

PROGRAMMIERPROJEKT

„VALENZ UND SEMANTIK“

ARBEITSBERICHT

eingereicht von:

Corinne M. Parigi
Neunbrunnenstrasse 110
8050 Zürich
Tel.: 01 300 30 72
e-Mail: parigi@gmx.net

Studentin im 6. Semester

Hauptfach: Publizistikwissenschaften
1. Nebenfach: Computerlinguistik
2. Nebenfach: Kriminologie

Yvonne A. Müller
Obere Büehlhalde 5
5634 Merenschwand
Tel.: 079 359 72 52
e-Mail: yvonna.mueller@gmx.ch

Studentin im 8. Semester

Hauptfach: Soziologie
1. Nebenfach: Computerlinguistik
2. Nebenfach: Kriminologie

Thema und Zeitpunkt
der Lehrveranstaltung:

Programmierprojekt am TIK
SS 2005

Betreuer: Tobias Kaufmann

Zürich, 18. Juli 2005

Inhaltsverzeichnis

1	Antivergesslichkeitsprogramme?	3
1.1	Aufgabenstellung	3
1.2	Aufbau der Arbeit	4
2	Theorieteil	5
2.1	Begriffe und Definitionen	5
2.1.1	Verben	5
2.1.2	Semantik	6
2.1.3	Valenz	7
2.2	Forschungsstand	8
2.2.1	Beth Levin	8
2.2.2	Sabine Schulte im Walde	8
2.2.3	Diverse Projekte	9
2.2.3.1	GermaNet	9
2.2.3.2	TiGer – Korpus	9
2.2.4	Schlussfolgerungen für unsere Arbeit	10
3	Vorgehen	11
3.1	Aus harten Fakten Informationen lesen	11
3.2	Mustersuche von Hand	12
3.2.1	Kategoriensuche	12
3.2.2	Mustersuche	13
3.2.3	Schwierigkeiten	16
3.2.3.1	Ambiguitäten (Mehrdeutigkeiten)	16
3.2.3.2	Redensarten	16
4	Ergebnisse	18
4.1	Vorstellung Kategorien	18
4.2	Hierarchiebaum	22
4.3	Muster zu den einzelnen Kategorien	24
4.4	Evaluation der Ergebnisse	25

5	Fazit / Ausblick	26
6	Literaturverzeichnis	28
7	Anhang	29
7.1	Programm für den Vergleich der Syntaxstrukturen.....	29
7.2	Muster	30
7.3	Hierarchiebaum.....	32
7.4	Muster in Prolog-Notation	33
7.5	Programm zur Evaluation	42
7.6	Programm zum Abarbeiten der Kategorien und Muster	45

Tabellenverzeichnis

Tabelle 1: Matrix Verben.....	13
Tabelle 2: Notationskonventionen.....	14
Tabelle 3: Beispiel Muster.....	15

1 Antivergesslichkeitsprogramme?

Jeder, der sich schon einmal damit beschäftigt hat, ein Lexikon für die maschinelle Sprachverarbeitung zu erstellen, kennt das Problem: Es gibt dutzend Varianten, wie man ein Wort verwenden kann. Besonders deutlich wird dieses Problem bei den Verben. Die Unterscheidung nach transitiven und intransitiven Verben reicht bei Weitem nicht aus, um alle möglichen Satzbauweisen eines Verbs zu erfassen. Manche Verben verlangen einen Nebensatz, andere besitzen die Eigenschaft, dass sie reflexiv sein können und so weiter. Da kann es schnell passieren, dass man eine mögliche Satzstruktur eines Verbs beim Aufbau des Lexikons vergisst.

Im Computerzeitalter sind wir es gewöhnt, dass Computer und vor allem auch Software uns unangenehme oder mühselige Dinge abnehmen. Da liegt es auf der Hand, dass uns ein Computerprogramm bei der Erstellung eines Lexikons behilflich ist und wir so keine mögliche Satzstruktur eines Verbs vergessen. Doch wie soll dieses Programm aussehen? Wissenschaftler haben unzählige Ressourcen erarbeitet, die uns die Erstellung eines Lexikons erleichtern: Semantische Netze, Baumbanken und vieles mehr. Eines scheint allen als Grundlage gemein: Die Semantik. Sie schafft Verbindungen und Beziehungen zwischen einzelnen Verben und kann daher helfen, ein Lexikon zu erstellen, in dem keine Einträge vergessen werden.

1.1 Aufgabenstellung

Doch nur die Tatsache, dass alle möglichen Satzbauweisen, die ein Verb erlaubt, im Lexikon eingetragen sind, stellt noch keine Verbesserung in punkto Praktikabilität des Lexikons dar. Zur Veranschaulichung dienen folgende Sätze, die mit dem Verb „sagen“ gebildet werden können:

„Ich sage die Wahrheit.“

„Ich sage ihm die Wahrheit.“

„Ich sage meinem Freund, dass ich morgen abreise.“

„Ich sage es dir noch.“

„Ich sage dir noch, ob ich komme.“

„Ich sage immer wieder, es muss bald etwas geschehen.“

Ohne Weiteres kann das Verb „sagen“ durch die Verben „mitteilen“ oder „schreiben“ ersetzt werden. Das bedeutet also, dass die Einträge im Lexikon für „sagen“, „mitteilen“ und „schreiben“ identisch sind. Wenn man nun all diese Möglichkeiten bei jedem Verb im Lexikon anführt, plustert sich das Lexikon zu einer immensen Grösse auf und wird immer redundanter. Das Programmierprojekt soll nun dazu dienen,

1. das Lexikon schlanker und damit effizienter zu machen,
2. einen Weg zu finden, wie der Ersteller eines Lexikons möglichst keinen Eintrag mehr vergessen kann und
3. die Möglichkeit der Erweiterung des Lexikons zu gewährleisten, bzw. bestimmte Strukturen für einen Eintrag zu blockieren.

Dies führt zu folgender Aufgabenstellung:

In der Semesterarbeit soll ein Ansatz entwickelt werden, um aus semantischen Merkmalen von Verben deren Valenzstruktur vorherzusagen. Die Aufgabe besteht insbesondere darin, eine Menge von geeigneten semantischen Merkmalen zu identifizieren und eine Abbildung von diesen Merkmalen auf die mögliche Valenzeigenschaften zu finden.

Der Ansatz soll in Prolog implementiert werden. Es wird ein Korpus mit den Valenzeigenschaften von 100-200 Verben zur Verfügung gestellt. Das Ziel ist es, für diese Verben die zulässigen Valenzstrukturen möglichst exakt vorherzusagen, das heisst, die Anzahl der notwendigen Blockierungen oder Einfügungen zu minimieren.

1.2 Aufbau der Arbeit

Nach diesem Kapitel, das als kurze Einführung ins Thema dienen soll, werden im folgenden Kapitel zunächst Definitionen angeführt und Begriffe erläutert, die als Grundlage für die vorliegende Arbeit dienen. Zudem wird der aktuelle Stand der Forschung in diesem Bereich aufgezeigt. Danach folgt die detaillierte Beschreibung, wie wir das Programmierprojekt in Angriff genommen haben. Bevor in einem letzten Kapitel ein Fazit gezogen und ein Ausblick gewagt werden, werden in Kapitel 4 die erzielten Ergebnisse ausführlich vorgestellt.

2 Theorieteil

Im folgenden Teil der Arbeit werden zunächst die Begriffe erklärt, die als Grundlage für die gesamte Arbeit zu betrachten sind.

2.1 Begriffe und Definitionen

2.1.1 Verben

Das Verb ist ein Wort, das normalerweise eine Handlung, einen Zustand oder ein Geschehen ausdrückt. Im Satz nimmt das Verb als Prädikat neben dem Subjekt die zentrale Stelle ein, wodurch es bestimmt, wie viele Ergänzungen notwendig sind, um den Satz vollständig zu machen (vgl. Wikipedia: 2005). Auf diese Eigenschaft der Verben wird unter dem Punkt 2.1.3 Valenzstruktur noch näher eingegangen. Laut dem Metzler Lexikon Sprache sind die Verben diejenige Wortart, deren Repräsentanten rund einen Viertel des gesamten deutschen Wortschatzes ausmachen. Weiter sind die Verben durch bestimmte syntaktische, morphologische und semantische Merkmale gekennzeichnet (vgl. Glück 2000: 771).

Verben werden unterschieden nach transitiven und intransitiven Verben. Transitive Verben beschreiben eine auf ein Objekt gerichtete Tätigkeit, wie zum Beispiel: „Sie liest gerade“. Zudem verfügen transitive Verben über ein Akkusativ-Objekt (auch direktes Objekt genannt), welches aber nicht immer auf den ersten Blick sichtbar sein muss. Beispiel:

„Herr Meier isst gerade.“

„Herr Meier isst gerade den Salat.“

Werden transitive Verben passiviert, wird das Akkusativ-Objekt automatisch zum Subjekt des passiven Satzes:

„Der Salat wird von Herrn Meier gegessen.“

Intransitive Verben weisen nicht unbedingt ein Objekt auf, wie das folgende Beispiel zeigt:

„Sie weint.“

Es können aber Ergänzungen hinzukommen, die sowohl fakultativ wie auch obligatorisch sein können. Eine Ergänzung, die fakultativ oder obligatorisch gebraucht werden kann ist beispielsweise „an der Limmat“.

Beispiele:

„Die Kinder spielen“ vs.

„Die Kinder spielen an der Limmat“ und

„Zürich liegt an der Limmat“ vs.

*„Zürich liegt“.

Die meisten Verben sind Vollverben, das heisst, diese Verben geben dem Satz die Bedeutung. Um die zusammengesetzten Tempora darstellen zu können, muss man aber auf Hilfsverben wie „haben“ oder „sein“ zurückgreifen, wobei die meisten Hilfsverben in nicht zusammengesetzten Tempora auch selbständig gebraucht werden können. Beispiel

„Ich bin glücklich.“

Hilfsverben gehören zudem zu einer kleinen Klasse von Verben, die zusammen mit einem Vollverb eine grammatische Struktur bilden können:

„Ich will schreiben“. (vgl. Wikipedia 2005)

2.1.2 Semantik

Die Semantik bezeichnet eine wissenschaftliche Teildisziplin der Sprachwissenschaft, welche die Bedeutung von Zeichen, speziell von Sprachzeichen, erforscht (vgl. Glück 2000: 618). So verbinden deutschsprachige Menschen mit der Zeichenfolge A-P-F-E-L einem Gegenstand der realen Welt, der folgende Merkmale aufweist: es ist eine Frucht, die rot, grün oder gelb ist, als essbar gilt, saftig ist und meist süsslich schmeckt. In Gedanken sehen wir ein Bild dieser Frucht. Somit verbinden wir die Zeichenkette A-P-F-E-L mit der Bedeutung: „Apfel“.

