

Fehlerbehandlung in Mensch-Maschine-Dialogen

Von der Philosophisch-Historischen Fakultät der Universität Stuttgart
zur Erlangung der Würde eines Doktors
der Philosophie (Dr. phil.) genehmigte Abhandlung

Vorgelegt von
Petra Gieselmann
aus Bielefeld-Gadderbaum

Hauptberichter: Prof. Dr. Christian Rohrer
Mitberichter: Prof. Dr. Alexander Waibel

Tag der mündlichen Prüfung: 05.03.2007

Institut für maschinelle Sprachverarbeitung
Universität Stuttgart
2007

Danksagung

Diese Arbeit entstand an der Universität Karlsruhe am Institut für theoretische Informatik im Interactive Systems Lab von Prof. Waibel. Danken möchte ich meinem Hauptberichter Prof. Christian Rohrer, der diese interdisziplinäre Zusammenarbeit nicht nur über Fakultäts-, sondern sogar über Universitätsgrenzen hinweg erst ermöglichte. Danken möchte ich auch meinem Zweitberichter Prof. Alexander Waibel, der es mir als Computerlinguistin durch meine Arbeit an seinem Institut ermöglichte, detaillierten Einblick in die Informatik zu gewinnen und an interdisziplinären Projekten mitzuarbeiten.

Teile der vorliegenden Arbeit entstanden im Rahmen des Sonderforschungsbereiches 588 "Humanoide Roboter - Lernende und kooperierende multimodale Roboter" an der Universität Karlsruhe. In dem Zusammenhang möchte ich mich besonders bedanken bei den Projektpartnern, die es mir ermöglicht haben, Tests mit einem wirklichen Roboter durchzuführen und die Probleme in der Roboterentwicklung näher kennenzulernen.

Diese Arbeit wäre nicht ohne das hervorragende internationale Umfeld des Interactive Systems Lab der Universität Karlsruhe und die Unterstützung zahlreicher Kolleginnen und Kollegen möglich gewesen. Über die Jahre waren dies: Keni Bernhardt, Maria Danninger, Matthias Denecke, Hazim Ekenel, Christian Fügen, Hartwig Holzapfel, Thilo Köhler, Muntssin Kolss, Florian Kraft, Kenichi Kumatani, Kornel Laskowski, John McDonough, Florian Metze, Judith Mühl, Kai Nickel, Jürgen Reichert, Margit Rödder, Ivica Rogina, Thomas Schaaf, Rainer Stiefelhagen, Sebastian Stücker, Tanja Schultz, Alicia Tribble, Michael Voit, Matthias Wölfel. Besonderer Dank geht an Matthias Denecke und vor allem Hartwig Holzapfel für die Entwicklung eines Basis-Dialogmanagers, den ich in meiner Arbeit weiterentwickeln konnte.

Herzlich bedanke möchte ich mich auch bei all denen, die diese Arbeit Korrektur gelesen und mich mit ihren wertvollen Kommentaren bei der Fertigstellung unterstützt haben. Nennen möchte ich hier insbesondere: Regine Endsuleit, Judith Mühl, Dr. Thomas Schaaf, Prof. Prisca Stenneken und Daniel Wolf.

Unzähligen Benutzer in den verschiedensten Benutzertests möchte ich herzlich danken, auch wenn viele durch die Anonymität des Internets nicht zu nennen sind.

Besonderer Dank gebührt auch unserem Systemadministrator Dan Valsan, der sich stets um das korrekte Funktionieren der Hardware kümmerte. Und natürlich möchte ich mich auch bei unseren Sekretärinnen Annette Römer und Silke Dannenmaier bedanken, die sich um so viele doch so wichtige Kleinigkeiten kümmerten.

II

My special thanks goes to Prof. Mari Ostendorf with whom I enjoyed many insightful discussions during her stay at the university of Karlsruhe and during my stay at the University of Washington. This helped me a lot to focus on my research topic.

Vor allem möchte ich aber meinem Freund Daniel Wolf danken, der mich immer unterstützt und ermutigt hat, diese Arbeit weiterzuverfolgen, besonders auch in harten Zeiten. Danken möchte ich auch all meinen Freundinnen und Freunden, mit denen ich in zahlreichen Gesprächen Teile dieser Arbeit diskutieren konnte. Dank geht auch an meine Eltern, die mich auf meinem etwas ungewöhnlichen Weg durch Wirtschaft und Forschung unterstützt haben.

Inhaltsverzeichnis

I Einführende Darstellung der Problematik und Stand der Forschung	1
1 Einleitung	3
1.1 Einleitung und Motivation der Arbeit	3
1.2 Gegenstand und Ziel der Arbeit	6
1.3 Übersicht der Ausarbeitung	6
2 Stand der Forschung	9
2.1 Einleitung	9
2.2 Ansätze zur Modellierung von Dialogsystemen	9
2.2.1 Stand der Forschung im Bereich von Dialogsystemen	9
Symbolische Dialogansätze	9
Statistische Dialogansätze	11
Multimodale Dialogsysteme	11
2.2.2 Die TAPAS Dialogmanagementtools	12
Dialoggrammatik und Domänenmodell	13
Aufgabenmodell	13
Generierungsschablonen	14
Datenbanken	14
Dialogstrategie und Interaktionsmuster	15
2.3 Fehler in Mensch-Maschine-Dialogen	15
2.3.1 Einleitung	15
2.3.2 Vermeidung von Fehlern	16
Fehler im Bereich des Lexikons	16
Fehler im Bereich der Syntax	16
Fehler im Bereich von Semantik und Pragmatik	18
Zusammenfassung	18
2.3.3 Erkennung und Behandlung von Fehlern	18
Verbessern der Spracherkennung für Fehlersituationen	19
Erkennung von Fehlern durch die Benutzerreaktion	20
Erkennung von Fehlern mit Hilfe von Konfidenzmaßen	21
Generierungsstrategie: Auswirkungen der Systemantworten	21
Klärungsdialoge	23
Modelle zur Fehlerbehandlung	23
Ansätze zur Fehlerbehandlung in ARIADNE	25

2.3.4	Vergleich verschiedener Dialogsysteme und ihrer Leistungen	26
2.3.5	Zusammenfassung	27
2.4	Analyse zwischenmenschlicher Dialoge	27
2.4.1	Einleitung	27
2.4.2	Austausch von Informationen zwischen menschlichen Dialogpartnern	28
2.4.3	Klärungs- und Metadialoge zur Auflösung von Fehlern	29
2.4.4	Fehler in zwischenmenschlichen Dialogen mit ungleichen Dialogpartnern	31
2.4.5	Formen der Metakommunikation	33
2.4.6	Weltwissen und Kontextmanagement	33
2.5	Fehlermetriken	34
2.6	Zusammenfassung	35
II	Benutzerstudien zu Fehlerdialogen	37
3	Benutzertests in Dialogsystemen	39
3.1	Einleitung	39
3.2	Entwicklung eines Prototypen	40
3.3	Verschiedene Formen von Benutzertests	40
3.4	Zusammenfassung	42
4	Multimodale Benutzertests	45
4.1	Vorgehensweise	45
4.2	Ergebnisse des Experimentteils ohne Aufgaben	45
4.2.1	Generelle Ergebnisse	45
4.2.2	Detaillierte Fehleranalyse und -klassifikation	46
4.2.3	Benutzerverhalten nach dem Auftreten von Problemen und Fehlern	48
4.3	Ergebnisse für den aufgabenbasierten Teil	49
4.3.1	Generelle Ergebnisse	49
4.3.2	Detaillierte Fehleranalyse und -klassifikation	49
4.3.3	Benutzerverhalten nach dem Auftreten von Problemen und Fehlern	50
4.4	Zusammenfassung	51
5	Webbasierte Benutzertests	53
5.1	Einleitung	53
5.2	Vortest	53
5.2.1	Benutzerfragebogen	53
5.2.2	Auswertung der generellen Fragen	54
5.2.3	Auswertung der spezifischen Fragen	56
5.3	Aufgabenbasierter Benutzertest	59
5.3.1	Vorgehensweise	59
5.3.2	Ergebnisse	60

5.3.3	Detaillierte Fehleranalyse und -klassifikation	61
5.3.4	Benutzerverhalten nach dem Auftreten von Problemen und Fehlern	63
5.4	Benutzertest ohne Aufgaben	64
5.4.1	Vorgehensweise	64
5.4.2	Ergebnisse	64
5.4.3	Detaillierte Fehleranalyse und -klassifikation	65
5.4.4	Benutzerverhalten nach dem Auftreten von Problemen und Fehlern	66
5.4.5	Auswertung der Benutzerbefragung	67
5.5	Zusammenfassung	68
6	Fehlerklassifikation	71
6.1	Einleitung	71
6.2	Vergleich der verschiedenen Benutzertests	71
6.2.1	Generelle Unterschiede und Gemeinsamkeiten in den Be- nutzertests	71
6.2.2	Auftreten der Fehlertypen in den Benutzertests	73
6.2.3	Benutzerverhalten bei Problemen und Fehlern	75
6.3	Zusammenfassung	75
III	Strategien zur Fehlervermeidung und -behebung	77
7	Evaluation der Kommunikationsstrategien	79
7.1	Einleitung	79
7.2	Auswirkungen der Benutzerstrategie	79
7.2.1	Einleitung	79
7.2.2	Benutzertest	80
7.2.3	Diskussion der Ergebnisse	81
Objektive Maße	81	
Auswirkungen der Dialogstrategie auf den kommunika- tiven Erfolg	81	
Zusammenhang von subjektiven und objektiven Maßen für den Erfolg	83	
7.2.4	Auswirkungen der generellen Konversationsstrategie	84
7.3	Auswirkungen der Roboterstrategie	85
7.3.1	Einleitung	85
7.3.2	Konfidenzmaße	86
Einleitung	86	
Vorgehensweise	87	
Ergebnisse	87	
7.3.3	Hold-Strategie	89
Einleitung	89	
Spracherkennungsfehler	90	
Neuer Subdialog	90	
Evaluation	90	

7.4	Zusammenfassung	91
8	Auswirkungen einer größeren Grammatik	93
8.1	Einleitung	93
8.2	Datensammlung und Erweiterung der Grammatik	93
8.3	Ergebnisse	94
8.3.1	Vergleich der Parsing- und Turnfehlerraten	94
8.3.2	Vergleich der Anteile verschiedener Fehlerklassen	94
8.4	Zusammenfassung	97
9	Dynamische Erweiterung ling. Ressourcen	99
9.1	Einleitung	99
9.2	Verwandte Arbeiten	100
9.3	Verwendete Mechanismen	100
9.3.1	Vererbungsmechanismen	100
9.3.2	Erweiterte Datenbankfähigkeiten	101
9.3.3	Auswirkungen auf den Spracherkenner	102
9.4	Architektur der Rezeptapplikation	102
9.4.1	Einleitung	102
9.4.2	Die Rezeptdatenbank	105
9.4.3	Finden von Rezepten mit Hilfe des Namens bzw. der Zutaten	106
9.5	Experimente	106
9.5.1	Einleitung	106
9.5.2	Textbasierter Benutzertest	107
9.5.3	Gesprochensprachlicher Benutzertest	108
9.6	Zusammenfassung	110
10	Integration des Internets als Wissensbasis	111
10.1	Einleitung	111
10.2	Verwandte Arbeiten	111
10.3	Das QA-System EPHYRA	112
10.3.1	Wissensannotierung	113
10.3.2	Musterlernen	113
10.3.3	Umformulierung der Benutzeranfragen	114
10.3.4	Antwortgenerierung	115
10.4	Integration von QA-System und Dialogmanagement	115
10.4.1	Architektur von EPHYRA	115
10.4.2	Das ONE4ALL-System	115
10.4.3	Erstellung einer QA-Grammatik mit Hilfe der Ressourcen aus Ephyra	118
10.5	Evaluation	119
10.5.1	Einleitung	119
10.5.2	Evaluation des QA-Systems EPHYRA	120
10.5.3	Evaluation des Gesamtsystems	120
10.6	Zusammenfassung	122

11 Auflösen von Ellipsen und Anaphern	123
11.1 Einleitung	123
11.2 Integration deiktischer Pronomina	124
11.2.1 Einleitung	124
11.2.2 Vorteile multimodaler Interaktion	124
11.2.3 Der Gestenerkenner	125
11.2.4 Multimodale Integration	126
11.3 Auflösung von Personalpronomina	128
11.3.1 Einleitung	128
11.3.2 Verwandte Arbeiten	128
Mechanismen zur Referenzauflösung	128
Referenzauflösung in Dialogsystemen	129
11.3.3 Kontextmodellierung zur Referenzauflösung	130
Einleitung	130
Das Kontextmodell	130
Benutzerstudie zur Auflösung von Personalpronomina	131
11.4 Auflösung elliptischer Äußerungen	132
11.5 Zusammenfassung	133
12 Auflösung komplexer Äußerungen	135
12.1 Einleitung	135
12.2 Analyse der in den Benutzertests vorkommenden verknüpften Äußerungen	136
12.3 Auflösung verknüpfter Nominalphrasen	136
12.4 Auflösung verknüpfter Verbalphrasen	137
12.5 Zusammenfassung	138
13 Klärungsdialoge und Metakommunikation	141
13.1 Einleitung	141
13.2 Zwischenmenschl. Reparaturdialoge	141
13.2.1 Einleitung	141
13.2.2 Analyse der Fehlerdialoge	142
Ergebnisse	142
Indikatoren für Fehler	143
Strategien zur Fehlerbehebung	144
Detaillierte Analyse des Beispieldialogs	147
13.2.3 Zusammenfassung	147
13.3 Vom Benutzer initiierte Metakommunikation	148
13.3.1 Einleitung	148
13.3.2 Nachfragen vom Benutzer	149
13.3.3 Korrektur von Benutzeräußerungen	151
13.3.4 Benutzeräußerungen zum Austesten des Roboters	153
13.3.5 Diskussion und Evaluation der vom Benutzer angestoßenen Metakommunikation	154
13.4 Vom Roboter initiierte Metakommunikation	154
13.4.1 Einleitung	154
13.4.2 Erkennen problematischer Situationen	155

13.4.3	Finite-State-Automat	159
13.4.4	Benutzerstudie	163
13.5	Zusammenfassung	164
IV	Evaluation und Fazit	167
14	Diskussion und Evaluation	169
14.1	Einleitung	169
14.2	Evaluation	169
14.2.1	Einleitung	169
14.2.2	Benutzerstudie	170
14.3	Übertragbarkeit auf andere Domänen	174
14.4	Zusammenfassung	175
15	Fazit	177
15.1	Beiträge	177
15.2	Zukünftige Arbeiten	179
Abstract		181
Literaturverzeichnis		191
V	Appendix	211
40	Fragen aus dem QA TREC Korpus 2004	213

Abbildungsverzeichnis

2.1	Der Dialogmanager und seine Ressourcen	13
2.2	Dialogzustände und ihre Übergänge	14
3.1	In den Benutzertests eingesetzte Plattformen und Roboter	41
3.2	Das webbasierte Interface für den humanoiden Roboter	42
5.1	Benutzerfragebogen für den Vortest im Internet	54
5.2	Benutzerbefragung: Erste Aufgabe für den Roboter	56
5.3	Benutzerbefragung: Rezepte, die der Roboter kennen sollte	57
5.4	Benutzerbefragung: Instruktionen für die Vorbereitung eines Abends mit Freunden	57
5.5	Benutzerbefragung: Erkennungsrate der Benutzeräußerungen für verschiedene Aufgaben	58
5.6	In den aufgabenbasierten Tests verwendete Aufgaben	59
6.1	Gegenüberstellung der auftretenden Fehlertypen in den durchgeführten Experimenten	73
6.2	Gegenüberstellung der Fehlertypen im Bereich der neuen Konzepte in den durchgeführten Experimenten	74
8.1	Vergleich der Parsingraten in der 1. und der 2. Version der Roboterapplikation	95
8.2	Vergleich der Turnfehlerraten in der 1. und der 2. Version der Roboterapplikation	95
8.3	Vergleich der Anteile der Fehlerklassen in der 1. und der 2. Version der Roboterapplikation	96
8.4	Vergleich der Anteile der Fehlerklassen innerhalb der neuen Konzepte in der 1. und der 2. Version der Roboterapplikation	96
9.1	Beispiel für eine Datenbankschnittstellen-Definition für Rezepte .	101
9.2	Architektur der Rezeptapplikation	102
9.3	Entwicklung der Turnfehlerraten bei unterschiedlichen Vokabulargrößen	109
10.1	Architektur des QA-Systems EPHYRA	113
10.2	Die Architektur des Gesamtsystems ONE4ALL	116
11.1	Beispiel für die multimodale Integration von Sprache und Gesten	126

11.2	Semantische Repräsentation des Satzes “Put it on the table.” (“Leg es auf den Tisch.”)	130
11.3	Generischer Move zum Triggern von Nachfragen bei sehr kurzen Benutzeräußerungen	132
12.1	Definition eines generischen Sprechakts mit mehreren Objekten und der dazugehörigen Objektliste	137
13.1	Klassifikation der Äußerungen auf Metakommunikationsebene und ihre Anteile in den Benutzertests	149
13.2	Generische Regel zur Korrektur von Benutzeräußerungen	153
13.3	Finite-State-Automat mit vier Zuständen	159
13.4	Hierarchische Struktur der Fähigkeiten des Roboters	163
13.5	Auswertung des Fragebogens für Version A vs. Version B	165
14.1	Gegenüberstellung der Ergebnisse: Version F vs. Baselineversion	172
14.2	Auswertung des Fragebogens für die Baselineversion vs. Version F	172
14.3	Gegenüberstellung der Ergebnisse: Version F vs. Internetttest	173

Tabellenverzeichnis

2.1	Einige Dialogsysteme und ihre Rate an Verständnisfehlern	26
2.2	Fehlerbehebungsstrategien in der zwischenmenschlichen Kommunikation	30
4.1	Ergebnisse des Benutzertests ohne vorgegebene Aufgaben	46
4.2	Fehleranalyse der aufgetretenen Fehler im Experimentteil ohne vorgegebene Aufgaben	47
4.3	Ergebnisse des Benutzertests mit Aufgaben	49
4.4	Fehleranalyse der aufgetretenen Fehler im aufgabenbasierten Teil	50
4.5	Vergleich der aufgetretenen Fehlertypen im Teil mit vs. ohne Aufgaben	51
5.1	Benutzerbefragung: Gut vs. schlecht für den Roboter geeignete Aufgaben	55
5.2	Ergebnisse des aufgabenbasierten Benutzertests	60
5.3	Detaillierte Fehleranalyse des aufgabenbasierten Benutzertests	61
5.4	Ergebnisse des Benutzertests ohne Aufgaben	65
5.5	Fehleranalyse der aufgetretenen Fehler im Experiment ohne Aufgaben	66
5.7	Benutzerbefragung zu dem Test ohne Aufgaben	68
5.8	Vergleich der aufgetretenen Fehlertypen in den Experimenten mit vs. ohne Aufgaben	69
6.1	Vergleich der Turnfehlerrate für die durchgeführten Benutzertests	71
6.2	Vergleich der auftretenden Fehlertypen in den durchgeführten Experimenten	72
7.1	Objektive Maße für den Dialogerfolg	82
7.2	Subjektive Maße für den Dialogerfolg	83
7.3	Korrelationen zwischen den verschiedenen subjektiven und objektiven Dialogmaßen	83
7.4	Korrelationskoeffizient zwischen Äußerungslänge und Erfolg	84
7.5	Detektion falsch erkannter Äußerungen	88
7.6	Gegenüberstellung der Raten an richtigen Zuordnungen zu den Klassen korrekt vs. falsch erkannte Worte	88
7.8	Rate an abgebrochenen bzw. finalisierten Dialogzielen mit der Hold-Strategie	92

8.1	Verschiedene Versionen des Prototyps und ihre Fähigkeiten	94
8.2	Vergleich der Raten im Bereich der neuen Konzepte in der 1. und der 2. Version der Roboterapplikation	97
9.3	Benutzerstudie zur Abdeckung des erweiterten Systems mit zu- sätzlichen Rezepten	107
9.4	Ergebnisse der Benutzerstudie mit dem erweiterten System mit zusätzlichen Rezepten	108
9.5	Auswertung der Benutzerbefragung	109
10.1	Auswertung der TREC8-Fragen für die benutzten QA-Techniken	121
10.2	Vergleich des grammatikbasierten und des statistischen Sprach- erkennters	121
11.1	Anzahl an Turns mit deiktischen Pronomina und Zeigegesten in einem Experiment mit dem Haushaltsroboter	127
11.2	Anzahl an Turns mit Personalpronomina in einem Experiment mit dem Haushaltsroboter	131
12.1	Raten der Verknüpfungen von Nominal- vs. Verbalphrasen in den Benutzertests	136
13.1	Indikatoren für Fehler in der Mensch-Roboter-Kommunikation . .	143
13.2	Fehlerbehebungsstrategien in der Mensch-Roboter-Kommunikation	144
13.4	Generische vs. applikationsspezifische metakommunikative Äuße- rungen	150
13.5	Aufstellung aller neuen Sprechakte zur Metakommunikation . . .	154
13.6	Vergleich der Ergebnisse mit vs. ohne Metakommunikation für die zweite Version der Roboterapplikation	155
13.7	Auswertung der Benutzerstudie für die Version A vs. Version B .	164
14.1	Auswertung des Benutzertests für die neue Version F mit Fehler- behandlung	171
15.1	Turn Error Rates Within Different Testing Conditions.	184
15.2	Comparison of the Rates of the Error Classes	185
15.3	User Study Comparing the Baseline Version with Version F . . .	190

Teil I

Einführende Darstellung der Problematik und Stand der Forschung

Kapitel 1

Einleitung

1.1 Einleitung und Motivation der Arbeit

Seit es Computer gibt, existiert auch der Wunsch des Menschen, mit ihnen reden zu können wie mit einem anderen Menschen. Eines der berühmtesten Beispiele dafür ist sicherlich Eliza (Weizenbaum 1996), ein Computerprogramm, das einen Psychologen simuliert, mit dem der Benutzer ein Therapiegespräch führen kann. In vielen Science-Fiction-Filmen finden sich auch immer wieder Beispiele für solche sprechenden Maschinen, wie beispielsweise HAL in “2001: Odysee im Welt- raum” (Clark 2001) oder auch der Computer auf dem “Raumschiff Enterprise”. So reichen erste Dialogsysteme bereits zurück bis in die Anfänge der künstlichen Intelligenz in den fünfziger Jahren (cf. auch McTear 2002; Bolt 1980).

Dennoch hatten diese Dialogsysteme bis vor wenigen Jahren noch mit so vielen Problemen zu kämpfen, dass sie kaum für einen praktischen Einsatz ge- eignet waren. Erst in letzter Zeit ist es durch die stetigen Verbesserungen im Bereich von Spracherkennung und Sprachverstehen und das Aufkommen von immer schnelleren und mächtigeren Rechnern möglich geworden, solche Systeme für den realen Einsatz zu bauen. Nach wie vor gibt es aber noch eine ganze Reihe ungelöster Probleme, die zum einen auf die Komplexität natürlicher Spra- che und zum anderen auf den immensen Fundus an vernetztem Weltwissen und Kontextbeziehungen, über den Menschen verfügen, zurückzuführen sind.

Eine der bislang größten Herausforderungen liegt darin, ein solches Dia- logsystem auch für den realen Einsatz unter Alltagsbedingungen zu entwerfen. Bisher fehlt den Systemen dafür noch die nötige Fehlerrobustheit, um in Situa- tionen, in denen das System etwas falsch verstanden hat und es zu Problemen kommt, angemessen reagieren zu können. In dieser Arbeit geht es genau um solche Fehler im Dialog, wie sie vermieden und während des laufenden Dialogs wieder behoben werden können, wenn sie nicht vorher zu vermeiden waren.

Der folgende Beispieldialog mit einem Haushaltsroboter, der seinen Benut- zern in der Küche hilft, verdeutlicht Fehler und ihre Auswirkungen, wie sie bis- her im Dialog auftreten (Da der Dialog im Original auf Englisch war, findet sich die deutsche Übersetzung zum besseren Verständnis jeweils in Klammern. Fer- ner sind neben der Benutzeräußerung auch die vom Spracherkenner erkannten Hypothesen der Äußerungen in eckigen Klammern in Kursivschrift angegeben.):

1. Benutzer: Please set the table. [*please set the table*]
(Bitte decke den Tisch. [*bitte decke den Tisch*])
2. Roboter: For how many persons do you want me to set the table?
(Für wie viele Personen soll ich den Tisch decken?)
3. Benutzer: We have two persons. [*I wish to open those mugs*]
(Wir sind zu zweit.
[*ich würde gerne diese Becher öffnen*])
4. Roboter: <Keine Reaktion>
5. Benutzer: Two persons please. [*open those lamps please*]
(Zwei Personen bitte. [*öffne diese Lampen bitte*])
6. Roboter: For how many persons do you want me to set the table?
(Für wie viele Personen soll ich den Tisch decken?)
7. Benutzer: Two. [*Hello*]
(Zwei. [*Hallo*])
8. Roboter: Hello! My name is Robbi. I am a little robot who can
help you in the kitchen. What do you want me to do?
(Hallo! Ich heiÙe Robbi. Ich bin ein kleiner Roboter, der
dir in der Küche helfen kann. Was kann ich für dich tun?)
9. Benutzer: (lacht) [*not*]
([*nicht*])
10. Roboter: How can I help you?
(Wie kann ich dir helfen?)
11. Benutzer: Please set the table. [*please set the table*]
(Bitte decke den Tisch. [*bitte decke den Tisch*])
12. Roboter: For how many persons do you want me to set the table?
(Für wie viele Personen soll ich den Tisch decken?)
13. Benutzer: +smack+ Two. [*for two*]
(Zwei. [*für zwei*])
14. Roboter: Which kind of glasses do you want me to put on
the table? Wine or long drink glasses?
(Welche Gläser soll ich auf den Tisch stellen?
Weingläser oder Wassergläser?)
15. Benutzer: +smack+ Longdrink. [*longdrink*]
(Wassergläser. [*Wassergläser*])
16. Roboter: <Keine Reaktion>
17. Benutzer: +smack+ Longdrink. [*something to eat*]
(Wassergläser. [*etwas zu essen*])
18. Roboter: Which kind of glasses do you want me to put on
the table? Wine or long drink glasses?
(Welche Gläser soll ich auf den Tisch stellen?
Weingläser oder Wassergläser?)
19. Benutzer: Wine glasses. [*wine glasses*]
(Weingläser. [*Weingläser*])
20. Roboter: Do you want a meal with or without dessert?
(Soll das Essen mit oder ohne Nachtisch sein?)
21. Benutzer: No. [*no*]
(Nein. [*nein*])

22. Roboter: Do you want a meal with or without dessert?
(Soll das Essen mit oder ohne Nachtisch sein?)
23. Benutzer: With. *[with]*
(Mit. *[mit]*)
24. Roboter: How can I help you?
(Wie kann ich dir helfen?)

Es lassen sich zwei Arten von Fehlern unterscheiden: *Nicht-Verstehen* vs. *Missverstehen*. Beim Nicht-Verstehen ist das System nicht in der Lage, irgendeine Information aus der Äußerung des Benutzers zu ziehen. Dies kann beispielsweise dadurch hervorgerufen werden, dass die Grammatik die Benutzeräußerung nicht abdeckt und sie daher nicht geparkt werden kann, wie in Zeile 23 des oben aufgeführten Beispieldialogs mit unserem Haushaltsroboter zu sehen ist, wo “with” (“mit”) als einzelnes Wort nicht zu einem vollständigen Parse führt. Ebenso kann es aber auch auf der pragmatischen Ebene zum Nicht-Verstehen kommen, wenn die Äußerung des Benutzers nicht zum bereits vorhandenen Diskurs passt und deshalb nicht integriert werden kann. Dies zeigt sich im Beispieldialog in der Zeile 3, in der die falsch verstandene Benutzeräußerung “*i wish to open those mugs*” (“*ich würde gerne diese Becher öffnen*”) nicht zur vorhandenen Information im Diskurs passt, weshalb das Dialogsystem daraufhin keine Reaktion zeigt. Beim Missverstehen hingegen liegt das Problem darin, dass die Äußerung zwar geparkt und die semantische Interpretation in den Diskurs integriert werden kann, diese aber nicht der Intention des Benutzers entspricht, sondern er etwas anderes gemeint hat. Dies findet sich beispielsweise in Zeile 7, wo der Spracherkennung “hello” (“hallo”) verstanden hat und daraufhin den Benutzer begrüßt, der Benutzer aber eigentlich dem System mitteilen wollte, dass er den Tisch für zwei Personen decken lassen möchte.

Im Folgenden geht es daher nicht nur um Spracherkennungsfehler, die auch schon von anderen Forschern im Detail untersucht worden sind (Choularton, Dale 2004; Filisko, Seneff 2004; Gorrell 2003), sondern vor allem um Verständnisfehler, bei denen das Dialogsystem die semantische Auflösung nicht korrekt erstellen konnte oder die Benutzereingaben nicht richtig in den vorhandenen Diskurs eingebaut wurden. Solche Fehler führen zu gravierenden Problemen im Dialog, die sich häufig nur sehr aufwendig wieder beheben lassen, wie an dem oben erläuterten Beispiel deutlich wird. Hier versucht der Benutzer immer wieder, den Roboter den Tisch für zwei Personen mit Longdrink-Gläsern und Löffeln für einen Nachtisch decken zu lassen. Doch da es im Laufe des Dialogs zu zahlreichen Fehlern kommt, muss der Benutzer immer wieder von vorne beginnen, was dazu führt, dass er schließlich genervt von dem System aufgibt. In dieser Arbeit werden Methoden entwickelt, um Fehler soweit als möglich zu vermeiden und um darüberhinaus die Fehlerbehebung zu vereinfachen, da es nicht möglich ist, alle Fehler schon im Vorfeld zu vermeiden.

Wie auch schon Allen’s praktische Dialoghypothese (Allen et al. 2000a) herausstellt, ist es nach wie vor nicht möglich, ein generelles Dialogsystem, das in allen Situationen adäquat mit dem Benutzer kommunizieren kann, zu entwi-

ckeln. Daher wird in dieser Arbeit als Beispiel für ein Dialogsystem die Dialogkomponente eines humanoiden Roboters herangezogen, der dem Menschen im Haushalt helfen soll ¹. Die Vielfältigkeit der Mensch-Roboter-Kommunikation führt dabei zu einer sehr interessanten und für die Dialogkomponente herausfordernden Form der Mensch-Maschine-Interaktion. Um die Übertragbarkeit der entwickelten Mechanismen auf andere Domänen zu gewährleisten, wird ihre Generalisierbarkeit im Detail überprüft.

Dieser Roboter kann verschiedene Tätigkeiten in der Küche verrichten, wie z.B. etwas holen, etwas an- bzw. ausschalten, dem Benutzer Rezepte mitteilen, im Kühlschrank nachsehen, was noch da ist, etwas öffnen bzw. schließen, usw. (Gieselmann et al. 2003; Stiefelhagen et al. 2004). Der Roboter ist vor allem für ältere Menschen gedacht, die nicht mehr mobil genug sind für verschiedene Haushaltstätigkeiten, und soll daher auch einfach zu bedienen sein und ein humanoides Verhalten aufweisen. Das bedeutet, dass er - genau wie ein menschlicher Diener - in der Lage sein soll, sprachliche Anweisungen von einem Benutzer entgegenzunehmen und diese in einem gegebenen Kontext richtig auszuführen. Gerade für einen solchen Roboter ist es daher für die Benutzerakzeptanz und seine Alltagstauglichkeit wichtig, dass er wenig Fehler in der Kommunikation macht und sich diese auch leicht wieder korrigieren lassen.

Da Fehlerbehandlung in allen natürlichsprachlichen Anwendungen, in denen der Mensch mit einer Maschine kommuniziert, wie beispielsweise maschinelle Übersetzung oder auch Frage-Antwort-Systeme, eine wichtige Rolle spielt, müssen die entwickelten Algorithmen allgemein auf gesprochensprachliche Eingaben übertragbar sein. In einer abschließenden Analyse wird daher die Generalisierbarkeit und Übertragbarkeit auf andere Domänen und Systeme gezeigt.

1.2 Gegenstand und Ziel der Arbeit

Der Gegenstand dieser Arbeit ist eine datengetriebene Analyse der Fehler, die in der Mensch-Roboter-Kommunikation auftreten mit dem Ziel, diese möglichst im Vorfeld zu vermeiden. Es wird eine Fehlerklassifikation aufgestellt und es werden Methoden für die Vermeidung der verschiedenen Fehlerklassen entwickelt und evaluiert. Darüberhinaus werden auch generische Methoden zur Fehlerbehebung für die Fälle implementiert, die nicht vorher vermieden werden konnten, ebenfalls mit Hilfe datengetriebener Analysen. Damit soll es ermöglicht werden, Dialogsysteme über die Laborumgebung hinaus in realen Situationen einsetzen zu können. Dies wird am Beispiel eines Haushaltsroboters diskutiert und evaluiert.

1.3 Übersicht der Ausarbeitung

Diese Ausarbeitung gliedert sich in vier Teile: Der erste Teil beschäftigt sich mit dem Stand der Forschung in den Bereichen, die hier eine Rolle spielen. Dazu werden verschiedene Ansätze für Mensch-Maschine-Dialogsysteme beleuchtet.

¹Teile dieser Arbeit entstanden im Rahmen des SFB 588 "Humanoide und kooperierende Roboter", der von der DFG gefördert wird.

Im Anschluss wird die menschliche Informationsverarbeitung im Dialogbereich erläutert. Dabei geht es auch um Fehlerdialoge in zwischenmenschlichen Dialogen, die hier als Vorbild für Mensch-Roboter-Dialoge dienen.

Der zweite Teil beschäftigt sich mit den durchgeführten Benutzertests und Datensammlungen und der Klassifikation von Fehlern im Dialog, die die Grundlage für die folgenden Arbeiten zur Fehlervermeidung und -behebung bilden. Zunächst erfolgt eine detaillierte Analyse von Fehlern, die bei der Mensch-Roboter-Interaktion auftreten können. Dazu werden verschiedene aufeinander aufbauende Benutzerstudien und Datensammlungen, bei denen der Roboter dem Menschen im Haushalt zur Hand geht und einfache Tätigkeiten verrichtet, durchgeführt, um eine große Menge an möglichst realistischen Daten gewinnen zu können, die nicht nur unter Laborbedingungen entstanden sind.

Im dritten Teil werden verschiedene Methoden zur Fehlervermeidung und -behebung vorgestellt. Zur Fehlervermeidung werden zusätzliche Wissensquellen in den Dialogmanager integriert. Außerdem werden Mechanismen zur Anapherresolution, Kontextmodellierung, Auflösung von Ellipsen, multimodalen Fusion und zum Umgang mit komplexen, zusammengesetzten Äußerungen entwickelt und evaluiert. Zur Fehlerbehebung werden verschiedene Strategien für effektive Klärungsfragen untersucht. Metakommunikation (cf. Bußmann 1990: Metakommunikation als Kommunikation über Kommunikation), wie sie in den durchgeführten Benutzertests vorkommt, wird analysiert, um eine effektivere Kommunikation gewährleisten zu können. Außerdem wird ein Mechanismus entwickelt, der es dem Roboter erlaubt, problematische Situationen zu erkennen und diese selbst durch Metakommunikation aufzulösen.

Im vierten Teil werden die entwickelten Methoden anhand eines abschließenden Benutzertests evaluiert. Dabei geht es darum, das System mit allen entwickelten Mechanismen zur Fehlerbehandlung zu testen und es mit dem Basissystem zu vergleichen. Das besondere Augenmerk liegt hier auf der Übertragbarkeit der entwickelten Mechanismen auf andere Domänen und Systeme. Danach folgt das Fazit der gesamten Arbeit und eine Diskussion der zukünftigen Arbeiten im Hinblick auf mögliche Erweiterungen dieses Systems.

Kapitel 2

Stand der Forschung

2.1 Einleitung

Zunächst werden existierende maschinelle Dialogsysteme dargestellt, wobei die verschiedenen Ansätze erläutert und auf ihre spezifischen Vor- bzw. Nachteile eingegangen wird. Danach folgt die Vorstellung der hier verwendeten Dialogmanagementtools TAPAS und ihrer spezifischen Eigenschaften (Holzapfel 2005). Der zweite Abschnitt dieses Kapitels beschäftigt sich mit Fehlern in Mensch-Maschine-Systemen, wie diese zustande kommen, welche Möglichkeiten es zu ihrer Vermeidung gibt, und wie sie sich beheben lassen. Ferner geht es hier um eine Einordnung vorhandener Dialogsysteme und ihrer Leistungen und Grenzen, um den Wert einer zusätzlichen Fehlerbehandlungskomponente einschätzen zu können. Da sich die Erwartungen der Benutzer an ein Dialogsystem am zwischenmenschlichen Dialog orientieren, werden in diesem Kapitel außerdem zwischenmenschliche Dialoge analysiert. Die zentrale Frage ist, inwieweit Fehler in zwischenmenschlichen Dialogen und deren Auflösung Hinweise geben können auf die sinnvolle Verarbeitung von Fehlern in Mensch-Maschine-Systemen. Den Abschluss dieses Kapitels bildet ein Abschnitt über Evaluationsmetriken und Verfahren, wie sich Fehler in einem Dialogsystem objektiv messen lassen.

2.2 Ansätze zur Modellierung von Dialogsystemen

2.2.1 Stand der Forschung im Bereich von Dialogsystemen

Generell lassen sich symbolische und statistische Dialogansätze unterscheiden, wobei die symbolischen noch weiter unterteilt werden können in finite-state, rahmenbasiert vs. informationsbasiert.

Symbolische Dialogansätze

Bei finite-state Systemen wird der Dialog über eine begrenzte Anzahl von Dialogzuständen und deren Übergängen modelliert. D.h., es gibt eine vordefinierte Anzahl von Schritten, die der Benutzer auf seinem Weg zum Dialogziel durchläuft. Dadurch sind die Äußerungen des Benutzers sehr stark vorgegeben, was

den Vorteil hat, dass Vokabular und Grammatik gut im Vorfeld definiert werden können. Dies erleichtert die Arbeit für die Sprachverstehenskomponente. Gleichzeitig ist es aber auch sehr nachteilig für den Benutzer, sich an ein so starres, vorgegebenes System halten zu müssen. Vor allem, wenn er im Nachhinein Änderungen an den eingegebenen Daten machen möchte, beispielsweise weil etwas falsch verstanden wurde, so ist dies sehr aufwändig und kompliziert. Dies zeigt sich auch an “Nuance”, einem automatischen Kontoführungssystem, das auf dem finite-state Ansatz beruht (McTear 2002). Zusammenfassend lässt sich sagen, dass der finite-state Ansatz vor allem durch seine Einfachheit besticht und sich für gut strukturierte Aufgaben eignet. Demgegenüber versagt er bei nicht so strukturierten Aufgaben, wie z.B. Flugbuchungen, wo es schwer ist, die Reihenfolge der einzelnen Subdialoge vorauszusagen, aufgrund seiner mangelnden Flexibilität.

Eine flexiblere und natürlichere Dialogführung bieten demgegenüber rahmenbasierte Systeme, bei denen der Ablauf des Dialogs nicht vorgegeben ist. Hierbei stellt das System dem Benutzer Fragen, deren Antworten vom System in freie Plätze (cf. englisch: Slots) einer Vorlage eingefügt werden, wobei die Reihenfolge, in der der Benutzer dem System Informationen zur Verfügung stellt, beliebig ist, und sogar mehrere Informationen gleichzeitig gegeben werden können. Allerdings führt dies zu höheren Anforderungen an die Sprachverstehenskomponente. Daher wird in rahmenbasierten Dialogsystemen meist mit “Konzeptspotting” gearbeitet; d.h., die Eingabe wird nach bestimmten Konzepten durchsucht und alles, was in kein Konzept passt, wird ignoriert. Phillips Zuginformationssystem arbeitet beispielsweise in dieser ereignisgetriebenen Art und Weise (Aust et al. 1995). Statt einfacher Rahmen werden heutzutage meist komplexe Repräsentationsformen bevorzugt, wie beispielsweise E-Formulare, Schemata oder Typenhierarchien (Rudnicky et al. 1999; Goddeau et al. 1996; Dencke 2002a), um die Struktur der vom Benutzer erhaltenen Informationen zu modellieren und festzulegen, welche Aktionen das Dialogsystem vornimmt, um weitere Informationen zu bekommen. Doch auch die Flexibilität von rahmenbasierten Systemen stößt an ihre Grenzen, wenn es um komplexe Aufgaben geht, wie z.B. Planungsaufgaben oder andere gemeinschaftliche Interaktionsformen oder wenn sich der Zustand der Umwelt während des Dialogs ändert, so dass nicht alle möglichen Ereignisse vorauszusehen sind, oder wenn die Benutzer zu stark variieren in ihrem Wissen über die Aufgabe, so dass zu viele Antworten vorherzusehen wären. Solche komplexen Aufgaben lassen sich nur von informationsbasierten Systemen lösen.

Informationsbasierte (vielfach auch *agentenbasiert* genannte) Systeme gehen von einem Ansatz der künstlichen Intelligenz aus, wobei die Kommunikation als Interaktion zwischen zwei Agenten angesehen wird. Hierbei geht es um Problemlösen mit Hilfe einer komplexen Kommunikation von Benutzer, System und der zugrundeliegenden Applikation. Das Dialogmodell zieht auch den Kontext mit Hilfe von Kontextmanagementkomponenten in Betracht, was dazu führt, dass der Dialog sich dynamisch weiterentwickeln kann mit aufeinanderaufbauenden Stufen. Außerdem haben solche Systeme auch den Vorteil, dass sie mit gemischter Initiative arbeiten im Gegensatz zu den anderen Ansätzen, wo die Initiative immer vom System ausgeht. D.h., der Benutzer kann hier auch selbst

die Initiative ergreifen und sogar etwas einbringen, was nicht vom System gefragt wurde, was natürlich für die Sprachverstehenskomponente eine größere Herausforderung bedeutet, weil der Benutzer nun nicht restringierte natürliche Sprache eingeben kann. Circuit-Fix-It-Shop, ein System, das Benutzern bei Problemen in Stromkreisen hilft (Smith, Hipp 1994), arbeitet beispielsweise mit diesem informationsbasierten Ansatz. Der entscheidende Unterschied zu den anderen beiden Ansätzen liegt darin, dass hier dynamisch gemeinsam mit dem Benutzer eine Lösung entwickelt wird. Inwieweit die Aufgabe schon erledigt ist, wird mit Hilfe von merkmalsbasierten Formalismen, die auf der mit Phrasenstrukturregeln oder auch Unifikationsgrammatiken geparsten Eingabe aufsetzen, ermittelt.

Statistische Dialogansätze

Demgegenüber stehen die statistischen Dialogsysteme, die in den letzten Jahren von verschiedenen Forschungsgruppen, wie beispielsweise AT & T, entwickelt worden sind. Das von AT&T entwickelte Auskunftssystem “How May I Help You” arbeitet dabei mit bedeutungstragenden Worten und Phrasen (cf. engl. salient expressions) für eine bestimmte Aufgabe (Gorin et al. 2002), die durch informationstheoretische Maße bestimmt werden. Der Vorteil solcher statistischen Systeme liegt darin, dass sie wirklich das abbilden, was die Benutzer sagen, und nicht das, was sich der Entwickler der Grammatik ausgedacht hat. Über die robustere Verarbeitung von Daten hinaus ist der Vorteil von statistischen Systemen, dass zusätzliches Wissen gelernt werden kann, wodurch natürlich weniger Expertenwissen zur Erstellung des Systems nötig ist, was es weniger aufwendig macht. Allerdings benötigen die Systeme ungeheure Mengen an Daten, um ein erfolgreiches Lernen durchführen zu können, was sich als sehr nachteilig erweist und den Grund dafür darstellt, dass solche Systeme bisher erst als Forschungsprototypen in sehr stark beschränkten Domänen existieren.

Multimodale Dialogsysteme

Mittlerweile finden sich immer mehr multimodale Dialogsysteme, die verschiedene Modi fusionieren und in eine semantische Repräsentation integrieren. Angefangen von dem geradezu klassischen Beispiel für multimodale Interaktion mit Sprache und Zeigestift, dem vielzitierten “Put that there” von Bolt (Bolt 1980), über multimodale Systeme mit kombinierten Eingabemöglichkeiten aus Sprache und Zeigestift bzw. Gestik (Bringert et al. 2005; Johnston 1998) bis hin zu komplexen Systemen, die sowohl verschiedene Ein- als auch Ausgabemodi verwenden. Gerade für humanoide Serviceroboter, wie beispielsweise SIG (Okuno et al. 2002), der akustische und visuelle Daten zur sozialen Interaktion nutzt, oder auch BIRON (Bielefeld Robot Companion), der visuelle Reize, Laserdaten und akustische Daten auswertet (Fink et al. 2004; Haasch et al. 2004), ist es wichtig, so viele Informationen wie möglich über die Umwelt aufnehmen und gemeinsam verarbeiten zu können.

Die Generierung und Abstimmung verschiedener Ausgabemodalitäten spielt z.B. bei animierten Figuren, wie AdApt (Gustafson et al. 2000) oder COMIC (Foster 2004) eine große Rolle. Hier gibt es Planungskomponenten, die die mul-

timodale Ausgabe vorbereiten.

Auch in der vorliegenden Arbeit spielen multimodale Aspekte eine Rolle (siehe auch Kapitel 11.2). Allerdings werden weitere Modalitäten neben Sprache nur insoweit genutzt, wie sie zur Vermeidung von Fehlern bzw. deren Behebung beitragen können. Die Generierung einer multimodalen Ausgabe ist nicht vorgesehen, obwohl es sicherlich auch ein interessantes Forschungsgebiet für die Zukunft darstellt, zu ermitteln, inwieweit multimodale Ausgaben die Anzahl der auftretenden Fehler im Gegensatz zu unimodalen Ausgaben beeinflussen können.

2.2.2 Die TAPAS Dialogmanagementtools

Im Folgenden werden die hier verwendeten Dialogmanagementtools TAPAS (Holzapfel 2005), die auf dem Dialogsystem ARIADNE (Denecke 2002a) beruhen, näher beschrieben. Dieser Dialogmanager wurde für die vorliegende Arbeit ausgewählt, weil er neben einem vielversprechenden informationsbasierten Ansatz, auch Möglichkeiten zum Rapid Prototyping bietet, da er sprach- und domänenunabhängig ist und nur die domänen- und sprachabhängigen Teile für eine neue Applikation implementiert werden müssen, während die generellen Konzepte bereits vorhanden sind und wiederverwendet werden können. Dafür werden vektorisierte, kontextfreie Grammatiken und an die Objektorientierung angelehnte Vererbungsmechanismen benutzt, die in den folgenden Abschnitten näher erläutert werden. Außerdem sind auch bereits Möglichkeiten, den Dialogzustand zu evaluieren, und generelle Ein- bzw. Ausgabemechanismen implementiert, die genutzt werden können.

Informationsbasiert ist dieser Ansatz in dem Sinne, dass Dialogzustände durch die Spezifität der vorhandenen Information beschrieben werden. Gleichzeitig vereint er aber auch Merkmale rahmenbasierter Ansätze und ist damit flexibler als rein informationsbasierte Ansätze, indem Dialogziele definiert werden, die die Aufgaben mit all ihren Parametern angeben, die das Dialogsystem erfüllen kann.

Darüberhinaus werden multidimensionale Merkmalsstrukturen in diesem Dialogmanager benutzt (Denecke, Yang 2000), die die Möglichkeit bieten, nicht nur semantische Informationen an einem Baumknoten zu speichern, sondern auch Informationen zu der Eingabemodalität, wie beispielsweise Konfidenzmaße der Eingabe. Auf diese Weise ist es möglich, gezielt nach Worten mit einer besonders schlechten Konfidenz nachzufragen.

Der Dialogmanager benötigt verschiedene aufgaben- und domänenabhängige Ressourcen (wie auch auf Abbildung 2.1 zu sehen): Eine Ontologie, eine Spezifikation der Dialogziele, Datenbankregeln, eine Grammatik und Generierungsschablonen für die Sprachausgabe. Außerdem gibt es eine Dialogstrategie, die entscheidet, wie neue Information interpretiert und integriert wird. Diese setzt sie sich aus generischen Interaktionsmustern zusammen, die die Verarbeitung von Informationen im Diskurs generell und domänenunabhängig festlegen.

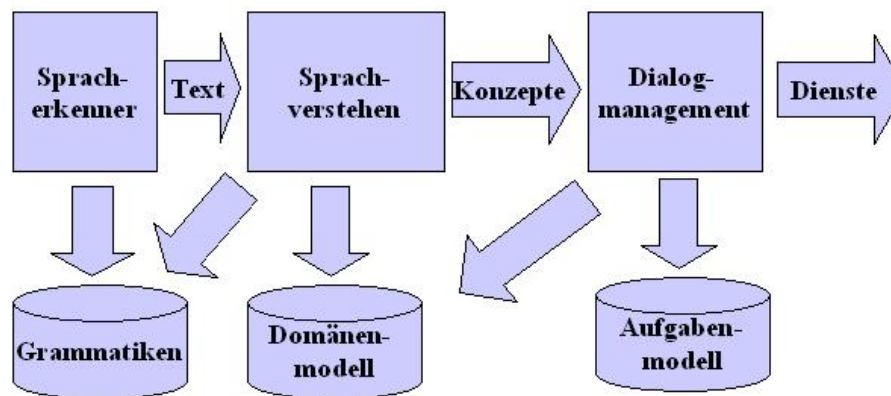


Abbildung 2.1: Der Dialogmanager und seine Ressourcen

Dialoggrammatik und Domänenmodell

Zunächst einmal wird die Eingabe des Benutzers mit Hilfe einer kontextfreien Grammatik, die mit zusätzlichen Informationen aus dem Domänenmodell angereichert ist, geparkt. Dieses Domänenmodell enthält dabei alle Konzepte, die das System kennt, um die Benutzeräußerungen verstehen zu können. Es ist als eine Ontologie von Objekten, Aktionen und Eigenschaften aufgebaut, die voneinander erben können. Auf diese Weise ist es auch möglich, auf die domänenunabhängige, generelle Ontologie zuzugreifen, die Konzepte wie z.B. verschiedene Sprechakte, generelle Dialogziele, Objekte, Aktionen und Eigenschaften, von denen dann die spezifischen wiederum erben können, enthält. Diese Kombination von Grammatik und Domänenmodell wird durch die Verwendung von vektorisierten, kontext-freien Grammatiken, die aus Nonterminalen von n-dimensionalen Vektoren partiell organisierter Elemente bestehen (Denecke 2000), ermöglicht.

Auf diese Weise kann in der Grammatik syntaktische Information von semantischer getrennt werden, wodurch die Wiederverwendung von syntaktischer Information ermöglicht wird. Beispielsweise kann die Konstruktion komplexer Nominalphrasen in einem generellen Grammatikteil behandelt werden, während die eigentliche semantische Instantiierung im domänenabhängigen Teil erfolgt.

Diese vom Dialogmanager verwendete Grammatik kann auch wieder in eine nicht-vektorierte, kontextfreie Grammatik konvertiert und vom Spracherkenner benutzt werden. Dies hat den Vorteil, dass beide Komponenten die gleiche linguistische Wissensbasis nutzen.

Aufgabenmodell

Im Aufgabenmodell sind die Dialogziele definiert, die als eine Beschreibung eines Formulars, das mittels des Mensch-Maschine-Dialogs gefüllt wird, gesehen werden können (Denecke 2002a). D.h., dass die Dialogziele von der Information, die der Benutzer im Laufe des Dialogs gibt, spezifiziert werden, und dass sie aus Objekten, Aktionen und Eigenschaften bestehen, die in der Ontologie definiert

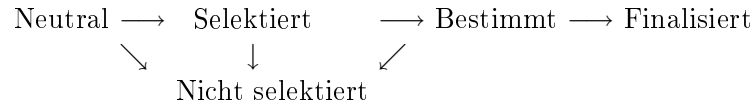


Abbildung 2.2: Dialogzustände und ihre Übergänge (entnommen aus:(Denecke 2002a)

sind. Insofern sind die Dialogziele die Verbindung zwischen dem Domänenmodell und den Diensten, die der Dialogmanager ausführen kann.

Wenn ein Dialogziel erkannt wird, sucht der Dialogmanager im Diskurs nach den entsprechenden Parametern, wie Objekten, Aktionen und Eigenschaften. Ist die Merkmalsstruktur noch unterspezifiziert, wird ein Klärungsdialog initiiert.

Generierungsschablonen

Alle Informationen vom Dialogmanager zum Benutzer werden in natürlicher, gesprochener Sprache ausgegeben. Dafür werden so genannte Generierungsschablonen verwendet, in denen die Äußerungen des Dialogmanagers abhängig vom Dialogzustand und der aktuellen Situation definiert sind. Der Dialogzustand ist über die Information in den Dialogzielen (Denecke 2002a) definiert, wie auch in der Abbildung 2.2 deutlich wird.

Am Anfang des Dialogs ist der Dialogzustand neutral. Dann werden einige Dialogziele ausgewählt und der Zustand wird selektiert. Ein selektiertes Dialogziel wird genau dann bestimmt, wenn es das einzige ist, was ausgewählt ist. Das Dialogziel wird finalisiert, wenn alle im Dialogziel spezifizierten Informationen im Diskurs vorliegen - d.h., dass alle Variablen spezifiziert sind. Nicht selektiert wird der Dialogzustand, wenn die Informationen in der hereinkommenden Hypothese des Spracherkenners nicht konsistent sind zu der bereits im Diskurs vorhandenen Information. Außerdem kann der Dialogzustand auch nicht selektiert werden, wenn der Diskurs zuvor leer war und die hereinkommende Benutzeräußerung keinen Sprechakt, sondern nur Objekte bzw. Eigenschaften enthält und damit auch kein Dialogziel ausgewählt werden kann.

In diesen Generierungsschablonen wird auf der einen Seite festgelegt, was der Dialogmanager den Benutzer in welcher Situation fragt. Auf der anderen Seite wird hier definiert, was der Dialogmanager als Antwort erwartet und wie diese neue Information im Diskurs zu integrieren ist.

Datenbanken

Die Datenbank enthält alle Objekte und ihre Eigenschaften, die in der Applikation eine Rolle spielen. Auf diese Weise kann der Dialogmanager nach verschiedenen Instanzen eines Objekts und seinen Eigenschaften, wie z.B. dem Ort dieses Objekts in der Umwelt, in der Datenbank suchen. Das ist besonders wichtig für die Integration von Gesten in den Dialogmanager, wie im Kapitel 11.2 noch verdeutlicht wird.

Dialogstrategie und Interaktionsmuster

Schließlich wird in der Dialogstrategie festgelegt, wie verschiedene Arten von Informationen in einem bestimmten Dialogzustand weiterverarbeitet werden. Die Dialogstrategie greift dabei auf unterschiedliche generische Interaktionsmuster zurück, die definieren, wie Informationen dem Diskurs hinzugefügt bzw. wieder daraus gelöscht werden können (Denecke 2002a).

Generische Interaktionsmuster können als Interaktionen zwischen Anwender und Dialogsystem angesehen werden. Sie beschreiben, wie Informationen generell im Diskurs integriert werden und sind als Zustandsübergang zwischen abstrakten Dialogzuständen definiert. Sie werden instantiiert und über die Zeit hin analog zur Diskursstruktur entwickelt, solange bis ein Dialogziel erreicht ist.

2.3 Fehler in Mensch-Maschine-Dialogen

2.3.1 Einleitung

Wie bereits in der Einleitung dieser Arbeit erwähnt, gibt es zwei Arten von Fehlern im Dialog: *Nicht-Verstehen* vs. *Missverstehen*. Beide Arten von Fehlern können darauf zurückgeführt werden, dass Dialogsysteme immer nur ein begrenztes Wissen haben und auf eine Domäne beschränkt sind. D.h., Fehler resultieren aus fehlender Abdeckung auf einer der folgenden drei Ebenen:

- Lexikalische Ebene: Wenn Worte verwendet werden, die nicht im Vokabular sind, hat dies Fehlerkennungen im Spracherkenner zur Folge, die zu einer falschen oder gar keiner semantischen Repräsentation führen (cf. engl.: *out of vocabulary (OOV)*)
- Syntaktische Ebene: Wenn bekannte Worte in unbekanntem syntaktischen Konstruktionen verwendet werden, führt dies ebenfalls zu einer falschen semantischen Repräsentation. Sprachmodellbasierte Spracherkenner haben hier bei genügend großen Trainingsmengen zwar den Vorteil gegenüber grammatikbasierten Erkennern, dass dies seltener auftritt, aber die intendierte Bedeutung kann dennoch häufig nicht in eine korrekte semantische Repräsentation umgewandelt werden. Ebenso können Skip-Mechanismen robuster semantischer Parser zwar die fehlende Abdeckung kompensieren, doch auch hier treten Probleme auf, dies in eine sinnvolle semantische Repräsentation zu überführen.
- Semantisch-pragmatische Ebene: Wenn der Benutzer Funktionen, die im System in dieser Weise nicht vorgesehen sind, anspricht, und sich somit außerhalb der Domäne des Systems befindet, kann keine oder nur eine fehlerhafte semantische Repräsentation aufgebaut werden (cf. engl.: *out of domain (OOD)*).

Die folgenden Abschnitte geben einen Überblick über andere Forschungsarbeiten zu den hier diskutierten Problemen und ordnen diese Arbeit in den gesamten Forschungskontext ein. Dabei geht es zunächst um die Fehlervermeidung.

Danach werden die Erkennung und Behandlung von Fehlern in der Spracherkennung und generelle Methoden zur Fehlerbehandlung im Dialogmanagement genauer untersucht. Schließlich folgt ein Vergleich verschiedener Dialogsysteme und ihrer Leistungen, um den Mehrwert von Fehlervermeidungs- und Fehlerbehandlungsmechanismen einordnen zu können.

2.3.2 Vermeidung von Fehlern

Fehler resultieren im Dialogbereich aus fehlender Abdeckung in einem der drei in der Einleitung genannten Bereiche: Lexikon, Syntax und Semantik bzw. Pragmatik. In allen drei Bereichen gibt es Forschungsarbeiten, die sich mit Möglichkeiten der Fehlervermeidung beschäftigen.

Fehler im Bereich des Lexikons

Fehler im Bereich des Lexikons lassen sich durch Systeme, die neue Worte lernen können, vermeiden; gleichzeitig werden mit Hilfe dieser Lernalgorithmen auch Fehler behoben. Bereits 1979 beschäftigte sich Carbonell mit inkrementellem Lernen von neuen Wörtern in dem POLITICS System (Carbonell 1979). Dort wurden neue Worte aufgrund der kontextuellen Erwartungen (syntaktisch, semantisch und pragmatisch) gelernt, die als Bedingungen für die syntaktische Kategorie und die semantische Rolle des neuen Wortes verwendet wurden. So ging Carbonell beispielsweise davon aus, dass das neue Wort entweder ein Nomen, Adjektiv oder Adverb sein müsste, wenn es auf "the" folgt. Darüberhinaus gab es noch andere Bedingungen, die sich aus dem weiteren Satzverlauf und auch aus einem vorher zusammengestellten Weltwissen ergaben. Dennoch konnte das System allein mit diesen Bedingungen das neue Wort oftmals nicht auf eine Wortart oder eine thematische Rolle herunterbrechen.

Daher beschäftigen sich neuere Systeme mit dem interaktiven Lernen neuer Wörter (Schaaf 2003; Seneff et al. 2003). Solche Systeme sind in der Lage, ein unbekanntes Wort als solches zu detektieren. Dann kann das System in einem interaktiven Dialog mit dem Benutzer die Bedeutung des Wortes und seine semantischen und syntaktischen Einschränkungen ermitteln. Danach buchstabiert der Benutzer dieses neue Wort, damit es mit seiner korrekten Schreibweise ins Vokabular aufgenommen werden kann. Zur akustischen Modellierung der unbekanntes Worte werden dabei sogenannte Head-Tail-Modelle verwendet (Schaaf 2003). Die statistische Modellierung erfolgt über die Erzeugung von automatischen Klassen aus einem großen Korpus, wie dem Broadcast News-Korpus (für eine ausführliche Beschreibung dieses vom Linguistic Data Consortium vertriebenen Korpus mit Nachrichtentexten von verschiedenen Nachrichtensendern und Zeitungen siehe Graff (2002)).

Fehler im Bereich der Syntax

Darüberhinaus gibt es Systeme, die versuchen Fehler im Bereich der Syntax zu vermeiden, indem sie nach einer fehlgeschlagenen Analyse die Ursache ermitteln, warum ein Satz nicht geparkt werden konnte. Dafür untersuchen Kiyono et al. in ihrem System die Teilparses und ermitteln alle möglichen Änderungen an

der existierenden Grammatik, die zu einem vollständigen Parse führen würden (Kiyono, Tsujii 1993). Die Elemente, die sich in keinem Teilparse finden, werden als unbekannt angesehen und es wird versucht, sie einem anderen Element als rechte Seite zuzuordnen. Problematisch daran ist, dass ein solches System übergeneralisieren könnte, so dass beispielsweise eine Regel $S \rightarrow Wort_1 \dots Wort_n$, wobei $Wort_1 \dots Wort_n$ den kompletten Satz ergeben, erstellt werden kann, die den gesamten Satz abdeckt. Dafür gibt es wiederum Regeln, die diese Übergeneralisierung verhindern, indem z.B. nur 4 Tochterknoten für eine neue Regel zugelassen sind.

Ferner gibt es Systeme, die durch interaktives Lernen sowohl Fehler im Bereich des Lexikons als auch Fehler im Bereich der Syntax vermeiden und sogar beheben können. Gavalda beschreibt ein solches System (Gavalda 2000a, b), das interaktiv neue Worte und syntaktische Konstruktionen vom Benutzer lernt und damit auch Möglichkeiten zur Fehlerbehebung bietet. Es stellt eine Kombination aus kontextfreien Grammatiken, statistischen Methoden, Interaktion mit dem Benutzer, Regelerwerb und Generalisierung dar. Kann ein Satz mit Hilfe der kontextfreien Grammatik nicht geparkt werden, so wird der interaktive Parsing-Algorithmus aufgerufen. Dieser parst zunächst den Satz in einem speziellen Modus, in dem alle Nichtterminale als oberster Knoten eines Parsebaums fungieren können. Das Ergebnis ist eine Reihe von Subnbäumen und ein oder mehrere nicht-parsbare Worte, die zusammen die Bedeutung der Äußerung ausmachen. Um den wirklichen Wurzelknoten des gesamten Parsebaums herauszufinden, werden interne Wissensquellen, wie parataktische und hypotaktische Vorhersagemodelle, und externe Wissensquellen, wie das Applikationsmodell und der Benutzer, verwendet. Ist der Wurzelknoten ermittelt, so werden die entsprechenden Tochterknoten mit Hilfe der Ontologie und natürlich wieder mit Informationen vom Benutzer gesucht. Aus dem so entstandenen Parsebaum werden dann Grammatikregeln generiert, damit die Grammatik in Zukunft diese Äußerung parsen kann. Auch die Vorhersagemodelle werden aktualisiert. Dabei wird darauf geachtet, dass die Ambiguität nicht größer werden darf und vorher korrekte Parses nicht wieder zerstört werden.

Eine andere Möglichkeit, um Fehler im Bereich der Syntax zu vermeiden, stellen n-gram-Sprachmodelle dar, da sie eine größere generelle Abdeckung aufweisen als handgeschriebene Grammatiken. Problematisch daran ist aber immer die spezielle Abdeckung für eine bestimmte Domäne. Dies ist gerade bei der in dieser Arbeit betrachteten Küchen-domäne sehr schwierig, da Sprachkorpora meist aus Zeitungstexten gewonnen werden und daher diese Domäne eher schlecht abgedeckt ist. Daher bieten sich Kombination aus einer handgeschriebenen Grammatik und n-Grammen für eine größere Abdeckung an. Beispielsweise benutzen Xu und Rudnicky ein generelles n-gram-Sprachmodell am Anfang einer Kommunikation oder in nicht näher spezifizierten Situationen (Xu, Rudnicky 2000). Stellt das System eine Klärungsfrage und erwartet damit eine bestimmte Art der Antwort vom Benutzer, so wird ein grammatikbasiertes Sprachmodell mit nur einer eingeschränkten Anzahl an Regeln, die auf die aktuelle Situation zugeschnitten sind, benutzt.

Fehler im Bereich von Semantik und Pragmatik

Fehler auf semantisch-pragmatischer Ebene lassen sich durch eine stärkere Einbindung des Kontextes vermeiden. Die Auflösung von Kontext in zwischenmenschlichen Dialogen ist ein intuitiver Prozess und daher unabdingbar für einen natürlichen, benutzerfreundlichen Mensch-Maschine-Dialog. Es gibt einige Ansätze, die sich mit der Einbindung von Kontextinformationen in den aktuellen Diskurs beschäftigen (Landragin 2002; Landragin, Romary 2003; Gustafson et al. 2000; Haasch et al. 2004; Kumar et al. 2003; Johnston et al. 1997). Der umfangreichste ist jedoch der von Filisko und seinen Mitarbeitern entwickelte Server zur Auflösung des Kontexts im Galaxy Dialogsystem (Filisko, Seneff 2003). Diese Komponente ist dabei nicht nur dafür verantwortlich, die Äußerung im Kontext von vorhergehenden Benutzeräußerungen, sondern auch im räumlichen und zeitlichen Kontext unter Einbeziehung von Inferenzen und gemeinsamem Weltwissen zu interpretieren. Daher besteht sie aus verschiedenen Teilbereichen: Auflösung von anaphorischen und deiktischen Pronomina, Nachschlagen der Dialoghistorie für den aktuellen Diskurs, Auflösung von Ellipsen und Satzfragmenten und Ergänzen von fehlenden Informationen mit Hilfe von externen Wissensquellen.

Darüberhinaus können Fehler auf semantisch-pragmatischer Ebene auch dadurch entstehen, dass der Benutzer eine andere Erwartung an das System hat und daher auf Funktionen referiert, die das System nicht ausführen kann. Dies ist besonders schwierig, weil in diesem Fall das vom Benutzer verwendete Vokabular nicht mehr übereinstimmt mit dem vom System verstandenen. Dafür ist es wichtig, dass das System dem Benutzer verdeutlicht, was es kann und was nicht.

Zusammenfassung

Insgesamt finden sich in der Literatur die verschiedensten Möglichkeiten, um Fehler vermeiden zu können: Einige Systeme nutzen Lernfunktionen, um die niemals vollständigen linguistischen Ressourcen vergrößern zu können, andere machen sich zusätzliche Wissensquellen, wie beispielsweise den Kontext, zu Nutze. Was den Systemen bisher noch fehlt, ist ein integrierter Ansatz zur Fehlervermeidung und anschließenden -behandlung. Hinzu kommt, dass das Wissen, was sich die Systeme interaktiv angeeignet haben, meist unvollständig und damit auch in anderen Domänen und Kontexten u.U. nicht wiederverwendbar ist.

2.3.3 Erkennung und Behandlung von Fehlern

Wie in verschiedenen Forschungsarbeiten bereits untersucht wurde, liegt eine wichtige Ursache für Fehler in Dialogsystemen in der fehlerhaften Spracherkennung (Choularton, Dale 2004; Filisko, Seneff 2004; Gorrell 2003; Gurevych, Porzel 2003). Es gibt nun die folgenden Möglichkeiten, dem zu begegnen:

- Verbessern der Spracherkennung, um die Anzahl der Fehler zu minimieren
- Verbessern der Erkennung von Fehlern und der Algorithmen zur Fehlerbehandlung

Diese beiden Möglichkeiten stehen nicht im Widerspruch zueinander, sondern können sich gegenseitig ergänzen. Darum beschäftigen sich die folgenden Abschnitte zunächst mit der Verbesserung der Spracherkennung in bestimmten Kontexten und der Erkennung von Fehlern durch das Verhalten des Benutzers, bevor es um Konfidenzmaße geht, die sowohl zum Erkennen der Fehler als auch zum Verbessern der Spracherkennung genutzt werden können. Im Anschluss daran werden die Auswirkungen der Generierungsstrategie, also der Antworten des Systems an den Benutzer, im Fehlerfalle beleuchtet. Auch bei dem Abschnitt über Klärungsdialoge geht es darum, wie das System effektive Rückfragen an den Benutzer stellen kann, damit dieser möglichst schnell sein Ziel erreicht. Anschließend werden verschiedene Modelle zur Fehlerbehandlung, wie sie in heutigen Dialogsystemen verwendet werden, vorgestellt: Angefangen von heuristischen und entscheidungstheoretischen Modellen bis hin zu Modellen, die mit Lernen durch Verstärkung arbeiten. Schließlich werden noch die Fehlerbehandlungsstrategien dargelegt, die bereits in ARIADNE implementiert wurden (Denecke 2002a), da das hier verwendete Dialogsystem TAPAS auf die in ARIADNE entwickelten Mechanismen zurückgeht.

Verbessern der Spracherkennung für Fehlersituationen

Zur Verbesserung der Spracherkennung in bestimmten Kontexten arbeiten einige Forschergruppen mit der Aktivierung von Subgrammatiken im Spracherkenner (Lemon 2004; Fügen et al. 2004; Stent et al. 1999). Dabei bestimmt der Kontext, welche Grammatikregeln bevorzugt werden bzw. welches Sprachmodell aktiviert wird. Am Anfang und in nicht näher spezifizierten Situationen wird ein generelles n-gram-Sprachmodell verwendet, während ein spezialisiertes Sprachmodell, das sowohl als n-gram als auch als grammatikbasiertes Sprachmodell realisiert sein kann, in spezifischen Situationen basierend auf der vorhergehenden Systemnachfrage benutzt wird (Xu, Rudnicky 2000; Fosler-Lusier, Kuo 2001). In Solsona et al. (2002), wird das zustandsunabhängige n-gram-Sprachmodell auch kombiniert mit einer zustandsabhängigen Finite-State-Grammatik, indem die akustischen Konfidenzmaße verglichen werden. Auf diese Weise können Perplexität und Wortfehlerraten signifikant reduziert werden.

