Universität Karlsruhe (TH)

Studienarbeit

Lernen von Vor- und Nachnamen im natürlichsprachigen Mensch-Roboter-Dialog

Student:	Stefan Ultes
Betreuer:	Prof. Dr. A. Waibel
Betreuer:	Dipl.-Inform. H. Holzapfel
Tag der Abgabe:	30.06.2008

Sommersemester 2008

Fakultät für Informatik
Institut für Theoretische Informatik (ITI)

Mein besonderer Dank gilt Hartwig Holzapfel, der mich stets mit hilfreichen Impulsen unterstützt hat und der durch sein überragendes Engagement bei der Betreuung meiner Studienarbeit eine große Hilfe war.

Inhaltsverzeichnis

1. Einleitung **1**
- 1.1. Motivation . . . 1
- 1.2. Überblick . . . 1
- 1.3. Aufbau . . . 2

2. Grundlagen **3**
- 2.1. Natürlichsprachige Dialogsysteme . . . 3
- 2.2. Automatische Spracherkennung . . . 4
- 2.3. Dialogmanager . . . 5
  - 2.3.1. Verstehen natürlicher Sprache . . . 5
  - 2.3.2. Zustandsmodellierung . . . 6
  - 2.3.3. Personenidentifikation . . . 6
- 2.4. Sprachsynthese . . . 7
- 2.5. Evaluationsmethoden . . . 7
  - 2.5.1. Objektive Evaluationsmethoden . . . 8
  - 2.5.2. Subjektive Evaluationsmethoden . . . 8

3. Dialog **11**
- 3.1. Grammatik und Ontologie . . . 12
- 3.2. Zweistufige Namensdatenbank . . . 14
- 3.3. Dialogstrategien . . . 15
  - 3.3.1. Vorverarbeitung . . . 15
  - 3.3.2. Dialogstrategie für langen Dialog . . . 16
  - 3.3.3. Dialogstrategie für kurzen Dialog . . . 20

4. Experimente **27**
- 4.1. Beschreibung des Testsystems . . . 27
  - 4.1.1. IslEnquirer . . . 27
  - 4.1.2. Multimodale PersonID . . . 29
- 4.2. Durchführung der Experimente . . . 30

5. Evaluation **33**
- 5.1. Objektive Bewertung . . . 33

5.2. Subjektive Bewertung 34

6. Zusammenfassung & Ausblick **37**

A. Fragebogen **39**

Index **44**

Abbildungsverzeichnis

3.1. Grammatikregel zum Erkennen von Namen 13
3.2. Strategie für langen Dialog: Behandlung von Vornamen . . 17
3.3. Strategie für langen Dialog: Behandlung von Nachnamen . 18
3.4. Strategie für langen Dialog: Behandlung von Vor- und Nachnamen 19
3.5. Strategie für kurzen Dialog: Behandlung von Vornamen . . 21
3.6. Strategie für kurzen Dialog: Behandlung von Nachnamen . 22
3.7. Strategie für kurzen Dialog: Repeat-Modul für Vornamen . 23
3.8. Strategie für kurzen Dialog: Behandlung von Vor- und Nachnamen 24

4.1. Beispiel eines Dialoges mit impliziter Bestätigung 30

Tabellenverzeichnis

5.1. Wortfehlerraten der Dialoge (kurz/lang) 33
5.2. Dialoglänge 34
5.3. Wortfehlerraten der Dialoge (chronologisch) 34
5.4. Subjektive Bewertung 35
5.5. Subjektive Bewertung nach Benutzergruppen 36

1. Einleitung

1.1. Motivation

Im menschlichen Alltag dringen Roboter in immer mehr Bereiche vor. Eine der wichtigsten Herausforderungen stellt hierbei die Akzeptanz des Roboters durch den Menschen dar [BMS+05]. Dabei steht eine intuitive und natürliche Interaktion zwischen Mensch und Roboter an erster Stelle, wofür multimodale Kommunikationskanäle verwendet werden. Die Sprache ist dabei neben Gestik und Sehen eines der natürlichen Kommunikationsmittel. Für die Kommunikation mit einem Roboter mittels Sprache ist es wichtig, dass der Roboter intern ein Modell des Benutzers verwendet, um einzelne Benutzer identifizieren und wiedererkennen zu können. Auf diese Weise ist man in der Lage beispielsweise benutzerspezifische Aktionen auszuführen. Außerdem kann der Roboter Informationen über den Benutzer für seine Aktionen verwenden. Ein Beispiel hierfür wäre bei einem Haushaltsroboter das Wissen, wie ein Benutzer seinen Kaffee trinkt.

Für dieses Benutzermodell ist der Vor- und Nachname einer Person von besonderer Bedeutung. Erstens müssen Benutzer in diesem Modell mit einer eindeutigen ID verbunden werden, wofür sich der Name der Person eignet, da diese die natürliche, in der zwischenmenschlichen Interaktion verwendete ID ist. Dabei benutzt man den Vor- und Nachname, da so eine bessere Differenzierung zwischen verschiedenen Menschen erreicht werden kann. Zweitens muss der Roboter verstehen über welche Person gesprochen wird, wenn ein Mensch mit ihm über einen anderen Menschen redet. Ohne das Wissen über die Zuordnung von Name zu Person ist dies schlichtweg unmöglich.

1.2. Überblick

Die Aufgabe des Namenlernens stellt mehrere Herausforderungen. Eine der größten bildet die Frage, wie einfach entschieden werden kann, ob ein genannter Name ein Vor- oder Nachname ist. In dieser Arbeit werden dafür verschiedene Mittel miteinander kombiniert. Zuerst wird anhand des

Dialoges versucht, diese Entscheidung zu treffen. Dies ist häufig möglich, jedoch wird in der Arbeit gezeigt, dass dies nicht alle Fälle abdeckt. Für diese Fälle entscheidet man anhand einer zuvor berechneten Wahrscheinlichkeit.

Herausfordernd ist auch die Aufgabe, die Erkennungsleistung des Systems bei Namen zu verbessern. Dafür wird zunächst die Menge Namen, die erkannt werden sollen, eingeschränkt, wenn das Dialogsystem einen bestimmten Namenstyp erwartet. Zusätzlich wird die Erkennung der Namen hierarchisch aufgebaut. Das hat die Erkennungsleistung merklich verbessert.

Das Lernen der Namen soll in einem Dialog statt finden, der natürlich sein und gleichzeitig gut funktionieren soll. Um diesen Spagat zu meistern werden zwei Dialogstrategien vorgestellt. Es wird sich herausstellen, dass sie verschieden stark auf die Punkte Natürlichkeit und Funktionalität eingehen. Weiter wird eine Identifikation der Personen in den Dialogen durchgeführt.

1.3. Aufbau

In Kapitel 2 werden die Grundlagen eines natürlichsprachigen Dialogsystems vorgestellt. Dabei wird kurz die Funktionsweise eines Spracherkenners, der Sprachsynthese und des Dialogmanagers beschrieben. Im Kapitel Dialog (Kapitel 3) wird in einem Unterkapitel über die Modellierung der Grammatik und Ontologie gesprochen. Weiter wird eine Namensdatenbank vorgestellt und im Abschnitt über die Vorverarbeitung der Dialoge wird berichtet, wie die Entscheidung, ob ein Name als Vor- oder Nachname angesehen wird, realisiert ist. Danach werden verschiedene Dialogstrategien vorgestellt und im Detail erklärt. Die zur Bewertung des Systems durchgeführten Tests und das dazu gehörige Testsystem werden in Kapitel 4 beschrieben. Die Auswertung selbst mit subjektiver und objektiver Bewertung des Systems ist in Kapitel 5 erläutert. Schließlich findet sich im letzten Kapitel (Kapitel 6) eine Zusammenfassung der Arbeit mit einem kleinen Ausblick auf zukünftige Aufgaben.

2. Grundlagen

2.1. Natürlichsprachige Dialogsysteme

Ein natürlichsprachiges Dialogsystem oder englisch „Spoken Dialog System" setzt sich aus mehreren Komponenten zusammen, die in einer vorgegebenen Reihenfolge die Eingabe verarbeiten. Diese Kette von Modulen wird in jedem Turn (einzelne Benutzereingabe) durchlaufen um zu jeder Nutzereingabe die passende Ausgabe des Systems zu erzeugen. Die wichtigsten Komponenten sind dabei die Spracherkennung, die NLU-Einheit, der Dialogmanager und das Sprachsynthesemodul.

Die automatische Spracherkennung erzeugt aus mittels Nahsprechmikrophon eingegebener Sprache eine Textrepräsentation des Gesprochenen. In der NLU-Einheit (*Natural Language Understanding*, „Verstehen natürlicher Sprache") wird der erzeugte Text in eine semantische Repräsentation des Gesprochenen konvertiert. Diese hat die Form einer typisierten Merkmalstruktur (*Typed Feature Structure*, TFS) [Car92]. Bei mehreren Eingabemodalitäten existiert für jede dieser Modalitäten eine eigene Einheit zur semantischen Interpretation der Eingabe. Diese TFS wird anschließend im Kontext des aktuellen Dialogzustandes interpretiert. Der Dialozustand setzt sich dabei aus der Gesamtheit an bisherigen Äußerungen, dem Diskurs und internen Variablen zusammen. Bei dieser Interpretation können auch Konvertierungsregeln angewendet werden, um die Eingabe TFS auf die Erwartung des Dialogsystems abzubilden. Auf Grundlage der daraus resultierenden semantischen Repräsentation wählt die Dialogstrategie die nächste Systemaktion aus (*Move*). Dabei ist ein Move eine abstrakte Repräsentation und beinhaltet mehrere Einzelaktionen wie zum Beispiel Systemausgaben oder Datenbankzugriffe.struktur (*Typed Feature Structure*, TFS) [Car92]. Bei mehreren Eingabemodalitäten existiert für jede dieser Modalitäten eine eigene Einheit zur semantischen Interpretation der Eingabe. Diese TFS wird anschließend im Kontext des aktuellen Dialogzustandes interpretiert. Der Dialozustand setzt sich dabei aus der Gesamtheit an bisherigen Äußerungen, dem Diskurs und internen Variablen zusammen. Bei dieser Interpretation können auch Konvertierungs-

regeln angewendet werden, um die Eingabe TFS auf die Erwartung des Dialogsystems abzubilden. Auf Grundlage der daraus resultierenden semantischen Repräsentation wählt die Dialogstrategie die nächste Systemaktion aus (*Move*). Dabei ist ein Move eine abstrakte Repräsentation und beinhaltet mehrere Einzelaktionen wie zum Beispiel Systemausgaben oder Datenbankzugriffe.

Die Kommunikation zwischen den einzelnen Modulen wird über einen Kommunikationsagenten realisiert. Der IslEnquirer [Put08] und das Tapas Framework [Hol08], auf denen diese Arbeit basieren, verwenden dazu das One4All-Protokoll [1].

2.2. Automatische Spracherkennung

Bei heutigen automatischen Spracherkennern funktioniert die Spracherkennung auf statistischer Basis. Dabei wird der wahrscheinlichste Kandidat $\hat{W}$ aller Wortsequenzen W geschätzt, wobei eine Sequenz von Merkmalsvektoren X gegeben ist.

$$\hat{W} = \arg\max_{W} P(W|X) = \arg\max_{W} P(X|W)P(W)$$

$P(X|W)$ repräsentiert das akustische Modell und ist die Wahrscheinlichkeit für das Auftreten eines bestimmten Merkmalsvektors X unter der Bedingung, dass ein Wort W aufgetreten ist. Dies wird meist mittels Hidden Markov Modellen (HMMs) für einzelne Phoneme modelliert. $P(W)$ stellt das Sprachmodell dar. Es wird häufig durch eine kontextfreie Grammatik modelliert, wobei alle Sätze, die nicht in der Grammatik liegen, eine Wahrscheinlichkeit von Null erhalten. Ein Vorteil der Modellierung mittels kontextfreier Grammatik ist, dass nur Hypothesen erkannt und an die NLU-Einheit weitergegeben werden, die für sie auch interpretierbar sind. Dies gilt allerdings nur, wenn beide Module die gleiche Grammatik verwenden. Eine weitere Möglichkeit, das Sprachmodell zu modellieren, stellt das Konzept der n-Gramme dar, das in heutigen Spracherkennern häufig Anwendung findet. Dabei werden aufgrund eines Trainingskorpus für jede Wortfolge eine Wahrscheinlichkeit berechnet, mit der ein Wort auf eine Sequenz von n-1 vorangegangenen Wörtern folgt. Grundlegend für dieses Modell ist dabei die Markov-Annahme.

