Lehrstuhl für Mensch–Maschine–Kommunikation
Technische Universität München

Diskriminative Methoden zur automatischen Spracherkennung für Telefon–Anwendungen

Josef G. Bauer

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. J. Eberspächer
Prüfer der Dissertation:	1.	apl. Prof. Dr.-Ing., Dr.-Ing. habil. G. Ruske
	2.	Univ.-Prof. Dr.-Ing. G. Färber

Die Dissertation wurde am 23.1.2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.6.2001 angenommen.

Zusammenfassung

Im Mittelpunkt der vorliegenden Arbeit stehen Methoden zur Verbesserung der Erkennungsleistung eines Systems zur automatischen Erkennung von gesprochener Sprache. Im Besonderen werden hierbei Methoden zur Parameterschätzung für *Hidden–Markov–Modelle*, dem Lernen des Systems anhand einer Trainingsstichprobe, betrachtet. Von dem Standardschätzverfahren *Maximum Likelihood* ist bekannt, daß es bei den praktisch immer gegebenen Einschränkungen durch eine begrenzte Trainingsstichprobe und vereinfachte Modellannahmen nicht zu einer optimalen Lösung im Sinne der Erkennungsgenauigkeit führt. Beim Maximum Likelihood Verfahren wird lediglich der Parametersatz gefunden, dessen Modellwahrscheinlichkeit für die Trainingsmenge ein Maximum annimmt. Jedoch ist es durch den Einsatz von sogenannten diskriminativen Methoden andererseits möglich, einen auf optimale Trennbarkeit der Klassen ausgerichteten Parametersatz zu finden. Bei dem Verfahren *Minimum Classification Error (MCE)*, auf das diese Arbeit aufbaut, können die Modellparameter sogar bezüglich einer für die Anwendung entscheidenden Größe wie der Wortfehlerrate optimiert werden.

Mit Hilfe der an die Modellstrukturen des untersuchten Systems angepaßten Lernalgorithmus konnte gezeigt werden, daß sich durch die MCE–basierte Nachschätzung von Verteilungsmittelpunkten signifikante Verbesserungen der Worterkennungsrate erreichen lassen. Eine Analyse der Beziehungen zwischen der Menge freier Systemparameter und der Menge zur Verfügung stehender Trainingsmuster ergab, daß das diskriminative Training für Systeme mit wenig freien Parametern besonders wirksam ist. Weiterhin zeigte sich, daß das MCE–Verfahren das Potential großer Trainingsdatenbanken besser nutzen kann als das Maximum Likelihood Verfahren.

Zur Einstellung eines kritischen Parameters der beim MCE–Verfahren notwendigen Fehlerapproximation wurde ein Schema erstellt, das es erlaubt, die optimale Einstellung des Parameters leicht zu finden. Mit Hilfe einer speziellen Normierungstechnik konnte das zur Optimierung notwendige Gradientenabstiegsverfahren so verbessert werden, daß ein gutes Konvergenzverhalten einfach erreicht werden kann. Durch die Einführung einer sogenannten standardisierten Schrittweite konnte das in der praktischen Anwendung diffizile Problem der Schrittweitensteuerung für das Gradientenabstiegsverfahren erheblich entschärft werden. Durch diese drei vorgestellten algorithmischen Erweiterungen vereinfacht sich die Anwendung des MCE–Trainingsverfahren in der Praxis erheblich.

Anhand mehrerer, für Telefondialogsysteme wichtiger Erkennungs– und Trainingsaufgaben wurden verschiedene Varianten des diskriminativen Trainings verglichen. Dabei wurden neben wortschatzunabhängigem Training folgende Trainings– bzw. Erkennungsaufgaben betrachtet: Einzelworterkennung mit kleinem Wortschatz, kontinuierliche Erkennung von Ziffern und Erkennung von Buchstabenfolgen. Besonderer Wert wurde hierbei auf den Zusammenhang zwischen dem Ziel der Optimierung und der für die Anwendung relevanten Größe, der Wortfehlerrate, gelegt, was in der Literatur bisher kaum berücksichtigt wurde. Dabei erwiesen sich für das untersuchte System globale, auf die eigentliche Anwendung ausgerichtete Optimierungskriterien gegenüber lokalen Kriterien wie der Minimierung der Zustandsfehlerrate in den verwendeten Hidden–Markov–Modellen als überlegen. Für die Optimierung der Fehler auf Ebene von Wörtern und Wortfolgen konnten signifikante Verbesserungen für alle Anwendungsfälle erzielt werden. Für ein auf einen kleinen Wortschatz optimiertes Training konnte eine Reduktion der Wortfehlerrate von bis zu 48% und für wortschatzunabhängiges Training bei größerem Vokabular von bis zu 9% erzielt werden.

Das MCE–Trainingsverfahren wurde um ein Konzept zur Verwendung von a–priori–Wort-Verwechslungsmatrizen erweitert, mit deren Hilfe unter anderem Aussprachevarianten für das diskriminative Training berücksichtigt werden können. Um die für die Durchführung der Parameteroptimierung erforderliche hohe Rechenzeit zu reduzieren, wurden zwei Verfahren zur Beschleunigung des Trainingsverfahrens vorgestellt. Mit diesen Verfahren konnte das Training für unterschiedliche Anwendungen in weniger als der Hälfte der sonst erforderlichen Zeit durchgeführt werden. Im Hinblick auf die Verwendung von sogenannten Füllwörtern, auf die nicht–stationäre Geräusche abgebildet werden, wurden speziell abgestimmte Algorithmen eingeführt. Durch die vorgeschlagene konsistente Integration der Füllwörter in das MCE–Training für Wortfolgen konnte die Wortfehlerrate für Ziffernketten nochmals um 18% reduziert werden.

Vorwort

Die vorliegende Arbeit ist während meiner Mitarbeit in der Sprachverarbeitungsgruppe der Siemens AG in München Neuperlach entstanden. Basis für die wissenschaftliche Arbeit war weiterhin die Kooperation zwischen Siemens und dem Lehrstuhl für Mensch–Maschine–Kommunikation der Technischen Universität München.

An erster Stelle möchte ich meinem Doktorvater Prof. Dr.-Ing. Günther Ruske für die hochschulseitige Betreuung der Arbeit herzlich danken. Der rege Gedankenaustausch mit ihm und seiner Forschungsgruppe hat wesentlich zum Gelingen dieser Arbeit beigetragen. Für die Übernahme des Koreferats und der damit verbundenen Mühen danke ich Prof. Dr.-Ing. Georg Färber.

Weiterhin gebührt mein besonderer Dank Dr. Harald Höge, dem Leiter der Sprachverarbeitungsgruppe bei Siemens. Er hat mir stets den Rücken für die Forschung freigehalten und war mir ein wichtiger Diskussionspartner.

Unterstützung für meine Arbeiten habe ich nicht zuletzt bei den Mitarbeitern der Siemens Spracherkennungsgruppe erhalten. An erster Stelle möchte ich hier Dr. Joachim Köhler danken, ohne dessen Entwicklungsarbeiten an der Siemens–eigenen Software für Hidden–Markov–Modellierung *ear* meine Implementierungen nicht denkbar gewesen währen. Weiterhin danke ich Dr. Udo Bub, Udo Hain, Dr. Jochen Junkawitsch, Dr. Erwin Marschall und Ute Ziegenhain für ihre Unterstützung und Motivation.

An dieser Stelle möchte ich auch allen Entwicklern von freier Software danken. Ohne so hervorragende Programme wie *emacs*, TEX oder *gnuplot* hätte ich meine Ideen nur mit erheblich mehr Mühen praktisch umsetzen können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Spracherkennung als Mittel der Mensch–Maschine–Kommunikation

Seit nunmehr mehreren Jahrzehnten arbeiten Forscher an der Entwicklung einer automatischen Spracherkennung, der *hörenden Maschine*. Seit dem Einsatz von stochastischen Methoden, insbesondere von sogenannten Hidden–Markov–Modellen (HMM) ab ca. 1980, hat diese Disziplin rasante Fortschritte gemacht. Auch die wachsende Verfügbarkeit von Sprachdatenbanken hat der Erforschung dieser Systeme großen Auftrieb beschert.

Nicht zuletzt dem exponentiellen Wachstum der Leistungsfähigkeit von Mikroprozessorsystemen ist es zu verdanken, daß spracherkennende Systeme gegen Ende der 90'er Jahre in viele Bereiche des täglichen Lebens Einzug gehalten haben. Besonders populär ist Spracherkennungssoftware zum Diktieren von Texten an Personalcomputern. Diese Systeme bieten schon heute eine akzeptable Erkennungssicherheit, wobei jedoch meist eine Trainingsphase zur sprecherspezifischen Anpassung des Systems an den Benutzer notwendig ist. Der Nutzen dieser Diktiersysteme liegt hauptsächlich in der schnelleren Texteingabe im Vergleich zur Eingabe über Tastatur — eine nicht–professionelle Beherrschung der Maschinenschrift vorausgesetzt.

Das wohl größere Potential für die automatische Spracherkennung bietet die Nutzung des größten Automaten der Welt — des Telefonnetzes. Die 10 oder 12 Tasten eines normalen Telefons bieten dem Benutzer von interaktiven Diensten ein sehr beschränktes Eingabemedium. Dienste, die auf dieses Eingabemedium bauen, sind meist sehr bedienungsunfreundlich und in ihrer Leistungsfähigkeit stark eingeschränkt. Dabei bietet sich eine fast unendliche Vielzahl von Anwendungsmöglichkeiten für Telefondialogsysteme. Angefangen von Auskunftssystemen, wie einer Fahrplanauskunft, bis hin zu komplexen Interaktionen wie der Abwicklung von Bankgeschäften sind die Möglichkeiten fast unbegrenzt. Ist ein Telefondialogsystem an das Telefonnetz angeschlossen, steht die angebotene Dienstleistung quasi an jedem Ort der Erde zur Verfügung.

Der stark wachsende Markt der Mobiltelefonie kann hier als Katalysator für den Markt für interaktive Dienste über Telefon dienen.

Mehrere Gesichtspunkte machen die Spracherkennung über Telefon jedoch relativ schwierig. Zum einen steht das System im allgemeinen nicht nur einem Benutzer zur Verfügung, was die sprecherunabhängige Funktionsweise der Erkennung notwendig macht.

Des weiteren bereiten die geringe Bandbreite des Telefonkanals und die Vielfalt der durch verschiedene Mikrophone und Übertragungskanäle bestehenden variablen Übertragungscharakteristiken erhebliche Probleme.

Die Mehrzahl der heute angebotenen sprachgesteuerten Dienste sind von einfacher Natur. Häufig wird einfach die Eingabe über die Tasten des Telefons durch eine Eingabe der entsprechenden Wörter ersetzt. Das Ergebnis ist oft nicht sehr benutzerfreundlich und wenig leistungsfähig. Es existieren auch schon einzelne Systeme mit größerem Wortschatz und Erkennung kontinuierlicher Sprache, die einen besseren Benutzerkomfort bieten.

Allen Systemen gemeinsam ist die heute noch sehr beschränkte Leistungsfähigkeit der Spracherkennung. Eine hohe Erkennungsgenauigkeit ist jedoch Grundvoraussetzung für eine Akzeptanz eines Telefondialogsystems durch die Benutzer. Man kann davon ausgehen, daß ein heutiger automatischer Spracherkenner im Schnitt zehnmal soviele Wörter falsch oder nicht erkennt wie ein Mensch. Untersuchungen belegen, daß die reduzierte Erkennungsgenauigkeit der heutigen Systeme zu einem Großteil nicht auf die fehlende Integration von höherem Wissen (Pragmatik) zurückzuführen ist. Einen großen Schwachpunkt stellt die akustisch–phonetische Modellierung dar. Den Kern der akustisch–phonetischen Modellierung stellen in einem auf Hidden–Markov–Modellen–basierten System die anhand der Trainingsstichprobe geschätzten Parameter der Modelle — die Referenzmuster der Sprachlaute — dar.

Zur Verbesserung der akustisch–phonetischen Modellierung besteht zum einen die Möglichkeit, die prinzipielle Struktur der Modelle zum ändern. Ein weiterer Ansatzpunkt ist die Verbesserung der Schätzmethoden für die Parameter der Modelle. Der auch als Training bezeichnete Vorgang des Lernens von Modellparametern steht im Mittelpunkt der vorliegenden Arbeit. Ausgehend von einem praxisnahen Forschungssystem werden sogenannte diskriminative Lernverfahren untersucht. Diesen über die Standardmethoden hinausgehenden Methoden ist eine Analyse der Wechselwirkung von korrekter Klassifikation und möglicher Falschklassifikation gemeinsam. Das Ziel der Arbeit ist eine Verbesserung der Erkennungsleistung von Spracherkennungssystemen mittels diskriminativer Methoden. Besondere Berücksichtigung findet dabei stets die Einsetzbarkeit in realen Telefondialogsystemen.

1.2 Aufbau eines Spracherkennungssystems

Bis auf sehr wenige Ausnahmen, bei denen der Klassifikator direkt auf das Zeitsignal der Sprache aufsetzt ([Liaw und Berger, 1998]), lassen sich Systeme zur automatischen Spracherkennung in die Funktionsblöcke Merkmalsextraktion und Klassifikation aufteilen (siehe Abbildung 1.1).

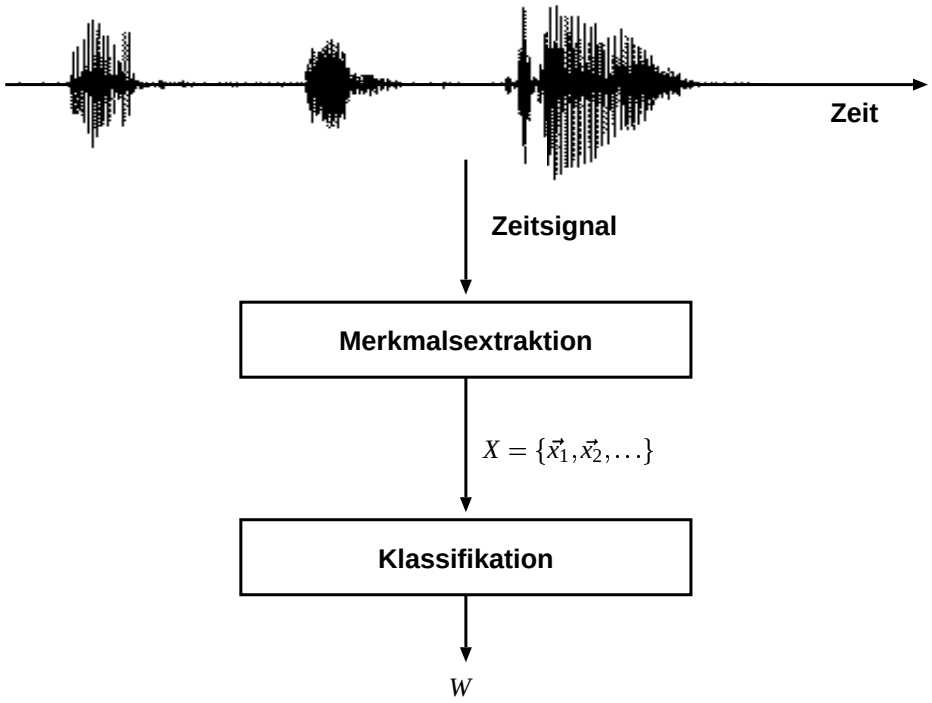

Abbildung 1.1: *Aufbau eines Spracherkennungssystems aus Merkmalsextraktion und Klassifikation*

Merkmalsextraktion Der Eingang der Merkmalsextraktion ist mit dem zeitlich abgetasteten und quantisierten (digitalisierten) Zeitsignal der über ein Mikrophon aufgezeichneten Sprache belegt. Zwischen dem Mikrophon und der Digitalisierung befindet sich eine Übertragungsstrecke, die vielfältig gestaltet sein kann. Es kann sich dabei um das Telefonnetz, einfache Leitungen, Funkstrecken oder eine Mischung aus unterschiedlichen Übertragungsmedien handeln. Natürlich kann das digitale Sprachsignal auch über verschiedene Medien übertragen worden sein, bevor es in die Merkmalsextraktion eingespeist wird.

Das Ergebnis der Merkmalsextraktion ist im allgemeinen eine Abfolge X von hochdimensionalen Merkmalsvektoren, die aus dem Zeitsignal gewonnen wurden. Die Aufgabe der Merk-

malsextraktion besteht in einer Datenreduktion mit einer an den Klassifikator angepaßten Strukturierung der für die Klassifikation relevanten Information. Üblicherweise wird dabei eine Analyse des Signals im Frequenzbereich mit Hilfe einer Zeit–Frequenztransformation, wie der Fourieranalyse, verwendet. Auch im dem untersuchten System kommt eine schnelle Fourieranalyse (*fast fourier transformation, FFT*) zum Einsatz, die alle 10 ms auf einen gefilterten und gefensterten, 25 ms langen Abschnitt des Zeitsignals angewandt wird. Die gewonnenen Energien werden schwellwertbegrenzt und logarithmiert. Das logarithmierte Spektrum wird an äquidistanten 24 Punkten der Mel–Skala ([Zwicker und Fastl, 1990]) mit *sinc*–Kernen ($\frac{\sin(x)}{x}$) gefaltet (cepstrale Glättung) und lautheitsnormiert. Zur Elimination von linearen Übertragungsfunktionen, die z.B. durch unterschiedliche Mikrophone entstehen können, werden die logarithmierten Energien über der Zeit hochpaßgefiltert. Diesen Vorgang nennt man auch Kanalkompensation ([Hauenstein und Marschall, 1995]). Die logarithmierte Energie des Sprachfensters wird ebenfalls über der Zeit hochpaßgefiltert und dem Merkmalsvektor als eine weitere Komponente hinzugefügt. Darüberhinaus bilden die erste und zweite zeitliche Ableitung der Energiekomponente weitere Merkmalskomponenten. Die 24 logarithmierten Kanalenergien werden zu 12 Komponenten gemittelt, und deren erste und zweite zeitliche Ableitungen vervollständigen den Merkmalsvektor.

Die hier nur kurz beschriebene Methode der Merkmalsextraktion ist als *Melfilter* bekannt. Genauere Angaben können [Zünkler, 1991] und [Westendorf, 1995] entnommen werden. Eine sehr vollständige und aus der Sicht dieser Arbeit aktuelle Darstellung findet sich in [Junkawitsch, 2000].

Klassifikation Die Aufgabe der Klassifikation ist es, ausgehend von der Abfolge der Merkmalsvektoren X die gesprochene Folge von Wörtern W zu bestimmen. Neben den in dieser Arbeit verwendeten Hidden–Markov–Modellen werden auch künstliche neuronale Netze ([Robinson, 1994]) oder auch sogenannte hybride Systeme aus Hidden–Markov–Modellen und neuronalen Netzen ([Reichl, 1996]) verwendet. Auf Hidden–Markov–Modelle wird in den Abschnitten 2 und 3 detailliert eingegangen.

1.3 Stand der Technik bei diskriminativen Methoden für die Spracherkennung

Von der Standardmethode Maximum Likelihood zur Schätzung von HMM–Parametern ist bekannt, daß sie bei den in der Praxis gegebenen Einschränkungen wie falschen Modellannahmen und zu kleine Trainingsmenge zu keiner optimalen Lösung im Sinne einer minimalen Fehlerrate führt. Deshalb werden für die automatische Spracherkennung schon seit Ende der 80’er Jahre

diskriminative Methoden eingesetzt, die direkt auf eine maximale Klassentrennbarkeit und somit auf einer maximale Erkennungsrate abzielen. Die folgenden Abschnitte stellen die wichtigsten wissenschaftlichen Beiträge zum Thema diskriminative Methoden für die Spracherkennung kurz vor.

1.3.1 Optimierungskriterien

MMI Bei der ersten Veröffentlichung zum Thema *Maximum–Mutal–Information (MMI)* für die Spracherkennung dürfte es sich um [Bahl u. a., 1986] handeln. Dort wurde die MMI–basierte Optimierung eines sprecherabhängigen diskreten Hidden–Markov–Modells erfolgreich durchgeführt. In engem Zusammenhang dazu stehen die in [Brown, 1987] beschriebenen Arbeiten. Dort wurde das MMI–Kriterium auch für sprecherunabhängige Erkennung benutzt. Sehr erfolgreich wurde MMI von den Autoren von [Cardin u. a., 1991b], [Cardin u. a., 1991a] und [Normandin u. a., 1994] sowohl für die Ziffernkettenerkennung als auch für kontinuierliche Erkennung bei größerem Vokabular eingesetzt. Das MMI–Kriterium wird bis heute in sehr vielen Arbeiten verwendet: z.B. in [Neukirchen und Rigoll, 1997], [Valtchev u. a., 1997], [Warakagoda und Johnsen, 1999] oder [Willett u. a., 1999].

Von den Autoren von [Bahl u. a., 1986] wurde der Begriff *corrective training* geprägt. Das Kriterium ist in [Bahl u. a., 1988] beschrieben. Die Kombination von MMI und corrective training, corrective MMI, wird in [Cardin u. a., 1991a] beschrieben. Aber auch in neueren Arbeiten wie [Schlüter und Macherey, 1998] wird corrective MMI verwendet. Der Vorteil von corrective training und corrective MMI liegt hauptsächlich in einem reduzierten Rechenzeitbedarf, obwohl auch von einer Verbesserung der Erkennungsleistung berichtet wird ([Bahl u. a., 1993]).

MCE Zu den ersten Veröffentlichungen zum Kriterium *Minimum Classification Error (MCE)* zählen [Ljolje u. a., 1990], [Chang und Juang, 1992], [Chou u. a., 1992] und [Juang und Katagiri, 1992]. Ein Grund für die vielen erfolgreichen Anwendungen des MCE–Kriteriums dürfte in den zur Verfügung stehenden Parametern zur Steuerung des Einflusses der Trainingsmuster liegen. Das Hauptanwendungsgebiet von MCE liegt bei den kleinen Wortschätzen wie Ziffern ([Chou u. a., 1992], [Chou u. a., 1993], [Rahim u. a., 1997], [Chou u. a., 1994]) und Buchstaben ([Chang und Juang, 1992], [Juang und Katagiri, 1992], [Euler, 1995], [Juang u. a., 1997], [Gelin-Huet u. a., 1999]). Aber auch zu mittleren ([Gandhi und Jacob, 1998]) sowie größeren Wortschätzen ([Schlüter, 2000]) finden sich Veröffentlichungen.

Vergleich zwischen MMI und MCE Ein direkter und konsistenter Vergleich von MMI und MCE findet sich in [Reichl und Ruske, 1995] und [Schlüter und Macherey, 1998]. In beiden Arbeiten wird auf die theoretische Ähnlichkeiten der Kriterien hingewiesen und über ähnliche Ver-

besserungen bei experimentellen Untersuchungen berichtet. Das MCE–Kriterium lieferte jedoch stets etwas bessere experimentelle Resultate.

Andere Kriterien Neben den meist verwendeten Kriterien MMI und MCE finden sich in der Literatur noch weitere Kriterien. Einige Kriterien stehen in enger Verwandtschaft zu MMI und MCE. In [Beaufays u. a., 1999], [Wu und Guo, 1999] und [Povey und Woodland, 1999] werden an MMI angelehnte Kriterien verwendet. Die in [Gelin-Huet u. a., 1999] und [Nogueiras-Rodriguez und Marinõ, 1998] verwendeten Kriterien sind stark an das MCE–Kriterium angelehnt. In [Bahl und Padmanabhan, 1998] und [Gao u. a., 1999] wird ein spezielles diskriminatives Maß verwendet, um die Anzahl der Mixturkomponenten bei HMMs zu steuern.

1.3.2 Methoden zur Durchführung der Optimierung

Gradientenverfahren In den ersten Veröffentlichungen zum MMI–Kriterium ([Bahl u. a., 1986] und [Brown, 1987]) wurden ausschließlich Gradientenverfahren zur Optimierung verwendet. Die meisten Veröffentlichungen zum MMI–Kriterium verwenden jedoch den erweiterten Baum–Welch–Algorithmus, auf den im nächsten Abschnitt eingegangen wird. Im Gegensatz dazu ist für das MCE–Kriterium die Verwendung von Gradientenverfahren die Standardmethode. Insbesondere wird das Verfahren des *General Probabilistic Descend (GPD)* ([Chang und Juang, 1992], [Chou u. a., 1992], [Euler und Zinke, 1992], [Juang und Katagiri, 1992]) verwendet, für das es einen theoretischen Konvergenz–Beweis gibt ([Juang und Katagiri, 1992]). Oft wird sogar das Kürzel GPD als Synonym für diskriminatives Training, basierend auf MCE mit GPD als Optimierungsverfahren, eingesetzt ([Chou u. a., 1992] und [Euler und Zinke, 1992]). Für das dabei als kritisch einzuschätzende Problem der Schrittweitensteuerung ([Euler und Zinke, 1992]) gibt es in der Literatur so gut wie keine Lösungsvorschläge.

Erweiterter Baum–Welch In [Gopalakrishnan u. a., 1991] wurden die Grundlagen zur Übertragung des Baum–Welch–Algorithmus auf das Problem der MMI–basierten Parameterschätzung gelegt. Dieser *erweiterte Baum–Welch* bietet insbesondere bezüglich der Steuerung der Konvergenz Vorteile gegenüber Gradientenverfahren und hat sich als Standardmethode für die MMI–basierte Optimierung schnell durchgesetzt (z.B. [Cardin u. a., 1991b], [Kapadia u. a., 1993] und [Valtchev u. a., 1997]). Der erweiterte Baum–Welch–Algorithmus wird nur selten für die MCE–basierte Optimierung verwendet. Ein Beispiel findet sich in [Schlüter u. a., 1997], wo auch auf die Ähnlichkeiten zwischen GPD und erweiterten Baum–Welch bei bestimmten Randbedingungen hingewiesen wird.

Andere Verfahren In der Literatur finden sich vereinzelt andere Optimierungsverfahren, hauptsächlich in Zusammenhang mit dem MCE–Kriterium: z.B. [Shimodaira u. a., 1998]. Als Besonderheit ist hier die Optimierung mittels evolutionärer Algorithmen zu nennen, die in [Rudolph, 1999] beschrieben wird.

1.3.3 Trainingsaufgaben

Kleine und mittlere Wortschätze Insbesondere zum MCE–Kriterium wird bei der Mehrheit der Veröffentlichungen eine Optimierung für kleine Wortschätze wie Ziffern oder Buchstaben beschrieben: z.B. [Chou u. a., 1992] und [Chang und Juang, 1992] (vgl. auch Abschnitt 1.3.1). Die durch das diskriminative Training gegenüber dem Maximum–Likelihood–Training erzielten Verbesserungen sind hierbei beträchtlich. Sie liegen im Bereich von 10% bis 50% Reduktion der Wortfehlerrate. Aber auch das MMI–Kriterium wurde sehr erfolgreich für die Optimierung eines Ziffernkettenerkenners eingesetzt: z.B. [Cardin u. a., 1991b] und [Cardin u. a., 1991a].

Für die kleinen Wortschätze werden meist wort– (z.B. [Chou u. a., 1992] und [Euler, 1995]) oder wortfolgenbasierte ([Chou u. a., 1993], [Chou u. a., 1994]) Kriterien eingesetzt, da dadurch gezielt für die eigentliche Anwendung optimiert werden kann.

Beispiele für diskriminatives Training bei mittleren Wortschätzen sind [Bahl u. a., 1986] und [Normandin u. a., 1994]. Auch hier wurden wort– oder wortfolgenbasierte Kriterien verwendet. Verbesserungen bei solchen Anwendungen liegen nur im Bereich von ca. 10% Reduktion der Wortfehlerrate.

Phonemerkennung Eine Vielzahl von Veröffentlichungen beschäftigt sich mit der Optimierung der Phonemerkennungsrate mit Hilfe eines auf Ebene von isolierten Phonemen angesiedelten Kriteriums. Beispiele hierfür sind: [Merialdo, 1991], [Kapadia u. a., 1993], [Reichl und Ruske, 1995] und [Nogueiras-Rodriguez und Marinõ, 1998]. Ein Kriterium auf Ebene von Phonemfolgen wird in [McDermott und Karagiri, 1997] verwendet. Höhere Phonemerkennungsraten werden dort aber auch mit einem Kriterium auf Ebene isolierter Phoneme erzielt. Die bezüglich der Phonemerkennungsrate erzielten Verbesserungen liegen bei den meisten Veröffentlichungen im Bereich von 5% bis 20% Reduktion der Phonemfehlerrate. Leider wird in den wenigsten Veröffentlichungen eine Auswirkung des Phonem–basierten diskriminativen Trainings auf die Worterkennungsleistung beschrieben. Eine Ausnahme hierzu stellt [Nogueiras-Rodriguez und Marinõ, 1998] dar, wo eine Reduktion der Wortfehlerrate von bis zu 40% berichtet wird. Allerdings wird dort ein spezielles Verfahren zur Optimierung bezüglich eines vorgegebenen Wortschatzes verwendet.

Große Wortschätze In der Literatur finden sich auch einige Anwendungen von diskriminativem Training für große Wortschätze: z.B. [Valtchev u. a., 1997], [Neukirchen und Rigoll, 1997], [Willett u. a., 1997], [Ruske u. a., 1998], [Gao u. a., 1999], [Beaufays u. a., 1999], [Povey und Woodland, 1999], [Schlüter, 2000]. Bei einigen Veröffentlichungen wie [Valtchev u. a., 1997] werden Kriterien auf Wortfolgenebene verwendet, was wegen der notwendigen kontinuierlichen N–best Suche einen sehr hohen Rechenaufwand zur Folge hat. Die erzielten Verbesserungen bzgl. der Worterkennungsrate liegen bei etwa 5% bis 10%. Es werden jedoch auch Optimierungskriterien auf Zustandsebene eingesetzt, die bezüglich der notwendigen Rechenleistung klare Vorteile bieten. In [Povey und Woodland, 1999] wird z.B. ein MMI–basiertes Kriterium auf Zustandsebene verwendet. Dort wird von einer Verbesserung der Worterkennungsrate von ca. 10% berichtet.

1.3.4 Spezielle Anwendung von diskriminativem Training

HMM–Parameter Neben den Standardparametern bei Hidden–Markov–Modellen wie Mittelpunktsvektoren und Varianzen werden in einigen Veröffentlichungen zusätzliche Modellierungsparameter eingeführt und diese dann diskriminativ geschätzt. Dies sind z.B. Zustandsgewichte ([Wolfertstetter, 1997]) oder Gewichte für Codebücher bei diskreten Hidden–Markov–Modellen ([Hernando u. a., 1995]).

Mehrere Autoren verwenden diskriminative Kriterien, um die Anzahl der Mixturkomponenten pro Zustand zu steuern. Beispiele hierfür sind [Normandin, 1995], [Bahl und Padmanabhan, 1998], [Gao u. a., 1999] und [Schlüter u. a., 1999].

Merkmalsextraktion und –transformation Eine Möglichkeit zur Merkmalstransformation mittels diskriminativer Methoden stellt die *Lineare Diskriminanz–Analyse* (LDA) ([Haeb-Umbach u. a., 1993], [Eisele u. a., 1996]) dar, deren Zielfunktion jedoch völlig anders gestaltet ist als bei den bisher erwähnten Kriterien wie MMI.

In einigen Veröffentlichungen werden Parameter einer linearen Merkmalstransformation mittels diskriminativer Kriterien, die sonst zur Optimierung der HMM–Parameter verwendet werden, nachgeschätzt. Solche lineare Merkmalstransformationen werden z.B. in [Ayer u. a., 1993] und [Ruske u. a., 1998], [Euler, 1995] und [Schlüter, 2000] verwendet. Dort kommt sowohl das MMI–Kriterium als auch das MCE–Kriterium zum Einsatz. Der Schwerpunkt bei den in [Euler, 1995] beschriebenen Arbeiten liegt in der gemeinsamen Optimierung der Merkmalstransformation und der HMM–Parameter. Die diskriminative Optimierung von nicht–linearen Merkmalstransformationen, meist basierend auf neuronalen Netzen, wird z.B. in [Warakagoda und Johnsen, 1999], [Reichl, 1996] und [Rahim u. a., 1997] beschrieben.

Weiterhin werden diskriminative Methoden auch benutzt, um gezielt Teile des Merkmals-

vektors zu vernachlässigen, ohne daß sich dadurch die Erkennungsleistung deutlich verringert. Ein Beispiel hierfür ist [de la Torre u. a., 1997]. In [Biem und Katagiri, 1997] wird die Bestimmung von Filterbankparametern für die Merkmalsextraktion basierend auf dem MCE–Kriterium beschrieben.

Andere Anwendungen Neben der Parameterschätzung für Hidden–Markov–Modelle und Merkmalsextraktion werden diskriminative Methoden auch vereinzelt für andere Teile eines Spracherkennungssytems verwendet. Die Autoren von [Chengalvarayan und Deng, 1998] benutzen das MCE–Kriterium zur Parameterschätzung von sogenannten Trajektorienmodellen, die eine Erweiterung zu HMMs darstellen. Ähnlich hierzu sind auch die Arbeiten zu *Markov–Graphen* in [Wolfertstetter, 1997]. In [Warnke u. a., 1999] werden die Kriterien MMI und MCE für die Schätzung von Sprachmodellparametern verwendet. In [Chou, 1997] wird das MCE–Kriterium dazu verwendet, baumartige Wahrscheinlichkeitsdichtefunktionen für die beschleunigte Berechnung der Emissionswahrscheinlichkeiten zu konstruieren. In einigen Veröffentlichungen wird der Einsatz des MCE–Kriteriums zur Optimierung von speziellen Modellen für die Rückweisung von Wörtern außerhalb des Wortschatzes berichtet: z.B. [Rahim u. a., 1995], und [Sukkar u. a., 1997]. Als letztes Beispiel für eine spezielle Anwendung von diskriminativen Methoden soll hier [Neukirchen und Rigoll, 1997] genannt werden, wo die Parameter einer Vektorquantisierung für diskrete HMMs mit Hilfe des MMI–Kriteriums bestimmt werden.

Kapitel 2

Stochastische Modellierung mit Hidden–Markov–Modellen

2.1 Statistischer Klassifikator

Die Aufgabe eines Klassifikators in der automatischen Spracherkennung ist es — ausgehend von einer Abfolge von akustischen Merkmalen X — die gesprochene Folge von Wörtern W zu bestimmen ([Höge, 1993]). Die Bayes'sche Entscheidungsregel besagt, daß der optimale Klassifikator das Klassifikationsergebnis durch Bestimmung derjenigen Klasse mit maximaler a–posteriori Wahrscheinlichkeit auswählt. Im Sinne der obigen Aufgabenstellung lautet die Klassifikationsregel:

$$W_{\text{Bayes}} = \underset{W}{\operatorname{argmax}} P(W|X) \tag{2.1}$$

Die Menge der möglichen Klassen ist hier also die Menge der möglichen Wortfolgen $\{W\}$, über die zu maximieren ist. Um eine im Bayes'schen Sinne optimale Entscheidung zu treffen, benötigt man eine vollständige Beschreibung des Zusammenhangs $P(W|X)$, welche in der Praxis nicht verfügbar ist.

Durch einige einfache Umformungen von (2.1) erhält man:

$$W_{\text{Bayes}} = \underset{W}{\operatorname{argmax}} \frac{P(X|W)P(W)}{P(X)} \tag{2.2}$$

Da in (2.2) das Ergebnis unabhängig von $P(X)$ ist, kann die Entscheidungregel auch lauten:

$$W_{\text{Bayes}} = \underset{W}{\operatorname{argmax}} P(X|W)P(W) \tag{2.3}$$

Der Vorteil von (2.3) gegenüber (2.1) zeigt sich im folgenden Abschnitt im Zusammenhang mit Hidden–Markov–Modellen.

Um eine Klassifikation nach (2.3) vornehmen zu können, müssen $P(X|W)$ und $P(W)$ bekannt sein. In der Praxis benötigt man mathematische Modelle für die beiden Zusammenhänge. Im allgemeinen werden mathematische Modelle die gesuchten Wahrscheinlichkeiten nicht vollständig und korrekt beschreiben. Das grundsätzliche Ziel einer statistischen Modellierung muß jedoch immer sein, möglichst umfassende und korrekte Modelle zu finden, für die gelten muß:

$$\hat{P}(X|W) \approx P(X|W) \tag{2.4}$$

$$\hat{P}(W) \approx P(W) \tag{2.5}$$

Das mathematische Modell $\hat{P}(X|W)$ wird als *akustisches Modell* und $\hat{P}(W)$ als *Sprachmodell* bezeichnet. Während auf Sprachmodelle in dieser Arbeit im Abschnitt 3 kurz eingegangen wird, sollen in den folgenden Abschnitten Hidden–Markov–Modelle detaillierter beschrieben werden.

2.2 Definition eines HMM

Hidden–Markov–Modelle ([Rabiner, 1989], [Picone, 1990], [Rabiner und Juang, 1993], [Schukat-Talamazzini, 1995]) können stochastische Erzeugungsprozesse für Folgen von Mustern beschreiben. In der automatischen Spracherkennung dienen sie zur approximativen Beschreibung des Zusammenhangs $P(X|W)$. Legt man ein HMM λ_W für eine Wortfolge W zugrunde, so läßt sich die Wahrscheinlichkeit für die zu einer gesprochenen Wortfolge korrespondierenden Merkmalsfolge X näherungsweise durch

$$\hat{P}(X|W) = P(X|\lambda_W) \tag{2.6}$$

bestimmen.

Hidden–Markov–Modelle sind stochastische Automaten, bestehend aus einer Menge von Zuständen (engl. *states*, Zustandsindex s) und möglichen Übergängen, die jeweils mit Wahrscheinlichkeiten belegt sind. Während die *Übergangswahrscheinlichkeiten* $a_{s,s'}$ die Wahrscheinlichkeiten von Übergängen zwischen zwei Zuständen s und s' beschreiben, geben die *Emissionswahrscheinlichkeiten* $b_s(\vec{x})$ die Wahrscheinlichkeit einer Emission/Aussendung eines Merkmalsvektors $\vec{x}$ in einem Zustand s an. Die Zustände eines Hidden–Markov–Modells werden hierbei als verborgen (engl. *hidden*) bezeichnet, da bei der Betrachtung einer Merkmalsfolge X diese von der unterliegenden Abfolge von Zustandsübergängen im Modell entkoppelt ist.

Vollständig beschrieben wird ein HMM durch einen Satz von Parametern

$$\Lambda = \{\pi_s, a_{s,s'}, b_s(\vec{x})\}. \tag{2.7}$$

Die *Einsprungswahrscheinlichkeiten* π_s geben hierbei an, mit welcher Wahrscheinlichkeit zu Beginn des Prozesses ein Einsprung in den Zustand s erfolgt. $b_s(\vec{x})$ dient zunächst als Platzhalter

für die Parameter der noch festzulegenden Wahrscheinlichkeitsdichtefunktionen, die die Emissionswahrscheinlichkeiten der Zustände beschreiben.

2.3 Modell–Topologien

Unter der Topologie eines Hidden–Markov–Modells versteht man die grundsätzliche Ausprägung der Struktur im Sinne der Anordnung der Zustände und deren Verknüpfung durch die möglichen Übergänge. Zum einen verwendet man im allgemeinen festgelegte Strukturen wie *links–rechts–Modelle*. Zum anderen werden Modelle zur Beschreibung von Wörtern sehr oft aus einer Menge von kleineren Modellen hierarchisch aufgebaut. Die folgenden Abschnitte beschreiben die verwendeten Basis–Topologien sowie die Möglichkeiten eines hierarchischen Aufbaus. Auf die implizit erfolgende Verknüpfung von Modellen zur Beschreibung von Wortfolgen oder Folgen anderer kleinerer Einheiten wird erst im Abschnitt 3 eingegangen.

2.3.1 Basis–Topologie

Im Rahmen dieser Arbeit werden ausschließlich Hidden–Markov–Modelle mit einer sogenannten Bakis–Topologie verwendet. Bei dieser Art der Modellstruktur wird der Bezug zwischen dem zeitlichen Ablauf des Spracherzeugungsprozeß und den Zuständen besonders deutlich. Mit einem zeitlichen Ablauf ist ein Durchlaufen der Zustände von links nach rechts verbunden. Hierbei kann maximal ein Zustand übersprungen werden. Weiterhin sind Selbstübergänge und Übergänge in den jeweils nächsten Zustand erlaubt. Sämtliche Übergangswahrscheinlichkeiten werden

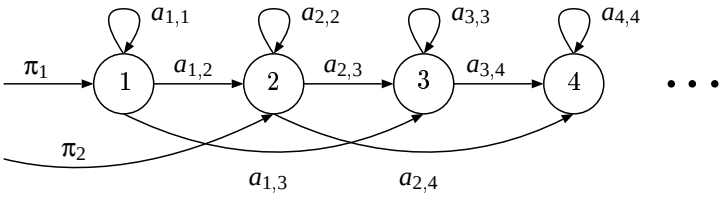

Abbildung 2.1: *HMM mit Bakis–Topologie*

im verwendeten System auf einen Freiheitsgrad reduziert:

$$a_{s,s'} = \begin{cases} a_0 & : \; s = s' \\ a_0 & : \; s' = s+2 \\ 1-2a_0 & : \; s' = s+1 \end{cases} \tag{2.8}$$

Der Selbstübergang und das Überspringen eines Zustands tragen also die Wahrscheinlichkeit a_0. Der Übergang in den jeweils nächsten Zustand trägt die Wahrscheinlichkeit $1 - 2a_0$. In der Praxis wird der verbleibende freie Parameter a_0 nicht anhand der Trainingsstichprobe geschätzt, sondern auf einen konstanten (bewährten) Wert gesetzt.