Ein anderer Ansatz zum Verbessern der Spracherkennung basiert darauf, dass der Spracherkenner eine Liste möglicher Erkennungshypothesen für jede Äußerung ausgibt, die im Anschluss an die Erkennung neu sortiert werden kann. So haben sich einige Forscher mit dem Neusortieren der n-besten Liste aufgrund von verschiedenen Merkmalen aus der Spracherkennung, dem semantischen Parsen oder dem Dialogmanagement (Chotimongkol 2001; Moore et al. 1995) beschäftigt. Bousquet-Vernhettes und Vigouroux benutzen beispielsweise einen domänenabhängigen konzeptuellen Parser (Bousquet-Vernettes, Vigouroux 2003), um falsche Hypothesen, die nicht erfolgreich geparkt werden konnten, herauszufiltern. Gurevych und Porzel (Gurevych, Porzel 2003) schließen Hypothesen aus, die inkonsistent mit der Domänenontologie sind. Dazu bauen sie neben den Konfidenzmaßen aus der Spracherkennung ein Domänenmaß auf, das angibt, wie wahrscheinlich eine Äußerung in der gegebenen Domäne ist. Zollo (Zollo 2003) benutzt eine kontextfreie Grammatik, die die Benutzeräußerungen in

Sprechakte umwandelt. Damit lassen sich Äußerungen, die vom Sprachmodell des Spracherkenners nicht geparkt werden konnten, ermitteln. Chotimongkol benutzt ein lineares Regressionsmodell, das aus einer Kombination von syntaktischen und semantischen Merkmalen besteht (Chotimongkol 2001). Interessant an ihrer Arbeit ist, dass sie auch menschliche Spracherkennung berücksichtigt und vergleicht, welches Wissen Menschen benutzen, um Hypothesen neu zu sortieren.

Erkennung von Fehlern durch die Benutzerreaktion

Darüberhinaus beschäftigen sich einige Forscher mit dem Erkennen von Fehlern anhand der Reaktionen des Benutzers, da sich herausgestellt hat, dass Benutzer Korrekturen anders aussprechen und es zu *Hyperartikulation* kommt (Swerts et al. 2000; Soltau, Waibel 2000). Oviatt (Oviatt 1998) beschreibt die wichtigsten prosodischen Merkmale, die zusammengenommen Hyperartikulation ausmachen, folgendermaßen:

- Einfügen oder Verlängern der Pausen
- Mehr Unterbrechungen in der Äußerung (cf. engl. disfluency)
- Dehnen von Phonemen
- Überdeutliche Aussprache

Außerdem gibt es auch im Bereich der Konversationsanalyse eine ganze Reihe von Studien, die nahe legen, dass das menschliche Verhalten bei Fehlern vorhersagbar ist (Sacks et al. 1974; Boden, Zimmerman 1991). Empirische Studien belegen dies, indem sie eine spezielle Modellierung von Fehlerdialogen nutzen (Stifelman 1993; Choularton, Dale 2004; Hirschberg et al. 2004; Bousquet-Vernettes et al. 2003) und dadurch verbesserte Möglichkeiten zum Erkennen und zur Behandlung von Fehlern liefern. Stifelman (Stifelman 1993) klassifiziert die Strategien, die Menschen benutzen, nachdem ein Fehler aufgetreten ist, wie folgt:

- Exakte Wiederholung
- Teilweise Wiederholung
- Umformulierung

Dabei lässt sich die Umformulierung noch weiter in die Vereinfachung der Aussage, Erklärung und Unterteilen in mehrere Äußerungen unterteilen. In ähnlicher Weise haben auch Shine et al. (Shine et al. 2003) Fehlerstrategien von Benutzern klassifiziert. So arbeiten Hirschberg et al. und auch Stifelman mit angepassten Sprachmodellen, um Fehlerdialoge besser modellieren zu können, und erreichen damit eine signifikante Reduzierung von Wortfehlerraten (Hirschberg et al. 2004; Stifelman 1993).

Erkennung von Fehlern mit Hilfe von Konfidenzmaßen

Zur Erkennung und Diagnose von Fehlern werden häufig Konfidenzmaße aus dem Spracherkenner herangezogen. Sie basieren auf dem Vergleich der Erkennungswahrscheinlichkeiten verschiedener Hypothesen des Erkenners. Dabei können sie entweder direkt auf Kenngrößen während der Suche des Erkenners zugreifen, wie beispielsweise die Anzahl der Hypothesen, Scores o.ä., oder auf dem Erkennungsergebnis z.B. in Form eines Worthypothesengraphen oder einer Lattice arbeiten (Schaaf, Kemp 1997; Kemp, Schaaf 1997).

In letzter Zeit werden außerdem verstärkt auch Maße außerhalb des Spracherkenners benutzt, um Verbesserungen zu erzielen (den Bosch et al. 2001; Carpenter et al. 2001). So verwenden Bohus und Rudnicky (Bohus, Rudnicky 2002) neben reinen Spracherkennungsmerkmalen, wie beispielsweise der Anzahl von Worten und die Rate an Worten mit zu geringer Konfidenz, auch Merkmale aus den Bereichen Parsing und Dialogmanagement, wie z.B. Anzahl nicht parsbarer Worte, Dialogzustand, Dauer des aktuellen Dialogzustands, Anzahl an Turns seit Beginn des Dialogs, erwartete Konzepte, usw., um eine Konfidenz für die gesamte Äußerung ermitteln zu können. Sie zeigen mit Hilfe von Experimenten, dass diese Merkmale mit verschiedenen automatischen Klassifizierungstechniken, wie Bayesschen Netzen, Support Vector Machines, Boosting und neuronalen Netzen, benutzt werden können, um aussagekräftige Konfidenzmaße über gesamte Äußerungen zu ermitteln. Dabei lassen sich die besten Ergebnisse mit Boosting erzielen. Darüberhinaus haben sie auch ein Modell, das das Vertrauen des Systems in die erkannten Äußerungen über mehrere Benutzerturns hinweg modelliert, entwickelt (Bohus, Rudnicky 2005). Dafür wird ein maschineller Lernansatz benutzt, der das Vertrauen in ein erkanntes semantisches Konzept anhand der gegebenen Konfidenzen und der darauf folgenden expliziten bzw. impliziten Bestätigungen durch den Benutzer berechnet.

Auch San-Segundo und seine Mitarbeiter gehen in der Berechnung der Konfidenz über die reinen Spracherkennermerkmale hinaus, indem sie neben Merkmalen auf Wortebene auch Merkmale auf Äußerungs- und Konzeptebene in ihrem neuronalen Netz benutzen, um falsch erkannte Worte und Äußerungen, die außerhalb der Domäne sind, zu erkennen (San-Segundo et al. 2000). Ebenso haben auch Walker et al. einen Erkenner für Missverständnisse entwickelt, der auf Merkmalen aus der Dekodierung, dem Dialogmanagement und dem Sprachverstehen basiert (Walker et al. 2000).

Konfidenzmaße können außerdem auf verschiedenen Granularitätsstufen verwendet werden. Bisher berechnen die meisten Systeme Konfidenzen zwar auf Äußerungsebene, allerdings gibt es auch schon Ansätze mit Konfidenzen auf Wortebene (Hazen et al. 2002), was für das Dialogmanagement ein sehr wichtiges Maß ist, um nach einzelnen Worten, die eine geringe Konfidenz haben, nachfragen zu können.

Generierungsstrategie: Auswirkungen der Systemantworten

Zunächst stellt sich die Frage, ob die Äußerung des Benutzers explizit oder implizit bestätigt werden sollte, um es dem Benutzer bei der Korrektur einer

Äußerung so einfach wie möglich zu machen. Bei einer expliziten Bestätigung fragt das System nach einer Äußerung des Benutzers, ob es ihn richtig verstanden hat, indem es die letzte Benutzeräußerung wiederholt bzw. leicht umformuliert. Implizite Bestätigung bedeutet demgegenüber, dass das System das, was es von der vorhergehenden Äußerung des Benutzers verstanden hat, in eine neue Anfrage an den Benutzer integriert. Dies fördert die Natürlichkeit und die Geschwindigkeit des Dialogs und der Benutzer muss nicht immer wieder das, was er gerade gesagt hat, bestätigen. Andererseits birgt die implizite Bestätigung das Risiko, dass der Benutzer eher verwirrt wird (Weegels 2000) und dass es für ihn schwieriger ist, etwas falsch Verstandenes zu korrigieren.

Um die Vorteile beider Methoden kombinieren zu können und gleichzeitig ihre Nachteile zu vermeiden, haben Krahrmer et al. (Krahrmer et al. 2001) in einer detaillierten Studie ermittelt, wie Benutzer auf diese beiden Arten der Bestätigung im Falle einer erfolgreichen Kommunikation und demgegenüber im Fehlerfalle reagieren. Daraus haben sie eine Reihe von Merkmalen, wie beispielsweise die Länge der Äußerung, die Satzstellung, usw., extrahiert und diese in positive und negative Hinweise unterteilt. Positive Hinweise führen zu einer impliziten und negative zu einer expliziten Bestätigung. Damit ist es für den Benutzer einfacher, Fehler zu korrigieren, und trotzdem einen natürlichen Dialog zu führen.

Außerdem sind einige Wizard-of-Oz-Studien durchgeführt worden, um zu ermitteln, wie Menschen mit Spracherkennungsfehlern umgehen. Dabei übernehmen die Versuchspersonen die Rolle des "Wizard" in diesen Studien: Sie hören die Äußerung des Benutzers nicht, sondern bekommen nur den Text, der vom Spracherkennung erkannt wurde - eventuell mit zusätzlichen Konfidenzmaßen versehen (Zollo 2003; Skantze 2003). In Zollos Studie geben die Versuchspersonen selbst bei recht guter Spracherkennung dem Benutzer sehr viele Rückmeldungen über das, was verstanden wurde (Zollo 2003), was sich aber eventuell auch darauf zurückführen lässt, dass die Versuchspersonen ihre Aufgaben missverstanden haben und versuchen einen maschinellen Wizard zu simulieren. Skantze (Skantze 2003) beobachtet demgegenüber, dass die Versuchspersonen in einer sehr ähnlichen Studie den Benutzern nicht explizit zeigen, dass etwas falsch oder überhaupt nicht verstanden wurde, sondern versuchen, den Dialog durch andere Fragen weiter aufrecht zu erhalten. Die sehr unterschiedlichen Ergebnisse der Studien verdeutlichen, dass Menschen noch weitere Fehlerbehandlungsmethoden als heutige Dialogsysteme benutzen, die auch vom Kontext des Dialogs abhängig sind. So gibt es hier noch einigen Forschungsbedarf, um zu klären, warum sich diese Ergebnisse so stark unterscheiden und welche Methoden der Fehlerbehandlung in der zwischenmenschlichen Kommunikation eine Rolle spielen.

Shin et al. (Shin et al. 2002) haben 161 Dialoge von der NIST 2000 Communicator Evaluation (Walker et al. 2001) in bezug auf das Systemverhalten analysiert (explizite Bestätigung, implizite Bestätigung, Hilfe, Wiederholung der letzten Systemäußerung, Abweisen der Äußerung, nicht Weiterverfolgen), um herauszufinden, wie die Benutzer bemerken, dass ein Fehler aufgetreten ist. Die Ergebnisse unterstreichen das Bedürfnis des Benutzers nach einer expliziten Bestätigung, da die Benutzer mehr Zeit benötigen, um den Faden des Dialogs

wieder aufzunehmen, und sogar öfter versagen, wenn sie die Fehler nur durch implizite Bestätigungen bemerken. Ein Ansatz, der die Ausgabe der Erkennung von Fehlern nutzt, um zwischen unterschiedlichen Maßen an Systeminitiative unterscheiden zu können, ist in Bulyko et al. (2005) beschrieben.

Darüberhinaus gibt es auch Arbeiten zu der Anpassung der Generierungsstrategie des Dialogmanagers an Fehlersituationen. Ebenfalls mit Dialogen aus dem NIST 2000 Communicator Korpus (Walker et al. 2001) haben Goldberg et al. (Goldberg et al. 2003) untersucht, wie die Art der Antwort (exakte Wiederholung, paraphrasierte Wiederholung, Abbruch, usw.) die Erfolgsrate und die Benutzerzufriedenheit beeinflusst. Die Ergebnisse zeigen, dass Entschuldigungen des Systems in Kombination mit Paraphrasen zu geringeren Wortfehlerraten bei der Verarbeitung der Benutzeräußerungen führen, was zu einer schnelleren Fehlerbehebung und größeren Benutzerzufriedenheit beiträgt.

Klärungsdialoge

Wurde ein Fehler mit den zuvor beschriebenen Methoden erkannt, so gibt es verschiedene Möglichkeiten, wie das System darauf reagieren kann. Die einfachste ist, anzuzeigen, dass etwas falsch verstanden wurde, und die Frage an den Benutzer zu wiederholen. Allerdings führt dies auch zu Frustrationen beim Benutzer, so dass einige Forscher nach anderen Lösungen gesucht haben: Gorrell et al. (Gorrell et al. 2002) klassifizieren die nicht verstandenen Äußerungen mit Hilfe eines Entscheidungsbaums, um sinnvolle Rückfragen an den Benutzer stellen zu können. Dazu verwenden sie Merkmale wie die einzelnen Worte und ihre Konfidenzen, die Anzahl der Worte in der Äußerung und das Vorkommen bestimmter Schlüsselworte.

Raux und Eskenazi (Raux, Eskenazi 2004) versuchen ebenfalls die Rückfragen an den Benutzer möglichst geschickt zu stellen. Dafür ermitteln sie die nächste grammatikalisch mögliche Äußerung zu der vom Spracherkenner erkannten mit Hilfe der Levenstein-Distanz (Levenstein 1996). Die Levenstein-Distanz ist als eine Metrik auf dem Raum der Symbolsequenzen definiert: Sie gibt die zwischen zwei Zeichenketten A und B vorhandene Distanz an, die gleich der minimalen Anzahl von Editierschritten ist, mit denen Sequenz A in Sequenz B umgewandelt werden kann. Als elementare Editierschritte gelten dabei:

- Ersetzen eines Symbols durch ein anderes (cf. engl. *Substitution*)
- Löschen eines Symbols (cf. engl. *Deletion*)
- Einfügen eines Symbols (cf. engl. *Insertion*)

Diese so ermittelte Äußerung wird benutzt, um eine Nachfrage an den Benutzer, ob es das war, was er sagen wollte, zu stellen.

Modelle zur Fehlerbehandlung

Die meisten Fehlerbehandlungsmethoden, die in heutigen Dialogsystemen verwendet werden, basieren auf heuristischen Modellen (Bousquet-Vernettes et al.

2003; McTear et al. 2003; Krahmer et al. 2001; Komatani, Kawahara 2000). Einige verwenden sogar handgeschriebene Regeln, um aufgrund von zu geringen Konfidenzen mit potentiellen Fehlern umgehen zu können.

Eine interessante Ausnahme stellt das entscheidungstheoretische Modell von Horvitz und Paek dar (Horvitz, Paek 2001). Sie erweitern die später erläuterte Grounding-Theorie (siehe auch Kapitel 2.4.2), die aus der Kommunikationsanalyse zwischenmenschlicher Gespräche stammt und Aussagen darüber macht, wie neue Informationen dort in eine gemeinsame Wissensbasis aufgenommen werden. Um über die richtigen Korrekturaktionen in einem Dialog entscheiden zu können, benutzen sie Konfidenzen als einen Faktor in einem Bayesschen Netz. Dabei betrachten sie Dialog als Aktion in Unsicherheit (cf. engl. *action under uncertainty*) (Paek, Horvitz 2000) und schlagen eine Architektur mit den folgenden vier unabhängigen Ebenen vor, um Überzeugungen analysieren und Grounding-Entscheidungen modellieren zu können: Kanal, Signal, Intention und Konversation. Die Unsicherheiten innerhalb einer Ebene werden in einem Bayesschen Netz explizit repräsentiert, während die Grounding-Entscheidungen auf erwartetem Nutzen und Berechnungen der Informationswerte basieren. Das Modell ist zum einen praktisch einsetzbar und weist auf der anderen Seite eine solide theoretische Basis auf. Allerdings gibt es noch Schwachstellen in dem Modell: Vieles muss manuell erstellt werden, wie die Dienste der verschiedenen Grounding-Aktionen, die Parameter der Netze, usw. Außerdem wurde das Modell bisher nur für "Command-and-Control"-Applikationen getestet und müsste erst noch auf die generelle Klasse der aufgabenorientierten Dialogsysteme erweitert werden.

Eine Reihe von Forschern hat in letzter Zeit Methoden zum Lernen durch Verstärkung (cf. engl. *reinforcement learning*) angewandt, um optimale Richtlinien zur Dialogkontrolle ermitteln zu können (Levin et al. 2000; Scheffler, Young 2002; Singh et al. 2000), wobei auch die Fehlerbehandlung eine Rolle spielt. Dabei wird das Dialogmanagement als ein Markov-Entscheidungsprozess angesehen. Der Zustandsraum wird üblicherweise als eine Abstraktion der möglichen Dialogzustände modelliert. Die Menge an Aktionen entspricht direkt der Menge der möglichen Systemaktionen oder zumindest einer Teilmenge davon und die Belohnungsstruktur kann auf die verschiedensten Dialogevaluationsmetriken zurückgreifen.

Bei Singh et al. (Singh et al. 2000) werden Belohnungen basierend auf einem binären Maß zur Erledigung einer Aufgabe definiert und verschiedene Möglichkeiten der Initiative, der Bestätigung und dem Stellen von Klärungsfragen untersucht und gelernt. In der anschließenden Evaluation zeigt das System signifikante Verbesserungen und es stellt sich heraus, dass die gelernte Dialogstrategie sogar besser ist als andere, manuell erstellte.

Das größte Problem dieser Ansätze liegt in der immens großen Menge an benötigten Trainingsdaten. Als Ausweg benutzen einige Forscher daher Benutzersimulationen, um Daten zu gewinnen. Levin et al. (Levin et al. 2000) entwickeln mit Hilfe von überwachtem Lernen aus einem Dialogkorpus ein einfaches Benutzermodell und benutzen dieses, um daraus einen generellen Trainingskorpus ableiten zu können. Das Benutzermodell von Scheffler und Young (Scheffler, Young 2002) geht noch darüberhinaus und modelliert explizit mögliche Fehler

und Bedingungen, um Benutzerkonsistenz zu gewährleisten.

Ein entscheidender Vorteil dieser Ansätze liegt darin, dass das Dialogverhalten direkt aus der Erfahrung gelernt wird und so gut an die Charakteristika einzelner Domänen angepasst werden kann. Darüberhinaus basieren sie auf einer soliden theoretischen Basis. Allerdings skalieren die heutigen Ansätze noch nicht gut mit der Anzahl an Aufgaben, die das System ausführen kann, und die Strategien sind in anderen Domänen oder auch nur für andere Aufgaben derselben Domäne nicht wiederverwendbar.

Ansätze zur Fehlerbehandlung in ARIADNE

Da die TAPAS Dialogmanagementtools (Holzapfel 2005) auf die im Dialogmanager ARIADNE implementierten Dialogalgorithmen zurückgehen (Denecke 2002a), sollen hier auch kurz die Fehlerbehandlungsmöglichkeiten, die bereits in ARIADNE entwickelt wurden, erläutert werden. Es gibt dort ein Korrekturinteraktionsmuster, das dazu dient, fehlinterpretierte Information zurückzunehmen bzw. zu überschreiben. Dafür werden fünf Varianten des Korrekturinteraktionsmusters instantiiert, die es erlauben:

- den Dialog neu zu starten,
- die letzte Äußerung zurückzunehmen,
- die letzte Äußerung durch etwas Anderes zu ersetzen,
- einen Teil der letzten Äußerung zurückzunehmen,
- einen Teil der letzten Äußerung durch etwas Anderes zu ersetzen.

Alle beschriebenen Situationen sind dabei durch eine Diskrepanz zwischen der Benutzerintention und der durch das System angenommenen semantischen Repräsentation der Benutzerintention gekennzeichnet. D.h. die semantische Interpretation der Benutzerintention muss korrigiert werden, wobei Seiteneffekte auf den aktuellen Dialogzustand zu beachten sind.

Allerdings bleiben einige Fragen ungeklärt, wie z.B., wie das Dialogsystem dem Benutzer die Möglichkeiten dieser Instantiiierungen der Korrekturinteraktionsmuster mitteilt, da er sie nur dann richtig einsetzen kann, wenn er ihre Eigenschaften genau kennt. Auf der anderen Seite erscheinen die momentan gewählten natürlichsprachlichen Realisierungen, wie beispielsweise “abort” (“abbrechen”) oder “undo” (“rückgängig machen”), wenig intuitiv. Daher ist es entscheidend, das Verhalten der Benutzer im Fehlerfalle genauer zu analysieren und sinnvolle natürlichsprachliche Realisierungen zu ermitteln.

Ferner gibt es hier keine system-initiierte Instantiiierung des Korrekturmusters, da die den Merkmalsstrukturen zugrundeliegende Beschreibungssprache dafür nicht ausdrucksstark genug ist. Lediglich Typ- und Konsistenzprüfungen werden in ARIADNE vorgenommen.

Insgesamt heißt es auch in der Dissertation, in der ARIADNE entworfen und implementiert wird, dass “die vorliegende Implementierung von einem vollständigen, natürlichen Korrekturalgorithmus noch weit entfernt ist” (Denecke 2002a, Seite 143).

System	Fehlerrate
CMU Communicator	32%
CU Communicator	27%
How May I Help You	36%
Jupiter	28%

Tabelle 2.1: Einige Dialogsysteme und ihre Rate an Verständnisfehlern

2.3.4 Vergleich verschiedener Dialogsysteme und ihrer Leistungen

Wie bereits in dem Beispiel im Kapitel 1.1 deutlich wurde, kämpfen heutige Dialogsysteme mit zahlreichen Verständnisfehlern. Auch in der Literatur werden relativ hohe Fehlerraten berichtet, wie in der Tabelle 2.1 mit einigen bekannten Dialogsystemen und ihren Fehlerraten deutlich wird. Es wird dabei immer die Anzahl an falsch erkannten Benutzeräußerungen ausgewertet, unabhängig von der Anzahl falsch erkannter Worte in einer Benutzeräußerung, die in der Spracherkennung als wichtigstes Evaluationsmaß dient. Eine Benutzeräußerung wird dann als “falsch erkannt” angesehen, wenn sie in ihrer Gesamtheit nicht der Intention des Benutzers entspricht.

Die Rate falsch erkannter Benutzereingaben liegt bei den hier vorgestellten Dialogsystemen bei ungefähr 30% (cf. Tabelle 2.1, unabhängig von ihrer Domäne, Sprache oder dem verwendeten Dialogansatz. So sind beispielsweise der “CMU Communicator” (Carpenter et al. 2001) und der “CU Communicator” (San-Segundo et al. 2000) Urlaubsplanungssysteme, die Hotels und Autos reservieren und Kurztrips planen können. “How May I Help You” (Walker et al. 2000) ist demgegenüber ein Telefonweiterleitungssystem für die AT&T-Kunden-Hotline. “Jupiter” (Hazen et al. 2002) ist wiederum in einer ganz anderen Domäne zu Hause und bietet seinen Benutzern Informationen übers Wetter an. Hinzu kommt, dass die Fehlerraten noch weit höher sind, wenn man nur Fehlerkorrekturdialoge betrachtet. D.h., durch die veränderte Sprechweise und die Hyperartikulation kommt es dazu, dass 2-3 mal so viel Äußerungen falsch verstanden werden wie in einem normalen Dialog (Swerts et al. 2000; Levow 1998).

Ferner sind bei der in dieser Arbeit betrachteten Kommunikation mit einem Haushaltsroboter höhere Fehlerraten zu erwarten, da die Benutzer nicht im Detail wissen, was der Roboter kann und wie sie sich ihm gegenüber am besten verhalten. Diese Domäne ist damit nicht so klar eingegrenzt, wie beispielsweise die Flugauskunftsdomäne (Carpenter et al. 2001) oder die der Telefonweiterleitungssysteme (Walker et al. 2000). Diese höheren Fehlerraten bestätigen sich, wie in den folgenden Kapiteln deutlich wird, so dass es eine große Herausforderung darstellt, eine generelle Lösung für die Fehlerbehandlung zu finden, um in Zukunft effizientere Dialoge führen zu können.

2.3.5 Zusammenfassung

Den bisherigen Arbeiten im Bereich von Fehlererkennung und -behebung fehlt ein durchgängiges, domänen- und sprachunabhängiges Modell, das beides adäquat abdecken kann. Dies ist angesichts der nach wie vor recht hohen Fehleraten in heutigen Dialogsystemen (siehe auch Kapitel 2.3.4) von besonderem Interesse. Es bleibt außerdem zu klären, welche Arten von Fehlern wie häufig auftreten und wie sich diese am besten vermeiden lassen.

Ferner liegt in der vorliegenden Arbeit der Fokus verstärkt auf semantischen Fehlern, die sich nicht oder zumindest nicht nur aus Spracherkennungsfehlern ergeben, wie dies in den anderen dargestellten Arbeiten zumeist der Fall ist. Darüberhinaus werden nicht nur Probleme, die sich aus Verstehensfehlern der Maschine ergeben, sondern auch solche, die durch die Unerfahrenheit des Benutzers mit einem solchen System (in diesem Fall: mit einem Küchenroboter), entstehen, untersucht. Gerade die Mensch-Roboter-Kommunikation ist in dieser Hinsicht besonders kritisch, weil der Benutzer nur eine vage Vorstellung von den begrenzten Fähigkeiten des Roboters hat. Außerdem werden Erkenntnisse aus zwischenmenschlichen Dialogen auf Mensch-Maschine-Dialoge übertragen, denn auch dort finden sich immer wieder Situationen, in denen Missverständnisse und Fehler im Dialog auftreten, die gelöst werden müssen, wie im folgenden Abschnitt erläutert.

2.4 Analyse zwischenmenschlicher Dialoge

2.4.1 Einleitung

Nachdem im vorangegangenen Abschnitt beschrieben wurde, wie heutige Dialogsysteme versuchen, mit Fehlern im Dialog auf die eine oder andere Art und Weise fertig zu werden, beschäftigt sich dieser Abschnitt mit der zwischenmenschlichen Kommunikation. Dabei stellt sich die Frage, ob sich über die bereits beschriebenen Strategien hinaus noch weitere Anhaltspunkte in der zwischenmenschlichen Kommunikation finden lassen, die für eine effektive Fehlerbehandlung in Mensch-Roboter-Dialogen verwendbar sind. Dies ist aufgrund der von Reeves und Nass herausgefundenen Tatsache möglich, dass Menschen, wenn sie mit Computern in natürlicher Sprache kommunizieren, ihre kommunikativen Fähigkeiten nutzen, wie sie sie in zwischenmenschlichen Gesprächen auch anwenden (Reeves, Nass 1996).

Dabei geht es zunächst um den Austausch von Informationen zwischen menschlichen Gesprächspartnern. Im Anschluss daran werden Klärungsdialoge, wie sie in zwischenmenschlichen Dialogen zur Fehlerbehebung benutzt werden, als ein Beispiel für effektive Fehlerbehebung herangezogen. Danach geht es speziell um Dialoge mit ungleichen Gesprächspartnern, die eine unterschiedliche Kompetenz in Grammatik und Vokabular der verwendeten Sprache aufweisen, und deren Übertragbarkeit auf den Mensch-Roboter-Dialog. Der darauffolgende Abschnitt beschäftigt sich mit den verschiedenen Formen der Metakommunikation zur Fehlerbehandlung. Als Metakommunikation bezeichnet man im zwischenmenschlichen Dialog die Kommunikation über Kommunikation (Bußmann

1990): D.h., der Sprecher erläutert oder modifiziert durch unmittelbaren Bezug auf vorhergehende Äußerungen diese durch Korrektur, Präzisierung, Stellungnahme, Kommentar, o.ä. Zum Schluss werden Erkenntnisse aus der menschlichen Spracherkennung in bezug auf den Einsatz von Weltwissen und Kontextmanagement in Dialogen dargelegt.

2.4.2 Grounding: Austausch von Informationen zwischen menschlichen Dialogpartnern

Ein essentieller Bestandteil des kommunikativen Prozesses ist das Etablieren von gemeinsamem Wissen zwischen den Teilnehmern eines Dialogs sowohl in der zwischenmenschlichen wie auch in der Mensch-Roboter-Kommunikation. Dieser Prozess, der *Grounding* genannt wird, bezeichnet das Hinzufügen neuer Informationen zur gemeinsamen Wissensbasis (Traum 1999; Traum, Dillenbourg 1998; Poesio, Traum 1998), so dass die Gesprächspartner einen Bezugsrahmen teilen. D.h., eine zentrale Größe für eine erfolgreiche Kommunikation ist das Wissen über den Gegenstandsbereich und das Wissen über das Wissen des anderen.

In der zwischenmenschlichen Kommunikation haben wir daher effiziente Strategien zum Grounding, wie beispielsweise nonverbale Signale oder auch verbale Zustimmung bzw. Ablehnung. Laut Clark und Schäfer (Clark, Schäfer 1989) wird die gemeinsame Wissensbasis (cf. engl. *common ground*) erweitert, indem die Informationen, die nur einem Gesprächspartner zur Verfügung stehen, in einer sogenannten *Präsentationsphase* in die Kommunikation eingebracht werden. Erst wenn der andere Gesprächspartner in einer sogenannten *Akzeptanzphase* deutlich gemacht hat, dass er sie aufgenommen hat, werden sie Teil der gemeinsamen Wissensbasis. Dabei versuchen beide Gesprächspartner jedoch die Verständigung mit möglichst geringem Aufwand zu realisieren, wie es auch der Konversationsmaxime von Grice entspricht (Grice 1975): Die gemeinsame Wissensbasis wird im Dialogverlauf erweitert, aber gegebenenfalls auch wieder restrukturiert, wenn einer der Gesprächspartner bemerkt, dass eine seiner Annahmen falsch war. Die beiden Phasen werden bei Clark und Schäfer (Clark, Schäfer 1989) wie folgt dargestellt:

Präsentationsphase: **A** präsentiert **B** eine Äußerung **u**. **A** macht das unter der Voraussetzung, dass er, wenn **B** einen Hinweis **e** dafür liefert, dass er **u** verstanden hat, glauben kann, dass **B** versteht, was **A** mit **u** meint.

Akzeptanzphase: **B** akzeptiert die Äußerung **u**, indem er einen Hinweis **e** liefert, dass er glaubt zu verstehen, was **A** mit **u** meint. **B** macht das unter der Voraussetzung, dass, sobald **A** den Hinweis **e** wahrgenommen hat, dieser auch glaubt, dass er **B** verstanden hat.

D.h., nach diesen beiden Phasen ist in der gemeinsamen Wissensbasis von **A** und **B**, dass **B** verstanden hat, was **A** meint.

In weitergehenden Forschungen (Traum 1998, 1999; Traum, Dillenbourg 1998) erläutert Traum einige Probleme dieses Modells, wie beispielsweise die starre Unterteilung in zwei verschiedene Phasen, die in der Realität nicht haltbar ist, und schlägt stattdessen das *grounding acts model* vor. Dort gibt es verschiedene *grounding acts*, die eine Erweiterung der Sprechakte, wie sie in (Austin 1962; Searle 1969) definiert sind, darstellen und zusätzliche Akte, wie

z.B. turn-taking, grounding, usw. beinhalten. Dabei spielt das *grounding criterion* eine entscheidende Rolle: Es stellt ein quantitatives Maß für die Zuverlässigkeit der Information und für die Kosten, sie in die gemeinsame Wissensbasis aufzunehmen, dar. So stellt sich immer die Frage, wie schädlich es für den weiteren Gesprächsverlauf ist, eine bestimmte Information nicht in die gemeinsame Wissensbasis aufzunehmen.

Auch Cahn & Brennan (Cahn, Brennan 1999) beschäftigen sich mit der Frage, wie sich dieses Modell auf den Mensch-Maschine-Dialog übertragen lässt. Dabei benutzen sie jedoch ein sehr eingeschränktes System, in dem der Benutzer den Sprechakt durch Klicken auf einen entsprechenden Button auswählt. Dadurch kann der Benutzer nur genau definierte Nachfragen stellen und es ist kein wirklich freier Dialog, wie er bei dem Küchenroboter gedacht ist.

In ähnlicher Weise versucht auch Bernsen (Bernsen et al. 1996) den Mensch-Maschine-Dialog durch die Definition von über die Gricesche Konversationsmaxime hinausgehenden Prinzipien (Grice 1975) weiter einzuschränken. Beachtet der Dialogdesigner beim Bau der Dialogkomponente diese Prinzipien, so nehmen die Probleme während des Dialogs zwar ab (Dybkjær et al. 1996; Bernsen et al. 1996), lassen sich aber nicht gänzlich vermeiden.

Insgesamt gibt es somit zwar einige Forschungsarbeiten über die Analyse der zwischenmenschlichen Kommunikation und insbesondere über den Prozess des *groundings*, die Einblicke in diese Kommunikation erlauben, aber leider sind diese Analysen eher qualitativer Natur und liefern daher wenig praktische Hinweise, die sich für die Implementierung in der Dialogkomponente eines Robotersystems eignen würden. Darüberhinaus stellt es für ein Dialogsystem eine große Herausforderung dar, auf der einen Seite Spracherkennungsfehler zu kompensieren und auf der anderen Seite kein zu vorsichtiges Grounding mit vermehrten Nachfragen vom Dialogsystem zu verwenden, da dies zu einer sehr unnatürlichen Kommunikation führen würde (Gabsdil 2003).

2.4.3 Klärungs- und Metadialoge zur Auflösung von Fehlern

Metakommunikation wird in zwischenmenschlicher Kommunikation häufig zum Nachfragen oder Reparieren von Äußerungen (Bernsen et al. 1996) verwendet. Dies führt in der Mensch-Maschine-Kommunikation oft zu Problemen, da Dialogsysteme nur schlecht mit Metakommunikation umgehen können, weil sie außerhalb der eigentlichen Domäne des Systems liegt und außerdem das eher starre Schema der Dialoginitiative unterbricht, indem der Benutzer nicht auf eine Klärungsfrage antwortet, sondern selbst eine (vom System nicht erwartete) Nachfrage stellt. Stattdessen stellen die meisten heutigen Dialogsysteme genau definierte Nachfragen an den Benutzer und erwarten eindeutige Antworten darauf. D.h., der unterlegene Dialogpartner, in diesem Fall also die Maschine, kontrolliert momentan in den meisten Dialogsystemen noch den Dialog (cf. engl. *idiot savant*), was mit den in dieser Arbeit entwickelten Fehlerbehandlungsmechanismen in Zukunft vermieden wird.

Auch Perlis und Purang (Perlis et al. 1998) benutzen Metadialoge und Metalogik, um mit Fehlern umgehen zu können. Ihre Theorie basiert auf aktiver Logik (Miller, Perlis 1993) und berücksichtigt damit auch Zeit als einen Fak-

	Selbstreparatur	Fremdreparatur
Selbstinitiierung	Selbstkorrektur	Bitte um Ausdruckshilfe
Fremdinitiierung	Nachfragen	Fremdinitiierte Fremdreparatur

Tabelle 2.2: Fehlerbehebungsstrategien in der zwischenmenschlichen Kommunikation

tor im Gegensatz zu anderen Logiken. Da Informationen, die sich im Diskurs des Dialogmanagers befinden, u.U. später wieder revidiert werden müssen, weil beispielsweise eine falsch erkannte Äußerung vom Benutzer korrigiert wird, ist dieser Zeitfaktor enorm wichtig. Purang (Purang 2001) führt diese Theorie weiter und benutzt aktive Logik als generelle Basis für ein Dialogsystem, um Fehler zu erkennen und zu reparieren.

Um einen natürlicheren Dialog zu erreichen, haben bereits 1977 Schegloff et al. Klärungsdialoge und Reparaturdialoge in zwischenmenschlicher Kommunikation näher untersucht. Sie haben ein Modell dazu aufgestellt und anhand der von ihnen entwickelten ethnomethodologischen Analyse evaluiert (Schegloff et al. 1977). Dieses Modell beschreibt die Mechanismen, die die Dialogpartner nutzen, um mit Fehlern im Dialog umzugehen und diese zu reparieren. Zwei entscheidende Faktoren dabei sind

- die Unterscheidung zwischen selbstinitiiertem und fremdinitiiertem Reparatur
- die Unterscheidung zwischen Selbstreparatur und Fremdreparatur bei der eigentlichen Ausführung der Reparatur

Selbstinitiiert bedeutet dabei, dass der Anstoß von Sprecher A kommt, während bei der Fremdinitiierung die Aufforderung dazu von Sprecher B ausgeht. Typische Fremdinitiierungen sind z.B. Nachfragen. Wie auch in der Tabelle 2.2 dargestellt, gibt es also vier verschiedene Strategien zur Fehlerkorrektur. Allerdings entscheiden allein die Dialogpartner, ob eine Äußerung oder ein Teil einer Äußerung ein Problem darstellt und korrigiert werden muss. Nicht jeder Fehler muss korrigiert werden, wenn er die Kommunikation nicht behindert. Nicht einmal jede Reparatur wird von den Dialogpartnern als die Folge eines Fehlers angesehen.

Durch dieses einfache Modell können Schegloff et al. die plausible Hypothese aufstellen (Schegloff et al. 1977), dass vor der Fremdkorrektur dreimal die Gelegenheit zur Selbstkorrektur gegeben ist, selbst wenn die Initiative dazu vom Gesprächspartner ausgeht. Dabei weisen sie ausdrücklich darauf hin, dass dieses System nur unter gleichgestellten Kommunikationspartnern zu finden ist. Die Bereitschaft zur Fremdkorrektur kann sich signifikant erhöhen, wenn an dem Gespräch jemand teilnimmt, der “not yet competent in some domain” (“noch nicht kompetent in der Domäne”) ist (Schegloff et al. 1977, Seite 54). Zu klären bleibt dabei, inwiefern ein nicht-menschlicher Gesprächspartner wie der Roboter als kompetent bzw. nicht kompetent angesehen wird.

2.4.4 Fehler in zwischenmenschlichen Dialogen mit ungleichen Dialogpartnern

Fehler in zwischenmenschlichen Dialogen und deren Lösung insbesondere bei Dialogen mit ungleichen Gesprächspartnern wie Kindern, Sprachschülern oder Ausländern sind in der vorliegenden Arbeit von besonderem Interesse, weil auch der Mensch-Roboter-Dialog als ein Dialog mit ungleichen Gesprächspartnern angesehen werden kann, wobei ein Gesprächspartner, nämlich der Roboter, weniger kompetent in Grammatik und Vokabular ist. Das Interessante an diesen drei Gruppen ist, dass sie zwar Probleme bei der Kommunikation haben, aber mit Hilfe von metalinguistischen und metakognitiven Fähigkeiten die Kommunikation dennoch aufrechterhalten können (Perlis et al. 1998). D.h., wenn ein Dialog fehlerzuschlagen droht, benutzen Menschen die folgenden Strategien, um dies zu verhindern (Faerch, Kasper 1983; Tarone 1981):

- Problemlösungsstrategien
 - Paraphrasen
 - Restrukturierung des Satzes
 - Annähern bzw. Einkreisen der fehlenden Information
 - Wiederholung
 - Metasprachlicher Transfer
 - nach Hilfe Fragen
- Funktionale Reduktion
 - Funktionale Reduktion der Absicht
 - Antwortvorgabe
 - Themawechsel
- Formale Reduktion
 - Morphosyntaktische Reduktion
- Ratifizierung
 - Zustimmung
 - Wiederholung des Indikators

Problemlösungsstrategien wendet der Benutzer an, indem er die Bedeutung der Äußerung mit weiteren Kommunikationsmitteln erklärt. Beispielsweise formuliert er die Äußerung um oder wechselt die Sprache. Funktionale Reduktionen ändern die ursprüngliche Bedeutung. Der Benutzer reduziert in dem Fall sein kommunikatives Ziel, um weitere Probleme zu verhindern. Formale Reduktion wird durch eine Vereinfachung der Grammatik bzw. des Vokabulars erreicht. Ratifikation dient als Zustimmung zur vorhergehenden Äußerung. Die Strategien können natürlich auch kombiniert werden. Paraphrasen und Umstrukturierungen lassen sich beispielsweise gleichzeitig anwenden. Auch Umstrukturierungen und morphosyntaktische Reduktion sind in der gleichen Äußerung vorstellbar.

Interessant dabei ist, dass diese Strategien unabhängig von der grammatischen Kompetenz in einer bestimmten Sprache benutzt werden. Ähnliche Konzepte finden auch in den hier entwickelten Mechanismen Anwendung (cf. Kapitel 13), so dass das Dialogsystem in der Lage ist, auf den Benutzer einzugehen, der versucht, Fehler zu vermeiden bzw. bereits entstandene Fehler zu beheben.

Von den im vorangegangenen Abschnitt erläuterten Problembehandlungsmechanismen bei Verstehensschwierigkeiten tritt zwischen ungleichen Gesprächspartnern meist eine fremdinitiierte Selbstkorrektur auf (Marti 2001). D.h., der weniger kompetente Gesprächspartner zeigt an, dass er etwas nicht verstanden hat, und der andere Gesprächspartner versucht, eine neue Formulierung dafür zu finden. Auf das Mensch-Roboter-Szenario übertragen bedeutet dies, dass der Roboter dem Menschen anzeigen muss, dass er etwas nicht verstanden hat. Dabei kann es sich um einzelne Worte handeln oder auch um komplexe Zusammenhänge. Bei einzelnen Worten geschieht dies in zwischenmenschlichen Dialogen beispielsweise mit Ausländern durch Wiederholen des unbekanntes Wortes (Bremer 1997). Der menschliche Gesprächspartner versucht dann eine Korrektur bzw. Reparatur vorzunehmen. Die Herausforderung für den Dialogmanager liegt dabei darin, zunächst einmal zu erkennen, dass ein Fehler vorliegt und dann einen Reparaturdialog zu initiieren.

In zwischenmenschlichen Dialogen findet man die folgenden Indikatoren für Nicht-Verstehen (Bremer 1997; Marti 2001):

- Signalisieren von Nichtverstehen
- Erfragen von Zusatzinformationen
- Wiederholung eines zentralen Elements
- Implizite Indikatoren: Ignorieren des Fehlers oder Abbrechen

Diese Indikatoren können auf verschiedenen Ebenen realisiert werden: So kann das Signalisieren von Nichtverstehen auf prosodischer Ebene geschehen, ebenso aber auch durch eine explizite Umformulierung und syntaktische Umstellung des vorangegangenen Satzes. Diese werden im Kapitel 13 auf die Mensch-Roboter-Kommunikation übertragen, um sicherzustellen, dass der Roboter erkennt, dass ein Reparaturdialog begonnen wurde und auch selbst ähnliche Mechanismen verwendet, um einen Reparaturdialog zu beginnen.

Die genaue Analyse von Fehlern in zwischenmenschlichen Dialogen erfolgt mit Mitteln der Konversationsanalyse (Sacks et al. 1974; Atkinson, Heritage 1984), wo verschiedene Arten von Gesprächen im Hinblick auf Regeln und Verfahren untersucht werden, mit denen die Kommunikationspartner ihre Interaktion praktisch gestalten. Eine solche Konversationsanalyse der Fehlerdialoge aus den Benutzertests erfolgt daher im Kapitel 13 über die Fehlerbehebung. Dabei geht es auch darum, inwieweit sich die Kommunikationsstrategien in der zwischenmenschlichen Kommunikation von denen in der Mensch-Maschine-Interaktion unterscheiden, wie dies die Computertalk-Hypothese (Krause, Hitzberger 1992) nahe legt.

2.4.5 Formen der Metakommunikation

Wie schon Perlis und seine Mitarbeiter (Perlis et al. 1998) darlegen, ist ein Dialog erst mit Metakommunikation und Metadialog ein wirklicher Dialog. Ohne solche Fähigkeiten gerät er sehr schnell an seine Grenzen und ist nicht mit der menschlichen Fähigkeit der Konversation zu vergleichen. Generell lassen sich in der Metakommunikation somit die folgenden Mechanismen finden, mit deren Hilfe Probleme in zwischenmenschlichen Dialogen angesprochen und Klärungen vorgenommen werden können:

- Hyperartikulation
- Wiederholung exakt gleicher Äußerungen
- Paraphrasen
- Pragmatische Korrelationen, wie beispielsweise Kontraste, Vergleiche

Hyperartikulation dient dabei, ebenso wie die Wiederholung exakt gleicher Äußerungen, vor allem als Indikator für Nicht-Verstehen. Die Herausforderung für den Dialogmanager liegt hier darin, die Hyperartikulation zu bemerken und die Äußerung korrekt zu erkennen, obwohl dies bei hyperartikulierten Äußerungen besonders schwierig ist (Soltau, Waibel 2000). Bemerkt ein Dialogpartner, dass er nicht verstanden wurde, so formuliert er seine letzte Äußerung neu. Dabei können solche Paraphrasen die letzte Äußerung einerseits vereinfachen oder andererseits auch ausführlicher erklären (Tarone 1981). Pragmatische Korrelationen werden in der zwischenmenschlichen Kommunikation besonders in Dialogen mit Ausländern oder Kindern verwendet, um diesen die Bedeutung der Äußerung noch mehr verdeutlichen zu können (Marti 2001). Das Kapitel 13 geht darauf ein, inwieweit diese Metakommunikationsmechanismen sich auch in der Mensch-Roboter-Kommunikation finden und inwieweit sie sich für eine effizientere Kommunikation ausnutzen lassen.

2.4.6 Weltwissen und Kontextmanagement

Interessant ist darüberhinaus, dass, wie empirische Studien zeigen, auch menschliche Spracherkennung nicht perfekt ist. So haben Shinozaki und Furui (Shinozaki, Furui 2003) in einem Experiment mit einzelnen japanischen Worten herausgefunden, dass zwischen 85% und 92% der Worte von Menschen korrekt erkannt werden konnten, während die Erkennungsraten vom Spracherkennungssystem zwischen 84% und 94% lagen. Damit sind Menschen im Erkennen einzelner Worte nicht erheblich besser als ein Spracherkennungssystem und ihr weitaus besseres Verstehen in alltäglichen Dialogen lässt sich vor allem durch zusätzlichen Nutzen von Weltwissen erklären. Weltwissen bezieht sich dabei auf die folgenden drei Bereiche:

- Syntax: Menschen verifizieren alles, was sie hören, ob es konform zu ihrer Grammatik ist. Wenn es nicht grammatikalisch korrekt sein sollte, so wird das Gesagte erneut analysiert und eine Hypothese für einen grammatikalisch richtigen Satz daraus gebildet. Da dies bereits während des Zuhörens geschieht, kommt es z.T. vor, dass falsche Hypothesen wieder

verworfen werden müssen, weil sich der Satz anders weiterentwickelt, als zunächst angenommen. Dies wird beispielsweise deutlich an den sogenannten “Holzweg-Sätzen” (cf. engl. *garden path sentences*), bei denen Menschen zunächst einen Satz aufgrund des Kontextes falsch parsen und dieses Parsing dann wieder revidieren müssen (Crain, Steedman 1985).

- Semantik: Menschen testen darüberhinaus, ob eine Äußerung semantisch sinnvoll ist. Dafür wird Umweltwissen verwendet.
- Kontext: Der situative Kontext wird benutzt, um herauszufinden, ob eine Äußerung in der gegebenen Situation sinnvoll ist.

Dies verstärkt die Notwendigkeit, zusätzliches Weltwissen im Dialogmanagement zu integrieren, um Fehler, die in der Spracherkennung unvermeidlich sind, im weiteren Dialogverlauf umgehen bzw. einfacher wieder auflösen zu können.

2.5 Fehlermetriken

Als bekannteste Fehlermetrik in der Spracherkennung ist sicherlich die Levenstein-Distanz zu nennen (Levenstein 1996), die bereits im Kapitel 2.3.3 vorgestellt wurde. Allerdings ist diese Fehlermetrik im Bereich der Dialogverarbeitung nicht ausreichend, da sie nur auf Wortebene funktioniert und nicht berücksichtigt, welche Worte für die gesamte Benutzeräußerung bedeutungstragend sind und welche nicht, wie beispielsweise Modalpartikel wie “bitte” o.ä. Daher braucht man hier ein Maß, was angibt, ob der Sinn einer Benutzeräußerung korrekt verstanden wurde. Ein solches Maß ist auf der Turnebene anzugeben und ist die *Turn Error Rate*. Diese gibt die Rate an, wie viele Turns vom System nicht in die richtige Semantik übertragen werden konnten.

Problematisch daran ist, dass eine Äußerung nur falsch oder richtig sein kann, unabhängig davon, wie viel der semantischen Repräsentation richtig erkannt wurde. Im Gegensatz dazu verwendet Hirschberg (Hirschberg et al. 2004) *Konzeptakkuratheit* als ein Fehlermaß. Dabei bedeutet eine Konzeptakkuratheit von 1, dass alle Konzepte korrekt erkannt wurden, unabhängig davon, ob falsch erkannte Worte in der Hypothese enthalten waren. Generell können solche semantischen Maße nicht automatisch ermittelt werden, da dazu die erkannte Äußerung zunächst transkribiert und dann mit ihrer semantischen Repräsentation versehen werden muss. Daher ist die Evaluation von Dialogsystemen sehr aufwendig.

Daneben gibt es noch Fehlermetriken, die sich auf komplexe Benutzerinteraktionen beziehen und zusätzlich die Benutzerzufriedenheit als ein Maß miteinbeziehen (Walker et al. 1997; Behringer et al. 2002). Sie benutzen generelle Performanzfunktionen, die sich aus verschiedenen Maßen, wie Leistung des Benutzers, Anzahl an Turns, Aufgabenlösung, etc. zusammensetzen. Alle diese Maße werden Z-score normalisiert, anhand der Benutzerzufriedenheit gewichtet und so in einem Maß zusammengefasst. Da ein solches Maß jedoch stark von der Domäne und den zu lösenden Aufgaben abhängt, eignet es sich vor allem dazu, zwei Dialogsysteme mit einer ähnlichen Domäne bzw. zwei verschiedene Versionen eines Dialogsystems zu vergleichen.

Außerdem gibt es auch verschiedene Heuristiken, wie z.B. die von Williams et al. in Embassi benutzte (Williams et al. 2002). Das Problem solcher Heuristiken ist immer, daß sie nur Anhaltspunkte liefern können, wo der Dialogdesigner nachschauen könnte, aber keine detaillierten Metriken liefern.

2.6 Zusammenfassung

Als Dialogmanagementsystem soll TAPAS eingesetzt werden (cf. Kapitel 2.2.2), da es den Vorteil bietet, neben einem vielversprechenden informationsbasierten Ansatz in Kombination mit einem rahmenbasierten Ansatz, auch Möglichkeiten zum Rapid Prototyping nutzen zu können. Ferner lässt sich die Integration multimodaler Komponenten durch die multidimensionalen Merkmalsstrukturen gut realisieren.

Betrachtet man den Stand der Forschung im Bereich des Fehlermanagements in Dialogsystemen, so fällt auf, dass es zwar einige interessante Ansätze zur Fehlerbehandlung gibt, diese sich aber meist auf bestimmte Teilbereiche beziehen und stark auf Spracherkennungsfehler fokussieren. Diese Arbeit geht demgegenüber auf Fehler ein, die sich daraus ergeben, dass der Benutzer nur ein begrenztes Wissen über die Möglichkeiten und Fähigkeiten des Systems hat und daher von diesem auch während des Dialogs unterstützt werden muss (cf. Kapitel 13). Außerdem wird hier zunächst eine umfassende Analyse von in Mensch-Roboter-Dialogen auftretenden Fehlern durchgeführt (cf. Kapitel 3, 4, 5, 6), um eine zu starke Einschränkung auf bestimmte Teilbereiche zu vermeiden und die Allgemeingültigkeit der hier entwickelten Fehlerbehandlungsstrategien sicherzustellen.

Darüberhinaus soll der großen Bedeutung von Kontextwissen, das in zwischenmenschlichen Dialogen viel zum Verständnis des Dialogs und der Vermeidung von Fehlern beiträgt, in dem hier vorgestellten Ansatz zur Fehlerbehandlung Rechnung getragen werden. Dabei werden einerseits weitere Wissensquellen und Kontextmanagementmechanismen in das System integriert (cf. Kapitel 9, 10, 11), und auf der anderen Seite versucht das System dem Benutzer mehr Wissen über seinen Kontext zu vermitteln, um das Verständnis des Benutzers zu erhöhen und ein stärkeres Grounding zu ermöglichen (cf. Kapitel 13).

Außerdem haben die Analysen zwischenmenschlicher Dialoge gezeigt, wie wichtig effiziente Fehlerbehebungsstrategien sind. Metakommunikation ist ein essentieller Bestandteil jeder Kommunikation und nur mit ihrer Hilfe lassen sich Fehler in einer für den Benutzer einfachen und nachvollziehbaren Art und Weise auflösen. Daher werden zwischenmenschliche Metakommunikationsstrategien auf die Mensch-Roboter-Kommunikation übertragen, so dass es für den Benutzer einerseits leichter ist, den Roboter auf Fehler aufmerksam zu machen und andererseits auch Fehler zu korrigieren (cf. Kapitel 13).

Dazu wird in den folgenden Kapiteln zunächst untersucht, welche Typen von Fehlern in der Mensch-Roboter-Kommunikation mit welcher Häufigkeit auftreten (cf. Kapitel 3, 4, 5, 6). Mit Hilfe dieser Analyse wird dann eine Komponente entwickelt, die, soweit als möglich, Fehler vermeidet und darüberhinaus Möglichkeiten zur Fehlerbehebung bietet.

Teil II

Benutzerstudien zu Fehlerdialogen

Kapitel 3

Benutzertests in Dialogsystemen

3.1 Einleitung

Für die Entwicklung von Dialogsystemen müssen zunächst einmal die Aufgaben, die das System ausführen kann, das Vokabular und die Sprachstrukturen, die es versteht, bestimmt werden. Dafür werden heutzutage die folgenden Methoden verwendet (McTear 2002):

- Analyse von ähnlichen zwischenmenschlichen Dialogen
- Wizard-of-Oz-Experimente
- Entwicklung von Prototypen

Die Analyse von zwischenmenschlichen Dialogen ist besonders dann von Interesse, wenn es sehr ähnliche Mensch-Mensch-Dialoge gibt, wie z.B. bei Flug- oder Zugauskunftssystemen (Allen et al. 2000b; Stallard 2000) oder bei Anrufweiterleitungssystemen (Gorin et al. 2002). In diesem Fall können die Dialogdaten gleichzeitig mit aufgenommen werden, während der Benutzer mit einem menschlichen Operator spricht, um ihn z.B. nach einem Flugticket zu fragen. Demgegenüber stehen Szenarien, in denen es keine wirklich ähnlichen zwischenmenschlichen Situationen gibt und wo die Aufgaben nicht so gut strukturiert und klar eingegrenzt sind, wie es beispielsweise auch in dem in dieser Arbeit verwendeten Küchenszenario der Fall ist. Hier ist es nur schwer möglich, solche Daten anhand zwischenmenschlicher Dialoge sammeln und hinterher verwenden zu können.

In Wizard-of-Oz-Experimenten simuliert ein Mensch die Rolle des Systems (McTear 2002) und beantwortet die Benutzeranfragen meist mit Hilfe von synthetischer Sprachausgabe. Der Nachteil solcher Experimente liegt darin, dass es für den Menschen, der das System simuliert, sehr schwer ist, sich auch nur annähernd wie ein Roboter zu verhalten, was dazu führt, dass die Daten für eine gegebene Domäne nicht objektiv und damit nur teilweise zuverlässig und repräsentativ sind (McTear 2002).

Darüberhinaus, da sowohl Wizard-of-Oz-Experimente als auch das Sammeln und Analysieren zwischenmenschlicher Dialoge sehr aufwendig sind und damit

nur wenig Daten akquiriert werden können, werden Benutzertests für Dialogsysteme auch häufig mit Hilfe von Prototypen durchgeführt. Zue und seine Mitarbeiter haben beispielsweise zunächst einige Daten mit Hilfe eines Wizard-of-Oz-Experiments gesammelt und dann daraus einen Prototypen entwickelt, der seitdem benutzt wird, um große Mengen an Daten für ihr Wetterinformationssystem (Zue et al. 2000) zu akquirieren. Mittlerweile kann das System zwar nicht mehr als Prototyp bezeichnet werden, aber es werden nach wie vor Daten damit gesammelt, um es auf diese Weise weiterentwickeln zu können.

Daher wurde in der vorliegenden Arbeit ebenfalls ein Prototyp entwickelt, der mit Hilfe ständiger Benutzertests und Datensammlungen weiterentwickelt wird. Die Entwicklung eines Prototypen hat sich auch deshalb angeboten, weil das verwendete Dialogsystem TAPAS sich besonders dafür eignet (cf. Kapitel 2.2.2). Ferner dienen die Benutzertests und Datensammlungen auch zur Evaluierung der neu entwickelten Komponenten zur Fehlervermeidung und -behebung.

3.2 Entwicklung eines Prototypen

Der erste Prototyp enthielt 28 Dialogziele, 160 Dialogmoves und 70 Ontologiekonzepte (für die Erklärung der einzelnen Ressourcen und was sie genau enthalten siehe Abschnitt 2.2.2). Darüberhinaus wurden mehr als 650 Grammatikregeln geschrieben und ein Lexikon mit mehr als 250 Einträgen entwickelt. Diese Prototypgrammatik wurde mit Hilfe erster Benutzertests weiterentwickelt, so dass die erste Version der Roboterapplikation entstand (siehe Tabelle 8.1), die auch für die in den Kapiteln 4 und 5 erläuterten Benutzertests verwendet wurde.

Es wurden mehrere aufeinander aufbauende Benutzerstudien durchgeführt, um einen Überblick über die Fehler, die im Mensch-Roboter-Dialog auftreten, zu gewinnen. Dafür wurde der Küchenroboter bzw. das entsprechende Textinterface zu der Dialogkomponente des Roboters verwendet. Die Benutzer bekamen einfache Aufgaben, die sie mit dem Roboter ausführen sollten, wie beispielsweise den Roboter etwas holen oder bringen, den Tisch decken, Lampen an- bzw. ausschalten, vom Roboter Rezepte erfahren, im Kühlschrank nach den Vorräten sehen, Kaffee oder Tee kochen lassen, usw.

Der Prototyp wurde kontinuierlich weiterentwickelt und an neue Aufgaben angepasst, die der Roboter ausführen soll. Beispielsweise beinhaltet die zweite Version zusätzliche Befehle zur Steuerung der Roboterplattform (Bewegen der Arme, Hände, Finger und der gesamten Plattform, Initialisieren und Starten der einzelnen Bewegungskomponenten, usw.). Hinzu kamen Erkenntnisse aus den verschiedenen im Folgenden näher erläuterten Benutzertests, die ebenfalls integriert wurden. Eine Übersicht über die unterschiedlichen Versionen findet sich im Kapitel 8 in Tabelle 8.1.

3.3 Verschiedene Formen von Benutzertests

Für die multimodalen Benutzertests wurden verschiedene Versionen des Roboters benutzt, wie man sie auf der Abbildung 3.1 sehen kann (für eine detaillierte

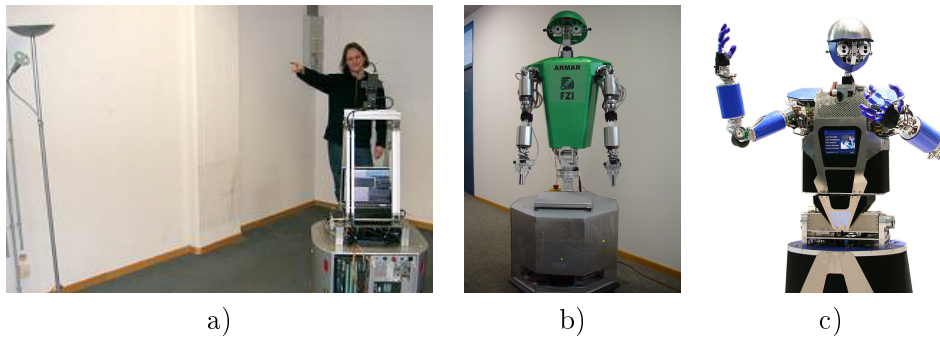


Abbildung 3.1: In den Benutzertests eingesetzte Plattformen und Roboter: Abb.a) Das Entwicklungssystem mit Komponenten zur Spracherkennung, Sprachsynthese, Personen- und Gestentracking, Dialogmanagement und multi-modalen Fusion von Sprache und Gesten. Abb. b) und c): Die beiden humanoiden Roboter Armar II und Armar III, die im SFB 588 in Karlsruhe entwickelt wurden.

Beschreibung der dort verwendeten Hardware siehe Asfour et al. (2006)). Die Benutzer konnten mit dem Roboter mittels Sprache und Zeigegesten interagieren. Dafür bekamen sie ein Nahbesprechungsmikrophon; ferner wurden die Benutzer von Stereokameras auf dem Roboter verfolgt, um so Zeigegesten auflösen zu können (siehe auch Kapitel 11.2).

Da der eigentliche Roboter aus vielen verschiedenen Komponenten besteht, wie beispielsweise dem Spracherkenner, dem Gestenerkenner, dem Dialogmanager, den Bewegungskomponenten, usw. (Gieselmann et al. 2003; Stiefelhagen et al. 2004), wurden die Benutzertests zum Teil auf die Dialogkomponente beschränkt, wofür ein Webinterface mit einer textbasierten Schnittstelle entwickelt wurde (siehe Abbildung 3.2). Auf diese Weise können Segmentierungs- oder Spracherkennungsprobleme oder Probleme, die aus dem falschen Verhalten des Roboters resultieren, zunächst ausgeklammert werden und es ist möglich, den Fokus auf Dialog- und Sprachverstehensprobleme zu legen. Gleichzeitig kann das auch zu Problemen führen, da der Benutzer den Roboter und seine Umgebung nicht sehen kann, sondern nur sprachliche Informationen vom Roboter zurückbekommt, was dieser machen wird. Um sich nicht nur auf die Vorstellungskraft der Benutzer verlassen zu müssen, werden daher im Folgenden immer beide Methoden der Benutzertests verwendet und die Ergebnisse einander gegenübergestellt (siehe Kapitel 6).

Da Benutzertests möglichst realistisch, aber gleichzeitig auch einfach und komfortabel für den Benutzer sein sollten, wurden zwei Arten von Benutzertests durchgeführt:

- Aufgabenbasierte Benutzertests: D.h. der Benutzer bekommt eine oder mehrere konkrete Aufgaben, die er mit dem Roboter lösen soll. Da die Aufgaben vorgegeben sind, ist das insofern unrealistisch, dass der Versuchsleiter immer Aufgaben aussuchen wird, die der Roboter auch lösen kann.
- Benutzertests ohne Aufgaben: Hier wurde die Situation simuliert, dass

Human-Robot-Communication in the Kitchen



You: Good morning

Robbi: Hello! My name is Robbie. I am a little robot who can help you in the kitchen. What do you want me to do?

You: make me a cup of tea

Robbi: Do you want milk?

Talk with me:

no thanks

Send

Abbildung 3.2: Das webbasierte Interface für den humanoiden Roboter

der Benutzer einen neuen Roboter gekauft hat und nun mit ihm allein zu Hause ist und daher machen kann, was er will. Diese Art der freien Interaktion ist für die Benutzer weit schwieriger, weil sie nur eine sehr ungefähre Vorstellung haben, was der Roboter überhaupt kann. Daher bieten sich solche Tests vor allem dann an, wenn das System schon relativ weit entwickelt ist, um den Benutzer nicht zu sehr zu frustrieren.

Wie auch bereits von Dybkjaer und Bernsen (Dybkjær, Bernsen 2000) erläutert, sollten die Szenarien in einem Benutzertest so entworfen sein, dass sie den Benutzer in seiner Interaktion mit dem System nicht beeinflussen. Ein anderes Problem ist, dass die Aufgaben in einem solchen Benutzertest nicht notwendigerweise den Aufgaben entsprechen, die ein Benutzer von sich aus mit dem Roboter durchführen würde. Daher ist es wichtig, neben den aufgabenbasierten Tests auch immer wieder generelle Benutzerbefragungen oder Benutzertests ohne Aufgaben durchzuführen, um sicherzustellen, dass die Abdeckung des Systems den Benutzervorstellungen entspricht.

3.4 Zusammenfassung

Zunächst wurde ein prototypisches Robotersystem entworfen, das als Grundlage für Benutzertests und zur Weiterentwicklung des Systems diente. Dieses System wurde in verschiedenen Entwicklungsstufen sowohl für webbasierte Tests als auch für multimodale Benutzertests mit dem wirklichen Roboter verwendet. Auf diese Weise war es möglich, auf der einen Seite eine große Menge an Daten

zu sammeln (mit Hilfe des Internetttests) und auf der anderen Seite realistische Daten unter Alltagsbedingungen, die nicht in einer Laborumgebung entstanden sind, mit dem wirklichen Roboter zu bekommen.

In den folgenden Kapiteln werden die verschiedenen Benutzertests genauer vorgestellt:

- Multimodale Benutzertests (cf. Kapitel 4)
 - aufgabenbasiert
 - ohne Aufgaben
- Webbasierte Benutzertests (cf. Kapitel 5)
 - aufgabenbasiert
 - ohne Aufgaben

Vergleicht man die aktuell mögliche Mensch-Roboter-Kommunikation mit der zwischenmenschlichen Kommunikation, so fällt auf, dass wir in der zwischenmenschlichen Kommunikation effiziente Strategien haben, um Fehler zu vermeiden und auch, um sie wieder beheben zu können. Dies ist nach wie vor eine große Herausforderung für die Mensch-Roboter-Kommunikation, die mit realistischen Dialogen in alltäglichen Situationen konfrontiert ist und sehr fehlertolerant sein muss. Dabei werden die verschiedenen Ergebnisse der in den folgenden beiden Kapiteln durchgeführten Benutzertests gegenübergestellt, um im Anschluss daran eine Fehlertypologie aufstellen zu können (cf. Kapitel 6). Es wird analysiert, wo Fehler auftreten, wie sie sich klassifizieren lassen, warum sie auftreten und wie sie sich in Zukunft vermeiden lassen. Außerdem erfolgt eine genaue Analyse der Fehlerdialoge (Dialoge, in denen der Benutzer versucht, einen aufgetretenen Fehler zu korrigieren).

Um die Vergleichbarkeit der unterschiedlichen Experimente zu gewährleisten, wurden alle mit dem Spracherkennungsaufgenommenen Daten transkribiert und die Auswertungen erfolgten anhand der transkribierten Daten. Alle Benutzertests fanden mit einem englischen System statt. Daher sind alle Beispiele der Benutzeräußerungen im englischen Original mit ihrer deutschen Übersetzung wiedergegeben, um dem Leser eine bessere Vorstellung der benutzten syntaktischen Konstruktionen zu geben. Die Instruktionen und die Fragebögen für die Versuchspersonen sind in dieser Ausarbeitung zum besseren Verständnis in deutscher Übersetzung angegeben.

Zur Auswertung der folgenden Benutzertests werden Parsing- und Turnfehlerrate ermittelt. Beide basieren auf einem *Turn* als grundlegender Einheit, wobei ein *Turn* definiert ist als eine Benutzeräußerung. Diese kann aus mehreren Sätzen oder auch nur aus einem Teilsatz bestehen. Die Parsingrate gibt den Anteil der Benutzeräußerungen an, die vom Dialogsystem geparkt werden können:

$$\text{Parsingrate} = \frac{\text{Anzahl geparkter Turns}}{\text{Gesamtzahl Turns}} \times 100$$

Je höher die Parsingrate ist, desto mehr Sätze können erfolgreich geparkt werden. Demgegenüber gibt die Turnfehlerrate den Anteil der Benutzeräußerungen

an, die vom Dialogsystem nicht in die intendierte Semantik des Benutzers übertragen werden können:

$$\textit{Turnfehlerrate} = \frac{\textit{Anzahl fehlerhafter Turns}}{\textit{Gesamtzahl Turns}} \times 100$$

Je höher die Turnfehlerrate ist, desto mehr Sätze können nicht verstanden werden und desto schlechter funktioniert damit die Interaktion. Alle als fehlerhaft klassifizierten Benutzerturns werden dann anhand der Gründe, warum keine korrekte semantische Repräsentation aufgebaut werden konnte, klassifiziert. Diese Fehlerklassifikation dient als Grundlage der im Teil III entwickelten Fehlerbehandlungsmethoden. Ferner wird die durchschnittliche Anzahl an Turns, die die Benutzer benötigen, um eine Aufgabe zu erledigen, berechnet, und als ein Maß für den Erfolg der Benutzerinteraktion herangezogen:

$$\textit{Turns pro Aufgabe} = \frac{\textit{Anzahl Turns}}{\textit{Anzahl gelöster Aufgaben}}$$

Kapitel 4

Multimodale Benutzertests

4.1 Vorgehensweise

Der multimodale Benutzertest wurde mit vier Benutzern mit Hilfe des bereits in Kapitel 3 erläuterten Roboters durchgeführt. Die Benutzer wurden zunächst eingewiesen und es wurde ihnen erklärt, wie Headset und Mikrofon funktionieren und dass sie mit dem Roboter wie mit einem menschlichen Gesprächspartner reden können, wobei auch Zeigegesten verwendet werden dürfen. Dann wurde den Benutzern das Küchenszenario und die folgenden Fähigkeiten des Roboters erklärt:

- Einfache Hol- und Bringdienste
- An- und Ausschalten von Lichtern
- Informationen zu einigen Rezepten
- Informationen über den Inhalt des Kühlschranks

Zunächst haben die Benutzer keine Aufgaben bekommen, sondern konnten mit dem Roboter in dem oben beschriebenen Küchenszenario frei interagieren. Im Anschluss mussten sie zwei konkrete Aufgaben lösen:

1. Holen einer Tasse
2. Anschalten einer Lampe

Die Benutzer wurden aufgefordert, jede Aufgabe mehrmals durchzuführen, um verschiedene Interaktionsmöglichkeiten ausprobieren zu können.