Die Semantik kümmert sich auch um die Frage, wie Sinn und Bedeutung von komplexen Begriffen aus denen von einfachen Begriffen abgeleitet werden können und stützt sich dabei in der Regel auf die Syntax.

Die Semantik unterscheidet zwischen folgenden Ebenen, die aufeinander aufbauen:

- Lexikalische Semantik: Bedeutung von Wörtern, innere Strukturierung des Wortschatzes insgesamt (Synonym, Antonym).
- Satzsemantik: Bedeutung von grösseren syntaktischen Einheiten (Phrasen und Sätze) anhand der Bedeutung von Wörtern und einem festen Inventar an Verknüpfungsregeln ableiten.

- Textsemantik: Analyse der Kombination von Sätzen als reelle oder hypothetische Sachverhalte zu Erzählungs-, Beschreibungs- oder Argumentationszusammenhängen.
- Diskurssemantik: Analyse auf der Ebene von Texten verschiedener Personen, die miteinander in Beziehung stehen (Diskussion, Unterhaltung, Lehrveranstaltung).

In der Logik und Informatik wird die Semantik als Bedeutungsaspekt von formalen Systemen und Sprachen verwendet. Diese Fachrichtung wird als Formale Semantik bezeichnet. Sie ist insbesondere in der Berechenbarkeitstheorie, der Komplexitätstheorie und zur Verifikation der Korrektheit von Computerprogrammen entscheidend (vgl. Wikipedia 2005).

Unter einem Semantischen Merkmal versteht man schliesslich die Bedeutung eines Wortes sowohl aus wissenschaftlicher (theoretischer) wie natürlicher (praktischer) Sicht (vgl. Wikipedia 2005).

2.1.3 Valenz

Unter der Valenz eines Verbs versteht man die Wertigkeit von Verben, auch Stelligkeit genannt. In der Umgebung eines Verbs sind nur bestimmte, klar definierte syntaktische Einheiten möglich. Die Verbvalenz ist also die Anzahl und die Art der Ergänzungen, die notwendig sind, um mit dem Verb einen Satz zu bilden (vgl. Glück 2000: S. 767). Da der Satz

*„Er betrachtet.“

noch nicht vollständig ist und somit noch keine Bedeutung hat, muss er mit einem Satzglied ergänzt werden, um einen Sinn zu erhalten. Beispiel:

„Er betrachtet ein Foto.“

Bei der Verbvalenz unterscheidet man zwischen nullwertig (avalent → Es schneit.), einwertig (monovalent → Der Beamte arbeitet.), zweiwertig (Ich liebe dich.) und dreiwertig (Ich gebe Klaus ein Buch). Dabei gibt es Verben, die sowohl zwei- wie auch dreiwertig sein können. Zum Beispiel gratulieren:

„Ich gratuliere dir.“ (zweiwertig) vs.

„Ich gratuliere dir zum Geburtstag.“ (dreiwertig).

Die Ergänzungen beeinflussen die Valenz eines Verbs. Dadurch, dass die Ergänzungen fakultativ oder obligatorisch sein können, ist ihr Einfluss auf die Verbvalenz unterschiedlich. Die obligatorischen Ergänzungen sind notwendig,

damit ein Satz grammatikalisch korrekt ist. Die fakultativen Ergänzungen können weggelassen werden, gelten aber trotzdem als spezifische Konstruktion des Verbs.

Bei unserer Arbeit haben wir uns entschieden, vor allem die obligatorischen Ergänzungen zu berücksichtigen und die fakultativen zunächst noch wegzulassen.

Zusammenfassend sei festgehalten, dass die Valenzstruktur die Struktur eines Satzes ist, die sich aus der Gesamtheit der Einzelvalenzen seiner Wörter ergibt (vgl. Wikipedia 2005).

Nachdem in diesem Kapitel die der Arbeit zu Grunde liegenden Definitionen und Begriffe behandelt wurden, wird im nächsten Kapitel der aktuelle Forschungsstand aufgezeigt.

2.2 Forschungsstand

Der aktuelle Forschungsstand stützt sich primär auf die zwei Wissenschaftlerinnen Beth Levin und Sabine Schulte Im Walde und ihre Arbeiten ab. Die Theorien dieser zwei Forscherinnen und weitere Internetprojekte werden in den folgenden Abschnitten kurz vorgestellt.

2.2.1 Beth Levin

Beth Levin schrieb 1993 ihr Buch über die Englischen Verbklassen, die sie erstellt hatte. Dabei hatte sie alle (oder zumindest möglichst alle) englischen Verben in je eine Klasse einzuteilen versucht. Ihr ging es dabei nicht darum, möglichst viele Verben in eine Klasse einteilen zu können, sondern möglichst erschöpfend Klassen zu definieren und mit Verben füllen zu können. Daher bildete sie auch Klassen, die nur mit einem oder zwei Verben gefüllt werden konnten (vgl. Levin 1993). Aufgrund der Tatsache, dass Beth Levin sich alleine auf die englische Sprache bezog, konnten wir ihre Arbeit als Anhaltspunkt und ihre Klassen als Orientierung benutzen. Eine direkte Übernahme ihres Systems war jedoch nicht möglich.

2.2.2 Sabine Schulte im Walde

Im Jahre 2003 schrieb Sabine Schulte im Walde eine sehr ähnliche Arbeit wie Beth Levin für die deutsche Sprache. Auch Schulte im Walde ging so vor, dass sie zuerst Kategorien erstellte, die sie mit Verben mit ähnlichem Sinn füllte. Die

Kategorien stammen zum grössten Teil von Beth Levin ab und wurden von Frau Schulte im Walde auch mit dem englischen Namen benannt.

Sabine Schulte im Walde gewann an und für sich keine neuen Erkenntnisse, sie übertrug Beth Levins System der Verbkategorisierung beinahe vollständig auf die deutsche Sprache (vgl. Schulte Im Walde 2003)

2.2.3 Diverse Projekte

Im Zusammenhang mit lexikalischen und semantischen Ressourcen existieren bereits einige Projekte, die sich auf verschiedenen Ebenen mit dem Thema eines Online-Lexikons auseinandersetzen. Dabei treten vor allem folgende Projekte in den Vordergrund: WordNet, GermaNet, VerbNet, Negra, TiGer-Korpus. In diesem Abschnitt möchten wir vor allem auf zwei der oben genannten Projekte etwas näher eingehen, die dem unsrigen von der Zielausrichtung wohl am nächsten stehen.

2.2.3.1 GermaNet

GermaNet ist ein lexikalisch-semantisches Netz, das in das Euro WordNet (multilinguistische lexikalisch-semantische Datenbank) integriert worden ist.

Dieses Netz verbindet deutsche Nomen, Verben und Adjektive, indem sie semantisch in einer Kategorie gruppiert werden.

Diese Kategorien werden dann auf semantische Relationen untereinander untersucht und dadurch miteinander in Beziehung gesetzt. GermaNet ist sehr ähnlich aufgebaut wie das englische WordNet und wird gerne als Online-Thesaurus angesehen, welcher eine explizite Ontologie definiert (vgl. GermaNet 2005).

2.2.3.2 TiGer – Korpus

Das Tiger-Projekt baut auf den Ergebnissen des Negra-Projekts auf. Ziele des Projekts sind die Verfeinerung der Negra-Annotationsrichtlinien und die Annotation von weiteren 40'000 Zeitungstext-Sätzen.

Das TiGer-Korpus besteht also aus einer Baumbank, die ohne speziell entwickelte Suchwerkzeuge nicht verwertbar ist. Die ungeordneten Teile des Baums sind die Prädikat-Arguments-Strukturen, welche als Graphen mit kreuzenden Kanten annotiert sind.

Die Tiger Treebank besteht aus ungefähr 700'000 Tokens, was ungefähr 40'000 Sätze gleich kommt. Diese Sätze wurden aus deutschen Zeitungstexten gene-

riert, vor allem von der Frankfurter Rundschau. Das Korpus wurde halbautomatisch mit syntaktischen Strukturen getagged (vgl. TiGer Uni Stuttgart 2005).

2.2.4 Schlussfolgerungen für unsere Arbeit

Zuerst gingen wir davon aus, dass wir bestimmte semantische Muster definieren, sie dann den verschiedensten Verben zuordnen und erst anschliessend überprüfen, ob sich irgendwelche Kategorien aus den jeweiligen Verben ergeben.

Aus dem Forschungsstand schlossen wir, dass wir uns auf dem richtigen befinden, wenn wir zuerst versuchen, die Verben auf Grund ihrer Bedeutung in verschiedene Kategorien einzuteilen und erst in einem zweiten Schritt schauen, ob sich grammatikalische Parallelen unter den Verben einer Kategorie ergeben. Sowohl Beth Levin als auch Sabine Schulte Im Walde sind so vorgegangen.

3 Vorgehen

Nachdem im vorigen Kapitel der aktuelle Stand der Forschung erläutert wurde, wird in diesem Kapitel beschrieben, wie das Vorgehen bei der Mustersuche aussah. Hierzu werden zwei verschiedene Wege aufgezeigt, von denen der eine zu ungenügenden Ergebnissen führte und deshalb verworfen wurde.

3.1 Aus harten Fakten Informationen lesen

Nach Beth Levin lassen sich wie bereits besprochen Verben in Kategorien einteilen. Diese Kategorien bilden die semantischen Merkmale der Verben ab und erlauben es, Verben mit gleicher Bedeutung zusammenzufassen. Für unsere Zwecke waren aber nur die Kategorien sinnvoll, die viele Verben enthalten. Denn wenn eine Kategorie beispielsweise nur aus zwei Verben besteht, macht das ein Lexikon nicht viel schlanker. Da es sich bei unserem Projekt um deutsche Verben handelt, wollten wir zunächst untersuchen, ob sich die Einteilung in Kategorien als sinnvoll erweist. Dies überprüften wir anhand der Informationen, die uns bereits zur Verfügung standen.

Basis des Projektes ist ein Lexikon, in dem Verben und ihre möglichen Syntaxstrukturen notiert sind. Das Lexikon besteht aus circa 200 Verben und deren Syntaxstrukturen. Die erste Idee war, dass die Informationen aus dem Lexikon, also die „harten“ Fakten, bereits einen ersten Eindruck vermitteln können, um von der semantischen Bedeutung der Verben auf deren syntaktische Struktur zu schliessen.

