Implementierung und Vergleich diskriminativer Verfahren für Spracherkennung bei kleinem Vokabular

Diplomarbeit am Lehrstuhl für Informatik VI der RWTH Aachen

Prof. Dr.-Ing. H. Ney

vorgelegt von:

Cand. Inform. Wolfgang Macherey
Matrikelnummer 191 280

Gutachter:

Prof. Dr.-Ing. H. Ney
Prof. Dr. G. Lakemeyer

Betreuer:

Dipl.-Physiker Ralf Schlüter

Hiermit versichere ich, daß ich die vorliegende Diplomarbeit selbständig verfaßt und keine weiteren als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge, Graphiken und Tabellen, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Quellenverweise gekennzeichnet.

Aachen, den 20. November 1998

(Wolfgang Macherey)

Die Dichter behaupten, dass die Naturwissenschaften den Sternen ihre Schönheit nehmen und sie zu einer Ansammlung von Gas-Atomen machen. Auch ich kann die Sterne in der Nacht sehen und fühlen. Die Grenzenlosigkeit des Himmels weitet meine Phantasie. Und mein Auge kann am Firmament Licht wahrnehmen, das vor einer Million Jahre entstanden ist. Dem Geheimnis schadet es nicht, etwas mehr darüber zu wissen.

Richard Feynman, Amerikanischer Physiknobelpreis-Träger von 1965

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Danksagung

Ich möchte an dieser Stelle all denen danken, die mir in der Zeit der Erstellung dieser Diplomarbeit und während meines Studiums zur Seite standen.

Ich danke im besonderen:

Herrn Prof. Dr.-Ing. Hermann Ney für die sehr aktuelle Themenstellung und sein großes Interesse während der Anfertigung dieser Diplomarbeit,

Herrn Dipl.-Physiker Ralf Schlüter für die vorbildliche Betreuung und die zahlreichen Diskussionen,

Herrn Prof. Dr. G. Lakemeyer für die Übernahme des Koreferates,

Meinen Kollegen am Lehrstuhl für das große Interesse und die Unterstützung, auf die ich immer wieder gestoßen bin,

und vor allem meinen Eltern Ria und Peter Macherey für ihre Liebe, Unterstützung und das Ermöglichen meines Studiums.

Kapitel 1

Einleitung

Die automatische Erkennung natürlich gesprochener Sprache gehört zu den wichtigen technischen Innovationen im Bereich der Mensch-Maschine-Kommunikation. Unter den hierzu vorgeschlagenen und untersuchten Verfahren hat der statistische Ansatz in Verbindung mit der *Hidden-Markov-Modellierung* die automatische Spracherkennung am nachhaltigsten beeinflußt und bildet bis heute den *de facto* Standard.

Grundlage des statistischen Verarbeitungsparadigmas ist die *Bayessche Entscheidungsregel*. Hat man eine Folge akustischer Beobachtungsvektoren als Merkmale einer lautsprachlichen Äußerung gegeben, liefert die Bayessche Entscheidungsregel eine Methode, um diejenige Wortfolge zu finden, die durch die Beobachtungen am besten erklärt wird. Dabei wird das Erkennungsergebnis entscheidend durch die Einstellung der Verteilungsparameter des zugrunde gelegten statistischen Modells bestimmt.

Die Schwierigkeit besteht nun darin, die freien Parameter des statistischen Klassifikators vorab in einer Lernphase so zu schätzen, daß die erkannte Wortfolge möglichst häufig der gesprochenen Wortfolge entspricht. Hierzu trainiert man das System auf Basis der Beobachtungen einer vorher festgelegten Lernstichprobe. Da die Zahl der Freiheitsgrade häufig mehrere tausend bis hunderttausend zu schätzende Parameter umfaßt, bildet das Training ein komplexes Optimierungsproblem.

Üblicherweise verwendet man zur Schätzung der Modellparameter die aus der Statistik bekannte *Maximum-Likelihood* (ML) Methode. Konventionelle Trainings-Algorithmen, die auf der Grundlage dieses Schätzers arbeiten, trainieren jedoch nur klassenindividuell und lassen den Überlapp zwischen den Klassen unberücksichtigt. Eine Alternative hierzu bilden die *diskriminativen Lernverfahren*. Im Unterschied zum konventionellen ML-Ansatz berücksichtigen diskriminative Verfahren auch klassenfremde Trainingsdaten und konzentrieren sich somit verstärkt auf die Klassentrennung. Dies hat in den letzten Jahren zu signifikanten Verbesserungen in der Erkennungsleistung geführt.

Gleichwohl sieht man sich bei der praktischen Umsetzung mit erheblichen Problemen konfrontiert. Neben dem deutlich höheren Berechnungsaufwand, den die diskriminativen Verfahren im Vergleich zum konventionellen ML-Ansatz erfordern, erweisen sich zahlreiche bisher bewährte Modellannahmen im Kontext des diskriminativen Trainings als wenig brauchbar.

Zu den Zielen dieser Diplomarbeit gehören daher neben der Entwicklung und Implementierung effizienter Algorithmen auch der Entwurf und die Realisierung neuer, praxistauglicher Modelle. Diese Modelle sollen dann sowohl im diskriminativen Training als auch im konventionellen ML-Ansatz Verwendung finden. Das Hauptziel der Arbeit bilden Untersuchungen zur Verbesserung der Erkennungsleistung mittels diskriminativer Verfahren. Zudem wird ein qualitativer Vergleich der diskriminativen Verfahren untereinander durchgeführt. Die Zielsetzungen der Arbeit sind im einzelnen:

- **Vergleich diskriminativer Kriterien**
  Im Rahmen dieser Diplomarbeit wird eine Klasse diskriminativer Kriterien untersucht, zu denen das *Corrective Training* (CT)-, das *Maximum Mutual Information* (MMI)-, das falsifizierende Training (FT)- und das *Minimum Classification Error* (MCE)-Kriterium gehören. Die Kriterien unterscheiden sich im wesentlichen durch die Verwendung einer optionalen Glättungsfunktion, einem Exponenten zur Umwichtung der Satzwahrscheinlichkeit, sowie der Auswahl der zu betrachtenden konkurrierenden Wortfolgen. Während beim CT- und FT-Kriterium die Menge der konkurrierenden Wortfolgen auf die beste (ggf. inkorrekt) erkannte Wortfolge eingeschränkt wird, greift man beim MMI- und MCE-Kriterium auf *Wortgraphen* oder *N-Besten-Listen* zurück. Obwohl letztere immer nur eine Approximation an Wortgraphen darstellen, werden sie auf Grund ihrer einfachen Struktur von vielen Forschungsgruppen eingesetzt. Im Rahmen dieser Arbeit werden dagegen Wortgraphen vewendet, da sie eine exaktere und zugleich effizientere Bestimmung der Wortwahrscheinlichkeiten ermöglichen.

- **Vergleich diskriminativer Methoden**
  Zur Reestimation der Verteilungsparameter stehen im diskriminativen Training zwei Optimierungsmethoden zur Verfügung: der *erweiterte Baum* Algorithmus und ein *Gradientenverfahren*. Für eine spezielle Wahl der Schrittweiten im Gradientenverfahren lassen sich für beide Methoden formal nahezu identische Reestimationsgleichungen herleiten [Schlüter et al. 97]. Gegenstand des Vergleichs ist die Fragestellung, ob beide Methoden auch bei Mischverteilungen übereinstimmende Resultate liefern.

- **Effiziente Wortgraphgenerierung**
  Bei Verwendung des MMI- und MCE-Kriteriums muß während des Trainings in jeder Iteration für jede Äußerung ein Wortgraph generiert werden, um die Summe der Satzwahrscheinlichkeiten über alle konkurrierenden Wortfolgen bestimmen zu können. Unter der Annahme, daß sich die Anordnung der Worthypothesen nicht mehr ändert, sondern nur eine zeitliche Verschiebung in den Wortgrenzen stattfindet, kann eine erneute Erkennung auf diejenigen Hypothesen eingeschränkt (fokussiert) werden, die der aus einer vorherigen Iteration stammende Wortgraph an dieser Stelle vorsieht. Eine solche eingeschränkte Erkennung wird implementiert und hinsichtlich ihrer Stabilität, der erforderlichen Wortgraphdichte und des Laufzeitgewinns untersucht.

- **Untersuchung diskriminativer Verfahren bei hoher Parameter-Anzahl**
  Viele Forschungsgruppen berichten von einer Abnahme der Performanz bei steigender Parameterzahl [Valtchev et al. 96, Normandin 96]. Unter Verwendung einer (die Performanz der konventionellen ML-Methode) ausschöpfenden Zahl an Einzelverteilungen soll diese Aussage für kleines Vokabular überprüft werden.

- **Parameter-Splitting im diskriminativen Training**
  Mit Hilfe diskriminativer Statistiken lassen sich gezielt diejenigen Einzelverteilungen bestimmen, die die Beobachtungen im Training auch nach vielen Iterationen nur unzureichend, das heißt mit einem großen Fehler behaftet, erklären. Konzentriert man sich beim Splitting auf diese Verteilungen, so können auch mit wenigen Mischverteilungen gute Modelle geschätzt werden. Gemäß diesem Prinzip wird ein entsprechendes Verfahren implementiert und getestet.

- **Vergleich: Viterbi vs. Baum-Welch**
  Bisher wurden die diskriminativen Verfahren im Rahmen der statistischen Spracherkennung ausschließlich in der *Viterbi*-Näherung angewendet. Bei dieser Näherung handelt es sich um eine Maximum-Approximation an das auf L. E. Baum zurückgehende Baum-Welch Training. Gegenstand der Untersuchung ist die Fragestellung, wie die Viterbi-Näherung und das Baum-Welch Training die durch diskriminative Verfahren erzielte Performanz beeinflussen.

- **Vergleich: konventionelle LDA vs. lineare MMI-Analyse (LMA)**
  Eines der bekannten Probleme in der Mustererkennung tritt in Verbindung mit hochdimensionalen Merkmalsräumen auf. Je höher die Dimension des Merkmalsraumes ist, desto schwieriger wird es, zuverlässige Klassifikatoren zu konstruieren, selbst wenn die Beobachtungen im hochdimensionalen Raum gut separierbar sind. Die Ursache hierfür liegt darin, daß die Anzahl der Trainingsdaten, die benötigt werden, um einen statistischen Klassifikator sicher zu schätzen, exponentiell mit der Dimension des Merkmalsraumes wächst. Eine häufig verwendete Strategie ist es daher, den hochdimensionalen Raum auf einen Unterraum abzubilden, wobei die Separierbarkeit der Klassen bestmöglichst erhalten bleiben soll. Dies ermöglicht die *lineare Diskriminanzanalyse* (LDA), die bereits seit Jahren erfolgreich in der Spracherkennung eingesetzt wird. Ausgehend vom MMI-Kriterium erfährt die LDA eine Erweiterung, die zur *linearen MMI-Analyse* (LMA) führt. In dieser Arbeit werden die Möglichkeiten der LMA im Vergleich zur LDA untersucht.

Zu den oben genannten Punkten werden zahlreiche Experimente auf dem englischsprachigen TI-Digits-Korpus, bzw. dem deutschsprachigen SieTill-Korpus durchgeführt. Bei beiden Datensammlungen handelt es sich um Korpora für die sprecherunabhängige Verbundziffernkettenerkennung. Die erzielten Resultate werden abschließend mit den Ergebnissen anderer Forschungsgruppen verglichen.

Um den Unterschied zwischen dem ML-Ansatz und dem diskriminativen Training aufzuzeigen, beginnt die Ausarbeitung in Kapitel 2 mit einer kompakten Einführung über die Grundlagen der statistischen Spracherkennung, wie sie im Rahmen des ML-Ansatzes Verwendung finden. Hierauf aufbauend wird in Kapitel 3 der diskriminative Ansatz beschrieben. Kapitel 4 und 5 behandeln Aspekte der Implementierung, die die Organisation der Suche im diskriminativen Training und die Berechnung der *forward-backward* Wahrscheinlichkeiten betreffen. Im Anschluß an das Kapitel 6 über lineare Merkmalstransformationen wird in Kapitel 7 ein Verfahren zum Splitting im diskriminativen Training vorgestellt. Eine Beschreibung der Experimente sowie eine Diskussion der Ergebnisse schließen die vorliegende Arbeit ab.

Kapitel 2

Grundlagen der statistischen Spracherkennung

Die Aufgabe der automatischen Spracherkennung ist die rechnerbasierte Umwandlung natürlich gesprochener Sprache in einen Text. Was dem Menschen jedoch als selbstverständliche Fähigkeit erscheinen mag, ist in der Umsetzung auf eine Maschine mit zahlreichen Schwierigkeiten verbunden. Einige der Ursachen hierfür sind in den folgenden Punkten zusammengefaßt.

- Die Aussprache eines Lautes kann stark variieren, sogar wenn es sich um dasselbe Wort und denselben Sprecher handelt.
- Die Sprechgeschwindigkeit unterliegt in der Regel starken Schwankungen.
- Die akustische Realisierung eines Lautes hängt im allgemeinen auch vom lautlichen Kontext ab, d.h. von den vorangehenden und nachfolgenden Lauten (man bezeichnet diesen Effekt auch als *Koartikulation*).
- Eine weitere Rolle spielen Aussprachevarianten, insbesondere die lautverschleifenden Assimilationen, bei denen mitunter ganze Lautgruppen ausgelassen werden.
- Schließlich kann das Sprachsignal durch Hintergrundgeräusche gestört sein.

Weitere Faktoren, die den Schwierigkeitsgrad des Erkennungsproblems bestimmen, sind die Größe des verfügbaren Wortschatzes, die Verwendung isoliert oder kontinuierlich gesprochener Sprache, sowie die sprecherabhängige oder sprecherunabhängige Erkennung.

2.1 Aufbau eines Spracherkennungssystems

Ausgangspunkt der statistischen Spracherkennung ist das digitalisierte Sprachsignal. Dieses Signal ist allerdings aufgrund seiner hohen Redundanz für die Klassifikation noch ungeeignet. Man extrahiert daher eine Folge von Merkmalsvektoren, die während der akustischen Analyse aus dem digitalisierten Sprachsignal gewonnen werden. Die Merkmalsvektoren geben die spektrale Energieverteilung des Sprachsignals wieder und werden unter Verwendung einer schnellen Fouriertransformation (FFT) alle 10 Millisekunden erzeugt, so daß man eine zeitliche Folge von Beobachtungen $x_1,...,x_T \in I\!R^D$ über die Zeitachse $t = 1,...,T$ erhält. Hierbei bezeichnet D die Dimension des Merkmalsraumes.

Die Aufgabe eines Spracherkenners besteht nun darin, zu einer Folge akustischer Vektoren $x_1,...,x_T$ die gesprochene Wortfolge $w_1,...,w_N$ zu finden. Die Güte des Erkennungssystems wird dabei nach der Anzahl der auftretenden Erkennungsfehler auf einer bisher ungesehenen Teststichprobe beurteilt. Im allgemeinen ermittelt man die Fehlerrate mit Hilfe der *Levenshtein*-Distanz. Diese gibt die minimale Anzahl der Worteinfügungen (*insertions*), Wortlöschungen (*deletions*) bzw. Wortersetzungen (*substitutions*) an, die notwendig sind, um die gesprochene Wortfolge in die erkannte umzuwandeln Die Levenshtein-Distanz ist somit ein direktes Maß für den Grad der Übereinstimmung zwischen erkannter und gesprochener Wortfolge.

In der statistischen Spracherkennung versucht man nun, den stochastischen Zusammenhang von $X = x_1,...,x_T$ und $W = w_1,...,w_N$ zu erfassen und damit die durchschnittliche Fehlerrate eines Klassifikators zu minimieren. Der statistische Ansatz stützt sich hierbei auf die *Bayessche Entscheidungsregel*. Aussage dieser Regel ist, daß die Fehlerrate im Mittel minimal ist, wenn die erkannte Wortfolge $\mathcal{R}(X)$ so gewählt wird, daß die a-posteriori Wahrscheinlichkeit $Pr(w_1^N|x_1^T)$ ihr Maximum erreicht:

$$\mathcal{R}(X) = \underset{w_1,...,w_N}{\operatorname{argmax}} \left\{ Pr(w_1^N|x_1^T) \right\} \tag{2.1}$$

Die Posterior-Wahrscheinlichkeit läßt sich nun mit Hilfe der Bayesschen Identität wie folgt umschreiben:

$$Pr(w_1^N|x_1^T) = \frac{Pr(x_1^T|w_1^N) \cdot Pr(w_1^N)}{Pr(x_1^T)} \tag{2.2}$$

Unter der Annahme, daß die Wahrscheinlichkeit der akustischen Vektorfolge $Pr(x_1^T)$ unabhängig von der Wortfolge w_1^N ist, braucht sie für die Argument-Maximierung in Gleichung (2.1) nicht weiter berücksichtigt zu werden. Damit ergibt sich die folgende Form der Bayesschen Entscheidungsregel:

$$\mathcal{R}(X) = \underset{w_1,...,w_N}{\operatorname{argmax}} \left\{ Pr(x_1^T|w_1^N) \cdot Pr(w_1^N) \right\} \tag{2.3}$$

Die Entscheidungsregel kann also auf die Maximierung des Produkts zweier Wahrscheinlichkeitsverteilungen zurückgeführt werden. Diese sind:

- die klassenbedingte oder auch akustische Wahrscheinlichkeit $Pr(x_1^T|w_1^N)$. Diese Größe beschreibt die Wahrscheinlichkeit für die Beobachtung der akustischen Vektorfolge $x_1,...,x_T$ bei gegebener Wortfolge $w_1,...,w_N$ und wird durch die akustische Modellierung bestimmt,
- die a-priori Wahrscheinlichkeit $Pr(w_1^N)$. Diese Größe beschreibt die Wahrscheinlichkeit für das Auftreten der Wortfolge $w_1,...,w_N$ und kann z.B. durch statistische Sprachmodellierung beschrieben werden.

Für die Erkennung müssen beide Wahrscheinlichkeiten bestimmt und geeignet kombiniert werden. Diese Aufgabe löst ein Suchprozeß, in dessen Verlauf der Rechner eine große Zahl an Wortfolgen, sogenannte Hypothesen bewertet. Abbildung 2.1 zeigt schematisch den Aufbau eines Erkennungssystems.

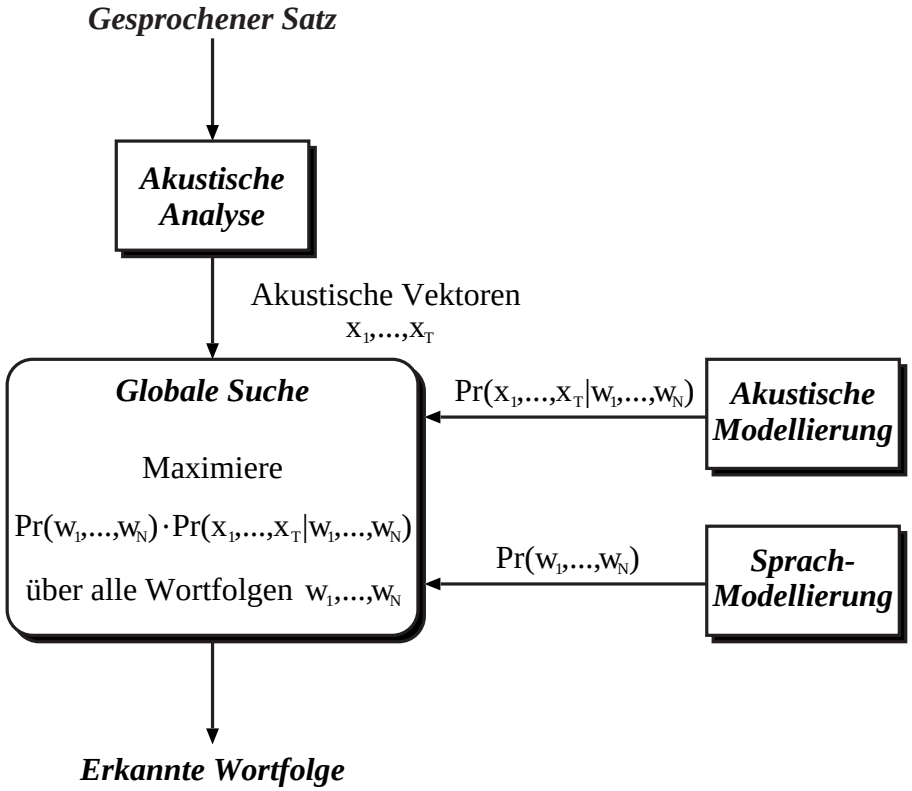


Abbildung 2.1: Architektur eines Spracherkennungssystems

2.2 Akustische Modellierung

Die akustische Wahrscheinlichkeit verknüpft eine Folge von Merkmalsvektoren mit den Wörtern des Vokabulars. Hierzu wird jedes Wort zuvor in einem Aussprachelexikon durch eine Folge von Wortuntereinheiten beschrieben. Die Art der verwendeten Wortuntereinheit richtet sich dabei nach der Vokabulargröße und dem Umfang des zur Verfügung stehenden Trainingsmaterials. Während bei der Ziffernkettenerkennung üblicherweise Ganzwortmodelle eingesetzt werden, verwendet man bei Vokabulargrößen ab hundert und mehr Wörtern hauptsächlich Silben, Halbsilben oder Phoneme.
Betrachtet man nun die sprachliche Realisierung einer Wortfolge, so zeigt sich, daß die konstituierenden Laute mit variabler Dauer und unterschiedlicher spektraler Zusammensetzung realisiert werden. Jeder Laut der Äußerung wird daher eine unbekannte Anzahl akustischer Vektoren auf sich vereinigen. Die zeitliche Verzerrung unterschiedlicher Wortaussprachen und die spektrale Variation müssen nun durch eine geeignete Modellierung erfaßt werden. Hier hat sich die Beschreibung durch *Hidden-Markov-Modelle* (HMM) bewährt, die die Variation im akustischen Signal durch einen zweistufigen stochastischen Prozeß zu erfassen suchen.

Das HMM ist ein stochastischer, endlicher Automat, mit dem Folgen von kontinuierlichen Merkmalsvektoren kompakt beschrieben werden können. Der Automat besteht aus einer Menge abstrakter Zustände S_q=$\{s_1,...,s_q\}$ und Transitionen $s_i \curvearrowright s_j$. Jeder Zustand s des Automaten ist mit einer Emissionsverteilung $Pr(x|s)$ assoziiert, die jedem Merkmals-

vektor x diejenige Wahrscheinlichkeit zuordnet, mit der der Vektor in s beobachtet werden kann. Übergänge zwischen den Zuständen werden durch sogenannte Transitionswahrscheinlichkeiten $Pr(s_t|s_{t-1})$ beschrieben. Ein Pfad $s_1,...,s_T$ in dem Automaten entspricht nun einer zeitlichen Zuordnung der Merkmalsvektoren $x_1,...,x_T$ auf die Zustände des Automaten und kann durch die Verbundwahrscheinlichkeit $Pr(x_1^T, s_1^T)$ bewertet werden. Mit der Bayesschen Dekompositionsregel läßt sich diese nun wie folgt umformen:

$$Pr(x_1^T, s_1^T) = \prod_{t=1}^{T} Pr(x_t, s_t|x_1^{t-1}, s_1^{t-1}). \tag{2.4}$$

Unter der Annahme, daß die Wahrscheinlichkeiten der bedingten Verbundereignisse nur von den unmittelbaren Vorgängerzuständen abhängen (dies ist die Markov-Eigenschaft), erhält man:

$$Pr(x_1^T, s_1^T) = \prod_{t=1}^{T} Pr(x_t, s_t|s_{t-1}) \tag{2.5}$$

$$= \prod_{t=1}^{T} Pr(x_t|s_{t-1}, s_t) \cdot Pr(s_t|s_{t-1}). \tag{2.6}$$

Zusätzlich nimmt man für $Pr(x_t|s_{t-1}, s_t)$ die Unabhängigkeit von s_{t-1} an:

$$Pr(x_1^T, s_1^T) = \prod_{t=1}^{T} Pr(x_t|s_t) \cdot Pr(s_t|s_{t-1}). \tag{2.7}$$

Das Produkt aus Emissions- und Transitionswahrscheinlichkeiten kann als ein Maß dafür interpretiert werden, wie gut die Beobachtungsvektorfolge für einen gegebenen Pfad durch das Markovmodell erklärt wird. Die nicht linearen Schwankungen in der Sprechgeschwindigkeit werden dabei durch unterschiedliche Folgen von Transitionen aus der Menge der erlaubten Zustandsübergänge erfaßt. Um nun die Wortuntereinheiten des Aussprachelexikons, beispielsweise die Phoneme mit ihren akustischen Realisierungen zu verknüpfen, wird jedes Phonem mit einem eigenen Hidden-Markov-Modell assoziiert.

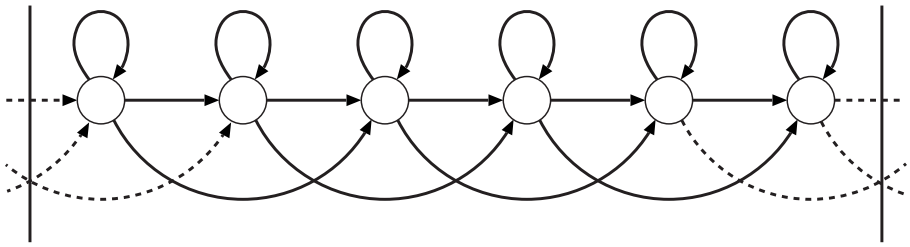


Abbildung 2.2: Markov-Modell eines Phonems

In der Praxis verwendet man zur akustischen Modellierung der Phoneme häufig einen aus sechs Zuständen bestehenden Automaten in Kombination mit einem *Bakis*-Modell (vgl. Abbildung 2.2). Das Bakis-Modell schränkt die erlaubten Transitionen auf das Verbleiben in einem Zustand (*loop*), den Übergang zum nächsten Zustand, (*forward*) sowie das

Überspringen des benachbarten Zustandes (*skip*) ein. Soll nun die akustische Wahrscheinlichkeit einer Wortfolge zu einer gegebenen Folge von Merkmalsvektoren bestimmt werden, werden zunächst die mit den zugehörigen Wortuntereinheiten assoziierten Markovmodelle (Mikroautomaten) verkettet und anschließend die Summe der Verbundwahrscheinlichkeiten über alle Pfade des resultierenden Makroautomaten bestimmt. Die Summe verläuft dabei über exponentiell (in der Anzahl der Zustände) viele Pfade, kann aber mit Methoden der dynamischen Programmierung auf ein Maß reduziert werden, daß durch $\mathcal{O}(T \cdot |S_q|)$ beschränkt ist (vgl. Abschnitt 3.3). Es bezeichne nun $[s_1^T]|w_1^N$ diese Menge aller Pfade des verketteten Automaten zur Wortfolge w_1,..., w_N, dann gilt für die akustische Wahrscheinlichkeit

$$\begin{aligned} Pr(x_1^T|w_1^N) &= \sum_{[s_1^T]|w_1^N} Pr(x_1^T, s_1^T) \\ &= \sum_{[s_1^T]|w_1^N} \prod_{t=1}^{T} Pr(x_t, s_t|x_1^{t-1}, s_1^{t-1}) \qquad \text{Bayes} \\ &= \sum_{[s_1^T]|w_1^N} \prod_{t=1}^{T} Pr(x_t|s_t) \cdot Pr(s_t|s_{t-1}) \qquad \text{HMM 1. Ordnung.} \end{aligned}$$

2.3 Sprachmodellierung

Eine weitere in den Entscheidungsprozeß einfließende Wissensquelle ist das Sprachmodell $Pr(w_1, ..., w_N)$. Diese Verteilung gibt die Wahrscheinlichkeit für das Auftreten der Satzhypothese w_1,..., w_N an. Sie ist unabängig von den akustischen Beobachtungen und wird daher *a-priori* geschätzt. Mit Hilfe der Dekompositionsregel läßt sich die Verteilung wie folgt faktorisieren:

$$Pr(w_1, ..., w_N) = \prod_{n=1}^{N} Pr(w_n|w_1, ..., w_{n-1}) \tag{2.8}$$

Die Vorgängerwörter w_1,..., w_{n-1} zu einem Wort w_n bezeichnet man auch als Historie von w_n. Im allgemeinen ist es nicht möglich, die gesamte Historie eines Wortes in der Modellierung der linguistischen Wahrscheinlichkeit zu berücksichtigen, da sich bereits bei einer Vokabulargröße von 20000 Wörtern und einer mittleren Satzlänge von 10 Worteinheiten 10^{48} mögliche Wortfolgen ergeben. Zur Vereinfachung kann die Historie daher auf die m unmittelbaren Vorgänger eingeschränkt werden:

$$Pr(w_n|w_1, ..., w_{n-1}) \approx Pr(w_n|w_{n-m}, ..., w_{n-1}) \tag{2.9}$$

Das resultierende Modell ist eine Markovkette m-ter Ordnung und wird üblicherweise als $(m+1)$-Gramm-Modell bezeichnet [Ney und Nießen 95]. Typische Modelle für großes Vokabular sind das Bigramm- ($m = 1$) bzw. das Trigramm-Sprachmodell ($m = 2$). Bei kleinem Vokabular verwendet man in der Regel ein Unigramm- ($m = 0$) oder ein Zerogramm-Sprachmodell. Beim Zerogramm-Sprachmodell wird jedem Wort des Vokabulars die gleiche a-priori Wahrscheinlichkeit zugeordnet. Da das Zerogramm-Sprachmodell in einem Erkennungsprozeß die Dekodierung zu vieler Worte pro Zeiteinheit einschränkt, wird es auch als „Wortstrafe“ (*word penalty*) bezeichnet.

2.4 Erkennung

Die Aufgabe eines Erkennungsprozesses besteht in der Dekodierung der gesprochenen Wortfolge aus den akustischen Beobachtungsvektoren einer lautsprachlichen Äußerung. Prinzipiell muß der Rechner hierzu die akustische Vektorfolge x_1^T systematisch auf alle erlaubten Zustandsfolgen s_1^T alignieren, die konsistent zu einer Wortfolge w_1^N sind (vgl. Abbildung 2.3). Die Suche verläuft zeitsynchron und kann mit Methoden der dynamischen Programmierung effizient realisiert werden [Ney und Aubert 96].

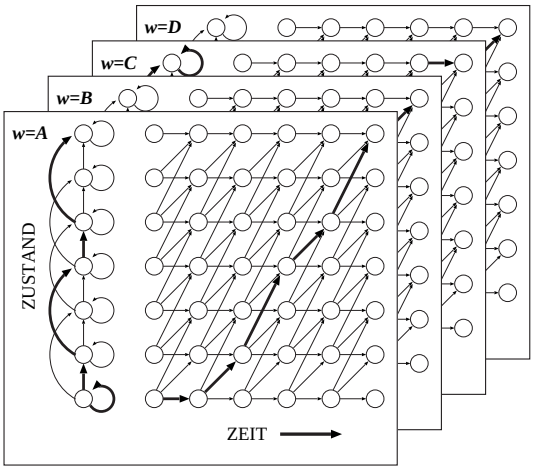


Abbildung 2.3: Parallele Dekodierung der Wörter {A, B, C, D}

Gemäß der Bayesschen Entscheidungsregel definiert sich die erkannte Wortfolge als diejenige Wortsequenz, die das Produkt aus Sprachmodellwahrscheinlichkeit und akustischer Wahrscheinlichkeit maximiert:

$$[w_1^N]_{\text{opt}} = \underset{[w_1^N]}{\text{argmax}} \left\{ Pr(w_1^N) \cdot \sum_{[s_1^T]|w_1^N} Pr\big(x_1^T, s_1^T | w_1^N\big) \right\} \tag{2.10}$$

Da die erkannte Wortfolge für die gegebenen Modellparameter unter allen Wortfolgen die größte a-posteriori Wahrscheinlichkeit besitzt, wird die Suche auch als *Maximum-A-Posteriori* (MAP)-Dekoder bezeichnet.

Zur Reduktion des Aufwands für die Berechnung der Emissionswahrscheinlichkeiten läßt sich die Summe über alle Pfade $[s_1^T]$ unter der Wortfolge w_1^N auch durch die Zustandsfolge mit der größten Emissionswahrscheinlichkeit approximieren:

$$[w_1^N]_{\text{opt}} = \underset{[w_1^N]}{\text{argmax}} \left\{ Pr(w_1^N) \cdot \max_{[s_1^T]|w_1^N} \left\{ Pr\big(x_1^T, s_1^T | w_1^N\big) \right\} \right\} \tag{2.11}$$

Dabei steht allerdings weniger eine gute Näherung der Summe im Vordergrund als vielmehr die Modellannahme, daß die Wortsequenz entlang der wahrscheinlichsten Zustandsfolge identisch zum Argumentmaximum aus Gleichung (2.10) ist. Die Maximum-Approximation auf Zustandsebene bezeichnet man auch als Viterbi-Näherung [Viterbi 67, Ney 90].

Um die Zahl der zu bewertenden Hypothesen einzuschränken, wird der Suchraum unter Verwendung sogenannter *Pruning*-Strategien reduziert. Die wichtigste Pruning-Technik bildet das *akustische Pruning* im Zusammenhang mit der *Strahlsuche*, bei der nur diejenigen Hypothesen im Suchprozeß „überleben", deren Bewertungen einen (durch die lokal beste Hypothese bestimmten) Schwellwert nicht überschreiten. Weniger aussichtsreiche Alternativen werden dagegen aus der Menge der aktiven Hypothesen entfernt. Da die Strahlsuche nicht in jedem Fall die exakte Lösung zu finden vermag, kann dies bei der Erkennung zu einer weiteren Art von Fehlern, den *Suchfehlern* führen.

Die Abbildung 2.4 zeigt die aktiven Zustände über die Zeitachse für die (fehlerfreie) Viterbi-Dekodierung der Ziffernfolge „*oh-nine-four-six-seven-five-four*" im akustischen Pruning. Das Pausen-Modell (*silence*) dient zum Erfassen möglicher Sprechpausen oder Nicht-Sprachanteile im akustischen Signal.

