bewegt das Wort „поступающий“ hinein – wohl bemerkt immer in dem Bewußtsein, daß eine unzutreffende Zuordnung vom System moniert würde. Schaltet

man nun auf dem rechts aus dem Partiziprahmen austretenden Pfeil, so werden in einem sich dadurch öffnenden Menüfenster die möglichen, nach rechts anschließenden Erweiterungen des Attributs angezeigt (Abb. 10). Darunter findet man auch die präpositionale Gruppe, die im Russischen im Gegensatz zum Deutschen grundsätzlich rechts vom Attribut anschließt (bei der Übersetzung ins Deutsche muß also immer umgestellt werden). Durch Schalten auf dem Menüpunkt „Präpositionale Gruppe“ wird ein entsprechender Rahmen in das Strukturschema eingefügt. So fährt man fort, bis alle Glieder der Nominalgruppe erfolgreich zugeordnet sind, was vom System mit einem akustischen Abschlußsignal positiv quittiert wird.

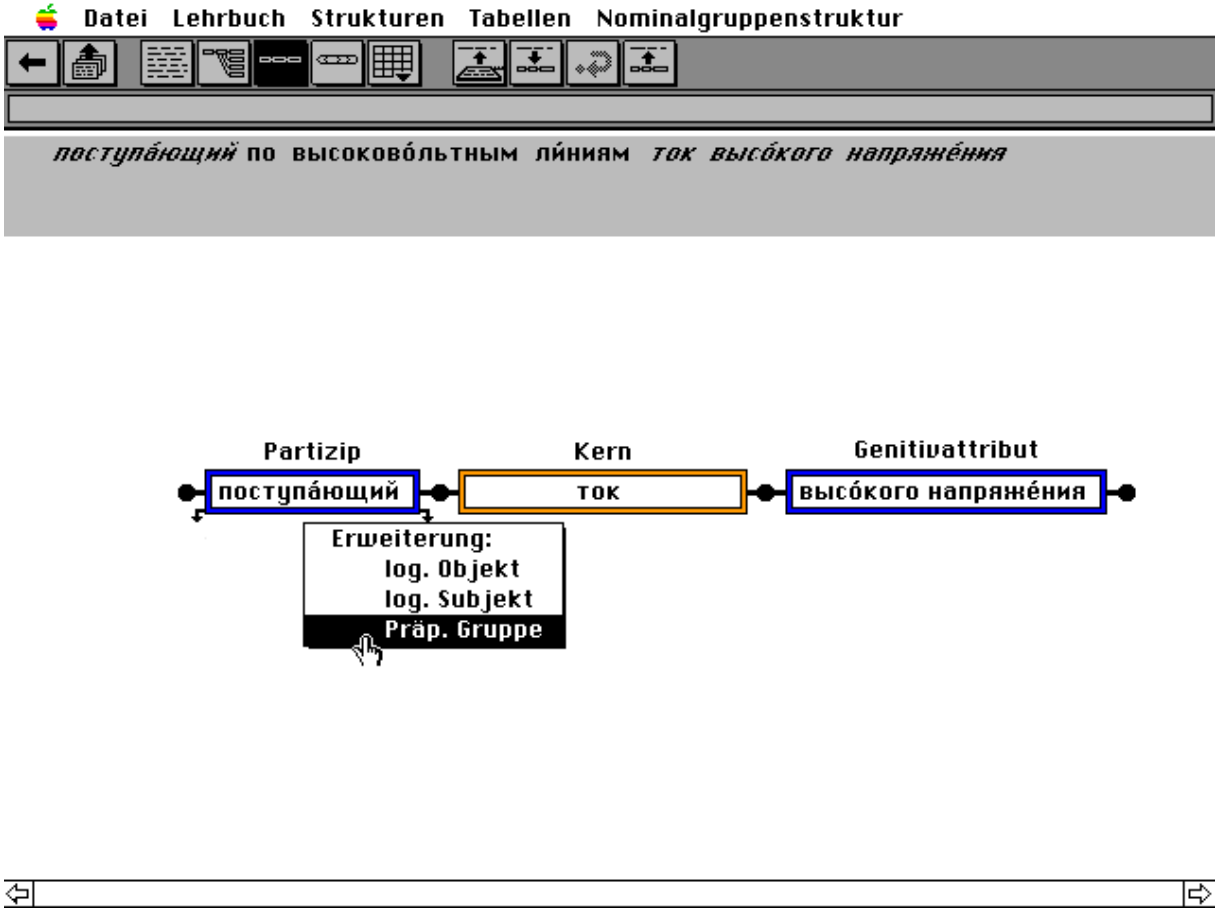


Abb. 10: Syntaktische Struktur der Nominalgruppe

4.6 Wortbildungsanalyse und Morphonologie

Die zuvor zugeordnete präpositionale Gruppe „по высоковольтным линиям“ [über/durch Hochspannungsleitung] enthält als mittleres ein Wort, das in Bezug auf die Wortbildung Aufmerksamkeit verdient. Wir übertragen es auf die übliche Weise in das zuständige Schema, jetzt nun in das, in dem die Wortbildungsstruktur untersucht wird. Die Aufgabe besteht nun zuerst einmal darin, das Wort in Wortbausteine (Morpheme) zu zerlegen, was durch versetzen einer keilförmigen Marke unter dem Wort geschieht. Bevor die Wortbausteine in das Strukturschema hineinbewegt werden können, muß dieses im Hinblick auf den konkreten Fall angepaßt werden. Initialisiert man

den Vorgang, so muß zunächst die Abfrage beantwortet werden, wieviele Wortkerne das Wort enthält. Schaltet man auf dem Wert „2“, was auf das zu behandelnde Wort zutrifft, so erscheint das Schema, das die allgemeine Struktur zweikerniger Wörter aufzeigt. Hier werden nun die Glieder des segmentierten Wortes zugeordnet (Abb. 11). Die Kontrolle über die richtige Zuordnung übernimmt, wie immer, das System. Im gegebenen Falle überflüssige Präfixfelder können zuvor von Hand durch Schalten in der Spitze entfernt werden, oder aber sie werden mit Abschluß der Zuordnung vom System automatisch entfernt. (Durch Schalten in der Spitze von Feldern öffnet sich zunächst ein Menü, in dem man die Operation „Löschen“ oder „Ergänzen“ von Feldern – z. B. zum Aufbau von Präfixketten – auslöst.) Der erfolgreiche Abschluß der Analyse wird wie immer akustisch quittiert.

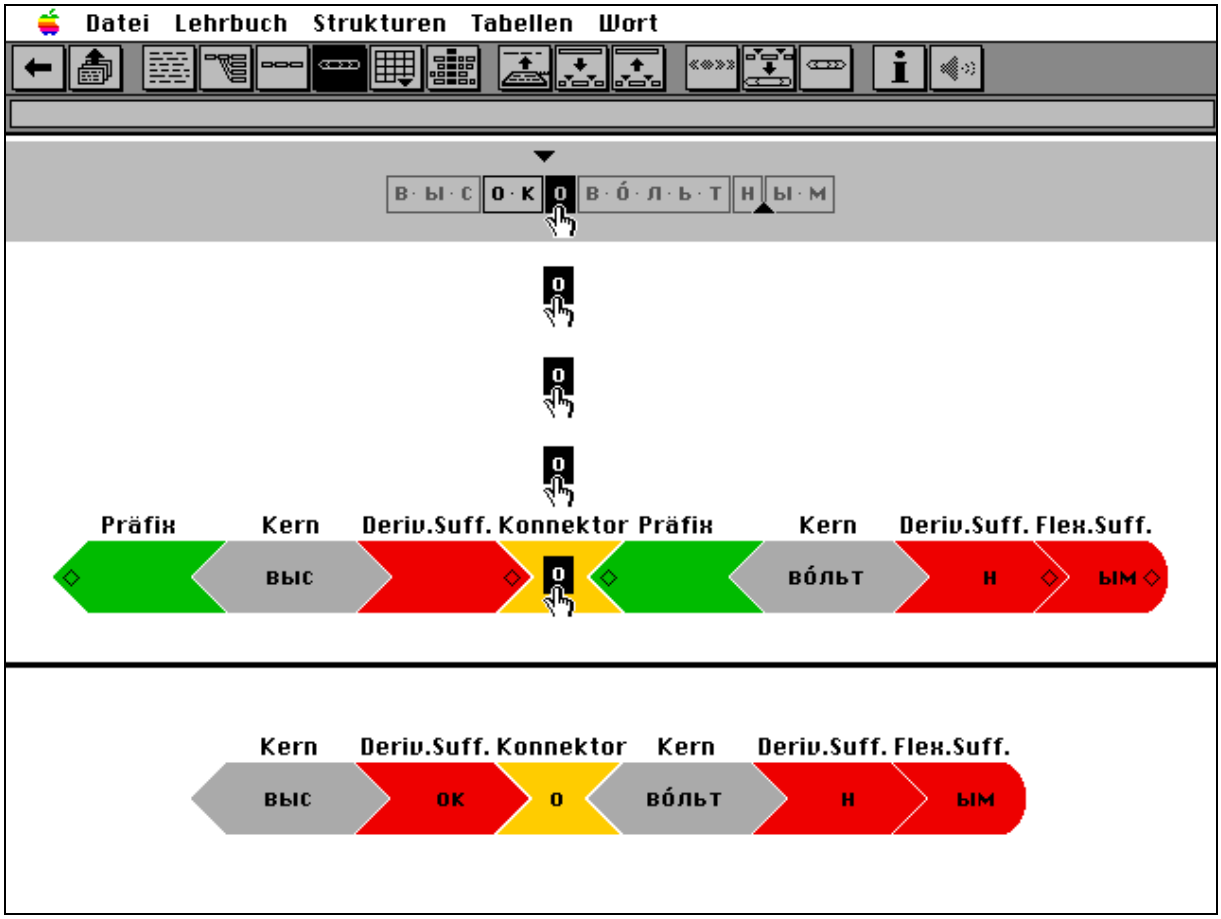


Abb. 11: Untersuchung der Wortstruktur – mehrkerniges Wort

Die Leistungsfähigkeit des Moduls „Wortbildungsanalyse“ soll noch an einem anderen Beispiel aus dem Text gezeigt werden. Die Analyse des Wortes „проводник“ [(elektrischer) Leiter] ist trivial. Erwähnung verdient lediglich, daß nach Zuordnung der drei Morpheme das System zur Komplettierung der Struktur in das Feld für das Flektionssuffix automatisch die Nullendung (für maskuline Substantive „harter“ Flexion) setzt (Abb. 12 links). Das Wort „проводник“ eignet sich jedoch besser als das vorausgehende, um weitere Funktionen im Arbeitsbereich „Wortbildungsanalyse“ zu demonstrieren. Durch Schalten auf der entsprechenden graphischen Symbol holt man das Wortnest aus der Datenbank, dem das gerade in Arbeit befindliche Wort angehört. Das

Wort „проводник“ finden wir hier markiert vor (Abb. 12 rechts). Das Wortnest zeigt, und das macht es besonders interessant, eine ausgeprägte Allomorphie des Kernmorphems. Der Lernende kommt so und so nicht darum herum, sich mit derartigen Allomorphien auseinanderzusetzen. Zur weiteren Aufklärung schaffen wir uns durch Schalten auf dem Symbol „Info“ ein Fenster in die Datenbank, durch das uns die morphonologischen Verhältnisse dargestellt werden. Nachdem wir auf die Abfrage hin, ob wir uns über Präfixe, Suffixe oder Kernmorpheme informieren wollen, das letztere bejaht haben, können wir uns nun durch die Liste von Kernmorphemen durchscrollen, wo wir auch den zuvor durch Segmentierung gefundenen Wortkern „-вод-“ wiederfinden. Markieren wir ihn, so werden uns seine Allomorphe angezeigt. Markieren wir nur in der Liste der Allomorphe den Wortkern „-вес-“, so wird uns nach Schalten auf „Vokal“ zu dem Vokalwechsel „о“ nach „е“ die Regel angegeben, daß es sich hier um einen Ablaut handelt. Nach Umschalten auf „Konsonant“ wird uns die Regel angegeben, nach der der Konsonantenwechsel von „д“ nach „с“ vor sich geht (siehe Abb. 13). In einem weiteren Fenster werden Wörter aufgeführt, die Beispiele für diesen Konsonantenwechsel sind.

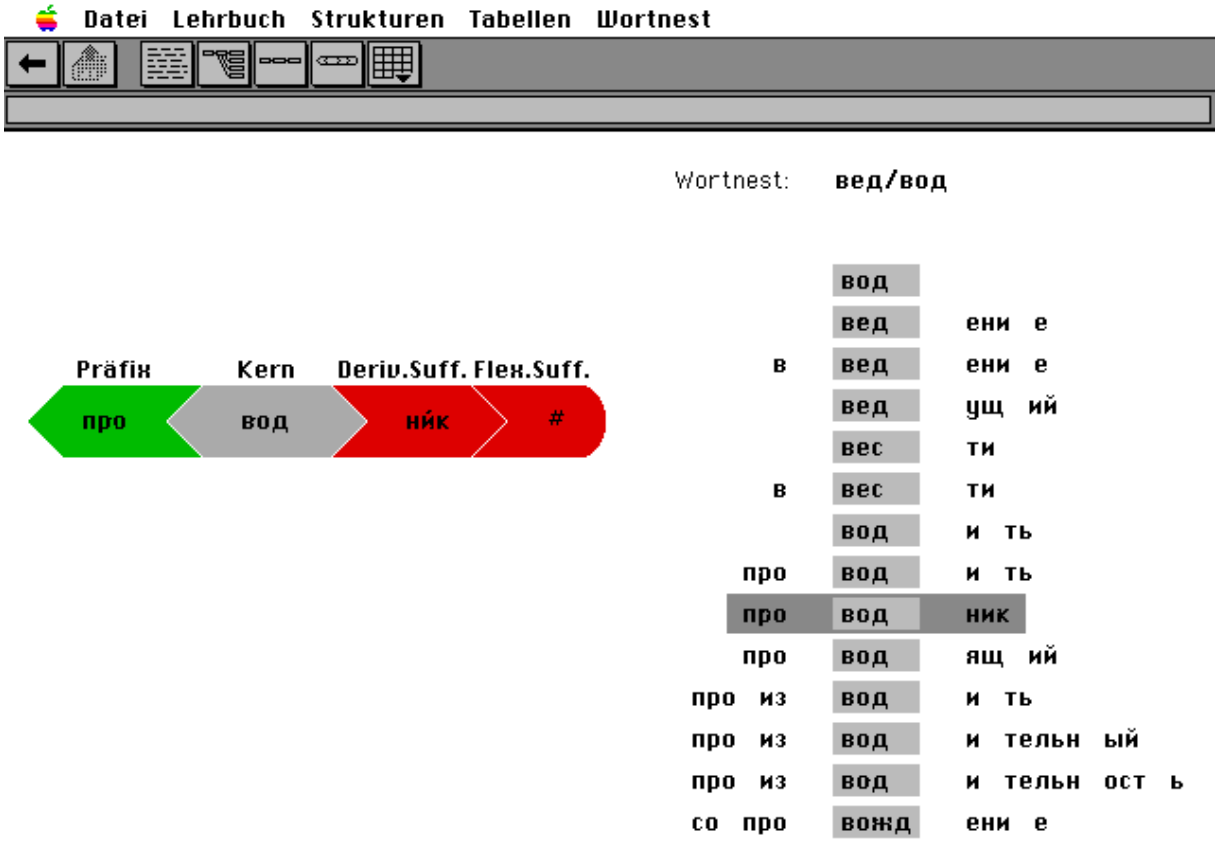


Abb. 12: Wortstruktur und Wortnest

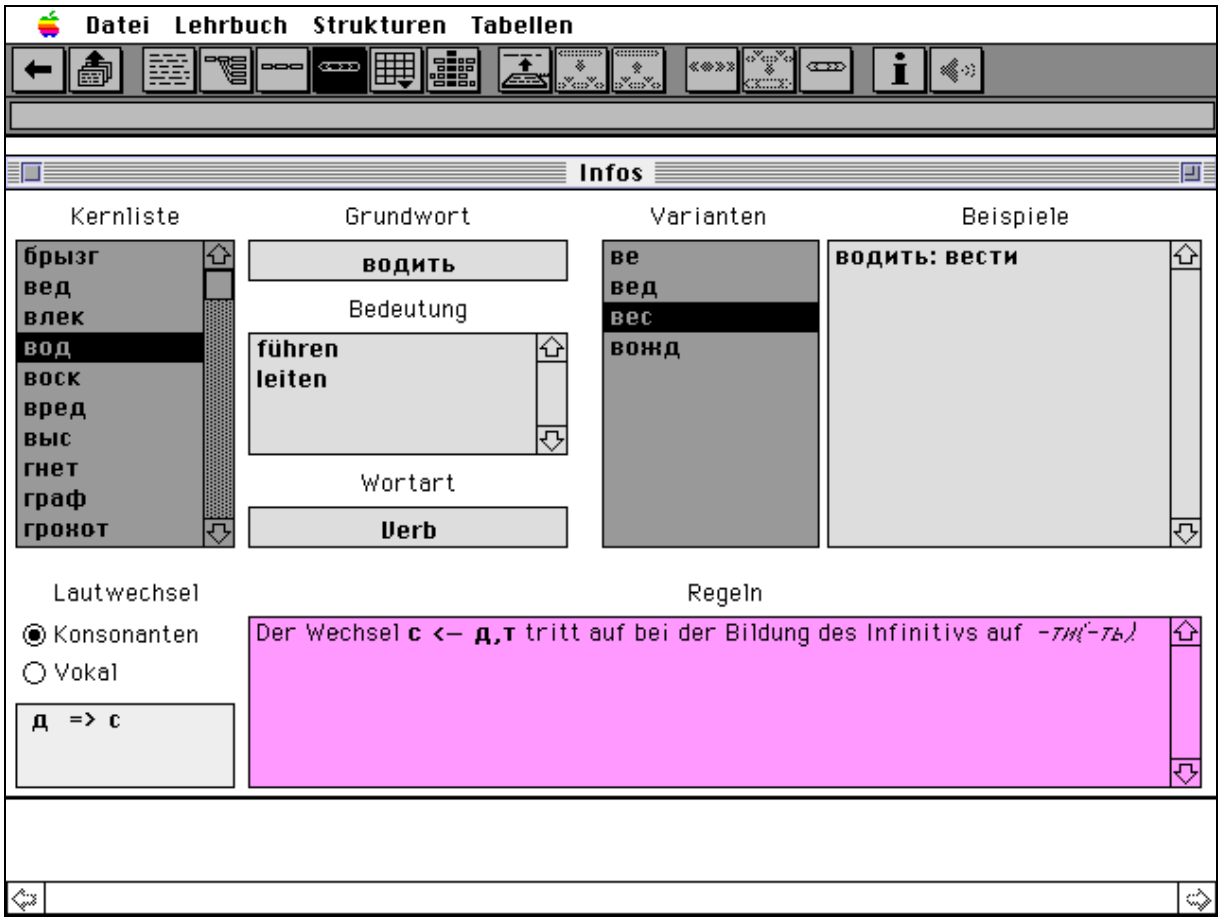


Abb. 13: Morphemklassen – Allomorphie („Varianten“) und Regeln für den Lautwechsel

5 Drei verschiedene didaktisch-methodische Nutzungsmöglichkeiten

Das System ermöglicht – und intendiert das auch – drei verschiedene Arbeitsweisen:

- Präsentation: Der Lehrende arbeitet unmittelbar an der Leinwand – er führt dem Publikum gleichsam vor, wie Sprache funktioniert.
- Interaktion: Der Lehrende stellt Fragen, die aus dem Publikum heraus beantwortet werden – indem an der Leinwand für alle sichtbar mit Hilfe der Laserpistole z. B. die Glieder eines Satzes in ein Strukturschema bewegt werden. Völlig gleichberechtigt und nur statistisch (leider) nachgeordnet ist die Vertauschung der Frage- und Antwortrollen: Aus dem Publikum werden Fragen gestellt, die der Lehrende durch Aktion an der Leinwand beantwortet.
- Experimentieren und Üben: Hierbei arbeitet eine Studentengruppe selbständig, das System übernimmt die notwendigen Kontrollen.

Die Anlage des Systems stellt in Rechnung, daß es nicht nur zur didaktischen Arbeit mit einem Text, also etwa zur didaktischen Begleitung einer Übersetzung des Textes, eingesetzt wird, sondern auch zu einer Orientierung innerhalb des grammatischen Sy-

stems, um (sich) einen Überblick zu vermitteln, oder auch zum Nachschlagen und zum Lernen grammatischer Details.

6 Anmerkungen zu Geschichte und Status des Projektes

Das vorgestellte Visualisierungssystem entwickelte sich aus einem langjährig geführten Kurs „Russisch für Naturwissenschaftler und Ingenieure“ heraus, bei dem es um die Fertigkeiten geht, die man braucht, um Fachtexte lesen zu können. An einem der letzten Durchgänge durch das Curriculum nahm auch eine kleine Gruppe von Informatikern teil, die Technisches Russisch als Anwendungsfach im Diplomstudiengang Mathematik studieren. Sie konnten bei ihren Bemühungen um den Erwerb der russischen Fachsprache gleichsam am eigenen Leibe die Problematik der eingangs skizzierten didaktischen Visualisierungsmittel spüren. Die eigenen Erfahrungen mit dem didaktischen Prozeß konnten auf diese Weise in die Programmierung einfließen. Die Ergebnisse der aus dem Lehrbetrieb heraus gewachsenen Entwicklung fließen längst schon wieder in den Lehrbetrieb zurück, wo sie bereits Früchte tragen, aber auch Gegenstand der Erprobung sind.

Das System befindet sich also noch in der Entwicklung. Im Arbeitsbereich „Syntax“ sind die Module, die den einfachen erweiterten Satz und die Nominalgruppen behandeln, realisiert; im Arbeitsbereich „Formenbildung“ die Module für Substantiv, Zahlwort, Adjektiv und Partizip. Der Arbeitsbereich „Wortbildung“ ist weitgehend realisiert; hier fehlt nur noch das Modul, das die Wortbildungsmuster darstellt. Der Arbeitsbereich „Lexikon/Terminologie“ existiert erst in Form von arbeitsfähigen Modellen. Die realisierten Module und Modelle sind zusammen mit dem Datenhintergrund zu einem unter den Bedingungen des praktischen Lehrbetriebs arbeitsfähigen Gesamtsystem zusammengefügt, das bereits einige Stufen der praktischen Erprobung durchlaufen hat. In Arbeit sind außer den wichtigsten ausstehenden Module, die Satzgefüge, Wortbildungsmuster und die Formenbildung in den Kategorien Verben und Pronomen anbetreffen, sowie eine Benutzeroberfläche für die Datenbank, die es dem Lehrenden ermöglicht, ohne intimere Kenntnisse des Datenbanksystems bei einem Einsatz neuer Texte die notwendige Datenaufbereitung vorzunehmen.

Gregor Thurmair, GMS München

Ein multifunktionales Lexikon

1. *Kontext: Das Projekt AVENTINUS*
2. *Anforderungen an eine lexikalische Datenbank*
3. *Organisation des lexikalischen Materials*
4. *Implementierung*

1 Kontext: Das Projekt AVENTINUS

Das folgende Lexikonkonzept ist im Rahmen eines Forschungsprojektes der EU zu sehen. Das LE-Projekt AVENTINUS (*Advanced Information System for Multinational Drug Enforcement*, THURMAIR 1997a) bezweckt, verschiedene Verfahren der Sprachtechnologie in einen gemeinsamen Rahmen zu stellen, der in der Applikation Drogenbekämpfung gegeben ist. Schwerpunkt ist einerseits die Erschließung neu eintreffenden Materials (im internationalen Drogenkontext meistens in nicht beherrschten Fremdsprachen), andererseits die Analyse und das Finden von Informationen zu einem gegebenen Fall bzw. Problem. Das Projekt soll diese beiden Szenarien unterstützen und folgende Komponenten beisteuern:

- Übersetzungskomponenten (einfache Termsubstitution, Translation Memories und volle maschinelle Übersetzung, je nach Sprache)
- informationsverarbeitende Komponenten (Faktenextraktion aus Texten, multilinguales Indexing)
- Suchkomponenten (Suche in Texten und Fakten, Fuzzy-Suche im Bereich von Namen)

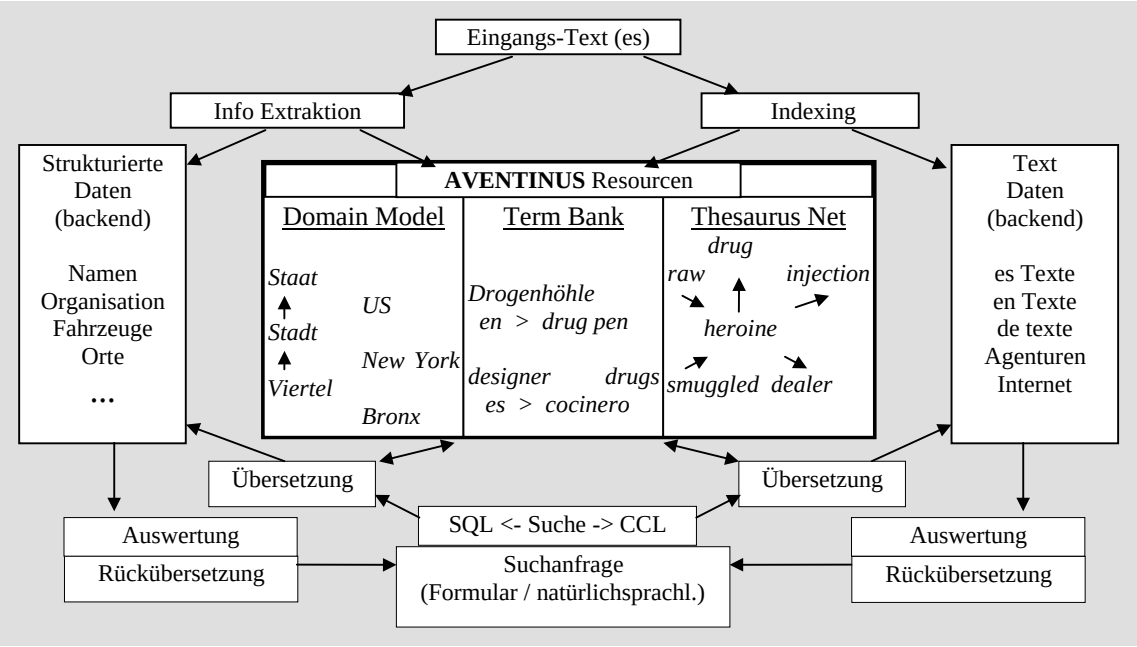


Abb. 1: AVENTINUS und seine linguistischen Ressourcen

All diese Komponenten sollen so modular gestaltet sein, daß sie in die DV-Umgebung der Anwender (Polizeiorganisationen aus verschiedenen Ländern) eingebettet werden können.

2 Anforderungen an eine lexikalische Datenbank

Die meisten der beteiligten Komponenten greifen auf lexikalische Ressourcen zurück. Es ist deshalb erforderlich, die Anforderungen zu analysieren, die die jeweiligen Komponenten an das Lexikon haben, um redundante Datenhaltung zu vermeiden.

2.1 Übersetzungskomponenten

Typischerweise enthalten Übersetzungsumgebungen drei Komponenten: Eine terminologische Datenbank, die für Übersetzer und einfache Übersetzungsprogramme benutzt wird; ein *Translation Memory*, das zumeist auf Satzbasis operiert und bereits übersetzte Sätze im neuen Text ersetzt; und eine Komponente zur maschinellen Übersetzung. Außer dem *Translation Memory* stellen die anderen Komponenten erhebliche Anforderungen an die Lexikon-Datenbank.

2.1.1 Nachschlagen im Lexikon

Der allgemeinste Fall ist, daß man im Lexikon nachschlägt, um Informationen über Termini zu erhalten. In der Anwendung von AVENTINUS handelt es sich vornehmlich um Termini aus dem Drogenbereich, die an der Universität Göteborg gesammelt wurden (SJÖGREEN 1998). Sie werden üblicherweise terminologisch beschrieben und enthalten:

- eine *Definition* (bei Drogen normalerweise die chemische Zusammensetzung)
- eine juristische *Klassifikation* (*vollkommen verboten*, *nur medizinisch verwendbar* usw.; mit nationalen Varianten)
- eine *Kategorisierung* nach Synonymen, Ober- und Unterbegriffen, Slang-Termini usw.
- *fremdsprachliche Äquivalente* für diesen Term

Diese Informationen können dem Benutzer angezeigt werden, wenn er einen Terminus nachschlägt. Weiterführende Informationen (Wirkungsweise der Drogen, typische Konsumierungstechniken usw., vgl. SNPF 1996) können ebenfalls angegeben werden; das ist jedoch im Projekt nicht realisiert. Mehrere Tausend einschlägiger Termini sind im Projekt gesammelt worden.

2.1.2 Termsubstitution

Diese Komponente ermittelt Termini in Texten und schlägt sie in der Lexikon-Datenbank nach. Es handelt sich also um eine Einzelwort-Übersetzung; bei diesem Verfahren werden die Wörter der Dokumentensprache lemmatisiert und in der Lexikon-Datenbank nachgeschlagen; Treffer werden im Text angezeigt bzw. in einem Glossar zusammengestellt.

Offensichtlich ist eine lexikalische Datenbank die Kernkomponente dieses Verfahrens. Es sind folgende Informationen erforderlich, wenn gute Qualität geliefert werden soll:

- die einzelnen *Termini* müssen gespeichert sein, zumindest mit ihrer *Wortart*-Angabe (andernfalls mag es Fehler bei der Lemmatisierung geben).
- da die meisten Termini *Mehrwortbegriffe* sind (Adjektiv-Nomen- oder Nomen-PP-Verbindungen), müssen Mehrwortbegriffe behandelt werden können; die bloße Übersetzung der Einzelteile eines Terminus führt in der Regel zu Fehlern
- zu jedem Terminus ist die Angabe der *Übersetzung* erforderlich
- im Fall von mehrdeutigen Übersetzungen ist die Angabe einer semantischen Kategorisierung notwendig; üblicherweise wird hier das *Fachgebiet* angegeben.

Die im Projektkontext zu beherrschenden Sprachen sind Englisch, Französisch, Spanisch, Deutsch, Schwedisch, Russisch und Arabisch.

2.1.3 Maschinelle Übersetzung

Für einige der behandelten Sprachen gibt es maschinelle Übersetzungssysteme. Im Kontext von AVENTINUS wird das T1-System eingesetzt (SCHWALL & THURMAIR 1997); es übersetzt von Englisch nach Deutsch und Spanisch (in beide Richtungen).

Die Anforderungen von Übersetzungssystemen an eine Lexikon-Datenbank sind bekanntermaßen massiv; sie betreffen den Kernbereich der linguistischen Beschreibungen von Einträgen.

Um zu verhindern, daß die Lexikon-Datenbank proprietäre Informationen verwaltet, wurden die Ergebnisse aus anderen Projekten (v. a. OTELO, vgl. THURMAIR 1997b) benutzt, deren Ziel es ist, eine „systemneutrale“ Repräsentation linguistischer Informationen zu schaffen. Die beteiligten Systeme sind METAL/T1, LOGOS, IBM und PaTrans (ein Eurotra-Nachfolge-System). Die entsprechenden Informationsarten wurden identifiziert und in einen Beschreibungskontext gebracht. Die Informationen sind:

- im *morphologischen* Bereich: Genus, Numerus, Flexionsklasse usw.
- im *syntaktischen* Bereich: syntaktischer Typ, Argumente

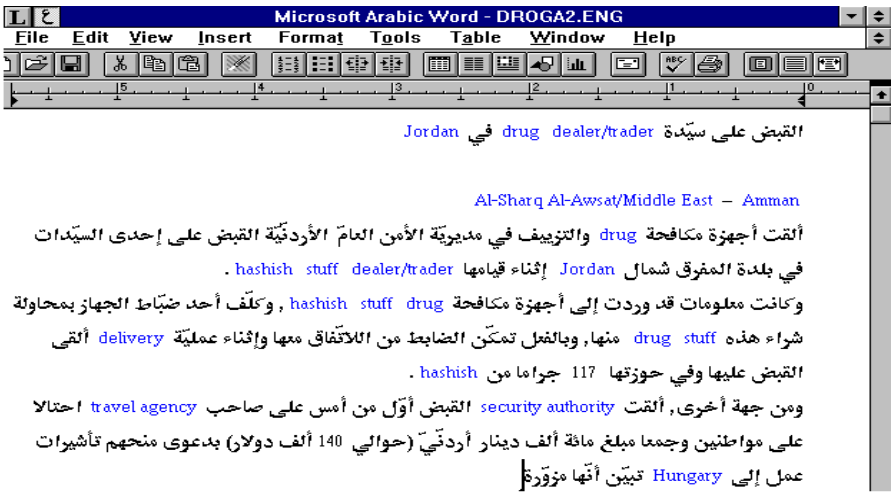


Abb. 2: Beispiel der multilingualen Termsubstitution

- im *semantischen* Bereich: semantischer Typ, natürliches Genus, Aspekt u. a.
- dazu werden *Transfer*-Informationen benötigt, die insofern über die bei der Term Substitution genannten hinausgehen, als sie *formale Tests und Aktionen* beschreiben, denen die analysierten Sätze genügen müssen.

Da die Lexikon-Datenbank „nur" einen gemeinsamen Kern von Informationen ablegt, ist es erforderlich, Konvertierungen zu schreiben, die diese Daten in die jeweiligen proprietären Systeme importieren lassen. Dazu sind entsprechende Austauschformate zu entwickeln.

2.2 Informationsextraktion

Informationsextraktion – im Kontext von AVENTINUS wird das LaSIE System verwendet (vgl. AZZAM et al. 1997, GAIZAUSKAS et al. 1997) – beschränkt sich in der ersten Phase auf die Erkennung bestimmter Informationsobjekte: Personennamen, Institutionen und Firmen, Orts- und Datumsangaben, usw. Diese Objekte sollen aus Texten extrahiert und in strukturierter Form weiterverarbeitet werden.

Manche dieser Informationsobjekte sind ausschließlich lexikalisch identifizierbar (etwa Drogennamen, Autotypen usw.); dafür müssen in der Lexikon-Datenbank die entsprechenden Kategorisierungen vorliegen. Es handelt sich um die spezielle *Semantik der Domäne*; sie ist anwendungsspezifisch und zu unterscheiden von globalen semantischen Kategorisierungen, wie sie in MT-Systemen verwendet werden.

Andere Objekte müssen kontextuell identifiziert werden; dazu gehören die Namen von Personen oder Firmen, Datumsangaben usw. Dazu bedient sich LaSIE einer semantischen Grammatik, deren terminale Symbole bestimmte lexikalische Einheiten sind: Namen von Monaten, Bezeichnungen von Firmen („*AG*", „*GmbH*", „*Ltd*"), Titel von Personen („*Firmensprecher*"), Listen von Vornamen, Synonyme („*IBM*" <-> „*Big Blue*") usw. Dieses Material muß im Lexikon vorhanden und entsprechend beschrieben sein; es ist ebenfalls domänenspezifisch.

Das Verhältnis dieser lexikalischen Einheiten zueinander ist in einem *Modell der Domäne* beschrieben, das zur Laufzeit interpretiert wird, etwa im Fall von Koreferenzbeziehungen, anaphorischen Relationen usw., wenn mehrere Objekte entweder als referenzidentisch oder als Teilnehmer von Szenarien oder Ereignissen beschrieben werden (GAIZAUSKAS 1995). Werden solche Beziehungen multilingual beschrieben (etwa im Fall von AVENTINUS), so muß das Domänenmodell in der Lexikon-Datenbank repräsentiert sein, sodaß verschiedene sprachliche Ausprägungen auf die gleichen Knoten des Domänenmodells zugreifen können (AZZAM et al. 1997).

Auch hier sind die Lexikondaten in die internen Zugriffsstrukturen der jeweiligen Komponente zu überführen; dazu sind wieder Konvertierungen über ein Austauschformat erforderlich.

2.3 Multilinguales Indexing und Retrieval

Eines der Ergebnisse der Evaluierungen im Bereich des Information Retrieval (etwa der TREC-Programme) ist der Umstand, daß intellektuelle und automatische Indexierung zwar annähernd gleich gute Ergebnisse bringen, jedoch typischerweise sehr verschiedene Mengen an relevanten Dokumenten liefern. AVENTINUS versucht durch Kombination beider Techniken die Suchergebnisse zu verbessern, indem es automatische Indexierung verwendet, jedoch im Retrieval die Suchfrage-Formulierung durch das Anbieten von möglicherweise ebenfalls relevanten Termini unterstützt (ähnlich dem Projekt REALIST, THURMAIR et al. 1986). Dazu werden dem Benutzer Verweise angeboten, die auf zusätzliches Material aufmerksam machen, das sich zur Suche eignet.

Dieses Material wird in der Lexikon-Datenbank gespeichert und über einen speziellen Browser dem Benutzer zugänglich gemacht. Es handelt sich um die folgenden Typen von Verweisen:

- *Thesaurus*-Relationen, wie Synonyme und Antonyme, Ober- und Unterbegriff, siehe-auch-Verweise. Sie müssen manuell gepflegt werden. Es handelt sich inhaltlich um die gleichen Relationen wie bei der terminologischen Beschreibung
- *Linguistische* Relationen, wie morphologische Verwandtschaft (Wörter mit gleichen Stämmen, über eine Stemmer-Komponente zu erschließen), syntaktische Verwandtschaft (Erkennung von Head-Modifier-Relationen im Vokabular, vgl. SCHWARZ 1990)
- andere Relationen wie *Abkürzung_von*, *Slangterm_ für* usw.
- *statistische* Information: Zu jedem Terminus wird seine Frequenz und die Zahl der Dokumente festgehalten, in denen er auftritt. Die Randbedingungen des AVENTINUS-Projektes lassen eine aussagefähige statistische Analyse des Bestandes nicht zu; grundlegende Informationen sollen jedoch zur Verfügung stehen.

Die Lexikon-Datenbank muß nicht nur in der Lage sein, zu einem Eintrag die entsprechenden Relationen zu verwalten; sie muß auch in der Phase der Vorbereitung der Suche als Thesaurus verwendbar sein, mit dessen Hilfe die Suchfrage optimiert werden kann. Sie muß somit Querverweise zwischen lexikalischen Einträgen speichern. Teilweise sind diese Verweise identisch mit denen aus dem terminologischen Bereich, teilweise sind sie maschinell erzeugbar.

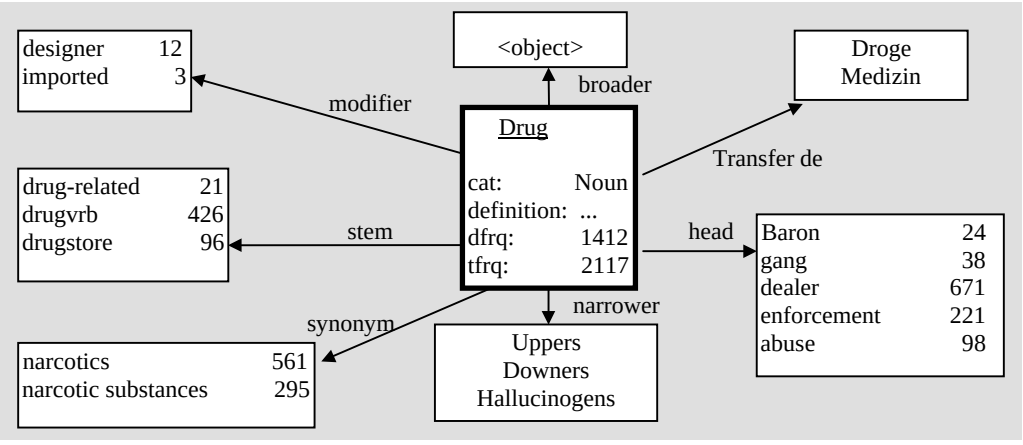


Abb. 3: Thesaurus-Netz für Indexing und Retrieval

Es sei darauf hingewiesen, daß im Bereich der Verweise (und der Frequenzangaben) der Inhalt der Lexikon-Datenbank abhängig vom Dokumentenbestand ist, der bearbeitet werden soll; dies ist üblicherweise bei einem Lexikon nicht, beim Index eines Retrievalsystems jedoch sehr wohl der Fall. Insofern handelt es sich bei der beschriebenen Lexikon-Datenbank um eine hybride Form aus beiden Wissensquellen.

Der Aspekt der Multilingualität des Retrievals beeinflußt ebenfalls das Design des Lexikons: Im multilingualen („cross-linguistic“) Retrieval ist die Suchanfrage aus der Interfacesprache in die Dokumentensprache zu übersetzen, und die Trefferdokumente sind zurück in die Interfacesprache zu bringen. Für beide Fälle werden die oben beschriebenen Übersetzungskomponenten verwendet; dies bedeutet jedoch eine Verzahnung der dort benötigten Lexikoninformationen mit denen der Retrievalkomponente: Konventionelle Indizes enthalten keine Übersetzungen von Indextermen. Insofern bedingt das multilinguale Retrieval eine Änderung der bisherigen Systemarchitektur. Eine zusätzlich erforderliche Komponente ist etwa die Erzeugung von Flexionsformen aus den im Lexikon gespeicherten Grundformen, um die von den jeweiligen IR-Systemen erzeugten (Vollformen-) Indizes bedienen zu können; dafür ist wiederum im Lexikon Information bereitzustellen (Wortart, Flexionsmuster).

Im Kontext von AVENTINUS bedeutet die Verwendung der Lexikon-Datenbank als Frontendkomponente eines Retrieval-Systems nicht nur die Möglichkeit, über Querverweise und Übersetzungen die Domäne besser inhaltlich zu erschließen; sie bedeutet zudem eine verbesserte Repräsentation des Suchvokabulars: Speicherung in Grundform statt in Vollformen, Speicherung von Mehrwortbegriffen als semantischen Einheiten, und Speicherung von Attributen zu Termini (so kann etwa das Ergebnis der Informationsextraktions-Komponente für Textretrieval genutzt werden).

2.4 Suche in strukturierten Datenbanken

Es soll in AVENTINUS möglich sein, kombiniert in Texten und Fakten zu suchen. Natürlichsprachliche Anfragen müssen sowohl in Text-Retrieval-Abfragesprachen als auch in SQL-Statements übersetzt werden. Im letzteren Fall ist eine volle syntaktische und semantische Analyse der Eingabefrage erforderlich.

In AVENTINUS wird ein mehrstufiger Ansatz verfolgt, der im Bereich der syntaktischen Analyse auf die entsprechenden linguistischen Informationen zurückgreift und im Bereich der semantischen Analyse die Verbindung der jeweiligen Begriffe mit dem Modell der Domäne versucht, wie es auch von der Informationsextraktions-Komponente verwendet wird. Hierzu wird lexikalisches Material aus der Lexikon-Datenbank benötigt. Der dritte Schritt, die Konvertierung von der Domäne auf die aktuelle Datenbankstruktur (im Fall von AVENTINUS der Europol-Datenbank) wird außerhalb der Lexikon-Datenbank verwaltet.

Es sei darauf hingewiesen, daß der Aspekt der Multilingualität auch im Bereich der Faktenverarbeitung eine Rolle spielt (THURMAIR & WOMSER-HACKER 1996), etwa im Bereich der Ortsangaben (*Mailand – Milan – Milano*) oder im Bereich der Perso-

nennamen (unterschiedliche Transliterationen russischer oder arabischer Namen), sodaß sich auch hier lexikalische Aufgaben stellen, die im Kontext mit der Backend-Datenbank zu lösen sind.

3 Organisation des lexikalischen Materials

Auf Basis der Analyse der Anforderungen ist nun das erforderliche lexikalische Material zu gruppieren. Dabei wurden die folgenden Kriterien zugrundegelegt:

- Es wurde unterschieden zwischen Material, das projektspezifisch ist, und solchem, das allgemein verwendbar ist: Genusangaben und andere linguistische Eigenheiten, aber auch Synonyme oder Fachgebietszuordnungen, werden als anwendungsunabhängig betrachtet; die spezielle semantische Klassifikation oder das Domänenmodell hingegen sind spezifisch für das AVENTINUS-Projekt. Im Datenmodell wurde deshalb nach Projekt unterschieden und zwischen globalen Teilen und projektspezifischen Teilen getrennt. So kann die Datenbank mehrere Projekte simultan unterstützen (neben AVENTINUS etwa auch das Projekt OTELO).
- Die Daten sollten so organisiert sein, daß es möglich ist, nur bestimmte Komponenten von AVENTINUS zu unterstützen, ohne von anderen Komponenten abzuhängen. Es soll also maximale Modularität gelten: Wenn etwa Anwender keine mehrsprachigen Daten haben, soll die Lexikon-Datenbank nicht dazu verpflichten, Angaben zu Transfers zu machen. Man muß also einzelne Sektoren der Datenbank ausblenden können.

Das Datenmodell, das der Lexikon-Datenbank zugrundegelegt worden ist, ist eine Weiterentwicklung von Lexikonarbeiten, wie sie in den Projekten MULTILEX (MODIANO 1992) und EUROLANG (GAMRAT et al 1992a) entwickelt worden sind.

Im Unterschied zur terminologischen Theorie, die sich an (mehrsprachigen) Begriffen ausrichtet (ein Begriff als semantische Einheit hat verschiedene Benennungen in den verschiedenen Sprachen), und im Unterschied zur Lexikographie, die von Artikeln ausgeht, in denen ein Eintrag verschiedene Lesarten haben kann, gruppiert die Lexikon-Datenbank die Daten nach lexikalischen Einheiten; dabei handelt es sich um monolingual definierte semantische Einheiten.

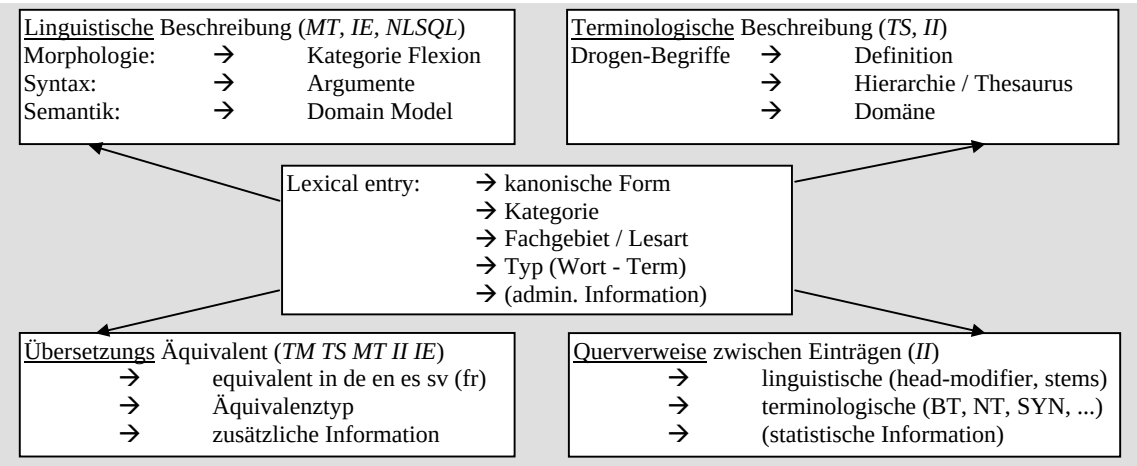


Abb. 4: Organisation der lexikalischen Datenbank

Die Definition semantischer Einheiten ist bekanntermaßen eine theoretisch wie praktisch nicht triviale Aufgabe; im Rahmen der Lexikalischen Datenbank wird eine semantische Unterscheidung pragmatisch dann vollzogen, wenn sie in der Analyse Auswirkungen hat, d. h. wenn sich distributionelle Unterschiede ergeben (anderes Genus, anderer Verbrahmen usw.), die sich in der formal-linguistischen lexikalischen Beschreibung des Eintrags niederschlagen. Es werden dann mehrere lexikalische Einheiten gebildet, die sich durch Lesartennummern unterscheiden.

Um jede lexikalische Einheit herum gruppieren sich die lexikalischen Beschreibungen. Sie folgen, wie üblich, einem Merkmal-Wert-Schema. Werte haben verschiedene Typen: *String* (etwa für Kommentare), *Member* (ein Element aus einer vorgegebenen Wertemenge, etwa beim natürlichen Genus), *Set* (mehrere Elemente aus einer Wertemenge). Die Werte sind atomar; wenn sie in der Verarbeitung in einen speziellen Formalismus eingebettet sein sollen, so ist es Aufgabe des Konverters von der Lexikon-Datenbank in diesen Formalismus, für die korrekte Einbettung zu sorgen. Die Datenbank selbst ist in dieser Hinsicht theorieneutral. Die Merkmal-Wert-Paare der Datenbank sind in verschiedene sog. Sektoren aufgeteilt.

3.1 Allgemeine Information

Dieser Sektor bildet das Kernstück eines Eintrags. Er beschreibt die lexikalische Einheit als solche. Lexikalische Einheiten werden identifiziert durch

- eine kanonische Form (bei adjektivischen Ausdrücken die schwach flektierte Form: *Beamte, ewige Licht*); es kann sich auch um einen Mehrwortbegriff handeln
- eine Wortartangabe (im Englischen sind also *the run* und *to run* zwei lexikalische Einträge, die natürlich aufeinander verweisen können)
- eine Sprachenangabe
- eine Fachgebietsangabe. Diese dient zur pragmatischen Bestimmung der semantischen Information: Während Lesarten im allgemeinen nur relativ zu anderen Lesarten eines Bestandes definiert werden (Was bedeutet *laufen_1*?) und deshalb kaum objektivierbar sind, sind Definitionen in der Regel bei Lexikoneinträgen nicht zu finden. Die einzige „semantische“ Beschreibung, die tatsächlich im Material auftritt und von fast allen Systemen unterstützt wird, ist die Fachgebietsangabe, die insofern pragmatisch als erstes Lesartenkriterium verwendet wird. Innerhalb der Fachgebiete kann nach Lesarten weiter unterschieden werden; die Lesart ist dann näher zu beschreiben.

Weitere Informationskategorien sind:

- Lesartennummer
- Dialektangabe (*Kiberer – österreichisch*)
- Klassifikation nach Eintragstyp (*Einzelwort – Kompositum – Mehrwortbegriff – Abkürzung – Phrase*); sie bestimmt die linguistische Beschreibung des Eintrags

(etwa muß bei Mehrwortbegriffen der Head des Eintrags beschrieben werden, und Abkürzungen flektieren anders als nicht abgekürzte Formen)
- administrative Angaben (Autor und Datum des Eintrags, Kommentare usw.)

Die administrativen Angaben werden vom System gesetzt, obligatorisch sind lediglich die vier identifizierenden Kriterien.

3.2 Linguistische Information

In diesem Sektor wird alle Information gespeichert, die die lexikalische Einheit formal beschreibt. Er ist insofern der Fokus für alle maschinellen Verarbeitungsprogramme, wie maschinelle Übersetzung, syntaktische Analysekomponenten usw.

Die linguistische Information ist unterteilt in solche, die sich auf die lexikalisch Einheit insgesamt bezieht (Verbrahmen, semantischer Beschreibung), und solche, die sich auf einzelne Stammvarianten (Allomorphe) bezieht, üblicherweise morphologische Angaben (Plural bei Nomina, Komparation bei Adjektiven, Tempusangabe bei Verben).

Da sich die linguistischen Annotationen bisher einer Standardisierung sperren (Projekte wie EAGLES oder PAROLE versuchen dem entgegenzuwirken, vgl. UNDERWOOD & NAVARRETTA 1997), ist im vorliegenden Fall wiederum pragmatisch auf die bei OTELO vertretenen MT-Systeme zurückgegriffen und ein größter gemeinsamer Nenner an Informationen festgelegt worden, die alle Systeme nutzen können; speziell darüber hinausgehende Codierungen werden von der Lexikon-Datenbank nur partiell unterstützt. Solche allgemein verwendeten Informationen sind:

- Genus, Flexionsklasse, Komparationsverhalten bei der Morphologie, inkl. Beschreibung der Struktur und Flexionsart eines Mehrwortbegriffs
- syntaktischer Typ und Subkategorisierung, inkl. Reflexivität für die Syntax
- allgemeiner semantischer Typ (*belebt, menschlich* usw.), Aspekt, natürliches Genus zur semantischen Beschreibung.

Diese Information wird von den meisten MT-Systemen verwendet; danach beginnt der proprietäre Bereich der Lexikoninformation; er wird von der Lexikon-Datenbank nicht unterstützt.

3.3 Terminologische Information

Terminologische Information beschreibt die „pragmatische“ Dimension des Eintrags, d. h. seine Verwendung im jeweiligen Kontext. Sie dient dem humanen Nachschlagen wie dem maschinellen Zugriff.

Annotationen in diesem Bereich sind etwa: Definition, Kontext, Bestand (zu welchem Bestand der Eintrag gehört), Quelle (woher er stammt), Produktangabe (wenn er nur für ein bestimmtes Produkt, z. B. SAP/R3, gilt), Verweis auf das Domänenmodell usw. Strukturen von terminologischen Einträgen sind etwa in HOHNHOLD 1990 beschrieben.

3.4 Transfer- Information

Während der linguistische und der terminologische Sektor einen lexikalischen Eintrag für sich beschreiben, geben die Sektoren des Transfers und der Querverweise Verbindungen eines Eintrags mit anderen Einträgen an. Sie definieren Links zwischen Einträgen, wobei im Transfer verschiedene Sprachen, bei den Querverweisen die gleiche Sprache involviert sind. Ein Transferverweis speichert, ausgehend von einem lexikalischen Eintrag, vor allem:

- den zielsprachlichen Eintrag, ebenfalls durch kanonische Form, Wortart, Sprache und Fachgebiet gekennzeichnet
- den Typ der Äquivalenz (*volle – partielle – keine* Äquivalenz, letzteres etwa bei Dienstgraden oder Schulabschlüssen, die verschiedene nationale Realitäten reflektieren)
- spezielle Bedingungen für einen gegebenen Transfer (Tests und Aktionen, verwendet von maschinellen Übersetzungssystemen)

Auf die Transferinformationen greifen die multilingualen Komponenten von AVENTINUS zu.

3.5 Querverweise

Hier werden Links gespeichert, die zwischen Einträgen derselben Sprache bestehen. Die Links gehören zu verschiedenen Typen, die von den Benutzern editiert werden können. Wesentliche Links im Kontext von AVENTINUS sind z. B.

- Synonym / Antonym, Ober-/Unterbegriff. Sie klassifizieren den Bestand entlang der klassischen Thesaurusrelationen
- Stem: Hier werden alle Einträge mit Links aufeinander versehen, die den gleichen Wortstamm haben. Dies wird mit Hilfe von Kompositazerlegern und Stemmern ermittelt
- Head / Modifier: Hier werden alle Wörter, die als Head oder als Modifier zu einem Eintrag auftreten können, angegeben (Diese Information ist korpusabhängig). Es lassen sich daraus semantische Ähnlichkeiten ableiten (RUGE 1995).
- Andere relevante Links im Kontext von AVENTINUS sind: *has_abbreviation*, *slangterm_ for*, sowie verschiedene geographische (*is_ located_ in*) und logische (*part_ of*) Relationen.

Die Querverweise dienen zuvörderst der Verbesserung der Suchfragenformulierung im Retrieval. Es lassen sich jedoch auch andere Verwendungen finden, z. B. Auswertung von Synonymverweisen bei den Transfers der maschinellen Übersetzung: Wenn etwa *bestehen* im Kontext *Examen* mit *to pass* zu übersetzen ist, soll dies auch für *Prüfung Staatsexamen Klausur* usw., also z. B. für Synonyme und Unterbegriffe, gelten.

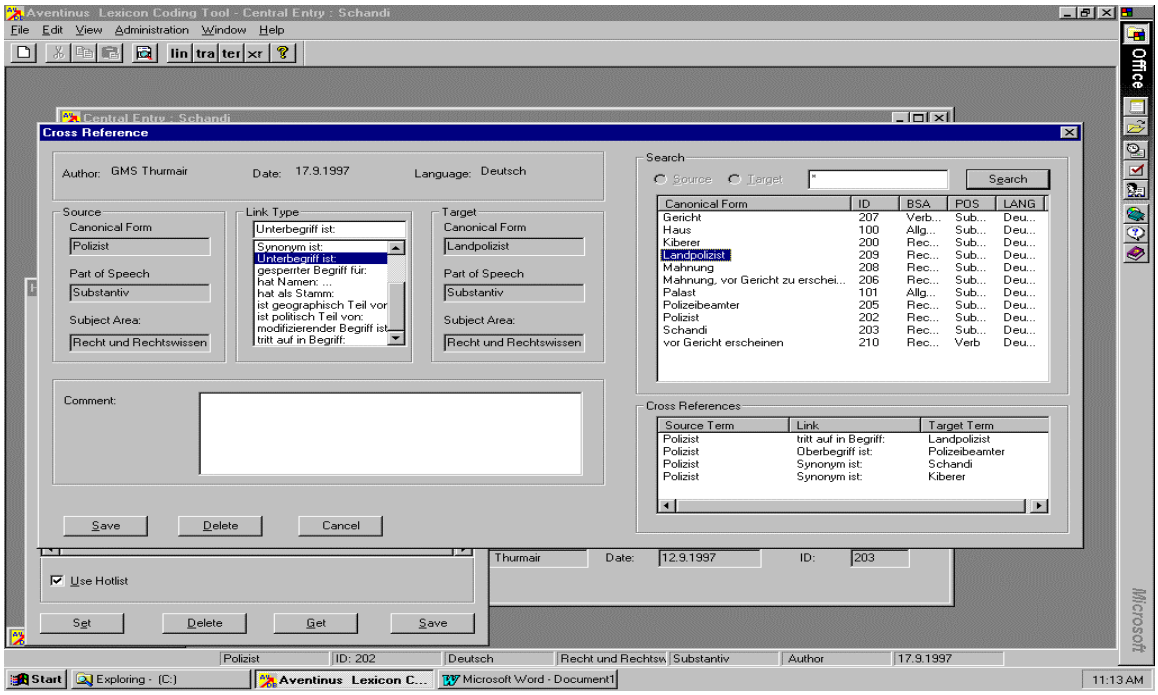


Abb. 5: Coding Tool: Querverweise

4 Implementierung

Die Lexikon-Datenbank ist in der beschriebenen Form implementiert und in mehreren Projekten eingesetzt worden.

4.1 Software

4.1.1 Datenmodell

Die Lexikon-Datenbank in der beschriebenen Form ist in Oracle, Access und SQL-Server implementiert. Sie besteht aus ca. 20 Tabellen, die sich einteilen lassen in

- Datentabellen, die die eigentlichen Einträge enthalten; spezielle Tabellen verwalten mengenwertige Attribute;
- Definitionstabellen, die die jeweiligen Werte der Datentabellen definieren und über entsprechende Foreign_Keys operieren; diese Tabellen bestimmen auch, welche Werte für welche Sprache und welche Wortart gelten sollen (so ist der Genuswert *feminin* für deutsche und französische, aber nicht für schwedische oder englische Nomina zulässig);
- Übersetzungstabellen, die die Anzeige der Werte in der Benutzerschnittstelle regeln; sie sollen sprachspezifisch adaptierbar sein.

Detailliertere Information zum Datenmodell findet sich in RITZKE 1996.

4.1.2 Zugriff

Auf die Daten des Lexikons wird in verschiedener Weise zugegriffen: Die Datenbank bietet eine *Bibliothek* an, mit der zur Laufzeit per Programm Information abgefragt werden kann: Es ist möglich, Einträge zu suchen und auszugeben, Die Querverweise

dienen zuvörderst der Verbesserung der Suchfragenformulierung im Retrieval. Es lassen sich jedoch auch andere Verwendungen finden, z. B. Auswertung von Synonymverweisen bei den Transfers der maschinellen Übersetzung: Wenn etwa *bestehen* im Kontext *Examen* mit *to pass* zu übersetzen ist, soll dies auch für *Prüfung Staatsexamen Klausur* usw., also z. B. für Synonyme und Unterbegriffe, gelten.

- Transfers von Einträgen, linguistische Informationen, Links zu anderen Einträgen usw. Diese Möglichkeit wird etwa von den Term-Substitutions-Komponenten genutzt.
- Darüber hinaus gibt es eine *Codier-Oberfläche*, in der von Benutzern mit entsprechender Berechtigung alle Merkmale eines Eintrags gesucht und editiert werden können. Das Coding-Tool ist gemäß den Sektoren der Datenbank strukturiert; es erfordert nicht, daß ein Eintrag vollständig in allen Sektoren codiert ist, sondern erlaubt, daß ein Eintrag in manchen Sektoren unterspezifiziert ist; dies trägt dem Umstand Rechnung, daß in der Regel mehrere Personen mit unterschiedlichem Know-How an der Codierung eines Eintrags beteiligt sind: Linguisten für den formalen Teil, Übersetzer für die Transfers, Fachexperten für die Terminologie, usw. Die Codieroberfläche enthält ebenfalls administrative Funktionen zur Benutzerverwaltung und zur Pflege benutzerspezifischer Tabellen, wie der Fachgebietshierarchie, einigen terminologischen Angaben (Bestand, Quelle, Produkt, etc.); sie dürfen von einer speziellen Klasse von Benutzern gepflegt werden.

4.1.3 Austausch

Die Lexikon-Datenbank hat u. a. die Aufgabe, die jeweiligen Systemkomponenten wie Informationsextraktion oder maschinelle Übersetzung, mit lexikalischen Ressourcen zu versorgen. Sie bietet dazu online-Zugriffe an; oft ist es jedoch vorteilhafter, optimierte Datenstrukturen zu nutzen, die die jeweiligen Komponenten für ihr lexikalisches Material bereitstellen. Dies bedeutet, daß es spezielle Compiler geben muß, die die jeweils benötigten Ressourcen in die proprietären Formate der jeweiligen Komponenten konvertieren.

Während einige Compiler aus Optimierungsgründen direkte Konversionen vornehmen, operieren andere über ein Austauschformat. Ein solches Austauschformat ist speziell im Kontext des OTELO-Projektes (THURMAIR 1997b) von herausragender Bedeutung, da in diesem Projekt eine gemeinsame Ressource für mehrere Komponenten verschiedener Hersteller geführt werden soll, was die Datenbank in der Projektausprägung für OTELO mit der Aufgabe versieht, Buch darüber zu führen, welche Komponente welchen Eintrag wann erhalten hat.

Für diese Zwecke definiert die Datenbank ein spezielles Austauschformat, das „Open Lexicon Interchange Format“ (OLIF). Es handelt sich um ein XML-basiertes Format, das Austauschdateien erzeugt und akzeptiert. Diese Dateien enthalten einen Header und einen Body. Im Body werden die einzelnen Einträge in Form von Merkmal-Wert-Paaren abgelegt; im Header werden i. W. die verwendeten Merkmale und Werte definiert, sodaß der Body zuverlässig analysiert werden kann. Details sind in THURMAIR, RITZKE & MCCORMICK 1998 beschrieben.

```
<OLIF>
<Header>
<Author          =    Kainer>
<Date            =    12.06.1997>
<FEADEF>                                        ; a feature declaration section
<canonical_form  =    string>
<lang            =    de>
<POS             =    Noun Vrb>                ; gives all possible values
<SA              =    GV ECON>
</FEADEF>
</Header>
<Body>
<Entry>                                         ; a minimal entry description
<MONO>
<canonical_form  =    „Brot“>                  ; a string value
<lang            =    de>
<POS             =    Noun>                    ; a member value
<SA              =    GV>
</MONO>
</Entry>
</OLIF>
```

Abb. 6: OLIF-Austauschformat: Minimalbeispiel

Alle Einträge, die von der Lexikon-Datenbank exportiert oder in sie importiert werden, sind im OLIF-Format von anderen Komponenten ansprechbar. Diese anderen Komponenten (etwa das T1-Übersetzungssystem) stellen Filter bereit, die zu jeder Merkmal-Wert-Kombination angeben, was im Zielsystem geschehen muß:

- Den einfachsten Fall stellen Umbenennungen dar (*Category* statt *Part_of_Speech*).
- Es gibt Umgruppierungen (so wird die Eigenschaft *Eigenname* in manchen Systemen als syntaktische, in anderen als semantische Information klassifiziert).
- Schließlich gibt es komplette Änderungen etwa im Bereich der Verbrahmen; hier ist jedes Merkmal-Wert-Paar u. U. anders zu behandeln, abhängig vom (Nicht-) Vorhandensein anderer Merkmale usw.; man hat es mit komplexen Recodierungen zu tun, die die Spezifikation solcher Filter erschweren.

Zudem sind Fälle zu betrachten, in denen für das Zielsystem notwendige Informationen nicht codiert sind, und umgekehrt etwa zu importierende Daten unterspezifiziert sind und z. B. keine Fachgebietsangabe enthalten.

Filter dieser Art werden im Rahmen von OTELO für T1, Logos, Patrans und SAPterm entwickelt.

4.2 Ressourcen

Die Lexikon-Datenbank ist im Rahmen der Projekte AVENTINUS und OTELO mit Material verschiedener Provenienz gefüllt worden. So läßt sich testen, ob die Software den Anforderungen signifikanter Datenmengen Genüge leistet.

- Im Kontext von AVENTINUS sind Drogentermini gesammelt und codiert worden (SJÖGREEN 1998). Sie enthalten die Begriffe, Definitionen, Klassifikation (Ober- und Unterbegriffe) und Übersetzungen (deutsch – englisch – schwedisch – spanisch). Es handelt sich um etwa 4000 Begriffe.

- Darüber hinaus sind etwa 20000 Begriffe aus verschiedenen Glossaren zusammengetragen worden, die zwei- bis fünfsprachig sind (einschließlich russisch) und Terminologie aus dem Bereich der Rechtspflege und der öffentlichen Sicherheit beinhalten. Diese Terminologie ist zu Testzwecken in die Datenbank importiert; sie muß jedoch in vielen Fällen von Fachexperten validiert werden.
- Im Kontext von OTELO sind mehrere tausend Termini aus dem Bereich der Finanzbuchhaltung importiert worden; sie stammen aus einer klassischen terminologischen Datenbank.

Insgesamt ist das Datenvolumen ausreichend, um die praktische Einsetzbarkeit der Datenbank testen zu können.

Lothar Lemnitzer

Komplexe lexikalische Einheiten in Text und Lexikon

1 Einleitung

Gegenstand dieses Beitrags sind Verfahren für die Extraktion komplexer lexikalischer Zeichen aus großen Textkorpora. Diese Verfahren wurden von mir im Rahmen meiner Dissertation implementiert und anhand einiger Fallstudien getestet. Das Programm wird z. Zt. bei meinem jetzigen Arbeitgeber, der ZERES GmbH in Bochum, für die Erstellung von Phrasenwörterbüchern eingesetzt.

2 Das Problem – komplexe lexikalische Einheiten

In der Mehrzahl der Fälle kann bei der maschinellen Analyse eines Textes nach der Anwendung simpler Segmentierungsverfahren auf Zeichenketten zurückgegriffen werden, die auf Einheiten im Lexikon abgebildet („lemmatisiert“) werden. Die syntaktische und semantische Information in diesen Lexikoneinträgen steht somit für die weitere Analyse zur Verfügung.

Dieser einfache Mechanismus greift nicht bei lexikalischen Einheiten, die im Text durch zwei oder mehrere, nicht notwendigerweise aufeinanderfolgende, Zeichenketten instantiiert werden.

Das folgende Beispiel aus dem Information Retrieval zeigt, daß die Abbildung einer lexikalischen Einheit auf mehrere textuelle Einheiten bzw. umgekehrt bereits vor einer tiefergehenden syntaktisch-semantischen Analyse der textuellen Einheiten notwendig ist.

2.1.1 Beispiel

Eine Anfrage an eine chemische Datenbank enthält das Stichwort „ausfällen“. Der Retrievalmechanismus sollte in der Lage sein, diesem Suchterm Texte zuzuordnen, in denen *fällen, fällt* etc. und *aus* an u. U. weiter auseinanderliegenden Stellen vorkommen.

Zunächst soll eine Klassifikation komplexer Lexeme vorgestellt und Verfahren zu deren Ermittlung aus Textkorpora präsentiert werden. Es wird im folgenden davon ausgegangen, daß diese lexikalischen Einheiten noch nicht im Lexikon des entsprechenden Textanalysesystems vorhanden sind, was angesichts der existierenden NLP-

Lexika und der Diversität und Wandelbarkeit natürlicher Sprachen sicherlich keine unrealistische Annahme ist.

3 Klassen komplexer lexikalischer Zeichen

Es werden, bezüglich ihrer Form- und Inhaltsseite, folgende Gruppen komplexer lexikalischer Zeichen unterschieden:

- Komplexe Funktionswörter (kF), z. B. /sondern auch/, /manch ein/. Diese lexikalischen Einheiten erfüllen genau eine grammatische Funktion, von einer Bedeutung im engeren Sinne kann bei diesen nicht gesprochen werden. Eingliedrige lexikalische Zeichen mit einer komplexen Binnenstruktur (elZ). Die Teile treten in manchen Fällen im Text getrennt auf. Hierzu gehören u. a. die Partikelverben im Deutschen (/auf-hören/), reflexive Verben im Spanischen und Italienischen (/macharse/, /lavar-si/) und Nomen in den skandinavischen Sprachen (/hus-et/). Charakteristisch ist die Zuschreibung einer Bedeutung (oder eines Semems) zu dem gesamten Zeichen. Wie das Beispiel /aufhören/ zeigt, läßt sich die Bedeutung nicht aus der Bedeutung der Bestandteile rekonstruieren.
- Mehrgliedrige lexikalische Zeichen, denen als ganzes eine Bedeutung zugeschrieben wird (mlZ). Diese läßt sich aus den Bestandteilen nicht rekonstruieren. Hierzu gehören die idiomatischen Wendungen oder Phraseme (/jmdm einen Bären aufbinden/, /rote Zahlen/). Phraseme weisen oftmals eine hohe Binnenvarianz auf, was ihre automatische Identifikation in Texten erschwert (vgl. WOTJAK 1992).
- Mehrgliedrige lexikalische Zeichen mit kompositioneller Bedeutung. Diese werden Kollokationen genannt (koll). Sie unterscheiden sich von den „freien“ Verbindungen lexikalischer Zeichen durch die Arbitrarität der gegenseitigen Selektion. Ein Kollokator bindet ein oder mehrere lexikalische Zeichen als Kollokanten, zuungunsten anderer, bedeutungsgleicher oder -ähnlicher lexikalischer Einheiten.

4 Identifikation komplexer lexikalischer Zeichen in Texten

Die bisherige, immerhin schon recht umfangreiche Literatur zur Identifikation und Extraktion komplexer lexikalischer Einheiten aus Korpora konzentriert sich auf den Bereich der Kollokationen, wobei der Term Kollokation allerdings unterschiedlich weit gefaßt wird. Am einen Ende dieser Skala befinden sich Autoren, die Kollokation mit signifikant häufigem Kovorkommen der Bestandteile identifizieren (vgl. z. B. BENSONs Kritik an KJELLMERs Kollokationswörterbuch – BENSON 1995, KJELLMER 1994 – sowie entsprechende Hinweise auf die kontextualistische Tradition in der Nachfolge von FIRTH in LEHR 1996). Am anderen Pol Ende sich die Autoren, die Kollokationen als „typische, spezifische und charakteristische Zweierkombinationen von Wörtern“ auffassen (HAUSMANN 1985:118, so auch SMADJA 1992). Neben Kollokation (hier im engeren Sinne verstanden) können und sollten aber auch die anderen

Typen lexikalischer Einheiten aus Korpora extrahiert und in ein gutes NLP-Lexikon übernommen werden.

5 Verfahren der Extraktion

5.1 Konkordanzen

Zunächst wird eine Zeichenkette, deren typische Begleiter ermittelt werden sollen, ausgewählt. Zu dieser Zeichenkette (oder, im Falle einer abstrakten Einheit, Menge von Zeichenketten) werden Konkordanzen aus einem Korpus extrahiert.

5.2 Auswahl und Charakterisierung kovorkommender Zeichenketten

Alle Zeichenketten, die in einer vordefinierten Umgebung (Kotext) der Kernzeichenkette vorkommen, werden extrahiert und auf Zeichenkettentypes abgebildet. Die quantitativen Eigenschaften des Kernzeichenkettentyps und der kovorkommenden Zeichenkettentyps werden in Beziehung zueinander gesetzt. Ergebnis ist eine Ordnung der kovorkommenden Zeichenketten nach der Signifikanz ihres Kovorkommens mit der Kernzeichenkette. „Signifikanz“ wird hier als quantitativer Term verstanden, der sich in einer durch eine statistische Methode ermittelten Kennziffer für jede Zeichenkette niederschlägt.

Mehrere statistische Verfahren wurden hierfür in der Literatur vorgeschlagen: *Mutual Information Index* (MI, CHURCH et al. 1991), *z-Score* (SMADJA 1992), *Maximum Likelihood Ratio* (mlr, DUNNING 1993). Anhand eines Beispiels soll gezeigt werden, daß diese statistischen Verfahren je nach Aufgabenstellung mehr oder weniger gut geeignet sind (s. u. Abschnitt 6). Es erwies sich während der Tests, daß eine Variante der *Maximum Likelihood Ratio*-Statistik für die Ermittlung häufig vorkommender Zeichenkettenkombinationen in großen Datenmengen am besten geeignet ist. Die entsprechenden Formeln sowie die Testergebnisse sind in Abschnitt 6 wiedergegeben.

Allen diesen Verfahren ist es jedoch eigen, daß sie eine Zufallsverteilung der vorgefundenen Entitäten (Zeichenketten) voraussetzen. Dies ist jedoch in bezug auf den Gegenstandsbereich, Texte einer natürlichen Sprache, eine unzutreffende Idealisierung. Das einzige getestete verteilungsfreie Verfahren, der Fisher-Test, erwies sich allerdings als nicht anwendbar, da bei der großen Menge an zu verarbeitenden Daten die numerischen Werte in einem Bereich lagen, der von gegenwärtigen Computern nicht verarbeitet werden kann. Deshalb wurde schließlich gänzlich auf Teststatistiken verzichtet und die Daten lediglich nach der durch die jeweils angewendete Statistik gewonnenen Kennziffer geordnet.

5.3 Komplexe Funktionswörter

Als Beispiel wird hier die Kernzeichenkette *manch* gewählt. Aus der Tabelle in Abschnitt 6 läßt sich ersehen, daß *manch ein* eine typische Verbindung ist. Wenn diese

Kombination die Funktion eines komplexen Indefinitpronomens hat, dann gilt, daß die syntaktische Umgebung dieser Einheit den Umgebungen anderer, atomarer Indefinitpronomen ähnlicher sein sollte als den Umgebungen von Indefinitpronomen-Artikel-Folgen.

Dies wurde anhand eines getaggten und manuell korrigierten Korpus von 300 000 Textwörtern überprüft. Die Umgebungen der entsprechenden Einheiten, die Wortklasse des linken und rechten Elements jeder betrachteten Zeichenkette, wurden aufsummiert und als Vektoren repräsentiert. Ähnlichkeit der Kotexte konnte so quantifiziert werden als Distanz der Vektoren zueinander.

Die Ergebnisse sind in Abschnitt 6 aufgeführt. Sie sprechen in der Tat dafür, *manch ein* als komplexes Indefinitpronomen einzustufen.

5.4 Kollokationen

Als Beispiel wurde die Kernzeichenkette *harten* gewählt. Ein Blick auf die Tabellen in Abschnitt 6 zeigt zweierlei: Erstens entsprechen die Ergebnisse der Ordnungstatistiken *z-Score* und *Maximum Likelihood Ratio* eher der intellektuellen Bewertung von Zeichenkettenpaaren als Kollokationen als die Ordnung durch den *Mutual Information Index* als Kennziffer. Zweitens befinden sich in den höheren Rängen auch Zeichenketten, die bei einer strengeren Auswahl nicht als Kollokanten akzeptiert werden würden. Intellektuelle Auswahl ist hier vonnöten, evtl. unterstützt durch weitere Filter, wie sie SMADJA vorschlägt.

5.5 Einfache mehrgliedrige lexikalische Zeichen

Als Beispiel wurde die Kernzeichenkette *weg* gewählt. Um die Auswahl der Kollokanten zu präzisieren, wurde das zugrundeliegende Korpus automatisch getaggt. Gesucht wurde sodann in jeder Belegstelle nach dem letzten vor der Kernzeichenkette liegenden Verb. Dabei wurde die Fehlerrate des Tagging-Prozesses in Kauf genommen in der Erwartung, daß bei der Menge der Daten sich dieser Fehler egalisiert: Die irrtümlich ausgewählten Zeichenketten des Kotextes befinden sich an dem für die weitere Analyse der Daten irrelevanten unteren Ende der Liste. Zusätzlich wurden die Kollokanten lemmatisiert, so daß in der Tabelle in Abschnitt 6 als Kollokanten die von *weg* präfigierten Simplexverben in der Grundform erscheinen.

