

Sprachtechnologie

Ein Überblick

Kai-Uwe Carstensen



Sprachtechnologie

Ein Überblick

Kai-Uwe Carstensen

Homepage:

<http://www.kai-uwe-carstensen.de>

URL dieses PDF:

<http://www.kai-uwe-carstensen.de/Publikationen/Sprachtechnologie.pdf>

Email:

contact at kai-uwe-carstensen.de

Version 2.2
23. April 2017

© 2017 Kai-Uwe Carstensen



Der Text und die mit LaTeX erstellten Grafiken sind bis einschließlich der Version 2.2 unter einem Creative Commons Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung-3.0 Lizenzvertrag lizenziert. Siehe dazu <http://creativecommons.org/licenses/by-nc-nd/3.0/deed.de>. Zur Nennung meines Namens gehört ein Verweis auf meine Homepage (<http://www.kai-uwe-carstensen.de>).

Inhaltsverzeichnis

1 Einleitung	1
2 Sprachtechnologie: Grundlegendes	7
2.1 Allgemeines zur Sprachtechnologie: Unterscheidungen	8
2.1.1 Terminologie der beteiligten Disziplinen	8
2.1.2 Computerlinguistik vs. Sprachtechnologie	8
2.1.3 Tiefe/flache Verarbeitung	13
2.1.4 Gesprochene/geschriebene Sprache	13
2.2 Sprachtechnologie: Grobcharakterisierung der Anwendungen	13
2.2.1 Ausdrucksform	14
2.2.2 Komplexität	16
2.2.3 Funktionalität	17
2.3 Ausblick	18
2.4 Literatur	20
3 Natürlichsprachliche Systeme (NLS)	23
3.1 Allgemeines zu NLS	24
3.2 Geschichte	25
3.3 Aspekte und Anforderungen	29
3.3.1 Qualitätskriterien	29
3.3.2 Probleme maschineller Sprachverarbeitung	32
3.4 Architekturen und Systeme	38
3.4.1 Systeme mit tiefer Verarbeitung	38
3.4.2 Systeme mit flacher Verarbeitung	41
3.4.3 Andere Architekturen	44
3.5 Ausblick	45
3.6 Literatur	47
4 Natürlichsprachliche Suche	49
4.1 Allgemeines zur natürlichsprachlichen Suche	50
4.2 Geschichte	51
4.3 Aspekte und Anforderungen	54
4.4 Literatur	56

5 Informationsextraktion (IE)	57
5.1 Allgemeines zur IE	58
5.2 Geschichte	59
5.3 Aspekte und Anforderungen	62
5.4 Architekturen und Systeme	64
5.4.1 FASTUS	64
5.4.2 GATE	65
5.4.3 KNOWITALL und TEXTRUNNER	66
5.5 Evaluation: Die Message Understanding Conference (MUC)	69
5.6 Literatur	72
6 Natürlichsprachliche Schnittstellen zu Datenbanken (NLIDB)	73
6.1 Allgemeines zu NLIDB	74
6.2 Geschichte	76
6.3 Aspekte und Anforderungen	80
6.3.1 Probleme	80
6.3.2 Anforderungen	82
6.4 Architekturen und Systeme	84
6.4.1 Grobarchitektur von NLIDB	84
6.4.2 Designempfehlungen für NLIDBs	84
6.4.3 STEP	85
6.4.4 PRECISE	88
6.5 Ausblick	89
6.6 Literatur	91
7 Frage-/Antwort-Systeme (QAS)	93
7.1 Allgemeines zu QAS	94
7.2 Geschichte	94
7.3 Aspekte und Anforderungen	95
7.3.1 Allgemeines	95
7.3.2 Open domain QA	99
7.4 Architekturen und Systeme	101
7.4.1 WebQAS	101
7.4.2 EXTRANS: Wissensbasierte Antwortextraktion	108
7.4.3 DeepQA	110
7.5 Evaluation: TREC-QA	110
7.6 Ausblick	112
7.7 Literatur	114
8 Dialogsysteme (DS)	115
8.1 Allgemeines zu DS	116
8.1.1 Funktionalität von DS	116

8.1.2	Grobklassifikation von DS	117
8.2	Aspekte und Anforderungen	119
8.3	DS-Architektur	121
8.3.1	Überblick	121
8.3.2	Dialogsteuerung/-management	123
8.3.3	Dialogmodellierung	124
8.4	Architekturen und Systeme	128
8.4.1	Konversationelle Agenten (TRAIN S, TRIPS)	128
8.4.2	Rationale Interaktion (ARTIMIS)	132
8.4.3	Der Information State Ansatz (TRINDI)	133
8.5	Literatur	137
9	Sprachdialogsysteme (SDS)	139
9.1	Allgemeines zu SDS	140
9.2	Geschichte	141
9.3	Aspekte und Anforderungen	142
9.3.1	Allgemeines	142
9.3.2	Technische Leistungsmerkmale	143
9.3.3	Wirtschaftlichkeit	144
9.3.4	Voice User Interface	145
9.4	Evaluation	146
9.5	Ausblick	146
9.6	Literatur	147
10	Multimodale Systeme (MMS)	149
10.1	Allgemeines zu MMS	150
10.2	Geschichte	152
10.3	Aspekte und Anforderungen	154
10.3.1	Multimodal fusion	154
10.3.2	Multimodal fission	156
10.3.3	Kommunikatives Verhalten	157
10.4	Architekturen und Systeme	158
10.4.1	SMARTKOM	159
10.4.2	REA	161
10.4.3	MAX	162
10.5	Literatur	165
11	Sprachgenerierung (NLG)	167
11.1	Allgemeines zur NLG	168
11.2	Geschichte	169
11.3	Aspekte und Anforderungen	173
11.4	Architekturen und Systeme	175

INHALTSVERZEICHNIS

11.5 Evaluation	176
11.6 Literatur	178
12 Maschinelle Übersetzung (MÜ)	181
12.1 Allgemeines zur MÜ: Mythen und ihre Klärung	182
12.2 Geschichte	184
12.3 Aspekte und Anforderungen	186
12.4 Architekturen und Systeme	189
12.4.1 Verarbeitungstiefe	189
12.4.2 Verwendete Techniken	192
12.5 Evaluation	194
12.6 Ausblick	194
12.7 Literatur	196
13 Textzusammenfassungssysteme (TZS)	197
13.1 Allgemeines zu Textzusammenfassungen	198
13.2 Geschichte	201
13.3 Architektur, Ansätze und Methoden	202
13.4 Systeme	203
13.5 Evaluation: Die Document Understanding Conferences (DUC)	205
13.6 Ausblick	207
13.7 Literatur	207
14 Fazit	211
15 Epilog	213
Literaturverzeichnis	215
Index	233

Kapitel 1

Einleitung

Seit mehr als einem halben Jahrhundert existieren ernsthafte und ernst zu nehmende Versuche, menschliche Sprache maschinell zu verarbeiten. Maschinelle Übersetzung oder „natürliche“ Dialoge mit Computern gehören zu den ersten Ideen, die den Bereich der späteren *Computerlinguistik* oder *Sprachtechnologie* abgesteckt und deren Vorhaben geleitet haben.

Heute ist dieser auch *maschinelle Sprachverarbeitung* (*natural language processing*, *NLP*) genannte Bereich stark ausdiversifiziert: Durch die rapide Entwicklung der Informatik ist vieles vorher Unvorstellbare Realität (z. B. automatische Telefonauskunft), einiges früher Unmögliche immerhin möglich geworden (z. B. Handhelds mit Sprachein- und -ausgabe als digitale persönliche (Informations-)Assistenten). Es gibt verschiedene Anwendungen der Computerlinguistik, von denen einige den Sprung in die kommerzielle Nutzung geschafft haben (z. B. Diktiersysteme, Textklassifikation, maschinelle Übersetzung).

Immer noch wird an *natürlichsprachlichen Systemen* (*natural language systems*, *NLS*)¹ verschiedenster Funktionalität (z. B. zur Beantwortung beliebiger Fragen oder zur Generierung komplexer Texte) intensiv geforscht, auch wenn die hoch gesteckten Ziele von einst längst nicht erreicht sind (und deshalb entsprechend „heruntergefahren“ wurden). Wo die maschinelle Sprachverarbeitung heute steht, ist allerdings angesichts der vielfältigen Aktivitäten in der Computerlinguistik und Sprachtechnologie weder offensichtlich noch leicht in Erfahrung zu bringen (für Studierende des Fachs und erst recht für Laien). Wie wichtig dies aber für die Außendarstellung der Sprachverarbeitungsbranche und die Computerlinguistik als Disziplin ist, fasst Steedman in folgende Worte: „It [...] means telling the public in honest terms how to think about what we do, what can be relied on and what the really hard problems are, in good time and bad“ (Steedman 2008, S. 144).

Der **Grund für die Existenz dieses Buches** ist entsprechend schnell erklärt: Für eine Vorlesung zur Sprachtechnologie, die ich 2005 und 2007 an der Universität Freiburg (und später: den Unis Bochum und Siegen) hielt, fand ich keine geeignete, einen Überblick liefernde Literatur. Zwar gibt es einige Einführungs- und Handbücher, die das „wie“ der Computerlinguistik und Sprachtechnologie in unterschiedlicher Breite und Tiefe erklären. Mit der Ausnahme von Sprachtechnologie-orientierten Büchern wie Cole et al. (1997) und Oertle (2006) existiert

¹ Zum Teil werden solche komplexen Softwaresysteme zur Sprachverarbeitung auch „Sprachapplikation“ genannt

jedoch wenig Aktuelles dazu, „was“ in diesem Bereich geleistet werden kann und soll, was prinzipiell möglich oder unmöglich bzw. was realisierbar oder schon realisiert ist.

So werden in *Carstensen et al.* (2009) die Anwendungsgebiete der Computerlinguistik schlaglichtartig beleuchtet, jedoch können die Gemeinsamkeiten und Unterschiede sprachtechnologischer Software auf diese Weise kaum sichtbar gemacht werden. Es zeigt sich, dass die „System“ebene in der Computerlinguistik jenseits von Grundlagen, Theorien und Methoden kaum systematisch behandelt wird.

Dies betrifft erstens allgemeine Aspekte wie die Funktionalität und Architektur solcher Systeme (Was leistet ein System vom Typ X und wie ist es aufgebaut?); zweitens die prinzipiell zu behandelnden und aktuell behandelten Phänomene selbst, deren Beherrschung die Kompetenz des Systems ausmachen (mit anderen Worten: warum ist es überhaupt so schwierig, sprachtechnologische Anwendungen zu entwickeln?); drittens die vergangenen und zukünftigen Entwicklungen in dem jeweiligen Bereich (die es beispielsweise erlauben, Investitionen in diesem Bereich zu planen). Indem es diese unterschiedlichen Aspekte behandelt, geht das vorliegende Buch auch weit über *Oertle* (2006) hinaus, das sich im Wesentlichen auf die reine Darstellung computerlinguistischer Anwendungen beschränkt.

Ziele und Adressaten des Buchs

Ein Ziel dieses Buches ist, es, die aktuelle Literaturlage in dieser Hinsicht zu verbessern, indem spezifisch systembezogene Aspekte der Computerlinguistik als Überblick über die Sprachtechnologie zusammengetragen werden.

Eine weitere Motivation, dieses Buch zu schreiben, entstammt der von mir oft beobachteten mangelnden Informiertheit potenzieller Anwender maschineller Sprachverarbeitung in Bezug auf den Stand der Kunst in der Computerlinguistik bzw. auf die Leistungsfähigkeit ihrer Systeme. Dies widerspricht den Vorteilen, die die Sprachtechnologie ihren Anwendern bringen würde: „[...] personal experience suggests that many companies would benefit from the application of NLP skills but are not aware of this“ (*Dale et al.* 2003).

Skepsis gegenüber oder Unverständnis bzgl. der Sprachtechnologie (s. Abb. 1.1) basiert in der Regel auf einer Mischung von Informationen und Fehlinformationen aus den Medien, überzogenen Erwartungen, Fehleinschätzungen, eigenen enttäuschenden Erfahrungen, kursierenden Gerüchten, Mythen usw. im Hinblick darauf, was mit Sprachtechnologie machbar ist oder nicht. So kennen laut einer Studie des Fraunhofer-Instituts für Arbeitswirtschaft und Organisation (IAO) zur Akzeptanz von Sprachapplikationen² nur 54% der Befragten überhaupt sprachtechnologische Anwendungen. Nach *Horndasch et al.* (2005) halten die Autoren der IAO-Studie daher umfassende Öffentlichkeitsarbeit (und gute Beispielapplikationen) für notwendig. Allerdings hat sich hier in den letzten Monaten/Jahren viel getan. Den wohl markantesten Einschnitt in Bezug auf die Nutzung, Wahrnehmung und Akzeptanz von Sprachtechnologie wird wohl die iPhone-Komponente Siri bewirken, da sie praktische Relevanz mit großer Verbreitung koppelt und ein Vorreiter für ähnliche Software-Komponenten in anderen Systemen sein wird.

Die Erfahrung zeigt zudem, dass der Informationsfluss zwischen Technologieentwicklern

² M. Peissner, J. Biesterfeldt, F. Heidmann, *Akzeptanz und Usability von Sprachapplikationen in Deutschland*. Technische Studie, Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO).



aus: V. Fromkin, R. Rodman, N. Hyams (2003), *An Introduction to Language* (Boston: Thomson, Heinle), 7. ed., S. 429.

Abbildung 1.1: **Die Haltung eines Laien zur Sprachverarbeitung – eine Karikatur.**

und -anwenden in diesem Bereich noch verbesserungswürdig ist, denn auch die Entwickler sind oft unterinformiert. Ihnen fehlt in der Regel Bedarfsinformation: Was wollen oder brauchen Anwender? Was sind ihre Probleme? Können diese gelöst werden? Wenn ja, innerhalb welchen Zeitrahmens und unter welchen sinnvollen Einschränkungen (z. B. durch Verwendung restrukturierter Sprache, für spezifische Einsatzzwecke, in eingeschränkten Domänen)?

Das zweite Ziel dieses Buches ist es daher zu zeigen, dass die Sprachtechnologie zwar keine „ultimativen“ Lösungen – z. B. im Bereich „maschinelle Übersetzung“ – entwickeln kann (das ist, wenn überhaupt, auf lange Zeit noch unmöglich) und auch nicht will, dass sie die Entwicklung machbarer und bedarfsorientierter Anwendungen aber mittlerweile systematisch betreibt.

Die Adressaten dieses Buches sind somit sowohl Studierende der Computerlinguistik/ Sprachtechnologie als auch interessierte Laien und potentielle Anwender. Ihnen wird ein Überblick über das Gebiet computerlinguistischer Anwendungen gegeben, ohne auf methodische und theoretische Details eingehen zu wollen und zu müssen (empfehlenswerte Einführungen hierzu sind Carstensen et al. 2009, Cole et al. 1997, Dale et al. 2000, Indurkhya und Damerau 2010, Jurafsky und Martin 2008, Lobin und Lemnitzer 2004, Lobin 2009, Mitkov 2003). Stattdessen erhalten die Studierenden einen Überblick über die Einsatzmöglichkeiten der Methoden, die ihnen vermittelt werden/wurden. Laien hingegen bekommen einen Einblick in die Anwendungsmöglichkeiten und -grenzen der Sprachtechnologie.

Terminologisches

Maschinelle Sprachverarbeitung ist ein heterogenes interdisziplinäres Gebiet. Sie lässt sich in die linguistisch orientierte *theoretische Computerlinguistik* i.e.S., die den Philologien zu-

Stuttgart, 2004

zuordnende *Linguistische Datenverarbeitung*³, das kognitionsorientierte *Natural Language Processing* und die praxisorientierte *Sprachtechnologie* (s. hierzu Carstensen et al. 2009:2f) unterteilen.

Dieser Subklassifizierung streng zu folgen ist aber hinderlich bei der Entwicklung natürlichsprachlicher Systeme: Zum Einen sind die Grenzen der einzelnen Bereiche sowieso fließend; zum Anderen liefert jeder Bereich letztendlich einen sinnvollen und notwendigen Beitrag in der Computerlinguistik. Wer hier kurzsichtige Abgrenzungen propagiert (z. B.: „Sprachtechnologie sollte ausschließlich von Software-Ingenieuren entwickelt werden“ o. ä.) verhindert gerade den notwendigen Informationsfluss und somit Innovation.

Wenn also im Folgenden von Sprachtechnologie die Rede ist, so ist damit keineswegs nur der eingeschränkte Bereich der Entwicklung von Sprachsoftware gemeint, sondern eher der gesamte Bereich der maschinellen Sprachverarbeitung aus der Anwendungsperspektive. Analog dazu: wenn von Computerlinguistik gesprochen wird, so bezeichnet dies nicht ausschließlich den Elfenbeinturm der Sprachverarbeitung an den Universitäten, sondern ist eher als synonym zum gesamten Bereich der maschinellen Sprachverarbeitung anzusehen.

Zur spezifischen **Terminologie** in diesem Buch: Manchmal ist es schwierig, oft unmöglich, ohne englische Termini auszukommen (z. B. existiert der Terminus *Speech* für „gesprochene Sprache“ im Englischen und ist somit viel prägnanter als Kontrast zu *Text*). Außerdem trifft es immer noch zu, dass der angloamerikanische Bereich und Englisch als Wissenschaftssprache dominieren. Ich werde deshalb nicht versuchen, auf entsprechende englische Termini und Kürzel zu verzichten.

Was das Buch nicht abdeckt/nicht leisten kann

Dieses Buch ist wie erwähnt nicht als eine weitere Einführung in die Computerlinguistik gedacht. Linguistische und informatische Grundlagen sowie die für das Fach konstitutiven Methoden werden hier, wenn nicht vorausgesetzt, so doch weitgehend ausgeblendet. Es kann und soll auch nicht alle möglichen Anwendungen der Computerlinguistik aufzählen, sondern setzt einen Schwerpunkt auf die Darstellung komplexerer natürlichsprachlicher Systeme. Die Breite des Bereichs dieser Anwendungen wird im folgenden Überblickskapitel skizziert, auch wird dort die hier vorgenommene Einschränkung auf die Betrachtung natürlichsprachlicher Systeme nachvollziehbar werden.

Dieses Buch kann und will außerdem kein Überblick über den aktuellen Stand der Sprachtechnologie sein (d.h., eine Auflistung entsprechender Firmen und Systeme). Das ist einerseits nicht möglich (Sprachtechnologie ist mittlerweile ein milliardenschwerer IT-Sektor) und auch nicht sinnvoll (die Fluktuation der Firmen ist sehr hoch und die tatsächliche Performanz der Systeme meist schwer einschätzbar). Eine solche Aufgabe muss kleineren, aktuelleren (Einzel-)Studien überlassen werden.

In diesem Zusammenhang ist es übrigens einerseits interessant, andererseits frustrierend zu beobachten, wie schnell Firmen im Sprachtechnologiesektor gegründet werden und dann

³ Dieser Terminus ist allerdings veraltet: Vor kurzem wurde die Gesellschaft für Linguistische Datenverarbeitung in „Gesellschaft für Sprachtechnologie und Computerlinguistik“ (GSCL) umbenannt.

wieder vom Markt verschwinden. Die Ursache hierfür ist in der betriebswirtschaftlichen Realität computerlinguistischer Firmengründungen zu suchen: Zwar gibt es heute eine große Anzahl von „Spin-offs“ universitärer Forschungsprojekte in Form von Firmen, die von Ex-Mitarbeitern solcher Projekte gegründet werden. Allerdings werden diese Start-ups in der Regel von Geldgebern nur anfinanziert, eine Geldquelle, die nach kurzer Zeit wieder versiegt. Bis dahin schaffen es lange nicht alle dieser Firmen, den Break-even-Point zu erreichen und somit auf eigenen Beinen zu stehen – u. a. aufgrund des Problems der Akquirierung möglicher Kunden, das nicht selten aus der oben beschriebenen mangelnden Informiertheit über aktuelle Sprachtechnologie resultiert.

Was das Buch leisten kann/will

Grundsätzlich versteht sich dieses Buch als eine Art **Meta-Überblick** (bildlich gesprochen: die Spitze der Spitze des Eisbergs) über den Bereich der *Natürlichsprachlichen Systeme*, da oft Überblicksartikel über einzelne Anwendungsgebiete die wesentliche Grundlage bilden. Ein solches Vorgehen ist notwendigerweise selektiv und bei dieser Auswahl subjektiv, wofür ich beim Leser um Verständnis bitte. Diese Selektion ist aber in jedem Fall Prinzip: da an jeder Stelle beliebig detaillierter und ausladender dargestellt werden könnte, würde sich schnell eine Informationsmenge ergeben, die eher hinderlich ist als dass sie nützt.

In den folgenden Kapiteln werden die Fortschritte und Perspektiven –aber auch die Grenzen– der maschinellen Sprachverarbeitung doch recht deutlich aufgezeigt werden. Es wird sich zeigen, dass der flüssig kommunizierende, mit Leichtigkeit in andere Sprachen übersetzende Computer noch Utopie ist, dass aber insbesondere aufgrund des technischen Fortschritts heute sprachtechnologische Software entwickelt wird, die zum Teil nicht nur erstaunt, sondern mittlerweile auch lukrativ ist.

Der Überblick hat im Übrigen mittlerweile vorwiegend historischen Wert. Zur Einordnung in aktuelle Entwicklungen s. vor allem das Fazit und den Epilog.

Zur Lektüre

Die Kapitel des Buches sind weitestgehend einheitlich aufgebaut und beginnen jeweils mit einer kurzen, groben Charakterisierung des behandelten Themenbereichs. Erst im Verlauf der folgenden Unterkapitel (Allgemeines, Geschichte, Aspekte und Anforderungen, Architekturen und Systeme, Evaluation) werden die Inhalte zunehmend spezifischer, so dass z. B. Laien, die vorzeitig aus der Lektüre „aussteigen“, zwar Details verpassen, aber trotzdem einen gewissen Überblick über den Bereich erlangt haben. Vereinzelt wird vertiefende Information (zu bestimmten Phänomenen oder Systemen) in Form kurzer Exkurse angeboten. Jedes Kapitel schließt mit Hinweisen auf grundlegende (bzw. vertiefende und weiterführende) Lektüre.

Ich habe mich entschieden, Weblinks in der Mehrzahl der Fälle nicht mehr explizit aufzuführen, sondern durch Unterstreichung ihrer Beschreibungen zu kennzeichnen.

Der Vorteil einer Bildschirmlektüre eines Pdf-Dokuments wie diesem ist das schnelle Verfolgen von Hyperlinks. Dieses klappt bei Querverweisen innerhalb des Dokuments allerdings nicht immer (so dass manchmal zu einer anderen Seite, dem Anfang des Abschnitts, gesprungen wird). Ich bitte das zu entschuldigen.

Zur Literatur

Die relativ schlechte Literaturlage zur Thematik, die Schnelllebigkeit der Branche sowie der Wunsch nach Aktualität bewirken, dass ich oft auf Quellen aus dem Web zurückgreife bzw. auf solche verlinke. Ich verzichte allerdings, auch wenn dies formal schlechter Stil ist, in den entsprechenden (Literatur-)Angaben generell auf die Kennzeichnung des letzten Besuchs der Webseite. Soweit nicht speziell markiert, ist also von einer Gültigkeit der Links zumindest im Zeitraum 2005-2009 auszugehen. Generell versuche ich, die Links auf Webseiten sprachtechnologischer Systeme, Firmen o. Ä. aktuell zu halten und ggf. zu entfernen. Ich distanziere mich ausdrücklich von geänderten, nicht im Bereich der Sprachtechnologie angesiedelten Inhalten.

Auf ein Glossar computerlinguistischer/sprachtechnologischer Termini habe ich verzichtet. Ich hoffe, dass unbekannte Ausdrücke entweder im Kontext verständlich sind bzw. (z. B. per Fußnote) gemacht werden. Sollte dies im Einzelfall nicht zutreffen, empfehle ich das im Internet verfügbare Glossar des Instituts für Computerlinguistik der Universität Zürich.

Danksagung

Ich danke Konrad Feldmeier, Zeno Gantner, Michael Hess, Antony Jameson, Susanne Jekat, Ingrid Renz, Ipke Wachsmuth und den Studenten meiner Veranstaltungen zur Sprachtechnologie für Informationen und hilfreiches Feedback. Besonderer Dank geht an Georg Rehm für wertvolle Kommentare zu einer Vorversion dieses Buchs.

Dank geht auch an Till Tantau für PGF und damit die Möglichkeit, „pretty good functional“e LaTeX-Grafiken zu erstellen (die Alternative waren/wären gewesen die ausschließliche Verwendung Platz fressender hässlicher Bild-Kopien).

Kapitel 2

Sprachtechnologie: Grundlegendes

Die Sprachtechnologie befasst sich mit der Entwicklung marktreifer Anwendungen der **maschinellen Sprachverarbeitung (Computerlinguistik, CL)**. Ein paar reale (!) Szenarien sollen andeuten, was hierunter zu verstehen ist:

- Ein Richter am Amtsgericht diktiert kurze Beschlüsse direkt in seinen PC. Funktionierende Spracherkennung erlaubt es ihm, den Text sofort zu überblicken und ggf. Details zu ergänzen – im Gegensatz zu einem Arbeitsablauf, der Tonband und Sekretariat involviert.^a
- Ein Sitzungsprotokoll eines spanischen EG-Abgeordneten, abgefasst in seiner Heimatsprache, soll auch auf Englisch und Französisch zur Verfügung gestellt werden. Der an die *Generaldirektion Übersetzung (DGT)* der EG-Kommission abgeschickte Übersetzungsauftrag folgt einem festgelegten Arbeitsablauf, in dem maschinelle Übersetzung als sprachliche Ressource einen nicht weg zu denkenden Bestandteil darstellt (im Jahr 2006 wurden 1 541 518 Seiten im Rahmen des DGT workflows maschinell übersetzt)^b
- Sie nutzen den Handy- Service einer großen deutschen Telekommunikationsfirma und haben Fragen zu ihren aktuellen Kosten-Positionen. Ein automatisches System (Sprachdialogsystem) nimmt ihre Anfragen entgegen und gibt Ihnen die gewünschte Information.^c Eine zunehmende Anzahl solcher Sprachportale sind rund um die Uhr erreichbar, z. B. um Auskunft über das aktuelle Kinoprogramm oder die letzten Bundesligaergebnisse zu geben.
- Sie haben eine Firma und wollen ein automatisches Informationsportal im Internet anbieten. Künstliche interaktive Assistenten („Lingubots“) ermöglichen es einem Kunden auch außerhalb der Bürozeiten, durch natürliche Dialoge Informationen über Ihr Firmenangebot zu erfragen.^d

^a Siehe hierzu eine Fallstudie.

^b s. hierzu die Broschüre „Übersetzung und Mehrsprachigkeit“ der EG.

^c Dieses System, das die Arbeit von bis zu 1600 Agenten gleichzeitig erledigen kann, wurde auf den VOICE Awards 2006 mit dem Preis für die beste deutschsprachige Sprachapplikation ausgezeichnet. S. hierzu einen Innovationsreport.

^d s. z. B. <http://www.artificial-solutions.com>

2.1 Allgemeines zur Sprachtechnologie: Unterscheidungen

2.1.1 Terminologie der beteiligten Disziplinen

Sprachtechnologie ist notwendigerweise im Bereich der maschinellen Sprachverarbeitung angesiedelt, die in den 50er Jahren des 20. Jahrhunderts aus einer Symbiose von Linguistik und Informatik entstand.

Wie Abb. 2.1 zeigt, teilt sich die maschinelle Sprachverarbeitung in vier verschiedene Bereiche auf, die aus unterschiedlicher Schwerpunktsetzung resultieren und zum Teil als eigenständige Disziplinen realisiert sind (s. auch *Carstensen et al. 2009:10f*): die **theoretisch orientierte Computerlinguistik** (im engeren Sinn), für die maschinelle Sprachverarbeitung im Wesentlichen als Instrument zur Falsifizierung ihrer Theorien und Modelle eingesetzt wird; die **Linguistische Datenverarbeitung (LDV)**, die (zumindest von ihren Wurzeln her) primär an der zweckorientierten Verarbeitung von Sprachdaten (Lexika, Korpora etc.) interessiert ist; die maschinelle Sprachverarbeitung im engeren Sinn (**natural language processing, NLP**), die als Teilgebiet der Künstliche-Intelligenz-Forschung (KI) Sprache als ein kognitives Phänomen auffasst und zu simulieren versucht (dies bedeutet insbesondere eine Verknüpfung von Sprach- und Wissensverarbeitung); **Sprachtechnologie** (*language engineering, human language technology, HLT*), die in erster Linie an kommerziell einsetzbaren Sprachanwendungen interessiert ist.

Diese Unterteilung impliziert ein unterschiedliches Interesse an bzw. eine unterschiedliche Auffassung/Bewertung von der Entwicklung sprachverarbeitender Systeme (auch: **natürlichsprachliche Systeme/natural language systems (NLS)**): *theoretisch* motivierte Systeme sind in der Regel ungeeignet als angewandte Systeme, *praktische* Anwendungen der LDV erreichen nicht die Komplexität natürlichsprachlicher Systeme, komplexe Systeme des NLP sind meistens kommerziell nicht verwendbare *Forschungssysteme* und HLT-Systeme sind grundsätzlich im Sinne der Realisierbarkeit eingeschränkte *kommerzielle* Systeme.

2.1.2 Computerlinguistik vs. Sprachtechnologie

Bis in die 1990er Jahre traf es zu, dass die Ausbildung in der Computerlinguistik eher Theorielastig war und die Entwicklung von Sprachtechnologie überwiegend in speziellen Abteilungen großer Software-Firmen betrieben wurde. Aus dieser Situation resultiert eine Aussage von Frederick Jelinek, einem IBM-Forscher, der die Entwicklung von Spracherkennungssoftware ausschließlich auf der Basis statistischer Verfahren propagierte: „Whenever I fire a linguist our system performance improves“.¹ Selbst in jüngerer Zeit finden sich entsprechende Meinungen: „Wer Sprachprodukte entwickeln möchte, stelle keinen Computerlinguisten ein!“ (Heyer 2002)

Der Grund für diese Einstellung ist die **rationalistische** –d.h. generell auf theoretischen Überlegungen, Annahmen und Modellen basierenden– Vorgehensweise bei der Entwicklung

¹ in seinem Vortrag „Applying Information Theoretic Methods: Evaluation of Grammar Quality“, Workshop on Evaluation of NLP Systems, Wayne PA, December 1988.

2.1 ALLGEMEINES ZUR SPRACHTECHNOLOGIE: UNTERScheidungen

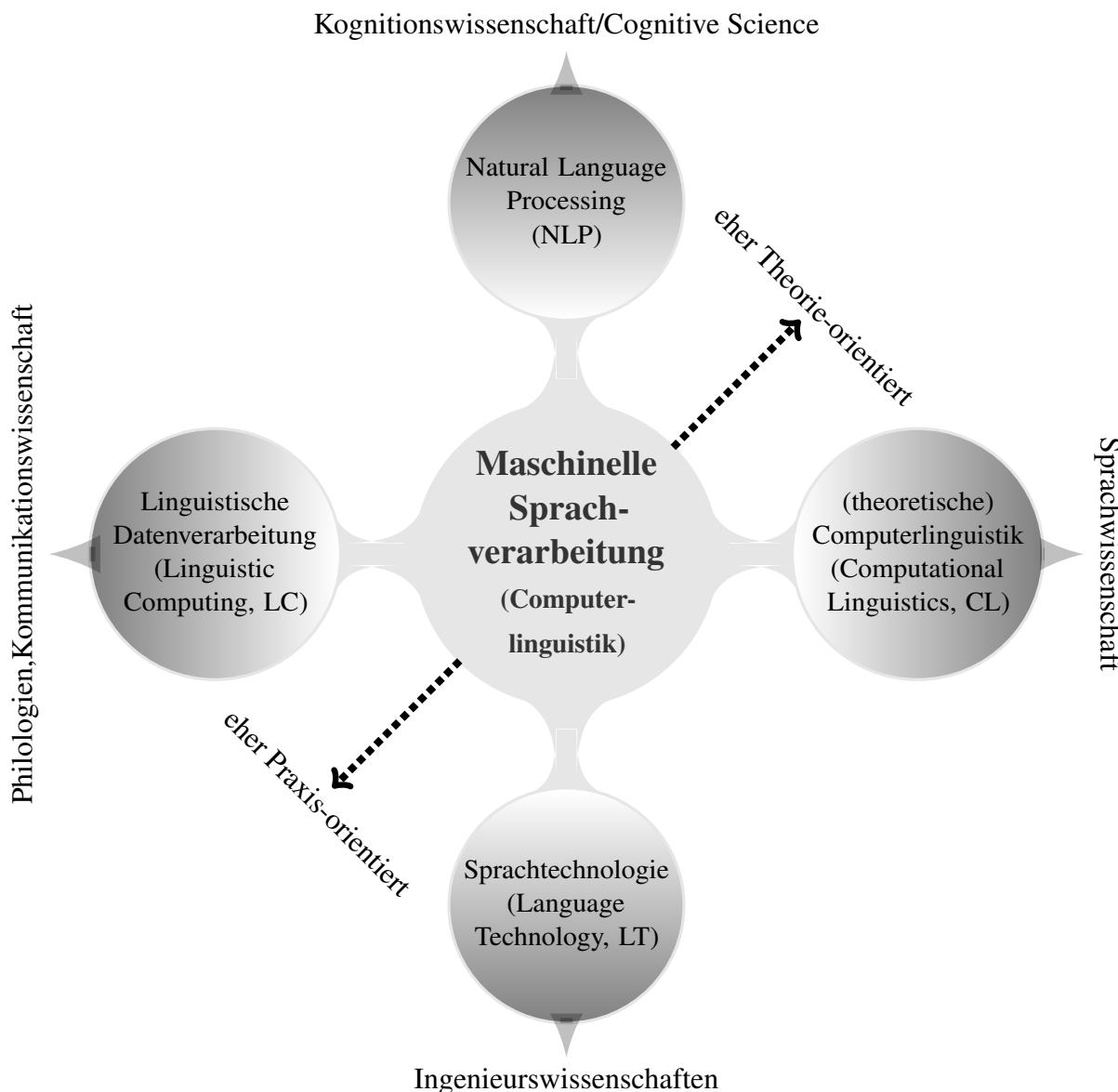


Abbildung 2.1: **Facetten der Computerlinguistik**

von NLS, die in den Jahrzehnten zuvor üblicherweise gewählt worden war. Hierbei nahm die Systementwicklerin die zentrale Position ein, indem sie, theoriegeleitet, sprachliches und sprachrelevantes Wissen über formale Zwischenebenen in funktionierende Sprachsoftware zu überführen versuchte. Allerdings wiesen die NLS erhebliche Defizite auf, darunter mangelnde Performanz (Schnelligkeit) und fehlende **Robustheit** (die Fähigkeit, auch bei/mit Problemen weiter zu funktionieren). Ähnlich wie in der Künstliche-Intelligenz-Forschung wurde diese Vorgehensweise ab Ende der 1980er Jahre zunehmend als gescheitert angesehen.

In dieser Zeit setzte daher eine Wende ein, die gemeinhin als „**statistical turn**“ der Compu-

terlinguistik bezeichnet wird. Sie markiert den Übergang zur überwiegenden Einnahme einer **empiristischen** Position bei der Entwicklung von NLS. Grundlage und Ausgangspunkt sind in diesem Fall immer beobachtbare Sprachdaten (basierend auf Erhebungen oder Sammlungen, **Korpora**, entsprechender sprachlicher Äußerungen), deren statistische Auswertung die empirische Basis sprachtechnologischer Komponenten darstellt.²

In der Computerlinguistik ist seither in mehrfacher Hinsicht eine Tendenz zur Anwendungsorientiertheit mit entsprechend praktischer Ausrichtung („**(natural) language engineering**“) zu verzeichnen. Dies verringert den terminologischen Kontrast zwischen „Computerlinguistik“ und „Sprachtechnologie“ und erübriggt die oft geforderte Isolation der technisch-praktischen von den theoretischen Aspekten der maschinellen Sprachverarbeitung.³ Diese Konvergenz akademischer und industrieller Entwicklungstraditionen wird von den Organisatoren des Workshops „Bridging the Gap: Academic and Industrial Research in Dialog Technologies“⁴ explizit gewürdigt:

We are pleased to see some real convergence from both industry and academic side. While academic researchers are proposing and building practical dialog systems, industrial researchers are starting to implement sophisticated learning and uncertainty modeling into their system.

(aus dem Vorwort der Workshop-Proceedings)

Allerdings – auch wenn die statistischen Verfahren hauptsächlich am Erfolg der Sprachtechnologie in den letzten Jahren verantwortlich sind – ist blindes Vertrauen in sie keineswegs gerechtfertigt, wie Hausser allein in Bezug auf die Spracherkennung schreibt: „[...] the search space of statistical methods will continue to be so large that it constitutes the biggest single obstacle to ever obtaining successful automatic speech recognition“ (Hausser 2002, S. 127).

Zudem ist noch nicht einzuschätzen, wie weit eine solche Methodik trägt und ob nicht auch andere Aspekte ein tragende Rolle spielen werden („many technologies will only become successful once they truly reflect the fact that language is ultimately a cognitive process“, Crocker 2002). Bill Woods –einer der Pioniere der maschinellen Sprachverarbeitung– zieht das folgende Fazit:

The range of open problems cannot be solved by mere application of one, or even a few, existing, well-understood techniques such as first-order logic, Bayesian statistics, and machine-learning algorithms, although these are clearly parts of the solution.

(aus einem Vortrag, gehalten auf einem Workshop der HLT-NAACL-2006)

Weiterhin mehren sich auch die warnenden Worte vor einem statistischen Tunnelblick, sowohl bzgl. des Werts der Linguistik für die maschinelle Sprachverarbeitung (Spärck Jones 2007)

² Der Ratio/Empirie-Dichotomie entsprechende gegensätzliche Wortfelder sind: ‚symbolisch‘, ‚Wissensorientiert‘, ‚Regel-basiert‘, ‚intellektuell‘ und ‚analytisch‘ versus ‚numerisch‘, ‚Korpus-basiert‘, ‚statistisch‘.

³ Ein Vortrag von Jelinek anno 2004 hat übrigens den Titel „Some of my Best Friends are Linguists“.

⁴ auf der NAACL-HLT-2007-Tagung

als auch im Hinblick auf die –nachweislich nachlassende– Offenheit der Branche gegenüber Resultaten anderer Disziplinen im Allgemeinen (Reiter 2007). So kommen beispielsweise Reape und Mellish (1999) bei der Untersuchung eines Aspekts der Sprachgenerierung zu folgendem klaren Schluss:

We fail to see why linguistic theory is any less relevant to any branch of natural language engineering or application than, say, mathematical or theoretical physics is to nuclear power plant design.

Schließlich zeigt die Praxis in Forschung und Entwicklung während der letzten Jahre eine Abkehr von Schwarz-Weiß-Denken, wie der folgende Kommentar des bedeutendsten deutschen Sprachtechnologen Wolfgang Wahlster nahelegt:

[...] hat sich auf dem Gebiet seit 2000 viel getan. Man hatte danach fast 10 Jahre lang einen Hype in der rein statistischen Sprachverarbeitung, die aber in den letzten beiden Jahren zur Ernüchterung führte. Einige US-Projekte von DARPA wurden sogar eingestellt, weil einfach die Qualität der Analyse und Generierung bei rein statistischen Lernverfahren zu schlecht war. Jetzt setzt man wieder vermehrt auf die Integration von statistischen Lernverfahren über annotierten Korpora und hochwertigen Wissensquellen, in denen der wissenschaftliche Erfahrungsschatz und die Erkenntnisse mehrerer Generationen von Sprachwissenschaftlern codiert wird [...]⁵.

Die 'Ratio vs. Empirie'-Kontroverse wird sich daher eher im Wohlgefallen der User solcher NLS auflösen, deren Erfolg auf einer cleveren Mixtur symbolischer und statistischer Verfahren basiert. Der praktische Erfolg eines NLS ist jedenfalls nicht durch eine vermeintlich notwendige Abgrenzung der 'Sprachtechnologie' von der 'Computerlinguistik' bestimmt, sondern eher durch pragmatische Lösungen für auftretende Probleme. Hierbei kann gerade die Intuition und Fachkenntnis einer Computerlinguistin zu einem entscheidenden Performance-Boost führen. Nicht umsonst finden Computerlinguistik-Spezialisten heutzutage leicht eine Anstellung in Sprachtechnologiefirmen, oder gründen selbst welche, hier ein Beispiel einer –nicht gerade unprätentiösen– Selbstdarstellung:

Teragram Corporation is the market leader in multilingual natural language processing technologies that use the meaning of text to distill relevant information from vast amounts of data. **Founded in 1997 by innovators in the field of computational linguistics**, Teragram alone offers the speed, accuracy and global language support that customers and partners demand to retrieve and organize growing volumes of digital information. (aus: <http://www.teragram.com>, Hervorh. von mir)

Fehlende Fachkenntnis hingegen ist nicht selten Ursache überzogener oder falscher Erwartungen – und zwar nicht nur auf Nutzer, sondern auch auf Entwicklerseite. Ein Beispiel ist die

⁵ aus: W. Wahlster (2011), Interview mit Ubbo Visser, *Künstliche Intelligenz*, 25 (4), S. 342.

intelligente Suche im Web: Louis Monier (Entwickler der Altavista-Suchmaschine und gegenwärtig bei Google) berichtet, er bekomme jährlich eine Vielzahl Angebote neuer Suchmaschinen-Firmen:

Jedesmal haben sie irgendeinen kleinen Hebel, von dem sie annehmen, er sei riesig. Sie realisieren nicht, dass sie nur eine bessere Türklinke an einer [Boeing] 747 haben. Nun müssen sie nur noch die 747 bauen. Keine der Ideen, die ich gesehen habe, ist überzeugend.

(frei übersetzt aus: Quest for last word in search, *Sunday Times*, 19.11.2006)

Ähnliches lässt sich über zu einseitig technische Herangehensweisen an die Sprachtechnologie sagen, weshalb vor zu vollmundigen Versprechungen in Firmen-Hochglanzbroschüren, Projekt-Proposals o.Ä. gewarnt werden muss: „there does seem to be a grotesque misjudgement of the capabilities of natural language systems and underestimation of how complex human language is“ (*Copestake und Spärck Jones* 1990, in einer Reaktion auf euphorische Pressemeldungen zu einer natürlichsprachlichen Datenbankschnittstelle).

Dabei ist „Suche im Web“ das Thema, mit dem sich die Sprachtechnologie profilieren kann. **Natürlichsprachliche Suche** ist das Stich/Hype-Wort, wegen dem in den letzten Jahren Millionen in entsprechende Start-ups investiert wurden, z. B. in die Firma **POWERSET** (jetzt **BING**). Dass hierbei auch symbolische Verfahren der Computerlinguistik noch/wieder eine Rolle spielen, zeigt der folgende Ausschnitt⁶ aus dem Interview mit Peter Norvig, Mitherausgeber der Standard-Einführung in die Künstliche Intelligenz⁷ und Forschungsdirektor bei **GOOGLE**:

VentureBeat: Has Google hired anyone yet to focus on the possibility of using symbolic/deep approaches?

Norvig: Google has several teams focused on natural language and dozens of Googlers with a PhD in the field, including myself.

Der nächste Schritt, an dem schon lange geforscht, aktuell aber auch verstärkt gearbeitet wird, sind sogenannte **Antwortmaschinen**, also sprachtechnologische Anwendungen im Bereich der **Fragebeantwortung (question answering)**, die nicht nur *Suchergebnisse* auf eine Anfrage liefern, sondern konzise *Antworten*. Hier hat der Erfinder von Mathematica, Stephen Wolfram, mit seiner Antwortmaschine **WOLFRAM|ALPHA** —um in der Analogie zu bleiben— einen sehr wendigen Düsenzweisitzer für jedeperson gebaut hat, aber keinesfalls einen flexibel einsetzbaren Jumbo. **WOLFRAM|ALPHA** beinhaltet innovative Methoden zum Finden bestimmter Antworten, ist aber weit von der Allgemeinheit von Frage-/Antwortsystemen entfernt, wie sie in der Sprachtechnologie erforscht werden (s. Kap. 7). Einen vergleichbaren Service bietet **TRUE KNOWLEDGE**.

Während **WOLFRAM|ALPHA** mittlerweile in verschiedener Form existiert (bzw. eingebunden ist, u.a. in die Persönlicher-Digitaler-Assistent-Komponente **SIRI**), hat das IBM-System

⁶ Matt Marshall, „Powerset’s search technology scoop, may scare Google“, *Venture Beat*.

⁷ Stuart Russel, Peter Norvig, *Artificial Intelligence: A Modern Approach*.

WATSON, das in *Jeopardy!* gegen zwei Champions angetreten und gewonnen hat, Geschichte geschrieben, indem es gezeigt hat, dass qualitativ hochwertige Sprachverarbeitung im Bereich der Fragebeantwortung realisierbar ist (s. Kap. 7, genauer: S. 110). Breitenwirkung wird allerdings eher durch Apples **SIRI** erreicht werden, das zwar nicht so intelligente, aber dafür praxisrelevante Leistung erbringt.

Fasst man also „Computerlinguistik“ allgemein als eine Verbindung von Linguistik und Informatik zum Zwecke der maschinellen Sprachverarbeitung auf, so kann „Sprachtechnologie“ guten Gewissens als Bereich der ‚Anwendungen der Computerlinguistik‘ bezeichnet werden.

2.1.3 Tiefe/flache Verarbeitung

Eine weitere Unterscheidung ist die zwischen „tiefer“ und „flacher“ Verarbeitung (*deep vs. shallow processing*). Tiefe Verarbeitung involviert in der Regel eine vollständige Analyse von Sprache auf allen relevanten Wissensebenen, flache Verarbeitung macht dagegen Abstriche und erfordert typischerweise nur partielle Analysen auf ausgewählten Repräsentationsebenen. **Hybride** Systeme sind solche, die beide Verfahrenstypen verwenden und deren jeweilige Vorteile entsprechend ausnutzen.

2.1.4 Gesprochene/geschriebene Sprache

Im Deutschen muss explizit zwischen *geschriebener* und *gesprochener Sprache* unterschieden werden, da es hier im Gegensatz zum Englischen („speech“) kein eigenes Wort für gesprochene Sprache gibt.

Interessanterweise fällt heutzutage noch ein großer Teil sprachtechnologischer Applikationen in den Bereich der Verarbeitung von ‚Speech‘, und dort in erster Linie auf die Sprachschallbasierte Verarbeitung der menschlichen Stimme (’Voice‘).

Es kann an dieser Stelle nur angedeutet werden, dass eine umfassende maschinelle Sprachverarbeitung weitaus mehr erfordert als eine funktionierende Zuordnung von Sprachlauten zu Wörtern. Man muss also nicht nur systematisch zwischen geschriebener und gesprochener Sprache unterscheiden, sondern auch beachten, dass „Sprachtechnologie“ oft mehr Kompetenz suggeriert als sie zur Zeit in der Lage ist zu leisten.

2.2 Sprachtechnologie: Grobcharakterisierung der Anwendungen

Anwendungen der CL lassen sich im Wesentlichen anhand dreier Kriterien unterscheiden. Das erste betrifft die sprachliche **Ausdrucksform** –d.h. den Unterschied geschriebener vs. gesprochener Sprache– und führt entsprechend zur Differenzierung von „Text“-Technologie und „Speech“-Technologie.

Das zweite betrifft die **Komplexität** der Anwendungen, die von aufgabenspezifischen abhängigen Modulen/Plug-ins (z. B. für Rechtschreibkorrektur oder Textzusammenfassung in größeren Textverarbeitungsprogrammen) und Werkzeugen über relevante Komponenten komplexer Systeme (wie z. B. natürlichsprachliche Schnittstellen) bis hin zu eigenständigen natürlichsprachlichen Systemen reichen.

Das dritte Unterscheidungskriterium ist der **Funktionalitätsbereich/Zweck** sprachtechnologischer Software.

2.2.1 Ausdrucksform

Maschinelle Verarbeitung gesprochener Sprache („Speech“-Technologie)

Hier ist zwischen der Erkennung gesprochener Sprache (*Spracherkennung, automatic speech recognition (ASR)*) und der Produktion gesprochener Sprache (*Sprachsynthese, speech synthesis, text-to-speech synthesis (TTS)*⁸) zu unterscheiden. Beide Termini sind nicht zu verwechseln mit **Sprachverstehen** bzw. **Sprachgenerierung**, die auch tiefergehende Aspekte der Syntax-, Semantik- und Wissensverarbeitung usw. umfassen.

Spracherkennung und -synthese gehören zu den am stärksten wachsenden Anwendungsbereichen der CL und haben sich von einstigen Insellösungen (z. B. Text-to-speech-Synthese für Blinde) zu Produkten im Consumerbereich entwickelt (z. B. für das automatische Vorlesen eingehender Emails), aber auch zu solchen in höherwertigen sicherheitsrelevanten Anwendungen (z. B. Stimmenerkennung bzw. -authentifizierung). Entsprechend betrug der Gesamtumsatz in der Speech-Branche (Software, Hardware, Dienste)⁹ im Jahr 2004 etwa 1 Mrd Dollar und wird für 2009 auf 2,7 Mrd. Dollar prognostiziert (*Hong 2006*).

Im Bereich der Maschinellen Übersetzung existieren die Bereiche der **maschinellen Übersetzung gesprochener Sprache** (*spoken language translation, SLT*) und der **maschinellen Simultanübersetzung** (*speech-to-speech-translation, SST*)¹⁰. Der Unterschied besteht darin, dass bei der SLT gesprochener Input in *Text* der Zielsprache übersetzt wird.

Maschinelle Verarbeitung von Text (Texttechnologie)

Im Bereich geschriebener Sprache existieren folgende prominente Anwendungsgebiete:

- Rechtschreib- und Grammatikkorrektur
- Computergestützte Lexikographie und Terminologie
- Textbearbeitung, -filterung, -extraktion, -indexierung -kategorisierung
- Dokumentenerschließung und -klassifikation, Textstelzenzugriff
- Textmining
- Einfache Verfahren der Textzusammenfassung, Fragebeantwortung und Antwort-extraktion

⁸ TTS ist in diesem Bereich Stand der Kunst, auch wenn eine inhaltsbasierte Sprachsynthese („concept-to-speech synthesis“) wünschenswert ist und daher angestrebt wird.

⁹ s. hierzu auch Kap. 9.

¹⁰ auch: maschinelles Simultandolmetschen

- Maschinelle Übersetzung (MÜ)

Rechtschreib- und Grammatikkorrektur wird benötigt für flasch geschiebene Wörtern und Sätze. Zumindes Rechtschreibkorrektur gehört seit einiger Zeit zum Standard bei Text-Editoren wie MS-Word. Ihre Nützlichkeit ist ein wesentlicher Grund für die zunehmende Akzeptanz von Sprachsoftware.

Computergestützte Lexikographie und Terminologie ist das klassische Gebiet der Linguistischen Datenverarbeitung. Hier geht es um die automatische Erstellung und Pflege von Lexika und (Fach)Wortschatzen.

Textbearbeitung, -filterung, -extraktion, -indexierung -kategorisierung sind Verfahren, die im Workflow des Wissensmanagement (s. hierzu mehr auf Seite 17) ihren Einsatz finden. Z.B. werden eingehende Dokumente (Emails o.ä.) automatisch an firmeninterne Dokument-Standards angepasst, kategorisiert und je nach Thema/Sachgebiet weitergeleitet. Sie werden indexiert, um sie angemessen (für späteren Zugriff wiederfindbar) abzulegen. In diesem Arbeitsablauf wird Irrelevantes/Unerwünschtes ausgefiltert, relevante Information wird extrahiert und entsprechend weiterverwendet.

Bei der *Dokumentenerschließung* (auch: Dokumentenzugriff, *document retrieval*) handelt es sich um die klassischen Verfahren des *Information Retrieval* (*Informationserschließung*, s. z. B. Salton und McGill 1987; Baeza-Yates und Ribeiro-Neto 1999): Auf eine Anfrage hin sollen Dokumente (möglichst alle und nur die gewünschten) geliefert werden. Voraussetzung hierfür ist die Strukturierung von Dokumentkollektionen durch Verfahren der Textklassifikation, bei denen ähnliche Dokumente gleichen (vorgegebenen) Klassen zugeordnet werden.

Textstelzenzugriff/-erschließung (*Passage retrieval*) ist eine Verfeinerung des *document retrieval*: Es sollen nur Passagen oder Phrasen als genauere Antworten auf eine Anfrage geliefert werden.

Textmining i. e. S. ist die Anwendung des *Data mining* (Aufdecken unbekannter Zusammenhänge in Daten) auf Texte.¹¹ Während *Information retrieval* generell das Finden von Information relativ zu einer Anfrage meint, handelt es sich hier um das Auffinden oder Entdecken unbekannter, neuer Information in Texten. Im weiteren Sinn bezeichnet *Text mining* das ganze Gebiet des informatischen Umgangs mit Texten.

Bei *einfachen Verfahren der Textzusammenfassung, Fragebeantwortung und Antwortextraktion* kommen –im Gegensatz zu natürlichsprachlichen Systemen– überwiegend flache, statistische Verfahren der CL zum Einsatz. Die Ergebnisse sind daher qualitativ minderwertig und lassen sich eher für eine Sichtung/Vorauswahl verwenden.

Sprachtechnologie

Sprachtechnologie im weiteren Sinn ist im Idealfall in der Lage, mit beiden Ausdrucksformen (Lautsprache, Schriftsprache) umzugehen. Komplexe natürlichsprachliche Systeme verfügen über entsprechende Module zur Erkennung und Synthese und implementieren in der Regel eine Funktionalität (Fragen beantworten, Dialoge führen etc.), die über die der Text- und Speech-Technologie hinausgeht (s. die folgenden Kapitel).

¹¹ s. hierzu Hearst (1999) und Marti Hearst, What is text mining.

2.2.2 Komplexität

Einsatz computerlinguistischer Methoden in (semi-)eigenständigen Werkzeugen

Dies sind im Wesentlichen Systeme, die eine unterstützende Funktion innehaben (z. B. die Erstellung von Ressourcen für die CL). Hierzu gehören

- Lexikographie- und Terminologiesysteme
- Grammatik-Werke
- Korpora-Verwaltungswerkzeuge
- Werkzeuge der Computer-gestützten Übersetzung

Lexikographie- und Terminologiesysteme sind Systeme zum Aufbau bzw. zur Pflege lexikalischer und terminologischer Ressourcen im Rahmen des o.g. Anwendungsbereichs. Traditioneller Einsatz ist in Verlagen und Dokumentations-/Übersetzungsfirmen, zunehmend aber auch für/in computerlinguistischen Anwendungen selbst.

Grammatik-Werke werden für die Erstellung und Pflege von Ressourcen für 'tiefe' grammatische Verfahren benötigt. Hierzu gehören u.a. Grammatik-Editoren, -Compiler und - Debugger. Ein Beispiel ist die „Grammar Writer's Workbench“ für LFG-Grammatiken bei Xerox Parc. *Helbig* (2008) liefert Beispiele für Werke im Bereich der Semantik und Wissensmodellierung.

Korpora-Verwaltungswerkzeuge umfassen einerseits **Abfragesysteme für reine Textkorpora**, z. B. traditionelle Systeme für Keyword-in-context (KWIC)-Abfragen oder das komplexere SARA (SGML-Aware Retrieval Application) für das British National Corpus (BNC); andererseits **Abfragesysteme für annotierte Korpora**, d.h. entweder für Korpora von mit Syntaxanalyseinformationen annotierten Texten (*tree banks, Baumbanken.*)¹² oder für **Sprachdatenbanken**, d.h., Korpora gesprochener Sprache (s. hierzu *Draxler* 2009).

Werkzeuge der Computer-gestützten Übersetzung (*computer aided translation*, CAT), finden ihren Einsatz in dem Bereich der maschinellen Übersetzung, in dem die Übersetzung nicht vollautomatisch durchgeführt wird: z. B. für das Speichern und Abrufen übersetzter Textteile (*Translation memories*) oder die Nutzung mehrsprachiger alignierter Korpora (s. Kapitel 12).

Einsatz computerlinguistischer Methoden/Module/Komponenten als Bestandteile komplexer Systeme

Hier existieren die folgenden Einsatzgebiete:

- Natürlichsprachliche Schnittstellen
- Computergestütztes Sprachlehren/-lernen oder *Computer-Assisted Language Learning* (CALL)
- Elektronische Kommunikationshilfen im Bereich *Alternative and Augmentative Communication* (AAC)

¹² Ein Beispiel für ein entsprechendes Abfragetool ist **TIGERSEARCH** (Institut für maschinelle Sprachverarbeitung, IMS, Universität Stuttgart).

Natürlichsprachliche Schnittstellen existieren als **natürlichsprachlicher Zugang** zu Datenbanken, Internet etc., für die **natürlichsprachliche Bedienung** von Geräten, Systemen etc. oder ermöglichen die **einfache natürlichsprachliche Kommunikation** mit Geräten, Systemen etc. (einfache Frage-/Antwort-Interaktion). Sie erlauben einem Benutzer einen flexibleren Umgang mit der jeweiligen Applikation (s. aber zu den damit verbundenen technischen Schwierigkeiten in Kapitel 6). Durch Verbesserung der Spracherkennungstechnologie bieten insbesondere auch die Bereiche der stimmgesteuerten Geräte und der automatischen telefonischen Interaktion (Sprachdialogsysteme, s. Kap. 9) Perspektiven. Siehe auch Kapitel 8 zu elaborierten Aspekten natürlichsprachlicher Schnittstellen.

Im *Computergestützten Sprachlehren/-lernen* werden z. Z. erst einfache computerlinguistische Methoden und Ressourcen verwendet (z. B. Rechtschreib- und Grammatikkorrektur). Eine kompetentere Sprachtechnologie würde hier allerdings enorme Möglichkeiten bieten, z. B. intelligente Dialoge mit den Schülern zum Zwecke der Problemlösung und Hilfestellung.

Elektronische Kommunikationshilfen erfordern eine benutzerangepasste Verwendung von Methoden der Computerlinguistik, um spezifische Handicaps zu kompensieren. Sie werden jedoch insgesamt ebenfalls von verbesserten Verfahren der Spracherkennung und -synthese profitieren (und mit ihnen ihre Nutzer). Es sollte unmittelbar einleuchten, dass z. B. eine sprachgestützte Steuerung von Geräten einen unschätzbar wert für viele körperlich Benachteiligte darstellt.

Natürlichsprachliche/multimodale Systeme

Natürlichsprachliche Systeme bilden den zentralen Gegenstand dieses Buchs und werden ab dem folgenden Kapitel näher beschrieben. An dieser Stelle ist anzumerken, dass moderne Informationssysteme multimodal ausgerichtet sind, d.h., dass z. B. versucht wird, bei der Analyse einer Äußerung auch die jeweilige Gestik und/oder Mimik der äußernden Person mitzuberücksichtigen oder bei der Generierung von strukturiertem Informationsmaterial auch grafische Elemente angemessen zu platzieren (s. Kap. 10). Wenn also hier von natürlichsprachlichen Systemen die Rede ist, so sind, wenn der Kontext dies nicht ausschließt, damit auch multimodale Systeme mit entsprechender natürlichsprachlicher Kompetenz gemeint.

2.2.3 Funktionalität

Der Kern der maschinellen Sprachverarbeitung besteht aus dem spezifischen Geflecht von Mensch, Computer, Sprache und Information. Einzelne Anwendungen lassen sich deshalb auch dadurch charakterisieren, auf welche Beziehung in diesem Geflecht sie den Schwerpunkt setzen. Dieser Zusammenhang ist in Tabelle 2.1 auf der nächsten Seite dargestellt, wobei im Folgenden näher behandelte komplexe Anwendungen hervorgehoben sind (zu einem Überblick über einfache Anwendungen der Computerlinguistik s. auch Carstensen et al. 2009; Loblin und Lemnitzer 2004).

Wie Maybury (2001) zeigt, spielen die meisten dieser Anwendungen eine wesentliche Rolle im **Wissensmanagement** (*knowledge management*). Diese Zukunftsdisziplin ist die Summe sprachtechnologischer Kompetenz angewandt für die Bedürfnisse von Unternehmen

Beziehung	Anwendung	Zweck
Computer–Information	Dokument- und Wissensverarbeitung, Textklassifikation, Text mining, Ontologieerstellung	Automatische Erstellung/ Pflege von Informations-/ Wissens- Ressourcen
Mensch–Sprache	CALL, AAC	Maschinelle Sprachverarbeitung als direkter Mehrwert für den Menschen
Computer–Sprache	Terminologie-/Lexikonwerkzeuge	Automatische Erstellung/ Pflege sprachlicher Ressourcen
Sprache–Computer	Spracherkennung, Sprachsynthese, Maschinelle Übersetzung	Automatische Behandlung sprachlicher Phänomene
Mensch–Information	Natürlichsprachlicher Zugang zu Datenbanken, Frage-/Antwort-Systeme	Natürlichsprachlicher Zugriff auf Information
Information–Mensch	Generierungs- und Textzusammenfassungssysteme	Natürlichsprachliche Informationspräsentation
Sprache–Information	Informationsextraktion	Automatisches Extrahieren von Information aus Texten
Mensch–Computer	Human Computer Interaction (HCI): Dialogsysteme, Multimodale Systeme	Kommunikation mit Computer

Tabelle 2.1: Anwendungen nach Funktionalität

im Informationszeitalter. Dabei geht es darum, existierende Dokumente und Expertenwissen im Unternehmen zu strukturieren und –wie auch relevantes Wissen außerhalb des Unternehmens– für die Mitarbeiter einfach verfügbar zu machen.

2.3 Ausblick

Auf den ersten Blick sind die Leistungen der Sprachtechnologie ernüchternd: Maschinelle Übersetzung ist seit fast 60 Jahren in der Entwicklung, liefert aber nur in eingeschränkten Kontexten Ergebnisse von höherer Qualität (etwa seit Beginn der 1980er). Einzig die Spracherkennung ist so weit fortgeschritten, dass sie für kleine Domänen allgemein verwendbar ist. Bei der Textzusammenfassung und der Fragebeantwortung gibt es seit Mitte der 1990er bzw. Anfang der 2000er Performanzsprünge, jedoch sind beide Bereiche weit von allgemeiner Verwendbarkeit ihrer Systeme entfernt. Nur sinnvoll beschränkte, aber gleichzeitig innovative

NLS können momentan erfolgreich sein (s. z. B. Kap. 9).

Auf den zweiten Blick stellt die Entwicklung der Sprachtechnologie allerdings eine Erfolgs-story dar (s. *Lockwood und Joscelyne* 2003): Vor 25 Jahren kaum existent, gab es 2003 allein in Deutschland mehr als 60 Firmen im Bereich Human Language Technology (davon 32 im Bereich gesprochener Sprache). 13 Firmen entwickelten mehrsprachige Anwendungen, und von 60 Firmen wurden Sprachprodukte und -services vertrieben. Zudem gab es 80 Forschungseinrichtungen im Bereich HLT: rein kommerzielle Abteilungen (bei IBM, Philips, Bosch etc.), angewandte Forschungseinrichtungen (Fraunhoferinstitut) und mehr als 20 Unis mit Forschung im Bereich CL und HLT. In Europa nehmen Deutschland, Finnland, Großbritannien und die Niederlande eine führende Position ein, neben den „Global Playern“ USA und Japan.

In einer Studie über den Umfang der Sprachindustrie innerhalb der EG¹³ wird der „Wert“ der Sprachindustrie innerhalb der EG im Jahr 2008 auf etwa 8.4 Mrd Euro geschätzt und mit einer Verdopplung bis 2015 gerechnet.

Einige Vorteile der Sprachtechnologie (oft als Folge einer reifen Speech-Technologie oder HLT-verbesserten Informationstechnologie erkennbar) sind die folgenden:

- Kostenersparnis (z. B. bei Anbietern von Beratung oder Auskünften)
- Gewinnschaffung (durch innovative Angebote)
- verbesserte Kommunikation im Geschäftsbereich, sowohl mit Geschäftspartnern ((partielles) automatisches Dolmetschen von Gesprächen, maschinelle Übersetzung von Dokumenten zur Sichtung) als auch mit Kunden (z. B. rund-um-die-Uhr-Präsentation des Firmenangebots durch automatisierte Kundenberatung im Internet)
- bessere/einfachere/intuitivere Bedienung von Geräten
- einfacherer und besserer Zugang zu Information (automatische Fragebeantwortung)
- Zeitersparnis bei Recherchen (automatisierte Suche nach Information, deren Filterung sowie deren Benutzer-anangepasste Präsentation, z. B. durch entsprechende Textzusammenfassung)

Moderne, aktuell avisierte Anwendungen sind z. B. die folgenden:

- Geschäftsüberwachung (allgemeiner: *business intelligence*). Hier geht es um die Bedürfnisse von Firmen, auf dem neuesten Stand der Information bzgl. eigener Entwicklungen (z. B. zu Patentverletzungen) oder der von Konkurrenten (Konkurrentenanalyse) zu sein und zu bleiben.
- Intelligente Suche im Web à la [TRUE KNOWLEDGE](#) oder [WOLFRAM|ALPHA](#): das Liefern angemessener Antworten auf natürlichsprachliche Suchanfragen. Die Vi-

¹³ A. Rinsche, N. Portera-Zanotti (2009): *Study on the size of the language industry in the EU*, Study report to the Directorate General for Translation of the European Commission, DGT-ML-Studies 08, August 2009. The Language Technology Centre Ltd.

sion hier ist die Automatisierung der mühsamen Sichtung relevanter Information. Sie kann durch verbesserte und komplexere Frage-/Antwort-(QA)-Techniken zusammen mit automatischer Textzusammenfassung in der Kommunikation mit Informations-Agenten erreicht werden.

- Intelligente (Touristeninfo-)Handhelds à la iPhone mit **SIRI**. Sie sollen per multi-modaler und -medialer Interaktion die verschiedensten Auskünfte zu Informationsbedürfnissen eines Nutzers geben (Reiseinformationen, Karten, sonstige Auskünfte (Kino, Theater, Fernsehen usw.), ggf. per Web), wo immer er sich befindet. Im Idealfall wird sich die Sprachtechnologie zu einer Schlüsseltechnologie im Informationszeitalter entwickeln: als unentbehrliches Bindeglied zwischen Mensch und Maschine und als Voraussetzung für die Herstellung Benutzer-freundlicher Produkte.
- Entscheidungsfindung als Fragebeantwortung bzgl. gegebener Informationssourcen. Dies ist das Szenario, das **WATSON** anhand des *Jeopardy!*-challenge erfolgreich bewältigt hat (s. S. 110), das nachfolgend aber auf praxisrelevante Kontexte übertragen werden soll. Ein Beispiel ist vorliegende Information über bestimmte Krankheitssymptome, aus dem ein hypothetisches **FLEMING** z.B. „Infektion, zu behandeln mit Antibiotika“ auf die Frage „Was ist die Krankheit?“ antworten könnte.

Die folgenden Kapitel sollen einige Information mehr zu natürlichsprachlichen Systemen im Allgemeinen sowieso zu Herkunft und Stand der Kunst einzelner NLS-Typen geben.

2.4 Literatur

Willée et al. (2002) präsentieren einen Querschnitt durch die aktuelle Computerlinguistik/Sprachtechnologie (s. auch die Rezension in *Carstensen* 2003). *Heyer et al.* (2006) bieten eine kompakte Einführung in die Grundlagen, Methoden und Anwendungen der Texttechnologie aus stark praxisorientierter Perspektive (Text als zu verarbeitender Rohstoff)¹⁴, *Pickett und Schroeder* (1999) einen Überblick über die Speech-Technologie. *Lobin und Lemnitzer* (2004) ist ein sehr gut lesbares Handbuch zur –inhaltlich etwas weiter gefassten– Texttechnologie, in dem verschiedenen Themen (u.a. Metadaten, Systemarchitekturen, Automatisches Zusammenfassen, Informationserschließung, Text-to-speech-Synthese) ausreichend Platz für eine das Verständnis fördernde Lektüre eingeräumt wird. Absoluten Anfängern im Bereich Computerlinguistik mit Interesse an Texttechnologie und deren Methoden empfehle ich *Lobin* (2009).

Das umfassendste Informationsangebot zu verschiedenen Aspekten der Sprachtechnologie findet sich auf dem vom DFKI (Deutsches Forschungszentrum für Künstliche Intelligenz GmbH) angebotenen und vom BmBF (Bundesministerium für Bildung und Forschung) geförderten Informationsportal „Language Technology World“. Dieses Informationsportal ist –neben einem Demonstrationszentrum und einem Evaluationszentrum– eines der drei Standbeine des „Deutschen Kompetenzzentrums für Sprachtechnologie“ im DFKI.

¹⁴ Unter Anderem enthält dieses Buch auch ein Glossar texttechnologischer Termini.

Für diejenigen, die sich für die theoretischen und methodischen Aspekte der Sprach¹⁵-verarbeitung interessieren, ist die „ACL Anthology“, das digitale Archiv der Papiere von Sprachverarbeitungskonferenzen, besonders interessant. Neben den theoretisch ausgerichteten Konferenzen und *Computational Linguistics* (der Zeitschrift der *Association for Computational Linguistics*, ACL) finden sich hier auch die Papiere anwendungsnäherer Konferenzen wie der ANLP (Applied Natural Language Processing) sowie solche spezifischer Workshops oder Special Interest Groups (SIGs).

¹⁵ Language, nicht nur Speech!

Kapitel 3

Natürlichsprachliche Systeme (NLS)

Susan Calvin:	<i>There is a robot in this formation that does not belong. Identify it.</i>
NS5 Robots:	<i>One of us.</i>
Susan Calvin:	<i>Which one?</i>
NS5 Robots:	<i>One of us.</i>
Detective Del Spooner: (with sarcasm)	<i>How much did you say these things cost?</i>

(aus: *I, Robot (Film)*)

Natürlichsprachliche Systeme können allgemein als Realisierungen einer komplexen Anwendung der CL aufgefasst werden. Auch wenn solche Systeme gegenwärtig überwiegend noch nur als Forschungsprototypen existieren, so zeigt sich doch seit den 80er Jahren eine zunehmende Kommerzialisierung solcher Anwendungen.

Gleichzeitig besteht ein Konsens im Bereich der maschinellen Sprachverarbeitung darüber, dass die Zeit angesichts der Entwicklung der Computertechnologie reif ist, das Potential der Sprachtechnologie auszunutzen und ihre Produkte ingenieurmäßig zu entwickeln.

In Ermangelung einer klaren Definition lassen sich NLS dadurch charakterisieren, dass sie

- **natürlichsprachliche Äußerungen auf systematische Weise verarbeiten.**
D.h., sie basieren auf computerlinguistischen Prinzipien und -Methoden (bzw. solchen der Künstlichen Intelligenz (KI)). Dies steht im Gegensatz zu, z. B., der Verwendung von Schlüsselwort-Reaktion-Mustern (Parade-Beispiel ist das Programm **ELIZA** von Joseph Weizenbaum, s. *Weizenbaum* 1966 und den Exkurs auf Seite 26)
- **wissensbasiert** sind
 - sprachliches und nicht-sprachliches Wissen ist deklarativ oder zumindest theoriegeleitet abgelegt (vs. direkt/performanz-orientiert programmiert)
 - die Systeme verfügen über interne Repräsentationen, die manipuliert werden

- erfordern i.d.R. eine „tiefe“ Analyse (im Gegensatz zu eher flachen Analysen im Information Retrieval)
- Klassiker: **SHRDLU** (Terry Winograd, s. *Winograd 1972* und den Exkurs auf Seite 30)
- eine **kommunikative Funktion** (z. B. Dialog führen) **oder eine sprachspezifische Aufgabe** (z. B. Übersetzen) haben

3.1 Allgemeines zu NLS

Diese Charakterisierung ist suboptimal, da gerade moderne Systeme versuchen, so weit wie möglich auf Wissensbasiertheit im rationalistischen Sinn zu verzichten. Auch perfektionieren z. B. einige Typen von Dialogsystemen den von **ELIZA** (s. den Exkurs auf Seite 26) geprägten Ansatz, Rationalität nur vorzutäuschen. Eine striktere Charakterisierung müsste sich damit auf die Aspekte „komplexe Anwendung“ und „kommunikative Funktion/sprachspezifische Aufgabe“ beschränken.

Es ist somit nicht unbedingt sinnvoll, aktuell existierende Systeme als Ausgangspunkt der Charakterisierung von NLS zu nehmen. Stattdessen bietet es sich an, die Eigenschaften eines „idealen“ NLS zu betrachten. Solch ein System sollte die folgenden Fähigkeiten besitzen:

- sprachliche Eingaben angemessen analysieren („verstehen“)
- sprachliche Ausgaben angemessen generieren
- Fragen angemessen beantworten: Information ist explizit im vorhandenen Wissen enthalten, aus Informationsquellen abrufbar oder implizit gegeben, d. h. inferierbar
- sinnvolle, kooperative Dialoge führen
- wohlgeformte von nicht-wohlgeformten Texten bzw. Diskursen unterscheiden und entsprechend reagieren („Wie bitte?“)
- Paraphrasen/Zusammenfassungen liefern
- von einer Sprache in eine andere übersetzen
- Input und Output unterschiedlicher Modalitäten angemessen verarbeiten und koordinieren

Reale NLS verfügen in der Regel nur über eine Auswahl dieser Kompetenzen und sind daher, je nach Schwerpunkt,

- **Sprach-/Textverständssysteme**, die sprachliche Eingaben mit Hintergrundwissen verrechnen, Schlüsse (**Inferenzen**) daraus ziehen und angemessen reagieren (Fragen beantworten, Hinweise geben etc.). Diese Technologie ist allerdings, nach einigen wenig weit tragenden Versuchen in den 70er und 80er Jahren, schlicht nicht-existent.

- **Sprach-/Textgenerierungssysteme**, die aus nicht-sprachlichem Material verständliche Sprache produzieren (z. B. Berichte, Beschreibungen, Anweisungen, Hilfetexte).
- **Natürlichsprachliche Zugangssysteme zu Information oder Geräten**, die zu Fragen relevantes Informationsmaterial in angemessener Form liefern oder die auf Anforderungen/Anweisungen hin bestimmte Aktionen auslösen.
- **Frage-/Antwort-Systeme**, die auf Fragen konzise Antworten geben (und nicht nur passendes Informationsmaterial liefern).
- **Dialogsysteme**, die eine natürliche Konversation mit dem Computer ermöglichen.
- **Textzusammenfassungssysteme**, die nicht nur relevante Textteile zu kürzeren Texten (Extrakten) zusammenstellen, sondern eine neue kürzere Version (Abstrakt) aus der relevanten Information eines Textes generieren.
- **Maschinelle Übersetzungssysteme**

3.2 Geschichte

Sieht man einmal von den Anfängen im Bereich der Maschinellen Übersetzung (s. hierzu Kap. 12) ab, lässt sich die Entwicklungsgeschichte natürlichsprachlicher Systeme grob in vier Phasen einteilen:

Erste Fingerübungen und Gehversuche (1960er)

Systeme in dieser Phase versuchten sich an „semantischer Informationsverarbeitung“ (und somit an einer Abgrenzung zum inhaltslosen Shannon'schen Informationsbegriff)¹, verwendeten aber überwiegend einfache Mittel (Muster, Heuristiken) sowohl für syntaktische als auch für semantische Verarbeitung. Beispiele hierfür sind **BASEBALL** und **SAD-SAM** (s.u. für mehr Information zu einzelnen Systemen).

Während es sich bei den Systemen dieser Phase überwiegend um einfache Frage-/Antwort- oder Dialog-Systeme handelte, hatte das etwas elaboriertere System **SIR** einen umfassenderen Anspruch: „This report discusses the problem of developing a computer[system] which 'understands.'“²³

¹ Ein Interessengebiet, das charakteristisch für die (damalige) sprachorientierte KI bzw. das oben angesprochene NLP ist.

² aus der Einleitung der Dissertation „SIR: a computer program for semantic information retrieval“ von Bertram Raphael, 1964.

³ Manaris charakterisiert „Verstehen natürlicher Sprache“ wie folgt: „A system 'understands' natural language, if, in response to some input, it creates a conceptual structure corresponding to that input, updates an existing conceptual structure, or makes an appropriate modification to a knowledge base“ (Manaris 1998, S. 23).

ELIZA

Der Vorläufer heutiger Chatterbots, **ELIZA** (*Weizenbaum* 1966), ist ein oft re-implementiertes Programm, das einem unbedarften Benutzer suggeriert, ein funktionierendes Dialogsystem zu sein. Es erscheint wie ein kompetenter Psychotherapeut, der sensitiv auf die Eingaben seines menschlichen Dialogpartners reagiert. Tatsächlich verbirgt sich hinter dieser Fassade nichts weiter als die Suche nach vorgegebenen Schlüsselwörtern mit Hilfe von Schablonen, für die bestimmte Reaktionen oder Reaktionsmuster festgelegt wurden (entsprechend handelt es sich nicht um ein wissensbasiertes System). Nebenbei werden automatisch bestimmte Ersetzungen für die Antwort vorgenommen.

Die folgende beliebige Auswahl solcher Muster aus einer beliebigen Implementation (hier die Python-Implementation des Natural Language Toolkits) soll dies veranschaulichen. Sie zeigt, wie eine Antwort allein aufgrund von Textschablonen (Kombination spezifischer Wörter/Wortfolgen und variabler Textteile) ausgewählt werden kann (man beachte den möglichen Rückbezug auf die variablen Teile).

Beispiele für Eingabe-/Antwort-Muster:

```
pairs = (
    (r'I am (.*)',
     ("Did you come to me because you are %1?", "How long have you been %1?", "How do you feel about being %1?")),
    (r'(.*) computer(.*)',
     ("Are you really talking about me?", "Does it seem strange to talk to a computer?", "How do computers make you feel?", "Do you feel threatened by computers?")),
    (r'(.*)',
     ("Please tell me more.", "Let's change focus a bit... Tell me about your family.", "Can you elaborate on that?", "Why do you say that %1?", "I see.", "%1."))
)
```

Eingabe-/Antwort-Ersetzungen:

```
reflections = {
    "am":"are", "was":"were", "i":"you", "i'd":"you would",
    "i've":"you have", "i'll":"you will", "my":"your",
    "are":"am", "you've":"I have", "you'll":"I will", "your":"my",
    "yours":"mine", "you":"me", "me":"you" }
```

Ambitionierte wissensbasierte Experimentalsysteme (1970er)

Winograd (1972) führt den Aspekt des Verstehens natürlicher Sprache, der die 1970er Jahre geprägt hat, bereits in seinem Titel. Winograds zentrale Annahme war, dass ein Leser eines Textes Wissen nutzt, um diesen zu verstehen. In Bezug auf die Sätze

The city councilmen refused to give the women a permit for a demonstration because they feared violence.

The city councilmen refused to give the women a permit for a demonstration because they advocated revolution.

wies er darauf hin, dass spätestens bei einer Übersetzung in eine Sprache wie das Französische eine Auflösung der Anapher „they“ notwendig wird, was wiederum ein tieferes Verständnis der ausgedrückten Gegebenheiten voraussetzt. Winograds System **SHRDLU** war das bis dato umfangreichste, das systematisch auf verschiedene Komponenten der Repräsentation sprachlichen und nicht-sprachlichen Wissens zurückgriff.

In der Folge wurde die Verwendung von Wissensrepräsentationen Standard in NLS. Die Systeme in dieser Phase waren bereits sehr viel komplexer als die vorhergehenden, aber selten ausgereift. Am Ende der 1970er Jahre stellte sich angesichts unterschiedlichster Ansätze und Formate außerdem heraus, dass die Frage, was unter ‚Wissensrepräsentation‘ überhaupt zu verstehen ist, völlig ungeklärt war.

Logifizierung und beginnende Kommerzialisierung (1980er)

Als Reaktion auf die unzufriedenstellende Situation im Bereich Wissensrepräsentation wurden die existierenden ad-hoc-Formalismen zur Wissensdarstellung zunehmend durch vereinheitlichende Logik-orientierte Formate ersetzt. In dieser Hinsicht konsequent war das **LILOG** (LInguistische und LOGische Methoden für das maschinelle Verstehen des Deutschen)-System, das einen vorerst letzten Versuch in Richtung Textverstehen unternahm. Mittlerweile war deutlich geworden, dass die Aufgabe des Textverstehens eine viel zu große für die gegenwärtige maschinelle Sprachverarbeitung war. Eine Randbemerkung: Momentan scheint die Zeit allerdings reif für eine Renaissance dieser Fragestellung zu sein:

Over the last two decades or so, Natural Language Processing (NLP) has developed powerful methods for low-level syntactic and semantic text processing tasks such as parsing, semantic role labeling, and text categorization. Over the same period, the fields of machine learning and probabilistic reasoning have yielded important breakthroughs as well. It is now time to investigate how to leverage these advances to understand text.

(Etzioni et al. 2007, S. 1)

In den 1980ern entstanden außerdem die ersten kommerziellen Systeme, z. B. **INTELLECT**, eine natürlichsprachliche Schnittstelle zu Datenbanken, und **METAL**, ein von Siemens (weiter)entwickeltes System für maschinelle Übersetzung.

Redefinition im Internetzeitalter (1990er ff)

Der mangelnde Erfolg wissenbasierter Systeme (prägnanter ausgedrückt: ihr Scheitern) führte dazu, die computerlinguistischen Methoden (wieder) mehr an der beobachtbaren sprachlichen Oberfläche auszurichten und zunehmend empirisch orientierte statistische Verfahren zu verwenden. Im Rahmen dieses „statistical turns“ in der CL und durch das gleichzeitige Aufkommen des Internets wurden die Ziele der Sprachtechnologie und die Aufgaben natürlichsprachlicher Systeme an realistische Anforderungen und Möglichkeiten angepasst. Gegenwärtige Entwicklungen sehen NLS –zumindest im Bereich einfacherer Aufgaben wie Fragebeantwortung– in erster Linie als Erweiterungen und Verbesserungen des Information retrieval.

Im Bereich komplexerer (z. B. Dialog-)Systeme wird Wissensbasiertheit als Verfahren weiter eingesetzt, jetzt allerdings verstärkt aus anwendungsorientierter Perspektive und im Rahmen realistischer Szenarien (siehe z. B. SMARTKOM).

Einige ausgewählte Systeme

Das Folgende ist eine repräsentative Auswahl natürlichsprachlicher Systeme (zu frühen Überblicken s. Simmons 1965 und Tennant 1981).⁴

- | | |
|-------|--|
| 1963: | BASEBALL (<i>Green et al.</i> 1963): Information Retrieval-Programm, das Baseball-Informationen auf einfache Fragen lieferte (s. auch den Exkurs auf Seite 76). |
| | SAD-SAM („Sentence Appraiser and Diagrammer – Semantic Analyzing Machine“) (<i>Lindsay</i> 1963): Erste syntaktische und semantische Analysen, um Sätze über Familienbeziehungen (z. B. „John’s father, Bill, is Mary’s son“) zu verarbeiten und in einer internen, Frame-artigen Datenstruktur (<i>family unit</i>) abzulegen, so dass die Beantwortung weiterer Fragen möglich wurde („Who is a grandchild of Mary?“). SAD-SAM ist ein negatives Musterbeispiel für Wissensdarstellungen, die weder allgemein noch auf andere Domänen übertragbar sind. |
| 1968: | SIR („Semantic Information Retrieval system“) (<i>Raphael</i> 1968): System, das natürlichsprachliche Eingaben analysierte und als Fakten ablegte. Es verfügte über einfache logischen Folgerungsmechanismen, die es für Inferenzen zur Fragebeantwortung verwendete. Somit ist es ein Vorläufer des späteren LILOG -Systems (s.u.). |
| 1969: | TLC („Teachable Language Comprehender“) (<i>Quillian</i> 1969): Vorläufer der <i>semantischen Netzwerke</i> als einem der bekanntesten Formate für Wissensrepräsentationen (s. hierzu <i>Carstensen</i> 2009); erzeugte interne Repräsentation für einen Text (allerdings von zweifelhaftem Format), jedoch ohne Deduktionsmöglichkeit. |
| 1972: | SHRDLU (<i>Winograd</i> 1972), s. auch den Exkurs auf Seite 30
LUNAR (<i>Woods et al.</i> 1972): Beantwortung von Fragen an (echte) Datenbanken zu (echtem) Mondgestein |

⁴ Es sollte klar sein, dass die Anzahl der Systeme in neuerer Zeit unüberschaubar wird.

