Diskriminative Modellkombination in Spracherkennungssystemen mit großem Wortschatz

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Peter Beyerlein

aus Berlin

Berichter:
Universitätsprofessor Prof. Dr.-Ing. Hermann Ney
Universitätsprofessor Prof. Dr.-Ing. Günther Ruske

Tag der mündlichen Prüfung: 25.10.2000

“Everything should be made as simple as possible, but no simpler”

Albert Einstein

Vorwort

Ich widme diese Arbeit meiner Tochter Diana, die im Verlaufe der letzten Jahre viel Geduld mit ihrem Vater zeigen mußte.

Die vorliegende Dissertation entstand während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter an den Philips Forschungslaboratorien Aachen. Ich danke Herrn Professor Dr.-Ing. Hermann Ney am Lehrstuhl für Informatik VI der Rheinisch-Westfälischen Technischen Hochschule Aachen, herzlich für seine Betreuung und die zahlreichen Anregungen, die zu dieser Arbeit beitrugen. Ich bedanke mich bei Herrn Prof. Dr.-Ing. Günther Ruske am Lehrstuhl Mensch-Maschine-Kommunikation der Technischen Universität München für die Erstellung des Zweitgutachtens zu dieser Arbeit.
Ferner danke ich Herrn Dr. Thomas Eisele und Herrn Dr. Martin Oerder für die Möglichkeit, diese Arbeit während meiner Beschäftigung in der Forschungsgruppe Spracherkennung der Philips Forschungslaboratorien Aachen zu erstellen.

Meinhard Ullrich, Dr. Dietrich Klakow, Dr. Jochen Peters und Bernd Rüber danke ich für die Reviews meiner Veröffentlichungen und Laborberichte sowie für die stimulierende intellektuelle Umgebung in der die Arbeit entstehen konnte.

Inhaltsverzeichnis

Teil I

Problemstellung

Kapitel 1

Einleitung

1.1 Automatische Spracherkennung

Die gesprochene Sprache ist ein wichtiges Mittel für die zwischenmenschliche Kommunikation. Die maschinelle Erkennung gesprochener Sprache ist in der wissenschaftlich-phantastischen Literatur, wie z.B. in 'Vorbidden Planet' und 'Star Trek', bereits seit Jahrzehnten eine Selbstverständlichkeit. In der realen Welt versuchen Wissenschaftler an Universitäten und in der Industrieforschung seit Jahrzehnten eine zuverlässige sprachliche Übertragung von Wortfolgen zwischen Mensch und Maschine zu ermöglichen, um das taktile und visuelle Kommunikationsmedium zu entlasten und die hohe Bandbreite des Sprachkanals von bis zu 300 Wörtern pro Minute ausnutzen zu können.
Das maschinelle Spracherkennungsproblem, d.h. die maschinelle Transformation einer als Zeitsignal vorliegenden sprachlichen Äußerung unbekannten Inhalts in die Schriftform, ist bisher nur bruchstückhaft und unbefriedigend gelöst. Warum ist automatische Spracherkennung problematisch ? Diese Frage kann man aus drei Gründen stellen, da erstens die meisten Menschen die Sprachkommunikation als alltägliche Fertigkeit erleben, da es zweitens keine zufriedenstellende Theorie der Sprache gibt und da drittens kein Spracherkennungsprototyp existiert, der eine mit dem Menschen vergleichbare Leistung hervorbringt. Dabei darf jedoch nicht vergessen werden, daß diese Fähigkeit ein mehrjähriges, durch liebevolle Eltern überwachtes und integriertes Training aller Kommunikationsorgane des Menschen im Kindesalter erfordert. Zusätzlich sind dem Kind von Geburt an sowohl optimal angepaßte Sinnesorgane zur Erfassung der Umwelt gegeben als auch die komplexe bisher weitgehend unverstandene 'Maschine', auf der die Lern- und Erkennungsprozesse ablaufen.
Eine an die menschliche Leistung heranreichende technische Realisierung erfordert unter anderem die Lösung folgender Probleme:

1. Der Mensch ist nicht in der Lage, eine sprachliche Äußerung exakt akustisch zu wiederholen. Dieses Problem verschärft sich durch die fehlende Reproduzierbarkeit der akustischen Umgebungsbedingungen.

2. Jeder Mensch hat eine andere Aussprache. Variationen entstehen durch die angeborene Anatomie des Sprechers, den erworbenen Dialekt, Akzente, rhetorische Fähigkeiten etc.

3. In Abhängigkeit von der Situation können die Sprechgeschwindigkteit, die Betonung, die Lautstärke sowie auch der Sprachstil und die Wortwahl variieren.

4. Die Erkennung eines gesprochenen Satzes ist eine hochkomplexe Optimierungsaufgabe. Ist der Sprecher in der Lage, bei einem Vokabular von 10^5 Wörtern einen kontinuierlichen Satz mit einer Länge von 10 Wörtern zu formulieren, so muß der Empfänger des Sprachsignals innerhalb von Sekunden einen von $10^{5 \cdot 10}$ potentiell möglichen Sätzen erkennen.

5. Sprechen und Zuhören erfordern Denken, und es gibt bisher nur rudimentäres Wissen über die Denkprozesse, die dem Interpretieren und Verstehen der gesprochenen Äußerung zugrunde liegen. Darum können Spracherken-

nungssysteme in der Gegenwart die gesprochene Wortfolge nur raten, aber nicht wirklich 'verstehen'.

Durch diese Probleme haben sich im Verlaufe der Spracherkennungsforschung mehrere Systemklassen mit unterschiedlichem Schwierigkeitsgrad ausgeprägt. Die Systemklassen für automatische Spracherkennung können wie folgt untergliedert werden:

(1) Robuste Einzelworterkennung (Vokabulargröße < 100 Wörter)

(2) Dialogsysteme für spontane Äußerungen (Vokabulargröße < 2000 Wörter)

(3) Diktiersysteme für vorgegebene Themenbereiche (5000 bis 64000 Wörter)

(4) Prototypen zur Verschriftung der Sprache in Fernseh-, Radiosendungen und Telefonkonversationen

Die Schwierigkeit der Spracherkennungsaufgabe nimmt von Systemklasse 1 bis zur Systemklasse 4 zu, folglich nimmt die Leistung bestehender Systemlösungen ab. Die Leistung eines Spracherkennungssystems wird im allgemeinen in Form der Wortfehlerrate gemessen. Die Wortfehlerrate ist dabei die relative Häufigkeit von Wortfehlern im erkannten Text bezogen auf die Anzahl der tatsächlich gesprochenen Wörter. Die Wortfehlerrate der besten Systeme in Systemklasse 1 variiert zwischen 0.1% und 3%, die der Systemklassen 2 und 3 zwischen 3% und 10%. Diese 3 Systemklassen wurden in den vergangenen Jahrzehnten von technologisch führenden Unternehmen bis hin zur Marktreife entwickelt. Für die Systemklasse 4 existieren weltweit nur wenige Forschungsprototypen. Sie hat die ausgezeichnete Eigenschaft, daß sie den wesentlichen Teil der Spracherkennungsaufgaben, die bei der Mensch-Maschine Kommunikation auftreten können, abdeckt. Aus diesem Grunde fokussiert sich der experimentelle Teil der vorliegenden Arbeit auf die automatische Spracherkennung von Nachrichten- und Fernsehsendungen.

1.2 Statistische Modellierung

Der bisher erfolgreichste Ansatz für die Lösung der Probleme der automatischen Spracherkennung basiert auf der Wahrscheinlichkeitsrechnung. Er wird auch 'statistischer' Ansatz genannt. Bei der Verwendung des statistischen Ansatzes wird die sprachliche Äußerung als 'Beobachtung' eines stochastischen Prozesses aufgefaßt, die gesprochene Wortfolge wird als 'Klasse' betrachtet, die die 'Beobachtung' emittiert hat. Der statistische Ansatz bedient sich der Bayesschen Entscheidungstheorie, um Erkenner mit minimaler Fehlerrate zu konstruieren [Duda$^+$ 1973]. Entsprechend dieser Theorie muß eine Beobachtung x in die Klasse k eingeordnet werden, wenn k die höchste a-posteriori Wahrscheinlichkeit unter allen Klassen besitzt.
Wegen der in Abschnitt 1.1 beschriebenen Vielgestaltigkeit des Sprachsignals ist es bisher zu kompliziert gewesen, die 'wahre' Wahrscheinlichkeitsverteilung der Sprache zu finden bzw. formal zu erfassen. Sie wird durch eine Modellverteilung approximiert. In Hochleistungsspracherkennungssystemen besteht diese Modellverteilung aus einer log-linearen Kombination eines akustischen Basismodells mit einem Sprachmodell. Jedes der beiden Basismodelle wird mit bekannten Schätzverfahren auf jeweils geeigneten Trainingsdaten optimiert. Anschließend werden beide Basismodelle zu einer optimalen Entscheidungsregel kombiniert. Die optimale Gewichtung des Einflusses der beiden Basismodelle geschieht manuell oder mit heuristischen Optimierungsmethoden. Die verschiedenen Modellierungsideen, mit denen man versucht, die Vielgestaltigkeit der Sprache zu erfassen, lassen sich bei dieser Strukturierung entweder nur in das akustische oder nur in das Sprachmodell integrieren. Beide Basismodelle können dadurch hochkomplex und untrainierbar werden.
In der vorliegenden Arbeit wird eine alternative Herangehensweise untersucht. Zunächst wird die Strukturierung in genau ein akustisches Basismodell und genau ein Sprachmodell aufgegeben. Zusätzliche Modellierungsideen werden in Form zusätzlicher separater Basismodelle erfaßt. Das Spracherkennungssystem besteht nun aus einer log-linearen Kombination beliebiger und beliebig vieler geeigneter Basismodelle. Die Basismodelle haben dadurch eine geringere Komplexität und sind besser trainierbar. Außerdem entsteht ein höherer Freiheitsgrad bei der Balancierung des Einflusses der verschiedenen Modelle. Durch den höheren Freiheitsgrad entsteht andererseits das wichtige und bisher ungelöste Problem, die Gewichtung der Basismodelle bezüglich der Wortfehlerrate des resultierenden Systems zu optimieren.

1.3 Sinn, Zweck und Inhalt dieser Arbeit

Um die optimale Ausnutzung aller gegebenen Basismodelle zu ermöglichen, schlage ich in dieser Arbeit ein Verfahren zur Bestimmung der optimalen Entscheidungsregel für eine Kombination von beliebigen und beliebig vielen Basismodellen vor. Dabei wird eine Beschränkung auf die log-lineare Kombinationsform vorgenommen. Diese Form entsteht durch die Anwendung des Maximum-Entropie- Prinzips. Sie ist insbesondere für die Kombination von unabhängigen Basismodellen geeignet. Als Optimierungskriterium dient die empirische Wortfehlerrate der resultierenden Verteilung auf einer repräsentativen Trainingsdatenmenge. Mit Hilfe der optimalen log-linearen Kombination von Basismodellen, die ich '*Diskriminative Modellkombination*' genannt habe, können beliebig viele geeignete Basismodelle der Sprache unabhängig voneinander strukturiert und trainiert werden. Durch das neue Verfahren wird eine optimale Ausnutzung aller vorhandenen Modelle ermöglicht. Sinn und Zweck dieses Verfahrens ist es, die Forschung nach besseren Modellierungen der menschlichen Sprache zu vereinfachen, indem der aufwendige Prozeß der Systemoptimierung automatisiert wird. Um dieser Hypothese Evidenz zu verleihen, wird an Baseline-Systemen der Systemklassen 3 und 4 der Philips-Spracherkennungsforschung auf zwei international bekannten Aufgaben, der WSJ-Aufgabe und der HUB4-Aufgabe, gezeigt, daß bereits einfache Anwendungen der Diskriminativen Modellkombination zu signifikanten Wortfehlerratenreduktionen führen. Obwohl das neue Verfahren eine vollständige Neubewertung aller bisherigen Modellierungsversuche der Spracherkennungsforschung ermöglicht, beschränkt sich die vorliegende Arbeit in ihrem experimentellen Teil auf die Kombination von Basismodellen des Philips-Spracherkenners auf der Systemebene (akustische Modelle und Sprachmodelle verschiedener Eigenschaften), da weltweit eine zu starke Diversifikation der verwendeten Modelle stattgefunden hat. Deswegen wird im Anschluss an die Darstellung der Theorie die Anwendbarkeit des Verfahrens auf weiteren Gebieten der Automatischen Sprachverarbeitung diskutiert.

Kapitel 2

Das Modellkombinationsproblem

2.1 Statistischer Ansatz für Spracherkennung

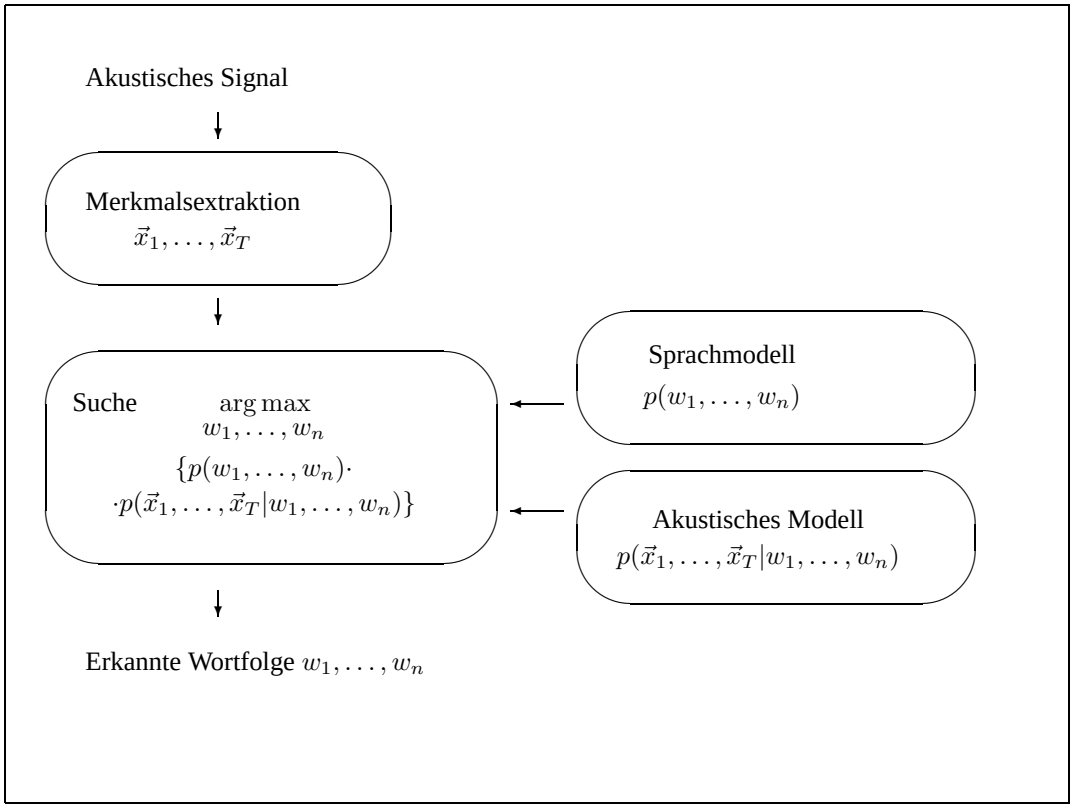

Abbildung 2.1: Aufbau eines auf dem statistischen Ansatz basierenden Spracherkenners

Abbildung 2.1 zeigt ein Blockdiagramm der Architektur des verwendeten Spracherkenners. In einem Vorverarbeitungsschritt - **Merkmalsextraktion** - wird das Sprachsignal digitalisiert und in eine Folge von akustischen Merkmalsvektoren $\vec{x}_1, \ldots, \vec{x}_T$ transformiert. Da das Sprachsignal und somit auch die Folge der Merkmalsvektoren nicht exakt reproduzierbar ist (vgl. Abschnitt 1.1), wird ein statistischer Ansatz gewählt, um seine Generierung

zu modellieren. Entsprechend der statistischen Entscheidungstheorie wird diejenige Wortfolge $w_1, \ldots, w_L$ hypothetisiert, die das Produkt $p(w_1, \ldots, w_L) \cdot p(\vec{x}_1, \ldots, \vec{x}_T | w_1, \ldots, w_L)$ maximiert [Jelinek 1976]. Der erste Faktor, die a-priori Wahrscheinlichkeit der Wortfolge $w_1, \ldots, w_L$, ist unabhängig von der akustischen Beobachtung und wird vollständig durch das **Sprachmodell** bestimmt. Er beinhaltet statistisches Wissen über typische Wortfolgen und ermöglicht somit eine syntaktische und semantische Einschränkung der potentiellen Wortfolgenmenge. Das **akustische Modell** ist im zweiten Faktor enthalten. $p(\vec{x}_1, \ldots, \vec{x}_T | w_1, \ldots, w_L)$ ist die bedingte Wahrscheinlichkeit für die Beobachtung der akustischen Vektorfolge $\vec{x}_1, \ldots, \vec{x}_T$, wenn die Wortfolge $w_1, \ldots, w_L$ gesprochen wurde. Diese Wahrscheinlichkeitsverteilungen werden während des Trainings des Spracherkennungssystems berechnet. Der dem akustischen Modell zugrundeliegende stochastische Prozeß ist ein sogenanntes "Hidden-Markoff-Modell (HMM)". Das Hidden-Markoff-Modell besteht zunächst aus einer Markoffkette [Fisz 1989], wobei jeder Zustand s der Markoffkette einen Teillaut der betrachteten Wortfolge repräsentiert. In jedem Zustand befindet sich eine Emissionsverteilung, die den erwarteten Merkmalsvektor für diesen Teillaut beschreibt. Durch die beim Durchlaufen der HMM-Zustände emittierten Merkmalsvektoren wird eine Lautfolge modelliert, die für die Wortfolge typisch ist. Die Variabilität von Aussprache und Sprechgeschwindigkeit wird durch das Durchlaufen verschiedenster Zustandsfolgen ein und desselben HMM's modelliert.
Für die Erkennung mit großem Vokabular werden die Wörter in Phoneme aufgeteilt, deren Hidden-Markoff-Modelle gut trainierbar sind und die die notwendige Flexibilität für die Variation des Erkennungsvokabulars bieten. Die Phonemfolgen der verschiedenen Wörter werden im **Aussprachelexikon** definiert.
Die Erkennung der gesprochenen Wortfolge $w_1, \ldots, w_L$ wird mit Hilfe eines Optimierungsverfahrens (**Suche**) durchgeführt, welches die Informationen des Sprachmodells, des akustischen Modells und des Aussprachelexikons kombiniert.

2.2 Problematik

Die Modellverteilung, die die wahre Verteilung der Sprache approximieren soll, besteht in der Theorie aus dem Produkt eines akustischen Basismodells, eines Sprachmodells und eines Normierungsterms. Ergebnis dieser Strukturierung sind eine Vereinfachung der Modellannahmen, eine effiziente Ausnutzung der vorhandenen Trainingsdaten und eine Arbeitsteilung in die akustische Modellierung und die Sprachmodellierung.

Die weltweite Forschungsarbeit konzentrierte sich bisher weitgehend auf die Verbesserung des akustischen und des Sprachmodells und auf die Entwicklung einer möglichst zeit- und speichereffizienten Dekodierung der gesprochenen Äußerung.

Während der vergangenen Jahrzehnte hat sich in der Spracherkennungsforschung auf experimentellem Wege herausgestellt, daß die künstliche Balancierung des Einflusses von akustischem Modell und Sprachmodell mittels einer log-linearen Kombination beider Modelle zu besseren Erkennungsergebnissen führt als die aus der Bayesschen Entscheidungsregel abgeleitete Produktformzerlegung. Ursache dafür sind inkorrekte Modellannahmen und die begrenzte Trainingsdatenmenge. Da die optimale Balancierung zweier Modelle mit heuristischen Optimierungsverfahren schnell zum Erfolg führt, wurde diesem Widerspruch zwischen Theorie und Experiment bisher wenig Beachtung geschenkt.

In der vorliegenden Arbeit wird aus der beschriebenen experimentellen Beobachtung ein neuer theoretischer Ansatz abgeleitet, der zum einen die empirische Wortfehlerrate der log-linearen Kombination beider Modelle auf einer Trainingsstichprobe optimiert und zum anderen die optimale Kombination von mehr als nur zwei Modellen erlaubt.

Dieser Ansatz wird im folgenden mit **Diskriminative Modellkombination (DMC)** bezeichnet. Er besteht aus zwei Elementen:

- der log-linearen Kombination einer beliebigen Menge von geeigneten Modellen und
- der Optimierung der empirischen Wortfehlerrate dieser Kombination auf einer Trainingsstichprobe.

In dieser Arbeit soll gezeigt werden, daß die Approximation der wahren Wahrscheinlichkeitsverteilung der Sprache mit Hilfe einer log-linearen Kombination beliebiger und beliebig vieler geeigneter Modelle in einem Spracherken-

nungssytem für großes Vokabular zu besseren Erkennungsergebnissen führt als die ursprüngliche Aufteilung in genau ein akustisches und genau ein Sprachmodell. Dabei wird auf folgende Punkte eingegangen:

- Optimierungskriterium der Kombination,
- Art der kombinierten Modelle,
- Form der Modellkombination,
- Hierarchie-Ebene der Kombination.

2.3 Stand der Wissenschaft

2.3.1 Optimierung der Wortfehlerrate

Das Kriterium, an welchem der Erfolg des Spracherkennungssystems gemessen wird, ist die Wortfehlerrate. Die Wortfehlerrate ist die relative Häufigkeit von eingefügten, eliminierten oder substituierten Wörtern. Sie wird durch einen Vergleich der tatsächlich gesprochenen Äußerung mit der erkannten Äußerung bestimmt.

Die Aufteilung der Modellverteilung des Spracherkenners in das akustische Modell und das Sprachmodell gestattet eine Konzentration auf die Optimierung des jeweiligen Modells. Dabei werden sowohl in der akustischen als auch in der Sprachmodellierung weitgehend Maximum-Likelihood-Schätzverfahren eingesetzt [Duda$^+$ 1973], [Jelinek 1976], [McLachlan$^+$ 1997].

In der Sprachmodellierung wurde in den vergangenen Jahren auch das Maximum-Entropie-Prinzip angewendet [Jelinek 1995], welches zu einer log-linearen Verteilungsform führt. Die freien Parameter des Maximum-Entropie-Sprachmodells werden im wesentlichen mit Maximum-Likelihood-Verfahren geschätzt.
Ein weit verbreitetes Verfahren zur Bestimmung der Parameter der log-linearen Verteilung wird in [Darroch$^+$ 1972] beschrieben.

Um das akustische Modell besser an die Erkennungsaufgabe anzupassen, wurden in den letzten Jahren diskriminative Trainingsmethoden entwickelt. Hier werden die zuvor mit Maximum-Likelihood-Methoden bestimmten initialen Parameter so modifiziert, daß die Satzfehlerrate (relative Häufigkeit fehlerhaft erkannter Sätze) des resultierenden Klassifikators auf einer Trainingsstichprobe möglichst gering ausfällt [Juang$^+$ 1992].

An diesem Punkt stellt sich die Frage nach einem gemeinsamen Verfahren, welches unabhängig von der Art des betrachteten Modells sowohl die Idee von Maximum-Entropie als auch die Idee des diskriminativen Trainings berücksichtigt. In der vorliegenden Arbeit wird ein solches Verfahren vorgestellt. Dieses Verfahren ist theoretisch auf alle hierarchischen Ebenen des Spracherkennungssystems anwendbar. Um dies zu verdeutlichen, werden Experimente zur log-linearen Kombination von mehreren 'vortrainierten' akustischen Modellen und Sprachmodellen analysiert. Trotz der geringen Anzahl der hier zu optimierenden Parameter konnten bei den Experimenten signifikante Fehlerratenreduktionen beobachtet werden. Ausserdem wird theoretisch gezeigt, wie das Verfahren für die Schätzung der deutlich größeren Menge von Verteilungsparametern des akustischen Modells angewendet werden kann. Eine wichtige Eigenschaft dieses Verfahrens ist die direkte numerische Optimierung der Wortfehlerrate des resultierenden Systems.

Da das Optimierungskriterium Wortfehlerrate auf einer Zählung der Fehlklassifikationen des Spracherkenners beruht, ist es nicht differenzierbar. Das hat zu der Definition einer Reihe von Ersatzkriterien geführt, deren Optimierung mit numerischen Verfahren zu einem optimalen Spracherkenner führen kann [Fukunaga 1990], [Juang$^+$ 1995], [Ney 1995].

Das erste Ersatzkriterium ist die geglättete empirische Klassifikationsfehlerrate, die beim sogenannten 'Minimum Classification Error (MCE)' Training in [Juang$^+$ 1995] durch ein numerisches Gradientenverfahren minimiert wird. Dieses Verfahren optimiert jedoch die Satzfehlerrate und nicht die Wortfehlerrate. Es ist für das diskriminative Training von Spracherkennungssystemen mit großem Vokabular und längeren Sätzen nicht geeignet.

Ein anderes Ersatzkriterium ist das MMI-Kriterium ('Maximum Mutual Information') [Brown 1987], das in aktuellen Arbeiten für die akustische Modellierung bei kleinem Vokabular [Normandin 1995] und für die akustische Modellierung bei großem Vokabular [Valtchev 1995] erfolgreich eingesetzt wurde. Der Nachteil dieses Kriteriums besteht darin, daß es nicht direkt die Wortfehlerrate des Erkenners beinhaltet.

Schließlich liefert die Theorie der linearen Klassifikatoren in [Anderson$^+$ 1962] bzw. in [Fukunaga 1990] Methoden zur optimalen Klassifikation von mehrdimensionalen Merkmalsvektoren. Wie in der vorliegenden Arbeit noch gezeigt wird, können die Modellbewertungen in Merkmalsvektoren umgewandelt werden, wodurch diese Theorie für die vorliegende Arbeit relevant wird. Unabhängig davon unterliegt auch diese Theorie einigen Nachteilen. So wird hier vorausgesetzt, daß die Linearkombination der Komponenten des gebildeten Merkmalsvektors gaußverteilt ist. Außerdem wird auch hier die Satzfehlerrate optimiert und nicht die Wortfehlerrate.

Faßt man die beschriebene Situation zusammen, so wird deutlich, daß für verschiedene Spracherkennungssysteme und für verschiedene Komponenten eines einzigen Spracherkenners unterschiedliche Optimierungskriterien vorgeschlagen und eingesetzt werden. Diese Kriterien approximieren mehr oder weniger gut die Wortfehlerrate des resultierenden Klassifikators. Diskriminative Methoden werden bisher hauptsächlich für die Optimierung der Parameter der Emissionsverteilungen der HMM-Zustände des akustischen Modells eingesetzt. Über laufende Untersuchungen zur optimalen Einstellung des Sprachmodellfaktors berichten [Jelinek 1997] und [Wessel 1999].

Wie sich die oben beschriebenen Optimierungsverfahren bezüglich der Kombination von Modellen beliebiger Eigenschaften verhalten werden, ist bisher ungeklärt.

2.3.2 Datengetriebene Modellkombination

Die Wichtigkeit der Balancierung des akustischen Modells und des Sprachmodells wurde bereits in [Bahl$^+$ 1980] und in [Lee 1989] herausgestellt. Bisher wird diese Balancierung im wesentlichen mit manuellen Methoden vollzogen. Ein erster Ansatz für eine automatisierte wortabhängige Balancierung von akustischem Modell und Sprachmodell ist in [Huang$^+$ 1993] in der sogenannten 'Unified Stochastic Engine' zu finden. Hier wird ein wortspezifischer Sprachmodellfaktor eingeführt, welcher bezüglich einer geglätteten Satzfehlerrate optimiert wird. Die traditionelle log-lineare Dekomposition des Klassifikators in genau ein akustisches Modell und genau ein Sprachmodell wird beibehalten, während die vorliegende Arbeit eine log-lineare Dekomposition in beliebig viele verschiedene akustische Modelle und Sprachmodelle vorschlägt.

Mehrere Sprachmodelle verschiedener Eigenschaften (z.B. unterschiedlicher Kontextlänge) werden üblicherweise linear oder über den Maximum-Entropie-Ansatz [Jelinek 1995] log-linear kombiniert, wobei bei der Bestimmung der Parameter die Maximum-Likelihood-Schätzung verwendet wird und keine direkte Optimierung bezüglich der Wortfehlerrate durchgeführt wird. Aktuelle Ergebnisse über die Kombination von Lücken-Sprachmodellen mit unterschiedlicher Kontextlänge werden in [Klakow 1998b] diskutiert.

Bei der Einführung von akustischen Modellen mit langreichweitigem Kontext, wie z.B. von wortübergreifenden Triphonen oder Pentaphonen, entsteht im Philips-System für großes Vokabular das Problem, daß solche Modelle nur mit Hilfe eines $\mathcal{N}$-Best-Verfahrens mit kleinem $\mathcal{N}$ ($\mathcal{N} = 5 \ldots 30$) erfolgreich eingesetzt werden können [Beyerlein$^+$ 1997a]. Dieses Problem wurde auch in anderen Systemen, wie z.B. in [Digalakakis$^+$ 1995], beobachtet. Im Gegensatz dazu veröffentlichten andere Spracherkennungsgruppen mit Hochleistungssystemen die Fähigkeit, solche komplexen Modelle bereits in der ersten Stufe ihrer Systeme einsetzen zu können. Das Problem wird zur Zeit aktiv in den Philips-Forschungslaboratorien bearbeitet. Die Frage, ob die log-lineare Interpolation langreichweitiger akustischer Modelle mit robust geschätzten Modellen kürzeren Kontexts dieses Problem entschärft, ist bisher für das Philips-System ungeklärt.

Faßt man die genannten Punkte zusammen, so kristallisieren sich folgende Fragestellungen heraus:

- In welcher Form sollte eine Menge von Verteilungsmodellen kombiniert werden ?
- Kann ein modellunabhängiges Schema verwendet werden, kann auf Wissen über die Herkunft (akustisches

Modell, Sprachmodell) der Modelle verzichtet werden ?

- Läßt sich die Modellkombination bezüglich der Wortfehlerrate des Spracherkennungssystems optimieren ?

2.3.3 Form der Modellkombination

Wenn kein Wissen über die geeignete Form der Modellkombination vorliegt, oder wenn die zu kombinierenden Modelle unabhängig voneinander sind, dann liegt die Anwendung der log-linearen Kombinationsform nahe. Unabhängig davon wird diese Kombinationsform auch durch das Maximum-Entropie-Prinzip motiviert, wenn kein Wissen über die Eigenschaften der zu kombinierenden Modelle vorliegt.
Diese Kombinationsform ist der wesentliche Leitfaden der vorliegenden Arbeit.

Parallel zu dem in dieser Arbeit beschriebenen Verfahren wurde ROVER ('Recognizer Output Voting Error Reduction') veröffentlicht [Fiscus 1997]. Dieses Verfahren ermöglicht die Kombination der erkannten Worthypothesen verschiedener Spracherkenner mittels eines einfachen Mehrheitsvotingverfahrens. Durch Hinzufügen von Konfidenzinformationen zu den Worthypothesen kann ROVER als vereinfachter Spezialfall von DMC interpretiert werden. Unter anderem ist ROVER auf die Kombination von vollständigen Spracherkennungssystemen spezialisiert; eine Kombination beliebiger Teilmodelle von Spracherkennungssystemen ist mit diesem Verfahren wenig sinnvoll.

In aktuellen Hochleistungssystemen wie [Woodland$^+$ 1998] , [Gauvain$^+$ 1998], [Chen$^+$ 1998], [Kubala$^+$ 1998], [Sankar$^+$ 1998], [Wegmann$^+$ 1998], [Beyerlein$^+$ 1998a] und [Seymore$^+$ 1998] werden seit Jahren erfolgreich lineare Kombinationen von Gauß- oder Laplaceverteilungen in Mischverteilungen verwendet, um die unbekannte Form der Emissionsverteilungen der HMM-Zustände zu approximieren.
Bei der Kombination von Lücken-Sprachmodellen wurde in [Klakow 1998b] gegenteilig gezeigt, daß die log-lineare Kombinationsform zu besseren Erkennungsergebnissen führt als die lineare Kombinationsform.

Faßt man die gemachten Aussagen zusammen, so ergibt sich ein eher unklares Bild, ob es eine modellunabhängige Kombinationsform gibt und welche der genannten Kombinationsformen von Modellen optimal ist.

2.3.4 Baseline-System

In der vorliegenden Arbeit wurde der Hochleistungsspracherkenner der Philips Forschungslaboratorien Aachen verwendet. Dieser Erkenner wurde mehrfach für das weltweite Benchmarking auf Radio- und Fernsehsprachdaten (HUB4) eingesetzt. Das Philips-HUB4-System wurde aufbauend auf dem Philips-WSJ-System [Aubert$^+$ 1994] und dem Philips-NAB-System [Dugast$^+$ 1995a] in den Jahren 1995 bis 1997 entwickelt.

2.4 Zielstellung der Arbeit

Zielstellung der Arbeit ist die Entwicklung eines Verfahrens zur Optimierung der Kombination von Basismodellen (akustische Modelle, Sprachmodelle) eines Spracherkennungssystems. Es soll gezeigt werden, daß durch die Zusammenführung der log-linearen Kombinationsform mit diskriminativen Trainingsmethoden ein modellunabhängiges Schema für die Kombination beliebiger und beliebig vieler Modelle abgeleitet werden kann.

Dabei soll die konventionelle Trennung zwischen der optimalen Einstellung des akustischen Gesamtmodells und der optimalen Einstellung des Sprachmodells aufgehoben werden. Die Gewichte aller kombinierten akustischen Basismodelle und Sprachmodelle sollen gemeinsam mit ein und demselben Kriterium optimiert werden. Das Optimierungskriterium soll dabei die Wortfehlerrate des Erkenners möglichst genau approximieren.

Die vorliegende Arbeit geht davon aus, daß das Kriterium Wortfehlerrate optimal für die Leistungsbewertung eines Spracherkennungssystems ist. Unabhängig davon sollen ausgewählte Kriterien, die die Wortfehlerrate des Systems mehr oder weniger gut approximieren, theoretisch diskutiert werden.

Die Arbeit soll zeigen, daß die durch die Anwendung des Maximum-Entropie Prinzips naheliegende log-lineare

Kombinationsform geeigneter ist als die lineare Kombination oder das sich auf der Ebene des erkannten Textes anbietende einfache Votingverfahren.

Es soll gezeigt werden, daß das abgeleitete Modellkombinationsverfahren unabhängig von der Ebene der Modellkombination und der Art der Modelle ist. Dazu soll beispielhaft ein Verfahren für die optimale log-lineare Sprachmodellinterpolation, ein Verfahren für die optimale Balacierung von Übergangsverteilung und Emissionsverteilung des HMM's und ein Verfahren zur Schätzung der Parameter von Gaußverteilungen in den HMM-Zuständen skizziert werden.

Um die gesamte Zielstellung zu erreichen, sind zusätzlich zur Ableitung der Theorie der Diskriminativen Modellkombination folgende experimentelle Aufgaben zu lösen:

- Es ist zu zeigen, daß mit DMC der sogenannte Sprachmodellfaktor optimal eingestellt wird. Außerdem ist zu zeigen, daß der Einsatz von mehr als zwei Modellen zu einer besseren Erkennungsleistung führt als die Kombination zwischen dem besten akustischen und dem besten Sprachmodell.

- Das $\mathcal{N}$-Best-Problem beim Einsatz von akustischen Modellen mit langreichweitigem Kontext soll durch die log-lineare Kombination mit robusten Modellen entschärft werden. Dazu soll die Arbeit zeigen, daß die log-lineare Interpolation von wortinternen Triphonmodellen mit wortübergreifenden Triphonmodellen und Pentaphonmodellen die Wortfehlerrate des besten Triphon-Systems signifikant senkt, ohne daß ein $\mathcal{N}$-Best-Verfahren angewendet werden muß. Ein weiteres Ziel der vorliegenden Arbeit ist es, zu zeigen, daß auch die log-lineare Interpolation von Sprachmodellen die Wortfehlerrate des Erkenners senkt. Zusammenfassend soll die Erkennunsleistung des Philips-HUB4-Systems mit Hilfe des Modellkombinationsverfahrens verbessert werden.

- Die Leistung der linearen Kombination ist mit der Leistung der log-linearen Kombination zu vergleichen. Auf der Ebene der erkannten Worthypothesen von verschiedenen Erkennungssystemen ist die Leistung der Diskriminativen Modellkombination mit dem einfachen Votingverfahren (ROVER) zu vergleichen.

2.5 Gliederung der Arbeit

Die Arbeit ist wie folgt aufgebaut (siehe Abbildung 2.2): Kapitel 1 enthält eine Einleitung in die automatische Spracherkennung.
Kapitel 2 beinhaltet eine Darstellung des in der vorliegenden Arbeit behandelten Modellkombinationsproblems.
Dabei werden die Problematik, der Stand der Wissenschaft und die Zielstellung der Arbeit beschrieben. Kapitel 3 behandelt die für das Verständnis der Arbeit wesentlichen Elemente der statistischen Modellierung, insbesondere die Strukturierung in Basismodelle. Kapitel 4 beschäftigt sich mit der Theorie der 'Diskriminativen Modellkombination (DMC)', die den Schwerpunkt der vorliegenden Arbeit darstellt.
Kapitel 5 zeigt die Relation von DMC zu anderen naheliegenden Verfahren, d.h. zur Theorie der linearen Klassifikatoren und zur sogenannten 'Unified Stochastic Engine'.
Kapitel 6 und 7 beschreiben die experimentelle Validierung des DMC Verfahrens auf weltweit verwendeten Datenbasen. Kapitel 8 diskutiert alternative Kombinationsverfahren für die Modelle. Kapitel 9 beschreibt schließlich weitere Modellkombinations-Aufgaben im Bereich der Sprachkommunikationsforschung, die mit DMC gelöst werden können. Kapitel 10 enthält die Zusammenfassung der Ergebnisse sowie den Beitrag der vorliegenden Arbeit zur Wissenschaft.

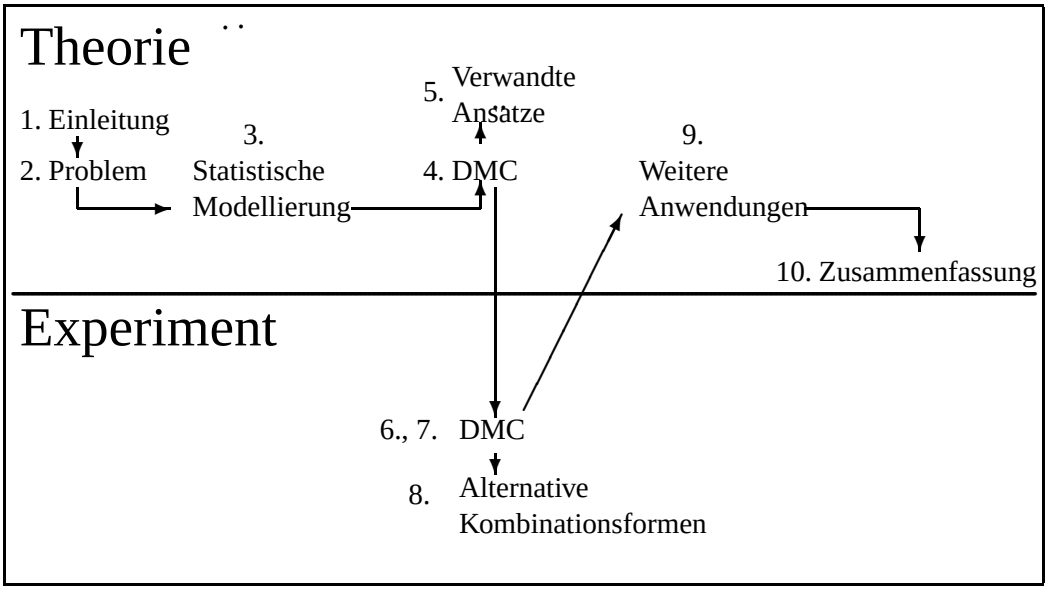

Abbildung 2.2: Gliederung der Arbeit

In dieser Arbeit werden Kenntnisse

- der Bayesschen Entscheidungstheorie [Duda$^+$ 1973], [Fukunaga 1990],
- der statistischen Spracherkennungsmethode [Rabiner$^+$ 1993],
- des Maximum-Entropie-Prinzips [Cover$^+$ 1991], [Kapur$^+$ 1992], und
- über das diskriminative Training [Rabiner$^+$ 1993], [Juang$^+$ 1995]

vorausgesetzt.