Der Spracherkenner, der in diesem System zum Einsatz kommt, ist Janus. Dabei stellt Janus noch drei erweiterte Funktionen zur Verfügung.

[1] Ein multimodales System entwickelt am ISL, Universität Karlsruhe

Janus bietet die Erkennung von unbekannten Wörtern, also Wörter, die nicht im Vokabular stehen (OOV, *Out Of Vocabulary*). Außerdem kann sich Janus an den Sprecher adaptieren, so dass die Erkennungsleistung bei längeren Dialogen mit der selben Person stetig zunimmt. Des weiteren wird eine Erkennung von bestimmten Wortklassen in zwei Phasen unterstützt, was bei der Namenserkennung zu deutlich besseren Erkennungsraten führt. Dieses Thema wird in 3.2 vertieft.

Zur Realisierung einer exakten orthographischen Repräsentation unbekannter Wörter wird ein eigenes Modell für das Buchstabieren von Wörtern verwendet. Hidden Markov Modelle modellieren dabei jeden einzelnen Buchstaben. Trainiert wurden diese Modelle auf deutschen Vor- und Nachnamen, da in dieser Arbeit dies die einzigen unbekannten Wörter sind.

2.3. Dialogmanager

2.3.1. Verstehen natürlicher Sprache

Das Verstehen natürlicher Sprache bildet den Kern eines Dialogsystems. Ziel dieses Moduls ist aus einer Nutzereingabe in Textform Semantik zu extrahieren. Dazu wird die Eingabe in ihre semantische Repräsentation transformiert, dargestellt durch eine typisierte Merkmalsstruktur (*Typed Featured Structure*, TFS) [Car92]. Eine solche TFS beinhaltet Merkmal-Wert-Paare und beschreibt mit ihnen die Eigenschaften eines semantischen Konzepts. Dabei kann ein solches Attribut ein atomarer Wert oder erneut eine TFS sein. Jede TFS hat einen bestimmten Typ, der in einer Ontologie beschrieben ist. Diese Ontologie ist hierarchisch aufgebaut und unterstützt das Konzept der Vererbung. Durch die Darstellung der Semantik als TFS lässt sich die neue Eingabe auch durch Transformation und Unifikation in den Diskurs einbetten. Die Unifikation wird außerdem für die Fusion von semantischen Repräsentationen aus verschiedenen Eingabemodalitäten benutzt.

Zum Verstehen der Sprache wird eine semantische kontextfreie Grammatik verwendet. In der Grammatik wird unterschieden zwischen Terminalen und Nichtterminalen. Die Nichtterminale unterscheiden sich wiederum in syntaktische und semantische Einträge. Syntaktische Einträge dienen lediglich der Grammatikstruktur, wohingegen semantische Einträge ein Konzept der Ontologie widerspiegeln. Zusätzlich dazu enthält ein semantischer Eintrag eine grammatische Klasse und einen zusätzlichen,

frei wählbaren Tag. Das Janus Framework und das Tapas Dialog Toolkit [Hol08], dessen NLU-Einheit in dieser Arbeit verwendet wird, nutzen dieselbe Grammatik. Dabei existiert eine zusätzliche Funktionalität: Es ist möglich, die Grammatik dynamisch zu erweitern, in dem Inhalt eines Nichtterminalknotens aus einer Datenbank geladen wird. Dies ist in dieser Arbeit besonders wichtig, da ansonsten die Namen von Hand in die Grammatik eingetragen werden müssten. Außerdem ist es nur so möglich, während des Dialogs gelernte Namen der Grammatik hinzuzufügen.

2.3.2. Zustandsmodellierung

Die Modellierung des Systemzustandes ist ein wichtiger Teil des Dialogmanagers, da häufig Ausgaben aufgrund des bisherigen Dialogverlaufes inklusive des Diskurses ausgewählt werden müssen. Die drei für diese Arbeit wichtigen Modelle werden hier vorgestellt:

- **Slot Model** Im *Slot Model* stehen atomare Informationen in so genannten Slots. Dabei hat jeder Slot einen eigenen Namen und alle Slots zusammen stellen das bisher im Dialog gesammelte Wissen dar.

- **State Model** Das *State Model* beschreibt durch eine Menge an Variablen den internen Dialogzustand.

- **Session Model** Das *Session Model* gibt Auskunft darüber, ob momentan eine Session läuft. Eine Session ist in dieser Arbeit ein kompletter Namenlerndialog. Das Modell kann noch zusätzliche Informationen tragen, z.B. wie lange zwischen einzelnen Sessions gewartet werden soll. Weiter beinhaltet es Funktionen wie z.B. die Kontrolle über das Training der Personenidentifikationskomponenten.

2.3.3. Personenidentifikation

Die Identifiation von Personen funktioniert über akustische und visuelle Merkmale. Aus den akustischen Merkmalen wird durch ein Gauss-Mixtur-Modell auf phonetischen Ketten eine Sprecher-ID erzeugt. Bei der visuellen Erkennung wird das System Arthur verwendet, das mittels diskreter Kosinustransformation Personen anhand der Gesichter erkennt und außerdem mehrere Personen gleichzeitig tracken kann. Diese beiden Modalitäten werden zu einer multimodalen Hypothesenliste mit zugehöriger Konfidenz fusioniert.

2.4. Sprachsynthese

Da ein Dialog bekanntlich aus mindestens zwei Sprechern besteht, muss das Dialogsystem auch in der Lage sein, mit dem Benutzer zu sprechen. Dafür wird ein Sprachsynthesewerkzeug verwendet, das Text in Sprache wandelt (*Text-to-speech system*, TTS). Dies kann auf zwei verschiedene Arten geschehen. Bei einem konkatenativen Ansatz werden kurze Schnipsel natürlicher Sprache aneinander gehängt. Werden die Übergänge zwischen den einzelnen Schnipseln noch geglättet und der Sprache Prosodie hinzugefügt, erhält man eine menschenähnliche Stimme. Nachteil dieses Verfahrens ist, dass man eine große Menge Audiodaten benötigt. Der zweite Ansatz ist modellbasiert. Durch Training der Modellparameter kann so die Stimme angepasst werden, wofür nur eine kleine Menge Audiodaten benötigt wird. Jedoch wirkt die erzeugte Stimme maschinell und unnatürlich.

In dieser Arbeit wird das Sprachsynthesewerkzeug Cepstral[2] verwendet. Es unterstützt eine automatische Abbildung von Graphemen zu Phonemen, die mittels Entscheidungsbäumen trainiert wird. Zusätzlich dazu können Graphem-Phonem-Zuordnungen von Hand in ein Wörterbuch eingetragen werden. Dies ist besonders bei Namen hilfreich, da diese häufig anderen Regeln gehorchen als normale Sprache. Zur Modifizierung der Sprachausgabe unterstützt Cepstral die Speech Synthesis Markup Language (SSML)[3]. Besondere Bedeutung hat dabei die Möglichkeit der Betonung einzelner Wörter oder Wortgruppen. Dies wird durch den Tag *emphasis* realisiert. Nebenprodukt der Betonung ist ein langsameres und deutlicheres Aussprechen, was insbesondere bei der Aussprache von Namen sehr hilfreich ist.

2.5. Evaluationsmethoden

Das Ziel einer Evaluation eines Dialogs ist, dessen Güte festzustellen. Hierfür werden objektive, messbare Werte verwendet. Da es sich dabei um eine Interaktion zwischen Roboter und Mensch handelt, sind allerdings auch subjektive Kriterien wichtig. Weiter sind bisher Arbeiten vorgestellt worden, bei denen beides kombiniert wurde, um so eine bessere Vorhersage der Systemgüte zu ermöglichen [WLKA97, WLKA98, WKL00]. Dies wurde in dieser Arbeit nicht betrachtet.

[2] http://www.cepstral.com
[3] http://www.w3.org/TR/speech-synthesis/

2.5.1. Objektive Evaluationsmethoden

Die Auswahl an objektiven Evaluationsmethoden ist eingeschränkt. Mögliche Metriken sind Dialoglänge, Erfolgsrate oder Wortfehlerrate (*Word Error Rate*, WER). Bei der Dialoglänge wird die Anzahl an Turns gezählt, die ein Dialog durchschnittlich dauert. Die Erfolgsrate gibt das Verhältnis zwischen Dialogen, bei denen das Ziel erreicht wurde, und allen Dialogen wieder. Die Wortfehlerrate ist eines der wichtigsten Maße zur Bewertung eines natürlichsprachigen Dialogsystems. Sie vergleicht die Hypothese des Systems mit der Referenz, die durch Transskription dessen, was der Benutzer wirklich gesagt hat, entsteht. Dabei werden die Unterschiede (Einfügen, Löschen und Ersetzen) gezählt und durch die Anzahl an Wörtern der Referenz geteilt. So ergibt sich folgende Formel:

$$WER = \frac{\#Einfuegen + \#Loeschen + \#Ersetzen}{Referenzlaenge}$$

2.5.2. Subjektive Evaluationsmethoden

Um einen Dialog anhand subjektiver Kriterien zu evaluieren, müssen Eindrücke der Benutzer ermittelt werden. Eine weit verbreitete Möglichkeit dafür sind Fragebögen. Bei der Erstellung eines Fragebogens müssen in mehreren Punkten grundsätzliche Entscheidungen getroffen werden.

Zunächst stellt sich die Wahl zwischen Aussagen und Fragen. Aussagen sind strikt positiv oder negativ gestellt und einfach zu erstellen. Durch eine Likert-Skala soll der Benutzer seinen Grad der Zustimmung angeben. Allerdings wird durch eine bewertende Formulierung der Benutzer häufig in eine Richtung gedrängt und er wird voreingenommen. Dies kann man durch wertfrei formulierte Fragen beheben, allerdings ist die Erstellung solcher Fragen wesentlich anspruchsvoller und schwieriger. Zusätzlich müssen noch passende Antworten gefunden werden. Die Antworten werden bei geschlossenen Fragen mittels einer Rating-Skala ermittelt, wobei die Benutzer die Wahl innerhalb des vorgegebenen Antwortspektrums haben.

Unabhängig von der Wahl zwischen Aussagen und Fragen gibt es die Möglichkeit, eine gerade oder eine ungerade Anzahl an Antwortmöglichkeiten anzubieten. Bei ungerader Anzahl existiert eine Antwort, die eine neutrale Meinung ausdrückt. Benutzer neigen dazu, häufig die neutrale Antwort auszuwählen ohne richtig zu überlegen. Dadurch können die Ergebnisse verfälschte werden. Bei einer geraden Anzahl ist dies durch Fehlen einer neutralen Antwort nicht möglich, jedoch können Benutzer, die tat-

sächlich eine neutrale Meinung haben, diese nicht auswählen und werden gezwungen, sich für eine Seite zu entscheiden. Auch hierdurch können die Ergebnisse verfälscht werden.

Allgemein sollte außerdem darauf geachtet werden, dass sich die positive Bewertung nicht immer auf der gleichen Seite der Skala befindet, da so die Befragten den Fragebogen aufmerksamer durchlesen müssen.

Für die Auswertung eines Fragebogens müssen die durchschnittlichen Antworten auf die einzelnen Fragen (bzw. Aussagen) und die Standartabweichung bzw. die Varianz der Antworten ermittelt werden. Um Aufschlüsse über einzelne Aspekte des Systems zu gewinnen, ist eine einzelne Betrachtung der Fragen notwendig. Doch diese Auswertung ist aufwendig und da nur einzelne Aspekte pro Frage bewertet werden, sind klare Aussagen wie beispielsweise Akzeptanz des System schwer herauszufiltern.

Von Müller et al. [MSBK07] wurde deshalb ein Verfahren vorgestellt, das generelle Aussagen finden soll. Auf den Ergebnissen der einzelnen Fragen wurde dabei eine Hauptkomponentenanalyse (*Principal Component Analysis*, PCA) angewendet, um dahinterliegende Aussagen herauszufinden. In [MSBK07] wurde dies erfolgreich für zwei Beispielfragebögen durchgeführt: den SASSI-Fragebogen [HG00] und ITU-T Rec, P.851 [IT03]. Zusätzlich wurde die Validität und die Reliabilität der durch die PCA ermittelten Aussagen mittels Cronbachs α gezeigt.