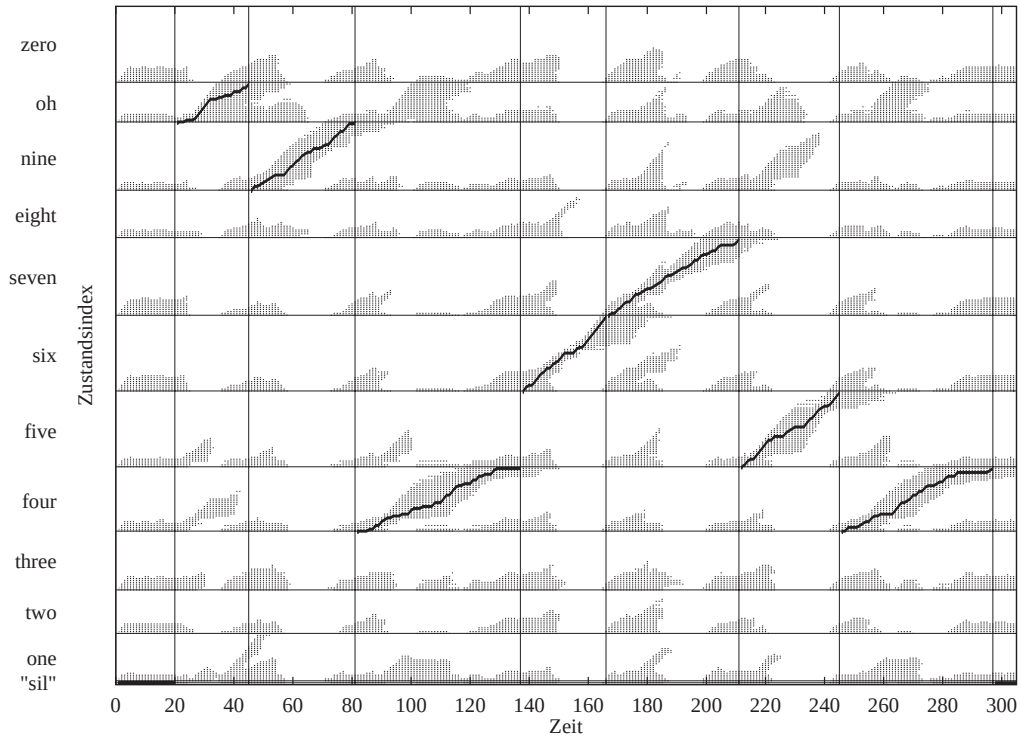


Abbildung 2.4: Suchraum in der Viterbi-Dekodierung für die Ziffernkette 0-9-4-6-7-5-4

Während des Suchprozesses werden für jede betrachtete Hypothese Rückwärtszeiger auf die jeweils beste Vorgängerhypothese angelegt. Die beste erkannte Wortfolge ergibt sich dann aus der Pfadbeschriftung entlang der Rückwärtszeiger zur besten endenden Hypothese. Um Unsicherheiten im akustischen Signal und bei der Bestimmung der Wortgrenzen zu verringern, können Alternativen zur wahrscheinlichsten Satzhypothese im Suchprozeß durch die Generierung von Wortgraphen erfaßt werden. Wortgraphen sind gerichtete, azyklische Graphen mit genau einer Quelle und einer Senke. Die Kanten des Graphen sind mit dem jeweiligen Worthypothesenindex und einer aus der Erkennung stammenden Bewertung beschriftet. Die Knoten geben die Start- bzw. Endzeit der jeweiligen Hypothese an.

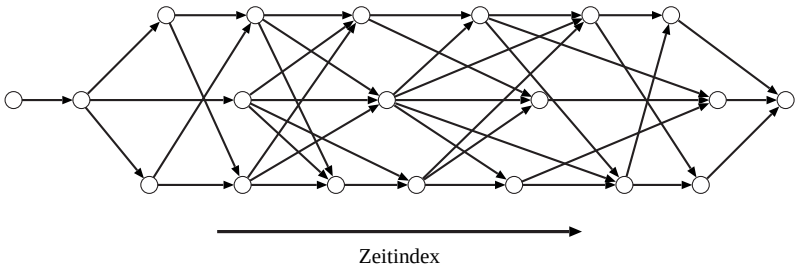


Abbildung 2.5: Worthypothesengraph

2.5 Konventionelles Maximum-Likelihood (ML) Training

Gegeben seien R Trainingsäußerungen die jeweils aus einer Folge X_r von akustischen Beobachtungsvektoren $x_{r,1}, x_{r,2},..., x_{r,T_r}$ und der zugehörigen Wortfolge $W_r = w_1,..., w_{N(r)}$ bestehen. Ziel eines konventionellen Trainings unter Verwendung des Maximum-Likelihood Schätzers (ML) ist es nun, den Parametersatz λ eines statistischen Klassifikators so zu schätzen, daß die Modelle die Trainingsäußerungen möglichst genau erfassen. Dies geschieht im ML-Ansatz durch Maximieren des Produkts der Emissionswahrscheinlichkeiten $Pr(X_r|W_r)$ über alle Trainingsäußerungen r:

$$f_{ML}(\lambda) = \prod_{r=1}^{R} Pr_\lambda(X_r|W_r) \tag{2.12}$$

Äquivalent hierzu ist die logarithmierte Variante:

$$F_{ML}(\lambda) = \sum_{r=1}^{R} \log Pr_\lambda(X_r|W_r) \tag{2.13}$$

Die Emissionsverteilungen werden im Rahmen dieser Arbeit durch multivariate Mischverteilungsdichten modelliert. Eine solche Mischverteilung wird als gewichtete Summe Gaußscher Unimodalverteilungen definiert. $L(s)$ gibt die Zahl der Mischungskomponenten an:

$$p(x|\theta_s) = \sum_{l=1}^{L(s)} c_{s,l}\; p(x|\theta_{s,l}) \tag{2.14}$$

Dabei gilt für die Einzelverteilungen

$$p(x|\theta_{s,l=\{\mu_{s,l},\,\Sigma_{s,l}\}}) = \frac{1}{\sqrt{\det(2\pi\Sigma_{s,l})}}\; \exp\Big(-\frac{1}{2}(x-\mu_{s,l})^\top\, \Sigma_{s,l}^{-1}\,(x-\mu_{s,l})\Big). \tag{2.15}$$

Hierbei bezeichnet $\mu_{s,l}$ den Mittelwert der Emissionsverteilung zur Mischungskomponente l des Zustands s und $\Sigma_{s,l}$ die zugehörige Kovarianzmatrix. $c_{s,l}$ ist das Mischverteilungsgewicht. Die Gewichte einer Mischung sind normiert, d.h. es gilt $\sum_l c_{s,l} = 1$. Damit setzt sich der Parametersatz λ für die akustische Modellierung wie folgt aus den Verteilungsparametern θ_s aller Mischverteilungen zusammen:

$$\lambda = \Big\{\theta_s \;:\; s \in \{s_1, \ldots, s_I\}\Big\} \tag{2.16}$$

Dabei gilt für die zustandsabhängigen Parametersätze $\theta_s = \{c_{s,l},\, \theta_{s,l} \;:\; l \in \{1,...,L(s)\}\}$.

Die Zielfunktion (2.13) kann nun unter Verwendung des *Expectation Maximization* (EM)-Algorithmus maximiert werden. Hierbei handelt es sich um ein einfaches, konsistentes Iterationsverfahren, mit dem neben den Mittelwerten und Varianzen auch die Mischverteilungsgewichte $c_{s,l}$ trainiert werden können. Speziell für die Anwendung bei Markovmodellen wird der EM-Algorithmus auch als *Baum-Welch-Training* bezeichnet [Baum und Eagon 67]. Ausgehend von einem Startwert λ_0 wird die Zielfunktion unter Verwendung der folgenden Hilfsfunktion optimiert [Ney und Eiden 95]:

$$Q(\lambda, \overline{\lambda}) = \sum_{r=1}^{R} \sum_{[s_1^{T_r}]|W_r} p_\lambda(s_1^{T_r}|x_{r,1}^{\ T_r}) \cdot \log p_{\overline{\lambda}}(x_{r,1}^{\ T_r}, s_1^{T_r}) \tag{2.17}$$

Der neue Parametersatz $\overline{\lambda}$ ergibt sich dabei durch Differentiation der Hilfsfunktion nach den Mischverteilungsparametern als Funktion von λ. Die innere Summe verläuft wiederum über alle möglichen Pfade des zur Wortfolge W_r gehörenden stochastischen Automaten. Jeder Pfad bildet wieder eine zeitliche Zuordnung der Beobachtungen $x_{r,t}$ auf Zustände des Automaten. Zur Vereinfachung kann die innere Summe auch durch denjenigen Pfad angenähert werden, der die größte Emissionswahrscheinlichkeit produziert. Diese Maximum-Approximation bezeichnet man auch als *Viterbi-Training*. Wie in Kapitel 3 gezeigt wird, folgen die Bestimmungsgleichungen für die neuen Mischverteilungsparameter im ML-Training auch für einen Spezialfall des diskriminativen Ansatzes. Aus diesem Grund wird an dieser Stelle auf eine Herleitung der Reestimationsgleichungen für das ML-Training verzichtet und auf den Abschnitt 3.7 verwiesen.

2.6 Diskriminativer Ansatz

Die ML-Zielfunktion aus Gleichung (2.13) erfaßt nur die Emissionswahrscheinlichkeit für die korrekte Transkription W_r einer Trainingsäußerung r. Unberücksichtigt bleibt dagegen, welche Emissionswahrscheinlichkeiten für Wortfolgen $W \neq W_r$ produziert werden. Da man jedoch in einem Erkennungsprozeß zu besseren Resultaten kommt, wenn die Emissionswahrscheinlichkeit der gesprochenen Wortfolge im Mittel auch unter allen konkurrierenden Wortfolgen maximal ist, sollte es das Ziel des Trainings sein, möglichst viele Trainingsäußerungen korrekt zu erkennen. Dies ist das Schlüsselprinzip diskriminativer Lernverfahren.

Im Unterschied zum ML-Ansatz berücksichtigen diskriminative Verfahren auch klassenfremde Trainingsdaten und konzentrieren sich somit verstärkt auf die Klassentrennung. Häufig verwendete Kriterien zum diskriminativen Training sind das *Minimum Classification Error* (MCE)-Kriterium [Ayer et al. 93, Chou et al. 92, Chou et al. 93, Paliwal et al. 95, Reichl und Ruske 95, Wolfertstetter 96], das die Satzfehlerrate auf den Trainingsdaten approximiert und das *Maximum Mutual Information* (MMI)-Kriterium [Kapadia et al. 93, Normandin und Morgera 91, Normandin 91, Normandin et al. 94, Normandin 96, Valtchev et al. 96], mit dem direkt die *a-posteriori* Wahrscheinlichkeiten der Trainingsäußerungen und damit die Klassentrennung optimiert werden. Das MMI-Kriterium ist äquivalent zum *Maximum A-Posteriori* (MAP)-Kriterium [Normandin 96].

Die MMI-Zielfunktion ist definiert als die Summe über die Logarithmen der *a-posteriori* Wahrscheinlichkeiten aller Trainingsäußerungen r:

$$\begin{aligned} F_{MMI}(\lambda) &= \sum_{r=1}^{R} \log p_\lambda(W_r|X_r) \\ &= \sum_{r=1}^{R} \log \frac{p_\lambda(X_r, W_r)}{p_\lambda(X_r)} \\ &= \sum_{r=1}^{R} \log \frac{p_\lambda(X_r|W_r)p(W_r)}{\sum_W p_\lambda(X_r|W)p(W)}. \end{aligned} \tag{2.18}$$

Eine Optimierung der Zielfunktion versucht nun zum einen, die Emissionswahrscheinlichkeit für jede tatsächlich gesprochene Wortfolge W_r zu maximieren. Zum anderen wird zugleich für jede Äußerung versucht, die Summe der Emissionswahrscheinlichkeiten über alle Konkurrenzäußerungen, gewichtet mit ihrer relativen Sprachmodellwahrscheinlichkeit, zu minimieren und so die Klassentrennung zu verbessern [Schlüter 96]. Da das Sprachmodell hierbei unmittelbar in das MMI-Kriterium eingeht, konzentriert sich das Training weniger auf jene akustischen Modelle, die durch das Sprachmodell leicht unterschieden werden können. Aus informationstheoretischer Sicht entspricht die Optimierung des MMI-Kriteriums einer Maximierung der *Transinformation* oder auch *wechselseitigen Information*. Mit der Definition für die Transinformation

$$\mathcal{I}_\lambda(X_r, W_r) = \log \left(\frac{p_\lambda(X_r|W_r)}{\sum_W p_\lambda(X_r|W)p(W)} \right)$$

ist die MMI-Zielfunktion als Kriterium äquivalent zur Maximierung der Transinformation:

$$F_{MMI}(\lambda) = \sum_{r=1}^{R} \mathcal{I}_\lambda(X_r, W_r) - \log p(W_r).$$

Die Transinformation mißt die Unbestimmtheit, mit der die korrekte Wortfolge W_r ihrer akustischen Realisierung X_r zugeordnet werden kann. Je kleiner diese Unbestimmtheit ist, desto größer ist die Transinformation und damit die Sicherheit, mit der die Zuordnung erfolgt. Eine Maximierung der Zielfunktion entspricht also zugleich einer Minimierung der Unbestimmtheit.

2.7 Stand der Forschung

In der Literatur findet sich bereits eine Fülle unterschiedlichster Arbeiten zum diskriminativen Training in der Spracherkennung. Die folgende Aufstellung gibt eine Übersicht über die Entwicklung und den Stand der Forschung.

MMI-Kriterium
Ein Vergleich zwischen dem ML-Verfahren und dem MMI-Kriterium wird erstmals von der IBM Spracherkennungs-Gruppe in [Bahl et al. 86] für ein sprecherabhängiges Isoliert-Wort-Erkennungssystem auf einem Vokabular von 2000 Wörtern beschrieben. Gegenüber einem konventionellen ML-Training wird eine Verbesserung der Fehlerrate von 18% relativ für das MMI-Kriterium berichtet. In [Merialdo 88] wird das MMI-Kriterium erfolgreich für die sprecherabhängige Phonem-Erkennung in kontinuierlich gesprochener Sprache verwendet. Die Maximierung der Zielfunktion erfolgt mit Hilfe eines Gradientenverfahrens. In [Chow 90] wird das MMI-Kriterium zum Training der Verteilungsparameter des auf HMM basierenden *BYBLOS*-Systems für großes Vokabular auf dem *DARPA Resource Management Speech*-Korpus (991 Wörter) eingesetzt. Die Menge der konkurrierenden Hypothesen wird hierbei durch eine 10-Besten Liste approximiert. Die Parameteroptimierung geschieht ebenfalls unter Verwendung eines Gradientenverfahrens. Zusätzlich werden Kodebuch-Exponenten zur Wichtung der Dimensionen des Merkmalsraumes trainiert. Der Verbesserung in der Wortfehlerrate von 11% relativ steht allerdings eine Verschlechterung der Satzfehlerrate von 3.6% relativ gegenüber. Eine Übertragung des erweiterten Baum Algorithmus [Gopalakrishnan et al. 91] für diskrete Verteilungen auf kontinuierliche Dichtefunktionen findet sich in [Normandin und Morgera 91]. Experimente auf dem TI/NIST Korpus zur Verbundziffernkettenerkennung erzielen eine Reduktion in der Satzfehlerrate von annähernd 50%. Untersuchungen auf dem TI-Digits Korpus für verschiedene Wortmodelle im Rahmen eines MMI-Trainings finden sich in [Cardin et al. 93]. Die berichtete Relativverbesserung der Wortfehlerrate liegt zwichen 5% und 10%. Experimente auf dem *Wall Street Journal* Korpus WSJ0 und WSJ0+ werden in [Valtchev et al. 96] durchgeführt. Im Unterschied zu den oben genannten Gruppen arbeitet das Training direkt auf Wortgraphen. Zur Reduktion des Rechenaufwandes wird der Suchraum in der Erkennungsphase ab der zweiten Iteration auf die Worthypothesen des aus der ersten Iteration generierten Wortgraphen eingeschränkt. Das MMI-Training erzielt hierbei eine Relativverbesserung der Wortfehlerrate zwischen 5% und 10%.

MCE-Kriterium
Eine Untersuchung über den Performanzgewinn für die Verbundziffernkettenerkennung mittels eines MCE-Trainings auf dem TI-Digits Korpus findet sich in [Chou et al. 92]. Gegenüber dem mit der konventionellen ML-Methode trainierten Referenzsystem wird eine Verbesserung in der Satzfehlerrate von ca. 8% relativ berichtet. In [Bauer 97] wird das MCE-Kriterium zum Trainieren Gaußscher Mischverteilungsdichten verwendet. Die Reestimation der Verteilungsparameter geschieht unter Verwendung eines Gradientenverfahrens. Experimente auf der *German Voice Mail* Datensammlung, einem deutschsprachigen Korpus für isoliert gesprochene Ziffernketten, erzielen hierbei eine Relativverbesserung von 48% für Einzel- und 26% für Mischverteilungen. Erste Ansätze für die Verwendung des MCE-Kriteriums zur diskriminativen Merkmalsextraktion finden sich in [Biem und Katagiri 93] sowie [Paliwal et al. 95].

Vergleich diskriminativer Kriterien
Ein Vergleich zwischen dem MMI- und dem MCE-Kriterium wird in [Reichl und Ruske 95] beschrieben. Ein Training zur sprecherunabhängigen Phonemerkennung unter Verwendung des erweiterten Baum-Algorithmus führt zu einer Relativverbesserung der Wortfehlerrate von 6% für das MMI-Kriterium und 13% für das MCE-Kriterium. In [Schlüter und Macherey 98] wird ein verallgemeinernder Ansatz für diskriminative Kriterien vorgestellt. Dieser Ansatz umfaßt sowohl daß MMI- als auch das MCE-Kriterium. Neben einer theoretischen Behandlung des MCE-Kriteriums finden sich hier zahlreiche Experimente zum MMI-Training auf Wortgraphen sowie der CT-Näherung für Einzel- und Mischverteilungen unter Verwendung gepoolter und zustandsabhängiger Varianzvektoren. Ebenfalls untersucht wird der Performanzgewinn durch den Einsatz einer linearen Diskriminanzanalyse.

Optimierungsverfahren
In [Kapadia et al. 93] werden verschiedene Optimierungsmethoden hinsichtlich ihres Konvergenzverhaltens im Training untersucht. Unter den Methoden befinden sich der erweiterte Baum-Algorithmus, sowie verschiedene Gradientenverfahren (konjugierter Gradient, steilster Abfall, Hesse-Methode). Die schnellste Konvergenz wird hier für den erweiterten Baum-Algorithmus berichtet. In [Schlüter et al. 97] werden für eine bestimmte Wahl der Schrittweiten im Gradientenverfahren formal nahezu identische Reestimationsgleichungen zu denen aus dem erweiterten Baum-Algorithmus hergeleitet (vgl. Abschnitt 3.5.2). Experimente auf dem TI-Digits Korpus und dem SieTill-Korpus bestätigen die formale Ähnlichkeit durch eine hohe Übereinstimmung für das Konvergenzverhalten im Training und für die erzielten Resultate auf den Testkorpora.

Lineare Transformationen und Kodebuch-Exponenten
In [Ayer 92, Ayer et al. 93] wird ein Verfahren zur Verbesserung der Erkennungsleistung mittels iterativ trainierter linearer Merkmalstransformationen beschrieben. Das Verfahren arbeitet diskriminativ und erzeugt eine Sequenz von linearen Abbildungen, deren Komposition eine *Ganzwort-adaptive* LDA ergeben. Experimente auf einem englisch-sprachigen Korpus für die sprecherunabhängige Erkennung der Buchstaben „B C D E G P T V“ erzielen eine Verbesserung in der Fehlerrate von annähernd 45% relativ. Nachteilig wirkt sich jedoch die hohe Anzahl benötigter Trainingsschritte aus. Beispielsweise erfordert das zitierte Resultat über 200 Iterationen. Veröffentlichungen, die den Einsatz sogenannter Kodebuch-Exponenten zur differenzierteren Wichtung der Dimensionen eines Merkmalsraumes behandeln, bilden die Artikel [Chow 90, Normandin und Mogera 91, Cardin et al. 92]. Gegenüber einer uniformen Wichtung der Dimensionen des Merkmalsraumes erzielen Experimente auf dem TI-Digits Korpus bei Verwendung der Kodebuch-Exponenten in [Normandin und Morgera 91] eine Relativverbesserung zwischen 21% für ein neu aufgesetztes ML-Training und weitere 11% relativ für ein anschließendes MMI-Training.

Diskriminatives Baum-Welch Training
In [Bridle 90] wird ein diskriminatives Baum-Welch Training als rekurrentes neuronales Netz interpretiert. Für die Lernphase wird ein *Back-Propagation-Algorithmus* vorgeschlagen, der die gleiche Form wie der rückläufige Zweig des Baum-Welch Algorithmus in einem ML-Training für Hidden-Markov-Modelle besitzt. Unter Verwendung eines speziellen Fehlerkriteriums sind die verwendeten Ableitungen von ähnlicher Gestalt wie in einem diskriminativen Baum-Welch Training für das MMI-Kriterium. Experimente zu dem beschrie-

benen Ansatz werden nicht durchgeführt, so daß hier keine Aussagen darüber getroffen werden können, ob mit der beschriebenen Methode Verbesserungen für die Spracherkennung zu erwarten sind.

2.8 Beitrag dieser Diplomarbeit

Die berichteten Relativverbesserungen zeigen, daß die Performanz von Spracherkennern, die auf der Basis des ML-Kriteriums trainiert wurden, in vielen Fällen mit Hilfe diskriminativer Methoden verbessert werden kann. Trotz dieser bereits beeindruckenden Resultate gibt es jedoch noch zahlreiche offene Fragen, von denen einige im Rahmen dieser Diplomarbeit beantwortet werden sollen. Die folgende Aufstellung faßt die bearbeiteten Problemschwerpunkte und ihre bisherige Behandlung in der Literatur zusammen.

- In dieser Arbeit wird ein allgemeiner Ansatz für diskriminative Kriterien vorgestellt. Auf der Grundlage dieses Ansatzes kann der Performanzgewinn unterschiedlichster diskriminativer Verfahren direkt miteinander verglichen werden. Ein solcher Vergleich wird für das MMI- und das MCE-Kriterium, sowie den hiermit verwandten Kriterien CT und FT durchgeführt.

- Bis auf [Valtchev et al. 96] verwenden sämtliche Forschungsgruppen eine Maximum-Approximation (*single best*) oder N-Besten-Listen, um eine Näherung für die Größe $p_\lambda(X_r)$ zu erhalten:

$$p_\lambda(X_r) = \sum_W p_\lambda(X_r, W) \approx \sum_{W \in \left\{ W_1,\ldots,W_N \mid \forall V \exists i \in \{1,\ldots,N\}: \; p_\lambda(X_r,V) \le p_\lambda(X_r,W_i) \right\}} p_\lambda(X_r, W)$$

  Innerhalb dieser Arbeit wird die in einem vorherigen Erkennungsschritt bestimmte Menge von konkurrierenden Hypothesen dagegen auf sämtliche in einem Wortgraphen enthaltene Satzhypothesen ausgedehnt. Dies führt nicht nur zu einer wesentlich zuverlässigeren Schätzung für $p_\lambda(X_r)$, sondern ist darüber hinaus auch weitaus effizienter als das Verfahren auf N-Besten-Listen. Untersucht wird ferner der Einfluß, den die beiden Näherungen „Maximum-Approximation“ und „Wortgraph“ auf die Performanz des Trainings haben. Für das MCE-Kriterium erfolgt der Einsatz von Wortgraphen in dieser Form sogar erstmalig in der Spracherkennung. Die speziell hierzu entwickelten Verfahren bilden einen wesentlichen Aspekt dieser Arbeit.

- Um das diskriminative Training in einer vertretbaren Anzahl an Iterationen durchführen zu können, müssen die Schrittweiten bei Verwendung diskreter Verteilungen in der Praxis wesentlich größer gewählt werden, als es der Konvergenzbeweis aus [Gopalakrishnan et al. 91] vorsieht. Werden statt dessen kontinuierliche Verteilungen verwendet, so ergeben sich bei Übertragung des Konvergenzbeweises sogar infinitesimal kleine Schrittweiten [Normandin 91]. Bei der Wahl der die Konvergenz steuernden Iterationskonstanten ist man daher (bislang) auf empirische Größen angewiesen. Werden die Schrittweiten jedoch zu groß gewählt, kann dies im Verlauf des Trainings zu einem drastischen Anstieg der Fehlerrate führen. Zu diesem Zweck wird eine Methode entwickelt, die die Schrittweite anhand der auf den Trainingsdaten ermittelten Fehlerrate adaptiert.

- Die Bestimmung der konkurrierenden Hypothesen erfolgt üblicherweise durch einen vorherigen Erkennungsschritt. Erkennungen sind jedoch sehr rechenintensiv und beanspruchen daher den überwiegenden Teil der im diskriminativen Training anfallenden Zeit. Um die Bestimmung der Menge der konkurrierenden Hypothesen zu beschleunigen, werden in dieser Diplomarbeit neue Verfahren zur Fokussierung der Suche entwickelt und getestet.

- In der Literatur werden der *erweiterte Baum* (EB)-Algorithmus und das Gradientenverfahren häufig als unterschiedliche Optimierungsmethoden behandelt. Für eine spezielle Wahl der Schrittweiten im Gradientenverfahren lassen sich für die Verteilungsparameter jedoch Reestimationsgleichungen bestimmen, die nahezu identisch zu denen des EB-Algorithmus sind. Die formale Übereinstimmung wird durch Experimente verifiziert.

- Bisher wurden die diskriminativen Verfahren in der Spracherkennung ausschließlich in der Viterbi-Näherung durchgeführt. Unbeantwortet ist bislang die Frage, ob die hierdurch erzielte Performanz nicht auch mit Hilfe des Baum-Welch Trainings für die konventionelle ML-Methode erreicht werden kann. Neben dieser Untersuchung werden für das MMI-Kriteriun auch erste Ergebnisse für ein diskriminatives Baum-Welch Training präsentiert.

- Ausgehend von der konventionellen linearen Diskriminanzanalyse (LDA) wird ein neues, von diskriminativen Kriterien abgeleitetes LDA-Kriterium hinsichtlich der Verwendbarkeit für die Spracherkennung untersucht.

- In [Normandin 96] wird eine Methode vorgestellt, um auch im diskriminativen Training Mischverteilungen zu splitten. Basierend auf dem Grundprinzip dieser Methode wird ein neues, verbessertes Verfahren entwickelt, das zur Auswahl der zu splittenden Verteilungen das MMI-Kriterium und zum Schätzen der Modellparameter das ML-Kriterium verwendet.

Kapitel 3

Diskriminatives Training

In diesem Kapitel erfährt das MMI-Kriterium aus Abschnitt 2.6 eine Erweiterung auf zusätzliche Kriterien, wie MCE, Gini und Varianten. Die Rechnungen entstammen zum Teil der lehrstuhlinternen Zusammenstellung [Schlüter 96] und bildeten auch die Grundlage der Arbeiten [Schlüter et al. 97] sowie [Schlüter und Macherey 98].

3.1 Allgemeiner Ansatz für diskriminative Kriterien

Gegeben seien wiederum R Trainingsäußerungen, die jeweils aus einer Folge X_r akustischer Beobachtungsvektoren $x_{r,1},...,x_{r,T_r}$ und der zugehörigen Wortfolge W_r bestehen. Ferner bezeichne $p_\lambda(W_r|X_r)$ die a-posteriori Wahrscheinlichkeit der Wortfolge W_r unter der Beobachtungsvektorfolge X_r und $p_\lambda(X_r|W)$ die Emissionswahrscheinlichkeit für die Beobachtung der akustischen Vektoren X_r bei gegebener Wortfolge W. Die Verbundwahrscheinlichkeit $p_\lambda(X_r, W)$ heißt im folgenden auch Satzwahrscheinlichkeit zur Wortsequenz W. Die Sprachmodellwahrscheinlichkeit $p(W)$ wird als gegeben vorausgesetzt.

Der Ansatz für das MMI-Kriterium kann nun wie folgt durch die Verwendung einer zusätzlichen Glättungsfunktion f und einem Exponenten $\alpha \in [1, \infty)$ zur Umwichtung der Satzwahrscheinlichkeiten erweitert werden:

$$F_D(\lambda) = \sum_{r=1}^{R} f\Big(\log \frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)} \Big). \tag{3.1}$$

Hierbei bezeichnet $\mathcal{M}_r$ die Menge der konkurrierenden Wortfolgen. Da die exakte Berechnung des Nenners im allgemeinen die Auswertung einer Summe über unendlich viele Wortfolgen W erfordert, wird die Menge $\mathcal{M}_r$ in der Praxis durch N-Besten Listen oder Wortgraphen angenähert. Die N-Besten Listen bzw. die Wortgraphen müssen hierzu in einem vorherigen Erkennungsschritt generiert werden.

Das verwendete diskriminative Kriterium wird nun im wesentlichen durch die spezielle Wahl der Glättungsfunktion f, dem Wichtungsexponenten α, sowie der Menge der konkurrierenden Wortfolgen $\mathcal{M}_r$ bestimmt. Die Glättungsfunktion f führt dabei zu einer Wichtung des Beitrags der gesamten Trainingsäußerung. Die Wichtung hängt ausschließlich von dem *diskriminativen Abstandsmaß* $\log \rho_\lambda(W_r|X_r)$ zwischen den gesprochenen und

den konkurrierenden Wortfolgen ab [Schlüter et al. 98]:

$$\log \rho_\lambda(W_r|X_r) = \log \frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)}. \tag{3.2}$$

Insbesondere ergibt sich für den Fall $\alpha = 1$ und $f = \mathrm{id}$ wiederum das MMI-Kriterium aus Gleichung (2.18). Werden zudem keine konkurrierenden Wortfolgen betrachtet ($\mathcal{M}_r = \emptyset$), erhält man wieder das konventionelle ML-Kriterium[1]. Tabelle 3.1 gibt eine Übersicht über die wichtigsten Kriterien, die in dem verallgemeinernden Ansatz enthalten sind.

Tabelle 3.1: Diskriminative Kriterien und Maximum-Likelihood Kriterium

Kriterium	Glättungs-funktion $f(z)$	Ableitung $f'(z)$	Wortfolgen in $\mathcal{M}_r$	Gewicht α
ML	id	1	$\emptyset$	–
MMI	id	1	alle konk. Wortfolgen (einschließlich W_r)	1
CT			beste erkannte Wortfolge	(∞)
MCE	$\frac{1}{1+e^{2\beta z}}$	$-\frac{\beta}{2}\frac{1}{\cosh^2(\beta z)}$	alle konk. Wortfolgen *ohne* W_r	≥ 1
FT			beste, inkorrekt erkannte Wortfolge	(∞)
Gini	$1 - e^z$	$-e^z$	alle konk. Wortfolgen (einschließlich W_r)	≥ 1

Während im MMI-Kriterium die Menge der konkurrierenden Wortfolgen alle potentiellen Wortfolgen - inklusive der gesprochenen - einschließt, wird im *Corrective Training* (CT) der Nenner des MMI-Kriteriums durch die beste erkannte Wortfolge approximiert. Formal geht das CT-Kriterium aus dem MMI-Kriterium hervor, wenn man für den Wichtungsexponenten α den Grenzübergang gegen ∞ bildet:

$$F_{CT}(\lambda) = \lim_{\alpha\to\infty} \sum_{r=1}^{R} \log \left(\frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)} \right)^{1/\alpha} = \sum_{r=1}^{R} \log \frac{p_\lambda(X_r|W_r)p(W_r)}{p_\lambda(X_r|W_{\text{best}})p(W_{\text{best}})}$$

Hierbei bezeichnet W_{best} die beste erkannte Wortfolge, d.h. diejenige Wortsequenz, die unter den gegebenen Verteilungsparametern λ die größte Satzwahrscheinlichkeit produziert.

[1] Für den Nenner in Ausdruck 3.1 ist dann eine geeignete positive Konstante $\neq 0$ zu wählen, da das Kriterium ansonsten nicht definiert ist.

Für das *Minimum Classification Error* (MCE)-Kriterium wird die logistische Funktion $1/(1+e^{2\beta z})$ zur Glättung herangezogen. Die Ableitung dieser Sigmoidfunktion bewirkt eine Herabwichtung des Beitrags all jener Trainingsäußerungen, für die das diskriminative Abstandsmaß aus Gleichung (3.2) Werte liefert, die entweder deutlich größer als null oder sehr viel kleiner als null sind. Dies ist nur bei all jenen Trainingsäußerungen der Fall, für die eine Erkennung unter den gegebenen akustischen Parametern entweder sehr sicher die gesprochene Wortfolge liefert ($\log \rho_\lambda(W_r|X_r) \gg 0$) oder bei denen die Bewertung der gesprochenen Wortfolge deutlich schlechter als die vieler inkorrekter Wortfolgen ist ($\log \rho_\lambda(W|X_r) \ll 0$). Letzteres kann zum Beispiel bei fehlerhaften Transkriptionen oder erschwerenden Randbedingungen (wie z.B. Rauschen, Störgeräusche etc.) auftreten. Es ist zu beachten, daß die Menge $\mathcal{M}_r$ der konkurrierenden Wortfolgen die gesprochene Wortfolge W_r ausschließt.

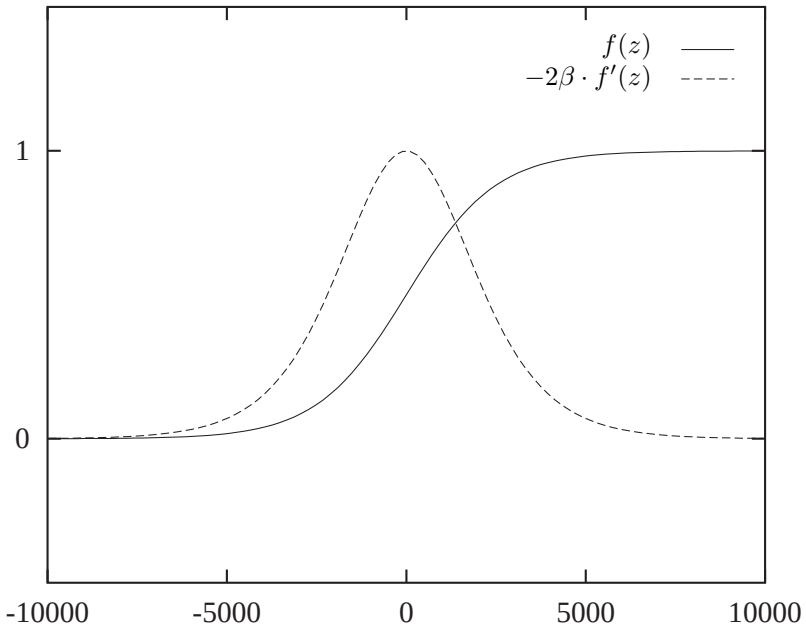


Abbildung 3.1: Verlauf der Sigmoidfunktion $f(z) = 1/(1+e^{2\beta z})$ und ihrer Ableitung

Das *falsifizierende Training*[2] (FT) geht – ähnlich wie das CT-Kriterium aus der MMI-Zielfunktion – für den Grenzübergang $\alpha \to \infty$ aus dem MCE-Kriterium hervor. Die Menge $\mathcal{M}_r$ entält dann nur die beste inkorrekt erkannte Wortfolge.