6 Ergebnisse

6.1 Mutual Information Index

Die Größe der *Mutual Information* (*MI*) zweier Ereignisse *a* und *b* wird dargestellt als die Differenz der Wahrscheinlichkeit *I* des Kovorkommens beider Ereignisse $I_{ab} = -\log_2 P(a,b)$ im Ereignisraum und der Wahrscheinlichkeit des Vorkommens beider Ereignisse unabhängig voneinander in diesem Ereignisraum (*P(a)* bzw. *P(b)*):

$$MI(a,b) = \log_2 \frac{P(a,b)}{P(a)P(b)} = \log_2 P(a,b) - \log_2 (P(a)P(b))$$

Für das Beispiel *harten* ergibt dies folgende Ordnung (in diesem und den beiden folgenden Beispielen wurden alle kovorkommenden Zeichenketten in einem Fenster von sieben Einheiten mit der Kernzeichenkette im Zentrum betrachtet):

word y	***fy***	***fxy***	***erg***
Disputes	4	1	14.479224
Bühnenbretter	4	1	14.479224
Steinzellen	4	1	14.479224
Bundesligageschäft	5	1	14.157296
Bremsmanövern	5	1	14.157296
Bandagen	64	11	13.938656
Farbkontrast	6	1	13.894261
Anhängerbetrieb	6	1	13.894261
Gitarrenriffs	8	1	13.479224
Verdrängungswettbewerbs	8	1	13.479224
...			

6.2 z-Score

Die Häufigkeit des Kovorkommens von Kernzeichenkette und Begleiterzeichenkette bildet die numerische Kenngröße, d. h. den Wert, den die Zufallsvariable *X* für das entsprechende Zeichenkettenpaar annimmt. Man geht hier davon aus, daß diese Zufallsvariable normalverteilt ist mit Erwartungswert μ und Varianz σ. μ und σ werden durch den Mittelwert und die Standardabweichung der untersuchten Stichprobe geschätzt. Dadurch wird die Transformation der $N(\mu, \sigma^2)$-verteilten Zufallsvariablen *X* in eine $N(0,1)$-verteilte Zufallsvariable *Y* möglich:

$$Y = \frac{X - \mu}{\sigma}$$

Für das Beispiel *harten* ergibt dies folgende Ordnung:

word y	***fy***	***fxy***	***erg***
Wettbewerb	4747	12	11.445311
Bandagen	64	11	10.445311
und	16689	10	9.445311
Kern	2119	9	8.445311
Wettbewerbs	761	9	8.445311
Drogen	936	5	4.445310
Bedingungen	4389	4	3.445310
lVorgehen	1974	4	3.445310
Kampf	5524	4	3.445310
internationalen	8747	4	3.445310
Haltung	4381	4	3.445310
Währungen	827	4	3.445310
....			

6.3 Maximum Likelihood Ratio

Für die beiden Ereignisse: *Zeichenkette kommt vor* und *Zeichenkette kommt nicht vor* (ermittelt für jede Position im Text und für die beiden Zeichenketten *A* und *B*) wird ermittelt, ob das Verhältnis von Vorkommen und Nichtvorkommen von *A* unter *B* signifikant anders ist als dieses Verhältnis unter ¬*B*. Dafür wird eine Vierfeldertafel für die vier verschiedenen Ereignistypen gebildet und deren Randsummen ermittelt. Nach der *Maximum Likelihood*-Verteilung ergibt sich folgende Ordnung für *harten*:

word y	***fy***	***fxy***	***erg***
Bandagen	64	11	214.875122
Wettbewerb	4747	12	131.113022
Wettbewerbs	761	9	126.062775
Kern	2119	9	107.590057
und	16689	10	80.670708
Drogen	936	5	62.014259
Währungen	827	4	48.806198
Konkurrenz-kampf	212	3	43.057476
Vorgehen	1974	4	41.859688
Zweikampf	444	3	38.604744
Konturen	517	3	37.690151
Haltung	4381	4	35.529675
...			

6.4 Komplexes Funktionswort

Zugrundegelegt wurde ein ca. 35 Millionen Token umfassendes Textkorpus. Es wurden für alle 600 Vorkommen von *manch* die Umgebung von drei Zeichenketten links und drei Zeichenketten rechts betrachtet (Fenster von sieben Einheiten inkl. Kernzeichenkette). Es wurde die *Maximum Likelihood Ratio* als Kenn- und Ordnungsgröße für die kovorkommenden Zeichenketten von *manch* gewählt.

word y	***fy***	***fxy***	***erg***
einer	97904	151	1410.875854
anderer	7262	34	385.555664
anderem	9983	28	288.609741
ein	158354	34	180.590317
andere	27772	20	152.119812
anderen	38979	17	112.662346
anderes	4580	11	109.671936
einen	70716	18	100.885048
einem	81852	16	81.797462
kritischer	326	2	23.652391
...			

Den größten Anteil an den Begleiterzeichenketten bilden die verschiedenen flektierten Formen der Lexeme /eine(r,s)/ und /andere(r,s)/. Es wurden für *manch* und *ein, eine* etc. die Kontexte mit denen von Indefinitpronomen (Ria und Rip) und Pronomen-Artikel-Kombinationen verglichen. Je größer der Vergleichswert für die Kotextvekto-

ren, um so ähnlicher sind die Klassen. Es wurden jeweils die Vorfeldvektoren, die Nachfeldvektoren und beide Vektoren zusammen (Gesamt) berücksichtigt.

X₁	***X₂***	***Gesamt***	***Vorfeld***	***Nachfeld***
Rip	manch einer	0.681367	0.582980	0.779754
Ria	manch einer	0.244798	0.453153	0.036443
Rip:Du	manch einer	0.087039	0.174078	0.000000
Ria:Du	manch	0.000000	0.000000	0.000000

Analyse und Vergleich von Kotextvektoren für Strings und Wortklassen.

Erläuterung der Spaltenbezeichnungen:

X_1, X_2 Erstes, zweites Vergleichselement
Gesamt Wert des Vergleichs des gesamten Kotextes (Vektor des Vorfeldes und Vektor des Nachfeldes) von X_1 und X_2
Vorfeld Vergleich bezogen auf die Vorfeldvektoren;
Nachfeld Vergleich bezogen auf die Nachfeldvektoren.
Rip pronominal verwendete Indefinitpronomen
Ria attributiv verwendete Indefinitpronomen
Du unbestimmter Artikel.

Die Tabelle zeigt, daß die Folge *manch einer* in bezug auf die syntaktische Umgebung der Klasse der pronominal verwendeten Indefinitpronomen (Rip) am ähnlichsten ist.

6.5 Präfixverben

Es liegt das o. g. Korpus zugrunde und es wurde die o. g. Ordnungsstatistik angewendet. Zuvor wurden hier jedoch die Kollokanten auf ihre Grundform (Lemma) abgebildet. Es wurden außerdem nicht alle Elemente des Umfeldes betrachtet, sondern nur das jeweils letzte vor der Kernzeichenkette stehende Verb (ermittelt durch automatisches Tagging des zugrundeliegenden Korpus).

wordy	***fy***	***fxy***	***erg***
sein	46696	311	3234.180176
fallen	3773	123	1654.873413
gehen	5730	62	693.695129
kommen	10056	60	599.872681
laufen	2584	48	588.704834
werfen	1443	27	331.155273
bleiben	6995	31	290.986938
nehmen	16852	33	256.580017
schnappen	28	12	228.026871
stecken	941	18	221.461212
wollen	11908	27	217.594360
ziehen	4756	19	174.282028
rennen	159	11	164.101135
fahren	3841	15	136.875610
...			
schlagen	2024	1	5.124152
lesen	3060	1	4.383187
nutzen	4247	1	3.825902
schaffen	6568	1	3.146245

Nach den Punkten folgt der unterste Teil der Tabelle. Wie man sieht, ist in diesem Fall die gesamte Tabelle für die Bildung lexikalischer Einheiten relevant.

Uwe Quasthoff, Universität Leipzig

Projekt *Der Deutsche Wortschatz*

1 Einleitung

Die Sammlung und Aufarbeitung lexikalischer Daten ist bei größeren Projekten häufig auf mehrere Mitwirkende verteilt, wichtiger Bestandteil ist aber auch die zentrale Koordinierung und Qualitätskontrolle der erfaßten Daten. Ziel des vorgestellten Projektes ist es, eine völlig neue Art einer „selbstorganisierenden“ Koordinierung zu erproben. Die Aufgabenstellung ist wegen des Wunsches nach einer großen Zahl von Mitwirkenden einfach gewählt: Ziel ist die Erfassung eines möglichst großen Teils des deutschen Fachwortschatzes. Die Erfassung erfolgt mit Hilfe einer vorgegebenen Software durch Mitarbeiter aus den verschiedensten Fachrichtungen auf freiwilliger Basis, die Qualitätskontrolle wird ebenfalls dezentral organisiert, indem bereits erfaßtes Material zur Diskussion gestellt wird und Änderungen vorgeschlagen werden können. Die Rolle der zentralen Koordinierung beschränkt sich dabei auf die (halbautomatische) Organisation des Informationsflusses und eine allgemeine Aufsichtsfunktion.

2 Konzept der Sammlung

Grundlage der vorliegenden Projekts ist eine Sammlung von Wortformen der deutschen Sprache, gesammelt auf der Grundlage umfangreicher elektronisch verfügbarer Textkorpora. Bisher liegen ca. 2,5 Millionen Wortformen vor, der ausgewertete Text bestand zum größten Teil aus Zeitungstexten, zu einem geringeren Teil auch aus Fachtexten oder speziellen Wortlisten. Dementsprechend ist der Wortschatz der geschriebenen Umgangssprache zu einem sehr großen Teil abgedeckt. Dagegen gibt es große Lücken im fachsprachlichen Bereich, die sich auch nicht ohne Änderung der Erfassungsmethode beseitigen lassen.

Deshalb soll mit dem Projekt ein völlig neuer Weg beschritten werden. Mit der Sammlung zusammen wird eine Software auf CD-ROM zur Verfügung gestellt, die es ermöglicht, aus einem vorgelegten Text die neuen, der Sammlung unbekannten Wörter zu ermitteln. Damit können dezentral Nutzer aus ihnen vorliegenden, aber möglicherweise nicht allgemein zugänglichen Texten diese neuen Wörter ermitteln, durch manuelle Kontrolle eventuelle Rechtschreibfehler beseitigen und die so entstandene

Wortliste zurücksenden, damit sie in die zentrale Sammlung aufgenommen werden kann.

Weiterhin soll der externe Nutzer die Möglichkeit haben, Wortformen zum Entfernen aus der zentralen Sammlung vorzuschlagen, um beispielsweise mit orthographischen Fehlern behaftete Einträge entfernen zu können. Ziel ist ein bewußter Umgang mit möglicherweise fehlerbehafteten Daten.

Ein weiterer Schwerpunkt besteht in der zentralen Bearbeitung der zurückgesandten Wortlisten. Auch hier soll der manuelle Aufwand minimiert werden, und es soll getestet werden, in wieweit die Verantwortung für die Qualität der Sammlung dezentral gelassen werden kann. Durch regelmäßige Updates erhalten die dezentralen Nutzer die aktualisierten zentralen Daten.

2.1 Zum dezentralen Sammeln

Das Vorgehen des Sammeln von Material durch Mitwirkende auf freiwilliger Basis ist nicht neu. Im Bereich der Lexikographie wurde beispielsweise Material für das Wörterbuch der obersächsischen Mundarten von zunächst 1600 Mitwirkenden zusammengetragen, von denen 400 zu einer längerfristigen Zusammenarbeit bereit waren (BERGMANN 1993:X).

Das folgende Beispiel mit wesentlich mehr mitwirkenden stammt aus der Biologie: Vor ca. zehn Jahren rief der britische Marienkäferforscher Michael MAJERUS seine Landsleute zum Beobachten dieser Insekten auf. 30.000 Teilnehmer sorgten für eine einmalige Feldstudie (vgl. MAJERUS 1994).

2.2 Warum die Sammlung von Vollformen?

Die Sammlung von Vollformen bringt gegenüber der Sammlung von Grundformen mehr Material in die Sammlung, das von einer gewissen Redundanz ist. Die Entscheidung zur Sammlung von Vollformen hat folgende Gründe:

- Bei Auswertung von Volltext ist die Sammlung von Vollformen relativ einfach, die zusätzliche Reduktion auf Grundformen ist eine mögliche Fehlerquelle.
- Die vorliegende Redundanz kann zur Fehlerkorrektur genutzt werden, wenn beispielsweise das orthographisch fehlerhafte Wort einzeln mehreren flektierten Formen des korrekten Wortes gegenübersteht.
- Aussagen über das Nichtvorkommen bestimmter flektierter Formen sind möglich.
- Speicherplatzprobleme treten bei der Vollformensammlung nicht auf, da als Distributionsmedium die CD-ROM zur Verfügung steht.

2.3 Zusammenwirken von dezentraler Erfassung und zentraler Verwaltung

Die folgende Übersicht zeigt die verschiedenen Arbeitsschritte sowohl bei den Nutzern als auch bei der halbautomatischen zentralen Lexikonverwaltung.

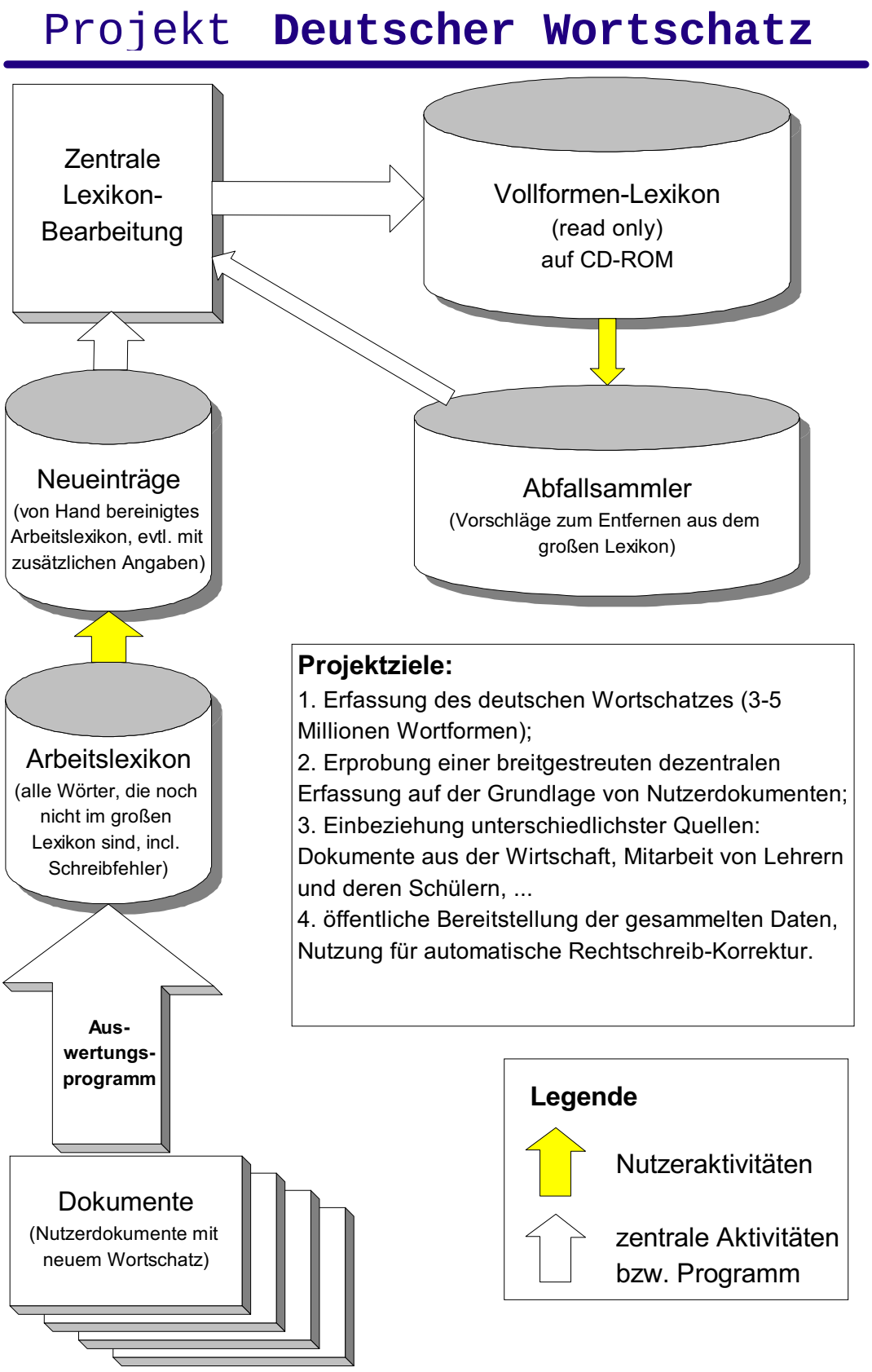


Abb. 1: Übersicht der Ablaufprozesse im Projekt Deutscher Wortschatz.

3 Erweiterte Information in der zentralen Sammlung

3.1 Struktur der zentral vorliegenden Daten aus der Textanalyse

Die zentral vorliegenden Daten enthalten mehr Informationen, als für die Erfassung neuer Wortformen notwendig ist (vgl. QUASTHOFF 1998). Die vorliegenden Daten haben folgende Form:

- Frequenz: Häufigkeit des Auftretens dieser Wortform. Diese so ermittelte Häufigkeit ist nur für hochfrequente Wörter zuverlässig, für niederfrequente Wörter ist sie stark abhängig von den ausgewerteten Korpora, da niemals ein repräsentativer Querschnitt „aller Texte“ vorliegen wird. Außerdem kann gegenwärtig die Häufigkeit bei den dezentral erfaßten Wortformen nicht berücksichtigt werden. Aussagekräftig ist also allenfalls die gemessene relative Häufigkeit der Wortformen.
- Wortform mit Information über Groß- / Kleinschreibung
- Weiterhin erfaßt werden ein Beispielsatz sowie Beispieltyp (z. B. Zeitungstext, Fachtext, ...). Damit lassen sich in vielen Fällen orthographische Fehler sicherer erkennen als nur mit der Wortform.

4 Nutzungsmöglichkeiten der Datenbank

4.1 (Halb-)Automatische Vervollständigung der Angaben

Im folgenden werden kurz einige Möglichkeiten skizziert, mittels automatischer Verfahren linguistische Informationen zu extrahieren.

4.1.1 Automatische Ergänzung der grammatischen Angaben, Verweis auf Grundform

Aus anderen maschinenlesbaren Wörterbüchern liegen Wortlisten mit dazugehörigen grammatischen Angaben vor. Ziel ist, aus Grammatikangaben für wenige Grundformen automatisch Grammatikangaben für möglichst viele Wörter zu erzeugen.
Es ist bis auf wenige Ausnahmen möglich,

- automatisch flektierten Formen den dazugehörigen Grundformen zuzuordnen, falls die Grundform in einer solchen zusätzlichen Liste ist,
- für Wortmengen, die aus Kandidaten für flektierte Formen einer Grundform bestehen, die Flexionsklasse der Grundform automatisch zu bestimmen sowie Verweise von den flektierten Formen auf die Grundform automatisch zu erzeugen.

4.1.2 Lexikonbasierte Kompositazerlegung

Eine lexikonbasierte Kompositazerlegung ist möglich, sobald die Bestandteile eines Kompositums im Lexikon vorhanden sind. Durch die Größe der vorliegenden Wortliste lassen sich zusätzliche Informationen wie z. B. Mengen anderer Komposita mit einem bestimmten Bestandteil an einer festen Position ermitteln. Damit lassen sich häufig Mehrdeutigkeiten bei der Kompositazerlegung auflösen, beispielsweise wird

für *Bundeswehreinheit* die Zerlegung *Bundesweh-Reinheit* unterdrückt, da es keine weiteren Komposita mit Kopf *Bundesweh-* gibt.

Die erfolgreiche Zerlegung eines Kompositums wiederum gestattet in den meisten Fällen die eindeutige Bestimmung der Flexionsklasse des Kompositums.

4.2 Praktische Anwendungen

4.2.1 Rechtschreibkontrolle

Die üblichen Verfahren zur Rechtschreibkontrolle bei Textverarbeitungsprogrammen basieren auf Wortlisten und funktionieren folgendermaßen: Ist ein Wort nicht in der Wortliste des Systems enthalten, so wird der Nutzer benachrichtigt und es werden mehr oder weniger passende Verbesserungsvorschläge gemacht. Wegen des geringen Umfangs der Wortlisten wird auch häufig bei korrekten Wortformen zurückgefragt. Dieses Verhalten kann mit einer umfangreicheren Wortliste verbessert werden. Momentan enthält die Wortliste etwa viermal so viele Einträge wie die Rechtschreibkontrollen der großen Textverarbeitungen. Bei einer angestrebten weiteren Vergrößerung des Wortschatzes und einer Überarbeitung für spezielle Zwecke der Rechtschreibkontrolle wird damit eine deutliche Überlegenheit erreicht.

4.3 Neue Abfragemöglichkeiten

Das Vorliegen lexikalischer Daten in elektronischer Form ermöglicht prinzipiell neue Abfragemöglichkeiten (vgl. BLÄSER & WERMKE 1990). Speziell mit den obengenannten Daten lassen sich folgende Abfragen realisieren:

- Welche flektierten Formen gehören zur gleichen Grundform wie eine bestimmte Wortform? In welchem Verhältnis stehen die dazugehörigen Frequenzen?
- Welche Komposita mit einem bestimmten Bestandteil (z. B. *wehr*) sind vorhanden?
- Welcher Fachbegriff (z. B. *Benutzeroberfläche* oder *Benutzungsoberfläche*) wird häufiger gebraucht?

5 Fehlerkorrektur in der Datenbank

5.1 Manuelle Korrektur durch verteilte Nutzer

Trotz großer Sorgfalt bei der automatischen Sammlung enthält die Wortliste ca. 1-2% fehlerbehaftete Einträge. Diese resultieren sowohl aus orthographischen Fehlern im Original als auch Fehlern bei der automatischen Übernahme aus dem vorliegenden maschinenlesbaren Text (z. B. Verwechslung von Bindestrich und Silbentrennung, falsche Interpretation von Sonderzeichen, abruptes Dateiende mitten im Wort, ...)

Weiterhin sind einige Fachbegriffe (z. B. *Betone* als Plural von *Beton*) nur den jeweiligen Experten bekannt, Nichtfachleute können hier Fehler vermuten. Dementsprechend muß bei der Auswertung von dezentral korrigiertem Material mit teilweise widersprüchlichen Angaben gerechnet werden. Da möglicherweise auch manuell eine korrekte Entscheidung ohne Fachleute kaum getroffen werden kann, bleibt nur die

Möglichkeit, solche Einträge als strittig zu markieren und als solche zur öffentlichen Diskussion zu stellen.

5.2 Automatische Korrekturmöglichkeiten

Durch die beim Sammeln zusätzlich erhaltenen Daten ergeben sich Möglichkeiten zur automatischen Korrektur von Fehlern in der Datensammlung wie oben beschrieben. Gleichzeitig lassen sich Informationen über die Häufigkeiten spezieller Fehler gewinnen (vgl. QUASTHOFF 1998).

6 Dienstprogramme

6.1 Textauswertung mit Satzsegmentierer

In der Wortliste sind die Wortformen nach Groß- und Kleinschreibung unterschieden. Da am Satzanfang keine Aussage über Groß- oder Kleinschreibung getroffen werden kann, ist es wichtig, diese Stellen zuverlässig zu erkennen. Deshalb werden bei der Auswertung eines Textes zunächst die Satzgrenzen ermittelt. Wörter am Satzanfang werden wegen der unklaren Groß- / Kleinschreibung nicht weiter berücksichtigt. Von den restlichen Wörtern wird geprüft, ob sie bereits in der Liste der bekannten Wörter vertreten sind. Darin nicht gefundene Wörter werden in ein Nutzerwörterbuch *Neue Vollformen* aufgenommen.

6.2 Die Wörterbuchverwaltung

Mit Hilfe der Wörterbuchverwaltung kann der Nutzer das von ihm erstellte Wörterbuch *Neue Vollformen* editieren und beispielsweise Wörter mit orthographischen Fehlern entfernen. Analog können fremdsprachige Ausdrücke oder nicht übliche Abkürzungen aussortiert werden. Weiterhin kann die Listenverwaltung zur Nachbearbeitung der Liste der bekannten Wörter verwendet werden. Diese Liste mit einem Umfang von über zwei Millionen Einträgen wird sicher immer Einträge enthalten, die offensichtlich fehlerhaft oder zumindest strittig sind. Mit dem Aussortieren dieser Einträge wird die Qualität der Grundliste ständig erhöht.

Die Nutzer sind anschließend aufgefordert, die von ihnen erzeugten Listen an die Zentralen Dienste per Diskette oder e-mail zurückzuschicken. Wünschenswert (aber nicht Bedingung) ist auch die Überlassung der ausgewerteten Texte, um für die zentrale Sammlung zusätzlich Beispielsätze zu den neuen Wörtern erzeugen zu können. Denkbar ist auch eine Sachgebietszuordnung auf der Grundlage der Texte.

6.3 Zentrale Dienste

Aufgabe der zentralen Dienste ist, die von den jeweiligen Nutzern zurückgesandten Wortlisten (also z. B. *Neue Vollformen* und *Schreibfehler*) in die zentrale Sammlung zu integrieren. Für *Neue Vollformen* werden zusätzlich Beispielsätze gespeichert, falls entsprechende Texte mitgeliefert werden. Die Behandlung fehlerhafter Einträge ist

komplizierter. Einmal kann ein Wort für fehlerhaft gehalten werden, obwohl es das nicht ist. Zum anderen wird im Falle eines häufigen Fehlers dieser wieder unbemerkt auftreten und so das fehlerhafte Wort wieder aufgenommen, anschließend wieder entfernt usw. Um dies zu vermeiden, wird eine Liste der strittigen Fälle geführt, in der alle Wertungen zu einem solchen Wort eingetragen und zu einem späteren Zeitpunkt für einen Entscheidungsvorschlag genutzt werden.

7 Ausblick

Das Projekt testet zunächst die Machbarkeit einer dezentralen selbstorganisierenden lexikalischen Sammlung. Dabei sollen Erfahrungen über die Bereitschaft zur Mitarbeit sowie die zu erwartende Qualität gesammelt werden. Bei positiven Ergebnissen ist es denkbar, im nächsten Schritt auch zusätzliche Angaben zu neuen Wörtern zu sammeln. Folgende Möglichkeiten bieten sich an:

- Ergänzung nicht vorhandener bzw. Kontrolle vorhandener Grammatik- und Sachgebietsangaben entsprechend einem vorgegebenen Schema
- Einordnen von Fachbegriffen in einen Thesaurus oder ein Klassifikationssystem analog (vgl. WEHRLE & EGGERS 1967, DORNSEIFF 1970, CHAPMAN 1993). Herstellen weiterer Beziehungen zwischen Wörtern entsprechend der lexikalischen Funktionen der *Meaning-Text Theory* (vgl. STEELE 1990).
- Seit 1998 ist die Datenbank im World Wide Web zugänglich (*http://wortschatz.uni-leipzig.de*).

Jürgen Oesterle

Semi-automatische Extraktion lexikalischer Information aus Korpora (SELIK)

1 Einleitung

Der manuelle Aufbau elektronischer Wörterbücher mit syntaktischer und semantischer Information als Grundlage von Grammatiken zur Bearbeitung von nicht-restringiertem Text ist sehr zeitintensiv und schwierig. Im Folgenden soll gezeigt werden, wie durch das Zusammenspiel eines umfangreichen elektronischen Wörterbuches mit morphosyntaktischer Information, des am CIS entwickelten CISLEX[1], und einer Grammatik für Nominalphrasen (NPn) und Präpositionalphrasen (PPn) des Deutschen aus großen Korpora semi-automatisch lexikalische Information gewonnen werden kann. Bei der lexikalischen Information kann es sich sowohl um die Subkategorisierungseigenschaften von Wörtern handeln als auch um Informationen darüber, welche Modifikationen möglich sind, wobei ich mich im Folgenden auf die Extraktion von Subkategorisierungsrahmen beschränken werde.

2 Aufbau des Systems

Das folgende Schaubild (Abb. 1) gibt einen Überblick über die einzelnen Komponenten des Systems, die anschließend genauer beschrieben werden. Der Überblick ist so aufgebaut, daß neben der allgemeinen Beschreibung der einzelnen Stadien der Bearbeitung eines Korpus anhand eines Beispielsatzes aus dem Korpus, nämlich des Satzes *Der Mann singt schöne Lieder*, gezeigt wird, wie dieser in den einzelnen Schritten bearbeitet wird.

Im ersten Schritt wird das Gesamtkorpus mit einem von MAIER 1995 entwickelten Lemmatisierer bearbeitet. Dieser zerlegt auf der Grundlage des CISLEX, das für die Wörter des Deutschen deren Wortart sowie ihre morphologischen Eigenschaften enthält, den Text in seine Einheiten. Dies geschieht in der Form, daß in dem Text diese Einheiten sowie die dazugehörige lexikalische Information durch SGML-Annotationen markiert werden. Bei diesen Einheiten kann es sich um einfache Wörter handeln, wobei

[1] Vgl. GUENTHNER & MAIER (1996).

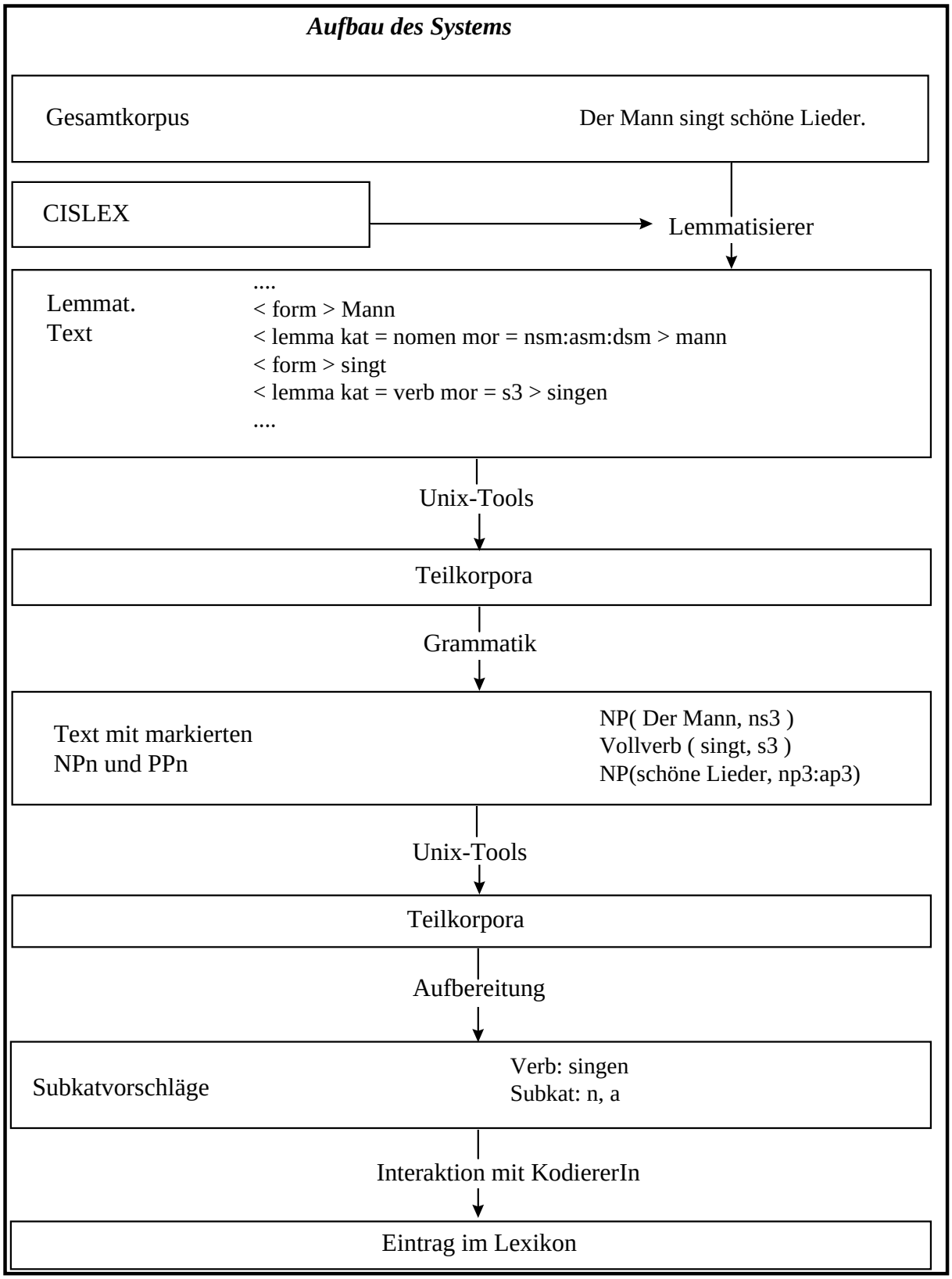


Abb. 1: Systemaufbau

in diesem Fall der Lemmatisierer nur das Wort im Lexikon nachschaut und die dort verzeichnete Information in SGML-Format wiedergibt.[2] Ist das Wort nicht im Lexikon

[2] Im vorliegenden Beispiel wird angegeben, daß die Form „Mann“ erkannt worden ist. Zu dieser Form gibt es die Lemmainformation, daß die Stammform „mann“ lautet, daß es sich um ein Nomen

vorhanden, so wird eine Analyse als Kompositum versucht. Des weiteren werden auch Einheiten analysiert, bei denen es sich um Sonderformen handelt, d. h. Zeichenketten, die Ziffern oder sonstige nicht-alphabetische Zeichen enthalten, wie z. B. *5jährig* oder Datumsangaben wie *2.3.96*. Diese Funktionalität ist von großer Bedeutung, wenn man nicht-restringierten Text behandeln will, da Formen dieser Art je nach Textsorte einen nicht unerheblichen Anteil am Gesamttext haben. Ist keine dieser Strategien erfolgreich, so wird das Wort als unbekannt markiert, wobei dies fast ausnahmslos auf unbekannte Namen zutrifft. Schließlich werden noch mit Hilfe von Heuristiken Satzgrenzen in dem Text festgestellt und markiert.

Im nächsten Schritt können mit Hilfe von Unix-Tools aus dem Gesamtkorpus Teilkorpora erstellt werden, wobei die Kriterien, nach denen dies geschieht, von der Wortart abhängen, für die Subkategorisierungsrahmen gesucht werden. In jedem Fall besteht das Ziel darin, für die weitere Verarbeitung solche Sätze auszuwählen, für die der Vorschlag von Subkategorisierungsrahmen möglichst eindeutig ist, d. h. daß möglichst wenig *noise* vorhanden ist.

Diese lemmatisierten Teilkorpora fungieren dann als Input zu der NP-Grammatik. Hierbei handelt es sich um eine Prolog-Grammatik, die die am häufigsten vorkommenden NPn und PPn des Deutschen erfaßt. Die Regeln sind im Format einer *definite clause grammar* geschrieben, die die zulässigen Abfolgen von syntaktischen Kategorien sowie Kongruenzphänomene festlegt, wie z. B. daß Determinatoren, attributive Adjektive und Nomina in Bezug auf Numerus, Genus, Kasus und Flexion kongruieren müssen.

Der Output der NP-Grammatik besteht dann aus dem lemmatisierten Text, in dem jedoch alle NPn und PPn markiert und ambige Lemmata desambiguiert sind, falls sich dies aufgrund der NP-Analyse ergibt. Mit Hilfe der Grammatik können so komplexe nominale Strukturen erkannt werden. Im Anschluß kann es noch einmal zu einer Bildung von Teilkorpora kommen, je nachdem welche Wortarten untersucht werden.

In der Aufbereitungsphase können unterschiedliche Prozeduren aufgerufen werden. Interessiert man sich z. B. für die Komplemente von Verben, so kann nach dem Ergebnis der Grammatik die NP *schöne Lieder* sowohl als Nominativ als auch als Akkusativ fungieren. Da jedoch das Subjekt in Bezug auf den Numerus mit dem Verb kongruieren muß, wird eine Prozedur aufgerufen, die diese Kongruenz überprüft. Deshalb kommt im vorliegenden Fall nur die NP *Der Mann* als Subjekt, d. h. Nominativ-NP in Frage, da die andere NP, nämlich *schöne Lieder* im Plural steht, während das Verb im Singular ist. Diese NP muß infolgedessen als Akkusativobjekt fungieren, da es sich bei dem Verb nicht um ein Kopulaverb handelt, wobei dies ebenfalls in der Aufbereitungsphase überprüft wird. Weiterhin wird in dieser Phase sichergestellt, daß nur solche Subkategorisierungsrahmen vorgeschlagen werden, die nicht schon ins Lexikon aufgenommen worden sind. In dem betrachteten Beispiel bleibt also nach der

handelt und daß dieses Nomen entweder Nominativ, Singular, maskulin oder Akkusativ, Singular, maskulin oder aber Dativ, Singular, maskulin sein kann. Des weiteren gibt es eine Einheit „singt“, bei der es sich um ein Verb mit der Stammform „singen“ handelt, das in der 3. Person Singular steht.

Aufbereitung nur ein möglicher Verbrahmen übrig, der dem Kodierer vorgeschlagen wird. Bestätigt dieser den Vorschlag, so wird die entsprechende Information ins Lexikon übernommen.

Die Interaktion mit dem Kodierer ist notwendig, da die Ergebnisse der Grammatik zum einen selbstverständlich nicht hundertprozentig richtig sind. Zum anderen ist es aber auch nach der Aufbereitungsphase häufig der Fall, daß, im Gegensatz zu dem hier betrachteten Beispiel, mehrere Subkategorisierungsrahmen in Frage kommen. Dies ist insbesondere dann der Fall, wenn PPn wie in (1) im Satz vorkommen:

(1) Der Mann antwortete dem Mädchen auf die Frage.

In solch einem Fall kann von dem System nicht erkannt werden, ob die PP *auf die Frage* in den Verbrahmen aufgenommen werden soll oder nicht, worauf später noch genauer eingegangen werden wird. Deshalb werden in einer Situation wie dieser dem Kodierer unterschiedliche Einträge angeboten, aus denen er dann einen auswählen kann. Für das Beispiel (1) würde ihm eine Liste präsentiert mit den Optionen, daß es sich bei *antworten* um ein Verb mit einem Nominativ- und einem Dativobjekt, sowie einem Präpositionalobjekt mit der Präposition *auf* handeln kann oder als weitere Option, daß es sich um ein Verb mit einem Nominativ- und einem Dativobjekt handelt, da vom System nicht erkannt werden kann, daß die PP in diesem Beispiel nicht als Adverbial fungieren kann. Für den Fall, daß keine der angegebenen Optionen korrekt ist, kann der Kodierer entweder den Subkategorisierungsrahmen selbst eingeben oder aber er hat die Möglichkeit, die Bearbeitung des Satzes ganz abzubrechen. In diesem Fall erfolgt dann natürlich kein Lexikoneintrag und die Bearbeitung des Korpus wird mit dem nächsten Satz fortgesetzt.

Gegen dieses Vorgehen läßt sich einwenden, daß diese Art der Information zum Teil schon in Form von Lexika vorhanden ist[3] bzw. von einem guten Linguisten ohne Hilfe einer Grammatik kodiert werden kann. Dem kann jedoch entgegengehalten werden, daß es durch diese Methode schnell möglich ist, festzustellen, ob die in Texten auftretenden Subkategorisierungsrahmen auch durch diese Lexika abgedeckt werden. Außerdem kann davon ausgegangen werden, daß auch ein guter Linguist nicht sämtliche Verwendungsweisen eines Ausdrucks parat hat, insbesondere wenn auch unterschiedliche Modifikatoren mitbetrachtet werden. Schließlich kann man aufgrund dieser Vorgehensweise auch Aussagen über die Häufigkeit von bestimmten Konstruktionsmustern gewinnen, wenn man die analysierten Texte mit ihrer Analyse in einer Datenbank speichert. Erkenntnisse dieser Art sind sowohl für statistisch basierte Parse-Ansätze von Interesse als auch für regelbasierte um auftretende Ambiguitäten aufzulösen.

Wie bereits erwähnt, stellt das oben beschriebene Beispiel nahezu den Idealfall der Extraktion lexikalischer Information dar, da der Satz vollständig analysiert werden kann. Im allgemeinen gibt es jedoch eine Reihe unterschiedlicher Faktoren, die die Extraktion erschweren, wie bereits mit dem Beispiel (1) angedeutet wurde. Im Folgen-

[3] Vgl. z. B. HELBIG & SCHENKEL (1983).

den sollen diese genauer beschrieben, sowie Strategien angegeben werden, wie die häufigsten Probleme überwunden werden können oder anders ausgedrückt, wie man dazu kommt, daß dem Kodierer möglichst wenig falsche Vorschläge gemacht werden, sodaß die Kodierung möglichst effizient erfolgen kann. Ein entscheidender Punkt ist hierbei selbstverständlich der Abdeckungsgrad der NP-Grammatik, die deshalb zunächst kurz beschrieben werden soll.

3 NP-Grammatik

Bei der Erstellung jeglicher Grammatik stellt sich die Frage nach der Abdeckung, d. h. wie groß der Anteil der richtig analysierten Strukturen in einem Text ist. Dieser hängt in erster Linie von zwei Faktoren ab, nämlich zum einen von der Größe des Lexikons und dem Abdeckungsgrad der Syntaxregeln und zum anderen von der Art der Strukturen, die erkannt werden sollen. Was Letzteres betrifft, so ist das Ziel, alle Folgen von Wörtern in einem beliebigen Text zu erkennen, die aufgrund ihrer Wortart, deren Abfolge, sowie ihrer morphologischen Eigenschaften als NPn analysiert werden können. Weitergehende syntaktische und semantische Restriktionen werden zunächst nicht berücksichtigt. Anhand der folgenden Beispiele sollen die verfolgten Strategien etwas genauer erläutert werden:

(2) [Peter] hat [das Auto] [am Dienstag] [dem Mechaniker] verkauft.

(3) [Peter] hat [dem Verkauf des Autos] [an den Mechaniker] zugestimmt.

(4) [Der Verkauf des Autos mit der hellen Farbe an den Mechaniker] erfolgte [am Dienstag].

Die Grammatik ist so angelegt, daß in Abhängigkeit von der Setzung eines bestimmten Flags, auf das später noch eingegangen werden wird, versucht wird, jeweils die längstmögliche NP zu extrahieren, wie in den obigen Beispielen durch die Klammerungen angedeutet wird. Die Länge der NPn ergibt sich durch die in der Grammatik festgelegten Abfolgen von syntaktischen Kategorien. So besagen die Regeln z. B., daß ein Nomen eine PP wie *mit der hellen Farbe* oder *an den Mechaniker* als postnominalen Begleiter haben kann oder auch eine genitivische NP wie z. B. *des Autos*. Nicht möglich ist jedoch, daß eine dativische NP wie *dem Mechaniker* als Begleiter eines Nomens fungiert. Wie man an den Beispielen sieht, werden von der Grammatik jedoch unterschiedliche syntaktische und semantische Funktionen von PPn nicht unterschieden, d. h. die PPn *am Dienstag* und *an den Mechaniker* werden genau gleich behandelt, da im Lexikon noch keine Information vorhanden ist, um diese zu unterscheiden, sondern diese Information ja gerade extrahiert werden soll.

Ohne ins Detail zu gehen, seien hier kurz die wichtigsten Konstruktionen zusammengefaßt, die von der Grammatik abgedeckt werden:

– Modifikatoren von Nomen in Form von APn und postnominalen PPn und genitivischen NPn. Dabei können die APn selber wieder komplex sein und NPn, PPn und Adverbien als Begleiter haben wie in *der im Sommer mit großer Mehrheit wiedergewählte Präsident des kleinen Landes am Äquator*

- Koordinationen von NPn, PPn und APn
- Verschiedene Arten von Appositionen, wie z. B. *der Manager der Firma Schulz, Hans Peter Mustermann*

Mit den implementierten Konstruktionen wird ein Großteil der vorkommenden NP-Strukturen abgedeckt, wobei insbesondere bei Koordinationen und Appositionen die Berücksichtigung der Zeichensetzung wichtig ist.

Nach diesem kurzen Überblick über den Umfang der Grammatik, sollen im nächsten Abschnitt einige Methoden erläutert werden, wie die Präzision des Retrieval erhöht werden kann.

4 Strategien bei der Extraktion lexikalischer Information

Bis jetzt haben wir SELIK nur dazu benutzt, um die Subkategorisierungseigenschaften von Verben, Nomen und Adjektiven zu bestimmen. Dabei haben wir satzwertige Komplemente wie z. B. Infinitivsätze oder durch Konjunktionen eingeleitete Nebensätze außer acht gelassen, da es in solchen Fällen schwer ist, festzustellen, welche Komplemente zu welchen subkategorisierenden Elementen gehören. So ist z. B. in Sätzen wie

(5) Peter versucht, das Auto zu verkaufen.
(6) Peters Versuch, das Auto zu verkaufen.

nicht klar, daß *das Auto* Objekt zu *verkaufen* ist und *Peter* das logische Subjekt dieses Verbs. Deshalb werden solche Sätze aus dem lemmatisierten Korpus herausgenommen bevor mit der Analyse durch die NP-Grammatik begonnen wird. Dies bedeutet, daß aus dem ursprünglichen Korpus ein Teilkorpus gebildet wird, in dem nur noch einfache Sätze, d. h. solche mit genau einem Hauptverb, enthalten sind. Die Erstellung dieses Teilkorpus ist sehr einfach, da in dem Ausgangskorpus die Satzgrenzen markiert sind, und die Muster für einfache Sätze durch reguläre Ausdrücke kodiert werden können, die sehr effizient durch Unix-Tools oder *Perl* verarbeiten werden können. Solch ein Vorgehen erlaubt es, von so großen Korpora auszugehen, daß auch nach Tilgung von komplexen Sätzen das verbleibende Restkorpus groß genug ist, um als Grundlage für SELIK zu dienen.

Eine weitere Steigerung der Präzision des Retrieval ist möglich, wenn man die Anzahl der Ambiguitäten reduziert, wie nachfolgend beschrieben wird.

4.1 Behandlung von Ambiguitäten

Die für die Festlegung der Subkategorisierungseigenschaften in erster Linie relevanten Ambiguitäten sind die zwischen der Interpretation einer PP als Komplement und als Modifikator[4], sowie das Problem des „PP-Attachment“, d. h. der Frage, zu welchem

[4] Die Terminologie für die Bezeichnung der unterschiedlichen Begleiter von Nomen, Verben und Adjektiven ist in der Literatur sehr unterschiedlich. Mit *Komplementen* bezeichnen wir solche Be-

Element eine PP als Begleiter fungiert. Beide Fälle können an den bereits erwähnten Beispielen (2) und (3) erläutert werden.

Betrachten wir zunächst das Problem des PP-Attachment. Hierbei handelt es sich um die Frage, zu welchem Element eine PP als Komplement (oder als Modifikator) fungiert. So kann die PP *am Dienstag* in dem Satz (2) von der syntaktischen Struktur her sowohl als Modifikator zu *verkaufen* als auch zu *Auto* analysiert werden. Entsprechendes gilt für die PP *an den Mechaniker* im Beispiel (3), die sowohl als Komplement von *zustimmen* als auch von *Verkauf* interpretiert werden kann. Wie man an der Klammerung in (2) und (3) sieht, werden beide Vorkommen der PPn als Begleiter des Verbs analysiert.

Anders sieht es in dem Beispiel (4) aus: Hier wird die *PP an den Mechaniker* als Teil einer NP analysiert, d. h. als Komplement des Nomens *Verkauf*. Während die Beispiele (2) und (3) ambig sind bezüglich des Attachment der PP, ist die Situation in (4) eindeutig: Hier kann die PP nur als Begleiter des Kopfnomens der NP fungieren, nicht aber als Begleiter des Verbs des Satzes. Diese unterschiedliche Behandlung von PPn was die Frage des PP-Attachment betrifft wird durch das Setzen eines entsprechenden Flags erreicht, worauf weiter oben bereits kurz hingewiesen wurde, d. h. je nachdem, ob eine PP im Vorfeld steht oder nicht wird das Flag unterschiedlich gesetzt. Damit wird der Tatsache Rechnung getragen, daß PPn, die als Teil einer NP analysiert werden können, die im Vorfeld steht, nicht als Begleiter des Verbs in Betracht kommen (vgl. (4)), während PPn, die im Mittelfeld stehen, sowohl als Begleiter eines Verbs als auch eines Nomens interpretiert werden können (vgl. (2) und (3)). Deshalb werden diese PPn syntaktisch als Schwesterkategorien des Verbs behandelt und können als potentielle Verbkomplemente vorgeschlagen werden. Dieses Vorgehen hat zum einen den Vorteil, daß PPn im Vorfeld gar nicht erst als Begleiter des Verbs vorgeschlagen werden. Zum anderen kann man sich auf NPn mit eingebetteten PPn im Vorfeld beschränken, wenn man die Subkategorisierungseigenschaften von Nomen untersucht, denn in diesem Fall ist man dann sicher, daß die PP nur als Begleiter des Nomens in Frage kommt. Dies bedeutet, daß SELIK zur Bestimmung der Komplemente eines Nomens nur solche NPn betrachtet, die im Vorfeld stehen. PPn, die im Mittelfeld stehen, werden wie NPn auch als Kandidaten für Verbkomplemente vorgeschlagen. Für das Beispiel (2) wird demnach als erste Option für das Verb *verkaufen* vorgeschlagen, daß es einen Subkategorisierungsrahmen hat, der aus einem Nominativkomplement besteht, nämlich *Peter*, einem Akkusativkomplement (*das Auto*), einem Dativkomplement (*dem Mechaniker*) und einem PP-Komplement (*am Dienstag*). Als weitere Option wird ein Rahmen vorgeschlagen, der die PP nicht enthält und für die sich in diesem Beispiel der Kodierer dann entscheiden würde, da die PP als Adverbial fungiert und infolgedessen nicht im Verbrahmen aufgeführt werden wird. Auch bei der

gleiter, die im Subkategorisierungsrahmen eingetragen werden oder anders ausgedrückt solche, die für die Valenz dieser Wörter relevant sind. *Modifikatoren* sind Begleiter, die frei vorkommen und auch als Adverbiale oder Adjunkte bezeichnet werden.

Bestimmung des Verbrahmens für das Verb *zustimmen* wird man sich für die Option entscheiden, bei der die PP, nämlich *an den Mechaniker*, nicht als Teil des Verbrahmens aufgeführt ist, da die PP in diesem Fall als Komplement des Nomens *Verkauf* fungiert.

Kommen wir jetzt noch zum Beispiel (4). Wie bereits erwähnt, beschränken wir uns bei der Festlegung von Komplementen für Nomen auf NPn mit eingebetteten PPn im Vorfeld, da hier die PP nicht als Verbkomplement analysiert werden kann. Allerdings sieht man an (4), daß es trotzdem eine „Attachment-Ambiguität“ geben kann und zwar in Bezug auf das Nomen, bezüglich dessen eine PP als Begleiter fungiert: Die PP *mit der hellen Farbe* kann als potentielles Komplement von *Verkauf* und *Autos* fungieren, und die PP *an den Mechaniker* sowohl als potentielles Komplement zu *Farbe* als auch zu *Verkauf* und *Autos* in Frage kommen. Deshalb gibt es in dem System die Option, daß bei der Suche nach präpositionalen Nomenkomplementen nur PPn betrachtet werden, die direkt dem Kopfnomen der NP im Vorfeld folgen. Dies hat den Vorteil, daß die Zuordnung zu dem Nomen eindeutig ist. Andererseits kann es vorkommen, daß mögliche Komplemente unberücksichtigt bleiben. So würde das betrachtete Beispiel (4) überhaupt nicht berücksichtigt bei der Suche nach Nomenkomplementen, da dem Kopfnomen der NP, nämlich *Verkauf*, erst eine genitivische NP folgt und dann erst PPn.

Wie man an diesem Beispiel sieht, kommen neben PPn genitivische NPn als Begleiter von Nomen in Frage und wie das Beispiel (3) zeigt, werden genitivische NPn immer als Begleiter von Nomen analysiert und nicht als solche von Verben, was rein syntaktisch in einer Konstellation wie (3) möglich wäre. Der Grund dafür liegt darin, daß NPn, die im Genitiv stehen können, zum einen meist nicht ambig bezüglich des Kasus sind, und zum anderen die Anzahl der Verben, die Genitivkomplemente haben, relativ klein ist, sodaß für die weitaus meisten Fälle die Analyse einer NP als Genitivattribut eines Nomens korrekt ist.

Indem die Stellung von NPn bzw. PPn innerhalb des Satzes berücksichtigt wird, kann das Problem des PP-Attachment in Bezug auf unterschiedliche Kategorien, die als Kopf in Frage kommen, verkleinert werden. Damit bleibt noch das Problem, daß PPn sowohl als Komplemente als auch als Modifikatoren fungieren können, wie auch schon an den obigen Beispielen gezeigt wurde. Hierzu ist anzumerken, daß das System nur dann eine Analyse präferieren kann, wenn Information bezüglich des semantischen Typs von nominalen Ausdrücken zur Verfügung steht. Dann kann etwa aus der Tatsache, daß es sich bei *Dienstag* um einen Wochentag handelt, geschlossen werden, daß die PP *am Dienstag* als adverbiale Temporalangabe fungiert. Dies hat dann zur Folge, daß sie nicht als mögliches Komplement vorgeschlagen wird. Als erster Schritt in diese Richtung wurde damit begonnen, die in LANGER 1996 vorgenommene semantische Klassifizierung der einfachen Nomina im CISLEX in das System zu integrieren, um so in bestimmten Fällen die eine oder andere Alternative mit großer Wahrscheinlichkeit präferieren zu können.

Neben der Ambiguität zwischen Verb- und Nomenbegleiter ist natürlich auch eine Ambiguität zwischen Adjektiv- und Verb- bzw. Nomenbegleiter möglich:

(7) Peter überreichte die Urkunden auf der Versammlung stolzen Kindern.

In diesem Beispiel könnte die PP *auf der Versammlung* rein von der syntaktischen Struktur her sowohl als Begleiter des Nomens *Urkunden* als auch als Begleiter des Adjektivs *stolzen*, sowie als Begleiter des Verbs analysiert werden, was in diesem Fall korrekt ist. Um solche Ambiguitäten auszuschließen, kann man sich bei der Behandlung von Komplementen von Adjektiven auf Konstruktionen beschränken, in denen die NP, innerhalb derer das Adjektiv steht, mit einem Determinator vorkommt, wie etwa in dem folgenden Beispiel:

(8) Peter überreichte die Urkunden den auf ihre Leistung stolzen Kindern.

In diesem Fall kommt die PP *auf ihre Leistung* nur als Begleiter des Adjektivs in Frage und es stellt sich nur noch das schon oben im Zusammenhang mit Verbbegleiter angesprochene Problem der Unterscheidung von Komplementen und Modifikatoren.

4.2 Berücksichtigung statistisch relevanter Vorkommen

Eine weitere Möglichkeit, um möglichst präzise Vorschläge für PP-Komplemente zu bekommen, besteht darin, daß man nur solche Paare aus subkategorisierenden Wörtern und PPn vorschlägt, die relativ häufig im Korpus vorkommen. Damit können zum einen natürlich Vorkommen unterdrückt werden, die auf einer falschen Analyse eines Satzes beruhen und deshalb nur relativ selten auftreten. Zum anderen beruht diese Strategie auf der Annahme, daß präpositionale Komplemente mit einer bestimmten Häufigkeit in Korpora vorkommen, sodaß seltene Kombinationen nicht berücksichtigt werden sollten. Hierbei spielt natürlich auch eine wichtige Rolle, in welchem Stadium der Extraktion von Subkategorisierungsrahmen man sich befindet, d. h. wieviele Rahmen schon extrahiert worden sind: Zu Beginn wird man sich auf die am häufigsten vorkommenden Kombinationen beschränken, während in einem späteren Stadium, wenn auch seltene Vorkommen ins Lexikon aufgenommen werden sollen, u. U. alle potentiellen Kombinationen in einem Text angeschaut werden müssen. Allerdings ist dann auch die Anzahl der überhaupt in Frage kommenden Kombinationen lange nicht mehr so groß, da selbstverständlich Kombinationen, die schon ins Lexikon aufgenommen worden sind, nicht mehr vorgeschlagen werden.

Die Methoden zur Einschränkung von Ambiguitäten sind damit noch nicht ausgeschöpft, doch sollten die gegebenen Beispiele ausreichen, um einen Eindruck über die Möglichkeiten zu bekommen, die sich ergeben, wenn man partiell analysierte Korpora und weiteres linguistisches Wissen ausnützt, um eine möglichst effiziente Vorgehensweise bei der Suche nach Subkategorisierungseigenschaften zu bekommen. Wie die effizienteste Vorgehensweise im Einzelfall aussieht, hängt natürlich von einer Reihe von Faktoren ab, wobei hier die wichtigsten kurz genannt werden sollen:

- Größe der zu Verfügung stehenden Korpora
- Speicherplatz für die lemmatisierten und getaggten Texte
- Rechnerkapazität um den Lemmatisierer und vor allem die Grammatik laufen zu lassen
- Anzahl der zur Verfügung stehenden KodiererInnen

Der Vorteil der Verfahren zur Steigerung der Präzision des Systems liegt darin, daß je nach gegebener Situation nur bestimmte syntaktische Konstellationen berücksichtigt werden und sich so das System auf die jeweilige Situation gut anpassen läßt. Sind zum Beispiel die Ressourcen in Bezug auf die drei zuerst genannten Punkte sehr gut, so kann man sich auf die Betrachtung von syntaktischen Konstellationen beschränken, die zu einer minimalen Ambiguität beim Vorschlag von möglichen Komplementen führen, da aufgrund der Größe des Korpus dann immer noch genügend Kandidaten für potentielle Komplemente übrig bleiben.

5 Vergleich mit anderen Systemen

Bei den in der Literatur beschriebenen Ansätzen zur Extraktion von Subkategorisierungseigenschaften aus Korpora werden ebenfalls lemmatisierte Texte als Grundlage genommen. Hierauf werden dann bei ECKLE & HEID 1996 mit Hilfe von regulären Ausdrücken Muster gesucht, aus denen sich Subkategorisierungsrahmen ablesen lassen. ABNEY 1996 erwähnt, daß eine Grammatik in Form von Kaskaden von endlichen Automaten verwendet wurde, um Subkategorisierungsrahmen zu extrahieren. ECKLE & HEID müssen sich auf die Ausdrucksmöglichkeiten beschränken, die sich durch die Verwendung von regulären Ausdrücken über den getaggten Texten ergeben, was gegenüber der Verwendung einer Unifikationsgrammatik wie wir sie verwenden von Nachteil ist, da z. B. Kongruenzphänomene nicht oder nur mit großem Aufwand berücksichtigt werden können. Das System von ABNEY geht insofern eher in die Richtung von SELIK, als auch er eine Grammatik verwendet, um lexikalische Informationen zu extrahieren. Allerdings geht er nicht auf die Vorgehensweise ein, wie man zu den Subkategorisierungsrahmen kommt, sodaß ein detaillierter Vergleich mit seinem Ansatz nicht möglich ist.

6 Weitere Anwendungen der NP-Grammatik

Der Hauptvorteil des beschriebenen Ansatzes besteht darin, daß die Extraktion von lexikalischer Information nur eine von mehreren Anwendungen der NP-Grammatik darstellt. Dies bedeutet, daß das System so modular aufgebaut ist, daß die gleiche Grammatik und der gleiche Lemmatisierer auch verwendet werden können, um z. B. aus nicht restringierten Texten NPn für das Information Retrieval zu extrahieren.[5]

Eine andere Anwendung besteht darin, getaggte Texte zu desambiguieren. Dies kann zum einen auf der Wortebene sein wie in dem Satz

[5] Vgl. z. B. EVANS (1991).

(9) Der behauptete Widerspruch läßt sich leicht aufklären.

In diesem Satz kann die Form *behauptete* ohne Betrachtung des Kontextes sowohl als Adjektiv als auch als finites Verb im Imperfekt analysiert werden, während durch die Einbettung des Wortes in eine NP die Wortart desambiguiert wird. Zum anderen können morphologische Ambiguitäten aufgelöst werden, wie in der Konstruktion

(10) dem kleinen Jungen

Hier kann *kleinen* allein u. a. Genitiv, Dativ oder Akkusativ sein, während der Kontext deutlich macht, daß es sich um einen Dativ handeln muß. Weiterhin kann durch die von dem Kodierer getroffenen Entscheidungen das ursprüngliche Korpus verbessert werden, indem Ambiguitäten aufgelöst werden. Dieses verbesserte Korpus kann dann für verschiedene andere Anwendungen wie z. B. statistische Untersuchungen oder zum Training für Parser verwendet werden, die getaggte Texte voraussetzen.
Schließlich könnte ein so aufbereitetes Korpus von Nutzen sein bei der Erstellung von Lexika, die die unterschiedlichen Verwendungsweisen von Wörtern durch Belegstellen aus Texten illustrieren.[6] Hierzu kann man die Wörter je nach ihren unterschiedlichen Subkategorisierungsrahmen unterscheiden. Des weiteren könnte man unterschiedliche Verwendungsweisen von Wörtern entdecken, die von unterschiedlichen semantischen Eigenschaften ihrer Komplemente herrühren, sobald NPn mit semantischen Merkmalen ausgezeichnet sind.

7 Zusammenfassung

Es wurde gezeigt, wie die Implementierung einer Grammatik für einen bestimmten Teil des Deutschen, nämlich Nominal- und Präpositionalphrasen, verwendet werden kann, um semi-automatisch lexikalische Informationen aus Korpora zu extrahieren. Diese Information kann dann im nächsten Schritt verwendet werden, um sowohl die NP-Grammatik leistungsfähiger zu machen, indem diese dann z. B. die Komplemente von Nomen behandeln kann, als auch eine Satzgrammatik für einfache Sätze zu implementieren. Diese Methode läßt sich weiter fortsetzen bis man schließlich zu einer Grammatik für beliebige Sätze kommt.

Voraussetzung ist, daß in jedem Schritt die Extraktion von lexikalischer Information effizient erfolgen kann. Dazu wurde gezeigt, wie dies mit Hilfe unterschiedlicher technischer Mittel wie z. B. Unix-Tools und der Anwendung von linguistischem Wissen, wie etwa über die Typologie des Deutschen, erreicht werden kann.

.

[6] Vgl. KRISHNAMURTHY (1996), der dafür eintritt, daß Korpora viele der Aufgaben von Wörterbüchern übernehmen sollten.

Jan Lass, Fachhochschule Flensburg

Terminologiedatenbank T42

Die Äquivalenzbeziehung im Zentrum eines hierarchisch strukturierten mutlifunktionalen Terminologieverwaltungssystems

1 Einleitung

Ausgangspunkt des Projekts Terminologische Datenbank[1] der Fachhochschule Flensburg war die Suche nach einer Terminologiedatenbank (im folgenden TDB), die einem bestimmten Bedarf gerecht wird. Mit ihr sollen terminologische Daten sowohl für den Menschen als Benutzer erfaßt, aufbereitet und bereitgestellt werden, als auch Daten für maschinelle Übersetzungssysteme unmittelbar oder mittelbar nutzbar gemacht werden. Sie soll somit als produktivitätssteigerndes Online-Hilfsmittel für Übersetzer und Terminologen dienen.

Im Verlauf des ersten Projektabschnitts wurden bestehende Softwaresysteme daraufhin untersucht, ob sie direkt oder in erweiterter beziehungsweise modifizierter Form genutzt werden könnten. Da alle untersuchten Systeme in unterschiedlichen Bereichen Schwächen aufwiesen, fiel die Entscheidung für ein eigenes Softwaresystem, das sich auf einen umfangreichen Katalog von Benutzeranforderungen stützt.

2 Arbeiten mit Terminologiedatenbanken – Theoretische Vorüberlegungen

Der Katalog von Benutzeranforderungen stützt sich auf Beschreibungen aus der Literatur sowie Erfahrungen im Umgang mit TDB.

DIN 2342 beschreibt Terminologie als den Gesamtbestand der Begriffe und ihrer Benennungen in einem Fachgebiet. SCHMITZ 1994b setzt fachsprachliches Wissen mit terminologischem Wissen gleich. Da SCHMITZ in diesem Zusammenhang stellvertretend für viele Autoren steht, kann die Grundforderung an eine TDB wie folgt formu-

[1] Das Projekt Terminologische Datenbank unter Leitung von Prof. Dr. Klaus SCHUBERT, Studiengang Technikübersetzen, Fachhochschule Flensburg, wird mit Mitteln des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie gefördert (Förderungskennzeichen F0 966.00).

liert werden: Eine TDB soll fachsprachliches Wissen verwalten, also Begriffe und ihre Benennungen aus einem oder mehreren Fachgebieten aufnehmen und bereitstellen. Zur Beschreibung der Begriffe und ihrer Benennungen dienen Informationskategorien. Betrachtet man diese Kategorien, so kristallisiert sich auch hier ein Standard heraus, der von vielen Autoren gefordert beziehungsweise in bestehenden Systemen umgesetzt wurde. Zu diesen Informationskategorien gehören, zusätzlich zu Begriff, Benennung und Fachgebiet grammatische Angaben, Angaben zur Verwendung, Quellenangaben, Definitionen und Verwaltungsdaten für mehrere Sprachpaare. Geht man über diese Standardanforderungen hinaus, zeichnet sich ein eher diffuses Bild. Hier galt und gilt es, Forderungen und Anregungen verschiedenster Art zu sammeln und umzusetzen, da diese in bestehenden Systemen nicht oder nur zum Teil realisiert wurden.

2.1 Ordnungsstrukturen in der TDB

Besonders bei anwachsenden Datenmengen ist es von großer Wichtigkeit, Ordnungsprinzipien in der TDB durchzusetzen. Dabei geht es darum, Gruppierungen zu bilden und Relationen (SCHMITZ 1994b) aufzuzeigen. Hierbei bietet es sich an, Begriffe und auch Sachgebiete nicht nur einzeln und alleinstehend zu betrachten, sondern in hierarchischen Systemen (FELBER 1993:122). Diese hierarchischen Systeme sorgen für Übersichtlichkeit und Konsistenz in der Datenbank.

Auch das Erzeugen von Teilbeständen (HOHNHOLD 1990:202) ist ein wichtiges Ordnungsinstrument, das für Übersichtlichkeit und Aktualisierbarkeit der TDB sorgt. Über Teilbestände lassen sich zum Beispiel alle Einträge eines abgegrenzten Sachgebiets abrufen und bearbeiten. SCHMITZ 1994a spricht hier von der Selektion fachsprachlichen Wissens.

SCHMITZ 1994a fordert weiterhin neben der gewohnten alphabetischen Sortierung eine Vorkehrung für eine oder mehrere alphabetisch korrekte Einordnungen von Mehrwortbenennungen. Insbesondere gilt dies, wenn es darum geht, den Datenbestand sortiert auszudrucken oder in anderer Form sortiert weiterzugeben.

Der Verweis auf oder das Einbinden von Abbildungen (ARNTZ und PICHT 1991:248) ist ebenfalls ein Ordnungsinstrument, das für Übersichtlichkeit und Anschaulichkeit der Datensammlung sorgt.

2.2 Terminologiearbeit als dynamischer Prozeß

ARNTZ und PICHT 1991:220 ff. teilen Terminologiearbeit in Vorstufen systematischer Terminologiearbeit und systematische Bearbeitung von Terminologien ein. Die Vorstufen sind die punktuelle Untersuchung, die Kompilation von Fachwörtern und die Bearbeitung größerer grob strukturierter Begriffsfelder. Hierbei geht es also eher darum, zum Beispiel konkrete Übersetzungsprobleme zu lösen. Die systematische Bearbeitung von Terminologien ist tiefergehend. Sie setzt sich genau mit der Einteilung von Fachgebieten sowie mit der detaillierten Untersuchung der Terminologie ausein-

ander, so daß Begriffssysteme entstehen, denen Benennungen und detaillierte Angaben zugeordnet werden.

Die Betrachtungen von ARNTZ und PICHT zeigen, daß Terminologiearbeit ein dynamischer Prozeß ist. Daten werden gesammelt, bearbeitet, überarbeitet und bereitgestellt. Diese Dynamik muß bei der Entwicklung eines Softwaresystems, das Terminologiearbeit unterstützt, berücksichtigt werden.

2.3 Benutzergruppen der TDB

Unterschiedliche Benutzergruppen, also etwa der Terminologe im Gegensatz zum Übersetzer, haben unterschiedliche Anforderungen an eine TDB. Da beide Gruppen aber ihr unterschiedliches Interesse auf den gleichen Datenbestand beziehen, sollte ein ideales System beiden Gruppen gerecht werden (MELBY 1988). So legt der Übersetzer beispielsweise vor allem Wert auf schnellen Zugriff und auf eine Anbindung an eine Textverarbeitung (siehe auch „Bereitstellen von Terminologie“). Für den Terminologen hingegen stehen eher Funktionen im Vordergrund, die das Erfassen, Verwalten und Bereitstellen von Daten unterstützt. Hier ist Flexibilität gefordert, die schnellen Zugriff einerseits und detailliertes Arbeiten andererseits gewährleistet.

2.4 Bereitstellen von Terminologie

Das Bereitstellen beziehungsweise die Nutzung von Terminologie kann am Bildschirm, über Glossare (HOHNHOLD 1990: 203) sowie als Kommunikation mit anderen Systemen erfolgen. Bei der Bereitstellung wird wiederum benutzerspezifische Flexibilität (SCHMITZ 1994a) gefordert, wenn es um Suchfunktionen oder die Darstellung am Bildschirm beziehungsweise auf dem Papier geht.

Die Kommunikation mit anderen Systemen stellt technische Anforderungen. Dazu gehören unter anderem Datenformate für den Im- und Export von Daten, Datenformate für das Bereitstellen von Terminologie für maschinelle Übersetzungssysteme sowie maßgeschneiderte Prozeduren bei der Anbindung an eine Textverarbeitung (MELBY 1988).

3 TDB T42 im Überblick

Die in Entwicklung befindliche TDB T42 wird höchsten Ansprüchen an Modularität gerecht. Das fertige System soll in unterschiedlich ausgestatteten Versionen verfügbar sein, die verschiedenen Anwendergruppen gerecht werden. Dieser Beitrag soll einen Überblick über alle bisher implementierten sowie die geplanten Funktionen geben, die dann später in einigen der unterschiedlichen Versionen nur zum Teil enthalten sein werden.

Die Grundstruktur der TDB mit der Aufteilung in eine Datenbank, die ausschließlich die reinen Daten enthält, und einer Datenbank, in der die Funktionen gespeichert

sind, ermöglicht zum einen unterschiedlich ausgestattete Versionen und zum anderen nachträgliche Erweiterungen des Systems um neue Funktionen.

In der TDB gibt es grundsätzlich nur einsprachige Einträge, wobei jede Sprache in einer gesonderten Tabelle gespeichert wird. Es ist aber aus pragmatischen Gründen durchaus möglich, Einträge in zwei Sprachen gleichzeitig zu erfassen. Die Beziehung zwischen Einträgen zweier Sprachen wird in einer Verknüpfungstabelle hergestellt (Äquivalenzbeziehung). Ein zweisprachiger Eintrag ist somit lediglich als Verknüpfung zweier einsprachiger Einträge vorhanden. Diese Äquivalenzbeziehung bildet das Zentrum der Datenbank und ist für mein Dafürhalten eine wirkliche Neuerung im Bereich Terminologiedatenbanken.

Die Funktionalität der TDB läßt sich in verschiedene Bereiche gliedern:

Arbeitsablauf: (vgl. ARNTZ u. PICHT 1991)

- Verwalten
- Erfassen
- Bearbeiten
- Auffinden
- Bereitstellen

Verwalten von Informationsgruppen:

- reine Terminologiedaten
- Daten zu Fachgebietsklassifikation / Begriffssystemen
- Umgebungs-/Verwaltungsdaten (Quellen, Projekte usw.)
- Exportformate

Besonderes Augenmerk bei der Entwicklung der TDB wurde und wird auf Terminologiearbeit als dynamischen Prozeß gelegt, der in mehreren Schritten erfolgen kann.

Die TDB wird unter der Entwicklungsumgebung für Datenbanken *Microsoft Access* entwickelt, die den vollen Nutzen aus der grafischen Benutzeroberfläche *Microsoft Windows* zieht und häufig schon zur Grundausstattung eines modernen PC gehört. Darüber hinaus läßt sich so eine Netzwerkfähigkeit auf einfache Weise realisieren.

Im folgenden soll auf die einzelnen Funktionen der TDB näher eingegangen werden. Auch bei dieser Betrachtung wird vom natürlichen Arbeitsprozeß ausgegangen.

4 Funktionalität – allgemein

Die TDB ermöglicht Terminologiearbeit in allen Stufen (vgl. ARNTZ & PICHT 1991), also das Erfassen, Bearbeiten, Auffinden und Bereitstellen von Terminologie in unterschiedlich detaillierter Form. In T42 ist dies für die drei Arbeitssprachen Deutsch, Englisch und Dänisch möglich (mit der Option auf Erweiterung um zusätzliche Arbeitssprachen). Daten zu Quellen, Projekten, Kunden und so fort werden gesondert erfaßt und gehen jeweils als Codeschlüssel in die Terminologiedaten ein, können von dort aus aber in detaillierter Form eingesehen werden.

Die TDB unterstützt dabei den Benutzer bei minimaler Erlernzeit durch reichhaltige Funktionen und komfortable Bedienoberflächen. Die Bedienung des Systems erfolgt wahlweise per Maus oder Tastatur über Pulldown-Menüs oder Schaltflächen (Symbolleisten). Eine Online-Hilfe ist in Planung.

Das Erfassen von Daten erfolgt über sogenannte Formulare, die als Eingabemasken dienen. Es wird so weit wie möglich durch Vorgaben (siehe oben) oder über Auswahllisten (Kombinationsfelder), Optionsschalter und dergleichen unterstützt und teilautomatisiert.

Neben den eingangs erwähnten Standardfunktionen zum Erfassen, Bearbeiten und Bereitstellen von Terminologie (s. „Arbeiten mit Terminologiedatenbanken – Theoretische Vorüberlegungen“) werden in der TDB auch die im weiteren Verlauf beschriebenen Anforderungen umgesetzt. Diese Umsetzung soll im folgenden beschrieben werden.

4.1 Automation

Der Benutzer kann Vorgaben für Datengruppen wählen, die beim Erfassen von Terminologiedaten automatisch eingefügt werden. So kann zum Beispiel für eine Reihe von Einträgen automatisch die Quellenangabe gemacht werden. Insgesamt können bis zu 60% der Felder eines terminologischen Eintrages automatisch ausgefüllt werden. Darüber hinaus wird das Suchen von Daten, das Erzeugen von Teilbeständen, der Export von Daten sowie das Erstellen und Ausdrucken von Glossaren durch interaktive Benutzerführung stark vereinfacht und teilautomatisiert.

4.2 Datenbankpflege und -verwaltung

Funktionen, mit denen eine einfache Datensicherung bequem zu realisieren ist, erhöhen die Sicherheit des Systems. Darüber hinaus bietet *MS Access* an sich schon eine Funktion zum Komprimieren von Datenbanken, mit der der benötigte Speicherplatz für die Datenbank in Grenzen gehalten wird.

Über Verwaltungsfunktionen werden die unterschiedlichen Rechte an die Benutzer vergeben. Die obengenannten Verwaltungsfunktionen sind noch nicht implementiert.

5 Daten erfassen

5.1 Allgemein

Neue Einträge können automatisch nachgeschlagen werden, um Doppeleingaben zu vermeiden oder Verweise zu verwandten Einträgen herzustellen.

5.2 Sachgebietsklassifikation erfassen

Terminologiearbeit soll wissenschaftlich sein (FELBER 1993:6). Sie beschäftigt sich also mit wahren und unwahren Aussagen. Zum Beispiel wird untersucht, inwieweit eine Benennung der Sprache A im Fachgebiet X eine adäquate Entsprechung in der

Sprache B hat. Da Terminologie immer an ein Fachgebiet gebunden ist (DIN 2342, 1992), muß also vor der obengenannten Untersuchung ein entsprechendes Fachgebiet festgelegt worden sein.

Sachgebietsklassifikation dient der Einteilung der Terminologie in Fachgebiete. Es können beliebig viele Fachgebiete mit Untergebieten auf verschiedenen Ebenen erfaßt werden, so daß bei enger Benutzerführung hierarchische Systeme aufgebaut werden. Dabei werden automatisch Codes vergeben, die die Identifizierung einzelner Fachgebiete ermöglichen und Aufschluß über die Position in einem hierarchischen System geben. Grundsätzlich läßt das System aber auch andere Klassifikationen zu, die in den terminologischen Einträgen vermerkt werden können.