1977:	GUS (<i>Bobrow et al.</i> 1977): Flugauskünfte, s. auch den Exkurs auf Seite 119 LIFER (<i>Hendrix</i> 1977): <i>Natürlichsprachliche Schnittstelle zu Datenbanken</i> ; Vorgänger des ersten kommerziellen Systems QA
1975,1978:	SAM („Script Applier Mechanism“) (Schank)/ PAM („Plan Applier Mechanism“)(Wilensky): Textverstehenssysteme (s. <i>Riesbeck und Schank</i> 1982), bei denen das Verstehen durch schematische Wissensstrukturen wie „Scripts“ und „Pläne“ geleitet war.
1976-80:	PLIDIS („Problem-Lösendes Informationssystem mit Deutsch als InteraktionsSprache“)(Institut für Deutsche Sprache, s. <i>Berry-Rogghe und Wulz</i> 1978): Frage-/Antwort-System zur Abwasserüberwachung
1978-81: Ende 70er:	HAM-RPM (<i>Jameson et al.</i> 1980): Dialogsystem, Hotelzimmerreservierung HEARSAY (<i>Erman et al.</i> 1980): Verstehen gesprochener Sprache; Verwenden einer zentralen Datenstruktur für die Vermittlung der Kommunikation zwischen den Systemkomponenten (<i>Blackboardarchitektur</i>) MÉTÉO übersetzt Wettervorhersagen
1981-1985:	HAM-ANS (Hamburger anwendungsorientiertes natürlichsprachliches System, <i>Hoeppner et al.</i> 1983): Dialogsystem; Nachfolger von HAM-RPM
1984:	INTELLECT (<i>Harris</i> 1984): Eine der ersten <i>kommerziellen</i> natürlichsprachlichen Schnittstellen zu Datenbanken, basierend auf dem Vorgängersystem ROBOT (<i>Harris</i> 1977)
1986-1991:	LILOG : Textverstehenssystem (<i>Herzog und Rollinger</i> 1991); das System analysierte Texte, nahm das darin enthaltene Wissen in seine Wissensbasis auf und sollte daraufhin in der Lage sein, Fragen zu den Texten zu beantworten (und zwar insbesondere auch solche, die nur implizit –d.h. durch Schlussfolgerungen/Inferenzen– logisch daraus ableitbar waren).
1993-2000:	VERBMOBIL : Übersetzungssystem für Spontansprache (spezifisch beschränktes Dolmetschsystem); u.a. im Bereich Terminabsprache (<i>Wahlster</i> 2000)
2000-2003:	SMARTKOM : Intelligente dialogische multimodale Benutzerschnittstelle zu Informationstechnologie-Anwendungen (<i>Wahlster et al.</i> 2001)
2004-2007	SMARTWEB : Synthese multimodaler Dialogsysteme (s. Kap. 10) und Systemen, die beliebige einfache Fragen mithilfe des Web beantworten (open-domain QAS, s. Kap. 7), s. <i>Wahlster</i> (2007)

3.3 Aspekte und Anforderungen

3.3.1 Qualitätskriterien

Nachdem in den Anfängen der CL überwiegend Einzelsysteme geschaffen wurden, deren Leistungen aufgrund mangelnder Evaluation(smöglichkeiten) schwer einzuschätzen waren,

SHRDLU

Das von Terry Winograd entwickelte Programm [SHRDLU](#) ist das erste komplexe wissensbasierte sprachverarbeitende System, das separate Komponenten für die Repräsentation sprachlichen und nicht-sprachlichen Wissens aufweist, um damit die verschiedensten Phänomene im Zusammenspiel von Sprache und Handeln in Bezug auf eine begrenzte Domäne (simulierte Blöcke-Welt, s. Abb. 3.1 auf der nächsten Seite) zu bewältigen.

Das System ist in der Lage, per simuliertem Greifarm die Welt zu verändern und jederzeit über den Zustand der Welt, die Eigenschaften der beteiligten Objekte und die eigenen Handlungen Auskunft zu geben und bei Unklarheiten angemessen zu reagieren:

- | | |
|----------------|--|
| Person: | Pick up a big red block. |
| Computer: | OK. (macht es) |
| Person: | Grasp the pyramid. |
| Computer: | I don't understand which pyramid you mean. |
| Person: | Find a block which is taller than the one you are holding and put it into the box. |
| Computer: | By 'it', I assume you mean the block which is taller than the one I am holding. |
| Computer: | OK. (macht es) |
| Person: | What does the box contain? |
| Computer: | The blue pyramid and the blue block. |
| Person: | What is the pyramid supported by? |
| Computer: | The box. |
| Person: | How many blocks are not in the box? |
| Computer: | Four of them. |
| Person: | Is at least one of them narrower than the one which I told you to pick up? |
| Computer: | Yes, the red cube. |
| Person: | Is it supported? |
| Computer: | Yes, by the table. |
| Person: | Can the table pick up blocks? |
| Computer: | No. |
| Person: | Can a pyramid be supported by a block? |
| Computer: | Yes. |
| Person: | Does a steeple |
| Computer | |
| <unterbricht>: | Sorry, I don't know the word „steeple“. |
| Person: | A „steeple“ is a stack which contains two green cubes and a pyramid. |
| Computer: | I understand. |
| Person: | Are there any steeples now? |
| Computer: | No. |
| Person: | Build one. |
| Computer: | OK. (macht es) |

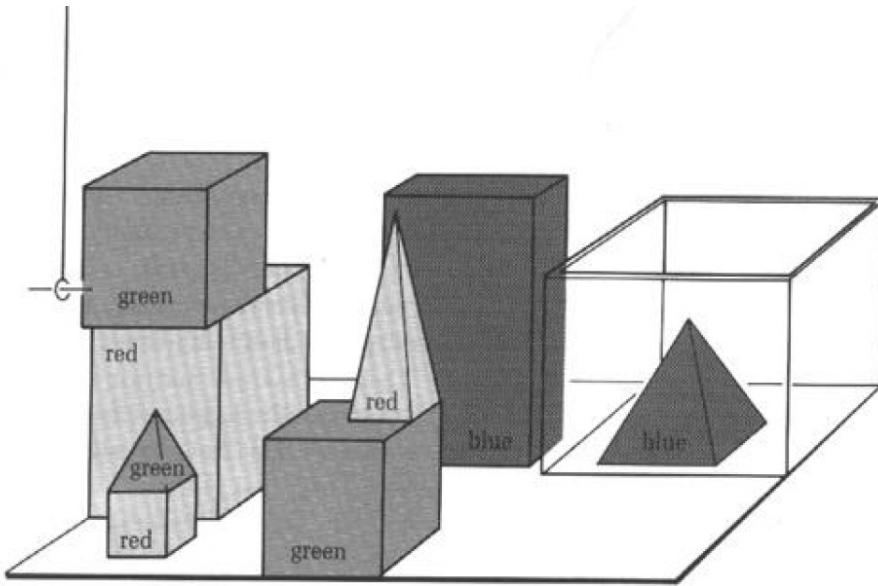


Abbildung 3.1: Simulierte Blöcke-Welt in SHRDLU

wird heute mit Blick auf eine wachsende Sprachtechnologie zunehmend auf eine Objektivierung der Bewertung eines Systems geachtet. Die Kriterien, die hierfür in Frage kommen, sind die folgenden (s. auch *Hahn* 1996):

- **Linguistische Abdeckung:**

Welche sprachlichen Phänomene können erfolgreich vom System behandelt werden? Während bei IR-nahen Anwendungen (z. B. Informationsextraktion) Lücken in der Abdeckung tolerierbar sind, führt ein Mangel an sprachlicher Kompetenz der NLS bei den Nutzern generell zu geringer Akzeptanz.

Mangelnde Einschätzung und Unsicherheit der Nutzer darüber, was ein NLS (verstehen) kann, haben oft dazu geführt, einfache (aber transparentere) Benutzeroberflächen in Systemen zu verwenden.

Einen Eindruck davon, was sich hinter dem Terminus „linguistische Abdeckung“ verbirgt, bietet *Hobbs* (2004).

- **Multilingualität:**

Bietet das System seine Fähigkeiten in mehreren Sprachen an?

- **Konzeptuelle Abdeckung:**

die Breite des Weltwissens bzw. der Behandlung nicht-sprachlicher Phänomene.

- **Robustheit:**

Inwieweit ist das System in der Lage, mit Problemen bei der Verarbeitung umzugehen. Im Idealfall führen Probleme zum graduellen Abfall der Systemleistung („graceful degradation“) und nicht zum Absturz.

- **Modularität und Transparenz:**

Inwieweit lassen sich die Systemkomponenten mit geringem Aufwand einzeln betrachten, auswerten, debuggen, austauschen, wiederverwenden?

- **Domänenunabhängigkeit (d. h., Übertragbarkeit/Portabilität):**
Lässt sich das System (mit geringem Aufwand) auch in anderen Domänen anwenden?
- **Einsatz für verschiedene Aufgaben (Flexibilität):**
Inwieweit ist das System auch für andere Aufgaben (als die aktuelle) verwendbar (bzw. leicht anzupassen)?
- **Effizienz und Korrektheit (Performanz):**
Arbeitet das System schnell? Arbeitet es korrekt (d.h. ohne Fehler)? Kann man sich auf das System verlassen (**Verlässlichkeit**), d.h. werden z. B. natürlichsprachliche Anweisungen richtig und vollständig umgesetzt?
- **Effektivität (Güte gemäß der funktionalen Anforderungen).**
Inwieweit ist das System in der Lage, (genau) das zu tun, was von ihm erwartet wird?
- **Sicherung der Effizienz im praktischen Einsatz (Skalierbarkeit):**
Arbeitet das System auch noch unter realistischen Anforderungen (große Anzahl von gleichzeitigen Nutzern) effizient oder bricht es unter der Last der Anfragen zusammen?
- **Gewährleistung der Usability (Gebrauchstauglichkeit) nach Prinzipien der Software-Ergonomie:**
Sind allgemeine Erkenntnisse dieses Bereichs berücksichtigt worden?
- **Aufwand für die Erstellung eines Industrie-konformen Systems (Produktivität):**
Wie viel kostet es an Zeit- und Geldaufwand, den Systemprototypen für den spezifischen Einsatz zu reimplementieren?
- **Nachhaltigkeit⁵ (Sustainability):**
Wie beständig sind die Ressourcen und Verfahren des Systems, so dass es mit wenig Aufwand gepflegt und/oder erweitert werden kann?

3.3.2 Probleme maschineller Sprachverarbeitung

Einer der Aspekte von NLS, der (Computer-)Linguisten überwiegend bekannt, Laien hingegen überwiegend unverständlich ist, betrifft die Frage, warum es überhaupt so schwierig ist, Systeme zu konstruieren, die auch nur annähernd über eine der menschlichen Sprachfähigkeit ähnliche Sprachkompetenz verfügen. Eine –wenn auch grobe– Antwort hierauf lässt sich anhand dreier grundlegender Problembereiche geben:

- der Sprachverständensproblematik

⁵ Den Hinweis auf diesen Punkt verdanke ich Georg Rehm.

- der Wissensverarbeitungsproblematik
- der Sprachgenerierungsproblematik.

Beim Sprachverstehen hat man es generell mit **Ambiguität** zu tun, d.h. zu einem Element existieren mehrere Möglichkeiten, es zu kategorisieren/zu interpretieren. Hier muss jeweils eine ausgewählt werden (*Disambiguierung*).

Bei der Wissensverarbeitung ist das zentrale Problem das der **Begrenzung** des inferentiellen Aufwandes für eine sprachbezogene Aufgabe: welche möglichen Schlussfolgerungen (*Inferenzen*) sollen gezogen werden?⁶ Dieses Problem tritt bei rein nicht-sprachlicher Verarbeitung auf (z. B.: welche Faktoren werden für die Auswahl einer geeigneten Route für eine Wegbeschreibung verwendet, s. *Carstensen 1991*)? Aber auch bei sprachlicher Verarbeitung –z. B. der noch selten versuchten maschinellen Verarbeitung literarischer Texte oder der Planung zusätzlicher Information für Wegbeschreibungen („Sie kommen übrigens an einer Kirche vorbei, die schon im 13. Jahrhundert...“)– stellt sich die Frage nach einer sinnvollen Begrenzung möglicher Inferenzen. Eine Voraussetzung für eine erfolgreiche Wissensverarbeitung ist außerdem eine angemessene **Repräsentation** des für ein NLS relevanten Wissens (s. hierzu auch Abschnitt 3.4.1 auf Seite 38).

Die Sprachgenerierung ist generell durch das Problem der **Auswahl** gekennzeichnet, dem zum Einen die verschiedensten Möglichkeiten der *Perspektivierung* einer thematisierten Gegebenheit zugrunde liegt (z. B. ein Glas Wasser als 'halb voll' oder 'halb leer' darstellen, eine Person als 'kritisch' oder 'nörglerisch' charakterisieren, eine Situation als 'verkaufen von X durch Y an Z' oder als 'abkaufen des X von Y durch Z' beschreiben), zum Anderen die uneingeschränkte Möglichkeit, zu einem Gesprächsgegenstand weitere Hintergrundinformation hinzuzufügen. Bei den *Entscheidungen*, die hierfür getroffen werden müssen, sind eine Reihe von Aspekten der Kommunikationssituation zu berücksichtigen ((gemeinsames) Vorwissen von Sprecher und Hörer, sozialer Status, Zeitdruck etc.); gleichzeitig muss der inhaltliche und sprachliche Zusammenhang (*Kohärenz* und *Kohäsion*) des Resultats gewährleistet sein.

Die **Sprachverstehensproblematik** umfasst die Probleme, die bei der Analyse sprachlicher Eingaben zu bewältigen sind (in etwa: „Was meinte der Sprecher mit dem, was er sagte?“). *Schukat-Talamazzini (1995, S. 8ff)* nennt in Bezug auf die Spracherkennung die folgenden: *Kontinuität*, *Variabilität*, *Komplexität* und *Ambiguität*.

Gesprochene Sprache ist gekennzeichnet durch die **Kontinuität** des Sprachschalls, in dem Wort- Phrasen- oder Satzgrenzen schwer zu identifizieren sind. Hinzu kommt die artikulatorische Verschleifung beim Sprechen (so kann z. B. *Flying planes can be dangerous* als *Fly in planes can be dangerous* missverstanden werden oder umgekehrt). Dieses Problem der **Segmentierung** tritt auch in der Schriftsprache auf. Hier ist die Trennung von Wörtern weniger problematisch, dafür ist die Trennung von Sätzen nicht immer eindeutig (in der Regel durch Verwendung des Punktes als Satzendekennzeichner oder als Abkürzungspunkt), da Information über die Satzmelodie (Prosodie) fehlt.

⁶ Dies ist eine der Sichtweisen auf das, was in der Künstlichen Intelligenz das „**Frame Problem**“ genannt wird, s. dazu *Dennett (1984)* und den Exkurs auf der nächsten Seite.

Das Frame Problem

Once upon a time there was a robot, named R1 by its creators. Its only task was to fend for itself. One day its designers arranged for it to learn that its spare battery, its precious energy supply, was locked in a room with a time bomb set to go off soon. R1 located the room, and the key to the door, and formulated a plan to rescue its battery. There was a wagon in the room, and the battery was on the wagon, and R1 hypothesized that a certain action which it called PULLOUT(WAGON,ROOM) would result in the battery being removed from the room. Straightaway it acted, and did succeed in getting the battery out of the room before the bomb went off. Unfortunately, however, the bomb was also on the wagon. R1 knew that the bomb was on the wagon in the room, but didn't realize that pulling the wagon would bring the bomb out along with the battery. Poor R1 had missed that obvious implication of its planned act.

Back to the drawing board. „The solution is obvious,“ said the designers. „Our next robot must be made to recognize not just the intended implications of its acts, but also the implications about their side-effects, by deducing these implications from the descriptions it uses in formulating its plans.“

They called their next model the robot-deducer R1D1. They placed R1D1 in much the same predicament the R1 had succumbed to, and as it too hit upon the idea of PULLOUT(WAGON,ROOM), it began, as designed to consider the implications of such a course of action.

It had just finished deducing that pulling the wagon out of the room would not change the color of the room's walls, and was embarking on a proof of the further implication that pulling the wagon out would cause its wheels to turn more revolutions than there were wheels on the wagon - when the bomb went off.

Back to the drawing board. „We must teach it the difference between relevant implications and irrelevant implications,“ said the designers. „And teach it to ignore the irrelevant ones.“ So they developed a method of tagging implications as either relevant or irrelevant to the project at hand, and installed the method in their net model, the robot-relevant-deducer, R2D1.

When they subjected R2D1 to the test that had so unequivocally selected its predecessors for extinction, they were surprised to find it sitting, Hamlet-like, outside the room containing the bomb, the native hue of its resolution sickled o'er with the pale case of thought, as Shakespeare has aptly put it. „DO something!“ its creators yelled.

„I am,“ it replied. „I'm busily ignoring some thousands of implications I have determined to be irrelevant. Just as soon as I find an irrelevant implication, I put it on the list of those I must ignore, and...“ the bomb went off.

(aus: *Dennett (1984)*)

Variabilität bezeichnet die Tatsache, dass linguistische Einheiten akustisch auf unterschiedlichste Weise realisiert sein (vulgo: sich anhören) können, bedingt/beeinflusst durch den Übertragungskanal (z. B. Telefon), Umgebungsgeräusche, Sprechweise (z. B. Tempo, Anspannung), individuelle, physische Sprechermerkmale (Alter, Geschlecht, Vokaltraktanatomie, Gesundheitzzustand) und habituelle, linguistische Sprechermerkmale (Dialekt, Soziolekt, Idiolekt). Vor allem wird die Aussprache eines Lautes durch dessen aktuale Umgebung im Wort/Satz beeinflusst. Die Variabilität erhöht sich, wenn man zusätzlich die Realisierungsmöglichkeiten geschriebener Sprache in Betracht zieht. Ein Beispiel ist die George Bernard Shaw zugeschriebene Frage nach der Aussprache von „ghoti“, mit der dieser Umstand persifliert werden soll. Die Antwort ist die Aussprache von „fish“, die sich wie folgt zusammensetzt: das ‚f‘ aus ‚lauGH‘, das ‚i‘ aus ‚wOmen‘ und das ‚sh‘ aus ‚naTion‘.

Mit **Komplexität** sind die technischen Aspekte des Speichern und Verarbeitens von Information gemeint. Hierzu gehört einerseits die Erstellung akustischer Modelle, die die Aussprachemöglichkeiten von Wörtern darstellen und andererseits der Verarbeitungsaufwand beim Abgleich des Sprachinputs mit den Modellen. Die Komplexität erhöht sich, je feiner die Analyse ist (z. B. in Form höherer Abtastraten) und je mehr Merkmale verwendet werden, um die verschiedenen Faktoren zu repräsentieren, aus denen die Variabilität resultiert.

Ambiguität bezeichnet zunächst die zur Variabilität konverse Tatsache, dass eine bestimmte sprachliche Realisierung mehr als einer linguistischen Einheit zugewiesen werden kann (z. B., entspricht die Lautfolge [ra:t] dem Wort ‚Rat‘ oder –wegen Auslautverhärtung– dem Wort ‚Rad‘?). Eine Folge davon sind Miss-Verständnisse wie das von *die alte Minna* bei einer faktischen Äußerung von *die alten Männer*.

Diese Beobachtung kann verallgemeinert werden: Ambiguität lässt sich als das dann entstehende Problem bezeichnen, wenn eine *linguistische Einheit* auf verschiedene Weisen klassifiziert werden kann. Ambiguität ist deswegen ein notorisches Problem in der Computerlinguistik, weil sie gleichzeitig auf verschiedenen Ebenen der Analyse bestehen kann (s. den Exkurs auf der nächsten Seite).

Ein Großteil der Methoden, die in der Computerlinguistik entwickelt worden sind, befasst sich mit der Auflösung der Ambiguität mehrdeutiger Einheiten. Einige nutzen direkt die Beschränkungen (**Constraints**) aus, die durch weitere vorliegende Information gegeben ist (z. B. disambiguieren ‚der Mann‘ in *der Mann geht* die Ambiguität von ‚geht‘). Andere vermeiden die oft schwierige/unmögliche Disambiguierung und ermöglichen eine Weiterverarbeitung durch **Unterspezifikation** (z. B. bzgl. der Skopusbewegungen in der Satzsemantik). Schwierige Probleme wie das der Auflösung von Anaphern (**Anaphernresolution**) wie *er, sie, das, der Mann* bilden eigene Forschungsgebiete mit unterschiedlichsten Lösungsansätzen.

Weitere Aspekte der Sprachverständensproblematik sind

- die Erkennung/Identifizierung von Fehlern
- das Rekonstruieren ausgelassener Information
- das Entdecken von Fehlannahmen

Eine der Fehlkonzeptionen rationalistischer Theorien war es, davon auszugehen, dass man für

Ambiguitätsarten

Anhand linguistischer Phänomene können mindestens die folgenden Arten von Ambiguität unterschieden werden:

- Phonetische Ambiguität (Homophonie)
 - *Miene* vs. *Mine*, *mehr* vs. *Meer*
- Orthographische A. (Homografie)
 - *umfahren* vs. *úmfahren*, *Móntage* vs. *Montáge*
- Morphologische A.
 - *Staubecken*: Stau-becken vs. Staub-ecken
 - *Basisversorgungsantragsformular*
- Lexikalische A. (Homonymie)
 - *Band*, *Ball*, *Bank*
- Morpho-syntaktische A.
 - *geht* [singular oder plural?]
- Wortart-A.
 - *das* [Artikel, Relativpronomen...?]
- Strukturell-syntaktische A.
 - *Paul sah den Mann auf dem Hügel mit dem Teleskop*
 - *Franz hat den Torwart mit den blauen Handschuhen erschlagen*
 - *Paula lässt die Verwandten aus Köln herzlich grüßen*
 - *Flying planes can be dangerous*
 - *Time flies like an arrow*
- Satzsemantische A. (Skopuss-A.)
 - *Jeder CL-Student muss zwei Programmiersprachen beherrschen, die für das Studium relevant sind* [sind es insgesamt 2 oder mehr?]
 - *3 Studies sind an 2 haarigen Fragen in der Prüfung gescheitert*
- Semantische A. (Mehrdeutigkeit, Polysemie)
 - *Brücke*, *Schlange* etc.
 - *Schule*
[vgl.: ... wurde 1911 gebaut, ... ist eine nützliche Einrichtung etc.]
- Pragmatische A.
 - *Ganz schön kühl hier drin*
[Start eines Smalltalks, Aufforderung, die Heizung aufzudrehen...?]
- Anaphern- A.
 - *Wenn dein Mitarbeiter dir wieder einen Beschwerdebrief schickt, dann falte ihn mal richtig zusammen!*

ein NLS nur korrekte Strukturen zu spezifizieren braucht, bei denen Inkorrektheit als Nebenprodukt geliefert wird. Allerdings ist schon die exhaustive Spezifikation korrekter Strukturen schwierig und führte zu den angesprochenen Problemen mit der Robustheit entsprechender Systeme. Außerdem ist eine Markierung einer fehlerhaften Eingabe als „nicht verarbeitbar“ unzureichend. In Bereichen wie Rechtschreib-/Grammatik-Korrektur, Tutoriellen/Lehr-/Lern-Systemen oder in Dialogsystemen kommt es vielmehr darauf an, Fehler zu lokalisieren, zu identifizieren und entsprechend auf sie zu reagieren. Dies kennzeichnet **Fehlerbehandlung** als eigenständige Aufgabe des Sprachverstehens.

Nicht alle Information, die per Sprache übermittelt werden soll, muss auch sprachlich ausgedrückt werden. Daher existieren eigentlich fehlerhafte, aber zugelassene (und somit wieder grammatisch korrekte) syntaktische Auslassungen (**Ellipsen**), die im Verlauf des Sprachverständnisses erkannt und korrekt rekonstruiert werden müssen, z. B. im folgenden Diskurs:

A: *Ich glaub⁷, ich spinn.*

B: *Ich auch.*

(„Ich glaube auch, ich spinne“ vs. „Ich glaube auch, dass du spinnst“).

Nicht alles von dem, was per Sprache übermittelt wird, aber nicht sprachlich ausgedrückt werden muss, ist durch Auslassung gekennzeichnet. Mit anderen Worten, in einer Äußerung wird vieles implizit vorausgesetzt („präsupponiert“), ohne dass es explizit als wahr angegeben („assertiert“) wird. Ein Beispiel: bei *Mein Mentaltrainer hat seine Nervenkrise überwunden*. kann der Hörer davon ausgehen, dass der Sprecher einen Mentaltrainer hat und dass dieser (gerade) eine Nervenkrise hatte. Eine der Aufgaben beim Sprachverständnis ist es, die **Präsuppositionen** einer Äußerung zu erkennen, zu überprüfen, mit bisherigem Wissen zu verknüpfen (zu *binden*) oder stillschweigend dem eigenen Wissen hinzuzufügen (zu *akkomodieren*).

Dabei kann es vorkommen, dass ein Sprecher von falschen Annahmen ausgeht, die von einem NLS als **Präsuppositionsverletzungen** erkannt werden müssen. Hier ist zwischen solchen falschen Annahmen zu unterscheiden, bei denen die Präsposition in der gegebenen Situation nicht zutrifft („falsche Präsposition“) und solchen, bei denen die Präsposition prinzipiell nicht erfüllt sein kann („Fehlannahme (misconception)“):⁸

> *Können Sie mir sagen, wie ich meinen Vertrag vorzeitig kündigen kann?*

Sie können ihren Vertrag leider gar nicht vorzeitig kündigen. [falsche Präsposition]

> *Können Sie mir sagen, wie viel eine Festplatte mit 2 GB RAM kostet?*

Nein, meinen Sie den Preis einer Festplatte oder den von 2GB Arbeitsspeicher? Arbeitsspeicher und Festplattenspeicher sind unterschiedliche Dinge. [Fehlannahme eines Computerlaien]

⁷ Man beachte hier auch die in der Umgangssprache erlaubte Auslassung des 'e' (Apokope).

⁸ Die faktisch korrekte Antwort „Nein.“ wäre eine äußerst benutzerunfreundliche Reaktion auf die falschen Annahmen des Kunden.

3.4 Architekturen und Systeme

3.4.1 Systeme mit tiefer Verarbeitung

Abb. 3.2 zeigt die generische Architektur von wissensbasierten NLS (s. auch *Wahlster 1982*). Sie veranschaulicht die Aspekte des **Verstehens** (Analyse einer natürlichsprachlichen Eingabe, Konstruktion einer Bedeutungsrepräsentation dieser Eingabe, Verarbeitung der Repräsentation) und der **Generierung** (Planung einer Ausgaberepräsentation, Synthese einer entsprechenden natürlichsprachlichen Ausgabe).

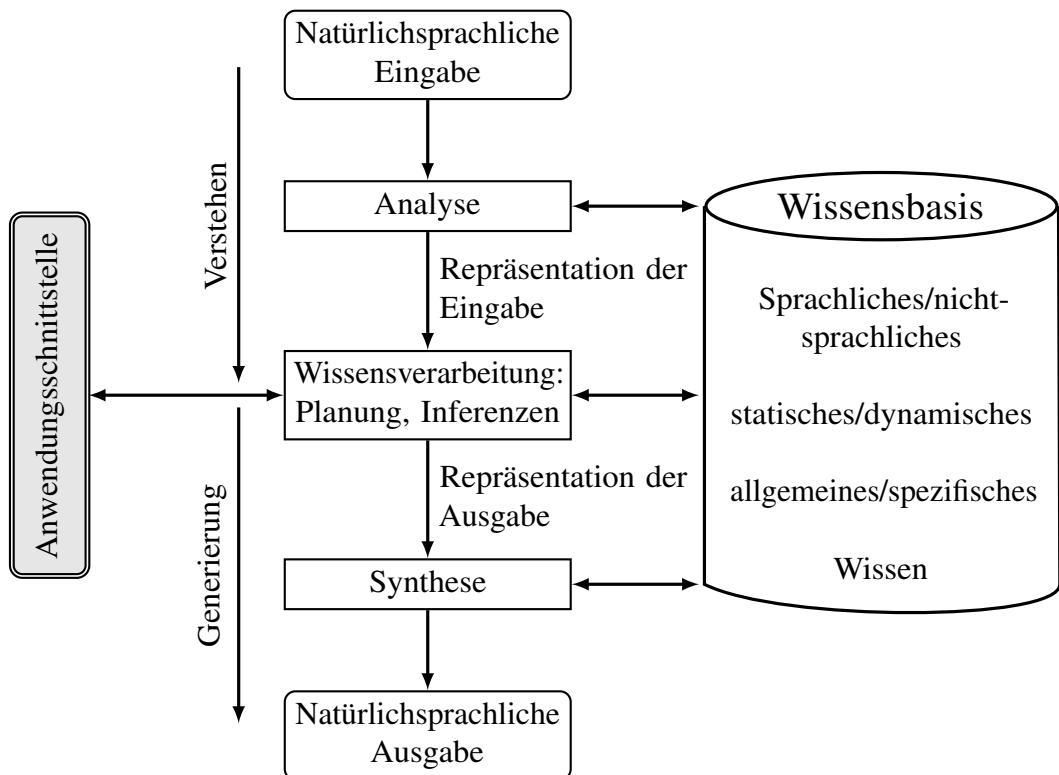


Abbildung 3.2: Generische Architektur eines NLS

Die zentrale Komponente ist die der Wissensverarbeitung, die zum Einen bei der Interpretation und Produktion von Sprache involviert ist, aber auch das Schlussfolgern über Weltwissen sowie die Planung von Handlungen wie dem Umgang mit Geräten/Anwendungen umfasst. Das in einem NLS repräsentierte Wissen ist in der so genannten Wissensbasis abgelegt. Es lässt sich mindestens anhand der folgenden Aspekte unterteilen: *Modalitätsart*, *Wissenstyp*, *Spezifizität*, *Speicher-/Gedächtnistyp*.

Wissensbasis:Modalitätsart

Die Wissensbasis enthält sowohl sprachliches als auch nicht-sprachliches Wissen. Auf sie wird an verschiedener Stelle zugegriffen (z. B. für die Disambiguierung der Frage, die Konstruktion einer Antwort, die Planung von Handlungen). Diese Wissensbasis ist von externen Datenquellen zu unterscheiden, auf die das NLS über bestimmte (z. B. Datenbank-)Anwendungen Zugriff hat (z. B. für die Beantwortung von Fragen).

Auch wenn die sprachliche Modalität in NLS eine wesentliche Rolle spielt, nimmt die Bedeutung anderer Modalitäten (Gestik, Mimik etc.) zu, insbesondere im Rahmen der Multimodalen Systeme (s. Kap. 10). Dort werden auch in anderen Modalitäten Schemata, Regeln oder Instanzen für bestimmte Verhaltensweisen gespeichert. Neben modalitätsspezifischen Repräsentationen nimmt man zudem modalitätsunspezifische („amodale“) vermittelnde konzeptuelle Strukturen an, die z. B. in der Maschinellen Übersetzung eine Rolle als Interlingua spielen.

Wissensbasis:Wissenstyp

Es gibt eine Reihe verschiedener Kenntnistypen, über das ein NLS verfügt. Hierzu gehören sprachliche Kenntnis (lexikalisches, morphologisches, syntaktisches, semantisches und pragmatisches Wissen sowie Wissen über die Struktur und Eigenschaften von Dialogen und Texten), Wissen über die Welt („Weltwissen“), Inferenzwissen (welche Schlüsse gezogen werden können), Handlungswissen, räumliches Wissen etc., sowie Wissen über die Aufgaben, Aspekte oder Benutzer des NLS.

Da sie ihre jeweiligen Phänomene im System abbilden, spricht man von den mit diesen Wissenstypen assoziierten Prinzipien, Regeln und Repräsentationen auch als entsprechenden *Modellen* (also z. B. Sprachmodell, Dialogmodell, Weltmodell, Domänenmodell, Aufgabenmodell, Benutzermodell etc.).

Wissensbasis:Spezifizität

Diese Kategorie bezeichnet den Allgemeinheitsgrad repräsentierten Wissens, das hier anhand des Weltwissens erläutert werden soll. Eine einfache Unterscheidung ist die zwischen generischem und spezifischem Wissen.

Zu *generischem* Wissen gehören solche Wissensanteile, die so allgemein sind, dass sie in jedem Fall in einem NLS benötigt werden.

Spezifisches Wissen hingegen umfasst die Kenntnis von Dingen, die nicht allgemein gelten, sondern z. B. nur in speziellen Bereichen (*Domänen*) oder in spezifischen Situationen/Kontexten (entsprechend unterscheiden sich beispielsweise die diskursbereichsspezifischen Wissensanteile in Flugauskunftssystemen und tutoriellen Systemen zur Botanik).

Unterschieden wird auch zwischen ontologischem, begrifflichem und Fakten-Wissen:

Ontologisches Wissen besteht aus den basalen Kategorien/Unterscheidungen (Raum, Zeit, Objekt vs. Masse, begrenzt/unbegrenzt etc.), mithilfe derer wir als Menschen die Welt erfassen, und die entsprechend dem repräsentierten Wissen in NLS zugrunde liegen.

Begriffliches (oder konzeptuelles) Wissen charakterisiert die Dinge der Welt (Hund, Katze, Maus etc.), mit denen ein NLS im weitesten Sinn umgehen können soll. Hierzu gehören die Eigenschaften dieser Dinge, einerseits notwendige (Hunde sind Säugetiere), andererseits

typische, die für jede Instanz der jeweiligen Klasse *per Default* angenommen werden können (Vögel können typischerweise fliegen). Dabei gilt, dass spezifischere Information die Defaults der Oberklasse „überschreiben“ kann (Pinguine sind Vögel, fliegen allerdings nicht). Begriffliches/konzeptuelles Wissen ist hierarchisch organisiert (daher: *Konzepthierarchien*, auch: **Ontologien**).

Faktenwissen (oder assertionales Wissen) enthält Aussagen über konkrete Dinge (Peter ist ein Mensch, Tweety ist ein Vogel) und deren Eigenschaften, die wiederum den typischen Annahmen widersprechen können (Tweety hat einen Flügel gebrochen, kann also nicht fliegen).

Abb. 3.3 zeigt einen Ausschnitt des intern darzustellenden Wissens, das einem Satz wie *Peter isst Erdnüsse im Cinestar* zugrunde liegt (s. hierzu Carstensen 2009).

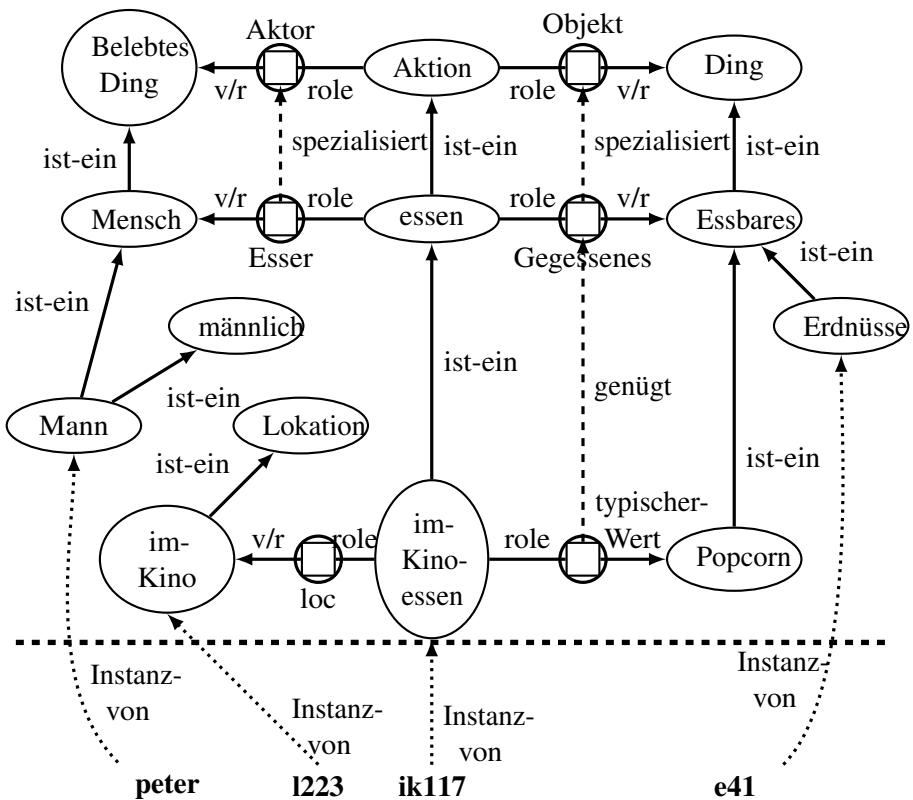


Abbildung 3.3: **Wissensrepräsentationsstruktur für den Satz Peter isst Erdnüsse im Cinestar**

Wissensbasis: Speicher-/Gedächtnistyp

Entsprechend der Unterscheidung von Langzeit- und Kurzzeitgedächtnis wird zwischen *statischem* (das bisher beschriebene) und *dynamischem* Wissen unterschieden. Dynamisches Wissen bezieht sich auf Eigenschaften der aktuellen Verarbeitung, also z. B. auf die Buchführung bzgl. des Verlaufs eines Dialogs (Dialog-/Diskurshistorie) und der Eigenschaften des aktuellen Benutzers/Redepartners.

Analyse

Die **tiefe Analyse** besteht aus den in Abb. 3.4 auf der nächsten Seite⁹ dargestellten Verarbeitungsschritten. Anhand von Sprach- oder Schrifterkennung werden die Wortformen der Eingabe erkannt (bei Texten wird hier auch von Textnormalisierung gesprochen); deren Struktur wird analysiert (**morphologische Verarbeitung**) und führt zu einer Liste der mit morphosyntaktischer Information angereicherten Wörter eines Satzes. Die Analyse der Satzstruktur (**syntaktische Verarbeitung/ Parsing**) ist üblicherweise die Eingabe für die **semantische Verarbeitung** (Semantikkonstruktion), die der Syntaxstruktur eine –heute normalerweise unterspezifizierte– Bedeutungsrepräsentation zuweist. Durch die **pragmatische Verarbeitung** wird der Bereich semantischer Interpretationen auf die in der jeweiligen Situation (bezogen auf dem Wissen über den aktuellen Text/Diskurs/Dialog bzw. über die Welt) angemessene eingeschränkt.

Terminologische Vorsicht ist geboten bzgl. der Interpretation von „Analyse“ und „Verstehen“. Beispielsweise wird im Bereich der Verarbeitung gesprochener Sprache unter „Analyse“ zum Teil die Abbildung von gesprochener Sprache auf Text (eigentlich: **Spracherkennung/speech recognition**) verstanden, während „Verstehen“ dort der Analyse in Abb. 3.2 entspricht.

Synthese

Analog zur Analyse muss man zwischen einer Synthese im weiteren Sinn (der systematischen Konstruktion sprachlicher Strukturen) gemäß Abb. 3.2 und der Synthese im engeren Sinn (Sprachsynthese) als –noch übliche– Abbildung eines Textes auf eine entsprechende Lautform unterscheiden. Die engere Interpretation wird erst durch in der Forschung befindliche Ansätze zur *concept-to-speech synthesis* erweitert.

3.4.2 Systeme mit flacher Verarbeitung

Eine **flache Analyse** unterscheidet sich strukturell kaum von tiefer Analyse (s. Abb. 3.5 auf Seite 43). Die wesentlichen Unterschiede betreffen die folgenden Punkte:

- es werden soweit möglich keine verschiedenen internen Repräsentationen verwendet, sondern es wird eine Input-Repräsentation mit Annotationen angereichert
- es werden möglichst (komputationell) einfache Verfahren angewendet
- Verarbeitung ist nicht notwendigerweise vollständig, sondern in der Regel *partiell*

Die **Tokenisierung** ist die Zerlegung des Inputs in einzelne Elemente (ggf. mit entsprechender Vorverarbeitung), im einfachsten Fall in Wörter und Satzzeichen. Durch das **Wortart-Tagging** (POS-Tagging) werden den Token Wortarten zugewiesen (gemäß bestimmter vorgegebener Wortartenmengen, sog. *POS-Tag-Sets*). Beim **Chunk-Parsing** werden größere Einheiten identifiziert (z. B. einfache Nominalphrasen), das **semantische Tagging** weist inhaltliche Label zu. Das **pragmatische Tagging** umfasst Aspekte flacher Diskursverarbeitung, z. B. Dialogatkennzeichnung bei Dialogen oder die Auflösung von Koreferenzen (d.h. zwei Ausdrücke beziehen sich auf dasselbe Objekt, wie „Peter“ und „der Mann“).

⁹ Diese Abbildung geht auf eine Folie von Hans Uszkoreit zurück.

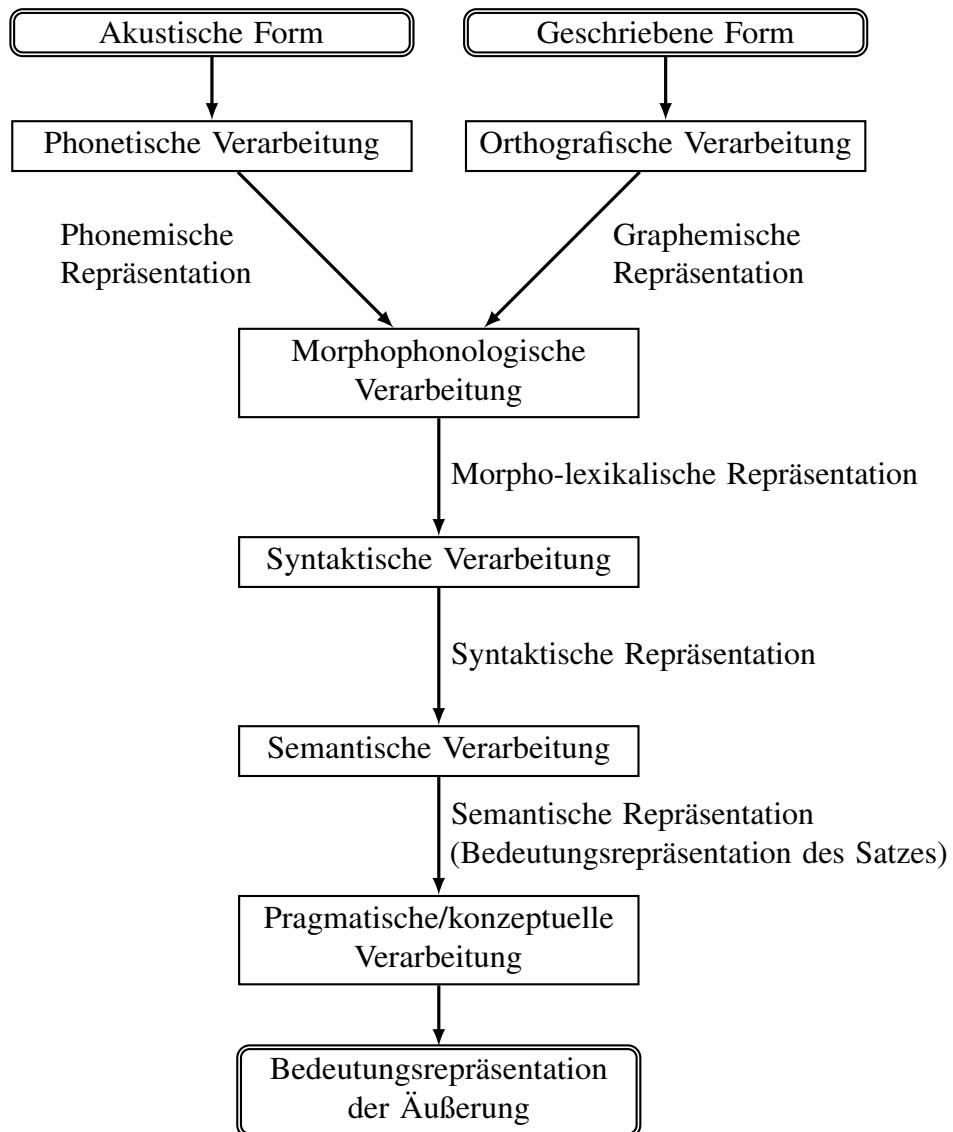


Abbildung 3.4: **Tiefe Analyse**

Auf flacher Verarbeitung beruhende Systeme verzichten generell auf die zentrale Verarbeitung und auf eine elaborierte Wissensbasis. Statistisch-basierte Systeme unterscheiden sich zusätzlich darin, dass –neben spezifischen Ressourcen– anstatt auf eine Wissensbasis grundsätzlich nur auf (annotierte) Sprachdaten (**Korpora**) zugegriffen wird. In einer Trainingsphase werden Modelle relevanter Sprachinformationen erstellt (s. Abb. 3.6 auf der nächsten Seite), die dann z. B. zur Auflösung von Ambiguitäten bei der Sprachverarbeitung verwendet werden.

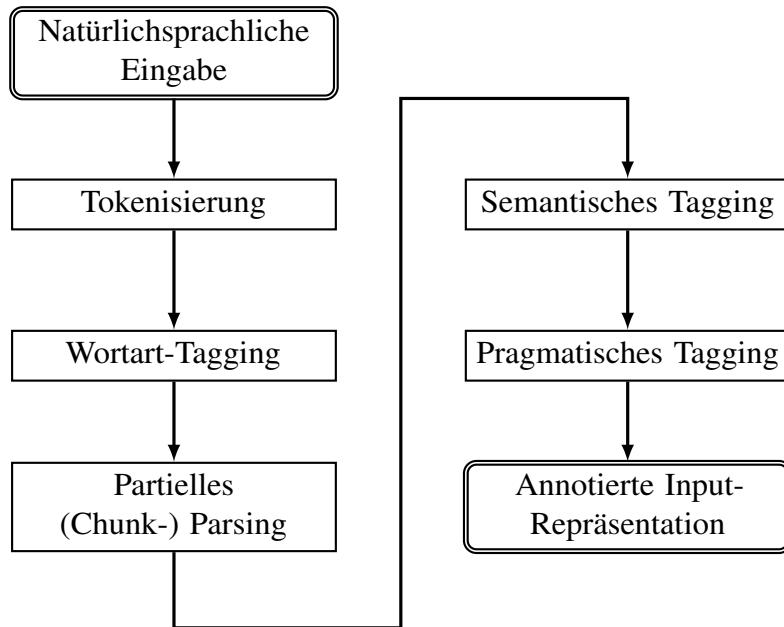


Abbildung 3.5: Flache Analyse

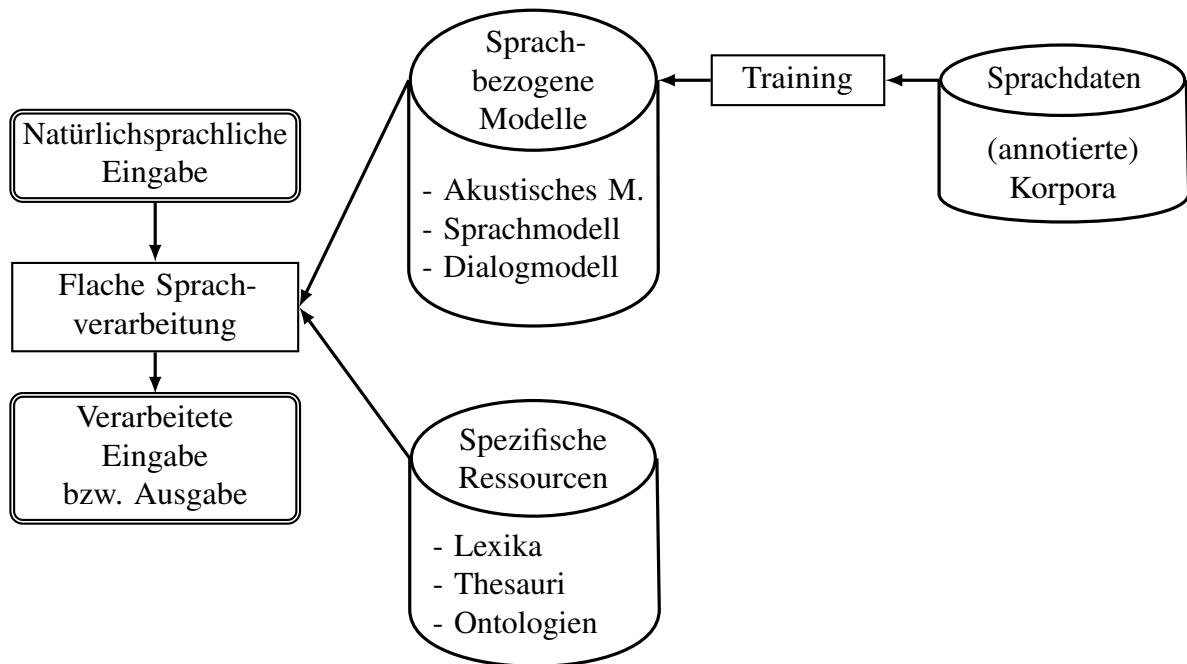


Abbildung 3.6: Struktur Korpus-basierter Systeme

3.4.3 Andere Architekturen

Es gibt zunehmend Mischformen (**hybride Systeme**) tiefer und flacher Verarbeitung: so wurde in **VERBMOBIL**, einem System zum maschinell gestützten Dolmetschen (s. *Wahlster 2000*), eine Architektur mit „parallelen“ Verläufen flacher und tiefer Verarbeitung realisiert. Statistische Übersetzung war dabei nur eines von zwei flachen Verfahren (s. Abb. 3.7)¹⁰. In modernen multimodalen Systemen, in denen außer Sprache auch z. B. Gestik und Mimik verarbeitet wird, ist die generische Architektur um zusätzliche Komponenten erweitert (s. dazu Kap. 10).

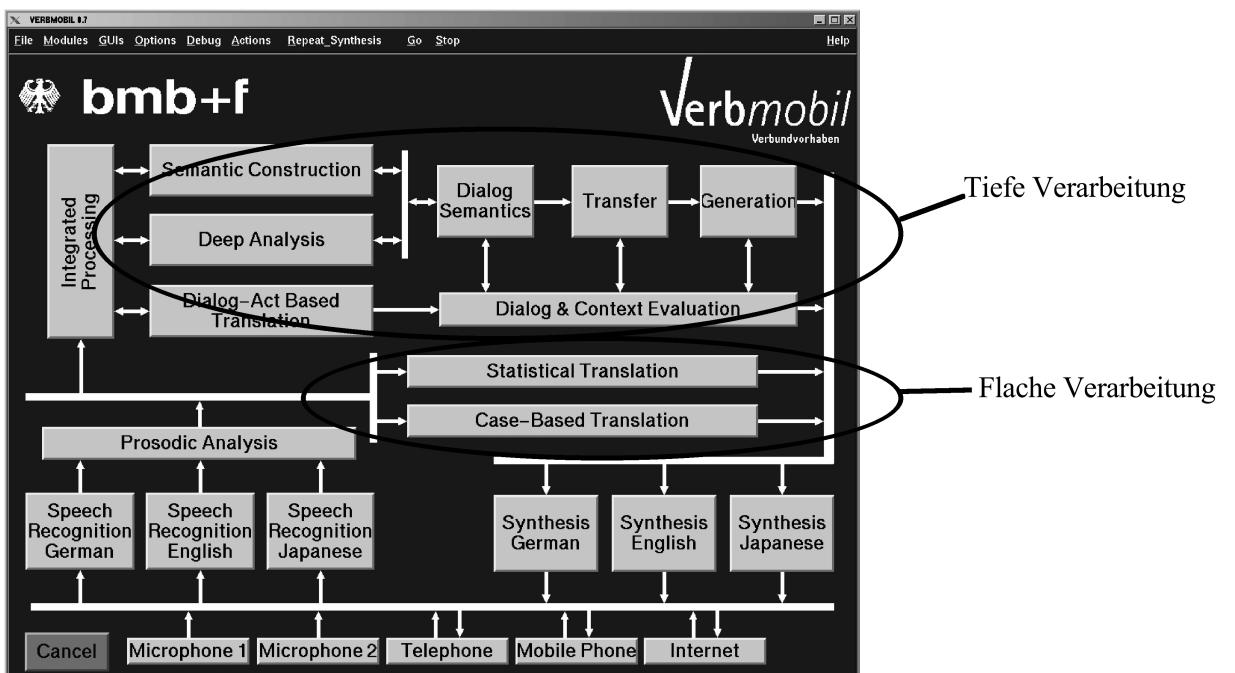


Abbildung 3.7: Architektur von **VERBMOBIL**

Anders als die klassischen natürlichsprachlichen Systeme mit ihrer linearen Abfolge in Bezug auf Daten- und Kontrollfluss (*pipeline-Architektur*)¹¹ weisen moderne Systeme eine flexiblere Architektur auf. Die Variationsbreite reicht von Systemen mit einem Blackboard, bei dem alle Komponenten Zugriff auf einen Pool von Daten/Repräsentationen haben (z. B. das **HEARSAY-II**-System, s. *Erman et al. 1980*), über Systeme mit einer zentralen Kontrollkomponente (vgl.

¹⁰ Abgebildet ist die Benutzeroberfläche des Forschungsprototypen von **VERBMOBIL** (aus: Wolfgang Wahlster, *VERBMOBIL Erkennung, Analyse, Transfer, Generierung und Synthese von Spontansprache*, auf der auch die Architektur des Systems abgebildet ist).

¹¹ S. aber *Wahlster* (1982, S. 20f) zu frühen Architekturmodell-Alternativen wie dem hierarchischen Modell (mit zentraler Steuerungskomponente), dem Kaskadenmodell (lokales Feedback zwischen linear geordneten Komponenten) und dem heterarchischen Modell (keine Hierarchie, keine festgelegte Verarbeitungsrichtung).

das „Whiteboard“ in der RAGS¹²-Architektur, s. *Cahill et al.* 2001; die vermittelnde Instanz („Hub“) in der GALAXY-II-Architektur, s. *Seneff et al.* 1998; oder den „Facilitator“ der Open Agent Architecture „OAA“, s. *Martin et al.* 1999) bis hin zu Multi-Blackboard-Architekturen mit verteilten, eigenständigen Komponenten wie in SMARTKOM (*Wahlster* 2006a).¹³

3.5 Ausblick

Als Trends in der Sprachtechnologie lassen sich die folgenden Entwicklungen nennen:¹⁴

Zunehmend wichtige Rolle empirischer Daten

Die Verwendung flacher, auf statistischen Methoden basierenden Verfahren verlangt die Existenz umfangreicher Datenkorpora als Ressourcen für entsprechende Lernverfahren. Aus diesem Grund avanciert die Sammlung empirischer Daten zu einem notwendigen Schritt in der Entwicklung von NLS.

Dekomposition in kleine Verarbeitungsschritte

Die Geschichte zeigt, dass der Ansatz, die Komplexität maschineller Sprachverarbeitung mithilfe komplexer holistischer Systeme zu bewältigen, momentan kaum erfolgreich ist. Eine gegenwärtig verfolgte Alternativstrategie besteht darin, diese Komplexität systematisch auf eine Menge weniger komplexer Verarbeitungsschritte zu reduzieren, durch die Strukturen verschiedener Repräsentationsebenen eines NLS miteinander in Beziehung gesetzt werden. Diese Verarbeitungsschritte stellen sich als Notationstransformationen dar, die einzeln gelernt werden können. Auf diese Weise wird nicht nur die Konstruktion von NLS handhabbarer gemacht, sondern es wird zudem die Möglichkeit geschaffen, die einzelnen Bestandteile der Sprachverarbeitung unterschiedlich zu (re)komponieren und so NLS für verschiedene Aufgaben zu konfigurieren bzw. adaptieren.

Zunehmend wichtige Rolle komparativer Evaluation (*Evaluationskonferenzen*) zur Objektivierung von Technologiefortschritt

Die Erfahrungen aus den ersten drei Phasen der Entwicklung von NLS bewirkten einen Bedarf an Vergleichbarkeit sprachtechnologischer Systeme, um erfolgreiche Systeme bzw. Techniken zu identifizieren und um den Transfer der Sprachtechnologie aus den Forschungslaboren hin zu kommerziellen Produkten zu beschleunigen. Von nationalen Einrichtungen der USA (u.a. DARPA, CIA, NIST) eingerichtete Programme zur Förderung der Texttechnologie waren die Vorreiter dieser Entwicklung.

So wurden z. B. im Rahmen des TIPSTER-Programms verschiedene Konferenzen durchgeführt, auf denen teilnehmende Systeme, die standardisierte Aufgaben lösen mussten, miteinan-

¹² Reference Architecture for Generation Systems

¹³ s. hierzu auch *Poller* 2002; *Herzog et al.* 2004.

¹⁴ z.T. inspiriert durch den Vortrag „Three (and a Half?) Trends: The Future of Natural Language Processing“ von Eduard Hovy, gehalten am 7.12.2005 auf einem Kolloquium an der University of Colorado at Boulder.

der verglichen wurden (daher auch die Bezeichnung **Shared Task Evaluation Conferences**). Beispiele sind die *Message Understanding Conference* (MUC) zu Informationsextraktionssystemen (s. die Abschnitte 5.2 und 5.5), die *Text Retrieval Conferences* (TREC) zu Texterschließung und Fragebeantwortung (s. S. 110) sowie die *Text Summarization Evaluation Conference* (SUM-MAC) zu Textzusammenfassungssystemen (die z.T. im Rahmen des *Translingual Information Detection Extraction and Summarization* (TIDES) Programms als *Document Understanding Conferences* (DUC, s. S. 205) weitergeführt wurde).

Gegenwärtig werden Aufgaben wie Fragebeantwortung (*question answering*, das Erkennen von impliziter Information (*recognizing textual entailment*), s. auch S. 113), Textzusammenfassung (*summarization*) und das Füllen einer Wissensbasis mit aus Texten gezogener Information (*knowledge base population*) als einzelne Bestandteile (*tracks*) der *Text Analysis Conference* (TAC) des NIST komparativ verfolgt.

Hobbs (2004) diskutiert die Frage, auf welche Weise und anhand welcher Gesichtspunkte eine Evaluation von NLS durchgeführt werden kann/sollte. Er schlägt vor, zwischen Kompetenz und Performanz zu unterscheiden und diese getrennt zu bewerten. Mit der **Kompetenz** eines NLS ist danach die Fähigkeit gemeint, *überhaupt* bestimmte Phänomene im Bereich der maschinellen Sprachverarbeitung behandeln zu können (s. die elaborierte Liste, die Hobbs anführt). Als **Performanz** eines NLS bezeichnet er die quantifizierbare Funktionalität des Systems in der Umgebung, in der es eingesetzt ist (z. B., wie viele der an es gestellten Fragen werden von einem Frage-/Antwortsystem korrekt/zufriedenstellend beantwortet?).¹⁵ Kapitel 9 liefert ein Beispiel für praxisorientierte Performanzevaluation.

Verstärkte Kollaboration von Forschung, Anwendung und Geldgebern

Als Weiterführung komparativer Evaluation zeichnet sich gegenwärtig ein Trend ab, der auf noch mehr Transparenz abzielt. Diese soll dadurch erreicht werden, dass sich Forschung, Anwender und Geldgeber auf ein gemeinsames Vorgehen in einem „kollaborativen Forschungsprozess“ einigen. Hierzu gehört, systematisch interessante Probleme, Architekturen, Bewertungsmechanismen und Entwicklungsprozesse zu bestimmen, so dass Forschungsfortschritte transparenter und schneller kommuniziert werden können. Ein Beispiel ist das Open Advancement of Question Answering Systems (OAQA, s. *Ferrucci* 2009).

Unterstützende maschinelle Sprachverarbeitung

Eine Möglichkeit, mit der inhärenten Komplexität maschineller Sprachverarbeitung umzugehen, besteht darin, grundsätzlich vom Desiderat einer vollständigen Automatisierung abzurücken und stattdessen **Computer-gestützte Sprachverarbeitung** (**computer-aided natural language processing**) als Leitbild in der Sprachtechnologie zu etablieren. Diese vermeidet die Probleme sprachtechnologischer Aufgaben (Ambiguitätsauflösung etc.) durch Einbindung des Menschen in den Prozess und garantiert gleichzeitig den praktischen Mehrwert (erhöhte Effizienz, Kostenersparnis etc.). Beispiele hierfür finden sich in den Kapiteln 12 und 13.

¹⁵ vgl. auch die auf Seite 29 genannten Qualitätskriterien für NLS.

3.6 Literatur

Einen immer noch sehr gut lesbaren Überblick über die Aufgaben und Aspekte der Verarbeitung natürlicher Sprache für natürlichsprachliche Systeme bietet *Wahlster* (1982). *Cunningham* (2005a) befasst sich mit Software-Architekturen für angewandte, d.h. überwiegend auf flacher Verarbeitung basierende, textuelle NLS, *Herzog et al.* (2004) mit solchen für Speech- bzw. multimodale NLS. Eine wunderbare Überblicksdarstellung des gesamten Bereichs der maschinellen Sprachverarbeitung ist *Menzel* (2007).

Kapitel 4

Natürlichsprachliche Suche

I think the problem, to be quite honest with you, is that you've never actually known what the question is.

(Deep Thought in *The Hitchhiker's Guide to the Galaxy*
(*Ich hätte bei der Auswahl dieses Zitats wirklich nicht gedacht, wie aktuell es schon bald sein würde*
(s. WATSON, S. 110).))

Wir sind daran gewöhnt, bei einer Suche im Internet Stichwörter einzugeben (eingeben zu müssen) und daraufhin eine zum Teil ausufernde Liste von Ergebnissen präsentiert zu bekommen, von denen der größte Teil wiederum irrelevant ist. Die sprichwörtliche Google-Suche hat mit cleveren Bewertungsmechanismen dazu beigetragen, dass relevante Ergebnisse immerhin meistens zuerst angezeigt werden.

Internetsuche ist ein Spezialfall des **Information Retrieval**, der Informationserschließung. Hierbei besteht das Problem für Benutzer üblicherweise darin, die Suchanfrage so zu gestalten, dass die gewünschten Ergebnisse möglichst genau getroffen werden (s. hierzu den Exkurs auf der nächsten Seite).

Natürlichsprachliche Suche verspricht hier einen qualitativen Quantensprung dahingehend, dass *mehr relevante und gleichzeitig weniger irrelevante* Treffer einer Suche geliefert werden, was durch Ausnutzung von Aspekten der natürlichen Sprache erreicht wird —in Abgrenzung zu den einfacheren Schlüsselwort-basierten Verfahren des Information Retrieval.

Ein Beispiel: der Unterschied in der Suche nach „books for children“ bzw. „books about children“. Bei üblichen Suchverfahren fließen nur Inhaltswörter in die Suche ein (Funktionswörter wie Konjunktionen und Präpositionen werden als sog. *Stoppwörter* eliminiert), so dass dieser Unterschied verloren geht. Natürlichsprachliche Suche erhält die sprachliche Information und generiert somit eine präzisere Anfrage.

Außerdem soll idealerweise eine qualitative Verbesserung der Suche durch eine *Verfeinerung der Ergebnisse* erreicht werden.

Gegenwärtig wird natürlichsprachliche Suche eher in Form sprachtechnologischer Komponenten für das Information retrieval realisiert und weniger als kompakte NLS.

Information retrieval (IR) und Volltextsuche

Information retrieval (IR) basiert im Wesentlichen darauf, dass für ein beliebiges Textdokument Information derart abgespeichert wird, dass es später anhand einer Suchanfrage („query“) mit ausgewählten Wörtern (Schlüsselwörtern, *keywords*) wieder auffindbar (erschließbar) ist.

Klassischerweise dienten entsprechende Wörter (sog. Index-Terme) dazu, einen Text zu charakterisieren (z.B. in Bibliothekssystemen). D.h., man musste im Prinzip wissen, welche Suchwörter dem jeweiligen System bekannt sind. Heutzutage fungieren mehr oder weniger alle Wörter eines Dokuments als Indexterme, weswegen man bei dieser Art von IR auch von **Volltextsuche** spricht.

Indexterme sind nicht notwendigerweise identisch mit den Wörtern des Textes, sondern sind üblicherweise das Resultat einer **Normalisierung** (Reduzierung auf Grundform), oft werden häufig auftretende Wörter (z.B. Funktionswörter) als sog. „Stoppwörter“ ganz ausgeschlossen.

Die Suche nach Dokumenten wurde früher als Schlüsselwort-basierte Suche nach logischen Kriterien durchgeführt (sog. **Boole'sche Suche**), z.B. suchte man mit 'Bank AND (sitzen OR setzen OR ausruhen)' nach Dokumenten über entsprechende Sitzgelegenheiten, mit 'Bank AND NOT (sitzen OR setzen OR ausruhen)' demnach eher nach Dokumenten über Geldinstitute.

Heute ist die Information zu Dokumenten in der Regel komplexer strukturiert (kein Entweder-oder-Vorkommen im Text, sondern ein gewichteter Wert; außerdem Positionsinformation der Wörter). Die Suchwörter sind meistens AND-verknüpft, allerdings weniger starr als früher, und Positionsinformation wird berücksichtigt (s. die Reihenfolge der angezeigten Treffer in Google). Gleichwohl lässt sich in der Suchanfrage auf dem Vorkommen eines Suchwortes bestehen ('+SUCHWORT') oder auf dessen Nicht-Vorkommen ('-SUCHWORT'), oder auf dem exakten Vorkommen einer bestimmten Wortfolge (durch Einschluss in Hochkommata).

Dies führt dazu, dass nicht-linguistische Phrasensuchen (mehrere aufeinanderfolgende Suchwörter wie 'Unterricht im Wald') oder exakte Treffer möglich werden. Im Gegensatz zu sprachtechnologischer Phrasensuche (s.u.) führt dies allerdings oft noch zu inkorrekten Treffern bzw. unvollständigen Ergebnissen.

4.1 Allgemeines zur natürlichsprachlichen Suche

„Natürlichsprachliche Suche“ ist gegenwärtig nicht klar definiert. Es lassen sich mindestens drei Interpretationen des Terminus unterscheiden:

- eine Verbesserung Schlüsselwort-basierter Suche durch Sprachtechnologie, vor allem durch Bearbeitung der Eingabe

- ein anderer, inhaltsorientierter Ansatz der Suche (sog. **Semantische Suche** (*semantic search*)); Semantische Suche basiert auf extensiver Vorverarbeitung der Dokumente, insbesondere auch mit Methoden der Informationsextraktion (s. hierzu Kap. 5); anstelle Wort-basierter Indexterme werden inhaltsbasierte (*semantische*) gestellt
- eine erweiterte, komplexere Auffassung von Suche als Fragebeantwortung (s. hierzu Kap. 7); hieraus leitet sich die Bezeichnung **Antwortmaschine** aus dieser modernen Sichtweise ab

Verfahren natürlichsprachlicher Suche führen entsprechend zu anderen bzw. verfeinerten **Resultaten**:

- **Textstellen** (*passages*) anstelle von ganzen Dokumenten (→ *passage retrieval*)
- **Fakten** anstelle von Text (→ *fact retrieval*)
- **Antworten** (→ *question answering*)

Konkrete Systeme mit natürlichsprachlicher Suche unterscheiden sich im Wesentlichen dahingehend,

- was die Quellen der Information und Gegenstandsbereich der Suche sind (strukturierte Datenbanken bis unstrukturierte Texte im Web)
- wie „tief“ analysiert wird (morphologisch, syntaktisch, semantisch)
- ob Suchanfrage **und** Quelltexte analysiert werden
- ob die Domänen (der Gegenstandsbereich der Suche) beschränkt sind oder nicht
- wie bewertet wird (z.B. mit oder ohne Berücksichtigung der Verweisstruktur)
- welche Information als Ergebnis präsentiert wird (ganze Dokumente, spezifische Passagen oder genaue Antworten)

4.2 Geschichte

Wie auf Seite 12 erwähnt, werden momentan große Erwartungen in den Bereich der natürlichsprachlichen Suche gesetzt, für die in den nächsten Jahren gezeigt werden muss, ob sie erfüllt werden können. Tatsächlich ist das Desiderat einer flexiblen, „natürlichen“ Suche schon sehr viel älter und fällt in den Bereich der **natürlichsprachlichen (Retrieval-)Schnittstellen** (zu Datenbanken)(s. Kap. 6). Wesentliche Aufgaben/Probleme/Aspekte der natürlichsprachlichen Suche sind dort bereits untersucht worden.¹

Ein klassischer Bereich ist der der Informationssysteme, in dem schon vor dem Internetzeitalter der Bedarf nach einer effektiven intuitiven Suche gerade für Laien bestand, die die

¹ Ein anderer relevanter Bereich ist der des **Textmining**, in dem es eher um „Finden“ als um „Suche“ geht, der aber entsprechende Verfahren der Verarbeitung von Text voraussetzt.

jeweiligen (semi-)formalen Anfragesprachen nicht beherrschten. Urahn dieses System ist wahrscheinlich das System **PROTOSYNTHEX** von R. Simmons (Anfang der 1960er Jahre, s. hierzu *Tenant* 1981 sowie den Exkurs auf der nächsten Seite).

Ein aktuelleres Beispiel für einen Lösungsansatz stellt das System **OSIRIS** dar, das —als Zusammenarbeit der Computerlinguistik-Abteilung und der Bibliothek der Uni Osnabrück— eine natürlichsprachliche Suche nach Buchinformationen ermöglichte (s. *Ronthaler* 2001). Ziel dieses Projekts war es, Benutzern des Bibliotheksinformationssystems jeweils spezifische Treffer für die Suche nach „Wald im Unterricht“ bzw. „Unterricht im Wald“ liefern zu können. Hierfür wurden bewusst nur Phrasen als Ergänzung des Satzanfangs „Ich suche Literatur zum Thema“ erlaubt, um die Komplexität der Analyse einzuschränken.

Das erste, seit 1993 im Internet verfügbare Frage/Antwort-System **START** (*Katz* 1997, s. auch Kap. 7) basierte zwar auf einer Wissensbasis, allerdings wurde dort bereits vorgeschlagen, ausgewählte Wissensquellen im Web so zu annotieren, dass eine natürlichsprachliche Suche mit der Technologie von **START** zu qualitativ hochwertigen Antworten führt. Dies ist eine Idee, die generell dem Semantic Web zugrunde liegt, aber auch eine, die gegenwärtig von **POWERSET** verfolgt wird, indem auf einen beschränkten, sprachtechnologisch bearbeiteten Teil des Webs, Wikipedia, fokussiert wird.

Einen anderen –einfachen aber aufwändigen– Weg schlug Ende der 1990er Jahre die Suchmaschine **ASKJEEVES** (jetzt: ask.com) ein. Hier waren ebenfalls ganze Fragen erlaubt, die allerdings auf bereits im System vorhandene Frage-Antwort-Paare (300 Mio, Stand 2009) in einer Datenbank abgebildet wurden, die zuvor mit großem Menschenaufwand erstellt worden war. Aus passenden ähnlichen/gleichen Fragen konnte der Benutzer auswählen. So konnten häufige, in verschiedener Form gestellte Anfragen erfasst werden.

Etwa zur gleichen Zeit entstand die (mittlerweile verschwundene) Suchmaschine **ELECTRIC MONK**. Sie ist insoweit ein klassisches Beispiel für natürlichsprachliche Suche, als sie natürliche Fragen als Suchanfragen erlaubte, diese aber mit cleveren Algorithmen jeweils auf eine einfache Schlüsselwortanfrage an **ALTAVISTA** abbildete. Dies bedeutete nicht nur eine natürlichere Schnittstelle für Benutzer, sondern auch spezifischere Anfragen und somit eine verbesserte Trefferquote.²

Mittlerweile sind die Methoden im Bereich robuster Syntaxverarbeitung und flacher ontologischer Semantik soweit entwickelt, dass Firmen wie **POWERSET** und **HAKIA** Erfolgsschancen eingeräumt werden. Beispielsweise wurde **POWERSET** im August 2008 von **MICROSOFT** aufgekauft. Beide Firmen starten allerdings mit eingeschränkten Domänen: **POWERSET** fokussiert auf **WIKIPEDIA**, **HAKIA** verwendet vorwiegend als vertrauenswürdig gekennzeichnete Webseiten, **COGNITION** bietet natürlichsprachliche Suche primär für Firmen-eigene Textmengen an. Die prototypische *Antwortmaschine* ist momentan **WOLFRAM|ALPHA** (aber s. auch **WATSON**, S. 110), den Übergang von einer Such- zur Antwortmaschine repräsentiert gegenwärtig am besten **POWERSET/BING**, die sowohl traditionelle Suchergebnisse wie auch –wenn möglich–

² Beispiele hierfür werden in einem [Artikel von WIRED](#) beschrieben. Hier findet sich auch der mutmaßliche Grund für das Ende der Suchmaschine: es gab keine Absprache mit den Besitzern von **ALTAVISTA**.

konzise Antworten liefern.

Natürlichsprachliche Suche wird zudem im Bereich Business Intelligence eingesetzt. So arbeitet **TERAGRAM** mit **EASYASK.COM** zusammen und ist von **SAS**, einem Marktführer in diesem Bereich, aufgekauft worden. Beispiele für (weitere) aktuelle Firmen im Bereich der natürlichsprachlichen Suche finden sich auf den folgenden Webseiten:

- <http://www.hakia.com>
- <http://www.bing.com>
- <http://www.wolframalpha.com>
- <http://www.hapax.com>
- <http://www.intellexer.com>
- <http://www.netbase.com>
- <http://www.cognition.com>

Protosynthex (Simmons ~1960-65, nach Tenant 1981)

PROTOSYNTHEX (von: Prototyp des Projekts „*Synthesis of complex verbal behavior*“) beantwortete Fragen durch Suche in (Ausschnitten) einer Enzyklopädie (der *Golden Book Encyclopedia*).

PROTOSYNTHEX involvierte das Erstellen eines invertierten Indexes der Enzyklopädie (unter Auslassung von Stoppwörtern) mit der Annotation, in welchem Bereich/Artikel/Abschnitt/Satz ein Wort vorkam, sowie eine rudimentäre Wortnormierung und eine einfache syntaktische Analyse der Frage.

Die Inhaltswörter der Frage wurden im Index nachgeschlagen und der jeweils kleinste textuelle Bereich ihres Vorkommens identifiziert. Bei mehreren Treffern wurde ein Ranking abhängig von der Häufigkeit der Wörter vorgenommen, und schließlich wurden durch syntaktische Analyse der besten Treffer potentielle Fehlantworten ausgefiltert (z. B. „Boys fly kites“ als Antwort auf die Frage „How do kites fly?“). Das Programm war interaktiv konzipiert, um Ambiguitäten aufzulösen.

Entsprechend konnten auf die Frage „What do worms eat?“

„Worms eat grass“ und „Grass is eaten by worms“ als Treffer identifiziert,

„Birds eat worms“ ausgeschlossen und

„Worms eat their way through the ground“ bzw. „Horses with worms eat grain“ zumindest als potentielle Treffer angeboten werden.

Die Oberflächen-/Wortorientiertheit des Ansatzes hat dazu geführt, dass der Nachfolger **PROTOSYNTHEX II** semantisch orientiert war und als einer der Vorläufer heutiger semantischer Netzwerke gelten kann.

4.3 Aspekte und Anforderungen

Natürlichsprachliche Suche involviert mindestens die folgenden Aspekte, in aufsteigender Komplexität:

Fehlertoleranz

Ein klassisches Problem der Suche sind fehlerhafte Eingaben, die nicht ausschließlich zum Misserfolg führen, sondern durch das System abgefangen werden sollten (wie heutzutage bei Googles „Meinten Sie: ...“ der Fall). Die bekanntesten Verfahren hierfür basieren auf der sogenannten **Levenshtein-Distanz**, die ein Maß für den orthografischen Unterschied zweier Wörter darstellt. Beschränkt man diese Distanz, lassen sich leicht Abweichungen von einem Wort („Fehler“ wie Auslassungen, Einfügungen, Vertauschungen) erkennen.

Schreibvarianten

Schon in Anfragen an Datenbanken traf man frühzeitig auf das Problem, dass Suchwörter manchmal auf verschiedene Weise geschrieben werden können, aber nur in einer Variante im System bekannt sind. Eine Lösung hierfür ist die sog. **phonetische Suche**, in der gleichklingende Wörter (z.B. „Delfin“ und „Delphin“, „Mayer“ und „Meier“) auf einen einzigen Indexterminus abgebildet werden. Das bekannteste Verfahren hierfür ist der **Soundex**-Algorithmus. In seiner klassischen Version generiert es die Zeichenfolge D415 für „Delphin/Delfin“. Gerade um auch die deutsche Orthografie verbunden mit Ausdrücken und Schreibungen anderer Sprachen angemessen zu behandeln, sind allerdings komplexere angepasste Versionen dieses Verfahrens notwendig.

Wortvarianten: linguistische Ähnlichkeit

Wortvarianten sind linguistisch motivierte Versionen eines Wortes, deren Ähnlichkeit einem System erst einmal unbekannt ist (konjugierte – „schreiben“ vs. „schreibst“ – oder deklinierte – „Frau“ vs. „Frauen“ – Formen). Einfache Verfahren hierzu reduzieren („normalisieren“) die Indextermini und Suchwörter auf entsprechende invariante Formen wie „Frau“ bzw. „schreib“ (sog. **Stemming**), komplexere Verfahren der Sprachtechnologie verwenden Methoden der (**flextions)morphologischen Analyse** mit Hilfe endlicher Automaten. Diese wird beispielsweise in Fällen der Umlautung („Haus“ vs. „Häuser“) oder Ablautung („treffen“, „trifft“, „traf“, „getroffen“) notwendig.

Im Übrigen kann es sein, dass dieselben Wörter verschiedene Dinge bezeichnen (**Homonyme** bzw. **Polyseme**), z.B. *single* (nicht verheiratet vs. einzelnes Lied/Schallplatte). Im Idealfall wird die Mehrdeutigkeit durch den vorhandenen Kontext der Anfrage disambiguiert.

Komplexe Wörter

Einige Wörter sind inhärent komplex (z.B. die sog. *Nominalkomposita*), was bei Suchanfragen berücksichtigt werden sollte. Beispielsweise möchte man bei einer Suche nach *pommes de terre* keine Seiten über Äpfel und die Welt erhalten, umgekehrt ist man bei einer Anfrage *Computerlinguistikstudium* froh über Treffer, die vom *Studium der Computerlinguistik* handeln. Verfahren der **lexikalischen Analyse** bestimmen, welche Wörter ein Kompositum bilden und in

welche Wörter sich ein solches wiederum aufspalten lässt. Hierfür werden z.B. Kombinationen lexikalischer und morphologischer Automaten verwendet.

Konzeptvarianten: lexikalische/semantische/konzeptuelle Ähnlichkeit

Eine verbesserte Suche lässt sich erzielen, wenn nicht nur nach **denselben** Worten in verschiedener Form, sondern auch nach inhaltlich verwandten Wörtern gesucht wird („lexikalische Expansion der query“). Relevant hierfür sind insbesondere bedeutungsgleiche (*synonyme*) (EU, Europäische Union) Termini oder solche, die einem anderen bezogen auf eine Taxonomie direkt untergeordnet (*hyponym*) oder übergeordnet (*hypernym*) sind.³

Werden gleichzeitig mehrere zusammenhängende Wörter (eine Phrase) auf diese Weise bearbeitet, so ergibt sich ein erster Ansatz zur Behandlung des *Paraphrasenproblems* im IR, d. h., zu einer Anfrage auch Textstellen mit ähnlicher Formulierung zu finden. Beispielsweise lassen sich für eine Anfrage „car washing“ bei Verwendung der Hypernyme auch Texte über „automobile cleaning“ finden.

Einschlägige Ressourcen für die Verfahren zur Ermittlung von Konzeptvarianten sind Thesauri und vor allem lexikalisch-konzeptuelle Netzwerke wie **Wordnet**⁴.

Die Paraphrasengenerierung ist nicht trivial, da sie notwendigerweise auch die Auflösung von Ambiguitäten involviert. Beispielsweise ist es nicht sinnvoll, für *draw money* als Alternative *paint money* zu generieren, im Gegensatz zur Variante *draw cash*. Diese Varianten können anwendungs- bzw. domänen spezifisch sein und werden daher sinnvollerweise in verschiedenen Ressourcen (allgemeinen, domänen spezifischen, kundenspezifischen Lexika/Thesauri etc.) gespeichert.

Wortkombinationen: Phrasensuche

Dinge, nach denen man sucht, werden oft nicht nur durch ein Wort beschrieben, sondern durch Wortkombinationen. Standardsuchverfahren sind hierfür in der Regel entweder zu lax oder zu rigide. So kann eine Anfrage *Kirche und Recht* auch zu einem Blogeintrag mit dem Text „du hast Recht, Kirche und Staat sind...“ als Treffer führen, während eine exakte Suche 'Kirche und Recht' einen Text zu „Kirche und römisches Recht“ ggf. nicht liefert.

Eine qualitativ hochwertige **Phrasensuche** im Sinne der Sprachtechnologie impliziert – vereinfacht – eine Analyse von *Kirche und Recht* als **Nominalphrase** und favorisiert entsprechend Treffer vom selben Typ. Hierzu gehört, dass Modifikationen von Nomina durch Adjektive wie *römisches* zugelassen sind, da sich der Phrasentyp nicht ändert.

In dieselbe Kategorie gehören Beispiele wie *Unterricht im Wald/Wald im Unterricht*: Hierbei handelt es sich um – jeweils unterschiedliche – Kombinationen von Nomina mit Präposition und Nominalphrase (einer **Präpositionalphrase**), deren Struktur bei der natürlichsprachlichen

³ Beispiel: 'Hecht' ist ein Hyponym zu 'Süßwasserfisch' und 'Süßwasserfisch' ist ein Hypernym von 'Hecht'.

⁴ Wordnet ist eine computerlinguistische Ressource, in der verschiedene lexikalisch-semantische Bezüge (Bedeutungsverwandtschaften) zwischen Wörtern nachgeschlagen werden können, s. wordnet.princeton.edu. Wordnet ist mittlerweile eine der Standard-Komponenten (bei der Entwicklung) von NLS.

Suche ausgenutzt wird, so dass jeweils nur Treffer zum gesuchten Thema geliefert werden.

Satzfunktionen

Ist eine Analyse von Nominalphrasen gegeben, ist noch nicht gewährleistet, dass die Rolle/Funktion einer solchen Phrase im Satz bei der Suche berücksichtigt wird. Beispielsweise sollte für eine Anfrage *Wen besiegte Kramnik* nicht „Anand besiegte Kramnik“ geliefert werden, da nach Kramnik als **Subjekt** (und nicht: Objekt) des Besiegens gefragt wird. Dies erfordert eine entsprechende **syntaktische Analyse** von Eingabe und Treffer.

Syntaktische Analyse führt ebenfalls zur Disambiguierung mehrdeutiger Wörter, z. B. zur Unterscheidung der Bedeutung von „check“ in *Can I check how much cash I have in the bank?* (Verb im Prädikat) und *Will the bank cash my check?* (Substantiv im Objekt).

Satz-/Frageform

Das ideale Ergebnis einer Suche ist die Antwort auf die zugrunde liegende Frage. Entsprechend lässt sich an der **Frageform** bereits manchmal nicht nur ablesen, wie die Antwortform sein sollte, sondern auch möglicherweise, wo gesucht werden sollte. Ein Beispiel sind „Wie kann ich/man ...“-Fragen, für die sich eine Suche auf FAQ- oder Manual-Sites anbietet. Eine solche eingeschränkte Websuche bietet <http://www.HowDoYa.com> an.

Zu elaborierten Techniken der Behandlung von Suchanfragen als Fragen s. Kap. 7.

Rückfragen

Im Falle von mehrdeutigen Anfragen kann natürlichsprachliche Suche die Generierung möglicher Alternativen für Suchanfragen beinhalten, aus denen der Benutzer die gewünschte auswählt. Im Idealfall werden z. B. für die Anfrage „Was muss ich beim Schimmel beachten?“ die Alternativen ‚Pflege von Schimmeln (Pferde)‘ und ‚Vermeiden/Beseitigen von Schimmel (Pilz)‘ angeboten.

4.4 Literatur

Empfehlenswerte Einführungen in das Information Retrieval sind *Baeza-Yates und Ribeiro-Neto* (1999) und *Manning et al.* (2008). Grundlegende Aspekte der Verarbeitung von Text werden in *Heyer et al.* (2006) und *Jurafsky und Martin* (2008) kompakt präsentiert.

Kapitel 5

Informationsextraktion (IE)

You contain information. I need to know how to get at it.

(John Anderton in: *Minority Report (Film)*)

Information retrieval (Informationserschließung, IR) befasst sich generell mit dem Zugang zu bzw. dem Zugriff auf Text(teil)e einer großen Dokumentmenge. Ergebnisse einer Anfrage (einer Menge von Schlüsselwörtern) sind die relevanten Text(teil)e, die dem Benutzer für die intendierten Zwecke (Sichten, Sammeln, Weiterverarbeiten) dargeboten werden (s. hierzu den Exkurs auf Seite 50).