Für den Spezialfall $\alpha = 1$ und $\beta = 1/2$ ist das *Gini*-Kriterium im MCE-Kriterium enthalten. Das Gini-Kriterium und seine Verallgemeinerung im Fall $\alpha > 1$ sind aus theoretischer Sicht nur von untergeordneter Bedeutung und werden hier nicht weiter betrachtet.

[2]Der Begriff *falsifizieren* bedeutet, „eine Hypothese durch empirische Beobachtung zu widerlegen“. Die Bezeichnung bildet das Pendant zum *Corretive Training* für das MCE-Kriterium.

3.2 Optimierung der Zielfunktion

Zur Optimierung der diskriminativen Zielfunktion bzgl. der Parameter aus der akustischen Modellierung wird $F_D(\lambda)$ formal nach λ differenziert:

$$\frac{\partial F_D(\lambda)}{\partial \lambda} = \sum_{r=1}^{R} f'\left(\log \frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)} \right) \cdot \cdot \frac{\partial}{\partial \lambda} \left\{ \log \left[p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)\right] - \log \left[\sum_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W) \right] \right\}$$

Die Differentiation der Logarithmen der Emissionswahrscheinlichkeiten nach einem zustandsabhängigen Parameter $\theta_s \in \lambda$ ergibt unter Verwendung der in Abschnitt 2.2 beschriebenen Markov-Modellierung:

$$\frac{\partial \log p(x_1^T|W)}{\partial \theta_s} = \frac{1}{p(x_1^T|W)} \frac{\partial}{\partial \theta_s} \sum_{[s_1,\ldots,s_T]|W} \prod_{\tau=1}^{T} p(x_\tau|s_\tau)\, p(s_\tau|s_{\tau-1})$$

$$= \frac{1}{p(x_1^T|W)} \sum_{[s_1,\ldots,s_T]|W} \sum_{t=1}^{T} \left[\frac{\partial}{\partial p(x_t|s)} \prod_{\tau=1}^{T} p(x_\tau|s_\tau)\, p(s_\tau|s_{\tau-1}) \right] \frac{\partial p(x_t|s)}{\partial \theta_s}$$

Mit der Auswertung der partiellen Differentiation und einer nachfolgenden Umstellung der Terme erhält man:

$$= \frac{1}{p(x_1^T|W)} \sum_{t=1}^{T} \frac{\partial p(x_t|s)}{\partial \theta_s} \cdot \sum_{[s_1^{t-1},s_{t+1}^T]|W} \delta_{s,s_t} \left[p(s_t|s_{t-1}) \prod_{\tau\neq t,\tau=1}^{T} p(x_\tau|s_\tau) p(s_\tau|s_{\tau-1}) \right]$$

$$= \frac{1}{p(x_1^T|W)} \sum_{t=1}^{T} \frac{1}{p(x_t|s)} \frac{\partial p(x_t|s)}{\partial \theta_s} \cdot \sum_{[s_1^{t-1},s_t=s,s_{t+1}^T]|W} \left[\prod_{\tau=1}^{T} p(x_\tau|s_\tau) p(s_\tau|s_{\tau-1}) \right]$$

$$= \sum_{t=1}^{T} \frac{\sum_{[s_1^{t-1},s_{t+1}^T]|W} p(x_1^T, s_1^T, s_t = s)}{p(x_1^T|W)} \frac{\partial \log p(x_t|s)}{\partial \theta_s}$$

$$= \sum_{t=1}^{T} p(s_t = s|x_1^T, W) \frac{\partial \log p(x_t|s)}{\partial \theta_s}$$

Zur Vereinfachung der Schreibweise werden nun die beiden folgenden Hilfsgrößen (die sogenannten *Forward-Backward* (FB)-Wahrscheinlichkeiten) eingeführt:

$$\gamma_{r,t}(s|W) := p_\lambda(s_t = s|X_r, W)$$

$$\gamma_{r,t}(s) := \sum_{W\in\mathcal{M}_r} \frac{p_\lambda^\alpha(X_r|W)p^\alpha(W)}{\sum_{V\in\mathcal{M}_r} p_\lambda^\alpha(X_r|V)p^\alpha(V)} \gamma_{r,t}(s|W)$$

$$= p_\lambda(s_t = s|X_r) \qquad \text{falls } \alpha = 1.$$

Die Größe $\gamma_{r,t}(s|W)$ beschreibt die Wahrscheinlichkeit, daß ein Zeitanpassungspfad für die Wortfolge W bei gegebener Beobachtungsfolge X_r zum Zeitpunkt t durch den Zustand s führt. In ähnlicher Weise ist $\gamma_{r,t}(s)$ als eine Art gewichtete Wahrscheinlichkeit definiert, mit der zu einer gegebenen akustischen Vektorfolge X_r (unabhängig von einer Wortfolge) zum Zeitpunkt t der Zustand s hypothetisiert wird. Die Größe $\gamma_{r,t}(s)$ wird auch als *verallgemeinerte* oder *generalisierte* FB-Wahrscheinlichkeit bezeichnet. $\gamma_{r,t}(s|W)$ heißt im folgenden auch *bedingte* oder *spezielle* FB-Wahrscheinlichkeit. Die Methoden zur Berechnung der speziellen und verallgemeinerten FB-Wahrscheinlichkeiten sind Gegenstand des Abschnitts 3.3.

Wie in Abschnitt 2.5 beschrieben, werden die Emissionsverteilungen in dieser Arbeit durch Mischverteilungen modelliert:

$$p(x_{r,t}|s,\theta_s) = \sum_l c_{s,l}\, p(x_{r,t}|\theta_{s,l}),$$

Unter Verwendung der Ableitung des Logarithmus der Emissionswahrscheinlichkeit nach den Verteilungsparametern einer Mischungskomponente l

$$\frac{\partial \log p(x|\theta_s)}{\partial \theta_{s,l}} = \frac{c_{s,l}\, p(x|\theta s, l)}{\sum\limits_{\widetilde{l}} c_{s,\widetilde{l}} p(x|\theta_{s,\widetilde{l}})} \cdot \frac{\partial \log p(x|\theta_{s,l})}{\partial \theta_{s,l}} \tag{3.3}$$

folgt dann für die Differentiation der Zielfunktion $F_D(\lambda)$ nach einem zustandsabhängigen Parameter $\theta_{s,l} \in \lambda$:

$$\begin{aligned}\frac{\partial F_D(\lambda)}{\partial \theta_{s,l}} = \sum_{r=1}^{R} \alpha\, f'\!\left(\log \frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum\limits_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)} \right) \cdot \\ \cdot \sum_{t=1}^{T_r} \big[\gamma_{r,t}(s|W_r) - \gamma_{r,t}(s)\big] \frac{c_{s,l}\, p(x_{r,t}|\theta_{s,l})}{\sum\limits_{\widetilde{l}} c_{s,\widetilde{l}} p(x_{r,t}|\theta_{s,\widetilde{l}})} \frac{\partial}{\partial \theta_{s,l}} \log p(x_{r,t}|\theta_{s,l}).\end{aligned}$$

Mit Hilfe der Definition für die diskriminativen Erwartungswerte

$$\begin{aligned}\Gamma_{s,l}\big(g(x)\big) = \sum_{r=1}^{R} \alpha\, f'\!\left(\log \frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum\limits_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)} \right) \cdot \\ \cdot \sum_{t=1}^{T_r} \big[\gamma_{r,t}(s|W_r) - \gamma_{r,t}(s)\big] \frac{c_{s,l}\, p(x_{r,t}|\theta_{s,l})}{\sum\limits_{\widetilde{l}} c_{s,\widetilde{l}} p(x_{r,t}|\theta_{s,\widetilde{l}})} \cdot g(x_{r,t})\end{aligned} \tag{3.4}$$

kann der Ausdruck für die Ableitung der Zielfunktion schließlich wie folgt vereinfacht werden:

$$\frac{\partial F_D(\lambda)}{\partial \theta_{s,l}} = \Gamma_{s,l}\left(\frac{\partial}{\partial \theta_{s,l}} \log p(x|\theta_{s,l}) \right).$$

3.2.1 Maximum-Approximation für Mischverteilungen

Es ist nun möglich, unter Verwendung einer Maximum-Approximation für Mischverteilungen die Ableitung nach einem Parameter $\theta_{s,l}$ aus Gleichung (3.3) zu vereinfachen. Hierzu wird die Vorschrift für Mischverteilungen aus Gleichung (2.14) durch den folgenden Ausdruck ersetzt:

$$p(x|\theta_s) = p(x|\theta_{s,l}) \qquad \text{mit} \quad l := \underset{\widetilde{l}}{\operatorname{argmax}}\big\{\, c_{s,l} \;\cdot\; p(x|\theta_{s,\widetilde{l}}) \,\big\}$$

Die Maximum-Approximation für Mischverteilungen wird sehr häufig bei Gauß-Verteilungen eingesetzt, da der exponentielle Abfall dieser Funktion dazu führt, daß die Summe durch eine einzelne Dichte dominiert wird (vgl. hierzu Abbildung 3.2).

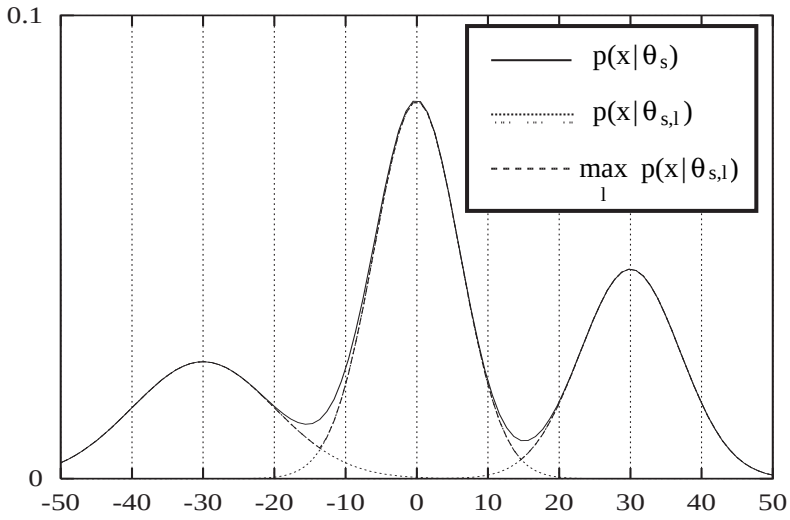


Abbildung 3.2: Maximum-Approximation bei Gaußschen Mischverteilungsdichten

Im folgenden wird mit $l(x,s)$ der Index der Mischungskomponente bezeichnet, die unter der Beobachtung x für den Zustand s die größte Wahrscheinlichkeit produziert:

$$l(x,s) = \underset{l}{\operatorname{argmax}}\big\{\, c_{s,l} \;\cdot\; p(x|\theta_{s,l}) \,\big\}.$$

Es gilt dann für die Ableitung der logarithmierten Emissionswahrscheinlichkeit nach einem zustandsabhängigen Parameter $\theta_{s,l}$:

$$\frac{\partial \log p(x|\theta_s)}{\partial \theta_{s,l}} \overset{\text{Max.-Approx.}}{=} \frac{\partial \log \big[c_{s,l} \cdot p(x|\theta_{s,l})\big]}{\partial \theta_{s,l}}.$$

Entsprechend folgt nun für die Definition der diskriminativen Erwartungswerte:

$$\Gamma_{s,l}\big(g(x)\big) \approx \sum_{r=1}^{R} \alpha\, f'\!\left(\log \frac{p_\lambda^\alpha(X_r|W_r)p^\alpha(W_r)}{\sum\limits_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r|W)p^\alpha(W)} \right) \cdot$$

$$\cdot \sum_{t=1}^{T_r} \big[\gamma_{r,t}(s|W_r) - \gamma_{r,t}(s)\big] \cdot \delta(l, l(x_{r,t}, s)) \cdot g(x_{r,t})$$

3.3 Berechnung der speziellen Forward-Backward (FB) Wahrscheinlichkeiten

Die FB-Wahrscheinlichkeiten geben die a-posteriori Wahrscheinlichkeiten einzelner Zustände zu definierten Zeitpunkten an. Sie ermöglichen hierdurch die Bestimmung der wahrscheinlichsten Zustandsfolge im Sinne der Bayes-Regel. Das Berechnen und die Akkumulation der FB-Wahrscheinlichkeiten bilden daher eine der zentralen Aufgaben des Trainings und der Erkennung. In diesem und dem nachfolgenden Abschnitt werden die Methoden zur Bestimmung der speziellen und verallgemeinerten FB-Wahrscheinlichkeiten beschrieben. Dabei unterscheidet man, ob die Bestimmung der Satzwahrscheinlichkeiten für die exakte Berechnung nach dem Baum-Welch (BW)-Algorithmus erfolgt oder gemäß dem Viterbi-Algorithmus in der Maximum-Approximation. Für den BW-Algorithmus gilt dabei stets die Einschränkung $\alpha = 1$, da der Wichtungsexponent eine geeignete Zerlegung der Summierung verhindert und somit nicht ohne weiteres angesetzt werden kann.

3.3.1 Baum-Welch Training

Grundsätzlich kann die Berechnung der speziellen FB-Wahrscheinlichkeiten $\gamma_{r,t}(s|W)$ auf die Berechnung der Größe

$$\gamma_{r,t}(\sigma, s|W) := p_\lambda(s_{t-1} = \sigma, s_t = s|X_r, W)$$

zurückgeführt werden. Die Größe $\gamma_{r,t}(\sigma, s|W)$ beschreibt die Wahrscheinlichkeit, daß ein Zeitanpassungspfad für die Wortfolge W unter der Beobachtungsfolge X_r zum Zeitpunkt $t-1$ durch den Zustand σ und zum Zeitpunkt t durch den Zustand s führt. Entsprechend ergibt sich $\gamma_{r,t}(s|W)$ dann aus $\gamma_{r,t}(\sigma, s|W)$ durch Summation über alle Vorgängerzustände σ:

$$\gamma_{r,t}(s|W) = \sum_\sigma \gamma_{r,t}(\sigma, s|W)$$

Für den Fall, daß ein Bakis-Modell verwendet wird, sind die Vorgängerzustände σ auf diejenigen Zustände eingeschränkt, von denen der Zustand s durch eine *loop*-, *forward*- oder *skip*-Transition erreicht werden kann.

Die Größe $\gamma_{r,t}(\sigma, s|W)$ wird im Baum-Welch Training durch Kombination der *Forward*- und *Backward*-Wahrscheinlichkeiten berechnet. $\gamma_{r,t}(\sigma, s|W)$ wird hierzu wie folgt in den Vorwärts- bzw. Rückwärts-Anteil zerlegt:

$$\begin{aligned}\gamma_{r,t}(\sigma, s|W) &= \frac{p_\lambda(X_r, s_{t-1} = \sigma, s_t = s|W)}{p_\lambda(X_r|W)} \\ &= \underbrace{\left[\sum_{[s_1^{t-2}, s_{t-1}=\sigma]|W} p_\lambda(x_{r,1}^{\;t-1}, s_1^{t-1})\right]}_{\substack{\textit{Forward}\text{-Wahrscheinlichkeit} \\ a_{r,t-1}(\sigma; W)}} \cdot \frac{p(s|\sigma)}{p_\lambda(X_r|W)} \cdot \underbrace{\left[\sum_{[s_t=s, s_{t+1}^{T_r}]|W} p_\lambda(x_{r,t}^{\;T_r}, s_t^{T_r})\right]}_{\substack{\textit{Backward}\text{-Wahrscheinlichkeit} \\ b_{r,t}(s; W)}}\end{aligned}$$

Die *Forward*-Wahrscheinlichkeit für den Zustand $s_{t-1} = \sigma$ ergibt sich dann aus der Summe aller Verbundwahrscheinlichkeiten $p_\lambda(x_{r,1}^{\;t-1}, s_1^{t-1})$, deren Pfade zum Zeitpunkt $t-1$ im

Zustand σ münden. Analog ist die *Backward*-Wahrscheinlichkeit für den Zustand $s_t = s$ als Summe über die Verbundwahrscheinlichkeiten $p_\lambda(x_{r,t}^{T_r}, s_t^{T_r})$ aller Pfade definiert, die zum Zeitpunkt t aus dem Zustand s herausführen (vgl. hierzu Abbildung 3.3).

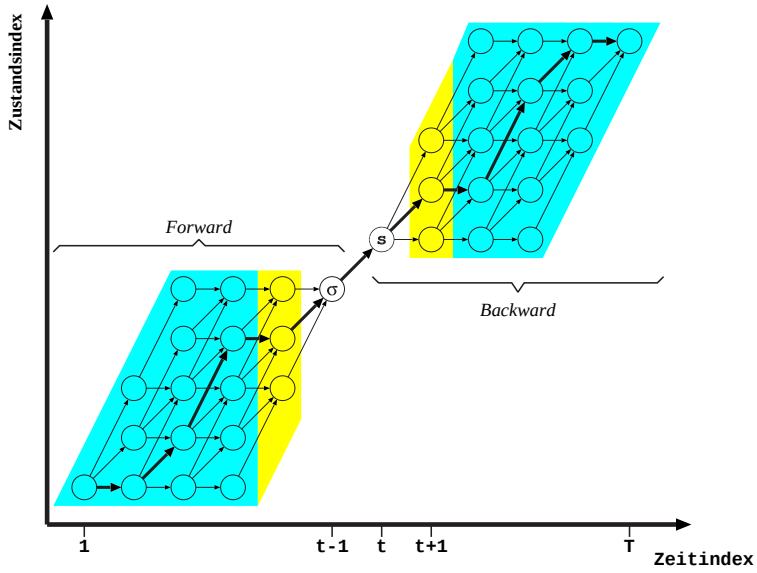


Abbildung 3.3: Prinzip zur Berechnung der bedingten FB-Wahrscheinlichkeit $\gamma_{r,t}(\sigma, s|W)$

Die direkte Auswertung der oberen Gleichung würde nun eine Summation über $3^{T-1}/(2 \cdot (T-1))$ Zustandsfolgen [Ney und Nießen 95] und damit $3^{T-1}/(T-1)$ Multiplikationen erfordern – ein Aufwand, der exponentiell mit der Dauer der Äußerung wächst. Eine weitaus effizientere Methode stützt sich auf die *rekursive* Definition der *Forward*- bzw. *Backward*-Wahrscheinlichkeit. Diese Methode ist linear bezüglich des Produkts aus der Dauer T einer Äußerung und der Zahl $|S_q|$ der Zustände der verketteten Wortautomaten, d.h. $\mathcal{O}(T \cdot |S_q|)$. Das Rechenschema zur Bestimmung der *Forward*-Wahrscheinlichkeit wird hierbei durch die folgende Rekursionsgleichung bestimmt:

$$a_{r,t}(s;W) = p_\lambda(x_{r,t}|s,W) \cdot \left(\sum_\sigma a_{r,t-1}(\sigma;W) \cdot p(s|\sigma) \right)$$

Dabei verläuft die Summe über alle Vorgängerzustände σ von s. Entsprechend ist die Rekursionsgleichung zur Bestimmung der *Backward*-Wahrscheinlichkeit definiert:

$$b_{r,t}(s;W) = \left(\sum_\sigma b_{r,t+1}(\sigma;W) \cdot p_\lambda(x_{r,t+1}|\sigma) \cdot p(\sigma|s) \right)$$

Im Unterschied zur vorherigen Gleichung verläuft die Summe hier über alle Zustände σ, die gemäß dem zugrundegelegten Transitionsmodell auf den Zustand s folgen können.

Die Berechnung der *Forward*-Wahrscheinlichkeit erfolgt nun durch spaltenweises Auffüllen der Elemente der Vorwärtsmatrix $[a_{r,t}(s;W)]$ in Richtung fortschreitender Zeit (siehe Abbildung 3.4).

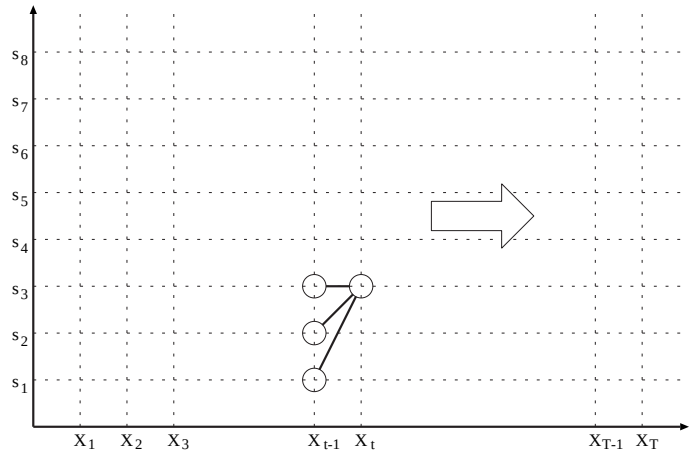


Abbildung 3.4: Rechenschema zur Bestimmung der Vorwärtsmatrix $[a_{r,t}(s;W)]$

In ähnlicher Weise ist die Rückwärtsmatrix $[b_{r,t}(s,W)]$ für die Berechnung der *Backward*-Wahrscheinlichkeiten zu bestimmen. Hier beginnt das spaltenweise Auffüllen der Matrix jedoch mit dem Zeitindex $t = T_r$ und verläuft dann rückwärts über die Zeitachse. Unter Verwendung der folgenden Identität für den Normierungsfaktor $1/p_\lambda(X_r|W)$

$$p_\lambda(X_r|W) \equiv a_{r,T_r}(s_{T_r};W) \equiv b_{r,0}(s_0;W)$$

genügen die bedingten FB-Wahrscheinlichkeiten der nachstehenden Beziehung:

$$\gamma_{r,t}(s|W) = a_{r,t}(s;W)b_{r,t}(s;W) \cdot \frac{1}{p_\lambda(X_r|W)},$$
$$\gamma_{r,t}(\sigma,s|W) = a_{r,t-1}(\sigma;W) \cdot p(x_{r,t}|s,W) \cdot p(s|\sigma) \cdot b_{r,t}(s;W) \cdot \frac{1}{p_\lambda(X_r|W)}.$$

Hierbei bezeichnet s_{T_r} den Endzustand zum Wortautomaten der Wortfolge W und s_0 den Anfangszustand des Automaten.

3.3.2 Viterbi-Training

In der Viterbi-Näherung wird zur Berechnung von $\gamma_{r,t}(s|W)$ statt der Summe über alle Pfade, die zum Zeitpunkt t durch den Zustand s führen allein der Zeitanpassungspfad mit der größten Satzwahrscheinlichkeit herangezogen. Die FB-Wahrscheinlichkeiten für die Zustände des *besten* Zeitanpassungspfades sind dann aufgrund der Normierungsbedingung für Wahrscheinlichkeiten gerade 1 und 0 für alle anderen Zustände, d.h.:

$$\gamma_{r,t}(s|W) \overset{\text{Viterbi}}{\approx} \delta(s, s_t(X_r,W)).$$

Hierbei bezeichnet $s_t(X_r,W)$ den Zustandsindex zum Zeitpunkt t des besten Zeitanpassungspfades für die Wortsequenz W unter der Folge X_r akustischer Beobachtungsvektoren. Da in der Viterbi-Näherung nur zu einem Pfad eine Satzwahrscheinlichkeit existiert, die ungleich null ist, reduziert sich der Aufwand für die Bestimmung des besten Zeitanpassungspfades auf die Berechnung der *Forward*-Wahrscheinlichkeit (d.h. das Füllen der Vorwärtsmatrix).

3.4 Diskriminatives Training unter Verwendung von Wortgraphen

Im diskriminativen Training wird jede Trainingsäußerung r gegen eine Menge $\mathcal{M}_r$ von konkurrierenden Hypothesen diskriminiert. Üblicherweise wird die Menge $\mathcal{M}_r$ aus den Satzhypothesen eines vorherigen Erkennungsschrittes gewonnen. Diese Vorgehensweise ist legitim, da die erkannten Wortfolgen unter den gegebenen akustischen Parametern die größten Satzwahrscheinlichkeiten produzieren und damit nahezu die gesamte Wahrscheinlichkeitsmasse der generalisierten FB-Wahrscheinlichkeitsdichteverteilungen auf sich vereinigen. In vielen Fällen konnte bereits eine Verbesserung der Erkennungsleistung gegenüber dem konventionellen ML-Ansatz beobachtet werden, wenn die Menge der konkurrierenden Hypothesen allein auf die beste (ggf. inkorrekt) erkannte Wortfolge[3] eingeschränkt wurde [Normandin und Mogera 91, Cardin et al. 96, Reichl und Ruske 95, Schlüter et al. 97, Schlüter und Macherey 98]. Es erweist sich jedoch als vorteilhaft, wenn zu jeder Äußerung mehrere konkurrierende Hypothesen betrachtet werden, da dies in der Regel zu einer schnelleren Konvergenz der Zielgröße $F_D(\lambda)$ führt.

Eine Möglichkeit, Alternativen zur besten erkannten Wortfolge in der Menge der konkurrierenden Hypothesen zu berücksichtigen, ergibt sich aus der Verwendung von N-Besten-Listen. N-Besten-Listen enthalten die Wortfolgen der N besten Satzhypothesen, d.h. diejenigen Wortsequenzen W, die zu einer Äußerung r die größten Satzwahrscheinlichkeiten produzieren. Für jeden Eintrag W_n der N-Besten-Liste muß dann eine Zeitanpassung, sowie eine Bestimmung der Größen $\gamma_{r,t}(s|W_n)$ (mit dem Gewicht $p_\lambda(X_r|W_n)/p_\lambda(X_r)$) durchgeführt werden.

Nachteile dieses Verfahrens sind sicherlich der hohe Speicherplatzverbrauch bei Verwendung sehr großer Listen und für den Fall der Viterbi-Näherung die damit häufig auftretenden Redundanzen in der Berechnung der FB-Wahrscheinlichkeiten. Unterscheiden sich die einzelnen Sazthypothesen nämlich nur in wenigen Wörtern oder Phonemen voneinander, dann gibt es oft übereinstimmende Teilsequenzen der Zustandsfolgen aus den jeweiligen Zeitanpassungen, für die jedoch das Gewicht $p_\lambda(X_r|W)/p_\lambda(X_r)$ immer wieder erneut zu berechnen ist (vgl. hierzu Abbildung 3.5). Insbesondere werden für sehr lange Sätze sehr große N-Besten-Listen erforderlich, um vergleichbare Anzahlen von Kokurrenzhypothesen pro gesprochenem Wort vorliegen zu haben.

Eine wesentlich effizientere und zugleich exaktere Methode zur Berechnung der generalisierten FB-Wahrscheinlichkeiten arbeitet auf Worthypothesengraphen und wird im folgenden für das Baum-Welch-Training und den Viterbi-Algorithmus vorgestellt.

3.4.1 Generalisierte FB-Wahrscheinlichkeiten im Baum-Welch Training

Im Falle $\alpha = 1$ geben die generalisierten FB-Wahrscheinlichkeiten die Wahrscheinlichkeit an, zu einem Zeitpunkt t überhaupt, daß heißt unabhängig von einer Wortfolge W, den Zustand s zu hypothetisieren:

$$\gamma_{r,t}(s) = p_\lambda(s_t = s|X_r) = \sum_{W \in \mathcal{M}_r} \frac{p_\lambda(X_r|W)p(W)}{p_\lambda(X_r)} \gamma_{r,t}(s|W)$$

[3] Zu diesen Verfahren gehören insbesondere das CT- und das FT-Kriterium.

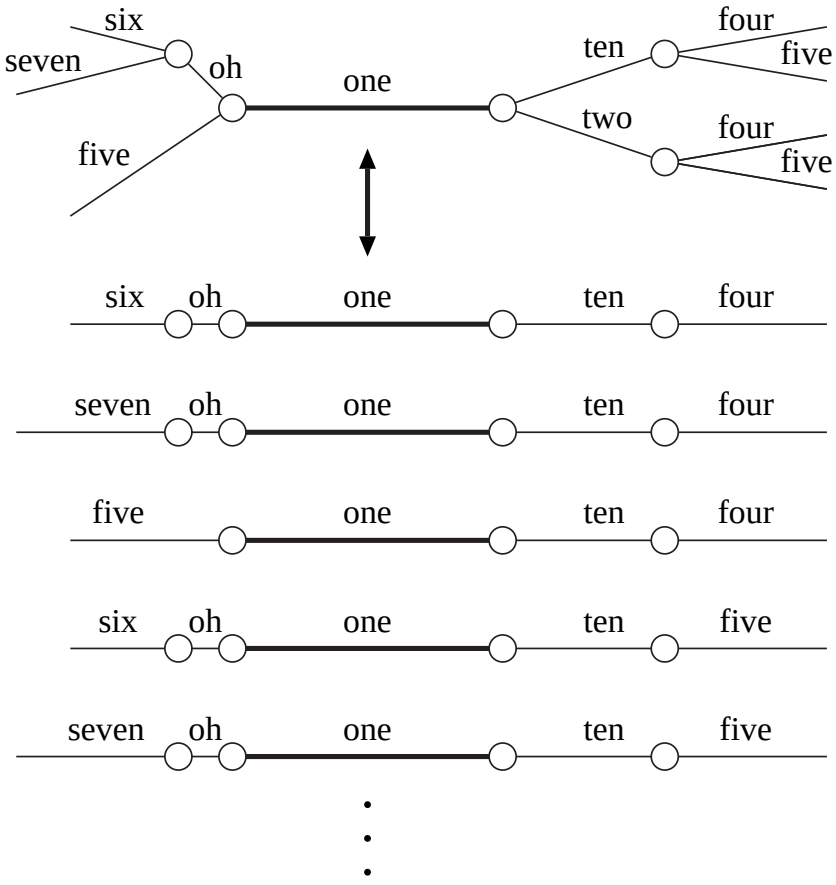


Abbildung 3.5: Ausschnitt eines Wortgraphen mit zentraler Hypothese **one** und den zugehörigen Wortfolgen aus der N-Besten Liste

Schränkt man die Wortfolgen W auf die in einem Wortgraphen enthaltenen Hypothesen ein, so ergibt sich $\gamma_{r,t}(s)$ als Summe über die wortbezogenen *Forward*- und *Backward*-Wahrscheinlichkeiten $a_{r,t}(s;w)$ bzw. $b_{r,t}(s;w)$:

$$\gamma_{r,t}(s) = \sum_{\{w|t_a(w)\leq t\leq t_e(w)\}} a_{r,t}(s;w)\cdot b_{r,t}(s;w)\cdot\frac{1}{p_\lambda(X_r)}$$

Die Worthypothesen w sind dabei auf jene Hypothesen eingeschränkt, die zum Zeitpunkt t im Wortgraphen existieren, d.h. all jene Wörter w, für deren Anfangszeit $t_a(w)$ und Endzeit $t_e(w)$ der Zeitpunkt t in das Intervall $[t_a(w), t_e(w)]$ fällt. Zusätzlich sind in der Rekursionsgleichung für die *Forward*-Wahrscheinlichkeit die Übergänge all jener Endzustände σ_e von Wörtern v zu berücksichtigen, die den Zustand s im Wort w erreichen können:

$$a_{r,t}(s;w) := p_\lambda(x_{r,t}|s,w)\cdot\Bigg(\sum_\sigma a_{r,t-1}(\sigma,w)\cdot p(s|\sigma) + \sum_v\sum_{\sigma_e} a_{r,t-1}(\sigma_e;v)\cdot p(s|\sigma)\cdot p(w|v)\Bigg)$$

Entsprechend ist die Rekursionsgleichung für die *Backward*-Wahrscheinlichkeit zu erweitern:

$$b_{r,t}(s;w) := \Bigg(\sum_{\sigma} b_{r,t+1}(\sigma;w) \cdot p_\lambda(x_{r,t+1}|\sigma) \cdot p(\sigma|s) + \sum_{u} \sum_{\sigma_a} b_{r,t+1}(\sigma_a;u) \cdot p_\lambda(x_{r,t+1}|\sigma_a) \cdot p(u|w) \Bigg)$$

Die Summe im zweiten Term verläuft hier über all jene Wörter u, deren Anfangszustände σ_a vom Zustand s des Wortes w erreicht werden können.

3.4.2 Worthypothesen-Wahrscheinlichkeiten im Viterbi-Training

Unter Berücksichtigung eines Wichtungsexponenten $\alpha \in [1, \infty)$ gilt für die generalisierten FB-Wahrscheinlichkeiten

$$\gamma_{r,t}(s) = \sum_{W \in \mathcal{M}_r} \frac{p_\lambda^\alpha(X_r, W)}{\sum_{V \in \mathcal{M}_r} p_\lambda^\alpha(X_r, V)} p(s_t = s|X_r, W). \tag{3.5}$$

In der Viterbi-Näherung können nun für jede Worthypothese w eines Wortgraphen eindeutige Anfangs- und Endzeiten $t_a(w)$ bzw. $t_e(w)$ angegeben werden, die durch einen vorab erfolgten Erkennungsschritt bestimmt sind. Da hiermit die Berechnung der Zeitanpassung für jedes Wort w auch einzeln durchgeführt werden kann, ist es möglich, die Summe über alle Wortfolgen W aus Gleichung (3.5) in die Folgen $W = (W_a(w), w, W_e(w))$ aufzuspalten. Hierbei bezeichnen $W_a(w)$ und $W_e(w)$ diejenigen Worthypothesenfolgen, die w vorausgehen bzw. die auf w folgen. Die Abhängigkeiten der Anfangs- und Endzeiten $t_a(w)$ bzw. $t_e(w)$, sowie der Wortfolgen $W_a(w)$ bzw. $W_e(w)$ werden zur Vereinfachung der Schreibweise im folgenden ausgelassen, d.h. $t_a \equiv t_a(w)$, $t_e \equiv t_e(w)$, $W_a \equiv W_a(w)$ und $W_e \equiv W_e(w)$. Wird mit $p_{[t_a,t_e]}(w|X_r)$ die Wahrscheinlichkeit definiert, daß ein Wort w unter der gegebenen Folge akustischer Vektoren X_r im Zeitintervall $[t_a, t_e]$ hypothetisiert wird,

$$p_{[t_a,t_e]}(w|X_r) := \sum_{W_a, W_e} \frac{p_\lambda^\alpha(X_r, W_a, w, W_e)}{\sum_{V \in \mathcal{M}_r} p_\lambda^\alpha(X_r, V)},$$

dann folgt für die generalisierten FB-Wahrscheinlichkeiten

$$\gamma_{r,t}(s) = \sum_{w|t_a \le t \le t_e} p_\lambda(s_t = s|w, x_{r\,t_a}^{t_e}) \cdot p_{[t_a,t_e]}(w|X_r).$$

Die Größe $p_{[t_a,t_e]}(w|X_r)$ wird auch als *Wortwahrscheinlichkeit* bezeichnet. Für den Fall $\alpha = 1$ ist die Wortwahrscheinlichkeit identisch zur a-posteriori Wahrscheinlichkeit für die Hypothetisierung des Wortes w im Zeitintervall $[t_a, t_e]$ – unabhängig von Kontexten W_a, W_e.

