Universität Karlsruhe (TH)

# Verfolgen der Sprecheraufmerksamkeit mit Hilfe der Ausgabe des Spracherkenners

Studienarbeit am Institut für Logik, Komplexität und Deduktionssysteme
Prof. Dr. A. Waibel
Fakultät für Informatik
Universität Karlsruhe (TH)

von

cand. inform.
**Michael David Katzenmaier**

Betreuer:
Prof. Dr. A. Waibel
Dr. I. Rogina

Tag der Anmeldung: 1. Dezember 2002
Tag der Abgabe: 28. Februar 2003

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 28. Februar 2003

# Inhaltsverzeichnis

# 1. Einleitung

Wir sind heute in einem Zeitalter, in welchem Maschinen den Menschen immer mehr Arbeiten abnehmen können. Tatsächlich wird uns durch Computer, computergesteuerte Maschinen oder Roboter bereits vieles erleichtert.

Die dahinter stehenden Systeme könnten von ihren immanenten Möglichkeiten her gesehen häufig noch viel effektiver genutzt werden. Ein Grund, warum dies oft nicht gelingt, ist, dass sie nicht in der gewünschten Einfachheit steuerbar sind. Wäre es möglich, sich mit jemanden zu unterhalten und mitten im Gespräch einer Maschine per Sprache Befehle zu erteilen, dann wäre dies sehr komfortabel und benutzerfreundlich. Dies hätte vermutlich eine breite Akzeptanz des Benutzerkreises zur Folge.

Im Alltag könnte so z.B. dem „intelligenten Raum" gesagt werden, welche Musik gespielt werden soll oder welches Bild das LCD an der Wand anzeigen soll.

Wäre dies möglich, so hätten es z.B. auch Tetraplegiker viel einfacher, denn sie könnten in ihrem Alltag vieles per Sprache steuern; nicht zuletzt ihren Rollstuhl.

Um dies Realität werden zu lassen, muss die Maschine erkennen können, wann sie gemeint ist und wann nicht. Dieses Problem soll in vorliegender Arbeit angegangen werden.

## 1.1 Zielsetzung der Arbeit

Ziel ist es demnach, Mensch-Maschine Dialoge von Mensch-Mensch Dialogen zu unterscheiden. Bei einem Wechsel von einerseits Befehlen und andererseits allgemeiner Konversation, ist es nicht immer offensichtlich, wen der Sprecher im Moment meint. Die Befehle könnten grundsätzlich an anwesende Menschen gestellt oder an die entsprechende Maschine gerichtet sein. Für die sprachgesteuerte Maschine ist allerdings wichtig, dass sie erkennt, wann sie gemeint ist bzw. wann nicht.

Die Aufgabe der Studienarbeit besteht darin, diese Differenzierung zu finden. So soll eine Lösung gefunden werden, wie in einem Gespräch unterschieden werden kann, ob eine Person oder eine Maschine angesprochen ist. Die intelligente Maschine

ist hier ein Oberbegriff für beispielsweise einen Roboter oder einen „intelligenten Raum“ usw. Diesen Maschinen soll per Sprache die Bedienung der Stereoanlage, des Telefons, des Herdes, der Heizung, der Klimaanlage, der Rollläden, der Beleuchtung etc. aufgetragen werden. Dabei muss die Maschine erkennen können, ob es sich bei der Aussage um einen Mensch-Mensch Dialog – im folgenden Konversation genannt – oder um einen Mensch-Maschine Dialog – im folgenden Befehl oder Befehle genannt – handelt.

Schwierig wird es allerdings für die Maschine, wenn im Mensch-Mensch Dialog nur erzählt wird, dass man einen Roboter hat, der verschiedene Dinge tun kann. Dann ist dem Roboter nämlich nichts aufgetragen worden – im Gegensatz zu einem direkten Auftrag. Es muss in solchen Situationen von der Maschine unterschieden werden können, ob die Person A der Person B z.B. nur mitteilt, dass sie einen Roboter hat, der Bier holt, wenn man ihm das sagt, oder, ob dem Roboter gesagt wird: „Hole Bier!“. Der Roboter muss demzufolge erkennen, dass er im ersten Fall nicht gemeint ist. Diese Diskriminierung bzw. Differenzierung ist Gegenstand der Problemstellung.

Beginn und Voraussetzung der Problemanalyse sind demzufolge die gesprochenen Worte. Sie liefern die Datenbasis, die entweder als reines akustisches Signal verwertet wird oder als deren Transkript für die weitere Analyse dient. Es wird also entweder von dem Gesprochenen ausgegangen oder von dessen Transkripten. Teilweise wird in dieser Arbeit von den Transkripten ausgegangen, um festzustellen wie gut das Ergebnis mit einem perfekten Erkenner wäre.

Die Studienarbeit konzentriert sich auch nur auf die Modalität des gesprochenen Wortes. Andere Modalitäten, wie z.B. die Blickrichtung, werden hier nicht betrachtet. Es ist sinnvoller, die Problemstellung aus ökonomischen Gründen zunächst allein durch Sprachmodalität anzugehen, um damit eine konkrete und präzis erfassbare Problemlösung zu erhalten. Je genauer das Ergebnis beschrieben werden kann, desto hilfreicher ist es dann für die beabsichtigte Erweiterung bei Hinzunahme anderer Modalitäten. Für die Erweiterung - in einer späteren Arbeit (Diplomarbeit) - können dann die optimierten Ansätze mit den besten Ergebnissen benutzt werden. Sie dienen dann als gute Ausgangsbasis für weitere Verbesserungen.

Bei der alleinigen Betrachtung der Sprachmodalität, nimmt auch nur diese Einfluss auf das Ergebnis und so ist die Ursache einer Änderung der Klassifikationsgenauigkeit konkreter feststellbar. Die Sprachmodalität ist von allen Modalitäten diejenige, welche die meisten für die Befehlsdetektierung relevanten Informationen beinhaltet. So könnte allein durch die Information über den Standort des Sprechers nicht festgestellt werden, wer der Adressat der Botschaft ist. Aber die Botschaft für sich alleine genommen drückt dies meist schon aus. Doch wäre die Information über den Ort des Senders der Botschaft hilfreich, um das Erkennen der Botschaft beim Adressaten zu verbessern. Mit den Informationen der Blickrichtung oder bestimmter Gesten ist zwar auch eine Befehlsdetektierung denkbar, aber es ist anzunehmen, dass sie größere Fehlerraten hätten. Dies müsste jedoch experimentell festgestellt werden.

Zunächst sollen hiermit Erfahrungen gesammelt werden: Kann allein durch Hören ein brauchbares Ergebnis erzielt werden? Inwiefern kann es auch noch ohne weitere Modalitäten optimiert werden?

Um dieses Ziel zu erreichen, ist folgendes Vorgehen geplant:

1. Auswahl von Merkmalen der Hypothesen des Spracherkenners, welche zur Klassifikation hilfreich sein könnten.
2. Trainieren und Testen verschiedener Klassifikatoren:
   (a) Bayes-Klassifikator
   (b) Trennhyperebene
   (c) Multi-Layer-Perceptron

In dieser Arbeit werden verschiedene Merkmale und die oben genannten Klassifikatoren ausgewertet.

Befehle zu detektieren, ist selbstverständlich nur ein Stück auf dem Weg, intelligente Maschinen immer mehr in unseren Alltag mit einzubeziehen. So sind an der Universität Karlsruhe in der Forschung mit humanoiden Robotern gleich mehrere Institute beteiligt [sfb]. Auch die Forschung in Zusammenhang mit „intelligenten Räumen" ist noch längst nicht abgeschlossen [fame].

# 2. Grundlagen

## 2.1 Die zur Entscheidung herangezogenen Merkmale

Als sinnvoll und brauchbar bei der Entscheidungsfindung können verschiedene Merkmale herangezogen werden. So ist die Satzlänge ein Kriterium, das einen Hinweis liefern kann, ob es sich um einen Befehl oder um eine Konversation handelt. Ein Befehl ist im allgemeinen eher kurz und prägnant gegenüber einem Satz oder Satzgefüge in einer Konversation. Selbstverständlich reicht dies allein nicht als Entscheidungsmerkmal.

Ein weiteres Merkmal ist die Perplexität. Perplexitäten können unter verschiedenen Sprachmodellen berechnet werden und diese wiederum unter verschiedenen Vergleichskriterien zur Entscheidungsfindung herangezogen werden. Dazu mehr im Kapitel Entscheidungsmethoden.

Im Folgenden ist mit Befehl ein Befehl an eine Maschine gemeint. Befehle an Menschen sollen hier generell zur Konversation gezählt werden.

### 2.1.1 Perplexität

Die Entropie ($H_s$) einer ergotischen Quelle $s$, welche die Symbolfolge $W$ produziert, errechnet sich folgendermaßen:

$$H_s = \lim_{N \to \infty} \frac{1}{N} \, log P_s(W)$$

Wird anstatt einer unendlich langen Ausgabe, die Ausgabe der Länge $N$ betrachet, dann gilt für große $N$ ungefähr:

$$H_s \approx \frac{1}{N} \, log P_s(W)$$

Nun wird als ergotische Quelle ein Sprachmodell $s$ betrachet. Dann kann die Entropie $H_s$ einer Wortfolge $W$ bzgl. dieses Sprachmodells mit Hilfe der Wahrscheinlichkeit

$P_s(W)$ dieser Wortfolge errechnet werden. Die Wahrscheinlichkeit und Entropie derselben Wortfolge ist je nach betrachtetem Sprachmodell unterschiedlich, hängt also von diesem ab. Wird die Länge der Wortfolge – sie kann als Ausgabe des Sprachmodells betrachtet werden – wieder $N$ genannt, dann ergibt sich mit obiger Approximation und folgender Gleichung für die Perplexität:

$$PP_s = 2^{H_s}$$

diese Gleichung:

$$PP_s = P_s(W)^{-\frac{1}{N}}$$

Je größer die Entropie eines Satzes oder je größer die Perplexität, desto schwieriger ist es auch, den Satz zu erkennen. Eine Erkennung mit Perplexität $PP_s$ kann verglichen werden mit dem Erkennen von Sprache mit einem gleichverteilten Vokabular der Größe $PP_s$.

Hier werden die Perplexitäten zum einen unter einem Sprachmodell errechnet, das auf Befehlen basiert. Genauer gesagt, handelt es sich meist um Anfragen an ein Navigationssystem [vodi]. So z.B. Anfragen nach einer Straße oder ganz allgemein einem bestimmten Ort oder Stelle, – und sei es das nächste Restaurant.

Zum anderen werden die Perplexitäten unter einem Sprachmodell errechnet, welches überwiegend Mensch-Mensch Dialoge enthält, in diesem Fall Terminabsprachen [verb].

Diese beiden Sprachmodelle waren von den zur Verfügung stehenden Sprachmodellen am geeignetsten. Anweisungen an ein Navigationssystem z.B. sind im Allgemeinen in imperativer Form, wie nach der deutschen Grammatik die meisten Befehle. Folglich verspricht das Modellieren von Befehlen durch das Erstellen eines Sprachmodells auf diesen Daten Erfolg. Hilfreich beim Erkennen spontaner Sprache ist sicher ein Sprachmodell, welches auch auf spontaner Sprache erzeugt wurde. Und dies ist bei Terminabsprachen der Fall.

Liegt der Wert der Perplexität vom Befehlsmodell herrührend unter dem Wert der Perplexität des Konversationsmodells, dann kann davon ausgegangen werden, dass mit größerer Wahrscheinlichkeit ein Befehl vorliegt. Soweit die Hypothese.

### 2.1.2 Satzlänge

Die Satzlänge ist ein Kriterium, das dank der deutschen Grammatik sehr hilfreich sein kann. Ein Befehlssatz hat in den meisten Fällen eine bestimmte Länge. Dieser ist im Vergleich zu Erzählsätzen eher kurz formuliert, – „Hol mir ein Bier!", „Fege die Straße!"„Bring mir die Zeitung!".

Ein Mensch-Mensch Dialogsatz hingegen ist meist länger, so z.B. beim Informationsaustausch in der allgemeinen Konversation. Nur selten ist er kürzer, wie z.B. bei einer Antwort oder einer Bestätigung – „Ja!", „Mhm!" etc.

Demnach wird erwartet, dass die Satzlängen bei Befehlen in einem eher kurzen Intervall liegen und die Satzlängen der Konversationen zum größten Teil darüber und nur manchmal darunter liegen. Überschneidungen sind dabei zu erwarten.

### 2.1.3 Korrelation zwischen Hypothesen zweier Erkenner

Ein weiteres Merkmal ist die Korrelation der Ausgaben zweier verschiedener Erkenner. Einer der beiden Erkenner ist auf eine kontextunabhängige Grammatik (CFG) aufgebaut, wobei die Grammatik mit der Intention geschrieben ist, dass sie nur Befehle erkennt. Allgemeine Sätze sollen nicht durch die Grammatik ableitbar sein. – Dieser Erkenner wird nachfolgend als CFG-Erkenner bezeichnet.

Der andere Erkenner basiert auf einem Sprachmodell, das anstatt auf einer CFG auf $N$-Grammen aufgebaut ist. Dieser sollte eine allgemeine Erkennung durchführen; im Idealfall also Befehle und Konversationen erkennen können. Durch diesen Erkenner sollten dann einerseits Befehle an die Maschine und andererseits auch allgemeine Sätze einer beliebigen Konversation erkannt werden.

Idee und Forderung sind nun, dass untersuchte Befehle sowohl der auf einer kontextfreien Grammatik als auch der auf $N$-Grammen basierende Erkenner erkennen sollte.

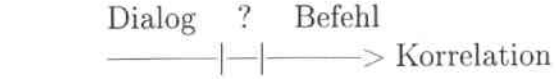
Demnach müsste eine Korrelation der beiden Hypothesen der jeweiligen Erkenner einen Treffer ergeben und der Befehl als solcher eindeutig detektiert werden. Wobei Treffer bedeutet, dass eine sehr hohe Korrelation, d.h. Übereinstimmung eintritt. Im hypothetischen Idealfall wird dann bei beiden Erkennern der gleiche Satz/Befehl erkannt.

