universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

„Implementierung einer CALL-Anwendung unter Verwendung von NLP-Methoden mit dem Schwerpunkt auf semantischer Annotation“

Verfasser

Georg Pitschmann

angestrebter akademischer Grad

Magister der Philosophie (Mag. phil.)

Wien, im November 2008

Studienkennzahl lt. Studienblatt:	A 328
Studienrichtung lt. Studienblatt:	Sprachwissenschaft
Betreuer:	Ass.-Prof. Dipl.-Ing. Dr. Ernst Buchberger

Inhaltsverzeichnis

Vorbemerkungen

Auf die Verwendung des Binnen-I wurde in dieser Arbeit verzichtet. Alle Formen sind, falls aus dem Kontext nicht anders ersichtlich, geschlechtsneutral zu verstehen.

Sowohl im laufenden Text als auch in den verwendeten Beispielen wurden ungrammatische bzw. semantisch nicht akzeptable Satzkonstruktionen mit einem Asterisk (*) markiert. Satzkonstruktionen, die nur eingeschränkt akzeptabel sind, wurden mit einem Fragezeichen (?) markiert.

Kapitel 1

Einleitung

Bereits bald nach dem ersten Aufkommen von Computern wurde begonnen, diese im Bereich der Sprachdidaktik zu nutzen. Ausgehend von den ersten Versuchen in den sechziger Jahren des letzten Jahrhunderts entwickelte sich *Computer Assisted Language Learning*, kurz *CALL*, zu einer Disziplin mit großem wirtschaftlichem Potenzial. Bereits 1994 machten ein Fünftel des gesamten europäischen Multimediamarktes Anwendungen aus dem Bereich des CALL aus (Nerbonne 2003: 671). Während die ersten CALL-Anwendungen aufgrund der noch eingeschränkten technischen Möglichkeiten auf wenige teure universitäre Großprojekte beschränkt waren, reichen zwischenzeitlich bereits ein Computer sowie geringe Programmierkenntnisse aus, um entsprechende Anwendungen herzustellen. Dementsprechend groß ist die Anzahl an einschlägigen Angeboten, von kostenlosen Vokabelübungen im WWW bis hin zu kostspieligen Multimedia-Anwendungen auf CD-ROM kann auf ein breites Spektrum an Applikationen zurückgegriffen werden.

Um einen erwachsenen Schüler auf ein Niveau zu bringen, welches einfache Kommunikation in der Fremdsprache erlaubt, benötigt ein Lehrer unter Zuhilfenahme von traditionellen Methoden 60 bis 100 Stunden (FSI 1973, zitiert nach Nerbonne 2002), darüber hinaus gehende sprachliche Fähigkeiten verlangen einen deutlich höheren Aufwand. Die Überlegung, zumindest einen Teil dieses Aufwands nach Möglichkeit durch maschinelle Anwendungen abzudecken, ist naheliegend, wobei der Beitrag des Computers nicht als Ersatz, sondern als Ergänzung zur Arbeit des Lehrers gesehen werden muss. Der den Lehrer ersetzende Supercomputer ist nach wie vor reine Fiktion und wird dies, wirft man einen Blick auf die derzeitigen Möglichkeiten von CALL-Anwendungen, noch lange Zeit bleiben. CALL bietet jedoch durchaus Möglichkeiten, dem Lehrer zeitintensive Prozeduren wie die Erstellung und Kontrolle von Übungen abzunehmen oder zumindest zu erleichtern.

CALL stellt ein relativ junges, interdisziplinäres Feld dar, wobei Psychologie, Künstliche Intelligenz, Instruktionsdesign, Mensch-Computer-Interaktion sowie Computerlinguistik als die wichtigsten Teildisziplinen gesehen werden können (Levy 1997: 48). Die Psychologie liefert u.a. Erkenntnisse aus dem Zweitspracherwerb, für die systematische Planung, Entwicklung und Auswertung der Lernunterlagen sind Methoden aus dem Bereich des Instruktionsdesigns notwendig. Für die benutzergerechte Umsetzung, eine ansprechende Gestaltung sowie eine optimale Gebrauchstauglichkeit der entsprechenden Programme ist die Mensch-Computer-Interaktion, eine Teildisziplin der Informatik, zuständig. Nachdem sich die Computerlingusitk mit der Verarbeitung von natürlicher Sprache beschäftigt, sollte man annehmen, dass ihr eine wichtige Stellung innerhalb des CALL zukommt. Tatsächlich ist der Beitrag der Computerlinguistik im Bereich des CALL jedoch bisher ziemlich gering. Anwendungen wie Parsing, Tagging sowie Methoden aus der maschinelle Übersetzung oder Computer Aided Translation (CAT) werden in den seltensten Fällen in CALL-Applikationen verwendet. In der Regel werden die in CALL-Anwendungen eingesetzten sprachlichen Daten vorgefertigt in Datenbanken gespeichert und während des Programmablaufes der jeweiligen Übungssituation entsprechend abgerufen.

Das dieser Arbeit zugrundeliegende Projekt mit dem Projektnamen *SCall* (*Semantisches CALL*), welcher in weiterer Folge verwendet wird, stellt den Versuch dar, eine CALL-Anwendung zu entwerfen sowie umzusetzen, die auf Methoden aus der Computerlinguistik basiert, um zu zeigen, dass derartige CALL-Anwendungen trotz der derzeitigen Situation durchaus sinnvoll umgesetzt werden können. Ausschlaggebend für die Wahl des Themas war ein im Zuge eines Praktikums am Institut für Medizinische Kybernetik und Artificial Intelligence des Zentrums für Hirnforschung der Medizinischen Universität Wien realisiertes CALL-Programm mit dem Namen *Cica*[1]. *Cica* wurde als Hilfsprogramm zum Training von in einer Sprache immer wiederkehrenden grammatikalischen Strukturen entwickelt. Anhand von benutzerseitig eingegebenen Grammatikstrukturen sowie ergänzenden, mit grammatikalischen Informationen annotierten Wortlisten generiert das Programm Übungsaufgaben. Ein (vereinfachtes) Beispiel für die Mustereingabe sowie daraus resultierende Beispielphrasen (in diesem Fall Übungen zur deutschen Adjektiv-Nomen-Kongruenz) wird in Tabelle 1.1 gezeigt.

Werden die für die Generierung erforderlichen Daten gewissenhaft eingegeben, generiert *Cica* semantisch korrekte Übungsbeispiele und erfüllt somit seinen primären Zweck. Die Problematik der Anwendung liegt jedoch in der fehlenden Berücksichti-

[1]Bis vor kurzem existierte eine Homepage für dass Projekt (http://www.cica.dd.vu), die jedoch nicht mehr online ist und daher davon ausgegangen werden kann, dass *Cica* zur Zeit nicht weiterentwickelt wird.

Eingabemuster	Vokabular	Generierung
das hohe Haus	klein	der hohe Tisch
	breit	die kleinen Häuser
	Tisch	das hohe Regal
	Regal	der breite Tisch
		usw.

Tabelle 1.1: Eingabe- und Generierungsmuster entsprechend der Funktion von Cica.

gung jeglicher Semantik in den generierten Satzstrukturen. Folglich hat die Erweiterung des in Tabelle 1.1 verwendeten Vokabulars durch beispielsweise das Nomen *Katze* semantisch zweifelhafte Generierungen wie *die hohe Katze* oder *die breite Katze* zur Folge. Ergo liegt es am Benutzer, dafür Sorge zu tragen, dass in den jeweiligen Übungen nur Vokabeln verwendet werden, die semantisch korrekte Konstruktionen zulassen. Und genau dieses Problem soll in SCall durch die Berücksichtigung der Semantik während des Generierungsprozesses verhindert werden. Die grundsätzliche Konzeption von SCall entspricht jedoch, mit Ausnahme der semantischen Erweiterungen, in etwa jener von Cica: ausgehend von Eingaben des Benutzers werden Übungsbeispiele für den Unterricht in den unterstützten Sprachen generiert.

Während *Cica* ursprünglich für Deutsch-Ungarisch konzipiert wurde und aufgrund des modularen Aufbaus für die Verwendung mit weiteren Sprachen vorgesehen war, wurde SCall für die Verwendung im Schwedischunterricht für deutschsprachige Schüler und vice versa umgesetzt. Schwedisch wurde einerseits gewählt, da der Großteil der erhältlichen CALL-Anwendungen für Sprachen mit großer Sprecheranzahl wie etwa Englisch, Französisch, Spanisch etc. konzipiert wurde bzw. wird und für (relativ) kleine Sprachen wie eben Schwedisch bisher wenige einschlägige Anwendungen existieren, andererseits standen für Schwedisch sämtliche für die Umsetzung des Programms notwendigen Ressourcen (Korpus, Wörterbuch Deutsch-Schwedisch) und Applikationen (Parser, morphologischer Parser) zur Verfügung. Eine Expansion von SCall auf andere Sprachen ist prinzipiell möglich, wird jedoch nicht beabsichtigt.

SCall ist in zwei grundsätzlich unabhängige Prozesse unterteilt, einen Analyse- und einen Generierungsprozess. Während des *Analyseprozesses* werden sämtliche für die Generierung notwendigen lexikalischen Daten auf Basis von vom Benutzer eingegebenen Beispielsatzpaaren ermittelt. Diese Sätze werden während eines Analyseverfahrens, bestehend aus Parsing, Part-of-speech-Tagging (PoS-Tagging) sowie Lexical-Lookup, verarbeitet. Durch das Parsen wird die Phrasenstruktur der zugrundeliegenden Sätze ermittelt, das PoS-Tagging ermittelt sowohl die Wortar-

ten als auch die Grundformen bzw. Lemmata der Wörter. Der Lexical-Lookup-Prozess vergleicht die Lemmata der Satzpaare und sucht in einem Wörterbuch nach Entsprechungen. Anschließend kann der Benutzer mit Hilfe eines Interface die analysierten Sätze kontrollieren und gegebenenfalls Fehler korrigieren sowie die Semantik bearbeiten. Während des *Generierungsprozesses* werden, ausgehend von den im Zuge des Analyseprozesses ermittelten Daten, Sätze für die jeweilige Sprache generiert, die in weiterer Folge als Grundlage für Übungen dienen.

Der Schwerpunkt des Programms liegt auf der Umsetzung einer geeigneten semantischen Annotierung des Lexikons. Während des Analyseprozesses wird die Semantik durch die Annotierung der Nomen durch sog. Selektionsrestriktionen sichergestellt. Selektionsrestriktionen sind im weitesten Sinne Labels, welche die Verwendung von Wörtern auf bestimmte semantische Umgebungen beschränken. In SCall dienen die Konzepte aus der lexikalischen Ressource *WordNet* als Selektionsrestriktionen. Aus einer Reihe von vom Programm vorgeschlagenen, aus *WordNet* ermittelten Restriktionen wählt der Benutzer eine Restriktion aus, die im Lexikon gespeichert und während der Generierung berücksichtigt wird. Um vom Programm generierte Strukturen auf semantische Korrektheit zu prüfen, wurde zusätzlich zu den Selektionsrestriktionen ein statistischer, korpusbasierter Ansatz gewählt. Anhand eines Korpus, in diesem Fall das World Wide Web (WWW), werden Teilstrukturen aus den generierten Sätzen auf deren Vorkommen untersucht und nur bei ausreichender Frequenz als semantisch korrekt postuliert. Der Hauptteil der Arbeit liegt somit auf zwei semantischen Ansätzen: zum einen der Annotation von Lexikoneinträgen durch Selektionsrestriktionen, zum anderen auf der korpusbasierten Kontrolle von vom Programm generierten sprachlichen Strukturen.

Die Arbeit ist in zwei Teile gegliedert: einen theoretischen sowie einen praktischen Teil. Im theoretischen Teil, der die Kapitel 2-4 umfasst, werden die theoretischen Grundlagen diskutiert, während im zweiten, praktischen Teil der Arbeit der Aufbau sowie die Funktionsweise von SCall gezeigt werden. Im zweiten Kapitel *CALL und die problematische Beziehung zu NLP* wird die geschichtliche Entwicklung des Computer Assisted Language Learning beleuchtet. Weiters wird das nicht unproblematische Verhältnis zwischen CALL und Natural Language Processing thematisiert, gefolgt von einer Zusammenstellung einiger CALL-Programme mit NLP-Hintergrund. Im dritten Kapitel *Selektionsrestriktionen auf Basis von Konzepten aus Ontologien oder Taxonomien* wird das Konzept der Selektionsrestriktionen zur semantischen Annotation erörtert. Da die für den Einsatz in SCall gewählten Selektionsrestriktionen den Konzepten einer Ontologie entsprechen, wird ein kurzer Überblick über den Einsatz von Ontologien in der Wissensverarbeitung sowie über grundsätzliche Regeln zum Aufbau und zur Klassifikation von Ontologien gegeben.

Ferner werden Ontologien für den linguistischen Gebrauch thematisiert, mit einem Schwerpunkt auf der Mikrokosmos-Ontologie sowie WordNet. Im vierten Kapitel *Das WWW als Korpus zur statistischen Semantikkontrolle* wird der Ansatz zur statistischen Kontrolle von semantischen Konstruktionen vorgestellt. Als das der Kontrollprozedur zugrundeliegende Korpus wurde das WWW gewählt. Aus diesem Grund wird die Tauglichkeit des WWW als linguistisches Korpus untersucht und zu diesem Zweck das WWW mit speziell für die linguistische Verwendung konzipierten Korpora wie dem *Negra-Korpus* verglichen.

Im zweiten Teil der Arbeit werden der Aufbau sowie die Funktionsweise des Programms erläutert. In weiterer Folge werden die vor allem für die Benutzer wichtigen Funktionen des Graphischen Interface sowie der für die Generierung von Übungssätzen zuständige Programmteil erklärt.

Kapitel 2

CALL und die problematische Beziehung zu NLP

2.1 Computer Assisted Language Learning: Definition und geschichtliche Entwicklung

2.1.1 Definition

Unter dem Begriff *CALL* bzw. *Computer Assisted Language Learning* versteht man den Einsatz von Computern im Fremdsprachenunterricht. Levy (1997: 1) definiert den Begriff *CALL* folgendermaßen: „Computer-Assisted Langauge Learning [CALL] may be defined as 'the search for and study of applications of the computer in language teaching and learning'". Unter CALL kann eine relativ große Anzahl von Anwendungen, welche im Sprachunterricht zum Einsatz kommen, subsumiert werden. Diese Anwendungen reichen von der Verwendung von digitalen Videosequenzen über Dialogsysteme bis hin zu Programmen zur Rechtschreibkontrolle, womit CALL letztendlich ein riesiges, interdisziplinäres Gebiet abdeckt. Als die einflussreichsten Disziplinen für CALL nennt Levy (1997: 49) Psychologie, Künstliche Intelligenz, Computerlinguistik, Instruktionsdesign (Didaktisches Design) und Mensch-Computer-Interaktion.

Während das Akronym CALL zum Großteil auf den Begriff *Computer Assisted Language Learning* zurückgeführt wird (Levy 1997: 1), kommt in der Literatur mitunter auch der Begriff *Computer Aided Language Learning* vor (beispielsweise in Ehsani/Knodt 1998: 54). Interessanterweise wird auf den Umstand, dass das Akronym CALL auf zwei verschiedene Langformen referiert, keine Rücksicht genommen. Tatsächlich werden die beiden Begriffe scheinbar völlig synonym verwendet. So ist beispielsweise der Titel des Artikels „Speech Technology in Computer-

Aided Language Learning: Strengths and Limitations of a New CALL Paradigm“ (Ehsani/Knodt 1998) in der ERIC-Datenbank (*Education Resource Information Center*) folgendermaßen gelistet: „Speech Technology in Computer-*Assisted* Language Learning: Strengths and Limitations of a New CALL Paradigm“. Davon abgesehen wird jedoch *Computer Assisted Language Learning* bei weitem häufiger verwendet. Neben der Bezeichnung CALL existieren noch eine Reihe weiterer Begriffe für Gebiete, welche zumindest Überlappungen mit CALL aufweisen. Vor allem in älteren Publikationen findet man den Begriff Computer-Assisted Instruction (CAI). Während jedoch CALL sämtliche Rollen des Computers im Sprachunterricht abdeckt, wird CAI nur noch für Programme mit Tutorial- und Drill-and-practice-Hintergrund verwendet (Levy 1997: 81). Von größerem Interesse ist der Begriff des ICALL (Intelligent CALL). Während für CALL-Anwendungen sämtliche zur Verfügung stehenden Methoden verwendet werden können, setzt ICALL den Einsatz von Techniken aus der künstlichen Intelligenz voraus (Levy 1997: 79), beispielsweise NLP (Natural Language Processing)-Techniken (mehr dazu in 2.2.3). Relativ häufig werden die Begriffe *Computer-Enhanced Language Learning (CELL)* und *Technology-Enhanced Language Learning (TELL)* verwendet. Während CELL die Betonung auf die Verbesserung des Sprachenlernens durch den Computer legt, wird in TELL der Begriff des Computers auf den der Technologie erweitert, da der Computer eben nur einen Teil einer Vielzahl von im Sprachunterricht verwendeten Technologien darstellt (Levy 1997: 81). Sowohl CELL als auch TELL referieren jedoch auf das gleiche Gebiet wie CALL. Da sich der Begriff CALL im Laufe der Zeit zum gängigsten Term entwickelt hat (Levy 1997: 82), wird dieser in weiterer Folge in dieser Arbeit ausnahmslos verwendet.

2.1.2 Geschichtliche Entwicklung

2.1.2.1 Anfänge ab 1960: PLATO und TICCIT

Als eine der frühesten CALL-Anwendungen gilt das von der Universität Illinois gestartete *PLATO*-Projekt (Programmed Logic for Automatic Teaching Operations) (Levy 1997: 15). Das erste PLATO-System (PLATO I) lief auf einem zentralen Rechner, auf den man von Terminals (bestehend aus Bildschirm und Tastatur) aus zugreifen konnte. Die Übungsbeispiele für das PLATO-System wurden mit der speziell für derartige Zwecke entwickelten Programmiersprache *TUTOR* erstellt (Wooley 1994: 5). Durch die Vernetzung der Terminals verfügten die Benutzer über die Möglichkeit der Kommunikation in Form eines (eingeschränkten) Email-Systems (Levy 1997: 15-16). Dies ermöglichte sowohl Lehrer-Schüler- als auch Schüler-Schüler-Kommunikation. Die Verwendung von TUTOR zur Erstel-

lung der Übungen hatte sowohl Vor- als auch Nachteile. Die Vorteile von TUTOR waren die einfache Syntax und damit leichte Erlernbarkeit. Ein großer Nachteil des Systems war jedoch die Tatsache, dass nur TUTOR-Programme auf dem System laufen konnten, was anspruchsvollere Anwendungen verhinderte (Hart 1995: 36). Nach insgesamt vier Versionen und einem gescheiterten Versuch der kommerziellen Verwendung von PLATO (Hart 1995: 30ff) und dem allmählichen Aufkommen von billigeren Hardware-Alternativen in Form von PCs lief das Projekt schließlich aus.

Ein knappes Jahrzehnt nach PLATO wurde das *TICCIT*-Projekt 1971 an der Brigham Young University gestartet. Das TICCIT-System (*Time-Shared, Interactive, Computer Controlled Information Television*) kombinierte TV- mit Computertechnologie (Levy 1997: 18). Im Gegensatz zu PLATO, dem ein zentrales System zugrunde lag, wurden im TICCIT-System maximal 128 Terminals über einen Minirechner gesteuert, womit die beiden Systeme zumindest aufgrund der unterschiedlichen Herangehensweisen in einem Konkurrenzverhältnis standen (Hart 1995: 17). Aufgrund des Umstandes, dass TICCIT Video-, Text- und Audio-Daten kombinierte, sieht Levy (1997: 18) TICCIT als die erste Multimedia-Anwendung im Bereich der CAI (Computer Assisted Instruction). Sowohl PLATO als auch TICCIT beeinflussten die Entwicklung von CALL auf zwei Arten: Einerseits beinhalteten beide Systeme bedeutende CALL-Komponenten in Form von Kursunterlagen für zahlreiche Sprachen. Andererseits waren diese Projekte quasi Laboratorien zur Beschäftigung mit CALL und schufen die Grundlagen für eine professionelle Infrastruktur (Chapelle 2001: 6).

2.1.2.2 Entwicklung ab 1980

Ab der zweiten Hälfte der 1970er änderte sich die Auffassung hinsichtlich des Sprachenlernens grundsätzlich. Die bis dahin geltenden, von Skinners Behaviorismus geprägten pädagogischen Richtlinien wurden abgelöst durch die Komplexität des Sprachenlernens und -unterrichtens berücksichtigende, neue Methoden (Levy 1997: 21), z.B. *Communicative Language* Teaching (*CLT*). Nicht nur die pädagogischen, sondern auch die technischen Grundlagen für CALL änderten sich mit dem allmählichen Auftreten von auch für Privatpersonen erschwinglichen Kleinrechnern wie dem Commodore PET oder dem Apple II (Levy 1997: 22). Durch den Umstand, dass Besitzer eines Computers nicht mehr länger auf beispielsweise von Universitäten bereitgestellte Zentralrechner angewiesen waren, konnte das allgemeine Interesse für CALL gesteigert werden (Chapelle 2001: 8), was Anfang der 1980er Jahre einen regelrechten CALL-Boom zur Folge hatte (Levy 1997: 22). Mittels Programmiersprachen wie *BASIC* oder dem 1987 eingeführten *HyperCard* konnten einfache CALL-Applikationen ohne großen Aufwand geschrieben werden, ein-

zige Einschränkung war der Grad der Programmierkenntnisse des Autors. Auch die Entwicklung von Textverarbeitungsprogrammen wie *WordMaster* oder *WordStar* trugen zum kontinuierlichen Einsatz von Computern im Sprachunterricht bei. Doch auch die Tradition der Großprojekte wie PLATO wurde durch das vom MIT initiierte *Athena Language Learning Project* (ALLP) fortgeführt. Drei zu dieser Zeit neue Technologien wurden im Zuge von ALLP berücksichtigt: NLP bzw. Verarbeitung gesprochener Sprache (Speech Processing) und interaktives Video (Reuer 2004: 20).

2.1.2.3 Entwicklung ab 1990

Das CALL der Neunziger Jahre wurde in erster Linie von der Entwicklung des Internets beeinflusst. Durch den sukzessiven Ausbau des Netzes kamen immer mehr Benutzer in den Genuss der Vorzüge des neuen Mediums und CALL-Anwendungen wie das Projekt *The International Email Tandem Network* (Brammerts 1996) wurden realisiert. Das 1993 an der Ruhr-Universität in Bochum gestartete Netzwerk stellt die notwendige Umgebung zur Verfügung, welche den Beteiligten das Erlernen einer Fremdsprache in Form von Sprachtandems ermöglicht. Das Netzwerk besteht aus einer großen Anzahl von Subnets, beispielsweise eines Spanisch-Französischen, welches unter anderem ein zweisprachiges Forum zum Gedankenaustausch für die Teilnehmer bereitstellt (Levy 1997: 33). Nachdem das Netzwerk am Beginn des Betriebes aufgrund des damals noch eingeschränkten Internetzugangs auf Arbeitsplätze in den teilnehmenden Universitäten beschränkt war (Brammerts 1996: 5), ist die Teilnahme inzwischen von jedem mit Internetverbindung ausgestattetem Computer aus möglich. Aus CALL-Sicht muss jedoch angemerkt werden, dass der Beitrag des Computers im Falle des Tandem Networks relativ gering ist. Streng genommen dient es nur als Medium zum Datenaustausch in Verbindung mit Programmen beispielsweise zur Textverarbeitung. Die eigentlich „lernspezifische“ Arbeit muss von den Anwendern übernommen werden, womit das Tandem Network gemäß der Definition von Hardisty et al (1989: 5) im eigentlichen Sinne keine echte CALL-Anwendung ist:

> „It [CALL] is the term most commonly used by teachers and students to describe the use of computers as part of a language course. It does not refer to the use of a computer by a teacher to type out a worksheet.“

2.1.3 CALL aus pädagogischer Sicht

In der Periode vom erstmaligen Auftreten von CALL bis in die Gegenwart haben sich nicht nur die technischen, sondern auch die pädagogischen Grundlagen weitgehend verändert. Warschauer (1996: 3ff) unterteilt CALL in drei diachrone Phasen: behavioristisches CALL (*Behavioristic CALL*), kommunikatives CALL (*Communicative CALL*) und integratives CALL (*integrative CALL*). Das *behavioristische CALL* basiert auf der zu jener Zeit geltenden behavioristischen Auffassung von Sprachenlernen, beeinflusst von den Arbeiten Skinners. Die während dieser Phase entstandenen Programme (z.B. PLATO, vgl. 2.1.2.1) generierten Übungen, welche auf dem Prinzip des Erlernens einer Sprache durch stetiges Wiederholen von Beispielen basierten, so genannten „Drill and practice"-Übungen.

Ab der zweiten Hälfte der 70er Jahre änderten sich schließlich die pädagogischen Grundlagen und die behavioristischen Ansätze wurden durch neue ersetzt. In diese Periode fällt laut Warschauer der Wechsel vom behavioristischen zum *kommunikativen CALL*. Underwood (1984: 19ff) erstellte eine Liste mit Voraussetzungen für kommunikatives CALL.
Als wichtigste Punkte zählt Underwood auf:
- Kommunikation als Ziel
- Implizites statt explizites Erlernen von Grammatik
- Fabrikation anstelle von vorfabrizierten Beispielen
- Vermeidung von andauernder Beurteilung des Lernenden, keine Belohnung in Form von Gratulationsmeldungen, blinkenden Lichtern oder Klängen
- wichtigste Eigenschaft des kommunikativen CALL: „It is fun!" (Salaberry 1996: 8).

Die dritte Phase, das integrative CALL, wurde schließlich ab Ende der 1980er Jahre, begünstigt durch die Entwicklung des Internets sowie die Fortschritte auf dem Multimedia-Sektor, eingeleitet. Durch die Verknüpfung von verschiedenen Medien auf dem Computer („Hypermedia") wurden Anwendungen möglich, die unterschiedliche Prozesse, welche bis dahin als separate Programme ausgeführt werden mussten, in eine einzige Umgebung transferierten. So konnten Lese-, Schreib-, Hör- und Sprechübungen kombiniert werden (Warschauer 1996: 7). Trotz der Vorteile, welche durch die Entwicklung der Multimedia-Applikationen für CALL geschaffen wurden, wurden seitens der Entwickler schwerwiegende Fehler gemacht. Während vielen Pädagogen, die Anwendungen selbst mit Hilfe von Entwicklungswerkzeugen wie *Hypercard* erstellten, ganz einfach die Übung bzw. die nötige Erfahrung auf dem Gebiet fehlten, wurden kommerzielle Lernprogramme ohne Rücksicht auf pädagogische Prinzipien, auf deren Berücksichtigung während der Entwicklung aus Kostengründen verzichtet wurde, auf den Markt gebracht (Warschauer 1996: 8).

Warschauer (1996: 8) führt ein weiteres, fundamentaleres Problem an: „Today's computer programs are not yet intelligent enough to be truly interactive". Trotz der zur Verfügung stehenden technischen Möglichkeiten waren die Programme nicht mit dem notwendigen Know-how ausgerüstet, um die Leistungen der Benutzer in „intelligenter Art und Weise" zu kommentieren. Die Berücksichtigung von in integrativen CALL-Anwendungen vernachlässigten Faktoren ist eine der Aufgaben und Ziele des *intelligenten CALL (ICALL)*, welches als nächste Phase gesehen werden kann (Warschauer 1996: 8). Der neben der Entwicklung im Multimedia-Bereich für integratives CALL maßgebende Faktor war die Entwicklung des Internets. Eine durch das World Wide Web ermöglichte Anwendung ist das bereits in 2.1.2.3 erwähnte *International Email Tandem Network*. Durch das Internet wurde Konversation sowohl in asynchroner (Email) als auch in synchroner (*Real Time*) Form möglich.

2.2 NLP in CALL-Anwendungen

2.2.1 Der Unterschied zwischen NLP und Sprachtechnologie

In der Literatur über CALL wird sowohl vom Begriff *NLP* (*Natural Language Processing*; automatische Verarbeitung natürlicher Sprache) als auch vom Begriff *Sprachtechnologie* Gebrauch gemacht, um auf gleiche Konzepte zu referieren. Glück (2000: 475) definiert *NLP* folgendermaßen:

> „NLP umfaßt den gesamten Forschungs- und Anwendungsbereich, der durch die Disziplinen Computerlinguistik, Linguistische Datenverarbeitung, Sprachorientierte Künstliche-Intelligenz-Forschung und Sprachtechnologie abgedeckt wird."

Glücks (2000: 671) Definition für *Sprachtechnologie*:

> „In Abgrenzung zu den z.T. eher grundlagenorientierten Fragestellungen und Methoden der Computerlinguistik, der Linguistischen Datenverarbeitung und der sprachorientierten Künstliche-Intelligenz-Forschung verwendete Bez. für Forschungs- und Entwicklungsarbeiten, die die Umsetzung von theoret. Resultaten in technolog. Anwendungen wie praxistaugl. maschinelle Übersetzungssysteme, Spracherkenner, Frage-Antwortsysteme usw. fokussieren."

Gemäß dieser Definition können die Begriffe NLP und Sprachtechnologie am Gebiet des CALL an und für sich synonym verwendet werden: jede Verwendung von

NLP-Techniken in CALL-Anwendungen ist ja per se eine Umsetzung von theoretischen Resultaten in eine technologische Anwendung, ergo Sprachtechnologie. Die Grundlagen sind die gleichen, mit dem Unterschied, dass die Verwendung des Terms *Sprachtechnologie* eine gewisse, letztendlich jedoch redundante, Spezifizierung darstellt.

2.2.2 Die Problematik hinter dem Einsatz von NLP-Technologien in CALL-Applikationen

Um der falschen Annahme zu entgehen, dass sämtliche Programme im Bereich des CALL per Definition Sprachtechnologie darstellen (Nerbonne et al. 1998: 1), ist es notwendig, jene Prozeduren, welche unter Sprachtechnologie (und somit NLP) zu subsumieren sind, von anderen zu trennen. Nerbonne et al. (1998: 1) definieren Sprachtechnologie folgendermaßen: „By this [language technology] we understand technology which carries out tasks specific to language". Unter diese *specific tasks* fallen laut Nerbonne et al. Anwendungen wie Lemmatisierung, Part-Of-Speech-Disambiguierung, Parsing, Textgenerierung, Sprachsynthese. Hypertext, digitale Audio- und Videotechnologie, Datenbanktechnologie und Netzwerkkommunikation sind folglich keine sprachtechnologischen Anwendungen. Dazu zählen auch Techniken, welche auf den ersten Blick auf Sprachtechnologie basieren, jedoch nicht als solche zu bezeichnen sind. Reuer (2004: 19) nennt als Beispiel Programme, die über Mikrofon eingegebene Daten auf korrekte Aussprache überprüfen. Sowohl die Analyse des Inputs als auch der Vergleich mit der korrekten Form basieren nicht auf NLP-Methoden.

Das Gebiet des CALL ist, wie in 2.1.1 erwähnt, riesig, werden doch sämtliche Computer-Anwendungen, welche das Erlernen von Sprachen unterstützen, zu dieser Disziplin gezählt. Der Anteil jener Anwendungen, die Methoden des NLP bzw. Sprachtechnologie nutzen, ist im Vergleich dazu jedoch gering (Zock 1996: 1002). Reuer (2004: 19) konnte im Jahr 2004 am deutschen Markt kein einziges ICALL-Programm finden. Während in Disziplinen wie der maschinellen Übersetzung großer Forschungsaufwand betrieben wurde, wurde CALL von computerlinguistischer Seite mit sehr wenig Aufmerksamkeit bedacht (Zock 1996: 1002), obwohl das Gebiet des CALL ein riesiges Feld, größer als die Computerlinguistik selbst (Nerbonne 2003: 671) abdeckt und aus wirtschaftlicher Sicht höchst interessant ist: Von den 1994 am europäischen Markt für Multimedia umgesetzten 1,9 Milliarden Euro wurden 28 Prozent für Bildungszwecke ausgegeben, ein Fünftel davon machten wiederum CALL-Anwendungen aus (Nerbonne 2003: 673).

Die geringe Präsenz der NLP innerhalb des CALL kann unter anderem auf ein gewisses Kommunikationsdefizit zwischen den benachbarten Disziplinen zurückgeführt werden. Während Computerlinguisten generell wenig Interesse in CALL zeigen, werde wiederum die Arbeit der Computerlinguisten seitens der CALL-Experten ignoriert (Zock 1996: 1002). Eine funktionierende Kommunikation sei jedoch fundamental, einerseits ist es für auf dem Gebiet der Pädagogik Tätige wichtig, Sprachtechnologien besser zu verstehen, um intelligente Systeme zu konstruieren, andererseits sollten Computerlinguisten beispielsweise über Ansprüche, welche Anwender an CALL-Software stellen, Bescheid wissen (Nerbonne et al. 1998: 1). Einen weiteren Grund für die Problematik sieht Salaberry (1996: 12) in der mangelhaften Ausgereiftheit sprachtechnologischer Methoden:

> „In essence, linguistics has not been able to encode the complexity of natural language in a finite number of discrete rules [...]. That problem has been acknowledged by several of the most adamant proponents of Intelligent CALL. Holland (1995) lists the reasons that have prevented ICALL from becoming an alternative answer to CALL. The most important reason for this failure is that NLP (Natural Language Processing) programs - which underlie the development of ICALL - cannot account for the full complexity of natural human languages [...]. As a consequence, ICALL programs present the following shortcomings: NLP-based programs are still at an experimental stage, most NLP applications require the use of large and expensive computers, and consequently, little attention is given to interface design."

Scheinbar wird die Komplexität von natürlicher Sprache unterschätzt: Probleme, die im Zuge des Athena Project (vgl. 2.2.4.1) während der Arbeiten an den NLP-relevanten Teilen auftraten, beschreibt Felshin (1995: 271):

> „NLP is hard. When we initiated our project, we naively thought that we could successfully build an NLP system in two to three years that could analyze and respond in real time to [written] input in any one of four European languages, up to the level of a fourth semester student. Instead, it took us five years to build a system that can process second- to fourth- semester level input pretty well and often in something approaching real time ... Grammar writing eventually expanded to fill all available time, preventing us from implementing more than prototypes of the numerous applications based on NLP that we had originally intended to create."