2.3.2 Lautdauermodellierung

Durch die Übergangswahrscheinlichkeiten ist eine implizite Modellierung der Verweildauer in einem Zustand des Hidden–Markov–Modells und somit in Lauten bzw. Lautuntereinheiten gegeben. Die implizite Verweildauermodellierung läßt sich dadurch verfeinern, daß mehrere Zustände mit identisch modellierten Emissionswahrscheinlichkeiten in Folge geschaltet werden ([Zünkler, 1991]). Während bei einem einzelnen Zustand die Wahrscheinlichkeit für die Verweildauer streng monoton sinkt, kann durch eine Folge von Zuständen eine bestimmte wahrscheinlichste Verweildauer für diese Folge von Zuständen eingestellt werden. In den in dieser Arbeit beschriebenen Systemen wird diese Technik eingesetzt, wobei eine Folge von zwei Zuständen mit identischer Modellierung der Emissionswahrscheinlichkeit verwendet wird.

2.3.3 Phonetischer Ansatz

Um die Anzahl der Parameter eines akustischen Modells niedrig und die Parameter somit schätzbar zu halten, hat sich in der Sprachverarbeitung die Verwendung von sprachlichen Untereinheiten bewährt. Im Rahmen dieser Arbeit wird für einige Anwendungen ein phonetisch motivierter Ansatz gewählt, bei dem die Hidden–Markov–Modelle für Wörter aus Modellen für Phoneme aufgebaut werden ([Bub, 1999], [Köhler, 2000]). Die Wissensquelle, die für jedes Wort die Aussprache in Form der zugehörigen Phonemfolge liefert, wird *phonetisches Lexikon* oder *Aussprachelexikon* ([Ziegenhain u. a., 1998]) genannt. Im Rahmen dieser Arbeit wird das *Spicos*–Phoneminventar ([Höge, 1990], [Ney und Noll, 1994]) verwendet, das 39 verschiedene Phoneme enthält.

Ein weiterer Vorteil eines phonetischen Ansatzes besteht in der Möglichkeit, durch die Verkettung von einzelnen Phonemmodellen beliebige Wörter modellieren zu können. Man bezeichnet diese Fähigkeit des Systems auch als *type–in*–Fähigkeit, da neue Wörter durch Eintippen dem System hinzugefügt werden können. Prinzipiell ist ein Eintippen der Lautsprache (Phoneme) erforderlich. Es existieren jedoch Ansätze zur automatischen Transformation von Rechtschrift in Lautschrift ([Ziegenhain u. a., 1998]), durch die in der Praxis die Spezifikation der Rechtschrift ausreichend ist.

Hintergrundphonem Neben den Phonemen für sprachliche Laute wird im untersuchten System zusätzlich das Hintergrund–„phonem“ *si* (engl. **si**lence) verwendet. Dieses spezielle „Pho-

nem“ besteht aus nur einem HMM–Zustand und modelliert die Sprachpausen.

Kontextunabhängige Phonemmodelle Im einfachsten Fall wird ein Modell für ein Wort aus der Folge von Phonemen entsprechend der Aussprache gebildet. In der verwendeten Notation besteht ein Phonem nun wiederum aus 3 Segmenten. In der Abbildung 2.2 ist z.B. das Phonem /a/ aus den Segmenten a.0, a.1 und a.2 aufgebaut. Ein Segment besteht nun wiederum aus zwei HMM–Zuständen, wobei diese mit einer identischen Emissionswahrscheinlichkeit modelliert werden (vgl. Abschnitt. 2.3.2). Das erste Segment eines Phonems kann auch als Anlaut, das zweite als Mittelsegment und das dritte Segment als Auslaut bezeichnet werden. Ein Phonem besteht hier also aus 6 Zuständen. Im Fall von kontextunabhängigen Phonemmodellen benötigt man also ca. $3 \cdot 39$ verschiedene Emissionswahrscheinlichkeiten, mit deren Hilfe beliebige Wörter modelliert werden können.

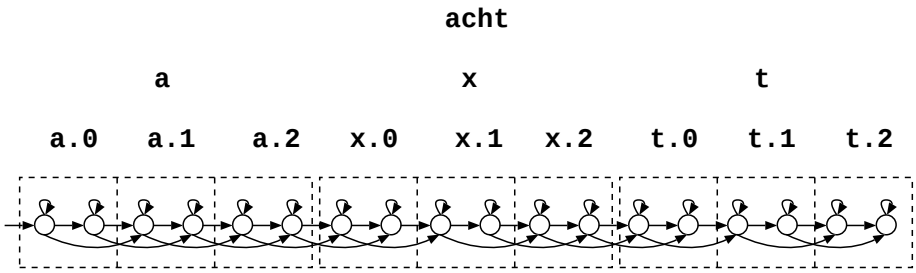

Abbildung 2.2: *Hierarchischer Aufbau eines HMM für das Wort* acht *aus kontextunabhängigen Phonemen, Segmenten und HMM–Zuständen.*

Kontextabhängige Phonemmodelle Die Beeinflussung von Lauten durch benachbarte Laute ist in natürlicher Sprache quasi immer gegeben. Diesem als *Koartikulation* bezeichneten Phänomen kann in Systemen zur automatischen Spracherkennung durch sogenannte kontextabhängige Phonemmodelle ([Schwartz u. a., 1985], [Lee, 1990]) Rechnung getragen werden. Hierbei werden Phoneme in Abhängigkeit von benachbarten Phonemen unterschiedlich modelliert. Der Grad der Kontextabhängigkeit ist hier durch die Anzahl der die Aussprache beeinflussenden benachbarten Phoneme gegeben. Wird zum Beispiel ein Phonem in Abhängigkeit des vorangehenden und des nachfolgenden Phonems modelliert, so spricht man von *Triphonen* ([Young und Woodland, 1993]).

Wird nun jedes Phonem in Abhängigkeit von zwei anderen beliebigen Phonemen realisiert, so kommt man auf eine Gesamtzahl von ca. 40^3 Phonemmodellen. Bei einer solch großen Zahl von Sprachuntereinheiten ist nun die Schätzbarkeit der Parameter bei im allgemeinen endlicher Trai-

ningsmenge nicht gegeben. Um die Anzahl der zu schätzenden Parameter zu reduzieren, werden sogenannte *tying*–Techniken (engl. Verknüpfen/Verkleben) eingesetzt, durch die Emissionswahrscheinlichkeiten zwischen mehreren Phonemen bzw. Segmenten geteilt werden. In dem vorliegenden System werden zwei Methoden zum tying von Triphonsegmenten eingesetzt ([Bauer, 2000]). Zum einen wird ein Regelwerk verwendet, das besagt, daß Anlaut–Segmente nur in Abhängigkeit des vorangehenden Phonems, das Mittelsegment ganz ohne Abhängigkeiten und der Auslaut nur in Abhängigkeit des nachfolgenden Phonems modelliert wird ([Zünkler, 1991]). In der Abbildung 2.3 ist der Aufbau des Wortes *acht* aus kontextabhängigen Segmenten entsprechend der beschriebenen Regeln in einer speziell darauf abgestimmten Notation dargestellt.

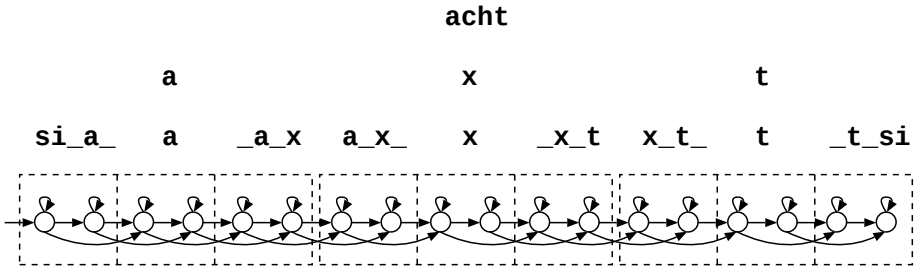

Abbildung 2.3: *Hierarchischer Aufbau eines HMM für das Wort* acht *aus Phonemen und kontextabhängigen Segmenten.*

Durch die Verwendung des Regelwerks ergibt sich eine Anzahl von $40 + 40^2$ Segmenten. Da in der Praxis auch diese kleinere Anzahl von Modellen nicht sinnvoll geschätzt werden können, wird ein weiteres, datengetriebenes Verfahren zum tying verwendet. Hierbei werden alle zu einem Phonem und einer Position (Anlaut, Mittelsegment, Auslaut) gehörenden Segmente, die im Trainingsmaterial mit einer geringen Häufigkeit gesehen werden, in einem Sammelsegment zusammengefaßt ([Bauer, 2000]).

2.3.4 Ganzwort–Modellierung

Inwieweit ein auf sprachlichen Untereinheiten wie Phonemen basierender Ansatz von Vorteil sein kann, ist im wesentlichen von der Menge des zur Schätzung der Modellparameter zur Verfügung stehenden Trainingsmaterials abhängig. Für eine Menge von Anwendungen, wie z.B. die Erkennung von Ziffernwörtern, steht jedoch Trainingsmaterial in einem solchem Umfang zur Verfügung, daß auf die Verwendung von phonetischen Sprachuntereinheiten verzichtet werden kann. Man spricht von Ganzwort–Modellierung, wenn die Modelle für die einzelnen Wörter aus Zuständen aufgebaut sind, die nur in dem jeweiligen Wort und an der jeweiligen Position ver-

wendet werden.

In der Praxis läßt sich eine Ganzwortmodellierung einfach erreichen, indem ein Wort aus speziellen Ganzwort–„Phonemen“ aufgebaut wird, die genau das zu modellierende Wort und die Position beschreiben. Damit läßt sich eine konsitente Darstellung für phonetische Modellierung und Ganzwortmodellierung realisieren. In Abbildung 2.4 ist diese Vorgehensweise am Beispiel der Wortes *acht* erklärt. Bei der in Abbildung 2.4 dargestellten Modellierung erhält das Wort *acht*

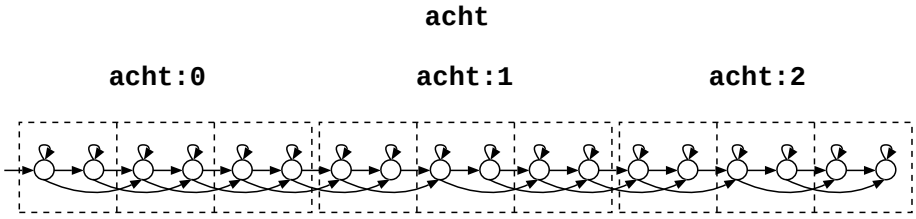

Abbildung 2.4: *Aufbau eines Ganzwort–HMM für das Wort* acht *aus speziellen Ganzwort–„Phonemen“.*

ebensoviele Ganzwort–„Phoneme“, wie es (richtige) Phoneme hat. Diese Einteilung ist natürlich nicht zwingend. Für verschiedene Anwendungen erweist es sich sogar außerordentlich günstig eine größere Anzahl von Ganzwort–„Phonemen“ zu verwenden (vgl. Abschnitt 6.13.3).

2.4 Modellierung der Verteilungs–Dichten

Den stärksten Bezug zu den zu modellierenden Merkmalsvektorfolgen erhält ein Hidden–Markov–Modell durch die Emissionswahrscheinlichkeiten $b_s(\vec{x})$,

$$b_s(\vec{x}) = p(\vec{x}|s) \tag{2.9}$$

die die Emission/Aussendung von Merkmalsvektoren in einem Zustand s modellieren.

2.4.1 Misch–Verteilungen

Die Emissionswahrscheinlichkeiten werden im allgemeinen durch Mischverteilungen realisiert. Hierbei wird eine gewichtete Linearkombination aus mehreren Basisfunktionen gebildet:

$$b_s(\vec{x}) = \sum_{m=1}^{M_s} c_{s,m} \cdot p_{s,m}(\vec{x}) \tag{2.10}$$

Die einzelnen Basisfunktionen $p_{s,m}(\vec{x})$ bezeichnet man auch als *Moden*. M_s ist die Anzahl der Moden im Zustand s. Die Gewichtung der Moden erfolgt hierbei über die Mixturkoeffizienten $c_{s,m}$, für die gelten muß:

$$\sum_{m=1}^{M_s} c_{s,m} = 1 \tag{2.11}$$

Im einfachsten Fall ist $M_s = 1$. Dann spricht man von sogenannten *single densities* Modellen.

Bei entsprechend hoher Dimensionalität der Merkmalsvektoren und endlicher Zahl von Moden zur Modellierung einer Emissionswahrscheinlichkeit läßt sich die gewichtete Summe der Moden durch das Maximum aus Modenwahrscheinlichkeit und Mixturkoeffizient annähern:

$$b_s(\vec{x}) \approx \max_m \{c_{s,m} \cdot p_{s,m}(\vec{x})\} \tag{2.12}$$

Dabei wird die Annahme zugrundegelegt, daß es in der Summe einen dominierenden Summanden gibt. Dies kann durch die wenig dichte Besetzung des hochdimensionalen Raums mit Basisfunktionen begründet werden. Durch diese Vorgehensweise wird zum einen der Rechenaufwand reduziert und zum anderen lassen sich Ableitungen für unimodale Verteilungsdichten meist unverändert auf Mischverteilungen übertragen. In dieser Arbeit wird immer die beschriebene Näherung verwendet.

2.4.2 Basisfunktionen

Als Basisfunktionen werden in der automatischen Spracherkennung mit Hidden–Markov–Modellen fast ausschließlich *Gauß*–Verteilungen eingesetzt:

$$p_{s,m}(\vec{x}) = N(\vec{x}, \vec{\mu}_{s,m}, \Sigma_{s,m}) \tag{2.13}$$

Dabei ist $N(\vec{x}, \vec{\mu}_{s,m}, \Sigma_{s,m})$ eine Normalverteilung für $\vec{x}$ mit Mitellpunktsvektor $\vec{\mu}_{s,m}$ und Kovarianzmatrix $\Sigma_{s,m}$.

$$N(\vec{x}, \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\vec{\mu})} \tag{2.14}$$

Hierbei werden ausschließlich diagonal besetzte Kovarianzmatrizen $\Sigma_{s,m}$ verwendet:

$$\Sigma_{s,m} = diag(\sigma^2_{s,m,d}) \tag{2.15}$$

Die Verwendung der nur diagonal besetzten Kovarianzmatrizen ist zumindest teilweise bei Verwendung komponentenweise nicht–korrelierter Merkmalsvektoren gerechtfertigt. Diese Eigenschaft kann durch Anwendung der Linearen Diskriminanz–Analyse (LDA) erreicht werden. Der Hauptgrund, daß sich die Verwendung von diagonalen Kovarianzmatrizen in der Spracherkennung mit Hidden–Markov–Modellen durchgesetzt hat, ist jedoch wahrscheinlich in der schlechten Schätzbarkeit von Varianzen zu suchen: rein diagonal besetzte Matrizen haben im Vergleich zu voll besetzten Matrizen eine erheblich reduzierte Anzahl von Parametern.

Bei allen beschriebenen Versuchen wird weiterhin ein einziger globaler Varianzparameter σ_0 verwendet:

$$\sigma_{s,m,d} = \sigma_0 \tag{2.16}$$

Experimentelle Untersuchungen haben gezeigt, daß diese einfache Art der Varianzmodellierung in Verbindung mit der LDA zu optimalen Erkennungsleistungen führt. Als theoretische Begründung hierfür läßt sich die Eigenschaften der *whitening*–transformierten Merkmalsvektoren nach der Anwendung der LDA aufführen (vgl. Abschnitt 5).

2.4.3 Kontinuierliche und semikontinuierliche Modelle

In Gleichung (2.10) wurden die Basisfunktionen als spezifisch für einen Zustand gewählt: $p_{s,m}(\vec{x})$. Für diesen Fall der zustandsspezifischen Basisfunktionen spricht man von kontinuierlichen Verteilungsdichten (engl. *Continuous Densities* HMM).

Als semikontinuierliche (engl. *Semi Continuous* Densities HMM) Verteilungsdichten bezeichnet man den Fall, wenn allen Zuständen ein gemeinsamer Satz von Basisfunktionen zugrundegelegt wird: $p_m(\vec{x})$ ([Huang u. a., 1990], [Bellegarda und Nahamoo, 1990], [Duchateau u. a., 1998]). Die Mischverteilungen der Zustände unterscheiden sich dann nur noch durch die Mixturkoeffizienten:

$$b_{s(\vec{x}),\,\mathrm{SCDHMM}} = \sum_{m=1}^{M} c_{s,m} \cdot p_m(\vec{x}) \tag{2.17}$$

In diesem Fall spricht man auch von einem *codebook* mit M codebook–Vektoren. Der Vorteil der semikontinuierlichen Modellierung besteht in erster Linie in einer reduzierten Anzahl von Systemparametern, was wiederum das Schätzproblem für die Parameter der Basisfunktionen entschärft. Bei dieser Art der Modellierung ist die Anzahl der Mixturkoeffizienten meist erheblich höher und die Gesamtzahl der Basisfunktionen meist erheblich niedriger, als bei kontinuierlicher Dichtenmodellierung.

Kapitel 3

Decodierung von Hidden–Markov–Modellen

Wie in Abschnitt 2.2 ausgeführt, werden Hidden–Markov–Modelle in der automatischen Spracherkennung dazu verwendet, Wahrscheinlichkeiten von Merkmalsvektorfolgen für Wortfolgen zu modellieren. Bei ihrer Anwendung zur Klassifikation von Sprachabschnitten (repräsentiert durch ihre Merkmalsvektorfolge) besteht nun die Aufgabe in der Bestimmung der Wortfolge mit der höchsten Erzeugungswahrscheinlichkeit nach Gleichung (2.3):

$$W_{\text{Bayes, HMM}} = \underset{W}{\operatorname{argmax}}\, P(X|\lambda_W)\hat{P}(W) \tag{3.1}$$

Das Prizip dieser *globale Suche* ([Ney, 1984], [Hauenstein, 1993], [Plannerer, 1995]) genannten Aufgabenstellung ist in Abbildung 3.1 grafisch veranschaulicht (nach [Ney, Sommersemester 1995]).

Geht man zunächst von einem Sprachmodell aus, das für alle Wortfolgen W dieselbe Wahrscheinlichkeit $\hat{P}(W)$ liefert, so reduziert sich die Aufgabe auf die Bestimmung der Erzeugungswahrscheinlichkeiten $P(X|\lambda_W)$, gefolgt von einer Maximumbestimmung.

Legt man nun für ein HMM die Menge möglicher Zustandsabfolgen $\{\Theta_o\}$ für eine Merkmalsfolge der Länge $T(X)$ in einem HMM zugrunde, so berechnet sich die Erzeugungswahrscheinlichkeit aus der Summe der Wahrscheinlichkeiten für die verschiedenen Zustandsfolgen:

$$P(X|\lambda_W) = \sum_{\Theta \in \{\Theta_o\}} P(X|\lambda_W, \Theta) \tag{3.2}$$

Die Erzeugungswahrscheinlichkeit bei gegebener Zustandsfolge

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_{T(X)}\} \tag{3.3}$$

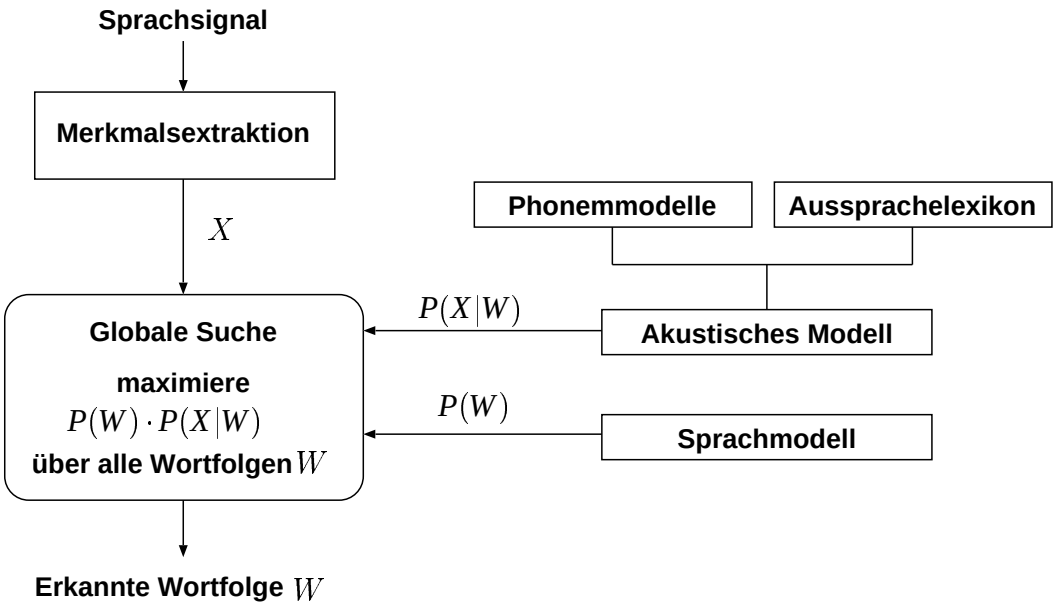

Abbildung 3.1: *Aufgabenstellung globale Suche*

kann als Produkt von Übergangswahrscheinlichkeiten und Emissionswahrscheinlichkeiten berechnet werden:

$$P(X|\lambda_W,\Theta) = \pi_{\theta_1} b_{\theta_1}(\vec{x}_1) \cdot \prod_{t=1}^{T(X)-1} a_{\theta_t,\theta_{t+1}} \cdot b_{\theta_{t+1}}(\vec{x}_{t+1}) \tag{3.4}$$

Da im Ausdruck (3.2) über alle Zustandfolgen summiert und der Ausdruck (3.4) für jede Zustandsfolge berechnet werden muß, muß in der Praxis ein Algorithmus mit geringerer Komplexität eingesetzt werden. Neben dem im folgenden Abschnitt beschriebenen Viterbi–Algorithmus, der nur eine approximative Berechnung erlaubt, existiert noch die Methode der Berechnung mit Hilfe der Vorwärts– und Rückwärtswahrscheinlichkeiten. Diese exakte Methode ist von großer Bedeutung in Verbindung mit einer Maximum–Likelihood–Schätzung der HMM–Parameter nach dem Baum–Welch–Algorithmus ([Bahl u. a., 1983]). Da dieser Algorithmus in der vorliegenden Arbeit nicht verwendet wird, sei an dieser Stelle auf andere Literatur verwiesen ([Schukat-Talamazzini, 1995], [Rabiner, 1989], [Picone, 1990]).

3.1 Viterbi–Suche

Bei der Berechnung von Erzeugungswahrscheinlichkeiten mit Hilfe des Viterbi–Algorithmus wird die Summation in (3.2) auf eine Maximierung über alle möglichen Pfade reduziert:

$$P_{\text{Viterbi}}(X|\lambda_W) = \max_{\Theta} P(X|\lambda_W, \Theta) \tag{3.5}$$

Der Viterbi–Algorithmus ist nun in der Lage, in einem Schritt sowohl diese maximale Erzeugungswahrscheinlichkeit als auch den dahinterliegenden Pfad und somit die Wortfolge mit der maximalen Wahrscheinlichkeit zu bestimmen.

Grundalgorithmus Der Viterbi–Algorithmus basiert auf einer zeitsynchronen Abarbeitung aller HMM–Zustände. Als Hilfsvariable wird die sogenannte Vorwärtswahrscheinlichkeit $\alpha_s(t)$ eines Zustands s im Zeitschritt t verwendet. Zu jedem Zeitschritt t können nun die Vorwärtswahrscheinlichkeiten der Zustände aus den Vorwärtswahrscheinlichkeiten des letzten Zeitschritts, den Übergangswahrscheinlichkeiten und den Emissionswahrscheinlichkeiten berechnet werden. In jedem Zustand wird dabei nur jeweils der Vorgänger–Zustand berücksichtigt, für den die Vorwärtswahrscheinlichkeit im neuen Zeitschritt maximal wird. Zur Pfadrückverfolgung muß neben der Berechnung der Vorwärtswahrscheinlichkeiten auch eine Information über den Vorgänger–Zustand festgehalten werden.

Formal kann der Viterbi–Algorithmus wie folgt dargestellt werden:

- Initialisierung:

$$\alpha_s(1) = \pi_s \cdot b_s(\vec{x}_1) \tag{3.6}$$

$$\Psi_s(t) = 1 \tag{3.7}$$

- Rekursion:

$$\alpha_s(t) = b_s(\vec{x}_t) \cdot \max_{s'} \{\alpha_{s'}(t-1) \cdot a_{s',s}\} \tag{3.8}$$

$$\Psi_s(t) = \operatorname*{argmax}_{s'} \{\alpha_{s'}(t-1) \cdot a_{s',s}\} \tag{3.9}$$

- Finalisierung:

$$P_{\text{Viterbi}}(X|\lambda) = \alpha_{\hat{s}}(T), \text{ mit } \hat{s} \text{ als Endzustand von } \lambda \tag{3.10}$$

$$\Psi_{\hat{s}}(T), \Psi_{\Psi_{\hat{s}}(T)}(T-1), \ldots \text{(Pfadrückverfolgung)} \tag{3.11}$$

Nach der Initialisierung vor dem Zeitschritt $t = 1$ muß die Rekursion T mal (T ist die Anzahl der Merkmalsvektoren in der Merkmalsfolge X) durchgeführt werden. Nach dem Zeitschritt T kann an den Endzuständen der Wörter die maximale Wahrscheinlichkeit abgelesen und mit Hilfe der Marker $\Psi_s(t)$ derjenige Pfad rekonstruiert werden, für den die Erzeugungswahrscheinlichkeit ihr Maximum erreicht hat.

Verbundworterkennung In den bisherigen Ausführungen war stets von einem HMM λ_W für eine Wortfolge W die Rede. Bei einer Verbundworterkennung (kontinuierliche Erkennung) wäre es demnach notwendig, für alle möglichen Wortfolgen HMMs für die entsprechenden Wörter hintereinanderzuschalten. In der Praxis ist dies natürlich aufgrund der Komplexität dieser Modellvorstellung nicht möglich. Die Erkennung von Wortfolgen wird durch zusätzliche Übergänge im Suchraum, bestehend aus den HMMs für Einzelwörter wie in Abbildung 3.2 dargestellt, realisiert ([Ney u. a., 1992]). Der sogenannte *Suchraum* bezeichnet hier die Zustände und ihre möglichen Übergänge, die beim Such–Algorithmus abgearbeitet werden müssen. Durch die Zu-

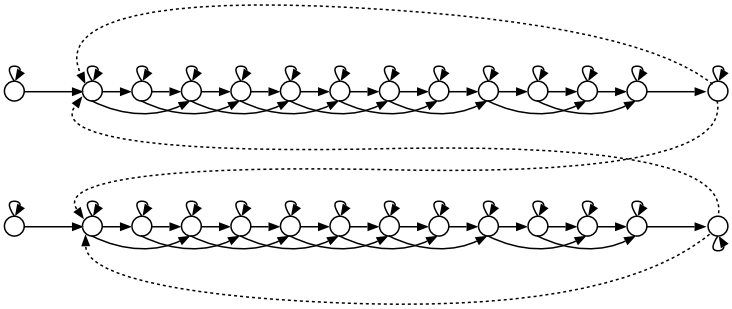

Abbildung 3.2: *Suchraum für zwei Wörter; vor und nach jedem Wort befindet sich ein Pausezustand; gestrichelt gezeichnete Übergänge sind im Verbundwortmodus notwendig.*

rückführung der Pfade aus den Pause–Zuständen der Wörter wird eine Verbundworterkennung ermöglicht, ohne die Komplexität gegenüber der Einzelworterkennung deutlich zu erhöhen. Die Darstellung in Abbildung 3.2 ist aus Gründen der Übersichtlichkeit insofern vereinfacht, als daß im realen System die Pause–Zustände auch übersprungen werden können.

Erzwungener Viterbi–Algorithmus Anders verhält es sich bei der Anwendung des Viterbi–Algorithmus für das Maximum–Likelihood–Training von Hidden–Markov–Modellen (vgl. Abschnitt 4). Dabei wird ein *erzwungener* (engl. *forced*) Viterbi–Algorithmus verwendet, dessen Aufgabe sich darauf beschränkt, den besten Pfad bei einer gegebenen Wortfolge zu ermitteln. In diesem Fall ist es ausreichend, den Suchraum aus der Hintereinanderschaltung von HMMs für

die einzelnen Wörter aufzubauen.

Pfadrückverfolgung auf verschiedenen Ebenen In der Darstellung von oben werden Marker zur Pfadrückverfolgung bis auf Ebene der HMM–Zustände benötigt. Je nach Anwendung können solche Marker auch weggelassen oder in reduzierter Form verwendet werden. Im einfachsten Anwendungsfall — der Einzelworterkennung — ist eine Pfadrückverfolgung gar nicht notwendig, da das Wort mit maximaler Wahrscheinlichkeit nur anhand des Endzustands eines Wortes mit der maximalen Vorwärtswahrscheinlichkeit identifiziert werden kann. Bei einer Verbundworterkennung ist zur Identifizierung der Wortfolge mit der höchsten Wahrscheinlichkeit im Prinzip die Information über den Pfad notwendig. Hier ist es jedoch ausreichend, Übergänge an Wortgrenzen zu berücksichtigen, da die Abfolge der Zustände innerhalb der Worte für das Klassifikationsergebnis nicht relevant ist. Bei der Schätzung von HMM–Parameter (Training) schließlich ist es notwendig, den Pfad bis auf die Ebene der Zustände zu rekonstruieren, da dort eine eindeutige Zuordnung von Merkmalsvektoren zu Zuständen benötigt wird.

Strahlsuche Unter den Begriff *Strahlsuche* oder auch *fokussierte Suche* versteht man eine Suche mit einer Eingrenzung auf eine reduzierte Menge von Pfaden. Hierbei werden alle Pfade, deren Wahrscheinlichkeit um einen bestimmten Faktor unter der Wahrscheinlichkeit des aktuell wahrscheinlichsten Pfads liegt nicht weiterverfolgt. Der Suchraum wird also um wenig wahrscheinliche Pfade beschnitten (engl. *prune*) — daher der englische Name für dieses Vorgehen: *pruning*. Im Prinzip stellt natürlich eine fokussierte Suche ein Abweichen vom Optimalitätsprinzip dar. In der Praxis hat sich jedoch gezeigt, daß sich bei geeigneter Wahl des Faktors der Suchaufwand deutlich reduziert werden kann, ohne daß die Ergebnisse nennenswerte Unterschiede zur nicht–fokussierten Suche zeigen ([Hauenstein, 1993]).

Bei den in dieser Arbeit dargestellten Experimenten wird die Strahlsuche sowohl zu Zwecken des Trainings (Parameterschätzung) als auch der Evaluierung von HMMs (Erkennung) eingesetzt. In beiden Fällen kann mit diesem Verfahren die notwendige Rechenzeit deutlich reduziert werden.

3.2 N–best Suche

Unter einer *N–best Suche* versteht man einen Suchalgorithmus, der neben der Wortfolge mit der maximalen Erzeugungswahrscheinlichkeit weitere mögliche Wortfolgen mit ihren Wahrscheinlichkeiten liefert ([Steinbiss, 1989], [Schwartz und Chow, 1990]). Das Ergebnis der Suche ist eine Menge von N Tupeln mit den N wahrscheinlichsten Wortfolgen und ihren Wahrscheinlichkeiten. Zu Zwecken der Parameterschätzung wird neben der Wortfolge und der Wahrscheinlichkeit auch der Pfad bis auf Zustandsebene benötigt (vgl. Kapitel 6).

Einzelwortsuche Ist bei einer reinen Einzelwortsuche der Suchraum wie in Abbildung 3.2 (ohne gestrichelte Übergänge) aufgebaut, so ist das Problem der N–best Suche im Grunde trivial. Dort ist es ausreichend, an den möglichen Endzuständen (Pause–Zustände nach den Wörtern) die Wahrscheinlichkeiten für die einzelnen Wörter abzulesen. Durch die Notwendigkeit einer N–best Suche erhöht sich hier der Aufwand für die Suche nicht.

Verbundworterkennung Ein völlig anderes Bild ergibt sich für eine kontinuierliche Suche (Suchraum wie in Abbildung 3.2 mit gestrichelten Übergängen). Dort gibt es Rekombinationspunkte, an denen potentiell Pfade mit unterschiedlicher Wort–Historie in Konkurrenz stehen. Treffen nun zwei Pfade mit unterschiedlicher Wort–Historie aufeinander, so wird nur der wahrscheinlichere weiterverfolgt und der andere geht unwiederbringbar verloren. Werden nun an den möglichen Endzuständen Such–Ergebnisse abgelesen, so ist in keinster Weise gesichert, daß sich die N wahrscheinlichsten Wortfolgen unter ihnen befinden.

In der Literatur werden prinzipiell zwei verschiedene Vorgehensweisen für die Generierung von N–best Wortfolgen vorgeschlagen: Einstufige Verfahren ([Steinbiss, 1989])und zweistufige Verfahren ([Schwartz und Austin, 1991]). Für die Untersuchungen in dieser Arbeit wird ein einstufiges Verfahren verwendet, das ohne Näherungen die N wahrscheinlichsten Wortfolgen liefert. Hierbei wird in allen Zuständen nicht nur der beste Pfad geführt, sondern die N besten Pfade mit unterschiedlicher Wort–Historie werden parallel geführt. Nach Abarbeitung aller Zeitschritte ist damit gesichert, daß sich die Pfade mit den N wahrscheinlichsten Wortfolgen an den Endzuständen ablesen lassen.

Durch die parallele Verarbeitung verschiedender Pfade in den Zuständen erhöht sich natürlich der rechnerische Aufwand für die Suche. Für die Untersuchungen in Kapitel 6 konnte die Anzahl der notwendigen Ergebnisse auf 2 festgelegt werden, was zu einer Erhöhung des Rechenaufwands für die Suche um deutlich weniger als den Faktor 2 führt.

3.3 Neg–Log–Transformation

Bei der Umsetzung von Hidden–Markov–Modellen auf Rechensystemen ist es fast unverzichtbar, statt mit normalen Wahrscheinlichkeiten mit logarithmisch transformierten Größen zu rechnen ([Hauenstein, 1993]). Die in den Ausdrücken für die Erzeugungswahrscheinlichkeiten auftretenden Produkte aus sehr vielen kleinen Wahrscheinlichkeiten würden selbst bei sehr langen Fließkommazahlen sehr schnell zu numerischen Problemen führen. Wird statt dessen in einem logarithmisch transformierten Bereich gerechnet, können Produkte von Wahrscheinlichkeiten auf Summen von logarithmierten Wahrscheinlichkeiten zurückgeführt werden. Damit vermeidet man sowohl den Umgang mit sehr kleinen Zahlenwerten als auch rechenintensive Multiplikationen.

Im Rahmen dieser Arbeit wird eine speziell skalierte logarithmierte Transformation verwendet, die zur Folge hat, daß sich die transformierten Emissionswahrscheinlichkeiten auf einen euklidischen Abstand reduzieren lassen. Die neg–log–transformierte Emissionswahrscheinlichkeit, der sogenannte Emissionsscore, wird wie folgt definiert:

$$B_s(\vec{x}) := -\sigma_0^2 \cdot \log b_s(\vec{x}) \tag{3.12}$$

Legt man nun Gauß–Verteilungen (vgl. Abschnitt 2.4.2), eine globale Varianz σ_0 entsprechend (2.16) und die Näherung aus (2.12) zugrunde, so ergibt sich folgender einfacher Ausdruck:

$$\begin{aligned} B_s(\vec{x}) &= \max_m\{-2\sigma_0^2 \log c_{s,m} + |\vec{x} - \vec{\mu}_{s,m}|^2\} + const && (3.13)\\ &= \max_m\{C_{s,m} + |\vec{x} - \vec{\mu}_{s,m}|^2\} + const && (3.14) \end{aligned}$$

Dabei ist

$$C_{s,m} = -2 \cdot \sigma_0^2 \cdot \log c_{s,m} \tag{3.15}$$

Der Term *const* in 3.13 bezeichnet einen Term, der nicht von den anderen Variablen (*s*, *m* und $\vec{x}$) abhängig ist. Setzt man voraus, daß absolute Wahrscheinlichkeiten nicht von Interesse sind, kann dieser konstante Term vernachlässigt werden. Für die Bestimmung der wahrscheinlichsten Wortfolge jedenfalls, spielen absolute Wahrscheinlichkeiten keine Rolle.

Die entscheidenden Terme in (3.13) sind also der neg–log–transformierte Mixturkoeffizient — auch Mixturstrafe ($C_{s,m}$) genannt — und der euklidische Abstand zwischen Merkmalsvektor und dem Mittelpunktsvektor der bestpassenden Mode. In der Praxis wird ein Großteil der für den Erkennungsvorgang notwendigen Rechenzeit in der Berechnung dieser euklidischen Abstände verbraucht.

Für die gesamte Erzeugungswahrscheinlichkeit einer Merkmalsfolge ergibt sich im neg–log–transformierten Bereich ausgehend von (3.4) der folgende Ausdruck:

$$\begin{aligned} -\sigma_0^2 \cdot \log P(X|\lambda_W, \Theta) &= \Pi_{\theta_1} + B_{\theta_1}(\vec{x}_1) \\ &\quad + \sum_{t=1}^{T-1} \{A_{\theta_t,\theta_{t+1}} + B_{\theta_{t+1}}(\vec{x}_{t+1})\} + const && (3.16)\\ &= \Pi_{\theta_1} + \sum_{t=1}^{T-1} A_{\theta_t,\theta_{t+1}} \\ &\quad + \sum_{t=1}^{T} B_{\theta_t}(\vec{x}_t) + const && (3.17) \end{aligned}$$

Dabei ist

$$\Pi_s := -\sigma_0^2 \cdot \log \pi_s \tag{3.18}$$

und

$$A_{s,s'} := -\sigma_0^2 \cdot \log a_{s,s'}. \tag{3.19}$$

3.4 Wortanfangsstrafe

Eine sehr probate Heuristik ist die Verwendung einer sogenannten *Wortanfangsstrafe* bei der kontinuierlichen Erkennung. Insbesondere bei der kontinuierlichen Erkennung ohne ein Sprachmodell ergibt sich in der Praxis mit Standardeinstellungen oft eine sehr hohe Worteinfügungsrate. Das heißt, in der Erkennung werden mehrere kurze Wörter gegenüber wenigen (langen) Wörtern bevorzugt. Die Wortanfangsstrafe wird nun in das Erkennungssystem eingebracht, indem ihr Wert immer zu Beginn eines neuen Wortes (Einsprung in den ersten Zustand eines Wortes) auf den akkumulierten Pfadscore aufaddiert wird. Mit Hilfe dieses Parameters läßt sich nun die Tendenz zu vielen oder wenigen Wörtern im Erkennungsergebnis leicht steuern.

In dieser Arbeit wird zur Einstellung des Parameters Wortanfangsstrafe folgende Vorgehensweise verwendet: Auf dem Trainingsmaterial wird die Wortanfangsstrafe iterativ solange verändert, bis die Einfügungsrate in etwa gleich der Auslöschungsrate entspricht. Der so erhaltene Wert wird dann unverändert zur Erkennung auf dem Testmaterial verwendet.

3.5 Sprachmodelle

Bereits in Abschnitt 2.1 wurde der Begriff des Sprachmodells ([Witschel, 1993]) eingeführt. Ein Sprachmodell beschreibt die Wahrscheinlichkeit $P(W)$ des Auftretens einer Wortfolge W. In der Spracherkennung mit Hidden–Markov–Modellen werden meist sogenannte N–gram Sprachmodelle eingesetzt. Dabei ergibt sich die Wahrscheinlichkeit $P(W)$ als Produkt bedingter Wahrscheinlichkeit, wobei die einzelnen Faktoren nur von N Wörtern abhängen. Bei Bi–gram Sprachmodellen z.B. hängt eine einzelne Wortwahrscheinlichkeit $P(w_1|w_2)$ nur vom aktuellen Wort w_1 und dem vorangegangenen Wort w_2 ab. Ein Bi–gram Sprachmodell kann leicht in einen Suchraum wie den in Abbildung 3.2 eingebracht werden. Dort würden die Wortübergangswahrscheinlichkeiten einfach an die zu den Wortanfängen zurückführenden Übergänge angetragen werden.