Handelt es sich allerdings bei dem untersuchten Satz um eine Konversation, dann ist zu erwarten, dass die Grammatik nichts Sinnvolles erkennen kann, der andere Erkenner hingegen durchaus etwas Brauchbares liefert. Dies führt zu einer niedrigen Korrelation zwischen den Hypothesen. Damit wäre die Korrelation zwischen beiden Erkennern ein signifikantes Merkmal zur Identifikation der Nachricht.

Jedoch ist der Idealfall, dass beide Erkenner – im Falle eines Befehls – immer das gleiche erkennen, unrealistisch. Auch ist zu erwarten, dass im Falle einer Konversation meistens doch ein gewisser Grad an Korrelation vorhanden ist. So ist zu erwarten, dass für die Korrelation ungenaue Werte anzutreffen sind. Das Problem ist aber, dass deutlich zwischen Befehl und Konversation diskriminiert werden muss. Dies kann durch eine Entscheidungsgrenze durchgeführt werden (s.Abb. 2.1). Jedoch kann es besser sein, in einem bestimmten Intervall die Aussage zu treffen, dass eine genaue Festlegung auf Befehl oder Konversation nicht erfolgen kann. Es ist nämlich denkbar, dass in einem gewissen Bereich keine eindeutige Zuordnung erfolgen kann. Somit würde die Grenze zu einem Intervall. Oberhalb und unterhalb dieses Intervalls/Bereichs ist die Entscheidung dann wieder eindeutig. (s.Abb. 2.2)

Abb. 2.1


Abb. 2.2




Denkbar wäre dann, solche Spezialfälle gesondert zu behandeln und sinnvolle Zusatzinformationen für Entscheidungen hinzuzunehmen. Zum Beispiel könnte in einem

solchen Fall mehr Kontext einbezogen werden und danach erst entschieden werden, ob ein Befehl vorliegt. Wurde im Satz vorher so etwas wie „ Home-Multi-Media-Terminal an" gesagt, dann handelt es sich eher um einen Befehl. Werden hingegen dort Eigennamen angesprochen oder Satzteile, wie „da habe ich gesagt", dann war es wohl eher kein Befehl.

Auch könnten in solchen Fällen andere Entscheidungskriterien (siehe dazu der Abschnitt 2.2 Entscheidungsmethoden) und/oder andere differenzierende Merkmale zu Rate gezogen werden.

### 2.1.4 Parsebarkeit

Wie bereits im Abschnitt „Korrelation zwischen Hypothesen zweier Erkenner" erwähnt, ist die Grammatik des CFG-Erkenners so geschrieben, dass sie nur Befehle erkennt.

Um dies auszunutzen, kann die Parsebarkeit (oder Ableitbarkeit) durch diese Grammatik als weiteres Merkmal genommen werden. Dabei wird untersucht, ob die Hypothese des $N$-Gramm-Erkenners oder das Transkript – im Falle der Simulation eines perfekten Erkenners – durch die CFG ableitbar ist. Ist ein Ableitungspfad in der Grammatik enhalten, dann liegt Parsebarkeit vor, andernfalls nicht.

Damit ergibt die Parsebarkeit einen digitalen Wert, der als Wahrheitswert gesehen werden kann und im Idealfall bei einem Befehl wahr anzeigt und im anderen Fall unwahr.

## 2.2 Entscheidungsmethoden

In diesem Abschnitt wird auf die theoretischen Hintergründe des methodischen Ansatzes eingegangen. Es werden dabei die Ideen von ihrer mathematischen Seite genauer beleuchtet.

In den Experimenten wird zum einen für die Modellierung der Mensch-Mensch (HH) Dialoge ein Sprachmodell verwendet, welches auf den Daten des Verbmobil-Projekts [verb] erstellt wurde. Diese Daten beinhalten Terminabsprachen zweier Menschen und somit HH-Dialoge. Nachfolgend wird dieses Sprachmodell HH1-Sprachmodell genannt.

Daten des Vodis-Projekts [vodi] werden zum Erstellen zur Modellierung von Befehlen an Maschinen verwendet. Hier handelt es sich um Instruktionen an ein Navigationssystem; folglich um Mensch-Maschine (HM) Dialoge. Dieses Sprachmodell wird im Folgenden HM1-Sprachmodell genannt.

Zum anderen wird für die Modellierung der HH-Dialoge ein Sprachmodell benutzt, das auf den eigenen Transkriptteilen der Konversationen erstellt wurde – nachfolgend HH2-Sprachmodell genannt. Und für die Befehlsmodellierung wird ein Sprachmodell verwendet, das wiederum auf den eigenen Befehlsanteilen der Transkripte erstellt wurde. Dieser wird im Folgenden als HM2-Sprachmodell bezeichnet.

Die Befehle wurden allerdings durch zusätzlich gesammelte Befehle ergänzt, da die ohnehin kleine Datenmenge durch das seltenere Auftreten der Befehle sonst zu klein gewesen wäre.

### 2.2.1 Der wahrscheinlichkeitstheoretische Ansatz

Beim wahrscheinlichkeitstheoretischen Ansatz werden die Wahrscheinlichkeiten einer Äußerung (z.B. „Ein Wasserfall wäre schön." ) unter verschiedenen Sprachmodellen betrachtet, um herauszufinden, ob ein Mensch-Mensch oder ein Mensch-Maschine Dialog stattfindet. Die These ist hierbei die Annahme, dass Befehle innerhalb eines Gesprächs eher an Maschinen gerichtet werden als an Menschen. Dass Befehle nicht immer an Maschinen gerichtet sind, zeigt die Situation einer Mutter, die ihrem Kind etwas befiehlt.

In einer Konversation kommen Imperative seltener vor als in einer Folge von Anweisungen und dies wird in der Sprachmodellierung berücksichtigt. So ergeben die Wahrscheinlichkeiten für ein bestimmtes Wort in der Konversationsmodellierung einen anderen Wert als in der Befehlsmodellierung.

Die a-priori Wahrscheinlichkeit dafür, dass ein bestimmtes Wort gesprochen wird, wird Unigramm genannt. Hingegen wird die bedingte Wahrscheinlichkeit eines bestimmten Wortes unter der Bedingung, dass vorher ein anders bestimmtes Wort gesagt wurde, Bigramm genannt. Die Wahrscheinlichkeit eines Wortes unter der Voraussetzung, dass vorher zwei andere bestimmte Wörter gesagt wurden, wird dann entsprechend Trigramm genannt. Alle solche bedingten Wahrscheinlichkeiten werden mit dem Begriff $N$-Gramme zusammengefasst. Da mit größer werdendem $N$ die $N$-Gramme immer seltener vorkommen, ist die Betrachtung für große $N$ immer weniger hilfreich, zumal auch der Rechenaufwand für die Betrachtung mehrerer verschiedener $N$-Gramme steigt. Bei der Erstellung eines Sprachmodells werden deswegen im Allgemeinen nur Unigramme, Bigramme und Trigramme betrachtet.

Die Wahrscheinlichkeit einer Äußerung $u$ unter der Bedingung, dass ein Mensch-Mensch Dialog $HH$ oder Mensch-Maschine Dialog $HM$ stattfand, wird mit $p(u|x)$ abgekürzt; für $x$ aus der Menge $\{HH, HM\}$. Die Wahrscheinlichkeit ohne Bedingung wird mit $p(u)$ abgekürzt.

Die Hypothese ist nun, dass $p(HM|u)$ größer ist als $p(HH|u)$, falls die Äußerung $u$ ein Befehl ist und kleiner, falls es sich um eine allgemeine Konversation handelt.

Die Wahrscheinlichkeit $p(HM|u)$ der aktuellen Äußerung $u$ kann nicht berechnet, sondern nur mit Hilfe des Sprachmodells geschätzt werden. Da jedoch die Wahrscheinlichkeit $p(u|HM)$ einfacher direkt aus den Merkmalsmessungen zu schätzen ist als die Wahrscheinlichkeit $p(HM|u)$ wird folgende Gleichung (Bayes-Regel) angewandt:

$$p(x|u) = \frac{p(u|x) \cdot p(x)}{p(u)}$$

Da herausgefunden werden soll, welche Dialogart bei der gegebenen Äußerung wahrscheinlicher ist, d.h. welches der beiden Modelle $HH$ oder $HM$ wahrscheinlicher ist, wird über alle $x$ maximiert. Und weil sich außerdem die Äußerung $u$ über den gesamten Zeitraum der Maximumsuche nicht ändert, ist auch die Wahrscheinlichkeit für diese Äußerung $p(u)$ in diesem Zeitraum konstant. Demnach kann das Maximum auch gefunden werden, wenn anstatt über die Formel:

$$\frac{p(u|x) \cdot p(x)}{p(u)}$$

über den Term:

$$p(u|x) \cdot p(x)$$

maximiert wird.

### 2.2.2 Vergleichsmethoden

Beim Vergleich der Perplexitäten werden zwei unterschiedliche Sprachmodelle benutzt, hingegen werden beim Vergleich der beiden Hypothesen auf Korrelation zwei verschiedene Spracherkenner verwendet.

So werden bei der in Zukunft Perplexitätenvergleichsidee (Vgl-PP) genannten Idee die auf zwei verschiedenen Sprachmodellen berechneten Perplexitäten zur Problemlösung genutzt. Diese Entscheidungsmethode wurde zweimal experimentell durchgeführt. Einerseits wurden das HH1- und der HM1-Sprachmodell zur Perplexitätenberechnung verwendet und andererseits das HH2- und HM2-Sprachmodell.

Für jeden (transkribierten) Satz wird die Perplexität unter dem jeweiligen Sprachmodell berechnet. Tragen wir die Werte der Konversationsmodellierung an der $x$-Achse und die Werte der Befehlsmodellierung an $y$-Achse ab, so erhalten wir ein zweidimensionales Diagramm. In diesem zweidimensionalen Merkmalsraum wird dann versucht, die beiden Ereignisse, einerseits ein Konversationssatz und andererseits ein Befehl, voneinander zu trennen. Es wird also versucht, die Menge der Punkte in diesem Merkmalsraum, die von Befehlssätzen kommen, von der Menge der Punkte, die von Konversationssätzen kommen, abzugrenzen. Eine mögliche Vorschrift wäre die Größerrelation. Welche Perplexität kleiner ist, dessen Hypothese oder Voraussetzung bzw. Bedingung wird angenommen. Ist die Perplexität unter einer Befehlsmodellierung kleiner, dann wird gefolgert, es handle sich um einen Befehl. Ist sie größer, dann wird gefolgert, es sei ein Teil der Konversation. Nennen wir die Perplexität unter dem Befehlsmodell $pp_m$ und die unter dem Konversationsmodell $pp_h$, dann sieht die Vorschrift folgendermaßen aus:

$$pp_m < pp_h \implies u = \text{Befehl}\ , \text{sonst}\ u = \text{Konversation}$$

Dies käme in dem zweidimensionalen Diagramm einer Winkelhalbierenden gleich. Oberhalb der Winkelhalbierenden wäre es ein Befehl und unterhalb ein Konversationsteil; so die Hypothese. (Falls an der Ordinate $pp_m$ und an der Abszisse $pp_h$ abgetragen wird.)

Da andere Trennfunktionen wie die Winkelhalbierende bessere Ergebnisse erzielen, soll hier die optimale Funktion mit Hilfe von Trainieren neuronaler Netze gefunden werden. Als Eingabe dieses Netzes dienen dann die Perplexitäten der beiden Sprachmodelle. Siehe Abschnitt „Der neuronale Ansatz".

Es ist auch möglich, die Werte eines weiteren Sprachmodells hinzuzunehmen oder andere Merkmale, so dass mehr als zwei Dimensionen entstehen. Spätestens dann ist es das beste, wenn ein neuronales Netz trainiert wird, um die Klasseneinteilung vorzunehmen. Aber auch schon im zweidimensionalen Fall wird die beste Trennlinie oder Klasseneinteilung dadurch erreicht, dass ein neuronales Netz verwendet wird. Da künstliche neuronale Netze weitgehend bekannt sind und es auch genügend Literatur dazu gibt, wird der interessierte Leser auf die Literatur [Bish95] verwiesen.

In der Hypothesenvergleichsidee (Korrelation als Entscheidungsgrenze) werden ebenfalls zwei Erkenner verwendet. Bei dem einen Erkenner handelt es sich um einen auf dem HH1-Sprachmodell basierenden Spracherkenner. Er basiert also auf einer $N$-Gramm-Grammatik. Dagegen basiert der andere der beiden Erkenner auf einer kontextfreien Grammatik. Es handelt sich um den bereits erwähnten CFG-Erkenner. Der auf den $N$-Grammen basierende Erkenner, dessen Sprachmodell (HH1-Sprachmodell) auch bei der Perplexitätenvergleichs-Idee verwendet wurde, wird nachfolgend auch $N$-Gramm-Erkenner genannt.

Wird, wie oben beschrieben, bei einem $N$-Gramm-Erkenner das vermeintlich Erkannte durch die Wahrscheinlichkeiten der $N$-Gramme beeinflusst, so ist es bei dem CFG-Erkenner die kontextfreie Grammatik. In dem CFG-Erkenner wird also die Akustik nicht durch solche Wahrscheinlichkeiten beeinflusst, sondern durch die Pfade im Ableitungsbaum der CFG. Passt das Gesagte auf einen Pfad im Ableitungsbaum unter den vielen Möglichkeiten der CFG, dann wird dieser Satz genommen und nicht der, der auf keinen Ableitungsbaum passt. Der CFG-Erkenner versucht folglich beim Erkennen einen Ableitungspfad in der Grammatik für das durch die Akustik Erkannte zu finden oder mit anderen Worten einer Hypothese, die ableitbar ist, den Vorrang zu geben.