Mittels eines eigens zu diesem Zweck erstellen Prolog-Programmes (Code siehe Anhang) sollten die Verben aus dem Lexikon ausgeschrieben werden, die über die selbe Syntaxstruktur verfügen. Dadurch erhofften wir uns, Regelmäßigkeiten in den Strukturen und der Semantik der Verben zu finden. Tatsächlich fanden wir gewisse Regelmäßigkeiten, die eine Einteilung in Kategorien als mögliches Vorgehen sinnvoll erscheinen liessen. Wir fanden heraus, dass bestimmte Syntaxstrukturen auf eine Klasse von Verben schliessen lassen, die sich semantisch gleichen. Folgende Beispiele sollen dies veranschaulichen:

Die Verben

überholen, turnen, tauchen, tanzen, reiten, laufen, krabbeln, fliegen, fahren und drängeln

haben alle eine Syntaxstruktur gemeinsam. In dieser Gruppe befinden sich also ausschliesslich Verben, die eine Fortbewegung ausdrücken.

Die folgenden Verben haben ebenfalls eine Syntaxstruktur gemeinsam:

telefonieren, sprechen, schreiben, schimpfen, rufen, lesen und erzählen.

Auch hier zeigt sich, dass sich diese Verben in ihrer Bedeutung ähnlich sind. Es handelt sich um Verben der Kommunikation.

Leider wird diese Übereinstimmung zu einem grossen Teil dadurch hervorgerufen, dass transitive und intransitive Verben naturgemäss die gleiche Grundstruktur aufweisen. Vor diesem Hintergrund lässt sich auch erklären, warum die folgenden Verben eine gemeinsame Syntaxstruktur haben:

wollen, verschlafen, schmecken, regnen, sammeln, kosten, bekommen und haben.

Diesen Verben ist nur gemein, dass sie alle transitiv sind. Eine semantische Ähnlichkeit aller Verben ist nicht ersichtlich.

Die anfängliche Hoffnung, bereits auf Grund der Fakten im Lexikon einen Hinweis auf eine mögliche Kategorisierung der Verben zu finden, konnte nicht erfüllt werden. Dennoch bestärkten uns die Ergebnisse der Auswertung in der Annahme, dass Kategorien zur Einteilung der Verben sinnvoll sind. Im folgenden Abschnitt wird nun aufgezeigt, wie diese Kategorien gebildet wurden und wie aus den Kategorien Muster für die Syntaxstrukturen entstanden sind.

3.2 Mustersuche von Hand

Dieser Abschnitt beschäftigt sich zunächst damit, welche Kategorien unserer Arbeit zu Grunde liegen. Anschliessend wird aufgezeigt, welche Schreibweise für die Muster zum Einsatz kam.

3.2.1 Kategoriensuche

Bei der Suche nach Kategorien haben wir uns stark an den Kategorien von Beth Levin und Sabine Schulte Im Walde orientiert, haben aber ihre Kategorien nicht eins zu eins übernommen. Wir suchten nach den Kategorien, die möglichst viele Verben beinhalten. Bei der Bildung der Kategorien suchten wir zu-

erst einen Oberbegriff für Verben, die alle eine ähnliche Semantik haben. Beispielsweise Kommunikation, Wahrnehmung und Aktivitäten. Nun stellten wir eine Liste mit Verben auf, die unserer Meinung nach in die gefundenen Kategorien gehören (eine genaue Auflistung aller Kategorien findet sich in Kapitel 4 dieser Arbeit). Die Hoffnung war, dass die Verben einer Kategorie auch in puncto Valenzstrukturen einheitlich sind, das heisst, dass Muster gefunden werden können.

3.2.2 *Mustersuche*

Bei Mustern handelt es sich um Gemeinsamkeiten der Verben, was den Aufbau eines Satzes angeht, der mit diesen Verben gebildet werden kann. Zur Veranschaulichung soll die Kategorie „Kommunikation“ in Bezug auf die Gemeinsamkeiten der Verben aufgezeigt werden.

Kommunikation ist ein Prozess, an dem zwei Parteien beteiligt sind. Zum einen gibt es eine Partei, die Information aussendet (Sender), zum anderen eine Partei, die Information empfängt (Empfänger). Inhalt einer Kommunikation ist immer eine Nachricht, Botschaft oder auch bloss Information. Wenn wir unterstellen, dass diese Merkmale jedem Verb der Kategorie zugeordnet werden können, ergibt sich eine Matrix:

Verb	Sender	Nachricht	Empfänger
sagen	X	X	X
erzählen	X	X	X
berichten	X	X	X
usw.			

Tabelle 1: Matrix Verben

Diese Matrix könnte als Muster aufgefasst und in eine Art Muster-Code übersetzt werden. Doch nicht in jeder Kategorie lässt sich eine derartige Matrix finden. Tatsächlich ist es so, dass die Kategorie „Kommunikation“ praktisch die einzige Kategorie ist, bei der eine solche Matrix konsistent erstellt werden kann. Aus diesem Grund suchten wir nach einer anderen Variante, Regelmässigkeiten in den Kategorien zu finden. In so genannten Valenzlexika findet man zu vielen Verben (leider nicht zu allen) sämtliche Satzbauarten, also Valenzstrukturen zu diesen Verben. Wir suchten nun also sämtliche Verben einer Kategorie im Valenzlexikon und erstellten eine Tabelle, in die die möglichen Valenzstruk-

turen eingetragen wurden. Die Einträge in der Tabelle wurden verglichen und die Gemeinsamkeiten in Form eines Musters festgehalten. Um eine einheitliche Schreibweise zu garantieren, hielten wir uns strikt an folgende Konventionen:

Subjekt (subj)	NP _[str]	Nominalphrase mit strukturellem Kasus	<i>Er</i> schläft.
	NP _[exp]	Expletivum	<i>Es</i> regnet.
	NP+S _{dass}	NP-Korrelat mit dass-Satz	Mich hat es gefreut, <i>dass du kamst</i> .
	NP+S _{que}	NP-Korrelat mit Fragesatz	Mich hat es interessiert, <i>wer das war</i> .
	NP+S _{inf}	NP-Korrelat mit Infinitivphrase	Mir erschien es dumm, <i>ihn zu fragen</i> .
	S _{dass}	dass-Satz	<i>Dass du kommst</i> , freut mich.
	S _{que}	Fragesatz	<i>Wer das war</i> interessiert mich nicht.
	S _{inf}	Infinitivphrase	<i>ihn zu fragen</i> schien unangebracht.
	S _{VII}	Verb-zweit-Satz	Mir scheint, <i>das ist die Lösung</i> .
Direktes Objekt (dobj)	NP _[str]	Sie macht es sich nicht einfach.	
	NP _[exp]	Ich kenne <i>sie</i> .	
	NP+S _{dass}	Ich habe es gewusst, <i>dass es stimmt</i> .	
	NP+S _{que}	Ich habe es gewusst, <i>worum es geht</i> .	
	NP+S _{inf}	Ich habe es geschafft, <i>aufzuhören</i> .	
Indirektes Objekt (iobj)	NP _[str]	Er gibt <i>ihr</i> das Buch.	
dativus commodi / incommodi (dativus_cic)	Der dativus commodi beschreibt die Instanz, zu deren Nutzen eine Handlung ausgeführt wird. Der dativus incommodi verweist auf die Person, gegen deren Willen etwas geschieht.		Das Kind hat <i>der Mutter</i> Blumen gepflückt. Gestern ist <i>mir</i> die Vase kaputt gegangen.
Pertinenz-Element (pertinenz_elm)	Das Pertinenz-Element bezeichnet den Besitzer einer Entität, die als ein Argument des Verbs realisiert ist.		Der Hund hat <i>ihn</i> ins Bein gebissen.
[]	Eckige Klammern drücken Optionalität aus.		
{ }	Geschweifte Klammern zeigen an, dass es sich um eine Menge von Möglichkeiten handelt, aus denen gewählt werden kann.		

Tabelle 2: Notationskonventionen

Um ein Muster zu finden, das alle Verben einer Kategorie abdeckt, mussten wir Kompromisse eingehen. So kamen beispielsweise Ergänzungen (Situativergänzungen, Artergänzungen oder Direktivergänzungen) in unseren Mustern nicht zum Tragen, da sie praktisch bei jedem Verb auftreten können, unabhängig von dessen Semantik. Einzige Ausnahme bilden Verben, bei denen eine Ergänzung zwingend notwendig ist:

„Zürich liegt an der Limmat.“

*Zürich liegt.“

In diesen Fällen ist die Ergänzung notwendig, damit der Satz korrekt bleibt. Des Weiteren wurden Präpositionalobjekte weitgehend ausser Acht gelassen.

Unter Einhaltung der oben angeführten Schreibweise ergaben sich Muster der folgenden Art für die einzelnen Kategorien:

Muster Mitteilung:

subj=NP _[str] und	<i>Er spricht.</i>
(
dobj=NP _[str] oder	<i>Er erzählt eine Geschichte.</i>
dobj=NP _[str] +S _{dass} oder	<i>Er erzählt es, dass die Kinder gespielt haben.</i>
dobj=NP _[str] +S _{que} oder	<i>Er sagt es, ob er kommen wird.</i>
dobj=S _{dass} oder	<i>Er erzählt, dass er den Berg erklommen hat.</i>
dobj=S _{que} oder	<i>Er sagt, warum er das getan hat.</i>
dobj=S _{VII}	<i>Sie sagt, sie habe ihn gesehen.</i>
)	
[
und	
iobj=NP _[str]	<i>Sie sagt mir, sie habe ihn gesehen.</i>
].	

Tabelle 3: Beispiel Muster

Alle Muster sind ausführlich im Anhang zu finden.

3.2.3 Schwierigkeiten

In diesem Abschnitt werden die Schwierigkeiten und Probleme beschrieben, die sich bei der Mustersuche und Kategoriebildung ergeben haben. Dies dient auch zur Klärung der Frage, wie erschöpfend die Muster die Kategorien beschreiben können.

3.2.3.1 Ambiguitäten (Mehrdeutigkeiten)

Natürliche Sprachen lassen oft für die gleiche Abfolge von Wortformen verschiedene Arten von Analysen zu. Dasselbe gilt auch für Verben. Nehmen wir als Beispiel das Verb „abnehmen“, wird die Ambiguität sichtbar:

„Das Telefon läutete, doch niemand *nahm ab*.“

„Die Zahl der ungeklärten Mordfälle *nahm* in den letzten Jahren *ab*.“

Das Verb „abnehmen“ hat also zwei unterschiedliche Bedeutungen. In welche Kategorie gehört das Verb nun? Eine Möglichkeit wäre, das Verb in all jenen Kategorien zu erfassen, in die es auf Grund seiner Semantik gehört. Dies ist aber ein sehr aufwendiges Prozedere, da keine Bedeutung vergessen werden dürfte. Voraussetzung hierfür wäre, dass alle Verben, die in unseren Kategorien vorkommen, auf mögliche Ambiguitäten geprüft werden müssten. Aus zeitlichen Gründen war uns dies für das Projekt nicht möglich. Wir entschieden uns deshalb, jedes Verb nur in einer Kategorie unterzubringen und allfällige zusätzliche Bedeutungen ausser Acht zu lassen.