Teil II

Theorie der Diskriminativen Modellkombination

Kapitel 3

Statistische Modellierung

3.1 Maximum-Entropie-Verteilungen

Ist keine Information über den möglichen Ausgang eines zufälligen Ereignisses gegeben, so verlangt wissenschaftliche Objektivität, allen Alternativen eine gleiche Wahrscheinlichkeit einzuräumen. Diese Denkweise führt zum Prinzip der maximalen Unsicherheit, d.h. zum **Maximum-Entropie-Prinzip** [Jaynes 1957]. Im Kontext der statistischen Modellierung wird dieses Prinzip wie folgt angewendet, um eine gesuchte Wahrscheinlichkeitsverteilung zu konstruieren:
Als Unsicherheitsmaß für die Wahrscheinlichkeitsverteilung p des zufälligen Ereignisses ω wird die von Shannon definierte **Entropie** H verwendet:

$$H(p) \quad = \quad -\sum_{\omega \in \Omega} p(\omega) \log p(\omega) \tag{3.1}$$

Information über p sei in Form von Erwartungswerten $a_j, j = 1, \ldots, M$ der M **charakterisierenden Funktionen** $m_j(\omega)$**,** $j = 1, \ldots, M$ gegeben.

$$\sum_{\omega \in \Omega} p(\omega) \cdot m_j(\omega) \quad = \quad a_j, \qquad\qquad j = 1, ..., M. \tag{3.2}$$

Sind die Erwartungswerte a_j unbekannt, so können sie mit statistischen Methoden geschätzt werden. Eine weit verbreitete Schätzmethode ist die Maximum-Likelihood-Schätzung [Kapur$^+$ 1992].
Um die Verteilung p zu finden, maximieren wir $H(p)$ unter den Nebenbedingungen (3.2) und der Normalisierungsbedingung für Wahrscheinlichkeiten:

$$\sum_{\omega \in \Omega} p(\omega) \quad = \quad 1. \tag{3.3}$$

Diese Optimierungsaufgabe läßt sich mit Hilfe der Methode der Lagrangeschen Multiplikatoren [Bronstein$^+$ 1957] lösen. Das Ergebnis ist eine Wahrscheinlichkeitsverteilung der Form:

$$p(\omega) \quad = \quad C(\Lambda) \cdot \exp \left\{ \sum_{j=1}^{M} \lambda_j m_j(\omega) \right\}, \tag{3.4}$$

wobei

$$C(\Lambda) \quad = \quad \left(\sum_{\omega \in \Omega} \exp \left\{ \sum_{j=1}^{M} \lambda_j m_j(\omega) \right\} \right)^{-1} \tag{3.5}$$

eine Normierungskonstante ist.
Damit die Abhängigkeit von $\Lambda = (\lambda_1, \ldots, \lambda_M)^T$ deutlich wird, wird im folgenden $p(\omega) = p_\Lambda(\omega)$ geschrieben.

Um die Werte λ_j so zu bestimmen, daß die vorgegebenen Erwartungswertgleichungen (3.2), (3.3) erfüllt sind, wird häufig das sogenannte 'Generalized Iterative Scaling' [Darroch$^+$ 1972] angewendet.

3.2 Log-Lineare Modellkombination

Die vorliegende Arbeit beschäftigt sich mit der Klassifikation von Beobachtungen x in eine Klasse $k = 1, \ldots, K$. Dazu wird mit a-posteriori Wahrscheinlichkeiten $p_\Lambda(\omega) = p_\Lambda(k|x)$ gearbeitet. Sind die charakterisierenden Funktionen $m_j(k|x)$ selbst wieder logarithmierte Verteilungsmodelle, d.h. ist $m_j(k|x) = \log p_j(k|x)$, so ist die Gesamtverteilung $p_\Lambda(k|x)$ eine log-lineare Kombination von Verteilungsmodellen. Sie lautet:

$$p_\Lambda(k|x) \quad = \quad C(\Lambda, x)\exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k|x)\right\}, \tag{3.6}$$

wobei

$$C(\Lambda, x) \quad = \quad \left(\sum_{k'} \exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k'|x)\right\}\right)^{-1} \tag{3.7}$$

die von der Beobachtung x und dem Koeffizientenvektor Λ abhängige Normierungskonstante darstellt.

Eine a-posteriori Verteilung $p_j(k|x)$, deren Logarithmus als charakterisierende Funktion in die Maximum-Entropie-Verteilung $p_\Lambda(k|x)$ eingeht, wird im folgenden kurz **Basismodell** genannt. Die Verteilungsform (3.6) wird mit dem Begriff **log-lineare Kombination von Basismodellen** bezeichnet.

3.2.1 Basismodelle

Für die Bildung der Basismodelle der log-linearen Modellkombination werden im Verlaufe dieser Arbeit folgende Sprachmodelle:

- phrasenbasiertes Unigramm-Sprachmodell
- phrasenbasiertes Bigramm-Sprachmodell
- phrasenbasiertes Bigramm-Lücken-Sprachmodell mit einer Lückenlänge von 1
- phrasenbasiertes Trigramm-Sprachmodell

und folgende akustischen Modelle verwendet:

- Training auf HUB4 Datenbasis
  - wortinterne Triphone,
  - wortinterne Triphone, die MLLR-adaptiert wurden,
  - wortübergreifende Triphone,
  - wortübergreifende Triphone, die MLLR-adaptiert wurden,
  - wortinterne Pentaphone, die MLLR-adaptiert wurden,
- Training auf WSJ0+1 Datenbasis
  - wortinterne Triphone, die MLLR-adaptiert wurden,
  - wortübergreifende Triphone, die MLLR-adaptiert wurden,
  - wortinterne Pentaphone, die MLLR-adaptiert wurden.

Diese Modelle werden in den Kapiteln 6 und 7 genauer beschrieben. Da die Sprachmodelle in Form von a-priori Verteilungen $p(k)$ und die akustischen Modelle in Form von klassenspezifischen Verteilungen $p(x|k)$ vorliegen, werden zunächst zugehörige Basismodelle definiert. Die in den folgenden beiden Unterabschnitten durchgeführte Definition des Sprach(basis)modells und des akustischen Basismodells dienen einer vereinfachten Darstellung der DMC-Theorie.

3.2.1.1 Sprachmodell als Basismodell

Die a-priori Wahrscheinlichkeit einer Wortfolge k wird in der automatischen Spracherkennung im allgemeinen über sogenannte Sprachmodelle $p(k)$ beschrieben. Das zugehörige Basismodell $p_{LM}(k|x)$ wird wie folgt definiert:

$$p_{LM}(k|x) \quad = \quad p(k). \tag{3.8}$$

Das Basismodell $p_{LM}(k|x)$ ist dabei nicht von der akustischen Äußerung x, sondern ausschließlich von der Wortfolge k abhängig. Die Unabhängigkeit von der akustischen Äußerung gehört zu den Modellannahmen des Basismodells $p_{LM}(k|x)$.

3.2.1.2 Akustisches Basismodell

Das akustische Basismodell $p_{AM}(k|x)$ wird aufbauend auf dem akustischen Modell $p(x|k)$ wie folgt definiert:

- Zunächst wird die Modellannahme getroffen, daß die a-priori Verteilung $p_{AM}(k)$ des akustischen Basismodells gleichverteilt ist: $p_{AM}(k) = 1/K$. Dabei ist K die Anzahl der betrachteten Klassen.
- Außerdem wird $p_{AM}(x|k) = p(x|k)$ gesetzt.

Dann gilt für das so konstruierte akustische Basismodell:

$$\begin{aligned} p_{AM}(k|x) &= \frac{p_{AM}(x|k) \cdot p_{AM}(k)}{\sum_{k'} p_{AM}(x|k') \cdot p_{AM}(k')} \\ &= \frac{p(x|k) \cdot 1/K}{\sum_{k'} p(x|k') \cdot 1/K} \\ &= \frac{p(x|k)}{\sum_{k'} p(x|k')} \end{aligned} \tag{3.9}$$

Das akustische Basismodell $p_{AM}(k|x)$ unterscheidet sich also lediglich durch einen klassenunabhängigen Term von der klassenspezifischen Verteilung $p(x|k)$ des akustischen Modells.

3.2.2 Dekomposition in beliebige Basismodelle

Die log-lineare Kombination von akustischen Modellen $p_i(x|k), i = 1, \ldots, I$ und Sprachmodellen $p_j(k)$, $j = 1, \ldots, J$ in eine a-posteriori Verteilung

$$p_{\Lambda}(k|x) \quad = \quad \frac{\prod_i p_i(x|k)^{\lambda_i} \prod_j p_j(k)^{\lambda_j}}{\sum_{k'} \prod_i p_i(x|k')^{\lambda_i} \prod_j p_j(k')^{\lambda_j}} \tag{3.10}$$

kann mit Hilfe von (3.9) und (3.8) vereinfacht werden:

Bezeichnet man das zum akustischen Modell $p_i(x|k)$ zugehörige Basismodell mit $p_{AM,i}(k|x)$ und das zum Sprachmodell $p_j(k)$ zugehörige Basismodell mit $p_{LM,j}(k|x)$, so ergibt sich:

$$p_{\Lambda}(k|x) \quad = \quad \frac{\prod_i p_{AM,i}(k|x)^{\lambda_i} \prod_j p_{LM,j}(k|x)^{\lambda_j}}{\sum_{k'} \prod_i p_{AM,i}(k'|x)^{\lambda_i} \prod_j p_{LM,j}(k'|x)^{\lambda_j}} \tag{3.11}$$

Die Äquivalenz von (3.11) und (3.10) kann durch Einsetzen von (3.9) und (3.8) in (3.11) gezeigt werden. Da die Aufteilung in akustische und in Sprachmodelle für die weitere Darstellung der Theorie nicht erforderlich ist, wird

von dieser Aufteilung im folgenden abgesehen und nur noch von Basismodellen gesprochen. Zur Unterscheidung der Basismodelle werden diese mit Indizes versehen, wobei ein spezieller Index j für ein spezielles akustisches Basismodell oder Sprachbasismodell oder irgendein anderes Basismodell steht. Die Dekomposition einer Verteilung $p(k|x)$ in M Basismodelle wird im weiteren Verlauf der Arbeit wie folgt notiert:

$$p_{\Lambda}(k|x) = \frac{\prod_{j=1}^{M} p_j(k|x)^{\lambda_j}}{\sum_{k'} \prod_{j=1}^{M} p_j(k'|x)^{\lambda_j}} \tag{3.12}$$

3.3 Minimierung der Fehlerrate

Das eigentliche Ziel bei der Entwicklung eines Mustererkennungssystems ist eine minimale Fehlerrate dieses Systems auf Testdaten, die im Training nicht gesehen wurden. Dazu wählt man eine genügend große Trainingsdatenmenge, die die zukünftigen Testdaten möglichst gut repräsentiert. Beide Datenmengen sollten durch den gleichen stochastischen Prozeß generiert werden. Optimiert man das Erkennungssystem auf repräsentativen Trainigsdaten, so wird sich das System auch auf den unbekannten Testdaten optimal verhalten. In der akustischen Modellierung haben sich im wesentlichen zwei Schätzmethoden durchgesetzt, zum einen die Maximum-Likelihood-Schätzung und zum anderen diskriminative Verfahren. Während die Maximum-Likelihood-Schätzung das Ziel hat, die gegebenen Trainingsdaten möglichst genau reproduzieren zu können, konzentrieren sich diskriminative Verfahren darauf, die Anzahl der Fehlklassifikationen auf den Trainingsdaten zu minimieren.

3.3.1 Motivation für diskriminative Verfahren

Aus der Optimalität der Maximum-Likelihood-Schätzung einer a-posteriori Verteilung folgt nicht zwangsläufig die Optimalität der Entscheidungsregel, wenn die Form der wahren a-posteriori Verteilung unbekannt ist und Modellannahmen notwendig werden. Ein Ansatz, der dieses Problem zu umgehen versucht, besteht in der direkten Formulierung des Schätzproblems als Fehlerminimierungsproblem. Diesen Ansatz nennt man **Diskriminatives Training**. Ziel des diskriminativen Trainings ist es, die Beobachtungen x korrekt zu klassifizieren und nicht die Verteilung der Daten (k, x) zu schätzen. Eine Schwierigkeit, die mit dem diskriminativen Training verbunden ist, liegt in der Formulierung einer optimierbaren Zielfunktion.

3.3.2 Zielfunktion - Wortfehlerrate

Die Zielfunktion bei der Optimierung eines Erkenners ist die Anzahl der Fehler, die der Erkenner auf den Trainingssätzen macht, geteilt durch die Anzahl der Wörter, die in den Trainingssätzen enthalten sind. Die Trainingsdatenbasis besteht aus den N Äußerungen $x_n, n = 1, \ldots, N$. Sie beinhalten die Wortfolgen k_n der Länge L_n. Die vom Erkenner hypothetisierten Wortfolgen werden mit k bezeichnet. Die Fehlerzahl der Wortfolge k bezogen auf die Wortfolge k_n wird mit Hilfe der **Levenshtein-Distanz** $\mathcal{L}(k, k_n)$ gemessen. Dann kann die vom freien Parameter Λ abhängende Fehlerrate $E(\Lambda)$ des Erkenners wie folgt berechnet werden:

$$E(\Lambda) = \frac{1}{\sum_{n=1}^{N} L_n} \cdot \sum_{n=1}^{N} \mathcal{L}\left(k_n, \arg\max_k \left(p_\Lambda(k|x_n)\right)\right) \tag{3.13}$$

Die Fehlerrate des Erkenners kann auch in Form der Funktion (3.14) notiert werden. Hier geht das logarithmierte Verhältnis der a-posteriori Wahrscheinlichkeiten der Klasse k und der korrekten Klasse k_n ein:

$$E(\Lambda) = \frac{1}{\sum_{n=1}^{N} L_n} \cdot \sum_{n=1}^{N} \mathcal{L}\left(k_n, \arg\max_k \left(\log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right)\right) \tag{3.14}$$

Diese Darstellung der Wortfehlerrate des Erkenners wird im folgenden bevorzugt, da die in dieser Arbeit verwendeten Optimierungskriterien von Termen der Form $\log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}$ abhängen. Durch diese Schreibweise entfällt sowohl die Abhängigkeit von der Normierungskonstanten der Verteilung p_{Λ} als auch die exponentielle Form von p_{Λ}. Das logarithmierte Verhältnis zweier Wahrscheinlichkeitsbewertungen (engl. 'Log Likelihood-Ratio') wird

häufig auch in anderen Arbeiten zur Beschreibung von Klassifikationsaufgaben der statistischen Mustererkennung verwendet, wie z.B. in [Fukunaga 1990], S.51.
Die Aufgabe eines diskriminativen Verfahrens ist nun, die Parameter Λ der Verteilung p_Λ so einzustellen, daß die Fehlerrate $E(\Lambda)$ so klein wie möglich wird. Wie man der Zielfunktion ansieht, ist eine direkte Optimierung über ein numerisches Gradientenverfahren nicht möglich, da die Zielfunktion eine stückweise konstante Funktion ist ! Aus diesem Grunde muß man auf numerische Gradientenverfahren verzichten oder die Zielfunktion durch eine geglättete Zielfunktion approximieren. Ein sehr aufwendiges Verfahren, welches direkt die Zielfunktion optimiert, ist die **Stochastische Suche**, die im folgenden Abschnitt skizziert wird. Anschließend wird auf das Standardverfahren zur Minimierung eines geglätteten Fehlerkriteriums - das **Minimum Classification Error (MCE) Training** - eingegangen.

3.3.3 Stochastische Suche

Ein sehr simples, wenn auch sehr aufwendiges Verfahren zur Bestimmung des optimalen Wertes von Λ ist die sogenannte stochastische Suche mit Hilfe der Methode der konkurrierenden Punkte [Müller$^+$ 1986]. Dabei werden verschiedene Werte $\Lambda_1, \ldots, \Lambda_n$ mit Hilfe eines gleichverteilten Zufallsgenerators generiert und nach ihrer Optimalität sortiert. In der lokalen Umgebung der besten m dieser Punkte wird dann das Verfahren wiederholt und anschließend unter den erhaltenen $m(n+1)$ Punkten wieder die besten m Punkte ausgewählt. Das Verfahren wird zyklisch wiederholt, wobei die Größe der lokalen Umgebung immer weiter eingeschränkt wird. Nach einer bestimmten Anzahl von Zyklen wird das Verfahren abgebrochen. Dieses Verfahren wird in vereinfachter Form beim weit verbreiteten 'Tuning' von Spracherkennungssystemen eingesetzt. Hier werden die Werte $\Lambda_1, \ldots, \Lambda_n$ durch menschliche Spracherkennungsexperten mehr oder weniger gut geraten. Eine solche Herangehensweise ist sicher nur bei der Optimierung weniger Parameter sinnvoll, wie z.B. beim Einstellen von Sprachmodellfaktor und Worteinfügungsstrafe.

3.3.4 MCE-Training

Dieses Trainingsverfahren ist aus der Literatur [Juang$^+$ 1992], [Rabiner$^+$ 1993], S. 291, [Juang$^+$ 1995] bekannt und wird zum besseren Verständnis der vorliegenden Arbeit kurz skizziert. Als Optimierungskriterium dient die sogenannte **geglättete empirische Fehlerrate**

$$E_{MCE}(\Lambda) \quad = \quad \frac{1}{N} \sum_{n=1}^{N} \ell(x_n, k_n, \Lambda) \tag{3.15}$$

auf den gegebenen Trainingsdaten. $\ell(x_n, k_n, \Lambda)$ wird dabei in [Juang$^+$ 1995] als geglättete Mißklassifikationsfunktion der Beobachtung x_n bezeichnet und wie folgt definiert:

$$\ell(x_n, k_n, \Lambda) \quad = \quad \left(1 + A \cdot \left(\frac{1}{K-1} \sum_{k \neq k_n} \exp\left\{ -\eta \log \frac{p_\Lambda(k_n|x_n)}{p_\Lambda(k|x_n)} \right\} \right)^{-\frac{B}{\eta}} \right)^{-1} . \tag{3.16}$$

Dabei ist wie bisher K die Anzahl der Klassen. Die Werte $A > 0, B > 0, \eta > 0$ müssen geeignet eingestellt werden. Die Funktion ℓ besitzt einen Wert zwischen 0 und 1, abhängig davon, ob der Klassifikator korrekt entscheiden wird oder nicht. Die geglättete empirische Fehlerrate $E_{MCE}(\Lambda)$ wird mit Hilfe des **Generalized Probabilistic Descent Theorems** [Juang$^+$ 1992] minimiert. Entsprechend diesem Theorem kann Λ durch die Iterationsvorschrift

$$\Lambda^{(I+1)} \quad = \quad \Lambda^{(I)} - \varepsilon U \nabla_\Lambda E_{MCE}(\Lambda)_{|\Lambda=\Lambda^{(I)}} \tag{3.17}$$

optimiert werden. Die Matrix U muß dabei positiv definit sein, der Wert ϵ bestimmt die Schrittweite bei der Optimierung von Λ. Dieses Verfahren kann als Gradienten-Abstiegs-Verfahren aufgefaßt werden.

3.4 Diskussion

Maximum-Entropie und Diskriminatives Training sind zwei etablierte Denkmodelle in der maschinellen Spracherkennung. Eine direkte Verbindung beider Ansätze scheint wegen ihrer unterschiedlichen Ziele zunächst nicht möglich zu sein. Die grundlegende Idee der vorliegenden Arbeit besteht nun darin, daß die Erwartungswerte a_j der Maximum-Entropie-Verteilung nicht aus einer Maximum-Likelihood-Schätzung stammen, sondern diskriminativ trainiert werden. Da die unbekannten Koeffizienten $\Lambda = (\lambda_1, \ldots, \lambda_M)^T$ von den Erwartungswerten a_j abhängen, kann man die Koeffizienten Λ auch direkt diskriminativ trainieren, es entsteht zwangsläufig die gleiche Verteilung. Aus dieser Diskussion ergibt sich folgende Hypothese über die optimale Kombination von Basismodellen:
Wenn die wahre Form der gesuchten Verteilung unbekannt ist, dann motiviert das Maximum-Entropie-Prinzip eine log-lineare Verteilungsform (3.6). Das Diskriminative Training optimiert die freien Parameter Λ der Verteilung (3.6) bezüglich der Klassifikationsaufgabe.
Diese Hypothese wird in den folgenden Kapiteln erhärtet. Methoden zur diskriminativen Bestimmung der Parameter Λ werden im folgenden Kapitel beschrieben.

Kapitel 4

Diskriminative Modellkombination (DMC)

4.1 Motivation

Die statistische Spracherkennungsmethode bedient sich der Bayesschen Entscheidungstheorie, um Erkenner mit minimaler Fehlerrate zu konstruieren [Duda$^+$ 1973], S. 10. Entsprechend dieser Theorie muß eine Beobachtung x in diejenige Klasse k eingeordnet werden (kurz $x \in k$), für die bei gegebener a-posteriori Verteilung $\pi(k|x)$ gilt:

$$\forall k' = 1, ..., K; k' \neq k: \quad \log \frac{\pi(k|x)}{\pi(k'|x)} \geq 0. \tag{4.1}$$

Die Entscheidungsregel wird aus den in Abschnitt 3.3.2 genannten Gründen in Form von logarithmierten Wahrscheinlichkeitsverhältnissen dargestellt.
Der Term $\log(\pi(k|x)/\pi(k'|x))$ wird als **Diskriminantenfunktion** bezeichnet [Duda$^+$ 1973], S. 20. Spezialisiert man die Entscheidungsregel (4.1) auf die Erkennung ganzer Sätze, so werden beobachtete Äußerungen $\vec{x}_1^T = (x^1, \ldots, x^T)$ der zeitlichen Länge T in gesprochene Wortfolgen $w_1^L = (w^1, ..., w^L)$ der Länge L klassifiziert. Die wahre a-posteriori Verteilung der menschlichen Sprache $\pi(w_1^L|\vec{x}_1^T)$ konnte wegen ihrer komplizierten Gestalt bisher formal nicht erfaßt werden. Sie muß durch eine Modellverteilung $p(w_1^L|\vec{x}_1^T)$ approximiert werden. Dabei wird die Form der Verteilung $p(w_1^L|\vec{x}_1^T)$ durch intuitives Wissen vorgegeben, die unbekannten Parameter der Verteilung werden auf Trainingsdaten geschätzt. Die gewonnene Verteilung $p(w_1^L|\vec{x}_1^T)$ wird anschließend in die Bayessche Entscheidungsregel eingesetzt. Die Äußerung $\vec{x}_1^T$ wird dann derjenigen Wortfolge w_1^L zugeordnet, für die gilt:

$$\forall {w'}_1^{L'} \neq w_1^L: \quad \log \frac{p(w_1^L|\vec{x}_1^T)}{p({w'}_1^{L'}|\vec{x}_1^T)} \geq 0. \tag{4.2}$$

Durch Umformung der Diskriminantenfunktion

$$\begin{aligned} g(\vec{x}_1^T, w_1^L, {w'}_1^{L'}) &= \log \frac{p(w_1^L|\vec{x}_1^T)}{p({w'}_1^{L'}|\vec{x}_1^T)} \\ &= \log \frac{p(w_1^L)p(\vec{x}_1^T|w_1^L)}{p({w'}_1^{L'})p(\vec{x}_1^T|{w'}_1^{L'})}, \end{aligned} \tag{4.3}$$

erhält man in natürlicher Weise die Trennung zwischen dem Sprachmodell $p(w_1^L)$ und dem akustisch-phonetischen Modell $p(\vec{x}_1^T|w_1^L)$. Das Sprachmodell $p(w_1^L)$ beschreibt dabei die Wahrscheinlichkeit für das Auftreten der Wort-

folge w_1^L an sich, das akustisch-phonetische Modell $p(\vec{x}_1^T|w_1^L)$ bewertet die Wahrscheinlichkeit, daß beim Sprechen der Wortfolge w_1^L das akustische Signal $\vec{x}_1^T$ entsteht. Beide Modelle werden unabhängig voneinander geschätzt, wodurch die akustischen Trainingsdaten und die Sprachmodelltrainingsdaten optimal ausgenutzt werden können. Durch eine Abweichung der Form der Verteilung p von der unbekannten Verteilung π kann die Entscheidungsregel (4.3) suboptimal sein, obwohl die Verteilung p optimal geschätzt wurde. Dieser Umstand motiviert die Verwendung diskriminativer Verfahren. Diskriminative Verfahren optimieren die Verteilung p direkt bezüglich der empirischen auf Trainingsdaten gemessenen Fehlerrate der Entscheidungsregel. Das einfachste Beispiel für eine solche diskriminative Optimierung ist die Verwendung des sogenannten **Sprachmodellfaktors** λ . Dabei wird (4.3) wie folgt modifiziert

$$g^\lambda(\vec{x}_1^T, w_1^L, {w'}_1^{L'}) \quad = \quad \log \frac{p(w_1^L)^\lambda \cdot p(\vec{x}_1^T|w_1^L)}{p({w'}_1^{L'})^\lambda \cdot p(\vec{x}_1^T|{w'}_1^{L'})} \,. \tag{4.4}$$

Experimentelle Erfahrungen [Lee 1989] zeigen, daß die Fehlerrate der Entscheidungsregel (4.4) sinkt, wenn $\lambda > 1$ gewählt wird. Die Ursache für diese Abweichung von der Theorie (d.h. $\lambda = 1$) liegt offensichtlich in der unvollständigen oder fehlerhaften Modellierung der Wahrscheinlichkeit des Verbundereignisses $(w_1^L, \vec{x}_1^T)$. Letzteres ist unvermeidbar, da unser Wissen über den generierenden Prozeß des Ereignisses $(w_1^L, \vec{x}_1^T)$ zu unvollständig ist.
Vielfältige akustisch-phonetische Modellierungen und Sprachmodellierungen wurden bisher analysiert. Literatur dazu findet man unter anderem in den Proceedings der ICASSP (International Conference on Acoustics Speech and Signal Processing), EUROSPEECH und ICSLP (International Conference on Speech and Language Processing) der letzten 20 Jahre. Ziel dieser Analysen war es, die beste Modellierung für die jeweilige Erkennungsaufgabe zu finden.
Im Gegensatz dazu wird bei der in der vorliegenden Arbeit vorgeschlagenen Vorgehensweise nicht nach der besten Modellierung für jede neue Aufgabe gesucht, sondern alle verfügbaren akustisch-phonetischen Basismodelle und Sprachmodelle in einer log-linearen Modellkombination zusammengefaßt (vgl. Abschnitt 3.2) und diese bezüglich der Erkennungsaufgabe diskriminativ optimiert (vgl. Abschnitte 3.3, 3.4).
Das Neue an diesem Ansatz ist, daß nicht versucht wird, die bekannten Eigenschaften der Sprache in ein einziges akustisch-phonetisches Verteilungsmodell und ein einziges Sprachmodell zu integrieren, die dann komplex und schwer trainierbar werden. Die verschiedenen akustisch-phonetischen und grammatischen Eigenschaften werden hier in separate Basismodelle $p_j(w_1^L|\vec{x}_1^T), j = 1, \ldots, M$ abgebildet (vgl. Abschnitt 3.2), auf geeigneten Daten trainiert und anschließend in eine Verteilung

$$\begin{aligned} p_{\Lambda}(w_1^L|\vec{x}_1^T) \quad &= \quad C(\Lambda) \cdot \prod_{j=1}^{M} p_j(w_1^L|\vec{x}_1^T)^{\lambda_j} \\ &= \quad \exp\left\{ \log C(\Lambda) + \sum_{j=1}^{M} \lambda_j \log p_j(w_1^L|\vec{x}_1^T) \right\} \end{aligned} \tag{4.5}$$

integriert. Der Einfluß des Basismodells p_j auf die Verteilung p_{Λ} wird durch den Koeffizienten λ_j bestimmt. Der Faktor $C(\Lambda)$ garantiert dabei die Erfüllung der Normierungsbedingung für Wahrscheinlichkeiten. Die freien Koeffizienten $\Lambda = (\lambda_1, ..., \lambda_M)^T$ müssen dabei so eingestellt werden, daß die resultierende Diskriminantenfunktion

$$g^\Lambda(\vec{x}_1^T, w_1^L, {w'}_1^{L'}) \quad = \quad \log \frac{\Pi_{j=1}^M p_j(w_1^L|\vec{x}_1^T)^{\lambda_j}}{\Pi_{j=1}^M p_j({w'}_1^{L'}|\vec{x}_1^T)^{\lambda_j}} \tag{4.6}$$

eine möglichst geringe Fehlerrate (3.14) (vgl. Abschnitt 3.3) besitzt. Die **Diskriminative Modellkombination** verallgemeinert die bisherige Herangehensweise an das Entscheidungsproblem (4.1) unter Berücksichtigung experimenteller Beobachtungen [Lee 1989]. Die Diskriminantenfunktionen in (4.4) oder (4.3) sind einfache Spezialfälle der Diskriminantenfunktion (4.6).

4.2 Diskriminatives Training der Modellkombination

Zunächst bietet sich ein diskriminatives Training von Λ mit Hilfe der MCE-Methode ([Juang$^+$ 1995], vgl. Abschnitt 3.3.4) an. Dieses Kriterium beinhaltet jedoch die geglättete Satzfehlerrate und nicht die Wortfehlerrate.

Ein Versuch, die Wortfehlerrate in das MCE-Trainingskriterium zu integrieren, wird in Abschnitt 4.4 beschrieben. Ein Verfahren, welches direkt die geglättete empirische Wortfehlerrate optimiert (MWE-Training), wird im darauf folgenden Abschnitt 4.5 vorgestellt. Da die bisher verwendeten Glättungsmethoden zu numerischen Iterationsverfahren führen, stellt sich die Frage nach einer geschlossenen Lösungsgleichung für Λ. Als Optimierungskriterium für eine geschlossene Lösung bietet sich zunächst (motiviert durch die Lösungsform bei der Optimierung der geglätteten empirischen Wortfehlerrate) der *Quadratmittel-Abstand zwischen der Diskriminantenfunktion g^{Λ} und einer monotonen Funktion der Fehlerrate $\tilde{\mathcal{L}}$* an (Abschnitt 4.6). Ungeachtet der geschlossenen Lösungsgleichung für Λ hat dieses Kriterium den Nachteil, daß es nicht unmittelbar die Wortfehlerrate des Klassifikators optimiert. Durch die Synthese aus dem MWE-Kriterium in Abschnitt 4.5 und einer in den λ_j quadratischen Zielfunktion (Abschnitt 4.6) entsteht schließlich die **parabolisch geglättete Fehlerrate**. Dieses Kriterium ist zum einen direkt mit der empirischen Wortfehlerrate des Erkenners verbunden und führt zum anderen zu einer geschlossenen Lösungsgleichung für die Koeffizienten der diskriminativen Modellkombination (Abschnitt 4.7). Es beruht auf der Approximation der geglätteten Indikatorfunktion beim MWE-Training durch den geeigneten Abschnitt eines Parabelastes.

4.3 Optimierung auf der Satzebene

DMC ist auf verschiedenen Hierarchieebenen des Spracherkennungssystems anwendbar. Da sich die erhaltenen Ergebnisse leicht auf andere Hierarchieebenen übertragen lassen (siehe Kapitel 9), wird im folgenden weiterhin von einer Optimierung auf der Satzebene ausgegangen. Dazu wird vorausgesetzt, daß die Trainingsdaten in Form des Textes und der Basismodellbewertungen des gesamten Satzes vorliegen. Dabei definiert jede Wortfolge eine Klasse k. Die aus dem akustischen Signal der gesprochenen Äußerung extrahierte Merkmalsvektorfolge $\vec{x}_1^T$ wird im folgenden kurz als Beobachtung x bezeichnet. Es seien N Trainingsbeobachtungen $x_n, n = 1, \ldots, N$ gegeben. Für jede Trainingsbeobachtung x_n bezeichnet k_n die korrekte Klasse und $k \neq k_n$ eine der $K-1$ rivalisierenden Klassen. Diese werden durch ein $\mathcal{N}$-Best-Verfahren [Tran$^+$ 1997] bestimmt.
Die Ähnlichkeit der Klassen wird durch die Levenshtein-Distanz $\mathcal{L}(k_n, k)$ beschrieben. Dabei ist die Anzahl der Wörter im Satz gleich L_n.

4.4 Modifiziertes MCE-Training

Im folgenden Abschnitt werden die Argumentationen aus [Juang$^+$ 1995] und aus Abschnitt 3.3 auf die Bestimmung der Koeffizienten Λ der log-linearen Verteilungskombination (4.5) angewendet. Der Schlüsselpunkt des geglätteten Optimierungskriteriums besteht darin, die Entscheidungsregel (4.1) in Form eines stetigen 'Mißklassifikationsmaßes' auszudrücken. Dafür gibt es viele Möglichkeiten. Überträgt man das in [Juang$^+$ 1995] verwendete Mißklassifikationsmaß $d(x_n, k_n, \Lambda)$ auf die Modellkombinationsaufgabe, so erhält man:

$$d(x_n, k_n, \Lambda) = \log \left(\frac{1}{K-1} \sum_{k \neq k_n} \exp \left\{ \eta \cdot \log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)} \right\} \right)^{\frac{1}{\eta}}, \tag{4.7}$$

wobei η eine positive Zahl ist. Dieses Mißklassifikationsmaß ist eine kontinuierliche Funktion des Koeffizientenvektors Λ. Wird η groß gewählt ($\eta \to \infty$), dann erhält man

$$\eta \to \infty : \qquad d(x_n, k_n, \Lambda) = \max_{k \neq k_n} \log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)}. \tag{4.8}$$

Ist nun $d(x_n, k_n, \Lambda) \leq 0$, so wird eine korrekte Entscheidung getroffen, anderenfalls wird eine Fehlentscheidung gefällt, die zu $\mathcal{L}(k, k_n)$ Fehlern führt. Durch Wahl eines endlichen positiven Wertes für η kann man mehrere rivalisierende Klassen k in die Optimierung des Koeffizienten Λ einbeziehen. Um die Definition der Zielfunktion

abzuschließen, wird wie in [Juang$^+$ 1995] auf das Mißklassifikationsmaß (4.7) eine Sigmoidfunktion angewendet. Die Zielfunktion lautet damit:

$$\ell(x_n, k_n, \Lambda) \quad = \quad \frac{1}{1 + \exp\left\{-a(b + d(x_n, k_n, \Lambda))\right\}}, \tag{4.9}$$

wobei a, b geeignet eingestellt werden müssen. Offensichtlich ist nun für $d(x_n, k_n, \Lambda) \leq 0$ (korrekte Entscheidung) der Wert der Zielfunktion $\ell(x_n, k_n, \Lambda)$ nahe 0, während er für $d(x_n, k_n, \Lambda) > 0$ (Fehlentscheidung) nahe 1 sein wird. Die mittlere geglättete empirische Satzfehlerrate aller Trainingsbeobachtungen ergibt sich schließlich aus

$$E_{MCE}(\Lambda) \quad = \quad \frac{1}{N} \sum_{n=1}^{N} \ell(x_n, k_n, \Lambda). \tag{4.10}$$

Die so definierte Zielfunktion beschreibt jedoch nur die Satzfehlerrate des Erkenners, da hier die Anzahl $\mathcal{L}(k, k_n)$ der Wortfehler der rivalisierenden Hypothesen nicht eingeht. Um dies zu kompensieren, wird das Mißklassifikationsmaß d für DMC auf Satzebene wie folgt modifiziert:

$$d'(x_n, k_n, \Lambda) \quad = \quad \log \left(\frac{1}{K-1} \sum_{k \neq k_n} \exp \left\{ \eta \cdot \mathcal{L}(k, k_n) \log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)} \right\} \right)^{\frac{1}{\eta}}. \tag{4.11}$$

Wird hier wieder ein großes η gewählt ($\eta \to \infty$), dann erhält man

$$\eta \to \infty : \qquad d(x_n, k_n, \Lambda) \quad = \quad \max_{k \neq k_n} \; \mathcal{L}(k, k_n) \cdot \log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)}. \tag{4.12}$$

Dadurch wird für großes η die Mißklassifikationsfunktion linear mit der Wortfehlerrate $\mathcal{L}(k, k_n)$ gewichtet. Die Gradientenbildung der entsprechend modifizierten Zielfunktion (4.10) wird in Anhang 11.1 beschrieben. Mit dem Iterationsindex I lautet das resultierende Iterationsverfahren :

$$\begin{aligned}
\lambda_j^{(0)} \quad &= \quad 0 \quad \text{(Gleichverteilung)} \\
\lambda_j^{(I+1)} \quad &= \quad \lambda_j^{(I)} - \varepsilon \sum_{n=1}^{N} \ell(x_n, k_n, \Lambda^{(I)}) \left(1 - \ell(x_n, k_n, \Lambda^{(I)})\right) \cdot \\
&\qquad \cdot \frac{\sum_{k \neq k_n} \mathcal{L}(k, k_n) \log \left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)} \right) \left[\frac{p_{\Lambda^{(I)}}(k|x_n)}{p_{\Lambda^{(I)}}(k_n|x_n)} \right]^{\eta \mathcal{L}(k, k_n)}}{\sum_{k \neq k_n} \left[\frac{p_{\Lambda^{(I)}}(k|x_n)}{p_{\Lambda^{(I)}}(k_n|x_n)} \right]^{\eta \mathcal{L}(k, k_n)}} \\
\Lambda^{(I)} \quad &= \quad (\lambda_1^{(I)}, \ldots, \lambda_M^{(I)})^T \\
&\qquad j = 1, \ldots, M.
\end{aligned} \tag{4.13}$$

Damit ist gezeigt, daß das MCE-Trainingskriterium durch eine Wortfehlerratenmessung gewichtet werden kann. Jedoch optimiert das modifizierte MCE-Trainingskriterium nicht direkt die empirische Wortfehlerrate. Eine im Rahmen dieser Arbeit neu entwickelte, bessere Approximation der empirischen Wortfehlerrate wird im folgenden Abschnitt behandelt. Auch dieses Kriterium führt zu einer iterativen Lösung.