Das Resultat eines Erkenners hängt nicht nur von der Grammatik selbst ab. So kann die Phonetik, die Qualität der Aufnahme und nicht zuletzt auch die Qualität der Quelle die Ergebnisse beeinflussen. Diese Problematiken sind jedoch nicht Gegenstand der Untersuchungen.

Hier gehen wir zunächst davon aus, dass der Spracherkenner perfekt ist. Wir gehen also davon aus, dass von dem auf einer Grammatik basierende Erkenner der Befehl erkannt wird.

Um das Ergebnis des $N$-Gramm-Erkenners zu verbessern, kann noch versucht werden, die Parameter des Sprachmodells zu verändern. Falls die meisten Äußerungen nicht im Sprachmodell enthalten sind, es somit die Erkennung eher in die falsche Richtung lenkt, ist es sinnvoll das Sprachmodell weniger zu gewichten und mehr die Akustik das Ergebnis beeinflussen zu lassen. So wurden verschiedene Werte für das Sprachmodellgewicht und die Wortübergangsstrafe ausprobiert, um die besten Parameter zu finden.

### 2.2.3 Der neuronale Ansatz

Wie bei der Entscheidungsmethode des Vergleichs der Perplexitäten bereits erwähnt, wurde versucht, die optimale Trennlinie bzw. Trennfunktion zur Befehlsdetektierung mit Hilfe eines neuronalen Netzes zu finden. Mit den ersten Ergebnissen konnte festgestellt werden, dass die Daten sehr nahe an der Linie der Winkelhalbierenden liegen. Nun könnten durch weitere Untersuchungen der Trainingsdaten eine noch bessere Trennfunktion als die Größenrelation (Winkelhalbierenden) gesucht werden. Da für neuronale Netze unter bestimmten Bedingungen nachgewiesen ist, dass mit ihrer Hilfe die beste Trennfunktion gefunden werden kann, wird hier ein solches verwendet.

Als Eingabedaten dienen dem künstlichen neuronalen Netz zunächst die Perplexitäten der Hypothesen der Spracherkenner. Dann wird mit Hilfe der Information, ob es sich um einen Befehl handelt oder nicht, trainiert, bis die optimale Trennlinie

gefunden ist (überwachtes Lernen). Beim Training des Netzes auf den Trainingsdaten wird das Ergebnis auf diesen Daten kontinuierlich besser. Doch irgendwann ist der Punkt erreicht, an dem das Netz die Trainingsdaten zu genau lernt und deshalb ein Generalisieren nicht mehr möglich ist. Dies wird durch ständiges Testen auf den Evalidierungsdaten festgestellt. Wird das Ergebnis auf den Evalidierungsdaten wieder schlechter, dann ist dieser Punkt erreicht und an dieser Stelle die optimale Trennlinie gefunden.

Hier handelt es sich um ein Netzgefüge mit zwei Eingaben und einer Ausgabe. Die Eingaben sind die beiden Perplexitäten wie in der Perplexitätenvergleichsmethode, und die Ausgabe wäre die Kodierung der Aussage des Vorliegens eines Befehls. Also die Aussage, ob es sich um einen Befehl handelt oder nicht. Die Kodierung der Ausgabe kann dann mit einem oder zwei Ausgängen verwirklicht werden. Ein 1/0-Wert wäre zwar als Wahrheitswert (Befehl oder kein Befehl) ausreichend, doch hätte das Netz bei zwei Ausgabewerten eine größere Mächtigkeit. Bei einem wie hier verwendeten Netz, dessen Neuronen mit allen Neuronen der jeweils nächsten Schicht verknüpft sind (fully connected network), kämen so die Verbindungen zum zweiten Ausgabeneuron hinzu. Mit den Verbindungen steigt dann die Komplexität und die Fähigkeit des Netzes, kompliziertere Funktionen zu erlernen. Dazu werden dann aber mehr Trainingsdaten benötigt als im Falle der Kodierung mit nur einem Ausgabewert und mit weniger Verbindungen des Netzes. Es müssen dann die zusätzlichen Gewichtungen der hinzugekommenen Verbindungen ebenfalls trainiert werden.

Da sich die Daten überschneiden und das Netz die Fähigkeit generalisieren zu können, nicht verlieren darf, dürfen nicht zu viele Neuronen im Hidden-Layer gewählt werden. Denn sonst hat das Netz die Fähigkeit, auch sehr komplexe Funktionen zu erlernen, was dazu führen wird, dass es die Trainingsdaten sozusagen „auswendig" lernt. Diese Eigenschaft wird Overfitting genannt und führt dazu, dass das Netz nicht mehr generalisieren kann. Wieviele Neuronen deshalb das beste Ergebnis liefern, müssen die Experimente zeigen. Sinnvoll ist es also aus dem oben genannten Grund, zuerst eine eher kleine Neuronenanzahl zu wählen und dann verschiedene Anzahlen von Neuronen auszuprobieren.

Bei Hinzunahme weiterer Merkmale ergeben sich dann entsprechend mehr Eingabedaten. Durch diese hat das Netz mehr Möglichkeiten, weil es dann die Gewichtung der zusätzlichen Verbindungen zwischen den Neuronen trainieren kann. Auch kann es diejenigen Merkmale, die signifikanter sind, stärker gewichten und das Ergebnis mehr von diesen relevanteren Eingaben abhängig machen.

# 3. Experimente

In diesem Abschnitt sollen die aus den theoretischen Ansätzen erarbeiteten Experimente vorgestellt und genauer beschrieben werden. Im nachfolgenden Kapitel werden dann die gefundenen Ergebnisse protokolliert und diskutiert.

Vom Neuronalen Ansatz ist das beste Ergebnis aller Entscheidungsmethoden zu erwarten, trotzdem ist es interessant, die anderen Methoden als Vergleich heranzuziehen.

## 3.1 Der wahrscheinlichkeitstheoretische Ansatz

Beim wahrscheinlichkeitstheoretischen Ansatz wird untersucht, mit welchem Sprachmodell die Wahrscheinlichkeit der gegebenen Äußerung $u$ größer ist. Also ob $p(HH|u)$ mit dem Konversationsmodell oder ob $p(HM|u)$ mit dem Befehlsmodell größer ist.

Da $p(u|HM)$ – wie im Kapitel Grundlagen erklärt – leichter zu schätzen ist als $p(HM|u)$, wird nicht diese Formel:

$$p(HM|u) > p(HH|u)$$

verwendet, sondern der Vergleich:

$$p(HM)\ p(u|HM) > p(HH)\ p(u|HH)$$

bevorzugt.

Da die Äußerung $u$ selbst keinen konkreten Zahlenwert darstellt und somit auch keine mathematische Grundlage liefert, anhand welcher Mittelwerte errechnet werden könnten, werden brauchbare Merkmale $m_x$ der Äußerung zunächst abstrahiert. Die Merkmale hängen dann von der Bedingung $x \epsilon \{HM, HH\}$ ab. Für die Wahrscheinlichkeiten $p(u|x)$ wird eine Gaußkurve $N(\mu, \Sigma, m_x)$ berechnet. Die Parameter $\mu$ und $\Sigma$ der Normalverteilung werden hierfür aus den Trainingsdaten errechnet. Als Merkmale des jeweiligen Sprachmodells dienen zunächst die berechneten Perplexitäten der Trainingssätze. – So wird der Mittelwert $\mu$ durch einfache Mittelwertsberechnung der Perplexitäten und die Standardabweichung $\Sigma$ durch die Differenz zwischen

dem Mittelwert der quadrierten Werte der Perplexitäten und des Quadrates des Mittelwertes berechnet.

Also:

$$\mu = E(m_x), \quad x \in \{HH, HM\}$$

und

$$\Sigma = E(m_x^2) - E(m_x)^2, \quad x \in \{HH, HM\}$$

Mit den Parametern $\mu$ und $\Sigma$ kann dann $p(u|x)$ mittels der Normalverteilung $N(\mu, \Sigma, m_x)$ geschätzt werden. Wie bereits erwähnt, dienen die anhand der unterschiedlichen Sprachmodelle errechneten Perplexitäten als Merkmale, welche zur Berechnung der Gaußfunktion herangezogen werden. Die Wahrscheinlichkeit $p(\text{„Hole Bier !"}|HH)$ wird demnach mit Hilfe der Normalverteilung berechnet, deren Eingabe die anhand einer Konversationsmodellierung errechneten Perplexität ist. Analog wird die Wahrscheinlichkeit $p(\text{„Hole Bier!"}|HM)$ durch die aus einer Befehlsmodellierung berechneten Perplexität erlangt.

$p(x)$ erhält man durch Zählen der relativen Häufigkeiten:

$$p(x) = \frac{\text{Anzahl } x}{\text{Gesamtanzahl}}, \quad x \in \{HH, HM\}$$

Demzufolge wird die Wahrscheinlichkeit eines Mensch-Maschine Dialogs $p(HM)$ mittels der Anzahl der Befehle durch die Gesamtanzahl aller Äußerungen ermittelt.

Die Befehldetektierung wird dann folgendermaßen vorgenommen: Ist die Wahrscheinlichkeit der gegebenen Äußerung unter dem Befehlsmodell größer, dann wird angenommen, es ist ein Befehl. Im anderen Fall, wenn die Wahrscheinlichkeit kleiner ist als die Wahrscheinlichkeit unter dem Konversationsmodell, dann wird angenommen es ist kein Befehl.

## 3.2 Vergleichsmethoden

### 3.2.1 Vergleich der Perplexitäten

In der ersten Vergleichsmethode wurde die Idee der Perplexitäten unter verschiedenen Sprachmodellen untersucht. Es wurde zum einen ein Sprachmodell auf Konversationen basierend und eines auf Befehlen basierend eingesetzt. Für jeden Satz können dessen Perplexitäten unter dem jeweiligen Sprachmodell errechnet werden. Dies ergibt je Satz zwei Werte, die als Punkte in einem zweidimensionalen Feld erfasst werden können. Siehe Abb. 3.1.

In der $x$-Richtung sind die Werte für das Konversationsmodell aufgetragen und auf der $y$-Richtung die des auf Befehle basierenden Modells.

Ideal wäre es nun, wenn die Punkte, die von Sätzen stammen, die Befehle waren, sich von den Punkten, die von Konversationssätzen stammen, unterscheiden. Unterscheiden in dem Sinn, dass eine Trennlinie, im sehr heuristischen Falle sogar eine Trenngerade, gefunden wird, welche die beiden Klassen, nämlich die der Befehle und die der Konversationssätze, voneinander deutlich abgrenzt. Im Experiment ist leider

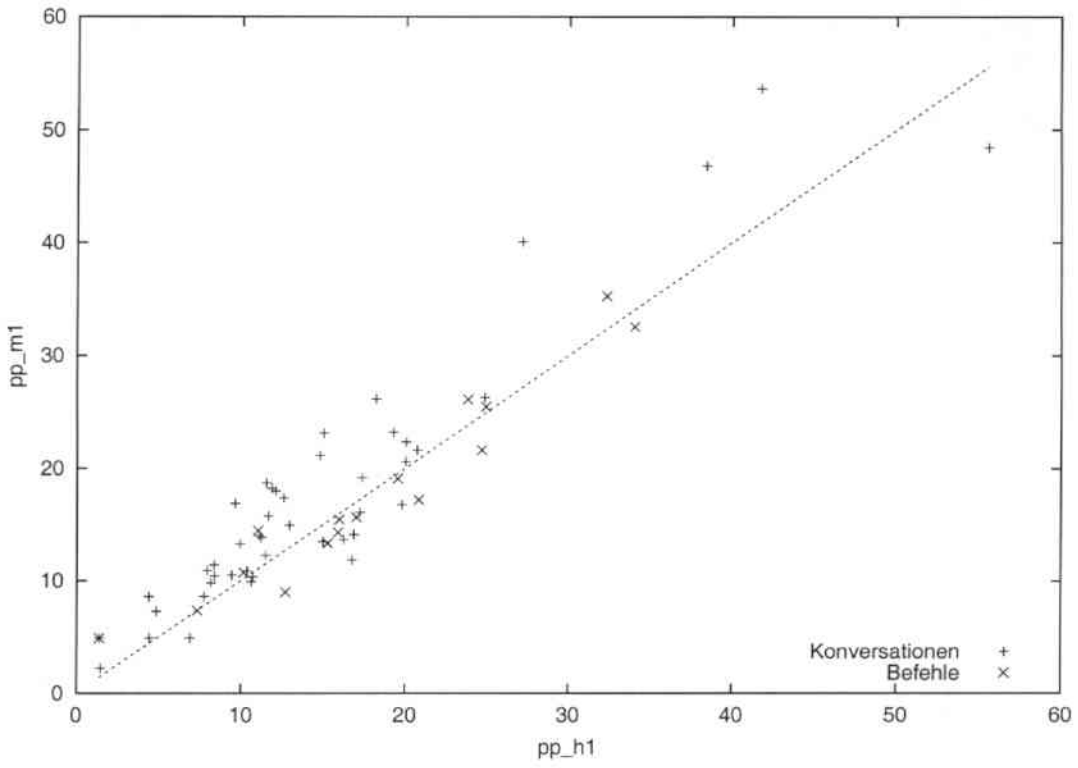



Abbildung 3.1: Diagramm der Perplexitäten (pp_h1/m1 mit HH1-/HM1-Sprachmodell errechnet; vgl. S.19 )

eine Überschneidung der beiden Punktmengen aufgetreten, was zu Fehlern in der Klassifizierung führt.

Andere Funktionen als die Winkelhalbierende werden wohl bessere Ergebnisse liefern. Deswegen soll die optimale Funktion mit Hilfe des Trainierens neuronaler Netze gefunden werden.