3. Dialog

Bei der Entwicklung eines Dialoges bzw. einer Dialogstrategie stehen neben der Funktionalität des Dialoges - das Dialogziel - auch andere subjektive Kriterien im Fokus der Entwickler. In dieser Studienarbeit wurde die Natürlichkeit des Dialoges als wichtigstes subjektives Kriterium in diesen Fokus gestellt. Demnach sollte ein guter Dialog gleichzeitig funktional und natürlich sein. Werden in einem Dialog nacheinander einzeln Vorname und Nachname abgefragt, kann das Dialogziel des Lernens von Vor- und Nachnamen zwar einfach erreicht werden, aber der Dialog wäre unnatürlich und einem Dialog zwischen zwei Menschen sehr unähnlich. Eine Dialogstrategie, die ein Maximum an Natürlichkeit erfüllen möchte, ist sehr komplex. Zum einen ist diese Komplexität mit einer grammatikbasierten Strategie nicht zu beherrschen, zum anderen enthält eine solche Strategie so viel Eingabemöglichkeiten, dass die Fehlerkennungsrate sehr hoch sein würde. Gesucht ist demnach ein Mittelweg zwischen den beiden Extremen.

Die Aufgabe stellt viele Herausforderungen. Eine davon bildet die Fragestellung, wie man den Benutzer dazu bekommt, seinen Vor- und Nachnamen zu nennen. Bei den hier vorgestellten Dialogstrategien wurde ein Ansatz gewählt, bei dem sich der Roboter zu Beginn selbst mit seinem Vor- und Nachnamen vorstellt um dann den Benutzer nach seinem Namen zu fragen. Dieses explizite Fragen nach dem Namen würde in der menschlichen Interaktion zwar unnatürlich wirken. Jedoch hat sich bei den ersten Tests herausgestellt, dass viele Benutzer nicht wussten, dass von ihm erwartet wurde, den eigenen Namen zu nennen. Dies stellt einen gravierenden Unterschied zur zwischenmenschlichen Kommunikation dar, wodurch deutlich wird, dass die Benutzer dem Roboter nicht gegenübertreten, wie sie das bei einem Menschen tun würden. Bereits dadurch wird klar, dass eine natürliche Kommunikation, wie sie zwischen Menschen möglich wäre, nicht erreicht werden kann. Einzig eine möglichst gute Annährung an dieses Ideal kann angestrebt werden.

Nennt der Benutzer seinen vollen Namen, geht das System davon aus, dass der Name in der Reihenfolge *Vorname Nachname* genannt wurde. Somit ist eine Zuordnung einfach. Problematisch wird es, wenn der Benutzer nur einen Namen nennt und nicht zusätzlich spezifiziert, ob der

genannte Name sein Vor- oder Nachname ist, beispielsweise „My name is Fritz". Durch eine geeignete Modellierung der Grammatik und der Hintergrunddatenbank kann dies in der Dialogstrategie jedoch erkannt und aufgelöst werden.

Hat das System den Vor- und Nachnamen gelernt, soll die Person von nun an bekannt sein. Hierfür benötigt der Roboter einen Speicher, in dem bekannte Personen gespeichert sind. Zusätzlich muss dieser Speicher um neue Personen erweitert werden können. Dazu wird eine einfache Tabelle verwendet, in der Vor- und Nachname und eine eindeutige ID eingetragen werden. Es existieren noch zwei weitere Einträge, die zur Identifikation einzelner Namen als Vor- oder Nachname dienen. Diese Tabelle bildet das „Personengedächtnis" des Roboters. Die Wiedererkennung von Personen wird in dem hier beschriebenen Dialog durch die Arbeit von Philipp Große [Gro08] realisiert, die in 4.1.2 beschrieben ist. Dabei liefert die von ihm vorgestellte Multimodale Benutzer-ID eine Hypothese und eine Konfidenz. Ist die Konfidenz groß genug, wird die Hypothese mit der Tabelle der bekannten Personen verglichen. Existiert in ihr ein passender Eintrag, werden Vor- und Nachname ausgelesen und dem Benutzer vorgestellt, der diese dann verifzieren oder falsifizieren kann. Wird die Hypothese nicht bestätigt, handelt es sich nicht um die angenommene Person und das System beginnt mit dem Namenlernen.

3.1. Grammatik und Ontologie

Für die Extraktion von Semantik aus einem Textstring wird eine kontextfreie Grammatik gekoppelt mit einer Ontologie verwendet. In der Ontologie sind die bedeutungstragenden Objekte und Aktionen beschrieben. Das wichtigste Objekt des Dialogs ist das Objekt *obj_name*. Es enthält das Attribut *NAME* als String, das den Namen enthält, und das Attribut *PROB_IS_FIRST_NAME* als Integer, das für die Entscheidung wichtig ist, ob der Name als Vor- oder Nachname behandelt wird. Abgeleitet von diesem Objekt existieren die Objekte *obj_firstname* und *obj_lastname*. Diese zwei Typen erben von *obj_name* und dienen nur der Unterscheidung in der Grammatik. Des weiteren gibt es noch die Objekte *obj_correctedName* und *obj_correctedFullName*, die für die negative Bestätigung mit gleichzeitiger Korrektur des Namen in einem Turn verwendet werden.

Ein Teil der in der Ontologie definierten Aktionen dienen dem Dialogbeginn und -ende. Dafür gibt es die Aktionen *act_greeting* und *act_goodbye*.

Es existiert noch eine weitere Aktion, *act_isFirstLast*, mit der nachgefragt wird, ob ein Name Vor- oder Nachname ist. Dazu wird das Attribut *PROB_IS_FIRST_NAME* als Boolean verwendet. Die wichtigste Aktion ist jedoch *act_informName*. Sie enthält drei Attribute: *NAME*, *FIRSTNAME* und *LASTNAME*. *NAME* ist vom Typ *obj_name*, *FIRSTNAME* vom Typ *obj_firstname* und *LASTNAME* von Typ *obj_lastname*. Diese Aktion stellt den Sprechakt dar, in dem der Benutzer dem Roboter seinen Namen mitteilt.

Für fast jeden Ontologieeintrag existiert eine Regel in der Grammatik. Einzig der Eintrag *obj_name* hat keine eigene Regel und kann nur in Verbindung mit einer *act_informName* auftreten. Aufgrund ihrer syntaktischen Bedeutung tragen alle Objekte die Klasse N oder NP (Nomen oder Nomephrase). Das hat auch eine positive Auswirkung auf die Erkennungsleistung des Spracherkenners, da diese Regeln im Normalfall weniger stark gewichtet sind. Erwartet die Dialogstrategie an bestimmten Stellen ein solches Objekt, wird die entsprechende Regel besonders gewichtet. Äußerungen, die der Benutzer an jeder Stelle im Dialog machen kann, sollten immer erkannt werden. Ein Beispiel dafür ist das Verabschieden. Da eine solche Regel, die Ontologieeinträge vom Typ Aktion als Wurzel hat, meist vom Typ V oder VP (Verb oder Verbphrase) ist, hat sie im Spracherkenner eine normale Gewichtung und kann daher leicht erkannt werden.

```
<act_informName,VPctx,_> =
[<name_prefix>] <obj_name,N,obj> {NAME obj_name}
  | <name_prefix_fn> <obj_firstname,N,obj> {FIRSTNAME obj_firstname}
  | <name_prefix_ln> <obj_lastname,N,obj> {LASTNAME obj_lastname}
  | [<name_prefix>] <obj_firstname,N,obj> {FIRSTNAME obj_firstname}
    <obj_lastname,N,obj> {LASTNAME obj_lastname};
```

Abbildung 3.1.: Grammatikregel zum Erkennen von Namen

In Abbildung 3.1 ist die wichtigste Regel zur Erkennung von Namen zu sehen. Sie hat die syntaktische Klasse VPctx (Verbphrase Kontext), woraus eine weniger starke Gewichtung im Spracherkenner resultiert und diese Regel im Normalfall nicht angewendet wird. Nennt ein Benutzer seinen Namen, sagt er entweder nur den Namen oder er setzt noch einen Namensvorsatz *name_prefix* davor. Dabei handelt es sich um eine Wortfolge wie z.B. „my name is“ oder „I am“. Durch diesen Vorsatz kann bereits auf sprachlicher Ebene eine Zuordnung des Namens zum Vor- bzw. Nachnamen geschehen. Sagt der Benutzer beispielsweise „my first name

is" handelt es sich eindeutig um einen Vornamen. Dies wird durch die Konzepte *firstname_prefix* für Vornamen und *lastname_prefix* für Nachnamen modelliert. Die Grammatikregel modelliert außerdem den Fall, dass ein Benutzer seinen vollen Namen nennt. Dies wird durch die letzten zwei Zeilen in Abbildung 3.1 realisiert, wobei die Reihenfolge der Namen entscheidet, was als Vor- und was als Nachname betrachtet wird.

3.2. Zweistufige Namensdatenbank

Die Menge der Namen von Personen, die der Roboter kennen lernen soll, ist fast unendlich. Trotzdem muss dem Roboter eine Mindestanzahl an Namen bekannt sein. Denn wenn Benutzer jeden unbekannten Namen buchstabieren, ist dies unnatürlich. Um dem Roboter in die Lage zu versetzen, viele Namen zu erkennen auch wenn er keine Person mit dem entsprechenden Namen kennt, wird eine Tabelle mit den je Tausend häufigsten Vor- und Nachnamen erstellt. Grundlage für die Datenbank sind Telefonbucheinträge der Städte Berlin, Hamburg, München, Stuttgart und Karlsruhe aus dem Jahre 1997 entnommen aus D-INFO 97. Dabei wurden die Vor- und Nachnamen getrennt behandelt. Bei der Datenbankerstellung stellt sich allerdings heraus, dass es Namen gibt, die sowohl Vor- als auch Nachname sind. Um mit diesen Namen im Dialog umgehen zu können, wurde bei der Erstellung der Datenbank auch eine Wahrscheinlichkeit p berechnet, mit der der Name ein Vorname ist. Die Wahrscheinlichkeit, dass es sich um einen Nachnamen handelt, ist dann $1\text{-}p$. Diese Wahrscheinlichkeit p wurde zusätzlich zu jedem Namen in der Datenbank gespeichert, wobei ein Name, der nur Vorname ist, eine Wahrscheinlichkeit von 100% und ein Name, der nur Nachname ist, eine Wahrscheinlichkeit von 0% hat.

Wird diese Datenbank für die Namenserkennung in der grammatikbasierten Spracherkennung verwendet, ist mit hohen Fehlerraten zu rechnen. Diese Fehlerraten sind im Allgemeinen nicht zu vermeiden, allerdings kann diese Situation für bekannte Personen optimiert werden. Deshalb wurde auf Grundlage der Diplomarbeit von Stefan Ziesemer [Zie07] die Datenbank hierarchisch in zwei Stufen entworfen. Dabei enthält die obere Tabelle die Personen, die dem Roboter bekannt sind. Die eine Stufe darunterliegende Tabelle entspricht dann der beschriebenen Tabelle der tausend häufigsten Namen. Auch in der Tabelle der bekannten Personen muss zu jedem Namen eine Wahrscheinlichkeit gespeichert werden, mit der der Name ein Vorname ist. Hier wird allerdings jeder Vorname mit 100% und jeder Nachname mit 0% Wahrscheinlichkeit eingetragen. Diese

Einschränkung ist vertretbar, da ein Mensch einen Namen auch anhand der ihm bekannten Personen als Vor- oder Nachname klassifizieren würde. Existiert ein Name bei einer Person als Vorname und bei einer anderen als Nachname, kann es zu unvorhergesehenen Auswirkungen kommen. Dieser Spezialfall wurde hier nicht näher betrachtet.