4.2 Ergebnisse des Experimentteils ohne Aufgaben

4.2.1 Generelle Ergebnisse

Allen Benutzern gelang es, den Roboter einige Aufgaben ausführen zu lassen (durchschnittlich ca. 8 verschiedene Aufgaben, siehe auch Tabelle 4.1). Dies ist ein sehr gutes Ergebnis angesichts der Tatsache, dass die Benutzer keine

	Mittelwert
Gelöste Aufgaben	8,5
Gesamtzahl Turns	32
Turns pro Aufgabe	3,76
Turnfehlerrate	57,03 %
Parsingrate	69,53%

Tabelle 4.1: Ergebnisse des Benutzertests ohne vorgegebene Aufgaben

konkreten Aufgaben vorgegeben hatten, sondern nur generell wussten, was der Roboter kann. Die durchschnittliche Anzahl an Turns pro Aufgabe von 3,76 zeigt ferner, dass die Benutzer die Aufgaben in einer angemessenen Zeit (weniger als vier Turns) lösen können. Die Benutzer waren also in der Lage, mit dem Roboter zu interagieren und einige Aufgaben erfolgreich zu lösen, obwohl sie nur generelle Informationen über ihn hatten.

So fanden die meisten Benutzer auch Gefallen an der Kommunikation mit dem Roboter, denn sie interagierten mit ihm durchschnittlich etwa 32 Turns. Die Turnfehlerrate ist mit 57.03% verglichen mit Fehlerraten in anderen Dialogsystemen relativ hoch (siehe auch Kapitel 2.3.4), was einerseits darauf zurückzuführen ist, dass es sich bei diesem System um die erste Version des Prototypen handelte, deren Grammatikabdeckung noch recht gering war. Andererseits ist die Küchendomäne nicht so begrenzt wie beispielsweise die Reiseauskunftsdomäne, so dass sich dadurch das Problem ergibt, dass der Benutzer eine andere Vorstellung von den Fähigkeiten des Systems hat, als dass tatsächlich der Fall ist.

Im Folgenden werden alle die Äußerungen, die nicht korrekt verstanden wurden und zu Fehlern führten, näher beleuchtet. Dafür werden alle Äußerungen anhand der Gründe, warum sie nicht verstanden werden konnten, klassifiziert (siehe auch Tabelle 4.2).

4.2.2 Detaillierte Fehleranalyse und -klassifikation

Der Hauptgrund für Fehler lag hier in neuen semantischen und syntaktischen Konzepten (cf. Tabelle 4.2). Die Versuchspersonen verwendeten neue Formulierungen, wie beispielsweise "Hello mister robot!" ("Hallo Herr Roboter!"), um den Roboter zu begrüßen, die noch nicht von der prototypischen Grammatik abgedeckt waren. Auch einige wenige neue Objekte, wie z.B. "washing machine" ("Waschmaschine"), wurden von den Benutzern gebraucht. Die geringe Rate an neuen Objekten liegt sicherlich daran, dass die meisten Versuchspersonen nur auf die Objekte referiert haben, die sie auch in der Küche sehen konnten und sich wenig neue Objekte ausgedacht haben (im Gegensatz zu den im folgenden Kapitel dargestellten Tests über das Internet).

Einige Benutzer verwendeten ferner neue Aufgaben, die der Roboter momentan nicht in seiner Grammatik hat und daher nicht versteht, wie beispielsweise "Boil some milk please." ("Erhitze etwas Milch."). Der größte Teil der neuen Dialogziele liegt jedoch mit etwa 30% aller fehlerhaften Äußerungen im

	Anteil an allen Äußerungen	Anteil an nicht verstandenen Äußerungen
Neue Konzepte	42,19%	73,98%
Neue Formulierung	18,75%	32,88%
Neues Objekt	3,91%	6,85%
Neues Dialogziel	2,34%	4,11%
Metakommunikation	17,19%	30,14%
Ellipsen	4,69%	8,22%
Elliptische Äußerungen & Anaphern	4,69%	8,22%
Komplexität	1,56%	2,74%
Verbundene Äußerungen	1,56%	2,74%
Eingabeprobleme	8,59%	15,06%
Segmentierungsprobleme	7,81%	13,70%
Spontane Sprache	0,78%	1,36%

Tabelle 4.2: Fehleranalyse der aufgetretenen Fehler im Experimentteil ohne vorgegebene Aufgaben

Bereich der Metakommunikation. D.h., der Sprecher erläutert oder modifiziert durch unmittelbaren Bezug auf vorhergehende Äußerungen diese durch Korrektur, Präzisierung, Stellungnahme, Kommentar, usw. Hierzu zählen Äußerungen, wie “You just told me there was some milk in the refrigerator - is that right?” (“Du hast gerade gesagt, dass es Milch im Kühlschrank gibt - stimmt das?”), “Did you understand me?” (“Hast du mich verstanden?”), “Robbi, is it orange juice or apple juice?” (“Robbi, meinst du Orangen- oder Apfelsaft?”), etc. Auf der Metakommunikationsebene geht es darum, dass der Benutzer auf etwas vorher Gesagtes zurückkommt und eine Nachfrage dazu stellt, dieses näher spezifiziert bzw. korrigiert. Eine klassische Nachfrage auf dieser Ebene ist also “Hast du mich verstanden?”, die meist auf eine längere Pause vom Roboter, in der dieser nichts äußerte, folgte. Beim anderen Beispiel “Robbi, meinst du Orangen- oder Apfelsaft?” hat der Roboter dem Benutzer gerade gesagt, dass er ihm jetzt Saft bringen wird, da der Benutzer aber explizit Apfel- und nicht Orangensaft möchte, fragt er jetzt nochmals nach, um sicherzugehen, dass der Roboter ihn korrekt verstanden hat. Eine solche Komponente, die speziell mit Metakommunikation umgehen kann, wird in dieser Arbeit entwickelt (siehe Kapitel 13), da sich damit sehr viele Fehler vermeiden lassen (cf. Tabelle 4.2). Ferner stellt Metakommunikation eine besondere Herausforderung für Dialogsysteme dar, da der Benutzer selbst die Initiative ergreift und etwas sagt, was nicht unmittelbar auf die letzte Äußerung des Dialogsystems passt. Damit dient Metakommunikation meist zum Beheben von Fehlern.

Ferner verwendeten einige Benutzer elliptische Äußerungen und Anaphern, wie beispielsweise “To the fridge!” (“Zum Kühlschrank!”), nachdem der Roboter sich nicht gleich in Bewegung setzte, um die vom Benutzer gewünschten Tomaten zu holen. “To the fridge!” allein ist aber keine vollständige Äußerung,

sondern eine Ellipse, da der Sprechakt fehlt. Daher werden für den Umgang mit Ellipsen in dieser Arbeit entsprechende Mechanismen entwickelt, wie im Kapitel 11 beschrieben.

Daneben gab es auch einige wenige Äußerungen, die zu komplex waren, um korrekt verstanden zu werden, wie beispielsweise “the blue fork and the green knife” (“die blaue Gabel und das grüne Messer”) als Antwort auf die Nachfrage des Roboters, was er denn holen solle. Da die Anzahl an miteinander verbundenen Elementen theoretisch unendlich ist, gilt es hier, ein gutes Maß zu finden, um möglichst viele Benutzeräußerungen parsen zu können, ohne zu komplexe grammatikalische Konstrukte aufzubauen (siehe auch Kapitel 12).

Darüberhinaus gab es eine ganze Reihe von Äußerungen mit Eingabeproblemen. Dazu zählen zunächst die Äußerungen, die vorne bzw. hinten abgeschnitten sind, weil die automatische Sprachsegmentierung eine Äußerung zu früh abgeschnitten bzw. zu spät aufgenommen hat. Wenn das Sprachsignal also am Anfang einer Äußerung noch zu schwach ist oder während einer Äußerung zu schwach wird, weil der Benutzer eine Pause macht oder einfach leiser wird, kann es passieren, dass die Äußerung erst ab einem späteren Zeitpunkt aufgenommen bzw. dort abgeschnitten wird. Außerdem finden sich einige wenige spontansprachliche Phänomene, wie Satzabbrüche und Reformulierungen, in dem Benutzertest, wie z.B. “Robbi can you give me äh some a glass of water please.” (“Robbi, kannst du mir äh einige ein Glas Wasser geben, bitte?”), die nicht korrekt erkannt werden konnten. Insgesamt führten nur sehr wenige dieser Phänomene zu Fehlern, da der Spracherkennung bereits mit Füllwörter wie “äh”, “hmm”, usw. an den verschiedensten Positionen im Satz umgehen kann (Gieselmann et al. 2003). Problematisch ist in diesem Fall die Reformulierung, die mit “some a glass” (“einige ein Glas”) dazu führte, dass der Satz ungrammatikalisch wurde und damit nicht mehr von der Grammatik abgedeckt war. Insgesamt sind dies jedoch Fehler, die eher im Bereich der Spracherkennung bzw. -segmentierung zu lösen sind, weshalb die vorliegende Arbeit nicht weiter auf diesen Fehlertyp eingehen wird.

4.2.3 Benutzerverhalten nach dem Auftreten von Problemen und Fehlern

Die meisten Benutzer sind nicht sehr hartnäckig, wenn sie wie hier keine klar vorgegebenen Aufgaben haben, sondern schwenken recht schnell von einer Aufgabe zur nächsten, wenn der Roboter sie nicht gleich versteht. Daher finden sich in dem Experimentteil ohne Aufgaben kaum wirkliche Fehlerdialoge, in denen die Benutzer versuchen, dem Roboter etwas Bestimmtes klar zu machen.

Nur wenige Benutzer versuchten, einmal nicht verstandene Äußerungen umzuformulieren. So sagte ein Benutzer z.B. “What kind of juice do we have?” (“Was für Sorten Saft haben wir?”). Als der Roboter dies nicht verstand, versuchte es der Benutzer mit “Robbi, is it orange juice or apple juice?” (“Robbi, ist das Orangensaft oder Apfelsaft?”). Als der Roboter dies auch nicht verstand, entschloss sich der Benutzer, zu einer anderen Aufgabe überzugehen: “Robbi, I would like a cup of coffee please.” (“Robbi, ich hätte gern eine Tasse Kaffee.”).

	Mittelwert
Gelöste Aufgaben	28
Gesamtzahl Turns	95,25
Turns pro Aufgabe	3,40
Turnfehlerrate	22,57%
Parsingrate	89,32%

Tabelle 4.3: Ergebnisse des Benutzertests mit Aufgaben

4.3 Ergebnisse für den aufgabenbasierten Teil

4.3.1 Generelle Ergebnisse

Allen Benutzern gelang es, den Roboter die vorgegebenen Aufgaben mehrmals ausführen zu lassen. Da die Aufgaben relativ einfach waren und meist in ein bis zwei Turns von den Benutzern gelöst werden konnten, schafften es die Benutzer durchschnittlich sogar 28 Mal (siehe auch Tabelle 4.3). Die durchschnittliche Anzahl an Turns pro Aufgabe ist mit 3,4 etwas niedriger als bei dem Test ohne vorgegebene Aufgaben, wo sie bei 3,76 lag.

Die Turnfehlerrate ist mit 22.57% hier sehr niedrig. Dies liegt wahrscheinlich an den relativ einfachen Aufgaben, in denen es nicht viel Spielraum für die Benutzer gab. Dies erklärt auch die hohe Anzahl an Turns (durchschnittlich 95 Turns pro Benutzer): Die meisten Benutzer freuten sich, dass die Kommunikation so gut klappte und schickten den Roboter gleich wieder los, um eine weitere Tasse zu holen bzw. noch eine Lampe anzuschalten.

4.3.2 Detaillierte Fehleranalyse und -klassifikation

Die Hauptursache für Fehler lag auch hier in neuen syntaktischen oder semantischen Konzepten (cf. Tabelle 4.4), dicht gefolgt von Problemen mit Ellipsen und Eingabeproblemen.

Es gab einige neue Formulierungen, wie z.B. “Now could you turn on the big lamp one more time?” (“Kannst du jetzt die große Lampe nochmal anschalten?”), was daran liegen könnte, dass die Prototypgrammatik noch nicht genügend Formulierungen abdeckt. Es wäre möglich, dass eine größere Grammatik zu weniger Fehlern in dem Bereich führt, was in weiteren Studien untersucht wird (siehe auch Kapitel 8).

Die Benutzer stellten sehr wenige neue Aufgaben, da sie bereits Aufgaben vorgegeben hatten. Nur ein Benutzer probierte beispielsweise “Wash the dishes.” (“Mach den Abwasch.”) aus. Allerdings gab es auch hier wieder einige neue Aufgaben im Bereich der Metakommunikation, um falsch verstandene Information wieder zurücknehmen zu können, wie z.B. “No, no, Robbi, the orange cup.” (“Nein, nein, Robbi, den orangenen Becher.”). Gerade diese Korrektur von falsch verstandenen Informationen spielt eine große Rolle bei der Fehlerkorrektur und wird daher ausführlich in Kapitel 13 behandelt.

Neue Objekte gab es keine, was sich dadurch erklären lässt, dass den Ver-

	Anteil an allen Äußerungen	Anteil an nicht verstandenen Äußerungen
Neue Konzepte	8,03%	36,04%
Neue Formulierung	5,77%	25,58%
Neues Objekt	0%	0%
Neues Dialogziel	1,05%	4,65%
Metakommunikation	1,31%	5,81%
Ellipsen	7,09%	31,40%
Elliptische Äußerungen & Anaphern	5,25%	23,26%
Fehlender Kontext	1,84%	8,14%
Komplexität	0,26%	1,16%
Verbundene Äußerungen	0,26%	1,16%
Eingabeprobleme	7,08%	31,40%
Segmentierungsprobleme	6,56%	29,07%
Spontane Sprache	0,52%	2,33%

Tabelle 4.4: Fehleranalyse der aufgetretenen Fehler im aufgabenbasierten Teil

suchspersonen vorher die Lampen und die Becher, die sie in diesen Aufgaben verwenden sollten, gezeigt wurden, und sie daher auch auf keine weiteren Objekte referiert haben.

Fast ein Viertel aller fehlerhaften Äußerungen sind elliptische oder anaphorische Äußerungen, wie z.B. “the little one” (“die kleine”) als Antwort auf die Frage des Roboters, welche Lampe er denn anschalten soll. Hierfür ist es nötig, den Kontext im Dialogmanager zu halten, um solche Äußerungen auflösen zu können (siehe auch Kapitel 11).

Auch hier gibt es wieder einige wenige Äußerungen, die zu komplex sind, wie z.B. “Bring me the eggs, the milk and the noodles, please.” (“Bring mir Eier, Milch und Nudeln, bitte.”). Hier gilt es, wie bereits im vorangegangenen Abschnitt erläutert, eine Möglichkeit zu finden, mit einer sinnvollen Anzahl an verbundenen Elementen umgehen zu können (siehe auch Kapitel 12).

Ferner gab es wieder einige Segmentierungsfehler, die dazu führten, dass die Äußerungen zu früh abgeschnitten bzw. zu spät aufgenommen wurden, und ein paar spontansprachliche Phänomene. Diese sind beide den Eingabeproblemen zugeordnet worden.

4.3.3 Benutzerverhalten nach dem Auftreten von Problemen und Fehlern

Im Gegensatz zum Experimentteil ohne Aufgaben versuchten die Benutzer hier, wenn ein Fehler aufgetreten ist, den Roboter trotzdem dazu zu bringen, die angefangene Aufgabe zu lösen. Dabei wurden z.T. Reformulierungen verwendet, wie beispielsweise “Please switch on the lamp to my right.” (“Bitte schalte die Lampe zu meiner Rechten an.”) und dann “The lamp I am pointing to I want you to switch it on.” (“Die Lampe, auf die ich zeige, schalte sie bitte an.”). Einige

	Ohne Aufgaben	Mit Aufgaben
Turnfehlerrate	57,03%	22,57%
Gelöste Aufgaben	8,5	28
Gesamtzahl Turns	32	95,25
Neue Konzepte	73,98%	36,04%
Ellipsen	8,22%	31,40%
Komplexität	2,74%	1,16%
Eingabeprobleme	15,06%	31,40%

Tabelle 4.5: Vergleich der aufgetretenen Fehlertypen im Teil mit vs. ohne Aufgaben (Angabe sind hier nur die Raten der jeweiligen Fehlertypen an der Gesamtzahl fehlerhafter Äußerungen)

Benutzer wiederholten auch den gleichen Satz mehrfach.

Insgesamt scheinen die hier gewählten Aufgaben recht einfach zu sein, wie auch an der sehr niedrigen Turnfehlerrate deutlich wird, denn den Benutzern gelang es meist, zumindest beim zweiten Mal den Roboter zur Lösung der Aufgabe zu bringen. Dadurch gibt es nur relativ wenige Fehlerdialoge, die hier ausgewertet werden können.

4.4 Zusammenfassung

Vergleicht man die Ergebnisse der beiden Teile des Benutzertests (mit vs. ohne Aufgaben), so lässt sich feststellen, dass die gleichen Fehlertypen auftreten, aber die Verteilung dieser Fehlertypen bei dem aufgabenbasierten Teil sehr viel ausgeglichener ist, während bei dem Teil ohne Aufgaben ca. 74% der Probleme darauf beruhten, dass die Benutzer sowohl neue semantische als auch neue syntaktische Konzepte verwendet haben (siehe auch Tabelle 4.5). Das lässt sich dadurch erklären, dass die Benutzer ohne vorgegebene Aufgaben eine recht unterschiedliche Vorstellung davon hatten, was der Roboter für sie machen kann.

Insgesamt ist aber die Rangfolge der Fehlertypen in der Gesamtmenge der Fehler in beiden Experimentteilen gleich, auch wenn die Unterschiede im aufgabenbasierten Teil stärker ausgeprägt sind:

1. Neue Konzepte
2. Eingabefehler
3. Ellipsen
4. Komplexe Äußerungen

D.h., die meisten Fehler finden sich im Bereich der neuen syntaktischen und semantischen Konzepte (73,98% im Experimentteil ohne Aufgaben und 36,04% im aufgabenbasierten Experimentteil) und die wenigsten Fehler bei den komplexen Äußerungen (2,74% im Experimentteil ohne Aufgaben und 1,16% im aufgabenbasierten Experimentteil).

Die großen Unterschiede in der Turnfehlerrate, die ohne Aufgaben mit 57,03% gegenüber nur 22,57% im aufgabenbasierten Experimentteil mehr als doppelt so hoch ist, lässt sich ebenfalls auf die verschiedenen Vorstellungen der Benutzer über die Fähigkeiten des Roboters ohne vorgegebene Aufgaben zurückführen. Hinzu kommt, dass der Aufgabenteil mit Aufgaben immer der zweite Teil des Experiments war und die Benutzer sich daher an den Roboter und das Kommunizieren mit ihm gewöhnt hatten, so dass dies auch ein Grund für ihren größeren Erfolg sein könnte. Ferner waren die Aufgaben recht einfach, was ebenfalls zu einer geringeren Fehlerrate beigetragen haben mag.

In einem kurzen Nachinterview gaben die Benutzer an, dass ihnen die aufgabenbasierten Tests viel leichter gefallen sind. Dies wird auch beispielsweise an der höheren Gesamtzahl an Turns deutlich (durchschnittlich 95,25 Turns im aufgabenbasierten Experimentteil gegenüber 32 Turns im Experimentteil ohne Aufgaben), da die Benutzer im Experimentteil ohne Aufgaben irgendwann nicht mehr wussten, was sie noch ausprobieren sollten und darum eher aufgehört haben.

Ein solcher Benutzertest mit einem wirklichen Roboter ist sehr aufwendig - sowohl für den Versuchsleiter als auch für die Versuchspersonen. Ferner treten viele Fehler auf, die sich nur indirekt auf den Dialog beziehen und mit seiner Hilfe wieder gelöst werden müssen, wie beispielsweise Segmentierungs- und Spracherkennungsfehler, Probleme mit anderen Roboterkomponenten, usw. Hinzu kommt, dass der Roboter bisher nur eine sehr begrenzte Zahl an Aufgaben wirklich ausführen kann, was sich zur Zeit auf recht einfache Bringdienste beschränkt, wodurch es schwierig wird, interessante Aufgaben für einen Benutzertest zu finden.

Da sich diese Arbeit verstärkt mit semantischen Fehlern im Dialogbereich beschäftigt, werden im folgenden Kapitel Internetttests vorgestellt, mit denen es zum einen möglich ist, schneller mehr Benutzerdaten zu gewinnen, interessantere Aufgaben zu verwenden und sich zum anderen auch auf semantische Fehler im Dialogbereich zu konzentrieren.

Kapitel 5

Webbasierte Benutzertests

5.1 Einleitung

Der Vorteil eines webbasierten Benutzertests liegt darin, dass viele Benutzer überall auf der Welt teilnehmen können, wann immer sie wollen. Daher sind die Kosten, was Zeit und Geld angeht, geringer als in herkömmlichen Benutzerstudien (Schmidt 1997). Auch Reips (Reips 2002) sieht die Vorteile von webbasierten Experimenten in ihrer Geschwindigkeit, den geringen Kosten, der Möglichkeit, rund um die Uhr Experimente laufen zu lassen, und einem hohen Grad an Automatisierung.

Ein Problem bei webbasierten Experimenten liegt in der Tatsache begründet, dass kein Versuchsleiter bei den Benutzern ist, um ihnen, wenn sie etwas nicht verstanden haben, helfen zu können, so dass die Versuchspersonen ihre eigenen Schlüsse ziehen, ohne nachfragen zu können (Reips 2002). Um diesen Nachteil zu vermeiden, wurde ein Vortest durchgeführt (cf. Abschnitt 5.2), um eventuelle Probleme und Schwachstellen im Design des Experiments frühzeitig erkennen zu können.

Ein anderer Nachteil webbasierter Benutzertests ist, dass die Benutzer aufhören, wenn sie keine Lust mehr haben, weil kein Versuchsleiter da ist, der dafür sorgt, dass sie das Experiment zu Ende machen (Reips 2002). Dies stellt im Fall des Küchenroboters jedoch keinen Nachteil dar, denn es entspricht eher der realen Situation, wenn die Benutzer den Roboter zu Hause haben und selbst entscheiden können, ob sie ihn jetzt verwenden wollen oder nicht. Daher wurden all die Situationen, in denen die Benutzer frühzeitig abgebrochen haben, analysiert, um sie in Zukunft vermeiden zu können.

5.2 Vortest

5.2.1 Benutzerfragebogen

Als Vortest wurde ein Benutzerfragebogen (siehe Abbildung 5.1) entworfen, um einerseits Informationen von den Benutzern über ihre generellen Erwartungen an einen solchen Haushaltsroboter und andererseits auch spezifische Informationen, wie die Benutzer den Roboter bitten würden, eine gegebene Aufgabe

1. Sie kommen vom Einkaufen nach Hause und haben einen neuen Robbi 2006 gekauft, der Ihnen in der Küche helfen kann. Was ist das erste, was Sie ihn machen lassen?
2. Für welche Art von Aufgaben hätten Sie gern Hilfe durch einen Roboter?
3. Für welche Art von Aufgaben würden Sie nicht gern die Hilfe eines Roboters nutzen?
4. Welche Rezepte sollte der Roboter für Sie kennen?
5. Zum Abendessen haben Sie ihren besten Freund eingeladen. Welche Anweisungen geben Sie Ihrem Roboter?
6. Nach der langen Einkaufstour sind Sie durstig. Wie fragen Sie Robbi um Hilfe?
7. Sie sitzen vor Ihrem Rechner und würden gern etwas trinken und ein paar Kekse essen. Bitten Sie Robbi, Ihnen dies zu bringen!
8. Sie wollen Lebensmittel einkaufen, aber wissen nicht, was noch im Kühlschrank ist. Da Sie keine Lust haben, selbst aufzustehen und nachzusehen, fragen Sie Robbi um Hilfe.
9. Sie sehen den großen, roten, runden Becher auf dem Tisch in der rechten Ecke des Wohnzimmers. Bitten Sie Robbi, ihn zu holen.

Abbildung 5.1: Benutzerfragebogen für den Vortest im Internet

auszuführen, zu bekommen. Der Fragebogen wurde von insgesamt 24 Versuchspersonen ausgefüllt, die mit dem Roboter und seinen Fähigkeiten nicht vertraut waren. Die im Folgenden näher erläuterten Ergebnisse zeigen, dass die Benutzer generell von sehr ähnliche Konzepten der Aufgaben des Roboters ausgehen, die auch bei der Entwicklung des Prototypen verwendet wurden, so dass im eigentlichen Test keine größeren Probleme zu erwarten sind.

5.2.2 Auswertung der generellen Fragen

Für die generellen Fragen (Fragen 1-5 des Fragebogens in Abbildung 5.1), wurde zunächst überprüft, ob die Aufgaben, die die Benutzer vom Roboter erledigen lassen wollen, im hier entwickelten Domänenmodell vorkommen. Tabelle 5.1 gibt auf der einen Seite die Aufgaben an, die die Benutzer für gut geeignet für einen Roboter halten, und auf der anderen Seite die, die sie nicht von einem Roboter ausführen lassen wollen. Da die Versuchspersonen gebeten wurden, so viele Aufgaben wie möglich zu nennen, und damit Mehrfachnennungen häufig vorkommen, bezieht sich die angegebene Rate der genannten Aufgaben immer auf die Gesamtzahl der von allen Versuchspersonen genannten Aufgaben.

Dabei stellte sich heraus, dass die Benutzer sehr ähnliche Aufgaben genannt

Aufgaben für den Roboter	Gen. von	Nicht für den Roboter geeignete Aufgaben	Gen. von
Sauber machen	28,57%	Alle Aufgaben können vom Roboter gemacht werden	37,50%
Geschirr waschen	17,14%	Kochen	20,83%
Essen vorbereiten	14,29%	Essen abschmecken	16,67%
Einkaufsplanung	10,00%	Gespräche	8,33%
Aufräumen	8,57%	Schnelle Aufgaben, die der Benutzer lieber selbst macht	8,33%
Essen kochen	7,14%	Aufräumen	4,17%
Getränke holen	5,71%	Einkaufsplanung	4,17%
Einkaufen	2,86%		
Essensvorschläge	2,86%		
Andere Aufgaben	2,86%		

Tabelle 5.1: Auswertung der Benutzerbefragung: Aufgaben, die für den Roboter gut geeignet sind, vs. Aufgaben, die für ihn nicht geeignet sind. Sortiert nach der Häufigkeit, in der sie von den Benutzern genannt wurden.

haben, wie die, die bereits im Dialogmodell abgedeckt sind: Viele Benutzer wollen den Roboter Geschirr abwaschen (17,14%), Getränke holen (5,71%), kochen (7,14%) oder zumindest Essen vorbereiten (14,29%) lassen. Diese Aufgaben sind bereits im Prototyp abgedeckt. Nur Sauber machen (28,57%), was ebenfalls von sehr vielen Versuchspersonen genannt wurde, ist momentan nicht implementiert angesichts der Tatsache, dass der Roboter auch später nicht wirklich zum Sauber machen vorgesehen ist. Ebenso sind Einkaufen (2,86%) und die Einkaufsplanung (10,0%), bei der es um die Kontrolle der vorhandenen Vorräte und das rechtzeitige Wiederauffüllen geht, nicht im Prototyp enthalten, da es zu viele mögliche Formulierungsvarianten für diese Aufgaben gibt, die nicht alle von einem solchen Prototyp abgedeckt werden können. Hinzu kommt, dass einige Benutzer (4,17%) im Gegenteil sogar der Meinung sind, dass sie die Einkaufsplanung lieber selbst machen wollen, um einen besseren Überblick zu haben (cf. Tabelle 5.1).

Die Meinungen der Benutzer zu den Aufgaben, die sie nicht von einem Roboter erledigen lassen wollen, gehen sehr stark auseinander: Viele denken, dass alles von einem solchen Roboter gemacht werden kann (37,50%). Demgegenüber gibt es einige Versuchspersonen, die der Meinung sind, dass der Roboter nicht kochen (20,83%) oder zumindest die Gerichte nicht abschmecken (16,67%) sollte, weil sie glauben, dass er dazu nicht in der Lage ist. Andere (8,33%) wiederum vertreten die Ansicht, dass Gespräche eine rein menschliche Domäne sind, was angesichts der immer größer werdenden Zahl an Chatterbots im Internet, die nur für solche Gespräche da sind, interessant ist.

Ferner haben die Versuchspersonen, wenn sie gebeten wurden, sich vorzustellen, mit ihrem gerade neu erworbenen Roboter nach Hause zu kommen, an ähnliche Aufgaben gedacht, wie die, die sich generell gut für einen Roboter eignen (siehe auch Abbildung 5.2). Etwa 20% der Benutzer kamen noch zusätzlich

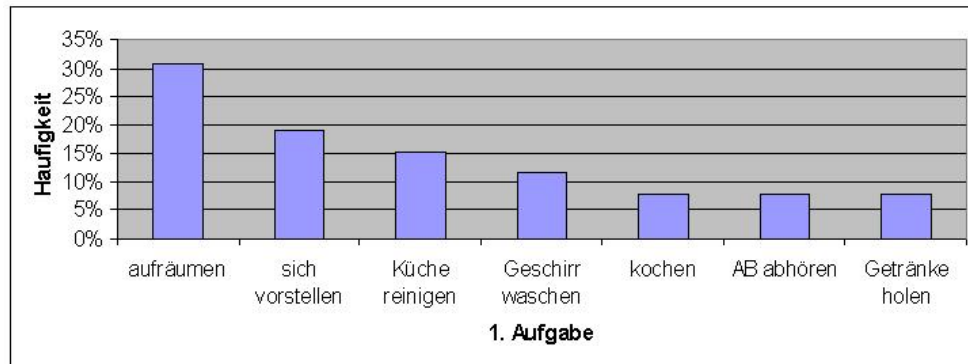


Abbildung 5.2: Auswertung der Benutzerbefragung zur ersten Aufgabe für den Roboter sortiert nach der Häufigkeit, in der sie von den Benutzern genannt wurden.

auf den Gedanken, sich dem Roboter vorzustellen und auf diese Weise von ihm auch Informationen über seine Fähigkeiten zu bekommen. Dies ist ebenfalls bereits in der ersten Version des Prototypen abgedeckt.

Auch die meisten Rezepte, die die Benutzer vom Roboter wissen wollten, sind bereits in der entwickelten Grammatik abgedeckt, wie beispielsweise Nudelgerichte (15,15%) und Salat (15,15%)(cf. Abbildung 5.3). Allerdings gibt es auch einige Versuchspersonen, die meinen, dass der Roboter einfach alle Rezepte (12,12%) kennen sollte, wofür es sich anbieten würde, eine große Rezept-Datenbank aus dem Internet anzubinden (Experimente zur Integration einer solchen Rezeptdatenbank finden sich in Kapitel 9). Es ist interessant, dass es einige Benutzer (6,06%) gibt, die den Roboter auf keinen Fall kochen lassen wollen, sondern ihm höchstens Vorbereitungen überlassen wollen. Sie scheinen zu befürchten, dass der Roboter ihr Essen verderben könnte.

Darüberhinaus sind auch die Aufgaben, an die die Benutzer bei der Vorbereitung eines Abends mit Freunden gedacht haben, wie beispielsweise Essen vorbereiten, Tisch decken, usw. größtenteils schon vom System abgedeckt (cf. Abbildung 5.4).

5.2.3 Auswertung der spezifischen Fragen

Bei den spezifischen Fragen, die sich in dem Fragebogen in den Fragen 6-9 finden (cf. Abbildung 5.1), wurde überprüft, inwieweit die Formulierungen der Benutzer von der aktuellen Grammatik abgedeckt sind. Dafür wurde die Rate der Benutzeräußerungen gemessen, die vom System geparkt werden konnten (siehe Abbildung 5.5): “Verstanden” bedeutet, dass die Äußerung komplett geparkt werden konnte und eine semantische Repräsentation aufgebaut wurde. Demgegenüber konnte bei “teilweise verstanden” nur ein Teil der Äußerung korrekt geparkt werden, ein anderer Teil nicht. Auf Basis dieser korrekt geparkten Teiläußerung kann der Dialogmanager Nachfragen stellen, so dass der Benutzer auch die zunächst nicht verstandenen Teile der Äußerung erklären kann. “Nicht verstanden” bedeutet, dass die Äußerung nicht geparkt werden konnte, und “Keine

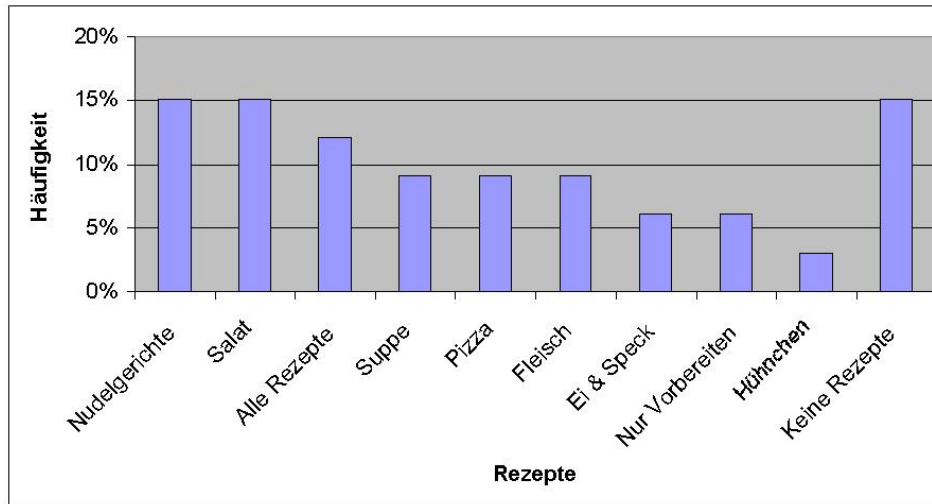


Abbildung 5.3: Auswertung der Benutzerbefragung zu Rezepten, die der Roboter kennen sollte, sortiert nach der Häufigkeit, in der sie von den Benutzern genannt wurden.

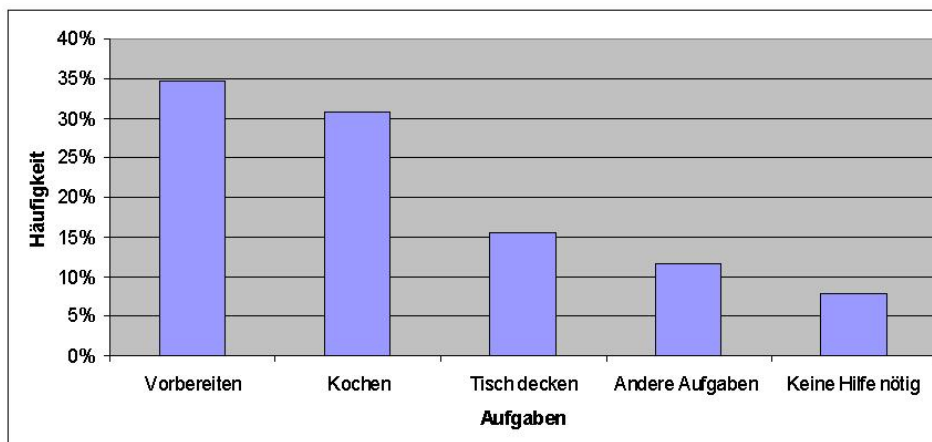


Abbildung 5.4: Auswertung der Benutzerbefragung zu den Instruktionen für die Vorbereitung eines Abends mit Freunden, sortiert nach der Häufigkeit, in der sie von den Benutzern genannt wurden.

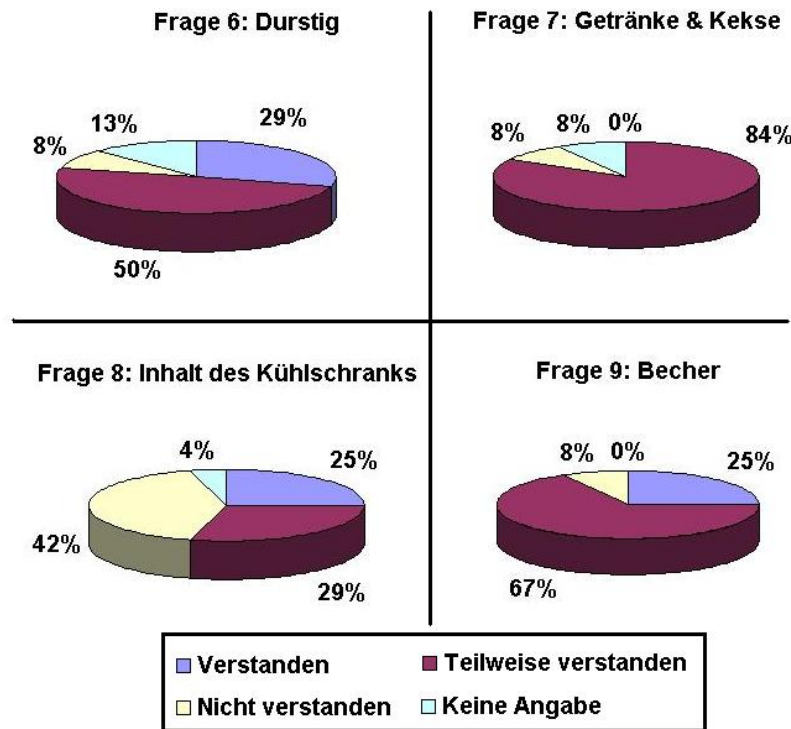


Abbildung 5.5: Auswertung der Benutzerbefragung: Erkennungsrate der Benutzeräußerungen für verschiedene Aufgaben

Angabe", dass der Benutzer diese Frage nicht beantwortet hat.

Die Ergebnisse zeigen, dass ungefähr 80% der Äußerungen verstanden oder zumindest teilweise verstanden werden konnten. Nur für die achte Frage nach dem Inhalt des Kühlschranks lag die Verstehensrate nur bei 55%, was sich dadurch erklären lässt, dass die Frage sehr frei formuliert war, so dass die Benutzer sich ganz unterschiedliche Situationen vorgestellt und damit auch verschiedene Formulierungen gewählt haben, die von "Please have a look in the fridge and tell me what I need to buy." ("Bitte sieh im Kühlschrank nach, was ich einkaufen muss.") bis hin zu komplexen Beschreibungen, wie beispielsweise "Robbi, my sister and her two kids will arrive tomorrow. They will stay with us for 5 days. Please print out a list of everything I should buy, if I don't go shopping again for a full week." ("Robbi, meine Schwester und ihre beiden Kinder kommen morgen und werden für 5 Tage bleiben. Bitte drucke eine Liste mit allem aus, was ich kaufen sollte, wenn ich den Rest der Woche nicht mehr einkaufen gehen will.") reichten.

Ferner scheint die siebte Frage besonders kompliziert gewesen zu sein, weil es keine Versuchsperson schaffte, dass der Roboter ihre Äußerung komplett verstand, sondern es wurden immer nur Teile verstanden. Das lag daran, dass die Prototypgrammatik des Dialogmanagers momentan noch keine komplexeren Äußerungen, die beispielsweise aus zwei Nominalphrasen bestehen, verstehen kann, wie "Robbi, can you bring me a glass of tea and some cookies?" ("Robbi, kannst du mir ein Glas Tee und Kekse bringen?"). Daher wurde bei dem Design des

1. Lassen Sie sich vom Robbi einen Tee mit Milch und Zucker machen.
2. Lassen Sie sich vom Robbi ein Wasser holen.
3. Lassen Sie sich vom Robbi einen blauen Becher bringen.
4. Sie würden gern Spaghetti Napoli kochen. Fragen Sie Robbi, wie das geht.
5. Sie haben sich ein paar Freunde zum Abendessen eingeladen. Bitten Sie Robbi den Tisch für Sie alle zu decken.
6. Lassen Sie sich vom Robbi einen Kaffee ohne Milch, aber mit Zucker machen.
7. Lassen Sie sich vom Robbi eine Cola holen.
8. Sagen Sie dem Roboter, dass er die kleine Lampe anschalten soll.
9. Stellen Sie sich vor, dass Sie abends von der Arbeit nach Hause kommen und Hunger haben. Nun wollen Sie, dass Robbi etwas für Sie kocht.
10. Stellen Sie sich vor, dass Sie gemütlich in Ihrem Sofa sitzen und sich gerade überlegen, was Sie heute Abend essen könnten. Da Sie zu faul sind, in die Küche zu gehen, bitten Sie Robbi, im Kühlschrank nachzusehen, was noch da ist.

Abbildung 5.6: In den aufgabenbasierten Tests verwendete Aufgaben

eigentlichen Tests darauf geachtet, dass die Aufgaben, die der Benutzer den Roboter machen lassen soll, einfach sind und nicht aus mehreren Unteraufgaben bestehen. Die Versuchspersonen können sich, gerade wenn sie den Roboter nicht sehen, nur schwer vorstellen, dass dieser nur einen Gegenstand in seinen Greifern halten kann und fordern ihn daher oftmals auf, mehrere Objekte auf einmal zu holen.

Insgesamt eignet sich die Roboterapplikation für einen webbasierten Test, da die Benutzer eine ähnliche Vorstellung von den Fähigkeiten des Roboters hatten und auch ähnlich Formulierungen verwendeten wie die, die bereits in der Grammatik abgedeckt sind, so dass es im Vortest keine größeren Probleme gab und damit auch keine Schwierigkeiten im eigentlichen Test zu erwarten sind.

5.3 Aufgabenbasierter Benutzertest

5.3.1 Vorgehensweise

Der Link zu dem Roboter-Webinterface wurde an verschiedene Newsgroups und experimentelle Webportale gesendet, um so viele Daten wie möglich bekommen zu können. Im Folgenden wurden Daten von über 70 Benutzern ausgewertet. Insgesamt ergibt dies ungefähr 1000 Turns; durchschnittlich sind das 15 Turns pro Benutzer. Alle Benutzer sprachen mit dem Roboter über das Webinterface

	Mittelwert
Gelöste Aufgaben	2,65
Gesamtzahl Turns	14,48
Turns pro Aufgabe	5,46
Parsingrate mit Satzzeichen	74,62%
Parsingrate ohne Satzzeichen	78,39%
Turnfehlerrate mit Satzzeichen	58,20%
Turnfehlerrate ohne Satzzeichen	49,94%
Abbruchrate	1,22%

Tabelle 5.2: Ergebnisse des aufgabenbasierten Benutzertests

(siehe Abbildung 3.2) und sollten den Roboter fünf der in Abbildung 5.6 vorgegebenen Aufgaben lösen lassen.

Dabei waren die Aufgaben so gewählt, dass es zum einen sehr einfache Aufgaben gab, wie z.B. die zweite Aufgabe (“Lassen Sie sich vom Roboter ein Wasser holen.”), die sich mit einer einzigen Benutzeräußerung lösen ließen. Daneben gab es auch kompliziertere Aufgaben, wie die erste (“Lassen Sie sich vom Roboter einen Tee mit Milch und Zucker machen.”), für die die Benutzer durchschnittlich ein paar Turns benötigten, bis hin zu komplexeren Situationsbeschreibungen, wie in Aufgabe 9 oder 10 (“Stellen Sie sich vor, dass ...”). Für die Auswertung wurden die Logprotokolle des Systems evaluiert. Diese Protokolle enthalten Informationen über die Benutzereingabe, wie diese Eingabe vom System geparkt werden konnte, und schließlich die Ausgabe des Systems.

5.3.2 Ergebnisse

Wie der Tabelle 5.2 zu entnehmen ist, gelang es den Benutzern, den Roboter mehr als die Hälfte der Aufgaben ausführen zu lassen. Die Benutzer benötigten dafür etwas mehr Turns als in den Tests mit dem Roboter: Die durchschnittliche Anzahl an Turns pro Aufgabe liegt hier bei 5,46, während es in dem aufgabenbasierten Test mit dem Roboter 3,4 Turns pro Aufgabe waren. Dies lässt sich zum einen dadurch erklären, dass die Aufgaben in dem Test mit dem Roboter einfacher waren. Zum anderen ist es für die Benutzer einfacher, wenn sie sich den Roboter nicht vorstellen müssen, wie in diesem webbasierten Test, sondern er direkt vor ihnen steht.

Die Turnfehlerrate war recht hoch, da die grammatikalische Abdeckung des Prototypen noch relativ gering war, und einige Benutzer fälschlicherweise Satzzeichen bzw. Ziffern miteingegeben haben, die dazu führten, dass die entsprechenden Sätze nicht korrekt geparkt werden konnten. Daher wurden alle fälschlicherweise eingegebenen Satzzeichen entfernt und die Parsingrate ohne diese Satzzeichen berechnet, die sehr viel besser war (cf. Tabelle 5.2). Auch die Turnfehlerrate war mehr als 5% niedriger ohne die Benutzeräußerungen mit fälschlicherweise eingegebenen Satzzeichen.

Da eine Interaktion, die aus weniger als fünf Turns besteht, bedeutet, dass der Benutzer mit dem Roboter weniger als eine Minute gesprochen hat, werden

	Anteil an allen Äußerungen	Anteil an nicht verstand. Äußerungen
Neue Konzepte	5,43%	64,58%
Neue Formulierung	3,34%	39,62%
Neues Objekt	0,53%	6,28%
Neues Dialogziel	0,80%	9,60%
Metakommunikation	0,76%	9,08%
Ellipsen	0,85%	10,64%
Elliptische Äußerungen & Anaphern	0,63%	7,50%
Fehlender Kontext	0,22%	3,14%
Komplexität	0,49%	5,76%
Verbundene Äußerungen	0,49%	5,76%
Eingabeprobleme	1,60%	19,02%
Satzzeichen	0,97%	11,52%
Ziffern	0,22%	2,62%
Gramm. falsche Äußerungen	0,41%	4,88%

Tabelle 5.3: Detaillierte Fehleranalyse des aufgabenbasierten Benutzertests

fünf Turns als die Grenze für eine Konversation angesehen. Nur sehr wenige Benutzer brachen den Test ab, nimmt man dieses Limit von fünf Turns an. Allen Benutzern, die den Test vorzeitig abgebrochen haben, gelang es nicht, sich dem Roboter in diesen ersten Turns verständlich zu machen, was meist an dem bereits erwähnten Problem mit den fälschlicherweise eingegebenen Satzzeichen lag.

Ungefähr 75% der Äußerungen konnten geparst werden, aber einige davon konnten nicht in die vollständige, korrekte Semantik übertragen werden, was die etwas höhere Turnfehlerrate erklärt. Im Folgenden werden alle die Äußerungen, die nicht korrekt verstanden wurden und zu Fehlern führten, näher beleuchtet. Dafür wurden wiederum alle Äußerungen anhand der Gründe klassifiziert, warum sie nicht verstanden werden konnten (siehe auch Tabelle 5.3).

5.3.3 Detaillierte Fehleranalyse und -klassifikation

Die Fehleranalyse zeigt, dass der Hauptgrund für Fehler auch bei dem webbasierten Benutzertest wiederum neue ontologische oder grammatikalische Konzepte waren (cf. Tabelle 5.3). Viele neue syntaktische Konstruktionen wurden verwendet, wie beispielsweise “Prepare a salad.” (“Bereite einen Salat vor.”), “I want you to cook spaghetti for me.” (“Ich möchte, dass du Spaghetti für mich kochst.”). Manchmal haben die Teilnehmer auch neue Worte für bekannte Objekte verwendet, wie z.B. “icebox” statt “fridge”. Dies könnte daran liegen, dass die Prototypgrammatik recht klein war. Es wäre möglich, dass eine größere Grammatik zu weniger Fehlern in dem Bereich führt, was in weiteren Studien untersucht wird (siehe auch Kapitel 8).

Auch einige neue Dialogziele wurden von den Benutzern verwendet, wie bei-

spielsweise “Switch yourself off.” (“Schalte dich ab.”), “Can you wash the dishes?” (“Kannst du das Geschirr waschen?”). Aber der größte Teil der neuen Dialogziele kann als Metakommunikation angesehen werden, wie bereits im vorhergehenden Kapitel angesprochen. Wenn der Roboter den Benutzer nicht versteht, versucht der Benutzer herauszufinden, was schiefgelaufen ist, und stellt Fragen, wie z.B. “Are you making the coffee?” (“Machst du Kaffee?”) oder “Can you understand me?” (“Kannst du mich verstehen?”). Daher braucht man zukünftig eine Komponente, die in der Lage ist, mit dieser Art Metakommunikation umzugehen, und Zugriff hat auf ein Kontextmodell und den Diskurs, um vorhergehende Benutzeräußerungen einbeziehen zu können (siehe auch Kapitel 13).

Da die Aufgaben vorgegeben waren, resultierten die meisten der anderen neuen Dialogziele daraus, dass der Benutzer eine andere Granularität annahm, als sie bisher implementiert ist. Einige Versuchspersonen haben eine feinere Unterteilung der Aufgaben des Roboters angenommen: Ein Benutzer hat beispielsweise dem Roboter im Detail erklärt, wie man Tee macht, angefangen mit dem Aufsetzen des Wassers, dem Legen des Teebeutels in die Tasse bis hin zum Einfüllen des kochenden Wassers in die Tasse. Aber momentan deckt die Grammatik nur Äußerungen auf einer generelleren Ebene, wie z.B. “Make me a cup of tea, please.” (“Mach mir bitte eine Tasse Tee.”), ab, so dass diese Benutzeräußerungen nicht verstanden werden konnten.

Sehr wenige neue Objekte wurden verwendet, wie z.B. “cupboard” (“Schrank”), “dustbin” (“Mülleimer”). Die Prototypgrammatik deckte bereits den allergrößten Teil der Objekte ab, da die Aufgaben vorgegeben waren.

Manchmal fehlte der Kontext, um eine Äußerung auflösen zu können, und auch elliptische Äußerungen und Anaphern finden sich relativ häufig in den gesammelten Daten, wie im folgenden Beispiel zu sehen ist:

Benutzer: Switch on the lamp.
 (Schalte die Lampe an.)
 Roboter: Which lamp do you want me to switch on?
 (Welche Lampe soll ich anschalten?)
 Benutzer: The small one.
 (Die kleine.)

Hier referenziert der Benutzer auf die Lampe, indem er sagt “the small one” (“die kleine”). Dafür werden Mechanismen entwickelt, um elliptische und anaphorische Äußerungen in Zukunft mit Hilfe des Kontexts auflösen zu können (siehe auch Kapitel 11).

Auf der anderen Seite gab es auch einige Äußerungen, die zu komplex sind und beispielsweise verbundene Sätze enthalten, was momentan noch nicht aufgelöst werden kann, wie z.B. “I need a cup of coffee that has about a quarter cup of milk in it.” (“Ich brauche eine Tasse Kaffee mit einem Viertel Becher Milch drin.”), “I want you to cook spaghetti for me and a coke.” (“Ich möchte dass du Spaghetti kochst und eine Cola.”), etc. In diesen Fällen muss sichergestellt werden, dass solche Äußerungen geparkt und verstanden werden können (siehe auch Kapitel 12).

Ferner gab es einige grammatikalisch falsche Äußerungen, wie z.B. “Can you put the light on?” (“Kannst du das Licht anstellen?”), “You bring coffee.” (“Du

bringen Kaffee.”), “I want drink.” (“Ich möchte trinken.”), usw. Da diese Äußerungen darauf zurückgeführt werden können, dass einige der Benutzer keine englischen Muttersprachler waren, soll in dieser Arbeit nicht näher darauf eingegangen werden.

5.3.4 Benutzerverhalten nach dem Auftreten von Problemen und Fehlern

Das Verhalten der Versuchspersonen, wenn sie sich mit Problemen oder Fehlern konfrontiert sahen, ist sehr interessant: Viele Benutzer haben immer kürzere Äußerungen verwendet, manchmal nur aus einem Wort bestehend, wenn sie merkten, dass der Roboter sie nicht verstand:

Benutzer: Get me the blue cup.
(Bring mir den blauen Becher.)
Roboter: Sorry. I didn't understand you, please try again!
(Entschuldigung, ich habe dich nicht verstanden.
Bitte versuch es nochmal!)
Benutzer: Blue cup.
(Blauer Becher.)

Hier hat der Benutzer beispielsweise nur “blue cup” (“blauer Becher”) wiederholt, als er bemerkte, dass der Roboter ihn anscheinend nicht verstanden hat.

Manche Benutzeräußerung sind nach dem Auftreten eines Fehlers sogar noch extremer verkürzt:

Benutzer: I want to drink coffee.
(Ich möchte Kaffee trinken.)
Roboter: I am sorry. I can't understand you.
(Es tut mir leid, ich habe dich nicht verstanden.)
Benutzer: kitchen-coffee.
(Küche-Kaffee.)

Nachdem der Roboter “I want to drink coffee.” (“Ich möchte Kaffee trinken.”) nicht verstanden hat, hat der Benutzer danach nur noch “kitchen-coffee” (“Küche-Kaffee”) gesagt, was das System sicherlich nie verstehen wird.

Ferner haben die Benutzer im Falle eines Fehlers verschiedene Worte ausprobiert und einfach abgewartet, ob der Roboter sie versteht. Leider wurden die meisten Worte zwar korrekt vom System verstanden, aber da sie nicht ohne Kontext interpretiert werden konnten, gab es keine Antwort vom Dialogsystem. Daher werden im Folgenden Methoden entwickelt, um mit einzelnen Worten besser umgehen zu können und dem Benutzer mehr Rückmeldungen zu geben, so dass er weiß, was verstanden wurde (siehe auch Kapitel 11).

Es erwies sich als ein generelles Problem, dass das Dialogsystem Teile einer Benutzeräußerung verstand, dies aber dem Benutzer nicht mitteilte, so dass er dachte, er wäre nicht verstanden worden:

Benutzer:	Can you get me the blue cup, please? (Kannst du mir den blauen Becher holen?)
Semant. Repräs.:	[generic:object,NP,_] { [robbi:obj_cup,N,_] { THE BLUE CUP } [robbi:please_NT] { PLEASE } }
Roboter:	Sorry. I didn't understand you, please try again! (Entschuldigung, ich habe dich nicht verstanden. Bitte versuch es nochmal!)
Benutzer:	Blue cup, please. (Blauer Becher bitte.)
Semant. Repräs.:	[generic:object,NP,_] { [generic:object,NP,Sg] { [robbi:obj_cup,N,Sg] { CUP } [robbi:please_NT] { PLEASE } } }

Hier hat der Benutzer den Roboter gebeten, ihm den blauen Becher zu holen. Aber da genau diese syntaktische Konstruktion nicht von der Prototypgrammatik abgedeckt war, gelangte nur das Objekt, nämlich "the blue cup" ("der blaue Becher") in den Diskurs, was der Benutzer nicht wusste und darum erneut nach dem blauen Becher fragte. Daher sind bessere Grounding-Mechanismen in Zukunft nötig (siehe auch Kapitel 11).

5.4 Benutzertest ohne Aufgaben

5.4.1 Vorgehensweise

Als nächstes wurde ein Benutzertest mit dem webbasierten Interface durchgeführt, ohne den Benutzern konkrete Aufgaben zu geben. Sie wussten lediglich, dass es sich bei dem System um einen Haushaltsroboter handelt, der dem Benutzer in der Küche helfen soll. Welche Aufgaben sie ihm gestellt haben, war ganz ihrer Phantasie überlassen.

An diesem Test haben über 40 Versuchspersonen teilgenommen, deren Dialoge ausgewertet wurden. Dafür wurden wiederum die Logprotokolle des Systems evaluiert, wie auch in Tabelle 5.4 zu sehen. Darüberhinaus wurde die Benutzerzufriedenheit für eine kleine Gruppe von Benutzern, die sich bereit erklärten, zusätzlich an dieser Befragung teilzunehmen, gemessen.

5.4.2 Ergebnisse

Allen Benutzern gelang es, den Roboter einige Aufgaben ausführen zu lassen. Durchschnittlich konnten etwa 3 Aufgaben erfolgreich von den Benutzern zu Ende gebracht werden (siehe Tabelle 5.4). Dies ist besonders interessant, wenn man bedenkt, dass die Versuchspersonen in dem Test mit Aufgaben auch jeweils ca. 3 Aufgaben gelöst haben, obwohl sie dort bereits vorgegebene Aufgaben hatten. Je länger die Benutzer mit dem Roboter redeten, desto mehr Aufgaben konnten sie erledigen lassen. Wie erwartet ist die durchschnittliche Anzahl an Turns pro Aufgabe mit 6,54 etwas höher als beim aufgabenbasierten Test, wo sie bei 5,46 lag. Die Benutzer konnten daher, obwohl sie nur eine generelle Vorstellung von

	Mittelwert
Gelöste Aufgaben	2,6
Gesamtzahl Turns	17
Turns pro Aufgabe	6,54
Parsingrate mit Satzzeichen	62,60%
Parsingrate ohne Satzzeichen	69,77%
Turnfehlerrate mit Satzzeichen	62,40%
Turnfehlerrate ohne Satzzeichen	50,93%
Abbruchrate	4,9%

Tabelle 5.4: Ergebnisse des Benutzertests ohne Aufgaben

dem Roboter hatten und nicht wussten, welche Aufgaben er wirklich erledigen kann, mit ihm zurecht kommen und ihn verwenden, um einige Aufgaben in einer sinnvollen Zeitspanne zu lösen.

Wieder gab es Eingabeprobleme, weil einige Benutzer Satzzeichen miteingegeben haben, die momentan von dem Internetinterface nicht verarbeitet werden können. Auch hier sieht man, dass die Turnfehlerrate weit niedriger und die Parsingrate deutlich höher ist, wenn die falsch eingegebenen Satzzeichen gelöscht wurden (cf. Tabelle 5.4). Wie erwartet ist die Turnfehlerrate hier insgesamt höher als im Test mit Aufgaben, da die Benutzer weniger Anhaltspunkte hatten, was der Roboter kann.

Setzt man wiederum 5 Turns, wie bereits im Abschnitt 5.3.2 erläutert, als Grenze für einen Dialog an, so haben nur wenige Benutzer den Test abgebrochen; die meisten scheinen Geschmack an dem Roboterdialog gefunden zu haben und haben eine ganze Zeitlang mit ihm geredet. Bei allen Benutzern, die den Test abgebrochen haben, zeigt sich wiederum, dass der Dialogmanager sie schon in den ersten Äußerungen nicht verstanden hat, was meist an dem Problem der fälschlicherweise eingegebenen Satzzeichen und Ziffern lag. Insgesamt lag die Abbruchrate höher als beim Test mit Aufgaben, was darauf zurückzuführen ist, dass die Benutzer sich mit Aufgaben sicherer fühlen und nicht so schnell aufgeben, wenn der Roboter die ersten paar Benutzeräußerungen nicht versteht.

5.4.3 Detaillierte Fehleranalyse und -klassifikation

Der Hauptgrund für Fehler waren auch diesmal neue syntaktische oder semantische Konzepte (cf. Tabelle 5.5). Durchschnittlich wurden etwa 3 neue Aufgaben, die der Roboter momentan nicht ausführen und auch nicht verstehen kann, von einem Benutzer eingeführt. Dabei ging es auch hier vor allem um Aufgaben, die die Metakommunikationsebene betreffen, wie z.B. "How many cups are there?" ("Wie viele Becher gibt es?"), "What recipes do you know?" ("Welche Rezepte kennst du?"), "I only have blue cups!" ("Ich habe nur blaue Becher!"), etc., wie auch bereits in den im vorangegangenen Kapitel beschriebenen Benutzertests mit dem Roboter deutlich wurde. Dies verstärkt die Notwendigkeit einer Komponente für Metakommunikation, wie sie in dieser Arbeit im Kapitel 13 dargestellt wird.

	Anteil an allen Äußerungen	Anteil an nicht verstandenen Äußerungen
Neue Konzepte	46,77%	74,94%
Neue Formulierung	15,78%	25,29%
Neues Objekt	8,32%	13,33%
Neues Dialogziel	12,34%	19,77%
Metakommunikation	10,33%	16,55%
Ellipsen	5,59%	8,97%
Elliptische Äußerungen & Anaphern	3,29%	5,29%
Fehlender Kontext	2,30%	3,68%
Komplexität	2,30%	3,68%
Verbundene Äußerungen	2,30%	3,68%
Eingabeprobleme	7,75%	12,41%
Satzzeichen, Ziffern & gramm. falsche Äußerungen	7,75%	12,41%

Tabelle 5.5: Fehleranalyse der aufgetretenen Fehler im Experiment ohne Aufgaben

Nur sehr wenige wirklich neue Objekte wurden von den Benutzern genannt. Stattdessen haben viele die Objekte oder Aktionen, die der Roboter bereits kannte, anders bezeichnet, als sie bisher in der Grammatik abgedeckt waren, wie beispielsweise “Bring me the pot.” (“Bring mir den Becher.”) statt “Bring me the cup.” (“Bring mir die Tasse.”). Auch hier bleibt wieder die Frage zu klären, inwieweit dies an der zu kleinen Grammatik lag oder inwieweit hier andere Mechanismen in Zukunft nötig sind, um solche Fehler vermeiden zu können (siehe Kapitel 8).

Außerdem erwarteten die Benutzer, dass der Roboter auch elliptische und unvollständige Äußerungen versteht, wie Menschen es können. Beispielsweise haben manche Benutzer versucht, die Nachfrage, welchen Becher sie haben möchten, einfach nur mit “blue” (“blau”) zu beantworten. Hier wird ein Kontextmodell benötigt, um diese Äußerungen auflösen zu können (siehe Kapitel 11).

Auch einige verknüpfte Äußerungen finden sich wieder in dem Benutzertest, wie z.B. “I want milk and toasts.” (“Ich möchte Milch und Toasts.”). Diese sollen zukünftig durch die hier entwickelte Komponente zur Auflösung komplexer Äußerungen gelöst werden (siehe Kapitel 12).

5.4.4 Benutzerverhalten nach dem Auftreten von Problemen und Fehlern

Wie auch bereits in dem multimodalen Benutzertest ohne Aufgaben deutlich wurde (siehe Kapitel 4), gehen die meisten Benutzer bei Fehlern in einem Test ohne vordefinierte Aufgaben sehr schnell davon aus, dass der Roboter die entsprechende Aufgabe nicht kennt, und versuchen es dann mit einer neuen Aufgabe. Daher finden sich in den gesammelten Daten nur wenige Fehlerdialoge.

Interessant ist, dass die Benutzer auch hier mit Wiederholen der Äußerung und Reformulierung versuchen, problematische Situationen wieder aufzulösen. Ein Benutzer versuchte auch einen Dialog abubrechen, indem er "Exit." ("Hör auf.") sagte. Ferner probierten auch hier die Benutzer verschiedene Formulierungen aus, wie an dem folgenden Beispiel deutlich wird, wo der Benutzer ausgetestet hat, welche Getränke der Roboter kennt:

Benutzer: Make me some breakfast.
(Mach mir Frühstück.)
Roboter: What do you want to drink?
(Was möchtest du trinken?)
Benutzer: Milk.
(Milch.)
Roboter: What do you want to drink?
(Was möchtest du trinken?)
Benutzer: Cacao?
(Kakao?)
Roboter: What do you want to drink?
(Was möchtest du trinken?)
Benutzer: Coffee?
(Kaffee?)

5.4.5 Auswertung der Benutzerbefragung

Außerdem wurde eine kurze Benutzerbefragung mit den Benutzern, die sich dafür zur Verfügung gestellt haben, durchgeführt. Bei den Fragen ging es darum, ob die Benutzer schon mit Informatik und Computern generell vertraut sind, ob sie den Roboter gut finden und ihn auch nutzen würden und welche Art von Verbesserungen sie sich vorstellen können. Die Hälfte der Benutzer kannte sich bereits vorher sehr gut mit Computern aus, die andere Hälfte nicht. Dies ist entscheidend, damit auch naive Benutzer, für die eigentlich ein solcher Haushaltsroboter gedacht ist, mit dem Roboter zurecht kommen und kein Vorwissen in dem Bereich nötig ist.

Wie die Tabelle 5.7 zeigt, waren die meisten Benutzer zufrieden mit dem System. Mehr als die Hälfte der Versuchspersonen mochte das System und würde einen solchen Roboter kaufen. Allerdings ist dabei zu beachten, dass die Teilnahme an der Benutzerbefragung freiwillig war, so dass wirklich unzufriedene Benutzer diesen Fragebogen eventuell nicht ausgefüllt haben. Insgesamt fällt auf, dass den Benutzern das System desto besser gefiel, je geringer ihre Turnfehlerrate war. Dies lässt sich darauf zurückführen, dass die Benutzer von der schlechten Verstehensrate eher frustriert sind und das System daher schlechter bewerten. Während sie umgekehrt bei einer besseren Verstehensrate, den Roboter positiver beurteilen.

Als Verbesserungen für die Zukunft haben die meisten Benutzer vorgeschlagen, dass das Dialogsystem mehr Vokabular und generell mehr Anfragen verste-

	Mittelwert
Alter	33,4
Computervorwissen	50,0%
Mag den Roboter	62,5%
Turnfehlerrate	63,3%
Gesamtzahl Äußerungen	23,75

Tabelle 5.7: Benutzerbefragung zu dem Test ohne Aufgaben

hen müsste, so dass der Benutzer nicht länger darüber nachdenken muss, welche Formulierung er für eine bestimmte Aufgabe verwenden sollte, sondern einfach wie mit einem anderen Menschen reden kann. Auch die Antworten und Nachfragen vom System haben die Benutzer manchmal irritiert, wenn der Roboter beispielsweise nachgefragt hat, welchen Becher er dem Benutzer bringen soll. Dies mag darauf zurückzuführen sein, dass die Benutzer den Roboter und seine Umgebung nicht sehen konnten und daher beispielsweise nicht wussten, dass es drei verschiedene Becher gab.

5.5 Zusammenfassung

Vergleicht man die Ergebnisse der beiden Benutzertests (mit vs. ohne Aufgaben), so gab es wie erwartet in dem Test mit Aufgaben weniger Probleme (siehe auch Tabelle 5.8): Die Turnfehlerrate lag im aufgabenbasierten Experimentteil nur bei 49,94% gegenüber 50,93% im Experimentteil ohne Aufgaben. Ferner lag die Abbruchrate im aufgabenbasierten Benutzertest nur bei 1,22%, während sie im Benutzertest ohne Aufgaben 4,9% betrug. Es konnten 78,39% der Benutzeräußerungen geparkt werden gegenüber nur 69,77% im Benutzertest ohne Aufgaben. und mehr Sätze konnten geparkt werden. Interessanterweise war die Zahl der gelösten Aufgaben sehr ähnlich (durchschnittlich 2,6 bzw. 2,65 Aufgaben im aufgabenbasierten Test), obwohl die Benutzer in einem Fall vorgegebene Aufgaben hatten, während sie im anderen Fall diese erst noch selbst kreieren mussten. Daher benötigten die Benutzer durchschnittlich nur 5,46 Turns pro Aufgabe im aufgabenbasierten Test, während es im Benutzertest ohne Aufgaben 6,54 Turns waren.

Ohne vordefinierte Aufgaben gab es ca. doppelt so viele neue Dialogziele (19,77% vs. 9,60% im aufgabenbasierten Test), und ungefähr doppelt so viele neue Objekte wurden von jedem Benutzer im Durchschnitt genannt (13,33% vs. 6,28% im aufgabenbasierten Test). Dies lässt sich darauf zurückführen, dass die Benutzer ohne vorgegebene Aufgaben eine recht unterschiedliche Vorstellung davon hatten, was der Roboter für sie erledigen kann. Dagegen gab es im aufgabenbasierten Teil erheblich mehr neue syntaktische Formulierungen (25,29% vs. 39,62% im aufgabenbasierten Test).

Generell finden sich hier wieder die gleichen Fehlertypen und sogar die gleiche Reihenfolge dieser Fehlertypen, wie sie auch schon bei den Experimenten mit dem Roboter zu sehen war (cf. Kapitel 4): Am meisten Fehler gab es im

	Ohne Aufgaben	Mit Aufgaben
Turnfehlerrate	62,40%	58,20%
Turnfehlerrate ohne Satzzeichen	50,93%	49,94%
Gelöste Aufgaben	2,6	2,65
Gesamtzahl Turns	17	14,48
Neue Konzepte	74,94%	64,58%
Ellipsen	8,97%	10,64%
Komplexität	3,68%	5,76%
Eingabeprobleme	12,41%	19,02%

Tabelle 5.8: Vergleich der aufgetretenen Fehlertypen in den Experimenten mit vs. ohne Aufgaben (Angabe sind hier nur die Raten der jeweiligen Fehlertypen an der Gesamtzahl fehlerhafter Äußerungen)

Bereich der neuen syntaktischen und semantischen Konzepte, dann folgten die Eingabefehler, darauf die Ellipsen und schließlich als kleinste Fehlerquelle die komplexen Äußerungen.

Darüberhinaus lässt sich feststellen, dass die neuen Aufgaben aus dem Bereich der Metakommunikation sich in beiden Tests (mit vs. ohne Aufgaben) sehr stark ähneln, wie beispielsweise Nachfragen, was der Roboter generell kann, ob der Roboter überhaupt macht, was der Benutzer ihm gerade aufgetragen hat, Korrekturen von falsch verstandenen Äußerungen, usw. Außerdem lassen sich in beiden Tests eine ganze Reihe von Benutzeräußerungen finden, wo der Benutzer versucht, den Roboter aus dem Konzept zu bringen, indem er beispielsweise die letzte Äußerung des Roboters einfach wiederholt oder ihm Anweisungen, bei denen er den Roboter und seine Fähigkeiten austesten will, wie z.B. “Bring me a dead body, please.” (“Bring mir eine Leiche.”), “Can you get rid of my ant problem?” (“Kannst du mein Programmierproblem lösen?”), “Rewire the house.” (“Verkabele das Haus neu.”), “Smash a plate.” (“Wirf einen Teller runter.”), gibt.

Insgesamt sind die Ergebnisse der Tests mit dem Roboter und der internetbasierten Tests somit vergleichbar, da sich ähnliche Fehlertypen und auch vergleichbare Verteilungen dieser Fehlertypen finden, was im folgenden Kapitel noch im Detail analysiert wird. Ziel ist es, die in diesen Benutzertests gewonnen Daten im Folgenden auszuwerten und damit eine Basis für ein neues Modul zur Fehlervermeidung und -behebung zu schaffen, das im dritten Teil dieser Arbeit entwickelt und erläutert wird.

Kapitel 6

Fehlerklassifikation

6.1 Einleitung

Dieses Kapitel gibt einen zusammenfassenden Überblick über die durchgeführten Benutzerstudien zu Fehlerdialogen und deren Ergebnissen. Insbesondere geht es darum, erste Ansatzpunkte zu finden, wie sich die gängigsten Fehler vermeiden lassen und wie man die Auswirkungen der Fehler minimieren kann. Dabei werden zunächst die Ergebnisse der verschiedenen Benutzertests verglichen, und die Übertragbarkeit dieser Ergebnisse auf den wirklichen Einsatz eines Küchenroboters im Haushalt wird ermittelt. Daraus ergibt sich eine Klassifikation der aufgetretenen Fehler, wobei für die hier ermittelten Fehlerklassen im dritten Teil dieser Arbeit Methoden entwickelt werden, um diese zu vermeiden oder sie während des Dialogs behandeln zu können.