Wie in Abb. 3.1 zu erkennen ist, sind die meisten Befehle unterhalb der Winkelhalbierenden und die reinen Konversationen oberhalb. Der Prozentsatz der Befehle, die als solche erkannt werden (Recall), ergab 63%. Nur jeder zweite Befehl wurde auch als solcher erkannt (50% Precision). Es wurden deswegen nicht alle erkannt, weil das Entscheidungskriterium war, 'unterhalb der Winkelhalbierenden zu liegen'.

### 3.2.2 Die Korrelation als Entscheidungsgrenze

Im zweiten Versuch ging es darum, das Ergebnis eines auf einer kontextfreien Grammatik basierenden Erkenners mit dem eines auf $N$-Grammen basierenden Erkenners zu korrelieren. Bei hoher Korrelation wird angenommen es handelt sich um einen Befehl und bei niedriger es handelt sich um keinen Befehl. Die hierbei verwendeten Erkenner sind die oben genannten HH1- und CFG-Erkenner.

Die Erstellung einer kontextfreien Grammatik für die vorliegende Problemstellung erweist sich als nicht ganz einfach. So soll zwar forciert werden, dass vielerlei mögliche Befehle in der Grammatik enthalten sind. Aber dies soll nicht dazu führen, dass in dem Ableitungsbaum ein Pfad enthalten ist, welcher kein Befehl ist. Es könnte auch passieren, dass ein Befehl gar nicht in der Grammatik enthalten ist, da bei der Erstellung der Grammatik an Befehle solcher Art nicht gedacht wurde und kein

Ableitungspfad dafür vorgesehen wurde. Dann kann allerdings ein derartiger Befehl auch nicht als solcher erkannt werden.

Deshalb ist es sinnvoll, die Grammatik mit den Trainingsdaten zu testen oder diese gleich bei der Erstellung heranzuziehen. Im Detail hieße dies, die Befehle aus einem Datensatz zu abstrahieren, und deren Parsebarkeit zu überprüfen. – Analog dazu bei den Konversationen die Nicht-Parsebarkeit.– Ein weiteres Problem stellt die Begrenzung der Datenmenge dar. Selbst wenn riesige Datenmengen gesammelt werden, kommen nie alle möglichen Befehle in den Daten vor.

Jedoch dürfte dies in der Praxis eher weniger ein Problem darstellen, da die Maschine, die den Befehl erkennen und ausführen soll, nur eine begrenzte Möglichkeit an Funktionalität vorweist.

Vorstellbar ist jedoch, dass in naher Zukunft humanoide Roboter eine Vielzahl von Funktionen haben, die schon nicht mehr eingegrenzt werden können. Zwar können die rein mechanischen Abläufe eingegrenzt werden, aber nicht die konkret umsetzbaren Befehle. Denkbar wäre, im Roboter ein Lernmechanismus zu integrieren, der sich auf den jeweiligen Benutzer einstellt. Dies wäre nicht unbedingt von Nachteil, denn es ist gut vorstellbar, dass ein Besitzer eines solchen Roboters oder evtl. auch ein Rollstuhlfahrer, der sein Transportmittel (Rollstuhl) per Sprache steuert, nicht möchte, dass jemand anderes diese Maschine steuern kann. Nicht vorzustellen, was passieren könnte, wenn sich jemand an einem hilflosen Rollstuhlfahrer seinen Spaß erlaubt. Vielmehr sollte gerade dann darauf geachtet werden, dass die Sprachsteuerung der Maschine nur von dem oder den entsprechenden Benutzern möglich ist. Dies ist zwar auch ein interessantes Problem, jedoch nicht Gegenstand dieser Arbeit. Wir konzentrieren uns nur auf das Auffinden von Befehlen an Maschinen innerhalb eines Gesprächs.

## 3.3 Der neuronale Ansatz

Ziel des neuronalen Ansatzes war es, die Trennlinie der Entscheidungsmethode „Vergleich der Perplexitäten" zu verbessern. Das künstliche neuronale Netz gibt zwar keine Trennlinie im Sinne einer Geraden aus, sondern eine wesentlich kompliziertere Funktion. Diese ordnet jede Eingabe einer Entscheidungsklasse (Befehl oder Konversation) zu. Ein Vergleich der Ergebnisse der einzelnen Entscheidungsmethoden kann über die Fehlerraten, Recall- und Precisionwerte vorgenommen werden.

Leider waren die Eingabedaten der beiden unterschiedlichen Erkenner – dem des auf Konversationen basierenden und dem des auf Befehlen basierenden –, nämlich deren Perplexitäten des betrachteten Satzes, nicht signifikant genug. Wegen diesen geringen Unterschieden der Perplexitätswerte, trainierte das Netz gar nicht und gab immer negative Befehlsdetektierung aus. Es entschied sich also nie für einen Befehl, da es dann am wenigsten Fehler machte. Deshalb wurde die Anzahl der Befehle der Anzahl an Konversationen angeglichen, indem in den Trainingsdaten jeder Befehl gleich viermal erschien. Damit das Netz aber nicht in irgendeine Richtung, die evtl. ungünstig wäre, lernt, wird die Reihenfolge der Trainingsdaten zufällig gewählt.

Auch wurde versucht, weitere Merkmale zu finden, die als zusätzliche Eingaben für das Netz dienen. Als sinnvoll erschien die Satzlänge, da sich im allgemeinen Befehl diese doch auf eine bestimmte Wortanzahl beschränkt, sich zumindest im häufigsten Fall im Mittelwert einfindet.

Da die beiden Erkenner (HH1- und CFG-Erkenner) zu schlecht waren – die Erkennungsrate lag jeweils unter 20% –, konnten die Ausgaben der beiden Spracherkenner nicht gut verglichen werden. Deshalb war die Frage entstanden, ob das Merkmal der Korrelation überhaupt mit einzubeziehen ist. Aus dieser Problematik heraus entstand eine neue Idee. Diese sieht vor, nicht die Korrelation als Merkmal zu nehmen, sondern die Tatsache der Parsebarkeit durch die kontextfreie Grammatik des CFG-Erkenners. (Es stellte sich jedoch heraus, dass die Korrelation als Entscheidungsgrenze eine trotz der schlechten Voraussetzungen erfolgreiche Methode war.)

Im Idealfall sollte der Ableitungspfad bei einer Dialogphrase immer leer sein und bei einem Befehl nicht. Letzteres sollte der Fall sein, falls bei der Erstellung der Grammatik an alle möglichen Befehlsarten gedacht wurde. Aber, ob dies in der Praxis immer der Fall ist? Jedoch kann dieser Fall eingeschränkt werden, da er um so unwahrscheinlicher ist, je mehr Daten daraufhin getestet werden. Bis eine solch große Datenmenge angesammelt werden kann, muss evtl. die Grammatik ständig erweitert werden.

In der Praxis ist bei einem intelligenten Roboter die Anzahl der möglichen Befehle jedoch durch die Funktionen des Roboters eingeschränkt. Wenn dem Roboter dann ein Befehl erteilt wird, den er nicht ausführen kann, dann ist es hinfällig, ob er diesen nun erkennt oder nicht. Er kann ihn sowieso nicht ausführen.

Die Parsebarkeit ist eigentlich eine Vereinfachung der Korrelation. Hatte diese nur angegeben, ob der Satz (Hypothese oder Transkript) parsebar ist oder nicht (1 oder 0), so kann die Korrelation alle Werte dazwischen annehmen. Bei identischen Sätzen, dem größten Maß an Übereinstimmung der Hypothesen, ist die Korrelation eins. Wird die Übereinstimmung immer kleiner, dann geht auch die Korrelation gegen Null. Die Parsebarkeit hingegen ist nur Eins, wenn eine Ableitung aus der CFG möglich ist – dies sollte der Fall sein, wenn die Korrelation Eins ist –, sonst Null. Sie ist also ein gröberes Maß.

Da das Ergebnis so gut wie möglich sein sollte, werden alle denkbaren, sinnvollen Eingaben für das Netz genutzt. Letztendlich standen die Perplexitäten unter den verschiedenen Sprachmodellen, die Satzlänge, die Parsebarkeit und die Korrelation als Eingaben für das Netz zur Verfügung.

Um feststellen zu können, wie sehr das Ergebnis von der Qualität des Erkenners abhängt, – wie auch schon bei anderen Entscheidungsmethoden durchgeführt –, können auch hier nicht die Hypothesen der Erkenner benutzt werden, sondern die Transkripte des Gesagten. So wird das ganze Experiment nochmals anstatt mit den Hypothesen mit den Transkripten durchgeführt.

Hierbei können für das Netz im Prinzip die gleichen Merkmale als Eingaben dienen. Bis auf die Korrelation können alle Merkmale analog zu den Hypothesen erlangt werden. Die Korrelation zwischen den beiden Hypothesen der Erkenner ist im Falle der Betrachtung der Hypothesen sinnvoll. Jedoch wenn die Transkripte betrachtet werden sollen, kann zwar für die Hypothese des $N$-Gramm-Erkenners das Transkript des Gesagten verwendet werden, aber was soll anstatt der Hypothese des CFG-Erkenners verwendet werden? Würde hier ebenfalls das Transkript genommen, wäre die Korrelation immer 100%. Die Korrelation könnte also nicht mehr zwischen Befehl und Konversation differenzieren.

Bei der Betrachtung der Transkripte wurde hier dann zum einen das Transkript selbst genommen und zum anderen die Hypothese des CFG-Erkenners; die Korrelation wurde dann zwischen Transkript und der Hypothese des CFG-Erkenners ermittelt.

# 4. Ergebnis und Diskussion

Die verwendeten Daten stammen von Aufnahmen aus einem Gesprächsszenario zweier männlicher Personen und einer Maschine. In den meisten Fällen handelte es sich bei der Maschine um einen Roboter, seltener um eine Stereoanlage, einem sogenannten Home-Multi-Media-Terminal, einem intelligenten Raum oder Haus. Die beiden Personen wurden jeweils separat mit Nahbesprechungsmikrofonen aufgenommen.

Die Maschine war jedoch imaginär, da es noch keine humanoiden Roboter oder intelligente Maschinen mit einer derartigen Funktionalität gibt. Es ist hingegen gerade Grund und Ziel dieser Arbeit in Zukunft das Betreiben solcher Maschinen zu ermöglichen; jedenfalls ansatzweise bezüglich der Befehlsdetektierung. Folglich existieren auch nur die Aufnahmen der Äußerungen der Personen und keine Aufnahmen der Antworten der Maschine. Als Gesprächsgrundlage diente ein Besuch-Szenario. Person A besuchte Person B, die ihr einen intelligenten Raum, einen Roboter usw. vorstellte. Ein solches Szenario wurde dreimal mit den gleichen Personen mit wechselndem Gastgeber durchgeführt, so dass sechs Aufnahmen als Datenbasis der Befehlsdetektierung zur Verfügung standen.

Die nachfolgenden Experimente beziehen sich alle auf diese Daten. Fünf der Aufnahmen wurden zum Training der Entscheidungsmethode verwendet, falls ein Training erforderlich war, und die sechste Aufnahme diente für die Testreihe.

Die Merkmale werden folgendermaßen abgekürzt:

- pp_h1 : Perplexität mit dem HH1-Sprachmodell errechnet
- pp_m1: Perplexität mit dem HM1-Sprachmodell errechnet
- pp_h2 : Perplexität mit dem HH2-Sprachmodell errechnet
- pp_m2: Perplexität mit dem HM2-Sprachmodell errechnet
- satzl : Die Satzlänge
- parsb: Die Parsebarkeit

Ob die Hypothesen oder die Transkripte verwendet wurden, drücken jeweils folgende Abkürzungen aus:

- hy: Hypothesen
- tr: Transkripte

## 4.1 Der wahrscheinlichkeitstheoretische Ansatz

Die Ergebnisse (s.Abb. 4.1-3) haben gezeigt, dass beim Wahrscheinlichkeitstheoretischen Ansatz mit den vier Merkmalen, pp_h1, satzl, pp_h2 und pp_m2, das beste Ergebnis erreicht werden konnte. Es entstand ein Recall von 25% und eine Precision von 100% für die Transkripte und ein Recall von 38% und eine Precision von 75% für die Hypothesen. Da aber insgesamt wenig Befehle vorkamen, ist die Fehlerrate relativ niedrig (jeweils 20%).