Unter Berücksichtigung eines m-Gramm-Sprachmodells kann die Bestimmung der Wortwahrscheinlichkeiten nun effizient über einen (für Wortgraphen modifizierten) FB-Algorithmus erfolgen. Hierzu betrachten wir die Sprachmodellwahrscheinlichkeiten $p(w|h)$, wobei $h = (h_1,...,h_{m-1})$ die Historie des Wortes w bezeichnet. Da es verschiedene Worthypothesenfolgen W_a mit identischem Suffix h geben kann, wird die Historie im folgenden

als Äquivalenzklasse definiert. Für zwei Worthypothesenfolgen W_a und V_a sei daher

$$W_a \sim_F^m V_a \qquad \text{gdw} \qquad \begin{aligned} W_a &= w_1, ..., w_i, h_1, ..., h_{m-1} \quad \text{und} \\ V_a &= v_1, ..., v_j, h_1, ..., h_{m-1}. \end{aligned}$$

Die Äquivalenzklassen werden im folgenden mit $[h]_{\sim_F^m}$ bezeichnet. Damit läßt sich nun die *Forward*-Wortwahrscheinlichkeit $\Phi_{r,t}(h)$ als Wahrscheinlichkeit definieren, daß die letzten $m-1$ Worthypothesen einer zum Zeitpunkt t endenden Wortsequenz W_a der Folge h entsprechen:

$$\Phi_{r,t}(h) = \sum_{W_a \in [h]_{\sim_F^m}} p_\lambda^\alpha(x_{r1}^t | W_a) \cdot p^\alpha(W_a).$$

Da eine Wortsequenz W_a auch weniger als m Hypothesen umfassen kann, muß die Berechnung der Verbundwahrscheinlichkeit $p(W_a)$ gegebenenfalls auf ein einfacheres Sprachmodell zurückgeführt werden. Die folgende Berechnungsvorschrift beschreibt den allgemeinen Fall für eine Wortsequenz $W_a = (a_1, ..., a_N)$:

$$p(W_a) = p(a_1) \cdot \prod_{i=2}^{m-1} p(a_i | a_1^{i-1}) \cdot \prod_{i=m}^{N} p(a_i | a_{i-m+1}^{i-1})$$

Die *Forward*-Wortwahrscheinlichkeit läßt sich nun mittels dynamischer Programmierung bestimmen, indem die Wortkanten eines Wortgraphen bezüglich ihrer Anfangszeiten sukzessive in aufsteigender Reihenfolge abgearbeitet werden:

$$\Phi_{r,t_e}(h_2^{m-1}, w) = p_\lambda^\alpha(x_{r t_a}^{t_e} | w) \cdot \sum_{h_1} \Phi_{r,t_a-1}(h_1, h_2^{m-1}) \cdot p^\alpha(w | h_1, h_2^{m-1})$$

Die Endzeiten $t_a - 1$ der Vorgängerhypothesen h_{m-1} ergeben sich dabei aus der Startzeit t_a der Hypothese w. Analog zur *Forward*-Wortwahrscheinlichkeit wird nun auf ähnliche Weise die *Backward*-Wortwahrscheinlichkeit $\Psi_{r,t}(f)$ definiert. Diese beschreibt die Wahrscheinlichkeit, daß die ersten $m-1$ Wörter einer zum Zeitpunkt t beginnenden Worthypothesensequenz der Folge $f = (f_1, ..., f_{m-1})$ entsprechen. Auch f wird als Äquivalenzklasse definiert:

$$W_e \sim_B^m V_e \qquad \text{gdw} \qquad \begin{aligned} W_e &= f_1, ..., f_{m-1}, w_m, ..., w_{m+i} \quad \text{und} \\ V_e &= f_1, ..., f_{m-1}, v_m, ..., v_{m+j}. \end{aligned}$$

Für die *Backward*-Wortwahrscheinlichkeit gilt damit

$$\Psi_{r,t}(f) = \sum_{W_e \in [f]_{\sim_B^m}} p_\lambda^\alpha(x_{r t}^{T_r} | W_e) \cdot p^\alpha(W_e).$$

Hierbei ergibt sich jedoch das folgende Problem: Da die Vorgängerworthypothesen zu W_e an dieser Stelle nicht bekannt sind, müßte das Sprachmodell für kürzere Wortfolgen W_e ähnlich wie oben auf einfachere Sprachmodelle zurückgeführt werden. Andererseits ergibt sich aus der Forderung, daß die *Forward*-Wortwahrscheinlichkeit zum Zeitpunkt $t = T_r$ für alle aus dem Wortgraphen herausführenden Kanten gleich der *Backward*-Wahrscheinlichkeit der zum Zeitpunkt $t = 1$ startenden Kanten ist, die Notwendigkeit, daß die gesamte Historie für f im Sprachmodell berücksichtigt werden muß. Deshalb wird die

Definition der *Backward*-Wortwahrscheinlichkeit wie folgt modifiziert, wobei der fehlende Anteil für das Sprachmodell erst in Gleichung (3.6) für die Berechnung der Wortwahrscheinlichkeit berücksichtigt wird:

$$\widetilde{\Psi}_{r,t}(f) = \sum_{W_e \in [f]_{\sim_B^m}} p_\lambda^\alpha(x_t^{T_r}|W_e) \cdot \prod_{i=m}^{M} p^\alpha(e_i|e_{i-m+1}^{i-1}).$$

Dabei gilt $W_e = (e_1,...,e_M)$. Auch hier läßt sich eine Gleichung zur Berechnung der *Backward*-Wortwahrscheinlichkeit mittels dynamischer Programmierung angegeben:

$$\widetilde{\Psi}_{r,t_a}(w, f_1^{m-2}) = p_\lambda^\alpha(x_{r\,t_a}^{t_e}|w) \cdot \sum_{f_{m-1}} \widetilde{\Psi}_{r,t_{e+1}}(f_1^{m-2}, f_{m-1}) \cdot p^\alpha(f_{m-1}|w, f_1^{m-2})$$

Die Hypothesen des Wortgraphen werden dann bezüglich ihrer Endzeiten in absteigender Reihenfolge abgearbeitet. Hierzu muß der Wortgraph zunächst nach Endzeiten sortiert werden. Ein entsprechendes Sortierverfahren, das im Mittel mit linearer Zeitkomplexität $\mathcal{O}(T_r)$ auskommt, wird in Kapitel 4 vorgestellt.

Mit den Definitionen für die *Forward*- und *Backward*-Wortwahrscheinlichkeiten ergibt sich nun die folgende Vorschrift für die Berechnung der Wortwahrscheinlichkeiten:

$$p_{[t_a,t_e]}(w|X_r) = \sum_{h_2^{m-1}} \sum_{f_1^{m-2}} \frac{\Phi_{r,t_e}(h_2^{m-1}, w) \cdot \widetilde{\Psi}_{r,t_a}(w, f_1^{m-2})}{\overline{p}_\lambda(X_r) \cdot p_\lambda^\alpha(x_{t_a}^{t_e}|w)} \cdot \prod_{i=1}^{m-2} p^\alpha(f_i|h_{i+1}^{m-1}, w, f_1^{i-1}) \tag{3.6}$$

Dabei gilt für den Normierungsfaktor $\overline{p}(X_r)$:

$$\begin{aligned}\overline{p}(X_r) &= \sum_{h_2^{m-1}} \sum_{w} \Phi_{r,T_r}(h_2^{m-1}, w) \\ &= \sum_{f_1^{m-2}} \sum_{w} \widetilde{\Psi}_{r,1}(w, f_1^{m-2}) \cdot \left[\prod_{i=1}^{m-2} p^\alpha(f_i|w, f_1^{i-1})\right] \cdot \left(p^\alpha(w)\right)^{\operatorname{sgn}\lfloor \frac{m}{2} \rfloor}\end{aligned}$$

Aus den allgemeinen Vorschriften ergeben sich nun für die Fälle eines Zero- bzw. Unigramm-Sprachmodells die folgenden Gleichungen:

Zero- und Unigramm-Sprachmodell:

$$\begin{aligned}\Phi_{r,t_e}(w) &= p_\lambda^\alpha(x_{r\,t_a}^{t_e}|w) \cdot p^\alpha(w) \cdot \sum_v \Phi_{r,t_a-1}(v) \\ \Psi_{r,t_a}(w) &= p_\lambda^\alpha(x_{r\,t_a}^{t_e}|w) \cdot \sum_u \Psi_{r,t_e+1}(u) \cdot p^\alpha(u) \\ p_{[t_a,t_e]}(w|X_r) &= \frac{\Phi_{r,t_e}(w) \cdot \Psi_{r,t_a}(w)}{\overline{p}(X_r) \cdot p_\lambda^\alpha(x_{r\,t_a}^{t_e})} \\ \overline{p}(X_r) &= \sum_v \Phi_{r,T_r}(v) = \sum_u \Psi_{r,1}(u) \cdot p^\alpha(u)\end{aligned}$$

3.5 Optimierung der Mischverteilungsparameter

Für die Reestimation der Mischverteilungsparameter stehen im diskriminativen Training grundsätzlich zwei Optimierungsmethoden zur Verfügung:

- der *erweiterte Baum* (EB)-Algorithmus und
- ein *Gradientenverfahren* (GD).

Der EB-Algorithmus geht auf [Gopalakrishnan et al. 91] zurück. Er verallgemeinert die „Baum-Eagon Ungleichung für homogene Polynome“ [Baum und Eagon 67] auf rationale Funktionen. Dadurch ist es möglich, auch komplexere Funktionen, wie z.B. die im diskriminativen Training verwendeten Zielgrößen, iterativ zu optimieren. In [Normandin und Morgera 91] erfährt der ursprünglich für diskrete Verteilungen entworfene Algorithmus eine Erweiterung auf kontinuierliche Verteilungsdichten. Im folgenden wird der EB-Algorithmus für den hier verwendeten allgemeinen diskriminativen Ansatz vorgestellt und in Abschnitt 3.5.2 formal mit einem Gradientenverfahren verglichen.

3.5.1 Erweiterter Baum Algorithmus

Die Optimierung der diskriminativen Zielfunktion geschieht im erweiterten Baum-Algorithmus unter Verwendung der folgenden Hilfsfunktion:

$$\begin{aligned} S(\lambda,\overline{\lambda}) = \sum_{s}\sum_{r=1}^{R} f'\Big(\frac{p_\lambda^\alpha(X_r,W_r)}{\sum\limits_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r,W)}\Big)\cdot\sum_{t=1}^{T_r}\left[\gamma_{r,t}(s|W_r)-\gamma_{r,t}(s)\right]\alpha\log p(x_{r,t}|\overline{\theta}_s) \\ + \sum_{s} D_s\cdot\int\Big\{p(x|\theta_s)\cdot\alpha\cdot\log p(x|\overline{\theta}_s)\Big\}\,dx. \end{aligned} \tag{3.7}$$

Bei den Größen D_s handelt es sich um zustandsabhängige Iterationskonstanten, mit denen die Geschwindigkeit der Konvergenz gesteuert werden kann. Ein Konvergenzbeweis zur Gleichung (3.7) existiert bislang nur für den Fall des MMI-Kriteriums ($\alpha=1$ und $f(z)=z$) bei Verwendung diskreter Verteilungen [Gopalakrishnan et al. 91]. Wird der Konvergenzbeweis auf kontinuierliche Verteilungen übertragen, gehen die ohnehin schon unpraktikabel kleinen Schrittweiten sogar gegen null [Normandin 91], weshalb unter realen Bedingungen auf eine „echte“ Konvergenz verzichtet werden muß. Für das damit verbleibende Problem der Schrittweitenbestimmung ist eine optimale Lösung bislang noch nicht bekannt.

Der Wichtungsexponent α im zweiten Term der Gleichung (3.7) kann nun zur Vereinfachung auch in die Iterationskonstanten D_s integriert werden. Für die Ableitung der Hilfsfunktion nach den neuen Parametern $\overline{\theta}_{s,l}$ einer Mischungskomponente ergibt sich dann:

$$\begin{aligned} \frac{\partial S(\lambda,\overline{\lambda})}{\partial\overline{\theta}_{s,l}} = \sum_{r=1}^{R} f'\Big(\frac{p_\lambda^\alpha(X_r,W_r)}{\sum\limits_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r,W)}\Big)\cdot\sum_{t=1}^{T_r}\left[\gamma_{r,t}(s|W_r)-\gamma_{r,t}(s)\right] \\ \cdot\alpha\cdot\frac{\overline{c}_{s,l}\,p(x_{r,t}|\overline{\theta}_{s,l})}{\sum\limits_{\widetilde{l}}\overline{c}_{s,\widetilde{l}}\,p(x_{r,t}|\overline{\theta}_{s,\widetilde{l}})}\cdot\frac{\partial\log p(x_{r,t}|\overline{\theta}_{s,l})}{\partial\overline{\theta}_{s,l}} \\ + D_s\cdot\int\left\{p(x|\theta_s)\cdot\frac{\overline{c}_{s,l}\,p(x|\overline{\theta}_{s,l})}{\sum\limits_{\widetilde{l}}\overline{c}_{s,\widetilde{l}}\,p(x|\overline{\theta}_{s,\widetilde{l}})}\cdot\frac{\partial\log p(x|\overline{\theta}_{s,l})}{\partial\overline{\theta}_{s,l}}\right\}dx \end{aligned}$$

In ähnlicher Weise folgt für die Optimierung bezüglich der Mischverteilungsgewichte $\overline{c}_{s,l}$:

$$\begin{aligned}\frac{\partial S(\lambda,\overline{\lambda})}{\partial\,\overline{c}_{s,l}} = \sum_{r=1}^{R} f'\Big(\frac{p_\lambda^\alpha(X_r,W_r)}{\sum\limits_{W\in\mathcal{M}_r} p_\lambda^\alpha(X_r,W)}\Big)\cdot\sum_{t=1}^{T_r}\left[\gamma_{r,t}(s|W_r)-\gamma_{r,t}(s)\right]\\ \cdot\,\alpha\cdot\frac{\overline{c}_{s,l}\,p(x_{r,t}|\overline{\theta}_{s,l})}{\sum\limits_{\widetilde{l}}\overline{c}_{s,\widetilde{l}}\,p(x_{r,t}|\overline{\theta}_{s,\widetilde{l}})}\cdot\frac{1}{\overline{c}_{s,l}}\\ +\,D_s\cdot\int\left\{p(x|\theta_s)\cdot\frac{\overline{c}_{s,l}\,p(x|\overline{\theta}_{s,l})}{\sum\limits_{\widetilde{l}}\overline{c}_{s,\widetilde{l}}\,p(x|\overline{\theta}_{s,\widetilde{l}})}\cdot\frac{1}{\overline{c}_{s,l}}\right\}dx\ ,\end{aligned}$$

wobei die folgende Ableitung für Emissionswahrscheinlichkeiten nach einem Mischverteilungsgewicht $\overline{c}_{s,l}$ verwendet wurde:

$$\frac{\partial\log p(x|\overline{\theta}_s)}{\partial\,\overline{c}_{s,l}} = \frac{\overline{c}_{s,l}\,p(x|\overline{\theta}_{s,l})}{\sum\limits_{\widetilde{l}}\overline{c}_{s,\widetilde{l}}\,p(x_{r,t}|\overline{\theta}_{s,\widetilde{l}})}\cdot\frac{1}{\overline{c}_{s,l}} \tag{3.8}$$

Zur Vereinfachung der Bestimmungsgleichungen für die neuen Verteilungsparameter wird im folgenden eine Näherung für die relativen Gewichte verwendet

$$\frac{\overline{c}_{s,l}\,p(x|\overline{\theta}_{s,l})}{\sum\limits_{\widetilde{l}}\overline{c}_{s,\widetilde{l}}\,p(x|\overline{\theta}_{s,\widetilde{l}})} \approx \frac{c_{s,l}\,p(x|\theta_{s,l})}{\sum\limits_{\widetilde{l}}c_{s,\widetilde{l}}\,p(x|\theta_{s,\widetilde{l}})}\ , \tag{3.8$'$}$$

d.h. für diesen Term werden die neuen durch die alten Parameter ersetzt. Verwendet man dagegen die in Abschnitt 3.2 beschriebene Maximum-Approximation für Mischverteilungen, so vereinfacht sich Gleichung (3.8) zu

$$\frac{\partial\log p(x|\overline{\theta}_s)}{\partial\,\overline{c}_{s,l}} = \frac{1}{\overline{c}_{s,l}}. \tag{3.8$''$}$$

Soweit nicht anders angegeben, wird für die Viterbi-Näherung im folgenden stets die Maximum-Approximation für Mischverteilungen verwendet. Mit der Definition der diskriminativen Erwartungswerte aus Gleichung (3.4) und der Definition der Erwartungswerte bzgl. der ursprünglichen Parameter $\{c_{s,l},\theta_{s,l}\}$

$$\Omega_{s,l}\left(g(x)\right) = \int\Big\{c_{s,l}\cdot p(x|\theta_{s,l})\cdot g(x)\Big\}\,dx$$

ergibt sich dann:

$$\frac{\partial S(\lambda,\overline{\lambda})}{\partial\overline{\theta}_{s,l}} = \Gamma_{s,l}\left(\frac{\partial\log p(x|\overline{\theta}_{s,l})}{\partial\overline{\theta}_{s,l}}\right) + D_s\,\Omega_{s,l}\left(\frac{\partial\log p(x|\overline{\theta}_{s,l})}{\partial\overline{\theta}_{s,l}}\right) \tag{3.9}$$

$$\frac{\partial S(\lambda,\overline{\lambda})}{\partial\,\overline{c}_{s,l}} = \big[\Gamma_{s,l}(1) + D_s\,\Omega_{s,l}(1)\big]\cdot\frac{1}{\overline{c}_{s,l}} \tag{3.10}$$

Für die Parameteroptimierung der Mischverteilungen werden nun noch die Ableitungen der Gaußschen Emissionsverteilungen benötigt:

$$\frac{\partial \log p(x|\theta_{s,l})}{\partial \mu_{s,l}} = \Sigma_{s,l}^{-1}(x - \mu_{s,l})$$
$$\frac{\partial \log p(x|\theta_{s,l})}{\partial \Sigma_{s,l}^{-1}} = \frac{1}{2}\Big(\Sigma_{s,l} - (x - \mu_{s,l})(x - \mu_{s,l})^T\Big)$$

Unter Verwendung der Identitäten

$$\Omega_{s,l}(1) = c_{s,l}$$
$$\Omega_{s,l}(x) = c_{s,l}\,\mu_{s,l}$$
$$\Omega_{s,l}(x \cdot x^T) = c_{s,l} \cdot \Big(\Sigma_{s,l} + \mu_{s,l} \cdot \mu_{s,l}^T\Big)$$

lassen sich die Reestimationsgleichungen für die neuen Mischverteilungsparameter dann durch Nullsetzen der Gleichungen (3.9) bzw. (3.10) wie folgt herleiten.

Iterationsgleichungen für Mittelwerte:

$$\frac{\partial S(\lambda, \overline{\lambda})}{\partial \overline{\mu}_{s,l}} \stackrel{!}{=} 0 \quad \Longleftrightarrow \quad \overline{\mu}_{s,l} = \frac{\Gamma_{s,l}(x) + D_s c_{s,l}\mu_{s,l}}{\Gamma_{s,l}(1) + D_s c_{s,l}} \tag{3.11}$$

Iterationsgleichungen für Kovarianzmatrizen:

$$\begin{aligned}\frac{\partial S(\lambda, \overline{\lambda})}{\partial \overline{\Sigma}_{s,l}^{-1}} \stackrel{!}{=} 0 \quad \Longleftrightarrow \quad & \overline{\Sigma}_{s,l} = \frac{\Gamma_{s,l}(x \cdot x^T) + D_s \Omega_{s,l}(x \cdot x^T)}{\Gamma_{s,l}(1) + D_s c_{s,l}} + \overline{\mu}_{s,l} \cdot \overline{\mu}_{s,l}^T \\ & - \underbrace{\frac{\Gamma_{s,l}(x) + D_s c_{s,l}\mu_{s,l}}{\Gamma_{s,l}(1) + D_s c_{s,l}}}_{=\overline{\mu}_{s,l}} \cdot \overline{\mu}_{s,l}^T - \overline{\mu}_{s,l} \cdot \underbrace{\frac{\Gamma_{s,l}(x^T) + D_s c_{s,l}\mu_{s,l}^T}{\Gamma_{s,l}(1) + D_s c_{s,l}}}_{=\overline{\mu}_{s,l}^T} \\ \Longleftrightarrow \quad & \overline{\Sigma}_{s,l} = \frac{\Gamma_{s,l}(x \cdot x^T) + D_s c_{s,l}(\Sigma_{s,l} + \mu_{s,l} \cdot \mu_{s,l}^T)}{\Gamma_{s,l}(1) + D_s c_{s,l}} - \overline{\mu}_{s,l} \cdot \overline{\mu}_{s,l}^T \end{aligned} \tag{3.12}$$

Iterationsgleichungen für Mischverteilungsgewichte:
Die Optimierung der Mischverteilungsgewichte muß unter Berücksichtigung der Nebenbedingung $\sum_l \overline{c}_{s,l} = 1$ erfolgen. Hierzu wird unter Verwendung des Lagrangeschen Multiplikators η die folgende modifizierte Hilfsfunktion differenziert:

$$\mathcal{S}(\lambda, \overline{\lambda}, \eta) = S(\lambda, \overline{\lambda}) - \eta \cdot \Big(\sum_l \overline{c}_{s,l} - 1\Big)$$

Die Ableitung von $\mathcal{S}$ nach $\overline{c}_{s,l}$ und Nullsetzen führt dann zu $\overline{c}_{s,l} = \Gamma_{s,l}(1) + D_s c_{s,l}/\eta$. Analog ergibt sich der Ausdruck für den Lagrange-Multiplikator durch Lösen der Gleichung $\partial\mathcal{S}/\partial\eta = 0$ und Einsetzen der Schätzung für $\overline{c}_{s,l}$:

$$\eta = \sum_l \Big(\Gamma_{s,l}(1) + D_s \overline{c}_{s,l}\Big) = \Gamma_s(1) + D_s$$

Damit folgt für die Reestimationsgleichungen der Mischverteilungsgewichte

$$\frac{\partial \mathcal{S}(\lambda, \overline{\lambda}, \eta)}{\partial \overline{c}_{s,l}} \stackrel{!}{=} 0 \quad \wedge \quad \frac{\partial \mathcal{S}(\lambda, \overline{\lambda}, \eta)}{\partial \eta} \stackrel{!}{=} 0 \quad \Longleftrightarrow \quad \overline{c}_{s,l} = \frac{\Gamma_{s,l}(1) + D_s c_{s,l}}{\Gamma_s(1) + D_s} \tag{3.13}$$

Zur Reduktion der Anzahl zu trainierender Parameter können die Kovarianzmatrizen auch zustandsabhängig bestimmt werden ($\Sigma_{s,l} \equiv \Sigma_s$). Die Iterationsgleichungen lauten dann:

$$\overline{\Sigma}_s = \frac{\Gamma_s(x \cdot x^T) + D_s\Big(\Sigma_s + \sum_l c_{s,l}\mu_{s,l} \cdot \mu_{s,l}^T\Big)}{\Gamma_s(1) + D_s} - \sum_l \frac{\Gamma_{s,l}(1) + D_s c_{s,l}}{\Gamma_s(1) + D_s}\,\overline{\mu}_{s,l} \cdot \overline{\mu}_{s,l}^T \qquad (3.14)$$

Wird dagegen eine zustandsunabhängige (gepoolte) Kovarianzmatrix trainiert ($\Sigma_{s,l} \equiv \Sigma$), erhält man die folgende Vorschrift:

$$\overline{\Sigma} = \frac{\Gamma(x \cdot x^T) + \sum_s D\Big(\Sigma + \sum_l c_{s,l}\mu_{s,l} \cdot \mu_{s,l}^T\Big)}{\Gamma(1) + D\sum_s 1} - \sum_{s,l} \frac{\Gamma_{s,l}(1) + D c_{s,l}}{\Gamma(1) + D\sum_s 1}\,\overline{\mu}_{s,l} \cdot \overline{\mu}_{s,l}^T \qquad (3.15)$$

3.5.2 Gradientenabstiegsverfahren

Geschieht die Parameteroptimierung mit Hilfe eines Gradientenverfahrens, so ergeben sich die Iterationsgleichungen für die Mittelwerte, Kovarianzmatrizen und Mischverteilungsgewichte wie folgt:

$$\begin{aligned}
\hat{\mu}_{s,l} &= \mu_{s,l} + \varepsilon_{\mu_{s,l}} \frac{\partial F(\lambda)}{\partial \mu_{s,l}} \\
\hat{\Sigma}_{s,l} &= \Sigma_{s,l} + \varepsilon_{\Sigma_{s,l}} \frac{\partial F(\lambda)}{\partial \Sigma_{s,l}} = \Sigma_{s,l} - \varepsilon_{\Sigma_{s,l}^{-1}} \frac{\partial F(\lambda)}{\partial \Sigma_{s,l}^{-1}} \\
\hat{c}_{s,l} &= \Big(c_{s,l} + \varepsilon_{c_{s,l}} \frac{\partial F(\lambda)}{\partial c_{s,l}}\Big) \Big/ \Big(1 + \sum_l \varepsilon_{c_{s,l}} \frac{\partial F(\lambda)}{\partial c_{s,l}}\Big)
\end{aligned}$$

In [Schlüter et al. 97] wird gezeigt, daß sich für die spezielle Wahl der Schrittweiten

$$\varepsilon_{\mu_{s,l}} = \frac{1}{\Gamma_{s,l}(1) + D_s c_{s,l}}\,\Sigma_{s,l} \qquad \text{bzw.} \qquad \varepsilon_{c_{s,l}} = \frac{\dfrac{\partial F(\lambda)}{\partial c_{s,l}} + D_s - 1}{\dfrac{\partial F(\lambda)}{\partial c_{s,l}}}\,c_{s,l}$$

Reestimationsgleichungen für die Mittelwerte und Mischverteilungsgewichte ergeben, die identisch zu denen aus dem erweiterten Baum-Algorithmus sind. Weiterhin unterscheiden sich die Iterationsgleichungen für die Kovarianzmatrizen nur geringfügig von denen aus dem EB-Algorithmus, wenn die folgenden Schrittweiten gewählt werden:

$$\begin{aligned}
\varepsilon_{\Sigma_{s,l}} &= -\frac{2}{\Gamma_{s,l}(1) + D_s c_{s,l}} \quad \Rightarrow \quad \hat{\Sigma}_{s,l} = \overline{\Sigma}_{s,l} + (\mu_{s,l} - \overline{\mu}_{s,l})(\mu_{s,l} - \overline{\mu}_{s,l})^T \\
\varepsilon_{\Sigma_s} &= -\frac{2}{\Gamma_s(1) + D_s} \quad \Rightarrow \quad \hat{\Sigma}_s = \overline{\Sigma}_s + \sum_l \frac{\Gamma_{s,l}(1) + D_s c_{s,l}}{\Gamma_s(1) + D_s}\,(\mu_{s,l} - \overline{\mu}_{s,l})(\mu_{s,l} - \overline{\mu}_{s,l})^T \\
\varepsilon_{\Sigma} &= -\frac{2}{\Gamma(1) + \sum_s D_s} \quad \Rightarrow \quad \hat{\Sigma} = \overline{\Sigma} + \sum_{s,l} \frac{\Gamma_{s,l}(1) + D_s c_{s,l}}{\Gamma(1) + \sum_{\widetilde{s}} D_s}\,(\mu_{s,l} - \overline{\mu}_{s,l})(\mu_{s,l} - \overline{\mu}_{s,l})^T
\end{aligned}$$

3.6 Wahl der Iterationskonstanten

Die Iterationskonstanten D_s bewirken für die Schätzgleichungen eine Beimischung der alten Mischverteilungsparameter. Je größer der Wert dieser Konstanten ausfällt, desto stärker entsprechen die neuen Parameter denen aus der vorherigen Iteration. Wählt man die Konstanten dagegen zu klein, so operieren die diskriminativen Kriterien zu lokal und führen dadurch zu einer unkontrollierten Oszillation der Zielgröße $F_D(\lambda)$.

Da die Abschätzungen der D_s aus dem Konvergenzbeweis [Gopalakrishnan et al. 91] bereits für diskrete Verteilungen keine brauchbaren Schrittweiten liefern, wird die Forderung nach einem garantiert monotonen Anstieg der Zielgröße zugunsten der Konvergenzgeschwindigkeit aufgegeben. Die (bislang) schnellste Konvergenz ergibt sich dann, wenn die D_s so eingestellt werden, daß alle Nenner und Varianzen der Schätzgleichungen positiv bleiben [Normandin 96]:

$$\Gamma_{s,l}(\cdot) + D_s c_{s,l} \geq \frac{1}{\xi_s} > 0$$
$$\overline{\sigma}^2_{s,l,d} = \left[\,\overline{\Sigma}_{s,l}\right]_{d\,d} > \nu > 0$$

Aus den beiden Ungleichungen ergeben sich unter Berücksichtigung der Reestimationsgleichungen (3.12), (3.14) bzw. (3.15) dann die folgenden Ausdrücke für die Wahl der Iterationskonstanten:

$$D_s = h \cdot \max_{l} \left\{ \frac{1}{c_{s,l}} \left(\frac{1}{\xi_s} - \Gamma_{s,l}(1) \right), D^{min}_{s,l} \right\} \qquad \text{für Gl. (3.12)}$$
$$D_s = h \cdot \max_{l} \left\{ \frac{1}{c_{s,l}} \left(\frac{1}{\xi_s} - \Gamma_{s,l}(1) \right), D^{min}_{s} \right\} \qquad \text{für Gl. (3.14)}$$
$$D = h \cdot \max_{s,l} \left\{ \frac{1}{c_{s,l}} \left(\frac{1}{\xi_s} - \Gamma_{s,l}(1) \right), D^{min} \right\} \qquad \text{für Gl. (3.15).}$$

Dabei gilt für die Konstanten $D^{min}_{(\cdot)}$:

$$D^{min}_{s,l} = \max_{d} \left\{ \frac{-\Gamma_{s,l}(x_d^2) + \nu\Gamma_{s,l}(1) + \left[2\Gamma_{s,l}(x_d) - \Gamma_{s,l}(1)\mu_{s,l,d}\right]\mu_{s,l,d} + \xi_s \left[\Gamma_{s,l}(x_d) - \Gamma_{s,l}(1)\mu_{s,l,d}\right]^2}{\sigma^2_{s,l,d} - \nu} \right\}$$
$$D^{min}_{s} = \max_{d} \left\{ \frac{-\Gamma_{s}(x_d^2) + \nu\Gamma_{s}(1) + \sum_l \left[2\Gamma_{s,l}(x_d) - \Gamma_{s,l}(1)\mu_{s,l,d}\right]\mu_{s,l,d} + \xi_s \sum_l \left[\Gamma_{s,l}(x_d) - \Gamma_{s,l}(1)\mu_{s,l,d}\right]^2}{\sigma^2_{s,d} - \nu} \right\}$$
$$D^{min} = \max_{d} \left\{ \frac{-\Gamma(x_d^2) + \nu\Gamma(1) + \sum_{s,l} \left[2\Gamma_{s,l}(x_d) - \Gamma_{s,l}(1)\mu_{s,l,d}\right]\mu_{s,l,d} + \sum_{s,l} \xi_s \left[\Gamma_{s,l}(x_d) - \Gamma_{s,l}(1)\mu_{s,l,d}\right]^2}{(\sigma^2_d - \nu) \sum_s 1} \right\}$$

Mit Hilfe des Faktors $h \in (1, \infty)$ kann die Geschwindigkeit der Konvergenz gesteuert werden. Für Einzelverteilungen ermöglicht $h = 2.0$ eine schnelle Konvergenz (vgl. Abschnitt 8.2.4). Bei Mischverteilungen sind in der Regel kleinere Werte ($h = 1.1$) zu wählen. Der Parameter ν wurde für die in Kapitel 8 beschriebenen Experimente stets auf 1 gesetzt, da für die verwendeten Modelle alle auftretenden Varianzen deutlich größer waren. Der zustandsspezifische Parameter $1/\xi_s$ wird nicht konstant gesetzt, sondern besitzt eine lineare Abhängigkeit zum Maximum des Betrags der diskriminativen Akkumulatoren:

$$\frac{1}{\xi_s} = a + \max_{l} \Big\{ \left|\Gamma_{s,l}(1)\right| \Big\}$$

Durch die Konstante a wird der Wertebereich des Parameters ξ_s nach oben beschränkt. Dies ist notwendig, da für sehr kleine Beträge der diskriminativen Akkumulatoren die Iterationskonstanten ansonsten zu groß würden und damit eine drastische Reduktion der Konvergenzgeschwindigkeit zur Folge hätten.