Neben dem Einbringen von höherem Wissen, das in etwa auf syntaktischer Ebene angesiedelt ist, liegt ein entscheidender Vorteil von Sprachmodellen in ihrer Fähigkeit, den Rechenaufwand für die Decodierung bei großem Wortschatz zu beschränken. Entscheidend ist hier die sogenannte Perplexität eines Sprachmodells. Die Perplexität gibt den mittleren Verzweigungsgrad eines Sprachmodells an. Der Verzweigungsgrad entspricht dem effektiv zu bearbeitenden Wortschatz, da die Anzahl der auf ein bestimmtes Wort möglicherweise folgenden Wörter im Mittel darauf beschränkt ist.

Kapitel 4

Maximum–Likelihood–Training von HMMs

4.1 Einführung

Als Standardverfahren für die Bestimmung der freien Parameter von Hidden–Markov–Modellen besitzt das Maximum–Likelihood (ML) Verfahren eine weite Verbreitung. Die Grundidee besteht hierbei in einer Maximierung der Erzeugungswahrscheinlichkeit einer Trainingsstichprobe mit den Merkmalsfolgen $\{X^r\}$:

$$\Lambda_{\mathrm{ML}} = \underset{\Lambda}{\operatorname{argmax}} \prod_{r=1}^{|\{X\}|} p(X^r|\lambda_{\Omega(X^r)}) \tag{4.1}$$

Die im Sinne von Maximum–Likelihood optimalen Parameter Λ_{ML} sind also diejenigen, für die die Erzeugungswahrscheinlichkeit ein Maximum annimmt.

Für die Bestimmung der optimalen Parameter in Sinne dieses Kriteriums existiert keine geschlossene Lösung. In der Praxis werden stets iterative Verfahren eingesetzt. Bei diesen iterativen Verfahren wird ein anfänglich suboptimaler Parametersatz immer wieder anhand einer Trainingsstichprobe nachgeschätzt, wobei das Optimierungsziel von Iteration zu Iteration besser erreicht wird, um dann in eine Sättigung zu gehen.

In den folgenden Abschnitten wird nun zunächst eine mögliche Bildung von Startwerten für die Parameter — sprich die Initialisierung — beschrieben. Diese Verfahren sind im allgemeinen eher heuristischer Natur und gehorchen keinem expliziten Optimierungsziel. Erst im Abschnitt 4.3 wird auf das iterative Verfahren zur Maximum–Likelihood–basierten Optimierung der Parameter eingegangen.

4.2 Initialisierung der Parameter

Die Initialisierung von Hidden–Markov–Modellen dient der Erstellung initialer Modellparameter, die als Startwerte für eine iterative Optimierung verwendet werden. Da die eigentliche Optimierung von diesen Startwerten nicht direkt abhängig ist, könnte man die Bedeutung der Initialisierung als gering einschätzen. In vielen Systemen werden auch Initialisierungen mit zufällig gewählten Parametern verwendet. Man spricht hier auch von einem *flat start*.

Soll eine nicht–zufällige Besetzung der Parameter erfolgen, so ist es notwendig, eine möglichst genaue Zuordnung der Zeitschritte zu den HMM–Zuständen zu verwenden. Wenn ein bestehendes Modell zur Verfügung steht, kann der Viterbi–Algorithmus dazu verwendet werden, diese Zuordnung automatisch zu erstellen. In allen anderen Fällen muß eine solche Zuordnung von Hand erstellt werden. Oft steht in der Praxis die Zuordnung der Zeit zu den HMM–Zuständen nicht unmittelbar zur Verfügung, sondern es sind lediglich die Phonem– oder Wortgrenzen gegeben. In diesen Fällen wird im beschriebenen System zwischen diesen Grenzen linear interpoliert; d.h. die Merkmalsvektoren werden äquidistant den Zuständen der Phoneme bzw. Wörter zugeordnet.

In dem dieser Arbeit zugrundeliegenden System werden zwei unterschiedliche Verfahren zur Bildung von initialen HMM–Parametern verwendet. Bei beiden Verfahren werden im allgemeinen mehr als nur eine Mode pro Segment erzeugt. Wird im Gegensatz dazu nur eine Mode pro Segment erzeugt, so kann man davon ausgehen, daß diese bei der iterativen Optimierung leicht im hochdimensionalen Raum verschiebbar ist. Bei der Erzeugung von vielen initialen Moden ist diese Annahme nicht sehr wahrscheinlich und der Initialisierung kommt eine größere Bedeutung zu. In den folgenden Abschnitten wird auf beide Verfahren zur Initialisierung näher eingegangen.

4.2.1 Einfache Initialisierung mit Auffüllen und Auslassen

Das grundsätzliche Vorgehen bei dieser Methode ist es, einige Merkmalsvektoren der Trainingsstichprobe direkt als Mittelpunktsvektoren von neuen Moden zu verwenden. Hierzu wird zunächst eine Statistik über das Auftreten von Zuständen in den zu den Trainingsstichproben korrespondierenden Modellen erstellt. Anschließend wird ausgehend von der Auftrittshäufigkeit eines Zustands N_s festgelegt, wieviele Moden dieser Zustand erhalten soll:

$$M_s = min(N_s; 2 + 4 \cdot \log_2 N_s) \tag{4.2}$$

Die Anzahl der für einen Zustand vorgesehenen Moden M_s wächst also logarithmisch mit der Größe des Datenmaterials, das zur Schätzung der Parameter in diesem Zustand zur Verfügung steht.

Steht nun fest, wieviele Merkmalsvektoren für die Initialisierung eines Zustands benötigt werden, so werden ausgehend von der Gesamtzahl der für den Zustand auftretenden Merkmals-

vektoren nach jedem verwendeten Vektor soviele Vektoren ausgelassen, daß sich die Zielzahl an Moden ergibt. Das Verfahren bildet also einen Teil der Trainingsstichprobe in Form der auftretenden Merkmalsvektoren direkt auf die Mittelpunktsvektoren $\vec{\mu}$ ab. Eine Mittelung über mehrere Merkmalsvektoren findet nicht statt, die Moden der Zustände werden alle gleich gewichtet:

$$c_{s,m} = \frac{1}{M_s} \tag{4.3}$$

4.2.2 Initialisierung mit Laufzeit–Clustering

Grundgedanke dieses Verfahren ist es, im Laufe der Initialisierung ausgehend von bestimmten Merkmalsvektoren Moden für die Zustände zu erzeugen, diese anhand von anderen Merkmalsvektoren nachzuschätzen und zum Ende der Initialisierung die Menge der erzeugten Moden wiederum zu reduzieren.

Erzeugen von Moden Die Merkmalsvektoren, welche neue Moden erzeugen sollen, sind zum einen die ersten für einen Zustand auftretenden und zum anderen diejenigen, deren Abstand von der nächsten bestehenden Mode des Zustands einen Schwellwert überschreitet:

$$\min_m |\vec{x} - \vec{\mu}_{s,m}|^2 > factor \cdot \sigma_0^2 \cdot D \quad \rightarrow \quad \text{neue Mode} \tag{4.4}$$

Es ergibt sich zunächst der frei wählbare Parameter $factor$, der empirisch optimiert werden muß. Da bei Verwendung der LDA (vgl. Abschnitt 5) die Varianz der Merkmalsvektoren immer normiert wird, stellt dies in der Praxis kein größeres Problem dar. Als günstig hat sich ein Wert von 1.5 erwiesen. Ist für einen Merkmalsvektor die Bedingung (4.4) nicht erfüllt, so wird der Mittelpunktsvektor der dem Merkmalsvektor nächsten Mode rekursiv nachgeschätzt, so daß die Mittelpunktsvektoren stets den gemittelten Vektor aus den zur (Nach–)Schätzung der Mode verwendeten Merkmalsvektoren darstellen. In der Praxis wird die mögliche Gesamtzahl von Moden für einen Zustand begrenzt, da nicht beliebig viele Moden pro Zustand erzeugt werden sollen. Bewährt hat sich eine Wahl dieser Obergrenze zu der zweifachen Anzahl von Moden pro Zustand, die die Modelle später tragen sollen.

Reduktion von Moden Oft besteht die Notwendigkeit, bei der Initialisierung einige oder sogar die meisten der zunächst erzeugten Moden wieder mit anderen Moden zusammenzufassen. Die Gründe hierfür können z.B. in den für eine Anwendung beschränkten Ressourcen liegen. Zum anderen erscheint es auch sinnvoll, Moden, die nur anhand weniger Merkmalsvektoren geschätzt worden sind, wieder zu entfernen. Hierbei wird eine Mode quasi entfernt, indem sie und die Mode des Zustands mit dem geringsten Abstand durch eine Mode ersetzt wird. Der Mittelpunktsvektor der neuen Mode wird durch eine gewichtete Summe der ursprünglichen Vektoren

gebildet:

$$\vec{\mu}_{\text{vereinigt}} = \frac{1}{N_{\text{alt1}} + N_{\text{alt2}}} (N_{\text{alt1}} \cdot \vec{\mu}_{\text{alt1}} + N_{\text{alt2}} \cdot \vec{\mu}_{\text{alt2}}) \tag{4.5}$$

N_{alt1} und N_{alt2} bezeichnen hier die Anzahl der Merkmalsvektoren durch deren Mittelung die Mittelpunktsvektoren entstanden sind. Das verwendete Verfahren erlaubt nun die Reduktion von Moden anhand mehrerer Kriterien. Das erste Kriterium ist die absolute Anzahl von Merkmalsvektoren $N_{s,m}$ aus denen der Mittelpunktsvektor der Mode entstanden ist. Das zweite Kriterium ist die relative Häufigkeit der Mode im Zustand:

$$c_{s,m} = \frac{N_{s,m}}{\sum_{m=1}^{M_s} N_{s,m}} \tag{4.6}$$

Das Verfahren erlaubt nun zum einen, untere Schranken in Form absoluter und relativer Häufigkeiten festzulegen, und zum anderen ist vorgesehen, eines der beiden Kriterien iterativ zu verändern, bis eine vorgegebene Anzahl von Moden erreicht ist. Diese Anzahl von Moden kann entweder global für alle Zustände oder pro Zustand gewählt werden.

In der Praxis werden die absoluten Schwellen so eingestellt, daß zunächst nur Moden mit sehr kleiner absoluten (ca. 10) Häufigkeit und relativen (ca. 0.0001) Häufigkeiten mit anderen Moden vereinigt werden. Als Kriterium, das verschärft wird, bis eine vorgegebene Modenzahl erreicht ist, erweisen sich die absoluten Häufigkeiten als geeignet. Nur bei stark inhomogenen Häufigkeiten der auftretenden Zustände müssen die relativen Häufigkeiten als Kriterium verwendet werden, da sonst zu selten auftretende Zustände kaum Moden und die häufig auftretenden Zustände sehr viele Moden erhalten.

4.3 Maximum–Likelihood–basiertes Viterbi–Training

In der vorliegenden Arbeit wird ausschließlich Maximum–Likelihood–Viterbi–Training verwendet, das einen Spezialfall des weithin verwendeten Baum–Welch–Trainings darstellt.

Beim Baum–Welch–Training erfolgt keine explizite Zuordnung von Merkmalsvektoren zu HMM–Zuständen, sondern es werden lediglich Wahrscheinlichkeiten für eine Emission in einem bestimmten Zustand berechnet. Beim Viterbi–Training reduzieren sich diese Wahrscheinlichkeiten zu einer festen Zuordnung von Merkmalsvektoren zu Zuständen in Form der Zustandsfolge Θ. Dies könnte auch in Wahrscheinlichkeiten mit den binären Werten 0 und 1 ausgedrückt werden. Mit der in Gleichung (2.12) getroffene Näherung findet sogar tatsächlich eine feste Zuordnung zu den Moden der Zustände statt.

Es wird eine Hilfsfunktion $\zeta_t(s,m)$ definieren, die angibt, ob zu einem Zeitpunkt t die Mode m des Zustands s dem Merkmalsvektor $\vec{x}_t$ zugeordnet wurde:

$$\zeta_t(s,m) = \begin{cases} 1 & \text{für } s = \theta_t \text{ und } m = \text{argmax}_{m'}\, c_{m',s} \cdot p_{s,m'}(\vec{x}) \\ 0 & \text{sonst} \end{cases} \tag{4.7}$$

4.3.1 Nachschätzung der Verteilungs–Parameter

Mit (4.7) lassen sich die Maximum–Likelihood–Schätzwerte für die nachzuschätzenden Mittelpunktsvektoren und Mixturkoeffizienten wie folgt ausdrücken:

$$\hat{\vec{\mu}}_{s,m} = \frac{1}{\sum_t \zeta_t(s,m)} \sum_t \zeta_t(s,m) \cdot \vec{x}_t \tag{4.8}$$

$$\hat{c}_{s,m} = \frac{1}{\sum_t \sum_{m'} \zeta_t(s,m')} \sum_t \zeta_t(s,m) \tag{4.9}$$

Die Neuschätzung des Mittelpunktsvektors $\hat{\vec{\mu}}_{s,m}$ ergibt sich also als Mittelung aller Merkmalsvektoren, die dem Zustand s und der entsprechenden Mode durch den Viterbi–Algorithmus zugeordnet werden. Der neue Mixturkoeffizient $\hat{c}_{s,m}$ ist identisch mit der relativen Wahrscheinlichkeit, daß beim Auftreten des Zustands s die spezielle Mode m ausgewählt wurde.

In der Praxis müssen die Summen in (4.8) und (4.9) natürlich über alle Äußerungen X^r der Trainingsstichprobe ausgeführt werden.

4.3.2 Löschen von Basisfunktionen

Ausgehend von den bei der Initialisierung (vgl. Abschnitt) 4.2) zum Vereinigen verwendeten Häufigkeits–Kriterien besteht auch bei der Iteration die Möglichkeit, bestimmte Moden zu entfernen. Der Unterschied zu dem bei der Initialisierung verwendeten Algorithmus besteht darin, daß die Moden nicht mit anderen vereinigt werden, sondern schlichweg gelöscht werden. Dies ist bei den Iterationen weniger kritisch, weil alle Merkmalsvektoren, die bei einer Iteration einer zu löschenden Mode zugeordnet werden, bei der nächsten Iteration auf einer der verbleibenden Moden zugeordnet werden.

Kapitel 5

Lineare Diskriminanz–Analyse

5.1 Grundprinzip und Optimierungsverfahren

Im Mittelpunkt der Linearen Diskriminanz–Analyse ([Ruske, 1988], [Fukunaga, 1972], [Duda und Hart, 1973]) steht eine Transformationsmatrix $\mathcal{A}^T$, mit der die ursprünglichen Merkmalsvektoren $\vec{y}$ in neue Merkmalsvektoren $\vec{x}$ linear transformiert werden:

$$\vec{x} = \mathcal{A}^T(\vec{y} - \bar{\vec{y}}) \tag{5.1}$$

Vor der Multiplikation mit der Transformationsmatrix wird im Merkmalsraum der Gesamtschwerpunkt $\bar{\vec{y}}$ abgezogen ([Hauenstein und Marschall, 1995]). Dadurch kann ein z.B. auf 8 Bit begrenzter Zahlenbereich von -128 bis +127 bei einer um den Mittelpunkt symetrischen Verteilung optimal ausgenüzt werden. Bei $\mathcal{A}^T$ handelt es sich um eine Matrix der Dimension $D_y \times D_x$, wobei D_y die Dimension des ursprünglichen Merkmalsvektors $\vec{y}$ und D_x die Dimension des neuen Merkmalsvektors $\vec{x}$ darstellt. Zunächst gelte: $D_x \leq D_y$. Aufgabe der LDA ist es nun, die Transformationsmatrix so zu bestimmen, daß die neuen Merkmale einem bestimmten Optimalitätskriterium genügen.

Die Ziele, die durch die Transformation mit der LDA–Matrix verfolgt werden, sind:

1. Dekorreliertheit der neuen Merkmale

2. Maximale diskriminante Information bleibt in $\vec{x}$ erhalten für den Fall einer reduzierten Dimensionalität $D_x < D_y$

Diese Ziele lassen sich anhand der sogenannten Scatter–Matrizen formulieren. Gegeben seien zunächst J unterschiedliche Klassen, die in einem Klassifikationsprozeß unterscheidbar sein sollen. Es wird die Annahme zugrunde gelegt, daß die Merkmalsvektoren der einzelnen Klassen

durch eine unimodale Gaußverteilung beschreibbar sind. Die within–Scatter–Matrix ist wie folgt definiert:

$$\mathcal{S}_w = \sum_{j=0}^{J} P(j)\Sigma_j \tag{5.2}$$

Σ_j ist die Kovarianzmatrix der zur Klasse j zugehörigen Merkmalsvektoren:

$$\Sigma_j = E\{(\vec{x}_j - \bar{\vec{x}}_j)(\vec{x}_j - \bar{\vec{x}}_j)^T\} \tag{5.3}$$

$E\{\cdot\}$ steht hierbei für den Erwartungswert. $\bar{\vec{x}}$ stellen die Mittelpunktsvektoren der einzelnen Klassen dar. Die within–Scatter–Matrix kann auch als mittlere Kovarianzmatrix bezeichnet werden.

Die between Scatter–Matrix $\mathcal{S}_b$ wird ausgehend von der Verteilung der Klassenschwerpunkte $\bar{\vec{x}}_j$ definiert:

$$\mathcal{S}_b = \sum_{j=0}^{J} P(j)(\bar{\vec{x}}_j - \bar{\vec{x}})(\bar{\vec{x}}_j - \bar{\vec{x}})^T \tag{5.4}$$

$\bar{\vec{x}}$ bezeichnet den globalen Schwerpunkt und $P(j)$ die Auftrittswahrscheinlichkeit einer Klasse j.

Mit Hilfe der Scatter–Matrizen läßt sich folgende Zielfunktion über die Spur $\mathrm{tr}(\cdot)$ des Matrixproduktes definieren:

$$\mathrm{tr}(\mathcal{S}_w^{-1}\mathcal{S}_b) \tag{5.5}$$

Die Scatter–Matrizen in diesem Ausdruck beziehen sich auf den transformierten Merkmalsraum ($\vec{y}$). Auf den Hintergrund dieser Zielfunktion wird erst in den weiteren Ausführungen eingegangen, damit er leichter verständlich wird. In der Literatur ([Fukunaga, 1972], [Hojas, 1994]) finden sich weitere verwandte Zielfunktionen, die hier nicht näher diskutiert werden sollen.

Im weiteren wird davon ausgegangen, daß sich die Transformation mit der Transformationsmatrix $\mathcal{A}^T$ aus mehreren einzelnen Transformationen zusammensetzt ([Hojas, 1994]):

$$\mathcal{A} = \mathcal{U} \cdot \mathcal{W} \cdot \mathcal{V} \tag{5.6}$$

Im folgenden sollen die drei Transformationsmatrizen $\mathcal{U}$, $\mathcal{W}$ und $\mathcal{V}$ mit folgender Bedeutung entwickelt werden:

- $\mathcal{U}$ führt zu einer Dekorrelation der Merkmale
- $\mathcal{W}$ normiert die Varianzen (whitening–Transformation)
- $\mathcal{V}$ maximiert die Trennbarkeit der Klassen

Betrachtet man die erste Forderung von oben nach mittlerer Dekorreliertheit, so erkennt man leicht, daß das einer Forderung nach einer within–Scatter–Matrix in Diagonalform gleichkommt.

Dies führt zu einer ersten Transformationsmatrix $\mathcal{U}$, die aus den Eigenvektoren der within–Scatter–Matrix des ursprünglichen Merkmalsraumes gebildet wird:

$$\mathcal{U} = \mathrm{EV}\{S_w\} \tag{5.7}$$

$\mathrm{EV}(\cdot)$ bezeichnet dabei das Eigenwertsystem, also die Matrix gebildet aus den Eigenvektoren der Matrix. Zur genauen Erklärung des Eigennwertproblems sei auf die Literatur verwiesen ([Fukunaga, 1972], [Duda und Hart, 1973]).

In [Fukunaga, 1972] findet sich der Hinweis, daß die Zielfunktion (5.5) für den Fall, daß $\mathcal{B}^T \mathcal{S}_w^{-1} \mathcal{B} = I$ (I ist die Einheitsmatrix) sich zu

$$J' = \mathrm{tr}(\mathcal{B}^T \mathcal{S}_b \mathcal{B}) \tag{5.8}$$

reduzieren läßt. Da durch die erste Transformationsmatrix $\mathcal{U}$ der Ausdruck $\mathcal{U}^T \mathcal{S}_b \mathcal{U}$ schon Diagonalform besitzt, kann durch die sogenannte whitening–Transformation erreicht werden, daß die within–Scatter–Matrix zur Einheitsmatrix wird. Dazu wird eine zweite Transformationsmatrix $\mathcal{W}$ verwendet, die Diagonalform besitzt und deren Elemente die reziproken Werte der Standardabweichungen der einzelnen Merkmalskomponenten darstellen. Diese berechnen sich aus den Eigenwerten von $\mathcal{S}_w$ potenziert mit $-\frac{1}{2}$:

$$\mathcal{W} = diag(\mathrm{EW}_d\{\mathcal{S}_w\}^{-\frac{1}{2}}) \tag{5.9}$$

Dieser Zusammenhang resultiert aus der Äquivalenz der Eigenwerte $\mathrm{EW}_d\{\mathcal{S}_w\}$ mit den Varianzen der einzelnen Merkmalskomponenten.

Die beiden ersten Transformationen führen nun zu der gewünschten Eigenschaft, daß die within–Scatter–Matrix im transformierten Raum zur Einheitsmatrix wird. $\mathcal{B}$ bezeichne nun die Zusammenfassung der beiden ersten Transformationen:

$$\mathcal{B} = \mathcal{U}\mathcal{W} \tag{5.10}$$

Die vereinfachte Zielfunktion (5.8) besagt nun, daß die Spur — also die Summe der Diagonalelemente — der between Scatter–Matrix nach den ersten beiden Transformationen zu maximieren ist. Anschaulich kann dies als Versuch interpretiert werden, die Mittelpunkte der Klassenverteilungen maximal zu entzerren. Dies führt wiederum auf ein Eigenwertproblem, dessen Lösung eine dritte Transformationsmatrix $\mathcal{V}$ ist:

$$\mathcal{V} = \mathrm{EV}\{\mathcal{B}^T \mathcal{S}_b \mathcal{B}\} \tag{5.11}$$

Wichtig hierbei ist, daß die Eigenvektoren in $\mathcal{V}$ entsprechend der Eigenwerte geordnet sind. Nur so werden die Komponenten von $\vec{x}$ entsprechend ihrem diskriminativen Gehalt sortiert. Die Streuung und damit der Abstand der Klassenmittelpunkte entlang der neuen Koordinatenachsen entspricht dem Eigenwert zu dem Eigenvektor, der diese Koordinatenachse aufspannt.

5.2 Anwendung für die Spracherkennung

5.2.1 Vorteile der Transformation für HMM–Systeme

Bei einem automatischen Spracherkennungssystem, das auf HMMs basiert, sind mehrere Eigenschaften einer Merkmalstransformation mittels LDA von Vorteil.

Unkorreliertheit und einheitliche Varianz Zunächst kommt die Eigenschaft der im Mittel unkorrelierten Merkmalskomponenten der Verwendung von diagonalen Kovarianzmatrizen für die Modellierung der Emissionswahrscheinlichkeiten entgegen. Die Merkmalskomponenten sind zwar nur im Mittel unkorreliert, der Fehler, der durch die diagonalen Kovarianzmatrizen entsteht, wird jedoch je nach ursprünglichem Merkmalsvektor dramatisch reduziert. Wird der Merkmalsvektor zum Beispiel aus den Energien von überlappenden Filterbänken gebildet, so ist die Korrelation dieser Komponenten naturgemäß hoch. Durch die whitening–Transformation sind weiterhin die Varianzen der Klassenverteilungen im Mittel über alle Komponenten gleich. Dies reduziert den Fehler, der durch die Verwendung von nur einer einzigen für alle Komponenten gültigen Varianz (vgl. Abschnitt 2.4.2) entsteht.

Ordnung der Komponenten nach diskriminativer Information Ein weiterer entscheidender Vorteil ist die Kompression der diskriminativen Information in den vorderen Komponenten des transformierten Merkmalsvektors ([Eisele u. a., 1996]). Die Transformationsmatrix, welche zunächst die Dimension $D_y \times D_y$ hat, kann um $D_y - D_x$ Zeilen verkürzt werden, was auf einem Merkmalsvektor mit der Dimension $D_x < D_y$ führt. In der Praxis kann somit der Merkmalsvektor ohne einen Verlust an Erkennungsgenauigkeit um einiges verkürzt werden. Dies hat neben dem Vorteil einer beschleunigten Berechnung der Emissionswahrscheinlichkeiten den Vorteil einer besseren Schätzbarkeit, da die Anzahl der freien Systemparameter reduziert wird.

5.2.2 Verwendung von Supervektoren

Wie schon ausgeführt besteht eine Einschränkung von Hidden–Markov–Modellen in der mangelnden Berücksichtigung der Korrelation der Merkmalsvektoren über der Zeit ([Wellekens, 1987], [Junkawitsch und Höge, 1998], [Chengalvarayan und Deng, 1998]). Ähnlich wie bei der Merkmalsextraktion, wo Ableitungen der Merkmalskomponenten dem Merkmalsvektor hinzugefügt werden, kann der Merkmalsvektor bei Verwendung der LDA nochmals erweitert werden ([Haeb-Umbach u. a., 1993], [Aubert u. a., 1993]). Ausgehend von einer Merkmalsfolge $Y' = \{\vec{y'}_t\}$ wird eine neue Merkmalsfolge Y als Eingabe für die LDA gewonnen, indem immer L benachbarte Merkmalsvektoren (erste und zweite Ableitungen inklusive) zu sogenannten Supervektoren zusammengefügt werden. Für einen Wert von $L = 2$ wird der Supervektor $\vec{y}_{t=1}$ durch

Zusammenfügung der Vektoren $\vec{y'}_{t=1}$ und $\vec{y'}_{t=2}$ gebildet, $\vec{y}_{t=2}$ aus $\vec{y'}_{t=2}$ und $\vec{y'}_{t=3}$ und so fort (vgl. Abbildung 5.1). Der Informationsgehalt der Eingangsmerkmale für die LDA–basierte Transfor-

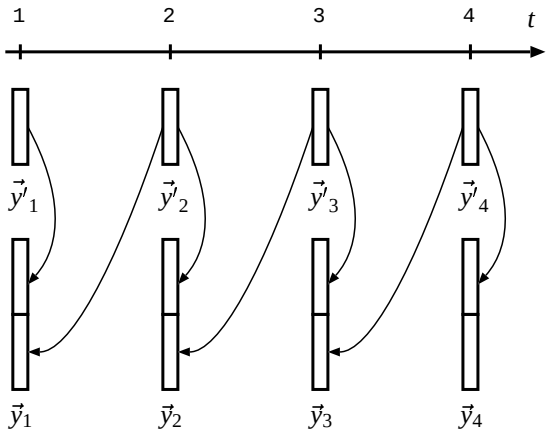

Abbildung 5.1: *Bildung der Supervektorenfolge Y aus Y'*

mation wird dadurch potentiell erhöht. Dies muß aber bei entsprechender Verkürzung des Ausgabevektors auf $D_x < D_y$ Komponenten nicht zu einer Vergrößerung des Merkmalsvektors und somit zu einer erhöhten Komplexität für den Klassifikator führen. Der Rechenaufwand für die lineare Transformation basierend auf der LDA steigt zwar linear mit der Dimension D_y, doch ist dieser wiederum relativ klein gegenüber dem Rechenaufwand für die Emissionswahrscheinlichkeiten, welcher nur von der Dimension der Vektoren nach der Transformation D_x abhängig ist.

5.2.3 Wahl der Klassen

Wie im Abschnitt 5.1 schon ausgeführt, ist eine der Voraussetzungen der LDA die Annahme, daß sich die Merkmalsvektoren einer Klasse durch eine unimodale Gaußverteilung beschreiben lassen. Somit kommen für die Wahl der Klassen nur Sprachabschnitte von kurzer Zeitdauer in Frage, für die Stationarität der Merkmalsvektoren angenommen werden kann. Zum Beispiel sind bei einem Einzelwortsytem zwar zunächst die Klassen, die unterscheidbar sein sollten, durch die Wörter definiert, für diese Wahl der Klassen kann aber in keinem Fall eine Stationarität angenommen werden. Aus diesem Grund werden bei der LDA für Spracherkennung meist kleine Einheiten wie Phoneme als Klassen verwendet. Oft werden auch noch kleinere Abschnitte wie Phonemsegmente verwendet ([Haeb-Umbach und Ney, 1992]).

Es stellt sich die Frage, welche Einheiten als Klassen für die LDA am besten geeignet sind. Zum einen sollen die zu den Klassen gehörigen Sprachabschnitte hinreichend kurz sein, damit

die Verteilung durch eine unimodale Gaußverteilung gut beschrieben werden kann. Zum anderen wiederspricht die Verwendung von kurzen Abschnitten der durch die Aufgabenstellung gegebenen Klassen wie z.B. Wörtern.

Besonders deutlich wird diese Problematik bei der Verwendung von Ganzwortmodellen. Hierbei wird unter Umständen ein fast identischer Laut durch zwei Klassen, nämlich Segmenten von Pseudo–Phonemen, modelliert. Dies trifft z.B. für die ersten Zustände von Ganzwortmodellen für die Wörter *Nein* und *Neun* zu. Für die LDA bedeutet dies, daß versucht wird, zwei fast identische Klassen trennbar zu machen. Allerdings ist auch in diesem Fall die Wahl der Klassen frei. Prinzipiell können zur Berechnung der LDA–Matrix und bei der Anwendung der LDA–Matrix völlig unterschiedliche Ansätze für die Topologie–Modellierung verwendet werden. So kann z.B. für die LDA–Berechnung ein phonetischer Ansatz mit kontextunabhängiger Phonemmodellierung und für das System mit den transformierten Merkmalen eine Ganzwortmodellierung angesetzt werden. Welche Vorgehensweise zu einem Optimum an Erkennungsleistung führt, kann nur experimentell bestimmt werden.

5.3 Experimente zur LDA

5.3.1 Experimente zur Einbeziehung von Kontext

Für alle Experimente in diesem und dem nächsten Abschnitt wurde die Deutsche Voice–Mail–Datenbank eingesetzt. Speziell werden Training und Erkennung auf einem Teil der Datenbank durchgeführt, in dem nur einzeln gesprochene Kommandowörter und Ziffern enthalten sind. Dabei wird ein Wortschatz von 62 Wörtern zugrundegelegt. Nähere Angaben zur Voice–Mail–Datenbank ([Hauenstein und Marschall, 1995]) und der verwendeten Trainings– und Testaufgabe mit Wortschatzgröße 62 (VM–62) finden sich im Anhang B.1.

Zunächst wird ein Maximum–Likelihood–Training mit den 52-dimensionalen cepstralen Merkmalsvektoren (vgl. Abschnitt 1.2) durchgeführt. Für die Modellierung auf phonetischer Ebene wird zunächst eine kontextabhängige Phonemmodellierung gewählt. Für die Initialisierung der Modelle wird die Methode mit Laufzeit–Clustering verwendet. Die Gesamtzahl der Gauß'schen Dichten wird zu 1838 gewählt. Dies entspricht einer Modellgröße, die auch in einem realen System mit recht begrenzten Ressourcen wie einem Signalprozessor–basierten Erkenner handhabbar wäre. Anschließend werden 5 Iterationen Maximum–Likelihood–Training durchgeführt. Auf dem Testset ergibt sich mit diesen Einstellungen eine Wortfehlerrate von 6.0%.

Zur Berechnung der LDA–Matrix wird für die Segmentierung der Trainingsmenge zunächst eine Ganzwortmodellierung gewählt. Darauf wird im folgendem Abschnitt noch genauer eingegangen. Für die phonetische Modellierung bei Verwendung der LDA–basierten Transformation wird wiederum die kontextabhängige Phonemmodellierung verwendet. Für die LDA–basierte

Transformation bzw. auch für die Berechnung der LDA–Matrix wird die Anzahl der verwendeten Supervektoren zwischen 1 und 3 variiert. Ausgehend von den 3 unterschiedlichen Einstellungen wird jeweils ein auf dem transformierten Merkmalen basierendes Hidden–Markov–Modell trainiert. Die Einstellungen für das ML–Training wurden wie für das Training mit dem cepstralen Merkmalen beschrieben gewählt.

Erkennungstests wurden mit einem auf 52 bzw. 24 Komponenten reduzierten transformierten Merkmalsvektor durchgeführt. Für dieses und viele andere Experimente wird die Dimensionalität des (transformierten) Merkmalsvektors auf den Wert 24 gesetzt. Zum einen ist dies dadurch begründet, daß ab einer Dimensionalität von 20 die Erkennungsleistung fast stagniert (vgl. Ende dieses Abschnitts). Zum anderen hat der Wert 24 den praktischen Vorteil, ein Vielfaches von 8 zu sein. Dies hat für die Umsetzung in Mikroprozessorsystemen und insbesondere bei Verwendung von Signalprozessoren Vorteile für den Speicherzugriff.

Die Fehlerraten, die sich mit den unterschiedlichen Einstellungen ergeben, sind in Tabelle 5.1 zusammengefaßt. Zunächst fällt auf, daß sich schon bei Verwendung von nur einem Super-

Vektoren für den Supervektor	Wortfehlerrate	
	mit 52 Komponenten	mit 24 Komponenten
0 (keine LDA)	6.0	—
1	3.2	3.4
2	2.6	3.0
3	2.6	2.8

Tabelle 5.1: *Vergleich von Fehlerraten auf dem Testamterial für eine unterschiedliche Anzahl von Vektoren zur Bildung des Supervektors für die LDA–basierte Transformation, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

vektor eine deutliche Reduktion der Fehlerrate um ca. 46% (relativ) von 6.0% auf 3.2% einstellt. Verwunderlich ist dies insofern, da ja eine lineare Transformation den Informationsgehalt des Merkmalsvektors nicht erhöhen kann. Der Grund, warum die LDA–basierte Transformation hier trotz identischer Dimensionalität vor und nach der Transformation einen erheblichen Gewinn an Erkennungsleistung bringt, ist in der Dekorreliertheit und Varianznormiertheit (whitening Transformation) der Merkmale zu suchen. Wie im Abschnitt 5.2 schon beschrieben, kommen die dekorrelierten und auf einheitliche Varianz normierten Merkmale der verwendeten einfachen Modellierung stark entgegen. Die Verwendung von diagonalen Kovarianzmatrizen mit einer einheitlichen Varianz ist bei den transformierten Merkmalen eher berechtigt als bei unnormierten

Merkmalen.

Bei 2 Vektoren für den Supervektor sinkt die Fehlerrate sowohl für den 52– als auch den 24–dimensionalen Merkmalsvektor gegenüber der Verwendung von nur einem Vektor als Supervektor. Beim Einsatz von 3 Vektoren zur Bildung des Supervektors reduziert sich die Fehlerrate nur für den Fall des 24–dimensionalen Merkmalsvektors. Die Reduktion der Fehlerrate beim 52–dimensionalen Merkmalsvektor durch Vergrößerung des Supervektors ist 18% bzw. 0% bei Verwendung von drei Vektoren. Bei 24 Komponenten beträgt die Reduktion der Fehlerraten 12% bzw. 7%.

Die durch Verwendung eines größeren Kontexts erzielte Reduktion der Fehlerraten ist also geringer als die Reduktion, die durch die LDA–basierte Transformation mit nur einem Vektor als Supervektor gegenüber dem Fall untransformierter Merkmale erreicht wurde. Dies läßt vermuten, daß die Eigenschaft der Dekorreliertheit und Varianznormiertheit für das vorliegende HMM–System entscheidender sind als der zusätzliche Informationsgehalt durch den erweiterten Kontext.

In Abbildung 5.2 ist die Abhängigkeit der Fehlerrate von der Anzahl der nach der LDA–basierte Transformation verwendeten Merkmalskomponenten aufgetragen. Die in der Grafik eingetragenen Wortfehlerraten ergeben sich bei Erkennung mit nur D Merkmalskomponenten wobei D zwischen 2 und 52 variiert wurde.

Für Werte von $D < 10$ sinkt die Fehlerrate zunächst rapide ab. Zwischen 10 und 20 reduziert sich die Fehlerrate nur noch um ca 3%. Die Gewinne bei $D > 20$ liegen nur noch im Bereich von wenigen Zehntel Prozent. Die Konzentration des diskriminativen Informationsgehalts in der oberen Komponenten des Merkmalsvektors erlaubt also eine deutliche Reduktion des Merkmalsvektors nach der LDA–basierten Transformation ohne einen größeren Verlust an Erkennungsleistung.

Für alle weiteren in dieser Arbeit beschriebenen Experimente wurde die Anzahl der für den Supervektor verwendeten Vektoren zu 2 gewählt.

5.3.2 Veranschaulichung der durch die LDA erzielten Wirkung

Um die charakteristischen Eigenschaften der transformierten Merkmale zu veranschaulichen, werden zunächst zwei Standardabweichungen betrachtet. σ_o ist die (komponentenspezifische) Standardabweichung des gesamten Merkmalsraumes. Für die Berechnung der zweiten betrachteten Standardabweichung werden die Merkmale bzgl. ihrer Klassenzugehörigkeiten betrachtet. Für jede Klasse (Zustand der Hidden–Markov–Modelle) wird zunächst die Standardabweichung des Merkmalsraums der Klasse berechnet. Die mittlere klassenspezifische Standardabweichung σ_s ergibt sich durch Mittelung der Standardabweichungen aller Klassen.

Die einzelnen Komponenten des transformierten Merkmalsvektors wurden nun in der be-

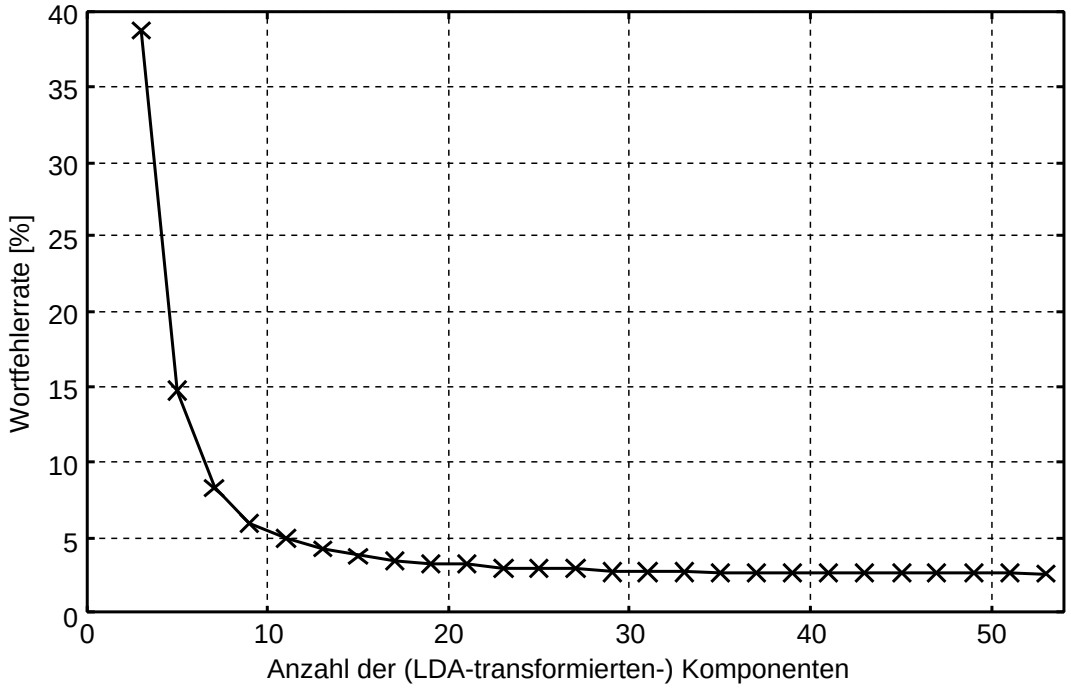

Abbildung 5.2: *Abhängigkeit der Fehlerrate auf dem Testmaterial von der Anzahl der nach der LDA–basierten Transformation verwendeten Merkmalskomponenten, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

schriebenen Weise analysiert und die komponentenspezifischen Standardabweichungen in Abbildung 5.3 visualisiert. Zunächst ist zu erkennen, daß die Standardabweichungen σ_s bis auf durch die Numerik bedingten Abweichungen über dem Komponentenindex konstant sind. Dies entspricht genau den gewünschten Eigenschaften der Merkmale nach der whitening Transformation. Zum anderen ist gut zu erkennen, daß die Standardabweichung σ_o für die oberen Merkmalskomponenten sehr viel größer ist als σ_s, jedoch circa ab Komponente 30 quasi mit σ_s zusammenfällt. In [Höge, 1999] findet sich eine informationstheoretische Interpretation dieser Standardabweichungen, die dort zur näherungsweisen Berechnung einer Entropie verwendet werden.