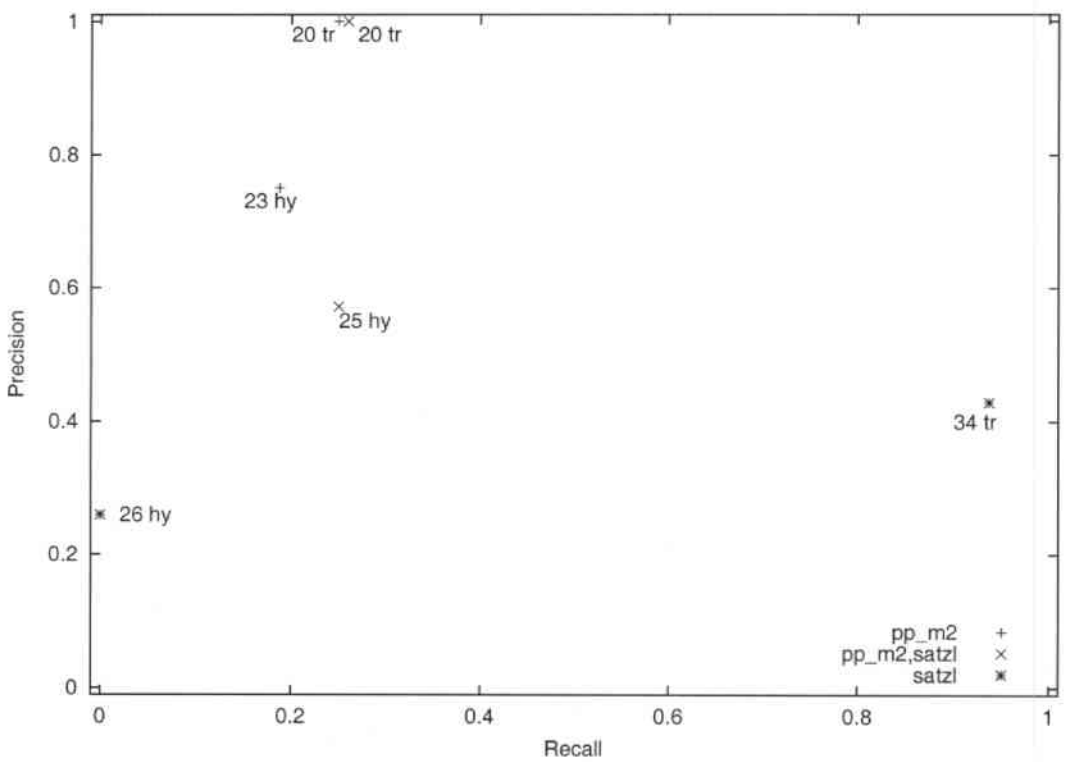



Abbildung 4.1: Wahrscheinlichkeitstheoretischer Ansatz mit 2 und 3 Merkmalen (immer pp_h2 verwendet; zusätzliche Merkmale angegeben)

Das Entscheidungskriterium der Befehlsdetektierung war, welche Wahrscheinlichkeit – für Befehl oder für Konversation – größer ist (vgl. Abs. 2.2.1 und 3.1). Nach einem einfachen Umformungsschritt kann folgende Ungleichung als Kriterium dienen:

$$p(u|HH) > \; p(u|HM)\,\frac{p(HM)}{p(HH)}$$

Anstatt mit dem Faktor $\frac{p(HM)}{p(HH)}$ zu multiplizieren, könnte auch mit einem sogenannten Fudgefaktor f multipliziert werden. Dies wäre zwar die mathematisch gesehen

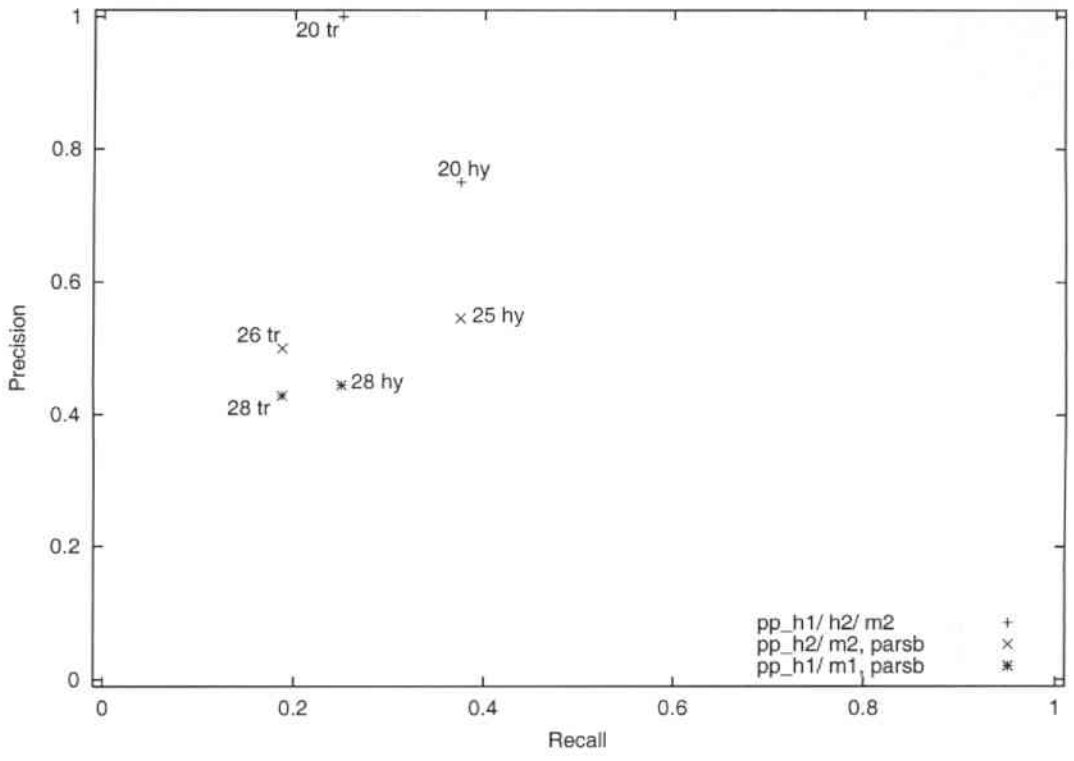



Abbildung 4.2: Wahrscheinlichkeitstheoretischer Ansatz mit 4 Merkmalen (immer satzl verwendet; zusätzliche angegeben)

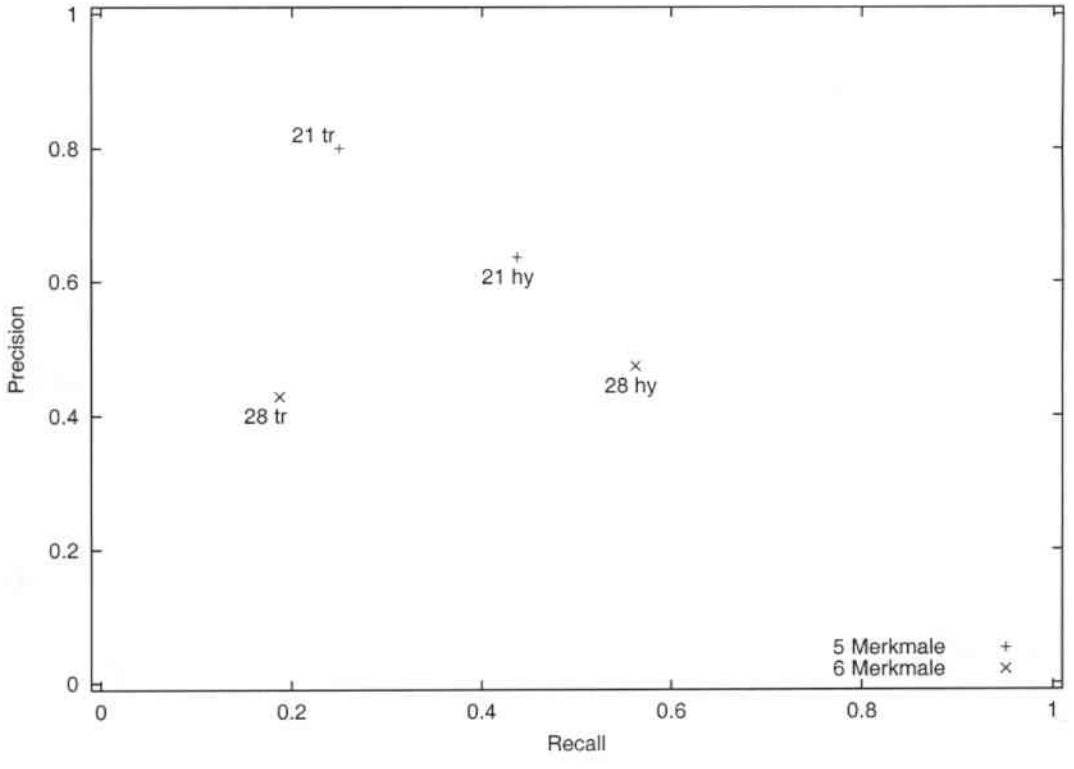



Abbildung 4.3: Wahrscheinlichkeitstheoretischer Ansatz mit 5 und 6 Merkmalen (alle pp und satzl; mit 6 Merkmalen zusätzlich parsb)

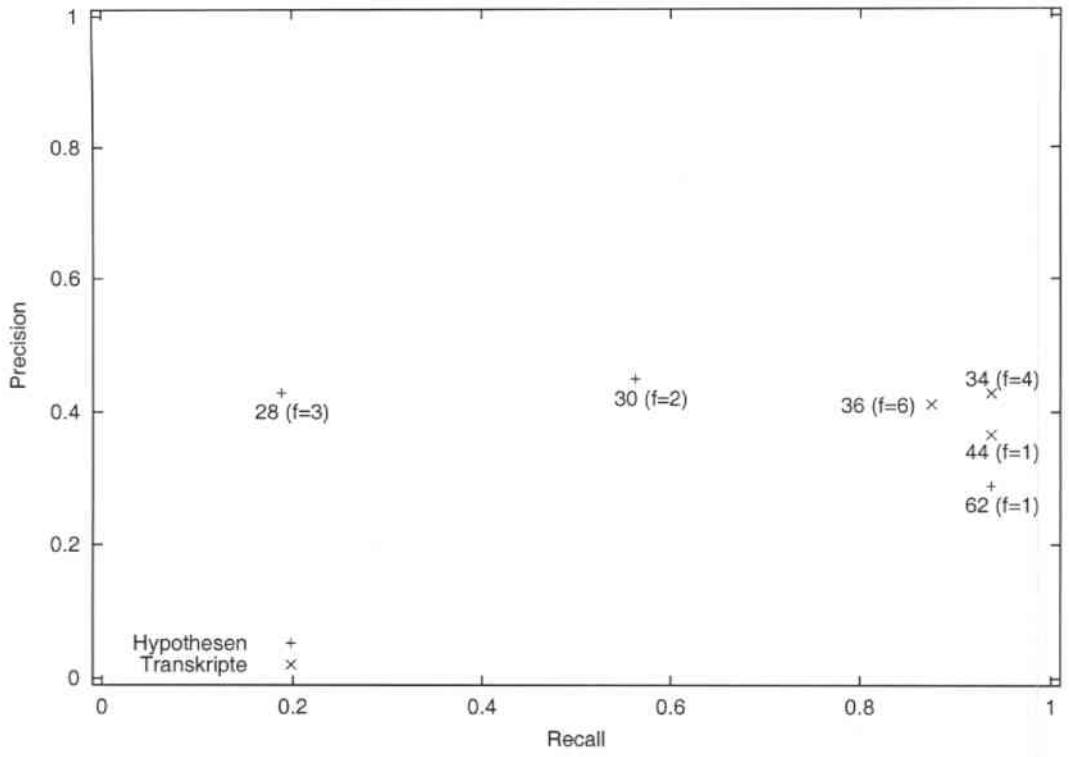



Abbildung 4.4: 2 Merkmale (pp_h2,satzl) mit verschiedenen Fudgefaktoren

nicht ganz korrekte Vorgehensweise, kann aber dazu führen, dass sich insgesamt das Ergebnis verbessert. Dies ist auch gelungen, indem mit Hilfe der Fudgefaktoren das Verhältnis der Wahrscheinlichkeiten verändert wurde. Es wurde eine Verbesserung der Ergebnisse sowohl mit 2 als auch mit 6 Merkmalen erzielt (s.Abb. 4.4 und 4.6). So konnte das beste Ergebnis, welches sonst nur mit 4 Merkmalen erreicht wurde, ebenfalls mit 2 und auch mit 6 Merkmalen erreicht werden.

Leider brachte es aber insgesamt keine Verbesserung mehr. Die Multiplikation mit Fudgefaktoren wurde für das jeweils beste Ergebnis mit der jeweiligen Anzahl an Merkmalen durchgeführt (s.Abb. 4.4-6 Fudgefaktoren). (Mit den Merkmalsanzahlen 3 und 5 wurde dieses Experiment nicht mehr durchgeführt, da von den 3 besten Ergebnisse ausgegangen wurde und diese mit den anderen Anzahlen erreicht worden waren.)

Für die Diagrammerstellung wurden die Werte Recall und Precision an den jeweiligen Achsen abgetragen, da diese Werte für das gewünschte Ziel der Befehlsdetektierung aussagekräftiger sind als die Fehlerrate. Die Fehlerrate ist jedoch an den Punkten in Prozent abzulesen. Bei den Diagrammen mit den Fudgefaktoren ist der Wert des jeweiligen Fudgefaktors f in Klammer angegeben.

Da $p(u|HH)$ und $p(u|HM)$ durch die Normalverteilung mit zwei Merkmalen unter anderem mit den Perplexitäten unter dem entsprechenden Modell HH oder HM als Eingabe berechnet wurde, entstand die Frage, ob nicht der alleinige Vergleich der Perplexitäten den gleichen Erfolg hätte. Die Experimente zeigten, dass diese Methode auch beinahe genauso gut war (s.Abb. 4.13).