6.2 Vergleich der verschiedenen Benutzertests

6.2.1 Generelle Unterschiede und Gemeinsamkeiten in den Benutzertests

Um sichergehen zu können, dass alle vier Typen von Benutzertests - der Test mit dem realen Roboter und der Test mit dem Webinterface sowohl mit als auch ohne Aufgaben - vergleichbare Ergebnisse liefern und somit die Resultate übertragbar sind, werden beide nun verglichen (cf. Tabelle 6.1 und Tabelle 6.2).

Insgesamt liegt die Turnfehlerrate bei den Tests mit dem wirklichen Roboter und bei dem webbasierten Test für den Experimentteil ohne Aufgaben in einem sehr ähnlichen Bereich. Demgegenüber fällt sie für den aufgabenbasier-

	Roboter	Webbasiert
mit Aufgaben	22,57%	58,20%
ohne Aufgaben	57,03%	62,40%

Tabelle 6.1: Vergleich der Turnfehlerrate für Benutzertests mit und ohne Aufgaben bzw. mit dem Roboter und mit dem Webinterface.

	Roboter ohne Aufg.	Roboter mit Aufg.	Webtest ohne Aufg.	Webtest mit Aufg.
Gelöste Aufgaben	8,5	28	2,6	2,65
Gesamtzahl Turns	32	95,25	17	14,48
Turns pro Aufgabe	3,76	3,40	6,54	5,46
Neue Konzepte gesamt	73,98%	36,04%	74,94%	64,58%
Neue Formulierung	32,88%	25,58%	25,29%	39,62%
Neues Objekt	6,85%	0%	13,33%	6,28%
Neues Dialogziel	4,11%	4,65%	19,77%	9,60%
Meta-Kommunikation	30,14%	5,81%	16,55%	9,08%
Ellipsen gesamt	8,22%	31,40%	8,97%	10,64%
Ellipt. Äußerungen & Anaphern	8,22%	23,26%	5,29%	7,50%
Fehlender Kontext	0%	8,14%	3,68%	3,14%
Komplexität	2,74%	1,16%	3,68%	5,76%
Eingabeprobleme	15,06%	31,40%	12,41%	19,02%

Tabelle 6.2: Vergleich der auftretenden Fehlertypen in den Experimenten mit und ohne Aufgaben bzw. mit dem Roboter und mit dem Webinterface (Angaben sind hier nur die Raten der jeweiligen Fehlertypen an der Gesamtzahl fehlerhafter Äußerungen)

ten Teil deutlich niedriger aus bei den Tests mit dem realen Roboter gegenüber den Webtests (siehe auch Tabelle 6.1). Dies liegt sicherlich z.T. daran, dass die Aufgaben beim Test mit dem Roboter sehr einfach waren, da er momentan keine komplexeren Aufgaben ausführen kann. Ferner wurde beim Internettest vom Benutzer mehr Vorstellungsvermögen verlangt, weil er nicht sehen konnte, was es in der Küche gab. Dies zeigt sich auch in der durchschnittlich benötigten Anzahl an Turns zur Lösung einer Aufgabe, die für aufgabenbasierte Tests im Internettest bei 5,46 liegt gegenüber 3,40 im Test mit dem Roboter. Generell ist die durchschnittlich benötigte Anzahl an Turns zur Lösung einer Aufgabe bei den aufgabenbasierten Tests niedriger, da die Benutzer sich nur auf Aufgaben beziehen, die der Roboter generell lösen kann.

Außerdem ist die Turnfehlerrate für Tests mit Aufgaben immer geringer als für Tests ohne Aufgaben, da der Benutzer in dem Fall weniger Anhaltspunkte hat, was er zum Roboter sagen kann, und daher mehr von der Grammatik nicht abgedeckte Äußerungen vorkommen.

Im Webtest wurden insgesamt weniger Aufgaben gelöst und auch die Anzahl der Turns war geringer (cf. Tabelle 6.2), was zum einen darauf zurückzuführen ist, dass die Benutzer ohne Versuchsleiter eher aufhören, da niemand da ist, der sie motiviert, noch weiter zu machen. Ferner hatten die Versuchspersonen im Internettest im aufgabenbasierten Teil die Anweisung erhalten, jede Aufgabe nur einmal auszuführen, während sie es im Test mit dem Roboter mehrmals

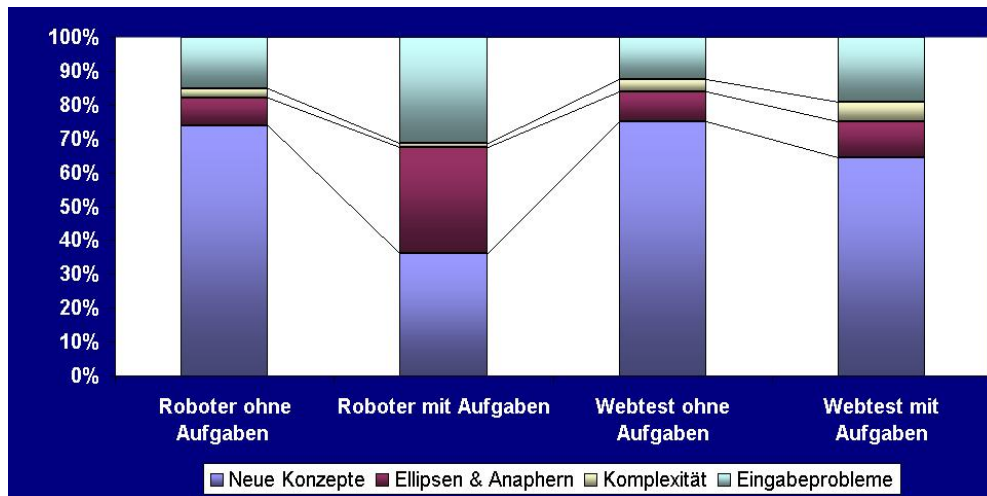


Abbildung 6.1: Gegenüberstellung der auftretenden Fehlertypen in den Experimenten mit und ohne Aufgaben bzw. mit dem Roboter und mit dem Webinterface

ausprobieren sollten.

6.2.2 Auftreten der Fehlertypen in den Benutzertests

Vergleicht man nun die aufgetretenen Fehler, so sieht man, dass in allen vier Fällen die gleichen Fehlertypen zu finden sind, nur die Verteilung leicht schwankt (siehe auch Abbildung 6.1). Sogar die Aufteilung der Fehlertypen in der Gesamtmenge der Fehler ist ebenfalls über alle vier Bedingungen gleich, so dass sich die meisten Fehler im Bereich der neuen Konzepte und die wenigsten Fehler bei den komplexen Äußerungen finden:

1. Neue syntaktische und semantische Konzepte
2. Eingabefehler
3. Ellipsen
4. Komplexe Äußerungen

Innerhalb der neuen Konzepte finden sich in den Tests mit Aufgaben mehr neue Formulierungen, während es in den Tests ohne Aufgaben mehr neue semantische Konzepte gibt (cf. Abbildung 6.2). Das lässt sich darauf zurückführen, dass die Benutzer ohne Aufgaben weniger Vorgaben haben, was für Aktionen der Roboter überhaupt ausführen kann. Insgesamt findet sich ein Großteil der Fehler im Bereich der neuen Konzepte in der Metakommunikation. Die Rangfolge zwischen neuen Objekten und Dialogzielen ist unterschiedlich zwischen den verschiedenen Formen der Benutzertests; beide Fehlertypen liegen jedoch zahlenmäßig sehr dicht zusammen, so dass Fehler in beiden Bereichen etwa gleich gewichtet werden können. Generell bleibt bei dem Fehlertyp der neuen Konzepte allerdings zu klären, inwieweit die prototypische Grammatik mit ihrer geringen

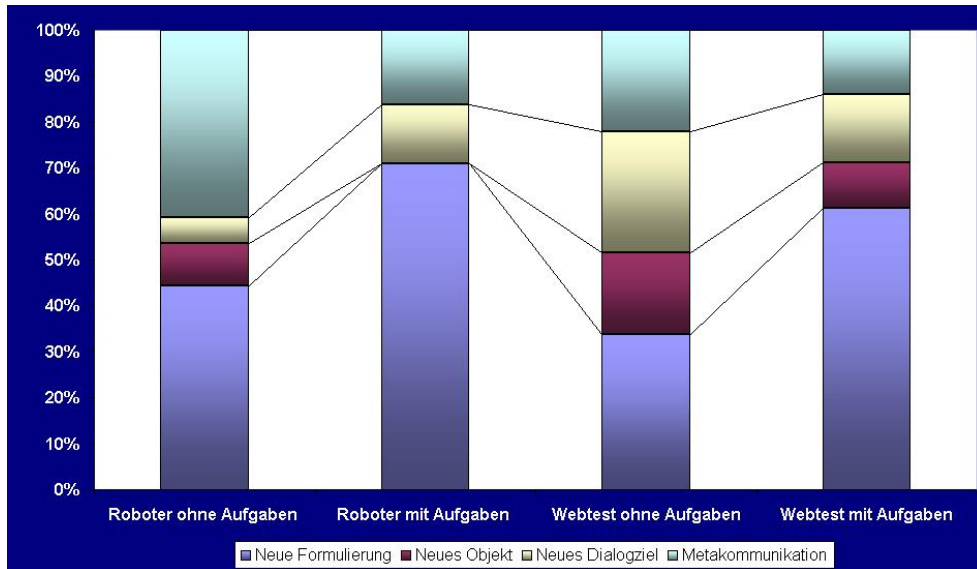


Abbildung 6.2: Gegenüberstellung der Fehlertypen im Bereich der neuen Konzepte in den Experimenten mit und ohne Aufgaben bzw. mit dem Roboter und mit dem Webinterface

Abdeckung hier zu Problemen führt, oder sich eine ähnliche Verteilung auch mit einer größeren Grammatik zeigt (siehe auch Kapitel 8).

Bei Eingabefehlern handelt es sich um Fehler, die im Internetttest durch fälschliche Eingabe von Satzzeichen und Ziffern und im Test mit dem Roboter durch Segmentierungsfehler oder zu laute Hintergrundgeräusche entstanden sind. Diese sind durch vor dem Dialogmanager vorgeschaltete Komponenten zu beheben und werden daher in der vorliegenden Arbeit nicht weiter thematisiert, da es hier um semantische Fehler im Dialog geht.

Die Behandlung von Ellipsen und Anaphern ist nicht nur im Zusammenhang mit der Fehlervermeidung wichtig, sondern vor allem auch für die Fehlerbehebung, bei der häufig elliptische Äußerungen in den Benutzertests zu finden waren (siehe Kapitel 4 und 5). Auf sie geht daher das Kapitel 11 näher ein. In den Benutzertests mit Aufgaben finden sich mehr Ellipsen als in den freien Benutzertests, was sicherlich auch mit der Art der Aufgaben zusammenhängt, da die Benutzer in dem Test mit dem Roboter explizit aufgefordert wurden auch Zeigegesten einzusetzen, was die Verwendung von deiktischen Pronomen fördert.

Die komplexen Äußerungen sind zwar die mit weitem Abstand kleinste Fehlerquelle. Da sie sich aber in allen Tests finden, wird im Kapitel 12 eine Möglichkeit dargestellt, solche Sätze generell zu modellieren. Auf diese Weise können auch komplexe Äußerungen verstanden werden, die der Roboter nicht ausführen kann, um den Benutzer dann zumindest darüber informieren zu können.

Insgesamt zeigt es sich, dass Internetttests und Tests mit dem Roboter zu den gleichen Fehlerproblemen führen. Beide ergänzen sich gut, da Tests mit dem realen Roboter der alltäglichen Mensch-Roboter-Interaktion, wie man sie sich im Idealfall vorstellt, ähnlicher sind, während Internettests den Vorteil bieten,

dass weit mehr Benutzerdaten gesammelt werden können. Außerdem können in einem solchen Internetttest auch Aufgaben gewählt werden, die der Roboter in Wirklichkeit momentan noch nicht lösen kann. Dies hat den Vorteil, dass sich längere Dialoge ergeben, die für eine Auswertung interessanter sind. So finden sich auch die interessantesten Auswertungen der Fehlerdialoge in den Benutzertests im Kapitel 5, in denen die Benutzer kompliziertere Aufgaben zu lösen hatten, die sich nicht nur in einem Benutzerturn lösen ließen, wie einige der Aufgaben in Kapitel 4.

6.2.3 Benutzerverhalten bei Problemen und Fehlern

In den Tests mit dem Roboter als auch in denen übers Internet (siehe Kapitel 4 und 5) zeigte sich, dass die Benutzer, wenn sie keine konkrete Aufgabe haben, beim Auftreten von Fehlern und Missverständnissen häufig nicht versuchen, diese zu klären, sondern zu einer anderen Aufgabe übergehen. Dies ist von besonderem Interesse, wenn man bedenkt, dass diese Tests ohne Aufgaben am stärksten der Alltagssituation ähneln, in der ein Benutzer zu Hause einen solchen Haushaltsroboter wirklich verwendet. Dies verdeutlicht, wie wichtig es ist, dass der Roboter den Menschen möglichst gut versteht und ihn im Fall von Missverständnissen bei ihrer Klärung unterstützt, um zu vermeiden, dass der Benutzer zu schnell aufgibt. Außerdem unterstreicht dies, wie wichtig es ist, dass der Roboter bei Problemen die Initiative ergreift und von sich aus den Benutzer über seine Fähigkeiten aufklärt.

In den aufgabenbasierten Tests haben die Benutzer sowohl übers Internet als auch mit dem Roboter versucht, den Roboter bei Problemen wieder auf den richtigen Weg zurückzuführen. Dazu haben sie folgende Strategien angewandt:

- Reformulierung der letzten Äußerung
- Exakte Wiederholung der letzten Äußerung
- Verwenden immer kürzerer Äußerungen, um zu testen, was der Roboter versteht

Um zukünftig solchen Reformulierungen besser begegnen zu können, wurde die Grammatik mit Hilfe von in weiteren Benutzertests gewonnenen Entwicklungsdaten erweitert (siehe Kapitel 8). Die exakte Wiederholung der letzten Äußerung ist sehr problematisch, da sie häufig hyperartikulierte ist und damit zu Erkennungsproblemen führt (Soltau, Waibel 2000); da dies aber eher im Bereich der Spracherkennung zu sehen ist, soll es hier nicht näher analysiert werden. Sehr kurze, elliptische Benutzeräußerungen sollen zukünftig von dem im Kapitel 11 vorgestellten Mechanismus behandelt werden.

6.3 Zusammenfassung

Insgesamt lassen sich die Ergebnisse der Benutzertests mit dem Roboter mit denen, die im webbasierten Test gewonnen wurden, angesichts der Tatsache, dass sie zu den gleichen Typen von Fehlern führen und diese auch in der gleichen Reihenfolge auftreten, vergleichen.

Der folgende dritte Teil dieser Arbeit beschäftigt sich nun mit Lösungen für die genannten Fehlertypen. Dabei wird es in den Kapiteln 8, 9, 10 und 13 um neue Konzepte gehen: Zunächst geht es um die Frage, inwieweit sich die große Anzahl an Fehlern in diesem Bereich darauf zurückführen lässt, dass es sich bei der in den Benutzertests verwendeten Grammatik um eine erste Prototypgrammatik handelte (cf. Kapitel 8), bevor dann Methoden zur dynamischen Integration von weiteren Wissensquellen zur Vermeidung von Fehlern im Bereich der neuen Konzepte erläutert werden. Dabei beschäftigt sich das Kapitel 9 mit der dynamischen Generierung von neuem Vokabular und das Kapitel 10 mit der Integration eines internetbasierten Frage-Antwort-Systems, um auch weitergehende Fragen des Benutzers beantworten zu können. Um die Auflösung der metakommunikativen Äußerungen geht es im Kapitel 13.

Ferner fällt bei den Benutzertests ohne Aufgaben auf, dass sie den Benutzern weit schwerer fallen und dass die individuellen Unterschiede sehr viel größer sind. Hier bietet es sich an, dem Benutzer durch gezielte Informationen vom Roboter über dessen Fähigkeiten weiter zu informieren und ihn damit in den Situationen, in denen er nicht weiß, wie er weiter vorgehen soll, besser zu unterstützen. Dies wird im Kapitel 13.4 thematisiert.

Die Auflösung von Anaphern und Ellipsen mit Hilfe eines erweiterten Kontextmanagements ist das Thema in Kapitel 11. Die Auflösung komplexer Äußerungen wird schließlich im Kapitel 12 dargestellt.

Zuvor beschäftigt sich das Kapitel 7 aber noch damit, inwieweit die Kommunikationsstrategie des Benutzers den Dialog und die auftretenden Fehler beeinflusst und wie die verschiedenen subjektiven und objektiven Maße des Dialogerfolgs miteinander zusammenhängen. Ferner geht es im Kapitel 7.3 um generelle Strategien zur Fehlervermeidung auf Seiten des Roboters, die aber nicht ausreichend sind und erst in Kombination mit den in den darauf folgenden Kapiteln dargestellten Mechanismen zu einer erfolgreicherer Mensch-Roboter-Interaktion führen.

Teil III

Strategien zur Fehlervermeidung und -behebung

Kapitel 7

Evaluation der Kommunikationsstrategien in bezug auf Fehler

7.1 Einleitung

Nachdem in den vorangegangenen Kapiteln untersucht wurde, welche Fehler sich in Mensch-Roboter-Dialogen ergeben und wie diese in verschiedene Fehlertypen klassifiziert werden können, geht es in diesem und den folgenden Kapiteln nun um die Fehlervermeidung und -behebung. Dieses Kapitel beschäftigt sich zunächst mit den Auswirkungen der vom Benutzer bzw. vom Roboter verwendeten Strategie und ihrem Zusammenhang zu Fehlern.

7.2 Auswirkungen der Benutzerstrategie

7.2.1 Einleitung

Um den Zusammenhang von kommunikativem Erfolg, Kommunikationsstrategie des Benutzers und Fehlern im Dialog ermitteln zu können, wurde eine Studie durchgeführt, die den Einfluss der Benutzerstrategie auf den kommunikativen Erfolg evaluiert (Gieselmann, Stenneken 2006). Dabei ist die zentrale Frage, ob und wie die Kommunikationsstrategie des Benutzers die Mensch-Roboter-Interaktion sowohl in der subjektiven Wahrnehmung des Benutzers als auch in objektiv messbaren Erfolgsmaßen beeinflusst. Da bestimmte Kommunikationsstrategien, die bewusst oder unbewusst bei der Interaktion mit einem nicht-menschlichen Partner angewandt werden, zum Kommunikationserfolg beitragen können, wurden die Teilnehmer der Studie angewiesen, unterschiedliche Kommunikationsstrategien zu verwenden. Dafür wurde der Einfluss der Benutzerstrategie auf den kommunikativen Erfolg evaluiert. Darüberhinaus wurden die verschiedenen objektiven und subjektiven Maße des Erfolgs verglichen, um zu sehen, inwieweit sich diese ergänzen können, und worin ihr Zusammenhang zu Fehlern liegt.

7.2.2 Benutzertest

In einem Benutzertest mit dem webbasierten System wurden die Daten von 39 Teilnehmern ausgewertet (66% Frauen, 34% Männer). Das Durchschnittsalter lag bei 29 Jahren. Die Teilnehmer gaben Durchschnittswerte von 2.92 für die Benutzung von Computern und etwas höhere Werte von 3.51 für Computerkenntnisse auf einer Skala von 1 bis 6, wobei 1 mehr Nutzung bzw. Kenntnisse bedeutete und 6 weniger Nutzung bzw. Kenntnisse (Von 1: mehr als 60 Stunden pro Woche bis zu 6: 0-3 Stunden pro Woche). Damit ergibt sich ein mittleres Maß an Computernutzung bzw. -kenntnissen über alle Versuchspersonen hinweg.

Um den Zusammenhang von kommunikativem Erfolg sowohl in bezug auf das subjektive Benutzerempfinden als auch auf den objektiv messbaren Aufgabenerfolg zu ermitteln, haben die Teilnehmer zwei verschiedene Instruktionen erhalten:

- Die “Kindinstruktion”: Die Benutzer sollten mit dem Roboter wie zu einem Kind sprechen.
- Die “Nicht-Kindinstruktion”: Hier bekamen die Benutzer keine detaillierten Instruktionen, wie sie mit dem Roboter reden sollten.

Die eine Hälfte der Teilnehmer bekam erst die Kindinstruktion und dann die Nicht-Kindinstruktion, bei der anderen Hälfte war es umgekehrt. Ferner wurden die bereits beschriebenen Aufgaben (siehe Abbildung 5.6) verwendet und in zwei gleich schwierige Aufgabensätze unterteilt. Die Zuweisung dieser Aufgabensätze war ausbalanciert, um Einflüsse der Aufgabensätze auf die Dialogstrategie und den Aufgabenerfolg zu vermeiden. Bei beiden Instruktionen redeten die Benutzer mit dem gleichen Roboter, glaubten aber, es wären zwei verschiedene.

Nachdem die Teilnehmer die Aufgabensätze mit beiden Instruktionen durchgeführt hatten, füllten sie den folgenden Fragebogen über ihren generellen Eindruck von dem System und ihre Erfahrungen während des Experiments aus:

1. Wie zufrieden sind Sie mit dem System?
2. Wie erfolgreich waren Sie?
3. Was hätte besser laufen können? Was glauben Sie, warum es nicht optimal geklappt hat?
4. Wann hat Robbi Sie nicht verstanden? Was war das Problem?
5. (a) Sie haben am Anfang die Instruktion erhalten, sich mit dem Roboter wie mit einem Kind zu unterhalten. Was haben Sie daraufhin gemacht?
(b) Haben Sie eine bestimmte Strategie angewandt, um mit dem Roboter zu reden?
6. Hat das Ihrer Meinung nach zum guten Gelingen des Experiments beigetragen?
7. Was haben Sie anders gemacht als beim ersten Teil des Experiments?

8. Würden Sie einen solchen Roboter gern benutzen, um Ihnen bei der Hausarbeit zu helfen? Warum / Warum nicht?
9. Was sollte Ihrer Meinung nach an den beiden Robotern verbessert werden?

Antworten auf qualitative Fragen, wie 1, 2 und 8 wurden wieder mit Zahlen von 1 bis 6 beantwortet, wobei 1 das Beste und 6 das Schlechteste war; andere Fragen mussten mit Freitext beantwortet werden.

Während der Interaktion der Benutzer mit dem System wurde der objektive Erfolg pro Benutzer gemessen mit Hilfe der Turnfehlerrate, der Anzahl an erfolgreich beendeten Aufgaben, der Anzahl an Benutzerturns, um eine Aufgabe zu lösen, der Anzahl an Benutzerturns, bevor eine Aufgabe endgültig abgebrochen wird, und der Gesamtzahl an Benutzerturns pro Benutzer.

7.2.3 Diskussion der Ergebnisse

Objektive Maße

Die Teilnehmer schafften es generell, den Roboter die vordefinierten Aufgaben erledigen zu lassen (siehe auch Table 7.1). Durchschnittlich konnten 3 Aufgaben gelöst werden. Die Anzahl an Turns pro abgebrochener Aufgabe (4,24) war etwas höher als die Zahl der Turns bei gelösten Aufgaben (2,94), da die Teilnehmer mit dem Roboter "gespielt" und verschiedene Formulierungen ausprobiert haben, wenn er sie nicht gleich verstanden hat. Die Gesamtzahl an Turns (22,68) zeigt, dass die meisten Benutzer eine ganze Weile mit dem Roboter geredet haben. Einige Benutzer haben sogar Geschmack daran gefunden und weiter mit dem Roboter geredet, obwohl bereits alle Aufgaben erledigt waren.

Die Turnfehlerrate war recht hoch (54,24%), was darauf zurückzuführen ist, dass es sich bei dem System um die erste Version des Prototypen handelte, die noch nicht alle möglichen Formulierungen der Benutzer abgedeckt hat. Außerdem haben wieder einige Benutzer die Instruktionen nicht beachtet und Satzzeichen miteingegeben, was zu einer höheren Turnfehlerrate führte (cf. auch Kapitel 5).

Auswirkungen der Dialogstrategie auf den kommunikativen Erfolg

Die Auswirkungen der Kindinstruktion vs. Nicht-Kindinstruktion spiegeln sich in qualitativen und quantitativen Maßen wieder. Innerhalb der quantitativen Maße beeinflusste die Instruktion vor allem die durchschnittliche Äußerungslänge, also die Anzahl Worte pro Benutzeräußerung. Es gab eine etwas geringere durchschnittliche Äußerungslänge bei der Kindinstruktion (*Mittelwert* = 5,02) verglichen mit der Nicht-Kindinstruktion (*Mittelwert* = 5,64).

Weitere Analysen, die sich mit der Reihenfolge der Instruktion als Between-Subjekt-Variable (cf. engl. between-subjects variable) beschäftigten, zeigten, dass der Effekt der Kindinstruktion auf die durchschnittliche Äußerungslänge durch die Reihenfolge, in der die Instruktionen den Teilnehmern präsentiert wurden, moduliert wurde. Der Effekt war minimal signifikant: $p = 0,053$. Der Effekt der kürzeren Äußerungslänge in der Kindinstruktion trat dabei vor allem auf, wenn die Kindinstruktion im zweiten Block gegeben wurde. Dies könnte

	Mittelwert
Gelöste Aufgaben	3,17
Abgebrochene Aufgaben	1,71
Turns pro gelöste Aufgabe	2,94
Turns pro abgebrochene Aufgabe	4,24
Gesamtzahl Turns	22,68
Länge der Äußerung	5,33
Turnfehlerrate	54,24%

Tabelle 7.1: Objektive Maße für den Dialogerfolg

darauf zurückzuführen sein, dass die Teilnehmer, die die Kindinstruktion bereits im ersten Block bekommen haben, diese auch im zweiten Block weiterverwendet haben, unabhängig von der eigentlichen Instruktion.

Neben den Unterschieden in der durchschnittlichen Äußerungslänge zeigten die Teilnehmer auch qualitative Unterschiede in der Kommunikationsstrategie in der Kind- vs. Nicht-Kindinstruktion. Diese wurden in den Fragebögen nach dem Test deutlich: Etwa die Hälfte der Versuchspersonen gab an, bei der Kindinstruktion kurze, einfache, eindeutige und vollständige Sätze verwendet zu haben, wie beispielsweise “I want a pancake.” (“Ich möchte einen Pfannkuchen.”), “I am hungry.” (“Ich habe Hunger.”), “Make a cup of coffee.” (“Mach eine Tasse Kaffee.”). Die Nicht-Kindinstruktion hingegen führte dazu, dass die Benutzer komplexere und längere Sätze verwendeten, wie z.B. “Put one liter water to heat and when it boils, add spaghetti and cook it for 10 minutes, then prepare the napoli sauce, finally rinse the spaghetti and mix them with the napoli sauce.” (“Mach einen Liter Wasser heiß und, wenn es kocht, füge die Spaghetti hinzu und koche sie für 10 Minuten, dann bereite die Napoli-Sauce vor, gieße schließlich die Spaghetti ab und vermische sie mit der Napoli-Sauce.”), “Go into the kitchen and stand in front of the cupboard.” (“Geh in die Küche und stell dich vor den Schrank.”). Einige Teilnehmer, die die Kindinstruktion im ersten Block hatten, gaben an, dass sie das gleiche auch im zweiten Block gemacht hätten, was auch wiederum den gerade beschriebenen Reihenfolgeeffekt der durchschnittlichen Äußerungslänge bestätigt.

Obwohl die Instruktionen das kommunikative Verhalten beeinflussten, hatten sie nur einen nicht-signifikanten Einfluss auf den kommunikativen Erfolg. So ließen sich Tendenzen zu einer niedrigeren Turnfehlerrate in der Kindinstruktion verglichen mit der Nicht-Kindinstruktion beobachten. Ebenso schienen die Versuchspersonen zufriedener mit dem System in der Kindinstruktion zu sein, was sich in niedrigeren Werten in den subjektiven Evaluationsmaßen, die in dem Fragebogen abgefragt wurden, niederschlug. Daher wurden die Daten über beide Bedingungen zusammengefasst (Objektive Maße in Tabelle 7.1 und subjektive Maße in Tabelle 7.2). Paarweise Vergleiche für die subjektiven Maße zeigten, dass die Effekte der Instruktion für alle drei Variablen nicht signifikant waren (p -Werte größer als 0,20). In ähnlicher Weise ergaben auch paarweise Vergleiche für die objektiven Erfolgsmaße keine Signifikanz (p -Werte größer als 0,10).

Variable	Frage	Mittelwert
Zufrieden	Wie zufrieden sind Sie mit dem System?	3,38
Erfolgreich	Wie erfolgreich waren Sie?	3,22
Erneut	Würden Sie einen solchen Roboter wieder benutzen?	3,72

Tabelle 7.2: Subjektive Maße für den Dialogerfolg

Variable 1	Variable 2	Korrelationskoeffizient	Signifikanzniveau
Erneut	Zufrieden	0,609	0,001
Erneut	Erfolgreich	0,395	0,001
Zufrieden	Erfolgreich	0,681	0,001
Turnfehlerrate	# gelöster Aufg.	-0,61	0,001
Turnfehlerrate	# Turns (gel. Aufg.)	-0,14	n.s.
# gelöster Aufg.	# Turns (gel. Aufg.)	0,43	0,001

Tabelle 7.3: Korrelationen zwischen den verschiedenen subjektiven und objektiven Dialogmaßen

Da die Instruktionen in dieser Studie sehr implizit gegeben wurden (“reden wie mit einem Kind”), ließen sie Raum für individuelle Interpretationen. Es ist möglich, dass direktere Instruktionen mit Beispielen des Kommunikationsstils deutlichere Effekte bei den Erfolgsmaßen zeigen würden.

Zusammenhang von subjektiven und objektiven Maßen für den Erfolg

Zur differenzierteren Analyse der verschiedenen Erfolgsmaße wurden weitere Untersuchungen durchgeführt. Die Ergebnisse zeigen eine starke Verbindung zwischen den einzelnen subjektiven Maßen, also der “emotionalen” Evaluation, der Erfolgseinschätzung und der Entscheidung, das System nochmal zu benutzen (siehe Tabelle 7.3). In ähnlicher Weise finden sich signifikante Korrelationen zwischen den objektiven Maßen, wie Turnfehlerrate, Anzahl der gelösten Aufgaben und Anzahl der Turns pro gelöster Aufgabe.

Weitere Analysen beschäftigten sich mit der Verbindung von subjektiven (Zufrieden, Erneut, Erfolgreich) und objektiven (Turnfehlerrate, Anzahl an gelösten Aufgaben, Anzahl an Turns pro gelöster Aufgabe) Maßen. Eine signifikante Korrelation findet sich für das objektive Maß “Gesamtzahl an gelösten Aufgaben” und das subjektive Maß “Bereitschaft, das System wieder zu benutzen” (p -Werte kleiner als 0,05). Obwohl andere Korrelationen nicht signifikant wurden, so ist doch eine numerische Tendenz erkennbar: Wie erwartet waren die Benutzer zufriedener und fühlten sich erfolgreicher, je mehr Aufgaben gelöst werden konnten.

Eine ähnliche numerische Beziehung lässt sich auch bei der Anzahl an Turns pro gelöster Aufgabe beobachten: Hier führt eine niedrigere Anzahl zu besseren Einschätzungen in den subjektiven Variablen. D.h., je schneller die Benutzer

Variable 1	Variable 2	Korrelationskoeffizient	Signifikanzniveau
Äußerungslänge	Erfolgreich	0,23	0,05
Äußerungslänge	# an gelösten Aufg.	-0,34	0,001
Äußerungslänge	# an Turns (gel. Aufg.)	-0,31	0,001
Äußerungslänge	Turnfehlerrate	0,33	0,001

Tabelle 7.4: Korrelationskoeffizient zwischen Äußerungslänge und Erfolg

eine Aufgabe lösen, desto zufriedener sind sie, desto erfolgreicher fühlen sie sich und desto eher würden sie erneut einen solchen Roboter benutzen.

Diese Ergebnisse zusammen mit den Analysen der freien Antworten der Versuchspersonen im Fragebogen legen nahe, dass die Benutzer, die generell positiv gegenüber einem Roboter eingestellt sind, ihm auch eher einige Missverständnisse vergeben und ihm eine gute Bewertung geben, während Teilnehmer, die einen solchen Roboter nicht benutzen wollen, ihm schlechtere Bewertungen geben, unabhängig von der zu verwendenden Kommunikationsstrategie. So gibt es Benutzer, die den Roboter eher als Werkzeug betrachten und andere Benutzer, die ihn wirklich als Kommunikationspartner sehen. Dies wird auch deutlich an dem Kommunikationsstil der Benutzer gegenüber dem Roboter, wie er von Batliner et al. bei Kindern untersucht wurde (Batliner et al. 2004): Auch hier findet sich eine deutliche Instrumentalisierung des Roboters auf der einen Seite und eine Anerkennung des Roboters als “wirklichen Dialogpartner” auf der anderen Seite. Diese generelle Einstellung der Benutzer gegenüber Robotern scheint stärker zu sein, als eine zusätzliche Strategie, die den Benutzern an die Hand gegeben wird. Dies unterstreicht die Wichtigkeit der subjektiven und objektiven Maße für die Auswertung von Dialogen, wie sie auch bereits in Paradise (Walker et al. 1997) beschrieben wird. Darüberhinaus zeigt es, dass auch die generelle Einschätzung des Benutzers in zukünftigen Studien stärker in Kombination mit den anderen Maßen betrachtet werden sollte.

Die durchschnittliche Äußerungslänge korrelierte signifikant mit dem subjektiven Maß des Erfolgs und mit allen objektiven Maßen (siehe Tabelle 7.4). Dies legt nahe, dass die Äußerungslänge gut den Erfolg vorhersagen kann, was allerdings in diesem Fall auch auf die Prototypgrammatik zurückgeführt werden kann, die noch nicht genügend lange Äußerungen abdeckt.

7.2.4 Auswirkungen der generellen Konversationsstrategie

Wie die oben beschriebenen Ergebnisse nahe legen, gibt es einen sehr starken Einfluss der generellen Einstellung der Benutzer gegenüber künstlichen Gesprächspartnern, wie einem Roboter. Diese Einstellung scheint auch einen stärkeren Einfluss auf den Erfolg der Benutzer zu haben als die hier gegebenen Instruktionen. Ähnliche Vorstellungen wurden auch bereits in anderen Studien beschrieben, die die Benutzer anhand ihrer Kommunikationsstrategie in zwei Benutzergruppen aufteilten (Fischer 2006; Clark 1999):

- *Spieler*: Diese Benutzer betrachten den Roboter als Kommunikationspartner und geben daher vor, das Spiel zu spielen, an einer realen Konversation teilzunehmen. Diese Gruppe ist kooperativer und spricht den Roboter direkt an.
- *Nicht-Spieler*: Diese Benutzer sehen den Roboter als ein Werkzeug und geben nicht vor, an dem Spiel einer realen Konversation teilzunehmen. Diese Gruppe behandelt den Roboter als Mittel zum Zweck ohne irgendeine Form von Persönlichkeit.

Obwohl es nicht das eigentliche Ziel dieser Studie war, konnten dennoch alle Benutzer aufgrund ihres Sprachstils zu einer der beiden Gruppen zugeordnet werden. Dabei wurden ähnliche Kriterien, wie in Fischer (2006) verwendet: Spieler sprachen den Roboter direkt mit “Robbi” an, verwendeten höflichere Formulierungen, wie “thanks” (“danke”), “would you please” (“könntest du bitte”), wohingegen Nichtspieler den Roboter nie direkt angesprochen haben und mehr Imperative benutzten. Interessanterweise haben die Benutzer ihre Konversationsstrategie im Lauf des Experiments nicht geändert, so dass es keine klare Unterscheidung zwischen den beiden Instruktionen gab. Dies unterstreicht die Überlegenheit der Benutzereinstellung gegenüber dem Roboter über die gegebenen Instruktionen.

Darüberhinaus fanden sich numerische Tendenzen, dass Spieler in beiden Instruktionen erfolgreicher waren als Nicht-Spieler: Ihre Turnfehlerrate war geringer (Spieler: 52,38, Nicht-Spieler: 60,47) und sie lösten mehr Aufgaben (Spieler: 3,28, Nicht-Spieler: 2,78). Gleichzeitig führte ihre Kooperativität zu einer höheren Anzahl an Turns, besonders bei den Aufgaben, die die Benutzer nicht lösen konnten und schließlich abbrechen (Spieler: 4,69, Nicht-Spieler: 2,73). Dies unterstreicht ihre höhere Motivation, den Roboter eine Aufgabe erfüllen zu lassen. Außerdem haben Spieler mehr Klärungsdialoge angestoßen in unklaren Situationen als Nicht-Spieler.

Dies zeigt, dass die Konversationsstrategie der Benutzer unabhängig von externen Instruktionen zu sein scheint, was umso mehr die Notwendigkeit guter Fehlerbehandlungsmechanismen unterstreicht, wie sie in den folgenden Kapiteln untersucht werden sollen.

7.3 Auswirkungen der Roboterstrategie

7.3.1 Einleitung

Auf Seiten des Roboters wurden zwei generelle Fehlervermeidungsstrategien implementiert: Die Verwendung von Konfidenzmaßen und die Hold-Strategie. Konfidenzmaße erlauben eine Aussage darüber, wie zuverlässig die Erkennung vom Spracherkenner ist. Damit ist es möglich, auf potentielle Fehlerkennungen im Dialogmanager einzugehen. Im daran anschließenden Abschnitt geht es um die sogenannte Hold-Strategie, die mit Hilfe aller im Diskurs vorhandener Information Inkonsistenzen erkennt und diese dem Benutzer zur Korrektur anbietet, um weitergehende Fehler zu vermeiden.

7.3.2 Konfidenzmaße

Einleitung

Wie bereits im Kapitel 2.3.3 erläutert, liefern Konfidenzmaße ein Maß für die Güte der erkannten Äußerung und ihrer einzelnen Worte, um mögliche Fehlerkennungen im Spracherkenner finden zu können. Dieses Konfidenzmaß kann dann im Dialog genutzt werden, um beispielsweise einzelne Worte mit sehr schlechter Konfidenz zu ermitteln und für diese Worte einen Klärungsdialog anzustoßen.

In dem an unserem Institut entwickelten Spracherkenner Janus finden sich zwei Arten von Konfidenzmaßen (Finke et al. 1997):

- auf Basis der a posteriori Wahrscheinlichkeit
- auf Basis des Consensus

Die a posteriori Wahrscheinlichkeit wird von dem im Spracherkenner verwendeten Hidden-Markov-Modell ausgegeben und gibt den Pfad durch die Zustände des Hidden-Markov-Modells an (Finke et al. 1997). Dieser Pfad kann mittels Dekodierung bestimmt werden. Aus dem Pfad lässt sich auch die Zustandsfolge der einzelnen Worte bestimmen. D.h., das akustische Signale wird in Teilabschnitte zerlegt, die den einzelnen Worten der erkannten Hypothese entsprechen. Für jedes Wort lässt sich dann die a posteriori Wahrscheinlichkeit $p^I(W)$, also die wahrscheinlichste Wortsequenz für die gegebene akustische Beobachtung, bestimmen. Die a posteriori Wahrscheinlichkeit für das im Zusammenhang erkannte Wort lässt sich bestimmen durch die Differenz der Wahrscheinlichkeit bei Eintritt in das Wort p_E und der Wahrscheinlichkeit bei Austritt aus dem Wort p_A in der logarithmischen Transformation. p_E und p_A lassen sich mittels Forward- bzw. Backward-Algorithmus berechnen. Daher ergibt sich für die Bestimmung der Konfidenz auf Basis der a posteriori Wahrscheinlichkeit die folgende Formel:

$$Conf_{Post}(w) = \log(e^{|p(w)-p^I(w)|} + 1)$$

Je höher also das Konfidenzmaß $Conf_{Post}(w)$ ist, desto geringer ist der Unterschied zwischen den beiden Wahrscheinlichkeiten und desto ähnlicher sind sich demzufolge die global und die lokal gefundene Wortsequenz. Ist der Unterschied zwischen den beiden Wahrscheinlichkeiten sehr groß, so nähert sich das Konfidenzmaß an 0 an.

Die Grundlage der Konfidenz auf Basis des Consensus ist ein Konfusionsnetzwerk (Mangu et al. 2000). D.h., der Worthypothesengraph wird geclustert, so dass gleiche Worte als eine Einheit dargestellt und die verschiedenen Cluster zu einer Ordnung zusammengefügt werden. Dann lässt sich die Konfidenz auf Basis des Consensus ablesen durch:

$$\begin{aligned} Conf_{Cons}(x) &= \sum_{x \in C_i, Word(x)=w_i} p(x) \text{ wenn } w_i \neq -' \\ &= \sum_{x \in C_i} p(x) \text{ wenn } w_i = -' \end{aligned}$$

D.h., die Konfidenz auf Basis des Consensus schwankt zwischen 0 und 1 und ist hoch, wenn sich das erkannte Wort in dem Cluster mit hoher Wahrscheinlichkeit befindet.

Vorgehensweise

Zunächst wurde der Spracherkenner mit akustischen Daten von acht Sprechern, die dem Roboter Anweisungen für seine Aufgaben in der Küche gegeben haben, trainiert. Diese Daten stammen auch aus einem Trainingsdatenset, das weitere Daten aus der Mensch-Roboter-Kommunikation im Küchenszenario enthält und ausführlich im Kapitel 8 beschrieben wird. Neben dem grammatikbasierten Spracherkenner wurde auch ein Trigramm-Sprachmodell aufgebaut aus der ausgerollten Grammatik der Version 1 des Dialogsystems (für einen Überblick über die verschiedenen Versionen siehe Tabelle 8.1).

Allen Trainingsdaten wurde die entsprechende semantische Referenzrepräsentation zugeordnet. Mit Hilfe der Konfidenz auf Basis der a posteriori Wahrscheinlichkeit und auf Basis des Consensus wurde untersucht, ob sich falsch erkannte Äußerungen bzw. Worte von richtig erkannten unterscheiden lassen, um auf dieser Basis Klärungsdialoge anstoßen zu können. Dafür wurden die vor allem im Bereich des Information Retrieval bekannten Maße Präzision (Anzahl der korrekt erkannten Daten durch die Anzahl aller erkannten Daten) und Recall (Anzahl korrekt erkannter Daten durch die Anzahl aller zu erkennenden Daten) und der F-Wert benutzt, der in einem Maß Präzision und Recall gewichtet (cf. z.B. (Cooper 1997)). Der F-Wert berechnet sich dabei wie folgt:

$$F - Wert = \frac{(b^2 + 1)Precision * Recall}{b^2(Precision + Recall)}$$

Der Parameter b bestimmt die Gewichtung von Präzision und Recall: Bei $b = 1$ herrscht ein Gleichgewicht, bei steigendem b gewinnt der Recall an Bedeutung, während bei fallendem b die Präzision stärker gewichtet wird. Für die folgenden Ergebnisse wurde $b = 1$ gewählt, um ein Gleichgewicht von Präzision und Recall zu erreichen.

Ergebnisse

Bei der Erkennung von falsch erkannten Äußerungen liefern sowohl die Konfidenz auf Basis der a posteriori Wahrscheinlichkeit als auch die auf Basis des Consensus recht gute Ergebnisse, wie in Tabelle 7.5 zu sehen: 85,1% der falsch erkannten Äußerungen können mit Hilfe der Konfidenz auf Basis des Consensus bei einer Präzision von 0,96 gefunden werden. Bei der Konfidenz auf Basis der a posteriori Wahrscheinlichkeit ist die Trefferquote zwar mit 0,84 etwas geringer, dafür ist aber die Präzision höher, so dass sich in beiden Fällen ein F-Wert von 0,92 ergibt.

Da für den Dialog die Erkennung von falsch erkannten Worten, nach denen gezielt nachgefragt werden kann, interessanter ist als die Erkennung ganzer falsch erkannter Äußerungen, wurde untersucht, inwieweit die Konfidenz dafür

	Rate (in %)	Treffer- quote	Präzision	F-Wert
Konfidenz auf Basis der a posteriori Wkt.	84,3	0,84	1,00	0,92
Konfidenz auf Basis des Consensus	85,1	0,88	0,96	0,92

Tabelle 7.5: Detektion falsch erkannter Äußerungen

	Grammatikbasierter Spracherkenner	n-Gram-basierter Spracherkenner
Konfidenz auf Basis der a posteriori Wahrscheinlichkeit	59,2%	65,9%
Konfidenz auf Basis des Consensus	58,9%	67,4%

Tabelle 7.6: Gegenüberstellung der Raten an richtigen Zuordnungen zu den Klassen korrekt vs. falsch erkannte Worte für verschiedene Spracherkenner und Konfidenzmaße

brauchbare Anhaltspunkte liefert: Mit dem grammatikbasierten Spracherkenner erreicht man eine Einordnung der Worte in die korrekten Klassen “falsch erkannt” vs. “richtig erkannt” zu 59,2% mit Hilfe der Konfidenz auf Basis der a posteriori Wahrscheinlichkeit. Benutzt man die Konfidenz auf Basis des Consensus, so findet man dort mit 58,9% eine leicht schlechtere Zuordnung.

Bessere Ergebnisse lassen sich mit dem n-gram-basierten Spracherkenner erreichen, da hier die Kosten für den Wechsel eines Pfades auf Grund der kürzeren Historie nicht so hoch sind, wie auch in Tabelle 7.6 zu sehen. Durch die stark eingeschränkten Pfade bei einem grammatikbasierten Spracherkenner sind die Unterschiede zwischen den einzelnen Pfaden größer und es kommt häufiger vor, dass der korrekte Pfad überhaupt nicht enthalten ist.

Darüberhinaus wurde untersucht, inwieweit Lernverfahren, wie Entscheidungsbäume, neuronale Netze, Support Vector Machines und Bayes, noch bessere Ergebnisse erzielen können. Dabei ist das beste Verfahren der Entscheidungsbaum, der ein Training auf semantischer Ebene vornimmt und damit 70,7% der Worte in die korrekte Kategorie “falsch erkannt” vs. “richtig erkannt” erkannt einordnen kann. Mit neuronalen Netzen erreicht man zwar einen sehr guten Recall-Wert, der aber durch die höhere Zahl an korrekt erkannten Worten in den Trainingsdaten zustande kommt und damit insgesamt zu einer schlechteren Rate führt. Auch Support Vector Machines können aus den Daten keine weiteren Informationen extrahieren, da mit der gierigen Suche (cf. engl. greedy search) genau dann der kleinste Fehler erreicht wird, wenn alle Worte als korrekt erkannt eingeordnet wurden, was aber nicht dem Ziel der Klassifikation dient.

Mit Hilfe von Konfidenzen für einzelne Worte ist es also möglich, Klärungsdialoge anzustoßen, die für Worte mit einer geringen Konfidenz ermitteln, ob

der Benutzer etwas anderes gemeint hat. Dabei erweist es sich als vorteilhaft, dass die Präzision gegen 1 geht, was bedeutet, dass kaum eine falsch verstandene Hypothese als korrekt angenommen wird und somit vom Dialogsystem nachgefragt werden kann. Insgesamt liefern jedoch die Konfidenzen bei einem grammatikbasierten Spracherkenner, wie wir in verwenden, dem Dialogmanager keine zuverlässige Information darüber, ob es sich um ein falsch verstandenes Wort handelt oder nicht, da die Zuordnungsraten zu den Klassen "falsch erkannt" bzw. "richtig erkannt" mit 65,9% zu gering ist. Abhilfe ließe sich durch eine Kombination aus grammatikbasiertem und n-gram-basierten Spracherkenner schaffen, wie sie von Liu et al. beschrieben wird (Liu et al. 2005). Dies geht jedoch über den Rahmen dieser Arbeit, deren Fokus auf dem Dialog liegt, hinaus, und wird daher hier nicht näher untersucht.

7.3.3 Hold-Strategie

Einleitung

Hierbei handelt es sich um eine Strategie, die die bereits im Diskurs vorhandene Information effektiver ausnutzt, und dem Benutzer Möglichkeiten bietet, nach einer falsch verstandenen Äußerung den Dialog einfacher fortsetzen zu können (Holzapfel, Gieselmann 2004). Die Strategie basiert auf der Beobachtung, dass Probleme immer dann auftreten, wenn die Benutzereingabe inkonsistent mit der Information, die sich bereits im Diskurs befindet, ist. Dies kann zu einer Situation führen, in der der Benutzer sehr viel Zeit braucht, um seine falsch verstandene Information wieder zu korrigieren, was mit Hilfe der Hold-Strategie vermieden werden soll.

Es gibt zwei Situationen, in denen das System eine unerwartete Antwort vom Benutzer bekommt:

1. Es gab einen Erkennungsfehler. D.h., dass der Benutzer etwas Anderes gesagt hat, als das, was erkannt wurde, und dass diese falsch erkannte Äußerung darum nicht in den Diskurs integriert werden kann.
2. Der Benutzer sagt nicht das, was das System gerade erwartet. Beispielsweise stellt der Dialogmanager eine Klärungsfrage, aber der Benutzer beantwortet sie nicht, sondern sagt etwas Anderes, um z.B. ein neues Dialogziel zu starten. Dies ist der Fall, wenn der Benutzer auf die bereits erwähnte Metakommunikationsebene wechselt und eine für das Dialogsystem unerwartete Nachfrage stellt.

Es lassen sich verschiedene Strategien unterscheiden, um inkonsistente Informationen auflösen zu können. Das System kann

- den alten Dialog abbrechen (weil bisher das falsche Dialogziel angenommen wurde),
- einen neuen Subdialog starten (und danach dann zu dem alten Dialogziel zurückkehren),

- oder die neue Eingabe ignorieren und den Benutzer seine Eingabe wiederholen lassen (z.B. bei Spracherkennungsfehlern).

Um nun ermitteln zu können, welche der Strategien in der aktuellen Situation angebracht ist, wird zunächst überprüft, ob es sich um einen Spracherkennungsfehler handeln könnte und wenn nicht, wird davon ausgegangen, daß der Benutzer einen neuen Subdialog starten möchte.

Spracherkennungsfehler

Es gibt einige Kriterien, die auf einen Erkennungsfehler hindeuten können. Beispielsweise ist in diesen Fällen die Konfidenz für die hereinkommende Hypothese sehr gering oder es findet sich in der n-besten Liste des Spracherkenners eine besser passende Hypothese.

Darüberhinaus wird überprüft, ob die Eingabe selbst schon zur Finalisierung eines Dialogziels führt, was darauf hindeutet, dass sie korrekt ist und die vorhergehende Äußerung eventuell einen Fehler enthielt. Ebenso wird nachgeprüft, ob der vorhergehende Dialogzustand nicht selektiert war, was bedeutet, dass die alte Eingabe wahrscheinlich falsch war und daher gestrichen werden kann (zur Erläuterung der Dialogzustände siehe Kapitel 2.2.2).

Außerdem wird evaluiert, ob der vorhergehende Turn bereits inkonsistente Informationen enthielt. Ist dies der Fall, wird die Diskursinformation gelöscht. So kann vermieden werden, dass der Benutzer in einer nicht wieder aufzulösenden Situation hängen bleibt, da nach zwei aufeinanderfolgenden nicht selektierten Dialogzuständen der Diskurs wieder gelöscht wird und der Benutzer damit von vorn mit einem leeren Diskurs beginnen kann.

Neuer Subdialog

Nachdem überprüft wurde, dass ein Spracherkennungsfehler in der vorliegenden Situation sehr unwahrscheinlich ist, wird davon ausgegangen, dass der Benutzer einen neuen Subdialog starten möchte. Dafür wird ein neuer Subdialog eröffnet und die alte Eingabe gespeichert, um später wieder auf sie zugreifen zu können. So ist es möglich, die nächste Spracherkennershypothese zu überprüfen, ob sie zu dem Subdialog gehört oder nicht. Wenn ja, wird der Subdialog fortgesetzt, wobei der Diskurs erhalten bleibt, um nach dem Subdialog dorthin zurückkehren zu können. Andernfalls wird der Subdialog abgebrochen.

Evaluation

Es wurde ein Benutzertest mit vier Sprechern durchgeführt. In dem Test fanden sich einige Beispiele für die erfolgreich angewandte Hold-Strategie (In eckigen Klammern sind wiederum die vom Spracherkennner erkannten Hypothesen angegeben.):

- Benutzer: Robbi, please set the table. [*robbi, please set the table*]
 (Robbi, bitte decke den Tisch. [*Robbi bitte decke den Tisch*])
- Roboter: For how many persons do you want me

to set the table?
 (Für wie viele Personen soll ich den Tisch decken?)

Benutzer: Eight persons. *[take espresso]*
 (Acht Personen. *[nimm Espresso]*)

Roboter: For how many persons do you want me
 to set the table?
 (Für wie viele Personen soll ich den Tisch decken?)

Benutzer: For eight persons. *[for eight persons]*
 (Für acht Personen. *[für acht Personen]*)

Roboter: Do you want long drink glasses or wine glasses?
 (Möchtest du Wein- oder Longdrink-Gläser?)

Benutzer: Long drink glasses. *[long drink glasses]*
 (Longdrink-Gläser. *[Longdrink-Gläser]*)

Roboter: Do you want your meal with or without dessert?
 (Gibt es ein Essen mit oder ohne Nachtisch?)

Benutzer: Of course with dessert. *[put forks please there]*
 (Natürlich mit Nachtisch. *[leg die Gabeln bitte hierher]*)

Roboter: Do you want your meal with or without dessert?
 (Gibt es ein Essen mit oder ohne Nachtisch?)

Benutzer: With dessert. *[with dessert]*
 (Mit Nachtisch. *[mit Nachtisch]*)

→ Dialogziel finalisiert

Hier sieht man, dass die falsch erkannte Äußerung “Put forks please there.” (“Leg die Gabeln bitte hierher.”) nicht zu einem Abbruch des gesamten Dialogziels führt, sondern die Diskursinformation beibehalten wird, so dass die nächste Benutzeräußerung zu einem erfolgreichen Finalisieren des Dialogziels führt.

Um die Hold-Strategie zu evaluieren, wurde die Anzahl an abgebrochenen bzw. erfolgreich finalisierten Dialogzielen gezählt. Wie aus der Tabelle 7.8 hervorgeht, erhöht sich die Anzahl an finalisierbaren Dialogzielen mit der Hold-Strategie signifikant: Es können etwa doppelt so viele Dialogziele finalisiert werden.

7.4 Zusammenfassung

Es hat sich gezeigt, dass zwar sowohl die Dialogstrategie des Benutzers als auch die des Roboters Einfluss auf die Kommunikation und ihr Gelingen haben, diese aber durch weitere Fehlerbehandlungsmechanismen für eine effektivere Kommunikation unterstützt werden müssen. Obwohl die Strategie, die der Benutzer während der Kommunikation mit dem Roboter anwendet, das kommunikative Verhalten beeinflusst, hat sie keinen signifikanten Einfluss auf den kommunikativen Erfolg. D.h., es ist umso wichtiger, Fehler in Dialogen vermeiden oder wenigstens beheben zu können. Mit Hilfe von Konfidenzmaßen ist es ferner möglich, Klärungsdialoge anzustoßen, die für Worte bzw. ganze Äußerungen mit einer geringen Konfidenz ermitteln, ob der Benutzer etwas anderes gemeint hat.

	Abgebrochene Dialogziele	Finalisierte Dialogziele
Strategie mit Hold (in %)	84,62	15,38
Strategie ohne Hold (in %)	69,23	30,77

Tabelle 7.8: Rate an abgebrochenen bzw. finalisierten Dialogzielen mit der Hold-Strategie

Konfidenzmaße sind jedoch in unserem grammatikbasierten Spracherkennungsbis-her nicht zuverlässig genug, so dass andere Fehlerbehandlungsmechanismen an ihre Stelle treten müssen.

Die Hold-Strategie ist sicherlich ein erster Ansatz auf dem Weg zur Fehlervermeidung. Allein ist sie aber nicht mächtig genug, da sie keine Möglichkeit bietet, Fehler zu erkennen, sondern nur eine Möglichkeit, die Konsequenzen von Fehlern abzumildern, indem der Benutzer sozusagen eine zweite Chance bekommt. Daher kann sie nur als Zusatzstrategie angesehen werden, die die eigentlichen Strategien, die in den folgenden Kapiteln beschrieben werden, unterstützt.

Die nun folgenden Kapitel beschäftigen sich daher mit Fehlervermeidungsmechanismen. Dabei geht es zunächst um die Auswirkungen einer größeren Grammatik auf die Fehlerklassen, bevor dann auf die Fehlervermeidungsstrategien anhand der drei erläuterten Hauptfehlerkategorien (neue Konzepte, Anaphoraauflösung, komplexe Äußerungen; cf. Kapitel 6) eingegangen wird.

Kapitel 8

Auswirkungen einer größeren Grammatik auf die Fehlertypen

8.1 Einleitung

Da sich in den Benutzertests zeigte (cf. Kapitel 4 und 5), dass die Ursache vieler Fehler in neuen syntaktischen und semantischen Konzepten zu finden ist, wird hier evaluiert, ob und inwiefern dieses Problem in der relativ kleinen Prototypgrammatik begründet liegt und inwiefern sich die Fehlerverteilung mit einer größeren Grammatik verschiebt. Dafür wurde die Grammatik mit Hilfe einer umfangreichen Datensammlung manuell erweitert. Es wird evaluiert, ob sich mit der erweiterten Grammatik der Anteil der neuen Konzepte an der Gesamtmenge der Fehler verschiebt und inwieweit sich die Verteilung der Fehlerklassen innerhalb der neuen syntaktischen und semantischen Konzepte verändert.

8.2 Datensammlung und Erweiterung der Grammatik

Für die Weiterentwicklung der Prototypgrammatik wurden verschiedene Datensammlungen sowohl mit dem Roboter als auch über die Weboberfläche durchgeführt. Um sicherzustellen, dass der Entwicklungskorpus nicht zu stark eingeschränkt ist und dem Testkorpus nur in der Domäne entspricht, nicht aber in den von den Benutzern durchzuführenden Teilaufgaben, bekamen die Benutzer hier andere Aufgaben als in den in den Kapiteln 4 und 5 vorgestellten Benutzertests bzw. gar keine Aufgaben. Ferner nahmen an diesen Benutzertests andere Versuchspersonen, die den Roboter und seine Fähigkeiten noch nicht kannten, teil. Die so gewonnen fast 4000 Benutzeräußerungen wurden für die Version 2 zur Entwicklung benutzt (siehe Tabelle 8.1).

Außerdem sind in der zweiten Version auch Erweiterungen enthalten, die unabhängig von den durchgeführten Datensammlungen und Benutzertests sind und sich auf die Integration weiterer Fähigkeiten des Roboters beziehen, wie beispielsweise die Steuerung der Roboterplattform. Dazu zählen Bewegungen der Arme, Hände, Finger und der gesamten Plattform, Initialisieren und Starten

	1. Version	2. Version
Anzahl Dialogziele	32	97
Anzahl Dialogmoves	152	252
Anzahl Ontologiekonzepte	137	229
Anzahl Grammatikregeln	827	1468
Anzahl Lexikoneinträge	242	410
Anzahl Entwicklungsdaten	–	3723
Parsingrate		
Roboter mit Aufgaben	89,32%	95,31%
Roboter ohne Aufgaben	69,53%	92,71%
Webtest mit Aufgaben	74,62%	92,05%
Webtest ohne Aufgaben	62,60%	90,50%
Turnfehlerrate		
Roboter mit Aufgaben	22,57%	12,11%
Roboter ohne Aufgaben	57,03%	40,16%
Webtest mit Aufgaben	58,20%	29,05%
Webtest ohne Aufgaben	62,40%	44,18%

Tabelle 8.1: Verschiedene Versionen des Prototyps und ihre Fähigkeiten

der einzelnen Bewegungskomponenten, Fahren und Drehen der Plattform, usw. Das erklärt auch, dass sich die Anzahl der Dialogziele zwischen den Versionen 1 und 2 etwa verdreifachte. Getestet wurde dann auf den bereits in den Kapiteln 4 und 5 verwendeten ca. 2000 Benutzeräußerungen.

Benutzeräußerungen im Entwicklungskorpus, die zur Metakommunikation zu zählen sind, wurden nicht in die linguistischen Ressourcen der nächsten Version integriert, sondern gesondert gesammelt, wie im Kapitel 13 erläutert.

8.3 Ergebnisse

8.3.1 Vergleich der Parsing- und Turnfehlerraten

In Tabelle 8.1 ist eine deutliche Verbesserung sowohl in der Parsingrate als auch eine weit geringere Anzahl an falsch verstandenen Turns in der zweiten Version zu sehen. Die Parsingrate liegt für die zweite Version bei etwa 90% (cf. Abbildung 8.1). Die Turnfehlerrate sinkt für Tests ohne Aufgaben auf etwa 42%; für Tests mit Aufgaben liegt sie sogar nur zwischen 12% und 29% (cf. Abbildung 8.2).

8.3.2 Vergleich der Anteile verschiedener Fehlerklassen

Betrachtet man nun die Raten der verschiedenen Fehlerklassen in beiden Versionen, so verschiebt sich, wie erwartet, der Anteil der einzelnen Fehlerklassen (siehe Abbildung 8.3): Der Anteil der Fehler in der Klasse der neuen Konzepte geht leicht zurück, während die Anteile der anderen Fehlerklassen steigen. Größere Veränderungen finden sich jedoch im Bereich der neuen Konzepte selbst:

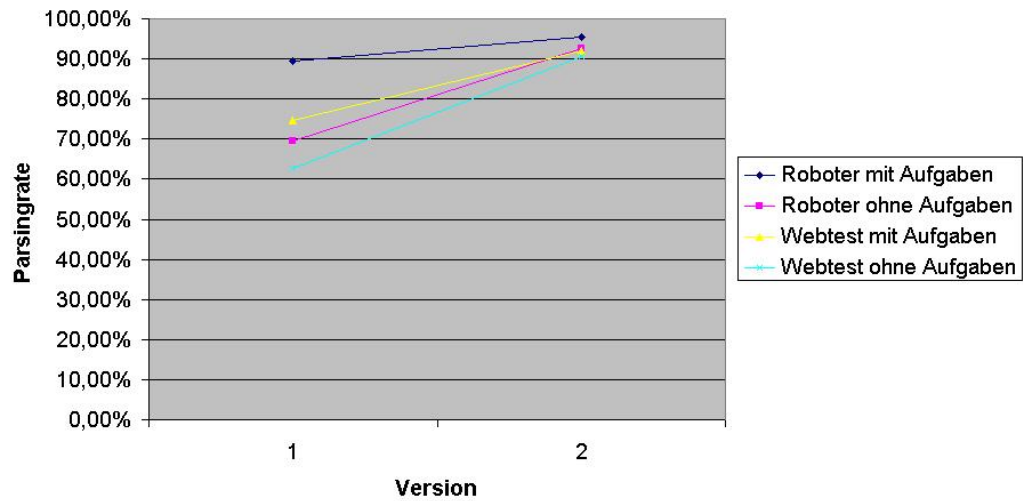


Abbildung 8.1: Vergleich der Parsingraten in der 1. und der 2. Version der Roboterapplikation

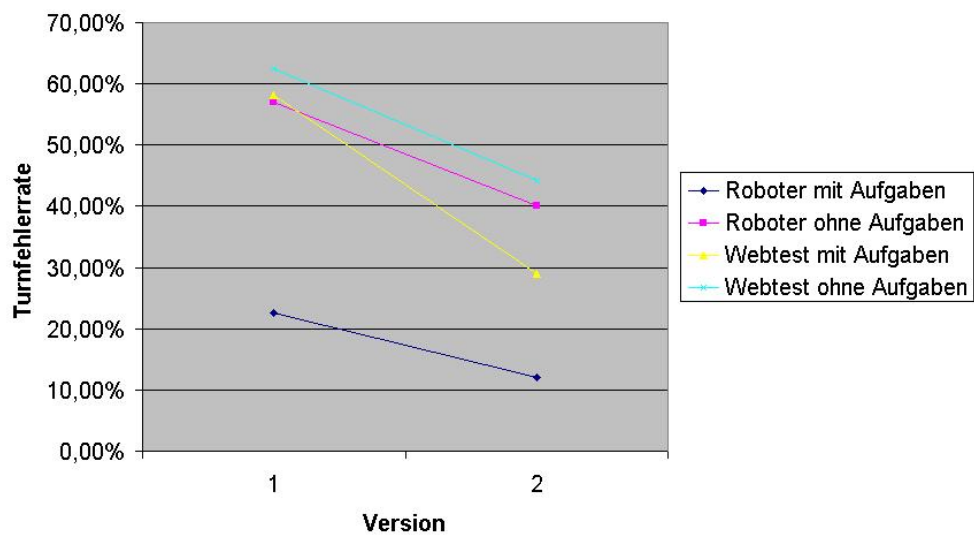


Abbildung 8.2: Vergleich der Turnfehlerraten in der 1. und der 2. Version der Roboterapplikation

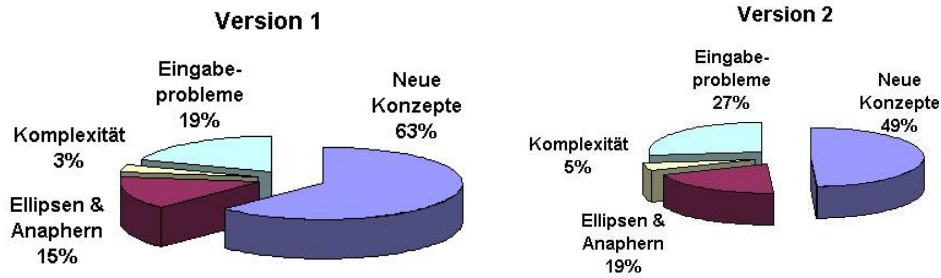


Abbildung 8.3: Vergleich der Anteile der Fehlerklassen in der 1. und der 2. Version der Roboterapplikation

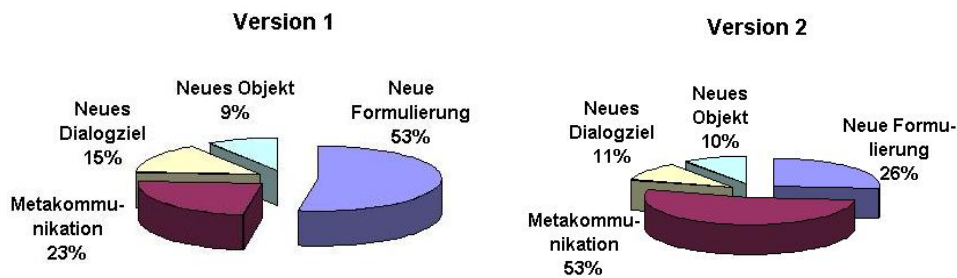


Abbildung 8.4: Vergleich der Anteile der Fehlerklassen innerhalb der neuen Konzepte in der 1. und der 2. Version der Roboterapplikation

Hier steigt der Anteil der Fehler im Bereich der Metakommunikation in der Version 2 erheblich an (siehe auch Abbildung 8.4 und Tabelle 8.2), so dass sich damit der größte Anteil der Fehler in diesem Bereich befindet. Dafür gibt es sehr viel weniger neue Formulierungen und auch weniger neue Dialogziele, da viele bereits durch die zusätzlichen Entwicklungsdaten abgedeckt waren, obwohl diese sich auf andere Aufgaben für die Benutzer bezogen. Die Verteilung der Fehler im Bereich der neuen Objekte ist in beiden Fällen sehr ähnlich, was darauf hindeutet, dass sie mit einer größeren Grammatik auch nur schwer auszumergen sind. Insgesamt lässt sich sagen, dass Metakommunikation bei größeren Entwicklungsdaten eine immer wichtigere Rolle für die Fehlerbehandlung einnimmt, worauf das Kapitel 13 noch genauer eingeht. Neue Objekte spielen auch eine wichtige Rolle bei der Fehlerbehandlung, weshalb hier ein Lösungsvorschlag für weniger Fehler in dem Bereich im Kapitel 9, wo es um die dynamische Erweiterung des Vokabulars geht, dargestellt wird. Fehler im Bereich von neuen Formulierungen und neuen Dialogziele lassen sich hingegen mit einer größeren Grammatik deutlich verringern, weshalb sie in dieser Arbeit nicht weiter thematisiert werden.

Betrachtet man die verschiedenen Formen an Benutzertests im Einzelnen, so fällt auf, dass die Tendenzen über alle Bedingungen gleichermaßen gut zu erkennen sind (cf. Tabelle 8.2): Der Anteil an Metakommunikation steigt in der Version 2 überall, während die Anteile der anderen Fehlerklassen abnehmen. Lediglich im Bereich der neuen Objekte sind die prozentualen Unterschiede von der Version 1 zur Version 2 insgesamt sehr gering und variieren in den unterschiedlichen Testformen leicht. Diese Ergebnisse unterstreichen die Wichtigkeit der

	Roboter ohne Aufg.	Roboter mit Aufg.	Webtest ohne Aufg.	Webtest mit Aufg.
Neue Formulierung				
1. Version	44,44%	70,97%	33,74%	61,35%
2. Version	23,33%	30,0%	13,0%	38,69%
Metakommunikation				
1. Version	40,74%	16,13%	22,09%	14,05%
2. Version	63,33%	60,0%	50,41%	39,42%
Neues Dialogziel				
1. Version	5,56%	12,90%	26,38%	14,86%
2. Version	3,33%	10,0%	20,33%	8,76%
Neues Objekt				
1. Version	9,26%	0%	17,79%	9,73%
2. Version	10,0%	0%	16,26%	13,14%

Tabelle 8.2: Vergleich der Raten im Bereich der neuen syntaktischen und semantischen Konzepte in der 1. und der 2. Version der Roboterapplikation

Integration der Metakommunikation und der Behandlung von neuen Objekten zur Fehlerbehandlung in künftigen Dialogsystemen.