5.3 Begriffssysteme erfassen

Diese Funktion dient der Einordnung von Begriffen in Begriffssysteme. Es können beliebig viele Begriffe mit Unterbegriffen in einem hierarchischen System erfaßt werden. Der Aufbau der Systeme entspricht dem der Sachgebietsklassifikation.

5.4 Terminologie erfassen

Terminologie kann sowohl ein- als auch zweisprachig erfaßt werden. Beim einsprachigen Arbeiten kann später eine Zuordnung von Einträgen der verschiedenen Sprachen erfolgen, so daß Äquivalente auffindbar werden. Das Prinzip der Äquivalenzbeziehung *kann* (*muß* aber nicht!) vom Anwender akzeptiert werden.

Das Erfassen erfolgt in unterschiedlich detaillierter Form. Ausgehend von einer „schnellen Notiz“, die mit anderen Mitarbeitern diskutiert werden soll, bis hin zum sorgfältig recherchierten Eintrag bietet das System Möglichkeiten, Terminologie zu erfassen und aufzubereiten.

6 Ordnungsinstrumente

Neben den Angaben zu Sachgebieten und Begriffssystemen können detaillierte Informationen zu einzelnen Kategorien, zum Beispiel Quellenangaben, schnell über den entsprechenden Codeschlüssel aufgefunden werden. Insofern sind einerseits stets detaillierte Informationen verfügbar, ohne aber die Übersichtlichkeit des Systems zu beeinträchtigen.

Verweise auf terminologische Einträge können einfach während der Erfassung erzeugt werden. Über diese Verweise kann ein Haupteintrag insbesondere in ausgedruckten Glossaren auch auf Umwegen über Wortumstellungen, Synonyme oder verwandte Einträge aufgefunden werden.

Abbildungen können direkt als Bitmap in einen Datensatz übernommen werden und so für Anschaulichkeit und Übersichtlichkeit sorgen. Es ist kein umständliches Suchen nach Abbildungen außerhalb der Datenbank notwendig.

7 Daten bearbeiten

Sind Einträge erfaßt, können Sie später beliebig bearbeitet werden. Zum Auffinden von zu bearbeitenden Daten stehen reichhaltige Filterfunktionen zur Verfügung, mit denen die entsprechen Daten (Terminologie, Fachgebiete, Quellen usw.) aufgefunden werden können.

Zum Bearbeiten dienen im wesentlichen die beim Erfassen verwendeten Eingabemasken, in denen dann allerdings nur ausgewählte Daten erscheinen. Je nach Art der Daten sollen Kontrollmechanismen zum Beispiel auf Gefahren beim Verändern von hierarchischen Strukturen (z. B. Begriffssysteme) aufmerksam machen. Darüber hinaus können ausgewählte Datenmengen zum Teil automatisch verändert werden.

8 Auffinden

Beim Auffinden von Terminologie kommen sowohl Filter als auch eingebaute Suchfunktionen zum Einsatz. So kann die angezeigte Datenmenge eingeschränkt oder gezielt nach einzelnen Einträgen gesucht werden. In Glossaren kann ein terminologischer Haupteintrag auch über Verweise aufgefunden werden.

Beim gezielten Suchen nach Einträgen kann mit Stellvertreterzeichen (sogenannte Sternsuche) gearbeitet werden. Es erscheinen dann ein oder mehrere Suchergebnisse zunächst als einfache Liste. Zu einzelnen Einträgen der Liste können weitere Informationen aufgerufen werden. Gesucht werden können neben Benennungen auch Synonyme, Kontextbeispiele und Definitionen, Abkürzungen und dergleichen.

9 Bereitstellen

Die Terminologiedaten werden in unterschiedlicher Form bereitgestellt. Daten können am Bildschirm ausgegeben (Siehe „Auffinden“), ausgedruckt oder zur Verwendung in anderen Systemen exportiert werden.

9.1 Export

Über den Export von Daten können gesammelte Einträge gezielt zusammengestellt und für Anwender anderer Datenbanksysteme oder für maschinelle Übersetzungssysteme nutzbar gemacht werden. Derzeit ist bereits eine Exportprozedur implementiert, die Textdateien im Importformat für das maschinelle Übersetzungssystem Logos vollautomatisch erzeugt. In der Planung ist auch ein Exportformat, das sich mit dem Intercoder im maschinellen Übersetzungssystem Metal weiterverarbeiten läßt. Andere Exportformate (variabel gestaltete Textformate und dergleichen) sind in Vorbereitung oder schon implementiert.

9.2 Import

Als Umkehrung beziehungsweise Ergänzung zu den Exportfunktionen sind Importfunktionen für diverse Formate geplant.

9.3 Drucken

Grundsätzlich lassen sich alle vorhandenen Daten ausdrucken. Begriffssysteme und Klassifikationssysteme lassen sich in ihrer Gesamtheit oder in Teilen ausdrucken. Suchergebnisse können mit unterschiedlich detaillierten Informationen ausgedruckt werden. Auch für Listen mit Quellen, Kundendaten usw. stehen automatisierte Möglichkeiten zum Ausdrucken zur Verfügung.

Terminologie beziehungsweise Teilbestände können in ein- oder mehrsprachigen Glossaren/Wörterbüchern zusammengefaßt werden. Dafür steht eine Vielzahl von unterschiedlichen Layouts zur Verfügung. Darüber hinaus können zum Beispiel auch Abkürzungsverzeichnisse erstellt werden.

Die Dokumente können direkt gedruckt oder in geeigneten Anwendungsprogrammen nachbearbeitet werden.

9.4 Anbindung an Textverarbeitung

Die Anbindung an ein Textverarbeitungsprogramm erfolgt unter Windows über eine DDE-Verbindung (DDE = Dynamic Data Exchange). Diese Anbindung wird für Microsoft Word realisiert, kann aber auch für andere Anwendungen hergestellt werden. Benennungen können dann von der Textverarbeitung aus gesucht und gefundene Einträge in den Text eingebunden werden. Diese Funktion wird derzeit im Rahmen einer Diplomarbeit im Studiengang Technikübersetzen der FH Flensburg umgesetzt.

10 Zusammenfassung

Die in Entwicklung befindliche TDB unterstützt den Benutzer in allen Phasen der Terminologiearbeit. Dabei wird insbesondere Wert auf Übersichtlichkeit und Visualisierung von Arbeitsabläufen gelegt.

Das System ist so angelegt, daß spätere Erweiterungen um Funktionen unabhängig von bestehenden Datenbeständen problemlos eingebunden werden können.

Die Besonderheit des Systems liegt unter anderem darin, daß es Terminologiearbeit in aufwendiger und durchstrukturierter Form ermöglicht aber nicht erzwingt. Die Vernetzung der gespeicherten Informationskategorien führt dabei einerseits zu geringem Eingabeaufwand (z. B. Auswählen/Vorgeben eines kurzen Codeschlüssels für eine Quellenangabe), wobei andererseits ein einfaches Aufrufen von Zusatzinformationen zu einem Codeschlüssel möglich ist. Informationen verstecken sich also nicht länger hinter kryptisch anmutenden Zahlen- und Buchstabenkombinationen und werden somit einem breiteren Publikum zugänglich.

Katja Krüger, Daimler-Benz AG, Stuttgart

Mehrsprachige computergestützte Texterschließung für Übersetzer und Terminologen

1 Einleitung

Wir[1] beschreiben ein integriertes System zur Termerkennung für Übersetzer und Terminologen, das in Zusammenarbeit mit dem Sprachendienst der Mercedes-Benz AG entwickelt wurde (KRÜGER 1996).

Der Grundgedanke dieses Systems ist, daß elektronisch gespeicherte Texte, insbesondere bereits abgeschlossene Übersetzungsaufträge, als umfangreiche online-„Nachschlagewerke“ für Übersetzer dienen können, wenn man sie in geeigneter Weise erschließt. Für den Übersetzer sind dabei unter anderem die folgenden Informationen interessant: Bereitstellung von Äquivalentkandidaten aus früheren Übersetzungen, Hilfestellung bei der Auswahl zwischen verschiedenen Übersetzungskandidaten durch Bereitstellung von Kontextbelegen sowie Unterstützung beim Glossaraufbau.

Um diese Funktionen zu realisieren, wurden die Arbeitsabläufe bei den Kunden einerseits und (bei den Entwicklern) vorhandene computerlinguistische Tools andererseits untersucht und Wege gesucht, durch geeignete Kombination der vorhandenen Werkzeuge die wichtigsten Aufgaben der Übersetzer und Terminologen sinnvoll zu unterstützen. Der Arbeitsablauf bei der Übersetzung wird daraufhin untersucht, wo sich Eingriffsmöglichkeiten für die Anwendung von Texterschließungswerkzeugen bieten.

Zu den vorhandenen Texterschließungswerkzeugen zählen Tools zur linguistischen Extraktion von Textbelegen aus Textcorpora und Tools zur morphosyntaktischen Analyse (SCHILLER 1994). Außerdem ein Termerkennungs- und –extraktionsverfahren. Das integrierte System, Termland, kann über einen Webbrowser (z. B. Netscape) benutzt werden.

Nachfolgend ein Beispiel für die Fragestellungen, die mit Termland bearbeitet werden können: Wie wird Englisch ‚Manager‘ in der Textsorte „Managerleitfäden

[1] Ganz herzlichen Dank an Dr. Ulrich HEID für viele äußerst hilfreiche Kommentare.

zum Thema Organisation“ übersetzt? Der Benutzer[2] bekommt Kontextbelege aus früher erstellten Übersetzungen Englisch-Deutsch, aus denen folgendes hervorgeht:

- *sales manager* – Verkaufschef – Verkaufsleiter
- *finance manager* – der für die Finanzen Verantwortliche
- *senior manager* – Vorgesetzter etc.

Termland bietet die Möglichkeit, Listen von Termen und deren Übersetzungen im Kontext systematisch zu extrahieren und erleichtert somit den Glossaraufbau.

2 Anwenderziele und Aufgabenstellung

Das System Termland ist das Ergebnis der Integration verschiedener computerlinguistischer Methoden und Werkzeugkomponenten zur Texterschließung und stellt folgende Funktionalitäten für den Übersetzer bereit:

- Vorschläge für Äquivalentkandidaten;
- Information über verwendete Termini, z. B. Kontextbelegsammlung;
- Unterstützung bei der Äquivalentauswahl.

Leitlinie für die Entwicklung von Termland war dabei die konkrete Benutzungssituation, wie sie im Sprachendienst bei der Mercedes-Benz AG vorliegt. Das System Termland dient nicht nur als online-Nachschlagewerk für Übersetzer und Terminologen, sondern bietet auch Unterstützung bei der Glossararbeit.

Glossareinträge enthalten wenigstens Äquivalenzinformation für jeden Eintrag, unter Umständen auch Kontextbelege, Definitionen und Hinweise auf Benutzungssituationen (um z. B. Corporate Language oder textsorten- oder zielgruppenspezifische Ausdrücke zu identifizieren). Termextraktion automatisch durchzuführen ist sehr erstrebenswert, da

- einheitliche Terminologie für bessere Qualität sorgt,
- aber manuell nur sehr zeitraubend erstellt werden kann.

Letztere Möglichkeit entfällt also de facto, da Übersetzer in der Regel unter sehr hohem Zeitdruck arbeiten. Die benötigte Information kann auf zwei Arten verfügbar gemacht werden: im interaktiven Betrieb über das Benutzerinterface oder zur Wortlistenerstellung im Batchbetrieb.

[2] Im ganzen Text soll Benutzer abkürzend für Benutzer oder Benutzerin etc. stehen.

3 Computerlinguistische Grundlagen

3.1 Texterschließung

Texterschließung findet typischerweise in zwei Phasen statt:

1. Die linguistische Vorverarbeitung und die Annotation von maschinenenlesbarem Text mit der gewünschten Information (etwa Wortklassenannotation, Lemmatisierung etc.) und
2. die Abfrage der Texte anhand dieser Information, die Extraktion von relevantem Material.

Allgemein gilt: je mehr der Text in der Vorverarbeitung mit Annotationen versehen wird, desto komplexere Anfragen kann man anschließend stellen und umso zielgenauer Anfragen können formuliert werden. In Termland werden folgende Extraktionswerkzeuge benutzt:

- Der Corpusanfrageprozessor *cqp* (vgl. SCHULZE & CHRIST 1994),
- der Makroprozessor *MP* für die *cqp*-Sprache (vgl. SCHULZE 1996)
- und die Morphologieschnittstelle *ims-infl*, aufbauend auf DMOR (SCHILLER 1994).

Zur morphologischen Analyse von Komposita wurde ein *Perl*-Programm geschrieben, das auf der am IMS vorhandenen Morphologieschnittstelle *ims-infl* aufbaut.

3.2 Extraktionsverfahren für Termkandidaten

Es werden zwei Verfahren zur Termextraktion benutzt, ein symbolisches und ein statistisches. Das von JAUSS 1996 entwickelte Termextraktionsverfahren *C-Term* beruht auf der Charakterisierung des Terms durch seine spezifischen morphologischen Komponenten. Auf Grundlage dieser Beschreibungen werden verschiedene Suchmuster formuliert, mit denen im Corpus all die Terme gefunden werden, die dem Suchmuster entsprechen.

Die Suchmuster unterscheiden sich, je nachdem, ob Abkürzungen, Einworttermini oder Mehrworttermini gefunden werden sollen: Abkürzungen werden auf der Grundlage charakteristischer Buchstabenfolgen extrahiert. Mit Hilfe von Präfix- und Suffixlisten sowie mit Listen von *fachspezifischen Bausteinen*[3] werden Einworttermini gefunden, die die gesuchten Morpheme enthalten. Die Suchmuster für Einworttermini werden danach unterschieden, ob sie mit Hilfe von Präfixlisten einen typischen Wortanfang suchen oder mit Suffixlisten ein typisches Wortende, oder ob sie mit einer Liste fachspezifischer Bausteine, die im Wortstamm auftreten können, operieren. Die Extraktion von Mehrworttermini beruht auf einer regulären Grammatik der *charakteristischen Part-of-Speech Shapes*.

[3] Die Liste dieser fachspezifischen Bausteine umfaßt Morpheme, die von JAUSS 1996 auf Grundlage der von ihr verwendeten Texte als typisch für den Automobilbereich identifiziert wurden.

Die alternativen Suchmuster werden als Menge von *CQP*-Anfragen formuliert und können mit Hilfe des Makroprozessors der Anfragesprache als Filter über Texte benutzt werden. Die Ergebnisse einer solchen Anfrage werden in zwei Formaten geliefert: die Termkandidaten mit Kontextbelegen aus dem Corpus, oder, alternativ, die Termkandidaten als Wortliste mit Häufigkeitsangabe, nach absteigender Häufigkeit sortiert.

Neben *C-Term* verwenden wir auch eine Methode zur Termidentifikation durch relative Häufigkeit: das Verfahren nach AHMAD (vgl. AHAMD ET AL. 1992) basiert auf dem Vergleich der relativen Häufigkeit eines Wortes in einem Fachcorpus und in einem allgemeinsprachlichen Corpus. Dabei lautet die These, daß ein Term im Fachcorpus signifikant häufiger ist, als im allgemeinsprachlichen Text. Bei unserer Implementierung des AHMAD-Algorithmus kann man eine variable untere Mindestgrenze für die absolute Häufigkeit eines Wortes angeben.

3.3 Beispiele

3.3.1 Einworttermini

Die Suchanfragen für Einworttermini (und deren Vorkommen in Mehrworttermini) operieren auf der Annahme, daß viele Nominalterme typische Suffixe oder Präfixe enthalten[4]. Die Extraktionsroutinen suchen nach Kandidaten, die eines der Präfixe enthalten (Beispiel 1) und/oder eines der Suffixe, wobei nach Nomina (Beispiel 2) und Adjektiven (Beispiel 3) unterschieden wird.

(1) *ab.+, auf.+, ent.+, ge.+, haupt.+, her.+, kurz.+, mit.+, nach.+, neben.+, neu.+, um.+, ver.+, voll.+, vor.+, zer.+, zu.+, anti.+, bi.+, mega.+, mikro.+, multi.+, radial.+, semi.+, super.+, ultra.+, zentral.+, ad.+, ex.+, in.+, ko.+, pro.+, re.+, sub.+, trans.+.*

(2) *.+chen, .+e, .+el, .+er, .+grad, .+heit, .+nis, .+schaft, .+tum, .+ung, .+werk, .+wesen, .+zeug, .+age, .+ial, .+gramm, .+graph, .+id, .+ik, .+tion, .+tät, .+um, .+iv, .+ix, .+lyse, .+ment, .+anz, .+ul, .+ur, .+ismus, .+ator.*

(3) *.+arm, .+artig, .+bar, .+bedingt, .+dicht, .+echt, .+eigen, .+end, .+fähig, .+fertig, .+förmig, .+frei, .+gleich, .+haft, .+haltig, .+ig, .+isch, .+leer, .+lich, .+los, .+reich, .+sam, .+seitig, .+sicher, .+stark, .+voll, .+weise, .+weit, .+wert, .+widrig, .+ial, .+uell, .+bel, .+tiv.*

In Beispiel 4 werden Termkandidaten gezeigt, die mit den Suffixen aus Beispiel 2 gefunden werden. Typische Beispiele für Noise, das bei dieser Extraktionsmethode entsteht, finden sich in Beispiel 5; die Anfrage produzierte auf einem Corpus mit ca. 35.000 Wortformen ungefähr 8% Noise.

---

[4] Listen solcher Affixe für das Deutsche sind von REINHARDT et al. 1992 veröffentlicht worden. Aus diesen Listen und aus Auszügen aus der Terminologiedatenbank *Interfass* der Mercedes Benz AG hat JAUSS 1996 die Listen entwickelt, die wir in unseren Anfragen benutzen.

(4) *Partikelfilter, Hinterachse, Motorleistung, Außenspiegel, Grenzgeschwindigkeit, Werkzeug, Regeneration, Motormodul, Partikelemission, Sichtkanal, Kohlenmonoxid, Fahrpedal, Potentiometer, übertragungsfunktion, Gesamtsystem, Effizienz, Außentemperatur.*

(5) *Verwendung, Geschichte, Wirklichkeit, Pfadfinder, Leben, Problem, Million, Globus, Novum, Kenntnis, Identität.*

Bei der Termextraktion werden neben Präfix- und Suffixlisten auch Listen mit *fachspezifischen Bausteinen* (≅ Morpheme) benutzt, da die Terme aus einem begrenzten Fachbereich (hier Automobiltechnik) immer wieder auftretende Morpheme aufweisen. Ein paar dieser fachspezifischen Komponenten werden in Beispiel 6 gezeigt.

(6) *.*fahr.*, .*motor.*, .*trieb.*, .*bau.*, .*stoff.*, .*halt.*, .*rad.*, .*kraft.*, .*dreh.*, .*achs.*, .*gas.*, .*elektr.*, .*brems.*, .*system.*, .*steuer.*, .*auto.*, .*lenk.*, .*druck.*, .*techn.*, .*filt.*, .*gang.*, .*reif.*, .*hub.*, .*leit.*, .*schutz.*, .*heck.*, .*start.*, .*spur.*, .*kanal.*, .*tank.*, .*kabel.*, .*park.*.*

3.3.2 Mehrworttermini

Die Part-of-Speech-Shapes, mit denen Mehrworttermini gesucht werden, liefern typischerweise Ergebnisse wie in Beispiel 7. Beispiele für Rauschen, das bei dieser Anfrage auftritt, finden sich in Beispiel 8 .

(7) *bei kurzem Halt, bei angezogener Feststellbremse, durch geringfügige Preiserhöhungen beim Dieselkraftstoff, im städtischen Raum, mit SA Hydrostat, mit vertretbarem Aufwand, ohne ausreichende Gebläsewirkung, von großem Vorteil, zur aktiven Sicherheit, über hochelastische Hohlschnurdichtungen;*

(8) *auf Partikelfilter mit Vollstromregeneration, bei Vollgas am Fahrpedal, um Kratzer und Druckstellen, von PROMETHEUS bei Daimler-Benz*

Das in Beispiel 8 aufgeführte Rauschen zeigt deutlich die Grenzen regulärer Grammatiken: Die Resultate sind nicht immer wohlgeformte phrasale Konstrukte.

4 Systemintegration und Entwurf der graphischen Benutzeroberfläche

4.1 Die Philosophie der Benutzeroberfläche

Bei der Entwicklung des Interfaces stellten wir fest, daß unterschiedliche Benutzer unterschiedliche Eingriffspunkte in den Gesamtablauf der halbautomatisierten terminologischen Arbeit wünschen.

Bei der Bündelung von Teilabläufen in der Anforderungsanalyse wurde klar, daß nicht für jeden Benutzer die Zwischenschritte, die zum Ergebnis einer bestimmten Aufgabe führen, gleich wichtig oder interessant sind. Ähnlich den einzelnen „Fahrstraßen“ bei einem Gleisbildstellwerk wurden unterschiedliche Pfade durch die Gesamtheit der Aufgaben ermittelt, die je nach Benutzer direkt und ohne Unterbrechung zum Ziel führen oder „unterwegs“ mehr oder weniger Eingriffspunkte für den Benutzer bieten:

- Bei einem Übersetzer liegt z. B. aus Zeitgründen der Fokus primär auf der Tatsache, daß die gestellte Aufgabe (z. B. Termextraktion) *erledigt* wird; *wie* ist zunächst irrelevant.
- Für Terminologen hingegen ist es u. U. interessant, an verschiedenen Punkten ins Geschehen eingreifen zu können.
- Für Power User, Linguisten oder Entwickler sind alle Eingriffspunkte ins Geschehen interessant.

Ein weiteres Unterscheidungskriterium zwischen den Benutzern ist, inwieweit sie ihre „Weichen" selber stellen oder lieber automatisch durch das System geschleust werden wollen. Das automatische Durchschleusen ähnelt, sehr bildlich gesprochen, der Durchfahrt durch eine Autowaschanlage: das gewünschte Resultat wird vorher ausgewählt, der Ablauf, wie man dort hingelangt, wurde von jemand anderem vorab festgelegt; man selber kann in den Ablauf nicht mehr eingreifen.

4.2 Funktionalitäten des Gesamtsystems für verschiedene Benutzer

4.2.1 Übersetzer

Für den Übersetzer ist in Termland Hilfe für drei Aufgabenstellungen vorgesehen[5]:

1. Automatische Termextraktion:
   Die automatische Termextraktion umfaßt die Extraktion von wahlweise Einworttermini, Mehrworttermini oder Abkürzungen und als Defaultoption die Extraktion aller Termkandidaten[6].
   Der Benutzer hat die Möglichkeit, aus einer Auswahlliste deutscher Inputtexte auszuwählen. Aus allen verfügbaren Extraktionsmethoden haben wir für jeden der drei Typen von Termini (Einwortterm, Mehrwortterm oder Abkürzung) die optimale ausgewählt. Der unter Zeitdruck stehende Übersetzer profitiert von dieser Vorentscheidung, indem er das beste Verfahren verwenden kann, ohne selber die Verfahren im Detail kennen oder gar testen zu müssen.
2. Bereitstellen von Übersetzungsvorschlägen aus Corpora für eine interaktive Benutzereingabe:
   Der Benutzer wählt hier zunächst ein aligniertes Textpaar aus und kann anschließend in einem Eingabefeld eine Anfrage an dieses Corpuspaar stellen. Der Corpusanfrageprozessor wählt aus der Quellsprache all die Sätze aus, die auf die Anfrage passen. Diese werden tabellarisch mit dem jeweils alignierten Satz der Zielsprache

[5] Wir sind nicht der Ansicht, daß die von uns vorgenommene Verteilung von einzelnen Aufgaben auf die Benutzer endgültig ist. Dies ist ein erster Vorschlag, in dessen Weiterentwicklung auf jeden Fall (Benutzer)-Kritik einfließen sollte.

[6] Dieser Default wurde ausgesucht, da er für den Übersetzer auf die schnellste Art und Weise die größtmögliche Termkandidatenliste liefert. Dies ist normalerweise die vom Übersetzer gewünschte Funktionalität. Die anderen Möglichkeiten werden angeboten, um dem Wunsch eines Benutzers zu entsprechen, der nur einen bestimmten Termtyp sehen möchte (z. B. eben alle Abkürzungen), ohne dabei durch das Anzeigen der anderen Belege „gestört zu werden".

dargestellt. Abbildung 4 im Anhang zeigt z. B. die erste Seite des Ergebnisses einer Anfrage nach dem englischen Wort *manager*.

3. Extraktion von Termkandidaten und anschließende Bereitstellung von Übersetzungsvorschlägen für alle Termkandidaten. Dieser Punkt ist eine Kombination aus Punkt 1. oben und der Corpussuche in alignierten Corpora unter 2.

4.2.2 Terminologe

1. Automatische Termextraktion:
   Die Funktion „Automatische Termextraktion“ umfaßt die Extraktion von wahlweise Einworttermini, Mehrworttermini oder Abkürzungen mit Hilfe unterschiedlicher Suchverfahren, aus denen der Benutzer auswählen kann. Im Unterschied zu der Termextraktionsseite beim Benutzertyp Übersetzer wird hier sehr viel mehr bezüglich der Suchstrategie differenziert. So kann der Terminologe auswählen, mit Hilfe welcher Affixliste er suchen möchte, welche Kategorie der gesuchte Term haben soll etc.
2. Suche nach Corpusbelegen im monolingualen Corpus:
   Hier wird dem Benutzer die Möglichkeit gegeben, nach einsprachigen Textbelegen zu suchen, um Termkandidaten zu verifizieren[7].
3. Bereitstellen von Übersetzungsvorschlägen aus Corpora für eine interaktive Benutzereingabe:
   Das „Terminologen-Fenster“ für alignierte Corpussuche und das entsprechende Fenster im Übersetzermenü sind identisch.
4. Vergleich von Auszügen aus der terminologischen Datenbank des Benutzers (Listen) mit Corpusbelegen:
   Die Termliste des Benutzers wird gegen die Wortliste des ausgewählten Corpus abgeglichen, d. h. der Benutzer erhält eine dreispaltige Ausgabe:
   - All die Termkandidaten, die nur in seiner Datenbank sind,
   - all die Termkandidaten, die nur im Corpus sind und
   - die Termkandidaten, die sowohl in der Datenbank als auch im Corpus gefunden wurden.

4.2.3 Power User

Der Power User interagiert typischerweise mit dem System nicht nur über die Benutzeroberfläche sondern bewegt sich durch eine Art Entwicklungsspirale: er muß bei seiner Arbeit häufig zwischen dem Testen von Programmen auf der Shell und Überle-

[7] Dies geschieht um festzustellen, ob ein Termkandidat überhaupt im Text vorkommt, wenn ja, wie oft und in welchem Kontext. Die Textverifikation ist bei der Glossarerstellung nützlich und kann in einem weiterführenden Schritt auch zur Identifizierung von Wortreihen dienen (z. B. Suche nach allen Termen die mit *-platz* aufhören).

gungen zur Integration der Resultate solcher Tests in eine nächste Version von TermLand abwechseln.

Der Power User ist der Benutzer, der *alle* Funktionalitäten des Systems benutzt und kennt. Zusätzlich zu den Aufgaben der anderen Benutzertypen, die er über deren Seiten anwählen kann, gibt es eine Vielzahl von Aufgaben: Entwicklertätigkeiten jeder Art, z. B. Vorverarbeitung von Daten, Inspektion, Test und Verbesserung der Termextraktionsmethoden durch das Einsetzen weiterer linguistischer Tools. Die Power User Funktionen sind bislang nur prototypisch entwickelt worden, da der Schwerpunkt auf der Bereitstellung der Werkzeuge für Übersetzer und Terminologen lag.

5 Implementierung

5.1 Allgemeines

Zur Implementierung der Benutzeroberfläche wurden HTML-Seiten verwendet, die über `cgi`-Skripte mit den Tools kommunizieren, die die Kernaufgaben erledigen.

5.2 Systemarchitektur

DATENFLUSS **zwischen den einzelnen Modulen**

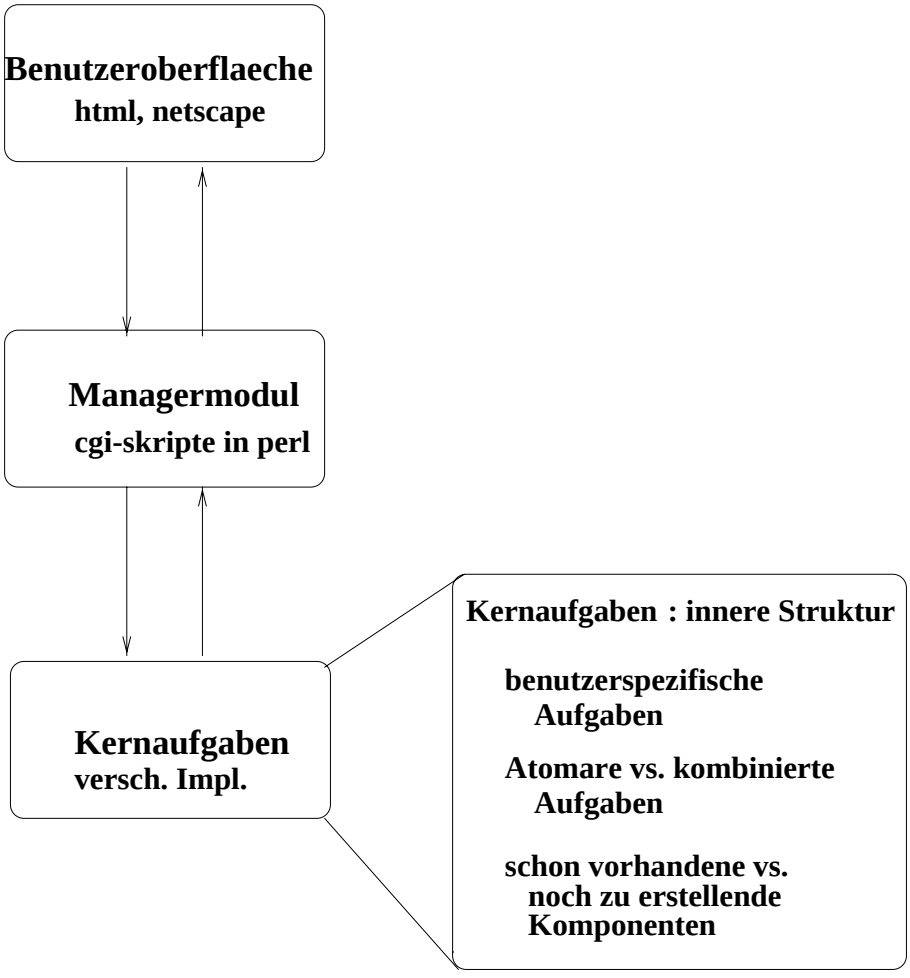


Abb. 1: Systemarchitektur: Datenfluß zwischen den einzelnen Modulen

Die Architektur des Systems teilt sich in drei Module: das Interfacemodul, das Managermodul und das Kernaufgabenmodul (vgl. Abb. 1). Das Interfacemodul beherbergt alle Prozeduren, die für die Präsentation von Information in der Benutzeroberfläche verantwortlich sind, während das Kernaufgabenmodul die Prozeduren zur Erledigung von spezifischen Aufgaben durch ‚externe' Tools, wie etwa *cqp, MP* etc. zur Verfügung stellt. Diese beiden Module kommunizieren nicht direkt miteinander sondern über das Managermodul, dadurch haben die Prozeduren klare Schnittstellendefinitionen und das ganze System wird übersichtlicher und modularer und kann somit auch effizienter weiterentwickelt werden.

Eine Schnittstelle zu Termland ist jeder gängige WWW-Browser; von uns wurde *Netscape* eingesetzt. Eine auf Hypertext basierende Oberfläche erlaubt es, Information so zu strukturieren, daß der Benutzer interaktiv entscheiden kann, welchen Detailliertheitsgrad an Information er angezeigt haben will. Benutzerspezifische Eingaben erlauben die Parametrisierung einer Anfrage an das System; auf dieselbe Weise kann der Benutzer auch das Darstellungsformat seiner Ergebnisse im Browser beeinflussen.

In HTML-Seiten kann allerdings nur statische Information abgelegt werden. *CGI*, das *Common Gateway Interface*, ist eine Möglichkeit, ein Programm über eine Web-Seite mit *benutzerspezifischen Parametern* aufzurufen und das Resultat dieses Programms im Browser darzustellen. Ein Nachteil bei der Benutzung von *CGI* ist, daß die Benutzereingaben nicht im Client sondern erst im Server verarbeitet werden. Das kann zu Verzögerungen und unnötiger Belastung des Kommunikationskanals zwischen Client und Server führen. Eine Möglichkeit, um Benutzereingaben *clientseitig* zu prüfen und damit Zeit und unnötige Kommunikationswege zu sparen, ist, die Benutzung der Skriptsprache *JavaScript*. Damit lassen sich innerhalb von HTML-Seiten Funktionen definieren, deren Aufruf vor dem Aufruf eines CGI-Skriptes z. B. Corpusanfragen oder andere Parameter auf Korrektheit und Konsistenz überprüft.

6 Ergebnisse

Termland ist ein System zur computerunterstützten Glossarerstellung und ermöglicht zwar noch lange keine *vollautomatische* Glossarerstellung, versucht jedoch Ergebnisse der automatischen Termextraktion so weit wie möglich für den Übersetzer oder Terminologen zu erschließen. Zum einen werden sinnvolle Termkandidaten zur manuellen Auswahl bereitgestellt, zum anderen kann die Gesamtheit der elektronisch vorliegenden, schon übersetzten Texte während des Übersetzens als „Online-Nachschlagewerk" für den Übersetzer dienen.

Abb. 2 zeigt die Ergebnisse der Termextraktion mit *C-Term*, (P = Präfixe, S = Suffixe, N = Nomina, ADJ = Adjektive, V = Verben, FB = Fachspezifische Bausteine).

	Anzahl	**Noise**	**Beispiele:** akzeptable Termkandidaten	**Beispiele:** Noise
Einfache Anfragen:				
P + N	2990	13,5%	Hinterachse, Vollgas	Bedeutung, Gegensatz
P + ADJ	1755	46%	gefiltert, verstellbar	unabhängig, vorhanden
P + V	1603	31%	beschleunigen, einspritzen	beibehalten, entscheiden
S + N	6349	8%	Geschwindigkeit, Produktion	Kenntnis, Wirklichkeit
S + ADJ	2674	29,5%	bleifrei, lieferbar	gemeinsam, unmittelbar
FB + N	5376	2%	Dachluke, Motorbremse	Nachteil, Umgang
FB + ADJ	786	22%	thermodynamisch, betriebswarm	nachhaltig, nutzlos
FB + V	585	29%	einbauen, herausfiltern	verkraften, aufteilen
Abbr.	783	1%	km/h/s, FCKW-frei, TE-24	MANN, BOSCH-Einspritzpumpe
Kombinierte Anfragen:				
P + S + N	2270	13%	Sonderausstattung, Verstellhebel	Verhältnis, Beschreibung
P + S + ADJ	1508	51%	einstellbar, hochporös	entsprechend, abhängig
P + FB + N	1713	6,5%	Leergas, Innenraumluft	Aufwand, Vorteil
P + FB + ADJ	371	28%	hochbelastet, ungefiltert	vorhanden, entsprechend
P + FB + V	469	28%	zuschalten, ansaugen	bewirken, erhalten
S + FB + N	4233	2,5%	Lasersensor, Kraftübertragung	Erteilung, Schimmelbildung
S + FB + ADJ	747	20,5%	elektronisch, nachgeschaltet	typisch, wirkungsvoll
P + S + FB + N	1318	4%	Abgasleitung, Feststellbremse	Erfahrung, Aufteilung
P + S + FB + ADJ	363	24%	vollelektronisch, hochbelastbar	ermittelt, vorausgegangen

Abb. 2: Statistiken und Beispiele für Resultate der Termextraktion

7 Anhang: Bildschirmabzüge aus Termland

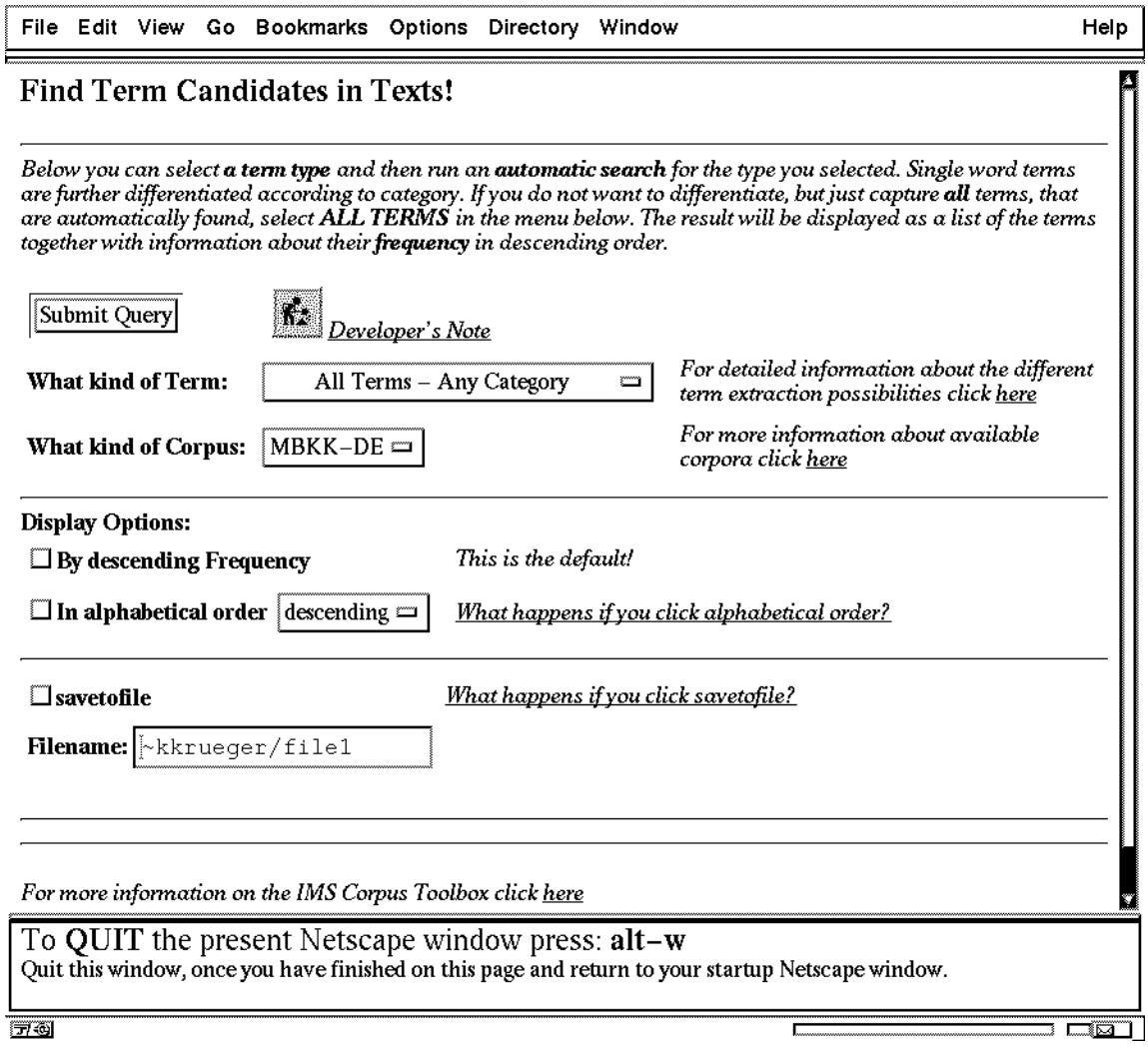
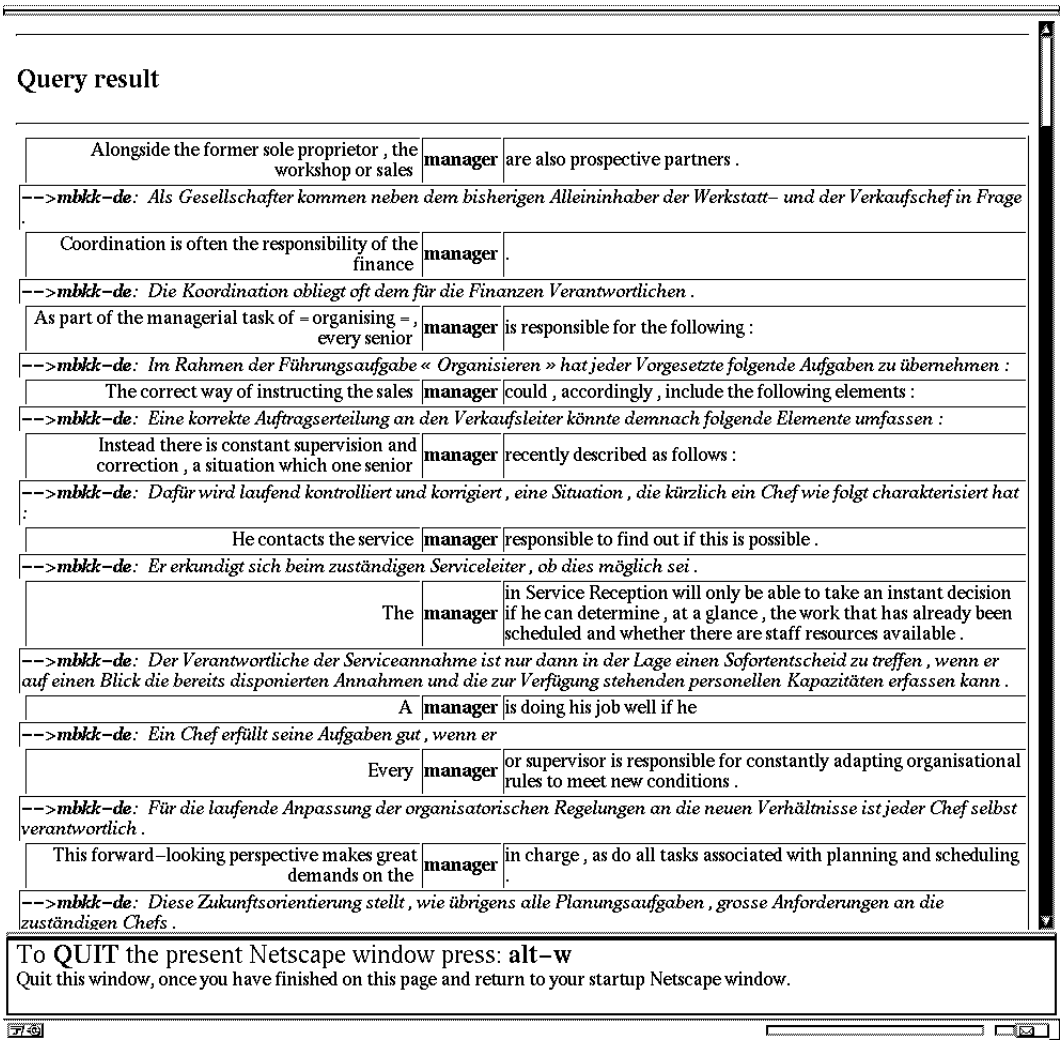


Abb. 3: Termextraktion aller Termkandidaten über das Interface Termland



Query result

Alongside the former sole proprietor , the workshop or sales	**manager**	are also prospective partners .
*-->**mbkk-de**: Als Gesellschafter kommen neben dem bisherigen Alleininhaber der Werkstatt- und der Verkaufschef in Frage .*		
Coordination is often the responsibility of the finance	**manager**	.
*-->**mbkk-de**: Die Koordination obliegt oft dem für die Finanzen Verantwortlichen .*		
As part of the managerial task of = organising = , every senior	**manager**	is responsible for the following :
*-->**mbkk-de**: Im Rahmen der Führungsaufgabe « Organisieren » hat jeder Vorgesetzte folgende Aufgaben zu übernehmen :*		
The correct way of instructing the sales	**manager**	could , accordingly , include the following elements :
*-->**mbkk-de**: Eine korrekte Auftragserteilung an den Verkaufsleiter könnte demnach folgende Elemente umfassen :*		
Instead there is constant supervision and correction , a situation which one senior	**manager**	recently described as follows :
*-->**mbkk-de**: Dafür wird laufend kontrolliert und korrigiert , eine Situation , die kürzlich ein Chef wie folgt charakterisiert hat :*		
He contacts the service	**manager**	responsible to find out if this is possible .
*-->**mbkk-de**: Er erkundigt sich beim zuständigen Serviceleiter , ob dies möglich sei .*		
The	**manager**	in Service Reception will only be able to take an instant decision if he can determine , at a glance , the work that has already been scheduled and whether there are staff resources available .
*-->**mbkk-de**: Der Verantwortliche der Serviceannahme ist nur dann in der Lage einen Sofortentscheid zu treffen , wenn er auf einen Blick die bereits disponierten Annahmen und die zur Verfügung stehenden personellen Kapazitäten erfassen kann .*		
A	**manager**	is doing his job well if he
*-->**mbkk-de**: Ein Chef erfüllt seine Aufgaben gut , wenn er*		
Every	**manager**	or supervisor is responsible for constantly adapting organisational rules to meet new conditions .
*-->**mbkk-de**: Für die laufende Anpassung der organisatorischen Regelungen an die neuen Verhältnisse ist jeder Chef selbst verantwortlich .*		
This forward-looking perspective makes great demands on the	**manager**	in charge , as do all tasks associated with planning and scheduling .
*-->**mbkk-de**: Diese Zukunftsorientierung stellt , wie übrigens alle Planungsaufgaben , grosse Anforderungen an die zuständigen Chefs .*		

Abb 4: Ergebnis einer Corpusanfrage nach EN manager in alignierten Corpora

Markus Schulze, Universität Erlangen-Nürnberg

Morphologie, Syntax und Semantik im Rahmen der linksassoziativen Grammatik

8 Einleitung

Dieser Artikel stellt MALAGA vor: ein System zur Entwicklung linksassoziativer Grammatiken (LAGs) für die morphologische und syntaktisch-semantische Analyse. Dazu werden in Abschnitt 2 zunächst die allgemeinen Eigenschafen von LAGs erläutert, soweit dies für das weitere Verständnis notwendig ist. In Abschnitt 3 werden die allgemeinen Eigenschaften und die Bestandteile des Systems MALAGA vorgestellt, bevor dann die Möglichkeiten von MALAGA in den folgenden drei Abschnitten 4-6 anhand von Beispielgrammatiken ausführlicher demonstriert werden.

Das erste Beispiel in Abschnitt 4 ist dabei die DMM, eine Morphologie-Grammatik für das Deutsche zur Lemmatisierung und Kategorisierung von Wortformen aus freien Texten. An diesem System sind die Kompaktheit des Regelsystems sowie die hohe Abdeckungsrate beispielsweise des LIMAS-Korpus besonders interessant.

Als Beispiel für eine Syntax-Grammatik wird in Abschnitt 5 eine Grammatik vorgestellt, die in der Lage ist, komplexe Nominalphrasen aus freien Texten zu analysieren. Bei dieser Grammatik ist in erster Linie der Aspekt der Robustheit und der Hypothesenbildung interessant.

Abschließend zeigt der Abschnitt 6 eine in einer Lehrveranstaltung entwickelte Grammatik zur syn-semantischen Analyse einfacher Aussagesätze. Diese Grammatik wurde im Rahmen eines Proseminars mit Studenten ohne Vorkenntnisse in Grammatikentwicklung erstellt.

MALAGA ist für nichtkommerzielle Forschungszwecke frei verfügbar, in der Form von Quellcode und vorkompilierten Binaries für die Plattformen HP-UX/HP 9000 Serie 700 sowie Linux/Pentium-PCs. Die entsprechenden Daten, wie auch die MALAGA-Dokumentation vom MALAGA-Entwickler (BEUTEL 1997) ist auf dem WWW unter *http://www.linguistik.uni-erlangen.de/Malaga.de.html* zu finden.

9 Skizzierung der LAG

Dieser Abschnitt beschreibt die wichtigsten Eigenschaften der LAG. Eine genauere Beschreibung der LAG ist in HAUSSER 1989 zu finden.

Die linksassoziative Grammatik (LAG) arbeitet nach dem *Prinzip der möglichen Fortsetzungen*: Beginnend mit einem leeren Satzanfang, dem ersten Wort und einem *initialen Regelpaket* wird der jeweils aktuelle *Satzanfang* mit dem aktuellen *nächsten Wort* gemäß den Regeln des aktuellen Regelpakets zu einem *neuen Satzanfang* verbunden. Eine Regel aktiviert dabei ein neues Regelpaket, das diejenigen Regeln aufli-

stet, die im nächsten Analyseschritt – also dem Anfügen des neuen *nächsten Wortes* an den Satzanfang – verwendet werden können.

Schematisch läßt sich die Anwendung der Regel r_i auf einen Satzanfang, repräsentiert durch seine Kategorie KAT_{SA}, und das zugehörige nächste Wort, repräsentiert durch seine Kategorie KAT_{NW}, wie folgt darstellen:

$$KAT_{SA} + KAT_{NW} \xrightarrow{r_i} KAT_{SA'}, rp_i \quad (1)$$

Die Oberflächen des *Satzanfangs* und des *nächsten Wortes* werden dabei zur Oberfläche des *neuen Satzanfangs SA'* konkateniert, während die Kategorie des neuen Satzanfangs $KAT_{SA'}$ durch die Anwendung der kategorialen Operation der Regel r_i erzeugt wird. Der neue Satzanfang kann dann wiederum mit Hilfe der Regeln des zu r_i gehörenden Regelpakets rp_i mit dem nächsten Wort verbunden werden. Für die Morphologie gilt das entsprechende für *Wortanfang* und *nächstes Allomorph*. Morphologische und syntaktische Grammatiken arbeiten mit denselben Prinzipien und Beschreibungsmitteln.

Die LAG beruht auf dem *Prinzip der möglichen Fortsetzungen* im Gegensatz zu dem Prinzip der möglichen Substitutionen, das die Grundlage der Phrasenstruktur-Grammatik (PSG) und der Kategorial-Grammatik (CG) bildet. Dieses Prinzip führt – anders als bei der PSG oder der CG – zu regelmäßigen Ableitungsbäumen, wie in der untenstehenden Abbildung 1 zu sehen ist. Die regelmäßige Baumstruktur veranschaulicht die *zeitlineare* Grundstruktur der LAG: Eine Eingabe wird von links nach rechts fortschreitend – also genau in der Äußerungs- und Rezeptionsreihenfolge – analysiert.

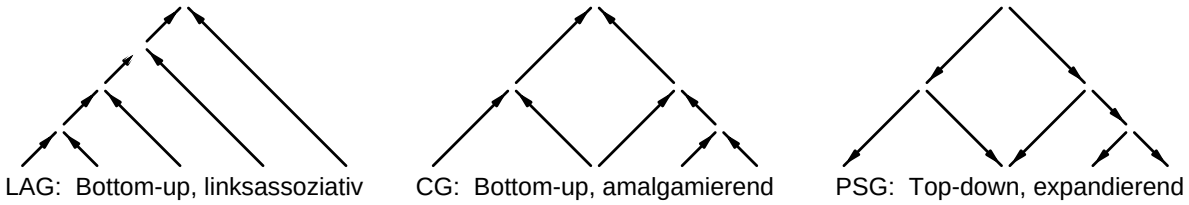


Abbildung 1: Ableitungsbäume der LAG, der CG und der PSG.

Weiterhin hat die Verwendung des Prinzips der möglichen Fortsetzungen zur Folge, daß die LAG eine Komplexitätshierarchie bildet, die orthogonal zur Chomsky-Hierarchie liegt. So lassen sich beispielsweise viele in der Chomsky-Hierarchie kontextsensitive Sprachen (z. B.: $a^n b^n c^n$, a^{2^n}) in der LAG in linearer Zeit parsen (siehe HAUSSER 1992).

Eine weitere wichtige Eigenschaft der LAG ist die Typentransparenz, die BERWICK und WEINBERG 1984 (2. Kap.) erstmals als methodologisches Prinzip beschreiben. Typentransparenz bedeutet, daß sich die Organisation einer Grammatik und die logische Struktur ihrer Regeln in der logischen Struktur des Parsers genau widerspiegeln. Zwischen LAGs und den zugehörigen Parsern besteht dadurch *Ein-/Ausgabeäquivalenz*: Parser wenden die Regeln der Grammatik direkt und in der gleichen Reihenfolge wie die Grammatik an und benötigen keine Zwischenstrukturen (Tabellen, *charts* o.ä.).

LAGs arbeiten mit der sukzessiven Konkatenation kategorisierter Oberflächen. Die sich daraus ergebende zeitlineare Ableitungsordnung ist dahingehend kognitiv adäquat, daß sie bei der Analyse (und Generierung) der linearen Produktions- und Rezeptionsreihenfolge natürlicher Sprache folgt. Die Zeitlinearität ist dem Modell der LAG inhärent, während sie bei der PSG nur mittelbar durch Verwendung spezieller Algorithmen zu erreichen ist.

10 Skizzierung von MALAGA

MALAGA[1] ist ein System zur Entwicklung linksassoziativer Grammatiken, das an der Abteilung für Computerlinguistik der Universität Erlangen-Nürnberg entwickelt wurde. Dieses System besteht aus einer Sprache zur Implementierung von Morphologie- und Syntax-Grammatiken, einem Compiler für diese Sprache, einem System zur graphischen Ausgabe der Analyse und ihres Ergebnisses, einer Entwicklungsumgebung, die unter anderem einen Debugger beinhaltet, Funktionsbibliotheken (*C, Perl*) zum Einbinden von MALAGA in andere Anwendungen sowie einer Benutzerschnittstelle zur interaktiven Verwendung von MALAGA. Weiterhin existiert der sogenannte LASP-Style, ein LaTeX-Package zur Darstellung attribuierter LAGs.

Das Design von MALAGA orientiert sich primär an den Bedürfnissen der Analyse natürlicher Sprache, ist jedoch ebenso zur Implementierung formaler Grammatiken geeignet.

10.1 Die Grammatikentwicklungssprache MALAGA

Dieser Abschnitt gibt eine kurze Beschreibung der MALAGA-Sprache. Eine ausführlichere Beschreibung findet sich in der auf dem WWW verfügbaren Dokumentation (BEUTEL 1997). Die MALAGA-Sprache zur deklarativen Grammatik-Implementierung ist speziell dazu entwickelt, auf die Kategorien von Satzanfang und nächster Wortform zuzugreifen und eine Kategorie für den neuen Satzanfang zu konstruieren.

Die grundlegenden Werte der MALAGA-Sprache sind *Symbole*, die zur Bezeichnung von Kategorien oder Subkategorien verwendet werden, *Zeichenketten*, die beispielsweise zur Repräsentation von konkreten Oberflächen oder Lemmata benutzt werden, und *Fließkommazahlen*, die vor allem bei der Gewichtung Anwendung finden. Diese einfachen Werte können zu den *strukturierten Werten* Liste oder Verbund kombiniert werden.

Die MALAGA-Sprache ist nicht typisiert: Die Struktur von Werten kann frei definiert werden und jeder Variablen kann jeder beliebige Wert zugewiesen werden. Weiterhin ist in der MALAGA-Sprache das Konzept der Anweisungsblöcke realisiert, das sich insbesondere auf die Gültigkeitsbereiche von Variablen auswirkt. Neben den frei

[1] Das Akronym MALAGA steht für: MALAGA akzeptiert linksassoziative Grammatiken mit Attributen.

definierbaren Variablen existieren einige vordefinierte Variablen, die beispielsweise die Kategorie des aktuellen Satzanfangs und des nächsten Wortes enthalten, sowie benutzerdefinierbare komplexe Konstanten, die sich insbesondere zum Anlegen von Tabellen eignen.

Die Inhalte von MALAGA-Variablen können untereinander und mit Konstanten verglichen und auf Kongruenz geprüft sowie durch Anweisungen verändert werden. Weiterhin verfügt MALAGA über strukturierte Kontrollanweisungen, die eine Anweisungsfolge auswählen oder wiederholen sowie eine Aufspaltung des Analysepfades erzeugen können. Dabei sind die Anweisungen der MALAGA-Sprache frei von Seiteneffekten. Die verschiedene Analysepfade werden unabhängig voneinander verfolgt. MALAGA-Grammatiken terminieren immer; das Erzeugen von Endlosschleifen oder Rekursionen ist unmöglich.

Im Rahmen von MALAGA wird auf das Konzept der Unifikation von Attribut-Werte-Strukturen verzichtet. Stattdessen bietet die MALAGA-Sprache die effizientere Möglichkeit, Attribut-Werte-Strukturen explizit zu kreieren oder zu verändern. Des weiteren existieren Operatoren, die es ermöglichen, aus bestimmten Werten reduzierte (weniger komplexe) Werte zu erzeugen.

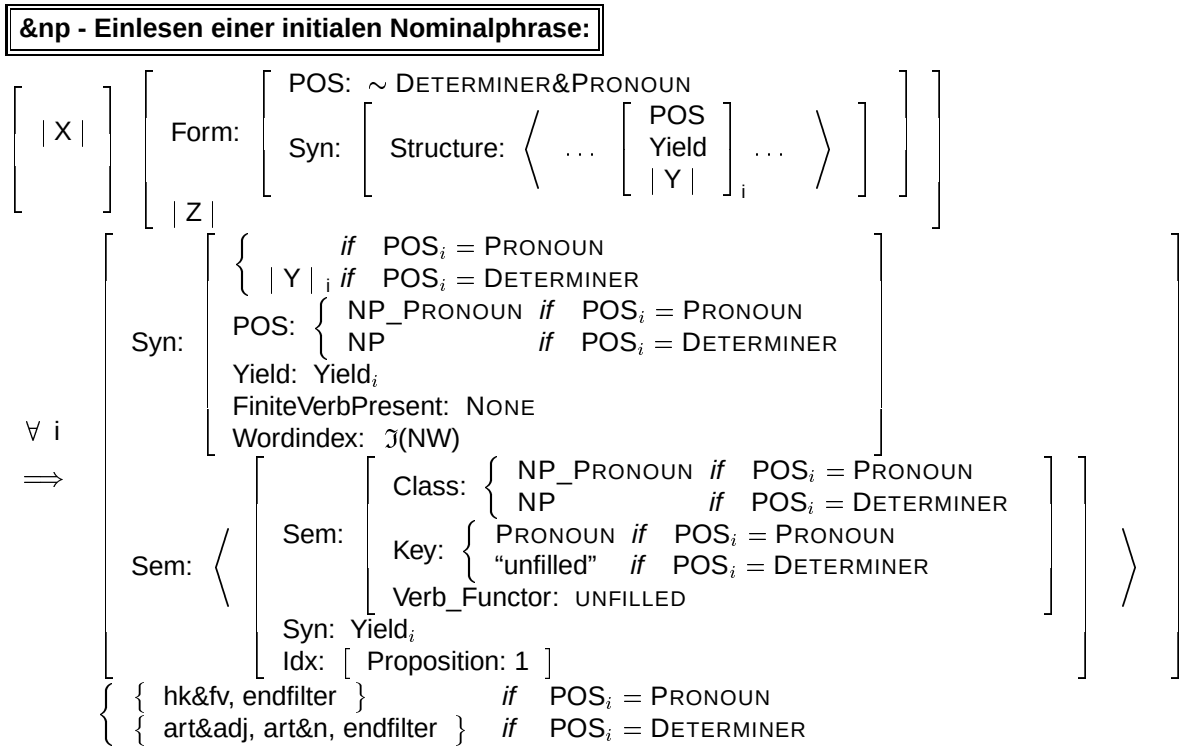


Abbildung 2: Die Regel &np aus DS3 mit Hilfe des LASP-Styles formatiert

Um einen Eindruck der MALAGA-Sprache zu geben, zeigen die Abbildungen 2 und 3 eine Regel der Grammatik DS3 (siehe auch Abschnitt 6), einmal mit Hilfe des LASP-Styles formatiert und einmal als Quelltext. Die ausgewählte Regel liest am Satzanfang stehende einfache (Pronomina) oder komplexe Nominalphrasen ein. In der LASP-formatierten Regel ist die prinzipielle Struktur einer Regeln zu erkennen: Ein Satzanfang und ein nächstes Wort werden auf einen neuen Satzanfang und ein neues Regel-

paket abgebildet. Der alte Satzanfang und das nächste Wort werden dabei durch Bedingungen repräsentiert, die ihre Kategorien erfüllen müssen, damit die Regel angewendet werden kann. Während für den Satzanfang keinerlei Beschränkung gilt (die Ausdrücke |X|, |Y| und |Z| stehen für beliebige Inhalte), ist für das nächste Wort die Wortklasse Determinativ oder Pronomen vorgeschrieben, wobei ein Teil der Struktur der Kategorie eines Determinativs oder Pronomens angegeben ist, um bei der Spezifikation des neuen Satzanfangs auf die Werte der angegebenen Attribute referenzieren zu können.

Die von der DS3 verwendete Morphologie-Grammatik DMM liefert in dem Analyseergebnis für ein Artikel oder Pronomen unter dem Attribut `Structure` eine Liste der verschiedenen Verwendungsarten der analysierten Oberfläche. Für den Artikel der sind dies beispielsweise die Verwendung als maskuliner Artikel im Nominativ Singular, als femininer Artikel im Genitiv oder Dativ Singular, als Artikel im Genitiv Plural oder die pronominale Verwendung.

Die vorliegende Regel `&np` erzeugt für jede dieser Verwendungsarten einen eigenen Analysepfad. Diese Pfade werden von MALAGA (quasi-)parallel verfolgt, wobei die Pfade der nicht zutreffenden Verwendungsarten nach wenigen Schritten als ungrammatisch erkannt werden. Die Erzeugung der verschiedenen Pfade für die einzelnen Verwendungsarten wird in der LASP-Formatierung mit Hilfe des Index *i* und dem Symbol $\forall i$ über dem Pfeil zur Kategorie des neuen Satzanfangs dargestellt.

Hinter dem Ergebnispfeil ist die Gestalt der neuen Satzanfangskategorie dargestellt, die sich bei Anwendung der Regel ergibt. Der von der Regel zu konstruierende Verbund ist dabei zum Teil explizit angegeben, zum Teil werden Werte bestimmter Attribute des nächsten Wortes übernommen, und zum Teil hängen Werte von Fallunterscheidungen ab. Durch Fallunterscheidungen bestimmt Werte sind durch eine öffnende geschweifte Klammer vor den Alternativen und die zugehörigen nachgestellten Bedingungen (Schlüsselwort `if`) dargestellt. In der vorliegenden Regel wird das neue Regelpaket ebenfalls über eine Fallunterscheidung bestimmt.

In der Quelltextversion dieser Regel in Abbildung 3 können die einzelnen Schritte auf der Ebene der MALAGA-Sprache verfolgt werden. Die Regel testet zunächst (2), ob das aktuelle nächste Wort ein Determinativ oder Pronomen ist. Anschließend (3) wird eine Verzweigung über die verschiedenen Verwendungsarten (s. o.) des Determinativs durchgeführt. Die Anweisung `CHOOSE` erzeugt dabei einen neuen Analysepfad für jede der Verwendungsarten, die in der Liste `structure` enthalten sind. Die entstehenden Pfade unterscheiden sich lediglich in der Belegung der Variablen `$DetPron`, die durch die `CHOOSE`-Anweisung für jeden Pfad mit einer neuen Verwendungsart belegt wird.

Die beiden folgenden Anweisungen (4,5) erzeugen den Syntax- und den Semantikteil der resultierenden Verbundstruktur. Dabei werden diese Teilstruktur zunächst so angelegt, wie es für die Verarbeitung für Determinativa notwendig ist.

In der folgenden Fallunterscheidung (6-14) wird danach unterschieden, ob die aktuell untersuchte Verwendungsart einem Pronomen (7-10) oder einem Determinativ

(12,13) entspricht. Handelt es sich um eine pronominale Verwendungsart, so werden der Syntax- und der Semantikteil entsprechend angepaßt (7-9), danach wird ein Ergebnis generiert indem der Syntax- und der Semantikteil zusammengefügt werden und ein Regelpaket angeben wird (10).

Im Fall einer Verwendung als Determinativ wird der Syntaxteil um die in der Kategorie dieser Verwendungsart enthaltene kombinatorische Information erweitert (12). Diese Information wird benötigt, um eine korrekte Konkatenation der noch folgenden Bestandteile der begonnenen Nominalphrase sicherzustellen. In der folgenden Zeile (13) wird dann ein neuer Satzanfang aus dem Syntax- und dem Semantikteil erzeugt und ein Regelpaket angegeben.

```
COMBI_RULE &np;
  START $SentStart, NEXT $Word, INDEX $Index, SURF $OFl;          (1)

  ? $Word.Form.POS ~ Determiner&Pronoun;                          (2)

  CHOOSE $DetPron IN $Word.Form.Syn.Structure;                    (3)

  $Syn := [POS:               NP                                  (4)
           Yield:             $DetPron.Yield
           Wordindex:         $Index,
           FiniteVerbPresent: None];
  $Sem := [Sem: [Class:        NP,                                (5)
                 Key:          "unfilled"
                 Verb_Functor: "unfilled"],
           Syn: $DetPron.Yield,
           Idx: [PropositionIndex: 1,
                 Wordindex:        $Index]]

  IF $DetPron.POS = Pronoun THEN                                  (6)
    SET $Syn.POS       := NP_Pronoun;                             (7)
    SET $Sem.Sem.Class := NP_Pronoun;                             (8)
    SET $Sem.Sem       += [Key: Pronoun];                         (9)
   RESULT [Syn: $Syn, Sem: <$Sem>], RULES hk&fv, endfilter;      (10)
  ELSEIF $DetPron.POS = Determiner:                              (11)
    SET $Syn := $DetPron + $Syn;                                 (12)
    RESULT [Syn:$Syn,Sem:<$Sem>],RULES art&adj,art&n,endfilter; (13)
  END IF;                                                        (14)
END COMBI_RULE;
```

Abb. 3: Der Quelltext der Regel &np aus DS3

10.2 Logischer Aufbau des MALAGA-Compilers

In der Abbildung 4 ist der innere Aufbau einer MALAGA-Morphologie-Grammatik schematisch dargestellt. Zunächst wird zwischen der Ebene des Quelltextes und der Ebene der Laufzeitumgebung unterschieden. Der Grammatik-Entwickler schreibt auf der Quelltextebene des MALAGA-Systems sein Grammatik-System, das ein Lexikon aus kategorisierten Grundformen sowie Flexions- und Wortbildungsmorphemen, Regeln zur Ableitung der Allomorphe aus den Grundformen, Regeln zur Konkatenation

der Allomorphe bei der Analyse von Wortformen und eine Liste der dabei verwendeten Symbole sowie ein Verzeichnis aller zur Grammatik gehörenden Dateien – die sogenannte Projektdatei – umfaßt. Hier wird noch einmal deutlich, daß MALAGA gemäß dem LA-Morph-Ansatz die Allomorphe vor der Laufzeit ableitet und während der Laufzeit nur noch auf diese Allomorphe und nicht mehr auf die Grundformen zurückgreift.

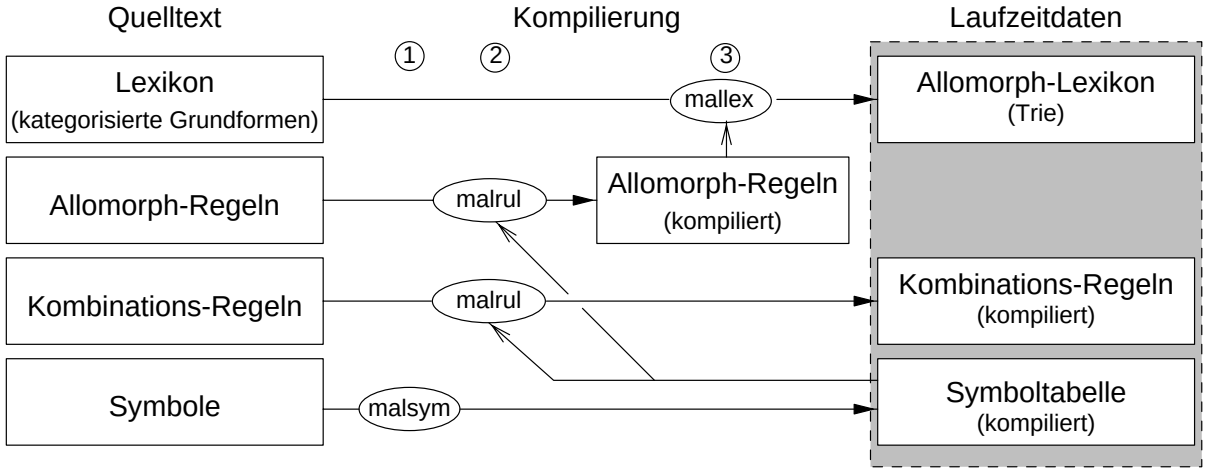


Abb. 4: Schematischer Aufbau von MALAGA.

Die Quelldateien werden dann durch einen dreistufigen Kompilierungsprozeß in ein internes Format übersetzt, das zur Laufzeit von MALAGA verwendet wird. Mit Hilfe der Projektdatei und des Programms `malmake` wird die Kompilierung aus Sicht des Grammatikentwicklers zu einem einstufigen Prozeß vereinfacht, der in Abbildung 5 dargestellt ist.

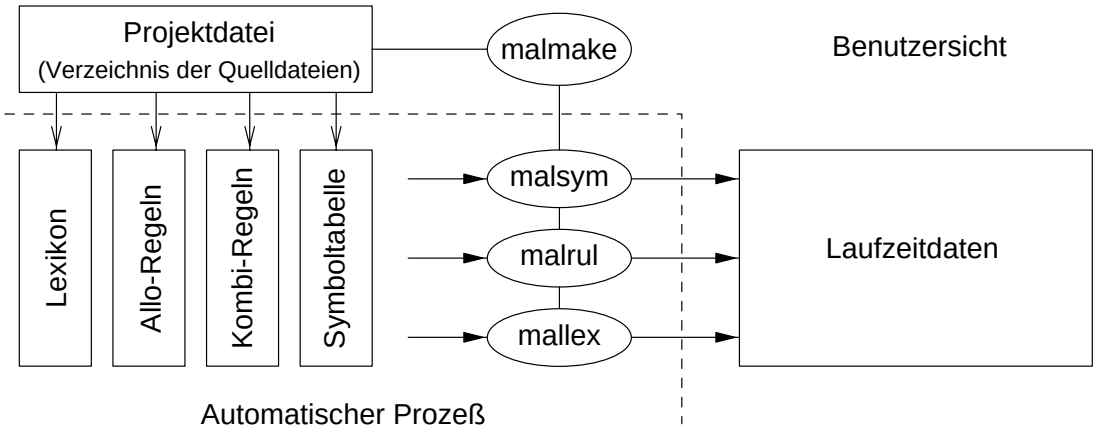


Abb. 5: Benutzersicht und interner Prozeß der MALAGA-Kompilierung.

Bei Syntax-Grammatiken kommt zu dem angeführten noch eine Datei mit syntaktischen Kombinationsregeln hinzu, wobei die Morphologie-Grammatik gegenüber der Syntax modular gestaltet ist. Durch den Einsatz eines in der MALAGA-Sprache geschriebenen Filters kann die Ausgabe der Morphologie umstrukturiert werden, so daß eine Syntax-Grammatik prinzipiell mit verschiedenen Morphologie-Grammatiken unabhängig vom genauen Ausgabeformat verwendet werden kann. Sofern die von einer Morphologie gelieferte Kategorisierung für die Belange der Syntax vollständig ist,

kann die Kategorie einer analysierten Wortform durch einen entsprechenden Ausgabefilter in das benötigte Format gebracht werden.

10.3 Die interaktive Benutzerschnittstelle mit graphischer Darstellung

Die interaktive Benutzerschnittstelle stellt Befehle zur Verfügung, um einzelne Worte oder Sätze sowie ganze Textdateien morphologisch oder syntaktisch zu analysieren oder das vollständige Paradigma zu einer Grundform zu generieren. Weiterhin kann das Ausgabeformat über verschiedene Befehle konfiguriert werden: So können beispielsweise einzelne Attribute der Kategorien ausgeblendet, das Format der Ausgabe bei der Dateianalyse verändert oder Ausgabefilter aktiviert werden.

Ein weiteres wichtiges Werkzeug der Benutzerschnittstelle ist der Grammatik-Debugger. Da LAGs typentransparent sind, entspricht jeder Verarbeitungsschritt des Parsers direkt einer Anweisung der Grammatik. Der Debugger erlaubt es, die Grammatik schrittweise abzuarbeiten und dabei die Kategorien und Variableninhalte zu betrachten, um so Fehler oder Lücken in der Grammatik leichter aufspüren zu können. Der Debugger verwendet dabei zur Anzeige des aktuellen Stands der Analyse das weiter unten beschriebene Modul zur graphischen Ausgabe, wodurch sich insbesondere die Möglichkeit ergibt, mit dem Debugger Analysezustände aus der graphischen Darstellung des Analysebaums direkt anzusteuern.
Die Abbildung 6 zeigt einen Screenshot der graphischen Ausgabe von MALAGA. Die weiteren Abbildungen von Analysebäumen und Ergebnissen sind dann mit Hilfe der Möglichkeit zum Erzeugen von Postscript-Dateien erstellt, die die graphische Ausgabe bietet.

Das linke Fenster in der Abbildung 6 zeigt den MALAGA-Analysebaum, der bei der morphologischen Analyse der Oberfläche Ausgabemodul mit Hilfe der DMM (siehe Abschnitt 4) entsteht. Die verschiedenen Pfade von der Wurzel des Analysebaums (links) zu seinen Blättern (rechts) sind Analysepfade, die von der DMM (quasi-) parallel verfolgt wurden. Parallele Analysepfade entstehen, wenn mehrere mögliche nächste Allomorphe gefunden werden (im Beispiel das Substantiv Au und das Präfix Aus am Wortanfang). Weiterhin können aus einem Satzanfang durch das Anfügen eines Allomorphs mittels verschiedener Regeln mehrere Satzanfänge mit jeweils unterschiedlichen Fortsetzungsmöglichkeiten erzeugt werden.

Über den einzelnen Pfadstrecken ist jeweils die Oberfläche des eingelesenen Allomorphs vermerkt, darunter die Regel, die dieses Allomorph mit dem Satzanfang zu dem durch den folgenden Knoten dargestellten neuen Satzanfang verbunden hat. Kann ein Satzanfang von keiner Regel mit einem möglichen nächsten Morphem verbunden werden, so bricht der entsprechende Pfad ab (dargestellt durch ein kleines gefülltes Quadrat).

In der Abbildung 6 ist dies beispielsweise im zweiten Segment des ersten Pfades zu sehen. Das erste gefunden Allomorph ist in diesem Pfad das Substantiv Au, das näch-

ste das Flexions- und Kompositionsallomorph s. Der Satzanfang Au kann mit diesem Allomorph jedoch nicht verbunden werden, also bricht der Pfad an dieser Stelle ab.

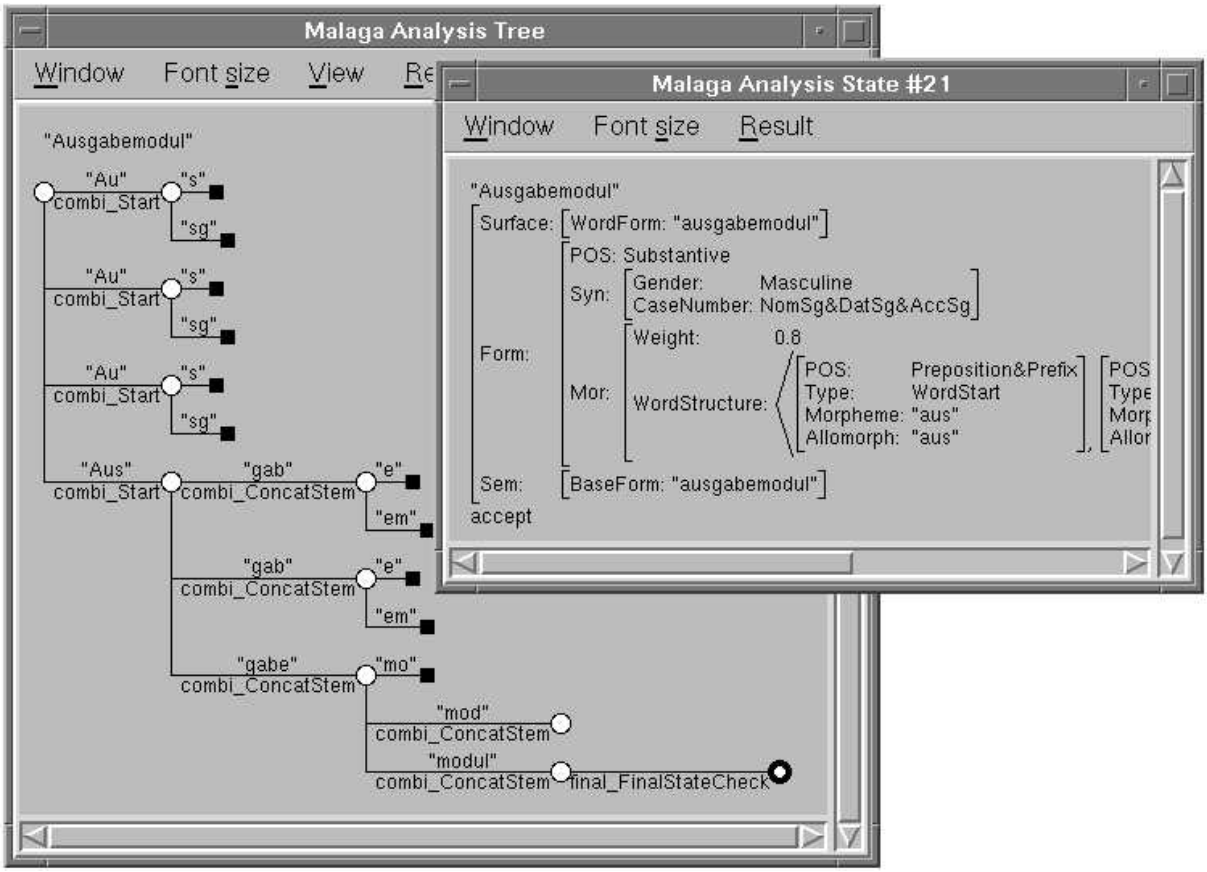


Abb. 6: Screenshot der graphischen Analysedarstellung von MALAGA

Die Konkatenation des Präfixes Aus mit dem Substantiv gabe und dem weiteren Substantiv modul führt dagegen zum Erfolg und passiert auch die abschließende Endzustandsüberprüfung (`FinalStateCheck`), ersichtlich an dem Doppelkreis am Ende des Analysepfades.