3.2.3.2 Redensarten

Viele Verben finden sich in Redewendungen und geflügelten Worten wieder. Doch in Redewendungen haben Verben oft eine Valenzstruktur, die sonst kein Verb der gleichen Kategorie besitzt. Folgende zwei Sätze sollen dies verdeutlichen:

„Er schwitze Blut und Wasser.“

„Er starb einen schweren Tod.“

Im Allgemeinen haben Verben wie *schwitzen*, *sterben*, *atmen* oder *bluten* kein direktes Objekt. Dies ist nur in Redewendungen der Fall. Würden diese möglichen Strukturen für alle Verben der selben Kategorie vorausgesetzt, würde dies das Muster und damit die Kategorie unbrauchbar machen. Damit das Muster

aber dennoch zur Kategorie passt, haben wir die Valenzstrukturen der Verben, die in einer Redewendung vorkommen, ausser Acht gelassen.

Es gäbe noch einige Schwierigkeiten, auf die man näher eingehen könnte. Sie sind aber für den vorliegenden Bericht von geringer Bedeutung und werden daher hier nicht erwähnt.

Nachdem nun aufgezeigt wurde, wie die Kategorien und Muster gebildet wurden, werden sie im nächsten Kapitel ausführlich vorgestellt und beschrieben.

4 Ergebnisse

In den folgenden Abschnitten werden die Ergebnisse unseres Programmierprojektes vorgestellt. Zunächst werden alle Kategorien ausführlich vorgestellt, danach folgen der entstandene Hierarchiebaum und das Beispiel eines Musters. In einem letzten Schritt werden die Ergebnisse der Evaluation präsentiert.

4.1 Vorstellung Kategorien

Bei der Suche nach Gruppen von Verben mit ähnlicher Bedeutung sind wir auf die folgenden Kategorien gestossen. In den runden Klammern nach dem Kategoriennamen findet sich der Name der Kategorie, den wir im Prolog verwendeten.

- **Gartenarbeit** (gartenarbeit): beinhaltet die Verben: *säen, pflanzen, pflücken, ernten, sammeln*. In dieser Kategorie fassen wir Tätigkeiten zusammen, die im Garten oder auf dem Feld, eventuell im Wald, ausgeführt werden. Dass diese Kategorie ohne weiteres noch ergänzbar ist (mit zum Beispiel giessen), ist uns voll bewusst. Gerade hier möchten wir betonen, dass wir auch bei der Benennung unserer Kategorien einen gewissen Kompromiss eingegangen sind. Unter Gartenarbeit könnte man sich auch etwas vorstellen, was nicht unbedingt mit dem Verb sammeln ausgedrückt wird. Wir wollten jedoch vermeiden, dass unsere Kategorien Namen wie „Eine Vielzahl von verteilten Dingen an einem Ort vereinigen“ erhalten. Zum einen ist dies im Programmcode unpraktisch und zum anderen sind solche Namen für Kategorien zwar äusserst ausführlich beschreibend, sie engen aber eine Kategorie auch zu sehr ein.
- **Sterben** (become_not_alive): beinhaltet die Verben: *sterben, ertrinken, verbluten, ersticken*. Diese Kategorie enthält Verben, die alle ausdrücken, dass der „Sprechende“ sein Leben beenden muss, dies durch natürliche oder unnatürliche Einflüsse.
- **Vegetatives Nervensystem** (veg_nervensystem): beinhaltet die Verben: *atmen, bluten, husten, schwitzen*. In dieser Kategorie befinden sich die Verben, die eine körperliche Tätigkeit ausdrücken, welche vom Menschen nicht bewusst ausgeführt werden. Uns ist dabei bewusst, dass bluten an und für sich keine Aktivität des vegetativen Nervensystems ist.

Da man jedoch nicht bewusst eine Blutung auslösen oder stoppen kann, fanden wir dieses Verb in dieser Kategorie passend.

- **Fortbewegung mit / ohne Instrument** (fortbewegung_mit_instrument) / Fortbewegung ohne Instrument (fortbewegung_ohne_instrument): beinhaltet die Verben: *fahren* und *reiten*, bzw.: *laufen*, *schwimmen*, *kriechen*. Da man sich sowohl mit einem Mittel (Auto, Fahrrad, Pferd) oder auch auf seinen blossen Füßen fortbewegen kann, haben wir uns entschlossen, daraus zwei verschiedene Kategorien zu machen. Dabei stellten wir fest, dass sich diese zwei Kategorien in ihren Valenzstrukturen nicht unterscheiden. Wir entschieden uns aber dafür, es bei zwei Kategorien zu belassen, da es in unseren Augen einen Sinnunterschied macht, ob man sich auf seinen eigenen Füßen oder mit Hilfe eines Fahrzeuges fortbewegt. Abgesehen davon muss man bedenken, dass man gewisse Unterschiede zwischen diesen zwei Kategorien feststellen könnte, wenn man die Kategorisierung auf alle möglichen Ergänzungen, von denen wir doch gewisse weggelassen haben, ausweiten würde.
- **Zerstören_mensch** (mensch): beinhaltet die Verben: *töten*, *umbringen*, *erwürgen*, *erdrosseln*, *erschliessen*, *erstechen*, *ermorden*, *vergiften*. Uns fiel auf, dass man nicht nur Sachen, sondern eben auch Menschen ‚zerstören‘ kann. Dieses Zerstören_mensch beinhaltet Verben, die zum Tod durch Fremdeinwirkung führen.
- **Zerstören_Sache** (sache): beinhaltet die Verben: *zerstören*, *abreissen*, *demolieren*, *schneiden*, *brechen*, *reissen*, *zertrümmern*. Verben, die eine zerstörerische Tätigkeit ausdrücken, die auf eine Sache bezogen ist, werden in dieser Kategorie zusammengefasst. Ob man dabei die Verben schneiden, brechen und reissen in dieser Form belässt oder mit dem Präfix „zer-“, verstärkt (zerschneiden), hat auf die Valenzstruktur keinen Einfluss.
- **Nahrungsmittelverarbeitung** (nahrungsmittelverarbeitung): beinhaltet die Verben: *kochen*, *braten*, *dämpfen*, *garen*, *rösten*, *backen*, *grillen*, *grillieren*. Da das Essen in unserem Leben einen grossen Stellenwert einnimmt, konnten wir es nicht unterlassen, diese Kategorie zusammenzustellen.

-
- **Schaffen / Kreieren** (allgemein): beinhaltet die Verben: *bauen, erschaffen, bilden, kreieren, zusammensetzen, zusammenfügen, schmieden, produzieren, erzeugen*. In dieser Kategorie findet man Verben, die die Produktion oder Herstellung einer Sache ausdrücken.
 - **Besitzänderung** (besitzaenderung): beinhaltet die Verben: *geben, nehmen, schenken, ersitzen, kaufen, leihen, mieten, stehlen, entreissen, entwenden, rauben, wegnehmen*. Die Verben dieser Kategorie drücken einen Besitzwechsel aus. Ob dieser Besitzwechsel gewollt oder ungewollt ist und ob er einseitig oder mehrseitig ist, konnten wir beim Verarbeiten dieser Kategorie nicht mehr berücksichtigen und haben es bei einer allgemeinen Kategorie belassen. Gerade diese Kategorie könnte man aber unterteilen, zum Beispiel in die Untergruppen Transaktion-geben und Transaktion-nehmen. Unser Ziel war jedoch, eine möglichst homogene Kategorie mit vielen Verben zu erstellen. Aus diesem Grund belassen wir es bei der einen Kategorie „Besitzänderung“.
 - **Mitteilung** (mitteilung_mit_hilfsmittel, mitteilung_ohne_hilfsmittel): beinhalten die Verben: *faxen, mailen, smsen, morsen, chatten bzw. sprechen, sagen, rufen, erzählen, berichten*. Kommunikation ist eine sehr grosse Kategorie und daher lag es uns am Herzen, dass wir uns in einer Unterkategorie auf die Mitteilung an sich konzentrieren. Daraus ergaben sich die zwei Gruppen Mitteilung, die mit einem – meist technischen – Hilfsmittel (Telefon, Morseapparat) geschieht und Mitteilung, die alleine mit dem menschlichen Sprechapparat möglich ist.
 - **Wahrnehmung** (wahrnehmung): beinhaltet die Verben: *sehen, riechen, schmecken, fühlen, hören*. Die Tätigkeiten, welche mit unseren fünf Sinnen möglich sind, gehören unseres Erachtens zu den grundlegenden Tätigkeiten. Daher bildeten wir eine Kategorie, welche die fünf Wahrnehmungsverben beinhaltet.
 - **Vermittlung von Wissen** (wissensvermittlung): beinhaltet die Verben: *lehren, unterrichten, instruieren, anleiten*. In dieser Kategorie werden alle Tätigkeiten zusammengefasst, die mit der Vermittlung von Wissen zusammenhängen.

-
- **Wahrheitswert einer Mitteilung** (wahrheitswert): beinhaltet die Verben: *behaupten, gestehen, zugeben, leugnen, feststellen, widerlegen, festhalten, bezweifeln*. Den Wahrheitswert einer Mitteilung festzustellen und zu argumentieren, stellt einen grossen Teil der zwischenmenschlichen Kommunikation dar. Daher versuchten wir mit möglichst vielen Verben dieser semantischen Kategorie eine endgültige Kategorie zu bilden. Durch die Heterogenität dieser Verben mussten wir auf die eine oder andere Tätigkeit (z.B. meinen) verzichten.
 - **Musische Arbeit** (musische_aktivitaeten): beinhaltet die Verben: *komponieren, musizieren, vorführen, aufführen, basteln, tanzen*. Unter Musischen Aktivitäten verstehen wir Tätigkeiten, welche mit künstlerischen Dingen zu tun haben. Aufgrund der grossen Bandbreite dieser Verben könnte man diese Kategorie auch aufteilen in eine Kategorie, welche die schöpferischen Tätigkeiten beinhaltet und eine zweite, welche Verben des Präsentierens zusammenfasst. Wie bereits bei der Kategorie der Besitzänderung angesprochen, wollten wir uns auf möglichst grosse Kategorien konzentrieren und haben uns auch in diesem Fall gegen eine weitere Unterteilung entschieden.
 - **Geräusche** (geraeusche): beinhaltet die Verben: *knistern, knacken, rascheln, quietschen, brummen, rattern, knarren*. In dieser Kategorie befinden sich alle Verben, die ein Geräusch ausdrücken, das von der Umwelt und nicht dem „Sprecher“ selber verursacht wird.
 - **Wetter** (wetter): beinhaltet die Verben: *regnen, schneien, hageln, blitzen, donnern, nieseln, winden, stürmen*. Da nicht nur die Menschen Tätigkeiten ausführen können, erachteten wir es als wichtig, auch diese Kategorie zu berücksichtigen.
 - **Tierlaute** (tierlaute): beinhaltet die Verben: *bellen, brüllen, quaken, miauen, muhen, wiehern, fiepen, piepsen, grunzen, pfeifen, zwitschern, singen*. Tiere äussern sich nicht nur, auch sie üben Tätigkeiten aus, wobei wir uns dann aber dafür entschieden, nur die Tierlaute zu berücksichtigen. Dank ihrer grammatikalischen Ähnlichkeit konnten wir alle in einer Kategorie belassen. Hier kann man anmerken, dass auch ein Mensch Laute von sich geben kann, die sehr stark an Tierlaute erinnern. Diese

Kategorie hängt jedoch am „sächlichen“ Ast, was eine gemeinsame Klasse von menschlichen und tierischen Lauten verhindert.