In Abbildung 5.4 wird versucht, die Verhältnisse im Merkmalsraum grafisch zu veranschaulichen. In der linken Bildhälfte ist eine Projektion in den zweidimensionalen Raum x_1/x_2, in der rechten Bildhälfte der Merkmalsraum x_5/x_6 dargestellt. Mit den großen Kreisen sind die Verteilungen der gesamten Merkmalsräume, mit den kleinen Kreisen die klassenspezifischen Verteilungen dargestellt. Während die klassenspezifischen Verteilungen bzgl. ihrer Ausdehnung konstant sind, ist der gesamte Merkmalsraum in der Ebene x_5/x_6 sehr viel kleiner als in x_1/x_2. Für die oberen Merkmalskomponenten (z.B. x_1/x_2) kann man sich die einzelnen klassenspezifischen Verteilungen als weit im Merkmalsraum verteilt vorstellen. Die Trennbarkeit der Klassen

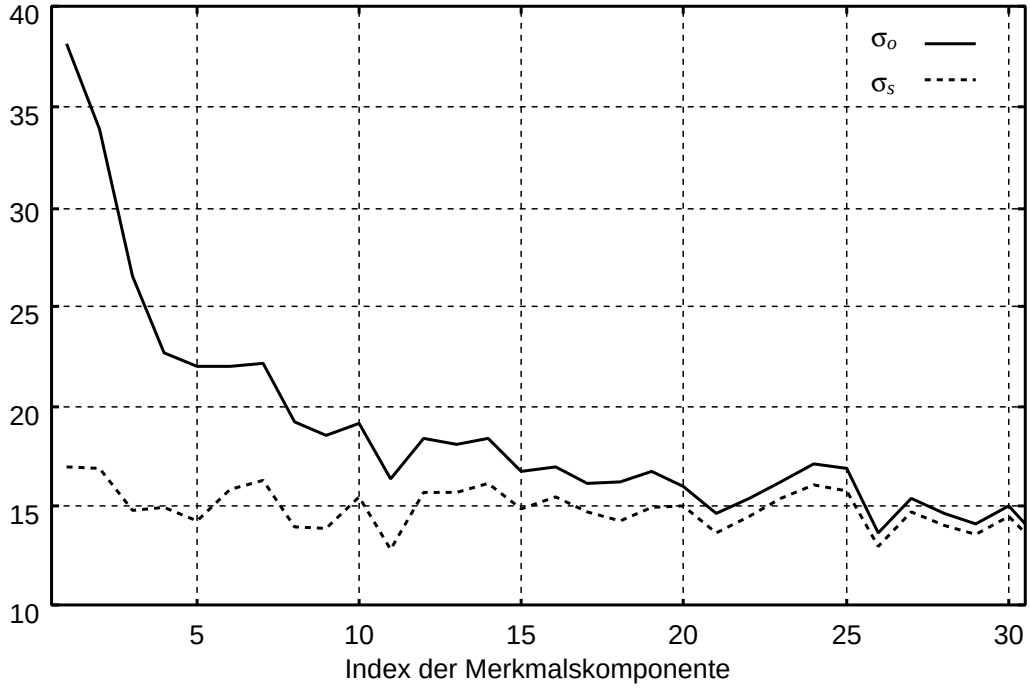

Abbildung 5.3: *Inter– und Intra–Klassen–spezifische Standardabweichungen σ_o und σ_s in Abhängigkeit der Merkmalskomponente nach der LDA–basierten Transformation, Trainingsmaterial von VM–62*

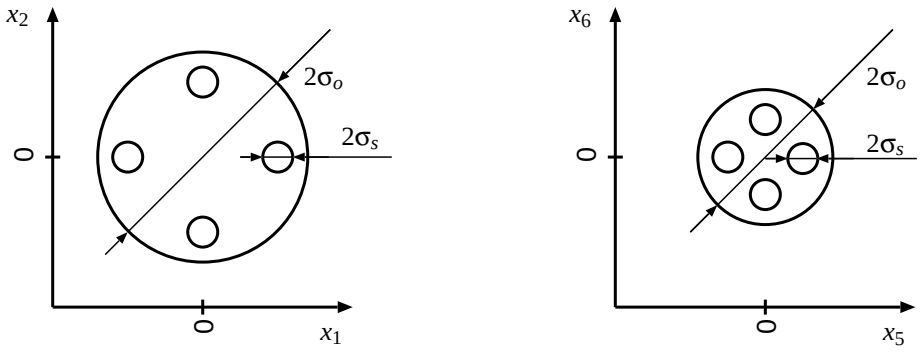

Abbildung 5.4: *Lage der Klassen in der Projektion in zweidimensionale Räume (schematisierte Darstellung)*

ist hier optimal. Für die unteren Komponenten liegen die klassenspezifischen Verteilungen näher zusammen. Für Komponenten ab Index 30 liegen die Verteilungen der Klassen schließlich so nahe beieinander, daß eine Trennbarkeit quasi nicht mehr gegeben ist. Der Anteil an klassendiskriminierender Information in den untersten Komponenten ist fast Null.

5.3.3 Experimente zur Wahl der Klassen

In [Haeb-Umbach und Ney, 1992] findet sich eine Untersuchung zur Wahl der Klassen für die LDA. Dort wurden Phoneme, Zustände und Moden der Zustandsverteilungen verglichen. Das Ergebnis der Untersuchungen war, daß sich für die Verwendung von Zuständen die besten Erkennungsergebnisse erzielen lassen. Für die in diesem Abschnitt beschriebenen Untersuchungen wird nun zwar die Wahl der Klassen auf Zustände fixiert, unterschiedliche Verhältnisse entstehen aber durch eine unterschiedliche Ausprägung der phonetischen Modellierung.

Mit der Wahl der phonetischen Modellierung wird die Anzahl und Art der Klassen für die Berechnung der LDA festgelegt. Es werden drei verschiedene Modellierungen betrachtet: kontextunabhängige Phonemmodellierung, kontextabhängige Phonemmodellierung und Ganzwortmodellierung. Alle Angaben beziehen sich hier auf den Wortschatz mit 62 Wörtern aus der Voice–Mail–Datenbank. Für die kontextunabhängige Phonemmodellierung ist die Zahl der Klassen mit 115 am kleinsten. Für kontextabhängige Phonemmodellierung liegt die Zahl der Klassen bei 447 und für Ganzwortmodellierung schließlich bei 1153.

In der vorliegenden Versuchsreihe wurden drei verschiedene LDA–Matrizen entsprechend der drei Möglichkeiten für die Phonemmodellierung berechnet. In Tabelle 5.2 sind auch die Fehlerraten für die Modellierung ohne LDA–basierte Transformation eingetragen. Dort wurden unabhängig von der phonetischen Modellierung 1838 Gauß'sche Dichten verwendet. Darin ist der Grund für die hohe Wortfehlerrate bei Ganzwortmodellierung zu suchen. An dieser Stelle war die mittlere Anzahl von Dichten pro Zustand wohl zu klein gewählt. Ansonsten ergibt sich ohne

vor LDA		nach LDA	
Kontext	Fehlerrate	Fehlerrate NKTX	Fehlerrate KTX
NKTX	7.2	3.8	2.9
KTX	6.0	3.6	2.7
GW	11.9	3.4	2.6

Tabelle 5.2: *Vergleich von Fehlerraten auf dem Testmaterial für verschiedene Arten der Kontextmodellierung vor bzw. nach Anwendung der LDA: Kontextunabhängige Modellierung (NKTX), Kontextabhängige Modellierung (KTX), und Ganzwortmodellierung (GW), Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

LDA–basierte Transformation für die kontextabhängige Phonemmodellierung ein etwas besseres Ergebnis als für die kontextunabhängige Phonemmodellierung.

Für den transformierten Merkmalsvektor wurde die Dimension für diesen Versuch ebenfalls

auf den Wert 52 gesetzt. Für die Modellierung mit der LDA–basierten Transformation wurden die Varianten kontextabhängige und kontextunabhängige Phonemmodellierung untersucht. Auffällig ist, daß der Gewinn durch die kontextabhängige Phonemmodellierung mit LDA–basierter Transformation sehr viel höher ist als ohne die Transformation. Das entscheidende Ergebnis ergibt sich durch die Variation der phonetischen Modellierung zur Berechnung der LDA–Matrix. Die Fehlerrate reduziert sich eindeutig bei steigender Zahl der Klassen zur Berechnung der LDA–Matrix.

Unter gewissen Gesichtspunkten wäre ein anderes Ergebnis zu erwarten gewesen. Vergleicht man z.B. die kontextunabhängige Phonemmodellierung mit der Ganzwortmodellierung, so erscheint eine Maximierung der Klassifikationsleistung für die Phonemmodellierung sinnvoller. Bei der Ganzwortmodellierung gibt es mehrere Klassen, die fast identischen artikulatorischen Ereignissen entsprechen. So werden bei der Ganzwortmodellierung z.B. die Anfangsteile von *zwei* und *zwo* mit unterschiedlichen Zuständen modelliert, obwohl die beiden Wortanfänge akustisch sehr ähnlich sind. Aus dieser Überlegung heraus erscheint es sinnvoller, die Klassentrennbarkeit bei einer phonetischen Modellierung mit wenigen und akustisch möglichst verschiedenen Einheiten zu optimieren als bei sehr vielen Einheiten, wie bei der Ganzwortmodellierung. Allen theoretischen Überlegungen entgegen ist das experimentelle Ergebnis jedoch eindeutig.

Kapitel 6

Diskriminative Nachschätzung von HMM–Parametern

6.1 Einführung und Motivation

Das Verfahren der Maximum–Likelihood (ML) Schätzung ist zunächst unabhängig von der Klassifikationsleistung definiert. Ziel der Optimierung ist nur die Maximierung der Erzeugungswahrscheinlichkeit für die Trainingsmuster. Nur unter sehr eingeschränkten Voraussetzungen kann gezeigt werden, daß ein System, dessen Parameter nach diesem Verfahren geschätzt wurden, die optimale Klassifikationsleistung erreicht ([Brown, 1987]). Die Voraussetzungen hierfür sind eine stochastische Modellierung, die den realen Prozeß vollständig beschreibt und eine ausreichend große Trainingsstichprobe, die eine hinreichend genaue Schätzung der Parameter erlaubt. Bei der Anwendung des ML Verfahrens für Hidden–Markov–Modelle in der Spracherkennung ist meist keine dieser Voraussetzungen erfüllt. Zum einen weisen Hidden–Markov–Modelle klare Defizite wie die mangelnde Berücksichtigung der zeitlichen Korrelation der Merkmale auf ([Wellekens, 1987]). Zum anderen ist das in der Praxis vorhandene Sprachmaterial, das zum Training zur Verfügung steht, immer endlich und oft nicht ausreichend.

Eine Charakteristik des Maximum–Likelihood–Schätzverfahrens läßt sich an seiner Definition (vgl. Gleichung (4.1)) ablesen: in den einzelnen Beiträgen der Zielfunktion sind nur Parameter des zur jeweiligen Stichprobe korrespondierenden Modells enthalten. Wechselwirkungen zwischen den Parametern der unterschiedlichen Modelle werden also nicht explizit berücksichtigt. Sucht man nun nach einer Gemeinsamkeit der bekannten, als *diskriminativ* bezeichneten Verfahren wie MMI und MCE, so finden sich Beiträge in der Zielfunktion, welche die genannten Wechselwirkungen repräsentieren. Während bei der ML Schätzung die Erzeugungswahrscheinlichkeit der Trainingsmuster maximiert wird, so beinhalten diskriminative Verfahren üblicherweise eine Minimierung der Erzeugungswahrscheinlichkeit durch Modelle, die nicht mit den

Trainingsmustern korrespondieren.

In zahlreichen Publikationen wurden verschiedene diskriminative Verfahren wie *Minimum Classification Error* (MCE), Maximum–Mutual–Information (MMI) oder *corrective training* verglichen und darüber hinaus verschiedene Kombinationen wie *corrective MMI* beschrieben. Sowohl die Theorie als auch verschiedenste experimentelle Ergebnisse zeigen, daß die Unterschiede zwischen den Verfahren, insbesondere zwischen MMI und MCE, nicht essentiell sind ([Schlüter, 2000]). Insgesamt ist erkennbar, daß in der Formulierung von MCE mehr nutzbare Freiheitsgrade enthalten sind und die experimentell erzielten Ergebnisse teilweise geringfügig besser sind als bei MMI ([Reichl, 1996], [Schlüter und Macherey, 1998]). In allen nachfolgenden Untersuchungen wird die Zielfunktion für die diskriminative Schätzung der Modellparameter ausgehend von dem Verfahren *Minimum Classification Error* (MCE) gewählt, das im folgenden Abschnitt detaillierter dargestellt wird.

6.2 Grundprinzip von MCE

Übersetzt man den Begriff *Minimum Classification Error* aus dem Englischen wörtlich ins Deutsche, so erhält man die Bezeichnung *Minimaler Klassifikationsfehler*. Die Idee hinter dem MCE–Verfahren ist es also, den Klassifikator mit der geringstmöglichen Klassifikationsfehlerrate zu finden. Überträgt man dies auf das Gebiet der Parameterschätzung für die automatischen Spracherkennung, so ergibt sich als Ziel die Minimierung der Fehlerrate, die ein Klassifikator auf einer gegebenen Menge von Trainingsstichproben liefert. In der Praxis wird dazu eine Zielfunktion l_{MCE} in Abhängigkeit der Modellparameter Λ und den Trainingsdaten $\{S_r\}$ definiert, welche die Klassifikationsfehlerrate für die Trainingsdaten approximiert:

$$\Lambda_{\mathrm{MCE}} = \underset{\Lambda}{\operatorname{argmin}}\, l_{\mathrm{MCE}}(\Lambda, \{S_r\}) \tag{6.1}$$

Um die Optimierung der Parameter datengetrieben durchführen zu können, muß die Zielfunktion differenzierbar sein. Das macht die Verwendung einer Approximation der Fehlerrate notwendig. Die exakte Fehlerrate als Mittel von diskreten binären Werten 0 für korrekte Klassifikation in 1 für Fehlklassifikation ist eine nicht–stetige und nicht–differenzierbare Funktion.

Die Zielfunktion l_{MCE} wird zunächst als Erwartungswert über Summanden für die einzelnen Muster $\{S_r\}$ gebildet:

$$l_{\mathrm{MCE}}(\Lambda, \{S_r\}) = \frac{1}{R} \sum_{r=0}^{R} l(S_r, \Lambda) \tag{6.2}$$

R bezeichnet heribei die Anzahl der Trainingsmuster. Die einzelnen Summanden $l(S_r, \Lambda)$ stellen eine stetige Approximation des Klassifikationsfehlers für die einzelnen Muster S_r dar. Hierbei wird eine Diskriminanzfunktion $d(S, \Lambda)$ in eine Sigmoidfunktion eingebettet, sodaß sich (bei

einem unbeschränkten Wertebereich für d) für l Werte zwischen 0 und 1 ergeben:

$$l(S,\Lambda) = \frac{1}{1+e^{-\gamma d(S,\Lambda)}} \tag{6.3}$$

Im Prinzip wären auch andere Funktionen für die Einbettung der Diskriminanzfunktion denk-

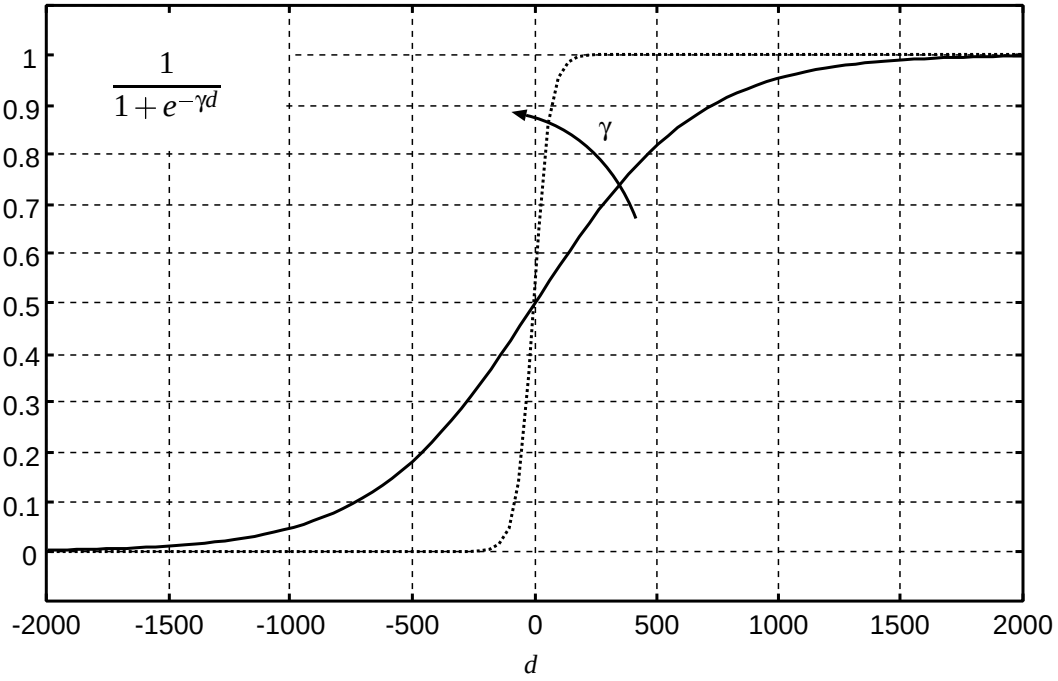

Abbildung 6.1: *Die Sigmoidfunktion $l(d)$ für zwei verschiedene Werte von γ*

bar. Die Diskriminanzfunktion d beschreibt die Fähigkeit des Klassifikators, ein Muster richtig klassifizieren zu können, wobei positive Werte von d für eine schlechte und negative Werte von d für eine gute Fähigkeit zur richtigen Klassifikation des Musters stehen.

Sei die korrekte Klasse für das Muster S durch $i = \Omega(S)$ gegeben, so lautet die Definition für die Diskriminanzfunktion:

$$d(S,\Lambda) = g(S,\lambda_i) + \log\left(\frac{1}{J-1}\sum_{j\neq i} e^{-g(S,\lambda_j)\eta}\right)^{\frac{1}{\eta}} \tag{6.4}$$

Mit der logarithmierten Modellwahrscheinlichkeit (Score):

$$g(S,\lambda) = -\sigma_0^2 \cdot \log P(S|\lambda) \tag{6.5}$$

In (6.4) wird die Differenz zwischen der logarithmierten Wahrscheinlichkeit des korrekten Modells λ_i und dem Logarithmus eines gewichteten Mittels der Wahrscheinlichkeiten aller anderen

Modelle λ_j gebildet. J ist hierbei die Zahl der konkurrierenden Modelle. Ein großer Exponent η bewirkt eine starke Gewichtung des Maximums $P(S|\lambda_h) = \max_j P(S|\lambda_j)$ mit $j \neq i$. Für den Fall $\eta \to \infty$ ([Euler und Zinke, 1992]) reduziert sich der zweite Summand in (6.4) auf den Score des bestpassenden konkurrierenden Modells λ_h mit $h = \text{argmax}_{j \neq i} P(S|\lambda_j)$:

$$d(S,\Lambda)|_{\eta\to\infty} = g(S,\lambda_i) - g(S,\lambda_h) \tag{6.6}$$

Die Definition der Diskriminanzfunktion nach (6.6) wird in der Literatur häufig verwendet (z.B. [Euler und Zinke, 1992], [Rainton und Sagayama, 1992]). Zum einen besteht zwar der Nachteil, daß nur maximal zwei Modelle einen Beitrag zur Zielfunktion für ein Muster liefern, zum anderen hat aber durch diese Festlegung die Diskriminanzfunktion d den maximalen Bezug zur Klassifikationsleistung. Es ist sichergestellt, daß das Vorzeichen der Diskriminanzfunktion eineindeutig mit dem Vorhandensein eines Klassifikationsfehlers verknüpft ist. Für alle experimentellen Untersuchungen wird diese Definition (6.6) der Diskriminanzfunktion d mit $\eta \to \infty$ gewählt. Würde zusätzlich in (6.3) der Parameter $\gamma \to \infty$ gewählt, so würde die Zielfunktion exakt die Fehlerrate des Klassifikators angeben. Dies ist jedoch nicht sinnvoll, da die Zielfunktion nur für Werte von $\gamma < \infty$ differenzierbar ist.

6.3 Gradientenbasiertes Optimierungsverfahren für MCE

6.3.1 Gradientenverfahren

Das MCE–Verfahren wird sehr oft in Zusammenhang mit dem Verfahren des *General Probabilistic Descend* (GPD) genannt. Obwohl der Begriff GPD häufig sogar als Synonym für MCE in Kombination mit GPD verwendet wird, ist es natürlich nicht das einzig mögliche einsetzbare Optimierungsverfahren. In der Literatur wird neben erweiterten Gradientenverfahren z.B. von höherer Ordnung häufig der erweiterte Baum–Welch–Algorithmus ([Gopalakrishnan u. a., 1989], [Gopalakrishnan u. a., 1991]) eingesetzt. Die Vorteile dieser anderen Optimierungsverfahren liegen in einer potentiell besseren Konvergenz und für das erweiterte Baum–Welch–Verfahren in der leichteren Einhaltung von Nebenbedingungen, wie sie für die Mixturkoeffizienten gegeben sind. Im Rahmen dieser Arbeit wird jedoch auf das GPD Verfahren aufgebaut, welches im folgenden beschrieben wird.

Das GPD Verfahren ist ein iteratives Schätzverfahren mit einer über dem Iterationsindex k variablen Schrittweite ε_k. Die nachgeschätzten Parameter Λ_{k+1} nach der Iteration k ergeben sich wie folgt aus den Parametern Λ_k vor der Iteration:

$$\Lambda_{k+1} = \Lambda_k - \varepsilon_k \nabla l(\{S_r\}, \Lambda_k) \tag{6.7}$$

Es kann gezeigt werden, daß das Verfahren für k gegen ∞ theoretisch zu einem lokalen Minimum der Zielfunktion konvergiert, wenn die Folge ε_k folgenden Bedingungen genügt: $\sum_{k=0}^{\infty} \varepsilon_k = \infty$,

$\sum_{k=0}^{\infty} \varepsilon_k^2 < \infty$ und $\varepsilon_k \geq 0$ ([Juang und Katagiri, 1992]). In der Praxis wird meist eine linear sinkende Folge $\varepsilon_k = \varepsilon_0(1 - \frac{k}{K})$ bei K Iterationen verwendet.

Für die in dieser Arbeit dargelegten Experimente wird jedoch statt einer Folge sinkender Werte von ε_k ein fester Wert ε_0 verwendet. Zum einen hat sich die potentiell schnellere Konvergenz durch höhere Werte für die Schrittweite in den ersten Iterationen nicht bewährt ([Bauer, 1997]). Zum anderen vermeidet man durch eine konstante Schrittweite eine vorherige Festlegung der Anzahl von Iterationsschritten K, die für eine linear sinkende Folge zumindest theoretisch notwendig ist.

6.3.2 Nachschätzformeln

o sei ein Parameter aus der Menge der Parameter Λ. Die grundsätzliche Vorschrift zur Berechnung des nachgeschätzten Parameters $\hat{o}$ lautet nun:

$$\hat{o} = o - \varepsilon \cdot \frac{\partial l(S,\Lambda)}{\partial o} \tag{6.8}$$

Die partielle Ableitung in (6.8) läßt sich nun weiter aufspalten:

$$\hat{o} = o - \varepsilon \cdot \frac{\partial l(d(S,\Lambda))}{\partial d(S,\Lambda)} \cdot \frac{\partial d(S,\Lambda)}{\partial o} \tag{6.9}$$

$$= o - \varepsilon \cdot \frac{\partial l(d(S,\Lambda))}{\partial d(S,\Lambda)} \cdot \left(\frac{\partial g(S,\lambda_i)}{\partial o} - \frac{\partial g(S,\lambda_h)}{\partial o} \right) \tag{6.10}$$

Dabei sind λ_i und λ_h die Modelle entsprechend der korrekten Klasse $i = \Omega(S)$ und der besten konkurrierenden Klasse h. Dieser erste Ausdruck in (6.10) stellt eine Art Gewichtungsfunktion dar, die unabhängig von der Art des Parameters ist. Im zweiten Ausdruck finden sich die beiden partiellen Ableitungen entsprechend den beiden Modellen λ_i und λ_h.

Die partielle Ableitung der Zielfunktion l läßt sich dabei folgendermaßen berechnen:

$$\frac{\partial l(d(S,\Lambda))}{\partial d(S,\Lambda)} = \gamma \cdot l(S,\Lambda) \cdot (1 - l(S,\Lambda)) \tag{6.11}$$

In Abbildung 6.2 ist die Sigmoidfunktion $l(d)$ und ihre Ableitung nach d grafisch dargestellt. Interpretiert man diese Ableitung als Gewichtungsfunktion (vgl. (6.9)), so wird deutlich, wie stark der Einfluß einer Äußerung in Abhängigkeit vom Diskriminanzmaß d ist: den größten Einfluß besitzen Äußerungen mit $d \approx 0$, also Äußerungen, bei denen die Klassifikation unsicher ist. Andererseits sinkt der Einfluß für steigende Werte von $|d|$ wieder. Äußerungen, die also eindeutig richtig oder eindeutig falsch klassifiziert worden sind, tragen zur Nachschätzung der Parameter kaum bei. Dies erscheint sinnvoll, da zum einen Muster, die keine Problemfälle bei der Klassifikation darstellen, nicht verstärkt gelernt werden müssen. Zum anderen ist die Annahme

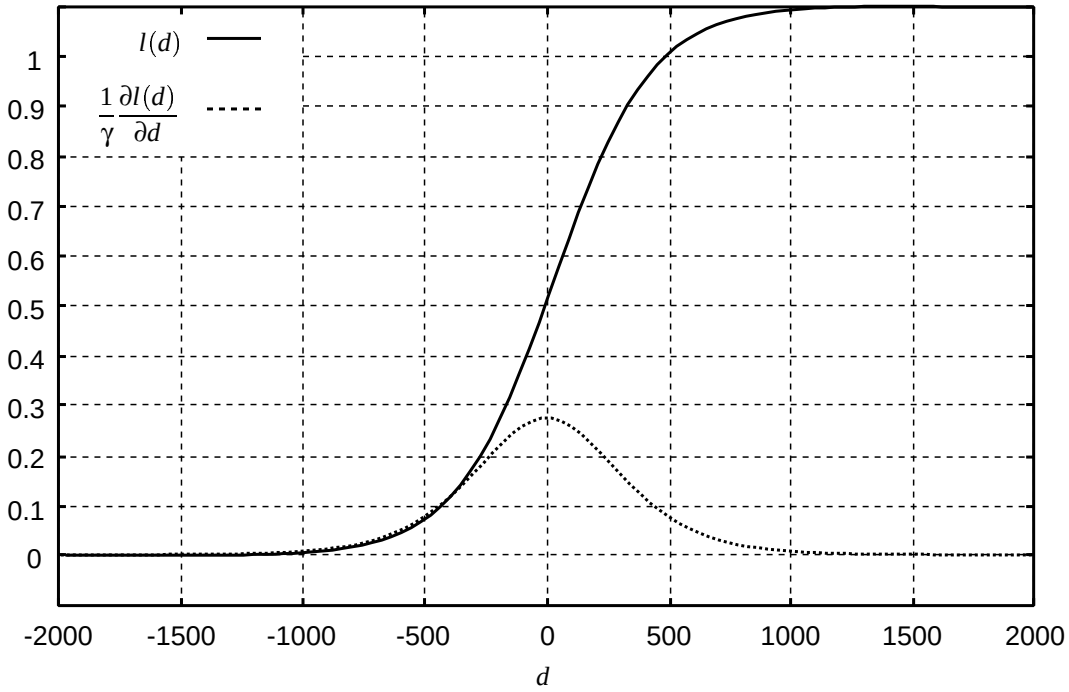

Abbildung 6.2: *Die Sigmoidfunktion $l(d)$ und ihre partielle Ableitung (skaliert mit $1/\gamma$)*

berechtigt, daß Muster, die grob falsch klassifiziert wurden, nicht sinnvoll zur Verbesserung des Klassifikators genutzt werden können. Oft handelt es sich dabei um Ausreißer, möglicherweise verursacht durch Fehler in der Transkription der Datenbank.

6.4 Auswahl der Parameter zur Nachschätzung

Wie bei der Maximum–Likelihood–Schätzung kommen bei dem zugrundeliegenden Spracherkennungssystem prinzipiell zwei Parametersätze zur Neu– bzw. Nachschätzung in Betracht. Das sind die Mittelpunktsvektoren der Gauß'schen Wahrscheinlichkeitsdichtefunktionen sowie die Mixturkoeffizienten, welche die Gaußdichten in einem Zustand gewichten. Eine Schätzung der globalen Varianz und der globalen Übergangsstrafen wäre prinzipiell auch möglich. Der Einfluß der genannten globalen Parameter auf die Erkennungsleistung des Gesamtsystems ist jedoch relativ gering, und eine Optimierung im Sinne einer maximalen Klassifikationsleistung ist durch einfache empirische Optimierung einfacher realisierbar; d.h. aufgrund der wenigen Freiheitsgrade kann die Optimierung durch *Durchfahren* eines breiten Wertebereichs der Parameter erfolgen.

6.4.1 Mixturkoeffizienten

In einem Hidden–Markov–Modell gibt es genau so viele Mixturkoeffizienten wie es Moden gibt: $\sum_s M_s$. Die Anzahl dieser Werte ist also relativ gering im Vergleich zur Anzahl der Parameter für die Mittelpunktsvektoren. Anders würde es sich bei einem System mit semikontinuierlichen Modellen verhalten. Dort besitzt jeder Zustand soviele Mixturkoeffizienten wie es codebook–Vektoren gibt. In der Literatur ([Reichl, 1996]) findet sich auch deshalb die Vorgehensweise, bei semikontinuierlichen Modellen nur die Mixturkoeffizienten nachzuschätzen und die Mittelpunktsvektoren unverändert zu lassen. Bei kontinuierlichen Hidden–Markov–Modellen ist nun zum einen der Einfluß dieser wenigen Parameter relativ gering, zum anderen kann die geringere Parameteranzahl zu einer besseren Generalisierung führen. Grundsätzlich erscheint die Nachschätzung der Mixturkoeffizienten in dem vorliegenden System als sinnvoll.

Geht man von einem HMM als stochastischem Modell aus, muß für die Mixturkoeffizienten eines Zustands die Normierung auf Summe gleich 1 gelten (vgl. Gleichung (2.11)). In der Literatur findet sich deshalb häufig die Vorgehensweise, daß dies durch eine geeignete Nebenbedingung bei der Optimierung erhalten bleibt ([Wolfertstetter, 1997]). Mit der Anwendung des MCE–Verfahrens kann man aber von einem Verlassen der stochastischen Modellierung im eigentlichen Sinne sprechen. Während bei der Maximum–Likelihood–Schätzung die Systemparameter Werte annehmen sollen, die möglichst den *wahren* Werten entsprechen, kann man bei MCE diese Idee eigentlich über Bord werfen. Das einzige Optimierungskriterium bei MCE ist die approximierte Erkennungsrate. Dieser Argumentation folgend wird in den dargestellten Arbeiten eine Normierung der Mixturkoeffizienten außer acht gelassen. Damit ist es dann auch möglich, die Mixturkoeffizienten für single densities Modelle ($M_s = 1$) nachzuschätzen, was die Normierung auf Summe 1 natürlich verbieten würde.

Statt der eigentlichen Mixturkoeffizienten sollen nun die neg–log–transformierten Mixturkoeffizienten, also die Mixturstrafen $C_{s,m}$ nach Gleichung (3.15), Gegenstand der Optimierung sein. Die Optimierung ohne Nebenbedingungen bietet auch theoretisch einen Vorteil gegenüber der Optimierung mit Nebenbedingung: da keine Normierung im Zustand mehr verlangt ist, können die Mixturstrafen quasi über Zustandsgrenzen hinweg im Sinne einer Gewichtung bestimmter Zustände in Wechselwirkung treten. Das Nachschätzen der Mixturstrafen ohne Nebenbedingung kann potentiell eine optimierte Gewichtung der Moden in einem Zustand und der Zustände untereinander wie in [Wolfertstetter und Ruske, 1994] und [Wolfertstetter, 1997] bewirken.

Nachschätzformel Ausgehend von der allgemeinen Nachschätzformel (6.10) läßt sich für die Mixturstrafen folgende Nachschätzformel formulieren:

$$\hat{C}_{s,m} = C_{s,m} - \varepsilon \cdot \frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)} \cdot \sum_{t=1}^{T} \zeta_t^i(s,m) + \varepsilon \cdot \frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)} \cdot \sum_{t=1}^{T} \zeta_t^h(s,m) \qquad (6.12)$$

Dabei sind $\zeta_t^i(s,m)$ und $\zeta_t^h(s,m)$ die Zuordnungsfunktionen entsprechend der Definition (4.7) für das korrekte Modell λ_i und das beste konkurrierende Modell λ_h. Man sieht, daß die Mixturstrafen erniedrigt werden, wenn eine Mode bei der Decodierung des korrekten Modells angesprochen wurde und erhöht werden, wenn eine Mode bei der Decodierung eines konkurrierenden Modells angesprochen wurde. Damit reduziert sich die Erzeugungswahrscheinlichkeit des Modells λ_h und erhöht sich die Erzeugungswahrscheinlichkeit des Modells λ_i für die Merkmalsfolge X.

6.4.2 Verteilungsschwerpunkte

Die Mittelpunktsvektoren der Gaußverteilungen $\vec{\mu}_{s,m}$ stellen in dem vorliegenden HMM–Erkennungssystem die mit Abstand wichtigsten Einflußgrößen dar. Bei kontinuierlichen Hidden–Markov–Modellen ist allein die Anzahl der freien Parameter dieser Gruppe im Normalfall um mehrere Größenordnungen höher als z.B. die der Mixturstrafen. Ein Nachschätzen dieser Parameter erscheint zumindest für kontinuierliche HMM sehr sinnvoll.

Nachschätzformel Setzt man die Berechnungsvorschriften für die neg–log–transformierten Modellwahrscheinlichkeiten (3.16) und (3.13) in die allgemeine Nachschätzformel (6.10) ein, so erhält man folgende Vorschrift für die Berechnung der nachgeschätzten Mittelpunktsvektoren:

$$\begin{aligned}\hat{\vec{\mu}}_{s,m} = \vec{\mu}_{s,m} \quad &+ \quad 2\varepsilon \cdot \frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)} \cdot \sum_{t=1}^{T} (\vec{x}_t - \vec{\mu}_{s,m}) \cdot \zeta_t^i(s,m) \\ &- \quad 2\varepsilon \cdot \frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)} \cdot \sum_{t=1}^{T} (\vec{x}_t - \vec{\mu}_{s,m}) \cdot \zeta_t^h(s,m) \end{aligned} \tag{6.13}$$

In (6.13) sieht man sehr schön, wie die Mittelpunktsvektoren entweder in Richtung des Merkmalsvektors oder in Gegenrichtung verschoben werden. In Abbildung 6.3 ist dieser Sachverhalt nochmals im zweidimensionalen Fall grafisch veranschaulicht. Dort wäre $\vec{\mu}_1$ der Mittelpunktsvektor einer Mode die bei der Decodierung des korrekten Modells angesprochen wird. Folglich wird die Neuschätzung $\hat{\vec{\mu}}_1$ des Mittelpunktsvektors in Richtung des Merkmalsvektors $\vec{x}$ verschoben. Der Mittelpunktsvektor $\vec{\mu}_2$ und die Neuschätzung $\hat{\vec{\mu}}_2$ hingegen gehören zu einer Mode, die bei der Decodierung des besten konkurrierenden Modells angesprochen wurde. Die Verschiebung findet dort genau vom Merkmalsvektor weg statt.

6.5 Experimente zur Auswahl der Parameter

Ausgangspunkt der in diesem Abschnitt beschriebenen Versuche ist das Ergebnis der in Abschnitt 5.3.1 und 5.3.3 beschriebenen Experimente. Dort wurde durch Maximum–Likelihood–Training ein HMM erzeugt, daß mit 24 (LDA–transformierten) Merkmalskomponenten eine

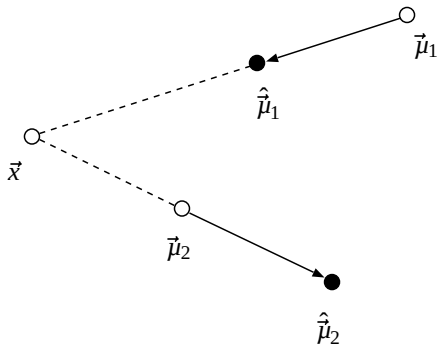

Abbildung 6.3: *Verschiebung von Mittelpunktsvektoren zum Merkmalsvektor $\vec{x}$ hin ($\vec{\mu}_1$) oder vom Merkmalsvektor weg ($\vec{\mu}_2$)*

Fehlerrate von 3.0% auf dem VM–62 Testset produziert. Das entsprechende HMM besitzt eine Gesamtzahl von 1838 Gauß'schen Verteilungsfunktionen und somit auch 1838 Mixturstrafen und 1838 Mittelpunktsvektoren mit je 24 Komponenten.

Wie bei allen Experimenten mit diskriminativer Nachschätzung der HMM–Parameter diente ein Maximum–Likelihood–Modell als Ausgangswert für die Nachschätzung. Mit dem Optimierungskriterium Minimaler Wortfehler, das erst im Abschnitt 6.12 genauer beschrieben wird, wurden je 15 Iterationen diskriminativem Trainings durchgeführt. In einem ersten Schritt wurden lediglich die Mixturstrafen nachgeschätzt, während die Mittelpunktsvektoren nicht verändert wurden. In einem zweiten Versuch wurden andererseits die Mixturstrafen beibehalten und nur die Mittelpunktsvektoren diskriminativ nachtrainiert. Im dritten und letzten Versuch dieser Reihe wurden sowohl Mixturstrafen als auch die Mittelpunktsvektoren simultan nachgeschätzt. In Tabelle 6.1 sind die Ergebnisse dieser drei Versuche zusammengefaßt.

nachgeschätzte Parameter	Wortfehlerrate [%]
keine / ML–Ausgangsmodell	3.0
nur Mixturstrafen	2.1
nur Mittelpunkte	1.7
Mittelpunkte und Mixturstrafen	1.6

Tabelle 6.1: *Vergleich von Fehlerraten auf dem Testmaterial bei diskriminativer Nachschätzung unterschiedlicher Parameter, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

Allein durch die Nachschätzung der Mixturstrafen wird eine Reduktion der Wortfehlerra-

te um 30% auf 2.1% absolut erzielt. Bei der alleinigen Nachschätzung der Mittelpunktsvektoren reduziert sich die Fehlerrate im Vergleich zum Maximum–Likelihood–Fall um ca. 43% auf 1.7% absolut. Werden nun Mixturstrafen und Mittelpunktsvektoren gleichzeitig diskriminativ trainiert, ergibt sich eine Fehlerrate von 1.6%. Als Ergebnis bleibt festzuhalten, daß die Nachschätzung der Mittelpunktsvektoren das leistungsfähigere Mittel zur Reduktion der Fehlerrate ist, obwohl in diesem Fall auch die Reduktion der Fehlerrate allein durch Nachtraining der Mixturstrafen signifikant ist. Dahingegen erweist sich die Reduktion der Fehlerrate durch simultanes Nachtraining von Mixturstrafen und Mittelpunktsvektoren gegenüber der alleinigen Nachschätzung der Mittelpunktsvektoren mit einer Absenkung von nur 0.1% absolut als nicht signifikant.

Mittels zweier weiterer Versuche soll nun untersucht werden, welches Potential die Methoden bieten, wenn nur eine minimale Anzahl an nachzutrainierender Parameter vorhanden ist. Hierzu wird ein Maximum–Likelihood trainiertes HMM mit single densities verwendet. Die Gesamtzahl der Dichten ist damit gleich der Anzahl der verschiedenen Segmente: 447.

In Tabelle 6.2 sind die Ergebnisse für die Nachschätzung der Mixturstrafen in Abhängigkeit der Modellgröße dargestellt. Während bei dem größeren HMM eine Reduktion der Wortfehler-

Anzahl Dichten	Wortfehlerrate [%]	
	ML–Ausgangsmodell	MWF–Modell
447	6.8	6.0
1838	3.0	2.1

Tabelle 6.2: *Reduktion der Fehlerrate auf dem Testmaterial durch diskriminatives Training (Minimaler Wortfehler: MWF) der Mixturstrafen von Modellen mit unterschiedlicher Anzahl von Dichten, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

rate um 30% erzielt wurde, konnte ein Nachtraining der Mixturstrafen bei single densities die Fehlerrate lediglich um ca. 12% senken.

In Tabelle 6.3 finden sich die Ergebnisse bei Nachschätzung der Mittelpunktsvektoren. Hierbei zeigt sich, daß ein Nachtraining der Mittelpunktsvektoren bei kleinerer Modellgröße sogar eine stärkere Reduktion der Fehlerrate (ca. 48%) liefert als beim größeren HMM (ca. 43%).