3.6.1 Optimierung der Mischverteilungsgewichte

Für die Reestimation der Mischverteilungsgewichte nach Gleichung 3.13 wird in [Gopalakrishnan et al. 89] zu langsame Konvergenz berichtet. Die Ursache wird auf eine Überbewertung statistisch nur schwach vetretener Mischungskomponenten zurückgeführt, was die Bildung sehr großer Iterationskonstanten $D_{(\cdot)}$ zur Folge hat. Aus diesem Grund wird eine schneller konvergierende Variante vorgeschlagen, wobei die Gewichtsparameter mit einem Faktor multipliziert werden, der schwach besetzte Mischungskomponenten herunterwichtet. Für die Iterationsgleichungen ergibt sich dann

$$\bar{c}_{s,l} = \frac{\Gamma^{spk}_{s,l}(1) + \Gamma^{gen}_{s,l}(1)}{\Gamma^{spk}_{s}(1) + \Gamma^{gen}_{s}(1)} \cdot \frac{\Gamma_{s,l}(1) + C_s}{\sum\limits_{\widetilde{l}} c_{s,\widetilde{l}}\, \Gamma_{s,\widetilde{l}}(1) + C_s}\; c_{s,l}\;, \tag{3.16}$$

wobei die Größen $\Gamma^{spk}_{s,l}(\cdot)$ und $\Gamma^{gen}_{s,l}(\cdot)$ die Aufspaltung der diskriminativen Erwartungswerte in ihre Anteile für die gesprochene Wortfolge bzw. alle konkurrierenden Wortfolgen bezeichnen:

$$\Gamma^{spk}_{s,l}\big(g(x)\big) = \sum_{r=1}^{R} \alpha\, f'\!\left(\log \frac{p^{\alpha}_{\lambda}(X_r, W_r)}{\sum\limits_{W \in \mathcal{M}_r} p^{\alpha}_{\lambda}(X_r, W)} \right) \cdot \sum_{t=1}^{T_r} \gamma_{r,t}(s|W_r) \frac{c_{s,l}\, p(x_{r,t}|\theta_{s,l})}{\sum\limits_{\widetilde{l}} c_{s,\widetilde{l}} p(x_{r,t}|\theta_{s,\widetilde{l}})} \cdot g(x_{r,t})$$

$$\Gamma^{gen}_{s,l}\big(g(x)\big) = \sum_{r=1}^{R} \alpha\, f'\!\left(\log \frac{p^{\alpha}_{\lambda}(X_r, W_r)}{\sum\limits_{W \in \mathcal{M}_r} p^{\alpha}_{\lambda}(X_r, W)} \right) \cdot \sum_{t=1}^{T_r} \gamma_{r,t}(s) \frac{c_{s,l}\, p(x_{r,t}|\theta_{s,l})}{\sum\limits_{\widetilde{l}} c_{s,\widetilde{l}} p(x_{r,t}|\theta_{s,\widetilde{l}})} \cdot g(x_{r,t})$$

Die Wahl der neuen Iterationskonstanten C_s wird in [Gopalakrishnan et al. 89] mit

$$C_s = \max_{l} \Big\{ -\Gamma_{s,l}(1), 0 \Big\} + \varepsilon$$

angegeben, wobei ε eine kleine positive Kostante ist (hier: $\varepsilon = 1$). Einem Argument aus [Merialdo 88] folgend, wird in [Normandin 91] eine Reestimationsgleichung angegeben, die noch schnellere, jedoch auch instabilere Konvergenz verspricht. Sie ergibt sich, wenn die in Gleichung 3.13 eingehende Ableitung

$$\frac{\partial F(\lambda)}{\partial c_{s,l}} = \frac{\Gamma_{s,l}(1)}{c_{s,l}}$$

durch

$$\frac{\partial F(\lambda)}{\partial c_{s,l}} = \frac{\Gamma^{spk}_{s,l}(1)}{\Gamma^{spk}_{s}(1)} - \frac{\Gamma^{gen}_{s,l}}{\Gamma^{gen}_{s}(1)}$$

ersetzt wird.

Für die Iterationsgleichungen der Mischverteilungsgewichte ergibt sich dann:

$$\overline{c}_{s,l} = \frac{\dfrac{\Gamma_{s,l}^{spk}(1)}{\Gamma_{s}^{spk}(1)} - \dfrac{\Gamma_{s,l}^{gen}(1)}{\Gamma_{s}^{gen}(1)} + C_s}{\displaystyle\sum_{\widetilde{l}} c_{s,\widetilde{l}} \left[\frac{\Gamma_{s,\widetilde{l}}^{spk}(1)}{\Gamma_{s}^{spk}(1)} - \frac{\Gamma_{s,\widetilde{l}}^{gen}(1)}{\Gamma_{s}^{gen}(1)} \right] + C_s} \, c_{s,l} \tag{3.17}$$

mit

$$C_s = \max_{l} \left\{ - \left[\frac{\Gamma_{s,l}^{spk}(1)}{\Gamma_{s}^{spk}(1)} - \frac{\Gamma_{s,l}^{gen}(1)}{\Gamma_{s}^{gen}(1)} \right], 0 \right\} + \varepsilon$$

wobei die Größe ε durch den folgenden Ausdruck gegeben ist:

$$\varepsilon := \begin{cases} \varepsilon_0 & , \text{ falls } \max\limits_{l} \left\{ \left| - \left[\dfrac{\Gamma_{s,l}^{spk}(1)}{\Gamma_{s}^{spk}(1)} - \dfrac{\Gamma_{s,l}^{gen}(1)}{\Gamma_{s}^{gen}(1)} \right] \right| \right\} = 0 \\ h \cdot \max\limits_{l} \left\{ \left| - \left[\dfrac{\Gamma_{s,l}^{spk}(1)}{\Gamma_{s}^{spk}(1)} - \dfrac{\Gamma_{s,l}^{gen}(1)}{\Gamma_{s}^{gen}(1)} \right] \right| \right\} & , \text{ sonst} \end{cases}$$

Praktische Anmerkungen

Speziell für den Fall, daß das CT-Kriterium verwendet wird, können die diskriminativen Akkumulatoren $\Gamma_s^{gen}(1)$ für einen Mischverteilungsindex s gleich null werden. Die Ausdrücke in den Gleichungen 3.16 und 3.17 sind dann nicht mehr definiert und können folglich nicht mehr zur Reestimation der Mischverteilungsgewichte herangezogen werden. Aus diesem Grund wurde die Reestimation der Gewichte mit einem expliziten Test auf singuläre Nenner implementiert. Kommt es während der Reestimation zu einer Division durch null, wird automatisch die nächste, numerisch stabilere Variante herangezogen. Auf diese Weise wird für jedes Gewicht zunächst die Gleichung 3.17 zur Bestimmung des neuen Parameters $\overline{c}_{s,l}$ herangezogen und dann, falls erforderlich, die Gleichung 3.16 und schließlich Gleichung 3.13.

3.6.2 Adaptive Schrittweitenbestimmung

Obwohl die Wahl der Schrittweite $h = 2$ für die ersten Iterationen in den meisten Fällen eine schnelle Konvergenz erlaubt, kann die Einstellung für die höheren Iterationen zu niedrig sein und damit sporadisch zu einem (teils drastischen) Anstieg der Fehlerrate führen. Die Ursache hierfür liegt darin, daß die Schätzungen der neuen Mischverteilungsparameter stets durch die Verteilungsparameter der vorherigen Iteration geglättet werden müssen, um eine zu starke Konzentration auf fehlerhaft klassifizierte Beobachtungen zu vermeiden. Bei einer zu klein gewählten Schrittweite geht dieser Einfluß der alten Verteilungsparameter für die Reestimation jedoch verloren und nicht selten vermögen die neu geschätzten Parameter dann nur noch ausschließlich die zuvor falsch klassifizierten Beobachtungen korrekt zu erkennen.

Um dieses Problem zu umgehen, wird in [Normandin 96] die Schrittweite in jeder Iteration sukzessive um einen konstanten Faktor verkleinert, so daß mit zunehmendem Iterationsindex auch der Einfluß der alten Mischverteilungsparameter wächst. Diese Methode besitzt

jedoch zwei entscheidende Nachteile. Zum einen kann so nicht garantiert werden, daß ein lokales Optimum überhaupt erreicht wird, da die Schrittweiten sehr schnell sehr klein werden und damit eine künstliche Konvergenz der Zielfunktion auf einen Punkt außerhalb eines lokalen Minimums erzwingen können. Zum anderen hat sich gezeigt, daß kleinere Sprünge während der Iterationen dazu führen können, von einem lokalen Minimum in ein benachbartes, besseres Minimum zu gelangen. Aus diesem Grund wurde ein Verfahren zur adaptiven Schrittweitenbestimmung entwickelt und implementiert, das gemäß dem folgenden Schema arbeitet: Steigt die Fehlerrate in der Iteration $i+1$ gegenüber der Iteration i um mehr als das doppelte an, wird die Reestimation der Verteilungsparameter am Ende der Iteration i mit der kleineren Schrittweite (d.h. dem größeren Faktor) $h_{\text{neu}} = 2 \cdot h_{\text{alt}} + 1$ wiederholt und das Training mit dem Beginn der Iteration $i+1$ erneut aufgesetzt. Die Größenordnung der neuen Schrittweite ist im allgemeinen jedoch korpusspezifisch zu wählen und muß daher empirisch bestimmt werden.

3.7 Übergang zum konventionellen ML-Training

Wie in Abschnitt 2.5 beschrieben, geschieht die Optimierung der Mischverteilungsparameter im konventionellen ML-Training unter Verwendung der folgenden Hilfsfunktion (der sogenannten *Kullback-Leibler-Statistik* oder Q-Funktion):

$$Q(\lambda, \overline{\lambda}) = \sum_{r=1}^{R} \sum_{[s_1^{T_r}]|W_r} p_\lambda(s_1^{T_r}|x_{r,1}^{T_r}) \cdot \log p_{\overline{\lambda}}(x_{r,1}^{T_r}, s_1^{T_r})$$

Diese Form ist äquivalent zu dem folgenden Ausdruck (vgl. Anhang A.2):

$$= \sum_s \sum_{r=1}^{R} \sum_{t=1}^{T_r} \Bigg[\underbrace{\gamma_{r,t}(s|W_r) \cdot \log p_{\overline{\lambda}}(x_{r,t}|\overline{\theta}_{s_t})}_{=:q_1} + \underbrace{\sum_\sigma \gamma_{r,t}(\sigma, s|W_r) \cdot \log p(s|\sigma)}_{=:q_2} \Bigg]$$

Hierbei werden die Emissionswahrscheinlichkeiten durch $\sum_s \sum_r \sum_t q_1$ berücksichtigt, während die Transitionswahrscheinlichkeiten durch $\sum_s \sum_r \sum_t q_2$ erfaßt werden.

Setzt man nun in Gleichung 3.7 die generalisierten FB-Wahrscheinlichkeiten sowie sämtliche Iterationskonstanten D_s zu null und wählt ferner $f = \text{id}$ als Glättungsfunktion, so geht die Hilfsfunktion für die Parameteroptimierung im diskriminativen Training in die Kullback-Leibler-Statistik $Q(\lambda, \overline{\lambda})$ für das Training der Emissionswahrscheinlichkeiten im konventionellen ML-Ansatz über. Die Iterationsgleichungen für die Mischverteilungsparameter ergeben sich dann ebenfalls in analoger Weise:

Iterationsgleichungen für Mittelwerte (ML):

$$\frac{\partial Q(\lambda, \overline{\lambda})}{\partial \overline{\mu}_{s,l}} \stackrel{!}{=} 0 \quad \Longleftrightarrow \quad \overline{\mu}_{s,l} = \frac{\Gamma_{s,l}^{spk}(x)}{\Gamma_{s,l}^{spk}(1)} \tag{3.18}$$

Iterationsgleichungen für Kovarianzmatrizen (ML):

$$\frac{\partial Q(\lambda,\overline{\lambda})}{\partial\overline{\Sigma}_{s,l}} \stackrel{!}{=} 0 \quad \Longleftrightarrow \quad \overline{\Sigma}_{s,l} = \frac{\Gamma^{spk}_{s,l}(x\cdot x^T)}{\Gamma^{spk}_{s,l}(1)} + \overline{\mu}_{s,l}\cdot\overline{\mu}^T_{s,l} - \underbrace{\frac{\Gamma^{spk}_{s,l}(x)}{\Gamma^{spk}_{s,l}(1)}}_{=\overline{\mu}_{s,l}}\cdot\overline{\mu}^T_{s,l} - \overline{\mu}_{s,l}\cdot\underbrace{\frac{\Gamma^{spk}_{s,l}(x^T)}{\Gamma^{spk}_{s,l}(1)}}_{=\overline{\mu}^T_{s,l}}$$

$$\Longleftrightarrow \quad \overline{\Sigma}_{s,l} = \frac{\Gamma^{spk}_{s,l}(x\cdot x^T)}{\Gamma^{spk}_{s,l}(1)} - \overline{\mu}_{s,l}\cdot\overline{\mu}^T_{s,l} \tag{3.19}$$

Im Falle zustandsspezifischer und gepoolter Kovarianzmatrizen ergeben sich die Gleichungen zu

$$\overline{\Sigma}_s = \frac{\Gamma^{spk}_{s}(x\cdot x^T)}{\Gamma^{spk}_{s}(1)} - \sum_l \frac{\Gamma^{spk}_{s,l}(1)}{\Gamma^{spk}_{s}(1)}\,\overline{\mu}_{s,l}\cdot\overline{\mu}^T_{s,l} \tag{3.20}$$

bzw.

$$\overline{\Sigma} = \frac{\Gamma^{spk}(x\cdot x^T)}{\Gamma^{spk}(1)} - \sum_{s,l} \frac{\Gamma^{spk}_{s,l}(1)}{\Gamma^{spk}(1)}\,\overline{\mu}_{s,l}\cdot\overline{\mu}^T_{s,l} \tag{3.21}$$

Iterationsgleichungen für Mischverteilungsgewichte (ML):
Die Optimierung der Mischverteilungsgewichte erfolgt ebenfalls unter Berücksichtigung der Nebenbedingung $\sum_l \overline{c}_{s,l} = 1$. Hierzu wird unter Verwendung des Lagrangeschen Multiplikators η wieder eine modifizierte Hilfsfunktion differenziert:

$$\mathcal{Q}(\lambda,\overline{\lambda},\eta) = Q(\lambda,\overline{\lambda}) - \eta\cdot\Big(\sum_l \overline{c}_{s,l} - 1\Big)$$

In ähnlicher Weise führt nun die Ableitung von $\mathcal{Q}$ nach $\overline{c}_{s,l}$ und Nullsetzen zu $\overline{c}_{s,l} = \Gamma^{spk}_{s,l}(1)/\eta$. Der Ausdruck für den Lagrange-Multiplikator ergibt sich dann analog durch Lösen der Gleichung $\partial\mathcal{Q}/\partial\eta = 0$ und Einsetzen der Schätzung für $\overline{c}_{s,l}$:

$$\eta = \sum_l \Gamma^{spk}_{s,l}(1) = \Gamma^{spk}_{s}(1)$$

Damit folgt schließlich für die Reestimationsgleichungen der Mischverteilungsgewichte

$$\frac{\partial\mathcal{Q}(\lambda,\overline{\lambda},\eta)}{\partial\overline{c}_{s,l}} \stackrel{!}{=} 0 \quad \Longleftrightarrow \quad \overline{c}_{s,l} = \frac{\Gamma^{spk}_{s,l}(1)}{\Gamma^{spk}_{s}(1)} \tag{3.22}$$

Kapitel 4

Organisation der Suche im diskriminativen Training

Bei Verwendung des MMI- und MCE-Kriteriums wird während des Trainings in jeder Iteration für jede Äußerung ein Wortgraph generiert. Die in dem Wortgraphen enthaltenen Satzalternativen bilden die Menge der konkurrierenden Hypothesen zur Trainingsäußerung r. In diesem Kapitel wird die Organisation und Implementierung der Suche im diskriminativen Training für das Baum-Welch Training und das Viterbi-Training beschrieben. Dabei wird gezeigt, daß viele im Kontext des diskriminativen Trainings auftretenden Schwierigkeiten wie das Problem der *redundanten Silence-Kanten* oder das für das MMI-Kriterium notwendige Einfügen der gesprochenen Wortfolge in den Wortgraphen durch geeignete Modellierungen bereits während der Suche gelöst werden können. In Bezug auf die Wahl der HMM-Topologie können unerwünschte Effekte, wie die Bildung sogenannter *Quasi-Mischverteilungen*[1] durch ein geeignetes Zusammenfassen von aufeinander folgenden Zuständen verhindert werden. Zusätzlich werden zwei Verfahren zur Beschleunigung der Suche im diskriminativen Training vorgestellt.

4.1 Bildung von Quasi-Mischverteilungen

Im Unterschied zum konventionellen ML-Ansatz reagieren viele diskriminative Verfahren wie das CT- und das MMI-Kriterium wesentlich sensibler auf Ausreißer in den Trainingsdaten. Ist beispielsweise jeder Zustand einer Markovkette eineindeutig mit einer unimodalen Gauß-Verteilung assoziiert, kann dies im diskriminativen Training bei Verwendung eines Bakis-Modells dazu führen, daß bestimmte Zustände ausschließlich zur Erfassung dieser Ausreißer trainiert werden. Zwar hat dies eine bessere Abdeckung des Trainingskorpus zur Folge, gleichzeitig entfallen aber auf die hiervon betroffenen Zustände nur sehr wenige Beobachtungen. Die Schätzungen der jeweiligen Verteilungsparameter sind dementsprechend unsicher und konzentrieren im Verlauf der Trainingsiterationen immer stärker auf die eher untypischen Beobachtungen. Im Gegenzug werden die typischen Beobachtungen mit einer *skip*-Transition an einem so trainierten Ausreißermodell vorbeibewegt, um eine möglichst große Emissionswahrscheinlichkeit zu produzieren[2]. Dieses Verhalten ist in Abbildung 4.1

[1] Streng genommen sind die Quasi-Mischverteilungen nicht der Suche zuzuordnen, sonder gehören in den Bereich der akustischen Modellierung.

[2] Eine typische Beobachtung kann durch ein Ausreißermodell nur schlecht erklärt werden und würde in Folge dessen zu einer sehr kleinen Emissionswahrscheinlichkeit führen.

illustriert. Der gestrichelt dargestellte Zustand s_2 bezeichnet hierbei das Ausreißermodell. Die Markovkette zeigt ein Verhalten, das der Verwendung von Mischverteilungen ähnelt. Der Effekt wird daher im folgenden auch als „Bildung von *Quasi-Mischverteilungen*“ bezeichnet.

Im Rahmen eines CT-Trainings für unimodale Gauß-Verteilungen und zustandsspezifische Varianzen hat dies bereits dazu geführt, daß nur noch eine Beobachtung auf den Zustand eines Ausreißermodells entfiel, so daß eine Schätzung der Varianz nicht mehr möglich war.

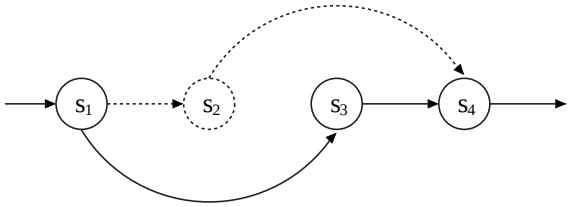


Abbildung 4.1: Quasi-Mischverteilungen bei Markovketten

Selbst wenn auf die Ausreißermodelle hinreichend viele Beobachtungen entfallen, so daß eine Reestimation der Mischverteilungsparameter technisch noch möglich ist, führt dies in der Regel zu einer Überadaption des Trainingskorpus und damit aufgrund einer schlechten Generalisierbarkeit der Ausreißer zu größeren Fehlerraten bei der Erkennung bislang ungesehener Testdaten. Quasi-Mischverteilungen erlauben darüber hinaus auch keine eindeutige Interpretation der sogenannten Zeitverzerrungsstrafe. Die Zeitverzerrungsstrafe bildet eine Art vereinfachte Transitionswahrscheinlichkeit, bei der die *forward*-Übergänge gegenüber den *loop*- und den *skip*-Transitionen begünstigt werden.

Quasi-Mischverteilungen treten jedoch nicht nur im diskriminativen Training auf, sondern konnten bereits im konventionellen ML-Training beobachtet werden. Die geringe Sensibilität gegenüber Ausreißern verhindert hier jedoch im allgemeinen eine Überadaption des Trainingskorpus.

Um nun die Bildung von Quasi-Mischverteilungen zu vermeiden, werden jeweils zwei benachbarte Zustände zu einem *Segment* zusammengefaßt. Beide Zustände sind dann über den Segmentindex mit derselben Mischverteilung zu assoziieren, so daß Beobachtungen, die auf nur einen der beiden Zustände entfallen, der gemeinsamen Mischverteilung zugeordnet werden. Dadurch ist gewährleistet, daß auch bei Verwendung eines Bakis-Modells kein Mischverteilungsindex eines Wortautomaten ausgelassen werden kann (siehe Abbildung 4.2).

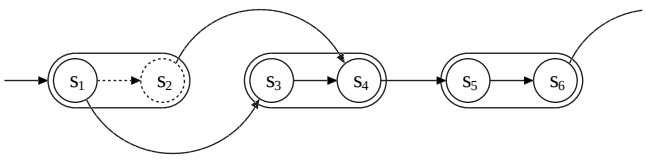


Abbildung 4.2: Segmentierte Zustandsfolge

4.2 Suche für zeitabhängige Wortgraphen

Da die verfügbare Standard-Suche für kleines Vokabular die Generierung von Wortgraphen noch nicht ermöglicht hat, wurde im Rahmen dieser Arbeit ein Suchverfahren für die Generierung *zeitabhängiger* Wortgraphen (bei kleinem Vokabular) implementiert. Dabei bedeutet die Bezeichnung „zeitabhängig“, daß der resultierende Wortgraph zu jedem Zeitpunkt nur höchstens einen Knoten besitzt. Die spezielle Form der Wortgraphen wurde im Hinblick auf die Struktur des FB-Algorithmus zur Berechnung der generalisierten FB-Wahrscheinlichkeiten gewählt. Die Suche erfolgt in der Viterbi-Näherung und verwendet ein lineares Aussprachelexikon.

4.2.1 Produktion von Wortgraphen

Die grundlegende Suchstrategie zur Dekodierung der gesprochenen Wortfolge basiert auf dem *one-pass* Algorithmus [Vintsyuk 71, Bridle et al. 82, Ney 84, Ney und Aubert 96], mit dem diejenige Wortsequenz bestimmt werden kann, die gemäß der Bayesschen Entscheidungsregel die akustischen Beobachtungen (unter den gegebenen Modellen) am besten zu erklären vermag. Die Arbeisweise dieses Verfahrens wird im folgenden unter dem Aspekt der Integration in ein System zur Produktion von Wortgraphen beschrieben.

Zeitabhängige Baumsuche

Die Bestimmung der besten erkannten Wortfolge ist gemäß der Bayesschen Entscheidungsregel (unter Verwendung der Viterbi-Näherung) äquivalent zur Suche nach derjenigen Zustandsfolge, dessen zugehörige Wortsequenz die a-posteriori Wahrscheinlichkeit maximiert:

$$[w_1^N]_{\text{opt}} = \underset{[w_1^N]}{\text{argmax}} \left\{ Pr(w_1^N) \cdot \max_{[s_1^T]|w_1^N} \left\{ Pr\left(x_1^T, s_1^T | w_1^N\right) \right\} \right\}$$

Das Optimierungsproblem wird mittels der dynamischen Programmierung gelöst, wobei das Auffinden des global optimalen Pfades auf die lokalen Lösungen einfacherer Teilprobleme zurückgeführt wird. Die Gesamtlösung ergibt sich dann durch Rekombination der zwischengespeicherten Teillösungen.

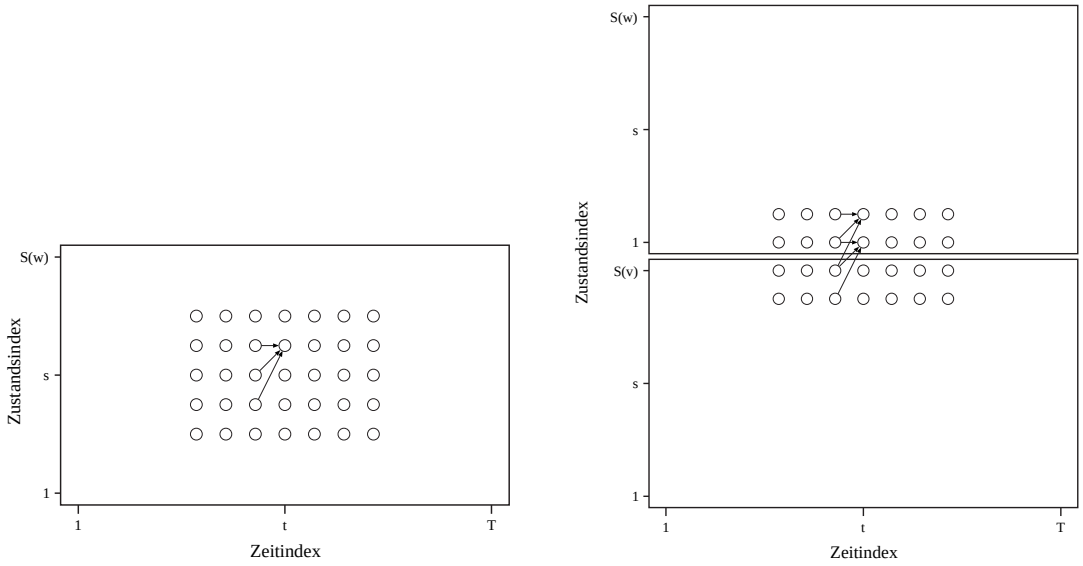
Ein Teilproblem der Suche bildet das Auffinden des lokal optimalen Pfades zum Zeitpunkt t (Teilpfad). Jedem Teilpfad wird dabei die folgende Bewertung zugeordnet:

$$-\log p_\lambda(x_1^t, s_1^t | w_1^n) - \log p(w_1^n)$$

Laufen innerhalb eines Wortes mehrere Pfade in einem Gitterpunkt zusammen, so wird nur der best-bewertetste Teilpfad weiter verfolgt. An den Wortgrenzen findet eine Rückkopplung mit dem Sprachmodell statt, so daß zwei Arten von Rekombinationen unterschieden werden müssen [Ney 93]:

- Rekombination im Wortinneren
- Rekombination an den Wortgrenzen

In Abbildung 4.3 ist die Rekombination für Pfade im Wortinneren (a), sowie die Rekombination an den Wortgrenzen (b) dargestellt. Dabei bezeichnet $S(w)$ den Endzustand des Wortautomaten w. Für die Rekombination an den Wortgrenzen wird im folgenden ein



(a) Intra-Wort-Rekombination

(b) Inter-Wort-Rekombination

Abbildung 4.3: Pfad-Rekombinationen in der Viterbi-Suche

Unigramm-Sprachmodell angenommen.

Wird $Q(t, s; w)$ definiert als die Bewertung des besten Pfades, der zum Zeitpunkt t im Zustand s des Wortes w endet, dann läßt sich $Q(t, s; w)$ durch die folgende Rekursionsgleichung beschreiben:

$$Q(t, s; w) = \min_{\sigma}\Big\{ Q(t-1, \sigma; w) - \log p_\lambda(x_t.s|\sigma, w) \Big\}$$

Um die erkannte Wortfolge zu rekonstruieren, werden zusätzlich noch Rückwärtszeiger benötigt, die auf die jeweils beste Vorgängerhypothese verweisen. Bezeichnet $B(t, s; w)$ die Startzeit des besten, zum Zeitpunkt t im Zustand s des Wortes w endenden Pfades, dann gilt

$$B(t, s; w) = B(t-1, \sigma_{\max}(t, s; w); w),$$

wobei $\sigma_{\max}(t, s; w)$ den besten Vorgängerzustand zur Hypothese $(t, s; w)$ bezeichnet. Um Wortgrenzen zu berücksichtigen, wird ein spezieller Zustand $s = 0$ eingeführt, aus dem heraus neue Worthypothesen gestartet werden können. Die Rekombination über die Vorgängerwörter v an einer potentiellen Wortgrenze geschieht dann unter Verwendung der folgenden Hilfsgröße:

$$H(w; t) := \min_{v}\Big\{ Q(t, S(v); v) - \log p(w) \Big\}$$

Entsprechend sind nun die Größen Q und B für die Wortgrenzen zu erweitern:

$$Q(t-1, 0; w) = H(w; t-1)$$
$$B(t-1, 0; w) = t-1$$

Der Zeitpunkt, zu dem eine Wortgrenze hypothetisiert wird, ist durch den folgenden Ausdruck bestimmt:

$$\tau(t;w) = B(t, S(w); w)$$

Die Auswertung von $\tau(t;w)+1$ liefert dann gerade die Startzeit der Worthypothese w.

Auswahl der Hypothesen für den Wortgraphen

Für die Konstruktion des Wortgraphen werden nur solche Hypothesen herangezogen, deren Bewertungen sehr nahe bei der lokal besten Hypothese liegen. Zur Beschreibung der Wortgraph-Konstruktion benötigen wir daher noch die folgenden Definitionen:

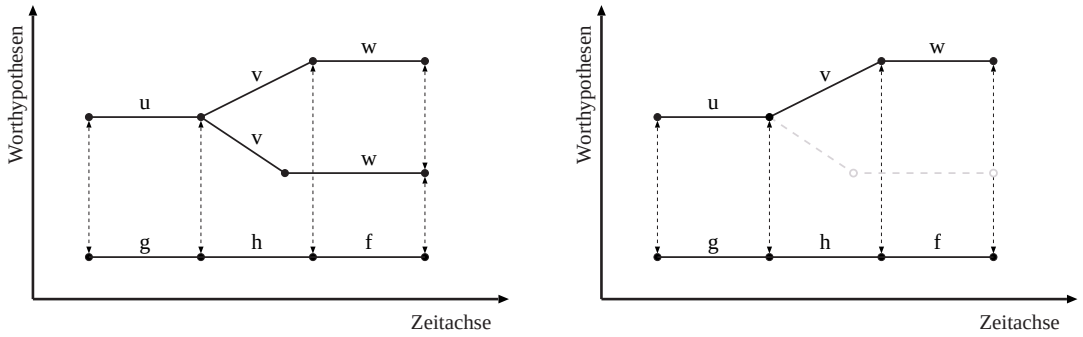
$$\begin{aligned} h(w;\tau,t) &:= -\log p_\lambda(x_{\tau+1}^t|w) \\ G(w_1^n;t) &:= -\log p_\lambda(x_1^t|w_1^n) - \log p(w_1^n) \end{aligned}$$

Hierbei ist die Größe $h(w;\tau,t)$ definiert als der negative Logarithmus der Wahrscheinlichkeit, daß das Wort w die akustischen Vektoren $x_{\tau+1},...,x_t$ produziert. $G(w_1^n;t)$ ist definiert als der negative Logarithmus der Verbundwahrscheinlichkeit, daß die Wortsequenz w_1^n die akustischen Vektoren $x_1,...,x_t$ generiert. Mit Hilfe der beiden Definitionen läßt sich der Beitrag des negativen Logarithmus der Emissionswahrscheinlichkeit einer einzelnen Worthypothese w_n zur Verbundwahrscheinlichkeit $p_\lambda(x_1^t, w_1^n)$ dann wie folgt isolieren:

$$\underbrace{x_1, \ldots, x_\tau}_{G(w_1^{n-1};\tau)} \underbrace{x_{\tau+1}, \ldots, x_t}_{h(w_n;\tau,t)} \underbrace{x_{t+1}, \ldots, x_T}_{\ldots}$$

Wie experimentelle Untersuchungen gezeigt haben, müssen die zu generierenden Wortgraphen bestimmten Anforderungen genügen, um einen Performanzgewinn im diskriminativen Training erzielen zu können. So sollte ein Wortgraph zum einen möglichst viele verschiedene Wortfolgen enthalten, um eine brauchbare Näherung für die (im Idealfall unendlich große) Menge $\mathcal{M}_r$ von konkurrierenden Satzhypothesen zu bilden. Weiterhin sollte jeder Wortgraph zu jeder repräsentierten Satzhypothese nur die beste zeitliche Segmentierung enthalten. Der Grund hierfür liegt darin, daß zusätzliche Segmentierungen identischer Wortfolgen eine Pfadvervielfältigung im Wortgraphen bewirken und damit zu einer Verzerrung bei der Berechnung der FB-Wahrscheinlichkeiten führen. Enthält der Wortgraph beispielsweise viele verschiedene zeitliche Segmentierungen ein und derselben Satzhypothese, dann entfällt auch ein großer Teil der Wahrscheinlichkeitsmasse bei der Berechnung der FB-Wahrscheinlichkeiten auf die hierzu gehörenden Worthypothesen. Umgekehrt nimmt der Anteil der Wahrscheinlichkeitsmasse, die auf nur wenig repräsentierte Satzhypothesen entfällt, in entsprechender Weise ab. In Abbildung 4.4 (a) ist ein solcher Wortgraph exemplarisch dargestellt.

Während die Worthypothesenfolge (g,h,f) durch nur einen Weg in dem Wortgraphen beschrieben werden kann, gibt es für die Wortsequenz (u,v,w) dagegen zwei mögliche Wege. Das Auftreten unterschiedlicher Zeitanpassungen für identische Wortfolgen führt hier also zu einer Vervielfachung der Pfade, die durch die Worthypothese u führen. Die Konsequenz hieraus ist eine deutlich größere FB-Wahrscheinlichkeit für die Worthypothese u, die zu Lasten der FB-Wahrscheinlichkeit der Satzhypothese (g,h,f) geht.



(a) Pfadvervielfältigung durch Mehrfachvorkommen gleicher Wortfolgen in einem Wortgraphen

(b) Wortgraph nach Entfernen des redundanten Teilpfades (v, w)

Abbildung 4.4: Wortgraph mit und ohne redundante Pfade

Da der in Abschnitt 3.4.2 beschriebene FB-Algorithmus für Wortgraphen jedoch nur eine Summation über unterschiedliche Wortfolgen vorsieht, ist es erforderlich, die redundanten Pfade aus dem Wortgraphen soweit wie möglich zu eliminieren. Dieses Ziel wird im folgenden durch Kombination der sogenannten *Wortpaarapproximation* sowie einem zusätzlichen Pruning auf Wortgraphen erreicht. Für das vorherige Beispiel ergibt das Entfernen des redundanten Teilpfades (v, w) dann die Abbildung 4.4 (b).

Im allgemeinen ist es jedoch nicht möglich, sämtliche Redundanzen aus einem Wortgraphen zu entfernen, ohne zugleich die Kompaktheit der Struktur einzubüßen. Wollte man dies erreichen, müßten sämtliche in dem Wortgraphen enthaltenen Worthypothesen auf bestimmte Kontexte eingeschränkt werden. Der hiermit verbundene buchungstechnische Overhead führt jedoch dann sehr schnell zu einer Größenordnung, die exponentiell in der Zahl der Kanten des Wortgraphen ansteigt.