4.2 Hierarchiebaum

Unsere Kategorien fügten wir zu einem Hierarchiebaum zusammen (siehe Anhang). Ziel war es, einen etwas komplexeren Baum zu erstellen, damit wir überprüfen konnten, ob unsere Ideen auch in komplexeren Zusammenhängen funktionieren. Die Kategorien, welche die Muster beinhalten, befinden sich alle auf der untersten Ebene. Diese „Musterkategorien“ werden zu allgemeineren Kategorien zusammengefasst und jene wiederum verallgemeinert, so dass wir auf der obersten Stufe die zwei Kategorien „menschlich“ und „sächlich“ haben. Bei der Suche nach einem Muster, bzw. Auswahl einer Kategorie, wird die suchende Person von oben nach unten geleitet. So muss sie zuerst die Auswahl treffen, ob die Tätigkeit einen Menschen oder eine Sache betrifft, bzw. ob das Subjekt, das die Handlung ausführt, menschlich oder sächlich ist. Entschliesst sich der Benutzer für sächlich, stehen ihm drei Kategorien zur Auswahl: Geräusche, Wetter und Tierlaute. Wählt der Benutzer eine dieser Kategorien an, werden ihm alle möglichen Valenzmuster plus die Liste der Verben, die zu dieser Kategorie gehören, ausgegeben.

Wird hingegen die Kategorie menschlich ausgewählt, stehen nachher vier Unterkategorien zur Auswahl: die Kategorie der Aktivität, die der Fortbewegung, die des Zustandes / der Form und die der Kommunikation.

Bei der Kategorie der Aktivität haben wir zwischen bewusster und unbewusster Aktivität unterschieden. Die unbewussten menschlichen Aktivitäten beinhalten als Unterkategorie die Aktivitäten des Vegetativen Nervensystem des Menschen.

Im Gegensatz zu den unbewussten Aktivitäten verstehen wir unter den bewussten diejenigen Aktivitäten, die eine Person willentlich ausführt. Bei dieser Kategorie haben wir uns für die Verben der Gartenarbeit als die eine und für die Verben des Sterbens (da dies der Mensch zwar nicht unbedingt willentlich, jedoch bewusst erlebend ausführt) als die andere Unterkategorie entschieden.

Bei der Kategorie der menschlichen Fortbewegung unterscheiden wir zwischen Fortbewegung mit Instrument, sprich mit einem Hilfsmittel (Fahrzeug, Bus, ...) und Fortbewegung ohne Hilfsmittel.

Die menschliche Kategorie Zustand / Form hat weitere Unterkategorien, die Tätigkeiten im Zusammenhang mit Zustands- oder Formveränderungen ausdrücken. Diese Verben werden den folgenden Kategorien zugeordnet: Mensch, Sache, Nahrungsmittelverarbeitung, Allgemein und Besitzänderung. Die Kategorien Mensch und Sache werden in der Überkategorie Zerstören zusammengefasst. Zur Zustandsänderung gehört jedoch nicht nur die Zerstörung, sondern auch das Schaffen und Kreieren, wozu die Kategorien Nahrungsmittelverarbeitung und Allgemein gehören. Keine Veränderung des Zustands wird durch die Kategorie Besitzänderung ausgedrückt.

Die Kategorie der Kommunikation ist mit ihren Unterkategorien Mitteilung, Informationsgewinnung und Musische Arbeit ungefähr gleich komplex aufgebaut wie die Kategorie des Zustands. Die Unterkategorie Mitteilung wird aufgeteilt in Mitteilung mit bzw. ohne Hilfsmittel (Instrument), die Unterkategorie Informationsgewinnung hingegen wird aufgeteilt in die Kategorien Wahrnehmung, Wissensvermittlung und die Erkennung des Wahrheitswerts.

Der Baum zeigt deutlich, dass gewisse Kategorien bereits mit einigen Unter- und Unter-Unterkategorien versehen sind, wo hingegen andere Äste nicht so kompliziert aufgebaut sind. In einigen Bereichen, vor allem bei den menschlichen Aktivitäten, unter Umständen aber auch in der Kategorie der Kommunikation, kann der Baum offensichtlich noch erweitert werden. Uns ging es primär darum, einen Anfang zu machen, der zum Weiterbauen animiert.

4.3 Muster zu den einzelnen Kategorien

Statt jedes Muster einzeln vorzustellen, haben wir beschlossen, ein einziges Muster als Beispiel für alle detailliert vorzustellen. Die anderen Muster wurden analog zu diesem Schema gebildet.

Die Muster sehen in ihrem Aufbau alle wie folgt aus (die übrigen Muster sind im Teil 7: Anhang zu finden):

```
muster(kategorienname,[verb1, verb2, verb3],  
  parameters(  
    deep_object( _ ),  
    dativus_cic(none,),  
    pertinenz_elm(none),  
    perfekt_aux( _ )),  
  arguments([arg_subj(np_str), arg_dobj(np_str), arg_iobj(np_str)]))
```

Unser Prädikat heisst immer gleich, nämlich *muster*. Anschliessend ist der Name des Musters angegeben, in diesem Fall *kategorienname* und die Liste, welche alle Verben beinhaltet, die in diese Kategorie gehören. Hier: *verb1*, *verb2* und *verb3*.

Von den Parametern interessierten uns vor allem der Dativus (in)commodi, hier *dativus_cic()* genannt, und das Pertinenzelement, hier als *pertinenz_elm()* angegeben. Das Tiefenobjekt und das Hilfsverb zur Bildung des Perfekts haben wir mit einer anonymen Variablen gekennzeichnet, da wir diese Information nicht verwertet haben. Weiter interessierten uns natürlich die Argumente (*arguments*), von denen wir uns jedoch auf die Subjekte (*arg_subj()*), die direkten Objekte (*arg_dobj()*), die indirekten Objekte (*arg_iobj()*) und die absolut notwendigen Präpositionalgefüge (*arg_pcmp([])*) konzentrierten. Zusätzliche Angaben wie fakultative Direktivergänzungen oder Artergänzungen liessen wir beiseite.

4.4 Evaluation der Ergebnisse

Unsere Aufgabe beinhaltete einen Testlauf eines Programms (siehe Anhang) über ein Lexikon von ungefähr 200 Verben mit Beispielsätzen, um zu überprüfen, ob alles funktioniert. Ziel war, möglichst viele Verben des Lexikons und ihre Valenzstrukturen mit unserem Programm zu matchen. Unser Programm untersuchte dabei die Verben und die dazugehörenden Muster auf Übereinstimmungen in den Valenzstrukturen.

Wie unsere Ergebnisse zeigen, wurden die meisten Muster des Lexikons von unseren Mustern erfasst und als Übereinstimmung angezeigt. Bei der genaueren Kontrolle der Resultate, stellten wir folgende Punkte fest:

1. Da wir uns bei der Kategorienbildung nicht nur am Lexikon, sondern primär an unserem eigenen Wortschatz orientiert hatten, mussten wir feststellen, dass wir in unseren Kategorien über 20 Verben auflisteten, die nicht im Lexikon vorhanden waren.
2. Einzelne Abweichungen entstanden, weil wir (in Absprache mit unserem Betreuer) gewisse Angaben wie Art- oder Direktivergänzungen weglassen hatten. Dies hatte zur Folge, dass die Valenzstruktur in unserem Programm nicht mehr zu 100 Prozent mit derjenigen aus dem Testlexikon übereinstimmte und dadurch keinen Match anzeigte.
3. Bei der Erstellung unseres Programms hatten wir uns auf die Grundbedeutungen und einfacheren Strukturen der Verben und Beispielsätze konzentriert. Wir liessen allzu verschachtelte Sätze und veraltete Konstruktionen beiseite, insbesondere auch aus dem Grund, dass es Kategoriebildung vereinfachte. Da das Testlexikon aber gewisse Verschachtelungen und Formulierungen berücksichtigte und mit seinen 200 Verben und Beispielsätzen auch nicht erschöpfend war, fielen gewisse Übereinstimmungen leider weg.
4. Eine kleine Menge von Nicht-Übereinstimmungen folgte aus der Tatsache, dass auf beiden Seiten zwischendurch ein Schreib- oder Überlegungsfehler passierte.

5 Fazit / Ausblick

Ziel des Programmierprojektes war es, Valenzstrukturen von Verben auf Grund ihrer Semantik vorherzusagen. Dieses Ziel wurde in einem (allerdings sehr beschränkten Rahmen) erreicht. Wir haben die Verben nach ihren semantischen Merkmalen zunächst in Kategorien eingeteilt. Diese Idee ist nicht neu, denn schon Beth Levin und Sabine Schulte im Walde haben diesen Weg gewählt, um Verben einzuteilen. Die Schwierigkeit bestand darin, möglichst homogene Kategorien zu finden, was häufig an der Ambiguität von Verben oder an zu unterschiedlichen Valenzstrukturen scheiterte. Wir mussten uns letztlich zwischen Funktionalität und Perfektion entscheiden. Denn wollte man alle Verben der deutschen Sprache so in Kategorien einteilen, dass allen Verben in der Kategorie exakt die selben Valenzstrukturen zugewiesen werden können, müsste man unzählige Kategorien bilden. Das hätte den Effekt, dass in vielen Kategorien nur ein oder zwei Verben enthalten wären. Wir haben uns entschieden, das eine oder andere Merkmal eines Verbs nicht zu berücksichtigen, wenn das Verb dadurch aus der Reihe tanzen und nicht mehr ins Muster passen würde. Fakt ist aber, dass durch dieses Vorgehen wertvolle Information verloren geht. Denn lässt man gewisse für ein Verb typische Eigenschaften einfach weg, beraubt man sich der Vielfalt der möglichen Valenzstrukturen des Verbs.