Zusammenfassend kann gesagt werden, daß Mittelpunktsvektoren im Vergleich zu den Mixturstrafen das höhere Potential zur Verbesserung der Erkennungsleistung bei diskriminativem Training besitzen. Insbesondere bei kleiner Modellgröße, welche im Kontext dieser Arbeit besonders interessant ist, kann durch diskriminative Nachschätzung der Mittelpunktsvektoren eine signifikante Verbesserung erzielt werden. Die Verbesserung, die sich durch simultanes Nachtraining von Mittelpunktsvektoren und Mixturstrafen ergibt, hat sich als nicht signifikant gegenüber

Anzahl Dichten	Wortfehlerrate [%]	
	ML–Ausgangsmodell	MWF–Modell
447	6.8	3.5
1838	3.0	1.7

Tabelle 6.3: *Reduktion der Fehlerrate auf dem Testmaterial durch diskriminatives Training (Minimaler Wortfehler: MWF) der Verteilungsmittelpunkte von Modellen mit unterschiedlicher Anzahl von Dichten, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

dem alleinigen Nachtraining der Mittelpunktsvektoren erwiesen. Für alle folgenden Experimente wird sich deshalb ein Nachtraining der Modelle auf eine Neuschätzung der Mittelpunktsvektoren beschränken.

6.6 Anpassung der Fehlerfunktion

In diesem Abschnitt soll die Frage im Vordergrund stehen, wie der Parameter γ der Sigmoidfunktion für ein bestimmtes Training optimal zu wählen ist.

Eine mögliche Vorgabe für die Wahl von γ könnte sein, daß die Fehlerrate so gut wie möglich approximiert wird. Dies spräche für einen großen Wert, so daß die Sigmoidfunktion fast zur Sprungfunktion wird. Für $\gamma = \infty$ (und $\eta = \infty$) wäre zwar die Zielfunktion gleich der wirklichen Fehlerrate (vgl. Abschnitt 6.2), aber die Zielfunktion wäre nicht mehr differenzierbar.

Für die Bestimmung eines geeigneten Werts für γ müssen wohl andere Argumente mit einbezogen werden. Wichtig ist in diesem Zusammenhang die Interpretation der partiellen Ableitung der Zielfunktion nach d als Gewichtungsfunktion in Abhängigkeit der Diskriminanzfunktion d (vgl. Abschnitt 6.3.2). Ausgehend von dieser Gewichtungsfunktion und den Einfluß des Parameters γ auf diese Funktion soll nun ein praktisch anwendbares Schema entwickelt werden, das die Bestimmung eines geeigneten Werts γ erlaubt:

1. Durchführung einer MCE–Iteration mit einem beliebigen Wert für γ mit Bestimmung eines Histogramms für die Diskriminanzfunktion d (Ergebnis der Iteration wird verworfen).
2. Bestimmung des Werts γ in der Art, daß die Gewichtungsfunktion $\frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)}$ gegen 0 konvergiert, sobald auch die Häufigkeit für positive Werte von d gegen 0 konvergiert.
3. Durchführung aller MCE–Iterationen mit dem bestimmten Wert γ.

Die Vorgehensweise wird im folgenden Abschnitt anhand des Beispiels in Abbildung 6.4 näher erläutert.

6.7 Experimente zur Anpassung der Fehlerfunktion

Ausgangspunkt der in diesem Kapitel beschriebenen Experimente ist der Einzelworterkenner für 62 Wörter, wie er auch schon in Abschnitt 6.5 Anwendung gefunden hat. Abbildung 6.4 zeigt nun zwei Histogramme für die auftretenden Werte von d vor und nach dem diskriminativen Training mit Zielfunktion Minimaler Wortfehler (vgl. Abschnitt 6.12) sowie den Verlauf der Gewichtungsfunktion $\frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)}$. Es wird nun das Histogramm für Werte von d vor dem diskrimi-

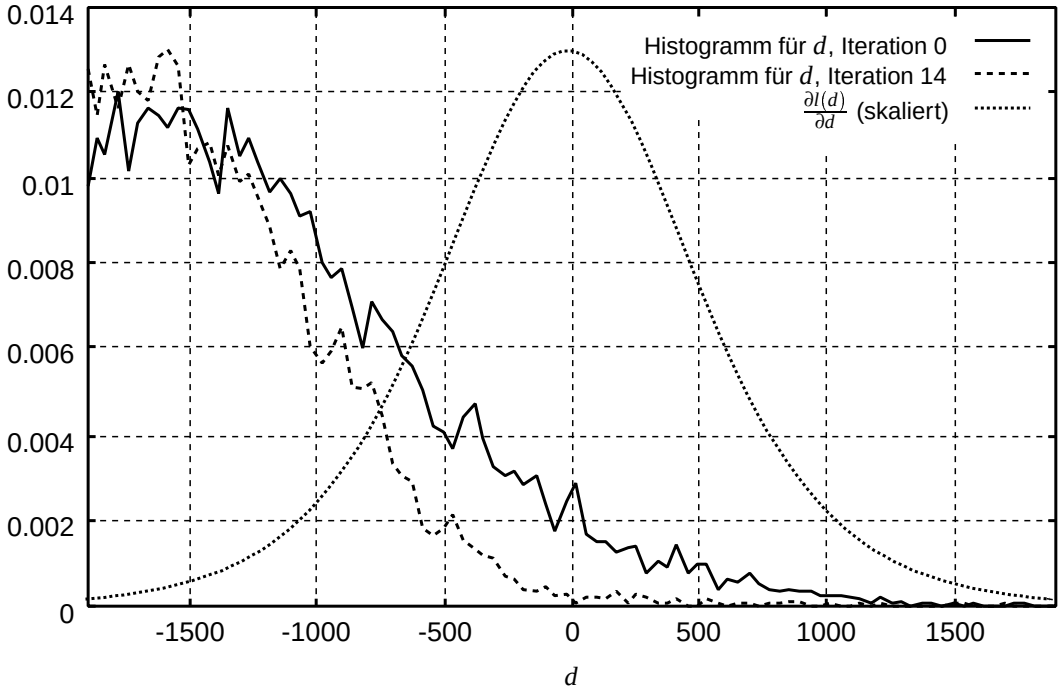

Abbildung 6.4: *Histogramm für Werte von d vor (Iteration 0) und nach (Iteration 14) diskriminativem Training, sowie* $\frac{\partial l(d)}{\partial d}$ *(skaliert) bei* $\gamma = 0.003$

nativem Training (durchgezogene Kurve) betrachtet. Für $d > 0$ sinkt die Häufigkeit von Werten d stark ab und erreicht spätestens bei $d \approx 2000 = \hat{d}$ annähernd den Wert 0. Nach Punkt zwei des Schemas aus Abschnitt 6.6 soll nun der Parameter γ so bestimmt werden, daß die Gewichtungsfunktion $\frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)}$ auch für diesen Wert von $\hat{d} = 2000$ stark abgesunken ist. Zunächst wird folgende Näherung angenommen:

$$\frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)}|_{d \gg \gamma^{-1}} \approx 1 - l(d(X,\Lambda)) \tag{6.14}$$

Mit dieser Näherung läßt sich für die Bestimmung von γ bei der Vorgabe

$$l(\hat{d}) = 0.01 \cdot \max_{d} l(\hat{d}) \tag{6.15}$$

folgende einfache Gleichung angeben:

$$\gamma = \frac{-1}{\hat{d}} \cdot \ln \frac{0.01}{1-0.01} \tag{6.16}$$

Im Beispiel ergibt sich bei $\hat{d} = 2000$ ein Wert von $\gamma \approx 0.003$. Der (geeignet skalierte) Verlauf der Gewichtungsfunktion ist in Abbildung 6.4 für $\gamma = 0.003$ eingezeichnet (gepunktete Kurve).

Für das zweite in Abbildung 6.4 dargestellte Histogramm (gestrichelte Kurve) wurden nun 15 MCE–Iterationen mit $\gamma = 0.003$ durchgeführt und in der fünfzehnten Iteration das Histogramm der auftretenden Werte von d bestimmt. Dieses zweite Histogramm ist mit einer gestrichelten Kurve eingezeichnet. Setzt man nun das Histogramm vor dem MCE–Training und das Histogramm nach dem MCE–Training in Beziehung, so fällt der Bezug zur Gewichtungsfunktion leicht ins Auge: die Häufigkeiten sind offensichtlich in einem Maße durch das diskriminative Training reduziert worden, wie es die Gewichtungsfunktion angibt. Die stärkste Reduktion findet sich im Bereich von $d \approx 0$, wohingegen die beiden Histogramme für $|d| > 1500$ kaum unterschiedlich sind.

In den Abbildungen 6.5 und 6.6 sind die entsprechenden Histogramme und Gewichtungsfunktionen für $\gamma = 0.001$ und $\gamma = 0.05$ dargestellt. Gegenüber dem als optimal angenommenen Wert von $\gamma = 0.003$ ist der Wert von $\gamma = 0.001$ entschieden zu klein und der Wert von $\gamma = 0.05$ entschieden zu groß. Entsprechend ist die Gewichtungsfunktion in Abbildung 6.5 sehr viel breiter und in Abbildung 6.6 sehr viel schmäler als in Abbildung 6.4.

Zunächst wird die Auswirkung eines zu kleinen Werts für γ wie in Abbildung 6.5 betrachtet. An dem Histogramm für d nach dem MCE–Training wird deutlich, wie sich die veränderte Gewichtungsfunktion auf das diskriminative Training auswirkt. Die Häufigkeiten von d werden auf einen viel breiteren Bereich als bei $\gamma = 0.003$ (Abbildung 6.4) reduziert. Allerdings ist auch festzustellen, daß die Häufigkeiten im Bereich um $d \approx 0$ nicht so stark reduziert werden konnten wie im Fall des optimalen Werts für γ. Man könnte es so formulieren, daß sich das Training auf einen zu großen Bereich von d konzentriert hat, aber nicht in der Lage war, für die Mehrzahl der Trainingsmuster das Optimierungsziel zu erreichen.

Im Gegensatz dazu erweist sich der Bereich, in dem das diskriminative Training bei $\gamma = 0.05$ (Abbildung 6.6) wirksam geworden ist, als viel zu klein. In das Histogramm für d ist quasi eine kleine Kerbe gedrückt worden. Nur in einem kleinen Bereich um $d \approx 0$ wurden die Häufigkeiten stark dezimiert, im äußeren Bereich von ca. $|d| > 100$ findet sich fast kein Unterschied zwischen den Histogrammen vor und nach dem diskriminativen Training. Das MCE–Training hat sich offensichtlich auf Äußerungen mit kleiner Diskriminanzfunktion konzentriert und sich auch auf

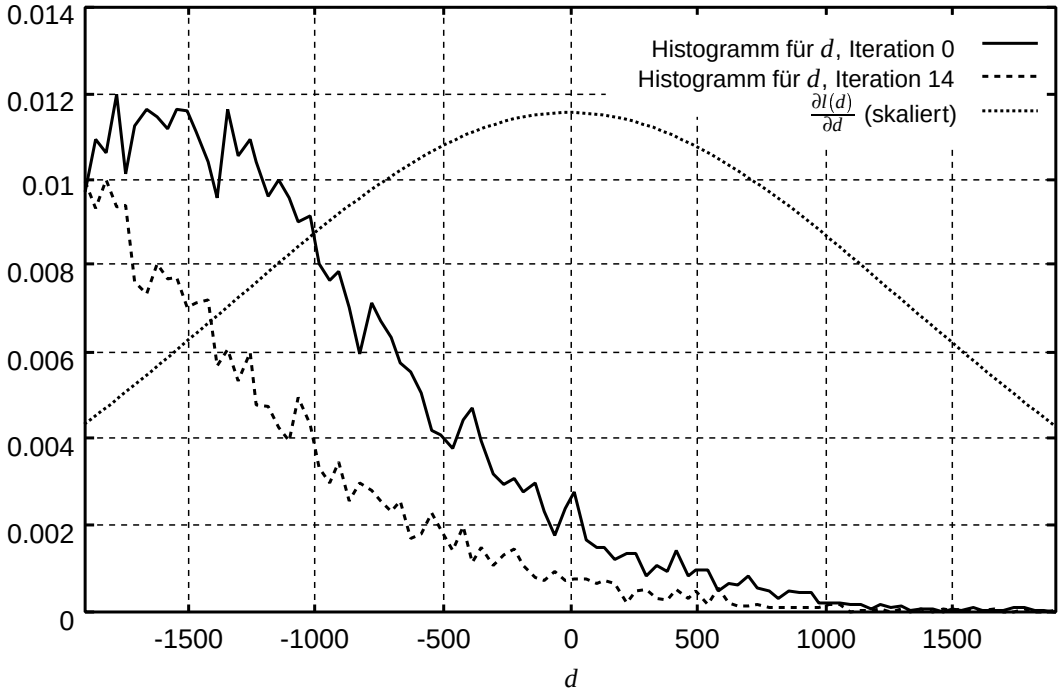

Abbildung 6.5: *Histogramm für Werte von d vor (Iteration 0) und nach (Iteration 14) diskriminativem Training, sowie* $\frac{\partial l(d)}{\partial d}$ *(skaliert) bei* $\gamma = 0.001$

Falschklassifikationen mit $d \gg 0$ kaum ausgewirkt. Vergleicht man die auf dem Trainingsmaterial erzielten Fehlerraten, so fällt auch auf, daß für den zu großen Wert von $\gamma = 0.05$ die Fehlerrate mit 1.5% dreimal so hoch ist wie im Fall des optimalen Werts $\gamma = 0.003$.

In Tabelle 6.4 sind noch einmal die auf Trainingsmaterial und Testmaterial erzielten Fehlerraten nach diskriminativem Training mit den drei unterschiedlichen Werten für γ dargestellt. Während der Unterschied zwischen den auf dem Testmaterial erzielten Fehlerraten für den opti-

γ	Wortfehlerrate	
	Trainingsmaterial	Testmaterial
0.001	1.0	1.9
0.003	0.5	1.7
0.05	1.5	2.3

Tabelle 6.4: *Fehlerraten auf dem Testmaterial bei diskriminativem Training für verschiedene Werte von* γ*, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

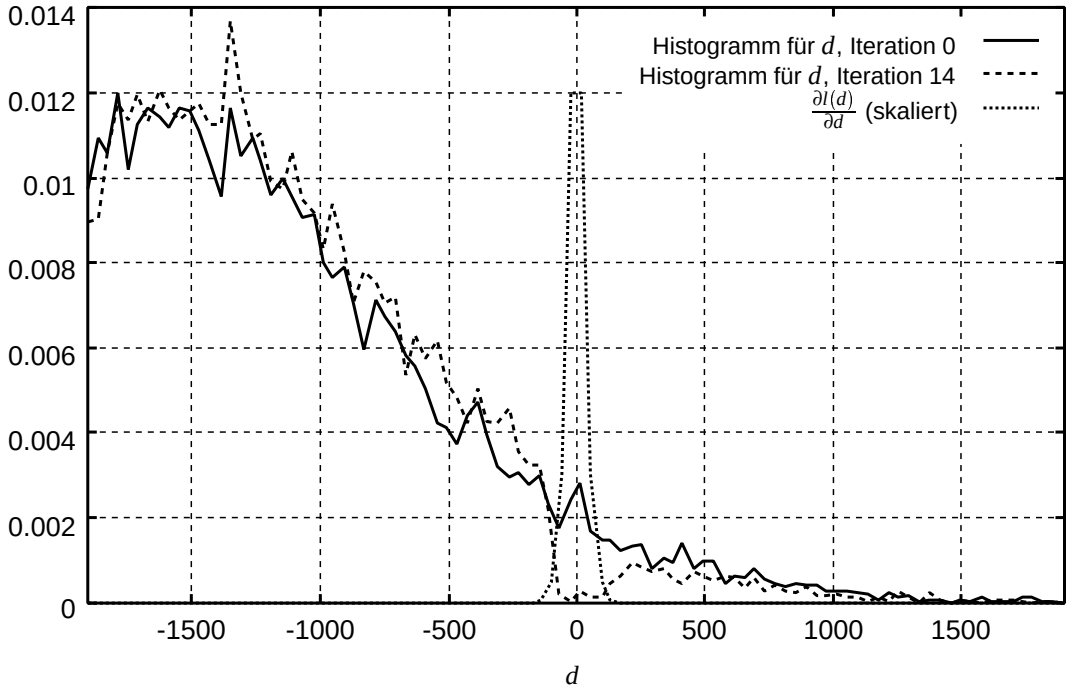

Abbildung 6.6: *Histogramm für Werte von d vor (Iteration 0) und nach (Iteration 14) diskriminativem Training, sowie* $\frac{\partial l(d)}{\partial d}$ *(skaliert) bei* $\gamma = 0.05$

malen und den zu kleinen Wert für γ mit 0.2% (absolut) nicht sehr hoch ist, ist der Unterschied zwischen dem Fall $\gamma = \gamma_{\text{opt}} = 0.003$ und $\gamma_{\text{opt}} < \gamma = 0.05$ sehr signifikant.

Zusammenfassend kann gesagt werden, daß sich das in Abschnitt 6.6 vorgestellte Schema zur Bestimmung von γ als sehr geeignet erweist. In der Praxis erscheint es insbesondere wichtig, der Wert von γ nicht zu groß und somit die Breite der Gewichtungsfunktion nicht zu klein zu wählen.

6.8 Normierung der Gradienten und Standardisierte Schrittweite

In diesem Abschnitt soll zum einen die Frage beantwortet werden, wie in der Praxis eine geeignete Schrittweite ε gewählt werden kann, und zum anderen soll eine spezielle Normierung der Gradienten eingeführt werden.

6.8.1 Normierung der Gradienten

Theoretisch betrachtet ist es eigentlich eindeutig, wie die Parameternachschätzung ausgehend von der Nachschätzformel (6.10) auszusehen hat. Es liegt jedoch der Gedanke nahe, die Gradienten Δo für einen Parameter o entsprechend der Anzahl der Ereignisse zu normieren, die zur Bildung von Δo beigetragen haben. Geht man beispielsweise von einer bzgl. der Häufigkeit der einzelnen Wörter stark inhomogenen Trainingsmenge aus, so könnte durch geeignete Normierung erreicht werden, daß die Modelle der häufigsten Wörter nicht überproportional nachgeschätzt werden.

Ausgehend von dieser Idee wird folgendes Schema zur Normierung der Gradienten definiert:

1. Während der Iteration wird die Anzahl der Beiträge N_o zur Nachschätzung des Parameters o festgehalten. Diese Beiträge werden als $\Delta o'$ akkumuliert. Gezählt und aufaddiert werden nur Beiträge, bei denen die Gewichtungsfunktion $\frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)}$ mindestens dem Maximum der Gewichtungsfunktion mal einem festem Faktor beträgt.

2. Zum Ende der Iteration wird der akkumulierte Beitrag mit $1/N_o$ skaliert und aufaddiert:

$$\hat{o} = o + \varepsilon \cdot \frac{\Delta o'}{N_o} \tag{6.17}$$

$$\hat{o} = o + \varepsilon \cdot \Delta o \tag{6.18}$$

Die Beschränkung auf Beiträge mit hoher Gewichtungsfunktion ist insbesondere wichtig, damit nicht sehr viele sehr kleine Beiträge zur Herunterskalierung des Gradienten führen. Als Faktor hat sich in der Praxis ein Wert von $1/100$ bewährt.

6.8.2 Standardisierte Schrittweite

Im folgenden soll nun ein Schema entwickelt werden, das es erlaubt, einen geeigneten Wert für die Schrittweite in der Praxis einfach zu bestimmen:

1. Bei der ersten MCE–Iteration werden die Beiträge zum normierten Gradienten Δo als Zufallsvariable betrachtet und dessen Standardabweichungen $\sigma_{\Delta o,k=1}$ bestimmt.

2. Mit Hilfe dieser Standardabweichungen läßt sich nun eine Art normierter Schrittweite definieren:

$$\hat{o} = o + \frac{\varepsilon}{\sigma_{\Delta o,k=1}} \cdot \Delta o \tag{6.19}$$

3. Der Normierungsfaktor $1/\sigma_{\Delta o,k=1}$ wird für alle weiteren Iterationen $k > 1$ beibehalten.

Bei der Betrachtung der Beiträge zu den Gradienten als Zufallsvariable wird zusätzlich noch nach Art der Parameter unterschieden. Konkret werden die Normierungsfaktoren für die Mittelpunktsvektoren und die Mixturstrafen getrennt bestimmt. Für den Fall der Mittelpunktsvektoren ergibt sich die Zufallsvariable durch die Menge aller Gradienten $\Delta o_{s,m,d}$, wobei der Zustandindex s, der Modenindex m und der Dimensionsindex d über ihren Wertebereichen variiert werden. Für die Gradienten der Mixturstrafen $\Delta o_{s,m}$ werden nur Zustandindex und Modenindex variiert.

6.9 Experimente zum erweiterten Gradientenverfahren

Anhand verschiedener Versuche soll in diesem Abschnitt untersucht werden, ob zum einen die Gradientennormierung nach Gleichung (6.18) von praktischem Vorteil ist, und wie zum anderen mit Hilfe der standardisierten Schrittweite die Konvergenz des diskriminativen Trainings gesteuert werden kann.

Grundlage der Untersuchung ist wiederum der Einzelworterkenner für 62 Wörter, der im Rahmen der Versuche im Abschnitt 5.3.3 nach dem Maximum–Likelihood–Prinzip trainiert und auch im letztem experimentellen Abschnitt 6.7 verwendet wurde. Das Maximum–Likelihood–Modell wurde jeweils 20 Iterationen diskriminativen Trainings (Minimaler Wortfehler) der Mittelpunktsvektoren mit Zielfunktion Minimaler Wortfehler unterzogen. In der ersten Versuchsreihe (Abbildung 6.7) wurde die Normierung der Gradienten nicht durchgeführt. Dem entspricht ein künstliches Festhalten der Werte N_o in Gleichung (6.18) auf dem Wert 1. In der zweiten Versuchsreihe (Abbildung 6.8) wurde die Gradientennormierung nach (6.18) angewandt. Für beide Versuchsreihen wurde die (standardisierte) Schrittweite für die Mittelpunktsvektoren ε in einem Bereich von 0.25 bis 1.75 in Schritten von 0.25 variiert.

In Abbildung 6.7 und 6.8 sind nun jeweils die Wortfehlerraten auf dem Trainingsmaterial nach jeder Iteration aufgetragen. Vergleicht man das Konvergenzverhalten mit und ohne Gradientennormierung anhand der Abbildungen 6.7 und 6.8, so ist sehr leicht zu erkennen, daß die Gradientennormierung von praktischem Nutzen ist. Im Fall ohne Gradientennormierung kann eigentlich nur für einen Wert von $\varepsilon = 0.5$ von verhältnismäßig guter Konvergenz gesprochen werden. Im Fall mit Gradientennormierung ergibt sich für den gesamten Bereich von $\varepsilon = 0.5$ bis $\varepsilon = 1.5$ ein gutes Konvergenzverhalten.

Ähnlich den Verläufen der Fehlerraten auf dem Trainingsmaterial ergibt sich auch das Bild für die Fehlerraten auf dem Testmaterial, die in Tabelle 6.5 aufgelistet sind. Mit Gradientennormierung ergibt sich dort ein Bereich von $\varepsilon = 0.75$ bis $\varepsilon = 1.75$, in dem sich die Fehlerraten nicht signifikant unterscheiden. Ohne Gradientennormierung ergibt sich lediglich ein Bereich $\varepsilon = 0.5$ bis $\varepsilon = 1.0$.

Ausgehend von den obigen Ergebnissen wurde für alle folgenden Versuche die Gradientennormierung verwendet. Für die standardisierte Schrittweite für die Nachschätzung der Mittel-

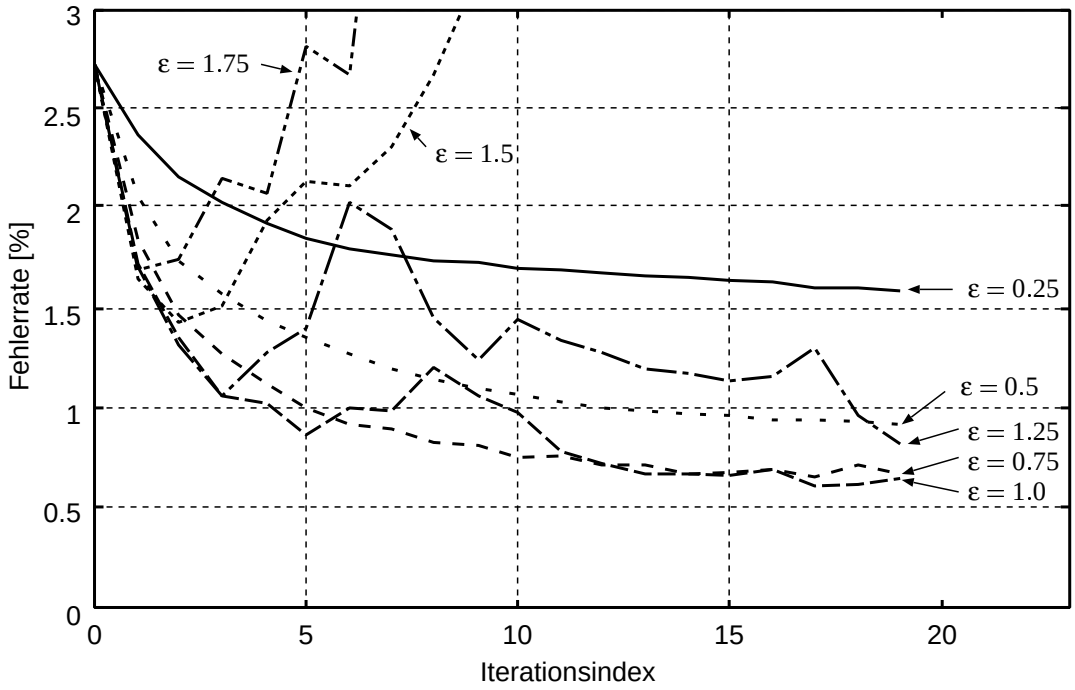

Abbildung 6.7: *Konvergenzverhalten für unterschiedliche Schrittweiten* ε *ohne Gradientennormierung, Verlauf der Fehlerrate auf dem Trainingsmaterial (Trainingsmenge von VM–62)*

punktsvektoren wurde zunächst immer ein Wert von $\varepsilon = 1.5$ verwendet. Bei einigen Versuchen hat sich jedoch mit dieser Schrittweite keine einwandfreie Konvergenz ergeben. In diesen Fällen war es teilweise nötig, die Schrittweite auf 1.25 oder 1.0 zu reduzieren. Bei Nachschätzung der Mixturstrafen (vgl. Abschnitt 6.5) erwies sich ein erheblich höherer Wert der (normierten) Schrittweite von 100 als geeignet.

6.10 Zusammenhang zwischen freien Parametern, Trainingsmenge und Wirksamkeit diskriminativer Methoden

Einer der Gründe, die für die Anwendung diskriminativer Methoden sprechen, ist die mangelnde Optimalität des Maximum–Likelihood–Prinzips für den Fall, daß das Modell den Prozeß der Spracherzeugung nur unzureichend beschreibt (vgl. Abschnitt 6.1). Ausgehend von diesem Gedanken liegt die Vermutung nahe, daß diskriminative Methoden umso notwendiger sind, je schlechter die Modelle die Wirklichkeit beschreiben. Besonders grob ist eine Modellierung beispielsweise, wenn für die Modellierung der Emissionswahrscheinlichkeiten eine sehr kleine Anzahl von Basisfunktionen verwendet wird.

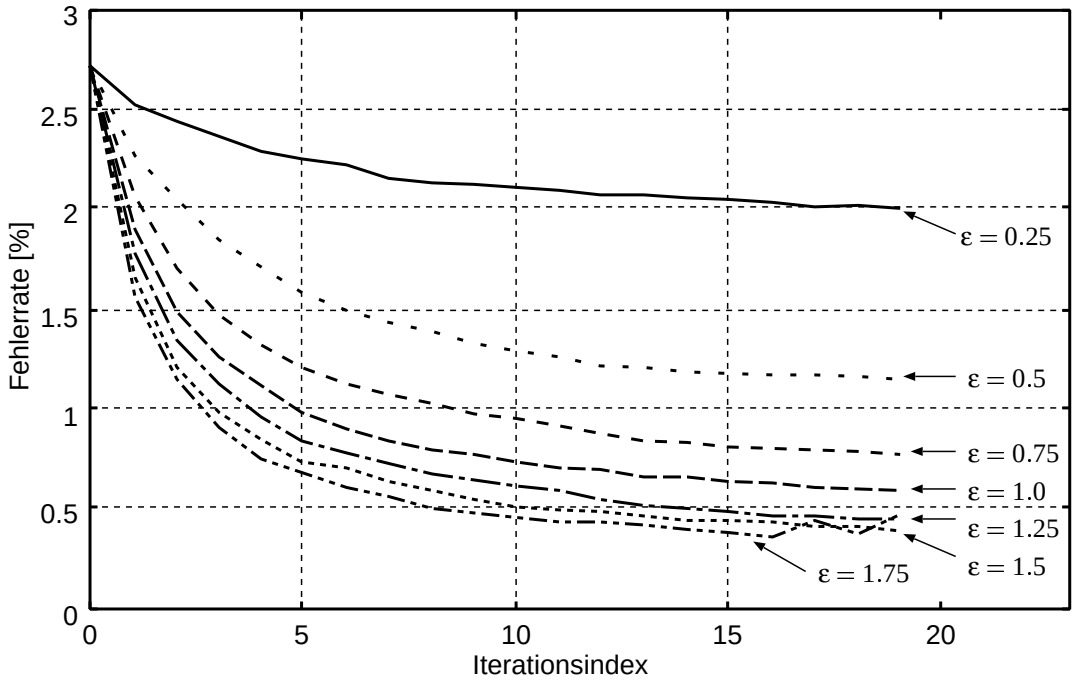

Abbildung 6.8: *Konvergenzverhalten für unterschiedliche Schrittweiten* ε *mit Gradientennormierung, Verlauf der Fehlerrate auf dem Trainingsmaterial (Trainingsmenge von VM–62)*

Eine weiterer Punkt, der für die Optimalität des Maximum–Likelihood–Prinzips grundsätzlich nicht erfüllt ist, ist eine ausreichend große Menge von Trainingsmaterial. Hiervon könnte man die These ableiten, daß diskriminative Methoden bei wenig Trainingsmaterial von besonderem Vorteil gegenüber Maximum–Likelihood sind.

Eine gelungene Darstellung des Zusammenhangs zwischen der Anzahl freier Parameter und der Größe der Trainingsmenge bei Maximum–Likelihood–basiertem Training findet sich in [Bub, 1999]. Dort wurde sogar die gleiche Datenbank und das gleiche Vokabular (VM–62) verwendet wie bei den Versuchen im folgenden Abschnitt. Im folgenden Abschnitt wird nun versucht, den genannten Zusammenhang für die Verhältnisse bei diskriminativer Parameterschätzung zu ergründen.

6.11 Experimente zur Menge freier Parameter und Größe der Trainingsmenge

Beide Versuchsreihen beruhen im folgenden wiederum auf der Trainings– und Erkennungsaufgabe VM–62. Sämtliche Parameter bis auf die teilweise variierte Modellgröße entsprechen denen

ε	Wortfehlerrate	
	ohne Gradientennormierung	mit Gradientennormierung
0.25	2.2	2.5
0.5	1.9	2.0
0.75	1.8	1.9
1.0	1.8	1.8
1.25	2.4	1.7
1.5	(Divergenz)	1.7
1.75	(Divergenz)	1.7

Tabelle 6.5: *Wortfehlerraten auf dem Testmaterial mit und ohne Gradientennormierung bei unterschiedlicher (normierter) Schrittweite ε, Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

in den vergangenen experimentellen Abschnitten (z.B. 6.9).

Die Ergebnisse der ersten Versuchsreihe sind in Abbildung 6.9 dargestellt. Dort wurden zunächst drei Maximum–Likelihood–Modelle mit einfachen (447) Dichten (single densities), mit insgesamt 1838 Dichten und mit 3678 Dichten erstellt. Das entspricht einer mittleren Anzahl 1, ca. 4 und ca. 8 Dichten pro Segment. Zunächst wurden für diese drei Modelle die Wortfehlerraten auf dem Trainingsmaterial sowie dem Testmaterial gemessen. Die gemessenen Fehlerraten der Maximum–Likelihood–Modelle sind in Abbildung 6.9 mit den schraffierten Balken dargestellt.

Für alle drei Modellgrößen wurde ausgehend von den Maximum–Likelihood–Modellen ein diskriminatives Training mit Zielfunktion Minimaler Wortfehler durchgeführt. Die mit den diskriminativ trainierten Modellen erzielten Fehlerraten sind in der Abbildung 6.9 mit den unschraffierten Balken dargestellt.

Für alle Versuche gelten die beiden folgenden Aussagen: Die Fehlerraten auf dem Trainingsmaterial ist bei gleicher Modellgröße und gleichem Optimierungsprinzip stets niedriger als auf dem Testmaterial. Die durch die diskriminativ trainierten Modelle erzielten Fehlerraten sind bei gleicher Modellgröße und gleicher Stichprobe stets niedriger als die der Maximum–Likelihood–Modelle.

Die Verbesserung der Fehlerrate auf dem Testmaterial durch das diskriminative Training reduziert sich mit steigender Modellgröße. Während bei den einfachen Dichten die Reduktion der Fehlerrate ca. 50% beträgt, ergibt sich für 1838 bzw. 3678 Dichten nur eine Reduktion von ca. 43% bzw. ca. 30%. Offensichtlich gilt folgendes Prinzip: die Wirksamkeit des diskriminativen Trainings reduziert sich mit steigender Anzahl freier Parameter bzw. der Modellgröße.

Bei einem ähnlichen Experiment nur mit Einzelziffern, das in [Bauer, 1997] veröffentlicht

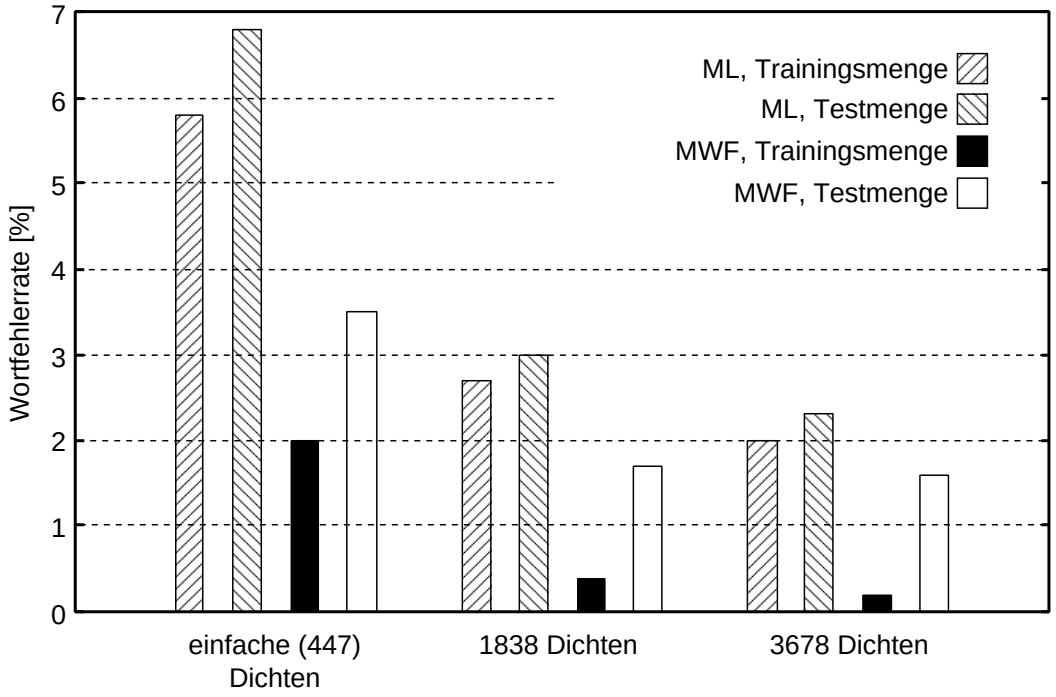

Abbildung 6.9: *Wortfehlerraten von Maximum–Likelihood (ML) Ausgangsmodell und diskriminativ trainiertem (Minimaler WortFehler: MWF) Modell bei Variation der Modellgröße (Gesamtzahl der Verteilungsdichten), Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62.*

wurde, ergab sich beim Übergang von 1000 Dichten auf 2000 Dichten ein leichter Anstieg der Testfehlerrate für das diskriminativ trainierte Modell. Dabei wurde für die Maximum–Likelihood–Modelle mit 1000 bzw. 2000 Dichten die gleiche Testfehlerrate gemessen. Offenbar hat sich dort eine Überanpassung des diskriminativ trainierten Modells an das Trainingsmaterial eingestellt, die zu einem Verlust an Erkennungsgenauigkeit auf dem Testmaterial geführt hat.

In einer zweiten Versuchsreihe wurde die Anzahl der Dichten mit 1838 stets gleich belassen, aber die Größe des Trainingsmaterials variiert. Von der Gesamtzahl von 526 Sprechern der Trainingsdatenbank wurden 5%, 10%, 20%, ...und 100% der Sprecher zum Training verwendet. Wie in den bisherigen Versuchen wurde zunächst ein Maximum–Likelihood–Modell trainiert und dieses anschließend mit Zielfunktion Minimaler Wortfehler nachtrainiert. Es wurden jeweils die Wortfehlerrate auf dem Trainingsmaterial und dem Testmaterial gemessen. Der Verlauf der Fehlerrate bei variierender Größe des Trainingsmaterials wird in Abbildung 6.10 dargestellt. Zunächst ist erkennbar, daß die Fehlerrate auf dem Testmaterial für beide Optimierungsprinzipien stetig mit wachsender Größe des Trainingsmaterials sinkt (nicht–signifikante Abweichungen außer acht gelassen). Das diskriminative Training scheint jedoch die zunehmende Menge

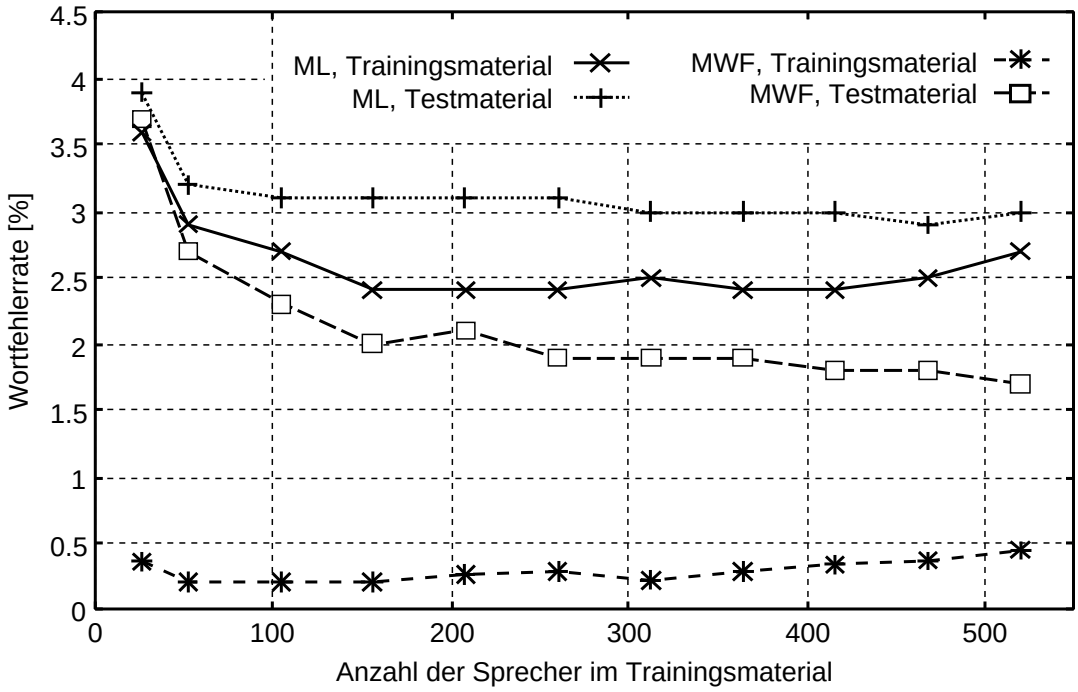

Abbildung 6.10: *Verlauf der Wortfehlerraten für Maximum–Likelihood (ML) bzw. Minimaler WortFehler (MWF) Parameterschätzung bei unterschiedlicher Größe der Trainingsmenge. Training: (Teile der) Trainingsmenge von VM–62, Test: Testmenge von VM–62*

an Trainingsmaterial besser ausnutzen zu können: der Abstand zwischen den Testfehlerraten Maximum–Likelihood und Minimaler Wortfehler nimmt mit wachsendem Trainingsmaterial zu, wobei die Maximum–Likelihood Testfehlerrate ab ca. 300 Sprecher nicht mehr signifikant sinkt.