MALAGA erlaubt es, die Kategorien jedes einzelnen Zustands des Analysebaums sowie der eingelesenen Allomorphe zu betrachten. Durch Anklicken eines Zustandes oder eines Allomorphs wird ein zweites Fenster geöffnet, das die Kategorie des angewählten Zustands bzw. Allomorphs zeigt – wie das rechte Fenster in der Abbildung 6, das die Kategorie des Analyseergebnisses zeigt.

Die graphische Darstellung der Analyseergebnisse ist momentan mit dem `Tcl/Tk`-System realisiert, wird aber in *Java* reimplementiert.

11 DMM – Deutsche MALAGA-Morphologie

Die deutsche MALAGA-Morphologie DMM wurde von Oliver LORENZ (LORENZ 1997) entwickelt und folgt in ihrem logischen Aufbau dem LA-Morph-Ansatz (siehe HAUSSER 1998). Kennzeichnend für diesen Ansatz ist das Prinzip der Oberflächenkompositionalität. Die Kombinationsregeln der DMM konkatenieren konkrete Oberflächen (Allomorphe). Die Allomorphe werden dabei vor der Laufzeit aus dem Lexi-

kon der kategorisierten Grundformen generiert und während der Analyse durch die morphologischen Kombinationsregeln konkateniert.

Die wesentlichen Bestandteile der DMM (bzw. einer linksassoziativen Morphologie-Grammatik allgemein) sind einerseits das Lexikon kategorisierter Grundformen zusammen mit den Allomorphregeln zur Erzeugung des Allomorphlexikons und andererseits die morphologischen Kombinationsregeln zur Konkatenation der Allomorphe.

Ein wichtiger Gesichtspunkt bei dem Entwurf der DMM war es, linguistische Information in erster Linie in den Kategorien der Lemmata oder Allomorphe beziehungsweise während der Analyse in der Kategorie des aktuellen analysierten Satzanfangs abzulegen. Daneben ist linguistische Information in tabellarischer Form angelegt; beispielsweise in der Tabelle der Ablautreihen deutscher Verben. Die fünf Kombinationsregeln der DMM arbeiten demzufolge auf einem hohen Abstraktionsniveau.

Neben der DMM wurden mit dem System MALAGA Morphologien für weitere Sprachen wie das Koreanische (siehe LEE 1995) oder das Italienische (siehe WETZEL 1996) implementiert.

11.1 Das Lexikon

Das Lexikon der DMM besteht mittlerweile aus 49000 Grundformeinträgen in folgender Zusammensetzung:

Zahl der Einträge		***Quelle/Referenz***
20300	Substantive	Entspricht den Substantiven aus WAHRIG 1986
11100	Adjektive	Entspricht den Adjektiven aus WAHRIG 1986
10600	Namen und Akronyme	Halbautomatische Extraktion aus Korpora
6200	Verben	Entspricht den Verben aus MATER 1966-72
820	Funktionswörter, Partikeln, Suffixe, Präfixe und Präfixoide	Computerlinguistik Erlangen

Tabelle 1: Zusammensetzung des DMM-Lexikons

11.2 Die Allomorphregeln

Die Allomorphregeln der DMM sind nach den Wortarten gegliedert. Die verschiedenen Regeltypen, die im folgenden beschrieben werden, sind für diese Bereiche jeweils separat realisiert.

Bei der Ableitung der Allomorphe werden in der DMM die vier Regularitätsgrade regulär (z. B. reden), semiregulär (z. B. lächeln, heikel), semiirregulär (z. B. geben, Haus) und irregulär (z. B. haben, sein) unterschieden.

Die Allomorphe regulärer und semiregulärer Paradigmen werden direkt aus ihren Lemmata abgeleitet, ohne daß die Oberflächen der Grundformen markiert werden müssen, wobei semireguläre Paradigmen über den Vergleich von Oberflächenmustern erkannt werden können.

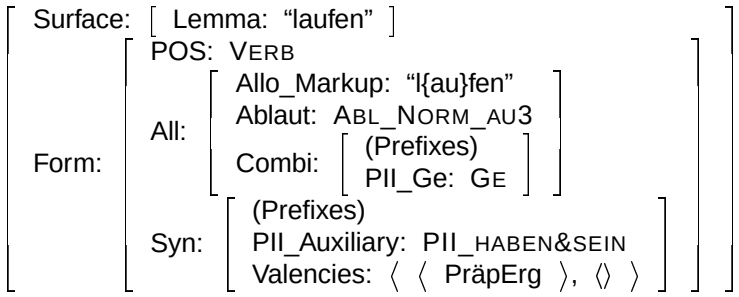


Abb. 7: DMM-Lemma für das Verb laufen

Die Formen semiirregulärer Paradigmen (z. B. laufen) werden über spezielle Regeln generiert, wobei in den jeweiligen Lemmata die Oberflächen der Grundformen Markierungen tragen und die Kategorien gegebenenfalls weitere Informationen – wie beispielsweise die Ablautreihe – enthalten. Irreguläre Paradigmen schließlich werden zusammen mit den Kategorien der morphologisch unveränderlichen Wörter vollständig im Lexikon abgelegt und von einer eigenen Regel ins Allomorphlexikon übertragen.

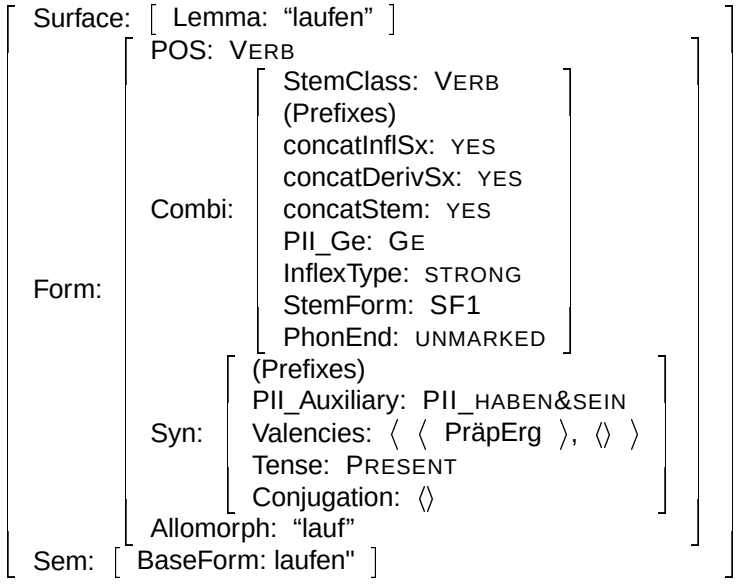


Abb. 8: Eines der drei automatisch abgeleiteten Allomorphe des Lemmas laufen

Ein weiterer Regeltyp erzeugt für Verben und Adjektive anhand morphologischer, phonotaktischer und teilweise lexikalisierter Kriterien Information über Restriktionen bei der Suffigierung. Bei einigen Paradigmen sind Schreibungsänderungen bei der Allomorphbildung (z. B. Faß → Fäss) zu berücksichtigen, die in der DMM durch eigene Regeln behandelt werden. Außerdem erfordern einige Wörter bei der Komposition und Derivation die Erzeugung von Stammverkürzungen (z. B. Erlangen → Erlang/er, Achse → Achs/last) sowie die Zuweisung möglicher Fugenelemente.

11.3 Die Kombinationsregeln

Während der linksassoziativen Analyse einer Wortform wird ihre Oberfläche von links nach rechts in Allomorphe zerlegt, wobei die Kombinationsregeln überprüfen, ob die Konkatenation eines möglichen nächsten Allomorphs an den bestehenden Wortanfang möglich ist. Ist dies der Fall, so werden die Oberflächen von Wortanfang und nächstem

Allomorph zur Oberfläche des neuen Wortanfangs konkateniert. Während das neue Regelpaket die prinzipiellen Möglichkeiten der Fortsetzung nach dem Feuern der zugehörigen Regel angibt, wird die spezielle Information über die Fortsetzungsmöglichkeiten des neuen Wortanfangs in seiner Kategorie abgelegt. Die Kombination der Allomorphe wird in der DMM über vier Kombinationsregeln für die Flexion, die Derivation und die Komposition sowie weitere Regeln für Zahlen und Zahlwörter gesteuert.

Jede Analyse einer Wortform beginnt mit dem Einlesen des ersten Allomorphs durch die Startregel, die gleichzeitig eine Standardkategorie für die weitere Analyse erzeugt. Die nachfolgenden Analyseschritte werden durch vier weitere Kombinationsregeln gesteuert.

Die Suffigierungsregel kombiniert den Wortanfang mit einem Suffix, dabei kann es sich sowohl um ein Flexions- oder Derivationssuffix als auch um ein Fugenelement handeln. Eine Regel zur Stammkonkatenation kombiniert Wortanfänge mit Stämmen, wobei der Wortanfang mit verschiedenen Morphemen enden kann:

Präfix/Präfixoid	(z. B. in vor + betet, Ur + tier)
ge oder zu	(z. B. in auf/ge + laufen)
Stamm	(z. B. in See + stern, Auto + bahn)

Die vierte Kombinationsregel zur Präfixkonkatenierung kombiniert einen Wortanfang mit einem Präfix und analysiert beispielsweise Komposita wie Not/aus/gang. Darüber hinaus existieren weitere Regeln für die Analyse von Zahlen oder Zahlwörtern.

11.4 Leistungsmerkmale

Momentan ist die DMM in der Lage, 96% der laufenden Wortformen des LIMAS-Korpus und 83% seiner uniquen Wortformen zu analysieren. Bei den nichterkannten Wortformen handelt es sich überwiegend um *hapax legomena*, wobei Eigennamen einen wesentlichen Anteil ausmachen. Die Analysegeschwindigkeit beträgt ungefähr 350 Wortformen pro Sekunde (auf einer HP9000/735 112MB RAM oder HP Visualize B132L, 64MB RAM).[2]

12 RNPP – Robuster Nominalphrasen-Parser für das Deutsche

Der hier vorgestellte robuste Nominalphrasen-Parser für das Deutsche (RNPP), der im Rahmen einer Magisterarbeit (SCHNEIDER 1997) entwickelt wurde, analysiert atomare (z. B. Pronomen) und komplexe Nominalphrasen in beliebigen Eingabesätzen. Die zugrundeliegende grammatische Beschreibung lehnt sich an den in WEINRICH 1993 beschriebenen Begriff der Nominalklammer an, wobei das Konzept des von WEINRICH postulierten Nullartikels als Phrasenöffner bei artikellosen Phrasen jedoch gemäß dem Prinzip der Oberflächenkompositionalität verworfen wurde. Stattdessen übernimmt hier das jeweils erste Element einer Nominalphrase – das können neben dem Artikel prinzipiell alle bei WEINRICH als prädeterminierend bezeichneten Elemente sein – die Funktion des Phrasenöffners.

---

[2] Für die Leistungsmerkmale anderer Systeme siehe z. B. HAUSSER 1996.

RNPP erlaubt die Analyse komplexer Nominalphrasen mit eingebetteten Partizipialphrasen in freien Texten. Das Ergebnis der Analyse ist wahlweise eine komplette syntaktisch-semantische Analyse der gefundenen Nominalphrasen oder eine Analyse ihrer Klammerstruktur, was insbesondere für das Tagging freier Texte interessant ist.

12.1 Kombinationsregeln

Diesem Ansatz gemäß enthält die Grammatik drei Regeln zur Analyse komplexer Nominalphrasen: Zum Einlesen des phrasenöffnenden Elements (Regel Open), zum Einlesen phrasenfortsetzender Elemente (Regel Continue) und zum Einlesen des phrasenschließenden Elements (Regel Close).

Darüber hinaus gibt es Regeln zum Einlesen atomarer Phrasen (Regel Atomic) und NP-fremder Elemente, also beispielsweise Verben oder Teile eines Verbalkomplexes (Regel NonNP), und eine Regel zum Einlesen von unbekannten Wortformen bei gleichzeitiger Hypothesenbildung (Regel FallBack).

```
"Der vom Plenum vorgeschlagene Kandidat für die Ernennung zum Vorstand"
[ Segments:   "{Der [[vom Plenum] vorgeschlagene] Kandidat} {für die Ernennung} {zum Vorstand} "
  SynWeight:  0.125 ]
accept
```

Abb. 9: Klammerstruktur-Analyse der Phrase Der vom Plenum vorgeschlagene Kandidat für die Ernennung zum Vorstand.

Durch unterschiedliche Ausgabefilter besteht die Möglichkeit, RNPP entweder eine syntaktische Analyse erzeugen zu lassen, oder im Tagger-Modus nur die Klammerstruktur der analysierten Phrasen zu markieren. Abbildung 9 zeigt die Klammerstrukturanalyse der Phrase Der vom Plenum vorgeschlagene Kandidat für die Ernennung zum Vorstand, während Abbildung 10 die syntaktische-semantischen Analyse der Phrase Der vom Plenum vorgeschlagene Kandidat zeigt.

```
"Der vom Plenum vorgeschlagene Kandidat"
[ Segments:  ⟨ [ Surface:      "Der [[vom Plenum] vorgeschlagene] Kandidat"
                 Structure:    "(Det+PartG+Subst)"
                 Type:         PartP
                 Syn:          [ Yield:   Sg3
                                 Gender: ⟨Masculine⟩ ]
                 LowerFunctors: ⟨ [ Surface:      "vom Plenum"
                                    Structure:    "(Prep+Subst)"
                                    Type:         PP
                                    HigherFunctor: "vorgeschlagene"
                                    Syn:          ⟨ [ Yield:   AdvL
                                                      Gender: ⟨Neuter⟩ ] ⟩
                                    LowerFunctors: ⟨⟩
                                    Condition:    complete ] ⟩
                 Condition:    complete ] ⟩
  SynWeight: 0.15 ]
accept
```

Abb. 10: Syntaktisch-semantische Analyse der Phrase Der vom Plenum vorgeschlagene Kandidat.

12.2 Robustheit und Desambiguierung

Um Nominalphrasen aus beliebigen Sätzen extrahieren zu können, ist es notwendig, auch die Teile einlesen zu können, die nicht zu den von der Grammatik beschriebenen Elementen gehören. In der Abbildung 11 werden beispielsweise das Verb setzte und das zugehörige abgetrennte Präfix ein mit Hilfe der Regel NonNP eingelesen.

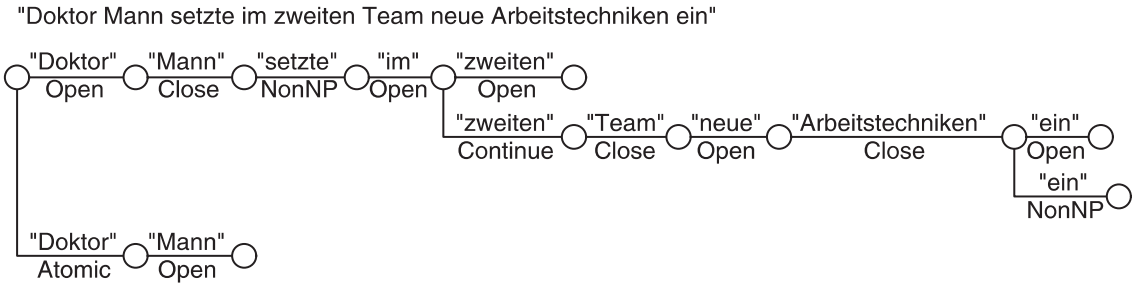


Abb. 11: Analysebaum des Satzes Doktor Mann setzte im zweiten Team neue Arbeitstechniken ein. *Die nicht zu NPs gehörenden Satzteile werden durch die Regel NonNP eingelesen.*

Weiterhin ist ein robustes Verhalten des Parsers dann notwendig, wenn entweder eine ungrammatische NP im Eingabesatz vorliegt oder der Parser Anfang oder Ende einer Nominalphrase falsch erkannt oder kategorisiert hat. Solche Strecken der Eingabe können zwar nicht analysiert werden, das robuste Wiederaufsetzen ermöglicht es jedoch, folgende Nominalphrasen korrekt zu analysieren.

Neben ungrammatischen Eingaben oder Eingaben unbekannter grammatischer Struktur können unbekannte Wortformen die robusten Regelmechanismen aktivieren. Dabei ist es insbesondere für die korpusbasierte Lexikonarbeit interessant, wenn eine unbekannte Wortform nicht einfach überlesen wird, sondern stattdessen anhand des grammatischen Umfelds eine Hypothese über ihre Wortart und ihre grammatische Kategorie generiert wird.

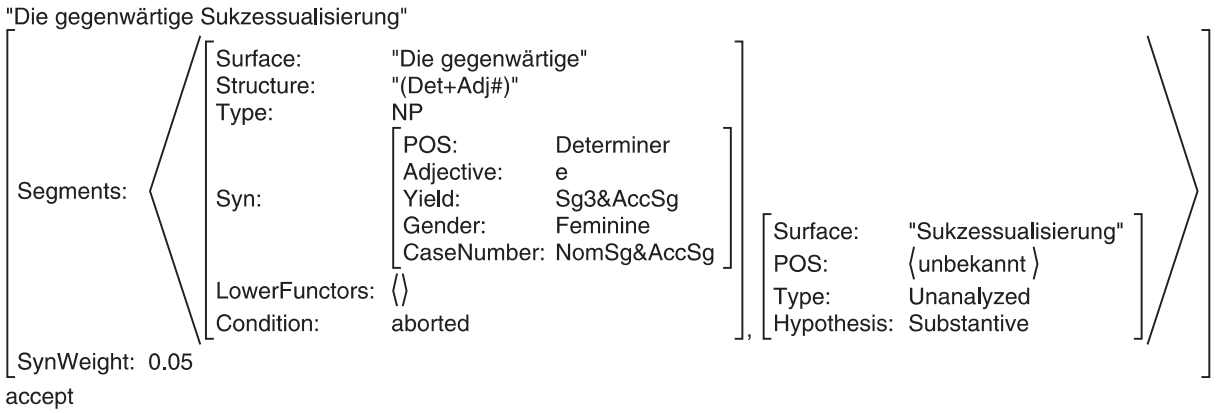


Abb. 12: RNPP-Analyseergebnis zu der Eingabe Die gegenwärtige Sukzessualisierung, *wobei die Oberfläche* Sukzessualisierung *der verwendeten Morphologiegrammatik unbekannt ist.*

Bei der Analyse der NP Die gegenwärtige Sukzessualisierung, die in der Abbildung 12 zu sehen ist, generiert die Regel `FallBack` für die unbekannte Wortform Sukzessualisierung die Hypothese Substantiv und setzt die Analyse des Satzes fort.

Wie in den bisherigen Abbildungen von Analysebäumen (Abbildungen 6 und 11) zu sehen und durch das folgende Beispiel illustriert, entstehen durch das Feuern verschiedener Regeln bei Satzanfängen mit unterschiedlichen Fortsetzungsmöglichkeiten parallele Regelpfade.

So könnte der Phrasenanfang Der zweifelhaften durch das nächste Wort Damen zu einer Nominalphrase im Dativ oder Genitiv abgeschlossen oder beispielsweise zu Der zweifelhaften Damen verfallene Mann fortgesetzt werden. Ohne die Möglichkeit zur robusten Fortsetzung brechen die unzutreffenden Pfade des Analysebaums nach wenigen Wortformen wieder ab. Da die verschiedenen Pfade jedoch vollkommen unabhängig voneinander verfolgt werden, würden bei einer robusten Grammatik zunächst alle Pfade mit robusten Mechanismen fortgesetzt, auch wenn in anderen Pfaden vollständigere Analysen vorliegen. Aus diesem Grunde verwendet der RNPP die MALAGA-Funktion zum Beschneiden des Analysebaums, das sogenannte *pruning*. Nach jedem Analyseschritt – also jeweils nach Einlesen der nächsten Wortform – werden alle aktive Pfade durch eine Pruning-Regel miteinander verglichen. Nach einer frei definierbaren Gewichtungsfunktion werden die einzelnen Pfade gewichtet (beim RNPP beispielsweise nach der Länge der analysierten Phrasen und dem Gewicht der Wortformen), anschließend werden dann Pfade deaktiviert, deren Gewicht unter einem frei wählbaren Schwellenwert (hier beispielsweise das arithmetische Mittel aller Gewichte) liegen. Auf diese Weise ist sichergestellt, daß nur „vielversprechende“ Analysepfade fortgesetzt werden und schlechtere Analysen absterben.

13 DS3 – Ein Semantikparser für das Deutsche

Als weiteres Beispiel wird in diesem Abschnitt eine einfache Grammatik zur Erzeugung einer syn-semantischen Repräsentation der Eingabesätze vorgestellt. Die dieser Repräsentation zugrundeliegende Semantiktheorie wird in HAUSSER 1998 beschrieben und kann im Rahmen dieses Artikels nicht ausführlich erläutert werden.

Das Analyseergebnis der DS3 besteht aus verzeigerten semantischen Token. Ein Token repräsentiert dabei jeweils eine eingelesene Inhalts-Wortform durch ihre Lemmatisierung und ihre Kategorisierung. Die syntaktische Funktor-Argument-Strukturen der Eingabesätze werden über eine bidirektionale Verzeigerung gespeichert, die beispielsweise die Verbindung einer Verbalvalenz zum zugehörigen Valenzfüller repräsentiert. Dabei ist diese Verzeigerung im wesentlichen durch Indizes realisiert, wodurch der hierarchische Baum, der die syntaktische Struktur eines analysierten Satzes repräsentiert, nicht insgesamt sondern in der Form einzelner Token in einer Datenbank abgelegt werden kann.

Die DS3-Grammatik wurde vom Autor im Rahmen eines Proseminars zusammen mit Studenten ohne Vorkenntnisse des MALAGA-Systems oder der Grammatikentwicklung erstellt. Die DS3 besteht insgesamt aus sieben Kombinationsregeln, die einfache deutsche Deklarativsätze analysieren können.

Die semantischen Token werden von der DS3 während der syntaktischen Analyse inkrementell aufgebaut. Bei jedem Einlesen einer Wortform wird entweder ein neues Token angelegt oder ein bereits bestehendes verändert. Bestimmte Attribute innerhalb eines neuangelegten Tokens bleiben dabei zunächst unbesetzt. Die folgende Abbildung 13 zeigt die Token, die beim Einlesen des Satzanfangs Der verliebte generiert werden. Das erste Token wird bereits beim Einlesen des Artikels Der generiert, wobei das Attribut `ADV_Mod` noch fehlt. Da Der hier der Beginn einer NP, aber selber noch kein Inhaltswort ist, ist das Token nur minimal gefüllt: Einzig die Klasse des entstehenden Elements ist bereits eingetragen. Beim Einlesen der nächsten Wortform verliebte wird dann zum einen ein weiteres Token angelegt, zum anderen wird in das erste Token das zusätzliche Attribut `ADV_Mod` eingetragen. Dieses Attribut bildet zusammen mit dem Attribut `NP_Functor` des Tokens für das modifizierende Adjektiv die bidirektionale Verzeigerung zwischen dem Token des noch nicht eingelesenen Kerns der NP und dem attributiv verwendeten Partizip verliebte.

Das zweite Beispiel in Abbildung 14 zeigt anhand der semantischen Token zu dem Satzanfang Der Mann verschenkte zum einen ein vollständiges Token zum Kern einer NP sowie die Verzeigerung eines Verbs mit einem seiner Valenzfüller.

Im Token zu der Oberfläche Der Mann sind – im Gegensatz zum entsprechenden Token in Abbildung 13 das Attribut Key, das die Grundform Mann als Schlüssel enthält, sowie das Verzeigerungsattribute `Verb_Functor` gefüllt. Weiterhin ist im Token der NP mit dem Kern "Mann" angegeben, welche Valenz im Verb durch sie gefüllt wurde. Im Token zur Oberfläche verschenkte findet sich die symmetrische Verzeigerung, wobei der Wert des Verzeigerungsattributs `Arguments` als Liste angelegt ist, um weitere Valenzfüller aufnehmen zu können.

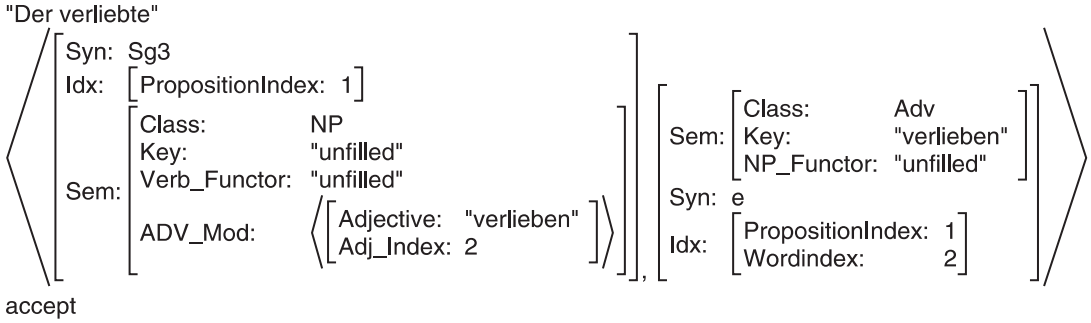


Abbildung 13: Semantische Token des Satzanfanges Der verliebte

Bei jedem Analyseschritt legt die DS3 alle verfügbare syn-semantische Information in Form von neuen Token oder Modifikationen in der Liste der semantischen Token ab. Die eigentliche syntaktische Analyse arbeitet auf einem eigenen Bereich der Kategorie, der einer der Kategorie einer einfachen syntaktischen Analyse entspricht und hier nicht abgebildet ist.

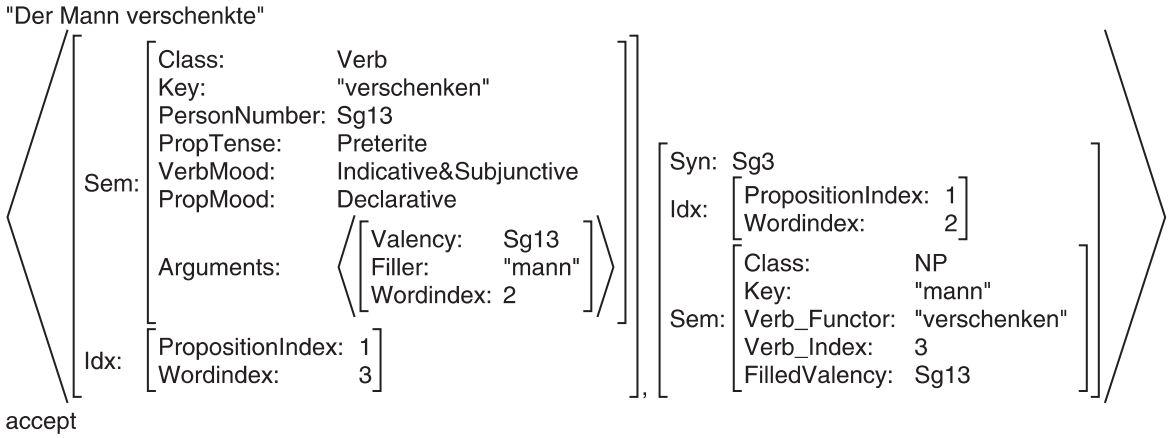


Abb. 14: Semantische Token des Satzanfanges Der Mann verschenkte

Derartige syn-semantische Token können dann in einer Datenbank abgespeichert werden. Eine solche Datenbank kann beispielsweise die Grundlage eines Systems für semantische Inferenzen bilden, wie dies z. B. in HAUSSER 1998 beschrieben wird.

Petra Maier, Universität München/CIS

Defaultzuweisung morpho-syntaktischer Kategorien

1 Einleitung

Neben der Erstellung elektronischer Wörterbücher ist vor allem auch deren Pflege und Erweiterung eine wichtige Aufgabe der Computerlexikographie. Für einen breiten Einsatz elektronischer Wörterbücher ist es notwendig, daß auch ein nicht speziell ausgebildeter Benutzer das Wörterbuch schnell und korrekt beispielsweise an neue Fachwortschätze anpassen kann. Wünschenswert ist natürlich ein System, das es erlaubt, das Wörterbuch auf der Basis eines Fachkorpus vollautomatisch zu erweitern. Eine automatische Kodierung ist aber vor allem im Bereich der semantischen und syntaktischen Merkmale ausgesprochen problematisch.[1] Für den Bereich der morpho-syntaktischen Kodierung ist das Problem insbesondere dann, wenn man als Datenbasis bereits ein nahezu vollständiges elektronisches Wörterbuch des Allgemeinwortschatzes zur Verfügung hat, durchaus lösbar.

Das Ziel dieses Beitrags ist es, ein Verfahren vorzustellen, mit dem eine solche Anpassung im Bereich der morpho-syntaktischen Kategorien automatisch bzw. halbautomatisch erfolgen kann. Eine weitere Anwendung ist die Behandlung unbekannter Wortformen direkt bei der Lemmatisierung.

Das Verfahren basiert auf einer Suffixanalyse kombiniert mit lokalen Grammatiken zur Bestimmung der Wortart. Im Gegensatz zu herkömmlichen Suffixanalysen, wie sie schon sehr früh zur lexikonunabhängigen Lemmatisierung benutzt wurden (siehe etwa WILLÉE 1979 oder SCHOTT 1978), liegt diesem Verfahren sowohl eine systematische Suffixanalyse der lexikalischen Datenbasis, als auch eine Untersuchung über die Vorkommenshäufigkeiten der einzelnen morphologischen Klassen in umfangreichen Korpora zugrunde.

---

[1] Für einen Ansatz zur Kodierung von Subkategorisierungsrahmen für deutsche Verben vergleiche etwa LANGER, MAIER & OESTERLE (1996).

2 Problemstellung

Ausgangspunkt ist das CISLEX-Wörterbuchsystem (GUENTHNER & MAIER 1996). Für die morpho-syntaktische Kodierung werden 11 verschiedene Wortarten unterschieden. Die flektierenden Wortarten sind weiter aufgeteilt in morphologische Subklassen, die zu insgesamt 670 verschiedenen morpho-syntaktischen Kategorien im Lexikon führen.

Bei der automatischen bzw. halbautomatischen Anpassung des CISLEX an neue Fachwortschätze geht es darum, unbekannte Wörter des Ausgangskorpus mit ihrer Lemmainformation (das ist Grundform, Wortart und morphologische Klasse) zu versehen, um dann das gesamte Paradigma dieser Wörter im CISLEX zu erfassen. Wir beschränken uns hier auf reine Wortformen (zur Behandlung von Sonderformen wie Abkürzungen, numerischen Konstruktionen usw. siehe MAIER 1995). Der erste Schritt ist die vollständige Lemmatisierung des Textes auf der Basis des CISLEX.[2]

Die verbleibenden unbekannten Wörter werden mit einer Fehlerliste verglichen, um bereits beobachtete Rechtschreibfehler zu eliminieren. Für die nun noch verbleibenden unbekannten Wörter muß die Wortart bestimmt werden und, im Falle der flektierenden Wortarten, die Grundform sowie die zugehörige morphologische Klasse berechnet werden. Aus dieser Information wird dann das gesamte Paradigma generiert und ins Lexikon integriert.

3 Art der Lexikoneinträge

Beim CISLEX wird eine ähnliche Strategie verfolgt wie bei den DELA-Wörterbüchern des LADL (GROSS 1991). Neben dem CISLEX-EF, das die Kodierung der einfachen Grundformen des Deutschen enthält, wird für Verarbeitungszwecke das entsprechende Vollformenlexikon CISLEX-Flex verwendet. Ein Eintrag im CISLEX-EF hat die Form

[<Grundform>,.<Kategorie>]

und ein Eintrag im CISLEX-Flex hat die Form

[<Vollform>,<Grundform> . <Kategorie>(:<morph. Merkmale>)*]

Die (morpho-syntaktische) Kategorie ist im Falle der nichtflektierenden Wortarten die Wortart, möglicherweise weiter untergliedert nach Distributionseigenschaften. Im Falle der flektierenden Wortarten besteht die morpho-syntaktische Kategorie aus der Wortart und einer Nummer für das Paradigma. Bei Nomen ist die morpho-syntaktische Kategorie ein Tripel aus Genus, Flexionstyp im Singular und Flexionstyp im Plural.

Beispiele für Grundformeinträge:

[Abend,.mask(NS2,NP2)]
[abrupt,.ADJ11]
[adeln,.VSW6]

---

[2] Insbesondere erlaubt das umfangreiche Eigennamenlexikon des CISLEX mit ca. 1 Mio. Einträgen, die Anzahl der unbekannten Formen relativ gering zu halten. Bei Zeitungstexten bleiben im Durchschnitt weniger als 2 % der Tokens unbekannt.

Beispiele für Vollformeinträge:
```
[Abende,Abend.mask(NS2,NP2):nmM:gmM:amM]
[abruptes,abrupt.ADJ11:neNzp:neNxp:aeNzp:aeNxp]
[adelte,adeln.VSW6:1eVi:3eVi:1eVc:3eVc]
```

Bei den Nomen werden 320 verschiedene Kategorien unterschieden, bei den Verben 265 und bei Adjektiven 30. Rechnet man die morphologischen Merkmale noch hinzu, so ergeben sich mehr als 2500 mögliche verschiedene Kategorien für die flektierenden Wortarten.

4 Lexikonerweiterung

Bei der Lexikonerweiterung muß einer unbekannten Wortform eine lexikalische Kategorie zugeordnet werden. Es genügt dabei nicht, sich nur auf die Wortart und morphologische Subklasse zu beschränken, da für einen neuen Eintrag auch die Grundform benötigt wird, die sich nur aufgrund der morphologischen Merkmale der Wortform berechnen läßt. Das heißt, einer unbekannten Wortform muß eine der mehr als 2500 vollständigen Kategorien zugeordnet werden. Das Verfahren hat zwei Arbeitsmodi:

- *vollautomatisch*: In diesem Fall wird der neue Lexikoneintrag vollautomatisch erstellt. Hier läßt sich eine gewisse Fehlerrate nicht vermeiden, allerdings entfällt die relativ zeitaufwendige manuelle Kontrolle.
- *interaktiv:* In diesem Fall muß der Bearbeiter die vorgeschlagene Kategorie bestätigen (im Falle einer eindeutigen Lösung) oder aus der nach Wahrscheinlichkeit geordneten Vorschlagsliste eine Kategorie auswählen. Für den Fall, daß keiner der Vorschläge korrekt war, steht dem Benutzer ein Modul zur manuellen Eingabe des Paradigmas zur Verfügung. Aufgrund weniger Formen wird hier die entsprechende Kategorie bestimmt.

Für den interaktiven Modus muß also neben dem Defaulter auch noch ein Eingabetool für morpho-syntaktische Kategorien zur Verfügung stehen. Abbildung 1 liefert einen Überblick über die verschiedenen Schritte: Die unbekannte Wortform wird einer Suffixanalyse unterzogen. Im Falle einer automatischen Lexikonerweiterung wird aufgrund der Suffixanalyse die wahrscheinlichste Kategorie bestimmt. Im Falle der interaktiven Bearbeitung werden die *n* wahrscheinlichsten Kategorien bestimmt. Als Maximalzahl für die zur Auswahl gestellten Kategorien wurde hier fünf gewählt. Damit ist die Chance, daß sich die korrekte Kategorie in der Auswahl befindet sehr hoch, andererseits ist es auch für den Benutzer eine zumutbare Menge.[3] Ist die korrekte Kategorie unter den Vorschlägen, so wird sie ausgewählt, andernfalls wird die Form ans Eingabetool zur manuellen Bestimmung des Paradigmas übergeben. In allen Fällen besteht der letzte Bearbeitungsschritt darin, aufgrund der morphologischen Merkmale

[3] Wenn dieser Maximalwert zu groß gewählt wird, ist es in vielen Fällen einfacher das Paradigma über das Eingabetool manuell zu bestimmen.

und der morphosyntaktischen Klasse die Grundform zu bestimmen. Die Grundform wird mit der morphosyntaktischen Kategorie ins Grundformenlexikon aufgenommen. Außerdem werden aufgrund des Paradigmas alle zugehörigen Flexionsformen generiert und ins Vollformenlexikon aufgenommen.

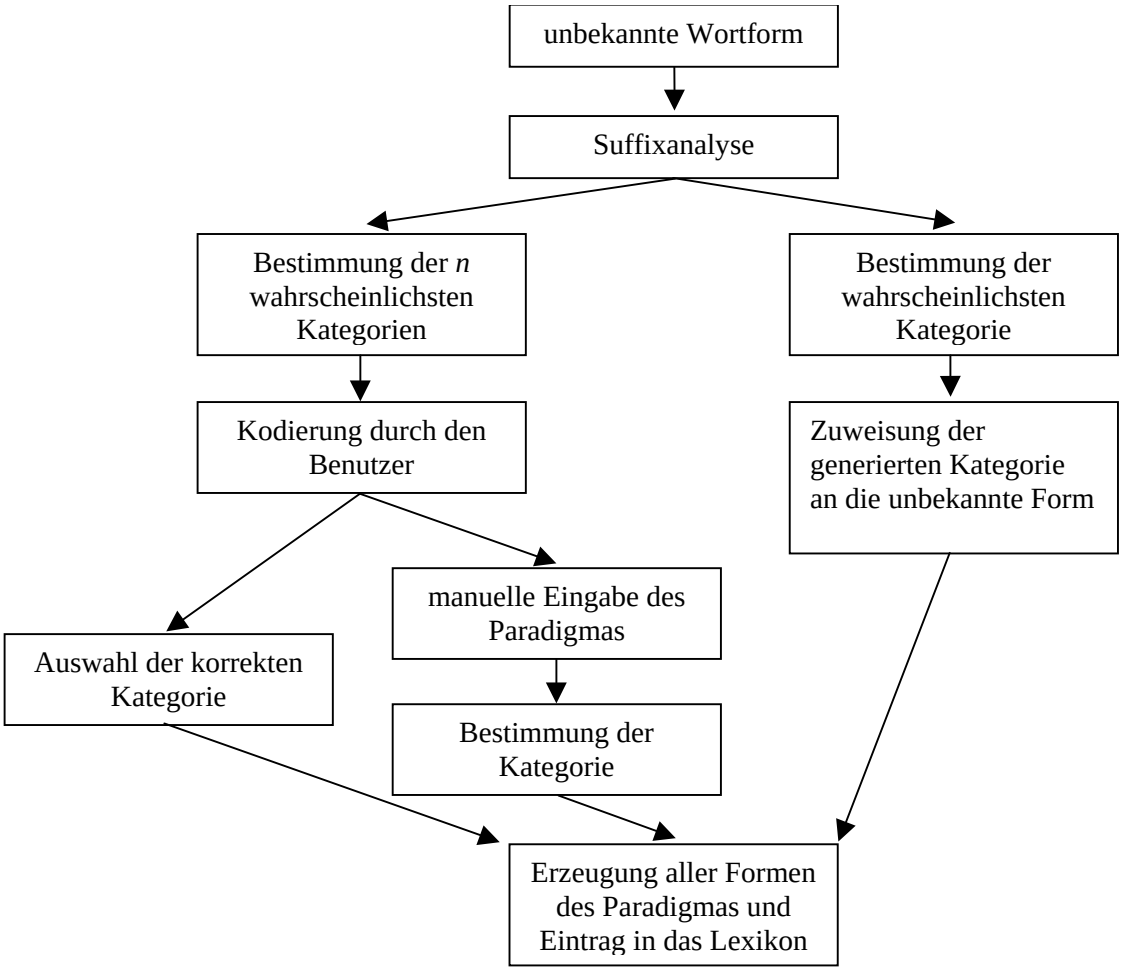


Abb. 1: Überblick zum Verarbeitungsablauf

5 Grundannahmen

Der Suffixanalyse des CISLEX-Flex zum Zwecke der automatischen Klassifikation liegen folgende Annahmen zugrunde:

(1) Es kann davon ausgegangen werden, daß im Bereich der Funktionswörter kaum unbekannte Wörter zu erwarten sind. Lediglich bei den Adverbien sind Neuzugänge zu erwarten, diese werden sich allerdings auf Derivationen mit den Suffixen *-dings, -lings, -wärts, -halber/n, -lei, -maßen, -weg, -weise* beschränken. Bei einer separaten Behandlung dieser Fälle, kann also der gesamte Funktionswortbereich für die Suffixanalyse ausgeklammert werden.

(2) Es wird davon ausgegangen, daß die Groß-/Kleinschreibung weitgehend die Zuordnung zu den beiden Gruppen Nomen oder Eigennamen und Adjektive, Verben usw. liefert. Damit läßt sich die Suffixanalyse im Lexikon aufteilen in a) den nominalen Bereich (N-Suffixe) und b) Adjektive und Verben (AV-Suffixe).

(3) Im AV-Bereich dient die Suffixanalyse in erster Linie der Wortartbestimmung, da die morphologische Klasse dann in der Regel (bis auf Umlautphänomene bei der Adjektivkomparation) aufgrund der orthographischen Eigenschaften der Grundform bereits festgelegt ist.
(4) Im N-Bereich muß die Suffixanalyse sowohl die Zuordnung zu einer morphologischen Klasse als auch die Genusbestimmung ermöglichen.
(5) Es wird davon ausgegangen, daß im Adjektiv-Verb-Bereich sämtliche unregelmäßigen oder starken Formen ebenso wie Suppletivformen bereits erfaßt sind. Diese Klassen sind von der Analyse ausgenommen, da sie die Ergebnisse nur verfälschen würden.

Im Folgenden wird das Verfahren nur am Beispiel der Klassifikation von Nomen vorgestellt. Das entsprechende Verfahren und die entsprechenden Tests ergaben im Bereich der Adjektive und Verben annähernd dieselben Werte, so daß die Ergebnisse übertragen werden können.

6 Lokale Grammatiken zur Wortartbestimmung

Eine grobe Wortartbestimmung ist Grundlage der suffixbasierten Feinklassifikation. Dieses Problem kann aufgrund der oben gemachten Annahmen weitgehend reduziert werden auf die Unterscheidung von Eigennamen und Nomen. Doch auch im Adjektiv-Verb-Bereich können die Ergebnisse durch eine separate Wortartbestimmung wesentlich verbessert werden. Die Wortartbestimmung wird durch lokale Grammatiken durchgeführt, die durch Transduktoren formalisiert werden.

Abbildung 2 zeigt einen kleinen Automaten, der typische Kontexte für Personennamen darstellt und es damit gestattet, unbekannte Wörter, die in dieses Raster passen als Vor- bzw. Nachname zu klassifizieren. Jeder Pfad durch den Automaten stellt einen Kontext dar, in dem ein unbekanntes Wort (bezeichnet durch die Kategorie *UK*) als Eigenname zu klassifizieren ist. Die Knoten eines solchen Automaten können Wörter, Kategorien (gekennzeichnet durch spitze Klammern) oder auch selbst Automaten (grau unterlegt) enthalten. Unter einem Knoten wird die gewünschte Ausgabe angefügt. Die Kategorien in dem Beispielautomaten sind *EN;vor* für Vornamen, *EN;nach* für Nachnamen, *N;hum* für Nomen mit dem Merkmal *menschlich* und *UK* für unbekannt. Außerdem werden die Subautomaten *TITEL* und *ANREDEN* verwendet, die Muster für alle Formen von Anreden und Titel enthalten.

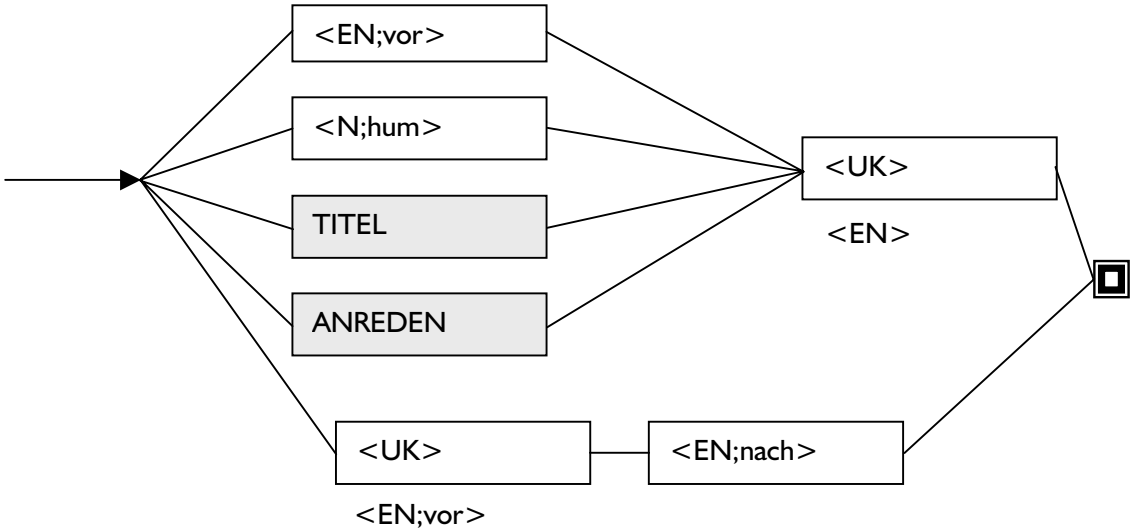


Abb. 2: Automat zur Verarbeitung von Personennamn

Bei diesem Automaten wird vorausgesetzt, daß er nur auf großgeschriebene unbekannte Wörter angewandt wird. Ein großgeschriebenes unbekanntes Wort wird dann als Eigenname erkannt, wenn es

- nach einem Vornamen auftritt (*<EN;vor> <UK>*)
- nach einem Personenbezeichner auftritt (*<N;hum> <UK>*)
- nach einem Titel, der selbst wiederum auch komplex sein kann (*TITEL <UK>*)
- nach einer möglicherweise komplexen Anrede (*ANREDEN <UK>*)
- vor einem Nachnamen (*<UK> <EN;nach>*)

Damit sind Fälle abgedeckt wie Karl-Friedrich XYZ, der Patient XYZ, Herr Dr. rer. nat. XYZ oder XYZ Müller. Weitere Beispiele für lokale Grammatiken zur Wortartdesambiguierung finden sich in MAIER 1995 fürs Deutsche und in SILBERZTEIN 1989 fürs Französische.

7 Klassifizierung der Nomen

Die Ausgangsbasis für die Suffixanalyse sind 100.000 flektierte Nomen des CISLEX. Diese 100.000 Wortformen entsprechen ca. 40.000 Grundformen. Insgesamt gibt es 213 verschiedene morpho-syntaktische Klassen (das sind Kombinationen aus den 13 Singularklassen und 113 Pluralklassen). Kombiniert mit der Genusinformation ergibt das 320 verschiedene morpho-syntaktische Kategorien, die zusammen mit den morphologischen Merkmalen 900 volle Kategorien ergeben. Für die Zwecke der Analyse wurden die morphologischen Merkmale ersetzt durch einen Grundformkode, der die Information enthält, welche Operationen ausgeführt werden müssen, um die Grundform zu erzeugen.

Das Ziel der Untersuchung war eine möglichst gute Zuordnung von Suffix[4] und morphologischer Kategorie. Dazu wird aufgrund der im Lexikon enthaltenen Information ein Suffixbaum konstruiert.

Der Kategorienvorschlag ergibt sich dann indem das maximale Suffix S_{max} des unbekannten Wortes ermittelt wird, das im Suffixbaum enthalten ist. Aus den möglichen Kategorien, die S_{max} haben kann, müssen dann die besten ausgewählt werden.[5]

7.1 Konstruktion eines Suffixbaums

Der erste Schritt bei der Konstruktion des Suffixbaums ist das Einlesen der invertierten Suffixe des CISLEX in einen Trie, dessen externe Knoten die Liste der morphologischen Subklassen mit den Grundformkodes enthalten. Ein Trie ist ein Baum, für dessen Knoten und Kanten folgendes gilt:

- Es gibt genau einen Wurzelknoten.
- Jeder interne Knoten hat mindestens ein Kind und (außer der Wurzel) genau einen Vorgänger.
- Jeder Knoten entspricht einem Buchstaben.
- Jeder Pfad von der Wurzel bis zu einem Blatt entspricht einem Eintrag (hier einem Suffix).

Aus Praktikabilitätsgründen werden nicht alle Suffixe sondern nur Suffixe bis zur Länge acht benutzt. Damit ist gewährleistet, daß auch im Falle langer Flexive (wie beispielsweise bei den flektierten Komparationsformen der Adjektive) noch genügend Zeichen der Grundform verarbeitet werden. Andererseits besteht nicht die Gefahr, daß das Verfahren explodiert.

Aus diesem Trie werden dann alle redundanten Knoten gelöscht. Redundante Knoten entstehen dann, wenn es Suffixe s_1 und s_2 gibt, die dieselben morphologischen Subklassen und Grundformkodes haben und s_1 wiederum ein Suffix von s_2 ist. Mit anderen Worten, wenn die Information eines Suffix durch Hinzunahme weiterer Zeichen am Anfang seine Information nicht mehr verändert, so möchte man natürlich nur die kürzeste Form im Trie haben. Ein Knoten k kann genau dann aus dem Trie gelöscht werden, wenn er als einziges Kind einen externen Knoten k_0 hat und der unmittelbare Vorgängerknoten ein Kind hat, das identisch mit k_0 ist.
Eine Untersuchung der so eingetragenen Suffixe ergab, daß es einerseits sehr spezifische Suffixe gibt, die nur eine sehr geringe Anzahl von Kategorien zulassen, und ande-

---

[4] Mit dem Begriff *Suffix* beziehe ich mich in dieser Untersuchung nicht auf morphologische Einheiten. Der Begriff ist in rein formalem Sinn als echter *End-String* zu verstehen.

[5] Ein ähnlicher Ansatz wird auch von SCHMID (1994a/b) verfolgt. In seinem Verfahren wird ein Suffixentscheidungsbaum konstruiert, mit dessen Hilfe eine suffixbasierte Wortartdesambiguierung durchgeführt wird. Die Menge der möglichen Kategorien ist in diesem Fall natürlich wesentlich geringer. Der Suffixbaum liefert Gewichtungen für bestimmte Analysen, so daß mit statistischen Methoden dann die Gesamtfolge desambiguiert werden kann. In unserem Fall ist die Wortart bereits festgelegt, es muß hier die morphologische Feinklassifikation vorgenommen werden.

rerseits sehr unspezifische Suffixe, die eine Vielzahl von Kategorien zulassen und die meist auch relativ kurz sind. Um eine Trennung zwischen spezifischen und unspezifischen Suffixen zu erhalten, werden nur solche Suffixe im Trie behalten, die maximal fünf Kategorien zulassen. Für Wörter, die mit diesem Suffixbaum nicht klassifiziert werden können (d. h., daß kein Suffix des Wortes im Suffixbaum enthalten ist), wird ein zweiter Trie aufgebaut, der nur Wortendbigramme enthält. Diesen Bigrammen sind dann nur die jeweils fünf häufigsten Kategorien zugeordnet.

Diese Aufspaltung hat den Vorteil, daß man beispielsweise die Möglichkeit hat, für die automatische Klassifikation nur die spezifischen Suffixe zu verwenden, um so eine sehr hohe Trefferrate zu garantieren. Dabei bleiben natürlich nicht klassifizierbare Wörter übrig, die dann interaktiv kodiert werden müssen.

Für diejenigen spezifischen Suffixe, die keine eindeutige Zuordnung zuließen, ebenso wie für die Endbigramme wurde für die jeweils möglichen Kategorien eine lexikonspezifische Gewichtung vorgenommen, nach der die maximal fünf Kategorien in der Liste angeordnet werden. Das heißt, die einzelnen Kategorien wurden danach bewertet, welcher Anteil der Lexikoneinträge mit diesem Suffix jeweils dieser Kategorie angehören. Das entspricht der Berechnung der bedingten Wahrscheinlichkeit *P(k|s)*, der Wahrscheinlichkeit, daß die Kategorie *k* auftritt unter der Bedingung, daß das Suffix *s* auftritt.

7.2 Ergebnisse

Das oben beschriebene Verfahren wurde auf verschiedene Arten getestet. Als Testsets dienten die Vollformen des CISLEX (T1), eine Liste der 60.000 hochfrequentesten Nomen extrahiert aus vier Jahrgängen Süddeutscher Zeitung (T2) und eine Liste mit 12.625 Nomenvollformen aus medizinischen Fachtexten.[6] Für die einzelnen Testsets ergaben sich folgende Werte:

	T1	***T2***	***T3***
1. Vorschlag korrekt	91,19%	71,81%	77,98%
2. Vorschlag korrekt	5,02%	8,48%	7,32%
3. Vorschlag korrekt	1,59%	2,27%	1,31%
4. Vorschlag korrekt	0,68%	1,25%	0,47%
5. Vorschlag korrekt	0,33%	0,56%	0,31%
kein Vorschlag zutreffend	1,18%	15,27%	12,16%

Erwartungsgemäß fallen die Ergebnisse für die Spalte T1, also für die Nomen des CISLEX, die ja auch gleichzeitig die Trainingsdaten waren, sehr gut aus: für über 91% ist der erste Vorschlag korrekt, und nur für knapp über 1% traf keiner der Vorschläge

[6] Ein ähnlicher Ansatz wird auch von SCHMID (1994a/b) verfolgt. In seinem Verfahren wird ein Suffixentscheidungsbaum konstruiert, mit dessen Hilfe eine suffixbasierte Wortartdesambiguierung durchgeführt wird. Die Menge der möglichen Kategorien ist in diesem Fall natürlich wesentlich geringer. Der Suffixbaum liefert Gewichtungen für bestimmte Analysen, so daß mit statistischen Methoden dann die Gesamtfolge desambiguiert werden kann. In unserem Fall ist die Wortart bereits festgelegt, es muß hier die morphologische Feinklassifikation vorgenommen werden.

zu. Die schlechtesten Ergebnisse lieferten die Nomen aus dem Zeitungskorpus (T2). Bei 15,27% traf keiner der Vorschläge zu und lediglich 71,81% konnten mit dem ersten Vorschlag klassifiziert werden. Für das Fachvokabular (T3), bei dem es kaum Überschneidungen mit den Trainingsdaten gab, sind die Werte deutlich besser als bei den SZ-Nomen. Fast 78% stimmten mit dem ersten Vorschlag überein und nur 12,16% hatten keinen zutreffenden Vorschlag. Wenn das Verfahren eine korrekte Kategorie liefert, dann ist mit sehr hoher Wahrscheinlichkeit bereits der erste Vorschlag korrekt (93% für T1, 84% für T2 und 89% für T3).

Das Problem bei der Gewichtung der Klassen nach Frequenz im Lexikon ist, daß das Vorkommen der Kategorien im Lexikon nicht dem Vorkommen in Texten entspricht, und dadurch die Ergebnisse verfälscht werden können.

8 Analyse von Frequenzlisten

Um die Probleme der lexikonbasierten Gewichtung der Kategorien bei mehrdeutigen Suffixen zu umgehen, wurde ein weiterer Test durchgeführt, bei dem die Auswahl und Gewichtung der einzelnen Kategorien und Suffixe mittels einer Frequenzliste ermittelt wurde. Die Frequenzliste umfaßt die 60.000 häufigsten Nomenformen bezogen auf vier Jahrgänge Süddeutsche Zeitung. Anhand dieser Frequenzliste wurde eine Frequenzliste der Suffix-Kategorien-Paare erstellt. Die verschiedenen Kategorien zu einem Suffix wurden gemäß ihrer Rangordnung in der Frequenzliste bewertet.

Neben der Frequenzliste für Suffix-Kategorien-Paare bezogen aufs CISLEX wurde auch die Frequenzliste für Suffix-Kategorien-Paare aus dem SZ-Korpus erstellt. Ein Vergleich dieser Frequenzlisten zeigt, daß sich die Verteilung von morphologischen Subklassen mit den entsprechenden Suffixen in Gebrauchstexten deutlich von der im Lexikon unterscheidet. Da an dieser Stelle nicht auf interessante Details aus diesem Bereich eingegangen werden kann, sollen hier nur einige wichtige Auffälligkeiten erwähnt werden.

- Die Anzahl der Suffix-Kategorien-Paare und die Streuung ist für die Lexikon-Frequenzliste wesentlich größer.
- Während die Auftretenshäufigkeiten für die Lexikondaten – abgesehen von den acht frequentesten Paaren – relativ gleichmäßig abnehmen, ist bei den SZ-Daten nach einem relativ ähnlich verteilten Anfangssegment ein rapider Abfall der Auftretenshäufigkeiten zu beobachten.
- Die Suffixe der hochfrequenten Suffix-Kategorien-Paare sind für das SZ-Korpus im Schnitt wesentlich kürzer als die der Lexikondaten.
- Es gibt bei den häufigsten Paaren nahezu keine Überschneidungen zwischen beiden Listen.

Diese Beobachtungen verdeutlicht auch die folgende Tabelle, die die unterschiedliche Verteilung der zehn häufigsten Suffix-Kategorien-Paare bezogen auf das SZ-Korpus und das CISLEX zeigt.

SZ-Korpus		***CISLEX***	
Häufigkeit	*Suffix:Kategorie*	*Häufigkeit*	*Suffix:Kategorie*
1.03%	*rung,*.fem(NS0,NP3)	2.36%	*rinnen,*.fem(NS0,NP5)
0.75%	*erung,*.fem(NS0,NP3)	2.05%	*keiten,*.fem(NS0,NP3)
0.70%	*tion,*.fem(NS0,NP3)	1.16%	*gkeit,*.fem(NS0,NP3)
0.65%	*tung,*.fem(NS0,NP3)	1.05%	*ierungen,*.fem(NS0,NP3)
0.59%	*tag,*.mask(NS1,NP2)	1.05%	*ierung,*.fem(NS0,NP3)
0.55%	*schaft,*.fem(NS0,NP3)	0.84%	*heiten,*.fem(NS0,NP3)
0.55%	*haft,*.fem(NS0,NP3)	0.58%	*tinnen,*.fem(NS0,NP5)
0.55%	*chaft,*.fem(NS0,NP3)	0.49%	*tungen,*.fem(NS0,NP3)
0.51%	*lung,*.fem(NS0,NP3)	0.49%	*ereien,*.fem(NS0,NP3)
0.49%	*zent,*.neut(NS1,NP2)	0.45%	*hkeit,*.fem(NS0,NP3)

Auffällig ist, daß in der CISLEX-Spalte unter den ersten zehn Paaren zweimal Suffixe auf *innen*, also Pluralformen der *in*-Ableitung von maskulinen Personen- oder Tierbezeichnungen auftreten. Dieses Suffix kommt im SZ-Korpus unter den ersten zehn Paaren überhaupt nicht vor. Das liegt natürlich daran, daß das CISLEX hinsichtlich produktiver Derivationen systematisch vervollständigt wurde, während diese femininen Formen in Gebrauchstexten bekanntermaßen relativ selten auftreten. In beiden Fällen sind Suffixe auf *ung* relativ stark vertreten. Suffixe auf *haft/schaft* dagegen, die bei den SZ-Daten gleich mehrfach unter den ersten zehn vertreten sind, kommen bei den Lexikondaten unter den ersten zehn überhaupt nicht vor.

Das im vorigen Kapitel beschriebene Verfahren wurde nun mit den neuen Gewichtungen der Kategorien nochmals auf die drei Testsets T1 (Vollformen des CISLEX), T2 (Vollformen der SZ) und T3 (medizinisches Fachvokabular) angewendet. Das Ergebnis dieses Testlaufs zeigt die folgende Tabelle:

	T1	***T2***	***T3***
1. Vorschlag korrekt	87,36%	68,77%	76,20%
2. Vorschlag korrekt	4,53%	7,50%	6,83%
3. Vorschlag korrekt	1,23%	1,56%	1,00%
4. Vorschlag korrekt	0,39%	0,70%	0,35%
5. Vorschlag korrekt	0,12%	0,22%	0,08%
kein Vorschlag zutreffend	6,38%	21,24%	15,55%

Die Ergebnisse gegenüber einer lexikonbasierten Gewichtung fallen wider Erwarten schlechter aus. Selbst für T2, das ja mit den Trainingsdaten weitgehend übereinstimmt,[7] haben sich die Ergebnisse deutlich verschlechtert, sowohl was die Werte für den ersten Vorschlag anbetrifft, als auch die aufgrund der Vorschlagsliste überhaupt nicht zu klassifizierenden Wörter. Vergleichsweise am wenigsten beeinflußt durch das leicht geänderte Auswahlkriterium sind die Werte für T3, das Fachvokabular.

[7] Allerdings wurde bei der Erstellung der Suffix-Kategorien-Paare die tatsächliche Frequenz im Korpus und nicht die Frequenz bezogen auf die Wortliste T2 verwendet.

9 Bewertung

Insgesamt hat sich gezeigt, daß in normalen Texten (hier Zeitungstexte) die Varianz der morphologischen Kategorien an sich, ebenso wie die Varianz der Suffix-Kategorien-Paare wesentlich geringer ist als im Lexikon. Dieses Ergebnis überrascht nicht. Auf den ersten Blick erstaunlicher ist jedoch die Tatsache, daß die unterschiedliche Bewertung der verschiedenen Kategorisierungsmöglichkeiten trotz erheblicher Unterschiede keine großen Unterschiede im Ergebnis nach sich zieht. Dazu muß man allerdings beachten, daß etwa im Falle von T1 bereits 74 % der Formen eine eindeutige Kategorie zugeordnet bekommen (dieser Wert ist natürlich für T2 und T3 geringer). Da T3 für den intendierten Einsatz die typische Situation darstellt, sollte das Verfahren gewählt werden, das für T3 die besten Daten und die größere Bandbreite an Suffixen liefert, also das lexikonbasierte Verfahren.

Das relativ schlechte Testergebnis bei Zeitungstexten zeigt einmal mehr, daß eine rein suffixbasierte Lemmatisierung, selbst wenn sie auf einer sorgfältigen Suffixanalyse eines fast vollständigen elektronischen Wörterbuchs basiert, im Hinblick auf eine morpho-syntaktische Feinklassifikation keine zufriedenstellenden Ergebnisse liefert. Für die gegebene Problemstellung, die automatische bzw. halbautomatische Klassifikation unbekannter Wörter, sind die Ergebnisse von T2 jedoch irrelevant.
Eine genauere Betrachtung der falsch klassifizierten Formen zeigte, daß es sich dabei hauptsächlich um einmorphige Wörter bzw. um Komposita mit einmorphigen Köpfen handelte, bei denen man davon ausgehen kann, daß sie vollständig im Lexikon erfaßt sind. Was noch durch die Beobachtung unterstützt wird, daß die Kategorienzuordnung bei diesen Wörtern nicht über die eigentliche Suffixliste geschah, sondern durch die Heuristik auf dem Endbigramm. Die Erfahrung zeigte jedoch, daß fürs CISLEX unbekannte Wörter im Normalfall Derivationen (insbesondere Fachbegriffe mit produktiven Suffixen) sind. Diese Erfahrungen beziehen sich auf allgemeine Gebrauchstexte. Es ist zu erwarten, daß in Korpora bestehend aus reinen Fachtexten vor allem Fremdwörter und Fachbegriffe mit spezifischen Suffixen auftreten, die weder im Lexikon noch in allgemeinen Korpora sehr frequent sind.

Beide Tests zeigten, daß der vierte und fünfte Vorschlag nur in seltenen Fällen die korrekte Kategorie enthält. Wenn überhaupt eine korrekte Kategorie vorgeschlagen wird, so zu durchschnittlich 90% als erster Vorschlag. Das bedeutet, daß eine Einschränkung auf die ersten drei Vorschläge die Ergebnisse kaum verschlechtert. Der Benutzer muß dann nur noch unter drei Möglichkeiten auswählen, was sicher eine Vereinfachung darstellt.

Für die automatische Klassifizierung verspricht ein Verfahren, daß nur die spezifischen Suffixe verwendet und alle Wörter, die auf diese Art nicht kodiert werden können einer manuellen Bearbeitung übergibt, eine hohe Erfolgsrate. Denn aufgrund der Konstruktion kann es im Normalfall nicht vorkommen, daß bei einer Klassifikation

nur auf Grundlage der spezifischen Suffixe keiner der Vorschläge zutrifft,[8] es ist somit eine Trefferrate von etwa 90% zu erwarten.

10 Ausblick

Eine mögliche Verbesserung dieses Verfahrens für Fachkorpora, wäre die Erstellung einer korpusspezifischen Suffix-Kategorienliste. Das heißt, statt wie oben beschrieben, die Suffix-Kategorien-Paare nur aus dem Lexikon zu extrahieren, würden diese Paare zusätzlich auch aus den lemmatisierten Wortformen des Korpus extrahiert. Damit könnte erreicht werden, daß fachbereichsspezifische Suffixe besser erfaßt werden. Im Gegenzug könnten zusätzlich die einmorphigen nativen Einträge des CISLEX schwächer gewichtet werden.

Für die automatische Klassifikation wäre es von Interesse zusätzlich zur Anordnung der möglichen Kategorien je Suffix auch die bedingte Wahrscheinlichkeit jeder dieser Kategorien für das Suffix mit abzuspeichern. Liegen diese Werte für mindestens die ersten beiden Vorschläge sehr nah beieinander, kommen also für das Suffix mindestens zwei Kategorien gleichermaßen in Frage, so deutet das entweder auf eine systematische morphologische Ambiguität hin, oder aber auf eine relativ schwache Diskriminationsfähigkeit des Suffix. Im beiden Fällen sollte dann eine manuelle Kontrolle erfolgen.

Des Weiteren kann das Verfahren durch Einbeziehung des morphologischen Kontexts verbessert werden, da dieser häufig Rückschlüsse auf die morphologischen Merkmale und damit eine bessere Grundformreduktion zuläßt.

Das Suffix -*loge* beispielsweise läßt zwei Kategorien zu:

1. *loge,.mask(NS4,NP4)* mit Grundform *loge*, wie in *Biologe*
2. *loge,.mask(NS1,NP2)* mit Grundform *log*, wie in *Nekrologe*

In einem Kontext wie *ein Xologe* kommt bei Berücksichtigung der Kongruenz nur die erste Möglichkeit in Frage, da *loge,.mask(NS1,NP2)* nur im Plural bzw. Genitiv-Singular auftritt.

[8] Wenn für ein Wort keiner der Vorschläge zutrifft, so bedeutet das, daß alle Wörter im Lexikon mit dem für die Klassifizierung ausschlaggebenden Suffix einer anderen Kategorie angehören. Das ist bei dem Umfang des Lexikons jedoch nicht zu erwarten.

Anke Kölzer, Daimler-Benz AG, Ulm

Lexana – Ein System zur Lexikon- und Grammatikanalyse für kategoriale Unifikationsgrammatiken

1 Zusammenfassung

In natürlichsprachlichen Dialogsystemen werden im Allgemeinen kontextfreie Grammatiken eingesetzt, mit deren Hilfe Äußerungen von Sprechern syntaktisch und semantisch analysiert werden können. Bedingt durch die Komplexität natürlicher Sprache ist die Entwicklung solcher Grammatiken aufwendig und es ist oft schwer nachzuvollziehen, ob die Grammatik das Gewünschte leistet. In dem vorliegenden Bericht wird ein Werkzeug zur Qualitätssicherung von unifikationsbasierten Grammatiken für natürlichsprachliche Anwendungen beschrieben. Es wurde in Form einer Diplomarbeit an der Universität Koblenz in Kooperation mit der Daimler-Benz AG in Ulm entwikkelt und für Kategoriale Unifikationsgrammatiken implementiert.

2 Einleitung

Systeme zur Verarbeitung natürlicher Sprache spielen auf dem heutigen Softwaremarkt trotz intensiver Forschung immer noch eine untergeordnete Rolle. Das liegt in erster Linie daran, daß die Komplexität natürlichsprachlicher Phänomene so hoch ist, daß meist nur Lösungen für stark eingegrenzte Problemstellungen entwickelt werden können. Nichtsdestotrotz übt die Idee vom sprachverstehenden und "`sprechenden"' Computersystem auf viele Menschen eine große Faszination aus. Der Nutzen eines solchen Systems zur Unterstützung z. B. bei zwischenmenschlicher Kommunikation zum einen, wie Online-Übersetzung bei Telefonaten, woran in Projekten wie Verbmobil gearbeitet wird (vgl. WAHLSTER 1993, 1996) und Mensch-Maschine-Kommunikation (Telefon-Banking etc.) zum anderen ist nicht von der Hand zu weisen, wodurch die Entwicklung in den Forschungslabors von Industrie und Universitäten immer wieder einen Ansporn erhält.

Ein System, das in der Lage ist, fließend gesprochene Sprache zu verarbeiten, muß unter anderem mit folgenden Problemen umgehen können, die komplexitätstheoretisch schwer zu bewältigen sind und so Echtzeitkriterien im Wege stehen:

Sprecherunabhängigkeit: Menschen haben verschiedene Stimmlagen und dialektische Einfärbungen, die von einem System einheitlich repräsentiert werden müssen.

Kontinuität: Gesprochene Sprache ist ein Fluß von Lauten; es gibt keine Pause zwischen einzelnen Wörtern. Das System muß also segmentieren können, um die sprachlichen Einheiten zu ermitteln.

Robustheit: Störgeräusche wie ein laufender Motor im Auto oder Rauschen im Telefon müssen erkannt und ausgeblendet werden, um in jedem Fall Spracherkennung zu ermöglichen.

Vokabularumfang: Für realistische Konversationsanwendungen ist ein umfangreiches Vokabular notwendig, um einem Sprecher bzw. einer Sprecherin eine Vielfalt von Formulierugsvarianten zu ermöglichen. Auch besonders in Systemen wie Fahrzeugleitsystemen muß eine Vielzahl von Städte- oder Straßennamen bekannt sein, so daß leicht ein Lexikon mit ca. 10.000 Wörtern notwendig werden kann.

Linguistische Analyse: Syntax und Semantik sprachlicher Einheiten müssen adäquat repräsentiert werden, um die Äußerungen einer Sprecherin bzw. eines Sprechers richtig analysieren zu können. Besonders schwierig dabei ist die Verarbeitung bestimmter sprachlicher Phänomene wie Ellipsen, Anaphora, Polysemie etc. Darüber hinaus sind auch pragmatische Aspekte wichtig, da sprachliche Strukturen i. a. nur im Kontext korrekt interpretiert werden können.

Dialogführung: Bei konversationsorientierten Systemen gibt es potentiell unendlich viele Möglichkeiten, was ein Benutzer sagen kann. Für alle Situationen, die denkbar sind, muß das System eine Reaktion kennen, um den Dialog fortsetzen zu können und eine passende Äußerung zu machen. Durch die hohe Komplexität ist hier eine Modellierung mit Zustandsübergangskonzepten i. a. nicht möglich. Statt dessen werden hier oft komplexe Prädikatmuster eingesetzt, die bestimmte Aktionen auslösen.

3 State of the art

Robuste Spracherkenner, die in der Lage sind, mit Phänomenen wie Sprecherunabhängigkeit und Kontinuität umzugehen, werden bereits kommerziell eingesetzt, z. B. das Sprachbediensystem *Linguatronic* von Daimler Benz, das u. a. in Mercedes-Fahrzeugen zur Bedienung von Telefongeräten, Radio, Klimaanlage etc. genutzt wird. Der Vokabularumfang realistischer Anwendungen beschränkt sich allerdings auf ca. 1000 bis 2000 Wörter. Bei größeren Lexika werden die Berechnungen so umfangreich, daß Echtzeitverhalten nicht mehr garantierbar ist. Auch in puncto linguistischer Analyse und Dialogführung ist man noch weit von dem entfernt, was Menschen beim Kommunizieren leisten. Trotzdem haben diverse Forschungsprojekte wie das ESPRIT-Projekt SUNDIAL (vgl. SIMPSON & FRASER 1993, PECKHAM 1993) Grundlagen geschaffen, in absehbarer Zeit Konversationssysteme zu entwickeln, die die Forschungsebene verlassen und kommerziell einsetzbar sind.

Aufbauend auf dem ESPRIT-Projekt SUNDIAL (vgl. vgl. SIMPSON & FRASER 1993, PECKHAM 1993) wurde als Kooperationsprojekt zwischen dem Bayerischen Forschungszentrum für Wissensbasierte Systeme (*FORWISS*) in Erlangen und der Daim-

ler-Benz AG – Forschung und Technik – in Ulm das System „SYSLID“[1] (vgl. HANRIEDER & HEISTERKAMP 1994, MECKLENBURG, HANRIEDER & HEISTERKAMP 1995, HANRIEDER ET AL. 1996) entwickelt. Diese Prolog-Implementation einer Kategorialen Unifikationsgrammatik (*Unification Categorial Grammar*, *UCG*) für das Deutsche wird zur Analyse sprecherunabhängiger, gesprochener Sprache z. B. in der Zugauskunftsdomäne und für Call-Center von Versicherungen eingesetzt.

Für Problemstellungen der Sprachanalyse gibt es sehr unterschiedliche Typen von Grammatiken. Oft liegt z. B. eine größere Menge von i. a. kontextfreien Regeln vor, die auf einem Lexikon als Tokenmenge operiert. In Verbindung mit diversen zusätzlichen Mechanismen für die verschiedenen Phänomene natürlicher Sprache beschreiben solche Grammatiken auf mehr oder minder übersichtliche Weise den Aufbau der sprachlichen Strukturen.

Bezeichnend für die in SYSLID verwendete kategoriale Unifikationsgrammatik ist es dagegen, daß die Regeln, die korrekte sprachliche Ausdrücke beschreiben, nicht als eine von konkreten Strukturen abstrahierende Menge vorliegen, sondern fast vollständig bei den einzelnen Einträgen des Lexikons kodiert sind. Dadurch ist zum einen nur noch schwer nachvollziehbar, was die Grammatik leistet, d. h. welche Strukturen überhaupt analysierbar/generierbar sind und ob diese tatsächlich korrekte Phrasen[2] des Deutschen darstellen. Zum anderen ist kaum zu ermitteln, welche Lexikoneinträge nicht erreichbar sind, d. h. in keiner Phrase vorkommen können.

Um ein System wie das vorliegende besser verstehen und warten zu können und die Performanz zu erhöhen, ist es wünschenswert, automatische Coverage-Analysen auf einem gegebenen Lexikon durchführen zu lassen. Ziel solcher Analysen ist es, die Strukturen, die einer konkreten Grammatik zugrunde liegen, transparent zu machen. Schwachstellen und Fehler sollen aufgedeckt werden. Die vorliegende Arbeit beschreibt einen Lösungsansatz für diese Aufgabenstellung. Das System *Lexana* (*Lex*ikon- und Grammatik*ana*lysesystem für Kategoriale Unifikationsgrammatiken) implementiert den vorgeschlagenen Lösungsansatz.

4 Aufgabenstellung

Da ein natürlichsprachliches Informationssystem die Anfragen/Eingaben eines Anrufers sofort verarbeiten muß, spielt die Performanz eine besondere Rolle. Insbesondere der Parser in Zusammenhang mit einem konkreten Lexikon muß schnell korrekte Ergebnisse liefern. Akzeptiert die Grammatik viele Strukturen, die ein menschlicher Sprecher ablehnen würde (Übergenerierung), so führt das zu einer Explosion des Suchraums beim Parsing und damit zu einer Verschlechterung des Laufzeitverhaltens. Auch die Gefahr einer Fehlinterpretation steigt dadurch an. Die Motivation für die Entwicklung des hier beschriebenen Analysesystems ist daher, die Schwachstellen von

---

[1] *Syn*taktisch-*S*emantische *Li*nguistikkomponente für Sprachverstehende *D*ialogsysteme.