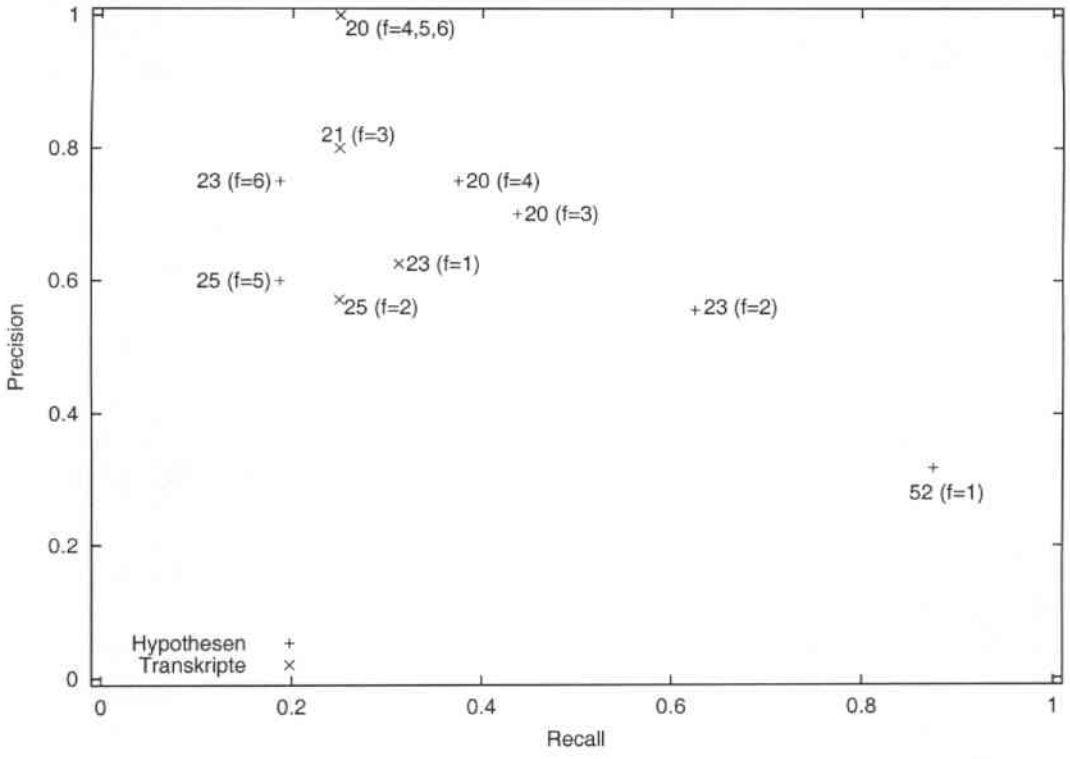



Abbildung 4.5: 4 Merkmale (pp_h1,satzl,pp_h2,pp_m2) mit verschiedenen Fudgefaktoren

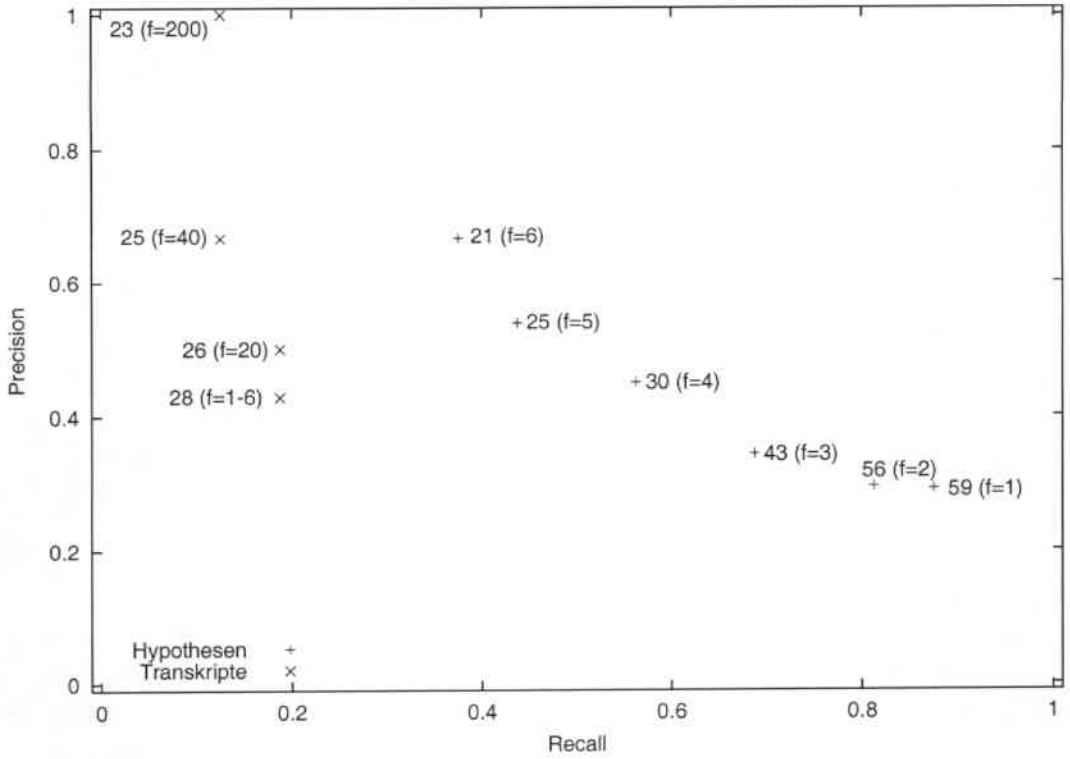



Abbildung 4.6: 6 Merkmale (alle 4 pp,satzl, parsb) mit verschiedenen Fudgefaktoren

## 4.2 Vergleichsmethoden

### 4.2.1 Vergleich der Perplexitäten

Da es sich bei den Daten zur Erstellung des HH1-Sprachmodells hauptsächlich um Terminabsprachen handelte und ein Dialog in einer privaten Umgebung im allgemeinen andere Inhalte hat, ist zu erwarten, dass sich das Ergebnis mit einem spezifischeren Erkenner noch weiter verbessern lässt. Auch für das Modellieren der Befehle ist ein Sprachmodell auf genau solchen Daten, die untersucht werden sollen, am geeignetsten.

Um solche Sprachmodelle zu erhalten, wurden zwei weitere Sprachmodelle auf den Transkripten basierend erstellt (HH2- und HM2-Sprachmodelle s.o.). Da diese Datenbasis wesentlich kleiner ist, wiegen sich Vorteil und Nachteil evtl. wieder auf. Wie sich herausstellte, überwog der Vorteil bei der Betrachtung der Transkripte und der Nachteil bei der Betrachtung der Hypothesen.

Die Tabelle unten zeigt die Werte der Testdaten zum einen mit den Perplexitäten unter den Sprachmodellen HH1 und HM1 (h1) berechnet und zum anderen mit den Perplexitäten unter den Sprachmodellen HH2 und HM2 (h2) berechnet. Diese Berechnungen der Perplexitäten wurden sowohl für die Hypothesen, den Ausgaben des $N$-Gramm-Spracherkenners, als auch – zum Vergleich – für die Transkripte durchgeführt. Die erste Spalte enthält jeweils die Werte mit der Winkelhalbierenden als Trennlinie und die zweite Spalte die Werte mit der „optimalen" Geraden als Trennlinie. Interessant ist, dass sich die Werte meist kaum voneinander unterscheiden. Daran wird deutlich, wie nahe sie an der Winkelhalbierenden liegen, was bedeutet, dass die Perplexitätswerte relativ gleich groß sind, also sich die Entropie der betrachteten Äußerung unter den beiden Sprachmodellen nicht stark unterscheidet. Es fällt auch auf, dass die Fehlerrate im Falle der Hypothesenbetrachtung sich nicht ändert, aber bessere Werte für Recall und Precision erreicht werden konnten. – Die Tabelle soll lediglich den geringen Unterschied der Winkelhalbierenden (direkt) und der optimalen Geraden (linear) als Trennlinie aufzeigen. Mehr Überblick verschafft das Diagramm 4.7.

| | Hypo | | | | Trans | | | |
|---|---|---|---|---|---|---|---|---|
| | h1 | | h2 | | h1 | | h2 | |
| | direkt | linear | direkt | linear | direkt | linear | direkt | linear |
| Fehlerrate | 26 % | 26 % | 39 % | 39 % | 34 % | 33 % | 21 % | 30 % |
| Recall | 63 % | 69 % | 75 % | 88 % | 44 % | 56 % | 69 % | 75 % |
| Precision | 50 % | 50 % | 38 % | 39 % | 37 % | 41 % | 58 % | 46 % |

Fehlerrate 50% heißt: in 50% der Fälle ist unsere Aussage, ob es ein Befehl ist oder ob es kein Befehl ist, falsch. Das wäre nicht besser als Zufall. Allerdings können die Zahlen trügen, da eine bessere Bilanz –nämlich 26% Fehlerrate – gezogen wird, wenn behauptet wird, es war immer eine Dialogphrase, was daran liegt, dass Befehle selten vorkommen. Auch wenn dann die Fehlerrate besser wäre, das Ergebnis somit scheinbar besser, ist dies im eigentlichen Sinne der Befehlsdetektierung nicht der Fall. Deshalb ist es wichtig, in erster Linie die Werte für Recall und Precision zu betrachten.

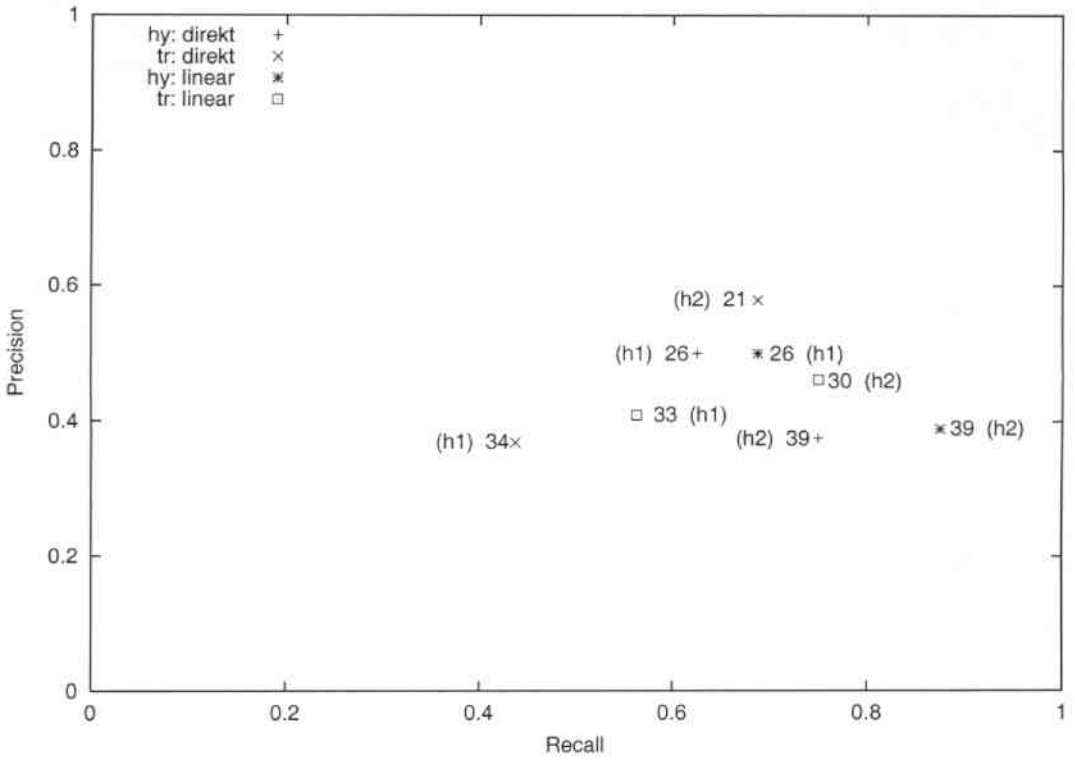



Abbildung 4.7: Ergebnis des Perplexitätenvergleichs

Im Schaubild 4.7 sind die Recall- und die Precision-Werte der Ergebnisse abgetragen. An den Punkten selbst ist die Fehlerrate angegeben. Zunächst wurden, wie bereits erwähnt, die Perplexitäten direkt verglichen, nämlich, welche kleiner ist (dies entspricht in einem Diagramm einer Winkelhalbierenden als Trennlinie). Dann wurde auch versucht die Punkte mit einer optimalen linearen Gerade zu trennen (linear). Hierbei haben jedoch die gängigen Algorithmen zur Fehlerminimierung versagt. Diese haben als optimale Gerade eine Gerade jenseits der Datenpunkte gefunden, die alles den Konversationen zuordnet und wegen der geringen Anzahl an Befehlen eine Fehlerrate von nur 26% aufweist. Deshalb musste ein anderer Weg zum Finden der besten Geraden gesucht werden.

Letztendlich wurde der Schwerpunkt beider Mengen errechnet und eine Mittelgerade in Richtung der Hauptachse der Daten als Entscheidungsgrenze gewählt. Wie im Schaubild zu erkennen ist, konnte jedoch auch dadurch keine wesentliche Verbesserung erzielt werden. Dies war wegen der relativ starken Überschneidung der Daten und der nahen Lage an der Winkelhalbierenden vorauszusehen (s.Abb. 3.1).

Ebenfalls ist in dem Schaubild 4.7 zu erkennen, dass das Ergebnis im Falle der Betrachtung der Transkripte mit den HH2- und HM2-Sprachmodellen besser ist. Dies war zu erwarten, da diese ja auf den selben Transkripten erstellt wurden. Hingegen konnten bessere Ergebnisse seitens der Hypothesen mit den HH1- und HM1-Sprachmodellen erzielt werden.

### 4.2.2 Die Korrelation als Entscheidungskriterium

An den Schaubildern (4.8 - 4.11) ist erkennbar, dass der Wert der Korrelation sehr schwankend ist und die Befehle sich wieder mit den Konversationen überschneiden.

Eine Verbesserung der Spracherkenner würde hier evtl. weiterhelfen, doch dazu müssten noch weitere Daten gesammelt werden und die CFG einerseits erweitert werden – um mehr Befehle zu erkennen – , aber auch andererseits gezielt eingeschränkt werden – um keine Konversationen fälschlicherweise zu erkennen. Dies würde jedoch den Rahmen dieser Arbeit bei weitem sprengen.