Die These, daß diskriminative Methoden bei kleiner Trainingsstichprobe besonders effektiv sind (vgl. Abschnitt 6.10), bestätigt sich in der vorliegenden Versuchsreihe nicht. Vielmehr erweist sich das diskriminative Training bei großer Trainingsstichprobe als besonders wirksam.

Erstaunlicherweise zeigt die Fehlerrate auf dem Trainingsmaterial sowohl für Maximum–Likelihood als auch für Minimaler Wortfehler ein Minimum bei mittlerer Größe des Trainingsmaterial. Dabei ist die Dynamik bei Maximum–Likelihood-basiertem Training sehr viel höher als bei diskriminativem Training. Eine Erklärung für das Ansteigen der Fehlerraten auf dem Trainingsmaterial bei sehr kleiner Trainingsmenge läßt sich ohne weiteres nicht angeben. Eine mögliche Erklärung liegt in einer Überanpassung auf einem kleinem Teil der Trainingsmenge aufgrund des schlechten Verhältnisses zwischen freien Parametern und Größe der Trainingsstichprobe.

6.12 Wahl der Klassen

Das MCE–Verfahren wurde in Abschnitt 6.2 zunächst unabhängig von einer möglichen Wahl der Klassen für beliebige Muster S definiert. Obwohl in allen experimentellen Untersuchungen zum MCE–Verfahren in den vorhergehenden Abschnitten Wörter als Klassen verwendet wurden, ist dies keineswegs die einzige Möglichkeit. Eine andere mögliche Wahl der Klassen wären HMM–Zustände, wie bereits bei der Linearen Diskriminanz–Analyse (vgl. Abschnitt 5) verwendet. In der Praxis ergeben sich für die möglichen Klassendefinitionen gravierende Unterschiede bei der Durchführung des diskriminativen Trainings. Praktische Vor– und Nachteile in Bezug auf Rechenaufwand und Komplexität sind sowohl von der Wahl der Klassen als auch von der konkreten Trainings– und Erkennungsaufgabe abhängig. In den folgenden Abschnitten werden nun verschiedene Klassendefinitionen beschrieben und die praktischen Eigenschaften erläutert. Der Einfluß der Klassenwahl auf die Erkennungsgenauigkeit der Systeme hingegen läßt sich nur anhand von experimentellen Untersuchungen klären und wird im Abschnitt 6.13 beleuchtet.

6.12.1 HMM–Zustände

Ähnlich der Klassendefinition bei der LDA im Abschnitt 5 ist eine mögliche Wahl der Klassen für das MCE–Verfahren die der HMM–Zustände. Einem Muster S in Gleichung (6.2) entspricht damit ein einzelner Merkmalsvektor $\vec{x}$. Den Modellen λ (Gleichung (6.4) bzw. (6.6)) entsprechen die Emissionswahrscheinlichkeiten der Zustände $b_s(\vec{x})$ entsprechend Gleichung (2.9). Die Zielfunktion entspricht damit einer Approximation der Zustandfehlerrate, also der Klassifikationsfehlerrate für isolierte Merkmalsvektoren wobei die Klassen den HMM–Zuständen entsprechen.

Im Sinne einer möglichst hohen Korrelation der Zielfunktion mit dem Ziel der realen Applikation ist eine solche Definition der Zielfunktion eigentlich wenig sinnvoll. Zum einen haben jedoch die Untersuchungen im Abschnitt 5 gezeigt, daß prinzipiell ein Zusammenhang zwischen der Verwechselbarkeit von HMM–Zuständen und Wörtern gegeben ist. Zum anderen ergeben sich praktische Vorteile in Bezug auf die zum Training notwendige Rechenleistung bei Zielfunktion Minimaler ZustandsFehler (MZF).

Eine besondere Rolle spielt in diesem Zusammenhang der Pause–Zustand (vgl. Abschnitt 2.3.3). Zum einen ist der Pause–Zustand aufgrund der großen nichtsprachlichen Abschnitte sehr viel häufiger als die anderen Zustände der Wörter. Zum anderen ist eine Klassifikation von isolierten Signalabschnitten in Sprache bzw. Nicht–Sprache äußerst schwierig. Aus genannten Gründen wird die Zustandsfehlerrate grundsätzlich ohne Einbeziehung der nicht–sprachlichen Abschnitte definiert. In der Praxis wird dies so umgesetzt, daß die sprachlichen Abschnitte mittels einer erzwungenen Viterbizuordnung (vgl. Abschnitt 3.1) bestimmt werden.

Mit den beschriebenen Voraussetzungen ergibt sich in der Praxis das in Abbildung 6.11 dargestellte Schema zur Umsetzung eines Trainings mit Zielfunktion Minimaler Zustandsfeh-

ler (MZF).

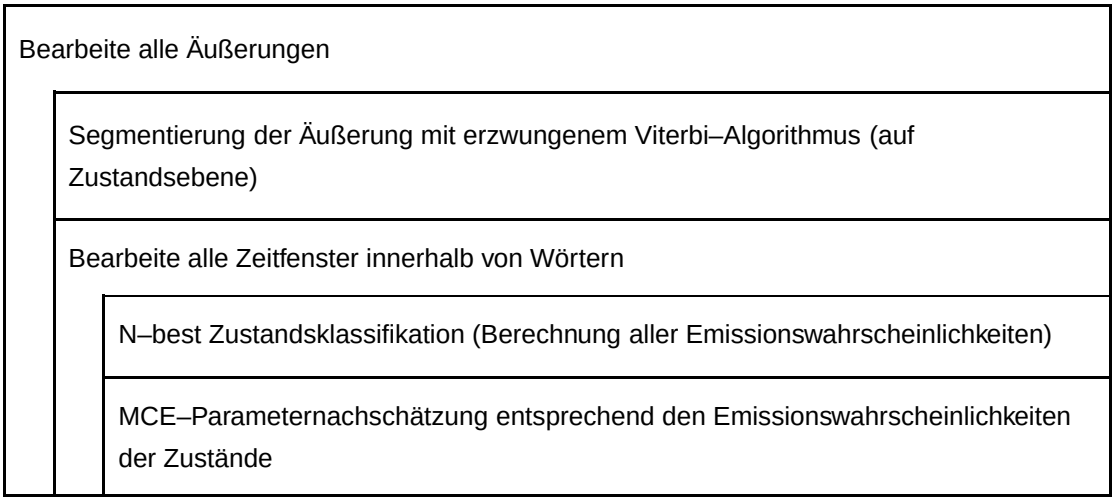

Abbildung 6.11: *Struktogramm für Training mit Zielfunktion Minimaler Zustandsfehler*

Der rechenaufwendigste Schritt bei der praktischen Umsetzung besteht in der Berechnung aller Emissionswahrscheinlichkeiten bzw. der Emissionsstrafen. Der Aufwand hierfür steigt linear mit der Anzahl der Moden in einem Hidden–Markov–Modell. Doch selbst bei großen HMMs (etliche tausend Dichten) ist dies auf modernen Rechensystemen (Taktfrequenz ca. 500 MHz) in Realzeit realisierbar. Daneben existieren auch Algorithmen zur Reduktion des Rechenaufwands für die Emissionsberechnung, die eine Berechenbarkeit in einem Bruchteil von Realzeit ermöglichen ([Beyerlein, 1994], [Ortmanns u. a., 1997]). Im Rahmen dieser Arbeit werden jedoch keine solchen Algorithmen verwendet.

6.12.2 Phoneme

Eine weitere mögliche Wahl der Klassen für das MCE–basierte Training, die in der Literatur ([Reichl und Ruske, 1995]) sehr häufig verwendet wird, ist die der Phoneme bei Systemen mit phonetischem Ansatz (vgl. Abschnitt 2.3.3). Im Fokus dieser Arbeit soll hierbei die Fähigkeit des Systems stehen, eine Merkmalsvektorfolge einem einzelnen Phonem richtig zuzuordnen. Die Zielfunktion entspricht dabei einer Approximation der Einzelphonemfehlerrate des Systems. Dazu wird die gesamte Merkmalsfolge einer Äußerung mittels erzwungenem Viterbi–Algorithmus in einzelne Abschnitte, die je einem Phonem entsprechen, zerlegt. Diesen Abschnitten zugehörige Folgen von Merkmalsvektoren entsprechen den Mustern S in Gleichung (6.2). Den Modellen in Gleichung (6.2) entsprechen die Hidden–Markov–Modelle für Phoneme — im Deutschen sind dies ca. 40 Modelle.

Für die praktische Umsetzung des Verfahrens ergibt sich das in Abbildung 6.12 durch ein Struktogramm dargestellte Schema. Die größte Rechenleistung in der praktischen Umsetzung

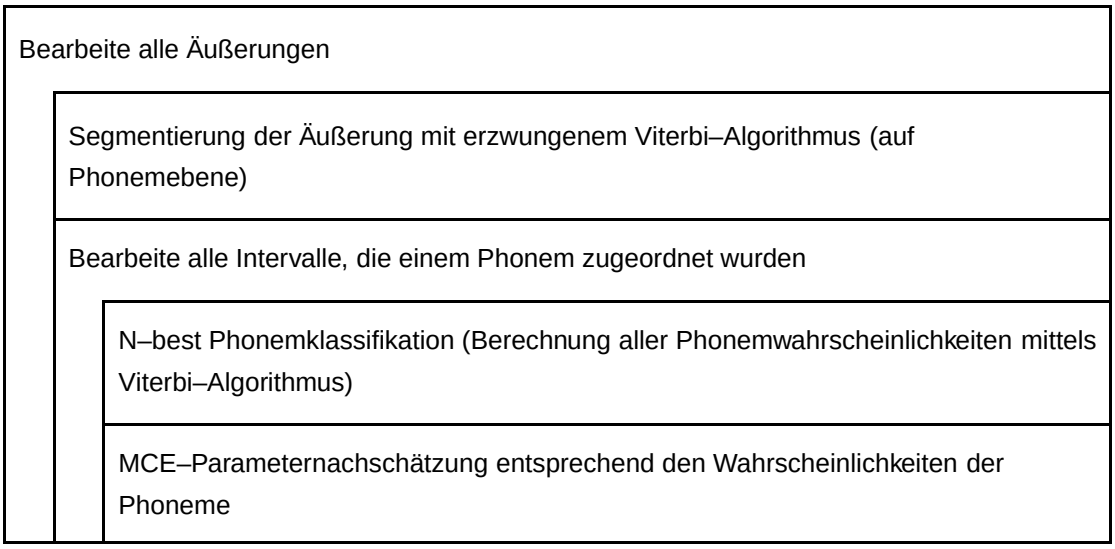

Abbildung 6.12: *Struktogramm für Training mit Zielfunktion Minimaler Phonemfehler*

wird für die N–best Phonemklassifikation benötigt. Dabei ist jedoch auch für große Modelle mit sehr vielen Dichten die Verarbeitung in einem Bruchteil der Realzeit bei Mikroprozessoren mit Taktfrequenzen um die 500 MHz möglich.

Ähnlich wie bei der Zielfunktion Minimaler Zustandsfehler ist auch die Phonemerkennungsrate nicht direkt mit der aus der Sicht der Anwendung relevanten Größe — der Worterkennungsrate — korreliert. Trotzdem wird die Phonemerkennungsrate häufig als Gütemaß eines Spracherkennungssystems eingesetzt ([L. F. Lamel, 1993]), da es eine allgemeine Aussage über die Leistungsfähigkeit erlaubt — unabhängig vom Wortschatz. Neben dieser theoretischen Eignung besitzt die Zielfunktion Minimaler PhonemFehler (MPF) auch den Vorteil, daß die notwendige Rechenleistung eine Verarbeitung des Trainingsmaterials auf schnellen Mikroprozessoren in Realzeit oder schneller erlaubt.

Untersucht werden im Rahmen dieser Arbeit nur kontextunabhängige Phonemmodelle. Wollte man das Verfahren für kontextabhängige Phonemmodelle verwenden, so müßte man der Phonemklassifikation die entsprechenden Kontexte vorgeben. Prinzipiell könnte die Optimierung aber auch hierfür angewendet werden.

Denkbar wäre auch eine Definition der Zielfunktion im Sinne einer kontinuierlichen Phonemerkennung, wobei dies dann als Minimaler PhonemFolgenFehler (MPFF) bezeichnet werden könnte. Für die getroffene Einschränkung, daß für das MCE–Training nur das korrekte Modell

und das beste konkurrierende Modell ($\eta = \infty$, vgl. Abschnitt 6.2) verwendet wird, erscheint eine Verwendung dieser Zielfunktion jedoch kaum sinnvoll. Da in der Praxis die (Einzel–) Phonemerkennungsrate deutlich unter 90% liegt, ergibt sich aufgrund der hohen Anzahl von Phonemen in der gesamten Äußerung eine Phonemfolgenfehlerrate von fast 100%. Damit ergibt sich eine praktisch kaum zu minimierende Zielgröße.

6.12.3 Wörter

Mit der Verwendung von Wörtern als Klassen ist die Zielfunktion direkt mit der für die reale Anwendung — zumindest bei Einzelworterkennung — relevanten Größe korreliert. Die Zielfunktion approximiert die Wortfehlerrate auf dem Trainingsmaterial. Damit birgt ein diskriminatives Training mit Wörtern als Klassen theoretisch ein höheres Potential zur Optimierung im Sinne der Anwendung als die Verwendung von Zuständen oder Phonemen, bei denen die Korrelation zwischen Zielfunktion und Zielgröße der Anwendung nur bedingt gegeben ist.

Um beim diskriminativen Training mit Zielfunktion Minimaler WortFehler (MWF), das eigentlich auf die Optimierung eines Einzelworterkenners abzielt, auch kontinuierlich gesprochene Sprache als Trainingsmaterial verwenden zu können, wird in der praktischen Umsetzung eine Segmentierung in Einzelwörtern vorgeschaltet. Das für die Umsetzung des diskriminativee Trainings bei dieser Definition der Zielfunktion geeignete Schema ist in Abbildung 6.13 als Struktogramm dargestellt.

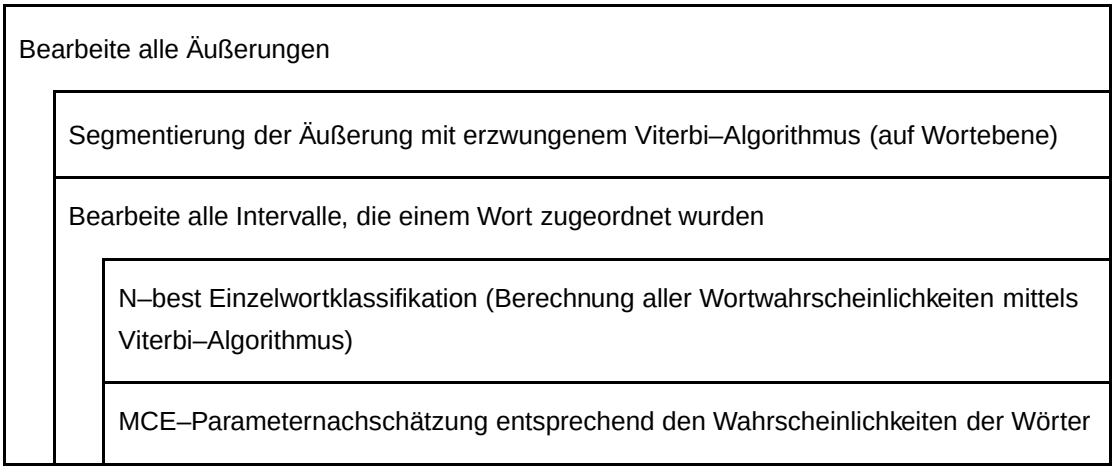

Abbildung 6.13: *Struktogramm für Training mit Zielfunktion Minimaler Wortfehler*

Den rechenintensiven Kern des Algorithmus zum MWF–Training bildet die N–best Einzelwortklassifikation mittels Viterbi–Algorithmus. Die Rechenzeit für die Einzelwortklassifikation

steigt in etwa logarithmisch mit der Anzahl der Wörter. So benötigt eine Einzelwortklassifikation für einige zig Wörter selbst bei großer Modellgröße nur einen Bruchteil der Realzeit (Taktfrequenz des Mikroprozessors ca. 500 MHz). Andererseits sind auch sinnvolle Anwendungen denkbar, bei denen die Wortschatzgröße für das MWF–Training mehrere tausend Wörter betragen kann. Dies führt in der Praxis jedoch schnell zu Verarbeitungszeiten, die ein Vielfaches der Realzeit betragen.

Die Größe des beim MWF–Training verwendeten Wortschatzes richtet sich üblicherweise nach der Anzahl der im Trainingsmaterial auftretenden Wörter. Zumindest für die Segmentierung in Einzelwörter ist der volle Wortschatz notwendig. Für die N–best Einzelwortklassifikation könnte der Wortschatz auf das für die Anwendung relevante Vokabular eingeschränkt werden. Beim sogenannten *Generalisten–Training* ist dieses Vokabular jedoch nicht gegeben. Ziel eines Generalisten–Trainings ist die Erzeugung von Hidden–Markov–Modellen für Phoneme (oder andere wiederverwertbare Einheiten), die für einen beliebigen Wortschatz zu optimalen Erkennungsergebnissen führen sollen. Die Notwendigkeit für ein Generalisten–HMM ergibt sich in der Praxis durch Anwendungen, bei denen der Wortschatz erst zur Laufzeit spezifiziert werden kann oder bei Vokabularien, zu denen kein spezielles Trainingsmaterial vorliegt.

Während also ein MWF–Training für ein Einzelwortsystem bei relativ kleinem Wortschatz und geeignetem Trainingsmaterial mit geringem Rechenaufwand durchgeführt werden kann, sind die Verarbeitungszeiten für ein Generalisten–Training meist sehr groß. Für ein Generalisten–Training besitzt also ein diskriminatives Training mit Zielfunktion Minimaler Zustands– oder Phonemfehler einen gravierenden praktischen Vorteil in Form erheblich kürzerer Verarbeitungszeiten.

6.12.4 Wortfolgen

Während für Anwendungen mit reiner Einzelworterkennung das MWF–Kriterium als optimal im Sinne der Applikation angesehen werden kann, ist dies für Anwendungen mit kontinuierlicher Worterkennung nicht der Fall. Neben den bei Einzelworterkennung auftretenden Wortsubstitutionen (Verwechslungen) ergeben sich bei kontinuierlicher Erkennung auch Wortauslöschungen und Worteinfügungen. Die Wortfehlerrate bei kontinuierlicher Erkennung wird im allgemeinen aus der Summe von Wortsubstitutionen, Wortauslöschungen und Worteinfügungen bestimmt. Eine andere geeignete Maßzahl zur Messung der Qualität eines kontinuierlichen Erkenners ist die Satz– oder auch Wortfolgenfehlerrate.

Definiert man Wortfolgen in ihrer Gesamtheit als Klassen für das MCE–Training, so approximiert die Zielfunktion die Wortfolgenfehlerrate auf dem Trainingsmaterial. Dies birgt zwar den Nachteil, daß eine gesamte Äußerung nur entweder ganz falsch oder ganz richtig ist, jedoch werden Wortauslöschungen und Worteinfügungen im Gegensatz zum MWF–Kriterium berück-

sichtigt.

In Abbildung 6.14 ist das Struktogramm für die Umsetzung des Trainingskriteriums Minimaler WortFolgenFehler (MWFF) dargestellt. Obgleich die Umsetzung des MWFF–Kriteriums

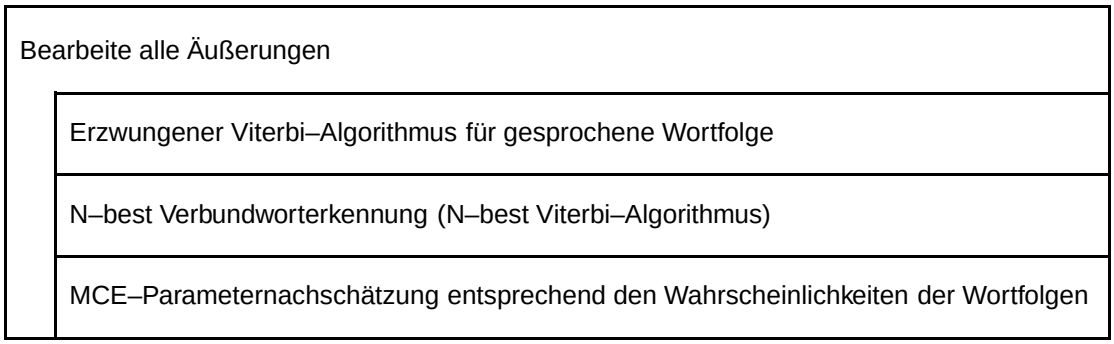

Abbildung 6.14: *Struktogramm für Training mit Zielfunktion Minimaler Wortfolgenfehler*

wegen der notwendigen kontinuierlichen N–best Suche ([Chou u. a., 1993]) sehr viel aufwendiger ist als z.B. die Umsetzung des MWF–Kriteriums, ist das Struktogramm in Abbildung 6.14 aufgrund der weggefallenen Vorsegmentierung sehr viel einfacher als das in Abbildung 6.13.

Die kontinuierliche N–best Suche muß mit $N = 2$ realisiert werden. Dann ist auch für den Fall, daß die beste erkannte Wortfolge der korrekten Wortfolge entspricht, sichergestellt, daß die beste konkurrierende Wortfolge gefunden wird. Selbst bei $N = 2$ ergibt sich in der Praxis ein recht hoher Rechenaufwand für die N–best Verbundworterkennung. Für sehr kleine Wortschätze wie z.B. Ziffern liegt die Verarbeitungsdauer in der Größenordnung der Realzeit. Jedoch bereits ab Wortschatzgrößen von einigen hundert Wörtern benötigt eine N–best Verbundworterkennung ohne Sprachmodell eine im Vergleich zur Dauer der Sprachproben sehr hohe Rechenzeit.

Bei den im Rahmen dieser Arbeit betrachteten Generalisten–Trainings wird eine Mischung aus phonetisch reichen Sätzen sowie Sprachmaterial verschiedenster Art, wie etwa Zeitangaben, verwendet. Das zur Verfügung stehende Sprachmaterial ist durch die SpeechDat–Datenbank vorgegeben ([SpeechDat–Internetseite, 2000], [Höge u. a., 1997]). Zu diesem inhomogenen Sprachmaterial läßt sich nicht ohne weiteres ein Sprachmodell erstellen, das diese Domäne sinnvoll beschreibt. Aus diesem Grunde wird im Rahmen dieser Arbeit das MWFF–Kriterium nur im Zusammenhang mit kleinen Wortschätzen betrachtet. In der Literatur finden sich Beispiele, wo diskriminative Kriterien basierend auf N–best Verbundworterkennung — jedoch in Verbindung mit einem Sprachmodell — verwendet wurden. Dort stammten Trainings– und Testmaterial aus einer eingegrenzten Domäne wie etwa Terminabsprachen, für die sich geeignete Sprachmodelle berechnen lassen.

6.13 Experimente zur Wahl der Klassen

Nachdem in den vorangegangenen Abschnitten verschiedene Möglichkeiten zur Wahl der Klassen für das diskriminative Training definiert und im Hinblick auf ihre Realisierbarkeit diskutiert wurden, sollen nun die Eigenschaften der Kriterien hinsichtlich der Erkennungsleistung untersucht werden. Dies kann natürlich nur anhand von experimentellen Untersuchungen geschehen. In den nachfolgenden Abschnitten werden mehrere grundlegende und für den praktischen Einsatz in Telefondialogsystemen wichtige Erkennungs– bzw. Trainingsaufgaben betrachtet und die sich ergebenden Fehlerraten ausgehend von den möglichen diskriminativen Kriterien verglichen.

6.13.1 Experimente mit Einzelworterkennung bei kleinem Wortschatz

Die erste Domäne, die Gegenstand der Untersuchungen sein soll, ist die Einzelworterkennung bei kleinem Wortschatz und bei ausreichend Trainingsmaterial mit diesem Wortschatz. Die in Telefondialogsystemen auftretenden Wortschätze bestehen häufig aus den Ziffern z.B. als Ersatz für eine Eingabe mittels Telefontastatur und aus einigen Kommandowörtern wie dem Wort *Hilfe* ([Littel und Höge, 1996], [Littel u. a., 1998]). Als konkrete Aufgabe soll hier die VM–62 Trainings– bzw. Erkennungsaufgabe dienen, die bereits in den vorangegangenen experimentellen Abschnitten verwendet wurde.

Für diese Aufgabe bietet sich natürlich das Kriterium des Minimalen WortFehler (MWF) an, das auch so in den vorangegangenen Abschnitten eingesetzt wurde. Neben dem MWF–Kriterium soll anhand dieser Aufgabe auch noch das Kriterium des Minimalen ZustandsFehler (MZF) untersucht werden. Zunächst soll ausgehend von einem Maximum–Likelihood optimierten Modell mit einer Fehlerrate von 3.0 % auf VM–62 mit 24 Merkmalskomponenten der prinzipielle Zusammenhang zwischen Zustandsfehlerrate und Wortfehlerrate betrachtet werden. Dazu ist in Abbildung 6.15 die Zustandsfehlerrate und die Wortfehlerrate in Abhängigkeit der Anzahl verwendeter Merkmalskomponenten aufgetragen. Der untere Teil der Abbildung 6.15 entspricht der in Abbildung 5.2 dargestellten Kurve.

Vergleicht man nun den Verlauf der beiden Kurven in Abbildung 6.15, so fällt auf, daß die Wortfehlerrate schon bei etwa 20 Komponenten ihrem Minimum sehr nahe kommt und fast nicht weiter sinkt. Andererseits sinkt die Zustandsfehlerrate auch bei mehr als 25 Komponenten noch und erreicht ihr ungefähres Minimum erst bei ca. 30 Komponenten.

Dieser einfache Vergleich zeigt, daß Wortfehlerrate und Zustandsfehlerrate natürlich korreliert sind, der Zusammenhang jedoch recht begrenzt ist. Die genauen Verhältnisse werden nun anhand von Messungen der Wort– und Zustandsfehlerraten bei unterschiedlichen Trainingskriterien ermittelt.

Das mit MWF–Kriterium trainierte Modell wurde bereits in Abschnitt 6.5 beschrieben. Der Realzeitfaktor ergibt sich hier für eine Iteration zu 0.17 (Pentium III, 700 MHz); das heißt, der

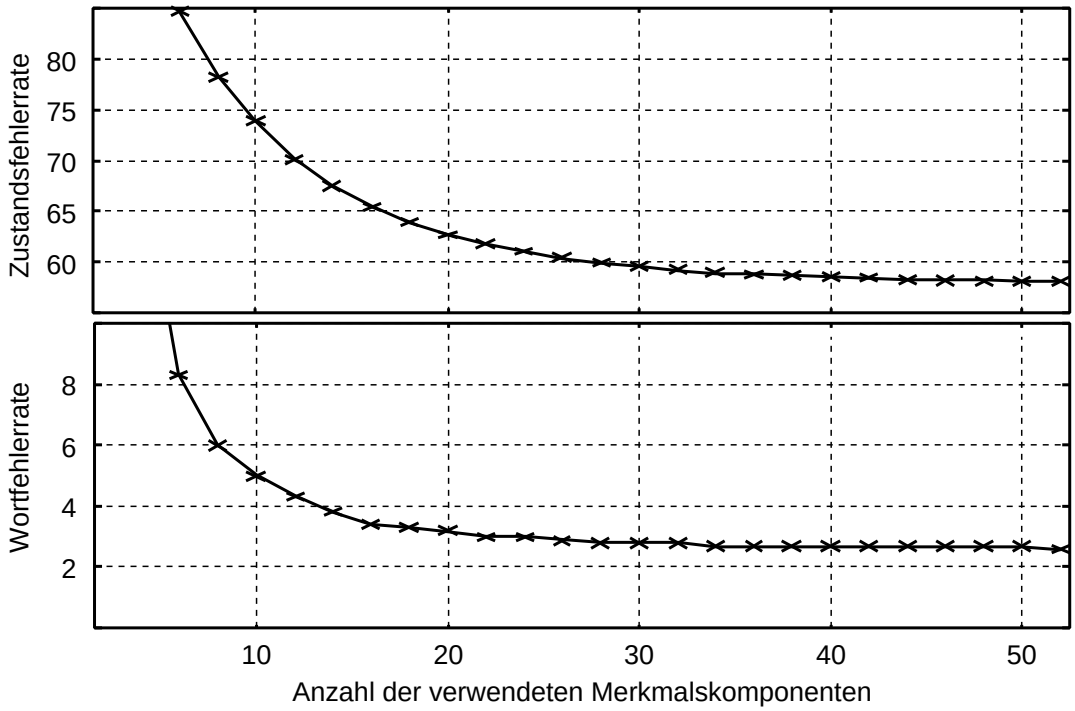

Abbildung 6.15: *Verlauf von Wortfehlerrate und Zustandsfehlerrate in Abhängigkeit der Dimensionalität der Merkmalsvektoren, Trainingsmenge von VM–62*

Zeitaufwand für eine Iteration entspricht 17% der Gesamtdauer des Trainingsmaterials. Bei einer Gesamtzahl von 15 Iterationen entspricht dies einem Realzeitfaktor von 2.57. Zusätzlich werden für dieses Modell die Zustandsfehlerraten auf Trainings– und Testmaterial bestimmt.

Weiterhin wurde ein Modell mit Zielfunktion MZF ausgehend von dem Maximum–Likelihood–Modell trainiert. Hierzu wurden 20 Iterationen von MZF–Training durchgeführt. Als Realzeitfaktor bei 20 Iterationen ergab sich ein Wert von $0.045 \times 20 = 0.9$. Somit fällt in Bezug auf den Rechenaufwand die Bilanz trotz der höheren Anzahl von Iterationen für das MZF–Training positiv aus. In Bezug auf die Fehlerrate ergibt sich jedoch ein völlig anderes Bild. Alle gemessenen Wort– und Zustandsfehlerraten sind in der Tabelle 6.6 zusammengestellt.

Obgleich die Zustandsfehlerraten sowohl auf dem Trainings– als auch auf dem Testmaterial durch das MZF–Training jeweils um mehr als 30% reduziert werden, steigt die Wortfehlerrate auf beiden Stichproben signifikant an. Betrachtet man andererseits die Veränderung der Zustandsfehlerraten vom Maximum–Likelihood Fall zum MWF–Fall, so fällt auf, daß trotz signifikanter Reduktion der Wortfehlerraten ein Anstieg der Zustandsfehlerraten um ca. 10% zu verzeichnen ist. Im Vergleich der Trainingskriterien Maximum–Likelihood, Minimaler Zustandsfehler und Minimaler Wortfehler ergibt sich für das untersuchte Beispiel eine negative Korrelation von Zu-

Trainings–kriterium	Trainingsmaterial		Testmaterial	
	Zustands–fehlerrate	Wort–fehlerrate	Zustands–fehlerrate	Wort-fehlerrate
ML	60.2	2.7	61.1	3.0
MZF	41.5	3.5	42.8	3.5
MWF	64.6	0.8	65.5	1.7

Tabelle 6.6: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML), Minimaler Zustandsfehler (MZF) und Minimaler WortFehler (MWF), Training: Trainingsmenge von VM–62, Test: Testmenge von VM–62*

standsfehlerrate und Wortfehlerrate.

In einem bereits in [Bauer, 1998] veröffentlichten Experiment mit Amerikanischen Einzelziffern ergibt sich ein ähnliches Bild, nur mit dem Unterschied, daß die Wortfehlerrate auf dem Testmaterial nicht ansteigt, aber auch nicht signifikant sinkt.

Macht man den praktischen Nutzen der Modelle an deren Worterkennungsrate fest, was natürlich sinnvoll ist, so scheint der praktische Nutzen von MZF–Training für Anwendungen wie die untersuchten nicht gegeben.

6.13.2 Experimente mit Generalisten–Training

Die Zielsetzung eines Generalisten–Trainings ist die Gewinnung von Modellen, die nicht für einen bestimmten Wortschatz optimiert sind (vgl. Abschnitt 6.12.4). Im Gegensatz zum einem Spezialisten, einem auf kleinem Wortschatz spezialisierten HMM, versteht man unter einem Generalist ein HMM, daß nicht für einen speziellen Wortschatz ausgelegt ist. In diesem Sinne waren alle in den vorangegangenen Abschnitten vorgestellten Hidden–Markov–Modelle Spezialisten. In Telefondialogsystemen werden Generalisten sehr oft verwendet, da an vielen Stellen im Dialog Wörter erkannt werden müssen, für die kein spezielles Trainingsmaterial vorliegt. In den folgenden Experimenten wird als Trainingsmenge für den Generalisten das breit gemischte Sprachmaterial von der SpeechDat–M–Datenbank (SDM) verwendet. Mit dem sich ergebenden Wortschatz von mehreren Tausend Wörtern ist der Rechenzeitbedarf für ein MWF–Training sehr hoch. Damit werden MZF– und MPF–Training interessant, da bei diesen Varianten die Rechenzeit nicht vom Wortschatz abhängt.

MZF–Training Zunächst wird die Eignung des MZF–Kriteriums für diese Trainingsaufgabe untersucht. Ausgangspunkt der Experimente ist ein auf der SDM–Trainingsaufgabe Maximum–

Likelihood optimiertes Modell mit 1888 Gauß'schen Verteilungsdichten. Das entspricht einer mittleren Anzahl von 16 Dichten pro Monophon–Segment. Wie in den vorangegangenen Experimenten wurde eine LDA–basierte Transformation angewandt und der Merkmalsvektor anschließend auf 24 Komponenten reduziert. Auf der Erkennungsaufgabe VM–62 ergibt sich eine Wortfehlerrate von 6.8% für dieses ML–Modell. Die Fehlerrate des Generalisten ist — wie zu erwarten war — erheblich höher als die des Spezialisten.

Das beschriebene ML–Modell wird 15 Iterationen MZF–Trainings unterzogen. Der sich ergebende Realzeitfaktor ist $15 \times 0.048 = 0.72$. Durch das diskriminative Training reduziert sich damit die Zustandsfehlerrate auf dem Trainingsmaterial von 64% auf 50%. Mißt man jedoch die Wortfehlerrate auf dem VM–62 Testset, so ergibt sich eine um 50% von 6.8% auf 10.2% erhöhte Wortfehlerrate. Die Ergebnisse sind in Tabelle 6.7 nochmal zusammengefaßt. Wie schon

Optimierungskriterium	Zustandsfehlerrate (Trainingsmaterial)	Wortfehlerrate (Testmaterial)
ML	64	6.8
MZF	50	10.2

Tabelle 6.7: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML) und Minimaler Zustandsfehler (MZF), Training: Trainingsmenge von SDM, Test: Testmenge von VM–62*

im vergangenen Abschnitt ergibt sich eine negative Korrelation der Wortfehlerrate mit der Zustandsfehlerrate. Ein praktischer Nutzen kann aus dem MZF–Training auch hier nicht gezogen werden.

MPF–Training In einem weiteren Experiment soll nun das Potential von diskriminativem Training mit Zielfunktion Minimaler PhonemFehler (MPF) ausgelotet werden. Ausgangspunkt der Optimierung ist das auch schon für das MZF–Training verwendete ML–Modell. Es werden 15 Iterationen MPF–Training durchgeführt, wobei sich ein Realzeitfaktor von $15 \times 0.036 = 0.54$ ergibt.

Sowohl für das ML–Modell als auch für das MPF–Modell wurden nun verschiedene Phonem– und Wortklassifikationsfehlerraten auf Trainings– und Testset gemessen. Zur Messung der Phonemklassifikationsfehlerrate wurde zum einen eine Einzelphonemklassifikation nach Segmentierung in Einzelphoneme und zum anderen eine kontinuierliche Phonemerkennung durchgeführt. Hierbei wurde statt dem gesamten Trainings– und Testmaterial von SDM eine Untermenge phonetisch reicher Sätze (vgl. Anhang B.3.2) verwendet. Dies dient nur dem Zweck der

Rechenzeitersparniss, sollte aber keinen systematischen Unterschied machen. Daneben wurde die Wortfehlerrate auf dem VM–62 Testmaterial bestimmt. Alle Fehlerraten sind in Tabelle 6.8 zusammenfaßt.

Optimierungskriterium	Phonemfehlerrate				Wortfehlerrate
	KIP		KPE		
	TR	TE	TR	TE	
ML	50.1	50.9	65.2	65.7	6.8
MPF	40.2	42.7	57.7	59.7	7.4

Tabelle 6.8: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML) und Minimaler PhonemFehler (MPF) bei Klassifikation Isolierter Phoneme (KIP), Kontinuierlicher PhonemErkennung (KPE) und Worterkennung. TR: Trainingsmaterial, TE: Testmaterial, Training: Trainingsmenge von SDM bzw. SDM–PR für Phonemerkennung, Test: Testmenge von VM–62 für Worterkennung bzw. SDM–PR für Phonemerkennung.*

Es ergibt sich die durchgängige Tendenz, daß durch das MPF–Training die Phonemfehlerraten sowohl bei der Klassifikation isolierter Phoneme als auch bei der kontinuierlichen Phonemerkennung deutlich reduziert werden. So ergibt sich z.B. eine Reduktion der Phonemfehlerrate auf dem Testmaterial von ca. 11% bei der Klassifikation einzelner Phoneme bzw. 9% bei der kontinuierlichen Phonemerkennung. Trotzdem ergibt sich für die Worterkennung ein signifikant negativer Einfluß des MPF–Trainings: die Wortfehlerrate auf dem VM–62 Testset steigt von 6.8% auf 7.4%. Damit ergibt sich trotz des deutlich positiven Einflusses auf die Phonemerkennung kein praktisch nutzbarer Vorteil durch das MPF–Training.

MWF–Training In einer letzten Versuchsreihe zum Generalisten–Training soll nun das diskriminative Training mit Zielfunktion Minimaler WortFehler (MWF) untersucht werden. Auf dem ML–Modell werden 10 Iterationen von MWF–Training unter Verwendung des gesamten, im Trainingsmaterial auftretenden Wortschatzes durchgeführt. Für eine Iteration ergibt sich ein Realzeitfaktor von 1.02. Bei den durchgeführten 10 Iterationen ergibt dies einen gesamten Realzeitfaktor von 10.2. Die Gesamttrainingsdauer für das MWF–basierte Generalisten–Training beträgt damit ca. 13 Tage. Diese Dauer liegt noch in einem vertretbaren Rahmen für die Entwicklung eines Spracherkennungssystems, wobei der zeitliche Rahmen für solch ein Gesamtprojekt noch deutlich höher anzusetzen wäre.

In Tabelle 6.9 sind die Wortfehlerraten für das ML–Modell und das MWF–Modell zusammengefaßt. Hierbei sind natürlich die Wortschätze für Trainings– und Testmaterial in völlig an-

Optimierungskriterium	Wortfehlerrate	
	Trainingsmaterial	Testmaterial
ML	35.5	6.8
MWF	22.8	6.4

Tabelle 6.9: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML) und Minimaler Wortfehler (MWF), Training: Trainingsmenge von SDM, Test: Testmenge von VM–62*

deren Größenordnungen. Während für das Testmaterial wie bisher der Wortschatz aus den 62 Wörtern besteht, liegt die Größe des Wortschatzes bei der Worterkennung auf dem Trainingsmaterial bei 2701. Daher sind sind die gemessenen Wortfehlerraten stark unterschiedlich; für das ML–Modell: 35.5% auf dem Trainingsmaterial und 6.8% auf dem Testmaterial (vgl. oben). Das MWF–Training reduziert die Wortfehlerrate auf dem Trainingsmaterial deutlich um ca. 35% auf 22.8%. Auf dem Testmaterial wird die Wortfehlerrate lediglich um ca. 6% auf 6.4% reduziert. Diese Reduktion liegt gerade noch in einem signifikanten Bereich.

An dieser Stelle muß angemerkt werden, daß im Trainingsmaterial der SpeechDat–M–Datenbank (SDM) über 5000 Äußerungen mit Wörtern aus dem Wortschatz von VM–62 enthalten sind. Im Vergleich zu der Gesamtzahl von ca. 100000 Wörtern in der gesamten Trainingsmenge von SDM ist das zwar nicht viel, es ist jedoch möglich, daß die Verbesserung auf dem VM–62 Wortschatz einzig in diesen Äußerungen aus dem Zielvokabular begründet ist. Ausgehend von dieser Überlegung soll anhand eines Trainings ganz ohne Wörter aus dem Zielvokabular in der Trainingsmenge das MWF–Training evaluiert werden.