Die Evaluation unseres Programms ergab, dass wir eine hohe Übereinstimmung erreicht haben, wenn ein Verb sowohl in unseren Kategorien, als auch im bereits vorhandenen Lexikon vorkam. Allerdings ist nur ein Teil unserer Verben auch tatsächlich im Lexikon vertreten. Hauptursache hierfür ist sicher, dass wir auch Kategorien wie beispielsweise „Kommunikation“ gebildet haben, die moderne Verbe wie faxen oder mailen enthält. Solche Verben sind im bestehenden Lexikon nicht zu finden. Auch Wetterverben fehlen im Lexikon gänzlich, waren für uns aber eine äusserst homogene Kategorie. Ein weiterer Grund dafür, dass im Lexikon weit mehr Verben vorhanden sind, die nicht in eine unserer Kategorien aufgenommen wurden, war die angestrebte Homogenität der Kategorien. Wie bereits erwähnt haben wir Merkmale weggelassen, wenn sie das Muster gesprengt hätten.

Um eine Wissensbasis für ein Programm zu schreiben, das alle Verben der deutschen Sprache kennt und deren Valenzstruktur auf Grund der Semantik

vorhersagen kann, müsste man mehr Kategorien bilden. So könnte man feinere und spezifischere Muster für einzelne Verben finden. Der Aufbau einer solchen Sammlung von Kategorien ist zwar aufwendig, wenn man sich aber an unser Vorgehen hält, ist es durchaus machbar.

Der zeitliche Rahmen unserer Programmierprojektes war naturgemäss begrenzt, weshalb wir lediglich den Grundstein für weitere Arbeiten legen konnten. Doch die Idee, Valenzstrukturen von Verben rein auf Grund ihrer semantischen Merkmale vorherzusagen, scheint in die Realität übersetzbar zu sein. Der Aufbau unseres Programms erlaubt es jederzeit, noch weitere Kategorien und Muster zu implementieren. Die Frage ist nur, ob sich der Aufwand lohnt. Denn die Lexikologie beschäftigt sich als eigenständige Wissenschaft schon lange mit der Frage, wie ein perfektes Lexikon aufgebaut sein muss. Pauschal lässt sich diese Frage aber nicht beantworten, denn es kommt immer darauf an, für welche Zwecke ein Lexikon erstellt wurde.

6 Literaturverzeichnis

GermaNet: In: <http://www.sfs.nphil.uni-tuebingen.de/lisd/>. 14.07.2005.

Glück, Helmut (Hrsg.) (2000): Metzler Lexikon Sprache. Stuttgart.

Levin, Beth (1993): English verb classes and alternations: A preliminary investigation. Chicago.

Schulte im Walde, Sabine (2003): Experiments on the automatic induction of german semantic verb classes. In: AIMS (Arbeitspapiere des Instituts für maschinelle Sprachverarbeitung), Universität Stuttgart, Vol. 9, Nr. 2.

Schumacher, Helmut / Engel, Ulrich (Hrsg.) (1976): Kleines Valenzlexikon deutscher Verben. Tübingen.

TiGer-Korpus: In: <http://www.ims.uni-stuttgart.de/projekt/TIGER>. 14.07.2005.

Wikipedia (2005): In: <http://de.wikipedia.org/wik>. 14.07.2005.

7 Anhang

7.1 Programm für den Vergleich der Syntaxstrukturen

```

/* Schreibt saemtliche Syntaxstrukturen und die zugehoerigen Verben
aus */
/* Aufruf: lexikon_vergleich und tools konsultieren, such., schreib.
Aufraeumen mit killemall.*/

/**Code fuer den Vergleich**/
/*****/

schreiballe :-
verb(ID,P,A,examples([E|_]),verb(ID2,P2,A2,examples([E2|_])),
    P==P2,
    A==A2,
    write(ID), write(' : '),write(ID2),nl,
    writestring(E),write(' : '),writestring(E2),nl,
    fail.

/** fuehlt die Verben in die Synstruct-Praedikate ein **/

notiere(A,B,_ ,Wort):- syn-
struct(A,B,_ ,Wortliste),member(Wort,Wortliste),!.
notiere(A,B,_ ,Wort):- re-
tract(synstruct(A,B,Bsp,Worter)),assertz(synstruct(A,B,Bsp,[Wort|Worte
r])),!.
notiere(A,B,Beispiel,Wort):- assertz(synstruct(A,B,Beispiel,[Wort])).

/** macht alle synstruct-praedikate weg **/

killemall:-retract(synstruct(_,_ ,_ ,_)),fail,!.
killemall:-write('sind weeeeggg...., und jeeeetzt?'),nl.

/** gibt das Resultat in menschlesbarer Form aus **/

schreib:-synstruct(A,B,C,D),
write('*****'),nl,
write('Struktur: '), write(A), write('; '), write(B),nl,nl,
write('Beispiel: '), writestring(C),nl,nl,
write(D),nl,nl,fail.

schreib.

such:-verb(ID,P,A,examples([E|_]),
    notiere(P,A,E,ID),
    fail.

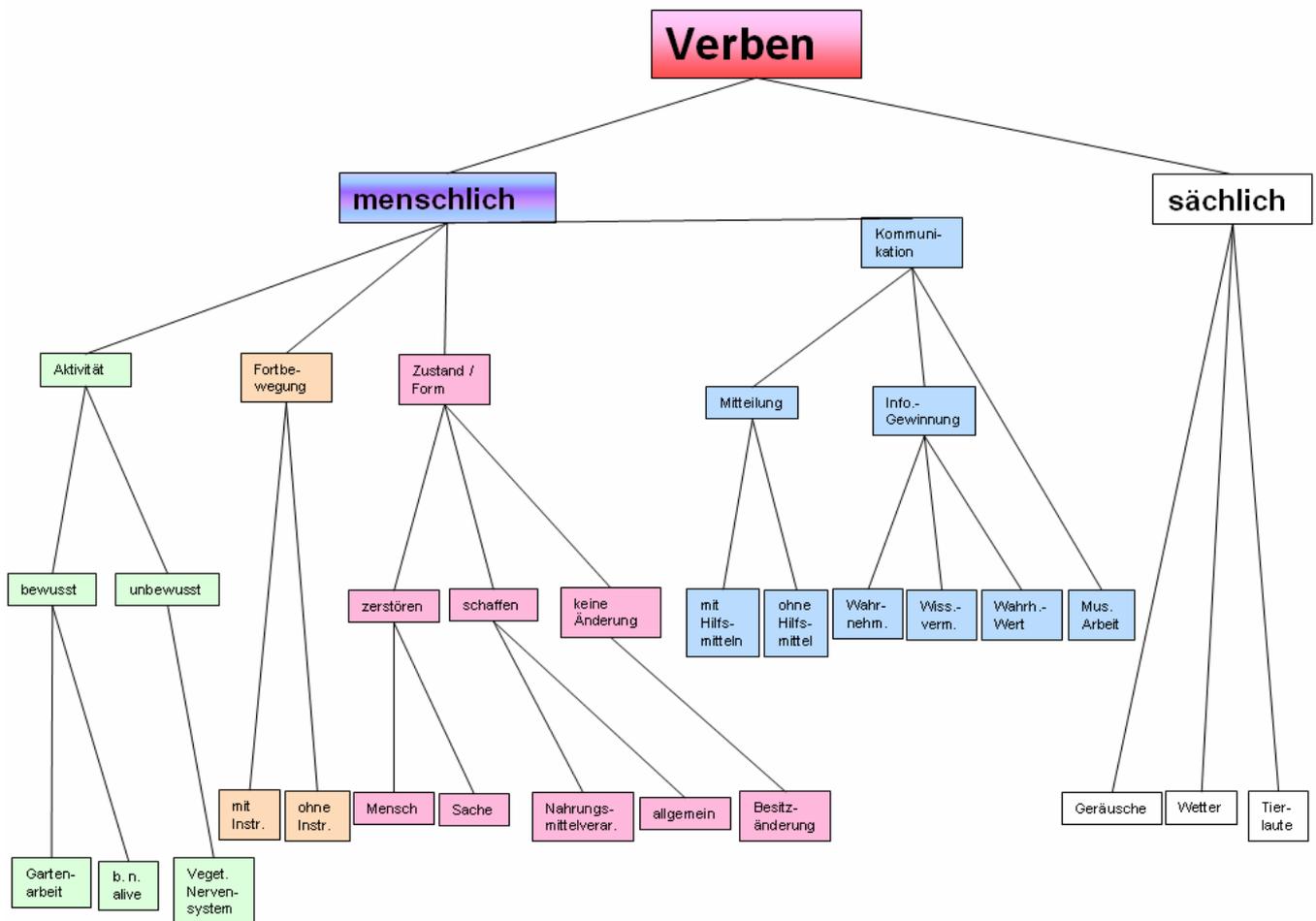
```

7.2 Muster

Kategorie	Muster	
become not alive	Subj=NP _[str]	Ich sterbe.
vegetatives Nervensystem	Subj=NP _[str]	Ich blute.
Fortbewegung mit Instrument	Subj=NP _[str] [und (dobi=NP _[str] oder Erg _{dir} oder Erg _{dir} und Pert_elm _[dativ])]	Ich fahre. Ich fahre ein altes Auto. Ich fahre nach Hause. Ich fahre ihm über den Fuss.
Fortbewegung ohne Instrument	Subj=NP _[str] [und (dobi=NP _[str] oder Erg _{dir} oder Erg _{dir} und Pert_elm _[dativ])]	Ich fahre. Ich laufe einen Marathon Ich laufe nach Hause. Er läuft mir über den Weg.
zerstören Mensch	Subj=NP _[str] und dobi=NP _[str] [und PP _{mit(NP)}]	Ich töte meinen Mann. Ich töte meinen Mann mit einem Eispickel.
zerstören Sache	Subj=NP _[str] und dobi=NP _[str] [und Pert_elm _[dativ]]	Ich demoliere ein Regal. Ich demoliere mir die Fresse.
Gartenarbeit	Subj=NP _[str] und dobi=NP _[str] [und dativus_cic]	Ich pflücke Beeren. Ich pflücke dir Blumen.
Nahrungsmittel- verarbeitung	Subj=NP _[str] und dobi=NP _[str] [und iobi=dativus_cic]	Ich backe einen Kuchen. Ich backe dir einen Kuchen.
schaffen Allgemein	Subj=NP _[str] und dobi=NP _[str] [und (Ergpräp _[aus] oder dativus_cic)]	Wir bilden einen Kreis. Ich erschaffe ein Kunstwerk aus Lehm. Ich erschaffe dir ein Kunstwerk aus Lehm.
Besitzänderung	Subj=NP _[str] und dobi=NP _[str] [und iobi=NP _[dat]]	Ich kaufe das Buch. Ich kaufe dir ein Buch.
Mitteilung mit Hilfsmittel	Subj=NP _[str] [und (dobi=NP _[str] oder iobi=NP _[str])]	Ich chatte. Ich maile dir den Hierarchie- baum.
Mitteilung ohne Hilfsmittel	Subj=NP _[str] [und iobi=NP _[dat] [und (dobi=NP _[str] oder dobi=S _{class} oder dobi=S _{que} oder dobi=S _{vII})]	Ich rede. Ich erzähle eine Geschichte. Ich verkünde (dir), dass er erhängt wird. Ich schreibe (dir), wie es mir geht. Er erzählt (dir), er sei krank gewesen.