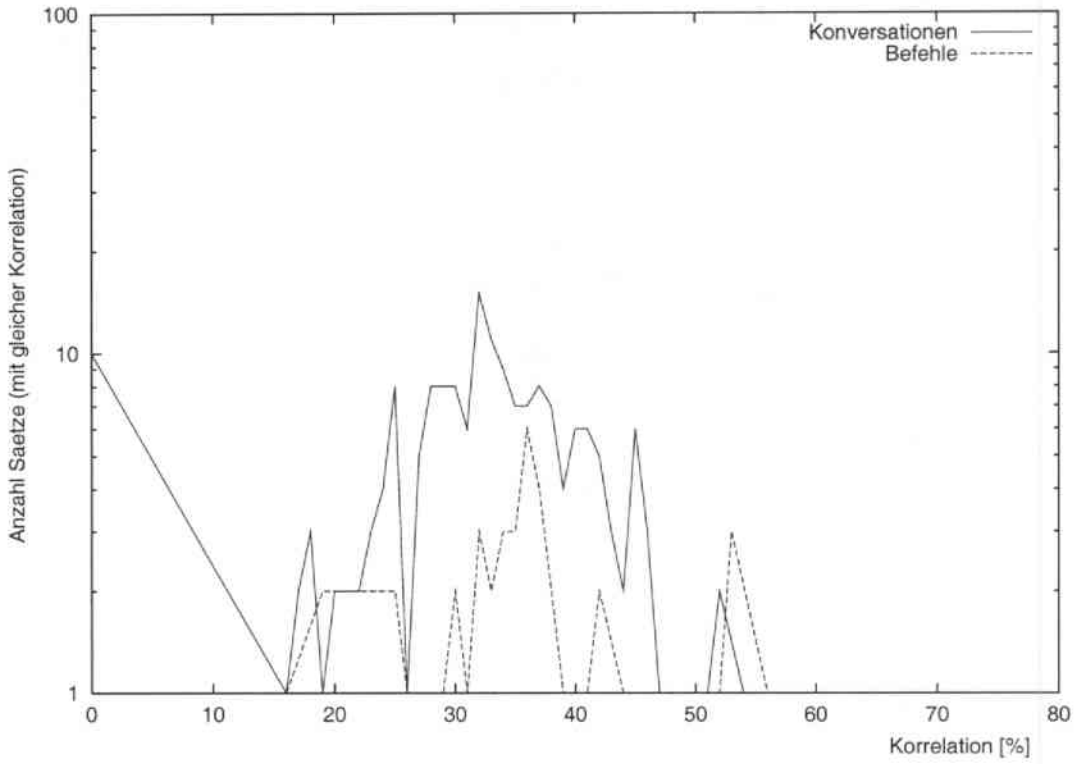



Abbildung 4.8: Histogramm der Buchstaben-Korrelationswerte der Hypothesen

Die Ergebnisse im Diagramm 4.12 zeigen die Werte für den brauchbarsten Schwellwert auf den Testdaten. Dieser konnte durch Austesten verschiedener Schwellwerte gefunden werden. Als Kriterium galt eine kleine Fehlerrate und vor allem hohe Recall- und Precisionwerte. Der beste Schwellwert für die Wortkorrelation war 10% (sowohl auf den Transkripten wie auf den Hypothesen) und für die Buchstabenkorrelation auf den Hypothesen 50% und auf den Transkripten 45%. Werden hingegen die besten Schwellwerte der Trainingsdaten auf den Testdaten verwendet – was der korrekten Vorgehensweise entspräche –, dann stellt sich der Klassifikator auf den Testdaten im Falle der Hypothesen als weniger sinnvoll heraus. Er hat dann zwar nur eine Fehlerrate von 26%, klassifiziert aber nie einen Befehl.

Auf den Transkripten hingegen erhalten wir ein sehr gutes Ergebnis im Falle der Wortkorrelation:

| | Wortkorrelation | Buchstabenkorrelation |
|---|---|---|
| Fehlerrate | 16 % | 25 % |
| Recall | 38 % | 6 % |
| Precision | 100 % | 100 % |

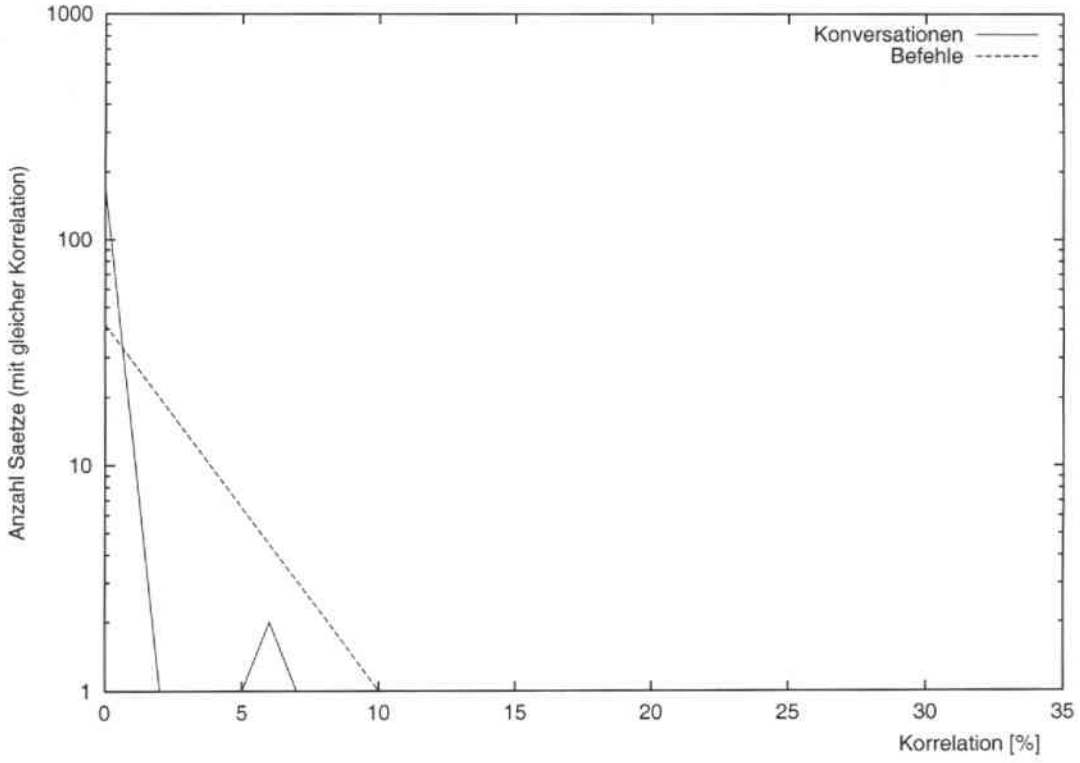



Abbildung 4.9: Histogramm der Wort-Korrelationswerte der Hypothesen

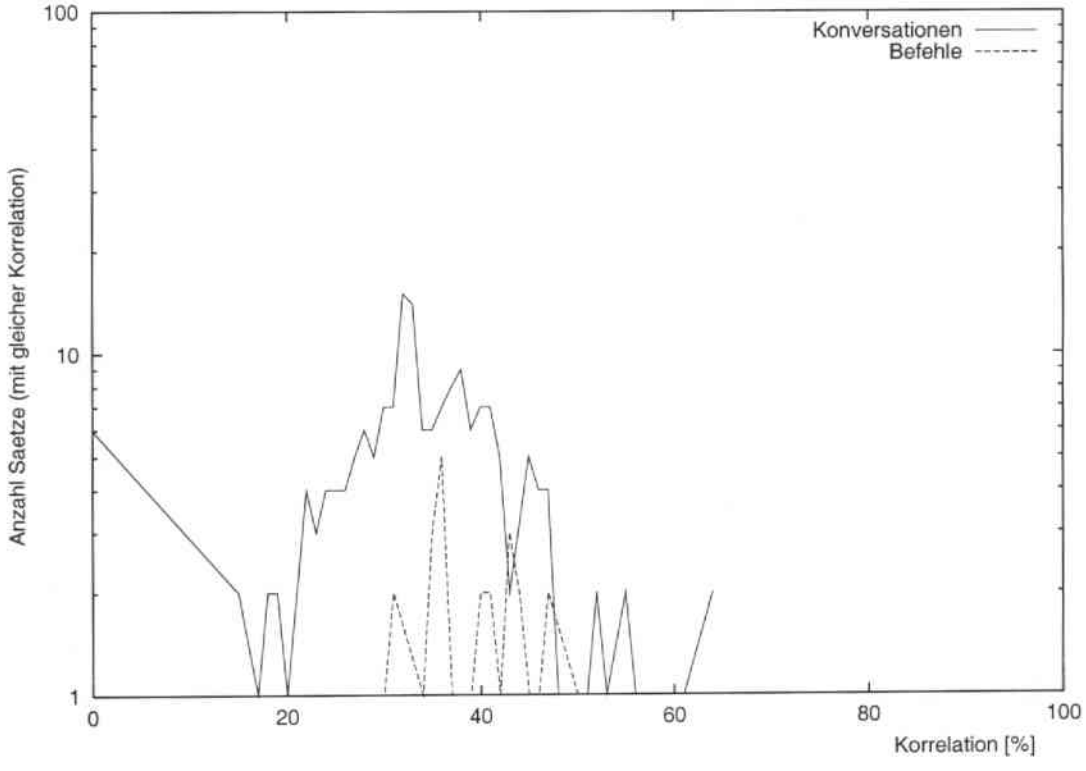



Abbildung 4.10: Histogramm der Buchstaben-Korrelationswerte der Transkripte

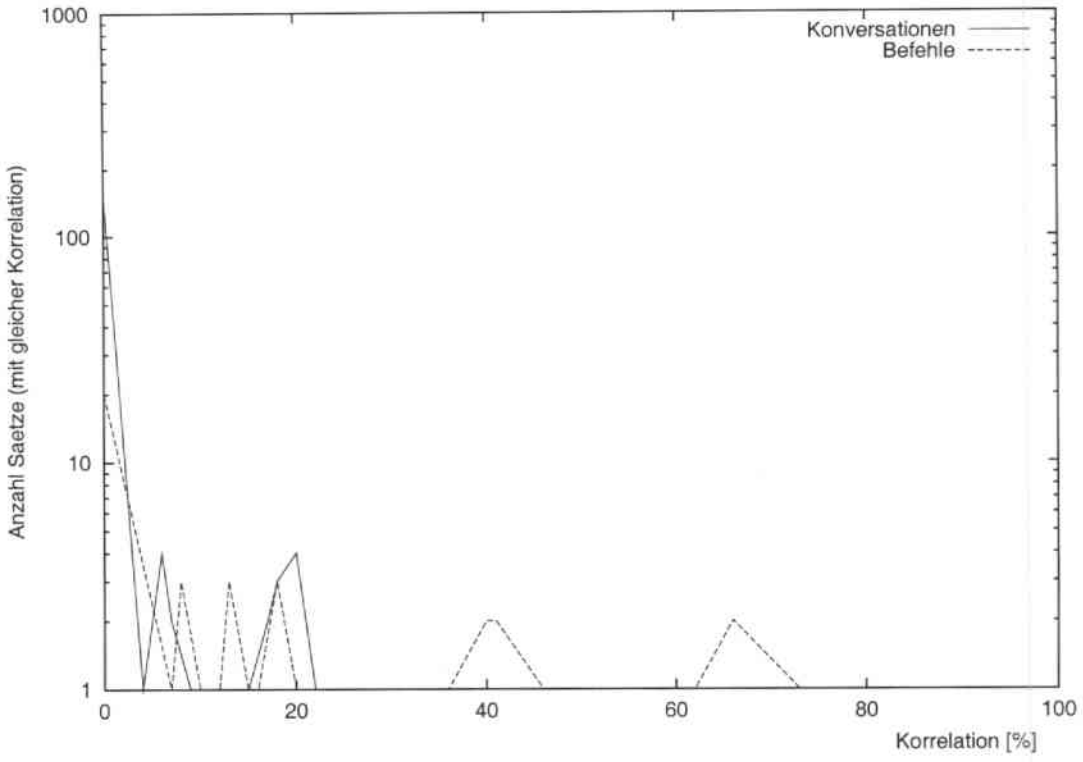



Abbildung 4.11: Histogramm der Wort-Korrelationswerte der Transkripte

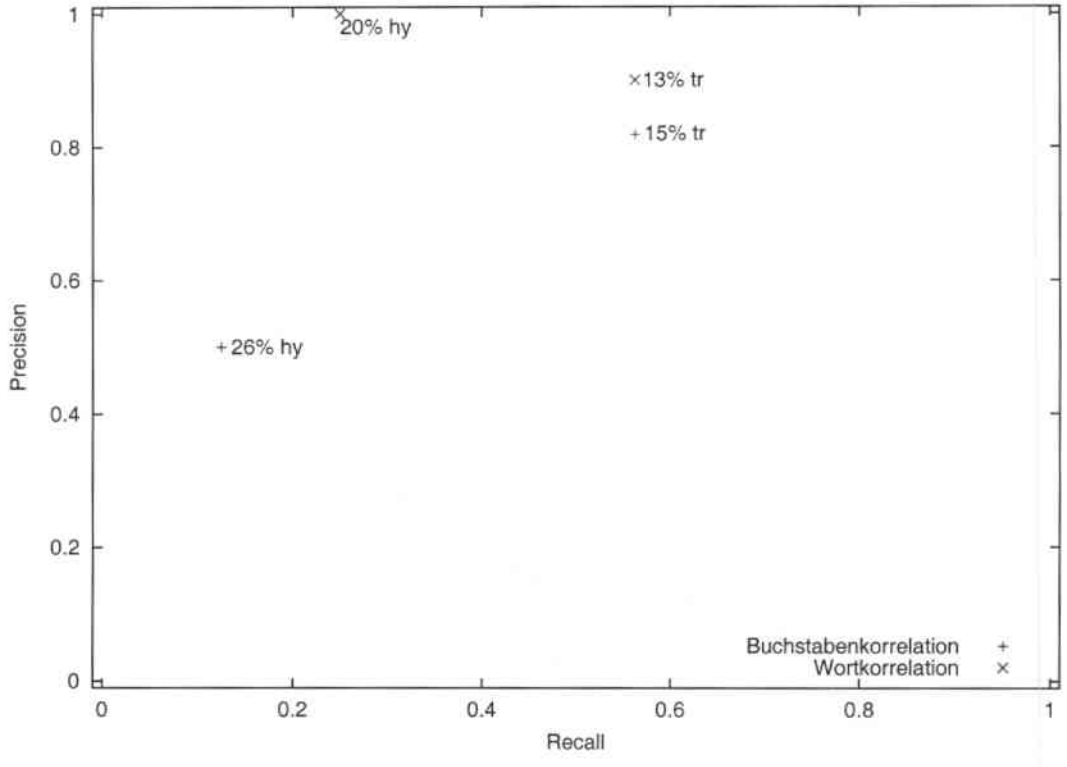



Abbildung 4.12: Diagramm der Korrelationsergebnisse

## 4.3 Der neuronale Ansatz

Das Problem ist nach wie vor, dass die Befehlsteile und Konversationsteile der Gespräche unter den Sprachmodellen Werte annehmen, die sich in ihren Bereichen (Wertebereichen) überschneiden. Wie in dem zweidimensionalen Diagramm in Abb. 3.1 zu sehen ist, sind die beiden Punktmengen der Konversationssätze und der Befehle nicht disjunkt, sondern sie überschneiden sich.