Dieses weitere MWF–basierte Generalisten–Training erfolgt nun unter Verwendung des phonetisch reichen Materials der SpeechDat–M–Datenbank (SDM–PR). Weiterhin wurde der Wortschatz für dieses Training auf die in der Trainingsmenge SDM–PR auftretenden 1700 Wörter begrenzt. So erklärt sich auch die Wortfehlerrate auf dem Trainingsmaterial von 24.1%, die doch deutlich niedriger liegt als bei dem vorangegangenen MWF–Training mit über 2700 Wörtern. Die Wortfehlerrate auf dem VM–62 Testmaterial für das ML–Modell liegt mit 9% weit über der des ML–Modells, das auf dem gesamten SDM Trainingsmaterial traininert wurde. Zumindest für den Maximum–Likelihood Fall scheint sich das Fehlen der Wörter aus dem Zielvokabular stark negativ auf die Erkennungsleistung auszuwirken. Besonders interessant soll aber an dieser Stelle aber nur die Reduktion der Fehlerraten durch das MWF–Training sein. Dabei reduzierte sich die Wortfehlerrate auf dem Testmaterial um ca. 9% auf 8.2% und auf dem Trainingsmaterial wurde sie um ca. 45% auf 12.8% reduziert. Die Erkennungsraten sind in Tabelle 6.10 nochmals zusammengestellt.

Optimierungskriterium	Wortfehlerrate	
	Trainingsmaterial	Testmaterial
ML	24.1	9.0
MWF	12.8	8.2

Tabelle 6.10: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML) und Minimaler Wortfehler (MWF), Training: Trainingsmenge von SDM–PR, Test: Testmenge von VM–62*

Das Ergebnis dieses Experiments zeigt, daß auch bei einem vollkommen nicht–wortschatzspezifischen Generalisten–Training die Worterkennungsleistung durch ein diskriminatives Training mit Zielfunktion Minimaler WortFehler (MWF) signifikant gesteigert werden kann.

Zusammenfassend kann festgestellt werden, daß die Kriterien Minimaler ZustandsFehler (MZF) und Minimaler PhonemFehler (MPF) auch bei einem Generalisten–Training nicht geeignet sind, um die Worterkennungsleistung des diskriminativ nachgeschätzten Modells zu erhöhen. Andererseits konnte durch die Optimierung mit Zielfunktion Minimaler WortFehler (MWF) die Wortfehlerrate des Generalisten reduziert werden. Es muß allerdings festgestellt werden, daß die Reduktion der Fehlerraten mit 9% bzw. 6% deutlich geringer ist als beim wortschatzspezifischen MWF–Training (vgl. z.B. Abschnitt 6.5). Die erzielte Reduktion der Fehlerraten beim diskriminativen Training mit großem Wortschatz ist jedoch vergleichbar mit den in der Literatur zu findenden Werten, die auch bei 10% liegen.

6.13.3 Experimente mit Ziffernketten

Ein in Telefondialogsystemen häufig gebrauchter und wichtiger Wortschatz ist der Ziffernwortschatz. Insbesondere ist die Erkennung verbunden gesprochener Ziffern ([Rabiner u. a., 1988], [Juang und Wilpon, 1994]) ein schneller und komfortabler Eingabemodus für Zahlen. Nicht zuletzt aufgrund der Wichtigkeit der sogenannten Ziffernkettenerkennung finden sich in vielen Sprachdatenbanken wie der SieTill–Datenbank [ELRA–Internetseite, 2000] große Mengen von Ziffernketten. Sowohl die große praktische Bedeutung der Erkennungsaufgabe als auch das Vorhandensein genügend großer Trainingsstichproben spricht für die Optimierung des Ziffernkettensystems mittels diskriminativer Methoden.

Neben dem Trainingskriterium Minimaler WortFehler (MWF), das sich für die Optimierung von Erkennungssystemen geeignet zeigt, bietet sich hier weiterhin das Optimierungskriterium Minimaler WortFolgenFehler (MWFF, vgl. Abschnitt 6.12.4) an. Die größere Rechenzeit für MWFF–Training sollte aufgrund des kleinen Wortschatzes hier nicht der begrenzende Faktor sein.

Ausgangspunkt des Experiments ist ein Maximum–Likelihood optimiertes HMM, das auf den Ziffernketten der SieTill–Datenbank (vgl. Anhang B.2.1) trainiert wurde. Wiederum wurde hierbei die LDA–basierte Transformation eingesetzt und der Merkmalsvektor bis auf 24 Komponenten reduziert. Auf lexikalischer Ebene wurde eine Ganzwortmodellierung mit einer hohen Anzahl von insgesamt 242 Zuständen für die 11 Ziffernwörter eingesetzt. Die Gesamtzahl der eingesetzten Verteilungsdichten wurde zu 2000 festgelegt. Der Realzeitfaktor für das ML–Training mit 5 durchgeführten Iterationen ergibt sich zu 0.06. Auf dem SieTill–ZK Testset ergibt sich für das ML–Modell eine Wortfehlerrate von 4.8%. Weitere Details zu den Erkennungsraten des Modells finden sich in Tabelle 6.11.

Trainings– kriterium	Trainingsmenge				Testmenge			
	WFR	AUSL	EINF	WFFR	WFR	AUSL	EINF	WFFR
ML	3.8	0.9	1.0	9.8	4.8	1.0	1.5	12.2
MWF	2.1	0.9	0.9	5.6	3.7	1.0	1.4	9.5
MWFF	1.8	0.8	0.4	4.1	3.4	1.0	1.0	8.8

Tabelle 6.11: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML), Minimaler WortFehler (MWF) und Minimaler WortFolgenFehler (MWFF). Wortfehlerrate: WFR, Auslöschungen: AUSL, Einfügungen: EINF, Wortfolgenfehlerrate: WFFR. Training: Trainingsmenge von SieTill–ZK, Test: Testmenge von SieTill–ZK*

Zunächst wird eine MWF–basierte Optimierung mittels 15 Iterationen durchgeführt. Dabei bemißt sich der Realzeitfaktor zu 0.15, liegt also um ca. den Faktor 2.5 höher als beim ML–Training. Sowohl auf dem Trainings– als auch auf dem Testmaterial ergibt sich eine deutliche Reduktion der Fehlerraten durch die MWF–Optimierung. So wird z.B. die Wortfehlerrate auf dem Testmaterial um ca. 23% auf 3.7% reduziert. Wortauslöschungen wurden hierbei gar nicht und Worteinfügungen nur um einen Zehntel Prozentpunkt absolut reduziert. Die Wortfolgenfehlerrate sinkt durch das MWF–Training um ca. 22%.

In einem zweiten Experiment wurden 15 Iterationen MWFF–Training durchgeführt. Der gesamte Realzeitfaktor ergibt sich hierbei zu 0.32, was also nochmal um einen Faktor 2 langsamer ist als das MWF–Training. Für das MWFF–Modell ergibt sich eine Wortfehlerrate von 3.4% und eine Satz– oder Wortfolgenfehlerrate von 8.8%. Dies stellt für beide Fehlerraten eine signifikante Reduktion gegenüber den Ergebnissen mit dem MWF–Modell dar. Während wiederum die Wortauslöschungen gegenüber dem ML–Fall nicht reduziert wird, wird die Worteinfügungsrate von 1.5% auf 1.0% deutlich reduziert. Die gleiche Tendenz findet sich für die Erkennungsergebnisse auf dem Trainingsmaterial. Darin zeigt sich die überlegene Wirksamkeit des MWFF–Trainings

gegen Worteinfügungen und Wortauslöschungen, die ja so beim MWF–Training nicht berücksichtigt werden.

Zusammenfassend können sowohl MWF– als auch MWFF–Training als sehr wirksam zur Steigerung der Erkennungsleistung eines Ziffernkettenerkenners eingestuft werden. Dies deckt sich auch mit den in der Literatur zu findenden Ergebnissen ([Chou u. a., 1993], [Schlüter u. a., 1997]). Im direkten Vergleich ist zwar die benötigte Rechenzeit beim MWFF–Training in etwa doppelt so hoch wie beim MWF–Training, die Wort– und Wortfolgenfehlerrate des MWFF–Modells sind jedoch auch signifikant niedriger als die des MWF–Modells. In der Praxis empfiehlt sich die Verwendung des MWFF-Kriteriums, solange es im Rahmen des zur Verfügung stehenden Trainingsmaterials einsetzbar ist. Bei sehr begrenzter Größe der Trainingsstichprobe könnte sich die bessere Ausnutzung durch das MWF–Training positiv auswirken, da dort jede Wortverwechslung und nicht nur jede Satzverwechslung berücksichtigt wird. Hierzu wurden im Rahmen dieser Arbeit jedoch keine experimentellen Untersuchungen unternommen.

6.13.4 Experimente mit Buchstabieren

Ein weiterer in Telefondialogsystemen nützlicher Eingabemodus ist die Erkennung von Buchstabierungen ([Galler und Junqua, 1997]). Insbesondere für den Fall, daß das zu buchstabierende Wort aus eine Liste möglicher Wörter kommt, ist die Fehlerrate bei Eingabe über Buchstabieren sehr viel geringer als bei einer Worterkennung ([Meyer und Hild, 1997], [Kellner u. a., 1998], [Bauer und Junkawitsch, 1999]). Der Wortschatz, der im Rahmen dieser Arbeit für Buchstabieren gewählt wurde, umfaßt nur die ca. 26 Buchstaben und nicht die erweiterte Aussprache über Wörter (*A wie Anton*). Sowohl in der SpeechDat–Datenbank als auch in der SieTill–Datenbank findet sich sehr umfangreiches Sprachmaterial mit diesem Wortschatz. Dort wurden die Buchstaben kontinuierlich gesprochen, wobei die Aufgabe meist in einer Buchstabierung eines Namens bestand.

Im folgenden soll nun untersucht werden, wie diskriminative Verfahren zur Steigerung der Erkennungsleistung bei einem Spracherkennungssystem zur Buchstabenerkennung eingesetzt werden können. Speziell sollen hier das Kriterium Minimaler WortFehler (MWF) und Minimaler WortFolgenFehler (MWFF) untersucht werden. Die Erkennungsleistung wird durch einen kontinuierlichen Erkenner mit dem Buchstabenwortschatz jedoch ohne Sprachmodell evaluiert. Dabei wird die Wort– bzw. Buchstabenfehlerrate (Substitutionen + Auslöschungen + Einfügungen) und die Wortfolgen– bzw. Buchstabenfolgenfehlerrate gemessen.

Zunächst wurde ein nach dem Maximum–Likelihood Prinzip optimiertes Hidden–Markov–Modell erstellt. Dazu wurden die gleichen Einstellungen wie in den vergangenen Abschnitten, also Merkmalsvektor mit 24 Komponenten nach LDA–basierter Transformation, benutzt. Für die lexikalische Modellierung wurde eine Ganzwortmodellierung gewählt. Die Gesamtzahl der

Gauß'schen Dichtefunktionen wurde zu 1500 gewählt. Es sei hier angemerkt, daß sich die in [Bauer und Junkawitsch, 1999] veröffentlichetn Erkennungsraten auf ein Modell mit 3000 Dichten bezogen sind und sich deshalb von denen in dieser Arbeit leicht unterscheiden. Der Realzeitfaktor für 5 Iterationen ML–basiertes Training auf dem Trainingsmaterial der SpeechDat– und SieTill–Datenbank (SDM+SieTill–BU, vgl. Anhang B.5.1) beträgt 0.09. Für die Wortfehlerrate auf dem Testmaterial der SpeechDat–II–Datenbank (SD2–BU, vgl. Anhang B.4.1) ergibt sich ein Wert von 22.2%. Details zu Worteinfügungen und Wortauslöschungen finden sich in der Tabelle 6.12. Da die einzelnen Äußerungen meist aus vielen Buchstaben (8.8 im Mittel) bestehen, ergibt

Trainingskriterium	WFR	AUSL	EINF	WFFR
ML	22.2	1.8	2.9	75.4
MWF	17.3	2.1	2.0	67.2
MWFF	17.2	1.7	1.9	65.4

Tabelle 6.12: *Vergleich von Fehlerraten für die Trainingskriterien Maximum–Likelihood (ML) und Minimaler WortFehler (MWF) und Minimaler WortFolgenFehler (MWFF). Wortfehlerrate: WFR, Auslöschungen: AUSL, Einfügungen: EINF, Wortfolgenfehlerrate: WFFR. Training: SDM+SieTill–BU, Test: SD2–BU*

sich ein sehr hoher Wert von 75.4% für die Wortfolgenfehlerrate.

Das erste angewandte diskriminative Trainingskriterium ist nun das MWF–Kriterium. Für 15 Iterationen ergibt sich ein Realzeitfaktor von 0.53; das heißt, daß das MWF–Training ca. 6 mal soviel Rechenzeit benötigt wie das Maximum–Likelihood–Training. Die Wortfehlerrate auf dem Testmaterial ergibt sich für das MWF–Modell zu 17.3%, was einer Reduktion der Fehlerrate um 22% entspricht. Auch die Wortfolgenfehlerrate wurde um ca. 11% auf 67.2% reduziert. Auffällig ist, daß die Wortauslöschungen nach dem MWF–Training leicht gestiegen und die Worteinfügungen um mehr als 30% zurückgegangen sind. Im Gegensatz zu den Ergebnissen mit Ziffernketten im vorangegangenen Abschnitt ist im Fall der Buchstaben das MWF–Training sehr wohl in der Lage, Wortauslöschungen und Worteinfügungen in der Summe deutlich zu reduzieren. Die Erhöhung der Wortauslöschungen läßt sich damit erklären, daß die Modelle nach dem diskriminativen Training so „scharf" geworden sind, daß sie auch auf die Wörter des Vokabulars nicht immer passen. Diese Schärfe der Modelle erklärt auch die deutlich reduzierte Worteinfügungsrate.

Das zweite für die Buchstabenerkennung eingesetzte Optimierungskriterium ist das Kriterium Minimaler WortFolgenFehler (MWFF). Ausgehend von dem ML–Modell wurden 10 Iterationen mit Zielfunktion MWFF durchgeführt. Als Realzeitfaktor ergibt sich ein Wert von 5.12, womit sich die gesamte Rechenzeit für das MWFF–Training auf dem SDM+SieTill–BU Material

auf ca. 29 Stunden summiert. Trotz des um ca. einen Faktor 10 höheren Rechenzeitbedarfs gegenüber dem MWF–Training bleibt der Aufwand für diese Trainingsaufgabe mit Optimierungsziel MWFF durchaus im Bereich des Möglichen und Sinnvollen. Die Wortfehlerrate auf dem Trainingsmaterial ergibt sich zu 17.2%; ist also nicht signifikant verschieden von der Wortfehlerrate des MWF–Modells. Da ja das Kriterium der Optimierung die Wortfolgenfehlerrate war, ist es nicht sehr verwunderlich, daß sich die Wortfolgenfehlerrate trotz fast identischer Wortfehlerrate gegenüber dem MWF–Modell um 2 Prozentpunkte reduziert. Auch Wortauslöschungen und Worteinfügungsrate sind gebenüber den Ergebnissen nach dem MWF–Training leicht reduziert.

Als Ergebnis der Untersuchungen in diesem Abschnitt läßt sich feststellen, daß die Kriterien Minimaler WortFehler (MWF) und Minimaler WortFolgenFehler (MWFF) gleichermaßen geeignet sind, die Erkennungsleistung eines Buchstabenerkenners zu erhöhen. Die Reduktion der Wort– und Wortfolgenfehlerrate war für beide Kriterien fast gleich bis auf einen minimalen Vorteil des MWFF–Modells bezüglich der Wortfolgenfehlerrate. Da bei der gegebenen Trainingsaufgabe der Rechenaufwand für das MWFF–Training um ca. einen Faktor 10 höher ist, bietet sich in der Praxis an, nur eine MWF–Optimierung zu verwenden.

6.14 A–priori Verwechslungsmatrizen

Für alle Kriterien, bei denen eine N–best Suche für einzelne Einheiten durchgeführt wird, läßt sich die Menge der in der Suche bearbeiteten Klassen leicht einschränken. Für die in dieser Arbeit behandelten Kriterien würden sich hierzu die Kriterien Minimaler Zustandsfehler, Minimaler Phonemfehler und Minimaler Wortfehler, nicht jedoch Minimaler Wortfolgenfehler eignen. Für die kontinuierliche N–best Suche könnte man zwar einzelne Wörter außer acht lassen, dies würde jedoch nur bewirken, daß nur Wortfolgen auftreten, die diese Wörter nicht enthalten. Bei den Suchverfahren für isolierte Zustände, Phoneme und Worte für MZF–, MPF– und MWF–Training hingegen ist es möglich, einige Klassen (bestimmte Zustände, Phoneme oder Wörter) aus dem Suchraum zu entfernen und sie somit für die Klassifikation unberücksichtigt zu lassen.

Die naheliegendste Anwendung für a–priori Verwechslungsmatrizen sind sogenannte Aussprachevarianten. Im Französischen können z.B. viele Worte mit oder ohne dem letztem Vokal im Wort ausgesprochen werden. Im Normalfall sind solche Aussprachevarianten im phonetischen Lexikon durch zwei Einträge, also durch zwei Wörter repräsentiert. Bei der Erkennung ist es dann unerheblich, welche der Aussprachevarianten erkannt wird. Üblicherweise werden vor einer Auswertung des Erkennungsergebnisses alle Aussprachevarianten eines Wortes auf ein gemeinsames Symbol (Wort) abgebildet. Besteht nun die Aufgabe in einer Minimierung der Wortverwechslungen mittels MWF–Training, so macht es natürlich keinen Sinn, die Verwechslungen der Aussprachevarianten untereinander zu minimieren. Ziel von a–priori Verwechslungsmatrizen ist es hier, bei der N–best Suche die Aussprachevarianten des gesprochenen Wortes

auszuschließen. Damit wird ausgeschlossen, daß Verwechslungen mit Aussprachevarianten den Optimierungsprozeß beeinflussen.

Allgemein sei eine a–priori Verwechslungsmatrix $\mathcal{E}$ wie folgt definiert: sei i die korrekte Klasse für ein Muster S, so wird die Klasse j bei der Bestimmung der besten konkurrierenden Klasse (Gleichung (6.6)) nur dann berücksichtigt, wenn das Element der Verwechslungsmatrix $e_{i,j}$ ungleich 0 ist.

Für die Berücksichtigung der oben erwähnten Aussprachevarianten werden zunächst alle Elemente der a–priori Verwechslungsmatrix zu 1 gewählt. Dann werden jedoch Elemente $e_{i,j}$ zu 0 gesetzt, wenn ein Wort w_j eine Aussprachevariante des Wortes w_i ist.

Eine weitere mögliche Anwendung von a–priori Verwechslungsmatrizen ist die Beschleunigung des diskriminativen Trainings durch Einschränkung des Suchraums bei der N–best Suche. Möglich ist dies bei einer Aufteilung der Klassen in solche, die zu der korrekten Klasse verwechselbar bzw. nicht verwechselbar sind. Auf diese Anwendung der a–priori Verwechslungsmatrizen wird in Abschnitt 6.17 noch genauer eingegangen.

6.15 Behandlung von Füllwörtern

Ein für die Praxis sehr störender Effekt in der automatischen Spracherkennung sind Falschalarme bzw. Einfügungen durch nicht–stationäre Geräusche wie Räuspern oder Husten. Während z.B. bei Simulationen die Annahme zulässig ist, daß in einer Äußerung genau ein Wort enthalten ist, ist dies in einer realen Anwendung nicht gegeben. Üblicherweise wird bei einer Anwendung der Erkenner zu einem bestimmten Zeitpunkt angeschaltet und erst wieder abgeschaltet, nachdem ein Wort erkannt wurde. Räuspert sich also ein Benutzer bevor er ein Wort spricht, wird möglichweise das Räuspern auf ein Wort abgebildet und als wahrscheinlich falsches Erkennungsergebnis gewertet, bevor das eigentliche Wort gesprochen wurde. In einer Simulation basierend auf digitalen Sprachaufnahmen in Dateien könnte dieser Fall dadurch abgefangen werden, daß pro Datei nur ein Wort erkannt wird.

Ein möglicher Ansatz zur Vermeidung von solchen Falschalarmen ist die Verwendung von Konfidenzmaßen ([Junkawitsch u. a., 1997], [Junkawitsch, 2000]). Dabei werden im Erkennungsergebnis potentiell schlecht erkannte Wörter entfernt. Besonders bei der kontinuierlichen Erkennung ist die Anwendung von Konfidenzmaßen jedoch schwierig, da die Aussage über die Korrektheit eines Wortes mittels Konfidenzmaß sehr schwierig ist. Beispielsweise führt die Abbildung eines Geräusches und eines Wortes auf nur ein erkanntes Wort zu einem Problem, da das Geräusch das Konfidenzmaß des Wortes stark beeinflußt. Ein anderer — parallel kombinierbarer — Ansatz ist die Verwendung von sogenannten Füllwörtern, auf die nicht–stationäre Geräusche abgebildet werden können. Dabei wird für nicht–stationäre Geräusche ein HMM verwendet, das im Prinzip den Hidden–Markov–Modellen für die Wörter des Wortschatzes gleicht. Das im un-

terscuhten System eingesetzte Füllwort besitzt 12 Zustände.

Beim Maximum–Likelihood–Training können Füllwörter nur trainiert werden, wenn nicht–stationäre Geräusche in der Transkription der Datenbank verschriftet sind, wie das z.B. in der SpeechDat–Datenbank ([SpeechDat–Internetseite, 2000]) der Fall ist. Sind Geräusche verschriftet, so können sie z.B. alle auf ein gemeinsames Füllwort abgebildet und dies genauso wie alle anderen Wörter behandelt werden. Ein Problem tritt auf, wenn in einer Datenbank keine Geräusche verschriftet sind, wie das z.B. in der SieTill–Datenbank der Fall ist. Man wird im allgemeinen so vorgehen, daß man die Füllwörter z.B. auf der SpeechDat–Datenbank trainiert und den eigentlichen Wortschatz auf der anderen Datenbank.

Bei der Verwendung von Füllwörtern tritt häufig das Problem auf, daß das Füllwort entweder zu dominant ist, also häufig anstatt anderer Wörter erkannt wird, oder nur sehr selten erkannt wird. Beim diskriminativen MCE–Training ergeben sich hierfür Lösungsansätze, die in den folgenden Abschnitten beschrieben werden.

6.15.1 Füllwörter für Einzelworterkennung

dominantes Füllwort Werden in der Erkennung sehr viele Wörter aus dem eigentlichen Wortschatz auf das Füllwort abgebildet, so muß das Ziel sein, das Modell für das Füllwort für die anderen Wörter unwahrscheinlicher zu machen. Dies läßt sich im allgemeinen nur mit diskriminativen Methoden realisieren.

Ein mögliches Vorgehen besteht in Verwendung einer a–priori Verwechslungsmatrix für Wörter und einem künstlichen Konstanthalten bestimmter Modellparameter. Die Verwechslungsmatrix ist dabei so zu wählen, daß bei der N–best Suche immer nur das Füllwort berücksichtigt wird. So werden für das diskriminative Training nur Verwechslungen von Wörtern des Wortschatzes mit dem Füllwort berücksichtigt. Dadurch wird eine optimale Ausnutzung des Datenmaterials in Hinsicht auf das spezifische Problem gewährleistet. Als besonderer Vorteil erweist sich hier, daß für diese Art der Nachschätzung der Füllwortparameter keine verschrifteten Geräusche im Trainingsmaterial vorhanden sein müssen.

Das beschriebene Vorgehen sollte als eigener Schritt nach einem Maximum–Likelihood–Training und einem „normalen“ diskriminativen Training ohne Berücksichtigung des Füllworts erfolgen. Insbesondere empfiehlt es sich, nur die Parameter des Füllworts nachzuschätzen, da die Eigenschaften der Modelle für den Wortschatz so nicht negativ beeinflußt werden können.

selten ansprechendes Füllwort Die umgekehrte — in der Praxis aber eher seltener beobachtete — Problematik ergibt sich, wenn ein Maximum–Likelihood trainiertes Füllwort kaum anspricht, sondern Geräusche weiterhin auf Modelle für den eigentlichen Wortschatz abgebildet werden. In einem solchen Fall muß das Ziel darin bestehen, daß die Modelle für den Wortschatz

für Geräusche unwahrscheinlicher gemacht werden.

Für diesen Fall müßte eine a–priori Verwechslungsmatrix so gewählt werden, daß nur eine Verwechslung von Geräuschen mit Wörtern aus dem Wortschatz zugelassen wird. Die Parameter des Füllwortmodells sollten hierbei künstlich konstant gesetzt werden, damit dieses Modell nicht noch unwahrscheinlicher für Geräusche gemacht wird.

Die beiden in diesem Abschnitt vorgeschlagenen Vorgehensweisen sind nur für das Kriterium Minimaler Wortfehler sinnvoll einsetzbar, da nur dort Verwechslungsmatrizen eingesetzt werden können. Im folgenden Abschnitt soll nun ein Ansatz beschrieben werden, der eine gesonderte Behandlung von Füllwörtern beim Kriterium Minimaler Wortfolgenfehler erlaubt.

6.15.2 Füllwörter für Wortkettenerkennung

Wie in Abschnitt 6.14 beschrieben, können Verwechslungsmatrizen nicht ohne weiteres in der kontinuierlichen N–best Suche für das MWFF–Training eingesetzt werden. Somit kommen die im vorangegangenen Abschnitt beschriebenen Methoden für das MWFF–Training nicht in Frage.

Ziel einer Integration von Füllwörtern in das MWFF–Training muß die Anpassung der Verhältnisse beim Training an die bei der Erkennung sein. Für die Erkennung werden Füllwörter genauso behandelt wie die Wörter aus dem Wortschatz; d.h. es wird eine beliebige Abfolge von Wörtern aus dem Wortschatz und Füllwörtern erkannt. Für das MWFF–Training müssen nun sowohl der erzwungene Viterbi–Algorithmus als auch die kontinuierliche N–best Suche angepaßt werden.

Der erzwungene Viterbi–Algorithmus muß dahingehend erweitert werden, daß zwischen den Wörtern beliebige Einfügungen von Füllwörtern zugelassen werden. Dies läßt sich durch ein Sprachmodell lösen, wobei der erzwungene Viterbi–Algorithmus dann durch einen kontinuierlichen Erkenner mit Sprachmodell realisiert werden muß.

Die kontinuierliche N–best Suche (vgl. Abschnitt 3.2) kann durch eine spezielle Erweiterung so modifiziert werden, daß genau die N besten Wortfolgen ohne Berücksichtigung von Füllwörtern gefunden werden. Dies wird dadurch erreicht, daß bei der Abspeicherung der Wort–Historie Füllwörter einfach unberücksichtigt bleiben. Dadurch können Pfade, die sich nur durch unterschiedliche Einfügungen von Füllwörtern unterscheiden, rekombinieren. Für die Weiterverarbeitung der Suchergebnisse ist das Auftreten von Füllwörtern in der Worthistorie ohne Belang. Vielmehr müssen sogar die Füllwörter entfernt werden, um die erkannten Wortfolgen mit der korrekten Wortfolge zu vergleichen. Bei der Aufzeichnung der Historie auf Zustandebene werden durchlaufene Modelle für Füllwörter natürlich berücksichtigt.

Mit dem beschriebenen Algorithmus ist es möglich, Füllwörter für das MWFF–Training optimal zu berücksichtigen, ohne daß eine Verschriftung der Geräusche im Trainingsmaterial notwendig ist.

6.16 Experimente mit spezieller Behandlung von Füllwörtern

In diesem Abschnitt werden Experimente beschrieben, mit denen anhand des Beispiels Ziffernketten (vgl. Abschnitt 6.13.3) die Methoden zur Behandlung von Füllwörtern (Abschnitt 6.15) evaluiert werden.

Zunächst wurde auf einem Teil der Trainingsmenge in der SieTill–Datenbank mittels Maximum–Likelihood–Training ein Füllwort generiert. Zu diesem Zweck wurden durch eine kontinuierliche Erkennung ca. 200 Äußerungen identifiziert, bei denen Worteinfügungen zu Erkennungsfehlern geführt haben (Methode nach [Gretter, 1997]). In diesen Äußerungen wurden anschließend Geräusche verschriftet, so daß dieser Korpus zum Maximum–Likelihood–Training eines Füllworts dienen konnte.

In einem ersten Test wurde den ML–, MWF–, und MWFF–Modellen (vgl. Abschnitt 6.13.3) das Maximum–Likelihood trainierte Geräuschwort hinzugefügt und die veränderte Erkennungsleistung auf dem Testmaterial gemessen. Die Anzahl der Zustände des Hidden–Markov–Modells des Füllworts wurde zu 12 gewählt. Für das ML–Modell und das MWF–Modell ergab sich durch das Maximum–Likelihood–Füllwort schon eine beträchtliche Reduktion der Wortfehlerrate um je 16%. Alle Ergebnisse sind in Tabelle 6.13 zusammengestellt. Für das MWFF–Modell zeigt sich keine Veränderung in der Worterkennungsrate. Für alle drei Modelle ergibt sich eine deutliche Erhöhung der Wortauslöschungen und Reduktion der Worteinfügungen. Im vorliegenden Fall ist also ein dominantes Füllwort aufgetreten, das zu vielen Wortauslöschungen führt.

In einem zweiten Schritt wurde das dominante Füllwort mit den in Abschnitt 6.15.1 beschriebenen Methoden mittels MWF–Training nachgeschätzt. Für das MWF–Grundmodell führt das MWF–Füllwort nur zu einer nicht–signifikanten Reduktion der Wortfehlerrate um 0.1 Prozentpunkt, wobei aber Worteinfügungen und Wortauslöschungen sehr viel ausgeglichener sind als mit dem ML–Füllwort. Für das MWFF–Grundmodell führt das MWF–Füllwort zu einer deutlichen Reduktion der Wortfehlerrate gegenüber der Kombination mit dem ML–Füllwort. Dies ist hauptsächlich auf die Reduktion von Worteinfügungen zurückzuführen.

In einem letzten Experiment wurden das ML–Grundmodell und das ML–Füllwort mit Hilfe des integrierten Verfahrens aus Abschnitt 6.15.2 gemeinsam trainiert. Die erzielte Wortfehlerrate von 2.8% ist die niedrigste Fehlerrate aller getesteten Kombinationen. Auch die Wortfolgenfehlerrate von 7.1% ist noch einmal deutlich niedriger als für die anderen Kombinationen. Auffällig ist das Ungleichgewicht von Wortauslöschungen und Worteinfügungen, die nur etwa halb so häufig auftreten wie Wortauslöschungen. Dieses Ungleichgewicht scheint sich für die minimale Wortfolgenfehlerrate mit Berücksichtigung des Füllworts zu ergeben. Beim MWFF–Training ohne Füllwort war dieses Ungleichgewicht nicht zu beobachten.

Zusammenfassend erweist sich die Verwendung eines Füllworts für die Erkennung von Ziffernketten als sehr vorteilhaft. Auch ohne spezielle (diskriminative) Behandlung des Füllworts

Trainings–kriterium Ziffern	Trainings–kriterium Geräuschwort	WFR	AUSL	EINF	WFFR
ML	—	4.8	0.9	1.0	12.2
ML	ML	4.0	1.1	0.7	10.0
MWF	—	3.7	1.0	1.4	9.5
MWF	ML	3.1	1.6	0.3	7.7
MWF	MWF	3.0	1.1	0.7	7.7
MWFF	—	3.4	1.0	1.0	8.8
MWFF	ML	3.4	1.8	0.2	8.3
MWFF	MWF	3.0	1.1	0.5	7.6
MWFF	MWFF	2.8	1.0	0.5	7.1

Tabelle 6.13: *Vergleich von Fehlerraten bei verschiedenen Trainingskriterien für die Ziffern selbst und das Geräuschwort. Kriterien sind Maximum–Likelihood (ML), Minimaler Wort-Fehler (MWF) und Minimaler WortFolgenFehler (MWFF). Wortfehlerrate: WFR, Auslöschungen: AUSL, Einfügungen: EINF, Wortfolgenfehlerrate: WFFR. Training: Trainingsmenge von SieTill–ZK, Test: Testmenge von SieTill–ZK*

reduziert die Verwendung eines einfachen Füllworts die Wortfehlerrate um bis zu 16%. Durch beide diskriminative Verfahren (MWF– und MWFF–Training) zur Behandlung von Füllwörtern konnte eine weitere Reduktion der Wortfehlerrate erreicht werden. Die integrative Behandlung von eigentlichem Wortschatz und Füllwort im MWFF–Training hat hierbei die niedrigste Wortfehlerrate erzielt, die fast 18% unter der Fehlerrate des MWFF–Modells ohne Füllwort liegt.

6.17 Beschleunigung des Trainingsverfahren

Die Experimente in den vorangegangenen Abschnitten haben gezeigt, daß der Zeitbedarf für diskriminatives Training doch erheblich sein kann. Im Vergleich zum Maximum–Likelihood–Training ergibt sich meist ein um den Faktor 10 erhöhter Rechenzeitbedarf. Insbesondere für das MWF–basierte Generalisten–Training ist der Zeitbedarf durch den großen Wortschatz sehr hoch. In den folgenden zwei Abschnitten werden zwei Verfahren zur Beschleunigung des diskriminativen MCE–Trainings vorgestellt.

6.17.1 Beschleunigung durch Verwechslungsmatrizen

Der rechenzeitintensive Schritt beim MWF–Training ist die N–best Suche. Diese mußte z.B. für das Generalisten–Training in Abschnitt 6.13.2 für 2701 Wörter durchgeführt werden. In der Praxis ist es natürlich so, daß nicht jedes Wort mit allen anderen Wörtern verwechselbar ist. Vielmehr könnte man sich pro Wort auf eine kleinere Liste zu diesem Wort verwechselbarer Wörter beschränken. Ist eine solche Liste bekannt, kann diese über den Mechanismus der Verwechslungsmatrizen leicht in das MWF–Training integriert werden. Im folgenden wird ein Ansatz zur automatischen Generierung eines solchen Liste beschrieben.

Grundidee des Verfahrens ist es, Merkmalsvektorfolgen mit Hilfe der Hidden–Markov–Modelle zu generieren und diejenigen Wörter in die Liste der verwechselbaren aufzunehmen, für die die Modellwahrscheinlichkeit für die synthetisierte Merkmalsfolge relativ hoch ist. Das Struktogramm in Abbildung 6.16 beschreibt den vorgeschlagenen Algorithmus.

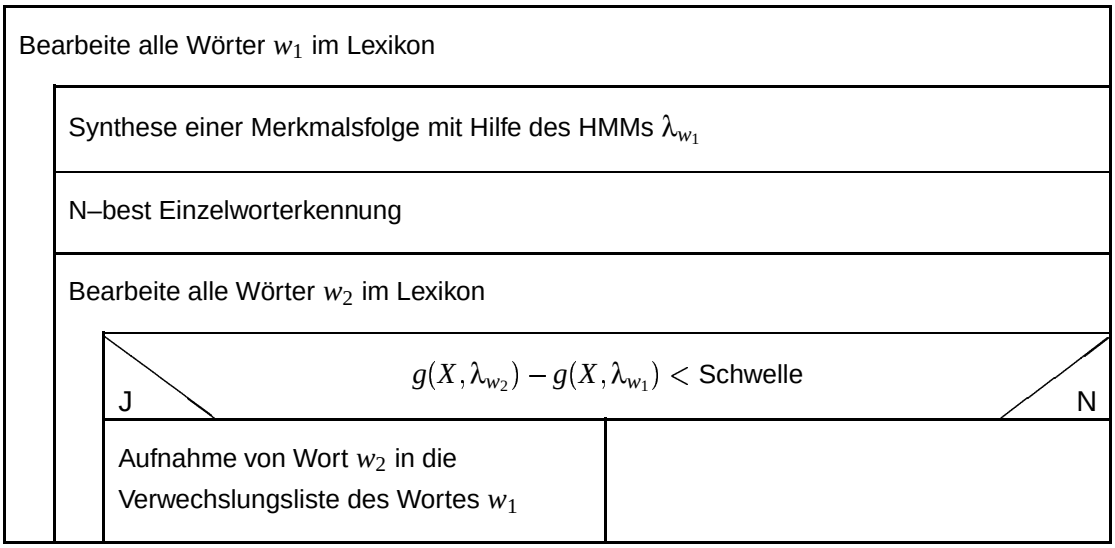

Abbildung 6.16: *Struktogramm für die automatische Generierung von Verwechlungslisten*

Zur Generierung einer Merkmalsfolge aus dem Wortmodell werden einfach Merkmalsvektoren durch gewichtete Mittelung der Mittelpunktsvektoren der Zustände erzeugt. Die Anzahl der Merkmalsvektoren pro Zustand wird durch eine Simulation der Zustandsübergänge entsprechend der Übergangswahrscheinlichkeiten festgelegt.

Die Wahl der Schwelle wurde in den im Abschnitt 6.18 beschriebenen Experimenten sehr konservativ gewählt, so daß die Anzahl der Wörter in den Verwechslungslisten im Mittel ungefähr der halben Wortschatzgröße beträgt.

6.17.2 Beschleunigung durch Reduktion der Trainingsmenge

Die Grundidee des in diesem Abschnitt vorgeschlagenen Verfahrens besteht in der Beschränkung des Trainingsmaterials für alle weiteren Iterationen auf die Muster, die in der ersten Iteration einen Beitrag zur Parameternachschätzung geliefert haben. Wie in Abschnitt 6.8.1 schon ausgeführt, gehen in die Paramerternachschätzung nur solche Muster ein, für die die Gewichtungsfunktion $\frac{\partial l(d(X,\Lambda))}{\partial d(X,\Lambda)}$ mindestens einen festen Faktor mal dem Maximum dieser Funktion (bei $d = 0$) beträgt. Der Faktor wurde wie in Abschnitt 6.8.1 beschrieben zu $1/100$ gewählt.

Die erste Iteration diskriminativen Trainings wird also mit der gesamten Trainingsmenge durchgeführt. Während dieser ersten Iteration wird nun eine Liste mit denjenigen Trainingsmustern erstellt, für die die Gewichtungsfunktion den beschriebenen Schwellwert überschritten hat. Für alle weiteren Iterationen werden nun nur noch die Trainingsmuster aus der reduzierten Liste verwendet.

Das Verfahren bietet sich insbesondere für MWF–Training auf Einzelwörtern und MWFF–Training an. Bei MPF– und MWF–Training auf Wortfolgen ergibt sich meist eine nur wenig reduzierte Trainingsmenge, da meist eine der vielen Muster in einer Äußerung das Schwellwertkriterium erfüllt und somit die Äußerung weiterverwendet werden muß.

6.18 Experimente zur Beschleunigung des Trainingsverfahrens

Beschleunigung durch Verwechslungsmatrix In einem ersten Experiment wird versucht, das MWF–basierte Generalisten–Training mit Wortschatzgröße 2701 (vgl. Abschnitt 6.13.2) mit Hilfe einer Verwechslungsmatrix zu beschleunigen. Dazu wurde mit Hilfe des in Abschnitt 6.17.1 beschriebenen Verfahrens eine Verwechslungsmatrix generiert, die den Suchraum im Mittel um den Faktor 0.5 verkleinert. Für den Realzeitfaktor wurde bei Berücksichtigung der Verwechslungsmatrix der Wert 5.3 gemessen. Durch die Verwechslungsmatrix wurde also eine Beschleunigung um den Faktor 2 erreicht. In der Praxis bedeutet dies eine Verkürzung der Trainingsdauer um ca. 6 Tage. Für die Performance des diskriminativ nachtrainierten Modells ergab sich der gleiche Wert von 6.4% Wortfehlern wie im Fall ohne Verwechslungsmatrix. Weitere Versuche zur Reduktion der Verwechslungslisten wurden nicht unternommen. Es ist jedoch denkbar, daß eine weitere Beschleunigung realisierbar ist.

Beschleunigung durch Reduktion der Trainingsmenge In einem zweiten Versuch wurde eine Reduktion der Trainingsmenge eingesetzt, um das MWF–Training auf dem 62 Wörter umfassenden Wortschatz der Voice–Mail–Datenbank (vgl. z.B. Abschnitt 6.5) zu beschleunigen. Für den Faktor, der den Beitrag zur Optimierung steuert (vgl. Abschnitt 6.8.1), wurde der bereits

erwähnte Wert von $1/100$ verwendet. Nach der ersten Iteration mit der vollen Trainingsmenge ergibt sich eine um ca 60% reduzierte Trainingsmenge für die folgenden Iterationen. Die Anwendung dieser reduzierten Trainingsmenge reduzierte den Rechenzeitbedarf gegenüber der vollen Trainingsmenge um ca 80%. Das gesamte Rechenaufwand für das diskriminative Training hat sich also um etwas weniger als 80% reduziert. Bezüglich der Erkennungsleistung des nachtrainierten Modells konnte keine Veränderung zum Fall der vollen Trainingsmenge gemessen werden.

Zusammenfassung Zusammenfassend können die vorgeschlagenen Methoden zur Reduktion der für das diskriminative Training benötigten Rechenzeit als recht wirksam eingestuft werden. Sowohl für das Training mit kleinem Wortschatz als auch für das Training mit größerem Wortschatz ergab sich eine erhebliche Reduktion der Rechenzeit.

Kapitel 7

Diskussion und Ausblick

7.1 Diskussion

In der vorliegenden Arbeit wurden diskriminative Methoden zur Optimierung der Erkennungsleistung eines automatischen Spracherkennungssystems untersucht. Bei dem betrachtetem System handelt es sich um ein System auf der Basis von Hidden–Markov–Modellen.