Um den Wortgraphen nun kompakt zu halten und Redundanzen weitgehend zu vermeiden, könnten zu einem Zeitpunkt t unter den endenden Hypothesen mit gleichem Worthypothesenindex alle bis auf die beste Hypothese verworfen werden. Eine solche Vorgehensweise erwies sich jedoch als zu restriktiv, da hierdurch auch zahlreiche Vorgängerhypothesen verloren gingen. Statt dessen wird zeitsynchron zur Suche eine Liste der besten endenden Wortpaare mitgeführt. Eine zu einem Zeitpunkt t endende Hypothese w wird dann nur in dem Fall verworfen, wenn es zu jedem ihrer Vorgänger v ein alternatives, besser bewertetes Worthypothesenpaar gleicher Indexfolge (v, w) gibt. Diese sogenannte *Wortpaarapproximation* wird auch bei der Suche mit wortabhängigen Baumkopien auf großem Vokabular zur Bestimmung der Wortgrenzen verwendet [Ney und Nießen 95]. Um dynamisch die Menge der besten endenden Wortpaare bei der zeitabhängigen Suche bewerten zu können, werden die folgenden Hilfsgrößen definiert:

$$U(t; w) := \min\Big\{Q(t, S(w); w),\ U(t-1; w) + h(\text{sil}, t-1, t)\Big\}$$

$$P(t; v, w) := \min\Big\{U(\tau(t; w); v) - \log p(w) + h(w; \tau(t; w), t),\ P(t-1, v, w) + h(\text{sil}, t-1, t)\Big\}$$

Hierbei bezeichnet $U(t;w)$ die Bewertung der besten zum Zeitpunkt t endenden Worthypothese mit Index w. In ähnlicher Weise ist $P(t;v,w)$ definiert als die beste Bewertung aller zum Zeitpunkt t endenden Wortpaare mit Indexfolge (v,w). Die Addition der Größe $h(\text{sil}, t-1, t)$ zu den Bewertungen der zum Zeitpunkt $t-1$ geendeten Worthypothesen bzw. Wortpaarhypothesen ermöglicht es hierbei, Hypothesen mit identischem Index aber unterschiedlicher Endzeit über Silence-Kanten hinweg miteinander zu vergleichen.

Da die Zugriffe auf die Liste der zum Zeitpunkt t endenden besten Wortpaare stets nur für den aktuellen Zeitindex erfolgen, ist es – im Gegensatz zu den Größen $U(t;w)$ – nicht erforderlich, die Einträge für $P(t;v,w)$ über den gesamten Zeiraum 1,...,t vorzuhalten.

In Tabelle 4.1 sind die wichtigsten Schritte des Algorithmus zur Generierung der zeitabhängigen Wortgraphen zusammengefaßt. Die verwendeten Datenstrukturen sind aus den Abbildungen 4.5 und 4.6 ersichtlich.

Tabelle 4.1: *One-pass* Algorithmus für die Konstruktion von Wortgraphen

INITIALIZATION:

proceed over time t from left to right
proceed over all active tree hypotheses
proceed over all active word hypotheses
proceed over all active state hypotheses
ACOUSTIC LEVEL: - forward recombination - expansion of hypotheses arrays
proceed over all active tree hypotheses
proceed over all active word hypotheses
proceed over all active state hypotheses
prune unlikely state hypotheses: - purge hypotheses arrays
keep word hypothesis active iff at least one state hypothesis has survived
keep tree hypothesis active iff at least one word hypothesis has survived
process word ends:
- store word hypotheses in word graph - insert spoken word hypothesis if necessary - start new tree hypothesis

POST PROCESSING:
- remove dead hypotheses - mark spoken word sequence - combine silence frames to silence arcs - insert forward pointers

Zu jedem Zeitpunkt t werden nun die endenden Hypothesen w_n bezüglich ihrer Bewertung $G(w_1^{n-1}, w_n; t)$ in aufsteigender Reihenfolge sortiert. Anschließend wird die Menge aller besten endenden Worthypothesenpaare (v,w) bestimmt. Sofern zum Zeitpunkt $t-1$ ein Wortpaar mit identischer Worthypothesenindexfolge geendet hat, wird dessen Bewertung $G(u_1^{n-2}, v, w; t-1)$ um $h(\text{sil}, t-1, t)$ vergrößert und mit der Bewertung der neuen Hypothese verglichen. Jede Worthypothese w ist nun mit einem Zähler $c(w;t)$ assoziiert, der die Anzahl der Vorgänger v angibt, für die das Paar (v, w) unter allen endenden identischen Wortpaaren die beste Bewertung hatte. Besitzt nun das neue zum Zeitpunkt t endende Wortpaar (v, w) eine bessere Bewertung als das zum Zeitpunkt $t-1$ geendete (und um Silence verlängerte) Worthypothesenpaar, so wird der Zähler der neuen Hypothese um eine Einheit vergrößert, während der Zähler der alten Hypothese dekrementiert wird. Ist die Bewertung dagegen schlechter (d.h. größer), bleiben die Zählerstände unverändert.

$$c(w;t) := \Big|\Big\{v \;:\; U(\tau(t;w);v) - \log p(w) + h(w;\tau(t;w),t) \\ < \; P(t-1;v,w) + h(\text{sil}, t-1, t)\Big\}\Big|$$

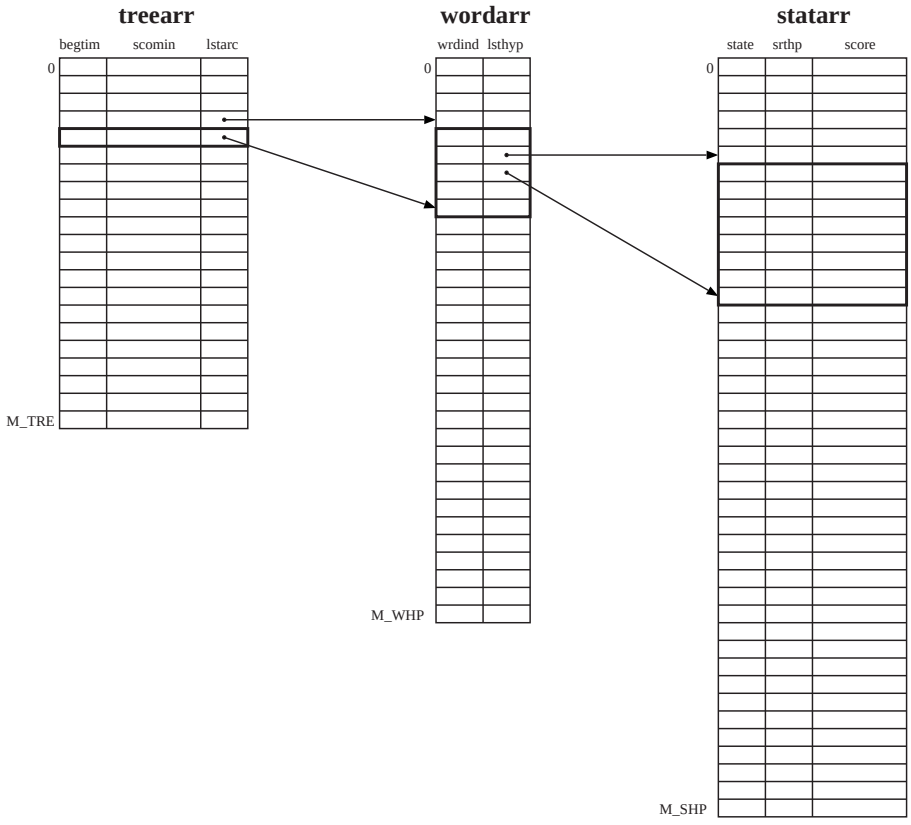


Abbildung 4.5: Datenstruktur für die zeitabhängige Baumsuche

Unter den zum Zeitpunkt t endenden Hypothesen werden nun all jene in den Wortgraphen aufgenommen, deren Zählerstand $c(w;t)$ größer null ist. Abschließend wird die Liste der besten endenden Wortpaare aktualisiert.

Wortgraph-Pruning

Die mit den Vergleichen der Bewertungen einzelner Worthypothesenpaare einhergehende Modifikation von Zählerständen kann nun dazu führen, daß die $c(w;t)$-Akkumulatoren nach Abschluß des Suchprozesses für einige, bereits in den Wortgraphen eingefügte Hypothesen wieder gleich null sind. Wird nun die Zeitspanne, die zwischen zwei zu vergleichenden Wortpaarhypothesen liegen darf, nach oben durch ein Δ_t beschränkt, so kann der $c(w;t)$-Zählerstand als neues Pruning-Kriterium zur Reduktion des Wortgraphen herangezogen werden. Hierbei werden dann alle Hypothesen aus dem Wortgraphen entfernt, deren $c(w;t)$-Zähler einen Stand von null aufweisen. Auf diese Weise wird ein großer Anteil der zur Pfadvervielfachung beitragenden redundanten Kanten wieder aus dem Wortgraphen entfernt. Die Wahl des Δ_t ist abhängig von der akustischen Modellierung und der zeitlichen Auflösung des akustischen Signals. Für die im Rahmen dieser Arbeit verwendeten Ganzwortmodelle wurde $\Delta_t = 8$ gewählt.

Aufbau und Struktur der Wortgraphen

Die Datenstruktur des Wortgraphen ist in Abbildung 4.6 dargestellt. Für eine zum Zeitpunkt t endende Worthypothese w werden die folgenden Größen gespeichert:

$$
\begin{aligned}
\texttt{wrdtrac[trb].begtim} &:= \tau(t;w)+1 \\
\texttt{wrdtrac[trb].endtim} &:= t \\
\texttt{wrdtrac[trb].locsco} &:= h(w;\tau(t;w),t) \\
\texttt{wrdtrac[trb].scomin} &:= G(u_1^{n-1},w;t) \\
\texttt{wrdtrac[trb].ftrb} &:= c(w;t)
\end{aligned}
$$

Die Sequenz von Zeigern `timtrac[`t_e`-1].btrb+1` bis `timtrac[`t_e`].btrb` ermöglicht dabei den Zugriff auf alle zum Zeitpunkt t_e endenden Worthypothesen, sortiert nach der Reihenfolge ihrer Bewertungen $G(u_1^{n-1},w;t)$.

Zur Berechnung der generalisierten FB-Wahrscheinlichkeiten muß der Wortgraph zusätzlich um Vorwärtsverweise erweitert werden, um einen Zugriff auf alle im Wortgraphen enthaltenen Hypothesen mit identischer Startzeit zu ermöglichen. Hierzu werden die Hypothesen des Wortgraphen nach Startzeiten t_a sortiert und für gleiche Startzeiten in aufsteigender Reihenfolge nach ihren Bewertungen $G(u_1^{n-1},w;t)$. Zur Sortierung bezüglich der Startzeiten bietet sich ein Histogramm-Verfahren (*bucket sort*) mit linearer Laufzeit an. Für gleiche Startzeiten können die Hypothesen dann mit einem schnellen Sortierverfahren (z.B. *heapsort*) bezüglich ihrer Bewertungen in aufsteigender Reihenfolge geordnet werden. Die Vorwärtszeiger `ftrb` werden nun so initialisiert, daß jeder Zeiger auf sich selbst verweist. Anschließend werden die Hypothesen des Wortgraphen wieder nach Endzeiten sortiert (Histogramm-Verfahren) und für gleiche Endzeiten wie zuvor in aufsteigender Reihenfolge bezüglich der Bewertungen $G(u_1^{n-1},w;t)$ (*heapsort*). Zur Korrektur der Vorwärtsverweise müssen nun noch sämtliche `ftrb`-Zeiger invertiert werden, d.h. es sind jeweils Quelle und Ziel eines jeden Zeigers zu vertauschen. Die Vertauschung kann *in situ* erfolgen, wenn die Reihenfolge der zu modifizierenden Einträge durch die `ftrb`-Zeigerfolge

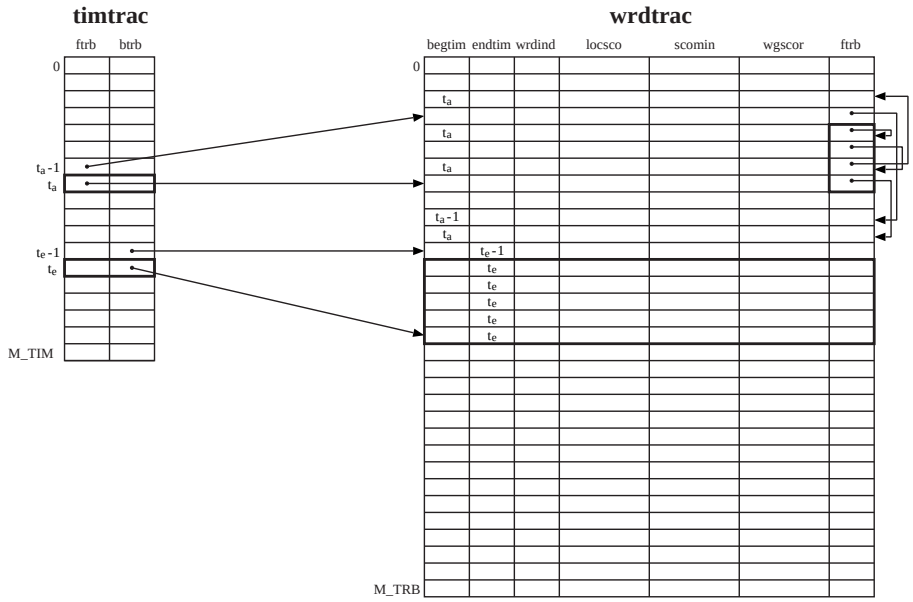


Abbildung 4.6: Datenstruktur für Wortgraphen

bestimmt wird. Diese Folge kann aus mehreren Teilfolgen von jeweils ringförmig verketteten Zeigern (sogenannten Partitionen) bestehen. Jede dieser Partitionen wird nun gesondert abgearbeitet. Um die nächste, noch zu bearbeitende Partition aufzufinden, müssen bereits modifizierte Einträge markiert werden. Der Zugriff auf die Komponenten aller Hypothesen des Wortgraphen mit gleicher Beginnzeit t_a ergibt sich dann durch die folgende indirekte Adressierung (hier exemplarisch für den Worthypothesenindex dargestellt):

$$\begin{array}{c} \texttt{wrdtrac[wrdtrac[timtrac[}t_a\texttt{-1].ftrb+1].ftrb].wrdind} \\ \vdots \\ \texttt{wrdtrac[wrdtrac[timtrac[}t_a\texttt{].ftrb].ftrb].wrdind} \end{array}$$

Der Anteil der Hypothesen mit gleicher Startzeit bzw. Endzeit ist gemessen an der Zahl aller im Wortgraphen enthaltenen Hypothesen nur sehr klein, so daß der Aufwand für das Sortieren nach Bewertungen als konstant angesehen werden kann. Die Komplexität für die Bereitstellung der Vorwärtsverweise wird somit durch das Histogramm-Sortierverfahren bestimmt und erfolgt daher im Mittel in linearer Zeit $\mathcal{O}(T)$.

Einfügen der gesprochenen Wortfolge in den Wortgraphen

Speziell für den Fall des MMI-Kriteriums ist es erforderlich, daß die korrekte Zeitanpassung der gesprochenen Wortfolge als Satzhypothese im Wortgraphen enthalten ist. Nun wäre es möglich, im Anschluß an die Suche die im Wortgraphen fehlenden Hypothesen der gesprochenen Wortsequenz nachträglich hinzuzufügen. Die Hypothesen der gesprochenen Wortfolge können jedoch auch dort, wo erforderlich, *während* der Suche in die Menge der aktiven Hypothesen aufgenommen werden. Im Unterschied zur ersten Variante besitzt die zweite Methode den Vorteil, daß die gesprochene Wortsequenz auch bei Suchfehlern so stets die Fokussierung des Suchraumes mitbestimmt.

Die zeitliche Segmentierung sowie die Bewertungen der gesprochenen Wortfolge werden aus der Alignierung der akustischen Vektoren X_r auf die Zustände des zur Wortsequenz W_r gehörenden Automaten gewonnen. Um vergleichbare Bewertungen zu erhalten, müssen die Rekombinationsmöglichkeiten des Such-Algorithmus und der Zeitanpassung (für eine gegebene Wortfolge) übereinstimmen. Insbesondere bedeutet dies, daß das Sprachmodell nun auch bei der erzwungenen Zeitanpassung explizit berücksichtigt werden muß[3].

Redundante Silence-Kanten

Trotz der Wortpaarapproximation und dem Wortgraphpruning führte gegenüber dem *Corrective Training* die Verwendung von Wortgraphen zur Berechnung der Posterior-Wahrscheinlichkeiten in einem ersten Ansatz zunächst zu deutlichen Verlusten in der Performanz des diskriminativen Trainings. Als Ursache konnte das Vorkommen all jener Silence-Hypothesen $(\mathrm{sil}; t_a, t_e)$ ausgemacht werden, für die ein alternativer, aus Silence-Kanten bestehender Weg existierte, der die Knoten der Zeitpunkte t_a und t_e miteinander verband. Definiert man die Länge eines Weges in einem Graphen als Anzahl n der Kanten, aus denen sich der Weg zusammensetzt, so heißt eine Silence-Kante $(\mathrm{sil}; t_a, t_e)$ *redundant*, wenn es einen längeren, ausschließlich aus Silence-Kanten bestehenden Weg gibt, der die Knoten der Zeitpunkte t_a und t_e miteinander verbindet. Ähnlich wie die Mehrfachvorkommen identischer Wortfolgen verursachen die redundanten Silence-Kanten eine Vervielfachung von Satzhypothesen, die sich hauptsächlich in der zeitlichen Segmentierung der Silence-Hypothesen unterscheiden. In Abbildung 4.7 ist dieser Sachverhalt dargestellt. Bezeichnen W_a und W_e wieder diejenigen Worthypothesenfolgen, die dem Zeitpunkt t_a vorausgehen bzw. die auf den Zeitpunkt t_e folgen, dann ergeben sich die Wortfolgen für die Hypothetisierung des Wortes w_1 in den Zeitgrenzen t_a und t_e zu W_a, w_1, W_e. Im Unterschied hierzu gibt es jedoch drei Möglichkeiten, eine Sprechpause im gleichen Zeitintervall zu hypothetisieren:

$$\begin{array}{ll} ① & W_a, \mathrm{sil}_1, \mathrm{sil}_5, W_e \\ ② & W_a, \mathrm{sil}_3, \mathrm{sil}_2, W_e \\ ③ & W_a, \mathrm{sil}_3, \mathrm{sil}_4, \mathrm{sil}_5, W_e \end{array}$$

Aufgrund der Normierungsbedingung müssen sich nun zu jedem Zeitpunkt t die Wortwahrscheinlichkeiten $p_{[t_a,t_e]}(w|X_r)$ sämtlicher Hypothesen w mit Beginnzeit $t_a \leq t$ und Endzeit $t_e \geq t$ zu 1 summieren:

$$\forall\, t: \sum_{\{w \in \mathcal{M}_r | t_a \leq t \leq t_e\}} p_{[t_a,t_e]}(w|X_r) \equiv 1$$

Die Vervielfältigung von Wegen im Wortgraphen bewirkt somit wiederum, daß ein zu großer Anteil der Wahrscheinlichkeitsmasse auf die Wortfolgen W_a, W_e, sowie auf die redundanten Silence-Kanten entfällt. Werden nun vor der Berechnung der generalisierten FB-Wahrscheinlichkeiten all jene Silence-Kanten aus dem Wortgraphen entfernt, die von einer Folge kurzer Silence-Kanten überbrückt werden, läßt sich dieses Problem umgehen. Die zeitabhängig organisierte Suche ermöglicht jedoch auch eine weitaus elegantere Lösung, die mit einer Modifikation des Pausen-Modells verbunden ist.

[3] Für das konventionelle ML-Training ist die Berücksichtigung des Sprachmodells in der Viterbi-Approximation unkritisch, da der Logarithmus der Sprachmodellwahrscheinlichkeit als konstanter, additiver Offset die Entscheidungen der dynamischen Programmierung nicht beeinflußt.

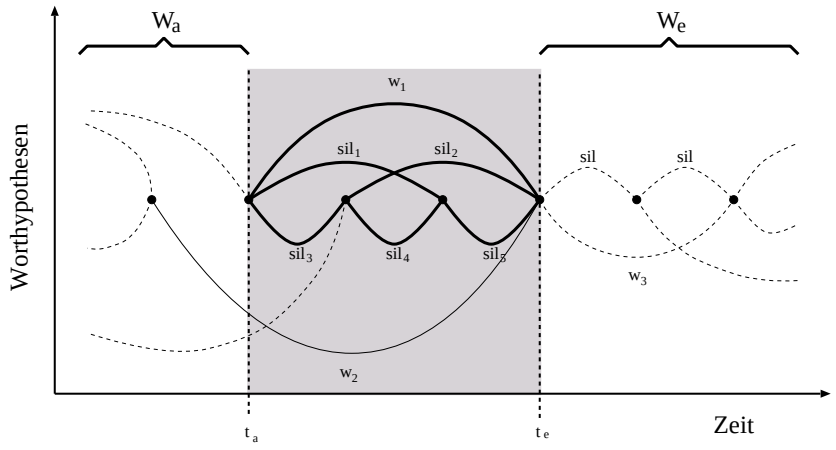


Abbildung 4.7: Wortgraph mit redundanten Silence-Kanten

Pausen-Modell im diskriminativen Training

Üblicherweise wird das Pausen-Modell in der Spracherkennung durch einen einzelnen Zustand realisiert, wobei durch das wiederholte Verharren in diesem Zustand Phasen unterschiedlich lang anhaltender Sprechpausen erfaßt werden können. Wird nun in der Suche die *loop*-Transition als Zustandsübergang für das Pausen-Modell verboten, so kann jede Silence-Hypothese nur noch genau einen akustischen Vektor erklären. Länger andauernde Sprechpausen können dann durch eine Folge unmittelbar aufeinander treffender Silence-Hypothesen beschrieben werden. In Kombination zu dieser Modifikation des Pausen-Modells wurde die Suche ferner so realisiert, daß Silence-Hypothesen grundsätzlich nicht gepruned werden dürfen. Jeder generierte Wortgraph enthält somit eine durchgehende Folge sehr kurzer Silence-Hypothesen (*Silence-Frames*). Abbildung 4.8 veranschaulicht diesen Zusammenhang. Um die Anzahl der Kanten im Wortgraphen zu reduzieren, werden nach Abschluß des Suchprozesses in einer Nachverarbeitungsphase (*post processing*) alle maximalen Teilfolgen von Silence-Kanten bestimmt, die weder von endenden noch beginnenden Worthypothesen unterbrochen werden, und durch jeweils eine „große“ Silence-Hypothese ersetzt.

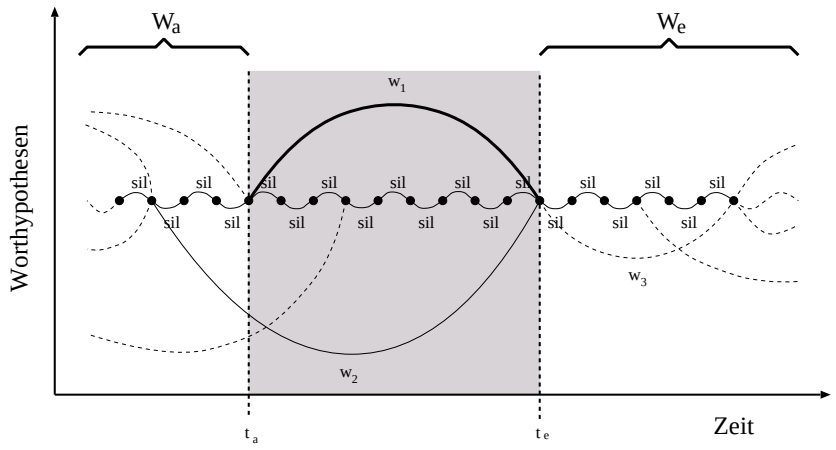


Abbildung 4.8: Wortgraph ohne redundante Silence-Kanten

4.3 Verfahren zur Beschleunigung der Suche

Gegenüber dem konventionellen ML-Ansatz benötigen die diskriminativen Verfahren ein Vielfaches der Rechenzeit, wobei die Bestimmung der konkurrierenden Hypothesen mittels eines Erkennungsschrittes den Hauptanteil ausmachen. Eine Reduktion des Berechnungsaufwandes für die Erkennung trägt daher zu einer erheblichen Verkürzung der Laufzeiten im Training bei. Aus diesem Grund wurden zwei Verfahren zur Beschleunigung der Suche entwickelt:

- die *überwachte, dynamisch fokussierte Suche* und
- die *Einschränkung der Erkennung auf vorherige Wortgraphen.*

Beide Verfahren nutzen die Bekanntheit der gesprochenen Wortfolge im Training aus und ermöglichen hierdurch eine schärfere Fokussierung des Suchraumes. Die verringerte Anzahl erforderlicher Abstandsberechnungen trägt somit zu einer deutlichen Abnahme der Laufzeiten bei. Die Verfahren arbeiten unabhängig voneinander und können daher problemlos kombiniert werden.

4.3.1 Überwachte, dynamisch fokussierte Suche

Die „überwachte, dynamisch fokussierte Suche“ basiert auf einer äußerungsspezifischen Einstellung des Schwellwertes für das akustische Pruning. Jede Erkennung wird zunächst mit einem sehr kleinen Schwellwert gestartet, der gerade ausreicht, um Suchfehler bei etwa 90% aller Trainingsäußerungen zu vermeiden. Für die verbleibenden 10% wird die Suche erneut durchgeführt, wobei der Pruning-Schwellwert sukzessive vegrößert wird. Als Kriterium für das Auftreten eines Suchfehlers gilt der Vergleich der Satzwahrscheinlichkeiten zwischen gesprochener und bester erkannter Wortfolge $p_\lambda(W_r, X_r)$ bzw. $p_\lambda(W_{\text{best}}, X_r)$. Ist nun die Satzwahrscheinlichkeit der besten erkannten Wortfolge kleiner als die Satzwahrscheinlichkeit der gesprochenen Äußerung, liegt ein Suchfehler vor, da die gesprochene Wortfolge offensichtlich die akustischen Beobachtungen besser zu erklären vermag. In diesem Fall wird der akustische Pruning-Schwellwert größer gewählt und die Suche erneut gestartet. Gilt dagegen $p_\lambda(W_{\text{best}}, X_r) \leq p_\lambda(W_{\text{best}}, X_r)$, wird das Ergebnis als Suchfehlerfrei angenommen. Die Wortsequenz W_{best} bildet dann unter den Elementen $W \in \mathcal{M}_r$ die beste konkurrierende Wortfolge.

4.3.2 Einschränkung der Erkennung auf Wortgraphen

In einem ersten Ansatz wurde zur Reduktion des Berechnungsaufwandes für ein diskriminatives Training unter Verwendung von Wortgraphen ab der zweiten Trainingsiteration eine akustische Neubewertung der Worthypothesen innerhalb der aus der ersten Iteration ermittelten Zeitgrenzen durchgeführt. Untersuchungen zu diesem Verfahren erbrachten jedoch keine oder nur geringe Verbesserungen. Die Ursache für den nur mäßigen Performanzgewinn ist in erster Linie auf die Modellannahme zurückzuführen, daß die Zeitgrenzen der Hypothesen eines Wortgraphen in jeder Iteration unverändert bleiben. Aus diesem Grund wurde eine Methode entwickelt, die die zu startenden Hypothesen auf einen aus der vorherigen Iteration stammenden Wortgraphen einschränkt, gleichzeitig aber die Wortgrenzeninformation relaxiert. So können zu jedem Zeitpunkt τ nur diejenigen Hypothesen gestartet werden, deren Beginnzeit im alten Wortgraphen in einer Umgebung von

τ liegen. Diese Umgebung wird durch das Intervall $[\tau - \Delta\tau, \tau + \Delta\tau]$ definiert (vgl. Abbildung 4.9). Die Einschränkung auf die hierbei ermittelten Nachfolgehypothesen reduzieren den Suchraum und führen damit zu einer erheblichen Beschleunigung der Suche. Insbesondere erlaubt die Methode auch die Erkennung neuer Wortfolgen, die als Satzhypothesen im ursprünglichen Wortgraphen nicht enthalten sind.

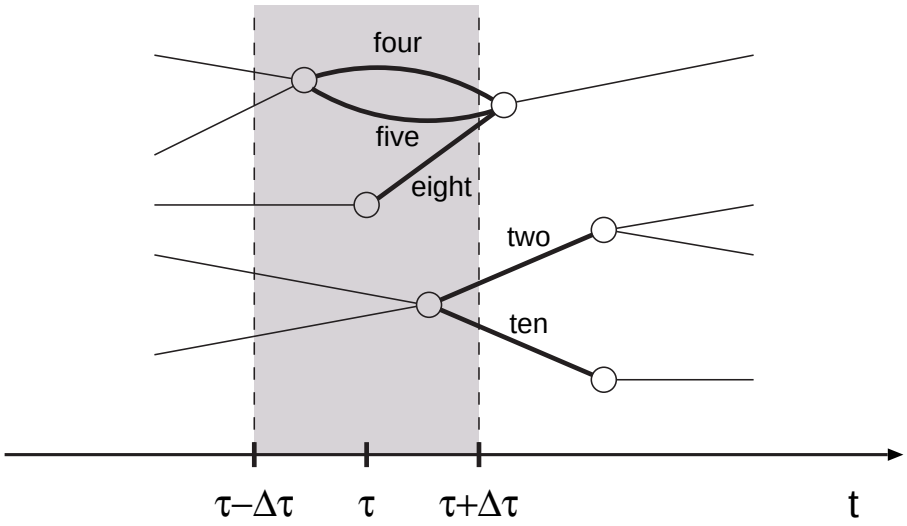


Abbildung 4.9: Ausschnitt eines Wortgraphen mit Worthypothesen, deren Beginnzeiten in das Intervall $[\tau - \Delta\tau, \tau + \Delta\tau]$ fallen.

4.3.3 Laufzeitverhalten

In den folgenden Tabellen sind die Laufzeitmessungen zur dynamisch fokussierten Suche (Tabelle 4.2) sowie zur Kombination von dynamisch fokussierter Suche und eingeschränkter Wortgrapherkennung (Tabelle 4.3) für ein MMI-Training auf dem SieTill-Korpus aufgeführt[4]. Der Eintrag in der Rubrik „akustischer Schwellwert“ gibt jeweils den initialen akustischen Pruningschwellwert an, mit dem die überwachte, dynamisch fokussierte Suche gestartet wurde. Zu jedem Schwellwert wurde jeweils ein MMI-Training über 20 Iterationen durchgeführt wobei für das Training zur Kombination der beiden Beschleunigungsverfahren ab der zweiten Iteration zusätzlich die eingeschränkte Wortgrapherkennung verwendet wurde. Die Spalten WER[%] und SER[%] geben jeweils die prozentuale Wortfehlerrate (*word error rate*) bzw. Satzfehlerrate (*string error rate*) derjenigen Iteration an, die im Verlauf des Trainings die kleinste Fehlerrate produzierte. Da die eingeschränkte Wortgrapherkennung eine sehr scharfe Fokussierung des Suchraumes bewirkt, ist hier für den Fall sehr dünner Wortgraphen eine erhöhte Wahrscheinlichkeit für das Auftreten von Suchfehlern gegeben. Weil aber die gesprochene Wortfolge bei Verwendung des MMI-Kriteriums stets als Hypothese in dem einschränkenden Wortgraphen enthalten ist, können Suchfehler hier auch zur Dekodierung der tatsächlich gesprochenen Wortfolge führen. Dies war zum Beispiel für die Einstellung der Pruningschwellwerte auf die Größen 200 und 300 der Fall. So enthält die Tabelle 4.3 für diese Einträge kleinere Wortfehlerraten, obwohl eine volle Suche die in Tabelle 4.2 aufgeführten Fehlerraten bestätigen würde. Aus diesem Grund muß bei Verwendung der eingeschränkten Wortgrapherkennung ein hinreichend dichter Wortgraph verwendet werden, um diese Art der Suchfehler zu vermeiden.

[4]Eine Beschreibung des SieTill-Korpus findet sich in Abschnitt 8.1.

Die Spalte „WGD“ gibt jeweils die Dichte des Wortgraphen an, wobei die Dichte als das Verhältnis aus der Anzahl der gesprochenen Wörter sowie der Anzahl der im Wortgraphen befindlichen Kanten definiert ist. Wie aus den Tabellen 4.2 und 4.3 ersichtlich wird, ergeben sich bei der Verwendung überdichter Wortgraphen deutliche Verluste in der Performanz des Trainings (siehe hierzu die Fehlerraten zu den akustischen Pruningschwellwerten 400 und 500). Die Ursache hierfür ist in der großen Redundanz der produzierten Wortgraphen zu finden, die durch die dann häufiger auftretenden Pfadvervielfältigungen eine starke Verzerrung bei der Berechnung der generalisierten FB-Wahrscheinlichkeiten hervorrufen.

Trotz des vergleichsweise kleinen Umfangs des verwendeten Vokabulars von 11 Ziffern, ergeben sich deutliche Gewinne im Echtzeitfaktor (RTF) vor allem für die dichten Wortgraphen. Dieses Ergebnis macht die eingeschränkte Wortgrapherkennung insbesondere für den Einsatz des diskriminativen Trainings bei großem Vokabular interessant.

Tabelle 4.2: Laufzeitmessungen für ein MMI-Training auf dem SieTill-Trainingskorpus unter ausschließlicher Verwendung der überwachten, dynamisch fokussierten Suche.

SieTill: Beschleunigung der Suche								
Korpus	LDA	Krit.	akustischer Schwellwert	WGD	del - ins - sub	WER[%]	SER[%]	RTF
Train.	**ja**	MMI	200	7.2	279- 99 -560	2.19	6.06	0.197
			300	18.6	271- 87 -540	2.10	5.78	0.354
			400	56.4	262-122-608	2.31	6.41	0.645
			500	127.4	247-183-638	2.49	6.87	1.141

Tabelle 4.3: Laufzeitmessungen für ein MMI-Training auf dem SieTill-Trainingskorpus unter Verwendung der überwachten, dynamisch fokussierten Suche in Kombination mit der eingeschränkten Wortgrapherkennung.