Informationsextraktion (IE) dagegen ist eine Bezeichnung für spezifischere Techniken des „**Herausziehens“ gewünschter Information aus einzelnen Texten** zum Zweck des Aufbaus strukturierter Informationsressourcen wie z. B. eine Datenbank mit Name, Adresse, Forschungsgegenstand und Produkt von Firmen im Bereich Biotechnologie aus elektronischen Newslettern.

Während die Informationsressourcen früherer IE-Ansätze aus Korpora, Manuals o.Ä. bestanden, gilt das Hauptinteresse heute der im Web verfügbaren Information, also auch solcher in **beliebigen, unstrukturierten** Texten.

Anwendungen sind z. B.

- Ein verbessertes Information Retrieval, in dem für grobe Suchergebnisse angezeigt wird, *wer in welchen Dokumenten welche Rolle* spielt.
- Eine verbesserte *Business Intelligence*, so dass die für eine Firma relevante Information automatisch nach interessanten Zusammenhängen gesichtet wird (was sagen bestimmte Quellen darüber aus, *welche Firmen Verlust gemacht, den Besitzer gewechselt, ein neues Patent angemeldet haben usw.*)
- Eine intelligente Aufbereitung unstrukturierter Daten (z. B. Email-Sammlungen) im Hinblick auf *die in ihnen enthaltenen, für die jeweilige Firma relevanten, Fakten*.

5.1 Allgemeines zur IE

Bei der Informationsextraktion handelt es sich um ein Verfahren,

- das automatisch in unstrukturiertem Text relevante Information erkennt, extrahiert, und zu strukturierten Daten zusammenfügt
- das (in der traditionellen Form) von Experten vordefinierte Schablonen (*templates*) bzw. Muster (*patterns*) verwendet
- das wesentlich auf computerlinguistischen (flachen) Verarbeitungsmethoden basiert (erheblich mehr als IR)
- das nicht auf die Bedeutung der Texte schaut, um die gewünschte Information zu finden
- dessen Input einerseits Templates sind, die die relevante Information charakterisieren, und andererseits reale Texte, in denen sich die gesuchte Information möglicherweise befindet (also z. B. mit IR ausgesuchte Dokumente)
- dessen Output die mit relevanter Information instantiierten Templates sind (d.h. gefüllt mit extrahierten Textstrings)

Die Informationsextraktion aus unstrukturierten Web-Texten (s. hierzu *Popescu* 2007) hat einige Vorteile:

- **Redundanz:**
dieselbe Information kommt häufig mehrfach vor (was zur Einschätzung der Informations„güte“ ausgenutzt werden kann)
- **Paraphrasen:**
dieselbe Information tritt häufig in unterschiedlicher sprachlicher Realisierung auf
- **einfachere Sprache:**
manche Realisierungen sind einfacher als andere und daher möglicherweise mit einfacheren Methoden extrahierbar (im Gegensatz zu Ressourcen mit eher komplexer Sprache, z. B. Zeitungsartikel)
- **breite Abdeckung:**
d.h., die für NLS geforderte konzeptuelle Abdeckung wird eher gewährleistet als bei spezifischen Ressourcen
- **Suchmaschinen:**
Sie sind verfügbar und gewährleisten den Zugriff auf die gesuchte Information. Außerdem liefern sie wertvolle Information über die Ko-Okkurrenz sprachlicher Ausdrücke.

Allgemein wird hierdurch die Gefahr verringert, gesuchte Information zu verpassen. Die Nachteile von Web-IE liegen in der erhöhten Gefahr von

- **inkorrekt Information** (auch bei mehrfachem Auftreten) und
- **ungrammatischer Spache** (was den Zugriff erschwert).

Eine wesentliche Anwendung neben dem der eigentlichen Zusammenstellung gewünschter Information (z. B. der Aufbau von Ontologien) besteht in der Verbesserung anderer Informations-technologie-Techniken (v.a. der Verfeinerung der Indexterme im IR und der Dimensionen in der Textklassifikation) oder als ergänzende Komponente (z. B. zum Filtern der IR-Suchresultate oder Hervorheben relevanter Information in automatischen Zusammenfassungen).

Die zentralen Aufgaben der IE bestehen in der Erkennung von **Entitäten** bestimmten Typs (*Named Entities* wie Firmennamen), von **Relationen** zwischen Objekten (z. B. wer ist der CEO welcher Firma?) und **Ereignissen** (z. B. wann wurde welche Firma von welcher anderen übernommen?). Hierbei ist zu gewährleisten, dass alle relevanten sprachlichen Variationen dieser Aspekte erfasst werden.¹ Damit ergeben sich für ein IES drei grundlegende Anforderungen:

- es muss bestimmt werden, was die wichtigen zu erfassenden Fakten (in einer bestimmten Domäne) sind
- für jedes Faktum muss bestimmt werden (d.h., es müssen **Muster** gebildet werden), auf welche verschiedenen Weisen es sprachlich ausgedrückt werden kann
- es müssen die *Instanzen* der Muster in den Texten gefunden werden

Abbildung 5.1 auf der nächsten Seite zeigt ein etwas komplexeres Beispiel für die IE (aus: *Hobbs et al. 1997*): die Extraktion von Information zu terroristischen Handlungen aus Nachrichten mithilfe eines sogenannten *scenario templates*. Die Einträge (Zeilen) des Templates geben an, nach welcher Art von Information gesucht wird (insgesamt: die Information über das zu extrahierende Ereignis). Sie werden mit den gefundenen Informationen in den Texten instantiiert.

5.2 Geschichte

Bereits in den 1970er Jahren arbeitete Naomi Sager an der Umwandlung von Arzt-Berichten in Datenbank-verwendbare Form (s. *Hirschman und Sager 1982* und den Exkurs auf Seite 62). Um 1980 existierten bereits eine Reihe von Ansätzen und Systemen zur IE, u.a. solche zur

- Ereigniserkennung aus Newswire-Nachrichten mit Scripts (**FRUMP**, s. *DeJong 1982*)
- Extraktion von Satellitenflug-Informationen aus Berichten (*DaSilva und Dwiggins 1980*)
- Extraktion von Treffen und Beziehungen der Personen in Texten über historische franz. Persönlichkeiten (*Zarri 1983*)
- Extraktion von Information über Pflanzen aus Blumenführer (*Cowie 1983*)

Ab 1987 fanden die *Message Understanding Conferences (MUC)* statt, die IE als zentralen Gegenstand hatten und ihn als eigenen Bereich in der Sprachtechnologie etabliert haben.

¹ Hier kommen z. B. Anaphern- und Koreferenzauflösung ins Spiel.

San Salvador, 19 Apr 89 (ACAN-EFE)

Salvadoran President-elect Alfredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime.

...

Garcia Alvarado, 56, was killed when a bomb placed by urban guerrillas on his vehicle exploded as it came to a halt at an intersection in downtown San Salvador.

...

Vice President-elect Francisco Merino said that when the attorney general's car stopped at a light on a street in downtown San Salvador, an individual placed a bomb on the roof of the armored vehicle.

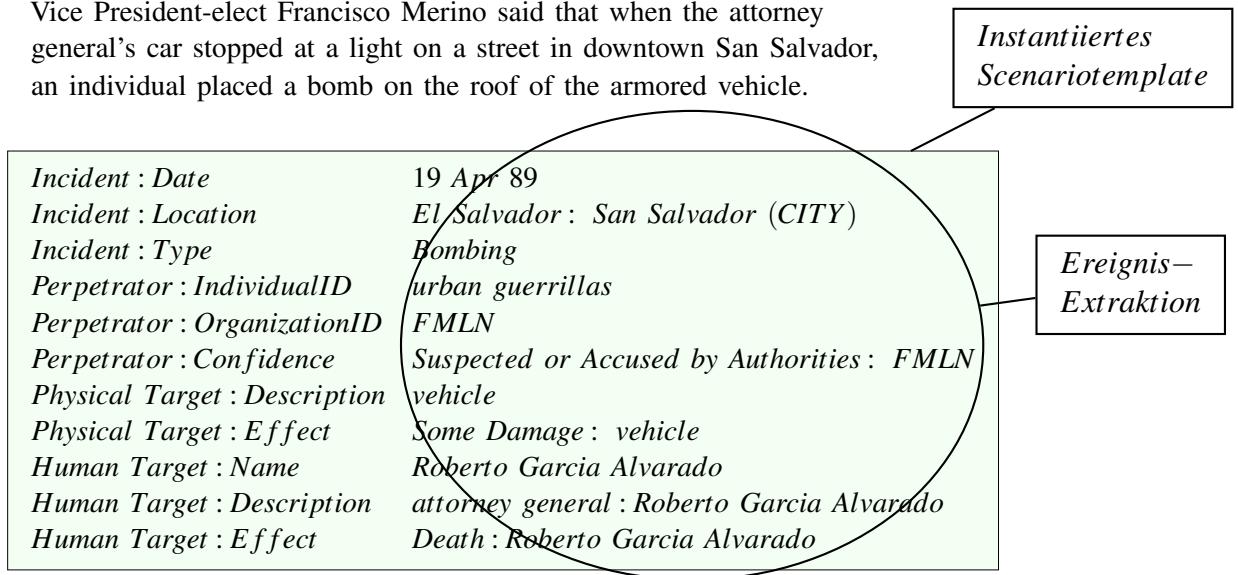


Abbildung 5.1: Beispiel für IE

Diese wie auch weitere Evaluationskonferenzen (TREC, DUC) und Forschungsprogramme (z. B. TIPSTER) verdankten ihre Existenz und Förderung der amerikanischen (militärischen) Einrichtungen, insbesondere der DARPA (*Defense Advanced Research Projects Agency*). Das Ziel war es, die Technologieentwicklung in diesem Bereich der Sprachtechnologie systematisch zu fördern. Die entscheidende Idee hierfür war die Verwendung eines gemeinsamen Testsets und die damit mögliche vergleichende Evaluation der Teilnehmer.

Von 1991-1999 wurden allein im TIPSTER-Programm, das das Zusammenwirken von IR und IE untersuchen sollte, mehr als 15 Projekte unterstützt. Seit 1999 werden diese Unternehmungen im ACE (*Automatic Content Extraction*) Programm des amerikanischen NIST (*National Institute of Science and Technology*) weitergeführt. Schwerpunkte hierbei sind die Erfassung von Bedeutung („semantisches Tagging“), die Untersuchung von IE anhand weiterer Sprachen (Arabisch, Chinesisch) und in weiteren Modalitäten (gesprochene Sprache, Bilder). Eine anwendungsorientierte Entwicklung von IE-Systemen findet im Rahmen des DARPA-Programms TIDES (*Translingual Information Detection, Extraction and Summarization*) statt. Zum besseren Verständnis der Entwicklung der IE hier ein **Überblick über die Veränderungen in den MUC:**

- **MUC-1** (1987), **MUC-2** (1989): IE aus Marine-Nachrichten
- **MUC-3** (1991), **MUC-4** (1992): Zeitungs- und Newswire-Texte über terroristische Vorfälle in Lateinamerika; größeres Textkorpus; komplexere Templates
- **MUC-5** (1993): Zum ersten Mal Teil des TIPSTER-Programms; Texte über Joint ventures (**JV**) und Halbleiterfabrikation/micrelectronics (**ME**); Einführung hierarchischer (d.h. geschachtelter) Templates
- **MUC-6** (1995): Nachrichten über Management-Wechsel; neue Teilaufgaben (z. B. Named Entity (**NE**) Recognition, das Erkennen von Eigennamen); Fokus auf Aufgabenunabhängigkeit und Portabilität eines IE-Systems. Für MUC-6 war eine Entwicklung hin zu tieferem „Verständnis“ bei der IE geplant, nämlich Koreferenz (auf welche NEs beziehen sich Anaphern im Text?), lexikalische Disambiguierung (Auswahl genau einer der möglichen Lesarten eines Wortes), Prädikat/Argument-Strukturen (Identifikation von Satzprädikat, Subjekt, Objekt etc.). Die letzten beiden Aufgaben stellten sich als zu komplex heraus und wurden gestrichen.
- **MUC-7** (1998): Satellitenstart-Berichte

Heute existieren Firmen wie zum Beispiel **TEMIS** und **ATTENSITY**, die im Schnittbereich von Information retrieval, Text mining und Business Intelligence durch die Extraktion relevanter Information aus eingehenden Texten einen Mehrwert für Firmen schaffen.

2009 wurde im Rahmen der **Text Analysis Conference (TAC)** ein Knowledge Base Population (**KBP**)-Track eingeführt, in dem Informationsextraktion Voraussetzung ist für das Anreichern einer Ontologie mit spezifischer (assertionaler) Information (d. h., Faktenwissen) zu bestimmten Entitäten.

Seit der MUC-2 werden die Evaluationsmaße **precision** (*Genauigkeit*) und **recall** (*Vollständigkeit*, auch: *Ausbeute*) sowie das aus beiden zusammengesetzte **F-measure** (*F-Maß*) verwendet, s. Abbildung 5.2.²

<u>precision (P):</u>	<u>recall (R):</u>	<u>F-measure (F):</u>
$\frac{\# \text{korrekte Antworten}}{\# \text{gegebene Antworten}}$	$\frac{\# \text{korrekte Antworten}}{\# \text{mögliche korrekte Antworten}}$	$\frac{(\beta^2 + 1)PR}{\beta^2 P + R}$

Abbildung 5.2: **Evaluationsmaße**

² '#' heißt „Anzahl“ und β ist ein Gewichtungsfaktor für die Präzision gegenüber dem Recall (im einfachsten Fall ist $\beta = 1$).

Das Linguistic String Projekt (LSP)

Das Linguistic String Projekt von Naomi Sager begann 1965 mit syntaktischen Analysen und entwickelte innerhalb der nächsten zwei Jahrzehnte eines der ersten komplexen IES. Wesentlich für den Ansatz im LSP-Projekt (insbesondere für die Performanz des Systems) war die Beschränkung auf eine sogenannte „*sublanguage*“, d.h., eine Fachsprache, die in einer bestimmten Domäne (hier: der klinischen Domäne) verwendet wird und die die Entdeckung und Nutzung spezifischer syntaktischer Muster mit eingeschränktem lexikalisch-semantischen Inventar ermöglichte.

Die zentrale Anwendung des LSP-Systems war die Extraktion von Information aus Patientenberichten, um sie geordnet in einer Datenbank ablegen zu können (*Sager 1978; Hirschman und Sager 1982; Sager et al. 1994*). Hierzu wurden Klassen von Bezeichnern für Objekte und Ereignisse und deren Kookkurrenz-Muster identifiziert. Beispiele hierfür sind

- PATIENT + V-PT ^a+ SIGN-SYMPTOM
Patient has had a cold for 5 days.
She has anorexia.
- LAB + LABRESULT
Blood cultures were positive for Pneumococcus.
Repeat cultures confirmed a pathogen.

Mithilfe solcher Muster konnten aus den Berichten relevante Informationen extrahiert und in ein komplexes Informationsformat über den Status und die Behandlung eines Patienten überführt werden.

Die Arbeiten im LSP-Projekt erforderten einen erheblichen Aufwand an computerlinguistischem Engineering und sind daher nicht repräsentativ für heutige IES, trotzdem sind die Ideen von Sager über ihre Mitarbeiter (u.a. R. Grishman und J. Hobbs) und deren Systeme (*Grishman et al. 1991; Hobbs et al. 1992*) in die aktuellen Entwicklungen eingeflossen.

^a V-PT=Aussage über einen Patienten

5.3 Aspekte und Anforderungen

Generell lässt sich bei der Entwicklung von IE-Systemen (IES) zwischen intellektuellen (Knowledge Engineering) Ansätzen und auf machine learning Verfahren (automatischem Training) basierenden Ansätzen unterscheiden. Intellektuelle Ansätze zeichnen sich dadurch aus, dass

- die Grammatiken von Hand konstruiert werden
- Domänenmuster vom menschlichen Experten durch Introspektion oder durch Korpusinspektion entdeckt werden

- mühevolle(s) Tuning und Optimierung notwendig ist

Klassische Machine learning Ansätze gewinnen Regeln aus annotierten Korpora, für deren Erstellung ebenfalls ein großer manueller Aufwand zu betreiben ist.

Der **Vorteil des Knowledge Engineering** ist der, dass sich mit guter Ausbildung und Erfahrung performante Systeme entwickeln lassen. In diesem Bereich sind die besten Systeme (noch) „handgemacht“. Nachteile sind der grundsätzlich aufwändige Entwicklungsprozess, der aufwändige Anpassungsprozess an neue Domänen sowie die Tatsache, dass die benötigten Ressourcen (Geld, Lexika, Experten). Die **Vorteile trainierbarer Systeme** sind dagegen:

- die einfache Portabilität: ein einmal entwickeltes allgemeines automatisches Verfahren kann leicht auf verschiedene Datensets appliziert werden
- Systemexpertise ist nicht notwendig für Anpassungen
- datengetriebene Regel-Akquisition sichert die volle Abdeckung der Beispiele

Allerdings fehlen möglicherweise (gute) Trainingsdaten und der Aufwand zur Beschaffung kann sehr groß und teuer sein (ggf. werden sehr umfangreiche Trainingsdaten benötigt). Außerdem sind trainierbare Systeme weniger flexibel: Änderungen in den Spezifikationen können eine (teure) Re-Annotierung der Trainingsdaten erforderlich machen. Hieraus lässt sich ableiten, wann welches Verfahren angewandt werden sollte. Die jeweiligen Optionen lassen sich wie in Tabelle 5.1 angegeben gegenüberstellen.

Regel-Ansatz	Training-Ansatz
Ressourcen (Geld) sind verfügbar Regelschreiber verfügbar Trainingsdaten knapp oder teuer Extraktionsspezifikationen ändern sich möglicherweise Höchstmögliche Performanz ist gefordert	Keine Ressourcen Keine erfahrenen Regelschreiber verfügbar Genug (günstige) Trainingsdaten Extraktionsspezifikationen stabil Gute Performanz ist angemessen/ ausreichend

Tabelle 5.1: **Anwendbarkeit der IE-Verfahren**

Die Abhängigkeit von der Existenz und der Qualität manuell annotierter Korpora zusammen mit dem aufkommenden Interesse an **open domain information extraction** (d.h. IE aus großen heterogenen/domänenunspezifischen Informationsquellen (= Web), s. Surdeanu und Harabagiu 2002; Banko *et al.* 2007) hat dazu geführt, dass anstelle der überwiegenden Verwendung **überwachter Verfahren** (IE aus annotierten Korpora oder aus der Interaktion mit Benutzern) in letzter Zeit zunehmend **unüberwachte** (oder schwach überwachte/ semi-überwachte) **Verfahren** verwendet werden, die auf der Basis weniger Beispiele (sog. *seeds*, s. z.B Xu *et al.* 2007) eigenständig Extraktionsmuster und Instanzen solcher Muster sammeln.

5.4 Architekturen und Systeme

5.4.1 FASTUS

Hobbs (*Hobbs 1993*) nennt die folgenden Aspekte/Funktionen eines generischen IES (destilliert aus den verschiedensten Forschungsprototypen) an:

- **Textaufteilung:** Segmentierung des Eingabetextes in Abschnitte.
- **Vorverarbeitung:**
Hier werden Satzgrenzen erkannt, einzelne Sätze werden tokenisiert. Ergebnis der **Tokenisierung** ist eine Folge lexikalischer Einheiten, d.h. unter anderem auch die Wortart (part-of-speech, POS). Die Vorverarbeitung umfasst daher auch das sogenannte **POS-Tagging**, die Zuweisung der Wortart zu einem Token. Hier geschieht auch die Erkennung von Mehr-Wort-Lexemen.
- **Filter:**
Da nicht alle Sätze für die IE relevant sind (und ihre Verarbeitung Aufwand und Zeit erfordern würde), werden irrelevante Sätze ausgefiltert. Im einfachsten Fall wird nach allen für die IE einschlägigen Strings gescannt; Sätze ohne Treffer brauchen nicht weiter berücksichtigt werden.
- **Präparser:**
Um das Parsing i.e.S. (die syntaktische Analyse eines Satzes) zu vereinfachen, werden hier schon kleine „chunks“ sprachlicher Strukturen erkannt.
- **Parser:**
(flache)(partielle) syntaktische Analyse. Das Ergebnis sind Syntaxbäume oder Fragmente davon.
- **Fragmentkombination:**
Zusammenfügen der Fragmente, wenn ein vollständiger Parse nicht möglich ist
- **Semantischer Interpreter:** Zuweisen einer semantischen Struktur/logischen Form
- **Lexikalische Disambiguierung**
- **Koreferenz-Resolution/Diskursverarbeitung**
- **Template Generator** Die Generierung der Templates in der gewünschten Form

FASTUS (*Finite State Automaton Text Understanding System*) (s. *Hobbs et al. 1997*), eine Instanz eines solchen generischen Systems, zeichnete sich nicht nur durch gute Performanzwerte in MUC-4 aus, sondern beeindruckte (seine Autoren) insbesondere durch den relativen Performanzgewinn (Faktor 180!) gegenüber seinem Vorgängermodell **TACITUS**, das noch auf tiefer Analyse basierte. Dabei war die Entwicklungszeit kurz (ca. 6 Monate), und ein Entwicklungssprung von 13.5% auf 49.7% F-score (s. S. 61) konnte in nur 3 1/2 Wochen erreicht werden. Möglich wurde dies durch die Realisierung von IE-Verarbeitungsstufen als endliche Automaten und durch deren Hintereinanderschaltung in Kaskaden. Nacheinander bearbeitete Ebenen waren die der

- **Komplexen Wörter:** Mehrworterkennung, Eigennamen (*set up; joint venture; Peter Pan*)
- **Einfachen Phrasen:** NPs, VPs, Partikel (*more than 30 people; the newly elected president*)
- **Komplexen Phrasen:** Komplexe NPs und VPs (*The joint venture, Bridgestone Sports Taiwan Co.,*)
- **Domäneneignisse:** Anhand der erkannten Phrasen einfache Ereignisse erkennen
- **Vereinigung von Strukturen:** Strukturen aus verschiedenen Textteilen (satzübergreifend!) zusammenfügen, wenn sie eine Entität betreffen

5.4.2 GATE

GATE (*General Architecture for Text Engineering*, Univ. of Sheffield, s. Cunningham et al. 2002; Bontcheva et al. 2004) ist eine IE-Softwareumgebung mit einer Funktionalität auf verschiedenen Ebenen der Informationstechnologie:

- Sie ist eine Software-Architektur für Sprachtechnologie, d.h., es umfasst die für die Aufgabe wesentlichen Komponenten
- Sie ist ein Framework (class library), das die Architektur implementiert und die Technologie anderen Anwendungen zugänglich macht
- Sie ist eine Entwicklungsumgebung mit grafischer Oberfläche, um Komponenten zu entwickeln

GATE ist gekennzeichnet durch eine modulare Handhabung von Wissens-, Verarbeitungs- und GUI-Komponenten. Das erste Release erschien im Jahre 1996, seither wurde GATE von Hunderten Organisationen lizenziert. Sie ist implementiert in Java und frei verfügbar als Open-Source auf der [GATE](#)-Homepage.³

Abbildung 5.3 auf der nächsten Seite (nach: Cunningham et al. 2005) zeigt ein komplexes IES, das sich aus Komponenten von zwei verschiedenen auf GATE basierenden IES zusammensetzt, nämlich das ältere **LASIE** (*LargeScale IE system*) sowie das jüngere **ANNIE** (*A Nearly New IE system*).

Die Komponenten dieses Systems beinhalten überwiegend flache Verfahren der Sprachverarbeitung in Form endlicher Automaten/Transducer. Unter einem **Gazetteer** versteht man eine Menge von Listen, mithilfe derer z. B. Firmenbezeichner („Ltd.“, „AG“ etc.), Personentitel („Dr.“, „Prof.“, „Ing.“ etc.) oder Ähnliches erkannt und markiert werden. Der **semantische Tagger** markiert (bereits annotierte) Token als zu einer komplexeren Kategorie gehörig (Namenerkennung, Datumserkennung o.ä., → *named entity recognition*). Hierfür (wie auch für die Satzzerlegung) wird JAPE (Java Annotations Pattern Engine) verwendet, mithilfe derer komplexe reguläre Muster angegeben und für die Annotation verwendet werden können.

³ vergleichbar mit GATE ist **UIMA** (*Unstructured Information Management Applications*), ein ursprünglich bei der IBM entwickeltes Framework/SDK.

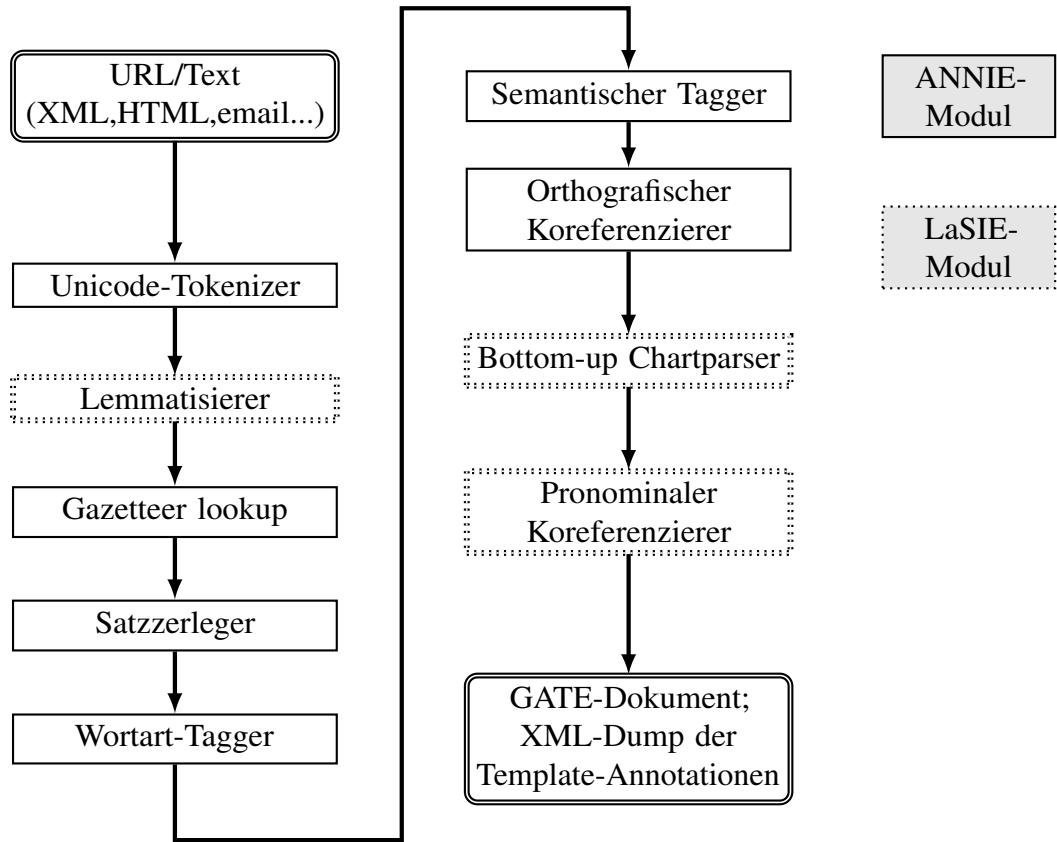


Abbildung 5.3: **GATE-basiertes System**

Der **orthografische Koreferenzer** erkennt, dass sich z. B. „Papst Benedikt“ und „der Papst“ auf dieselbe Person beziehen, der **pronominalen Koreferenzer** bewerkstelligt die Auflösung pronominaler Bezüge im Text (worauf sich „er“, „sie“, „es“ o.ä. beziehen).

5.4.3 KNOWITALL und TEXTRUNNER

Das System **KNOWITALL** (*Etzioni et al. 2005*) repräsentiert eine Architektur für ein *open domain IES* und legt den Schwerpunkt auf drei Aspekte:

- **Domänenunabhängigkeit:** Eingabe sind domänenunabhängige Extraktionsmuster anstelle domänenspezifischer Muster; IE aus heterogenen Informationsquellen soll möglich werden
- **Skalierbarkeit:** mit dem Web als Informationsquelle soll die Verarbeitung großer Datenmengen möglich werden; dies wird durch Nutzung von Suchmaschineninformationen ermöglicht; gleichzeitig wird die Geschwindigkeit der IE erhöht
- **Automatisierung:** sie wird durch unüberwachte Lernverfahren bei der Bildung von Extraktionsregeln und dem dadurch möglichen Verzicht auf manuell erstellte Trainingsdaten erreicht

Gegenstand sind allerdings leichtere Extraktionsaufgaben wie das Auffinden von Instanzen bestimmter Klassen und Relationen. Ziel ist es, die Anzahl der Treffer aus inhomogenen Informationssourcen zu maximieren (hoher recall) und gleichzeitig einen hohen Grad an Genauigkeit (precision) zu gewährleisten.

KNOWITALL erwartet als Eingabe eine Anzahl domänenunabhängiger Muster (*rule templates*, als Beispiel s. (1)) und eine Menge von Prädikaten, die den Fokus der Informationsextraktion bilden (und die jeweils den Kern eines Musters bilden). Im einfachsten Fall sind Prädikate Klassennamen (z. B. STADT, LAND, KINOFILM), im komplexeren Fall Relationsnamen, für deren Argumente Klassen angegeben werden müssen (STADT befindet_sich_in LAND, PERSON spielt_mit_in KINOFILM). Für jedes Prädikat werden wenige Varianten (z. B. Stadt, Kleinstadt, Großstadt für STADT) als *labels* angegeben.

(1)	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding-right: 20px;"><i>Prädikat :</i></td><td>KLASSE</td></tr> <tr> <td><i>Muster :</i></td><td><i>NP1 „so wie“ NPListe2</i></td></tr> <tr> <td><i>Beschränkungen :</i></td><td><i>head(NP1) = plural(label(Klasse))</i> <i>eigenname(head(each(NPListe2)))</i></td></tr> <tr> <td><i>Bindungen :</i></td><td><i>Klasse(head(each(NPListe2)))</i></td></tr> </table>	<i>Prädikat :</i>	KLASSE	<i>Muster :</i>	<i>NP1 „so wie“ NPListe2</i>	<i>Beschränkungen :</i>	<i>head(NP1) = plural(label(Klasse))</i> <i>eigenname(head(each(NPListe2)))</i>	<i>Bindungen :</i>	<i>Klasse(head(each(NPListe2)))</i>
<i>Prädikat :</i>	KLASSE								
<i>Muster :</i>	<i>NP1 „so wie“ NPListe2</i>								
<i>Beschränkungen :</i>	<i>head(NP1) = plural(label(Klasse))</i> <i>eigenname(head(each(NPListe2)))</i>								
<i>Bindungen :</i>	<i>Klasse(head(each(NPListe2)))</i>								

Die Verarbeitung in **KNOWITALL** verwendet eine flache syntaktische Analyse, um basale Kategorisierungsfehler zu vermeiden. So bewirken beispielsweise die Beschränkungen in (1), dass das Muster (und seine Instanzen) zwar auf Sätze wie (2.a) matchen, aber nicht auf solche wie (2.b): der Kopf von NP1 ist die Pluralform eines der Labels (z. B. 'Städte') und der Kopf einer NP aus NPListe2 muss ein Eigenname sein. Durch die Bindungen wird jeder Match aus NPListe2 der Klasse zugewiesen.

- (2) a. *Städte so wie Brindisi, Bari, Lecce, Taranto und Matera sind in knapp einer Stunde mit dem Auto erreichbar.*
 b. *Fachaufsichtsbehörde für die Landkreise und die kreisfreien Städte so wie oberste Fachaufsichtsbehörde für die amtsangehörigen Städte und Gemeinden ist die fachlich zuständige oberste Landesbehörde.*

Die in Abbildung 5.4 auf der nächsten Seite dargestellte Architektur von **KNOWITALL** besteht im Wesentlichen aus drei Komponenten. Die **Bootstrapping-Komponente** erzeugt automatisch aus allgemeinen Mustern⁴ und den jeweiligen Prädikaten einerseits domänenspezifische *Extraktionsregeln* und andererseits *Diskriminatoren* (einfache Aussagen wie „X ist eine Stadt“, „X spielt in Y“)⁵, die von der Bewertungskomponente verwendet werden.

Anhand der nicht-variablen Teile eines Extraktionsregelmusters (z. B. „Städte so wie NPListe2“) werden die *Schlüsselwörter* für eine Suchanfrage entnommen (also 'Städte so wie'). Die von der Suchmaschine gelieferten Webseiten werden durch den **Extraktor** einzeln mithilfe der Extraktionsregel flach analysiert.

⁴ Weitere Muster sind z. B. „NP ist eine <Klasse>“, „NP oder andere <Klasse>“.

⁵ Die Variablen werden von der Bewertungskomponente mit den extrahierten NPs instantiiert.

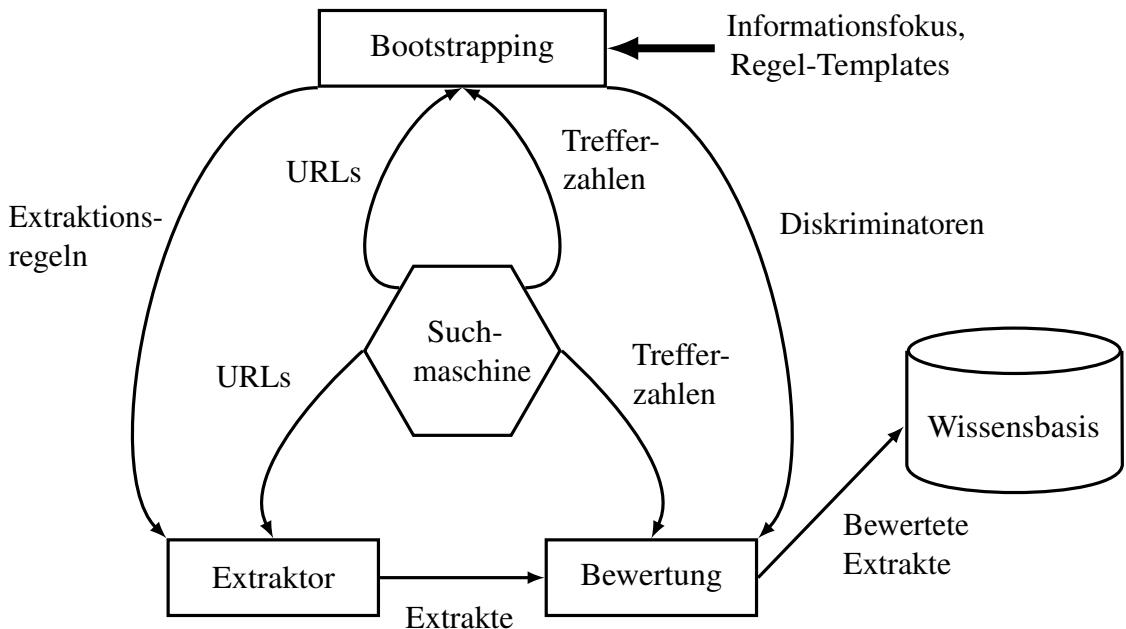


Abbildung 5.4: Architektur von KNOWITALL

Die Kernidee von KNOWITALL besteht darin, dass die **Bewertungskomponente** die Qualität der gefundenen Extrakte überprüft. Dazu wird die relative Wahrscheinlichkeit des gemeinsamen Auftretens eines Extrakts mit den von der Bootstrapping-Komponente automatisch erstellten Diskriminatoren berechnet. Diese *pointwise mutual information* (PMI) lässt sich auf einfache Weise über die Webindizes der Suchmaschine ermitteln (s. (3): I=Instanz=Extrakt, D=Diskriminator).⁶ Mit einem spezifischen Verfahren lässt sich dann der Schwellwert ermitteln, oberhalb dessen ein Extrakt als tatsächliche Instanz einer Klasse gelten kann (Extrakt und Wahrscheinlichkeit werden in jedem Fall in die Wissenbasis von KNOWITALL eingetragen).

Etzioni et al. zeigen, dass zusätzlich zu den o.g. Aspekten mit weiteren Verfahren (Lernen von Mustern, Subklassen-Lernen und Listenextraktion) ein erheblicher Zuwachs an recall erreicht wird, bei etwa gleichbleibender Genauigkeit von 90%.

$$(3) \quad PMI(I, D) = \frac{|Treff(D+I)|}{|Treff(I)|}$$

Ein wesentlicher Nachteil von KNOWITALL ist die Tatsache, dass bei der Verarbeitung eine erhebliche Anzahl an Suchmaschinenanfragen generiert werden, insbesondere weil jedes Prädikat einzeln abgearbeitet wird.

Das Nachfolgesystem TEXTRUNNER (*Banko et al.* 2007) fokussiert auf die Extraktion zweistelliger Relationen. Als konsequente Weiterentwicklung von open domain IE analysiert

⁶ An dieser Stelle baut KNOWITALL stark auf die Größe und Redundanz des Webs.

es ein Textkorpus vollständig automatisch auf relevante Relationen, ohne einen spezifischen menschlichen Input (Fokus, Muster) zu benötigen (sog. *open information extraction (OIE)*). Hierzu wird das Korpus (hier: 9 Mio Webseiten) in *einem* Durchgang analysiert. Vorher wird nur anhand eines relativ kleinen Trainingskorpus die „Güte“ einer extrahierten Relation gelernt.

Als Resultat ergibt sich –auch durch Verzicht auf tiefe Verarbeitung– eine erheblich verbesserte Effizienz in **TEXTRUNNER** sowie die Garantie der Skalierbarkeit (die Verarbeitungszeit steigt nur linear mit der Korpusgröße). Außerdem konnte ein Zuwachs an Effektivität festgestellt werden: verglichen mit ähnlichen automatischen Ansätzen (und erst recht mit manuellen Verfahren) ergab die automatische Relationsextraktion eine sehr viel höhere Zahl unterschiedlicher Relationen.⁷ *Banko et al.* weisen außerdem darauf hin, dass anhand der gefundenen Tupel ein invertierter Index erstellt wird, der für ein Benutzer-freundliches *question answering* (s. Kapitel 7) verwendet werden kann.

5.5 Evaluation: Die Message Understanding Conference (MUC)

Der Ablauf innerhalb einer MUC bestand aus zwei Phasen. In der ersten Phase (Vorbereitungsphase) wurde zunächst die zu behandelnde Domäne ausgesucht. Abhängig davon wurden die IE-Aufgabe(n) festgelegt, die Form des verlangten Outputs spezifiziert (d.h. eine Menge von Templates definiert) und ein Textkorpus aufgebaut (geteilt in Trainings- und Testkorpus). Die Dokumente des Korpus wurden dann von Experten manuell unter den gegebenen Vorgaben analysiert und annotiert. Das Ergebnis waren Referenzwerte, wann welches Template bzgl. eines Textes wie ausgefüllt wurde. Daraufhin erhielten die Teilnehmer die Trainingsdaten für die Entwicklung. Die **in den MUC verwendeten (Template)Aufgaben** sind die folgenden:

- **Named Entity Aufgabe/Eigennamenerkennung (NE):**
Das Markieren (*mark-up*) von Personen, Organisationen, Lokationen etc.
- **Koreferenz (Coreference) Aufgabe (CO):**
Die Markierung, welche sprachlichen Ausdrücke (NE, Anaphern etc.) sich auf dasselbe Objekt beziehen.
- **Template Element Aufgabe (TE):** Das Extrahieren der Eigenschaften von NEs.
- **Template Relation Aufgabe (TR):**
Das Extrahieren einfacher Beziehungen zwischen NEs (z. B. ANGESTELLTER_VON).
- **Scenario Template Aufgabe (ST):**
Das Extrahieren komplexer Beziehungen
(z. B. JOINT_VENTURE: wer und was ist beteiligt?)

In der zweiten (Evaluations-)Phase wurden diese Aufgaben von den teilnehmenden Systemen anhand von Testdaten bewältigt. Mithilfe einer zunehmend automatisierten Bewertungsprozedur

⁷ wobei bei den ca. 60 Mio extrahierten Relationstupeln die Schwierigkeit der Evaluation der Systemresultate deutlich wird

wurden die Systemausgaben mit den Referenzwerten verglichen und zu aussagekräftigen Ergebniswerten zusammengefasst. Die Bewertung basierte auf einer Tabelle, die für jeden Eintrag eines Frames enthält, wie oft er vom Experten ausgefüllt wurde und wie oft er vom System ausgefüllt wurde, und dazu jeweils die Anzahl der genauen Übereinstimmungen (matches), der partiellen Übereinstimmungen, inkorrektener Einträge (Eintrag im falschen Feld), unbekannter Einträge (kein match) sowie fehlender Einträge.

Abbildung 5.5 zeigt ein Beispiel für XML-ausgezeichnete NE. Hier markiert ENAMEX eine 'Entity Name expression', NUMEX eine 'Numerical expression'. Abbildung 5.6 veranschaulicht das Mark-up von Koreferenzen und Abbildung 5.7 auf der nächsten Seite gibt ein Beispiel für ein instantiiertes Scenario-Template zu einem gegebenen Text (aus: *Grishman und Sundheim 1996*).

Mr. <ENAMEX TYPE='PERSON'>Dooner</ENAMEX> met with <ENAMEX:TYPE= 'PERSON'>Martin Puris</ENAMEX>, president and chief executive officer of <ENAMEX TYPE= 'ORGANIZATION'> Ammirati & Puris</ENAMEX>, about <ENAMEX TYPE= 'ORGANIZATION'> McCann</ENAMEX> s acquiring the agency with billings of <NUMEX TYPE= 'MONEY'>\$400 million</NUMEX>, but nothing has materialized.

Abbildung 5.5: **Beispiel für NE**

Maybe <COREF ID='136' REF='I34'>he</COREF>'II even leave something from <COREF ID='138' REF='I39'><COREF ID='137' REF='I36'>his </COREF> office </COREF> for <COREF ID='I40' REF='91'>Mr. Dooner</COREF>. Perhaps <COREF ID='144'>a framed page from the New York Times, dated Dec. 8, 1987, showing a year-end chart of the stock market crash earlier that year </COREF>. <COREF ID='I41' REF='I37'>Mr. James</COREF> says <COREF ID='142' REF='I41'> he</COREF> framed <COREF ID='143' REF='I44' STATUS='OPT'>it</COREF> and kept <COREF ID='145' REF='I44'>it</COREF> by <COREF ID='146' REF='I42'>his </COREF> desk as a „personal reminder. It can all be gone like that.“

Abbildung 5.6: **Beispiel für CO**

Betrachtet man die Ergebnisse der MUC-Evaluationen (aufgeführt in Tabelle 5.2 auf Seite 72), so zeigt sich, dass maschinelle IE insgesamt recht schwierig ist. Zwar liegt der F-Wert bei NE bei über 90% (und belegt, dass diese spezifische Technik „reif“ ist), jedoch schafft es der „allgemeine Fall“, das Füllen eines Scenario-Templates ST, nicht über die 60%-Hürde – ein recht mageres Ergebnis und ein Zeichen für die prinzipielle Schwierigkeit allgemeiner IE. Hobbs (2002) schlägt vier mögliche Gründe als Ursache für diese Hürde vor:

- mangelhafte Performanz beim Zusammenfügen der Information (insbesondere: Koreferenzresolution)

McCann has initiated a new so-called global collaborative system, composed of world-wide account directors paired with creative partners. In addition, Peter Kim was hired from WPP Grout's J. Walter Thompson last September as vice chairman, chief strategy officer, world-wide.

```

<SUCCESSION_EVENT-9402240133-3> :=
    SUCCESSION_ORG: <ORGANIZATION-9402240133-1>
    POST: "vice chairman, chief strategy officer,
          world-wide"
    IN_AND_OUT: <IN_AND_OUT-9402240133-5>
    VACANCY_REASON: OTH_UNK

    < IN_AND_OUT-9402240133-5> :=
        IO_PERSON: <PERSON-9402240133-5>
        NEW_STATUS: IN
        ON_THE_JOB: YES
        OTHER_ORG: <ORGANIZATION-9402240133-8>
        REL_OTHER_ORG: OUTSIDE_ORG
    <ORGANIZATION-9402240133-1> :=
        ORG_NAME: "McCann"
        ORG_TYPE: COMPANY
    <ORGANIZATION-9402240133-8> :=
        ORG_NAME: "J. Walter Thompson"
        ORG_TYPE: COMPANY
    <PERSON-9402240133-5> :=
        PER_NAME: "Peter Kim"

```

Abbildung 5.7: Beispiel für ST

- gegenwärtige IE schafft nur die Extraktion *expliziter* Information; der Rest würde **Inferenzen** erfordern
- 60% ist das, was man mit vernünftigem Zeitaufwand schafft (Monat(e)), 65% würden bereits ein Jahr dauern usw.
- die Hürde ergibt sich aus der Multiplikation der Fehler bei der Erkennung von Teilen eines Satzes (bei 4 Entitäten pro Teilsatz und einer Genauigkeit von 90% macht das $0,9^4 \approx 0,65\%$)

Er plädiert für eine pragmatische Sicht auf die Hürde: In Bezug auf realistische IE-Anwendungen (Sichten interessanter Information aus großen Textmengen) ist ein Recall von 60% besser als

	NE	CO	TE	TR	ST
MUC-3					R < 50% P < 70%
MUC-4					F < 56%
MUC-5					JV F < 53% ME F < 50%
MUC-6	F < 97%	R < 63% P < 72%	F < 80%		F < 57%
MUC-7	F < 94%	F < 62%	F < 87%	F < 76%	F < 51%

Tabelle 5.2: **MUC-Evaluationswerte**

keine Information. Eine Genauigkeit von 60% ist besser als die Alternative stundenlangen Sichtens von Suchmaschinenergebnissen mit viel geringeren Genauigkeitswerten. Eine genauere Betrachtung der Experten zeigt übrigens, dass IE auch für Menschen nicht so einfach ist. Die Analyse der Inter-Annotator-Übereinstimmung ergab, dass diese nur zwischen 60 und 80% liegt. Hinzu kommt, dass ein durchschnittlicher Analyst das 40-fache an Zeit braucht, verglichen mit maschineller IE.

5.6 Literatur

Eine wunderbare Einführung in die Informationsextraktion bietet *Appelt und Israel* (1999). Gute kürzere Überblicksartikel sind *Cowie und Wilks* (2000) und *Neumann* (2004), etwas elaborierter sind *Gaizauskas und Wilks* (1998) und *Cunningham* (2005b). *Neumann* (2009) bietet eine sehr gute aktuelle Darstellung der Informationsextraktion als Bestandteil modernen Informationsmanagements.

Ein geschichtlicher Überblick über die MUCs findet sich in *Grishman und Sundheim* (1996). Eine exzellente konzise Darstellung von IE sowie insbesondere etlicher IES bietet *Kaiser und Miksch* (2005).

Kapitel 6

Natürlichsprachliche Schnittstellen zu Datenbanken (NLIDB)

Tony Stark: *Okay, let's see what this thing can do. What's SR-71's record?*
Jarvis: *The altitude record for fixed wing flight is 85,000 feet, sir.*

[Jarvis: 'Just A Rather Very Intelligent System']

(aus: *Iron man (Film)*)

Natürlichsprachliche Schnittstellen zu Datenbanken (*natural language interfaces to databases (NLIDB)*) erlauben es einem Benutzer, Fragen an Datenbanken in natürlicher Sprache zu formulieren (und nicht in einer formalen DB-Anfragesprache).

Im Gegensatz zu Frage-/Antwort-Systemen (Kapitel 7) sind die Antworten allerdings nicht notwendigerweise sprachlich, sondern in der DB-üblichen Ausgabeform (also z. B. Tabellen). Gleichzeitig ist nur ein eingeschränkter Zugriff auf den Diskurskontext (mit Anaphern oder Ellipsen) möglich, wie im folgenden Beispiel.

- > *Who is the youngest employee in the sales department?*
John Smith.
- > *What is his salary?*
\$25000.
- > *Does any employee in the sales department earn less?*
Yes, George Adams.

Solche Datenbank-„front ends“ (die DB stellt das „back end“ dar) waren eine der ersten Anwendungen in der maschinellen Sprachverarbeitung und als Forschungsthema zu natürlichsprachlichen Schnittstellen beliebt: „[t]he undisputed favorite application for natural language interfaces has been data base query.“ (Carbonell 1984). Man glaubte, dass die Aufgabe eingeschränkt genug war, um durchführbar zu sein (gleichzeitig erhielt man direkt ein praktisches „Ergebnis“), dass natürlichsprachliche Schnittstellen sehr nützlich sein würden und dass die Erfahrung für die Entwicklung komplexerer Aufgaben wertvoll sein würde.

6.1 Allgemeines zu NLIDB

Betrachtet man eine Beispielanwendung (universitäre Lehre-Daten gemäß dem Datenbankschema in Abbildung 6.1), so sieht man, dass einer einfachen Frage wie (4.a) in der Regel eine komplexe Anfrage auf Datenbankebene (s. (4.b)) gegenüber steht.

- (4) a. *Who teaches Smith mathematics?*

b.

```
SELECT lPeople.Name
      FROM People lPeople, People sPeople, Takes, Courses
            WHERE lPeople.Id = Courses.LecturerId
                  AND (Courses.Name = 'Basic Math' OR 'Algebra')
                  AND Courses.Id = Takes.CourseId
                  AND Takes.StudentId = sPeople.Id
                  AND sPeople.Name LIKE '%Smith';
```

PEOPLE	<i>Id^a</i>	Name	Adress
STUDENTS	<i>StudentId</i>		
LECTURERS	<i>LecturerId</i>	Salary	
COURSES	<i>CourseId</i>	CourseName	LecturerId
TAKES	<i>StudentId</i>	<i>CourseId</i>	Grade
REQUIRES		<i>CourseId</i>	RequiredCourseId

^a Ein *kursiver* Eintrag kennzeichnet den Primärschlüssel, ein **fett gedruckter** Eintrag den Fremdschlüssel.

Abbildung 6.1: Datenbankschema Lehre-Daten

Die Vorteile von NLIDB sind generell

- Natürlichkeit: das Fehlen einer (mühsam zu lernenden) künstlichen Sprache.
- Universalität: Allgemeinheit, Domänenunabhängigkeit, Portabilität.
- Ausdrucksmächtigkeit: die bessere Eignung für bestimmte Konstrukte in Anfragen (insbesondere Allquantifikation, Negation): „Nenne mir alle Abteilungen, die (nicht) WindowsNT verwenden.“.
- Flexibilität: die Möglichkeit, denselben Sachverhalt auf verschiedene Arten auszudrücken.
- Effektivität: selbst komplexe Anfragen können relativ einfach gestellt werden.
- (die wenn auch beschränkte) Diskursfähigkeit, die ein Einschränken der gewünschten Antworten erlaubt.
- Meta-Level-Anfragen: Anfragen über die Struktur der DB

Diesen Vorteilen stehen einige **Nachteile der NLIDB** gegenüber:

- Sprachliche Abdeckung ist (dem Benutzer) nicht klar, dies führt zu:
Falschen positiven Annahmen: System kann eine Anfrage X, aber nicht eine strukturell ähnliche Y beantworten, z. B. in **MASQUE**:
 - > *What are the capitals of the countries bordering the Baltic and bordering Sweden? [ok]*
 - > *What are the capitals of the countries bordering the Baltic and Sweden? [¬ ok]*

Falschen negativen Annahmen: System kann eine Anfrage X nicht, aber dafür eine strukturell ähnliche Y beantworten

- Unklarheit bei Fehlern, ob sie auf mangelnder sprachlicher Abdeckung (*linguistic coverage*) oder konzeptueller Abdeckung (*conceptual coverage*) beruhen
- Benutzer erwarten mehr Intelligenz des Systems als vorhanden ist. Dies führt erstens in die Irre und zweitens zu nachhaltiger Enttäuschung.
- NL ist nicht immer angemessenes Medium: Natürlichsprachliche Ausdrücke sind oft ambig; das Stellen von NL-Fragen kann aufwändiger sein als das selektive Ausfüllen von Anfrageformularen bzw. das Anklicken von Optionen
- Mühsame Konfiguration (Lexikonaufbau und -pflege)

Ein allgemeines **Problem** bei Anfragen an DBs ist die Unklarheit darüber, ob oder wann ein **Fehler** vorliegt:

> *How many students failed CS 381K in Summer 1987?*

0

Hier ist nicht klar, ob a) keiner durchgefallen ist, b) CS 381K im Sommer 1987 nicht angeboten wurde oder c) es gar keinen Kurs CS 381K gibt. Bei den Fällen b) und c) handelt es sich gemäß linguistischer Terminologie um *Präsuppositionsvorstellungen*, d.h. die Voraussetzungen für eine sinnvolle Fragebeantwortung sind gar nicht gegeben.

Weitere Fehler, die bei NLIDB auftreten können, sind Schreibfehler, Missverständnisse (z. B. bei Präsuppositionsvorstellungen oder bei vagen Anfragen (*Which lecturer teaches many courses?*)) oder Systemfehler (keine Antwort trotz angemessener Frage).

Für eine **effektive Verwendung** des NLIDB ist somit eine **robuste Verarbeitung** notwendig, die diese Fehler auffängt (automatische Korrektur bei Schreibfehlern, Anwendung von Heuristiken bei Syntaxfehlern). Gleichzeitig ist es unumgänglich, dem Benutzer **Feedback** über vorliegende Fehler und Probleme zu geben (z. B. eine Rückmeldung über Präsuppositionsvorstellungen). Hierfür bieten sich generell natürlichsprachliche Umschreibungen einer Anfrage (Paraphrasen) an.

6.2 Geschichte

Seit den 1960er Jahren existieren Forschungssysteme zu natürlichsprachlichem Zugriff auf Datenbanken. Beispiele sind das **BASEBALL**-System (*Green et al.* 1963) und **LUNAR** (*Woods et al.* 1972). Im System **RENDEZVOUS** (*Codd 1974*) wurden dem Benutzer bereits Paraphrasen für eine natürlichsprachliche Anfrage dargeboten, damit die Korrektheit der Interpretation des Systems nachvollziehbar/einsehbar war und so falsche Annahmen des Benutzers vermieden werden konnten.

Baseball (*Green et al. 1963*)

Das System **BASEBALL** beantwortete Fragen zu Monat, Tag, Ort, Mannschaft und Ergebnis zu jedem Baseballspiel eines bestimmten Jahres, wobei die entsprechenden Informationen in einer Datenbank abgelegt war.

Auf der Ebene der Abfragesprache gab es die Möglichkeit, eine komplexe Frage (z. B. „An wievielen Tagen im Juli spielten 8 Mannschaften?“) auf eine Abfolge einfacherer Anfragen abzubilden, wobei das Ergebnis einer Anfrage die Eingabe der nächsten darstellte (*embedded query*):

[Für jeden Tag im Juli: Welche Mannschaften spielten?] → [31 Listen von Spiellisten]
→ [Davon: An welchem Tag spielten 8 Mannschaften?]

Auf sprachlicher Ebene fand schon das statt, was heute **Eigennameerkennung** (*named entity recognition*) genannt wird (z. B. die Erkennung komplexer Terme wie ’New York’). Anhand flacher syntaktischer Verarbeitung wurden Subjekt (S) und Objekt (O) bei folgenden Phänomenen identifiziert:

- Passivsätze:
What teams [O] were beaten by the Red Sox [S]?
- Objektvorstellung:
How many games [O] did the Yankees [S] play in July?
- Normalstellung:
What teams [S] won ten games [O] in July?

Semantisch wurden Determinatoren (z. B. *which, who*), Modifikatoren (z. B. *winning*) und Verbbedeutungen (z. B. *beat*) auf entsprechende Anfragen abgebildet und **semantische Ambiguitäten** (z. B. zwischen Ort und Mannschaftsnamen) aufgelöst:

Who beat the Yankees on July 4?
 ↪ TEAM(winning)=?
 TEAM(losing)=YANKEES
 MONTH=JULY
 DAY=4

Die frühen Systeme wiesen allerdings in der Regel eine enge Verzahnung von Sprache und Datenbank auf, d.h. insbesondere eine enge „Passung“ der sprachlichen Strukturen und der jeweiligen Datenbankterminologie. Ein Beispiel hierfür ist die Verwendung **semantischer Grammatiken** (**PLANES**, s. Waltz 1978; **LADDER**, s. Hendrix et al. 1978), bei denen sprachliche Ausdrücke nicht (ausschließlich) anhand syntaktischer Kriterien modelliert wurden, sondern auch durch semantische (inhaltliche) Kriterien. Dies bedeutete das Schreiben spezifischer Regeln, die zwar die Ambiguität der Sprache reduzierten und die Verarbeitung erleichterten, gleichzeitig aber eine (zu) enge Verknüpfung mit der jeweiligen Anwendungsdomäne mit sich brachten.

Semantische Grammatiken (*Hendrix et al. 1978*)

Hier ein Ausschnitt aus der Grammatik von **LADDER**, die semantisch motivierte Kategorien aufwies:

```

S -> <present> the <attribute> of <ship>
S -> How many <ship> are there
<present> -> what is | [can you] tell me
<attribute> -> length | beam | class
<ship> -> the <shipname> | <classname> class ships
<shipname> -> Kennedy | Enterprise
<classname> -> Kitty Hawk | Lafayette

```

Hiermit ließen sich Fragen wie die folgenden stellen:

- *How many Lafayette class ships are there?*
- *What is the length of the Kennedy?*
- *Can you tell me the class of the Kennedy?*

In den 1980er Jahren rückte der Aspekt der **Portabilität** von NLIDB in den Vordergrund. Sie wurde durch die Annahme einer logischen Zwischenrepräsentation (z. B. in **CHAT-80**, s. Warren und Pereira 1982 und den Exkurs auf der nächsten Seite) möglich, durch die das NLIDB und die DB austauschbar wurden. Weitere behandelte Themen waren die Konfigurierbarkeit durch Nicht-Computerlinguisten (z. B. in **TEAM**, s. Grosz 1983) und einfache Erweiterbarkeit des Lexikons (z. B. in **ASK**, s. Thompson und Thompson 1983) von NLIDB.

Seit den 1980er Jahren wurden –meist aus den Erfahrungen der Forschungsprototypen heraus– die ersten kommerziellen Systeme entwickelt. Beispiele hierfür sind **INTELLECT**, **EASYASK** (Nachfolger von **INTELLECT**), **ENGLISH QUERY** (Microsoft).

In den 1980er Jahren begann das Interesse an NLIDB zu schwinden. Grundlegende Probleme waren identifiziert und behandelt, kommerzielle Systeme wurden entwickelt. Entsprechend wurde die Forschung an NLIDBs als zu eingeschränkt empfunden. Das Interesse in der angewandten Forschung veränderte sich in unterschiedliche Richtungen:

Chat-80

Das Prolog-System entstammt der Dissertation von Fernando Pereira (einem Pionier der logisch-orientierten CL) und beinhaltet eine systematische logisch-semantische Behandlung natürlichsprachlicher Phänomene von Fragen und Fragenbeantwortung (s. Warren und Pereira 1982).

CHAT-80 setzt den Schwerpunkt auf die Erstellung einer logischen Repräsentation von Fragen, anhand derer Anfragen an eine Prolog-Wissensbasis mit geografischer Information gestellt werden können.

Auszug aus der Wissensbasis:

```
% country(Country,Region,Latitude,Longitude,
%       Area/1000,Area mod 1000,
%       Population/1000000,Population mod 1000000 / 1000,
%       Capital,Currency)
country(afghanistan,indian_subcontinent,33,-65,254,861,18,290,
        kabul,afghani).
country(albania,southern_europe,41,-20,11,100,2,350,tirana,lek).

city(athens,greece,1368). city(bangkok,thailand,1178).
borders(albania,greece). borders(albania,mediterranean).
sea(mediterranean).
```

Beispiele für Fragen und anhand der Wissensbasis generierten Antworten:

> what percentage of countries border each ocean?

[arctic_ocean,2], [atlantic,35], [indian_ocean,14] and [pacific,20].

> what percentage of countries border an ocean?

60

> what are the continents no country in which contains more than two cities whose population exceeds 1 million?

[africa,antarctica,australasia].

> does afghanistan border china?

Yes.

> what is the total area of countries south of the equator and not in australasia?

10228 ksqmiles.

- Loslösung hin zu Textverstehen, Wissenrepräsentation
- Erweiterung hin zu Dialogen und Benutzerorientiertheit
- Veränderung hin zu textuellen Applikationen (Information retrieval, Summarizing)
- Verlangen nach Robustheit

Heute, nach mehr als 30 Jahren theoretischer und praktischer Erfahrung im Bereich der NLIDB muss konstatiert werden, dass es eine Schnittstelle, die

- mehr oder weniger unrestringierten NL-Input akzeptiert und korrekt interpretiert
- für ungeübte Benutzer einfach verwendbar ist
- leicht an unterschiedliche Datenbasen angepasst werden kann

nicht gibt. Reale NLIDB erfordern immer noch einen hohen Konfigurationsaufwand. Abbildung 6.2 deutet an, dass ein natürlichsprachlicher Zugriff auf Datenbanken mehr als nur den Einzeler der Frage benötigt. Entsprechend existieren mit grafischen oder Formular-basierten ID erfolgreiche Alternativen zu NLIDB, deren Funktion und Fähigkeit transparenter für den (naiven) Benutzer ist und die außerdem leichter zu konfigurieren sind. Erst neuere NLIDB-Systeme setzen hier mit flachen Verfahren an, die Konfigurationsaufgabe automatisch zu lösen.

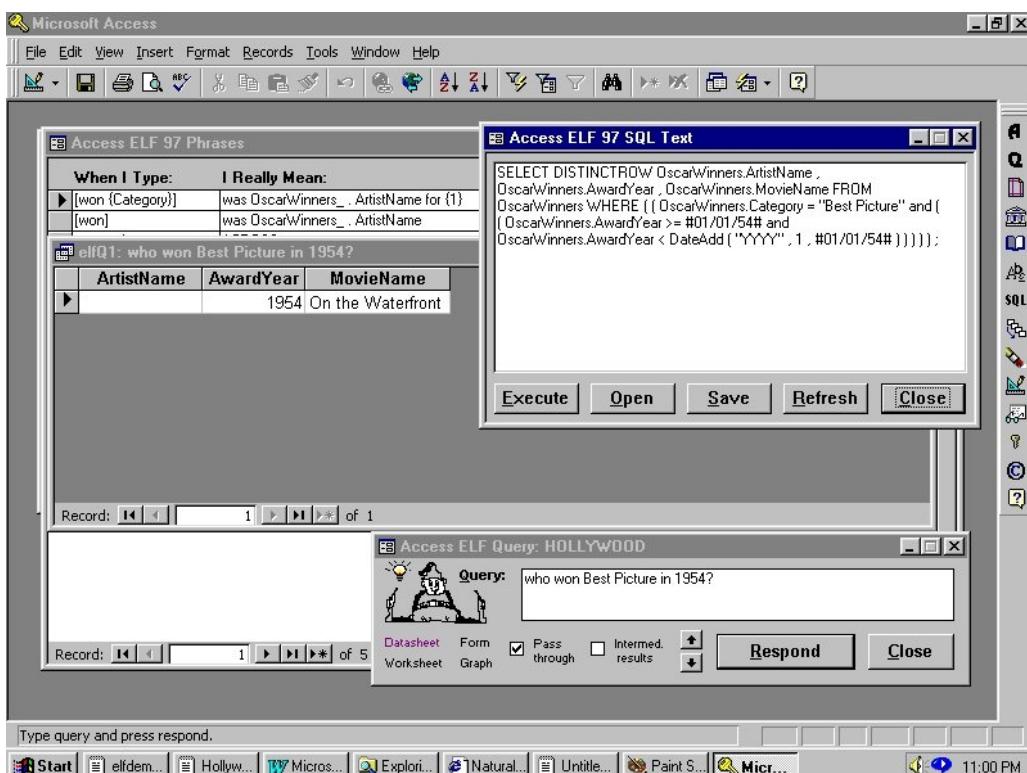


Abbildung 6.2: Screenshot von Access Elf

Einen ganz anderen Ansatz verfolgt **WOLFRAM|ALPHA**. Die Idee hier war nicht, eine Schnittstelle zu einer Datenbank zu entwickeln, sondern zu einer *Wissensbank*, wobei die dort enthaltene Information systematisch aus Datenbanken gewonnen (importiert) wird (sog. 'curated data'). Anhand dieser Wissensbank können einfache Fragen direkt beantwortet werden, komplexere ggf. indirekt mithilfe zusätzlicher (Inferenz-/Schlussfolgerungs-)Methoden. Auf diese Weise verlagert sich das Problem der NLIDB hin zu dem Problem der Konzeption einer 'Antwortmaschine'.

6.3 Aspekte und Anforderungen

6.3.1 Probleme

Die Verwendung von NLIDB bringt eine Reihe **linguistischer Probleme/Aspekte** mit sich, die im Allgemeinen aus der Mehrdeutigkeit (**Ambiguität**) des sprachlichen Ausdrucks resultieren (die aufgelöst und ggf. dem Benutzer zur Auswahl angeboten werden müssen)¹. Hierzu gehören:

- **Lexikalische Ambiguität:** Mehrdeutigkeit einzelner Wörter
 - > *Which courses does every lecturer teach?*
[Ausschluss von *golf courses* u.ä. notwendig.]
- **Modifier attachment:**
Die Frage, an welche Konstituenten Präpositionalphrasen, Relativsätze o.ä. angehängt werden. Im folgenden Beispiel: handelt es sich um Schuh-herstellende Angestellte oder um die entsprechende Abteilung?
 - > *List all the employees in the division making shoes*
[List all the employees <1 which make shoes>in the division <2 that makes shoes>]
- **Distributivität eines Operators:**
Bei koordinierten Strukturen als Argument eines Operators kann dieser sich auf die die Gruppe oder auf deren Bestandteile beziehen.
 - > *List the average salary of scientists and engineers*
[Gehalt beider Gruppen zusammen oder jeder einzelnen?]
- **Quantorenskopus:**
Die Frage, wie die Quantoren in der logischen Form geordnet sind: handelt es sich um einen bestimmten Kurs, oder können die Kurse verschieden sein?
 - > *Has every student taken some course?*

¹ So bietet das in *Biveroni* (2002) –dem die meisten der folgenden Beispiele entnommen sind – analysierte NLIDB **ENGLISH QUERY** eine Paraphrase als Verifikationsmöglichkeit für den Benutzer an. Systemparaphrasen sind in eckigen Klammern durch den Courier-Font angezeigt.

[<1 For some specific course>For every student <2 for some course>has he/she taken it?]

- **Konjunktion & Disjunktion:**

Das sprachliche „und“ kann sowohl Konjunktion als auch Disjunktion ausdrücken.

> *Show the products named both Chai and Chang.*

[List the products whose name is Chai or whose name is Chang.]

- **Referenzambiguität: Anaphorik:** Auf welche Objekte beziehen sich Anaphern?

> *List the managers*

[eine Liste von 5 Namen.]

> *Which of them are Canadian citizens?*

[eine Liste von 2 Namen.]

> *Which of them are experts in foreign contracts?*

[alle, oder nur die Kanadischen?]

- **Ellipsen:** Welche Information ist ausgelassen / muss ergänzt werden?

> *What percentage of employees are professional people?*

[eine Abbildung mit prozentueller Zuordnung.]

> *People above job level 15?*

[1. What percentage of people above job level 15 are professional people?

2. What percentage of employees are above level 15?

3. List the people above job level 15.]

- **Nominalkomposita:**

Implizit existieren generell Relationen zwischen den Nomina, die bei der Umwandlung in eine DB-Anfrage aufgedeckt werden müssen.

> *Show the product customers.*

[Which customers order products?]

> *Show the customer territories.*

[Show the customers that order products since fiscal 1997 and their territories.]

Customers don't have territories.

Employees have territories.

Die Flexibilität der Sprache führt außerdem dazu, dass ggf. Umwandlungen oder sonstige Präzisierungen vorgenommen werden müssen. Beispiele dafür sind

- **Aktiv-/Passiv-Konvertierung:**

Die Umwandlung eines passivischen Ausdrucks in die kanonische aktivische Form

- > *Northwoods Cranberry Sauce is sold by whom?*
[Who sells Northwoods Cranberry Sauce?]
- **Telegrafische Äußerungen:** Die sparsamste Art sprachlicher Interaktion
 - > *Product.*
[List the products.]
 - > *Orders, customer, Alfreds Futterkiste.*
[Show the orders and their customers named Alfreds Futterkiste.]
- **Aggregatfunktionen:**
Zusammenfassen einer Gruppe zu einem Objekt, für das Operatoren definiert sind.
 - > *What is the average age of the employees?*
[What is the average employee age?]
- **Temporale Aspekte:** Anfragen, die zeitliche Ausdrücke enthalten
 - > *Which employees will be older than 50 last year?*²
[Which employees will be more than 50 years old in fiscal 2001?]

Ein spezifisches Problem der NLIDB besteht in der Automatisierung der Abbildung natürlichsprachlicher Bezeichner auf ihre entsprechenden Pendants in der Datenbank. Dass dies nicht immer trivial ist zeigt das sogenannte "Doktor-an-Bord"-Beispiel. Gegeben ist eine Datenbank zu Schiffen, in denen das Vorhandensein eines Doktors angegeben ist.

Schiff	Crew	Doktor
<i>Vincent</i>	420	<i>y</i>
<i>Invincible</i>	514	<i>y</i>
<i>Sparrow</i>	14	<i>n</i>
...

Bei einer Frage wie *Gibt es einen Doktor auf der Vincent?* wird die Nominalphrase „einen Doktor“ standardmäßig in ein mengenwertiges Prädikat abgebildet (dessen Werte in einer entsprechenden Tabelle erwartet werden). Genau dies liegt hier mit dem binären Doktorattribut jedoch nicht vor. Entsprechend müssen solche Fälle mit komplexeren Mechanismen abgefangen werden.

6.3.2 Anforderungen

An das Design eines NLIDB müssen einige grundlegende Anforderungen gestellt werden. Zentral ist dabei die Trennung der *Daten* in der DB von dem *Wissen*, das für die Beantwortung

² Man beachte, dass das System 'last year' richtig interpretiert, den Konflikt mit der *temporalen* Information (Futur der Kopula) aber nicht erkennen kann bzw. nicht beachtet.

der Fragen benötigt wird. Dies betrifft einerseits die sprachlichen Komponenten (Analysekopponente mit Grammatik und Lexikon), aber auch zusätzliches Wissen über die Domäne, das zur Flexibilität des NLIDB beiträgt. Für das DB-Beispiel in Abbildung 6.1 auf Seite 74 ist es u.a. sinnvoll, über das Wissen zu verfügen,

- dass Studierende und Lehrende Menschen sind
- dass „Basic Math“ und „Algebra“ Mathematikkurse sind
- dass „teach“ einer Relation zwischen Lehrenden und Studierenden ist
- dass Studierende bestimmte Voraussetzungen für das Belegen von Kursen erfüllen müssen (erfolgreiches Belegen anderer Kurse).

Dieses intensionale, terminologische Wissen ist zu unterscheiden von dem extensionalen, assertionalen Wissen, das als Daten in der DB vorliegt, und sollte nicht unbedingt in der Datenbank abgelegt werden. Es kann für Inferenzen verwendet werden, die bei der Übersetzung logischer Repräsentationen in eine Datenbankabfrage benötigt werden, z. B. bei

- **Coercion** („Umwandlung“): der Überführung in eine kanonische Form
 - > *Who are Smith's lecturers?*
[Who lectures the courses which Smith takes?]
- **Konzeptueller Vervollständigung**: der Explizierung impliziter Relationen
 - > *Who teaches maths?*
[Who teaches a course on maths?]

Als Grundanforderungen an ein NLIDB lassen sich daher nennen:

- modularer Aufbau (strikte Trennung von sprachlichen und nicht-sprachlichen Aspekten, d.h., Unabhängigkeit der Analysekopponente); eine logische Form als Repräsentation der sprachlichen Eingabe vermittelt zwischen sprachorientierten und Datenbank-orientierten Modulen
- Portabilität (d.h., Domänenunabhängigkeit, Übertragbarkeit auf andere „database back ends“)
- Erweiterbarkeit (z. B. die Erweiterung eines domänenunabhängigen Kerns des Lexikons mit jeweils domänenspezifischen Terminologieanteilen)

Eine wesentliche Anforderung an NLIDBs ist außerdem ihre **Verlässlichkeit** (reliability), da gerade die „Aufweichung“ der Eindeutigkeit von Ausdrücken einer formalen Datenbankanfragesprache durch natürlichsprachliche Anfragen zu Fehlern führen kann. Für die Gewährleistung der Verlässlichkeit existieren unterschiedliche Ansätze: Das System **PRECISE** (Popescu et al. 2003) berechnet anhand der semantischen „Passung“ von sprachlichen Elementen und Datenbankeinträgen automatisch den Status einer Frage als vom System „behandelbar“ (*semantically tractable questions*) und hält dabei die Modularitätsrichtlinie ein. Das System **STEP** (Minock 2005a) hingegen weist dem Anwender eines NLIDB („administrator“) mehr Gewicht zu, indem dieser wie in dem **RENDEZVOUS**-System (Codd 1974) semantische Grammatiken spezifizieren muss, durch die mögliche Fragen eingeschränkt werden (aber gleichzeitig die Modularität

beeinträchtigt wird). Die Fragen werden auf Ausdrücke eines eingeschränkten Logik-Kalküls abgebildet (und von dort aus wieder zurück für Paraphrasen und Klärungsdialoge), was die Verlässlichkeit sichert.

6.4 Architekturen und Systeme

6.4.1 Grobarchitektur von NLIDB

Die Grobstruktur eines NLIDB ist in Abbildung 6.3 dargestellt. Kernbestandteile sind die Analyse und Übersetzungskomponente sowie die Inferenzkomponente (zentrale Wissensverarbeitung). Im einfachsten Fall werden Ergebnisse der DB-Abfragen direkt ausgegeben, fortgeschrittenere/benutzerfreundlichere Systeme verwenden eine (hier gestrichelte) Generierungskomponente. Die Übersetzungskomponente greift auf Domänenwissen zu.

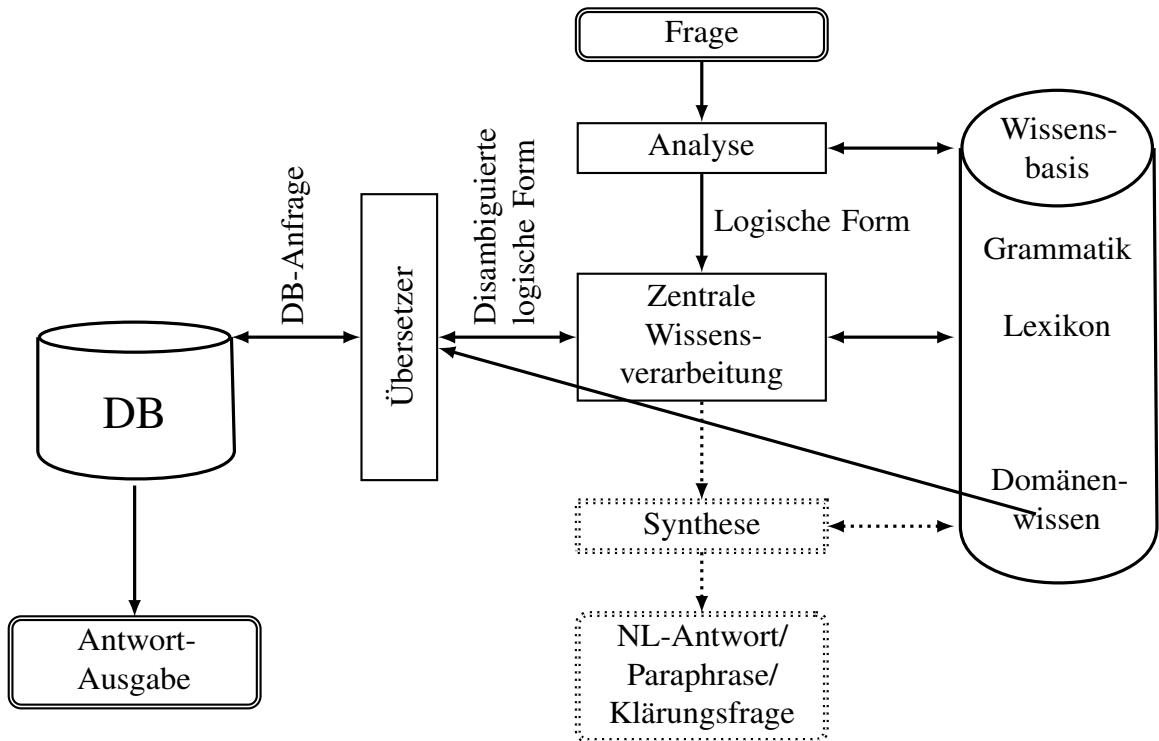


Abbildung 6.3: **Grobstruktur eines NLIDB**

6.4.2 Designempfehlungen für NLIDBs

Ogden und Bernick (1997) weisen darauf hin, dass sich Benutzer eines NLIDBs zwei Sorten von Kenntnissen aneignen müssen, um vernünftig mit ihm umgehen zu können, nämlich bzgl.

- der Fähigkeiten des Systems
- der vom System akzeptierten Sprache, um diese Fähigkeiten nutzen zu können

Sie verweisen auf empirische Analysen, die zeigen, dass im Bereich der NLIDBs in der Regel unvollständige Kenntnisse beider Sorten vorliegen, so dass oft falsche Erwartungen im Hinblick auf die Systemkompetenz vorliegen oder unvollständige Anfragen gestellt werden. Nach den Autoren gibt es zwei generelle Ansätze, solche Kenntnislücken zu vermeiden: zum Einen eine mehr oder weniger elaborierte Anpassung des Systems an mögliche Erwartungen von Benutzern (was erfahrungsgemäß kaum vollständig gelingen wird), zum Anderen ein mehr oder weniger aufwändiges Training der Benutzer (was dem Anspruch eines NLIDB auf Benutzerfreundlichkeit inhärent widerspricht). Auf der Grundlage ihrer Analysen empfehlen sie die Sicherstellung der folgenden Voraussetzungen beim Design eines NLIDB:

- **klare Definition der Anwendungsdomäne und Benutzergruppe.**
- **eine Phase inkrementeller Adaptation.** Erfahrungen zeigen, dass ein auf Benutzerinteraktion basierendes iteratives bzw. inkrementelles Design erfolgreich sein kann.
- **Zugang zu Meta-Wissen.** Da bei natürlichsprachlichem Zugang der Zwang zum Lernen formaler Anfragestrukturen wegfällt, sind Hinweise auf das, was erfragbar ist, sinnvoll.
- **eine breite oder wohl-definierte syntaktische Abdeckung.** Erfahrungen zeigen, dass Benutzer generell Probleme mit transparenten Beschränkungen der sprachlichen Eingabe haben, dass sie sich andererseits aber schnell an klar definierte Beschränkungen gewöhnen. Sinnvoll kann daher die Verwendung sog. **kontrollierter Sprache³** sein, worunter man die Verwendung eines *Ausschnitts* des prinzipiell möglichen Sprachinventars (Syntax, Lexik ...) in einem NLS versteht (s. den Exkurs auf der nächsten Seite).
- **Feedback ermöglichen.** Hierzu gehört einerseits das Anzeigen einer Paraphrase der Benutzereingabe –ein implizites Training für mögliche andere Eingaben– und andererseits das Anzeigen der ausgelösten Systemaktionen.

6.4.3 STEP

Das System **STEP** (von: Schema Tuple Expression Processor) (*Minock 2005a*) verkörpert einen pragmatisch orientierten Ansatz im Schnittstellenbereich von Relationale-Datenbank-Technologie und sprachorientierter KI, der frühe Ansätze der Sprachtechnologie (Muster, semantische Grammatiken) auf clevere Weise mit modernen Ansätzen der Wissensrepräsentation und -verarbeitung (Subsumptionshierarchien, entscheidbare Logiken, Theorembeweiser) verbindet. Dabei wird der Aspekt der Portabilität (und somit der linguistischen Abdeckung) zugunsten der Verlässlichkeit in den Hintergrund gerückt.

³ Ein prominentes Beispiel ist das an der Univ. Zürich entwickelte *Attempto Controlled English*.

PENG – Ein Beispiel für eine kontrollierte Sprache

Eine kontrollierte Sprache beschränkt die möglichen natürlichsprachlichen Eingaben eines Systems und verringert so das Fehlerpotential (z. B. Ambiguitäten). Ein Problem dabei ist allerdings wieder die Intransparenz der sprachlichen Möglichkeiten.

Schwitter (2002) stellt die kontrollierte Sprache **PENG** (Processable ENGlish) vor, die aus einer beschränkten Grammatik und einem domänen spezifischen Lexikon besteht. Anders als andere kontrollierte Sprachen verfügt **PENG** zusätzlich über einen Look-ahead-Editor, der einem Benutzer die Bürde des Lernens der Sprache abnimmt: Er zeigt während der Eingabe die nächstmöglichen Fortsetzungen an. Schwitters Beispiel ist die wortweise Eingabe des Satzes *Wolves are animals*:

```
Wolves [ are | relative clause ]
Wolves are [ noun,pl. | not | comparative clause ]
Wolves are animals [ ! | relative clause | coordination ]
```

Kern des Ansatzes sind **Tupelausdrücke** (auch: *Anfrageausdrücke*) des relationalen Datenbankschemas, d.h. solche Ausdrücke einer beschränkten Logiksprache, die jeweils eine Menge von Tupeln der Datenbank denotieren. Die Beschränkung besteht darin, dass nur Bedingungen über einer getypten freien Variable angegeben werden dürfen, wobei andere Variablen in diesen Bedingungen immer existentiell abgebunden sein müssen. Auf diese Weise wird die Entscheidbarkeit der Verwendung dieser Ausdrücke gesichert (auch wenn nicht mehr alle möglichen Fragen repräsentiert werden können).

Charakteristisch für **STEP** ist der Gebrauch eines **phrasalen Lexikons**, in dem elementaren Tupelausdrücken durch einen „Administrator“ phrasale Muster zugewiesen werden. Ein Eintrag des phrasalen Lexikons ist von der Art $< q : p^+ >$, wobei q den Anfrageausdruck repräsentiert und p^+ eines oder mehrere phrasale Muster, die die Bedeutung von q ausdrücken.

In den Mustern ist jeweils angegeben, in welcher syntaktischen Rolle der sprachliche Ausdruck den Tupelausdruck realisiert (*head*, *complement*, *modifier*, *answer*), mit der optionalen Möglichkeit, Merkmale zuzuweisen (*sing*, *pl*). Der folgenden Einträge geben einfache Beispiele für die Zuordnung sprachlicher Muster zu Tupelausdrücken.

```
<{x|Country(x)}:
  Head['countries'](pl )
  Head['country'](sing )
  Head['nations'](pl )
  Head['nation'](sing )

<{x|Country(x) ∧
  (∃y1)(In_Continent(y1) ∧ y1.country = x.code ∧ y1.continent = 'Asia')}:
```

Modifier [’Asian’]

Complement[’of Asia’]>

Die Muster können auch Parameter enthalten, die einen Text des Musters direkt mit einem Wert aus dem Tupelausdruck assoziieren⁴:

```
<{x|Country(x) ∧ x.population < c1}:
  Complement[’with population less than c1’]
  Complement[’with less than c1 people’]
  Complement[’with fewer than c1 people’]>
```

Während das phrasale Lexikon überwiegend domänenpezifisch ist und so an neue Domänen angepasst werden muss, sind die **Satzschablonen** (*templates*), in denen phrasale Ausdrücke vorkommen können, weitgehend domänenunabhängig. Die Funktion `ref` erschließt den entsprechenden Anfrageausdruck aus der vorkommenden Phrase, während die Funktion `gapped(q, f)` eine topikalisierte Konstituente (`f`) an der richtigen Stelle in `q` einfügt. Satzschablonen als eine Art flacher Repräsentanten syntaktischer Information werden wie folgt bestimmten Frage-Typen zugeordnet

```
<q(value):
  Template [’ref(q)’]
  Template [’(list|give mel...) · ref(q).’]
  Template [’(which|what) · ref(f) · (is|are|d|old|does|has|have) · gapped(q, f )?’]...>
<q(truth):
  Template [’(is|are) · there · ref(q)?’]...>
<q(count):
  Template [’how many · ref(q) · (is|are) · there?’]...>
<q(where):
  Template [’where · (is|are) · ref(q)?’]...>
```

Ein wesentlicher Schritt in **STEP** ist die Kompilation des phrasalen Lexikons in eine **Subsumptionshierarchie**⁵. Sie hat die Funktion, Paraphrasen einer natürlichsprachlichen Frage zu finden, um sie dem Benutzer im Rahmen eines Feedbacks zu präsentieren. Hierzu werden die unmittelbaren Elternknoten des in die Hierarchie eingeordneten Tupelausdrucks der Frage berücksichtigt.