Tatsächlich sind zufriedenstellende bzw. komplette Modelle menschlicher Sprache auf computerlinguistischer Basis nicht existent und in nächster Zukunft ist

nicht damit zu rechnen (Nerbonne et al. 1998: 2). Es existieren jedoch sehr wohl NLP-Techniken, welche den Anforderungen für CALL-Anwendungen standhalten. Sowohl phonologische als auch morphologische computerlinguistische Modelle für zahlreiche Sprachen sind beispielsweise zuverlässiger als menschliche Analysen (Nerbonne et al. 1998: 2). Trotzdem ist eine gewisse negative Auffassung hinsichtlich NLP-Methoden in CALL-Anwendungen zweifellos vorhanden (Nerbonne 2003: 678).

Ein weiteres Faktum, welches die Verwendung von NLP-Methoden in CALL-Programmen zumindest bis in die 1990er Jahre hinein bremste, war die Limitierung der damals verwendeten Computer sowohl hinsichtlich (Arbeits-)Speicher als auch Geschwindigkeit (Reuer 2004: 19). Sprachressourcen wie Lexika und Korpora waren lange Zeit aufgrund von eingeschränkten Speicherressourcen nicht sinnvoll verwendbar. Dieses Manko wurde jedoch durch die rasante Entwicklung auf dem Hardwaresektor in den letzten Jahren behoben.

2.2.3 CALL auf NLP-Basis sowie ICALL

Ein Gebiet, welches in den letzten Jahren innerhalb des CALL verstärkt forciert wurde, ist ICALL (Intelligent CALL). Unter ICALL werden jene CALL-Anwendungen, die unter Berücksichtigung von Methoden aus der Künstlichen Intelligenz (Artificial Intelligence, daher AI als Abkürzung) erstellt werden, zusammengefasst. Einer der Schwerpunkte der AI-Forschung ist die Entwicklung von Experten-Systemen, d.h. Programmen, welche in bestimmten Bereichen menschliche Expertise simulieren (Salaberry 1996: 11). Ein Ziel des ICALL ist die Umsetzung derartiger Expertensystemen für den Sprachunterricht. Ob jedoch ein Einsatz von Expert-Systemen in CALL-Systemen möglich und v.a. sinnvoll ist, wird bezweifelt (Salaberry 1996: 11). Maschinelle Tutoren, welche natürliche Sprache lehren, sollten laut Salaberry (1996: 11) über folgende Basiskomponenten verfügen:
- Repräsentation des subjektspezifischen Wissens
- ein Modell des zu Lernenden
- ein Sprachverstehsystem
- die Fähigkeit, aus Erfahrung zu lernen

Während die erstgenannte Komponente zumindest in Ansätzen realisierbar ist, scheinen die drei weiteren Komponenten weit entfernt von einer ernst zunehmenden Verwirklichung zu sein, womit eine Realisierung eines derartigen maschinellen Tutors für eine Anwendung in der Sprachlehre (noch) nicht möglich scheint. Neben den bereits erwähnten unausgereiften Lösungen aus der NLP ist auch die AI noch weit entfernt davon, menschliche Kommunikation angemessen zu simulieren (Salaberry 1996: 12).

2.2.4 CALL-Programme mit NLP-Hintergrund: ein Fallbeispiel

Sowohl in den von PLATO als auch von TICCIT generierten Übungen wurde, wie bereits erwähnt, keine Rücksicht auf Methoden aus der NLP genommen. Die meisten Übungen waren vielmehr Zusammenstellungen von in einer Datenbank gespeicherten Beispielen. Zu diesen „traditionellen" CALL-Anwendungen zählen Lückentexte, Multiple-Choice-Übungen und Sortierübungen in allen möglichen Varianten (Reuer 2004: 28). Diese Art von CALL-Anwendungen ist nach wie vor populär, wenn auch, vom pädagogischen Standpunkt gesehen, nicht immer empfehlenswert. Für die Erstellung derartiger Übungen ist weder großes programmiertechnisches Wissen notwendig, noch linguistisches Hintergrundwissen. Durch Programme wie *Gapmaster*[1] stehen Entwicklern Werkzeuge zur Verfügung, welche das Erstellen von CALL-Übungen selbst für Ungeübte innerhalb kürzester Zeit ermöglichen, aufgrund der inzwischen ausreichend hohen Übertragungsgeschwindigkeiten können sowohl Sound- als auch Videodateien ohne technische Hürden integriert werden.

2.2.4.1 Das Athena Language Learning Project

Eines der ersten CALL-Projekte, die unter Verwendung von NLP-Methoden realisiert wurden, war das Athena Language Learning Project (ALLP). Das ALLP wurde Mitte der 1980er Jahre unter anderem am Massachusetts Institute of Technology (MIT) mit eben dem Ziel, CALL-Materialien für die am MIT gelehrten Sprachen zu entwickeln, gestartet (Kramsch et al 1985: 31). Im Zuge des Projektes wurden einige Übungen für den Englisch-Spanisch-Unterricht entwickelt, jedoch nur teilweise realisiert. In der ersten Übung, *LINGO* (Language Instruction through Graphics Operations), muss der Benutzer einen Poltergeist durch Instruktionen in der Fremdsprache bei der Reinigung eines Raumes unterstützen. Durch die Verwendung eines Dialog-Simulators sind relativ komplexe Konversationen mit dem Poltergeist möglich. Der Poltergeist interagiert mit dem Benutzer in Form eines natürlichen Dialogs, gibt Korrekturvorschläge für grammatikalische Fehler oder Fehler in der Satzstellung sowie Feedback (Kramsch et al 1985: 32). In der zweiten Übung, *NO RECUERDO*, agiert der User als Journalist und muss in einem fiktiven Abenteuer die Hintergründe für das Verschwinden eines Wissenschafters aufklären. Während des Abenteuers wird der Benutzer mit einer Reihe von sprachlichen Aufgaben konfrontiert, mit dem Ziel, Hörverständnis, Wortschatz sowie Schreibfähigkeit in Spanisch zu verbessern. In der fiktiven Umgebung führt der Benutzer Interviews mit Passanten, liest Essays über spanische Kultur, schreibt Berichte und bekommt

[1] Hersteller: Wida Software (http://www.wida.co.uk/)

Feedback in Form von Korrekturvorschlägen. Sowohl von LINGO als auch von NO RECUERDO wurden Demo-Versionen produziert und obwohl die Euphorie bei den Entwicklern groß war (Morgenstern und Murray 1995: 37), wurden die Programme aufgrund von Hardwarelimitationen zu keinem Zeitpunkt in realen Unterrichtssituationen getestet (Reuer 2004: 20). Die dritte Anwendung des ALLP bietet eine Reihe von interaktiven Grammatikübungen an. Ausgehend von englischen Paraphrasen muss der User spanische Sätze bilden. Auch diese Übung wurde aufgrund von Hardwarelimitationen nicht vollständig implementiert, immerhin konnte sie jedoch, im Gegensatz zu den ersten beiden Übungen, in realen Situationen von Sprachschülern verwendet werden (Reuer 2004: 20). Während der Entwicklung der ALLP-Übungen kamen eine Reihe von NLP-Techniken zum Einsatz, unter anderem ein Parser zur Analyse der Inputsätze, ein System zur Wissensrepräsentation sowie eine Reihe von Routinen zum Aufspüren von Schreibfehlern, inkorrekter Grammatik und semantisch inkohärenten Statements (Kramsch et al 1985: 31). Von Beginn an war das ALLP-System sprachunabhängig konzipiert, als Repräsentation diente eine Art Interlingua. Die Syntax wird gemäß der Government-Binding-Theorie unter Verwendung von Konzepten wie S-, D- und CF-Struktur generiert, für eine möglichst genaue semantische Analyse des Inputs kamen sowohl im Lexikon als auch in der Grammatik thematische Rollen zur Verwendung (Reuer 2004: 22). Die Lexikoneinträge waren dementsprechend umfangreich (sh. Abb. 2.1).

```
(:voice active
(:thematic-role agent
:syntax((:case subject :type dp :required-p t)))
(:thematic-role theme
:syntax((:case empty :type vmax :spec (or indicative infini-
tive):required-p t)))
(:thematic-role destination
:syntax((:case indirect-object :type dp :required-p t))))
```

Abbildung 2.1: ALLP-Lexikon-Frame für das englische Verb *to tell*

Wie bereits erwähnt, konnten die im Zuge des ALLP entwickelten Anwendungen nur zum Teil auch tatsächlich realisiert und im Unterricht eingesetzt werden. Für NO RECUERDO existiert immerhin eine Website[2], die jedoch nicht mehr aktuell zu sein scheint. Als Projektstatus wird „Currently still in development“ angegeben, als Datum für Beta-Tests ist auf der Website 1997 (!) vorgesehen.

[2]URL: http://web.mit.edu/fll/www/projects/NoRecuerdo.shtml [01.05.2008]

Kapitel 3

Selektionsrestriktionen auf Basis von Konzepten aus Ontologien oder Taxonomien

3.1 Das Konzept der Selektionsrestriktionen

3.1.1 Was sind Selektionsrestriktionen?

Der Fokus des dieser Arbeit zugrundeliegenden Programms SCall liegt auf der Generierung von semantisch korrekten deutschen und schwedischen Konstruktionen bzw. Sätzen. Wie bereits in der Einleitung erwähnt, wird in Programmen, die keine Rücksicht auf die Semantik nehmen, wie etwa dem Referenzprogramm *Cica*, davon ausgegangen, dass der Benutzer entweder dafür Sorge trägt, dass nur Wörter verwendet werden, die abhängig vom jeweiligen Kontext semantisch korrekte Konstruktionen zur Folge haben, oder semantisch inkorrekte Übungsbeispiele in Kauf nimmt. Nimmt man folglich die in Tabelle 3.1 enthaltenen Vokabeln als Korpus, um Konstruktionen gemäß dem dargestellten Muster zu generieren, sind aufgrund der Wahl des Vokabulars keine semantischen Probleme zu erwarten.

Eingabemuster	**Vokabular**
das hohe Haus	Adj.: klein, breit, hoch, schmal; Nomen: Tisch, Kasten, Gebäude;

Tabelle 3.1: Mögliches Eingabemuster sowie Vokabular in *Cica*

Ergänzt man das Lexikon jedoch durch Wörter, die mit den bereits enthaltenen semantisch nicht korrelieren, wie dem Adjektiv *intelligent* oder dem Nomen *Maus*, ist die Generierung von Konstruktionen wie (1a) oder (1b) zu erwarten.

(1) (a) ?Der intelligente Kasten.
(b) ?Die breite Maus.

Während die für die Generierung notwendige syntaktische Information in Form des Musters (Tab. 3.1, Spalte 1) vorhanden ist und die notwendige morphologische Information entweder aus dem Lexikon extrahiert oder gegebenenfalls von einem entsprechenden Programm wie *PC-Kimmo* oder *Morphix* generiert werden kann, ist die Verfügbarkeit von semantischer Information aus dem Input nicht möglich. Eine Möglichkeit, die Generierung von semantisch korrekten Konstruktionen zu gewährleisten, besteht darin, das Vokabular durch semantische Information zu ergänzen. Im Fall der Sätze aus (2) würde beispielsweise die Information darüber, dass in der Regel nur Menschen und Tiere intelligent sein können[1] und folglich das Adjektiv *intelligent* nur entsprechende Nomen spezifizieren kann, dazu führen, dass die Generierung des Satzes (2a) ausgeschlossen wäre.

Eine Methode, die Semantik von Wörtern durch derartige Informationen in ein Lexikon einzubeziehen, ist die Annotation des Lexikons durch sogenannte *Selektionsrestriktionen* (bzw. *Selectional Restrictions* oder auch *Selectional Preferences*). Dieses Konzept wurde innerhalb der Linguistik bereits in den 1960er Jahren thematisiert (z.B. in Chomsky 1965: 78ff) und seitdem in zahlreichen (computerlinguistischen) Theorien berücksichtigt. Selektionsrestriktionen sind im Grunde Beschränkungen, die die Verwendung von Wörtern auf bestimmte semantische Umgebungen einschränken. Durch die Annotation mit Selektionsrestriktionen erhalten Wörter jene semantischen Informationen, die (zumindest zu einem gewissen Grad) semantisch korrekte Konstruktionen garantieren. Ein mögliches Verwendungsgebiet ist die Sicherstellung von semantisch korrekt generierten Verbvalenzen. Auf syntaktischer Ebene besitzen Verben die Eigenschaft, den Rahmen ihres Vorkommens durch Valenzen zu definieren; so verfügt das Verb *schlafen* über nur eine Valenz (Subjekt), welche Konstruktionen, die ein Akkusativ-Objekt enthalten, wie **Peter schläft das Bett*, aus syntaktischer Sicht unmöglich machen. Die Verbvalenzen geben zwar Auskunft über die geforderte syntaktische Struktur, die semantischen

[1] Die Tatsache, dass das Adjektiv *intelligent* nicht nur zur Spezifizierung von belebten Dingen verwendet werden kann, sondern auch für die von abstrakten wie in *intelligentes Design* oder *intelligente Frage* wird hier nicht berücksichtigt, da es sich in diesen Fällen um eine andere Lesart von *intelligent* handelt: im ersten Fall spezifiziert das Adjektiv das Nomen bezüglich seiner kognitiven Fähigkeiten.

Code	Beschreibung	Code	Beschreibung
A	Animal	B	Female Animal
C	Concrete	D	Male Animal
E	Solid or Liquid (not gas): S + L	F	Female Human
G	Gas	H	Human
I	Inanimate Concrete	J	Movable Solid
K	Male Animal or Human = D + M	L	Liquid
M	Male Human	N	Not Movable Solid
O	Animal or Human = A + H	P	Plant
Q	Animate	R	Female = B + F

Tabelle 3.2: Semantische Codes aus LDOCE (Auszug)

Eigenschaften werden jedoch nicht berücksichtigt. Ergo sind Konstruktionen wie **Das Bett schläft* aus rein syntaktischer Sicht akzeptabel, obwohl der Agens von *schlafen* aus semantischer Sicht entweder ein Mensch oder Tier sein muss. Durch die Annotation mit den entsprechenden Selektionsrestriktionen kann die Belegung der Agensposition des Wortes *schlafen* jedoch darauf eingeschränkt werden, dass das entsprechende Nomen entweder vom Typ *HUMAN* oder *ANIMAL* ist, während sämtliche Nomen, welche nicht über diese semantische Eigenschaft verfügen, die Agensposition des Verbs folglich auch nicht besetzen können. Der entsprechende Lexikoneintrag für das Verb *schlafen* könnte folgendermaßen aussehen, wobei die Annotationen in eckigen Klammern die Selektionsrestriktionen repräsentieren:

schlafen — V — Agens [+HUMAN] [+ANIMAL]

Somit sind Konstruktionen wie *Das Buch schläft* aus semantischer Sicht ausgeschlossen, da *Buch* weder einer Instanz von *HUMAN*, noch einer Instanz von *ANIMAL* entspricht.

Momentan existieren jedoch (noch) sehr wenige lexikalische Ressourcen, deren Einträge über semantische Informationen, die als Selektionsrestriktionen verwendet werden können, verfügen. Eines der wenigen derartigen Lexika ist das *Longman Dictionary of Contemporary English* (kurz LDOCE[2]). LDOCE ist ein in erster Linie für den Unterricht gedachtes Online-Lexikon mit insgesamt mehr als 45 000 Einträgen. Neben der Information über Aussprache, Worttrennung, Wortart sowie einer kurzen Beschreibung sind die Einträge in LDOCE mit 32 unterschiedlichen semantischen Codes versehen (Tab. 3.2). Bruce & Guthrie (1992: 1189) ordneten die Codes aus LDOCE in einer Hierarchie (Abb. 3.1), die als Grundlage für die Verwendung der Codes als Selektionsrestriktionen dient (z.B. in Stevenson & Wilks 2001). Die Kategorisierung der semantischen Codes dient einerseits dazu, die Codes

[2]URL: www.ldoce.com; LDOCE online: http://pewebdic2.cw.idm.fr/ [30.05.2008]

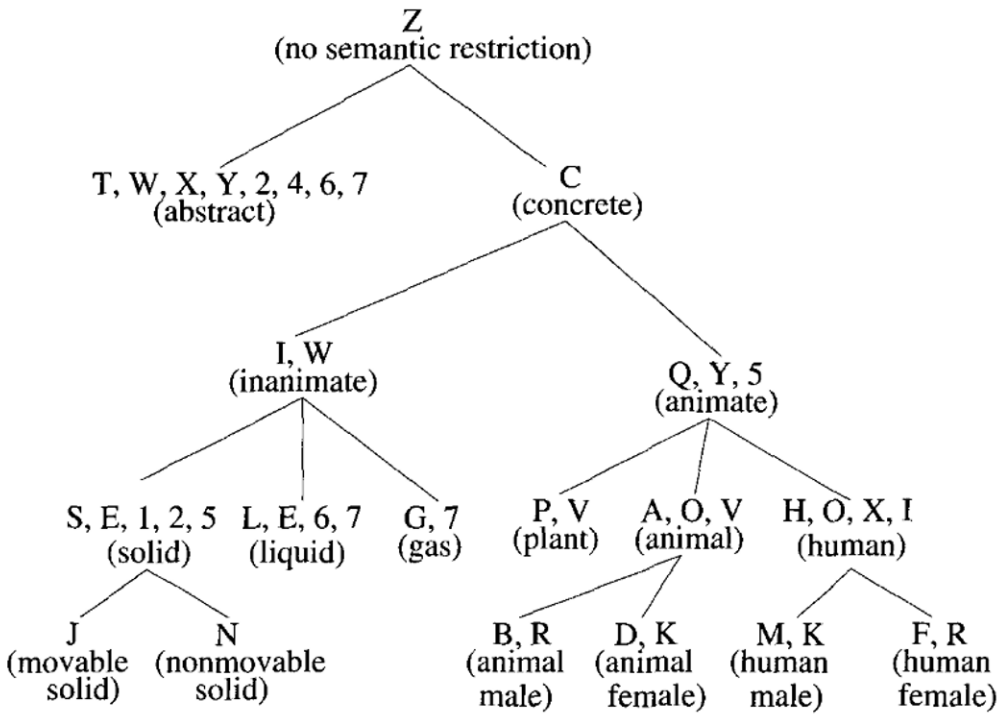

Abbildung 3.1: Hierarchie der LDOCE-Semantikcodes, nach Bruce & Guthrie 1992.

in Relation zueinander zu bringen, andererseits ergibt sich durch die taxonomische Hierarchie die Möglichkeit der Generalisierung von Restriktionen. Folglich können unter der Selektionsrestriktion *animate* bzw. *Q* (sh. Tab. 3.2) sämtliche belebten Objekte zusammengefasst werden. Aufgrund des letztendlich doch eingeschränkten Umfangs stoßen die Selektionsrestriktionen aus LDOCE jedoch relativ rasch an Grenzen. Im Fall von physischen Objekten (z.B. *Buch* oder dem Namen *Peter*) können die Selektionsrestriktionen aus Abb. 3.1 meist noch ausreichende semantische Informationen liefern. Durch die Restriktionen *human male* - *M* kann *Peter* relativ gut semantisch kategorisiert werden. Ähnliches gilt für Adjektive: jedes Adjektiv, das als Spezifikation für das Nomen *Peter* angewendet werden kann, kann für jedes andere Nomen mit den Selektionsrestriktionen *human male M* verwendet werden. Die Problematik der Hierarchie von Bruce & Guthrie liegt jedoch im Bereich der abstrakten Begriffe. Da es nur eine einzige Kategorie für diese Begriffe gibt, wird die semantische Unterscheidung derartiger Begriffe mit großer Wahrscheinlichkeit nicht oder nur sehr eingeschränkt funktionieren, vor allem, wenn man bedenkt, dass die Klasse der Abstrakta eine riesige Anzahl von semantisch sehr unterschiedlichen Objekten enthält. Und obwohl die Objekte wenige semantische Gemeinsamkeiten haben, müssen sie aufgrund fehlender Alternativen mit der gleichen Selektionsrestriktion *ABSTRACT* versehen werden (vgl. Bsp. 2).

(2) *Vokabular:* Unfall [+abstract]; Idee [+abstract]; gut [+abstract ...]
Generierungen: die gute Idee; ?der gute Unfall

Die semantischen Codes aus LDOCE stellten sich, abgesehen davon, dass LDOCE ohnehin nur für Englisch existiert und eine vergleichbare Ressource für Deutsch oder Schwedisch nicht gefunden werden konnte, folglich als zu unpräzise für den Einsatz als Selektionsrestriktionen in SCall heraus. Aus diesem Grund wird in SCall ein anderer Ansatz zur Annotation des Lexikons mit Selektionsrestriktionen gewählt. Da als Quelle für Selektionsrestriktionen keine entsprechende Ressource vorhanden ist, sollen die Restriktionen während der Laufzeit der Anwendung vom Benutzer ausgewählt werden. Die genaue Auswahlprozedur wird im zweiten Teil der Arbeit unter 5.5.3 beschrieben. Um eine möglichst präzise Bandbreite an Restriktionen zu erreichen, werden die Konzepte einer Ontologie bzw. Taxonomie als Selektionsrestriktionen herangezogen. Eine Übersicht über Ontologien im Allgemeinen und über speziell für linguistische Anwendungen konzipierte Ontologien sowie die Wahl einer als Grundlage für die Selektionsrestriktionen in SCall geeigneten Ressource werden im Abschnitt 3.2 diskutiert.

3.1.2 Einschränkungen von Selektionsrestriktionen

An dieser Stelle muss angemerkt werden, dass Selektionsrestriktionen als praktisches Hilfsmittel, jedoch keinesfalls als Lösung sämtlicher Semantik-Probleme gesehen werden können. In vielen Fällen sind dem Einsatz von Selektionsrestriktionen aufgrund von sprachlichen Nuancen deutliche Grenzen gesetzt. So sind beispielsweise die Substantive *Dachboden* und *Keller* Räume, welche in der Mikrokosmos-Ontologie (vgl. 3.2.6.4) als Instanzen des Konzeptes [+BUILDING-PART] kategorisiert werden können. Demnach würde die Analyse des Satzes *Der Koffer ist auf dem Dachboden* der Präposition die Selektionsrestriktion [+BUILDING-PART] zuweisen und folglich die Generierung des Satzes *?Der Koffer ist auf dem Keller* zulassen. Ein ähnliches Beispiel aus dem Französischen nennt Saint-Dizier (2006: 12). Die Phrase *Boire dans un verre* kann als *aus einem Glas trinken* übersetzt werden (wörtlich übersetzt *in ein Glas trinken*). Während anstelle von *verre* Trinkgefäße wie *verre, tasse, bol* (*Glas, Tasse, Schale*) als Argumente eingesetzt werden können, wird *boire dans une bouteille* (*Flasche*) nicht akzeptiert. Das mag laut Saint-Dizier auf den engen Flaschenhals und der daraus resultierenden, von den vorher genannten Trinkgefäßen abweichenden, Trinktechnik zurückzuführen sein. Jedenfalls ist auch hier Weltwissen notwendig, welches nur sehr schwer durch Selektionsrestriktionen abgedeckt werden kann. Ähnlich problematisch ist die Verwendung der

norwegischen Präpositionen *på* und *i* für Ortsangaben. In beiden Fällen können die Präpositionen mit *in* übersetzt werden, wobei geographische Kenntnisse notwendig sind, um die unterschiedlichen Präpositionen richtig zu verwenden. Während *på* teilweise für Ortschaften im Landesinneren von Norwegen verwendet wird, wird *i* für Städte und Ortschaften, die an der Küste liegen, verwendet.

3.2 Ontologien als Grundlage für die Wahl der Selektionsrestriktionen

3.2.1 Ontologie als philosophische Disziplin

Der Begriff *Ontologie* hat sich seit den frühen 1990er Jahren innerhalb der Informatik, der Künstlichen Intelligenzforschung und weiteren verwandten Disziplinen etabliert. Der ursprüngliche, aus dem griechischen *ontos* (=sein) und *logos* (=Lehre, Wort) zusammengesetzte Term bezeichnet in der Philosophie die Allgemeine Metaphysik, die sich mit der Lehre vom Seienden beschäftigt. Im Gegensatz zur speziellen Metaphysik beschäftigt sich die Ontologie mit der Beschreibung der Strukturen des Wirklichen und Nichtwirklichen (des *Seienden*), jedoch nicht mit der Erklärung dieser Phänomene (Meixner 2004: 9). Der griechische Philosoph Parmenides von Elea (5. u. 4. Jh. v.Chr.) beschäftigte sich als erster mit derartigen Fragen (Gómez-Pérez 2004: 3). In seiner Kategorienschrift setzte sich Aristoteles mit den Fragen der Ontologie auseinander, von ihm stammen auch die berühmten zehn Kategorien, eine hierarchische Zusammenstellung von verschiedenen Arten des Seins, *um alles in der Welt gesagte* zu klassifizieren (Gómez-Pérez 2004: 3)(Abb 3.2). Die zehn Kategorien beschreibt Aristoteles folgendermaßen:[3,4]

> „*Wesen* ist, um es im Umriß zu sagen zum Beispiel: Mensch, Pferd. Ein *Wieviel* ist zum Beispiel: zwei Ellen lang, drei Ellen lang. Ein *Wie-Beschaffen* ist zum Beispiel: weiß, der Grammatik kundig. Ein *In-bezug-auf* ist zum Beispiel: doppelt, halb, größer. [...] Ein *Wo* ist zum Beispiel: im Lykeion, auf dem Marktplatz. Ein *Wann* ist zum Beispiel: gestern, voriges Jahr. Ein *Liegen* ist zum Beispiel: steht, sitzt. Ein *Haben* ist zum Beispiel: beschuht, bewaffnet. Ein *Tun* ist zum Beispiel: schneidet, zündet an. Ein *Widerfahren* ist zum Beispiel: wird geschnitten, wird angezündet."

[3] Aristoteles, Kategorien, IV, deutsche Übersetzung aus: Rath 1998.
[4] Passagen in kursiver Schreibung nachträglich eingefügt.

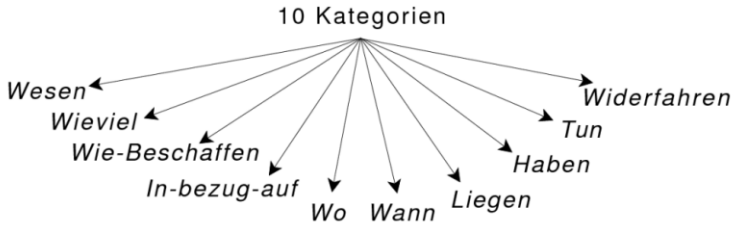

Abbildung 3.2: Aristoteles' zehn Kategorien

Der Philosoph Porphyrius nahm eine weitere Unterteilung von Aristoteles' Konzept *substance* vor, um eine detailliertere Unterscheidung der Begriffe zu erreichen (Nugues 2006: 345). Die Erweiterung der Hierarchie folgte einem rekursiven Schema. Um einer Kategorie, von Porphyrius *Genus* genannt, eine untergeordnete hinzuzufügen, werden die Elemente der Kategorie entsprechend bestimmter Eigenschaften, der *Differentiae*, unterteilt. Im Fall von Porphyrius' Genus *Animal* trennen zum Beispiel die Differentiae *rational* und *irrational* Menschen (*Human*) von Biestern (*Beasts*).

3.2.2 Ontologien in der Wissensverarbeitung

Mitte der 1990er Jahre wurde der Begriff Ontologie in mit Wissensverarbeitung in Verbindung stehenden Disziplinen relevant. Um die klassische Ontologie nicht mit dem neuen Gebiet zu verwechseln, schlugen Guarino und Giaretta (1995, zitiert aus Gómez-Pérez et al 2004: 6) vor, die beiden Disziplinen durch Groß- (*Ontology* für die philosophische Disziplin) bzw. Kleinschreibung (*ontology* für die Ontologie in der Wissensverarbeitung) zu unterscheiden. Die in der wissenschaftlichen Literatur wohl am häufigsten zitierte Definition (Gómez-Pérez et al 2004: 6) des Begriffes Ontologie stammt von Gruber (1993: 199):

> „An ontology is an explicit specification of a conceptualization."

Etwas präziser ist jedoch die Definition von Studer et al. (1998: 185), eine Erweiterung der Definition Grubers:

> „An Ontology is a formal, explicit specification of a shared conceptualization."

Formal bedeutet in dieser Definition, dass die Ontologie maschinenlesbar ist, also in digitaler Form vorliegen muss. *Explicit* verweist darauf, dass die verwendeten Konzepte, Attribute etc. explizit definiert sein müssen. Die *Conceptualization* ist ein abstraktes Modell von bestimmten Phänomenen in der realen Welt. Eine detaillierte

Definition für den Begriff *Conceptualization* bzw. Konzeptualisierung gibt Gruber (1993: 199):

> „A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly."

Der Zusatz *shared* bedeutet schließlich, dass das von der Ontologie dargestellte Wissen von einer bestimmten Gruppe von Individuen akzeptiert sein muss.

3.2.3 Bestandteile einer Ontologie

In graphischer Darstellung entsprechen Ontologien einem Netzwerk, wobei die Knoten Begriffen bzw. Konzepten entsprechen, während die Kanten Relationen zwischen den Begriffen darstellen. Mahesh und Nirenburg (1995a: 1) definieren den Aufbau einer Ontologie, die für NLP-Anwendungen konstruiert wird, folgendermaßen:

> „An ontology for NLP purposes is a body of knowledge about the world (or a domain) that: a) is a repository of primitive symbols used in meaning representation; b) organizes these symbols called concepts in a tangled subsumption hierarchy; and c) further interconnects these symbols using a rich system of semantic and pragmatic relations defined among the concepts."

Neben Konzepten und den Relationen beinhaltet eine Ontologie ferner Instanzen und Axiome. Ein wichtiger Grundsatz im Aufbau von Ontologien ist Vererbung.

3.2.3.1 Konzepte

Konzepte sind abstrakte Ansammlungen von Eigenschaften. Konzepten können weitere Konzepte oder Instanzen untergeordnet sein. Eine wichtige Eigenschaft von Konzepten ist der Umstand, dass sie keinen realen Objekten entsprechen. Reale Objekte werden durch Instanzen von Konzepten dargestellt.

3.2.3.2 Instanzen

Während Konzepte die Eigenschaften vorgeben, repräsentieren deren Instanzen in einer Ontologie Objekte oder Individuen. Mit anderen Worten sind Instanzen reale

Objekte, welche sämtliche Eigenschaften des Konzeptes, von dem sie instanziiert sind, besitzen. Folglich sind die Städte *Wien, Linz, Graz* Instanzen des Konzepts *Stadt*.

Die Aufgabe von Instanzen ist die Repräsentation des in der Ontologie darzustellenden Wissens. Dieses Wissen ist je nach Art der Ontologie bzw. des Einsatzgebietes unterschiedlich, in für linguistische Zwecke verwendeten Ontologien sind Instanzen sprachliche Elemente, meist Wörter, aber auch größere grammatikalische Einheiten wie Phrasen.

3.2.3.3 Axiome

Axiome liefern innerhalb einer Ontologie geltende Aussagen, die immer als wahr gelten. Normalerweise werden Axiome dazu verwendet, Wissen, welches nicht durch andere Komponenten der Ontologie dargestellt werden kann, zu repräsentieren (Gómez-Pérez et al 2004: 14). Ein Beispiel für ein Axiom in einer Ontologie für Reiseinformation wäre die Aussage *Es ist nicht möglich, von den USA per Zug nach Europa zu reisen* (aus Gómez-Pérez et al. 2004: 32).

3.2.3.4 Relationen

Relationen dienen der Beschreibung der Beziehungen zwischen den Konzepten einer Ontologie. In der von Aristoteles und von Porphyrius erweiterten Hierarchie stehen die Konzepte durch Hyponymie in Relation, das heißt, das untergeordnete Konzept stellt ein Teilgebiet des übergeordneten Konzeptes dar. Die in Ontologien für den linguistischen Gebrauch (z.B. in WordNet, vgl. 3.2.6.1) verwendeten Relationsarten sind u.a. Hyponymie, Meronymie, Antonymie und Entailment.

3.2.3.4.1 Hyponymie Die wohl gängigste für ontologische Darstellungen verwendete Relationsart ist die der Hyponymie, also semantisch-begriffliche Unterordnung wie in Abb. 3.3 anhand von *human* illustriert.

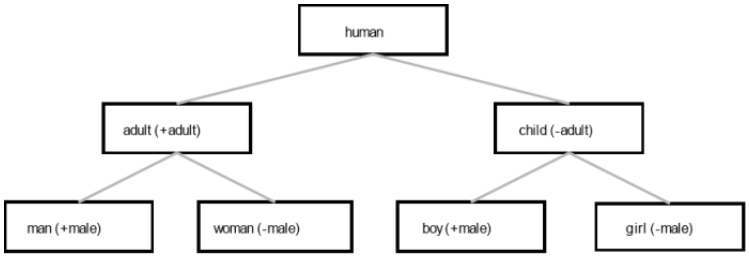

Abbildung 3.3: Hyponymie

Hierarchien, die ausschließlich auf Hyponymie-Relationen basieren, werden auch als Taxonomien bezeichnet. Zu Kontrollzwecken kann in der Regel als Substitutionsrahmen die Phrase *X ist eine Art von Y* (bzw. *X is a kind of Y*) verwendet werden, wobei X das Hyponym und Y das übergeordnete Konzept, das Hyperonym, repräsentiert. In Hyperonymie-Verhältnissen erhalten die Hyponyme sämtliche Eigenschaften des Hyperonyms (Vererbung, vgl. 3.2.3.5). Durch das Hinzufügen von zusätzlichen Eigenschaften wird eine Spezifizierung des übergeordneten Konzeptes sowie eine Unterscheidung von den Schwesterkonzepten erreicht. Das Konzept *man* aus Abb. 3.3 erbt sämtliche Eigenschaften des Hyperonyms *adult*: jeder Mann hat alle Eigenschaften, die auch ein Mensch hat. Durch die zusätzliche Eigenschaft *+male* (*Differentia*) wird das Konzept *man* von seinem Schwesternkonzept *woman* abgegrenzt. Das neu hinzugefügte Feature wird wiederum, mitsamt den vom übergeordneten Konzept übernommenen, an die dem aktuellen (*man*) durch Hyponymie untergeordneten Konzepte (z.B. *verheiratete Männer* vs. *nicht verheiratete Männer*) weitergegeben bzw. vererbt.