3.3. Dialogstrategien

Im Rahmen der Studienarbeit wurden zwei Dialogstrategien entwickelt. Die erste Strategie verwendet konsequent explizite Bestätigungen und ist darauf ausgelegt, in jedem Fall den vollen Namen der Person zu lernen. Dies kann dazu führen, dass bei häufiger Falscherkenung des Namens der Roboter wiederholt nachfragt oder den Benutzer buchstabieren lässt. Ein Fall wie dieser sollte nur äußerst selten auftreten. Tritt er jedoch auf, wirkt der Dialog sehr unnatürlich. Um diese Unnatürlichkeit zu vermeiden wurde noch eine zweite Dialogstrategie entworfen, bei der es nur eine begrenzte Anzahl an Rückfragen gibt und somit das unnatürliche, ständige Wiederholen nicht auftreten kann. Zusätzlich wurde in der zweiten Strategie teilweise eine implizite Bestätigung realisiert, was den Dialog kürzer und noch natürlicher wirken lassen soll.

3.3.1. Vorverarbeitung

Bei der Integration in Tapas [Hol08] bildet die Dialogstrategie die Instanz, die den Dialogablauf steuert, indem sie den nächsten Move auswählt. Um die Moveauswahl sinnvoll bewerkstelligen zu können, muss ein Weg gefunden werden, einen Namen als Vor- oder Nachnamen zu identifizieren, wenn dies nicht aus dem Dialog ableitbar ist. Zur Lösung wird das Feld *PROB_IS_FIRST_NAME* ausgelesen. Dabei wird der Bereich in verschieden Teilbereiche unterteilt. Die unteren 10% und die oberen 10% werden eindeutig zugeordnet. Im Bereich von 10% bis 90% kann keine eindeutige Zuordnung gemacht werden.

Aufgrund dieser Einteilung wird ein Name als Vorname in folgenden Fällen klassifiziert:

- Die Wahrscheinlichkeit ist bei über 90% und es ist bisher noch kein Name als Vorname bestätigt worden.
- Der Nachname ist bekannt und wurde bereits bestätigt.

- Beim Nachfragen, ob es sich um einen Vor- oder Nachnamen handelt wurde der Name vom Benutzer als Vorname erklärt.

Die Klassifizierung des Nachnamens erfolgt analog zu der des Vornamens. Trifft keiner der oberen drei Fälle zu, wird nur nach der Wahrscheinlichkeit unterschieden. Da zwischen 10% und 90% keine eindeutige Zuordnung gemacht werden kann, wird hier eine Hypothese erstellt, die der Benutzer bestätigen soll. Hat der Roboter beispielsweise bei dem Namen „Fritz" auf Grund der Wahrscheinlichkeit von 73,4% die Hypothese, dass es sich um einen Vornamen handelt, kann der Benutzer gefragt werden „Is Fritz your first name?". Da die Entscheidung im Bereich um die Wahrscheinlichkeit von 50% sehr ungenau ist, wird im Bereich zwischen 40% und 60% direkt nachgefragt, ob es sich bei dem Namen um einen Vor- oder Nachnamen handelt.[1]

Nach dieser Vorverarbeitung der Eingabe wird die eigentliche Strategie durchgeführt.

3.3.2. Dialogstrategie für langen Dialog

Die Strategie ist in drei verschiedene Fälle unterteilt:

1. Bisher wurde nur der Vorname genannt (Abb. 3.2).
2. Bisher wurde nur der Nachname genannt (Abb. 3.3).
3. Es wurde bereits Vor- und Nachname genannt (Abb. 3.4).

Fälle 1 und 2 Tritt der erste Fall das erste Mal auf, ist der Name im Allgemeinen noch nicht bestätigt. Deshalb muss zuerst überprüft werden, ob es sich um ein OOV handelt. Ist dies der Fall, wird der Benutzer gebeten, seinen Vornamen zu buchstabieren. Hierbei wird davon ausgegangen, dass ein Wiederholen des Namens kein besseres Ergebnis liefert, da angenommen wird, dass der Benutzer seinen Namen bei wiederholtem Aussprechen jedes Mal gleich aussprechen wird. Wurde kein OOV sondern ein Name erkannt, wird überprüft, ob davor bereits nach einer Bestätigung des Namens gefragt wurde. Falls es bisher nicht geschehen ist, wird es an dieser Stelle durchgeführt. Dabei hat der Benutzer drei verschiedene

[1] Bei dieser Einteilung handelt es sich um frei gewählte Grenzen, die sich in der Praxis bewährt haben.

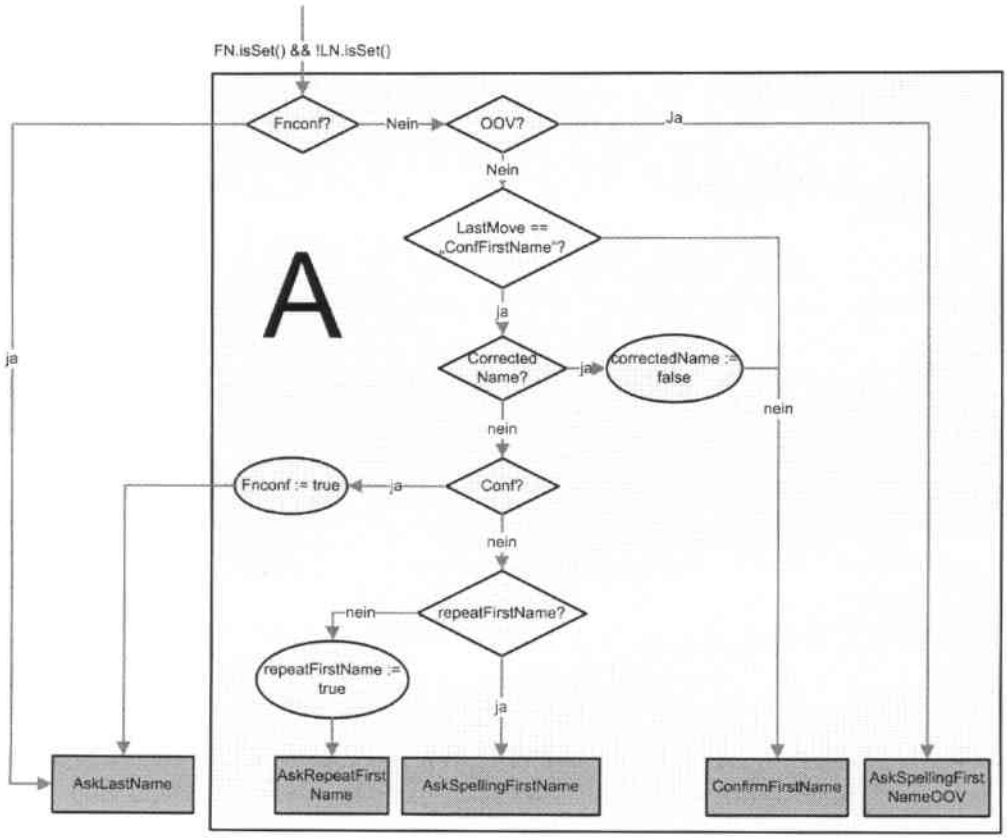

Abbildung 3.2.: Strategie für langen Dialog: Behandlung von Vornamen

Antwortmöglichkeiten: Er kann bestätigen, ablehnen oder einen korrigierten Namen sagen (*correctedName*). Nennt der Benutzer einen anderen Namen, wird erneut nach einer Bestätigung des Namens gefragt. Bei Bestätigung wird der Vorname als bestätigt markiert und mit dem Fragen nach dem Nachnamen fortgefahren. Lehnt der Benutzer den Namen jedoch ab, wird er entweder gebeten, seinen Vornamen zu wiederholen, oder, falls das bereits geschehen ist, zu buchstabieren. Nach dem Wiederholen bzw. Buchstabieren wird der Benutzer erneut dazu aufgefordert, den Namen zu bestätigen. Sollte der Name abgelehnt werden, muss der Benutzer wieder buchstabieren. Dies kann so lange weiter gehen, bis eine maximale Anzahl an Turns aufgetreten ist und der Dialog beendet wird, ohne den Namen zu lernen.

Sollte der erste Fall auftreten, obwohl der Vorname bereits bestätigt ist, wird direkt mit dem Fragen nach dem Nachnamen fortgefahren.

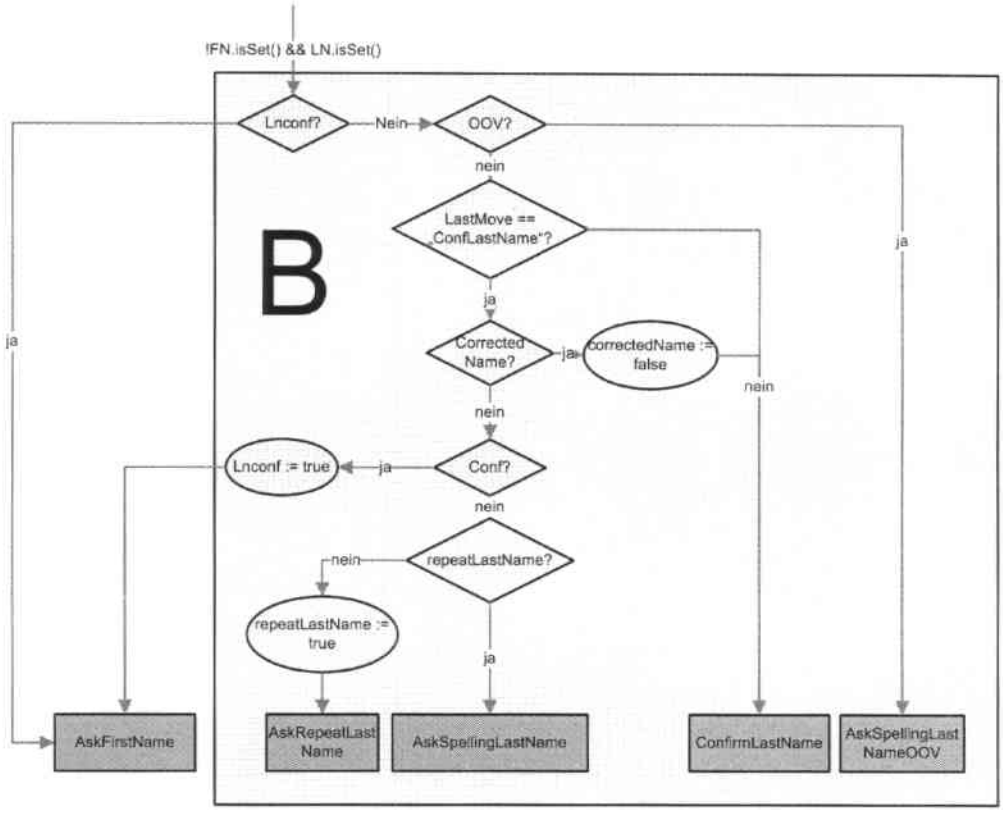

Abbildung 3.3.: Strategie für langen Dialog: Behandlung von Nachnamen

Der zweite Fall gestaltet sich analog zum ersten Fall. Ist der Nachname bestätigt, wird mit dem Fragen nach dem Vornamen fortgefahren.

Fall 3 Der dritte Fall ist verglichen mit den ersten beiden wesentlich komplexer. Er kann in zwei Situationen auftreten: nur ein Name ist nicht bestätigt oder beide Namen sind nicht bestätigt. Der erste Grund ergibt sich als Folge der Fälle 1 und 2. Ein unbestätigter Vor- und Nachname tritt auf, wenn zu Beginn der Benutzer, statt nur den Vornamen wie in Fall 1 oder nur den Nachnamen wie in Fall 2 zu nennen, seinen Vor- und Nachname nennt.