[2] Als Phrase bezeichnet man eine aus ein oder mehreren Wörtern bestehende syntaktisch zusammengehörende Substruktur eines Satzes bzw. den Satz selbst.

Grammatiken aufzudecken, so daß ein Grammatikentwickler bzw. eine Grammatikentwicklerin Abhilfe schaffen und die Sicherheit und Performanz des natürlichsprachlichen Informationssystems erhöhen kann.

Die Aufgabenstellung für die Diplomarbeit war es daher, einen Algorithmus mit folgenden Möglichkeiten zu entwickeln und zu implementieren:

1. Automatische Generierung von Phrasen, die durch die Grammatik und ein entsprechendes Lexikon ermöglicht werden;
2. Musteranalyse: dem Benutzer/der Benutzerin soll angezeigt werden können, nach welchen Schemata solche Phrasen aufgebaut sind;
3. Ermitteln aller Wörter im Lexikon, die nicht erreichbar sind.

5 Allgemeine Beschreibung von Kategorialen Unifikationsgrammatiken

Im folgenden wird eine allgemeine Einführung in den Grammatiktyp *UCG* gegeben, der auf Arbeiten von ZEEVAT, CALDER und KLEIN (ZEEVAT 1988, ZEEVAT, KLEIN & CALDER 1987) basiert. Kategoriale Unifikationsgrammatiken arbeiten auf einem Lexikon mit sogenannten *Signs* als Einträgen. Signs setzen sich aus vier Hauptkomponenten zusammen, die zum einen Informationen über den Eintrag selbst liefern, zum anderen über die Struktur von Signs, mit denen er mittels einer Verknüpfungsregel – der Funktionalen Applikation – kombiniert werden darf, um neue Strukturen zu erzeugen. Das Ergebnis einer solchen Kombination ist wieder ein Sign. Auf diese Weise erhält man die Sprache über einem Lexikon als die *Hülle* aller Lexikoneinträge unter *Funktionaler Applikation*. Einfach strukturierte Signs, d. h. solche, die keine Kombinationspartner verlangen, sind i. a. baumartige Strukturen von Merkmalen und Werten; solche Werte sind teilweise von einem einfachen Typ, z. B. ein String, teilweise sind sie komplex strukturiert, d. h. wieder aus Merkmalen und ihren Werten zusammengesetzt. Allgemein lassen sich Signs durch gerichtete, azyklische Graphen repräsentieren. Sie haben folgenden Grundaufbau:

1. *mor*: Morphologische Repräsentation der Struktur, d. h. des Wortes oder der Phrase
2. *syn*: Syntaktische Kategorie. Sie legt über die syntaktischen Eigenschaften des Wortes/der Phrase selbst auch die Struktur möglicher Kombinationspartner (Argumente) fest
3. *sem*: Semantische Kategorie. Hier wird ein hierarchisches Typkonzept auf der Basis von *SIL* (*Semantic Interface Language*, vgl. MCGLASHAN, ANDRY & NIEDERMAIR 1990) eingesetzt, das zur Interpretation der Äußerungen dient.
4. *order*: Anordnung. Das Attribut beschreibt die Reihenfolge in der Satzglieder auftreten können.

Beispiele für Signs sind in den Abbildungen 2 bis 5 gegeben.

Def: Der Begriff *Sign* ist induktiv definiert wie folgt. Vorgegeben sei genau eine für die aktuell vorliegende Grammatik ausgewählte Wertebereichsstruktur *WB*, die Aufbau und Typisierung der Signs festlegt (vgl. Abb. 1).

- *Einfach aufgebaute Signs ohne Argumente:*
  Eine Instanz *IB*, die wie in *WB* vorgegeben strukturiert und typisiert ist und bei der das Merkmal *syn:args* nicht vorkommt, ist ein *Sign*.
- *Komplex strukturierte Signs, deren Struktur selbst wieder Signs enthält:*
  Ist *Sg* ein einfach aufgebautes Sign, so ist auch *Sg* erweitert um das Merkmal *syn:args*, dem als Wert eine Menge von Signs zugeordnet ist, ein *Sign*.

Komplexe Signs heißen *Funktoren* oder *Funktorkategorien*, einfache heißen *Basis-Signs* oder *Basiskategorien*.

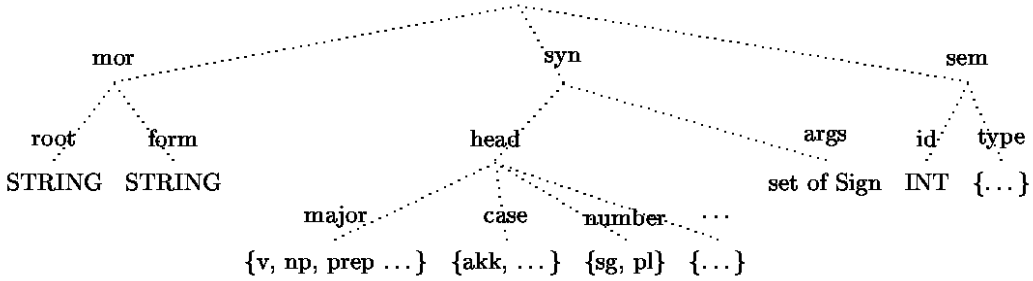


Abb. 1: Beispiel – Wertebereichsstruktur

Def: Gegeben sei ein komplexes Sign *Sg* mit der Argumentmenge *ArgSet*.
Dann heißt *ArgSet* ***aktiver Teil*** von *Sg*.

Def: Gegeben die Signs *Functor*, *Arg*, *Result*;
Result heißt ***Instanziierung*** von *Functor* in Bezug auf *Arg*, falls es aus der Unifikation eines Elementes des aktiven Teils von *Functor* mit *Arg* resultiert.

Def: Gegeben sei ein komplexes Sign *Sg* mit dem aktiven Teil *ArgSet*, in der das Sign *Arg* enthalten ist, und eine Verknüpfungsfunktion für die morphologischen Repräsentationen (z. B. Konkatenation von Strings).
Das Sign *StrSg*, dessen Wert für die morphologische Repräsentation sich aus der Verknüpfung der morphologischen Repräsentationen von *Sg* und *Arg* ergibt und dessen Argumentmenge aus *ArgSet* vermindert um *Arg* besteht, heißt ***Stripping*** von *Sg* bzgl. *Arg*.

Def: Gegeben seien zwei Signs *Functor* und *Partner*.
Dann heißt das Stripping der Instantiierung von *Functor* in Bezug auf *Partner* ***Funktionale Applikation*** von *Functor* mit *Partner*.

Def: Die morphologischen Repräsentationen aller wohlgeformten Signs heißen ***wohlgeformte Ausdrücke*** (in den Beispielen die Werte von *mor:form*).

Def: Gegeben sei ein Lexikon mit wohlgeformten[3] Signs. Die von diesem Lexikon erzeugte ***Sprache*** ist die Menge aller wohlgeformten Ausdrücke, die man durch Hüllenbildung des Lexikons unter Funktionaler Applikation erhält.

---

[3] Signs heißen wohlgeformt, wenn sie sogenannten Feature-Cooccurrence-Restrictions genügen. Diese besagen z. B., daß ein Merkmal *Kasus* nur dann auftreten kann, wenn ein Wort die syntaktische Kategorie Nomen hat.

Abb. 2 und 3 zeigen Signs für die Präposition *nach* und das Nomen *Köln*. In Abb. 4 und 5 ist dargestellt, wie die *Funktionalen Applikation* auf diesen Signs durchgeführt wird.

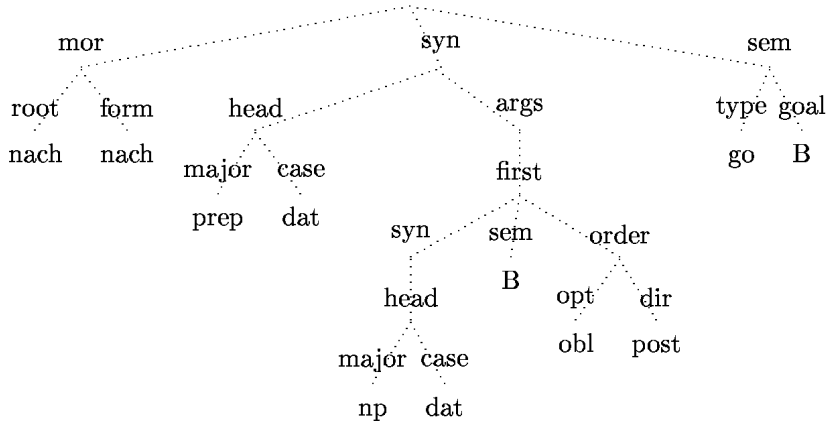


Abb. 2: Sign für die Präposition nach

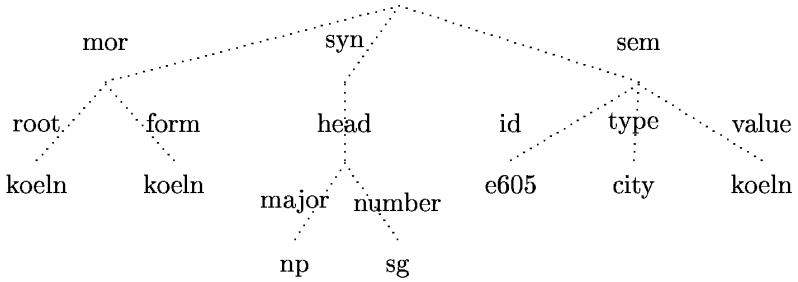


Abb. 3: Sign für das Nomen Köln

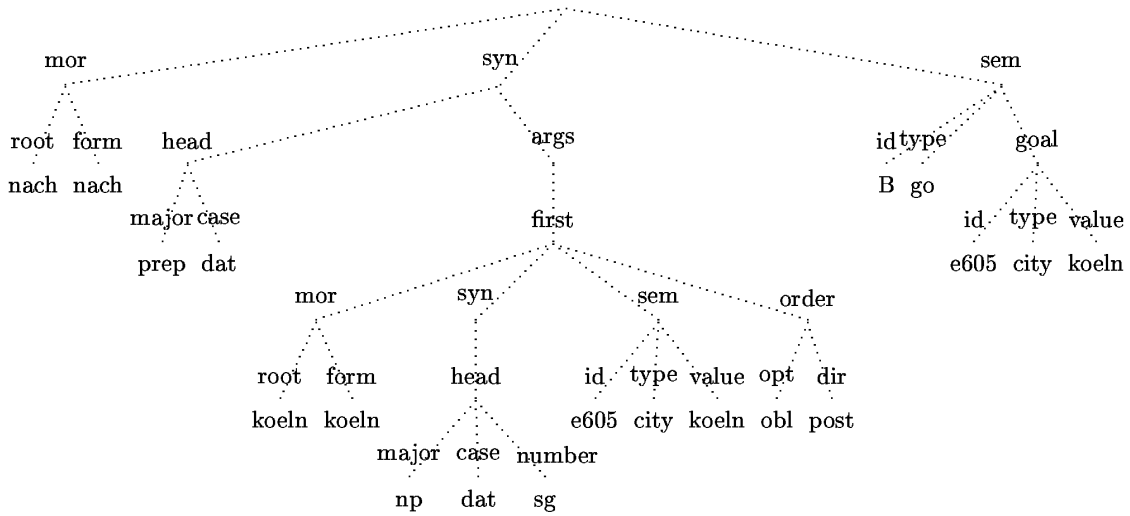


Abb. 4: Ergebnis der Instanziierung

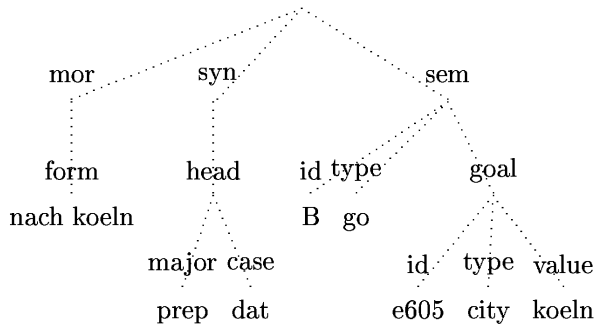


Abb. 5: Ergebnis des Stripping/der Funktionalen Applikation

6 Ein graphbasierter Lösungsansatz

Als Konzept zur Lösung der gestellten Aufgaben wurde eine graphbasierter Ansatz gewählt. Ein für jedes gegebene Lexikon berechenbarer Graph klassifiziert Signs und repräsentiert Relationen solcher Sign-Klassen untereinander. Die Vorgehensweise wird im folgenden beschrieben.

Um die Sprache über einem bestimmten Lexikon potentiell ermitteln zu können, müssen miteinander „verträgliche“ lexikalische Signs (d. h. Lexikoneinträge) gefunden werden, so daß sie mit Hilfe der Anwendung der unter Abschnitt 5 definierten Funktionalen Applikation kombinierbar sind. Die Ergebnisse der Kombinationen und deren erneute Verknüpfung mit vorhandenen Signs liefern alle gesuchten Teilphrasen. Da ein Ausprobieren der Verknüpfung von beliebigen Sign-Folgen aus dem Lexikon nicht effizient berechenbar ist, ist es sinnvoll, alle Wörter des Lexikons in Klassen einzuteilen. Diese lassen sich bestimmen durch die Attribut-Werte-Mengen, die über einem Lexikon auftreten können und die die Kombinierbarkeit der Signs untereinander festlegen. Die Attributwerte der Signs lassen sich in zwei wesentliche Teile untergliedern:

- Attributwerte, die das Sign selbst beschreiben, nämlich alle außer *syn:args*; diese bzw. die entsprechenden Attribute werden im folgenden als *Eigenattributwerte* bzw. *Eigenattribute* bezeichnet.
- Attributwerte, die mögliche Kombinationspartner beschreiben; potentielle Partner sind alle Signs in *syn:args*; *args* wird im folgenden als *Partnerattribut* bezeichnet und sein Wert als *Partnerattributwert*.

Als Basis für die Generierung/Beschreibung aller möglichen Strukturen über einem Lexikon ist ein Graph geeignet, der auf solche Sign-Informationen spezialisiert ist, d. h. der zum einen die Klassen in Bezug auf ihre Eigenattribute widerspiegelt, zum anderen die Relation jeder Klasse zu ihren möglichen Partnerklassen. Ein solcher Graph muß abhängig von einem gegebenen Lexikon nur einmal berechnet werden, kann dann gespeichert und immer wieder zur Laufzeit zur Generierung von Phrasen/Signs traversiert werden. Knoten in diesem Graphen repräsentieren Sign-Klassen und Kanten Relationen zwischen diesen Klassen. Der Aufbau der zur Repräsentation von UCG-Lexika verwendeten Graphen wird im folgenden erläutert.

Knoten: Es gibt zwei Arten von Knoten:

- *Basisknoten:* Knoten, die die durch die Eigenattribute der Signs ermöglichten Klassen repräsentieren. Diese Sign-Klassen werden im folgenden auch als *Basisklassen* bezeichnet. Eine Basisklasse faßt eine Menge von Signs im Lexikon zusammen, die die gleichen Attributwerte bzgl. einer vorgegebenen Teilmenge[4] der Eigenattribute haben. Attribute dieser Knoten sind die Wertebelegung dieser Teilmenge und ein Klassenname.
- *Spezialisierungsknoten:* Knoten, die Sign-Klassen repräsentieren, die Spezialisierungen der Basisklassen darstellen. Die entsprechenden Sign-Klassen werden im folgenden als *Spezialisierungsklassen* bezeichnet. Signs, die in einer Spezialisierung zusammengefaßt werden, haben sowohl eine Übereinstimmung auf der vorgegebenen Teilmenge der Eigenattribute als auch auf den von ihnen geforderten Argumenten. Eine solche Spezialisierung ist damit festgelegt durch die von einem Sign verlangte Argumentmenge, also durch das Partnerattribut. Die verschiedenen von einem Basisknoten abgeleiteten Knoten symbolisieren also Klassen von Signs, deren Eigenattribut-Wert-Zuordnung gleich ist und die sich in der Menge der geforderten Argumente unterscheiden. Diese Knoten haben als Attribut eine Menge von Verweisen auf konkrete Signs im Lexikon, die in diese Klasse fallen. Alle im Lexikon vorhandenen Signs lassen sich solchen Spezialisierungsknoten zuordnen.

Kanten: Die Kanten symbolisieren Relationen zwischen diesen Klassen.

Spezialisierungskanten: Sie zeigen an, daß eine Basisklasse sich in Spezialisierungsklassen unterteilt. Dementsprechend handelt es sich um gerichtete Kanten von Basisknoten zu Spezialisierungsknoten.

Argumentkanten: Da jedes durch den Partnerattributwert gegebene Argument selbst wieder ein Sign ist und damit unter eine der durch die Basisknoten gegebenen Klassen fällt, werden von den Spezialisierungsknoten gerichtete Argumentkanten zu allen potentiellen Argumentklassen, also den für solche Argumente stehende Basisknoten, gezogen. Da es für die Berechnung neuer Signs wichtig ist, wie die Signs anzuordnen sind (d. h. wie das *order*-Attribut belegt ist), erhalten diese Kanten ein Attribut, das darüber Auskunft gibt.

Kompatibilitätskanten: Die Signs im Partnerattributwert eines anderen Signs beschreiben eine Mindestanforderung an die Partner. Es ist nur verlangt, daß ein Partner mit einem Sign aus dieser Menge unifizierbar ist. Er kann also bzgl. der vorgegebenen Attribute unterspezifiziert sein oder auch mehr Attribute haben als gefordert. Um also bei der Generierung neuer Signs alle potentiellen Partner zu finden, muß Information darüber vorliegen, welche Klasse bzgl. der relevanten Attributwerte

[4] Hier ist die Vorgabe einer Teilmenge notwendig, um die relevanten Gemeinsamkeiten der im Lexikon vorhandenen Signs, die in eine Klasse fallen sollen, festzulegen. Außerdem entscheiden einige bestimmte Eigenattribute maßgeblich über die Kombinierbarkeit von Signs.

mit welcher unifizierbar und damit kompatibel ist. Dazu dient dieser unattributierte, ungerichtete Kantentyp zwischen Basisknoten.

Relevante Signs

Wie oben erwähnt, ist es notwendig, für die Einteilung der Lexikon-Signs in Klassen eine Menge von Eigenattributen vorzugeben. Durch diese Menge wird zum einen festgelegt, wie granular die Klassengenerierung durchzuführen ist (wenig Attribute → wenig Klassen, viele Attribute → viele Klassen), zum anderen, auf welche Gemeinsamkeiten von Signs der Benutzer Wert legt.

Genaugenommen handelt es sich nicht um eine Attributmenge, sondern um eine Menge von Attributpfaden – im folgenden *RelPaths* für *relevant paths* genannt. Daß hier Pfadmengen anstelle von Attributmengen gefordert sind, liegt darin begründet, daß bestimmte Attributbezeichner in einem Sign mehrfach auftreten können. Durch die Pfade wird exakt festgelegt, welche Stellen gemeint sind.

Eine sinnvolle Pfadmenge[5] ist z. B. die Menge aus allen syntaktischen Eigenattributen, gegeben durch *syn:head*, und das Eigenattribut *sem:type*. Diese Attribute sind für die Kombinierbarkeit von Signs ausschlaggebend:

$$\textbf{\textit{RelPaths}} = \{\textit{syn:head}, \textit{sem:type}\}$$

Alle Lexikon-Signs, die bzgl. der Werte dieser Attribute gleich sind, fallen in eine Klasse, d. h. unter einen der oben beschriebenen Basisknoten.[6] Wenn man aus einem Lexikon-Sign genau die durch *RelPaths* festgelegten Attribute extrahiert, erhält man wiederum ein Sign, das dann ein Knotenattribut der Basisknoten darstellt. Für dieses Sign wird im folgenden der Begriff *Relevantes Sign* verwendet.

Generierung von Graphinstanzen

Folgende Vorgehensweise beschreibt das Generieren der Knoten aus einem gegebenen Lexikon:

- Vergleiche für jedes Sign im Lexikon, ob eine Klasse, d. h. ein Basisknoten, existiert, der das gleiche Relevante Sign hat wie dieses Sign.
  - Falls nicht, generiere einen solchen mit diesen Attributen.
- Untersuche, ob der entsprechende Knoten schon eine Spezialisierung hat, der die gleiche Argumentmenge hat wie das aktuell untersuchte Sign.
  - Falls nicht, generiere eine solche Spezialisierung.
- Ordne das Sign dem Spezialisierungsknoten zu.
- Für jedes Argument des untersuchten Signs untersuche, ob schon ein Basisknoten mit dieser Attribut-Wert-Kombination bzgl. *RelPaths* existiert.
  - Falls nicht, generiere einen solchen.

---

[5] Da je nach gewünschter Strukturanalyse auch andere Pfadmengen sinnvoll sein können, ist das relevante Sign in *Lexana* einstellbar.

[6] Genaugenommen werden sie einer Spezialisierung dieses Basisknotens zugeordnet.

Die Kanten zwischen den entstandenen Klassenknoten erhält man folgendermaßen:
– Generiere von den Basisknoten zu jeder ihrer Spezialisierungen eine Spezialisierungskante.
– Generiere von jeder Spezialisierungsklasse eine Argumentkante zu allen Basisknoten, die von dieser Spezialisierung als Argument gefordert sind. Versehe die Kanten mit dem Sign als Attribut, das aus dem *order*-Merkmal des jeweiligen Arguments und dem dazugehörigen Wert aufgebaut ist.
– Untersuche paarweise alle Basisknoten auf Unifizierbarkeit der Werte bzgl. *RelPaths* und generiere entsprechend Kompatibilitätskanten.

6.1 Generierung von Phrasen/Signs

Anhand eines gegebenen Lexikons kann eine Instanz der beschriebenen Graphklasse generiert werden. Mit Hilfe dieser Graphen ist folgende Vorgehensweise möglich:

Der Benutzer wählt eine Basisklasse aus. Die für die Ermittlung aller zulässigen Strukturen zu dieser Klasse relevanten Signs sind die *Basis-Signs*. Diese zeichnen sich dadurch aus, daß sie keine obligatorischen Argumente offen haben. Ziel ist es also, solche Basis-Signs zu finden. Zum einen sind Signs auszugeben, die in einer der Spezialisierungen dieser Basisklasse enthalten sind und keine obligatorischen Argumente haben. Zum anderen müssen Signs mit obligatorischen Argumenten zu Basis-Signs expandiert werden, indem man passende Argument-Signs berechnet und diese einsetzt.

Generierung der Basis-Signs einer Klasse (*GenSigns*):
- Gib alle Signs aus, die bereits in einer Spezialisierung dieser Klasse vermerkt wurden und Basis-Signs sind.
- Generiere zu allen Signs *Sg* in den Spezialisierungen dieser Klasse weitere Basis-Signs:
  – Bestimme für alle obligatorischen Argumente von *Sg* die Menge der Klassen, die zu der Argumentklasse kompatibel sind
  – Generiere für alle diese Klassen Signs (Rekursion – Aufruf *GenSigns*) und verknüpfe sie mit Hilfe der Funktionalen Applikation mit *Sg*.
  – Gib die durch diese Verknüpfung entstandenen Signs aus.

Ablaufbeispiel

Die Abb. 6 zeigt einen Beispielgraphen und seine Traversierung zur Sign-Generierung für die Klasse *v_ fin_1*. Die Pfeile mit ihren Nummern zeigen die Reihenfolge der Traversierung. Die Klassennamen für diesen Graph sind willkürlich mit Hilfe der zugehörigen Attributwerte konstruiert. *v_ fin_1* umfaßt z. B. bestimmte finite (*fin*) Verben (*v*). Um die Phrase „ich will eine Auskunft“ zu generieren, wird folgendes u. a. durchgeführt:

– *v_ fin_1* hat eine Spezialisierung, in der aber keine Signs enthalten sind, die eine „Gültige Phrase“ darstellen. Also müssen die Argumentkanten verfolgt werden, um die nötigen Argument-Signs zu generieren.

– Über die Argumentkanten findet man eine Beschreibung für das erste Argument – das Subjekt *np_nom_3*. Dieser Klasse ist keine Spezialisierung zugeordnet, durch die man ein passendes Argument finden könnte. Also müssen die hierzu kompatiblen Klassen untersucht werden.
– Über die Kompatibilitätskante gelangt man zu der Klasse *np_nom_1* (Pfeil 3). Hier existiert eine Spezialisierung mit dem Eintrag *ich*, die keine weiteren Argumente obligatorisch verlangt.
– Damit ist das erste potentiell passende Argument-Sign gefunden und wird durch funktionale Applikation mit dem ersten Sign in der Spezialisierung von *v_ fin_1 – will* – verknüpft.
– Dieses Teilergebnis hat noch ein offenes Argument, also muß ein Sign für dieses generiert werden. Die Pfeile 5 ff. zeigen die Abfolge.

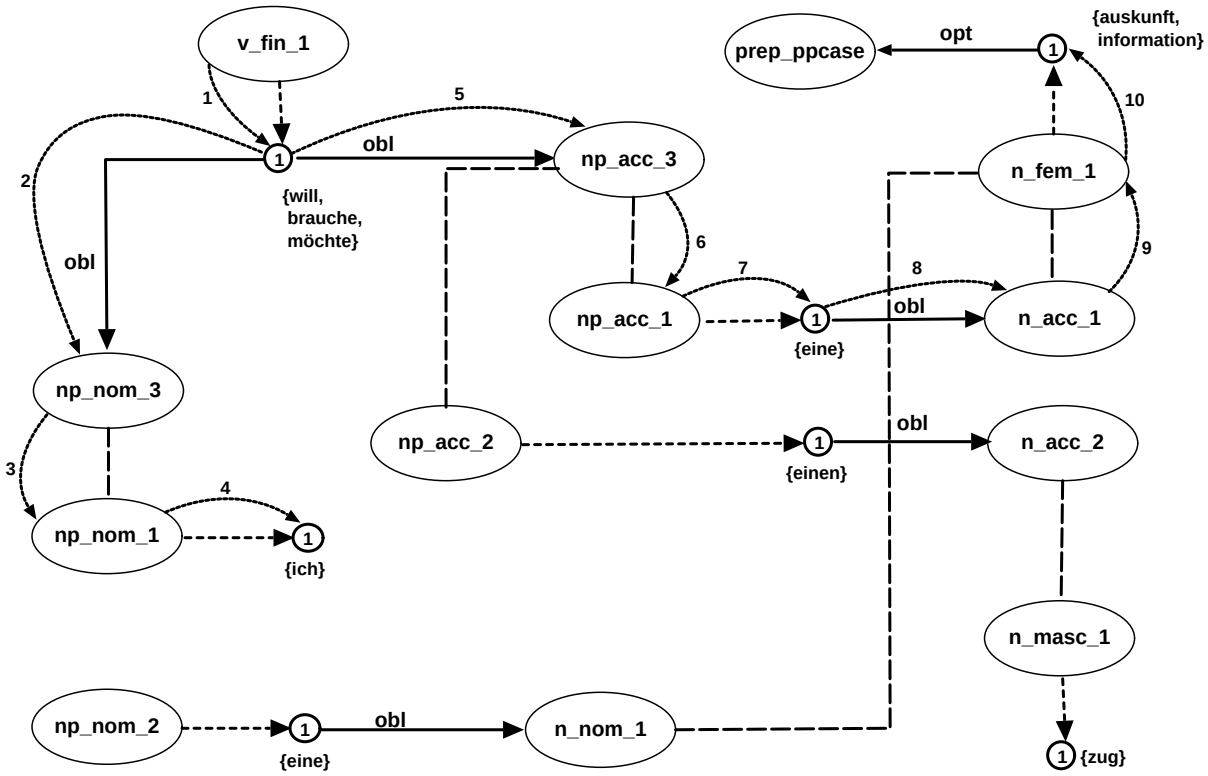


Abb. 6: Traversierung

7 Abschließende Bemerkungen

Mit den beschriebenen Algorithmen konnten die gestellten Aufgaben gelöst werden. *Lexana* ermöglicht es, automatisch Phrasen für ein gegebenes Lexikon zu generieren und eine Fehleranalyse für die im Lexikon enthaltenen Strukturen durchzuführen. Einer Grammatikentwicklerin bzw. einem -entwickler ist somit ein Werkzeug gegeben, das sowohl während der laufenden Konzipierung als auch nach der abgeschlossenen Spezifikation einer Grammatik und der damit verbundenen Lexika ständige Kontrolle ermöglicht.

Der beschriebene Ansatz ist auf verwandte Grammatiktypen, die merkmals- und unifikationsbasiert arbeiten, grundsätzlich übertragbar. Aktuell ist eine Anpassung für eine von *FORWISS* entwickelte Phrasenstrukturgrammatik geplant.

Weitere Details z. B. zur Behandlung von Zyklen bei der Graphtraversierung und zur Komplexitätsreduktion der Algorithmen sind in der Diplomarbeit (siehe KÖLZER 1997) beschrieben.

Sebastian Göser, IBM Deutschland Entwicklung GmbH, Böblingen

High Speed Parsing of Extraction Grammars: The ExGram Approach

As information extraction components become part of industrial applications, there is a need for robust, highly efficient extraction methods. Extractors are to be applied to gigabytes of real-world text. Information extraction has significantly advanced through the MUC series of conferences (see e.g. GRISHMAN and SUNDHEIM, 1996). Traditional, "full-fledged" parsing methods have been criticized repeatedly both with respect to correctness of MUC results achieved and their performance characterisitics (see e.g. APPELT et al. 1993). Partial parsing methods, on the other hand, have been considered successful in correctness and performance on a number of extraction tasks.

The ExGram framework discussed in the talk embodies high-speed text extraction, and extractor building methods on the basis of the ACSG-P formalism. ACSG-P is a procedural derivative of a unification grammar framework for linguistic robustness (GOESER 1992). As an annotated constituent structure grammar, ACSG-P consists of a context-free component with symbol annotations describing a kind of functional structures, namely, recursive predicate frames with functional slots typed either for arguments or modifiers.

The ExGram system includes a parser and a grammar compiler for extraction grammars as well as a number of facilities to combine and extend a given extractor. The parser scales up to the multi-megabyte range. Due to its robustness features, ExGram grammars can be applied to extraction from multi-topical and severely mal-edited text, such as e.g. e-mail correspondence. At the same time, the ACSG-P formalism is general in the sense that quite different extraction tasks in different languages, using different grammar writing styles, may be solved using its facilities.

The talk started with an overview of the ExGram framework, and its underlying ACSG-P formalism. Then, details were given about the ExGram framework, emphasizing the robustness and performance features of the parser. The main focus was on the concept of rule-based approach within a framework. To illustrate the capabilities of the ExGram system, two example extractors, namely, an English extractor for simple facts on product-producer relationships, and a Spanish extractor for proper names were described, thereby trying to substantiate the performance claims through one extraction experiment for each of them. The talk concluded with first results in semi-automatically extending extraction grammars and an outlook to ExGram's further development.

Andreas Mertens, FernUniversität Hagen

Robustes Parsing mit Wortagenten

1 Einleitung

Die Akzeptanz eines natürlichsprachlichen Systems beim Benutzer läßt sich durch den Einsatz eines robusten Parsingverfahrens erheblich erhöhen. Robuste Sprachverarbeitung bedeutet für die Analyse vor allem, daß sie vom Benutzer eingegebene unbekannte Wörter und ungrammatische Ausdrücke verarbeiten kann.[1] Im Mittelpunkt dieses Beitrags steht die Verarbeitung unbekannter Wörter im Informationssystem LINAS[2], das es dem Benutzer einer Literaturdatenbank ermöglicht, seine Anfragen natürlichsprachlich per Tastatur einzugeben (vgl. MERTENS, SCHULZ & HELBIG 1995). Für das Fehlen von lexikalischen Informationen über eingegebene Wörter sind u. a. folgende Faktoren verantwortlich:

- Unvollständigkeit des Lexikons;
- Eigennamen;
- Rechtschreib- bzw. Tippfehler in der Benutzereingabe;
- Groß- vs. Kleinschreibung;
- Neubildungen (z. B. Komposita, Ad-hoc-Ausdrücke).

Ein Teil der unbekannten Wörter bzw. Wortformen wird bereits durch die morphologisch-lexikalische Analyse behandelt, da sie in der Lage ist, Endungs- und Wortbildungshypothesen zu liefern. Darüber hinaus werden unter Zugriff auf grammatisches Wissen für morphologisch nicht erfolgreich analysierte Eingabewörter Wortklassenhypothesen aufgestellt, wobei für die offenen Wortklassen der Adjektive, Nomen und Verben konkurrierende Wortagenten aktiviert werden. Die daraus resultierenden alternativen Interpretationen werden heuristisch bewertet und die wahrscheinlichste Hypothese wird ausgewählt.

2 Wortorientiertes Parsing

Im Bereich der automatischen Sprachverarbeitung gibt es Analyseansätze, die die Repräsentation grammatischen Wissens in Form von Wortexperten vorschlagen. Dazu

---

[1] Spezielle Probleme, die sich beispielsweise im Bereich der akustischen Erkennung ergeben (vgl. z. B. MECKLENBURG, HEISTERKAMP & HANRIEDER 1995), werden hier bewußt ausgeklammert. Zur Diskussion des Robustheitsbegriffs in der maschinellen Sprachverarbeitung vgl. z. B. MENZEL 1995.

[2] Das Akronym LINAS steht für **L**iteraturrecherche **i**n **na**türlicher **S**prache.

gehören u. a. das von SMALL entwickelte Wortexperten-Parsing, die wortklassengesteu-erte funktionelle Analyse von HELBIG, das wortorientierte Parsen von EIMERMACHER und das lexikalisch verteilte Text-Parsing von HAHN. Bei allen hier genannten Ansätzen handelt es sich um wortorientierte Analyseverfahren. In der Theorie des Wortexperten-Parsing wird die Verarbeitung natürlicher Sprache als verteilter Prozeß von interagierenden Wörtern betrachtet. Jedes Wort stellt einen aktiven Prozeß dar, den sog. Wortexperten, der unter Zuhilfenahme von linguistischem und Weltwissen seine eigene Bedeutung im aktuellen Kontext mit anderen Wortexperten aushandelt (vgl. SMALL 1987). Das von EIMERMACHER entwickelte wortorientierte Parsen basiert auf der Idee der Wortexperten, unterscheidet allerdings zwischen Wortklassenexperten, die die grammatischen Informationen repräsentieren, und Wortexperten, die die Beziehungen zwischen den einzelnen Wörtern eines Satzes analysieren (vgl. EIMERMACHER 1988).

Neuere Ansätze verbinden wortzentrierte Sprachbeschreibungsmodelle mit einer agentenorientierten Modellierung der Sprachverarbeitungsprozesse. Bei der Analyse mit Wortaktoren im *ParseTalk*-Modell von HAHN, SCHACHT und BRÖKER handelt es sich um ein objektorientiertes Parsing-System, in dem das grammatische Wissen vollständig in das Lexikon integriert ist (vgl. HAHN, SCHACHT & BRÖKER 1994). Jedes Wort eines gerade analysierten Satzes aktiviert einen Wortaktor, der mit anderen bereits initialisierten Wortaktoren kommuniziert, um so die Gesamtbedeutung des Satzes zu konstituieren. Beim Analyseansatz mit Wortagenten (vgl. HELBIG & MERTENS 1994) besteht die Grundidee darin, daß jedes Wort syntaktische und semantische Erwartungen auslöst, die erst im weiteren Verlauf der Analyse erfüllt werden können. Nach Einbeziehung weiterer Wörter in die Verarbeitung werden die hinzukommenden Informationen daraufhin überprüft, ob sie zu den ausgelösten Erwartungen passen. Dieser Analyseansatz wird zur Zeit zu einem robusten Parsingverfahren weiterentwikkelt und besitzt u. a. folgende Eigenschaften:

- Der Analyseansatz ist prozedural.
- Wortagenten verarbeiten wortklassen-, wort- und wortformspezifisches Wissen.
- Die Verarbeitung verläuft wortweise inkrementell von links nach rechts.
- Unbekannte Wörter werden als Endungs-, Wortbildungs- bzw. Wortklassenhypothesen repräsentiert und robust verarbeitet.
- Die Desambiguierung wird durch heuristische Verfahren unterstützt.

Für jedes vom Benutzer eingegebene Wort wird ein Wortagent aktiviert, der über wortklassenspezifisches Wissen verfügt (z. B. die für die Wortklasse des Eingabeworts gültigen Grammatikregeln) und der durch die morphologisch-lexikalische Analyse mit wortspezifischer Information (z. B. semantische Sorten und Merkmale, Subkategorisierungsinformation) und wortformspezifischer Information (z. B. morpho-syntaktische Merkmale für Kongruenzüberprüfungen) angereichert wird. Bei lexikalischen Mehrdeutigkeiten (z. B. Polysemie, Homographie) gibt es jeweils für die einzelnen Lesarten zugehörige Wortagenten, die untereinander in Konkurrenz stehen und unter-

schiedliche Interpretationen der Eingabe erzeugen. Diese Möglichkeit zur Aktivierung von konkurrierenden Wortagenten ist eine wesentliche Voraussetzung für die Verarbeitung von alternativen Wortklassenhypothesen bei unbekannten Wörtern (vgl. Abbildung 4).

3 Lexikon und morphologisch-lexikalische Analyse

Die Lexikoneinträge enthalten u. a. morphologisches (MOR), syntaktisches (SYN) und semantisches Wissen (SEM) sowie gemischte Information für die Subkategorisierung (SEL). Einen Ausschnitt des Lexikoneintrags zum Verb *schreiben* zeigt Abbildung 1. Ausgewählte lexikalische Information soll an diesem Beispiel näher erläutert werden.

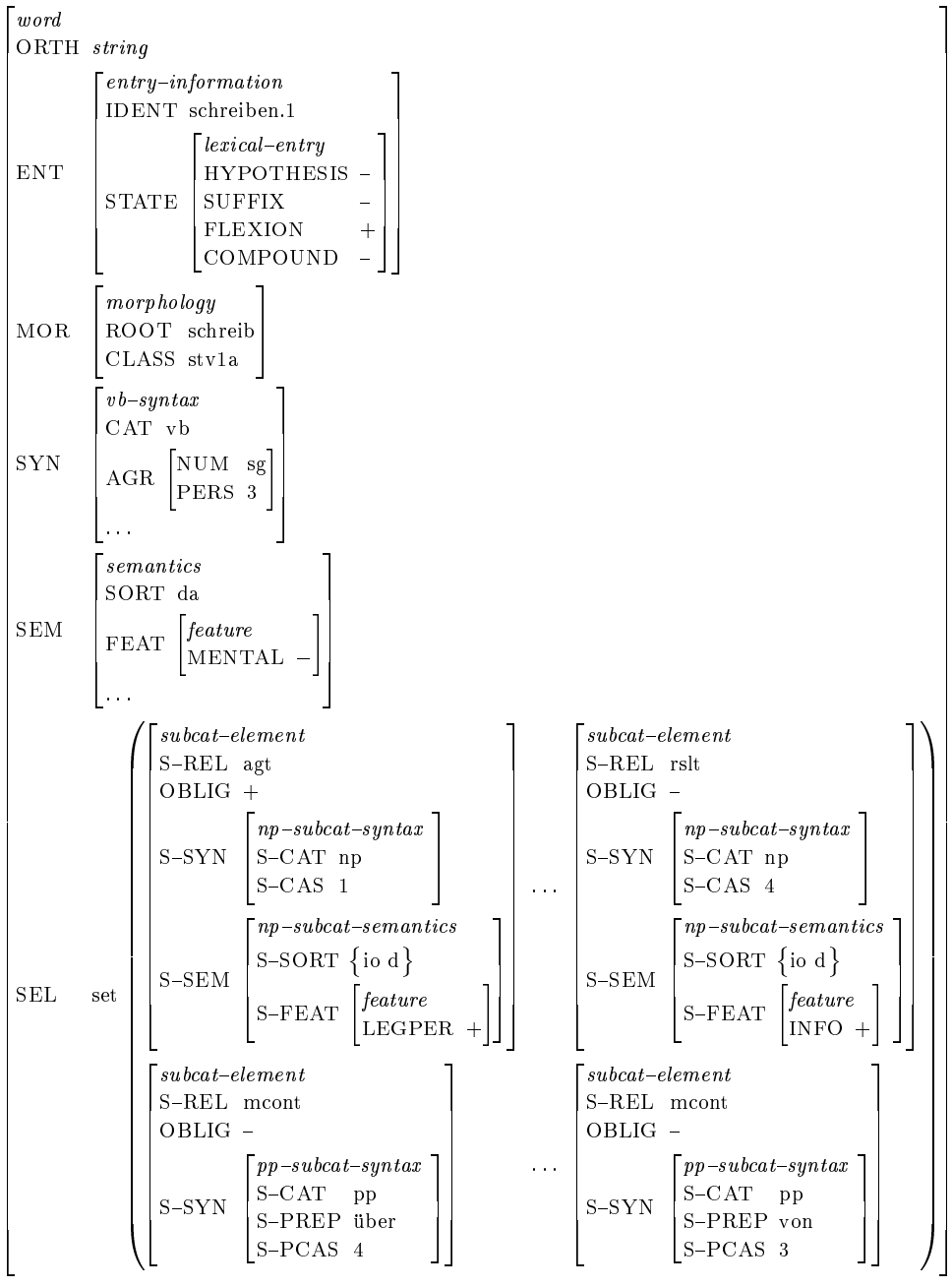


Abb. 1: Lexikoneintrag zu schreiben *(Ausschnitt)*

Das Merkmal ORTH, dessen Wert vom Typ *string* sein muß, dient zur Verwaltung der aktuellen Wortform (z. B. *schreiben*). ENT beinhaltet neben dem unter IDENT verzeichneten Verweis auf das zugehörige Konzept `schreiben.1` spezielle Information darüber, ob es sich bei der gesamten Merkmalstruktur um sicheres bzw. unsicheres Wissen handelt. So enthalten Lexikoneinträge unter dem Merkmal STATE eine Merkmalsstruktur vom Typ *lexical-entry*, der u. a. angibt, daß es sich um sicheres morphologisch-lexikalisches Wissen handelt. Demgegenüber werden Merkmalstrukturen, die die morphologisch-lexikalische Analyse aufgrund von Endungs-, Wortbildungs- und Wortklassenhypothesen erzeugt, unter STATE mit Merkmalstrukturen vom Typ *hypothetical-entry* versehen. Eine Gegenüberstellung der jeweils unterschiedlich belegten Merkmalwerte unter STATE zeigt Abbildung 2 beispielhaft für die lexikalische Merkmalstruktur von *neuer*, für die Endungshypothese *kroxeliges*, für die Wortbildungshypothese *Computerprogramm* und für das unbekannte Wort *Kroxel*.

lexical–entry	
HYPOTHESIS	–
SUFFIX	–
FLEXION	+
COMPOUND	–

neuer

hypothetical–entry	
HYPOTHESIS	+
SUFFIX	+
FLEXION	+
COMPOUND	–

kroxeliges

hypothetical–entry	
HYPOTHESIS	+
SUFFIX	–
FLEXION	–
COMPOUND	+

Computerprogramm

hypothetical–entry	
HYPOTHESIS	+
SUFFIX	–
FLEXION	–
COMPOUND	–

Kroxel

Abb. 2: Morphologisch-lexikalisches vs. hypothetisches Wissen

Unter dem Merkmal MOR ist der Stamm (schreib) und die Flexionsklasse (stv1a) eingetragen. SYN enthält unter CAT die Kategorie (Verb V) und unter AGR die Agreement-Information. Semantische Information wird unter dem Merkmal SEM eingetragen. Dazu gehört insbesondere die semantische Sorte (dynamischer Vorgang, der aktiv von einem Handlungsträger bewirkt wird; Sorte da) und die Information, daß es sich bei der Handlung um einen physischen Vorgang handelt ([MENTAL -]). Gemischte syntaktische und semantische Information zur Argumentstruktur steht unter dem Merkmal SEL. Im Eintrag von *schreiben* (vgl. Abbildung 1) sind beispielhaft der Handelnde (semantische Relation „Agent“ agt) und das Produkt der Handlung (semantische Relation „Result“ rslt) angegeben. Das erste Argument ist obligatorisch (OBLIG +) und wird durch eine Nominalphrase (S-CAT np) im Nominativ (S-CAS 1) realisiert. Es muß von der Sorte *ideelles Objekt* (*io*) oder *Diskretum* (*d*) sein und zusätzlich das Merkmal *Legale Person* ([LEGPER +]) erfüllen. Das zweite angegebene Argument ist fakultativ (OBLIG -) und wird durch eine Nominalphrase im Akkusativ realisiert. Es muß ebenfalls von der Sorte io oder d sein und zusätzlich das Merkmal „Informationsträger“ ([INFO +]) erfüllen. Darüber hinaus sind in dem Lexikoneintrag noch beispielhaft zwei fakultative Argumente eingetragen, die durch Präpositionalphrasen (eingeleitet durch *über* bzw. *von*) realisiert werden. Es handelt sich dabei um den Tiefenkasus mcont (geistiger Inhalt).

Für jedes eingegebene Wort bzw. für jede eingegebene Wortform aktiviert die morphologisch-lexikalische Analyse[3] unter Zugriff auf die lexikalischen Merkmal-

[3] Als morphologische Analyse wird eine erweiterte Version der in LISP implementierten Analyse MORPH von HANRIEDER eingesetzt (vgl. z. B. HANRIEDER 1991, 1994).

strukturen einen bzw. im Falle von lexikalischen Mehrdeutigkeiten mehrere Wortagenten, die jeweils über wortklassen-, wort- und wortformspezifisches Wissen verfügen. Falls bei einer unbekannten Wortform eine Endungs- oder Wortbildungshypothese zutrifft, so wird ein als Endungs- bzw. Wortbildungshypothese markierter Wortagent erzeugt, der mit einer entsprechenden Merkmalstruktur angereichert wird. Für den Fall, daß die morphologisch-lexikalische Analyse für ein unbekanntes Eingabewort keine Endungs- bzw. Wortbildungshypothese erzeugen kann, werden mehrere Wortagenten aktiviert, die jeweils als Wortklassenhypothese gekennzeichnet und mit entsprechenden Merkmalstrukturen versehen werden (vgl. Abbildung 5).

4 Robustes Parsing unbekannter Wörter

4.1 Grundprinzipien

Jeder durch die morphologisch-lexikalische Analyse aktivierte Wortagent erhält als Eingabe eine Menge von Merkmalstrukturen, die jeweils die alternativen Analyseergebnisse der vorhergehenden Wortagenten repräsentieren. Diese Menge faßt die Erwartungen der bereits analysierten Wörter zusammen (vgl. EXP_i in Abbildung 3). Aufgrund von wortklassen-, wort- und wortformspezifischem Wissen werden die übergebenen Strukturen ausgewertet.

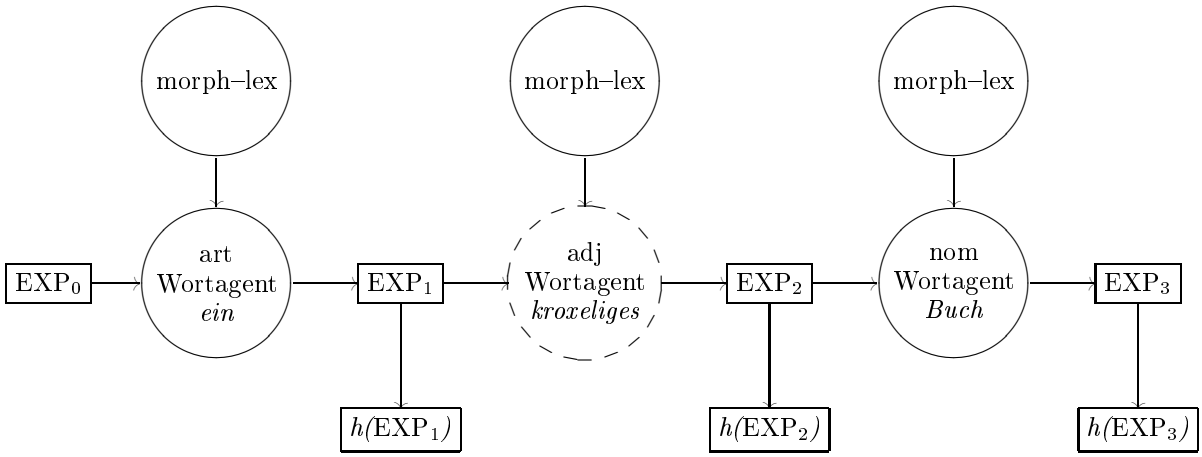


Abb. 3: Aktivierung der Wortagenten und Weitergabe der Erwartungen

Dabei versuchen die Wortagenten die von den Vorgängern bereits eingetragenen Erwartungen zu erfüllen (z. B. Komplettierung einer Nominalphrase, Anbindung einer Konstituente an das Verb) bzw. diese zu neuen Erwartungen zu verfeinern (z. B. Erweiterung einer bereits durch einen Artikel eröffneten Nominalphrase durch ein Adjektiv). Das Ergebnis wird dann an den nachfolgenden Wortagenten weitergegeben. Am Satz- bzw. Textanfang gibt es keine Erwartungen ($EXP_0 = \varnothing$). Somit kann der Wortagent, der das erste Eingabewort repräsentiert, keine Erwartungen erfüllen, sondern nur die eigenen Erwartungen in Form einer Merkmalstruktur an den nachfolgenden Wortagenten weiterreichen. In Abbildung 3 ist die Aktivierung der Wortagenten durch die morphologisch-lexikalische Analyse und das Weiterreichen der wortweise inkrementell erzeugten Analyseergebnisse EXP_i für die Eingabephrase *ein kroxeliges Buch* dargestellt. Das Ergebnis der morphologischen Analyse der Wortform *kroxeliges*

ist eine Adjektiv-Endungshypothese (vgl. den in Abbildung 3 gestrichelt eingezeichneten Wortagenten).

Die Ausgabestruktur eines Wortagenten EXP_i wird jeweils heuristisch ausgewertet und die dabei ermittelte optimale Lesart *h(*EXP_i*)* wird ausgegeben.[4] Dem nachfolgenden Wortagenten werden allerdings alle potentiell möglichen Lesarten übergeben, damit es möglich bleibt, daß bestimmte, zunächst unwahrscheinliche Erwartungen durch nachfolgende Wortagenten erfüllt werden können (z. B. bei erweiterten Partizipialkonstruktionen: [der [junge Schüler]$_{np}$ unterrichtende Lehrer Müller]$_{np}$).

Ist die eingegebene Wortform lexikalisch mehrdeutig, so werden konkurrierende Wortagenten aktiviert, die alternative Interpretationen erzeugen. Dieser Mechanismus greift ebenfalls bei solchen unbekannten Wörtern, für die die morphologisch-lexikalische Analyse keine Endungs- bzw. Wortbildungshypothesen erstellen kann. In diesem Falle werden Wortagenten für die offenen Wortklassen der Adjektive, Nomen und Verben aktiviert. Abbildung 4 zeigt jeweils die durch die Mehrdeutigkeit der Wortform *der* und die durch das unbekannte Wort *Bich* ausgelöste Aktivierung von konkurrierenden Wortagenten für die Eingabephrase *das neue Bich*.[5] Alternative Wortagenten erzeugen zunächst eigene Ergebnisstrukturen, die dann wieder als Menge von komplexen Merkmalstrukturen zusammengefaßt und an den nachfolgenden Wortagenten weitergegeben werden (vgl. Abbildung 4).

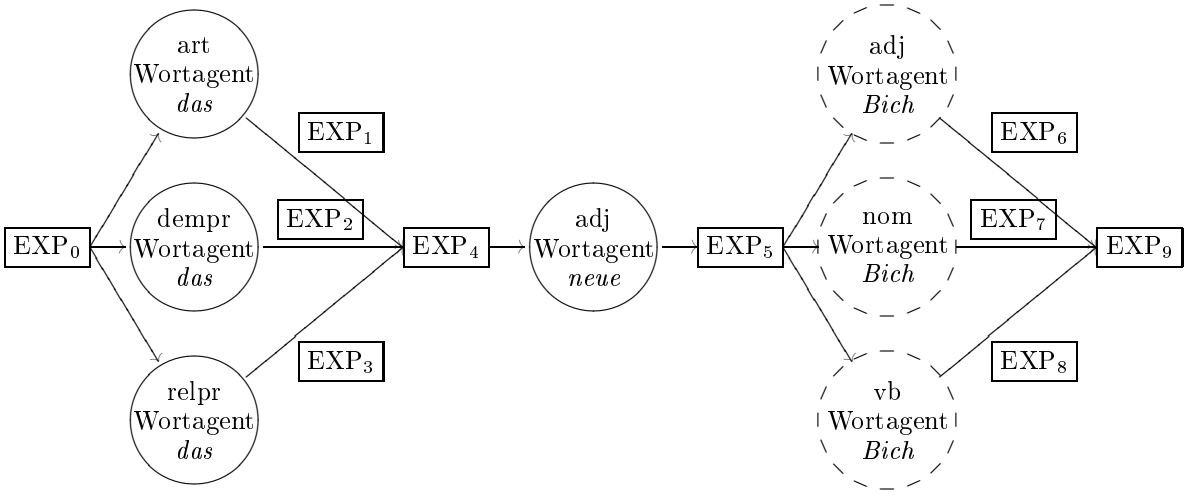


Abb. 4: Konkurrierende Wortagenten bei mehrdeutigen und unbekannten Wörtern

4.2 Aufgaben der Wortagenten

Der wesentliche Unterschied zwischen Wortagenten, die aufgrund von morphologischer bzw. grammatischer Information als Hypothesen gekennzeichnet werden, und

[4] Evtl. werden auch mehrere alternative Interpretationen ermittelt.

[5] Die in Abbildung 4 gestrichelt eingezeichneten Wortagenten repräsentieren Wortklassenhypothesen. Aus Gründen der robusten Verarbeitung wird die Groß- und Kleinschreibung erst bei der heuristischen Bewertung der insgesamt erzeugten Strukturen als zusätzliche Entscheidungsgrundlage herangezogen. Die heuristischen Bewertungen *h(*EXP_4*)*, *h(*EXP_5*)* und *h(*EXP_9*)* sind aus Gründen der Übersichtlichkeit in Abbildung 4 weggelassen worden.

den nicht markierten Wortagenten besteht in der für die Verarbeitung zur Verfügung stehenden Information. Bei Endungs-, Wortbildungs- und Wortklassenhypothesen sind die Merkmalstrukturen nicht voll spezifiziert. Die Arbeitsweise ist bei beiden Formen von Wortagenten jedoch prinzipiell gleich. Typische Aufgaben der Wortagenten bei der Bearbeitung der Erwartungsstrukturen sind beispielsweise:

- Kongruenzüberprüfungen (bzgl. Kasus, Genus, Numerus, Person) und wortweise inkrementeller Aufbau von syntaktischen Strukturen (vgl. Abbildung 6);
- Markierung der erzeugten (Teil-)Strukturen mit grammatischen Informationen (Erweiterungsregeln);
- Kompatibilitätstests (bzgl. semantischer Sorten und Merkmale) und Aufbau von semantischen Netzen (vgl. Abbildung 6);
- Verarbeitung von alternativen syntaktisch-semantischen Ergebnisstrukturen u. a. m.

4.3 Analysebeispiel

Die robuste Verarbeitung unbekannter Wörter mit Hilfe von Wortagenten soll in diesem Abschnitt an dem Beispielsatz *Der kroxelige Autor schreibt ein neues Bich*. erläutert werden. Da die morphologische Analyse für das unbekannte Wort *kroxelige* eine Adjektiv-Endungshypothese erstellen kann, wird ein entsprechend gekennzeichneter Wortagent aktiviert, der mit der in Abbildung 5 (linke Seite) angegebenen Merkmalstruktur angereichert wird. Bis auf die Sorte „graduierbare Qualität“ (gq) werden die semantischen Merkmale für die Endungshypothese nur durch die jeweiligen Typen belegt.[6]

Bei dem unbekannten Wort *Bich* trifft keine Endungs- sowie keine Wortbildungshypothese zu. Deshalb werden dafür als Wortklassenhypothesen gekennzeichnete Wortagenten erzeugt (vgl. Abbildung 4). Die zugehörigen Merkmalstrukturen, jeweils für die Adjektiv-, Nomen- und Verbhypothese, sind auf der rechten Seite in Abbildung 5 dargestellt.[7] Für die Adjektivhypothese wird aufgrund von grammatischem Wissen (Wortstellung innerhalb einer eröffneten Nominalphrase) das Merkmal USE auf „attributiver Gebrauch“ (attr) gesetzt. Da es keine Endungsinformation gibt, die auf die Komparativ- bzw. Superlativform verweist, erhält das Merkmal DEG den Wert „Positiv“ (pos).

Die für *Bich* jeweils unter AGR angegebenen Typen *adj-flexion*, *nom-flexion* und *vb-flexion* repräsentieren die Disjunktion aller unter diesem Merkmal möglichen Werte. Wie bei den Endungshypothesen werden auch bei den Wortklassenhypothesen die semantischen Merkmale – bis auf SORT – nur durch die jeweiligen Typen belegt. Die Adjektivhypothese erhält unter SORT die Werte „totale Qualität“ (tq) und „graduier-

---

[6] Neben den graduierbaren Qualitäten (z. B. *klein*, *teuer*) gibt es in der zugrundeliegenden Ontologie noch totale (z. B. *leer*, *grün*), assoziative (z. B. *chemisch*, *philosophisch*), ordnende (z. B. *nächster*) und relationale Qualitäten (z. B. *invers*, *äquivalent*). Zu den verwendeten semantischen Darstellungsmitteln vgl. HELBIG & SCHULZ 1997.

[7] Zur Belegung der Merkmalwerte unter STATE vgl. die Angaben zu *Kroxel* in Abbildung 2.

bare Qualität“ (gq) und die Nomenhypothese u. a. die Werte „Diskretum“ (d), „Substanz“ (s), „temporales Abstraktum“ (ta) und „ideelles Objekt“ (io). Für Verben wird unter SORT die allgemeine Sorte „Situation“ (si) eingetragen, die statische (Zustände) und dynamische Situationen (Handlungen, Geschehnisse) zusammenfaßt.

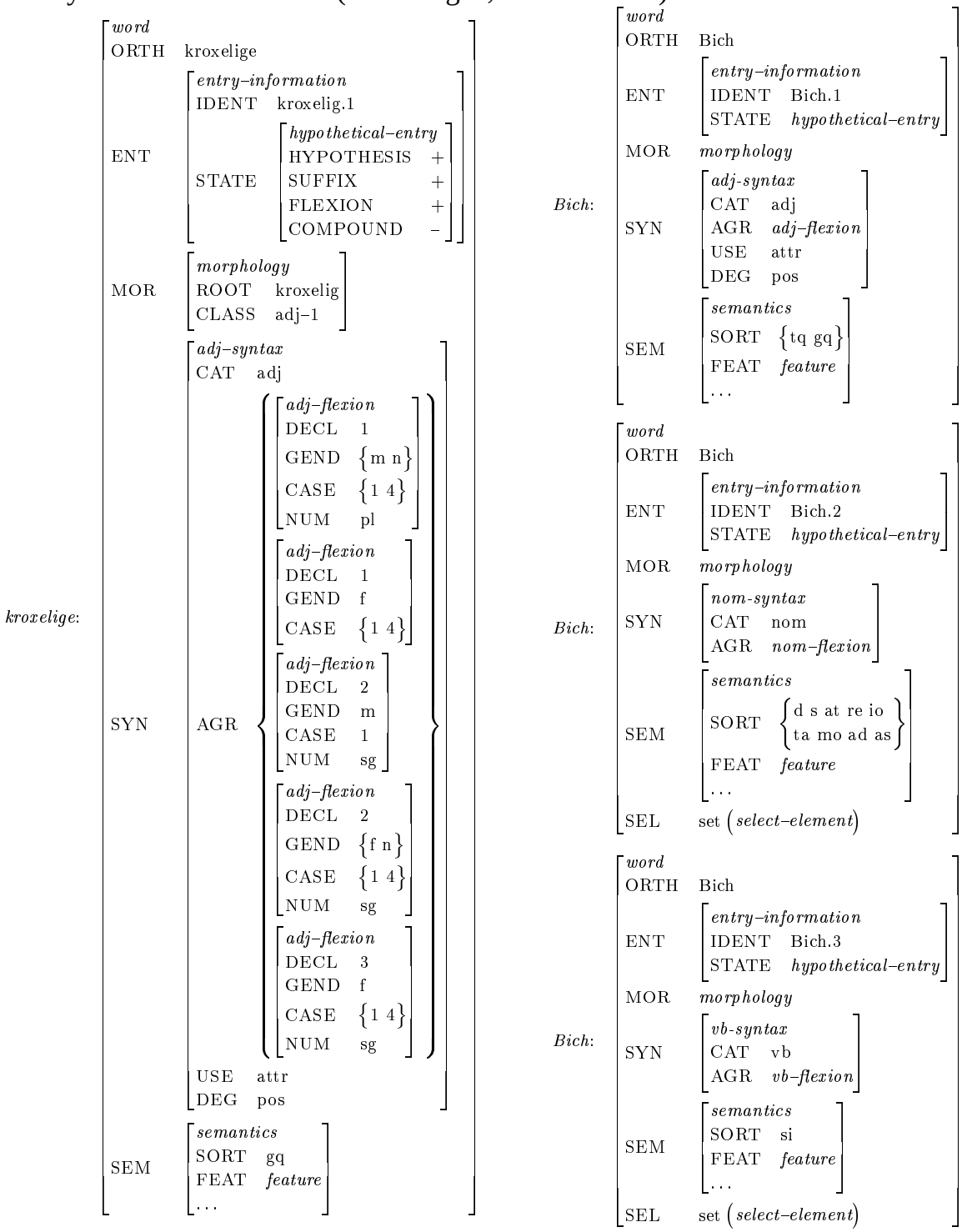


*Abb. 5: Merkmalstrukturen zu **kroxelige** und **Bich***

In diesem Beitrag kann aus Platzgründen nicht die gesamte Analyse des Beispielsatzes erläutert werden. Deshalb werden in Abbildung 6 nur ausschnittsweise Ergebnisstrukturen der Analyse vorgestellt. Im oberen Teil der Abbildung befinden sich die syntaktischen Merkmalstrukturen für die im Beispielsatz enthaltenen Nominalphrasen und für das Verb.

Die Spezialisierung der Merkmalwerte geschieht nicht unifikationsbasiert, sondern wird von den Wortagenten mit Hilfe von speziellen Funktionen zur Kongruenzüberprüfung durchgeführt. In der Mitte der Abbildung ist die Struktur des semantischen Netzes angegeben. Die darin enthaltenen Konzeptknoten `c.0`, `c.1` und `c.2` werden durch die im unteren Teil der Abbildung aufgeführten semantischen Merkmalstrukturen charakterisiert.[8]

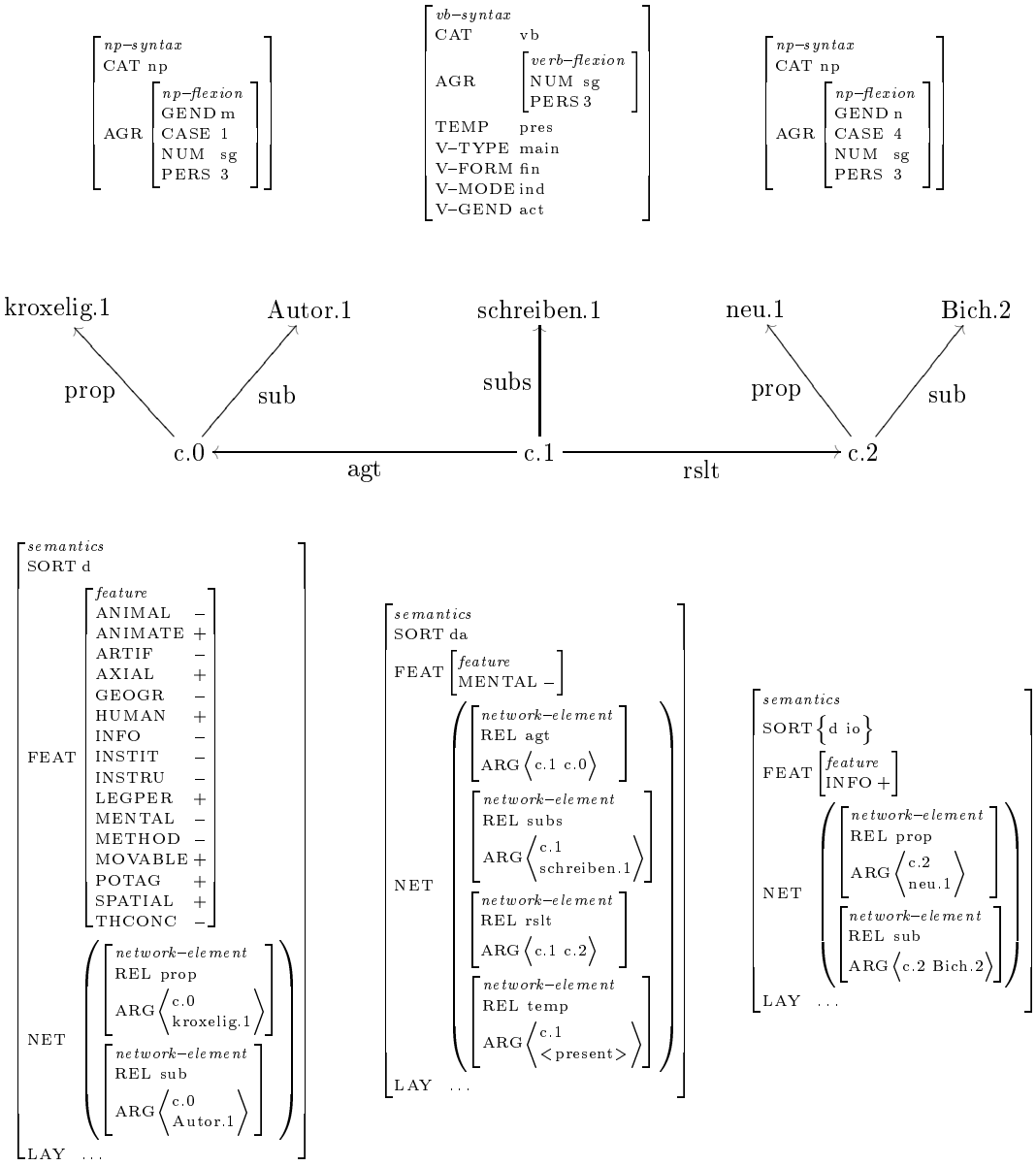


Abb. 6: Ergebnisstrukturen zu Der kroxelige Autor schreibt ein neues Bich.

Die Zuordnung der Kasusrelationen agt (Agent) und rslt (Resultat) zum Konzeptknoten `c.1` und die Spezialisierung der Sorten- und Featureinformation zum Konzeptknoten `c.2` wird mittels der Subkategorisierungsinformation im Lexikoneintrag zum Verb *schreiben* vorgenommen (vgl. Abbildung 1). Die Subkategorisierungsinformation ist ebenfalls für die Auflösung der Nominativ-Akkusativ-Mehrdeutigkeit von *ein neues Bich* verantwortlich (vgl. CASE-Wert für die zweite NP in Abbildung 6).

[8] Die Relationen SUB und SUBS stehen für die Subordination von Objekten bzw. Situationen. PROP steht für die Beziehung zwischen einem Objekt und dessen Eigenschaften.

5 Heuristische Bewertung der erzeugten Strukturen

Durch die Aktivierung von konkurrierenden Wortagenten kann sich die Zahl der erzeugten und potentiell möglichen Lesarten für eine Phrase sehr stark erhöhen. Für die Auswahl der wahrscheinlichsten Interpretation(en) ist ein heuristisches Verfahren entwickelt worden, welches die von einem Wortagenten erzeugten Interpretationen bewertet und anschließend daraus die optimale(n) Interpretation(en) auswählt. Das Ziel des Verfahrens ist es, aus den insgesamt erzeugten Strukturen nur die Lesarten herauszufiltern, die bereits komplettierte Phrasen enthalten.[9] Zusätzlich wird die Auswahl dahingehend gesteuert, daß die Lesarten möglichst lange Phrasen und somit insgesamt eine geringe Anzahl an Phrasen enthalten. Nachfolgend ist das Verfahren in einer vereinfachten Form angegeben:

- In der Menge M werden alle erzeugten Lesarten zur Eingabephrase gesammelt.
- Trage in die Menge M_1 alle Lesarten aus M ein, die nur komplettierte Phrasen enthalten.
- Trage in die Menge M_2 alle restlichen Lesarten aus M ein.[10]
- Wähle aus M_1 die Lesarten aus, die eine minimale Anzahl MIN_1 an komplettierten Phrasen enthalten und entferne alle anderen Lesarten aus M_1.
- Wähle aus M_2 die Lesarten aus, die eine minimale Anzahl MIN_2 an Phrasen enthalten und entferne alle anderen Lesarten aus M_2.
- Falls $MIN_1 \leq MIN_2$, wähle M_1, sonst M_2.

Durch dieses Verfahren ist auch die in Abbildung 6 angegebene Interpretation von *Bich* als Nomen ausgewählt worden. Diese Auswahl läßt sich verkürzt folgendermaßen begründen:

- Die Adjektivinterpretation von *Bich* erweitert zwar die eröffnete Nominalphrase *ein neues* zu *ein neues Bich*, aber komplettiert sie nicht.
  [[*Der kroxelige Autor*]$_{\text{np-comp}}$ [*schreibt*]$_{\text{v}}$]$_{\text{vp-comp}}$ [*ein neues Bich*]$_{\text{np-open}}$
- Die Nomeninterpretation von *Bich* erweitert die Nominalphrase zu *ein neues Bich*. Zusätzlich läßt sich diese komplettierte Nominalphrase als fakultatives Komplement an das Verb *schreiben* anbinden.
  [[*Der kroxelige Autor*]$_{\text{np-comp}}$ [*schreibt*]$_{\text{v}}$ [*ein neues Bich*]$_{\text{np-comp}}$]$_{\text{vp-comp}}$
- Die Verbinterpretation von *Bich* resultiert in der eröffneten Nominalphrase *ein neues* und zusätzlich in der eröffneten Verbalphrase *Bich*.
  [[*Der kroxelige Autor*]$_{\text{np-comp}}$ [*schreibt*]$_{\text{v}}$]$_{\text{vp-comp}}$ [*ein neues*]$_{\text{np-open}}$ [[*Bich*]$_{\text{v}}$]$_{\text{vp-open}}$

Als weitere Kriterien, die bei der heuristischen Bewertung zusätzlich eingesetzt, hier aber nicht näher erläutert werden, sind die folgenden zu nennen:

---

[9] Eine Nominal- bzw. Präpositionalphrase wird dann als „komplettiert“ betrachtet, wenn sie durch ein Nomen abgeschlossen ist. Bei Verbalphrasen müssen für die Komplettierung alle obligatorischen Komplemente vollständig interpretiert und an das Verb angebunden sein.

[10] Das eigentliche Verfahren unterscheidet hier noch weitere Mengen, z. B. die Menge der Lesarten, die nur eröffnete Phrasen enthalten.

- Groß-/Kleinschreibung bei Gleichbewertung;
- Bewertung der Kompatibilität von semantischen Merkmalen (z. B. bei der Anbindung von Komplementen an das Verb);
- Entfernen von Ergebnisstrukturen, die im Vorfeld des Verbs eröffnete Nominal- bzw. Präpositionalphrasen enthalten;
- Verschachtelungstiefe (z. B. bei erweiterten Partizipialkonstruktionen).

Jean-Yves Lalande, Universität zu Köln

VisualGBX: Ein objektorientiertes CAD-System zur Repräsentation und Evaluation linguistischer Theorien

1 Einleitung

In diesem Beitrag wird das vom Verfasser am Institut für Sprachliche Informationsverarbeitung der Universität zu Köln entwickelte System VisualGBX vorgestellt. Visual GBX in ein objektorientiertes CAD-System, das die Arbeit des Sprachwissenschaftlers durch die Einbindung von Visualisierungs-, Experten- und Datenbankfunktionen in einer graphischen Oberfläche modelliert und zum großen Teil automatisiert.[2] Das System ermöglicht die Implementation und die Simulation syntaktischer Analysen und wird am Institut für Sprachliche Informationsverarbeitung in erster Linie als Werkzeug zur Bereitstellung linguistischen Wissens, das für die maschinelle Übersetzung benötigt wird, benutzt. Die hierbei angewandte linguistische Theorie ist eine von Jürgen ROLSHOVEN (Uni Köln) entworfene objektorientierte Reinterpretation neuerer Entwicklungen der generativen Grammatik (von der GB-Theorie der ausgehenden 80iger Jahre bis heute, *Minimalist Program* miteinbezogen), die gerade in Bezug auf maschinelle Sprachverarbeitung neue interessante Perspektiven eröffnet.

Zunächst werden die Konzeption des Systems VisualGBX (2) und die Grundlagen des objektorientierten linguistischen Ansatzes erläutert (3). Anhand eines konkreten Implementationsbeispiels – der Analyse der Verbzweitstellung im Deutschen – wird dann unter Anwendung von VisualGBX gezeigt, wie Grundkonzepte des objektorientierten Ansatzes (Klassen, Vererbung, *message passing* u. a.) sich auf die syntaktische Analyse auswirken (4). Sichtbar dabei wird u. a., inwiefern der vorgestellte objektorientierte Ansatz konvergent zu neueren Entwicklungen in Bezug auf Minimalität und Lokalität im Rahmen des *Minimalist Program* (CHOMSKY 1992) ist (5) und unter Anwendung von VisualGBX als interaktivem Entwicklungstool zu neuen Lösungsansätzen führen kann.

---

[1] Die folgende Analyse baut auf WILDER (1995) auf. Vgl. hierzu 5. unten.

[2] Das System VisualGBX wird in LALANDE (1997) ausführlich beschrieben. Große Teile hiervon werden im vorliegenden Beitrag zusammengefaßt. Darüber hinaus wird eine neue objektorientierte minimalistische Implementation der Verbzweitstellung eingeführt, welche zu Generalisierungen bezüglich des Entwurfs objektorientierter syntaktischer Analysen führt.

2 Linguistische Forschungsarbeit mit VisualGBX

Die linguistische Hypothesenbildung innerhalb der GB-Theorie bedient sich bekanntlich syntaktischer Strukturbäume, um sprachliche Phänomene im Rahmen einer expliziten Theorie der menschlichen Sprachfähigkeit zu beschreiben und zu erklären. Diese Baumstrukturen unterliegen strengen Wohlgeformtheitsbedingungen, die universelle, z. T. aber sprachspezifisch parametrisierte abstrakte grammatische Prinzipien widerspiegeln und von der Theorie aufgedeckt und formalisiert werden. Da es um die Erforschung der universellen Eigenschaften von natürlichen Sprachen und der den Unterschieden zwischen diesen Sprachen entsprechenden Parametereinstellungen geht, ist es dabei unumgänglich, jede von der Theorie aufgestellte Hypothese zwecks Evaluierung und ggf. Falsifizierung mit der größtmöglichen Zahl sprachlicher Daten zu konfrontieren. Zur Formulierung und Überprüfung von Hypothesen werden daher im Rahmen der generativen Grammatik für zahlreiche Sätze aus verschiedenen Sprachen zahlreiche Baumstrukturen unter stetiger Berücksichtigung der von der Theorie herausgearbeiteten Prinzipien erstellt und auf ihre Wohlgeformtheit hin überprüft. Die Verfahrensweise ist hierbei folgende: Ausgehend von den zu untersuchenden sprachlichen Daten werden Hypothesen in Form von linguistischen Baumstrukturen und von Prinzipien, von denen die Wohlgeformtheit dieser Strukturen abhängt, aufgestellt. Durch den Abgleich von Strukturen und Prinzipien werden die Hypothesen evaluiert. Verifizierte Hypothesen, in Form von Strukturen und/oder Prinzipien, müssen festgehalten und bei der Überprüfung weiterer als Erweiterung des linguistischen Wissens mitberücksichtigt werden. Falsifizierte Hypothesen dagegen müssen als solche protokolliert werden; sie können ggf. zum Zwecke weiterer Tests modifiziert werden.

VisualGBX unterstützt bzw. übernimmt teilweise vollständig bestimmte Aufgaben dieses sprachwissenschaftlichen Erkenntnisprozesses bei der Aufstellung und der Evaluierung linguistischer Hypothesen. Dies wird in Abbildung 1 (aus LALANDE 1997:80) durch die Schattierungen hervorgehoben.