4.5 Minimierung der geglätteten Wortfehlerrate (MWE)

Die Wortfehlerrate $E(\Lambda)$ (3.14) wird in diesem Abschnitt durch folgende glatte Funktion E_{MWE} approximiert:

$$E_{MWE}(\Lambda) = \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) S(k, n, \Lambda). \tag{4.14}$$

Die Funktion $S(k, n, \Lambda)$ ist dabei eine Indikatorfunktion. Sie nimmt im ungeglätteten Fall den Wert 1 an, wenn sich der Klassifikator für die Hypothese k entscheidet, sonst den Wert 0. [1] Ist S also eine Indikatorfunktion mit den beschriebenen Eigenschaften und ist $\mathcal{L}(k, k_n)$ die Levenshtein-Distanz der beiden Wortfolgen k, k_n, dann stellt Gleichung (4.14) die exakte empirische Wortfehlerrate (3.14) dar. Um ein differenzierbares Kriterium zu erhalten, wird die Indikatorfunktion S durch folgende geglättete Indikatorfunktion ersetzt.

$$S(k, n, \Lambda) = \frac{p_\Lambda(k|x_n)^\eta}{\sum_{k'} p_\Lambda(k'|x_n)^\eta}. \tag{4.15}$$

Eine ähnliche Indikatorfunktion wurde bereits in [Ney 1995], S.115, für die Glättung des Klassifikationsfehlers eingesetzt. Um das resultierende Trainingskriterium mit den anderen in diesem Kapitel betrachteten Kriterien in Beziehung setzen zu können wird S umgeformt:

$$S(k, n, \Lambda) = \frac{\exp\left\{\eta \log \dfrac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right\}}{\sum_{k'} \exp\left\{\eta \log \dfrac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)}\right\}}. \tag{4.16}$$

Je größer der Exponent η wird, um so genauer approximiert die Funktion $S(k, n, \Lambda)$ die Entscheidung des Klassifikators. Der Wert η sollte jedoch nicht zu groß gewählt werden, um die Anwendung numerischer Verfahren zu ermöglichen. Außerdem können dann auch schwächere Rivalen in die Summation eingehen, wodurch die Schätzung der Parameter Λ robuster wird. In den Experimenten wurde $\eta = 3$ verwendet, da hier die beiden Effekte - Annäherung an die exakte empirische Fehlerrate und numerische Optimierbarkeit - ausbalanciert waren. Durch diese Wahl von η kann die Summation in der Funktion S auf eine endliche Anzahl wahrscheinlichster Wortfolgen beschränkt werden, was die praktische Implementierung wesentlich vereinfacht.
Die Ableitung von $E_{MWE}(\Lambda)$ nach Λ (siehe Anhang 11.2.1) ergibt:

$$\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} = \frac{\eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} S(k, n, \Lambda) \tilde{\mathcal{L}}(k, n, \Lambda) \log \frac{p_i(k|x_n)}{p_i(k_n|x_n)}$$

wobei

$$\tilde{\mathcal{L}}(k, n, \Lambda) = \mathcal{L}(k, k_n) - \sum_{k' \neq k_n} S(k', n, \Lambda) \mathcal{L}(k', k_n) \tag{4.17}$$

die Differenz der Fehlerzahl der Hypothese k zur mittleren gewichteten Fehlerzahl der Trainingsbeobachtung x_n ist. Der Wert $\tilde{\mathcal{L}}(k, n, \Lambda)$ kann als 'mittelwertfreie' Fehlerzahl interpretiert werden.
Mit Hilfe dieser Ableitung läßt sich schließlich ein entsprechendes Gradienten-Abstiegs-Verfahren konstruieren. Das resultierende Iterationsverfahren lautet mit dem Iterationsindex I:

$$\lambda_j^{(0)} = 0 \quad \text{(Gleichverteilung)}$$

$$\lambda_j^{(I+1)} = \lambda_j^{(I)} - \frac{\varepsilon \cdot \eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} S(k, n, \Lambda^{(I)}) \tilde{\mathcal{L}}(k, n, \Lambda^{(I)}) \log \frac{p_j(k|x_n)}{p_j(k_n|x_n)}$$

[1] Der Einfachheit wegen wird angenommen, daß es genau eine Hypothese mit maximaler Wahrscheinlichkeit gibt.

$$\Lambda^{(I)} = (\lambda_1^{(I)}, \ldots, \lambda_M^{(I)})^T$$

$$j = 1, \ldots, M. \tag{4.18}$$

Der Koeffizient λ_j berechnet sich aus einer gewichteten Korrelation der Funktion $\tilde{\mathcal{L}}(k, n, \Lambda^{(I)})$ der Levenshtein-Distanz und der Diskriminantenfunktion $\log \frac{p_j(k|x_n)}{p_j(k_n|x_n)}$. Diese Lösungsform entsteht auch bei der Verwendung von Quadratmittelmethoden. Da diese außerdem zu einer geschlossenen Lösung führen, beschäftigt sich der nächste Abschnitt mit dem Quadratmittelansatz.

4.6 Quadratmittelansatz

Ist die Anzahl der Fehler, die bei der Entscheidung für die Satzhypothese $k, k = 1, \ldots, K$ entstehen, bereits vor der Klassifikation bekannt, so läßt sich ein Klassifikator mit minimaler Fehlerrate konstruieren, indem jeweils die Hypothese mit minimaler Fehlerzahl ausgewählt wird. Mit Hilfe der Levenshtein-Distanz $\mathcal{L}(k, k_n)$ kann folglich auf den Trainingsdaten ein idealer Klassifikator gebildet werden. Außerdem kann jede monotone Funktion $\breve{\mathcal{L}}(k, k_n)$ der Levenshtein-Distanz $\mathcal{L}(k, k_n)$ als ideale Diskriminantenfunktion aufgefaßt werden. Um eine geschlossene Lösung für die Koeffizienten Λ zu finden wird in diesem Abschnitt der mittlere quadratische Abstand der Diskriminantenfunktion der Modellkombination zur idealen Diskriminantenfunktion $\breve{\mathcal{L}}(k, k_n)$ minimiert. Das Quadratmittelkriterium $D(\Lambda)$ (engl. Mean Squared Error Criterion, MSE-Kriterium) lautet:

$$D(\Lambda) = \frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} \left(\log \frac{p_{\Lambda}(k_n|x_n)}{p_{\Lambda}(k|x_n)} - \breve{\mathcal{L}}(k, k_n) \right)^2 . \tag{4.19}$$

Die Summation über k bezieht alle rivalisierenden Klassen in das Kriterium ein. Die Verteilung p_{Λ} wird so bestimmt, daß auf den Trainingsdaten (x_n, k) das Log Likelihood-Verhältnis zwischen korrekter und fehlerhafter Hypothese dem Wert $\breve{\mathcal{L}}(k, k_n)$ möglichst ähnlich ist.
Um $D(\Lambda)$ zu minimieren, wird $D(\Lambda)$ vereinfacht und nach den Komponenten des Koeffizientenvektors Λ abgeleitet. Aus (4.19) folgt zunächst mit (4.5)

$$D(\Lambda) = \frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} \left(\sum_{j=1}^{M} \lambda_j \log \frac{p_j(k_n|x_n)}{p_j(k|x_n)} - \breve{\mathcal{L}}(k, k_n) \right)^2 . \tag{4.20}$$

Wir leiten den rechten Term in der obigen Gleichung nach $\lambda_i, i = 1, ..., M$ ab und setzen die Ableitung auf Null:

$$\begin{aligned} 0 &= \frac{\partial}{\partial \lambda_i} \frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} \left(\sum_{j=1}^{M} \lambda_j \log \frac{p_j(k_n|x_n)}{p_j(k|x_n)} - \breve{\mathcal{L}}(k, k_n) \right)^2 \\ 0 &= \frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} \left(\log \frac{p_i(k_n|x_n)}{p_i(k|x_n)} \right) \cdot \\ &\quad \cdot \left(\sum_{j=1}^{M} \lambda_j \log \frac{p_j(k_n|x_n)}{p_j(k|x_n)} - \breve{\mathcal{L}}(k, k_n) \right) . \end{aligned} \tag{4.21}$$

Vertauschung der Summationsreihenfolge liefert:

$$\sum_{j=1}^{M} \lambda_j \left[\frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} \left(\log \frac{p_i(k_n|x_n)}{p_i(k|x_n)} \right) \left(\log \frac{p_j(k_n|x_n)}{p_j(k|x_n)} \right) \right] =$$

$$\frac{1}{(K-1)\cdot N}\sum_{n=1}^{N}\sum_{k\neq k_n}\breve{\mathcal{L}}(k,k_n)\left(\log\frac{p_i(k_n|x_n)}{p_i(k|x_n)}\right).$$

Der optimale Koeffizientenvektor Λ ist folglich Lösung des Gleichungssystems $P = Q\Lambda$,

$$\Lambda \quad = \quad Q^{-1}P, \tag{4.22}$$

mit

$$Q_{i,j} \quad = \quad \frac{1}{(K-1)\cdot N}\sum_{n=1}^{N}\sum_{k\neq k_n}\left\{\log\frac{p_i(k_n|x_n)}{p_i(k|x_n)}\right\}\left\{\log\frac{p_j(k_n|x_n)}{p_j(k|x_n)}\right\},$$
$$(i,j=1,...,M),$$

und

$$P_i \quad = \quad \frac{1}{(K-1)\cdot N}\sum_{n=1}^{N}\sum_{k\neq k_n}\breve{\mathcal{L}}(k,k_n)\left\{\log\frac{p_i(k_n|x_n)}{p_i(k|x_n)}\right\},$$
$$(i=1,...,M). \tag{4.23}$$

Man beachte, daß Q die Autokorrelationsmatrix der Diskriminantenfunktionen der vorgegebenen Verteilungsmodelle ist und nicht singulär sein darf, wenn es eine eindeutige Lösung für Λ geben soll. Der Vektor P beinhaltet den Zusammenhang zwischen den Diskriminantenfunktionen der vorgegebenen Verteilungsmodelle und der Funktion $\breve{\mathcal{L}}(k,k_n)$ der Levenshtein-Distanz.
Damit geht die auf den Trainingsdaten gemessene Wortfehleranzahl jeder Hypothese $k \neq k_n$ linear in die Koeffizienten $\lambda_1,...,\lambda_M$ ein ! Unabhängig davon optimiert das Quadratmittelkriterium nicht direkt die Fehlerrate des Klassifikators. Greift man deswegen auf die geglättete Wortfehlerrate in Abschnitt 4.5 zurück und wählt man die Indikatorfunktion S so, daß eine quadratische Zielfunktion in Λ entsteht, so erhält man die **parabolisch geglättete Fehlerrate**, die zu einer geschlossenen Lösung führt und gleichzeitig die Wortfehlerrate direkt optimiert.

4.7 Parabolisch geglättete Fehlerrate

Die Wortfehlerrate $E(\Lambda)$ (3.14) wird wie in Abschnitt 4.5 durch folgende glatte Funktion E_{PWE} approximiert:

$$E_{PWE}(\Lambda) \quad = \quad \frac{1}{\sum_{n=1}^{N}L_n}\sum_{n=1}^{N}\sum_{k\neq k_n}\mathcal{L}(k,k_n)S(k,n,\Lambda). \tag{4.24}$$

Der Beitrag jeder einzelnen Hypothese zur Gesamtfehlerrate hängt somit von der Form von S und vom Wert der Diskriminantenfunktion dieser Hypothese ab. Eine mögliche Wahl für S ist:

$$S(k,n,\Lambda) = \tilde{S}\left(\log\frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)}\right). \tag{4.25}$$

Die Funktion $\tilde{S}(x)$ ist dabei eine im Intervall $-B < x < A, A > 0, B > 0$ quadratische Funktion. Sie beinhaltet den in Abbildung 4.1 dargestellten Parabelast:

$$\tilde{S}(x) \quad = \quad \left(\frac{x+B}{A+B}\right)^2; \tag{4.26}$$

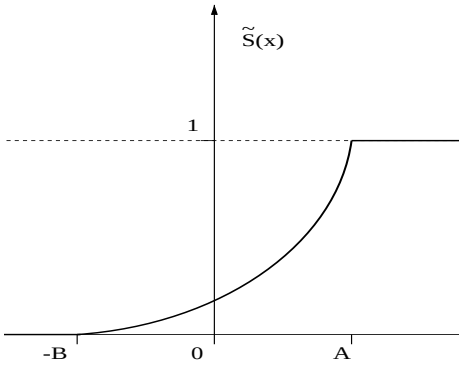

Abbildung 4.1: Parabelast als Alternative zur Sigmoidfunktion

sonst nehme $\tilde{S}$ den Wert 0 an. Die Werte A, B sollten so gewählt werden, daß

$$-B < \log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} < A \tag{4.27}$$

für jedes geeignete Tripel (k, k_n, x_n) und jedes normierte Λ

$$\sum_{j=1}^{M} \lambda_j = 1 \tag{4.28}$$

gilt. Eine Parabel hat die angenehme Eigenschaft, daß ihre Ableitung eine lineare Funktion von x ist, was schließlich eine geschlossene Lösung ermöglicht. Optimierung von $E_{PWE}(\Lambda)$, gegeben die Nebenbedingung (4.28), durch Minimierung der in Λ quadratischen Lagrangefunktion

$$L(\Lambda, \alpha) = \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) \cdot \tilde{S}\left(\log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right) + \alpha\left(\sum_{j=1}^{M} \lambda_j - 1\right) \tag{4.29}$$

führt zu folgender Matrixgleichung:

$$(\alpha, \Lambda^T)^T = BQ'^{-1}P', \text{ mit} \tag{4.30}$$

$$\begin{aligned}
Q'_{0,0} &= 0, \quad Q'_{0,j} = 1, \quad Q'_{i,0} = \frac{1}{2}(A+B)^2 \\
Q'_{i,j} &= \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) \cdot \left\{\log \frac{p_i(k_n|x_n)}{p_i(k|x_n)}\right\}\left\{\log \frac{p_j(k_n|x_n)}{p_j(k|x_n)}\right\}, \\
&\quad (i, j = 1, ..., M), \\
P'_0 &= \frac{1}{B}
\end{aligned}$$

$$P_i' = \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) \left\{ \log \frac{p_i(k_n|x_n)}{p_i(k|x_n)} \right\},$$
$$(i = 1, ..., M). \tag{4.31}$$

4.8 Vergleich mit Maximum-Likelihood und MMI

Es entsteht die Frage, ob die Modellkombination nicht auch mit Hilfe des weitverbreiteten Maximum-Likelihood-Kriteriums:

$$L(\Lambda) = \sum_{n=1}^{N} \log p_{\Lambda}(k_n, x_n) \tag{4.32}$$

optimiert werden kann. Zunächst sei an dieser Stelle auf die bereits in den Abschnitten 3.3 und 4.1 geführte Diskussion verwiesen, die im wesentlichen beinhaltet, daß das Maximum-Likelihood-Kriterium die gesehenen Daten optimal erklärt, jedoch nicht die Wortfehlerrate minimiert. Desweiteren führt die Verwendung der Verbundverteilung $p_{\Lambda}(k_n, x_n) = p_{\Lambda}(k_n|x_n) p_{\Lambda}(x_n)$ wegen der Normierungsbedingung für Wahrscheinlichkeitsverteilungen zu einer komplizierten Zielfunktion. Bei genauerer Betrachtung des Kriteriums erhalten wir:

$$L(\Lambda) = \sum_{n=1}^{N} \log \frac{\exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k_n, x_n)\right\}}{\sum_k \int_y \exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k, y)\right\} dy}. \tag{4.33}$$

Das Maximum-Likelihood-Kriterium hat den wesentlichen Nachteil, daß der Normierungsterm $\sum_k \int_y \exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k, y)\right\} dy$ in die Optimierung eingeht. Da die funktionale Form der verwendeten Basismodelle nicht eingeschränkt werden soll, ist eine geschlossene Lösung des Integrals (auch nach Differentiation bezüglich Λ) nicht möglich. Damit wäre zum einen die numerische Integration einer sehr komplexen Funktion erforderlich. Zum anderen müßte eine Summation über alle potentiellen Wortfolgen k erfolgen. Da in den Experimenten die Summe über alle Wortfolgen durch die Summe der wahrscheinlichsten Wortfolgen approximiert werden muß, entsteht hier ein größerer Approximationsfehler, als beim MWE-Kriterium.

Für das MMI-Kriterium (Maximale Transinformation, engl. 'Maximum Mutual Information') ergibt sich eine ähnliche Situation. Es lautet für die Kombination von Basismodellen auf Satzebene:

$$I(\Lambda) = \sum_{n=1}^{N} \log p_{\Lambda}(k_n|x_n) \tag{4.34}$$
$$= \sum_{n=1}^{N} \log \frac{\exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k_n|x_n)\right\}}{\sum_k \exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k|x_n)\right\}}. \tag{4.35}$$

Dem kritischen Leser wird auffallen, daß das in (4.34) dargestellte Kriterium die Äquivokation und nicht die Transinformation zwischen der Wortfolge und der akustische Beobachtung beschreibt. Die Äquivokation wird in der Informationstheorie zusammen mit Fano's Ungleichung [Cover$^+$ 1991], S.39, genutzt, um eine untere Schranke für die Fehlerrate zu bestimmen. Sie eignet sich deswegen als Optimierungskriterium für Klassifikationsaufgaben. Ist die a-priori Wahrscheinlichkeit der Wortfolge k_n gleichverteilt, so sind Äquivokation und Transinformation bis auf eine additive Konstante identisch. Aus diesem Grunde wird im Kontext der Spracherkennung häufig irreführend der Transinformationsbegriff, und nicht der Äquivokationsbegriff verwendet. Das Kriterium erfordert analog zum Maximum-Likelihood-Kriterium eine Summation über alle potentiellen Satzhypothesen k'. Damit entsteht auch hier ein größerer Approximationsfehler, als beim MWE-Kriterium.
Das Äquivokationskriterium wurde bereits als Optimierungskriterium für die Bestimmung eines worthypothesenabhängigen Sprachmodellfaktors eingesetzt [Vergyri 1997] (vgl. Abschnitt 5.2.3). Die Worthypothesen eines

Lattices werden dabei in verschiedene Konfidenzklassen eingeteilt. Jede Konfidenzklasse besitzt einen spezifischen Sprachmodellfaktor. Bei der Optimierung der spezifischen Sprachmodellfaktoren mit Hilfe des Äquivokationskriteriums konnte in [Vergyri 1997] die Wortfehlerrate jedoch weder auf den Trainingsdaten noch auf den Testdaten gesenkt werden. Auf der Basis dieses Ergebnisses wurde dort die Schlußfolgerung gezogen, die Wortfehlerrate direkt zu optimieren.

Das MWE-Kriterium

$$E_{MWE}(\Lambda) \quad = \quad \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k,k_n) \frac{p_{\Lambda}(k|x_n)^{\eta}}{\sum_{k'} p_{\Lambda}(k'|x_n)^{\eta}} \,. \tag{4.36}$$

hat damit gegenüber dem Maximum-Likelihood-Kriterium und dem Äquivokations-Kriterium einen großen Vorteil. Der Approximationsfehler der Summation über eine endliche Anzahl von Satzhypothesen ist hier klein, da alle Wahrscheinlichkeiten $p_{\Lambda}(k|x_n)$ in den Experimenten auf den Wallstreet-Journal-Daten und den Broadcast-News-Daten mit dem Exponenten $\eta = 3$ potenziert werden. Es erfordert insbesondere gegenüber dem Maximum-Likelihood-Kriterium keine numerische Integration. Hinzu kommt, daß das Kriterium die empirische Wortfehlerrate sehr genau approximiert.

Zusammenfassung:

- Das Maximum-Likelihood-Kriterium und das Äquivokations-Kriterium sind nur Ersatzkriterien für die Wortfehlerrate und außerdem schwer zu implementieren. Das MWE-Kriterium besitzt keinen dieser Nachteile.

4.9 Zusammenfassung der DMC-Theorie

Der theoretische Kern der vorliegenden Arbeit besteht aus folgenden Schwerpunkten:

A: Allgemeine log-lineare Kombination von M Basismodellen $p_j(k|x), j = 1, \ldots, M$
(Verteilung der Maximum-Entropie-Familie)

$$p_{\Lambda}(k|x) \quad = \quad \frac{\prod_{j=1}^{M} p_j(k|x)^{\lambda_j}}{\sum_{k'} \prod_{j=1}^{M} p_j(k'|x)^{\lambda_j}}$$

Die allgemeine Modellkombination kann auf akustische und Sprachmodelle spezialisiert werden. Damit wird im weiteren Verlauf der Arbeit auf Wallstreet-Journal-Daten und auf Broadcast-News-Daten experimentiert. Eine Äquivalenzumformung ergibt für akustische Modelle $p_i(x|k), i = 1, \ldots, I$ und Sprachmodelle $p_j(k), j = 1, \ldots, J$

$$p_{\Lambda}(k|x) \quad = \quad \frac{\prod_i p_i(x|k)^{\lambda_i} \prod_j p_j(k)^{\lambda_j}}{\sum_{k'} \prod_i p_i(x|k')^{\lambda_i} \prod_j p_j(k')^{\lambda_j}} \,.$$

B: Diskriminatives Training der freien Koeffizienten $\Lambda = (\lambda_1, \ldots, \lambda_M)^T$ der Modellkombination auf N Trainingssätzen $n = 1, \ldots, N$, die in Form der akustischen Beobachtung x_n, der korrekten Wortfolge k_n und der rivalisierenden Wortfolgen $k \neq k_n$ vorliegen. Diskriminatives Trainingskriterium:

$$E(\Lambda) \quad = \quad \sum_{n=1}^{N} f(x_n, k_n, \Lambda)$$

B.1: MCE-Kriterium: $f(x_n, k_n, \Lambda) = \ell(x_n, k_n, \Lambda)$
Iterative Optimierung der Satzfehlerrate.

B.2: MWE-Kriterium: $f(x_n, k_n, \Lambda) = \sum_{k \neq k_n} \mathcal{L}(k, k_n) \frac{p_{\Lambda}(k|x_n)^{\eta}}{\sum_{k'} p_{\Lambda}(k'|x_n)^{\eta}}$

Iterative Optimierung der Wortfehlerrate.

B.3: MSE-Kriterium: $f(x_n, k_n, \Lambda) = \sum_{k \neq k_n} \left(\log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} - \breve{\mathcal{L}}(k, k_n) \right)^2$

Minimierung des Abstandes zwischen Diskriminantenfunktion und monotoner Funktion der Wortfehlerrate $\breve{\mathcal{L}}$ führt zu geschlossener Lösung.

B.4: Parabelansatz: $f(x_n, k_n, \Lambda) = \sum_{k \neq k_n} \mathcal{L}(k, k_n) \left(\frac{\log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} + B}{A + B} \right)^2$.

Synthese aus B.2 und B.3, d.h. quadratische Approximation von B.2 . Optimierung der Wortfehlerrate und geschlossene Lösung.

C: Die Lösungsgleichungen sind strukturell ähnlich. In allen betrachteten Verfahren (B.1-B.4) werden die Parameter $\Lambda = (\lambda_1, \ldots, \lambda_M)^T$ durch gewichtete Korrelationen zwischen einer geeigneten Funktion der Fehlerrate und den Diskriminantenfunktionen der Basismodelle berechnet.

C.1: Allgemeine Struktur der iterativen Gradienten-Abstiegs-Verfahren:

$$\triangle\lambda_j \quad = \quad \epsilon \sum_{n=1}^{N} \sum_{k \neq k_n} w(k, k_n, x_n, \Lambda) \cdot \tilde{\mathcal{L}}(k, n, \Lambda) \cdot \log \frac{p_j(k_n|x_n)}{p_j(k|x_n)}\,, \tag{4.37}$$

wobei w eine geeignete Indikatorfunktion und $\tilde{\mathcal{L}}$ eine geeignete Funktion der Levenshtein-Distanz ist.

C.2: Die implizite Abhängigkeit von Λ in den iterativen Verfahren wird in den geschlossenen Lösungsverfahren durch die explizite Einbeziehung der Korrelationen der Diskriminantenfunktionen aller Basismodelle kompensiert. Die allgemeine Struktur der Lösungsgleichungen lautet hier:

$$\lambda_j \quad = \quad \sum_{i=1}^{M} C_{ij} \sum_{n=1}^{N} \sum_{k \neq k_n} \cdot \breve{\mathcal{L}}(k, k_n) \cdot \log \frac{p_i(k_n|x_n)}{p_i(k|x_n)}\,, \tag{4.38}$$

wobei $C = Q^{-1}$ die zu dem einzelnen Verfahren zugehörige inverse gewichtete Autokorrelationsmatrix der Diskriminantenfunktionen der Basismodelle darstellt. $\breve{\mathcal{L}}$ ist eine monotone Funktion der Wortfehlerrate.

Kapitel 5

Diskussion Verwandter Ansätze

Dieses Kapitel setzt sich zum einen mit der Verwandtschaft der **Theorie der Linearen Klassifikatoren** zu den in Kapitel 4 betrachteten Optimierungsverfahren auseinander und zum anderen mit der sogenannten **Unified Stochastic Engine**, die ebenfalls in Verbindung mit DMC zu sehen ist.

5.1 Basismodelle als Merkmale

In diesem Abschnitt wird das Modellkombinationsproblem mit Hilfe der Theorie der linearen Klassifikatoren [Anderson$^+$ 1962] bearbeitet. Es seien N Trainingsbeobachtungen $x_n, n = 1, \ldots, N$ gegeben. Diese werden in K Klassen $k = 1, \ldots, K$ klassifiziert. Für jede Trainingsbeobachtung x_n bezeichnet $k = k_n$ die korrekte Klasse und $k \neq k_n$ eine der $K-1$ rivalisierenden Klassen. Außerdem seien die M Basismodelle $p_j(k|x_n), j = 1, \ldots, M$ gegeben. Zunächst kann aus den Bewertungen der verschiedenen Basismodelle $p_j(k|x_n)$ für die Satzhypothese k bei gegebener Äußerung x_n folgender Merkmalsvektor $\mathbf{Y}_{k,x_n}$ konstruiert werden:

$$\mathbf{Y}_{k,x_n} \quad = \quad (\log p_1(k|x_n), \ldots, \log p_M(k|x_n))^T. \tag{5.1}$$

Für die weiteren Betrachtungen wird die Klasseneinteilung von $\mathbf{Y}_{k,x_n}$ in zwei Klassen ω_c, ω_r notwendig: $\mathbf{Y}_{k,x_n}$ gehöre zur Klasse ω_c, wenn k die korrekte Klassenzuordnung für die Äußerung x_n ist, $k = k_n$. Anderenfalls gehöre $\mathbf{Y}_{k,x_n}$ zu der zweiten Klasse ω_r (Zuordnung zu einer rivalisierenden Klasse $k \neq k_n$). Damit wurde ein 2-Klassenproblem konstruiert und die Theorie der linearen Klassifikatoren kann angewendet werden, wenn $\mathbf{Y}_{k,x_n}$ normalverteilt oder hochdimensional ist. Diese Einschränkung wird im folgenden Abschnitt begründet. Für die weiteren Betrachtungen wird angenommen, daß diese Bedingung erfüllt sei.

5.1.1 Lineare Klassifikatoren

Das Design eines optimalen linearen Klassifikators $h(\mathbf{Y}_{k,x_n}) = \Lambda^T \mathbf{Y}_{k,x_n}$, der M-dimensionale Merkmalsvektoren $\mathbf{Y}_{k,x_n}$ in Klassen ω_c, ω_r einteilt, wird in [Fukunaga 1990], S. 131, ausführlich beschrieben. Der Vektor $\mathbf{Y}_{k,x_n}$ wird entweder nach ω_c oder nach ω_r klassifiziert, abhängig davon, ob $h(\mathbf{Y}_{k,x_n}) \leq \theta$ oder $h(\mathbf{Y}_{k,x_n}) > \theta$ gilt.

Um ein optimales Klassifikationsergebnis zu erreichen, müssen die Parameter Λ, θ so eingestellt werden, daß die Fehlerrate im projizierten eindimensionalen h-Raum minimal ist. Ist nun $\mathbf{Y}_{k,x_n}$ normal verteilt, dann ist auch $h(\mathbf{Y}_{k,x_n})$ normal verteilt. Ist $\mathbf{Y}_{k,x_n}$ nicht normal verteilt, ist aber $M \gg 1$, dann ist $h(\mathbf{Y}_{k,x_n})$ ähnlich zu einer Normalverteilung, da $h(\mathbf{Y}_{k,x_n})$ eine Summe von M Zufallszahlen ist und somit unter bestimmten Bedingungen der zentrale Grenzwertsatz [Fisz 1989] angewendet werden kann. Eine sehr strenge Bedingung für die Anwendbarkeit des zentralen Grenzwertsatzes ist zum Beispiel, daß alle M Elemente des Merkmalsvektors $\mathbf{Y}_{k,x_n}$ die gleiche Verteilung besitzen.
Ist nun $h(\mathbf{Y}_{k,x_n})$ normal verteilt, dann kann die Separabilität zwischen den beiden Klassen durch eine Abstandsfunktion f der Mittelwerte η_c, η_r und der Varianzen σ_c, σ_r im h-Raum gemessen werden. Im originalen $\mathbf{Y}_{k,x_n}$-Raum werden die Mittelwertvektoren mit μ_c, μ_r bezeichnet und die Kovarianzmatrizen mit Σ_c, Σ_r. Der optimale Vektor Λ wird dann durch [Fukunaga 1990], S. 134 :

$$\Lambda \quad = \quad [w_c\Sigma_c + w_r\Sigma_r]^{-1} (\mu_r - \mu_c) \tag{5.2}$$

bestimmt, wobei die Gewichte w_c und w_r von der funktionalen Form der Abstandsfunktion f abhängen.
In [Fukunaga 1990], S. 135, wird für die Funktion f das Fisher-Kriterium:

$$f(\eta_c, \sigma_c, \eta_r, \sigma_r) \quad = \quad \frac{(\eta_c - \eta_r)^2}{\sigma_c^2 + \sigma_r^2} \tag{5.3}$$

eingesetzt. Daraus folgt $w_c = w_r = \frac{1}{2}$ und $\Lambda = 2\left[\Sigma_c + \Sigma_r\right]^{-1}(\mu_r - \mu_c)$.

5.1.2 Relation zu DMC

Durch die Optimierung der Paramter Λ bezüglich der Klassifikationsfehlerrate im h-Raum ist zu erwarten, daß die mittlere Differenz

$$F(\Lambda) \quad = \quad \sum_{n=1}^{N} \sum_{k \neq k_n} h(\mathbf{Y}_{k_n, x_n}) - h(\mathbf{Y}_{k, x_n}) \tag{5.4}$$

steigt. Für diese Differenz erhalten wir nach Umformung:

$$\begin{aligned} F(\Lambda) &= \sum_{n=1}^{N} \sum_{k \neq k_n} h(\mathbf{Y}_{k_n, x_n}) - h(\mathbf{Y}_{k, x_n}) && (5.5) \\ &= \sum_{n=1}^{N} \sum_{k \neq k_n} \sum_{j=1}^{M} \lambda_j \left(\log p_j(k_n|x_n) - \log p_j(k|x_n)\right) \\ &= \sum_{n=1}^{N} \sum_{k \neq k_n} \log \frac{p_{\Lambda}(k_n|x_n)}{p_{\Lambda}(k|x_n)}\,. && (5.6) \end{aligned}$$

Die Vergrößerung der mittleren Differenz $F(\Lambda)$ führt damit offensichtlich zu einer Optimierung der Modellkombination. Für Λ ergibt sich:

$$\begin{aligned} \Lambda &= 2 \cdot \left[\Sigma_c + \Sigma_r\right]^{-1}(\mu_c - \mu_r) \\ \mu_{ri} &= \frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} \log p_i(k|x_n) \\ \Sigma_{rij} &= \frac{1}{(K-1) \cdot N} \sum_{n=1}^{N} \sum_{k \neq k_n} (\log p_i(k|x_n) - \mu_{ri})(\log p_j(k|x_n) - \mu_{rj}) \\ \mu_{ci} &= \frac{1}{N} \sum_{n=1}^{N} \log p_i(k_n|x_n) \\ \Sigma_{cij} &= \frac{1}{N} \sum_{n=1}^{N} (\log p_i(k_n|x_n) - \mu_{ci})(\log p_j(k_n|x_n) - \mu_{cj}). \end{aligned} \tag{5.7}$$

Man beachte die Ähnlichkeit der Struktur der Gleichungen zu den Lösungsgleichungen des Quadratmittelansatzes (4.22) und der parabolisch geglätteten empirischen Fehlerrate (4.30). Die aus der Theorie der linearen Klassifikatoren erhaltene Lösungsgleichung für den Vektor Λ enthält analog zu diesen beiden Lösungen die Korrelation zwischen den Diskriminantenfunktionen der Basismodelle. Sie berücksichtigt jedoch nicht die Wortfehlerrate der einzelnen Satzhypothesen.

5.2 Verallgemeinerte Modellkombination

In diesem Abschnitt wird die Verwandtschaft zwischen DMC, der Unified Stochastic Engine (USE) [Huang$^+$ 1993] und dem 'akustisch sensitiven Sprachmodell' [Vergyri 1997] analysiert. Dazu ist es zunächst erforderlich, eine **verallgemeinerte Modellkombination** in Basismodelle einzuführen, die dann auf die drei genannten Ansätze spezialisiert werden können. Die verallgemeinerte Modellkombination wird wie folgt angesetzt:

$$p(k|x) = \frac{\prod_{j=1}^{M} p_j(k|x)^{\lambda_j(k,x)}}{\sum_{k'} \prod_{j=1}^{M} p_j(k'|x)^{\lambda_j(k',x)}} . \tag{5.8}$$

Der Ansatz (5.8) wird in den folgenden Abschnitten verwendet, um die Beziehungen zwischen dem Sprachmodellfaktor, der 'Unified Stochastic Engine (USE)' und DMC zu analysieren. Die verallgemeinerte Modellkombination entsteht zum einen aus der log-linearen Kombination von beliebigen Basismodellen, wie sie in DMC verwendet wird. Zum anderen werden analog zu USE Exponenten λ_j verwendet, die von der akustischen Beobachtung x und der betrachteten Klasse k abhängig sind. Die endgültige Form der Modellverteilung p wird durch die Wahl der Basismodelle p_j und die Vereinfachungen der Funktionen λ_j bestimmt. Durch Variierung der Funktion λ_j kann die in der vorliegenden Arbeit behandelte log-lineare Modellkombination mit USE und dem akustisch sensitiven Sprachmodell in Beziehung gesetzt werden. Gemeinsamkeiten wie auch wesentliche Unterschiede beider Ansätze werden damit deutlich. Zunächst wird auf die einfachste Spezialisierung, die Balancierung des Einflusses des akustischen Modells und des Sprachmodells, eingegangen.

5.2.1 Spezialisierung auf den Sprachmodellfaktor

Setzt man in (5.8) die Anzahl der Basismodelle auf $M = 2$, verwendet man als erstes Basismodell das Sprachmodell $p_1(k|x) = p_{LM}(k)$ und als zweites Basismodell das akustische Modell $p_2(k|x) = \frac{p_{AM}(x|k)}{\sum_{k'} p_{AM}(x|k')}$ und nehmen die λ_j folgende Werte an: $\lambda_1 = \lambda, \lambda_2 = 1$, so erhält man als Gesamtverteilung:

$$p(k|x) = \frac{p_1(k|x)^{\lambda_1} p_2(k|x)^{\lambda_2}}{\sum_{k'} p_1(k'|x)^{\lambda_1} p_2(k'|x)^{\lambda_2}} \tag{5.9}$$

$$= \frac{p_{LM}(k)^{\lambda} p_{AM}(x|k)}{\sum_{k'} p_{LM}(k')^{\lambda} p_{AM}(x|k')} \tag{5.10}$$

Der Wert λ wird **Sprachmodellfaktor** genannt (vgl. Abschnitt 4.1). Er dient der optimalen Balancierung des Einflusses von Sprachmodell und akustischem Modell bei der Suche nach der gesprochenen Wortfolge.
In [Bahl$^+$ 1980] und [Lee 1989] wird berichtet, daß durch die künstliche Verstärkung des Einflusses des Sprachmodells auf die Entscheidungsregel die Anzahl der durch den Erkenner fehlerhaft eingefügten Wörter reduziert werden kann. Diese Modifikation ist inzwischen weltweit zum Standard geworden.

5.2.2 Spezialisierung auf USE (Unified Stochastic Engine)

In [Huang$^+$ 1993] wird die sogenannte **Unified Stochastic Engine** eingeführt. Ziel ist dabei, die Unabhängigkeitsannahme zwischen akustischem Modell und Sprachmodell aufzuweichen und ein gemeinsames Schätzverfahren für akustische und Sprachmodellwahrscheinlichkeiten zu entwickeln. Dabei wird jedoch wie in (5.10) die Aufteilung in genau ein akustisches Modell und genau ein Sprachmodell beibehalten. Eine log-lineare Kombination mehrerer akustischer und Sprachmodelle wird nicht betrachtet. Das Neue an USE ist die Zerlegung der Sprachmodellwahrscheinlichkeit $p_{LM}(k)$ in ein Produkt über die einzelnen Wörter der Wortfolge $k = [w_1, \ldots, w_L] = w_1^L$. Für die Gesamtverteilung ergibt sich dann:

$$p(w_1^L|x) = \frac{p_{AM}(x|w_1^L) \cdot \prod_{i=1}^{L} p_{LM}(w_i|w_1^{i-1})^{\lambda_i(x,w_1^i)}}{\sum_{v_1^{L'}} p_{AM}(x|v_1^{L'}) \cdot \prod_{i=1}^{L'} p_{LM}(v_i|v_1^{i-1})^{\lambda_i(x,v_1^i)}.}$$

Die freien Parameter Λ dieser Verteilung werden bezüglich der Satzfehlerrate des Klassifikators auf den Trainingsdaten optimiert. Aufgrund der unzureichenden Trainingsdatenmenge zur Optimierung der Funktionen λ_j wird in [Huang$^+$ 1993] in den zugehörigen Experimenten die Abhängigkeit von der akustischen Beobachtung x weggelassen, d.h. $\lambda_i(x, w_1^i) = \lambda_i(w_1^i)$.

5.2.3 Spezialisierung auf ein akustisch sensitives Sprachmodell

Die Grundidee eines akustisch sensitiven Sprachmodells [Vergyri 1997] besteht darin, das Sprachmodell von der gesprochenen akustischen Äußerung x und dem tatsächlich im Spracherkenner eingesetzten akustischen Modell abhängig zu machen. Das Sprachmodell sollte damit in der Lage sein, eventuelle Schwächen des akustischen Modells zu kompensieren. Wegen des komplexen akustischen Raumes kann eine direkte Abhängigkeit der Wahrscheinlichkeit der Wortfolge k von der Akustik x nicht modelliert werden [Vergyri 1997]. Als Vereinfachung wird deswegen ein worthypothesenabhängiger Sprachmodellfaktor angesetzt, der den Einfluß des akustischen Modells und des Sprachmodells situationsabhängig modifizieren soll. Dabei wird die Satzhypothese k in die Folge der Worthypothesen $k = [wh_1, \ldots, wh_L] = wh_1^L$ zerlegt, wobei die Worthypothesen wh_i durch das Wort und die Position der Hypothese wh_i im Lattice definiert sind. Analog wird die Äußerung in die zugehörigen Äußerungsabschnitte aufgeteilt: $x = (x_1, \ldots, x_L)$. Formal wird damit Gleichung (5.8) wie folgt spezialisiert:

$$p(k|x) = \frac{\prod_{l=1}^{L} p_{LM}(wh_l|wh_1^{l-1})^{\lambda_{LM,c(wh_l,k,x)}} \cdot p_{AM}(x_l|wh_l)^{\lambda_{AM,c(wh_l,k,x)}}}{\sum_{k'=wh'^{L'}_1} \prod_{l=1}^{L'} p_{LM}(wh'_l|wh'^{l-1}_1)^{\lambda_{LM,c(wh'_l,k',x)}} p_{AM}(x_l|wh'_l)^{\lambda_{AM,c(wh'_l,k',x)}}} \tag{5.11}$$

Dabei wird die Gewichtung des Sprachmodells und des akustischen Modells von einer Klasseneinteilung $c(wh_l, k, x)$ der Worthypothesen wh_l abhängig gemacht, die auf der Basis von Konfidenzbewertungen durchgeführt wird.