Nennt der Benutzer bereits zu Beginn seinen vollen Namen wird versucht, den Namen als ganzes zu bestätigen. Als Vorbedingungen dürfen dabei weder der Vor- noch der Nachname bestätigt sein. Außerdem darf noch nicht versucht worden sein, den vollen Namen bestätigen zu lassen. Werden die Vorbedingungen eingehalten, kann die Bestätigung des gan-

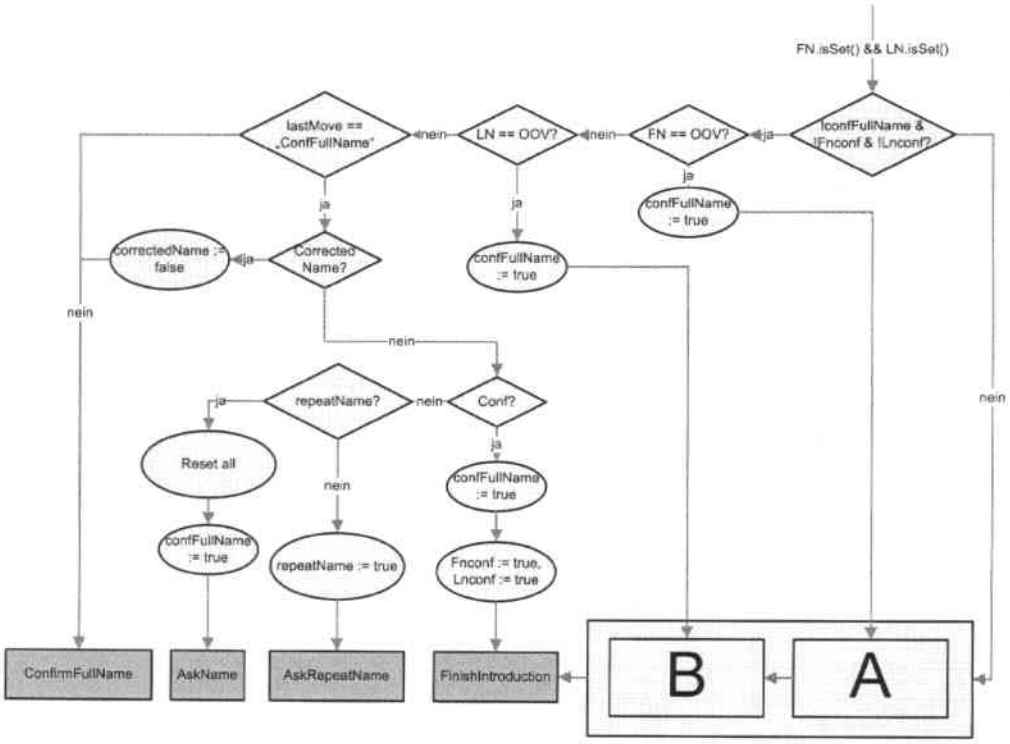

Abbildung 3.4.: Strategie für langen Dialog: Behandlung von Vor- und Nachnamen

zen Namens erfolgen. Auch in diesem Fall kann ein OOV als Vor- oder Nachname auftreten[2], was vor der Bestätigung überprüft werden muss. Tritt ein OOV auf, wird der Bestätigungsversuch des ganzen Namens abgebrochen und mit der Behandlung des OOV im Modul A bzw. Modul B fortgefahren. Hierbei findet in Modul A die Behandlung der Vornamen (siehe Abb. 3.2) und in Modul B die Behandlung der Nachnamen (siehe Abb. 3.3) statt. Tritt kein OOV auf, wird überprüft, ob bereits nach der Bestätigung gefragt wurde. Ist dies nicht der Fall, wird um Bestätigung gebeten, ansonsten müssen analog zu der Behandlung der einzelnen Namen korrigierte Namen und die Bestätigung bzw. Ablehnung des vollen Namens behandelt werden. Bestätigt der Benutzer den Namen, ist die Strategie erfolgreich gewesen und der Benutzer wird verabschiedet. Wird der Name allerdings abgelehnt, soll der Benutzer seinen Namen wiederho-

[2]Der Fall, dass beide Namen OOV sind, wird verhindert, indem zwei OOV in ein einzelnes OOV umgewandelt werden. Dieses OOV wird dann bereits bei der Vorverarbeitung behandelt, siehe 3.3.1.

len, falls dies noch nicht geschehen ist. Wurde der Name bereits einmal wiederholt, werden die erkannten Namen gelöscht und die Strategie beginnt von vorne, indem der Benutzer erneut nach seinem Namen gefragt wird. Diesmal wird jedoch die Bestätigung des vollen Namens ausgelassen und nur einzelne Namen werden behandelt.

Die Behandlung einzelner Namen tritt auch auf, wenn bereits einer der Namen als bestätigt markiert ist. Dabei wird zuerst der Vorname und anschließend der Nachname verarbeitet (Module A und B). Wird in Modul A kein Move ausgewählt, ist der Vorname bereits bestätigt und Modul B wird gestartet. Wird auch hier kein Move ausgewählt, wurde auch bereits der Nachname bestätigt. Somit ist die Strategie erfolgreich abgeschlossen und der Benutzer wird verabschiedet.

3.3.3. Dialogstrategie für kurzen Dialog

Um diese Strategie kurz zu halten, wird der Vorname und der Nachname je maximal einmal wiederholt und maximal einmal buchstabiert. Realisiert wird dies durch die Variablen *FNspelled* für die Buchstabierung und *FNrepeated* für das Wiederholen. Wurde ein Name wiederholt und buchstabiert und trotzdem nicht erkannt, wird der Benutzer noch einmal nach dem Namen gefragt ohne den Namen bestätigen zu lassen, um den Eindruck zu erwecken das System hätte den Namen verstanden. Wurde in diesem Fall wenigstens ein Name erkannt, wird am Ende des Dialogs der Benutzer mit einem „unbekannt"-Label gespeichert. Beim nächsten Dialog erkennt dadurch das System, dass der Benutzer bereits mit dem System gesprochen hat und versucht nun, nur noch den fehlenden Namen zu erkennen.

Wie die Strategie für einen langen Dialog ist auch diese in folgende drei Fälle unterteilt:

1. Bisher wurde nur der Vorname genannt (Abb. 3.5).
2. Bisher wurde nur der Nachname genannt (Abb. 3.6).
3. Es wurden bereits Vor- und Nachname genannt (Abb. 3.8).

Die ersten zwei Fälle sind analog, deshalb wird nur der erste beschrieben. Sie können auftreten, wenn auf die Frage des Systems nach dem Namen des Benutzers nur ein Name genannt wird. Durch das Konzept der impliziten Bestätigungen, das in dieser Strategie teilweise realisiert werden soll, entsteht eine weitaus komplexere Strategie als in 3.3.2.

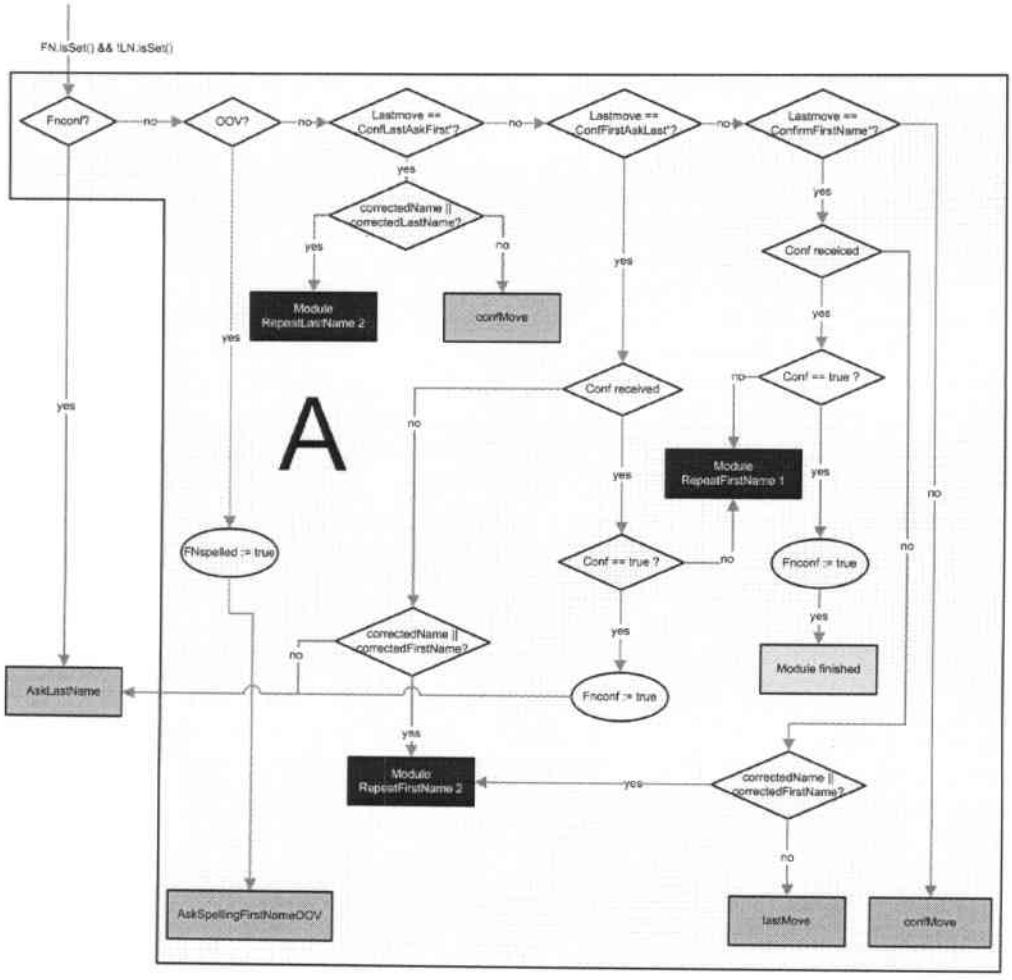

Abbildung 3.5.: Strategie für kurzen Dialog: Behandlung von Vornamen

Fälle 1 und 2 Zu Beginn der Verarbeitung wird überprüft, ob der Vorname schon als bestätigt markiert ist, da dann die Verarbeitung des Vornamens übersprungen und der Nachname verarbeitet wird. Ist der Vorname jedoch unbestätigt und es wurde ein OOV erkannt, wird der Benutzer gebeten seinen Vornamen zu buchstabieren und *FNspelled* wird auf *true* gesetzt. Wurde kein OOV erkannt, gibt es vier Fälle:

- *Lastmove* == *ConfLastAskFirst*
  In der letzten Äußerung des Systems wurde die Hypothese des Nachnamens genannt und nach dem Vornamen gefragt, was das Konzept der indirekten Bestätigung repräsentiert. Korrigiert der Benutzer die Namenshypothese, wird das Modul *RepeatLastName 2* aufge-

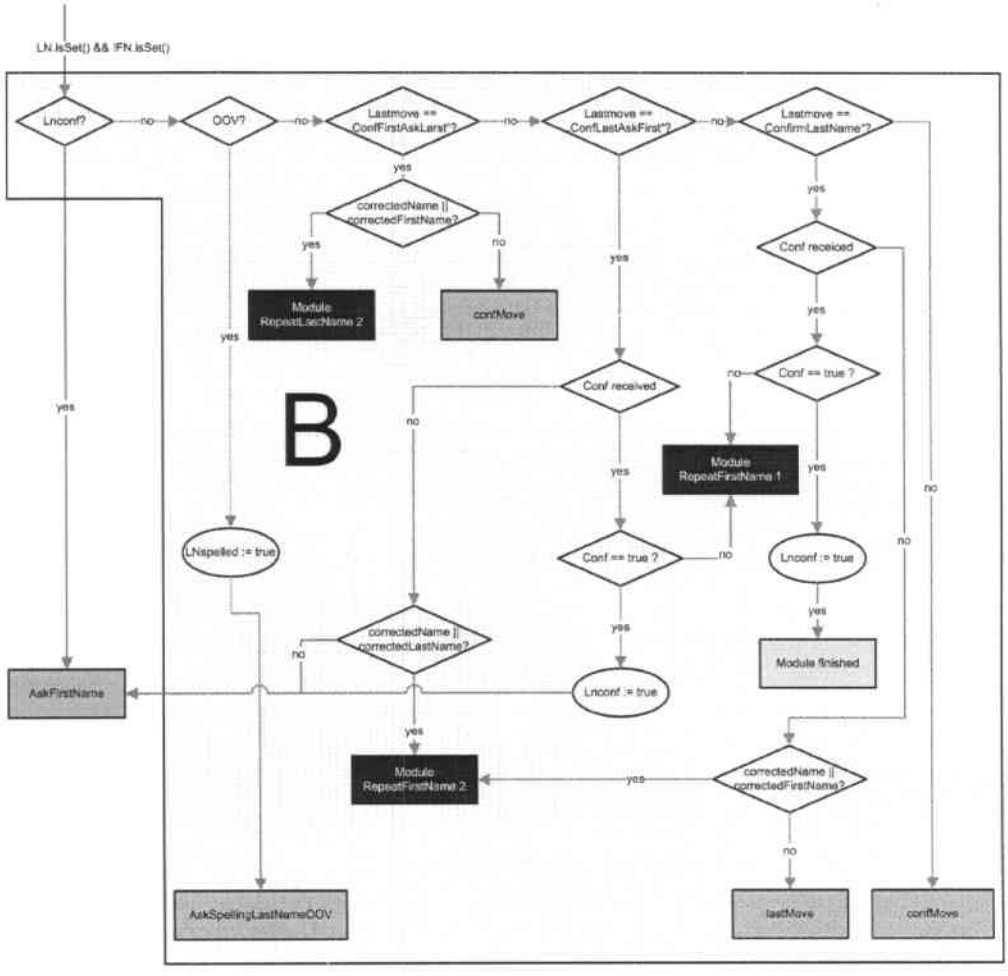

Abbildung 3.6.: Strategie für kurzen Dialog: Behandlung von Nachnamen

rufen. Ansonsten wird als nächstes nach einer Bestätigung gefragt (*confMove*). Je nachdem, ob bereits ein Name bestätigt wurde oder nicht handelt es sich bei diesem Move entweder um *ConfFirstAskLast*, wenn der Nachname noch nicht bestätigt ist, oder ansonsten um *ConfirmFirstName*. Bei *ConfFirstAskLast* wird eine Ausgabe erzeugt, in der der Vorname noch einmal genannt und anschließend nach dem Nachnamen gefragt wird, um eine implizite Bestätigung zu realisieren.