⁴ zu komplexeren Musterspezifikationen s. *Minock* (2005a)

⁵ Eine Subsumptionshierarchie macht explizit, welcher Tupelausdruck spezifischer als welcher andere ist (und somit von diesem *subsumiert* wird). Ein Beispiel: $\{x|Country(x)\}$ subsumiert $\{x|Country(x) \wedge x.population < 1000000\}$.

Die Verarbeitung der Satzschablonen und Anfrageausdrücke gewährleistet (unter Verwendung eines spezifischen Parsers) die Verlässlichkeit des Systems auch bei fehlerhaften Benutzeranfragen und ermöglicht in diesem Fall ein gezieltes Feedback. **Kooperativität** wird in **STEP** durch folgende Maßnahmen erreicht (die Beispiele sind fiktiv):

- Aufdecken von Fehlannahmen und entsprechende Rückmeldung, z. B.:
„Sie möchten eine Festplatte mit 2GB RAM. Es gibt keine Festplatte mit weniger als 20GB Speicher“
- Aufdecken von falschen Präspositionen und entsprechende Rückmeldung, z. B.:
„Sie möchten Ihren Vertrag vorzeitig kündigen. Man kann einen XY-Vertrag nur unter Einhaltung der 1-monatigen Kündigungsfrist zum Jahresende kündigen“
- Paraphrase als Rückmeldung vor eigentlicher Datenbankanfrage zur Bestätigung
- Anfragenrelaxierung (*query relaxation*), wenn keine Antworten gefunden wurden, z. B.:
„Es gibt keine Festplatte mit mehr als 200GB für unter 40Euro, aber es gibt eine Festplatte mit 160GB für 39.95Euro“
- Intensionale Fragebeantwortung⁶, wenn zu viele Antworten gefunden werden, z. B.:
„Sie möchten einen USB-Stick für unter 50 Euro. Dafür gibt es 73 Angebote.“

STEP ist sicher für kleinere Domänen ein überschaubares, verlässliches und benutzerfreundliches System. Es ist allerdings zu vermuten, dass die Nachteile in verschiedenen Bereichen (z. B. linguistische Abdeckung, Transparenz, Konfigurationsaufwand, Effizienz) mit wachsender Domänengröße erheblich zunehmen werden.

6.4.4 PRECISE

Das System **PRECISE** (*Popescu et al. 2003; Popescu et al. 2004*) basiert auf der grundlegenden Beobachtung, dass natürliche Sprache zwar komplex und kompliziert, andererseits aber auch einfach und klar sein kann. *Popescu et al.* versuchen, den Begriff „einfache Frage“ formal als „semantisch/inhaltlich behandelbare Frage“ (*semantically tractable question*) zu erfassen, um **verlässliche NLIDBs** zu ermöglichen. Im Gegensatz zu **STEP** besteht das Ziel dabei nicht darin, benutzerfreundliche Paraphrasen zu ermöglichen, sondern Nicht-Behandelbarkeit zu identifizieren und daraufhin eine Paraphrase des Benutzers einzufordern. Ein weiteres Ziel ist die **Portabilität** des Systems, nicht zuletzt durch die Verwendung eines eigenständigen statistischen Parsers.

Semantisch behandelbare Fragen sind im Prinzip solche, die sich eindeutig auf Elemente des Datenbankschemas abbilden lassen. Um dies zu erreichen, wird in **PRECISE** das sprachliche Inventar (das Lexikon) weitgehend automatisch aus den Datenbankeinträgen (d.h., den

⁶ d.h., eine Umschreibung der Antwortmenge

Namen der Relationen, Attribute und Werte) extrahiert.⁷ Hierzu gehört auch die Aufnahme von Information über die spezifische Verwendung eines Wortes als Beschränkungen (*restrictions*), z. B. dass 'from' in einer Flugauskunftsdomäne einen Flug mit dem Abflugort assoziiert, 'on' einen mit einer Tagesangabe. Das Lexikon besteht daher in einer Abbildung der Token (auch komplexe Wörter wie 'New York') auf Datenbankinformation, wie in (5).

(5) Lexikoneintrag für 'from'
from(flight f1, city c1):
 f1.fromAirport=
 airportService.airportCode
 AND airportService.cityCode=
 c1.cityCode

Eine Eingabe in PRECISE (die Frage) wird zunächst vollständig *tokenisiert*. Tokenisierung involviert z. B. die Zuordnung einer Phrase *price of breakfast* zu dem Token *breakfast price* (das auf ein Datenbankattribut Breakfast.Price verweist). Bei vollständiger Tokenisierung muss jedes Wort der Frage genau einem Token zugewiesen sein.

Ein allgemein verwendbarer⁸ *statistischer Parser* analysiert die Struktur der Frage. Bei syntaktischen Ambiguitäten können die *restrictions* aus dem Lexikon schon frühzeitig inkorrekte Lesarten ausschließen. So führt lexikalisch-semantische Information dazu, dass die Phrase „on monday“ in der Frage „What are the flights from Boston to Chicago on monday?“ nicht an „Chicago“, sondern an die „flight“-NP angehängt (*attacht*) wird.

Ein *matcher* erfüllt die Aufgabe, eine gültige Interpretation einer Frage zu finden. Er muss eine eindeutige Abbildung der token auf Datenbankelemente herstellen, wobei u.a. die *restrictions* und die *attachment*-Informationen berücksichtigt werden.⁹ Die semantische logische Form einer gültigen Interpretation wird von einem *Anfragegenerator* in eine Datenbankabfrage umgewandelt.

Semantisch behandelbare Fragen charakterisieren somit eine semiautomatisch definierte kontrollierte Sprache, die die Performanz des Systems garantiert, welche, wie Popescu et al. (2004) anhand einer experimentellen Evaluation mit ATIS¹⁰-Fragen zeigen, zu den besten der verfügbaren Systeme gehört. Während PRECISE lokale Phänomene (lex. Ambiguität, Wort-Ellipsen, lokale Attachment-Ambiguitäten) behandeln kann, sollte allerdings klar sein, dass es an vielen anderen Phänomenen (die bei der Evaluation ausgeschlossen wurden) scheitert.

6.5 Ausblick

Es gibt eine Reihe von Aspekten, die den Bereich einfacher Fragebeantwortung verlassen und komplexere Methoden erfordern (s. Androutsopoulos und Ritchie 2000):

⁷ Ausnahmen sind manuell hinzugefügte Synonyme sowie Präpositionen etc.

⁸ der nur durch ein einfaches Re-Training mit domänspezifischen Fragen angepasst werden muss

⁹ Diese Aufgabe wurde als spezifisches Graphensuche- (*MaxFlow*-) Verfahren implementiert.

¹⁰ Air Travel Information System

Inferenzen

Nicht alle erfragbare Information ist explizit in der Datenbank gespeichert, dafür aber vielleicht aus vorhandener Information *erschließbar*. Beispiele sind Fragen wie *Which patients need attention overnight?*. Hier könnten allgemeine Regeln spezifizieren, wann ein Patient besondere Aufsicht erfordert. Anhand solcher Inferenzregeln und der gegebenen Patientendaten kann die Antwortmenge dann erschlossen werden.

DB updates

Eine Erweiterung rein abfrageorientierter NLIDB besteht darin, auch Assertionen zuzulassen bzw. die natürliche Sprache in NLIDB zu einer Kommandosprache zu erweitern (s. z. B. Minock 2006). Ein Beispiel ist das explizite Zuweisen eines neuen Wertes an ein Attribut über eine Aussage:

- > *What is the home port of the Tokyo Maru?*
Yokohama
- > *Home port of the Tokyo Maru is Hong Kong.*
Yokohama has been replaced by Hong Kong as home port of Tokyo Maru.

Weitere Fragetypen

Die Struktur der in der DB enthaltenen Information bietet Ansatzpunkte für sogenannte **Meta-Fragen** wie *What are the properties of employees?*. In ähnlicher Weise fragen **Modale Fragen** nach Information, die sich durch Ausnutzen von Integritätsbeschränkungen der DB beantworten lassen: *Can an employee be 18 years old?* Eine Behandlung **temporaler Fragen** erlaubt es, auch indirekt bzw. relativ zu markanten temporalen Referenzpunkten die spezifischen Zeitwerte in der DB zu adressieren: *Was Smith employee of Test Ltd last year/during summer 2003/...?*. Eine Behandlung temporaler und aspektueller Phänomene in natürlichsprachlichen Anfragen an Datenbanken findet sich in Androustopoulos et al. (1998).

NLIDB im Web

Angesichts des Stands der Technik im Information Retrieval und den Bemühungen im Bereich domänenübergreifender Fragebeantwortung über unstrukturierten Daten (*open domain question answering*, s. Kap. 7) stellt sich die Frage nach möglichen Anwendungsgebieten im Bereich **Web-zugänglicher domänenbeschränkter NLIDB, die auf strukturierten Daten operieren**.

Nach Minock (2005b) existieren auch heutzutage Anwendungsmöglichkeiten als Pendant zu Formular-basierten Schnittstellen in Bereichen wie Geografiedaten-Zugriff, Fahrplan-/Reise-/Event-Auskunft etc. Voraussetzung für den Erfolg solcher Systeme wäre, dass deren beschränkte Domäne klar umschrieben, komplex und praktisch relevant ist und dass sich der System(erstellungs-/wartungs)aufwand und das Benutzerinteresse die Waage halten. Minock liefert einige Beispiele für aktuelle Web-NLIDB.

6.6 Literatur

Androutsopoulos et al. (1995) ist eine detaillierte Einführung in NLIDB. In *Androutsopoulos und Ritchie* (2000) wird eine elaboriertere Architektur moderner NLIDB präsentiert und diskutiert. Ein etwas älterer, aber gut lesbarer Überblick über NLIDB ist *Copestake und Spärck Jones* (1990). Einen sehr guten Überblick mit vielen praktischen Beispielen bietet *Biveroni* (2002).

Kapitel 7

Frage-/Antwort-Systeme (QAS)

Deckard: She's a replicant, isn't she?
Tyrell: I'm impressed. How many questions does it usually take to spot them?

(aus: *Blade Runner (Film)*)

Von Ödipus bis zur Kandidatin bei Günter Jauch als repräsentativem Mitglied der heutigen Wissensgesellschaft: alle wären sicher dankbar (gewesen) über ein Gerät, dass auf Knopfdruck konzise Antworten auf die ihm gestellten Fragen liefert – lange Zeit eine Utopie, nun immerhin work in progress ...

Ein Frage-/Antwort-System (**Question Answering System, QAS**) beantwortet einfache natürlichsprachliche Fragen relativ zu einer Wissensbasis/Quelltextsammlung und **produziert eine angemessene** (im Kontext adäquate, informative) **natürlichsprachliche Antwort**.

Es kann ggf. in einen spezifischen Klärungsdialog mit dem Benutzer eintreten (bzgl. der Frage oder der Antwort), allerdings hat ein QAS keine allgemeine Kompetenz, Dialoge zu führen (s. *Dialogsysteme*, Kapitel 8).

Das Kennzeichnende an einem QAS ist seine Fähigkeit, eine **Antwort** auf eine Anfrage zu geben (und nicht nur Suchergebnisse zu präsentieren). Dazu sammelt es relevante Information von den Quellen (die (semi-)strukturiert oder unstrukturiert sein können, d.h. von einer Datenbankanfrage bis zu einer Google-Suche), verschmilzt diese Information (bzw. fasst sie zusammen) und sammelt außerdem Information über den epistemischen Status der Antwort (Verlässlichkeit, Aktualität, Faktizität etc.).

Der praktische Wert in Sphinx-losen Zeiten: auch wenn ggf. eine längere Bearbeitungszeit der Anfrage zu erwarten ist, so entfällt das mühsame Sichten der Suchergebnisse – effektiv also eine Zeitersparnis.

7.1 Allgemeines zu QAS

Die wichtigsten **Anforderungen**, die ein QAS idealerweise erfüllen soll, sind:

- **Gutes Zeitverhalten:**
Zum Einen das Liefern einer schnellen Antwort, zum Anderen Aktualität (d.h. die schnelle Einbindung neuer Information).
- **Genauigkeit/Korrektheit:**
Die Antworten sind korrekt. Widersprüche können in den Daten entdeckt und angezeigt werden. Die Antworten sind unter Rückgriff auf umfangreiches Weltwissen so genau wie für den Zweck angemessen.
- **Benutzeradaptivität:**
Das QAS passt sich an den Wissensstand, das spezielle Interesse und die Wünsche des Benutzers an und präsentiert die Antwort im gewünschten Ausgabeformat (Verwendung eines Benutzermodells). Hierzu gehören die Multimedialität von Ein- und Ausgabe (Text, Grafik, Video, Audio) sowie die Anpassung an die jeweilige Sprache (Multilingualität). Es erlaubt außerdem die interaktive Bestimmung bzw. Einschränkung des Fragekontextes und des Antwortraums.
- **Vollständigkeit:**
Zusammensetzung der Antwort aus verschiedenen Quellen (*answer fusion*) Inferenzen, Verbinden von domänen spezifischem mit allgemeinem Wissen
- **Relevanz:** Möglichkeit zur Klärung geben

7.2 Geschichte

Bereits Simmons (*Simmons 1965*) nennt nicht weniger als 15 QAS, darunter **BASEBALL** (System zur Beantwortung von Fragen wie *Who did the Red Sox lose to on July 5th?*, s. *Green et al. 1963* und den Exkurs auf Seite 76) und **SAD-SAM** (System zur Abfrage und zum systematischen Aufbau von Familienbeziehungen, s. *Lindsay 1963*).

Auf diese ersten experimentellen, zum Teil sehr oberflächlichen QAS (wie z. B. **ELIZA**) folgte eine Generation **wissensbasierter Frage-Antwort-Systeme**, in denen verschiedene, an der Fragebeantwortung beteiligte Wissensaspekte systematisch repräsentiert und mit zunehmend allgemeineren Verfahren verwendet wurden. Meilensteine hiervon sind

- Dialogmodellierung in der KI, **SHRDLU** von Terry Winograd (*Winograd 1972*)
- Natürlichsprachliche Abfrage von strukturierten Datenbanken, **LUNAR** von Woods (*Woods et al. 1972*) (*What is the average concentration of aluminum in high alkali rocks?*)
- Psychologie des Textverständens (story comprehension), **QUALM** von Wendy Lehnert (*Lehnert 1978*), Anwendung der Schank'schen Scripts für die Fragebeantwortung in sehr eingeschränkter Domäne

- Berkeley Unix Consultant (Fragen und Antworten zum UNIX-Betriebssystem, s. Wilensky et al. 1994)

Die Nachteile wissensbasierter QAS sind bekannt: Sie untersuchten zwar wichtige Aspekte von Fragen und Antworten, ihre Anwendung(sdomäne) war jedoch sehr eingeschränkt; außerdem wiesen sie nicht selten erhebliche Mängel –u.a. im Hinblick auf Modularität, Transparenz und Skalierbarkeit– auf.

Ein zentrales Problem war und ist insbesondere die Beschränktheit ihrer Wissensbasis, z. B. bei Expertensystemen. Sie führt dazu, dass die Funktionalität des Systems –in der Regel unerwartet von und meistens nicht nachvollziehbar für den Benutzer– erheblich beschnitten ist. Beispiele dafür sind das Nicherkennen falscher Eingaben, das Liefern falscher Antworten oder sogar das völlige Systemversagen aufgrund fehlenden Hintergrundwissens.

Ein Lösungsversuch dieses Problems war der explizite Aufbau einer enzyklopädischen Wissensbasis im Rahmen des Projekts **CYC** (Lenat und Guha 1989). Dieser Versuch ist mindestens unabgeschlossen, wenn nicht gescheitert. Als Alternative wurden die Anstrengungen zur (Standardisierung der) Erstellung von Ontologien verstärkt, auf deren Grundlage der Aufbau enzyklopädischer Wissensbasen systematisch möglich sein soll.

Eine andere Reaktion auf das Versagen wissensbasierter QAS ist die Entwicklung Dokument- bzw. **Text-basierter Frage-Antwort-Systeme**, die das Gros aktueller QAS ausmachen. Text-basierte QAS sind eine konsequente Erweiterung des IR in Richtung Fragebeantwortung und Dialog.

Solche Text-basierten QAS weisen –anders als Expertensysteme– keine Beschränkung bzgl. des Gegenstandsbereichs von Fragen auf (sog. *open domain question answering*). Sie basieren auf der Annahme, dass die Antwort auf eine Frage schon irgendwo mindestens einmal stehen wird. Dies bewährheit sich sehr oft und ist relativ einfach bei Faktfragen. Allerdings treten schon hier Probleme auf, die aus dem Fehlen einer gepflegten Wissensbasis resultieren, nämlich z. B. bei Inkorrektheit, Inkonsistenz oder Inaktualität der berücksichtigten Texte. Text-basiertes QA wird zunehmend schwieriger, wenn die Antwort zusammengesetzt oder erschlossen werden muss.

Ein wesentlicher Fortschritt in der Entwicklung von QAS wurde durch die Etablierung von Evaluationskonferenzen, in denen Fragebeantwortung systematisch als spezifisches Textretrieval untersucht wurde (*Text Retrieval Conferences, TREC*, s.u.).

7.3 Aspekte und Anforderungen

7.3.1 Allgemeines

QAS unterscheiden sich vor allem durch die **Art und Nutzung der zugrunde liegenden Ressourcen**. Dies sind

- **die Struktur der Datenquelle:**
Strukturierte Daten (Einträge einer Datenbank), Semi-strukturierte Daten (z. B.

annotierter Text) oder reiner („unstrukturierter“) Text (z. B. Zeitungsartikel)

- **die Art der Datenquelle:**

Vorgegebene komplexe Textsammlung (wie bei der TREC), gesamtes Internet, einfache Textsammlung (Enzyklopädie, Buch) oder einzelner Artikel

- **die Eigenschaften der Datenquelle:**

z. B. ihre Größe, Dynamizität (Wie aktuell ist sie? Wie oft wird sie upgedated?), Qualität

- **die Domäne:**

Domänen-spezifisches System (z. B. Hilfe-System), allgemeines Anfragesystem, oder verschiedene Modalitäten (z. B. Bild- oder Tondatenbanken)

Die **Kompetenz** von QAS lässt sich generell anhand der folgenden Fähigkeiten messen, die sie besitzen:

- **Faktenfragen beantworten können:**

z. B. *Wer war der erste Mann auf dem Mond?* Nach Antworten wird direkt (textueller Match eines aus der Frage gewonnenen Suchmusters) gesucht.

- **Einfache Schlüsse ziehen können:**

Beziehung zwischen Frage und gefundener Antwort muss erschlossen werden (kein direkter „match“ von Frage und Antwort).

- **Die Antwort aus mehreren Dokumenten zusammenstellen können**

(answer fusion)

- **Spekulative Fragen durch Vergleiche beantworten können:**

z. B. *Wird Merkel eine gute Kanzlerin sein?*

- **Mit dem Benutzer interagieren können:**

Klärende Rückfragen ermöglichen

Spezifische Unterschiede zwischen QAS ergeben sich daraus, welche Fragen und Antworten möglich und zugelassen sind. Bei **Fragen** ergeben sich Differenzierungen je nach

- **Antworttyp:**

Fakten (kurze Phrasen, Sätze, größere Text), Meinungen („Was sagst du dazu?“) oder Zusammenfassungen („Wie geht das?“)

- **FrageTyp:**

Ja/Nein-Fragen (Entscheidungsfragen), W-Fragen (Ergänzungsfragen), indirekte Anfragen („Ich würde gerne wissen...“) und Aufforderungen („Nenne die letzten drei Bundeskanzler...“)

- **Schwierigkeitsgrad der Frage:**

z. B. *Wer ist der Bundeskanzler der Bundesrepublik Deutschland?* (einfach) vs. *Warum will Bundeskanzler Schröder die Agenda 2010 einführen?* (schwierig)

Fragenklassifikation nach Lehnert (1978) und Graesser et al. (1992)

Verifikation (Ist eine Aussage wahr?):

Ist Moussaka ein afrikanischer Tanz?

Vergleich (Wie ähnlich ist X zu Y?):

In welcher Weise ist Florida ähnlich zu China?

Disjunktion (Ist X oder Y der Fall?):

Erhöhen oder vermindern die Berge den Regen in Oregon?

Konzeptvervollständigung (Wer? Was? Wann? Wo?):

Wo ist die größte Bevölkerungsdichte in Europa?

Definition (Was bedeutet X?):

What is a factorial design?

Beispiel (What is an example of X?):

What is an example of an ordinal scale?

Interpretation (How is a particular event interpreted or summarized?):

Does the graph show a main effect for „A“?, What happened yesterday?

Merkmalspezifikation (Welche qualitativen Eigenschaften hat X?): *What is George like?*

Quantifizierung (Was ist der Wert einer quantitativen Variable?):

How many rooms are in the house?

Verursachung (Was verursacht(e) das Eintreten eines Ereignisses?):

How does warm air get to Ireland?

Folge (Was sind die Konsequenzen eines Ereignisses/Zustands?):

What happens to the warm winds when they reach the mountains?

Zielgerichtetheit (Was sind die Motive hinter den Aktionen eines Agenten?):

Why did Roger move to Chicago?

Befähigung (Welches Objekt/Mittel befähigt einen Agenten, eine Aktion durchzuführen?):

What device allows you to measure an earthquake?

Instrument/Durchführung (Wie erreicht ein Agent ein bestimmtes Ziel?):

How does a person perform long division?

Erwartung (Warum trat ein erwartetes Ereignis nicht ein?):

Why doesn't this doll have a mouth?

Beurteilung (Der Fragende will, dass der Antwortende eine Idee beurteilt oder einen Rat gibt, was zu tun ist.): *What do you think about the new taxes?*

Assertion (Der Sprecher drückt aus, dass er oder sie irgendeine Information vermisst.):

I don't understand what this message on the computer means.

Aufforderung/Anweisung (Der Sprecher fordert den Hörer direkt auf, ihm eine bestimmte Information zu geben.):

Please tell me how to get a printout of this file.

Antwortarten (frei nach Webber 1992)

Allgemeine Unterscheidung

- Antwort (i.e.S.) d.h. die angefragte Information/Aktion ,
- Erwiderung (response) enthält
 - Antwort+Zusätzliche relevante Information (oder Aktionen)
 - Information/Aktionen anstelle der Antwort

Beispiel (*Welche Zensur habe ich im Test?*)

Antwort (i.e.S.): *sehr gut*

Zusätzliche relevante Information (oder Aktionen):

- A glaubt, dass F die I. gebrauchen kann:
du findest die Liste an der Tür
- A nimmt an, dass F. einen Plan hat, der nicht erfüllt werden kann:
du findest die Liste aber nicht an der Tür
- A glaubt, dass die I. die Antwort rechtfertigt/erklärt, so dass F. sie akzeptiert:
ich checke die Ergebnisse immer noch einmal durch
- Klärung der /Evidenz für die Antwort:
Hier schau, eine 1.0

Information/Aktionen anstelle der Antwort: *sage ich dir nicht*

Zusätzliche relevante Information/Aktionen für den Antwortersatz:

- Korrektur einer Fehlannahme, dass A. eine Antwort geben kann (deontisch):
darf ich dir nicht sagen
- Korrektur einer Fehlannahme, dass A. eine Antwort geben kann (modal):
kann ich dir nicht sagen (hab die Note nicht im Kopf)
- Korrektur einer Fehlannahme, dass A. eine Antwort geben kann (faktisch):
die Ergebnisse gibt's erst nächste Woche (Tests noch nicht korrigiert)
- Korrektur einer Fehlannahme, dass A. eine Antwort geben will (intentional):
die Ergebnisse (sind bekannt, aber) werden erst nächste Woche bekannt gegeben
- Korrektur einer Fehlannahme, dass A. eine Antwort geben will (attitudinal):
keine Lust
- Korrektur einer mögl. Fehlannahme bzgl. Fehlen der Antwort:
du bist jedenfalls nicht durchgefallen
- Ablehnung der Frage:
was fällt dir ein, mich das zu fragen?...

Für die **Antworten** gelten die folgenden Unterscheidungsmöglichkeiten:

- **Antworteigenschaften:**

Die Antwort sollte möglichst korrekt, exakt, nützlich und nicht-fehlleitend sein. Je nach Anforderungen des Benutzers sollten Detailliertheitsgrad (kurz und prägnant vs. ausführlich) und Ausgabeform (ausformulierte Sprache vs. Liste) wählbar sein.

- **Erzeugung der Antworten:**

Kann durch „cut and paste“ oder durch Generierung der Antwort (→ Sprachgenerierung, automatische Textzusammenfassung) umgesetzt werden.

Im Hinblick auf die **Interaktion mit dem Benutzer** sind die folgenden Aspekte relevant:

- **Eingehen auf die Expertise des Benutzers:**

Für allgemeine Kompetenzunterschiede wie den zwischen Laien und Experten (so wie z. B. : Alter, Sprachkenntnisse) werden *generische/stereotype Benutzermodelle* benötigt.

- **Eingehen auf die Vorlieben und Absichten des Benutzers:**

Ein QAS kann und sollte unterschiedliche Strategien bei der Suche und unterschiedliche Arten der Ergebnispräsentation offerieren, beispielsweise die Möglichkeit, Fragen sukzessive zu verfeinern und Suchergebnisse einzuengen. Dabei sollte es die jeweiligen Absichten des Benutzers berücksichtigen (Kontext/Zweck einer Frage, beruflicher/privater Einsatz). Entsprechend werden spezifische/individuelle Benutzermodelle dafür benötigt, die jeweiligen Präferenzen des Benutzers zu verwalten.

- **Dialogische Fähigkeiten:**

Einige QAS weisen eingeschränkte Fähigkeiten zur dialogischen Interaktion mit dem Benutzer auf. Ideal ist selbstverständlich ein QAS als Dialogsystem mit Textzusammenfassungskompetenz (für die Kondensierung der gefundenen Antworttexte).

7.3.2 Open domain QA

Im Infomation Retrieval (IR) existieren mehrere **Arten der Informationserschließung** zu einer Anfrage:

- **Document retrieval:** erschließt *Dokumente*, die relevant sind
- **Passage retrieval:** erschließt *Abschnitte*, die relevant sind
- **Answer extraction:** erschließt den *genauen Text*, der die Frage beantwortet

Text-/Dokument-basiertes QA ist eine computerlinguistisch orientierte Entwicklung des IR, die diese Verfahren beinhaltet und in verschiedenen Hinsichten verfeinert. Dabei werden auch Methoden der IE und des klassischen QA verwendet.

Für **open domain QA** gilt, dass weder die Art der zugrundeliegenden Wissensquellen noch die Inhalte der vom Benutzer gestellten Fragen beschränkt sind. Beliebige Fragen sollen also

zu „offenen“ Informationsquellen (heute in der Regel das Web) beantwortbar sein, wobei die Aufgabe des Text-basierten QA darin besteht, aus der großen Menge möglicher Antworten die beste oder die wenigen besten auszuwählen (sog. *answer-pinpointing*).

Der Vorteil **Web-basierter QAS** ist der hohe Grad an *Vollständigkeit* (auf eine Frage wird es fast immer irgendwo eine Antwort geben) und *Aktualität* (ein Problem manuell gepflegter Datenbasen). Ihr Nachteil besteht in dem Problem der *Validierung* (falsche/fehlerhafte Informationen sind nicht (immer) als solche erkennbar) und der *Überlagerung bzw. zeitlicher Zuordnung* von Information (Altes wird durch Neues nicht unbedingt ersetzt, sondern existiert weiter).

WebQAS sind in der Regel auf den Service verfügbarer Suchmaschinen angewiesen. Sie müssen sich folgenden **Herausforderungen** stellen:

- **Bildung der richtigen Querys** an die Suchmaschine
Querys dürfen nicht zu spezifisch sein (dies ergibt keine Resultate), andererseits aber auch nicht zu unspezifisch (zu viele Resultate).
- **Rauschen** (durch unerwünschte Treffer)
Schlüsselwortsuchen treffen auf alle möglichen Seiten im Web, da die Semantik der gewünschten Frage nicht übermittelt werden kann. Z.B. führt die Query „first american in space“ auch zu Treffern wie „first american voting in space“
- **Faktizität** (→ falsche Treffer erkennen)
Es ist nicht-trivial, den Wahrheitswert einer Aussage auf einer Webseite zu erkennen. Z.B. findet sich die Aussage „John Glenn was the first american in space“ auf einer Seite über „common misconceptions in astronomy“. Aber schon die Verwendung von Negation („was not ...“) oder anders ausgedrückter Verneinung („I thought that... But I was wrong.“) stellt Probleme für heuristische Herangehensweisen.
- **Ressourcenbeschränkungen**
Es ist in der Regel sinnvoll und notwendig, für eine Frage mehrere Suchmaschinen-Querys zu generieren. Dies ist „teuer“ (an Systemaufwand und Zeit), außerdem stellt der Benutzer in der Regel nur beschränkte Geduld-Ressourcen zur Verfügung.

Das **generelle Vorgehen** beim open domain QA ist das folgende:

- Frageanalyse und Bestimmung des Antworttyps
- Vorauswahl an geeigneten Antwortpassagen (die die Antwort enthalten können)
- Verarbeitung der Passagen, um Antworten des richtigen Typs zu finden
- Ordnen der Antwortkandidaten und Ausgabe der wahrscheinlichsten Antwort

Der „Klassiker“ des open domain QA ist das System **MURAX** (Kupiec 1993), das die Grolier's online encyclopedia als Textsammlung verwendete und Fragen aus dem „Trivial Pursuit“-Spiel zuließ. Letztere waren jedoch stark eingeschränkt: Es konnten nur „closed-class questions“ gestellt werden (solche, die mit einer NP beantwortet werden), wie z. B. *Wer schrieb „Hamlet“?*. Der Klassiker Web-basierter QAS ist das System **START** ([Link1](#),[Link2](#)) .

Neben diesem ersten WebQAS (seit 1993) sind heute eine Reihe von Systemen als Online-QAS im Internet zugänglich (oder gerade in Entwicklung). Beispiele dafür sind

- <http://www.answerbus.com>
- <http://www.answers.com>
- <http://www.cognitionsearch.com>
- <http://www.languagecomputer.com>
- <http://www.ask.com> (früher: Ask Jeeves)
- <http://www.hakia.com>
- <http://www.teragram.com>
- <http://www.bing.com> (enthält Powerset)
- <http://www.wolframalpha.com> (seit Mai 2009)

Allerdings wären diese Systeme vielleicht eher Einzelfälle gewesen (oder sogar nicht existent), wenn das gesamte Paradigma Text/open domain/Web-basierten QAs nicht einen eminenten Auftrieb durch die Aktivitäten im Rahmen des QA-Zweigs („QA tracks“) der Text Retrieval Conferences (*TREC*) erhalten hätte.

7.4 Architekturen und Systeme

7.4.1 WebQAS

Die generische Architektur eines WebQAS weist die in Abb. 7.1 dargestellten drei zentralen Prozesskomponenten *Frageverarbeitung*, *Passagenerschließung* und *Antwortextraktion* auf (s. Surdeanu und Pasca 2002), die im Folgenden kurz beschrieben werden.

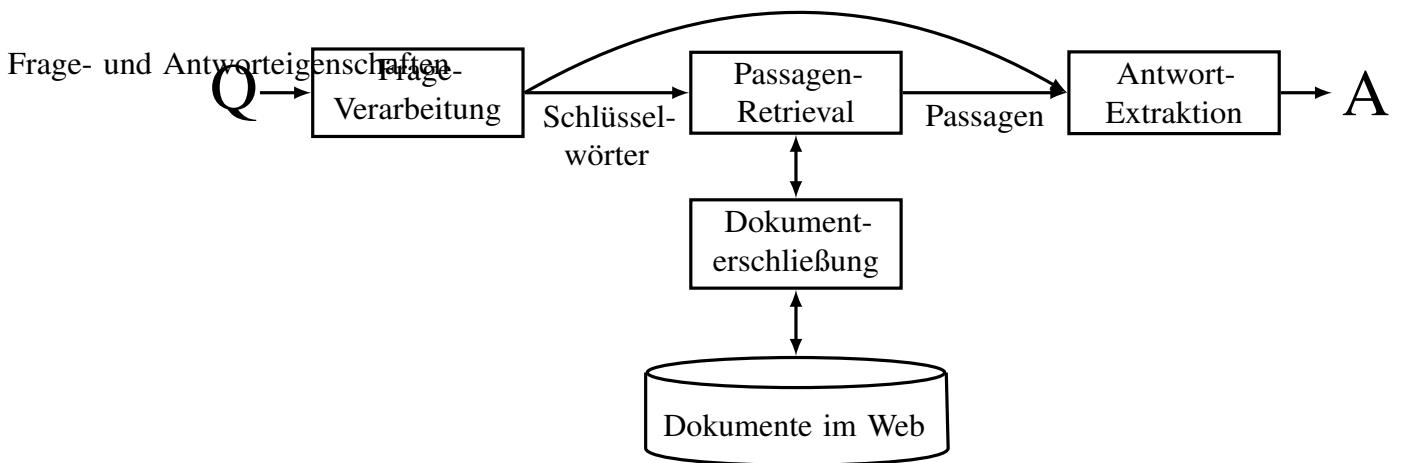


Abbildung 7.1: Generisches WebQAS

Frageverarbeitung

Fragen werden in der Regel geparst und es werden charakteristische Informationen extrahiert. Hierzu gehören: der **Typ der Frage** (abhängig von den Fragewörtern: „what“, „when“, „how“ etc.), der **Typ der gewünschten Antwort** (syntaktische Kategorien wie „NP“, semantische Kategorien wie „PERSON“ für *Who is John Doe?*, „ORT“ für *Where is Azkaban?* etc.) und zum Teil der **Fokus der Frage** (in der Frage zusammen mit einem Fragewort genannte Instanz des Antworttyps, z. B. „country“ in *which country*). Eine wichtige Rolle spielt an dieser Stelle die Erkennung von Namen (*named entity recognition (NER)*). Je nach Grad der sophistication des Systems wird ggf. auch die logische Semantik der Frage eruiert.

Weiterhin werden aus der Frage **Schlüsselwörter** gewählt bzw. gebildet, die für das IR nach relevanten Passagen (die *query*) verwendet werden können. Eine essentielle Methode zur Verbesserung der Trefferwahrscheinlichkeit ist das Hinzufügen semantisch verwandter Wörter mit Hilfe von Wordnet (*query expansion*).

Passagenerschließung

Mit den gewählten Schlüsselwörtern werden –in der Regel mehrere– Querys gebildet und an die Suchmaschine gesendet (wichtig: die Verwendbarkeit Boole'scher Verfahren). Aus den gelieferten Dokumenten werden die geeigneten Passagen ausgewählt und der Antwortextraktion übergeben. Die Art der Passagen richtet sich entweder nach der Struktur des Textes, oder es werden Fenster bestimmter Größe über dem Dokument definiert. Für die Bewertung (das *ranking*) der gelieferten Passagen werden Aspekte der Frage relativ zu den Schlüsselwort-Matches ausgenutzt, z. B. inwieweit sich die Reihenfolge der Wörter in der Frage wiederfindet, wieviele Schlüsselwörter enthalten sind und ob sich Frage-Wort und Match in der Groß-/Kleinschreibung unterscheiden.

Antwortextraktion

Die gelieferten Passagen sind immer noch zu groß, um als Antwort im engeren Sinn gelten zu können. Außerdem sind sie von unterschiedlicher Qualität. Entsprechend müssen die Antworten extrahiert und geordnet werden. Hierfür existieren unterschiedliche Verfahren (ein Kontinuum von flach zu wissensbasiert): eine eingeschränktere, feinere Version Schlüsselwort-basierter Passagenfilterung (System **LASSO**); ein Matching von Antworttyp-spezifischen *Mustern* mit den Passagen (System **WEBCLOPEDIA**); ein Matching von Fragensemantik (semantische Repräsentation der erwarteten Antwort) und Antwortkandidat (System **FALCON**); eine Bewertung der Antwort danach, wie gut sie die Frage begründet (per logisch-abduktivem Verfahren) (**FALCON**). In der Regel werden die Antwortkandidaten geparst, zumindest um eine Eigennamenerkennung zu gewährleisten.

LASSO: Query-Bildung mit Schlüsselwörtern

Die Effektivität des Systems **LASSO** (Sieger des TREC-8 QA tracks) basiert wesentlich auf einer ausgeklügelten Schlüsselwort-basierten Query-Bildung. Ausgangspunkt ist die Verwendung Boole'scher Suche (im Gegensatz zu Vektorraum-Verfahren), die „entweder-oder“-Anfragen erlauben. Dies ermöglicht eine Feineinstellung der Querys bzw. deren Justierung mit Hilfe

von Heuristiken. Außerdem wird nicht eine „One-shot“-Querybildung verwendet, sondern ein „retrieval loop“, nach dem Art und Anzahl der Schlüsselwörter so lange modifiziert werden, bis die Passagenqualität zufriedenstellend ist. In **LASSO** werden die folgenden Heuristiken für die Aufnahme von Wörtern in die Query verwendet:

- 1 Alle Nicht-Stopwörter in Zitaten/Anführungen
- 2 Alle Nomina von erkannten named entities
- 3 Alle komplexen Nominale mit ihren adjektivischen Modifikatoren
- 4 Alle anderen komplexen Nominale
- 5 Alle Nomina mit ihren adjektivischen Modifikatoren
- 6 Alle anderen Nomina
- 7 Alle Verben der Frage
- 8 Der Fragefokus (der in den vorherigen Schritten übersprungen wurde)

Anfänglich werden die ersten sechs Heuristiken (in der angeführten Reihenfolge) verwendet. Ggf. werden weitere Schlüsselwörter gemäß den Heuristiken 7+8 hinzugefügt. Sollte eine Query zu spezifisch geraten sein, so werden Schlüsselwörter in umgekehrter Reihenfolge der Heuristiken fallengelassen. Dies ergibt beispielsweise die folgenden Zuordnungen von Frage und Schlüsselwörtern:

- What researcher discovered the vaccine against Hepatitis-B?
Hepatitis-B, vaccine, discover, researcher
- What is the name of the French oceanographer who owned Calypso?
Calypso, French, own, oceanographer
- What U.S. government agency registers trademarks?
U.S., government, trademarks, register, agency
- What is the capital of Kosovo?
Kosovo, capital

MULDER: Linguistisch motivierte Query-Bildung

Das System **MULDER** ist ein Beispiel für den Einsatz linguistisch orientierter Verfahren bei der Bildung von Querys für die Passagenerschließung. Hierbei wird die Idee verfolgt, aus der Frage mögliche Formen der Antwort zu generieren, u.a. auf die folgende Weise:

- **Verbkonversion:**
Umformung der Frage mit flektiertem Hilfsverb und Partizip Perfekt Passiv-Form des Verbs in ein einfaches flektiertes Verb nach dem Schema
do_[MERKMALE]..Verb_{Infinitiv} → Verb_[MERKMALE]. Dies ergibt die Transformation:
When did Nixon visit China? → Nixon visited China...

- **Query-Erweiterung:**
Finden von Eigenschaftsnomen für ein Adjektiv (→ Wordnet):
How tall is the Feldberg? → the height of the Feldberg is
- **NP-Bildung:** Quotierung von NPs/NEs: '*Peter Müller*'
- **Transformation:**
Übernahme von Transformationsregeln aus der schon recht betagten generativen Transformationsgrammatik, z. B.
Subject-Aux-Movement: *Who was the... → 1. was the... 2. the ... was*
Subject-Verb-Movement: *Who shot jfk? → shot jfk*

Die Verwendung solcher Techniken führt zu einem Vorsprung gegenüber Systemen, die ausschließlich Schlüsselwörter verwenden. Dies zeigt ein Vergleich der Performanz **MULDERS** mit der von Google und AskJeeves. Abbildung 7.2 zeigt, dass der Aufwand des Benutzers, die richtige Antwort in den von den Systemen gelieferten Resultaten zu finden, bei **MULDER** am geringsten ist (weniger Benutzaufwand für eine bestimmte Resultsqualität, d.h. den Anteil korrekt beantworteter Fragen); außerdem wird insgesamt eine höhere Qualität erreicht.

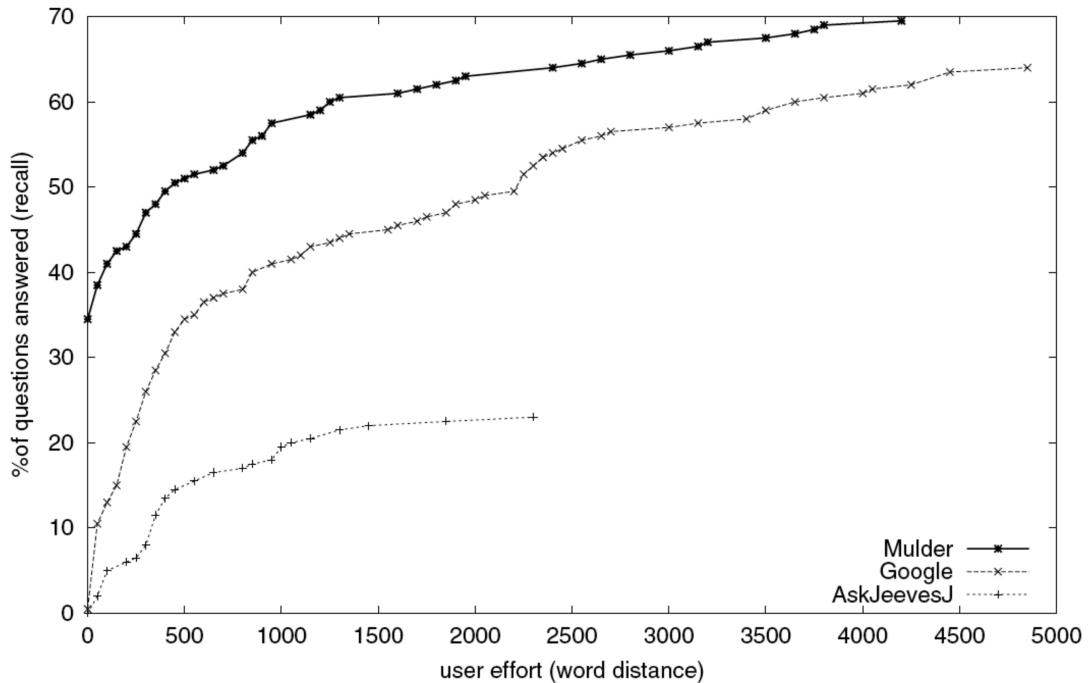


Abbildung 7.2: Performanz **MULDERS** im Vergleich (aus *Kwok et al. 2001*)

WEBCLOPEDIA: Antworttypen

Ein herausragendes Merkmal des Systems **WEBCLOPEDIA** (s. Hovy *et al.* 2001) ist seine umfassende Typologie von Antworttypen und ihre Verwendung für das QA. Antworttypen werden hier **Qtargets** genannt (da sie die Ziele einer Frage charakterisieren). **WEBCLOPEDIA** verfügt über mehr als 180 Qtargets, die aus einer Analyse von mehr als 17000 Fragen gewonnen wurden.

Der Sinn dieser feinen Unterscheidung liegt in ihrer Verwendung: Hovy *et al.* geben ein allgemeines Verfahren an, wie aus Instanzen von Frage- und Antworttypen, die an Suchmaschinen geschickt werden, **Antwortmuster** (und deren Genauigkeit) automatisch gelernt werden können. Mit Hilfe dieser Muster kann eine spezifische Antwortextraktion durchgeführt werden. Das folgende Beispiel zeigt die Zuordnung eines Antworttyps und seiner Extraktionsmuster:

ANTWORT-TYP=INVENTOR (Frage: *Who invented NAME?*):

```
<ANSWER> invents <NAME>
the <NAME> was invented by <ANSWER>
<ANSWER>'s invention of the <NAME>
<ANSWER>'s <NAME> was
<NAME>, invented by <ANSWER>
That <ANSWER>'s <NAME>
```

Wie Ravichandran und Hovy (2002) einräumen, können mit diesem Verfahren keine von der Frage stark abweichende (paraphrasierende) Antworten erfasst werden. Auch zeigt sich schnell, dass oft syntaktische oder semantische Information, d.h. parts-of-speech (POS) oder ontologische Eigenschaften von Lexemen, viele Fehler vermeiden könnten. Ein Beispiel ist das Muster für Ortsangaben (<NAME> in <ANSWER>):

Q: *Where are the Rocky Mountains?*

A: *„Denver's new airport, topped with white fiberglass cones in imitation of the Rocky Mountains **in the background**, continues to lie empty“

Q: *Where are zebras most likely found?*

A: *„in the dictionary“

Während **background** und **dictionary** sich so mit Hilfe sprachlicher Kriterien als mögliche Lokationen disqualifizieren ließen, benötigt man für Beispiele wie das folgende zusätzlich enzyklopädisches oder Welt-Wissen:

Q: *Where do lobsters like to live?*

A: *„on the table“

Eine weitere grundsätzliche Einschränkung Muster-basierter Antwortextraktion zeigt sich bei großen textuellen Distanzen zwischen (<NAME> und <ANSWER>) etwa durch intervenierende Relativsätze wie in

A: „London, which has one of the most busiest airports in the world, lies on the banks of the river Thames“

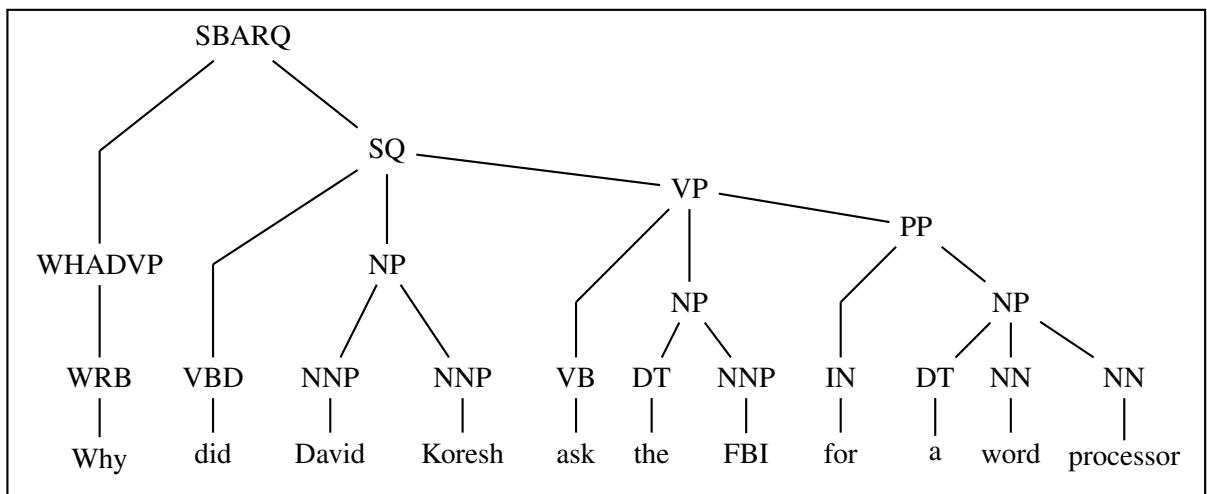
Allerdings kann man hier auf die Redundanz des Webs hoffen. Als Alternative bietet sich eine –wenn auch weitaus aufwändiger– tiefe Analyse (volles Parsing) der Antwortkandidaten an, die dieses Problem en passant erledigt.

FALCON: Integration wissensbasierter Verarbeitung

Das Parsen von Frage und Antwortkandidaten (jeweils in eine semantische Repräsentation) ist eine grundlegende Eigenschaft des Systems FALCON (s. Abb. 7.3).

Frage: *Why did David Koresh ask the FBI for a word processor?*

Parse:



Semantische Repräsentation:

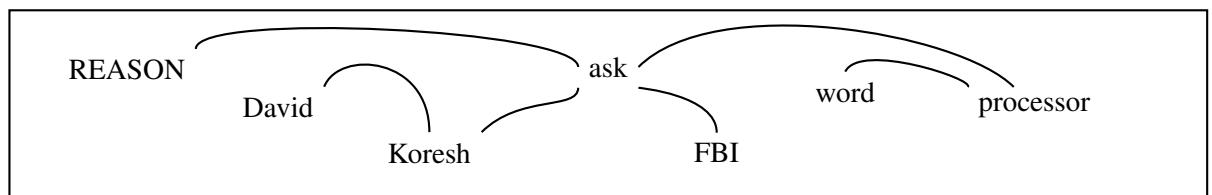


Abbildung 7.3: Parsing eines Satzes in eine semantische Repräsentation (aus: Harabagiu et al. 2000b)

Es fügt dem *retrieval loop* seines Vorgängers LASSO zwei weitere Schleifen hinzu:

- **Morphologische und lexikalische Alternanten:**

Als Bewertung des Ergebnisses der einfachen Schlüsselwortsuche wird eine Unifikation der semantischen Repräsentationen von Frage und Antwortkandidat gefordert.

Ist dies nicht erfüllt, werden weitere/andere Lexeme für die Suche in Betracht gezogen. Hierfür ersetzen Synonyme (z. B. *killer*, *assassin*) oder morphologische Varianten/Derivate (z. B. *invent*, *inventor*) die bisherigen Schlüsselwörter.

- **Semantische Alternanten:**

Als weiterer Filter für Antwortkandidaten wird verlangt, dass ihre logische Form mit derjenigen der Frage unifizierbar ist. Ist dies nicht der Fall, so werden semantisch verwandte Wörter als Schlüsselwörter ausgewählt (z. B. *like* für *prefer*).

Die Erstellung logischer Formen (LF) ist ein einfaches Verfahren, bei dem unter Berücksichtigung syntaktisch-semantischer Eigenschaften Wörter in n-stellige Prädikate überführt werden, z. B.:

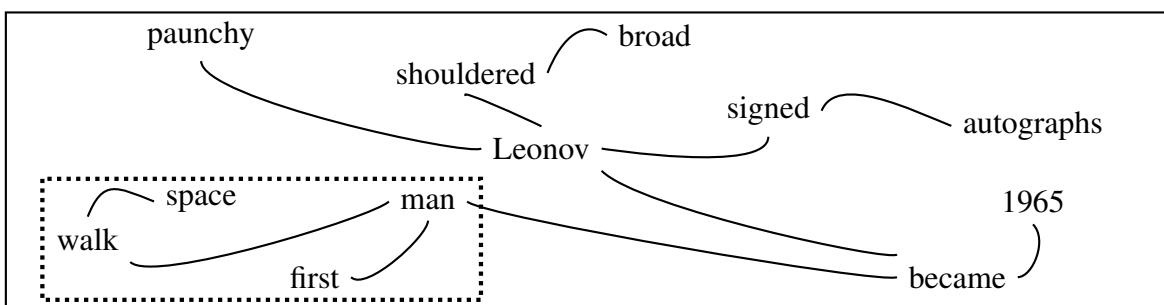
Q: *Who was the first Russian astronaut to walk in space?*

LF: $\text{first}(x) \ \& \ \text{astronaut}(x) \ \& \ \text{Russian}(x) \ \& \ \text{space}(z) \ \& \ \text{walk}(y \ z \ x) \ \& \ \text{HUMAN}(x)$

Auf diese Weise ist es möglich, einen Antwortkandidaten wie den in Abb. 7.4¹ als für die Antwortextraktion valide anzuerkennen.

Antwort: *The broad-shouldered but paunchy Leonov, who in 1965 became the first man to walk in space, signed autographs.*

Semantische Repräsentation der Antwort:



Logische Form der Antwort:

```
paunchy(x) & shouldered(e1, y, x) & broad(y) & Leonov(x) &
first(z) & man(z) & space(t) & walk(e2, t, z) & became(e3, z, u, x)
& 1965(u) & autographs(v) & signed(e4, v, x) & HUMAN(x) & DATE(u)
```

Abbildung 7.4: Bildung einer logischen Repräsentation

¹ Dieses Beispiel, wie auch die leicht abgeänderte Darstellung, entstammt Harabagiu et al. 2000a.

7.4.2 EXTRANS: Wissensbasierte Antwortextraktion

Ein von Web-basiertem QA grundsätzlich verschiedener Ansatz wird im System **EXTRANS** verfolgt. Hier wird nicht nur von einer „tiefen“ Analyse (vollständigem Parsing und mindestens flacher semantischer Analyse) von Frage und Antwortkandidaten, sondern auch von einer solchen Analyse der gesamten zugrunde liegenden Dokumentsammlung ausgegangen (s. Abb. 7.5).

Dies ist selbstverständlich momentan nur für eingeschränkte Anwendungsdomänen machbar, entsprechend ist der Ansatz bisher auf geschlossene Dokumentsammlungen angewendet worden, nämlich zum QA bzgl. der Unix man pages (z. B. *How can I copy a file?*), technischer Dokumente (Airbus-Manuals) und Forschungspapieren im Genombereich.

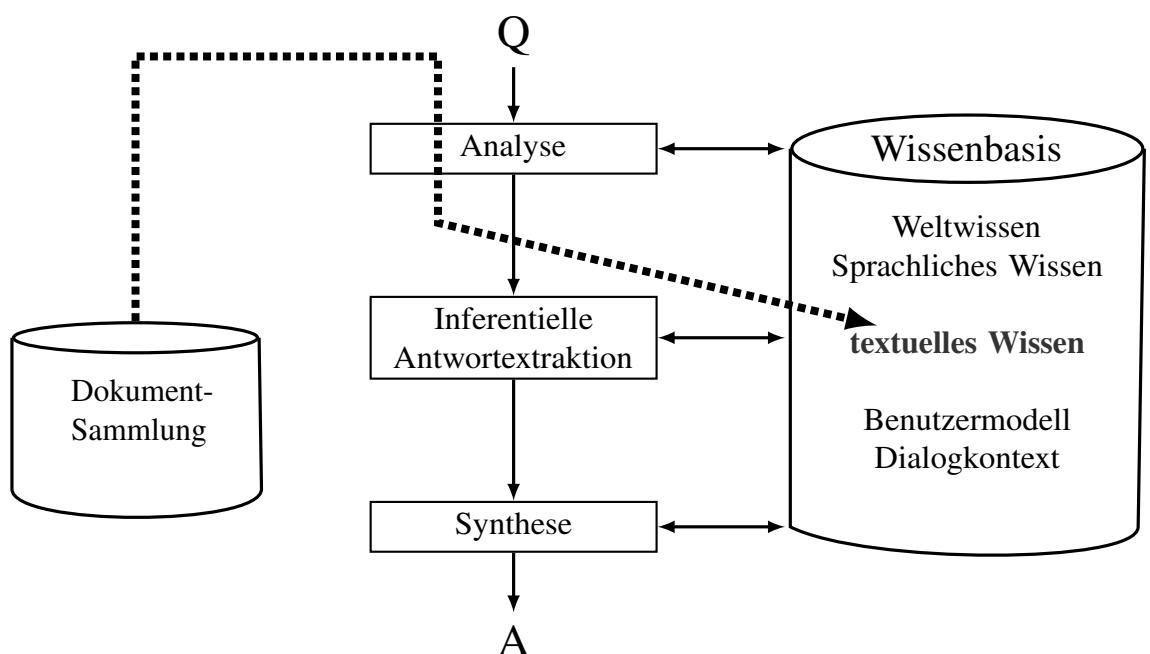


Abbildung 7.5: Architektur eines wissensbasierten QAS

Das Charakteristische an **EXTRANS** ist die Überführung von Dokumenten und Anfrage in eine kanonische semantische Form (*MLF, minimal logical form*) und das Durchführen des notwendigen IR auf semantischer Ebene. Hierzu gehört, dass die semantischen Prädikate intern mit Information über die Textposition, d.h., die Position des geprägten Satzes (DokumentId, AbsatzId, SatzId), markiert sind. Eine Query besteht aus den semantischen Prädikaten der Frage als „bag of semantic relations“ (analog zu dem klassischen „bag of words“-Ansatz des IR) und liefert im Idealfall (wenn eine Antwort vorhanden ist) äußerst präzise Treffer²,

² **EXTRANS** legt somit den Schwerpunkt auf die Präzision der Fragebeantwortung und nicht auf die Allgemeinheit des Verfahrens, ganz im Gegensatz zu Web-basierten QAS.

da zusätzlich zu den verwendeten Wörtern auch ihre Beziehungen zueinander bei der Suche berücksichtigt werden. Ein Beispiel (LF(A) ist die logische Form einer Antwort A; zu einer näheren Erläuterung s. Mollá *et al.* 2003) ist in Abb. 7.6 veranschaulicht.

- A:** In normal flight configuration, each IDG supplies its own distribution network via its Line Contactor (GLC).

LF(A):

```
holds(a1), prop(in, p1, [a1, x4]),
evt(configure, x4, [x3, a5]),
object(configuration, a2, [x4, x3, a5]),
object(flight, a3, [x3]), explication(x4, x6),
object(each, a4, [x6]), prop(normal, p2, [x4]),
evt(supply, e8, _), object(idg, a6, [x7]),
object(network, a7, [x12]), compound_noun(x11, x12),
object(distribution, a8, [x11, a9, a10]),
evt(distribute, x11, [a9, a10]), prop(via, p12, [x12,
x16]),
object('Contactor', a11, [x6]), explication(x16, x18),
object(glc, a12, [x18]), prop(own, p10, _), keyw('Line').
```

Abbildung 7.6: Beispiel für flache logische Form aus EXTRANS

Ergibt sich kein unmittelbarer Match von Query und Dokumenten, so kann auch hier eine schrittweise Relaxierung der Query durchgeführt werden. Ähnlich den semantischen Alternanten in FALCON erlaubt auch EXTRANS die Berücksichtigung sinnverwandter semantischer Elemente (Synonyme, Hyponyme) bei der Suche. Dies wird als Durchführung bestimmter Inferenzen aufgefasst, die wiederum aus Effizienzgründen präkompiliert werden und so direkt verfügbar sind.

Aufgrund der Tatsache, dass WOLFRAM|ALPHA und TRUE KNOWLEDGE Fragen primär auf der Basis von (Schlussfolgerungen über) einer formalen Repräsentation von Information beantworten, sind diese Antwortmaschinen eher mit EXTRANS als mit WebQAS verwandt. Was insbesondere WOLFRAM|ALPHA dabei auszeichnet, ist dass viel 'Wissen' in den Algorithmen steckt, domänenspezifische Fragen anhand der richtigen Information konzise zu beantworten, z.B.:

- Q** number of inhabitants of Rome?
 (Anfrage an Wolfram|Alpha, 30.6.09)
- A** 2.563 million people (2004 estimate)

7.4.3 DeepQA

Seit ca. 2007 existiert ein von der IBM finanziertes Projekt, das zwar nicht tiefe Analyse im Sinne logischer Repräsentationen wie **EXTRANS** durchführt, das aber flache Verfahren ähnlich wie **FALCON** vertieft (*Ferrucci et al.* 2010). Bekannt geworden ist DeepQA durch das System **WATSON**, das im Februar 2011 im Fernsehquiz *Jeopardy!* gegen zwei menschliche Champions antrat und gewann.

WATSON ist insofern ein Meilenstein, als gezeigt worden ist, dass effektives open domain QA in zwar begrenzten, aber realen Kontexten realisierbar ist. Hierzu musste gewährleistet werden, dass das System mit relativ komplexen Fragen umgehen, mit hoher Präzision korrekte Antworten finden und mit ebenso hoher Zuversichtlichkeit aus Alternativen die beste auswählen kann, und zwar alles mit hoher Geschwindigkeit.

Ermöglicht wurde der Erfolg von **WATSON** durch hohen Forschungsaufwand (überdurchschnittlich großes Team und intensive Kollaboration mit 8 Universitäten), durch eine innovative Softwarearchitektur (die durch Erweiterbarkeit und Evaluierung einzelner Komponenten charakterisiert ist) und eine enorme Rechenpower (fast 3000 vernetzte Prozessoren), sowie durch mehrere Terabyte große Wissensressourcen, die vor dem Einsatz des Systems anhand bekannter Wissensressourcen (Enzyklopädien, Wörterbücher, Wordnet etc.) und selektiven Web-Harvesting aufgebaut wurden (zu einer Übersicht über **WATSON** und seine Entwicklung, mit einem Schwerpunkt auf die sprachtechnologischen Aspekte, s. Vortrag Ferrucci).

Der Kern des Vorgehens lässt sich an der Bezeichnung der DeepQA-Architektur ablesen: „massively parallel probabilistic evidence-based“. Er besteht darin, für eine analysierte Frage so viele findbare Evidenzen wie möglich gleichzeitig zu verarbeiten und trotzdem schnell zu einem guten Ergebnis zu gelangen.³ Der innovative Mehrwert besteht darin, ständig die gesammelten Hinweise zu bewerten, aufeinander zu beziehen, Uninteressantes abzuschneiden und schließlich durch Algorithmen der Zuverlässigkeitseinschätzung eine Rangordnung der Ergebnisse herzustellen, an der sich die am wahrscheinlichsten richtige Antwort ablesen lässt. Im Vergleich zu neueren unüberwacht ablaufenden Informationsextraktionsverfahren wie z.B. **TEXTRUNNER** (s. S. 68), die ebenfalls enorme Menge an relevanten Informationselementen bereitstellen, schafft es **WATSON** also, systematisch die Spreu vom Weizen zu trennen.

7.5 Evaluation: TREC-QA

Die **TREC** ist eine Workshop-Serie, die die Infrastruktur für eine „large-scale“ Auswertung von Texterschließungstechnologien zur Verfügung stellt. Dies sind: realistische Testkollektionen (tausende MB Daten), einheitliche, angemessene Bewertungsverfahren sowie ein Forum für den Austausch von Forschungsideen und für die Diskussion von Forschungsmethodologien. Die Anforderungen, die in der TREC an die teilnehmenden Systeme gestellt werden, werden ständig hinterfragt und geändert. Insgesamt lässt sich ein erheblicher Anstieg der Anforderungen

³ Die Autoren berichten, dass **WATSON** auf einem PC über zwei Stunden für eine Fragebeantwortung rechnen würde.

über die Jahre feststellen, woraus gleichzeitig ein theoretischer und praktischer Fortschritt in der Technologie resultiert. Der Effekt der TREC zeigt sich nicht zuletzt am stattfindenden Technologietransfer. Beispielsweise wurzelt das LCC-System in einem TREC-QA-Sieger. Die erste TREC-Konferenz fand 1992 statt, 1999 war Start des TREC-QA tracks (der zum letzten Mal 2007 stattfand).

Der TREC-8 QA track (1999) verwendete ein Zeitungskorpus mit einem Umfang von 1900MB Daten. Teilnehmende Systeme mussten auf 200 Fakten-basierte Fragen kurze Antworten („**factoids**“) generieren. Die Fragen wurden nach Sichtung der Daten konstruiert, um sicherzustellen, dass eine explizite Antwort garantiert vorkommt. Für jede der Fragen sollten 5 Antworten in Rangordnung geliefert werden (mit Zeiger auf die jeweiligen Dokumente); das Format der Antworten war auf eine Größe von 50 bzw. 250 Byte festgelegt. Jede Antwort wurde von 2 Personen ausgewertet, die entschieden, ob eine Systemantwort die gewünschte Antwort enthielt. Als Bewertungsverfahren für die Gesamtgüte der Antworten wurde der *MRR* (*Mean Reciprocal Rank*) verwendet (der *reciprocal rank* ist der inverse Rangplatz p der ersten korrekten Antwort: 1/p).

Der TREC-9 QA track (2000) basierte auf 3000MB Daten aus demselben Korpus. Es wurden 693 Fragen verwendet, wobei 500 aus Suchmaschinen-Logs (Encarta, Excite) entnommen wurden (die übrigen waren Umformulierungen). Die Antwort war jeweils wieder in den Daten vorhanden.

Der TREC-10 QA track (2001) enthielt wieder eine Faktoid-Aufgabe, allerdings waren nur 50Byte-Antworten zugelassen, und die richtige Antwort musste nicht in den Daten vorhanden sein. Zusätzlich existierte eine **Listen-Aufgabe**, d.h., es wurden (An-)Fragen gestellt, auf die mehrere Antwortinstanzen zu liefern waren (z. B. *Name 20 countries that produce coffee.*). Letztere konnten auf mehrere Dokumente verteilt sein. Eine **Kontext-Aufgabe**, für die Diskursobjekte über mehrere Fragen hinweg beibehalten werden mussten, ergab keine interessanten Resultate und wurde fallengelassen.

Der TREC-11 QA track (2002) verwendete Nachrichtentexte aus dem AQUAINT Korpus (3 GB, ca. 1033000 Dokumente). Für die Hauptaufgabe wurde diesmal eine exakte Antwort verlangt (d.h., nicht nur den Text, der sie enthält), wobei jeweils nur eine Antwort pro Frage (insgesamt 500) erlaubt waren. Die Frage-Antwort-Paare mussten in einer Konfidenzreihenfolge abgegeben werden, so dass das Bewertungsverfahren in (6) angewendet werden konnte:

$$(6) \quad \frac{1}{Q} \sum_{i=1}^Q \frac{\text{Anzahl korrekte Antworten bis zum i-ten Rang}}{i}$$

Der TREC-12 QA track (2003) verlangte für die Hauptaufgabe ebenfalls eine exakte Antwort für 500 Fragen. Allerdings existierten nun drei Fragetypen: Fakten (410), Listen (40) und **Definitionen** (50) (*Who is Colin Powell?*). Die Gesamtbewertung setzte sich wie folgt zusammen (FB, LB und DB sind die Bewertungen für die Fragen der jeweiligen Typen): $\frac{1}{2}FB + \frac{1}{4}LB + \frac{1}{4}DB$. LB und DB basieren auf dem F-Maß ($\frac{2PR}{(P+Q)}$), FB ist Anteil der korrekten Antworten. Neben der Hauptaufgabe wurde eine **Passagenaufgabe** gestellt, die unexakte Antworten in einem Textfenster von 250 Byte erlaubte (oder „NIL“, wenn keine Antwort gefunden wurde). Bewertungsmaßstab der 413 faktoiden Fragen war der Anteil der korrekten Antworten.

Der TREC-13 QA track (2004) unterschied sich von dem in 2003 dadurch, dass der Kontext vorangegangener Fragen mit berücksichtigt wurde. Systeme mussten Serien von Faktoid- und Liste-Fragen bzgl. eines Themas beantworten und dann „andere“ Information zu diesem Thema liefern („Other-question“, „Und sonst?“), die durch die vorangegangenen Fragen nicht abgedeckt war.

Eine zusätzliche Aufgabe des TREC-14 QA track (2005) war die „relationship task“, bei der eine Frage *relativ* zu einer bestimmten Aussage („topic statement“), die einen Kontext vorgab, beantwortet werden musste. Das System sollte jeweils die Evidenz für die Antwort mitliefern (die gefundene Information, ähnlich der Other-question).

Der TREC-15 QA track (2006), wie auch der TREC-16 QA track (2007), beinhaltete mit der ciQA (complex, interactive QA)-task eine Verallgemeinerung der relationship-task.⁴ Vorgegeben waren Muster („template“) für Fragen, z. B. ‚What evidence is there for transport of [goods] from [entity] to [entity]?‘. Eine komplexe Frage („topic“) bestand dann aus einer Kontext-Aussage („narrative“) und einer instantiierten Frageschablone:

Template	What evidence is there for transport of [drugs] from [Bonaire] to [the United States]?
Narrative	The analyst would like to know of efforts made to discourage narco traffickers from using Bonaire as a transit point for drugs to the United States. Specifically, the analyst would like to know of any efforts by local authorities as well as the international community.

Gleichzeitig bestand auch die Möglichkeit, ein Web-QA-System zu entwickeln, mit dem ein Gutachter in Bezug auf die Fragebeantwortung interagieren kann/muss. Diese Interaktivität durfte beliebig gestaltet werden (grafisch oder in Richtung eines Dialogsystems, s. Kap. 8). Die Daten für den TREC-16 QA track betragen mittlerweile 3GB Newswire-Dokumente und 136 GB Blog-Texte.

Eine multilinguale Variante und Fortführung der TREC ist das „Cross Language Evaluation Forum“ (**CLEF**), das ebenfalls einen QA-track besitzt.

2008 (aber nicht 2009) existierte ein QA-Track der **Text Analysis Conference (TAC)**. Die Aufgabe bestand darin, Meinungsfragen relativ zu einem Korpus von Blog-Texten zu beantworten, z. B. „Which rock bands do college students like?“ oder „What do people like about Ikea?“.

7.6 Ausblick

Eine generelle gegenwärtige Entwicklung im QA ist die Zusammenführung verschiedener Ansätze und unterschiedlicher Verfahren für die Aufgabe der Fragebeantwortung sowie ein zunehmendes Interesse an „traditionellen“ Informationsressourcen (d.h., Datenbanken). Ein Beispiel hierfür ist das hybride System **QUETAL** (*Frank et al. 2005*), das zum Einen sowohl

⁴ Die Hauptaufgabe blieb die Beantwortung einer Serie von Fragen.

flache wie auch tiefe Analysetechniken verwendet und zum Anderen Antwortextraktion bzgl. unterschiedlicher Typen von Datensammlungen (unstrukturiert bis strukturiert) ermöglicht. Hierzu erstellt es aus einer Frage eine flache semantische Analyse, die mit Hilfe entsprechender Regeln in eine sprachunabhängige konzeptuelle Repräsentationsform übersetzt wird (die *proto query*). Diese wird dann in eine Query des jeweiligen Daten-Formats (Datenbanken, Ontologien) überführt (*search query*).

Weiterhin ist eine zunehmende Involvierung des Benutzers in den Prozess der Beantwortung immer komplexerer Fragen zu beobachten, die immer umfangreichere Antworten erfordern. Zum Beispiel führt das System **HITIQA** (Small et al. 2004) eine interaktive Fragebeantwortung für *analytische* Fragen durch, deren Antwort sich nicht auf ein einfaches Faktenelement reduzieren lässt. Das System führt mit dem Benutzer einen einfachen Dialog, der auf die Einschränkung der gefundenen Antwortkandidaten bzw. auf eine Verfeinerung der Frage ausgerichtet ist. Die Präsentation des Ergebnisses involviert dabei eine einfache *Textzusammenfassung*. Dazu werden spezifische und generische Annahmen über den Benutzer benötigt, die auf die zunehmende Bedeutung der *Benutzermodellierung* im QA hinweisen.

Die steigende Komplexität im QA führt außerdem dazu, dass den Fragen und Antworten selbst mehr Aufmerksamkeit geschenkt wird. Zum Einen wird untersucht, wie komplexe Fragen –d.h. insbesondere auch Fragen, die in der für den Fragenden intuitiven Form gestellt wurden– in einfachere Fragen zerlegt werden können (*Fragedekomposition*), zum Anderen, wie Informationen aus separat extrahierten Antwortteilen zu einer kohärenten Gesamtantwort zusammengefügt werden können (*Antwortfusion*). Letzteres kann daher auch Techniken der *Sprachgenerierung* involvieren.

Schließlich erfährt der Begriff der **Schlussfolgerung** (*Inferenz, Entailment*), der überwiegend mit wissenbasierten Ansätzen im Bereich des Textverständnisses (z. B. im Projekt **LILOG**, s. Herzog und Rollinger 1991) assoziiert war, eine Renaissance. So lässt sich aus dem Text (7) schließen, dass Yoko Ono John Lennons Witwe ist und eine entsprechende Frage *Wer ist John Lennons Witwe?* beantworten. Diese spezifische Aufgabe ist Gegenstand des **Recognizing Textual Entailment (RTE)**, das 2007 zum dritten Mal als PASCAL⁵ Challenge stattfand und seit 2008 ein Track der **Text Analysis Conference (TAC)** ist .

- (7) *Yoko Ono enthüllte eine Bronzestatue ihres verstorbenen Ehemannes John Lennon, um die offizielle Umbenennung des englischen „Liverpool Airport“ in „Liverpool John Lennon Airport“ abzuschließen.*

In praktischer Hinsicht geht der Weg in Richtung „semantischer Antwortmaschinen“ à la **WOLFRAM|ALPHA** und **SIRI**: (multimodale (s. Kap. 10)) Geräte, als hand-held, im Auto o.ä., die durch Funk-Zugriff auf das Web auch in mobilen Anwendungssituationen eine (multimodale) Beantwortung beliebiger Fragen erlauben (s. das Projekt **SMARTWEB**, Wahlster 2007). Um die Entwicklung von Frage-/Antwort-Systemen zu forcieren, zeichnet sich ein kollaborativer Ansatz von Entwicklern, Anwendern und Geldgebern ab (sog. Open Advancement of Question Answering Systems, s. Ferrucci 2009).

⁵ Pattern Analysis, Statistical Modelling and Computational Learning.

7.7 Literatur

Ein lesenswertes Papier, das historische Aspekte (wie die in dessen Anhang aufgeführten Taxonomien von Lehnert und Graesser) mit aktuellen Fragestellungen und zukunftsgerichteten Planungen verbindet, ist *Burger et al.* 2002. Einen guten Überblick über Systeme für QA in beschränkten Domänen liefern *Mollá und Vicedo* 2007.

Kapitel 8

Dialogsysteme (DS)

[Quaid wakes up in a Johnnycab]

Douglas Quaid: *Where am I?*
Johnnycab: *You're in a Johnnycab.*
Douglas Quaid: *I mean, what am I doing here?*
Johnnycab: *I'm sorry. Would you please
rephrase the question?*
Douglas Quaid: *How did I get in this taxi?*
Johnnycab: *The door opened. You got in.*
[Johnnycab rolls his eyes]

(aus: *Total Recall (Film)*)

Ein **Dialogsystem (DS)** ist ein natürlichsprachliches, interaktives System, das dem Benutzer erlaubt, Zugang zu Systemfähigkeiten durch das Führen eines Dialogs zu erlangen. Dies bedeutet, dass sich die Interaktion von Benutzer und System nicht, wie bei QAS, auf eine Interaktionseinheit (wie 'Frage-Antwort') beschränkt, sondern dass ein DS die inhärente Fähigkeit besitzen muss, größere sprachliche Interaktionsstrukturen (Diskurse, Dialoge) bewältigen zu können und auch mit unterspezifizierten Äußerungen, die nur in Bezug auf den Kontext verständlich sind, umgehen können muss.

I'm sorry, Dave, I'm afraid I can't do that

(HAL in: *2001: A Space Odyssey*)

Im Idealfall sind Dialogsysteme außerdem in der Lage, über ihre eigenen Fähigkeiten, Ziele und Wünsche „nachzudenken“ und ihrem Dialogpartner das Resultat ihrer Überlegungen mitzuteilen, was an der malevolenten Äußerung von HAL besonders deutlich wird.

8.1 Allgemeines zu DS

8.1.1 Funktionalität von DS

Die umfangreichere Funktionalität von Dialogsystemen erfordert z. B. eine Buchführung über die Dialoghistorie, Wissen über die Struktur von Dialogen (*Dialogmodell*) und über den Benutzer im Spezifischen und Allgemeinen (*Benutzermodelle*). Sie umfasst generell *Sprachverstehen* und -*Generierung*, d.h. die Interpretation bzw. Produktion einer Äußerung im gegebenen Kontext und unter Ausnutzung aller in Abb. 3.2 auf Seite 38 dargestellten Wissenstypen. Die „Tiefe“ von Verstehen und Generierung variiert allerdings. Bei angewandten DS werden – vor allem im Hinblick auf die Optimierung der Systemperformanz – in der Regel flachere Verfahren eingesetzt, was den Verzicht auf bestimmte Systemaspekte und -komponenten bedeutet.

Ein wesentlicher Gradmesser der Kompetenz eines DS ist die Art der Initiative, die es erlaubt. In einfachen Systemen übernimmt ausschließlich das System die Initiative, kompetentere Systeme überlassen auch dem Benutzer die Initiative (mit dem Risiko, dass das System den Benutzer nicht versteht). Angestrebt werden sog. *mixed initiative* Dialoge, also die Möglichkeit für beide Dialogpartner, jeweils die Initiative zu übernehmen und ein DiskurstHEMA einzuführen. Dies erfordert die Fähigkeit des „*turn-takings*“, d.h. zu wissen, wann ein Dialogpartner „dran ist“/den turn übernehmen darf.

Die Funktionalität eines DS geht entsprechend über reine Fragebeantwortung hinaus und ermöglicht dem Benutzer im Idealfall die kooperative Lösung von Problemen (z. B.: *Ich möchte gerne morgen nach Hamburg fahren*). Kooperativität umfasst dabei einerseits das Erreichen eines gemeinsamen Ziels (z. B. Benutzer *und System* wollen das Problem des Benutzers lösen) und andererseits das Etablieren/Aufrechterhalten des *common ground* (in etwa: gemeinsamer Kenntnisstand, damit man „über dieselben Sachen redet“).¹ Dies erfordert auch die Modellierung pragmatischer Aspekte. So stellt sich in einem „natürlichen“ Dialog oft die Frage, welche Handlung genau mit einer Äußerung durchgeführt wird. Prototypisch hierfür sind Äußerungen, die sog. *indirekte Sprechakte* ausdrücken (*Können Sie mir sagen, ...*). Hier sollte nicht mit *Ja* geantwortet werden.

Eine spezifische Form von DS sind **Sprachdialogsysteme** (*spoken dialogue systems, SDS*, s. auch Kap. 9). Sie erlauben die Ein- und Ausgabe gesprochener Sprache (Spracherkennung, Sprachsynthese) und sind gegenwärtig insbesondere für kommerzielle Anwendungen interessant. Dafür sind sie mit den Problemen der Verarbeitung gesprochener Sprache konfrontiert und müssen diese kompensieren – sowohl praktisch im Hinblick auf Aspekte der Sprachsignalverarbeitung (z. B. Rauschen) als auch theoretisch, wie z. B. die erhöhte Komplexität durch den Umgang mit ungrammatischen, unvollständigen oder gar nicht verstandenen Äußerungen (z. B. Rückfragen). Sie sind daher in der Gesamtkompetenz (Flexibilität, Domänenunabhängigkeit) meist stark eingeschränkt.

DS sind heute generell auf eine Aufgabe wie Auskunftserteilung, Beratung, Kooperation bei

¹ Unter „grounding“ versteht man entsprechend die Bestätigung, dass ein Beitrag des Dialogpartners verstanden wurde.

Problemlösung, Steuerung (von Automobilen) oder Ausbildung (Tutorielles System) festgelegt. Nahezu alle heute im Einsatz befindlichen kommerziellen Systeme sind Informationssysteme (z. B. Reiseauskünfte) oder Transaktionssysteme (Bestellung, Vermittlung, finanzielle Transaktionen). Es ist jedoch offensichtlich, wie groß das Potential (insbesondere das von SDS) bei zunehmender Kompetenz auch für die anderen Aufgabenbereiche ist.

Dialogsysteme können im Übrigen als Spezialfälle multimodaler Systeme (s. Kap. 10) angesehen werden, deren Kern multifunktional ist und deren Analyse- und Generierungskomponenten mit weiteren Modalitäten außer Sprache umgehen können.

8.1.2 Grobklassifikation von DS

Der Terminus „Dialogsystem“ lässt sich auf sehr unterschiedliche Systeme anwenden, die im Folgenden zumindest in grobe Klassen (Chatter-/Lingubots, Sprachdialogsysteme (SDS), Dialogsysteme (DS)) eingeteilt werden sollen, auch wenn die Grenzen zwischen diesen Klassen fließend sind.

ELIZA-Nachfolger: Chatterbots

Ein wesentliches Merkmal von **ELIZA** (und Nachfolgern wie **PARRY** und **RACTER**) war es, den Eindruck eines tatsächlichen Dialogs zu erzeugen, ohne hierfür „tiefgehende“ Intelligenz zu verwenden. Die intellektuelle Inkompétence dieser Programme wurde weitgehend dadurch zu verstecken versucht, dass das Programm eine spezifische Persönlichkeit (bei **ELIZA** ein Therapeut, bei **PARRY** ein Schizophrener) zu suggerieren versuchte. Letztlich konnten solche einfachen Muster-basierten DS aber nur auf den ersten Blick beeindrucken und sonst nicht überzeugen.

Mit dem 1990 eingerichteten Loebner Preis wurde ein Versuch unternommen, die Entwicklung solcher Dialog-aufrechterhaltender „Schwatz-Programme“ (**Chatterbots**) systematisch voranzutreiben (mit anfänglich recht billigen Lösungen²).

Aktuell sehr erfolgreich ist **A.L.I.C.E.** (Sieger 2000, 2001, 2004), das die „Reaktions-Einheiten“ (bestehend u.a. aus Stimulus-Muster und Response-Schablone) in der *Artificial Intelligence Markup Language* (AIML) notiert. Der Bezug zur Tradition des Behaviourismus ist dabei ernst gemeint, wie das folgende Zitat zeigt:

Academics are fond of concocting riddles and linguistic paradoxes that supposedly show how difficult the natural language problem is. ‘John saw the mountains flying over Zurich’ or ‘Fruit flies like a banana’ reveal the ambiguity of language and the limits of an **A.L.I.C.E.**-style approach (though not these particular examples, of course, **A.L.I.C.E.** already knows about them). In the years to come we will only advance the frontier further. [...] The work of those seeking to explain natural language in terms of something more complex than stimulus response will take place beyond our frontier, increasingly in the hinterlands occupied by only the rarest forms of language. Our territory of language already contains the highest population of sentences that people use. Expanding the borders even

² s. die Kritik von Stuart Shieber.

more we will continue to absorb the stragglers outside, until the very last human critic cannot think of one sentence to 'fool' **A.L.I.C.E.**

(Richard S. Wallace, *AIML Overview*)

Aus computerlinguistischer/kognitionswissenschaftlicher Perspektive ist diese Haltung bestenfalls naiv zu nennen, für die Bildung von DS im Allgemeinen ist sie unsinnig, da das Vortäuschen von Intelligenz Chatterbot- aber nicht DS-charakteristisch ist. Auch aus Sicht behaviouristischer Modellierung von intelligentem Sprachverhalten muss eingeräumt werden, dass die Stimuli nicht allein sprachlich sind, sondern auch in den nicht-sprachlichen Aspekten der jeweiligen Situation bestehen (mal ganz abgesehen von den in der Kognitionswissenschaft in den Vordergrund gerückten internen Faktoren wie Gedächtnis, Aufmerksamkeit und sonstigen Aspekten der Struktur und Funktion des Gehirns).³

Eine weniger ideologische, pragmatische und damit sinnvolle Variante sind die **Lingubots**. Sie dienen gewissermaßen als Ersatz für ein Call-Center und bieten auf Web-Seiten die Möglichkeit, in natürlicher Sprache Informationen abzufragen. Dies geschieht in einem Dialog und (höchstwahrscheinlich) mit Chatterbot-ähnlichen Techniken, allerdings wird das Dialoginteresse immer wieder auf das Angebot der jeweiligen Website gelenkt (insbesondere bei Nicht-Verstehen des Lingubots). Interessanterweise hat der komplexeste Lingubot der Firma **ARTIFICIAL SOLUTIONS** den Loebner-Preis 2008 gewonnen.

Die Chat-Technologie lässt sich jedoch noch viel weitgehender kommerzialisieren. So bietet **UPSELLIT.COM** automatische Chats an verschiedenen Stellen in Web-Interaktionen mit Usern an. Beispielsweise werden Kunden, die das Ausfüllen eines Formulars abbrechen, in ein automatisches „Gespräch“ verwickelt, in dem Hilfestellung angeboten wird, die letztlich zur Wiederaufnahme des Ausfüllens führen soll. Letzteres resultiert angeblich in einem um 7-20% erhöhten Anteil an Abschlüssen.

SDS in der Speech-Technologie

Die Verbesserungen in der Spracherkennung haben ein weites Feld an kommerziellen Anwendungen eröffnet, insbesondere im Telefoniebereich. SDS sind hier Erweiterungen von IVR (Interactive Voice Response) Anwendungen in Richtung einfacher Dialogfähigkeit, z. B. für sprachgeleitete Auskunftssysteme. Andere Anwendungen finden sich z. B. im Bereich der Steuerung von Geräten, z. B. bei SDS im Kfz (s. Hamerich *et al.* 2005) oder sogar im Weltraum (s. das Projekt Clarissa). Diese Art der SDS-Entwicklung entstammt weniger der Computerlinguistik, sondern eher der ingenieurwissenschaftlichen Domäne.

DS als Forschungssysteme

Beginnend mit Systemen wie **SHRDLU** oder **GUS** (Bobrow *et al.* 1977), hat sich die Entwicklung wissensbasierter DS bis in heutige Zeit fortgesetzt. Auch wenn man versucht, wie in anderen Bereichen Korpus-basiert vorzugehen, sind wesentliche Systeme noch dem rationalistischen Paradigma zuzuordnen.