5.2.4 Spezialisierung auf DMC

Verwendet man konstante Werte $\lambda_j = const.$, so entsteht die log-lineare Modellkombination

$$p(k|x) \quad = \quad \frac{\prod_{j=1}^{M} p_j(k|x)^{\lambda_j}}{\sum_{k'} \prod_{j=1}^{M} p_j(k'|x)^{\lambda_j}} \,,$$

die im Rahmen von DMC verwendet wird. Diese wurde in Abschnitt 3.2 und Kapitel 4 ausführlich betrachtet. Die Vielfalt der Anwendungsmöglichkeiten dieser allgemeinen Form der Modellkombination wird im Kapitel 9 deutlich gemacht.

5.3 Zusammenfassung

In diesem Kapitel wurde die Verwandtschaft von DMC zu anderen bekannten Ansätzen der Mustererkennung diskutiert.
Bei der Auseinandersetzung mit der Theorie der linearen Klassifikatoren wurden folgende Schlußfolgerungen gezogen:

- Die Bewertungen der Basismodelle für eine Satzhypothese können in einen Merkmalsvektor überführt werden, so daß die Theorie linearer Klassifikatoren anwendbar erscheint.

- Die Theorie linearer Klassifikatoren setzt voraus, daß die Linearkombination der Bewertungen $\log p_j(k|x)$ der Basismodelle normalverteilt ist. Dadurch kann die Minimierung der Klassifikationsfehlerrate auf die Minimierung des Überlapps zweier eindimensionaler Gaußdichten zurückgeführt werden.

- Die Lösungsgleichungen berücksichtigen die Korrelation der Diskriminantenfunktionen der Basismodelle $p_j(k|x)$.
- Die Lösungsgleichungen berücksichtigen nicht die Wortfehlerraten der Satzhypothesen.

Damit scheint die Theorie der linearen Klassifikatoren für die Bildung einer optimalen Modellkombination in Spracherkennungssystemen mit großem Vokabular nicht geeignet zu sein.
Die Verwandtschaft der Diskriminativen Modellkombination mit USE und einem akustisch sensitiven Sprachmodell wurde analysiert. Dabei ergaben sich folgende Gemeinsamkeiten:

- log-lineare Strukturierung der Sprachmodellverteilung in USE, log-lineare Strukturierung in Wortklassen für das akustisch sensitive Sprachmodell und log-lineare Strukturierung der Gesamtverteilung in DMC,
- diskriminative Optimierung der freien Parameter der log-linearen Kombination

und folgende Unterschiede:

- In USE wird eine Strukturierung in genau ein akustisches Modell und eine wortspezifische log-lineare Kombination von M-gramm Sprachmodellwahrscheinlichkeiten vollzogen. Bei der Bildung des akustisch sensitiven Sprachmodells wird ebenso von einem vorgegebenen Sprachmodell und einem vorgegebenen akustischen Modell ausgegangen und eine Strukturierung in wortklassenabhängige Teilmodelle vorgenommen. Bei der Verwendung von DMC wird eine Dekomposition in beliebige Basismodelle vorgenommen. Die Auswahl und konkrete Gestaltung der Modelle ist bei DMC nicht beschränkt.
- USE optimiert beim diskriminativen Training die Satzfehlerrate, obwohl in [Huang$^+$ 1993] als Erfolgsmaß die Wortfehlerrate verwendet wird.
  DMC optimiert die Wortfehlerrate direkt.

Teil III

Exemplarische Optimierung eines Spracherkenners mit DMC

Kapitel 6

Entwicklung von DMC auf der WSJ0-Datenbasis

6.1 Die Wall-Street-Journal-Aufgabe (WSJ-Aufgabe)

Für die Entwicklung des DMC-Algorithmus wurde die im Jahre 1992 entstandene WSJ0-Datenbasis [Paul$^+$ 1992] eingesetzt. Die WSJ0-Datenbasis besteht aus vorgelesenen Artikeln des 'Wall Street Journals', einer bekannten amerikanischen Tageszeitung. Die DMC-Software wurde auf dem männlichen Teil der akustischen Trainings- und Testdaten entwickelt. Die Trainingsdatenbasis ist sprecherunabhängig und besteht aus 3608 Sätzen von 42 verschiedenen Sprechern, das sind ca. 6 h gesprochene Sprache. Die verwendeten sprecherunabhängigen Testdatenbasen Development'92 und Eval'92 werden in Tabelle 6.1 beschrieben. Für das Sprachmodelltraining wurde die aus ca. 37 Millionen Wörtern bestehende WSJ0-Textdatenbasis verwendet.

Tabelle 6.1: Struktur des männlichen Teils der ARPA WSJ0-Testdatenbasis, die im Jahre 1992 für die weltweite Evaluation von Spracherkennern eingesetzt wurde

Datenbasis	offizielle Bezeichnung	Sprecher	Sätze	Wörter
Development'92	si_dt_05'92-m	6	247	4067
Eval'92	si_et_05'92-m	4	163	2653

6.2 Das Philips-Baseline-System

Der folgende Abschnitt faßt den Stand des Hochleistungsspracherkenners der Philips Forschungslaboratorien Aachen im Jahre 1993 zusammen [Ney 1990], [Ney$^+$ 1992], [Ney$^+$ 1994], [Steinbiss$^+$ 1993], [Aubert$^+$ 1994]. Das im folgenden beschriebene WSJ0-System war Ausgangspunkt der vorliegenden Arbeit. Die Merkmalsextraktion, akustische Modellierung und Suche in dem verwandten RWTH-System wurden bereits detailiert in den Arbeiten [Welling 1998], [Beulen 1999a], [Ortmanns 1998] beschrieben. Aus diesem Grunde wird im folgenden nur auf wesentliche Charakteristika des Spracherkenners eingegangen.

6.2.1 Merkmalsextraktion

Das akustische Signal wird tiefpaßgefiltert und mit einer Abtastrate von 16 kHz digitalisiert. Die darauf folgende Filterbankanalyse ist ein Standardverfahren ([Rabiner$^+$ 1993], S. 158, [Ney 1990]) und wird in einem zeitlichen Abstand von 10 ms auf dem digitalisierten Sprachsignal durchgeführt:

- Bildung eines Hamming-Fensters mit einer Länge von 25ms,
- Anwendung einer 512-Punkte-FFT-Signaltransformation,

- Berechnung des Leistungsdichtespektrums,

- Extraktion eines Filterbankvektors $\vec{z}_t \in \mathbf{R}^{30}$ aus dem Leistungsdichtespektrum zum Zeitpunkt t mit 30 auf der mel-skalierten Frequenzachse äquidistanten Mittenfrequenzen zwischen 200 Hz und 6,4KHz und einem abgeschnittenen $sin(x)/x$ Bandfilter.

- Logarithmierung der Komponenten von $\vec{z}_t$.

- Bildung des Mittelwertes der logarithmierten Komponenten von $\vec{z}_t$. Dieser Mittelwert wird als Pseudoenergie e_t bezeichnet [Häb 1999].

- Bildung eines Merkmalsvektors $\vec{y}_t \in \mathbf{R}^{30}$ durch Normierung mit der Pseudoenergie $y_{t,i} = \log z_{t,i} - e_t, i = 1, \ldots, 30$.

Um die Variabilität des akustischen Kanals zu kompensieren, wird jeder Vektor $\vec{y}_t$ mit dem Langzeitspektrum

$$\overline{y}_t = \frac{1}{2\triangle T + 1} \sum_{\tau=-\triangle T}^{\triangle T} y_{t-\tau} \tag{6.1}$$

normiert ($\triangle T = 300$). Der normierte Vektor lautet dann $\tilde{y}_t = \vec{y}_t - \overline{y}_t$. Um die temporale Struktur des Sprachsignals einzubeziehen, werden zu jeder der 30 Vektorkomponenten noch der Anstieg (symmetrische Differenz erster Ordnung) und die Krümmung (symmetrische Differenz zweiter Ordnung) in der lokalen zeitlichen Umgebung des aktuellen Vektors gebildet. Der letztendlich verwendete Merkmalsvektor
$\vec{\tilde{x}} \in \mathbf{R}^{63}$ besteht aus 30 spektralen Intensitäten, 15 Differenzen 1. Ordnung

$$\triangle \tilde{y}_{t,i} = \frac{1}{2} \left(\tilde{y}_{t+2,i} - \tilde{y}_{t-2,i} \right), \tag{6.2}$$

15 Differenzen 2. Ordnung

$$\triangle\triangle \tilde{y}_{t,i} = \triangle(\triangle \tilde{y}_{t,i}) \tag{6.3}$$

sowie aus 3 Komponenten, die die Energie und deren Anstieg sowie Krümmung beinhalten:

$$\vec{\tilde{x}} = \begin{bmatrix} \tilde{y}_{t,1} \\ \tilde{y}_{t,2} \\ \vdots \\ \tilde{y}_{t,30} \\ \frac{1}{2}(\triangle y_{t,1} + \triangle y_{t,2}) \\ \frac{1}{2}(\triangle y_{t,3} + \triangle y_{t,4}) \\ \vdots \\ \frac{1}{2}(\triangle y_{t,29} + \triangle y_{t,30}) \\ \frac{1}{2}(\triangle\triangle y_{t,1} + \triangle\triangle y_{t,2}) \\ \frac{1}{2}(\triangle\triangle y_{t,3} + \triangle\triangle y_{t,4}) \\ \vdots \\ \frac{1}{2}(\triangle\triangle y_{t,29} + \triangle\triangle y_{t,30}) \\ e_t \\ \triangle e_t \\ \triangle\triangle e_t \end{bmatrix} \tag{6.4}$$

Dieser Vektor wird anschließend einer 'Linearen Diskriminantenanalyse (LDA)' unterzogen [Häb$^+$ 1992], [Häb$^+$ 1993]. Die LDA-Transformation ist ein bekanntes Verfahren in der statistischen Mustererkennung, welches die Diskrimination zwischen verschiedenen Klassen in einem hochdimensionalen Vektorraum verbessert. Die Grundidee der LDA liegt in der Berechnung einer linearen Transformation, sodaß ein geeignetes Separabilitätsmaß zwischen den zu unterscheidenden Klassen (HMM-Zustände, vgl. Abschnitt 6.2.2) maximiert wird. Mit Hilfe der LDA-Matrix werden jeweils 3 aufeinander folgende 63-komponentige Merkmalsvektoren in einen 35-komponentigen LDA-Merkmalsvektor transformiert. Dadurch wird längerreichweitiger akustischer Kontext in die LDA-Transformation einbezogen. Als Resultat entsteht die in Abbildung 2.1 dargestellte Vektorfolge $\vec{x}_1, \ldots, \vec{x}_T$.

Eine ausführliche Analyse von Verfahren für die Merkmalsextraktion in einem ähnlichen Spracherkennungssystem mit großem Vokabular wird in [Welling 1998] durchgeführt.

6.2.2 Akustische Modellierung

Die Beziehung zwischen der erhaltenen Merkmalsvektorfolge $\vec{x}_1, \ldots, \vec{x}_T$ und der Wortfolge $w_1, \ldots, w_L$ wird über das akustische Modell $p(\vec{x}_1, \ldots, \vec{x}_T | w_1, \ldots, w_L)$ hergestellt (Abb. 2.1). Dabei wird die Wortfolge $w_1, \ldots, w_L$ mit Hilfe des Aussprachelexikons in die Phonemfolge $\phi_1, \ldots, \phi_H$ umgewandelt und anschließend mit Hilfe der Zustände $s_1, \ldots, s_T$ des Hidden-Markoff-Modells die Ähnlichkeit zwischen Phonemfolge und Merkmalsvektorfolge ermittelt. Zur kompakten Darstellung der genannten Folgen wird eine kürzere Schreibweise eingeführt:

- Merkmalsvektorfolge $\vec{x}_1^T = \vec{x}_1, \ldots, \vec{x}_T$
- Zustandsfolge $s_1^T = s_1, \ldots, s_T$
- Phonemfolge $\phi_1^H = \phi_1, \ldots, \phi_H$
- Wortfolge $w_1^L = w_1, \ldots, w_L$.

Das Aussprachelexikon [Aubert$^+$ 1994] basiert auf dem Dragon-Lexikon, welches im Jahre 1992 vom U.S. amerikanischen Unternehmen Dragon Systems Inc. weltweit zur Verfügung gestellt wurde. Das Dragon-Lexikon umfaßt alle 29000 Wörter, die in der WSJ0-Trainingsdatenbasis wie auch in der zugehörigen WSJ0-Testdatenbasis auftreten. Dieses Lexikon wurde durch manuelle Eingriffe verbessert. Das in der vorliegenden Arbeit verwendete Lexikon basiert auf einem Phoneminventar mit 44 Elementen und stellt im Mittel 1.1 Aussprachevarianten pro Wort zur Verfügung. Mit Hilfe des Aussprachelexikons wird schließlich die Wortfolge w_1^L des betrachteten Satzes in eine Phonemfolge ϕ_1^H umgewandelt.

Um die starke Koartikulation in der englischen Sprache zu modellieren, werden kontextabhängige Phoneme verwendet. Dazu werden die Phoneme des Phoneminventars zusätzlich vom Vorgängerphonem und vom Nachfolgerphonem in der Phonemfolge des betrachteten Wortes abhängig gemacht. Dadurch entstehen aus den 44 Phonemen (Monophone) potentiell 44^3 sogenannte **Triphone**. Da der Wortschatz des Trainings- und Erkennungslexikons beschränkt ist, werden nicht alle potentiellen Triphone benötigt. Bei einer begrenzten Menge von Trainingsmaterial, wie bei der WSJ0-Aufgabe, kann es Triphone geben, die für die Konstruktion der Triphonfolge der zu erkennenden Wörter benötigt werden, die jedoch im Trainingsmaterial nicht oder sehr selten vorkommen. Jedes un- oder untertrainierte Triphon wird auf das zugehörige Monophon abgebildet. Diesen Mechanismus nennt man **Monophon-Backing-Off**. Ein Monophon-Backing-Off wird dann durchgeführt, wenn das Triphon weniger als 150-mal im Trainingskorpus auftritt. Im WSJ0-Korpus treten 736 verschiedene Triphone mindestens 150 mal auf. Somit werden insgesamt 780 (736+44) Phonemmodelle trainiert. Um selten gesehene oder ungesehene Triphone besser modellieren zu können, setzt man als bessere Alternative zum Monophon-Backing-Off ein Cluster-Verfahren ein, das auf Entscheidungsbäumen beruht [Odell 1995], [Beulen 1999a]. Dieses Verfahren stand für das hier beschriebene Baseline-System nicht zur Verfügung, wurde aber im Verlaufe der vorliegenden Arbeit eingeführt (vgl. Abschnitt 6.4). Im Ergebnis dieser phonetischen Modellierung wird die Phonemfolge ϕ_1^H in eine Triphonfolge ξ_1^H umgewandelt. Formal gesehen wird die Berechnung von $p(\vec{x}_1^T|w_1^L)$ durch den Phonemansatz auf die Berechnung von $p(\vec{x}_1^T|\xi_1^H)$ zurückgeführt.

Für jedes kontextabhängige Phonem ξ_i wird ein Hidden-Markoff-Modell [Jelinek 1976] trainiert. Eine detailierte Beschreibung des Trainingsalgorithmus in einem ähnlichen Spracherkenner findet man in [Welling 1998] und in [Beulen 1999a].
Wir betrachten zunächst die Wahrscheinlichkeit $p(\vec{x}_1^T, s_1^T)$ für das Verbundereignis des Auftretens einer Merkmalsvektorfolge $\vec{x}_1^T$ und einer Zustandsfolge s_1^T. Sie berechnet sich entsprechend dem HMM zu

$$p(\vec{x}_1^T, s_1^T) \quad = \quad q(s_0) \prod_{t=1}^{T} q(s_t|s_{t-1}) \cdot p(\vec{x}_t|s_t). \tag{6.5}$$

Die Topologie des HMM's ist phonemunabhängig und besteht aus einer transienten linearen Markoffkette mit 6 Zuständen, wobei die ersten beiden, die mittleren beiden und die letzten beiden Zustände die gleiche Emissionsverteilung verwenden. Die Übergangswahrscheinlichkeiten $q(s_t|s_{t-1})$ sind phonemunabhängig und zustandsunabhänig. Sie erlauben nur 3 Übergangsarten:

Loop: $q(s|s) \neq 0$, d.h. Verweilen im Zustand,

Next: $q(s|s-1) \neq 0$, d.h. Übergang in den Folgezustand,

Skip: $q(s|s-2) \neq 0$, d.h. Sprung in den übernächsten Zustand.

Die in den Zuständen enthaltenen Emissionsverteilungsdichten $p(\vec{x}|s)$ sind Mischungen aus Laplace-Dichten:

$$p(\vec{x}|s) \quad = \quad \sum_{i=1}^{I} w_i b_i(\vec{x}|s), \tag{6.6}$$

wobei gilt:

$$0 \leq w_i \leq 1, \quad i = 1, \ldots, I, \tag{6.7}$$

$$\sum_{i=1}^{I} w_i = 1 \tag{6.8}$$

und

$$b_i(\vec{x}|s) \quad = \quad \frac{1}{2^D \prod_{d=1}^{D} v_d} \exp\left\{ -\sum_{d=1}^{D} \frac{|x_d - \mu_{s,i,d}|}{v_d} \right\} \tag{6.9}$$

ist. Der wesentliche Vorteil der Mischverteilungen besteht zum einen darin, daß die unbekannte 'wahre' Verteilungsform durch die Summe von Einzelverteilungen approximiert werden kann. Zum anderen läßt sich die Aussprachevariabilität besser erfassen.
Sei $Z(\xi_1^H)$ die Menge aller Zustandsfolgen, die die Triphonfolge ξ_1^H repräsentieren. Die Wahrscheinlichkeit der Merkmalsfolge bei vorgegebener Phonemfolge kann nun mit Hilfe des Hidden-Markoff-Models (6.5) wie folgt berechnet werden:

$$p(\vec{x}_1^T|\xi_1^H) \quad = \quad \sum_{s_1^T \in Z(\xi_1^H)} p(\vec{x}_1^T, s_1^T). \tag{6.10}$$

Das Maximum-Likelihood-Training der Parameter des Hidden-Markoff-Modells basiert auf dem Expectation-Maximization Algorithmus [McLachlan$^+$ 1997]. Im Gegensatz zum direkt aus dem EM-Algorithmus abgeleiteten Baum-Welch Training wird in den Philips Forschungslaboratorien auf den sogenannten Viterbi-Ansatz (Viterbi-Approximation) zurückgegriffen. Der Viterbi-Ansatz basiert auf der Approximation der Summe in (6.10) durch das Maximum:

$$p(\vec{x}_1^T|\xi_1^H) \quad \approx \quad \max_{s_1^T \in Z(\xi_1^H)} p(\vec{x}_1^T, s_1^T), \tag{6.11}$$

wodurch der gesamte Trainings- und Erkennungsalgorithmus deutlich vereinfacht wird. Der Trainingsalgorithmus wird in [Ney 1990] beschrieben. Er besteht aus einem iterativ wiederholten 2-Schritt-Verfahren:

- Im ersten Schritt wird bei gegebener Zuordnung der HMM-Zustände s_t zu den Trainingsbeobachtungen $\vec{x}_t$ eine Parameterschätzung der Emissionsverteilungsdichten $p(\vec{x}|s)$ durchgeführt. Dabei werden mit Hilfe eines Splittingverfahrens die Mischverteilungen (6.6) der HMM-Zustände gebildet.
- Im zweiten Schritt wird bei gegebenen Mischverteilungen eine neue Zuordnung der HMM-Zustände zu den Trainingsbeobachtungen durchgeführt. Diese Zuordnung wird mit Hilfe der sogenannten Zeitanpassung [Rabiner$^+$ 1993], S. 200 ff., durchgeführt.

Die drei verwendeten Übergangswahrscheinlichkeiten des Hidden-Markoff-Models werden im beschriebenen System manuell eingestellt.

6.2.3 Sprachmodellierung

Das Sprachmodell liefert für jede Wortfolge $w_1^L = w_1, \ldots, w_L$ einen Schätzwert für die Wahrscheinlichkeit $p(w_1^L)$. Dazu wird diese Wahrscheinlichkeit zunächst in ein Produkt bedingter Wahrscheinlichkeiten umgeformt:

$$p(w_1^L) \quad = \quad \prod_{i=1}^{L} p(w_i|w_1, \ldots, w_{i-1}). \tag{6.12}$$

Die standardmäßig eingesetzte M-gramm Methode für die Bestimmung der bedingten Wahrscheinlichkeiten $p(w_i|w_1, \ldots, w_{i-1})$ basiert auf der Modellannahme, daß das Wort w_i durch $M-1$ Vorgängerwörter genügend gut vorhergesagt werden kann, d.h.

$$p(w_i|w_1, \ldots, w_{i-1}) \quad = \quad p(w_i|w_{i-M+1}, \ldots, w_{i-1}). \tag{6.13}$$

Die Historienlänge $M-1$ hängt von der Menge der zur Verfügung stehenden Trainingsdaten und den zur Verfügung stehenden Rechenressourcen ab. Für $M = 2$ und $M = 3$ erhält man die weitverbreiteten Bigramm- und Trigramm-sprachmodelle. Diese Sprachmodelle werden üblicherweise mit Maximum-Likelihood-Verfahren auf umfangreichen Trainingstexten geschätzt. Wenn die Menge der Trainingsdaten beschränkt ist, können Wortpaare oder -tripel

im Trainingskorpus fehlen oder zu selten auftreten. Für diesen Fall werden die zugehörigen Wahrscheinlichkeiten mit **Backing-Off** Methoden [Kneser$^+$ 1995] bestimmt. Als Qualitätskriterium für das Sprachmodell gilt die **Perplexität** PP:

$$PP(w_1^L) = \sqrt[L]{\prod_{i=1}^{L} p(w_i|w_1,\ldots,w_{i-1})} \tag{6.14}$$

und die Anzahl der verwendeten Verteilungsparameter. Die Perplexität des Sprachmodells beschreibt die mittlere Anzahl der Wörter pro Wortposition im Satz, zwischen denen sich das Erkennungssystem mit Hilfe der akustischen Information entscheiden muß. Tabelle 6.2 faßt Perplexitäten und Parameteranzahl des verwendeten Bigrammsprachmodells (bg) und Trigrammsprachmodells (tg) für die WSJ0-Aufgabe zusammen.

Tabelle 6.2: Übersicht über WSJ0-Sprachmodelle für ein Vokabular von 5000 Wörtern, WSJ0-Training, Perplexitätsmessung des Bigrammsprachmodells (bg) und des Trigrammsprachmodells (tg) erfolgt auf Development'92, Eval'92, Development'93 und Eval'93

Modell	Kurzbezeichnung	Parameter	Perplexität
Bigramm	bg	$1.6 \cdot 10^6$	128
Trigramm	tg	$11 \cdot 10^6$	80

6.2.4 Suche

Eine integrierte zeitsynchrone Beamsuche wird eingesetzt, um die Wortfolge des gesprochenen Satzes bei einem großen Vokabular von 5000 bis 64000 Wörtern zu bestimmen [Ney$^+$ 1992]. Bei der integrierten Sucharchitektur stehen alle Wissensquellen (soweit es ihre Komplexität erlaubt) gleichzeitig bei der Bewertung der hypothetisierten Wortfolge zur Verfügung. Außerdem beziehen sich alle betrachteten Hypothesen auf den gleichen Abschnitt des Sprachsignals (Zeitsynchronität). Diese beiden Eigenschaften erlauben eine drastische Reduktion des aktiven Suchraums durch das Verwerfen von weniger gut bewerteten Hypothesen (Pruning). Bei einer zeitsynchronen Suche mit großem Vokabular erweist sich die Baumorganisation des Aussprachelexikons als vorteilhaft. Dabei werden Wortanfangsbruchstücke, die die gleiche Phonemfolge aufweisen, in einem Phonembaum zusammengelegt. Dadurch werden die zugehörigen Phoneme nur einmal hypothetisiert, was den Suchraum erheblich einschränkt. Der erreichte Kompressionsfaktor wird während der Erkennung noch weiter verstärkt, da die Anzahl der Hypothesen an den Wortanfängen drastisch ansteigt. Dieser Anstieg wird durch die Unsicherheit des Erkennungssystems an den Wortanfängen bedingt, da hier die Verwechselbarkeit mit anderen Wörtern am höchsten ist. Aus Komplexitätsgründen wird im beschriebenen System eine einstufige Bigrammsuche durchgeführt, d.h. im ersten Erkennungspaß wird nur das Bigrammsprachmodell eingesetzt. Um auch das Trigrammsprachmodell ausnutzen zu können, wird während der Bigrammsuche ein **Lattice** erzeugt (siehe Abschnitt 6.3.1). Das Lattice ist ein Netzwerk, das die Menge der während der Suche hypothetisierten Worthypothesen enthält. Dabei werden im wesentlichen Zeit-, Bewertungs- und Vorgängerwortinformationen abgespeichert. Auf dem Lattice wird mit Hilfe einer weiteren zeitsynchronen dynamischen Programmierung eine Trigrammsuche mit dem Trigrammsprachmodell durchgeführt. Eine detaillierte Beschreibung der Suchalgorithmen in dem verwandten RWTH-System wird in [Ortmanns 1998] vorgenommen. Effiziente Verfahren zur Berechnung der Likelihood $b_i(\vec{x}|s)$ während der Erkennung werden in [Beyerlein 1994] und in [Beyerlein 1995] analysiert.

6.2.5 Erkennungsgenauigkeit

Tabelle 6.3 zeigt die Wortfehlerraten, die das beschriebene Spracherkennungssystem auf den in Abschnitt 6.1 beschriebenen Testdaten hat. Zwei Fehlerraten werden angegeben, die Fehlerrate des mit wortinternen Triphonmodellen und einem Bigrammsprachmodell gebildeten Systems (ww+bg) und die Fehlerrate des korrespondierenden Trigrammsystems (ww+tg). Die gezeigten Fehlerraten entsprechen dem weltweiten Stand der Technologie in den Jahren 1992 und 1993.

Tabelle 6.3: Wortfehlerraten (in %) für ein Vokabular von 5000 Wörtern, Bigrammsprachmodell (bg), Trigrammsprachmodell (tg), WSJ0-Training wortinterner Triphonmodelle (ww)

Modellierung	si_dt_05'92-m	si_et_05'92-m
ww+bg	9.4	5.4
ww+tg	8.0	4.3

6.3 Implementierung von DMC

Die in Kapitel 4 beschriebenen Verfahren zur Optimierung der Modellkombination basieren auf der Summe von Bewertungen aller vorkommenden Satzhypothesen. Diese Summe wird aus Komplexitätsgründen auf die Menge aller Satzhypothesen eingeschränkt, die während der Erkennung hypothetisiert werden. Da ein Satz aus einer Folge von Wörtern besteht, lassen sich die Satzhypothesen kompakt in **Lattices** speichern [Aubert$^+$ 1995], [Ney$^+$ 1996], [Odell 1995].

6.3.1 Lattice-Organisation

Ein Lattice beschreibt dabei ein Netzwerk (gerichteter azyklischer Graph) von Worthypothesen (Kanten des Graphen). Jede Worthypothese w besteht aus folgenden Einträgen

- Wort
- Startknoten der Worthypothese im Lattice
- Endknoten der Worthypothese im Lattice
- Endzeit der Worthypothese
- Bewertung $\log p_1(w|x)$
- Bewertung $\log p_2(w|x)$
  $\vdots$
- Bewertung $\log p_M(w|x)$

Das Lattice wird im allgemeinen während einer einstufigen Bigramm- oder Trigrammsuche erzeugt. Entsprechend dem verwendeten akustischen und Sprachmodellkontext muß das Lattice expandiert werden. Im Ergebnis dieser Latticemanipulation entsteht ein neues Lattice, bei dem jeder Knoten eindeutig durch einen speziellen akustischen und Sprachmodellkontext definiert ist. Beide Kontexte werden in der Implementierung ausschließlich über Wortfolgen definiert. Die Dichte des verwendeten Lattice, d.h. die mittlere Anzahl der Worthypothesen bezogen auf die Anzahl der gesprochenen Wörter ist in Tabelle 6.4 dargestellt.

Tabelle 6.4: Latticedichten für das Training der DMC-Koeffizienten Λ und die Erkennung mit Hilfe der trainierten DMC-Koeffizienten Λ, Messung erfolgte auf der WSJ0-Datenbasis und der HUB4-Datenbasis

Datenbasis	Latticedichte
si_dt_05'92, si_et_05'92	10
HUB4'96 (Kapitel 7)	600
HUB4'97 (Kapitel 7)	500

6.3.2 DMC-Training

Für das diskriminative Training der Koeffizienten $\Lambda = (\lambda_1, \ldots, \lambda_M)^T$ müssen zunächst die rivalisierenden Wortfolgen $k = 1, \ldots, K$, $k \neq k_n$ bestimmt werden. Dazu wird für jeden Trainingssatz mit Hilfe einer einstufigen Suche ein Lattice erzeugt (vgl. Abschnitt 6.3.1). Aus dem Lattice werden anschließend vollständige Satzhypothesen extrahiert. Dazu bietet sich das weitverbreitete $\mathcal{N}$-Best-Verfahren an. Die hier verwendete $\mathcal{N}$-Best-Erkennung ist in [Tran$^+$ 1997] ausführlich beschrieben. Die Größe von $\mathcal{N}$ wird für die verschiedenen Trainingsdatenbasen in Tabelle 6.5 angegeben. Mit dieser $\mathcal{N}$-Best-Konfiguration liegt die $\mathcal{N}$-Best-Fehlerrate auf den si_dt_05'92-Daten

Tabelle 6.5: Anzahl der $\mathcal{N}$ besten Hypothesen für das Training der DMC-Koeffizienten Λ auf den WSJ0-Entwicklungsdaten und den HUB4'96-Entwicklungsdaten

Datenbasis	$\mathcal{N}$
si_dt_05'92	5,...,30
HUB4'96	500,...,700

bereits in der Nähe der Latticefehlerrate von 1%-3%. Da die HUB4-Aufgabe (Kapitel 7) deutlich schwieriger ist, liegt hier die Latticefehlerrate bei ca. 10%-15% (vgl. Abschnitt 7.3.2). Die für die Optimierung von Λ verwendete Anzahl von Klassen $k = 1, \ldots, K$ ist durch den Wert $\mathcal{N}$ nach oben beschränkt. Ist $\mathcal{N}$ zu klein, dann sind die Satzhypothesen nicht variabel genug, um Koeffizienten Λ zu schätzen, die sich in einer freien Erkennung oder einer Erkennung auf einem dichten Wortgraphen optimal verhalten. Die Anzahl der betrachteten Hypothesen $\mathcal{N}$ kann andererseits aus Mangel an Speicherressourcen nicht beliebig weit erhöht werden.
Um zu garantieren, daß die Anzahl $\mathcal{N} = 500, \ldots, 700$ nicht zu klein ist, wurde das Training der Koeffizienten Λ für $\mathcal{N} = 2000, \ldots, 3000$ und für $\mathcal{N} = 5000, \ldots, 8000$ wiederholt. Die auf der Entwicklungsdatenbasis gemessenen Wortfehlerraten sinken um 1% für $\mathcal{N} = 2000, \ldots, 3000$ und bleiben bei größerem $\mathcal{N}$ konstant. Somit ist mit $\mathcal{N} > 500, \ldots, 700$ kein signifikanter Gewinn gegeben. Da der Ressourcenaufwand mit $\mathcal{N}$ steigt, wurden alle weiteren Experimente mit $\mathcal{N} = 500, \ldots, 700$ durchgeführt.
Die in den Experimenten verwendeten Optimierungsgleichungen (4.18) und (4.30) wurden ohne wesentliche Modifikationen implementiert. Die Summation in den Gleichungen wurde auf die oben diskutierte Anzahl von Satzhypothesen $\mathcal{N}$ beschränkt. In Gleichung (4.18) werden mit $\eta = 3$ in den Experimenten optimale Ergebnisse erzielt. Die Werte A, B in Gleichung (4.30) werden so eingestellt, daß ca. 10%-50% der Hypothesen, die direkt an der Klassengrenze liegen, in den Bereich des Parabelastes fallen. Die anderen Hypothesen werden bei der Optimierung vernachlässigt. Mit diesen Einstellungen waren die erhaltenen Koeffizienten aus beiden Gleichungen nahezu indentisch.

6.3.3 DMC-Erkennung

Bei der Erkennung kann wie auch im Training, auf den Normierungsterm im Nenner der log-linearen Gesamtverteilung

$$p_\Lambda(w_1^N|x_1^T) = \frac{\prod_j p_j(w_1^N|x_1^T)^{\lambda_j}}{\sum_{w_1'^{N'}} \prod_j p_j(w_1'^{N'}|x_1^T)^{\lambda_j}}$$

verzichtet werden, da er bei der Erkennung keinen Einfluß auf den Ausgang der Entscheidung für die optimale Wortfolge hat bzw. im Training durch Quotientenbildung entfällt. Als erkannte Wortfolge wird diejenige Wortfolge w_1^N ausgegeben, für die gilt:

$$w_1^N = \arg\max_{w_1'^{N'}} \sum_j \lambda_j \cdot \log p_j(w_1'^{N'}|x_1^T)$$

Diese Entscheidungsregel kann nach Aufteilung der Wortfolgen $w_1^{'N'}$ in die einzelnen Worthypothesen $w_1^{'N'} = (w'_1, \ldots, w'_{N'})$ weiter umgeformt werden:

$$w_1^N = \arg\max_{w_1^{'N'}} \sum_{i=1}^{N'} \sum_j \lambda_j \cdot \log p_j(w'_i|x_1^T)$$

Mit Hilfe der Koeffizienten Λ kann damit für jede Worthypothese w'_i im Lattice die Gesamtbewertung $\sum_j \lambda_j \cdot \log p_j(w'_i|x_1^T)$ berechnet und abgespeichert werden. Die optimale Wortfolge w_1^N, die sich aus der Gesamtbewertung ergibt, wird mit Hilfe einer dynamischen Programmierung [Rabiner$^+$ 1993], S. 204 ff., ermittelt.

6.4 Anwendung von DMC

Dieser Abschnitt beschreibt die Anwendung des DMC-Verfahrens auf das WSJ0-System. Zunächst werden die verwendeten Basismodelle beschrieben. Anschließend werden Experimente zur optimalen Einstellung des Sprachmodellfaktors und zur Bildung einer Modellkombination aus 2 akustischen und 3 Sprachmodellen zusammengefaßt.

6.4.1 Basismodelle

Als Basismodelle wurden folgende Sprachmodelle eingesetzt:

bg: Bigrammsprachmodell. Es beschreibt die Wahrscheinlichkeit $p(u|v)$, daß das Wort u auf das Wort v folgt. Das eingesetzte Sprachmodell wurde auf den WSJ0-Textdaten mit Hilfe der in [Kneser$^+$ 1995] beschriebenen Techniken geschätzt.

tg: Trigrammsprachmodell. Es beschreibt die Wahrscheinlichkeit $p(u|vw)$, daß das Wort u auf die Wortfolge wv folgt. Das eingesetzte Sprachmodell wurde analog zum Bigramm auf den WSJ0-Textdaten geschätzt.

fg: Viergrammsprachmodell. Es beschreibt die Wahrscheinlichkeit $p(u|vwx)$, daß das Wort u auf die Wortfolge xwv folgt. Es wurde wie das Bi- und das Trigramm auf den WSJ0-Textdaten geschätzt.

Außerdem wurden folgende akustisch-phonetische Modelle verwendet:

ww: Wortinternes Triphonmodell. Es basiert auf der Modellannahme, daß die Koartikulation zwischen den Phonemen nur innerhalb der Wörter stattfindet. Die für die WSJ0-Aufgabe verwendeten HMM-Zustände der Triphone werden mit Hilfe eines entscheidungsbaum-basierten Cluster-Verfahrens [Beyerlein$^+$ 1997a] in 3000 Zustände zusammengefaßt. Die Mischverteilungen der Zustände bestehen aus je 32 Laplaceverteilungen. Alle 96000 Laplace-Verteilungen besitzen einen gemeinsamen Deviationsvektor. Die Mischverteilungen der HMM-Zustände wurden mit Hilfe der in Abschnitt 6.2.2 beschriebenen Viterbi-Approximation trainiert.

xw: Wortübergreifendes Triphonmodell. Es wurde im Verlaufe der vorliegenden Arbeit erstmalig erfolgreich im Philips-System eingesetzt. Bei diesen Modellen kann eine Koartikulation zwischen benachbarten Wörtern hypothetisiert werden. Die erforderlichen Änderungen in Training und Erkennung sind in [Beyerlein$^+$ 1997a] beschrieben. Hier werden die wortübergreifenden Triphonzustände zu 4000 Zustandsclustern mit je 32 Dichten zusammengefaßt. Alle 128000 Laplace-Verteilungen besitzen einen gemeinsamen Deviationsvektor. Die Mischverteilungen der HMM-Zustände wurden auch hier mit Hilfe der Viterbi-Approximation trainiert.

Tabelle 6.6 faßt Perplexitäten und Parameteranzahl des Bigrammsprachmodells (bg) und des Trigrammsprachmodells (tg) zusammen.

6.4.2 Experimentelle Ergebnisse

Die Wortfehlerraten für die Anwendung des DMC-Verfahrens werden in Tabelle 6.7 zusammengefaßt. Dabei wird der Koeffizientenvektor Λ mit Hilfe des MWE-Kriteriums bestimmt.

Aus der Tabelle wird deutlich, daß

Tabelle 6.6: Übersicht über die Sprachmodelle, die für DMC auf der WSJ0-Datenbasis mit einem Vokabular von 5000 Wörtern eingesetzt wurden, WSJ0-Training, Messung der Perplexitäten auf Development'92, Eval'92, Development'93 und Eval'93

Modell	Kurzbezeichnung	Parameter	Perplexität
Bigramm	bg	$1.6 \cdot 10^6$	128
Trigramm	tg	$11 \cdot 10^6$	80
Viergramm	fg	$32 \cdot 10^6$	72

Tabelle 6.7: Wortfehlerraten (in %) für die diskriminative Kombination (DMC) von Bigrammsprachmodell, Trigrammsprachmodell, Viergrammsprachmodell, wortinternem Triphonmodell und wortübergreifendem Triphonmodell, WSJ0-Training, Vokabulargröße 5000 Wörter, M: Anzahl der Basismodelle

Modelle	M	si_dt_05'92-m	si_et_05'92-m
ww+bg (baseline)	2	9.4	5.4
ww+bg (DMC)	2	9.5	5.5
xw+bg (baseline)	2	8.2	5.2
xw+bg (DMC)	2	8.1	5.1
xw+tg (baseline)	2	7.0	4.0
xw+tg (DMC)	2	7.0	4.0
xw+fg (baseline)	2	6.6	3.5
xw+fg (DMC)	2	6.6	3.5
ww+xw +bg+tg+fg (DMC)	5	6.3	3.2 (-8%)

- mit dem DMC-Verfahren der optimale Sprachmodellfaktor automatisch bestimmt werden kann und
- eine Kombination von insgesamt fünf Basismodellen zu einer um 8% Prozent relativ geringeren Fehlerrate führt als die beste paarweise Kombination der fünf Basismodelle.

Damit konnte die Hypothese über eine optimale Kombination von Basismodellen mit Hilfe des beschriebenen DMC-Verfahrens für die WSJ0-Datenbasis validiert werden.