8.4 Zusammenfassung

Erweitert man die Anzahl an Entwicklungsdaten innerhalb einer Domäne, so verschiebt sich die Fehlerverteilung erwartungsgemäß, so dass der Anteil der Fehler in der Klasse der neuen Konzepte zurückgeht (von 63% auf 49%), während die Anteile der anderen Fehlerklassen ansteigen. Die Mehrzahl der Fehler bei den neuen Konzepten liegt nun im Bereich der Metakommunikation (53%), während die Fehler im Bereich von neuen Formulierungen von 53% auf 26% und die im Bereich von neuen Dialogzielen von 15% auf 11% zurückgehen. Fehler im Bereich von neuen Objekten bleiben prozentual etwa gleich und schwanken zwischen 9% und 10%, weshalb sie zur Fehlervermeidung auch ein sehr interessantes Thema darstellen. Da eine manuelle Integration neuer Konzepte und Erweiterung der Grammatiken immer sehr aufwendig ist, geht das folgende Kapitel näher auf die automatische Generierung und Nutzung weiterer linguistischer Ressourcen ein. Dabei geht es darum, mit Hilfe automatischer Verfahren, neue Konzepte zu generieren, um die Grammatik so zu erweitern.

Kapitel 9

Dynamische Erweiterung linguistischer Ressourcen

9.1 Einleitung

Wie wir gesehen haben, lassen sich viele Fehler im Dialog darauf zurückführen, dass das Vokabular des Roboters eingeschränkt ist. Vielfach werden neue Objekte, die der Roboter nicht kennt, oder auch nur neue Formulierungen für bekannte Objekte von den Benutzern gebraucht. Alle Lexikoneinträge und Grammatikregeln müssen im Vorfeld aufwendig manuell erstellt werden und alles abdecken, was der Benutzer sagen könnte. Dieses Kapitel beschäftigt sich daher mit einer Erweiterung des Dialogsystems, die es in die Lage versetzt, dynamisch neues Vokabular hinzuzufügen. D.h., dass neue Informationen aus der Datenfülle des Internets generiert und dann in der Mensch-Roboter-Kommunikation verwendet werden können. Dafür werden die Informationen aus dem Internet zunächst aufbereitet und strukturiert und dann in einer Datenbank, auf die der Dialogmanager zur Laufzeit zugreift, gespeichert.

Als einen Testfall wurde eine Rezeptdatenbank im Internet ausgewählt, die in den Dialogmanager unseres Haushaltsroboters integriert wurde, so dass er jetzt einige tausend Rezepte und Zutaten kennt (Gieselmann, Waibel 2006). Aus der Rezeptdatenbank im Internet werden Informationen zu Rezeptnamen, Zutaten und Kochmethoden heruntergeladen und in einer lokalen Datenbank, die mit dem Dialogmanager verbunden ist, gespeichert.

Im folgenden Abschnitt werden verwandte Arbeiten vorgestellt, die sich ebenfalls mit der dynamischen Integration neuer Konzepte in einen Dialogmanager beschäftigen. Im Anschluss daran werden die Methoden beschrieben, die diese dynamische Vokabularerweiterung ermöglichen, wobei im Besonderen auf die Vererbungsmechanismen im Dialogmanager eingegangen wird, die sich hierfür ausnutzen lassen. Ferner geht es um die Auswirkungen dieser Erweiterung auf den Spracherkennung. Zum Schluss werden Experimente und ihre Ergebnisse erläutert, die den Nutzen dieser dynamischen Vokabularerweiterung für den Benutzer unterstreichen.

9.2 Verwandte Arbeiten

Rayner und seine Mitarbeiter benutzen das bekannte *Plug-and-Play-Konzept* und übertragen es auf Applikationen zum Sprachverstehen in intelligenten Räumen (Rayner et al. 2001). D.h., das System erlaubt eine dynamische Rekonfiguration der Sprachverarbeitungs-komponenten zur Laufzeit, so dass neue Geräte jederzeit angeschlossen und auch per Sprache angesprochen und bedient werden können. Sie benutzen dafür eine Kerngrammatik, die generelle Benutzeräußerungen abdeckt und leicht erweitert werden kann, wenn neue Geräte mit ihren Lexikoneinträgen und ihren Grammatikregeln hinzukommen. Ihr Ansatz ähnelt dem hier verwendeten nur insofern, als dass hier ebenfalls eine Kerngrammatik verwendet wird, die sich auch mit neuen Lexikoneinträgen und Grammatikregeln erweitern lässt. Der große Unterschied liegt aber in der Tatsache begründet, dass sie eine bestimmte Anzahl vordefinierter Lexikoneinträge und Grammatikregeln pro Gerät benutzen, während bei dem hier vorgestellten Ansatz im Vorfeld nicht klar ist, was aus dem Internet zurückkommt. D.h., hier werden keine vordefinierten Lexikoneinträge benutzt, sondern sie werden erst generiert basierend auf der Information, die sich in der Internetdatenbank befindet.

Darüberhinaus gibt es die Arbeiten von Chung et al., die versuchen, die Restriktionen eines vordefinierten, fixen Vokabulars zu überwinden. Sie zeigen ein Dialoginterface mit der Möglichkeit zur dynamischen Integration von neuem Vokabular, um kontext-spezifische Vokabulare, die jederzeit während des Dialogs geändert werden können, zu unterstützen (Chung et al. 2004). Sie benutzen ebenfalls dynamische Informationsquellen aus dem Internet. Als einen Testfall haben sie die Restaurantdomäne ausgewählt und beschränken die dynamische Generierung zunächst auf Restaurantnamen. In dieser Beziehung ähnelt ihr Ansatz dem hier vorgestellten; allerdings wird hier ein informationsbasierter Dialogmanager mit einer Ontologie, die weitergehende semantische Informationen beinhaltet als ihr Finite-State-Ansatz, verwendet. Außerdem benutzen sie ein generelles akustisches Modell für alle neuen Worte, wohingegen hier verschiedene automatisch generierte Modelle für die neuen Worte verwendet werden.

9.3 Verwendete Mechanismen

9.3.1 Vererbungsmechanismen

Wie bereits im Kapitel über die verwendeten Dialogmanagementtools TAPAS (siehe Kapitel 2.2.2) deutlich wurde, wird hier ebenfalls eine Kerngrammatik benutzt, wie sie auch von Rayner und seinen Mitarbeitern verwendet wird (Rayner et al. 2001). Neue Teile der Grammatik können auf Konzepte der Kerngrammatik zugreifen und auch Informationen von dieser Kerngrammatik erben mit Hilfe der Ontologie (Denecke 2000). Ferner können auch komplexe Regeln aus der Kerngrammatik in die applikationsspezifische Grammatik importiert werden.

Beispielsweise erbt der Knoten *obj_recipe* von *generic:object*, was in der Kernontologie definiert ist, und hat verschiedene Instantiierungen, die aus der Datenbank geladen werden, wie in Abbildung 9.1 zu sehen. Ein anderes Beispiel

```

database Recipes obj_recipe jpkg://localhost:5454/Recipe?jpkg {
  dbtable Recipe obj_recipe {
    dbfield name = [generic:NAME];
    dbfield ing1 = [ING1];
    dbfield ing2 = [ING2];
    dbfield ing3 = [ING3];
    dbfield ing4 = [ING4];
  };
};

```

Abbildung 9.1: Beispiel für eine Datenbankschnittstellen-Definition für Rezepte

ist der Knoten *obj_ingredient*, der von *obj_eatable* erbt. *obj_eatable* hinwiederum erbt ebenfalls von *generic:object*.

Auf diese Weise sind sowohl *obj_eatable* als auch *obj_ingredient* Objekte eines Sprechakts, etwas zu essen bzw. etwas zu essen bringen. Wohingegen nur *obj_ingredient* ein Objekt des Sprechakts “Fragen nach einem Rezept anhand einer Liste von Zutaten” ist. Die Instanzen dieser in der Ontologie definierten Objekte werden entweder in der Grammatik oder in einer zusätzlichen Datenbank definiert, wie im Folgenden dargestellt.

9.3.2 Erweiterte Datenbankfähigkeiten

Datenbanken enthalten zusätzliche Informationen für Objekte, wie beispielsweise ihren Zustand oder ihren Ort im Raum (cf. Kapitel 2.2.2). Außerdem ist es damit möglich, einen Teil des Lexikons in einer Datenbank abzulegen, der erst zur Laufzeit in die Grammatik eingebunden wird und damit flexibler als vordefinierte Grammatikregeln ist. Ein Beispiel dafür ist eine solche Liste von Rezepten und den benötigten Zutaten, nach denen der Benutzer fragen kann.

Um eine Frage wie beispielsweise “How can I cook R?” (“Wie kann ich R kochen?”) verstehen zu können, muss der Rezeptname R in der Grammatik enthalten sein. Daher werden während der Grammatikentwicklung Nonterminale mit einem Datenbankimport-Statement definiert, das den Datenbanknamen und den Tabelleneintrag mit seiner lexikalischen und semantischen Repräsentation enthält:

```

<obj_recipe,N,_> = import jpkg://localhost:5454/Env?jpkg
Recipe name generic:NAME import;

```

Eine Benutzeräußerung, wie z.B. “How can I cook Spaghetti Napoli?” (“Wie koche ich Spaghetti Napoli?”) wird zunächst analysiert, von der Grammatik geparkt und mit den vorhandenen Dialogzielen verglichen. Dafür ist das Objekt “Spaghetti Napoli” als ein *obj_recipe* definiert und kann daher mit Hilfe der Datenbank aufgelöst werden, in der sich die verschiedenen Rezeptnamen finden. Auf diese Weise kann die Benutzeräußerung in eine vollständige semantische Repräsentation umgewandelt werden, die dem Dialogservice *Finden des Rezepts R* entspricht, so dass dieser Dialogservice ausgeführt wird, indem der Roboter dem Menschen erklärt, wie man Spaghetti Napoli kocht.

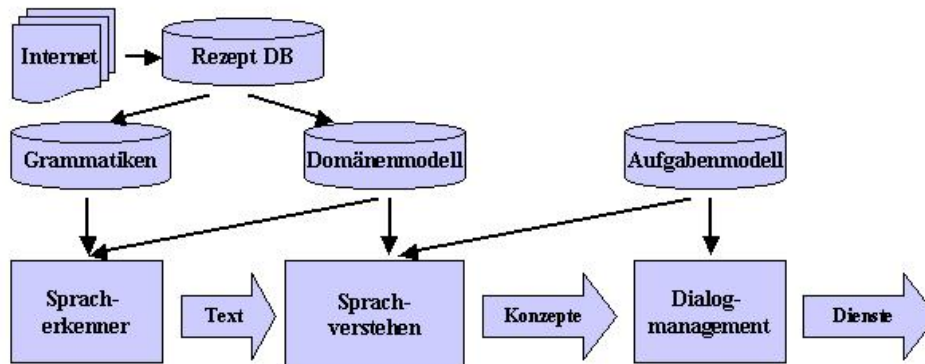


Abbildung 9.2: Architektur der Rezeptapplikation

9.3.3 Auswirkungen auf den Spracherkennung

Da der Dialogmanager und der Spracherkennung die gleiche Grammatik benutzen, um Inkonsistenzen zwischen den verschiedenen linguistischen Ressourcen zu vermeiden, erscheinen damit auch zahlreiche neue Worte im Vokabular des Spracherkenners. Um diese automatisch in die phonetische Umschrift des Spracherkenners transkribieren zu können, wurden Regeln zur Buchstaben-Phonem-Konvertierung benutzt (Black et al. 1998), so dass diese Worte dann auch ins Vokabular des Spracherkenners aufgenommen und damit von ihm erkannt werden können.

9.4 Architektur der Rezeptapplikation

9.4.1 Einleitung

Eine Übersicht über die im Folgenden näher beschriebene Architektur findet sich in Abbildung 9.2: Informationen aus einer Internet-Datenbank werden zunächst aufbereitet und strukturiert und dann in einer Datenbank, auf die der Dialogmanager zur Laufzeit zugreift, gespeichert. Von dort aus werden sie in die Grammatik und die Ontologie eingebunden.

Die Rezeptapplikation ist in der Lage, den Benutzer über einige tausend Rezepte zu informieren. Der Benutzer kann den Roboter nach Rezepten fragen, indem er den Rezeptnamen verwendet, wie z.B. “Robbi, please tell me how to cook Coque au Vin.” (“Robbi, bitte erklär mir, wie ich Coque au Vin mache.”), oder indem er Zutaten angibt: “What can I cook with tomatoes, peppers, and cucumbers?” (“Was kann ich mit Tomaten, Paprika und Gurken kochen?”). In beiden Fällen durchsucht der Dialogmanager seine Datenbank nach den entsprechenden Rezepten. Wenn mehr als eins gefunden wird, gibt der Roboter dem Benutzer eine Liste der möglichen Rezepte zurück bzw. bittet ihn, die Zutaten noch näher zu spezifizieren. Wenn schließlich ein Rezept ausgewählt ist, kann der Benutzer den Roboter nach einer vollständigen Liste der Zutaten fragen und nach der genauen Zubereitung, wie auch in den Beispieldialogen zu sehen:

1. Beispieldialog:

Benutzer: Hi Robbi! (Hallo Robbi!)

Roboter: Hello! My name is Robbi. I am a little robot who can help you in the kitchen. What do you want me to do?
(Hallo! Ich heie Robbi. Ich bin ein kleiner Roboter und kann dir in der Kche helfen. Kann ich irgendwas fr dich tun?)

Benutzer: Robbi, do you know how to cook Spaghetti Bolognese?
(Robbi, weit du, wie man Spaghetti Bolognese kocht?)

Roboter: Yes, I can give you the recipe of Spaghetti Bolognese. Do you want me to list now all the ingredients or to start with the cooking method?
(Ja, ich kann dir das Rezept fr Spaghetti Bolognese geben. Soll ich jetzt alle Zutaten auflisten oder lieber erst die Zubereitung erklren?)

Benutzer: I want to know the ingredients now.
(Ich wsste erst gern die Zutaten.)

Roboter: You need the following ingredients: Spaghetti, beef, oil, tomato and oregano.
(Du brauchst die folgenden Zutaten: Spaghetti, Rinderhack, l, Tomaten und Oregano.)

Benutzer: Tell me how many steps I need for this recipe please.
(Sag mir bitte, wie viele Kochschritte ich fr dieses Rezept brauche.)

Roboter: You need four steps to cook Spaghetti Bolognese. Do you want to start with the cooking method?
(Du brauchst vier Schritte zum Zubereiten von Spaghetti Bolognese. Soll ich dir jetzt die Zubereitung erklren?)

Benutzer: Yes please. (Ja, bitte.)

Roboter: You can cook it in four steps. Step 1: Peel and finely chop the onion. Heat the oil in a frying pan, and fry the onion until it is transparent, then the meat, crumbling it into the pan with your fingers. Add the tomatoes and the puree, salt, pepper and oregano. Mix together well, and simmer over a low heat.
(Du kannst es in vier Schritten kochen: Schritt 1: Schle zuerst die Zwiebeln und schneide sie. Erhitze das l in einer Pfanne, bis es durchsichtig ist und dann flle das Gehackte ein. Fge die Tomaten, das Pree, die Paprika, Salz, Pfeffer und Oregano hinzu. Dann misch alles gut durch und lass es bei schwacher Hitze weiter kcheln.)

Benutzer: Show me the step number two now.
(Zeige mir den zweiten Schritt jetzt.)

Roboter: Step 2: Bring to the boil a large saucepan of salted water. When the water is boiling, put in the spaghetti without breaking it.
(Schritt 2: Bringe einen groen Topf mit Wasser zum Kochen. Wenn es kocht, wirf die Spaghetti hinein,

ohne sie zu zerbrechen.)

2. Beispieldialog:

- Benutzer: What can I cook with tomatoes?
(Was kann ich mit Tomaten kochen?)
- Roboter: I know 10 recipes with tomatoes: I'll give you the first three recipes. Number 1: Spaghetti sauce, number 2: Spaghetti sauce with meatballs, and number 3: Spaghetti al pomodoro e basilico.
(Ich kenne 10 Rezepte mit Tomaten: Ich gebe dir jetzt erstmal die ersten 3. Nummer 1: Spaghettisoße, Nummer 2: Spaghettisauce mit Fleischbällchen, und Nummer 3: Spaghetti al pomodoro e basilico.)
- Benutzer: Tell me the next recipes, please.
(Nenn mir die nächsten Rezepte bitte.)
- Roboter: Number 4: Spaghetti sauce italiano, number 5: Tomato salad, and number 6: Anchovy and cherry tomato bruschetta.
(Nummer 4: Spaghetti mit italienischer Soße, Nummer 5: Tomatensalat, Nummer 6: Sardellen-Kirschtomaten-Bruschetta.)
- Benutzer: I would like the recipe number two please.
(Ich möchte das zweite Rezept bitte.)
- Roboter: Okay! Do you want me to list you now all the ingredients or to start with the cooking method?
(Okay! Soll ich zuerst alle Zutaten auflisten oder mit der Zubereitungsmethode anfangen?)
- Benutzer: Please tell me the complete ingredients list.
(Bitte gib mir alle Zutaten.)
- Roboter: You need the following ingredients: Water for saute. 1 cup chopped onions, 2 cloves garlic, 1 can chopped peeled tomatoes, 1 can tomato sauce, 1 tablespoon parsley, 1 bay leaf, 1/2 teaspoon oregano, 1/4 teaspoon sugar, 1/4 teaspoon salt, 1/8 teaspoon pepper and 1 cup cooked bulgur.
(Du brauchst die folgenden Zutaten: Wasser zum Kochen. 1 Becher geputzte Zwiebeln, 2 Zehen Knoblauch, 1 Dose geschälte Tomaten, 1 Dose Tomatensoße, 1 Teelöffel Petersilie, 1 Lorbeerblatt, 1/2 Teelöffel Oregano, 1/4 Teelöffel Zucker, 1/4 Teelöffel Salz, 1/8 Teelöffel Pfeffer und 1 Becher gekochter Bulgur.)

Da die Zubereitungsschritte sehr kompliziert und lang werden können, so dass der Benutzer den gesprochenen Instruktionen des Roboters eventuell nicht folgen kann, kann der Roboter sie dem Benutzer auch auf einem Bildschirm darstellen.

9.4.2 Die Rezeptdatenbank

Zunächst wurden verschiedene Rezeptdatenbanken im Internet evaluiert und schließlich “<http://fooddownunder.com/>” ausgewählt, weil dies mit mehr als 200 000 Rezepten die größte ist. Die Rezeptseiten aus dem Internet werden geparkt und die Information wird in einer Mysql-Datenbank gespeichert. Auf diese Weise kann ein Online-Zugriff auf die Datenbank zur Laufzeit des Programms vermieden werden, der sehr zeit- und ressourcenaufwendig wäre. Zudem ist die Applikation so unabhängig und kann auch laufen, wenn die entsprechende Rezeptseite im Internet nicht online sein sollte. Der Benutzer kann jedoch den Roboter jederzeit darum bitten, neue Rezepte aus dem Internet herunterzuladen, wenn er beispielsweise das Rezept, nach dem er gerade sucht, nicht in der aktuellen Datenbank findet. Dann wird auf die Internetseiten der Rezeptdatenbank zugegriffen und die Informationen werden in die Mysql-Datenbank geschrieben, so dass sie dem Benutzer beim nächsten Mal wieder zur Verfügung stehen.

Da die Rezeptdatenbank sehr groß ist und die Einträge auch z.T. unterschiedlich strukturiert sind, weil sie z.B. von verschiedenen Benutzern stammen, wurden die Daten zunächst gesäubert und strukturiert. Beispielsweise finden sich dort viele Rezepte doppelt oder auch Rezeptnamen wie “Spaghetti Neopolitana From Dannii Minogue”, die für den Benutzer nicht nachvollziehbar sind, so dass er sie nie benutzen würde. Daher wurden alle doppelten Rezeptnamen ebenso wie alle mit mehr als vier Teilstrings entfernt.

Abhängig von der Größe der Mysql-Datenbank kann die Suche in einigen Sekunden bis hin zu einer halben Minute ausgeführt werden. Dabei lässt sich die Performanz verbessern, wenn die Datenbank kleiner ist, was aber wiederum dazu führt, dass auf weniger Rezepte zugegriffen werden kann. Daher ist es auch möglich, bestimmte Kategorien von Rezepten, wie beispielsweise Cocktailrezepte, nicht aus dem Internet herunterzuladen, um damit die Datenbank nicht zu groß werden zu lassen. Insgesamt enthält die MySQL-Datenbank 30800 Rezepte, da diese Größe eine gute Abdeckung bei gleichzeitiger Datenbankperformanz gewährleistet, so dass der Benutzer nicht auf eine Antwort vom Roboter warten muss, sondern diese in weniger als einer halben Sekunde erhält.

Die Datenbank besteht aus drei Tabellen: Eine Rezept-Tabelle, eine Liste mit allen bedeutungstragenden Worten der Rezeptnamen (d.h. alle Wörter außer Funktionswörtern), und eine Liste mit allen in den Rezepten vorkommenden Zutaten. Die Rezept-Tabelle enthält die Namen der Rezepte, die verschiedenen Zutaten und die benötigten Kochschritte. Die Liste mit den in den Rezeptnamen enthaltenen Worten erlaubt es dem Benutzer nicht nur den exakten Namen des Rezepts, wie er in der Datenbank gespeichert ist, zu benutzen, sondern auch nur einen Teil des Rezeptnamens zu äußern, wie beispielsweise “Spaghetti Carbonara” statt “Spaghetti alla Carbonara”. Die Liste mit den Zutaten wird benutzt, um die benötigte Menge einer Zutat von dem eigentlichen Namen der Zutat abtrennen zu können, so dass der Benutzer durch Angabe einiger Zutaten nach einem entsprechenden Rezept suchen kann, wie im Dialogbeispiel oben zu sehen.

9.4.3 Finden von Rezepten mit Hilfe des Namens bzw. der Zutaten

Um ein Rezept mit Hilfe des Namens zu finden, wird zunächst die Datenbank nach dem exakten Namen durchsucht, den der Benutzer angegeben hat. Wird nichts gefunden, so folgt eine Anfrage an die Datenbank mit den in dem Rezeptnamen enthaltenen bedeutungstragenden Worten. Die Ergebnisse werden dann sortiert nach der Anzahl der gefundenen Worte, so dass das Rezept, das die meisten Worte des vom Benutzer genannten Rezeptnamens enthält, an erster Stelle steht. Insgesamt bekommt der Benutzer höchstens zehn verschiedene Rezepte zurück, um die Kapazität seines Kurzzeitgedächtnisses nicht mit einer Vielzahl an Rezepten zu überlasten. Daher nennt der Roboter dem Benutzer auch immer nur drei Rezepte auf einmal. Der Benutzer hat nun die Möglichkeit, nach weiteren Rezepten zu fragen bzw. seine Suche weiter einzuschränken, indem er noch mehr Zutaten angibt.

Um ein Rezept mit Hilfe der angegebenen Zutaten zu finden, wird ein sehr ähnliches Verfahren benutzt, so dass der Benutzer wiederum eine sortierte Liste zurückbekommt, passend zu den von ihm genannten Zutaten. D.h., es wird eine Anfrage an die Datenbank gestellt mit allen Zutaten und die Ergebnisliste wird sortiert nach der Anzahl der in den Rezepten vorkommenden vom Benutzer genannten Zutaten. Auf diese Weise bekommt der Benutzer wenigstens ein Rezept, das zumindest einer Zutat entspricht, die er angegeben hat. Da es sich wiederum um eine sortierte Liste handelt, erhält der Benutzer zunächst das Rezept mit den meisten ihn interessierenden Zutaten.

Sobald der Benutzer ein Rezept ausgewählt hat, bleibt dieses Rezept im Diskurs des Dialogs, so dass der Benutzer nun auch mit Pronomen oder elliptischen Ausdrücken darauf zugreifen kann, wie auch in dem obigen Beispieldialog zu sehen ist. Der Benutzer kann nach der Art der Zubereitung fragen, sich nach den verschiedenen Schritten der Zubereitung erkundigen oder sich eine Liste aller fehlenden Zutaten nennen lassen.

9.5 Experimente

9.5.1 Einleitung

Es wurden verschiedene Benutzertests durchgeführt, um die Möglichkeiten des erweiterten Systems detailliert zu evaluieren. Dafür wurde zunächst ein textbasierter Test mit sechs Benutzern verschiedener Nationalitäten durchgeführt, in dem sie nach ihren fünf Lieblingsrezepten gefragt wurden. Anhand dessen lässt sich auswerten, inwieweit die Rezeptdatenbank diese Rezeptnamen abdeckt, welche Antworten der Benutzer vom System erhält und ob diese seinen Erwartungen entsprechen. Danach folgte ein Test mit dem Gesamtsystem inklusive des Spracherkenners. Dieser Benutzertest sollte neben der Benutzerzufriedenheit mit dem Gesamtsystem auch evaluieren, inwieweit der Spracherkennung mit den automatisch generierten phonetischen Umschriften der Rezepte und Zutaten zurechtkommt und inwieweit dies die Erkennungsgenauigkeit beeinflusst.

	Rate
Parsing	90,0%
Recall	74%
Präzision	40%
	Durchschnittl. Anzahl
Gefundene Rezepte	9,13
Zur Benutzerintention passende Rezepte	3,71

Tabelle 9.3: Benutzerstudie zur Abdeckung des erweiterten Systems mit zusätzlichen Rezepten

9.5.2 Textbasierter Benutzertest

Die Ergebnisse des textbasierten Benutzertests, in dem die Benutzer ihre fünf Lieblingsrezepte angaben, zeigten, dass 90% der Benutzeräußerungen geparkt und vom System verstanden werden konnten (siehe Tabelle 9.3). Obwohl kein Rezeptname exakt so, wie der Benutzer ihn genannt hatte, in der Datenbank vorkam, so waren doch einzelne Worte der Rezeptnamen vorhanden, so dass jeder Benutzer durchschnittlich etwa 9 Rezepte zurückbekam.

Andererseits entsprachen die Ergebnisse der Suche mit Worten aus den Rezeptnamen häufig nicht den von dem Benutzer gewünschten Resultaten. Beispielsweise führt die Benutzereingabe “Rigatoni al Forno” dazu, dass der Benutzer verschiedene “al forno”-Rezepte bekommt, aber keine mit “Rigatoni”. Dies liegt daran, dass die Benutzereingabe in drei Strings “Rigatoni + al + forno” unterteilt wird, was dazu führt, dass Rezepte, die die beiden Strings “al + forno” enthalten, höher bewertet werden als die, die nur “Rigatoni” enthalten. Ein besonderes Problem ist es dabei, dass viele Rezeptnamen fremdsprachliche Worte enthalten, so dass es auch mit Hilfe eines Part-of-Speech-Taggers sehr schwierig ist, die richtigen Wortarten zu bestimmen, um damit die bedeutungstragenden Worte ermitteln zu können. Wie in Tabelle 9.3 zu sehen, liegt die durchschnittliche Anzahl an zur Benutzerintention passenden Rezepten nur bei 3,7 pro Benutzer. D.h., dass der Benutzer zwar fast immer Rezepte vom System geliefert bekommt, diese aber häufig nicht seiner Intention entsprechen.

Diese Tatsache spiegelt sich auch in den Werten für Präzision und Recall wider. Der Recall lässt sich dabei definieren als die Anzahl der relevanten Rezepte in der Ergebnisliste geteilt durch die Anzahl der relevanten Rezepte in der gesamten Datenmenge. Wohingegen die Präzision angibt, wie viele nicht-relevante Rezepte zurückgegeben werden, was definiert ist als die Anzahl der relevanten Rezepte in der Ergebnisliste geteilt durch die Gesamtzahl der Rezepte in der Ergebnisliste. Wie in Tabelle 9.3 zu sehen, liegt der Recall bei etwa 74%, was bedeutet, dass die meisten relevanten Rezepte auch gefunden werden. Dabei ist zu beachten, dass der Recall sogar bei 100% läge, wenn dem Benutzer nicht nur höchstens 10 Rezepte präsentiert würden, sondern alle relevanten Rezepte. Allerdings bekäme der Benutzer dann, wenn er sich nach generellen Rezepten erkundigt, wie beispielsweise “Pizza”, einige Tausend Rezepte zurück.

Gleichzeitig ist die Präzision mit 40% recht gering, so dass der Benutzer

Turnfehlerrate	Rate gelöster Aufgaben	Anzahl Turns
71,99%	91,67%	580

Tabelle 9.4: Ergebnisse der Benutzerstudie mit dem erweiterten System mit zusätzlichen Rezepten

auch eine ganze Reihe nicht-relevanter Rezepte zurückbekommt, was sich z.T. auf das bereits angesprochene Problem zurückführen lässt, dass das System nicht weiß, welcher Teil des Rezeptnamens der bedeutungstragende ist. Im folgenden Benutzertest mit dem Gesamtsystem soll daher auch evaluiert werden, wie die Benutzer mit solchen irrelevanten Informationen umgehen und ob sie lieber mehr und dafür nicht so relevante Rezepte erhalten oder weniger Rezepte, die dafür alle relevant sind, bevorzugen.

9.5.3 Gesprochensprachlicher Benutzertest

Sechs Benutzer nahmen an dem Test mit dem integrierten System teil, davon ein Drittel weiblich und zwei Drittel männlich. Sie arbeiten alle sehr oft mit einem Rechner, die Hälfte hat schon einen Spracherkenner benutzt, und einer ein Dialogsystem. Die Benutzer haben zunächst frei mit dem Roboter interagiert: Sie wussten nur, dass es sich um einen Haushaltsroboter handelt und sie ihn auch nach Rezepten fragen können. Danach bekamen sie fünf Aufgaben, in denen sie den Roboter nach verschiedenen Rezepten sowohl mit Hilfe von Zutaten als auch mit Hilfe von Rezeptnamen befragen sollten.

Die Benutzerzufriedenheit wurde in einem kurzen Interview nach dem Test ermittelt, um herauszufinden, inwieweit die Benutzer mit dieser erweiterten Version zurecht kommen und wie sie die vom Roboter zurückgelieferten Rezepte beurteilen. Dabei wurden die Benutzer nach ihrer generellen Einstellung zu einer Unterhaltung mit einem Roboter in natürlicher Sprache gefragt, was sie sich noch von einem solchen Roboter erwarten würden, ob sie glauben, dass er wirklich hilfreich ist, und ob sie ihn tatsächlich benutzen würden.

Obwohl nur 28,01% aller Benutzeräußerungen vom Roboter richtig verstanden und in die korrekte Semantik übertragen werden konnten, gelang es den Benutzern, die meisten Aufgaben zu lösen (cf. Table 9.4). Nur ein Benutzer wurde nach mehreren aufeinanderfolgenden Fehlerkennungen immer nervöser und brach daher einige Aufgaben vorzeitig ab. Alle anderen haben den Test bis zum Ende durchgeführt. Die Benutzer haben ihre Äußerungen zum Teil mehrfach wiederholt, um zu sehen, ob sich die Antworten vom System ändern. Viele Benutzer fingen auch an zu hyperartikulieren, wenn sie einmal nicht verstanden wurden, was die gesamte Erkennungsrate verschlechterte.

Die Ergebnisse des gesprochensprachlichen Benutzertests zeigen ferner, dass die Benutzer mit dem System zufrieden sind (siehe Tabelle 9.5): Generell gefällt es allen Benutzer, sich mit einem Roboter in natürlicher Sprache zu unterhalten. Einige sind skeptisch, was die sprachlichen Fähigkeiten des Roboters angeht, und versuchen ihn im Test auch mit immer neuen Formulierungen aufs Glatteis zu führen. Die Mehrheit der Benutzer war der Meinung, dass der Roboter erst

	Ja	Nein	keine Antwort
Kann der Roboter beim Kochen helfen?	16,67%	16,67%	66,67%
Waren die Antworten des Roboters verständlich?	83,33%	16,67%	0%
Würden Sie den Roboter benutzen?	83,33%	16,67%	0%

Tabelle 9.5: Auswertung der Benutzerbefragung

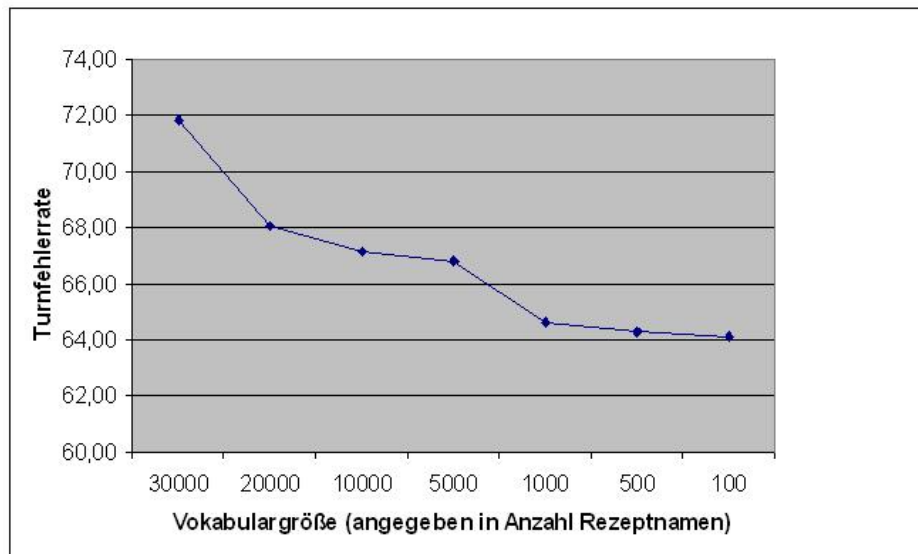


Abbildung 9.3: Entwicklung der Turnfehlerraten bei unterschiedlichen Vokabulargrößen

besser verstehen lernen und langsamer sprechen sollte, bevor er tatsächlich beim Kochen helfen kann (cf. Tabelle 9.5). Dennoch würden die meisten Benutzer ihn als Alternative zu einem Kochbuch benutzen, weil man mit ihm einfach Rezepte nach Zutaten suchen kann, was in einem Kochbuch sehr schwierig ist. Nur 16% sehen den Roboter nicht als Alternative zum Kochbuch, da sie die Verständigung für zu aufwendig und langwierig halten (cf. Tabelle 9.5). Ferner zeigte die Befragung, dass die Benutzer lieber eine größere Menge Rezepte zurückbekommen, auch wenn diese nicht alle ihrer ursprünglichen Intention entsprachen. Sie bevorzugten es, selbst in diesen Rezepten weiter zu stöbern und sich davon inspirieren zu lassen, wie es in einem Kochbuch auch der Fall ist.

Die hohe Turnfehlerrate (cf. Tabelle 9.4) erklärt sich vor allem aus der sehr großen Perplexität, die durch die riesige Menge an Rezeptnamen entstanden ist. Da diese ca. 30 000 Rezeptnamen alle laut Grammatik an derselben Stelle stehen können, muss der Spracherkenner hier allein aufgrund der Akustik sich für einen Rezeptnamen entscheiden, was zu der schlechten Erkennungsgenauigkeit führt. Um die Auswirkungen der Perplexität näher untersuchen zu können, wurde das Vokabular verkleinert und die Turnfehlerraten für verschiedene Vo-

abulargrößen von 100 - 30 000 Rezeptnamen evaluiert. Es zeigt sich, dass die Turnfehlerrate abnimmt mit abnehmender Vokabulargröße, wobei eine optimale Vokabulargröße etwa bei 1000 Rezeptnamen zu finden ist mit einer Turnfehlerrate von 64,09% (cf. Abbildung 9.3). Danach ist das Vokabular wiederum zu klein, um alle verwendeten Rezeptnamen abzudecken.

9.6 Zusammenfassung

Mit Hilfe zusätzlicher Wissensquellen ist es möglich, die linguistische Wissensbasis des Roboters zu erweitern. Auf diese Weise kann dynamisch neues Vokabular hinzugefügt werden. Benutzertests haben gezeigt, dass damit die Abdeckung verbessert werden kann und die Benutzer den Dialog insgesamt sehr positiv bewerten (83,33% würden einen solchen Roboter benutzen). Allerdings sinkt die Erkennungsgenauigkeit bei einer großen Menge neu hinzugefügter Worte, da die zusätzlichen Worte nur an wenigen ausgewählten Stellen der Grammatik stehen können und damit an diesen Stellen eine sehr große Perplexität vorliegt. So liegt die Erkennungsgenauigkeit der Turns für 30 000 Rezepte nur bei 28,01%, während sie sich für 1000 Rezepte auf 35,91% steigern lässt.

Hinzu kommt, dass für jedes Konzept, das auf diese Weise dynamisch neue Informationen aus dem Internet bezieht, eine Reihe von spezifischen Regeln nötig sind, um dies zu ermöglichen: Zunächst müssen die entsprechenden Informationen im Internet gefunden, dann geparkt und strukturiert werden, um in die Datenbank des Dialogmanagers eingebunden werden zu können. Diese Konzepte müssen ferner in der Ontologie eingebunden sein, um in der Grammatik als Terminale zur Verfügung stehen zu können. Dies ist zwar einfacher, als die entsprechenden Einträge manuell zu erstellen, setzt aber dennoch eine strukturierte Datenbank voraus. Das folgende Kapitel beschäftigt sich daher damit, inwieweit sich unstrukturierte Datenbanken, wie das Internet, direkt in den Dialogmanager einbinden lassen.

Kapitel 10

Integration des Internets als Wissensbasis

10.1 Einleitung

Wie sich in den Benutzertests gezeigt hat, gehen die Benutzer davon aus, dass der Roboter allwissend ist: Sie stellen Fragen wie “Why is the sky blue?”, “What’s the wheather like in Chicago?” (“Warum ist der Himmel blau?”, “Wie ist das Wetter in Chicago?”). Daher wurde ein web-basiertes Frage-Antwort-System (QA-System) eingebaut, das all die Benutzeräußerungen, die von einem QA-System beantwortet werden können, bekommt, und dem Benutzer mit Hilfe einer Textausgabe einen Antwortsatz zurückliefert. Damit lässt sich das Internet als eine weitere Wissensbasis in das Dialogsystem integrieren.

10.2 Verwandte Arbeiten

QA-Systeme gibt es bereits seit den 60er Jahren. Damals waren sie noch auf eine Domäne beschränkt und arbeiteten mit einer großen manuell erstellten Datenbank im Hintergrund, wie z.B. BASEBALL, ein System über die amerikanische Baseballliga und ihre Ergebnisse einer Saison (Green et al. 1963).

Mittlerweile sind QA-Systeme nicht mehr auf eine Domäne begrenzt und verwenden eine unstrukturierte Ansammlung von Dokumenten zum Beantworten der Fragen, wie beispielsweise das Internet oder große Zeitungskorpora. AnswerBus (Zheng 2002) ist z.B. ein solches web-basiertes System, das mit Anfragen in den verschiedensten Sprachen umgehen kann, die mit dem online-Übersetzungssystem Babelfish von Altavista ins Englische übersetzt werden. Aus diesen Anfragen werden dann Funktionsworte und sehr häufige Worte entfernt und die Verben werden in ihre entsprechende Form in einer möglichen Antwort umgewandelt. Diese Liste von Schlüsselworten wird an verschiedene Suchmaschinen im Internet geschickt, wie z.B. Google und Yahoo, und die besten Dokumente, die zurückgegeben werden, werden geparkt und in ihre einzelnen Sätze zerlegt. Diese Sätze werden nun evaluiert anhand verschiedener Faktoren, wie der Anzahl an vorkommenden Schlüsselworten oder ihrer Position in den

Suchergebnissen, und es wird eine sortierte Liste an Antwortsätzen zurückgegeben.

Das von Microsoft Research entwickelte web-basierte QA-System AskMSR macht sich vor allem die Redundanz des Internets zunutze und verzichtet auf linguistisch aufwendige Techniken (Banko et al. 2002; Brill et al. 2002). Zunächst werden die Benutzeranfragen umformuliert in Phrasen, wie sie in den Antworten vorkommen können. Diese Phrasen werden an Internetsuchmaschinen geschickt und die Schnipsel, die die Suchmaschinen als Ergebnisse zurückliefern, in 1-, 2- und 3-gramme unterteilt. Diese n-gramme werden verglichen mit dem erwarteten Antworttyp der Frage.

Das LAMP-System (Zhang, Lee 2002) benutzt Textmuster, um Fragen zu klassifizieren und Schlüsselphrasen aus der Frage zu extrahieren. Für jede Klasse gibt es einige Vorlagen, die beschreiben, wie eine Frage dieser Klasse aussehen kann. Daneben gibt es für jede Frageklasse auch eine Menge an Textmustern, die benutzt werden, um Antworten aus den Textschnipseln, die die Suchmaschinen zurückliefern, zu extrahieren. Diese Muster werden automatisch gelernt und angewandt mit Hilfe von Frage-Antwort-Paaren aus vergangenen TREC-Evaluationen (Text REtrieval Conference (TREC): Weitere Details sind auf der Webseite zu finden: <http://trec.nist.gov/>).

Alle diese Systeme haben interessante Aspekte, die in dem hier verwendeten QA-System EPHYRA (Schläfer et al. 2006) kombiniert werden, um ihre Vorteile auszunutzen unter gleichzeitiger Vermeidung ihrer Nachteile: So ähnelt die Architektur von EPHYRA in groben Zügen AnswerBus. Auch die Umformulierung der Fragen, Erkennung der Satzteile und Auswertung der Antworten in eine sortierte Liste funktioniert in ähnlicher Weise in AnswerBus und in AskMSR. Der in LAMP verwendete Musterlernansatz wurde in EPHYRA erweitert und verallgemeinert, so dass eine größere Anzahl generischer Muster gelernt werden kann. Aranea (Lin, Katz 2002) ist eine interessante Kombination aus Wissensanalyse und -annotierung, wie sie in ähnlicher Weise auch in EPHYRA benutzt wird.

Ferner arbeiten verschiedene Frage-Antwort-Systeme heutzutage schon mit einer gesprochenen Eingabe und einer Anbindung an ein Dialogsystem (op den Akker et al. 2005; Hori et al. 2003). Dabei wird die Fähigkeit des Dialogsystems zu weiteren Nachfragen für die Klärung einer Benutzeranfrage ausgenutzt, so dass diese näher spezifiziert werden kann. Als eine der größten Herausforderungen stellt sich dabei vor allem die gesprochensprachliche Eingabe ohne Domänenbeschränkung heraus, die nach wie vor noch keine befriedigenden Ergebnisse liefert.

10.3 Das QA-System EPHYRA

Unser QA-System EPHYRA (Schläfer et al. 2006) ist ein modulares erweiterbares Framework, das es erlaubt mehrere QA-Ansätze zu integrieren. EPHYRA arbeitet web-basiert und ist für alle Domänen geeignet (cf. engl. open-domain). Wie in der Gesamtarchitektur in Abbildung 10.1 zu sehen, gibt es die drei klassischen QA-Komponenten: Analyse der Fragen, Information Retrieval (IR) und Antwortgenerierung. Die Komponente, die die Fragen analysiert, nimmt als In-

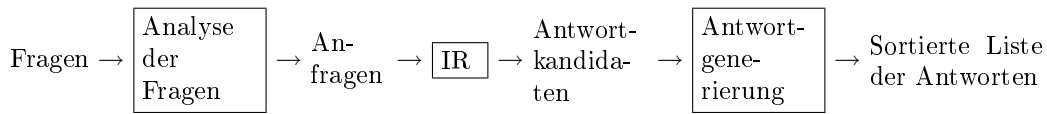


Abbildung 10.1: Architektur des QA-Systems EPHYRA

put einen Fragestring und überführt ihn in eine oder mehrere Anfragen, die ans Information Retrieval weitergegeben werden. Die Aufgabe des Information Retrievals ist es nun im Internet nach Antworten zu suchen. Die Ergebnisse werden weitergegeben an die Antwortgenerierung, wo unwahrscheinliche Antworten herausgefiltert werden und eine Liste mit sortierten Antworten zurückgegeben wird.

Innerhalb jeder dieser drei Komponenten lassen sich verschiedene QA-Ansätze modular integrieren. Derzeit sind drei verschiedene Ansätze enthalten: Wissensannotierung (cf. engl. knowledge annotation), Musterlernen und Formulierungen von Anfragen ans Internet als Backup-Strategie. Allen gemeinsam ist eine generelle Analyse der Frage, in der Funktionsworte entfernt und Formulierungen verändert werden, und eine gemeinsame Antwortgenerierung, in der unwahrscheinliche Antworten herausgefiltert und alle anderen sortiert werden.

10.3.1 Wissensannotierung

Wissensannotierung wird für häufige Fragen und solche, die mit generischen Methoden schwierig zu beantworten sind, benutzt, wie z.B. Definitionsfragen oder auch Fragen nach dem aktuellen Wetter. Die Antworten werden direkt aus strukturierten Webseiten, wie beispielsweise Wikipedia oder dem CIA World Factbook, oder Webservices, wie z.B. einem Wetterservice, gewonnen.

Eine Datei mit Fragemustern in Form von regulären Ausdrücken dient während der Analyse der Frage dazu herauszufinden, ob für eine bestimmte Frage Wissensannotierung verwendet werden kann. Jedes Muster enthält einen Ausdruck, der den relevanten Inhalt der Frage identifiziert, um in den entsprechenden Quellen nachsehen zu können.

Da es einen großen manuellen Aufwand bedeutet, solche strukturierte Quellen zu verwenden, weil die Webseiten unterschiedlich aussehen und damit verschiedene Parser benötigen, um die Antworten extrahieren zu können, lohnt sich dieser Ansatz nur für häufige Fragen. Für andere Fragen wurde daher der Ansatz des generellen Musterlernens entwickelt neben einer Backup-Strategie, die mit Umformulierungen von Anfragen arbeitet.

10.3.2 Musterlernen

Beim Musterlernen werden Textmuster gelernt, die auf Textpassagen aus der Antwortextraktion angewandt werden. Dazu wird das System trainiert auf Frage-Antwort-Paaren, die mit Hilfe von konventionellen Suchmaschinen gefunden werden können. Dabei wird während der Analyse der Frage ein neuer Ansatz zur Interpretation der Frage benutzt, der von der ursprünglichen Formulierung

abstrahiert, indem er davon ausgeht, dass eine Frage nach einer bestimmten *Eigenschaft* eines *Ziels* in einem speziellen *Kontext* fragt, wie an dem folgenden Beispiel zu sehen:

Who won the Nobel Peace Prize in 1988?
 PROPERTY: WINNER
 TARGET: Nobel Peace Prize
 CONTEXT: 1988

Aus der Frage des Benutzers nach dem Gewinner des Friedensnobelpreises 1988 wird also das Ziel “Friedensnobelpreis”, die Eigenschaft “Gewinner” und der Kontext “1988” extrahiert. Während der Informationssuche im Internet wird nach Textpassagen gesucht, die das Ziel und möglichst auch die Eigenschaft und den Kontext, enthalten. Aus den gefundenen Textpassagen wird in der Antwortgenerierung nun die gesuchte Information mit Hilfe eines entsprechenden Antwortmusters rausgezogen.

Für jede Eigenschaft, wie z.B. WINNER, DATE, NAME, wird zunächst manuell eine Menge an Fragemustern spezifiziert. Diese Fragemuster werden auf eine Benutzeranfrage angewandt, um Ziel, Eigenschaft und Kontext extrahieren zu können. Auf der anderen Seite gibt es für jede Eigenschaft auch ein Antwortmuster, was benötigt wird, um aus einer im Internet gefundenen Textpassage wiederum Ziel, Eigenschaft und Kontext extrahieren zu können, um die Frage zu beantworten. Diese Antwortmuster werden automatisch gelernt mit Hilfe von Frage-Antwort-Paaren aus älteren TREC-Evaluationen als Trainingsbeispiele.

10.3.3 Umformulierung der Benutzeranfragen

Als Backup-Strategie wird die Frage des Benutzers während der Analyse mit Hilfe verschiedener linguistischer Techniken umformuliert und von der Information-Retrieval-Komponente an Google und Yahoo geschickt. Dabei wird versucht, die Formulierungen zu finden, die wahrscheinlich in den Textpassagen, die die Antwort enthalten, vorkommen. Zusätzlich, um sicherzustellen, dass zumindest irgendein Ergebnis zurückgeliefert wird, wird eine Menge von Schlüsselwörtern (cf. engl. bag of words) kreiert und auch an die Suchmaschinen geschickt.

Zunächst werden die in der Benutzeräußerung vorkommenden Worte in ihre entsprechende Form in einer möglichen Antwort umgewandelt. Hilfsverbkonstruktionen, wie sie in Fragen vorkommen, werden beispielweise durch die entsprechenden Verbformen ersetzt, und die Satzstellung wird an die in einem Aussagesatz vorliegende angepasst. Um zu überprüfen, ob die Frage ein Hilfsverb enthält und welches das Hauptverb des Satzes ist, wird ein Part-of-Speech-Tagger eingesetzt (OpenNLP 2005).

Schließlich wird noch eine Menge von Schlüsselwörtern erstellt, indem alle bedeutungstragenden Worte aus der Benutzeranfrage extrahiert werden. Liegt die Anzahl der gefundenen Worte oberhalb eines festgelegten Schwellwertes, so werden die häufigsten Worte gelöscht, wozu ein Frequenzwörterbuch auf der Basis von Zeitungsartikeln herangezogen wird. Damit erreicht man eine möglichst

präzise Anfrage an die Suchmaschinen, die nicht zu viele Worte enthält, um den Suchprozess nicht unnötig zu verlängern.

10.3.4 Antwortgenerierung

Während der Antwortgenerierung werden unwahrscheinliche Antworten ausgefiltert. Dies sind z.B. Antworten, die nur aus Funktionsworten oder Zeichen wie Klammern oder Anführungszeichen bestehen, oder die selbst wieder Fragen darstellen. Um die mit Hilfe der verschiedenen Ansätze gefundenen unterschiedlichen Antworten gewichten zu können und nur die beste an einen Benutzer des Dialogsystems weiterzugeben, werden die Antworten innerhalb der Antwortgenerierung mit verschiedenen Gewichten versehen und sortiert. Dabei erhalten Antworten, die mit Hilfe der Wissensannotierung gefunden wurden, die höchsten Werte, da sie präzise auf die Benutzeranfrage antworten. Danach folgen Antworten, die mit Hilfe des Musterlernens erstellt wurden, und schließlich Antworten, die durch die Backup-Strategie ermittelt wurden.

10.4 Integration von QA-System und Dialogmanagement

10.4.1 Architektur von EPHYRA

EPHYRA bietet verschiedene Schnittstellen: Ein Kommandozeileninterface, ein Webinterface und eine Schnittstelle zum ONE4ALL-System, in das auch der Dialogmanager des hier verwendeten Roboters integriert ist. Damit ist es möglich, das System in den unterschiedlichsten Konfigurationen zu nutzen. Da es im Folgenden um die Integration in das hier betrachtete Robotersystem geht, wird hier nur das ONE4ALL-System und die Möglichkeit der Integration in dieses ausführlich beschrieben.

10.4.2 Das ONE4ALL-System

Das ONE4ALL-System ist ein Toolkit, das es erlaubt, Sprachapplikationen mit dem Spracherkenner JANUS (Finke et al. 1997) zu erstellen. Als Standardkomponenten sind dort ein zentraler Kommunikationsserver, der Spracherkenner JANUS und ein Receiver, der die Benutzeräußerung aufnimmt, enthalten. Darüberhinaus sind in dem hier beschriebenen Szenario auch eine Komponente zur Textausgabe, ein Sprachsegmentierer, die TAPAS Dialogmanagementtools und das QA-System EPHYRA angeschlossen, wie auf Abbildung 10.2 zu sehen.

Der Kommunikationsserver ist verantwortlich für die Kommunikation zwischen den verschiedenen Komponenten. Daher melden sich alle Komponenten bei ihm als Agenten an, um Mitteilungen an den Server schicken und auch Mitteilungen vom Server bekommen zu können. Die Kommunikation ist eine Blackboard-Architektur, wobei alle Komponenten sich mit Hilfe einer Socket-Verbindung mit dem Kommunikationsserver verbinden.

Innerhalb dieser Gesamtarchitektur entscheidet also zunächst der Sprachsegmentierer, ob eine Benutzeräußerung vorliegt. Ist dies der Fall, dann ist es

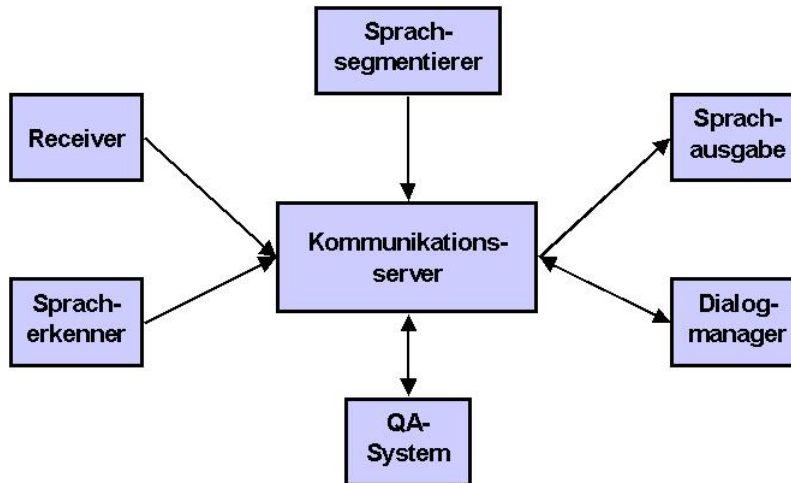


Abbildung 10.2: Die Architektur des Gesamtsystems ONE4ALL

die Aufgabe des Spracherkenners, diese Benutzeräußerung zu erkennen und an den Dialogmanager weiterzugeben. Der Dialogmanager entscheidet, ob eine Äußerung ans QA-System weitergegeben wird, damit dies die Frage beantwortet, oder ob es sich um eine Äußerung aus der Küchen-domäne handelt, die innerhalb des Dialogmanagers selbst behandelt werden kann. Wird eine Äußerung an das QA-System EPHYRA weitergegeben, so ermittelt dies die beste Antwort auf die gegebene Frage. Diese beste Antwort wird an den Kommunikationsserver zurückgeschickt, der es an die Textausgabe weitergibt. Bleibt der Score der besten Antwort jedoch unterhalb eines bestimmten Schwellwerts, so wird eine Mitteilung an den Dialogmanager geschickt, damit der einen Klärungsdialog initiieren kann.

Für die Interaktion des Dialogmanagers mit EPHYRA kann man entweder einen statistischen Spracherkennung benutzen, um damit sowohl die Küchen-domäne als auch die generellen Wissensfragen abzudecken, oder einen grammatik-basierten Spracherkennung mit einer um die Fragedomäne erweiterten Grammatik, oder eine Kombination aus beidem. Beides hat Vor- bzw. Nachteile, weshalb sich eine Kombination beider Ansätze anbietet:

- *Statistischer Spracherkennung*: Hier wird innerhalb des Spracherkenners ein allgemeines Sprachmodell verwendet, um möglichst viele Äußerungen abdecken zu können. Der Dialogmanager dahingegen arbeitet mit einer Grammatik, die auf die Küchen-domäne beschränkt ist und klassifiziert alle Äußerungen, die er nicht parsen kann als Fragen, die an das QA-System weitergeschickt werden.

Diese Annahme hat den Nachteil, dass alle Äußerungen, die nicht aus der Küchen-grammatik stammen, von dem QA-System beantwortet werden sollen. Aufgrund von Fehlern in der Sprachsegmentierung werden jedoch häufig auch Atem- oder Hintergrundgeräusche als Äußerungen klassifiziert, die dann nicht von der Küchen-grammatik geparkt werden können

und damit beim QA-System landen, das diese ebenfalls nicht sinnvoll auflösen kann. Hinzukommen Spracherkennungsfehler, die dazu führen, dass Sätze, die eigentlich in der Robotergrammatik enthalten sind, vom allgemeinen Sprachmodell falsch erkannt und damit vom Dialogmanager auch falsch klassifiziert werden. Hier würde es sich daher anbieten, Konfidenzen zu nutzen und nur Äußerungen mit hohen Konfidenzwerten an EPHYRA weiterzugeben, um so beispielsweise Geräusche herausfiltern zu können.

- *Grammatikbasierter Spracherkenner*: Hier benutzen Spracherkenner und Dialogmanager die gleiche Grammatik. Dafür wird die aktuelle Robotergrammatik erweitert um ein Modul, das mögliche Benutzeranfragen enthält. Die Terminale kommen dabei für die Eigennamen (cf. engl. named entities) aus der gleichen Liste, die auch innerhalb von EPHYRA benutzt wird. Die anderen Elemente der Grammatik orientieren sich an den in EPHYRA bereits definierten regulären Ausdrücken und den Fragen aus dem TREC-Korpus. Innerhalb des Dialogmanagers gibt es ein Dialogziel, das die jeweilige Anfrage eines Benutzers an EPHYRA beinhaltet. Ist dieses Dialogziel finalisiert, so wird die Frage an EPHYRA weitergegeben. Fehlen noch Informationen vom Benutzer, weil er entweder eine zu allgemeine Frage gestellt hat oder Teile der Frage nicht korrekt erkannt werden konnten, so wird ein Klärungsdialog initiiert.

Hierin liegt auch der entscheidende Vorteil dieses Ansatzes: Obwohl es sicherlich sehr viele Fragen gibt, die von einer solchen Grammatik nicht abgedeckt werden, ist es mit Hilfe von Klärungsdialogen für den Benutzer möglich, seine Frage interaktiv näher zu spezifizieren. Dies ist möglich durch die semantische Analyse der Benutzeräußerung, wie sie von der Grammatik geleistet wird.

- *Statistischer Spracherkenner & semantische Analyse*: Eine Kombination beider Ansätze beinhaltet einen statistischen Spracherkenner, der auch die ausgerollte Grammatik des Dialogmanagers enthält, und eine regelbasierte semantische Analyse innerhalb des Dialogmanagers, die neben der Küchengrammatik auch eine Analyse der QA-Fragen beinhaltet, um Klärungsdialoge initiieren zu können.

Generell stellen Eigennamen ein großes Problem dar, da sie häufig in den Fragen vorkommen und oft nicht im Vokabular des Spracherkenners enthalten sind. Daher wurden zusätzliche Lexikoneinträge aus den Listen generiert, die auch innerhalb von EPHYRA zu ihrer Klassifizierung benutzt werden. Für alle diese Eigennamen wurden mit Hilfe von Graphem-Phonem-Konvertierungen Einträge für das Lexikon des Spracherkenners erstellt, ähnlich wie dies auch bereits für die neuen Rezeptnamen im vorangegangenen Kapitel beschrieben wurde. Auch hier ergibt sich damit wiederum das Problem der großen Perplexität angesichts der Tatsache, dass die Listen der Personen-, Organisations- und Ortsnamen insgesamt ca. 300 000 Einträge enthalten (Personennamen: 39 078, Organisationsnamen: 45 454, Ortsnamen: 211 960). Daher wurden aus diesen Listen zufällig Einträge ausgewählt, so dass insgesamt ca. 65 000 Eigennamen im Lexikon verblieben und mit ihnen die folgenden Tests durchgeführt.

Darüberhinaus können beide Arten von Spracherkennern niemals alle Fragen abdecken. Hier bietet es sich an, mit Konfidenzen zu arbeiten, so dass der Dialogmanager in Zweifelsfällen nachfragen kann. Darin liegt auch die Stärke der Kombination beider Ansätze, so dass Fragen hier interaktiv beantwortet werden können. Ferner stellt sich bei einer Kombination von Dialogmanager und QA-System generell das Problem, dass die Ausgaben an das QA-System keine Groß- und Kleinschreibung und Interpunktionszeichen mehr enthalten, wie dies in geschriebenen Fragen der Fall ist. Dies erschwert das Tagging der Satzteile, was zu Fehlinterpretationen führen kann. Dieses Problem und seine Lösungsmöglichkeiten werden im Folgenden näher beleuchtet, wenn es um die Evaluation des Gesamtsystems geht.

10.4.3 Erstellung einer QA-Grammatik mit Hilfe der Ressourcen aus Ephyra

Für diese Grammatik wurden die vorhandenen regulären Ausdrücke, die in der Wissensannotierung, der Frageumformulierung und dem Lernen von Fragemustern verwendet werden, in kontextfreie Grammatikregeln umgewandelt. Dabei wird nicht nur die Syntax der Fragen berücksichtigt, sondern auch eine semantische Auflösung anhand der Fragemuster vorgenommen, um damit gegebenenfalls Klärungsdialoge initiieren zu können.

Da die Frageumformulierungen sich auf keine semantisch definierten Fragen beziehen, sondern eine Backup-Technik darstellen, dienen sie zur Definition möglicher syntaktischer Konstruktion. Die regulären Ausdrücke, die in der Wissensannotierung angegeben sind, werden demgegenüber umgeformt in kontextfreie Grammatikregeln und werden der semantischen Klasse der "Wissensannotierung" zugeordnet.

Ebenso werden die regulären Ausdrücke der Fragemuster in kontextfreie Grammatikregeln umgewandelt, deren Bedeutung durch ihre Eigenschaft bestimmt wird. Für die Eigenschaft "ABBREVIATION" sind in EPHYRA die folgenden regulären Ausdrücke definiert:

```
<what> is the (abbreviation|acronym|chemical
(formula|name|symbol)) for <T>
how <T> <be> abbreviated
how you abbreviate <T>
```

Diese werden in die folgenden Grammatikregeln umgewandelt:

```
<act_askQuestionAbbrev,V,_> =
what <be> the (abbreviation|chemical
(formula|name|symbol)) for <obj_name,NP,_>
| how can <obj_name,NP,_> be abbreviated
| how do you abbreviate <obj_name,NP,_>;
```

Viele Fragen sind allerdings nicht vollständig definiert, wie an dem folgenden regulären Ausdruck zur Umformulierung der Frage darüber, wie etwas war, deutlich wird:

(?i)how (is|are|was|were) (.*)

Hier fehlt am Ende die Angabe des Subjekts. Da die Grammatikregeln für den Spracherkenner vollständig definiert sein müssen, wurde an dieser Stelle ein Eigennamen hinzugefügt, wobei es sich entweder um einen Ort oder um einen Personennamen handeln kann. Die entsprechende Grammatikregel lautet daher:

<act_askQuestion,V,_> =
how (is|are|was|were) (<obj_location,NP,_>|<obj_name,NP,_>);

Die Eigennamen stammen aus der im vorangegangenen Abschnitt erläuterten Liste der Eigennamen, die auch in EPHYRA selbst benutzt wird.

Ebenso ergibt sich bei den Fragemustern des Lernansatzes das Problem, dass das Ziel in den Mustern nicht definiert ist, und daher auch in den Grammatikregeln ergänzt werden muss. D.h. der folgende reguläre Ausdruck:

what <T> (eats?|ate)

wird in diese Grammatikregel umgewandelt:

what [kind of] food (eats|ate) <obj_name,NP,_>

In diesem Fall gibt der Name des Musters - food - Aufschluss darüber, was an der Stelle des Ziels stehen kann. Außerdem wird am Ende der Grammatikregel wiederum das Nonterminal <obj_name,NP,_> ergänzt, um auf die Eigennamenliste zu referieren.

Allerdings gibt es auch Muster aus dem Lernansatz, die sich nur unvollständig in eine Grammatikregel umformen lassen. Dazu zählt beispielsweise der folgende reguläre Ausdruck:

<what> <be> the (cause|reason) (for|of) <T>

Hier gibt es zu viele Möglichkeiten, was das Ziel der Frage, also <T>, sein könnte. Daher lässt sich <T> nur interaktiv bestimmen, indem weitere Klärungsfragen an den Benutzer gestellt werden.

Um dennoch zu gewährleisten, dass möglichst viele Fragen an das System gestellt werden können, wurden insgesamt ca. 2200 Fragen aus früheren TREC-Evaluationen in die Grammatik aufgenommen und diesen auch die entsprechenden semantischen Eigenschaften zugeordnet.

10.5 Evaluation

10.5.1 Einleitung

Zunächst wurde das QA-System EPHYRA allein evaluiert, um seine Potentiale bei einer Integration im Dialog besser einschätzen zu können. Im Anschluss daran wurde das Gesamtsystem, bestehend aus dem Dialogsystem in Kombination mit dem QA-System, evaluiert, um zu ermitteln, inwiefern diese Kombination eine Verbesserung der Gesamtperformanz bietet.

10.5.2 Evaluation des QA-Systems EPHYRA

Zunächst wurde das EPHYRA-System anhand der 200 Fragen aus der achten Text REtrieval Conference (TREC8) evaluiert. Für jede Frage gab EPHYRA fünf Antworten zurück, die dann manuell evaluiert und mit den Antwortmustern von TREC8 verglichen wurden. Eine Antwort wurde als korrekt angesehen, wenn sie zumindest so präzise war wie das entsprechende Antwortmuster.

Die Präzision ist die Rate an Fragen, bei denen zumindest eine der ersten fünf Antworten richtig war. Um den Rang der Antwort miteinbeziehen zu können, wurde der mittlere reziproke Rang (cf. engl. Mean Reciprocal Rank (MRR)) berechnet, der auch zur Evaluation in TREC8 benutzt wird. Dabei ist der MRR die Umkehrfunktion des Rangs der ersten richtigen Antwort bzw. 0, wenn alle Antworten falsch sind:

$$MRR = \frac{1}{n} \sum_n^{i-1} RR(question_i)$$

$$RR(question_i) = \begin{cases} 0, & \text{Rank} = 0 \\ \frac{1}{n}, & \text{Rank} \in 1, \dots, 5 \end{cases}$$

Wie in Abbildung 10.1 zu sehen, können 4 Fragen mit Mitteln der Wissensannotierung beantwortet werden. Davon wird leider eine Frage falsch beantwortet, da die gefundene Seite in Wikipedia nicht korrekt geparkt werden konnte. Ungefähr gleich viele Fragen werden von dem Musterlernverfahren wie von dem Backup-Verfahren behandelt. Präzision und MRR sind für das Musterlernverfahren besser. Hinzu kommt, dass die Antworten vom Musterlernverfahren exakter sind als die vom Backup-Ansatz, der z.T. nur Satzfragmente liefert, was das Verständnis für den Benutzer erschwert.

Etwa 50% der Fragen werden vom Musterlernansatz beantwortet. Für einige Fragen existiert keine angemessene Klasse oder das Fragemuster enthält die aktuelle Frage nicht. Andere Fragen lassen sich generell mit diesem Ansatz nicht behandeln, weil komplexe semantische Transformationen nötig wären, wie z.B. bei der Frage “When did Spain and Korea start ambassadorial relations?” (“Wann haben Spanien und Korea diplomatische Beziehungen aufgenommen?”). Hinzukommen Fehler im Bereich des Part-of-Speech-Taggings und grammatikalische Fehler in den Fragen, wie z.B. bei “How many people in Tucson?” (“Wie viele Menschen in Tucson?”).

Insgesamt wird somit etwa die Hälfte der Fragen richtig beantwortet. Innerhalb des Dialogs wäre es darüberhinaus auch möglich, dem Benutzer Informationen mitzugeben, wie sicher sich das System bei einer bestimmten Antwort ist. Ähnlich wie in zwischenmenschlichen Dialogen ist eine Antwort, bei der der Dialogpartner angibt, dass er unsicher ist, besser als keine Antwort, weil sie den anderen Dialogpartner wieder auf neue Ideen bringen kann, so dass dieser neue Fragen stellt.

10.5.3 Evaluation des Gesamtsystems

Danach wurde ein Test mit dem Gesamtsystem inklusive des im Dialog integrierten QA-Systems durchgeführt. Dazu wurden 40 Fragen aus dem Korpus von

	Wissens- annotierung	Muster- lernen	Backup	Insgesamt
# Fragen	4	96	99	200
# Richtige Antworten	3	55	47	105
Präzision	0,75	0,57	0,47	0,53
MRR	0,75	0,40	0,32	0,36

Tabelle 10.1: Auswertung der TREC8-Fragen für die benutzten QA-Techniken

	Grammatikbasiert	Statistisch
Wortfehlerrate	82,78%	59,71%
Rate an korrekt erkannten Bedeutungen	38,46%	41,03%

Tabelle 10.2: Vergleich des grammatikbasierten und des statistischen Spracherkenners: Wortfehlerraten und Rate an korrekt erkannten Bedeutungen, die für weitere Klärungsfragen im Dialog genutzt werden können

QA-TREC 2004 entnommen (der nicht zur Systementwicklung benutzt wurde) und diese an das System gestellt (Siehe Anhang V).

Es zeigt sich, dass mit dem grammatikbasierten Spracherkennner genau eine Frage vollständig korrekt erkannt wurde und mit dem statistischen Spracherkennner überhaupt keine. Dennoch konnte vielen Sätzen mit Hilfe der Grammatik die korrekte semantische Eigenschaft zugeordnet werden (cf. Tabelle 10.2), so dass es für den Benutzer damit möglich ist, mit Hilfe von Klärungsfragen des Dialogmanagers interaktiv die Frage näher zu spezifizieren und sie im Anschluss daran an EPHYRA zu übergeben. Hier zeigt sich auch, dass eine Kombination aus statistischem Spracherkennner mit einer Grammatik die meisten semantischen Eigenschaften korrekt ermittelt. Ferner ist beim statistischen Spracherkennner die Wortfehlerrate geringer, was sich dadurch erklären lässt, dass die Grammatik weniger Freiheiten am Ende eines Satzes zulässt, weil die Kosten für den Wechsel eines Pfades aufgrund der längeren Historie hier viel größer sind. Die Wortfehlerraten liegen insgesamt sehr hoch, was sich dadurch erklären lässt, dass der Spracherkennner nicht für die QA-Domäne trainiert wurde.

Die Probleme hinsichtlich der Performanz sind ähnlich denen, die bereits in dem im vorangegangenen Kapitel beschriebenen Mechanismus zur dynamischen Generierung von linguistischen Ressourcen auftraten: Das System wird durch die große Menge an neu hinzugekommenem Vokabular sehr viel langsamer, so dass das Starten des Systems nun einige Minuten in Anspruch nimmt und auch die Auswertung der Benutzeräußerungen verzögert sich merklich. Ein Grund dafür liegt sicherlich in den verwendeten Komponenten, die in dieser Form nicht für große Anwendungen vorgesehen sind: Der grammatikbasierte Spracherkennner JANUS (Finke et al. 1997) und das Dialogsystem TAPAS, das vor allem für die schnelle Prototypentwicklung vorgesehen ist (Holzapfel 2005; Denecke 2002b).