- *Lastmove == ConfFirstAskLast*
  Der vorangegangene Move war *ConfFirstAskLast*. Hat der Benutzer seinen Vornamen bestätigt, wird dieser als bestätigt markiert und

erneut nach dem Nachnamen gefragt. Bei der Ablehnung des Vornamens wird das Modul *RepeatFirstName 1* aufgerufen. Wenn der Benutzer weder den Namen bestätigt noch abgelehnt, statt dessen aber die Hypothese des Systems verbessert hat, wird das Modul *RepeatFirstName 2* aufgerufen. Diese Repeat-Module werden später im Detail erklärt. Wurde keine der beschriebenen Eingaben erhalten, wird der Vorname als bestätigt markiert und der Benutzer wird nach seinem Nachnamen gefragt.

- *Lastmove == ConfirmFirstName*
  In diesem Fall wurde bei der letzten Äußerung des Systems nach der expliziten Bestätigung des Vornamens gefragt. Bestätigt der Benutzer den Vornamen, wird dieser als solcher markiert. Da diese explizite Bestätigung nur auftreten kann, wenn der Nachname bereits bestätigt wurde, wird an dieser Stelle das Modul verlassen. Wurde der Vorname jedoch abgelehnt, wird das Modul *RepeatFirstName 1* aufgerufen. Sollte der Benutzer den Namen nicht bestätigt und nicht abgelehnt haben, wird die vorangegangene Äußerung wiederholt. Hat jedoch der Benutzer hat die Namenshypothese korrigiert, wird das Modul *RepeatFirstName 2* aufgerufen.

- *sonst*
  Wurde in der vorangegangenen Äußerung des Systems versucht, weder den Vornamen noch den Namen implizit zu bestätigen, wird als nächstes nach einer Bestätigung gemäß *confMove* gefragt.

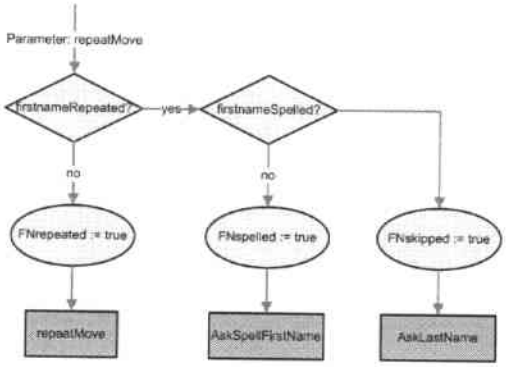

Abbildung 3.7.: Strategie für kurzen Dialog: Repeat-Modul für Vornamen

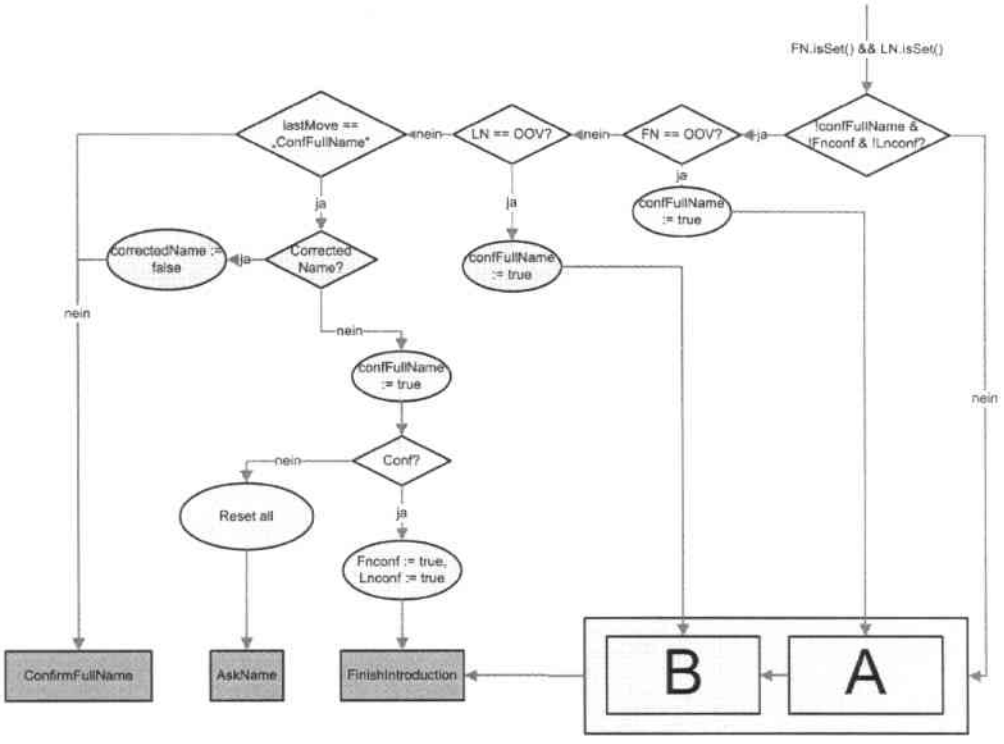

Abbildung 3.8.: Strategie für kurzen Dialog: Behandlung von Vor- und Nachnamen

Fall 3 Der dritte Fall funktioniert bei der Strategie für einen kurzen Dialog fast genau so wie bei der Strategie für einen langen Dialog. Einzig die Wiederholung des vollen Namens, falls dieser vom Benutzer als falsch erklärt wurde, entfällt.

Die Module *FirstNameRepeat* und *LastNameRepeat* Abhängig vom bisherigen Dialogverlauf sind verschiedene Behandlungen im Falle einer Wiederholung des Namens notwendig. Aus diesem Grund ist diese Behandlung in zwei Modulen gekapselt, die analog für Vor- und Nachnamen funktionieren. Im Folgenden wird deshalb nur das Modul *FirstNameRepeat* erklärt (siehe Abb. 3.7). Die Varianten 1 und 2 unterscheiden sich nur durch einen Aufrufparameter. Wurde ein Name weder buchstabiert noch wiederholt, wird der Benutzer zuerst gebeten, den Namen zu wiederholen (Variante 1) oder eine Bestätigung auszuführen (Variante 2). Dabei wird *FNrepeated* auf *true* gesetzt. Wurde der Name bereits wiederholt, soll der Benutzer den Namen buchstabieren. Auch hier wird *FNspelled* auf *true*

gesetzt. Wurde der Name trotz Wiederholens und Buchstabierens immer noch nicht richtig erkannt, wird der Benutzer noch ein letztes Mal nach seinem Vornamen gefragt und *FNskipped* auf *true* gesetzt um eine weitere Verarbeitung des Vornamens zu verhindern.

4. Experimente

Zur Durchführung der Experimente wurde ein System verwendet, das sich aus mehreren Teilkomponenten zusammen setzt. In diesem Kapitel sollen diese Teilkomponenten und ihre Interaktion erklärt werden. Des weiteren wird der Ablauf der Experimente und die Rahmenbedingungen, denen die Experimente unterlagen, beschrieben.

4.1. Beschreibung des Testsystems

Das System besteht aus einem Stereokamerasystem mit schwenkbarem Kopf, jeweils einem Nah- und Fernsprechmikrophon und Lautsprechern. Diese Komponenten sind mit einem Computer verbunden, auf dem die Verarbeitung der Audio- und Videodaten und die Steuerung des Dialogs durchgeführt werden. Zum Steuern wird der Dialogmanager Tapas [Hol08] verwendet. Auf dessen Grundlage hat Felix Putze für seine Diplomarbeit [Put08] einen modularen Ansatz einer Dialogstrategie auf Grundlage eines Entwurfes von Hartwig Holzapfel [HW07] implementiert. Der Namenlerndialog wird als ein Modul des Systems aufgerufen. Die Spracherkennung wird mit Janus realisiert und die Sprachausgabe mit der Software Swift.

Zur Personenidentifikation werden FaceID und VoiceID herangezogen. In [Gro08] wird von Philipp Große ein Verfahren vorgestellt, in dem Modalitäten zu einer gemeinsamen Multimodalen ID vereinigt werden und für die Hypothesen eine Konfidenz berechnet wird. Aufgrund dieses Verfahrens wird in diesem System die Personenidentifikation durchgeführt.

4.1.1. IslEnquirer

Ziel des IslEnquirer ist, mittels Dialogen soziale Informationen über Personen zu sammeln, die am ISL arbeiten [Put08]. Die gewonnenen Informationen werden genutzt, um soziale Netzwerke, wie beispielsweise Arbeitsgruppen, und die Rollen, die die Personen in den Netzwerken einnehmen, zu erkennen. Die resultierende interne Struktur des Instituts wird auf ei-

ne Internetseite abgebildet. Dabei werden Änderungen, die während der Laufzeit des Systems entstehen, dynamisch übernommen.

Beim Entwurf der Architektur des IslEnquirers wurde strikt darauf geachtet, einzelne, sinnvoll zusammenhängende Dialogteile zu indentifizieren und diese zu kapseln. Dadurch entstand eine modulare Struktur mit mehreren Dialogmodulen. Jedes Modul beinhaltet einen eigenen Teil des Dialoges und die Übergänge zwischen den einzelnen Modulen sind mittles eines deterministischen endlichen Automaten als globale Strategie realisiert. Die Eingabe wird von jedem Modul selbst interpretiert. Auf dieser Interpretation und auf dem Diskurs basierend wird zusätzlich der nächste Move in den Modulen ausgewählt. Außer der Moveauswahl in den einzelnen Modulen gibt es noch eine globale Auswahl, die modulunabhängig ist und bei der Moves immer ausgewählt werden können. Ein Beispiel dafür wäre die Aufforderung des Benutzers, die letzte Ausgabe zu wiederholen. Alle Module können auf ein gemeinsames Slotmodell zugreifen, worüber der intermodulare Austausch von Daten statt findet. Die Module des Originalsystems sind: Idle (nichts tun), Begrüßung/Aufgabeninformation, Namenlernen, Registrierung neuer Benutzer, soziale Benutzermodellierung, Mitarbeiter-/Konzeptname lernen.

Als Ergebnis entstand ein robustes System, dem es möglich ist, komplett autonom zu agieren. Dabei hat es die Fähigkeit, sein zukünftiges Verhalten zu optimieren, in dem es Informationen aus vergangenen Dialogen speichert. Um Datenkonsistenz zu gewährleisten muss in einem solchen Fall das Speichern sofort erfolgen. Dies wird mittels einer MySQL-Datenbank realisiert, die darüber hinaus eine einfache Anzeige und Änderung der Daten ermöglicht.