6.5 Zusammenfassung

Der DMC-Ansatz (vgl. Abschnitt 4.9) wurde experimentell auf dem männlichen Teil der WSJ0-Datenbasis getestet:

A: Aus akustischen Modellen $p_i(x|k), i = 1, 2$

i=1: Wortinternes Triphonmodell, Viterbi-Training auf der WSJ0-Datenbasis,

i=2: Wortübergreifendes Triphonmodell, Viterbi-Training auf der WSJ0-Datenbasis

und Sprachmodellen $p_j(k), j = 1, 2, 3$

j=1: Bigrammsprachmodell, Training auf der WSJ0-Datenbasis,

j=2: Trigrammsprachmodell, Training auf der WSJ0-Datenbasis,

j=3: Viergrammsprachmodell, Training auf der WSJ0-Datenbasis

wurde die log-lineare Modellkombination

$$p_{\Lambda}(k|x) \quad = \quad \frac{\prod_i p_i(x|k)^{\lambda_i} \prod_j p_j(k)^{\lambda_j}}{\sum_{k'} \prod_i p_i(x|k')^{\lambda_i} \prod_j p_j(k')^{\lambda_j}} \, .$$

gebildet.

B: Als diskriminatives Trainingskriterium wurde

$$E(\Lambda) \quad = \quad \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) S(k, n, \Lambda)$$

verwendet, wobei die Funktion S entsprechend dem Kriterium B.4 (Abschnitt 4.9) gewählt wurde. Die Menge der rivalisierenden Wortfolgen $k = 1, \ldots, K, k \neq k_n$ wurde mit Hilfe einer N-Best-Erkennung [Tran$^+$ 1997] definiert. Die Anzahl K der betrachteten Klassen lag dabei in Abhängigkeit von der Trainingsäußerung zwischen 5 und 30. Der Einfluß der Größe von K auf das Training der Koeffizienten Λ wird im folgenden Kapitel diskutiert.

Die zu Beginn der vorliegenden Arbeit gegebene Baseline-Fehlerrate des Hochleistungsspracherkenners der Philips Forschungslaboratorien auf den si_et_05'92-m-Daten (Tabelle 6.3) konnte durch den zusätzlichen Einsatz von

- wortübergreifenden Triphonmodellen (xw)
- Viergrammsprachmodell (fg)
- DMC

um 25% von 4.3% auf 3.2% reduziert werden. Auf den Entwicklungsdaten (si_dt_05'92-m) wurde die Fehlerrate um 21% von 8.0% auf 6.3% reduziert.
Um die erhaltenen Ergebnisse zu untermauern, wurde das DMC-Verfahren auf der deutlich größeren HUB4-Datenbasis getestet. Wie im folgenden Kapitel deutlich werden wird, besitzt diese Spracherkennungsaufgabe einen höheren Schwierigkeitsgrad.

Kapitel 7

Experimente auf der HUB4-Datenbasis

7.1 Die Broadcast-News-Aufgabe (BN-Aufgabe)

Die Spracherkennungsforschung hat sich bis zum Jahr 1995 unter anderem auf Diktieranwendungen mit großem Vokabular in Studioqualität konzentriert. Seit dem Jahre 1995 fokussiert sich die Forschung stärker auf die Erkennung von natürlicher Alltagssprache. Eine unerschöpfliche Datenquelle sind dabei die Audioaufnahmen von Radio- und Fernsehsendungen. Diese Daten werden im folgenden als **Broadcast-News-Daten (BN-Daten)** bezeichnet. Die BN-Sprachdaten haben folgende Eigenschaften:

- Unbekannte Satzgrenzen, bei bisherigen Sprachdatensammlungen waren die Satzgrenzen vorgegeben.
- Vielfältige und sich schnell ändernde akustische Umgebung. Die Qualität des Sprachsignals leidet unter Hintergrundmusik, Rauschen, Hintergrundsprechern, sowie unter häufiger Verwendung des Telefonkanals.
- Nichtmuttersprachliche Akzente und regionale Dialekte.
- Alltagssprache und Sprecherwechsel mitten in der Äußerung. Der Sprechstil variiert von professionellen Berichten bis zu spontaner Konversation.
- Unvorhersehbare Änderungen des Themas durch die Aktualität der Berichte wie auch durch ungeplante Reaktionen während eines Disputes.

Dadurch erhöht sich die Schwierigkeit der Spracherkennungsaufgabe und damit die Komplexität der eingesetzten Spracherkennungsprototypen. Weltweit ist nur eine geringe Zahl von Hochleistungsspracherkennern bekannt, die auf der BN-Aufgabe verwendbare Erkennungsergebnisse liefern. Dabei sind im wesentlichen die Sprachen Englisch, Spanisch, Chinesisch (Mandarin) und Japanisch vertreten. Die Leistung dieser Systeme wird seit 1996 in der von NIST organisierten und von DARPA gesponsorten **HUB4-Evaluation** getestet. Die BN-Daten werden vom Linguistic Data Consortium (LDC) für die HUB4-Evaluation zur Verfügung gestellt [Graff 1997]. Das LDC nahm 130 Stunden von Radio- und Fernsehsendungen auf und ließ die Daten transkribieren. Davon werden 74 h für Trainingszwecke und 10 h für die Definition von Testdatenbasen verwendet. Die Aufnahmen wurden bei einer mit gewöhnlichen Audiorecordern von den Fernsehsendern ABC, CNN, CSPAN und dem Radiosender NPR bereit gestellt. Die Abtastrate betrug dabei 16 kHz. Die Tabellen 7.1, 7.2 fassen die Aufteilung der Trainingsdatenbasis und des Testpools zusammen. Zusätzlich akquirierte das LDC für das Sprachmodelltraining das 'BN-Archive'. Es beinhaltet 140 Millionen Wörter Text aus Sendungen der vergangenen 5 Jahre.

Da die HUB4-Datenbasis verschiedenste Spracherkennungsaufgaben in sich vereinigt, wurde sie ursprünglich in die sogenannten **Fokusbedingungen (F-Konditionen)** aufgeteilt. Die verwendeten Fokusbedingungen sind in Tabelle 7.3 dargestellt. Tabelle 7.4 beschreibt die Testdaten der HUB4-Evaluation im Jahre 1996, die als Entwicklungsdaten in der vorliegenden Arbeit verwendet werden. Tabelle 7.5 faßt die Aufteilung des für die HUB4-Evaluation im Jahre 1997 verwendeten Testmaterials zusammen. Das Testmaterial besteht aus einem einzigen Sprachsignal mit einer Länge von 3 h. Eine senderspezifische Aufteilung liegt hier nicht mehr vor. Aus der Tabelle ist zu ersehen, daß auch dieses Testmaterial trotz Konzentration auf 'geplante Nachrichtensprache' ein breites Spektrum an Spracherkennungsaufgaben abdeckt. Der Datensatz wird verwendet, um das DMC-Verfahren zu testen.

Tabelle 7.1: Zusammensetzung der Trainingsdatenbasis HUB4'97 aus verschiedenen regelmäßig ausgestrahlten Nachrichtensendungen der Sender ABC, CNN, CSPAN, NPR

Netzwerk	Sendung	Menge (h)
ABC	Nightline	7.7
ABC	World Nightly News	6
ABC	World News Tonight	4.1
CNN	Early Edition	3
CNN	Early Prime News	6
CNN	Headline News	5.4
CNN	Prime News	4
CNN	The World Today	4.6
CSPAN	Washington Journal	12
NPR	All Things Considered	15
NPR	Marketplace	6
Total		73.8

Tabelle 7.2: Zusammensetzung des Testpools HUB4'97 aus verschiedenen regelmäßig ausgestrahlten Nachrichtensendungen der Sender ABC, CNN, CSPAN, NPR (die Evaluierungsdaten für die jeweilige Evaluierung werden von NIST aus dem Testpool mit einem Zufallsgenerator entnommen).

Netzwerk	Sendung	Abkürzung	Menge (h)
ABC	Prime Time News	ABC P.T.N.	1
CNN	Morning News	CNN M.N.	2
CNN	World View	CNN W.V.	1
CSPAN	Washington Journal	CSP W.J.	2
NPR	Morning Edition	NPR M.E.	2
NPR	Market Place	NPR M.P.	1
NPR	The World	NPR T.W.	1
Total			10

7.2 Das Philips-HUB4-System

Dieser Abschnitt beschreibt die Erweiterungen des WSJ0-Systems der Philips Forschungslaboratorien Aachen (Abschnitt 6.2), die für die Erkennung von BN-Daten erforderlich waren. Zunächst wurde das System an die Erkennung mit einem Vokabular von 64k Wörtern angepaßt. Die dazu notwendigen Veränderungen sind in [Dugast$^+$ 1995a] beschrieben. Das dort beschriebene System wurde für die Broadcast-News-Aufgabe weiterentwickelt.

7.2.1 Überblick

Das Philips-HUB4-System ist ein wortübergreifendes Triphon-System (Abschnitt 6.4) mit MFCC-Merkmalen und Laplace-Mischverteilungen (Abschnitt 6.2.2). Es entstand aus dem in Abschnitt 6.2 beschriebenen WSJ0-System. Die Erweiterungen gegenüber dem WSJ0-System können wie folgt zusammengefaßt werden:

- Kanal- und Sprechernormierung werden mit Hilfe von Langzeitspektrumnormierung, Varianznormierung, Vokaltraktlängennormierung und MLLR-Adaption durchgeführt (Abschnitt 7.2.3).

- Ein Segmentierungsverfahren wird verwendet, um aus dem bis zu 3 h langen Sprachsignal satzähnliche Segmente zu gewinnen (Abschnitt 7.2.2). Anschließend wird ein Sprechercluster-Verfahren eingesetzt, um Segmente des gleichen Sprechers für eine MLLR-Adaption zusammenzufassen.

Tabelle 7.3: Übersicht über die HUB4-Fokusbedingungen, jeder Fokus beinhaltet eine spezielle Herausforderung für die automatische Spracherkennung

	Beschreibung
F0	Geplante Nachrichtensprache, z.B. Verlesen von Nachrichten
F1	Spontane Sprache, z.B. Fernsehdiskussionen
F2	F0+F1 über Telefon, z.B. Telefoninterview
F3	F0 + Musik im Hintergrund, z.B. am Beginn einer Sendung
F4	F0 + Störungen im Hintergrund, z.B. Beifall
F5	F0 + Sprache mit Akzent, z.B. Chinesischer Akzent
FX	Alles andere, z.B. Telefoninterview mit einem Chinesen

Tabelle 7.4: Zusammensetzung der HUB4'96-Evaluationsdaten (#Wrt.: Anzahl der Wörter) aus den Sendungen CNN Morning News (CNN M.N.), CSPAN Washington Journal (CSP W.J.), NPR The World (NPR T.W.) und NPR Market Place (NPR M.P.), diese Daten werden als Trainingsdaten für DMC verwendet

Name	#Wrt.	Name	#Wrt.	Name	#Wrt.	Name	#Wrt.
File 1 CNN M.N.	4644	File 2 CSP W.J.	5742	File 3 NPR T.W.	4846	File 4 NPR M.P.	4558
F0	1684	F0	538	F0	1334	F0	2299
F1	1286	F1	350	F1	848	F1	851
F2	0	F2	1568	F2	0	F2	144
F3	204	F3	0	F3	453	F3	699
F4	1454	F4	115	F4	14	F4	211
F5	0	F5	0	F5	74	F5	221
FX	16	FX	0	FX	2123	FX	133

- Aus häufigen Wortfolgen werden Phrasen gebildet, um Verschleifungs- und Koartikulationseffekte von typischen Wortfolgen explizit modellieren zu können (Abschnitt 7.2.4) und um die Reichweite des Sprachmodells zu erhöhen (Abschnitt 7.2.5).
- Geschlechtsunabhängige wortinterne und wortübergreifende Triphone werden auf 46 h des HUB4-Trainingsmaterials trainiert (Abschnitt 7.2.4).
- Um akustische und Sprachmodelle mit längerreichweitigem Kontext einsetzen zu können, wird eine Multi-Paß-Erkennung durchgeführt, beginnend mit einer einstufigen Bigrammerkennung mit wortinternen Triphonen und endend mit einer Trigramm-Wortgraph-Erkennung mit Hilfe von MLLR-adaptierten wortübergreifenden Triphonen (Abschnitt 7.2.6).

7.2.2 Automatische Segmentierung

Das Audiosignal (mit einer Länge von 3 h) wird mit einer von NIST zur Verfügung gestellten Grobsegmentierung in 769 Rohsegmente aufgeteilt. Die Rohsegmente werden mit einem Phonemerkenner [Kubala$^+$ 1997] in kurze männliche, weibliche und nicht-sprachliche (NSP) Segmente verfeinert. Der Phonemerkenner selbst basiert auf einer einstufigen zeitsynchronen Phonem-Bigramm-Erkennung, bei der männliche und weibliche Modelle konkurrieren. Die kontextunabhängigen Phonemmodelle werden geschlechtsspezifisch auf den BN-Daten trainiert. Tabelle 7.6 zeigt die resultierende Verteilung der Segmente.

Die NSP-Segmente (ca. 8 min) sind von der weiteren Erkennung ausgeschlossen. Die verbleibenden Segmente werden zusätzlich in die Klassen Schmalband und Breitband eingeteilt. Diese Klassifikation wird für das Segmentclustern verwendet. Anschließend findet mit Hilfe eines agglomerativen Bottom-Up-Cluster-Verfahrens eine Gruppierung der Segmente in Sprechercluster statt. Als Abstandsmaß dient dabei die Kullback-Leibler-Distanz

Tabelle 7.5: Zusammensetzung der HUB4'97-Evaluationsdaten aus den verschiedenen Fokusbedingungen F0, F1, F2, F3, F4, F5, Fx. Diese Daten wurden von NIST mit einem Zufallsgenerator aus dem HUB4'97-Testpool entnommen.

Fokusbedingung	F0	F1	F2	F3	F4	F5	FX	Alle
Wörter	13197	6566	4881	1571	3350	669	2598	32832

Tabelle 7.6: Aufteilung der automatisch erzeugten Segmente bezüglich Bandbreite des Kanals und Geschlecht des Sprechers; Ausgangspunkt für die automatische Segmentierung ist das 3 h lange Sprachsignal der HUB4'97 Evaluierung, das Pausen, Musik, Störungen, Kanal- und Sprecherwechsel enthält, die nicht annotiert sind.

Segmente	Alle	Weiblich	Männlich
Schmalband	271	66	205
Breitband	1160	417	743
NSP	227	-	-
Alle	1658	483	948

der Gaußverteilungen der Merkmalsvektoren der Segmente. Die Gruppierung wird separat auf den männlichen und weiblichen Schmalband- und Breitbandsegmenten durchgeführt. Im Ergebnis des Clusterns erhält man 106 Segmentcluster mit einer mittleren Dauer von 100 Sekunden. Die beschriebene automatische Segmentierung (UE) wird in Tabelle 7.7 mit einer manuellen Segmentierung (PE), die von NIST zur Verfügung gestellt wurde, verglichen. In der Tabelle werden Fehlerraten nach einer Bigrammerkennung mit wortinternen Triphonen (Paß (I)) und nach Paß (VI), einer Trigrammwortgraph-Erkennung mit wortübergreifenden Triphonen, dargestellt. Eine detaillierte Beschreibung der Erkennungspässe findet man in Abschnitt 7.2.6 . Die zugehörigen akustischen Modelle und Sprachmodelle werden in den Abschnitten 7.2.4 und 7.2.5 beschrieben. Aus der Tabelle wird ersichtlich, daß die Wortfehlerrate durch Verwendung der automatischen Segmentierung um 5% relativ gegenüber einer manuellen Segmentierung steigt.

Tabelle 7.7: Wortfehlerraten (%) auf der HUB4'96-Datenbasis für die manuelle Segmentierung ('partitioned evaluation', PE) und die automatische Segmentierung ('unpartitioned evaluation', UE); die Segmentierungen werden für den Paß (I), d.h. eine einstufige Bigrammsuche mit wortinternen Triphonmodellen, und den Paß (VI), d.h. eine Trigramm-Wortgraphensuche mit adaptierten wortübergreifenden Triphonmodellen, verglichen.

	File1-4	CNN M.N. File1	CSP W.J. File2	NPR T.W. File3	NPR M.P. File4
PE, (I)	35.0	36.7	33.4	39.7	30.3
UE, (I)	36.3	37.1	35.3	40.4	32.4
PE, (VI)	28.7	29.0	26.4	34.2	25.6
UE, (VI)	30.2	30.0	29.1	35.1	26.8

7.2.3 Merkmalsextraktion

Eine ausführliche Beschreibung der verwendeten Merkmalsextraktion findet man in [Häb$^+$ 1998]. Das System basiert auf MFCC Merkmalen [Rabiner$^+$ 1993], S. 189. Der Merkmalsvektor $\vec{x}_t$ besteht aus 16 MFCC Koeffizienten $y_{t,i}, i = 0, \ldots, 15$ (inclusive Energie $y_{t,0}$), 16 zugehörigen Regressionskoeffizienten 1. Ordnung,

$$\triangle y_{t,i} \quad = \quad y_{t+2,i} - y_{t-2,i} + 0.5 \cdot (y_{t+1,i} - y_{t-1,i}), \quad i = 0, \ldots, 15 \tag{7.1}$$

und dem zur Energie gehörenden Regressionskoeffizienten 2. Ordnung

$$\triangle\triangle y_{t,0} \quad = \quad y_{t-2,0} - 0.5 \cdot y_{t-1,0} - y_{t,0} - 0.5 \cdot y_{t+1,0} + y_{t+2,0}. \tag{7.2}$$

Der resultierende Vektor hat 33 Komponenten und wird einer Langzeitspektrumnormierung und einer Varianznormierung unterworfen [Häb$^+$ 1998]. Auf den MFCC-Vektor wird analog zu Abschnitt 6.2.1 eine LDA-Transformation angewendet. Die geschlechtsunabhängige LDA-Matrix wird zuvor auf den akustischen BN-Trainigsdaten geschätzt. Die 35 LDA-Komponenten mit den größten Eigenwerten bilden schließlich den LDA-Merkmalsvektor $\vec{x}_t$. **Vokaltraktlängennormierung (VTN)** [Lee$^+$ 1996] wird sowohl im Training als auch in der Erkennung genutzt. Die Idee von VTN besteht in der Reduktion des Einflusses der Vokaltraktlänge auf den Merkmalsvektor. Dazu wird eine lineare Verzerrung der Frequenzachse durch eine entsprechende Verschiebung der Mittenfrequenzen der Mel-Filterbank realisiert. Die Auswahl des Verzerrungsfaktors wird mit Hilfe eines Maximum-Likelihood-Ansatzes vollzogen [Lee$^+$ 1996]. Aus den Ergebnissen von Tabelle 7.8 kann man schlußfolgern, daß man mit Hilfe von VTN auf geschlechtsabhängige Modelle (GD) verzichten kann. Aus diesem Grund wurde bei der weiteren Systementwicklung mit geschlechtsunabhängigen Modellen (GI) gearbeitet. Eine ausführliche Analyse von VTN in einem ähnlichen Hochleistungsspracherkennungssystem wird in [Welling 1998] gegeben.

Tabelle 7.8: Effekt der Vokaltraktlängennormierung (VTN) auf die Wortfehlerrate (%); die Messung erfolgt auf der HUB4'96-Datenbasis unter der Verwendung der automatischen Segmentierung (UE) und einer einstufigen Bigrammsuche mit wortinternen Triphonmodellen

	Szenario	File 1-4, UE, Paß (I)
GD	kein VTN	36.3
GD	VTN in der Erkennung	35.6
GI	VTN in Training und Erkennung	35.4

Die Merkmalsextraktion wird auf alle Daten in gleicher Weise angewendet, Daten mit Telefonbandbreite werden nicht gesondert behandelt.

7.2.4 Akustische Modellierung

Das HUB4-Aussprachelexikon basiert auf dem Aussprachelexikon des North-American-Business-News-Systems [Dugast$^+$ 1995a]. Die phonetische Transkription neuer Wörter erfolgte mit Hilfe eines automatischen Systems [Besling 1994] und durch manuelle Korrektur. Typische Wortfolgen werden in sogenannte **Phrasen** umgewandelt und anschließend als eigenständige Einträge in das Lexikon und das Sprachmodell aufgenommen. Die Auswahl der 330 häufigsten Phrasen erfolgte auf dem Sprachmodelltrainingsmaterial (Abschnitt 7.2.5). Beispiele für typische kurze Phrasen sind: *in_the, of_the, on_the, to_the, and_the, you_know, for_the, to_be, I_think, that_the*. Zu den längeren Phrasen gehört z.B. *the_U._S._president*. Durch die Verwendung von Phrasen kann langreichweitiger akustischer- und Sprachmodellkontext modelliert werden. Typische Verschleifungen bei der Aussprache der Phrasen werden in Form von Aussprachevarianten in das Lexikon aufgenommen. Jeder der 65536 Lexikoneinträge hat im Mittel 1,15 Aussprachevarianten. Die Anwendung von wortinternen Triphonen bei Phrasen führt zu einer wortübergreifenden Triphonmodellierung zwischen phraseninternen Wörtern. Das Training des wortinternen und des wortübergreifenden Triphonmodells wird geschlechtsunabhängig auf einem manuell überprüften Teil (46 h) der akustischen BN-Trainingsdaten vollzogen. Im Unterschied zum Monophon-Backing-Off im WSJ0-System werden selten gesehene oder ungesehene Triphone mit Hilfe von **Entscheidungsbäumen** [Odell 1995], [Beyerlein$^+$ 1997a] trainiert. Dazu wird auf den vorhandenen Trainingsdaten mit Hilfe phonetischer Fragen ein Baum der Ähnlichkeitsklassen der Triphon-HMM-Zustände gebildet. Die Ähnlichkeitsklassen an den Blättern des Baumes definieren dann sogenannte Zustands-Cluster. Die HMM-Zustände innerhalb eines Zustands-Clusters besitzen per Definition ein und dieselbe Mischverteilung. Diese gemeinsame Benutzung einer Mischverteilung wird auch 'State-Tying' genannt. Durch das entscheidungsbaumbasierte Zustands-Clustern können selten gesehene oder auch ungesehene Triphone robust trainiert werden. Tabelle 7.9 faßt die Anzahl der Triphonzustände, der Zustands-Cluster und der zu trainierenden Parameter für das wortinterne Triphonmodell (ww) und für das

Tabelle 7.9: Anzahl der Triphone, Cluster, Dichten und Parameter der wortinternen Triphonmodelle (ww) und der wortübergreifenden Triphonmodelle (xw), die auf einem überprüften und korrigierten Teil (46 h) der HUB4'97-Trainingsdatenbasis trainiert werden

Basismodell	Triphone	Cluster	Dichten	Parameter
ww	11000	8000	340000	$11.9 \cdot 10^6$
xw	27000	10000	420000	$14.7 \cdot 10^6$

wortübergreifende Triphonmodell (xw) zusammen. Als Emissionsverteilungen werden kontinuierliche Laplace-Mischverteilungen eingesetzt, die einen gepoolten Deviationsvektor verwenden (vgl. Abschnitt 6.2.2). Die Parameter der Mischverteilungen werden mit einem Maximum-Likelihood-Training bestimmt, wobei wie in Abschnitt 6.2.2 die Viterbi-Approximation verwendet wird. Mit Hilfe des MLLR-Verfahrens (**Maximum Likelihood Linear Regression**) [Legetter 1994],[Thelen$^+$ 1997] erfolgt während der Erkennung eine Adaption der Mittelwertvektoren der Laplace-Dichten an die beobachteten Merkmalsvektoren der einzelnen Segmentcluster (Abschnitt 7.2.2). Nach einer Einteilung der Mittelwertvektoren in Ähnlichkeitsklassen, die auch Regressionsklassen genannt werden, wird auf der hypothetisierten Zustandsfolge für jede Regressionsklasse eine affine Transformation geschätzt. Die Regressionsklassen basieren auf phonetischem Wissen und werden dynamisch mit Hilfe einer Baumorganisation definiert. Die Menge an Adaptionsdaten ist abhängig von der Größe des jeweiligen Segmentclusters und definiert sowohl die Anzahl aktiver Regressionsklassen als auch die Struktur der MLLR-Transformationsmatrizen [Häb$^+$ 1998]. Bei der Adaption werden keine Beobachtungen verwendet, die in Pausen zwischen hypothetisierten Wörtern liegen. Die beschriebene Adaption wird sowohl für die wortinternen als auch für die wortübergreifenden Triphonmodelle eingesetzt. Die Fehlerratenreduktion durch die Kombination von VTN und MLLR liegt bei 8% relativ (siehe Tabelle 7.11, 7.12 Paß (III) in Abschnitt 7.2.6).

7.2.5 Sprachmodellierung

Für das Sprachmodelltraining wird vom LDC das BN-Archive zur Verfügung gestellt. Es besteht aus 140 Millionen Wörtern. Die für die Schätzung der Sprachmodelle eingesetzten Techniken werden ausführlich in [Klakow$^+$ 1998a] beschrieben. Die Vokabularauswahl für die Erkennung erfolgte in Abhängigkeit von der Häufigkeit der Wörter im BN-Archive. Um die Out-Of-Vocabulary (OOV) Rate des Systems zu senken, werden die 831 häufigsten Wörter aus den Transkriptionen des akustischen Trainingskorpus (TAT) und die Eigennamen aus der HUB4-Sprecherdatenbank zum Vokabular hinzugefügt. Das Vokabular besitzt 65536 Einträge und hat auf den HUB4'97-Testdaten eine OOV-Rate von 0.48%. Die Perplexitäten der erhaltenen Sprachmodelle werden in Tabelle 7.10 zusammengefaßt. Für das Bigramm erhält man durch Hinzufügen von Phrasen (np-bg $\rightarrow$ bg) eine Perplexitätsreduktion um 9% . Der Übergang zum Trigramm (bg $\rightarrow$ std-tg) führt zu einer Perplexitätsreduktion um 30%. Durch das mit dem Faktor 8 gewichtete Hinzufügen der akustischen Trainingsdaten (TAT, ca. 1.2 Millionen laufende Wörter) zum Textkorpus (std-tg $\rightarrow$ tg) sinkt die Perplexität um weitere 2%. Aus den Tabellen 7.11 und 7.12 wird ersichtlich, daß der Übergang vom Bigrammmodell 'bg' in Paß (III) zum Trigrammmodell 'tg' in Paß (IV) zu einer Fehlerratenreduktion um 6% relativ führt.

7.2.6 Suche

Die Suche nach der gesprochenen Wortfolge verläuft entsprechend einer Multi-Paß-Wortgraph-Erkennungs-Architektur [Aubert$^+$ 1995], [Ney$^+$ 1996], [Odell 1995], bei der von Paß zu Paß jeweils komplexere Modelle und kleinere Wortgraphen eingesetzt werden. Der verwendete Wortgraph-Erkenner basiert auf einem zeitsynchronen wortübergreifenden N-Phon-M-Gramm-Suchalgorithmus, der formal in [Beyerlein$^+$ 1997a] beschrieben ist. Die einstufige Suche wird aus Ressourcengründen mit wortinternen Triphonen und einem Bigrammsprachmodell durchgeführt, wodurch ein Lattice erzeugt werden kann. Der Wortgraph-Erkenner kann dieses Lattice in einem weiteren Paß einlesen, in einen Wortgraphen umwandeln, eine durch den Wortgraphen eingeschränkte Suche durchführen und wieder ein neues Lattice erzeugen. Wortgraph und Lattice unterscheiden sich darin, daß im Wort-

Tabelle 7.10: Perplexitäten für das Bigrammsprachmodell und das Trigrammsprachmodell auf der Evaluationsdatenbasis HUB4'97; das Training der Sprachmodelle erfolgte auf dem Broadcast-News-Text-Korpus (BN) und auf den Transkriptionen der akustischen Traingsdaten (TAT, 46 h)

Modell	Kurzbezeichnung	Perplexität
Bigramm, BN ohne Phrasen	np-bg	236.3
Bigramm, BN mit Phrasen	bg	215.7
Trigramm, BN mit Phrasen	std-tg	149.9
Trigramm, BN + 8·TAT mit Phrasen	tg	146.6

graphen nur noch die potentielle Abfolge der Wörter abgespeichert und die Zeitinformation eliminiert ist. Für die Erkennung der HUB4'97-Daten wird folgendes Multi-Paß-System verwendet:

(I) Eine einstufige Bigrammerkennung liefert die erkannte Wortfolge W_0. Mit Hilfe der Wortfolge W_0 wird die VTN-Transformation der Merkmale durchgeführt. Eine sich anschließende Bigrammerkennung mit VTN-Modellen auf den VTN-transformierten Merkmalen ergibt die erkannte Wortfolge W_1 und das Lattice L_1.

(II) VTN wird mit Hilfe der Wortfolge W_1 wiederholt und die transformierten Merkmale für alle weiteren Erkennungsschritte verwendet.

(III) Es schließt sich eine MLLR-Adaption der wortinternen Triphonverteilungen mit Hilfe der Wortfolge W_1 und der VTN-transformierten Merkmale an. Eine Wortgraph-Erkennung auf dem Lattice L_1 mit den adaptierten Modellen liefert die Wortfolge W_3 und das Lattice L_3.

(IV) Auf dem Lattice L_3 wird ein Trigramm-Rescoring durchgeführt. Anschließend wird das $\mathcal{N}$-Best-Verfahren in [Tran$^+$ 1997] angewendet, um das Lattice L_3 stark auszudünnen. Da die Segmente unter Umständen sehr lang sind und um genügend Variabilität in den $\mathcal{N}$ besten Sätzen zu erhalten, wird das Lattice L_3 an längeren hypothetisierten Pausen in Teillattices aufgebrochen. Auf jedem der Teillattices wird eine $\mathcal{N}$-Best-Dekodierung durchgeführt. Die Anzahl K dieser Teillattices hängt von der Länge des Segmentes und der Anzahl der Pausen im Segment ab. Durch dieses Aufbrechen werden nicht die $\mathcal{N}$ besten Sätze aus dem Lattice L_3 extrahiert, sondern die $\mathcal{N}^K$ besten Sätze. Die $\mathcal{N}^K$ besten Sätze sind im Lattice L_4 kompakt dargestellt.

(V) Auf dem ausgedünnten Lattice L_4 findet nun eine Wortgraph-Erkennung mit wortübergreifenden Triphonen und einem Trigrammsprachmodell statt. Ergebnis ist die wortübergreifende Phonemfolge P_5.

(VI) Mit Hilfe der wortübergreifenden Phonemfolge P_5 wird eine MLLR-Adaption der wortübergreifenden Triphonmodelle durchgeführt, wobei die gleichen Einstellungen wie in Paß (III) verwendet werden. Paß (V) wird anschließend mit den adaptierten wortübergreifenden Triphonmodellen wiederholt. Der resultierende Text dient als Erkennungsergebnis.

7.2.7 Erkennungsgenauigkeit

Die Tabellen 7.11 und 7.12 fassen die Erkennungsleistung des beschriebenen Systems auf den HUB4'96- und den HUB4'97-Testdaten zusammen.

Tabelle 7.11: Wortfehlerraten (%) des Philips-Forschungs-Spracherkennungssystems auf der HUB4'96-Testdatenbasis bei vorgegebener manueller Segmentierung (PE), aufgeschlüsselt für die Erkennungspässe (I-VI)

	File1-4	CNN M.N. File1	CSP W.J. File2	NPR T.W. File3	NPR M.P. File4
Paß (I)	35.0	36.7	33.4	39.7	30.3
Paß (III)	32.0	32.7	30.5	36.6	28.5
Paß (IV)	30.2	30.6	28.1	36.2	26.3
Paß (VI)	28.7	29.0	26.4	34.2	25.6

Tabelle 7.12: Wortfehlerraten (%) des Philips-Forschungs-Spracherkennungssystems auf der HUB4'97-Testdatenbasis bei automatischer Segmentierung (UE), aufgeschlüsselt für die Erkennungspässe (I-VI)

	HUB4'97
Paß (I)	29.0
Paß (III)	26.7
Paß (IV)	24.8
Paß (VI)	23.5

7.3 Anwendung von DMC

Dieser Abschnitt beinhaltet Tests, mit denen die Leistungsfähigkeit des DMC-Verfahrens auf den HUB4-Daten validiert wird. Dabei werden insgesamt 8 akustische Modelle und 4 Sprachmodelle miteinander kombiniert. Ausgangspunkt der DMC-Tests ist das in Abschnitt 7.2 beschriebene Spracherkennungssystem für Radio- und Fernsehsprachdaten. Durch die Bereitstellung von mehr Trainingsdaten im Jahre 1998 und durch folgende Modifikationen konnte die Baseline-Fehlerrate des HUB4-Systems von 23.5% auf 20.7% reduziert werden [Beyerlein$^+$ 1999a]:

- Bessere Segmentierung. Verbessert wurden die Grobsegmentierung, die Sprecherwechseldetektion, die Detektion nichtsprachlicher Bereiche und das Segmentclustern.
- Einstufige Trigrammsuche. Die Anzahl der Suchfehler reduzierte sich.
- Besseres Trigrammsprachmodell aufgrund größerer Trainingskorpora.
- Bessere akustische Modelle durch Verdopplung der verwendbaren Trainingsdatenmenge von 46 h auf 96 h, durch Anwendung geschlechtsspezifischer Modelle auf dieser größeren Trainingsdatenbasis sowie durch Überprüfung und Korrektur des Aussprachelexikons und der Trainingsskripten.

Zunächst werden die zu kombinierenden Basismodelle aufgelistet. Diese Modelle werden in Abschnitt 7.2 beschrieben. Sie unterliegen jedoch den eben aufgelisteten Modifikationen. Die Ausgangsfehlerrate, die als Baseline für das DMC-Verfahren dient, ist durch die Fehlerrate des MLLR-adaptierten wortübergreifenden Triphon-Trigramm-Systems gegeben (20.7%).

7.3.1 Basismodelle

Für die Diskriminative Modellkombination werden folgende akustische und Sprachmodelle genutzt:

- ww - wortinterne Triphone (ww),
- wwad - wortinterne Triphone (ww), die MLLR-adaptiert wurden (ad),
- xw - wortübergreifende Triphone (xw),

- xwad - wortübergreifende Triphone (xw), die MLLR-adaptiert wurden (ad),
- 5wwad - wortinterne Pentaphone (5ww)[1], die MLLR-adaptiert wurden (ad),
- tg - phrasenbasiertes Trigramm-Sprachmodell [Beyerlein$^+$ 1999a]
- ug - Unigrammwahrscheinlichkeiten aus Trigramm-Sprachmodell 'tg'
- bg - Bigrammwahrscheinlichkeiten aus Trigramm-Sprachmodell 'tg'
- d1bg - phrasenbasiertes Bigramm-Lücken-Sprachmodell mit einer Lückenlänge von 1 (Trigrammkontext)
- lltg - log-lineare Kombination aus ug, bg, tg, d1bg
- llac - log-lineare Kombination aus dem ww-Modell, xw-Modell und auf der WSJ0+1 - Trainingsdatenbasis trainierten ww-, xw-, 5ww-Modellen, die MLLR-adaptiert wurden. Die WSJ0+1 - Datenbasis ist eine Erweiterung der WSJ0-Datenbasis und enthält ca. 80 h vorgelesene Zeitungstexte [Aubert$^+$ 1994].

Die Sprachmodelle werden mit Hilfe der in Tabelle 7.13 beschriebenen Korpora trainiert. Das Trainingsverfahren ist detailiert in [Beyerlein$^+$ 1999a] beschrieben. Tabelle 7.14 faßt die Perplexitäten der Sprachmodelle zusammen.

Tabelle 7.13: Verwendete Textkorpora für die HUB4-Sprachmodelle, die mit Hilfe von DMC in den Erkenner integriert wurden

Name	Abkürzung	Anzahl Wörter
BN-Archive	BNA	$140 \cdot 10^6$
akustische Trainingstexte	TAT	$1.2 \cdot 10^6$
North American News	NANT	$1.4 \cdot 10^9$

Tabelle 7.14: Perplexität und Parameterzahl aller mit Hilfe von DMC in den Erkenner integrierten HUB4-Sprachmodelle auf den HUB4'96-Testdaten und den HUB4'97-Testdaten

Sprachmodell	Parameter	Perplexität auf HUB4'96	Perplexität auf HUB4'97
ug	siehe tg	760	818
bg	siehe tg	196	194
tg	$40 \cdot 10^6$	136	125
d1bg	$20 \cdot 10^6$	501	500

Tabelle 7.15 beinhaltet schließlich die Eigenschaften der verwendeten akustischen Modelle. Da die akustischen Modelle geschlechtsabhängig sind, werden die Charakteristika für das jeweilige Geschlecht separat aufgeführt. Die Mischverteilungen der HMM-Zustände aller verwendeten akustischen Modelle werden mit Hilfe des in Abschnitt 6.2.2 beschriebenen Maximum-Likelihood-Trainings und der Viterbi-Approximation geschätzt.

Die WSJ0+1 -Modelle haben ähnliche Eigenschaften, da die WSJ0+1 - Trainingdatenbasis eine ähnliche Größe hat wie der HUB4-Korpus und die gleiche Parametrisierung beim Training verwendet wird.

[1] Wortinterne Pentaphone fungieren im Kontext von Phrasen automatisch als wortübergreifende Pentaphone zwischen den in den Phrasen enthaltenen Wörtern

Tabelle 7.15: Anzahl der Cluster, Dichten und Parameter der wortinternen Triphonmodelle (ww), der wortübergreifenden Triphonmodelle (xw) und der wortinternen Pentaphonmodelle (5ww), die auf einem überprüften und korrigierten Teil (96 h) der HUB4'98-Trainingsdatenbasis trainiert werden

Akustisches Modell	Cluster	Dichten	Parameter
ww (männlich)	9300	402000	$14 \cdot 10^6$
ww (weiblich)	7800	291000	$10.1 \cdot 10^6$
xw (männlich)	10700	487000	$17 \cdot 10^6$
xw (weiblich)	8600	343000	$12 \cdot 10^6$
5ww (männlich)	10500	459000	$16.1 \cdot 10^6$
5ww (weiblich)	8200	296000	$10.4 \cdot 10^6$

7.3.2 Lattice-Qualität

Die Verbesserungsmöglichkeiten durch DMC hängen insbesondere auch von der Qualität der verwendeten Lattices (vgl. Abschnitt 6.3.1) ab. Um eine möglichst gute Lattice-Qualität zu erzielen, wurde das Lattice mit Hilfe einer einstufigen Trigrammsuche mit MLLR-adaptierten wortinternen Triphonmodellen erzeugt. Dieses Lattice wird bezüglich des wortübergreifenden Pentaphon-Trigramm Kontextes expandiert und abgespeichert. Um den Speicherbedarf für das Lattice zu limitieren, wird das Lattice während der Expansion so geprunt, daß seine Gesamtgröße erhalten bleibt. Das Pruning erfolgt mit Hilfe des wortübergreifenden Triphon-Trigramm-Systems. Nach Abschluß der Expansion werden die Bewertungen der verschiedenen Basismodelle eingetragen.
Als Qualitätsmaß für das Lattice wird die sogenannte Latticefehlerrate verwendet. Dazu wird derjenige Satz aus dem Lattice extrahiert, der die niedrigste Fehlerrate bezüglich des gesprochenen Textes aufweist. Die Latticefehlerrate ist durch die Fehlerrate dieses Satzes gegeben.
Die in diesem Kapitel angegebenen Wortfehlerraten werden mit der NIST-Fehlerzählungssoftware 'Sclite' gemessen, die bestimmte NIST-Regeln befolgt. Die NIST-Fehlerzählung weicht wie folgt von einer einfachen Fehlerzählung ab:

- Zulassung alternativer Erkennungsergebnisse mit gleicher Bedeutung, wie z.B. *that is* und *that's*,
- Zulassung alternativer Erkennungsergebnisse aufgrund undeutlicher Aussprache, wie z.B. *mexico's*, *mexico is* und *mexico has*,
- Ausschluß bestimmter Passagen des zu erkennenden Signals aus der Fehlerzählung, wie z.B. Wetterberichte.