SieTill: Beschleunigung der Suche								
Korpus	LDA	Krit.	akustischer Schwellwert	WGD	del - ins - sub	WER[%]	SER[%]	RTF
Train.	**ja**	MMI	200	7.2	275- 93 -549	2.14	5.93	0.105
			300	18.5	271- 87 -539	2.09	5.77	0.163
			400	63.7	262-122-608	2.31	6.41	0.381
			500	144.2	247-183-638	2.49	6.87	0.821

4.4 Suche im Baum-Welch Training

Wie in der Viterbi-Näherung können auch im diskriminativen Baum-Welch Training die generalisierten FB-Wahrscheinlichkeiten $\gamma_{r,t}(s)$ unter Verwendung eines Suchprozesses geschätzt werden. Hierzu wird $\gamma_{r,t}(s)$ wieder in eine Summe gewichteter, bedingter FB-Wahrscheinlichkeiten zerlegt:

$$\begin{aligned}\gamma_{r,t}(s) &= \sum_{W} p_\lambda(W|X_r) \cdot \gamma_{r,t}(s|W) \\ &= \sum_{W} p_\lambda(W|X_r) \cdot p(s_t = s|X_r, W) \\ &= \frac{1}{p_\lambda(X_r)} \cdot \sum_{W} \sum_{[s_1^{t-1}, s_t=s, s_{t+1}^{T_r}]|W} \prod_{\tau=1}^{T_r} p_\lambda(s_\tau, x_\tau|s_{\tau-1})\end{aligned}$$

Die Wortfolgen W können nun jeweils in eine Sequenz $W = (W_a, w, W_e)$ aufgespalten werden:

$$= \frac{1}{p_\lambda(X_r)} \cdot \sum_{(W_a,w,W_e)} \sum_{[s_1^{t-1}, s_{w(t)}=s, s_{t+1}^{T_r}]|(W_a,w,W_e)} \prod_{\tau=1}^{T_r} p_\lambda(s_\tau, x_\tau|s_{\tau-1})$$

Dabei ist zu beachten, daß der Zustand s zum Zeitpunkt t in dem mit w assoziierten Wortautomaten hypothetisiert wird. Da die Grenzzeiten einer Worthypothese im BW-Training nicht eindeutig bestimmt sind, müssen alle zeitlichen Segmentierungen der Wortfolgen (W_a, w, W_e) berücksichtigt werden.

Hierzu kann die Summe über die Wortsequenzen W_a, w, W_e nun wieder in den Anteil derjenigen Wortfolgen (W_a, w) zerlegt werden, deren Zeitanpassungspfade $[s_1^{t-1}, s_{w(t)} = s]$ zum Zeitpunkt t im Zustand s des Wortautomaten w münden, sowie in den Anteil der Wortfolgen (w, W_e), deren Zeitanpassungspfade $[s_{w(t)} = s, s_{t+1}^{T_r}]$ zum Zeitpunkt t aus dem Zustand s im Wortautomaten w herausführen. Damit ergibt sich für $\gamma_{r,t}(s)$ die folgende Form:

$$\begin{aligned}\gamma_{r,t}(s) = \frac{1}{p_\lambda(X_r)} \cdot & \underbrace{\left[\sum_{(W_a,w)} \sum_{[s_1^{t-1}, s_{w(t)}=s]|(W_a,w)} \prod_{\tau=1}^{t} p_\lambda(x_\tau, s_\tau|s_{\tau-1}) \right]}_{\equiv a_{r,t}(s;w)} \cdot \\ & \underbrace{\left[\sum_{(w,W_e)} \sum_{[s_{w(t)}=s, s_{t+1}^{T_r}]|(w,W_e)} \prod_{\tau=t+1}^{T_r} p_\lambda(x_\tau, s_\tau|s_{\tau-1}) \right]}_{\equiv b_{r,t}(s;w)}\end{aligned} \tag{4.1}$$

Der erste, nach dem Normierungsfaktor $1/p_\lambda(X_r)$ stehende Term entspricht dabei der iterativen Form der in Abschnitt 3.4.1 eingeführten rekursiven Definition für die *Forward*-Wahrscheinlichkeit. Entsprechend beschreibt der Term in der letzten Zeile von Gleichung (4.1) die iterative Form der *Backward*-Wahrscheinlichkeit.

Um nun all diejenigen Zeitanpassungspfade zur Worthypothese w zu separieren, die zum Zeitpunkt t den Zustand $s_t = s$ hypothetisieren, kann die Gleichung 4.1 wie folgt umgeschrieben werden:

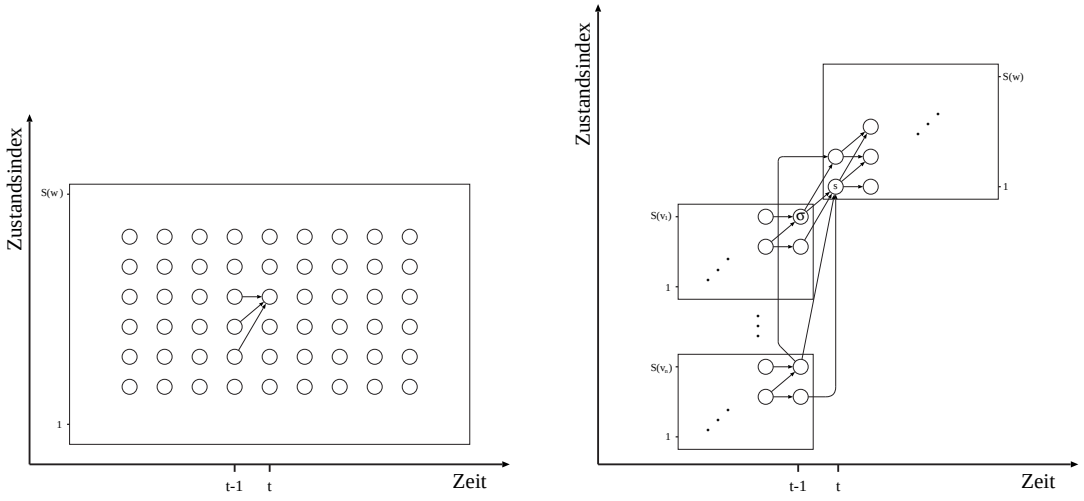
$$\gamma_{r,t}(s) = \frac{1}{p_\lambda(X_r)} \cdot \sum_{\substack{t_a,t_e \\ t_a \leq t \leq t_e}} \sum_{w\,:\,t_a(w)=t_a \wedge t_e(w)=t_e} \sum_{[s_{t_a}^{t-1}, s_t=s, s_{t+1}^{t_e}]|w}$$

$$\sum_{W_a\,:\,t_e(W_a)=t_a-1} \sum_{[s_1^{t_a-1}]|W_a} \sum_{W_e\,:\,t_a(W_e)=t_e+1} \sum_{[s_{t_e+1}^{T_r}]|W_e} \prod_{\tau=1}^{T_r} p_\lambda(s_\tau, x_\tau | s_{\tau-1})$$

Hierbei verläuft die Summe über all jene Grenzeiten $t_a \leq t \leq t_e$, in denen das Wort w hypothetisiert werden kann. Die Aufspaltung der Zeitanpassungspfade in einen Anteil für die Vorgänger W_a sowie für die Nachfolger W_e legt zur Bestimmung der generalisierten FB-Wahrscheinlichkeiten die Produktion eines wortbasierten Zustandsgraphen nahe.

Produktion von Zustandsgraphen

Im diskriminativen BW-Training wird während eines Suchprozesses ein wortbasierter Zustandsgraph produziert. Dieser Zustandsgraph gibt den gesamten Suchraum nach dem akustischen Pruning wieder und kann im Verlauf der Vorwärtsrekombination generiert werden. Im Unterschied zur Viterbi-Näherung müssen für die Berechnung der Größen $a_{r,t}(s;w)$ die *Forward*-Wahrscheinlichkeiten sämtlicher Vorgängerzustände σ akkumuliert werden, von denen der Zustand s im Wort w zu erreichen ist. Für die Inter-Wortrekombination bedeutet dies insbesondere, daß die *Forward*-Wahrscheinlichkeiten aller erreichbaren Endzustände σ_e von Vorgängerwörtern v in die Größe $a_{r,t}(s;w)$ einfließen. Abbildung 4.10 illustriert die Rekombinationsmöglichkeiten im Wortinneren sowie an den Wortgrenzen.



(a) Intra-Wort-Rekombination

(b) Inter-Wort-Rekombination

Abbildung 4.10: Pfad-Rekombinationen in der Baum-Welch Suche

In einer zweiten Stufe werden anschließend die Rückwärtsschritte zur Bestimmung der generalisierten FB-Wahrscheinlichkeiten durchgeführt. Dabei werden die Pfade aus der Rückwärtsrekombination auf den aus der Vorwärtsrekombination ermittelten Suchstrahl eingeschränkt. Die Rekombination im Wortinneren sowie an den Wortgrenzen erfolgt dabei ähnlich wie im Vorwärtsschritt. Der Algorithmus zur Bestimmung der generalisierten FB-Wahrscheinlichkeiten im diskriminativen BW-Training ist in Tabelle 4.4 skizziert. Die zugehörige Datenstruktur ist der Abbildung 4.12 zu entnehmen.

Zu Beginn werden die Anfangszustände sämtlicher Wörter w des Vokabulars hypothetisiert. Jede dieser Zustandshypothesen initiiert dabei den Beginn eines Suchstrahls. Innerhalb eines Wortautomaten erfolgt die Vorwärtsrekombination eines Zustandes s dann in der für das Baum-Welch Training üblichen Weise durch Akkumulation der *Forward*-Wahrscheinlichkeiten $a_{r,t-1}(\sigma; w)$ über alle Vorgängerzustände σ:

$$a_{r,t}(s; w) = p_\lambda(x_{r,t}|s, w) \cdot \left(\sum_\sigma a_{r,t-1}(\sigma, w) \cdot p(s|\sigma) \right)$$

Wird zu einem Zeitpunkt $t-1$ der Endzustand σ_e eines Wortautomaten hypothetisiert, können im darauf folgenden Zeitschritt t wieder alle nicht aktiven Anfangszustände von Wörtern w des Vokabulars gestartet werden. Jeder neu aktivierte Anfangszustand initiiert wiederum einen eigenen Suchstrahl. Dadurch ist es möglich, daß der Zustandsraum eines Wortautomaten mehrere parallele Suchstrahlen besitzt. Berühren sich zwei Suchstrahlen im Verlauf der Vorwärtsrekombination, werden sie zu einem gemeinsamen Strahl verschmolzen (vgl. Abbildung 4.11).

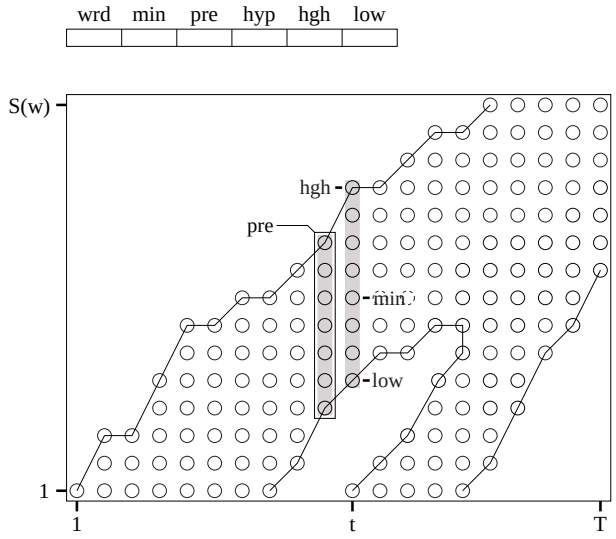


Abbildung 4.11: Suchraum im Baum-Welch Training für den Wortautomaten w

Für die Berechnung der *Forward*-Wahrscheinlichkeit eines Anfangszustandes s sind die Übergänge all jener Endzustände σ_e von Wörtern v zu berücksichtigen, die den Zustand s im Wort w erreichen können:

$$a_{r,t}(s;w) := p_\lambda(x_{r,t}|s,w) \cdot \Bigg(\sum_\sigma a_{r,t-1}(\sigma,w) \cdot p(s|\sigma) + \sum_v \sum_{\sigma_e} a_{r,t-1}(\sigma_e;v) \cdot p(s|\sigma) \cdot p(w|v) \Bigg)$$

Der Zugriff auf die zu einem Zeitpunkt t aktiven Hypothesen eines Wortautomaten erfolgt über das Feld `FB_Tim`. Jeder der auf das Feld `FB_Arc` verweisenden Zugriffsindizes `FB_Tim[`t`-1]+1`,...,`FB_Tim[`t`]` bezeichnet den gegenwärtig aktiven Teil (Fenster) des Suchraumes eines Wortautomaten. Die Komponente `wrd` gibt dabei den Worthypothesenindex des zugehörigen Wortautomaten an. Die Ausdehnung der Fensterbreite eines Suchstrahls wird durch die Komponenten `low` und `hgh` bestimmt. Dabei bezeichnet `low` den Zustandsindex am unteren Rand des Fensters und `hgh` entsprechend den Index des Zustandes am oberen Rand. `min` gibt den Zustandsindex an, der innerhalb des aktiven Fensters die größte *Forward*-Wahrscheinlichkeit besitzt. Die Komponente `pre` enthält einen Zugriffsindex auf jenen Eintrag des Feldes `FB_Arc`, der das zum Zeitpunkt `t-1` aktive Fenster zum gleichen Suchstrahl enthält. Wird der Suchstrahl an einer Wortgrenze zum Zeitpunkt t neu initiiert, besitzt die Komponente `pre` den Wert 0.

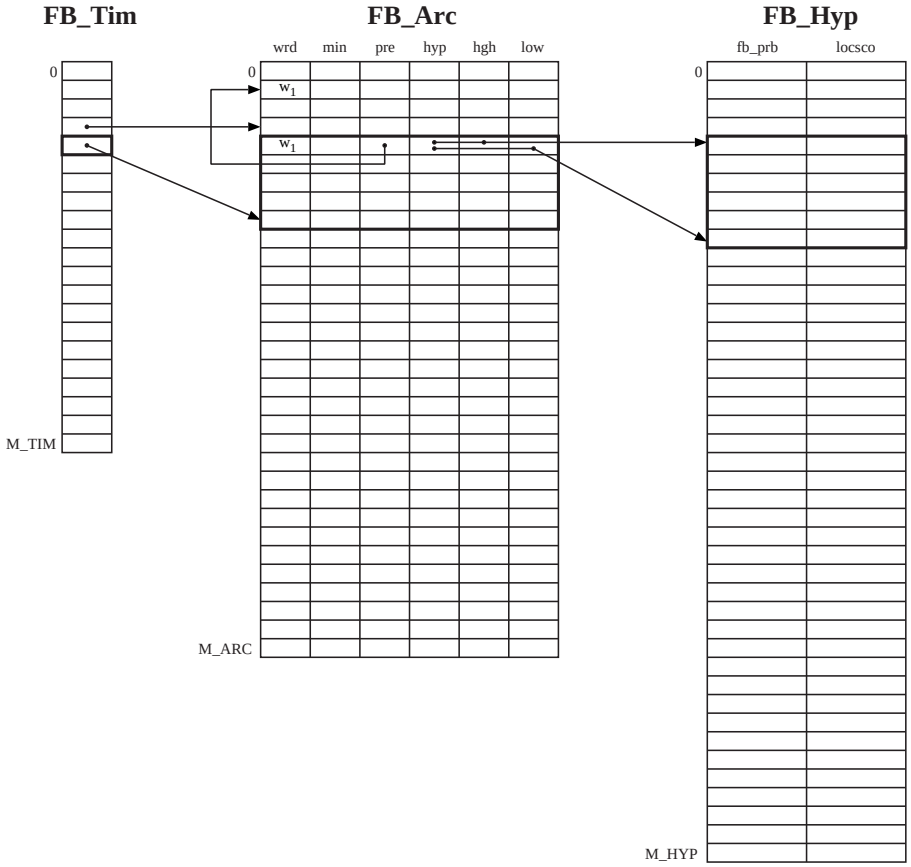


Abbildung 4.12: Datenstruktur für Zustandsgraphen

Tabelle 4.4: Algorithmus zur Bestimmung der generalisierten Forward-Backward-Wahrscheinlichkeiten im diskriminativen Baum-Welch Training

INITIALIZATION:

FORWARD RECOMBINATION: proceed over time t from left to right
determine all word hypotheses ending at timeframe $t-1$ proceed over all active arcs of timeframe $t-1$
ACOUSTIC LEVEL:
intra word recombination inter word recombination
start new beam iff word hypothesis is still active and lowest state index is greater than 1 start new word hypothesis iff at least one state-end hypothesis was still active in time frame $t-1$
compress FB-Hyp array melt two beams comming in contact to each other
calculate forward score: $-\log \sum_{s,w} a_{r,T_r}(s;w)$

BACKWARD RECOMBINATION: proceed over time t from right to left
determine all word hypotheses starting at timeframe $t+1$ calculate array of local backward scores for timeframe $t+1$ proceed over all active arcs of timeframe $t+1$
ACOUSTIC LEVEL: (the backward path will be embedded into the beam created by the forward path)
intra word recombination inter word recombination
compute and store unnormalized forward-backward scores update array of local backward scores
calculate backward score: $-\log \sum_{s,w} b_{r,0}(s;w)$

NORMALIZE FB-PROBABILITIES

Um für die Rückwärtsrekombinationen die Abstandsberechnungen nicht erneut durchführen zu müssen, werden innerhalb der Vorwärtsrekombination die Bewertungen für die Emissionswahrscheinlichkeiten der einzelnen Zustandshypothesen in dem Feld `FB_Hyp` unter der Komponente `locsco` gespeichert. Der Zugriff erfolgt über die Komponente `hyp` des Feldes `FB_Arc`, wobei die Fensterbreite `hgh`−`low`+1 die Zahl der abgelegten Zustandshypothesen angibt. Während der Rückwärtsrekombination werden die noch unnormierten generalisierten FB-Wahrscheinlichkeiten sukzessive bestimmt und unter der Komponente `fb_prb` gespeichert. Die eigentliche Normierung der FB-Wahrscheinlichkeiten erfolgt für alle im Feld `FB_Hyp` gespeicherten Zustandshypothesen erst nach Abschluß der Rückwärtsrekombination. Im Unterschied zur Suche für die Generierung von Wortgraphen in der

Viterbi-Näherung können bei der Produktion des wortbasierten Zustandsgraphen keine Redundanzen und damit keine Pfadvervielfältigungen auftreten. Die Schätzung der generalisierten FB-Wahrscheinlichkeiten ist damit wesentlich sicherer als bei Verwendung der Viterbi-Näherung.

Kapitel 5

Implementierung eines FB-Algorithmus für Wortgraphen

In diesem Kapitel wird die Implementierung des in Abschnitt 3.4.2 vorgestellten FB-Algorithmus für Wortgraphen beschrieben. Die Ausführungen beziehen sich dabei stets auf einen Wortgraphen, der zu einem in der Viterbi-Näherung durchgeführten Suchprozeß generiert wurde. Für die Bestimmung der Worthypothesen-Wahrscheinlichkeiten wird im folgenden die Verwendung eines Unigramm-Sprachmodells angenommen.

5.1 Berechnung der Wortwahrscheinlichkeiten

Gegeben sei ein zeitabhängig generierter Wortgraph, dessen Kanten gemäß Abschnitt 4.2 mit dem Worthypothesenindex w, der Beginn- und Endzeit t_a bzw. t_e, sowie dem negativen Logarithmus der wortbezogenen Emissionswahrscheinlichkeit $p_\lambda(x_{t_a}^{t_e}|w)$ und der Verbundwahrscheinlichkeit $p_\lambda(x_1^{t_e}, h_1^{n-1}, w)$ (zur besten Historie h_1^{n-1}) beschriftet sind.

Vor der eigentlichen Berechnung der FB-Wahrscheinlichkeiten werden zunächst die negativen Logarithmen sämtlicher wortbezogenen Emissionswahrscheinlichkeiten mit dem Wichtungsexponenten α skaliert. Die Bestimmung der *Forward*-Wahrscheinlichkeiten erfolgt dann im Anschluß hieran durch ein sukzessives Abarbeiten der im Wortgraphen enthaltenen Worthypothesen, wobei die Reihenfolge der Abarbeitung durch die Startzeiten der Worthypothesen bestimmt ist. Zu Beginn werden die noch unnormierten *Forward*-Wahrscheinlichkeiten sämtlicher Worthypothesen mit Beginnzeit $t_a = 1$ und Endzeit t_e als Produkt aus wortbezogener Emissionswahrscheinlichkeit und Sprachmodellwahrscheinlichkeit definiert:

$$\Phi_{r,t_e}(w) := p_\lambda^\alpha(x_{r\,t_a=1}^{\;t_e}|w) \cdot p^\alpha(w)$$

Für Hypothesen w mit Beginnzeit $t_a > 1$ und Endzeit t_e ergeben sich die ebenfalls noch unnormierten *Forward*-Wahrscheinlichkeiten dann aus der Summe der *Forward*-Wahrscheinlichkeiten sämtlicher zum Zeitpunkt $t_a - 1$ endenden Kanten sowie anschließender Multiplikation mit dem Produkt aus wortbezogener Emissions- und Sprachmodellwahrscheinlichkeit:

$$\Phi_{r,t_e}(w) = \left(\sum_{\{h\,:\,t_e(h)=t_a-1\}} \Phi_{r,t_a-1}(h) \right) \cdot p_\lambda^\alpha(x_{r\,t_a}^{\;t_e}|w) \cdot p^\alpha(w)$$

Die nachfolgende Tabelle faßt die wichtigsten Schritte des Algorithmus zur Berechnung der *Forward*-Wahrscheinlichkeiten für zeitabhängig generierte Wortgraphen zusammen:

Tabelle 5.1: Algorithmus zur Bestimmung der (noch unnormierten) *Forward*-Wahrscheinlichkeiten

procced over time t from left to right
procced over all word hypotheses starting at time frame t
procced over all arcs ending at time frame $t-1$
accumulate *forward* probabilities
add local score of actual word hypothesis store *forward* score

Die negativen Logarithmen der *Forward*-Wahrscheinlichkeiten werden jeweils unter der Komponente `scomin` in der Wortgraph-Struktur gespeichert (siehe Abbildung 4.5). In entsprechender Weise werden auch die *Backward*-Wortwahrscheinlichkeiten bestimmt, wobei für jede Worthypothese w mit Beginnzeit t_a und Endzeit t_e über die *Backward*-Wahrscheinlichkeiten $\widetilde{\Psi}_{r,t_e+1}(w)$ all jener Worthypothesen f zu summieren ist, die zum Zeitpunkt t_e+1 starten:

$$\widetilde{\Psi}_{r,t_a}(w) = p_\lambda^\alpha(x_{r\,t_a}^{t_e}|w) \cdot \left(\sum_{\{f\,:\,t_a(f)=t_e+1\}} \widetilde{\Psi}_{r,t_e+1}(f) \cdot p^\alpha(f) \right)$$

Die wesentlichen Schritte zur Berechnung der *Backward*-Wortwahrscheinlichkeiten sind in Tabelle 5.2 zusammengefaßt.

Tabelle 5.2: Algorithmus zur Bestimmung der (noch unnormierten) *Backward*-Wahrscheinlichkeiten

procced over time t from right to left
procced over all word hypotheses ending at time frame t
procced over all arcs starting at time frame $t+1$
accumulate *backward* probabilities
add local score of actual word hypothesis store *backward* score

Die negativen Logarithmen der *Backward*-Wahrscheinlichkeiten werden im Wortgraphen unter der Komponente `wgscor` gespeichert.

Der Normierungsfaktor $\overline{p}(X_r)$ ergibt sich nun aus der Summe der *Forward*-Wahrscheinlichkeiten aller zum Zeitpunkt $t = T_r$ endenden Kanten bzw. aus der Summe der *Backward*-Wahrscheinlichkeiten aller zum Zeitpunkt $t = 1$ startenden Kanten. In einem abschließenden Schritt werden dann die Wortwahrscheinlichkeiten aus dem Normierungsfaktor und der in den Feldern `scomin` bzw. `wgscor` gespeicherten Größen bestimmt. Da die wortbezogenen Emissionswahrscheinlichkeiten für jede Kante jedoch zweimal in die Berechnung der FB-Wahrscheinlichkeiten eingegangen sind, müssen die negativen Logarithmen der

Wortwahrscheinlichkeiten zusätzlich noch um die Größe $-\log p(x_{t_a}^{t_e}|w)$ vermindert werden. Die zuvor unter der Komponente `wgscor` gespeicherten negativen Logarithmen der *Backward*-Wahrscheinlichkeiten werden nun nicht mehr benötigt, so daß der Speicher für die Aufnahme der Wortwahrscheinlichkeiten zur Verfügung steht.

5.1.1 Numerik

Während der Bestimmung der *Forward*- und der *Backward*-Wahrscheinlichkeiten müssen die negativen Logarithmen der Größen Φ und $\widetilde{\Psi}$ wieder explizit in Wahrscheinlichkeiten umgerechnet werden, da sich die Summation im Logarithmus nicht mehr faktorisieren läßt:

$$
\begin{aligned}
-\log \Phi_{r,t_e}(w) &= -\log\left(p_\lambda^\alpha(x_{r\,t_a}^{\;t_e}|w)\cdot\sum_h \Phi_{r,t_a-1}(h)\cdot p^\alpha(w)\right)\\
-\log \widetilde{\Psi}_{r,t_a}(w) &= -\log\left(p_\lambda^\alpha(x_{r\,t_a}^{\;t_e}|w)\cdot\sum_f \widetilde{\Psi}_{r,t_e+1}(f)\cdot p^\alpha(f)\right)
\end{aligned}
$$

Aufgrund der Größenordnung der negativen Logarithmen, die in dem Bereich 10^4 - 10^6 liegen, ergibt sich jedoch ein numerisches Problem: Das Rechnen mit den zugehörigen Wahrscheinlichkeiten aus dem Intervall $[e^{-10^5},...,e^{-10^4}]$ übersteigt in der Regel den Darstellungsbereich einer Maschine. Vor der eigentlichen Summenbildung wird daher stets der Summand mit der größten Wahrscheinlichkeit bestimmt und vor die Summe gezogen. Werden mit $\mathrm{sco}_1,...,\mathrm{sco}_n$ die negativen Logarithmen der zu summierenden Wahrscheinlichkeiten $p_1,\dots p_n$ bezeichnet, dann gestaltet sich die Summenbildung wie folgt:

$$
\begin{aligned}
-\log\left(\sum_{i=1}^n \exp\left(-\mathrm{sco}_i\right)\right) &= -\log\left(\sum_{i=1}^n p_i\right)\\
&= -\log\left(p_{\max}\sum_{i=1}^n \frac{p_i}{p_{\max}}\right)\\
&= \mathrm{sco}_{\min} - \log\left(\sum_{i=1}^n \exp\left(\log p_i - \log p_{\max}\right)\right)\\
&= \mathrm{sco}_{\min} - \log\left(1 + \sum_{\substack{i=1\\ i\neq\min}}^n \exp\left(\mathrm{sco}_{\min} - \mathrm{sco}_i\right)\right)
\end{aligned}
\tag{5.1}
$$

Hierbei gilt:

$$
p_{\max} := \max\Big\{\, p_1,\dots,p_n \,\Big\} \qquad \text{und} \qquad \mathrm{sco}_{\min} := -\log p_{\max}
$$

Die Exponentiation erfolgt so nur noch für die wesentlich kleineren Score-Differenzen. Sollten dennoch vereinzelt Unterläufe für die Terme $\exp(\mathrm{sco}_{\min} - \mathrm{sco}_i)$ auftreten, so ist dies dadurch zu rechtfertigen, daß die zur Größe sco_i gehörende Wahrscheinlichkeit (für hinreichend kleine n) dann ohnehin keinen signifikanten Beitrag zu Summenbildung leisten kann.

5.2 Minimum Classification Error (MCE) auf Wortgraphen

Bei Verwendung des MCE-Kriteriums umfaßt die Menge der konkurrierenden Hypothesen sämtliche durch einen Wortgraphen repräsentierte Wortfolgen W, ausgenommen die gesprochene Wortfolge W_r. Damit nun die Bestimmung der Wortwahrscheinlichkeiten in der gleichen Weise wie beim MMI-Kriterium auf Wortgraphen erfolgen kann, müßte die gesprochene Wortfolge aus dem Wortgraphen entfernt werden. Im allgemeinen ist es jedoch nicht möglich, eine einzelne Satzhypothese aus einem Wortgraphen zu entfernen, ohne daß zugleich andere Satzhypothesen verloren gehen, denn Kanten der gesprochenen Wortfolge können Bestandteile anderer Hypothesen sein. Aus diesem Grund werden zur Bestimmung der Wortwahrscheinlichkeiten zunächst *alle* im Wortgraphen enthaltenen Satzhypothesen herangezogen. Die Satzwahrscheinlichkeit der gesprochenen Wortfolge wird dann nachträglich wieder abgezogen:

$$p_{[t_a,t_e]}(w|X_r) = \frac{\sum\limits_{\substack{\{W\in\mathcal{M}_r|W\neq W_r \\ \wedge w_{[t_a,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W)}{\sum\limits_{\{V\in\mathcal{M}_r|V\neq W_r\}} p_\lambda^\alpha(X_r,V)} = \frac{\sum\limits_{\substack{\{W\in\mathcal{M}_r| \\ w_{[t_a,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W) - \sum\limits_{\substack{\{W\in\mathcal{M}_r|W=W_r \\ \wedge w_{[t_a,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W)}{\sum\limits_{V\in\mathcal{M}_r} p_\lambda^\alpha(X_r,V) - \sum\limits_{\{V\in\mathcal{M}_r|V=W_r\}} p_\lambda^\alpha(X_r,V)}$$

Neben der besten Zeitanpassung der gesprochenen Wortfolge kann ein Wortgraph weitere Kopien der gesprochenen Sequenz mit abweichenden Grenzzeiten enthalten. Die Bewertungen dieser Kopien unterscheiden sich oftmals nur geringfügig von der Bewertung der besten Zeitanpassung. Für das MCE-Training ist es daher notwendig, *alle* Zeitanpassungen der gesprochenen Wortsequenz im Wortgraphen aufzufinden und zu markieren, um bei der anschließenden Berechnung der Wortwahrscheinlichkeiten die Summe der Verbundwahrscheinlichkeiten dieser Satzhypothesen wieder abzuziehen. Um die für die Praxis relevanten Zeitanpassungen der gesprochenen Wortfolgen im Wortgraphen zu identifizieren, wird eine N-Besten-Liste zu den im Wortgraphen enthaltenen Hypothesen erstellt. Die Größe dieser Liste muß so gewählt werden, daß bzgl. $p_\lambda(X_r, W_{\text{best}})$ sämtliche Sazthypothesen W mit nicht verschwindend geringer Satzwahrscheinlichkeit $p_\lambda(X_r, W)$ aufgenommen werden können. Für die im Rahmen dieser Arbeit durchgeführten Experimente zum MCE-Training war eine Liste der Größe $N = 100$ hinreichend, da die Satzwahrscheinlichkeiten der an den letzten Positionen der Liste auftretenden Hypothesen deutlich kleiner waren, als die Satzwahrscheinlichkeit der besten erkannten Wortfolge.

5.2.1 Effiziente N-Besten-Listen-Generierung

Für einen gegebenen Wortgraphen enthält jeder Eintrag der N-Besten-Liste eine Folge von Zeigern auf die Hypothesen des Wortgraphen. Jede dieser Zeigerfolgen beschreibt eine im Wortgraphen enthaltene Satzhypothese. Aufgrund der Verwendung eines Unigramm-Sprachmodells und der in Abschnitt 4.2 beschriebenen Sortierung des Wortgraphen sind die Worthypothesen monoton steigend bezüglich der Bewertung `scomin` angeordnet. Es ist daher möglich, die N-Besten-Liste mittels dynamischer Programmierung zu bestimmen, wobei die Zeitkomplexität für ein gegebenes N durch die bereits erfolgte Sortierung nur noch linear in der Anzahl der im Wortgraphen enthaltenen Worthypothesen ist.

Das Prinzip der N-Besten-Listen-Generierung basiert auf einer Präfixergänzung bereits in der Liste enthaltener Suffixe. Zu Beginn wird die N-Besten-Liste mit allen zum Zeitpunkt T_r endenden Hypothesen initialisiert. Die Hypothesen werden dabei in aufsteigender Reihenfolge bezüglich ihrer Bewertungen angeordnet, so daß die relative Position eines Eintrages (im folgenden das *Niveau* oder auch die *Höhe* genannt) mit der Größe der Satzwahrscheinlichkeit korrespondiert. In einer rückwärts über die Zeitachse laufenden Schleife werden nun die zu jedem Zeitpunkt t endenden Kanten bestimmt und zur Erweiterung der in der N-Besten-Liste befindlichen Suffixe herangezogen. Dabei werden nur solche Einträge in der N-Besten-Liste expandiert, die aufgrund ihrer Beginnzeit die zum Zeitpunkt t endenden Worthypothesen fortsetzen können. Die Präfixergänzung der übrigen Einträge wird solange suspendiert, bis ein entsprechender Zeitpunkt erreicht worden ist, für den die endenden Wortgraphhypothesen zur Expansion herangezogen werden können. Schlechter bewertete Hypothesenfolgen sinken im Verlauf der Abarbeitung auf ein niedrigeres Niveau herab und werden verworfen, falls die Liste keine weiteren Einträge mehr aufnehmen kann.

Die folgende Abbildung zeigt exemplarisch die Generierung einer N-Besten-Liste zu einem gegebenen Wortgraphen. Die Komponenten des Wortgraphen `locsco` und `scomin` sind dabei wie in Abschnitt 4.2 beschrieben definiert. Die Größe `sntsco` in der zugehörigen N-Besten-Liste gibt die akkumulierten `locsco`-Bewertungen der einzelnen Worthypothesen (unter Vernachlässigung des Sprachmodells) an, aus denen sich der Eintrag (d.h. die Satzhypothese) zusammensetzt.