Als interaktives graphikbasiertes Werkzeug zur Konstruktion linguistischer Theorien, unterstützt somit VisualGBX mit Hilfe von Techniken der Expertensystemtechnologie und des Computer Aided Design (CAD) jeden Entwicklungsschritt bei der Modellierung linguistischer Hypothesen:

- Die *Strukturen* werden vom Linguisten editiert und vom System graphisch dargestellt.
- Das *linguistische Wissen* speist sich aus drei Quellen: *Klassendefinitionen* (s. u.), Moduln, die *Methodenspezifikationen* enthalten, welche in LPS-Prolog formuliert sind, einer von ROLSHOVEN entwickelten deklarativen objektorientierten linguistischen Programmiersprache, deren prädikatenlogische Notation eng an die der GB-Theorie angelehnt ist (vgl. ROLSHOVEN 1991, LALANDE 1997:Kap.2.4 sowie 4.3 unten), und einem *Lexikon*.

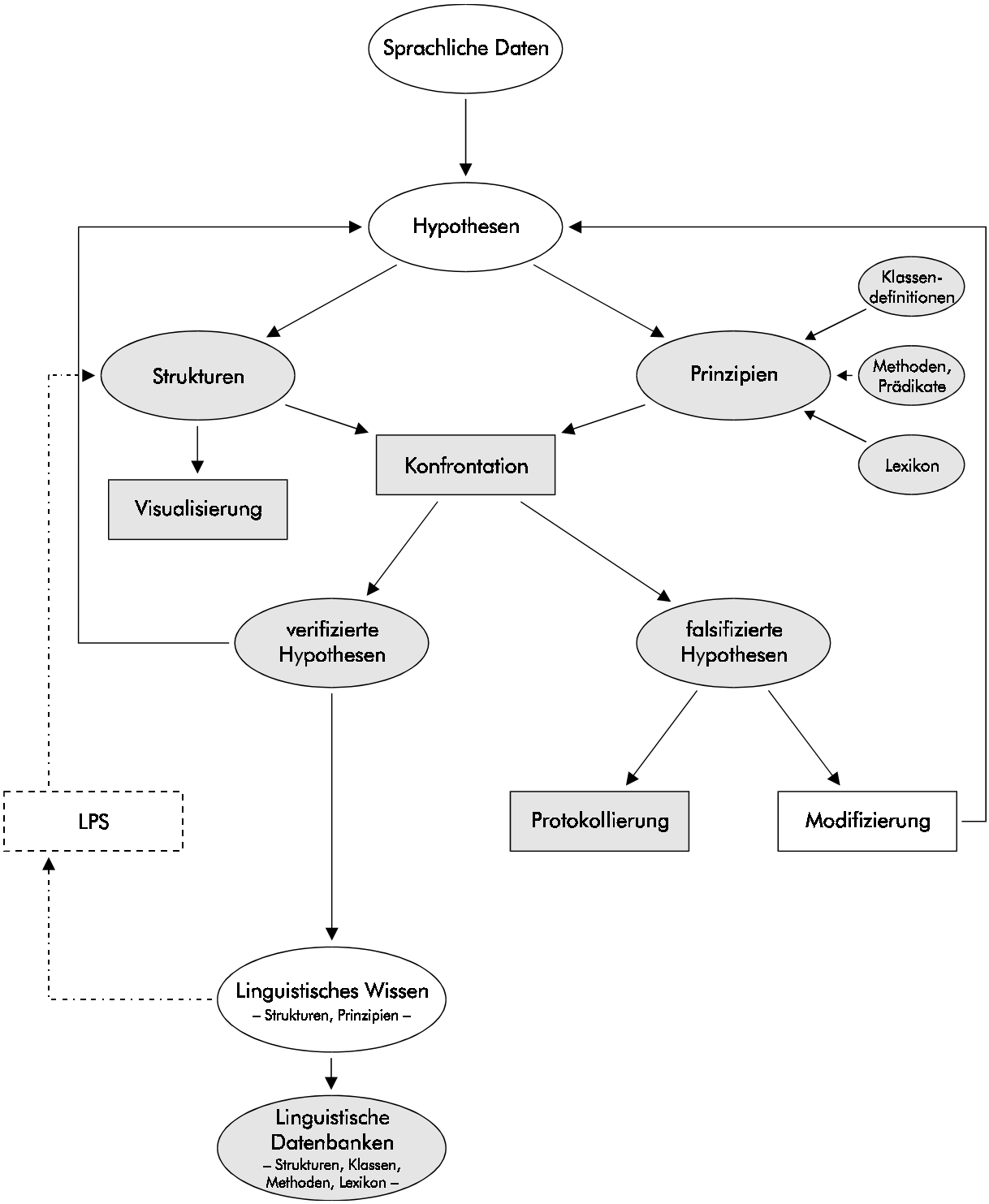


Abb. 1: Systemarchitektur von VisualGBX

- Die *interaktive Evaluation* von Strukturen und linguistischem Wissen gibt Rückschlüsse auf die Wohlgeformtheit der Strukturen und/oder die Gültigkeit des angewandten linguistischen Wissens.
- Strukturen können zur Bildung alternativer Hypothesen *konvertiert* werden, um erneut mit dem linguistischen Wissen konfrontiert zu werden.
- Linguistisches Wissen, das zur falschen Evaluierung bereits belegter Strukturen führt, kann seinerseits *verändert* werden – entweder in den Klassen- und/oder Methodendefinitionen, oder im Lexikon –, bevor es erneut angewandt wird.

- Lizensierte Daten – seien es Strukturen oder linguistische Prinzipien – werden in *linguistischen Datenbanken* gespeichert und können im Laufe der Forschungsarbeit stets wieder abgerufen werden.

Darüber hinaus kann evaluiertes Wissen zum Parsen und Generieren natürlicher Sprachen mit dem von ROLSHOVEN entwickelten System zur maschinellen Übersetzung LPS genutzt werden. Umgekehrt können Strukturen, die von LPS erzeugt werden, nach VisualGBX exportiert werden und zum Zwecke der Überprüfung ihrer Wohlgeformtheit und der Weiterentwicklung des für die automatische Analyse und Synthese natürlicher Sprache gebrauchten linguistischen Wissens mit alternativen Daten konfrontiert werden.

3 Ein objektorientierter linguistischer Ansatz

Wie eingangs erwähnt, beruht die syntaktische Analyse unter Anwendung von VisualGBX auf einem objektorientierten linguistischen Ansatz. Dieser wird, nach der Klärung einiger Grundbegriffe, im folgenden vorgestellt.

3.1 Grundbegriffe

In einem objektorientierten Ansatz wird eine *Objektwelt* in *Klassen* strukturiert, in denen das gleiche Verhalten einer Menge von Objekten kodiert ist. Objekte, die einer bestimmten Klasse angehören, werden *Instanzen* dieser Klasse genannt. Die Strukturierung in Klassen erfolgt mittels eines *Vererbungsmechanismus*: Einerseits erben die Instanzen einer Klasse die Eigenschaften dieser Klasse, andererseits können Eigenschaften von Klasse zu Klasse vererbt werden. In letzterem Fall entstehen Generalisierungshierarchien, in denen eine *Tochterklasse* von ihrer *Mutterklasse* erbt.

In dem hier beschriebenen linguistischen Ansatz sind die syntaktischen Kategorien die Objekte, die aufgrund ihrer Eigenschaften und ihres Verhaltens bezüglich der Strukturen, in die sie involviert sind, in Klassen unterteilt werden. Jeder syntaktischen Kategorie entspricht eine *konkrete Klasse*, auch *Knotenklasse* genannt. Darüber hinaus verwendet der objektorientierte Ansatz *abstrakte Klassen*, in denen gemeinsame Eigenschaften verschiedener Knotenklassen formalisiert werden. Diese gemeinsamen Eigenschaften werden von den abstrakten Klassen an ihre Tochterklassen vererbt (*statische Vererbung*, s. u.). Dadurch wird eine Klassifikation ermöglicht, in der hierarchische Beziehungen zwischen Klassen, und damit auch zwischen ihren Objekten, d. h. den syntaktischen Kategorien, berücksichtigt werden, was zur Vereinfachung und Strukturierung des für die Beschreibung und Erklärung sprachlicher Phänomene benötigten linguistischen Wissens beiträgt. Dies wird im folgenden an einem einfachen Beispiel konkretisiert.

3.2 Statische, instantiierende und dynamische Vererbung[3]

In dem hier vorgestellten objektorientierten Ansatz unterscheiden wir zwischen drei Arten der Vererbung: die *statische Vererbung* bezeichnet das Weiterreichen von Eigenschaften von einer Klasse zu einer anderen Klasse, die *instantiierende Vererbung* das Weiterreichen von Eigenschaften von einer Klasse zu ihren Instanzen, die *dynamische Vererbung* schließlich das Weiterreichen von Eigenschaften von einem Knoten (als Instanz einer Klasse) zu einem weiteren Knoten. Dies wird im folgenden an spezifischen Beispielen näher erläutert.

Die konkrete Realisierung eines Verbs wie *kaufen* in einem deutschen Satz ist in ihrer Eigenschaft als sprachliche Einheit der syntaktischen Kategorie V(erb) eine Instanz oder ein Objekt der Klasse der Verben. Die Klasse der Verben beinhaltet, wie alle Klassen, Eigenschaften, die allen Instanzen dieser Klasse gemeinsam sind. Das Verb *kaufen* teilt ja mit den anderen Verben des Deutschen bestimmte Eigenschaften, die z. T. nur für diese Klasse von syntaktischen Kategorien gelten. Zur Vereinfachung seien hier nur drei von diesen Eigenschaften genannt: die Fähigkeit zu regieren, die Rektionsrichtung und, da es sich hier um ein deutsches Verb handelt, die Eigenschaft, im Hauptsatz an zweiter Position stehen zu müssen, oder Verbzweiteigenschaft. Diese Eigenschaften sind zwar in ihrer Kombination für das (deutsche) Verb spezifisch, sie gelten aber auch z. T. für andere Klassen von syntaktischen Kategorien (nicht nur Verben regieren, alle Regentes haben eine Rektionsrichtung, Verbzweitstellung ist kein auf das Deutsche beschränktes Phänomen). Derartige Eigenschaften werden in einem objektorientierten Ansatz als Eigenschaften von abstrakten Klassen definiert, die über Vererbung an weitere Klassen weitergereicht werden. Wir brauchen für das hier angeführte (vereinfachte) Beispiel drei abstrakte Klassen, aus der die Klasse V der deutschen Verben erbt: die Klasse *Regens* der Regentes, die Klasse *Links* für die Rektionsrichtung nach links, und die Klasse *VerbZweit*. Die Eigenschaften dieser drei abstrakten Klassen werden über statische Vererbung[4] an ihre Tochterklasse V, die Klasse der (deutschen) Verben, weitergereicht. Die Instanz einer Klasse, wie das Verb *kaufen* hier, erbt ihrerseits alle Eigenschaften ihrer Klasse, was hier instantiierende Vererbung genannt wird. Dies wird in Abbildung 2 dargestellt.

Wie sehen die vererbten Eigenschaften aus? Im Bereich der Objektorientierung unterscheidet man bei der Spezifikation einer Klasse zwischen *deklarativen Attributen* (*Variablen*) und *prozeduralen Attributen* (*Methoden*). Variablen sind lokale Daten, die den inneren Zustand eines Objekts beschreiben. Methoden sind Operationen, die ein oder mehrere Objekt(e) involvieren und somit eine Schnittstelle zwischen Klassen bilden können.

---

[3] Die Begriffe *statische*, *instantiierende* und *dynamische Vererbung* stammen aus ROLSHOVEN (1997). Vgl. hierzu auch LALANDE 1997:43ff.

[4] Statisch, im Vergleich zu dem unter 4.3 beschriebenen dynamischen Prozeß des Weiterreichens von Methoden von Knoten zu Knoten beim Aufbau eines Strukturbaums.

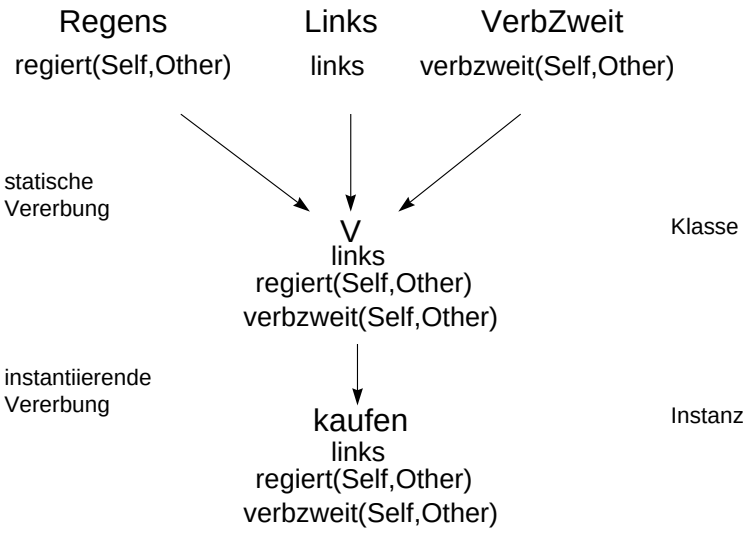


Abb. 2: Vererbungshierarchie von kaufen

Ein deklaratives Attribut ist z. B. das Merkmal *links*, das von der abstrakten Klasse *Links* an die Klasse der (deutschen) Verben vererbt wird. Nach der instantiierenden Vererbung enthält also das Verb *kaufen* in seiner Spezifikation das Merkmal *links*.

Die beiden anderen vererbten Eigenschaften sind dagegen prozedurale Attribute, Methoden. In dem hier vorgestellten objektorientierten linguistischen Ansatz sind Methoden linguistische Prinzipien, die als prädikatenlogische Klauseln formuliert sind und von deren Beweis die Wohlgeformtheit einer Struktur, in der sich ein Objekt der Klasse befindet, die diese Methode enthält, abhängt. Eine Methode kann *ein-*, *zwei-* oder *dreistellig* sein; sie enthält immer ein Argument (im prädikatenlogischen Sinne), das mit dem Objekt, das diese Methode enthält, instantiiert ist. Dieses Argument wird, der im Bereich der Objektorientierung gängigen Terminologie entsprechend (vgl. hierzu z. B. REISER & WIRTH 1992:238), durch die Variable `Self` symbolisiert. Mehrstellige Methoden enthalten ein nicht instantiiertes Argument, das mit der Variable `Other` symbolisiert wird.

In unserem Beispiel enthält das Verb *kaufen* also eine Methode `regiert` und eine Methode `verbzweit`, deren erstes Argument jeweils mit ihm selbst instantiiert ist, während das zweite Argument zunächst noch nicht instantiiert, also variabel ist. Wie es im Zuge der *dynamischen Vererbung* von Methoden zur Instantiierung des zweiten Arguments und somit zur Aktivierung der Methode kommt, läßt sich am besten unter Anwendung von VisualGBX an einem konkreten Analysebeispiel demonstrieren.

4 VisualGBX am Beispiel einer objektorientierten Analyse der Verbzweitstellung im Deutschen[5]

4.1 Klassendefinitionen

Die Eigenschaften von Klassen und die Vererbungsverhältnisse zwischen Klassen werden in Klassendefinitionen festgelegt. Für das hier gewählte Beispiel werden folgende abstrakte Klassendefinitionen benötigt (für eine genaue Beschreibung des Formalismus vgl. LALANDE 1997:44ff.):

(1) *Abstrakte Klassen*

```
CLASS Rechts | [rechts];
END;
CLASS Links | [links];
END;
CLASS Regens;
  regiert(Self,Other) [barriere],f;
END;
CLASS Regiert;
  regiert(Other,Self),f;
END;
CLASS VerbZweit;
  verbzweit(Self,Other);
END;
CLASS LeeresComp;
  top(Other,Self) >2-bar<;
  verbzweit(Other,Self);
END;
CLASS Topik;
  top(Self,Other) [wurzel], f;
END.
```

Wie aus den Definitionen der folgenden Knotenklassen ersichtlich, werden Vererbungsverhältnisse zwischen Klassen unmittelbar hinter dem Klassennamen angegeben, wobei das Symbol "<" benutzt wird und als "erbt aus" zu interpretieren ist:

(2) *Knotenklassen*

```
(* Deutsch *)
CLASS C°[-lex] < LeeresComp;
END;
CLASS V°[+finit, basis] < Regens, Links;
END;
CLASS V°[+finit, e] < VerbZweit;
END;
CLASS D" < Regiert, Topik;
END;
...
```

[5] Die folgende Analyse baut auf WILDER (1995) auf. Vgl. hierzu 5. unten.

Unter (1) und (2) sind lediglich die Klassendefinitionen abgedruckt, die für unser Analysebeispiel relevant sind. Die Klasse der nicht lexikalisch gefüllten C°-Knoten ist eine Tochterklasse der abstrakten Klasse `LeeresComp`, während die Klasse der finiten verbalen Köpfe in Basisposition Tochterklasse der abstrakten Klassen `Regens` und `Links` ist. Die entsprechende Klassendefinition der verbalen Köpfe im Französischen würde wie folgt lauten, da Verben im Französischen bekanntlich nicht nach links, sondern nach rechts regieren (SVO-Stellung im Französischen vs. SOV-Stellung im Deutschen):

(3)
```
(* Französisch *)
CLASS V° < Regens, Rechts;
END;
```

Die Klasse der in einer leeren Position gelandeten (durch `e`, für empty, gekennzeichneten) finiten Verben erbt ihrerseits aus der abstrakten Klasse `VerbZweit`. Schließlich ist die Klasse der Determiniererphrasen `D"` Tochterklasse der abstrakten Klasse `Regiert` der regierbaren Kategorien und gleichzeitig, wie alle maximalen Projektionen (vgl. hierzu unten), Tochterklasse der abstrakten Klasse `Topik`, was bedeutet, daß Instanzen dieser Klasse potentielle Topikelemente sind.

An diesem Beispiel zeichnet sich schon ab, inwiefern der objektorientierte Ansatz die Klassifikation und die Typologisierung von Sprachen mit Hilfe abstrakter Klassen unterstützt: Abstrakte Klassen stellen universelles linguistisches Wissen in Form von Attributen und Methoden bereit, welches unter Berücksichtigung sprachspezifischer und ggf. analysespezifischer Unterschiede an Nachkommenklassen vererbt wird. Somit spiegeln sich Parametereinstellungen, wie z. B. die Wahl der Rektionsrichtung, im objektorientierten Ansatz größtenteils in den Klassendefinitionen wider. Darauf werden wir unten zurückkommen. Zunächst wird im folgenden gezeigt, wie das hier formulierte Wissen unter Anwendung von VisualGBX zur Analyse sprachlicher Strukturen eingesetzt wird.

4.2 Visualisierung sprachlicher Strukturen unter VisualGBX

Die oben beschriebenen Klassendefinitionen sowie die Methodendefinitionen, auf die unter 4.3 näher eingegangen wird, werden bei der Initialisierung von VisualGBX eingelesen und bilden das linguistische Wissen des Systems.

Nun geht es darum, dieses linguistische Wissen zur Analyse sprachlicher Strukturen anzuwenden. Sprachliche Strukturen werden in VisualGBX auf der Grundlage des eingelesenen linguistischen Wissens in einem Format, das dem von Produktionsregeln ähnelt, beschrieben und vom System automatisch graphisch dargestellt (vgl. hierzu LALANDE 1997:12ff.).[6] Abbildung 3 zeigt die Visualisierung einer möglichen Struktur des Satzes

(4) ..., weil Hans das Buch kauft.

[6] VisualGBX ist in *Oberon-2* unter *Oberon/F*™ realisiert und läuft sowohl unter *Windows NT/95/3.11*™ als auch unter *Macintosh System 7.5*™. *Windows 3.1/NT/95* sind eingetragene Warenzeichen der Firma *Microsoft Corporation*, *Macintosh System 7.5* ein eingetragenes Warenzeichen der Firma *Apple Computer*, *Oberon/F* ein eingetragenes Warenzeichen der Firma *Oberon microsystems*.

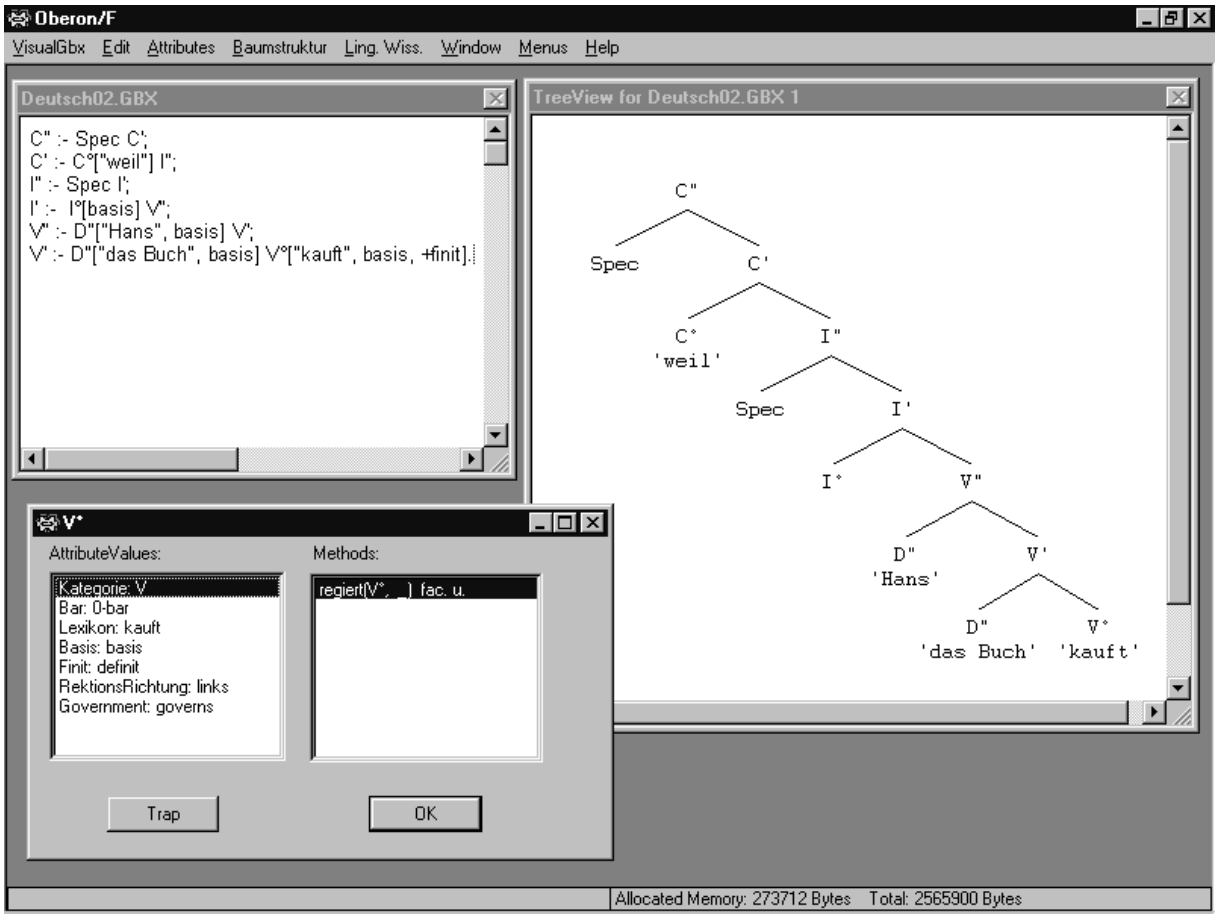


Abb. 3: Strukturviualisierung in VisualBGX

Bei ihrer Interpretation durch VisualGBX werden die Beschreibungsregeln (in Abbildung 3 im linken oberen Unterfenster 'Deutsch02.GBX' zu sehen) unter Rückgriff auf die Klassendefinitionen und das Lexikon des Systems in dynamische Datenstrukturen umgewandelt, die alle für die Analyse des Satzes benötigten linguistischen Informationen beinhalten. Jeder Knoten der Struktur wird aufgrund der Angaben in den Regeln als Objekt einer vordefinierten Klasse – welche natürlich aus Mutterklassen erben kann, s. o. – in einen Graph eingefügt, der vom System graphisch dargestellt wird.

Die Knoten des visualisierten Baums enthalten als Instanzen von bestimmten Klassen das linguistische Wissen, das zur Lizensierung bzw. Nichtlizensierung der Struktur benötigt wird. Dies sieht man, wenn man einen Knoten der Struktur anklickt, um dessen Eigenschaften sichtbar zu machen. In Figur 3 z. B. wurde der verbale Kopf in Basisposition angeklickt, was zur Ausgabe seiner Eigenschaften im linken unteren Fenster führte. Wie man sieht, enthält die Basisposition des Verbs, ihrer Klassendefinition entsprechend, u. a. das Merkmal `links` und eine Methode `regiert(Self, Other)`. Wie oben besprochen, ist das erste Argument der Methode mit dem Objekt, das sie enthält, instantiiert, während das zweite Argument auf der Ebene des Verbs nicht instantiiert ist. Gleiches gilt (in umgekehrter Reihenfolge) für die Methode `regiert(Other,Self)` der Komplement-DP *das Buch*. Wie werden nun diese Argumente instantiiert? Dies geschieht im Zuge der dynamischen Vererbung von mehrstelligen Methoden durch die Baumstruktur.

4.3 Dynamische Vererbung

Eine mehrstellige Methode wird innerhalb einer sprachlichen Struktur von ihrem Objekt ausgehend solange von Mutterknoten zu Mutterknoten nach oben gereicht, bis sie entweder mit derselben Methode mit komplementär instantiierten beiden ersten Argumenten zusammentrifft oder den Knoten erreicht hat, der den Skopus ihrer dynamischen Vererbung begrenzt (s. hierzu 4.4). Dies wird in Abbildung 4 graphisch dargestellt:

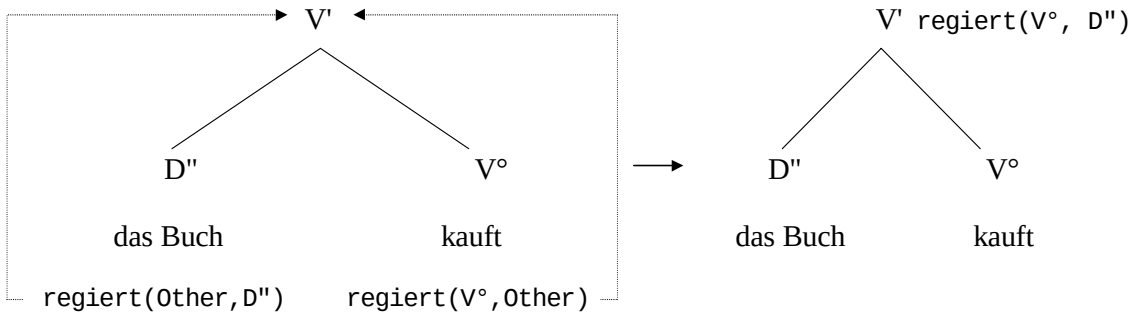


Abb. 4: Begrenzung der dynamischen Vererbung am Beispiel kaufen

Die Methoden `regiert(Self,Other)` des Verbs *kauft* aus obigem Beispiel und die Methode `regiert(Other,Self)` seines Komplements werden an die Mutter dieser Knoten, V', weitergereicht. Da die Argumente beider Methoden komplementär instantiiert sind, wird der Beweis der Methode `regiert` in V' ausgelöst. Methoden sind in dem hier beschriebenen objektorientierten Ansatz prädikatenlogische Klauseln, die in der von ROLSHOVEN entwickelten linguistischen Programmiersprache LPS-Prolog formuliert sind. In einem eigenen Modul ist `regiert` wie folgt definiert:

(5)
```
regiert(Alpha, Beta) :-
HatEigenschaft(Alpha, 'links'),
rechtsVon(Alpha, Beta);
HatEigenschaft(Alpha, 'rechts'),
linksVon(Alpha, Beta).
```

In der hier untersuchten Struktur wird die Methode bewiesen, da das Verb aufgrund seiner Klassendefinition das Merkmal `links` enthält und sein Komplement sich links von ihm befindet, d. h. in der kanonischen Richtung regiert wird. Dadurch ist die in Figur 4 dargestellte Teilstruktur hinsichtlich des Rektionsverhältnisses zwischen Verb und direktem Objekt lizensiert. Das Scheitern einer Methode – sei es weil ihr Beweis zu einem negativen Ergebnis führt oder weil er nicht ausgelöst werden kann, weil die Methode im Zuge ihrer dynamischen Vererbung in einem Grenzknoten hängengeblieben ist (s. u.) – führt hingegen zu einer Fehlermeldung des Systems und somit einem Hinweis auf eine Nichtlizensierung der untersuchten Struktur. Dies wird im folgenden am Beispiel der Verbzweitstellung gezeigt.

4.4 Analyse der Verbzweitstellung im Deutschen: Objektorientiert und minimalistisch

Wir können uns jetzt der angekündigten objektorientierten Analyse der Verbzweitstellung im Deutschen genauer zuwenden. In der in Abbildung 5 graphisch dargestellten Struktur des Satzes *Das Buch kauft Hans* nimmt das finite Verb, wie in jedem deutschen Hauptsatz, die zweite Position ein.

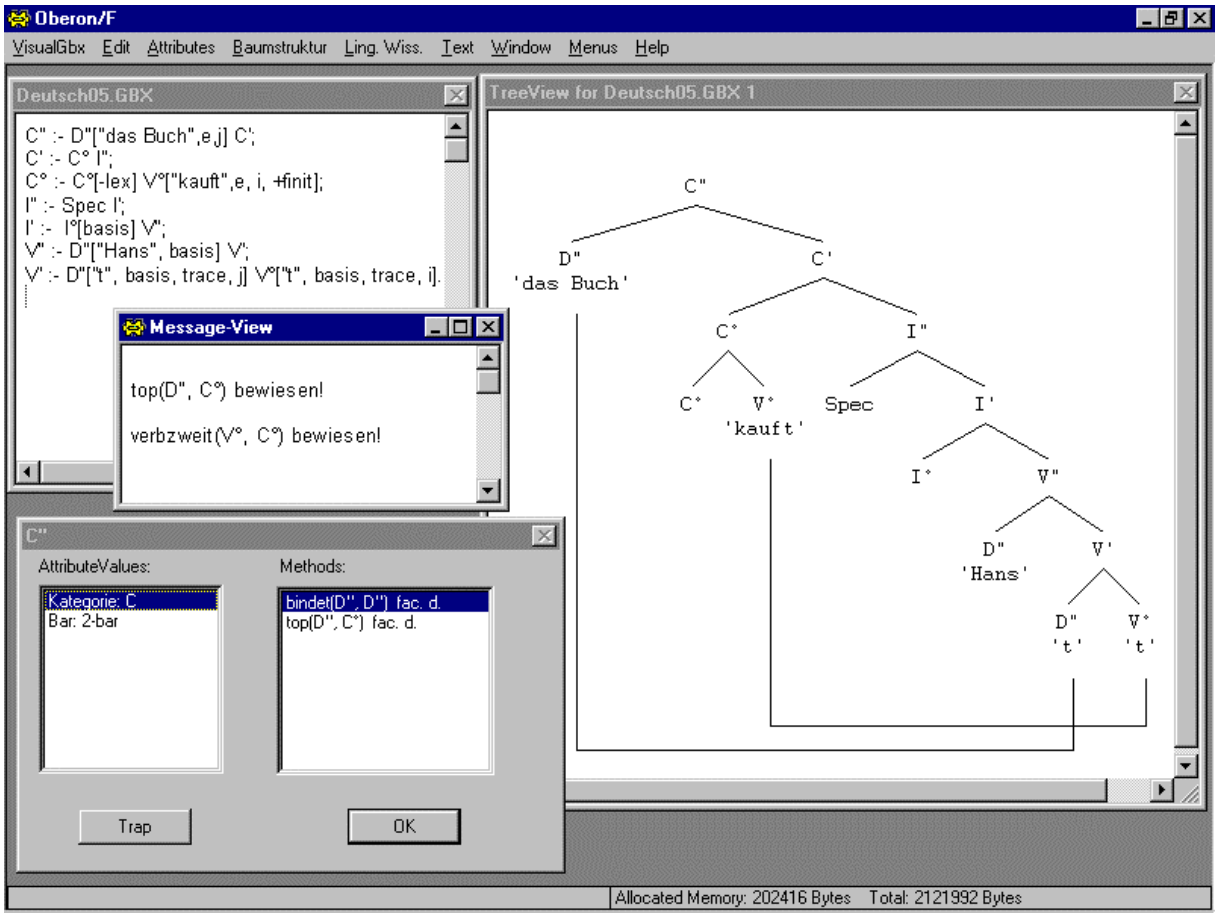


Abb. 5: Visualisierung der Struktur von Das Buch kauft Hans

In dieser Struktur ist die Complementizer-Position nicht durch eine nebensatzeinleitende Konjunktion gefüllt, was in der Implementation der entsprechenden Klassen vereinfachend durch das Merkmal `-lex` symbolisiert wurde (s. (2) oben), so daß der Knoten C° zwei Methoden aus der abstrakten Klasse `LeeresComp` erbt: `top(Other, Self)` und `verbzweit(Other,Self)`. Beide Methoden sind obligatorisch, d. h. sie müssen bewiesen werden, damit die Struktur lizensiert ist.[7]

Damit sie bewiesen werden, muß es in der Struktur Knoten geben, die die entsprechenden Methoden mit komplementär instantiierten Argumenten enthalten und Positionen besetzen, die den Beweis der beiden Methoden im Zuge ihrer dynamischen Vererbung ermöglichen. Das Pendant zu `top(Other,Self)` wird in Form von `top(Self,Other)` von allen maximalen Projektionen aus der abstrakten Klasse `To-`

[7] Methoden sind defaultmäßig obligatorisch. Die Fakultativität einer Methode wird in den Klassendefinitionen durch '`f`' hinter der betroffenen Methode angegeben. Vgl. z. B. die Klassendefinition von `Topik` unter (1).

`pik` geerbt. M. a. W. sind alle maximalen Projektionen Topikalisierungskandidaten. Die Methode `top(Other,Self)` von C° kann erst in dem in der Klassendefinition von `LeeresComp` in (1) hinter dem Methodennamen angegebenen Zielknoten `>2-bar<` bewiesen werden, d. h. erst auf der C"-Ebene.[8] Die Methode `top(Other, Self)` kann hingegen nur bis zur Mutter ihres Objektes dynamisch vererbt werden, da in der Klassendefinition von `Topik` kein Grenzknoten hinter der Methode angegeben wurde (Defaultfall). Daraus folgt, daß der Beweis von `top` nur dann gewährleistet ist, wenn sich eine maximale Projektion in SpecCP befindet. M. a. W. ist `top` eine Methode, die nur in einem Spec-Head-Verhältnis bewiesen werden kann (vgl. hierzu 5. unten), da nur in dieser Konfiguration beide Instanzen der Methode mir komplementär instantiierten Argumenten zusammentreffen können.

Die Methoden `verbzweit(Self,Other)` und `verbzweit(Other,Self)` dagegen können beide nur bis zur Mutter ihres Objekts dynamisch vererbt werden (s. (1)). `verbzweit` kann also nur in einer Schwesterbeziehung bewiesen werden, m. a. W. wenn das Verb an C° adjungiert wurde. Entsprechend der Rückmeldung des Systems im 'Message-View' in Figur 5, in dem das Ergebnis der Methodenbeweise ausgegeben wird, ist dies in der dort dargestellten Struktur der Fall.

In einer Struktur, in der hingegen entweder das Verb nicht in C° bewegt wurde oder keine maximale Projektion topikalisiert wurde, landet eine der obligatorischen Methoden von C° in ihrem Grenz- bzw. Zielknoten, ohne bewiesen zu werden. Ein solches Scheitern einer obligatorischen Methoden weist auf die Ungrammatikalität der Struktur hin, welche durch eine entsprechende Meldung des Systems signalisiert wird (Abbildung 6).

Diese Beispiele zeigen, wie VisualGBX zur Filterung zulässiger und unzulässiger Strukturen, bzw. zulässiger und unzulässiger Prinzipien, die sich in Klassen- und Methodendefinitionen widerspiegeln, eingesetzt werden kann. Der Mechanismus der dynamischen Vererbung stellt hierbei einen Fall von *message passing* dar, bei dem die Methoden die Nachrichten sind, die zwischen Objekten ausgetauscht werden – allerdings in einer für *message passing* eingeschränkten Art und Weise, da nur bottom-up und nur bei Unifikation beider Argumente –. Wichtig hierbei ist die Möglichkeit der Einschränkung der Reichweite der dynamischen Vererbung über die Angabe von

---

[8] Die Reichweite der dynamischen Vererbung einer Methode ist defaultmäßig auf die Mutter des Objekts, das diese Methode enthält, beschränkt. Sie kann erweitert werden über die Angabe eines Grenzknotens oder eines Zielknotens in der Klassendefinition, aus der die betroffene Methode vererbt wird. Ein Grenzknoten definiert hierbei den Bereich, innerhalb dessen eine Methode bewiesen werden kann. Beispiel hierfür ist der Grenzknoten `[barriere]` der Methode `regiert(Self, Other)`, die nicht über eine Barriere hinaus dynamisch vererbt werden darf (vgl. hierzu LALANDE 1997:58ff.) und folglich unterhalb bzw. spätestens auf der Höhe ihres Grenzknotens bewiesen werden muß. Ein Zielknoten unterscheidet sich von einem Grenzknoten dadurch, daß er angibt, auf welcher Ebene eine Methode unifiziert werden muß: die Methode `top(Self,Other)`, die aus `LeeresComp` vererbt wird, kann erst auf der Ebene der ersten maximalen Projektion, von der ihr Objekt dominiert wird, bewiesen werden, darunter nicht. M. a. W. ist ein Zielknoten ein Grenzknoten, unterhalb dessen nicht unifiziert werden darf.

Grenz- bzw. Zielknoten für die zu verschickenden Messages. Gerade im Hinblick auf das Bottom-up-Generieren von sprachlichen Strukturen bei der Analyse und der Synthese natürlicher Sprache ist dies von besonderer Bedeutung, weil dadurch Lokalität und Minimalität bewahrt werden, was zu einer Minimierung des Aufwands und somit zu einer Effizienzsteigerung der syntaktischen Analyse führt: Entscheidungen über die Zulässigkeit des Linkens von zwei Kategorien bzw. zwei Teilbäumen können immer streng lokal gehalten und über das Ergebnis des *message passing* getroffen werden.

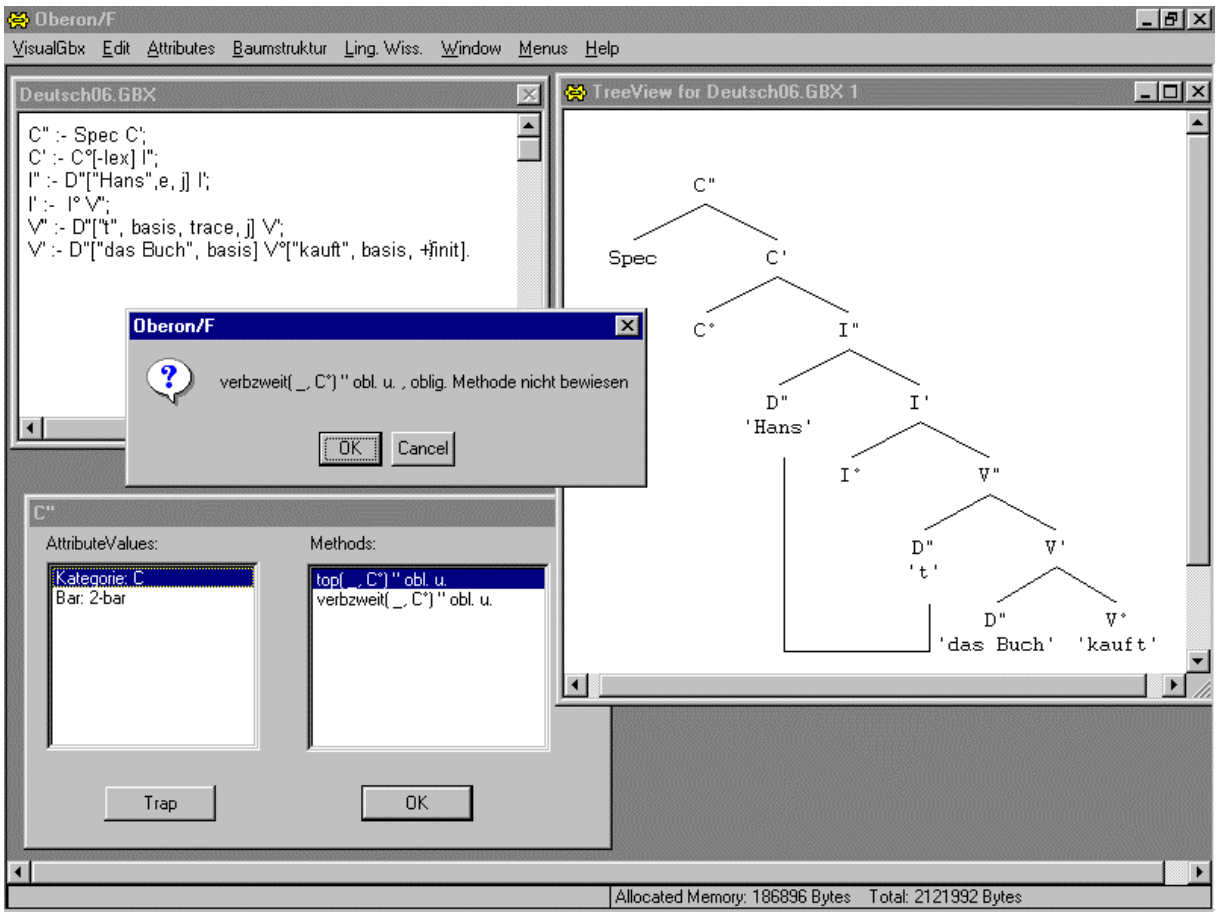


Abb. 6: Verarbeitung ungrammatischer Strukturen in VisualGBX

5 Ausblick: Objektorientierter Ansatz und *Minimalist Program*

Abschließend sollte noch kurz auf die Parallele zwischen dem vorgestellten objektorientierten Ansatz zur Sprachverarbeitung und den neueren Entwicklungen im Rahmen des *Minimalist Program* eingegangen werden sowie auf einige Generalisierungsmöglichkeiten beim Entwurf objektorientierter Implementationen.

Dem aufmerksamen Leser wird die Parallele zwischen beiden Ansätzen sicherlich nicht entgangen sein. Die eben vorgeschlagene Analyse der Verbzweitstellung im Deutschen ist nichts anderes als die direkte objektorientierte Umsetzung der Analyse von Verbzweitphänomenen nach WILDER 1995 im Rahmen des *Minimalist Program*. An dieser Stelle im Detail auf das *Minimalist Program* einzugehen, würde den Rahmen dieses Beitrags selbstverständlich bei weitem sprengen. Es sei lediglich auf einige Gemeinsamkeiten beider Ansätze kurz hingewiesen:

Im *Minimalist Program* (vgl. CHOMSKY 1992, 1995) wird davon ausgegangen, daß die Flexionsaffixe lexikalischer Kategorien mit Flexionsmerkmalen assoziiert sind. Funktionalen Kategorien enthalten ihrerseits (nicht morphologisch realisierte) Flexionsmerkmale, die den Merkmalen der lexikalischen Kategorien entsprechen. Es wird dabei unterschieden zwischen *N-Features*, die XPs enthalten, wie z. B. Kasusmerkmalen, und *V-Features*, die in lexikalischen Köpfen enthalten sind. Damit eine syntaktische Struktur lizensiert wird, müssen die Flexionsmerkmale der funktionalen Kategorien (N- und V-Features) mit denen der lexikalischen Kategorien *matchen*, wobei die Überprüfung des Matches, das *Checking* der Merkmale, nur in streng lokalen Konfigurationen möglich ist: N-Features können nur in einer Spec-Head- bzw. Adjunkt-Head-Konfiguration, V-Features nur durch Adjunktion an den diese Merkmale tragenden funktionalen Kopf gecheckt werden. Somit impliziert das *Checking* der N-Features XP-Bewegung in die Spezifiziererposition (bzw. XP-Adjunktion an die maximale Projektion) einer funktionalen Kategorie, das *Checking* der V-Features Kopf-zu-Kopf-Bewegung in die Position funktionaler Köpfe.

Ob diese Bewegung *overt* (sichtbar) ist oder erst auf der Ebene der Logischen Form (LF) stattfindet, hängt schließlich vom Typ der jeweiligen Merkmale ab: sogenannte *starke* Merkmale erzwingen eine overte Bewegung (weil sie vor LF – durch *Checking* – gelöscht werden müssen), *schwache* Merkmale dagegen können, dem Ökonomieprinzip *Procastinate* entsprechend, erst auf LF über eine *coverte* Bewegung gecheckt und gelöscht werden. Somit werden parametrische Variationen auf den Typ der zu checkenden Merkmale zurückgeführt.

Die Sicht des *Minimalist Program*s ist dabei eine dynamische, bei der nicht von einer vollständig aufgebauten Struktur ausgegangen wird, innerhalb derer verschiedentlich motivierte Bewegungsprozesse stattfinden, sondern bei der eine Struktur durch das Zusammenspiel von zwei Prozessen, dem Prozeß der *lexical insertion* und dem durch die *Checking Theory* motivierten Bewegungsprozeß, in einem Bottom-up-Verfahren aufgebaut wird.

Mit Hilfe des Klassenkonzepts und des Mechanismus der dynamischen Vererbung läßt sich die *Checking Theory* unmittelbar objektorientiert umsetzen. Während das Klassenkonzept die Verteilung der Merkmale in Form von zweistelligen Methoden an die lexikalischen und funktionalen Kategorien übernimmt, sorgt der Mechanismus der dynamischen Vererbung, mit dem das eben beschriebene Bottom-up-Verfahren simuliert wird, für das Zusammentreffen und somit die Aktivierung dieser zweistelligen Methoden innerhalb der Struktur und ermöglicht dadurch die Simulation der *Checking Theory*.

Objektorientiert interpretiert sind starke Merkmale obligatorische Methoden, die Bewegungen erzwingen, weil sie nur in den entsprechenden Konfigurationen bewiesen werden können (nämlich Spec-Head und Kopf-Adjunktion, wie im Falle der Methoden `top` und `verbzweit`); schwache Merkmale werden im objektorientierten Ansatz dagegen entweder durch fakultative Methoden realisiert oder einfach weggelassen und erzwingen somit keine overte Bewegung.

Die Parallele läßt sich generalisierend weiterführen: V-Features und N-Features entsprechen jeweils einem bestimmten Typ von Methoden, nämlich *V-Feature-Methoden* und *N-Feature-Methoden*, die sich, wie am obigen Beispiel bereits deutlich geworden ist, in der Reichweite ihrer dynamischen Vererbung unterscheiden: V-Feature-Methoden werden immer nur bis zur Mutter ihres Objektes dynamisch vererbt. Dies gilt sowohl für die V-Feature-Methode einer lexikalischen Kategorie (wie z. B. `verbzweit(Self,Other)` oben) als auch für die einer funktionalen Kategorie (s. `verbzweit(Other,Self)` oben). Bei N-Feature-Methoden dagegen verhält sich die Methode der lexikalischen Kategorie anders als die der funktionalen: Erstere (wie z. B. `top(Self,Other)` oben) wird hier auch nur bis zur Mutter ihres Objektes hochgereicht, letztere dagegen muß bis zur maximalen Projektion ihres Objekts vererbt werden (vgl. `top(Other,Self)` oben). Diese Typologisierung von zweistelligen Methoden reicht auch, um die *checking theory* unmittelbar objektorientiert umzusetzen.

Die Konvergenz zwischen objektorientiertem Ansatz und *Minimalist Program* ist unübersehbar. Zu welchen neuen lokalistisch-minimalistischen und somit kognitiv plausiblen Lösungen diese Ansätze führen werden, wird sich noch zeigen. VisualGBX wird z. Zt. im Rahmen eines von der Deutschen Forschungsgemeinschaft geförderten Projekts zur Untersuchung der Verbbewegung in den romanischen Sprachen eingesetzt. Die Ergebnisse dieser Untersuchung werden Anfang 1998 in einer Monographie veröffentlicht.

Bernhard Schröder, Universität Bonn

Pro-SGML: Ein Prolog-basiertes System zum Textretrieval

1 Einleitung

Zunehmend werden Texte, deren Struktur in SGML oder – besonders im *World Wide Web* – in der SGML-Spezialisierung HTML kodiert ist, verfügbar. Damit gewinnt Textretrieval, das auch SGML-Strukturierung berücksichtigt, an Bedeutung. In diesem Beitrag sollen einige Prolog-basierte Werkzeuge vorgestellt werden, die ein SGML-sensitives Retrieval unterstützen. Dabei ist nicht beabsichtigt, einen Werkzeugkasten als abgeschlossenes Produkt vorzustellen, sondern vielmehr einige Teile daraus und einige konzeptuelle Überlegungen zur Vervollständigung des Kastens. Dies geschieht in der Hoffnung, daß der Beitrag hilft, Bestrebungen in eine ähnliche Richtung zu koordinieren und Schnittstellen festzulegen.

Zunächst sollen kurz die relevanten Tatsachen zur Struktur von SGML-Dokumenten festgehalten werden, anschließend soll auf einige Anforderungen an programmierbare Retrievalformalismen eingegangen werden, um schließlich den in *ProSGML* verfolgten Ansatz vorzustellen.

2 Die Struktur von SGML-Dokumenten

SGML-Dokumente gehören immer einem (und evtl. auch mehr als einem) formal bestimmten Dokumenttyp an. Zur Flexibilität von SGML gehört es wesentlich, daß diese Dokumenttypen mit Hilfe von SGML in Form einer *Document Type Definition* (DTD) spezifiziert werden kann.

Die Gliederungselemente eines Textes wie beispielsweise Kapitel, Abschnitte, Unterabschnitte, Überschriften, Aufzählungen, Einzelpunkte in Aufzählungen und beliebige andere wie auch immer bestimmte zusammenhängende Textteile werden in der SGML-Terminologie als *Elemente* bezeichnet. Elementen können Attribute zugeordnet sein, die gewisse Eigenschaften der Elemente näher bestimmen.

Die Textgliederung ist durch SGML nur dahingehend restringiert, daß die Elementbeziehungen in dem Sinne streng hierarchisch sind, daß Elemente andere Elemente nur vollständig enthalten können, Elemente sich also nicht überlappen können. So ist es beispielsweise bei vielen Dokumenten nicht möglich, Seiten und Abschnitte

gleichermaßen als Elemente zu betrachten, da Abschnitte über Seitengrenzen hinweggehen können, Abschnitte ihrerseits aber auch nicht nur vollständige Seiten umfassen. Diese Restriktion kann umgangen werden, wenn einem Dokument mehrere konkurrierende Dokumenttypen zugeordnet werden, z. B. eine Gliederung in inhaltliche Gliederungseinheiten wie Kapitel, Abschnitte usw. und eine konkurrierende in layoutorientierte wie Bände, Seiten, Zeilen. Der SGML-Standard sieht eine derartige Kodierungsmöglichkeit in einem Dokument vor, jedoch wird dies keineswegs von allen SGML-basierten Systemen unterstützt. Das hier vorgestellte System kann auch mit konkurrierenden Dokumenttypen umgehen, doch soll diese Eigenschaft hier nicht weiter thematisiert werden.

Die Möglichkeiten der hierarchischen Gliederung eines Dokumenttyps werden in den Elementdeklarationen, genauer: in den Inhaltsmodellen, die den Elementen zugeordnet werden, beschrieben. Zu jedem Element wird angegeben, welche Elemente oder sonstige Daten (wie Zeichenfolgen) wie oft und in welcher Reihenfolge in ihm enthalten sein können. Diese Spezifikation erfolgt im wesentlichen durch einen erweiterten Backus-Naur-Formalismus, und die Spezifikationsmöglichkeiten sind somit zu denen einer kontextfreien Grammatik schwach äquivalent.

Neben den Inhaltsmodellen für Elemente enthalten DTDs meist noch Deklarationen sogenannter *Entitäten.* Dabei handelt es sich um Platzhalter für andere Zeichenketten, z. B. als leichter les- und memorierbare Alias-Bezeichnungen für Zeichenkodes nicht standardmäßig verfügbarer Zeichen, als Abkürzungen für Zeichenketten oder als Anweisungen an das System andere Dateien anstelle der Entität einzufügen.

SGML-DTDs können außer den genannten Deklarationen noch Angaben zu möglichen Weglassungen von Marken und Attributen enthalten. Ein Element wird in ausführlicher Schreibweise durch eine Anfangs- und eine Endmarke geklammert. In vielen Fällen können Marken jedoch aus dem Kontext ergänzt werden: Der Anfang eines neuen Abschnitts bedingt das Ende des vorhergehenden, zumindest bei Abschnitten gleicher Ebene. Die Endmarke des vorangehenden Abschnitts könnte also in dieser Umgebung ohne Verlust an Information entfallen. In einigen Fällen kann sogar eine Anfangsmarke eines Elements aus der Umgebung erschlossen werden. Sind die erschließbaren Marken als weglaßbar gekennzeichnet, dürfen sie tatsächlich entfallen.

Auch bei Attributen kann es zur Lesbarkeit und Kürze der Markierung beitragen, wenn Attribute mit einem Standardwert weggelassen werden können. Wird beispielsweise mit dem Element *Abschnitt* das Attribut *Schrifttyp* assoziiert, das den Schrifttyp dieses Abschnitts angeben soll, so erleichtert die Konvention, daß das Attribut nicht spezifiziert zu werden braucht, wenn der Standardschrifttyp des Dokuments vorliegt, die Kodierungsarbeit und die Übersichtlichkeit der Kodierung. Die Abkürzungsmöglichkeiten unter SGML verlieren selbstverständlich in dem Maße an Bedeutung, in dem die Editierarbeit durch Spezialeditoren unterstützt wird und die eingesparte Textmenge speichertechnisch unbedeutend wird.

3 SGML-sensitives Textretrieval

Soll Textretrieval von der mit Hilfe von SGML vorgenommenen Strukturierung von Dokumenten Gebrauch machen, so sollte in Suchanfragen auch eine Abhängigkeit der gesuchten Zeichenketten von Elementkontexten herstellbar sein. Eine Mindestanforderung wäre, daß die Suche auf den Kontext eines bestimmten Elements beschränkt werden kann. Die Formulierung derartiger Suchrestriktionen ist im allgemeinen bereits in Textretrievalsystemen mit flacher Elementeinteilung (wie z. B. in *FolioViews* oder durch Nutzung der Sprachunterscheidung auch in *WordCruncher* für Windows), wo Elemente nicht ihrerseits Unterelemente beinhalten können, möglich.

Läßt man wie in SGML auch eine Schachtelung von Elementen zu, so werden zusätzliche Bestimmungsoptionen für das Verhältnis von gesuchten Zeichenketten und Elementen wünschbar: Es ist zu unterscheiden zwischen dem unmittelbaren Enthaltensein in einem Element (derart, daß dieses Element das kleinste ist, das die gesuchte Zeichenkette enthält) und einem bloß mittelbaren. Ist beispielsweise *Abschnitt* ein Element, das *Zitate* als weitere Elemente enthalten kann, so kann es von Interesse sein, ob ein bestimmtes Wort überhaupt in einem Abschnitt mittelbar oder unmittelbar enthalten ist, ob es in einem *Zitat* enthalten ist (und somit nicht notwendigerweise zum aktiven Wortschatz des Autors gehört) oder im Gegenteil unmittelbar in einem *Abschnitt* enthalten ist und somit nicht zu einem *Zitat* gehört.

Sind in einem *Zitat* oder auch unmittelbar in einem *Abschnitt* weitere Elemente wie z. B. *Sätze* enthalten, so wird es möglicherweise sinnvoll nach Vorkommen von Zeichenketten zu suchen, die unmittelbar in einem *Satz* und mittelbar in einem *Abschnitt* enthalten sind, nicht aber mittelbar in einem *Zitat.* Allgemeiner betrachtet erfordert dies Möglichkeiten, Bedingungen an den Pfad, der von den umfassendsten zu den engsten Elementen führt, zu formulieren.

Schließlich ist es ein wesentlicher Teil der Annotationsmöglichkeiten unter SGML, Elemente mit Attributen versehen zu können. Nicht selten enthalten gerade diese die Informationen, auf die man eine Suche restringieren möchte: Wenn bei einem mehrbändigen Werk ein Element *Band* in der DTD eingeführt ist, das den Inhalt eines Bandes umfaßt, und die jeweilige Nummer des Bandes als Attribut bei diesem Element spezifiziert ist, so muß die Suche nach einem Wortvorkommen in Band 3 auf das Attribut des Elements Bezug nehmen können.

Die Mächtigkeit einer Suchmaschine hängt in SGML-Umgebungen wie auch im Zusammenhang anders strukturierter Dokumente wesentlich davon ab, welche Möglichkeiten zum einen gegeben sind, boolesche Kombinationen von Suchanfragen zu bilden, zum anderen davon, inwieweit Optionen bestehen, von bestimmten Zeichenketten oder Wortformen zu abstrahieren, wie, um nur einige zu nennen, durch die Verwendung regulärer Ausdrücke, durch eine automatische Lemmatisierung, durch eine phonetische oder fehlertolerante Suche oder durch die Einbeziehung statistischer Verfahren zur Ermittlung thematischer Nähe. Darauf soll hier mit Ausnahme der Verwendung regulärer Ausdrücke nicht weiter eingegangen werden; Module, die eine der

skizzierten Funktionalitäten beitragen und unter Prolog verfügbar gemacht werden können, sind in dem hier vorgestellten Konzept leicht zuzuschalten.

4 Beyond Text Retrieval

Für die meisten Bedürfnisse des Textretrieval scheint der oben ausgebreitete Wunschkatalog, was die Berücksichtigung der SGML-Strukturierung von Dokumenten angeht, weitestgehend abgeschlossen. Avanciertere SGML-Textbetrachter mit Suchfunktionalität bieten zumindest einige der genannten Suchoptionen in Form graphischer Anfrageoberflächen an.

Es gibt jedoch eine Reihe von SGML-Anwendungen, bei denen es nötig sein kann, den Wunschkatalog zu erweitern. Kodiert man syntaktische Konstituenten als SGML-Elemente, wie es in einer Treebank geschieht, so werden für syntaktische Untersuchungen nicht nur vertikale Beziehungen zwischen Elementen, sondern auch horizontale von Bedeutung sein, wenn es darum geht das syntaktisch annotierte Korpus linguistisch auszuwerten: Welche Artikel kommen in Artikel-Substantiv- oder Artikel-Adjektiv-Substantiv-Konstellationen usw. vor, bei denen das Substantiv *E-Mail* lautet? Bei solchen Fragestellungen geht es also um Abfolgemuster von Elementen.

Ebenfalls um die Frage, welche Abfolgekonstellationen von Elementen vorkommen, geht es bei vielen Validierungstests für SGML-Dokumente. Zwar nimmt ein SGML-Parser bereits eine formale Validierung vor, er überprüft für ein gegebenes Dokument jedoch nur, ob es sich um eine Instanz des in der DTD bestimmten Dokumenttyps handelt. Einschränkungen, die sich nicht in den Inhaltsmodellen der DTD darstellen lassen, und hierzu gehören alle Einschränkungen, die nicht durch kontextfreie Grammatiken darstellbar sind, sind durch diese Art der Validierung nicht überprüfbar.

In vielen Fällen wird auch mit Absicht auf eine DTD zurückgegriffen, die nicht alle beabsichtigten Beschränkungen von Elementkonstellationen berücksichtigt, weil man durch den Anwendungszweck an eine bestimmte DTD gebunden ist (z. B. an eine HTML-DTD bei Publikation im *World Wide Web*) oder weil man eine standardisierte oder eine für eine größere Menge von Dokumenten einheitliche DTD verwenden möchte. Nichtsdestoweniger kann es auch in solchen Fällen nötig sein, die intendierten Beschränkungen formal zu überprüfen. Der HTML-Dokumenttyp beispielsweise läßt es zu, daß Überschriften der Stufen H1 (größte Stufe), H2 (zweitgrößte Stufe), H3 (drittgrößte Stufe) usw. in beliebiger Reihenfolge unmittelbar oder mittelbar aufeinander folgen. Soll jedoch in einem HTML-Dokument sichergestellt werden, daß unmittelbar ohne einen zwischengeschalteten Textabschnitt aufeinanderfolgende Überschriften nur in der Konstellation einer Überschrift mit der nachfolgenden Überschrift in der nächstkleineren Stufe vorkommen dürfen und auf eine Überschrift eines bestimmten Typs nach einem oder mehreren Textabschnitten nur eine Überschrift des nächstkleineren Typs, desselben Typs oder größerer Typen vorkommen darf, so kann dies durch Suchanfragen verifiziert werden, die nach Konstellationen suchen, die von

diesem Schema abweichen. Endet die Suche ergebnislos, ist das Dokument formal in dieser Hinsicht überprüft.

Suchanfragen dieser Art setzen für die horizontale Kombinierbarkeit von Elementen ähnliche Spezifikationsoptionen voraus, wie für die von Zeichenketten oder Wortformen. Eine Adaption eines Formalismus regulärer Ausdrücke scheint hier viele Anwendungen abzudecken.

5 Formalismen zur programmierbaren Anfrage an SGML-Datenbanken

Einige der oben beschriebenen Aufgaben erfordern eine Anfragesprache mit skriptsprachlichen Erweiterungen. Diese kann auf unterschiedliche Weisen realisiert werden: Eine bestehende Skript- oder Programmiersprache kann um zuschaltbare Module oder Bibliotheken zur SGML-Datenbankanfrage erweitert werden, oder eine Anfragesprache mit Programmierelementen wird entwickelt.

Den ersten Weg gehen beispielsweise die *perlSGML*-Werkzeuge (vgl. *http://www.oac.uci.edu/indiv/ehood/perlSGML.html*), die unter der Skriptsprache *Perl* als Programme, Module oder Bibliotheken zur Verfügung stehen. Sie liefern dem Benutzer Möglichkeiten der DTD-Analyse und des Parsing von Dokumentinstanzen. Über diese können die *Perl*-eigenen Filtermöglichkeiten zur Anfrage und Suche eingesetzt werden.

Der zweite Weg wird von *SgmlQL* im Rahmen eines Multext-Projektes beschritten (vgl. *http://www.lpl.univ-aix.fr/projects/multext/*). Auf UNIX-Betriebssystemen steht mit dem *MtSgmlQL*-Interpreter ein Interpreter für *SgmlQL*-Skripts zur Verfügung, mit dem sich Anfragen an normalisierte SGML-Dateien formulieren lassen. Die *Normalisierung* von SGML-Dateien erreicht man durch Voranschaltung eines SGML-Parsers, der Verweise auf externe Dateien auflöst und in SGML zugelassene Auslassungsmöglichkeiten von kontextuell erschließbaren Anfangs- oder Endmarkierungen rückgängig macht. In *SgmlQL* lassen sich jedoch auch andere Operationen als Anfragen durchführen: Elemente können umbenannt werden, Attribute können manipuliert werden oder der Inhalt eines Elements kann dem nächst umfassenderen Element zugeschlagen werden, um einige Beispiele für den bereitgestellten Funktionsumfang zu nennen.

Die hier behandelten Prolog-Werkzeuge sind dem erstgenannten Weg zuzuordnen. Sie stellen unter der Programmiersprache Prolog Prädikate zur Anfrage an und Manipulation von SGML-Dateien zur Verfügung.

Ein anderes Merkmal, nach dem programmierbare Anfragewerkzeuge zu unterscheiden sind, betrifft die Interaktivität: Stellen sie eine Shell mit einer zeilenorientierten Anfragesprache und/oder eine graphische Benutzeroberfläche zur Verfügung oder handelt es sich um eine Programmbibliothek, deren Module ausschließlich aus anderen Programmen heraus aufrufbar sind? Bieten sie im Falle einer Shell oder einer graphischen Benutzeroberfläche auch Schnittstellen für den Aufruf aus Programmen? *PerlSGML* ist als Programmbibliothek konzipiert. Allerdings erlaubt *Perl* als Inter-

pretersprache eine relativ unkomplizierte Programmierung von interaktiven Schnittstellen. Bei *SgmlQL* hat man es mit einer programmierbaren interaktiven Shell zu tun, zu der Schnittstellen zum Aufruf aus Programmen bestehen. Beide Systeme bieten keine graphischen Schnittstellen an.

ProSGML verbindet als Prolog-Erweiterung die vom Prolog-Interpreter vorgegebene Shell mit einer Schnittstelle zu einer Programmiersprache. Dieselbe Anfragesyntax kann sowohl interaktiv als auch im Stapelmodus verwendet werden. Eine graphische Benutzeroberfläche bietet *ProSGML* derzeit noch nicht an, doch ist an eine Entwicklung einer graphischen Benutzeroberfläche mit Hilfe von gedacht (s. u.).

6 ProSGML

Die Programmiersprache Prolog eignet sich durch Ihre prädikatenlogische Basis sehr gut zur Darstellung komplexer Datenbankanfragen. Sie enthält aufgrund dieser prädikatenlogischen Basis bereits die booleschen Verknüpfungsmöglichkeiten, die mächtigere Anfragesprachen aufweisen. Darüber hinausgehend lassen sich in Prolog Anfragen bilden, die in SQL-artigen Anfragesprachen nicht realisierbar sind, aber in dem von Prolog abgedeckten Ausschnitt der Prädikatenlogik, der Hornklausellogik, definierbar sind.

Nehmen wir beispielsweise an, daß in einer relationalen Datenbank die Information abgelegt ist, welche geographischen Einheiten innerhalb welcher anderer liegen. Die Datenbank könnte z. B. die Information enthalten, daß Poppelsdorf in Bonn, Bonn in der Kölner Bucht, die Kölner Bucht in Nordrhein-Westfalen, Nordrhein-Westfalen in der Bundesrepublik und die Bundesrepublik in Europa liegt. Enthalten die Datensätze der Datenbank nur diese Instanzen der Relation *liegt_in* und nicht etwa die zusätzlichen mit der Information, daß beispielsweise Poppelsdorf in der Bundesrepublik liegt, ist also die Relation des "Liegens in" nur als "unmittelbares Liegen in" verstanden und die transitive Erweiterung dieser Relation in den Datensätzen der Datenbank nicht vorgenommen worden, so kann man mit Hilfe einer SQL-Abfrage nicht herausbekommen, ob Poppelsdorf in der Bundesrepublik liegt, zumindest nicht, solange man nicht weiß, wieviele Zwischenschritte des "unmittelbaren Liegens in" erforderlich sind, um dies zu ermitteln.

Verfügen wir jedoch über ein Prolog-Prädikat `liegt_in/2`, das auf die Instanzen der gleichnamigen Datenbankrelation zutrifft, so liefert die Anfrage

```
?- liegt_in(X,Y).
```

Anfragebeispiel 1

als Lösung genau die Instanzen der Relation. Definiert man das zusätzliche Prädikat `liegt_mittelbar_in/2` als

```
liegt_mittelbar_in(X,X).
liegt_mittelbar_in(X,Y) :-
     liegt_in(X,Z),
     liegt_mittelbar_in(Z,Y).
```

also als transitive und reflexive Erweiterung von `liegt_in/2` (jeder Ort liegt mittelbar in sich selbst, und X liegt mittelbar in Y, wenn X unmittelbar in einem Z liegt und Z mittelbar in Y), dann liefert die Anfrage

```
?- liegt_mittelbar_in(X,Y).
```

Anfragebeispiel 2

alle gesuchten Instanzen. Da die Prolog-Datenbasis jederzeit von der Eingabeaufforderung aus um Definitionen wie die von `liegt_mittelbar_in/2` erweitert werden kann oder neue höherstufige Prädikate derart definiert werden können, daß sie angewandt auf andere Prädikate die gewünschten Erweiterungen ergeben, bietet Prolog eine geeignete Basis für eine dynamisch erweiterbare Datenbankanfrageshell. Werden, um an das obige Beispiel anzuknüpfen, mehrfach reflexive und transitive Erweiterungen von Relationen benötigt, so kann bei geeigneter Definition ein höherstufiges Prädikat `transitiv/3` den Dienst für alle diese Relationen tun, den oben `liegt_mittelbar_in/2` für das eine Beispielprädikat `liegt_in/2` tut. Die Anfrage

```
?- transitiv(X,Y,liegt_in(X,Y)).
```

Anfragebeispiel 3

würde dann dieselben Instanzen liefern wie die in Anfragebeispiel 2 genannte Anfrage.

Gibt es auch im Bereich des Textretrieval Beispiele für einen Bedarf nach in ähnlicher Weise erweiterten oder modifizierten Relationen? Die Antwort auf diese Frage hängt natürlich wesentlich von der Art der Datenrepräsentation in der Textdatenbank ab. Ist beispielsweise in der Datenbank die Relation des *unmittelbaren* Enthaltenseins von Elementen in anderen Elementen gespeichert, was weit weniger Redundanz mit sich bringt, als alle mittelbaren und unmittelbaren Enthaltenseinsrelationen in der Datenbank zu repräsentieren, so entsteht, ähnlich dem geographischen Beispiel weiter oben, die Notwendigkeit, eine abgeleitete Relation des mittelbaren Enthaltenseins einzuführen.

Andere auf ähnlicher Ebene angesiedelte Anforderungen betreffen die Möglichkeit, Attributwerte von Elementen auf in ihnen enthaltene Elemente zu vererben, solange für sie keine anderen Attributwerte angegeben sind.

```
...
<FONT NAME=courier>Dies ist eine Textpassage in Courier.
<EM>Und dieser Teil ist hervorgehoben.
</EM></FONT>
...
```

Textbeispiel 1

Bei geeigneter Definition soll die Anfrage

```
?- element('EM', E), ererbtes_attribut(E,'NAME',courier).
```

Anfragebeispiel 4

also die Anfrage nach einem <*EM*>-Element mit dem von einem übergeordneten Element ererbten Attributwert *courier* für das Attribut *NAME,* das <*EM*>-Element im obigen Beispiel finden, nicht aber das <*EM*>-Element im folgenden Beispiel.

```
...
<FONT NAME=courier>Dies ist eine Textpassage in Courier.
<EM NAME=times>Und dieser Teil ist in einer anderen Schrift
hervorgehoben.
</EM></FONT>
...
```

Textbeispiel 2

Die Offenheit der Prolog-Schnittstelle gestattet es auch, die Anfragesprache um Prädikate zur Lemmatisierung, Relationen konzeptueller Ähnlichkeit sowie Funktionen zur statistischen Aggregierung einzubinden, soweit entsprechende Prolog-Bibliotheken zur Verfügung stehen.

```
?- lemma(fahren, WF), enthaelt_wortform(E,WF),  element('H1',E).
```

Anfragebeispiel 5: Die Anfrage ist zu lesen als: Suche alle Flexionsformen des Lemmas fahren, *die in Elementen des Typs* <H1> *enthalten sind.*

Anfrage 5 zeigt, vorausgesetzt ein Prolog-Prädikat `lemma/2` zur Lemmatisierung bzw. Flexionsformengenerierung steht zur Verfügung, wie alle Vorkommen von Flexionsformen von *fahren* in Überschriften des Typs <*H1*> gefunden werden können.

Ursprünglich ist ProSGML für Anfragen zur Kontrolle konsistenter Kodierung entworfen; dazu gehören (hier natürlichsprachlich paraphrasierte) Fragen wie die folgenden:

- Kommen <*WURZEL*>-Elemente nur innerhalb von <*MATH*>-Elementen vor?
- Kommt *griechischer* Text in *lateinischem* vor? Tritt das in *Fußnoten* auf? Und sind diese Textstellen länger als ein Buchstabe (handelt es sich also nicht um griechische Numerierung)? (Selbstverständlich ist vorausgesetzt, daß die hervorgehobenen Kategorisierungen von Textteilen in SGML kodiert sind.)
- Kommen römische Zahlen in Überschriften zweiter Ebene vor?

Anfragebeispiel 6: Beispiele für Textstruktur und Kodierung betreffende Anfragen als natürlichsprachliche Paraphrasen

Es gibt eine Reihe von Optionen, eine Textdatenbank mit einer Prolog-Schnittstelle zu implementieren. Welche Option die günstigste ist, ist in hohem Maße vom Zweck abhängig. Die Benutzerschnittstelle von *ProSGML* soll möglichst unabhängig von der gewählten darunterliegenden Datenbankstruktur sein und dem Benutzer eine unter veränderten Datenbankstrukturen gleichbleibende Sicht auf die Daten bereitstellen. Die wichtigsten Entscheidungen, die der dargestellten Implementierung unterhalb der Benutzersicht zugrunde liegen, betreffen die *Einheiten der Basissegmentierung* der Textdatenbank sowie die *Datenbankarchitektur*.

6.1 Aufbau von ProSGML

Von der *ProSGML*-Anfrageschnittstelle aus wird nicht direkt auf die SGML-kodierten Daten zugegriffen, sondern auf eine für die Anfrage vorbereitete Form der Daten. Derzeit arbeitet *ProSGML* mit zwei Formen der Textdatenrepräsentation:

1. der Repräsentation der Textdaten in einer relationalen Datenbank und
2. der Repräsentation der Textdaten in der Prolog-Datenbasis.

Die erstgenannte Option setzt die Verfügbarkeit einer Prolog-Schnittstelle zu der verwendeten relationalen Datenbank voraus, ist jedoch gerade bei größeren Textmengen aus Effizienzgründen der direkten Repräsentation des Textes in der Prolog-Datenbasis vorzuziehen. Diese Option wurde mit *Quintus-Prolog* 3.3 und der *ProDBI-ODBC*-Schnittstelle, Version 4.0, realisiert. Über die ODBC-Schnittstelle wurde auf die Textdatenbanken in einer *ORACLE7*-Serverinstallation und einer lokalen *MS-Access*-Installation zugegriffen.

Beide Optionen setzen die Normalisierung des SGML-kodierten Textes durch einen SGML-Parser voraus. Hier wurde der Freeware-Parser *nsgmls* von James CLARK verwendet (vgl. *http://www.jclark.com/sp/nsgmls.htm*). Die normalisierte Textinstanz kann nun durch ein relativ einfaches Filterprogramm, hier in *awk* programmiert, in ein für eine relationale Datenbank geeignetes Format oder in ein Prolog-Format überführt werden. Es bestehen unterschiedliche Segmentierungsoptionen, als textuelle Basissegmente können Wortformen bzw. Satzzeichen oder Zeilen oder Sätze (soweit in SGML kodiert) oder andere SGML-Elemente gewählt werden. Wichtig ist nur, daß die Basissegmente selbst nicht weitere SGML-Elemente enthalten. Die relationale Datenbank bzw. Prolog-Datenbank wird im wesentlichen aus drei Relationen aufgebaut:

Tabelle	Spalten	Prolog-Prädikat
Basissegmente	graphische Form Typ Zähler	`basissegment/3`
Elemente	Elementname Zähler (=Anfang) Ende	`element/3` bzw. `element/2`[1]
Attribut	Attributname Wert Elementzähler	`attribut/3`

Abb. 1: Tabellen zur relationalen Repräsentation der Textdatenbank

Auf die Instanzen der Relationen kann direkt über die ebenfalls in der Tabelle angegebenen Prolog-Prädikate zugegriffen werden. Auf der Grundlage von `basissegment/3` sind Prädikate für einzelne Wortformen und Satzzeichen definiert. Wenn Wortformen und Satzzeichen selbst Basissegmente sind, ist diese Definition trivial.

[1] In der zweistelligen Form werden Elementanfang und Ende in einer Liste zusammengenommen.

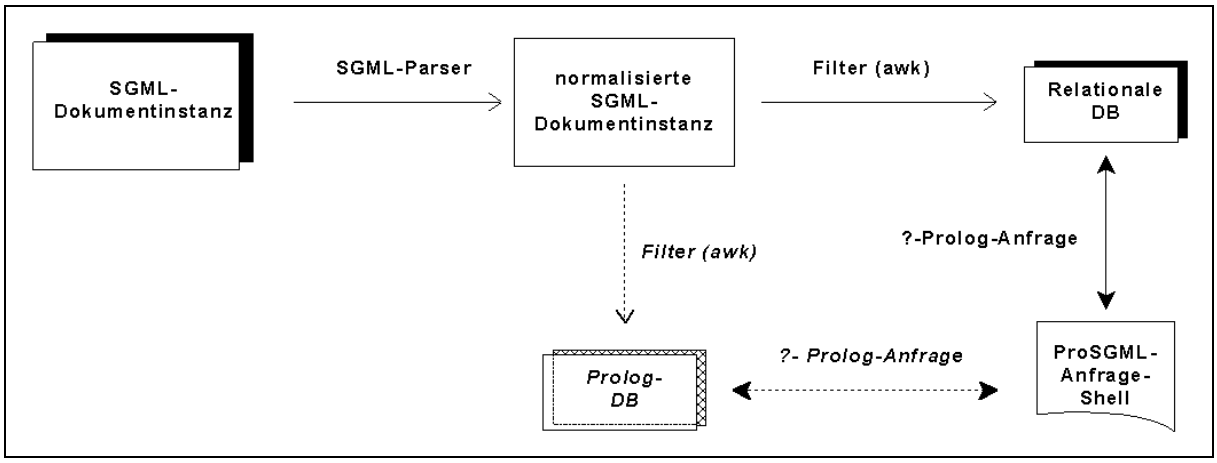


Abb. 2: Aufbau von ProSGML

Aufgrund dieser Basisprädikate sind die Prädikate der Benutzerschnittstelle definiert. Allerdings können auch die Basisprädikate selbst direkt an der Benutzerschnittstelle verwendet werden.