³ Eine Chatterbot-Plattform mit –anscheinend– höherem Anspruch und tieferer Künstlicher Intelligenz ist übrigens **SYLVIA** (Symbolically Isolated, Linguistically Variable, Intelligence Algorithms) von der **COGNITIVE CODE CORPORATION**.

GUS

Ein weiterer Dialog-Klassiker (nach **ELIZA** und **SHRDLU**) ist das Genial Understanter System (**GUS**) von Daniel Bobrow (*Bobrow et al.* 1977), das Mitte der 1970er Jahre bei Xerox PARC entwickelt wurde. Es behandelte die Domäne der Flugauskünfte und war in der Lage, flexible, kooperative Dialoge zu führen.

In dem folgenden Beispiel wird deutlich, dass indirekte Antworten und Zwischenfragen des Benutzers ohne Probleme bewältigt werden.

```

GUS: Hello. My name is GUS. I can help you plan a simple trip by air. Where do you
      want to go?
Ben: I want to go to San Diego on May 28.
GUS: What time do you want to leave?
Ben: I must be in San Diego before 10am.
GUS: Would you like P.S.A. flight 102 that arrives at 9.15am?
Ben: What is the next flight?
GUS: Air California flight 3.10 that leaves at 8.30am and arrives at 10.00am.
Ben: I'll take the first one.
GUS: In what name should I make a reservation, please?
```

Angewandte DS

Bei der Entwicklung angewandter DS werden überwiegend computerlinguistische Verfahren verwendet (oft handelt es sich um Produkte, die im Rahmen universitärer Spin-offs entwickelt wurden). Dabei wird praxisorientiert und gemäß den Richtlinien des Software-Engineering vorgegangen. Heute sind insbesondere die Grenzen zwischen den Forschungs- und angewandten Systemen sehr schwer zu ziehen: ein real eingesetztes DS kann äußerst rationalistisch ausgerichtet sein (s. S. 134) und ein Forschungssystem kann stark praxisorientiert sein (s. *Wahlster et al.* 2001).

8.2 Aspekte und Anforderungen

Ein ideales Dialogsystem ist zuallererst **benutzerfreundlich** und verhält sich dazu **rational** und **kooperativ**. Rationales Verhalten des Systems ergibt sich bzw. „folgt“ aus den Annahmen/dem Wissen über die Domäne, den Dialog, den Benutzer im Spezifischen und im Allgemeinen unter Berücksichtigung allgemeiner logischer Prinzipien (s. hierzu *Sadek* 1999). Kooperatives Verhalten des Systems ergibt sich aus der (Über-)Erfüllung der Wünsche des Benutzers und zeigt sich in flexiblem Agieren bei Verständnisproblemen.

Ein ideales Dialogsystem ist somit generell auf die Maximierung der Kooperativität ausgerichtet. Hierzu ist es **verhandlungsfähig**, **kontextsensitiv**, zeigt **interaktive Flexibilität** und liefert **kooperative Reaktionen und Antworten**.

Verhandlungsfähigkeit kennzeichnet ein DS, wenn es auch auf unklare/underspezifizierte Fragen eingeht und das gewünschte Diskursziel abklärt. Wesentliche Merkmale sind

- **Hilfestellung** bei unpräzisen Anfragen des Benutzers:
B: *Ich habe Hunger.*
S: *Soll ich Ihnen ein paar Restaurants in Ihrer Nähe nennen?*
- **Einschränkungen** möglicher Antworten verhandeln:
S: *Ich lasse die Gourmet-Restaurants aus, ist das ok?*
- **Umgang** mit leeren Antwortmengen:
S: *Zur Zeit hat kein Restaurant geöffnet. Was halten Sie von einem Döner-Imbiss?*
- **Eingehen** auf den Benutzer:
B: *Nee, gibt es denn keinen McDonalds in der Nähe?*

Kontextsensitivität bezeichnet die Eigenschaft eines DS, Anfragen im Kontext zu interpretieren und sprachliche „Ungenauigkeit“ entsprechend zu akzeptieren. Hierzu gehört die adäquate Reaktion auf

- **Rückfragen** des Benutzers, die sich auf vorangegangene Teile des Dialogs beziehen, und allgemeiner der
- **Umgang** mit sprachlichen Phänomenen wie Ellipsen, Anaphern, Deixis
B: *Wie weit ist das Restaurant denn?*
B: *Wie weit ist das denn?*
B: *Wie weit ist das denn von hier?*

Interaktive Flexibilität ist gegeben, wenn ein DS Abweichungen von typischen Dialogverläufen akzeptiert. Danach sollte es dem Benutzer erlaubt sein, Folgendes zu tun:

- **Rückfragen stellen** (B: *Kann ich da zu Fuß hin gehen?*)
- **Zusatzinformation einschieben** („Out-of-Focus-Antworten“)
S: *Wann wollen Sie nach Bern fahren?*
B: *Ich möchte aber durchfahren können.*
- **das Thema ändern** (B: *Wie spät ist es eigentlich?*)
- **das Dialogziel ändern** (B: *Nee, ich esse doch erst zu Hause etwas. Wann fährt der 17er Bus?*)
- **Missverständnisse des Systems korrigieren** (B: *Nee, keine Hotel-Restaurants.*)
- **sich selbst korrigieren** (B: *Äh, vielleicht doch, wenn's nicht zu teuer ist.*)
- **das DS unterbrechen** („Barge-In“)

Als **kooperative Reaktionen** können gelten:

- **Vervollständigende Antworten** (Überbeantwortung):

B: *Wann fährt der nächste Zug nach Basel?*
 S: *17:15, Gleis 3*
- **Korrigierende Antworten** (z. B. bei Präspositionsverletzungen):

B: *Wann fährt der nächste Zug nach Stuttgart?*
 S: *Es gibt keinen. Sie müssen immer in Karlsruhe oder Mannheim umsteigen.*
- **Vorschläge/Alternativen** nennen:

B: *Wann fährt der nächste Zug nach Bern?*
 S: *17:15, Gleis 3. Wenn Sie eine Stunde später fahren, müssen Sie aber nicht in Basel umsteigen.*
- **Nennen einschränkender Faktoren** (konditionale Antworten):

B: *Wann fährt der nächste Zug nach Bern?*
 S: *17:15, Gleis 3. Da müssen Sie aber in Basel umsteigen.*
- **Intensionale Antworten** (Umschreibung der Antwortmenge):

B: *Welcher ICE nach Zürich fährt durch Basel?*
 S: *Alle.*

Adäquate Antworten sind solche Antworten, die **relevant** (auf die Frage bezogen), **korrekt** (faktisch richtig), **prägnant** (keine überflüssige Information enthaltend), **vollständig**, **kohärent** (leicht verständlich) und bzgl. des gegebenen Dialogs **gerechtfertigt** (es ist klar für den Benutzer, warum die Antwort gegeben wurde) sind.

8.3 DS-Architektur

8.3.1 Überblick

Ein Dialogsystem ist eine Integration dessen, was vor allem früher als *natural language front end* (auch: *natural language interface*) bezeichnet wurde, und dem eigentlichen Anwendungssystem (dem *back end*). Diese Unterscheidung wird an der in Abb. 8.1 auf der nächsten Seite veranschaulichten Grobarchitektur von DS deutlich, die an die Architektur des Systems **MALIN** (*Flycht-Eriksson und Jönsson 2000*) angelehnt ist. Die Abbildung zeigt die Trennung der sprachlichen Dialogverwaltung (*Dialogsteuerung/-management*) und der nicht-sprachlichen Verwaltung der beteiligten Anwendungssysteme und Domänenaspekte mit Hilfe spezifischer Module. Stärker angewandte Systeme weisen in der Regel eine einfachere Architektur, eine weniger wissensintensive Dialogsteuerung und eine stärkere Bindung/Verzahnung von Dialogsteuerung und Hintergrundsystem auf (s. z. B. *Kellner 2004*).

Das **Dialogmodell** enthält die Information darüber, wie Dialoge (normalerweise, typischerweise) ablaufen. Dieses Wissen wird dazu verwendet, einen aktuell vorliegenden Dialogschritt zu klassifizieren oder den nächsten Dialogschritt zu bestimmen. Die **Dialoggeschichte** (*dialogue history*) verzeichnet den bisherigen Verlauf des Dialogs. An ihr kann abgelesen werden,

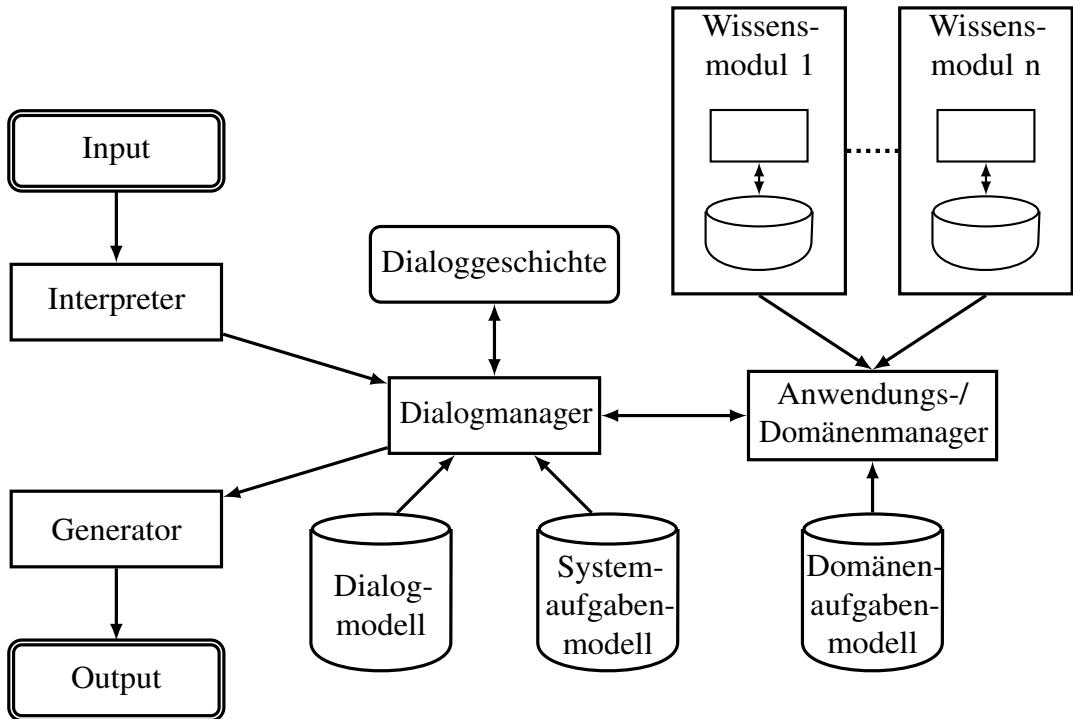


Abbildung 8.1: DS-Architektur (MALIN)

wann welche Referenten eingeführt wurden. Zusammen mit dem gemeinsamen Hintergrundwissen (dem Domänenmodell) lässt sich daraus ableiten, in welcher Weise auf Objekte sprachlich referiert werden kann bzw. muss (mit Pronomina, (in)definiten Kennzeichnungen, Namen).

Die Wissensmodule entsprechen dem **Domänenmodell** in einem NLS, in dem das relevante spezifische Wissen über bestimmte Bereiche abgelegt ist (hier als Repräsentationen plus Zugriffs-/Inferenzoperationen). Dieses gliedert sich in konzeptuelles, generisches Wissen („Eine Zugauskunft ist eine Auskunft über eine Zugverbindung von A nach B einer Auskunft gebenden Person P1 an eine Auskunft wünschende Person P2“) und Faktenwissen („Am 14.7.2005 erhielt Peter Müller eine Zugauskunft von Osnabrück nach Stuttgart“) (zu einer allgemeinen Darstellung (der Repräsentation) nicht-sprachlichen Wissens s. auch Carstensen 2009). Das **Systemaufgabenmodell** (*system task model*) charakterisiert die Funktionalität des DS, während das **Domäneaufgabenmodell** definiert, welche Aufgaben durch welches Anwendungssystem erledigt wird.

Ein **Benutzermodell** (*user model*, in MALINs Architektur nicht enthalten) liefert dem DS einerseits Annahmen über bestimmte Benutzertypen (sog. *a priori Benutzermodell*), andererseits verzeichnet es die Information über das Wissen und die Präferenzen bestimmter Benutzer.

8.3.2 Dialogsteuerung/-management

Die **Aufgabe der Dialogsteuerung** besteht in der Kontrolle des Dialogflusses und involviert Folgendes:

- **Dialogakt- und Intentionserkennung:**
anhand des sprachlichen Inputs muss bestimmt werden, welchen Dialogakt der Benutzer durchgeführt hat und was genau damit bezweckt wird
- **Update des Dialogkontexts:**
Aktualisierung der Dialoghistorie
- **Kommunikation mit der Anwendung:**
Weiterreichen von Anfragen an den Domänenmanager für die Interaktion mit der jeweiligen Anwendung und Abrufen des Ergebnisses
- **Planung des nächsten Dialogschritts:**
Abhängig von den gemäß Dialogmodell prinzipiell gegebenen Möglichkeiten und dem aktuellen Dialogkontext wird der nächste Dialogschritt ausgewählt

Der Zweck der Dialogsteuerung besteht einerseits darin, die Optionen des Benutzers restriktieren (Betonung auf „Steuerung“), und andererseits, die Optionen des Benutzers erweitern (Betonung auf „Management“).

Abb. 8.2 auf der nächsten Seite gibt einen Überblick über die verschiedenen Techniken, die zur Dialogsteuerung verwendet werden können. Bei der Verwendung von **finite state scripts**, „weiß“ das System zu jedem Zeitpunkt bzw. in jedem Systemzustand, wie es auf einen Input zu reagieren hat. Der Benutzer steuert somit das Systemverhalten durch seine jeweilige Wahl. Bei **Frame-basierter** Steuerung leitet das System den Dialog bzgl. *eines* bestimmten Kontexts durch gezielte Abfrage von Informationen, die in einem Frame (einer Wissensstruktur mit Attributen und ihren (möglichen) Werten) festgelegt sind (generalisierbar auf mehrere Kontexte). **Plan-basierte** Steuerung ermöglicht eine flexible Dialogsteuerung durch das beliebige Ansprechen von Themen, die für die Lösung eines Problems relevant sind. **Agenten-basierte** Steuerung zeichnet sich dadurch aus, dass das System zusätzlich zur Planung auch Änderungen der dynamischen Welt registrieren und selbständig agieren kann. Ein Beispiel ist folgender fiktiver Dialogabschnitt:

B: *Was ist der Status der Instandsetzungsarbeiten an der Bahnstrecke Bielefeld-Hannover?*

S: *Einen Augenblick. Ich bekomme grad die Meldung rein, dass auch auf der Bahnstrecke Bielefeld-Dortmund die Hochspannungsleitung durch Blitzeinschlag beschädigt ist.*

Verwendete Technik	Beispielaufgabe	Aufgaben-komplexität	Behandelte Dialogphänomene
finite-state script	Telefondurchwahl	einfach	Benutzer beantwortet Fragen
Frame-basiert	Zugauskunft		Benutzer stellt Fragen, einfache Klärungen durch das System
Mengen von Kontexten	Reisebüro-Agent		Wechsel zwischen vorbestimmten Themen (Hotel-/Flugbuchung etc.)
Plan-basierte Modelle	Küchendesign-Beratung		dynamisch generierte Themenstrukturen, kolaborative Verhandlungs-Subdialoge
Agenten-basierte Modelle	Katastrophenhilfe	komplex	verschiedene Modalitäten (z. B. geplante Welt und reale Welt)

Abbildung 8.2: *Alternativen der Dialogsteuerung (nach Allen et al. 2001)*

8.3.3 Dialogmodellierung

Allgemeines zur Dialogmodellierung

Bei DS, die über ein explizites Dialogmodell verfügen (im Gegensatz zu solchen, in denen das Wissen über die Struktur der Dialoge in die Dialogsteuerung integriert ist, wie bei den meisten einfachen/kommerziellen Systemen), stellt sich die Frage nach den Richtlinien für die Dialogmodellierung. Grundlegende theoretische Fragen sind folgende:

- Welche Eigenschaften von Äußerungshandlungen charakterisieren einen Dialog (einer bestimmten Gesprächssorte)?
- Welche Annahmen über die mentalen Zustände der Dialogteilnehmer müssen gemacht werden, damit Verhalten als rationaler kooperativer Dialog angesehen werden kann?
- Was wären rationale und kooperative Erweiterungen des gerade beobachteten Verhaltens?

Bei der Dialogmodellierung für ein Dialogsystem stellt sich die Frage, welche Vereinfachungen möglich, sinnvoll und notwendig sind. Verkomplizierend kommt hinzu, dass Mensch-Maschine-Dialoge nicht (unbedingt) von gleicher Art wie Mensch-Mensch-Dialoge sind. So ist computer-

generierte Sprache einerseits in der Regel noch einfacher, beschränkter, weniger flexibel, andererseits „perfekter“ (d.h., keine Tippfehler, Hesitationen, Wiederholungen; kein Vergessen von etwas, was schon erfahren wurde). Wesentlich für die Dialogmodellierung ist daher eine solide empirische Grundlage, d.h. die Verwendung von Korpora.

Menschliche Dialoge sind hierfür leicht zu erheben, aber per se keine Mensch-Maschine-Dialoge. Letztere lassen sich in **Wizard-of-Oz-Studien** simulieren. Hierzu wird ein funktionierendes DS vorgetäuscht: Probanden kommunizieren, ohne es zu wissen, mit einem Menschen, der die angesprochenen Eigenschaften des Computers nachahmt. Allerdings ist es nicht einfach, diese Täuschung aufrecht zu erhalten, ebenso wie es nicht-trivial ist, den Schwierigkeitsgrad der Aufgabe so zu justieren, dass die gewünschte und für eine umfassende Modellierung erforderliche Dialogvarianz elizitiert („herausgekitzelt“) wird. Eine sich anbietende Alternative ist daher das **Destillieren menschlicher Dialoge**, d.h. das Bereinigen von Mensch-Mensch-Dialogen von Aspekten, die bei Mensch-Maschine-Dialogen nicht auftreten bzw. die nicht relevant sind (s. hierzu den Exkurs auf Seite 129).

Die Verwendung von Korpora setzt die Klärung der Frage voraus, wie die Struktur von Dialogen beschaffen ist (und wie entsprechend Textteile markiert werden müssen). Moderne Ansätze gehen von drei Ebenen der Dialogstruktur aus: die Ebene der **transactions** (oder Dialog-Phasen), auf der Subdialoge charakterisiert werden, die einen wesentlichen Schritt der Dialogpartner realisieren, ihr Ziel zu erreichen (Beispiel: 'Eröffnung' eines Dialogs oder 'Verhandlung-führen')(s. hierzu Carletta et al. 1997).

Transaktionen bestehen aus **dialogue games**, die u.a. als Zweck/Ziel-orientierte Einheiten charakterisiert werden können (z. B. 'Angebot-machen', 'Angebot-ablehnen'). Diese wiederum setzen sich aus **dialogue moves** (z. B. 'Frage', 'Antwort') zusammen, die in der Regel grobe Typen der Bestandteile von *Adjazenzpaaren* sind ('Frage-Antwort', 'Vorschlag-Annahme'). Im System **VERBMOBIL** (Wahlster 2000) werden dialogue moves weiter differenziert in spezifische **dialogue acts** (dialogspezifische Sprechakte). Danach kann auf einen Akt 'Vorschlag' als move 'Erwiderung' entweder ein Akt der 'Rückmeldung' oder der 'Ablehnung' erfolgen.

Klassische Paradigmen der Dialogmodellierung

Generell kann bei der Dialogmodellierung unterschieden werden, ob ausschließlich die sprachliche Struktur des Dialogs oder auch die nicht-sprachlichen Aspekte des zugrundeliegenden Handelns und dessen Planung berücksichtigt werden. Entsprechend lassen sich klassischerweise *strukturelle* und *Plan-basierte* Ansätze unterscheiden (s. Cohen 1997).

Strukturelle Ansätze gehen davon aus, dass Dialoge eine typische, modellierbare Struktur aufweisen, die formal repräsentiert und auf diese Weise dem Dialogmanagement zur Verfügung gestellt werden kann. Eine einfache Art der Realisierung dieser Auffassung ist die Notation der Struktur in Form von **Dialoggrammatiken** (d.h. Modellierung von Dialogen analog zur strukturellen Beschreibung von Sätzen mit Phrasenstrukturregeln). Terminale Elemente einer solchen Grammatik sind Sprechakte (Frage, Antwort etc., d.h. die obigen dialogue acts), Nicht-terminale sind Initialisierung, Reaktion, Schluss, Einbettung etc. (die anderen o.g. Kategorien). Ein Beispiel hierfür zeigt Abb. 8.3 auf der nächsten Seite.



Abbildung 8.3: Beispiel für Dialoggrammatik

Diese Grammatiken lassen sich ggf. in **Endliche Automaten** –sehr effiziente Mechanismen– kompilieren. Insbesondere kommerzielle Systeme verwenden häufig solche **finite state systems**, in denen zu jedem Zeitpunkt (in jedem Zustand des Systems) klar ist, welche (Dialog-)Aktionen das System ggf. ausführen muss und in welchem Folgezustand es sich nach einer Aktionen von Benutzer oder System befindet (s. z. B. Abb. 8.4⁴).

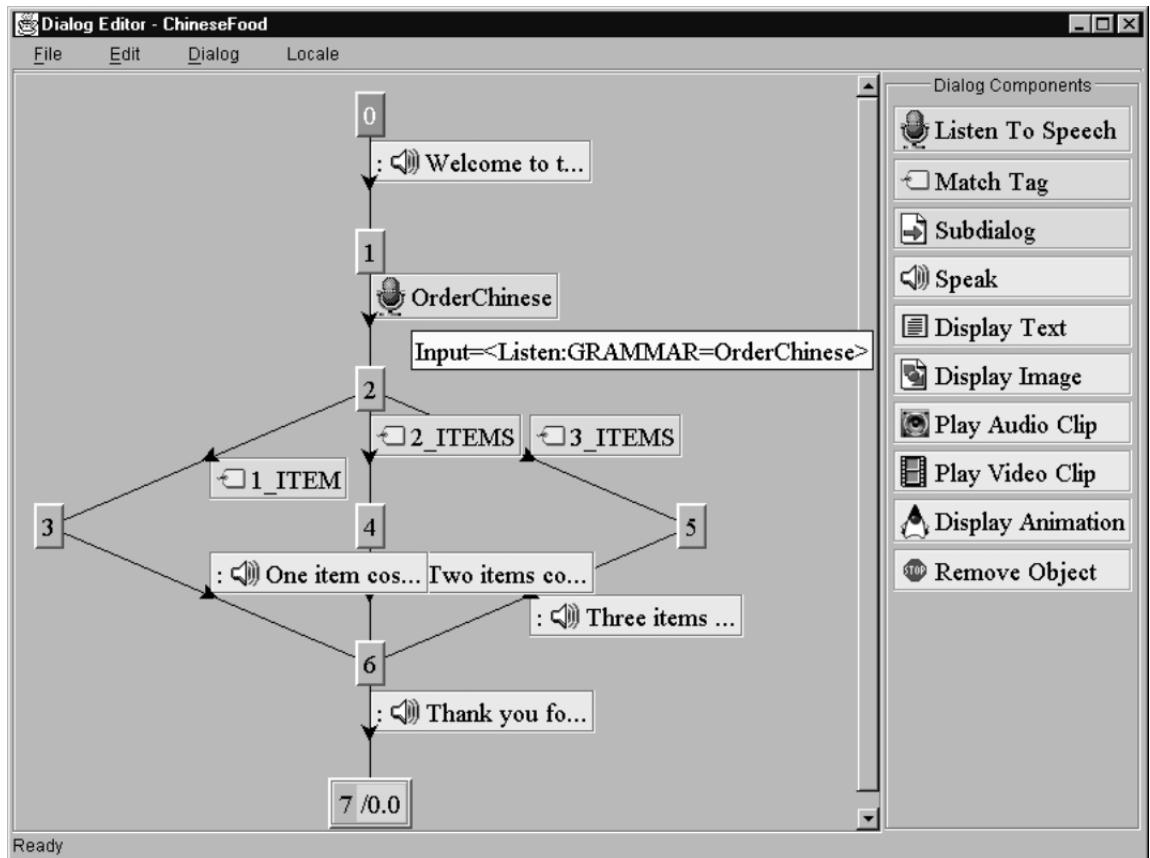


Abbildung 8.4: Beispiel für einen Endlichen Automaten in einem SDS

⁴ aus einer Folie von Matthias Denecke, Computerlinguist und CTO von Agilingua

Strukturelle Ansätze haben jedoch eine Reihe von Nachteilen (s. *McTear* 2002). Da sie überwiegend nur die *sprachliche Struktur* von Dialogen modellieren, bleiben semantische und pragmatische Aspekte unberücksichtigt. Dies führt z. B. bei der Erkennung von Dialogakten zu Problemen bei der Auflösung von Ambiguitäten, ebenso bei der Entscheidung zwischen mehreren möglichen Dialogschritten (durch das Fehlen zugrunde liegender Kriterien). Sie sind außerdem wenig geeignet in Bereichen, die eine inhärente Variabilität der Dialoge aufweisen. Eine Ausmultiplizierung aller Möglichkeiten macht ein solches System unhandlich und schlecht wartbar. Insbesondere Endliche Automaten sind sehr inflexibel, da sie keine Abweichung vom vorherbestimmten Dialogverlauf erlauben. Sie sind a priori ungeeignet in Fällen, wo der Dialogverlauf grundsätzlich unvorhersehbar ist. Außerdem sind sie nicht mehr deklarativ Dialog modellierend, sondern verkörpern gleichzeitig Kontrollinformation, die der Dialogmanager übernehmen sollte.

In **Plan-basierten** Ansätzen werden Äußerungen in einem Dialog als sprachliche *Handlungen* aufgefasst (als kommunikative, Dialog- oder Sprechakte), die zuvor geplant werden. Die Auswahl eines nächsten Dialogschritts richtet sich demnach nicht (allein) danach, was gemäß einer Repräsentation typischer Dialogverläufe anzunehmen ist, sondern im Wesentlichen danach, welche (Sprach-)Handlung angesichts der gegenwärtigen Situation zur Verfolgung der eigenen/gemeinsamen Ziele angemessen ist.

Dem Verstehen einer Äußerung liegt analog dazu das Schlussfolgern darüber zugrunde, welches Ziel der Dialogpartner mit ihr verfolgt und welche Handlung (als Teil eines Plans zum Erreichen des Ziels) er danach mit ihr durchführt. Dies bedeutet vor allem, dass eine Plan-basierte Dialogmodellierung die Berücksichtigung verschiedener Wissensquellen (Hintergrundwissen, Dialoggeschichte, Benutzermodell etc.) beinhaltet. Es führt dazu, dass solche Ansätze gut mit Abweichungen von typischen Dialogverläufen umgehen können und gleichzeitig der Multifunktionalität sprachlicher Handlungen (s. indirekte Sprechakte) gerecht werden.

Plan-basierte Ansätze basieren auf speziellen Planern, die besonders an die Aspekte der Sprachverarbeitung angepasst sind. Hierzu gehören insbesondere die Aspekte der *hierarchischen Planung*, d.h., die Planung auf verschiedenen Ebenen und die Zergliederung von Plänen in Teilpläne mit entsprechenden Subzielen (ein Vorteil davon ist die Verkleinerung der Suchräume) und der *reaktiven Planung* (um auch mit unerwarteten Äußerungen flexibel umgehen zu können, s. *Freedman* 2000). Die formale Basis Plan-basierter und anderer wissensintensiver Ansätze ist das **BDI-Modell** (Beliefs, Desires, Intentions), das die Voraussetzungen für rationales Verhalten charakterisiert (s. Abb. 8.5 auf der nächsten Seite).⁵ Es hat die folgenden Komponenten:

- Beliefs: Annahmen des Agenten über die Welt
- Desires: Die Wünsche des Agenten
- Intentions: Die aktuellen Absichten des Agenten
- Reasoning: Nachdenken, Schließen über dem Wissen
- Deliberation: Entscheiden, welche Handlungen den Wünschen dienen

⁵ S. auch *Traum* (1996) zu Vorschlägen für eine Erweiterung dieses Basismodells.

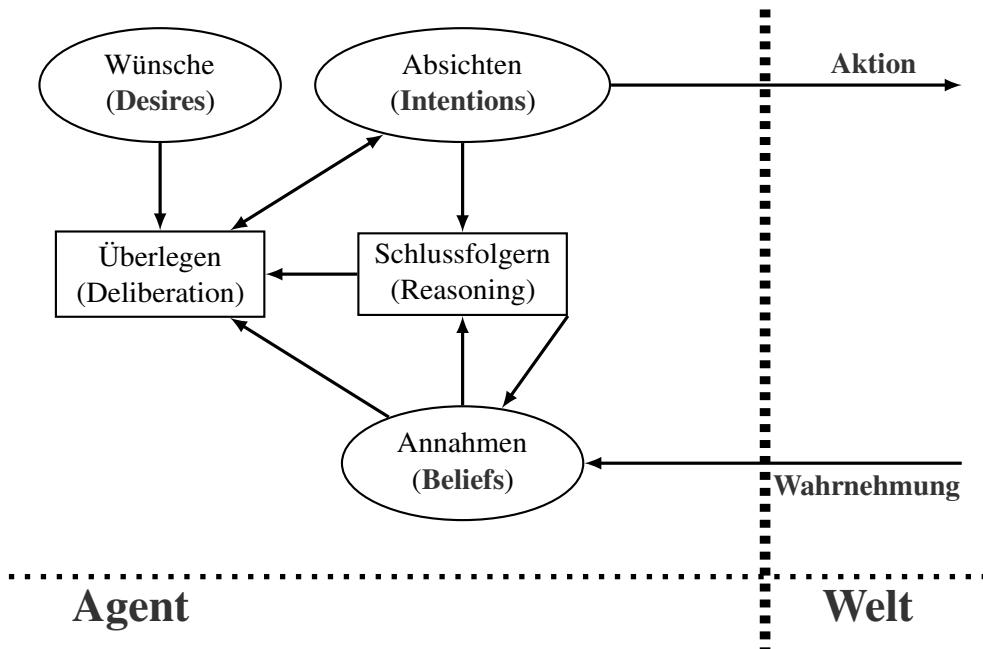


Abbildung 8.5: Das BDI-Modell

Üblicherweise wird ein System, das neben interner Informationsverarbeitung auch über die Fähigkeit zur Interaktion mit der Welt verfügt, als **Agent** bezeichnet, ein DS (bzw. dessen Dialogmanager) daher als *konversationaler Agent*.

Im Vergleich sind Strukturelle Ansätze der Dialogmodellierung gut geeignet für einfache Anwendungen, da sie sich einfach realisieren lassen und effiziente Systeme ergeben. Dafür weisen sie erhebliche Nachteile im Hinblick auf Flexibilität, Erweiterbarkeit, Generalisierbarkeit, Portabilität u.ä. auf. Plan-basierte Ansätze hingegen sind sehr viel allgemeiner und somit mächtiger, aber i.A. nicht effizient und gegenwärtig somit nicht unbedingt für kommerzielle Anwendungen geeignet. Die Entwicklung von DS beinhaltet daher grundsätzlich ein Zweck-gerichtetes Abwägen der Schwerpunktsetzung auf Robustheit oder Flexibilität.

8.4 Architekturen und Systeme

8.4.1 Konversationelle Agenten (**TRAIN**S, **TRIPS**)

Im Rahmen des **TRAIN**S-Projekts (s. Allen 1995, Ferguson et al. 1996) wurden eine Reihe von Forschungsprototypen eines Plan-basierten DS entwickelt, die als *kollaborative konversationale Agenten* (s. Abb. 8.6 auf Seite 130) fungieren sollten. Hierzu sollte das jeweilige **TRAIN**S-System dazu befähigt werden, sich mit dem Benutzer auf einen **gemeinsamen Plan (shared plan)** zu einigen, bestimmte Waren mit bestimmten Verkehrsmitteln von und zu bestimmten Orten zu bringen. Um diese Aufgabe zu erfüllen, muss das System u.a. Folgendes leisten:

Destillieren von Dialogen (nach Joensson und Dahlbäck 2000)

Das Destillieren von Dialogen dient der Gewinnung von Daten für Mensch-Maschine-Dialoge, und zwar insbesondere dann, wenn sowohl „echte“ Daten nicht erhoben werden können als auch Wizard-of-Oz-Experimente nicht durchführbar sind (z. B. ist es schwierig, einen Computer vorzutäuschen).

Daher werden Mensch-Mensch-Dialoge erhoben, bei denen insbesondere die Äußerungen des „System“-Dialogpartners modifiziert werden, um den Anschein eines echten Systems zu erzeugen (keine Wiederholungen, Pausen etc.).

Bei beiden Partnern werden „überflüssige“ Äußerungen entfernt (z. B. bei schon vorhandener oder nicht mehr gültiger Information). Ggf. werden Äußerungen hinzugefügt, um einen kohärenten Dialogverlauf zu sichern, z. B. die Frage S7 im destillierten Beispieldialog.

Der Original-Dialog:

- B4: yes I wonder if you have any mm buses or (.) like express buses leaving from Linköping to Vadstena (.) on Sunday
- S5: no the bus does not run on sundays
- B6: how can you (.) can you take the train and then change some way (.) because (.) to Mjölby 'n' so
- S7: that you can do too yes
- B8: how (.) do you have any such suggestions
- S9: yes let's see (4s) a moment (15s) now let us see here (.) was it on the sunday you should travel
- B10: yes right afternoon preferably
- S11: afternoon preferably (.) you have train from Linköping fourteen twenty nine
- B12: mm
- S13: and then you will change from Mjölby station six hundred sixty
- B14: sixhundred sixty
- S15: fifteen and ten

Der destillierte Dialog:

- B4: yes I wonder if you have any buses or (.) like express buses going from Linköping to Vadstena (.) on Sunday
- S5: no the bus does not run on sundays
- B6: how can you (.) can you take the train and then change some way (.) because (.) to Mjölby and so
- S7: when do you want to leave?
- B8: (..) afternoon preferably
- S9: you can take the train from Linköping fourteen and twenty nine and then you will change at Mjölby station to bus six hundred sixty at fifteen and ten

- den Ablauf der Aktionen evaluieren, erwartete Erfüllungszeiten kalkulieren, Planungskonflikte entdecken
- Details des vorgeschlagenen Plans ausfüllen, die nicht die Aufmerksamkeit des Benutzers erfordern
- Wege vorschlagen, auftretende Subprobleme zu lösen
- Präsentieren und Beschreiben des gegenwärtigen Plans und Zustands der Welt
- den Plan an die (simulierten) Agenten in der Welt weiterleiten und das Feedback auswerten
- die Korrekturen und Änderungen der Pläne mit dem Benutzer koordinieren, sogar während der Ausführung der Pläne

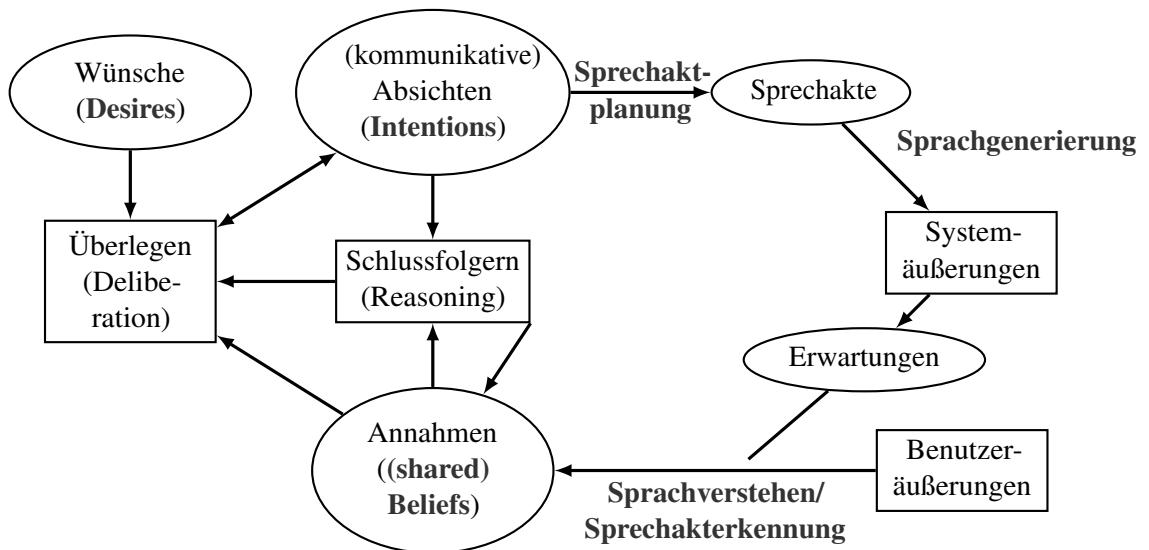


Abbildung 8.6: Das TRAINS-Modell eines konversationellen Agenten (s. Allen 1995)

Ein Beispiel des **TRAIN**S-Systems ist ein Dialog zum Erreichen eines gemeinsamen Plans für die Herstellung von Orangensaft (d.h. der dazu gehörigen Logistik). Abb. 8.7 auf der nächsten Seite zeigt diesen Dialog und den ihm zugrunde liegenden, vom System aufgebauten finalen Plan(graph). Die relevanten Ereignisse erscheinen als gerundete Rechtecke, Fakten als normale Rechtecke. Die geplanten Ereignisse sind schattiert dargestellt.

Neben den bekannten Aufgaben der Dialogsteuerung involviert das Plan-basierte Dialog Management in **TRAIN**S und seinem Nachfolger **TRIPS** insbesondere die Buchführung über den „**Mentalen Zustand**“ (*mental state*) des Agenten (seine BDIs) und über den **Diskurskontext** (mit Hilfe der Dialoggeschichte). Das Modell des mentalen Zustands enthält dazu

- geschachtelte Überzeugungen (beliefs) über die Domäne (vor allem auch die gemeinsamen Überzeugungen beider Dialogpartner, sog. **mutual beliefs**),
- eine Menge aktueller Dialogziele sowie

B: Wir sollen Orangensaft machen. Wir haben Orangen bei I und eine OS Fabrik bei B. Zugmaschine Z3 kommt voraussichtlich um 15 Uhr bei I an. Sollen wir die Orangen verfrachten?

S: Ja, soll ich anfangen, die Orangen in den leeren Wagen W1 bei I zu verladen?

B: Ja, und wir lassen ihn dann durch Z3 abholen. Ok?

S: Ok.

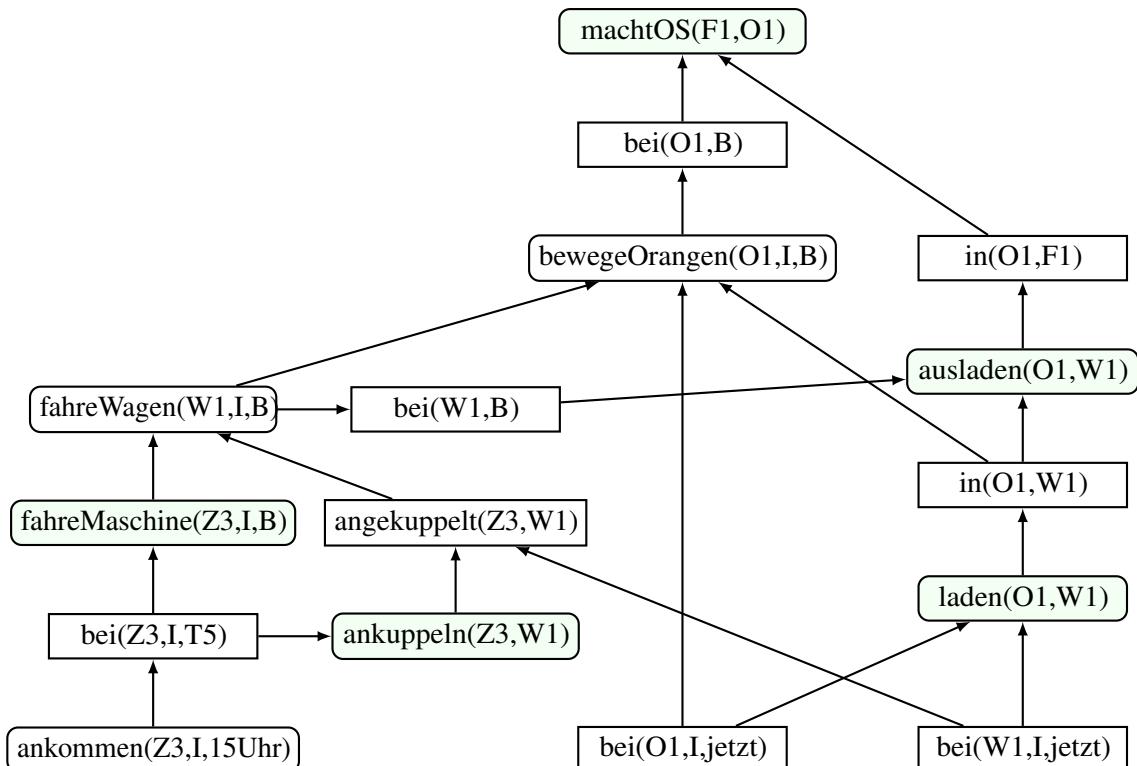


Abbildung 8.7: TRAINS-91 Dialog und zugrunde liegender Plan

- eine Menge intendierter Sprechakte.

In TRIPS enthält der **Diskurskontext** die folgenden Informationen:

- ein Modell der gegenwärtig salienten Objekte (für die Auflösung und Produktion anaphorischer Bezüge)
- die Struktur und Interpretation der unmittelbar vorangegangenen Äußerung (für die Unterstützung von Ellipsenauflösung und Klärungsfragen)
- den gegenwärtigen Stand des Turns – ob bereits einem der Dialogpartner zugewiesen oder noch offen

- die Diskursgeschichte, bestehend aus: die als Sprechakte interpretierten bisherigen Äußerungen mit dem Vermerk, welche der Äußerungen ge „grounded“ ist.
- eine Menge von **Diskursverpflichtungen** (*discourse obligations*), d.h., die Information über noch zu tätige Erwiderungen. Bei geschachtelten Dialogen können sie gestapelt sein (allerdings wird der Stapel nie groß werden).

Bei der Entscheidung, welcher **Dialogschritt** als Nächstes durchgeführt werden soll, wird nach der folgenden Prioritätsliste der Berücksichtigung relevanter Aspekte verfahren:

1. Diskursobligationen, die sich aus Sprechakteffekten ergeben (z. B. Akzeptieren oder Ablehnen eines Angebots)
2. Allgemeine Verpflichtung, den Redepartner nicht zu unterbrechen
3. Intendierte Sprechakte
4. Allgemeine Verpflichtung, geäußerten Inhalt zu „grounden“ (d.h., Sicherstellen gemeinsamer Überzeugungen durch Anerkennung oder Anfrage nach Anerkennung, oder durch Reparaturen)
5. Domänenabhängige Diskursziele (Annahme/Ablehnen von Vorschlägen; eigene Vorschläge machen oder Rückmeldung zu früheren Vorschlägen einfordern)
6. Höhere Diskursziele (Weiterführen der Konversation durch weiteres Reasoning über der Domäne)

TRAINs und **TRIPS** gehören zu den ambitioniertesten jemals entwickelten DS, die für ihre eingeschränkten Domänen eine für wissensintensive NLS erstaunliche Robustheit erzielen. Ihr wesentlicher Nachteil ist allerdings ihre Komplexität: Nicht für alle Aufgaben von DS wird der komplexe Planungs(erkennungs)apparat benötigt; Formalisten bemängeln dessen formale Grundlage und Praktiker die Inhomogenität der verwendeten Wissens(repräsentations)-strukturen (s. z. B. *Gurevych et al.* 2003) sowie die Produktivität insgesamt.

8.4.2 Rationale Interaktion (**ARTIMIS**)

Der Rationale Ansatz von Sadek (*Sadek* 1999) zeichnet sich durch zwei wesentliche Eigenschaften aus: eine „radikal“ formale Sicht auf das BDI-Modell mit Hilfe logischer Repräsentationen sowie dessen Umsetzung in ein bei der France Telekom eingesetztes angewandtes System (**ARTIMIS**, s. Abb. 8.9 auf Seite 134).⁶

Er gründet in der Annahme, dass ein zu „intelligenter“ Dialogführung fähiges System ein intelligentes System sein muss, in dem die Kommunikationsfähigkeit nicht primitiv (d.h. sozusagen von eigener Art, mit eigenen Komponenten), sondern in einer generelleren Kompetenz verankert ist, die rationales („vernünftiges“) Verhalten charakterisiert. Sadeks Ansatz basiert deshalb auf einer allgemeinen formalen logischen Beschreibung rationalen Verhaltens mit

⁶ Quelle der Grafik (mittlerweile nicht mehr erreichbar):

<http://www.francetelecom.com/sirius/rd/fr/ddm/fr/technologies/ddm200210/index1.php4>.

Theorembeweisen als Berechnungsverfahren. Er umfasst nicht nur die Dialogmodellierung selbst, sondern auch Aspekte kooperativen Dialogverhaltens. Dafür werden modale Operatoren (wie B für „belief“, I für „intention“) verwendet. Ein einfaches Beispiel ist in Abb. 8.8 gegeben.

KOMMUNIKATIVER AKT DES INFORMIERENS

$< i, \text{INFORM}(j, \phi) >$:

$B(i, \phi) \wedge \neg B(i, B(j, \phi))$ (Vorbedingung)

$B(j, \phi)$ (Rationaler Effekt)

[Ein Agent i informiert einen Agenten j über ϕ , wenn i eine Proposition ϕ glaubt und gleichzeitig nicht davon überzeugt ist, dass j ϕ glaubt. Als Effekt der Aktion glaubt j ϕ .]

Abbildung 8.8: Beispiel für den logischen Mechanismus rationaler Konversation

8.4.3 Der Information State Ansatz (TRINDI)

Der Information State-Ansatz (*Traum und Larsson 2003*) setzt einen anderen Schwerpunkt in der Architektur von DS. Zentral ist hier eine *Datenstruktur zur Repräsentation des Diskurszustands (information state)*, auf die von allen Komponenten des DS zugegriffen werden kann. Dialogmanagement wird jetzt ganz allgemein als Verwaltung dieses Information State (IS) aufgefasst, wobei die Aufgabe in dessen angemessenem **Update** besteht.

Dialogmodellierung stellt sich als die Spezifikation der Dialogschritte (**dialogue moves**) dar, deren Erkennung bzw. Auswahl in einem laufenden Dialog jeweils bestimmte Regeln zur Veränderung des IS (**update rules**) auslösen („triggern“). Ziel der Realisierung dieses Ansatzes im TRINDI-Projekt war es, eine praxisorientierte Generalisierung von Dialogsystemen und -theorien zu konzipieren und ein entsprechendes softwaretechnisches Framework zu entwickeln (TRINDIKit), das als Grundlage für spezifische DS dienen kann.

Bestandteile einer IS-basierten Dialogtheorie sind:

- eine Beschreibung informationaler Komponenten
- formale Repräsentationen dieser Komponenten
- eine Menge von Dialogschritten, die ein Update des IS triggern. Außerdem Regeln für das Erkennen und Realisieren dieser Schritte.
- eine Menge von Updateregeln, die das Update des IS regeln, inklusive einer Menge von Auswahlregeln, die die Auswahl des nächsten Dialogschritts bestimmen.
- eine Update-Strategie für die Entscheidung, welcher mögliche Dialogschritt angewandt werden soll.

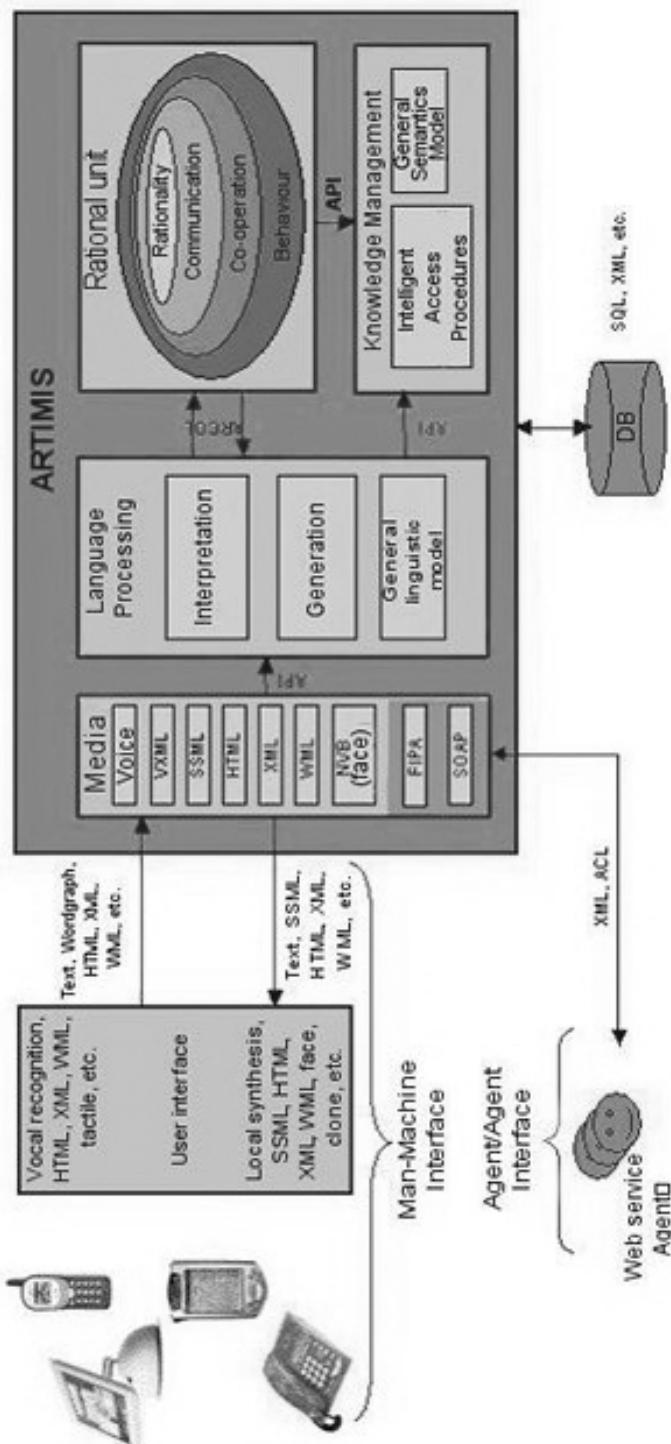


Abbildung 8.9: Die Architektur von ARTIMIS

Ein **Information State** enthält im Allgemeinen Informationen zu den folgenden Aspekten:

- Common Ground, u.a. Obligationen und Dialoggeschichte
- jetzige und vorherige Diskurseinheit
- Intentionen
- Agenda, d.h. unrealisierte Schritte des Diskursplans
- Zugang zu Ressourcen: Inferieren über dem Domänenwissen, Inhaltsplanung.

Damit ist 'Information State' ein sehr allgemeines Konzept. Entsprechend schreiben *Bos et al.*:

In this context 'information state' refers to all the information necessary to represent a dialogue, including information gained from previous utterances and motivation for future actions. In other words, it can be viewed alternatively as the 'conversational score', the 'discourse context', or 'mental state'. The actual content of an information state is not fixed – it is completely up to the developer which aspects of dialogue should be represented as part of the information state.

(*Bos et al.* 2002)

Ein stark vereinfachtes Beispiel aus der Reiseauskunftsdomäne soll die Arbeitsweise des IS-Ansatzes verdeutlichen (s. *Traum und Larsson* 2003). Hierzu werden folgende weitere Aspekte berücksichtigt (s. (8)): Die Unterscheidung in eigene (*private*) und gemeinsame (*shared*) Informationskomponenten (*AGenda* und *BELiefs*), ein Kellerspeicher (*stack*) der noch offenen Fragen (*questions under discussion, QUD*) und der letzte durchgeführte Dialogschritt (*last move, LM*). Über so spezifizierten Information States operieren Update-Regeln, von denen einige in (9)-(12) definiert sind (jeweils mit einem *BEDingungs-* und einem *EFFekt*-Teil).

$$(8) \quad \begin{array}{l} \text{PRIVATE : } \left[\begin{array}{l} \text{BEL : Menge(Propositionen)} \\ \text{AG : Stack(Aktionen)} \end{array} \right] \\ \text{SHARED : } \left[\begin{array}{l} \text{BEL : Menge(Eigenschaften)} \\ \text{QUD : Stack(Fragen)} \\ \text{LM : DIALOGUE MOVE} \end{array} \right] \end{array}$$

$$(9) \quad \begin{array}{l} \text{Update-Regel : wähleFrage} \\ \text{BED : } \left\{ \text{erster(PRIVATE.AG, aufwerfen}(F)\text{)} \right\} \\ \text{EFF : } \left\{ \text{setze(NEXT_MOVE, frage}(F)\text{)} \right\} \end{array}$$

$$(10) \quad \begin{array}{l} \text{Update-Regel : integriereSystemFrage} \\ \text{BED : } \left\{ \begin{array}{l} \text{wert(SHARED.LM, frage}(F)\text{)} \\ \text{erster(PRIVATE.AG, aufwerfen}(F)\text{)} \end{array} \right\} \\ \text{EFF : } \left\{ \begin{array}{l} \text{wert(SHARED.QUD, F)} \\ \text{pop(PRIVATE.AG)} \end{array} \right\} \end{array}$$

	<i>Update – Regel</i> : integriereBenutzerAntwort
(11)	<i>BED</i> : $\left\{ \begin{array}{l} \text{wert}(\text{SHARED}.LM, \text{antwort}(A)) \\ \text{erster}(\text{SHARED}.QUD, F) \\ \text{DOMÄNE} :: \text{relevant}(A, F) \\ \text{DOMÄNE} :: \text{reduziere}(F, A, PROP) \end{array} \right\}$
	<i>EFF</i> : $\left\{ \text{füge_hinzu}(\text{SHARED}.BEL, PROP) \right\}$

	<i>Update – Regel</i> : entferneFrageausQUD
(12)	<i>BED</i> : $\left\{ \begin{array}{l} \text{erster}(\text{SHARED}.QUD, F) \\ \text{in}(\text{SHARED}.BEL, PROP) \\ \text{DOMÄNE} :: \text{istAntwortauf}(PROP, F) \end{array} \right\}$
	<i>EFF</i> : $\left\{ \text{pop}(\text{SHARED}.QUD) \right\}$

In dem Beispiel hat das System einige Fragen, die als in der Agenda befindliche „Frage aufwerfen“-Aktionen repräsentiert sind (s. (13)). Als erster Schritt wird eine dieser Aktionen ausgewählt. Dies ergibt sich aus der Anwendbarkeit der Regel **wähleFrage** in (9).

	<i>PRIVATE</i> : $\left[\begin{array}{l} BEL : \{\} \\ AG : \langle \text{aufwerfen}(\lambda x. \text{zielort}(x)), \text{aufwerfen}(\dots), \dots \rangle \end{array} \right]$
(13)	<i>SHARED</i> : $\left[\begin{array}{l} BEL : \{\} \\ QUD : \langle \rangle \\ LM : \dots \end{array} \right]$

Der Effekt der Regelanwendung ist, dass die Frage nach dem Zielort als nächster Dialogschritt festgelegt wird. Der semantische Gehalt F wird an die Generierungskomponente weitergereicht und es wird z. B. die Frage *Wohin wollen Sie fahren?* ausgegeben. Das Stellen der Frage an den Benutzer wird als gemeinsam bekannter *last move* eingetragen. Damit kann die Regel (10) (**integriereSystemFrage**) feuern, durch die F als question under discussion notiert und aus der Agenda entfernt wird. Dies ergibt den IS (14).

	<i>PRIVATE</i> : $\left[\begin{array}{l} BEL : \{\} \\ AG : \langle \text{aufwerfen}(\lambda x. \text{abfahrtsort}(x)), \dots \rangle \end{array} \right]$
(14)	<i>SHARED</i> : $\left[\begin{array}{l} BEL : \{\} \\ QUD : \langle \lambda x. \text{zielort}(x) \rangle \\ LM : \text{frage}(\lambda x. \text{zielort}(x)) \end{array} \right]$

Die Antwort des Benutzers (z. B.: *nach Hamburg*) wird analysiert, der Zielort extrahiert und die Antwort als *last move* notiert. Jetzt feuert zuerst die Regel (11) (**integriereBenutzerAntwort**), die die Relevanz der Antwort für die Frage abprüft, den propositionalen Gehalt der Antwort extrahiert und schließlich zum gemeinsamen Wissen hinzufügt. Regel (12) (**entferneFrageausQUD**) schließt diesen Mini-Dialog ab, indem es die Frage aus dem Stack der noch offenen Fragen entfernt – dies aber erst, nachdem abgeprüft wurde, ob durch die Antwort die Frage zufriedenstellend geklärt wurde. (15) zeigt den hieraus resultierenden IS.

$$(15) \quad \begin{array}{l} \text{PRIVATE : } \left[\begin{array}{l} BEL : \{ \} \\ AG : \langle \textit{aufwerfen}(\lambda x. \textit{abfahrtsort}(x)), \dots \rangle \end{array} \right] \\ \text{SHARED : } \left[\begin{array}{l} BEL : \{ \textit{zielort(Hamburg)} \} \\ QUD : \langle \rangle \\ LM : \textit{antwort(Hamburg)} \end{array} \right] \end{array}$$

Abb. 8.10 veranschaulicht die Architektur des IS-basierten DS (angelehnt an die des **MAGICSTER**-Systems, s. Bos et al. 2002) mit ausgewählten Informationskomponenten und Schnittstellen. Sie zeigt den Information state als Struktur, die zwischen den Prozesskomponenten und den Ressourcen/ der Wissensbasis vermittelt. Gesteuert wird das System durch eine Kontrollkomponente, wodurch der Prozessfluss flexibler gestaltet wird.

Damit stellen sich IS-basierte DS als *Blackboard-Systeme* dar, in denen verschiedene Komponenten auf dieselben Datenstrukturen zugreifen können. IS-basierte DS dieser Art lassen sich z. B. mit dem im TRINDI-Projekt entwickelten Framework TRINDIKIT erstellen.

8.5 Literatur

Churcher (1997) ist ein Forschungsreport, in dem ein guter Überblick über die verschiedenen Aspekte von Dialogen und ihrer Modellierung in Dialogsystemen gegeben wird. *Wachsmuth* (2005) ist ein sehr verständlicher Überblick über den Einsatz des BDI-Modells in der Dialog-Technologie am Beispiel eines virtuell-anthropomorphen multimodalen (s. Kapitel 10) Dialogsystems, sog. „situierter künstlicher Kommunikator“.

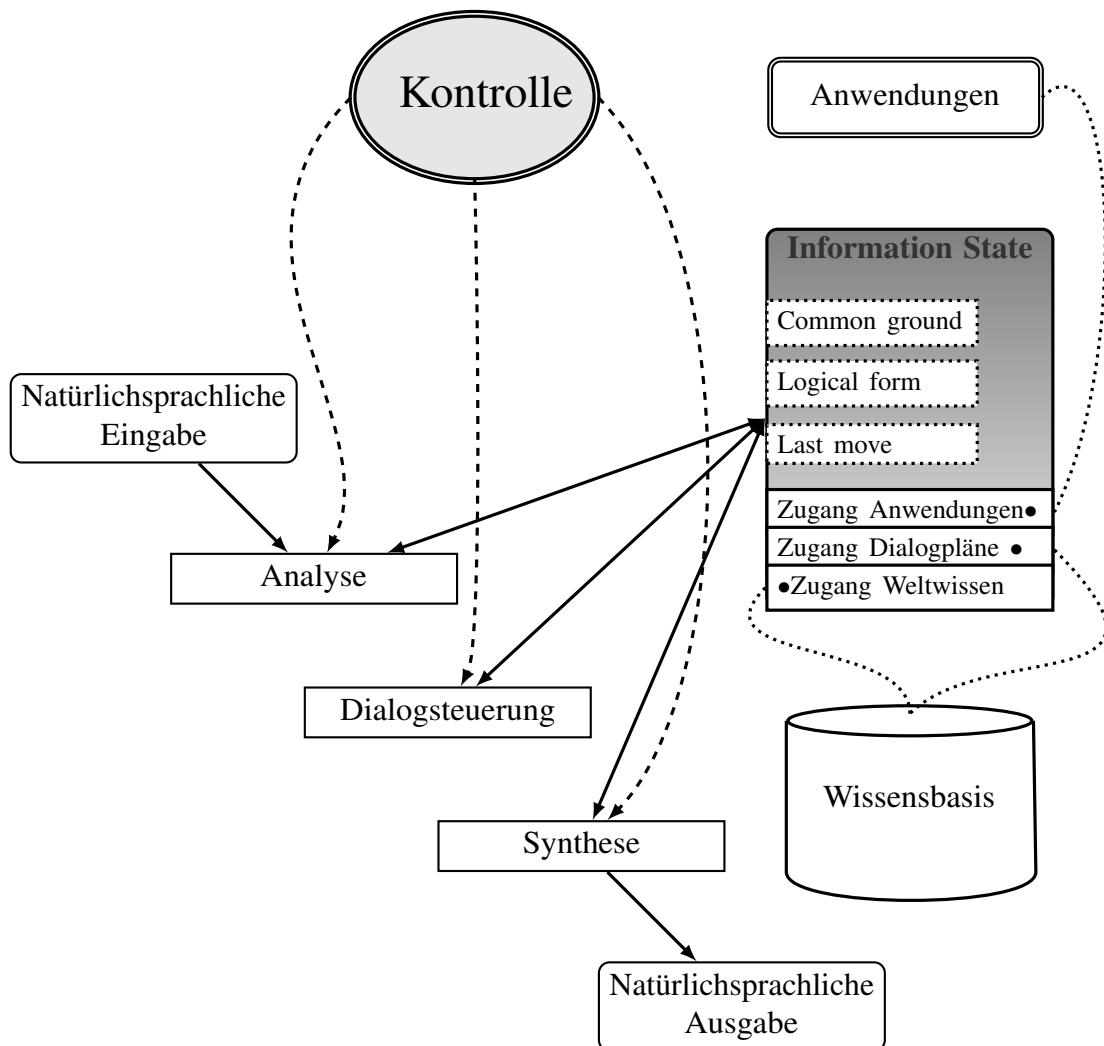


Abbildung 8.10: Architektur IS-basierter DS

Kapitel 9

Sprachdialogsysteme (SDS)

Computer ... Computer? ... Hello, Computer? ...

(Scotty in *Star Trek IV* im Dialog mit einem MacPlus)

„Wenn Sie mit der technischen Abteilung verbunden werden wollen, drücken Sie die 1... Wenn Sie weitere Information wünschen, sagen Sie 'Ja'“. Solche Beispiele automatisierter telefonischer Interaktion (**Interactive Voice Response (IVR)**) sind seit Jahren Standard im Kundendienst großer Firmen, meist auf der Basis spezifischer, speziell auf ihre Rechnerplattform zugeschnittener Softwaresysteme.

Sie werden traditionell dazu eingesetzt, häufig auftretende telefonische Interaktion zu automatisieren (Vorsortierung: Was will der Kunde?) und so Kosten (für menschliche Interaktion) zu sparen.

Sprachdialogsysteme (SDS) (auch: Sprachanwendungen bzw. -applikationen, Sprach- bzw. Voice Portale) ersetzen die meist starre Menüführung des Dialogs zunehmend durch adaptive Dialogtechnologie und verwenden gleichzeitig vorwiegend^a Spracherkennung und Sprachsynthese jeweils für die Verarbeitung von Input und Output. SDS bieten durch moderne Informationsdienste zunehmend einen Mehrwert gegenüber reiner Kostensparnis durch Automatisierung. Der moderne Einsatz eines SDS im Kontext entsprechender IT/TK/Web-Infrastrukturen und Applikationsanbindungen für ein spezifisches Geschäftsmodell wird üblicherweise **Sprachportal** genannt.

In Analogie an grafische Benutzerschnittstellen spricht man im Zusammenhang mit den Benutzeroberflächen der SDS auch von **Voice User Interfaces** (VUIs), von ihren charakteristischen Eigenschaften als ihrem **Hear and Feel** ihrer individuellen Gestaltung („**Persona**“, s. auch Kap. 10) sowie von ihren Sounds mit Wiedererkennungswert als **Earcons**.

^a Allgemeinaussagen müssen hier grundsätzlich abgeschwächt werden: Nicht nur wurden bei IVR-Systemen zum Teil schon Spracherkennung und -synthese verwendet, auch wird in modernen SDS teilweise immer noch/wieder mit aufgenommenen Stimmen gearbeitet.

9.1 Allgemeines zu SDS

Mit der Reifung der Sprachtechnologie vollzieht sich ein kontinuierlicher Wechsel von proprietären IVR-Systemen über IVR-Systeme, die Sprachtechnologie-Standards (wie VoiceXML¹) verwenden, zu SDS. Laut einer Studie des Marktanalysten Datamonitor werden sich die Einnahmen anhand proprietärer IVR-Systeme in Nordamerika und Europa/Nah-Ost/Afrika (EMEA) von 2005 bis 2009 um 35% auf 179 Mio Dollar reduzieren, die anhand Standard-basierter Plattformen auf 332 Mio Dollar nahezu verdoppeln (s. *Kurpinski 2005*).

Was SDS angeht, so existiert in Deutschland seit 2004 mit den jährlich stattfindenden Voice Awards und den damit verbundenen Evaluationen (*Steimel et al. 2005, Steimel et al. 2006*) erstmals eine Einrichtung, mit der die vielfältigen Aspekte moderner Sprachanwendungen quantifiziert und bewertet werden können. Die Kernaussage der beiden ersten Studien, auf die im Folgenden näher eingegangen wird, ist eindeutig: SDS sind in marktwirtschaftlicher Hinsicht „reif für den Mittelstand“, d.h. sowohl kostengünstig zu entwickeln als auch kostendeckend zu betreiben.

Vorteile von SDS gegenüber (touch-tone) IVR-Systemen (s. hierzu auch *Cohen et al. 2004, S. 10*) sind **System-seitig** die **Fähigkeit zur Erledigung komplexerer Aufgaben**, nämlich

- Benutzerweiterleitung (z. B. in Call-Centern) über natürliche Sprache
- Auskunft (Flug-, Bahn-, Kino- etc.)
- Transaktionen (z. B. Bankkonto-bezogene Aufgaben)
- Stimmverifizierung (Möglichkeit zur telefonischen Änderung von Passwörtern etc.);

Benutzer-seitig die **Intuitivität und größere Flexibilität der Dialogführung** und die daraus resultierende **Effizienz und größere Kundenfreundlichkeit**;

Anbieter-/Entwickler-seitig aufgrund modernerer Technologie die **schnellere Entwicklungszeit**, die zu einer kürzeren ROI (Return On Invest)-Spanne führen kann; außerdem der möglicherweise **höhere Grad der Automatisierung von Kundenanfragen**, der insgesamt zu einer **Reduzierung der Kosten** für die Kundenbetreuung führen kann; schließlich ein für die Kundenbindung wichtiges **erweitertes Service-Angebot** durch verbesserte System-Kompetenz.

Nachteile von SDS sind überwiegend mit einem „noch“ zu versehen, da sie von aktuellen Entwicklungen zunehmend gemildert werden (können). Hierzu gehört insbesondere die **Unausgereiftheit der Technologie** (z. B. bei der Spracherkennung, bei der Skalierbarkeit der Systeme, bei der Integration in bestehende Abläufe) und die daraus resultierende Unsicherheit der Investition und der Gewährleistung des Kundenvertrauens. Eine Auswertung der Einwände gegen SDS bei skeptisch eingestellten Unternehmen (s. *Klemann und Steimel 2004:20*) zeigt jedoch, dass Aspekte wie fehlendes Vertrauen in die Technik oder zu erwartende technische

¹ VoiceXML ist eine Markup-Sprache zur abstrakten Beschreibung einer Dialogstruktur (s. z. B. *McTear 2004*) bzw. ein mittlerweile vom W3C empfohlener *freier* Standard zur Kodierung von „Voice“-Anwendungen.

Integrationsprobleme eher nachrangig sind (insgesamt unter 10%).²

Laut *Klemann und Steimel* (2004, S. 15) sind die folgenden Technologien/Technologie-Anbieter an der Entwicklung von SDS beteiligt (hieran zeigt sich auch die Randlage der SDS in der Computerlinguistik/ Sprachtechnologie):

1. Basistechnologie: Entwicklung von Hardware (Server etc.) und Software (ASR, TTS, Stimmverifizierung)
2. Sprachdialog-Plattform: Proprietäre IVR-Systeme, IVR-Systeme mit VoiceXML-Schnittstelle oder reine VoiceXML-Plattformen
3. Sprachapplikationsentwicklungs- und -management-Software
4. Applikationsentwicklung und -integration: High-level- (z. B. Dialog Design) und Low-level- (Integration in bestehende IT-Umgebungen) Softwareentwicklung
5. Sprachapplikations-Dienstanbieter: Bereitstellen von Sprachapplikationen ohne weitere Dienstleistungen (Hosting) bzw. mit Mehrwertdiensten (full service hosting)

Sprachtechnologie im engeren Sinn kommt demnach vor allem in den Punkten 1 und 4 zum Einsatz. Allerdings vollzieht sich gegenwärtig ein Trend weg von technischen Fragen hin zur Kunden-bezogenen **Gebrauchstauglichkeit** (usability) der Technologie und ihrer Anwendungen. Entsprechend werden high-level-Aspekte wie Dialog Design und das Hear&Feel einer Sprachanwendung eine zunehmend größere Rolle spielen. Andererseits führen technische Reifung, Standards und Modularisierung dazu, dass technische Lösungen relativ einfach zusammengesteckt werden können, so dass auf die reine Anwendungsentwicklung fokussiert werden kann, mit der eine Geschäftsidee realisiert werden soll.

9.2 Geschichte

Meilensteine der Entwicklung von Sprachapplikationen sind die folgenden³:

1911:	„Radio Rex“: Spielzeug, das auf Zuruf reagiert
1939:	erster Speech synthesizer („Voder“ – Voice Operating Demonstrator), von Homer Dudley bei Bell Labs entwickelt, wird auf der Weltausstellung in New York vorgestellt
1952:	kleiner Worterkennner für telefonisch übermittelte Ziffern (Bell Labs)
1968:	erster Text-to-speech synthesizer für Englisch von Noriko Umeda (Electro-technical Lab., Japan)
ab ca. 1970:	erste IVR-Systeme (Touch-tone)
1980er:	Kommerzialisierung von Spracherkennung und -synthese

² Schwerer wiegen die befürchtete mangelnde Kundenakzeptanz (ca. 13%) oder schlicht andersartige Firmenphilosophie (ca. 23%) als Ablehnungsgründe.

³ Quellen: *Klemann und Steimel* (2004); *Hong* (2006); *Janet Baker* (2005), „Milestones in Speech Technology – Past and Future!“ ; *Lemmetty* (1999), Kap. 2.

1990:	Dragon Systems entwickeln erstes domänenunspezifisches Spracherkennungs-/Diktiersystem (Einzelworterkennung)
1997:	Dragon Systems entwickeln erstes Diktiersystem für kontinuierliche Sprache
2000:	TellME (später von Google aufgekauft) präsentieren das weltweit erste Voice Portal
seit 2004:	jährlich stattfindende VOICE Awards der VOICE Community

Entsprechend dem erwähnten Wechsel lässt sich die jüngere Entwicklung in diesem Bereich nach *Acomb et al.* (2007) grob in drei Phasen einteilen, die durch in der folgenden Tabelle aufgeführten Merkmale charakterisiert sind.

Phase	I	II	III
Zeitperiode	1994-2001	2000-2005	2004-heute
Fähigkeit (Art der Anwendung)	Auskunft	Transaktionen	Problemlösung
Beispiele	Package Tracking, Flugstatus	Bankverkehr, Börsenhandel, Zugreservierung	Kundenbetreuung, Technischer Support, Help Desk
Architektur	propriétär	VoiceXML (statisch)	VoiceXML (dynamisch)
Komplexität (Anzahl an Dialogmodulen)	10	100	1000
Interaktions-Turns	wenige	10	10-100
Interaktionsmodalität	geleitet	geleitet + Erkennung der User-Eingaben	Erkennung der User-Eingaben + beschränkter mixed-initiative Dialog

9.3 Aspekte und Anforderungen

9.3.1 Allgemeines

SDS konstituieren einen relativ jungen und sehr dynamischen Bereich der Kommunikationstechnologie (s. die in *Steimel und Klemann 2004* und *Klemann und Steimel 2004* dokumentierten Marktstudien). Mit den seit 2004 jährlich stattfindenden **VOICE Awards** wird einerseits dessen Breite und Entwicklung durch die teilnehmenden Systeme direkt wiedergegeben, gleichzeitig findet im Rahmen des Wettbewerbs eine Evaluation⁴ im Hinblick auf die Usability der Systeme

⁴ Anders als bei den Evaluationskonferenzen der Computerlinguistik (MUC etc.) ist die Teilnahme aufgabenunspezifisch. Ein diesen Konferenzen ähnlicher Wettbewerb war der *VOICE Contest*, bei dem es mittlerweile allerdings auch keine einheitliche Aufgabenstellung mehr gibt.

statt.⁵ Die folgenden Ausführungen beziehen sich auf die Ergebnisse dieser Evaluationen (s. *Steimel et al.* 2005 und *Steimel et al.* 2006).

Bewertet wurden 2005 35 und 2006 42 (2007: 28) Systeme. Sie entstammen hauptsächlich den Branchen *Telekommunikation* (ca. 18%), *Transport, Logistik, Verkehr* (ca. 18%) und *Medien und Unterhaltung* (ca. 17%). Der Zweck der Systeme ist zu mehr als 70% die Automatisierung telefonischer Abläufe, insbesondere im Bereich Auskunft, Service und Anrufweiterleitung. Komplementär dazu gibt es zunehmend Systeme, die zusätzlich zu basalen IT-Leistungen angeboten werden (*Mehrwertdienste*), deren primärer Zweck bereits die Wertschöpfung (vulgo: Geld verdienen) ist.

Nach den Autoren der Studien sind die folgenden Aspekte für die Charakterisierung (und den Vergleich) von SDS relevant: Die technischen Leistungsmerkmale eines SDS, seine Wirtschaftlichkeit sowie die Bedienbarkeit des Voice User Interfaces.

Radio Rex

Das erste und sogar kommerziell erfolgreiche Gerät mit eingebauter Spracherkennung wurde bereits 1911 verkauft. Es handelte sich um einen Spielzeughund namens „Rex“, der aus seiner Hütte gesprungen kam, wenn man seinen Namen rief.

Diese Ein-Wort-Erkennung basierte allerdings allein auf einfacher Mechanik:^a Ein Magnet hielt den Hund durch einen mit einer Sprungfeder versehenen Hebel in der Hütte. In dem Stromkreislauf, der den Magneten versorgte, war eine kleine metallene Brücke angebracht. Sie war so beschaffen, dass sie bei einer bestimmten Stimmfrequenz (die dem Vokal in „Rex“ entsprach) in Resonanz geriet und somit den Kreislauf unterbrach. Darauf löste sich der Hebel und der Hund wurde durch die Feder aus der Hütte geworfen.

^a Mechanische Ansätze zur Sprachsynthese existierten bereits früher, s. *Lemmetty* (1999).

9.3.2 Technische Leistungsmerkmale

Anhand der **Anzahl der Ports** eines SDS lässt sich dessen Leistungsfähigkeit relativ zu seiner Funktionalität ablesen: Je mehr Ports, desto mehr Personen können das System gleichzeitig nutzen. Es ist davon auszugehen, dass zur Zeit eher Systeme mit einfacherer Funktionalität über viele Ports verfügen (im „High Volume/Low Value“-Bereich), während bei moderneren kleinen Systemen einerseits die Funktionalität komplexer und/oder das erwartete Kundenaufkommen geringer, andererseits aber auch möglicherweise die Skalierbarkeit unklarer ist.

Quantitative Aspekte der Nutzung lassen sich anhand der **Anrufe pro Tag** (2006: Ø ca. 1200) sowie anhand der **Anzahl 'Unique User'** (Personen, die das SDS mindestens

⁵ unter Federführung des Evaluationszentrums für sprachtechnologische Systeme am DFKI (Prof. Anthony Jameson).

einmal benutzt haben; 2006: Ø ca. 45000) ablesen.

Die **Integrationstiefe** bezeichnet die Anzahl der Schnittstellen eines SDS zu anderen Systemen. Dies korreliert mit dem Grad der Modularität des Systems bzw. der Flexibilität seiner Nutzung. Hier ist ein klarer Trend festzustellen: Hatten 2005 bereits 89% der Systeme mehr als 2 Schnittstellen, so waren es 2006 bereits 100%.⁶

Die sprachtechnologische Qualität eines SDS wird in der verwendeten **Technik der Spracherkennung** reflektiert. Folgende Techniken werden unterschieden:

- **Einzelworterkennung:** Hier werden nur einzelne oder jeweils durch Pausen getrennte Wörter erkennt, die üblicherweise vom System vorgegeben werden („Wenn Sie ..., dann sagen Sie ‚Ja‘“)
- Bei **Verbundworterkennung** ist kontinuierliche Sprache (ohne Pausen zwischen den Wörtern) erlaubt. Einzelne (Schlüssel-)Wörter werden also aus dem analysierten Sprachinput isoliert.
- Die **Multislot-/Mehrsatz-Erkennung** entspricht der Informationsextraktion im Speech-Bereich. Sie bedeutet einen erhöhten Komfort, da mit einer Äußerung mehrerer Aspekte gleichzeitig angegeben werden können. Dazu werden bestimmte Wörter der Eingabe den Einträgen/Slots vorgegebener Muster zugeordnet. Ein bekanntes Beispiel ist „Ich möchte heute von Aachen nach Berlin fahren“, wo gleichzeitig die Slots „Datum“, „Abfahrtsort“ und „Zielort“ gefüllt werden.
- **Natürliche Spracherkennung/Verstehen natürlicher Sprache (NLU, natural language understanding)**⁷ erlaubt die Eingabe einfacher ganzer Sätze, wobei die relevante Information mithilfe semantischer/pragmatischer Interpretation erkannt und es entsprechend darauf reagiert wird. Beispielsweise würde die Frage „Kann ich mit dem Chef sprechen?“ zur Verbindung mit der entsprechenden Person (ggf. der Sekretärin der Person) führen.

Der Großteil der betrachteten Systeme verfügt zur Zeit über Verbundworterkennung (2005: 49%, 2006: 58%), Einzelworterkennung ist von 17% auf 11% zurückgegangen (der Rest verwendet die anderen Verfahren, mit einem scheinbar deutlichen Zuwachs von NLU).

9.3.3 Wirtschaftlichkeit

Die Wirtschaftlichkeit eines SDS lässt sich anhand der Faktoren *Aufwand, Kosten, Ersparnis und Erfolg* bemessen. Nimmt man die **Realisierungsdauer** eines Systems als Maß für den Aufwand, so ist dieser mit durchschnittlich 5-6 Monaten moderat. Dabei liegt der Anteil von Systemen, die (insbesondere auch mithilfe zusammensteckbarer Komponenten) innerhalb eines Monats erstellt werden, bereits bei 16%.

⁶ In der *Steimel und Klemann* (2004) zugrundeliegenden Studie von 2003 waren es nur 20%.

⁷ Nicht zu verwechseln mit „Sprachverstehen“, das mehr umfasst als zusätzliche semantische und pragmatische Regeln.

Bei den Kosten unterscheidet man **Investitionskosten**, **Betriebskosten** und **Kosten pro Anrufer**. Im Vergleich fällt auf, dass die Investitionskosten zwar gestiegen sind (von Ø 92000 EUR auf 142000 EUR), dass aber die Betriebskosten erheblich zurückgegangen sind (von Ø 533 EUR/Monat auf 80 EUR/Monat). Dabei liegt der Anteil der Systeme mit einem mtl. Betriebskostenaufwand von 1–100 EUR 2006 bei 59%. Die **Amortisierungsdauer (ROI)** liegt durchschnittlich bei 9-10 Monaten.

Die **Ersparnis** eines SDS zeigt sich nach Amortisation der Investitionskosten im Vergleich zum nicht-automatisierten Betrieb. Hier sind neben den Betriebskosten die Kosten pro Anrufer relevant, die mit Ø ca. 0,2 EUR/Anruf sehr niedrig liegen. Die durchschnittliche Ersparnis pro System hat sich daher von ca. 110000 EUR in 2005 auf ca. 210000 EUR in 2006 erhöht.

Der Erfolg lässt sich an der **Abschlussrate** ablesen, d.h. an dem Anteil der Anrufe, die erfolgreich durch das SDS abgewickelt wurden. Sie liegt zufriedenstellend hoch (2005:90%, 2006:86%).

9.3.4 Voice User Interface

Das VUI entscheidet oft über die Zufriedenheit des Kunden und somit langfristig auch über den Erfolg eines SDS. Im Rahmen des VOICE Awards werden vier Bedienungsfaktoren unterschieden: *Verständlichkeit/Erlernbarkeit, Effektivität der Bedienung, Fehler und Fehlerbehandlung sowie das Hear & Feel* des jeweiligen Systems.

Verständlichkeit/Erlernbarkeit bedeutet, dass dem Benutzer maximal klar sein sollte, was das System gesagt hat (Verständlichkeit der Systemäußerungen), was der Benutzer als Nächstes zu tun hat bzw. was das System überhaupt in der Lage ist zu verstehen (die Klarheit über erwartete Benutzereingaben) und wo im gesamten Ablauf einer Transaktion er sich gerade befindet (die Orientierung des Nutzers). Außerdem sollte das System keine unerwarteten Reaktionen zeigen (Natürlichkeit der Abfolgen im Dialog), und im Fall von Problemen sollte angemessene Hilfe(stellung) angeboten werden (Adäquatheit von Hilfe und zusätzlichen Informationen). Gerade weil Information in einem SDS linear-sprachlich dargeboten wird, existiert hier Potential für Verwirrung und Überforderung des Nutzers.⁸

Zur **Effektivität der Bedienung** gehört zunächst die Frage, ob überhaupt alle Ziele des Benutzers erfüllt werden können (Abdeckung von Benutzerzielen; z. B. mehr als eine Bestellung ermöglichen in einem Bestellsystem) und wenn ja, wieviel Zeit eine Transaktion in Anspruch nimmt (Zeitaufwand). Außerdem bietet sich ein weites Spektrum an Möglichkeiten, dem Benutzer eine aktive Rolle im Dialog zuzulassen (Steuerbarkeit), z. B. für Unterbrechungen/Barge-In, Abkürzungen oder Abbruch.

Fehler und Fehlerbehandlung charakterisiert sowohl die Anzahl und Art der Fehler des Systems wie die Art und Weise, mit diesen umzugehen. Gerade weil die Spracherkennung immer noch grundlegende Probleme bereitet, kommt flexiblen Strategien der Fehlerbehandlung, wie sie in einzelnen Systemen Anwendung finden, eine wesentliche Bedeutung zu.

⁸ Allerdings gibt es Ansätze zu multimedialer Kommunikation, also z. B. gleichzeitig Sprachtelefonie und Informationspräsentation per Internet. Zu multimedialer Kommunikation s. außerdem Kap. 10.

Das **Hear & Feel** eines SDS durch die Akustik der Sprachausgabe (und somit der Verständlichkeit), der stimmlichen und sprachlichen Eigenschaften der System-Persona (oder ggf. verschiedener Personae) sowie die Art bzw. Variation der Begrüßungsprompts bestimmt.

Bei den stimmlichen Eigenschaften bewirkt die Natürlichkeit aufgenommener Stimmen deren häufigen Einsatz, so dass Sprachsynthese dann nur für die hochgradig variablen Textteile verwendet wird (z. B. für Werte aus der Datenbank). Allerdings kann dies offenbar die Konsistenz des stimmlichen Eindrucks beeinträchtigen.

Bzgl. der sprachlichen Eigenschaften unterscheiden sich die Systeme zum Teil stark in der zielgruppenspezifischen Anpassung, z.B durch altersgerechte Wortwahl, regionale Färbung oder sogar Benutzersprache-Erkennung (bei mehrsprachigen Systemen). Dies kann die Sympathizität des SDS erhöhen und hat teilweise hohen Motivations- bzw. Unterhaltungswert (z. B. bei einem System mit aufgenommenen Äußerungen von Lukas Podolski oder der Synchronstimme von Angelina Jolie).