Leider steht keine äquivalente Software für die Bestimmung der Latticefehlerrate zur Verfügung. Um trotz dieser Einschränkung eine Abschätzung für die Qualität der verwendeten Lattices zu erhalten, wird die Latticefehlerrate mit Hilfe einer Standard-Fehlerzählung bestimmt, bei der keine Zeitabschnitte ausgeschlossen werden und auch keine Alternativen erlaubt sind. Eine weitere Einschränkung besteht darin, daß das Erkennungslexikon Phrasen enthält (vgl. Abschnitt 7.2.4). Deswegen enthält das Lattice zusätzlich zu einfachen Wort-Hypothesen auch Phrasen-Hypothesen. Damit muß die Lattice-Fehlerzählung auf Phrasenebene durchgeführt werden. Um die Abweichungen durch die Verwendung von Phrasen und die NIST-Regeln einschätzen zu können, wird in Tabelle 7.16 die NIST-Fehlerrate, die Phrasen-Fehlerrate und die Phrasen-Lattice-Fehlerrate angegeben. Mit diesen Fehlerraten wird anschließend eine Hochrechnung der Lattice-Fehlerrate nach NIST-Regeln angegeben (Nist-Lattice-Fehlerrate). Die Hochrechung erfolgt über die Gleichung:

$$\text{NIST-Lattice-Fehlerrate} = \frac{\text{NIST-Fehlerrate}}{\text{Phrasen-Fehlerrate}} \cdot \text{Phrasen-Lattice-Fehlerrate}.$$

Tabelle 7.16: Verschiedene Fehlermaße (NIST-Fehlerrate, Standard-Wort-Fehlerrate, Standard-$\mathcal{N}$-Best-Fehlerrate, Phrasen-Fehlerrate, Phrasen-Lattice-Fehlerrate) auf der HUB4'97-Testdatenbasis, Hochrechnung der Lattice-Fehlerrate nach NIST-Regeln

Fehlermaß	Meßwert
NIST-Fehlerrate	20.7%
Phrasen-Fehlerrate	28.2%
Phrasen-Lattice-Fehlerrate	18.4%
Hochrechnung NIST-Lattice-Fehlerrate	$\approx$ 13.5%

7.3.3 Experimentelle Ergebnisse

Die log-lineare Kombination der in Abschnitt 7.3.1 beschriebenen Basismodelle wurde auf den Entwicklungsdaten mit Hilfe des MWE-Verfahrens optimiert. Ähnliche Koeffizienten und damit fast identische Fehlerraten wurden mit dem geschlossenen Lösungsverfahren aus Abschnitt 4.7 beobachtet.
In den Tabellen werden folgende Symbole zur Abkürzung benutzt:

M - Anzahl kombinierter Modelle

G - Wortfehlerratengewinn pro zusätzlichem Parameter, $G = \dfrac{\Delta\text{WER}}{P}$.

Der Gewinn G wird angegeben, um die relative Stärke der einzelnen Systemkomponenten einschätzen zu können. Tabelle 7.17 faßt die Baseline-Ergebnisse für das HUB4-System zusammen. Die bereits genannte Fehlerrate von 20.7% des wortübergreifenden Triphon-Systems dient als Vergleichswert für das DMC-Verfahren.

Tabelle 7.17: Baseline-Wortfehlerrate (%) für die DMC-Tests auf der HUB4'97-Testdatenbasis; das Baselinesystem ist das MLLR-adaptierte wortübergreifende Triphon-Trigramm-System

Basismodelle	WER
ww+tg	23.2
wwad+tg	21.8
xwad+tg	20.7

Tabelle 7.18 zeigt die durch die Kombination der verschiedenen Basismodelle mit dem DMC-Verfahren erreichten Ergebnisse. Die zweite Ergebniszeile der Tabelle zeigt eine Verbesserung um 0.5% durch Interpolation von wortinternen und wortübergreifenden Triphonmodellen. Durch die Hinzunahme der sonst nicht verwertbaren Pentaphonmodelle kann die Fehlerrate um weitere 0.7% reduziert werden (dritte Ergebniszeile). Experimente mit einem eigenständigen wortinternen Pentaphon-Trigramm-System auf den Lattices schlugen fehl, da die Pentaphone nicht robust genug trainiert werden konnten. Damit ergibt sich eine Reduktion der Fehlerrate von 20.7% auf 19.5% allein durch Interpolation der drei genannten akustischen Basismodelle. Werden schließlich die oben genannten Sprachmodelle hinzuinterpoliert, so sinkt die Fehlerrate weiter um 0.6% auf 18.9%. Ein marginaler Gewinn ergibt sich aus der Integration der auf WSJ0+1 traininierten akustischen Basismodelle. Betrachtet man die Modellkombination als Ganzes, so ergibt sich eine Reduktion der Baseline-Fehlerrate um 9.1% relativ. Dabei ist zu beachten, daß die HUB4-Datenbasis einen gewissen Schwierigkeitsgrad besitzt. So liegt hier der Gewinn mit Standardmethoden (wie z.B. MLLR-Adaption, Übergang vom wortinternen auf das wortübergreifende Triphonmodell, Übergang vom Standard-Bigrammsprachmodell zum Standard-Trigrammsprachmodell) bei jeweils ca 6% relativ, obwohl mit diesen Methoden auf einfacheren Datenbasen deutlich höhere Gewinne erzielt werden können. Die Tabellen 7.19, 7.20, 7.21, 7.22, 7.23, 7.24 und 7.25 schlüsseln die oben diskutierten Ergebnisse für die verschiedenen F-Konditionen auf. Dabei ist bemerkenswert, daß der Gewinn durch das DMC-Verfahren auf der F0-Kondition, die den größten Anteil der Daten beinhaltet, 13.1% relativ beträgt. Vergleicht man die DMC-Ergebnisse mit der Latticefehlerrate (vgl. Abschnitt 7.3.2), dann wird deutlich, daß der Erkenner auch nach Anwendung einer optimierten Modellkombination signifikant verbessert werden kann.

Tabelle 7.18: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis

DMC-Basismodelle	M	WER (%)	G
xwad+tg (Baseline)	2	20.7	-
wwad+xwad+tg	3	20.2	0.25
wwad+xwad+5wwad+tg	4	19.5	0.4
wwad+xwad+5wwad+lltg	7	18.9	0.26
wwad+xwad+5wwad+lltg+llac	10	18.8	0.19

Tabelle 7.19: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition F0

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	13.0	-
wwad+xwad+tg	3	12.4	0.3
wwad+xwad+5wwad+tg	4	11.8	0.4
wwad+xwad+5wwad+lltg	7	11.4	0.23
wwad+xwad+5wwad+lltg+llac	10	11.3	0.17

Tabelle 7.20: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition F1

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	18.9	-
wwad+xwad+tg	3	18.7	0.1
wwad+xwad+5wwad+tg	4	18.2	0.23
wwad+xwad+5wwad+lltg	7	17.6	0.19
wwad+xwad+5wwad+lltg+llac	10	17.7	0.12

Tabelle 7.21: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition F2

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	29.9	-
wwad+xwad+tg	3	29.2	0.35
wwad+xwad+5wwad+tg	4	27.9	0.67
wwad+xwad+5wwad+lltg	7	27.1	0.4
wwad+xwad+5wwad+lltg+llac	10	26.8	0.31

Tabelle 7.22: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition F3

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	31.5	-
wwad+xwad+tg	3	30.4	0.55
wwad+xwad+5wwad+tg	4	30.6	0.3
wwad+xwad+5wwad+lltg	7	30.3	0.17
wwad+xwad+5wwad+lltg+llac	10	30.3	0.12

Tabelle 7.23: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition F4

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	23.3	-
wwad+xwad+tg	3	22.8	0.25
wwad+xwad+5wwad+tg	4	22.1	0.4
wwad+xwad+5wwad+lltg	7	21.6	0.24
wwad+xwad+5wwad+lltg+llac	10	21.4	0.19

Tabelle 7.24: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition F5

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	23.6	-
wwad+xwad+tg	3	22.6	0.5
wwad+xwad+5wwad+tg	4	20.8	0.93
wwad+xwad+5wwad+lltg	7	19.7	0.56
wwad+xwad+5wwad+lltg+llac	10	19.4	0.42

Tabelle 7.25: Wortfehlerraten (%) für die Kombination verschiedener akustischer Modelle und Sprachmodelle mit DMC auf der HUB4'97-Testdatenbasis: Kondition FX

DMC-Basismodelle	M	WER	G
xwad+tg (Baseline)	2	35.7	-
wwad+xwad+tg	3	35.6	0.05
wwad+xwad+5wwad+tg	4	35.1	0.2
wwad+xwad+5wwad+lltg	7	34.4	0.18
wwad+xwad+5wwad+lltg+llac	10	34.1	0.16

7.4 Zusammenfassung

Der DMC-Ansatz (Abschnitt 4.9) wurde experimentell auf der HUB4-Datenbasis getestet:

A: Aus akustischen Modellen $p_i(x|k), i = 1, \ldots, 6$

i=1: wortinterne Triphone, Training auf HUB4, MLLR-Adaption

i=2: wortübergreifende Triphone, Training auf HUB4, MLLR-Adaption

i=3: wortinterne Pentaphone, Training auf HUB4, MLLR-Adaption

i=4: wortinterne Triphone, Training auf WSJ0+1, MLLR-Adaption

i=5: wortübergreifende Triphone, Training auf WSJ0+1, MLLR-Adaption

i=6: wortinterne Pentaphone, Training auf WSJ0+1, MLLR-Adaption

und Sprachmodellen $p_j(k), j = 1, \ldots, 4$

j=1: Unigrammsprachmodell

j=2: Bigrammsprachmodell

j=3: Trigrammsprachmodell

j=4: Bigramm-Lücken-Sprachmodell mit einer Lückenlänge von 1 (Trigrammkontext)

wurde die log-lineare Modellkombination

$$p_{\Lambda}(k|x) = \frac{\prod_i p_i(x|k)^{\lambda_i} \prod_j p_j(k)^{\lambda_j}}{\sum_{k'} \prod_i p_i(x|k')^{\lambda_i} \prod_j p_j(k')^{\lambda_j}}$$

gebildet.

B: Als diskriminatives Trainingskriterium wurde

$$E(\Lambda) = \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) S(k, n, \Lambda)$$

verwendet, wobei die Funktion S entsprechend den Kriterien B.2 und B.4 (Abschnitt 4.9) gewählt wurde (beide Kriterien führten zu nahezu gleichen Ergebnissen). Die Menge der rivalisierenden Wortfolgen $k = 1, \ldots, K, k \neq k_n$ wurde mit Hilfe einer $\mathcal{N}$-Best-Erkennung ($\mathcal{N} = 500, \ldots, 700$) [Tran$^+$ 1997] definiert. Diese Erkennung wurde mit Hilfe des MLLR-adaptierten wortübergreifenden Triphonmodells und des Trigrammsprachmodells durchgeführt.

Die zu Beginn der vorliegenden Arbeit gegebene Baseline-Fehlerrate des MLLR-adaptierten wortinternen Triphon-Systems (Tabelle 7.17) konnte durch den zusätzlichen Einsatz von

- wortübergreifenden Triphonmodellen (xw)
- Lückensprachmodell (d1bg)
- DMC

um 13% von 21.8% auf 18.9% reduziert werden. Der Hauptanteil des Gewinns (9.1% relativ) konnte dabei mit DMC, d.h. mit der optimalen log-linearen Kombination von vortrainierten kontextabhängigen akustischen Modellen und Sprachmodellen erreicht werden. Der Gewinn durch DMC beruht im wesentlichen auf der Optimierung von 10 freien Parametern, die jedoch die Wirkung aller 140 Millionen Basismodellparamter auf die Entscheidungsregel direkt beeinflussen.
Obwohl Experimente mit einem eigenständigen Pentaphon-System nicht zu einer Verbesserung der Baseline-Fehlerrate führten, konnte mit Hilfe von DMC durch die Integration der Pentaphonmodelle in die Gesamtverteilung ein Fehlerratengewinn erzielt werden.

Kapitel 8

Diskussion Alternativer Kombinationsverfahren

8.1 Kombination von Spracherkennern

Für die Kombination der am HUB4-Benchmarking teilnehmenden Spracherkennungssysteme wurde ROVER ('Recognizer Output Voting Error Reduction') [Fiscus 1997] entwickelt. ROVER kombiniert die erkannten Texte $T_1, \ldots, T_N$ der N verschiedenen unabhängigen Spracherkennungssysteme mit dem Ziel, einen Text geringerer Fehlerrate zu produzieren. Dazu wird der Output der verschiedenen Spracherkennungssysteme in einem dünnen Wortgraphen zusammengefaßt, wobei identische Kanten, d.h. identische Worthypothesen mit gleichem Start- und Endpunkt im Graphen, mehrfach aufgenommen werden. Der Graph wird mit Hilfe einer dynamischen Programmierung erzeugt, bei der sukzessive der Outputtext T_{n+1} an den bereits aus den Texten $T_1, \ldots, T_n$ bestehenden Wortgraphen angepaßt wird. Für jede Kantenposition im Graphen wird nun über ein einfaches Votingverfahren das 'sicherste' Wort bestimmt. Mit Hilfe einer dynamischen Programmierung wird nun wiederum die sicherste Wortfolge selektiert.
Für die Kombination von verschiedenen Spracherkennungssystemen kann auch DMC verwendet werden. Der folgende Abschnitt vergleicht beide Verfahren. In den Vergleichsexperimenten wird die öffentlich zugängliche NIST-ROVER Software verwendet.

Voting mit ROVER

Für die Vergleichsexperimente zwischen ROVER und DMC werden folgende akustische und Sprachmodelle ausgewählt:

- wwad - wortinterne Triphone (ww), die MLLR-adaptiert wurden (ad),
- xwad - wortübergreifende Triphone (xw), die MLLR-adaptiert wurden (ad),
- 5wwad - wortinterne Pentaphone (ww), die MLLR-adaptiert wurden (ad),
- ug - phrasenbasiertes Unigramm-Sprachmodell,
- bg - phrasenbasiertes Bigramm-Sprachmodell,
- tg - phrasenbasiertes Trigramm-Sprachmodell,
- d1bg - phrasenbasiertes Lücken-Bigramm-Sprachmodell.

Die Modelle wurden in Abschnitt 7.3.1 beschrieben.

Bevor ROVER angewendet werden kann, müssen zunächst aus den gegebenen Basismodellen eigenständige Spracherkenungssysteme gebildet werden. Folgende Systeme mit konventioneller Architektur - (genau ein akustisches Modell, genau ein Sprachmodell) - werden konstruiert: *wwad+ug, wwad+bg, wwad+tg, wwad+d1bg,*

cwad+ug, cwad+bg, cwad+tg, cwad+d1bg, 5wwad+ug, 5wwad+bg, 5wwad+tg und *5wwad+d1bg* . Die mit ROVER und DMC durchgeführten Tests werden wie folgt bezeichnet:

- Test A = Interpolation des xwad+tg-Systems mit dem wwad+tg-System,
- Test B = Interpolation des xwad+tg-, wwad+tg- und 5wwad+tg-Systems,
- Test C = Interpolation aller 12 Systeme.

In Tabelle 8.1 werden die Fehlerraten dieser Systemkombinationen verglichen. Die ersten beiden Ergebniszeilen beinhalten die Baseline-Fehlerraten für das wortinterne Triphon-System und das wortübergreifende Triphon-System auf den verwendeten Lattices. Unabhängig davon zeigen die dritte und die vierte Ergebniszeile, daß die Interpolation der drei akustischen Modelle mit DMC die Fehlerrate von 20.7% auf 19.5% senkt. Mit ROVER wird eine Fehlerrate von 19.9% erreicht. Fügt man noch die Sprachmodelle ug, bg, d1bg hinzu (Test C), so sinkt die Fehlerrate mit DMC weiter auf 18.9% ab, während sie mit dem ROVER-Voting auf 20.2% ansteigt. Die drei hinzugefügten Sprachmodelle haben deutlich höhere Perplexitäten als das Trigrammmodell. Die Spracherkenner *wwad+ug, wwad+bg, wwad+d1bg, cwad+ug, cwad+bg, cwad+d1bg, 5wwad+ug, 5wwad+bg* und *5wwad+d1bg* sind folglich nicht so leistungsfähig wie die tg-Systeme *wwad+tg, cwad+tg, 5wwad+tg*. Die Kombination der Texte der tg-Systeme mit den Texten der ug/bg/d1bg-Systeme führt zwangsläufig zu einer höheren Gesamtfehlerrate. Dieses Beispiel macht deutlich, daß das ROVER-Voting nur für die Kombination von robust trainierten Gesamtsystemen mit ähnlicher Erkennungsleistung und unabhängiger Fehlerverteilung geeignet ist. ROVER ist nicht geeignet, um *aus Erkennungssicht unvollständige Modelle* zu integrieren, wie. z.B. das Lückensprachmodell.

Tabelle 8.1: Vergleich von ROVER und DMC auf der HUB4'97-Testdatenbasis; Verglichen werden die Interpolation des xwad+tg-Systems mit dem wwad+tg-System (Test A), die Interpolation des xwad+tg-, wwad+tg- und 5wwad+tg-Systems (Test B), die Interpolation von 12 Systemen (Test C).

Basismodelle	DMC	ROVER - Voting
wwad+tg	21.6	nicht sinnvoll
xwad+tg	20.7	nicht sinnvoll
Test A	20.2	22.5
Test B	19.5	19.9
Test C	18.9	20.2

Die Tabellen 8.2, 8.3 und 8.4 schlüsseln die Tests A-C nach den verschiedenen F-Konditionen des HUB4-Korpus auf.

Tabelle 8.2: Vergleich von ROVER und DMC auf der HUB4'97-Testdatenbasis, Test A, Interpolation des xwad+tg-Systems mit dem wwad+tg-System

Set	DMC Interpolation	ROVER Voting
F0	12.4	14.0
F1	18.7	20.6
F2	29.2	32.7
F3	30.4	35.4
F4	22.8	24.6
F5	22.6	24.5
FX	35.6	39.8
All	20.2	22.5

Aus den Tabellen 8.2, 8.3, 8.4 wird ersichtlich, daß die log-lineare Kombination mittels DMC unabhängig von der F-Kondition bessere oder ähnliche Fehlerraten liefert als das ROVER-Voting-Verfahren.

Tabelle 8.3: Vergleich von ROVER und DMC auf der HUB4'97-Testdatenbasis, Test B, Interpolation des xwad+tg-, wwad+tg- und 5wwad+tg-Systems

Set	DMC Interpolation	ROVER Voting
F0	11.8	12.1
F1	18.2	18.4
F2	27.9	28.5
F3	30.6	31.4
F4	22.1	22.6
F5	20.8	21.1
FX	35.1	36.6
All	19.5	19.9

Tabelle 8.4: Vergleich von ROVER und DMC auf der HUB4'97-Testdatenbasis, Test C, Interpolation von 12 Systemen

Set	DMC Interpolation	ROVER Voting
F0	11.3	12.9
F1	17.7	18.9
F2	26.8	28.5
F3	30.3	31.7
F4	21.4	22.7
F5	19.4	21.8
FX	34.1	35.0
All	18.9	20.2

Gewichtetes Voting

ROVER kann beim Voting auch vorgegebene Konfidenzmaße berücksichtigen. Diese Konfiguration kann als vereinfachtes DMC-Erkennungsverfahren interpretiert werden:

- Die Gewichtung der einzelnen Worthypothesen muß durch den Nutzer von ROVER über sogenannte Konfidenzmaße bereitgestellt werden. Der wesentliche Inhalt von DMC besteht in der Bestimmung solcher Gewichtungen.
- Verarbeitung sehr dünner Wortgraphen,
- Einschränkung auf die Kombination vollständiger Spracherkennungssysteme. Es ist zum Beispiel mit ROVER nicht sinnvoll, ein einzelnes akustisches Modell und ein einzelnes Sprachmodell zu kombinieren, da insbesondere das Sprachmodell unabhängig von der gesprochenen Äußerung ist. Umgekehrt kann DMC sowohl einzelne akustische und Sprachmodelle, als auch vollständige Spracherkennungssysteme als Basismodelle verwenden.

Da sich die vorliegende Arbeit auf die Kombination von Basismodellen konzentriert und wegen der genannten Beziehung der beiden Verfahren wird auf experimentelle Vergleiche zwischen dem gewichteten Voting und der Diskriminativen Modellkombination verzichtet.

8.2 Lineare Modellkombination

In diesem Abschnitt wird die log-lineare Kombinationsform mit der linearen Kombinationsform verglichen. Es seien N Trainingsbeobachtungen $x_n, n = 1, \ldots, N$ gegeben. Diese werden in K Klassen $k = 1, \ldots, K$ klassifiziert. Für jede Trainingsbeobachtung x_n bezeichnet $k = k_n$ die korrekte Klasse und $k \neq k_n$ eine der $K - 1$ rivalisierenden Klassen. Außerdem seien die M Basismodelle $p_j(k|x_n), j = 1, \ldots, M$ gegeben, die in eine Gesamtverteilung $p_{\Lambda}(k|x_n)$ integriert werden sollen. Durch die veränderte Kombinationsform

$$p_{\Lambda}(k|x_n) = \frac{\sum_{j=1}^{M} \lambda_j \cdot p_j(k|x_n)}{\sum_{j=1}^{M} \lambda_j} \tag{8.1}$$

müssen die Gleichungen für die Bestimmung der optimalen Koeffizienten der Modellkombination angepaßt werden. Dabei wird analog zur Optimierung der log-linearen Kombination in Abschnitt 4.5 vorgegangen. Das Optimierungskriterium Wortfehlerrate $E(\Lambda)$ (3.14) wird wie in Abschnitt 4.5 durch folgende geglättete Zielfunktion approximiert:

$$E(\Lambda) = \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) S(k, n, \Lambda). \tag{8.2}$$

Ableitung von $E(\Lambda)$ nach Λ (siehe Anhang 11.2.2) ergibt:

$$\begin{aligned} \frac{\partial E(\Lambda)}{\partial \lambda_j} = & \frac{\eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} S(k, n, \Lambda) \left(\mathcal{L}(k, k_n) - \overline{\mathcal{L}(k_n, \Lambda)} \right) \cdot \\ & \cdot \left(\frac{p_j(k|x_n)}{\sum_{i=1}^{M} \lambda_i p_i(k|x_n)} - \frac{p_j(k_n|x_n)}{\sum_{i=1}^{M} \lambda_i p_i(k_n|x_n)} \right), \end{aligned} \tag{8.3}$$

wobei

$$\overline{\mathcal{L}(k_n, \Lambda)} = \sum_{k' \neq k_n} S(k', n, \Lambda) \mathcal{L}(k', k_n) \tag{8.4}$$

die geglättete Fehleranzahl für die Trainingsbeobachtung x_n ist. Das resultierende Iterationsverfahren lautet schließlich:

$$\begin{aligned} \lambda_j^{(0)} &= 0 \quad \text{(Gleichverteilung)} \\ \lambda_j^{(I+1)} &= \lambda_j^{(I)} - \varepsilon \cdot \eta \sum_{n=1}^{N} \sum_{k \neq k_n} S(k, n, \Lambda^{(I)}) \left(\mathcal{L}(k, k_n) - \overline{\mathcal{L}(k_n, \Lambda^{(I)})} \right) \cdot \\ & \quad \cdot \left(\frac{p_j(k|x_n)}{\sum_{i=1}^{M} \lambda_i^{(I)} p_i(k|x_n)} - \frac{p_j(k_n|x_n)}{\sum_{i=1}^{M} \lambda_i^{(I)} p_i(k_n|x_n)} \right) \\ \Lambda^{(I)} &= (\lambda_1^{(I)}, \ldots, \lambda_M^{(I)})^T \end{aligned} \tag{8.5}$$

$$j = 1, \ldots, M. \tag{8.6}$$

Bevor die experimentellen Ergebnisse diskutiert werden, wird auf [Klakow 1998b] eingegangen. In dieser Arbeit wird deutlich gemacht, daß die log-lineare Interpolation von Sprachmodellen zu besseren Ergebnissen führt als die lineare Kombination. Als Optimierungskriterium wird in [Klakow 1998b] die Likelihood auf Crossvalidierungsdaten verwendet. Ein analoger Vergleich wurde auch für die Interpolation von wortinternen Triphonmodellen und wortübergreifenden Triphonmodellen durchgeführt [Beulen 1999b]. Auch hier zeigte die lineare Interpolation keinen Vorteil gegenüber der log-linearen Interpolation. Die Aussagen in [Klakow 1998b] und [Beulen 1999b]

bestätigen die folgenden in Tabelle 8.5 zusammengefaßten Experimente. Die Experimente wurden auf den 500–700 $\mathcal{N}$-Best-Hypothesen durchgeführt, die mit Hilfe der log-linearen Modellkombination ausgewählt wurden.
Zunächst wurde eine log-lineare Kombination und eine lineare Kombination aller vorhandenen Basismodelle durchgeführt. Die lineare Kombination zeigt eine um 0.7% höhere Fehlerrate als die log-lineare Kombination.
In weiteren Tests wurde ein Teil der Modelle log-linear und ein anderer Teil der Modelle linear kombiniert (hybride Kombination). Die Koeffizienten für die log-lineare Kombination wurden dabei getrennt von den Koeffizienten für die lineare Kombination optimiert.
Zunächst wurde eine log-lineare Kombination der Sprachmodelle und eine lineare Kombination der akustischen Modelle gebildet. Diese hybride Kombination hat eine um 1.0% höhere Fehlerrate als die log-lineare Kombination.
Die lineare Kombination aller Sprachmodelle und log-lineare Kombination aller akustischen Basismodelle führte zu einer um 1.2% höheren Fehlerrate als die log-lineare Kombination.
Alle drei Versuche zeigen, daß eine log-lineare Kombination auf der Ebene der Basismodelle leistungsfähiger ist als eine lineare Kombination.

Tabelle 8.5: Vergleich der Wortfehlerraten (%) der log-linearen, linearen und hybriden Kombination der akustischen Modelle und Sprachmodelle auf der HUB4'97-Testdatenbasis; die lineare und die hybride Kombination wird auf den 500–700 $\mathcal{N}$-Best-Hypothesen durchgeführt, die mit der log-linearen Modellkombination ausgewählt wurden.

Interpolationsform	WER (%)
alle Modelle log-linear	18.8
alle Modelle linear	19.5
akustische Modelle linear, Sprachmodelle log-linear	19.8
akustische Modelle log-linear, Sprachmodelle linear	20.0

8.3 Zusammenfassung

Die log-lineare Kombination liefert bei der Kombination von Basismodellen auf der HUB4-Datenbasis bessere Ergebnisse als die lineare Kombination. Die Ursachen dafür können wie folgt zusammengefaßt werden:

- Die verwendeten a-posteriori Wahrscheinlichkeiten werden auf der Satzebene bestimmt, dadurch findet sich für jedes spezielle Basismodell eine spezielle Hypothese, deren Pfadwahrscheinlichkeit im Verhältnis zu den anderen Hypothesen einen großen Wert annimmt, während die Pfadwahrscheinlichkeiten der anderen Hypothesen verschwinden. Durch die lineare Interpolation der Basismodellbewertungen findet damit im wesentlichen ein Voting statt.

- Im Gegensatz dazu erreicht man durch die Logarithmierung bei der log-linearen Kombination der Basismodelle auf Satzebene tatsächlich einen Glättungseffekt.

- Akustische Basismodelle und Sprachmodelle sind aus theoretischer Sicht statistisch unabhängig, d.h. ihre Verbundverteilung wird durch ein Produkt der Einzelverteilungen gebildet. Das legt den Vorteil einer log-linearen Kombination dieser beiden Basismodelltypen nahe.

Für die Kombination auf der Ebene der erkannten Texte von Spracherkennungssystemen bietet sich das ROVER-Voting-Verfahren an. Vergleicht man die erhaltenen Ergebnisse mit DMC, so zeigt DMC eine stärkere Reduktion der Fehlerrate. Das liegt unter anderem an folgenden Gründen:

- DMC bestimmt automatisch die optimalen Gewichte der Kombination der Basismodelle, während beim Voting eine einfache Mehrheitsentscheidung getroffen wird.

- DMC arbeitet direkt auf Wortgraphen, die vom Spracherkenner erzeugt werden, während ROVER durch die Kombination von erkannten Texten einer begrenzten Anzahl von vollständigen Spracherkennungssystemen nur eine begrenzte Auswahl an möglichen Satzhypothesen hat.

Durch die in diesem Abschnitt durchgeführten Kontrastuntersuchungen wird die Eignung von DMC für die optimale Kombination von Basismodellen deutlich.

Teil IV

Ausblick und Zusammenfassung

Kapitel 9

Ausblick - Weitere Anwendungen

DMC kann auf jeder hierarchischen Ebene eines Mustererkennungssystems angewendet werden. Für ein Spracherkennungssystem bieten sich unter anderem die Satzebene, die Phonemebene und die Zustandsebene an. Die Satzebene wurde bereits in den Kapiteln 4 bis 6 eingehend behandelt.

9.1 Phonemebene

9.1.1 Kombination von Phonemmodellen

Da Phoneme der Unterscheidung der Aussprache von verschiedenen Wörtern dienen, liegt eine Anwendung von DMC auf der Phonemebene nahe. Dazu werden Phonemmodelle als Basismodelle verwendet. Die akustische Äußerung wird in die zu den einzelnen Phonemen gehörenden Abschnitte segmentiert. Die Segmentierung kann über ein Viterbi-Alignment bestimmt werden. Formal wird die Log-Likelihood der Satzhypothese k in die Summe der Log-Likelihoods der einzelnen Phonemhypothesen h, die in k auftreten (kurz $h \in k$), umgeschrieben:

$$\log p(x|k) \quad = \quad \sum_{h \in k} \log p(x^h|h). \tag{9.1}$$

Die zugehörige log-lineare a-posteriori Verteilung lautet:

$$p_{\Lambda}(k|x) \quad = \quad \frac{p(k) \prod_{h \in k} p(x^h|h)^{\lambda_h}}{\sum_{k'} p(k') \prod_{h \in k'} p(x^h|h)^{\lambda_h}} \,. \tag{9.2}$$

Die Gewichte λ_h bestimmen den Einfluß der einzelnen Phoneme auf die Gesamtverteilung und können mit den in Kapitel 4 beschriebenen Verfahren optimiert werden.

9.1.2 Kombination von Phonemklassenmodellen

Um einen Mangel an Trainingsdaten zu kompensieren, kann auch eine gröbere Modellierung verwendet werden. Eine sinnvolle Strukturierung besteht z.B. in der Bildung der 3 Modelle: Vokalmodell (V), Konsonantenmodell (C), Pausenmodell (S):

$$\log p_V(x|k) \quad = \quad \sum_{h \in k} \delta(h, V) \log p(x^h|h) \tag{9.3}$$

$$\log p_C(x|k) \quad = \quad \sum_{h \in k} \delta(h, C) \log p(x^h|h) \tag{9.4}$$

$$\log p_S(x|k) \quad = \quad \sum_{h \in k} \delta(h, S) \log p(x^h|h), \tag{9.5}$$

wobei alle Bewertungen von Vokalen eines Satzes im Vokalmodell $p_V(x|k)$, alle Bewertungen von Konsonanten eines Satzes im Konsonantenmodell $p_C(x|k)$ und alle Pausen-Bewertungen eines Satzes im Pausenmodell $p_S(x|k)$ aufsummiert werden. Die Log-Likelihood-Bewertung der Satzhypothese lautet in diesem Falle:

$$\log p(x|k) = \log p_V(x|k) + \log p_C(x|k) + \log p_S(x|k). \tag{9.6}$$

Mit Hilfe dieser Dekomposition, kann wieder eine log-lineare Verteilung gebildet werden:

$$p_\Lambda(k|x) \quad = \quad \frac{p(k)^{\lambda_{LM}} p_V(x|k)^{\lambda_V} p_C(x|k)^{\lambda_C} p_S(x|k)^{\lambda_S}}{\sum_{k'} p(k')^{\lambda_{LM}} p_V(x|k')^{\lambda_V} p_C(x|k')^{\lambda_C} p_S(x|k')^{\lambda_S}}\,, \tag{9.7}$$

wobei λ_{LM} das Gewicht des Sprachmodells, λ_V das Gewicht des Vokalmodells, λ_C das Gewicht des Konsonantenmodells und λ_S das Gewicht des Pausenmodells sei.
Die Idee hinter dieser Verteilungsform besteht darin, daß die Vokale, Konsonanten und die Pausen eine unterschiedliche Bedeutung für die Klassifikation besitzen und deswegen unterschiedlich gewichtet werden sollten. Mit dieser Definition ist das DMC-Verfahren aus Kapitel 4 direkt anwendbar. Als Modelle bieten sich hier analog zur Satzebene kontextabhängige akustische Modelle und auch Sprachmodelle auf Phonemebene an. In Abschnitt 9.1.3 wird ein Anwendungsbeispiel dieser Verteilungsform betrachtet.

9.1.3 Sprachenunabhängigkeit

DMC wurde während des Johns Hopkins Sommerworkshops 1999 innerhalb des Projekts “Towards Language Independent Speech Recognition” eingesetzt [Beyerlein$^+$ 1999b]. Hier wurden die akustischen Modelle einer Zielsprache (z.B. Tschechisch) durch eine Kombination der Modelle verschiedener Quellsprachen (Weltsprachen wie z.B. Englisch, Spanisch, Russisch) approximiert. Ziel einer solchen Approximation ist die Konstruktion eines Spracherkenners in einer Zielsprache, für die nur eine kleine Trainingsdatenmenge vorhanden ist. Dabei sollen die großen Korpora der Quellsprachen optimal ausgenutzt werden. Die Aufgabe bestand beispielhaft darin, ein tschechisches Broadcast-News System zu konstruieren, welches mit einer geringen Menge tschechischer akustischer Daten und einer großen Menge von Hintergrunddaten aus anderen Sprachen, wie Englisch, Spanisch und Russisch trainiert werden soll. Dabei standen folgende zwei Fragestellungen im Mittelpunkt der Untersuchungen:

- Ist es prinzipiell möglich die Daten anderer Sprachen zu nutzen, um ein System in der Zielsprache zu verbessern ?
- Wieviel Trainingsdaten in der Zielsprache können eingespart werden, wenn akustische Modelle mehrerer Weltsprachen zur Verfügung stehen ?

Auf diese beiden Fragen wird in den folgenden Abschnitten eingegangen. Die DMC-Implementierung auf dem Johns Hopkins Sommerworkshop, wurde der Einfachheit halber auf N-Best Listen beschränkt. Das ermöglichte eine klare Schnittstellendefinition für das Projekt und eine vereinfachte Behandlung der Hypothesenmenge.

9.1.3.1 Die VOA-Aufgabe

Als Testdaten wurden tschechische Aufzeichnungen des Senders ’Voice-Of-America (VOA)’ aus dem 2. Quartal des Jahres 1999 verwendet. Diese Daten beinhalten im wesentlichen die Lewinski-Affäre und den Kosovo-Krieg. Um den Datenüberlapp zwischen Trainings- und Testdaten möglichst klein zu halten, wurde zum akustischen Training eine Stunde gelesener Daten in Tschechisch verwendet. Die Nutzung gelesener Daten entspricht einem realistischen Szenario beim Aufbau eines Systems in einer neuen Sprache. Es wurden außerdem akustische Modelle in den Sprachen Spanisch, Russisch und Englisch trainiert. Dabei wurden für Spanisch und Englisch Broadcast-News Daten verwendet. Die russische Datenbasis besteht aus ca. 3 Stunden russischer Militärsprache. Die spanischen, russischen und englischen Phonemmodelle wurden mit Hilfe von menschlichem Wissen (Joe Picone), auf die tschechischen Phonemmodelle abgebildet. Man beachte, daß diese Abbildungen zwar aus linguistischer Sicht optimal sein können, daß dabei jedoch Kanalunterschiede zwischen den Datenbasen ignoriert werden, die mit Hilfe statistischer Abbildungsverfahren kompensiert werden könnten. Statistische Verfahren standen erst am Ende des

Tabelle 9.1: Erkennungsleistung des spanischen, russischen und englischen Systems sowie des tschechischen Baseline-Monophon-Systems, eines tschechischen Triphonsystems und eines tschechischen Triphonesystems, welches auf 10h Sprachdaten trainiert wurde

model	WER in %
10h Spanisch, Broadcast News, Monophone	71.6
3h Russisch, Militärkommandos, Monophone	60.8
10h English, Broadcast News, Triphone, adaptiert an 1h Tschechisch	35.1
1h Tschechisch, Monophone, gelesen	33.4

Workshops zur Verfügung.
Mit Hilfe des tschechischen Monophon-Systems wurden 1000-Best Listen dekodiert, und diese wurden mit den verschiedenen akustischen Modellen bewertet. Diese Vorbereitungsarbeit wurde von Bill Byrne und Sanjeev Khudanpur geleistet. Als Erkennungssoftware wurde während des Workshops ein AT&T-Dekoder und die AT&T-FSM-Tools verwendet. Das DMC-Training und die DMC-Erkennung wurde mit Hilfe der Held-Out Methode auf der Testdatenbasis durchgeführt, da keine Entwicklungsdaten zur Verfügung standen.

9.1.3.1.1 Multilinguale System-Kombination Zunächst betrachten wir eine Kombination der akustischen Modelle der vier verwendeten Sprachen auf der Systemebene, d.h. die Satzhypothesen werden jeweils vollständig durch die akustischen Modelle der einzelnen Sprachen bewertet und die erhaltenen Bewertungen anschließend mit DMC kombiniert. Die Kombination der Modelle der verschiedenen Sprachen auf Systemebene lautet wie folgt:

$$\begin{aligned}\log p_{\Lambda}(k|x) &= C(\Lambda,x) + \lambda_{LM} L_{cz}(k) \\ &\quad + \lambda_{cz} A_{cz}(x|k) + \lambda_{sp} A_{sp}(x|k) \\ &\quad + \lambda_{ru} A_{ru}(x|k) + \lambda_{en} A_{en}(x|k), \end{aligned} \tag{9.8}$$

wobei $L_{cz}(k)$ das tschechische Sprachmodell, $A_{cz}(x|k)$ das tschechische akustische Modell, $A_{sp}(x|k)$ das spanische akustische Modell, $A_{ru}(x|k)$ das russische akustische Modell und $A_{en}(x|k)$ das englische akustische Modell ist.
Die Leistung der einzelnen akustischen Modelle in einer freien Erkennung zusammen mit dem tschechischen Sprachmodell ist in Tabelle 9.1 zusammengefaßt. Man beachte, daß das englische Triphonsystem zuvor an die tschechischen Trainingsdaten adaptiert wurde. Zusätzlich zeigt die Tabelle die Erkennungsleistung des tschechischen Baseline-Monophon-Systems, welches auf einer Stunde gelesener Daten trainiert wurde. Die Systeme in Tabelle 9.1 wurden auf Systemebene miteinander kombiniert. Die entsprechenden Erkennungsergebnisse befinden sich in Tabelle 9.2. Die Tabelle zeigt, daß sowohl das spanische als auch das russische Modell leicht zu einer Verbesserung beitragen. Durch das englische Modell wurde eine signifikante Verbesserung erzielt. Die Gesamtfehlerrate konnte von 33.4% auf 29.2% reduziert werden. Die Stärke der Verbesserung ist überraschend, da hier lediglich 4 freie Parameter optimiert wurden. Um die erreichte Fehlerrate von 29.2% besser bewerten zu können, wurden tschechische Triphonsysteme auf einer Stunde und auf 10 Stunden akustischer Daten trainiert. Die beiden System erreichten Fehlerraten von 30.7% bzw. von 27.1%. Aus diesem Vergleich ist zu ersehen, daß die DMC-Kombination ein tschechisches Triphonsystem unter gleichen Bedingungen schlägt und sich in der Leistungsfähigkeit einem auf 10 Stunden Trainingsdaten konstruierten tschechischen Triphonsystem annähert. Eine Integration der beiden tschechischen Triphonsysteme in die DMC-Kombination war aus Zeitgründen während des Workshops nicht mehr möglich.