Das heißt, die beiden Punktmengen sind nicht durch eine Funktion voneinander zu trennen, ohne dass die Eigenschaft, generalisieren zu können, verloren geht. Entweder es werden Fehler in der Zuordnung der Klassen gemacht oder die Funktion spiegelt die Trainingsdaten entsprechend wieder. Letzteres ist jedoch nicht hilfreich. Eine Verallgemeinerung auf andere Daten (z.B. Validierungs- und Testdaten) ist dann nicht mehr möglich. Deswegen ist es besser, einen gewissen Fehlerprozentsatz zu akzeptieren, um die Fähigkeit, neue Daten richtig zu klassifizieren, nicht zu verlieren. Das heißt, es soll generalisiert werden können. Dies wurde hier auch immer wieder mit Validierungsdaten getestet. Sobald die Ergebnisse auf den Validierungsdaten schlechter wurden, wurde mit weiterem Training der Funktion auf den Trainingsdaten verzichtet, um eine solche Spezialisierung auf diese Daten zu verhindern.

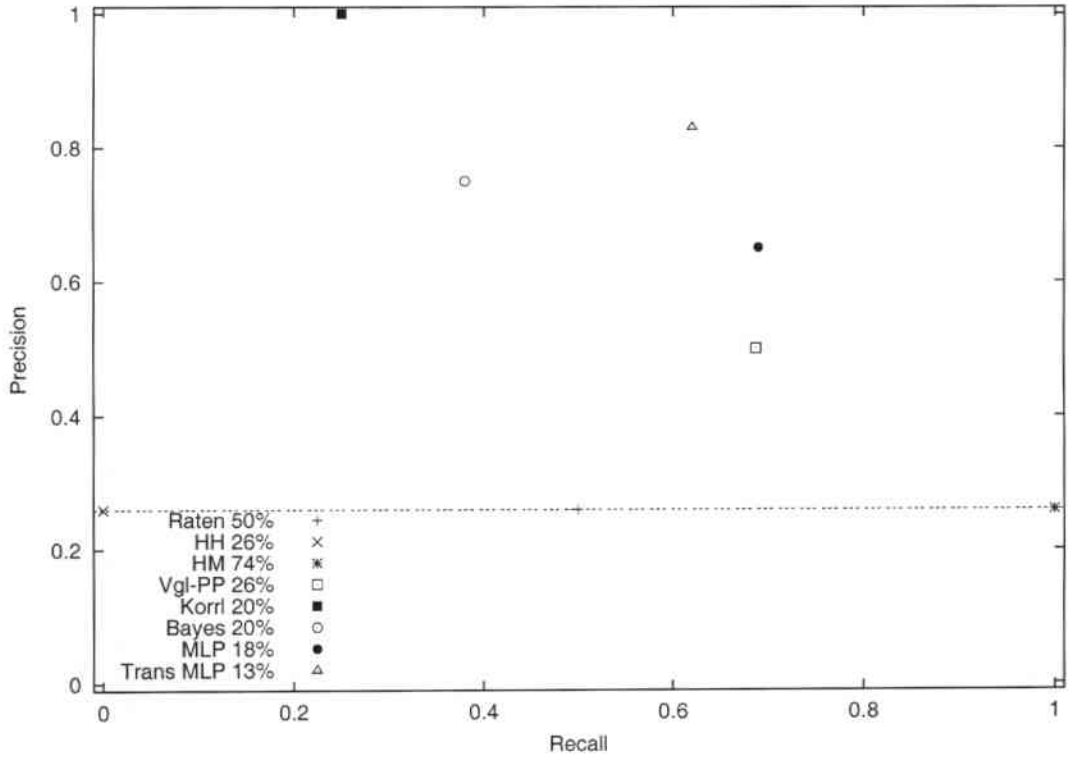



Abbildung 4.13: Vergleich der verschiedenen Methoden

Eine Gegenüberstellung der Ergebnisse der einzelnen Methoden ist in Schaubild 4.13 zu finden. Die Befehlsdetektierung rein zufällig vorzunehmen, also bei jeder Äußerung zu raten, ob es ein Befehl ist oder ob es kein Befehl ist, wird mit Raten bezeichnet. Das Ergebnis mit Klassifikatoren, die immer nur die gleiche Zuordnung durchführen, ist unter HH (immer Mensch-Mensch Dialog) bzw. unter HM (immer Mensch-Maschine Dialog) angegeben. Werden alle Sätze einer der beiden Klassen

zufällig zugeordnet, so wird das Ergebnis in der Mitte (Raten) erzielt. Bei fester Zuordnung aller Kandidaten, wird je nach Zuordnung der Punkt ganz rechts (HM) oder ganz links (HH) erhalten. Werden 50% fest zu Befehlen zugeordnet und der Rest zufällig, wird das Ergebnis in der Mitte von Raten und HM erzielt. Wird der Prozentsatz der fest zugeordneten Äußerungen variiert, werden alle Punkte auf der Verbindungslinie zwischen HH und HM erreicht. Demnach ist auf dem Diagramm 4.13 zu erkennen, dass die angewandten Entscheidungsmethoden alle deutlich besser als der Zufall sind, folglich alle Methoden erfolgreich waren.

Das Ergebnis des Perplexitätenvergleichs (Vgl-PP) ist von der Fehlerrate her gesehen zwar schlechter als der Korrelationsansatz (Korrl), aber vom Ziel, 100% Precision- und Recallwerte zu erhalten, besser. Der Wahrscheinlichkeitstheoretische Ansatz (Bayes(-Klassifikator)) ist so gesehen auch etwas schlechter, aber klar besser als der Perplexitätenvergleich.

Wie zu erwarten, hat allerdings der neuronale Ansatz (MLP, Multi-Layer-Perceptron) die besten Ergebnisse geliefert. So ist er mit einer Fehlerrate für den Transkripten von 13% klar allen anderen Methoden überlegen. Es wurden 62% der Befehle detektiert und eine Genauigkeit von 83% erreicht.

Auch für die Hypothesen konnte ein sehr gutes Ergebnis mit einer Fehlerrate von nur 18% erreicht werden. Die Werte Recall mit 69% und Precision mit 65% sind sehr ausgeglichen.

Dieses Ergebnis konnte für die Hypothesen mit sechs Merkmalen (alle vier verschiedene Perplexitäten, Satzlänge und Parsebarkeit) mit einem voll vernetzten Netz (fully-connected) mit sechs Eingängen, 25 versteckten Neuronen und zwei Ausgängen unter Benutzung des Standard-Backpropagation-Verfahrens bei einer Lernrate von 0.01 erreicht werden. Es wurde hierfür der Stuttgarter Neuronale Netzwerk Simulator [snns] verwendet.

Das beste Ergebnis wurde für die Transkripte bereits mehrmals erhalten und kam – bei ansonsten gleicher Netzstruktur – u.a. schon mit fünf versteckten Neuronen aus. Auch wurde das beste Ergebnis mit verschiedenen Lernraten u.a. schon mit einer Lernrate von 0.1 erreicht.

Gegen Ende der Arbeit wurde als weitere Merkmale das Vorkommen des Wortes Roboter und die Anzahl der vorkommenden Imperative im Satz untersucht. Dies brachte bei diesen Daten leider keine Verbesserung mehr. Hat jedoch dazu geführt, dass das beste Ergebnis mit jeder beliebigen Anzahl an versteckten Neuronen erzielt werden konnte. Es ist also zu erwarten, dass sich eine Steigerung des Ergebnisses bei mehr Daten erreichen lässt.

# 5. Zusammenfassung und Ausblick

Die Experimente haben gezeigt, dass Befehle sich nicht signifikant aus dem Gespräch hervorheben. Dies kann u.a. daran liegen, dass dies in der deutschen Sprache nicht so sehr der Fall ist, wie z.B. bei mehreren Befehlen in Abfolge. So wird vielleicht beim ersten Befehl noch eine Anrede oder – abstrakter – ein Code-Wort (z.B. Home-Multi-Media-Terminal oder Stereoanlage etc.) benutzt und evtl. auch ein klarer Imperativ. Aber bei Befehlen, die sich auf vorangehende Befehle beziehen, wird dies eher weggelassen. Dies zeigt folgendes Szenario:

„Roboter, bring uns zwei Bier!" [...] „Ok, stell sie da hin und mach sie auf!"

Es entfällt hier schon die Anrede, die ein Erkennen einfacher machen würde. Noch schwieriger wird es, wenn auch kein Imperativ mehr vorgefunden wird:

„Home-Multi-Media-Terminal, was waren die letzten Schlagzeilen?" [...] " Ich hätte gerne das Video vom Papst. "

oder wenn auf dem LCD-Bild an der Wand ein anderes Bild gezeigt werden soll:

„Ein Wasserfall wäre schön."

Letzteres könnte genauso gut eine Feststellung sein oder ein Wunsch, aber an den menschlichen Gesprächspartner gerichtet.

Bessere Ergebnisse würden auch mit dem Ansatz zweier Sprachmodelle erzielt werden können, wenn für das auf Mensch-Mensch Dialogen basierende Sprachmodell mehr Daten zur Verfügung stünden, die aus gerade solchen Gesprächen, in denen zwischendurch Befehle an Menschen erteilt werden, bestehen. Dann könnten die reinen Mensch-Mensch Dialoge (ohne Befehle) extrahiert werden und nur auf diesen Mensch-Mensch Dialoganteilen ein Sprachmodell generiert werden. Mit den Befehlen könnte analog verfahren werden, d.h. sie ebenfalls extrahieren und aus den Befehlsphrasen ein Sprachmodell erzeugen.

Dies würde dann wirklich bedeuten, dass die Modellierung entscheidend bessere Voraussetzungen hat, um in der gegebenen Problemstellung eine geeignete Differenzierung vorzunehmen. Ein neuronales Netz hätte dann wahrscheinlich auch die Möglichkeit, eine Klassifizierungsfunktion zu erlernen, ohne so speziell zu werden,

dass es die Trainingsdaten wiederspiegelt. Dann ist anzunehmen, dass bereits sehr gute Ergebnisse erzielt werden können, ohne dass das Netz die Trainingsdaten zu genau erlernt, also ein Generalisieren noch möglich ist. Ein Grund hierfür liegt darin, dass dann einfachere Trennfunktionen gefunden werden können.

Leider konnte dies bisher noch nicht umgesetzt werden, da die Daten dazu nicht zur Verfügung stehen. Auch müssten die Datenmengen sehr groß sein, denn Befehle kommen im Verhältnis zu Konversationssätzen nur sehr selten vor. Es sollte auch bei dem auf Befehlen generierten Modell eine genügend große Datenbasis vorhanden sein.

Bei den vorhandenen Datensätzen könnte für den ausschließlich auf Sprachmodellen basierenden Ansatz evtl. eine Verbesserung der Ergebnisse eintreten, wenn auf dem zweidimensionalen Diagramm die Daten so transformiert werden, dass die Trennlinie in ihrer Hauptausdehnung senkrecht zu einer Achse steht.

Auf jeden Fall ist zu erwarten, dass durch Hinzunahme mehrerer Modalitäten, wie z.B. Blickrichtung oder/und Standort des Sprechers etc. eine signifikante Ergebnissteigerung wahrscheinlich ist. So hilft der Standort des Sprechers bestimmt auch bei der Befehlsdetektierung, da angenommen werden kann, dass im Wohnzimmer der Befehl „Hol mir ein Bier!" wahrscheinlicher ist als unter der Dusche oder in ähnlichen, konkurrierenden Zusammenhängen.

Eine ebenfalls denkbare Erweiterung wäre das Festhalten des aktuellen Dialogstatus oder der Historie als "online Gespräch" mit dem Speichern von Begebenheiten, auf die später evtl. Bezug genommen wird.

Wichtig wäre dies zum Beispiel bei folgendem Gesprächsverlauf: „Hol mir eine Tasse aus dem Schrank." – „Stell sie auf den Tisch! " [10 Minuten später] „Gieß mir Tee ein! "

Abschließend ist zu sagen, dass sich mit einer ausreichend großen Sammlung von Daten aus einem problemgerechten Szenario der „Befehlsdetektierer" noch verbessern ließe; besonders mit der Hinzunahme mehrerer Modalitäten, durch die auch zusätzliche Feature/Funktionen realisiert werden könnten.

# Literatur


[Bish95] Christopher M. Bishop. *Neurol Networks for Pattern Recognition.* Bookcraft (Bath) Ltd, Midsomer Norton, Somerset. 1995.

[fame] *http://www.fame-project.org.*

[Hayk94] Simon Haykin. *Neural Networks, A Comprehensive Foundation.* Macmillan College Publishing Company. 1994.

[HeKP91] John Hertz, Anders Krogh und Richard G. Palmer. *Introduction to the Theory of Neural Computation.* Nr. 8. Addison-Wesley. März 1991.

[sfb] *http://www.sfb588.uni-karlsruhe.de.*

[snns] *http://www-ra.informatik.uni-tuebingen.de/SNNS.*

[verb] *http://verbmobil.dfki.deg.*

[vodi] *http://wwwisl.ira.uka.de/nojs/vodis.html.*

[WaLe90] Alex Waibel und Kai-Fu Lee. *Readings in Speech Recognition.* Morgan Kaufmann Publishers, Inc., San Mateo, California. 1990.