Die Hyponymierelation bietet sich in erster Linie zur Kategorisierung von konkreten Dingen, z.B. Tieren und dergleichen, an. In Taxonomien, die für linguistische Zwecke konzipiert sind und neben Nomen weitere Wortarten kategorisieren, kann die Verwendung von Hyponymierelationen Probleme bereiten. Im Gegensatz zu den Nomen ist beispielsweise der Substitutionsrahmen *X is a kind of Y* für Verben ausgesprochen schlecht geeignet: der Satz *To nibble is a kind of to eat* ist nicht korrekt (Fellbaum & Miller 1990: 566). Aus diesem Grund ersetzten Fellbaum & Miller den Substitutionsrahmen für Verben durch einen die Hyponymie-Relation Verben am ehesten entsprechenden Rahmen: V_1 *is a manner of* V_2 bzw. V_1 *ist to* V_2 *in some manner*. Diese semantische Relation bezeichneten Fellbaum & Miller (1990: 566) als Troponymie (abgeleitet vom griechischen Wort tropos = Art und Weise).

3.2.3.4.2 Meronymie Unter Meronymie versteht man Konzepte, die Teile des übergeordneten Konzeptes wiedergeben (aus diesem Grund ist die Bezeichnung *Part-Of-Relation* geläufig) Ist ein Begriff X ein Meronym eines Begriffes Y, so bezeichnet der Begriff X einen Teil des Begriffes Y. Die der Meronymie entgegengesetzte Relation ist die Holonymie. Das für Hyponymierelationen gültige Prinzip der Vererbung kann für Meronymie-Relationen nicht angewandt werden: ein *Motor* hat nicht die Eigenschaften des Holonyms *Auto* (siehe Abb. 3.4). Transitivität gilt für Meronymierelationen, im Gegensatz zur Hyponymie, nur in eingeschränktem Ausmaß (vgl. Lyons 1977: 312). Während *Türklinke* ein Meronym von *Tür* und *Tür* wiederum Meronym von *Haus* ist, ist *Türklinke* kein Meronym von *Haus*. Das

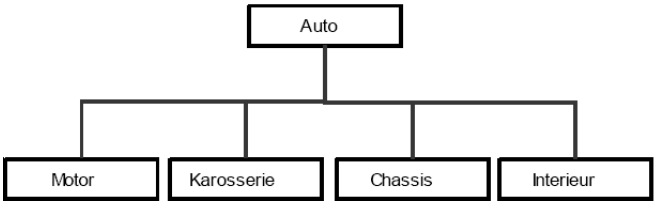

Abbildung 3.4: Meronymie

Verhältnis zwischen Meronymie und Hyponymie ist nicht unkompliziert, da Teile sowohl Meronyme als auch Hyponyme sein können. *Nib* ist nicht nur ein Meronym von *bird*, es ist auch ein Hyponym von *jaw*, das wiederum ein Meronym von *skull* und ein Hyperonym von *skeletal-structure* ist (Abb. 3.5). Ein in vielen Taxonomien

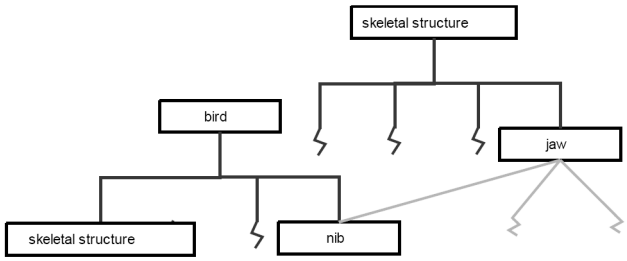

Abbildung 3.5: Diffizile Meronymie-Hyponymie-Beziehung für das Konzept *skeletal-structure*

gefundenes Problem liegt in der generellen Tendenz, Teile in einem zu hohen hierarchischen Level festzulegen. Wird beispielsweise *wheel* als Meronym von *vehicle* festgelegt, scheiden Fahrzeuge wie *snowmobile* aufgrund der fehlenden Räder aus.

3.2.3.4.3 Antonymie Eine besonders in Ressourcen mit Fokus auf die Wortbedeutung (z.B. WordNet, vgl. 3.2.6.1) verwendete Relation ist die Antonymie. Antonyme sind Wortpaare, welche sich in genau einem polaren Merkmal unterscheiden (*lebendig-tot, Ruhe-Bewegung*).

3.2.3.4.4 Entailment oder strikte Implikation Meronymie wird in erster Linie als Relationsart bei der Einteilung von Nomen verwendet. Zur Einteilung von Verben wird im Gegensatz dazu in zahlreichen Taxonomien (z.B. WordNet, vgl. 3.2.6.1) Entailment bzw. strikte Implikation als Relationsart eingesetzt. Entailment zwischen zwei Verben V_1 und V_2 ist dann gegeben, wenn das Verb V_1 den Vorgang von Verb V_2 bedingt. V_1 impliziert somit notwendigerweise V_2:

$$(V_1 \rightarrow V_2)$$

Die Verben *schnarchen* und *träumen* implizieren beispielsweise das Verb *schlafen*, da *schlafen* die Voraussetzung ist, um überhaupt *schnarchen* bzw. *träumen* zu können:

$$(schnarchen \rightarrow schlafen); (träumen \rightarrow schlafen)$$

Entailment ist unilateral, das bedeutet, dass, falls ein Verb V_1 ein Verb V_2 zur Bedingung hat, V_2 unmöglich V_1 zur Bedingung haben kann. Gegenseitiges Entailment würde Synonymie bedeuten. Negation bedingt eine Umkehrung der Entailmentrelation: wer nicht schläft, der schnarcht auch nicht.

$$(schnarchen \rightarrow schlafen); (\neg schlafen \rightarrow \neg schnarchen)$$

3.2.3.4.5 Temporale Relationen Durch temporale Relationen werden zeitliche Abläufe dargestellt. In Ontologien mit linguistischem Hintergrund sind temporale Relationen eher unüblich, in Domain-Ontologien können jedoch durch diese Relationsart standardisierte Abläufe, z.B. Operationsabläufe, in die Ontologie integriert werden.

3.2.3.5 Vererbung

Wenn von einem Konzept einer Hierarchie sämtliche Eigenschaften an ein untergeordnetes weitergegeben werden, wird von Vererbung gesprochen. Das Prinzip der Vererbung hat bereits Aristoteles in seiner Kategorienschrift thematisiert: [5]

> „Wenn eines über etwas anderes ausgesagt wird wie über ein Zugrundeliegendes, dann wird alles, was über das, was ausgesagt wird, ausgesagt. So wird zum Beispiel *Mensch* über diesen bestimmten Menschen ausgesagt, *Lebewesen* über *Mensch*, folglich wird *Lebewesen* auch über diesen bestimmten Menschen ausgesagt. Denn dieser bestimmte Mensch ist sowohl Mensch als auch Lebewesen.“

Werden die Regeln der Vererbung ohne Ausnahme eingehalten, spricht man von strikter Vererbung, ein Grundsatz, der im Aufbau von sowohl terminologischen als auch generellen Ontologien unbedingt eingehalten werden sollte (Nistrup Madsen et al 2004: 90). In manchen Fällen kann es jedoch notwendig sein, statt strikter Vererbung *Default*-Vererbung anzuwenden. Durch diese Art der Vererbung werden Eigenschaften nur typischerweise vererbt, müssen jedoch von Unterklassen, falls erforderlich, nicht übernommen werden, z.B. der flugunfähige Pinguin als Hyponym von *Vogel*, der mit der Eigenschaft *+kann fliegen* ausgestattet ist (Vossen 2003:

[5] Aristoteles, Kategorien, III, deutsche Übersetzung aus: Rath 1998.

467). Hält man sich jedoch an strikte Vererbung, müssen die Features dementsprechend angepasst werden, im Fall der Vögel darf die Eigenschaft *+kann fliegen* nicht zur Unterscheidung von anderen Tieren herangezogen werden, da diese Eigenschaft nicht für ausnahmslos alle Vögel (wie eben Pinguine oder auch Vogelsträuße) gilt. In diesem Fall müssen für das Konzept *Vogel* Eigenschaften gefunden werden, die für alle Vögel gelten und dementsprechend vererbt werden können. Als Hyperonyme des Konzeptes *Vogel* können durch Verwendung der Features *+kann fliegen* bzw. *-kann fliegen* die nichtfliegenden Vögel unter Einhaltung der Vererbungsgrundsätze berücksichtigt werden.

Während in streng hierarchischen Systemen eine Klasse genau einer Überklasse untergeordnet ist, kann es in bestimmten Fällen notwendig sein, einer Klasse mehrere Klassen überzuordnen. In diesem Fall wird dem Konzept von jedem übergeordneten genau ein Feature direkt zugewiesen, womit von multipler Vererbung gesprochen wird. Im Fall von multipler Vererbung werden folglich von allen übergeordneten Konzepten sämtliche Eigenschaften geerbt.

3.2.4 Sind Taxonomien Ontologien?

Es besteht kein allgemeiner Konsens darüber, ob Taxonomien als Ontologien zu behandeln sind. Meist wird der Begriff Ontologie im Hinblick auf Taxonomien sehr vage verwendet (Studer et al. 1998: 185). Es gibt sehr wohl Taxonomien, die als vollständige Ontologien akzeptiert werden; das *Yahoo!-Directory*, eine Taxonomie zur Unterstützung der Yahoo-Websuche, wird beispielsweise grundsätzlich als Ontologie bezeichnet (Lasilla & McGuinness, 2001). Studer et al. (1998: 185) führen zwei Eigenschaften an, die als Unterscheidungsmerkmale zwischen eine Ontologie und einer Taxonomie gesehen werden können: Ontologien haben (1) eine vielfältigere interne Struktur und sind (2) von einem bestimmten Konsens abhängig. Während Taxonomien aus Hyponymie-Relationen aufgebaut sind, sind die Konzepte einer Ontologie über ein breiteres Spektrum an Relationen verbunden. Folglich sind Taxonomien im Vergleich zu Ontologien Grenzen in der Darstellung von bestimmten Wissensrelationen gesetzt. Diffiziler ist der zweite Unterscheidungsgrund, die Frage des Konsens. Für Studer et al. (1998: 186) ist die Antwort auf die Frage *Konsens zwischen wem?* abhängig von einem Kontext:

> „For example, if a hospital is building up an ontology with knowledge about a particular disease - say AIDS - that can be consulted by all doctors in a hospital, then the consensus should be between the doctors envolved. If, on the other hand, a government wants to set up a nationwide network of bibliographic, ontology-based databases that can be

> consulted from nearly every terminal with an Internet connection in the country, then the consensus should be nation-wide (i.e. everybody should accept this ontology as workable).“

Dieser Konsens entspricht der *shared conceptualization* in der Definition von Studer et al. (sh. 3.2.2). Ontologien, die im Grunde Taxonomien entsprechen, werden von elaborierten Ontologien unterschieden, indem die Bezeichnungen *lightweight ontology* bzw. *heavyweight ontology* verwendet werden (Gómez-Pérez et al 2004: 8).

3.2.5 Klassifikation von Ontologien

3.2.5.1 Top-Level-Ontologien

Top-Level-Ontologien (bzw. Upper-level Ontologien oder auch World Models) beschreiben sehr generelle Konzepte und liefern Konzepte, denen sämtliche bereits existierenden Ontologien untergeordnet werden können (Gómez-Pérez et al 2004: 32). Top-Level Ontologien sind künstliche Gebilde, welche von Grund auf vollständig konstruiert werden (Mahesh und Nirenberg 1995a: 1). Aus diesem Grund existieren zahlreiche Top-Level-Ontologien mit den unterschiedlichsten Strukturen (z.B. GUM (Bateman et al. 1995), Mikrokosmos (Mahesh & Nirenburg 1995a)). Um die existierende Heterogenität aufzulösen, arbeitet die IEEE Standard Upper Ontology Working Group an einer standardisierten Upper Ontology, welche als Ausgangspunkt für Domain Ontologien dienen soll (Gómez Pérez et al 2004: 32). Innerhalb der NLP-Forschung hat sich die Meinung durchgesetzt, dass sämtliche NLP-Systeme, welche die Semantik von Texten darstellen oder verändern, nur unter Verwendung einer Ontologie funktionieren (Mahesh & Nirenberg 1995b: 1). Die für derartige Anwendungen konstruierten Ontologien sind grundsätzlich Top-Level-Ontologien.

3.2.5.2 Domain-Ontologien

Während Top-Level-Ontologien möglichst generelle Konzepte beschreiben, dienen Domain-Ontologien der Beschreibung von Konzepten innerhalb einer bestimmten Domäne (z.B. Medizin, Pharmazie etc.). Die Konzepte einer Domänen-Ontologie sind im Idealfall Spezifikationen von in Top-Level Ontologien definierten Konzepten (Gómez Pérez et al 2004: 33). Durch die Anbindung an Top-Level-Ontologien können unterschiedliche Domain-Ontologien miteinander in Relation gebracht werden.

3.2.6 Ontologien für den linguistischen Gebrauch

Neben eher generell gehaltenen Top-Level-Ontologien gibt es eine Reihe von speziell für linguistische Anwendungen konzipierten Ontologien. Der Zweck dieser Ontologien ist die Darstellung von sprachlichen Konstruktionen. Linguistische Ontologien werden für unterschiedliche Anwendungsgebiete konzipiert (z.B. WordNet: Lexikalische Ressource; Mikrokosmos: maschinelle Übersetzung; GUM: Sprachgenerierung), aus diesem Grund wird während der Konzeption der entsprechenden Ontologie, je nach Einsatzgebiet, auf unterschiedliche Kriterien Rücksicht genommen. Ein Kriterium ist die Abhängigkeit von bestimmten Sprachen. Manche Ontologien sind nur für die Verwendung in Verbindung mit einer einzigen Sprache konzipiert (WordNet, vgl. 3.2.6.1), andere sind multilingual, also für mehrere Sprachen konzipiert (GUM, vgl. 3.2.6.3). Einige Ontologien (z.B. EuroWordNet, vgl. 3.2.6.1) bestehen sowohl aus sprachabhängigen als auch sprachunabhängigen Elementen oder sind absolut unabhängig von bestimmten Sprachen (Mikrokosmos, vgl. 3.2.6.4) (Gómez-Pérez 2004: 79). Ein weiteres Kriterium betrifft die sprachliche Realisierung der durch die Ontologie dargestellten Konzepte bzw. deren Instanzen. Während die meisten Ontologien mit der Absicht konstruiert wurden, den jeweiligen Konzepten bzw. Instanzen sprachliche Realisierungen in Form von Wörtern zuzuordnen, wird in Ontologien wie GUM die Verknüpfung mit größeren grammatikalischen Einheiten wie Phrasen oder ganzen Satzgliedern forciert (Bateman et al. 1995: 6). Ferner kann das Verhältnis zwischen Konzepten und deren Instanzen als Kriterium herangezogen werden. In manchen Ontologien entspricht ein Konzept genau einem Wort einer natürlichen Sprache, in anderen entsprechen Konzepte keinem einzigen oder aber auch mehreren Wörtern (z.B. Mikrokosmos) (Gómez-Pérez 2004: 79). Besonders im Bereich der linguistischen Verwendung von Ontologien stellt sich natürlich die Frage, inwieweit sich eine Ontologie von einem Lexikon unterscheidet. Lexika beinhalten tatsächlich zu einem bestimmten Grad Information, welche in einer Ontologie gespeichert sind und eine Grenze kann nicht eindeutig gezogen werden. Aus diesem Grund sind je nach Anwendungsgebiet relativ große Überschneidungen hinsichtlich des Informationsgehalts der beiden Konzepte möglich. Ist die enthaltene Information jedoch von eher linguistischer Natur (PoS etc.), kann von einem Lexikon gesprochen werden (Mitkov 2003, 465). Mahesh (1996) unterscheidet Lexika von Ontologien folgendermaßen: sprachunabhängige Informationen sind in Ontologien gespeichert, während sprachabhängige Informationen Teil des Lexikons sind.

3.2.6.1 WordNet

WordNet ist eine speziell für Englisch entwickelte Datenbank, welche unter Berücksichtigung von psycholinguistischen Theorien aufgebaut ist (Gómez-Pérez 2004: 79). Die aktuelle Version, WordNet 3.0.[6], enthält etwa 80 000 Nomen, die in ca. 60 000 Konzepten organisiert sind. Eigennamen bzw. Proper nouns wurden nicht speziell berücksichtigt. Die elementare semantische Relation in WordNet ist Synonymie, sogenannte *Synsets* sind die grundlegenden Bausteine der Ressource. Ein Synset ist eine Menge von Wörtern, welche (in einem bestimmten Kontext) ausgetauscht bzw. als Synonyme verwendet werden können (Vossen 1997: 73). Ein mögliches Synset für das Konzept *motorgetriebenes, vierrädriges Fortbewegungsmittel* wäre folglich *[car, auto, automobile, machine, motorcar]*. Die in WordNet verwendeten Synsets sind nicht äquivalent zu Wörterbucheinträgen. In der Regel haben Einträge für polyseme Wörter unterschiedliche Erklärungstexte, während einem Synset eine einzige Erklärung zugrunde liegt. Ein einziger Wörterbucheintrag kann semantische Informationen enthalten, die in WordNet auf mehrere Synsets verteilt sind.

3.2.6.1.1 Lexikalische Relationen zwischen Nomen in WordNet Die primäre Hierarchie in WordNet besteht, wie in den meisten vergleichbaren Ressourcen, aus Hyponymie-Relationen. Wie allgemein in Taxonomien üblich, wird in WordNet strikt zwischen Konzepten und Wörtern (Elementen) unterschieden. Ein Wort kann niemals ein Hyponym eines anderen Wortes sein, sondern nur in terminaler Position als Hyponym eines Konzeptes. In der Regel verfügen Taxonomien bzw. Ontologien über ein einziges Top-Level-Konzept bzw. einen höchsten Knoten, von dem sämtliche Konzepte abzweigen. Die Nomen in WordNet werden jedoch in mehrere unterschiedliche Hierarchien unterteilt, wobei jede dieser Hierarchien über ein einziges Top-Level-Konzept verfügt. Diese Partitionierung bringt laut Miller (1998) unter anderem den Vorteil einer besseren Übersichtlichkeit. Für WordNet wurden 25 Top-Level-Konzepte ausgewählt, als Kriterium zur Unterscheidung wurden mögliche Adjektiv-Nomen-Kombinationen herangezogen. Diese 25 Top-Level-Konzepte können in weiterer Folge unter Berücksichtigung von bestimmten Eigenschaften weiter gruppiert und unter 11 *unique beginners* genannte Konzepte zusammengefasst werden. Meronymie-Relationen werden in WordNet in erster Linie in den Nomenhierarchien *body, artifact* und *quantity* verwendet. Von den in 3.2.3.4.2 angeführten Meronymie-Typen sind die drei Typen *Component-object, Member-collection* und *Stuff-object* codiert, wobei die Relation *Component-object* in WordNet mit Abstand am häufigsten zur Anwendung kommt.

[6]Datum: 10.06.2006

verbs of motion	verbs of perception	verbs of contact
verbs of communication	verbs of competition	verbs of change
verbs of cognition	verbs of consumption	verbs of creation
verbs of emotion	verbs of possession	verbs of bodily care and functions
verbs referring to social behaviour and interactions		

Tabelle 3.3: Unterteilung der Zustandsverben in WordNet (nach Fellbaum 1998: 70)

3.2.6.1.2 Lexikalische Relationen zwischen Verben in WordNet Die Verben innerhalb von WordNet wurden in zwei Hauptgruppen unterteilt: Verben, die Aktionen und Abläufe (*actions* und *events*) beschreiben sowie Verben, die Zustände (*states*) beschreiben. Die Zustandsverben wurden abermals in 14 spezifischere Domänen unterteilt (Tab. 3.3).

Statische Verben, die dem Konzept *sein* (TO BE) entsprechen (Modalverben), bilden eine eigene, semantisch heterogene Klasse. Zu dieser Klasse gehören ferner Auxiliare. Die Grenzen zwischen den für WordNet gewählten Verbdomänen sind sehr vage, da zahlreiche Verben nicht eindeutig einer einzigen Domäne zugeordnet werden können (die Verben *wonder, speculate, confirm* können z.B. sowohl als *verbs of cognition* als auch als *verbs of communication* eingestuft werden). Derartige Verben werden in WordNet als polyseme Wörter behandelt und zwei oder mehreren unterschiedlichen semantischen Domänen zugeordnet (vgl. Fellbaum 1998: 71). Analog zu den Nomen sind in WordNet auch die Verben in Synsets gruppiert. Da jedoch für das Englische nur sehr wenige tatsächlich synonyme Verben existieren, und folglich die Synsets in vielen Fällen aus nur einem einzigen Wort bestanden hätten, wurden Verben, die zumindest das gleiche Konzept repräsentieren, in Synsets zusammengefasst (sog. *near-synonyms*; vgl. Fellbaum 1998: 73). Dies betrifft in erster Linie Verben, deren gegenseitige Substitution aufgrund von unterschiedlichen kontextuellen Umständen nur eingeschränkt möglich ist (vgl. *purchase* ⇒ formaler als *buy*; vgl. Fellbaum 1998: 73). Idiomatische Verbalphrasen wie *kick the bucket* und *keep an eye on* wurden in WordNet berücksichtigt und in die entsprechenden Synsets integriert (sh. Abb. 3.6).

3.2.6.2 EuroWordNet

Das EuroWordNet (Vossen 1998), ein Projekt, an dem zahlreiche europäische Institutionen teilnahmen, wurde mit dem Ziel, andere europäische Sprachen in das WordNet zu integrieren, gestartet (Hanks 2004: 59). Ursprünglich wurden nur

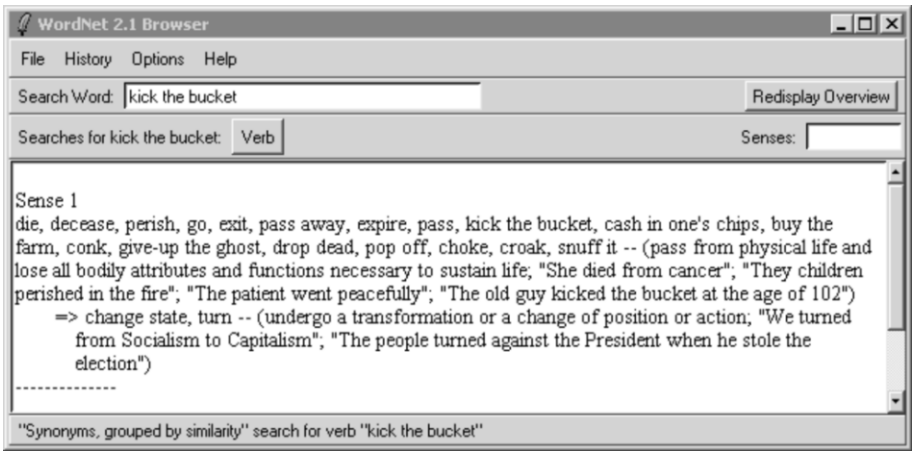

Abbildung 3.6: Synset für die idiomatische Verbalphrase *kick the bucket*, aus WordNet 2.1.

Deutsch, Italienisch, Spanisch und Englisch berücksichtigt, später wurde EuroWordNet u.a. durch Französisch und Tschechisch erweitert (Vossen 1997: 75). Die Struktur der semantischen (Haupt-)Relationen von EuroWordNet entspricht jener von WordNet, einige Änderungen wurden jedoch vorgenommen. Aufgrund von sprachabhängigen Unterschieden zwischen den Synsets der einzelnen Sprachen wurden in EuroWordNet zusätzliche, sprachspezifische Relationen eingeführt. Die Wordnets der einzelnen Sprachen sind durch einen Index (*Inter Lingual Index*, kurz *ILI*), miteinander verbunden. Jedem Synset einer Sprache entspricht ein sprachunabhängiges Synset im ILI, womit alle monolingualen Synsets mit dem gleichen Eintrag im Index als äquivalent gelten (Vossen 1997: 77). Dadurch entsteht die Möglichkeit, auf Wörter, welche einem Wort einer anderen Sprache semantisch entsprechen, zuzugreifen. Zusätzlich beinhaltet EuroWordNet zwei Ontologien, welche an den ILI gebunden sind. Die *Top-Concept-Ontology* ist eine Hierarchie von sprachunabhängigen Konzepten, welche semantische Unterscheidungen beschreiben, z.B. *Object, Substance, Location* etc. Die *Domain-Ontology* ordnet Bedeutungen nach Thematik, beispielsweise *Traffic, Road-Traffic, Air-Traffic, Sports* etc. Über den ILI können die Konzepte der beiden Ontologien auf sprachenspezifische Bedeutungen übertragen werden (Vossen 1997: 80).

Obwohl das Design sowohl der Datenbank als auch des Inter Lingual Index 1999 abgeschlossen wurde, werden weiterhin WordNets für das EuroWordNet entwickelt. Durch den ILI können die neuen Elemente in die EuroWordNet-Datenbank integriert und dadurch mit den bereits enthaltenen Sprachen in Relation gebracht werden (Gómez-Pérez 2004: 79). Derzeit werden WordNets für u.a. folgende Sprachen entwickelt: Schwedisch, Norwegisch, Dänisch, Baskisch, Rumänisch, Bulgarisch, Slowenisch. EuroWordNet wird in linguistischen Anwendungsgebieten wie semantisches Tagging von Texten, Interlingua-Repräsentation für *Information Retrieval*

und maschinelle Übersetzung (Hanks 2004: 59) und *Cross-Language-Information Retrieval* (Vossen 1997: 75) verwendet.

3.2.6.3 The Generalized Upper Model

Das *Generalized Upper Model (GUM)* ist eine in den 1980er Jahren begonnene Weiterentwicklung des *Penman Upper Model* (Bateman et al. 1995: 12). Im Gegensatz zu WordNet und EuroWordNet beschreibt GUM nicht die Semantik von Wörtern, sondern die Semantik von größeren grammatikalischen Einheiten (Nominalgruppen, Phrasen etc.) (Gómez-Pérez 2004: 82). GUM besteht aus zwei Hierarchien: eine Hierarchie repräsentiert Konzepte (gekennzeichnet durch den Top-Knoten *Um-Thing*), während die andere Hierarchie Relationen darstellt (*Um-Relation*).

Verwendung findet GUM vor allem in der Textgenerierung. Anfänglich wurde GUM nur für die Generierung von englischen Texten, in weiterer Folge jedoch auch multilingual eingesetzt (Bateman et al. 1995: 7)

3.2.6.4 Die Mikrokosmos-Ontologie

Das zentrale Ziel des an der New Mexico State University und einigen weiteren Institutionen Mitte der 90er Jahre gestarteten Mikrokosmos-Projektes (kurz myK) war die Entwicklung eines Systems, welches aus einem Text einer bestimmten Sprache eine sprachunabhängige Repräsentation (*Text Meaning Representation*; kurz *TMR*) generiert (Mahesh und Nirenberg 1995a: 1). Diese TMR sollten als Ausgangsdaten für ein wissensbasiertes, maschinelles System für die Übersetzung zwischen Spanisch und Englisch verwendet werden. Die durch die Analyse des Input-Textes der Quellsprache generierte, einer Interlingua-Repräsentation entsprechende TMR diente als Input für den Generator der Zielsprache. Zu diesem Zweck wurde eine weitgehend sprachunabhängige Ontologie entwickelt. Als Ausgangsdaten wurden Ontologien, welche für frühere Projekte der Carnegie Mellon Universität entwickelt wurden, herangezogen. Doch sowohl inhaltlich als auch qualitativ unterscheidet sich die Mikrokosmos-Ontologie deutlich von den Vorgängern (Mahesh und Nirenburg 1995a: 2). Da die Texte der zu übersetzenden Sprache bzw. Quellsprache ohne Einschränkungen verwendet wurden, musste die Ontologie ein möglichst breites Feld abdecken und somit absolut domänen-unabhängig sein (Mahesh und Nirenburg 1995a: 2). Ausgehend von den drei Top-Konzepten OBJECT, EVENT und PROPERTY erreicht die Ontologie eine Tiefe von teilweise mehr als 10 Ebenen bei durchschnittlich 5 Verzweigungen pro Knoten. Sprachneutrale (bzw. sprach-unabhängige) Bedeutungen sind in der Ontologie gespeichert, während die sprachspezifische Information im jeweiligen Lexikon gespeichert ist (Mahesh und

Nirenburg 1995a: 4). Somit stellt die Ontologie eine Möglichkeit zur Trennung zwischen sprachunabhängigem Weltwissen und linguistischem Wissen dar.

3.2.6.4.1 Aufbau der Mikrokosmos-Ontologie Die drei obersten Konzepte der Mikrokosmos-Ontologie sind OBJECT, EVENT und PROPERTY (Abb. 3.7). Von diesen Konzepten aus verzweigt sich die Ontologie taxonomisch. Jedes Konzept der Ontologie ist durch einen *Frame* repräsentiert. Innerhalb dieser Frames befindet sich Information darüber, wie die jeweiligen Konzepte mit anderen Konzepten in Relation stehen. Abb. 3.9 zeigt einen Frame für das Konzept *ACQUIRE*. Ein Frame besteht aus einer Reihe so genannter *Slots*, welche das Konzept defi-

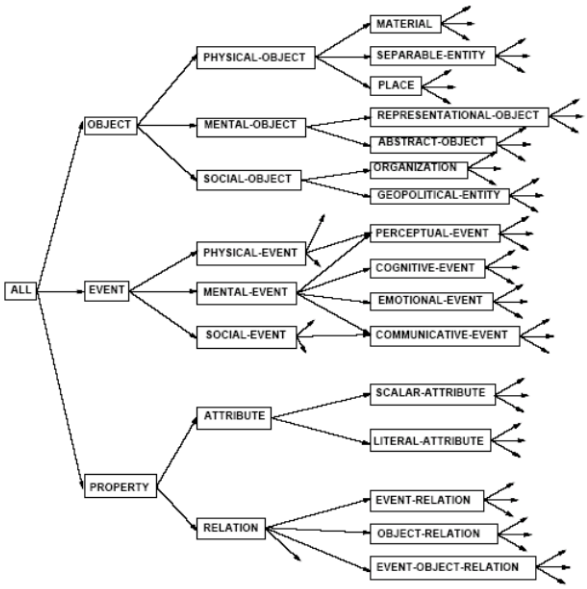

Abbildung 3.7: *Top-Level-Hierarchie* der Mikrokosmos-Ontologie

nieren. Ein Frame besteht aus folgenden Slots, welche jeweils aus einem Attribut und einem dazugehörenden Wert bestehen: ein Slot mit der Bezeichnung *Definition* enthält Information über das Konzept, welche nur für Suchzwecke notwendig ist; ein *time-stamp* genannter Slot mit für die Buchführung relevanter Information, Slots mit taxonomischer Information (Subklassen von Konzepten, Instanzen etc.) und weiteren Slots mit grammatikalischen Merkmalen (Topic, Agent etc.).

Die Verbindung zwischen Lexikon und Ontologie wird hergestellt, indem die Bedeutung des Lexikoneintrages auf ein Konzept übertragen wird. Die Slots der Frames des entsprechenden Konzepts werden mit der Information, welche für eine eindeutige Definition der Lexikonbedeutung notwendig sind, befüllt. Das spanische Verb *adquirir* (Lexikoneintrag in Abb. 3.8) ist eine Instanz des Konzepts *ACQUIRE* mit dem entsprechenden Frame (Abb. 3.9).

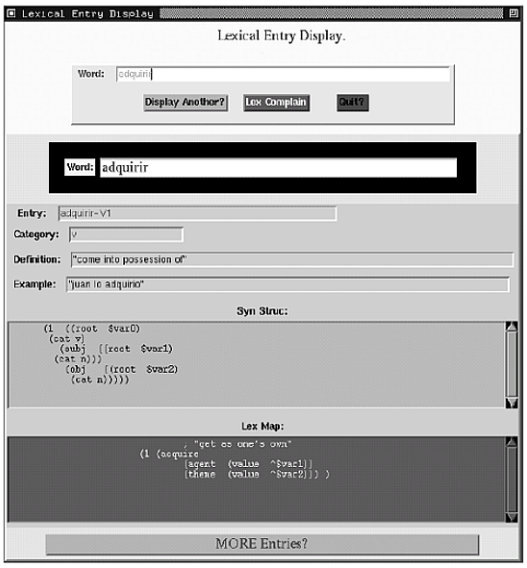

Abbildung 3.8: Mikrokosmos-Lexikoneintrag für das spanische Verb *adquirir* (aus Mahesh und Nirenburg 1995b: 11)

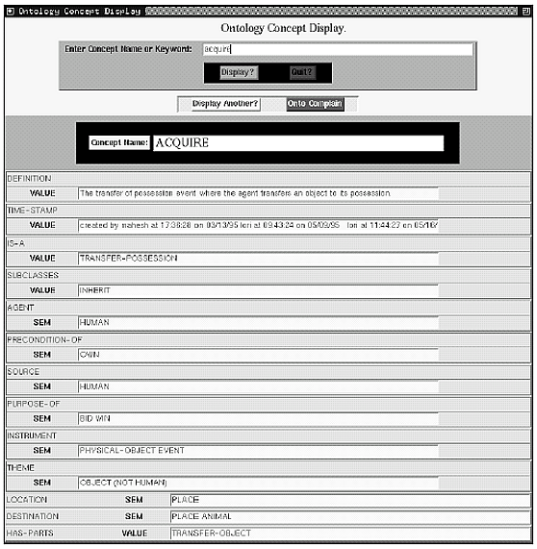

Abbildung 3.9: Mikrokosmos-Frame für das Konzept *ACQUIRE* (aus Mahesh und Nirenburg 1995b: 12).