In Tabelle 9.3 werden schließlich die Koeffizienten der DMC-Kombination aller gegebenen akustischen Modelle zusammengefaßt (vgl. Tabelle 9.2).

9.1.3.1.2 Multilinguale Phonemklassen-Kombination In diesem Abschnitt wird die oben beschriebene Anwendung von DMC verfeinert, indem die akustischen Modelle der einzelnen Sprachen in den Vokal-, den Konsonanten- und den Pausenbereich aufgeteilt werden (vgl. Abschnitt 9.1.2). Dadurch entstehen aus den drei Sprachen

Tabelle 9.2: DMC Ergebnisse auf VOA, 1000-Best Listen, Nutzung von wissensbasierten Phonemabbildungen, Kombination des tschechischen Sprachmodells $L_{cz}(k)$, des tschechischen akustischen Modells $A_{cz}(x|k)$, des spanischen akustischen Modells $A_{sp}(x|k)$, des russischen akustischen Modells $A_{ru}(x|k)$ und des englischen akustischen Modells $A_{en}(x|k)$.

	WER in %
Minimale N-Best Fehlerrate	19.8
Maximale N-best Fehlerrate	56.6
Baseline	33.4
$L_{cz} + A_{cz}$	32.7
$L_{cz} + A_{cz} + A_{ru}$	32.5
$L_{cz} + A_{cz} + A_{ru} + A_{sp}$	32.3
$L_{cz} + A_{cz} + A_{ru} + A_{sp} + A_{en}$	29.2

Tabelle 9.3: DMC-Koeffizienten der Kombination des tschechischen Sprachmodells $L_{cz}(k)$, des tschechischen akustischen Modells $A_{cz}(x|k)$, des spanischen akustischen Modells $A_{sp}(x|k)$, des russischen akustischen Modells $A_{ru}(x|k)$ und des englischen akustischen Modells $A_{en}(x|k)$

Modell	Koeffizient
tschechisches Sprachmodell	4
tschechisches akustisches Modell	0.7
spanisches akustisches Modell	0.9
russisches akustisches Modell	0.7
englisches akustisches Modell	9

Tabelle 9.4: DMC Ergebnisse auf VOA, 1000-Best Listen, Nutzung von wissensbasierten Phonemabbildungen, Kombination des tschechischen Sprachmodells $L_{cz}(k)$, der tschechischen, spanischen, russischen und englischen Vokal-, Konsonanten- und Pausenmodelle

	WER in %
Minimale N-Best Fehlerrate	19.8
Maximale N-Best Fehlerrate	56.6
Baseline	33.4
$L_{cz} + A_{cz}$	32.7
$L_{cz} + V_{cz} + C_{cz} + S_{cz}$	32.1
$L_{cz} + A_{cz} + A_{ru} + A_{sp}$	32.3
$L_{cz} + V_{cz} + C_{cz} + S_{cz}$ $+V_{ru} + C_{ru} + S_{ru} + V_{sp} + C_{sp} + S_{sp}$	31.8
$L_{cz} + A_{cz} + A_{ru} + A_{sp} + A_{en}$	29.2
$L_{cz} + V_{cz} + C_{cz} + S_{cz} + V_{ru} + C_{ru}$ $+S_{ru} + V_{sp} + C_{sp} + S_{sp} + V_{en} + C_{en} + S_{en}$	28.9

insgesamt 9 zu kombinierende akustische Teilmodelle. Um eine faire Vergleichsbasis zu schaffen, wurde die Aufteilung in den Vokal-, den Konsonanten- und den Pausenbereich zunächst nur für das tschechische akustische Modell vorgenommen:

$$\begin{aligned} \log p_{\Lambda}(k|x) &= C(\Lambda, x) + \lambda_{LM} L_{cz}(k) \\ &\quad + \lambda_{cz,V} V_{cz}(x|k) + \lambda_{cz,C} C_{cz}(x|k) + \lambda_{cz,S} S_{cz}(x|k). \end{aligned} \tag{9.9}$$

Ensprechend wird das Gesamtsystem mit den Phonemklassenmodellen aus den verwendeten Sprachen konstruiert:

$$\begin{aligned} \log p_{\Lambda}(k|x) &= C(\Lambda, x) + \lambda_{LM} L_{cz}(k) \\ &\quad + \lambda_{cz,V} V_{cz}(x|k) + \lambda_{cz,C} C_{cz}(x|k) + \lambda_{cz,S} S_{cz}(x|k) \\ &\quad + \lambda_{sp,V} V_{sp}(x|k) + \lambda_{sp,C} C_{sp}(x|k) + \lambda_{sp,S} S_{sp}(x|k) \\ &\quad + \lambda_{ru,V} V_{ru}(x|k) + \lambda_{ru,C} C_{ru}(x|k) + \lambda_{ru,S} S_{ru}(x|k) \\ &\quad + \lambda_{en,V} V_{en}(x|k) + \lambda_{en,C} C_{en}(x|k) + \lambda_{en,S} S_{en}(x|k). \end{aligned} \tag{9.10}$$

Tabelle 9.4 faßt schließlich die erhaltenen Erkennungsergebnisse zusammen. Die Ergebnisse in Tabelle 9.4 zeigen, daß die Strukturierung des multilingualen Systems in Phonemklassen zu einer leichten Verbesserung führt. Das multilinguale Phonemklassensystem ist mit 28.9% das beste System, welches im Rahmen des "Towards Lanuage-Independent Speech Recognition"-Projektes konstruiert werden konnte. Während des Workshops wurden folgende weitere Verfahren untersucht [Beyerlein$^+$ 1999b]:

- Manuelle linguistisch orientierte Phonemabbildungen von den Quellsprachen in die Zielsprache führten zu Wortfehlerraten zwischen 60.8% und 88.7%
- Automatische Phonemabbildungen mit Hilfe von einer Stunde Trainingsdaten in der Zielsprache führten zu Wortfehlerraten von 68.3% bis 68.7%.
- Die automatische HMM-Zustands-Abbildung mit Hilfe von einer Stunde Trainingsdaten in der Zielsprache führte zu Wortfehlerraten zwischen 54.5% und 70%.
- Die Adaption der akustischen Modelle der Quellsprachen an eine Stunde Trainingsdaten in der Zielsprache führte zu Wortfehlerraten zwischen 32.7% und 63%.

Das beste zu DMC konkurrierende Einzelverfahren bestand aus der akustischen Adaption eines englischen Triphonsystems an eine Stunde Trainingsdaten in der Zielsprache [Beyerlein$^+$ 1999b]. Es führte zu einer Fehlerrate

von 32.7%. [1] Damit war unter den gleichen Versuchsbedingungen die multilinguale DMC-Kombination von Phonemklassen mit einer Fehlerrate von 28.9% um 11.6% relativ besser, als das beste konkurrierende Einzelverfahren, das während des Workshops getestet wurde.

9.1.4 Zusammenfassung

Aus den Arbeiten an DMC während des 6 wöchigen Johns-Hopkins Sommerworkshops 1999 können folgende Schlußfolgerungen gezogen werden:

- DMC konnte erfolgreich im multilingualen Kontext eingesetzt werden
- Eine multilinguale Systeminterpolation zeigt eine bessere Erkennungsleistung als ein monolinguales System, welches auf einer Stunde Daten trainiert wurde.
- Leistungsschwache Modelle entfernter Sprachen können mit Hilfe von DMC zur Verbesserung der Erkennungsleistung der Modellkombination beitragen.
- Eine log-lineare Strukturierung der akustischen Bewertung des Satzes in Phonemklassen verbessert die Klassifikation.
- Die beste Erkennungsleistung konnte durch die multilinguale DMC-Interpolation von Phonemklassenmodellen erreicht werden.

9.2 Zustandsebene - Log-lineare Hidden-Markoff-Modelle

Wir gehen wie bisher von der Viterbi-Approximation aus, sodaß die Summe über alle Zustandsfolgen, die eine Wortfolge k repräsentieren, durch die optimale Zustandsfolge $s^{(k)}$ ersetzt wird (vgl. Abschnitt 6.2.2). Wir zerlegen nun die Log-Likelihood-Bewertung der Satzhypothese in die Bewertungen der einzelnen HMM-Zustände entlang des Viterbi-Pfades:

$$p(k|x) = \frac{p(k)Q(k,x)\prod_{t=1}^{T} p(\vec{x}_t|s_t^{(k)})}{\sum_{k'} p(k')Q(k',x)\prod_{t=1}^{T} p(\vec{x}_t|s_t^{(k')})}\,, \tag{9.11}$$

wobei

$$Q(k,x) \quad = \quad q(s_0^{(k)})\prod_{t=1}^{T} q(s_t^{(k)}|s_{t-1}^{(k)}) \tag{9.12}$$

das Produkt der Übergangswahrscheinlichkeiten zwischen den HMM-Zuständen beinhaltet.

9.2.1 Log-lineare Kombination der HMM-Zustände

Ein erster Strukturierungsschritt besteht analog zum vorhergehenden Abschnitt darin, den Beitrag der HMM-Zustände zur Gesamtbewertung unterschiedlich zu wichten:

$$p_{\Lambda}(k|x) = \frac{p(k)Q(k,x)p(\vec{x}_t|s_t^{(k)})^{\lambda_{\left[c(s_t^{(k)})\right]}}}{\sum_{k'} p(k')Q(k',x)p(\vec{x}_t|s_t^{(k')})^{\lambda_{\left[c(s_t^{(k')})\right]}}} \tag{9.13}$$

wobei durch $c(s_t^{(k)})$ eine Klasseneinteilung der HMM-Zustände in Z Zustandsklassen gegeben sei ($c \in \{1,\ldots,Z\}$) und $\lambda_{\left[c(s_t^{(k)})\right]}$ der zur Klasse $c = c(s_t^{(k)})$ gehörenden DMC-Koeffizient sei.

[1] Das System stand jedoch erst am Ende des Sommerworkshops zur Verfügung, sodaß es nicht mehr in die DMC-Kombination integriert werden konnte.

Diese Klasseneinteilung kann zum Beispiel so gewählt werden, daß alle Zustände, die zu einem Monophon c gehören, denselben Exponenten λ_c besitzen. Die Koeffizienten λ_c können schließlich mit Hilfe von DMC optimal eingestellt werden. Man beachte, daß der Nenner in Gleichung 9.13 entfällt, da in der vorliegenden Arbeit nur Quotienten der Wahrscheinlichkeiten $p_\Lambda(k|x)$ betrachtet werden. Eine praktische Implementierung der Gleichung 9.13 im logarithmischen Bereich würde im Philips-System die Summe der Abstände zwischen den akustischen Beobachtungen und den Mittelwertvektoren der Mischverteilungen durch eine gewichtete Summe derselben Abstände ersetzen. Die Gewichte werden dabei mit dem DMC-Verfahren bezüglich der Wortfehlerrate des Erkenners optimiert.

9.2.2 Emissionsverteilungen

Wir nehmen nun der Einfachheit halber an, daß die Emissionsverteilung jedes HMM-Zustands durch eine zustandsabhängige Gaußverteilung modelliert wird. Im Philips-System wird zusätzlich zur Viterbi-Approximation der Pfadwahrscheinlichkeit eine Maximum-Approximation der Wahrscheinlichkeitsdichte der Mischverteilung vorgenommen. Dabei wird diejenige Einzelverteilung aus der Mischverteilung ausgewählt, die für den betrachteten Merkmalsvektor die höchste Wahrscheinlichkeitsdichte besitzt. Im Anschluß an diese Auswahl, wird nur noch die einzelne Wahrscheinlichkeitsdichte betrachtet, sodaß die hier vorgenommene formale Vereinfachung keine Einschränkung bezüglich des Philips-Systems darstellt.
Wir verfeinern die Struktur der HMM-Verteilung, indem wir die einzelnen Gaußverteilungen in ihre Bestandteile zerlegen:

$$p_\Lambda(k|x) = \frac{p(k)Q(k,x)\prod_{t=1}^{T} e^{\sum_{i=1}^{D}\sum_{j=1}^{D}\lambda_{\left[c(s_t^{(k)}),i,j\right]}x_t^i x_t^j+\sum_{i=1}\lambda_{\left[c(s_t^{(k)}),i\right]}x_t^i}}{\sum_{k'}p(k')Q(k',x)\prod_{t=1}^{T} e^{\sum_{i=1}^{D}\sum_{j=1}^{D}\lambda_{\left[c(s_t^{(k')}),i,j\right]}x_t^i x_t^j+\sum_{i=1}\lambda_{\left[c(s_t^{(k')}),i\right]}x_t^i}} \tag{9.14}$$

Um die Schreibweise in diesem und in den folgenden Abschnitten kompakter zu gestalten, führen wir zunächst die folgenden **Zustandsindikatorsummen** ein:

$$\sum_{(c,t)|k} f(c,t) := \sum_{t=1}^{T}\sum_{c=1}^{Z}\delta(s_t^{(k)},c)f(c,t), \tag{9.15}$$

$$\sum_{(c,c',t)|k} f(c,c',t) := \sum_{t=1}^{T}\sum_{c=1}^{Z}\sum_{c'=1}^{Z}\delta(s_{t-1}^{(k)},c)\delta(s_t^{(k)},c')f(c,c',t). \tag{9.16}$$

Die erste Zustandsindikatorsumme beinhaltet die Summation über alle Zeitpunkte t und alle Zustandsklassen c, wobei der Zustand $s_t^{(k)}$ zur Zustandsklasse c gehören muß. Diese Summe wird für die kompakte Darstellung des log-linearen Emissionsverteilungsmodells genutzt. Bei der zweiten Zustandsindikatorsumme wird über alle Zustandspaare summiert, die eine Nachbarschaftsrelation auf dem optimalen Pfad besitzen. Sie wird im folgenden Abschnitt für die kompakte Repräsentation des log-linearen Zustandsübergangsmodells eingesetzt.
Die Struktur der log-linearen Verteilung kann damit wie folgt vereinfacht werden:

$$p_\Lambda(k|x) = \frac{p(k)Q(k,x)e^{\sum_{(c,t)|k}\sum_{i=1}^{D}\sum_{j=1}^{D}\lambda_{[c,i,j]}x_t^i x_t^j+\sum_{(c,t)|k}\sum_{i=1}^{D}\lambda_{[c,i]}x_t^i}}{\sum_{k'}p(k')Q(k',x)e^{\sum_{(c,t)|k'}\sum_{i=1}^{D}\sum_{j=1}^{D}\lambda_{[c,i,j]}x_t^i x_t^j+\sum_{(c,t)|k'}\sum_{i=1}^{D}\lambda_{[c,i]}x_t^i}} \tag{9.17}$$

Die freien Parameter $\lambda_{[c,i,j]}$ und $\lambda_{[c,i]}$ können nun mit Hilfe von DMC bezüglich der Wortfehlerrate des Spracherkennungssystems optimiert werden. Damit haben wir ein Schema zur Bestimmung der Parameter von Emissionsverteilungen der HMM-Zustände konstruiert. Es ist zu beachten, daß die aus den Koeffizienten $\lambda_{[c,i,j]}$ und $\lambda_{[c,i]}$

abgeleitete quadratische Form nicht notwendigerweise positiv definit ist. In diesem Falle kann man bei der erhaltenen Funktion nicht mehr von einer Gaußverteilung im klassischen Sinne sprechen[2]. Bei der Implementierung dieser Verteilungsstruktur ist zu beachten, daß nicht alle HMM-Zustandsklassen c in einer Satzhypothese enthalten sein müssen. In diesem Falle wird die zum Parameter $\lambda_{[c,i,j]}$ bzw. $\lambda_{[c,i]}$ zugehörige Bewertung des Basismodells auf den Wert 0 gesetzt. Damit kann eine log-lineare Kombination aller Elemente der Modellkombination für jede Satzhypothese durchgefürt werden, unabhängig davon, welche Teilmenge von Basismodellen tatsächlich für die Bewertung der Satzhypothese benötigt wird.

9.2.3 Übergangswahrscheinlichkeiten

DMC kann auch zur Optimierung der Übergangswahrscheinlichkeiten des Hidden-Markoff-Modells genutzt werden. Dabei ergibt sich eine interessante Übereinstimmung mit experimentellen Beobachtungen zur Einstellung der sogenannten Zeitverzerrungsstrafen des HMM's:
Wir gehen wie bisher von der Viterbi-Approximation aus, sodaß die Summe über alle Zustandsfolgen, die eine Wortfolge k repräsentieren, durch die optimale Zustandsfolge $s^{(k)}$ ersetzt wird (vgl. Abschnitt 6.2.2). Das Produkt aus den Übergangswahrscheinlichkeiten $Q(k,x)$, den Emissionsverteilungsdichten und der Sprachmodellwahrscheinlichkeit wird nun in eine log-lineare Kombination umgewandelt:

$$p(k|x) = \frac{p(k)Q(k,x)^{\lambda_Q}\left(\prod_{t=1}^{T} p(\vec{x}_t|s_t^{(k)})\right)^{\lambda_P}}{\sum_{k'} p(k')Q(k',x)^{\lambda_Q}\left(\prod_{t=1}^{T} p(\vec{x}_t|s_t^{(k')})\right)^{\lambda_P}}, \tag{9.18}$$

Aus dem Verhältnis von λ_P und λ_Q ergibt sich ein Faktor, mit dem die logarithmierten Übergangswahrscheinlichkeiten skaliert werden. Die skalierten logarithmierten Übergangswahrscheinlichkeiten nennt man Zeitverzerrungsstrafen. In Experimenten im WSJ0-System hat sich nach der manuellen Optimierung der Zeitverzerrungsstrafen (Abschnitt 6.2.2) gezeigt, daß dieser Faktor analog zum Sprachmodellfaktor signifikant vom Wert 1 abweicht. Das gleiche gilt auch für das HUB4-System. Damit wird auch hier die Anwendung von DMC motiviert.
Schließlich kann man das gesamte Hidden-Markoff-Modell in ein **log-lineares Hidden-Markoff-Modell** umwandeln:

$$p_{\Lambda}(k|x) = \frac{h(k,x)}{\sum_{k'} h(k',x)}, \tag{9.19}$$

mit

$$h(k,x) = p(k)e^{\lambda^0_{[c(s_0^{(k)})]}\log q_0(c(s_0^{(k)}))+\sum_{(c,c',t)|k}\lambda^0_{[c,c']}\log q(c,c')+\sum_{(c,t)|k}\sum_{i=1}^{D}\lambda^1_{[c,i]}x_t^i+\sum_{(c,t)|k}\sum_{i=1}^{D}\sum_{j=1}^{D}\lambda^2_{[c,i,j]}x_t^i x_t^j} \tag{9.20}$$

In diesem Modell sind alle Parameter $\lambda^1_{[.]}$, $\lambda^2_{[.]}$ der Emissionsverteilungen wie auch alle Zeitverzerrungsparameter $\lambda^0_{[.]}$ frei wählbar und können mit Hilfe von DMC bestimmt werden. Insbesondere bietet sich bei genügend vielen Trainingsdaten auch die geschlossene Matrixgleichung (4.30) zur Bestimmung aller Parameter des log-linearen Hidden-Markoff-Modells an.

9.2.4 Log-lineare Mischverteilungen

Bisher wird die Emissionsverteilung einer HMM-Zustandsklasse c im wesentlichen durch Linearkombinationen von Gaußverteilungen $v = 1, \ldots, V_c$ modelliert:

$$p(\vec{x}_t|c) = \sum_{v=1}^{V_c} \omega_{[c,v]} \frac{1}{(2\pi)^{D/2}|\Sigma_{[c,v]}|^{1/2}} e^{-1/2\cdot(\vec{x}-\vec{\mu}_{[c,v]})^T\Sigma^{-1}_{[c,v]}(\vec{x}-\vec{\mu}_{[c,v]})},$$

[2]Um den genannten Fall von vornherein auszuschließen, kann bereits während der Optimierung die Bedingung gestellt werden, daß quadratische Form positiv definit ist.

$$\sum_{c,v} \omega_{[c,v]} \quad = \quad 1 \tag{9.21}$$

Eine andere denkbare Mischverteilungsform ist die log-lineare Kombination von Gaußverteilungen:

$$p(\vec{x}_t|c) \quad = \quad \frac{\prod_{v=1}^{V_c} \left(e^{-1/2 \cdot (\vec{x}-\vec{\mu}_{[c,v]})^T \Sigma_{[c,v]}^{-1} (\vec{x}-\vec{\mu}_{[c,v]})} \right)^{\lambda_{[c,v]}}}{\int_{\vec{y}} \prod_{v=1}^{V} \left(e^{-1/2 \cdot (\vec{y}-\vec{\mu}_{[c,v]})^T \Sigma_{[c,v]}^{-1} (\vec{y}-\vec{\mu}_{[c,v]})} \right)^{\lambda_{[c,v]}} d\vec{y}} \tag{9.22}$$

Die Gewichte der log-linearen Mischverteilung können auch hier mit den in Kapitel 4 dargestellten Verfahren diskriminativ trainiert werden. Man beachte, daß bei positiven Gewichten $\lambda_1, \ldots, \lambda_V > 0$ die resultierende Verteilung selbst wieder zu einer Gaußverteilung wird. Insbesondere für die Adaption vortrainierter Verteilungen kann diese Verteilungsform von Vorteil sein. Die log-lineare Mischverteilung kann in natürlicher Weise in das log-lineare Hidden-Markoff-Modell integriert werden. Die resultierende Verteilung lautet:

$$p_{\Lambda}(k|x) \quad = \quad \frac{h(k,x)}{\sum_{k'} h(k',x)}\,, \tag{9.23}$$

mit

$$h(k,x) = p(k) e^{\lambda^0_{\left[c(s_0^{(k)})\right]} \log q_0(c(s_0^{(k)})) + \sum_{(c,c',t)|k} \lambda^0_{[c,c']} \log q(c,c') + \sum_{(c,t)|k} \sum_{v=1}^{V} \lambda^3_{[c,v]} (\vec{x}_t - \vec{\mu}_{[c,v]})^T {\Sigma_{[c,v]}}^{-1} (\vec{x}_t - \vec{\mu}_{[c,v]})} \tag{9.24}$$

Der komplizierte Nenner braucht nicht weiter beachtet zu werden, da in der vorliegenden Arbeit lediglich die Quotienten der Wahrscheinlichkeiten $p_{\Lambda}(k|x)$ betrachtet werden. Durch die log-linearen Kombination von Gaußverteilungen entsteht wieder eine Gaußverteilung. Damit kann ein zweistufiges Verfahren zur robusten Schätzung klassenspezifischer Gaußverteilungen konstruiert werden:

- Zunächst wird eine Menge klassenunabhängiger Gaußverteilungen trainiert, die den akustischen Raum möglichst optimal abdecken.
- Anschließend werden mit Hilfe von DMC aus den klassenunabhängigen Gaußverteilungen klassenspezifische Gaußverteilungen gebildet. Dabei wird das Gewicht, mit dem eine klassenunabhängige Gaußverteilung in die klassenspezifische Gaußverteilung eingeht, bezüglich der Wortfehlerrate des Erkenners optimiert.

Der Vorteil liegt hier in der optimalen Ausnutzung von Trainingsdaten und in der Beschleunigung des Erkenners durch die Einsparung von Log-Likelihood-Berechnungen.

9.3 Klassenspezifische Diskriminative Modellkombination

In diesem Abschnitt verallgemeinern wir die Vorgehensweise aus den Abschnitten 9.1 und 9.2. Gegeben seien Modelle $g_m(x_b, k_b)$, die einen Bereich b (z.B. Wort, Phonem, Zustand) aus der Satzhypothese k bezüglich eines Merkmales m (z.B. akustische Vektorkomponente) bewerten. Die Bewertungen sollen in log-linearer Form kombiniert werden. Der Gewichtsfaktor für die Bewertung des Modells $g_m(x_b, k_b)$ wird mit $\lambda_{[m,b]}$ bezeichnet. Die log-lineare Modellkombination lautet:

$$p_{\Lambda}(k|x) \quad = \quad \frac{\prod_{b \in k} \prod_{m=1}^{M} g_m(x_b, k_b)^{\lambda_{[m,b]}}}{\sum_{k'} \prod_{b \in k'} \prod_{m=1}^{M} g_m(x_b, k'_b)^{\lambda_{[m,b]}}} \tag{9.25}$$

Auch in dieser allgemeinen Formulierung muß der komplizierte Term im Nenner nicht weiter betrachtet werden, da das DMC-Verfahren auf der Quotientenbildung der Wahrscheinlichkeiten $p_{\Lambda}(k|x)$ beruht und somit der Nenner

entfällt. Die Bereiche - Wort, Phonem, Zustand - werden im Kontext der automatischen Spracherkennung selbst wieder als zu unterscheidende Klassen aufgefaßt. Aus diesem Grunde wird der beschriebene Ansatz als **Klassenspezifische Diskriminative Modellkombination** bezeichnet. Im engeren Sinne ist diese Begriffswahl auch deswegen sinnvoll, da mit diesem Ansatz auch die sogenannten klassenspezifischen Verteilungsmodelle (vgl. Abschnitt 9.2) trainiert werden können. Es ist leicht zu sehen, wie m, b zu wählen sind, um die in den Abschnitten 9.1 und 9.2 beschriebenen log-linearen Verteilungen zu erhalten.

Die in Gleichung (9.25) vorgenommene log-lineare Strukturierung der Wahrscheinlichkeitsbewertung eines Satzes in verschiedene Bereiche wurde bereits in [Vergyri 1997] vorgeschlagen. Dort wird der Satz in Worthypothesenklassen aufgeteilt. Ziel der Dekomposition in [Vergyri 1997] ist es, in Abhängigkeit von der Worthypothesenklasse das Sprachmodell, das akustische Modell und die Wortstrafe unterschiedlich zu gewichten (vgl. Abschnitt 5.2.3).

9.4 Optimierung von Sprachmodellen

Ein Spezialfall der diskriminativen Modellkombination ist die diskriminative Optimierung von log-linearen Sprachmodellkombinationen. Dazu faßt man wie bisher die Klasse k als Wortfolge auf, die Beobachtung x wird jedoch in den Sprachmodellverteilungen ignoriert. Die zu optimierende Verteilung lautet dann:

$$p_\Lambda(k) \quad = \quad \frac{\exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k)\right\}}{\sum_{k'} \exp\left\{\sum_{j=1}^{M} \lambda_j \log p_j(k')\right\}} \,. \tag{9.26}$$

Die Gewichte λ_j können mit Hilfe von DMC optimiert werden. In [Klakow 1998b] wird eine log-lineare Kombination von Sprachmodellen bezüglich des Maximum-Likelihood-Kriteriums optimiert.

9.5 Zusammenfassung

Tabelle 9.5 gibt einen Überblick über mögliche Anwendungen von DMC. Dabei wird deutlich, daß DMC unabhängig von der Art der verwendeten Modelle und der Hierarchieebene formuliert werden kann. Seine Leistungsfähigkeit wurde in Experimenten zur 'Systemoptimierung' nachgewiesen. Die zukünftige Forschungsarbeit wird zeigen, ob DMC auch in der experimentellen Anwendung so allgemein verwendbar ist, wie es zunächst durch seine theoretische Formulierung erscheint.

Tabelle 9.5: Übersicht über getestete und theoretisch sinnvolle Anwendungsmöglichkeiten von DMC

Aufgabe	DMC - Definition		
Systemoptimierung (Kapitel 4-7)	Klasse k	-	Wortfolge
	Beobachtung x	-	Merkmalsvektorfolge
	Basismodelle	-	akustische und Sprachmodelle
Emissionsverteilungen (Abschnitt 9.2.2)	Klasse k	-	HMM Zustand
	Beobachtung x	-	Merkmalsvektor
	Basismodelle	-	Verteilungen der Exponentialfamilie
Log-lineares HMM (Abschnitt 9.2)	Klasse k	-	HMM Zustandsfolge
	Beobachtung x	-	Merkmalsvektorfolge
	Basismodelle	-	Übergangsverteilungsmodell, Emissionsverteilungsmodell
Sprachmodellinterpolation (Abschnitt 9.4)	Klasse k	-	Wortfolge
	Beobachtung x	-	wird ignoriert
	Basismodelle	-	Sprachmodelle
Sprachübersetzung	Klasse k	-	englische Wortfolge
	Beobachtung x	-	deutsche Äußerung
	Basismodelle	-	akustische, Sprach- und Übersetzungsmodelle
Multilingualität (Abschnitte 9.1,9.1.3)	Klasse k	-	tschechische Phonemfolge
	Beobachtung x	-	tschechische Äußerung
	Basismodelle	-	englische, spanische, chinesische, russische Phonemklassenmodelle

Kapitel 10

Beiträge zur Wissenschaft

In dieser Arbeit ist ein Verfahren zur Kombination vortrainierter Teilmodelle von Spracherkennungssystemen entwickelt worden. Das Verfahren basiert auf dem diskriminativen Training der freien Parameter einer Verteilung, die der Maximum-Entropie-Familie angehört. Das Verfahren ist modellunabhängig und erlaubt die automatische Kombination beliebiger und beliebig vieler Modelle. Als Optimierungskriterium wurde die geglättete empirische Wortfehlerrate eingeführt. Mit Hilfe des neuen Verfahrens konnte die Baseline-Fehlerrate des Hochleistungsspracherkenners der Philips Forschungslaboratorien Aachen signifikant verbessert werden:

- auf dem männlichen Teil der WSJ0-Datenbasis von 3.5% auf 3.2% um 8% relativ,
- auf der HUB4'97-F0-Datenbasis (Reine Nachrichtensprache) von 13.0% auf 11.3% um 13% relativ und
- auf der gesamten HUB4'97-Testdatenbasis von 20.7% auf 18.8% um 9% relativ.

Dabei ist zu beachten, daß es auf den aus den aktuellen Massenmedien stammenden HUB4-Daten generell und insbesondere mit den bekannten Standardmethoden schwierig ist, den Spracherkenner zu verbessern.[1]
Die Verbesserungen sind auf folgende neue Erkenntnisse zurückzuführen, die im Rahmen dieser Arbeit erlangt wurden:

- Mit DMC läßt sich der optimale Sprachmodellfaktor automatisch bestimmen.
- DMC ist für die Glättung der akustischen Modelle geeignet. Mit Hilfe der Interpolation von MLLR-adaptierten wortinternen Triphonmodellen, wortübergreifenden Triphonmodellen und Pentaphonmodellen konnte die Baseline Fehlerrate um 6% relativ gegenüber dem besten MLLR-adaptierten wortübergreifenden Triphonsystem verbessert werden. Dabei konnte in der Erkennung auf das $\mathcal{N}$-Best-Verfahren verzichtet werden. Tests mit einem ausschließlich auf Pentaphonmodellen beruhenden System schlugen fehl, da die stark spezialisierten Pentaphonmodelle nicht ausreichend gut trainiert werden konnten.
- Die hinzugefügte log-lineare Kombination von Lückensprachmodellen führt zu einer Reduktion der Fehlerrate um 3% relativ.
- Für die Gewichtung der in den HUB4-Experimenten verwendeten Basismodelle wurden insgesamt 10 zusätzliche freie Parameter eingeführt. Diese wenigen Parameter beeinflussen durch die log-lineare Kombinationsform die Wirkung aller 140 Millionen Basismodellparameter auf die Entscheidungsregel. Damit konnte trotz der geringen Anzahl freier Parameter ein signifikanter Gewinn der Fehlerrate erreicht werden.
- Die log-lineare Kombination lieferte in jedem Experiment bessere Ergebnisse als die lineare Kombination bzw. ein einfaches Votingverfahren.

Die im Rahmen dieser Arbeit erzielten theoretische Ergebnisse können wie folgt zusammengefaßt werden:

[1] Mit der MLLR-Adaption erreicht man auf dieser Datenbasis eine Verbesserung um 6% relativ. Durch den Einsatz von wortübergreifenden Triphonen wird das System 5% relativ besser. Der Übergang vom Bigrammsprachmodell auf das Trigrammsprachmodell verringert die Fehlerrate um 6% relativ.

- Durch die diskriminative Optimierung einer Verteilung der Maximum-Entropie-Familie konnte ein Verfahren zur Kombination beliebiger Verteilungsmodelle in eine Gesamtverteilung abgeleitet werden. Dieses Verfahren minimiert die geglättete empirische Wortfehlerrate der Entscheidungsregel.
- Das Kriterium zur Minimierung der geglätteten empirischen Wortfehlerrate konnte so modifiziert werden, daß eine geschlossene Lösung für die Bestimmung der Parameter der Modellkombination gefunden wurde.
- Die Unabhängigkeit des Verfahrens von der Hierarchieebene der Klassifikationsaufgabe und von den verwendeten Modellen wurde beispielhaft demonstriert, indem Verfahren
  - für die Balancierung des Sprachmodellfaktors,
  - für die log-lineare Kombination von beliebigen akustischen Modellen und Sprachmodellen,
  - für die log-lineare Kombination von Phonemmodellen oder Phonemklassenmodellen
  - für die Balancierung des Einflusses der Übergangsverteilung und der Emissionsverteilung innerhalb der HMM's,
  - für die Schätzung von Parametern der Gaußverteilungen der HMM-Zustände und
  - für die optimale log-lineare Sprachmodellinterpolation

  diskutiert wurden.

Im Verlaufe einer zusätzlichen Studie während des 6 wöchigen Sommerworkshops am 'Center for Speech and Language Processing' der Johns-Hopkins-Universität in Baltimore im Juli und August 1999 konnte die Wortfehlerrate eines tschechischen Broadcast-News-Systems von 33.4% auf 28.9% gesenkt werden. DMC wurde hier für die Kombination von multilingualen Phonemklassenmodellen der Sprachen Tschechisch, Russisch, Spanisch und Englisch eingesetzt.

Literaturverzeichnis

[Anderson+ 1962] T.W. Anderson and R. R. Buhadur, “Classification into Two Multivariate Normal Distributions with Different Covariance Matrices”, Ann. Math. Stat., 1962, Vol. 33, 422-431

[Aubert+ 1994] X. Aubert, C. Dugast, H. Ney, V. Steinbiß, “Large Vocabulary Continuous Speech Recognition of Wall Street Journal Data”, Proceedings ICASSP Adelaide, 1994, Vol. II, 129-132.

[Aubert+ 1995] X. Aubert, H. Ney, “Large Vocabulary Continuous Speech Recognition Using Word Graphs”, Proceedings ICASSP Detroit, Michigan, U.S.A., 1995, Vol. I, 49-52

[Aubert+ 1996] X. Aubert, P. Beyerlein, M. Ullrich “A Bottom-Up Approach for Handling Unseen Triphones in Large-Vocabulary Continuous-Speech Recognition”, Proceedings ICSLP Philadelphia, Pennsylvania, U.S.A., 1996, Vol. I, 14-17

[Bahl+ 1980] L.R. Bahl, R. Bakis, F. Jelinek, R. Mercer, “Language Model/Acoustic Channel Balance Mechanism”, IBM Technical Disclosure Bulletin, 1980, Vol. 23, 3464-3465

[Bahl+ 1991] L.R. Bahl, V.P. de Souza, P.S. Gopalakrishnan, D. Nahamoo, M.A. Picheny, “Decision Trees for Phonological Rules in Continuous Speech”, Proceedings ICASSP Toronto, Canada, 1991, Vol. I, 185-188

[Bahl+ 1993] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, M.A. Picheny, “Context Dependent Vector Quantization for Continuous Speech Recognition”, Proceedings ICASSP Minneapolis, Minnesota, U.S.A., 1993, Vol. II, 632-635

[Besling 1994] S. Besling, “A Statistical System for Grapheme–to–Phoneme Conversion”, Proceedings Tenth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research: Reflections on the Future of Text, 5, 1994.

[Beulen 1999a] K. Beulen, “Phonetische Entscheidungsbäume für die automatische Spracherkennung mit großem Vokabular”, eingereichte Dissertation am Lehrstuhl für Informatik VI, Rheinisch-Westfälische Technische Hochschule Aachen, Februar 1999

[Beulen 1999b] K. Beulen, private communication, Lehrstuhl für Informatik VI, Rheinisch-Westfälische Technische Hochschule Aachen, March 1999

[Beyerlein 1994] P. Beyerlein, “Fast Log-Likelihood Computation for Mixture Densities in a High-Dimensional Feature Space”, Proceedings ICSLP Yokohama, Japan, 1994, 271-274.

[Beyerlein 1995] P. Beyerlein, M. Ullrich, “Hamming Distance Approximation For A Fast Log-Likelihood Computation for Mixture Densities”, Proceedings EUROSPEECH 1995, Madrid, Spain, 1995, 1083-1086

[Beyerlein+ 1997a] P. Beyerlein, M. Ullrich, P. Wilcox, “Modelling and Decoding of Crossword Context Dependent Phones in the Philips Large Vocabulary Continuous Speech Recognition System“, Proceedings EUROSPEECH Rhodes, Greece, 1997, Vol. 3, 1163-1166

[Beyerlein 1997b] P. Beyerlein, “Discriminative Model Combination, Theory and Application to Speech Recognition”, Philips Research Report No. 1276/97, Philips Research Laboratories Aachen, 1997

[Beyerlein 1997c] P. Beyerlein, “Discriminative Model Combination”, Proceedings 1997 IEEE ASRU Workshop, Santa Barbara, California, U.S.A., 1997, 238-245

[Beyerlein$^+$ 1998a] P. Beyerlein, X. Aubert, R. Häb-Umbach, D. Klakow, M. Ullrich, A. Wendemuth, P. Wilcox, “Automatic Transcription of Englich Broadcast News”, Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 85-90

[Beyerlein 1998b] P. Beyerlein, “Discriminative Model Combination”, Proceedings ICASSP Seattle, Washington, U.S.A., 1998, 481-484

[Beyerlein$^+$ 1999a] P. Beyerlein, X. Aubert, R. Häb-Umbach, M. Harris, D. Klakow, S. Molau, M. Pitz, A. Wendemuth, “The Philips/RWTH System for Transcription of Broadcast News”, DARPA 1999 Broadcast News Workshop, Herndon, Virginia, U.S.A., 1999

[Beyerlein$^+$ 1999b] P. Beyerlein, B. Byrne, J. M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Pterek, J. Picone, W. Wang, “Towards Language Independent Acoustic Modeling”, to appear in Proceedings 1999 IEEE ASRU Workshop, Keystone, Colorado, U.S.A., 1999

[Bronstein$^+$ 1957] I. N. Bronstein, K. A. Semendjajew, “Taschenbuch der Mathematik”, Gemeinschaftsausgabe Verlag Nauka, Moskau und Teubner Verlagsgesellschaft Leipzig, 22. Auflage, 1985

[Brown 1987] P.F. Brown, “The Acoustic-Modeling Problem in Automatic Speech Recognition”, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, U.S.A., 1987

[Chen$^+$ 1998] S. Chen, M.J.F. Gales, P.S. Gopalakrishnan, R.A. Gopinath, D. Kanevsky, P. Olsen, L. Polymenakos. “IBM’s LVCSR System for Transcription of Broadcast News Used in the 1997 HUB4 English Evaluation”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 69-74

[Cover$^+$ 1991] T.M. Cover, J.A. Thomas, “Information Theory”, John Wiley, New York, 1991

[Darroch$^+$ 1972] J.N. Darroch, D. Ratcliff, “Generalized Iterative Scaling for Log-Linear Models”, Ann. Math. Stat., 1972, Vol. 43, 1470-1480

[Digalakakis$^+$ 1995] V. Digalakakis, M. Weintraub, A. Sankar, H. Franco, L. Neumeyer, and H. Murveit, “Continuous Speech Dictation on ARPA’s North American Business News Domain”, Proceedings of the Spoken Language Systems Technology Workshop, Austin, Texas, U.S.A., 1995, 88-93

[Duda$^+$ 1973] R.O. Duda, P.E. Hart, “Pattern Classification and Scene Analysis”, John Wiley & Sons, Inc., New York, 1973

[Dugast$^+$ 1995a] C. Dugast, R. Kneser, X. Aubert, S. Ortmanns, K. Beulen, and H. Ney, “Continuous Speech Recognition Tests and Results for the NAB’94 Corpus”, Proceedings of the Spoken Language Systems Technology Workshop, Austin, Texas, U.S.A., 1995, 156-161

[Dugast$^+$ 1995b] C. Dugast, P. Beyerlein, R. Häb–Umbach, “Application of Clustering Techniques to Mixture Density Modelling for Continuous-Speech Recognition”, Proceedings ICASSP Detroit, Michigan, U.S.A., 1995, Vol. I, 524-527

[Fiscus 1997] J. G. Fiscus, “A Post-Processing System To Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)”, Proceedings 1997 IEEE ASRU Workshop, Santa Barbara, California, U.S.A., 1997, 347-354

[Fisz 1989] M. Fisz, “Wahrscheinlichkeitsrechnung und Mathematische Statistik”, VEB Deutscher Verlag der Wissenschaften, Berlin, 1989

[Fukunaga 1990] K. Fukunaga, “Introduction to Statistical Pattern Recognition”, Second Edition, Academic Press, New York, 1990

[Gauvain 1997] J.L. Gauvain, G. Adda. L. Lamel, and M. Adda-Decker, “Transcribing Broadcast News: The LIMSI Nov96 Hub4 System”, Proceedings of the DARPA Speech Recognition Workshop, Chantilly, Virginia, U.S.A., 1997, 56-60.