Als besondere Schwierigkeit darüberhinaus ergibt sich, dass die in QA-

Systemen sehr häufigen Eigennamen zu Problemen bei der Erkennung führen. Mit Hilfe der generierten Lexika lässt sich dies nur teilweise beheben aufgrund der bereits erwähnten Perplexität. Gerade hier spielt daher die Interaktion zwischen Benutzer und System eine große Rolle, da sich der Benutzer damit an seine Frage "herantasten" kann.

10.6 Zusammenfassung

Die Integration eines Frage-Antwort-Systems bietet den Vorteil, dass der Benutzer auch allgemeine Wissensfragen stellen kann, die nun vom Roboter beantwortet werden. Problematisch ist dabei die Abdeckung des Sprachmodells: Ein statistisches Sprachmodell ist häufig zu generell und eignet sich nicht für die hier vorliegende Domäne, während ein grammatikbasiertes Sprachmodell immer nur einen Teil der möglichen Fragen abdecken kann. Daher bietet sich eine Kombination der beiden Ansätze an, die die Fragen an das System in semantische Klassen einteilt, um dem Dialogmanager damit die Möglichkeit zu geben, Nachfragen an den Benutzer zu stellen, um eine Frage näher zu erklären. Auch die Ergebnisse der durchgeführten Tests zeigen, dass eine Kombination aus statistischem Spracherkenner mit einer Grammatik, die die Benutzeräußerungen ihren jeweiligen semantischen Eigenschaften zuordnet, am erfolgsversprechendsten ist, da der statistische Spracherkenner mit ca. 40% eine höhere Erkennungsgenauigkeit hat als der grammatikbasierte Spracherkenner, der nur bei ca. 20% liegt.

Außerdem ließe sich das QA-System EPHYRA innerhalb des Dialogsystems auch intern integrieren, um weiteres Hintergrundwissen zu liefern. So zeigten beispielsweise Benutzertests (Porzel, Gurevych 2002), dass Menschen je nach Kontext unterschiedliche Antworten geben auf die Frage, wo sich ein bestimmtes Kino befindet: Morgens gaben sie eine Antwort, die die Adresse enthielt, während abends eher eine Wegbeschreibung vom aktuellen Standpunkt aus gegeben wurde. Dabei spielte auch das Wetter eine Rolle oder die Tatsache, ob der Gesprächspartner Gepäck bei sich trägt, etc. D.h., implizit werden Annahmen von einem menschlichen Gesprächspartner gemacht, wofür sich der Dialogpartner interessiert, wobei der Kontext miteinbezogen wird. Ein solcher Kontext ließe sich von EPHYRA ermitteln, was beispielsweise das Wetter angeht; andere Aspekte könnten innerhalb der Roboterdomäne auch von den Kameras auf dem Roboter erfasst werden.

Kapitel 11

Auflösung elliptischer und anaphorischer Äußerungen

11.1 Einleitung

In diesem Kapitel geht es um die Integration von Kontextwissen zur Auflösung von elliptischen und anaphorischen Äußerungen. Wie in dem zweiten Teil dieser Arbeit anhand verschiedener Benutzertests gezeigt wurde, lassen sich eine ganze Reihe von Fehlern in der Mensch-Roboter-Kommunikation auf die fehlende Auflösung von Anaphern und Ellipsen zurückführen (siehe Tabelle 6.2 in Kapitel 6). In der direkten Mensch-Roboter-Interaktion finden sich vor allem deiktische Pronomina und Personalpronomina:

- Benutzer: Robbi, get the blue cup from the board.
(Robbi, hol den blauen Becher von der Anrichte.)
- Roboter: Going to take the blue cup from the board.
(Ich hole den blauen Becher von der Anrichte.)
- Benutzer: Bring it to me.
(Bring ihn mir.)
- Roboter: Going to bring you the blue cup.
(Ich bringe dir den blauen Becher.)
- Benutzer: Switch on that light.
+ Zeigegeste zur großen Lampe
(Mach das Licht an.)
- Roboter: Switching on the big lamp.
(Ich mache das große Licht an.)

Die Auflösung findet in beiden Fällen während der Erstellung der semantischen Repräsentation statt. Als Eingabe wird eine geparste Benutzeräußerung transformiert in eine semantische Repräsentation erwartet; die Ausgabe ist ebenfalls eine semantische Repräsentation, die nun angereichert ist mit Informationen aus der Referenzauflösung.

Zunächst geht es um die Integration von deiktischen Pronomina und Zeigegesten, wie sie auch bereits in den multimodalen Benutzertests aus Kapitel 4 integriert ist. Der zweite Abschnitt dieses Kapitels beschäftigt sich mit der Auflösung von Personalpronomina als einer weiteren Form von Anaphern. Schließlich

werden Kontextmanagement und die Auflösung unvollständiger oder elliptischer Benutzereingaben beleuchtet.

11.2 Integration von deiktischen Pronomina und Zeigegesten

11.2.1 Einleitung

Es liegt ein großes Potential in der Verknüpfung von Sprache mit Gestik zur Verbesserung der Mensch-Roboter-Interaktion, da dies der natürlichen zwischenmenschlichen Kommunikation viel näher kommt. Darüberhinaus können mit Hilfe multimodaler Interaktion auch nicht vorhandene oder falsch erkannte Informationen in einem Modus durch das Hinzufügen von Informationen anderer Modi kompensiert werden. Im Folgenden sollen daher zunächst die Vorteile von multimodalen Systemen und die Gründe, warum multimodale Interaktion effizienter für den Benutzer ist als unimodale, erläutert werden, bevor erklärt wird, wie die Informationen der verschiedenen Modalitäten geparkt und sinnvoll in das Gesamtsystem integriert werden können. Dazu wird auch der an unserem Institut entwickelte Gestenerkennung näher erläutert (Nickel, Stiefelhagen 2003).

11.2.2 Vorteile multimodaler Interaktion

Ein geradezu klassisches Beispiel für multimodale Interaktion mit Sprache und Gesten stellt das vielzitierte “Put that there” von Bolt dar (Bolt 1980). Seit damals gab es eine ganze Reihe von Forschungsarbeiten im Bereich der Spracherkennung und des Dialogmanagements (Weitere Details sind auch zu finden in Johnston et al. 1997; Johnston 1998; Cohen et al. 1997; Oviatt 1999a) so dass man nun beispielsweise in der Lage ist, kontinuierliche Sprache zu integrieren und damit eine natürlichere Kommunikation zu ermöglichen. Obwohl die Technologie also damals sehr viel schlechter war, war die Vision sehr ähnlich: Eine integrierte multimodale Architektur zu bauen, die den menschlichen Bedürfnissen entspricht.

Beide Modalitäten können sich ergänzen, so dass sich Ambiguitäten durch Fusion auflösen lassen, wie auch bereits in verschiedenen Forschungsarbeiten evaluiert wurde (cf. Oviatt 1999a; Nakagawa, Zhang 1994, etc.). Empirische Ergebnisse zeigen, dass Benutzer besser mit multimodalen Systemen arbeiten können als mit unimodalen (Oviatt 1999a; Ando et al. 1994; Cohen et al. 1997). Beide Modi lassen sich zur gegenseitigen Disambiguierung verwenden, so dass beispielsweise Gesten den reinen Sprachinput ergänzen und zur Auflösung von Anaphern beitragen, die mit rein linguistischen Informationen nur schwer aufzulösen sind. Dafür eignen sich zusätzliche Zeigegesten, die bestimmen, über welches Objekt gerade geredet wird, sehr gut.

Ein anderer Vorteil der multimodalen Interaktion (cf. Landragin 2002) liegt in der Tatsache begründet, dass in einigen Fällen die verbale Beschreibung eines Objekts wesentlich länger bzw. komplizierter ist als die entsprechende Geste. In diesen Fällen bevorzugen Benutzer eine Geste gegenüber einer gesprochenen-sprachlichen Äußerung. Auf der anderen Seite gibt es auch Fälle, in denen deik-

tische Gesten nicht benutzt werden, weil das fragliche Objekt zu klein, zu weit vom Benutzer entfernt ist, zu einer Gruppe von Objekten gehört, o.ä. Hier werden auch die Prinzipien der Gestalttheorie in Betracht gezogen, die bestimmen, ob jemand z.B. auf ein einzelnes Objekt zeigt oder eine Gruppe von Objekten (zur Gestalttheorie cf. Wertheimer 1923). Besonders gut für multimodale Systeme eignen sich ferner räumliche Aufgaben (Oviatt 1999b). Obwohl es in dem hier betrachteten Fall keine wirklich räumlichen Aufgaben gibt, sind doch Situationen zu finden, in denen verbale Beschreibungen wesentlich komplizierter sind als die entsprechenden Gesten, was dazu führt, dass der Benutzer Gesten bevorzugt.

In der zwischenmenschlichen Kommunikation ist Sprache die vorherrschende Modalität, die von Gesten unterstützt wird (Corradini et al. 2002). Das heißt, dass die sprachliche Äußerung die Interpretation der Gesten beeinflusst; beispielsweise weisen demonstrative Pronomen auch auf das mögliche Vorkommen von Gesten hin. Darüberhinaus gibt es verschiedene Studien, die belegen, dass Sprache und Gesten gemeinsam etwas ausdrücken (cf. beispielsweise Quek et al. 2002; McNeill, Duncan 2000), was bedeutet, dass sie die gleiche semantische Idee verkörpern, obwohl verschiedene Modalitäten benutzt werden. Daher erfolgt die Fusion der beiden Modi auf semantischer Ebene, und nicht, wie z.B. in Bolt (1980), auf der Ebene der Modalitäten.

In der Literatur findet man verschiedene Klassifikationen von Gesten (cf. Quek et al. 2002; McNeill, Duncan 2000, etc.). Die meisten klassifizieren die Gesten nach ihrer Funktion von reinem Gestikulieren über semaphorische Gesten bis hin zu wirklicher Zeichensprache. Hier liegt das Hauptaugenmerk demgegenüber auf deiktischen Gesten als einem Ansatzpunkt für multimodale Kommunikation. Aber da diese Gesten nur einen kleinen Ausschnitt aus der ganzen Mensch-Computer-Kommunikation bilden (Oviatt 1999b), ist das System erweiterbar, um auch mit verschiedenen anderen Typen von Gesten und generell auch mit anderen Informationen, die die Kameraaugen des Roboters bieten können, umgehen zu können.

11.2.3 Der Gestenerkennner

Zeigegesten können mit Hilfe einer Stereokamera von dem hier verwendeten Gestenerkennner erkannt werden (Nickel, Stiefelhagen 2003). Dazu wird zunächst ein Benutzermodell erstellt, das sich zusammensetzt aus der Position des Kopfes und der der Hände des Benutzers. Zeigegesten können dann in Echtzeit mit Hilfe von Hidden-Markov-Modellen, die auf Beispieltrajektorien trainiert wurden, erkannt werden. Die Bestimmung der Zeigerichtung erfolgt über die Kopf-Hand-Linie.

Der Gestenerkennner schickt sowohl die Information, wo sich der Punkt der Hand des Benutzers befindet, als auch die Zeigerichtung zum Dialogmanager, der daraufhin in seiner Datenbank aller Objekte der Umgebung nachsieht, ob auf eins dieser Objekte gezeigt wurde. Wenn kein Objekt gefunden werden kann, auf das der Benutzer gezeigt haben könnte, wird die Gesteneingabe ignoriert, und es wird davon ausgegangen, dass es sich um eine unbeabsichtigte Geste des Benutzers handelte. Dafür wird der Fehler zu der idealen Gerade von der Hand des Benutzers zum zu zeigenden Objekt berechnet und wenn er größer

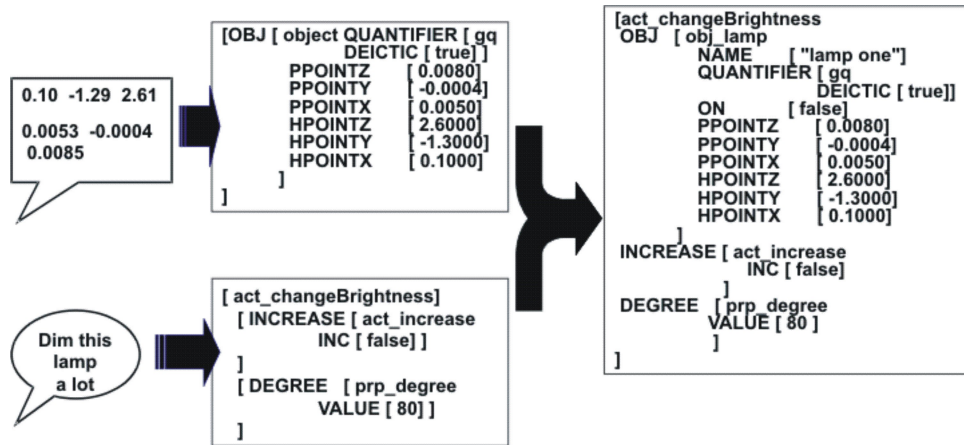


Abbildung 11.1: Beispiel für die multimodale Integration von Sprache und Gesten: “Dimm this lamp a lot.” (“Dimme diese Lampe sehr stark.”) + Zeigegeste

als 45° ist, kann man davon ausgehen, dass der Benutzer nicht auf das Objekt zeigen wollte. Liegt der Fehler demgegenüber unterhalb von 45° , so wird die semantische Repräsentation dieses Objekts in den Diskurs des Dialogmanagers übernommen und kann dann mit der Information vom Spracherkenner unifiziert werden.

11.2.4 Multimodale Integration

Es gibt verschiedene Möglichkeiten, wie Sprache und Gesten kombiniert werden können und wie sie sich dann in multimodaler Interaktion gegenseitig beeinflussen:

- Nur Sprache: Der Benutzer sagt etwas, ohne Gesten zu benutzen.
- Nur Gesten: Der Benutzer zeigt auf ein bestimmtes Objekt. Seine begleitende Äußerung enthält jedoch keine zusätzliche semantische Information. Das ist vor allem dann der Fall, wenn der Benutzer auf eine Klärungsfrage vom Dialogmanager antwortet. Beispielsweise hat der Roboter verstanden, dass der Benutzer ein Gerät anschalten möchte, aber nicht welches, und fragt daher nach, “What do you want me to switch on?” (“Was soll ich anschalten?”). Der Benutzer zeigt auf ein Objekt und sagt nur “This one.” (“Das da.”). In diesem Fall trägt die Geste alle Informationen und der Dialogmanager kann die sprachliche Eingabe sogar gänzlich ignorieren.
- Sprache und Gesten: Der Benutzer sagt etwas und zeigt gleichzeitig auf ein bestimmtes Objekt. In diesem Fall müssen beide Eingabeströme fusioniert werden, wie auch in der Abbildung 11.1 deutlich wird: Der Benutzer sagt “Dim this lamp a lot.” (“Dimme diese Lampe sehr stark.”) und zeigt gleichzeitig auf eine bestimmte Lampe. In diesem Fall kommt die Information über die auszuführende Aktion vom Spracherkenner, wohingegen

Gesamtzahl Turns	1151
Turns mit deiktischen Pronomina	125
Rate deiktischer Pronomina und Zeigegesten (in %)	10,86

Tabelle 11.1: Anzahl an Turns mit deiktischen Pronomina und Zeigegesten in einem Experiment mit dem Haushaltsroboter

das Objekt, das manipuliert werden soll, von der Geste näher bestimmt wird.

Die ersten beiden Möglichkeiten sind eigentlich unimodal, da hier die zusätzliche Information von der anderen Modalität ignoriert werden kann. Die eigentliche multimodale Interaktion findet sich in dem dritten Punkt, in dem beide Informationen auf semantischer Ebene fusioniert werden müssen unabhängig von ihrer Modalität.

Um diese deiktischen Gesten auflösen zu können, muss gleichzeitig eine Zeigegeste und ein deiktisches Pronomen vorliegen. Der Gestenerkennung und das multimodale Parsing werden benutzt und beide Eingabemodalitäten werden auf semantischer Ebene miteinander verbunden mit Hilfe von Zeitstempeln (Giesemann, Denecke 2003). Dafür werden die hereinkommenden Gesten mit Hilfe des Kontextmodells, das aus den verschiedenen Küchenobjekten besteht, wie z.B. Becher, Teller, Gabeln, Messer, Löffel und Lampen, ausgewertet. Eine nächsten Liste mit allen Zeigegesten, die zu einem bestimmten Zielobjekt aus dem Kontextmodell passen könnten, wird erstellt. Die Disambiguierung findet dann durch das Zusammenführen von Sprache und Gesten statt. Im Detail heißt das, dass beide Erkennung, sowohl der Spracherkennung als auch der Gestenerkennung, ihre Ausgaben zum Dialogmanager schicken. Die Ausgabe vom Spracherkennung wird analysiert mit Hilfe von kontextfreien Grammatiken und dem Domänenmodell, eine semantische Repräsentation wird aufgebaut und in den Diskurs gestellt (siehe auch Kapitel 2.2.2). Gleichzeitig wird auch die Information vom Gestenerkennung vom Dialogmanager verarbeitet und es wird überprüft, ob der Benutzer auf ein Objekt im Raum gezeigt haben könnte. Sollte das der Fall sein, so wird die semantische Repräsentation dieses Objekts auch in den Diskurs gestellt und die Informationen werden unifiziert. Dafür werden auch die Zeitstempel beider Eingaben verglichen. Darüberhinaus gibt es spezielle Dialogziele, die evaluieren, ob die Informationen der beiden Erkennung kompatibel sind. Auf diese Weise kann die Semantik beider Modi abgeglichen werden. Deiktische Pronomina ohne eine zugehörige Zeigegeste können momentan nicht aufgelöst werden.

Um die Häufigkeit von deiktischen Pronomina mit Zeigegesten zu ermitteln, wurde eine Benutzerstudie mit dem hier verwendeten Haushaltsroboter durchgeführt, in der die Benutzer mit dem Roboter mit Hilfe von Gesten und Sprache kommunizieren konnten. In ca. 10% der Sätze fanden sich Zeigegesten mit deiktischen Pronomina (siehe auch Tabelle 11.1).

11.3 Auflösung von Personalpronomina

11.3.1 Einleitung

Obwohl Dialogsysteme als auch Mechanismen zur Referenzauflösung sehr bekannte Forschungsgebiete sind, gibt es bisher wenig Untersuchungen dazu, wie diese beiden gut miteinander zu verknüpfen sind. Daher werden im Folgenden zunächst Arbeiten zu beiden Teilgebieten vorgestellt, bevor es darum geht, wie sie miteinander verbunden und im Dialogmanager des Roboters eingebaut werden können.

11.3.2 Verwandte Arbeiten

Mechanismen zur Referenzauflösung

Da es sehr viele Forscher gibt, die sich mit Referenzauflösung aus den unterschiedlichsten Blickwinkeln, wie beispielsweise Philosophie, Psychologie, Linguistik, Informatik, usw., beschäftigen, wird hier nur der Teil erwähnt, der für die vorliegende Arbeit von direktem Interesse ist.

Einer der ältesten Algorithmen zur Pronominaauflösung ist sicherlich Hobb's naiver Algorithmus (Hobbs 1977). Er durchläuft die Oberflächenstruktur der Parsebäume der Sätze in einem Text und sucht nach Nominalphrasen, die den richtigen Numerus und das richtige Genus haben als Antezedens für vorhandene Pronomina. Obwohl dieser Algorithmus recht einfach ist, funktioniert er sehr gut, so dass 90% der Pronomina damit aufgelöst werden können (cf. (Hobbs 1977)).

Die Theorie von Diskursstrukturen und Zentrenbildung, die von Grosz et al. erdacht und weiterentwickelt wurde (Grosz, Sidner 1986; Brennan et al. 1987; Grosz et al. 1995; Walker 1998), dient dazu, den Diskurskontext zu verfolgen und Pronomina zu binden. Zunächst wird eine Menge von Verbindungen festgelegt, die etwas ähnliches spezifizieren. Diese Menge wird mit Hilfe einiger Regeln gefiltert, klassifiziert und schließlich geordnet. Die Regeln basieren auf der Verbindung von Antezedens und Pronomen, wie beispielsweise parallele grammatische Funktionen, wann das Antezedens vorgekommen ist, usw. Darüberhinaus wird es bevorzugt, mit der gleichen Entität fortzufahren und diese nicht zu wechseln. Obwohl der Algorithmus wesentlich komplexer ist als der von Hobbs, sind die Ergebnisse recht ähnlich (cf. Tetreault, Allen 2003).

Als eine Erweiterung des Zentrenmodells benutzt Strube eine Liste von hervortretenden Diskursentitäten, die sogenannte S-list (Strube 1998). Diese Liste basiert auf dem Informationsstatus. Die Unterscheidung zwischen neuer und alter Information im Diskurs spielt eine entscheidende Rolle. Ferner enthält dieses Modell auch Präferenzen für inter- bzw. intrasententielle Anaphora, die im ursprünglichen Zentrenmodell noch nicht vorkamen.

Baldwin entwickelte CogNIAC, eine Maschinerie zum Auflösen von Pronomina (Baldwin 1995), die eine Menge von Regeln benutzt, um das korrekte Antezedens in einer Liste zu finden. Diese Regeln sind recht einfach gehalten, wie z.B. "Wenn es nur ein mögliches Antezedens im vorhergehenden Satz gibt, so benutze dieses"; es wird kein Weltwissen zur Auflösung der Pronomina ver-

wendet. Dennoch scheinen die Regeln sehr effizient zu sein, da er angibt, 92% korrekt auflösen zu können (cf. Baldwin 1995).

Alle bisher genannten Mechanismen wurden mit Hilfe von geschriebenen Texten entwickelt. Natürlich können sie auch bis zu einem gewissen Grad auf die gesprochene Kommunikation übertragen werden, müssen aber an deren spezielle Bedürfnisse angepasst werden, wie beispielsweise spontane Effekte in der gesprochenen Sprache. Daher beschäftigen sich die folgenden Abschnitte mit Referenzauflösungsmechanismen, die in gesprochensprachlichen Dialogen benutzt werden.

Referenzauflösung in Dialogsystemen

Bisher gibt es nur relativ wenige Dialogsysteme, die Referenzauflösungsmechanismen benutzen, da die meisten von ihnen speziell auf die Telefonkommunikation zugeschnitten sind, wie z.B. Flug- und Zugauskunftssysteme (McTear 2002; Allen et al. 1995; Stallard 2000), Telefonweiterleitungssysteme (Gorin et al. 2002), Wetterinformationssysteme (Zue et al. 2000) etc. und daher keine Referenzauflösung benötigen. Da es aber inzwischen immer mehr Systeme für die direkte Mensch-Maschine-Kommunikation, wie z.B. intelligente Räume oder auch Roboter, gibt, müssen auch solche Faktoren wie die situierte, kontextabhängige Kommunikation, die sich verändernde Umwelt, die multimodale Interaktion, usw. in Betracht gezogen werden.

So benutzen beispielsweise Kumar et al. einen Ansatz, der auf kognitiven Grammatiken basiert, die eine konzeptuelle Semantik voraussetzen (Kumar et al. 2003). Referenzdomänen identifizieren Repräsentationen für Teilmengen von kontextuellen Entitäten, auf die referiert werden kann, wie z.B. individuelle Objekte oder auch Ansammlungen von mehreren Objekten. Das interessante Merkmal an einer solchen Referenzdomäne ist der Teil, der zusammen mit Fokus und Wichtigkeit die Kriterien für die Auflösung definiert. Unterspezifizierte Referenzdomänen werden mit der existierenden Kontextstruktur mit Hilfe von Gruppierung und Assimilation erstellt. So können Referenzen aufgelöst werden, indem man den korrespondierenden Knoten in der Kontextstruktur sucht. Der gleiche Mechanismus wird sowohl für linguistische Ausdrücke als auch für andere Arten von Referenzen, wie Deixis und Pronomina, verwendet.

Andere Forscher (Landragin, Romary 2003) schlagen eine Klassifikation von referierenden Modi vor, die die Aktionen und Disambiguierungsprinzipien beschreiben, um das korrekte Antezedens zu finden. Referenzen können aufgelöst werden mit Hilfe von Unifizierung mit der im Kontext vorhandenen Information, so dass das Antezedens mit dem besten Unifizierungsergebnis genommen wird. Auf diese Weise werden ebenfalls deiktische Pronomina und Zeigegesten aufgelöst.

Für das Galaxy System wurde sogar ein ganzer Kontextresolutionsserver entwickelt (Filisko, Seneff 2003), der Mechanismen zum Reparieren, Anaphora- und Ellipsenauflösung, History-Funktionen, usw. enthält. Pronomina werden aufgelöst mit Hilfe einer Liste von Diskursentitäten, die nach einem möglichen Antezedens durchsucht wird.

```
[ act_put OBJ
  [ obj_puttable
    [ generic:NAME [ "it" ] ]
  ]
  [ DESTINATION
    [ DEST [ "table" ] ]
  ]
]
```

Abbildung 11.2: Semantische Repräsentation des Satzes “Put it on the table.” (“Leg es auf den Tisch.”)

11.3.3 Kontextmodellierung zur Referenzauflösung

Einleitung

Aus diesen Ansätzen wurde ein Modell zur Referenzauflösung entwickelt, das ähnliche Methoden benutzt und auch eine solche Liste möglicher Antezedens und Regeln für die Übereinstimmung von Antezedens und Pronomen verwendet. Das Modell ist dabei speziell auf die Mensch-Roboter-Kommunikation zugeschnitten, indem es beispielsweise auch Wissen über die Situation des Roboters ausnutzt.

Neben der Datenbank, die als Kontextmodell dient, spielt auch die Ontologie eine sehr wichtige Rolle in der Referenzauflösung, da sie benötigt wird, um die semantische Übereinstimmung zwischen dem Pronomen und dem Antezedens zu ermitteln: Wenn beide zur gleichen Kategorie, bzw. einer Unterkategorie, in der Ontologie gehören, kann man von einer semantischen Übereinstimmung sprechen. Deutlich wird dies an der in Abbildung 11.2 zu sehenden semantischen Repräsentation des Satzes “Put it on the table.” (“Leg es auf den Tisch.”): “it” (“es”) bezieht sich auf ein Objekt, das man nehmen kann, da das Verb “put” (“nimm”) in der Dialogzieldefinition ein “obj_puttable” verlangt. D.h., andere Antezedens sind von vorneherein semantisch ausgeschlossen. Wenn der Benutzer z.B. im vorhergehenden Satz gesagt hat, “Get the cup from the board.” (“Hol den Becher von der Anrichte.”), kann die Anrichte kein Antezedens für “it” (“es”) sein, da es zu einer anderen Kategorie in der Ontologie gehört. Auf diese Weise ist sichergestellt, dass nur semantisch sinnvolle Antezedens von dem Algorithmus berücksichtigt werden.

Das Kontextmodell

Das Kontextmodell ist als Datenbank realisiert. Dort werden neben den Informationen über die Umwelt, die für die Auflösung der deiktischen Gesten benötigt werden, auch mögliche Antezedens gespeichert in einer ähnlichen Weise wie in Strubes S-List (Strube 1998). Da nur nominale Antezedens für Pronomina in den bisherigen Benutzertests gefunden wurden, werden hier zunächst nur diese Pronomina betrachtet. Außerdem sind einige expletive Pronomina bereits von der Grammatik abgedeckt durch Ausdrücke wie beispielsweise “It is too dark in here.” (“Es ist zu dunkel hier.”). Andere können momentan nicht aufgelöst werden.

Gesamtzahl Turns	572
Turns mit Personalpronomina	37
Rate Personalpronomina (in %)	6,47

Tabelle 11.2: Anzahl an Turns mit Personalpronomina in einem Experiment mit dem Haushaltsroboter

Das Kontextmodell ist so implementiert, dass es ähnlich dem menschlichen Gehirn funktioniert und alte Antezedens wieder “vergisst” nach einer bestimmten Zeitspanne (Clark 1978). Kommt eine neue Benutzeräußerung, so wird das Kontextmodell mit den dort enthaltenen möglichen Antezedens aktualisiert.

Benutzerstudie zur Auflösung von Personalpronomina

In einer Benutzerstudie, in der die Benutzer den Roboter den Tisch decken lassen sollten, fanden sich in ungefähr 6% der Äußerungen Personalpronomina (siehe Tabelle 11.2). Mit Hilfe des Kontextmodells können sie auf zwei verschiedene Arten aufgelöst werden:

- aus dem Dialogkontext heraus, indem nur die Informationen der vorhergehenden Sätze betrachtet werden.
- aus der Situation heraus. Das bedeutet, dass eine Art simples Weltwissen benutzt wird. Beispielsweise kann man, wenn der Roboter einen Becher in seinen Greifern hält, und der Benutzer zu ihm sagt “Put *it* there.” (“Stell *es* dahin.”), davon ausgehen, dass sich das “*it*” (“*es*”) auf den Becher bezieht.

Einerseits wird damit die Information aus dem Kontextmodell, was sich in den Händen des Roboters befindet, zur Auflösung der Pronomina herangezogen. Andererseits wird in der Liste der möglichen Antezedens nach einem passenden gesucht, so dass Pronomen und Antezedens in ihren semantischen und syntaktischen Merkmalen übereinstimmen. D.h., dass sie den gleichen Numerus und das gleiche Genus haben, soweit es die Syntax angeht, und dass beide zur gleichen Kategorie oder einer Unterkategorie in der Ontologie gehören, um in bezug auf Semantik übereinzustimmen. Da die Antezedens nach ihrem Erscheinen sortiert sind und gelöscht werden, wenn sie zu alt sind, kann das erste mögliche Antezedens, das von diesem Algorithmus gefunden wird, genommen und seine semantische Repräsentation in den Diskurs eingefügt werden. Auch eine Kombination beider Algorithmen klingt vielversprechend. Es gibt nämlich eine ganze Reihe von Situationen, wo der Algorithmus, der auf den vorangegangenen Sätzen beruht, versagt, weil das vorhergehende Antezedens zu weit entfernt ist und darum nicht gefunden werden kann. Ebenso kann auch der andere Algorithmus versagen, wenn der Roboter etwas anderes in den Händen hat, als das, worauf der Benutzer referenziert.

Beide Algorithmen funktionieren trotz ihrer Einfachheit sehr gut im gegebenen Roboterszenario, so dass ungefähr 90% der Pronomina aufgelöst werden können. In der hier durchgeführten Benutzerstudie können sogar alle Pronomina

```

    move Deselected_Oprop {
variable:(Intention = deselected),
path: ($sem.[generic:OPROP] is defined),
->
bindings:
    internal://dialogue/say    #Which1 {$sem.[generic:OPROP]},
    ‘‘object do you mean?’’;
};

```

Abbildung 11.3: Beispiel eines zusätzlichen generischen Moves, der dafür sorgt, dass auch einzelne Worte im Diskurs bereits eine Nachfrage des Dialogmanagers triggern

nur aus der Situation heraus aufgelöst werden, so dass der komplexere Mechanismus mit dem möglichen Antezedens im Kontextmodell nicht benötigt wird. Dies kann allerdings auch daran liegen, dass das Tischdeck-Szenario recht einfach ist. Die Kombination beider Algorithmen wird in der Gesamtevaluation in Kapitel 14 berücksichtigt.

11.4 Auflösung elliptischer Äußerungen

Um elliptische Äußerungen besser auflösen zu können, ist es nötig, die Parsbarkeit dieser Äußerungen zu verbessern und anschließend deren weitere Behandlung im Diskurs zu optimieren. Dafür wurde zunächst die existierende Grammatik so umgeschrieben, dass Nominalphrasen in ihre Bestandteile zerlegt werden, damit auch Benutzeräußerungen, die aus nur einem Wort, wie z.B. “blue”, bestehen, geparkt werden können, da sich in den Benutzertests zeigte, dass Benutzer dazu tendieren, möglichst kurze Phrasen auszuprobieren, und es sogar mit einzelnen Worten versuchen (cf. Kapitel 6).

Außerdem gibt es jetzt zusätzliche Moves, die dafür sorgen, dass, auch wenn nur ein Wort im Diskurs ist, der Benutzer eine Rückfrage vom System bekommt. Dafür wurden generische Moves implementiert, wie am Beispiel in Abbildung 11.3 zu sehen. Hier gibt es einen Move, der dafür sorgt, dass der Benutzer, wenn er eine Eigenschaft, wie z.B. “blau” nennt, eine Rückfrage bekommt, welches blaue Objekt er meint. Wenn der Benutzer dies dann näher spezifiziert, können die Informationen im Diskurs zusammengefasst werden.

Das Problem dieser sehr kurzen Benutzeräußerungen ist, dass sie meist keinen Sprechakt enthalten und daher kein Dialogziel ausgewählt werden kann. Dies führt dazu, dass der Dialogzustand nach der Äußerung nicht selektiert ist und daher die Information im Diskurs wieder gelöscht wird (für eine detaillierte Beschreibung der Dialogzustände und ihrer Auswirkungen siehe Kapitel 2.2.2). Daher wurde ein zusätzlicher Mechanismus eingebaut, der in den Fällen, in denen der Dialogzustand nicht selektiert ist, die im Diskurs vorhandene Information zu Objekten oder Eigenschaften in das Kontextmodell wegschreibt. Diese Information bleibt dort erhalten und kann später mit den vom Benutzer

hinzugefügten Informationen zusammengefügt werden. Gelangt der Benutzer erneut in den nicht selektierten Dialogzustand, so wird die neue Information am Anfang hinzugefügt und bereits vorhandene rückt weiter nach hinten (analog zu der Liste möglicher Antezedens, die sich ebenfalls im Kontextmodell befindet).

Ferner erweist es sich als problematisch, dass diese extrem kurzen Äußerungen für den Spracherkenner sehr schwer zu erkennen sind, weil nur wenige Informationen vorliegen. Darauf geht das Kapitel 13 näher ein, wenn es um die Detektion von Fehlern geht.

11.5 Zusammenfassung

In diesem Kapitel wurden Algorithmen zur Integration von Kontextwissen zur Auflösung von Anaphern und elliptischen Äußerungen entwickelt. Dabei ging es vor allem um die Auflösung von deiktischen Pronomina und Personalpronomina. Ferner wurden generelle Dialogmechanismen implementiert, um sicherzustellen, dass elliptische Äußerungen korrekt behandelt werden können. Die Auswirkungen dieser Mechanismen werden zusammen mit den in den vorangegangenen und den folgenden Kapiteln beschriebenen Fehlerbehandlungsmechanismen in einer Abschlussevaluation in Kapitel 14 evaluiert und diskutiert.

Kapitel 12

Auflösung komplexer Äußerungen

12.1 Einleitung

In der in der vorliegenden Arbeit durchgeführten Fehlerklassifikation (siehe Kapitel 4 bis 6) gehören die Fehler, die durch zu komplexe Benutzeräußerungen entstanden sind, zu der kleinsten Fehlerkategorie. Dennoch sollen sie hier genauer analysiert werden, da der Benutzer momentan nicht nachvollziehen kann, was das Problem an diesen Äußerungen ist, weshalb er keine Antwort bekommt. Da Benutzeräußerungen immer nur bis zu einem gewissen Komplexitätsgrad in der Grammatik abgedeckt sein können, liegt es in der Natur der Sache, dass der Dialogmanager nicht alles verstehen kann. Daher werden im Folgenden nur solche komplexen Äußerungen, wie sie auch in den Benutzertests zu finden waren, analysiert.

Außerdem spielt hier das Problem der Sprachsegmentierung mit hinein. Der Sprachsegmentierer unterteilt die hereinkommenden akustischen Signale in Benutzeräußerungen und Stille aufgrund der vorhandenen Intensität des Signals und leitet die Benutzeräußerungen weiter an den Spracherkenner (Giesermann et al. 2003). Dabei kann es passieren, dass eine Benutzeräußerung fälschlicherweise in zwei Turns aufgeteilt wird, weil der Benutzer eine Pause mitten in der Äußerung gemacht hat. Andererseits kann es aber auch vorkommen, dass zwei Benutzeräußerungen fälschlicherweise zu einem Turn zusammengefasst werden, wenn die Pause zwischen den beiden Äußerungen zu kurz für den Sprachsegmentierer war. Da es an dieser Stelle zu weit führen würde, im Detail auf die Sprachsegmentierung einzugehen und diese zu verbessern, werden im Rahmen dieser Arbeit die Benutzeräußerungen, wie sie vom Sprachsegmentierer kommen, als gegeben angenommen und es werden Methoden implementiert, um mit zu langen Benutzeräußerungen umgehen zu können.

Verknüpfungen von	Rate
Nominalphrasen	62,75%
Verbalphrasen	37,25%

Tabelle 12.1: Raten der Verknüpfungen von Nominal- vs. Verbalphrasen in den Benutzertests in den Kapiteln 4 und 5

12.2 Analyse der in den Benutzertests vorkommenden verknüpften Äußerungen

In den durchgeführten Benutzertests (siehe Kapitel 4 und 5) finden sich zwei verschiedene Arten von zusammengesetzten Äußerungen, die momentan zu komplex sind, um vom System verstanden zu werden:

1. Verknüpfung von zwei oder mehr Nominalphrasen, wie beispielsweise “Bring me apples and tomatoes.” (“Bring mir Äpfel und Tomaten.”)
2. Verknüpfung von zwei oder mehr Verbalphrasen, wie z.B. “Get me coffee and put milk in it.” (“Hol mir einen Kaffee und tu Milch rein.”) Diese können sogar komplett unabhängig voneinander sein wie in dem folgenden Beispiel: “Hi Robbi. I would like some coke.” (“Hallo Robbi. Ich hätte gern etwas Cola.”)

Wie in Tabelle 12.1 zu sehen, findet sich der erste Fall sehr viel häufiger in den Daten. Besonders im internetbasierten Test gehen viele Benutzer davon aus, dass der Roboter zwei verschiedene Objekte auf einmal tragen kann. In den Tests mit dem Roboter ist das weit seltener der Fall, weil die Benutzer den Roboter und seine Fähigkeiten sehen und ihn daher wohl besser einschätzen können.

Der zweite Fall der Verknüpfung von Verbalphrasen tritt vor allem dann auf, wenn die Sprachsegmentierung zwei Benutzeräußerungen nicht voneinander trennt, sondern in einem Turn zusammengefasst hat. Darüberhinaus enthalten einige Benutzerturns auch nicht nur ein Dialogziel, sondern mehrere, da dies in der ineinander verwobenen zwischenmenschlichen Kommunikation sehr häufig vorkommt, aber momentane Dialogsysteme leider noch überfordert.

12.3 Auflösung verknüpfter Nominalphrasen

Im Fall der Verknüpfung mehrerer Nominalphrasen wurde eine neue generelle Klasse von Sprechakten definiert, die dafür sorgt, dass eine beliebige Anzahl von Objekten geparkt werden kann (siehe Abbildung 12.1). Dafür wird das erste Objekt in ein einzelnes Objekt geschrieben und die folgenden Objekte in eine Liste von Objekten, so dass man damit auf die einzelnen Teile der Liste zugreifen kann. Von dieser Klasse erben alle die Verben, bei denen mehr als ein Objekt möglich ist, wie beispielsweise Hol- und Bringdienste.

Dadurch dass es sich bei dem zweiten Objekt um eine Liste von Objekten handelt, ist es nicht nur möglich, Nominalphrasen mit zwei Objekten zu parsen,


```

class act_twoObjects inherits action {
    object : OBJ;
    objectlist : OBJ2;
};
class objectlist inherits object {
object : OBJECT1;
object : OBJECT2;
object : OBJECT3;
object : OBJECT4;
object : OBJECT5;
};

```

Abbildung 12.1: Definition eines generischen Sprechakts mit mehreren Objekten und der dazugehörigen Objektliste

sondern auch solche mit weit mehr Objekten. Aus praktischen Gründen bietet es sich jedoch an von nicht mehr als fünf verschiedenen Objekten auszugehen, da es auch in der zwischenmenschlichen Kommunikation nicht üblich ist, so lange Verknüpfungen zu verwenden, weil dies die Kapazität des Kurzzeitgedächtnisses des Zuhörers bald übersteigen würde. In den Benutzertests enthalten die Listen sogar höchstens drei Objekte.

Fast alle Benutzeräußerungen mit verknüpften Nominalphrasen der Benutzertests lassen sich damit auflösen. Übrig bleiben lediglich spontane Äußerungen, die nicht konsistent sind mit der aktuellen Ontologie und Grammatik und auch in einer zwischenmenschlichen Kommunikation nur im übertragenden Sinne gebraucht werden, wie z.B. “I want you to cook spaghetti for me and a coke.” (“Ich möchte, dass Du Spaghetti für mich kochst und eine Cola.”). Der Sprechakt “etwas für jemanden kochen” kann zwar durch die oben beschriebene Vererbung von dem generellen Sprechakt mehrere Objekte enthalten, aber diese Objekte müssen alle gekocht werden können, was auf Cola nicht zutrifft. Da solche metaphorischen Sprechakte immer ein Problem für ein Dialogsystem darstellen, weil in der Ontologie und der Grammatik nur die wörtliche Bedeutung und keine übertragenden Bedeutungen abgebildet werden, ist dies ein allgemeines Problem, das hier nur am Rande erwähnt werden soll.

12.4 Auflösung verknüpfter Verbalphrasen

Der Fall der Verknüpfung mehrerer Verbalphrasen ist komplizierter, weil der Benutzer zwei verschiedene Aufgaben für den Roboter in einem Turn verknüpft, so dass zwei Sprechakte in einem Turn vorkommen. Bisher wird jedoch von nur einem Sprechakt pro Benutzerturn und pro Dialogziel ausgegangen.

Es gibt nun die Möglichkeit, in der Applikation einige Sprechakte miteinander zu verknüpfen. Dies hat den Nachteil, dass es applikationsspezifisch bleibt und sehr aufwendig ist, wenn alle möglichen Kombinationen an Sprechakten beachtet werden sollen. Demgegenüber besteht auch die Möglichkeit, ein gene-

risches Dialogziel zu definieren, das zwei Sprechakte enthält:

```
goal DoubleAct {
  precondition:
    [ double_act
      ACT [ generic:action ]
      ACT2 [ generic:action ]
    ]
  ->
  bindings:
    internal://dialogue/say  "I am sorry,
    I can only do one thing at a time.";
  };
}
```

Dieses generische Dialogziel enthält damit alle Kombinationen an verschiedenen Verbalphrasen, die in der Applikation vorkommen.

Will man dem Benutzer keine generische Antwort geben, so kann man mit Hilfe der Generierungsschablonen (cf. Kapitel 2.2.2) wiederum applikationsspezifisch darauf reagieren. Da es jedoch relativ aufwendig ist, wenn man für jede mögliche Kombination an Sprechakten eine eigene Antwort definiert, bietet sich eine generische Antwort, wie in dem oben angegebenen Beispiel, an.

Da in allen Benutzertests nur Verknüpfungen von nicht mehr als zwei Verbalphrasen vorkamen, sollen hier auch keine längeren Verknüpfungsketten behandelt werden, die theoretisch jedoch denkbar wären und sich durch längere Verknüpfungen mehrerer Sprechakte realisieren ließen. Hinzu kommt, dass auch bereits mit einer Verknüpfung von zwei Verbalphrasen eine riesige Menge an neuen komplexen Sprechakten entsteht, da der Benutzer theoretisch alle möglichen Sprechakte in ihren verschiedenen Formulierungsvariationen miteinander verknüpfen kann, während es bei dem Fall der Verknüpfung von Nominalphrasen "nur" um eine Verknüpfung aller möglichen Objekte geht.

12.5 Zusammenfassung

Um komplexe Äußerungen parsen zu können, wurden zunächst die komplexen Äußerungen, die in den Benutzertests aufgetreten sind, genauer analysiert. Dabei stellte sich heraus, dass es sich in 62,75% aller Fälle um Verknüpfungen von Nominalphrasen handelt. Dafür wurde ein generischer Sprechakt definiert, der eine Liste mit bekannten Objekten abdecken kann. Von diesem Sprechakt erben die in der Roboterapplikation vorkommenden Sprechakte, so dass damit alle Äußerungen mit mehreren Objekten geparkt werden können. Als Grenze für die Größe der Liste wurden fünf Objekte gewählt, da mehr Objekte die begrenzte Kapazität des menschlichen Kurzzeitgedächtnisses überlasten würden und damit auch in zwischenmenschlicher Kommunikation unüblich sind. In den Benutzertests kamen sogar nur Listen mit höchstens drei Objekten vor.

Seltener sind Verknüpfungen von Verbalphrasen. Hierfür wurde ein generisches Dialogziel entworfen, das zwei Sprechakte enthält. In gleicher Weise lassen

sich auch Verknüpfungen von mehr als zwei Verbalphrasen realisieren. Hier wurde jedoch darauf verzichtet, da sie in den Benutzertests nicht aufgetreten sind.

Kapitel 13

Klärungsdialoge und Metakommunikation

13.1 Einleitung

Dieses Kapitel beschäftigt sich mit der Behandlung von Fehlern auf der Metakommunikationsebene. Die meisten dieser Fehler ergeben sich erst aus dem Versuch der Auflösung von Fehlern. Der Benutzer hat beispielsweise bemerkt, dass der Roboter ihn missverstanden hat und versucht dieses Missverständnis jetzt auf der Metakommunikationsebene zu klären. Solche Dialoge sind für den Dialogmanager eine besondere Herausforderung, weil es sich um einen Dialog über den Dialog handelt, der damit eine Ebene höher angesiedelt ist. Auf der anderen Seite bietet die Metakommunikation aber auch dem Roboter die Gelegenheit, selbst Klärungsdialoge anzustoßen, wenn er bemerkt, dass der Benutzer Probleme hat. Dazu sind Mechanismen notwendig, die Probleme im Dialog erkennen und die Generierung von darauf zugeschnittenen Antworten des Roboters zulassen.

Zunächst geht es darum, wie Menschen mit diesem Problem umgehen. Dabei soll ein Dialog zwischen zwei Gesprächspartner, die unterschiedlich kompetent in einer gegebenen Sprache sind, als Beispiel dienen. Danach wurden die in den Benutzertests aufgetretenen Metakommunikationsäußerungen genauer analysiert und klassifiziert. Anhand dieser Klassen werden dann generische Mechanismen zur Auflösung von Metakommunikation implementiert und getestet. Schließlich werden die Mechanismen erläutert, die es dem Roboter erlauben, Probleme im Dialog zu erkennen und daraufhin dann metakommunikative Maßnahmen zu ihrer Lösung zu ergreifen.

13.2 Reparaturdialoge in der zwischenmenschlichen Kommunikation

13.2.1 Einleitung

An dieser Stelle geht es um die zwischenmenschliche Kommunikation und die Klärung der Frage, inwieweit sich in ihr Ansatzpunkte für eine verbesserte

Fehlerbehandlung in der Mensch-Roboter-Kommunikation finden. Daher beschäftigt sich dieser Abschnitt zunächst mit der zwischenmenschlichen Kommunikation mit ungleichen Gesprächspartnern, von denen einer über weniger Grammatik- bzw. Vokabularkenntnisse verfügt, wie z.B. ein Ausländer, da wir auch im Mensch-Roboter-Dialog ungleiche Gesprächspartner haben. Diese Strategien aus der zwischenmenschlichen Kommunikation werden im Anschluss auf die Mensch-Roboter-Kommunikation übertragen. Dabei werden die Indikatoren für Fehler und die Fehlerbehandlungsstrategien in beiden Kommunikationsformen verglichen, da der Roboter die gleichen Strategien zur Fehlerbehandlung verwenden sollte, wie sie in der zwischenmenschlichen Kommunikation Anwendung finden (cf. Kapitel 2). Fehlerdialoge werden mit Mitteln der Kommunikationsanalyse analysiert, um herauszufinden, inwieweit sie sich von zwischenmenschlichen Dialogen unterscheiden und inwiefern sich der Mensch-Roboter-Dialog noch stärker an einen zwischenmenschlichen Dialog anpassen ließe.

13.2.2 Analyse der Fehlerdialoge

Es wurde eine Benutzerstudie mit vier Benutzern durchgeführt, und die Mensch-Roboter-Dialoge wurden anschließend in einer Black-Box-Evaluation analysiert, um sicherzustellen, dass die Dialoge objektiv von außen betrachtet werden und nicht aus einer Entwickler-Perspektive. Dafür wurden alle Äußerungen des Roboters als auch die des Benutzers ausgewertet, ohne darauf zu achten, ob die Äußerung vom Spracherkenner richtig erkannt wurde, ob sie korrekt geparkt werden konnte, oder in die richtige Semantik umgewandelt wurde. Zur Analyse der Dialoge wurden ethnomethodologische Methoden benutzt, wie die bereits im Kapitel 2 erwähnten von Schegloff (Schegloff et al. 1977).

Die Benutzer konnten frei mit dem Roboter interagieren. Sie erhielten keine Aufgaben, sondern wussten nur, dass sie gerade einen neuen Haushaltsroboter bekommen haben, der ihnen in der Küche helfen soll. Auf diese Weise erhält man eine größere Zahl an zu analysierenden Fehler als in einem aufgabenbasierten Benutzertest, weil die Benutzer auch Worte bzw. Konzepte verwendeten, die der Roboter nicht kannte.

Ergebnisse

Insgesamt gab es 212 Turns, 106 vom Benutzer und 106 vom Roboter. Von den 106 Roboterturns waren allerdings 26 leer; d.h., es erfolgte keine Reaktion vom Roboter. Diese Turns wurden ebenfalls als Fehler klassifiziert, weil der Roboter dem Benutzer die Antwort schuldig blieb.

Damit fanden sich insgesamt 71 Turns mit Fehlern in diesem Benutzertest. Wie erwartet zeigen die Ergebnisse, dass es sich meist um fremd-initiierte Selbstkorrektur (cf. Kapitel 2) handelte (94,36% aller Fehler wurden auf diese Weise korrigiert.). D.h., der Roboter hat dem Benutzer deutlich gemacht, dass er etwas nicht verstanden hat und der Benutzer hat daraufhin seine Äußerung versucht umzuformulieren. In den restlichen 5,74% der Fehlerkorrekturen hat der Benutzer unerwartet das Thema des Gesprächs gewechselt, was aber in allen Fällen darauf zurückzuführen war, dass der Roboter zuvor ebenfalls einen unerwarteten

Indikatoren für Fehler	Rate
Signalisieren von Nichtverstehen	24,0%
Erfragen von Zusatzinformationen	8,0%
Wiederholung eines zentralen Elements	5,33%
Ignorieren des Fehlers oder Abbrechen	62,67%

Tabelle 13.1: Indikatoren für Fehler in der Mensch-Roboter-Kommunikation

Themenwechsel vorgenommen hatte, den der Benutzer auf diese Weise ignoriert hat.

Indikatoren für Fehler

Da es sich bei den meisten Korrekturen um fremd-initiierte Selbstkorrektur handelt, finden sich die Indikatoren für Fehler vor allem in den Äußerungen des Roboters. Als Indikatoren für Fehler können einerseits prosodische Merkmale dienen, wie sie durch Hyperartikulation hervorgerufen werden und auch bereits von einigen Forschern untersucht wurden (Swerts et al. 2000; Soltau, Waibel 2000; Oviatt 1998). Auf der anderen Seite gibt es aber auch Indikatoren für Fehler in den erkannten Äußerungen selbst, die im Fokus des Interesses der vorliegenden Arbeit liegen.

Als Indikatoren für eine Fehlersituation fanden sich in den meisten Fällen (62,67% aller Fehler) Äußerungen, die den Fehler ignorierten, wie auch in Tabelle 13.1 zu sehen. Dabei handelte es sich bei der Hälfte der Äußerungen um solche, in denen der Roboter nicht auf die vorhergehende Benutzeräußerung reagierte, sondern sie ignoriert hat und dem Benutzer keine Antwort gab. In etwa 24% der Fehler zeigte der Roboter sein Nichtverstehen durch unspezifische Fragen, wie “How can I help you?” (“Wie kann ich dir helfen?”). In 8% aller Fehlersituationen fragte der Roboter nach Zusatzinformationen, wie z.B. “Where should I go ?” (“Wohin soll ich gehen?”), “Which cup do you want me to take?” (“Welchen Becher soll ich holen?”). In einigen wenigen Fällen (5,33%) hat der Roboter nur die vorhergehende Frage wiederholt.

In einer zwischenmenschlichen Kommunikation mit Ausländern lässt sich eine ausgeglichene Verteilung der Fehlerindikatoren mit etwa 22% in allen vier Kategorien finden, wie Marti in seiner Arbeit zeigt (Marti 2001). Um zu ermitteln, inwieweit dies die Mensch-Roboter-Kommunikation negativ beeinflusst, werden nun die eigentlichen Strategien zur Fehlerbehebung näher untersucht und es wird evaluiert, inwieweit sich diese ebenfalls von den in der zwischenmenschlichen Kommunikation verwendeten unterscheiden bzw. inwieweit sie trotzdem zu einer erfolgreichen Korrektur führen können.

Außerdem zeigen auch die Benutzerreaktionen auf die verschiedenen Indikatoren für Fehler, dass die Benutzer eine explizitere Indikation bevorzugen: Wenn der Roboter die vorhergehende Benutzeräußerung ignorierte, fühlten sich die Benutzer verloren und probierten verschiedene Strategien aus, um den Fehler aufzulösen, wie beispielsweise Paraphrasen, Abbrechen der Aufgabe und Übergang zu einer neuen Aufgabe, Nachfragen, was der Roboter kann, usw. Daher

Fehlerbehebungsstrategien	Rate
Problemlösungsstrategien	37,3%
Funktionale Reduktion	16,0%
Formale Reduktion	4,0%
Ratifizierung	2,7%

Tabelle 13.2: Fehlerbehebungsstrategien in der Mensch-Roboter-Kommunikation

fürte diese Art der Fehlerindikation im Benutzertest nie zur unmittelbaren Auflösung des Fehlers. Auch die Anzeige von Nichtverstehen mit Hilfe unspezifischer Fragen führte zu Verwirrung bei den Benutzern, weil sie nicht wussten, was der Roboter im Einzelnen nicht verstanden hatte. Daher konnten nur in 16% aller Fehlersituationen Fehler nach einer solchen Indikation aufgelöst werden, so dass der Roboter schließlich verstand, was der Benutzer wollte.

Auf der anderen Seite ließen sich Fehler in all den Fällen erfolgreich auflösen, in denen der Roboter dem Benutzer eine spezifische Frage nach weiterer Information stellte, und der Benutzer diese Frage beantwortete. Hier erwies es sich lediglich als problematisch, dass der Benutzer, wenn er zu diesem Zeitpunkt im Dialog keine Frage erwartete, diese auch nicht beantwortete, wodurch der Fehler ebenfalls nicht aufgelöst werden konnte. In all den Fällen, in denen der Roboter seine vorhergehende Frage wiederholte, versuchten die Benutzer mit Umformulierungen ihrer letzten Antwort die Situation zu klären. Dies gelang in der Hälfte aller Fälle. Bei der anderen Hälfte handelte es sich um Konzepte, die der Roboter nicht kannte und daher auch nicht verstand. Zusammenfassend lässt sich sagen, dass diese Resultate die Bedeutung der Klärungsfragen vom Roboter an den Benutzer unterstreichen. Bei der Auflösung von Fehlern spielen sie eine entscheidende Rolle und werden vom Benutzer dringend benötigt.

Strategien zur Fehlerbehebung

Momentan wird die Fehlerbehebung größtenteils vom Menschen erledigt angesichts der Tatsache, dass es sich zumeist um fremd-initiierte Selbstreparatur handelt. Nicht alle Fehler werden korrigiert, einige werden auch ignoriert. Zur Fehlerbehebung werden in dem Benutzertest größtenteils Problemlösungsstrategien benutzt (37,3%), wie auch Tabelle 13.2 zu entnehmen ist. Paraphrasen und Umformulierungen sind mit 14,67% aller Benutzeräußerungen sehr häufig. In 5,33% aller Benutzeräußerungen hat der Benutzer seine letzte Äußerung wörtlich nochmal wiederholt in der Hoffnung, dass der Roboter ihn beim zweiten Mal versteht.

Außerdem traten in 8% aller Benutzeräußerungen komplexe Korrekturäußerungen auf, die in den ursprünglichen Modellen der zwischenmenschlichen Kommunikation nicht abgedeckt waren (siehe Kapitel 2). Solche Äußerungen wurden verwendet, wenn der Roboter etwas gesagt hat, was der Benutzerintention widersprach. Der Benutzer versuchte dann die Äußerung des Roboters zu korrigieren, damit dieser versteht, was er wirklich gesagt hat. Dies wurde

in der zwischenmenschlichen Kommunikation vermutlich deshalb nicht benutzt, weil der Mensch davon ausging, dass der Ausländer solche komplexen Änderungen nicht verstehen würde. Interessanterweise scheinen die Benutzer in der Mensch-Roboter-Interaktion aber zu glauben, dass der Roboter dies versteht.

Der folgende Beispieldialog verdeutlicht die einzelnen Fehlerbehebungsstrategien, wie sie im Mensch-Roboter-Dialog angewandt wurden:

1. Benutzer: Robbi, do we have any orange juice?
(Robbi, haben wir Orangensaft?)
2. Roboter: I will have a look at the fridge. There are tomatoes, apples, noodles, eggs, milk, butter, cucumbers, coke and juice.
(Ich werde im Kühlschrank nachsehen. Da sind Tomaten, Äpfel, Nudeln, Eier, Milch, Butter, Gurken, Cola und Saft.)
3. Benutzer: What about orange juice.
Do we have any orange juice?
(Wie sieht's mit Orangensaft aus?
Haben wir Orangensaft?)
4. Roboter: With this stuff in your fridge you can make different dishes, such as spaghetti napoli, spaghetti pesto, a pancake and a salad. For which recipe are you looking?
(Mit den Sachen im Kühlschrank kannst du verschiedene Gerichte kochen, wie z.B. Spaghetti Napoli, Spaghetti Pesto, einen Pfannkuchen oder auch einen Salat. Für welches Rezept interessierst du dich?)
5. Benutzer: Robbi, is it orange juice or apple juice?
(Robbi, ist das Orangensaft oder Apfelsaft?)
6. Roboter: I will have a look at the fridge. There are tomatoes, apples, noodles, eggs, milk, butter, cucumbers, coke and juice.
(Ich werde im Kühlschrank nachsehen. Da sind Tomaten, Äpfel, Nudeln, Eier, Milch, Butter, Gurken, Cola und Saft.)
7. Benutzer: Please bring me this cup (+ Zeigegeste)
(Bitte bring mir diesen Becher.)
8. Roboter: Which cup do you want me to take?
(Welchen Becher soll ich nehmen?)
9. Benutzer: This cup. (+ Zeigegeste)
(Diesen Becher.)
10. Roboter: Where should I put it?
(Wo soll ich ihn hinstellen?)
11. Benutzer: Please bring me this cup.
(Bitte bring mir diesen Becher.)
12. Benutzer: Go. (Geh.)
13. Roboter: How much should I go forward?
(Wie weit soll ich gehen?)

14. Benutzer: Now bring me the blue cup.
(Nun hol mir den blauen Becher.)
15. Roboter: Where should I go?
(Wohin soll ich gehen?)
16. Benutzer: Bring me the blue cup.
(Hol mir den blauen Becher.)
17. Roboter: Where should I go?
(Wohin soll ich gehen?)
18. Benutzer: Bring me the yellow cup.
(Hol mir den gelben Becher.)
19. Roboter: Going to bring you the yellow cup.
(Ich bringe dir den gelben Becher.)
20. Benutzer: Bring me some food.
(Hol mir etwas zu essen.)
21. Benutzer: I asked you to bring me some food.
(Ich habe dich gebeten, mir etwas zu essen zu bringen.)
22. Roboter: How can I help you?
(Wie kann ich dir helfen?)
23. Benutzer: Some food please.
(Etwas zu essen, bitte.)
24. Roboter: Where do you want me to put it?
(Wohin soll ich es tun?)
25. Benutzer: I would like to eat a salad.
(Ich würde gern einen Salat essen.)
26. Roboter: For a salad you need tomatoes, cucumbers, and dressing.
(Für einen Salat brauchst du Tomaten,
Gurken und Dressing.)
27. Benutzer: Thanks. (Danke.)
28. Roboter: You are welcome. (Bitte sehr.)

Funktionale Reduktionen bestanden zumeist darin, das Thema komplett zu wechseln. In den Daten fand sich nur ein Fall mit einer funktionalen Reduktion der Intention (siehe Zeile 11 und 12 im Beispieldialog): Der Benutzer bat den Roboter, ihm einen Becher zu bringen, und als der Roboter ihn nicht verstand, änderte der Benutzer seine Intention und versuchte nunmehr den Roboter zu der Stelle gehen zu lassen, an der sich der Becher befand.

Es gab nur sehr wenige Beispiele einer formalen Reduktion. Meist waren die Umformulierungen der Benutzer sogar morphosyntaktisch komplizierter, wie auch in dem Beispieldialog in den Zeilen 20 und 21 zu sehen. Für Ratifizierungen fanden sich noch weniger Beispiele, was daran liegen könnte, dass der gesamte Dialog aufgabenorientiert war und der Benutzer sah, wenn der Roboter schließlich die Aufgabe, um die er ihn gebeten hatte, ausgeführt hat. Daher war keine weitere Bestätigung nötig.

Auch in zwischenmenschlichen Dialogen werden meist Problemlösungsstrategien verwendet, wie aus den Analysen von Mensch-Ausländer-Dialogen von

Marti hervorgeht (Marti 2001). Daher scheint dies auch eine in der Mensch-Roboter-Kommunikation angemessene Strategie zur Fehlerkorrektur zu sein, die die Benutzer bereits aus zwischenmenschlichen Dialogen kennen.

Detaillierte Analyse des Beispieldialogs

Der oben angegebene Beispieldialog beginnt mit einigen Problemlösungsstrategien auf Seiten des Benutzers (Zeilen 1-6). Er möchte Orangensaft, aber der Roboter teilt ihm lediglich mit, dass es Saft im Kühlschrank gibt. Daher versucht der Benutzer die nicht verstandene Information zu isolieren und fragt explizit nach, ob es sich um Apfel- oder Orangensaft handelt (siehe Zeile 5). Diese Frage scheint der Roboter aber auch nicht zu verstehen, so dass der Benutzer schließlich das Thema wechselt und den Roboter bittet, ihm einen Becher zu bringen. Da der Roboter die Benutzerintention nur teilweise verstanden hat, erklärt der Benutzer, welchen Becher er haben möchte mit Hilfe einer zusätzlichen Zeigegeste. Der Roboter versteht ihn immer noch nicht, so dass der Benutzer seine Intention ändert und nun den Roboter zunächst zu der Stelle gehen lassen will, wo der Becher ist (Formale Reduktion).

In den Zeilen 16 - 19 findet sich ein gutes Beispiel für eine funktionale Reduktion, wo der Benutzer zunächst den blauen Becher möchte, sich dann aber für den gelben entscheidet, nachdem der Roboter ihn nach mehreren Versuchen nicht verstanden hat. Wie bereits im vorangegangenen Abschnitt erläutert, finden sich nur sehr wenige morphosyntaktische Reduktionen. Stattdessen werden die Benutzeräußerungen sogar komplizierter, wenn sie umformuliert werden. Im Beispieldialog in den Zeilen 19 - 26 versteht der Roboter die Benutzerintention, ihm etwas zu essen zu bringen, mehrfach nicht. Der Benutzer versucht es erst mit komplexeren Paraphrasen (Zeile 23) und dann mit einer morphosyntaktischen Reduktion (Zeile 23). Schließlich entschließt er sich zu einer konkreteren Intention und sagt, dass er einen Salat essen möchte. Ein Beispiel für eine Ratifizierung findet sich in den letzten beiden Zeilen: Der Roboter nennt dem Benutzer die Zutaten zu einem Salat und der Benutzer bedankt sich bei ihm und bestätigt damit, dass seine Intention korrekt verstanden wurde.

Generell fällt auf, dass viele Fehler nicht repariert werden, sondern der Benutzer den Roboter dann eine andere Aufgabe ausführen lässt (siehe z.B. Zeile 7). Das lässt sich z.T. darauf zurückführen, dass die Benutzer keine konkreten Aufgaben hatten, sondern selbst entscheiden konnten, was der Roboter für sie machen sollte.

13.2.3 Zusammenfassung

Vergleicht man die zwischenmenschliche Kommunikation und die aktuelle Mensch-Roboter-Kommunikation, wie sie momentan in dem hier vorgestellten Basissystem möglich ist, so fällt auf, dass in der zwischenmenschlichen Kommunikation explizitere Strategien zur Fehlerindikation benutzt werden. Dennoch sind die Strategien zur Fehlerkorrektur und ihre Anwendung in der Kommunikation sehr ähnlich. Im Folgenden geht es um die Frage, inwieweit und in welcher Weise auch in der Mensch-Roboter-Kommunikation auf der Metakom-

munikationsebene Fehler gelöst werden können und welche Mechanismen implementiert wurden, um dies zu ermöglichen.

13.3 Vom Benutzer initiierte Metakommunikation

13.3.1 Einleitung

Alle Äußerungen in den durchgeführten Benutzertests und ebenso in der zur Erweiterung der Grammatik durchgeführten Datensammlungen (siehe Kapitel 4, 5 und 8), die die Metakommunikationsebene betreffen, wurden in verschiedene Klassen eingeteilt anhand der ihnen zugrunde liegenden Benutzerintention:

- *Nachfragen* vom Benutzer: Der Benutzer erkundigt sich z.B., ob der Roboter ihn richtig verstanden hat, was der Roboter gerade macht, ob er ein bestimmtes Objekt oder eine bestimmte Aufgabe kennt, was er gerade gesagt hat, etc.
- *Korrektur* von Benutzeräußerungen: Der Benutzer korrigiert die letzte Äußerung des Roboters. Dazu zählen auch Äußerungen, in denen der Benutzer versucht, den Diskurs des Dialogmanagers zu löschen, wie beispielsweise “Doesn’t matter.” (“Macht nichts.”), “Forget it.” (“Vergiss es.”)
- *Austesten* des Roboters: Der Benutzer testet die Fähigkeit des Roboters aus, indem er beispielsweise die letzte Äußerung des Roboters wiederholt oder Anweisungen an den Roboter gibt, von denen er vermutet, dass dieser sie nicht ausführen kann, wie z.B. “Bring me a dead body please.” (“Bring mir eine Leiche.”), “Can you get rid of my ant problem?” (“Kannst du mein ant-Problem lösen?”), “Rewire the house.” (“Verkabele das Haus neu.”), “Smash a plate.” (“Wirf den Teller auf den Boden.”). Zu dieser Kategorie zählen außerdem auch Beschimpfungen des Roboters. Daher geht die Antwort des Roboters hier nicht näher auf die zugrundeliegende Semantik der Benutzeräußerung ein, sondern vermittelt dem Benutzer lediglich, was der Roboter an Aufgaben erledigen kann.

Die Benutzerintention bei der ersten Klasse liegt darin herauszufinden, was der Roboter aktuell macht bzw. was er generell machen kann und ob er die letzte Anfrage vom Benutzer verstanden hat. Beim Austesten des Roboters dahingegen geht es darum, den Roboter und seine Fähigkeiten auf die Probe zu stellen. Beide Klassen können sich überschneiden: Wenn der Benutzer nachfragt, ob der Roboter eine bestimmte Aufgabe ausführen kann, so kann man dies entweder als eine Nachfrage klassifizieren oder als Austesten, je nach der vom Benutzer gewählten Formulierung. Daher wurde es in beiden Fällen so implementiert, dass der Benutzer eine generische Antwort vom Roboter bekommt, welche Fähigkeiten der Roboter hat.

Wie in Abbildung 13.1 zu sehen, sind die Benutzer in den webbasierten Tests mehr daran interessiert, den Roboter auszutesten. Hier beschimpfen sie den Roboter z.T. sogar, was sie in den Tests mit dem realen Roboter vermutlich deshalb nicht machen, weil ein menschlicher Versuchsleiter dabei ist. In den aufgabenbasierten Tests liegt generell der Anteil der Korrekturäußerungen höher als in

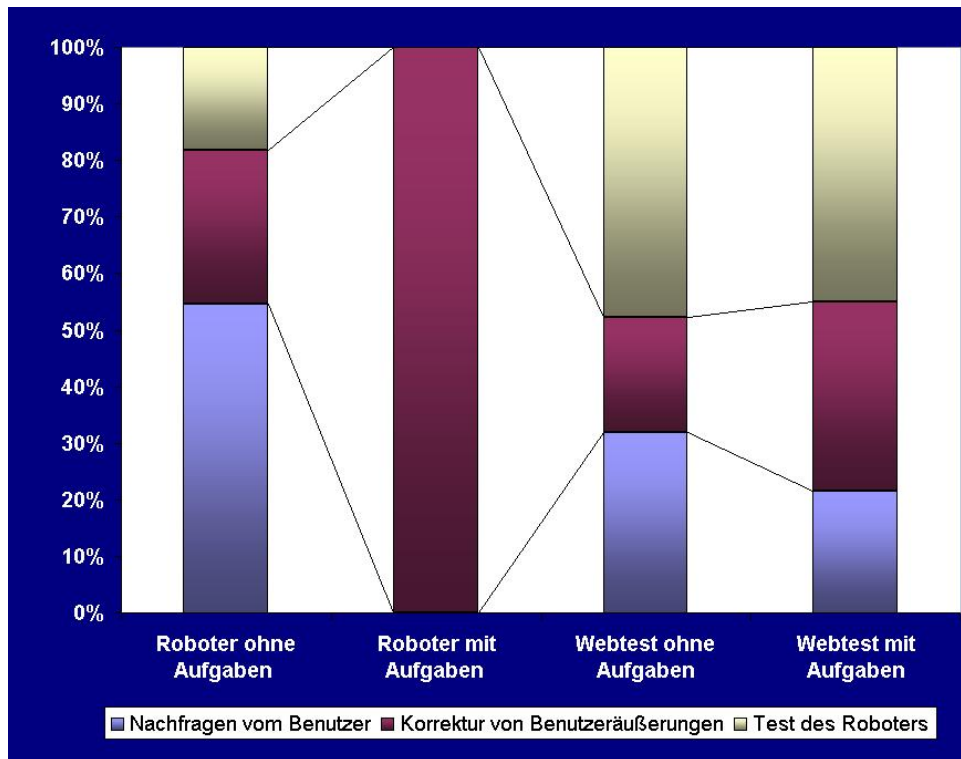


Abbildung 13.1: Klassifikation der Äußerungen auf Metakommunikationsebene und ihre Anteile in den Benutzertests

den Tests ohne Aufgaben, was sich dadurch erklären lässt, dass die Benutzer ohne konkrete Aufgaben häufig nicht versuchen, den Roboter zu korrigieren, sondern ihn diese falsch verstandene Aufgabe dann ausführen lassen, statt der ursprünglich von ihnen genannten. In den Tests ohne Aufgaben finden sich mehr Nachfragen von den Benutzern, da sie herauszufinden versuchen, was der Roboter tatsächlich kann. In den aufgabenbasierten Tests ist dies nicht nötig, da die Benutzer sich an den gegebenen Aufgaben festhalten können.