3.3 Wahl der den Selektionsrestriktionen in SCall zugrundeliegenden Taxonomie

Aufgrund der günstigen Konzeption bestand der ursprüngliche Plan darin, die Konzepte der Mikrokosmos-Ontologie als Grundlage für die Selektionsrestriktionen in SCall zu verwenden. Dieses Unterfangen stellte sich jedoch (leider aufgrund von sehr profanen Gründen) als nicht durchführbar heraus. Die Arbeiten an Mikrokosmos wurden zwischenzeitlich eingestellt und scheinbar besteht keine weitere Interesse in der Weiterführung des Projektes. Es existiert zwar nach wie vor eine Projektho-

mepage[7], der Funktionsumfang ist jedoch eingeschränkt und in erster Linie die für den Aufbau der Ontologie wichtigen Ressourcen sind nicht mehr abrufbar. Auch die am Projekt teilnehmenden Personen haben, wie es scheint, keine weitere Interesse an Mikrokosmos, anders ist nicht erklärbar, dass sämtliche Versuche der Kontaktaufnahme via Email ignoriert wurden.

Aus diesen Gründen wurde die Idee, Mikrokosmos als Grundlage der Selektionsrestriktionen zu verwenden, verworfen. Aufgrund der Integration von Deutsch und Schwedisch wäre sicherlich EuroWordNet eine sinnvolle Alternative gewesen, da die Ressource jedoch kostenpflichtig ist, wurde auf das kostenlose WordNet als Alternative zurückgegriffen. Die Integration der WordNet-Konzepte als Selektionsrestriktionen in SCall wird in 5.4.1 erläutert.

[7] URL: http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-htmls/onto.index.html [10.05.2008]

Kapitel 4

Das WWW als Korpus zur statistischen Semantikkontrolle

4.1 Einleitung

Trotz der in 3.1 gezeigten Methode der Annotation des Lexikons mit Selektionsrestriktionen kann die Generierung von semantisch fehlerhaften Konstruktionen aus zahlreichen Gründen nicht ausgeschlossen werden. Einerseits besteht die Gefahr, dass die Übergeneralisierung der Selektionsrestriktionen von Nomen in bestimmten Kontexten zu einem zu breiten Spektrum an möglichen Konstruktionen führt. Demgemäß kann die Annotation des Nomens *Brot* mit der Selektionsrestriktion *[food, solid food]* anstelle der restriktiveren Restriktion *[baked goods]* im Satz *Der Bäcker bäckt das Brot* die Generierung von Sätzen wie *?Der Bäcker bäckt die Kartoffeln* zur Folge haben (natürlich nur unter der Voraussetzung, dass *Kartoffel* ebenfalls mit der Restriktion *[food, solid food]* annotiert ist). Andererseits kann selbst bei äußerst restriktiver Vergabe der Selektionsrestriktionen niemals ausgeschlossen werden, dass aufgrund von fehlendem Weltwissen, das durch die Restriktionen unmöglich abgedeckt werden kann, inkorrekte Konstruktionen generiert werden (z.B. *der Löwe frisst die Antilope* vs. *?die Maus frisst die Antilope*). Um die von SCall generierten Sätze bzw. Teilstrukturen dieser Sätze auf semantische Richtigkeit zu kontrollieren und gegebenenfalls durch korrekte zu ersetzen, kommt in SCall ein statistisches, korpusbasiertes Verfahren zur Anwendung. Die grundsätzliche Überlegung ist folgende: Können die von SCall generierten Konstruktionen in einem bestimmten Korpus mit einer ausreichend hohen Frequenz gefunden werden, gelten sie als korrekt. Liegt die Anzahl der Funde der betreffenden Struktur jedoch unter einem bestimmten Wert, wird die Sequenz als semantisch nicht korrekt klas-

sifiziert und der entsprechende Satz entweder abgelehnt oder die inkorrekte durch eine neu zu generierende Phrase ersetzt.

Als Korpus für einen derartigen Test kommen grundsätzlich zwei verschiedene Korpusarten in Frage. Einerseits ein nach bestimmten Richtlinien speziell für linguistische Zwecke konstruiertes Korpus wie etwa das englische *British National Corpus* (kurz *BNC*) oder das deutsche *Tiger-Korpus*. Andererseits bietet sich als Alternative zu diesen Ressourcen das World Wide Web (WWW) an, welches zunehmend als Korpus auch für linguistische Zwecke Verwendung findet (z.B. Bickel 2006). Im folgenden Kapitel wird das WWW auf seine Tauglichkeit als linguistisches Korpus untersucht und es werden Vor- und Nachteile gegenüber klassischen Korpora aufgezeigt.

4.2 Die Eignung des WWW als linguistisches Korpus

Der linguistische Begriff Korpus umfasst im weiteren Sinn sprachliche Daten, die als Grundlage für wissenschaftliche Untersuchungen dienen (Glück 2000: 384). Dieser Definition entspricht ein sehr breites Spektrum an Ressourcen (u.a. Tonbandaufnahmen aus der Feldforschung, transkribierte Interviews etc.), in der Computerlinguistik und verwandten Disziplinen bezeichnet der Begriff Korpus in der Regel jedoch eine Sammlung von Texten in einer elektronischen Datenbank (vgl. Graeme 1998: 3). Auf diese Definition wird sich der Begriff Korpus in weiterer Folge beziehen. Aus historischer Sicht müssen Korpora jedoch nicht zwingend in elektronischer Form vorliegen. Vor dem ersten Aufkommen von Computern und der damit verbundenen Möglichkeit der Erstellung elektronischer Korpora existierten bereits große Ressourcen wie das für Wilhelm Kaedings *Häufigkeitswörterbuch der Deutschen Sprache* (vgl. McEnery (2003: 452)) zugrundeliegende Korpus, das aus Texten aus dem 19. Jahrhundert zusammengestellt wurde und mehr als 10 Millionen Wörter umfasste. Die ersten elektronischen Korpora kamen nach dem Auftreten der ersten Computer in den 1950er Jahren auf (McEnery 2003: 452). Bis in die 1980er Jahre des letzten Jahrhunderts stieg schließlich sowohl die Anzahl der verfügbaren Korpora, als auch die in diesen Korpora enthaltenen Datenmengen, mit der Konsequenz, dass ab den 1990er Jahren erstmals sogenannte *Very large corpora*, Korpora mit einem Umfang im Bereich von ab 100 Millionen Wörtern, zur Verfügung standen (z.B. British National Corpus (BNC) mit 100 Millionen enthaltenen Wörtern) (McEnery 2003: 452). Die größte derzeit für Deutsch verfügbare Ressource ist das Korpus des Instituts für deutsche Sprache *Cosmas* (*Corpus Search,*

Management and Analysis System) mit 940 Millionen öffentlich zugänglichen Textwörtern (Duffner & Näf 2006: 10). Im Gegensatz zu Korpora wie Cosmas und dem BNC, die im Grunde nur Text, jedoch keine Metadaten enthalten, sind annotierte Korpora mit zusätzlicher, linguistischer Information wie *Part-of-Speech (PoS)* oder Lemmata versehen. Diese Korpora sind aufgrund des während der Erstellung notwendigen, hohen Arbeitsaufwandes (noch) bedeutend kleiner als Ressourcen wie das BNC, können jedoch aufgrund der zusätzlichen Annotation für linguistische Anwendungen wie das Training von PoS-Taggern und dergleichen verwendet werden. Für Deutsch existiert als annotiertes Korpus u.a. das Negra-Korpus mit einer Größe von 355.096 Tokens (20.602 Sätze aus Zeitungstexten aus der Frankfurter Rundschau). Für Schwedisch (ca. 8,5 Millionen Sprecher) wurde das Korpus *Stockholm Umeå Corpus Version 2.0* (*SUC 2.0*, Gustafson-Capková & Hartmann 2006) entwickelt, das über knapp eine Million Wörter verfügt, wobei sämtliche Wörter mit PoS, morphologischen Informationen, Lemmata und weiteren Tags annotiert sind (Gustafson-Capková & Hartmann 2006: 2).

Neben den erwähnten, speziell für linguistische Zwecke aufgebauten Korpora existiert mit dem Internet ein riesiges, stetig wachsendes Archiv an Textmaterial, das durchaus als Korpus verstanden werden kann (Bickel 2006: 72). Über die tatsächliche Eignung des WWW als Korpus für linguistische bzw. sprachtechnologische Zwecke existieren jedoch unterschiedliche Meinungen. McEnery (2003: 449) versteht unter einem linguistischen Korpus eine sehr spezifische Sammlung sprachlicher Daten:

> „The term corpus should properly only be applied to a well-organized collection of data, collected within the boundaries of a *sampling frame* designed to allow the exploration of a certain linguistic feature (or set of features) via the data collected."

In manchen Publikationen wird neben dem Korpus der Begriff des Text-Archives (engl. *text archive*) verwendet (Kennedy 1998: 4). Im Gegensatz zu einem Korpus mit seinen spezifischen Eigenschaften (systematische, geplante und strukturierte Sammlung von Texten, speziell für linguistische Zwecke erstellt) ist ein Text-Archiv ein im Normalfall nicht strukturierter Textfundus, der nach opportunistischen Gesichtspunkten zusammengetragen wurde (Leech 1991: 11). Nimmt man die Definition McEnery's als Maßstab zur Beurteilung, inwieweit das WWW als Korpus gesehen werden kann, muss diese Frage klar negiert werden. Die Daten des WWW sind weder gut organisiert, noch wurden diese Daten innerhalb eines bestimmten Rahmens gesammelt. Das WWW ist vielmehr eine sehr arbiträre Anhäufung von Daten, die von Millionen von Benutzern aufgrund unterschiedlicher Interessen und

Anforderungen täglich erweitert wird (Bickel 2006: 72). Gemäß McEnery's Definition ist das WWW folglich kein zur linguistischen Verwendung geeignetes Korpus und entspricht eher einem Archiv. Kilgarriff & Grefenstette (2003: 2) bemängeln an Definitionen wie jenen von McEnery, dass die zwei grundsätzliche Fragestellungen *Was ist ein Korpus?* und *Was ist ein gutes Korpus (abhängig vom linguistischen Einsatzgebiet)?* vermischt werden. In derartigen Definitionen wird demnach weniger der Frage nachgegangen, ob ein bestimmtes Korpus gut für eine bestimmte linguistische Anwendung ist, sondern vielmehr der semantischen Frage, ob dieses Korpus *überhaupt* als solches deklariert werden kann bzw. deklariert werden darf. Kilgarriff & Grefenstette (2003: 2) bezeichnen folglich jede Sammlung von Texten als Korpus, wobei sie als Faktor die Domäne bzw. den Kontext, in welchem der Term benutzt wird, miteinbeziehen: „a corpus is a collection of texts when considered as an object of language or literary study" (Kilgarriff & Grefenstette 2003: 2). Gemäß dieser Definition kann das WWW sehr wohl als Korpus gesehen werden.

4.2.1 Größe der durch das WWW zugänglichen Textmenge

Der Begriff *WWW als Korpus* ist etwas ungenau, da angenommen werden kann, dass unter dieser Bezeichnung die Verwendung sämtlicher Seiten bzw. Dokumente im WWW als Korpus verstanden werden kann. Faktisch ist es jedoch im Grunde unmöglich, auf die riesige Anzahl der Dokumente, die über das WWW abrufbar sind, zuzugreifen, da die entsprechenden Suchwerkzeuge nicht existieren. Vielmehr stehen als Ressource jene Daten zur Verfügung, die im Index der gängigen Suchmaschinen (Google, Yahoo, AltaVista etc.) erfasst sind. Auf genau diese Datensätze, deren Anzahl im Verhältnis zur Gesamtanzahl existierender Webdokumente gesehen gering ist, wird in dieser Arbeit in weiterer Folge referiert, wenn von *WWW als Korpus* gesprochen wird. Mittels Suchmaschinen wie *Yahoo* oder dem inzwischen beinahe-Monopolisten *Google* kann zwar nur ein Bruchteil der im gesamten WWW befindlichen Dokumente durchsucht werden, laut Angaben von Google liegt die Anzahl von Dokumenten im Index der Suchmaschine jedoch bei immerhin mehr als 8 Milliarden Webseiten[1]. Die Anzahl deutschsprachiger Seiten im Index von Google schätzte Bickel (2006: 75) auf etwa 1,1 Milliarden, jene der Seiten im Index von AltaVista auf etwa 850 Millionen, wobei davon ausgegangen werden kann, dass sich die Anzahl seit dem Zeitpunkt der Erhebung im Jahr 2006 erhöht hat. Diese Zahlen stellen jedoch nur großzügige Schätzungen dar und wurden ermittelt, indem die Anzahl der Seiten, die über eine Domain eines deutschsprachigen Landes (z.B. .at und .de) verfügbar waren, addiert wurden (Bickel 2006: 75). Wie jedoch Bickel richtig anmerkt, sind derartige domainspezifische Anfragen in den letzten

[1] http://www.google.com/intl/de/options/ [31.01.2008]

Jahren erschwert worden, da in den betreffenden Ländern zwischenzeitlich neben den „klassischen“ Länderdomains zusätzliche Domains wie *.eu* oder *.tv* in Verwendung sind. Während also bereits die zahlenmäßige Erfassung der Menge der durch die Indizes der Suchmaschinen abgedeckten Dokumente schwer ist, gestaltet sich die Einschätzung der dadurch verfügbaren Textmenge (z.B. Anzahl der Wörter) noch komplexer, da die Textmenge in Webdokumenten von wenigen Wörtern bis hin zu einigen A4-Textseiten erreicht und daher nur sehr vage abgeschätzt werden kann (Bickel 2006: 75).

Lawrence & Giles (1999: 107) berechneten die Menge des in den Indizes der damals gängigen Suchmaschinen enthaltenen Textes annäherungsweise, indem sie davon ausgingen, dass eine Seite durchschnittlich ca. 18 Kilobyte Text beinhaltet, nach Bereinigung von nicht-textrelevanten Elementen wie HTML-Tags 7 bis 8 Kilobyte. Für die etwa 800 Millionen indexfähigen Webseiten im Jahr 1999 wurde folglich eine Menge von zirka 6 Terabyte aufrufbarer Text ermittelt. Für 2003 schätzten Kilgarriff & Grefenstette (2003: 5), ausgehend von Eigenangaben von Google, dass alleine durch die im Index von Google enthaltenen Dokumente eine Textmenge von mehr als 20 Terabyte abrufbar ist. Unter der Annahme, dass ein Wort, großzügig geschätzt, 10 Byte groß ist, errechneten Kilgarriff & Grefenstette eine Anzahl von etwa 2 Billionen Wörtern. Nimmt man die von Google aktuell angegebenen 8 Milliarden Seiten, die immerhin in etwa die zehnfache Menge der von Lawrence & Giles 1999 angenommenen Seiten ausmachen, ist die Anzahl der in diesen Dokumenten enthaltenen Wörter dementsprechend größer.

Einen alternativen Ansatz, die Anzahl der Wörter, die durch eine Suchmaschine abrufbar sind, zu eruieren, verfolgten Grefenstette & Nioche (2000: 238). Nachdem Funktionswörter wie *the, with, in etc.* in Texten in relativ stabiler Frequenz auftreten (vergl. dazu 4.2.2.4), kann die Anzahl der Funktionswörter von einem Korpus von bekannter Größe ermittelt und davon ausgehend die Anzahl dieser Wörter in einem Vergleichskorpus extrapoliert werden. In kleineren Textmengen wurden folglich pro untersuchter Sprache die jeweils 100 häufigsten Tokens ermittelt. Jene Tokens, die in mehr als einer Sprache vorkommen, wurden gestrichen, um die Verfälschung von Ergebnissen durch fremdsprachliches Material zu verhindern (z.B. *que*, das in zahlreichen romanischen Sprachen mit hoher Frequenz gefunden werden kann). Von den übrig gebliebenen Tokens wurden wiederum die 20 häufigsten als Prädiktoren für die weitere Bearbeitung ausgewählt. Die Häufigkeit dieser Prädiktoren in den im Index von AltaVista befindlichen Seiten wurde durch einfache Suche nach den Tokens ermittelt. Nachdem die Größe der Testkorpora und die relative Häufigkeit der Tokens bekannt waren, konnte mit Hilfe der durch die Suchabfragen ermittelten Web-Häufigkeit ein Wert für die Größe der Web-Textmenge extrapoliert werden.

Für das Jahr 2000 schätzten Grefenstette & Nioche (2000: 245) demnach die Anzahl der englischen Wörter auf 48 Milliarden, im März 2001 war dieser Wert bereits auf mehr als 76 Milliarden gestiegen (Kilgarriff & Grefenstette 2003: 7). Unter Verwendung der von Grefenstette & Nioche (2000: 239) ermittelten Prädiktoren und deren Frequenzen (Abb. 4.1, Spalten 1 bis 4) sowie den entsprechenden Suchergebnissen auf Google (Datum: 01.02.2008, Spalte 5) kann die derzeitige Menge an deutschen Wörtern im Google-Index annäherungsweise auf mehr als 150 Milliarden Wörter geschätzt werden[2].

Predictor	*Relative Frequency*	*Counts AltaVista (Feb. 2000)*	*Single Word Prediction*	*Counts Google (Feb. 2008)*	*Single Word Prediction*
oder	0,00561180	13566463	2417488685	995000000	177304964539
sind	0,00477555	11944284	2501132644	1080000000	226151961554
auch	0,00581108	15504327	2668062907	829000000	142658507541
wird	0,00400690	11286438	2816750605	605000000	150989543038
nicht	0,00646585	18294174	2829353295	1060000000	163938229312
eine	0,00691066	19739540	2856389983	952000000	137758188075
sich	0,00604594	17547518	2902363900	941000000	155641637198
ist	0,00886430	26429327	2981546992	1280000000	144399444965
auf	0,00744444	24852802	3338438083	1360000000	182686676231
und	0,02892370	101250806	3500617348	2110000000	72950556118
Average Prediction			*2881214444*		*155447970857*

Abbildung 4.1: Schätzung der im Index von Google enthaltenen deutschen Wörter

4.2.2 Vorteile des WWW gegenüber klassischen Korpora

4.2.2.1 Einfacher Zugang

Vorausgesetzt, man verfügt über eine Internetverbindung, kann im WWW mittels einer Suchmaschine innerhalb von sehr kurzer Zeit auf sehr große Datenmengen zurückgegriffen werden. Spezielle Korpora wie das Negra-Korpus stehen in der Regel für wissenschaftliche Zwecke frei zur Verfügung, verlangen jedoch eine Registrierung sowie eine Unterzeichnung einer Lizenz, was mit einem gewissen zeitlichen Aufwand verbunden ist.

4.2.2.2 Multilingualität

Während Korpora jeweils nur für Texte einer Sprache erstellt werden, enthält das WWW Textmaterial für eine sehr große Anzahl an Sprachen. Unter der Verwendung von Suchmaschinen können zwar sprachspezifische Versionen benutzt und

[2] Aufgrund von fehlenden Daten wurden nur 10 statt der ursprünglich 20 Prädiktoren zur Berechnung verwendet.

entsprechend den jeweiligen Sprachen eingestellt werden (die deutsche Ausgabe von Google kann beispielsweise sowohl nach deutschen Seiten als auch explizit nach Seiten auf Deutsch suchen), ohne derartige Spezifikationen kann jedoch auch sprachunabhängig gesucht werden. Bei der auf eine bestimmte Sprache reduzierten Suche besteht jedoch immer eine gewisse Gefahr von „verunreinigten“ Ergebnissen durch fremdsprachliches Material, welches dem gesuchten Muster entspricht. Sucht man beispielsweise nach dem Vorkommen für die deutsche Präposition „mit“, werden auch Treffer für das durch *MIT* abgekürzte Massachusetts Institute of Technology im Suchergebnis berücksichtigt.

4.2.2.3 Aktualität

Viele Korpora decken sowohl nur gewisse Textgenres (Negra-Korpus: Zeitungstexte aus der Frankfurter Rundschau) als auch zeitlich abgegrenzte Texte (SUC: 1990er Jahre (Gustafson-Capková & Hartmann 2006: 2)) ab. Neue sprachliche Entwicklungen werden entweder gar nicht oder aufgrund des mit der Korpuserweiterung verbundenen Arbeitsaufwandes mit Verspätung berücksichtigt. Aus diesem Grund sind Korpora im Bezug auf Phänomene des aktuellen Sprachgebrauchs (wie Neologismen etc.), welche durch das WWW Einzug in den Sprachgebrauch finden, teilweise sehr mangelhaft ausgestattet (Renouf et al. 2005: 1). Ein Beispiel ist das Verb *googeln*, ein ab ca. 2003 aufgekommener Neologismus, welcher 2004 in den Duden aufgenommen wurde. Im Negra-Korpus, das aus Texten besteht, die vor 2000 datieren, ist das Wort folglich kein einziges Mal enthalten, in der Wortschatz-Datenbank der Uni Leipzig gerade 18 mal, in COSMAS II 17 mal (Korpus: *Archiv der geschriebenen Sprache*, Suche am 11.02.2008). Im WWW können ferner große Ressourcen an relativ neuen Textgenres gefunden werden, die sich teilweise erst durch dessen Existenz entwickelt haben, z.B. die in Chatrooms verwendete Sprache (Renouf et al 2006: 50).

4.2.2.4 Korpusgröße

Die Größe der derzeit existierenden bzw. im Aufbau befindlichen linguistischen Korpora ist zweifelsfrei groß genug für gewisse Zwecke wie beispielsweise zum Training von PoS-Taggern, für andere, z.B. bestimmte statistische Zwecke sind sie jedoch trotzdem nicht ausreichend (Kilgarriff & Grefenstette 2003: 4). Selbst in sehr großen Korpora wie dem British National Corpus (>100 Millionen Wörter) existieren für seltene Wörter oder Kombinationen von Wörtern oft wenige oder gar keine Einträge (Kilgarriff & Grefenstette 2003: 4). Dieses Phänomen kann, zumindest annähernd, anhand des nach dem amerikanischen Linguisten George Kingsley Zipf benannten

Zipfschen Gesetzes erklärt werden. Ordnet man die Wörter nach ihrer Häufigkeit, indem man dem häufigsten Rang *eins* zuweist, dem zweithäufigsten Rang *zwei* etc., ergibt das Produkt aus Rang und Häufigkeit eine annähernd idente Konstante. Das Gesetz musste jedoch revidiert werden, da hinsichtlich sehr häufiger und sehr seltener Wörter Abweichungen bestehen und ist deshalb in erster Linie bei Wörtern mittlerer Häufigkeit zuverlässig. Die Wortverteilung in einem Text und folglich in einem aus einer endlichen Anzahl von Texten aufgebauten Korpus entspricht in etwa einer einfachen Zipfschen Verteilung (Abb. 4.2). Während das häufigste Wort einer Sprache n mal vorkommt, kann gemäß Zipf die Häufigkeit w für ein Wort mit dem Rang r folgendermaßen ermittelt werden: $w = 1/r * n$. Während also eine verhältnismäßig geringe Anzahl von Wörtern sehr oft vorkommt, ist ein sehr hoher Anteil von Wörtern in sehr geringen Frequenzen enthalten. Der Großteil der mehr als 100 Millionen im BNC enthaltenen Wörter kommt weniger als 50 mal vor, ein Wert, der zu gering ist, um statistisch stabile Rückschlüsse zuzulassen (Kilgarriff & Grefenstette 2003: 4). Ähnlich wie im Englischen verhält sich die Verteilung der Wörter im Deutschen. Die 30 häufigsten deutschen Wörter machen mehr als 31 Prozent der in einem Text enthaltenen Wörter aus, und 5,12 Prozent der häufigsten Wortformen bereits mehr als 90 Prozent der Textwörter (König 2001: 115). Im Negra Korpus 2.0, das aus aktuelleren Texten besteht als jene Korpora, auf die sich Königs Zahlen beziehen (Texte des 19. Jahrhunderts (König 2001: 115)), ist der Anteil der 30 häufigsten Wörter mit 30,1 Prozent zwar etwas geringer, trotzdem macht diese kleine, hauptsächlich aus Funktionswörtern wie Determinatoren und Präpositionen bestehende Menge, die nur knapp 0,07 Prozent der im Korpus enthaltenen Typen bzw. Types ausmacht[3], mehr als ein Viertel der Tokens aus.

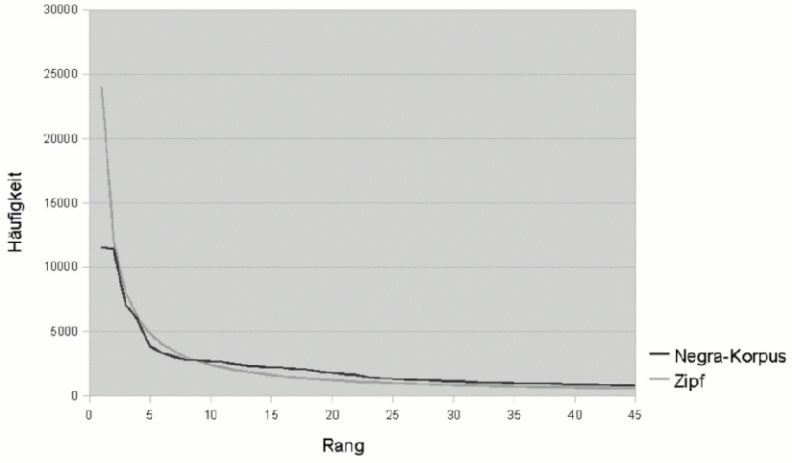

Abbildung 4.2: Verteilung der Häufigkeiten der 45 häufigsten Wörter im Negra-Korpus (tatsächliche und Zipf-Häufigkeit)

[3]Ein aus 290.754 Wörtern bestehendes, aus dem Negra-Korpus extrahiertes Testkorpus enthielt 42.306 verschiedene Wortarten bzw. Types

4.2.3 Nachteile des WWW gegenüber klassischen Korpora

4.2.3.1 Eingeschränkte Kontrolle über die Korrektheit der Inhalte

Webtext wird, im Gegensatz zu dem zur Erstellung von linguistischen Korpora benutzten Material, ohne große Rücksicht auf Korrektheit verfasst (Kilgarriff & Grefenstette 2003: 9). Daraus resultierend befinden sich im WWW unzählige Konstruktionen, welche hinsichtlich Grammatik, Syntax etc. falsch oder zumindest fragwürdig sind: Kilgarriff & Grefenstette (2003: 9) suchten über Google nach unkorrekten Schreibungen des Strings *„I believe“*. Während für die korrekte Schreibung 4 Millionen Seiten gefunden werden konnten, wurden für *„I beleive“* 70.900 und für *„I beleave“* immerhin noch 3910 Seiten gefunden[4]. Auch auf struktureller Ebene kann im Gegensatz zu den hinsichtlich Syntax kontrollierten Daten aus speziellen Korpora keine Garantie für die Richtigkeit der gelieferten Ergebnisse gegeben werden. Webtext enthält nämlich nicht nur laufenden Text, sondern auch Text aus Tabellen, Navigationselementen etc. Der von Suchmaschinen extrahierte Text wird ohne Rücksicht auf derartige störende Textelemente weiterverarbeitet, was zur Folge hat, dass die gelieferten Strings in vielen Fällen willkürliche Aneinanderreihungen von Nomen, Befehlswörtern oder dergleichen enthalten (vgl. 4.2.3.2). Dies hat zur Konsequenz, dass von Suchmaschinen übernommener Text, abhängig vom Verwendungszweck, mit großer Wahrscheinlichkeit nachbehandelt werden muss.

4.2.3.2 Eingeschränkte Suchfunktionen

Suchmaschinen wurden (bzw. werden) für Web-Benutzer entwickelt, deren Anforderungen an die entsprechende Maschine nicht mit jenen von Linguisten übereinstimmen. Diese „normalen“ Web-Benutzer haben klar definierte Informationsbedürfnisse wie die Lokalisierung einer bestimmten Seite, die Antwort auf eine bestimmte Frage oder das Auffinden einer Seite, deren Inhalt den Suchkriterien entspricht (Fletcher 2001: 5). Studien über das Verhalten dieser als typisch zu bezeichnenden Benutzer haben die Entwicklung von Suchmaschinen maßgeblich beeinflusst (Fletcher 2001: 6). Fletcher (2001: 6) nennt als Beispiel das eher geringe Interesse an der Verwendung von booleschen Funktionen (logische Operationen wie *AND, OR* oder *NEAR*), mit der Konsequenz, dass Suchmaschinen, die über eine größere Anzahl von booleschen Funktionen verfügen (sog. *„geek seek“*-Maschinen, z.B. AltaVista) und damit für linguistische Anwendungen interessant sind, verdrängt werden von Maschinen mit einem verhältnismäßig eingeschränkten Reservoir an booleschen Suchfunktionen, deren Stärken jedoch in für eben die Mehrzahl der Benutzer interessanten

[4]Ergebnisse für die gleiche Suche via Google 2008: *„I believe“*: 174 Millionen; *„I beleive“*: 4,2 Millionen; *„I beleave“*: 139.000; Datum der Suche: 03.02.2008.

Bereichen wie eine weitgehende Abdeckung des WWW liegen (z.B. Google) (Fletcher 2001: 6).

Die gängigen Suchmaschinen bieten zwar eine begrenzte Anzahl verschiedener Suchfunktionen an, diese sind jedoch nicht primär auf linguistische Zwecke ausgerichtet (Renouf et al. 2005: 1). Um die Eingabe zu optimieren bzw. den Suchbegriff zu verfeinern, stehen in Google die booleschen Operatoren *UND* (Konjunktion; +) und *ODER* (Disjunktion; |) sowie das Ausschlusskriterium (Minuszeichen; -) und die Anführungszeichen zur exakten Phrasensuche zur Verfügung. Ferner steht als Wildcard die Verwendung des Sternchens zur Verfügung, welches als Platzhalter für ein Wort eingesetzt werden kann. Das Sternchen kann als Platzhalter zwar nur für Wörter, nicht jedoch innerhalb von Wörtern verwendet werden (zumindest in Google). Folglich liefert der Suchstring *„*bruck“* Ergebnisse wie *Stadtgemeinde Bruck*, nicht jedoch *Innsbruck* oder *Vöcklabruck*. Auf die Berücksichtigung von Satzgrenzen sowie die Unterscheidung von Groß- und Kleinschreibung wird in Google ebenfalls keine Rücksicht genommen. Das letztgenannte Manko (zumindest handelt es sich um ein Manko aus linguistischer Sicht) kommt wiederum dem bereits erwähnten und von den Suchmaschinen-Anbietern protegierten normalen Benutzer zugute, indem unabhängig von der Verwendung von Groß- bzw. Kleinbuchstaben gleiche Resultate erzielt werden (Die Suche von sowohl *New York* als auch *new YoRk* und *New york* führen zu den gleichen Ergebnissen). Aufgrund dieser Defizite führen viele Suchabfragen, selbst unter Verwendung der Anführungszeichen zur exakten Phrasensuche, zu sehr willkürlichen Ergebnissen. So ergibt die Suche nach der Nomen-Verb-Struktur „house burns“ sehr wohl Treffer, die der gesuchten Struktur entsprechen (z.B. *...the city's famed Fenice opera house burns down during a restoration.*[5] oder *When a house burns up, it burns down.*), aber auch eine große Anzahl von Treffern, die den gesuchten String enthalten, deren Wortarten jedoch nicht jenen der im Suchstring entsprechenden Wörtern entsprechen. Folglich kann ein ursprünglich als Verb gesuchtes Element, z.B. *burns*, als Eigenname auftreten (wie in *...Bart vandalizes Burns's* ***house. Burns****'s watches Bart destroy his house...*).

Doch nicht nur die Suchfunktionen, auch der von den Suchmaschinen gelieferte Output ist nur eingeschränkt für linguistische Zwecke wie beispielsweise Konkordanz verwendbar. Die von den gängigen Suchmaschinen gelieferten Ergebnisse bestehen in der Regel aus drei Teilen: in der ersten Zeile des Ergebnisses wird der Titel der gefundenen Seite gezeigt. Sollte der Titel nicht bekannt sein, wird im Normalfall die URL der betreffenden Seite angezeigt. Weiters wird ein kurzer Auszug aus jenem Textteil auf der gefundenen Seite, der den Suchstring enthält, sowie die URL

[5] Beispiele sind den Suchergebnissen einer Suche des Strings „house burns“ mittels yahoo.de entnommen; Datum der Suche: 01.02.2008.

der entsprechenden Seite übermittelt. Der für linguistische Zwecke interessante Teil ist der kurze Auszug aus der entsprechenden Seite. Im Idealfall sind in diesem Auszug verwertbare Daten wie in (3a) enthalten (Suchbegriff: *Färöer* auf Google [20.02.2008]), die direkt oder nach geringfügiger Überarbeitung (hier: Entfernung des nachfolgenden Satzfragmentes) übernommen und weiterverarbeitet werden können. Die als Auszug gelieferten Textteile sind in der Regel jedoch sehr willkürlich und können Auszüge aus der entsprechenden Seite enthalten, welche nicht direkt mit dem gesuchten Ausdruck in Verbindung stehen, z.B. Menütitel wie in (3c) oder benachbarter Text wie in (3b).

(3) (a) Die Färöer sind eine kleine Inselgruppe, die man auf der Landkarte ungefähr in der Mitte zwischen Schottland und Island findet. Das nächste Land sind die ...

(b) Sie befinden sich in:. Nachrichten · Wetter · Europa; Färöer ... Färöer. Alle Städte:. A B C D E F G H I J K L M N O P Q R S T U V W X Y Z; T ...

(c) Startseite >. Färöer. -. - EM-Qualifikation · Übersicht · Kader · Trainer · Transfers · Begegnungen · News · Vereinsstat. ...