[Gauvain$^+$ 1998] J. L. Gauvain, L. Lamel, G. Adda. “The LIMSI 1997 HUB4-E Transcription System”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 69-74

[Graff 1997] D. Graff, “The 1996 Broadcast News Speech and Language Model Corpus”, Proceedings of the DARPA Speech Recognition Workshop, Chantilly, Virginia, U.S.A., 1997, 11-14

[Häb$^+$ 1992] R. Häb-Umbach, H. Ney, “Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition”, Proceedings ICASSP San Francisco, California, U.S.A., 1992, Vol. I, 13-16

[Häb$^+$ 1993] R. Häb-Umbach, D. Geller, H. Ney, “Improvements in Connected Digit Recognition Using Linear Discriminant Analysis and Mixture Densities”, Proceedings ICASSP Minneapolis, Minnesota, U.S.A., 1993, 239-242

[Häb$^+$ 1998] R. Häb-Umbach, X. Aubert, P. Beyerlein, D. Klakow, M. Ullrich, A. Wendemuth, P. Wilcox, “Acoustic Modeling in the Philips HUB4-System”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 116-119

[Häb 1999] R. Häb-Umbach, private communication, Philips GmbH Forschungslaboratorien Aachen, March 1999

[Huang$^+$ 1993] X. Huang, M. Belin, F. Alleva and M. Hwang, “Unified Stochastic Engine (USE) for Speech Recognition”, Proceedings ICASSP Minneapolis, Minnesota, U.S.A., 1993, 636-639

[Jaynes 1957] E.T. Jaynes, “Information Theory and Statistical Mechanics”, Physical Reviews, 1957, Vol. 106, 620-630

[Jelinek 1976] F. Jelinek, “Continuous Speech Recognition by Statistical Methods”, Proceedings of the IEEE, April 1976, Vol. 64, No. 10, 532-556

[Jelinek 1995] F. Jelinek, “Two New Approaches To Language Modeling: A Tutorial”, in Speech Recognition and Coding - New Advances and Trends, Springer-Verlag, Berlin-Heidelberg, 1995, 226-239

[Jelinek 1997] F. Jelinek, “Language Modeling for Speech Recognition”, Spoken Queries in European Languages Workshop on Multi-Lingual Information Retrieval Dialogs, Pilsen, Czech Republic, April 1997

[Juang$^+$ 1992] B.-H. Juang, S. Katagiri, “Discriminative Learning for Minimum Error Classification”, IEEE Trans. on Signal Processing, December 1992, SP-40, No. 12, 3043-3054

[Juang$^+$ 1995] B. H. Juang, W. Chou, C.H. Lee, “Statistical and Discriminative Methods for Speech Recognition”, in Speech Recognition and Coding - New Advances and Trends, Springer-Verlag, Berlin-Heidelberg, 1995, 41-55

[Kapur$^+$ 1992] J.N. Kapur, H.K. Kesavan, “Entropy Optimization Principles with Applications”, Academic Press, New York, 1992

[Klakow$^+$ 1998a] D. Klakow, X. Aubert, P. Beyerlein, R. Häb-Umbach, M. Ullrich, A. Wendemuth, P. Wilcox, “Language Model Investigations Related To Broadcast News”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 265-269

[Klakow 1998b] D. Klakow, “Log-Linear Interpolation of Language Models”, Proceedings ICSLP Sidney, Australia, 1998, 1695-1698

[Kneser$^+$ 1995] R. Kneser, H. Ney, “Improved backing–off for m-gram language modeling”, Proceedings ICASSP Detroit, 1995, Vol. I, 181-184

[Kneser 1997] R. Kneser, J. Peters, and D. Klakow. “Language model adaptation using dynamic marginals”, Proceedings EUROSPEECH Rhodes, Greece, 1997, 1971-1974

[Kubala+ 1997] F. Kubala, H. Jin, S. Matsoukas, L. Nguyen, R. Schwartz, J. Makhoul, “Advances in Transcription of Broadcast News”, Proceedings EUROSPEECH Rhodes, 1997, 927-930

[Kubala+ 1998] F. Kubala, J. Davenport, H. Jin, D. Liu, T. Leek, S. Matsoukas, D. Miller, L. Nguyen, F. Richardson, R. Schwartz, J. Makhoul. “The 1997 BBN BYBLOS System Applied to Broadcast News Transcription”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 35-40

[Lee 1989] K.-F. Lee “The Development of the SPHINX System”, Kluwer Academic Publishers, Boston, 1989

[Lee+ 1996] L. Lee, R. Rose, “Speaker normalization using efficient frequency warping procedures,” Proceedings ICASSP Atlanta, Georgia, U.S.A., 1996, Vol. I, 353-356

[Legetter 1994] C. J. Legetter, “Speaker Adaptation of HMM’s Using Linear Regression”, Cambridge University, Report CUED/F-INFEND/TR. 181, June 1994

[McLachlan+ 1997] G. J. McLachlan, T. Krishnan, “The EM Algorithm and Extensions”, John Wiley & Sons, Inc., New York, 1997.

[Müller+ 1986] P. H. Müller, V. Nollau, A. I. Polovinkin, “Stochastische Suchverfahren”, Fachbuchverlag Leipzig, 1986

[Ney 1990] H. Ney, “Acoustic Modelling of Phoneme Units for Continuous Speech Recognition”, Signal Processing V: Theory and Applications, 1990, 65-72

[Ney+ 1992] H. Ney, D. Mergel, A. Noll, A. Päseler, “Data Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition”, IEEE Trans. on Signal Processing, Feb. 1992, Vol. SP-40 , No. 2, 272-281

[Ney+ 1994] H. Ney, V. Steinbiss, R. Häb-Umbach, B.-H. Tran, U. Essen, “An Overview of the Philips Research System for Large-Vocabulary Continuous-Speech Recognition”, International Journal of Pattern Recognition and Artificial Intelligence, Jan. 1994, Vol. 8, No. 1, 1-38

[Ney 1995] H. Ney, “On The Probabilistic Interpretation of Neural Network Classifiers And Discriminative Training Criteria”, IEEE Trans. On Pattern Analysis and Machine Intelligence, Feb. 1995, Vol. 17, No. 2, 107-119

[Ney+ 1996] Ney H., Aubert X., “Dynamic Programming Search Strategies: From Digit Strings To Large Vocabulary Word Graphs”, in “Automatic Speech and Speaker Recognition”, Kluwer Academic Publishers, New York, 1996, 385-411

[Ney+ 1997] H. Ney, F. Wessel, S. C. Martin, “Statistical Language Modeling Using Leaving-One-Out”, in “Corpus-Based Methods in Speech and Language”, Kluwer Academic Publishers, New York, 1997, 174-207

[Normandin 1995] Y. Normandin, “Optimal Splitting of HMM Gaussian Mixture Components With MMIE Training”, Proceedings ICASSP Detroit, Michigan, U.S.A., 1995, Vol. I, 449-452

[Odell 1995] J. J. Odell: “The Use of Context in Large Vocabulary Speech Recognition”, Ph.D. thesis, University of Cambridge, England, 1995

[Ortmanns 1998] S. Ortmanns, “Effiziente Suchverfahren zur Erkennung kontinuierlich gesprochener Sprache”, Dissertation am Lehrstuhl für Informatik VI, Rheinisch-Westfälische Technische Hochschule Aachen, Juni 1998

[Paul+ 1992] D. Paul, J. Baker, “The Design for the Wall Street Journal-based CSR Corpus”, Proceedings of the DARPA Speech and Natural Language Workshop, Harriman, New York, U.S.A., 1992, 357-361

[Rabiner+ 1993] L. Rabiner, B.-H. Juang, “Fundamentals of Speech Recognition”, Prentice Hall, New Jersey, 1993

[Sankar+ 1998] A. Sankar, F. Weng, Z. Rivlin, A. Stolcke, R. R. Gadde, “Development of SRI’s 1997 Broadcast News Transcription System”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 91-96

[Schwartz 1997] R. Schwartz, H. Jin, F. Kubala, and S. Matsoukas, “Modeling those F-Conditions - Or Not”, Proceedings of the DARPA Speech Recognition Workshop, Chantilly, Virginia, U.S.A., 1997, 115-118

[Seymore+ 1998] K. Seymore, S. Chen, S. Doh, M. Eskenazi, E. Gouvea, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Stern and E. Thayer, “The 1997 CMU Sphinx-3 English Broadcast News Transcription System”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 55-59

[Siegler 1997] M. Siegler, U. Jain, B. Raj, R.M. Stern, “Automatic Segmentation and Clustering of Broadcast News Audio”, Proceedings of the DARPA Speech Recognition Workshop, Chantilly, Virginia, U.S.A., 1997, 97-99

[Steinbiss+ 1993] V. Steinbiss, H. Ney, R. Häb-Umbach, B.-H. Tran, U. Essen, R. Kneser, M. Oerder, H.-G. Meier, X. Aubert, C. Dugast, D. Geller, “The Philips Research System for Large Vocabulary Continuous-Speech Recognition”, Proceedings EUROSPEECH Berlin, Germany, 1993, Vol. 3, 2125-2128

[Thelen+ 1997] E. Thelen, X. Aubert, P. Beyerlein, “Speaker Adaptation in the Philips System for Large Vocabulary Continuous Speech Recognition”, Proceedings ICASSP Munich, Germany, 1997, 1035-1038

[Tran+ 1997] B.-H. Tran, F. Seide, V. Steinbiss, “A Word Graph Based N-Best Search in Continuous Speech Recognition”, Proceedings ICSLP Philadelphia, Pennsylvania, U.S.A., 1996, 2127-2130

[Valtchev 1995] V. Valtchev, “Discriminative Methods in HMM-based Speech Recognition”, Ph.D. thesis, University of Cambridge, England, 1995

[Vergyri 1997] D. Vergyri, F. Jelinek, “Acoustic Sensitive Language Modeling”, Report III on Language and Acoustic Modeling in Technical Report 1/97-6/97 of CLSP, Johns Hopkins University, Baltimore, Maryland, U.S.A., 1997

[Wegmann+ 1998] S. Wegmann, F. Scattone, I. Carp, L. Gillick, R. Roth und J. Yamron. “Dragon System’s 1997 Broadcast News Transcription System”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 60-65

[Welling 1998] L. Welling, “Merkmalsextraktion in Spracherkennungssystemen für grossen Wortschatz”, eingereichte Dissertation am Lehrstuhl für Informatik VI, Rheinisch-Westfälische Technische Hochschule Aachen, Oktober 1998

[Wessel 1999] F. Wessel, private communication, Lehrstuhl für Informatik VI, Rheinisch-Westfälische Technische Hochschule Aachen, March 1999

[Woodland+ 1998] P. C. Woodland, T. Hain, S. E. Johnson, T. R. Niesler, A. Türk, E. W. D. Whittaker, S. J. Young, “The 1997 HTK Broadcast News Transcription System”, Proceedings of the DARPA 1998 Broadcast News Transcription and Understanding Workshop, Lansdown, Virginia, U.S.A., 1998, 41-48

[Young+ 1993] S.J. Young, P. C. Woodland, “The Use of State-Tying in Continuous Speech Recognition”, Proceedings EUROSPEECH Berlin, Germany, 1993, Vol. 3, 2203-2206

[Young+ 1995] S.J. Young, J.J. Odell, P.C. Woodland, “Tree-based Tying for High-Accuracy Acoustic Modelling”, Proceedings of the Spoken Language Systems Technology Workshop, Austin, Texas, U.S.A., 1995, 286-291

Teil V

Anhang

Kapitel 11

Herleitungen

11.1 Minimierung der geglätteten Satzfehlerrate

Dieser Abschnitt beinhaltet die Herleitung des Iterationsverfahrens zur Optimierung der Koeffizienten der log-linearen Verteilung p_{Λ} bezüglich der empirischen Fehlerrate in (4.10). Zunächst wird die Verlustfunktion ℓ für die log-lineare Verteilungskombination (4.5) definiert. Anschließend wird sie nach den Parametern $\lambda_j, j = 1, \ldots, M$ differenziert und die zugehörige Iterationsgleichung aufgestellt.
Analog zu [Juang$^+$ 1995] ergibt sich für das Mißklassifikationsmaß d:

$$\begin{aligned} d(x_n, k_n, \Lambda) &= -\log p_{\Lambda}(k_n|x_n) + \log \left(\frac{1}{K-1} \sum_{k \neq k_n} \exp\left\{ \eta \cdot \log p_{\Lambda}(k|x_n) \right\} \right)^{\frac{1}{\eta}} \\ &= -\log p_{\Lambda}(k_n|x_n) + \log \left(\frac{1}{K-1} \sum_{k \neq k_n} \left[p_{\Lambda}(k|x_n) \right]^{\eta} \right)^{\frac{1}{\eta}} \\ &= \log \left(\frac{1}{K-1} \sum_{k \neq k_n} \left[\frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)} \right]^{\eta} \right)^{\frac{1}{\eta}} . \end{aligned} \tag{11.1}$$

Um den Abstand der rivalisierenden Wortketten zu der korrekten Wortkette einzubeziehen, wird folgende Gewichtung mit dem Fehlermaß $\mathcal{L}(k, k_n)$ vorgenommen:

$$d(x_n, k_n, \Lambda) = \log \left(\frac{1}{K-1} \sum_{k \neq k_n} \exp\left\{ \eta \cdot \mathcal{L}(k, k_n) \log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)} \right\} \right)^{\frac{1}{\eta}} . \tag{11.2}$$

Diese Gewichtung verstärkt den Beitrag von Rivalen k mit großer Fehlerrate zum Mißklassifikationsmaß d. Rivalen, die eine sehr niedrige Fehlerrate haben, gehen in das Mißklassifikationsmaß nur schwach ein. Nehmen wir an, daß der Wert η sehr groß gewählt wird ($\eta \to \infty$). Dann ist

$$d(x_n, k_n, \Lambda) = \max_{k \neq k_n} \mathcal{L}(k, k_n) \log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)}$$

und die Wortfehlerrate geht linear in das Mißklassifikationsmaß ein. Das gleiche gilt auch, wenn wir annehmen, daß nur genau ein Rivale k_r existiert. Dann ist

$$\begin{aligned} d(x_n, k_n, \Lambda) &= \log \left(\exp \left\{ \eta \cdot \mathcal{L}(k_r, k_n) \log \frac{p_\Lambda(k_r|x_n)}{p_\Lambda(k_n|x_n)} \right\} \right)^{\frac{1}{\eta}} \\ &= \mathcal{L}(k_r, k_n) \log \frac{p_\Lambda(k_r|x_n)}{p_\Lambda(k_n|x_n)} . \end{aligned}$$

Auch hier geht die Wortfehlerrate $\mathcal{L}(k_r, k_n)$ linear in das Mißklassifikationsmaß d ein. Für die Verlustfunktion ℓ in (4.9) erhält man durch Umformung:

$$\begin{aligned} \ell(x_n, k_n, \Lambda) &= (1 + \exp\{-a(b + d(\ldots))\})^{-1} \\ &= (1 + \exp\{-ab\} (\exp\{-d(\ldots)\})^a)^{-1} \end{aligned} \tag{11.3}$$

Einsetzen von (11.2) ergibt schließlich mit $A = \exp\{-ab\}, B = a$:

$$\ell(x_n, k_n, \Lambda) = \left(1 + A \left(\frac{1}{K-1} \sum_{k \neq k_n} \left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} \right]^{\eta \cdot \mathcal{L}(k, k_n)} \right)^{-\frac{B}{\eta}} \right)^{-1} . \tag{11.4}$$

Mit (11.4) kann nun die Ableitung der Funktion E_{MCE} in (4.10) nach dem Parameter λ_j bestimmt werden.

$$\frac{\partial E_{MCE}(\Lambda)}{\partial \lambda_j} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial \ell(x_n, k_n, \Lambda)}{\partial \lambda_j} . \tag{11.5}$$

Unter Verwendung der Kettenregel für die Ableitung von Funktionen gilt weiterhin:

$$\begin{aligned} \frac{\partial \ell(x_n, k_n, \Lambda)}{\partial \lambda_j} &= \frac{\partial}{\partial \lambda_j} \left(1 + A (\exp\{-d(x_n, k_n, \Lambda)\})^B \right)^{-1} \\ &= -\left(1 + A (\exp\{-d(x_n, k_n, \Lambda)\})^B \right)^{-2} \cdot \\ &\quad \cdot B \cdot A (\exp\{-d(x_n, k_n, \Lambda)\})^{B-1} \\ &\quad \cdot (-\exp\{-d(x_n, k_n, \Lambda)\}) \cdot \frac{\partial d(x_n, k_n, \Lambda)}{\partial \lambda_j} \\ &= \left(1 + A (\exp\{-d(x_n, k_n, \Lambda)\})^B \right)^{-2} \cdot \\ &\quad \cdot A (\exp\{-d(x_n, k_n, \Lambda)\})^B \cdot B \frac{\partial d(x_n, k_n, \Lambda)}{\partial \lambda_j} . \end{aligned}$$

Da $(1+x)^{-2} x = (1+x)^{-1}(1 - (1+x)^{-1})$ ist, erhalten wir mit (11.3):

$$\frac{\partial \ell(x_n, k_n, \Lambda)}{\partial \lambda_j} = B \cdot \ell(x_n, k_n, \Lambda) (1 - \ell(x_n, k_n, \Lambda)) \cdot \frac{\partial d(x_n, k_n, \Lambda)}{\partial \lambda_j} \tag{11.6}$$

Für die Ableitung von d gilt mit (11.2)

$$\frac{\partial d(x_n,k_n,\Lambda)}{\partial \lambda_j} = \frac{\partial}{\partial \lambda_j}\log\left(\frac{1}{K-1}\sum_{k\neq k_n}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}\right)^{\frac{1}{\eta}}$$

Mit weiteren Umformungen folgt:

$$\begin{aligned}
\frac{\partial d(x_n,k_n,\Lambda)}{\partial \lambda_j} &= \frac{1}{\eta}\frac{\frac{1}{K-1}\sum_{k\neq k_n}\frac{\partial}{\partial\lambda_j}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}{\frac{1}{K-1}\sum_{k\neq k_n}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}} \\
&= \frac{1}{\eta}\frac{\sum_{k\neq k_n}\frac{\partial}{\partial\lambda_j}\left[\frac{\prod_{i=1}^M p_i(k|x_n)^{\lambda_i}}{\prod_{i=1}^M p_i(k_n|x_n)^{\lambda_i}}\right]^{\eta\mathcal{L}(k,k_n)}}{\sum_{k\neq k_n}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}} \\
&= \left(\eta\sum_{k\neq k_n}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}\right)^{-1}\cdot \\
&\quad\cdot\sum_{k\neq k_n}\frac{\partial}{\partial\lambda_j}\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right)\eta\mathcal{L}(k,k_n)\lambda_j\cdot \\
&\quad\cdot\left[\prod_{i=1,i\neq j}^M\left(\frac{p_i(k|x_n)}{p_i(k_n|x_n)}\right)\lambda_i\right]^{\eta\mathcal{L}(k,k_n)}\cdot
\end{aligned}$$

Wegen $\frac{\partial}{\partial x}a^{bx} = (b\log a)a^{bx}$ folgt weiter

$$\begin{aligned}
\frac{\partial d(x_n,k_n,\Lambda)}{\partial \lambda_j} &= \left(\eta\sum_{k\neq k_n}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}\right)^{-1}\cdot \\
&\quad\cdot\sum_{k\neq k_n}\eta\mathcal{L}(k,k_n)\log\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right)\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right)\eta\mathcal{L}(k,k_n)\lambda_j\cdot \\
&\quad\cdot\left[\prod_{i=1,i\neq j}^M\left(\frac{p_i(k|x_n)}{p_i(k_n|x_n)}\right)\lambda_i\right]^{\eta\mathcal{L}(k,k_n)} \\
&= \frac{\sum_{k\neq k_n}\mathcal{L}(k,k_n)\log\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right)\left[\prod_{i=1}^M\left(\frac{p_i(k|x_n)}{p_i(k_n|x_n)}\right)\lambda_i\right]^{\eta\mathcal{L}(k,k_n)}}{\sum_{k\neq k_n}\left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}
\end{aligned}$$

$$= \frac{\sum_{k \neq k_n} \mathcal{L}(k,k_n) \log\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right) \left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}{\sum_{k \neq k_n} \left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}. \tag{11.7}$$

Durch Kombination der Gleichungen (11.5), (11.6) und (11.7) erhalten wir schließlich:

$$\frac{\partial E_{MCE}(\Lambda)}{\partial \lambda_j} = \frac{B}{N} \sum_{n=1}^{N} \ell(x_n,k_n,\Lambda)\,(1-\ell(x_n,k_n,\Lambda)) \cdot \frac{\sum_{k \neq k_n} \mathcal{L}(k,k_n) \log\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right) \left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}{\sum_{k \neq k_n} \left[\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}. \tag{11.8}$$

Setzt man in (3.17) $U = I$ (Einheitsmatrix), dann erhält man mit (11.8) das Iterationsverfahren für die Koeffizienten der log-lineraren Verteilungskombination:

$$\begin{aligned}
\lambda_j^{(0)} &= 0 \quad \text{(Gleichverteilung)} \\
\lambda_j^{(I+1)} &= \lambda_j^{(I)} - \varepsilon \sum_{n=1}^{N} \ell(x_n,k_n,\Lambda^{(I)}) \left(1-\ell(x_n,k_n,\Lambda^{(I)})\right) \cdot \\
&\quad \cdot \frac{\sum_{k \neq k_n} \mathcal{L}(k,k_n) \log\left(\frac{p_j(k|x_n)}{p_j(k_n|x_n)}\right) \left[\frac{p_{\Lambda^{(I)}}(k|x_n)}{p_{\Lambda^{(I)}}(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}}{\sum_{k \neq k_n} \left[\frac{p_{\Lambda^{(I)}}(k|x_n)}{p_{\Lambda^{(I)}}(k_n|x_n)}\right]^{\eta\mathcal{L}(k,k_n)}} \\
\Lambda^{(I)} &= (\lambda_1^{(I)},\ldots,\lambda_M^{(I)})^T \\
&\quad j = 1,\ldots,M.
\end{aligned} \tag{11.9}$$

11.2 Minimierung der geglätteten Wortfehlerrate

Das Optimierungskriterium Wortfehlerrate $E(\Lambda)$ (3.14) wird in diesem Abschnitt wie folgt durch eine geglättete Zielfunktion approximiert: Sei

$$\begin{aligned} E_{MWE}(\Lambda) &= \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) S(k, n, \Lambda) \\ &= \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k} \mathcal{L}(k, k_n) S(k, n, \Lambda), \end{aligned} \tag{11.10}$$

mit $\mathcal{L}(k_n, k_n) = 0$.

Um ein differenzierbares Kriterium zu erhalten, wird die Indikatorfunktion durch folgende geglättete Indikatorfunktion ersetzt:

$$S(k, n, \Lambda) = \frac{\left(\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} \right)^\eta}{\sum_{k'} \left(\frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)} \right)^\eta}. \tag{11.11}$$

Der Wert $E(\Lambda)$ wird nun nach den Koeffizienten λ_i der Modellkombination abgeleitet. Um die Ableitung für verschiedene Kombinationsformen nur einmal durchzuführen, wird zunächst die Hilfsfunktion

$$f_i(k, n, \Lambda) = \frac{\partial \log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}}{\partial \lambda_i} \tag{11.12}$$

eingeführt. Die Form der Hilfsfunktion hängt von der Form der Verteilung p_Λ ab. Durch Ableitung nach λ_i ergibt sich dann:

$$\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} = \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) \frac{\partial}{\partial \lambda_i} S(k, n, \Lambda).$$

Es folgt

$$\begin{aligned} \frac{\partial S(k, n, \Lambda)}{\partial \lambda_i} = \\ \frac{\eta f_i(k, n, \Lambda) \exp\left\{ \eta \log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} \right\} \sum_{k'} \exp\left\{ \eta \log \frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)} \right\}}{\left(\sum_{k'} \exp\left\{ \eta \log \frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)} \right\} \right)^2} - \\ \frac{\exp\left\{ \eta \log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} \right\} \sum_{k'} \eta f_i(k', n, \Lambda) \exp\left\{ \eta \log \frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)} \right\}}{\left(\sum_{k'} \exp\left\{ \eta \log \frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)} \right\} \right)^2} \end{aligned}$$

$$\frac{\partial S(k,n,\Lambda)}{\partial \lambda_i} =$$

$$S(k,n,\Lambda)\frac{\eta f_i(k,n,\Lambda)\sum_{k'}\exp\left\{\eta\log\frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)}\right\}}{\sum_{k'}\exp\left\{\eta\log\frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)}\right\}} -$$

$$\frac{S(k,n,\Lambda)\sum_{k'}\eta f_i(k',n,\Lambda)\exp\left\{\eta\log\frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)}\right\}}{\sum_{k'}\exp\left\{\eta\log\frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)}\right\}}.$$

Durch Verwendung der Funktion S kann man wieder kurz schreiben:

$$\begin{aligned}\frac{\partial S(k,n,\Lambda)}{\partial \lambda_i} &= S(k,n,\Lambda)\eta f_i(k,n,\Lambda) - S(k,n,\Lambda)\sum_{k'}\eta f_i(k',n,\Lambda)S(k',n,\Lambda)\\ &= \eta S(k,n,\Lambda)\left(f_i(k,n,\Lambda) - \sum_{k'} f_i(k',n,\Lambda)S(k',n,\Lambda)\right).\end{aligned}$$

Einsetzen liefert

$$\begin{aligned}\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} &= \frac{1}{\sum_{n=1}^N L_n}\sum_{n=1}^N\sum_k \mathcal{L}(k,k_n)\eta S(k,n,\Lambda)\cdot\\ &\quad\cdot\left(f_i(k,n,\Lambda) - \sum_{k'} f_i(k',n,\Lambda)S(k',n,\Lambda)\right)\\ &= \frac{1}{\sum_{n=1}^N L_n}\sum_{n=1}^N\sum_k \mathcal{L}(k,k_n)\eta S(k,n,\Lambda)f_i(k,n,\Lambda) -\\ &\quad\sum_{n=1}^N\sum_k \mathcal{L}(k,k_n)\eta S(k,n,\Lambda)\sum_{k'} f_i(k',n,\Lambda)S(k',n,\Lambda)\\ &= \frac{1}{\sum_{n=1}^N L_n}\sum_{n=1}^N\sum_k \mathcal{L}(k,k_n)\eta S(k,n,\Lambda)f_i(k,n,\Lambda) -\\ &\quad\sum_{n=1}^N\sum_{k'}\sum_k \mathcal{L}(k,k_n)\eta S(k,n,\Lambda)f_i(k',n,\Lambda)S(k',n,\Lambda)\\ &= \frac{\eta}{\sum_{n=1}^N L_n}\left\{\sum_{n=1}^N\sum_k S(k,n,\Lambda)f_i(k,n,\Lambda)\mathcal{L}(k,k_n) -\right.\\ &\quad\left.\sum_{n=1}^N\sum_{k'} S(k',n,\Lambda)f_i(k',n,\Lambda)\sum_k \mathcal{L}(k,k_n)S(k,n,\Lambda)\right\}\end{aligned}$$

Durch Vertauschen der Indizes k, k' erhalten wir schließlich:

$$\begin{aligned}\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} &= \frac{\eta}{\sum_{n=1}^{N} L_n} \left\{ \sum_{n=1}^{N} \sum_{k} S(k,n,\Lambda) f_i(k,n,\Lambda) \mathcal{L}(k,k_n) - \right. \\ &\quad \left. \sum_{n=1}^{N} \sum_{k} S(k,n,\Lambda) f_i(k,n,\Lambda) \sum_{k'} \mathcal{L}(k',k_n) S(k',n,\Lambda) \right\} \\ &= \frac{\eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k} S(k,n,\Lambda) f_i(k,n,\Lambda) \cdot \\ &\quad \cdot \left(\mathcal{L}(k,k_n) - \sum_{k'} S(k',n,\Lambda) \mathcal{L}(k',k_n) \right).\end{aligned}$$

Also ist

$$\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} = \frac{\eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k} S(k,n,\Lambda) f_i(k,n,\Lambda) \left(\mathcal{L}(k,k_n) - \overline{\mathcal{L}(k_n)} \right),$$

wobei

$$\overline{\mathcal{L}(k_n)} = \sum_{k'} S(k',n,\Lambda) \mathcal{L}(k',k_n)$$

die geglättete Fehlerrate der Trainingsbeobachtung x_n ist.

11.2.1 Log-lineare Kombination

Für den Fall der log-linearen Kombination ist

$$f_i(k,n,\Lambda) = \log \frac{p_i(k|x_n)}{p_i(k_n|x_n)}.$$

Für die gesamte Ableitung nach λ_i ergibt sich folglich

$$\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} = \frac{\eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k} S(k,n,\Lambda) \log \frac{p_i(k|x_n)}{p_i(k_n|x_n)} \left(\mathcal{L}(k,k_n) - \overline{\mathcal{L}(k_n)} \right).$$

11.2.2 Lineare Kombination

Für den Fall der linearen Kombination ist

$$\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} = \frac{\sum_i \lambda_i \cdot p_i(k|x_n)}{\sum_i \lambda_i \cdot p_i(k_n|x_n)}. \tag{11.13}$$

Dadurch erhält man für

$$f_i(k,n,\Lambda) = \frac{p_i(k|x_n)}{\sum_j \lambda_j p_j(k|x_n)} - \frac{p_i(k_n|x_n)}{\sum_j \lambda_j p_j(k_n|x_n)}. \tag{11.14}$$

Für die gesamte Ableitung nach λ_i ergibt sich folglich

$$\begin{aligned}\frac{\partial E_{MWE}(\Lambda)}{\partial \lambda_i} &= \frac{\eta}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{k} S(k,n,\Lambda) \left(\mathcal{L}(k,k_n) - \overline{\mathcal{L}(k_n)} \right) \cdot \\ &\quad \cdot \left(\frac{p_i(k|x_n)}{\sum_j \lambda_j p_j(k|x_n)} - \frac{p_i(k_n|x_n)}{\sum_j \lambda_j p_j(k_n|x_n)} \right).\end{aligned}$$

Symbolverzeichnis

Symbol	**Beschreibung**
H	Entropie
p	Wahrscheinlichkeitsverteilung
ω	Ereignis
Ω	Ereignismenge
M	Anzahl der zu integrierenden Modelle bzw. charakterisierenden Funktionen
$m_j(\omega), j = 1, \ldots, M$	charakterisierende Funktionen einer Maximum-Entropie Verteilung
$\lambda_j, j = 1, \ldots, M$	freie Parameter einer Maximum-Entropie Verteilung
C(...)	Normierungsterm einer Wahrscheinlichkeitsverteilung
x	Beobachtung (akustische Merkmale)
k, k'	Klasse (Wortfolge)
K	Anzahl aller Klassen $k, k' = 1, \ldots, K$
$\pi(k\|x)$	wahre a-posteriori Wahrscheinlichkeit der Sprache
$p(k\|x)$	Modellwahrscheinlichkeit für die Klasse k gegeben die Beobachtung x
$p_{LM}(k\|x)$	Basismodell, vom Sprachmodell abhängende Wahrscheinlichkeit
$p_{AM}(k\|x)$	Basismodell, vom akustischen Modell abhängende Wahrscheinlichkeit
$p_j(k\|x)$	Basismodell, von Modell j abhängende Wahrscheinlichkeit

Symbol	**Beschreibung**
$\Lambda = (\lambda_1, \ldots, \lambda_M)^T$	Koeffizienten der Modellkombination
$p_\Lambda(k\|x)$	Basismodellkombination für die Klasse k gegeben die Beobachtung x
x_n	Beobachtung (akustische Merkmale) des n-ten Trainingssatzes
k_n	korrekte Klasse (Wortfolge) des n-ten Trainingssatzes
L_n	Länge des n-ten Trainingssatzes (Anzahl der Worte)
$E(\Lambda)$	Fehlerrate des von den freien Koeffizienten Λ abhängenden Klassifikators
$\mathcal{L}(k, k_n)$	Levenshtein-Distanz zwischen Symbolfolge (Wortfolge) k und k_n
$\breve{\mathcal{L}}(k, k_n)$	Monoton steigende Funktion von $\mathcal{L}(k, k_n)$
$\tilde{\mathcal{L}}(k, n, \Lambda)$	Mittelwertfreie Fehlerzahl, Differenz der Fehlerzahl der Hypothese k zur mittleren gewichteten Fehlerzahl der Trainingsbeobachtung x_n
$d(x_n, k_n, \Lambda)$	Mißklassifikationsmaß (Rivalität) des n-ten Trainingssatzes
$\ell(x_n, k_n, \Lambda)$	Verlustfunktion des $n - ten$ Trainingssatzes
w_1^L	Wortfolge der Länge L
$\vec{x}$	akustischer Merkmalsvektor
$\vec{x}_1^T$	akustische Merkmalsvektorfolge der Länge T
g	Diskriminantenfunktion
S	(geglättete) Indikatorfunktion
P	Vektor, der die gewichtete Korrelation zwischen der Fehlerrate und den Diskriminantenfunktionen der Basismodelle enthält
Q	Matrix, die die gewichtete paarweise Korrelation der Diskriminantenfunktionen der Basismodelle enthält

Symbol	Beschreibung
$\mathbf{Y}_{k,x_n}$	Merkmalsvektor (Theorie der linearen Klassifikatoren), der aus den Logarithmen der Basismodellbewertungen besteht
ω_r	Klasse, die die Menge der Merkmalsvektoren $\mathbf{Y}_{k,x_n}$ mit $k \neq k_n$ beschreibt
ω_c	Klasse, die die Menge der Merkmalsvektoren $\mathbf{Y}_{k,x_n}$ mit $k = k_n$ beschreibt
μ_r, Σ_r	Mittelwertvektor und Kovarianzmatrix der Klasse ω_r
μ_c, Σ_c	Mittelwertvektor und Kovarianzmatrix der Klasse ω_c
$h(\mathbf{Y}_{k,x_n})$	Klassifikator, der den Merkmalsvektor $\mathbf{Y}_{k,x_n}$ in den eindimensionalen h-Raum abbildet
η_r, σ_r	Mittelwert und Varianz der Klasse ω_r im h-Raum
η_c, σ_c	Mittelwert und Varianz der Klasse ω_c im h-Raum
s	Zustand eines Hidden-Markoff-Modells
s_1^T	HMM-Zustandsfolge der Länge T
ϕ_1^H, ξ_1^H	Phonemfolge, Triphonfolge der Länge H

Abkürzungsverzeichnis

ARPA	Advanced Research Projects Agency
ASRU	Automatic Speech Recognition and Understanding
DARPA	Defense Advanced Research Projects Agency
DMC	Discriminative Model Combination
EUROSPEECH	European Conference on Speechprocessing
HMM	Hidden Markoff Modell
ICASSP	International Conference on Acoustics, Speech and Signal Processing
ICSLP	International Conference on Speech and Language Processing
IEEE	Institute of Electrical and Electronics Engineers
LDC	Linguistic Data Consortium
MCE	Minimum Classification Error
MWE	Minimum Word Error
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information
NIST	National Institute for Standardization
RWTH	Rheinisch-Westfälische Technische Hochschule
USE	Unified Stochastic Engine

ug	Unigrammsprachmodell
bg	Bigrammsprachmodell
tg	Trigrammsprachmodell
ww	Wortinternes Triphonmodell
wwad	MLLR-adaptiertes wortinternes Triphonmodell
xw	Wortübergreifendes Triphonmodell
xwad	MLLR-adaptiertes wortübergreifendes Triphonmodell
5wwad	MLLR-adaptiertes wortinternes Pentaphonmodell
d1bg	phrasenbasiertes Bigramm-Lücken Sprachmodell mit einer Lückenlänge von 1
lltg	log-lineare Kombination aus ug, bg, tg, d1bg
wsj	log-lineare Kombination aus auf der WSJ0+1 Trainingsdatenbasis trainierten ww-, xw-, 5ww-Modellen, die MLLR-adaptiert wurden

Tabellenverzeichnis

Abbildungsverzeichnis

Lebenslauf

Persönliche Daten

Peter Beyerlein, geboren am 04.05.1968 in Berlin

Schul-/Berufsausbildung

1974-1986	Schulausbildung in Berlin
1986-1987	Filmfabrik Wolfen, Bitterfeld, Berufsausbildung zum Elektromonteur

Studium

1987-1992	Studium, Technische Universität Dresden, Institut für Nachrichtentechnik
1988-1992	Berufung in die Meisterklasse “Mikroelektronik” der Technischen Universität Dresden
1990-1991	Praktikum, Internationales Institut für Kernforschung, Dubna bei Moskau, UdSSR
1991-1992	Diplomarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung

Beruf

seit 1992	Philips GmbH Forschungslaboratorien Aachen
1992-1996	Wissenschaftlicher Mitarbeiter
1997-2000	Teamleiter für DARPA Broadcast-News Transcription Benchmarking
2000-2001	Projektleiter EVAL, Recognition of Conversational Speech
seit 2000	Senior Scientist, Philips GmbH Forschungslaboratorien Aachen
seit 2001	Clusterleiter HAL, Human-like Automatic Learning

Freizeit/Hobbies

Segelfliegen, Fitness