6.2 Benutzerprädikate

Drei Gruppen grundlegender Prädikate stehen an der Benutzerschnittstelle zur Verfügung:

1. *Elementschaftsrelationen*
   oder Relationen des Enthaltenseins, dazu gehören insbesondere die folgenden:
   `enthaelt_element/4`
   `enthaelt_element_unmittelbar/4`
   `enthaelt_wortform/4`
   `enthaelt_wortform_unmittelbar/4`
   Die vierstellige Form der Prädikate ergibt sich daraus, daß als Argumente neben den Elementnamen bzw. der graphischen Form der Wortformen auch deren Zähler dienen. Zwei und dreistellige Varianten dieser Prädikate, bei denen Zähler nicht oder ausschließlich als Argumente dienen sind ebenfalls definiert.

2. *Vergleichsrelationen für reguläre Ausdrücke*
   Reguläre Ausdrücke werden auf zwei Ebenen bereitgestellt: auf der Wortformen- bzw. Basissegmentebene und auf der über Wortformen bzw. Basissegmente hinausgehenden Ebene, gewissermaßen der Textstrukturebene.
   Die regulären Ausdrücke auf Wortformenebene sind syntaktisch den regulären Ausdrücken unter *Perl* oder *awk* angelehnt. Das Prädikat
   `regex_segment/2`
   liefert zu einem gegebenen regulären Ausdruck passende Basissegmente aus der Textdatenbank.
   Die regulären Ausdrücke auf Textstrukturebene berücksichtigen Elementfolgen, Attribute und verschiedene Typen von Basissegmenten, die ihrerseits als reguläre Ausdrücke spezifiziert sein können.

Das Prädikat

`regex_struktur/3`

liefert zu einem gegebenen regulären Ausdruck Anfang und Ende der Textabschnitte (durch den jeweiligen Zähler repräsentiert), auf den der reguläre Ausdruck zutrifft.

3. Ausgabeprädikate

Ausgabeprädikate werden mit dem Anfang und dem Ende des auszugebenden Textabschnitts aufgerufen. Zwei Gruppen sind zu unterscheiden: Prädikate die den Text als Prolog-Liste zurückgeben und solche, die ihn in eine Datei oder auf den Bildschirm ausgeben. Zur ersten Gruppe gehört

`liste_von_bis/3`

zur zweiten Gruppe

`schreibe_von_bis/2`

Diese beiden Prädikate geben den gewählten Textabschnitt mitsamt allen enthaltenen Elementmarken aus. Varianten dieser Prädikate unterdrücken diese Marken oder umschließen den Textabschnitt mit den Marken der ihn umfassenden Elemente.[2]

Beispiele für *ProSGML*-Formulierungen der im Anfragebeispiel 6 oben natürlichsprachlich gestellten Fragen finden sich im folgenden Beispiel.

```
?- element('WURZEL',E1), \+ (element('MATH',E2),
     enthaelt_element(E2,E1)).
?- element('LANG',E1,EZ), attribut('NAME',griech,E1),
     element('LANG',E2,_), attribut('NAME',lat,E2),
     enthaelt_element(E2,E1),
     regex_struktur('wrx(.)',E1,EZ).
?- enthaelt_wortform('H2',S),
     regex_segment('[IVXLCDMivxlcdm]+\.?',S),
```

Anfragebeispiel 7: ProSGML-Formulierung der Anfragen aus Anfragebeispiel 6

7 Ausblick

Die Anfrageschnittstelle von *ProSGML* erschließt sich als Prolog-Erweiterung derzeit vollständig nur dem Prolog-kundigen Benutzer. An einer an SQL angelehnten Anfragesprache wird jedoch gearbeitet. Diese soll auch für die Generierung optimierter Datenbankanfragen sorgen. Wie auch bei anderen Prolog-Anfragen kann es ohne eine Kenntnis der prozeduralen Prolog-Semantik, also der Art und Weise, wie Anfragen ausgewertet werden, vorkommen, daß Anfragen zwar genau die gewünschten Ergebnisse liefern, dies aber in extrem ineffizienter Weise tun.

---

[2] Man beachte, daß dies zur Ausgabe einer in sich korrekten SGML-Dokumentinstanz führen kann, aber nicht muß. Ist beispielsweise der auszugebende Textabschnitt ausschließlich eine Überschrift, fordert jedoch das Inhaltsmodell, daß neben der Überschrift ein weiterer Absatz vorhanden sein muß, so ist der ausgegebene Text nicht mit diesem Inhaltsmodell konform.

Ebenfalls wurden auch erste Schritte in Richtung auf die Entwicklung einer graphischen Benutzerschnittstelle getan, hierzu bieten sich unter Prolog die Entwicklungsumgebung *XPCE* (vgl. *http://swi.psy.uva.nl/projects/xpce/home.html*) bzw. im speziellen Fall das Derivat *ProWindows 3.1* für Quintus-Prolog an.

Zu einer recht umfassenden Sprache zur Textmanipulation ließe sich ProSGML erweitern, wenn die Verwaltung verschiedener Textversionen ermöglicht würde und Editierfunktionen hinzugefügt würden. Die Verwendung von *ProSGML* als Filter für die Herstellung beliebiger Ausgabeformate ist derzeit nur über die Definition eigener Prolog-Prädikate möglich.

7.1 Anhang: Einige Performanzwerte

Die Performanz von *ProSGML* wurde für einige grundlegende Anfragen an einem Korpus (Auszug aus *Kants gesammelten Schriften*) von etwa 163.000 Wortformen gemessen. Das Korpus enthielt 13.092 SGML-Elemente. Als relationales Datenbankmanagementsystem wurde eine lokale Installation von *MS-Access* 2.0 verwendet, die Messungen wurden auf einem 100-Mz-Pentium-PC mit ca. 48,5 MB RAM unter Quintus-Prolog durchgeführt, von Quintus-Prolog aus wurde die Datenbank über die ODBC-Schnittstelle angesprochen. Die Zeitmessungen erfolgten durch Aufruf von `tcp_now/1` aus dem *Quintus-Prolog-tcp*-Paket unmittelbar vor und nach der jeweiligen Anfrage.

Anfrage	**Instanzen insgesamt**	**Zeit [s]**
alle Instanzen von 'die'	5.014	2,90
eine Instanz von 'die'	5.014	0,07
alle Instanzen von 'Kant'	2	0,07
eine Instanz von 'Kant'	2	0,07
alle Instanzen aller Wortformen	163.000	109,39
alle Instanzen von 'dada'	0	0,07
alle Instanzen des Elements LINE	12096	10,70
eine Instanz des Elements LINE	12096	0,07
alle Instanzen des Elements PAGE	345	0,39
eine Instanz des Elements PAGE	345	0,07
alle Instanzen des Elements DADA	0	0,07
alle Instanzen von 'Kant' im Element LINE	2	0,48
alle Instanzen von 'Kant' im Element PAGE	2	0,43
eine Instanz von 'Kant' im Element LINE	2	0,29
eine Instanz von 'Kant' im Element PAGE	2	0,24

Abb. 3: Einige Performanzwerte von ProSGML

Einige Versuche mit einem über ein LAN verbundenen *ORACLE7*-Server unter Windows NT zeigen eine Performanzverbesserung um etwa 30%. Die Performanz bei Benutzung der *ORACLE*-Schnittstelle scheint relativ konstant auch bei einer Vergrößerung des Datenbestands um einen Faktor 10 zu sein. Hier sollen allerdings noch weitere Messungen mit größeren Datenmengen vorgenommen werden.

Christa Womser-Hacker, Universität Regensburg
Walter Zettel, Bayerisches Landeskriminalamt, München

Experimentelle Ergebnisse zur Verwendung struktureller Texteigenschaften für eine gewichtete Indexierung

1 Einführung

Die Rolle der Textstruktur ist im Rahmen automatischer und intellektueller Abstracting-Verfahren näher untersucht worden (cf. z. B. ENDRES-NIGGEMEYER 1992, HAHN & REIMER 1986) und übernimmt auch im Bereich des Textverstehens und in der Schreibprozeß- und Hypertextforschung (cf. HAYES & FLOWERS 1983, KINTSCH & VAN DIJK 1983) eine wichtige Funktion. Aus der Perspektive des Information Retrieval (IR) ist die Hypothese zu verfolgen, inwieweit sich die Ergebnisse dieser Arbeiten für eine erweiterte Repräsentation von Dokumentinhalten nutzbar machen lassen. Die Grundannahme besteht in diesem Kontext darin, daß informationstragende Terme nicht an beliebigen Positionen im Text verteilt sind, sondern an bestimmten Punkten eine stärkere Konzentration aufweisen. Derzeit findet bei den Objekten bzw. Dokumenten ein starker Wandel statt, so daß zum einen Multimediaobjekte Eingang finden, zum anderen keine Einschränkungen bzgl. Länge und Struktur mehr vorliegen. Standardisierungen (ISO 8879 1986), wie sie sich z. B. in formalen Dokumentstrukturbeschreibungssprachen wie SGML bzw. HTML niederschlagen, erhalten die Struktur der Objekte und machen sie für die weitere Verarbeitung zugänglich. Die Einbeziehung älterer Dokumente ist dabei problematisch, da die Strukturinformation bei der Umsetzung auf ein elektronisches Medium meist getilgt wurde.

Für die Repräsentation von Texten haben sich bisher im IR statistische Ansätze (meist mit einer sehr einfachen morphologischen Komponente) durchgesetzt, welche auf der Basis von Termfrequenzen und -verteilungen Gewichtungen berechnen, die die Bedeutung der Terme widerspiegeln. Meist wirken hier verschiedene Verfahren (*term frequency tf, inverse document frequency IDF*, Relativierungen an der Dokumentlänge, der Kollektionsgröße etc.) zusammen. Trotz nahezu 30jähriger IR-Forschung stellt die Frage nach dem optimalen Repräsentationsverfahren vor allem bedingt durch die Heterogenität der Objekte ein immer noch interessantes Gebiet dar (cf. WOMSER-HACKER 1996). Neben Verfahren des Relevance Feedback, welche eine iterative Bewertung der Antwortdokumente durch den Benutzer in die Termgewichtung einfließen lassen, gewinnt zunehmend die Dokumentstruktur an Bedeutung, um sie für eine Termgewichtung nutzbar zu machen.

Im folgenden wird ein Experiment dargestellt, das gängige statistische Ansätze mit solchen vergleicht, welche formal-strukturelle Texteigenschaften zur Gewichtung von Termen benutzen.

2 Auswahl struktureller Texteigenschaften

Als empirische Basis konnte neben anderen Untersuchungen (z. B. PAICE 1990, PAIJMANS 1994) insbesondere auf ENDRES-NIGGEMEYER 1992 zurückgegriffen werden. Dort liegt eine Modellierung der kognitiven Vorgänge bei Repräsentationsexperten vor, wenn sie die Tätigkeiten des Abstrahierens, Indexierens und Klassifizierens vollziehen. ENDRES-NIGGEMEYER 1992 isoliert aus den zugrundeliegenden Fallbeschreibungen 552 Strategien[1], die bei den Experten beobachtet werden konnten. Im Hinblick auf eine Erweiterung der Indexierungssprache kann eine Untermenge dieser Strategien als relevant erachtet werden. Die Experten ließen sich bei ihrer Arbeit in starkem Maße von der formalen Struktur der Dokumente leiten, wobei formalstrukturelle (Titel, Abstract, Dokumentgliederung, Überschriften, Unterschriften etc.) und bildhaft besonders ausgezeichnete Stellen (Listen, Tabellen, Abbildungen etc.) eine wichtige Rolle bei der Ermittlung der Bedeutungselemente einnahmen. Daraus läßt sich u. E. die Begründung ableiten, daß derartige Elemente auch in der automatischen Indexierung erfolgreich eingesetzt werden können, da sie wichtige Bedeutungsträger sind.

3 Experiment

Ziel des bei der Informationswissenschaft der Universität Regensburg in Zusammenarbeit mit dem IZ Sozialwissenschaften in Bonn durchgeführten Experiments ist eine Antwort auf die Frage, welche Auswirkungen die Einbeziehung von Strukturinformation auf die Retrievaleffektivität hat und ob möglicherweise aufwendige statistische Gewichtberechnungen ersetzt bzw. ergänzt werden können. Dies basiert auf der Grundüberlegung, daß wichtige Informationen eher in exponierten Strukturelementen eines Dokuments auftreten.

3.1 Dokumente, Anfragen und Relevanzbewertung

Die Dokumentbasis bestand aus 170 Artikeln (ca. 10 Mb) der Kölner Zeitschrift für Soziologie und Sozialpsychologie der Jahrgänge 1989-1994, die in Volltextform vorlagen. Das IZ Sozialwissenschaften in Bonn stellte parallel dazu 18 zur Dokumentmenge kompatible Anfragen zur Verfügung und leistete die Relevanzbewertung der Dokumente.

[1] Die Zusammenführung im Modell ist problematisch, hier aber nicht von Relevanz.

Anfragebeispiele:

Nr. 10 Meßinstrumente, Meßmethoden, Messung und Methoden bzw. Methodologie in der Soziologie bzw. in der empirischen Sozialforschung
Nr. 13 Kriminalität oder Delinquenz bei Frauen
Nr. 15 Transformations(prozeß) in Deutschland, der DDR, den neuen Bundesländern und Osteuropa

Die Relevanzbewertung erfolgte nach einer vierstufigen Skala:

sehr_relevant, relevant, weniger_relevant, nicht_relevant.

Bei der Integration der Grundparameter, d. h. der Anzahl der relevanten und nichtrelevanten Dokumente, in die Bewertungsmaße, wurden verschiedenartige Mengenbildungen erprobt und deren Auswirkungen kontrolliert. Letztendlich fiel die Entscheidung (gemeinsam mit dem Juror), den Schnitt zwischen *relevant* und *weniger_relevant* anzusetzen.

3.2 Strukturelemente und Gewichtung

In ZETTEL 1996 wurde eine IR-Evaluierungsumgebung (PAIRS[2]) entwickelt, welche den Vergleich verschiedener Indexierungsansätze zuläßt.

PAIRS ist eine Sammlung von *Perl*-Skripten, die im Zusammenspiel mit dem SGML-Parser *Sgmls* aus den SGML-markierten Dokumenten eine spezielle invertierte Indexdatei erzeugt. Diese zeichnet sich dadurch aus, daß sie neben der Dateireferenz und Frequenz- und Vorkommensinformation auch die Strukturmerkmale enthält. PAIRS bietet eine Evaluationsumgebung, welche bzgl. einer vorgegebenen Bewertung einer Anfragensammlung einen Vergleich beliebiger strukturbedingter Gewichtungen ermöglicht. Zusätzlich ist eine einfache Retrievalshell mit boolescher Anfrage enthalten. Mit PAIRS können Faktoren, die keinen Einfluß auf den Vergleich nehmen sollen (z. B. verschiedene linguistische Komponenten, Stoppwortlisten), konstant gehalten werden.

Für das Experiment wurden unter Rückgriff auf die empirische Basis (cf. u. a. ENDRES-NIGGEMEYER 1992) 13 verschiedene Strukturmerkmale[3] unterschieden:

1. Term kommt im Titel oder Untertitel vor
2. Term kommt in einer Überschrift vor
3. Term kommt in einer Unterüberschrift vor
4. Term kommt in einer Unterunterüberschrift vor
5. Term kommt in einer Bild- oder Tabellenüber-/unterschrift vor
6. Term kommt in einer Aufzählung oder Liste vor
7. Term kommt im Abstract vor

---

[2] Perl as Information Retrieval System.

[3] Diese Liste läßt sich jederzeit erweitern. Auch müßten im weiteren kombinierte, hierarchisch geordnete tags einbezogen werden, sodaß z. B. eine Abstractmarkierung eine Absatzmarkierung verdrängt oder sich ein kombiniertes Element auf die Gewichtung auswirkt.

8. Term kommt in einer textinternen Zusammenfassung vor
9. Term kommt in einem Paragraphen (aber nicht an exponierter Position[4]) vor
10. Term steht am Anfang eines Paragraphen
11. Term steht am Schluß eines Paragraphen
12. Term kommt im Literaturverzeichnis vor
13. Term kommt in den Fußnoten vor

Die verwendete Gewichtungsfunktion basiert auf der Standardgewichtung wie sie in SALTON & MCGILL 1983 vorgeschlagen wird:

(Formel 1) $$w_i = tf_i \cdot idf_i$$

wobei

w_i = Gewicht von Term i
tf_i = Frequenz des Term i im Dokument
idf_i = inverse Dokumentfrequenz

Dabei ist letztere wie folgt definiert:

$$idf_i = \log \frac{N}{n} + 1$$

wobei

N = Anzahl der Dokumente in der Kollektion
n = Anzahl der Dokumente, in welchen Term i vorkommt

Zum Vergleich wird der Termfrequenz tf_i ein strukturbedingtes Termgewicht ts_i gegenübergestellt und obige Formel 1 wie folgt modifiziert:

(Formel 2) $$w_i = ts_i \cdot idf_i$$

Das Dokumentgewicht wird bei der Ähnlichkeitsbestimmung zwischen Anfrage und Dokumenten durch Addition der Gewichte der einzelnen Anfrageterme (sog. *vector dot product*) ermittelt. Die Varianten für ts_i werden in Tab. 2 zusammengefaßt.

3.3 Bewertungsmaße

Als Bewertungsmaß wurden der normalisierte *recall* und die normalisierte *precision* herangezogen (cf. BOLLMANN 1983 und SALTON & MCGILL 1983,180f.), die Maßzahlen für die Qualität der Rangreihung liefern. Beim *recall* geht man von einem idealen System aus, welches die relevanten Dokumente auf den vordersten Rängen liefert und vergleicht die ideale mit einer tatsächlichen Rangreihung. Dies geschieht, indem beide Reihungen als *recall*-Graph darstellt werden und die normalisierte Differenz zwischen den Graphen gebildet wird. Die folgende Abbildung zeigt ein Beispiel:

[4] Vgl. 10. und 11.

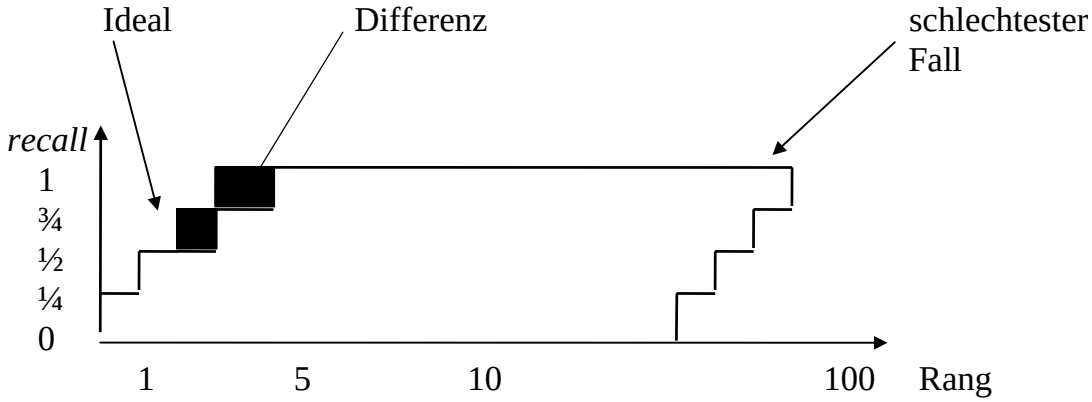


*Abb. 1: Rang-*recall*-Graph für den Idealfall und (1,2,4,6) (cf. SALTON & MCGILL 1983,181)*

Im vorliegenden Experiment mußten die Standardformeln für den normalisierten *recall* und die normalisierte *precision* modifiziert werden, da nicht immer gewährleistet werden konnte, daß der *recall* 1 ist, d. h. daß alle Dokumente gefunden wurden. Um Zufälligkeiten auszuschließen, wurde für die nicht-gefundenen Dokumente der schlechteste Fall angenommen, d. h. daß sie die letzten Rangplätze der Distribution einnehmen. Folglich ergeben sich für die Formeln folgende Modifikationen:

$R_{norm} = 1 - \frac{\sum_{i=1}^{REL} RANK_i - \sum_{i=1}^{REL} i}{REL(N - REL)}$	$\rightarrow$	$R_{norm_{wc}} = 1 - \frac{\sum_{i=1}^{FOUND} RANK_i + \sum_{i=1}^{REL-FOUND}(N-i) - \sum_{i=1}^{REL} i}{REL(N - REL)}$
$P_{norm} = 1 - \frac{\sum_{i=1}^{REL} \log RANK_i - \sum_{i=1}^{REL} \log i}{\log N!/(N - REL)!\,REL!}$	$\rightarrow$	$P_{norm_{wc}} = 1 - \frac{\sum_{i=1}^{FOUND} \log RANK_i + \sum_{i=1}^{REL-FOUND} \log(N-i) - \sum_{i=1}^{REL} \log i}{\log N!/(N-REL)!\,REL!}$ wobei *FOUND* = Anzahl der gefundenen Dokumente

Tab. 1: Formeln für normalisierten recall *und normalisierte* precision *und deren experimentbedingte Modifikationen*

Der einfachen Termfrequenz (*tf*) (cf. Formel 1) wurden im Experiment verschiedene strukturbezogene Gewichtungen gegenübergestellt, die in folgender Tabelle zusammengefaßt sind. Die Kombinationen orientieren sind zum einen an der gängigen Praxis (z. B. Titel und Abstract), zum anderen an weiterführenden Möglichkeiten, für deren Gültigkeit gewisse Hinweise vorlagen.

A1	Standard/Volltext
A2	nur Text, keine Überschriften etc.
A3	Komplement von A2
A4	nur Titel und Inhaltsverzeichnis
A5	Titel und Abstract
A6	Titel, Inhaltsverzeichnis und Abstract
A7	Titel, Inhaltsverzeichnis, Abstract und Literaturverzeichnis
A8	Gewichtung der Merkmale
A9	Optimierte Gewichtung

Tab. 2: Verschiedene Bewertungsvarianten

Für jede dieser Bewertungsvarianten wurde ein 13-dimensionaler Vektor erzeugt, wobei die einzelnen Dimensionen durch die verschiedenen Struktureigenschaften zustande kamen.

	A1	A2	A3	A4	A5	A6	A7	A8	A9
Titel	1	0	1	1	1	1	1	100	31
Überschrift	1	0	1	1	0	1	1	50	88
Unterüberschrift	1	0	1	1	0	1	1	25	7
Unterunterüberschrift	1	0	1	1	0	1	1	20	1
Bild-/Tab.-Unter-/Überschrift	1	0	1	1	0	1	1	20	3
Aufzählung/Liste	1	1	0	0	0	0	0	10	7
Abstract	1	0	0	0	1	1	1	10	238
Zusammenfassung	1	1	0	0	0	0	0	10	15
Paragraph-Mitte	1	1	0	0	0	0	0	1	9
1. Satz	1	1	0	0	0	0	0	10	8
letzter Satz	1	1	0	0	0	0	0	5	6
Literaturverzeichnis	1	0	1	0	0	0	1	5	18
Fußnoten	1	0	0	0	0	0	0	1	1

Tab. 3: Beispiel für Bewertungsvarianten

Für die Bewertungen A1-A7 gingen die mit „1“ belegten Felder und die entsprechenden Termfrequenzen in die Indexierung ein. In A8 erhielten die Struktureigenschaften heuristisch festgelegte Gewichtungsfaktoren, die mit der Termfrequenz multipliziert wurden. A9 bezieht sich auf das Ergebnis einer Optimierungsroutine, die sich an das Prinzip Genetischer Algorithmen anlehnt und das Ziel verfolgt, aus dem vorhandenen Wissen eine optimierte Verteilung der Gewichte zu entwickeln, d. h. unter Einbeziehung von bereits vorhandenen *recall*-Werten zu lernen, wie Gewichte optimal anzusetzen sind.

Jede Bewertung hat dabei 13 Erbfaktoren. Diese sind bestimmt durch den Gewichtsfaktor, der einem Strukturmerkmal zugewiesen ist. Bei der Optimierung treten jeweils 50 individuelle Gewichtungen in einer beliebigen Anzahl (Generationen) von Durchläufen bzgl. der Höhe des normalisierten *recall* gegeneinander an. Die Besten geben ihr Erbgut an die nächste Generation weiter. Hier waren dies die drei erstplatzierten und als gesetzte Gewichtung A8.

Bei der Vererbung fanden verschiedene Verfahren Anwendung. Die drei erstplatzierten und die gesetzte Gewichtung setzten sich in der nächsten Generation fort. Zusätzlich kamen drei neue Gewichtungen hinzu, indem die erstplatzierten paarweise durch Mittelung der Strukturmerkmalsgewichte gekreuzt wurden. Sechs weitere entstanden durch paarweises Kreuzen der ersten drei nach dem Verfahren Individuum A vererbt Faktor 1, Individuum B Faktor 2, Individuum A Faktor 3 usw. und umgekehrt. Die nächsten 12 wurden durch zufällige Mutation eines zufällig ausgewählten Erbfaktors aus den bereits erzeugten ersten 12 Gewichtungen gewonnen. Nr. 25 war gesetzt. Und schließlich ließen sich noch weitere 25 Gewichtungen durch zufällige Mutation einer zufälligen Anzahl von Erbfaktoren, welche wiederum zufällig ausgewählt

wurden, aus den bereits erzeugten ersten 25 generieren. Nach 50 Generation ergab sich als optimierte Gewichtung A9.

3.4 Ergebnisse

Die folgende Tabelle enthält für alle Bewertungen A1-A9 die gemittelten Werte. Zusätzlich zu normalisiertem *recall* und normalisierter *precision* werden *recall* und *precision* angegeben, um eine Vorstellung zu vermitteln, wie sich die Antwortmengen zusammensetzen:

		R_{norm}	P_{norm}	R	P
A1	Standard/Volltext	0.866	0.748	0.934	0.072
A2	nur Text, keine Überschriften etc.	0.860	0.747	0.929	0.083
A3	Komplement von A2	0.715	0.597	0.799	0.172
A4	nur Titel und Inhaltsverzeichnis	0.423	0.381	0.442	0.419
A5	Titel und Abstract	0.591	0.545	0.614	0.341
A6	Titel, Inhaltsverzeichnis und Abstract	0.629	0.560	0.658	0.318
A7	Titel, Inhaltsverzeichnis, Abstract und Literaturverzeichnis	0.782	0.661	0.845	0.152
A8	Gewichtung der Merkmale	0.857	0.724	0.934	0.072
A9	Optimierte Gewichtung	0.872	0.754	0.934	0.072

Tab. 4: (Normalisierte) recall- *und* precision-*Werte*

Die Ergebnisse wurden einer Kontrollgruppe B gegenübergestellt, welche als Referenz für die 13 einzelnen Merkmale diente. Hier ging jeweils nur ein Strukturmerkmal ein.

		R_{norm}	P_{norm}	R	P
B1	Titel/Untertitel	0.249	0.265	0.251	0.511
B2	Hauptüberschrift	0.253	0.262	0.257	0.457
B3	Unterüberschrift	0.071	0.090	0.071	0.320
B4	Unterunterüberschrift	0	0	0	0
B5	Bild-/Tab.-Unter-/Überschrift	0.074	0.086	0.075	0.201
B6	Inhaltsverzeichnis	0.303	0.302	0.312	0.198
B7	Abstract	0.572	0.533	0.592	0.339
B8	textinterne Zusammenfassung	0.117	0.139	0.118	0.208
B9	Paragraph (nicht-exponierte Position)	0.835	0.716	0.907	0.096
B10	Paragraph (1.Satz)	0.750	0.648	0.796	0.151
B11	Paragraph (letzter Satz)	0.716	0.594	0.768	0.157
B12	Literaturverzeichnis	0.635	0.552	0.689	0.161
B13	Fußnoten	0.426	0.364	0.454	0.129

Tab. 5: Kontrollgruppe mit einzelnen Strukturmerkmalen

Hinzu kamen diverse Zusatztests, die sich z. B. auf die Verteilung der einzelnen Merkmale innerhalb der Dokumente und die Termfrequenzen innerhalb der Strukturmerkmale bezogen (cf. ZETTEL 1996).

Neben der Tatsache, daß die einzelnen Ergebnisse oft ziemlich nah beieinander liegen, läßt sich für die Gruppe A feststellen, daß die Volltextindexierung einer Teiltextindexierung vorzuziehen ist, falls der Volltext vorliegt. Das beste Ergebnis bei ei-

ner Teiltextindexierung liefert der reine Text (A2), der kaum abfällt. Das ist ein starkes Indiz dafür, daß der Hauptinformationsträger doch der Text ist.

Das Ergebnis für A9 zeigt, daß durch Ausnutzung der Dokumentstruktur Effektivitätssteigerungen erreicht werden können. Die Rangreihenfolge der durch A9 generierten Distribution wies geringfügig höhere *recall*- und *precision*-Werte (in normalisierter Form) auf als die, welche für die normale Volltextindexierung A1 generiert wurde.

Steht der Volltext nicht für die Indexierung zur Verfügung, kann die Ergänzung der gängigen Kombinationsform „Titel und Abstract" um das Inhaltsverzeichnis und/oder das Literaturverzeichnis eine wesentliche Effektivitätssteigerung erbringen.

Betrachtet man die Werte der Gruppe B, so fällt vor allem auf, daß die Bewertungen B9, B10 und B11 recht gute Ergebnisse liefern. Dies liegt vor allem daran, daß diese den anteilsmäßig größten Textumfang haben, und somit auch am meisten Information enthalten dürften.

Interessant ist auch das relativ gute Abschneiden des Literaturverzeichnisses (B12). Es ist immerhin das viertbeste Einzelmerkmal und übertrifft damit die Werte des Merkmals „Abstract". Im Literaturverzeichnis dürften vor allem die Titel der angegeben Literaturstellen einen Einfluß haben. Beachtenswert ist auch das schlechte Abschneiden der Tabellen- und Bild-Über- bzw. Unterschriften. Gerade Abbildungen und Tabellen erwiesen sich in den empirischen Untersuchungen als Orte höchster Informationskonzentration. Dabei darf jedoch nicht vergessen werden, daß die Information wohl eher *in* der Tabelle oder der Abbildung steckt, als im erklärenden Text darüber oder darunter. Hier sind es wohl eher Informationsextraktionsverfahren, die sich als erfolgversprechend erweisen könnten.

Die Werte für die Abstract-Indexierung in den mittleren *recall*- und *precision*-Bereichen zeigen, daß durch die alleinige Indexierung dieses Elements starke Verluste in Kauf genommen werden müssen.

4 Zusammenfassung und Ausblick

Nimmt man die Ergebnisse von PAIJMANS 1994 und die hier erzielten zusammen, so muß man feststellen, daß die jeweils durchgeführten Experimente keinen Nachweis dafür liefern, daß sich durch Ausnutzung der Strukturinformation eine wesentliche Effektivitätssteigerung beim IR erzielen ließe. Dennoch weisen die Ergebnisse tendenziell darauf hin, daß die Dokumentstruktur sehr wohl ein wichtiger Informationsträger ist, dieser Aspekt aber bei der hier erreichten Analysetiefe auf der Ebene der Wortverteilungsstatistiken zu keinen substantiellen Verbesserungen führt.

Die anhand einer relativ kleinen Menge sozialwissenschaftlicher Texte gewonnenen Erkenntnisse müssen auf andere Textsorten, d. h. auch auf andere, textsortenspezifische Gewichtungsschemata, übertragen und dort erprobt werden. Problematisch ist, daß Volltexte nur in sehr geringem Umfang mit ihrem vollem Layout frei zur Verfügung stehen und man sich auch aus diesem, eher pragmatischen Grund mit suboptimalen Varianten zu beschäftigen hat.

Gregor Büchel, Fachhochschule Köln

Generierung von semantischen Netzen für Schlagwörter des Katalogbestandes einer Hochschulbibliothek

1 Einleitung

Für den Aufbau eines Datenbanksystems zum Online-Zugriff auf die Katalogbestände der Fachhochschulbibliothek Köln unter dem Internetdienst Word Wide Web (WWW) (LIER 1996) wurden die maschinenlesbaren Katalogdaten der Bibliothek auf ihre Abbildbarkeit in ein relationales Datenbanksystem untersucht (*http://wwwopac.fh-koeln.de:8080/opac_such.html/*). Besondere Probleme warfen die Schlagwortbestände des Kataloges für die Definition geeigneter relationaler Datenstrukturen auf, um insbesondere eine Suchmöglichkeit anzubieten, die den potentiellen WWW-Klienten mehr Komfort als eine bloße Freitextsuche bietet. Als ein erstes Ergebnis wurde ein Programmsystem zur maschinellen Generierung normierter Schlagwortbestände entwikkelt (HUCK 1996), das mit Methoden relationaler Datenbanktechnik operiert.

Auf diesem Ergebnis aufbauend soll nun ein Verfahren beschrieben werden, um aus dem vorliegenden normierten Schlagwortbestand und aus Relationen, die zwischen Entitäten des Katalogbestandes gegeben sind, maschinell ein sogenanntes semantisches Netz von Schlagwörtern zu generieren. In der Beschreibung des semantischen Netzes wird auf Methoden von SCHANK (SCHANK 1975) und NORMAN & RUMELHARDT (NORMAN & RUMELHARDT 1975) zurückgegriffen, die mittels geeignet gefärbter Graphen semantische Relationen untersuchten. Diese Methoden konnten bereits produktiv für die Beschreibung datenbankgestützter Verfahren zur Analyse definierender Phrasen, die in Textumgebungen bestimmter Verben (wie z. B. „heißen“, „nennen“, „bezeichnen“) auftreten (LENDERS 1985), verwendet werden (BÜCHEL 1995). Während diese Phrasen von bestimmten Verben „regiert“ werden, sind Phrasen in Schlagwortketten in der Hauptsache Nominalphrasen, die, quantitativ betrachtet, u. a. durch Abwesenheit prädikativ gebrauchter Verbformen gekennzeichnet sind. Das Ergebnis des nachfolgend beschriebenen Verfahrens ist ein semantisches Netz von Schlagwörtern, die untereinander durch Kanten verbunden sind, die aus maschinell bestimmbaren semantischen Ähnlichkeitsrelationen ermittelt werden. Lokale Teilgra-

phen dieses semantischen Netzes sollen künftig als Suchhilfen für WWW-Klienten des Online-Katalogsystems genützt werden können.

2 Ein datenbankgestützter Algorithmus zur Isolierung und maschinellen Generierung von normierten Schlagwortphrasen

Das bereits entwickelte Programmsystem zur maschinellen Generierung normierter Schlagwortbestände realisiert unter Verwendung von Methoden der relationalen Datenbanktechnik folgenden Algorithmus, der aus drei Arbeitsschritten besteht: 1. Segmentierung von Schlagwortketten, 2. Erzeugung von normierten Schlagwortphrasen, 3. Verdichtung der Hauptschlagwörter (Token). Zur Beschreibung des Verfahrens dient folgender Programmablaufplan. Der Algorithmus ist als *ESQL/C*-Programmsystem unter Verwendung eines Standard-RDBMS auf einem vernetzten UNIX-Server realisiert (HUCK 1996).

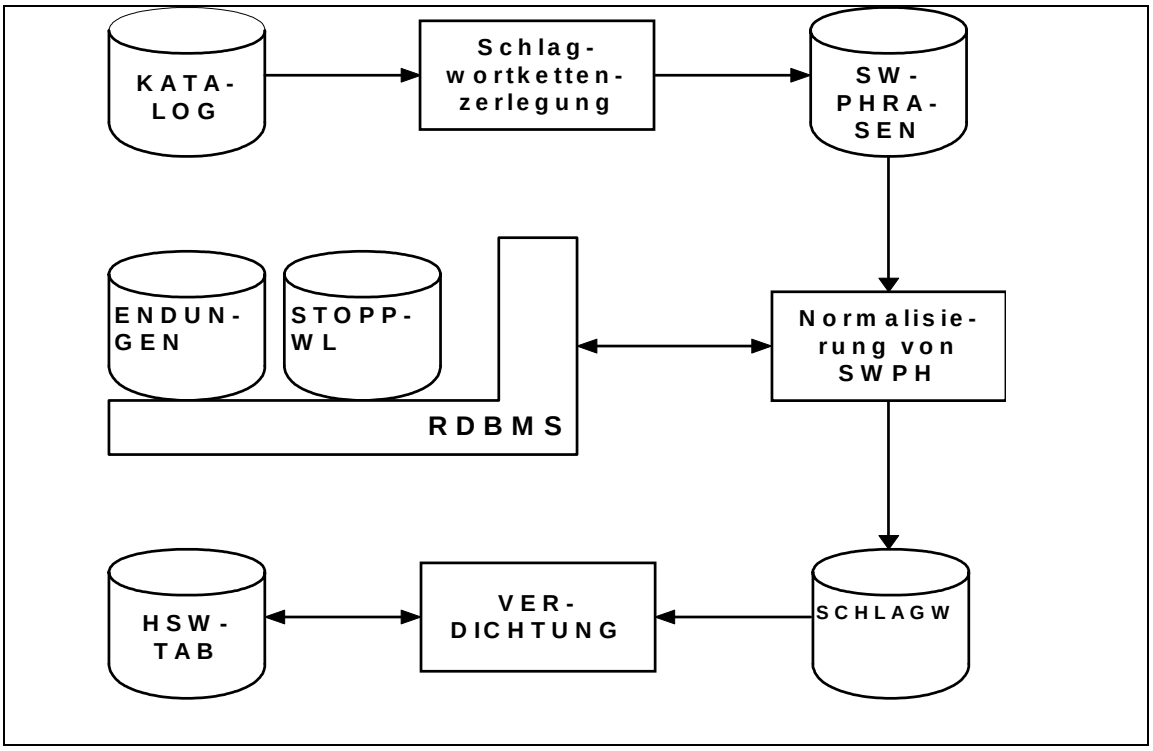


Abb. 1: Programmablaufplan:

2.1 Segmentierung von Schlagwortketten

Eine Schlagwortkette (SWK) kann aus einer oder mehreren Schlagwortphrasen (SWPH) bestehen. Sind es mehrere Schlagwortphrasen, so sind sie durch je ein Sonderzeichen getrennt. Im maschinellen Austauschformat für Bibliotheken (MAB-Format), in dem die elektronisch lesbaren Katalogdaten der Fachhochschulbibliothek vorliegen, wird dafür das ‘/’-Zeichen verwandt. Mit einem Zeichenkettenzerlegungsprogramm werden mehrphrasige Schlagwortketten zerlegt. Nach der Zerlegung werden alle Schlagwortphrasen in eine Datenbanktabelle (SWPHRASEN) geschrieben. Folgende Daten wurden für den untersuchten Katalogbestand ermittelt:

Nr.	***Anzahl***	***Entitäten***
(1)	158.300	untersuchte Titel des Kataloges
(2)	48.438	Titel von (1) mit mindestens einer SWK
(3)	58.549	Anzahl vorhandener SWK
(4)	139.855	Anzahl erhaltener SWPH

Anmerkungen:

Zu (2): Die Bibliothek der Fachhochschule ist keine Einrichtung, die eine Verschlagwortung durchführt. Für Titel, für die eine Verschlagwortung bei der Datenübergabe durch das Hochschulbibliothekszentrum NRW vorliegt, wird diese übernommen, etwas mehr als 30 Prozent der Titel aus (1) sind verschlagwortet.

Zu (3): Ein verschlagworteter Titel kann mehrere Schlagwortketten haben. Im arithmetischen Mittel liegen ca. 1,2 SWK pro Titel von (2) vor.

Zu (4): Pro Titel mit Schlagwortkette liegen im Mittel 2,88 Schlagwortphrasen vor.

2.2 Erzeugung von normierten Schlagwortphrasen

In einem ersten Arbeitsgang werden sämtliche Wortformen aller Schlagwortphrasen (SWPH) mit einer Stoppwortliste abgeglichen, die eine komparativ große Menge von Funktionswörtern der deutschen Sprache enthält (Präpositionen, Konjunktionen, Artikel, Numeralia usw.; insgesamt sind ca. 1400 Wortformen in der Stoppwortliste enthalten, die Stoppwortliste ist als Tabelle des relationalen Datenbanksystems gespeichert). Unter Verwendung der Stoppwortliste, unter Berücksichtigung beobachteter syntaktischer Muster der Schlagwortphrasen und unter Nutzung der in den Phrasen vorhandenen Groß-/Kleinschreibung konnte folgende Klasseneinteilung vorgenommen werden (HUCK 1996):

Klasse	***Phrasenmuster***	***Prozentsatz aller SWPH***
1	Phrase mit nur einem Wort, z. B. „Recht“	87.5
2	Zwei Wörter, z. B. „Harmonische Analyse“	10.1
3	Zwei Wörter durch Stoppwort verbunden, z. B. „Information und Dokumentation“	0.4
4	Zwei Wörter mit UND/ODER verbunden, Wortendung des ersten Wortes zu ergänzen, z. B. „Vor- und Frühgeschichte“	0.2
5	Personennamen: Syntax: <Nachname>,<Vorname1>[...<VornameN>], z. B. „Kant, Immanuel“	1.1
6	Drei Wörter ohne Kommata und ohne Stoppwörter, z. B. „Bund Deutscher Mietervereine“	0.1
7	Mehrere Wörter, keine Kommata, keine zu ergänzenden Wortendungen, z. B. „Regeln für die alphabetische Katalogisierung“	0.3
X	Alle anderen Fälle. Hierzu gehören z. B. „Vermeer van Delft, Jan“	0.3

Für Wortformen der Schlagwortphrasen aus den Klassen 1,2,3,4,5 und 7 konnte ein maschinelles Verfahren zur Generierung normierter Schlagwortphrasen (NSWPH) realisiert werden. Phrasen der Klasse 6 und X werden für eine nachfolgende manuelle Bearbeitung im Dialogbetrieb in eine besondere Datei geschrieben. Je nach erkanntem Klassenmuster erstellt das Programm aus der gegebenen Schlagwortphrase einen oder mehrere Einträge in die Zieltabelle SCHLAGW. Jeder Eintrag besteht aus einem sog. Hauptschlagwort (HSW) und der normierten Schlagwortphrase. Für Phrasen der maschinell verarbeiteten Klassen sind nachfolgend Beispiele angegeben:

Klasse	***Datensatz in SCHLAGW***	***SWPH***	***HSW***	***NSWPH***
1	1/1	Datenbanksystem	Datenbanksystem	Datenbanksystem
2	1/1	Logische Programmierung	Programmierung	Programmierung, Logische
3	1/2	Entscheidung bei Unsicherheit	Entscheidung	Entscheidung bei Unsicherheit
	2/2		Unsicherheit	Entscheidung bei Unsicherheit
4	1/2	Vor- und Frühgeschichte	Vorgeschichte	Vor- und Frühgeschichte
	2/2		Frühgeschichte	Vor- und Frühgeschichte
5	1/1	Kant, Immanuel	Kant	Kant, Immanuel
7	1/2	Differentialgleichung mit nacheilendem Argument	Differentialgleichung	Differentialgleichung mit nacheilendem Argument
	2/2		Argument	Differentialgleichung mit nacheilendem Argument

Für Schlagwortphrasen der Klasse 4 ist anzumerken, daß die unvollständigen Wortformen (die sog. Bindestrichwörter) nach einer maschinellen Endungsanalyse mit der fehlenden Wortendung ergänzt werden.

2.3 Verdichtung der Hauptschlagwörter (Token)

In der Zieltabelle SCHLAGW des Arbeitsschrittes 2.2 ist für jedes Token eines Paares (Hauptschlagwort, normiertes Schlagwortphrase) (=:(HSW, NSWPH)) ein Eintrag vorhanden. Mit einem auf Zeichenkettenvergleich beruhenden Verdichtungsprogramm wurde eine Tabelle der Types der Hauptschlagwörter erzeugt (HSW-Types). Folgende Tabelle gibt eine Übersicht über das Type-Token-Verhältnis der Hauptschlagwörter im Katalogbestand der Fachhochschulbibliothek Köln:

(1)	Token der Paare (HSW, NSWPH)	132.789
(2)	HSW-Types	16.103
(3)	Type-Token-Verhältnis	1/8,25

3 Definition eines semantischen Netzes für normierte Schlagwortphrasen eines Katalogbestandes

Die Erweiterung des WWW-OPAC-Systems für die Katalogbestände der Fachhochschulbibliothek Köln um den oben beschriebenen Algorithmus erzeugt eine gegenüber dem bisherigen System erweiterte Datenbank, die es gestattet, alle Relationen des nachfolgenden *entity relationship*-Modells (ER-Modell) mittels SQL-Abfragen zu recherchieren. Entitäten des Modells sind Fachgebiete, Hauptschlagwörter und Titel.

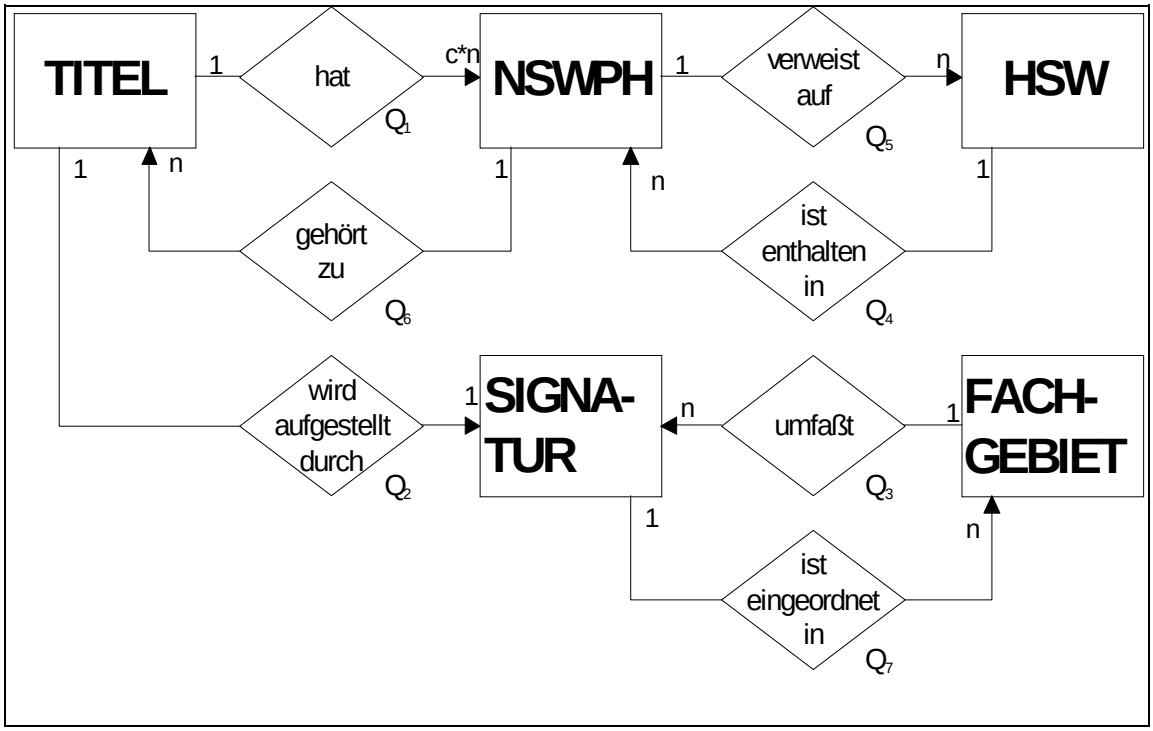


Abb.2: Entity-Relationship-Modell:

Jeder in der Bibliothek aufgestellte Titel (Titel mit Beständen) verfügt über eine Signatur. Die Signatur enthält Angaben über den Standort, über das Fachgebiet, dem der Titel zugeordnet ist und eine laufende Nummer des Titels innerhalb des Fachgebietes. Die Klassifikation der Fachgebiete folgt der Systematik des Hochschulbibliothekszentrum des Landes NRW, in der Fachgebiete mit einem dreistelligen Buchstabenschlüssel versehen sind. Das Fachgebiet „Datenbanken“ als Teil des Fachbereiches „Digitalrechner“ (TWG-TYI) ist z. B. mit dem Schlüssel TWY versehen. Das Fachgebiet „Fourieranalysis“ hat z. B. den Schlüssel TIE. Die Systematik des Hochschulbibliothekszentrum NRW (die sog. GHB-Aufstellungssystematik) kann unter WWW eingesehen werden (*http://www.hbz-nrw.de/hbz/ghb-sys/*).

Ein semantisches Netz, das als Suchhilfe für inhaltlich verwandte Schlagwörter dienen soll, wird unter Verwendung des obigen *entity relationship*-Modells durch folgenden Graphen beschrieben: Die Knoten des Graphen sind je nach Knotentyp in vier übereinanderliegenden Schichten angeordnet. Knotentypen des Graphen sind Fachgebiete (Schicht 1), Hauptschlagwörter (Schicht 2), normierte Schlagwortphrasen (Schicht 3) und Titel (Schicht 4). In folgender Abbildung ist ein Teilgraph des semantischen Netzes, der lokal durch die Eingabe eines Hauptschlagwortes (hier HSW =

„Datenbanken“) bestimmt ist, in Auszügen abgebildet. Zur Illustration sind darin die Knoten der Schichten 1, 2 exemplarisch angegeben, die Knoten der Schichten 3, 4 sind bloß durch formale Bezeichner dargestellt. Die Relationen R1, i1, g1, ä1, ä2, ä3, die die Kanten des Netzes einfärben, sind nach der Abbildung des Netzes beschrieben.

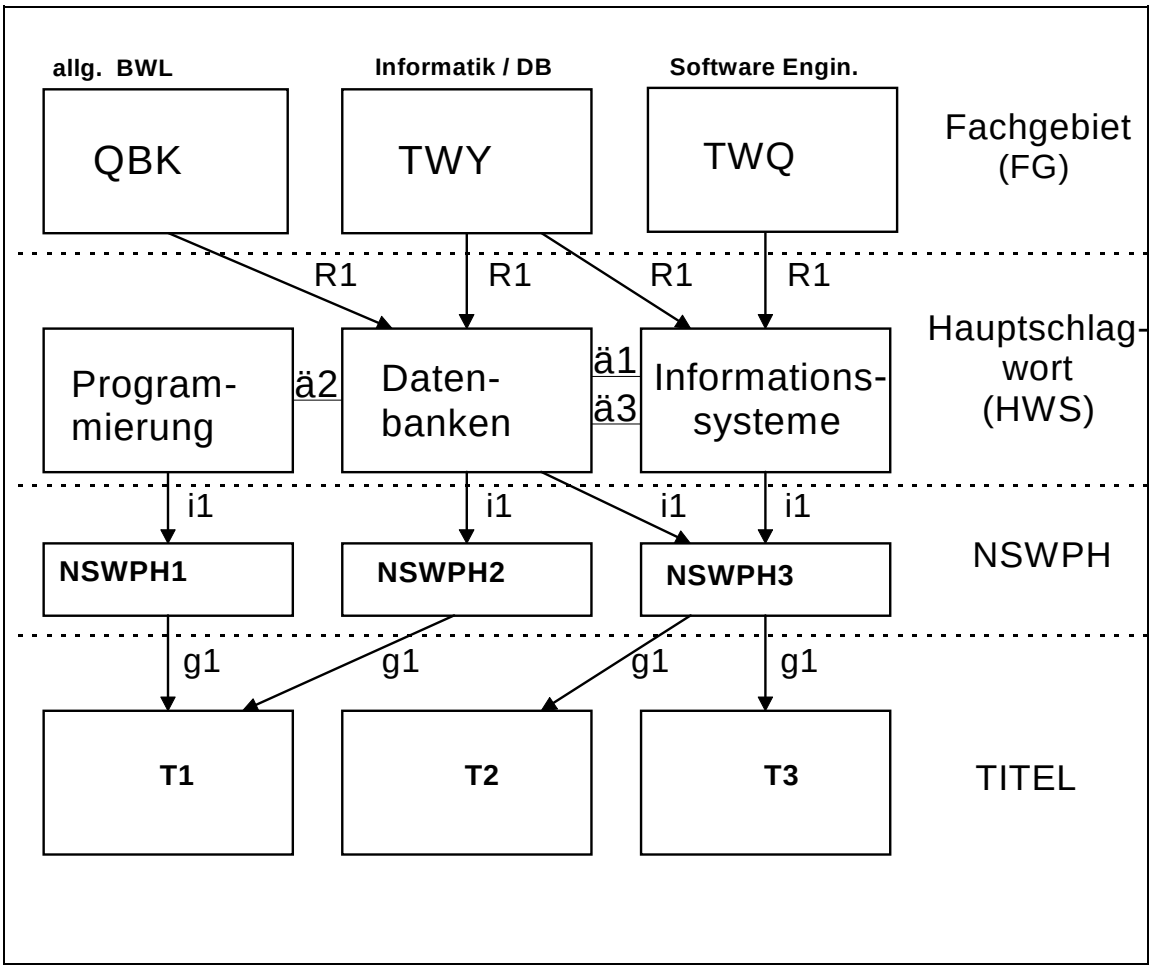


Abb. 3: Semantisches Netz von Schlagwortbeständen:

Beschreibung der Relationen:

1.Typ R1: R1(FG,HSW): Eine Kante vom Typ R1 verbindet einen Knoten FG (Fachgebiet) mit einem Knoten HSW (Hauptschlagwort) genau dann, wenn das HSW innerhalb von Titeln des Fachgebiets FG belegt ist.

2.Typ i1: i1(HSW,NSWPH): Eine Kante vom Typ i1 verbindet einen Knoten HSW mit einem Knoten NSWPH genau dann, wenn das HSW in der normierten Schlagwortphrase NSWPH enthalten ist.

3.Typ g1: g1(NSWPH,TITEL): Eine Kante vom Typ g1 verbindet einen Knoten NSWPH mit einem Knoten TITEL genau dann, wenn die normierte Schlagwortphrase NSWPH als Phrase in einer Schlagwortkette des Buches TITEL belegt ist.

4.Typ ä1: ä1(HSW1,HSW2): Eine Kante vom Typ ä1 verbindet zwei Knoten HSW1 und HSW2 genau dann miteinander, wenn es mindestens ein Fachgebiet FG als gemeinsamen R1-Parent-Knoten von HSW1 und HSW2 gibt.

5.Typ ä2: ä2(HSW1,HSW2): Eine Kante vom Typ ä2 verbindet zwei Knoten HSW1 und HSW2 genau dann miteinander, wenn es mindestens einen Titel des Katalogbestandes gibt, der in seinen Schlagwortketten mindestens je einen Beleg für eine normierte Schlagwortphrase NSWPH1 und NSWPH2 enthält, wobei NSWPH1 (NSWPH2) das Hauptschlagwort HSW1 (HSW2) als ein i1-Parent-Knoten hat.

6.Typ ä3: ä3(HSW1,HSW2): Eine Kante vom Typ ä3 verbindet zwei Knoten HSW1 und HSW2 genau dann miteinander, wenn es mindestens eine normierte Schlagwortphrase NSWPH0 gibt, so daß die beiden Relationentupel i1(HSW1,NSWPH0) **und** i1(HSW2, NSWPH0) belegt sind.

4 Beispiele lokaler Teilgraphen des gegebenen semantischen Netzes

Die Wirkungsweise des gegebenen semantischen Netzes als Suchhilfe für Klienten des Online-Bibliothekssystems soll anhand folgender Recherchen nach Titeln des Kataloges erläutert werden:

4.1 Recherche zum Stichwort: „Parallelverarbeitung“ (kleineres Beispiel):

4.1.1: Die Suchanfrage führt innerhalb des semantischen Netzes auf das Hauptschlagwort „Parallelverarbeitung“ (HSW1). HSW1 besitzt nur eine normierte Schlagwortphrase innerhalb des Netzes. Die Auswertung der R1- und g1-Kanten führt auf 10 Fachgebiete, in denen insgesamt 25 Titel zum gegebenen Hauptschlagwort belegt sind. Die Fachgebiete gehören den drei Fachgebietsgruppen „Theoretische Informatik“ (TVA-TVV, 4 Titel), „Digitalrechner“ (TWG-TYI, 12 Titel) und „Anwendung informationsverarbeitender Maschinen“ (TZA-TZX, 9 Titel) an.

4.1.2: Die Beschränkung der Auswertung der Ähnlichkeitsrelation ä1 auf das Fachgebiet TXF („Sonstiges zur Programmierung“ von Digitalrechnern), in dem die meisten Titel zum gegebenen Schlagwort HSW1 belegt sind (ä1(HSW1,-HSW2)∩R1(„TXF“,HSW2)), liefert 30 bezüglich der Ähnlichkeitsrelation **ä1** verwandte Schlagwörter HSW2. Darunter sind folgende HSW2 in mehr als einem Titel belegt: {„Occam“, „Parallelrechner“, „Programmierung“, „Programmiersprache“, „Transputer“}. Sämtliche dieser gefundenen Schlagwörter sind sachlich aufs Engste mit dem Suchbegriff „Parallelverarbeitung“ verwandt.

4.1.3: Die Auswertung der Ähnlichkeitsrelation **ä2** führt auf weitere Hauptschlagwörter HSW2, die mit HSW1 inhaltlich verwandt sind (ä2(HSW1,HSW2)): Im folgenden sind Beispiele für solche Schlagwörter HSW2 genannt, die **nicht** in 4.1.2 belegt sind: „Automatentheorie“, „Computerarchitektur“, „Mehrprozessorsystem“, „Petri-Netz“, „Sprache“ (letzteres HSW tritt mit großer Streuung auf, belegt ist in Bezug auf HSW1 die NSWPH „Sprache, formale“).

4.2 Recherche zum Stichwort: „Datenbanksystem“ (größeres Beispiel):

4.2.1: Die Suchanfrage führt innerhalb des semantischen Netzes auf das Hauptschlagwort „Datenbanksystem“ (HSW1). Dieses Hauptschlagwort besitzt sechs normierte Schlagwortphrasen (= belegte i1-Tupel):

Z. B.: „Datenbanksystem, deduktives“, „Datenbanksystem, objektorientiertes“, ..., „Datenbanksystem, verteiltes“. Die Auswertung der R1- und g1-Kanten führt auf 94 Titeln in 14 Fachgebieten. Bei den Fachgebieten tritt eine deutlich größere Streuung als 4.1.1 auf: Z. B. Fachgebiete der Fachgebietsgruppen „Allgemeine Betriebswirtschaftslehre“ (PZA-QCT), „Elektrische Nachrichtentechnik“ (YCA-YGY), „Fertigungstechnik“ (ZHU-ZLF) u. a. kommen hinzu.

4.2.2: Die Beschränkung der Ähnlichkeitsrelation ä1 auf das Fachgebiet mit den meisten Belegtiteln von HSW1 (ä1(HSW1,HSW2)∩R1(„TWY“,HSW2), TWY: „Datenbanken“) liefert 122 Schlagwörter HSW2, darunter 19, die in mehr als vier Titeln belegt sind. Z. B.: „Befehlsvorrat“, „Clipper“, „Datenbank“, „Informationssystem“, „Online-Recherche“, „SQL“. Man erkennt, daß diese Schlagwörter sachlich sehr eng mit dem Suchbegriff „Datenbanksystem“ verwandt sind.

4.2.3: Die Auswertung der Ähnlichkeitsrelation ä2 führt auf weitere Hauptschlagwörter HSW2, die mit HSW1 verwandt sind (ä2(HSW1,HSW2)): Beispiele für Schlagwörter HSW2, die nicht in 4.2.2 belegt sind: „CIM“, „Fuzzy-Logik“, „Schnittstelle“, „VLSI“. Man kann erkennen, daß diese Schlagwörter in einem entfernteren, meist über besondere Anwendungssysteme verbundenen, sachlichen Bezug zum Suchbegriff „Datenbanksystem“ stehen.

5 Schlußbetrachtungen und Perspektiven

Sämtliche Knoten des semantischen Netzes sind bereits als Tabellenelemente des im vorigen Kapitel beschriebenen Datenbanksystems gespeichert. Die Kanten können maschinell aus den *relationships*, die durch das obige ER-Modell gegeben sind, bestimmt werden. Z. B. wird eine Kante des Typs R1 durch eine Produktionsregel, die eine Hintereinanderschaltung der Relationen Q7, Q2, Q6, Q4 enthält, bestimmt. Entsprechende *ESQL/C*-Programme sind in Vorbereitung.

Zu den Kantentypen des semantischen Netzes ist anzumerken, daß die Typen R1, i1, g1 eine inhaltliche Verwandtschaft von Knoten beschreiben, die durch eine Hierarchie von Begriffen gebildet wird. Diese Verwandtschaften sind damit, als semantische Relationen betrachtet, Hyperonymien. Die Kantentypen ä1, ä2, ä3 stellen semantische Ähnlichkeitsrelationen in der horizontalen Struktur des semantischen Netzes dar. Die Größe eines anzuzeigenden Teilgraphen soll der Benutzer durch Parametrisierung der Bindungsstärken der Ähnlichkeitsrelationen gestalten können. (Es sollen „mehr“ „stärker gebundene“ Knoten als „schwächer gebundene“ Knoten angezeigt werden.).

Allgemein stehen dabei die Bindungsstärken der Ähnlichkeitsrelationen ä1, ä2, ä3 in folgender Beziehung:

$ä_2 << ä_1$: „Ähnlichkeit der Sachgebiete“ bindet stärker als „Ähnlichkeit in Katalogbelegen“.

$ä_2 << ä_3$: „Ähnlichkeit in normierten Schlagwortphrasen“ bindet stärker als „Ähnlichkeit in Katalogbelegen“.

Das dargestellte Verfahren zur Generierung von semantischen Netzen ist operationalisierbar auf Katalogbeständen des quantitativen Umfanges und des typischen Dateninventars von Hochschulbibliotheken. Besondere Bedingungen im Datenbestand für WWW-Modelle mit Ähnlichkeitsrelationen, die bei speziellen Konfigurationen für Seminarbibliotheken mit kleineren Beständen vorliegen (vgl. z. B. (BAPTIST 1996)), werden hier nicht gemacht. Weiterhin gestattet die obige Beschreibung des semantischen Netzes, die Abbildung von netzwerkartigen Graphen und nicht nur die Abbildung von Baumgraphen (vgl. z. B. DÄßLER 1996).

Michael Malburg, DFKI Kaiserslautern

Einsatz von Tagging-Verfahren zur Verbesserung der Texterkennung

1 Texterkennung und maschinelle Sprachverarbeitung?

Bei der Dokumentanalyse handelt es sich um ein Forschungsgebiet, das über seine Anwendung, also die Analyse gedruckter Dokumente, als ein Konglomerat verschiedener anderer Forschungsgebiete definiert ist. Derart fließen, neben grundlegenden Techniken aus der Informatik, v. a. Erkenntnisse der Bildverarbeitung, der natürlichsprachlichen Analyse und der Kognitionspsychologie in die Dokumentanalyse ein. Allgemeiner gesprochen liegt die Dokumentanalyse am Schnittgebiet zwischen Informatik, maschineller Sprachverarbeitung und Psychologie, wobei letztere oft erst bei genauerem Hinsehen als wirklich relevant angesehen wird. In diesem Papier wird von Arbeiten berichtet, die das Grenzgebiet zwischen informatikorientierter Dokumentanalyse und quantitativer Linguistik betreffen.

Allgemein dürfte unstrittig sein, daß die Dokumentanalyse durch Methoden der statistisch motivierten Sprachverarbeitung unterstützt werden kann und soll. Wie stark der durch eine solche Synergie zustandekommende Gewinn bereits in dem sehr technisch anmutenden Gebiet der Texterkennung sein kann, soll im Folgenden gezeigt werden.

Texterkennung bedeutet innerhalb der Dokumentanalyse das Teilgebiet Einzelzeichenerkennung (nicht nur im angelsächsischen Sprachraum mit OCR für *Optical Character Recognition* abgekürzt) und die nachgeschaltete Worterkennung.

Derzeitige kommerzielle Texterkennungskomponenten erreichen Erkennungsraten von 94% bis 98% auf Zeichenebene, was einer Fehlerrate von etwa 30% auf Wortebene entspricht; d. h. in etwa jedem dritten „erkannten“ Wort ist mit einem Erkennungsfehler zu rechnen. Daß dies von einer weitgehend naiven Texterkennung auch nicht anders zu erwarten ist, soll an zwei Beispielen erläutert werden:

In einer Antiquaschrift mit Serifen (etwa im vorliegenden *Times*) ist die Buchstabenfolge „rn“ (also „r“ + „n“) oft schwer vom Buchstaben „m“ zu unterscheiden. Bei einem Schriftbild wie „gern“ ist leicht mittels eines Lexikons entscheidbar, daß die Hypothese „GEM“ (der Lesbarkeit wegen in Großbuchstaben) ausscheidet – es sei denn, es gibt z. B. einen entsprechenden Eigennamen. Dies funktioniert nicht mehr bei

einem Schriftbild wie „Modern“, da sowohl die Hypothese „MODEM“ als auch „MODERN“ vom Lexikon unterstützt werden. Solche Ambiguitäten lassen sich frühestens im Wortkontext „beheben“.

In aktuellen Texterkennern werden teilweise bereits Lexika oder statistische Verfahren auf Buchstabenebene (meist *Hidden Markov-Modelle*) zur Verbesserung der Ergebnisse eingesetzt. Jedoch scheint die resultierende Ergebnisqualität seit einigen Jahren zu stagnieren, solange nicht weitere Quellen linguistischen Wissens zur Verbesserung der Texterkennung herangezogen werden.

Seit in kommerziellen Programmen zur Texterstellung sogenannte Syntaxchecker eingesetzt werden, ist auch in der Texterkennung von ähnlichen Verfahren die Rede, ohne daß bisher ernsthafte Erfolge in dieser Hinsicht zu verzeichnen wären. In diesem Papier soll der Einsatz syntaktischer (Wort-)Tagger in diesem Bereich betrachtet werden.

Vorab wird dazu im folgenden Abschnitt das weite Gebiet der Dokumentanalyse und der hier besonders relevante Bereich der Texterkennung umrissen. Darauf aufbauend werden in Abschnitt 3 mögliche Methoden der maschinellen Sprachverarbeitung genannt, die für einen Einsatz in der Dokumentanalyse in Frage kommen. Die von uns untersuchten und verglichenen Verfahren werden dabei detailliert beschrieben. Eine Beschreibung von Implementierung, Trainings- und Testkorpus folgt in Abschnitt 4. Da es keine Standardverfahren zur Bewertung von OCR-Nachbearbeitungen gibt, wird die hier verwendete Bewertungsmethode in Abschnitt 5 vorgestellt. Die Ergebnisse selbst werden in Abschnitt 6 präsentiert und besprochen.

2 Dokumentanalyse: Anwendungen und Arbeitsgebiete

Unter dem Begriff der Dokumentanalyse faßt man alle Vorgänge der automatischen Analyse handgeschriebener, gedruckter und gezeichneter Dokumente zusammen. Teilweise wird auch die Bearbeitung elektronisch vorliegender Dokumente, sei es in Form kodierter Texte (z. B. ASCII) oder nicht-kodierter Bilder (z. B. Faxe), unter Dokumentanalyse (im Folgenden kurz DA) verstanden. Meist liegt dann aber zumindest ein Teil der Daten in strukturierter oder formatierter Form vor, z. B. Dokumente in HTML-Notation oder ASCII-Texte mit manuell formatierten Tabellen.

Nimmt man die Analyse elektronischer Texte aus, so handelt es sich bei der Dokumentanalyse also um ein Analogon zur Verarbeitung gesprochener Sprache auf dem Gebiet der geschriebenen Sprache, wobei jedoch zu beachten ist, daß nicht-sprachliche Ausdrucksmittel wie Layout und Graphik einen beachtlichen Teil des Forschungsgebiets einnehmen.

2.1 Anwendungsgebiete

Anwendungen der Dokumentanalyse finden sich in allen Bereichen, wo eine elektronische Erfassung oder Archivierung von papiergebundenen Dokumenten benötigt wird. Neben Einzelanwendungen, die sich in jedem Büro erdenken lassen, sind z. B. in Bi-

bliotheken, bei Vertreibern von Literaturdatenbanken, in Patentämtern und in Katasterämtern große Datenmengen derart zu bearbeiten. Dabei handelt es sich oft, wie die Beispiele Kataster und Patente zeigen, um zusätzlich oder überwiegend graphische Dokumente.

Eine zunehmend interessante Anwendung der DA findet sich im Bereich der Bürokommunikation, wo derzeit ein starker Trend hin zu Dokumentarchivierungssystemen sowie zu sogenannten Workflow-Management-Systemen zu beobachten ist. In diesem Umfeld, welches auch den Nährboden der vorliegend beschriebenen Arbeit bildet, ist eine Dokumentanalyse mit zwei Zielen zugleich sinnvoll: Einerseits muß das Vorgangssystem bestimmte Informationen zu eingehenden Dokumenten mit dem Ziel inhaltsadäquater Weiterleitung oder Benachrichtigung erhalten; andererseits ist eine Archivierung von Dokumenten nur hilfreich unter Mitgabe gewisser „Indizierungsinformation“, welche ein späteres Auffinden erleichtert.

2.2 Exemplarische Arbeitsweise

Um die Arbeitsweise eines DA-Systems zu veranschaulichen, wollen wir den speziellen Fall eines solchen Systems für Bürokorrespondenz genauer ausführen. Das dabei als Grundlage verwendete System ist der *OfficeMAID*-Prototyp, der am DFKI in Kaiserslautern entwickelt wurde (DENGEL et al. 1995). Viele der i. F. genannten Eigenschaften sind jedoch weitgehend allgemeingültig und daher auch auf gänzlich andere Domänen anwendbar.

Die grobe Struktur von OfficeMAID umfaßt vier Phasen, welche meist sequentiell ablaufen: Bildvorverarbeitung; Strukturerkennung und -analyse; Texterkennung; und Informationsextraktion oder inhaltliche Analyse.

Ziel der Bildverarbeitung ist die Konvertierung des eingegangenen Dokumentbildes in eine normalisierte Form; in unserem Fall ein Binärbild, welches nur schwarze und weiße Bildpunkte enthält. Im Allgemeinen kann diese Phase eine Filterung von Rauschen, Eliminierung von Fehlern oder eine Farbklassifikation beinhalten.

Die Strukturerkennung isoliert im Dokumentbild textuelle und nicht-textuelle (Graphik, Photo) Bereiche; ermittelt gleich layoutete Bereiche von Text; und segmentiert diese in Absätze, Zeilen, Wörter, Zeichen und sogenannte Zusammenhangsgebiete – das sind die schwarzen Flecken, aus denen die Schriftzeichen zusammengesetzt sind: ein „ä“ besteht also aus drei Zusammenhangsgebieten, dem „a“ und den beiden Punkten. In Erweiterung zur Strukturanalyse wird zudem noch der Textfluß ermittelt und den bedruckten Dokumentzonen werden Strukturmarken zugeordnet, welche eine funktional-logische Bedeutung widerspiegeln, etwa „Überschrift“ oder „Absender“.

In der darauffolgenden Phase der Texterkennung wird endlich eine Konversion in „elektronische Schriftzeichen“ durchgeführt. Üblicherweise werden dazu erst Merkmale der Einzelzeichen ermittelt: Breite, Höhe, Ober- und Unterlänge, Schwärzungsgrad und Projektionen auf x- bzw. y-Achse sind nur einige solcher typischen Merkmale. Anhand dieser wird eine Klassifizierung der Einzelzeichen vorgenommen, wo-

nach erstmals die kodierten Symbole des internen Zeichensatzes vorliegen. Da diese typischerweise mit großen Unsicherheiten behaftet sind, setzen an dieser Stelle verschiedene Verfahren zur Verbesserung der Qualität ein, was ja auch Thema dieses Papiers ist. Bisher sind drei solche Nachbereitungsverfahren verbreitet: die Kombination der Ergebnisse mehrerer Einzelzeichenerkenner (vgl. JÄGER 1996), die kontextuelle Zeichengewichtung mittels Hidden Markov-Modellen und die lexikalische Verifikation der Zeichenhypothesen (vgl. WEIGEL 1995). Verfahren, welche darüber hinausgehend den Kontext der Wörter untereinander beachten, werden im nächsten Abschnitt 3 beschrieben. Einen Überblick zu diesen und weiteren Verfahren, die derzeit gebräuchlich sind, haben wir in DENGEL ET AL. 1996 veröffentlicht.

Die letzte, im allgemeinen Fall eher als fakultativ anzusehende Phase der DA ist die Informationsextraktion. Hier werden in der Regel stark domänenbezogene Verfahren eingesetzt, welche gewisse gewünschte Informationen aus dem Dokument extrahieren. Im Fall des `OfficeMAID`-Prototypen wird eine sogenannte Nachrichtentyp-Klassifikation durchgeführt (Angebot, Rechnung, Lieferschein etc., vgl. WENZEL ET AL. 1996); Absender und Empfänger werden ermittelt; je nach Nachrichtentyp können zudem Artikelbezeichnungen, Preise u. s. w. extrahiert werden.

2.3 Dokumentanalyse und maschinelle Sprachverarbeitung

Der Schwerpunkt der Dokumentanalyse liegt meist, wenn auch nicht immer, im bildorientierten Bereich, weshalb Aspekte der inhaltlichen Analyse der Dokumente in vielen Forschungsprojekten nicht oder nur wenig betrachtet werden. Dies hat zum einen historische Gründe, weshalb Dokumentanalyse manchmal als (fast) synonym zu Texterkennung verstanden wird. Zum anderen ergibt sich dies aus der sinnvollen Abgrenzung zu entsprechenden Teilen der maschinellen Sprachverarbeitung (i. F. kurz MS). Durch die zunehmende Anreicherung von Dokumentanalyse-Projekten und Prototypen durch Inhalte bzw. Komponenten der inhaltlichen Analyse kommt es somit zu einer leichten Verwischung der Grenzen zwischen inhaltlicher Analyse in der DA und innerhalb der MS.

Jedoch grenzen einige wenige Merkmale die beiden Bereiche klar ab. Seitens der Dokumentanalyse interessieren auch – und das nicht nur am Rande – nicht-sprachliche Äußerungselemente wie Layout und Graphik. Dieser Aspekt, der teilweise die im Vergleich zur gesprochenen Sprache fehlende Prosodie u. Ä. abdeckt, wird bisher in der MS weitgehend ignoriert. Hierdurch ergibt sich in der DA eine zunehmend nichtlineare, weil zwei- oder mehrdimensional orientierte Sichtweise der (sprachlichen) Äußerungen, welche besonders deutlich bei der Betrachtung von nur ganzheitlich interpretierbaren Reklamedokumenten zutage tritt (welche zugegebenermaßen – wohl wegen ihrer hohen Komplexität – bisher wenig von der DA betrachtet wurden). Ein weiterer, gradueller Unterschied ist die Betrachtung bzw. Nichtbetrachtung von hohen Hypothesenzahlen auf Wortebene. Die Anzahl von Hypothesen nach Bearbeitung durch eine Erkennungsmaschine – was auch eine Spracherkennungskomponente sein

kann – ist typischerweise höher als die lediglich durch lexikalische Ambiguitäten bedingte Alternativenzahl bei der Bearbeitung elektronischer Texte. Dies ist hinsichtlich der technischen Umsetzung von Methoden mitunter sehr wichtig.

3 Verfahren zur Verbesserung der Worterkennung

In Abschnitt 2.2 wurden bereits die Hauptkomponenten der Texterkennung angesprochen: Einzelzeichenerkennung und Worterkennung. Für den Rest dieses Papiers gehen wir von einer Texterkennung aus, welche nach der Einzelzeichenerkennung eine lexikonbasierte Worterkennung durchführt. Dies bedeutet, daß als Ergebnis der Texterkennung überwiegend bekannte Wörter vorliegen. Die wenigen unbekannten Wörter, die trotzdem geliefert werden, betreffen i. d. R. nicht-alphabetische Wörter, also reelle Zahlen, Telefonnummern etc., und solche Wortpositionen, für welche die Worterkennung kein passendes Wort im Wörterbuch finden konnte.

Von diesem Szenario ausgehend gibt es derzeit recht wenige Nachbearbeitungsverfahren, welche die so gelieferten Ergebnisse der Texterkennung verbessern; zudem werden die wenigsten solcher Verfahren – soweit bekannt – kommerziell genutzt. Nach Art der verwendeten Technik und des eingesetzten Wissens zur Nachbearbeitung kann man drei Gruppen von Verfahren unterscheiden:

- Wortorientierte statistische Verfahren, welche hochfrequente Wort-Tupel (meist Paare), sogenannte Wort-Kollokationen, besonders betrachten; meist geschieht dies ohne Beachtung der Reihenfolge und unter bestenfalls geringer Bewertung des Wortabstandes.
- Wortklassenorientierte statistische Verfahren, welche typischerweise *n*-Gramme (mit festgelegtem *n*) über Wortklassen verwenden; zusätzlich werden teilweise Frequenzen von Wörtern und Wortklassen sowie Wahrscheinlichkeiten für die Wortklassenzugehörigkeiten eingesetzt.
- Konstituentenorientierte statistische Verfahren nach Art des probabilistischen Parsing; dazu werden meist einfache syntaktische Konstruktionen, z. B. in Form kontextfreier Grammatiken, formuliert und die Regeln der Grammatik zusätzlich mit Wahrscheinlichkeiten annotiert.