Wahrnehmung	Subj=NP _[str] und (dobi=NP _[str] oder dobi=NP _[str] +S _{dass} oder dobi=NP _[str] +S _{que})	Ich sehe dich. Ich höre es, dass du kommst. Ich höre es, ob du kommst.
Wahrheitswert	Subj=NP _[str] und (dobi=NP _[str] oder dobi=S _{dass} oder dobi=S _{inf} oder dobi=S _{que})	Ich behaupte das Gegenteil. Ich behaupte es, dass du lügst. Ich gestehe es, dich umgebracht zu haben.
Wissensvermittlung	Subj=NP _[str] und [dobi=NP _[str] [und (dobi=S _{que} oder dobi=S _{inf})]]	Er unterrichtet. Er unterrichtet die Kinder. Er unterrichtet die Schüler, wie man liest. Er lehrt die Kinder zu lesen.
Musische Aktivitäten	Subj=NP _[str] [und (dobi=NP _[str] oder dobi+dativus_cic)]	Er komponiert. Er komponiert eine Arie. Er komponiert mir ein Lied.
Wetter	Subj=NP _[exp]	Es regnet.
Tierlaute	Subj=NP _[str] [und Erg _{art}]	Der Hund bellt. Der Hund bellt ununterbrochen.
Geräusche	Subj=(NP _[exp] NP _[str]) [und Erg _{art}]	Es knarrt. Der Boden knarrt. Das Gewinde quietscht laut.

7.3 Hierarchiebaum



7.4 Muster in Prolog-Notation

```
muster(veg_nervensystem,[atmen, bluten, husten, schwitzen],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str)]))).

muster(fortbewegung_mit_instrument,[fahren, reiten],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str)]))).

muster(fortbewegung_mit_instrument,[fahren, reiten],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str), arg_dobj(np_str)]))).

muster(fortbewegung_mit_instrument,[fahren, reiten],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str), arg_ergz(ergz_dir)]))).

muster(fortbewegung_mit_instrument,[fahren, reiten],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(dat),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str), arg_ergz(ergz_dir)]))).

muster(fortbewegung_ohne_instrument,[laufen, schwimmen, kriechen],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str)]))).

muster(fortbewegung_ohne_instrument,[laufen, schwimmen, kriechen],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ ),
    arguments([arg_subj(np_str), arg_dobj(np_str)]))).
```

```
muster(fortbewegung_ohne_instrument,[laufen, schwimmen, kriechen],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_ergz(ergz_dir)]))

muster(fortbewegung_ohne_instrument,[laufen, schwimmen, kriechen],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(dat),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_ergz(ergz_dir)]))

muster(mensch,[töten, umbringen, erwürgen, erdrosseln, erschiessen,
erstechen, ermorden, vergiften],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(mensch,[töten, umbringen, erwürgen, erdrosseln, erschiessen,
erstechen, ermorden, vergiften],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_pcmp(pp_mit_np)]))

muster(wetter,[regnen, schneien, hageln, blitzen, donnern, nieseln,
winden, stürmen],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_exp)]))

muster(gartenarbeit,[säen, pflanzen, pflücken, ernten, sammeln],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str)]))

muster(gartenarbeit,[säen, pflanzen, pflücken, ernten, sammeln],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_dobj(np_str), arg_dobj(np_str)]))
```

```
muster(gartenarbeit,[säen, pflanzen, pflücken, ernten, sammeln],
      parameters(
        deep_object( _ ),
        dativus_cic(yes),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str)]))

muster(gartenarbeit,[säen, pflanzen, pflücken, ernten, sammeln],
      parameters(
        deep_object( _ ),
        dativus_cic(yes),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(nahrungsmittelverarbeitung,[kochen, braten, dämpfen, garen,
rösten, backen, grillen, grillieren],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(nahrungsmittelverarbeitung,[kochen, braten, dämpfen, garen,
rösten, backen, grillen, grillieren],
      parameters(
        deep_object( _ ),
        dativus_cic(yes),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(become_not_alive,[sterben, ertrinken, verbluten, ersticken],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str)]))

muster(become_not_alive,[sterben, ertrinken, verbluten, ersticken],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(sache,[zerstören, abreissen, demolieren, schneiden, brechen,
reissen, zertrümmern],
      parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
      arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(sache,[zerstören, abreissen, demolieren, schneiden, brechen,
reissen, zertrümmern],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(dat),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(sache,[zerstören, abreissen, demolieren, schneiden, brechen,
reissen, zertrümmern],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(akk),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(allgemein,[bauen, erschaffen, bilden, kreieren, zusammensetzen,
zusammenfügen, schmieden, produzieren, erzeugen],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(allgemein,[ bauen, erschaffen, bilden, kreieren, zusammensetzen,
zusammenfügen, schmieden, produzieren, erzeugen],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str),
arg_pcmp([pp(prepare_austrat)])]))
```

```
muster(allgemein,[ bauen, erschaffen, bilden, kreieren, zusammensetzen,
zusammenfügen, schmieden, produzieren, erzeugen],
parameters(
  deep_object( _ ),
  dativus_cic(yes),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(allgemein,[ bauen, erschaffen, bilden, kreieren, zusammensetzen,
zusammenfügen, schmieden, produzieren, erzeugen],
parameters(
  deep_object( _ ),
  dativus_cic(yes),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str),
arg_pcmp([pp(prepare_austrat)])]))
```

```
muster(besitzaenderung,[geben, nehmen, schenken, ersitzen, kaufen,
leihen, mieten, stehlen, entreissen, entwenden, rauben, wegnehmen],
parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(besitzaenderung,[geben, nehmen, schenken, ersitzen, kaufen,
leihen, mieten, stehlen, entreissen, entwenden, rauben, wegnehmen],
parameters(
    deep_object( _ ),
    dativus_cic(yes),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(mitteilung_mit_hilfsmittel,[faxen, mailen, smsen, morsen],
parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(mitteilung_mit_hilfsmittel,[faxen, mailen, smsen, morsen],
parameters(
    deep_object( _ ),
    dativus_cic(yes),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))

muster(mitteilung_mit_hilfsmittel,[faxen, mailen, smsen, morsen],
parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str)]))

muster(mitteilung_mit_hilfsmittel,[faxen, mailen, smsen, morsen],
parameters(
    deep_object( _ ),
    dativus_cic(yes),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str)]))

muster(mitteilung_ohne_hilfsmittel,[sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
  deep_object( _ ),
  dativus_cic(yes),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_str)]))).
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_korr_dass)]))).
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
  deep_object( _ ),
  dativus_cic(yes),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_korr_dass)]))).
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_korr_que)]))).
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
  deep_object( _ ),
  dativus_cic(yes),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str), arg_dobj(np_korr_que)]))).
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
parameters(
  deep_object( _ ),
  dativus_cic(none),
  pertinenz_elm(none),
  perfekt_aux( _ )),
arguments([arg_subj(np_str),
arg_dobj(np_korr_hauptsatz)]))).
```

```
muster(mitteilung_ohne_hilfsmittel,[ sprechen, sagen, rufen, erzählen,
berichten, verkünden, reden, schreiben],
  parameters(
    deep_object( _ ),
    dativus_cic(yes),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str),
arg_dobj(np_korr_hauptsatz)]))).
```

```
muster(wahrnehmung,[sehen, riechen, schmecken, fühlen, hören],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_str)]))).
```

```
muster(wahrnehmung,[sehen, riechen, schmecken, fühlen, hören],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_korr_dass)]))).
```

```
muster(wahrnehmung,[sehen, riechen, schmecken, fühlen, hören],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_korr_inf)]))).
```

```
muster(wahrnehmung,[sehen, riechen, schmecken, fühlen, hören],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_korr_que)]))).
```

```
muster(wissensvermittlung,[lehren, unterrichten, instruieren, anlei-
ten],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str)]))).
```

```
muster(wissensvermittlung,[lehren, unterrichten, instruieren, anlei-
ten],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_str)]))).
```

```
muster(wissensvermittlung,[lehren, unterrichten, instruieren, anleiten],
```

```
    parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
    arguments([arg_subj(np_str), arg_dobj(np_korr_que)]))).
```

```
muster(wissensvermittlung,[lehren, unterrichten, instruieren, anleiten],
```

```
    parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
    arguments([arg_subj(np_str), arg_dobj(np_korr_inf)]))).
```

```
muster(musische_aktivitaeten,[komponieren, musizieren, vorführen, auf-  
führen, basteln, tanzen],
```

```
    parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
    arguments([arg_subj(np_str)]))).
```

```
muster(musische_aktivitaeten,[komponieren, musizieren, vorführen,  
aufführen, basteln, tanzen],
```

```
    parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
    arguments([arg_subj(np_str), arg_dobj(np_str)]))).
```

```
muster(musische_aktivitaeten,[komponieren, musizieren, vorführen,  
aufführen, basteln, tanzen],
```

```
    parameters(
        deep_object( _ ),
        dativus_cic(yes),
        pertinenz_elm(none),
        perfekt_aux( _ )),
    arguments([arg_subj(np_str), arg_dobj(np_str)]))).
```

```
muster(tierlaute,[bellen, brüllen, quaken, miauen, muhen, wiehern,  
fiepen, piepsen, grunzen, pfeifen, zwitschern, singen],
```

```
    parameters(
        deep_object( _ ),
        dativus_cic(none),
        pertinenz_elm(none),
        perfekt_aux( _ )),
    arguments([arg_subj(np_str)]))).
```

```
muster(geraeusche,[knistern, knacken, rascheln, quietschen, brummen,
rattern, knarren],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str)]))
```

```
muster(geraeusche,[knistern, knacken, rascheln, quietschen, brummen,
rattern, knarren],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_exp)]))
```

```
muster(wahrheitswert,[behaupten, gestehen, zugeben, leugnen, feststel-
len, widerlegen, festhalten, bezweifeln],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_str)]))
```

```
muster(wahrheitswert,[behaupten, gestehen, zugeben, leugnen, feststel-
len, widerlegen, festhalten, bezweifeln],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_korr_dass)]))
```

```
muster(wahrheitswert,[behaupten, gestehen, zugeben, leugnen, feststel-
len, widerlegen, festhalten, bezweifeln],
  parameters(
    deep_object( _ ),
    dativus_cic(none),
    pertinenz_elm(none),
    perfekt_aux( _ )),
  arguments([arg_subj(np_str), arg_dobj(np_korr_inf)]))
```