Die Interaktion mit verschiedenen Benutzern wird in Sessions aufgeteilt. Dabei entspricht eine Session der Interaktion mit genau einem Benutzer. Um dies modellieren zu können, wird ein Session-Modell verwendet, das den Zustand der Sessions und der Transitionen zwischen diesen Zuständen behandelt. Dabei wird bei einer Benachrichtigung, dass eine Session begonnen hat, die Aufnahme gestartet und mit der Personenidentifikation begonnen, bis eine Nachricht eintrifft, dass die Session beendet ist. Dies kann beispielsweise geschehen, wenn eine bestimmte Zeit lang keine Eingabe vom Benutzer erkannt wurde und demnach der Benutzer wohl kein Interesse hat, mit dem System zu sprechen. Eine neue Session wird ausgelöst, wenn entweder der Personentracker eine Person erkannt hat oder eine Spracheingabe erkannt wurde. Um das Auslösen einer neuen Session direkt nach dem Beenden des Dialoges zu verhindern, wird dies für einige Sekunden unterdrückt. Dadurch ist es dem Benutzer möglich, unbehel-

ligt zu gehen. Außerdem kann in dieser Zeit das System die neue Person trainieren und andere interne Operationen ausführen, die nicht online geschehen können.

Der IslEnquirer wurde als flexible Architektur entworfen. Dadurch ist es möglich, Module auszutauschen oder wegzulassen. Einzig der endliche Automat, der für die Zustandsübergänge zuständig ist, muss angepasst werden. Bei den Experimenten für diese Studienarbeit wurde von dieser Fähigkeit Gebrauch gemacht und nur das Idle-Modul aus dem ursprünglichen System übernommen. Die Module zur Begrüßung und zum Namenlernen wurden zusammengelegt und ersetzt, die restlichen Module wurden nicht verwendet.

4.1.2. Multimodale PersonID

Für die Personenidentifikation wurden Merkmale eines Gesichtserkenners und einer Stimmerkennung fusioniert. Dies wurde in [Gro08] vorgestellt. Ziele dabei waren die multimodale Fusion der einzelnen Modalitäten sowie ein Konfidenzmaß für das fusionierte Ergebnis zu finden. Für die Berechnung wurde dabei das System in sechs Teilsysteme aufgeteilt und auf den einzelnen Ebenen jeweils eine n-besten-Liste mit zugehöriger Konfidenz erstellt.

Konfidenz bedeutet hierbei ein Maß für die Zuverlässigkeit einer Klassifikationsaussage. Große weist darauf hin, dass Merkmale, die zur Berechnung einer Konfidenz verwendbar sind, je nach Klassifikationsaufgabe unterschiedlich sein können. In dieser Aufgabe gibt es zwei Arten von Merkmalen: Merkmale, die abhängig von der Modalität und Merkmale, die unabhängig von der Modalität anwendbar sind. Beispiele hierfür sind die Entropie der n-besten-Liste und die Differenz der beiden höchsten Scores. Beide finden in [Gro08] Anwendung; zusätzlich werden noch weitere modalitätenunabhängige Merkmale verwendet. Um aus den ausgewählten Merkmalen ein Konfidenzmaß zu berechnen, verwendet Große das Verfahren der logistischen Regression.

Bei der Hypothesenfusion wird das Konfidenzmaß erneut verwendet. Es dient zur Gewichtung der einzelnen Modalitäten. So werden die n-besten-Listen durch ihre Konfidenz gewichtet und für die nächste Ebene aufsummiert. Bei der multimodalen Fusion wird das Problem, dass die Listen aus verschiedenen Merkmalen bestehen, mittels eines Normalisierungsverfahrens gelöst.

Roboter:	Hello.
Benutzer:	Hi.
Roboter:	My name is Robbi Robinson. What is your name?
Benutzer:	My name is Hartwig.
Roboter:	So you're first name is Hartwig. What is your last name?
Benutzer:	My last name is Holzapfel.
Roboter:	Sorry, I have not understood your name. You can help me by spelling your name. Please spell your last name.
Benutzer:	H o l z a p f e l
Roboter:	So your name is Hartwig Holzapfel?
Benutzer:	Yes
Roboter:	Hi Hartwig holzapfel! It's a pleasure to meet you. Unfortunately I have not much time right now, but I hope to get to know you better another time. Good bye.
Benutzer:	Good bye.

Abbildung 4.1.: Beispiel eines Dialoges mit impliziter Bestätigung

4.2. Durchführung der Experimente

Um das entstandene System zu testen, wurden insgesamt 40 Dialoge von 10 verschiedenen Personen durchgeführt. Dabei wurde bei jeder Person abwechselnd die Strategie für den kurzen Dialog und die Strategie für den langen Dialog verwendet. Außerdem wurde bei der Hälfte der Personen mit der Strategie für den kurzen und bei der anderen Hälfte mit der Strategie für den langen Dialog begonnen, um eine Voreingenommenheit der Benutzer auszugleichen. Auf diese Weise wurden der kurze und der lange Dialog jeweils 20 Mal durchgeführt. Die Probanden setzten sich aus Studenten und Mitarbeitern der Universität Karlsruhe (TH) zusammen, wobei alle im Alter zwischen 20 und 30 Jahren waren.

Ein Beispiel für einen Dialog zwischen Roboter und Benutzer ist in Abbildung 4.1 zu sehen. Angewendet wird die Strategie für den kurzen Dialog, was durch die implizite Bestätigung erkennbar ist. Weiter hat der Roboter das erste Mal den Nachnamen nicht verstanden und statt dessen ein *OOV* detektiert, wodurch der Benutzer gebeten wird, den Namen zu buchstabieren.

Um später Informationen wie die Wortfehlerrate (*Word Error Rate*, WER) und die durchschnittliche Dialoglänge zu ermitteln, wurde bei allen Dialogen die Daten des Dialogmanagers und die Audio- und Videodaten

aufgezeichnet. Außerdem musste jeder Proband nach jedem Dialog einen Fragebogen ausfüllen, um eine subjektive Bewertung der Dialoge durchzuführen. Der Fragebogen wurde dabei durch eine Auswahl der Aussagen des SASSI-Fragebogens [HG00] erstellt und dahinter liegende Aussagen wie in [MSBK07] beschrieben ermittelt. Die ersten vier Dimensionen wurden dabei in aufsteigender Reihenfolge mit „Akzeptanz", „Transparenz", „Interaktionseffizienz" und „Kognitiver Anspruch" benannt. Aufgrund der Korellationsfaktoren, die die einzelnen Aussagen mit den vier Dimensionen haben, wurden 21 Aussagen ausgewählt und in den Fragebogen eingefügt. In Zusammenarbeit mit einer Gruppe von Soziologen wurden die Aussagen in Fragen umformuliert um Voreingenommenheit bei den Probanden zu vermeiden. Die Fragen konnten mit einer siebenwertigen Skala beantwortet werden. Der Fragebogen befindet sich im Anhang A.

5. Evaluation

Bei der Evaluation des Dialogs wurden sowohl objektive Kriterien wie Wortfehlerrate (*Word Error Rate*, WER) und Dialogerfolgsrate als auch subjektive Kriterien wie in 4.2 beschrieben verwendet.

5.1. Objektive Bewertung

Zu objektiven Bewertung wurden die WER und die Erfolgsrate ermittelt. Das Ergebnis der Auswertung ist in 5.1 zu sehen. Interessant an den Ergebnissen ist, dass kein wirklicher Unterschied zwischen der Strategie für den langen und der für den kurzen Dialog erkennbar ist. Bemerkenswert ist weiter, dass die Wortfehlerraten insgesamt sehr hoch sind. Diese schlechten Werte haben mehrere Ursachen. Zum einen sind ca. 50% der Fehler auf fehlerhafte Erkennung bei Namen zurückzuführen. Außerdem haben die Benutzer insgesamt nur sehr wenig gesagt. Durchschnittlich wurden pro Aussagen (pro Satz) nur etwas mehr als zwei Wörter gesagt (Tabelle 5.2. Dies ist auch eine der Ursachen für die schlechte Namenserkennung. Benutzer haben häufig nur ihren Namen genannt statt beispielsweise noch „my name is" davor zu setzen. Dadurch hat der Spracherkenner wesentlich mehr Wahlmöglichkeiten. Ein weiterer Grund für die schlechte Namenserkennung ist in der Diskriminanz zwischen englischem System und deutschen Benutzern begründet. Das englische System erkennt aufgrund der internen Phonem-zu-Graphem-Regeln Namen mit deutscher Aussprache sehr schlecht, insbesondere Nachnamen.

Betrachtet man allerdings die Dialoge chronologisch pro Benutzer, wie es in Tabelle 5.3 zu sehen ist, stellt man fest, dass bei dem jeweils ers-

	WER	**#Wörter**	**Erfolgsrate**
alle Dialoge	37,3%	953	71,8%
lange Dialoge	36,4%	489	70,0%
kurze Dialoge	38,1%	464	73,7%

Tabelle 5.1.: Wortfehlerraten der Dialoge (kurz/lang)

	Wörter/Dialog	Sätze/Dialog	Wörter/Satz
alle Dialoge	24,4	10,5	2,3
lange Dialoge	23,2	10,8	2,1
kurze Dialoge	25,7	10,1	2,5

Tabelle 5.2.: Dialoglänge

ten Dialog, den ein Testbenutzer mit dem System durchgeführt hat, die Erfolgsrate und die Wortfehlerrate sich mit der Anzahl der Dialoge, die ein Benutzer mit dem System geführt hat, kontinuierlich verbessert. Dies zeigt, dass Benutzer, die das System nicht kennen, Schwierigkeiten mit dem System haben. Außerdem deutet die Verbesserung der Werte mit fortschreitender Zeit darauf hin, dass sich die Personen dem System anpassen um so die Erfolgswahrscheinlichkeit zu verbessern. Der Benutzer passt sich dem System an und nicht, wie es eigentlich wünschenswert wäre, umgekehrt. Einzig bei den jeweils dritten Dialogen ist ein Ausreißer in der WER zu sehen. Der Grund dafür ist die Namenserkennung, die hier zufällig besonders schlecht war: Es wurden nur 25% der genannten Namen richtig erkannt.

	WER	#Wörter	Erfolgsrate
erste Dialoge	43%	221	44,4%
zweite Dialoge	35,1%	282	60,0%
dritte Dialoge	43,6%	273	80,0%
letzte Dialoge	23,7%	177	100,0%

Tabelle 5.3.: Wortfehlerraten der Dialoge (chronologisch)

5.2. Subjektive Bewertung

Bei der Auswertung der Fragebögen wurden die Aussagen mit Werten von -3 bis +3 bewertet:

-3	-2	-1	0	+1	+2	+3

Die dahinter liegenden Aussagen, die durch die PCA zustande kommen, wurden, wie in Tabelle 5.4 zu sehen ist, bewertet. Der lange Dialog hat dabei in allen vier Kategorien den kurzen Dialog geschlagen. Auch hier liegen die Werte im Bereich von -3 bis +3, wobei ein höherer Wert auch

immer eine bessere Bewertung darstellt. Um die Ursachen für das bessere Abschneiden des langen Dialogs zu ergründen, werden einzelne Aussagen betrachtet, bei denen die Bewertung besonders unterschiedlich ausgefallen ist.

Für die Akzeptanz sind die Aussagen „Das Arbeiten mit dem Roboter machte Spaß." (lang: 0,65; kurz: 0,45) und „Die Interaktion mit dem Roboter war nicht frustrierend" (lang: 0,5; kurz: 0,15) interessant. Anscheinend war der Dialog, der der Strategie für einen kurzen Dialog folgt, deutlich frustrierender. Als Grund hierfür wurde die implizite Bestätigung, die im kurzen Dialog verwendet wird, identifiziert. Dieses Konzept hat sich auch auf andere Kategorien ausgewirkt.

Betrachtet man die für die Transparenz wichtigen Aussagen „Der Benutzer wusste immer, was er zum Roboter sagen konnte." (lang: 1,7; kurz: 1,35) und „Der Roboter hat die Angaben des Benutzers gut verstanden." (lang: 0,55; kurz: -0,15) sowie in der Kategorie „Kognitiver Anspruch" die Aussage „Der Benutzer konnte dem Gesprächsverlauf immer folgen." (lang: 2,1; kurz: 1,6) stellt man fest, dass das Konzept der impliziten Bestätigung für den Benutzer nicht leicht zu durchschauen ist und dadurch die Bewertung stark beeinflusst hat.