4.2.3.3 Eingeschränkte Möglichkeiten für Programmierer

Obwohl die Indizes der größten Suchmaschinen mehrere Milliarden Seiten enthalten und die gängigen Datenübertragungslösungen inzwischen den Transfer von sehr großen Datenmengen in relativ kurzer Zeit zulassen, ist die Verfügbarkeit dieser Daten für Programmierzwecke durch Maßnahmen der Suchmaschinenanbieter eingeschränkt. Die für die Programmierung in Java von Yahoo zur Verfügung gestellte Programmierschnittstelle ermöglicht beispielsweise pro Suchabfrage die Übermittlung von höchstens 100 Suchresultaten[6]. Für repräsentative Abfragen ist diese Anzahl wahrscheinlich in den meisten Fällen zu gering. Immerhin kann das zurückgelieferte Suchergebnis auf einen bestimmten Ausschnitt des Gesamtergebnisses beschränkt werden. Dadurch ist der Zugriff nicht nur auf die 100 Resultate mit dem höchsten Ranking im Suchergebnis beschränkt, und im Grunde kann auf sämtliche aus jeweils hundert Resultaten bestehenden Ausschnitte zugegriffen werden. Folglich können durch die Wiederholung der gleichen Suchabfrage unter gleichzeitiger ständiger Inkrementierung des Ergebnisraumes größere Mengen an Resultaten ermittelt werden, die dadurch entstehende hohe Anzahl von Suchabfragen kann jedoch sehr lange Bearbeitungszeiten zur Folge haben. Abhängig von der Inter-

[6] In der Dokumentation der entsprechenden Java-Methode ist die Anzahl laut Kommentar auf maximal 50 Resultate beschränkt (Stand: 04.04.2008). Tatsächlich können jedoch bis zu 100 Resultate übermittelt werden.

netverbindung kann die Sammlung von größeren Mengen an Resultaten für einen einzigen Suchstring somit mehrere Minuten dauern.

4.2.4 Linguistische Optimierung von Suchergebnissen

Aufgrund der in 4.2.3 erwähnten Einschränkungen der gängigen Suchmaschinen ist die linguistische Verwendung der Ressource WWW in vielerlei Hinsicht limitiert. Ein Ausweg aus dieser Situation besteht darin, die von Suchmaschinen gelieferten Daten aufzubereiten, um sie in weiterer Folge ähnlich einem linguistischen Korpus zu verwenden. Kilgarriff (2006: 148) nennt diesen Prozess *Data cleaning*, bestehend aus der Suchabfrage, dem Download der Daten, anschließender 'Reinigung' und Beseitigung von Duplikaten, linguistischer Annotation und Bearbeitung mit einem geeigneten Werkzeug zur Suche in Korpora (Kilgarriff 2006: 148). Während die Aufgabe des Suchens sowie die Bereinigung von Duplikaten von der Suchmaschine abgenommen wird, stehen zur linguistischen Bearbeitung Werkzeuge wie PoS-Tagger, Lemmatisierer etc. zur Verfügung. Das eigentliche *Cleaning* geschieht zwischen den beiden Blöcken und beinhaltet das Entfernen von im Webtext enthaltenem, unerwünschtem Text wie Navigationsleisten, Menüpunkte, Metatext und dergleichen, dem Auffinden von struktureller Information wie Absätzen und die Umwandlung des Textes in eine für die weitere Verwendung geeignete Form. Die Bereinigungsprozedur stellt einen entscheidenden Schritt in der Vorbereitung des aus dem WWW gewonnenen Materials dar: je besser die Rohdaten bereinigt werden, desto besser ist das Ergebnis (Kilgarriff 2006: 149).

4.2.4.1 Programme zur Optimierung: KWiCFinder und WebCorp

KWiCFinder ist ein bereits 1997 von William Fletcher für Konkordanz- und Recherchezwecke entwickeltes Programm, das ursprünglich vor allem zur Anwendung im Sprachunterricht gedacht war (Fletcher 2001: 2). Im Unterschied zu den gängigen Suchmaschinen ist KWiCFinder nicht webbasiert, vielmehr muss das Programm auf der Festplatte installiert werden. Da die Installation des Programms auf das Windows-Betriebssystem beschränkt ist, ist keine Plattformunabhängigkeit gegeben und Benutzer von alternativen Betriebssystemen, z.B. UNIX, sind folglich von der Verwendung ausgeschlossen. Es existiert zwar eine plattformunabhängige, webbasierte *„light weight“*-Version mit dem Namen WebKWiC, die Funktion dieser Applikation ist jedoch nach Angaben des Autors zur Zeit nicht zufriedenstellend (Fletcher 2001: 15). KWiCFinder bietet ein reiches Inventar an jenen Funktionen und Wildcards, die in kommerziellen Suchmaschinen fehlen. Darüber hinaus stehen dem Benutzer in KWiCFinder sogenannte *Tamecards*, eine Art Formalismus,

der es erlaubt, mittels knapper Notation, ähnlich regulären Ausdrücken, eine Reihe alternierender Formen abzudecken, zur Verfügung. Die Eingabe von *s/iau/ng* wird in KWiCFinder zum Beispiel in die Formen *sing, sang, sung* expandiert (Fletcher 2001: 8).

Das WebCorp-Projekt wurde ab 1998 an der Birmingham City University mit dem Ziel, über Suchmaschinen gefundenen Webinhalt zur linguistischen Verwendung aufzuarbeiten, entwickelt:

> „The WebCorp project was an experiment to see whether we could develop a system to extract linguistic data from web text efficiently and present a quality of raw and analysed linguistic output that was similar to that derived from finite corpora and which met users' expressed needs." (Renouf et al 2006: 48)

Im Gegensatz zu KWiCFinder ist WebCorp webbasiert, deshalb kann das Service plattform-unabhängig und ohne Installation genutzt werden. Abbildung 4.3 zeigt die Architektur der Such- und Analyseroutine von WebCorp. Die Eingabe der Su-

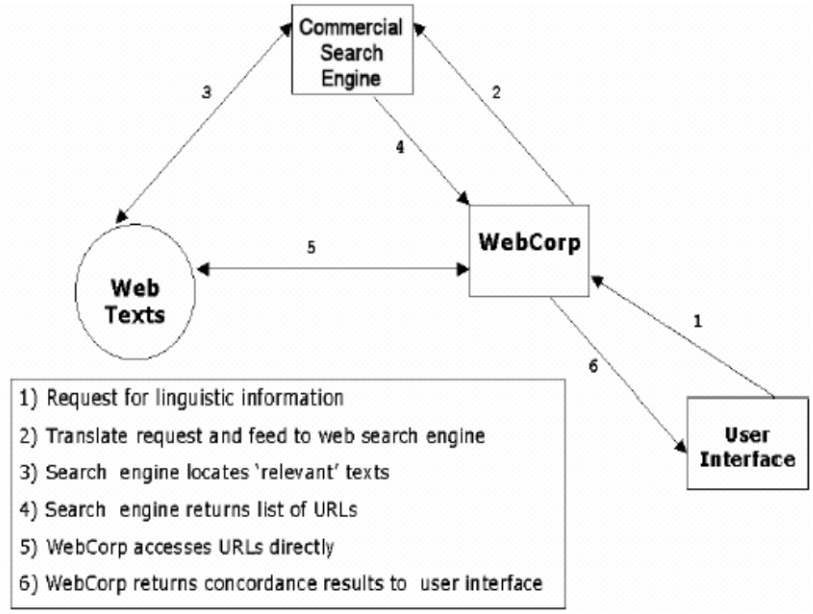

Abbildung 4.3: Architektur von *WebCorp*

che erfolgt ähnlich den Suchmaschinen wie Google über ein graphisches Interface. Die Eingaben werden von WebCorp aufbereitet und anschließend an die Suchmaschinen gesendet. Die von den Suchmaschinen als Ergebnis retournierten URLs werden von WebCorp direkt aufgerufen, der Text extrahiert und in aufbereiteter Form an das User Interface zurückgeschickt. WebCorp bietet ähnlich wie KWiCFinder eine Reihe von Suchfunktionen, die in den kommerziellen Suchmaschinen

nicht (mehr) berücksichtigt werden. Dazu gehören die Unterscheidung von Groß- und Kleinschreibung und die Verwendung von Platzhaltern auch *innerhalb* von Wörtern. Weiters kann die Suche durch die Wahl sowohl zwischen fünf verschiedenen Suchmaschinen (u.a. Google, AltaVista) als auch vier verschiedenen Gruppen von Zeitungsartikeln (*UK broadsheet, UK tabloid, French News, US News*) verfeinert werden (Renouf et al. 2006: 50). WebCorp hält weitere Online-Werkzeuge für linguistische Analysen bereit, z.B. ein *Wordlist Generator* genanntes Tool, welches die Wörter einer durch die URL lokalisierten Homepage in alphabetischer Liste oder nach Frequenz geordnet ausgibt.

Sowohl KWiCFinder als auch WebCorp bieten ein breites Spektrum an Suchfunktionen und liefern Ergebnisse, die hinsichtlich linguistischer Weiterverwendung gegenüber den von gängigen Suchmaschinen gelieferten Ergebnissen klare Vorteile besitzen (4.4). Beide Anwendungen haben jedoch den großen Nachteil der sehr geringen Geschwindigkeit. Während durch die Suche über konventionelle Suchmaschinen innerhalb von Sekundenbruchteilen auf eine sehr große Menge an Material zugegriffen werden kann, liefern sowohl KWiCFinder als auch WebCorp im Vergleich dazu Ergebnisse erst nach beträchtlicher Verarbeitungszeit. Abhängig von den Suchkriterien sowie weiteren Faktoren wie Hardware, Internetverbindung etc. dauern Suchvorgänge mitunter deutlich länger als eine Minute. Aus diesem Grund wurde weder die Benutzung von KWiCFinder noch jene von WebCorp zur Unterstützung der Semantikkontrolle in Betracht gezogen

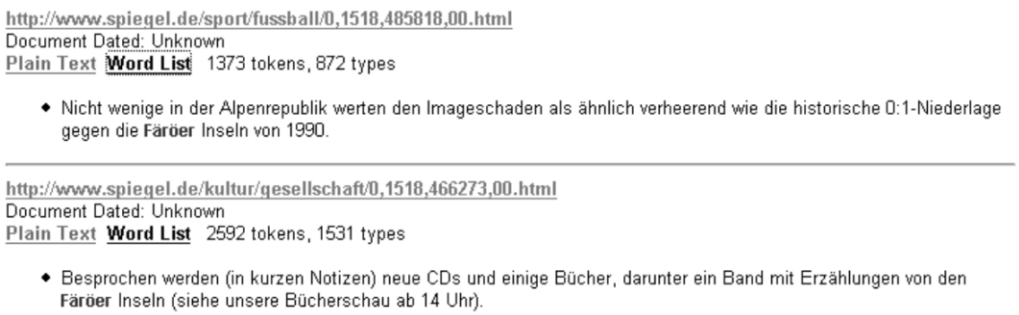

Abbildung 4.4: Ausschnitt aus einem WebCorp-Suchergebnis

4.3 Umsetzung der statistischen Semantikkontrolle

Die Überlegung, die der semantischen Kontrolle zugrunde liegt, ist im Grunde simpel. Die von SCall generierten Sätze werden in semantisch zusammengehörende Strings zerlegt und diese Strings auf deren Vorkommen im zugrundeliegenden Korpus, in diesem Fall dem WWW, untersucht. Die Suche in SCall wurde aus Gründen der Einfachheit auf Bigramme, das heißt Kombinationen aus zwei Wörtern,

beschränkt. Als relevante Daten kommen jene Bigramme in Frage, deren Elemente in semantischer Beziehung zueinander stehen und die aus semantischer Sicht folglich potentielle Fehlerquellen darstellen. Zu diesem Zweck werden für sämtliche von SCall generierten Sätze während des Generierungsprozesses Bigramme gebildet, die als Ausgangsdaten für die Semantikkontrolle dienen. Diese Bigramme werden im Korpus gesucht; somit wird festgestellt, ob die entsprechenden Wortkombinationen in jenen dem Korpus zugrundeliegenden Texten bereits verwendet wurden. Wird die Wortsequenz im Korpus mit einer ausreichenden Frequenz gefunden, wird davon ausgegangen, dass die Sequenz semantisch korrekt ist. In einem Korpus wie dem *Negra-Korpus*, das kontrolliert aufgebaut ist, reichen bereits wenige Treffer aus, um Konstruktionen als semantisch korrekt zu klassifizieren, da angenommen werden kann, dass die zugrundeliegenden Texte, im Fall des Negra-Korpus Artikel aus der Tageszeitung *Frankfurter Rundschau*, keine bzw. nur sehr wenige semantisch inkorrekte Konstruktionen enthalten. Anders verhält es sich mit Textmaterial aus dem WWW. Da die in 4.2.4 erwähnten Hilfswerkzeuge zur Optimierung von Suchergebnissen aus dem WWW in SCall nicht eingesetzt werden, wird die Semantikkontrolle mit Daten, die ohne Filterung aus Suchabfragen mittels einer geeigneten Suchmaschine gewonnen werden, durchgeführt. Da die Inhalte des WWW hinsichtlich der Semantik nicht kontrolliert sind, sind semantisch inkorrekte Elemente im Suchergebnis zu erwarten. Dementsprechend muss der Wert an Funden, der für die positive Bewertung der Semantik ausschlaggebend ist (=Schwellenwert), unter Berücksichtigung der zu erwartenden Fehlerhaftigkeit des Suchergebnisses gewählt werden. Die Wahl eines konstanten Schwellenwertes erweist sich jedoch als nicht zielführend, wie die semantische Bewertung der Bigramme in Tabelle 4.1 zeigt. Während die Bigramme *saure Milch* und *ranzige Butter* sowie das Bigramm *ranzige Milch* aufgrund der relativ hohen bzw. relativ niedrigen Trefferanzahl als semantisch korrekt bzw. inkorrekt bewertet werden können, ist die Bewertung des semantisch fragwürdigen Bigramms *saure Butter* schwierig, da immerhin mehr als 500 Entsprechungen gefunden werden konnten[7].

4.3.1 Die Wahl einer geeigneten Suchmaschine

Die Anbieter der meisten Suchmaschinen stellen Schnittstellen zur Verfügung, die es ermöglichen, Suchabfragen an die entsprechenden Suchmaschinen während der Laufzeit eines Programms zu senden und die Ergebnisse der Suche als Variablen bzw. Objekte (je nach gewählter Programmiersprache) an das Programm zurückgeben. Für die Suche in SCall wurde die Suchmaschine *yahoo* gewählt. Aufgrund

[7] Im Fall des Bigramms *saure Butter* wird das Suchergebnis zusätzlich durch das suchmaschinenseitige Ignorieren von Wortgrenzen verfälscht, da (semantisch wiederum korrekte) Bigramme wie *saure Buttermilch* im Ergebnis berücksichtigt werden.

Bigramm	Yahoo-Treffer	Beurteilung
saure Milch	57500	✓
ranzige Milch	182	✗
saure Butter	524	✓/✗
ranzige Butter	2620	✓

Tabelle 4.1: Bewertung der semantischen Korrektheit von ausgewählten Bigrammen anhand des Suchergebnisses

des bei weitem größeren Index und der daraus resultierenden Korpusgröße hätte sich natürlich die Suchmaschine von *google* angeboten, da jedoch zum Zeitpunkt der Programmierung von SimpleCall die entsprechende Schnittstelle auf der Entwicklerseite von *google* nicht angeboten wurde[8], musste auf die Dienste von *yahoo* zurückgegriffen werden.

4.3.2 Die Ermittlung eines Schwellenwertes

Der Umstand, dass aus dem WWW gewonnene sprachliche Daten aufgrund der unter 4.2.3 aufgezählten Defizite einen relativ hohen Anteil an Interferenzen enthalten, muss während der Auswertung der Suchergebnisse und in weiterer Folge im Zuge der Festlegung eines Schwellenwertes für die Mindestanzahl an Treffern für ein Bigramm berücksichtigt werden. Unter der Annahme, dass der relative Anteil an inkorrekten Suchergebnissen bzw. Verschmutzung konstant ist, steigt der absolute Anteil an Verschmutzung mit der Anzahl der Treffer. Die Wahl eines eher niedrigen Schwellenwerts würde aller Voraussicht nach dazu führen, dass semantisch inkorrekte Bigramme, die aus Elementen mit relativ hoher Einzelhäufigkeit bestehen, mit großer Wahrscheinlichkeit als korrekt markiert würden. Andererseits würde ein hoher Schwellenwert dazu führen, dass semantisch korrekte Konstruktionen, deren Elemente jedoch nur mit geringer Einzelhäufigkeit im Korpus vorkommen, aufgrund der Trefferanzahl fälschlicherweise als inkorrekt markiert würden. Aus diesem Grund wurden die relativen Häufigkeiten der Bestandteile der Bigramme zur Festlegung eines Wertes für die Mindestanzahl an Treffern herangezogen. Mit Hilfe von statistischen Methoden sollte aus den relativen Einzelhäufigkeiten ein Wert, der auf die zu erwartende Häufigkeit des Bigramms schließen lässt, ermittelt werden, und von diesem Wert ausgehend ein Schwellenwert festgelegt werden. Um jedoch die relativen Häufigkeiten von im Korpus enthaltenen Wörtern ermitteln zu können, musste die (unbekannte) Gesamtgröße der von der Suchmaschine abgedeckten Textmenge, die für die Feststellung der relativen Häufigkeit notwendig ist, mit Hilfe eines Korpus, dessen Größe bekannt ist, festgestellt werden (vgl. 4.2.1).

[8] Februar 2008

Wort	**Rel. Häufigkeit im Negra-Korpus**	**Yahoo-Treffer**	**Gesamtgröße**
nicht	0,0066	1730 Mio.	263,91 Mrd.
auch	0,0050	1550 Mio.	307,77 Mrd.
nach	0,0035	1680 Mio.	486,51 Mrd.
aber	0,0019	843 Mio.	447,56 Mrd.
durch	0,0017	993 Mio.	581,38 Mrd.
wenn	0,0012	859 Mio.	720,90 Mrd.
noch	0,0032	1150 Mio.	363,21 Mrd.
werden	0,0040	1450 Mio.	360,67 Mrd.
wird	0,0031	1230 Mio.	393,08 Mrd.
sein	0,0015	743 Mio.	496,87 Mrd.
einen	0,0025	1330 Mio.	534,62 Mrd.
sind	0,0031	1730 Mio.	564,44 Mrd.
oder	0,0020	1690 Mio.	848,58 Mrd.
haben	0,0020	1080 Mio.	528,84 Mrd.
einer	0,0032	1070 Mio.	334,73 Mrd.
Gemittelter Wert für Gesamtgröße:			482,20 Mrd.

Tabelle 4.2: Relative Häufigkeit von ausgewählten deutschen Wörtern und daraus ermittelte Gesamtgröße der im Index von *yahoo* enthaltenen deutschen Wörter

Als Referenzkorpus wurde ein Teilkorpus des Negra-Korpus mit einer Größe von knapp 310.000 Tokens herangezogen. In diesem Teilkorpus wurde die relative Häufigkeit von 15 ausgewählten Wörtern festgestellt, wobei die Auswahl dieser Wörter aus einer Liste der 200 häufigsten deutschen Wörter erfolgte (Liste aus König (2001: 114)). Um die spätere Verfälschung der Ergebnisse durch die Vermischung der Suchergebnisse mit fremdsprachlichem Material zu verhindern wurden Wörter, die neben Deutsch in weiteren Sprachen vorkommen (wie *in* (ital.); *den* (dänisch, schwedisch)), nicht berücksichtigt und nur Wörter, die (höchstwahrscheinlich) in erster Linie in deutschen Texten vorkommen, herangezogen (vgl. Ansatz von Grefenstette & Nioche in 4.2.1). Durch die Suche mittels Suchmaschine wurde die absolute Häufigkeit der entsprechenden Wörter innerhalb des Index von *yahoo* ermittelt, da die relative Häufigkeit der Wörter aus dem Negra-Korpus bekannt war, konnte die Gesamtmenge der Wörter annäherungsweise berechnet und ein Mittelwert interpoliert werden (Tabelle 4.2). Für die Menge der im Index der Suchmaschine Yahoo.de[9] enthaltenen deutschen Wörter wurde folglich eine Größe von etwa 480 Milliarden Wörtern ermittelt.

Durch die der Gesamtkorpusgröße kann die relative Häufigkeit sämtlicher im Korpus enthaltenen Wörter ausgerechnet werden. Ausgehend von den Einzelwahr-

[9] yahoo.de; Suchoption *„nur deutsche Begriffe“*; Suche am 21.06.2008

scheinlichkeiten kann in weiterer Folge die zu erwartende absolute Häufigkeit von Bigrammen, die als Schwellenwert vorgesehen ist, ermittelt werden. Hier liegt natürlich die Annahme nahe, die relative Häufigkeit von Bigrammen, analog zur Ermittlung z:B. der Wahrscheinlichkeiten von Zahlenfolgen in Lottoziehungen, durch die Multiplikation der relativen Häufigkeiten der Bestandteile zu ermitteln. Dabei muss jedoch ein wichtiger Umstand berücksichtigt werden: in einem Korpus, der aus Sätzen einer natürlichen Sprache besteht, sind die Bestandteile (bzw. die Wörter) nicht nach zufälligen (wie im Lotto), sondern nach systematischen, der entsprechenden Sprache zugrunde liegenden Regeln, kombiniert. Folglich gibt es Kombinationen, die aufgrund dieser Regeln nicht möglich sind, in der Berechnung der Wahrscheinlichkeit für Auftretenshäufigkeiten jedoch berücksichtigt werden. Infolgedessen würden deutsche Bigramme, die aus Wortartenkombinationen bestehen, die aufgrund der Regeln der deutschen Syntax nicht möglich sind, wie *(Det, Det)* (z.B. *(dieser das), (die die)*, auf gleiche Weise behandelt wie korrekte Bigramme wie *(Adj N)* (*(schönes Haus), (gelbe Fahrräder)*. Aus diesem Grund muss die syntaktische Struktur des Korpus während der Ermittlung eines Schwellenwertes berücksichtigt werden. Deshalb muss ermittelt werden, welche Wortkombinationen mit welcher Häufigkeit auftreten. So ist z.B. anzunehmen, dass Bigramme, die aus einem Artikel und einem Nomen bestehen ([Det N]), häufiger auftreten als etwa Bigramme wie [Det Det]. Um statistische Werte für die Distribution von Bigrammen zu ermitteln, muss wiederum auf Daten aus dem Negra-Referenzkorpus zurückgegriffen werden. Durch die enthaltene PoS-Information kann festgestellt werden, mit welcher relativen Häufigkeit Bigramme im Korpus enthalten sind.

Das aus Adjektiv und Nomen ([ADJA NN], wie in *großes Haus, kleines Kind*) bestehende Bigramm konnte im Referenzkorpus 16.570 mal gefunden werden. Bei der Gesamtanzahl von 312.447 Bigrammen (Wortanzahl minus 1) errechnet sich die relative Häufigkeit dieses Bigramms auf 5,3 Prozent (vgl. Tabelle 5.4) . Das heißt, das 5,3 Prozent aller Bigramme im Negra-Korpus dem Bigramm [ADJA NN] entsprechen. Da ein Bigramm aus zwei Wörtern besteht und Überschneidungen (in diesem Fall) ausgeschlossen sind, sind $16.570 \cdot 2 = 33140$ (10,6 Prozent) aller im Korpus enthaltenen Wörter in [ADJA NN] Bigrammen enthalten.

In weiterer Folge muss festgestellt werden, mit welcher Wahrscheinlichkeit die einzelnen Elemente der Bigramme in den entsprechenden Bigrammen vorkommen. Dies kann wiederum unter Verwendung des Negra-Korpus ermittelt werden. Für die im [ADJA NN]-Bigramm enthaltenen Nomen gilt demnach folgendes: von insgesamt 83190 im Korpus enthaltenen Nomen sind 16570 bzw. 19,91 Prozent aller Nomen in [ADJA NN]-Bigrammen enthalten. Für die Adjektive liegt die Wahr-

Bigramm	**Erläuterung**	**Beispiele (aus dem Negra-Korpus)**	**Anzahl Funde**	**Rel. Häufigkeit**
ADJA NN	Adjektiv - Nomen	gewagte Verbindungen, eigene Handschrift, berühmte Kollegen	16570	0,0530
ART NN	Artikel - Nomen	die Einflüsse, die Idee, eine Möglichkeit	26689	0,0854
NN VVFIN	Nomen - finites Verb	...auf den *Bändern befinden* sich..., ...die folkloristischen *Elemente beschränken* sich...	8389	0,0268
APPR ART	Präposition ohne Artikel - Artikel	aus dem, mit einem, von der	9875	0,0316
APPRART NN	Präposition mit Artikel - Nomen	vom Thema, zur Aufgabe, vom Vater, ans Kabel	5337	0,0171

Tabelle 4.3: Relative Häufigkeit ausgesuchter Bigramme im Negra-Korpus

scheinlichkeit bei 87,43 Prozent (18952 Funde). Durch die Ermittlung dieser Daten kann der Schwellenwert S wie folgt berechnet werden:

$$S = \frac{W_1 \cdot R_1}{G_K} \cdot \frac{W_2 \cdot R_2}{G_K} \cdot G_K \cdot R_B$$

W_1...Fundanzahl Wort1
W_2...Fundanzahl Wort2
G_K...Korpusgröße
R_B...relative Häufigkeit des Bigramms
R_1...relative Häufigkeit von Wort1 an entspr. Position im entspr. Bigramm
R_2...relative Häufigkeit von Wort2 an entspr. Position im entspr. Bigramm

Der unter Verwendung dieser Formel errechnete Wert gibt an, mit welcher statistischen (absoluten) Häufigkeit das die Wörter enthaltende Bigramm im Korpus auftritt. Dieser Wert wird folglich als Schwellenwert für die semantische Korrektheit angenommen. Übersteigt die Anzahl der Treffer für die Suche nach dem exakten Bigramm diesen Schwellenwert, wird das Bigramm als semantisch korrekt angesehen. In Tabelle 4.4 wird die Bewertung ausgesuchter [ADJA NN]-Bigramme anhand dieser Methode gezeigt.

ADJA-NN-Bigramm	**Yahoo-Treffer für Bigramm**	**Yahoo-Treffer für Adjektiv**	**Yahoo-Treffer für Nomen**	**Schwellenwert**	**Akz.**
runder Tisch	1230000	6690000	111000000	2570	✓
dummer Tisch	24	3220000	111000000	1237	✗
ranzige Butter	2790	411000	191000000	272	✓
saure Butter	295	2420000	191000000	1600	✗
braune Katze	1520	6090000	41300000	870	✓
grüne Katze	3650	59500000	41300000	8506	✗

Tabelle 4.4: Semantische Akzeptanz ausgesuchter Bigramme (*Akz. = Akzeptanz*)

Die Bigramme *dummer Tisch*, *saure Butter* und *grüne Katze* werden aufgrund des zu geringen Schwellenwertes folglich als semantisch nicht korrekt markiert.

Kapitel 5

SCall: Das Programm

5.1 Einleitung

Während in Teil I dieser Arbeit die theoretischen Grundlagen für die Umsetzung von SCall diskutiert wurden, werden in diesem Teil der Aufbau sowie die Funktionsweise des Programms erläutert.

Wie bereits in der Einleitung dieser Arbeit erwähnt, ist SCall in zwei Module unterteilt. Das erste Modul ist für den Eingabe- und Analyseprozess zuständig. Im Zuge dieses Prozesses werden Daten eingegeben, analysiert und für die Weiterverarbeitung aufbereitet. Dieser Prozess wird in 5.3 erläutert. Das zweite Modul ist für den Generierungsprozess verantwortlich. Im Zuge dieses Prozesses werden, ausgehend von jenen während der Eingabe und Analyse ermittelten lexikalischen und syntaktischen Daten, Übungsbeispiele für die Verwendung im Sprachunterricht generiert. Auf die Funktionsweise der Generierungsprozedur wird detailliert in 5.5 eingegangen.

SCall wurde zum Großteil in Java programmiert, zur Modifikation von Hilfsprogrammen wurden partiell Python (Spark-Parser, sh. 5.3.3), Perl (*TreeTagger*, sh. 5.3.1) und Common Lisp (*Morphix*, sh. 5.3.2) eingesetzt. Das Programm kann nur auf unixbasierten Betriebssystemen ausgeführt werden, da einige Komponenten (z.B. der Chunkparser *SPARK* sowie der in *TreeTagger* integrierte ChunkParser) auf die Verwendung unter Unix beschränkt sind. Die Implementierung von SCall geschah zu reinen Versuchszwecken, eine weitere Verwendung des Programms ist nicht angedacht. Aus diesem Grund wurde auf das für CALL-Anwendungen nicht unwichtige Interface-Design (vgl. Levy 1997: 49ff) keine bzw. nur sehr marginal Rücksicht genommen. Vielmehr wurde das Design auf ein für Versuchszwecke unbedingt notwendiges Maß reduziert.

5.2 Sprachspezifische Einschränkungen in SCall

Natürliche Sprache ist komplex. Diesem Umstand musste während der Entwicklung von SCall Rechnung getragen und der Funktionsrahmen entsprechend eingeschränkt werden. Die wichtigsten sprachlichen Einschränkungen sind in Tabelle 5.1 angeführt.

Einschränkung	**Erläuterung**
Keine Verarbeitung von Nebensätzen möglich	SCall kann nur ein finites Verb pro Eingabesatz analysieren. Aus diesem Grund können Nebensätze nicht verarbeitet werden.
Eingeschränkte Verwendung von Präpositionen	Präpositionalphrasen, die wie Präpositionen verwendet werden (*in Abhängigkeit von, im Einklang mit*), werden nicht als solche erkannt und können daher nicht verarbeitet werden.
Keine Berücksichtigung von Passivkonstruktionen	Die Analyse von Passivkonstruktionen ist nicht möglich.
Teilweise: Probleme mit Genitivkonstruktionen	Die Analyse von Genitivkonstruktionen wie *Peters Haus* bereiten dem Programm Probleme.

Tabelle 5.1: Sprachspezifische Einschränkungen in SCall (Auswahl)

5.3 Eingabe- und Analyseprozess

Während des Eingabe- und Analyseprozesses werden jene Daten generiert, die in weiterer Folge als Ausgangsbasis für den Generierungsvorgang dienen. Im Zuge dieses Prozesses soll dem Benutzer jene Analysearbeit abgenommen werden, welche rechnerseitig unter einem vertretbaren Zeitaufwand bewerkstelligt werden kann. Unter diese Aufgaben fallen das Feststellen der Wortklassen (*Part-of-speech-tagging*), Lemmatisierung (Auffinden der zugrundeliegenden Form der verwendeten Wörter), Parsing (Auffinden von syntaktischen Strukturen), morphologische Analyse sowie die Erstellung eines zweisprachigen Wörterbuches für die in den Beispielen verwendeten Wörter. Das Ziel der Prozedur ist eine möglichst präzise, benutzergerechte Aufbereitung der Daten, die eine einfache Manipulation zulässt. In weiterer Folge können die Daten im Verlauf dieses Prozesses vom Benutzer manipuliert werden, ferner werden die Selektionsrestriktionen ermittelt. Abbildung 5.1 zeigt ein Ablaufdiagramm der Eingabe- und Analyseprozedur, die einzelnen Analyseschritte werden im folgenden im Detail erklärt.

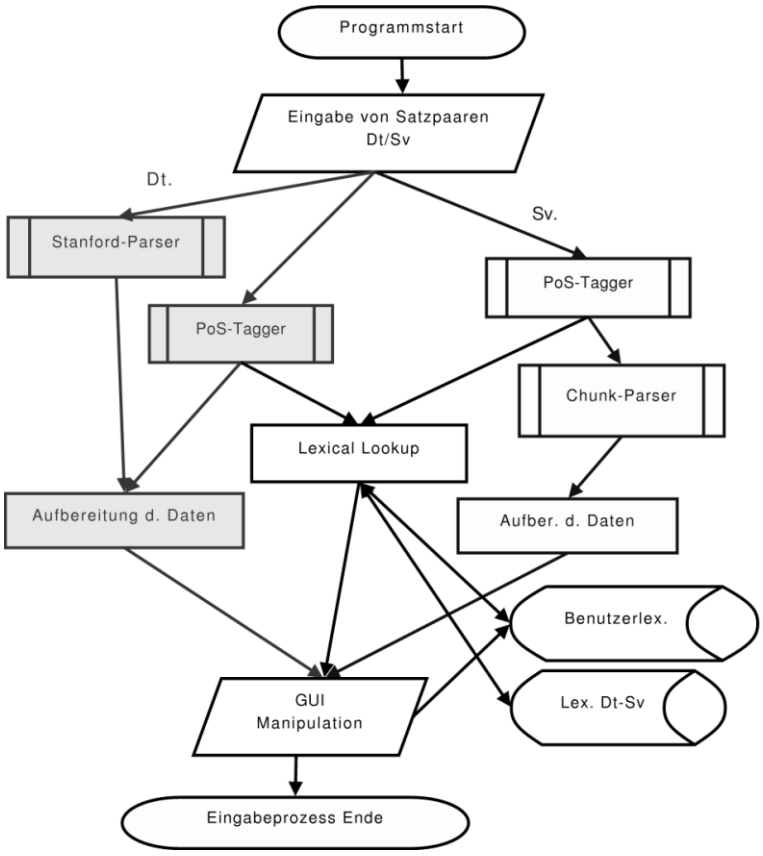

Abbildung 5.1: Ablaufdiagramm für das Eingabemodul von SCall

5.3.1 PoS-Tagging

Unter *Part-of-Speech Tagging* (kurz *PoS-Tagging*) versteht man die automatische Annotierung von Inputtext mit Part-of-Speech-Information (Voutilainen 2003: 220). Ein PoS-Tagger weist jedem Wort ein sogenanntes *Tag*[1] zu, das die PoS bzw. Wortartenkategorie des entsprechenden Wortes repräsentiert. In der Regel werden die Wortarten nach syntaktischen Kriterien (Verben, Substantivwörter, Adjektive etc.) unterteilt. Die meisten PoS-Tagger ermitteln neben PoS-Tags auch lexikosemantische sowie morphologische Information der analysierten Wörter (Voutilainen 2003: 220). Für das PoS-Tagging sowohl des deutschen als auch des schwedischen Inputs wird in SCall der am Institut für maschinelle Sprachverarbeitung an der Universität Stuttgart entwickelte probabilistische *TreeTagger* (Schmid 1994, Schmid 1995) verwendet. Der TreeTagger ist sprachunabhängig und kann im Grunde für die Analyse jeder beliebigen Sprache verwendet werden. Voraussetzung für die Adaption sind jedoch das Vorhandensein eines manuell annotierten (=mit PoS-Tags versehenen) Korpus in der entsprechenden Sprache sowie ein dazu korrespon-

[1]Scheinbar hat sich im Zusammenhang mit Pos-Tagging die Verwendung des Wortes *Tag* auch im deutschsprachigen Raum durchgesetzt. Tatsächlich konnte keine entsprechende, sinnhafte Übersetzung gefunden werden.

dierendes Lexikon. Ausgehend von diesen Ressourcen als Input generiert das im TreeTagger-Paket enthaltene Trainingsprogramm eine Parameterdatei, die in weiterer Folge zum PoS-Taggen von Texten in der entsprechenden Sprache verwendet wird. Für Deutsch steht eine Parameterdatei, die anhand von ca. 2 Millionen Wörtern aus dem Penn-Treebank-Korpus trainiert wurde (Schmid 1994: 6), zum kostenlosen Download zur Verfügung. In Tests an Auszügen aus dem Korpus der Stuttgarter Zeitung wurden Präzisionswerte im Bereich von über 96 Prozent erreicht (Schmid 1995: 7). In der deutschen Version des TreeTaggers werden zur Annotation die Tags des Stuttgart-Tübingen-Tagset (STTS) verwendet. Das STTS resultierte aus der Zusammenarbeit zwischen den Universitäten in Stuttgart und Tübingen, die es zum Ziel hatte, die damals bestehenden Tagsets zu vereinheitlichen, um bereits durchgeführte Korpusarbeit gegenseitig zu nutzen (Schiller et al. 1995: 3). Zwischenzeitlich hat sich das STTS zu einem Quasi-Standard zur Annotation von deutschem Material entwickelt. Das STTS besteht aus 54 verschiedenen Tags, wobei 48 Tags reine Wortformen repräsentieren und 6 zusätzliche Tags für weitere Annotationen wie Nichtwörter und Satzzeichen zur Verfügung stehen (Schiller et al. 1995: 6).