7.5 Programm zur Evaluation

```
/* Dieses Programm gibt alle Verben aus, die mit unseren Mustern */
/* im Lexikon gefunden wurden und auch in unserem Programm als */
/* Beispielverben aufgefuehrt sind. Achtung: Vor dem Aufruf */
/* 'menu.' muss das Lexikon konsultiert werden. */

node( verben, 'verben' ).
daughter( verben, 1, menschlich ).
daughter( verben, 2, saechlich ).

node( menschlich, 'menschlich' ).
daughter( menschlich, 1, aktivitaet ).
daughter( menschlich, 2, fortbewegung ).
daughter( menschlich, 3, zustandform ).
daughter( menschlich, 4, kommunikation ).

node( saechlich, 'saechlich' ).
daughter( saechlich, 1, geraeusche ).
daughter( saechlich, 2, wetter ).
daughter( saechlich, 3, tierlaute ).

    node( geraeusche, 'Geraeusche' ).
    node( wetter, 'Wetter' ).
    node( tierlaute, 'Tierlaute' ).

node( aktivitaet, 'Aktivitaet' ).
daughter( aktivitaet, 1, bewusst ).
daughter( aktivitaet, 2, unbewusst ).

    node( bewusst, 'bewusst' ).
    daughter( bewusst, 1, gartenarbeit ).
    daughter( bewusst, 2, become_not_alive ).

        node( gartenarbeit, 'Gartenarbeit' ).
        node( become_not_alive, 'become not alive' ).

node( unbewusst, 'unbewusst' ).
daughter( unbewusst, 1, veg_nervensystem ).

    node( veg_nervensystem, 'Vegetatives Nervensystem' ).

node( fortbewegung, 'Fortbewegung' ).
daughter( fortbewegung, 1, fortbewegung_mit_instrument ).
daughter( fortbewegung, 2, fortbewegung_ohne_instrument ).

    node( fortbewegung_mit_instrument, 'mit Instrument' ).
    node( fortbewegung_ohne_instrument, 'ohne Instrument' ).

node( zustandform, 'Zustand / Form' ).
daughter( zustandform, 1, zerstoeren ).
daughter( zustandform, 2, schaffen ).
daughter( zustandform, 3, keine_aenderung ).

    node( zerstoeren, 'Zerstoeren' ).
    daughter( zerstoeren, 1, mensch ).
    daughter( zerstoeren, 2, sache ).
```

```

node( mensch, 'Mensch').
node( sache, 'Sache').

node( schaffen, 'Schaffen').
daughter( schaffen, 1, nahrungsmittelverarbeitung ).
daughter( schaffen, 2, allgemein ).

node( nahrungsmittelverarbeitung, 'Nahrungsmittelverarbeitung').
node( allgemein, 'Allgemein').

node( keine_aenderung, 'keine Aenderung').
daughter( keine_aenderung, 1, besitztaenderung ).

node( besitztaenderung, 'Besitztaenderung').

node( kommunikation, 'Kommunikation' ).
daughter( kommunikation, 1, mitteilung ).
daughter( kommunikation, 2, info_gewinnung ).
daughter( kommunikation, 3, musische_aktivitaeten ).

node( musische_aktivitaeten, 'Musische Aktivitaeten').

node( mitteilung, 'Mitteilung' ).
daughter( mitteilung, 1, mitteilung_mit_hilfsmittel).
daughter( mitteilung, 2, mitteilung_ohne_hilfsmittel).

node( mitteilung_mit_hilfsmittel, 'mit Hilfsmittel').
node( mitteilung_ohne_hilfsmittel, 'ohne Hilfsmittel').

node( info_gewinnung, 'Informationsgewinnung' ).
daughter( info_gewinnung, 1, wahrnehmung).
daughter( info_gewinnung, 2, wissensvermittlung).
daughter( info_gewinnung, 3, wahrheitswert).

node( wahrnehmung, 'Wahrnehmung').
node( wissensvermittlung, 'Wissensvermittlung').
node( wahrheitswert, 'Wahrheitswert').

/* ***** Muster ***** */

Hier stehen die Muster in Prolog-Notation drin. Code siehe Kapitel
7.4!

/* ***** Ende Muster ***** */

menu :- menu( verben ).

menu( Node ) :-
    daughter( Node, _, _ ),
    !,
    displayChoice( Node ),
    write('Auswahl: '),
    read( Input ),nl,
    daughter( Node, Input, Daughter ),
    menu( Daughter ).

```

```
menu( Node ) :-
muster( Node, Beispielverben, Parameters, Arguments),
write('*****'),nl,
write( Parameters ),write(' '),write(Arguments),nl,
write_lexikoneintrag(Node,Beispielverben,Parameters,Arguments),nl,
fail.

menu(Node) .

member(X, [X|_]).
member(X, [_|Rest]):-
member(X, Rest).

write_lexikoneintrag(Node,Beispielverben,Parameters,Arguments):-
verb(ID,Parameters,Arguments,_),
member(ID,Beispielverben),
write(ID),write(' '),
fail.

write_lexikoneintrag(Node,Beispielverben,Parameters,Arguments):-
nl,write(Beispielverben),nl.

displayChoice( Node ) :-
node( Node, Text ),
write( Text ), write(': '), nl,
not ( daughter( Node, N, D ),
node( D, DText ),
write( N ), write(': '), write( DText ), nl,
fail
).

/* Schreibt die Ausgabe falls gewünscht in Dateien. */
/*Achtung: Die Eingabe durch den Benutzer muss im */
/*Blindflug erfolgen... Have even more fun :-)) */

tofile(Filename):-
tell(Filename),
menu,
told.
```

7.6 Programm zum Abarbeiten der Kategorien und Muster

```
/* Dieses Programm gibt dem Benutzer anhand der Eingaben alle */
/* moeglichen Muster eines Verbes an, das sich in einer */
/* bestimmten Kategorie befindet. mit 'menu.' wird das */
/* Programm aufgerufen. Have fun! */

/* *****Baumhierarchie***** */

node( verben, 'verben' ).
daughter( verben, 1, menschlich ).
daughter( verben, 2, saechlich ).

node( menschlich, 'menschlich' ).
daughter( menschlich, 1, aktivitaet ).
daughter( menschlich, 2, fortbewegung ).
daughter( menschlich, 3, zustandform ).
daughter( menschlich, 4, kommunikation ).

node( saechlich, 'saechlich' ).
daughter( saechlich, 1, geraeusche ).
daughter( saechlich, 2, wetter ).
daughter( saechlich, 3, tierlaute ).

    node( geraeusche, 'Geraeusche' ).
    node( wetter, 'Wetter' ).
    node( tierlaute, 'Tierlaute' ).

node( aktivitaet, 'Aktivitaet' ).
daughter( aktivitaet, 1, bewusst ).
daughter( aktivitaet, 2, unbewusst ).

    node( bewusst, 'bewusst' ).
    daughter( bewusst, 1, gartenarbeit ).
    daughter( bewusst, 2, become_not_alive ).

        node( gartenarbeit, 'Gartenarbeit' ).
        node( become_not_alive, 'become not alive' ).

node( unbewusst, 'unbewusst' ).
daughter( unbewusst, 1, veg_nervensystem ).

    node( veg_nervensystem, 'Vegetatives Nervensystem' ).

node( fortbewegung, 'Fortbewegung' ).
daughter( fortbewegung, 1, fortbewegung_mit_instrument ).
daughter( fortbewegung, 2, fortbewegung_ohne_instrument ).

    node( fortbewegung_mit_instrument, 'mit Instrument' ).
    node( fortbewegung_ohne_instrument, 'ohne Instrument' ).

node( zustandform, 'Zustand / Form' ).
daughter( zustandform, 1, zerstoeren ).
daughter( zustandform, 2, schaffen ).
daughter( zustandform, 3, keine_aenderung ).
```

```

node( zerstoeren, 'Zerstoeren').
daughter( zerstoeren, 1, mensch ).
daughter( zerstoeren, 2, sache ).

    node( mensch, 'Mensch').
    node( sache, 'Sache').

node( schaffen, 'Schaffen').
daughter( schaffen, 1, nahrungsmittelverarbeitung ).
daughter( schaffen, 2, allgemein ).

    node( nahrungsmittelverarbeitung, 'Nahrungsmittelverarbeitung').
    node( allgemein, 'Allgemein').

node( keine_aenderung, 'keine Aenderung').
daughter( keine_aenderung, 1, besitz_aenderung ).

    node( besitz_aenderung, 'Besitz_aenderung').

node( kommunikation, 'Kommunikation' ).
daughter( kommunikation, 1, mitteilung ).
daughter( kommunikation, 2, info_gewinnung ).
daughter( kommunikation, 3, musische_aktivitaeten ).

    node( musische_aktivitaeten, 'Musische Aktivitaeten').

    node( mitteilung, 'Mitteilung' ).
    daughter( mitteilung, 1, mitteilung_mit_hilfsmittel).
    daughter( mitteilung, 2, mitteilung_ohne_hilfsmittel).

        node( mitteilung_mit_hilfsmittel, 'mit Hilfsmittel').
        node( mitteilung_ohne_hilfsmittel, 'ohne Hilfsmittel').

node( info_gewinnung, 'Informationsgewinnung' ).
daughter( info_gewinnung, 1, wahrnehmung).
daughter( info_gewinnung, 2, wissensvermittlung).
daughter( info_gewinnung, 3, wahrheitswert).

    node( wahrnehmung, 'Wahrnehmung').
    node( wissensvermittlung, 'Wissensvermittlung').
    node( wahrheitswert, 'Wahrheitswert').

/* ***** Muster ***** */

Hier stehen die Muster in Prolog-Notation drin. Code siehe Kapitel
7.4!

/* ***** Ende Muster ***** */

/* *****Eingabemenu***** */
menu :- menu( verben ).

menu( Node ) :-
    daughter( Node, _, _ ),
    !,
    displayChoice( Node ),
    write('Auswahl: '),

```

```
read( Input ),nl,  
daughter( Node, Input, Daughter ),  
menu( Daughter ).
```

```
menu( Node ) :-  
muster( Node, Beispielverben, Parameters, Arguments),  
write('*****'),nl,  
write( Parameters ),write(' '),write(Arguments),nl,  
fail.
```

```
menu(Node):-  
muster( Node, Beispielverben, _, _),  
write('*****'), nl,  
write('Beispiele: '), write(Beispielverben),nl.
```

```
displayChoice( Node ) :-  
node( Node, Text ),  
write( Text ), write(': '), nl,  
not ( daughter( Node, N, D ),  
node( D, DText ),  
write( N ), write(': '), write( DText ), nl,  
fail  
).
```