Bei den erstgenannten, wortorientierten Verfahren werden die Wörter in Ihrer Oberflächenform direkt verwendet. Es erfolgt also keine morpho-syntaktisch orientierte Vorverarbeitung. Da die Zahl der Wortvollformen recht hoch ist, wird dabei bewußt auf die niederfrequenten Anteile der sprachlichen Äußerungen verzichtet.

Bei den wortklassenorientierten Verfahren, der zweiten Gruppe, werden statt einzelner Wörter Klassen derselben und deren Aneinanderreihung verwendet. Prinzipiell ist hier auch vorstellbar, statt der Wortklassen Einzelwörter (Wortformen) zu verwenden. Solche Ansätze sind aber technisch nur dann realisierbar, wenn ein überschaubarer Wortschatz vorliegt, da sonst die Zahl der Kombinationen zu groß wird.

Die dritte genannte Gruppe, die syntaxorientierten Verfahren, sind Kombinationen aus grammatik- bzw. parsing-orientierten und statistischen Verfahren. Sie haben gegenüber den zuvorgenannten den Vorteil, daß sie die Beschreibungsmächtigkeit und Präzision der Grammatikformalismen mit der frequenzbasierten Effizienz der statistischen Ansätze kombinieren (können). Diese Ansätze des stochastischen oder probabilistischen Parsing sind theoretisch weitgehend erforscht, wurden jedoch bisher nur selten im Gebiet der Dokumentanalyse eingesetzt.

Will man etwas stärker von den konkreten Verfahren abstrahieren, so kann man sagen, daß die Relevanz eines Trainingskorpus von oben nach unten abnimmt, und das bei gleichzeitiger Zunahme des manuellen Kodierungsaufwandes. Ebenso sinkt die Orientierung am einzelnen Lexem von oben nach unten zugunsten der Lexemklassen bis hin zu morphosyntaktischen Merkmalsvektoren. Zugleich steigt nach unten hin die Komplexität der Verfahren (nicht zwingend des Berechnungsaufwandes, sondern die Kompliziertheit ihrer Struktur) und der notwendigen Repräsentationsformalismen, was eine rechnergestützte Modellierung zunehmend erschwert.

Wir werden uns i. F. auf die zweitgenannte Verfahrensgruppe, also die wortklassenorientierten statistischen Verfahren, beschränken und die im Rahmen der vorliegenden Arbeit untersuchten Verfahren etwas detaillierter beschreiben. Da allen untersuchten Verfahren mehr oder weniger explizit die Theorie der Hidden Markov-Modelle zugrunde liegt, werden diese vorab in Abschnitt 3.1 erörtert. Der mit dieser Materie auch nur ansatzweise vertraute Leser kann getrost diesen Abschnitt überspringen. Danach folgen (Abschnitt 3.2 bis Abschnitt 3.4) die Beschreibungen der drei Verfahren nach FORNEY 1973, DE MARCKEN 1990 und KEENAN ET AL. 1991.

3.1 Grundlagen der Hidden Markov-Modelle

Hidden Markov-Modelle (kurz HMM) dienen der Beschreibung diskreter stochastischer Prozesse. Dies findet insbesondere Verwendung bei der Modellierung zeitlicher Prozesse, welche typischerweise Folgen von Ereignissen aufweisen. Ein allgemeines HMM wird beschrieben durch:

- die Menge der versteckten Zustände $S_1, \ldots, S_N$;
- die Menge der sichtbaren Symbole $V_1, \ldots, V_M$;
- die Wahrscheinlichkeitsverteilung der Zustandsübergänge A_{ij} mit i, j in [1..N] wobei A_{ij} die Wahrscheinlichkeit eines Übergangs von Zustand S_i nach S_j ist;
- die Wahrscheinlichkeitsverteilung für die Beobachtung der Symbole $B_j(k)$ mit k in [1..M] und j in [1..N] wobei $B_j(k)$ die Wahrscheinlichkeit der Beobachtung von Symbol V_k im Zustand S_j ist;
- die initiale Zustandsverteilung π_i mit i in [1..N] wobei π_i die initiale Wahrscheinlichkeit für Zustand S_i ist.

Was die Wahrscheinlichkeitswerte betrifft, so gehen wir im Folgenden davon aus, daß „prozentuale“ Wahrscheinlichkeitswerte vorliegen, also Zahlen zwischen 0 und 1, die

überwiegend multiplikativ verrechnet werden, was für eine konkrete Implementierung natürlich unsinnig wäre.

Durch obige Modellierung ist lediglich ein binäres HMM beschrieben, da durch die A_{ij} nur die Übergänge zwischen jeweils zwei Zuständen beschrieben werden; man spricht dann von den A_{ij} auch als 2-Grammen oder Bigrammen. Erweitert man die Wahrscheinlichkeitsverteilung A auf mehr als zwei Zustände, so erhält man entsprechend 3-Gramme oder Trigramme bzw. höhergradige n-Gramme. Weitere grundlegende Details zu HMM finden sich z. B. in RABINER 1988.

Um mit einem HMM das Verhalten natürlicher Sprache auf Wortebene zu modellieren, gibt es eine Reihe von Möglichkeiten, von denen jedoch nur eine von praktischem Belang ist. Zur Veranschaulichung sei vorab eine mögliche Übertragung von HMM auf die Buchstabenebene erläutert, welche bisweilen zur Unterstützung der Einzelzeichenerkennung verwendet wird. Dabei werden als versteckte Zustände S_i die Buchstaben des Alphabets gewählt, die sichtbaren Symbole sind gewisse, durch das OCR-Verfahren bestimmte, Merkmale der Buchstaben (beispielsweise Ober-/Unterlänge, Schwärzungsgrad; vgl. Abschnitt 2.2). Die Zustandsübergänge A_{ij} sind dadurch als die Kombinationswahrscheinlichkeiten zweier Buchstaben S_i und S_j festgelegt, während die $B_j(k)$ angeben, wie wahrscheinlich der Buchstabe S_j bei Beobachtung des Merkmals V_k ist. Da Merkmale dabei meist als Vektoren von Einzelmerkmalen repräsentiert werden, darf diese Erläuterung nur als stark vergröbert verstanden werden.

Anders sieht die Modellierung auf Wortebene aus, da dort die Anzahl der relevanten Wörter aus guten Gründen nicht als endlich angesehen werden kann. Eine unendliche Wortmenge hätte aber bei analoger Modellierung eine unendliche Zustandsmenge zur Folge. Daher werden die versteckten Zustände S_i nicht auf Einzelwörter, sondern auf Wortklassen abgebildet. Üblicherweise wählt man für die Wortklassen ein sogenanntes Tagset, das in der Regel mehrere zehn bis wenige hundert Tags enthält. Hinsichtlich ihrer linguistischen Beschreibungsstärke können solche Tagsets als Kompromiß angesehen werden zwischen einer feinen morphologischen Klassifizierung, welche alle Flexionsangaben einschließt, und einer groben Klassifizierung, welche gerade die Flexionsangaben wegläßt. Im Übrigen ist die Modellierung analog zu der genannten auf Zeichenebene.

Auf dieser relativ einfachen Basistheorie der HMM entstanden eine Reihe von Techniken, die für ihren praktischen Einsatz von großer Bedeutung sind. RABINER (vgl. RABINER 1988) unterteilt diese in drei zentrale Problemfelder:

- Wie berechne ich am effizientesten die Wahrscheinlichkeit einer gegebenen Symbolfolge (bei uns: Wortfolgen als Äußerungen oder Sätze) mit Symbolen aus den V_k bzgl. eines gegebenen Modells?
- Wie finde ich die „beste Zustandssequenz“ zu einer gegebenen Symbolfolge und wie kann ich sinnvoll den Begriff „beste Zustandssequenz“ an dieser Stelle definieren?

- Wie erstelle oder trainiere ich ein HMM, welches für eine gegebene Menge von Symbolfolgen, also einem textuellen Trainingskorpus, die beste Gesamtwahrscheinlichkeit („beste“ im Sinne der vorherigen Frage) für das Korpus liefert?

Die erste Frage ist für unsere Anwendung von eher akademischen Interesse, während ihre Lösung – die sogenannte Forward-Backward-Procedure – auch zur Lösung der beiden anderen Fragen beiträgt. Die zweite Frage wird durch die im Folgenden vorgestellten Verfahren beantwortet, bei welchen das bekannteste der VITERBI-Algorithmus ist, welcher selbst eine Weiterentwicklung der Forward-Backward-Procedure ist. Bei der letzten Frage dreht es sich schließlich darum, wo ein HMM herkommt, was klassischerweise durch den BAUM-WELCH-Algorithmus realisiert wird. Dies ist zwar auch in unserem Problemfeld interessant, wird jedoch im Weiteren nicht mehr beachtet.

In unserem Kontext ist nun das Ziel beim Einsatz von Tagging-Techniken, die wahrscheinlichste Sequenz von durch die Texterkennung gelieferten Worthypothesen zu finden. Dabei ist anzumerken, daß eine Texterkennung im Normalfall ein Hypothesennetz liefert, bei dem die Knoten des Netzes genau den Leerstellen zwischen den geschriebenen Wörtern entsprechen – man spricht hier auch von einem Trellis, da die graphische Darstellung dieser Netze klare spalierähnliche Trennstellen aufweist. Für solche Trellis-Strukturen sind die HMM hervorragend geeignet. Im allgemeinen Fall jedoch weisen Worthypothesennetze ambige Nahtstellen auf, die von unklar erkannten Leerstellen herrühren. Für solche Fälle lassen sich HMM nur noch mit einigen Behelfslösungen anwenden. Dies ist wiederum analog zum Problem der Wortgrenzenerkennung in der (akustischen) Sprachverarbeitung.

Die folgenden drei Abschnitte geben nun die eigentlichen Techniken wieder, die zur Lösung des zweiten Problemfeldes der HMM, dem Finden der besten Zustandssequenz, dienen. Wir werden uns in diesen Abschnitten an die Bezeichnungen aus diesem Abschnitt halten, um die Algorithmenskizzen so kurz wie möglich zu halten.

3.2 HMM à la FORNEY

Die klassische Art, um die beste Zustandssequenz zu einer gegebenen Symbolfolge zu bestimmen, stellt der VITERBI-Algorithmus dar, was das Thema der Arbeit FORNEY 1973 ist. Wie bereits die Problemformulierung in Abschnitt 3.1 anklingen ließ, ist eine der Hauptaufgaben bei der Lösung des zweiten HMM-Problems, den Begriff der Optimalität einer Zustandssequenz festzulegen. In der Praxis bedeutet dies im Wesentlichen, daß jeder Grundalgorithmus sein eigenes Optimalitätskriterium hat. Dem widerspricht nicht, daß es zu jedem solchen Algorithmus eine schier unüberschaubare Zahl von Spezialisierungen und Erweiterungen gibt, die demselben Optimalitätskriterium genügen und auf dem gleichen Basisalgorithmus aufbauen.

Das Optimalitätskriterium des VITERBI-Algorithmus ist, die höchste Gesamtwahrscheinlichkeit einer Zustandssequenz zu einer gegebenen Symbolfolge im Sinne der BAYES’schen Wahrscheinlichkeit zu erreichen. Formal geschrieben soll

- für eine Zustandssequenz $Z=S_{z(1)}, \ldots, S_{z(k)}$ und
- zu einer Symbolfolge $X=V_{x(1)}, \ldots, V_{x(k)}$
- der Wert $K(X,Z) = \pi_{Z(1)} * \prod_{i=1}^{k} B_{Z(i)}(x(i)) * \prod_{i=1}^{k-1} A_{Z(i),Z(i+1)}$

maximiert werden. Der Wert *K(X, Z)* kann als die Wahrscheinlichkeit der Zustandssequenz (also „des Pfades“) *Z* durch den Graphen bezüglich der Symbolfolge *X* aufgefaßt werden. Mit anderen Worten: Ziel des VITERBI-Algorithmus ist es, die Zustandssequenz zu finden, welche als Ganzes die wahrscheinlichste ist.

Dazu wird das HMM von links nach rechts schrittweise für *i* von 1 bis *k* (*k* ist die Länge der Zustandssequenz, vgl. obige Formeln) durchlaufen und in jedem Schritt für den zuletzt erreichten Zustand $S_{z(i)}$ die bis dahin erreichbare Maximalwahrscheinlichkeit und die davorliegende Zustandssequenz berechnet. Hierzu ist eine entsprechende Buchführung notwendig, um keine Möglichkeit zu übersehen.

3.3 HMM à la DE MARCKEN

Eine Modifikation des VITERBI-Algorithmus wird von DE MARCKEN 1990 vorgeschlagen: anstatt die Zustandsfolge zu suchen, die als Ganzes genommen die wahrscheinlichste ist, beschränkt sein Algorithmus sich darauf, die Anzahl der korrekt getaggten Wörter in der resultierenden Zustandsfolge zu maximieren. Dies bedeutet, daß für jedes Einzelsymbol $V_{x(i)}$ der Zustand $S_{z(i)}$ gewählt wird, welcher in der Summe über alle denkbaren Zustandssequenzen am wahrscheinlichsten ist. Dabei kann es jedoch passieren, daß als Endergebnis eine Zustandssequenz $Z=S_{z(1)}, \ldots, S_{z(k)}$ ermittelt wird, die als Ganzes genommen eine Wahrscheinlichkeit von Null hat, weil für eine Stelle die Übergangswahrscheinlichkeit $A_{z(i),z(i+1)}=0$ ist; vgl. dazu die Formel für *K(X, Z)* im letzten Abschnitt. Ist die Vollständigkeit der Lösungsberechnung für die gegebene Anwendungen wichtig, so ist diese Ausgabe in der Regel unerwünscht. Im vorliegenden Fall einer fehlerbehafteten, statistisch orientierten Anwendung innerhalb der Dokumentanalyse kann sich dieser vermeintliche Schönheitsfehler des Verfahrens in einen Vorteil verwandeln.

Um dies zu erläutern, müssen wir jedoch etwas weiter ausholen. In den nicht seltenen Fällen, wo ein Wort nicht erkennbar ist (schlechte Druckqualität oder Lücke im Lexikon), liegt für den VITERBI-Algorithmus eine Äußerungsgrenze (Grenze der Beobachtungssequenz) vor, da an dieser Stelle auch kein sinnvolles Tag gefunden werden kann. Für den modifizierten Algorithmus nach DE MARCKEN kann dies ebenso gehandhabt werden. Kritisch wird es in dem ähnlichen, für das Programm jedoch nicht unterscheidbaren Fall, daß ein Wort eigentlich zurückgewiesen werden müßte, da die Eingabe unzureichend ist oder im Lexikon kein wirklich passender Kandidat gefunden wird, die Erkennungskomponente jedoch ein ähnliches Wort als Ergebnis liefert. In diesem Fall liefert der VITERBI-Algorithmus typischerweise falsche Ergebnisse für die

Umgebung dieses Wortes, während der modifizierte DE MARCKEN-Algorithmus sich hier als robuster erweist, weil die Auswirkungen dieses Defekts nicht ganz so weit in die Umgebung des Wortes durchschlagen. Dies zeigt sich primär beim Vergleich von Einzelergebnissen, schlägt sich aber auch bei den Statistiken am Ende dieses Artikel nieder.

Die technische Realisierung dieses Algorithmus wird durch eine Modifikation der Forward-Backward-Procedure erreicht, wodurch i. W. die gleiche Komplexität wie beim VITERBI-Algorithmus des letzten Abschnitts erreicht wird. In beiden Fällen ist der Hauptaufwand quadratisch bezüglich der Anzahl der Zustände im HMM und ebenfalls quadratisch bezüglich der Länge der Beobachtungssequenz.

3.4 Bewertungsverfahren à la KEENAN et al.

Das dritte von uns untersuchte Verfahren wurde von KEENAN, EVETT und WHITROW 1991 (vgl. KEENAN ET AL. 1991) vorgestellt. Das Verfahren verzichtet einerseits auf die „mathematische Fundiertheit“ der Hidden Markov-Modelle, was seiner Komplexität zugute kommt, nimmt andererseits jedoch eine Beschreibungsgröße hinzu (im Vergleich zu HMM), was seine potentielle Leistungsfähigkeit erhöht. Die Autoren selbst betrachten das Verfahren auch nicht als eine Variante der HMM-Verfahren, sondern bezeichnen es schlicht als *scoring function*.

Vernachlässigt man die Wahrscheinlichkeitsverteilung der Anfangszustände, so werden bei Hidden Markov-Modellen gemäß obiger Modellierung für Worthypothesennetze die beiden folgenden statistischen Größen berücksichtigt:

- die Aufeinanderfolge der Wortklassen A_{ij} und
- die Klassenzugehörigkeit der Wortindividuen $B_j(k)$.

Bei KEENAN et al. 1991 findet sich hierzu eine abgewandelte Terminologie. Sie bezeichnen die A_{ij} (wie allgemein üblich) als *n*-Gramme und $B_j(k)$ als *grammatical frequency factor*, kurz GFF. Als weiteres statistisches Maß nehmen sie den *lexical probability factor* LPF hinzu, der angibt, mit welcher Wahrscheinlichkeit die Texterkennung glaubt, das Wort als solches erkannt zu haben. So wird die erwartungsorientierte (top-down) Strategie der Hidden Markov-Modelle durch eine erkennungsgetriebene (bottom-up) Strategie ergänzt.

Zur Bewertung der Einzelhypothesen und somit zum Finden eines optimalen Pfades durch das Hypothesennetz verwenden sie eine Bewertungsfunktion, welche eines bis drei der obigen Maße berücksichtigt. Dazu werden die Einzelwerte dieser Maße – unter gewissen Einschränkungen – i. W. miteinander multipliziert. Die Bewertungsfunktion berücksichtigt je nach Einstellung zwei bis drei benachbarte Wortpositionen im Hypothesennetz. Als Ergebnis wird eine Rangfolge der Alternativen geliefert, aus welchen dann z. B. die jeweils erstbeste ausgewählt werden kann.

Nach den Ergebnissen, die KEENAN et al. 1991 selbst präsentieren, lohnt sich die Verwendung aller drei betrachteten statistischen Größen erst bei Trigrammen. Dort

allerdings sollen die Ergebnisse signifikant besser sein als bei Bigrammen, auch wenn diese alle drei Größen verwenden.

4 Implementierung

Die Implementierungen wurden auf *Sun SPARCstations* unter C++ durchgeführt. Die in Abschnitt 3 beschriebenen drei Verfahren wurden implementiert, an einem getaggten Zeitungskorpus trainiert und an einem Korpus von Geschäftsbriefen getestet.
Das Programm für die HMM-basierte Bewertung nach FORNEY 1973 erhält als Parameter die Anzahl der Pfade, das nach DE MARCKEN 1990 die Anzahl der Tags, welche betrachtet werden sollen. Das Programm für die Bewertung nach KEENAN et al. 1991 erhält als Parameter die Anzahl der zu wählenden Wörter. Wie bereits erwähnt, wurden die folgenden Test nur für Bigramme durchgeführt.

Trainiert wurde anhand der rund 34 Millionen laufenden Wörter des Korpus der Tageszeitung Frankfurter Rundschau, welcher von der Universität Gesamthochschule Paderborn auf der CD-ROM „ECI Multilingual Corpus I“ (ECI 1994) zur Verfügung gestellt und am Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart maschinell (also relativ fehlerbehaftet) getaggt wurde. Dazu wurde das „alte Stuttgarter Tagset“ mit 52 Tags verwendet (dies liegt uns in Form der internen „Tagging-Guidelines“ von Anne SCHILLER vom 16. 12. 1994 vor).

Danach wurden die besagten Verfahren an einem hauseigenen Korpus von Geschäftsbriefen getestet. Dabei handelt es sich um diverse Bürokorrespondenz des DFKI, welche in den Jahren 1991 bis 1993 gesammelt wurde und von denen etwa 90 Dokumente für unsere Zwecke brauchbar (d. h. deutschsprachig) waren. Der *OCR-GroundTruth* für diese Dokumente, d. h. die korrekten Kontrolldaten auf Zeichenebene, existiert in Form von Textdateien, welche manuell erstellt wurden. Für die Tests wurden die Briefteile Absender, Empfänger, Betreff und Textrumpf herangezogen, die in der Größe etwa zwischen 50 bis 200 laufenden Wörtern rangieren.

Trotz der nicht ganz überzeugenden Qualität des (automatisch getaggten) Trainingskorpus und der großen linguistischen Diskrepanz zwischen Test- und Trainingskorpus (Zeitungstexte vs. Geschäftsbriefe) können die nachfolgend beschriebenen Ergebnisse als recht ermutigend gewertet werden.

5 Bewertungsmethode

Zur Auswertung der von den verschiedenen Verfahren gelieferten Ergebnisse haben wir die in den Message Understanding Konferenzen (MUC-4 1992) verwendeten Maßzahlen und Evaluierungsmetriken auf die vorliegende Anwendung übertragen. Dazu sind primär die folgenden Maßzahlen zu definieren.

POS (für englisch *possible*) für die Anzahl der Wörter im *GroundTruth*;
ACT (für englisch *actual*) für die Anzahl der von der Analyse gelieferten Wörter;

COR (für englisch *correct*) für die Anzahl der von der Analyse korrekt gelieferten Wörter;

SPU (für englisch *spurious*) für die Anzahl der von der Analyse gelieferten Wörter, für die jedoch kein *GroundTruth* angegeben ist;

Auf Basis dieser Maßzahlen werden nun die eigentlichen Evaluierungsmetriken wie folgt definiert.

REC (für englisch *recall*) = COR / POS, d. h. der Anteil der korrekten bzgl. der möglichen Worthypothesen aus dem *GroundTruth*;

PRE (für englisch *precision*) = COR / ACT, d. h. der Anteil der korrekten bzgl. der aktuell gelieferten Worthypothesen der Analyse;

OVG (für englisch *overgeneration*) = SPU / ACT, d. h. der Anteil der zuviel gelieferten Wörter, die im *GroundTruth* nicht gegeben sind.

Eine Alternative zu diesem Bewertungsverfahren wäre, die bei der Informationsextraktion ermittelten Ergebnisse auf Basis der verschiedenen Verfahren zur Verbesserung der Worterkennung zu vergleichen. Dies stellt für einen konkreten Fall von Anwendungsentwicklung sicherlich ein optimales Verfahren dar. Für eine etwas generellere Darstellung wie in diesem Papier ist sie jedoch ungeeignet. Wie wir bei der Entwicklung verschiedener Dokumentanalysesysteme feststellen konnten, kann bereits die geringfügige Änderung von Vorverarbeitungsschritten (zu denen die untersuchten Verfahren zu rechnen sind) entscheidenden Einfluß auf die Güte des Gesamtsystems haben. Dies ist einerseits ein Vorteil für das Tuning des Systems, da hierzu viele Möglichkeiten gegeben sind, andererseits erschwert es den „objektiven“ Vergleich der Vorverarbeitungsverfahren, da jeder Informationsextraktor im Extremfall ein anderes Optimalitätskriterium für die Vorverarbeitung hat. Dazu ein Beispiel: die im `OfficeMAID`-Prototypen eingesetzten Informationsextraktoren (siehe Abschnitt 2.2 am Ende) verwenden insgesamt drei verschiedene Parametrisierungen des lexikalischen Abgleichs, welche jeweils unterschiedliche Ergebnishypothesen bewirken.

6 Ergebnisse

Tabelle 1 zeigt die Ergebnisse der Auswertung für zwei extreme Beispieldokumente: ein sehr gut erkanntes Dokument mit 151 laufenden Wörtern Text und ein sehr schlecht erkanntes Dokument mit 122 laufenden Wörtern Text. Die übrigen, hier nicht weiter betrachteten Dokumente liegen im Wesentlichen zwischen diesen beiden Extrema, sodaß eine weitere Tabellenspalte mit den Mittelwerten nicht besonders aussagekräftig wäre. Wir wollen die in der Tabelle festgehaltenen Ergebnisse nun kurz diskutieren. Beim Lesen der Tabelle ist zu berücksichtigen, daß hoher *recall* und hohe *precision* sowie eine niedrige *overgeneration* erstrebenswert sind.

Man sieht schnell, daß die Erkennungsrate – unabhängig von der Nachbearbeitung – insgesamt ziemlich schlecht ist, was zwei Gründe hat. Zum ersten wurde beim lexikalischen Abgleich ein Wörterbuch verwendet, das aus dem Zeitungskorpus generiert

wurde. Daher treten im Lexikon viele relevante Wörter (z. B. „Rechnungsdatum“, „Angebotsanfrage“) überhaupt nicht auf und viele in Geschäftsbriefen hochfrequente Wörter haben unverhältnismäßig niedrige Häufigkeiten im Modell; so ist z. B. „Ihnen“ in Zeitungen etwa so häufig wie „örtlichen“, „Gegenteil“, (E. T. A.) „Hoffmann“ und „Nachwuchs“, d. h. etwa ein Tausendstel mal so häufig wie die häufigsten Wörter des Deutschen: „der“ und „die“. In Extremfällen führen diese niedrigen Frequenzen dazu, daß die Wörter beim Training des Hidden Markov-Modells ignoriert werden. Zum zweiten ist der bei den Tests eingesetzte Einzelzeichenerkenner keineswegs auf dem aktuellen Stand der Technik oder gar ein kommerzieller Erkenner, sondern es handelt sich um eine bejahrte Eigenentwicklung aus dem Vorgängerprojekt, die eine aus heutiger Sicht eher schlechte Erkennungsrate aufweist.

Eine Besonderheit an der Tabelle verdient kurze Beachtung. Maximaler *recall* wird für die originäre Worterkennung erreicht, was nicht weiter verwunderlich ist: die Nachbearbeitung durch statistische Verfahren kann nur aus den bereits gegebenen Hypothesen welche herausstreichen, sie kann keine neuen hinzuerfinden. Ebenso verhält es sich mit den anderen beiden Größen: *Precision* und *overgeneration* können durch diese Art der Nachbearbeitung nur verbessert werden, d. h. die *precision* kann nur numerisch ansteigen, die *overgeneration* nur numerisch abfallen.

	schlecht erkanntes Dokumentgut			*erkanntes Dokument*		
Verfahren	*recall*	*precision*	*overgen.*	*recall*	*precision*	*overgen.*
Worterkennung ohne Nachverarbeitung	38,41%	26,50%	55,00%	88,33%	46,29%	51,97%
HMM à la FORNEY, 1 Pfad	28,26%	43,33%	0,00%	60,00%	65,45%	0,00%
HMM à la FORNEY, 2 Pfade	28,99%	43,48%	2,17%	64,17%	65,25%	6,78%
HMM à la FORNEY, 10 Pfade	30,43%	42,86%	8,16%	70,00%	66,14%	13,39%
HMM à la DE MARCKEN, 1 Tag	34,06%	36,15%	30,77%	86,67%	61,90%	34,52%
HMM à la DE MARCKEN, 2 Tags	34,06%	36,15%	30,77%	86,67%	61,90%	34,52%
HMM à la DE MARCKEN, 10 Tags	34,78%	36,36%	31,82%	86,67%	61,18%	35,29%
Verfahren nach KEENAN et al., 1 Wort	32,61%	38,79%	22,41%	80,83%	58,79%	33,33%
Verfahren nach KEENAN et al., 2 Wörter	36,23%	33,33%	40,00%	84,17%	54,30%	40,86%

Tabelle 1: Ergebnisse der auf Tagging basierten Nachbearbeitung

Nun wollen wir uns den Unterschieden zwischen den einzelnen Verfahren zuwenden. Der VITERBI-Algorithmus nach FORNEY 1973 liefert insgesamt die beste Reduktion der Overgeneration – bei der Betrachtung von nur einem besten Pfad bei beiden Dokumenten sogar zufällig auf 0% – und die stärkste Verbesserung der *precision* auf über 42% (bei dem schlechten) bzw. 65% (bei dem guten Dokument), was zwangsläufig mit einer starken Verringerung des *recall* auf unter 31% bzw. 70% einhergeht. Diese hinsichtlich der *precision* sehr guten Ergebnisse werden, wie im Abschnitt 3.2 bereits erwähnt, mit einer etwas höheren Rechenkomplexität erkauft.

Im Unterschied dazu hebt der Algorithmus nach DE MARCKEN 1990 die *precision* nur etwas an, nämlich auf rund 36% bzw. 61%, wobei auch die Overgeneration auf einem recht hohen Niveau von über 30% bzw. 34% verbleibt. Positiv ist, daß dabei der *recall* weniger leidet: Er bleibt auf immerhin rund 34% bzw. 86%, wird also nur um

4% bzw. 2% verringert. Auffällig ist bei diesem Verfahren, daß bei beiden Dokumenten die Varianz der verschiedenen Metrikwerte über die Anzahl der betrachteten Pfade, also dem Parameter des Verfahrens, sehr gering ist: sie bleibt in allen Fällen unter einem Viertel Prozent (exakt 0,00245) und ist einmal (beim *recall* des gut erkannten Dokuments) sogar Null.

Beim Bewertungsverfahren nach KEENAN et al. 1991 kommen in etwa Werte zwischen den beiden erstgenannten Verfahren heraus, wobei hier jedoch die Varianz zwischen der Betrachtung eines oder zweier Wörter wesentlich höher ist. Das Verfahren liefert bei Betrachtung zweier Wörter im Falle beider Dokumente die vergleichsweise schlechtesten Ergebnisse (nur die Originaldaten der Texterkennung sind schlechter), wobei jedoch gegenüber den Ausgangsdaten sowohl *precision* als auch Overgeneration stark verbessert werden und zugleich der *recall* sehr hoch bleibt. Bei der Betrachtung nur des bestbewerteten Wortes ist das Ergebnis wesentlich zufriedenstellender: Der *recall* bewegt sich mit 33% bzw. 81% zwischen dem der beiden anderen Verfahren, ebenso die Overgeneration mit 22% bzw. 33%. Im Falle der *precision* ist es jedoch unterschiedlich: während diese beim schlechten Dokument mit 39% wesentlich gegenüber dem DE MARCKEN-Verfahren gewinnt, aber gegenüber dem FORNEY-VITERBI zurücksteht, liegt sie beim guten Dokument mit 59% etwas hinter beiden Ergebnissen.

Vergleicht man die Ergebnisse hinsichtlich der beiden Extremdokumente, so gewinnt FORNEYS VITERBI-Algorithmus in beiden Fällen, jedoch holen die beiden anderen Verfahren mit zunehmender Dokumentqualität stark auf. Auch wenn man die Verbesserung der *precision* gegen den Verlust des *recall* aufrechnet (z. B. mit dem sogenannten *f-Measure*), gewinnen die beiden letztgenannten Verfahren.

Insgesamt kann eine Empfehlung für eines der Verfahren nur bei Vorgabe einer klaren Zielsetzung ausgesprochen werden. Ist eine hohe Abdeckung hinsichtlich der Verfügbarkeit der korrekten Hypothesen in der Gesamthypothesenmenge erwünscht, so muß der *recall* hoch bleiben, weshalb keine Nachbearbeitung das Beste sein dürfte. Generell muß eine Unterscheidung hinsichtlich der Komplexität jener Komponenten getroffen werden, welche letztendlich die Worthypothesen konsumieren – also üblicherweise eine inhaltliche Analyse. Ist der Rechenaufwand dieser Weiterverarbeitung stark von der Anzahl der gelieferten Hypothesen abhängig, so ist eine Verwendung des VITERBI-Algorithmus nach FORNEY in Erwägung zu ziehen. Gegebenenfalls muß dabei eine iterative Verwendung der Nachbearbeitung berücksichtigt werden, um bei Mißerfolg doch noch an die benötigte Hypothese heranzukommen. Im schlimmsten Fall muß doch noch die gesamte Ursprungshypothesenmenge der Texterkennung abgearbeitet werden. Da ein häufiges Aufrufen des VITERBI-Algorithmus aber auch sehr teuer ist, wird oft ein Ausweichen auf eines der anderen Verfahren sinnvoller sein. Ebenso kann eine Unterscheidung hinsichtlich der Dokumentqualität eine Rolle spielen, da sich die Verfahren auch hierin unterscheiden. Es bleibt also stets im konkreten Fall abzuwägen, welches der drei Verfahren am ehesten hilft.

7 Danksagung

Diese Arbeit wäre nicht zustande gekommen, wenn nicht mehrere Leute mir ihre Unterstützung hätten zukommen lassen. Diesen sei an dieser Stelle gedankt: Arno SCHUMACHER für die Implementierung der Nachbearbeitungsalgorithmen; Sylvio TABOR für die Implementierung (Änderung, Neuimplementierung, …) der Auswertungsprozedur; der Arbeitsgruppe „Textcorpora“ am Institut für Maschinelle Sprachverarbeitung an der Universität Stuttgart – seinerzeit vornehmlich Oliver CHRIST – für Testdaten und Zubehör; sowie Brigitte KÖPKE und Claudia WENZEL fürs Korrekturlesen.

Diese Arbeit wurde teilweise gefördert vom Bundesministerium für Bildung und Forschung (BMBF) unter dem Förderkennzeichen FKZ-ITW-9401 (Projekt OMEGA) und FKZ-ITW-9702 (Projekt Virtual Office).

Literaturverzeichnis

ABNEY, S. (1996). „Partial Parsing Via Finite-State Cascades.“ In: Proc. ESSLLI 1996 Robust Parsing Workshop.

ABRAHAM, W.; ZWART, J.-W. (edd.) (1994). Minimalism and Kayne's Asymmetry Hypothesis [= Groninger Arbeiten zur Germanistischen Linguistik 25].

AHMAD, K., et al. (1992). $MLEX_d$. Standards for a Multifunctional Lexicon. Report Esprit Project 5304 MULTILEX.

AHMAD, K.; DAVIES, A.; FULFORD, H.; ROGERS, M. (1992). „The semi-automatic extraction of terms from texts.“ In: SNELL HORNBY, M. (ed.) (1992). Translation Studies – an Interdiscipline. Amsteradm/Philadelphia: John Benjamins.

ALLWRIGHT, D.; BAILEY, K. M. (1991). Focus on the Language Classroom, an Introduction to Classroom Research for Language Teachers. CUP.

APPELT, D. E. et al. (1993). “FASTUS – A Finite-State Processor for Information Extraction from Real-world Text.” In: Proc. 13th IJCAI, Chambery, 1172-1178.

ARNTZ, R.; PICHT, H. (1991). Einführung in die Terminologiearbeit. Hildesheim et al.: Olms.

AUSTIN, J. L. (1979). Zur Theorie der Sprechakte (How to do things with words). Stuttgart: Reclam [= Reclam Universal-Bibliothek Nr. 9396].

AZZAM, S. et al. (1997). „Using a Language Independent Domain Model for Multilingual Information Extraction.“ In: Proc. 35th ACL Annual Meeting/8th EACL Conference. Madrid, Juli 1997. San Francisco/CA: Morgan Kaufmann.

BACHUT, D. et al. (1992). EUROLANG, Modèle de transfert. Eurolang Report, Site.

BAPTIST, H. et. al. (1996). „Entwurf eines hypertextbasierten Katalogs für die Institutsbibliothek des Instituts für Informationswissenschaft“ in: KRAUSE et al. (1996), 97-108.

BARR, A.; FEIGENBAUM, E. A. (edd.) (1981). The Handbook of Artificial Intelligence. Vol. 1. Reading/MA.: Addison-Wesley.

BARSKY, R. F. (1997). Noam Chomsky – A Life of Dissent. Online. Internet. *http://mitpress.mit.edu/e-books/chomsky/.*

BENSON, M. (1995). „Review article – Goran Kjellmer: A dictionary of English Collocations.“ In: International Journal of Lexicography 8 (1995) 1, 65–68.

BERGMANN, G. (1993). Wörterbuch der obersächsischen Mundarten. Bd. 3. Berlin: Akademie-Verlag.

BERWICK, R. C.; WEINBERG, A. S. (1984). The Grammatical Basis of Linguistic Performance. Cambridge/MA: The MIT Press.

BEUTEL, B. (1997). MALAGA 3.0. Online-Dokumentation zum MALAGA-System. Friedrich-Alexander-Universität Erlangen-Nürnberg, Abteilung für Computerlinguistik. Online. Internet. *http://www.linguistik.uni-erlangen.de/tree/html/malaga/malaga.html*.

BLÄSER, B.; WERMKE, M. (1990). Projekt Elektronische Wörterbücher/Lexika: Abschlußbericht der Definitionsphase. IWBS Report 145, Stuttgart: IBM Deutschland GmbH, Wissenschaftliches Zentrum, Institut für Wissensbasierte Systeme.

BÖHLE, K.; RIEHM, U.; WINGERT, B. (1997). Vom allmählichen Verfassen elektronischer Bücher. Frankfurt am Main/New York: Campus.

BOLLMANN, P. (1983). The Normalized Recall and Related Measures. Live Bericht Nr. 2/83, Technische Universität Berlin, Fachbereich Informatik.

BOWERMAN, C. (1991). “Tutoring in an ITS for German Writing: The Student-Tutorial Module Synergy.” In: Proc. of the ICALL Workshop, UMIST, September 1991.

BREINDL, E. (1998). "Konzeption und Konversion: Zur simultanen Produktion von Printtext und Hypertext am Beispiel ‘Grammatik’". In: STORRER & HARRIEHAUSEN (1998).

BÜCHEL, Gr. (1995). „Können Verben semantische Relationen markieren?“ In: WEBER, N. (ed.) (1995). Semantik, Lexikographie und Computeranwendungen. Kolloquium, Bonn 27.-28.1.1995. Tübingen: Niemeyer.

CHAPMAN, R. L. (ed.) (1993[5]). Rogets international Thesaurus. London: Harper.

CHOMSKY, N. (1986b). Barriers, Cambridge/MA: The MIT Press.

CHOMSKY, N. (1989). "Some Notes on Economy of Derivation and Representation" in FREIDIN, R. (ed.) (1991) Principles and Parameters in Comparative Grammar. Cambridge/MA: MIT Press, 417-454.

CHOMSKY, N. (1992). "A Minimalist Program for Linguistic Theory." In MIT Occasional Papers in Linguistics 1, 1-25 [auch in: HALE, K.; KEYSER, S. J. (edd.) (1993). The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger. Cambridge/MA.: MIT Press].

CHOMSKY, N. (1995). The Minimalist Program, Cambridge/MA.: The MIT Press.

CHURCH, K. W.; GALE, W.; HANKS, P.; HINDLE, D. (1991). „Parsing, Word Associations, and Typical Predicate-Argument Relations.“ In: TOMITA, M. (ed.) (1991). Current Issues in Parsing Technologies. Norwell/MA: Kluwer, 103-112.

COVINGTON, M. A. (1994). Natural Language Processing for Prolog Programmers. Englewood Cliffs/NJ: Prentice Hall.

COX, Th. B. (1995). ORACLE Workgroup Server Handbook. Berkeley et al.: Oracle Press.

DÄßLER, R. (1996). „Knowledge Browser – ein VRML-basiertes Navigationstool für Information Retrieval Systeme im World Wide Web.“ In: KRAUSE et al. (1996), 199-211.

DE MARCKEN , C. G. (1990). "Parsing the LOB Corpus." In: Proc. 28th Annual Meeting of the Association for Computational Linguistics. Pittsburg/PA, 243-251.

DEGEN, H. (1996). „Multimediale Gestaltbereiche als Grundlage für Entwurfswerkzeuge in multimedialen Entwicklungsprozessen.“ In: KRAUSE et al. (1996), 213-226.

DENGEL, A.; BLEISINGER, R.; HOCH, R.; HÖNES, F.; MALBURG, M.; FEIN, F. (1995). "OfficeMAID – A System for Automatic Mail Analysis, Interpretation and Delivery." In: SPITZ, L.; DENGEL, A. (edd.). (1995). Document Analysis Systems. World Scientific Publishing: Singapore, 52-75.

DENGEL, A.; HOCH, R.; HÖNES, F.; MALBURG, M.; WEIGEL, A. (1996). "Techniques for Improving OCR Results." In: WANG, P. S. P.; BUNKE, H. (edd.) (1996). Handbook on Optical Character Recognition and Document Analysis. World Scientific Publishers Company.

DERANSART, P.; ED-DBALI, A.; CERVONI, L. (1996). Prolog: The Standard. Berlin et al.: Springer.

DILLON, A. (1994). Designing Usable Electronic Text. Ergonomic Aspects of Human Information Usage. London: Taylor & Francis.

DIN 2342 (1992). Begriffe der Terminologielehre: Grundbegriffe. Teil 1. Berlin: Beuth.

DORNSEIFF, F. (1970[7]). Der deutsche Wortschatz nach Sachgruppen. Berlin/New York: De Gruyter.

DUNNING, Ted (1993). „Accurate Methods for the Statistics of Surprise and Coincidence.” In: Computational Linguistics 19(1) (1993), 61-74.

DUQIENNOY, I. et al. (1992). EUROLANG, Modèle terminologique. Eurolang Report, Site.

EBERLEH, E. (1994). „Industrielle Gestaltungsrichtlinien für graphische Benutzeroberflächen.“ In: EBERLEH, OBERQUELLE & OPPERMANN (1994), 145-195.

EBERLEH, E.; OBERQUELLE, H.; OPPERMANN, R. (edd.) (1994[2]). Einführung in die Software-Ergonomie. Berlin/New York: de Gruyter.

ECI = European Corpus Initiative (1994). Multilingual Corpus I CD. ELSNET OTS, Utrecht University, Trans 10, 3512 JK Utrecht, The Netherlands.

ECKLE, J.; HEID, U. (1996). „Extracting raw material for a German subcategorization lexicon from newspaper text.“ In: Proc. 4th International Conference on Computational Lexicography, COMPLEX 1996, Budapest.

EIMERMACHER, M. (1988). Wortorientiertes Parsen. Dissertation, Technische Universität Berlin, Fachbereich Informatik.

ENDRES-NIGGEMEYER, B. (1992). Abstrahieren, Indexieren und Klassieren. Ein empirisches Prozeßmodell der Dokumentrepräsentation. Habilitationsschrift, Universität Konstanz.

EVANS, D. A. et. al. (1991). „Automatic Indexing Using Selective NLP and First-Order Thesauri.“ In: LICHNEROWICZ, A. (ed.) (1991). Intelligent Text and Image Handling. Proc. RIAO ’91, Barcelona, April 1991. Amsterdam: Elsevier.

FELBER, H. (1993). Allgemeine Terminologielehre und Wissenstechnik. Wien: TermNet.

FITZPATRICK, C.; GRIESZL, A. (1996). Tutoring Module – Requirements Specification and Functional Specification. RECALL Deliverable 13.

FORNEY, G. D. Jr. (1973). "The Viterbi Algorithm." In: Proc. IEEE 61, 268-278.

FURCHE, Andreas; WRIGHTSON, Graham (1997). Computer Money. Zahlungssysteme im Internet. Heidelberg: dpunkt.

FRAKES, W. B.; BAEZA-YATES, R. (edd.) (1992). Information Retrieval. Englewood Cliffs/NJ: P T R Prentice Hall.

FRAZIER, L.; FLORES D'ARCAIS, G. B.; COOLEN, R. (1993). „Processing discontinuous words: On the interface between lexical and syntactic processing.“ In: Cognition 47(3) (1993), 219–249.

GAIZAUSKAS et al. (1997). „Concepticons vs. Lexicons: An Architecture for Multilingual Information Extraction.“ In: PAZIENZA, M. T. (ed.) (1997). Information Extraction. A multidisciplinarey Approach to an Engineering Information Technology. Berlin :Springer.

GAIZAUSKAS, R. (1995). XI: A Knowledge Representation Language based on Cross-Classification and Inheritance. Technical Report, Univ. Sheffield.

GAMRAT, O. et al. (1992a). EUROLANG, Morphological Model. Eurolang Report, Site.

GAMRAT, O. et al. (1992b). EUROLANG, Syntactico-Semantic Model. Eurolang Report, Site.

GEIGER, Kyle (1995). Inside ODBC. Redmond/WA: Microsoft Press.

GERSTL, P.; GRIESZL, A. (1997). Summary report on demonstrator and user's response. RECALL Deliverable 39. Heidelberg: IBM Germany.

GLOOR, P. A. (1990). Hypermedia-Anwendungsentwicklung. Eine Einführung mit HyperCard-Beispielen. Stuttgart: Teubner.

GÖSER, S. (1992). „A Chart Parser for Robust Grammars.” In: Proc. COLING 1992, Nantes.

GÖSER, S. (1997). „Inhaltsbasiertes Information Retrieval. Die TextMining-Technologie.” In: GLDV-Forum 14(1) 1997, 48-52.

GRICE, H. P. (1975). „Logic and Conversation.“ In: COLE, P.; MORGAN, J. L. (edd.) (1975). Syntax and Semantics 3: Speech Acts. New York: Academic Press, 41-58.

GRICE, H. P. (1978). „Further Notes on Logic and Conversation." In: COLE, P. (1978). Syntax and Semantics 9. New York: Academic Press, 113-128.

GRISHMAN, R.; SUNDHEIM, B. (1996). Message Understanding Conference 6 – A Brief History, In: Proc. 16th International Conference on Computational Linguistics, COLING 96, Kopenhagen.

GROSS, M. (1991). La forme d'un dictionnaire électronique. LADL-Report, Laboratoire d'Automatique Documentaire et Linguistique, Université Paris 7.

GUENTHNER, F; MAIER, P. (1996). „Das CISLEX-Wörterbuchsystem.“ In: FELDWEG, H.; HINRICHS, W. (edd.) (1996). Lexikon und Text. Tübingen: Niemeyer [Lexicographia Series Maior Bd. 73].

HAEGEMAN, L. (1994[2]). Introduction to the Government & Binding Theory. Oxford & Cambridge/MA: Blackwell.

HAHN, U., REIMER, U. (1986). „TOPIC Essentials.“ In: Proc. 11th International Conference on Computational Linguistics, COLING 86, 497-503.

HAHN, U.; SCHACHT, S.; BRÖKER, N. (1994). „Concurrent, Object-Oriented Dependency Parsing: The *ParseTalk* Model.“ In: International Journal of Human-Computer Studies 41(1/2) (1994), 179-222.

HAIDER, H. (1985) "V-Second in German." In: HAIDER, H.; PRINZHORN, M. (edd.) (1985). Verb Second Phenomena in Germanic Languages. Dordrecht: Foris Publications, 49-75.

HAMMWÖHNER, R. (1990). Automatischer Aufbau von Hypertextbasen aus deskriptiv-expositorischen Texten. Ein Hypertext-Modell für das Information-Retrieval. Dissertation, Universität Konstanz, Fachbereich Informationswissenschaft.

HANRIEDER, G. (1991). Robustes Wortparsing. Lexikonbasierte morphologische Analyse (komplexer) deutscher Wortformen. Magisterarbeit, Universität Trier, Februar 1991.

HANRIEDER, G. (1994). „MORPH – Ein modulares und robustes Morphologieprogramm für das Deustche in Common Lisp.“ In: LDV-Forum 11(1) (1994), 30-38.

HANRIEDER, G. et al. (1996). Homepage: Informationen zu SYSLID. World Wide Web. *http://www.forwiss.uni-erlangen.de/~hanriede/syslid.html*, 1996.

HANRIEDER, G.; Heisterkamp, P. (1994). „Robust Analysis and Interpretation in Speech Dialogue.“ In: NIEMANN, H.; DE MORI, R.; HANRIEDER, G. (1994). Progress and Prospects of Speech Research and Technology. Proc. CRIM/FORWISS Workshop, München, 1994.

HASEBROOK, Joachim (1995). Multimedia-Psychologie. Heidelberg et al.: Spektrum.

HAUSER, R.; STORRER, A. (1994). "Dictionary Entry Parsing Using the LEXPARSE System." In: Lexicographica 9 (1993), 174-219.

HAUSMANN, F. J. (1985). „Kollokationen im deutschen Wörterbuch. Ein Beitrag zur Theorie des lexikographischen Beispiels.“ In: BERGENHOLTZ, H.; MUGDAN, J. (edd.) (1985). Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch Juni 1984. Tübingen, 118–129.

HAUSSER, R. (1989). Computation of Language. Berlin et al.: Springer.

HAUSSER, R. (1992). „Complexity in Left-Associative Grammar.“ In: Theoretical Computer Science, 106(2), 283-308, Amsterdam et al.: Elsevier.

HAUSSER, R. (1998). Grundlagen der Computerlinguistik [erscheint].

HAUSSER, R. (ed.) (1996). Linguistische Verifikation. Dokumentation zur ersten Morpholympics 1994. Tübingen: Niemeyer.

HAYES, J. R., FLOWERS, L. S. (1983). „Uncovering Cognitive Processes in Writing: An Introduction to Protocol Analysis.“ In: MOSENTHAL, P., TAMOR, L., WALMSLEY, S. A. (edd.) (1983). Research on Writing. Priciples and Methods. London.

HELBIG, G.; SCHENKEL, W. (1983). Wörterbuch zur Valenz und Distribution deutscher Verben. Tübingen: Niemeyer.

HELBIG, H.; MERTENS, A. (1994). „Word Agent Based Natural Language Processing.“ In: BOVES, L.; NIJHOLT, A. (edd.) (1994). Speech and Language Engineering, Proc. 8th Twente Workshop on Language Technology, Enschede, Universiteit Twente, Faculteit Informatica, 65-74.

HELBIG, H.; SCHULZ, M. (1997). „Knowledge Representation with MESNET – A Multilayered Extended Semantic Network.“ In: Working Notes 1997 AAAI Spring Symposium on Ontological Engineering, Stanford/CA, 64-72.

HELLER, K. (1996). IDS Sprachreport Extraausgabe: Rechtschreibreform. Online. Internet. 3.7.1996. *http://www.ids-mannheim.de/pub/reform.html*.

HENSCHEID, E.; LIEROW; C.; MALETZKE, E.; POTH, Ch. (1985). Dummdeutsch. Ein satirisch-polemisches Wörterbuch. Frankfurt am Main: Fischer Taschenbuch.

HERCZEG, M. (1994) Software-Ergonomie. Bonn et al.: Addison-Wesley.

HOHNHOLD, I. (1990). Übersetzungsorientierte Terminologiearbeit. Stuttgart: InTra.

HUCK, Dirk J. (1996). Entwicklung eines Datenbanksystems zur maschinellen Generierung normierter Schlagwortbestände. Diplomarbeit, FH Köln, FB Nachrichtentechnik, November 1996.

HUMPHREYS, L. et al. (1992). EUROLANG, Cross References Model. Eurolang Report, Site.

ISO 8879 (1986). Information Processing, Text and Office Systems, Standard Generalized Markup Language (SGML). 1. Aufl. International Organization for Standardization (ISO), Genf, Schweiz, 15. Oktober 1986.

ISSING, L. J.; KLIMSA, P. (edd.) (1995). Information und Lernen mit Multimedia. Weinheim: Psychologie Verlags Union.

JÄGER, T. (1996). "OCR and Voting Shell Fulfilling Specific Text Analysis Requirements." In: Proc. Fifth Annual Symposium on Document Analysis and Information Retrieval. Las Vegas/NA, 287-302.

JAUß, S. (1996). Erkennung und corpusbasierte Extraktion von Fachterminologie unter Einbeziehung der Terminologiedatenbank *Interfass* der Mercedes-Benz AG. Diplomarbeit, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung – Computerlinguistik.

KAPLAN, N.; MOULTHROP, S. (1993) „Seeing through the Interface: Computers and the Future of Composition." In: LANDOW, G. P.; DELANEY, P. (edd.) (1993). The Digital Word. Text-Based Computing in the Humanities. Cambridge/MA & London: The MIT Press, 253-270.

KASPER, E. & R. (1989) Objektorientiertes Programmieren in Smalltalk. Würzburg: Vogel.

KEENAN, F. G.; EVETT, L. J.; WHITROW, R. J. (1991). "A large vocabulary stochastic syntax analyser for handwriting recognition." In: International Conference on Document Analysis and Recognition. Saint-Malo, France, 74-82.

KINTSCH, W., VAN DIJK, T. A. (1983). Strategies of Discourse Comprehension. London: Academic Press.

KISS, T.; FOX, D.; GEURTS, B.; GEDIGA, G.; HAMILTON, S.; KRÜGER, A.; MURPHY, M. (1997). Final Report on RECALL: Repairing Errors in Computer-Assisted Language Learning. RECALL Deliverable 40. Publicly available from the Commission of the European Communities. Luxembourg.

KJELLMER, G. (1994). A Dictionary of English Collocations. Based on the Brown Corpus. 3 Volumes. Oxford 1994.

KÖLZER, A. (1997). *Lexana* – Ein System zur Lexikon- und Grammatikanalyse für Kategoriale Unifikationsgrammatiken. Diplomarbeit, Universität Koblenz–Landau, Abt. Koblenz, Februar 1997.

KRASHEN, S. D. (1995). Principles and practice in Second Language Acquisition. Oxford: Pergamon.

KRAUSE, J.; HERFURTH, M.; MARX, J. (edd.) (1996). „Herausforderungen an die Informationswissenschaft – Informationsverdichtung, Informationsbewertung und Datenvisualisierung", Proc. 5. Internationales Symposium für Informationswissenschaft, Humboldt-Universität zu Berlin, Oktober 1996. Konstanz: UVK Informationswissenschaft.

KRISHNAMURTHY, R. (1996). „The Data is The Dictionary: Corpus at the Cutting Edge of Lexikography." In: Proc. 4th International Conference on Computational Lexicography, COMPLEX 1996, Budapest.

KRÜGER, A. (1996). Diagnosis Module – Functional Specification. RECALL Deliverable 20.

KRÜGER, A.; DITTMANN, H.; HUBER, M.; KRUMEICH, A.; TEIKEN, W. (1996). Diagnosis Algorithm. RECALL Deliverable 25.

KRÜGER, A.; GEURTS, B. (1997). Survey of Errors, their Classification and Potential Causes. RECALL Deliverable 10.

KRÜGER, Katja (1996). Corpusbasierter Aufbau von Glossaren – ein integriertes System zur Extraktion von Termkandidaten aus deutschen und englischen Fachtexten. Diplomarbeit, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung – Computerlinguistik.

KUHLEN, R. (1991). Hypertext. Ein nicht-lineares Medium zwischen Buch und Wissenschaft. Berlin et al.: Springer.

KUHLEN, R. (1995). Informationsmarkt. Chancen und Risiken der Kommerzialisierung von Wissen. Konstanz: UVK [Schriften zur Informationswissenschaft; 15].

LALANDE, J.-Y. (1997). Verbstellung im Deutschen und Französischen. Unter Anwendung eines CAD-basierten Expertensystems. Tübingen: Niemeyer [= Linguistische Arbeiten 365].

LANGER S.; MAIER, P.; OESTERLE, J. (1996). „CISLEX – An Electronic Dictionary for German: Its Structure and a Lexicographic Application." In: KIEFER, F. et. al (edd.) (1996). Papers in Computational Lexicography, COMPLEX 96.

LANGER, S. (1996). Selektionsklassen und Hyponymie im Lexikon. Diss. phil. Universität München.

LAUREL, B. (1993). Computers as Theatre. Reading/MA et al.: Addsion-Wesley.

LEE, K. (1995). "Recursion Problems in Concatenation: A Case of Korean Morphology." In: Proc. PACLIC 10, 10th Pacific-Asian Conference on Language, Information and Computation, Hong Kong.

LEHR, A. (1996). Kollokationen und maschinenlesbare Korpora. Ein operationales Analysemodell zum Aufbau lexikalischer Netze. Tübingen 1996.

LEMNITZER, L. (1997). Extraktion komplexer Lexeme aus Textkorpora. Tübingen (erscheint 1997).

LENDERS, W. (1995). „The Semantic Structure of Definitions and their Transformation into a Computer's Memory." In: Proc. 7th International Conference on Computers and the Humanities.

LEVISON, Stephen C. (1983). Pragmatics. CUP.

LIER, B. (1996). Entwicklung eines Datenbanksystems zum Online-Zugriff aus die Katalogbestände der FH-Bibliothek im Internetdienst WWW. Diplomarbeit, FH Köln, FB Nachrichtentechnik, Oktober 1996.

LORENZ, O. (1997). Automatische Wortformerkennung für das Deutsche im Rahmen von MALAGA. Magisterarbeit, Friedrich-Alexander-Universität Erlangen-Nürnberg, Abteilung für Computerlinguistik.

LUCAS, R. (1997). ProDBI ODBC Interface for Quintus Prolog. Warwickshire: Keylink.

MAIER, P. (1995). Lexikon und automatische Lemmatisierung. Universität München, Centrum für Informations- und Sprachverarbeitung (CIS), CIS-Bericht 95-84.

MAJERUS, M. E. N. (1994). Ladybirds. London: Harper Collins.

MANN, William C.; THOMPSON, Sandra A. (1988). "Rhetorical Structure Theory: Toward a functional theory of text organization." In: Text. An interdisciplinary journal for the study of discourse 8(1/2) (1988), 243-281.

MARTIF ISO FDIS 1620: Terminology – Computer Applications – Data Categories.

MATER, E. (1966-1972). Deutsche Verben. Leipzig: Bibliographisches Institut.

MCCORD, M.; BERNTH, A. (1993). The LMT Book. IBM Research Technical Report.

MCDONALD, D. (1994). „Trade-offs between Syntactic and Semantic Processing in the Comprehension of Real Text." In: Intelligent Multimedia Information Retrieval Systems and Management. Proc. RIAO 94, New York, 94-105.

MCGLASHAN, S., ANDRY, F.; NIEDERMAIR, G. (1990). A Proposal for SIL. Technischer Bericht ESPRIT Project P2218 SUNDIAL, Dezember 1990.

MECKLENBURG, K.; HEISTERKAMP, P.; HANRIEDER, G. (1995). „A Robust Parser for Continuous Spoken Language Using Prolog." In: Proc. 5th International Workshop on Natural Language Understanding and Logic Programming (NLULP 95). Lissabon, 127-141.

MELBY, A. (1988). „The ideal lexical database system: A checklist of desirable features." In: VASCONCELLOS, M. (ed.) (1988). Technology as translation strategy. Binghamton: State University of New York at Binghamton.

MENZEL, W. (1995). „Robust Processing of Natural Language." In: WACHSMUTH, I.; ROLLINGER, C.-R.; BAUER, W. (edd.) (1995). KI-95: Advances in Artificial Intelligence – 19th Annual German Conference on Artificial Intelligence, Bielefeld. Berlin: Springer, 19-34.

MERTENS, A.; SCHULZ, M.; HELBIG, H. (1995). „Analyse mit Wortagenten im NLP-System LINAS." In: HITZENBERGER, L. (ed.) (1995). Angewandte Computerlinguistik. Proc. 9. GLDV-Jahrestagung, Regensburg, März 1995. Hildesheim/New York: Olms, 63-75.

MISGELD, Wolfgang D. (1991). ORACLE für Profis. München/Wien: Hanser.

MODIANO, N. (1992). MULTILEX: Definition of the Standard. Linguistic Architecture. Report Esprit Project 5304 MULTILEX.

MÖSSENBÖCK, H. (1992). Objektorientierte Programmierung in Oberon-2. Berlin: Springer.

MUC-4 = Proc. Fourth Message Understanding Conference (1992). San Mateo/CA: Morgan Kaufmann.

MURPHY, M. (1997). Learner Module – Design Specification. RECALL Deliverable 29.

MURPHY, M.; MCTEAR, M. J.; FITZPATRICK, F. J. (1996). Learner Model – Requirements Specification and Functional Specification. RECALL Deliverable 11.

NIELSEN, J. (1995). Multimedia and Hypertext. The Internet and Beyond. Boston: AP Professional.

NORMAN, D. A.; RUMELHARDT, D. E. (1975). Explorations in Cognition. San Francisco/CA: Freeman.

PAICE, C. D. (1990). „Constructing Literature Abstracts by Computer: Techniques and Prospects.“ In: Information Processing and Management 26(1), 171-186.

PAIJMANS, H. (1994). Relative Weights of Words, *http://pi0959.kub.nl:2080/Paai/Stinfon/Stinfon.html*.

PECKHAM, J. (1993). „A New Generation of Spoken Dialogue Systems: Results and Lessons from the SUNDIAL Project.“ In: Proc. 3rd European Conference on Speech Communication and Technology (EUROSPEECH '93), Berlin, September 1993, Vol.1, 33–40.

POLLOCK, J.-Y. (1989). "Verb Movement, Universal Grammar and the Structure of IP." In Linguistic Inquiry 20(3) (1989), 365-424.

QUASTHOFF, U. (1998). „Tools for Automatic Lexicon Maintenance: Acquisition, Error Correction and the Generation of Missing Values.“ In: RUBIO, A. et al. (edd.) (1998). Proc. of the First International Conference on Language Resources and Evaluation, Granada, May 1998. Paris: European Language Resource Association, 853-856.

QUASTHOFF, U.; WOLFF, Ch. (edd.) (1997[2]). Informations-CD-ROM des Instituts für Informatik. Universität Leipzig, Institut für Informatik Leipzig: Universität Leipzig, Institut für Informatik.

RABINER, L. R. (1988). "Mathematical Foundations of Hidden Markov Models." In: NIEMANN, H.; LANG, M.; SAGERER, G. (edd.) (1988). Recent Advances in Speech Understanding and Dialog Systems. NATO ASI Series F, Vol. 46, 183-205.

REARICK, Th. C. (1991). Automating the Conversion of Text into Hypertext. In: BERK, E.; DEVLIN, J. (edd.). Hypertext / Hypermedia Handbook. New York et al.: Intertext Publications McGraw-Hill, 113-140.

REINHARDT, W.; KÖHLER, K.; Neubert, G. (1992). Deutsche Fachsprache der Technik. Hildesheim/New York: Olms.

REIS, M. (1985) "Satzeinleitende Strukturen im Deutschen. Über COMP, Haupt- und Nebensätze, w-Bewegung und die Doppelkopfanalyse." In. ABRAHAM, W. (ed.) (1985). Erklärende Syntax des Deutschen, Tübingen: Narr [= Studien zur deutschen Grammatik 25], 271-311.

REISER, M.; WIRTH, N. (1992). Programming in Oberon: Steps beyond Pascal and Modula. New York: The ACM Press.

RINER, R. (1991). "Automated Conversion." In: BERK, Emily; DEVLIN, Joseph (edd.). Hypertext / Hypermedia Handbook. New York et al.: Intertext Publications McGraw-Hill, 95-111.

RITZKE, Joh. (1996). OTELO Lecicon Database. OTELO Report.

RITZKE, Joh. (1997). Common Lexical Resource Format: Format Specifications. OTELO Report.

ROHRBACHER, B. W. (1994). The Germanic VO Languages and the Full Paradigm: A Theory of V to I Raising. Amherst/MA.

ROLSHOVEN, J. (1991). "GB und sprachliche Informationsverarbeitung mit LPS." In ROLSHOVEN, J.; SEELBACH, D. (edd.) (1991). Romanistische Computerlinguistik Theorien und Implementationen. Tübingen: Niemeyer [= Linguistische Arbeiten 266], 133-158.

ROLSHOVEN, J. (1996a). "Lexikalisches Wissen in der maschinellen Übersetzung." In BLUMENTHAL, P.; ROVERE, G.; SCHWARZE, C. (edd.) (1996). Lexikalische Analyse romanischer Sprachen. Tübingen: Niemeyer [= Linguistische Arbeiten 353], 85-100.

ROLSHOVEN, J. (1996b). "Modularisierung oder Objektorientierung?" Ms., Universität zu Köln, Sprachliche Informationsverarbeitung.

ROLSHOVEN, J. (1997). "Bewegungen: GB und romanische Sprachwissenschaft." In DAHMEN, W. et al. (edd.) (1997). Neuere Beschreibungsmethoden der Syntax romanischer Sprachen. [= Romanistisches Kolloquium XI]. Tübingen: Narr.

RSREFORM (o.J.). Deutsche Rechtschreibung: Regeln und Wörterverzeichnis. Amtliche Regelung. *http://www.ids-mannheim.de/grammis/reform/inhalt.html*

RUGE, G. (1995). Wortbedeutung und Termassoziation. Methoden zur automatischen semantischen Klassifikation. Hildesheim/New York: Olms.

SALTON, G., MCGILL, M. J. (1983). Introduction to Modern Information Retrieval. New York et al.: McGraw-Hill.

SANDIG, B. (1989). "Stilistische Mustermischungen in der Gebrauchssprache." In: Zeitschrift für Germanistik 10(1) (1989), 133-150.

SANDIG, B. (1997). "Formulieren und Textmuster. Am Beispiel von Wissenschaftstexten." In: JAKOBS, E.-M.; KNORR, D. (edd.) (1997). Schreiben in den Wissenschaften. Frankfurt am Main: Peter Lang, 25-44.

SARRE, F., GÜNTZER, U. (1990). "Einsatz des Hypertext-Systems "HyperMan" für Online-Datenbankmanuale." In: GLOOR, Peter A.; STREITZ, Nobert A. (edd.). Hypertext und Hypermedia. Berlin et al.: Springer [= Informatik-Fachberichte], 112-123.

SCHANK, R. C. (1987). „Language and Memory.“ In: SPARCK JONES, K. et al. (edd.) (1987). Natural Language Processing, 170-191.

SCHANK, R. C. (ed.) (1975). Conceptual Information Processing. Amsterdam: North-Holland.

SCHILLER, A. (1994). Deutsche Flexions- und Kompositionsmorphologie auf 2-Ebenen-Basis. Technischer Bericht, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung – Computerlinguistik.

SCHMID, H. (1994a). „Part-of-Speech Tagging with Neural Networks.“ In: Proc. 15 International Conference on Computational Linguistics, COLING 94, Kyoto, Vol. I, 406-411.

SCHMID, H. (1994b). „Probabilistic Part-of-Speech Tagging Using Decision Trees.“ In: Proc. Conference on New Methods in Language Processing 94, Manchester.

SCHMITZ, K.-D. (1994a). "Sprachliche Einheiten und deren Verwaltung in Terminologieverwaltungssystemen." In: GRINSTED, Annelise; MADSEN, Bodil Nistrup (edd.). Festskrift til Gert Engel i anledning af hans 70-års fødselsdag. Frederiksberg: Samfundslitteratur.

SCHMITZ, K.-D. (1994b). Verarbeitung fachsprachlichen Wissens in Terminologieverwaltungssystemen. In: SPILLNER, Bernd (ed.). Fachkommunikation. Frankfurt am Main: Peter Lang.

SCHNEIDER, M. (1997). Ein robuster Nominal- und Partizipialphrasen-Parser für das Deutsche. Magisterarbeit, Friedrich-Alexander-Universität Erlangen-Nürnberg, Abteilung für Computerlinguistik.

SCHNEIDER, R. (1997). Datenbankintegration und Navigationsangebote in einem hypermedialen Informationssystem zur deutschen Verbvalenz. Magisterarbeit, Universität Trier, Fachbereich Computerlinguistik.

SCHNOTZ, W. (1994). Aufbau von Wissensstrukturen. Untersuchungen zur Kohärenzbildung bei Wissenserwerb mit Texten. Weinheim: Beltz.

SCHOTT, G. (1978). „Automatische Deflexion deutscher Wörter unter Verwendung eines Minimalwörterbuchs.“ In: Sprache und Datenverarbeitung I/1978, 62-77.

SCHULMEISTER, R. (1996). Grundlagen hypermedialer Lernsysteme. Bonn et al.: Addison-Wesley.

SCHULZE, B. M. (1996). MP user manual. Universität Stuttgart, Institut für maschinelle Sprachverarbeitung – Computerlinguistik.

SCHULZE, B. M.; CHRIST, O. (1994). The CQP User's Manual. Universität Stuttgart, Institut für maschinelle Sprachverarbeitung – Computerlinguistik.

SCHWALL, U.; THURMAIR, Gr. (1997). „From METAL to T1, Systems and Components for Machine Translation Applications.“ In: Proc. MT-Summit VI, San Diego/CA.

SCHWARZ, Ch. (1990). „Automatic Syntactic Analysis of Free Text.“ In: Journal of the American Society for Information Science 41(6) (1990), 408-417.

SEARLE, J. (1986^2). Sprechakte. Ein sprachphilosophischer Essay. Frankfurt am Main: Suhrkamp [= stw Bd. 458].

SHIEBER, St. (1986). An Introduction to Unification-Based Approaches to Grammar. Stanford/CA: CSLI Publications [= CSLI Lecture Notes 4].

SILBERZTEIN, M. (1989). Dictionnaires électroniques et reconnaissance lexical automatique. Masson. Paris.

SIMPSON, A.; FRASER, N. (1993). "Black Box and Glass Box Evaluation of the SUNDIAL System." In: Proc. 3rd European Conference on Speech Communication and Technology (EUROSPEECH'93), Berlin, 1993, 1423–1426.

SJÖGREEN, Ch. (1998). "Drug Terminology." In: Proc. 11th Nordic Conf. on Computational Linguistics, The University of Copenhagen, 1998.

SMADJA, F. (1993). „Retrieving Collocations from Text.“ In: Computational Linguistics 19(1) (1993), 143–177.

SMALL, St. (1987). „A Distributed Word-Based Approach to Parsing.“ In: BOLC, L. (ed.) (1987). Natural Language Parsing Systems. Berlin: Springer, 161-201.

SNPF (ed.) (1996). Basfakta om narkotika. Stockholm [ISBN 91-87514-03-6].

STEELE, J. (ed.). Meaning-Text Theory. Ottawa: University of Ottawa Press.

STEINMETZ, R. (1995). Multimedia-Techologie. Berlin et al.: Springer [korr. ND 1995].

STEINMÜLLER, W. (1994). Informationstechnologie und Gesellschaft. Einführung in die angewandte Informatik. Darmstadt: Wissenschaftliche Buchgesellschaft.

STERLING, L.; SHAPIRO, E. (1994[2]). The Art of Prolog. Cambridge/MA, London: The MIT Press.

STORRER, A. (1995). „Die Grammatik mit der Maus – Konzeption eines multimedialen Informationssystems zur deutschen Grammatik.“ In: HITZENBERGER, L. (ed.) (1995). Angewandte Computerlinguistik. Proc. 9. GLDV-Jahrestagung, Regensburg, März 1995. Hildesheim/New York: Olms, 291-303.

STORRER, A. (1997). "Grammatikographie mit Neuen Medien: Erfahrungen beim Aufbau eines grammatischen Informationssystems." In: Zeitschrift für Literaturwissenschaft und Linguistik 106 (1997), 46-77.

STORRER, A. (1998). "Hypermedia und Grammatikographie." In: STORRER & HARRIEHAUSEN (1998), 25-51.

STORRER, A.; HARRIEHAUSEN, B. (edd.) (1998). Hypermedia für Lexikon und Grammatik. Tübingen: Narr.

STRECKER, B. (1998). "Hypertext: Chance und Herausforderung für die Grammatikschreibung." In: STORRER & HARRIEHAUSEN (1998), 19-24.

STREITZ, N. A.; HERMANN, J. (1990). "Elaborating Arguments: Writing Learning, and Reasoning in a Hypertext Based Environment for Authoring." In: JONASSEN, D. H.; MANDL, H. (edd.) (1990). Designing Hypermedia for Learning. Heidelberg et al.: Springer, 407-439.

THURMAIR, Gr. (1997a). "Multilingual Information Processing. The AVENTINUS System." In: Proc. FBINAA Conference, Berlin.

THURMAIR, Gr. (1997b). „Exchange Interfaces for Translation Tools.“ In: Proc. MT-Summit VI, San Diego/CA, 1997.

THURMAIR, Gr., NIEDERMAIR, G., SCHWARZ, Ch., WESSEL, A. (1986). „REALIST (Retrieval Aids by Statistics and Linguistics). Systemkonzeption.“ In: SCHWARZ, Ch., THURMAIR, Gr. (edd.) (1986). Informationslinguistische Texterschließung. Hildesheim/New York: Olms.

THURMAIR, Gr., WOMSER-HACKER, Chr. (1996). „Multilingualität im wissensbasierten Faktenretrieval. In: KRAUSE et al. (1996), 121-131.

THURMAIR, Gr.; RITZKE, Joh.; MCCORMICK, S. (1998). "The Open Lexicon Interchange Format OLIF." In: Proc. TAMA, Wien.

UNDERWOOD, N., NAVARRETTA, C. (1997). Towards a Standard for the Creation of Lexica. ELRA Report.

WAHLSTER, W. (1993). " Verbmobil – Translation of Face-To-Face Dialogs." In: MT Summit IV, Kobe, Japan, Juli 1993.

WAHLSTER, W. et al. (1996). Homepage: Informationen zu Verbmobil. World Wide Web. *http://www.coli.uni-sb.de/~vm/*.

WAHRIG, G. (ed.) (1986). Deutsches Wörterbuch. München: Mosaik.

WEBELHUTH, G. (1995). Government and Binding Theory and the Minimalist Program. Cambridge/MA: Blackwell.

WEHRLE, H.; EGGERS, H. (1967[13]). Deutscher Wortschatz. Stuttgart: Klett.

WEIGEL, A.; BAUMANN, S.; ROHRSCHNEIDER, J. (1995). "Lexical Postprocessing by Heuristic Search and Automatic Determination of the Edit Costs." In: Fourth International Conference on Document Analysis and Recognition. Montreal, Canada, 857-860.

WEINRICH, H. (1993). Textgrammatik der deutschen Sprache. Mannheim: Dudenverlag.

WENZEL, C.; BAUMANN, S.; JÄGER, T. (1996). "Advances in Document Classification by Voting of Competitive Approaches." In: IAPR Workshop on Document Analysis Systems. Malvern/PA, 352-372.

WETZEL, Ch. (1996). Erstellung einer Morphologie für Italienisch in MALAGA. Studienarbeit, Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Programmier- und Dialogsprachen.

WIEGAND, H. E. (1987). "Wörterbuchartikel als Text." In: HARRAS, G. (edd.) (1987). Das Wörterbuch – Artikel und Verweisungsstrukturen. Jahrbuch 1987 des Instituts für deutsche Sprache. Düsseldorf: Schwann, 30-120.

WIEGAND, H. E. (1997). "Printed language dictionaries and their standardization: Notes on the progress toward a general theory of lexicography." In: HOCK, H. H. (ed.) (1997). Historical, Indo-European, and Lexicographical Studies. A Festschrift for Ladislav ZGUSTA on the Occasion of his 70th Birthday. Berlin/New York: de Gruyter, 319-393.

WILDER, C. (1995). "Derivational Economy and the Analysis of V2." In: FAS Papers in Linguistics 1 (1995), 117-156.

WILLÉE, G. (1979). "LEMMA – Ein Programmsystem zur automatischen Lemmatisierung deutscher Wortformen." In: Sprache und Datenverarbeitung I/1979, 45-60.

WILLÉE, Gerd (1982). „Das Programmsystem Lemma2 – Eine Weiterentwicklung von ‘Lemma’.“ In: IKP-Arbeitsberichte Nr.2, Bonn: IKP, 1-47.

WOMSER-HACKER, CH. (1996). Das MIMOR-Modell. Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval. Habilitationsschrift, Universität Regensburg, Informationswissenschaft.

WOTJAK, Barbara (1992). Phraseolexeme in System und Text. Tübingen, 1992.

ZEEVAT, H. (1988). „Combining Categorial Grammar and Unification.“ In: REYLE, U.; ROHRER, Ch. (edd.) (1988). Natural Language Parsing and Linguistic Theories. Dordrecht: Reidel, 202–229.

ZEEVAT, H.; KLEIN, E.; CALDER, J. (1987). "Unification Categorial Grammar." In: HADDOCK, KLEIN & MORRILL (edd.) (1987). Categorial Grammar, Unification Grammar and Parsing. University of Edinburgh, Centre of Cognitive Science [= Working Papers in Cognitive Science, Vol. 1], 195–222.

ZELLWEGER, P. T. (1992). „Toward a Model for Active Multimedia Documents.“ In: BLATTNER & DANNENBERG (1992), 39-52.

ZETTEL, W. (1996). Indexierung auf der Basis formaler Texteigenschaften am Beispiel sozialwissenschaftlicher Volltexte. Magisterarbeit, Universität Regensburg, Informationswissenschaft, November 1996.

ZIFONUN, G.; HOFFMANN, L.; STRECKER, B. (1997). Grammatik der deutschen Sprache. Berlin/New York: de Gruyter.

ZWART, J.-W. (1994). "Introduction." In ABRAHAM, W.; ZWART, J.-W. (edd.) (1994). Minimalism and Kayne's Asymmetry Hypothesis [= Groninger Arbeiten zur Germanistischen Linguistik 25], 1-17.