Neben der Annotation mit PoS-Tags ermittelt der TreeTagger auch die Lemmata bzw. Grundform der analysierten Wörter. Welche Form als Lemma angenommen wird, kann je nach Verwendungszweck und Tagger unterschiedlich sein, im allgemeinen entspricht das Lemma der je nach Sprache unterschiedlichen konventionalisierten Grundform (Glück 2000: 403). Diese Grundform generiert der TreeTagger in der Regel auch als Lemma: für Verben wird der Infinitiv Präsens ausgegeben, für Nomen den Nominativ Singular etc. Abweichend davon wird für für einige Wortarten, z.B. für definite Artikel (*der, die, das, dem* etc.), als Lemma eine nichtlexikalisierte Form (für definite Artikel: *d*) angegeben. Sollte für ein Inputwort kein Lemma gefunden werden, wird die Leerstelle durch das Tag <*unknown*> markiert. Dies trifft in erster Linie auf falsch geschriebene Wörter und in vielen Fällen auf Eigennamen zu. Das Problem der falsch geschriebenen Wörter erübrigt sich in SCall, da davon ausgegangen wird, dass der Input des Benutzers keine Rechtschreibfehler enthält. Aus diesem Grund wird in SCall generell angenommen, dass es sich im Falle eines mit einem <*unknown*>-Tag markierten Wortes um einen Eigennamen handelt. Deshalb werden in allen durch das <*unknown*>-Tag markierten Einträgen die Lemmata durch das entsprechende Eingabewort ersetzt. Da davon ausgegangen wird, dass es sich um Eigennamen handelt, können diese, mit Ausnahme von Genitivkonstruktionen, auch als Lemma übernommen werden. Beispiel (4) zeigt den mit STTS-Tags versehenen Output des deutschen TreeTaggers für den Satz *Peter Westenthaler wird sich wegen Falschaussage vor Gericht verantworten müssen.*

(4)	Peter	NE	Peter
	Westenthaler	NE	<unknown>
	wird	VAFIN	werden
	sich	PRF	er|es|sie|Sie
	wegen	APPR	wegen
	Falschaussage	NN	Falschaussage
	vor	APPR	vor
	Gericht	NN	Gericht
	verantworten	VVINF	verantworten
	müssen	VMINF	müssen
	.	$.	.

Für das PoS-Tagging der schwedischen Daten stand zum Zeitpunkt der Programmierung von SCall keine Parameterdatei für den TreeTagger zur Verfügung, weshalb eine entsprechende Datei unter Verwendung des Trainingsprogramms von TreeTagger generiert werden musste. Als Trainingskorpus diente ein ca. 1,1 Millionen Wörter umfassender Teilkorpus des *Stockholm-Umeå-Corpus* (*SUC*, (Gustafson-Capková & Hartmann 2006)). Das SUC ist mit morphosyntaktischen Tags annotiert, die im Gegensatz zu den Tags des STTS neben rein syntaktischen zusätzlich morphologische Information enthalten (vgl. Tabelle 5.2, 3. Spalte). Da der für das Parsing des schwedischen Input verwendete Chunk-Parser SPARK (mehr dazu in 5.3.3) als Input eine mit Tags aus dem für das *Parole*-Projekt entwickelten Tagset annotierte Datei benötigt, mussten die SUC-Tags in die Pendants des Parole-Tagsets transformiert werden. Die Transformation von SUC-Tags in Parole-Tags und vice versa ist jedoch weitgehend unproblematisch, da für beinahe jedes Tag eine exakte Entsprechung im jeweils alternativen Tagset gefunden werden kann. Für die geringe Anzahl von Tags, die über keine exakten Entsprechungen verfügen, werden alternative Tags verwendet. Das bezieht sich z.B. auf Partizipien, die im SUC mit einem eigenen Tag versehen werden, während sie in mit Parole-Tags annotierten Texten als Adjektive markiert werden (Gustafson-Capková & Hartmann 2006: 19). Während sich die Tags inhaltlich also nur marginal unterscheiden, ist der äußerliche Aufbau grundsätzlich unterschiedlich. Das SUC-Tagset ist für den menschlichen Betrachter generell leichter zu lesen (Gustafson-Capková & Hartmann 2006: 19), während die eher kryptisch anmutenden Parole-Tags in erster Linie auf maschinelle Lesbarkeit ausgelegt sind (vgl. Tabelle 5.2, 1. Spalte).

Um eine auf Parole-Tags basierende Parameterdatei zu generieren, wurden die Tags des SUC in dem erwähnten SUC-Teilkorpus durch die entsprechenden Parole-Tags

PAROLE	**Beschreibung (schwedisch)**	**SUC**
AF00PG0S	particip perf. utrum/neutrum plur. obest./best. genitiv	PC PRF UTR/NEU PLU IND/DEF GEN
AQPMSGDS	adjektiv positiv mask. sing. bestämd genitiv	JJ POS MAS SIN DEF GEN
NCNPG@DS	substantiv neutrum plur. bestämd genitiv	NN NEU PLU DEF GEN
D0@0P@S	determinerare utrum/neutrum pluralis obestämd/bestämd	DT UTR/NEU PLU IND/DEF

Tabelle 5.2: Auswahl an Parole-Tags und deren SUC-Entsprechungen

ersetzt. Von diesem nun mit Parole-Tags annotierten Korpus wurde die für den TreeTagger notwendige Parameterdatei für Schwedisch generiert. Analog zum deutschen TreeTagger wird neben dem PoS-Tag auch das Lemma erzeugt. Auch die Behandlung jener Wörter, für die kein Lemma gefunden werden kann und die folglich mit einem <*unknown*>-Tag markiert werden, erfolgt analog zu jenen des deutschen Outputs. Beispiel (5) zeigt den mit Parole-Tags versehenen Output des schwedischen TreeTaggers für den Satz *Halmstad kom in i matchen med ett mycket viktigt mål på tilläggstid i första halvlek.*

(5)

Halmstad	NP00N@0S	Halmstad
kom	V@IIAS	komma
in	QC	in
i	SPS	i
matchen	NCUSN@DS	match
med	SPS	med
ett	DI@NS@S	en
mycket	AQPNSNIS	mycken
viktigt	AQPNSNIS	viktig
mål	NCNSN@IS	mål
på	SPS	på
tilläggstid	NCUSN@IS	tilläggstid
i	SPS	i
första	MO00N0S	första
halvlek	NCUSN@IS	halvlek
.	FE	.

5.3.2 Morphologische Analyse

Die Verwendung des mit relativ komplexen Tags annotierten SUC-Korpus als Ausgangsressource für das Training des TreeTaggers hat zur Folge, dass die Analyse des schwedischen Inputs zumindest teilweise morphologische Information enthält. Der Genus der analysierten Nomen wird beispielsweise entweder durch ein N (Neutrum) oder ein U (Utrum) an der dritten Stelle der Parole-Tags markiert (vgl. Tabelle 5.3). Der Output des deutschen TreeTaggers enthält im Gegensatz dazu keinerlei morphologische Informationen. Nomen werden gemäß dem STTS entweder mit NN (Nomen) oder NE (Eigennamen) annotiert.

PAROLE-Tag	**morphologische Information**	**Beispiel**
`NCNSN@IS`	Substantiv (*NC*), Neutrum (*N*), Singular (*S*), Nominativ (*N*), unbestimmt (*IS*)	töväder
`NCNPG@DS`	Substantiv (*NC*), Neutrum (*N*), Plural (*P*), Genitiv (*G*), bestimmt (*DS*)	landstingens
`NCUPG@IS`	Substantiv (*NC*), Utrum (*N*), Plural (*P*), Genitiv (*G*), unbestimmt (*IS*)	nationers

Tabelle 5.3: Auswahl an Parole-Tags für schwedische Nomen

Für die Erstellung des SCall-Lexikons und in weiterer Folge für die Generierung sind einige morphologische Informationen wie das Genus der Nomen jedoch unbedingt notwendig, weshalb eine morphologische Analyse des Inputs notwendig ist. Diese Analyse wird vom Programm *Morphix* übernommen. Morphix ist ein speziell für Deutsch entwickeltes, regelbasiertes morphologisches Analyse- und Generierungswerkzeug (Finkler & Neumann: 1988). Da Morphix nur in einer *Lisp*-Version vorliegt, werden sämtliche Nomen des deutschen Inputs in das geeignete Format überführt und dem Lisp-Prozess übergeben. Die von Morphix analysierten Daten bezüglich Genus werden im Verlauf der weiteren Bearbeitung berücksichtigt.

5.3.3 Parsing

Unter Parsing[2] versteht man *„die Zuordnung einer Strukturbeschreibung zu Sätzen durch einen Computer“* (Glück 2000: 596). Die Aufgabe eines Parsers besteht also darin, einem gegebenen Satz eine syntaktische Struktur zuzuweisen (weiterführende Literatur zu Parsern: z.B. Carroll 2003). Zum Parsing der deutschen Sätze wurden in SCall zwei verschiedene Parser integriert: der auf probabilistischen Methoden beruhende Stanford-Parser sowie der im TreeTagger integrierte Chunk-Parser. Während der Stanford-Parser die gesamte Struktur eines Satzes ausgibt, ist die

[2]In deutschsprachigen Publikationen wird mitunter die Bezeichnung *Satzanalyse* verwendet, vgl. Glück 2000: 596

Aufgabe eines Chunk- oder auch Shallow-Parsers darauf beschränkt, den zu analysierenden Satz in einzelne Phrasen (bzw. Chunks) zu zerlegen, ohne jedoch diese in Zusammenhang zu bringen. Der Chunk-Parser sucht also nach Teilstrukturen innerhalb eines Satzes, ohne die Struktur des Satzes „als Ganzes“ zu berücksichtigen. Obwohl der Output des Stanford-Parsers qualitativ als hochwertiger eingeschätzt werden kann, liegt der Vorteil des Chunk-Parsers zweifelsohne in der Geschwindigkeit. Während der Stanford-Parser für die Analyse eines einzigen Satzes, abhängig von der verwendeten Hardware, bis zu 30 Sekunden benötigt, analysiert der Chunk-Parser den gleichen Satz in weniger als einer Sekunde. Ein weiterer Nachteil des Stanford-Parsers sind die PoS-Tags, die vom Parser produziert werden (gemäß dem STTS), während für den Output des Chunk-Parsers die Tags des PoS-Taggers verwendet werden. Die Wahl der vom Stanford-Parser generierten Tags ist eher unzuverlässig und in gewissen syntaktischen Konstruktionen grundsätzlich falsch. Eine Wortart, die generell mit falschen Tags annotiert wird, sind die Perfektpartizipien in Verbindung mit der finiten Form des Verbs *sein*. Grundsätzlich wird das finite Verb in diesen Fällen nicht als Auxiliar, sondern als Kopulaverb behandelt und das folgende Partizip Perfekt nicht als solches, sondern als prädikatives Adjektiv (ADJD). Folglich werden sowohl in (6a) als auch in (6b) sämtliche Satzteile rechts vom finiten Verb als Teile einer Adjektivphrase behandelt ((6c) und (6d) zeigen den Output der entsprechenden Sätze im Präteritum).

(6) (a) (ROOT (S (NE Peter) (VAFIN ist) (AP (PP (APPR in) (ART die) (NN Schule)) (ADJD gegangen.))))
(b) (ROOT (S (NE Maria) (VAFIN ist) (AP (ADJD krank) (ADJD gewesen.))))
(c) (ROOT (S (NE Peter) (VVFIN ging) (PP (APPR in) (ART die) (NN Schule.))))
(d) (ROOT (S (NE Maria) (VAFIN war) (AP (ADJD krank.))))

Aufgrund dieser und anderer Unzuverlässigkeiten hinsichtlich der PoS-Annotation werden die vom Stanford-Parser generierten Tags durch jene des TreeTaggers ersetzt. Im Fall der falsch annotierten Perfektpartizipien aus (6) würden die korrekten Tags *VVPP* bzw. *VAPP* (Bsp. (7a) und (7b)) übernommen werden.

(7) (a) Peter NE ist VAFIN in APPR die ART Schule NN gegangen VVPP . $.
(b) Maria NE ist VAFIN krank ADJD gewesen VAPP . $.

Zum Parsing der schwedischen Eingabe wurde der Chunk-Parser *SPARK* integriert. SPARK basiert auf einer kontextfreien Grammatik sowie Earleys Algorithmus (Ay-

cock 1998: 73; schwedische Implementierung: Megyesi 2002). Im Gegensatz zu Parsern wie dem Stanford-Parser müssen die von SPARK zu verarbeitenden Sätze mit PoS-Tags der bereits erwähnten schwedischen Version des Parole-Tagsets annotiert sein. Aus diesem Grund beginnt der Parseprozess für die schwedische Eingabe erst nach Beendigung des PoS-Taggings sowie der Umformung der Taggingdaten in das für SPARK erforderliche Inputformat (gemäß Tabelle 5.4).

Format	**Beispiel**	**Muster**
TreeTagger (Output)	Smygrustning *tab* NCUSN@IS *tab* smygrustning av *tab* SPS *tab* av raketvapen *tab* NCNPN@IS *tab* raketvapen	Wort *tab* Parole-Tag *tab* Lemma *tab* = Tabulator
SPARK (Input)	Smygrustning/NCUSN@IS av/SPS raketvapen/NCNPN@IS	Wort/Parole-Tag

Tabelle 5.4: Output- bzw. Inputformat von TreeTagger respektive SPARK-Parser

5.3.4 Lexikongenerierung

Neben dem Parsing sowie dem PoS-Tagging der Eingabesätze wird durch die Erstellung eines Lexikons ein lexikalischer Zusammenhang zwischen den Wörtern der Eingabesätze hergestellt. Als Ausgangsmaterial für die Erstellung des Lexikons diente eine von der Internet-Plattform *Pauker*[3] zur Verfügung gestellte Schwedisch-Deutsch-Wortliste. Pauker ist ein nicht-kommerzielles Online-Forum für Sprachlernende, das u.a. umfassende Online-Wörterbücher für die unterstützten Sprachen bereitstellt. Die Wörterbücher werden, ähnlich dem Prinzip von Wikipedia, ausschließlich durch die (registrierten) Benutzer erstellt sowie ergänzt, was zur Folge hat, dass deren Umfang stetig zunimmt. Das größte Wörterbuch auf Pauker stellt Portugiesisch mit mehr als 148.000 Einträgen[4] dar, für Schwedisch kann immerhin auf 46.502 Einträge zugegriffen werden[5]. Die Wörterbücher können entweder online in Form einer Suchmaske verwendet oder als *csv*-Datei auf einer lokalen Festplatte zur Weiterverarbeitung gespeichert werden. Für die Verwendung in SCall wurde die Lexikondatei für Schwedisch nachbearbeitet und überflüssige Information wie die für die Erstellung des Lexikons nicht relevanten Genusbezeichnungen oder zusätzlich angegebene Pluralendungen entfernt. Das Lexikon für SCall wird generiert, indem die deutschen, vom TreeTagger generierten, Lemmata in der Pauker-

[3]URL: www.pauker.at [31.03.2008]
[4]Stand: 31.03.2008
[5]Stand: 31.03.2008

Wortliste gesucht und alle gefundenen schwedischen Entsprechungen gespeichert werden. Diese Entsprechungen werden mit der Liste der für Schwedisch generierten Lemmata verglichen. Sollte eine Entsprechung gefunden werden, wird das entsprechende schwedische Lemma mit dem deutschen Äquivalent im Lexikon gespeichert.

Eigennamen sind mit Ausnahme von wichtigen Städte- und Ländernamen (z.B. Österreich-Österrike) in der Regel nicht im Wörterbuch enthalten, da Eigennamen, in erster Linie Vornamen, jedoch häufig homonym verwendet werden, werden jene mit dem PoS-Tag für Eigennamen (im STTS: NE) annotierten Lemmata ohne Umweg über das Lexikon direkt mit dem schwedischen Lemmaliste verglichen und im Falle eines Fundes der exakten Entsprechung im Lexikon gespeichert. Wird für ein deutsches Lemma keine Entsprechung gefunden, muss das korrespondierende schwedische Lemma während der benutzerseitigen Nachbehandlung ausgewählt werden.

Durch die Verwendung der *pauker.at*-Wortliste ist die Lexical-Lookup-Prozedur gewissen Einschränkungen unterworfen. Da die Eingabeform der Wörterbucheinträge nicht standardisiert ist, sind zahlreiche Einträge aufgrund von formalen Diskrepanzen trotz Nachbehandlung der Wortliste unbrauchbar. Ferner ist die Anzahl der Lexikoneinträge mit etwa 45.000 (wobei die zahlreichen Mehrfacheinträge mitgezählt sind) beschränkt, der für die Verwendung in SCall notwendige Grundwortschatz sollte jedoch abgedeckt sein.

5.3.5 Grammatikalische Funktionen

Nach dem Parsing und dem PoS-Tagging der Eingabedaten müssen die ermittelten Phrasen den entsprechenden grammatikalischen Funktionen (Subjekt, Prädikat, Objekte, etc.) zugewiesen werden, um in weiterer Folge die Verbvalenzen korrekt ermitteln zu können. Dieser Verarbeitungsschritt wird von SCall und somit im Gegensatz zu den meisten anderen Analyseprozeduren nicht von einer externen Anwendung durchgeführt.

Am Beginn dieser Prozedur wird für jeden zu analysierenden Satz ein Array generiert, das für jede mögliche grammatikalische Funktion (Prädikat, Subjekt, Direktes Objekt, Indirektes Objekt etc.) einen anfangs leeren Speicherplatz (in weiterer Folge als *Slot* bezeichnet) enthält. Anschließend wird dieses Array regelbasiert Slot für Slot mit Phrasen aus dem Parsing-Output „befüllt“. Die erste zu ermittelnde Funktion ist das Prädikat. Die für das Prädikat notwendigen Informationen können im Grunde vollständig aus dem PoS-Output gewonnen werden (finite und infinite Verben; abgetrennte Verbzusätze (STTS-Tag PTKVZ, wie in *er kommt an*) etc.). Nachdem das Prädikat gefunden wurde, wird, falls vorhanden, das entsprechende

Subjekt gesucht. Als Subjekt kommen Nominalphrasen und Terminale, die funktionell einer Nominalphrase entsprechen, vom Parser jedoch nicht als solche markiert worden sind (im Stanford-Parser üblich), in Frage. Zu diesen Terminalen gehören unter anderem Nomen, Personalpronomen (*ich, du etc.*; STTS-Tag: *PPER*), substituierende Demonstrativpronomen (*dies, jenes etc.* STTS-Tag: *PDS*) und substituierende Indefinitpronomen (*man, nichts, etwas etc.* STTS-Tag: *PIS*). Alle Phrasen, die keiner NP entsprechen, werden für die Ermittlung des Subjekts nicht berücksichtigt. Ferner werden Nominalphrasen, die aufgrund der strukturellen Position nicht für die gesuchte Funktion in Frage kommen, ausgeschlossen. Zu derartigen Phrasen gehören Nominalphrasen, die sich innerhalb von Präpositionalphrasen befinden (z.B. *(PP auf dem (NP hohen Berg))*).

In weiterer Folge wird im Satz jene NP gesucht, die *am ehesten* das Subjekt repräsentieren *könnte*. Als erste Annahme wird sowohl für Deutsch als auch Schwedisch jene NP gewählt, die innerhalb des Satzstruktur vor der Verbphrase bzw. dem finiten Verb steht. Wird an dieser Position eine NP gefunden, wird der Kasus der NP, falls möglich, festgestellt. Bei Eigennamen ist die Feststellung des Kasus, abgesehen vom Genitiv, nicht möglich, enthält die Phrase jedoch kasusmarkierende Bestandteile wie Artikel, kann aufgrund dieser Schlüsselwörter festgestellt werden, ob die Nominalphrase für den gesuchten Kasus in Frage kommt. Sollte die Kasuskontrolle für die aktuelle Funktion fehlschlagen, wird die „nächstwahrscheinliche“ Nominalphrase untersucht, im Fall des Subjekts die erste NP nach dem finiten Verb usw. Ist ein Subjekt gefunden, wird der Subjektslot mit der gefundenen Phrase gefüllt und nach der gleichen Prozedur nach einem direkten und einem indirekten Objekt gesucht. Diese sind, abhängig von den (leider unbekannten) Valenzen des Verbs, nicht obligatorisch. Sollte aufgrund von NP-Mangel (falls z.B. die einzige mögliche NP bereits als Subjekt in Verwendung ist wie im Satz *Peter fährt nach Spanien*) oder aufgrund von negativer Kasuskontrolle (falls der Kasus der gefundenen NP nicht dem notwendigen Kasus entspricht; wie im Satz *Gestern hat mich Peter angerufen*, falls das Dativobjekt gesucht wird) der Slot nicht mit einer Phrase befüllt werden können, wird er als leer markiert. Sollte das Verb des Satzes ein Kopulaverb sein (im Dt. *sein, bleiben, werden, scheinen, gelten als*), werden sämtliche Objektslots als leer markiert. In diesem Fall ist die Suche nach dem in Kopulakonstruktionen obligaten Prädikativ, das durch das Verb mit dem Subjekt verknüpft wird, von Interesse. Dieses Prädikativ kann sowohl durch eine NP (*Peter ist ein Lehrer*) als auch durch eine Adjektivphrase (*Peter ist sehr intelligent*) realisiert werden, die Suchprozedur verläuft jedoch analog zu jener für das Subjekt beschriebenen.

Ähnlich verläuft die Suche nach Präpositionalphrasen, die als Valenz angesehen werden, wie im Satz *Die Schulklasse aus Graz fährt mit dem Zug von Wien nach*

Salzburg. Da diese PPs thematische Rollen repräsentieren, werden sie in weiterer Folge als Obl_θ bezeichnet. Während pro Satz nur ein Subjekt, Prädikat etc. vergeben werden kann, ist die Obl_θ-Anzahl nicht begrenzt. Aus diesem Grund wird pro PP, die als Obl_θ in Frage kommt, ein Eintrag generiert und der Slot mit der entsprechenden Phrase befüllt. Jene PPs, die aufgrund der syntaktischen Struktur nicht als Obl_θ in Frage kommen, werden nicht berücksichtigt (z.B. *die Schulklasse (PP aus Graz)*).

Die Methode zur Feststellung der grammatikalischen Funktionen ist in hohem Maße abhängig von der Qualität des Parser-Outputs. Aus diesem Grund variieren die Ergebnisse dieser Prozedur und gegebenenfalls müssen die grammatikalischen Funktionen durch die entsprechenden Werkzeuge im User Interface (sh. 5.4) benutzerseitig zugewiesen werden.

5.4 Das User Interface

Nach der Eingabe der Übungssätze und der Verarbeitung der Daten erscheint das *Graphical User Interface* (kurz *GUI*) für die Bearbeitungsprozeduren in Form eines neuen Fensters (Abbildung 5.2). Dieses Interface ermöglicht dem Benutzer, die Ergebnisse der Datenverarbeitung zu kontrollieren und gegebenenfalls zu manipulieren. Ferner enthält das Interface die für die semantische Annotation notwendigen Werkzeuge.

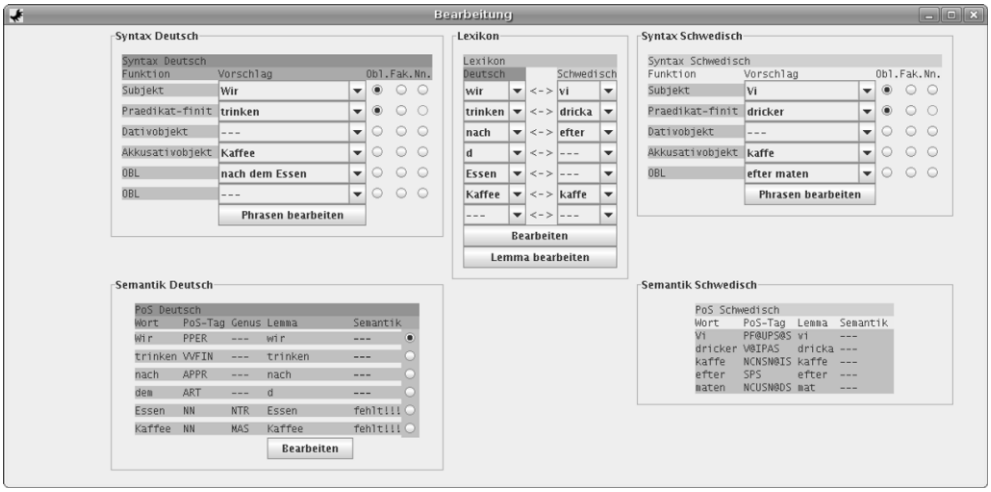

Abbildung 5.2: Bearbeitungsfenster in SCall

Das Fenster ist in fünf Bereiche unterteilt. Auf der linken Seite des Fensters werden in zwei Bereichen die aufbereiteten Ergebnisse sowohl des PoS-Taggings als auch des Parsings des deutschen Eingabesatzes gezeigt, auf der rechten Seite die entsprechenden Analyseergebnisse für Schwedisch. Zwischen den sprachenspezifischen

Ergebnisfeldern befindet sich das im Zuge der Lexical-Lookup-Prozedur generierte Wörterbuch.

In den Syntax-Feldern wird der in die grammatikalischen Funktionen zerlegte Eingabesatz dargestellt. Durch die Verwendung von Pull-Down-Menüs können alternative Einträge gewählt werden, wobei die Menüs in der Regel sämtliche Phrasen enthalten, die für die entsprechende Funktion in Frage kommen können (unabhängig von der Position im Satz). Dementsprechend können sämtliche Nominalphrasen als Subjekt ausgewählt werden[6], für Obl_θ-Ergänzungen sämtliche Präpositionalphrasen etc. Sollte die entsprechende Phrase aufgrund von Fehlern, die während des Parsings aufgetreten sind, nicht in der Liste des Menüs enthalten sein, besteht die Möglichkeit, zusätzliche Phrasen selbst auszuwählen und dadurch die entsprechenden Auswahllisten zu ergänzen. Ferner muss der Benutzer durch das Markieren bzw. Nichtmarkieren der Kontrollkästchen (bzw. der *Checkboxen*), die sich neben jeder Funktion befinden, entscheiden, ob die entsprechende Funktion im Bezug auf die Verbvalenzen obligatorisch bzw. fakultativ ist. Das Verb selbst sowie das Subjekt sind *per default* als obligatorisch markiert, die restlichen Funktionen müssen jedoch festgelegt werden.

Das Lexikon-Feld befindet sich in der Mitte des Bearbeitungsfensters. Die Wortpaare werden entsprechend dem Ergebnis der Lexical-Lookup-Prozedur in Pull-Down-Menüs dargestellt. Sollte keine Entsprechung gefunden worden sein, wird im dem korrespondierenden Wort entsprechenden Pull-Down-Menü ein leerer Eintrag (in SCall durch drei Minus markiert: —) dargestellt. Das Lexikon kann natürlich verändert werden, gegebenenfalls können Wörter ergänzt werden. Das Ergänzen von Wörtern ist in einigen Fällen aufgrund der durch die geringe Komplexität des zugrundeliegenden Lexikons bedingten Einschränkungen sogar obligatorisch. Das gilt z.B. für Wörter, deren Entsprechung in der Zielsprache aus mehr als einem Wort besteht, im Fall von Deutsch-Schwedisch betrifft das unter anderem einige Zirkumpositionen bzw. Klammer-Präpositionen wie das der deutschen Präposition *für* in einem Satz wie *ich tat es wegen dir* entsprechende schwedische *för ... skull* (*jag gjorde det för din skull*). Während des aufgrund der erwähnten Einfachheit des Lexikons auf die Suche nach Einzelwörtern beschränkte Lexical-Lookup-Prozederes würde in diesem Fall für *für* keine schwedische Entsprechung gefunden und das entsprechende Auswahlmenü als leer markiert werden. Im Lexikonfeld können ferner wortspezifische Änderungen vorgenommen werden, z.B. die Richtigstellung von falschen PoS-Tags.

[6] ausgenommen sind natürlich jene Nominalphrasen, die im Zuge der Ermittlung der grammatikalischen Funktionen nicht berücksichtigt wurden, vgl. 5.3.5

Im Semantikfeld, das sich unterhalb des Syntaxfeldes befindet, werden schließlich die Selektionsrestriktionen festgelegt. Dieser Vorgang wird in 5.4.1 beschrieben.

5.4.1 Umsetzung und Auswahl der Selektionsrestriktionen in SCall

Vier verschiedene Wortarten werden in SCall mit Selektionsrestriktionen annotiert: Nomen, Präpositionen, Adjektive und Verben. Die Selektionsrestriktionen der Nomen sowie teilweise jene der Adjektive werden benutzerseitig festgelegt, während die Restriktionen der Präpositionen und der Verben vom Programm zugewiesen werden.

5.4.1.1 Selektionsrestriktionen für Nomen

Die Selektionsrestriktionen für die Nomen werden vom Benutzer aus einer Liste aus von SCall generierten Vorschlägen ausgewählt, wobei die Zuteilung der Selektionsrestriktionen im User Interface im Semantikfeld durchgeführt wird. Das zu annotierende Nomen wird durch das Markieren des entsprechenden Kontrollkästchens ausgewählt. Durch die Betätigung des Schalters „Semantik bearbeiten“ werden für das entsprechende deutsche Nomen in einer von der Internetplattform *Pauker.at* (vgl. 5.3.4) zur Verfügung gestellten Liste sämtliche englische Übersetzungen gesucht und für die jeweiligen Wörter die entsprechenden Einträge in WordNet ermittelt. Da aufgrund von Homonymie oder Polysemie mit hoher Wahrscheinlichkeit mehr als eine einzige Übersetzung pro Nomen und in weiterer Folge pro Übersetzung mehrere WordNet-Einträge gefunden werden können, kann davon ausgegangen werden, dass für die meisten Nomen mehr als nur ein WordNet-Eintrag gefunden wird. Deshalb muss die Wahl der korrekten Restriktion vom Benutzer von SCall getroffen werden. Aus diesem Grund werden die gefundenen Restriktionen in einem Dialogfenster (Abbildung 5.3) gezeigt und der Benutzer hat die Möglichkeit, die seiner Meinung nach am besten geeignete Restriktion auszuwählen. Neben der gefundenen Restriktion werden zusätzlich sämtliche Hyperonyme angezeigt, wodurch die Möglichkeit besteht, den Generalisierungsgrad der Restriktion dem Kontext entsprechend anzupassen. Je höher eine gewählte Selektionsrestriktion in der Hierarchie liegt, desto ungenauer ist natürlich der Grad der Einschränkung. Andererseits wird durch die Wahl einer hierarchisch sehr tief liegenden Restriktion der Verwendungsrahmen eines Wortes dementsprechend eingeschränkt. Die Wahl des terminalen Konzeptes als Selektionsrestriktion schränkt den Verwendungsrahmen überhaupt nur auf das entsprechende Wort sowie dessen Synonyme ein, was vor allem im Fall von abstrakten Begriffen auch durchaus sinnvoll sein kann.

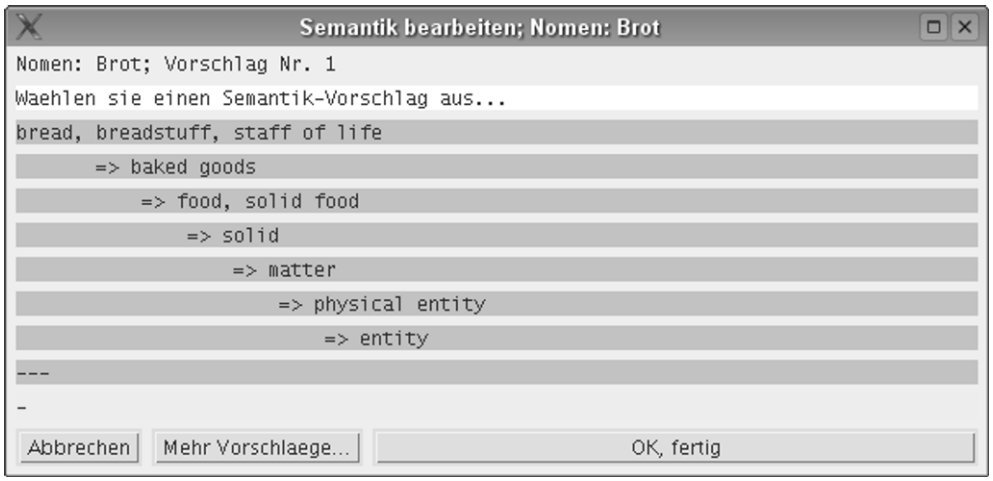

Abbildung 5.3: Auswahlfenster für Selektionsrestriktionen; Beispiel: Vorschlag für das Nomen *Brot*

5.4.1.2 Selektionsrestriktionen für Adjektive

Adjektive übernehmen die SelrRestr automatisch von den durch sie modifizierten Nomen. Wird folglich in der NP *die schwarze Katze* das Nomen mit der WordNet-Selektionsrestriktion *[carnivore]* annotiert, wird der Lexikoneintrag des Adjektivs *schwarz* durch die SelRestr *[carnivore]* ergänzt. Somit ist die Verwendung des Adjektivs *schwarz* auf die Modifikation von fleischfressenden Säugetieren beschränkt. Wird das Adjektiv in weiterer Folge in einem anderen Kontext analysiert (z.B. *das schwarze Motorrad*; Selektionsrestriktion von Motorrad: *[motor vehicle, automotive vehicle]*), wird der Lexikoneintrag um die entsprechende Selektionsrestriktion erweitert und das Verwendungsspektrum des Wortes somit vergrößert.