Mehr als die Hälfte aller Äußerungen sind, wie in Tabelle 13.4 zu sehen, nicht auf die spezielle Situation der Kommunikation mit einem Küchenroboter bezogen, sondern können auch in anderen Kontexten verwendet werden, wie beispielsweise “What do you do?” (“Was machst du?”), “Did you understand me?” (“Hast du mich verstanden?”). Darüberhinaus gibt es auch Formulierungen mit einem generischen Sprechakt und einem applikationsspezifischen Objekt, wie z.B. “Do you know X?” (“Kennst du X?”). Alle diese Äußerungen lassen sich daher generisch implementieren und können für andere Applikationen wiederverwendet werden.

13.3.2 Nachfragen vom Benutzer

Die Nachfragen vom Benutzer an den Roboter können verschiedene Hintergründe haben. Daher lassen sie sich unterteilen in die folgenden Unterkategorien:

	Anteil an Metakommunikation
Applikationsspezifische Äußerung	46,98%
Generische Äußerung	53,02%

Tabelle 13.4: Generische vs. applikationsspezifische metakommunikative Äußerungen über alle Benutzertests (Webbasiert und mit dem Roboter)

1. Sprechakt *act_clarify*:
 Hierbei handelt es sich um eine Nachfrage danach, was der Roboter gerade macht. Diese Frage sollte eigentlich von anderen Komponenten des Robotersystems, wie einer Kognition beispielsweise, durch Introspektion beantwortet werden, da der Dialogmanager nur weiß, was der Roboter machen soll, nicht aber, was er wirklich macht. Da dies über die Ziele der vorliegenden Arbeit weit hinausgehen würde, antwortet der Dialogmanager momentan generisch, dass er die vom Benutzer zuletzt gestellte Aufgabe ausgeführt hat und nun eine neue erwartet.
2. Sprechakt *act_clarifyDetail*:
 Mit dieser Nachfrage will der Benutzer ermitteln, ob der Roboter noch zuhört bzw. den Benutzer verstehen kann. Darauf antwortet der Roboter generisch, dass er bereit ist zum Zuhören und den Benutzer nach einer Aufgabe fragen.
3. Sprechakt *act_clarifyWord*:
 Hier wird nach einem bestimmten Wort gefragt: Der Roboter antwortet generisch, ob er das Wort kennt oder nicht. Um dies zu entscheiden, lassen sich Konfidenzen heranziehen. Kennt der Dialogmanager das Wort, d.h. liegt die Konfidenz hoch genug, so fragt er nach, was er damit machen soll. Kennt er das Wort nicht, d.h. liegt die Konfidenz unterhalb eines Schwellwertes, so fragt der Roboter den Benutzer nach einer Erklärung. Dies kann auch für spätere Lernmechanismen ausgebaut werden.
4. Sprechakt *act_clarifyAct*:
 Diese Nachfrage bezieht sich darauf, welche Tätigkeit der Roboter mit einem bestimmten Objekt gerade ausführt. Genau wie im ersten Fall bräuhete man hier eine Kognition, um diese Frage wirklich beantworten zu können. Daher handelt der Roboter hier proaktiv und fragt nach, was der Benutzer möchte, dass er mit dem Objekt macht.
5. Sprechakt *act_clarifyOutput*:
 Es wird nachgefragt, was der Roboter gerade gesagt hat. Dies passiert besonders oft in den Tests mit dem Roboter, wo die Ausgabe per Sprachsynthese erfolgte und daher für den Benutzer vor allem in Situationen, in denen er etwas ganz Anderes erwartet hat, schlecht verständlich war. Dies lässt sich ebenfalls generisch lösen, indem die letzte Äußerung des Roboters im Kontextmodell gespeichert wird, so dass der Roboter sie erneut wiedergeben kann.

6. Sprechakt *act_clarifyTask*:

Dies ist eine Nachfrage nach einer bestimmten Aufgabe. Der Benutzer erkundigt sich, ob der Roboter eine bestimmte Aufgabe ausführen kann. Kann der Roboter die Aufgabe ausführen, so wird diese Nachfrage als Aufforderung, sie auszuführen, verstanden und der Roboter erledigt die entsprechende Aufgabe sogleich. Kann der Roboter die entsprechende Aufgabe jedoch nicht ausführen, so erklärt er dem Benutzer, welche Aufgaben er stattdessen ausführen kann. Hier liegt auch die bereits beschriebene Überschneidung zu der Kategorie "Austesten des Roboters" vor.

7. Sprechakt *act_clarifyTaskDetails*:

Hierbei handelt es sich um eine Nachfrage nach applikationsspezifischen Details einer bestimmten Aufgabe. Der Benutzer fragt beispielsweise nach, wohin der Roboter gehen kann, welche Objekte er nehmen kann. Dies ist die einzige Klasse von Nachfragen, die sich nicht generisch modellieren lässt, da sie von den in einer spezifischen Applikation vorhandenen Aufgaben abhängt.

Alle diese Nachfragen wurden in der Ontologie und der Grammatik implementiert. Dabei wurde, soweit als möglich, eine generische Implementierung verwendet. Es wurden neue Sprechakte in der Ontologie eingeführt und entsprechende Grammatikregeln, die die Benutzeräußerungen parsen können, implementiert. Dadurch hat der Benutzer nun in für ihn unklaren Situationen die Möglichkeit, Nachfragen zu stellen, was sicherlich auch zu einer größeren Benutzerzufriedenheit beiträgt, wie in der Evaluation in Kapitel 14 noch untersucht wird.

13.3.3 Korrektur von Benutzeräußerungen

Korrekturen stellen für den Dialogmanager eine größere Herausforderung dar als die zuvor behandelten Nachfragen, da hier die bereits im Diskurs vorhandene Information geändert werden muss. Der Benutzer nimmt die Korrektur von Äußerungen entweder so vor, dass er die neue Information und die zu ersetzende in einer Äußerung nennt oder er versucht zunächst, die im Diskurs bereits vorhandene Information zu löschen. Dabei kann die zu löschende Information implizit oder explizit mitangegeben sein:

- *Überschreiben*: Der Benutzer überschreibt die im Diskurs vorhandene Information mit neuer Information. Dabei wird die zu entfernende Information im Diskurs nur implizit angegeben, dadurch dass sie mit der neuen Information inkompatibel ist. Mit Hilfe der von Denecke beschriebenen partiell unifizierten Merkmalsstrukturen (Denecke 2002a) ist es möglich, dass das System inferiert, welche augenscheinlich falsche Information aus dem Diskurs entfernt und durch welche neue ersetzt werden soll, wie das folgende Beispiel veranschaulicht:

Benutzer: Bring me a beer.
 (Bring mir ein Bier.)
 Roboter: Going to bring you some milk.
 (Ich bringe dir Milch.)
 Benutzer: No, I mean a beer.
 (Nein, ich meine ein Bier.)

- *Korrektur*: Der Benutzer korrigiert die im Diskurs vorhandene Information mit neuer Information. Dabei wird sowohl die zu entfernende Information als auch die hinzuzufügende Information explizit angegeben, wie das folgende Beispiel zeigt:

Benutzer: Bring me this cup. + Zeigegeste
 (Bring mir diesen Becher.)
 Roboter: Going to bring you some dressing.
 (Ich hole dir etwas Dressing.)
 Benutzer: No, I don't want you to bring me some
 dressing, I want you to bring me this cup.
 (Nein, ich möchte nicht, dass du mir
 Dressing bringst, sondern ich wollte, dass
 du mir diesen Becher holst.)

- *Löschen*: Der Benutzer löscht die im Diskurs vorhandene Information; es wird keine neue Information gegeben, wie auch an dem folgenden Beispiel zu sehen:

Benutzer: Bring me a salad.
 (Bring mir einen Salat.)
 Roboter: Going to bring you some apples.
 (Ich bringe dir Äpfel.)
 Benutzer: No, I don't want apples.
 (Nein, ich möchte keine Äpfel.)

Dabei kann die Korrektur von Benutzeräußerungen generisch erfolgen, wie beispielsweise "No, I meant X and not Y." ("Nein, ich meinte X und nicht Y."). Für den Fall wurden generische Regeln implementiert die dafür sorgen, dass Y durch X im Diskurs ersetzt wird (siehe Abbildung 13.2). Durch die Ausnutzung der Vererbung ist es möglich, dass die entsprechenden Objekte aus der Roboterapplikation kommen können.

Ferner kann sich die Korrektur von Benutzeräußerungen entweder auf ein komplettes Dialogziel beziehen oder nur auf ein Objekt. Daher können auch komplexe Sprechakte aus der eigentlichen Roboterapplikation eingebunden werden, wie beispielsweise: "No, I don't want coffee nor milk, I just want you to set the table, please." ("Nein, ich will weder Kaffee noch Milch, ich will nur, dass du den Tisch deckst."). Für den Fall sind ebenfalls generische Regeln implementiert, die die Ersetzung von Sprechakten ermöglichen. Somit kann der Benutzer auch komplette Sprechakte korrigieren.

Problematisch sind außerdem Korrekturen nach Finalisierung eines Dialogziels, da bei Finalisierung eines Dialogziels eine Botschaft mit der zu erledigenden Aufgabe an die anderen Komponenten des Dialogsystems geschickt wird, die


```

public <act_repair,VP,_>=
<act_overwrite,V,_> <object,NP,_> { RIGHT object }
<act_retract,V,_> <object,NP,_> { WRONG object }
public <act_retract,V,_>= [no] not
| i don't want
| i do not want
;
public <act_overwrite,V,_> = [no|but] i said
| [no|but] i meant
| [no|but] i want
| [no|but] i wanted
| [no|but] i mean
;

```

Abbildung 13.2: Generische Regel zur Korrektur von Benutzeräußerungen

nun widerrufen werden muss. Da dies aber nicht in den Bereich des eigentlichen Dialogsystems fällt, geht diese Arbeit nicht näher darauf ein.

13.3.4 Benutzeräußerungen zum Austesten des Roboters

Dies sind sicherlich die schwierigsten Äußerungen auf Metakommunikationsebene, weil der Benutzer absichtlich vom Roboter etwas möglichst Schwieriges verlangt, was dieser nicht verstehen soll. Daher wird an dieser Stelle nicht die Semantik der Äußerungen ermittelt, sondern der Dialogmanager handelt proaktiv, indem er dem Benutzer eine kurze Information darüber gibt, was er alles machen kann. Für Benutzer, die wirklich eine Aufgabe an den Roboter gestellt haben, die nicht ausgeführt werden konnte, führt dies zu einer befriedigenden Antwort, da sie nun wissen, bei welchen Aufgaben der Roboter sie generell unterstützen kann. Für Benutzer, die den Roboter austesten wollten, ist diese Information sicherlich auch nützlich, wenn es auch durchaus vorkommt, dass sie den Roboter weiter testen. Um diesen Benutzern möglichst viele Informationen über den Roboter geben zu können, antwortet der Dialogmanager jedes Mal mit verschiedenen Phrasen. Insgesamt wurden zehn Antwortalternativen implementiert, die von einer einfachen Aufzählung all der Aufgaben, die der Roboter erledigen kann, bis hin zu der direkten Nachfrage vom Roboter, ob der Benutzer ihn nur austesten will, reichen.

Da an dieser Stelle nicht versucht wird, die Semantik der Benutzeräußerung zu ermitteln, lassen sich diese Äußerungen ebenfalls generisch implementieren. Sofern es sich um Beschimpfungen oder seltsame Aufgaben handelt, werden sicherlich auch andere Applikationen außer dem Küchenroboter davon profitieren. Da die Antworten des Dialogsystems sich aber direkt auf die Aufgaben beziehen, die der Roboter wirklich ausführen kann, sind diese applikationsabhängig und müssen für andere Applikationen in ähnlicher Weise implementiert werden.

Sprechakt	Klasse	Reaktion
act_clarify	Nachfrage	Selektiertes Dialogziel wird genannt
act_clarifyDetail	Nachfrage	Nachfrage nach aktueller Aufgabe
act_clarifyAct	Nachfrage	Nachfrage nach aktuellem Aufgabe mit einem gegebenen Objekt
act_clarifyOutput	Nachfrage	Wiederholung der letzten Roboteräußerung
act_clarifyWord	Nachfrage	Nachfrage nach der Aufgabe mit diesem Wort
act_clarifyTask	Nachfrage	Aufgabe wird ausgeführt bzw. Aufgaben, die der Roboter ausführen kann, werden genannt
act_clarifyTaskDetails	Nachfrage	Applikationsspezifische Antwort
act_retract	Korrektur	Entfernen der alten Diskursinformation
act_overwrite	Korrektur	Überschreiben der alten Diskursinformation
act_repair	Korrektur	Überschreiben der alten Diskursinformation mit neuer
act_test	Test	Aufgaben, die der Roboter ausführen kann, werden genannt

Tabelle 13.5: Aufstellung aller neuen Sprechakte zur Metakommunikation

13.3.5 Diskussion und Evaluation der vom Benutzer angestoßenen Metakommunikation

Wie in Tabelle 13.5 zu sehen, wurden insgesamt elf neue Sprechakte definiert, die die Metakommunikation abdecken. Sie dienen zur Nachfrage, zur Korrektur und zum Test des Roboters. Nachdem die beschriebenen generischen Sprechakte zur Grammatik hinzugefügt wurden, wurden für alle applikationsspezifischen Metakommunikationsakte, die in der hier durchgeführten Datensammlung (cf. Kapitel 8) aufgetreten sind, ebenfalls Grammatikregeln geschrieben und auf dieser erweiterten Grammatik erneut die Parsingrate und die Turnfehlerrate ermittelt (siehe Tabelle 13.6). Man sieht eine leichte Verbesserung in der Parsingrate und eine weit geringere Anzahl an falsch verstandenen Turns in der Version mit Metakommunikation. Bei der Bedingung “Roboter ohne Aufgaben” bleibt die Parsingrate in etwa gleich, was sich durch die sehr einfachen Aufgaben erklären lässt, während auch hier die Turnfehlerrate zurückgeht.

13.4 Vom Roboter initiierte Metakommunikation

13.4.1 Einleitung

Auf der anderen Seite liegt in der Metakommunikationsebene auch eine zusätzliche Möglichkeit für den Roboter dem Benutzer seine Fähigkeiten zu vermitteln

	ohne Metakomm.	mit Metakomm.
Parsingrate		
Roboter mit Aufgaben	95,31%	95,05%
Roboter ohne Aufgaben	92,71%	93,75%
Webtest mit Aufgaben	92,05%	94,60%
Webtest ohne Aufgaben	90,50%	93,02%
Turnfehlerrate		
Roboter mit Aufgaben	12,11%	10,42%
Roboter ohne Aufgaben	40,16%	34,38%
Webtest mit Aufgaben	29,05%	23,85%
Webtest ohne Aufgaben	44,18%	37,21%

Tabelle 13.6: Vergleich der Ergebnisse mit vs. ohne Metakommunikation für die zweite Version der Roboterapplikation

und ihm bei Kommunikationsproblemen weiterzuhelfen: Da der Benutzer nicht genau weiß, was der Roboter kann, wie auch an dem im vorigen Abschnitt erläuterten Austesten der Systemfähigkeiten deutlich wird, ist es für eine erfolgreiche Kommunikation essentiell, dass der Benutzer sich im Lauf der Kommunikation ein klareres Bild vom Roboter und seinen Fähigkeiten machen kann. Dieses Wissen des Benutzers über das System wird in einem Wissensmodell gespeichert und steht damit auch in künftigen Interaktionen mit dem Roboter zur Verfügung.

Daher wurde eine neue Generierungsstrategie entwickelt, die problematische Situationen erkennt und dem Benutzer dann auf der Metakommunikationsebene hilft, wenn es zu Problemen kommt. Diese proaktive Handlungsweise des Roboters trägt dazu bei, dass Probleme schneller erkannt und gelöst werden können und der Benutzer nicht in eine Fehlerspirale gerät, wie dies häufig bei heutigen Dialogsystemen der Fall ist. Dies soll auch mit Hilfe des abschließenden Benutzertests im Kapitel 14 evaluiert werden.

Aus Sicht des Roboters liegt eine problematische Situation vor, wenn die Benutzeräußerung nicht oder nur teilweise geparkt werden konnte oder wenn die Benutzeräußerung mit der Erwartung des Roboters oder mit den bereits im Diskurs vorhandenen Informationen inkonsistent ist. Außerdem fragen die Benutzer auch explizit nach Hilfe oder versuchen die vorhergehende Äußerung zu korrigieren. Daher wird hier eine Strategie entwickelt, die diese verschiedenen Problemfälle basierend auf der Benutzeräußerung und dem Diskurskontext erkennt und auflöst.

13.4.2 Erkennen problematischer Situationen

Es gibt verschiedene Faktoren, die anzeigen, dass der Benutzer wahrscheinlich Hilfe benötigt oder ein Problem aufgetreten ist:

- *Die Benutzeräußerung kann nicht oder nur teilweise geparkt werden.* In dem Fall muss der Roboter ohne einen Hinweis darauf, was der Benutzer

gesagt hat, reagieren. Es ist wahrscheinlich, dass die Benutzeräußerung außerhalb der vom System abgedeckten Domäne liegt, wie auch an dem folgenden Beispiel deutlich wird:

- Benutzer: Please cook me something.
(Bitte koch mir etwas.)
- Roboter: I cannot make any dishes by myself.
But I can tell you the recipe, if you want to.
(Ich kann nicht selbst kochen, aber ich kann dir Rezepte geben.)
- Benutzer: Are you sure you can't cook?
(Bist du sicher, dass du nicht kochen kannst?)

Hier fragt der Benutzer nach, ob der Roboter wirklich sicher ist, dass er nicht kochen kann, was momentan nicht vom System abgedeckt ist und daher zu einer nicht-parsbaren Äußerung führt. Auch Hintergrundgeräusche oder Husten und Lachen können aufgetreten sein und zu einer solchen nicht-parsbaren Äußerung geführt haben.

Wie auch in den Benutzerstudien (siehe Kapitel 4 und 5) deutlich wurde, ist es für den Benutzer sehr wichtig ist, immer eine Antwort vom System zu erhalten, da er sonst keinen Anhaltspunkt hat, was und wie viel der Roboter verstanden hat. D.h., selbst wenn der Roboter nichts verstanden hat, muss er dies kommunizieren. Dabei bittet er den Benutzer zunächst, die Position des Mikrofons zu überprüfen, seine Äußerung zu wiederholen, und entschuldigt sich schließlich, dass er ihn immer noch nicht verstanden hat, und bittet um eine Umformulierung der Äußerung.

- *Weder in der Benutzeräußerung noch im Diskurs findet sich ein Sprechakt.* In dem Falle sind nur Objekte und Eigenschaften im Diskurs, die möglicherweise falsch erkannt wurden, da sie mit keinem Sprechakt verbunden sind (weitere Details zu solchen elliptischen Äußerungen finden sich in Kapitel 11). Hier lassen sich mehrere Fälle unterscheiden:
 - Es wird ein Objekt erkannt: In dem Fall fragt der Roboter nach, was der Benutzer mit dem Objekt machen möchte.
 - Es wird eine Eigenschaft erkannt: Hier stellt der Roboter eine Klärungsfrage, welches Objekt der Benutzer meint, wenn es mehrere Objekte mit der gleichen Eigenschaft gibt. Gibt es nur ein Objekt mit der erkannten Eigenschaft, so fragt der Roboter den Benutzer wiederum, was er mit diesem Objekt machen möchte.
 - Es wird eine Liste von Objekten erkannt: Im Gegensatz zu den ersten beiden Fällen haben die Benutzerstudien (cf. Kapitel 4) hier gezeigt, dass es sich bei diesen Listen von Objekten in allen Fällen um Fehlerkennungen handelte und die Benutzer an der Stelle etwas Anderes gesagt hatten, wie an dem folgenden Beispiel zu sehen:

Benutzer: Can you help me clean the kitchen.
 (Kannst du mir helfen,
 die Küche sauber zu machen?)
 Erkannt: *tomato milk green and kitchen*
 (*Tomaten Milch grün und Küche*)

Hier fängt der Dialog mit einer Benutzeräußerung an, die vom Roboter falsch verstanden wurde als eine Liste von verschiedenen Objekten. Da es sich um die erste Äußerung im Dialog handelte, war der Diskurs noch leer. Da die Benutzeräußerung ferner selbst ebenfalls keinen Sprechakt enthält, befindet sich im Diskurs kein Sprechakt, sondern nur eine Liste von Objekten. Diese Erkennungsprobleme bei langen Listen erklären sich vor allem durch die hohe Flexibilität der Listen, die es ermöglichen, alle Objekte der Ontologie als eine lange Liste einzugeben, und damit eine hohe Perplexität aufweisen (für Details über die Verkettung beliebiger Nominalphrasen siehe Kapitel 12). Solche Listen sind damit nur als Antwort auf eine Klärungsfrage vom Dialogmanager sinnvoll und werden in allen anderen Fällen als Fehlerkennungen klassifiziert. Daher entschuldigt sich der Roboter an dieser Stelle und fragt nach, was der Benutzer gesagt hat. Dies wird auch durch die Tatsache bestätigt, dass solche Listen von Objekten in den textbasierten Benutzerstudien übers Internet nicht vorkamen (cf. Kapitel 5), während Benutzeräußerung mit nur einem Objekt oder einer Eigenschaft besonders im Fall von vorher auftretenden Problemen jedoch sehr häufig in den webbasierten Tests zu finden waren.

- *Die Benutzeräußerung ist inkonsistent mit dem aktuellen Diskurs oder mit den Erwartungen des Roboters.* Das kann entweder daran liegen, dass die Benutzeräußerung falsch verstanden wurde oder dass der Benutzer etwas Unerwartetes gesagt und beispielsweise einen neuen Subdialog angefangen hat. In dem Fall ist es wichtig, die Diskursinformationen auch für den nächsten Turn beizubehalten, so dass eine einzelne falsch erkannte Äußerung nicht dazu führt, dass ein Dialogziel unmittelbar abgebrochen wird, sondern der Benutzer noch mit diesem Dialogziel fortfahren kann:

Benutzer: What can be made?
 (Was können wir machen?)
 Roboter: For which recipe are you looking?
 (Nach welchem Rezept suchst du?)
 Benutzer: Something with eggs.
 (Etwas mit Eiern.)
 Erkannt: *something to eat and eggs*
 (*etwas zu essen und Eier*)

In diesem Beispiel wurde die Benutzeräußerung “etwas mit Eiern” fälschlicherweise als “etwas zu essen und Eier” erkannt. Dies entspricht nicht den Erwartungen des Roboters, der auf einen Rezeptnamen als Antwort auf seine Klärungsfrage wartet.

- *Der Benutzer fragt immer wieder nach der gleichen Information.* In diesem Fall ist es sehr wahrscheinlich, dass es einen Erkennungsfehler gab, und dass der Benutzer z.B. nach etwas außerhalb der Domäne des Systems gefragt hat. Dies wird auch an dem folgenden Beispiel deutlich, in dem der Benutzer die Information darüber, was sich im Kühlschrank befindet, mehrfach erhält:

Benutzer: Let's start a new task. Can you help me with cooking dinner?
(Lass uns eine neue Aufgabe anfangen. Kannst du mir helfen, das Abendessen zu kochen?)

Roboter: I will have a look in the fridge. There are tomatoes, apples, eggs, milk, butter, cucumbers, coke, juice and soda.
(Ich sehe im Kühlschrank nach. Wir haben Tomaten, Äpfel, Eier, Milch, Butter, Gurken, Cola, Saft und Soda.)

Benutzer: Okay, sounds like we can make a salad with tomatoes and cucumbers.
(Okay, das klingt als könnten wir einen Salat mit Tomaten und Gurken machen.)

Roboter: I beg your pardon?
(Wie bitte?)

... <Mehrere Fehlerkennungen>

Benutzer: OK, what's a good salad recipe with tomatoes?
(Okay, was ist ein gutes Rezept für einen Tomatensalat?)

Roboter: I will have a look in the fridge. There are tomatoes, apples, eggs, milk, butter, cucumbers, coke, juice and soda.
(Ich sehe im Kühlschrank nach. Wir haben Tomaten, Äpfel, Eier, Milch, Butter, Gurken, Cola, Saft und Soda.)

- *Der Benutzer fragt explizit nach Hilfe.* Wenn der Benutzer beispielsweise den Roboter fragt "What can you do for me?" ("Wie kannst du mir helfen?"), wird der Sprechakt "nach Hilfe fragen" erkannt und der Benutzer bekommt Informationen vom Roboter über seine Fähigkeiten.
- *Der Benutzer korrigiert explizit eine vorhergehende Äußerung.* Bestimmte Worte, wie "nein", "nicht", "aber" in der Benutzeräußerung deuten daraufhin, dass der Benutzer einen Fehler korrigieren will. Hier greifen die bereits im Abschnitt 13.3.3 beschriebenen Sprechakte zur Korrektur von Benutzeräußerungen.

- Benutzer: Can you help me cut up the ingredients?
(Kannst du mir helfen, die Zutaten klein zu schneiden?)
- Roboter: I see. So you want the red cup.
(Ich verstehe. Du möchtest den roten Becher.)
- Benutzer: No, I would like you to help me cut the ingredients for the salad.
(Nein, ich möchte, dass du mir hilfst, die Zutaten für den Salat klein zu schneiden.)

Hier hat der Roboter die Aufforderung des Benutzers, ihm dabei zu helfen, die Zutaten für den Salat zu schneiden, falsch verstanden und dachte, der Benutzer wolle den roten Becher. Der Benutzer hat dies explizit verneint und dem Roboter erneut erklärt, dass er Hilfe benötigt beim Schneiden der Zutaten für den Salat.

13.4.3 Finite-State-Automat

Um problematische Situationen zu erkennen, wurde ein Finite-State-Automat (FSA) entwickelt, der die folgenden vier Zustände enthält (siehe Abbildung 13.3):

- **Fehlerzustand:** Der Benutzer will eine vorhergehende Äußerung korrigieren.
- **Hilfezustand:** Der Benutzer ist verwirrt und weiß nicht, wie er weiter machen kann und benötigt daher Hilfe vom Roboter über seine Fähigkeiten.
- **Normalzustand:** Es gibt momentan keine problematische Situation.
- **Startzustand:** Hier beginnt der Dialog mit einem zunächst leeren Diskurs. Der Benutzer kann auch explizit den Diskurs während des Dialogs wieder leeren und so in den Startzustand zurückkehren.

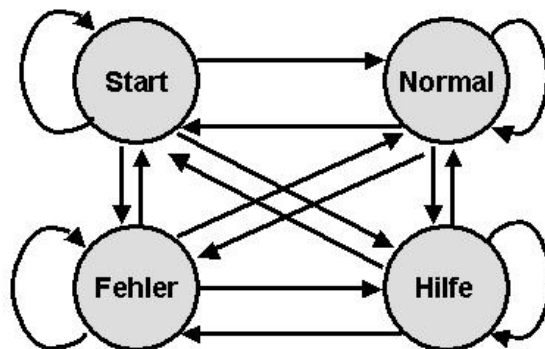


Abbildung 13.3: Finite-State-Automat mit vier Zuständen

Diese Zustände und ihre Übergänge sind wie folgt definiert:

$s_i = \text{Zustand}$ (inkl. Start, Normal, Fehler, Hilfe)

$n_i = \text{Fehlerspirale}$

$h_i = \text{Hilfsbedürftigkeit}$

$u_i = \text{Eingabetyp}$

wobei *Eingabetyp* = *OK*
 = *ERH* bei expliziter Nachfrage nach Hilfe
 = *EEC* bei expliziter Fehlerkorrektur
 = *EAB* bei explizitem Abbruch
 = *P* bei anderen Problemen

$d_i = 1$ wenn etwas im Diskurs steht
 = 0 andernfalls

$s_i = f(s_{i-1}, u_i, d_i, n_i, h_i)$
 = *Normal* wenn $u_i = ok$
 = *Fehler* wenn $u_i = EEC \vee u_i = P \wedge d_i = 1$
 = *Hilfe* wenn $u_i = ERH \vee u_i = P \wedge d_i = 0$
 = *Start* wenn $s_{i-1} = P \wedge n_{i-1} > N \vee u_i = EAB$

$n_i = n_{i-1} + 1$ wenn $u_i = EEC \vee u_i = P \wedge d_i = 1$
 = 0 andernfalls

$h_i = n_{i-1} + 1$ wenn $u_i = ERH \vee u_i = P \wedge d_i = 0$

$h_i = n_{i-1} - 1$ wenn $u_i = OK \wedge u_i > X$

Einerseits kann der Benutzer explizit Hilfe anfordern, eine Korrektur vornehmen oder den Diskurs leeren und auf diese Weise von einem Zustand in einen anderen gelangen. Auf der anderen Seite werden die Übergänge zwischen den Zuständen von den Benutzeräußerungen, den möglichen Problemen in ihnen, und der Art der Antworten des Roboters bestimmt, wie in der obigen Definition verdeutlicht. Der Dialog beginnt im Startzustand und geht von dort in den Normalzustand über, solange keine Probleme auftreten. Kommt es zu einer problematischen Situation, gibt es einen Übergang in den Fehlerzustand, wenn der Diskurs gefüllt ist und es daher für den Benutzer Informationen gibt, die er korrigieren möchte, bzw. in den Hilfezustand, wenn der Diskurs leer ist.

Der Benutzer bleibt solange im Hilfe- bzw. Fehlerzustand, wie die Probleme anhalten. Nach einer nicht-problematischen Äußerung gelangt der Benutzer wieder in den Normalzustand. Um vom Hilfezustand in den Fehlerzustand zu gelangen, muss die Benutzeräußerung Informationen enthalten, die in den Diskurs geschrieben wird und die der Benutzer korrigieren will. Umgekehrt, um vom Fehlerzustand zum Hilfezustand zu gelangen, muss der Diskurs durch die

Benutzeräußerung gelöscht werden, was beispielsweise dadurch geschehen kann, dass die Benutzeräußerung nicht den Erwartungen des Roboters entspricht und auch keinen Sprechakt enthält.

Um die aktuelle Situation zu verfolgen, werden die folgenden Variablen verwendet:

HILFSBEDÜRFTIGKEIT. Die Hilfsbedürftigkeit spiegelt das Bedürfnis des Benutzers nach Hilfe wider. Es gibt 3 verschiedene Bedingungen, unter denen diese Variable ansteigt: Wenn eine Benutzeräußerung nicht geparkt werden konnte oder nicht den Erwartungen des Roboters entspricht oder wenn der Benutzer diese Information bereits mehrfach bekommen hat. Der Wert der Variable sinkt wieder, wenn die Kommunikation funktioniert, so dass keine problematischen Situationen mehr auftreten. Dennoch geht sie nicht wieder auf 0 zurück, um anzuzeigen, dass es in diesem Dialog bereits gravierende Probleme gab. Damit ist auch gewährleistet, dass bei weiteren auftretenden Fehlern der Benutzer schnell wieder adäquate Hilfe bekommen kann.

FEHLERSPIRALE. Diese Variable spiegelt die Anzahl an Turns wieder, die der Benutzer benötigt, um einen Fehler zu korrigieren. Ihr Wert steigt an, wenn problematische Situationen vorliegen - ähnlich der Hilfsbedürftigkeit. Aber im Gegensatz zur Hilfsbedürftigkeit, wo der Diskurs leer ist, benötigt die Fehlerspirale eine Information im Diskurs, die der Benutzer ändern will. Außerdem wird die Fehlerspirale auf 0 gesetzt, wenn das Problem gelöst ist, wohingegen die Hilfsbedürftigkeit auf einem gewissen Niveau verbleibt.

BENUTZERWISSEN. Diese Variable enthält alle Informationen, die der Benutzer bereits im Verlauf des Dialogs erhalten hat zusammen mit einer Angabe darüber, wie oft er schon darüber informiert wurde. Wie sich auch in den Benutzertests gezeigt hat (siehe Kapitel 4 und 5), möchte der Benutzer die gleiche Information nicht mehrfach bekommen, wenn er nicht explizit danach fragt. So führte es bei vielen Benutzern zu Frust und Ärger, wenn sie immer wieder die gleiche Antwort vom Roboter erhalten, wie auch an dem folgenden Beispiel deutlich wird, an dessen Ende der Benutzer entnervt aufgibt:

- Benutzer: Is there bread for a sandwich?
(Haben wir Brot für ein Sandwich?)
- Roboter: I will have a look at the fridge.
There are tomatoes, apples, eggs, milk, butter,
cucumbers, coke and juice.
(Ich schaue mal im Kühlschrank nach.
Da sind Tomaten, Äpfel, Eier, Milch, Butter,
Gurken, Cola und Saft.)
- Benutzer: Is there any bread?
(Ist dort auch Brot?)
- Roboter: I will have a look at the fridge.
There are tomatoes, apples, eggs, milk, butter,
cucumbers, coke and juice.
(Ich schaue mal im Kühlschrank nach.
Da sind Tomaten, Äpfel, Eier, Milch, Butter,
Gurken, Cola und Saft.)

Diese Informationen beziehen sich beispielsweise auf den Inhalt des Kühlschranks und der Speisekammer, die Rezepte, die der Roboter kennt, und die Tätigkeiten, die der Roboter ausführen kann. Kennt der Benutzer diese Informationen bereits, so wird davon ausgegangen, dass ein Spracherkennungsfehler vorlag und daher fragt der Dialogmanager nach, ob der Benutzer tatsächlich diese Information noch einmal hören möchte oder ob er ihn missverstanden hat.

Außerdem ändern sich die Formulierungen der Antworten des Roboters, um sicherzustellen, dass der Benutzer nicht die exakt gleiche Antwort in zwei aufeinander folgenden Turns vom Roboter bekommt. Dies sollte zum einen die Benutzerzufriedenheit erhöhen und zum anderen dafür sorgen, dass der Benutzer verschiedene Formulierungen für die gleiche Anfrage benutzt, um damit die Erkennungsrate zu erhöhen. Wie in einer Evaluation von Goldberg et al. (Goldberg et al. 2003) gezeigt, führen verschiedene Formulierungen in den Antworten eines Dialogsystems dazu, dass auch die Benutzer unterschiedliche Formulierungen benutzen, um das Gleiche auszudrücken, und dass dadurch die Erkennungsrate erhöht werden kann und sich Fehlerspiralen verkürzen lassen.

Dieses Benutzerwissen lässt sich auch in einer generellen Wissenskomponente speichern, so dass der Roboter, wenn er wieder mit dem gleichen Benutzer spricht, bereits weiß, was er ihm schon alles gesagt hat. Fragt der Benutzer immer wieder nach den gleichen Informationen, so steigt die oben erwähnte Hilfsbedürftigkeit des Benutzers.

ERWARTUNG DES SYSTEMS. Diese Variable spiegelt die Erwartungen des Roboters in bezug auf die nächste Benutzeräußerung wider. Wenn die Benutzeräußerung nicht den Erwartungen entspricht, so steigt die Hilfsbedürftigkeit an. Dabei geht es vor allem um die Antworten des Benutzers auf Nachfragen des Roboters, da sich in diesen Fällen eine Erwartung aufbauen lässt. Diese spezifischen Erwartungen des Dialogmanagers sind an die jeweiligen Dialogziele und gerade gestellten Klärungsfragen geknüpft, so dass der Dialogmanager an dieser Stelle erwartet, dass der Benutzer eine ihm gestellte Frage beantwortet.

DIALOGLÄNGE. Diese Variable enthält die Anzahl an Benutzerturns seit Beginn des Dialogs. Mit ihrer Hilfe kann gesteuert werden, dass der Benutzer zu Beginn des Dialogs beispielsweise darin unterstützt wird, das Mikrofon richtig zu platzieren. Im weiteren Verlauf des Dialogs kann man dann davon ausgehen, dass das Mikrofon richtig aufgesetzt ist. Daher erhält der Benutzer nun bei Problemen Hilfe über die Fähigkeiten des Roboters, da es in dieser Situation wahrscheinlicher ist, dass der Benutzer etwas gesagt hat, was bisher nicht in der Grammatik enthalten ist.

Im Hilfezustand bekommt der Benutzer Informationen über die Fähigkeiten des Roboters. Da diese zu komplex sind, um sie in einer einzigen Äußerung zu erläutern, wurde eine hierarchische Struktur aller Fähigkeiten des Roboters entwickelt (siehe Abbildung 13.4), so dass der Benutzer zunächst generelle Hilfe erhält und schließlich immer detailliertere Informationen zu den Fähigkeiten des Roboters. Auf diese Weise bekommt der Benutzer weitere Informationen über den Roboter, die ihm auch zukünftig helfen können.

Darüberhinaus weist der Roboter den Benutzer auch explizit auf Aufgaben hin, die er nicht ausführen kann, um zu vermeiden, dass der Benutzer immer wieder danach fragt und es immer wieder zu Spracherkennungsfehler kommt.

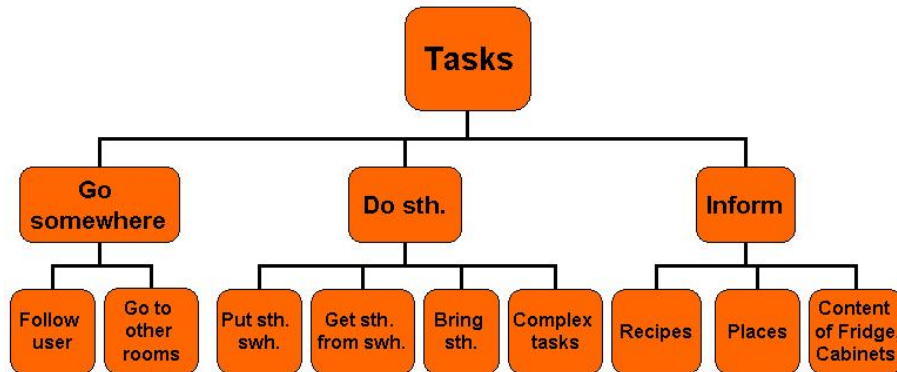


Abbildung 13.4: Hierarchische Struktur der Fähigkeiten des Roboters

Andererseits gelangt der Benutzer ebenfalls in den Hilfezustand, wenn er von sich aus nach einer Aufgabe fragt, die der Roboter nicht ausführen kann, wie beispielsweise “putzen”. Er bekommt in dem Fall weitere Informationen über die Aufgaben, die der Roboter ausführen kann.

Im Fehlerzustand wird der Diskurs aufrechterhalten und der Benutzer erhält somit die Möglichkeit die vorhergehende Äußerung zu korrigieren. Wenn der Fehler schließlich gelöst ist, wird die Fehlerspirale auf 0 gesetzt und der Benutzer gelangt zurück in den Normalzustand.

13.4.4 Benutzerstudie

Es wurde eine Benutzerstudie durchgeführt, um die Auswirkungen dieser Generierungsstrategie im Detail zu evaluieren. Dafür wurden zwei verschiedene Versionen entwickelt, wobei Version A nur einen gemeinsamen Zustand für Hilfe und Fehler hatte und Version B die oben erwähnte Trennung von Hilfe- vs. Fehlerzustand enthielt. Daher gab es in Version A nur eine Variable, die alle Probleme verfolgte, während in Version B Hilfsbedürftigkeit und Fehlerspirale unabhängig voneinander verwendet wurden. Außerdem bekamen die Benutzer in der ersten Version bei Problemen eine allgemeine Hilfe angeboten, während sie in der zweiten Version die Möglichkeit hatten, vorhergehende Äußerungen, die zu Problemen führten, zu korrigieren.

An der Benutzerstudie nahmen 16 Personen teil, 8 für Version A und 8 für Version B. Mehr als die Hälfte der Teilnehmer waren englische Muttersprachler; die restlichen Benutzer sprachen alle fließend englisch. Das Alter der Versuchspersonen reichte von 26 bis 50 Jahre, mit einem Durchschnittsalter von 36 Jahren. Alle kannten sich gut mit Computern aus, aber nur 6 von ihnen hatten bereits einmal mit einem Dialogsystem gearbeitet.

Die Benutzerstudie bestand aus 3 Teilen: Zunächst eine freie Interaktion mit dem Roboter, dann ein aufgabenbasierter Testteil und schließlich ein Fragebogen, in dem die Benutzer ihre Eindrücke während des Experiments wiedergaben. Bei den verwendeten Aufgaben handelte es sich um die gleichen wie in den anderen aufgabenbasierten Experimenten, die hier durchgeführt wurden (siehe Abbildung 5.6), da sie sich dort bereits bewährt haben. Der Fragebogen diente

	Version A	Version B
Turnfehlerrate (gesamt)	55,0%	48,2%
Turnfehlerrate (aufgabenbasiert)	48,3%	31,6%
Aufgabenerfüllungsrate	70,0%	96,3%
Turns pro Aufgabe	5,1	2,7

Tabelle 13.7: Auswertung der Benutzerstudie für die Version A (nur ein Zustand für Hilfe und Fehlerbehandlung) vs. Version B (Getrennte Zustände für Hilfe und Fehlerbehandlung)

dazu, den generellen Eindruck der Teilnehmer von dem System wiederzugeben und ihre Einschätzung sowohl zu ihrer eigenen Leistung als auch zu der des Systems zu ermitteln und ähnelt damit dem in Kapitel 7.2.2 dargestellten. Es gab die folgenden quantitativen Fragen, die die Benutzer mit Werten von 1-5, wobei 1 das beste und 5 das schlechteste war, beantworteten:

- Wie zufrieden sind Sie mit dem Roboter?
- Wie erfolgreich waren Sie bei dem Test?
- Würden Sie einen solchen Roboter gern benutzen, um Ihnen bei der Hausarbeit zu helfen?

Daneben gab es einige qualitative Fragen mit Freitextantworten, wo die Benutzer ihre Probleme darlegen konnten, mögliche Ursachen für diese Probleme und ihre Vorschläge für künftige Verbesserungen.

Die Ergebnisse zeigen, dass die Unterscheidung in Fehler- vs. Hilfezustand sehr vielversprechend ist, da die Turnfehlerraten geringer sind in Version B als in Version A (siehe Tabelle 13.7). Außerdem werden mehr Aufgaben erfolgreich abgeschlossen (96,3% gegenüber 70% in Version A) und die Benutzer benötigen erheblich weniger Turns pro Aufgabe (5,1 gegenüber 2,7 in Version A).

Auch die Ergebnisse des Fragebogens unterstützen dies, da in Version B der Roboter den Benutzern besser gefällt, sie sich für erfolgreicher gehalten haben und sie einen solchen Roboter eher nochmal benutzen würden (siehe Abbildung 13.5). In den Freitext-Antworten haben einige Benutzer der Version B sogar erwähnt, wie gut ihnen die Fehlerbehandlung und die Fähigkeit des Roboters, wieder aus problematischen Situationen herauszukommen, gefallen hat. Sie fanden es gut, wenn der Roboter Nachfragen stellte und lobten, dass der Roboter ihnen sehr schnell seine Fähigkeiten verdeutlicht hat. Generell gefiel es ihnen nicht, dass er so langsam war und momentan noch nicht so viele Fähigkeiten hat und beispielsweise weder saubermachen noch selbst kochen kann.

13.5 Zusammenfassung

Vergleicht man zwischenmenschliche Dialoge und Mensch-Roboter-Dialoge in bezug auf die verwendeten Strategien zur Fehlervermeidung und -behebung,

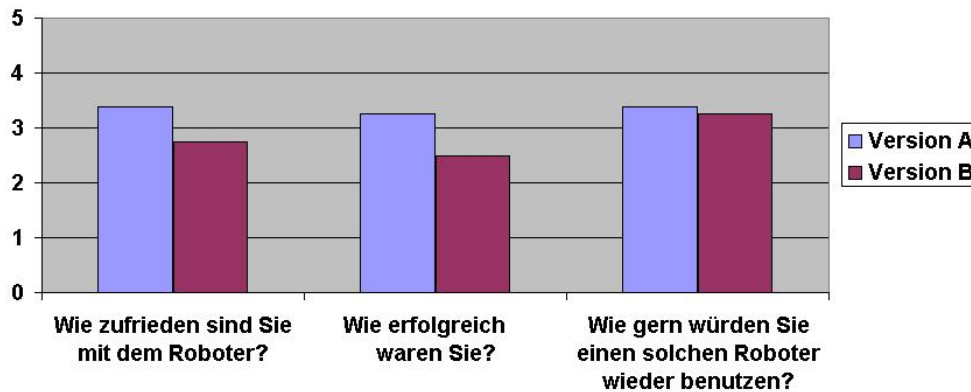


Abbildung 13.5: Auswertung des Fragebogens für Version A (nur ein Zustand für Hilfe und Fehlerbehandlung) vs. Version B (Getrennte Zustände für Hilfe und Fehlerbehandlung): Die Angaben reichen von 1 “sehr gut/sehr gern” bis 5 “überhaupt nicht/keinesfalls”

so fällt auf, dass, obwohl Menschen zum Verdeutlichen von Fehlern explizitere Strategien anwenden, dennoch die Strategien zur Fehlerkorrektur und ihre Anwendung in der Kommunikation sehr ähnlich sind.

Zur besseren Fehlerkorrektur wurden ferner einige neue Methoden implementiert zur Metakommunikation, die in einer Evaluation zusammen mit den in den vorhergehenden Kapiteln entwickelten Methoden zur Fehlervermeidung abschließend getestet werden (siehe Kapitel 14). Erste Ergebnisse mit den in den Robotertests gewonnenen Daten zeigen hier bereits eine deutliche Verbesserung, so dass mehr Turns in die korrekte Semantik übertragen werden können (45% gegenüber 51,8%).

Besonders interessant ist in diesem Zusammenhang auch die Möglichkeit des Dialogmanagers auf der Metakommunikationsebene zu interagieren. Hiermit kann der Roboter dem Benutzer seine Fähigkeiten vermitteln und vermeidet damit proaktiv Fehler, die dadurch entstehen, dass der Benutzer auf Aufgaben referenziert, die der Roboter nicht ausführen und auch nicht verstehen kann. Dies wirkt sich auch positiv auf die Benutzerzufriedenheit aus. Darüberhinaus kann diese Strategie auch zur schnelleren Klärung von problematischen Situationen beitragen, indem sie diese mit Hilfe bestimmter Problemindikatoren entdeckt und den Benutzern Hilfe anbietet.

Teil IV

Evaluation und Fazit

Kapitel 14

Diskussion und Evaluation der Mechanismen zur Fehlervermeidung und -behandlung

14.1 Einleitung

In den vorangegangenen Kapiteln wurden verschiedene Methoden zur Fehlervermeidung und -behandlung vorgestellt. Dafür wurden zum einen Mechanismen zur automatischen Wissenserweiterung vor Beginn des eigentlichen Dialogs integriert als auch Lösungen für häufige Fehler, die durch Ellipsen, Anaphern und komplexe Äußerungen entstehen, implementiert. Ferner wurden Methoden zur Metakommunikation und Fehlerbehebung entwickelt. Allen Ansätzen gemeinsam ist, dass sie sich generisch implementieren lassen und damit eine über das Roboterszenario hinausgehende Fehlervermeidung für Dialogsysteme bieten.

Im Folgenden soll nun anhand einer Evaluation gezeigt werden, dass diese Methoden zur Fehlerbehandlung objektiv messbar den Dialog mit dem Haushaltsroboter verbessern. Dafür wurde eine neue Version der Roboterapplikation entwickelt, die auf der in Kapitel 8 vorgestellten Version 2 der Roboterapplikation aufbaut und alle Fehlerbehandlungsmechanismen enthält (im Folgenden als Version F bezeichnet). Darüberhinaus beschäftigt sich dieses Kapitel mit der Übertragbarkeit der entwickelten Mechanismen zur Fehlerbehandlung auf andere Applikationen außerhalb der Roboterdomäne.

14.2 Evaluation

14.2.1 Einleitung

Es gibt verschiedene Evaluationsmethodologien für Dialogsysteme, angefangen bei recht einfachen, die beispielsweise eine Referenzantwort verwenden (Hirschmann et al. 1990), bis hin zu dem heutzutage bekanntesten Ansatz zur Dialogsystemevaluation namens *Paradise* (Walker et al. 1997). Dieser Ansatz erlaubt

es, zwei verschiedene Dialogsysteme bzw. Dialogstrategien mit Hilfe einer Performanzfunktion zu vergleichen (cf. Kapitel 2). Eine Erweiterung von *Paradise* namens *Promise* berücksichtigt multimodale Fragestellungen (Behringer et al. 2002) und verwendet den Pearson Korrelationskoeffizienten anstatt der mehrfachen linearen Regression zwischen Benutzerzufriedenheit und Kostenfunktionen, wie in *Paradise*. In *Promise* werden Informationsstücke als oberstes Konzept verwendet, und eine Aufgabe wird als erfolgreich abgeschlossen angesehen, wenn alle notwendigen Informationsstückchen vorhanden sind.

In der vorliegenden Arbeit wird eine Evaluation durchgeführt, die an *Paradise* (Walker et al. 1997) angelehnt ist und ebenfalls Merkmale, wie Turnfehler-rate, Anzahl an Turns, Erfolg der Aufgabenlösung und Benutzerzufriedenheit, verwendet. Es wird hier allerdings nicht versucht, ein einziges Maß zum Vergleich zu entwickeln, sondern es sollen bewusst mehrere Maße nebeneinandergestellt werden, um sie unabhängig voneinander vergleichen zu können. Außerdem hat es sich bereits in einigen Benutzerstudien gezeigt, dass sich die verschiedenen Erfolgsmaße gegenseitig in unterschiedlicher Art und Weise beeinflussen können (Gieselmann, Stenneken 2006).

14.2.2 Benutzerstudie

Um die Ergebnisse der Benutzerstudie mit denen vorhergehender Benutzertests vergleichen zu können, wurde wiederum das Küchenszenario gewählt und die Benutzer agierten zunächst frei mit dem Roboter und bekamen dann die gleichen Aufgaben, die sich bereits in den zuvor durchgeführten Benutzertests bewährt hatten (siehe Abbildung 5.6). Zum Abschluss füllten die Benutzer wiederum einen Fragebogen aus, um ihren generellen Eindruck von dem System wiederzugeben. Dieser enthielt die bereits im vorhergehenden Kapitel erläuterten Fragen, in denen die Benutzer ihre Einschätzung sowohl zu ihrer eigenen Leistung als auch zu der des Systems abgeben konnten.

Bei diesen Tests wurde eine Mischform zwischen webbasierten Tests und Tests mit dem Roboter gewählt, um die Vorteile beider Modi auszunutzen und ihre Nachteile vermeiden zu können: Die simulierte gesprochensprachliche Interaktion mittels eines Computers. D.h., die Versuchspersonen interagierten mit dem Dialogsystem über sprachliche Eingaben mit Hilfe des Spracherkenners und erhielten Rückmeldungen über sprachliche Ausgaben per Lautsprecher. Dazu wurde den Versuchspersonen zu Beginn des Experiments erklärt, dass sie sich vorstellen sollten, blind zu sein und einen Haushaltsroboter zu ihrer Hilfe zur Verfügung zu haben, mit dem sie per gesprochener Sprache kommunizieren können. Dieser Aufbau ist damit realistischer als ein webbasierter Test, da die Benutzer mit dem System per gesprochener Sprache kommunizieren, und gleichzeitig nicht so eingeschränkt wie der Test mit dem Roboter, bei dem nur Aufgaben verwendet werden können, die der Roboter ausführen kann.

Insgesamt nahmen an der Benutzerstudie 16 Personen teil. Alle Teilnehmer sprachen fließend englisch; 56,25% waren englische Muttersprachler. Die Altersstruktur der Versuchspersonen war sehr gemischt und reichte von 26 bis 50 Jahre mit einem Durchschnittsalter von 36 Jahren. Alle kannten sich gut mit Computern aus, aber nur 37,5% von ihnen hatten bereits einmal mit einem

	Ohne Aufgaben	Mit Aufgaben
Parsingrate	92,34%	96,30%
Turnfehlerrate	63,33%	39,90%
Rate gelöster Aufgaben	N/A	83,1%
Durchschn. Dialoglänge (Turns)	26,19	31,25
Turns pro Aufgabe	4,93	3,88

Tabelle 14.1: Auswertung des Benutzertests für die neue Version F mit Fehlerbehandlung

Dialogsystem gearbeitet.

Erwartungsgemäß verlief die Kommunikation im aufgabenbasierten Testteil erfolgreicher als in dem Testteil mit freier Interaktion (siehe Tabelle 14.1): Im aufgabenbasierten Testteil können mehr Turns geparkt werden und die Turnfehlerrate ist geringer. Außerdem scheinen die Benutzer motivierter zu sein, länger mit dem Roboter zu sprechen, da die durchschnittliche Dialoglänge beim aufgabenbasierten Testteil höher ist als bei dem Testteil ohne Aufgaben. Wie erwartet ist die durchschnittliche Anzahl an Turns pro Aufgabe geringer im aufgabenbasierten Teil als in der freien Interaktion, da die Benutzer sich nur auf Aufgaben beziehen, die der Roboter generell lösen kann.

Da sich dieser Test durch den Aufbau als simulierte gesprochensprachliche Interaktion von den in den Kapiteln 4 und 5 durchgeführten Tests mit dem Roboter bzw. über das Internet unterscheidet, lassen sich die Ergebnisse nicht unmittelbar übertragen. Daher wurde zur besseren Vergleichbarkeit der gleiche Benutzertest auch mit einer Kontrollgruppe durchgeführt. Hier wurde das Baselinesystem ohne Fehlerbehandlungsmechanismen verwendet, wie es auch in den in Kapitel 4 und 5 verwendet wurde. Die Versuchspersonen der Kontrollgruppe verwendeten wieder einen Computer, interagierten mit dem Dialogsystem über sprachliche Eingaben mit Hilfe des Spracherkenners und erhielten sprachliche Rückmeldungen per Lautsprecher. Zu der Kontrollgruppe gehörten 3 Benutzer, die allesamt englische Muttersprachler waren. Alle kannten sich gut mit Computern aus und 66% hatten bereits mit einem Dialogsystem gearbeitet.

Vergleicht man die Ergebnisse der beiden Tests, so sieht man eine deutliche Verbesserung in der Version F (cf. Abbildung 14.1): Die Turnfehlerrate sinkt und die Parsingrate steigt in beiden Testteilen - mit und ohne Aufgaben - von der Baselineversion zur Version F an. Es können sowohl mehr Äußerungen geparkt als auch in die korrekte Semantik übertragen werden. Hinzu kommt, dass es die Benutzer im aufgabenbasierten Teil schaffen, mehr Aufgaben zu lösen (65,0% in der Baselineversion gegenüber 83,1% in der Version F), und dass sie dafür sogar erheblich weniger Turns benötigen (6,72 in der Baselineversion gegenüber 3,88 in der Version F). Dies zeigt deutlich, dass die Benutzer mit der neuen Version effektiver arbeiten können als mit der Version ohne Fehlerbehandlungsmechanismen, da sie in kürzerer Zeit mehr Aufgaben lösen können.

Ausserdem sind die Benutzer mit der Version F zufriedener als mit der Baselineversion, wie sich aus der Auswertung des Fragebogens ableiten lässt (cf. Abbildung 14.2): In der Version F sind die Benutzer sowohl mit dem Roboter

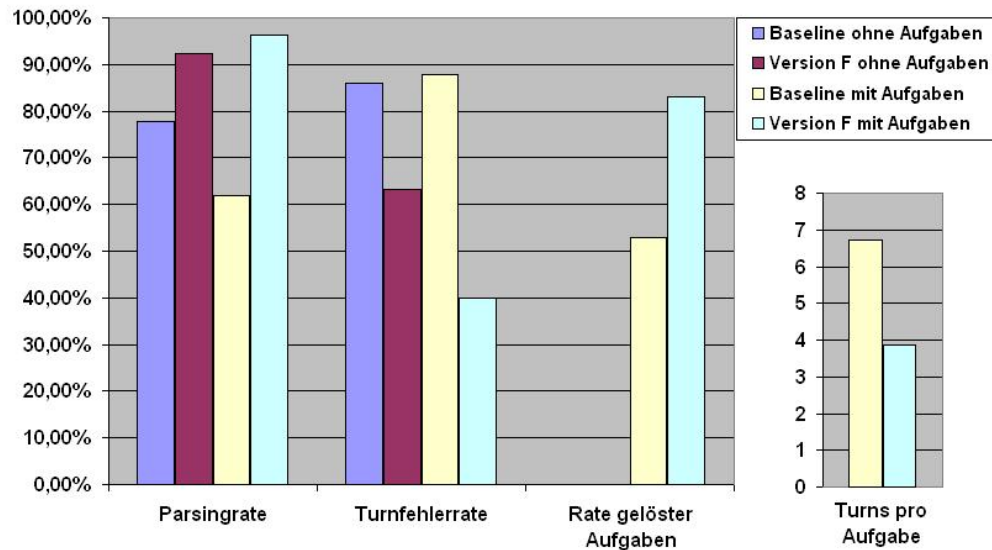


Abbildung 14.1: Gegenüberstellung der Ergebnisse der neuen Version mit Fehlerbehandlung (Version F) mit denen der Baselineversion

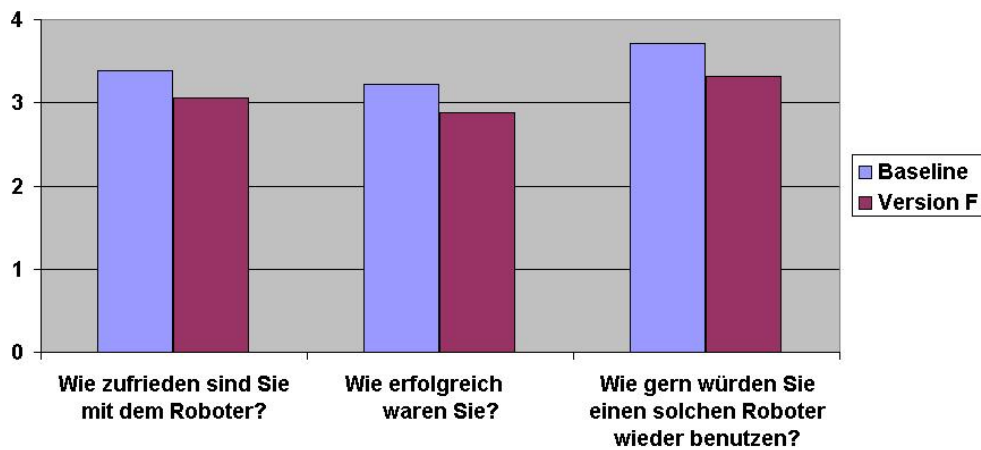


Abbildung 14.2: Auswertung des Fragebogens für die Baselineversion vs. Version F (mit Fehlerbehandlung): Die Angaben reichen von 1 "sehr gut/sehr gern" bis 5 "überhaupt nicht/keinesfalls"

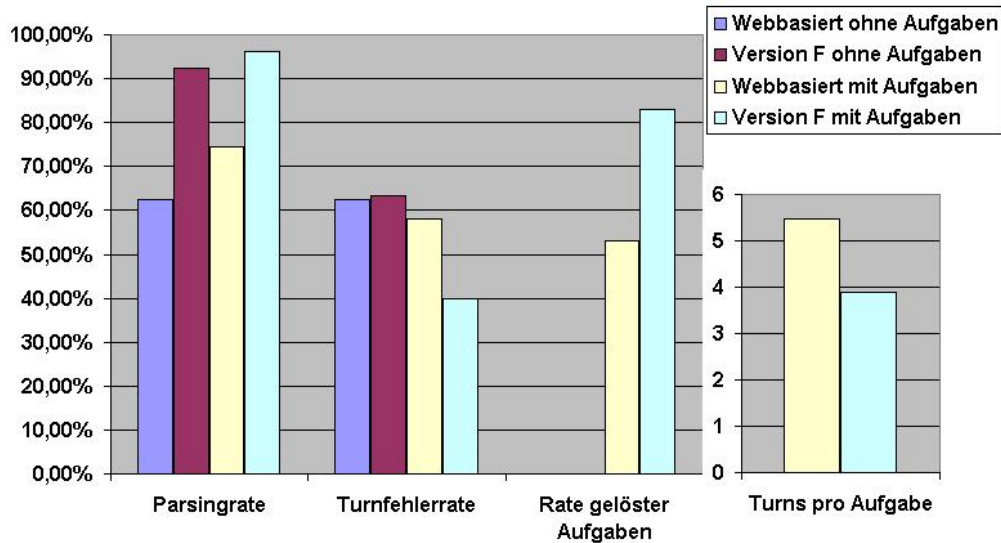


Abbildung 14.3: Gegenüberstellung der Ergebnisse der neuen Version mit Fehlerbehandlung (Version F) mit denen des Internettests

als auch mit ihrer eigenen Leistung zufriedener und tendieren eher dazu, einen solchen Roboter erneut zu benutzen.

Obwohl der webbasierte Test für die Benutzer einfacher war, da sie Text eingegeben konnten und nicht mit einem Spracherkenner und dessen Fehlerkennungen konfrontiert waren, ist auch hier eine deutliche Verbesserung zu erkennen, wenn man die Ergebnisse dieses Benutzertests mit denen des im Kapitel 5 beschriebenen webbasierten Tests vergleicht (siehe Abbildung 14.3). In beiden Testteilen (mit vs. ohne Aufgaben) konnten in der Version F mehr Sätze geparkt werden. Für den Experimentteil ohne Aufgaben sind die Turnfehlerraten für beide Versionen (Baseline vs. Version F) sehr ähnlich. Dies lässt sich darauf zurückführen, dass hier die individuellen Unterschiede eine größere Rolle spielen: Manche Benutzer sind sehr kreativ, was das Ausdenken neuer Aufgaben für den Roboter angeht, während andere nur wenig ausprobieren. Damit sind die Möglichkeiten für Fehler sehr unterschiedlich je nach Charakter des Benutzers. Im internetbasierten Test kommt hinzu, dass kein Versuchsleiter dort ist, der die Benutzer animiert, sich weitere Aufgaben auszudenken, so dass viele Benutzer nach wenigen Versuchen aufhören. Dies zeigt sich auch an der geringeren Anzahl an Benutzerturns, die bei dem Internettest bei 17 liegt (cf. Abbildung 5.4 in Kapitel 5), während sie bei dem Test mit der Version F bei 26,19 ist (cf. Abbildung 14.1). Damit haben die Benutzer im Test mit der Version F viel länger versucht, sich Aufgaben für den Roboter auszudenken, und damit auch mehr Möglichkeiten gehabt, Fehler zu machen. Das hat zur Folge, dass im Vergleich von Internettest und simulierter gesprochensprachlicher Interaktion bei der freien Interaktion der Vorteil der Fehlerbehandlungsmechanismen weniger deutlich wird und die Turnfehlerraten in beiden Versionen in etwa gleich hoch sind. Im aufgabenbasierten Teil, in dem die individuellen Unterschiede eine weit geringere Rolle spielen, sieht man demgegenüber, dass die Turnfehlerrate für die

neue Version F wieder deutlich niedriger ist.

Ferner haben die Benutzer im Webtest nur 53,0% der Aufgaben gelöst, während es mit der Version F 83,1% waren. Außerdem benötigten die Versuchspersonen im webbasierten Test durchschnittlich 5,46 Turns pro Aufgabe, während es in der Version mit Fehlerbehandlungsmechanismen nur 3,88 Turns pro Aufgabe waren, obwohl es sich um die gleichen Aufgaben handelte. Das bedeutet, dass auch hier zu sehen ist, dass die Benutzer mit der neuen Version mit Fehlerbehandlungsmechanismen effektiver arbeiten können: Sie können mehr Aufgaben in einer kürzeren Zeit lösen und sind damit schneller und erfolgreicher als mit der Basisversion des Dialogsystems.

14.3 Übertragbarkeit auf andere Domänen

Zu guter Letzt stellt sich noch die Frage, inwieweit die hier entwickelten Fehlerbehandlungsmechanismen auch über die Roboterapplikation hinaus nützlich sind und sich auf andere Domänen und Applikationen übertragen lassen. Viele der entwickelten Mechanismen sind generisch und damit unabhängig von der betrachteten Domäne.

Geht man davon aus, dass bereits ein Basis-Dialog-System vorhanden ist, so lassen sich die Fehlerbehandlungsmechanismen leicht hinzufügen. Das dynamische Lernen neuer Konzepte mit Hilfe von Informationen aus dem Internet lässt sich auch in anderen Domänen anwenden. Dazu ist es nötig, die entsprechenden Informationen aus dem Internet in eine strukturierte Datenbank einzufügen und auf diese Konzepte dann in Grammatik und Ontologie zu referenzieren. Auch die Integration des Internets als Wissensbasis kann in anderen Domänen genutzt werden: Das QA-System Ephyra ist integriert, Regeln lassen sich aus dem Format der regulären Ausdrücke von Ephyra ins kontextfreie Grammatikformat übersetzen. Ebenso lässt sich das entwickelte Kontextmodell zur Auflösung von Ellipsen und Anaphern in anderen Domänen wiederverwenden. Da die meisten metakommunikativen Äußerungen generisch sind, können auch diese in anderen Domänen wiederverwendet werden. Applikationsspezifische Äußerungen lassen sich bei Bedarf noch hinzufügen. Auch der Finite-State-Automat kann in anderen Domänen wiederverwendet werden, um Fehler zu entdecken. Nur die Antworten, die im Fehlerfalle generiert werden, müssen an die entsprechende Applikation angepasst werden, um den Benutzer unterstützen zu können.

Damit ist der hier entwickelte Ansatz zur Fehlervermeidung und -behandlung hervorragend geeignet, um sich auf andere Domänen übertragen zu lassen, sofern ein Basis-Dialog-System vorliegt. Da sich die hier verwendeten Dialogmanagementtools TAPAS besonders zum Rapid Prototyping eignen, lässt sich dieses Basis-Dialog-System ebenfalls in kurzer Zeit für eine neue Domäne entwickeln (Denecke 2002a; Holzapfel 2005). Somit ist die Übertragbarkeit auf neue Domänen gewährleistet.

Darüberhinaus lassen sich die gleichen Fehlerbehandlungsmethoden auch in anderen Applikationen mit einer gesprochensprachlichen Eingabemöglichkeit, wie z.B. QA-Systemen oder Übersetzungssystemen, einsetzen. In dem Fall sind die in der vorliegenden Arbeit entwickelten Methoden zur Metakommunikation

und zum Kontextmanagement von entscheidender Bedeutung, da sie heutige QA-Systeme, deren Fokus auf der reinen Beantwortung von Fragen mit Hilfe von Fakten liegt, sehr gut ergänzen. Dadurch wird eine erweiterte Interaktion des Benutzers mit dem System ermöglicht: Mit Hilfe von Klärungsfragen des Systems kann sichergestellt werden, dass die Frage des Benutzer richtig verstanden wurde, und das System ist in der Lage, auf Probleme und Nachfragen des Benutzers zu reagieren. Damit ist dies ein erster Schritt in Richtung sogenannter interaktiver QA-Systeme, wie sie von einigen Forschern diskutiert werden (Webb, Strzalkowski 2006; Dang et al. 2006). Das System ist in der Lage, problematische Situationen zu erkennen und dem Benutzer interaktiv bei der Beantwortung seiner Fragen zu helfen.

14.4 Zusammenfassung

Die hier entwickelten Mechanismen zur Fehlervermeidung und -behandlung haben sich nicht nur in den Tests der einzelnen Komponenten, die in den vorhergehenden Kapiteln (cf. Kapitel 9 bis 13) erläutert wurden, sondern auch in einem umfassenden Benutzertest mit einer neuen Version, die alle Fehlerbehandlungsmethoden umfasste, als wirkungsvoll erwiesen: Die Kommunikation verlief erfolgreicher und es gab erheblich weniger falsch verstandene Benutzerturns: Bei der freien Interaktion lag die Turnfehlerrate jetzt nur bei 63,33% gegenüber 85,87% in der Baselineversion. Bei dem aufgabenbasierten Benutzertest fiel die Turnfehlerrate sogar auf 39,90% (gegenüber 87,87% in der Baselineversion). Es gelang den Benutzern, 83,1% der Aufgaben zu lösen, während sie in der Baselineversion nur 53% schafften. Dabei benötigten sie im Schnitt nur 3,88 Turns pro Aufgabe, während in der Baselineversion 6,72 benötigt wurden. Die Benutzer sind damit nicht nur erfolgreicher und schneller mit den neu entwickelten Fehlerbehandlungsmechanismen, sondern auch insgesamt zufriedener mit der Interaktion, wie sich in dem Nachinterview zeigte.

Ferner sind die entwickelten Fehlerbehandlungsmechanismen auch in anderen Domänen einsetzbar, wenn ein Basisdialogsystem vorhanden ist, auf das sie aufsetzen können. Ebenso lassen sich die Fehlerbehandlungsmethoden auch auf andere Applikationen übertragen, um generell problematische Situationen erkennen zu können und dem Benutzer beispielsweise in einem QA-System interaktiv bei der Beantwortung seiner Fragen zu helfen. Damit ist es möglich, im Rapid-Prototyping-Verfahren ein neues Dialogsystem zu entwickeln und die beschriebenen Fehlerbehandlungsmethoden anschließend zu integrieren, um eine effektive Mensch-Maschine-Kommunikation zu gewährleisten.

Kapitel 15

Fazit

15.1 Beiträge

Heutige Dialogsysteme geraten schnell an ihre Grenzen, wenn Fehler im Dialog auftreten, so dass viele Benutzer frustriert aufgeben. Um dies zu vermeiden, wurde in dieser Arbeit ein datengetriebener Ansatz zur Fehlerklassifikation gewählt. Es wurden vier verschiedene Formen von Benutzertests durchgeführt, um eine repräsentative Menge an Daten gewinnen zu können: Benutzertests über das Internet und mit dem Roboter, jeweils mit und ohne vordefinierte Aufgaben. Alle auftretenden Fehler wurden anhand ihrer Ursachen klassifiziert und Lösungsansätze für die einzelnen Fehlerklassen entwickelt. Dabei geht es sowohl um die Vermeidung von Fehlern im Vorfeld der Interaktion als auch um die Fehlerbehandlung zur Laufzeit.