(Akzeptanz-)Probleme können z. B. dann entstehen, wenn ein Benutzer nicht der avisierten Zielgruppe entspricht („Sie wolle ihr Passwort ändere, Herr Bhramrath Singh, da müsset Sie eifach . . .“), oder wenn stimmliche und sprachliche Eigenschaften nicht zueinander passen.

9.4 Evaluation

Wie lassen sich SDS bewerten? Für den VOICE Award wurde ein dreistufiges Verfahren eingeführt, das sich aus einem **Vorinterview mit dem Betreiber**, einem **Anwendungstest** durch Laien und Experten sowie aus einem **Experturteil**. Das Betreiberinterview ist mit 35% (Anteil am Gesamturteil) gewichtet, dabei entfallen auf die dort erhobenen Informationen zu den technischen Leistungsmerkmalen und Wirtschaftlichkeit 15 bzw. 20%. Auf die Beurteilung des VUI durch die Laien und Experten entfallen je 25%. Damit bleiben 15% für das Experturteil über die Innovativität des SDS, d.h. inwieweit auf der technischen Ebene, in der Benutzerschnittstelle oder dem Geschäftsmodell neue Konzepte überzeugend eingeführt wurden.

Die Ergebnisse der Bewertung führen zu einem Ranking, anhand dessen verschiedene Preise vergeben werden. Wichtig hierbei ist, dass das Ranking selbst (und somit der Rang eines Systems) nicht veröffentlicht wird (da es möglicherweise Betreiber davon abhalten würde, mit ihren Systemen am Wettbewerb teilzunehmen).

9.5 Ausblick

Die folgenden Anwendungen sind Beispiele existierender sprachtechnologisch interessanter SDS und geben somit einen Eindruck von dem Potential der Technologie:

- Flugauskunft
- Fahrplanauskunft zu städtischem Verkehrsverbund
- Kinoinformationssystem

- automatische Taxizentrale
- Telebanking im Girobereich mit der Möglichkeit, Kontostandsabfrage, Daueraufträge, Überweisungen, Saldo-Umsatzabfragen, Kartensperrungen u.a. durchzuführen
- Tele-Brokerage mit Kauf, Verkauf, Orderänderung und -löschung, Bestandsabfrage, Orderstatus, Buying-Power, automatischer Plausibilitätsprüfung und Limitbedingungen
- automatische Telefonzentrale und sprachgesteuertes Kundenserviceportal einer Mietwagenvermittlung
- mehrsprachiges Infortainment-Sprachportal (Wetter, News, Sport, Horoskop usw.)
- telefonisches Bürgeramt (Auskunft zu den Bürgerbüros, Informationen zu KFZ-Zulassungen, Ausweisen, Meldewesen und Führungszeugnissen, Lohnsteuerkarten etc.)
- die stimmliche Steuerung (per Telefon) der Flugbahn eines Balles (im Internet) als Werbung für multimediales IT-Angebot

9.6 Literatur

McTear (2002) gibt einen umfassenden Überblick über SDS aus eher akademischer Perspektive (s. auch *McTear* 2004). Eine sehr schöne high-level Einführung in die Oberflächen-bezogene Entwicklung von SDS ist *Cohen et al.* (2004). *Hoffmeister et al.* (2007) bietet einen aktuellen Überblick zu und eine umfassende Darstellung von Sprachportalen und enthält außerdem ein sehr ausführliches Glossar.

Kapitel 10

Multimodale Systeme (MMS)

Dave Bowman: *I don't know what you're talking about, HAL?*
HAL: *I know you and Frank were planning to disconnect me, and I'm afraid that's something I cannot allow to happen.*
Dave Bowman: *Where the hell'd you get that idea, HAL?*
HAL: *Dave, although you took thorough precautions in the pod against my hearing you, I could see your lips move.*

(aus: *2001: A Space Odyssey*)

Das folgende, zugegebenermaßen noch utopische Szenario^a (ein Benutzer hält seinen Persönlichen Intelligenten Assistenten –PIA– vor sich), deutet an, was unter einem **Multimodalen System (MMS)** zu verstehen ist:

Ben: <spricht> Gibt's was Neues?
PIA: <spricht, mit Cate Blanchetts Synchronstimme> Man hat in Vaduz in Liechtenstein einen Goldschatz im Wert von 500Mio Euro gefunden.
Ben: Wo ist das denn?
PIA: <zeigt Vaduz auf einer Google-map mit konzentrisch pulsierenden roten Kreisen> hier
Ben: Aha. Da war doch auch dieser Steuerskandal?
PIA: <kurze Pause>Ja. <längere Pause> Ich soll dich d
Ben: <unterbricht>Ist das etwa Nazigold?
PIA: <kurze Pause> Ja. Willst du dir den Beitrag auf N-TV anschauen?
Ben: m-mm<schüttelt den Kopf>
PIA: Ok. Ich soll dich daran erinnern, dass heute die Oscar-Nacht ist. Sie wird auf 2 Sendern live übertragen.
Ben: Wo?
PIA: <zeigt entsprechende Programmübersichtsauswahl>
Ben: Nimm <berührt den Pay-TV-Sender-Eintrag> das auf.
PIA: <schickt per WLAN an das Home-Media-Center> Ok.

^a s. auch z. B. *Alexandersson et al. (2006)* und das Leitprojekt EMBASSI.

Ein MMS ist zunächst einmal **multimedial**, indem es mit verschiedenen Medien und Formaten operiert/interagiert (Audio, Video, Grafik, Text).

Multimodalität bezeichnet die zusätzliche Fähigkeit, mit den verschiedenen Wahrnehmungs- (Hören, Sehen, Fühlen, Riechen) und Handlungsarten (Sprechen, Berühren, Gestik, Mimik), die bei Kommunikation eingesetzt werden, umgehen zu können.

Ein **Multimodales System** ist daher im Idealfall ein Dialogsystem mit multimedialen Eigenschaften/Fähigkeiten, das multimodalen Input im Hinblick auf ausgedrückte Codes^a analysieren und entsprechend multimodalen Output produzieren kann.

^a **Codes** sind verschiedene Arten, Information zu repräsentieren: Symbolsysteme für Zeichen-, „Sprachen“, Bild-, „Sprachen“ und Gesten-, „Sprachen“. Bestimmte Codes können dann in verschiedenen Modalitäten (und schließlich Medien) dargestellt werden. Beispiele sind natürliche Sprachen als Code mit den Modalitäten Lesen/Hören/Schreiben (Spezialfälle: Brailleschrift lesen/Tasten und stumm sprechen/Lippenbewegungen).

10.1 Allgemeines zu MMS

Multimodale Systeme sind durch die Verarbeitung von mindestens zwei kombinierten Input-Modalitäten (wie Sprache, Stift, Berührung, Handbewegungen, Blickrichtung, Kopf- und Körperbewegungen) gekennzeichnet (siehe Oviatt 2002). Begünstigt durch den rasanten technischen und technologischen Fortschritt beginnen sie die traditionellen WIMP (Window, Icon, Menu, Pointing device)-Schnittstellen in Richtung einer für Menschen natürlicheren Interaktion abzulösen, indem sie das ganze Spektrum der kommunikationsbezogenen Handlungen des Menschen verwenden.

Wird die Schnittstelle durch eine Avatar-ähnliche Figur („Animierter Präsentationsagent“ oder **Persona**) repräsentiert/personifiziert (z. B. der virtuelle Kommunikationsassistent Smar-takus in **SMARTKOM**), so ist es möglich, das Spektrum der Input-Modalitäten auch für den Output zur Verfügung zu stellen (indem die Figur Mimik, Gestik und andere körperliche Bewegungen zeigt). In diesem Fall spricht man von *symmetrischer* Multimodalität (s. Wahlster 2006a), wie in Abb. 10.1 auf der nächsten Seite dargestellt.

Vorteile multimodaler Systeme sind allgemein die bessere Adaptierbarkeit eines MMS an seine Aufgaben (flexibler Einsatz der Modalitäten, sowohl durch den Designer als auch durch den Benutzer) und seine bessere Adaptivität der multimodalen Kommunikation an die Umwelt (Einsatz der Modalitäten durch das System) sowie die prinzipiell erhöhte Bandbreite für die Informationsübermittlung (durch mehr Informationskanäle). Im Einzelnen sind es

- die durch Modalitätsauswahl verbesserte Zweck- und Kunden-Orientierung beim Design eines MMS

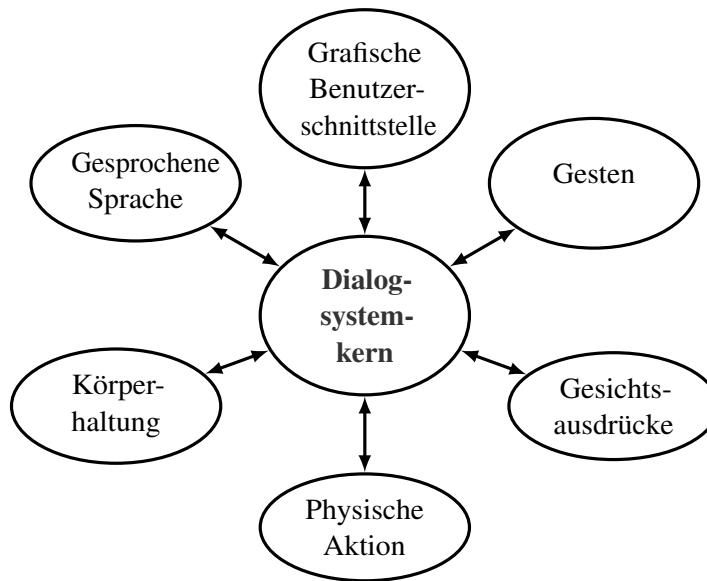


Abbildung 10.1: Modalitäten eines MMS (angelehnt an Wahlster 2006a:4)

- die erweiterten Interaktionsmöglichkeiten und die daraus resultierende Wahlmöglichkeit der Input-Modalitäten für eine bestimmte Aufgabe durch den Benutzer (vs. System-gesteuerter bestimmter „Eingaben“), d.h., die Wahl *wann welche (Kombination der) Modalitäten verwendet wird*; die Kommunikation wird auch *natürlicher* durch Anpassung an die menschlichen Modalitäten
- die Flexibilität im Gebrauch bei verschiedenen Benutzergruppen (die z. B. durch unterschiedliche(s) Alter, Fähigkeiten, Sprache, Präferenzen oder Beeinträchtigungen gebildet werden); jede Benutzergruppe kann die präferierte(n) Modalitäten wählen
- die erhöhte Effektivität aufgrund der Anpassungsmöglichkeiten an die jeweilige Situation. Gerade bei mobilen Geräten ist deren Benutzung durch vielfältige Situationsfaktoren beeinflusst. Eine Auswahlmöglichkeit kompensiert hier aktuelle Nachteile bestimmter Modalitäten (z. B. schlechte Grafik durch zu große Helligkeit; schlechte Akustik durch zu hohe Umgebungslautstärke)
- eine erhöhte Effizienz bei der Erledigung bestimmter Aufgaben (d.h., nicht unbedingt generell) durch die Möglichkeit multimodaler Interaktion
- eine größere Robustheit des Systems durch verbesserte Fehlertoleranz (Fehlervermeidung und Fehlerbehandlung), sowohl auf Benutzer- wie auf Systemseite. Beispielsweise können Benutzer eine „schwierige“ Modalität vermeiden (und somit Fehler), das System kann bei Problemen Input aus mehreren Modalitäten nutzen, z. B. bei der Auflösung von Ambiguitäten.

Nachteile multimodaler Systeme sind

- ggf. eine verminderte Qualität der Interaktion aufgrund der beschränkten Aufmerksamkeitsressourcen des Benutzers; das Registrieren und Verarbeiten multi-modaler Information kostet mentale Kapazität, die nicht immer ausreichend zur Verfügung steht (wegen weiteren Einflüssen bzw. Agieren in der Welt); dies kann zum Informationsverlust führen
- die erhöhte Komplexität des Systems, die zu Verwirrung (worauf soll ich jetzt achten?) oder zu psychischem Druck (verpasse ich auch keine Information?) führen kann

Es wird sicher noch dauern, bis die Technologie die Vorteile von MMS vollständig ausgenutzt und ihre Nachteile aufgefangen haben wird. Hierzu sind insbesondere noch technische (z. B. Sensoren, Prozessorenkapazität, Verkleinerung der Geräte), softwaretechnische (z. B. im Bereich Mienen-, Gestenerkennung) und konzeptionelle (bzgl. des Designs multimodaler Schnittstellen)¹ Fortschritte notwendig. Es sollte jedoch klar sein, dass MMS zu einer weitaus natürlicheren kommunikativen Interaktion mit Geräten führen werden, da auch bisher vernachlässigte Kommunikationskanäle wie Mimik, Gestik und Körpersprache ausgenutzt werden.

Bis ein solcher Idealzustand erreicht ist, ist allerdings mit individuellen Unterschieden in der Akzeptanz multimodaler Kommunikation mit Computern in bestimmten Anwendungsbereichen zu rechnen (s. *Boves* 2004).

10.2 Geschichte

Meilensteine der Entwicklung von MMS sind bei

- überwiegend multimodalem *Input*:
 - **PUT-THE-THERE** (*Bolt* 1980)
 - **QUICKSET** (*Cohen et al.*, 1998)(Systeme dieser Generation können meist gesprochene Sprache zusammen mit Zeigegesten auf Kartendarstellungen (per Touchpad, Maus, Stift) verarbeiten)
- überwiegend multimodalem *Output*:
 - **XTRA** (*Allgayer et al.* 1989): grafische Zeigegesten bei rechnergestütztem Ausfüllen von Steuerformularen
 - **COMET** (COordinated Multimedia Explanation Testbed, *Feiner und McKeown* 1991; Schema-basierte Output-Generierung),
WIP (Wissensbasierte Informations-Präsentation, s. *Wahlster et al.* 1995; Plan-basierte Generierung)
 - **PPP** (Personalized Plan-based Presenter, *André et al.* 1996; Verwendung einer Persona)

¹ s. hierzu *Sarter* (2006).

- multimodalem *Input und Output*:
 - **CUBRICON** (*Neal und Shapiro 1991*)
 - **ALFRESCO** (*Stock 1991*; Multimodales Informationssystem zu Wandmalereien)
 - **SMARTKOM** (*Wahlster 2006b*; symmetrisches MMS: Informationskiosk),
 - MATCH** (Multimodal Access To City Help, *Johnston et al. 2002*)

Während die ersten Systeme ausschließlich gesprochene Sprache in direkter Verbindung mit Mausklick-artigen Zeigegesten zur Auflösung von *deiktischen Ausdrücken* wie „hier“ und „da“ kombinierten, wurden ab Ende 1980er verstärkt NLP-Methoden eingesetzt, durch die mehr Flexibilität erreicht wurde. Allerdings erlaubte erst das **QUICKSET**-System eine größeres Spektrum an Gesten und behandelte die beiden Modalitäten gleichrangig.

Die Trends/Entwicklungen in den darauffolgenden Jahren betreffen die Aspekte Dialogfähigkeit, Einsatz von mehr Modalitäten (bis hin zur Verwendung von Gerüchen) sowie der Symmetrie der Multimodalität. Neue Modalitäten und ihre Verarbeitung betreffen dabei nicht nur Nutzer und System, sondern werden oft auch dazu genutzt, die Natürlichkeit der Persona in ihrer Gestik, Mimik und Körpersprache zu verbessern.

Ein zusätzlicher, eher technischer Aspekt betrifft den möglichen Einsatz von MMS in sicherheitsrelevanten Bereichen. Fügt man **Biometrie** (Abtasten von Körpermerkmalen (der Iris, der Hand etc.)) hinzu, wie in **SMARTKOM**, so eröffnen sich weite Absatzfelder für MMS.

Das System **PUT-THE-THERE** (*Bolt 1980*)

Gegenstand des am MIT entwickelten Systems war die Verwendung von Gesten zusammen mit gesprochener Sprache mit dem Ziel, die auf einer grafischen Oberfläche angezeigten Objekte einfach manipulieren zu können. Hierzu saß ein Benutzer in einem Büro-großen „Media room“ ca. 2,5m vor einer Wand-großen Projektionsfläche, am Handgelenk ein magnetischer Sender, dessen Bewegung in die Zeigerichtung des Arms (angezeigt durch ein kleines weißes „x“ auf der Wand) umgerechnet wurde.

Durch sprachliche Kommandos konnten einfache Figuren auf der Oberfläche kreiert (auch: verändert, umbenannt, entfernt) werden: „Create a blue square ... there.“ Hierbei zeigte der Zeitpunkt der Äußerung von „there“ die Position des Objekts an. Außerdem konnten die verfügbaren Objekte bewegt und positioniert werden: „Move the blue triangle to the right of the green square.“ Sowohl die Behandlung von Objekt- als auch die von Raumeigenschaften war allerdings stark vereinfacht.

Eine wesentliche Eigenschaft des Systems war nun der multimodale Bezug auf Objekte, bei dem die pronominale sprachliche Referenz durch gleichzeitige Zeigegesten aufgelöst wurde: „Put that there“.

10.3 Aspekte und Anforderungen

Multimodalität stellt zusätzliche Anforderungen an Dialogsysteme, die generell die Koordination der verschiedenen Modalitäten betreffen. Bei der **multimodalen Analyse** müssen die Eingaben der einzelnen Modalitäten nach unimodaler Erkennung aufeinander abgestimmt/synchronisiert und integriert werden (**multimodal fusion**). Bei der **multimodalen Outputgenerierung (multimodal fission)** müssen die zu präsentierenden Inhalte sinnvoll und zweckmäßig auf die verschiedenen Modalitäten verteilt werden.

10.3.1 Multimodal fusion

Es gibt prinzipiell fünf verschiedene Muster der Interaktion verschiedener Modalitäten innerhalb einer multimodalen Äußerung (z. B. als Antwort auf die Frage eines Stadtnavigationssystems mit Touchscreen: „Was möchten Sie gerne als Nächstes sehen?“):

- **redundant:**
die durch die Modalitäten gelieferte Information ist praktisch identisch
(„das Museum“+<Berühren des Museum-Symbols>)
- **komplementär:**
die Informationen ergänzen sich
(„das hier“+<Berühren des Museum-Symbols>)
- **übergebend (sequentiell):**
die Informationen werden erst in einer, dann in einer anderen Modalität spezifiziert
(„ähh, das“, danach <Berühren des Museum-Symbols>)
- **nebenläufig (concurrent):**
die Informationen sind unabhängig voneinander
(„Ist das weit?“+<Berühren des Museum-Symbols>)
- **kontradiktiorisch:**
die durch die Modalitäten gelieferte Information ist widersprüchlich
(„das Museum“+<Berühren des Oper-Symbols>)

Redundante Multimodalität kann dazu genutzt werden, unimodale *Ambiguitäten* aufzulösen. Üblicherweise liefern maschinelle Analysen mehr Hypothesen als gewünscht. Durch cross-modalen Vergleich können aber selbst Zweit- oder Drittplatzierte der unimodalen Analysen-Bestenlisten als richtige Kandidaten identifiziert werden. Ein weiteres gutes Beispiel für die Kompensation unimodaler Ambiguität ist ein multimodaler Texteditor: Zeigt man hier auf eine Stelle im Text (Absatz, Wort, Buchstabe?), kann die verbale Modalität disambiguieren („einrücken“, „Wort groß schreiben“ etc.). Auch unimodale *Fehlanalysen* können so kompensiert werden, z. B. wenn die Berührungsanalyse kein eindeutiges Objekt identifiziert konnte.

Ähnliches gilt auch für den Fall der *Komplementarität*, bei dem die Aufgaben der multimodalen Koreferenzresolution (Auflösung, auf welche Objekte der Welt sich durch mehrere Modalitäten gleichzeitig bezogen wird), der cross-modalen Koreferenzresolution (Auflösung,

auf welche Objekte einer Modalität sich mit einer anderen bezogen wird, z. B. „der blinkende Punkt auf der Karte zeigt das Museum“) und der Anaphernresolution (Auflösung, auf welche Objekte im vorherigen Dialog sich anaphorische Ausdrücke wie „es“ oder „das Gebäude“ beziehen) bewältigt werden müssen. Hier kann Ambiguität durch Ausdrücke entstehen, die sich auf den bisherigen Dialog oder auf die aktuelle Situation beziehen können („Möchten Sie als Nächstes das Museum anschauen? Das möchte ich <berührt den Screen>“).

Empirische Erkenntnisse zu multimodaler Interaktion

Was für einen Wert hat Multimodalität für Benutzerschnittstellen? *Oviatt* (1999) untersucht diese Frage, indem sie 10 populäre Ansichten („Mythen“) „currently fashionable among computationalists“ (*ibid.*, S. 75) einer empirischen Analyse unterzieht. Dabei gelangt sie zu folgenden Ergebnissen:

- 1 Benutzer eines MMS interagieren nicht notwendigerweise multimodal. Dies variiert je nach Art der Handlungen, die durchgeführt werden.
- 2 Die Kombination von gesprochener Sprache und Zeigegesten ist nicht das dominierende multimodale Integrationsmuster, sondern wird vor allem bei der Auswahl von Objekten angewendet. Der prozentuale Anteil dieses Musters sinkt ansonsten – wie auch Zeigegesten – auf unter 20%.
- 3 Multimodale Eingabe involviert nicht so häufig wie erwartet gleichzeitige Signale. Oft werden (komplementäre) Zeigegesten vor der sprachlichen Äußerung durchgeführt.
- 4 Auch wenn Sprache eine der möglichen Modalitäten ist, ist sie trotzdem nicht notwendigerweise der primäre Eingabemodus.
- 5 Multimodale Sprache unterscheidet sich von unimodaler Sprache. MM-Äußerungen sind oft weniger komplex und erheblich knapper, sie können sogar syntaktisch andere Strukturen aufweisen.
- 6 Multimodale Integration involviert nicht etwa in erster Linie Redundanz, sondern überwiegend Komplementarität.
- 7 Die Fehleranfälligkeit unimodaler Analysen addiert sich nicht bei multimodaler Fusion, sondern führt eher zu robusterer Erkennung der Eingaben – einerseits wegen Fehlervermeidungsmöglichkeiten auf Seiten des Benutzers, andererseits wegen cross-modaler Disambiguierung und Fehlerkorrektur.
- 8 Benutzer unterscheiden sich systematisch in der Verwendung der Eingabemöglichkeiten. Eine Berücksichtigung dieses Umstands kann die Komplexität der multimodalen Fusion möglicherweise reduzieren.
- 9 Eingabemodalitäten sind nicht gleichwertig dahingehend, dass sie dieselben Informationen übermitteln können.
- 10 Verbesserte Effizienz bzw. Schnelligkeit ist nicht der Hauptvorteil multimodaler Systeme. Weniger Fehler bei Systemnutzung sowie dessen Adaptierbarkeit und Adaptivität haben einen größeren Mehrwert.

Multimodale *Nebenläufigkeit* erhöht die Komplexität der Erkennung dessen, was mit der multimodalen Eingabe bewirkt werden soll. Während ein unimodaler Sprech-/Dialogakt schon mehrdeutig sein kann², verkompliziert sich dies im Fall multimodaler Dialogakte. Im Beispiel oben wird gleichzeitig eine Antwort gegeben und eine Frage gestellt.

Generell ist es die zentrale Aufgabe der multimedialen Fusion zu bestimmen, was zusammen gehört und was getrennt werden muss. Die Nebenläufigkeit zeigt, dass temporale Überlappung verschiedener Modalitätsinputs nicht notwendigerweise einen Zusammenhang implizieren. Anders herum /konvers dazu ist Nicht-Überlappung kein sicheres Trennsignal. *Sequentialität* der Eingaben ist ein Fall von Zusammenhang-trotz-zeitlicher-Trennung. Hinzu kommt, dass redundante/komplementäre Eingaben faktisch oft sequentiell vorgenommen werden (<Berühren des Museum-Symbols><Pause>, „das hier“), s. Punkt 3 des Exkurses auf der vorherigen Seite.

Eine wichtige Voraussetzung für eine effektive multimodale Integration der Eingaben ist eine explizite Repräsentation des multimodalen Eingabekontexts und eine Koordination der unimodalen Erkenner, um die Eingaben adäquat analysieren zu können. Dies ist auch wichtig, um die inhaltliche Wohlgeformtheit multimodaler Eingaben prüfen zu können und *kontradiktoriale* Inputs zu erkennen. Letzteres ist jedoch grundsätzlich komputationell schwierig, so dass *Johnston und Bangalore* (2005) schreiben: „The multimodal grammar we presented above only allows for non-contradictory input from the user.“

10.3.2 Multimodal fission

Im Bereich der Generierung natürlicher Sprache wird üblicherweise grob dazwischen unterschieden, **was** ausgedrückt werden soll und **wie** es ausgedrückt wird (s. z. B. *Reiter und Dale* 2000). Die Generierung multimodalen Outputs involviert zusätzlich den Aspekt der Auswahl, **in welcher Modalität** dies jeweils geschieht. Multimodal fission lässt sich daher aufteilen in die **Inhaltsbestimmung**, die Koordination der Verteilung bestimmter Inhalte auf Modalitäten und die modalitätsspezifische Realisierung der jeweiligen Inhalte.³

Die *Inhaltsbestimmung* umfasst die Auswahl relevanter Inhalte nach Diskursziel und Situationsparametern (Kontext, Benutzermodell etc.) sowie die Strukturierung der Inhalte, die z. B. je nach Benutzergruppe unterschiedlich ausfallen kann.

Im Verlauf der multimodalen Koordination muss für ein bestimmtes Stück Information bestimmt werden, durch welche Modalität(en) es am Besten ausgedrückt wird (**modality allocation**). Hierbei sind eine Reihe von Faktoren relevant: Charakteristika der Information, Charakteristika der Modalitäten, die Ziele der Dialogpartner, die Eigenschaften und Aufgaben des Benutzers, Ressourcenbeschränkungen und situative Einflüsse. Beispiele für entsprechende Regeln (*André* 2000:312) sind:

1. Präferiere Grafik bei konkreter Information (Form, Farbe, Textur)

² Klassisches Beispiel ist die Äußerung „Es zieht“, die eine situationelle Gegebenheit ausdrücken kann, oder aber eine Aufforderung, sie zu ändern (Tür/Fenster schließen).

³ Eine feinere Unterscheidung ist die zwischen 5 Ebenen der Generierung: Kontroll-, Inhalts-, Design-, Realisierungs- und Präsentationsdarstellungsebene. S. hierzu z. B. *André* (2000).

2. Präferiere Grafik vor Text bei räumlicher Information (z. B. Ort, Ausrichtung, Zusammensetzung)
3. Verwende Text für quantitative Information (z. B. *die meisten, einige, genau* etc.)
4. Präsentiere Objekte, die kontrastiert werden, in derselben Modalität

Die multimodale Koordination umfasst außerdem die Planung cross-modaler Referenzen sowie die räumliche und zeitliche Koordination der Ausgaben.

Bei der modalitätsspezifischen Realisierung der jeweiligen Inhalte (**modality realization**) werden die abstrakten Spezifikationen für eine bestimmte Ausgabe in eine konkrete Form überführt (ein wohlgeformter natürlichsprachlicher Satz, ein bestimmter Kartenausschnitt etc.), um dann in entsprechenden Medien ausgegeben zu werden.

State-of-the-art-Systeme verwenden Planungssysteme für die Outputgenerierung und integrieren einzelne Aspekte der multimodal fission, um eine flüssige und situationsgerechte Informationspräsentation zu erreichen. Hierfür ist eine maximale Explizitheit der verwendeten Repräsentation notwendig („no presentation without representation“), was auch zur Entwicklung diverser Markup-Sprachen für multimodale Präsentationen geführt hat.

10.3.3 Kommunikatives Verhalten

Die Anzahl möglicher Beziehungen zwischen Inhalt und Ausdruck erhöht sich in MMS erheblich. Inhalte können in verschiedenen Modalitäten ausgedrückt werden, und eine Modalität kann jeweils verschiedene Inhalte ausdrücken.

Hinzu kommt, dass schon in der sprachlichen Modalität zwischen der „eigentlichen“ Bedeutung einer Äußerung (propositionaler Aspekt, Semantik) und ihrer kommunikativen Bedeutung (kommunikativer Aspekt, Pragmatik; s. auch Fußnote 2 auf der vorherigen Seite) unterschieden werden muss. Im Folgenden sind einige Beispiele für die Realisierung verschiedener kommunikativer Funktionen als modalitätsspezifische Verhaltensweisen aufgeführt (s. Cassell 2001:73), die MMS erkennen und generieren können müssen⁴.

Initiation und Termination

- Auf neue Person reagieren → Kurzer Blick zum Anderen
- Nicht mehr an Konversation beteiligen → Umherschauen
- Verabschieden → den Anderen anschauen, Kopfnicken, Winken

Turn-taking

- Turn abgeben → Schauen, Augenbrauen heben (gefolgt von Schweigen)
- Turn fordern → Hand in den Gestenraum heben
- Turn übernehmen → Wegschauen, anfangen zu sprechen

Rückmeldung

- Rückmeldung fordern → den Anderen anschauen, Augenbrauen heben
- Rückmeldung geben → den Anderen anschauen, Kopfnicken

⁴ Hinzu kommt der große Bereich emotionaler Reaktionen wie (Des)Interesse, Ärger, Freude etc.

10.4 Architekturen und Systeme

Abb. 10.2 zeigt die Grobarchitektur multimodaler Systeme mit den jeweils komplexeren Input- und Output-Verarbeitungskomponenten.

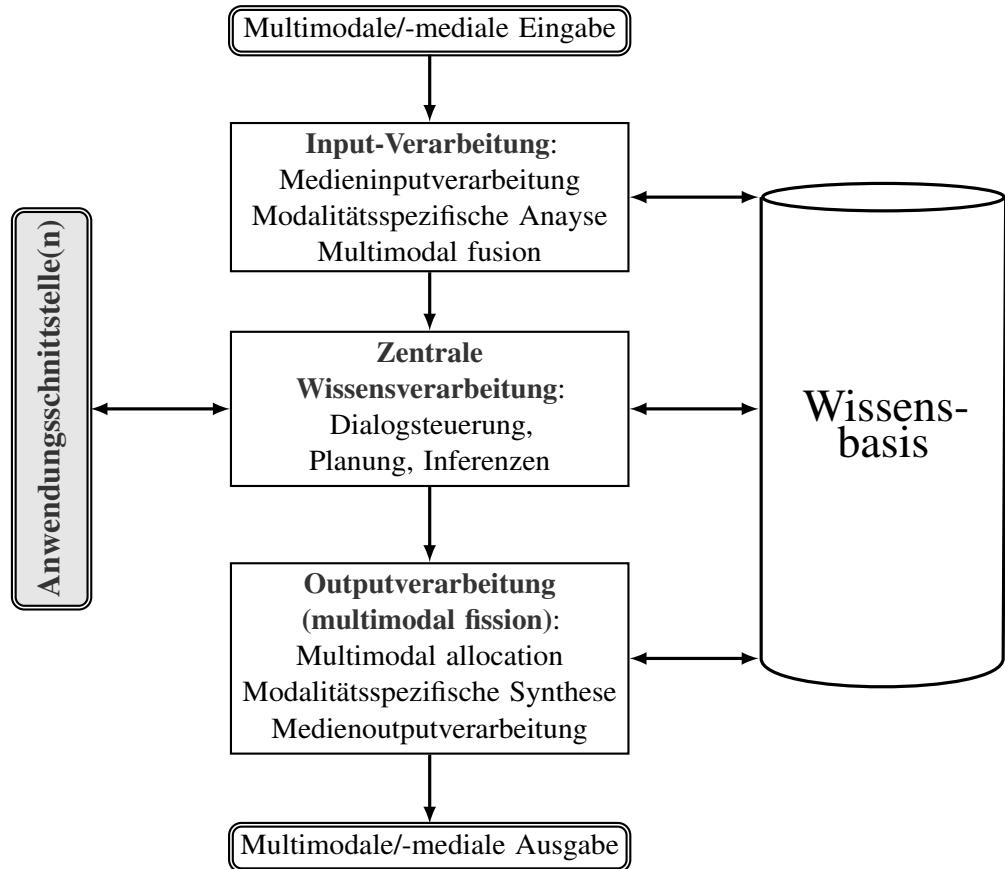


Abbildung 10.2: **Grobarchitektur eines MMS**

Die Konstruktion solcher Systeme erfordert (zur Zeit noch) einen sehr hohen Forschungs- und Entwicklungsaufwand (zu Stand und Perspektiven s. *Bunt et al. 2005*). Viele Systeme beschränken sich daher auf zwei verschiedene Modalitäten und außerdem auf entweder Input- oder Output-Funktionalität (s. *Foster 2002:3* für einen Überblick).

Ausnahmen sind insbesondere die Arbeiten von J. Cassell zu *Conversational Humanoids* bzw. *Embodied Conversational Agents*⁵ sowie die Arbeiten zu intelligenten Kommunikationsassistenten im Forschungsbereich „Intelligente Benutzerschnittstellen“ am DFKI, z. B. im Rahmen des SmartKom-Projekts. Die jüngsten in diesen Gruppen konstruierten Systeme zeichnen sich vor allem durch nicht-redundante symmetrische Multimodalität aus.

⁵ früher in der ’Gesture&Narrative Language Group’ des MIT Media Lab, heute am ’Center for Technology & Social Behavior’ der Northwestern University

10.4.1 SMARTKom

SMARTKom (entwickelt von 1999-2003) ist sicher eines der bislang größten und innovativsten computerlinguistischen Projekte. Als solches hat es nicht zuletzt Fortschritte auf Methodenebene erzielt, von denen Wahlster⁶ die wichtigsten zusammenfasst:

- nahtlose Integration und gegenseitige Disambiguierung von multimodalem Input und Output auf semantischer und pragmatischer Ebene
- Situiertes Verstehen von möglicherweise unpräzisem, mehrdeutigem oder unvollständigem multimodalem Input
- Kontextsensitive Interpretation dialogischer Interaktion auf der Grundlage von dynamischen Diskurs- und Kontextmodellen
- Adaptive Generierung von koordinierten, in Inhalt und Form zusammenhängenden (d.h., kohärenen und kohäsiven) multimodalen Präsentationen
- Halb- oder vollautomatische Erfüllung der vom Benutzer in Auftrag gegebenen Aufgaben durch die Integration von Informationsdiensten
- Intuitive Personifikation des Systems durch einen Präsentationsagenten

Entsprechend komplex ist die resultierende Architektur des **SMART-KOM**-Systems : **SMARTKom** realisiert eine symmetrische multimodale Interaktion zwischen dem Benutzer und Smartakus (s. das Bild rechts), einer computeranimierten stilisiert-anthropomorphen 3D-Figur mit der Fähigkeit zur Darstellung verschiedenster Gesten, Gesichtsausdrücke und Körperhaltungen. Hierbei überträgt der Benutzer Smartakus eine Aufgabe, die von diesem ggf. mithilfe eines kollaborativen Dialogs (zum Abgleich der genauen Interessen des Benutzers mit möglichen Plänen des Kommunikationsassistenten, s. hierzu auch Kap. 8) und unter Rückgriff auf digitale Dienste und spezifische Anwendungen erledigt wird.



Durch seine modulare Struktur kann das System an verschiedene Anwendungskontexte adaptiert werden (insgesamt bewältigt es mehr als 40 verschiedene Aufgabentypen). Im Projekt SmartKom wurden drei verschiedene Anwendungsszenarios realisiert (Abb. 10.3 auf der nächsten Seite):

- ein Kommunikationsassistent, der bei Telefon-, Fax-, Email- und Authentifizierungsaufgaben hilft („Public“: Station mit Touchscreen, Kameras für Gesichts- und Gestenkennung)
- ein Infotainment-Assistent, der Medieninhalte auswählen hilft und verschiedene TV-Anwendungen bedienen kann („Home“: wird über einen Tablett-PC bedient)
- ein mobiler Reiseassistent, der bei der Navigation (auch im Auto) sowie bei

⁶ in einem Vortrag „SmartKom: Modality Fusion for a Mobile Companion based on Semantic Web Technologies“ auf dem ‚Cyber Assist Consortium Second International Symposium‘ (März 2003)

der Auskunft zu Sehenswürdigkeiten über Location-Based Services (Mobilfunk-Dienste) hilft („Mobile“: Bedienung per PDA)

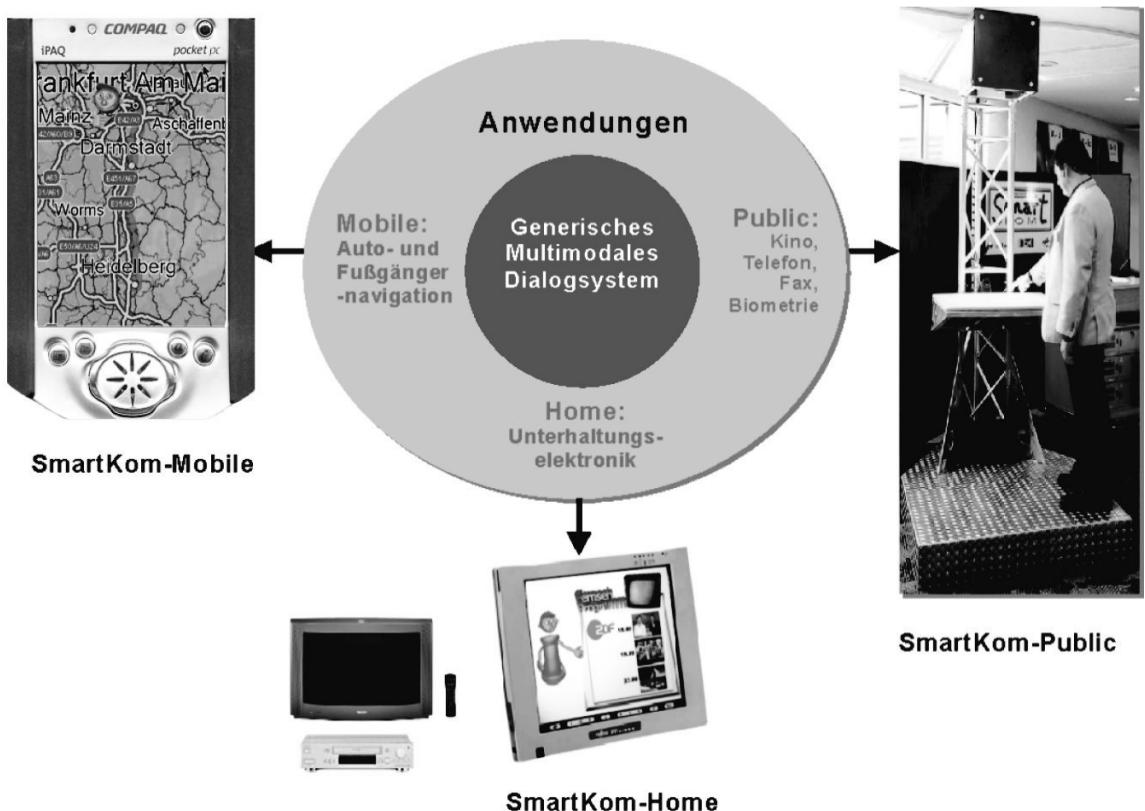


Abbildung 10.3: **Anwendungsszenarien in SmartKom** (aus: *Reithinger und Blocher 2003*)

Der folgende Dialog ist ein Beispiel für **SMARTKOMs** multimodale Interaktion. Sie besteht in diesem Fall aus der grafischen Darstellung der Persona und ihres Verhaltens, der grafisch-textuellen Information sowie dem sprachlichen Dialog.

Ben:	I would like to go to the movies tonight.
Smartakus:	[zeigt eine Liste mit Kinofilmen] This is a list of films showing in Heidelberg.
Ben:	Hmm, none of these films seem to be interesting... Please show me the TV program.
Smartakus:	[zeigt einen Fernsehprogrammausschnitt] Here [zeigt darauf] is a listing of tonight's TV broadcasts.
Ben:	Please tape the third one!

Sowohl der Benutzer als auch Smartkus verwenden hier multi-bzw. cross-modale Anaphern und Referenzen („This“, „these“, „Here“, „the third one“). Wesentlich dabei ist, dass unter ande-

ren Bedingungen (z. B. im Auto) die grafische Anzeige der Information durch eine sprachliche Auskunft ersetzt werden würde.

Diese Adaptivität an die situationalen Umstände ist im Folgeprojekt **SMARTWEB** fortgeführt und verfeinert worden (Wahlster 2007). Beispielsweise wurden sogar Interaktionen im Straßenverkehr unter Berücksichtigung des Belastungs-/Stresszustands eines Autofahrers sowie der Gefahrensituation eines Motorradfahrers untersucht.

10.4.2 REA

Die Arbeiten von Justine Cassell (z. B. *Cassell 2001*) basieren auf der Annahme, dass multimodale Interaktion mit künstlichen Systemen idealerweise der natürlichen Kommunikation von Menschen nachgebildet werden sollte.



Abbildung 10.4: **REA**

Hieraus folgt ihr Bestreben, anthropomorphe/humanoid lebensgroße Personas zu entwickeln (s. Abb. 10.4), mit denen ein natürlicher symmetrisch-multimodaler Dialog stattfinden kann (über eine entsprechend große Projektionsfläche, s. Abb. 10.5).

Eines ihrer Systeme ist **REA** (für „Real Estate Agent“ (Grundstücksmakler)), das bei der Suche nach einer geeigneten Immobilie behilflich ist. Ein Ausschnitt aus einer solchen Konversation mit **REA** ist auf Seite 164 zu sehen.

Die in Abb. 10.6 auf der nächsten Seite gezeigte Architektur von REA ähnelt der in Abb. 10.2 auf Seite 158. Ein Unterschied besteht in der hier vorhandenen „Abkürzung“ zwischen Input und Output durch „fest verdrahtete Reaktionen“. Sie sind dafür da, schnelle Verhaltensreaktionen zu ermöglichen, die die zeitintensive deliberative Verarbeitung umgehen/vermeiden, also z. B. unmittelbare körperliche Reaktionen der Persona auf die Aktionen des Benutzers (Zuwendung bei Herannahen, Augenkontakt etc.).



Abbildung 10.5: **REA im Gespräch**

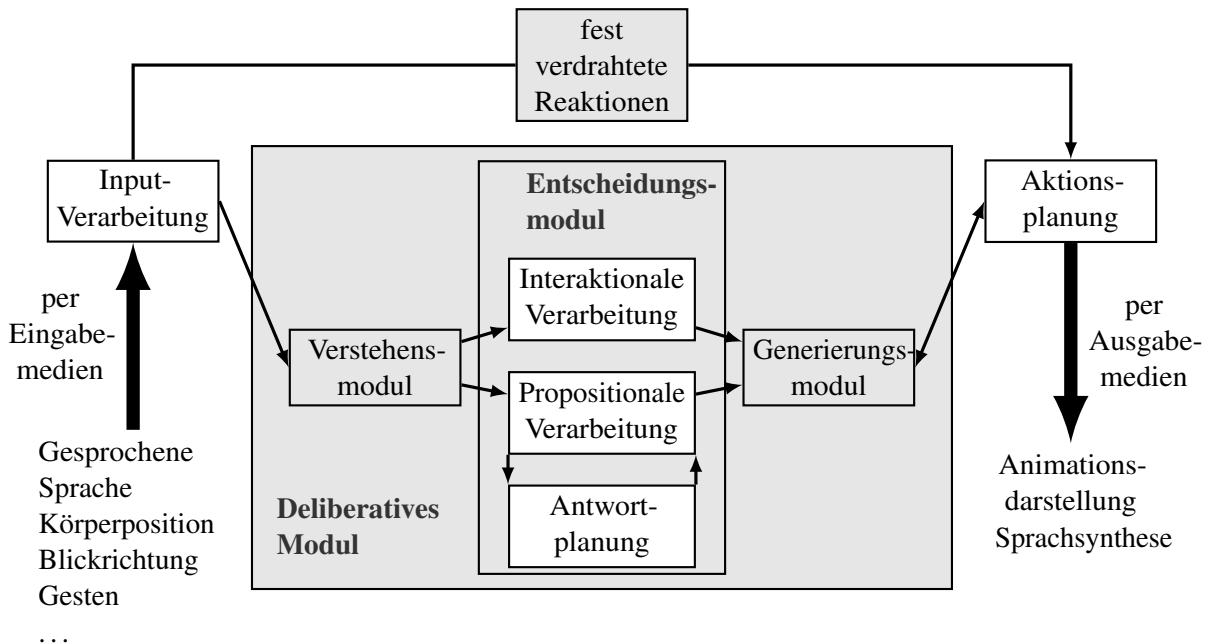


Abbildung 10.6: Architektur von REA (aus: Cassell 2001:75)

Ein zweiter wesentlicher Unterschied ist die klare Trennung von interaktionaler (auf die Kommunikation bezogener) und propositionaler (auf den Inhalt bezogener) Verarbeitung.

10.4.3 MAX

MAX ist ein an der Universität Bielefeld konstruierter künstlicher, verkörperter konversationeller Agent, an dem kontinuierlich Technologien multimodaler Systeme getestet und weiterentwickelt werden. Ursprünglich gedacht als künstlicher kollaborativer Partner in Konstruktionsaufgaben (daher: Multimodal Assembly eXpert, s. Kopp et al. 2003), wurde er später unter Anderem als virtueller Museumsführer im Heinz Nixdorf Museums-Forum (HNF) in Paderborn real eingesetzt (Kopp et al. 2005). **MAX** stellt ein Musterbeispiel für die anwendungsorientierte Mischung tiefer und flacher Verfahren dar.



Abbildung 10.7: MAX

Auf der einen Seite liegt ihm ein rationalistischer, kognitiv orientierter Ansatz zugrunde, der sich an dem BDI-Modell (s. S. 127) orientiert. Einzelne Dialogakte werden durch die Repräsentation ihrer kommunikativen Funktion modelliert, die drei Aspekte/Ebenen involviert: den *performativen* Aspekt auf der Interaktionsebene (Unterscheidung von 'Auskunft geben' und 'Auskunft einholen'); den *referentiellen* Aspekt (auf welche Ebene sich eine Äußerung

bezieht, z. B. die Unterscheidung von 'Wie geht es Dir?' (Dialogebene) und 'Wie geht es Laura?' (Inhaltsebene); den *Inhaltsaspekt* (propositionalen Gehalt einer Äußerung). Diese drei Komponenten werden für eine Dialogaktrepräsentation wie folgt notiert:

<performativer A.><referentieller A.><inhaltlicher A.>. Hieraus ergeben sich die folgenden (multimodalen) Darstellungen kommunikativer Funktionen⁷:

provide.interaction.greeting	(z. B. „Hallo!“)
askFor.content.name	(z. B. „Wie heißt du?“)
askFor.discourse.topic.sports	(z. B. „Lass uns über Sport reden.“)
provide.interaction.greeting	mit der Hand winken
provide.discourse.agreement	Kopf nicken
provide.content.ironical	zwinkern
provide.content	Hand heben
*.content.number-two	zwei ausgestreckte Finger

Auf der anderen Seite können die Regeln, durch z. B. die Interpretation der Eingaben gesteuert werden, so einfach sein, dass sie denjenigen von Chatterbots gleichen (s. (16)):⁸

```
(16) <rule name='interprete.type1.farewell'>
    <match>
        <keywords>     bye,cu,cya,exit,quit,ciao,ade,adios,
                      hasta*, auf wieder*,tschoe,tschues*,tschau,
                      und weg, so long,machs gut,bis bald,bis dann,
                      bis spaeter,wiedersehen
        </keywords>
    </match>
    <assert>    <convfunction modifier='farewell' filter='yes' />
    </assert>
</rule>
```

Moderne Versionen von **MAX** verfügen über ein Repertoire multimodaler emotionaler Äußerungen, die auf einer partiellen Integration emotionaler Aspekte in das BDI-Modell basieren.

⁷ wobei einzelne Aspekte auch unterspezifiziert sein können

⁸ S. hierzu ein beeindruckend elaboriertes Beispiel.

Eine Konversation mit REA

Ablauf einer tatsächlichen Interaktion zwischen einem Benutzer (Mike) und REA:

Mike nähert sich der Projektionsfläche. REA wendet ihm die Seite zu und schaut untermütig umher. Als Mike in den Bereich der 2 oben am Screen angebrachten Kameras kommt, wendet sich REA ihm zu und sagt:

REA: Hello. How can I help you?

MIKE: I'm looking to buy a place near MIT.

REA nickt und zeigt damit, dass sie versteht.

REA: I have a house to show you.

Ein Bild eines Hauses erscheint auf dem Schirm hinter REA, die blinzelt und erst auf das Haus, dann zu Mike schaut.

REA: It's in Somerville.

MIKE: Tell me about it.

REA schaut hoch und zur Seite, während sie plant, was sie sagen will.

REA: It's big.

REA macht eine ausladende Bewegung mit ihren Händen. Mike erhebt seine Hände, als wollte er sprechen, und so redet REA nicht weiter, sondern wartet darauf, dass er spricht.

MIKE: Tell me more about it

REA: Sure thing. It has a nice garden.

REA skizziert eine kurvige Geste mit ihren Händen und zeigt damit an, dass der Garten sich entlang der zwei Seiten des Hauses erstreckt.

MIKE: How far is it?

REA: It is five minutes to the Porter Square T station.

REA macht deutlich, dass es 5 Minuten zu Fuß von der T-Station sind, indem sie eine Trippel-Geste mit ihren Fingern durchführt.

MIKE: How big is the house?

REA: It has four bedrooms, three bathrooms...

Mike unterbricht REA, die sofort aufhört zu reden.

MIKE: Wait. Tell me. Where is the master bedroom?

REA: I'm sorry, I didn't catch that. What did you ask me?

MIKE: Where is the master bedroom?

REA: It's upstairs.

REA zeigt hoch.

MIKE: Where is the master bathroom?

REA: It's next to the bedroom.

REA bringt ihre Hände zusammen, um die Beziehung zwischen dem Bade- und dem Schlafzimmer anzuzeigen. Und die Haus-Tour geht weiter ...

10.5 Literatur

Als Einstieg in die Thematik bietet sich eine Google-Suche nach Justine Cassell, Mark Maybury, Wolfgang Wahlster und ihren Publikationen an. Elisabeth André, Mitkonstrukteurin einiger MMS der 1990er Jahre ([WIP](#), [PPP](#)), liefert in *André* (2003) einen kurzen Überblick zu MMS, *André* (2000) geht etwas ausführlicher auf den Aspekt der Generierung multimedialer Präsentationen ein. Siehe außerdem *Wachsmuth* (2005) zu der Frage nach dem Bewusstsein multimodaler künstlicher Agenten.

Kapitel 11

Sprachgenerierung (NLG)

Booth: Gun. Noun. Portable firearm. This device was widely utilized in the urban wars of the late twentieth century. Referred to as a pistol, a piece...

Simon Phoenix: Look I don't need a history lesson! C'mon, HAL, where are the god damn guns?

(aus: *Demolition man (Film)*)

Schon heute ist das Vorliegen von Daten für potentielle Nutzer kaum noch ein Problem, und selbst der automatische Zugriff darauf stellt immer weniger eine ernst zu nehmende Hürde dar (s. Kap. 5). Umso wichtiger wird es daher auch in Zukunft sein, relevante Information abhängig vom Kontext und zugeschnitten auf eine konkrete Benutzerin auszuwählen und dieser adäquat zu präsentieren.

(Angewandte) **Sprachgenerierung**^a (auch: *natural language generation (NLG)*) ist der Bereich der Sprachtechnologie, der sich mit dieser Aufgabe beschäftigt. Sprachgenerierungssysteme dienen z. B. dazu, automatisch personalisierte Wetterberichte (in verschiedenen) Sprachen zu produzieren oder einem Arzt zu ermöglichen, seiner Patientin eine ggf. detaillierte Beschreibung ihrer Medikation auf Knopfdruck mitzugeben.^b

Sprachgenerierung gestattet somit eine effektive *information delivery* auch dort, wo aus verschiedensten Gründen (wie räumliche, zeitliche oder Kosten-Beschränkungen) eine Mensch-gemachte Informationspräsentation nicht möglich oder sinnvoll ist.

Eine weitere wesentliche Aufgabe der Sprachgenerierung besteht darin, routinemäßig zu erstellende Dokumente (Briefe, Dokumentationen, Reports etc.) automatisch zu produzieren oder zumindest bei deren Erstellung zu assistieren.

^a nicht zu verwechseln mit *Sprachsynthese*, der reinen Produktion gesprochener Sprache aus Text

^b S. hierzu z. B. Goldberg et al. (1994), de Rosis et al. (1999).

11.1 Allgemeines zur NLG

Lax ausgedrückt ist Sprachgenerierung das konverse Pendant zum Sprachverstehen und man könnte daher annehmen, dass es weder in der Theorie („einfach die Analyse umdrehen“) noch in der Praxis („wir haben doch schon alle Komponenten“) Interessantes und/oder Neues bietet. Tatsächlich sind die beiden Gebiete der Sprachtechnologie sehr unterschiedlich, und zwar in mindestens den folgenden Aspekten:

1. **Allgemeinheitsanspruch/Abdeckung:** Während Sprachverstehen von vornherein einen hohen Allgemeinheitsanspruch hat/hatte¹ (und daran auch gescheitert ist), fällt es viel leichter zu akzeptieren, dass ein System zur Generierung von Wettervorhersagen aus Messdaten nicht gleichzeitig auch Beschreibungen antiker Vasen für Auktionen produzieren kann. NLG-Systeme sind entsprechend viel eingeschränkter in ihrem Anspruch und somit auch in ihrer Abdeckung sprachlicher/kommunikativer Kompetenz.
2. **Probleme:** Wie bereits erwähnt, erfordert Sprachverstehen die Behandlung des Ambiguitätsproblems, Sprachgenerierung die des Auswahlproblems.
3. **Prozesse:** Typischerweise werden beim Sprachverstehen **Inferenzprozesse** verwendet, um aus gegebener Information weitere abzuleiten; charakteristischerweise werden in der Sprachgenerierung eher **Planungs- und Entscheidungsprozesse** eingesetzt.
4. **Repräsentationen:** Typischerweise führt die Analyse beim Sprachverstehen zu semantischen, meistens an der Prädikatenlogik orientierten logischen Strukturen. Dies ist in der Sprachgenerierung nicht unbedingt der Fall. Weder involviert sie immer eine ausspezifizierte semantische Komponente, noch beginnt sie unbedingt mit semantischen Strukturen (sondern oft mit Daten oder Wissensstrukturen): „Existing comprehension systems as a rule extract considerably less information from a text than a generator must appreciate in generating one.“ (McDonald 1993).

Sprachgenerierung zeigt sich somit entgegen der obigen Annahme als ein eigenständiges Gebiet innerhalb der Sprachtechnologie, dessen Systeme im Wesentlichen dadurch bestimmt sind, dass **Daten bzw. Information** in einer bestimmten **Domäne** in eine durch **Planung** bestimmte **sprachliche Form** (d. h. einen Text) umgewandelt werden, die ein bestimmtes **kommunikatives Ziel** im Hinblick auf bestimmte **Benutzer** erfüllt.² Eine der gängigen, auf David McDonald zurückgehende Definition von NLG lautet daher wie folgt:

„Natural language generation is the process of deliberately constructing a natural language text in order to meet specified communicative goals“

¹ Z. B. wäre es unverständlich, wenn ein Sprachverstehenssystem einen Text zum Waldsterben verarbeiten („verstehen“) könnte, jedoch nicht einen über Kindergeburtstage.

² Wichtig sind daher insbesondere auch die Bereiche **Wissensrepräsentation/Ontologien** und **Benutzermodellierung**.

Es ist wichtig zu wissen, dass sich NLG mit wenigen Ausnahmen (s. z.B. die Arbeiten von Gerard Kempen) *nicht* primär am Prozess der menschlichen Sprachproduktion orientiert. Menschliche Sprachgenerierung verläuft generell **inkrementell**, d.h. es werden sukzessive Teile der Äußerung geplant und verbalisiert:

„Ich glaube, daß mancher großer Redner, in dem Augenblick, da er den Mund aufmachte, noch nicht wußte, was er sagen würde.“
Heinrich von Kleist³

Dies bedeutet aber, dass man nach der Äußerung eines Teils „nicht mehr zurück kann“, ein Problem, dass sich NLG-Systeme oft sparen, indem sie Texte als Ganzes erst planen, dann versprachlichen („realisieren“).

11.2 Geschichte

Als Forschungsgebiet, das sich mit der automatischen Produktion ganzer Texte⁴ befasst, hat sich Sprachgenerierung sehr viel später als das des Sprachverständens —nämlich erst in den 1980er Jahren— etabliert⁵. Ein Grund hierfür war sicher der, dass die maschinelle Sprachverarbeitung zunächst mit den verschiedensten Phänomenen/Problemen im Verstehens-/Analysebereich beschäftigt war. Außerdem liegt beim Sprachverständen ein klarer Input vor, wohingegen es bei der NLG viel weniger offensichtlich ist, was eigentlich der Ausgangspunkt des Prozesses ist.

Aber auch in der Linguistik und Psycholinguistik setzten Untersuchungen zur Sprachproduktion relativ spät und in geringerem Umfang ein. Beispielsweise datiert das Standardwerk zur Sprachproduktion aus dem Jahre 1989 (*Levelt* 1989).

Ein wesentlicher Anstoß, Texte automatisch zu produzieren, entstammt den ersten Ansätzen zur Repräsentation von Wissen mithilfe von semantischen Netzwerken/Netzen (*Quillian* 1967, *Simmons und Slocum* 1972). Vor allem die konzeptuellen Dependenzstrukturen Roger Schanks wurden nicht nur für die Simulation des Verstehens von Texten, sondern auch für die NLG verwendet (*Goldman* 1975)—zunächst mit dem Fokus auf der Generierung von *Sätzen* und ihren möglichen Paraphrasen: *Peter gab Paul ein Geschenk* ~ *Paul erhielt von Peter ein Geschenk* ~ *Paul bekam etwas von Peter geschenkt*.

Insbesondere die Berücksichtigung/Repräsentation von schematischen Ereignisstrukturen (sogenannten „Scripts“) sowie von Wissen über die Ziele von Personen und die Pläne (*Schank und Abelson* 1977), sie zu erreichen, führte dazu, dass nicht nur Abfolgen einzelner Sätze produziert werden konnten, sondern zunehmend auch zusammenhängende („kohärente“) Texte.

³ Über die allmähliche Verfertigung der Gedanken beim Reden, Internetausgabe. Version 11.02.
© Kleist-Archiv Sembdner, Heilbronn

⁴ Man beachte, dass bei Dialogsystemen oder im Bereich der maschinellen Übersetzung ja auch gewisse Aspekte der NLG involviert sind.

⁵ auch wenn frühe Systeme im Bereich der maschinellen Übersetzung oder Dialogsysteme bereits Aspekte der Sprachgenerierung enthielten.

Ein erstes Beispiel hierfür ist das System **TALE-SPIN**, das automatisch Fabeln generierte (*Meehan* 1976, s. den Exkurs auf dieser Seite).

Eine Möglichkeit jenseits von Scripts, Texte zu generieren, besteht darin, die inhärente Ordnung der beschriebenen Ereignisse auszunutzen. Dies ist z. B. für die Beschreibung von Tic-Tac-Toe-Spielen (*Davey* 1974) und später für Wegbeschreibungssysteme (s. z. B. den Exkurs auf Seite 172) umgesetzt worden.

TALE-SPIN: Generierung von Fabeln (*Meehan* 1976)

James Meehan, ein Schüler von Schank, implementierte das Programm **TALE-SPIN**, das einfache Geschichten (genauer: Fabeln) generierte.

Der Kern seines Ansatzes bestand darin, die jeweilige Fabel immer von einem Problem und seiner Lösung handeln zu lassen. Hierzu wurden einerseits sowohl die schematischen und Planungs-Aspekte der Theorien von Schank/Abelson (*Schank und Abelson* 1977) verwendet, andererseits konnte der Benutzer des Programms interaktiv die Handelnden und ihre Eigenschaften spezifizieren. Auf <http://grandtextauto.org/2006/09/13/the-story-of-meehans-tale-spin/> lässt sich die Entstehungsgeschichte von **TALE-SPIN** nachlesen.^a Hier sind Beispiele für automatisch generierte Fabeln:

ONCE UPON A TIME GEORGE ANT LIVED NEAR A PATCH OF GROUND. THERE WAS A NEST IN AN ASH TREE. WILMA BIRD LIVED IN THE NEST. THERE WAS SOME WATER IN A RIVER. WILMA KNEW THAT THE WATER WAS IN THE RIVER. GEORGE KNEW THAT THE WATER WAS IN THE RIVER. ONE DAY WILMA WAS VERY THIRSTY. WILMA WANTED TO GET NEAR SOME WATER. WILMA FLEW FROM HER NEST ACROSS THE MEADOW THROUGH A VALLEY TO THE RIVER. WILMA DRANK THE WATER. WILMA WASN'T THIRSTY ANYMORE.

GEORGE WAS VERY THIRSTY. GEORGE WANTED TO GET NEAR SOME WATER. GEORGE WALKED FROM HIS PATCH OF GROUND ACROSS THE MEADOW THROUGH THE VALLEY TO A RIVER. GEORGE FELL INTO THE WATER. GEORGE WANTED TO GET NEAR THE VALLEY. GEORGE COULDN'T GET NEAR THE VALLEY. GEORGE WANTED TO GET NEAR THE MEADOW. GEORGE COULDN'T GET NEAR THE MEADOW. WILMA WANTED TO GET NEAR GEORGE. WILMA GRABBED GEORGE WITH HER CLAW. WILMA TOOK GEORGE FROM THE RIVER THROUGH THE VALLEY TO THE MEADOW. GEORGE WAS DEVOTED TO WILMA. GEORGE OWED EVERYTHING TO WILMA. WILMA LET GO OF GEORGE. GEORGE FELL TO THE MEADOW. THE END.

^a insbesondere die Tatsache, dass das Programm und die Dissertation im Kern in drei Monaten entstanden sind.

Generell muss diese Ordnung für die Textstrukturplanung allerdings explizit hergestellt werden. Für ihr System **KDS** schlagen *Mann und Moore* (1981) hierfür das *Fragment&Compose*-Verfahren vor, das eine Informationsstruktur zunächst in propositionale Einzelteile (heutzutage

typischerweise *messages* genannt) aufbricht, die dann durch Ordnungs-, Filter- und Aggregationsregeln zu einer Textstruktur zusammengesetzt werden.

In den 1980er Jahren kristallisieren sich die zentralen Komponenten eines NLG-Systems heraus, nämlich eine **Inhaltsplanungs- und -strukturierungskomponente** und eine **Realisierungskomponente**, sowie nachfolgend feinere Unterscheidungen/Subkomponenten.

Innerhalb der Inhaltsaufbereitung lassen sich zwei verschiedene Herangehensweisen identifizieren: Entweder Generierung wird als **Planungsprozess** aufgefasst, in dem Inhalte in Bezug auf ein kommunikatives Ziel selektiert und strukturiert werden. **KAMP**⁶ (s. *Appelt* 1985) ist der Prototyp eines Planungs-zentrierten Systems. *Hovy* (1987) beschreibt die pragmatischen Aspekte, die bei der Generierung für unterschiedliche Benutzer / in unterschiedlichen Situationen berücksichtigt werden müssen. Das Teilgebiet der Sprachgenerierung, das sich verstärkt mit den (emotionalen) Belangen des Adressaten befasst, wird *Affective NLG* genannt.

Oder es wird (zusätzlich) **schematisches Wissen** darüber ausgenutzt, wie Texte und deren Teile typischerweise aufgebaut sind.⁷ Das klassische Beispiel für die Verwendung von *Textschemata* auf der Basis von rhetorischen Relationen (auch: Diskursrelationen)⁸ ist **TEXT** (s. *McKeown* 1985). *Rambow* (1990) betont die Notwendigkeit, spezifisches Wissen über die Struktur von Texten in bestimmten Domänen auszunutzen.

1968	Quillians Programm verbalisiert automatisch die Beziehungen zweier Wortkonzepte anhand der Verbindungen im semantischen Netzwerk
1974	PROTEUS , ein System, das die Züge eines Tic-Tac-Toe-Spiels beschreibt (<i>Davey</i> 1974)
1981	KDS (Knowledge Delivery system) (<i>Mann und Moore</i> 1981)
1980er	Erste Dissertationen mit komplexen NLG-Systemen/-komponenten: MUMBLE (<i>McDonald</i> 1980), KAMP (<i>Appelt</i> 1985), TEXT (<i>McKeown</i> 1985), PAULINE (<i>Hovy</i> 1987); erste Wegbeschreibungssysteme (s. auch den Exkurs zu Wegbeschreibungen auf der nächsten Seite)
1992	FoG (Forecast Generator) (<i>Goldberg et al.</i> 1994): erstes angewandtes NLG-System zur Generierung von Wettervorhersagen in langjährigem Einsatz
1996	PEBA-II (<i>Dale et al.</i> 1998): produziert Webseiten (mit dynamisch eingebetteten Hyperlinks) mit Beschreibungen/Vergleichen von Tieren, s. auch den Exkurs auf Seite 179; vgl. auch ILEX (Intelligent Labelling Explorer), ein System, das automatisch Beschreibungen von Objekten (z. B. in Museen) generiert (s. <i>O'Donnell et al.</i> 2001).
2009	ROADSAFE : automatische Generierung von Straßenzustandsberichten (um den Straßendienst zu informieren)

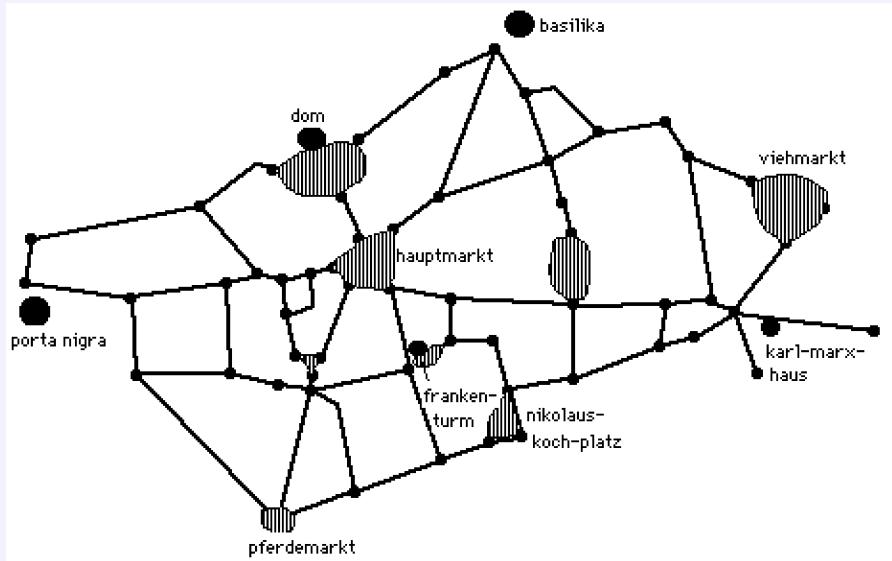
⁶ für Knowledge and Modalities Planner

⁷ Eine Synthese beider Herangehensweisen wird in *Hovy* (1993) vorgestellt.

⁸ Die einschlägige Theorie in diesem Bereich der Strukturierung von Text anhand von rhetorischen Diskursrelationen ist die *Rhetorical Structure Theory (RST)* (*Mann und Thompson* 1988). S. hierzu auch den Exkurs auf Seite 204.

Frühe Generierung von Wegbeschreibungen

In Carstensen (1988) wird ein System präsentiert, das für den folgenden Ausschnitt der Innenstadt von Trier Wegbeschreibungen generiert (s. auch Carstensen 1991).



Hier einige Beispiele für automatisch generierte Beschreibungen auf Fragen der Art „Wie komme ich von X nach Y?“, die auf einer internen räumlichen Repräsentation dieses Ausschnitts basieren:

route(karl_marx_haus, porta_nigra):

gehen sie bei der kreuzung geradeaus in die fleischstrasse. gehen sie immer die fleischstrasse entlang, an der post und interbook vorbei, bis zum hauptmarkt. gehen sie geradeaus ueber den hauptmarkt. biegen sie dann schraeg nach links in die simeonstrasse ein. gehen sie immer die simeonstrasse entlang, an karstadt und dem kaufhof vorbei. da ist dann die porta nigra.

route(viehmarkt, basilika):

gehen sie in die fahrstrasse. gehen sie immer die fahrstrasse entlang. biegen sie an der zweiten ecke links ein in die brotstrasse. gehen sie die brotstrasse entlang. biegen sie an der naechsten ecke schraeg nach rechts ein in die hosestrasse. gehen sie die hosestrasse entlang. biegen sie an der naechsten ecke schraeg nach rechts ein in die konstantinstrasse. gehen sie die konstantinstrasse entlang bis zur naechsten ecke. da ist dann die basilika.

route(basilika, frankenturm):

gehen sie die palaststrasse entlang. biegen sie an der naechsten ecke schraeg nach rechts ein in die grabenstrasse. gehen sie die grabenstrasse entlang bis zur naechsten ecke. da ist der hauptmarkt. gehen sie links ueber den hauptmarkt bis zur steipe. gehen sie da in die dietrichstrasse. gehen sie die dietrichstrasse entlang bis zur naechsten ecke. da ist dann der frankenturm.

Beginnend mit **MUMBLE** (*McDonald 1980*) werden ab den 1980er Jahren umfangreiche separate/domänenunabhängige Realisierungskomponenten entwickelt, die die sprachspezifischen Aspekte der Sprachgenerierung implementieren. Unterschiede bestehen insbesondere darin, auf welchem grammatischen Typ diese Generatoren basieren und von welcher Art ihr Input ist. Weite Verwendung finden

- **PENMAN** bzw. dessen Nachfolger **KPML**, das auf *systemisch-funktionalen Grammatiken* wie **NIGEL** für das Englische aufsetzt (s. *Bateman 1997*)⁹
- **FUF/SURGE**, das auf der funktionalen Unifikationsgrammatik aufsetzt (s. *Elhadad und Robin 1996*)
- **REALPRO** der Firma **CoGENTEX** das auf der *Meaning-Text-Theorie* (als einer Form der *Dependenzgrammatik*) aufsetzt (s. *Lavoie und Rambow 1997*)

Neuere Generatoren sind hingegen eher weniger an zugrundeliegenden Grammatiktheorien orientiert, sondern eher an praktischen Anforderungen wie Effizienz und Robustheit, z. T. auch im Sinn einer effektiven „quick and dirty“ Herangehensweise und mit Methoden flacher Sprachverarbeitung. Beispiele hierfür sind

- **TG/2** (s. *Busemann 2005*)
- **YAG** (Yet Another Generator, s. *McRoy et al. 2001; McRoy et al. 2003*)
- **SIMPLENLG**, eine praxisorientierte Realisierungskomponente (s. *Gatt und Reiter 2009*) ohne spezifischen grammatischen Hintergrund und mit vergleichsweise beschränkterer Abdeckung

11.3 Aspekte und Anforderungen

NLG involviert grundsätzlich die Entscheidungen

1. **was** zu sagen ist (*what to say*: **Inhaltsfestlegung**, d.h. Auswahl des Inhalts) sowie **wann was** gesagt wird (*when to say what*: **Inhaltsorganisation**, d.h. Planung der Struktur)
2. **wie** das zu Sagende ausgedrückt wird (*how to say*: Versprachlichung des Inhalts)

Hinter diesen Aufgaben versteckt sind zwei weitere Aspekte, deren Bedeutung gar nicht stark genug betont werden kann: die Entscheidung, was **nicht** gesagt werden soll (letztendlich gibt es immer etwas), und die Überbrückung der sogenannten **Generierungslücke** (*generation gap*). Mit letzterem ist der Umstand gemeint, dass, wenn eine bestimmte nicht-sprachliche Inhaltsstruktur gebildet/ausgewählt wurde, auch sichergestellt wird, dass sie sinnvoll durch Sprache ausgedrückt werden kann. Dieses Problem trat in frühen NLG-Systemen auf, die eine Zweiteilung bestehend aus Inhaltsbestimmung und -strukturierung einerseits sowie der rein sprachlichen Realisierung andererseits vornahmen.

⁹ dieser Ansatz scheint auch die längste Tradition zu haben, s. *Henrici (1965)*.

Zur Überbrückung der Generierungslücke weisen aktuelle Systeme daher typischerweise eine Dreiteilung wesentlicher Komponenten auf:

1. **Makroplanung**: Inhaltsbestimmung/-festlegung, Planung der Grobstruktur eines Textes anhand inhaltlicher Kriterien
2. **Mikroplanung**: Planung der Feinstruktur eines Textes auch anhand sprachrelevanter Kriterien
3. **(Oberflächen-)Realisierung**

Wie in anderen Anwendungen der Sprachtechnologie existiert auch in der NLG eine Unterscheidung zwischen tiefer und flacher Verarbeitung. Tiefe Verarbeitung ist dabei üblicherweise mit dem Begriff der **Planung** assoziiert, d. h. mit der Frage, auf welche Weise ein bestimmtes (kommunikatives) Ziel erreicht werden kann. Planung bedeutet hier, die situationsabhängig „beste“ Abfolge von (kommunikativen) Aktionen zu finden, um zum Ziel zu kommen. Verallgemeinert bedeutet dies, auf verschiedenen Ebenen die Bedingungen kennen zu müssen, die für die Oberflächenform eines Textes „verantwortlich“ sind.

Flache Verarbeitung ist eher mit dem Begriff des **Schemas** assoziiert, also mit eher starren Repräsentationsstrukturen, die das Wissen darüber darstellen, was auf einer bestimmten Ebene *typischerweise* an Struktur vorliegt. Dies bedeutet im Extremfall auch die Verwendung vorgefertigter Textstücke (*canned text*) oder mit spezifischen Informationen zu füllenden Text-Schablonen (*templates*), aus denen ein generierter Text zusammengesetzt wird.

Tiefe planungsbasierte Verfahren der NLG sind prinzipiell dynamischer, flexibler und gegenüber eher starren Schema-basierten Verfahren besser in der Lage, auf kontextspezifische Feinheiten zu reagieren. Allerdings sind sie auch komplexer und somit weniger praktikabel.

Oft stellen reale Daten den Input für ein Sprachgenerierungssystem dar (wie bei der Generierung von Wetterberichten).¹⁰ In diesem Fall gibt es grundsätzlich zwei Möglichkeiten, den Generierungsprozess zu gestalten: Entweder an den Daten orientiert, mit der Vorgabe, relevante Information aus ihnen zu ziehen und zu vermitteln (*bottom-up Ansatz*), oder ausgehend von Ziel-/Struktur-orientierten Vorgaben jeweils relevante Information aus den Daten zu ziehen und dann zu verbalisieren (*top-down Ansatz*). Häufig wird in der NLG ein Schema-gesteuerter top-down Ansatz verfolgt.

Die Aufgabe der **Mikroplanung**, dem Bindeglied zwischen Makroplanung und Realisierung, lässt sich am Besten anhand eines Beispiels verdeutlichen. Nachfolgend ist eine explizite (aber eben nicht adäquate) Verbalisierung einer bestimmten Gegebenheit¹¹.

Es gab ein Ereignis e1, in dem ein süßer Teig durch Erhitzen im Ofen in ein Objekt o mit fester Konsistenz überführt wurde, das zum Verzehr durch Menschen geeignet

¹⁰ Allerdings nicht immer, z. B. im Fall eines multimedialen Beschreibungssystems von Museumsgegenständen. Hier liegt die Information in Form repräsentierten Wissens über den Gegenstandsbereich vor.

¹¹ wobei dies bereits eine vereinfachte Darstellung ist. Eine adäquate formale Repräsentation der Gegebenheit würde sehr viel elaborierter ausfallen müssen, eine direkte Verbalisierung somit noch umfangreicher.

ist. Eine Person p1 namens 'Peter' war Verursacher dieses Ereignisses. Eine Person p2 namens 'Anke' war Verursacherin dieses Ereignisses. In zeitlicher Nähe folgte ein zweites Ereignis e2, bestehend aus zwei überlappenden Teilereignissen e3 und e4 des Einverleibens von o, jeweils mit p3 namens 'Peter' und p4 namens 'Anke' als Agenten (wobei p1 und p3 sowie p2 und p4 miteinander identisch sind), das zur Nicht-Existenz von o führte.

Um eine angemessenere Beschreibung dieser Gegebenheit (z. B. *Peter und Anke backten einen Kuchen und aßen ihn danach ganz auf.*) zu generieren, müssen die folgenden Aufgaben erledigt werden:

- **Lexikalisierung:** die Wahl sprachlicher Ausdrücke für nicht-sprachliche Inhalte, z. B. *backen* für 'Teig durch Erhitzen im Ofen in ein Objekt mit fester Konsistenz überführen', *Kuchen* für das resultierende Objekt usw.
- **Aggregation:** Das Zusammenfassen/Gruppieren ähnlicher Information/gleicher Aussagen, um z. B. koordinierte Strukturen wie *Peter und Anke* zu generieren
- **Generierung referierender Ausdrücke:** angemessene varierende sprachliche Beschreibungen derselben Objekte, z. B. die Wahl von „einen Kuchen“ für die erste, und des Pronomens *ihn* für die zweite Referenz auf den Kuchen.

11.4 Architekturen und Systeme

Die meisten der bis heute entwickelten NLG-Systeme (von denen viele in einem [NLG-Wiki](#) aufgelistet sind, weisen eine Architektur auf, die sich grob am Ablauf der geschilderten Aufgaben der NLG orientiert (sog. **Pipeline-Architektur**, s. Abbildung 11.1 auf der nächsten Seite) und in der sowohl Kontrolle als auch konstruierte Datenstrukturen sequentiell diesem Ablauf folgen (mit Ausnahmen, wie z. B. der integrierten Sprachgenerierung in Appelt's **KAMP**, s. hierzu *Reiter und Dale 2000*)¹².

Das **RAGS** (Reference Architecture for Generation Systems)-Projekt beschäftigte sich explizit mit Fragen der NLG-Architektur. Dabei wurde festgestellt, dass angewandte NLG-Systeme zwar überwiegend eine Pipeline-Architektur aufweisen, dass die Reihenfolge der abgearbeiteten Aufgaben jedoch keinesfalls identisch ist. Als Alternative wurde ein Vorschlag entwickelt, der einerseits eine flexiblere Kontrollfluss zwischen eigenständigeren Komponenten vorsieht und andererseits eine unabhängige Informationsverwaltungskomponente annimmt, die über verschiedene Daten-/Repräsentationsformate und deren Aufbau Buch führt. Als ein Beispiel wurde das System **RICHES** entworfen (zur Generierung von Gebrauchsanweisungen bestimmter Medikamente wie z. B. Nasentropfen, s. *Cahill et al. 2001*). Abbildung 11.2 auf Seite 177 zeigt eine vereinfachte Darstellung der Architektur dieses Systems.

¹² s. *Bateman und Zock (2003)* zu einer Version mit stärker interagierenden Komponenten.

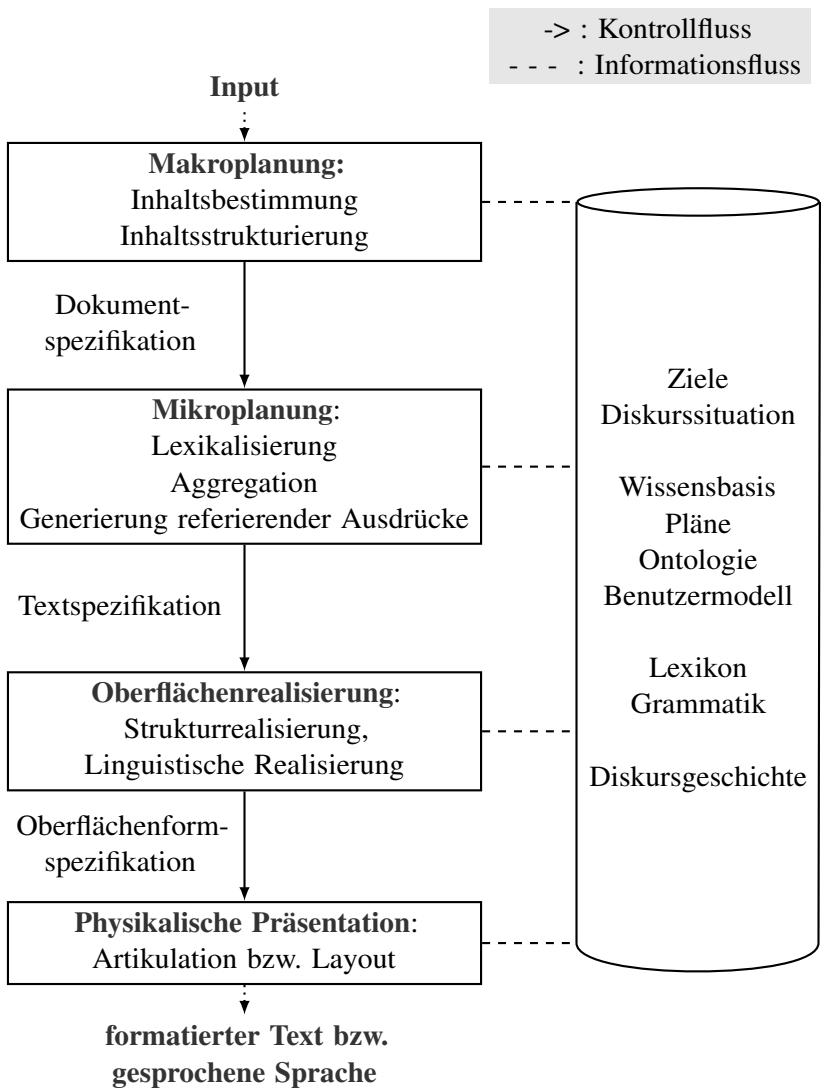


Abbildung 11.1: Standard (Pipeline)-Architektur eines NLG-Systems

11.5 Evaluation

Angeregt durch die *Shared Task Evaluation Conferences* (STEC) wie MUC und DUC in anderen Bereichen der Sprachverarbeitung wurde Ende der 1990er (s. z. B. Mellish und Dale 1998) und insbesondere in den 2000er Jahren das Thema „Evaluation“ (s. den *Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*) zunehmend in der NLG diskutiert.

Mellish und Dale (1998) weisen unter anderem darauf hin, dass anders als in anderen Bereichen der Sprachverarbeitung weder eindeutig ist, was der *Input* (Daten?, Wissensstrukturen?) eines NLG-Systems ist, noch was als *Output* (keine „richtige Antwort“ spezifizierbar) zu erwarten ist (s. auch Scott und Moore 2006 zu einer skeptischen Einschätzung).

Aus dem ersten Konsens des genannten Workshops, die Generierung referierender Aus-

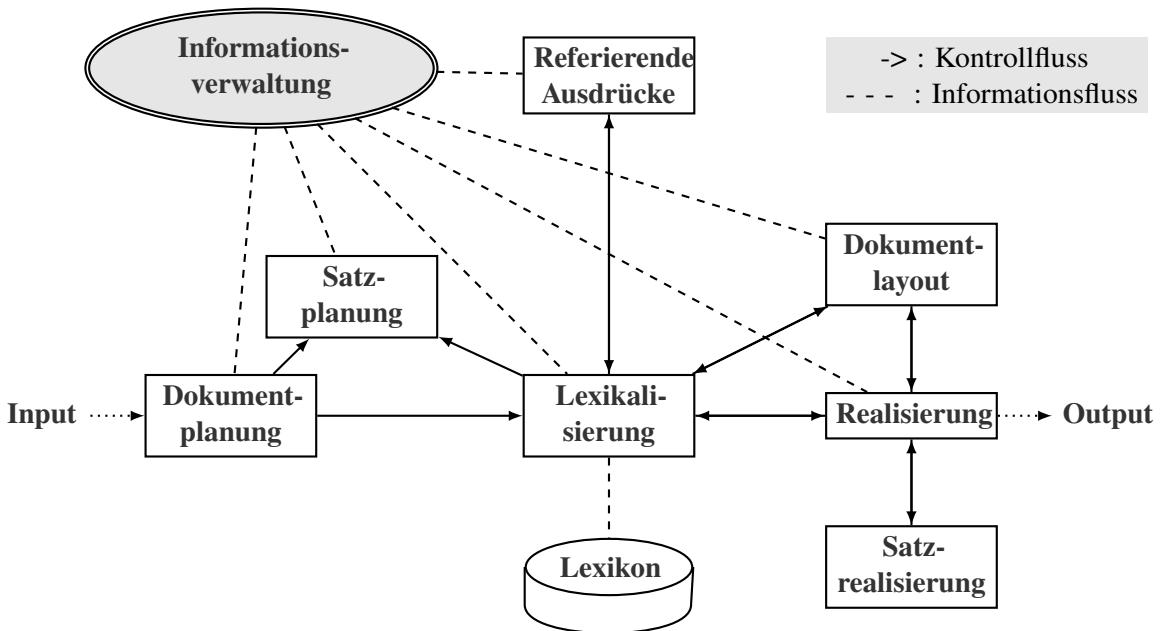


Abbildung 11.2: **RICHES-Architektur** (nach Cahill et al. 2001)

drücke (GRE) als charakteristische Aufgabe für eine NLG-Evaluationskonferenz zu verwenden, wird 2007 die 1. NLG-STEC (**A**ttribute **S**election for **G**RE (**ASGRE**) **C**hallenge) durchgeführt, mit der Aufgabe der Merkmalsauswahl für die Beschreibung von Objekten aus einer Inputrepräsentation.¹³ Hierbei werden Daten aus dem **TUNA**-Korpus zu Möbel- und Personenbeschreibungen verwendet. Der ASGRE-EC verfügt über drei *submission tracks*, wobei der erste die eigentliche Aufgabe enthält, die beiden anderen zum Einreichen weiterer Eingaben sowie verschiedener Evaluationsmethoden genutzt werden können, um die STEC offen und nicht zu restriktiv zu halten.