Adjektive haben die Eigenschaft, dass deren Reihenfolge in attributiver Stellung durch deren semantische Komplexität determiniert wird. Je komplexer ein Adjektiv aus semantischer Sicht ist, desto weiter vorne steht es in der dem modifizierten Nomen vorausgehenden Adjektivphrase. Die zugrundeliegende Komplexitätshierarchie sieht folgendermaßen aus:

Unregelmäßige Adjektive → Evaluative Adjektive → Dimensionsadjektive → Privativa → Farbenbezeichnungen → Nationalitätsbezeichnungen → NOMEN

Die unterschiedlichen Komplexitätsgrade werden in Tabelle 5.5 detailliert erklärt.

Um Phrasen wie (8a) zu verhindern, muss die Komplexität während der Generierung der Adjektivphrase berücksichtigt werden, indem die aus dem Lexikon gewählten Adjektive in die korrekte Reihenfolge (8b) gebracht werden.

(8) (a) ?? der österreichische unsympathische kleine H.C. Strache
(b) der kleine unsympathische österreichische Politiker H.C. Strache

Komplexizitätsgrad	Beschreibung	Beispiel
Unregelmäßige Adjektive	Nicht graduierbare, nur attributiv verwendete Adjektive.	ehemalig, vermeintlich
Evaluative Adjektive	Beurteilende Adjektive; Antonym vorhanden.	schön-hässlich, gut-schlecht
Dimensionsadjektive	Adjektive, die Dimensionen bezeichnen.	groß-klein, hoch-niedrig, breit-schmal
Privativa	Privativa bezeichnen meist negative Eigenschaften bzw. das Fehlen einer lokalen Eigenschaft. Sie verfügen über keinen Superlativ (z.B. blind-*blinder).	blind, tot, krank
Farbenbezeichnungen	Sämtliche Adjektive, die Farben bezeichnen.	grün, rot, gelb
Nationalitätsbezeichnungen	Adjektive, die Nationalitäten, Völker, Regionen u. dergl. bezeichnen.	französisch, afrikanisch, tirolerisch

Tabelle 5.5: Komplexitätsgrade von deutschen Adjektiven

Aus diesem Grund werden die Lexikoneinträge der Adjektive durch jeweils eine von sechs Selektionsrestriktionen, die den Knoten der oben gezeigten Komplexitätshierachie entsprechen, erweitert. Die Auswahl der Restriktionen muss vom Benutzer übernommen werden, indem sie im dafür vorgesehenen Dialogfenster (Abbildung 5.4) entsprechend ausgewählt werden.

5.4.1.3 Selektionsrestriktionen für Verben

Die Verben werden in SCall semantisch nicht kategorisiert, der Verwendungsrahmen jedoch durch die Belegung der Valenzpositionen durch Selektionsrestriktionen definiert. In den Lexikoneinträgen der Verben wird pro Valenz die SelRestr der in der entsprechenden Position möglichen Nomen berücksichtigt. Die Zuweisung der entsprechenden Nomen bzw. Nominalphrasen geschieht während der Eingabeprozedur (vg. 5.3.5). Im Fall von Obl_θ werden sowohl die Präposition als auch die maßgebliche SelRestr des die Präposition modifizierenden Nomens in das Lexikon aufgenommen (vgl. 5.4.1.4). In Abbildung 5.5 wird die (mögliche[7]) Analyse des Satzes *Fritz schenkte seinem Freund ein neues Auto* und der daraus resultierende Lexikoneintrag für das Verb *schenken* gezeigt.

[7] da die Selektionsrestriktionen benutzerseitig ausgesucht werden, sind andere Restriktionen in den entsprechenden Positionen durchaus denkbar.

Abbildung 5.4: Dialogfenster für die Auswahl von Selektionsrestriktionen für Adjektive

Inputsatz	Fritz ⇳	schenkte	seinem	Freund ⇳	ein	neues	Auto ⇳
SelRestr (f. Nomen)	[male, male person]			[person, individual]			[self-propelled vehicle]
Gramm. Funktion	Subjekt			O3			O4
Lexikoneintrag:	schenken	Subj[male, male person]		O3[person, individual]		O4[self-propelled vehicle]	

Abbildung 5.5: Vergabeablauf der Selektionsrestriktionen für Verben in SCall (Beispiel)

Der in 5.5 generierte Lexikoneintrag würde in weiterer Folge die Generierung von Sätzen, deren Subjekt, O3 bzw. O4 den im Eintrag für *schenken* enthaltenen Selektionsrestriktionen entsprechen, zulassen. In diesem Fall wäre die Verwendung durch die relativ restriktive SelRestr für O4 (*[self-propelled vehicle]*) sehr eingeschränkt, durch die Analyse von weiteren Eingabesätzen, die das Verb *schenken* enthalten, kann der Verwendungsrahmen durch die Erweiterung durch weitere Selektionsrestriktionen jedoch ausgeweitet werden. Durch den Satz *Fritz schenkte seinen Eltern ein kleines Haus* würde der O4-Eintrag beispielsweise durch die SelRestr *[housing, lodging, living accommodations]* ergänzt werden.

5.4.1.4 Selektionsrestriktionen für Präpositionen

In SCall werden Präpositionen mit Selektionsrestriktionen annotiert, um den semantischen Inhalt der von der Präposition abhängigen Nominalphrasen zu steuern.

Wie im Fall der Verben und der Adjektive werden diese Selektionsrestriktionen nicht vom Benutzer ausgesucht, sondern von SCall automatisch zugewiesen. Nachdem der Kopf der entsprechenden Nominalphrase ermittelt wurde, wird dessen Selektionsrestriktion im entsprechenden Lexikoneintrag der Präposition ergänzt. In der Präpositionalphrase *in der Kirche* würde die Selektionsrestriktion von *Kirche* (entsprechende SelRestr z.B. *[building, edifice]*) herangezogen werden. Analog zu den Verben (vgl. 5.4.1.3) kann der Verwendungsrahmen durch die Analyse von weiteren die entsprechende Präposition enthaltenen Sätzen ausgeweitet werden.

Neben den Selektionsrestriktionen wird in den Lexikoneinträgen der Präpositionen auch der Kasus, der durch Kasusrektion von den entsprechenden Präpositionen gefordert wird, vermerkt. Im Fall von Deutsch können Präpositionen Genetiv, Dativ oder Akkusativ fordern. Ferner stehen Präpositionen oft ohne erkennbare Kasusforderung, beispielsweise wenn das Substantiv artikellos gebraucht wird (z.B. *per Anhalter*)(Schröder 1990, 244). Angaben über die Kasusrektion werden während der morphologischen Analyse ermittelt, können jedoch gegebenenfalls vom Benutzer verändert werden.

5.5 Generierungsvorgang in SCall

5.5.1 Einleitung

Im Zuge dieses Prozesses werden Übungssätze und ausgehend davon Übungsaufgaben für die Verwendung im Fremdsprachenunterricht erzeugt. Das Generierungsmodul besteht im Grunde aus zwei Komponenten: dem Generierungsprozess und dem Prozess zur Erstellung der Übungsaufgaben.

Während des Generierungsprozesses wird, ausgehend vom im Zuge des Eingabe-Prozesses generierten, mit semantischer Information annotierten Lexikon und einer Grammatik eine vom Benutzer festgelegte Anzahl von Sätzen generiert. Die Generierung dieser Sätze erfolgt zufällig, der Fokus liegt auf der semantischen und grammatikalischen Richtigkeit der produzierten Sätze. Dieser Prozess wird in 5.5.2 im Detail beschrieben.

Im darauf folgenden Verarbeitungsprozess werden die generierten Sätze nach vom Benutzer ausgewählten Kriterien zu Übungsaufgaben verarbeitet bzw. umstrukturiert. In SCall wurde dieser Teil des Programms für die Erstellung von Lückentexten wie jenem in Abbildung 5.6 gezeigten Beispiel umgesetzt. Der Aufbau dieser Prozedur wird in 5.5.4 beschrieben.

PRESENT PERFECT SIMPLE OR PROGRESSIVE

1. Where is Mike? He ..... (go) on holiday.
2. ..... (you ever be) in Paris?
3. She ..... (not speak) Swedish for five years. I fear she will have difficulties in Sweden.
4 Frank ..... (try) to learn Japanese, but it is too difficult for him

Abbildung 5.6: Lückentext für die Verwendung im Englischunterricht

5.5.2 Der Generierungsprozess

5.5.2.1 Grundlegendes zur Generierung von natürlicher Sprache

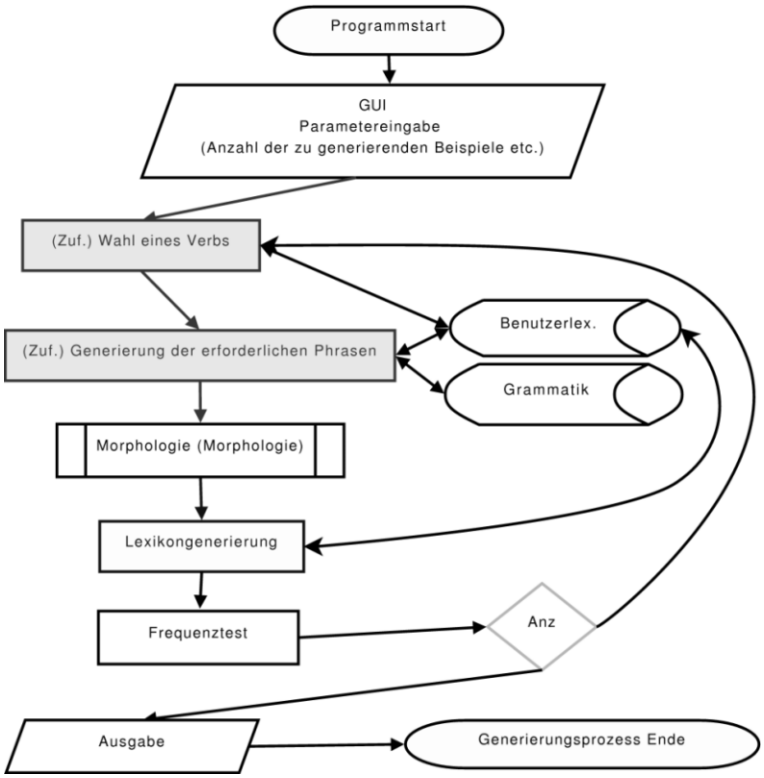

Abbildung 5.7: Ablaufdiagramm für das Generierungsmodul von SCall

Das Ziel von Sprachgenerierung ist die Überführung von nicht-linguistischen Daten in eine linguistische Form (Bateman & Zock 2003: 285). Bateman & Zock (2003: 287ff) unterteilen den Generierungsprozess in vier Tasks: Makroplanung (*macroplanning*), Mikroplanung (*microplanning*), linguistische Realisierung (*linguistic realization* oder *surface realization*) und Präsentation (*presentation*). Während der Makroplanungsphase wird, ausgehend von den vorhandenen Ressourcen und dem vorgegebenen bzw. gewünschten Ziel der Generierung, ein Textplan er-

stellt. Unter dem Ziel der Generierung werden u.a. die Art des Textes, der generiert werden soll (Bericht, Zusammenfassung etc.), die Länge des zu produzierenden Textes und der Modus (gesprochener oder geschriebener Text) zusammengefasst (Bateman & Zock 2003: 289). Im Zuge der *Mikroplanung* werden jene Daten, die für das gewünschte Ergebnis relevant sind, aufbereitet. Während der linguistischen Realisierung werden die in den vorangegangen Prozessen gesammelten und strukturierten Daten schließlich in grammatikalische Konstruktionen überführt. Dazu gehört die Auswahl der syntaktischen Funktionen (Subjekt, Objekt etc.) und daraus resultierend die morphologische Manipulation der Elemente. Im finalen Prozess, der *physischen Präsentation*, werden die generierten linguistischen Daten für das gewünschten Endprodukt nachbearbeitet. Als Endprodukt kann ein gedruckter Text gewünscht sein, wobei in diesem Fall im Präsentationsprozess die Bearbeitung des Layouts im Vordergrund steht. Sollte ein Text in akustischer Form Ziel der Generierung sein, ist eine weitere Bearbeitung mittels Sprachsynthese-Methoden erforderlich (vgl. dazu Dutoit & Stylianou 2003: 323ff).

Die von Bateman & Zock erstellten Prozesse beziehen sich weitestgehend auf die Generierung von *Texten*. Da SCall nur semantisch isolierte *Sätze* produziert, die zu den weiteren produzierten Sätzen in keiner textlinguistischen Beziehung stehen, muss auf satzübergreifende Phänomene wie etwa Kohärenz oder Kohäsion keine Rücksicht genommen werden. Folglich müssen große Teile der Makroplanung in SCall nicht in Betracht gezogen werden. Ein weiterer Aspekt, welcher in einem vollständigen Textgenerierungssystem im Gegensatz zu SCall von Wichtigkeit ist, ist die Frage des Inhalts. Während die Planung des Inhalts während der Makroplanungsphase geschieht, wird dieser in SCall im Grunde vollständig während der linguistischen Realisierung kreiert. Aufgrund des Fehlens eines vorgegebenen Inhaltes ist der Inhalt der von SCall produzierten Sätze nur von sekundärer Relevanz. Das Ziel von SCall besteht auch nicht darin, vordefinierte Inhalte wiederzugeben; die Inhalte der produzierten Sätze sind im Grunde unwichtig, solange Syntax und Semantik korrekt sind. Als den Inhalt in gewisser Weise einschränkend kann natürlich das vorhandene Lexikon angesehen werden: da nur Wörter aus diesem Lexikon Verwendung finden, ist der Inhalt der von SCall produzierten Sätze von den im Lexikon vorhandenen Einträgen abhängig.

5.5.2.2 Generierung in SCall

Die für die Generierung notwendigen Programmbibliotheken wurden großteils an das von Reiter & Venour (2008) entwickelte *Simplenlg*-Package angelehnt. Im Vergleich zum für Englisch realisierten *Simplenlg* musste das Generierungsmodul für

Deutsch aufgrund der komplexeren Morphologie (bedingt u.a. durch das Kasussystem) jedoch entsprechend komplexer ausgeführt werden.

Sowohl die inhaltliche bzw. semantische als auch die syntaktische Generierung beginnen in SCall mit der (zufälligen) Wahl eines Verbs. Neben den syntaktischen Informationen enthält der Lexikoneintrag des entsprechenden Verbs auch jene semantischen Restriktionen, die im Zuge des weiteren Aufbaus des Satzes die Semantik der direkt vom Verb abhängigen Satzelemente vorgibt. Wiederum zufällig, jedoch unter Berücksichtigung der Verbrestriktionen, werden die Köpfe der weiteren Phrasen, deren Anzahl durch die Valenzen des gewählten Verbs bestimmt ist, ausgewählt. Die gewählten Phrasenköpfe verfügen ihrerseits wiederum über Selektionsrestriktionen, die die Wahl der Modifikatoren beeinflussen.

Die Subjektphrase ist klassischerweise eine Nominalphrase, das Haupt dieser Phrase wird entsprechend den vom Verb vorgegebenen Restriktionen gewählt. Die weitere syntaktische Struktur innerhalb dieser Phrase wird vom ausgehend anhand von Regeln aus einer (sehr einfach gehaltenen) Grammatik generiert. Einige Regeln für die Generierung von Nominalphrasen in SCall werden in (9) gezeigt.

(9) (a) number—z###article—unbestimmt DET AP NOUN
(b) number—z###article—bestimmt DET AP NOUN
(c) number—sg PNOUN
(d) number—z DET AP NOUN

Das erste Element jeder Regel enthält lexikalische und morphologische Informationen, die für sämtliche Elemente der entsprechenden Phrase übernommen werden. Die Informationen in (9a) geben z.B. an, dass der Numerus zufällig festgelegt wird (dargestellt durch das *z*) und dass der Artikel unbestimmt ist. Diese Information ist in erster Linie für die spätere morphologische Generierung der Adjektive wichtig, da sie in attributiver Stellung unterschiedlich flektieren (*der schlechte Wein* vs. *ein schlechter Wein*). Sämtliche Phrasen werden mit semantisch zulässigen Grundformen aus dem Lexikon befüllt und in weiterer Folge hinsichtlich Kasus, Geschlecht etc. morphologisch übereingestimmt. Die morphologische Generierung wird, wie die morphologische Analyse der Eingabedaten (vgl. 5.3.2), vom Programm *Morhpix* durchgeführt. Vor der morphologischen Bearbeitung liegen sämtliche Wörter des generierten Satzes nur in der lexikalischen Form vor, die Information darüber, wie die einzelnen Wörter morphologisch manipuliert werden müssen, wird, wie bereits erwähnt, entweder vererbt oder zufällig bestimmt. Erster Fall tritt beispielsweise für die Vergabe des Kasus in Nominalphrasen, die eine grammatikalische Funktion besetzen, ein. Der Kasus dieser Nominalphrasen wird durch die im Lexikoneintrag des entsprechenden Verbs determinierten grammatikalischen Funktion bestimmt (Sub-

Lexem	**Für Morphix notwendige Information**	**Quelle der notw. Information**	**Morphix-Output**
dies	Kasus Numerus Genus	Verb (Valenz; O4 -> Akk.) Num.: Nomen (*Haus*) Genus: Nomen (*Haus*)	diese
groß	Komparation Artikel(j/n) Genus Numerus Kasus	*zufällig pos./komp./superl.* Nomen (*Haus*) Nomen (*Haus*) Nomen (*Haus*) Verb (Valenz; O4 -> Akk.)	großen
Haus	Kasus Numerus	Verb (Valenz; O4 -> Akk.) *zufällig Sg/Pl*	Häuser

Tabelle 5.6: Notwendige Morphologie-Information für die Generierung der Phrase *diese großen Häuser* in Morphix

jekt verlangt Nominativ, direktes Objekt verlangt Dativ etc.). Ähnlich verhält sich die Vergabe des Kasus in Präpositionalphrasen. In diesem Fall vergibt die Präposition den für die folgende Nominalphrase notwendigen Kasus. In manchen Fällen wird für die Morphologie notwendige Information nicht geerbt bzw. von externen Faktoren determiniert, sondern phrasenintern zufällig bestimmt. Dies betrifft z.B. den Numerus von Objekt-Nominalphrasen. Im Gegensatz zu Subjekt NPs müssen diese Phrasen keine Kongruenz mit dem Numerus des Verbs aufweisen, aus diesem Grund wird der Numerus für diese Phrasen zufällig gewählt. Tabelle 5.6 zeigt die für die Generierung in Morphix notwendigen Informationen sowie die jeweiligen Quellen für diese Information für die zu generierende (Akkusativ-)Nominalphrase *die großen Häuser* in einem Satz wie *Peter betrachtet die großen Häuser.*

Die notwendigen Informationen werden Wort für Wort ermittelt, in ein entsprechendes Übergabeformat umgewandelt und die entsprechenden morphologischen Realisierungen anschließend von Morphix generiert.

Vor dem nächsten Verarbeitungsschritt werden die generierten Sätze noch der statistischen Semantikkontrolle (5.5.3) unterzogen.

5.5.3 Statistische Semantikkontrolle

Die von SCall generierten Strukturen werden im Zuge der Semantikkontrolle in Bigramme unterteilt und diese werden wiederum mittels der in 4.3 erklärten Methode analysiert. Die Semantikkontrolle in SCall beschränkt sich auf jene Bigramme, die aus Adjektiv und Nomen bestehen. Wird ein Nomen durch mehrere Adjektive modifiziert, würde also, da nur Bigramme analysiert werden, nur das dem Nomen am

nächsten stehende Adjektiv auf semantische Richtigkeit getestet werden. Aus diesem Grund wird in derartigen Fällen pro Adjektiv ein Bigramm, bestehend aus eben dem Adjektiv und dem Nomen, generiert. Die Phrase *ein großer weißer Hai* würde demnach in die Bigramme *großer Hai* und *weißer Hai* zerlegt werden. Somit wird die Kontrolle sämtlicher Adjektive sichergestellt.

Die während der beschriebenen Prozedur generierten Bigramme werden von der Semantikkontrolle analysiert. Liegt die Anzahl der Treffer für die exakte Phrasensuche unter dem Schwellenwert, wird die entsprechende Phrase als semantisch nicht korrekt markiert und unter Verwendung von anderen Lexikoneinträgen neu generiert. Übersteigt die Anzahl der Treffer für die exakte Phrasensuche jedoch die Schwelle von 1000, wird die Phrase, unabhängig von der Größe des Schwellenwertes, als korrekt markiert, da davon ausgegangen werden kann, dass die Trefferanzahl hoch genug ist, um die Phrase als semantisch korrekt zu kategorisieren.

5.5.4 Generierung der Übungsbeispiele

Da neben den während des Generierungsprozesses erstellten Satz-Strings auf sämtliche lexikalische, semantische sowie syntaktische Information zurückgegriffen werden kann, besteht die Möglichkeit der Produktion eines breit gefächerten Spektrums an unterschiedlichen Übungsbeispielen. Für SCall wurde u.a. die Generierung von Lückentexten zur Übung der Tempusformen von Verben realisiert. Neben den Übungssätzen werden zusätzlich eine Lexikon-Liste mit den in den jeweiligen Sätzen verwendeten Wörtern sowie die Lösung der Übung bereitgestellt. Der Output einer (in diesem Fall auf einen Satz beschränkten) von SCall generierten Übung wird in (10) gezeigt.

(10) Setzen Sie die richtigen Formen der Verben ein.
1. Peter _____ in dem gelben Haus .
Verb: wohnen; Tempus: praesens.

wohnen = bo
Peter = Peter
in = i
gelb = gul
Haus = hus

Loesung der Uebung:

1. Peter wohnt in dem gelben Haus.

Im Zuge der Erstellung dieser Übungsaufgaben werden sämtliche Verben (finite als auch infinite) im Satzgefüge durch Linien ersetzt. Die Informationen darüber, welches Verb bzw. welcher Tempus eingesetzt werden soll, sowie das Lexikon werden aus den während der Generierung der Sätze erzeugten Daten ermittelt. Die Übungsbeispiele werden anschließend durch Überschriften und Erklärungen ergänzt, benutzergerecht strukturiert und in einer zum Ausdruck vorgesehenen Datei gespeichert.

Eine weitere für SCall implementierte Übungsart unterstützt das Training der Adjektiv-Nomen-Kongruenz für Schwedisch (Bsp. 11).

(11) Setzen Sie die richtige Form der Adjektive bzw. der Nomen ein.

1. en klok _____ (granne)

klok = klug
granne = Nachbar

2. den nya _____ (bil)

ny = neu
bil = Auto

3. två _____ böcker (gammal)

gammal = alt
bok = Buch

Lösungen:
1. en klok granne (singular, utrum, unbestimmt)
2. den nya bilen (singular, utrum, bestimmt)
3. två gamla böcker (plural, utrum, unbestimmt)

5.6 Evaluierung

5.6.1 Semantische und grammatische Korrektheit der generierten Strukturen

Die Evaluierung von SCall bestand aus der Kontrolle der semantischen bzw. syntaktischen Korrektheit der von SCall produzierten Sätze. Das den Evaluierungssätzen zugrundeliegende Ausgangslexikon wurde erstellt, indem 25 relativ willkürlich gewählte Satzpaare in SCall eingegeben und analysiert wurden. Die Lexikoneinträge wurden in SCall mit Selektionsrestriktionen annotiert, wobei angemerkt werden muss, dass die Selektionsrestriktionen prinzipiell eher übergeneralisiert wurden. Das so entstandene Lexikon (sh. Anhang A) wurde durch einige Adjektive ergänzt, um abwechslungsreiche Generierungen zu ermöglichen. Das endgültige (Test-)Lexikon verfügt über 101 Einträge. Ausgehend von diesem Lexikon wurden 480 Sätze gene-

Symbol	Beurteilung der Syntax	Beurteilung der Semantik
*	ungrammatisch	semantisch falsch
?	zweifelhaft	zweifelhaft
ok	korrekt	korrekt

Tabelle 5.7: Skala für die Beurteilung der Syntax bzw. der Semantik der Testsätze

Symbol	Erklärung	Anzahl (absolut)	prozentualer Anteil
*	ungrammatisch	16	3,3 %
?	zweifelhaft	36	7,5 %
ok	korrekt	428	89,2 %

Tabelle 5.8: Auswertung der Grammatikalitätsurteile für die Syntax

riert, die, in kleinere Tranchen (20-50 Sätze) zerlegt, von 14 Testpersonen beurteilt wurden. Beurteilt wurden sowohl die Syntax als auch die Semantik der Sätze, wobei jeweils drei verschiedene Beurteilungsmöglichkeiten zur Verfügung standen (sh. Tabelle 5.7).

5.6.1.1 Ergebnisse

5.6.1.1.1 Syntax und Semantik Als ungrammatisch bzw. syntaktisch falsch wurden von den Testpersonen 16 Sätze beurteilt, als syntaktisch zweifelhaft 36 Sätze. Die als syntaktisch falsch beurteilten Sätze (insgesamt 16) sind ausschließlich auf Annotationsfehler, die während des Eingabeprozesses gemacht wurden, zurückzuführen. Der Genus des Lexikoneintrages für *Torte* wurde beispielsweise fälschlicherweise (vom PoS-Tagger) als MAS (maskulin) markiert, was zur Folge hatte, dass Phrasen wie **einen frischen deutschen Torte* generiert wurden. Als syntaktisch falsch wurden ferner fälschlicherweise Sätze, deren Subjekt nicht im Vorfeld des Satzes (bzw. vor dem finiten Prädikat) stand, markiert. Syntaktisch korrekte Sätze wie *Die Mäuse hatte die Katze getötet* wurden aufgrund der falschen Annahme, dass das (vor dem Verb stehende) Subjekt nicht mit dem Verb kongruiert (in diesem Fall *die Mäuse* als Subjekt mit *hatte*), als syntaktisch falsch interpretiert. Als *semantisch* falsch wurden insgesamt 38, als semantisch zweifelhaft 122 Sätze beurteilt. Die detaillierten Ergebnisse sind in den Tabellen 5.8 und 5.9 enthalten.

Die generierten Sätze wurden in weiterer Folge der statistischen Semantikkontrolle unterzogen, wobei jene Sätze, die Konstruktionen enthielten, welche den Schwellenwert nicht erreichten, als falsch markiert wurden. Von den 480 generierten und beurteilten Sätze wurden 120 von der statistischen Semantikkontrolle abgelehnt, das entspricht einem Anteil von 25 Prozent. Der auf den ersten Blick hohe Anteil

Symbol	Erklärung	Anzahl (absolut)	prozentualer Anteil
*	semantisch falsch	38	7,9 %
?	zweifelhaft	122	25,4 %
ok	korrekt	320	66,7 %

Tabelle 5.9: Auswertung der Grammatikalitätsurteile für die Semantik

Symbol	Erklärung	Anzahl (absolut)	prozentualer Anteil
*	ungrammatisch	6	1,7 %
?	zweifelhaft	28	7,8 %
ok	korrekt	326	90,5 %

Tabelle 5.10: Auswertung der Grammatikalitätsurteile für die Syntax nach der statistischen Semantikkontrolle

ist in diesem Fall jedoch in erster Linie auf die oftmalige Verwendung des Adjektivs *riesig* in Verbindung mit Menschen wie in *der riesige Mann* oder *die riesigen Kinder* zurückzuführen. Derartige Bigramme konnten den Schwellenwert in der Regel nicht überschreiten und wurden daher abgelehnt. Die entsprechenden Sätze wurden jedoch auch von den Testpersonen in den meisten Fällen zumindest als *semantisch zweifelhaft* beurteilt. Die Verwendung eines in den konkreten Fällen günstigeren Adjektivs anstelle von *riesig*, z.B. *nett* oder *groß*, hätte die Zahl der abgelehnten Sätze mit großer Wahrscheinlichkeit erheblich verringert.

Berücksichtigt man nur jene 360 Sätze, die von der statistischen Semantikkontrolle nicht abgelehnt wurden, steigt die Anzahl der als korrekt beurteilten Sätze dementsprechend an. Die Anzahl der als *semantisch falsch* eingestuften Beispiele sinkt auf 3,9 % gegenüber den ursprünglichen 7,9 %. Von den 38 als semantisch falsch beurteilten Sätzen wurden immerhin 20 von der Semantikkontrolle abgelehnt (53 %), von den 122 als semantisch zweifelhaft eingestuften Sätzen 60 (49 %). Die als syntaktisch inkorrekt bzw. zweifelhaft beurteilten Konstruktionen waren von der statistischen Semantikkontrolle nur indirekt betroffen, indem syntaktisch inkorrekte Sätze als semantisch inkorrekt markiert und somit nicht weiter behandelt wurden.

Symbol	Erklärung	Anzahl (absolut)	prozentualer Anteil
*	semantisch falsch	14	3,9 %
?	zweifelhaft	64	17,8 %
ok	korrekt	282	78,3 %

Tabelle 5.11: Auswertung der Grammatikalitätsurteile für die Semantik nach der statistischen Semantikkontrolle

5.6.2 Fazit

Die für die Evaluierung generierten Sätze wurden aus einem aus etwa 100 Einträgen bestehenden Lexikon generiert, womit die Kombinationsmöglichkeiten (u.a. beeinflusst durch die Selektionsrestriktionen) letztendlich doch sehr eingeschränkt wurden, was sich im Ergebnis entsprechend widerspiegelt. Geht man von einem idealen Benutzer aus, das heißt, einem Benutzer, dem während der Eingabe- und Analysephase keine Fehler wie falsche Genuszuweisungen und dergleichen unterlaufen, sind syntaktische Fehler in den generierten Sätzen praktisch ausgeschlossen. Für die Semantik trifft dies jedoch nur zum Teil zu, vor allem unter der Berücksichtigung der Tatsache, dass die Beurteilungskriterien für die Richtigkeit von Semantik undeutlicher sind als jene für die Syntax. Satz (12) wurde beispielsweise von verschiedenen Testpersonen sowohl als *semantisch korrekt* als auch als *zweifelhaft* und *unkorrekt* beurteilt, während über die Korrektheit der Syntax kein Zweifel bestand.

(12) Die Nachbarn hatten diesen Hunden irgendeinen frischen Krebs gegeben.

Da Hunde *in der Regel* keine Krebse fressen, kann der Satz als semantisch unkorrekt gesehen werden. Andererseits ist es möglich, Hunden Krebse zu geben, um zu testen, ob sie die Krebse eventuell doch fressen. Somit wäre der Satz als semantisch korrekt einzustufen. Das Ziel einer Anwendung wie SCall sollte also auch darin liegen, nicht bloß semantisch durchaus korrekte Sätze wie (12) zu generieren, sondern vielmehr nur Sätze, die von der Mehrzahl der Benutzer als semantisch korrekt wahrgenommen werden. Dies trifft für das gezeigte Beispiele eben wahrscheinlich nicht zu.

Durch die Bearbeitung der Sätze durch die statistische Semantikkontrolle konnte der Anteil genau jener als zumindest eigenartig beurteilten Sätze entscheidend verringert werden, ein Indiz, das für den Einsatz dieser Methode spricht. Hier muss jedoch angemerkt werden, dass der für die statistische Semantikkontrolle berechnete Schwellenwert tendenziell zu hoch ist, was zur Folge hat, dass viele ganz offensichtlich korrekte Konstruktionen wie (13) als falsch markiert werden (Grund für die Ablehnung dieses Satzes war die zu geringe Trefferanzahl des Bigramms *freundlicher Vater* (134 Treffer bei einer Schwelle von mindestens 3711 Treffern)).

(13) Ein freundlicher Vater liest ein Buch.

Somit kann festgehalten werden, dass die von SCall generierten Sätze trotz der Verwendung der Selektionsrestriktionen semantisch zu vage sind und in weiterer Folge nicht für den Einsatz in den entsprechenden Übungen geeignet sind. Erst durch die Behandlung durch die statistische Semantikkontrolle kann ein zufriedenstellendes Ergebnis hinsichtlich semantischer Akzeptanz erreicht werden.

Kapitel 6

Resumé

6.1 Zusammenfassung

Ziel dieser Arbeit war die Entwicklung eines CALL-Programms für die Verwendung im Schwedischunterricht für deutschsprachige Schüler und vice versa. Das Hauptaugenmerk wurde auf eine geeignete Methode zur semantischen Annotation der verwendeten Lexika gelegt, um in weiterer Folge die Generierung von semantisch korrekten Übungssätzen sicherzustellen.

Im theoretischen Teil der Arbeit wurden drei Themenbereiche, die als Grundlage für die praktische Umsetzung der SCall genannten Anwendung dienen, behandelt. Im ersten Kapitel wurde die Geschichte des *Computer Assisted Language Learning* (CALL) von den ersten Versuchen in den sechziger Jahren bis in die Gegenwart beleuchtet. In weiterer Folge wurde die Rolle der Computerlinguistik im Bereich des CALL diskutiert und der Frage nachgegangen, warum der Anteil von NLP-Methoden innerhalb des CALL relativ bescheiden ist. Als ein Grund dafür können Kommunikationsdefizite, die zwischen den unterschiedlichen an CALL beteiligten Disziplinen vorhanden sind, genannt werden. Als weiterer und wohl wichtigster Grund für den geringen Anteil an NLP-Methoden wird die teilweise schlechte Ausgereiftheit dieser Methoden gesehen, welcher verhindert, dass diese verlässlich in CALL-Anwendungen eingesetzt werden können.