Zur Fehlerbehandlung versuchen Benutzer in der Mensch-Roboter-Interaktion intuitiv, Nachfragen zu stellen und Fehler zu korrigieren, wie sie es aus der zwischenmenschlichen Kommunikation kennen. Das haben die in der vorliegenden Arbeit durchgeführten Benutzerstudien bestätigt. Diese Art der Metakommunikation, die für Menschen ein selbstverständlicher Teil jeder Kommunikation ist (Bußmann 1990), fehlt heutigen Dialogsystemen weitgehend. Daher beschäftigt sich die vorliegende Arbeit besonders mit Mechanismen zur Auflösung von Metakommunikation als eine Möglichkeit der Fehlerbehandlung zur Laufzeit. Metakommunikation umfasst nicht nur die eigentliche Korrektur, sondern auch weitergehende Nachfragen, die die Dialogpartner zur Klärung des Dialogs äußern.

Die Fehlerursachen in den Benutzertests finden sich in den folgenden drei Bereichen:

1. Neue syntaktische und semantische Konzepte
2. Elliptische und anaphorische Benutzeräußerungen
3. Komplexe Benutzeräußerungen

Die meisten Fehler treten im Bereich der neuen Konzepte auf. Erweitert man die Grammatik mit Hilfe einer Datensammlung, so nimmt die Rate an neuen Konzepten erwartungsgemäß ab. Interessanterweise liegen die meisten Fehler

innerhalb der neuen Konzepte nun im Bereich der Metakommunikation. Daher wurden in der vorliegenden Arbeit sowohl Mechanismen zur Auflösung von Metakommunikation und zur Fehlerbehandlung als auch verschiedene Methoden zur Fehlervermeidung implementiert, um Fehler, soweit als möglich, schon im Vorfeld vermeiden zu können:

Fehlerbehandlung und Metakommunikation. Metakommunikation lässt sich an den folgenden Merkmalen oder einer Kombination aus diesen Merkmalen erkennen: Hyperartikulation, Paraphrasen, Wiederholung von Äußerungen, pragmatische Korrelationen, Kontraste, Buchstabieren, Korrekturen, Nachfragen, Ausprobieren des Systems. Anhand dieser Merkmale lässt sich eine Benutzeräußerung als Metakommunikation klassifizieren. Ist dies der Fall, so gibt es für jede Klasse Methoden, um die Benutzerintention ermitteln und den Metadialog auflösen zu können. Damit wurde in dieser Arbeit ein neuartiger, generischer Mechanismus zur Auflösung von Metakommunikation erarbeitet.

Darüberhinaus wurde ein Finite-State-Automat entwickelt, um problematische Situationen zu erkennen, und darauf aufbauend eine Strategie implementiert, die dem Benutzer hilft, wenn dieser nicht weiterkommt, und ihm die Fähigkeiten des Roboters erklärt. Damit ist der Roboter in der Lage, selbst Metakommunikation anzustoßen, um problematische Situationen aufzulösen. Diese Form des proaktiven Handelns bei Problemen gekoppelt mit den Fähigkeiten, Metakommunikation aufzulösen, gibt es in bisherigen Dialogsystemen meines Wissens nicht.

Fehlervermeidung im Bereich neuer syntaktischer und semantischer Konzepte. Hier wurde eine dynamische Erweiterung des Vokabulars in den Dialogmanager integriert und dies am Beispiel der Integration einer Internet-Rezeptdatenbank evaluiert. Ferner wurde das Internet als Wissensbasis mit Hilfe eines QA-Systems integriert, um weitergehende Benutzerfragen behandeln zu können. Damit wurden zwei neue Möglichkeiten geschaffen, das bisherige Dialogsystem mit neuen Konzepten zu erweitern.

Fehlervermeidung im Bereich elliptischer und anaphorischer Benutzeräußerungen. Dafür wurde ein Kontextmodell implementiert, das zur Auflösung von Anaphern und Ellipsen dient, und es wurden neue generische Mechanismen zur Behandlung von Ellipsen und Benutzeräußerungen ohne expliziten Sprechakt entwickelt.

Fehlervermeidung im Bereich komplexer Benutzeräußerungen. Dafür wurde ein generischer Mechanismus implementiert, der es erlaubt, zusammengesetzte Nominal- und Verbalphrasen zu parsen, um auf ihre einzelnen Bestandteile zugreifen zu können.

Durch diese einzigartige Kombination aus Fehlervermeidung und Fehlerbehandlung ist eine weitaus effektivere Mensch-Maschine-Interaktion möglich, wie anhand von Benutzerstudien nachgewiesen werden konnte. Damit schafft die vorliegende Arbeit eine Basis für künftige benutzerfreundlichere Dialogsysteme.

Die durchgeführten Studien haben erste Schritte in eine Richtung aufgezeigt, die in Zukunft unbedingt weiterverfolgt werden sollte, damit Dialogsysteme in der Lage sind, Fehler effizient zu behandeln.

Daneben wurden auch die Auswirkungen der Kommunikationsstrategie sowohl des Roboters als auch des Benutzers auf die Fehler im Dialog und die Zufriedenheit des Benutzers mit dem System evaluiert. Die Ergebnisse der in der vorliegenden Arbeit durchgeführten Benutzerstudie zeigen, dass die Benutzer in der um diese Fehlerbehandlungsmechanismen erweiterten Version erfolgreicher sind: Die Turnfehlerrate ist geringer, die Rate gelöster Aufgaben steigt und die durchschnittliche Anzahl benötigter Turns pro Aufgabe ist ebenfalls geringer. Die Benutzer sind damit in der Lage in kürzerer Zeit mehr Aufgaben zu lösen als mit dem Basisdialogsystem.

15.2 Zukünftige Arbeiten

Die in dieser Arbeit durchgeführte Analyse von Fehlern in Mensch-Maschine-Dialogen hebt die Relevanz von Metakommunikation in der Mensch-Maschine-Interaktion hervor. Auf der einen Seite liegt die Ursache vieler Fehler in fehlenden Möglichkeiten zur Auflösung von Metakommunikation im Dialog. Auf der anderen Seite bieten aber auch gerade metakommunikative Maßnahmen die Möglichkeit, Fehler im Dialog schnell und effizient wieder auflösen zu können. Daher ist die hier gezeigte Fähigkeit zur Metakommunikation für beide Dialogpartner, Mensch und Maschine, für die Entwicklung zukünftiger Dialogsysteme unerlässlich.

Ferner führen die in dieser Arbeit entwickelten Mechanismen zur Auflösung von Metakommunikation und zur Fehlerbehandlung allgemein zu einer effektiveren Mensch-Roboter-Kommunikation, wie anhand von Benutzerstudien gezeigt werden konnte. Darüberhinaus sind sie domänenunabhängig und lassen sich in andere natürlichsprachliche Anwendungen, wie QA-Systeme oder maschinelle Übersetzungssysteme, integrieren. Für die Zukunft bleibt zu evaluieren, inwieweit sich aufgrund dieser intuitiveren und effizienteren Kommunikationsmöglichkeiten auch die Benutzerakzeptanz natürlichsprachlicher Anwendungen insgesamt erhöhen lässt.

Desweiteren liegen zukünftige Arbeiten in der Integration eines Lernmechanismus, der die geschilderten Fehlerbehandlungsmechanismen durch die Möglichkeit, online neue Konzepte zu lernen, ergänzt. Damit lassen sich auch zukünftig Fehler vermeiden, indem der Benutzer nun dem System neue Konzepte beibringen kann, die er gern benutzen möchte. Dazu muss das System nicht nur neue Formulierungen für bekannte Konzepte lernen, sondern auch neue semantische Konzepte richtig in die vorhandene Ontologie aus Aktionen, Eigenschaften und Objekten einordnen. Besonders für den dargestellten Haushaltsroboter ist dies von großem Interesse, da er so neue Objekte und Aufgaben in der Küche kennen lernen kann. Hier bietet sich auch eine Verknüpfung mit anderen Komponenten des Roboters, wie beispielsweise der Objekterkennung, an.

Im Bereich der Mensch-Roboter-Kommunikation ist für die Zukunft eine stärkere Integration mit anderen Komponenten des Roboters wünschenswert,

wie beispielsweise einer Kognition. Diese könnte herausfinden, was der Roboter aktuell macht und eine entsprechende Nachfrage des Benutzers beantworten. Außerdem wäre es damit möglich, Aufgaben, die sich bereits in der Ausführung befinden, nachträglich zu korrigieren. Damit könnte man über finalisierte Dialogziele hinaus auf die gerade auszuführende Aufgabe zugreifen, um diese gegebenenfalls zu ändern.

Abstract

Error Handling in Man-Machine Dialogs

1 Introduction

In recent years, spoken dialog systems became more and more popular. The awareness of the problems emerging from system errors, especially in recognizing user input and understanding the user intention, increased. These errors may lead to a consequent confusion for both users and the system itself. The need to devise better strategies for detecting problems in man-machine dialogs and dealing with them gracefully has become paramount for spoken dialog systems. As an example for spoken dialog systems, a household robot which helps users in the kitchen is chosen in the present study. Within this scenario, the error robustness is of special importance because the users are naive and want to talk to the robot in the same way as to another human. The main contribution of this thesis is a detailed analysis of errors within man-machine dialogs and the development of solutions to all the errors found there. This results in a more efficient man-machine interaction as shown in a user study.

As an example, consider the following extract of a dialog with our robot, where “U” is the user and “R” is the household robot.

U: Could you please bring me a glass of water? I am very thirsty.
R: I am sorry, but I can only carry one thing at a time. What do you want me to bring to you?
U: I would like to have some water, please.
R: I see. So you want water.
U: Dear Robbi, I am still very thirsty and I would like to drink something.
R: [silence]
U: Drink, please.
R: In which direction should I go?
U: Please, go to the water tap and bring me some water.
R: Going to the board.

In the first utterance, the robot failed to understand that the user was asking it to bring a glass of water, but assumed that the user wanted two different objects. The user still tried to get some water, but felt confused by the answers of the robot, such as “I see. So you want water.”. Once, the robot even did not answer

at all so that the user has no clue what it understood. Recognition errors further complicate the misunderstanding, and not surprisingly lead to frustration. This dialog is a classic example for a present-day man-machine interaction. The goal of this thesis is to investigate methods for better error handling to avoid such frustrating dialogs and to support the user in problematic situations.

The present thesis evaluates errors and methods to deal with them in man-machine dialogs. The second section is about related work. The third section deals with user studies which result in an error classification with three classes of errors: New concepts, elliptical and anaphoric utterances and complex utterances. Before analysing in detail these error classes, the focus switches to the user's as well as the robot's communication strategy to evaluate its influence on errors and the communication in general. Since the error classification reveals that lots of errors are due to missing concepts in the linguistic resources, the fifth section deals with their automatic extension: Novel mechanisms for dynamic extension of the vocabulary and for the integration of the internet as an additional knowledge source are developed. The sixth section is about the resolution of ellipsis and anaphora which represent the second error class. Since the third group of errors in the error classification are complex utterances, the seventh section deals with a generic method to resolve them. The eighth section explores new concepts users utter within clarification dialogs and meta communication and develops mechanisms to deal with them. The evaluation of all these error handling strategies in an overall user test shows that the users are much more successful and can accomplish more tasks in less time compared to the baseline version without any error handling. The tenth section gives a conclusion and an outlook on future work.

2 Related Work

Dialog Approaches

In the literature we can find symbolic and statistic dialog approaches (McTear 2002). Since there are not enough data for a statistic dialog management within the household domain, this thesis concentrates on symbolic approaches. The TAPAS dialog tools collection (Holzapfel 2005) based on the approaches of the dialog manager ARIADNE (Denecke 2002a) is used. It is an information-based approach in so far that the dialog states are described by the specificity of information available in discourse. At the same time, it encloses features from frame-based approaches, namely defining dialog goals which specify the services with the corresponding parameters, and is therefore more flexible than purely information-based approaches. In addition, vectorized context-free grammars and inheritance mechanisms similar to object-oriented programming are used within this dialog manager (Denecke 2000) to support rapid prototyping and a modular development of new linguistic resources. The dialog manager uses typed feature structures (Carpenter 1992) to represent semantic input and discourse information.

Errors in Man-Machine Dialogs

The problems caused by errors in spoken dialog systems are well known and can result in user frustration and task failure. Until now, most of the research dealing with errors only take speech recognition errors into account: Different researchers evaluate methods for dialog state adaptation to the language model to improve speech recognition (Xu, Rudnicky 2000; Gorrell 2003). Also different stages and language models are used to reduce word error rates and perplexity in error dialogs (Fosler-Lusier, Kuo 2001; Solsona et al. 2002). Work on hyperarticulation concludes that speakers change the way they are speaking when facing errors in principle so that the acoustic model has to be adapted (Stifelman 1993; Hirschberg et al. 2004; Soltau, Waibel 2000). Choularton et al. and also Stifelman look for general strategies on error recognition and repair to prepare the speech recognizer for the special needs of error communication (Choularton, Dale 2004; Stifelman 1993).

My concern, however, is with slightly different analyses in order cope with errors more efficiently: This thesis concentrates on semantic errors and classifies the errors according to their reasons. For every error class, methods to avoid it are developed. Furthermore, repair dialogs are evaluated in order to be able to perform efficient error handling strategies so that it will be easier for the user to correct errors which could not be avoided. Also work from human-human communication is taken into account: Schegloff et al. developed a model describing the mechanisms the dialog partners use to handle errors in human-human communication (Schegloff et al. 1977). The special case of a communication with a less competent dialog partner, such as a foreigner, is evaluated in previous studies (Marti 2001). These insights from human-human communication will be transferred to man-machine interaction in the following sections.

3 User Studies

Different Testing Conditions

A prototypical dialog system serving as a base for user tests and the ongoing development of the system is implemented. Different testing conditions were evaluated: Web-based user tests which have the advantage that lots of users all over the world can participate whenever they like to (Schmidt 1997; Reips 2002) and also multimodal user tests to see how the user can get along with the robot.

Since the user tests should be as similar as possible to the real world situation and at the same time as comfortable as possible for the users, the following conditions within every test were evaluated:

- *User tests with predefined tasks*: Every user got predefined tasks to accomplish by means of the robot. This is easier for the user, but users might be primed by the predefined tasks.
- *User tests without predefined tasks*: The users were just told that they bought a new household robot to support them. They can freely explore

	Robot	Web-based
With Tasks	22.57%	58.20%
Without Tasks	57.03%	62.40%

Table 15.1: Turn Error Rates Within Different Testing Conditions.

and interact with it. This situation is much more realistic, but at the same time much harder for the users because they have only very vague clues what kind of tasks they can ask it for.

Results & Error Classification

All the user turns which could not be transferred to the correct semantics by the dialog system are defined as *errors* so that the turn error rate gives the rate of error turns on the whole number of user turns. As expected, the turn error rate for tests with tasks is always lower than without tasks (cf. Table 15.1) because the user has less clues what the robot is capable of. Especially the testing condition with predefined tasks within the robot-based test results in much less errors. This might be due to the fact that the tasks were easier than in the web-based test and that the users could watch the robot interacting.

Nevertheless, within all the testing conditions, the same error classes can be found according to the following reasons for failure:

- **New Concepts:** new formulations, new objects, new goals, metacommunication
- **Ellipsis & Anaphora:** elliptical utterances, anaphora, missing context
- **Concatenated Utterances:** complex nominal or verbal phrases
- **Input Problems:** punctuation & digits, background noise, grammatically wrong utterances

Also the rates of the error classes are similar so that most of the errors can be found in the area of new concepts, secondmost errors are ellipsis and the fewest errors belong to the class of concatenated utterances (cf. version 1 in Table 15.2).

Effects of the Size of the Grammar

Since most of the errors were in the class of new concepts, this might be due to the small size of the prototype grammar. Therefore, about 4000 user utterances to extend the prototype grammar were collected. This extended grammar has been evaluated with the approx. 2000 utterances from the tests described before. The results revealed that with a bigger grammar the parsing rate increased up to 92.5% and the turn error rate decreased to approximately 42% for tests without tasks and 20% for tests with tasks.

There were less errors caused by new semantic concepts: The rates for all the other error classes increased, whereas the rate of this class decreased (cf.

	Version 1	Version 2
New Concepts	62.39%	49.20%
Ellipsis & Anaphora	14.81%	19.37%
Complex Utterances	3.34%	4.64%
Input Problems	19.47%	26.80%

Table 15.2: Comparison of the Rates of the Error Classes

Table 15.2). In addition, most of the errors in the area of new concepts were now caused by metacommunication, whereas the number of errors due to new formulations and to new dialog goals decreased and the rate of errors due to new objects only marginally varied. Since the manual integration of new concepts is very time- and cost-intensive, a novel mechanism for dynamic vocabulary extension is developed (See section 5). In addition, to resolve metacommunication, a new generic mechanism is implemented, as described in section 8. Regarding the other error classes, such as ellipsis and complex utterances, new mechanisms to resolve them are also developed (See section 6 resp. 7).

4 Communication Strategies

User’s Communication Strategies

A web-based experiment with two different instructions for each participant is conducted: The “child instruction” (the users were asked to talk to the robot in the same way as they would do to a little child) and the “non-child instruction” (here, the users got no detailed instruction on how to talk to the robot) (Gieselmann, Stenneken 2006). Each participant got 10 predefined tasks. During the user interaction with the system, objective success per user is measured by means of the turn error rate, the number of successfully accomplished tasks and the number of user turns necessary to accomplish a task. After the participants had finished the task set under each instruction, they filled in a short user questionnaire about their experience during the experiment.

The results showed that even though the user strategy had qualitative and quantitative effects on the communicative behavior, it was not systematically related to the communicative success in objective and subjective measures. Rather, the subjective measures of success depended predominantly on the general attitude of the user towards robots. This strengthens the need for error handling mechanisms independent of any user strategy.

Robot’s Communication Strategies

On the robot’s side, a general strategy to avoid errors is implemented: The hold strategy lets the user respeak his input by *holding up* the hypothesis of having correct discourse information. It is based on the observation that problems occur, when the user utterance is inconsistent with the current discourse which might result in dead end situations where the user gets stuck (Holzapfel, Giesel-

mann 2004). There are two different reasons why the system gets an unexpected answer: A speech recognizer error occurred or the user did not say what the system expects, but eg. started a new subdialog.

Both cases are evaluated: A recognition error is assumed and the discourse information is discarded, if the confidence for the input hypothesis is low, a better fitting input hypothesis can be found in the n-best list, or if already the last turn contains incompatible information. On the other hand, if the new input can be finalized by itself, a recognizer error seems unlikely so that it is assumed that the user wants to start a new subdialog and the old input is stored on a stack. The next input hypothesis is checked whether it belongs to the subdialog. If so, the subdialog will be continued, but the discourse state is kept to be able to return to it after the subdialog will be finished. Otherwise, the subdialog will be aborted and the discourse will be deleted. The results of a user study are promising showing that about 31% of the dialog goals can be finalized with the hold strategy compared to 15% without the strategy. Nevertheless, this strategy can only be used to support the following error handling strategies.

5 Dynamic Extension of Linguistic Resources

Dynamic Generation of New Vocabulary from the Internet

The underlying objective is to build a dialog system that can flexibly incorporate new information from dynamic information sources across the internet (Giesemann, Waibel 2006). Therefore, information from the internet is structured and stored in a database accessed by the dialog manager at runtime and the new vocabulary is automatically integrated in grammar rules as terminals. As a test case, an internet recipe database integrated in the dialog manager of our household robot is used so that it is now able to understand several thousand recipe names and ingredients.

The recipes from the internet are parsed and the information is stored in a database to avoid unnecessary online access. The data from the internet are cleaned so that duplicated recipes and recipe names with more than four words (ie., “Spaghetti Neopolitana From Dannii Minogue”) are deleted. To find a recipe by name, first the database is searched for the exact name as specified by the user. If no result can be found, a query to the database with the content words of the recipe name (ie., function words are excluded) is executed. The results are ranked according to the number of words found so that the recipe matching most of the words given in the user utterance is in the first place. To find a recipe by ingredients, a similar procedure is used so that the result is again a ranked list of at most ten recipes according to the number of ingredients specified by the user. In this way, the user gets at least one recipe which covers at least one of the ingredients he specified.

The results of a user study with predefined tasks to be accomplished by means of this enlarged version revealed that the users managed to accomplish nearly all the tasks (91.67% task completion rate) and that they had a generally positive attitude towards the robot (83.33% of the participants would use it to help them). Nevertheless, the perplexity is very high given the fact that all the

30 000 recipe names belong to the same non-terminal and can therefore be on the same place according to the grammar. This results in an average turn error rate of 71.81%. To evaluate the effect of the perplexity, recipe names are randomly deleted and the turn error rate with different vocabulary sizes ranging from 100 to 30 000 recipes is evaluated. The results show that the optimal vocabulary size is about 1 000 recipe names which results in a turn error rate of only 64.09%.

In this way, dynamic integration of new information from the internet is possible. The only prerequisites are that the data can be structured, added to the database of the dialog manager and that a corresponding concept is available in the ontology. To reduce the perplexity, the number of instantiations belonging to one semantic concept should not exceed 1 000.

Integration of the Internet as a Knowledge Source

As found in different user studies, the users suppose the robot as all-knowing and also ask questions, such as “What’s the weather like in Chicago?”. Therefore, the web-based question answering system Ephyra which is based on pattern-learning technology (Schläfer et al. 2006) is integrated within the dialog manager. There are two possibilities for this integration:

- a statistical speech recognizer to cover as much user utterances as possible
- a grammar-based speech recognizer with an extended grammar to cover the question domain and the kitchen domain

A statistical speech recognizer means different linguistic resources within dialog management and speech recognition: A general language model within speech recognition and a grammar bound to the kitchen domain within the dialog manager classifying all the utterances it cannot parse as questions for Ephyra. This has the drawback that errors in speech segmentation result in sentences which cannot be parsed by the dialog grammar, too, and will therefore be transferred to Ephyra. Furthermore, speech recognition errors from the general language model result in wrong classification of sentences available in the dialog grammar. On the other hand, a grammar-based recognizer is very inflexible and can only parse questions defined in the grammar. Therefore, a combination of both approaches consisting of a language-model based speech recognizer which contains the rolled out dialog grammar and a grammar-based dialog manager to semantically analyse the user utterances and trigger clarification questions is the ideal solution.

This is supported by the results of an evaluation by means of 40 questions from TREC 2004 (cf. Text REtrieval Conference 2004): By means of the grammar-based speech recognizer only one question can be correctly recognized and by means of the statistical speech recognizer none. Nevertheless, the correct semantics can be assigned to about 40% of the sentences by means of the grammar so that a combination of a statistical speech recognizer and a grammar classifying the user utterances according to their semantic properties is very promising. In addition, the word error rate of a statistical speech recognizer is lower (59.71% compared to 82.78% for grammar-based speech recognition) given the fact that the cost for changing the path are higher in a grammar.

6 Resolution of Anaphora and Ellipsis

This thesis concentrates on the resolution of personal and deictic pronouns because of their common use in multimodal human-robot interaction. To resolve them a context model is used which contains information on the environment: All the available objects with their three-dimensional position in the room. In addition, possible antecedents are stored in the context model in a list similar to Strube's S-list (Strube 1998). The context model is implemented in a way similar to the human brain that it "forgets" old antecedents after a certain period of time (Clark 1978). Whenever a new user utterance comes in, the context model is updated with the possible antecedents.

For resolving deictic pronouns, a gesture recognizer and multimodal parsing of speech and gestures is used so that the information from both input modalities is merged on a semantic base by means of time stamps (Gieselmann, Denecke 2003). For resolving personal pronouns, the list of possible antecedents in the context model is evaluated whether there is an appropriate antecedent which agrees in its syntactic and semantic features with the pronoun.

To resolve elliptical utterances, their parsability is improved by implementing new grammar rules for short utterances found in the user tests. Also generic moves are implemented to trigger clarification questions even if the user only utters a single word without any speech act. Since user utterances without speech acts result in a deselected dialog state which leads to clearing the discourse, a new mechanism is implemented to store the objects available in a deselected dialog state also in the context model. Whenever the following user utterances contain personal pronouns, they can be resolved by using these objects as antecedents. New objects are added at the top of the stack, when the user enters again the deselected dialog state.

The results of a user study with the household robot show that all the personal pronouns can be resolved by this mechanism. Also elliptical utterances can be resolved successfully, but these short utterances are very likely to be misrecognized by the speech recognizer. Therefore, these short utterances also trigger clarification dialogs from the robot, as explained in chapter 8.

7 Resolution of Complex Utterances

In the user studies, there are two kinds of complex utterances:

1. Concatenation of two or more nominal phrases, such as "Bring me apples and tomatoes."
2. Concatenation of two or more verbal phrases, such as "Get me coffee and put milk in it.", "Hi Robbi. I would like some coke."

The first case is much more common and can be found in about two third of all concatenated utterances. The second case occurs especially when the speech segmentation does not recognize the end of a turn.

For the resolution of concatenated nominal phrases, a new class of speech acts is developed so that any number of objects can be parsed: The first object

is parsed as a single object and the other objects in a list of objects. All the verb classes in the application which need this property, such as bring or get something, inherit from this class. Since within the user tests, there were only concatenations of no more than two verbal phrases, a new generic dialog goal is created which consists of two speech acts to resolve concatenated verbal phrases. By means of these novel mechanisms, all concatenated phrases can be resolved.

8 Clarification Dialogs and Metacommunication

Comparing human-human and human-robot dialogs as far as the strategies for error avoidance and recovery are concerned, it becomes obvious that, although humans use more explicit strategies to indicate errors, the strategies for error correction and their use in the communication are very similar. Metacommunication is used to correct preceding utterances and to resolve errors within human-human communication. To deal with metacommunication also in human-robot dialogs, the user utterances in the user studies belonging to this category are evaluated and grouped according to the user intention:

- *Clarification Questions* from the user: The user wants to know eg., whether the robot understands him, what the robot is doing, etc.
- *Repair* of a user utterance: The user corrects the preceding utterance of the robot.
- *Test of the Robot*: The user tests the capabilities of the robot by giving instructions for tasks the robot can probably not accomplish; also insults are in this category.

Most of the metacommunication can be resolved by generic methods. Only very few application-specific adaptations are necessary. In this way, clarification questions and user utterances testing the robot can be implemented so that they do not take the semantics of the user utterance into account, but just classify the user utterance as belonging to this category. Corrections are the most challenging metacommunication for the dialog manager, since the information available in discourse has to be changed. In addition, the information to be deleted is sometimes not given or only implicitly given by the new information incompatible with the old one. The correction can either concern a whole dialog goal or only an object or a property within a dialog goal. It is also implemented generically: If the user says “No, I meant X and not Y.”, the objects X and Y are from the robot application and can be accessed via inheritance mechanisms.

In addition, the robot can itself initiate metacommunication to resolve possible problems: A new dialog strategy is developed which is sensitive to different types of miscommunications. A collection of problem indicators is used to drive state transitions in a 4-state indicator of start vs. normal vs. error correction vs. help modes of operation. The generation strategy is modified according to the type of problem. The results of a user study show that the task success rate, the concept accuracy and the user satisfaction can be improved substantially.

	Baseline	Version F
Without Tasks		
Turn Error Rate	77.86%	63.33%
Parsing Rate	85.87%	92.34%
With Tasks		
Turn Error Rate	61.97%	39.90%
Parsing Rate	87.87%	96.30%
Task Completion Rate	65.0%	83.10%
Turns per Task	6.72	3.88

Table 15.3: User Study Comparing the Baseline Version (without any Error Handling) with Version F (including all Error Handling Mechanisms Developed in the Present Thesis)

9 Evaluation

All the error handling mechanisms described above are implemented in one version called version F and evaluated within a user study with 16 users. Version F is compared to the baseline system without any error handling mechanisms. The results revealed that the users can get along with the robot a lot easier when provided with these extra error handling facilities (see Table 15.3): The turn error rate decreases significantly and the parsing rate increases.

As expected the turn error rate is generally lower for the condition with predefined tasks, since the users have more clues what to say. Still we find a significantly lower turn error rate and a higher parsing rate compared to the baseline version. In addition, the task completion rate is higher within version F. The users even needed fewer turns to accomplish the tasks, as shown by the turns per task ratio: They were more successful with version F while accomplishing more tasks in less time. In addition, they are more content with the interaction and are more likely to use the robot again than the users in the baseline version. This strengthens the need for the error handling mechanisms developed within this thesis.

10 Conclusion & Outlook

A date-driven method is used to evaluate the reasons for errors in human-robot communication and the following strategies to deal with them are implemented:

- dynamic extension of linguistic resources
- metacommunication
- anaphora resolution
- handling complex as well as elliptical utterances

A user study shows that the combination of these strategies results in a much more successful interaction: The turn error rate can be reduced by a relative improvement of 18.66% for the test without task and of 35.61% for the task-based test. Remaining failures of task completion are mostly due to speech recognition errors and out of domain utterances. To conclude, users can accomplish more turns in less time and are more content with a dialog system including the new error handling mechanisms.

These error handling mechanisms can also be applied to other domains, when a base dialog system is available. In addition, they can be transferred to other natural language applications to detect problematic situations and resolve them. For example, they can be integrated into a QA system to interactively help the user in asking questions. In this way, a more efficient man-machine dialog can be guaranteed. Since the mechanisms developed within this thesis are generic, they can be also used with different languages. At the moment, they are implemented for English, but can be ported to other languages using the same semantic concepts.

Future work might include the integration into a more complex cognitive architecture within the robot which enables online learning and a kind of introspection so that the dialog manager has information about the internal state of the robot. In addition, a learning mechanism might be integrated to complement the described error handling mechanisms with a possibility to interactively learn new concepts and words. And the integration in other natural language processing applications can be evaluated.

Literaturverzeichnis

op den Akker et al. 2005

AKKER, R. op d. ; BUNT, H. ; KEIZER, S. ; SCHOOTEN, B. van: From Question Answering to Spoken Dialogue: Towards an Information Search Assistant for Interactive Multimodal Information Extraction. In: *Proceedings of the Interspeech 05* (2005), S. 2793–2796

Allen et al. 2000a

ALLEN, J. ; BYRON, D. ; DZIKOVSKA, M. ; FERGUSON, G. ; GALESCU, L. ; STENT, A.: An architecture for a generic dialogue shell. In: *Natural Language Engineering* 6 (2000), Nr. 3-4, S. 213–228

Allen et al. 2000b

ALLEN, J. ; FERGUSON, G. ; MILLER, B. W. ; RINGGER, E. K. ; ZOLLO, Teresa S.: Dialogue systems: From theory to practice in TRAINS-96. In: *R. Dale, H. Moisl, and H. Somers, eds.: Handbook of Natural Language Processing* (2000), S. 347–376

Allen et al. 1995

ALLEN, J. ; SCHUBERT, L. ; FERGUSON, G. ; HWANG, C. ; KATO, T. ; LIGHT, M. ; MILLER, B. ; POESIO, M. ; TRAUM, D.: The TRAINS project: A case study in building a conversational planning agent. In: *Journal of Experimental and Theoretical Artificial Intelligence* 7 (1995), S. 7–48

Ando et al. 1994

ANDO, H. ; KITAHARA, Y. ; HATAOKA, N.: Evaluation of multimodal interface using spoken language and pointing gesture on interior design system. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'94)* 2 (1994), S. 567–570

Asfour et al. 2006

ASFUR, T. ; REGENSTEIN, K. ; AZAD, P. ; SCHRÖDER, J. ; BIERBAUM, A. ; VAHRENKAMP, N. ; DILLMANN, R.: ARMAR-III: An integrated humanoid platform for sensory-motor control. In: *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (2006)

Atkinson, Heritage 1984

ATKINSON, J. M. ; HERITAGE, J.: *Structures of Social Action. Studies in Conversation Analysis*. Cambridge UK: Cambridge University Press, 1984

Aust et al. 1995

AUST, H. ; ÖRDER, M. ; SEIDE, F. ; STEINBISS, V.: The Phillips automatic train timetable information system. In: *Speech Communication* 17 (1995), S. 249–262

Austin 1962

AUSTIN, J.L.: *How to do Things with Words*. Oxford University Press, Oxford, UK, 1962

Baldwin 1995

BALDWIN, B.: *CogNIAC: A High Precision Pronoun Resolution Engine*, University of Pennsylvania Department of Computer and Information Sciences, Diss., 1995

Banko et al. 2002

BANKO, M. ; BRILL, E. ; DUMAIS, S. ; LIN, J.: AskMSR: Question answering using the World Wide Web. In: *Proceedings of 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases* (2002)

Batliner et al. 2004

BATLINER, A. ; HACKER, C. ; STEIDL, S. ; NÖTH, E. ; D'ARCY, S. ; RUSSELL, M. ; WONG, M.: "You stupid tin box" - children interacting with the AIBO robot: A cross-linguistic emotional speech corpus. . In: *Proceedings of the 4th International Conference of Language Resources and Evaluation LREC 2004*, 2004, S. 171—174

Behringer et al. 2002

BEHRINGER, N. ; KARTAL, U. ; LOUKA, K. ; SCHIEL, F. ; TÜRK, I.: A Procedure for Multimodal Interactive System Evaluation. In: *Proceedings of the LREC Workshop on Multimodal Resources and Multimodal Systems Evaluation* (2002), S. 77–80

Bernsen et al. 1996

BERNSEN, N.O. ; DYBKJÆR, L. ; DYBKJÆR, H.: Cooperativity in human-machine and human-human spoken dialogue. In: *Discourse Processes* (1996), S. 213–236

Black et al. 1998

BLACK, A. ; LENZO, K. ; PAGET, V.: Issues in Building General Letter To Sound Rules. In: *Proceedings of the 3rd ESCA/COCOSDA Workshop on Speech Synthesis* (1998), S. 77–80

Boden, Zimmerman 1991

BODEN, D. ; ZIMMERMAN, D.H.: *Talk and Social Structure*. Polity Press, Cambridge and University of California Press, 1991

Bohus, Rudnicky 2002

BOHUS, D. ; RUDNICKY, A.: Integrating multiple language sources for utterance-level confidence annotation in the CMU communicator spoken dialogue system. In: *Technical Report CS-190, CMU* (2002)

Bohus, Rudnicky 2005

BOHUS, D. ; RUDNICKY, A.: Constructing Accurate Beliefs in Spoken Dialog Systems. In: *Proceedings of the ASRU 05* (2005), S. 272–277

Bolt 1980

BOLT, R.A.: Put that there: Voice and Gesture at the Graphics Interface. In: *Computer Graphics* 14 (1980), Nr. 3, S. 262–270

den Bosch et al. 2001

BOSCH, A. V. ; KRAMER, E. ; SWERTS, M.: Detecting problematic turns in human-machine interactions: Rule-induction versus memory-based learning approaches. In: *Proceedings of the Association for Computational Linguistics (ACL'01)* (2001), S. 499–506

Bousquet-Vernettes et al. 2003

BOUSQUET-VERNETTES, C. ; PRIVAT, R. ; VIGOUROUX, N.: Error Handling in Spoken Dialog Systems: Towards Corrective Dialogue. In: *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems* (2003), S. 41–45

Bousquet-Vernettes, Vigouroux 2003

BOUSQUET-VERNETTES, C. ; VIGOUROUX, N.: Recognition Error Handling by the Speech Understanding System to Improve Spoken Dialog Systems. In: *Proceedings of the ISCA workshop on Error Handling in Spoken Dialogue Systems* (2003), S. 113–118

Bremer 1997

BREMER, K.: *Verständigungsarbeit: Problembearbeitung und Gesprächsverlauf zwischen Sprechern verschiedener Muttersprachen*. Tübingen: Niemeyer, 1997

Brennan et al. 1987

BRENNAN, E. ; FRIEDMAN, Marilyn W. ; POLLARD, Carl J.: A Centering Approach to Pronouns. In: *Proceedings of the 25th Annual Meeting of the Association of Computational Linguistics (ACL '87)* (1987), S. 155–162

Brill et al. 2002

BRILL, E. ; DUMAIS, S. ; BANKO, M.: An analysis of the AskMSR question-answering system. In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (2002)

Bringert et al. 2005

BRINGERT, B. ; LJUNGLÖF, P. ; RANTA, A. ; COOPER, R.: Multimodal Dialogue System Grammars. In: *Proceedings of the Ninth Workshop on the Semantics and Pragmatics of Dialogue (DIALOR 05)* (2005), S. 53–60

Bulyko et al. 2005

BULYKO, I. ; KIRCHHOFF, K. ; OSTENDORF, M. ; GOLDBERG, J.: Error Correction Detection and Response Generation in a Spoken Language Dialogue System. In: *Speech Communication* 45 (2005), S. 271–288

Bußmann 1990

BUSSMANN, H.: *Lexikon der Sprachwissenschaft*. Alfred Kröner Verlag Stuttgart, 1990

Cahn, Brennan 1999

CAHN, J. E. ; BRENNAN, S. E.: A psychological model of grounding and repair in dialog. In: *Proceedings of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems* (1999), S. 25–33

Carbonell 1979

CARBONELL, J. G.: Towards a self-extending parser. In: *Proceedings of the ACL-79* (1979)

Carpenter 1992

CARPENTER, B.: *The Logic of Typed Feature Structures*. Cambridge University Press, 1992

Carpenter et al. 2001

CARPENTER, P. ; JIN, C. ; ZHANG, D. ; BOHUS, D. ; RUDNICKY, A.: Is this Conversation on Track? In: *Proceedings of the Eurospeech* (2001)

Chotimongkol 2001

CHOTIMONGKOL, A.: *Improving Speech Recognition Performance in a Dialogue System Using N-best Hypotheses Reranking*, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Diplomarbeit, 2001

Choularton, Dale 2004

CHOULARTON, S. ; DALE, R.: User Responses to Speech Recognition Errors: Consistency of Behaviour across Domains. In: *Proceedings of the Tenth Australian International Conference on Speech Science and Technology* (2004)

Chung et al. 2004

CHUNG, G. ; SENEFF, S. ; WANG, C. ; HETHERINGTON, L.: A Dynamic Vocabulary Spoken Dialogue Interface. In: *Proceedings of the Interspeech 04* (2004), S. 327–330

Clark 2001

CLARK, C.: *2001: Odyssee im Weltraum. Der Roman zum Film*. München : Heyne, 2001

Clark 1978

CLARK, H. H.: On Inferring what is Meant. In: *W. J. M. Levelt and G. B. Flores d'Arcais (Eds.). Studies in the Perception of Language* (1978), S. 295–322

Clark 1999

CLARK, H. H.: How do real people communicate with virtual partners? In: *Proceedings of 1999 AAAI Fall Symposium, Psychological Models of Communication in Collaborative Systems* (1999)

Clark, Schäfer 1989

CLARK, H. H. ; SCHÄFER, E. F.: Contributing to discourse. In: *Cognitive Science* 13 (1989), S. 259–294

Cohen et al. 1997

COHEN, P. ; JOHNSTON, M. ; MCGEE, D. ; OVIATT, S. ; PITTMAN, J. ; SMITH, I. ; CHEN, L. ; CLOW, J.: QuickSet: multimodal interaction for distributed applications. In: *Proceedings of the 5th ACM international conference on Multimedia* (1997), S. 31–40

Cooper 1997

COOPER, W. S.: On Selecting a Measure of Retrieval Effectiveness. In: *Jones, K. S. and Willett, P., (ed.): Readings in Information Retrieval* (1997), S. 191 – 204

Corradini et al. 2002

CORRADINI, A. ; WESSON, R. ; COHEN, P.: A Map-based System Using Speech and 3D Gestures for Pervasive Computing. In: *Proceedings of the IEEE International Conference on Multimodal Interfaces (ICMI'02)* (2002), S. 191–196

Crain, Steedman 1985

CRAIN, S. ; STEEDMAN, M. S.: On not being led up to the garden path: The use of context of the psychological syntax parser. In: *D. Dowty and L. Karttunen and A. Zwicky (Eds.). Natural Language Parsing* (1985), S. 320–358

Dang et al. 2006

DANG, H. T. ; LIN, J. ; KELLY, D.: Overview of the TREC 2006 Question Answering Track. In: *Proceedings of the The Fifteenth Text REtrieval Conference (TREC 2006)* (2006)

Denecke 2000

DENECKE, M.: Object-oriented Techniques in Grammar and Ontology Specification. In: *Proceedings of the Workshop on Multilingual Speech Communication* (2000), S. 59–64

Denecke 2002a

DENECKE, M.: *Generische Interaktionsmuster für aufgabenorientierte Dialogsysteme*, Interactive Systems Lab, Faculty of Computer Science, University of Karlsruhe, Diss., 2002

Denecke 2002b

DENECKE, M.: Rapid Prototyping for Spoken Dialogue Systems. In: *Proceedings of the 19th International Conference on Computational Linguistics* (2002), S. 1–7

Denecke, Yang 2000

DENECKE, M. ; YANG, J.: Partial Information in Multimodal Dialogue. In: *Proceedings of the IEEE International Conference on Multimodal Interfaces (ICMI'00)* (2000), S. 624–633

Dybkjær, Bernsen 2000

DYBKJÆR, L. ; BERNSEN, N.O.: Usability issues in spoken language dialogue systems. In: *Kuppevelt, J. v., Heid, U. and Kamp, H. (Eds.): Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering, Natural Language Engineering 6* (2000), S. 243–272

Dybkjær et al. 1996

DYBKJÆR, L. ; BERNSEN, N.O. ; DYBKJÆR, H.: Reducing miscommunication in spoken human-machine dialogue. In: *Proceedings of the AAAI Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication, 13th National Conference on Artificial Intelligence* (1996), S. 29–36

Faerch, Kasper 1983

FAERCH, C. ; KASPER, G.: Plans and strategies in foreign language communication. In: *Faerch, C. and Kasper, G. (eds.) Strategies in Interlanguage Communication* (1983), S. 20–60

Filisko, Seneff 2003

FILISKO, E. ; SENEFF, S.: A Context Resolution Server for the Galaxy Conversational Systems. In: *Proceedings of the Eurospeech* (2003)

Filisko, Seneff 2004

FILISKO, E. ; SENEFF, S.: Error detection and recovery in spoken dialogue systems. In: *Proceedings of the HLT-NAACL 2004 Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing* (2004)

Fink et al. 2004

FINK, G. A. ; FRITSCH, J. ; HOHENNER, S. ; KLEINEHAGENBROCK, M. ; LANG, S. ; SAGERER, G.: Towards Multi-Modal Interaction with a Mobile Robot. In: *Pattern Recognition and Image Analysis 14* (2004), Nr. 2, S. 173–184

Finke et al. 1997

FINKE, M. ; GEUTNER, P. ; HILD, H. ; KEMP, T. ; RIES, K. ; WESTPHAL, M.: The Karlsruhe-Verbmobil Speech Recognition Engine. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP-97* (1997), S. 83–86

Fischer 2006

FISCHER, K.: *What Computer Talk is and isn't. Human-Computer Conversation as Intercultural Communication..* Bd. 17. AQ-Verlag, 2006

Fosler-Lusier, Kuo 2001

FOSLER-LUSIER, E. ; KUO, H.K. J.: Using Semantic Information for Rapid Development of Language Models within ASR Dialogue Systems. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'01)* (2001), S. 553–556

Foster 2004

FOSTER, M.E.: Generating output in the COMIC multimodal dialogue system. In: *Talk presented at W3C Workshop on Multimodal Interaction* (2004)

Fügen et al. 2004

FÜGEN, C. ; HOLZAPFEL, H. ; WAIBEL, A.: Tight Coupling of Speech Recognition and Dialog Management - Dialog-Context Grammar Weighting for Speech Recognition. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2004)* (2004)

Gabsdil 2003

GABSDIL, M.: Clarification in Spoken Dialogue Systems. In: *Proceedings of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue* (2003), S. 28–35

Gavaldà 2000a

GAVALDÀ, M.: Epiphenomenal Grammar Acquisition with GSG. In: *Proceedings of the Workshop on Conversational Systems of the 6th Conference on Applied Natural Language Processing and the 1st Conference of the North American Chapter of the Association for Computational Linguistics (ANLP/NAACL-2000)* (2000)

Gavaldà 2000b

GAVALDÀ, M.: *Growing Semantic Grammars*, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Diss., 2000

Gieselmann, Denecke 2003

GIESELMANN, P. ; DENECKE, M.: Towards Multimodal Interaction with an Intelligent Room. In: *Proceedings of the Eurospeech* (2003), S. 2229–2232

Gieselmann et al. 2003

GIESELMANN, P. ; FÜGEN, C. ; HOLZAPFEL, H. ; SCHAAF, T. ; WAIBEL, A.: Towards Multimodal Communication with a Household Robot. In: *Proceedings of the Third IEEE International Conference on Humanoid Robots (Humanoids)* (2003)

Gieselmann, Stenneken 2006

GIESELMANN, P. ; STENNEKEN, P.: Communication with Robots: Evidence from a Web-based Experiment on Human-Computer Interaction. In: *Proceedings of the IEEE/ACL Workshop on Spoken Language Technology* (2006)

Gieselmann, Waibel 2006

GIESELMANN, P. ; WAIBEL, A.: Dynamic Extension of a Grammar-based Dialogue System: Constructing an All-Recipes Knowing Robot. In: *Proceedings of the Interspeech 06* (2006)

Goddeau et al. 1996

GODDEAU, D. ; MENG, H. ; J. POLIFRONE, S. S. ; BUSAYAPONGCHAI, S.: A

form-based dialogue manager for spoken language applications. In: *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP '96)* (1996), S. 701–704

Goldberg et al. 2003

GOLDBERG, J. ; OSTENDORF, M. ; KIRCHHOFF, K.: The Impact of Response Wording on Error Correction Subdialogues. In: *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems* (2003)

Gorin et al. 2002

GORIN, A. L. ; ABELLA, A. ; ALONSO, T. ; RICCARDI, G. ; WRIGHT, J. H.: Automated Natural Spoken Dialog. In: *IEEE Computer Magazine* 35 (2002), Nr. 4, S. 51–56

Gorrell 2003

GORRELL, G.: Recognition Error Handling in Spoken Dialogue Systems. In: *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia* (2003)

Gorrell et al. 2002

GORRELL, G. ; LEWIN, I. ; RAINER, M.: Adding intelligent help help to a mixed initiative spoken dialogue system. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)* (2002)

Graff 2002

GRAFF, D.: An overview of Broadcast News Corpora. In: *Speech Communication* 37 (2002), S. 15–26

Green et al. 1963

GREEN, B. ; WOLF, A. ; CHOMSKY, C. ; LAUGHERY, K.: Baseball: an automatic question answerer. In: *Feigenbaum, E. and Feldman, J. (eds.), Computers and Thought* (1963), S. 545 – 549

Grice 1975

GRICE, H. P.: Logic and conversation. In: *Syntax and Semantics* (1975), S. 41–58

Grosz et al. 1995

GROSZ, B. J. ; JOSHI, A. K. ; WEINSTEIN, S.: Centering: A Framework for Modeling the Local Coherence of Discourse. In: *Computational Linguistics* 21 (1995), Nr. 2, S. 203–226

Grosz, Sidner 1986

GROSZ, B. J. ; SIDNER, C. L.: Attention, Intentions, and the Structure of Discourse. In: *Computational Linguistics* 12 (1986), Nr. 3, S. 175–204

Gurevych, Porzel 2003

GUREVYCH, I. ; PORZEL, R.: Using knowledge-based scores for identifying best speech recognition hypothesis. In: *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems* (2003)

Gustafson et al. 2000

GUSTAFSON, J. ; BELL, L. ; BESKOW, J. ; BOYE, J. ; CARLSON, R. ; EDLUND, J. ; GRANSTRÖM, B. ; HOUSE, D. ; WIRÉN, M.: Adapt - a multimodal conversational dialogue system in an apartment domain. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'00)* (2000), S. 134–137

Haasch et al. 2004

HAASCH, A. ; HOHENNER, S. ; HÜWEL, S. ; KLEINEHAGENBROCK, M. ; LANG, S. ; TOPTISIS, I. ; FINK, G. A. ; FRITSCH, J. ; WREDE, B. ; SAGERER, G.: BIRON – The Bielefeld Robot Companion. In: PRASSLER, E. (Hrsg.) ; LAWITZKY, G. (Hrsg.) ; FIORINI, P. (Hrsg.) ; HÄGELE, M. (Hrsg.): *Proc. Int. Workshop on Advances in Service Robotics*. Stuttgart, Germany : Fraunhofer IRB Verlag, May 2004, S. 27–32

Hazen et al. 2002

HAZEN, T.-J. ; BURIANEK, T. ; POLIFRONI, J. ; SENEFF, S.: Recognition Confidence Scoring for Use in Speech Understanding Systems. In: *Computer Speech and Language* 16 (2002), Nr. 1, S. 49–67

Hirschberg et al. 2004

HIRSCHBERG, J. ; LITMAN, D. ; SWERTS, M.: Prosodic and Other Cues to Speech Recognition Failures. In: *Speech Communication* 43 (2004), S. 155–175

Hirschmann et al. 1990

HIRSCHMANN, L. ; A.DAHL, D. ; MCKAY, D. P. ; NORTON, L. M. ; LINEBARGER, M. C.: Beyond Class A: A Proposal for automatic Evaluation of Discourse. In: *Proceedings of the Speech and Natural Language Workshop* (1990), S. 109–113

Hobbs 1977

HOBBS, J. R.: Resolving Pronoun References. In: *Lingua* 44 (1977), S. 311–338

Holzappel 2005

HOLZAPFEL, H.: Towards Development of Multilingual Spoken Dialogue Systems. In: *Proceedings of the 2nd Language and Technology Conference* (2005)

Holzappel, Gieselmann 2004

HOLZAPFEL, H. ; GIESELMANN, P.: A Way Out of Dead End Situations in Dialogue Systems for Human-Robot Interaction. In: *Proceedings of the Third IEEE International Conference on Humanoid Robots (Humanoids)* (2004)

Hori et al. 2003

HORI, Ch. ; HORI, T. ; TSUKADA, H. ; ISOZAKI, H. ; SASAKI, Y. ; MAEDA, E.: Spoken Interactive ODQA System: SPIQA. In: *Proceedings of the Association of Computational Linguistics* (2003)

Horvitz, Paek 2001

HORVITZ, E. ; PAEK, T.: Harnessing Models of Users' Goals to Mediate Clarification Dialog in Spoken Language Systems. In: *Proceedings of the Eighth International Conference on User Modeling* (2001), S. 3–13

Johnston 1998

JOHNSTON, M.: Unification-based Multimodal Parsing. In: *Proceedings of the International Joint Conference of the ACL and the International Committee on Computational Linguistics* (1998), S. 624–630

Johnston et al. 1997

JOHNSTON, M. ; COHEN, P. ; D. MCGEE, D. ; OVIATT, S. ; PITTMAN, J. ; SMITH, I.: Unification-based Multimodal Integration. In: *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the European Chapter of the ACL* (1997), S. 281–288

Kemp, Schaaf 1997

KEMP, Th. ; SCHAAF, Th.: Estimating confidence using word lattices. In: *Proceedings of Eurospeech 97* (1997)

Kiyono, Tsujii 1993

KIYONO, M. ; TSUJII: Linguistic knowledge acquisition from parsing failures. In: *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics* (1993), S. 222–231

Komatani, Kawahara 2000

KOMATANI, K. ; KAWAHARA, T.: Generating effective confirmation and guidance using two-level confidence measures for dialogue systems. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'00)* (2000)

Krahmer et al. 2001

KRAHMER, E. ; SWERTS, M. ; THEUNE, M. ; WEEGELS, M.: Error detection in spoken human-machine interaction. In: *International Journal of Speech Technology* 4 (2001), Nr. 1, S. 19–29

Krause, Hitzenberger 1992

KRAUSE, J. ; HITZENBERGER, L.: *Computer Talk*. Georg Olms Verlag, 1992

Kumar et al. 2003

KUMAR, A. ; SALMON-ALT, S. ; ROMARY, L.: Reference Resolution as a Facilitating Process towards Robust Multimodal Dialogue Management: A Cognitive Grammar Approach. In: *International Symposium on Reference Resolution and Its Application to Question Answering and Summarization* (2003)

Landragin 2002

LANDRAGIN, F.: The Role of Gesture in Multimodale Referring Actions. In: *Proceedings of the 4th International Conference on Multimodal Interfaces* (2002)

Landragin, Romary 2003

LANDRAGIN, F. ; ROMARY, L.: Referring to Objects Through Sub-Contexts in Multimodal Human-Computer Interaction. In: *Seventh Workshop on the Semantics and Pragmatics of Dialogue (DiaBruck'03)* (2003), S. 67–74

Lemon 2004

LEMON, O.: Context-sensitive Speech Recognition in ISU-Dialogue Systems: Results for the Grammar-switching Approach. In: *J. Ginzburg and E. Vallduví (eds.), Catalog '04. Proceedings of the Eighth Workshop on the Semantics and Pragmatics of Dialogue* (2004), S. 49–55

Levenstein 1996

LEVENSTEIN, V. I.: Binary codes capable of correcting insertion and reversals. In: *Cybernetics and Control Theory 10* (1996), S. 707–710

Levin et al. 2000

LEVIN, E. ; PIERACCINI, R. ; ECKERT, W.: A stochastic model of human-machine interaction for learning dialogue strategies. In: *IEEE Transactions on Speech and Audio Processing* 8 (2000), Nr. 1, S. 1883–1886

Levow 1998

LEVOW, G.-A.: Characterizing and Recognizing Spoken Corrections in Human-Computer Dialogue. In: *Proceedings of COLING 1998* (1998), S. 736–742

Lin, Katz 2002

LIN, J. ; KATZ, B.: Extracting Answers from the Web Using Knowledge Annotation and Knowledge Mining Techniques. In: *Proceedings of the Eleventh Text REtrieval Conference* (2002)

Liu et al. 2005

LIU, P. ; TIAN, Y. ; ZHOU, J. ; SOONG, F.: Background Model Based Posterior Probability for Measuring Confidence. In: *Proceedings of the Interspeech 05* (2005)

Mangu et al. 2000

MANGU, L. ; BRILL, E. ; STOLCKE, A.: Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. In: *Computer, Speech and Language* 14 (2000), S. 373–400

Marti 2001

MARTI, J.: *Verstehensschwierigkeiten unter Nicht-MuttersprachlerInnen. Eine empirische Studie anhand von Asylsuchenden in Bern*, Lizentiatsarbeit, Universität Bern, Diss., 2001

McNeill, Duncan 2000

MCNEILL, D. ; DUNCAN, S.: Growth points in thinking-for-speaking. In: *Language and Gesture* (2000)

McTear et al. 2003

McTEAR, M. ; O'NEILL, I. ; HANNA, P. ; LIU, X.: Handling errors and determining confirmation strategies - an object-oriented approach. In: *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems* (2003), S. 129–132

McTear 2002

McTEAR, M. F.: Spoken Dialogue Technology: Enabling the Conversational Interface. In: *ACM Computing Surveys* 34 (2002), Nr. 1, S. 90–169

Miller, Perlis 1993

MILLER, M. ; PERLIS, D.: Presentations and this and that: logic in action. In: *Proceedings of the 15th Annual Conference of the Cognitive Science Society* (1993)

Moore et al. 1995

MOORE, R. ; APPELT, D. ; DOWDING, J. ; MORAN, M.: Combining Linguistic and Statistical Knowledge Sources in Natural Language Processing for ATIS. In: *ARPA Spoken Language Technology Workshop* (1995)

Nakagawa, Zhang 1994

NAKAGAWA, S. ; ZHANG, J.X.: An input interface with speech and touch screen. In: *Transactions of the Institute of Electronic Engineers Japan C* 114-C(10) (1994), S. 1009–1017

Nickel, Stiefelhagen 2003

NICKEL, K. ; STIEFELHAGEN, R.: Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In: *Proceedings of the 5th international conference on Multimodal interfaces (ICMI'03)* (2003), S. 140–146

Okuno et al. 2002

OKUNO, H. G. ; NAKADAI, K. ; KITANO, H.: Social Interaction of Humanoid Robot Based on Audio-Visual Tracking. In: *IEA/AIE '02: Proceedings of the 15th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Springer-Verlag, 2002. – ISBN 3-540-43781-9, S. 725–735

OpenNLP 2005

OPENNLP: <http://opennlp.sourceforge.net/index.html>. 2005

Oviatt 1998

OVIATT, S.: The cham model of hyperarticulate adaptation during human-computer error resolution. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)* (1998)

Oviatt 1999a

OVIATT, S.: Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. In: *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)* (1999), S. 576–583

Oviatt 1999b

OVIATT, S.: Ten Myths of Multimodal Interaction. In: *Communications of the ACM* 42 (1999), Nr. 11, S. 74–81

Paek, Horvitz 2000

PAEK, T. ; HORVITZ, E.: Conversation as action under uncertainty. In: *Proceedings of the 16th Conference on Action under Uncertainty and Artificial Intelligence* (2000), S. 455–464

Perlis et al. 1998

PERLIS, D. ; PURANG, K. ; ANDERSEN, C.: Conversational adequacy: Mistakes are the essence. In: *International Journal of Human-Computer Studies* 48 (1998), Nr. 5, S. 553–575

Poesio, Traum 1998

POESIO, M. ; TRAUM, D.: Towards an Axiomatization of Dialogue Acts. In: *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology)* (1998), S. 207–222

Porzel, Gurevych 2002

PORZEL, R. ; GUREVYCH, I.: Towards Context-adaptive Utterance Interpretation. In: *Proceedings of the 3rd SIGDial Workshop on Discourse and Dialogue* (2002), S. 154 – 161

Purang 2001

PURANG, K.: *Systems that Detect and Repair their Own Mistakes*, University of Maryland, Diss., 2001

Quek et al. 2002

QUEK, F. ; MCNEILL, D. ; BRYLL, B. ; DUNCAN, S. ; MA, X. ; KIRBAS, C. ; MCCULLOUGH, K.E. ; ANSARI, R.: Multimodal Human Discourse: Gesture and Speech. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 9 (2002), Nr. 3, S. 171 – 193

Raux, Eskenazi 2004

RAUX, A. ; ESKENAZI, M.: Non-native user in the Let's Go! spoken dialogue system: dealing with linguistic mismatch. In: *Proceedings of the HLT/NAACL* (2004), S. 217–224

Rayner et al. 2001

RAYNER, M. ; LEWIN, I. ; GORRELL, G. ; BOYE, J.: Plug and Play Speech Understanding. In: *Proceedings of 2nd SIGdial Workshop on Discourse and Dialogue* (2001), S. 1–10

Reeves, Nass 1996

REEVES, B. ; NASS, C.: *The media equation: How people treat computers, television, and new media like real people and places*. CSLI Publications/Cambridge University Press, 1996

Reips 2002

REIPS, U.-D.: Standards for Internet-based experimenting. In: *Experimental Psychology* 49 (2002), Nr. 4, S. 243–256

Rudnický et al. 1999

RUDNICKÝ, A.I. ; THYER, E. ; CONSTANTINIDES, P. ; TCHOU, C. ; SHERN, R. ; LENZO, K. ; XU, W. ; OH, A.: Creating natural dialogs in the Carnegie Mellon Communicator System. In: *Proceedings of the sixth European Conference on Speech Communication and Technology (ESCA)* (1999)

Sacks et al. 1974

SACKS, H. ; SCHEGLOFF, E. ; JEFFERSON, G.: A simple system for the organization of turn-taking in conversation. In: *Language* 50 (1974), Nr. 4, S. 696–735

San-Segundo et al. 2000

SAN-SEGUNDO, R. ; PELLON, B. ; WARD, W.: Confidence Measures for Dialogue Management in the CU Communicator System. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)* (2000)

Schaaf 2003

SCHAAF, Th.: *Erkennen und Lernen neuer Worte*, Interactive Systems Lab, Faculty of Computer Science, University of Karlsruhe, Diss., 2003

Schaaf, Kemp 1997

SCHAAF, Th. ; KEMP, Th.: Confidence measures for spontaneous speech recognition. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)* (1997)

Scheffler, Young 2002

SCHEFFLER, K. ; YOUNG, S.: Corpus-based dialogue simulation for automatic strategy learning and evaluation. In: *Proceedings of the Human Language Technologies (HLT'02)* (2002)

Schegloff et al. 1977

SCHEGLOFF, E. ; JEFFERSON, G. ; SACKS, H.: The preference for self-correction in the organization of repair in conversation. In: *Language* 53 (1977), S. 361–382

Schläfer et al. 2006

SCHLÄFER, N. ; GIESELMANN, P. ; SCHAAF, T. ; WAIBEL, A.: A Pattern Learning Approach to Question Answering within the Ephyra Framework. In: *Proceedings of the Ninth International Conference on TEXT, SPEECH and DIALOGUE* (2006)

Schmidt 1997

SCHMIDT, W. C.: World-Wide Web Survey Research: Benefits, Potential Problems, and Solutions. In: *Behavior Research Methods, Instruments & Computers* 29 (1997), Nr. 2, S. 274–279

Searle 1969

SEARLE, J.R.: *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, UK, 1969

Seneff et al. 2003

SENEFF, S. ; CHANG, G. ; WANG, C.: Empowering end users to personalize dialogue systems through spoken interaction. In: *Proceedings of the Eurospeech (2003)*, S. 749–752

Shin et al. 2002

SHIN, J. ; NARAYANAN, S. ; GERBER, L. ; KAZEMZADEH, A. ; BYRD, D.: Analysis of user behavior under error conditions in spoken dialogs. In: *Proceedings of the ICSLP 2002 (2002)*, S. 2069–2072

Shine et al. 2003

SHINE, J. ; NARAYANAN, S. ; GERBER, L. ; KAZEMZADEH, A. ; BYRD, D.: Analysis of user behaviour under error conditions in spoken dialog. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'03) (2003)*

Shinozaki, Furui 2003

SHINOZAKI, T. ; FURUI, S.: An Assessment of Automatic Speech Recognition Techniques for Spontaneous Speech in Comparison with Human Performance. In: *Proceedings of the Workshop on Spontaneous Speech Processing and Recognition (SSPR'03) (2003)*, S. 95–98

Singh et al. 2000

SINGH, S. ; LITMAN, D. ; KEARNS, M. ; WALKER, M.: Optimizing dialogue management with reinforcement learning: experiments with the NJFun system. In: *Journal of Artificial Intelligence Research* 16 (2000), S. 105–133

Skantze 2003

SKANTZE, G.: Exploring human error handling strategies: implications for spoken dialogue systems. In: *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems (2003)*, S. 325–341

Smith, Hipp 1994

SMITH, R. ; HIPPI, D.R.: *Spoken Natural Language Dialogue Systems: A Practical Approach*. Oxford University Press, New York, NY, 1994

Solsona et al. 2002

SOLSONA, R. A. ; FOSLER-LUSSIER, E. ; KUO, H.-K. J. ; POTAMIANOS, A. ; ZITOUNI, I.: Adaptive Language Models for Spoken Dialogue Systems. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'02) (2002)*, S. 37–40

Soltau, Waibel 2000

SOLTAU, H. ; WAIBEL, A.: Specialized acoustic models for hyperarticulated speech. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'00) (2000)*, S. 1779–1782

Stallard 2000

STALLARD, D.: Talk'n'Travel: A Conversational System for Air Travel Planning. In: *Proceedings of the Association for Computational Linguistics 6th Applied Natural Language Processing Conference (ANLP 2000)* (2000), S. 68–75

Stent et al. 1999

STENT, A. ; DOWDING, J. ; GAWRON, J. M. ; BRATT, E. O. ; MOORE, R.: The CommandTalk Spoken Dialogue System. In: *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL'99)* (1999), S. 183 – 190

Stiefelhagen et al. 2004

STIEFELHAGEN, R. ; FUEGEN, C. ; GIESELMANN, P. ; HOLZAPFEL, H. ; NICKEL, K. ; WAIBEL, A.: Natural Human-Robot Interaction using Speech, Gaze and Gestures. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2004), S. 53 – 60

Stifelman 1993

STIFELMAN, L. J.: User Repairs of Speech Recognition Errors: An Intonational Analysis. In: *Technical Report, Speech Research Group, MIT Media Lab* (1993)

Strube 1998

STRUBE, M.: Never Look Back: An Alternative to Centering. In: *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (1998), S. 1251–1257

Swerts et al. 2000

SWERTS, M. ; HIRSCHBERG, J. ; LITMAN, D.: Corrections in Spoken Dialogue Systems. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP'00)* (2000)

Tarone 1981

TARONE, Elaine: Some Thoughts on the notion of communication strategy. In: *TESOL Quarterly* 15 (1981), Nr. 3, S. 285–295

Tetreault, Allen 2003

TETREULT, J. ; ALLEN, J.: An Empirical Evaluation of Pronoun Resolution and Clausal Structure. In: *Proceedings of the 2003 International Symposium on Reference Resolution and its Applications to Question Answering and Summarization* (2003), S. 1–8

Traum 1998

TRAUM, D. R.: On Clark and Schäfer's Contribution Model and its Applicability to Human-Computer Collaboration. In: *Proceedings of the COOP 98 Workshop on Use of Clark's Models of Language for the Design of Cooperative Systems* (1998)

Traum 1999

TRAUM, D. R.: Computational Models of Grounding in Collaborative Systems. In: *Psychological Models of Communication in Collaborative Systems - Papers from the AAAI Fall Symposium (1999)*, S. 124—131

Traum, Dillenbourg 1998

TRAUM, D. R. ; DILLENBOURG, P.: Towards a Normative Model of Grounding in Collaboration. In: *Working notes of the ESSLLI-98 workshop on Mutual Knowledge, Common Ground and Public Information (1998)*

Walker et al. 2001

WALKER, M. ; ABERDEEN, J. ; BOLAND, J. ; BRATT, E. ; GAROFOLO, J. ; HIRSCHMAN, L. ; LE, A. ; LEE, S. ; NARAYANAN, S. ; PAPINENI, K. ; PELLON, B. ; POLIFRONI, J. ; POTAMIANOS, A. ; PRABHU, P. ; RUDNICKY, A. ; SANDERS, G. ; SENEFF, S. ; STALLARD, D. ; WHITTAKER, S.: DARPA Communicator dialog travel planning systems: the June 2000 data collection. In: *Proceedings of the Eurospeech 2001 2 (2001)*, S. 1371—1374

Walker et al. 2000

WALKER, M. ; WRIGHT, J. ; LANGKILDE, I.: Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System. In: *Proceedings of the 17th International Conference on Machine Learning (2000)*, S. 1111 – 1118

Walker 1998

WALKER, M. A.: Centering, Anaphora Resolution, and Discourse Structure. In: *Marilyn A. Walker, Aravind K. Joshi and Ellen F. Prince: Centering in Discourse (1998)*

Walker et al. 1997

WALKER, M. A. ; LITMAN, D. ; KAMM, C. A. ; ABELLA, A.: PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In: *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (1997)*, S. 271–280

Webb, Strzalkowski 2006

WEBB, N. ; STRZALKOWSKI, T.: *Proceedings of the HLT-NAACL 2006 Workshop on Interactive Question Answering. 2006*

Weegels 2000

WEEGELS, M.: Users' conceptions of voice-operated information services. In: *International Journal of Speech Technology 3 (2000)*, Nr. 2, S. 75–82

Weizenbaum 1996

WEIZENBAUM, J.: ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine. In: *Communications of the ACM 9 (1996)*, S. 36 – 45

Wertheimer 1923

WERTHEIMER, M.: Untersuchungen zur Lehre von der Gestalt II. In: *Psychologische Forschung* 4 (1923), S. 301–350

Williams et al. 2002

WILLIAMS, J. ; MICHELITSCH, G. ; MOEHLER, G. ; RAPP, S.: A Methodology for Evaluating Multimodality in Home Entertainment System. In: *Proceedings of the International Conference on Multimodal Interaction (ICMI'02)* (2002), S. 460–465

Xu, Rudnicky 2000

XU, W. ; RUDNICKY, A.: Language Modeling for Dialog System. In: *Proceedings of the International Conference of Speech and Signal Processing (ICSLP'00)* (2000)

Zhang, Lee 2002

ZHANG, D. ; LEE, W.: Web based Pattern Mining and Matching Approach to Question Answering. In: *Proceedings of the Eleventh Text REtrieval Conference* (2002)

Zheng 2002

ZHENG, Z.: AnswerBus Question Answering System. In: *Proceedings of the Human Language Technology Conference (HLT'02)* (2002)

Zollo 2003

ZOLLO, T.: Using Grammatical Analysis to Detect Misrecognitions. In: *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems* (2003), S. 83–88

Zue et al. 2000

ZUE, V. ; SENEFF, S. ; GLASS, J. ; POLIFRONI, J. ; PAO, C. ; HAZEN, T. J. ; HETHERINGTON, L.: JUPITER: A Telephone-Based Conversational Interface for Weather Information. In: *IEEE Transactions on Speech and Audio Processing* 8 (2000), S. 85–96

Teil V
Appendix

40 zufällig ausgewählte Fragen aus dem QA TREC Korpus 2004

1. what is the name of Durst's group
2. what record company is he with
3. when was James Dean born
4. what movies did he appear in
5. what does AARP stand for
6. who is its top official or CEO
7. how long does one study as a Rhodes scholar
8. what countries have Rhodes scholars come from
9. what kind of animal is an agouti
10. who founded the Black Panthers organization
11. who have been members of the organization
12. who are the members of this group
13. what are prions made of
14. who discovered prions
15. what researchers have worked with prions
16. what style of music do they play
17. what industry is Rohm and Haas in
18. what film introduced Jar Jar Binks
19. what actor is used as his voice
20. who are the members of the Rat Pack
21. what Las Vegas hotel was made famous by the Rat Pack

22. what is the primary symptom of a cataract
23. what kind of a community is a Kibbutz
24. what airlines have Concorde in their fleets
25. how many seats are in the cabin of a Concorde
26. what year was the first Concorde crash
27. how many Club Med vacation spots are there worldwide
28. where was Franz Kafka born
29. what was Gekko's profession
30. what kind of singer is Ice-T
31. what sport does Jennifer Capriati play
32. what was Jean Harlow's real name
33. what movies did she appear in
34. what leading men did she star opposite of
35. how many followers does it have
36. what is Florence Nightingale famous for
37. when did Amtrak begin operations
38. who is the president or chief executive of Amtrak
39. who were leaders of the Khmer Rouge
40. how many members are there in the singing group the Wiggles