Die 2. NLG-STEC (**R**eferring **E**xpression **G**eneration (**REG**) **C**hallenge 2008) bietet bereits acht submission tracks, von denen vier die Bearbeitung spezifischer Aufgaben erfordern. Dies sind neben der Merkmalsauswahl die *Realisierung* aus einer semantischen Repräsentation sowie eine Kombination beider (alle basieren auf dem TUNA-Korpus). Die vierte Aufgabe verwendet das **GREC-MSR**-Korpus¹⁴: gegeben ein Objekt, ein Diskurskontext und mehrere mögliche Beschreibungen, welche der Beschreibungen ist die adäquateste im Kontext?

2009 existieren vier STECs, die unter der Bezeichnung Generation Challenges 2009 zusammengefasst werden:

¹³ also beispielsweise die Auswahl von Merkmalen, wie sie im obigen Beispiel zu verschiedenen Beschreibungen des Objekts o führen.

¹⁴ Das GREC-MSR-Korpus besteht aus ~1000 einleitenden Abschnitten von Wikipedia-Artikeln, die bzgl. verschiedenen Referenzen auf den Gegenstand (*main subject references (MSR)*) eines Abschnitts per Hand annotiert wurden (s. Belz und Vargas 2007).

- **TUNA Challenge**, die die kombinierte Aufgabe der Generierung referierender Ausdrücke (Merkmalsauswahl+Realisierung) aus 2008 weiterführt (TUNA-REG).
- **GREC Challenges** mit zwei Aufgaben: einer Fortführung der GREC-MSR-Aufgabe¹⁵ sowie einer auf die Generierung von *Namen* (Named Entity Generation, NEG) beschränkten Aufgabe GREC-NEG.¹⁶
- **Generating Instructions in Virtual Environments (GIVE-1 Challenge)**: Hierbei interagiert ein Benutzer ähnlich einem Computerspiel mit einer virtuellen Umgebung, in der er ein Problem lösen muss. Ein NLG-System hat alle Information, dem Benutzer zu helfen und muss diese Information nun in Anweisungen an ihn umsetzen.

Die **Generation Challenges 2010** setzen die GIVE- und GREC- Challenges fort, es gibt zudem Aufgaben im Bereich Eigennamenerkennung (*named entity recognition*) und Fragegenerierung (*question generation*).

11.6 Literatur

Das Standardwerk zur (angewandten) NLG ist *Reiter und Dale* (2000). Eine kürzere Darstellung der NLG ist *Bateman und Zock* (2003). Weitere Beschreibungen von Textgenerierung und von angewandten NLG-Systemen liefern Helmut Horacek (*Horacek* 2009) bzw. John Bateman (*Bateman* 2009)¹⁷. *McRoy et al.* (2001) bietet einen sehr lesbaren vergleichenden Überblick über Realisierungskomponenten/Generatoren.

¹⁵ allerdings mit erweitertem GREC-2.0-Korpus (2000 Wikipedia-Artikel).

¹⁶ unter Verwendung eines eingeschränkten Korpus GREC-People von 1000 Wikipedia-Texten, die bzgl. der Erwähnung von Personen annotiert wurden.

¹⁷ s. auch seinen NLG-Überblick.

Peba-II

Dale et al. (1998) präsentieren ein System, das automatisch Beschreibungen von Tieren aus einer Wissensbasis erzeugt und als Webseiten darstellt. Mittels ebenfalls generierter Hyperlinks kann der Benutzer sich weiter informieren.

Das Besondere an Peba-II ist die Tatsache, dass diese Beschreibungen nicht nur vom allgemeinen Usertyp (Neuling/Experte) abhängig sind, sondern auch von der Interaktionsgeschichte (d.h., was schon bekannt ist). Dies bedeutet, dass das erneute Klicken auf denselben Link zu einer Webseite mit anders generiertem Text führen kann.

Hier ist ein Beispiel (von Maria Milosavljevic, s. [Quelle](#)) für eine von Peba-II generierte Webseite:

The Echidna

The Echidna, also known as the spiny Anteater, is a type of Monotreme that is covered in stiff, sharp spines mixed with long, coarse hairs. Although it is similar in appearance to the African Porcupine it is not closely related. The African Porcupine is a type of Rodent that has long sharp spines, up to 50cm long, which cover its whole back and can be raised by muscles under the skin. Like the African Porcupine, the Echidna has a brownish black coat and paler-coloured spines. The African Porcupine is twice the length of the Echidna (80.0 cm vs 47.5 cm). The Echidna has an average weight of 4.5 kg whereas the African Porcupine has an average weight of 25.0 kg. The Echidna is a carnivore and eats ants, termites and earthworms whereas the African Porcupine is a herbivore and eats leaves, roots and fruit.



The Echidna has the following subtypes:

- the short-beaked Echidna and
- the long-beaked Echidna.

The Echidna is about the same length as a domestic cat. It ranges from 2 kg to 7 kg in weight. It has a brownish black coat and paler-coloured spines. It has a small head. It has a prolonged, slender snout. It has no teeth. It uses its extensible, sticky tongue for catching ants, termites and other small insects. It is a carnivore and eats ants, termites and earthworms. It has powerful claws allowing for rapid digging of hard ground. It is found in Australia. It is active at dawn and dusk. It lives by itself. It has an average lifespan in captivity of 50 years.

This text is generated for the novice user level. If you would like the text for the expert user level click [here](#).

Kapitel 12

Maschinelle Übersetzung (MÜ)

Ack ack ack ack-ack!

Don't run, we are your friends!

(Maschinelle Übersetzung in dem Film *Mars Attacks*)

MÜS: Maschinelle Übersetzung System

MT: Machine Translation (=MÜ)

FAT: Fully Automatic Translation

FAHQ(M)T: Fully Automatic High Quality (Machine) Translation

CAT: Computer Aided Translation

MAHT: Machine Aided Human Translation

HAMT: Human Aided Machine Translation

Plakativ ausgedrückt gibt es **die** maschinelle Übersetzung genauso wenig wie **die** Raumfahrt. Zwar fliegen wir zum Mond und zum Mars (und Sonden sogar an den Rand des Sonnensystems), jedoch rechtfertigt dies eigentlich den Terminus „Raumfahrt“ nicht.

Auf „maschinelle Übersetzung“ trifft dasselbe zu: Sie ist weit davon entfernt, vollautomatisch für beliebige Eingaben Ergebnisse in hoher Qualität (FAHQMT) zu liefern.

Dessen ungeachtet leistet sie gute Arbeit bei realistischen Ansprüchen, und es bleibt die Hoffnung, dass sie doch machbarer ist als Raumfahrt...

12.1 Allgemeines zur MÜ: Mythen und ihre Klärung

Die Maschinelle Übersetzung ist so alt wie die CL selbst. Kaum eine Anwendung der CL ist so bekannt und gleichzeitig eine solche terra incognita für den Laien. Entsprechend ranken sich viele Gerüchte und Mythen um den Wert und die Fähigkeit der MÜ (s. Arnold et al. 1995).

- **Zeitverschwendug (M1):**
„MÜ ist Zeitverschwendug, da man niemals eine Maschine wird bauen können, die Shakespeare übersetzen kann“
- **Versagen (M2):**
„Es gab/gibt ein MÜS, das *The spirit is willing, but the flesh is weak* in das russische Equivalent von *The vodka is good, but the steak is lousy* und *hydraulic ram* ins französische Equivalent von *water goat* übersetzte. MÜ ist nutzlos, q.e.d.“
- **Nutzlosigkeit (M3):**
„Generell ist die Übersetzungsqualität von MÜS sehr niedrig. Das macht sie praktisch nutzlos.“
- **Bedrohung (M4):** „MÜ bedroht die Arbeitsplätze von Übersetzern.“
- **Bereits erledigt (M5):**
„Die Japaner haben ein System entwickelt, mit dem du übers Telefon reden kannst. Es übersetzt alles, was du sagst auf Japanisch und die Erwiderungen zurück ins Deutsche.“
- **Elegante Lösung existiert (M6):**
„Es gibt da eine ganz erstaunliche südamerikanische Indianersprache mit einer Struktur von solch logischer Perfektion und algebraischer Klarheit, dass sie ein zentrales Problem bei der Entwicklung von MÜS (das Finden einer einheitlichen Zwischensprache (*Interlingua*) löst.“
- **Stangenprodukt (M7):**
„MÜS sind Maschinen, und der Kauf eines MÜS soll so einfach ablaufen wie der Kauf eines Autos.“

Ad M1: Sicherlich ist es heute immer noch utopisch und generell möglicherweise unmöglich, Shakespeare (und vergleichbare literarische Autoren) zu übersetzen. Allerdings ist die Folgerung falsch. Die Relevanz der MÜ muss nicht von vermeintlichen Maximalzielen abhängig gemacht werden. Wie Arnold et al. (1994:6) es treffend formulieren: „The criticism that MT systems cannot translate Shakespeare is a bit like criticism of industrial robots for not being able to dance Swan Lake“.

Ad M2: Tatsächlich ist es heutzutage leicht, Schnitzer der MÜ live per online-Übersetzung zu erleben. Die „klassischen“ Beispiele für das Versagen der MÜ sind jedoch größtenteils auf verschlungenen Wegen entstandene und weitergeleitete Hörensagen-Geschichten.¹ Wieder sagt dies nichts über die Nützlichkeit der MÜ aus.

¹ s. hierzu John Hutchins, „’The whiskey was invisible’, or persistent myths of MT“. *MT News International* 11 (June 1995), 17-18.

Ad M3: Das **MÉTÉO**-System (Übersetzung von Wettervorhersagen Englisch –Französisch und Französisch– Englisch) war zwischen 1981 und 2001 in täglichem, kommerziell genutztem Einsatz. Dabei übersetzte es durchschnittlich 45000 Wörter täglich. In den 1980ern sparte der Dieselmashinenhersteller Perkins Engines mit einem MÜS etwa 4000 £ und bis zu 15 Wochen an jedem übersetzten Manual. Das heißt, wenn MÜ-Systeme manchmal auch einige Böcke schießen, so gibt es viele Situationen, in denen die Fähigkeit von MÜS, vernünftige (wenn auch nicht perfekte) Übersetzungen in kurzer Zeit zu bringen, wertvoll ist.

Ad M4: Nein. Einerseits ist der Bedarf an Übersetzung riesig und wird kaum abnehmen. Beispielsweise betragen die Ausgaben für Übersetzung in den EU-Ländern in 1995 500 Mio Euro, in 2000 2 Mrd Euro. Gleichzeitig stieg der weltweite Umsatz der Übersetzungsdienstleister von 2,981 Mrd US-Dollar in 2001 auf geschätzte 6 Mrd US-Dollar in 2006.² Aktuellere Schätzungen liegen eher über diesen Prognosen.³.

Andererseits sind die Beschränkungen von MÜS gegenwärtig so groß, dass sie eher einige der mühsamen, langweiligen, sich wiederholenden Übersetzungsjobs übernehmen und menschlichen Übersetzern somit erlauben, sich auf die interessanteren Aufgaben zu konzentrieren, bzw. darauf, wo die Fähigkeiten der Spezialisten wirklich gebraucht werden.

Ad M5: Generelle Übersetzung gesprochener Sprache ist pure Science Fiction, da sie noch komplexer und komplizierter als textuelle MÜ ist. Zwar wird an Dolmetsch-Systemen aktuell geforscht (s. z. B. das Projekt **VERBMOBIL** und das **TC-STAR**-Project), generell gibt es aber noch (zu) viele Probleme, die gelöst werden müssen, bevor MÜS in Reichweite der Fähigkeiten menschlicher Übersetzer kommen.

Ad M6: Dieser Mythos hat einen realen Ursprung (s. <http://www.atamiri.cc/>) und taucht regelmäßig in der Presse auf. Er ist deswegen selbst für Fachleute irritierend, da die Idee einer Interlingua, die zwischen den zu übersetzenden Sprachen vermittelt, zum Kern der MÜ gehört (es könnte ja jemand einen Durchbruch erzielt haben).

Allerdings ist der Beleg für die Behauptung in M6 nie erbracht worden (keine Ressourcen, nicht genug Sponsoren, nicht ausgereift etc.). Dies erinnert an die „kalte Fusion“, in der Garage entwickelt. Zudem zeugt M6 eher von Unkenntnis der Probleme der MÜ: Sprachverarbeitung involviert auf verschiedenen Ebenen subtilste Aspekte der Modellierung, da hilft keine ‚einfache‘ Zwischen-Sprache und entsprechend gibt es keine ‚magische Lösung‘. Die Wahrscheinlichkeit, dass eine natürliche Sprache (idiosynkratisch wie diese nunmal sind) als perfekte Interlingua dienen kann, geht gegen Null. Ähnliches gilt für Esperanto (das im DLT-Projekt als Interlingua verwendet wurde, s. Kapitel 17 in *Hutchins und Somers* 1992).

Ad M7: Nein, sie können nicht ‚nach Katalog‘ oder ‚von der Stange‘ gekauft werden. Jenseits der Entwicklungsprobleme der MÜ werden Benutzer immer eine Menge an Konfigurationsaufwand investieren müssen, bevor sich ein MÜS zweckgemäß und effektiv als eine Software-Komponente in einem spezifischen Workflow einsetzen lässt.

² Quelle: <http://www.tanner.de/de/infoportal/infoportalDetails.php?objectID=160>.

³ s. den von der EU finanzierten Report „Sprachtechnologien für Europa“, April 2006.

Den negativen Mythen stehen die **Argumente für MÜ** gegenüber, u.a.

- die effiziente Bewältigung der Aufgaben in multilingualen 'communities' (z. B. der EG) sowie bei globalen Produktions- und Vertriebsabläufen
- die notwendige (weil von Kunden gewünschte) Lokalisierung von Info-Material
- die zu geringe Anzahl menschlicher Übersetzer
- der hohe Preis für Übersetzung: es werden 4-6 Seiten Übersetzung (~2000 Wörter) pro Tag pro Übersetzer angesetzt; 40-45% der laufenden Kosten der EG- Institutionen sind 'Sprach-Kosten'

12.2 Geschichte

Ein wesentlicher Anfangsschritt in der Geschichte der MÜ stellt ein (als „**Weaver Memorandum**“ bekannt gewordener) Text von Warren Weaver dar, in dem er, auf den Erfolg der Code-Entschlüsselung im 2. Weltkrieg verweisend, die prinzipielle und systematisch zu erreichende Machbarkeit der MÜ darstellt:

I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.

(aus: Warren Weaver, „Translation“)⁴

Hieraus resultierte –insbesondere nach einer nach außen hin erfolgreichen MÜ-Demonstration an der Georgetown University– eine Zeit intensivster Forschungsförderung. Diesem Boom folgte nach kritischen Worten von Bar-Hillel der desaströse **ALPAC-Report**, der zum Erliegen der Förderung in den USA führte.

1949:	Weaver Memorandum
1950er:	Erste öffentliche MÜ-Demonstration (GU-IBM-Experiment) in 1954. Danach groß angelegte Förderung der MÜ in den USA, der Sowjetunion, Europa, Japan. Die Vision hierbei war FAHQT beliebiger Texte ohne Nachbereitung.
1960:	Bar-Hillel behauptet, FAHQT sei 'unmöglich' (<i>Bar-Hillel 1960</i>). Sein Argument: die Ambiguität der Sprache und die zu der Zeit als Unmöglichkeit einzuschätzende Auflösung durch repräsentiertes Weltwissen. Sein Beispiel: „John was looking for his toy box. Finally he found it. The box was in the pen. John was very happy.“ Hier muss ein System die Lesart 'Behältnis, Umzäunung' für „pen“ auswählen, das sei „unrealisierbar“.

⁴ abgedruckt in: Locke, W.N. and Booth, A.D., Hrsgs., *Machine translation of languages: fourteen essays* (Cambridge, Mass.: Technology Press of the Massachusetts Institute of Technology, 1955), S. 15-23.

	Hierzu ist anzumerken, dass das Problem immer noch besteht und aktuell ist, dass es aber nach Jahrzehnten der Erforschung von Wissensrepräsentation und Ontologien als weit weniger unmachbar erscheint.
1966:	ALPAC (Automatic Language Processing Advisory Committee) Report: Die MÜ(-Forschung) wird als Misserfolg gedeutet (zu langsam, zu ungenau, zu teuer), die Förderung wird zusammengestrichen. Gleichzeitig wird empfohlen, die Grundlagenforschung der maschinellen Sprachverarbeitung voranzutreiben, im Prinzip die Geburtsstunde der Computerlinguistik.

Der ALPAC-Report hatte außerdem zur Folge, dass der Fokus weg von FAHQT und hin zu einer differenzierteren Betrachtung der Rolle der MÜ gewendet wurde. Hierzu gehören

- die Berücksichtigung der tatsächlichen Bedürfnisse bei den Kunden, nämlich die Unterscheidung zwischen der *Verbreitung* (*dissemination*) übersetzter Dokumente (was gute Qualität erfordert), der *Sichtung* (*assimilation*) von Texten (wofür Grobübersetzungen, wie sich von MÜS geleistet werden können, ausreichend sind) und der Übersetzung (gesprochener Sprache) für Informationsaustausch.
- die Unterscheidung von FAT vs. HAMT oder MAHT (wann ist was sinnvoll/notwendig?)
- die Anerkennung der Expertise von Übersetzern, des 'Work-flows' bei der Übersetzung usw.
- die Etablierung der Auffassung einer Mehrgleisigkeit der MÜ: Übersetzungstools als Hilfsmittel von Übersetzern; lauffähige Systeme mit beschränkten Fähigkeiten (zur Nachbearbeitung, mit eingeschränkter, kontrollierter Sprache, in spezifischen Bereichen/Domänen); FAT.

Weitere Meilensteine der MÜ sind die folgenden:

1970:	SYSTRAN bei der USAF (Russisch↔Englisch, ca. 100000 Seiten Text pro Jahr zu Sichtungszwecken) eingeführt
1976:	EG kauft SYSTRAN (E↔F)
1977:	MÉTÉO (E↔F)- sub-language system (Wettervorhersagen, Montreal)
1970er Jahre:	Forschung in Grenoble (ARIANE), Saarbrücken (SUSY), Austin (METAL), Kyoto (MU)
1980-90:	EG fördert multilinguales MÜ-Projekt EUROTRA
späte 1980er:	erste kommerzielle MÜ-Systeme, z. B. METAL
1990er-2000:	VERBMOBIL (Projekt zur Übersetzung gesprochener Sprache, Kombination von Regel- und Statistik-Ansatz)
heute (z. B.):	Langenscheidts T1 (früher: METAL , allerdings nun um einen Faktor 100 günstiger), IBMs POWERTRANSLATOR , Web-MÜ-Dienste

12.3 Aspekte und Anforderungen

Es existieren zwei Typen von Anforderungen, die bei der Entwicklung von MÜS zu berücksichtigen sind. Dies sind zunächst solche, die sich aus den sprachlichen Phänomenen ableiten, die bei der MÜ (bzw. der Übersetzung allgemein) auftreten. Sie stellen die klassischen Probleme der CL dar: Ambiguität (als Analyseproblem), lexikalische Auswahl (als Generierungsproblem) und die sprachlichen Unterschiede jeweils zweier Sprachen (als spezifisches Übersetzungsproblem). Auf der Analyseseite gibt es verschiedene **Arten der Ambiguität** (s. auch den Exkurs auf Seite 36):

- **Syntaktische Ambiguität:**

Das Problem des Anhängens der Präpositionalphrase (PP-attachment) in die syntaktische Struktur. Das heißt, hat der Mann das Teleskop oder wird der Mann mit Hilfe eines Teleskops gesehen?

E: *I saw the man with a telescope*

- **Lexikalische Ambiguität:** Um welche Wortart handelt es sich?

E|S: *book | libro, reservar*

- **Semantische Ambiguität:** Welche Lesart eines Wortes ist gemeint?

- **Homografie** (Beziehung zwischen den Lesarten ist etymologisch nicht (mehr) direkt ersichtlich):

E|S: *ball | pelota, baile*

- **Polysemie** (systematische Beziehungen zwischen den Lesarten):

E|S: *kill | matar, acabar*

- **Komplexe semantische Ambiguität:**

Ihre Auflösung erfordert oft ein tieferes „Verstehen“, d.h., Zugriff auf typisches Wissen über die beteiligten Entitäten (Bar-Hillels Beispiel, dass eine Kiste nicht in einem Kugelschreiber enthalten sein kann).

- **Homografie:** (s.o.)

E: *The box was in the pen*

S: *La caja estaba en el corral / *la pluma*

- **Metonymie:** Nicht John selbst, sondern das Auto

E: *While driving, John swerved and hit the tree*

S: *Mientras que John estaba manejando, se desvió y golpeó con un arbol ('While John was driving, (itself) swerved and hit with a tree')*

- **Kontextuelle Ambiguität:**

Die Notwendigkeit, Information zur Auswahl einer Lesart aus dem Kontext zu ziehen (anhand der vorkommenden Objekttypen).

E: *The computer outputs the data; it is fast*

S: *La computadora imprime los datos; es rápida*

E: *The computer outputs the data; it is stored in ascii*

S: *La computadora imprime los datos; están almacenados en ascii*

- **Komplexe kontextuelle Ambiguität:**

In manchen Fällen reicht die Berücksichtigung von Typenwissen und typischem Wissen nicht aus. Dann muss auf den Kontext rekuriert werden, in dem möglicherweise Information bereits zur Verfügung gestellt wurde (hier: wer einen Stock hat).

E: *John hit the dog with the stick*

S: *John golpeó el perro con el palo / que tenía el palo*

Auf der Generierungsseite existiert das Problem der **Lexikalischen Auswahl**: Für eine bestimmte Lesart eines Wortes in der Quellsprache ist es oft nicht trivial, das geeignete Wort der Zielsprache auszuwählen,⁵ wie z. B. bei

S|E: *esperar | wait, hope*

D|E: *verstehen | know, understand*

Ein anderes Auswahlproblem besteht dann, wenn der Ausdruck der Quellsprache *underspezifiziert* bzgl. der Zielsprache ist. Dieses Phänomen tritt z. B. bei der Tempusgenerierung auf. So ist im Chinesischen Tempusinformation nicht explizit gegeben und es muss inferiert werden, ob es sich um einen Bericht von etwas Vergangenem oder eine aktuelle Beschreibung in der Gegenwart handelt:

C: *Wǒ bì Hángzhōu de fēngjǐng xīyǐnzhù le*

E: *I was captivated by the scenery of Hangchow*

E: *I am captivated by the scenery of Hangchow*

In ähnlicher Weise muss in einigen Sprachen die Abgeschlossenheit eines beschriebenen Ereignisses nicht notwendigerweise ausgedrückt werden, während dies in anderen Sprachen obligatorisch ist. Im folgenden Beispiel bedeutet der zweite Satz im ersten Fall eine Elaboration (keine Weiterentwicklung in der Zeit, Imperfektivität), im zweiten eine Fortführung (und somit Abgeschlossenheit des vorangegangenen).

E: *Mary went to Mexico. During her stay she learned Spanish.*

S: *Mary iba a Mexico. Durante su visita, aprendió español.*

E: *Mary went to Mexico. When she returned she started to speak Spanish.*

S: *Mary fue a Mexico. Cuando regresó, comenzó a hablar español.*

Eine dritte Klasse linguistischer Phänomene umfasst spezifische **Unterschiede** zwischen Quell- und Zielsprache:

⁵ Selbst naheliegende Übersetzungen können daher inkorrekt sein, wie z. B. *We excuse us* für *Wir entschuldigen uns*.

- **Thematische Divergenz:**

Die Auswahl eines Arguments einer Relation als Thema des Satzes.

E: *I like Mary*

S: *Mary me gusta*

- **Kopf-Wechsel-Divergenz:**

Die Bestimmung, welche semantische Information syntaktisch als Kopf einer Phrase (und welche als Argument bzw. Modifikator einer Phase) realisiert wird.

E: *I like to eat*

D: *Ich esse gern*

- **Strukturelle Divergenz:**

Z.B. Unterschiede in den Komplementen einander entsprechender Verben (NP vs. PP).

E: *John entered the house*

S: *Juan entró en la casa*

- **Kategorische Divergenz:**

Korrespondierende Wörter sind von unterschiedlicher sprachlicher Kategorie (hier: Adjektiv vs. Nomen), was Anpassungen (z. B. die Art des Verbs) nach sich zieht.

E: *I am hungry*

D: *Ich habe Hunger*

- **Konflationelle Divergenz:**

Ähnlich der strukturellen Divergenz werden gleiche Aspekte in verschiedenen Sprachen oft unterschiedlich stark zusammengefasst oder verdichtet.

E: *I stabbed John*

S: *Yo le di puñaladas a Juan* ('I gave knife-wounds to John')

Dabei liegt der Teufel generell im Detail. Zur Veranschaulichung hierfür wurde ein Beispielsatz von fünf Online-MÜ-Diensten übersetzt (s. Abb. 12.1 auf der nächsten Seite). Es zeigt sich, dass selbst bei einfachen Sätzen eine Spannbreite an Variation gegeben ist (verbunden mit Qualitätsunterschieden).

Der zweite Typ von Anforderungen an MÜS betrifft **die zu erwartenden Eigenschaften** der MÜS selbst (operationelle Anforderungen). Dies sind

- Erweiterbarkeit an neue Domänen und Sprachen:
- Fähigkeit, unterschiedliche Textstile/-typen zu behandeln
- einfache Pflege des Systems
- Einbettbarkeit in Workflow/ Integration mit anderer Benutzersoftware
- Evaluierbarkeit der Effektivität des Systems

Dies ist eine Software, die Texte in andere Sprachen übersetzt, ohne irgendwelche Fehler zu machen.

This is a software, which translates texts into other languages, without making any errors.
[\(http://babelfish.altavista.com/\)](http://babelfish.altavista.com/)

This is a software that translates text into other languages, without making any mistakes.
[\(http://translate.google.com/\)](http://translate.google.com/)

This is a software that translates texts into other languages without making any mistakes
[\(http://ets.freetranslation.com/\)](http://ets.freetranslation.com/)

This is software which translates texts into other languages without doing any mistakes
[\(http://translation2.paralink.com/\)](http://translation2.paralink.com/)

This is a software which translates texts into other languages without doing some mistakes
[\(http://www.online-translator.com/\)](http://www.online-translator.com/)

This is a software, which übersetzt texts into other languages, without making any errors
[\(<http://translator.live.com/Default.aspx> \(Windows Live Translator Beta\)\).](http://translator.live.com/Default.aspx)

Abbildung 12.1: Übersetzungsbeispiele, Stand Mitte 2009

12.4 Architekturen und Systeme

MÜS und ihre Architekturen lassen sich entlang zweier Dimensionen klassifizieren: der Tiefe der Verarbeitung und der Art der verwendeten Techniken. Anhand der Verarbeitungstiefe werden MÜS in Direkte, Transfer und Interlingua-Systeme unterteilt.

Wie Abb. 12.2 auf der nächsten Seite zeigt, führen Direkte MÜS Ausdrücke der Quellsprache nach oberflächlicher Analyse (Morphologie und einfachste Syntax) in die Zielsprache über, Transfersysteme mit zusätzlicher syntaktischer oder sogar semantischer Analyse, bei Interlingua-Systemen erfolgt eine Abbildung analysierter Sprache in eine sprachunabhängige Repräsentation (oder eine Zwischensprache), aus der wieder Zielsprachen-spezifische Strukturen generiert werden. Durch die Art und die Beziehung der jeweiligen Komponenten ist gleichzeitig die Architektur eines MÜS-Typs determiniert.

12.4.1 Verarbeitungstiefe

Direkte MÜS

Direkte MÜS gehören zu den ältesten Systemen für maschinelle Übersetzung (z. B. war das System der Georgetown University, aus dem später Systran hervorging, von diesem Typ),⁶ entsprechend sind sie recht einfach strukturiert. Meist handelt es sich um Wort-für-Wort-Übersetzungen, bei der eine morphologische Analyse vorgenommen wird. Hiermit werden immerhin Konstruktionen wie *It's* oder *Ein- und Ausgang* in die zugrundeliegenden Vollformen (*It is* und *Eingang und Ausgang*) abgebildet, so dass ihre ökonomische Übersetzung möglich

⁶ Allerdings existieren immer noch einfache MÜS, z. B. PC-Systeme, dieser Art

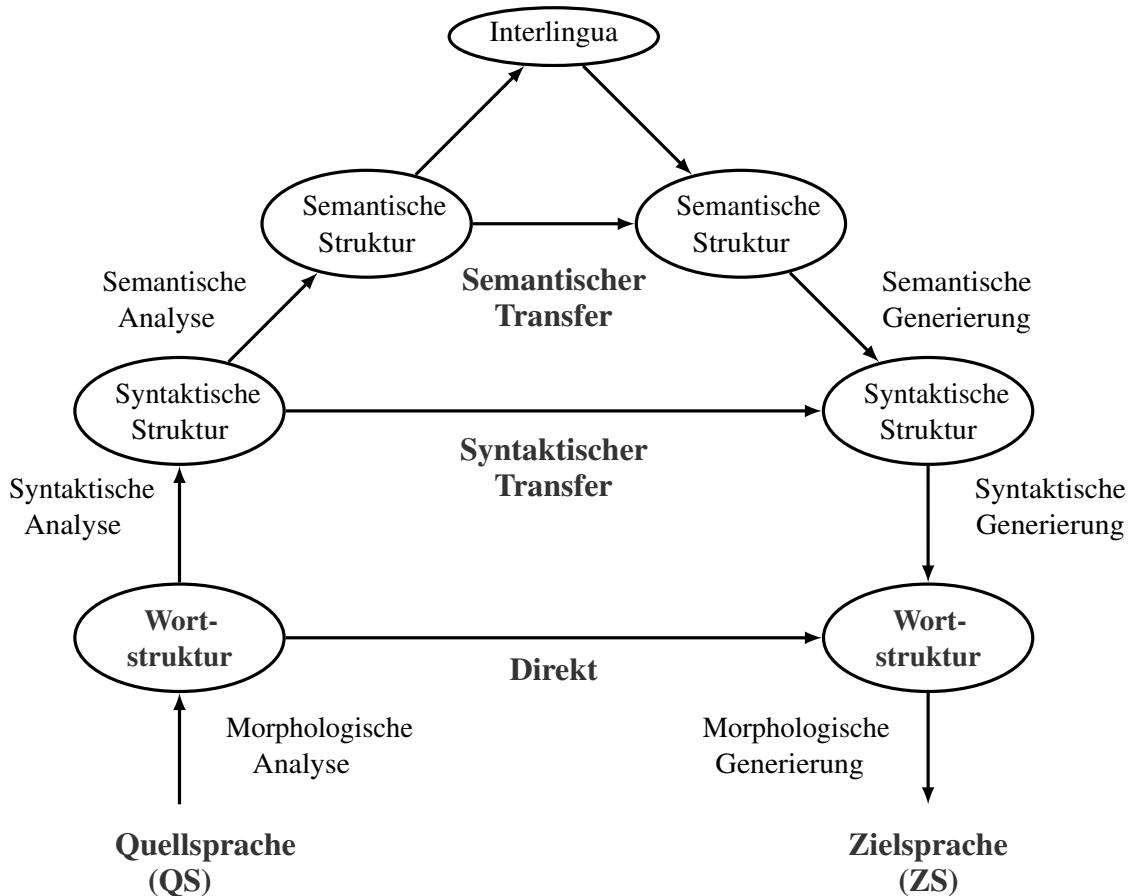


Abbildung 12.2: MÜ-Architekturen

wird. Ausschließlich Lexikon-basierte Übersetzung wird dadurch möglich, dass der lokale Kontext der Wörter berücksichtigt wird, z. B. mithilfe von Regeln wie

Much/many (E→D):

```

if preceding word is how return wieviel(e)
else if preceding word is as return soviel(e)
  
```

Da keine Syntaxanalyse vorgenommen wird, werden die syntaktischen Unterschiede zweier Sprachen durch Regeln repräsentiert, die ausschließlich über die beteiligten Wortkategorien definiert sind. Ein Beispiel ist die Adjektivstellung, die bzgl. dem Englischen (oder Deutschen) und dem Französischen divergiert:

E→F :

```

Eadj EN → FN Fadj
Preliminary installation → Installation prÈliminaire
  
```

Die Nachteile der Direkten MÜ sind zunächst die fehlende Abstraktion und damit allgemeinere Behandlung sprachlicher Phänomene. Hierdurch müssen unzählige Regeln geschrieben werden,

was einen großen Aufwand bedeutet und zudem eine erhebliche Unübersichtlichkeit nach sich zieht (von Nachteil für Erweiterungen und Pflege). Außerdem orientiert sich die Direkte MÜ zu direkt an der Quellsprache: komplexe, aber stilistisch adäquate Umstellungen sind so nicht möglich. Neben fehlenden Theorien für die Erstellung der Regeln und der gleichzeitigen Komplexität der Regelbearbeitungen ist Ambiguität das zentrale Problem der Direkten MÜS: da die Wortwahl in der Zielsprache ohne potentiell disambiguierenden Kontext vorgenommen muss, ergeben sich die entsprechenden Probleme bei der lexikalischen Auswahl.

Transfer-MÜS

Bei Transfer-Systemen geschieht die Abbildung zwischen den Sprachen nicht auf der Ebene der Lexeme, sondern der sprachspezifischen syntaktischen und/oder semantischen Strukturen. Hierdurch können bestimmte Arten der Ambiguität aufgelöst werden (lexikalische A. wie *book* als Verb oder Nomen). Gleichzeitig werden Generalisierungen möglich, welche die MÜS erheblich transparenter werden lassen. Grundlage dieser Art der Verarbeitung sind in der Regel unifikationsbasierte Grammatikformalismen wie LFG oder HPSG, mithilfe derer theoretisch motivierte Regeln erstellt werden können. Dadurch, dass nur Merkmalsstrukturen übertragen werden, können sprachspezifische Formen (wie die Genitivform im Französischen) modular in der Grammatik angegeben werden. Für deren Generierung werden dann keine Transferregeln benötigt.

D→F :

np (np (gen) , nbar ()) → np (det (def) , nbar () , np (gen)) [vereinfacht]
eines Königs Tochter → *la fille du roi*

Allerdings haben auch die Transfer-MÜS (d.h. die State-of-the-Art-symbolischen Systeme) eine Reihe von Nachteilen. So bringt die größere Komplexität der Transfer-MÜ eine entsprechend größere Fehleranfälligkeit mit sich: versagt eine Analysekomponente, ist der gesamte Übersetzungsvorgang in Mitleidenschaft gezogen. Transfer-MÜS sind daher weit weniger robust als Direkte MÜS. Wegen der Sprachspezifik des Transfers wächst die Anzahl der Transfermodule in etwa quadratisch ($n(n-1)$ Module bei n Sprachen), was einen enormen Aufwand für multilinguale Systeme bedeutet. Außerdem ist der Transfer abstrakter Strukturen nicht-trivial. Die Transfer-Regeln können sehr komplex (und somit unübersichtlich) werden. Sie sind dann zwar sehr viel kompakter als die Regeln der Direkten MÜ, aber ähnlich schwer zu warten und zu pflegen.

Interlingua-MÜS

Interlingua-MÜS verfügen über eine sprachneutrale Repräsentationsebene (Interlingua), mit der die sprachspezifischen Analyse- und Syntheseebenen über sogenannte „Verbindungsregeln“ (*linking/mapping rules*) verknüpft sind. Der wesentliche Vorteil dieses MÜS-Typs besteht in der geringen Kombinatorik bei der Übersetzung zwischen mehreren Sprachen: für n Sprachen werden insgesamt nur $2n$ MÜ-Module benötigt (für jede Sprache jeweils eines für die

Übertragung in die und aus der Interlingua). Tabelle 12.3 soll den Unterschied zu den beiden anderen MÜS-Typen veranschaulichen: Interlingua-MÜS sind immer dann attraktiv, wenn zwei

X LIKE Y → Y GUSTAR X	Direkt
gustar(SUBJ(ARG1:NP), OBJ1(ARG2:PREP)) → like(SUBJ(ARG2:NP), OBJ1(ARG1:NP))	Transfer
like/gustar: [CAUSE (X, [BE (Y, [PLEASED])])]	Interlingua

Tabelle 12.3: Verfahren der MÜ im Vergleich

Sprachen große strukturelle Unterschiede aufweisen (z. B. Chinesisch-Deutsch). Allerdings ist die Frage der Beschaffenheit einer geeigneten Interlingua immer noch nicht gelöst.

12.4.2 Verwendete Techniken

Bzgl. der bei der MÜ verwendeten Techniken können –wie generell in der CL und KI– drei allgemeine Klassen von MÜS unterschieden werden: Intellektuelle, Korpus-basierte und hybride Systeme als Mischung der ersten beiden Typen.⁷

Intellektuelle MÜS

Intellektuelle MÜS differenzieren sich nach den jeweils zugrunde liegenden Paradigmen für die Konzipierung der Repräsentation und Verarbeitung von Analyse, Synthese und Transfer. Folgende Paradigmen werden unterschieden:

- **Constraint-basierte MÜ (CBMT):** Im Wesentlichen LFG-basierte MÜ.
- **Wissensbasierte MÜ (KBMT):**
Interlingua-Ansatz mit starkem Bezug auf Weltwissen(srepräsentationen).
- **Lexikalisch basierte MÜ (LBMT):**
Überwiegend Lexikon-basierte Direkte oder Transfer-MÜ.
- **Regelbasierte MÜ (RBMT):** Kontrastiert mit LBMT.
- **Prinzipien-basierte MÜ (PBMT):**
An das Chomsky-Paradigma angelehnte MÜ, in der statt auf spezifische Regeln auf allgemeine Prinzipien zurückgegriffen wird.
- **Shake and Bake MÜ (S&BMT):**
Lexikalischer Transfer und Ausnutzung der Zielsprachengrammatik für die adäquate Ordnung der Zielsprachenwörter (dies als Lösung für Strukturveränderungen wie bei der Head-Switching-Divergenz).

⁷ Dorr et al. (1999) differenzieren hier in Linguistisch-/Nicht-linguistisch-basierte Systeme und deren Mischung.

Während diese Paradigmen im Allgemeinen auf FAT ausgerichtet sind (bei der wiederum keine hohe Qualität garantiert ist), existiert mit der **Dialog-basierten MÜ** (DBMT) ein Ansatz, der hochwertige Übersetzungen für monolinguale Benutzer (*casual users*) ermöglichen soll (daher auch: *personal MT*). Dies geschieht, indem in Bearbeitung befindliche Textstücke generell im Hintergrund verarbeitet werden, bei auftretenden Problemen (Ambiguitäten etc.) jedoch ein Klärungsdialog mit dem Benutzer geführt wird (Boitet 1997).

Korpus-basierte MÜS

Korpus-basierte MÜ setzt per nominem Datensammlungen übersetzter Texte (bilinguale bzw. *parallele* Korpora) voraus. Diese werden beispielsweise aus den Gesetzes- oder Parlaments-texten eines mehrsprachigen Landes wie z. B. Kanada oder den Dokumenten multinationaler Organisationen/Gemeinschaften wie der UN oder den EG erstellt. Aber auch die heutigen Suchmaschinen verfügen potentiell über ein großes Reservoir multilingualer Texte (vgl. Webseiten in verschiedensprachigen Ausgaben).

Die Verwendung dieser Korpora basiert auf einem Verarbeitungsschritt, in dem Textstücken der Quellsprache die ihnen entsprechenden der Zielsprache zugeordnet werden (man spricht hier von *Alignierung/alignment* dieser Textstücke). Die wesentlichen Paradigmen der MÜ in diesem Bereich sind dann

- **Beispiel-basierte MÜ** (EBMT)(auch: Case/ Memory/ Instance-based MT).

Dieser Ansatz beruht auf dem direkten Ausnutzen der im Korpus abgelegten Übersetzungen und involviert folgendes: a) der Suche nach einem Alignment, dessen QS-Teil dem zu übersetzenden (Teil-)Satz möglichst ähnlich ist, b) dem Anpassen des Alignments an den Satz und c) der Verwendung des modifizierten Alignments als Transfer-Relation.

- **Statistisch-basierte MÜ** (SBMT):

Hier wird die in den alignierten Texten enthaltene *Information* ausgenutzt. Dies sind a) die Wahrscheinlichkeiten, dass ein Wort der Zielsprache eine Übersetzung eines Wortes der Quellsprache ist (*translation model*) und b) die Wahrscheinlichkeiten, dass ein Wort der ZS auf ein anderes der ZS folgt ((*target- language model*)).

Die grundsätzlichen Probleme dieser Verfahren⁸ sind einerseits in der Abhängigkeit von vorliegenden bilingualen Korpora und andererseits ihre mangelnde Allgemeinheit. So existieren insgesamt nur wenige, auf bestimmte Domänen eingeschränkte Korpora, und selbst dann ist die Übersetzungsqualität noch nicht hochwertig und erfordert Nachbereitung. Zusätzliche Korpora müssen mit erheblichem Aufwand und/oder Qualitätsverlust (bei automatischem Alignment) erstellt werden. Gleichzeitig ist nicht garantiert, dass zusätzliche Daten die Qualität verbessern (sie kann sich sogar verschlechtern).

⁸ Es existiert auch eine auf **Neuronalen Netzwerken** basierende MÜ (NBMT). Allerdings ist hier noch nicht gezeigt worden, dass deren Verfahren skalierbar, d.h. für realistische Aufgaben und Anwendungen anwendbar, sind.

Hybride MÜS

Die gängigen intellektuellen Verfahren haben erfahrungsgemäß keinen Durchbruch bei der MÜ erzielen können (dies gilt aber auch für ausschließlich statistische Ansätze (s. das Scheitern des *Candidate*-Projekts bei der IBM)). Aus diesem Grund werden heutzutage bevorzugt hybride MÜS entwickelt, in denen sich die Vorteile der intellektuellen und Korpus-basierten Herangehensweisen nutzbringend ergänzen. Insbesondere ist RBMT besser geeignet für QS-Analyse, EBMT besser geeignet für Transfer und SBMT am besten geeignet für ZS-Generierung. Als Beispiel für ein hybrides MÜS s. auch das Dolmetschsystem **VERBMOBIL** (Abb. der Architektur auf Seite 44).

12.5 Evaluation

Unabhängig von der Frage, welche Anforderungen man in einem bestimmten Kontext an die MÜ stellen kann, will oder sollte (s. hierzu unten), lassen sich einige generelle Desiderata nennen, die als Evaluationskriterien für MÜS gelten können:

- **Verständlichkeit** – der übersetzte Text sollte lesbar und „natürlich“ sein
- **Genaugkeit/Originaltreue** – der Text muss bestimmte Charakteristika der QS erhalten
- **Akzeptabilität** – er muss dem beabsichtigten Zweck genügen (den Eigenschaften der jeweiligen sublanguage entsprechen)
- **Performanz** Schnelligkeit der Übersetzung
- **Kosten** – günstig
- **Zeitaufwand für Revision** – so wenig post-editing wie möglich
- **Fehlerzahl** – sollte sich in Grenzen halten
- **Cross-linguale Anwendbarkeit/Erweiterbarkeit** – muss verschiedene Sprachen in einheitlicher Weise behandeln und leicht auf neue Sprachen anwendbar sein
- **Pflege und Update der Daten** – Als Idealfall: Uniforme Analyse und Synthese, d.h. Verwendung derselben Datenstrukturen für Parsing und Generierung; einfaches Updating durch Ressourcenerweiterung ist möglich
- **Fehlertoleranz** – „graceful degradation“ statt Komplettversagen bei Fehlern
- **Kollaboration** – Gemeinsame Softwarekonstrukte und Datenstrukturen liegen dem komplexen, ggf. verteilt entwickelten MÜ-System zugrunde

12.6 Ausblick

Die bereits angesprochene Mehrgleisigkeit der gegenwärtigen MÜ bietet trotz des Status der MÜ als unausgereifter Technologie verschiedene realistische Möglichkeiten, praktikable Ergebnisse zu erzielen. Im Einzelnen existieren die folgenden praxisorientierten Richtungen:

Computer-basierte Übersetzungshilfen (MAHT)

Dies sind einerseits Systeme zum Terminologie-Management (d.h. Einbindung von Fachwortschatzen in die Übersetzung), „Translation memories“ (Systeme zum Abruf bekannter Übersetzungsmuster und Speichern neuer), und schließlich „Translation workstations“ (Systeme, die dem Übersetzer alle benötigten Tools und Ressourcen für effektives Arbeiten zur Verfügung stellt).

MÜS mit menschl. Unterstützung (HAMT)

Hierbei handelt es sich um eingeschränkte/beschränkte, aber lauffähige Systeme, deren Beschränktheit durch menschliche Vorbereitung (*pre-editing*) oder Nachbereitung (*post-editing*) kompensiert wird. Eine wesentliche Möglichkeit des pre-editing besteht in der Verwendung *kontrollierter Sprache*, d.h., der Eingabetext wird so verändert, dass er eindeutig ist (und so Übersetzungsprobleme durch Ambiguitäten vermieden werden). Die Aufgabe des post-editing ist die Korrektur von Fehlern der automatischen Übersetzung. Bei beiden Methoden wird davon ausgegangen, dass HAMT immer noch kosten- und zeitsparender ist als ausschließliche menschliche Übersetzung.

MÜS für Grob-Übersetzung

Hier wird die Minderwertigkeit der MÜ für bestimmte Aufgaben in Kauf genommen, da sie für die gewünschten Zwecke ausreicht. Dies können PC Systeme oder Online-Dienste für den Privatgebrauch oder für (semi-professionelle) Übersetzer sein, oder auch der Einsatz der MÜ zu Sichtungszwecken (bei Texten in unbekannten Sprachen) in Firmen und Behörden. Auch hier gilt, dass heutzutage mit geringen Kosten z. B. für PC-Software⁹ erhebliche Einsparungen an anderer Stelle möglich sind.

Insgesamt lässt sich eine realistische Haltung zum Einsatz von MÜS anhand folgender Standpunkte charakterisieren:

- **MÜ ist gegenwärtig nicht Übersetzung im üblichen Sinn (also nicht allgemein FAT), d.h. MÜ ersetzt generell *nicht* eine menschliche Übersetzung.** Stattdessen ist es als Computer-basiertes Werkzeug zu betrachten, und zwar im Wesentlichen für Übersetzer, für mehrsprachige Kommunikation und für einen rudimentären Zugang zu anderssprachlichen Informationsressourcen.
- **Perfektionismus ist nicht essentiell/notwendig.** Danach ist qualitativ hochwertige MÜ (HQMT) nicht generell eine praktische Maxime, sondern eher ein (bislang noch utopisches) Desiderat. Stattdessen wird publikationsreife Qualität weiterhin menschliche Editions-/ Revisionstätigkeit benötigen. Außerdem reicht für Informationssammlung (*assimilation*) und Informationsaustausch (*interchange*) sowieso oft mindere Qualität.
- **MÜ sollte nur soweit verwendet werden, wie sich Kosten/Aufwand sparen lassen.** Ansonsten bereitet MÜ selbst Kosten und Aufwand, die bei ihrem Einsatz

⁹ Ein Beispiel ist die von der Firma *Lingenio* vertriebene Software *translate*. S. hierzu [deren Homepage](#), auch zu den zugrunde liegenden computerlinguistischen Techniken.

berücksichtigt werden müssen.

- **Bewertung sollte auf realistischen Kriterien beruhen.** Die Leistung eines MÜS sollte nicht danach beurteilt werden, ob das System „reale“ Übersetzungen leistet oder ob es „gute“ Übersetzungen produziert, sondern danach, ob der Output brauchbar ist und ob der Gebrauch des Systems Zeit oder Geld spart.

12.7 Literatur

Dieser Abschnitt bezieht seine Information vor allem aus *Arnold et al.* (1995), *Dorr et al.* (1999) und *Hutchins und Somers* (1992) sowie aus den verschiedenen Papieren von Hutchins (*Hutchins* 2005). Eine sehr gut lesbare Einführung in die MÜ bieten *Jekat und Volk* (2009), *Maegaard* (1999) verkörpert eine praxisorientierte Standortbestimmung.

Kapitel 13

Textzusammenfassungssysteme (TZS)

The original Star Wars trilogy (Episodes IV, V, and VI) told the story of Luke Skywalker, a young farmboy who became a hero in the struggle to overthrow an evil empire and had to confront one of the Empire's staunchest henchmen, Darth Vader, who held the terrible secret of his father's fate. The new Star Wars trilogy will go back in time a full generation to tell the story of Anakin Skywalker, the innocent boy who will one day become the dreaded Darth Vader.

(*Star Wars in zwei Sätzen.*

Quelle:<http://www.scenebyscene.net/i/summary.html>)

Die automatische Zusammenfassung von Texten hat ein enormes Anwendungspotential, wie einige der möglichen Anwendungsszenarien zeigen (*Maybury und Mani* 2001):

- **Multimedia Nachrichtenzusammenfassungen:** Schau die Nachrichten und sag mir, was passiert ist, als ich weg war
- **Ärzte-Hilfen:** Fasse zusammen und vergleiche die empfohlenen Behandlungen für diesen Patienten
- **Meeting-Zusammenfassung:** Finde heraus, was auf der Tele-Konferenz passiert ist, die ich verpasst habe
- **Suchmaschinentreffer:** Fasse die von Suchmaschinen gelieferte Ergebnisliste zusammen
- **Informationssammlung:** Erstelle eine 500-Wort-Biografie von Osama bin Laden
- **Hand-held devices:** Erstelle eine Bildschirm-große Zusammenfassung eines Buchs
- **Elektronische Kommunikationshilfen:** Verdichte den Text und lese ihn für eine blinde Person vor

Die Entwicklung von TZS gehört damit zu den ambitioniertesten Anwendungsbereichen der Computerlinguistik. Warum das so ist, lässt sich aus den zwei folgenden Charakterisierungen der TZ ableiten:

The main goal of a summary is to present the main ideas in a document in less space.

(aus: *Radev et al.* 2002)

A summary is a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)

(aus: *Hovy* 2003)

Idealerweise erfordert die Textzusammenfassung zusätzlich zu der Fähigkeit, Texte auf allen relevanten Ebenen analysieren und produzieren zu können [!], die Bewältigung von Teilaufgaben, die mit der Auswahl und Präsentation der wesentlichen Aussagen zu tun haben, nämlich

- deren Identifikation
- die weitere Einschränkung bzgl. der Relevanz für den Nutzer
- die Vermeidung von Redundanz und Widersprüchen bei mehreren Texten
- die Sicherstellung des inhaltlichen Zusammenhangs sowie der Lesbarkeit/Wohlgeformtheit

Dies erklärt den state-of-the-art bei TZS, der dadurch charakterisiert ist, dass

- a) es keine hochwertigen, allgemein verwendbaren TZS gibt
- b) es sich immer noch überwiegend eher um ein Zusammen-setzen/stellen als um ein Zusammen-fassen handelt
- c) die Entwicklung der TZS gegenwärtig stark methodenorientiert ist und von flachen Verarbeitungsverfahren dominiert wird.

13.1 Allgemeines zu Textzusammenfassungen

Viele der kommerziell verfügbaren Systeme oder Systemkomponenten (wie Microsoft Office's **AUTOSUMMARIZE**) basieren auf einfachen statistischen Verfahren. Dies ermöglicht eine finanziell verwertbare Systemperformanz, hält aber höheren Qualitätsansprüchen nicht stand.¹ Dass trotzdem Fortschritt und Erfolge zu verzeichnen sind, ist auf die zunehmende, fruchtbare Zusammenarbeit unintelligenten IR-Methoden und cleverer computerlinguistischer Verfahren zurückzuführen.

Kriterien, anhand derer sich Textzusammenfassungen unterscheiden können, sind

¹ Ein Beispiel für ein solches einfaches System ist das in Abb. 13.1 auf der nächsten Seite dargestellte, das in Python für MacOSX entwickelt wurde. Es kombiniert einstellbare Struktur- und Relevanzfilter für automatische TZ mit der Möglichkeit zu Post-Editing (One-Klick-Indikation, welche (Teil-)Sätze der Zusammenfassung hinzugefügt bzw. von ihr entfernt werden sollen) im Sinne einer computergestützten TZ.

- **Zweck:** nur einen Eindruck vom Inhalt geben (*indikativ*), den wesentlichen Informationsgehalt wiedergeben (*informativ*), einen bewertenden Eindruck des Inhalts geben (*evaluativ/kritisch/bewertend*)
- **Quelle:** einzelnen Text oder mehrere Texte zusammenfassen (*single-document vs. multi-document*)
- **Kontext:** soll der Text unter bestimmten Gesichtspunkt/ in Bezug auf eine spezifische Frage eines Benutzers hin zusammengefasst werden (*topic-oriented*) oder ist die Zusammenfassung Anfrage-unabhängig (*generic*)
- **Form:** Extrakt vs. Abstrakt. Für ein Extrakt werden üblicherweise nur Auszüge des Originaltextes verwendet und ggf. sprachlich angepasst. Ein Abstrakt ist ein neuer Text, der auf der Grundlage der aus dem Originaltext herauskristallisierten Information generiert wird.
- **Fluss:** Die Zusammenfassung kann durch die Auslassungen wenig kohärent (fragmentarisch) oder durch Sprachgenerierung bzw. sprachliche Anpassung kohärent (zusammenhängend) sein.

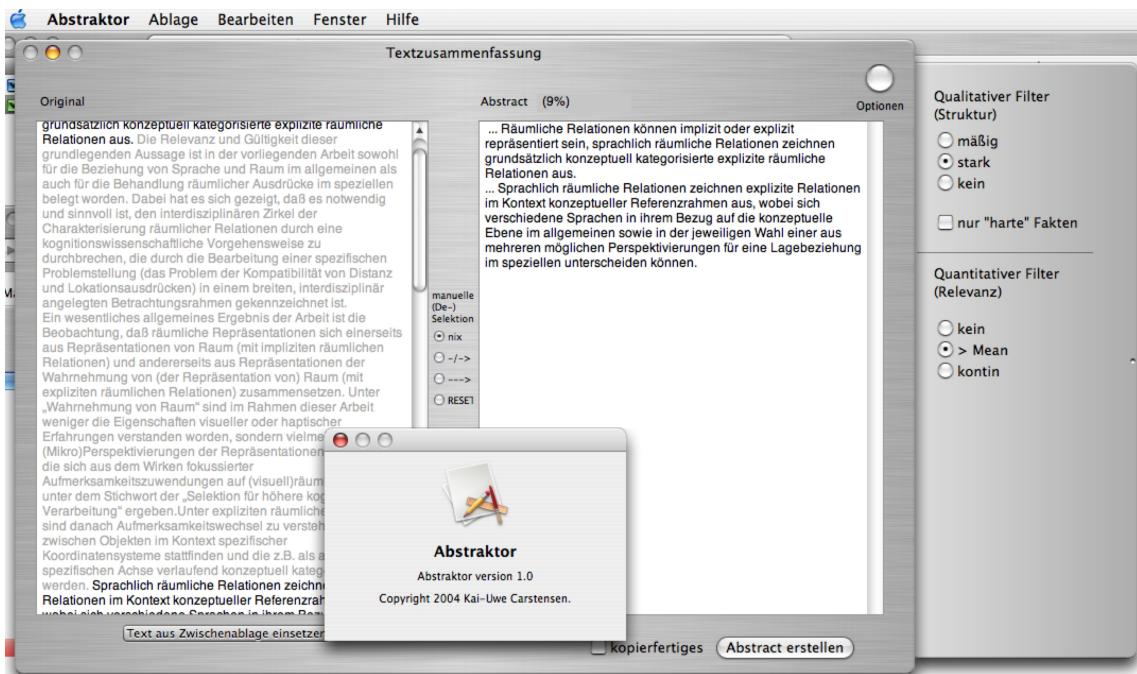


Abbildung 13.1: Screenshot eines typischen einfachen TZS-Interfaces

Während Extrakte sich durch die Wiederverwendung repräsentativer Textteile (Wörter, Phrasen, Sätze, Paragrafen) auszeichnen, stellen Abstrakte den extrahierten Inhalt nicht (vollständig) im selbem Wortlaut dar. Form-bestimmende Prozesse sind **Extraktion** (Identifizierung wichtigen Materials), **Abstraktion** (Reformulierung), **Fusion** (Kombination extrahierter Textteile) und

Klassifikation der Faktoren, die TZ beeinflussen, nach Sparck Jones^a

Inputfaktoren

Form: Beschaffenheit des Quelltextes

Gegenstandstyp: um welchen Text handelt es sich (für welche Leserschaft)?

Einheit: Einzel- vs. Multi-Dokument-Quelle

Zweckfaktoren

Situation: In welchem Kontext wird die TZ gebraucht

Leserschaft: generisch (beliebiger Leser) oder leser-spezifisch

Gebrauch: für die *Erschließung* der Quelle (soll sie gelesen werden?); als *Ersatz* (um nicht die ganze Quelle lesen zu müssen); zur *Sichtung* des Inhalts (was steht ungefähr darin?); zur Unterstützung/Verbesserung der Dokumentklassifikation

TZ-Typ: indikativ vs. informativ vs. kritisch

Bezug zur Quelle: Extrakt vs. Abstrakt

Output-Faktoren

Form: Beschaffenheit des Zieltextes

Inhalt

Form als determinierender Faktor hat folgende Aspekte:

Struktur: Unterteilung eines Dokuments (z. B. in Überschriften, Sektionen, Subsektionen)

Maßstab: die Größe des Dokuments (erfordert eine angepasste Verarbeitung)

Medium/Genre: Sprachtyp, der verarbeitet werden muss: ist es eine (sachliche) Nachricht oder ein (metaphernreicher) Roman, der verarbeitet werden muss?

Sprache: Sind Quelle bzw. Ziel ein- oder mehrsprachig, von gleicher oder verschiedener Sprache (*cross-lingual*)?

Format: in welch unterschiedliche Formate sind die Texte eingebettet (Beispiel: Webtexte; ein Artikel auf Spiegel-online ist durchsetzt von Werbungen, grafischen Elementen etc.); generell: wieviel Konversion ist notwendig, um einen Text sinnvoll verarbeiten zu können?

^a s. dazu Orăsan 2006

Kompression (unwichtiges Material auslassen). Insbesondere deswegen, weil die Form einer Textzusammenfassung generell als eine Verdichtung des Originaltextes angesehen werden kann, wird Textzusammenfassung z.T. auch als *Textkondensierung* bezeichnet.

Die **Reduktion** des Inhalts wird entweder als Kompressions-/Kondensations-/Reduktionsrate (in %) oder als Längenmaß (Wörter oder Sätze) angegeben. Dies ist Standard in den kommerziellen Systemen und Plug-Ins. Die **Informativität** einer TZ misst sich einerseits an der

Wiedergabegenaugigkeit und andererseits an der Relevanz in Bezug auf die Interessen des Benutzers.

Die eben genannten Unterschiede bei den Prozessen bedingen verschiedene Kriterien für die **Wohlgeformtheit** von TZ: Bei Extrakten ist auf die Vermeidung von Lücken, von Anaphern ohne Antezedenzen („dangling anaphors“) und von zerstörten Strukturen (Tabellen, Listen, Aufzählungen etc.) zu achten; bei Abstrakten steht die Sicherstellung zusammenhängendem, grammatischem, lesbarem Outputs im Zentrum.

Besondere Anforderungen stellen *multi-document TZ*. Hier müssen zunächst Überlappungen erkannt und entsprechend **Wiederholungen vermieden** werden. Analog dazu müssen Widersprüche erkannt und **Inkonsistenzen vermieden** werden. Da die zeitliche Darstellung der Ereignisse in den Quelltexten variieren kann, muss die **korrekte Darstellung der zeitlichen Reihenfolge** sichergestellt werden. Außerdem kann es notwendig sein, **zusätzlichen Text** (z. B. „einführende“ Sätze) zu **generieren**, um eine kohärente TZ zu produzieren.

13.2 Geschichte

Die Entwicklung von TZS ist ein Musterbeispiel für die charakteristischen „turns“ in der Computerlinguistik (zur Wissensbasiertheit in den 1970ern, zur Statistik in den 1990ern). Tabelle 13.1 gibt einen Überblick über wesentliche statistische und wissensbasierte Ansätze.

Autor(en)-Jahr	Thema	STAT	WISS
Luhn (1958)	Worthäufigkeit	x	
Edmundson (1969)	Weitere Faktoren	x	
Lehnert (1982)	Plot Units		x
Jacobs und Rau (1990)	SCISOR (Presseberichte über Unternehmensübernahmen)		x
McKeown et al. (1995)	STREAK (Kurzbeschreibung von Basketballspielen), PLANDOC (Zusammenfassungen von Planungsaktivitäten eines Telefonnetzwerks)		x
Kupiec et al. (1995)	Trainierbarer Textzusammenfasser	x	
Marcu (1997), Marcu (2000)	RST-basierte TZ		x
Hahn (1999)	TZ aus Wissensrepräsentationsstrukturen		x
-Mai 1998	Ende der Tipster Text Summarization Evaluation (SUMMAC)	x	
Hovy und Lin (1999)	SUMMARIST	x	
Ab 2001	Document Understanding Conferences (DUCs), s. hierzu mehr auf Seite 205	x	

Tabelle 13.1: Meilensteine der TZS-Entwicklung

13.3 Architektur, Ansätze und Methoden

Die generische Architektur eines Tzs besteht aus den Komponenten *Analyse*, *Transformation* und *Synthese* (auch entsprechend: *Themenidentifikation*, *Themenfusion* und *Zusammenfassungsgenerierung* nach Hovy 2005) wie in Abb. 13.2 abgebildet.

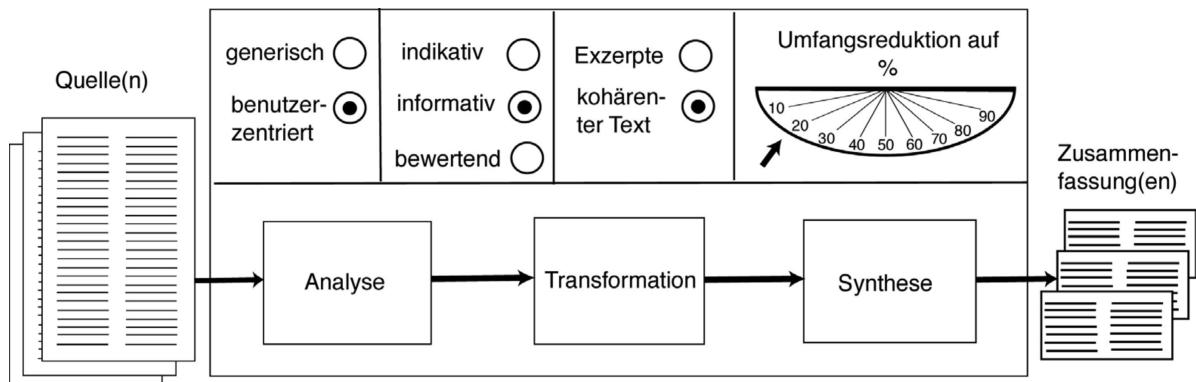


Abbildung 13.2: Architektur von Tzs (nach Endres-Niggemeyer 2004)

Wie Tabelle 13.1 auf der vorherigen Seite zeigt, werden auch im Bereich der Tzs „flache“ (Statistik-basierte) und „tiefe“ (wissensbasierte) Ansätze unterschieden, die sich wie folgt charakterisieren lassen. Das Ergebnis **flacher Ansätze** ist in der Regel ein Extrakt, die *Transformation* besteht in der Selektion salienter Einheiten und die *Synthese* stellt sich vorwiegend als Glättung der extrahierten Textteile dar.

Im Gegensatz dazu ist das Ergebnis **tiefer Ansätze** üblicherweise ein Abstrakt, die *Transformation* besteht aus der Anwendung von Operationen über Konzepte (z. B. Generalisierung) oder sprachlichen Strukturen (z. B. Diskursbäumen) und die *Synthese* entspricht nicht selten einer vollständigen Sprachgenerierungskomponente.

Um zu entscheiden, welche Sätze eines Dokuments für ein Extrakt selektiert werden sollen, verwenden **Verfahren der flachen Analyse** für die Thema-Identifikation eine Bewertungsprozedur, die auf verschiedenen Merkmalen basiert. Wesentliche allgemeine Merkmale sind

- die *Position* des Satzes (im Dokument, im Paragraph, in bestimmten Sektionen wie Titel, Einleitung oder Schluss)
- die *Häufigkeit* von Wörtern/Phrasen
- *Hinweis-Phrasen* (*cue phrases*, wie z. B. „Zusammenfassend“, „haben gezeigt, dass“) und hervorhebende Ausdrücke („insbesondere“ etc.)
- *Kontext-Termini*: Wörter/Phrasen, die auch in wichtigen Textteilen wie Überschriften etc. (oder in der Benutzeranfrage) auftreten
- *Kohäsive Merkmale* die anzeigen, d.h. wie stark ein Satz mit anderen Sätzen vernetzt ist (je größer der Grad der Vernetztheit, desto höher die Bewertung). Unterschieden wird zwischen grammatischer Kohäsion (Anaphern, Ellipsen und Konjunktionen)

und lexikalischer Kohäsion (Synonymie, Hypernymie, Wiederholung).

Spezifische Merkmale für die Bewertung sind vor allem die *Satzlänge*, das Auftreten von *Named Entities*, die *lexikalische Verknüpftheit* mit anderen Sätzen (Stichwort: **lexical chains**², s. Silber und McCoy 2002), *numerische Daten*, das Auftreten von *Pronomina*, *zeitliche Angaben* (Wochentags-, Monatsangaben) und *Anführungen* (quotations).

Tiefer gehende Verfahren analysieren die zugrundeliegende *Diskursstruktur* des Dokuments (s. hierzu den Exkurs auf der nächsten Seite), um die Vernetztheit und Relevanz von Sätzen zu bewerten. Hierzu gehören die Betrachtung des Dokumentformats, der thematischen, rhetorischen oder narrativen Struktur. Ein typisches Vorgehen in einem Diskurs-basierten Ansatz ist:

- Wahl einer Diskursstruktur-Theorie
- Parsing des Texts in einen Baum von Diskurssegmenten (die Blätter sind (Teil-) Sätze), wobei die Blätter keine assoziierte Semantik besitzen müssen
- Gewichtung der Knoten des Baums
- Auswahl der Blätter gemäß Gewichtung
- Ausgabe der selegierten Blätter für die Zusammenfassungssynthese

Die **flache Transformation** ist im Kern eine Kombination der erhobenen Einzelbewertungen. Die Selektion eines Satzes für ein Extrakt lässt sich danach direkt als lineare Merkmalskombination umsetzen, nämlich allgemein als (17) und etwas spezifischer z. B. als (18).

$$(17) \quad Wert(Satz) = \sum K_j P_j \quad (K=\text{Koeffizient}, P=\text{Parameter}, j=1\dots n)$$

$$(18) \quad Wert(Satz) = K_1 * Position(Satz) + K_2 * HinweisP(Satz) + K_3 * Häuf(Satz) + K_4 * Kontext(Satz)$$

13.4 Systeme

Abb. 13.3 auf Seite 205 zeigt die Architektur eines einfachen statistischen Systems zur Zusammenfassung einzelner Dokumente mit seiner typischen 3-Komponenten-Struktur³, Abb. 13.4 auf Seite 208 zeigt das GUI des erheblich komplexeren **SUMMARIST**-Systems (Hovy und Lin 1999).

Das **SUMMARIST**-System zeigt erste Ansätze zur Erweiterung rein Text-basierter Zusammenfassung hin zu inhaltsorientierten Verfahren (z. B. nicht nur das Zählen von Wörtern, sondern auch das Zählen der mit den Wörtern assoziierten Konzepte durch Verwendung von Ontologien)⁴.

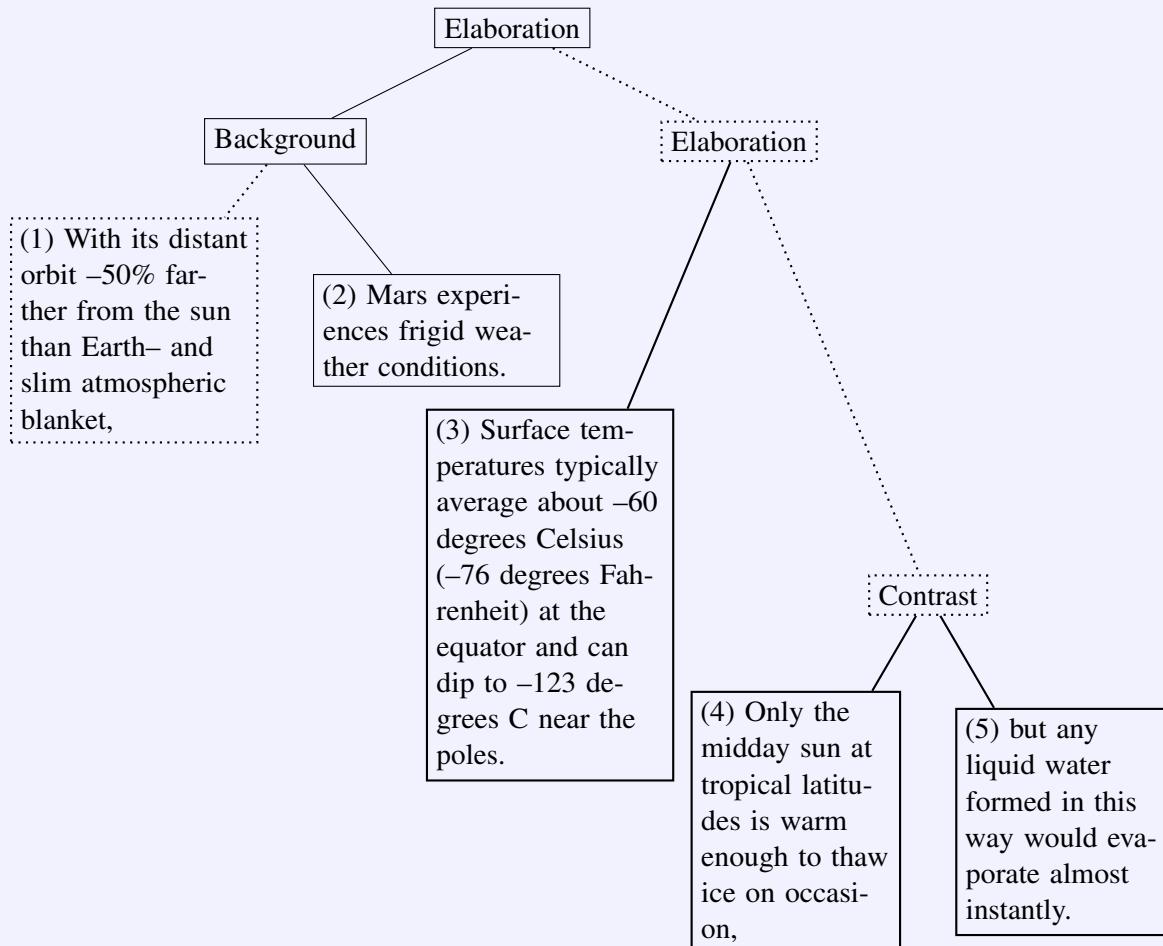
² das Aufeinanderfolgen inhaltlich aufeinander bezogener Wörter

³ online testbar auf der englischen **SWESUM**-Page

⁴ In Lin und Hovy (2002) wird ein entsprechendes Multi-Dokument-TZS vorgestellt.

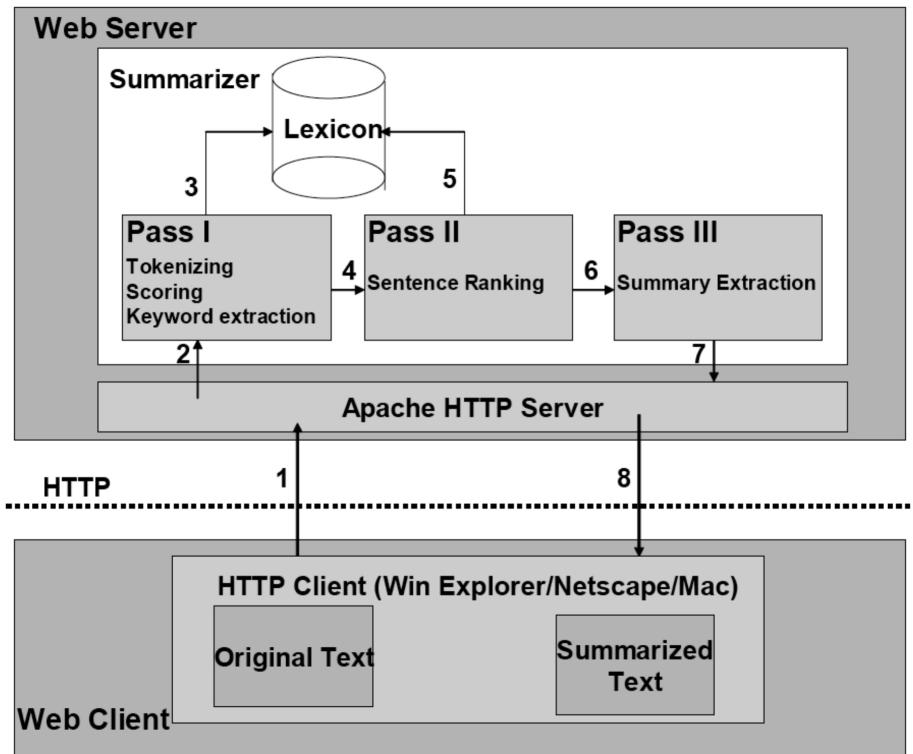
Textzusammenfassung anhand der Textstruktur

Gemäß verschiedener Theorien (s. hierzu *Marcu* 1997; *Marcu* 2000) weisen Texte eine hierarchische Struktur auf, die durch **Relationen zwischen Textabschnitten** etabliert wird (s. die folgende Grafik). Viele dieser Relationen sind asymmetrisch und markieren jeweils einen wichtigeren Teil (den „Nukleus“) und einen unwichtigeren (den „Satelliten“). Dies lässt sich für TZ ausnutzen, indem entlang der Struktur Nukleus aufgesammelt werden, bis die gewählte Länge der TZ erreicht ist (in der Grafik sind die Satelliten durch gepunktete Linien gekennzeichnet).



Für den Text, dessen Struktur hier abgebildet ist, lassen sich so je nach Anforderungen die folgenden TZ generieren:

- (2)
- (2),(3)
- (2),(3),(4),(5)


 Abbildung 13.3: Architektur eines einfachen statistischen TZS (nach *Mazdak* 2004)

13.5 Evaluation: Die Document Understanding Conferences (DUC)

Bei der Evaluation von TZ lassen sich die folgenden Typen unterscheiden: Die **Intrinsische Evaluation** misst allein die Outputqualität, z. B. im Vergleich mit einer „idealen“ TZ; die **Extrinsische Evaluation** misst die Güte der TZ in Bezug auf das Erreichen einer bestimmten Aufgabe, z. B. im Kontext eines Frage-Antwort-Szenarios. **Standardmaße** bei TZ sind deren *Länge* (als *Compression ratio CR* oder absolut, in Wörtern oder Bytes) bzw. deren *Informationsgehalt* (*Retention ratio RR*).

In den aktuellen Evaluationskonferenzen (hier: den DUCs) werden (halb)automatische Verfahren zur Evaluation eingesetzt. Hierzu gehört ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) als Verfahren zum automatischen Vergleich der handgefertigten Musterlösungen (*gold standards*) mit den von den zu evaluierenden Systemen gelieferten TZ (s. Lin 2004). SEE (*Summary Evaluation Environment*, s. Over und Yen 2004) ist eine Umgebung, die es einer evaluierenden Person (dem *assessor*) ermöglicht, die sprachliche Qualität einer TZ (*peer summary*) und die Abdeckung relativ zu dem gold standard (*model summary*) manuell zu bewerten.

Basierend auf dem in Baldwin et al. (2000) veröffentlichten Fahrplan für die Evaluation

von TZS (summarization roadmap) wurden und werden in den DUCs zunehmend komplexere Aufgaben an teilnehmende TZS gestellt, in einer Entwicklung von

- einfacher zu komplexer Textsorte
- einfachen zu anspruchsvollen Aufgaben, d.h. von
 - Extrakt zu Abstrakt
 - einzelnen zu mehreren Dokumenten als Quelle
 - Englisch zu anderen Sprachen
 - generischen zu fokussierten oder sich entwickelnden TZ
- intrinsischer zu extrinsischer Evaluation

Die folgende Auflistung gibt einen rudimentären Überblick über die jeweiligen Aufgaben der DUCs:

- **DUC-2001 auf der SIGIR im September 2001:**
Generische Zusammenfassungen von Nachrichtentexten aus einzelnen und Multi-Dokumenten; fixes Längenmaß von 50, 100, 200, und 400 Wörtern
- **DUC-2002 auf der ACL'02 im Juli 2002:**
Abstracts einzelner Dokumente sowie Dokumentmengen; fixes Längenmaß von 10, 50, 100, und 200 Wörtern. Extracts aus Dokumentmengen mit fixem Längenmaß von 200 und 400 Wörtern
- **DUC-2003 auf der HLT/NAACL im Mai 2003:**
Abstracts einzelner Dokumente sowie Dokumentmengen; fixes Längenmaß von 10 und 100 Wörtern. Multi-Dokumentzusammenfassungen von TDT⁵-Dokumentmengen im Hinblick auf vorgegebene Ereignisse, Blickwinkel und Fragen
- **DUC-2004 auf der HLT/NAACL im Mai 2004:**
Abstracts einzelner Dokumente sowie Dokumentmengen (TDT Englisch und übersetztes Arabisch). Sehr kurze (< 75 bytes) Einzeldokumentzusammenfassungen, kurze Multidokumentzusammenfassungen sowie Multidokument-Abstracts von TREC Dokumentmengen + „Who is X?“ Fragen
- **DUC-2005 auf der HLT/EMNLP 2005 im Oktober 2005:**
Generierung einer 250-Wort-TZ aus einer Dokumentmenge in Bezug auf eine Nutzeranfrage (*topic statement*)⁶
- **DUC-2006 auf der HLT/NAACL 2006 im Juni 2006:** dito
- **DUC-2007 auf der HLT/NAACL 2007 im Juni 2007:**
Zusätzliche „Aktualisierungsaufgabe“ (update task): fortlaufende 100-Wort-TZ von Dokumenten, jeweils unter Berücksichtigung der vorher bereits „gelesenen“ Dokumente

⁵ TDT=„Topic Detection and Tracking“.

⁶ man beachte hier die zunehmende Nähe zu *question answering* (Kapitel 7)

Seit 2008 existieren die DUC als *Summarization track* der **Text Analysis Conference (TAC)** (Text Analysis Conference).

13.6 Ausblick

Es existieren zwei Linien der konsequenten Erweiterung dieser Ansätze. Die eine besteht in der Kombination flacher mit tieferen Verfahren zu einem hybriden Ansatz der TZ. Hierbei wird neben der statistischen Ermittlung der Relevanz von Textteilen auch eine Auswahl anhand der Diskurssstruktur vorgenommen. Ein Beispiel ist das **PALSUMM**-System (*Thione et al.* 2004), das das FX Palo Alto's *Linguistic Discourse Analysis System* (**LIDAS**) –einen Diskursparser, der das sog. *Unified Linguistic Discourse Model* (U-LDM) implementiert– verwendet.

Der Diskursbaum, der aus der strukturellen Analyse resultiert, ermöglicht die Vergabe *struktureller* Salienzwerte für einzelne Textteile. Dazu wird mit dem frei verfügbaren statistischen **MEAD**-System eine Einschätzung der *semantischen* Salienz von Textteilen vorgenommen. Beide Werte werden kombiniert und systematisch durch den Baum weitergereicht. Eine Angabe eines Schwellwertes bewirkt so ein Beschneiden (*pruning*) des Baumes auf die für eine Textzusammenfassung relevanten Abschnitte (s. Abb. 13.5 auf Seite 209).

Die zweite Linie besteht darin, den Benutzer in den TZ-Prozess einzubinden und das Tzs als **Computer Aided Summarization** (CAS) bzw. **Computergestützte TZ** (CUTZ) zu konzipieren. Beispielsysteme sind **INEATS** (interaktive Multi-Dokument-TZ, s. *Leuski et al.* 2003) und **CAST** (Computer Aided Summarization Tool, s. *Orăsan et al.* 2003).

Die Komplexität automatischer TZ bringt es mit sich, dass alles jenseits von indikativen Extrakten bereits schwierig ist (mit entsprechenden Konsequenzen für die Qualität entsprechender Resultate) und dass kritische Abstracts oder TZ von gesprochener Sprache (Zusammenfassung einer Rede o. Ä.) auch weiterhin noch im Bereich utopischen Wunschedenkens angesiedelt sind.

13.7 Literatur

Einen guten Überblick über die Entwicklung von Tzs gibt als Reader *Mani und Maybury* (1999). *Endres-Niggemeyer* (2004) führt in die automatische Textzusammenfassung ein.