Im zweiten Kapitel wurde der Ansatz der semantischen Annotation des Lexikons in Form von Selektionsrestriktionen diskutiert. Durch die Selektionsrestriktionen wird die Verwendung der entsprechenden Lexikoneinträge auf bestimmte semantische Umgebungen eingeschränkt, womit die Generierung von semantisch korrekten Konstruktionen sichergestellt werden soll. Um ein möglichst breites Spektrum an unterschiedlichen Selektionsrestriktionen zu erhalten, wurden die Konzepte einer geeigneten Ontologie als Grundlage für die Selektionsrestriktionen gewählt. Als zu-

grundeliegende Ontologie wurde die WordNet-Ontologie ausgesucht, wobei jedem Konzept in der Ontologie eine Selektionsrestriktion in SCall entspricht.

Im dritten Kapitel wurde eine Methode zur statistischen Semantikkontrolle vorgestellt. Die statistische Semantikkontrolle beurteilt die semantische Richtigkeit von Bigrammen, indem die Einzelhäufigkeiten der Bestandteile im zugrundeliegenden Korpus gesucht und von den ermittelten Werten ausgehend ein Schwellenwert errechnet wird. Übersteigt die Suche des Bigramms (im exakten Wortlaut) diesen Wert, wird es als semantisch korrekt gewertet. Da die Semantikkontrolle in SCall das World Wide Web als Korpus benutzt, wurde der Frage nachgegangen, inwieweit das WWW überhaupt als linguistisches Korpus gesehen werden kann. Nimmt man die Größe bzw. die Menge des abrufbaren Textes als Maßstab, übertrifft das WWW die größten speziell für linguistische Zwecke entwickelten Korpora um ein vielfaches und stellt somit eine praktische Ressource für Tasks, die möglichst große Textmengen benötigen, dar. Ferner ist das WWW frei zugänglich und über weite Strecken sprachenunabhängig, d.h., dass Textmaterial in den meisten (verschrifteten) Sprachen gefunden werden kann. Die Nachteile des WWW als Korpus liegen unter anderem in der großen Menge an Störfaktoren, welche die Verwendung negativ beeinflussen. Zu diesen Faktoren sind ungrammatischer Text, Metainformation, Menütexte etc. zu zählen. Weiters sind die Such- und Bearbeitungsmöglichkeiten limitiert, da die gängigen Suchmaschinen nur ein sehr beschränktes Spektrum an Werkzeugen zur Verfügung stellen. Das WWW kann folglich durchaus als linguistisches Korpus gesehen und verwendet werden, wobei die Verwendungstauglichkeit je nach Einsatzgebiet variiert. Auf jeden Fall müssen jedoch die offensichtlichen Einschränkungen berücksichtigt werden.

Im praktischen Teil der Arbeit wurde der Aufbau sowie die Funktion von SCall erklärt. Ausgehend von vom Benutzer eingegebenen Satzpaaren sowie ebenfalls vom Benutzer aus von SCall ermittelten Vorschlägen ausgewählten Selektionsrestriktionen werden Sätze generiert, die als Ausgangsmaterial für Grammatikübungen im Fremdsprachenunterricht dienen. Die generierten Sätze werden in weiterer Folge der statistischen Semantikkontrolle unterzogen, um semantisch unkorrekte Konstruktionen, die trotz des Vorhandenseins der Selektionsrestriktionen generiert wurden, auszuschließen.

6.2 Testergebnisse

Die Umsetzung von SCall stellte einen Versuch dar, ein CALL-Programm mit einem möglichst hohen Anteil an NLP-Methoden zu realisieren. Eine stabile Umsetzung scheiterte am nur teilweise zufriedenstellenden Präzisionsgrad der verwendeten Me-

thoden. Eine entscheidende Schwachstelle in SCall stellen u.a. die gewählten Parser für Deutsch dar. Weder der im TreeTagger-Paket enthaltene Chunk-Parser noch der Stanford-Parser konnten die erwünschten, benutzerfreundlichen Ergebnisse liefern. Zu hoch war die Fehlerrate in den von den Parsern produzierten Outputs. Die daraus resultierende Nachbearbeitung wäre einem Benutzer in der Praxis wahrscheinlich nicht zumutbar. Auch das Morphologiewerkzeug für Deutsch (Morphix) zeigte Schwächen, die den Einsatz von entscheidend SCall einschränken, da, sobald dem Programm unbekannte Wörter übergeben werden, kein Output produziert wird. Einer der bereits erwähnten Hauptgründe für den geringen Einsatz von NLP im Bereich des CALL, nämlich jener der mangelhaften Ausgereiftheit sprachtechnologischer Methoden, bestätigte sich somit zum Teil auch für SCall. Das dem Analyseprozess in SCall zugrundeliegende Konzept müsste folglich grundsätzlich überarbeitet und Alternativen zu den offensichtlichen Schwächen der Anwendung gefunden werden.

Die in dieser Arbeit vorgeschlagene Methode der semantischen Annotation von Lexikoneinträgen durch Selektionsrestriktionen zur Verwendung in CALL-Anwendungen scheint nur zum Teil zu einem befriedigenden Ergebnis zu führen. Obwohl die zugrundeliegende Ontologie aus WordNet detailliert ist und präzise Differenzierungen zulässt, ist die semantische Information scheinbar nicht ausreichend, um die erwünschten Ergebnisse zu produzieren.

Für die statistische Semantikkontrolle konnte in einer Reihe von Probeläufen zufriedenstellende Ergebnisse erreicht werden, wobei jedoch festgestellt wurde, dass der errechnete Schwellenwert in der Regel etwas zu hoch angesetzt ist.

Durch die Kombination der beiden Methoden konnten jedoch durchaus annehmbare Ergebnisse erzielt werden.

6.3 Perspektiven

Trotz der erwähnten Probleme gilt für die in SCall verwendeten semantischen Ansätze, sowohl für die Annotation des Lexikons mit Selektionsrestriktionen als auch für die statistische Semantikkontrolle, dass der Einsatz in anderen sprachtechnologischen Anwendungen durchaus denkbar ist. Beide Methoden müssten jedoch gegebenenfalls optimiert werden. Im Fall der Selektionsrestriktionen würden sich alternative Ontologien bzw. Taxonomien als Grundlage anbieten. Die statistische Semantikkontrolle könnte durch die Optimierung der zugrundeliegenden statistischen Methoden zur Errechnung des Schwellenwertes zu einem leistungsstarken Semantikwerkzeug ausgebaut werden.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Das für die Generierung der Evaluierungsbeispiele verwendete Lexikon.
Die Selektionsrestriktionen werden aus Platzgründen verkürzt dargestellt.
Anmerkung: ä = ae, ö = oe, ü = ue.

Nomen

Apfel	NOUN	=>food,solidfood	NTR	äpple	NCNSN@IS
Auto	NOUN	=>vehicle	NTR	bil	NCUSN@IS
Brief	NOUN	=>writing,writtenmaterial,...	MAS	brev	NCNSN@IS
Brief	NOUN	=>writtencommunication,...	MAS	brev	NCNSN@IS
Brot	NOUN	=>food,solidfood	NTR	bröd	NCNSN@IS
Bruder	NOUN	=>person,individual,...	MAS	bror	NCUSN@IS
Buch	NOUN	=>product,production	NTR	bok	NCUSN@IS
Buch	NOUN	=>writtencommunication,...	NTR	bok	NCUSN@IS
Fahrrad	NOUN	=>artifact,artefact	NTR	cykel	NCUSN@IS
Fahrrad	NOUN	=>vehicle	NTR	cykel	NCUSN@IS
Fahrzeug	NOUN	=>vehicle	NTR	fordon	NCNSN@IS
Fisch	NOUN	=>food,solidfood	NTR	fisk	NCUSN@IS
Freund	NOUN	=>person,individual,...	MAS	kompis	NCUSN@IS
Hund	NOUN	=>mammal,mammalian	MAS	hund	NCUSN@IS
Idee	NOUN	=>idea,thought	FEM	idé	NCUSN@IS
Katze	NOUN	=>mammal,mammalian	FEM	katt	NCUSN@IS
Kind	NOUN	=>person,individual,...	NTR	barn	NCNSN@IS
Knochen	NOUN	=>animalmaterial	MAS	ben	NCNSN@IS
Krebs	NOUN	=>food,solidfood	MAS	räka	NCUSN@IS
Mann	NOUN	=>person,individual,...	MAS	man	NCUSN@IS
Maus	NOUN	=>placental,...	FEM	mus	NCUSN@IS
Mensch	NOUN	=>person,individual,...	MAS	människa	NCUSN@IS
Nachbar	NOUN	=>person,individual,...	MAS	granne	NCUSN@IS
Schwester	NOUN	=>person,individual,...	FEM	syster	NCUSN@IS
Techniker	NOUN	=>person,individual,...	MAS	tekniker	NCUSN@IS
Tisch	NOUN	=>artifact,artefact	MAS	bord	NCNSN@IS
Torte	NOUN	=>food,solidfood	FEM	tårta	NCUSN@IS
Vater	NOUN	=>ancestor,ascendant,...	MAS	far	NCUSN@IS
Vater	NOUN	=>relative,relation	MAS	far	NCUSN@IS
Wagen	NOUN	=>vehicle	MAS	bil	NCUSN@IS
Zeitung	NOUN	=>artifact,artefact	FEM	tidning	
	NCUSN@IS				
Zeitung	NOUN	=>writtencommunication,...	FEM	tidning	
	NCUSN@IS				

Proper Nouns / Eigennamen

Hans	PNOUN	=>person,individual,...	—	Hans
Maria	PNOUN	=>person,individual,...	—	Maria
Oslo	PNOUN	=>urbanarea,populatedarea	—	Oslo
Peter	PNOUN	=>person,individual,...	—	Peter
Stockholm	PNOUN	=>urbanarea,populatedarea	—	Stockholm

Adjektive

alt	ADJ	=>person,individual,...	eva	gammal
alt	ADJ	=>writtencommunication,...	eva	gammal
boese	ADJ	=>person,individual,...	dim	ond
braun	ADJ	=>artifact,artefact	col	brun
braun	ADJ	=>mammal,mammalian	col	brun
deutsch	ADJ	=>food,solidfood	nat	tysk
freundlich	ADJ	=>person,individual,...	eva	snäll
frisch	ADJ	=>food,solidfood	eva	—
frisch	ADJ	=>food,solidfood	eva	färsk
gruen	ADJ	=>structure,construction	eva	groen
gut	ADJ	=>idea,thought	eva	god
gut	ADJ	=>product,production	eva	bra
italienisch	ADJ	=>person,individual,...	nat	italiensk
italienisch	ADJ	=>ancestor,ascendant,...	nat	italiensk
klein	ADJ	=>mammal,mammalian	dim	liten
klein	ADJ	=>person,individual,...	dim	liten
lang	ADJ	=>writing,writtenmaterial,...	eva	lång
lustig	ADJ	=>person,individual,...	eva	rolig
nett	ADJ	=>writing,writtenmaterial,...	dim	god
neu	ADJ	=>artifact,artefact	eva	ny
neu	ADJ	=>idea,thought	eva	ny
neu	ADJ	=>product,production	eva	ny
neu	ADJ	=>vehicle	eva	ny
riesig	ADJ	=>person,individual,...	dim	jättestor
rot	ADJ	=>product,production	col	röd
rot	ADJ	=>vehicle col	röd	
schnell	ADJ	=>vehicle dim	snabb	
schoen	ADJ	=>person,individual,...	eva	vacker
schwarz	ADJ	=>mammal,mammalian	col	svart

Verben

entwickeln	VERB	=>person,individual,...	—	=>vehicle	
	—	—	utveckla		
essen	VERB	=>person,individual,...	—	=>food,solidfood	
	—	—	äta		
geben	VERB	=>person,individual,...	=>mammal,mammalian		
	=>food,solidfood		—	—	giva
geben	VERB	=>person,individual,...	=>person,individual,...		
	=>food,solidfood		—	—	ge
geben	VERB	=>person,individual,...	=>person,individual,...		
	=>vehicle	—	—	ge	
fressen	VERB	=>mammal,mammalian	—	=>placental,...	
	—	—	äta		
haben	VERB	=>person,individual,...	—	=>idea,thought	
	—	—	ha		
haben	VERB	=>person,individual,...	—	=>vehicle	
	—	—	ha		
kaufen	VERB	=>person,individual,...	—	=>artifact,artefact	
	—	—	köpa		
	—	—	köpa		
kaufen	VERB	=>person,individual,...	—	=>food,solidfood	
	—	in-=>urbanarea,populatedarea-dat	köpa		
kaufen	VERB	=>person,individual,...	—	=>food,solidfood	
	—	—	köpa		
kaufen	VERB	=>person,individual,...	—		
	=>product,production		—	—	
	köpa				
kaufen	VERB	=>person,individual,...	—	=>vehicle	
kommen	VERB	=>person,individual,...	—	—	—
	—	komma			
lesen	VERB	=>person,individual,...	—	—	—
	—	läsa			
lesen	VERB	=>person,individual,...	—		
	=>writtencommunication,...		—		
	—	läsa			
lesen	VERB	=>person,individual,...	—		
	=>writtencommunication,...		—		
	von-=>relative,relation-dat		läsa		
schreiben	VERB	=>person,individual,...	—		
	=>writing,writtenmaterial,...		—		
	an-=>ancestor,ascendant,...-dat		skriva		
schreiben	VERB	=>person,individual,...	—		
	=>writing,writtenmaterial,...		—	—	
	skriva				
sehen	VERB	=>person,individual,...	—		
	=>person,individual,...		—	—	
	se				
umbringen	VERB	=>person,individual,...	—		
	=>person,individual,...		—	—	
	ta livet av				

Sonstige Wortarten

an	PRAEP	=>ancestor,ascendant,...	dat	till	
aus	PRAEP	=>urbanarea,populatedarea	dat	från	
dies	DET	x	x	x	x
ein	DETID	x	x	x	x
er	PRON	=>person,individual,...	—	han	
ich	PRON	=>person,individual,...	—	jag	
in	PRAEP	=>urbanarea,populatedarea	dat	i	
irgendein	DETID	x	x	x	x
jen	DET	x	x	x	x
nil	DETDEF	x	x	x	x
sie	PRON	=>person,individual,...	—	hon	
von	PRAEP	=>relative,relation	dat	från	
wir	PRON	=>person,individual,...	—	vi	

Anhang B

Auswahl von 50 der 480 von SCall für Auswertungszwecke produzierten und beurteilten Sätze. Die in kursiv geschriebenen Sätze wurden von der statistischen Semantikkontrolle abgelehnt.
Anmerkung: ä = ae, ö = oe, ü = ue.

Hans hat dieses neue Auto gekauft.
Peter hat das neue Auto gekauft.
Peter hatte Autos gekauft.
Dieser freundliche Nachbar wird irgendeinem alten Menschen ein altes Auto geben.
Diese Maenner werden neue Autos kaufen.
Die Techniker kauften irgendein Buch.
Die Nachbarn hatten irgendein neues Fahrzeug gekauft.
Diesen riesigen Maennern hatten diese Kinder irgendein neues Fahrzeug gegeben.
Neue Autos entwickelt Hans.
Die Schwestern lesen ein altes Buch.
Die alten kleinen Freunde lesen diese Briefe.
Hans bringt ein freundliches Kind um.
Dieser kluge Vater kauft alte Fahrzeuge.
Kinder werden diesen frischen Torte kaufen.
Das alte Buch werden diese Nachbarn gelesen haben.
Irgendein netter Vater hatte Ideen.
Die Vaeter hatten eine gute Idee.
Der freundliche Mensch kam.
Ein altes Kind wird ein neues Fahrzeug gekauft haben.
Peter hatte die umgebracht.
Die Schwestern werden die Autos entwickeln.
Freunde kaufen diese frischen Brote.
Dem schoenen kleinen Vater gab Hans dieses neue Fahrrad.
Irgendein Techniker hat gelesen.
Maenner hatten irgendeinem kleinen Kind dieses frische Brot gegeben.
Diesen Kindern hat Peter ein frisches Brot gegeben.
Diese neuen Autos gaben Maenner Maennern.
Hans wird die Krebse kaufen.

Diese Brueder kauften irgendein neues Auto.
Vaeter hatten einen Brief an die Vaeter geschrieben.
Hans liest ein altes Buch von den Vaetern.
Vaeter werden diese gute Idee gehabt haben.
Buecher werden diese Nachbarn gelesen haben.
Hans kauft rote Autos.
Eine riesige Katze fraß die kleinen Maeuse.
Ein alter Vater wird neue rote Autos gehabt haben.
Die Maenner werden irgendeinen Brief geschrieben haben.
Hans wird einer kleinen Katze die Krebse geben.
Irgendein alter Mensch liest.
Die Maenner entwickeln die roten Fahrraeder.
Diese Kinder werden dieses neue Auto kaufen.
Diese riesige Schwester kauft diese neuen Buecher.
Die klugen kleinen Kinder geben den Nachbarn Krebse.
Diese klugen Maenner essen die frischen Brote.
Hans hat Briefe an irgendeinen klugen kleinen Vater geschrieben.
Maeuse werden Katzen gefressen haben.
Irgendein Mensch hatte diesen Schwestern Krebse gegeben.
Schwestern werden gekommen sein.
Der freundliche kleine Nachbar hatte diesem kleinen Vater den frischen Krebs gegeben.
An den italienischen Vater wird Hans einen Brief schreiben.

Abstract

CALL (*Computer Assisted Language Learning*) ist ein interdisziplinäres Feld, an dem neben zahlreichen anderen Disziplinen auch die Computerlinguistik partizipiert. Tatsächlich ist jedoch der Anteil an CALL-Programmen mit NLP-Hintergrund verschwindend gering. Als ein Hauptgrund für dafür wird die Tatsache, dass die meisten NLP-Methoden nicht ausgereift genug sind, um in CALL-Anwendungen vernünftig eingesetzt werden zu können, genannt.

Diese Arbeit stellt den Versuch dar, ein CALL-Programm zu entwickeln, das von einer möglichst hohen Anzahl an unterschiedlichen Ansätzen aus dem NLP Gebrauch macht, wobei der Schwerpunkt auf semantischer Annotation liegt. Die *SCall* (kurz für Semantisches CALL) genannte Anwendung generiert ausgehend von benutzerseitig eingegebenen Sätzen Übungsaufgaben zur Verwendung im Fremdsprachenunterricht, in diesem Fall für Deutsch und Schwedisch.

Im theoretischen Teil der Arbeit werden jene Bereiche, die als Grundlage für die praktische Umsetzung von SCall dienen, behandelt. Der Schwerpunkt der Arbeit liegt auf der Annotation des Lexikons mit semantischer Information, sogenannten Selektionsrestriktionen. Selektionsrestriktionen sind semantische Markierungen, welche die Verwendung von Wörtern auf bestimmte semantische Umgebungen einschränken, wodurch die Generierung von semantisch korrekten Konstruktionen sichergestellt werden soll. Als Basis für die Selektionsrestriktionen in SCall dient die der lexikalischen Ressource *WordNet* zugrundeliegende Ontologie, wobei jedes Konzept einer Selektionsrestriktion entspricht.

In weiterer Folge wird ein Ansatz zur statistischen Kontrolle von semantischen Konstruktionen vorgestellt, wobei das World Wide Web als linguistisches Korpus Verwendung findet. Mittels dieser Methode werden Bigramme auf deren semantische Richtigkeit kontrolliert. Im zugrundeliegenden Korpus werden die Einzelhäufigkeiten der Bestandteile der Bigramme festgestellt, davon ausgehend wird ein Schwellenwert für die statistisch zu erwartende Häufigkeit des Bigramms im exakten Wortlaut ermittelt. Anschließend wird die Häufigkeit des Bigramms im entsprechenden Korpus ermittelt; ist die Anzahl höher als der Schwellenwert, wird das Bigramm als semantisch korrekt eingestuft.

Im praktischen Teil der Arbeit werden der Aufbau sowie die Funktion von SCall erläutert. Ausgehend von vom Benutzer eingegebenen Satzpaaren sowie ebenfalls

benutzerseitig aus einer Reihe von vom Programm ermittelten Vorschlägen ausgewählten Selektionsrestriktionen werden Sätze generiert, die als Ausgangsmaterial für Grammatikübungen dienen. Die generierten Sätze werden in weiterer Folge der erwähnten statistischen Semantikkontrolle unterzogen, um die Verwendung von semantisch unkorrekten Konstruktionen, die trotz des Vorhandenseins der Selektionsrestriktionen generiert wurden, auszuschließen.

Lebenslauf

Georg Pitschmann, geb. 3. Dezember 1979

10.2001 - 11.2008	Studium an der Universität Wien
01.2004 - 08.2004	Auslandssemester an der Universität *Umeå Universitet*, Schweden (Erasmus-Stipendium)
08.2006 - 01.2007	Auslandssemester an der Universität *Copenhagen Business School*, Dänemark (CIRIUS-Stipendium)

Literatur

Aycock, John (1998): Compiling little languages in Python. *In: Proceedings of the 7th International Python Conference, November 10-13 1998, Houston, Texas.* S. 67-77.
oder:
URL: http://pages.cpsc.ucalgary.ca/~aycock/spark/paper.pdf [20.05.2008]

Bateman, John A. / Henschel, Renate / Rinaldi, Fabio (1995): *The Generalized Upper Model 2.0.* Technical report. Darmstadt: GMD/Institut für Integrierte Publikations- und Informationssysteme.

Bateman, John / Zock, Michael (2003): Natural Language Generation. *In:* Mitkov, Ruslan (Hg.): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. 284-304.

Bickel, Hans (2006): Das Internet als linguistisches Korpus. *Linguistik Online* 28(3). 71-83.

Brammerts, Helmut / Little, David (Hgg.) (1996): *Leitfaden für das Sprachenlernen im Tandem über das Internet.* Bochum: Brockmeyer.

Bruce, Rebecca / Guthrie, Louise (1992): Genus Disambiguation: A Study in Weighted Preference. *In: Proceedings of the 14th International Conference on Computational Linguistics. 23.-28. August 1992 Nantes.* New York, NY: ACM Digital Library. 1187-91.

Carroll, John (2003): Parsing. *In:* Mitkov, Ruslan (Hg.): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. S. 233-48.

Chapelle, Carol A. (2001): *Computer Applications in Second Language Acquisition. Foundations for teaching, testing and research.* Cambridge, UK: Cambridge University Press.

Chomsky, Noam (1965): *Aspects of the Theory of Syntax.* Cambridge, Mass.: MIT Press.

Duffner, Rolf / Näf, Anton (2006): Digitale Textdatenbanken im Vergleich. *Linguistik Online* 28(3). 7-22.

Dutoit, Thierry / Stylianou, Yannis (2003): Text-to-Speech Synthesis. *In:* Mitkov, Ruslan (Hg.): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. 323-338.

Ehsani, Farzad / Knodt, Eva (1998): Speech Technology in Computer-Assisted Language Learning: Strengths and Limitations of a New CALL Paradigm. *Language Learning and Technology*, 2(1). 54-73.

Fellbaum, Christine (1998): A Semantic Network of English Verbs. *In:* Fellbaum, Christine (Hg.): *WordNet: an electronical lexical database.* Cambridge, Mass [u.a.]: MIT Press. 69-104.

Fellbaum, Christine / Miller, George A. (1990): Folk Psychology or Semantic Entailment? *Psychological Review 97 (4).* 565-70.

Felshin, Sue (1995): The Athena Language Learning Project NLP System: A Multilingual System for Conversation-Based Language Learning. *In:* Holland, Melissa V. / Kaplan, Jonathan D. / Sams, Michelle R. (Hgg.): *Intelligent Language Tutors: Theory Shaping Technology.* Mahwah, NJ: Lawrence Erlbaum Associates. 257-72.

Finkler, Wolfgang / Neumann, Günter (1988): Morphix. A Fast Realization of a Classification-Based Approach to Morphology. *In:* Trost, Harald (Hg.): *4. Österreichische Artificial-Intelligence-Tagung. Wiener Workshop - Wissensbasierte Sprachverarbeitung. Proceedings.* Berlin et al.: Springer. S. 11-19.

Fischer Nilsson, Jörgen / Jensen, Per Anker (2006): Ontology-Based Semantics for Prepositions. In: Saint-Dizier, Patrick (Hg.): *Syntax and Semantics of Prepositions.* Dordrecht (NL): Springer. 229-44.

Fletcher, William H. (2001): Concordancing the Web with KWiCFinder. *In: Proceedings of the 3rd North American Symposium on Corpus Linguistics and Language Teaching.*
URL: http://www.kwicfinder.com/FletcherCLLT2001.pdf [18.02.2008]

Glück, Helmut (Hg.) (2000): *Metzler Lexikon Sprache.* 2. Auflage. Stuttgart, Wei-

mar: Metzler.

Gómez-Pérez, Asunción / Fernández-López, Mariano / Corcho, Oscar (2004): *Ontological Engineering*. London: Springer.

Grefenstette, Gregory / Nioche, Julien (2000): Estimation of English and non-English Language Use on the WWW. *In: Proceedings of RIAO'2000: Content-Based Multimedia Information Access, 12.-14. April 2000 Paris*. 237-46.
http://arxiv.org/ftp/cs/papers/0006/0006032.pdf [03.02.2008]

Gruber, Thomas R. (1993): A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2). 199-220.

Gustafson-Capková, Sofia / Hartmann, Britt (2006): *Manual of the Stockholm Umeå Corpus version 2.0*. Stockholm: Stockholms universitet / Institutionen för lingvistik.
URL: http://www.ling.su.se/staff/sofia/suc/manual.pdf [04.02.2008]

Hanks, Patrick (2003): Lexicography. *In:* Mitkov, Ruslan (Hg.): *The Oxford Handbook of Computational Linguistics*. Oxford, UK (et al.): Oxford University Press. 48-69.

Hardisty, David / Windeatt, Scott (1989): *CALL*. Oxford [u.a.] : Oxford University Press.

Hart, R. S. (1995): The Illinois PLATO Foreign Languages Project. *CALICO Journal*, 12(4). 15-37.

Helbig, Gerhard / Buscha, Joachim (1996): *Deutsche Grammatik. Ein Handbuch für den Ausländerunterricht*. Neubearbeitung. 2. Auflage. Berlin (et al.): Langenscheidt.

Kennedy, Greame (1998): *An Introduction to Corpus Linguistics*. London, New York: Longman.

Kilgarriff, Adam / Grefenstette, Gregory (2003): Introduction to the Special Issue on the Web as Corpus. *Computational Linguistics*, 29(3). 1-15.
http://www.kilgarriff.co.uk/Publications/2003-KilgGrefenstette-WACIntro.pdf [02.02.2008]

Kilgarriff, Adam (2006): Googleology is Bad Science. *Computational Linguistics*, 33(1). 147-151.

König, Werner (2001): *dtv-Atlas Deutsche Sprache.* 13. Auflage. München: Deutscher Taschenbuch Verlag.

Kramsch, Claire / Morgenstern, Douglas / Murray, Janet H. (1985): An Overview of the MIT Athena Language Learning Project. *CALICO Journal*, 2(4). S. 31-4.

Lawrence, Steve / Giles, C. Lee (1999): Accessibility of information on the Web. *Nature*, 400. 107-9.

Leech, Geoffrey (1991): The state of the art in corpus linguistics. In: Ajmer, Karin / Altenberg, Bengt (Hgg.): *English Corpus Linguistics: Studies in Honour of Jan Svartvik.* London: Longman. 8-29.

Levy, Michael (1997): *Computer-Assisted Language Learning. Context and Conceptualization.* Oxford, UK: Clarendon Press.

Levy, Michael (1999): Design Processes in CALL: Integrating Theory, Research and Evaluation. In: Cameron, Keith (Hg.): *CALL: Media, Design and Applications.* Lisse, NL (et al.): Swets and Zeitlinger. 83-108.

Mahesh, Kavi (1996): *What is an Ontology?* [online]. Las Cruces: New Mexico State University, Computing Research Laboratory.
URL: http://crl.nmsu.edu/Research/Projects/mikro/htmls/theoretical.tr-htmls/node5.html [12.10.2007]

Mahesh, Kavi / Nirenburg, Sergei (1995a): A Situated Ontology for Practical NLP. In: *Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, 19.-21. August 1995, Montreal.*

Mahesh, Kavi / Nirenburg, Sergei (1995b): Semantic Classification for Practical Natural Language Processing.

Meixner, Uwe (2004): *Einführung in die Ontologie.* Darmstadt: Wissenschaftliche Buchgesellschaft.

McEnery, Tony (2003): Corpus Linguistics. *In:* Mitkov, Ruslan (Hg.): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. 448-463.

Megyesi, Beáta (2002): *Data-Driven Syntactic Analysis - Methods and Applications for Swedish.* Thesis (PhD). Department of Speech, Music and Hearing; Kungliga Tekniska Högskola Stockholm.
URL: http://stp.lingfil.uu.se/%7Ebea/megyesi02-phdthesis-update.pdf
[31.03.2008]

Morgenstern, Douglas / Murray, Janet H. (1995): Tracking the Missing Biologist. *Humanities*, 16(5). 33-38.

Nerbonne, John (2003): Natural Language Processing in Computer-Assisted Language Learning. *In:* Mitkov, Ruslan (Hg.): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. 670-98.

Nerbonne, John / Jager, Sake / van Essen, Arthur (1998): Language Teaching and Language Technology. Introduction. *In:* Nerbonne, John / Jager, Sake / van Essen, Arthur (Hgg.): *Language Teaching and Language Technology.* Lisse, NL: Swets and Zeitlinger. 1-10.

Nistrup Madsen, Bodil / Erdman Thomsen, Hanne / Vikner, Carl (2004): Comparison of Principles Applying to Domain Specific versus General Ontologies. *In:* Alessandro Oltramari, Patrizia Paggio, Aldo Gangemi, Maria Teresa Pazienza, Nicoletta Calzolari, Bolette Sandford Pedersen, Kiril Simov (Hgg.): *Workshop Proceedings: OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments. LREC 04, Lisbon, Portugal.* Paris: ELRA. 90-95.
URL: http://www.id.cbs.dk/∼het/caoslit/CAOS_Ontolex2004.pdf [20.06.2008]

Nugues, Pierre M. (2006): *An Introduction to Language Processing with Perl and Prolog.* Berlin [u.a.] : Springer.

Rath, Ingo (Hg.) (1998): *Aristoteles: Die Kategorien.* Griechisch-Deutsch. Übers. und hrsg. von Ingo W. Rath. Stuttgart: Reclam.

Reiter, Ehud / Venour, Chris (2008): *A Tutorial for Simplenlg (Version 3.6).* Aberdeen: University of Aberdeen, Department of Computing Science.

URL: http://www.csd.abdn.ac.uk/~ereiter/simplenlg/simplenlg-v36.zip
[05.05.2008]

Renouf, Antoinette / Kehoe, Andrew / Banerjee, Jayeeta (2005): The WebCorp Search Engine: A holistic approach to web text search. *In: Proceedings from The Corpus Linguistics Conference Series. Corpus Linguistics 2005 Conference, Birmingham 2005.*
URL: http://www.corpus.bham.ac.uk/PCLC/cl2005-SE-pap-final-050705.doc
[18.02.2008]

Renouf, Antoinette / Kehoe, Andrew / Banerjee, Jayeeta (2006): WebCorp: an integrated system for web text search. *Language and Computers*, 59(1). 47-67.
URL: http://rdues.bcu.ac.uk/publ/WebCorp_integrated_system_DRAFT.pdf
[07.02.2008]

Reuer, Veit (2004): *Language Processing and Intelligent Computer-Assisted Language Learning.* Technical Report. Osnabrück: Institute of Cognitive Science, University of Osnabrück.

Saint-Dizier, Patrick (2006): Introduction to the Syntax and Semantics of Prepositions. In: Saint-Dizier, Patrick (Hg.): *Syntax and Semantics of Prepositions.* Dordrecht (NL): Springer. 1-26.

Salaberry, M. Rafael (1996): A Theoretical Foundation for the Development of Pedagogical Tasks in Computer Mediated Communication. *CALICO Journal* 14(1). 5-34.

Schiller, Anne / Stöckert, Christine / Teufel, Simone (1995): *Guidelines für das Tagging deutscher Textcorpora mit STTS.* Technical Report. Universität Stuttgart: Institut für maschinelle Sprachverarbeitung.

Schmid, Helmut (1994): *Probabilistic Part-of-Speech Tagging Using Decision Trees.* Stuttgart: Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung.
URL: http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf
[01.04.2008]

Schmid, Helmut (1995): *Improvements in Part-of-Speech Tagging with an Application to German.* Stuttgart: Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung.

URL: http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger2.pdf [01.04.2008]

Schröder, Jochen (1990): *Lexikon deutscher Präpositionen.* 2. Auflage. Leipzig: Verlag Enzyklopädie.

Stevenson, Mark / Wilks, Yorick (2001): The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics*, 27(3). 321-49.

Studer, Rudi / Benjamins, V. Richard / Fensel, Dieter (1998): Knowledge Engineering: Principles and Methods. *Data Knowledge Engineering*, 25(1-2). 161-97.

Underwood, John H. (1984): *Linguistics, Computers, and the Language Teacher: A Communicative Approach.* Rowley, Massachusetts: Newbury House Publishers.

Vossen, Piek (1998): Introduction to EuroWordNet. *Computer and the Humanities*, 32. 73-89.

Vossen, Piek (2003): Ontologies. *In:* Mitkov, Ruslan (Hg.) (2003): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. 464-82.

Voutilainen, Atro (2003): Part-of-Speech Tagging. *In:* Mitkov, Ruslan (Hg.) (2003): *The Oxford Handbook of Computational Linguistics.* Oxford, UK (et al.): Oxford University Press. S. 219-32.

Warschauer, Mark (1996): Computer Assisted Language Learning: an Introduction. *In:* Fotos, S. (Hg.): *Multimedia language teaching.* Tokyo: Logos International. 3-20.

Woolley, D.R. (1994). PLATO: The Emergence of On-Line community. *Computer-Mediated Communication Magazine*, 1(3). 5.
URL: http://www.december.com/cmc/mag/1994/jul/plato.html [30.05.2008]

Zock, Michael (1996): Computational Linguistics and its Use in Real World: the Case of Computer Assisted-Language Learning. In: *Proceedings of the 16th conference on Computational linguistics. Copenhagen, Denmark.* Morristown, NJ: Association for Computational Linguistics. 1002-4.