Für die Interaktionseffizient ist die Aussage „Die Reaktionsgeschwindigkeit war angemessen" (lang: 0,65; kurz: -0,1) interessant. Fast alle Probanden haben die Reaktionsgeschwindigkeit des Systems als zu lang spezifiziert. Bei den kurzen Dialogen wurde sie sogar negativ bewertet, was überraschend ist, da bei allen Dialogen das gleiche System verwendet wurde und somit auch die Reaktionsgeschwindigkeit gleich sein müsste. Als mögliche Erklärung könnte man erneut die Frustration der Benutzer heranziehen.

	lang	**kurz**
Akzeptanz	0,60	0,47
Transparenz	0,37	0,16
Interaktionseffizienz	0,45	0,28
Kognitiver Anspruch	0,60	0,47

Tabelle 5.4.: Subjektive Bewertung

Eine weitere interessante Aussage losgelöst von den Kategorien ist „Die Gesprächsdauer war angemessen." (lang: 0,8; kurz: 0,7). Bedauerlicherweise haben auch in diesem Punkt die langen Dialoge besser abgeschnitten. Bei näherer Betrachtung ist dies nicht verwunderlich, da die durchschnitt-

liche Dialoglänge sich in beiden Dialogen nicht besonders unterscheidet (siehe Tabelle 5.2). Dies bestätigt leider die ursprüngliche Idee nicht, dass der kurze Dialog wirklich kürzer sein sollte als der lange Dialog. Daraus kann man ableiten, dass trotz des wesentlich komplexeren Aufbaus der Strategie für den kurzen Dialog (Abb. 3.5) die Dialoglänge dadurch nicht wesentlich reduziert wird.

Betrachtet man die subjektiven Bewertungen jedoch getrennt für verschiedene Benutzergruppen, entsteht ein anderes Bild. Aufgeteilt in Benutzer, die das System kennen, und Benutzer, die das System noch nicht kennen, stellt man fest, dass bei der ersten Gruppe (Tabelle 5.5 rechts) die kurze Strategie deutlich besser abschneidet als die lange Strategie. Bei der zweiten Gruppe liegen wie schon zuvor die langen Dialoge vorne (Tabelle 5.5 links). Dieses Resultat deutet darauf hin, dass für Benutzer, die in der Interaktion mit einem Roboter oder einem ähnlichen System nicht geübt sind, eine geradlinige, einfache Strategie am Besten funktioniert. Benutzer hingegen, die bereits mehrmals mit einem solchen System interagiert haben, bevorzugen eine komplexere, abwechslungsreichere Strategie.

	System unbekannt		System bekannt	
	lang	**kurz**	**lang**	**kurz**
Akzeptanz	0,37	0,03	1,15	1,48
Transparenz	0,13	-0,29	0,91	1,23
Interaktionseffizienz	0,32	-0,03	0,73	1
Kognitiver Anspruch	0,43	0,06	0,99	1,4

Tabelle 5.5.: Subjektive Bewertung nach Benutzergruppen

6. Zusammenfassung & Ausblick

In dieser Arbeit wurden zum Lernen von Vor- und Nachnamen zwei verschiedene Dialogstrategien entworfen und evaluiert. Dabei wurde das Problem der Entscheidung, ob ein Name Vor- oder Nachname ist, mit mehreren Maßnahmen behoben. Zum Einen werden Informationen aus den Dialogen, zum Anderen Wahrscheinlichkeiten verwendet. Ein weitere Herausforderung war die Verbesserung der Erkennungsleistung. Es wurde versucht, dies mittels einer hierarchischen Datenbank zu erreichen. Das hat auch in vielen Fällen funktioniert, konnte jedoch nicht den erwarteten Erfolg erzielen, wie die Wortfehlerraten belegen. An dieser Stelle kann in weiteren Arbeiten noch angesetzt werden.

Ziel der verschiedenen Strategien war, die Dialogdauer unterschiedlich lang zu halten. Wie die Auswertung der Dialoge gezeigt hat, wurde dieses Ziel nicht erreicht. Daraus ist zu schließen, dass eine aufwendige und komplexe Strategie bezüglich der Dialoglänge keinen Vorteil bringt. Unterschiede bei der Erfolgsrate wurden auch nicht deutlich, bei beiden Strategien waren diese in einem Bereich, mit dem man zufrieden sein kann.

Die Eignung verschiedener Dialogstrategien für unterschiedliche Benutzergruppen stellt die zentrale Erkenntnis der Auswertung dar. Gezeigt wurde, dass ungeübte Benutzer einfache, geradlinige, gut funktionierende Dialoge ohne Besonderheiten bevorzugen. Dies wurde durch die Strategie für einen langen Dialog realisiert. Benutzern hingegen, die schon viele Male mit einem natürlichsprachigen Dialogsystem gearbeitet haben, bietet diese Strategie zu wenig Abwechslung. Durch die Strategie für einen kurzen Dialog wurde mehr Abwechslung in den Dialog gebracht, weshalb sie von geübten Benutzer bevorzugt wurde.

Für die Zukunft bietet eine genauere Evaluation weitere Erkenntnisse über die Qualität des Systems, insbesondere, wenn Personen, die Englisch als Muttersprache haben, zu den Testbenutzern gehören.

Fragebogen zur Untersuchung „Mensch-Roboter Kooperation“

1. Wie war das Arbeiten mit dem Roboter?

☐ ☐ ☐ ☐ ☐ ☐ ☐

es machte überhaupt keinen Spaß — neutral — es machte sehr viel Spaß

2. Wie reagierte der Roboter?

☐ ☐ ☐ ☐ ☐ ☐ ☐

völlig unerwartet — mittel — immer wie erwartet

3. Wie bewertest du die Kommunikation mit dem Roboter?

☐ ☐ ☐ ☐ ☐ ☐ ☐

ich konnte dem Gesprächsverlauf nie folgen — mittel — ich konnte dem Gesprächsverlauf immer folgen

4. Wie hast du mit dem Roboter kommuniziert?

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr künstlich — neutral — sehr natürlich

5. Wer hatte meistens die Kontrolle über den Kommunikationsverlauf?

☐ ☐ ☐ ☐ ☐ ☐ ☐

Roboter — beide gleichviel — ich

6. Wie war dein Gesamteindruck der Interaktion?

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr schlecht — mittel — sehr gut

7. Wie fühltest du dich während der Kommunikation mit dem Roboter?

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr angespannt — neutral — sehr entspannt

8. Wie empfandest du die Gesprächsdauer?

☐ ☐ ☐ ☐ ☐ ☐ ☐

absolut unangemessen — neutral — absolut angemessen

8.1 Falls (bei Frage 8) eher unangemessen:

☐ zu lang ☐ zu kurz

9. Wusstest du was du zu dem Roboter sagen konntest?

☐ ☐ ☐ ☐ ☐ ☐ ☐

überhaupt nicht — mittel — ja, zu jeder Zeit

10. Welches Maß an Konzentration verlangte die Interaktion mit dem Roboter?

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr wenig — mittel — sehr viel

Bitte wenden!

11. Wie hat der Roboter deine Angaben verstanden?

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr schlecht — mittel — sehr gut

11.1 (Bitte nur an Tag 2 ausfüllen) Hat dich der Roboter wieder erkannt?

Nein ☐ ☐ Ja

12. Wie bewertest du den Roboter insgesamt im Bezug auf seine Funktionalität?

eher schlecht ☐ ☐ eher gut

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr schlecht — mittel — sehr gut

13. Wie fandest du deine Interaktion mit dem Roboter?

sehr unangenehm	☐ ☐ ☐ ☐ ☐ ☐ ☐	sehr angenehm
sehr anstrengend	☐ ☐ ☐ ☐ ☐ ☐ ☐	gar nicht anstrengend
sehr langweilig	☐ ☐ ☐ ☐ ☐ ☐ ☐	sehr kurzweilig
sehr monoton	☐ ☐ ☐ ☐ ☐ ☐ ☐	sehr abwechslungsreich
sehr frustrierend	☐ ☐ ☐ ☐ ☐ ☐ ☐	überhaupt nicht frustrierend

14. Der Roboter macht eher...

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr viele Fehler — mittel — gar keine Fehler

14.1 Falls (bei Frage 14) eher viele Fehler: An wem lagen die Fehler in der Interaktion?

☐ ☐ ☐ ☐ ☐ ☐ ☐

immer am Roboter — beide — immer am Benutzer

15. Der Roboter ist...

sehr unfreundlich	☐ ☐ ☐ ☐ ☐ ☐ ☐	sehr freundlich
sehr schwer zu benutzen	☐ ☐ ☐ ☐ ☐ ☐ ☐	sehr leicht zu benutzen
sehr unflexibel	☐ ☐ ☐ ☐ ☐ ☐ ☐	sehr flexibel
akustisch sehr unverständlich	☐ ☐ ☐ ☐ ☐ ☐ ☐	akustisch sehr verständlich

16. Wie war die Reaktionsgeschwindigkeit des Systems?

☐ ☐ ☐ ☐ ☐ ☐ ☐

sehr unangemessen — mittel — sehr angemessen

16.1 Falls (bei Frage 16) eher unangemessen:

☐ ☐

zu schnell — zu langsam

17. Würdest du dich in Zukunft wieder mit dem Roboter unterhalten?

☐ ☐ ☐ ☐ ☐ ☐ ☐

auf keinen Fall — neutral — auf jeden Fall

Vielen Dank!

Literaturverzeichnis

[BMS+05] BURGHART, C. ; MIKUT, R. ; STIEFELHAGEN, R. ; ASFOUR, T. ; HOLZAPFEL, H. ; STEINHAUS, P. ; DILLMANN, R.: A cognitive architecture for a humanoid robot: a first approach. In: *5th IEEE-RAS International Conference on Humanoid Robots*, 2005, S. 357–362

[Car92] CARPENTER, Bob: *The logic of typed feature structures*. New York, NY, USA : Cambridge University Press, 1992. – ISBN 0–521–41932–8

[Gro08] GROSSE, Philipp: *Konfidenzbasierte multimodale Fusion von Audio und Video zur Personenidentifikation*, Universität Karlsruhe (TH), Studienarbeit, 2008

[HG00] HONE, Kate S. ; GRAHAM, Robert: Towards a tool for the Subjective Assessment of Speech System Interfaces (SASSI). In: *Nat. Lang. Eng.* 6 (2000), Nr. 3-4, S. 287–303. – ISSN 1351–3249

[Hol08] HOLZAPFEL, Hartwig: A Dialogue Manager for Multimodal Human-Robot Interaction and Learning of a Humanoid Robot. In: *Industrial Robots Journal* 35-6 (2008)

[HW07] HOLZAPFEL, Hartwig ; WAIBEL, Alex: Behavior Models for Learning and Receptionist Dialogs. In: *Proceedings of Interspeech*, 2007

[IT03] ITU-T, Rec. P.: *Subjective Quality Evaluation of Telephone Services Based on Spoken Dialogue Systems*. International Telecommunication Union, Geneva, 2003

[MSBK07] MÖLLER, Sebastian ; SMEELE, Paula ; BOLAND, Heleen ; KREBBER, Jan: Evaluating spoken dialogue systems according to de-facto standards: A case study. In: *Computer Speech & Language* 21 (2007), S. 26–53

[Put08] PUTZE, Felix: *Social User Model Acquisition through Network Analysis and Interactive Learning*, Universität Karlsruhe (TH), Diplomarbeit, 2008

[WKL00] WALKER, Marilyn ; KAMM, Candace ; LITMAN, Diane: Towards developing general models of usability with PARADISE. In: *Nat. Lang. Eng.* 6 (2000), Nr. 3-4, S. 363–377. – ISSN 1351–3249

[WLKA97] WALKER, Marilyn A. ; LITMAN, Diane J. ; KAMM, Candace A. ; ABELLA, Alicia: PARADISE: a framework for evaluating spoken dialogue agents. In: *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*. Morristown, NJ, USA : Association for Computational Linguistics, 1997, S. 271–280

[WLKA98] WALKER, M. ; LITMAN, D. ; KAMM, C. ; ABELLA, A.: Evaluating spoken dialogue agents with PARADISE: Two case studies. In: *Computer Speech & Language* 12 (1998), S. 317–347

[Zie07] ZIESEMER, Stefan: *Namenserkennung bekannter und unbekannter Namen*, Universität Karlsruhe (TH), Diplomarbeit, 2007