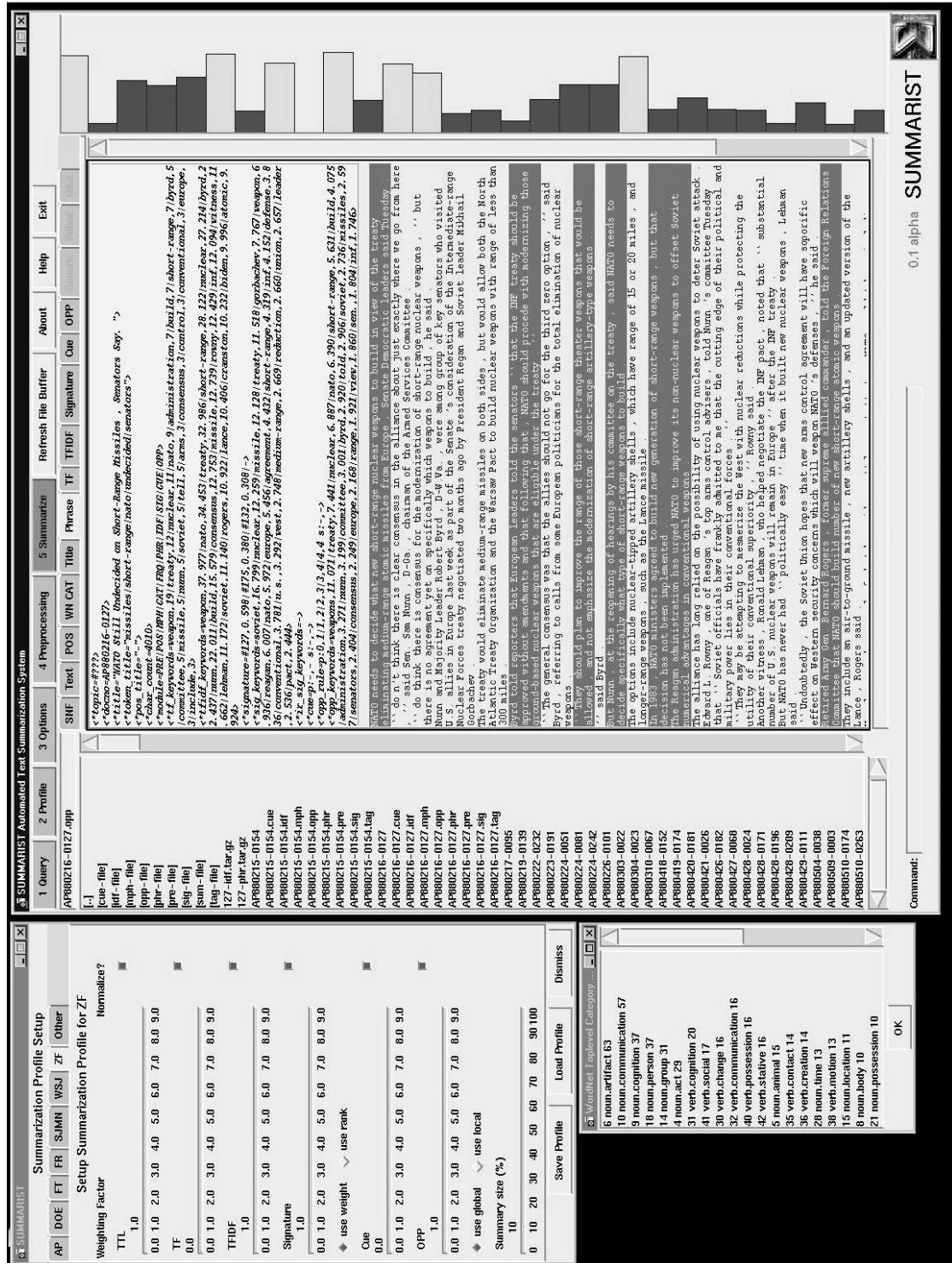


Abbildung 13.4: GUI von **SUMMARIST**, einem komplexeren statistischen Tzs

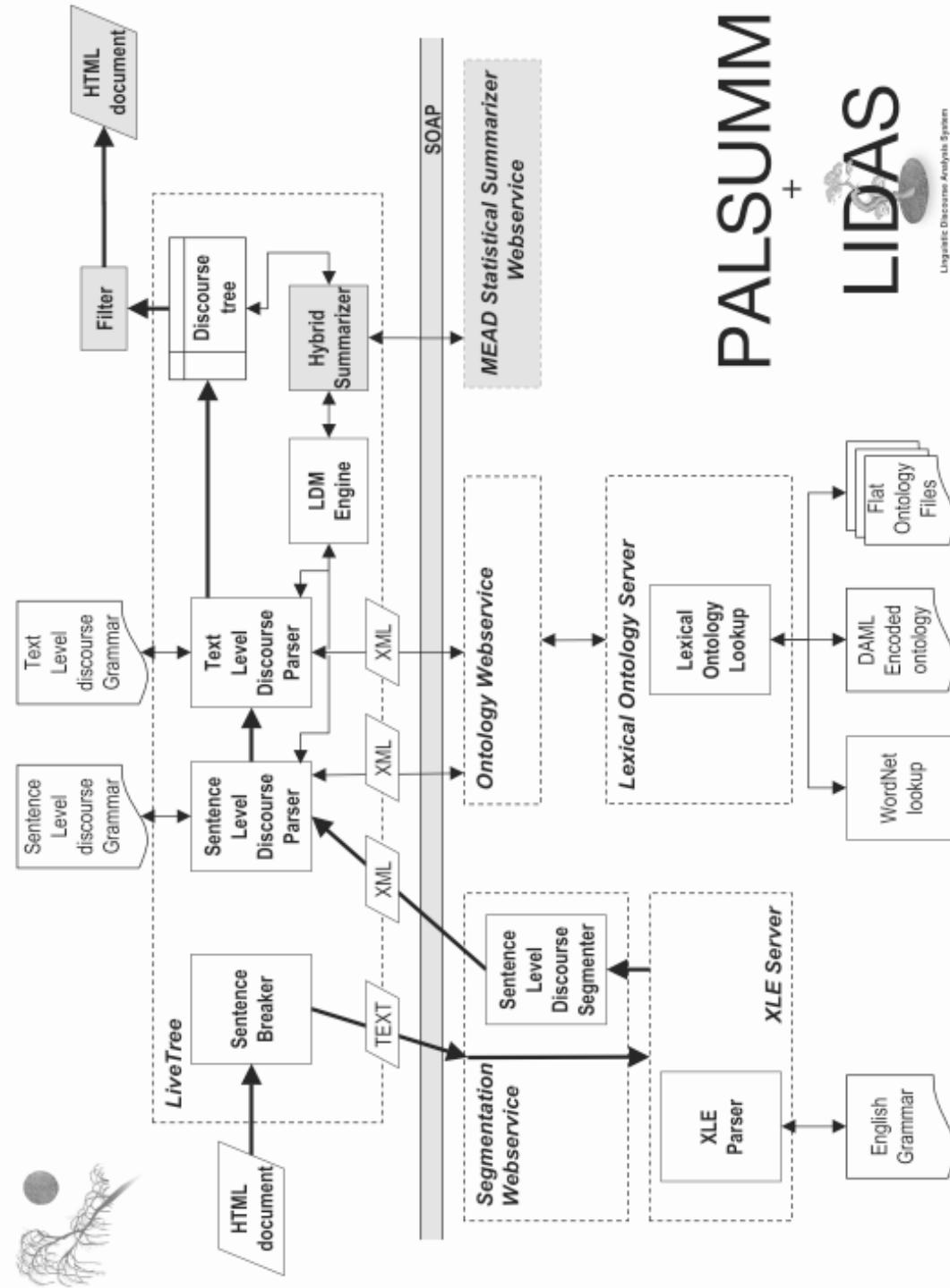


Abbildung 13.5: PALSUMM, ein hybrides TZS (nach *van den Berg et al. 2004*)

Kapitel 14

Fazit

One cannot choose but wonder. [...] for I, for my own part, cannot think that these latter days of weak experiment, fragmentary theory, and mutual discord are indeed man's culminating time!

(H.G. Wells: *The Time Machine* (Buch))

Eine kurSORISCHE LektüRE dieses Buches könnte dazu geführt haben, dass Leser zwar mehr über die Anwendungen der maschinellen Sprachverarbeitung wissen, aber doch kaum schlauer sind als zuvor: Sprachtechnologie ist zwar in der Praxis angekommen (s. **SIRI**), allerdings gilt dies nicht generell (s. Textzusammenfassung). Sie ist zwar innovativ (s. **WATSON**), jedoch sind viele Ideen schon früh vorhanden gewesen (s. z.B. **PROTOSYNTHEX** im Bereich natürlichsprachlicher Suche). Sie liefert teilweise bereits sehr gute Ergebnisse (s. **WOLFRAM|ALPHA**), aber doch nur für bestimmte, zum Teil sehr beschränkte Frage-/Aufgabenstellungen (s. Informationsextraktion), obwohl auch hier komplexere Lösungen schon vor längerer Zeit präsentiert wurden (s. **CHAT-80**).

Am enttäuschendsten ist möglicherweise die Tatsache, dass Sprachtechnologie der heutigen Zeit im Kern **ELIZA**-style und nicht **SHRDLU**-style ist, im besten Fall basiert sie auf wohlbekannten statistischen Verfahren, oft aber im Wesentlichen auf Hand-kodierten Mustern im Stile der Chatterbots.

Ist der Branche also ein Vorwurf zu machen? Die grobe Antwort hierauf ist: nein, denn Sprachverarbeitung ist so komplex, dass die Fortschritte in den letzten Jahren eher erstaunen. Ohne den ständigen Zuwachs an Rechenpower und den Fortschritten in der Softwaretechnologie wäre der gegenwärtige Stand nicht erreicht worden.

Die differenzierte Antwort hierauf ist: ja, und zwar den Theoretikern der Linguistik und Computerlinguistik, der Mathematik und Logik. Hier waltet oft ein Orientieren an aktuellen Moden und Paradigmen, wo eine Ziel-orientierte konzentrierte Problemlösung erforderlich wäre. Oder ein Festhalten an alten Modellen, wo die Notwendigkeit ihrer Modifikation auf der Hand liegt. Für den Bereich der Semantik habe ich es einmal wie folgt formuliert (s. Carstensen 2003):

Hierzu sollte die CL zwecks Vermeidung eines Dornröschenschlafs nicht einseitig auf die Ergebnisse der formalen/logischen Semantik bauen, sondern deren Resultate dankend zur Kenntnis nehmen und unter Berücksichtigung der Erkenntnisse weiterer Disziplinen (Informatik, Linguistik, KI, Kognitionswissenschaft) moderne, einsetzbare Antworten produzieren.

Einer entsprechenden Verallgemeinerung dieses Vorschlags habe ich nichts hinzuzufügen.

In jedem Fall ist es momentan völlig offen, ob wir (bereits) in 20 Jahren weitgehend performante natürlichsprachliche Systeme haben werden, oder ob selbst in 100 Jahren Anaphernauflösung immer noch nicht zufriedenstellend gelöst sein wird. Etwas Anderes zu behaupten würde derselben Kategorie zugeordnet werden müssen wie die folgenden falschen Vorhersagen:

Man will not fly for 50 years

[Wilbur Wright, 1901, 2 Jahre vor dem ersten erfolgreichen Flug ([Quelle](#))]

Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved.

[Marvin Minsky, 1967, vor mehr als einer Generation ([Quelle](#))]

Unter praktischen Gesichtspunkten ist es wohl sowieso wichtiger, dass z.B. (ältere) Leute in Gefahrensituationen bald weit verbreitet mit ihrem Handy durch einfachen Knopfdruck und intuitive Spracheingabe ("Hilfe", "Es brennt" etc.) einen entsprechenden Notruf aktivieren können. Für viele andere Bereiche ist es tolerabel und —insbesondere aus ökonomischen Gründen— vielleicht sogar begrüßenswert, dass das Problem hochqualitativer automatischer Sprachverarbeitung auf absehbare Zeit noch nicht gelöst sein wird.

Kapitel 15

Epilog

Nathan: *One day the AIs are going to look back on us the same way we look at fossil skeletons on the plains of Africa. An upright ape living in dust with crude language and tools, all set for extinction.*

(*Ex Machina (Film)*)

Im Jahr 2017 muss man fast schmunzeln, wenn man das Absetzen eines sprachlichen Notrufs als besondere Errungenschaft der Sprachtechnologie herausstellt, wie ich das im Fazit vor fünf Jahren getan habe. All die **SIRIS**, **CORTANAS**, **ALEXAs** und sonstige digitale Assistenten sind eher schon gar nicht weit davon entfernt, auf Zuruf Urlaube auf Bali zu buchen oder die netzfähige Brotbackmaschine zu programmieren.

WATSON findet schon jetzt Anwendung in den verschiedensten Bereichen (neben der Medizin) und wird m.A. nach noch große Erfolge feiern. Der Hype um autonomes Fahren (selbstverständlich ist in Autos schon jetzt sprachliche Interaktion möglich) ist in vollem Gange.

Deep learning als aktuelles Buzzword der KI bezeichnet Verfahren, die alle Probleme der Wissens- und Sprachverarbeitung zu lösen versprechen. Die potentiellen Gefahren der KI scheinen allerdings momentan schon so sehr angewachsen zu sein, dass prominente Wissenschaftler vor der Machtergreifung künstlicher über die natürliche Intelligenz warnen.

Mark Zuckerberg und seine Mitarbeiter verkünden in diesen Tagen, an nicht-invasiven Verfahren zum Gedankenlesen zu arbeiten. Und Ray Kurzweil verspricht mal wieder einiges, unter anderem dass wir unseren Geist in ein, zwei Jahrzehnten (oder so) in den digitalen Himmel werden hochladen können.

Es hilft, an dieser Stelle mal kurz Luft zu holen und die Brille gerade zu rücken. Als erstes: ja, es gab und gibt Fortschritt, der zum Teil beeindruckt. Allerdings ist es wichtig, immer die Größenordnungen und Fakten im Auge zu behalten. Mit den jetzigen Verfahren werden **SIRI** und Co. niemals intelligent werden, sondern immer programmierte täuschend echt agierende Idioten bleiben.

Deep learning ist nicht viel mehr als ein anderer Ausdruck für etwas, das man schon seit langem kennt (*Konnektionismus* oder *Künstliche neuronale Netze*). Selbstverständlich hat die Zusammenarbeit der Informatik mit der Kognitions- oder Neurowissenschaft hier

große Fortschritte erzielt. Diese sind aber *relativ* zum Stand zuvor. Ich sehe beispielsweise nicht, dass das einfache Stapeln neuronaler Schichten – das Merkmal des *deep learning* – direkt zum Erreichen höherer kognitiver Fähigkeiten führt (was Fachleute auch zugeben). Von Gedankenlesen und -hochladen ganz zu schweigen. In diesem Bereich scheint es sich um einen Hype zu handeln (der übrigens vor 25 und 50 Jahren oder so auch schon aufkam), in dem es in erster Linie um das Beeindrucken und Zutrauen von Investoren und Aktionären geht.

Und was die Gefahren angeht: Hier ist es nicht die Intelligenz von Maschinen, die wir in nächster Zeit fürchten müssen, es ist vielmehr der Unsinn, der durch fehlende Intelligenz entstehen kann, der uns Angst machen sollte. Ein Beispiel ist das autonome Fahren. Selbstredend kann man entsprechende Autos bauen, deren neuronale Netze anhand von Millionen gefahrener Kilometer trainiert worden sind. Trotzdem gibt es Zillionen unvorhergesehene Situationen und perzeptuelle Gegebenheiten, die ein solches System zu einer völlig falschen Handlung veranlassen können. Weil die Verfahren eben von einfacher, nicht von über Jahrmillionen evolutionär gereifter komplexer Ordnung sind. Es fehlt die höhere Intelligenz, die z.B. verhindert, dass sich ein am Herz operierender Chirurg mit einer ruckartigen Bewegung am Kopf kratzt (auch wenn es tatsächlich jucken sollte). Die ruckartige Bewegung eines Autos kann sowohl in der Innenstadt als auch auf der Autobahn ähnlich fatal sein. Und warum zum Teufel bedenkt keiner in dieser Diskussion, dass wir ja auch keine achtjährigen Kinder als Autofahrer auf die Straße lassen? Intelligent genug sind sie, relativ betrachtet (wir könnten sie ja zunächst einfach ein bisschen trainierend fahren lassen).

Eine andere Art von Unsinn kann durch sich selbst modifizierende Programme entstehen ([V14](#) repräsentiert die entsprechende Nachfolgergeneration digitaler Assistenten). Das kann zu nichts anderem als **intelligenten Viren** führen, die Schaden von entsprechender Komplexität und ungeahntem Ausmaß anrichten.

Lassen wir uns also überraschen, wieviel Sinn- bzw. Unsinnhaftes in der nächsten Zeit automatisch produziert wird.

Literaturverzeichnis

- Acomb, K., Bloom, J., Dayanidhi, K., Hunter, P., Krogh, P., Levin, E. und Pieraccini, R. (2007): „Technical Support Dialog Systems: Issues, Problems, and Solutions“. In: *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, S. 25–31. Rochester, NY: Association for Computational Linguistics.
- Alexandersson, J., Richter, K. und Becker, S. (2006): „i2home: Benutzerzentrierte Entwicklung einer offenen standardisierten Smart Home Plattform“. In: *Proceedings of USEWARE 2006 - Nutzergerechte Gestaltung technischer Systeme*, Nr. 1946. VDI-Verlag.
- Allen, J. F. (1995): „The TRAINS Project: A case study in building a conversational planning agent“. In: *Journal of Experimental and Theoretical AI (JETAI)*, 7, 7–48.
- Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L. und Stent, A. (2001): „Toward conversational human-computer interaction“. In: *AI Magazine*, 22(4), 27–37.
- Allgayer, J., Harbusch, K., Kobsa, A., Reddig, C., Reithinger, N. und Schmauks, D. (1989): „XTRA: A Natural-Language Access System to Expert Systems“. In: *Intl. J. of Man-Machine Studies*, 31, 161–195.
- André, E. (2000): „The Generation of Multimedia Presentations“. In: Dale, R., Moisl, H. und Somers, H., Hrsg., *Handbook of natural language processing*, S. 305–327. Marcel Dekker Inc.
- André, E. (2003): „Natural Language in Multimedia/Multimodal Systems“. In: Mitkov, R., Hrsg., *Handbook of Computational Linguistics*, S. 650–669,. Oxford University Press.
- André, E., Müller, J. und Rist, T. (1996): „The PPP Persona: A Multipurpose Animated Presentation Agent“. In: *Advanced Visual Interfaces*, S. 245–247. ACM Press.
- Androutsopoulos, I. und Ritchie, G. (2000): „Database Interfaces“. In: Dale, R., Moisl, H. und Somers, H., Hrsg., *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, S. 209–240. Marcel Dekker Inc.
- Androutsopoulos, I., Ritchie, G. und Thanisch, P. (1995): „Natural Language Interfaces to Databases – an Introduction“. In: *Journal of Natural Language Engineering*, 1(1), 29–81. Also: Research Paper no 709, Dept. of AI, University of Edinburgh, and cmp-lg/9503016, March 1995.
- Androutsopoulos, I., Ritchie, G. und Thanisch, P. (1998): „Time, Tense and Aspect in Natural Language Database Interfaces“. In: *Natural Language Engineering*, 4(3), 229–276.
- Appelt, D. E. (1985): *Planning English Sentences*. Cambridge: Cambridge University Press.
- Appelt, D. E. und Israel, D. J. (1999): „Introduction to Information Extraction Technology“. A tutorial prepared for IJCAI-99, Stockholm, Schweden.

- Arnold, D., Balkan, L., Meijer, S., Humphreys, R. und Sadler, L. (1995): „MACHINE TRANSLATION: An Introductory Guide“. <http://www.essex.ac.uk/linguistics/clmt/MTbook/HTML/book.html>.
- Baeza-Yates, R. A. und Ribeiro-Neto, B. A. (1999): *Modern Information Retrieval*. ACM Press / Addison-Wesley.
- Baldwin, N., Donaway, R., Hovy, E., Liddy, E., Mani, I., Marcu, D., McKeown, K., Mittal, V., Moens, M., Radev, D., Spärck Jones, K., Sundheim, B., Teufel, S., Weischedel, R. und White, M. (2000): „An Evaluation Roadmap for Summarization Research“.
- Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M. und Etzioni, O. (2007): „Open Information Extraction from the Web“. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Bar-Hillel, Y. (1960): „A Demonstration of the Non-Feasibility of Fully Automatic High Quality Translation“. reprinted in Yehoshua Bar-Hillel (1964), *Language and Information: Selected Essays on their Theory and Application* (Addison-Wesley Publ. Com. Inc.), 174–179.
- Bateman, J. und Zock, M. (2003): „Natural Language Generation“. In: Mitkov, R., Hrsg. , *The Oxford Handbook of Computational Linguistics*, S. 284–304. Oxford: Oxford University Press.
- Bateman, J. A. (1997): „Enabling technology for multilingual natural language generation: the KPML development environment“. In: *Natural Language Engineering*, 3(1), 15–55.
- Bateman, J. A. (2009): „Angewandte natürlichsprachliche Generierungs- und Auskunftsysteme“. In: Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, S. 633–641. Heidelberg: Spektrum Akademischer Verlag.
- Belz, A. und Vargas, S. (2007): „Generation of Repeated References to Discourse Entities“. In: *Proceedings of the 11th European Workshop on Natural Language Generation, Schloss Dagstuhl, Germany*.
- Berry-Rogghe, G. L. und Wulz, H. (1978): „An Overview of PLIDIS, a Problem Solving Information System with German as Query Language.“. In: *Natural Language Communication with Computers*, S. 87–132.
- Biveroni, C. (2002): „Natürlichsprachliche Datenbankschnittstellen: Sprachverstehen und Benutzerinteraktion.“. Diplomarbeit, http://www.cl.uzh.ch/fileadmin/user_upload/docs/studium/liz/lizarbeiten/lizbiveroni.pdf, Lehrstuhl Computerlinguistik der Universität Zürich.
- Bobrow, D., Kaplan, R., Kay, M. und et al. (1977): „GUS: A Frame-Driven Dialog System“. In: *Artificial Intelligence*, 8, 155–173.
- Boitet, C. (1997): „GETA’s MT methodology and its current development towards personal networking communication and speech translation in the context of the UNL and C-STAR projects“.
- Bolt, R. (1980): „Voice and gesture at the graphics interface“. In: *ACM Computer Graphics*, 14(3), 262–270.
- Bontcheva, K., Tablan, V., Maynard, D. und Cunningham, H. (2004): „Evolving GATE to Meet New Challenges in Language Engineering“. In: *Natural Language Engineering*, 10(3/4), 349—373.

- Bos, J., Carofiglia, V., de Carolis, N., Clark, R., Laaksolahti, J., Matheson, C., Oka, T., Rist, T., de Rosis, F., Pelachaud, C., Schmitt, M., Steedman, M. und Zinn, C. (2002): „Language and Speech Architecture for Prototype 1“. Arbeitspapier, MAGICSTER - Embodied Believable Agents.
- Boves, L. (2004): „Robust Conversational System Design“. In: *Proceedings of COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*. Norwich.
- Bunt, H., Kipp, M., Maybury, M. und Wahlster, W. (2005): „Fusion and Coordination for Multimodal Interactive Information Presentation: Roadmap, Architecture, Tools, Semantics“. In: Stock, O. und Zancanaro, M., Hrsg. , *Multimodal Intelligent Information Presentation*, Bd. 27, *Text, Speech and Language Technology*, S. 325–340. Kluwer Academic.
- Burger, J., Cardie, C. und et al. (2002): „Issues, Tasks and Program Structures to Roadmap Research in Question and Answering (QA)“. http://www.inf.ed.ac.uk/teaching/courses/tts/papers/qa_roadmap.pdf.
- Busemann, S. (2005): „Ten Years After: An Update on TG/2 (and Friends)“. In: Wilcock, G., Jokinen, K., Mellish, C., und Reiter, E., Hrsg. , *Proceedings of the Tenth European Natural Language Generation Workshop (ENLG 2005)*, S. 32–39.
- Cahill, L., Carroll, J., Evans, R., Paiva, D., Power, R., Scott, D. und van Deemter, K. (2001): „From RAGS to RICHES: exploiting the potential of a flexible generation architecture“. In: *Proceedings of ACL/EACL 2001*, S. 98–105. Toulouse, France.
- Carbonell, J. G. (1984): „Is There Natural Language after Data Bases?“. In: *COLING84*, S. 186–187.
- Carletta, J. C., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G. und Anderson, A. (1997): „The Reliability of a Dialogue Structure Coding Scheme“. In: *Computational Linguistics*, **23**(1), 13–31.
- Carstensen, K.-U. (1988): „WebS - ein System zur Generierung von Wegbeschreibungen“. LILOG-Memo, Nr. 11, IBM.
- Carstensen, K.-U. (1991): „Aspekte der Generierung von Wegbeschreibungen“. Report, Nr.190, IBM Germany Science Center, Institute for Knowledge Based Systems (IWBS), <http://www.kai-uwe-carstensen.de/AspektedGvWb.pdf>.
- Carstensen, K.-U. (2003): „Rezension von Willéé, Gerd/Schröder, Bernhard/Schmitz, Hans-Christian (eds.) (2002), Computerlinguistik. Was geht, was kommt? Sankt Augustin (= Sprachwissenschaft, Computerlinguistik und Neue Medien 4)“. In: Carstensen, K.-U., Hrsg. , *Learning and teaching (in) Computational Linguistics*. (= *Linguistik-online17*), S. 155–161. http://www.linguistik-online.de/17_03/rezension.html Auflage.
- Carstensen, K.-U. (2009): „Nicht-sprachliches Wissen“. In: Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, chapter 4. Ressourcen (http://www.kai-uwe-carstensen.de/Publikationen/Nicht-sprachliches_Wissen3.pdf). 3. Auflage, Spektrum Akademischer Verlag.
- Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. (2009): *Computerlinguistik und Sprachtechnologie – Eine Einführung*. 3. überarbeitete und erweiterte Auflage, Spektrum Akademischer Verlag.

- Cassell, J. (2001): „Embodied Conversational Agents: Representation and Intelligence in User Interface“. In: *AI Magazine*, **22**(3), 67–83.
- Churcher, G. (1997): „Dialogue management systems: a survey and overview“. Report, Nr.97.6, School of Computer Studies, University of Leeds.
- Codd, E. (1974): „Seven steps to rendezvous with the casual user“. In: *IFIP Working Conference Data Base Management*, S. 179–200.
- Cohen, M. H., Giangola, J. P. und Balogh, J. (2004): *Voice user interface design*. New York: Addison-Wesley.
- Cohen, P. (1997): „Dialogue Modeling“. In: Cole, R., Hrsg. , *Survey of the State of the Art in Human Language Technology*, S. 192–197. Cambridge University Press.
- Cole, R., Mariani, J., Uszkoreit, H., Varile, G., Zaenen, A., Zue, V. und Zampolli, A. (1997): *Survey of the State of the Art in Human Language Technology*. Cambridge University Press and Giardini.
- Copestake, A. und Spärck Jones, K. (1990): „Natural Language Interfaces to Databases“. In: *The Knowledge Engineering Review* 5 (4), S. 225–249.
- Cowie, J. (1983): „Automatic Analysis of Descriptive Texts“. In: *ACL Proceedings, Conference on Applied Natural Language Processing*. ACL, Santa Monica, Calif.
- Cowie, J. und Wilks, Y. (2000): „Information Extraction“. In: *Handbook of Natural Language Processing*, S. 241–260. R. Dale and H. Moisl and H. Somers.
- Crocker, M. W. (2002): „Computational Linguistics: Cognitive Science or Language Engineering?“. In: Willée, G., Schröder, B. und Schmitz, H.-C., Hrsg. , *Computerlinguistik—Was geht, was kommt?*, S. 48–51. Gardez! Verlag.
- Cunningham, H. (2005a): „Computational Language Systems, Architectures“. In: *Encyclopedia of Language and Linguistics, 2nd Edition*.
- Cunningham, H. (2005b): „Information Extraction, Automatic“. In: *Encyclopedia of Language and Linguistics, 2nd Edition*.
- Cunningham, H., Maynard, D., Bontcheva, K. und Tablan, V. (2002): „GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications.“. In: *40th Anniversary Meeting of the Association for Computational Linguistics(ACL'02)*. University of Pennsylvania, Philadelphia.
- Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., Dowman, M., Aswani, N. und Roberts, I. (2005): *Developing Language Processing Components with GATE*: The University of Sheffield.
- Dale, R., Aliod, D. M. und Schwitter, R. (2003): „Natural Language Processing in the Undergraduate Curriculum“. In: *Proceedings of the Fifth Australasian Computing Education Conference (ACE2003)*.
- Dale, R., Moisl, H. und Somers, H., Hrsg. (2000): *Handbook of natural language processing*. 1. Auflage, Marcel Dekker Inc.
- Dale, R., Oberlander, J., Milosavljevic, M. und Knott, A. (1998): „Integrating natural language generation and hypertext to produce dynamic documents“. In: *Interacting with Computers*, **11**(2), 109–135.

- DaSilva, G. und Dwiggins, D. (1980): „Towards a Prolog text grammar“. In: *SIGART*, **72**.
- Davey, A. (1974): *The Formalisation of Discourse Production*. Dissertation, University of Edinburgh.
- de Rosis, F., Grasso, F. und Berry, D. C. (1999): „Refining Instructional Text Generation after Evaluation“. In: *Artificial Intelligence in Medicine*, **17**(1), 1–36.
- DeJong, G. (1982): „An Overview of the FRUMP System“. In: Lehnert, W. und Ringle, M., Hrsg. , *Strategies for Natural Language Processing*. Lawrence Erlbaum Associates.
- Dennett, D. (1984): „Cognitive Wheels: The Frame Problem in AI“. In: Hookway, C., Hrsg. , *Minds, Machines, and Evolution*, S. 128–151. Cambridge University Press.
- Dorr, B. J., Jordan, P. W. und Benoit, J. W. (1999): „A Survey of Current Paradigms in Machine Translation“. In: Zelkowitz, M., Hrsg. , *Advances in Computers*, Bd. 49, S. 1–68. London: Academic Press.
- Draxler, C. (2009): „Sprachdatenbanken“. In: Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, chapter 4. Ressourcen. 3. Auflage, Spektrum Akademischer Verlag.
- Edmundson, H. P. (1969): „New methods in automatic extraction“. In: *Journal of the ACM*, **16**(2), 264–285. wiederaabgedruckt in: *Mani und Maybury* 1999, S. 23–42.
- Elhadad, M. und Robin, J. (1996): „An Overview of SURGE: A reusable comprehensive syntactic realization component“. Technical Report, Nr.96-03, Dept of Mathematics and Computer Science, Ben Gurion University, Beer Sheva, Israel.
- Endres-Niggemeyer, B. (2004): „Automatisches Textzusammenfassen“. In: Lobin, H. und Lemnitzer, L., Hrsg. , *Texttechnologie*. Tübingen: Stauffenburg.
- Erman, L. D., Hayes-Roth, F., Lesser, V. R., und Reddy, D. R. (1980): „The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty“. In: *Computing Surveys*, **12**(2), 213–253.
- Etzioni, O., Banko, M. und Cafarella, M. J. (2007): „Machine Reading“. In: *Proceedings of the 2007 AAAI Spring Symposium on Machine Reading*.
- Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. und Yates, A. (2005): „Unsupervised named-entity extraction from the Web: An experimental study“. In: *Artificial Intelligence*, **165**(1), 91–134.
- Feiner, S. K. und McKeown, K. R. (1991): „Automating the Generation of Coordinated Multimedia Explanations“. In: *Computer*, **24**(10), 33–41.
- Ferguson, G., Allen, J. F. und Miller, B. (1996): „TRAINS-95: Towards a Mixed-Initiative Planning Assistant“. In: *AIPS*, S. 70–77.
- Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N. und Welty, C. (2010): „Building Watson: An Overview of the DeepQA Project“. In: *AI Magazine*, **31**(3), 59–79.
- Ferrucci, D. e. a. (2009): „Towards the Open Advancement of Question Answering Systems“. Research Report, Nr.RC24789 (W0904-093), IBM Research Division.

- Flycht-Eriksson, A. und Jönsson, A. (2000): „Dialogue and Domain Knowledge Management in Dialogue Systems“. In: Dybkjaer, L., Hasida, K. und Traum, D., Hrsg. , *Proceedings of the First SIGdial Workshop on Discourse and Dialogue*, S. 121–130. Somerset, New Jersey: Association for Computational Linguistics.
- Foster, M. E. (2002): „Deliverable 6.1:State of the art review: Multimodal fission“. Arbeitspapier.
- Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jörg, B. und Schäfer, U. (2005): „Querying Structured Knowledge Sources“. In: *Proceedings of AAAI-05. Workshop on Question Answering in Restricted Domains*, S. 10–19. Pittsburgh, Pennsylvania.
- Freedman, R. (2000): „Using a Reactive Planner as the Basis for a Dialogue Agent“. In: *Proceedings of the Thirteenth Florida Artificial Intelligence Research Symposium (FLAIRS 2000)*. Orlando.
- Gaizauskas, R. und Wilks, Y. (1998): „Information Extraction: Beyond Document Retrieval“. In: *Journal of Documentation*, **54**(1), 70–105.
- Gatt, A. und Reiter, E. (2009): „SimpleNLG: A realisation engine for practical applications“. In: *Proceedings of ENLG-2009*, S. 90–93.
- Goldberg, E., Kittredge, R. und Driedger, N. (1994): „FoG: A New Approach to the Synthesis of Weather Forecast Text“. In: *IEEE Expert (Special Track on NLP)*, **9**(2), 45–53.
- Goldman, N. (1975): „Conceptual generation“. In: Schank, R. C., Hrsg. , *Conceptual Information Processing*, S. 289–372. North-Holland Publishing Co, Amsterdam.
- Graesser, A., Person, N. und Huber, J. (1992): „Computational Intelligence“. In: Lauer, T., Peacock, E. und Graesser, A., Hrsg. , *Questions and information systems*, chapter Mechanisms that generate questions, S. 67–81. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Green, B. F. J., Wolf, A. K., Chomsky, C. und Laughery, K. (1963): „Baseball: an automatic question answerer“. In: Feigenbaum, E. und Feldman, J., Hrsg. , *Computers and Thought*, S. 207–216. McGraw-Hill, New York.
- Grishman, R., Sterling, J. und Macleod, C. (1991): „Description Of The Proteus System As Used For MUC-3“. In: *Proceedings of the Third Message Understanding Conference*, S. 183–190. San Diego, CA: Morgan Kaufmann.
- Grishman, R. und Sundheim, B. (1996): „Message Understanding Conference - 6: A Brief History“. In: *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Bd. I, S. 466–471. Association for Computational Linguistics.
- Grosz, B. J. (1983): „TEAM, a transportable natural language interface system“. In: *ACL Proceedings, Conference on Applied Natural Language Processing*, S. 39–45.
- Gurevych, I., Porzel, R., Slinko, E., N., P., Alexandersson, J. und Merten, S. (2003): „Less is More: Using a Single Knowledge Representation in Dialogue Systems“. In: *In Proceedings of the HLT-NAACL Workshop on Text Meaning*. Edmonton, Canada.
- Hahn, U. (1996): „System, natürlichsprachliches“. In: Strube, G., Becker, B., Freksa, C., Hahn, U., Opwis, K. und Palm, G., Hrsg. , *Wörterbuch der Kognitionswissenschaft*, S. 715f. Stuttgart: Klett-Cotta.

- Hahn, Udo; Reimer, U. (1999): „Knowledge-based text summarization: Salience and generalization operators for knowledge base abstraction“. In: Mani, I. und Maybury, M., Hrsg. , *Advances in Automatic Text Summarization*, S. 215–232. Cambridge, MA: MIT Press.
- Hamerich, S. W., König, L. und Hennecke, M. E. (2005): „Sprachdialogsysteme im Kfz“. In: *KI*, **3**, 29–58.
- Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V. und Morarescu, P. (2000a): „FALCON: Boosting Knowledge for Answer Engines“. In: *Proceedings of the Text Retrieval Conference (TREC-9)*.
- Harabagiu, S., Pasca, M. und Maiorano, S. (2000b): „Experiments with Open-Domain Textual Question Answering“. In: *Proceedings of COLING-2000*, S. 292–298. Saarbrücken, Germany.
- Harris, L. R. (1977): „Robot: A high performance natural language data base query system“. In: *Proc. of the 5th IJCAI*, S. 903–904. Cambridge, MA.
- Harris, L. R. (1984): „Experience with INTELLECT: Artificial Intelligence Technology Transfer“. In: *The AI Magazine*, **2**(2), 43–50.
- Hausser, R. (2002): „Comparing Music and Language: data Structures, Algorithms, and Input-Output Conditions“. In: Willée, G., Schröder, B. und Schmitz, H.-C., Hrsg. , *Computerlinguistik—Was geht, was kommt?*, S. 120–127. Gardez! Verlag.
- Hearst, M. (1999): „Untangling Text Data Mining“. In: *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, S. 3–10.
- Helbig, H. (2008): *Wissensverarbeitung und die Semantik der Natürlichen Sprache*. 2. überarbeitete Auflage, Springer Verlag.
- Hendrix, G. (1977): „The LIFER manual: A guide to building practical natural language interfaces“. Technical Note, Nr.138, SRI International, Menlo Park, Ca.
- Hendrix, G., Sacerdoti, E., Sagalowicz, D. und Slocum, J. (1978): „Developing a natural language interface to complex data“. In: *ACM Transactions on Database Systems*, **3**(2), 105–147.
- Henrici, A. (1965): „Notes on the systemic generation of a paradigm of the English clause“. Technical report, O.S.T.I. Programme in the Linguistic Properties of Scientific English. (Reprinted in M.A.K. Halliday and J.R. Martin (Eds., 1981) *Readings in Systemic Linguistics*, London: Batsford.).
- Herzog, G., Ndiaye, A., Anmerten, S., Kirchmann, H., Becker, T. und Poller, P. (2004): „Large-scale software integration for spoken language and multimodal dialog systems“. In: *Natural Language Engineering*, **10**(3/4), 283–305.
- Herzog, O. und Rollinger, C.-R., Hrsg. (1991): *Text Understanding in LILOG, Integrating Computational Linguistics and Artificial Intelligence, Final Report on the IBM Germany LILOG-Project*, Bd.546, *Lecture Notes in Computer Science*. Springer.
- Heyer, G. (2002): „Computerlinguistik und die Entwicklung von Sprachprodukten“. In: Willée, G., Schröder, B. und Schmitz, H.-C., Hrsg. , *Computerlinguistik—Was geht, was kommt?*, S. 149–153. Gardez! Verlag.
- Heyer, G., Quasthoff, U. und Wittig, T. (2006): *Text Mining: Wissensrohstoff Text*. W3L-Verlag, Herdecke, Bochum.

- Hirschman, L. und Sager, N. (1982): „Automatic Information Formatting of a Medical Sublanguage.“. In: Kittredge, R. und Lehrberger, J., Hrsg. , *Sublanguage: Studies of Language in Restricted Semantic Domains*, S. 27–80. Walter de Gruyter, Berlin.
- Hobbs, J. R. (1993): „The Generic Information Extraction System“. In: *Fifth Message Understanding Conference (MUC-5)*, S. 87–92. Baltimore, Maryland.
- Hobbs, J. R. (2002): „Information Extraction from Biomedical Text“. In: *Journal of Biomedical Informatics*, **35**(4), 260–264.
- Hobbs, J. R. (2004): „Some Notes on Performance Evaluation for Natural Language Systems“. <http://www.isi.edu/~hobbs/performance-evaluation.pdf>.
- Hobbs, J. R., Appelt, D. E., Bear, J., Israel, D., Kameyama, M., Stickel, M. und Tyson, M. (1997): „FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text“. In: Roche, E. und Schabes, Y., Hrsg. , *Finite-State Language Processing*, S. 383–406. Cambridge, Massachusetts: MIT Press.
- Hobbs, J. R., Appelt, D. E., Bear, J., Israel, D. und Tyson, M. (1992): „FASTUS: A System for Extracting Information from Natural-Language Text“. Arbeitspapier, Nr.519, Menlo Park, California, SRI International.
- Hoeppner, W., Christaller, T., Marburger, H., Morik, K., Nebel, B., O’Leary, M. und Wahlster, W. (1983): „Beyond Domain-Independence: Experience With the Development of a German Language Access System to Highly Diverse Background Systems“. In: *IJCAI*, S. 588–594.
- Hoffmeister, J., Müller, C. und Westkämper, E. (2007): *Sprachtechnologie in der Anwendung – Sprachportale*. Springer Verlag.
- Hong, D. (2006): „An Introductory Guide to Speech Recognition Solutions“. White paper, Datamonitor.
- Horacek, H. (2009): „Textgenerierung“. In: Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, S. 436–465. Heidelberg: Spektrum Akademischer Verlag.
- Horndasch, A., Gallwitz, F., Haas, J. und Nöth, E. (2005): „Der mixed-initiative Ansatz als Basis für benutzerfreundliche Sprachdialsysteme“. In: *KI*, **3**, 38–41.
- Hovy, E. (1987): *Generating Natural Language under Pragmatic Constraints*. Dissertation, Yale University.
- Hovy, E. (1993): „Automated Discourse Generation Using Discourse Structure Relations“. In: *Artificial Intelligence*, **63**(1–2), 341–386.
- Hovy, E. (2003): „Text Summarization“. In: Mitkov, R., Hrsg. , *The Oxford Handbook of Computational Linguistics*, S. 583–598. Oxford University Press.
- Hovy, E., Gerber, L., Hermjakob, U., Junk, M. und Lin, C.-Y. (2001): „Question Answering in Webclopedia“. In: *Proceedings of the TREC-9 Conference*. NIST, Gaithersburg.
- Hovy, E. und Lin, C.-Y. (1999): „Automatic Text Summarization in SUMMARIST“. In: Mani, I. und Maybury, M., Hrsg. , *Advances in Automatic Text Summarization*, S. 81–94. MIT Press.
- Hovy, E. H. (2005): „Automated Text Summarization“. In: Mitkov, R., Hrsg. , *The Oxford Handbook of Computational Linguistics*, S. 583–598. Oxford: Oxford University Press.

- Hutchins, J. (2005): „Machine translation (computer-based translation)“. <http://www.hutchinsweb.me.uk>.
- Hutchins, W. J. und Somers, H. L. (1992): *An Introduction to Machine Translation*. London: Academic Press.
- Indurkhya, N. und Damerau, F. (2010): *Handbook of natural language processing*. Chapman & Hall/CRC machine learning & pattern recognition series. 2. Auflage, Taylor and Francis.
- Jacobs, P. S. und Rau, L. F. (1990): „SCISOR: Extracting Information from On-Line News“. In: *Communications of the ACM*, **33**(11), 88–97.
- Jameson, A., Hoeppner, W. und Wahlster, W. (1980): „The natural language System HAM-RPM as a hotel manager: Some representational prerequisites.“. In: Wilhelm, R., Hrsg. , *GI-10. Jahrestagung*, S. 459–473. Berlin, Heidelberg, N.Y.: Springer.
- Jekat, S. und Volk, M. (2009): „Maschinelle und computergestützte Übersetzung“. In: Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, chapter 5. Anwendungen. 3. überarbeitete und erweiterte Auflage, Heidelberg: Spektrum Akademischer Verlag.
- Joensson, A. und Dahlbäck, N. (2000): „Distilling Dialogues – A Method Using Natural Dialogue Corpora for Dialogue Systems Development“. In: *Proceedings of 6th Applied Natural Language Processing Conference and 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, S. 44–51. Seattle.
- Johnston, M. und Bangalore, S. (2005): „Finite-state Multimodal Integration and Understanding“. In: *Journal of Natural Language Engineering*, **11**(2), 159–187.
- Johnston, M., Bangalore, S., Stent, A., Vasireddy, G. und Ehlen, P. (2002): „Multimodal language processing for mobile information access“.
- Jurafsky, D. und Martin, J. H. (2008): *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2. überarbeitete und erweiterte Auflage, Prentice-Hall.
- Kaiser, K. und Miksch, S. (2005): „Information Extraction–A Survey“. Asgaard-tr-2005-6, Vienna University of Technology, Institute of Software Technology & Interactive Systems.
- Katz, B. (1997): „Annotating the World Wide Web Using Natural Language“. In: *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAOP '97)*.
- Kellner, A. (2004): „Dialogsysteme“. In: Carstensen, K.-U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, chapter 5. Anwendungen, S. 532–539. 2. Auflage, Spektrum Akademischer Verlag.
- Klemann, J. und Steimel, B. (2004): „Praxisreport: Service Automatisierung - Sprachanwendungen auf dem Vormarsch“. Arbeitspapier, STRATECO GmbH/mind Business Consultants.
- Kopp, S., Gesellensteller, L., Krämer, N. und Wachsmuth, I. (2005): „A conversational agent as museum guide – design and evaluation of a real-world application“. In: et al., T. P., Hrsg. , *Intelligent Virtual Agents*, Bd.3661, *LNAI*, S. 329–343. Berlin: Springer-Verlag.

- Kopp, S., Jung, B., Lessmann, N. und Wachsmuth, I. (2003): „Max—A Multimodal Assistant in Virtual Reality Construction“. In: *KI-Künstliche Intelligenz*, **4/03**, 11–17.
- Kupiec, J. (1993): „MURAX: A Robust Linguistic Approach for Question Answering Using an On-Line Encyclopedia“. In: Korfhage, R., Rasmussen, E. M. und Willett, P., Hrsg., *SIGIR '93: Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, S. 181–190. Pittsburgh, PA, USA: ACM Press.
- Kupiec, J., Pedersen, J. O. und Chen, F. (1995): „A Trainable Document Summarizer“. In: *Research and Development in Information Retrieval*, S. 68–73. wiederabgedruckt in: *Mani und Maybury* 1999, S. 55–60.
- Kurpinski, C. (2005): „Sprachdialogsysteme“. In: *Funkschau*, **15-16**, 36–38.
- Kwok, C. T., Etzioni, O. und Weld, D. S. (2001): „Scaling Question Answering to the Web“. In: *Tenth International World Wide Web Conference*, S. 150–161. Seattle, WA, USA.
- Lavoie, B. und Rambow, O. (1997): „A Fast and Portable Realizer for Text Generation Systems“. In: *Proceedings of the Fifth Conference on Applied Natural Language Processing*. Washington, DC.
- Lehnert, W. (1982): „Plot Units: A Narrative Summarization Strategy“. In: Lehnert, W. und Ringle, M., Hrsg., *Strategies for Natural Language Processing*. Hillsdale, N. J.: Lawrence Erlbaum Associates.
- Lehnert, W. G. (1978): *The Process of Question Answering*. Hillsdale, NJ.: Lawrence Erlbaum Associates.
- Lemmetty, S. (1999): „Review of Speech Synthesis Technology“. Diplomarbeit, Helsinki University of Technology, Department of Electrical and Communications Engineering.
- Lenat, D. B. und Guha, R. V. (1989): *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Leuski, A., Lin, C. Y. und Hovy, E. (2003): „iNeATS: Interactive multi-document summarization“. In: *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, S. 125–128. Sapporo, Japan.
- Levelt, W. (1989): *Speaking. From Intention to Articulation*. The MIT Press.
- Lin, C. und Hovy, E. (2002): „Automated multi-document summarization in NeATS“.
- Lin, C.-Y. (2004): „ROUGE: a Package for Automatic Evaluation of Summaries“. In: *In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*. Barcelona, Spain.
- Lindsay, R. K. (1963): „Inferential Memory as the Basis of Machines Which Understand Natural Language“. In: Feigenbaum, E. und Feldman, J., Hrsg., *Computers and Thought*, S. 217–233. McGraw-Hill, New York.
- Lobin, H. (2009): *Computerlinguistik und Texttechnologie*. W. Fink UTB.
- Lobin, H. und Lemnitzer, L., Hrsg. (2004): *Texttechnologie. Perspektiven und Anwendungen*. Tübingen: Stauffenburg.
- Lockwood, R. und Joscelyne, A. (2003): „Summary of Benchmarking HLT progress in Europe“. Euromap-report, Copenhagen, HLT Central.

- Luhn, H. P. (1958): „Automatic Creation of Literature Abstracts“. In: *IBM Journal of Research & Development*, **2**(2), 159–165. wiederaabgedruckt in: *Mani und Maybury* 1999, S. 15–21.
- Maegaard, B. (1999): „Machine Translation“. In: Hovy, E., Ide, N., Frederking, R., Mariani, J. und Zampolli, A., Hrsg. , *Multilingual Information Management: Current Levels and Future Abilities*. Pisa, Italy: Instituti Editoriali e Poligrafici Internazionali.
- Manaris, B. (1998): „Natural Language Processing: A Human-Computer Interaction Perspective“. In: Zelkowitz, M. V., Hrsg. , *Advances in Computers*, Bd. 47, S. 1–66. Academic Press, New York.
- Mani, I. und Maybury, M. T., Hrsg. (1999): *Advances in Automatic Text Summarization*. Cambridge, MA: MIT Press: MIT Press.
- Mann, W. C. und Moore, J. A. (1981): „Computer Generation of Multiparagraph English Text“. In: *American Journal of Computational Linguistics*, **7**(1), 17–29.
- Mann, W. C. und Thompson, S. A. (1988): „Rhetorical Structure Theory: Toward a functional theory of text organization“. In: *Text*, **8**(3), 243–281.
- Manning, C. D., Raghavan, P. und Schütze, H. (2008): *Introduction to Information Retrieval*. Cambridge University Press.
- Marcu, D. (1997): *The rhetorical parsing, summarization, and generation of natural language texts*. Dissertation, University of Toronto.
- Marcu, D. (2000): *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press.
- Martin, D. L., Cheyer, A. J. und Moran, D. B. (1999): „The Open Agent Architecture: A Framework for Building Distributed Software Systems“. In: *Applied Artificial Intelligence*, **13**(1-2), 91–128.
- Maybury, M. T. (2001): „Human Language Technologies for Knowledge Management: Challenges and Opportunities“. Arbeitspapier, The MITRE Corporation.
- Maybury, M. T. und Mani, I. (2001): „Automatic Summarization“. Tutorial, The MITRE Corporation.
- Mazdak, N. (2004): „FarsiSum - a Persian text summarizer“. Diplomarbeit, Department of Linguistics, Stockholm University.
- McDonald, D. D. (1980): *Natural Language Production as a Process of Decision Making under Constraint*. Dissertation, MIT, Cambridge, MA.
- McDonald, D. D. (1993): „Issues in the Choice of a Source for Natural Language Generation“. In: *Computational Linguistics*, **19**(1), 191–197.
- McKeown, K. (1985): *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge, U.K.: Cambridge University Press.
- McKeown, K., Robin, J. und Kukich, K. (1995): „Generating Concise Natural Language Summaries“. In: *Inf. Process. Manage.*, **31**(5), 703–733.
- McRoy, S. W., Channarukul, S. und Ali, S. S. (2001): „Creating Natural Language Output for Real-Time Applications“. In: *ACM SIGART publication 'intelligence': new visions of AI in practice*, **12**(2), 21–34.
- McRoy, S. W., Channarukul, S. und Ali, S. S. (2003): „An Augmented Template-Based Approach to Text Realization“. In: *Natural Language Engineering*, **9**(4), 381–420.

- McTear, M. F. (2002): „Spoken dialogue technology: enabling the conversational user interface“. In: *ACM Computing Surveys*, **34**(1), 90–169.
- McTear, M. F. (2004): *Spoken Dialogue Technology: Toward the Conversational User Interface*. Springer Verlag, London.
- Meehan, J. R. (1976): *The Metanovel: Writing Stories by Computer*. Dissertation, Yale University.
- Mellish, C. und Dale, R. (1998): „Evaluation in the context of natural language generation“. In: *Computer Speech and Language*, **12**, 349–373.
- Menzel, W. (2007): „Sprachverarbeitung“. In: Görz, G., Rollinger, C.-R. und Schneeberger, J., Hrsg., *Einführung in die künstliche Intelligenz*. R. Oldenbourg Verlag, München.
- Minock, M. (2005a): „A Phrasal Approach to Natural Language Interfaces over Databases“. Report, Nr.05-09, University of Umeå, Department of Computing Science.
- Minock, M. (2005b): „Where are the 'Killer Applications' of Restricted Domain Question Answering?“. In: *Proceedings of Knowledge and Reasoning in Question Answering (KRAQ-2005)*, S. 98–101.
- Minock, M. (2006): „Natural Language Updates to Databases through Dialogue“. In: *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems (NLDB-2006)*, S. 203–208. Klagenfurt, Austria.
- Mitkov, R., Hrsg. (2003): *Oxford Handbook of Computational Linguistics*. Oxford University Press.
- Mollá, D., Schwitter, R., Rinaldi, F., Dowdall, J. und Hess, M. (2003): „Answer Extraction from Technical Texts“. In: *IEEE Intelligent Systems*, **1**, 12–17.
- Mollá, D. und Vicedo, J. L. (2007): „Question Answering in Restricted Domains: An Overview“. In: *Computational Linguistics*, **33**(1), 41–61.
- Neal, J. und Shapiro, S. (1991): „Intelligent Multimedia Interface Technology.“. In: Sullivan, J. und Tyler, S., Hrsg. , *Intelligent User Interfaces*, S. 11–44. ACM Press.
- Neumann, G. (2004): „Informationsextraktion“. In: Carstensen, K.-U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, chapter 5. Anwendungen, S. 502–510. 2. Auflage, Spektrum Akademischer Verlag.
- Neumann, G. (2009): „Text-basiertes Informationsmanagement“. In: Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie – Eine Einführung*, chapter 5. Anwendungen. 3. Auflage, Springer Verlag.
- O'Donnell, M., Mellish, C., Oberlander, J. und Knott, A. (2001): „ILEX: an architecture for a dynamic hypertext generation system“. In: *Natural Language Engineering*, **7**, 225–250.
- Oertle, H. (2006): *Computerlinguistik: Schlüsseltechnologie des 21. Jahrhunderts*. Lulu.
- Ogden, W. C. und Bernick, P. (1997): „Using Natural Language Interfaces“. In: Helander, M., Hrsg. , *Handbook of Human-Computer Interaction*. North Holland: Elsevier.
- Orăsan, C. (2006): *Comparative evaluation of modular automatic summarisation systems using CAST*. Dissertation, Wolverhampton, UK, School of Humanities, Languages and Social Sciences, University of Wolverhampton.

- Orăsan, C., Mitkov, R. und Hasler, L. (2003): „CAST: a Computer-Aided Summarisation Tool“. In: *Proceedings of EACL2003*, S. 135 – 138. Budapest, Hungary.
- Over, P. und Yen, J. (2004): „An introduction to duc 2004 intrinsic evaluation of generic new text summarization systems“. In: *In Proceedings of DUC 2004 Document Understanding Workshop*. Boston.
- Oviatt, S. (2002): „Multimodal Interfaces“. In: Jacko, J. und Sears, A., Hrsg., *Handbook of Human-Computer Interaction*. New Jersey: Lawrence Erlbaum.
- Oviatt, S. L. (1999): „Ten myths of multimodal interaction“. In: *Communications of the ACM*, **42**(11), 74–81.
- Pickett, J. und Schroeder, J. (1999): „Speech Technology“. In: Pickett, J., Hrsg., *The Acoustics of Speech Communication*, S. 324–342. Allyn & Bacon: Boston, MA.
- Poller, P. (2002): „COMIC Deliverable 1.1: Review of the state of the art in system architectures“. Arbeitspapier.
- Popescu, A.-M. (2007): *Information Extraction from Unstructured Web Text*. Dissertation, University of Washington.
- Popescu, A.-M., Armanasu, A., Etzioni, O., Ko, D. und Yates, A. (2004): „Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability“. In: *Proceedings of COLING 2004*.
- Popescu, A.-M., Etzioni, O. und Kautz, H. (2003): „Towards a Theory of Natural Language Interfaces to Databases“. In: *Proceedings of IUI 2003*.
- Quillian, M. R. (1967): „Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities“. In: *Behavioral Science*, **12**, 410–430.
- Quillian, M. R. (1969): „The teachable language comprehender: a simulation program and theory of language“. In: *Communications of the ACM*, **12**(8), 459–476.
- Radev, D. R., Hovy, E. und McKeown, K. (2002): „Introduction to the special issue on summarization“. In: *Computational Linguistics*, **28**(4), 399–408.
- Rambow, O. (1990): „Domain Communication Knowledge“. In: *Proceedings of the Fifth International Workshop on Natural Language Generation*. Dawson, PA.
- Raphael, B. (1968): „SIR: A computer program for semantic information retrieval“. In: Minsky, M., Hrsg., *Semantic Information Processing*, S. 33–134. Cambridge, MA: MIT Press.
- Ravichandran, D. und Hovy, E. H. (2002): „Learning Surface Text Patterns for a Question Answering System“. In: *Proceedings of the 40th ACL conference*. Philadelphia.
- Reape, M. und Mellish, C. (1999): „Just what is aggregation anyway?“. In: *European Workshop on Natural Language Generation*.
- Reiter, E. (2007): „The Shrinking Horizons of Computational Linguistics“. In: *Computational Linguistics*, **33**(2), 283–287.
- Reiter, E. und Dale, R. (2000): *Building Natural Language Generation Systems*. Cambridge University Press.

- Reithinger, N. und Blocher, A. (2003): „SmartKom - Multimodale Mensch-Technik-Interaktion“. Arbeitspapier, Nr. 40, DFKI.
- Riesbeck, C. K. und Schank, R. C. (1982): *Inside Computer Understanding: Five Programs Plus Miniatures*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.
- Ronthaler, M. (2001): „Natürlichsprachliche Retrieval-Schnittstellen“. In: Carstensen, K.-U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R. und Langer, H., Hrsg. , *Computerlinguistik und Sprachtechnologie. Eine Einführung.*, S. 477–483. Spektrum Akademischer Verlag.
- Sadek, D. (1999): „Design Considerations on Dialogue Systems: From Theory to Technology - The Case of Artimis -“. In: *ESCA Interactive Dialogue in Multi-modal systems '99*.
- Sager, N. (1978): „Natural Language Information Formatting: The Automatic Conversion of Texts to a Structured Data Base.“. In: *Advances in Computers*, **17**, 89–162.
- Sager, N., Lyman, M., Bucknall, C., Nhàn, N. und Tick, L. (1994): „Natural Language Processing and the Representation of Clinical Data“. In: *Journal of the American Medical Informatics Association*, **1**(2), 142–160.
- Salton, G. und McGill, M. J. (1987): *Information Retrieval. Grundlegendes für Informationswissenschaftler*. McGraw-Hill, Maidenhead.
- Sarter, N. B. (2006): „Multimodal information presentation: Design guidance and research challenges“. In: *International Journal of Industrial Ergonomics*, **36**, 439–445.
- Schank, R. und Abelson, R. (1977): *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Erlbaum Assoc.
- Schukat-Talamazzini, E. (1995): *Automatische Spracherkennung. Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg Verlag.
- Schwitter, R. (2002): „English as a Formal Specification Language“. In: *Proceedings of the Thirteenth International Workshop on Database and Expert Systems Applications (DEXA 2002)*, W04: *Third International Workshop on Natural Language and Information Systems - NLIS*, S. 228–232. Aix-en-Provence, France.
- Scott, D. und Moore, J. (2006): „An NLG evaluation competition? Eight Reasons to be Cautious“. Technical Report, Nr.2006/09, Department of Computing, The Open University.
- Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P. und Zue, V. (1998): „Galaxy-II: A Reference Architecture for Conversational System Development“. In: *Proc. ICSLP*. Sydney, Australia.,
- Silber, H. G. und McCoy, K. F. (2002): „Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization“. In: *Computational Linguistics*, **28**(4), 487–496.
- Simmons, R. F. (1965): „Answering English questions by computer: a survey“. In: *Commun. ACM*, **8**(1), 53–70.
- Simmons, R. F. und Slocum, J. (1972): „Generating English Discourse from Semantic Networks“. In: *Communications of the ACM*, **15**(7), 891–905.

- Small, S., Strzalkowski, T., Lu, T., Ryan, S., Salkin, R., Shimizu, N., Kantor, P., Kelly, D., Rittman, R., Wacholder, N. und Yamrom, B. (2004): „HITIQA: Scenario-based question answering“. In: *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*.
- Spärck Jones, K. (2007): „Computational Linguistics: What About the Linguistics?“. In: *Computational Linguistics*, **33**(3), 437–441.
- Steedman, M. (2008): „On becoming a Discipline“. In: *Computational Linguistics*, **34**(1), 37–144.
- Steimel, B., Jameson, A., Klöckner, K. und Paulke, S. (2005): „VOICE Award 2005 - Die besten deutschsprachigen Sprachapplikationen [VOICE Award 2005 - The Best German Speech Applications]“. Arbeitspapier, m i n d Business Consultants, Strateco GmbH, DFKI.
- Steimel, B., Jameson, A., Klöckner, K. und Paulke, S. (2006): „VOICE Award 2006 - Die besten deutschsprachigen Sprachapplikationen [VOICE Award 2006 - The Best German Speech Applications]“. Arbeitspapier, m i n d Business Consultants, Strateco GmbH, DFKI.
- Steimel, B. und Klemann, J. (2004): „Praxisreport–Sprachapplikationen auf dem Prüfstand [executive version]“. Arbeitspapier, STRATECO GmbH/mind Business Consultants.
- Stock, O. (1991): „Natural Language and Exploration of an Information Space: The ALFresco Interactive System“. In: *Proc. of the 12th IJCAI*, S. 972–978. Sidney, Australia.
- Surdeanu, M. und Harabagiu, S. (2002): „Infrastructure for Open-Domain Information Extraction“. In: *Proceedings of the Human Language Technology Conference (HLT 2002)*. San Diego, California.
- Surdeanu, M. und Pasca, M. (2002): „Question Answering Techniques and Systems“. Dep. Llenguatges i Sistemes Informàtics.
- Tennant, H. (1981): *Natural Language Processing: An Introduction to an Emerging Technology*. Princeton, NJ, USA: Petrocelli Books, Inc.
- Thione, G. L., van den Berg, M., Polanyi, L. und Culy, C. (2004): „Hybrid Text Summarization: Combining external relevance measures with Structural Analysis“. In: *Proceedings of the ACL2004 Workshop Text Summarization Branches Out*. Barcelona, Spain.
- Thompson, F. B. und Thompson, B. (1983): „Introducing ASK, a Simple Knowledgeable System“. In: *Proceedings Conference on Applied Natural Language Processing*.
- Traum, D. (1996): „Conversational agency: The trains-93 dialogue manager“.
- Traum, D. und Larsson, S. (2003): „The Information State Approach to Dialogue Management“. In: Smith und Kuppeveld, Hrsg. , *Current and New Directions in Discourse & Dialogue*, S. 325–353. Kluwer Academic Publishers.
- van den Berg, M., Polanyi, L., Culy, C. und Thione, G. L. (2004): „A Rule Based Approach to Discourse Parsing“. In: *Proceedings of the 5th SIGdial Workshop in Discourse And Dialogue*, S. 108–117. Cambridge, MA USA.
- Wachsmuth, I. (2005): „’Ich, Max’ – Kommunikation mit künstlicher Intelligenz“. In: Herrmann, C. S., Pauen, M., Rieger, J. und Schicktanz, S., Hrsg. , *Bewusstsein: Philosophie, Neurowissenschaften, Ethik*, S. 329–354. München: Wilhelm Fink Verlag (UTB).

- Wahlster, W. (1982): „Natürlichsprachliche Systeme. Eine Einführung in die sprachorientierte KI-Forschung“. In: Bibel, W. und Siekmann, J., Hrsg., *Künstliche Intelligenz*, S. 203–283. Heidelberg: Springer.
- Wahlster, W. (2000): „Mobile Speech-to-Speech Translation of Spontaneous Dialogs: An Overview of the Final Verbmobil System“. In: Wahlster, W., Hrsg., *Verbmobil: foundations of speech-to-speech translation*, S. 3–21. Berlin: Springer.
- Wahlster, W. (2006a): „Dialogue Systems Go Multimodal“. In: Wahlster, W., Hrsg., *SmartKom: Foundations of Multimodal Dialogue Systems*, Cognitive Technologies Series, S. 3–27. Heidelberg, Germany: Springer.
- Wahlster, W., Hrsg. (2006b): *SmartKom: Foundations of Multimodal Dialogue Systems*. Cognitive Technologies Series. Heidelberg, Germany: Springer.
- Wahlster, W. (2007): „SmartWeb: Ein multimodales Dialogsystem für das semantische Web“. In: Reuse, B. und Vollmar, R., Hrsg., *40 Jahre Informatikforschung in Deutschland*. Heidelberg, Berlin, Springer.
- Wahlster, W., André, E., Bandyopadhyay, S., Graf, W. und Rist, T. (1995): „WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation“. In: Ortony, A., Slack, J. und Stock, O., Hrsg., *Communication from an Artificial Intelligence Perspective: Theoretical and Applied Issues*. Berlin: Springer.
- Wahlster, W., Reithinger, N. und Blocher, A. (2001): „SmartKom: Multimodal Communication with a Life-Like Character“. In: *Proceedings of Eurospeech 2001*, Bd. 3, S. 1547 – 1550. Aalborg, Denmark.
- Waltz, D. (1978): „An English question answering system for a large relational database“. In: *Communications of the ACM*, **21**(7), 526–539.
- Warren, D. H. D. und Pereira, F. C. N. (1982): „An Efficient Easily Adaptable System for Interpreting Natural Language Queries“. In: *American Journal of Computational Linguistics*, **8**(3-4), 110–122.
- Webber, B. (1992): „Question answering“. In: Shapiro, S. C., Hrsg., *Encyclopedia of Artificial Intelligence*, S. 814–822. New York, NY, USA: John Wiley & Sons, Inc.
- Weizenbaum, J. (1966): „ELIZA - A Computer Program For The Study Of Natural Language Communication Between Man And Machine“. In: *Communications of the ACM* **9**(1), S. 36–45.
- Wilensky, R., Chin, D. N., Luria, M., Martin, J., Mayfield, J., und Wu, D. (1994): „The Berkeley Unix Consultant project“. In: *Computational Linguistics*, **14**(4), 35–84.
- Willée, G., Schröder, B. und Schmitz, H.-C., Hrsg. (2002): *Computerlinguistik—Was geht, was kommt? Gardez! Verlag*.
- Winograd, T. (1972): *Understanding Natural Language*. New York: Academic Press.
- Woods, W., Kaplan, R. und Nash-Webber, B. (1972): „The Lunar Sciences Natural Language Information System: Final Report“. BBN Report, Nr.2378, Bolt Beranek and Newman, Cambridge, Mass.
- Xu, F., Uszkoreit, H. und Li, H. (2007): „A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity“. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, S. 584–591.

- Zarri, G. P. (1983): „Automatic Representation of the Semantic Relationships Corresponding to a French Surface Expression“. In: *ACL Proceedings, Conference on Applied Natural Language Processing*. ACL, Santa Monica, Calif.

Index

- A.L.I.C.E., 117
AAC, ↪ Alternative and Augmentative Communication
Abdeckung
 konzeptuelle, ↪ konzeptuelle Abdeckung
 linguistische, ↪ sprachliche Abdeckung
ACE, ↪ Automatic Content Extraction
AIML, ↪ Artificial Intelligence Markup Language
Alignierung, 193
ALPAC-Report, 184, 185
Alternative and Augmentative Communication,
 ↪ elektronische Kommunikationshilfen
Ambiguität, 33, 35, 186
Analyseproblem, 186
Anapher, 61, 69, 73, 120, 155, 160, 201, 202
Anaphernauflösung, 35, 155
Anaphernresolution, ↪ Anaphernauflösung
Anfragenrelaxierung, 88, 109
answer fusion, 94, 96
Antwortarten, 98
Antworten, 99, 121
Antwortextraktion, 14, 101, 102, 105
Antwortmaschine, 12, 51, 52
Antworttyp, 102, 105
AQUAINT Korpus, 111
Artificial Intelligence Markup Language, 117
ARTIMIS, 132
ASR, ↪ automatic speech recognition
Aufgabenmodell, 39
Ausbeute, 61
Auskunftssystem, 118
Auswahlproblem, 33
Automatic Content Extraction, 60
automatic speech recognition, ↪ Spracherkennung
Avatar, 150
Barge-In, 145
Baumbanken, 16
BDI-Modell, 127, 162
Benutzeradaptivität, 94
Benutzerinteraktion, 99
Benutzermodell, 39, 116, 122
Benutzermodellierung, 168
Benutzerschnittstelle, 155
Bing, 12
Biometrie, 153
Blackboard, 44
Blackboardarchitektur, 29
Boole'sche Suche, 50
business intelligence, 53, 57, 61
CALL, ↪ Computer-Assisted Language Learning
Call-Center, 140
canned text, 174
CAT, ↪ computer aided translation
Chat-80, 78, 211
Chatterbot, 26, 118, 163, 211
CL, ↪ Computerlinguistik
CLEF, ↪ Cross Language Evaluation Forum
Code, 150
Coercion, 83
cognitive process, 10
common ground, 135
compression ratio, 205

- computer aided summarization, ↪ Computer-gestützte Textzusammenfassung
computer aided translation, ↪ Computer-gestützte Übersetzung
computer-aided natural language processing, ↪ Computer-gestützte Sprachverarbeitung
Computer-Assisted Language Learning, ↪ computergestütztes Sprachlehren/-lernen
Computer-gestützte Übersetzung, 16, 181
Computer-gestützte Sprachverarbeitung, 46
Computer-gestützte Textzusammenfassung, 198, 207
computergestützte Lexikographie, 14
computergestütztes Sprachlehren/-lernen, 16
Computerlinguistik, 7–20
concept-to-speech synthesis, 14, 41
conceptual coverage, ↪ konzeptuelle Abdeckung
Constraint, 35
conversational humanoid, 158
Cross Language Evaluation Forum, 112
Cyc, 95
DARPA, 60
Data mining, 15
Datenbanken, 73, 112
deep processing, ↪ tiefe Verarbeitung
DeepQA, 110
Default, 40
Definitionsfrage, 111
Deixis, 120
deklarativ, 23
Destillieren (von Dialogen), 129
Dialogakt, 123, 162
multimodaler, 156
Dialogakterkennung, 123, 127
Dialoggeschichte, 121
Dialoggrammatik, 125
Dialoghistorie, 116
Dialogmanagement, 123, 133
Dialogmodell, 39, 116, 121
Dialogschritt, 121, 123, 127, 132, 135
Dialogsteuerung, 123
Dialogsystem, 115–137
dialogue act, ↪ Dialogakt
dialogue game, 125
dialogue history, ↪ Dialoggeschichte
dialogue move, ↪ Dialogschritt
Diktiersystem, 7
Disambiguierung, 33
discourse obligation, ↪ Diskursverpflichtung
Diskursrelationen, 171
Diskursverarbeitung, 64
Diskursverpflichtung, 132
document retrieval, ↪ Dokumenterschließung
Document Understanding Conferences, 46, 205–207
Dokumenterschließung, 14
Dokumentklassifikation, 14
Dolmetschen
maschinell gestütztes, 44
Domäne, 39
Domänenmodell, 39, 122
DS, ↪ Dialogsystem
DUC, ↪ Document Understanding Conferences
Earcon, 139
Effektivität, 32
Effizienz, 32, 140
Eigennamenerkennung, 65, 76, 102, 178
Einzelworterkennung, 144
elektronische Kommunikationshilfen, 16, 197
ELIZA, 23, 117, 211
Ellipse, 37, 73, 81, 120, 131, 202
Ellipsenauflösung, 131
embodied conversational agent, 158
endlicher Automat, 54, 65, 126
Erlernbarkeit, 145
ExtrAns, 108
F-Maß, 61
F-measure, ↪ F-Maß
fact retrieval, 51
FAHQ(M)T, ↪ fully automatic high quality (machine) translation
Fakten-retrieval, 51
Faktoid, 111
FAT, ↪ Fully Automatic Translation
Feedback, ↪ Rückmeldung

- Fehlannahme, 37, 88
Fehlerbehandlung, 145
Filter, 64
finite-state script, 123, 124
flache Verarbeitung, 13, 41, 44
Flexibilität, 32, 140
Frage-/Antwort-System, 93–114
Fragebeantwortung, 14
 intensionale, 88
Fragegenerierung, 178
Fragen, 96
Frageklassifikation, 97
Frageverarbeitung, 102
Fragmentkombination, 64
Frame Problem, 33
fully automatic high quality (machine) translation, 181, 184
Fully Automatic Translation, 181
- GATE, 65
Gazetteer, 65
Gebrauchstauglichkeit, 32, 141
gemeinsamer Plan, 128
Genauigkeit (precision), 61
generation gap, 173
Generierungslücke, 173
Generierungsproblem, 186
geschriebene Sprache, 13
gesprochene Sprache, 13
Gestik, 150
gold standard, 205
graceful degradation, 31
Grammatik-Werkbänke, 16
Grammatikkorrektur, 14
GREC-MSR-Korpus, 177
grounding, 116
GUS, 119
- HAMT, ↗ human aided machine translation
Hear&Feel, 141, 146
hierarchische Planung, 127
HLT, ↗ human language technology
Homografie, 36, 186
Homonymie, 36, 54
Homophonie, 36
- human aided machine translation, 181
human language technology, 8
hybride Systeme, 44, 194, 207
Hypernymie, 55, 203
Hyponymie, 55
- IE, ↗ Informationsextraktion
index term, 50
indirekter Sprechakt, 116, 127
Inferenz, 24, 29, 33
Inferenzen, 71
information retrieval, ↗ Informationserschließung
information state, 133
Informationserschließung, 15, 50, 57
Informationsextraktion, 57–72
Infotainment-Assistent, 159
Inhaltsfestlegung, 173
Inhaltsorganisation, 173
inkrementell, 169
Interactive Voice Response, 118, 139
Interlingua, 182, 189, 191
IR, ↗ information retrieval
IS, ↗ information state
IVR, ↗ Interactive Voice Response
- Künstliche-Intelligenz-Forschung, 8, 23
keyword, 50
KI, ↗ Künstliche Intelligenz-Forschung
Klärungsdialog, 93
Knowledge Base Population, 61
Knowledge Engineering, 62
knowledge management, ↗ Wissensmanagement
Kohärenz, 33
Kohäsion, 33, 203
Kommunikationsassistent, 150, 158, 159
Kompetenz, 46
Komplexität, 35
Kontinuität, 33
kontrollierte Sprache, 85, 185, 195
konversationeller Agent, 128
Konzepthierarchien, 40
konzeptuelle Abdeckung, 31, 75
Kooperativität, 88, 119

- Koreferenzauflösung, 41, 64, 154
Koreferenzresolution, ↪ Koreferenzauflösung
Korpora, 10, 16, 42, 125
 parallele, 193
Korpus-basiertes Verfahren, 10
Kundenfreundlichkeit, 140

language engineering, 8, 10
LDV, ↪ Linguistische Datenverarbeitung
Levenshtein-Distanz, 54
lexical chain, 203
Lexikalische Disambiguierung, 64
Lexikographie
 computergestützte, 14
LF, ↪ logische Form
Lilog, 113
Lingubot, 7, 118
linguistic coverage, ↪ linguistische Abdeckung
linguistische Abdeckung, ↪ sprachliche Abdeckung
Linguistische Datenverarbeitung, 8
Listenfrage, 111
Location-Based Services, 160
Loebner Preis, 117
Logik, 27, 77
logische Form, 107

MÜ, ↪ maschinelle Übersetzung
MÜS, ↪ maschinelle Übersetzung System
machine aided human translation, 181
machine learning Verfahren, 62
machine translation, ↪ maschinelle Übersetzung
MAHT, ↪ machine aided human translation
Makroplanung, 174
maschinelle Übersetzung, 181–196
 Paradigmen, 192
maschinelle Übersetzung System, 181
 direktes, 189
 Interlingua-, 191
 Transfer-, 191
maschinelle Sprachverarbeitung, 7–20, 185
Max, 162
Mean reciprocal rank, 111

Mehrworterkennung, 65
message, 171
Message Understanding Conference, 46, 59, 61, 69–72
Metonymie, 186
Mikroplanung, 174
Mimik, 150
misconception, ↪ Fehlannahme
mixed initiative dialogue, 116
modality allocation, 156
modality realization, 157
Modularität, 31
MRR, ↪ Mean reciprocal rank
MT, ↪ machine translation
MUC, ↪ Message Understanding Conference
Multi-Blackboard-Architektur, 45
Multilingualität, 31, 94, 112
multimedial, 150
multimodal fission, 154, 156
multimodal fusion, 154
multimodales System, 17, 149–165
Multimodalität, 150
 symmetrische, 150, 158, 161
Multislot-Erkennung, 144
Muster, 59

Nachhaltigkeit, 32
named entity, 59
named entity recognition, ↪ Eigennamenerkennung
natürliche Spracherkennung, 144
natürlichsprachliche Schnittstellen, 16
natürlichsprachliche Schnittstellen zu Datenbanken, 73–91
natürlichsprachliche Suche, 12, 49–56
natürlichsprachliches System, 8, 17, 23–47
 Beispiele, 28
 Evaluationskriterien, 31
 Fähigkeiten, 24
 Typen von, 24
 Wissensarten, 39
National Institute of Science and Technology, 60
natural language generation, 167

- natural language interfaces to databases, ↗ natürlichsprachliche Schnittstellen zu Datenbanken
natural language processing, 1, 8
natural language system, ↗ natürlichsprachliches System
nicht-sprachliches Wissen, 23, 39
NIST, ↗ National Institute of Science and Technology
NLG, ↗ natural language generation
NLIDB, ↗ natural language interfaces to databases
NLP, ↗ natural language processing
NLS, ↗ natural language system
NLU, ↗ natürliche Spracherkennung
numerisches Verfahren, 10
- Ontologien, 40, 59, 113, 168
open domain information extraction, 63
open domain QA, ↗ open domain question answering
open domain question answering, 95, 99–107
Other-question, 112
- Paraphrasenproblem, 55
PARRY, 117
Parser, 64
passage retrieval, ↗ Textstellezugriff, 51
Passagenerschließung, ↗ Textstellezugriff
Passagenfilterung, 102
Performanz, 9, 32, 46
Persona, 139, 150, 161
Perspektivierung, 33
phonetische Suche, 54
Phrasensuche, 50, 55
Pipeline-Architektur, 175
Plan-basierter Ansatz, 127
Planung, 168, 174
Polysemie, 36, 54, 186
Portabilität, 32, 63, 77, 83
POS-Tagging, ↗ Wortart-Tagging
Position
 empirizistische, 10
 rationalistische, 8
Präparser, 64
- Präsentationsagent, 150
Präsupposition, 37
 falsche, 37, 88
Präsuppositionsverletzung, 37, 75, 121
precision, ↗ Genauigkeit
Produktivität, 32
Prosodie, 33
Protosynthex, 52, 53, 211
- QAS, ↗ question answering system
Qtarget, 105
QUD, ↗ question under discussion
Quellsprache, 187
query, 50
query expansion, 102
query relaxation, ↗ Anfragenrelaxierung
question answering, 51
question answering system, ↗ Frage-/Antwort-System
question generation, 178
question under discussion, 135
- Rückmeldung, 88
rationales Verhalten, 127
REA, 161
Realisierung, 174
recall, ↗ Vollständigkeit
Rechtschreibkorrektur, 14, 75
reciprocal rank, 111
Recognizing Textual Entailment Challenge, 113
Reiseassistent, 159
reliability, ↗ Verlässlichkeit
retention ratio, 205
rhetorische Relationen, 171
Robustheit, 9, 31, 79
RTE, ↗ Recognizing Textual Entailment Challenge
- Salienzwert, 207
Schema, 174
Schlüsselwort, 50, 102
Schlussfolgerung, 29, 33
SDS, ↗ spoken dialogue system
Segmentierung, 33
semantic search, 51

- Semantic Web, 52
 semantically tractable question, 88
 semantische Grammatiken, 77
 semantische Informationsverarbeitung, 25
 semantische Repräsentation, 106
 semantische Suche, 51
 semantischer Interpreter, 64
 semantisches Netz, 169
 semantisches Netzwerk, 53
 shallow processing, ↪ flache Verarbeitung
 shared plan, ↪ gemeinsamer Plan
 Shared Task Evaluation Conferences, 46, 176
 SHRDLU, 24, 30, 118
 Simultanübersetzung
 maschinelle, 14
 Simultandolmetschen
 maschinelles, 14
 Siri, 2, 13, 20, 113, 211
 Skalierbarkeit, 32, 140
 Skopusambiguität, 35
 SmartKom, 150, 159
 SmartWeb, 113, 161
 Soundex-Algorithmus, 54
 speech, ↪ gesprochene Sprache
 speech recognition, ↪ Spracherkennung
 speech synthesis, ↪ Sprachsynthese
 Speech-Technologie, 13
 spoken dialogue system, ↪ Sprachdialogsystem
 Sprachanwendung, 139
 Sprachapplikation, 7, 139
 Sprachdaten, 10
 Sprachdatenbanken, 16
 Sprachdialog-Plattform, 141
 Sprachdialogsyste, 7, 116, 139–147
 Spracherkennung, 7, 14, 41, 118, 139, 141
 Spracherkennungssoftware, 8
 Sprachgenerierung, 167–178
 sprachliche Abdeckung, 31, 75
 Sprachmodell, 39
 Sprachportal, 139
 Sprachproduktion, 169
 Sprachsynthese, 14, 41, 139, 141
 Sprachtechnologie, 7–20
 Sprachverarbeitung
 Computer-gestützte, ↪ Computer-gestützte Sprachverarbeitung
 maschinelle, ↪ maschinelle Sprachverarbeitung
 Sprachverstehen, 25
 statistical turn, 9, 28
 statistisches Verfahren, 8, 10, 28, 201
 STEC, ↪ SharedTask Evaluation Conferences
 Stemming, 54
 Stimmverifizierung, 141
 Stopwort, 50
 struktureller Ansatz, 125
 Subsumptionshierarchie, 87
 Suchanfrage, 50
 SUMMAC, ↪ Text Summarization Evaluation Conference
 SUMMARIST, 203
 Sustainability, ↪ Nachhaltigkeit
 symbolisches Verfahren, 10
 Synonymie, 55, 203
 System
 angewandtes, 119
 Domänen-spezifisches, 96
 Forschungs-, 8, 23, 27, 76, 77, 118, 128
 hybrides, ↪ hybride Systeme
 kommerzielles, 8, 27, 77
 Korpus-basiertes, 42
 multimodales, ↪ multimodales System
 natürlichsprachliches, ↪ natürlichsprachliches System
 sprachverarbeitendes, 8
 trainierbares, 63
 TAC, ↪ Text Analysis Conference
 template, 58, 69
 Template Generator, 64
 Terminologie
 computergestützte, 14
 Management, 195
 Testkorpus, 69
 Text Analysis Conference, 46, 61, 112, 113, 207
 text mining, 61
 Text Retrieval Conferences, 46, 95
 Text Summarization Evaluation Conference, 46

- Text-Technologie , 13
text-to-speech synthesis, 14, 141
Textaufteilung, 64
Textbearbeitung, 14
Textextraktion, 14
Textfilterung, 14
Textindexierung, 14
Textkategorisierung, 14
Textmining, 14, 51
Textschemata, 171
Textstellerschließung, ↪ Textstellenzugriff
Textstellenzugriff, 14
Textverständen, 27, 94, 113
Textzusammenfassung, 14, 113, 197–207
Textzusammenfassungssystem, 197
Thesaurus, 55
TIDES, ↪ Translingual Information Detection Extraction and Summarization, 60
tiefe Verarbeitung, 13, 41, 44
TIPSTER, 45, 60
Tokenisierung, 41
Trainingskorpus, 69
TRAINs, 128
Transducer, 65
translation memory, 195
translation workstation, 195
Translingual Information Detection Extraction and Summarization, 46
TREC, ↪ Text Retrieval Conferences
TREC-QA, 110
True Knowledge, 12, 109
TTS, ↪ text-to-speech synthesis
TUNA-Korpus, 177
turn-taking, 116
TZS, ↪ Textzusammenfassungssystem

Übersetzung
 computergestützte, ↪ Computer-gestützte Übersetzung
 maschinelle, 7
Übersetzungshilfen, 195
Übersetzungsproblem, 186
überwachtes Verfahren, 63
UIMA, 65
unüberwachtes Verfahren, 63

Unterspezifikation, 35, 187
Usability, ↪ Gebrauchstauglichkeit
user model, ↪ Benutzermodell

vage Anfragen, 75
Variabilität, 35
Verarbeitung
 flache, ↪ flache Verarbeitung
 interktionale, 162
 propositionale, 162
 tief, ↪ tiefe Verarbeitung
 wissensbasierte, 106
Verbmobil, 44, 125, 185, 194
Verbundworterkennung, 144
Verfahren
 Korpus-basiertes, ↪ Korpus-basiertes Verfahren
 machine learning, ↪ machine learning Verfahren
 numerisches, ↪ numerisches Verfahren
 statistisches, ↪ statistisches Verfahren
 symbolisches, ↪ symbolisches Verfahren
 wissensbasiertes, ↪ wissensbasiertes Verfahren
Verlässlichkeit, 32, 83, 88
Verstehen, 25
Viv, 214
Voice, 13
VOICE Awards, 142
Voice Portal, 139
Voice User Interface, 139, 145
VoiceXML, 140
Vollständigkeit (recall), 61
Volltextsuche, 50
Vorverarbeitung, 64
VUI, ↪ Voice User Interface

Watson, 13, 20, 49, 52, 110, 211
Weaver Memorandum, 184
WebQAS, 100
Weltmodell, 39
Weltwissen, 39
Whiteboard, 45
WIMP, 150
Wissen

INDEX

- assertionales, 83
 - extensionales, 83
 - intensionales, 83
 - terminologisches, 83
 - wissensbasiertes Verfahren, 10, 201
 - Wissensbasiertheit, 24
 - Wissensbasis, 38, 39
 - Wissensmanagement, 17
 - Wissensrepräsentation, 33, 168
 - in NLIDB, 83
 - Wissensverarbeitung, 38
 - Wizard-of-Oz-Studien, 125
 - WolframAlpha, 12, 19, 52, 80, 109, 113, 211
 - Wordnet, 55, 102, 104
 - Wortart-Tagging, 41, 64
- Ziel
- kommunikatives, 174
- Zielsprache, 187

Versionsgeschichte:

(minimale Veränderungen werden nicht aufgeführt)

- 1.17: Natürlichsprachliche Suche erweitert; Literaturupdate
- 1.18: Leichte Änderungen 3.4.1, Literaturupdate, Indexupdate, Änderung der Weblinks
- 1.19: Colored my Pdf; Erweiterungen/Änderungen in Kap. 3,4
- 1.20: Systeme farbig
- 2.0: Kapitel über Sprachgenerierung hinzugefügt; Evaluation, soweit vorhanden, als eigenen Abschnitt
- 2.01: kleine (Formatierungs-)Änderungen
- 2.02: kleine (Index-)Änderungen
- 2.03: Watson
- 2.1: Fazit hinzugefügt; Siri; kleine Änderungen
- 2.2: Epilog hinzugefügt