Neben dem Standard Maximum–Likelihood–Verfahren zur Parameterschätzung für Hidden–Markov–Modelle ist aus der Gruppe der diskriminativen Verfahren seit Anfang der 90'er Jahre das Minimum Classification Error (MCE) Training bekannt. Im Rahmen der vorliegenden Arbeit wurden aufbauend auf das bekannte MCE–Verfahren algorithmische Erweiterungen und Verbesserungen entwickelt sowie experimentelle Untersuchungen anhand eines praxisnahen Forschungssystem durchgeführt. Die Art der Untersuchungen und algorithmischen Weiterentwicklung war dabei wesentlich an einer praktischen Umsetzbarkeit und Anwendbarkeit orientiert.

Erste grundsätzliche experimentelle Untersuchungen mit dem an die Besonderheiten der verwendeten HMM–Modellierung angepaßten MCE–Trainingsverfahrens zeigten, daß die diskriminative Nachschätzung der Verteilungsmittelpunkte (*Prototypen*) mittels MCE–Training ein vielversprechender Ansatzpunkt zur Steigerung der Erkennungsleistung ist. Eine Anpassung der anderen Systemparameter wie der Mixturkoeffizienten hat sich als nicht entscheidend für die Erkennungsleistung des Systems erwiesen. Die gefundenen Beobachtungen lassen sich durch die reduzierte Komplexität der verwendeten Modellstrukturen begründen, deren wesentlicher und entscheidender Bestandteil nur die Verteilungsmittelpunkte sind.

Eine Analyse der Beziehung zwischen der Wirksamkeit des MCE–Verfahrens und dem Verhältnis zwischen Anzahl freier zu schätzender Parameter und nutzbarem Trainingsmaterial lieferte folgendes Ergebnis: die Wirksamkeit diskriminativer Methoden erweist sich für Systeme mit wenig freien Systemparametern im Vergleich zur vorhandenen Trainingsmenge als maximal. Beispielsweise konnte durch diskriminatives Training für einen Einzelworterkenner eine

Reduktion der Modellgröße um den Faktor 2 bei gleicher Erkennungsleistung ermöglicht werden. Dies ist natürlich für eine reale Anwendung mit beschränktem Speicher und Rechenleistung sehr interessant. Weiterhin ist das MCE–Trainingsverfahren besser als das Standardverfahren Maximum–Likelihood in der Lage, das Potential einer großen Trainingsstichprobe auszunutzen.

Unter dem Aspekt der optimalen Nutzung des Trainingsmaterials wurde eine Methode zur Bestimmung des kritischen Parameters entwickelt, der beim MCE–Training den Einfluß der Trainingsmuster steuert. Durch diese Methode wird ein optimaler Kompromiß zwischen maximaler Ausnutzung der Trainingsstichprobe und Vermeidung von Überadaption gefunden. Für die Optimierung des HMM–Parametersatzes mittels Gradientenverfahren wurde eine Gradientennormierung entwickelt, die in der Praxis einen sehr positiven Einfluß auf das Konvergenzverhalten erzielt. Die neu entwickelten Verfahren erlauben eine praktisch stark vereinfachte Festlegung optimaler Steuergrößen für die Durchführung des diskriminativen Trainings.

Unter Verwendung von verschiedenen Datenbanken und unterschiedlichen Erkennungsaufgaben wurde eine konsistente Analyse bzgl. der Wahl der Klassen für das diskriminative Training durchgeführt. Folgende Definitionen der Klassen für das MCE–Training wurden untersucht: HMM–Zustände, Phoneme, Wörter und Wortfolgen. Die neu eingeführte Wahl der Zustände als Klassen wurde von einem diskriminativen Verfahren zur Merkmalstransformation, der Linearen Diskriminanz–Analyse, übernommen, das in dem untersuchten System mit dem MCE–Training kombiniert wird. Die untersuchten Trainingsaufgaben waren Training für Einzelworterkennung mit kleinem Wortschatz, wortschatzunabhängiges Training mit großem Vokabular, Training für Ziffernkettenerkennung und für Buchstabieren. Grundsätzlich erweist es sich stets als optimal, eine möglichst globale Zielfunktion zu wählen, die der jeweiligen Erkennungsaufgabe möglichst nahe kommt. Obwohl die Wahl von Zuständen und Phonemen Vorteile bezüglich der notwendigen Verarbeitungszeiten bietet, konnte hier keine Reduktion der Wortfehlerrate erreicht werden. Eine Reduktion der Wortfehlerrate wurde nur durch die Wahl von Wörtern und Wortfolgen als MCE–Klassen erzielt. Bei aufgabenspezifischem Training mit kleinem Wortschatz wurde eine Reduktion der Fehlerrate von bis zu 48% erzielt. Wortschatzunabhängiges diskriminatives Training bei größeren Wortschätzen konnte die Wortfehlerrate bis zu 9% reduzieren. Die Wahl von Wortfolgen als MCE–Klassen bedingt sehr hohe Verarbeitungszeiten für das diskriminative Training. Durch den erhöhten Aufwand konnte für die Ziffernkettenerkennung eine signifikante Steigerung der Erkennungsleistung erzielt werden. Für die Erkennung von Buchstaben konnte keine signifikante Verbesserung gegenüber der Wahl von Wörtern als MCE–Klassen beobachtet werden.

Um die rechenzeitintensive Trainingsphase zu verkürzen und das Verfahren somit in der Praxis besser nutzbar zu machen, wurden Methoden zur Beschleunigung des diskriminativen Trainings entwickelt. Bei der ersten Methode wird anhand eines einfachen Kriteriums die Trainingsmenge auf die entscheidenden Stichproben reduziert, wobei die Schrittweitensteuerung der

Gradientensuche speziell darauf abgestimmt ist. Durch diese Methode konnte bei einem Experiment die Rechenzeit für das diskriminative Training um fast 80% reduziert werden. Die zweite entwickelte Methode basiert auf der Verwendung von automatisch generierten a–priori–Wort-Verwechslungsmatrizen.

Das MCE–Trainingsverfahren wurde um die Verwendung von a–priori–Wort-Verwechslungsmatrizen erweitert. Zum einen ist hierdurch eine weitere Beschleunigung der Trainingsphase möglich, zum anderen können hierdurch in realen Trainingsmengen auftretende Effekte wie Aussprachevarianten behandelt werden.

Für das MCE–Trainingsverfahren mit Wortketten als Klassendefinition, das sich insbesondere für die Erkennung von Ziffernketten als vorteilhaft erwiesen hat, wurde ein Verfahren zur konsistenten Behandlung von Füllwörtern, welche nicht–stationäre Geräusche modellieren, entwickelt. Die konsistente Behandlung dieser Geräuschmodelle durch das diskriminative Training hat sich für Erkennung von Ziffernketten als vorteilhaft herausgestellt. Die Verwendung eines Füllworts in Verbindung mit dem speziell abgestimmten diskriminativen Training konnte die Wortfehlerrate auf Ziffernketten nochmals um ca. 18% reduzieren.

7.2 Ausblick

Aufbauend auf das vorgestellte Schema zur Steuerung des Wirksamkeitsparameters (γ) wäre die Implementierung einer vollautomatischen Bestimmung des Parameters denkbar. Ausgehend von der grafischen Darstellung des zu verwendenden Histogramms müßten entsprechende Heuristiken umgesetzt werden.

In Anlehnung an das der Literatur bekannte Verfahren zur Erzeugung neuer Mixturkomponenten könnte für das beschriebene System ein Verfahren zum Reduzieren von Mixturkomponenten entwickelt werden. Das Ziel wäre hierbei eine noch weiter verbesserte Erkennungsleistung bei reduzierter Modellgröße.

Im Rahmen dieser Arbeit wurde das Kriterium Minimaler Wortfolgenfehler (MWFF) nicht für größere Vokabularien (wortschatzunabhängiges Training) untersucht. Um dieses Thema praktisch zu untersuchen, müßten schnelle Methoden zur kontinuierlichen N–best Suche eingesetzt werden. Denkbar wäre auch eine Anwendung des MWFF–Kriteriums auf kurze Abschnitte von sprachlichen Äußerungen. Ähnlich wie bei der Anwendung des MWF–Kriteriums auf Abschnitte mit nur einem Wort könnte das MWFF–Kriterium auf eine kurze Folge von Wörtern angewandt werden. Auf diese Weise könnte möglicherweise eine Reduktion des Rechenaufwands erreicht werden, die eine Echtzeitverarbeitung ohne Sprachmodell erlaubt.

Interessant wäre auch die Verwendung von kontextabhängigen Phonemmodellen und von Modellen mit hoher Zahl an Verteilungsdichten für wortschatzunabhängiges Training. Dadurch könnte untersucht werden, ob das bisher beobachtete hohe Maß an Wirksamkeit der diskrimina-

tiven Methoden auch bei detaillierterer Modellierung gegeben ist.

Anhang A

Nomenklatur

A.1 Variablennamen

a	Zustandsübergangswahrscheinlichkeit
A	Zustandsübergangsstrafe
$\mathcal{A}$	Transformationsmatrix
α	Vorwärtswahrscheinlichkeit
b	Emissionswahrscheinlichkeit
B	Emissionsstrafe
c	Mixturkoeffizient
C	Mixturstrafe
d	Dimensionsindex oder Diskriminanzfunktion
D	Dimensionalität
e	Element einer Verwechslungsmatrix
$\mathcal{E}$	Verwechslungsmatrix
ε	Schrittweite beim Gradientenverfahren
η	Gewichtungskoeffizient bei der Diskriminanzfunktion
g	log–transformierte Wahrscheinlichkeit
γ	Parameter der Sigmoidfunktion
h	Klassenindex
i	Klassenindex
I	Einheitsmatrix
j	Klassenindex
J	Anzahl von (konkurrierenden) Klassen
k	Iterationsindex
K	Anzahl der Iterationen

l	MCE–Zielfunktion
L	Anzahl der Vektoren bei der Supervektor–Bildung
λ	Modell
Λ	Parametersatz
m	Modenindex
M	Modenanzahl
$\vec{\mu}$	Mittelpunktsvektor
N	Anzahl oder (absolute) Häufigkeit
$N(\cdot)$	Normalverteilung (Gauß)
o	allgemeiner Parameter
$\Omega(\cdot)$	(korrekte) Klasse
$P(\cdot)$	Wahrscheinlichkeit
$p(\cdot)$	Wahrscheinlichkeitsdichtefunktion
π	Einsprungswahrscheinlichkeit
Π	Einsprungsstrafe
r	Index eines Musters in der Trainingsmenge
R	Anzahl der Trainingsmuster
s	Zustandsindex
S	Muster
$\mathcal{S}_w$	Intraklassenscattermatrix
$\mathcal{S}_b$	Interklassenscattermatrix
σ	Streuung
Σ	Kovarianzmatrix
t	Zeitindex
T	Anzahl der Vektoren in einer Merkmalsvektorfolge
θ	Zustand in einer Zustandsfolge
Θ	Zustandsfolge
$\mathcal{U}$	Transformationsmatrix zur Dekorrelation
$\mathcal{V}$	Transformationsmatrix zur Maximierung der Klassentrennbarkeit
W	Wortfolge
$\mathcal{W}$	Transformationsmatrix zur whitening–Transformation
X	Folge von Merkmalsvektoren
$\vec{x}$	Merkmalsvektor
$\vec{y}$	nicht–LDA–transformierter Merkmalsvektor
ζ	Hilfsfunktion für die Zuordnung eines Merkmalsvektors zu einer Mode

A.2 Symbole

$(\,)^{-1}$	Inverse einer Matrix
$E\{\cdot\}$	Erwartungswert
$EV\{\cdot\}$	Eigenvektorsystem (Matrix aus Eigenvektoren)
$EW\{\cdot\}$	Eigenwert
$(\,)^{T}$	Transponierte einer Matrix
$tr(\,)$	Spur einer Matrix

Anhang B

Datenbanken und Trainings–/Erkennungsaufgaben

B.1 Datenbank Deutsche Voice–Mail

Die Datenbank *Deutsche Voice–Mail* kann durch die folgenden Attribute beschrieben werden:

Sprache	Deutsch
Abtastrate	8 kHz
Aufnahme–Umgebung	Telefon
Sprechart	Einzelworte

B.1.1 Trainings–/Erkennungsaufgabe VM–62

Die Trainings–/Erkennungsaufgabe *VM–62* kann durch die folgenden Attribute beschrieben werden:

Wortschatz	62 (Ziffern und Kommandos)
Anzahl Trainingssprecher	520
Wörter in der Trainingsmenge	27887
Gesamtdauer der Trainingsmenge	16 h 45 min
Anzahl Testsprecher	261
Wörter in der Testmenge	13600
Gesamtdauer der Testmenge	8 h 10 min

B.2 Datenbank SieTill

Die Datenbank *SieTill* kann durch die folgenden Attribute beschrieben werden:

Sprache	Deutsch
Abtastrate	8 kHz
Aufnahme–Umgebung	Telefon

Siehe auch [ELRA–Internetseite, 2000].

B.2.1 Trainings–/Erkennungsaufgabe SieTill–ZK

Die Trainings–/Erkennungsaufgabe *SieTill–ZK* (ZiffernKetten) kann durch die folgenden Attribute beschrieben werden:

Wortschatz	11 (Ziffern)
Sprechart	kontinuierlich
Anzahl Trainingssprecher	362
Wörter in der Trainingsmenge	42857
Gesamtdauer der Trainingsmenge	11 h 35 min
Anzahl Testsprecher	356
Wörter in der Testmenge	43092
Gesamtdauer der Testmenge	11 h 44 min

B.3 Datenbank Deutsche SpeechDat M

Die Datenbank *SpeechDat M* kann durch die folgenden Attribute beschrieben werden:

Sprache	Deutsch
Abtastrate	8 kHz
Aufnahme–Umgebung	Telefon

Siehe auch [Höge u. a., 1997], [SpeechDat–Internetseite, 2000] und [ELRA–Internetseite, 2000].

B.3.1 Trainingsaufgabe SDM

Die Trainingsaufgabe *SDM* (SpeechDat M) kann durch die folgenden Attribute beschrieben werden:

Wortschatz	2701
Sprechart	Einzelworte und kontinuierlich
Anzahl Trainingssprecher	667
Wörter in der Trainingsmenge	97448
Gesamtdauer der Trainingsmenge	30 h 54 min

Diese Trainingsmenge umfaßt verschiedenartige Äußerungen wie Kommandowörter, Ziffernketten, Buchstabieren, Datumsangaben, Geldbeträge, freigesprochene Zahlen und phonetisch reiche Wörter und Sätze.

B.3.2 Trainings–/Erkennungsaufgabe SDM–PR

Die Trainings–/Erkennungsaufgabe *SDM–PR* (Phonetisch Reiche Sätze) kann durch die folgenden Attribute beschrieben werden:

Wortschatz	1722
Sprechart	kontinuierlich
Anzahl Trainingssprecher	658
Wörter in der Trainingsmenge	34317
Gesamtdauer der Trainingsmenge	10 h 18 min
Anzahl Testsprecher	163
Wörter in der Testmenge	8387
Gesamtdauer der Testmenge	2 h 31 min

B.4 Datenbank Deutsche SpeechDat II

Die Datenbank *SpeechDat II* kann durch die folgenden Attribute beschrieben werden:

Sprache	Deutsch
Abtastrate	8 kHz
Aufnahme–Umgebung	Telefon

Siehe auch [Höge u. a., 1997], [SpeechDat–Internetseite, 2000] und [ELRA–Internetseite, 2000].

B.4.1 Erkennungsaufgabe SD2–BU

Die Erkennungsaufgabe *SD2–BU* (SD2: erste 1000 Sprecher von SpeechDat II, BUchstabieren) kann durch die folgenden Attribute beschrieben werden:

Wortschatz	31 (Buchstaben)
Sprechart	kontinuierlich
Anzahl Testsprecher	819
Wörter in der Testmenge	11090
Gesamtdauer der Testmenge	1 h 28 min

B.5 Mehrere Datenbanken umfassende Aufgaben

B.5.1 Trainingsaufgabe SDM+SieTill–BU

Die Trainingsaufgabe *SDM+SieTill–BU* (SpeechDat M und SieTill, BUchstabieren) kann durch die folgenden Attribute beschrieben werden:

Wortschatz	98 (hauptsächlich Buchstaben)
Sprechart	kontiniuierlich
Anzahl Trainingssprecher	1392
Wörter in der Trainingsmenge	17737
Gesamtdauer der Trainingsmenge	5 h 48 min

Literaturverzeichnis

[Aubert u. a. 1993] Aubert, X. ; Haeb-Umbach, R. ; Ney, H.: Continuous Mixture Densities and Linear Discriminant Analysis for Improved Context Dependent Acoustic Models. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. II, 1993, S. 648–451

[Ayer u. a. 1993] Ayer, C. M. ; Hunt, M. J. ; Brookes, D. M.: A discriminatively derived linear Transform for improved Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1993, S. 583–586

[Bahl u. a. 1988] Bahl, L. R. ; Brown, P. F. ; Souza, P. V. de ; Mercer, R. L.: A New Algorithm for the Estimation of Hidden Markov Model Parameters. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1988, S. 493–496

[Bahl u. a. 1993] Bahl, L. R. ; Brown, P. F. ; Souza, P. V. de ; Mercer, R. L.: Estimating Hidden Markov Models So As To Maximize Speech Recognition Accuracy. In: *IEEE Transactions on Speech and Audio Processing* 1 (1993), Nr. 1, S. 77–83

[Bahl u. a. 1983] Bahl, L. R. ; Jelinek, F. ; Mercer, R. L.: A Maximum Likelihood Approach to Continuous Speech Recognition. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5 (1983), März, S. 179–190

[Bahl und Padmanabhan 1998] Bahl, L. R. ; Padmanabhan, M.: A Discriminant Measure for Model Complexity Adaptation. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1998, S. 453–456

[Bahl u. a. 1986] Bahl, Lalit R. ; Brown, Perter F. ; Souza, Perter V. de ; Mercer, Robert L.: Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1986, S. 49–52

[Bauer 1997] Bauer, Josef G.: Enhanced Control and Estimation of Parameters for a Telephone

Based Isolated Digit Recognizer. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, S. 1531–1534

[Bauer 1998] Bauer, Josef. G.: Application of Discriminative Methods for Isolated Word Recognition. In: Balderjahn, I. (Hrsg.) ; Mathar, R. (Hrsg.) ; Schader, M. (Hrsg.): *Classification, Data Analysis and Data Highways*, Springer, 1998, S. 287–294

[Bauer 2000] Bauer, Josef G.: Triphone Tying Techniques for the Siemens HMM based Automatic Speech Recognizer. In: *Proc. Advances in Speech Technology, International Workshop (to appear)*. Maribor, Slovenia, 2000

[Bauer und Junkawitsch 1999] Bauer, Josef G. ; Junkawitsch, Jochen: Accurate Recognition of City Names with Spelling as a Fall Back Strategy. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 1, 1999, S. 263–266

[Beaufays u. a. 1999] Beaufays, Francoise ; Weintraub, Mitchel ; Konig, Yochai: Discriminative Mixture Weight Estimation for Large Gaussian Mixture Models. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, S. 337–340

[Bellegarda und Nahamoo 1990] Bellegarda, Jerome R. ; Nahamoo, David: Tied Mixture Continuous Parameter Modeling for Speech Recognition. 38 (1990), Dezember, Nr. 12, S. 2033–2045

[Beyerlein 1994] Beyerlein, P.: Fast Log–Likelihood Computation for Mixture Densities in a High Dimensional Feature Space. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1994, S. 22.1–22.4

[Biem und Katagiri 1997] Biem, Alain ; Katagiri, Shigeru: Cepstrum–Based Filter–Bank Design Using Discriminative Feature Extraction at Various Levels. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 2, 1997, S. 1503–1506

[Brown 1987] Brown, P.: *The Acoustic–Modeling Problem in Automatic Speech Recognition*. Pittsburg, PA, Computer Science Department, Carnegie–Mellon University, Dissertation, 1987

[Bub 1999] Bub, Udo: *Anwendungsspezifische Online–Anpassung von Hidden–Markov–Modellen in automatischen Spracherkennungssystemen*. München : Herbert Utz Verlag Wissenschaft, 1999

[Cardin u. a. 1991a] Cardin, Régis ; Normandin, Yves ; Mori, Renato D.: High–Performance Connected Digit Recognition using Maximum Mutual Information Estimation. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1991, S. 533–536

[Cardin u. a. 1991b] Cardin, Régis ; Normandin, Yves ; Mori, Renato D.: An Improved MMIE Training Algorithm for Speaker–Independent Small Vocabulary, Continuous Speech Recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1991, S. 537–540

[Chang und Juang 1992] Chang, Pao-Chung ; Juang, Biing-Hwang: Discriminative Template Training for Dynamic Programming Speech Recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1992, S. I–493–I–496

[Chengalvarayan und Deng 1998] Chengalvarayan, Rathinavelu ; Deng, Li: Speech Trajectory Discrimination using the Minimum Classification Error Learning. In: *IEEE Transactions on Speech and Audio Processing* 6 (1998), Nr. 6, S. 505–515

[Chou u. a. 1992] Chou, W. ; Juang, B.H. ; Lee, C.H.: Segmental GPD Training of HMM Based Speech Recognizer. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1992, S. I–473–I–476

[Chou u. a. 1993] Chou, W. ; Lee, C.-H. ; Juang, B.-H.: Minimum Error Rate Training Based on N–Best String Models. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1993, S. II–652–II–655

[Chou u. a. 1994] Chou, W. ; Lee, C.-H. ; Juang, B.-H.: Minimum Error Rate Training of Inter–Word Context Dependent Acoustic Model Units in Speech Recognition. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1994, S. S09–3.1–S09–3.4

[Chou 1997] Chou, Wu: Minimum Error Rate Training for Designing Tree–Structured Probability Density Function. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 2, 1997, S. 1507–1510

[Duchateau u. a. 1998] Duchateau, Jacques ; Demuynck, Kris ; Compernolle, Dirk V.: Fast and Accurate Acoustic Modeling with semi–continuous HMMs. In: *Speech Communications* (1998), Nr. 24, S. 5–17

[Duda und Hart 1973] Duda, Richard O. ; Hart, Peter E.: *Pattern Classification and Scene Analysis*. New York, Chichester, Brisbane, Toronto, Singapore : John Wiley & Sons, 1973

[Eisele u. a. 1996] Eisele, Thomas ; Haeb-Umbach, Reinhold ; Langmann, Detlev: A Comparative Study of Linear Feature Transformation Techniques for Automatic Speech Recognition. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1996, S. 252–255

[ELRA–Internetseite 2000] : *Internet Seite von ELRA*. 2000. – URL `http://www.icp.grenet.fr/ELRA`

[Euler 1995] Euler, S.: Integrated Optimization of Feature Transformation for Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1995, S. 109–112

[Euler und Zinke 1992] Euler, Stephan ; Zinke, Joachim: Experiments on the Use of the Generalized Probabilistic Descent Method in Speech Recognition. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1992, S. 157–160

[Fukunaga 1972] Fukunaga, Kainosuke: *Introduction to Statistical Pattern Recognition*. Kap. 3, S. 50–59. New York and London : Academic Press, 1972

[Galler und Junqua 1997] Galler, Michael ; Junqua, Jean-Claude: Robustness Improvements in the Continuously Spelled Names over the Telephone. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, S. 1539–1542

[Gandhi und Jacob 1998] Gandhi, Malan B. ; Jacob, John: Natural Number Recognition using MCE Trained Inter–Word Context Dependent Acoustic Models. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1998, S. 457–460

[Gao u. a. 1999] Gao, Yuqing ; Jan, Ea-Ee ; Padmanabhan, Mukund ; Picheny, Michael: HMM Training Based on Quality Measurement. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, S. 129–132

[Gelin-Huet u. a. 1999] Gelin-Huet, Cecile ; Rose, Kenneth ; Rao, Ajit: The Deterministic Annealing Approach for Discriminative Continuous HMM Design. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1999, S. 2717–2720

[Gopalakrishnan u. a. 1989] Gopalakrishnan, P. S. ; Kanevsky, D. ; Nádas, A. ; Nahamoo, D.: A Generalization of the Baum Algorithm to Rational Objective Functions. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1989, S. 631–634

[Gopalakrishnan u. a. 1991] Gopalakrishnan, P. S. ; Kanevsky, Dimitri ; Nádas, Arthur ; Nahamoo, David: An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems. In: *IEEE Transactions on Information Theory* Bd. 37, Januar 1991, S. 107–113

[Gretter 1997] Gretter, Roberto: *persönliche Kommunikation*. 1997

[Haeb-Umbach u. a. 1993] Haeb-Umbach, R. ; Geller, D. ; Ney, H.: Improvements in Connected Digit Recognition using Linear Discriminant Analysis and Mixture Densities. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. II, 1993, S. 239–242

[Haeb-Umbach und Ney 1992] Haeb-Umbach, R. ; Ney, H.: Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 1, 1992, S. 13–16

[Hauenstein 1993] Hauenstein, Alfred: *Optimierung von Algorithmen und Entwurf eines Prozessors für die automatische Spracherkennung*, Lehrstuhl für Integrierte Schaltungen der Technischen Universität München, Dissertation, 1993

[Hauenstein und Marschall 1995] Hauenstein, Alfred ; Marschall, Erwin: Methods for Improved Speech Recognition over Telephone Lines. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 1, 1995, S. 425–428

[Hernando u. a. 1995] Hernando, J. ; Ayarte, J. ; Monte, E.: Optimization of Speech Parameter Weighting for CDHMM Word Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1995, S. 105–108

[Höge 1993] Höge, H.: Statistische Modelle für die Spracherkennung. In: *Proc. DAGA*. Frankfurt am Main, 1993, S. 11–29

[Höge 1990] Höge, Harald: SPICOS II — A Speech Understanding Dialogue System. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1990, S. 1313–1316

[Höge 1999] Höge, Harald: Estimating an upper Bound for the Error Rate for Speech Recognition using Entropy. In: *International Journal of Electronics and Communications* 53 (1999), Nr. 4, S. 205–214

[Höge u. a. 1997] Höge, Harald ; Tropf, Herbert S. ; Winski, Richard ; Heuvel, Henk van den ; Haeb-Umbach, Reinhold ; Choukri, Khalid: European Speech Databases for Telephone Applications. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 3, 1997, S. 1771–1774

[Hojas 1994] Hojas, Kurt: *Lineare Diskriminanzanalyse für die Spracherkennung*, Technische Universität München, Lehrstuhl für Datenverarbeitung, Diplomarbeit, Februar 1994

[Huang u. a. 1990] Huang, X. D. ; Ariki, Y. ; Jack, M. A.: *Hidden Markov Models for Speech Recognition*. Edinburgh : Edinburgh University Press, 1990

[Juang und Wilpon 1994] Juang, B. H. ; Wilpon, J. G.: Recent Technology Developments in Connected Digit Recognition. In: *ICSLP*, 1994, S. 2135–2138

[Juang u. a. 1997] Juang, Biing-Hwang ; Chou, Wu ; Lee, Chin-Hui: Minimum Classification Error Rate Methods for Speech Recognition. In: *IEEE Transactions on Speech and Audio Processing* 5 (1997), Nr. 3, S. 257–265

[Juang und Katagiri 1992] Juang, Biing-Hwang ; Katagiri, Shigeru: Discriminative Learning for Minimum Error Classification. In: *IEEE Transactions on Signal Processing* 40 (1992), Nr. 12, S. 3043–3054

[Junkawitsch 2000] Junkawitsch, Jochen: *Detektion von Schlüsselwörtern in fließender Sprache*. Aachen : Shaker Verlag, 2000

[Junkawitsch und Höge 1998] Junkawitsch, Jochen ; Höge, Harald: Keyword verification considering the correlation of succeeding feature vectors. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 1, 1998, S. 221–224

[Junkawitsch u. a. 1997] Junkawitsch, Jochen ; Ruske, Günther ; Höge, Harald: Efficient methods for detecting keywords in continuous speech. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 1, 1997, S. 259–262

[Kapadia u. a. 1993] Kapadia, S. ; Valtchev, V. ; Young, S. J.: MMI Training for Continuous Phoneme Recognition on the TIMIT Database. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. II, 1993, S. 491–494

[Kellner u. a. 1998] Kellner, Andreas ; Rüber, Bernd ; Schramm, Hauke: Strategies for Name Recognition in Automatic Directory Assistance Systems. In: *IEEE Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA)*. Torino, Italy, September 1998, S. 21–26

[Köhler 2000] Köhler, Joachim: *Erstellung einer statistisch modellierten multilingualen Lautbibliothek für die Spracherkennung*. Aachen : Shaker Verlag, 2000

[L. F. Lamel 1993] L. F. Lamel, J.L. G.: High Performance Speaker–Independent Phone Recognition Using CDHMM. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 1, 1993, S. 121–124

[Lee 1990] Lee, K.: Context Dependent Phonetic Hidden Markov Models for Continuous Speech Recognition. In: *IEEE Trans. on Acoustics, Speech, and Signal Processing* 38 (1990), Nr. 4, S. 599–609

[Liaw und Berger 1998] Liaw, Jim-Shih ; Berger, T. W.: Robust speech recognition with dynamic synapses. In: *Proc. World Congress on Computational Intelligence and IEEE International Joint Conference on Neural Networks* Bd. 3, 1998, S. 2175–2179

[Littel und Höge 1996] Littel, B. ; Höge, H.: Robuste Einzelworterkennung für Telephonanwendungen. In: *Proc. ITG–Fachtagung SPRACHKOMMUNIKATION*. Frankfurt/Main, 1996, S. 65–68

[Littel u. a. 1998] Littel, Bernhard ; Bauer, Josef ; Janke, Siegfried: Speech Recognition for the Siemens EWSD Public Exchange. In: *Proc. IVTTA98*, 1998, S. 175–178

[Ljolje u. a. 1990] Ljolje, A. ; Ephraim, Y. ; Rabiner, L. R.: Estimation of Hidden Markov Model Prameters by Minimizing Empirical Error Rate. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1990, S. 709–712

[McDermott und Karagiri 1997] McDermott, Erik ; Karagiri, Shigeru: String–Level MCE for Continuous Phoneme Recognition. In: *EUROSPEECH*, 1997, S. 123–1236

[Merialdo 1991] Merialdo, Bernhard: Phonetc Recognition using Hidden Markov Models and Maximum Mutual Information Training. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1991, S. 111–114

[Meyer und Hild 1997] Meyer, Michael ; Hild, Hermann: Recognition of Spoken and Spelled Proper Names. In: *EUROSPEECH*, 1997, S. 1579–1582

[Neukirchen und Rigoll 1997] Neukirchen, Christoph ; Rigoll, Gerhard: Advanced Training Methods and New Network Topologies for Hybrid MMI–Connectionist/HMM Speech Recognition System. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, S. 3257–3260

[Ney und Noll 1994] Ney, H. ; Noll, A.: Acoustic–Phonetic Modeling in the SPICOS System. In: *IEEE Transactions on Speech and Audio Processing* 2 (1994), Nr. 2, S. 312–319

[Ney 1984] Ney, Hermann: The Use of a One–Stage Dynamic Programming Algorithm for Connected Word Recognition. In: *IEEE Trans. on Acoustics, Speech, and Signal Processing* 32 (1984), Nr. 2, S. 263–271

[Ney Sommersemester 1995] Ney, Hermann: *Sprachmodellierung und Suche*. Manuskript zur Vorlesung. Sommersemester 1995

[Ney u. a. 1992] Ney, Hermann ; Mergel, Dieter ; Noll, Andreas ; Paeseler, Annedore: Data driven search organization for continuous speech recognition. In: *IEEE Transactions on signal processing* 40 (1992), Februar, Nr. 2, S. 272–281

[Nogueiras-Rodriguez und Marinõ 1998] Nogueiras-Rodriguez, Albino ; Marinõ, José B.: Task Adaptation of Sub–Lexical Unit Models using The Minimum Confusibility Criterion on Task Independent Databases. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1998, S. 2983–2986

[Normandin 1995] Normandin, Yves: Optimal Splitting of HMM Gaussian Mixture Components with MMI Training. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1995, S. 449–452

[Normandin u. a. 1994] Normandin, Yves ; Lacouture, Roxane ; Cardin, Régis: MMIE Training for Large Vocabulary Continuous Speech Recognition. In: *ICSLP*, 1994, S. 1367–1370

[Ortmanns u. a. 1997] Ortmanns, S. ; Ney, H. ; Firzlaff, T.: Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 4, 1997, S. 143–146

[Picone 1990] Picone, Joseph: Continuous Speech Recognition Using Hidden Markov Models. In: *IEEE ASSP magazine* (1990), Juli, S. 26–41

[Plannerer 1995] Plannerer, Bernd: *Erkennung fließender Sprache mit integrierten Suchmethoden*, Lehrstuhl für Datenverarbeitung der Technischen Universität München, Dissertation, 1995

[Povey und Woodland 1999] Povey, D. ; Woodland, P. C.: Frame Discrimination Training of HMMs for Large Vocabulary Speech Recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, S. 333–336

[Rabiner und Juang 1993] Rabiner, L. ; Juang, B.H.: *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993 (Signal Processing Series)

[Rabiner u. a. 1988] Rabiner, L. R. ; Wilpon, J. G. ; Soong, F. K.: High Performance Connected Digits Recognition Using Hidden Markov Models. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 1, 1988, S. 119–122

[Rabiner 1989] Rabiner, Lawrence R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: *Proceedings of the IEEE* 77 (1989), Februar, Nr. 2, S. 257–286

[Rahim u. a. 1997] Rahim, Mazin ; Bengio, Yoshua ; LeCun, Yann: Discriminative Feature and Model Design for Automatic Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1997, S. 75–78

[Rahim u. a. 1995] Rahim, Mazin G. ; Lee, Chin/Hui ; Juang, Biing-Hwang: Discriminative Utterance Verification for Connected Digits Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1995, S. 529–532

[Rainton und Sagayama 1992] Rainton, D. ; Sagayama, S.: Minimum Error Classification Training of HMMs — Implementation Details and Experimental Results / IEICE. 1992. – Forschungsbericht

[Reichl und Ruske 1995] Reichl, W. ; Ruske, G.: Discriminative Training for Continuous Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1995, S. 537–540

[Reichl 1996] Reichl, Wolfgang: *Diskriminative Lernverfahren für die automatische Spracherkennung*. Aachen : Shaker Verlag, 1996

[Robinson 1994] Robinson, A.J.: An Application of Recurrent Nets to Phone Probability Estimation. In: *IEEE transactions on Neural Networks* Bd. 5, 1994, S. 298–305

[Rudolph 1999] Rudolph, Torsten: *Evolutionäre Optimierung schneller Worterkenner*. w.e.b. Universitätsverlag, 1999

[Ruske u. a. 1998] Ruske, G. ; Falterhauser, R. ; Pfau, T.: Extended Linear Discriminant Analysis (ELDA) for Speech Recognition. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)* Bd. III, 1998, S. 1095–1098

[Ruske 1988] Ruske, Günther: *Automatische Spracherkennung. Methoden der Klassifikation und Merkmalsextraktion*. München und Wien : Oldenbourg, 1988

[Schlüter 2000] Schlüter, Ralf: *Investigations on Discriminative Training Criteria*, Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen, Dissertation, 2000

[Schlüter u. a. 1997] Schlüter, Ralf ; Macherey, W. ; Kanathak, S. ; Ney, H. ; Welling, L.: Comparison of Optimization Methods for Discriminative Training Criteria. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1997, S. 15–18

[Schlüter und Macherey 1998] Schlüter, Ralf ; Macherey, Wolfgang: Comparison of Discriminative Training Criteria. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 1, 1998, S. 493–496

[Schlüter u. a. 1999] Schlüter, Ralf ; Macherey, Wolfgang ; Müller, Boris ; Ney, Hermann: A Combined Maximum Mutual Information And Maximum Likelihood Approach for Mixture Density Splitting. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1999, S. 1715–1718

[Schukat-Talamazzini 1995] Schukat-Talamazzini, Ernst G.: *Automatische Spracherkennung. Grundlagen, statistische Modelle und effiziente Algorithmen*. Braunschweig/Wiesbaden : Vieweg Verlag, 1995

[Schwartz u. a. 1985] Schwartz, R. ; Chow, Y. ; Kimball, O. ; Roucos, S. ; Krasner, M. ; Makhoul, J.: Context–Dependent Modeling for Acoustic–Phonetic Recognition of Continuous Speech. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1985, S. 1205–1208

[Schwartz und Austin 1991] Schwartz, Richard ; Austin, Steve: A Comparison of Several Approximate Algorithms for Finding Multiple (N–Best) Sentence Hypotheses. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1991, S. 701–704

[Schwartz und Chow 1990] Schwartz, Richard ; Chow, Yen-Lu: The N–Best Algorithm: An Efficient and Exact Procedure for Finding the N most Likely Sentence Hypotheses. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1990, S. 81–84

[Shimodaira u. a. 1998] Shimodaira, Hiroshi ; Rokui, Jun ; Nakai, Mitsuru: Improving the Generalization Performance of the MCE/GPD Learning. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1998, S. 3277–3280

[SpeechDat–Internetseite 2000] : *SpeechDat Internet Seite*. 2000. – URL `http://www.speechdat.org`

[Steinbiss 1989] Steinbiss, Volker: Sentence–Hypotheses Generation in a Continuous–Speech Recognition System. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 2, 1989, S. 51–54

[Sukkar u. a. 1997] Sukkar, Rafid A. ; Setlur, Anand R. ; Lee, Chin-Hui ; Jacob, Jon: Verifying and Correcting Recognition String Hypotheses using Discriminative Utterance Verification. In: *Speech Communication* (1997), Nr. 22, S. 333–342

[de la Torre u. a. 1997] Torre, Ángel de la ; Peinado, Antonio M. ; Rubio, Antonio J. ; Sánchez, Victoria: A DFE–Based Algorithm for Feature Selection in Speech Recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, S. 1519–1522

[Valtchev u. a. 1997] Valtchev, V. ; Odell, J. J. ; Woodland, P. C. ; Young, S. J.: MMI Training of Large Vocabulary Recognition Systems. In: *Speech Communication* (1997), Nr. 22, S. 303–314

[Warakagoda und Johnsen 1999] Warakagoda, Narada D. ; Johnsen, Magne H.: Neural Network Based Optimal Feature Extraction for ASR. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 1, 1999, S. 97–100

[Warnke u. a. 1999] Warnke, Volker ; Harbeck, Stefan ; Noth, Elmar ; Niemann, Heinrich ; Levit, Michael: Discriminative Estimation of Interpolation Parameters for Language Model Classifiers. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, S. 525–528

[Wellekens 1987] Wellekens, C. J.: Explicit time correlation in Hidden Markov Models for speech recognition. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* Bd. 1, 1987, S. 384–386

[Westendorf 1995] Westendorf, Christian M.: Das Signalanalyseprogramm melfilter v1.1. Beschreibung und Referenzhandbuch / Technische Universität Dresden, Institut für Technische Akustik, AG Sprachkommunikation. Februar 1995. – Forschungsbericht

[Willett u. a. 1999] Willett, Daniel ; Müller, Stefan ; Rigoll, Gerhard: A Discriminative Training Procedure Based on Language Model and Dictionary for LVCSR. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 6, 1999, S. 2757–2760

[Willett u. a. 1997] Willett, Daniel ; Neukirchen, Christoph ; Rottland, Jörg: Dictionary–Based Discriminative HMM Parameter Estimation For Continuous Speech Recognition Systems. In: *ICASPP*, 1997, S. 1515–1518

[Witschel 1993] Witschel, Petra: Constructing Linguistic Orientated Language Models for Large Vocabulary Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 2, 1993, S. 1199–1202

[Wolfertstetter und Ruske 1994] Wolfertstetter, F. ; Ruske, G.: Discriminative State–weighting in Hidden–Markov–Models. In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 1994, S. S07–9.1–9.4

[Wolfertstetter 1997] Wolfertstetter, Franz: *Verallgemeinerte stochastische Modellierung für die automatische Spracherkennung*. Aachen : Shaker Verlag, 1997

[Wu und Guo 1999] Wu, Jian ; Guo, Qing: A Novel Discriminative Method for HMM in Automatic Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)* Bd. 6, 1999, S. 2761–2764

[Young und Woodland 1993] Young, S.J. ; Woodland, P.C.: The Use of State Tying in Continuous Speech Recognition. In: *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 1993, S. 2203–2206

[Ziegenhain u. a. 1998] Ziegenhain, U. ; Harengel, S. ; Kaiser, J. ; Wilhelm, R.: Creating Large Pronunciation Lexica for Speech Applications. In: *Prc. First International Conference on Language Resources and Evaluations (LREC)*, 1998, S. 1039–1043

[Zünkler 1991] Zünkler, Klaus: *Spracherkennung mit Hidden–Markov–Modellen unter Nutzung von unterscheidungsrelevanten Merkmalen*, Lehrstuhl für Datenverarbeitung der Technischen Universität München, Dissertation, 1991

[Zwicker und Fastl 1990] Zwicker, E. ; Fastl, H.: *Psychoacoustics*. Berlin, Heidelberg, New York : Springer Verlag, 1990