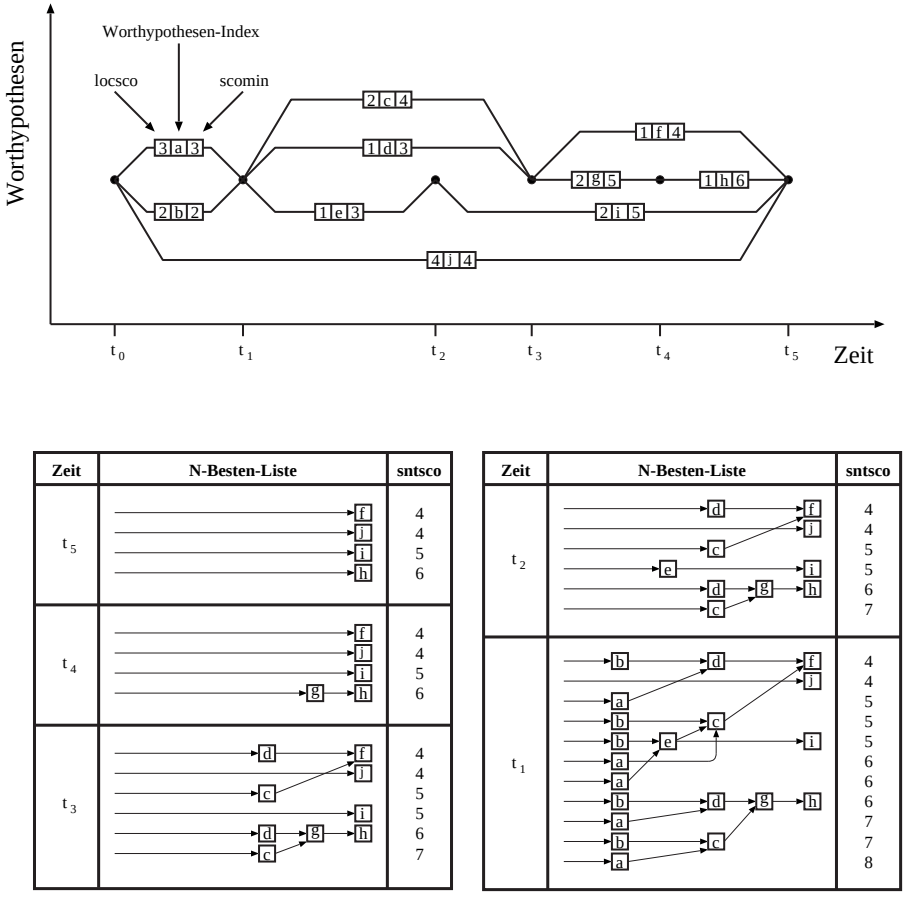


Abbildung 5.1: Beispiel für Wortgraph und zugehöriger N-Besten-Listen-Generierung

Kapitel 6

Lineare Merkmalstransformationen

Oftmals ist es möglich, die Performanz eines Spracherkenners durch Erweiterung der Beobachtungsvektoren um zusätzliche Merkmalskomponenten zu verbessern. Aufgrund der größer werdenden Dimension des Merkmalsraumes sowie der damit verbundenen Zunahme von Modellparametern und Berechnungsaufwand sind dieser Vorgehensweise in der Praxis jedoch enge Grenzen gesetzt. Sind die neuen Merkmale darüber hinaus mit den bisherigen Beobachtungsvektoren korreliert, macht sich zusätzlich die Tatsache bemerkbar, daß die Anzahl der Trainigsdaten, die benötigt werden, um den statistischen Klassifikator sicher zu schätzen, exponentiell mit der Dimension des Merkmalsraumes wächst (Stichwort: „*curse of the dimensionality*“ [Bellman 61]). Es ist daher sinnvoll, unter Verwendung von Merkmalstransformationen einen Unterraum zu bestimmen, der zum einen die Separierbarkeit der Klassen bestmöglichst erhält und zum anderen eine hinreichend stark reduzierte Dimension aufweist. Häufig verwendet man zur Merkmalstransformation eine lineare Abbildung φ, die zwischen der Merkmalsgewinnung (akustische Analyse) und der Klassifikation („Globale Suche“) angesiedelt ist:

$$\varphi \; : \; \left\{ \begin{array}{lcl} I\!R^D & \rightarrow & I\!R^{D'} \\ x & \mapsto & y = \varphi(x) \end{array} \right. \qquad \text{mit } D' \leq D$$

Eine solche Merkmalstransformation bildet die *lineare Diskriminanzanalyse* (LDA), die bereits seit Jahren erfolgreich in der Spracherkennung eingesetzt wird [Haeb-Umbach und Ney 92, Welling et al. 95].

6.1 Lineare Diskriminanzanalyse (LDA)

Die LDA ermöglicht das Auffinden einer linearen Abbildung, die sowohl die Ballungsgebiete der einzelnen Klassen kompakt hält als auch ihre Zentren voneinander entfernt [Schukat-Talamazzini]. Im Rahmen dieser Arbeit werden die Klassen mit den Zuständen der Markovmodelle identifiziert. Die zentralen Größen zur Bestimmung der LDA-Transformationsmatrix bilden die Intraklassen-Streuungsmatrix (*Within-Class-Scatter-Matrix*) **W** und die Interklassen-Streuungsmatrix (*Between-Class-Scatter-Matrix*) **B**. Beide Matrizen werden unter Verwendung einer gekennzeichneten Lernstichprobe bestimmt und sind wie

folgt definiert:

$$\mathbf{W} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \left(x_{r,t} - \mu_{s_{W_r}(t)}\right) \cdot \left(x_{r,t} - \mu_{s_{W_r}(t)}\right)^\top$$

$$\mathbf{B} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \left(\mu - \mu_{s_{W_r}(t)}\right) \cdot \left(\mu - \mu_{s_{W_r}(t)}\right)^\top$$

$$= \sum_{s} n_s \left(\mu - \mu_{s_{W_r}(t)}\right) \cdot \left(\mu - \mu_{s_{W_r}(t)}\right)^\top \qquad \text{mit } n_s = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \delta_{s,s_{W_r}(t)}$$

Die Intraklassen-Streuungsmatrix entspricht somit einer gepoolten, empirischen Kovarianzmatrix, die die Streuung der Beobachtungen um ihre jeweiligen Klassenzentren erfaßt. Die Interklassen-Streuungsmatrix mißt dagegen die Streuung der Klassenzentren μ_s um den globalen Mittelwert μ. Beide Matrizen bilden eine additive Zerlegung der Gesamtkovarianz (*Total-Scatter-Matrix*) $\mathbf{T}$:

$$\mathbf{T} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \left(x_{r,t} - \mu\right) \cdot \left(x_{r,t} - \mu\right)^T = \mathbf{W} + \mathbf{B}$$

Da die gewünschten Merkmale eine kleine Intraklassen-Streuung und eine große Interklassen-Streuung aufweisen sollen, wird ein Maß für die Separierbarkeit benötigt, das das Verhältnis aus $\mathbf{B}$ und $\mathbf{W}$ berücksichtigt. In [Fukunaga 90] werden hierzu vier äquivalente Optimierungskriterien vorgestellt, unter denen das Folgende auf Grund seiner Invarianz gegenüber nicht-singulären linearen Transformationen von besonderem Interesse ist:

$$\mathcal{F}(A) = \frac{\det(A^\top \cdot \mathbf{B} \cdot A)}{\det(A^\top \cdot \mathbf{W} \cdot A)} \tag{6.1}$$

Gesucht ist dabei eine dimensionsreduzierende lineare Abbildung A, die den Ausdruck in Gleichung 6.1 maximiert. Die Maximierung kann auf die Lösung eines verallgemeinerten Eigenwertproblems zurückgeführt werden:

$$\mathbf{B} \cdot v_d = \lambda_d \cdot \mathbf{W} \cdot v_d \qquad \text{mit } \lambda_d \in I\!R,\ v_d \in I\!R^D,\ d = 1, ..., D$$

Hierbei bezeichnen die Größen λ_d die Eigenwerte und v_d die zugehörigen Eigenvektoren. Die Spaltenvektoren der dimensionsreduzierenden Abbildung $A \in I\!R^{D \times D'}$ setzen sich nun aus den Eigenvektoren v_d mit den D' größten Eigenwerten λ_d zusammen, d.h. ist π eine Permutation der Menge $\{1,...,d,...,D\}$, so daß

$$\lambda_{\pi(1)} \geq, ..., \geq \lambda_{\pi(d)} \geq, ..., \geq \lambda_{\pi(D)}\,,$$

dann gilt

$$A := [v_{\pi(1)}, ..., v_{\pi(D')}].$$

Für eine gegebene Dimension D' wird das Kriterium (6.1) durch diese Wahl der Eigenwerte bzw. Eigenvektoren optimiert. Die Dimensionsreduktion ist hierbei ein wesentlicher Bestandteil der LDA, da die Transformation die Merkmale ansonsten nur zu dekorrelieren vermag. Dieser Effekt wäre jedoch bestenfalls bei der Verwendung von Kovarianzvektoren von Interesse, könnte dort jedoch auch durch eine wesentlich einfachere Hauptachsentransformation erzielt werden.

6.1.1 Effekt der LDA-Transformation

Das Prinzip der LDA basiert auf einer simultanen Diagonalisierung der beiden Matrizen $\mathbf{W}$ und $\mathbf{B}$. Im Fall $D' = D$ wird die Abbildung A so gewählt, daß $A^\top \cdot \mathbf{W} \cdot A = \mathbf{I}$ und $A^\top \cdot \mathbf{B} \cdot A = \mathbf{D}$ gilt, wobei $\mathbf{I}$ die Einheitsmatrix bezeichnet und $\mathbf{D}$ eine Diagonalmatrix. Die Operation kann in drei Schritte zerlegt werden [Valtchev 95], die einer Diagonalisierung der $\mathbf{W}$-Matrix (b) mit anschließender Skalierung der Koordinatenachsen (c) und einer zusätzlichen Diagonalisierung der $\mathbf{B}$-Matrix (d) ensprechen (vgl. Abbildung 6.1).

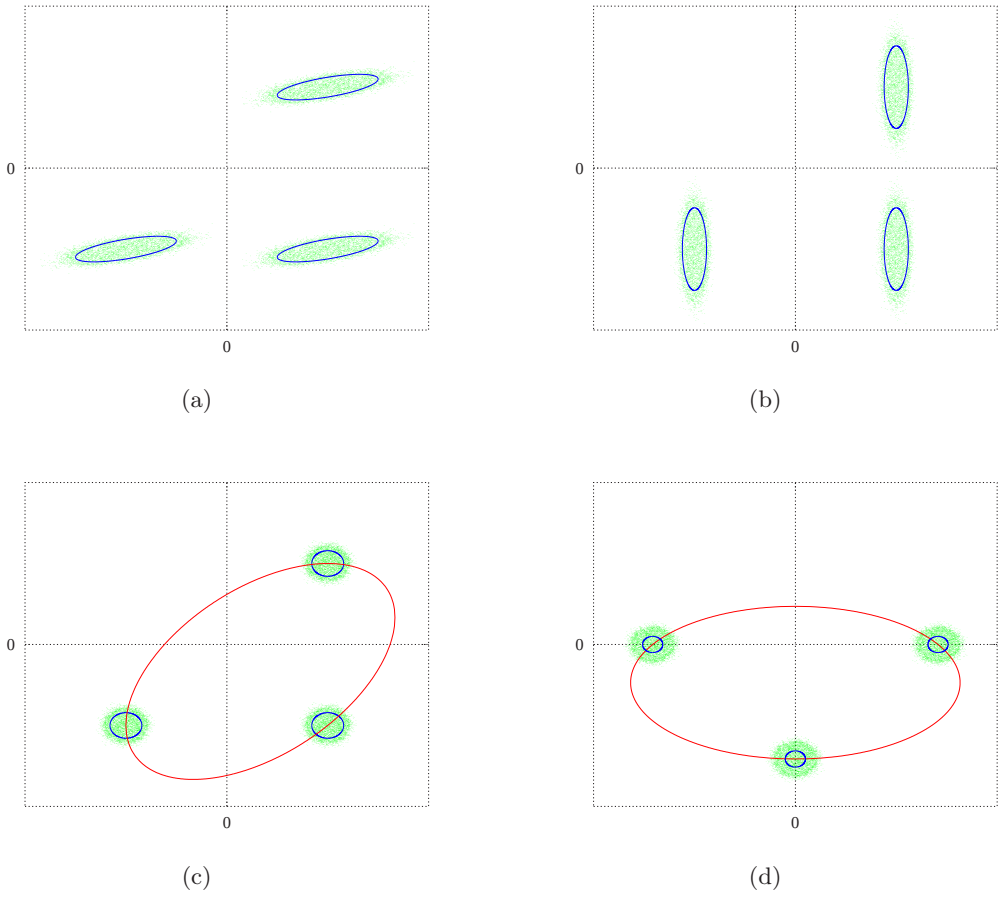


Abbildung 6.1: Effekt der LDA-Transformation

Um hochdimensionale Merkmalsvektoren zu gewinnen, werden im Rahmen dieser Arbeit zu jedem Zeitpunkt je drei aufeinanderfolgende Beobachtungsvektoren x_{t-1}, x_t und x_{t+1} zu *einem* Vektor zusammengefaßt. Damit ist die Dimensionserweiterung eigentlich nur künstlich aus den ursprünglichen Merkmalsvektoren herbeigeführt. Daß hierbei dennoch häufig Verbesserungen zu beobachten sind, hängt im besonderen damit zusammen, daß die Markovketten nach Gleichung (2.7) Kontextunabhängigkeit voraussetzen. Durch die Konkatenation der Merkmalsvektoren wird diese zumeist ungerechtfertigte Modellannahme wieder relativiert.

6.2 Lineare Maximum Mutual Information Analyse (LMA)

Bezeichnet A wiederum eine dimensionsreduzierende, lineare Abbildung für die Beobachtungsvektoren x

$$y = A^\top \cdot x, \qquad A \in I\!R^{D \times D'}$$

dann folgt unter Vernachlässigung der Normierungsbedingung für die Gaußschen Emissionsverteilungen[1]:

$$p_A(y|s) = p(A^T \cdot x|s) \propto e^{-\frac{1}{2}(x-\mu_s)^T \cdot A \cdot \Sigma^{-1} \cdot A^T \cdot (x-\mu_s)}.$$

Verwendet man nun diese Form der Emissionswahrscheinlichkeiten für den in Abschnitt 3.1 beschriebenen allgemeinen Ansatz für diskriminative Kriterien (Gleichung 3.1), dann ergibt sich für die Ableitung nach der Transformationsmatrix A der folgende Ausdruck:

$$\begin{aligned}\frac{\partial F_D(\lambda, A)}{\partial A} = \sum_{r=1}^{R} \sum_{s} \alpha\, f'\Bigg(\log \frac{p_\lambda^\alpha(X_r|W_r) p^\alpha(W_r)}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda^\alpha(X_r|W) p^\alpha(W)} \Bigg) \cdot \\ \cdot \sum_{t=1}^{T_r} \big[\gamma_{r,t}(s|W_r) - \gamma_{r,t}(s)\big] \cdot \frac{\partial \log p(x_{r,t}|s, A)}{\partial A}.\end{aligned} \tag{6.2}$$

Um Nicht-Linearitäten zu vermeiden, wird im folgenden die Modellannahme gemacht, daß die FB-Wahrscheinlichkeiten unabhängig von der Transformationsmatrix A sind. Mit dieser Voraussetzung werden nun die Satzgewichte unabhängig von der linearen Abbildung definiert:

$$f_r := \alpha\, f'\Bigg(\log \frac{p_\lambda^\alpha(X_r|W_r) p^\alpha(W_r)}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda^\alpha(X_r|W) p^\alpha(W)} \Bigg)$$

Danach ergibt sich dann das neue Kriterium zur Optimierung der Abbildung A durch Integration der Gleichung 6.2:

$$\overline{F}_D(\lambda, A) = \sum_{r=1}^{R} f_r \sum_{s} \sum_{t=1}^{T_r} \big[\gamma_{r,t}(s|W_r) - \gamma_{r,t}(s)\big] \log p(x_{r,t}|s, A) \tag{6.3}$$

Die logarithmierten Verteilungen können nun unter Verwendung der Spur (*trace*) in der folgenden Weise umgeformt werden:

$$\begin{aligned}-\log p(x_{r,t}|s, A) &= \frac{1}{2}(x_{r,t} - \mu_s)^\top \cdot A \cdot \Sigma^{-1} \cdot A^\top \cdot (x_{r,t}) + \log\det(2\pi\Sigma) \\ &= \frac{1}{2} tr\big[\Sigma^{-1} \cdot A^\top \cdot (x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top \cdot A\big] + \text{const}(A).\end{aligned}$$

[1] Für den Fall Gaußscher Verteilungen kann man zeigen, daß sich die Verteilungen der transformierten Merkmale für jede nicht-singuläre dimensionsreduzierende lineare Transformation in eindeutiger Weise aus der Verteilung der ursprünglichen Merkmale bestimmen läßt. Im Unterschied zu den linear transformierenden Mittelwerten ergibt sich für die Kovarianzmatrizen allerdings eine recht komplexe Abhängigkeit bezüglich der Kovarianzmatrizen zu den untransformierten Merkmalen [Schlüter 96]. Die Transformationseigenschaften der Mittelwerte sind in den folgenden Gleichungen bereits berücksichtigt.

Für das Kriterium aus Gleichung 6.3 ergibt sich damit

$$\begin{aligned}\overline{F}_D(\lambda, A) = \text{const}(A) - \sum_{r=1}^{R} f_r \sum_s \sum_{t=1}^{T_r} \left[\gamma_{r,t}(s|W_r) - \gamma_{r,t}(s)\right] \cdot \\ \cdot \frac{1}{2} tr\left[\Sigma^{-1} \cdot A^\top \cdot (x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top \cdot A\right] \\ = \text{const}(A) + \frac{1}{2} tr\left[\Sigma^{-1} \cdot A^\top \cdot \overline{\mathbf{B}} \cdot A\right] \\ - \frac{1}{2} tr\left[\Sigma^{-1} \cdot A^\top \cdot \overline{\mathbf{W}} \cdot A\right]\end{aligned}$$

Dabei gilt für die Definition der diskriminativen *Within-* bzw. *Between-Class-Scatter-*Matrizen:

$$\begin{aligned}\overline{\mathbf{W}} &= \sum_{r=1}^{R} f_r \sum_s \sum_{t=1}^{T_r} \gamma_{r,t}(s|W_r) \cdot (x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top \\ \overline{\mathbf{B}} &= \sum_{r=1}^{R} f_r \sum_s \sum_{t=1}^{T_r} \gamma_{r,t}(s) \cdot (x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top\end{aligned}$$

Im Fall der Viterbi-Näherung und des MMI-Kriteriums ($\alpha = 1$, $f_r = 1$) ist die diskriminative Within-Class-Scatter-Matrix wegen $\gamma_{r,t}(s|W_r) = \delta_{s,s_{W_r(t)}}$ identisch zur $\mathbf{W}$-Matrix aus dem konventionellen LDA-Ansatz. Für die diskriminative Between-Class-Scatter-Matrix ergibt sich ferner

$$\overline{\mathbf{B}} \stackrel{Viterbi}{=} \sum_{r=1}^{R} \sum_s \sum_{t=1}^{T_r} \gamma_{r,t}(s) \cdot (x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top$$

mit

$$\gamma_{r,t}(s) \stackrel{Viterbi}{=} \sum_{W \in \mathcal{M}_r} \frac{p_\lambda(X_r|W)p_\lambda(W)}{\sum_{V \in \mathcal{M}_r} p_\lambda(X_r|V)p(V)} \delta_{s,s_{W(t)}}.$$

Das Kriterium zur Maximierung des Verhältnisses der beiden diskriminativen Streuungsmatrizen bezüglich einer dimensionsreduzierenden Abbildung A wird nun analog zum konventionellen LDA-Ansatz gewählt

$$\overline{\mathcal{F}}(A) = \frac{\det(A^\top \cdot \overline{\mathbf{B}} \cdot A)}{\det(A^\top \cdot \overline{\mathbf{W}} \cdot A)},$$

wobei die Optimierung wieder auf die Lösung eines verallgemeinerten Eigenwertproblems zurückgeführt werden kann:

$$\overline{\mathbf{B}} \cdot v_d = \lambda_d \cdot \overline{\mathbf{W}} \cdot v_d \qquad \text{mit } \lambda_d \in I\!R,\ v_d \in I\!R^D,\ d = 1, ..., D$$

Die Spaltenvektoren der dimensionsreduzierenden Abbildung A ergeben sich dann in analoger Weise als die Eigenvektoren v_d zu den D' größten Eigenwerten λ_d.

6.2.1 Interpretation der LMA in der *Viterbi*-Näherung

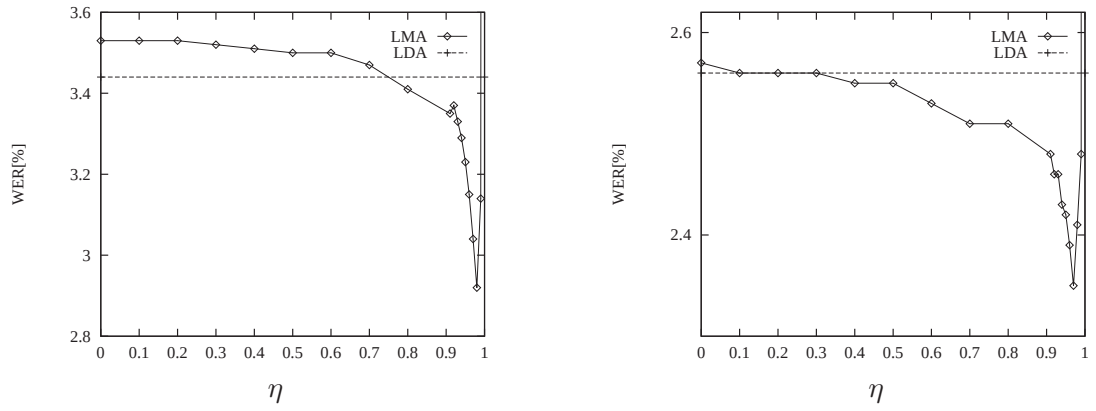
Wird die Größe $\gamma_{r,t}(s)$ als Wahrscheinlichkeit für die Zuordnung der akustischen Beobachtung $x_{r,t}$ zum Zustand s interpretiert, dann ist die Between-Class-Scatter-Matrix $\overline{\mathbf{B}}$ im wesentlichen nur durch die zentrierten Momente $(x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top$ derjenigen Beobachtungen bestimmt, deren Gewichte einen wesentlichen Beitrag leisten, d.h. die sehr sicher der Klasse s zugeordnet werden (hierzu zählen sowohl Beobachtungen, deren Zuordnung zur Klasse s korrekt ist als auch Beobachtungen, die häufig der falschen Klasse zugeordnet werden). Da die Optimierung der Kriterien $\mathcal{F}(A)$ und $\overline{\mathcal{F}}(A)$ unter der Nebenbedingung $v_d^\top \cdot \mathbf{W} \cdot v_d = const$ bzw. $v_d^\top \cdot \overline{\mathbf{W}} \cdot v_d = const$ erfolgt (siehe direkte Herleitung [Ney und Eiden 95], S. 132), bleiben die korrekt zugeordneten Beobachtungen in der Nähe ihrer Klassenzentren, während sich die falsch klassifizierten Beobachtungen aufgrund der Maximierung der Interklassenstreuung von den Zentren entfernen. Im Vergleich zum konventionellen LDA-Ansatz arbeitet das neue, von diskriminativen Verfahren abgeleitete Kriterium somit in stärkerem Maße lokal.

6.2.2 Glättung der linearen MMI-Analyse

Erste Experimente zeigten, daß sich das LMA-Kriterium in der oben dargestellten Form zu stark auf falsch klassifizierte Beobachtungen konzentriert und somit zu einem Verlust in der Performanz führt. Die Ursache hierfür ist auf die obere Näherungsannahme zurückzuführen, daß die Satzgewichte f_r unabhängig von der linearen Abbildung A sind. Um diesen Effekt zu kompensieren, wird die diskriminative Between-Class-Scatter-Matrix $\overline{\mathbf{B}}$ daher mit den Momenten $(\mu-\mu_s)(\mu-\mu_s)^\top$ aus der konventionellen Between-Class-Scatter-Matrix $\mathbf{B}$ unter Berücksichtigung des Gewichtes $\gamma_{r,t}(s)$ geglättet:

$$\overline{\mathbf{B}} = \sum_{r=1}^{R} f_r \sum_{s} \sum_{t=1}^{T_r} \gamma_{r,t}(s) \cdot \Big\{ \eta \cdot (x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top + [1-\eta] \cdot (\mu - \mu_s)(\mu - \mu_s)^\top \Big\}$$

Der Parameter η erlaubt hierbei einen kontinuierlichen Übergang zwischen den beiden Momenten $(x_{r,t} - \mu_s)(x_{r,t} - \mu_s)^\top$ und $(\mu - \mu_s)(\mu - \mu_s)^\top$. Die Wahl der Größe η ist korpusspezifisch und wurde empirisch bestimmt. Die folgende Abbildung zeigt die auf den Trainingsdaten des SieTill-Korpus ermittelten Wortfehlerraten für unterschiedliche Einstellungen des Parameters η (Einzelverteilungen, ein gemeinsamer (gepoolter) Kovarianzvektor für alle Zustände). Zum Vergleich enthalten beide Diagramme zusätzlich die bei Verwendung der konventionellen LDA erzielte Fehlerrate.



(a) *male speakers* (b) *female speakers*

Kapitel 7

Diskriminatives Splitting

Oftmals kann eine bessere Adaption der Beobachtungen im Training erzielt werden, wenn die Anzahl der Mischungskomponenten einer Emissionsverteilung erhöht wird (*Splitting*). Hierzu wird der Mittelwert μ einer Mischungskomponente in zwei neue Mittelwerte μ_+ und μ_- aufgespaltet, wobei die neuen Mittelwerte gegenüber dem alten Zentrum um einen kleinen Betrag („Epsilon“) gestört werden:

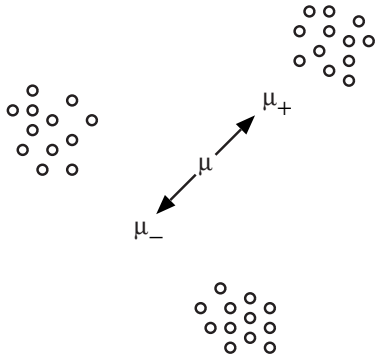


Abbildung 7.1: Aufspalten eines Mittelwertes

Bei Verwendung der Maximum-Approximation für Mischverteilungen erhalten die neuen Mischungskomponenten $p_+(x|\{\mu^+_{s,l}, \Sigma_{s,l}\})$ bzw. $p_-(x|\{\mu^-_{s,l}, \Sigma_{s,l}\})$ jeweils das alte Mischverteilungsgewicht $c_{s,l}$. Die Nebenbedingung, daß sich sämtliche Gewichte einer Mischverteilung zu eins summieren sollen, wird somit für eine Trainingsiteration aufgegeben und erst bei der nächsten Reestimation wieder hergestellt. Wird dagegen das Baum-Welch-Training verwendet, so muß die Nebenbedingung bereits für die unmittelbar folgende Iteration erfüllt sein, da hier nicht die Maximum-Approximation für Mischverteilungen verwendet wird. Die neuen Mischungskomponenten erhalten dann jeweils das halbe Mischverteilungsgewicht der ursprünglichen Mischung, d.h. $c_{s,l}/2$. Die Kovarianzmatrizen werden hier der Einfachheit halber über alle Mischungskomponenten einer Mischverteilung gepoolt.

Das Kriterium zum Splitten einer Mischverteilung im ML-Training richtet sich üblicherweise nach der relativen Anzahl der Beobachtungen, die auf eine Mischungskomponente entfallen. Wird hierbei ein zuvor festgelegter Schwellwert überschritten, erfolgt die Auf-

spaltung der betreffenden Mischungkskomponente in zwei neue Mischungen. Der Nachteil dieser Methode ist, daß hierdurch auch Mischverteilungen gesplittet werden können, die bereits eine gute (u. U. sogar optimale) Abdeckung der Beobachtungen im Training ermöglichen. Ein weiteres Splitten führt hier lediglich zu einem redundanten Anstieg der Parameterzahl. Zudem tendiert dieses Verfahren dazu, daß viele Mischungskomponenten verstärkt den Bereich in der Nähe des Klassenzentrums abdecken, obwohl es das Ziel sein sollte, die „Regionen“ an den Klassengrenzen zu erfassen. Eine mögliche Lösung für dieses Problem basiert auf dem MMI-Kriterium.

7.1 Interpretation der diskriminativen Statistiken

Betrachtet man die Zerlegung der diskriminativen Erwartungswerte $\Gamma_{s,l}(1)$ in den Anteil $\Gamma_{s,l}^{spk}(1)$ für die gesprochenen Wortfolgen und den Anteil $\Gamma_{s,l}^{gen}(1)$ für alle konkurrierenden Wortfolgen, so kann die Größe $\Gamma_{s,l}^{spk}(1)$ als ein Maß dafür interpretiert werden, wie häufig eine Trainingsbeobachtung $x_{r,t}$ auf das korrekte Modell, d.h. die richtige Klasse entfallen sollte. Die Größe $\Gamma_{s,l}^{gen}(1)$ kann dagegen als Maß dafür interpretiert werden, wie sehr die Beobachtungen dazu tendieren, einer konkurrierenden Klasse zugeordnet zu werden.

Ist nun die Differenz zwischen dem Anteil für die gesprochenen Wortfolgen und den konkurrierenden Wortfolgen sehr viel größer als null, d.h.

$$\Gamma_{s,l}^{spk}(1) - \Gamma_{s,l}^{gen}(1) \gg 0,$$

dann liegt offensichtlich ein Diskriminationsproblem vor, denn viele der zur Klasse s gehörenden Beobachtungen wurden während der Erkennung anderen Klassen $s' \neq s$ zugeordnet. Dieses Diskriminationsproblem wird nun dadurch gelöst, indem die Anzahl der Mischungskomponenten der mit dem Zustandsindex s assoziierten Mischverteilung erhöht wird.

Der Algorithmus zur Aufspaltung der Mischungen im diskriminativen Training ergibt sich dann wie folgt:

1. Führe einen MMI-Trainingsschritt durch und bestimme die diskriminativen Erwartungswerte $\Gamma_{s,l}(1)$.

2. Sortiere die diskriminativen Erwartungswerte $\Gamma_{s,l}(1)$ in aufsteigender Reihenfolge, d.h. bestimme eine Permutation π der Zustandsindizes, so daß gilt:

$$\Gamma_{\pi(s_1,l(s_1))} \leq ... \leq \Gamma_{\pi(s_n,l(s_n))} \leq ... \leq \Gamma_{\pi(s_N,l(s_N))}$$

3. Splitte alle Mischungskomponenten (s,l), für den der diskriminative Erwartungswert $\Gamma_{s,l}(1)$ größer als der Median über alle $\Gamma_{\widetilde{s},\widetilde{l}}(1)$ ist.

4. Wähle die gesplittete Mischungskomponente $\mu_{s,l}^{+}$ gemäß Gleichung (3.11) zu $\overline{\mu}_{s,l}$ und $\mu_{s,l}^{-}$ als den alten Mittelwert $\mu_{s,l}$. Verfahre entsprechend für die Mischverteilungsgewichte.

5. Verwende für die folgenden Trainingsiterationen wieder das konventionelle ML-Training. Bei einem erneuten Split beginne wieder mit Schritt 1.

Alternativ kann im Schritt 3 statt des Medians auch ein anderes Element aus der Liste der sortierten diskriminativen Erwartungswerte gewählt werden. Die Position des gewählten Elementes entscheidet darüber, wie groß der Anteil der zu splittenden Mischungskomponenten ist (beim Median sind dies im Mittel etwa 50% aller Mischungen). Die Kovarianzmatrizen werden wiederum als gepoolt über alle Mischungskomponenten einer Klasse s angenommen und für den MMI-Schritt gemäß Gleichung (3.14) bzw. (3.15) bestimmt.

Das Grundprinzip des Splitting im diskriminativen Training geht auf [Normandin 95] zurück. Der zuvor beschriebene Algorithmus enthält jedoch gegenüber dem von Normandin vorgeschlagenen Verfahren zwei wesentliche Änderungen:

- In [Normandin 95] wird ein konstanter Faktor $z \in (0, ..., 1]$ (dort: $z = 0.2$) auf den größten diskriminativen Erwartungswert $\max_{s,l}\{\Gamma_{s,l}(1)\}$ gegeben. Anschließend werden nur diejenigen Mischungskomponenten (s,l) aufgespalten, für die $\Gamma_{s,l}(1) \geq z \cdot \max_{s,l}\{\Gamma_{s,l}(1)\}$ gilt. Diese Vorgehensweise tendiert jedoch dazu, entweder zu viele oder aber zu wenige Mischungskomponenten zu splitten.

- Nach dem Splitting erfolgt das weitere Training der Mischverteilungsparameter in [Normandin 95] unter Verwendung des MMI-Kriteriums. Da daß MMI-Kriterium jedoch sehr empfindlich auf Ausreißer in den Trainingsdaten reagiert, besteht hier die Gefahr, daß die neuen Mittelwerte $\mu^{+}_{s,l}$ bzw. $\mu^{-}_{s,l}$ zu stark auf eher untypische Beobachtungen konzentrieren und die Parameterschätzung damit sehr schnell den Trainingskorpus überadaptiert. Ein Training mit Hilfe des ML-Kriteriums kann dagegen eine höhere Generalisierbarkeit ermöglichen.

In der folgenden Abbildung sind der Verlauf der Fehlerkurven für das konventionelle und das neue (diskriminative) Split-Kriterium für das SieTill-Testkorpus[1] dargestellt.

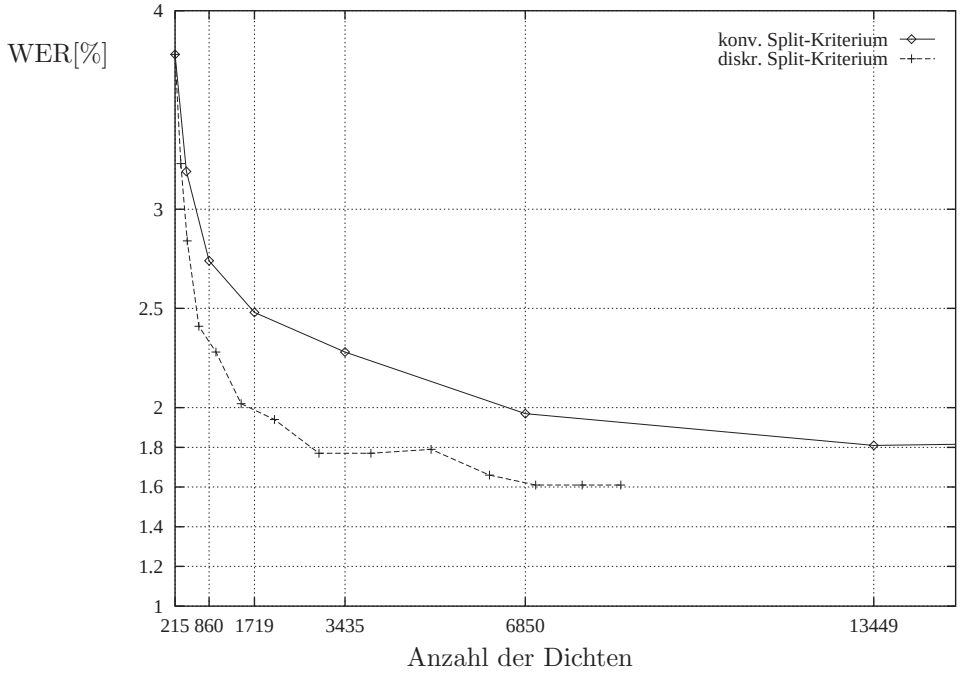


Abbildung 7.2: Konventionelles und diskriminatives Split-Kriterium

[1]Eine Beschreibung des SieTill-Korpus findet sich in Abschnitt 8.1.

Abbildung 7.3 zeigt für jede Iteration ein Histogramm über die Anzahl der Mischverteilungen für die einzelnen Zustände.

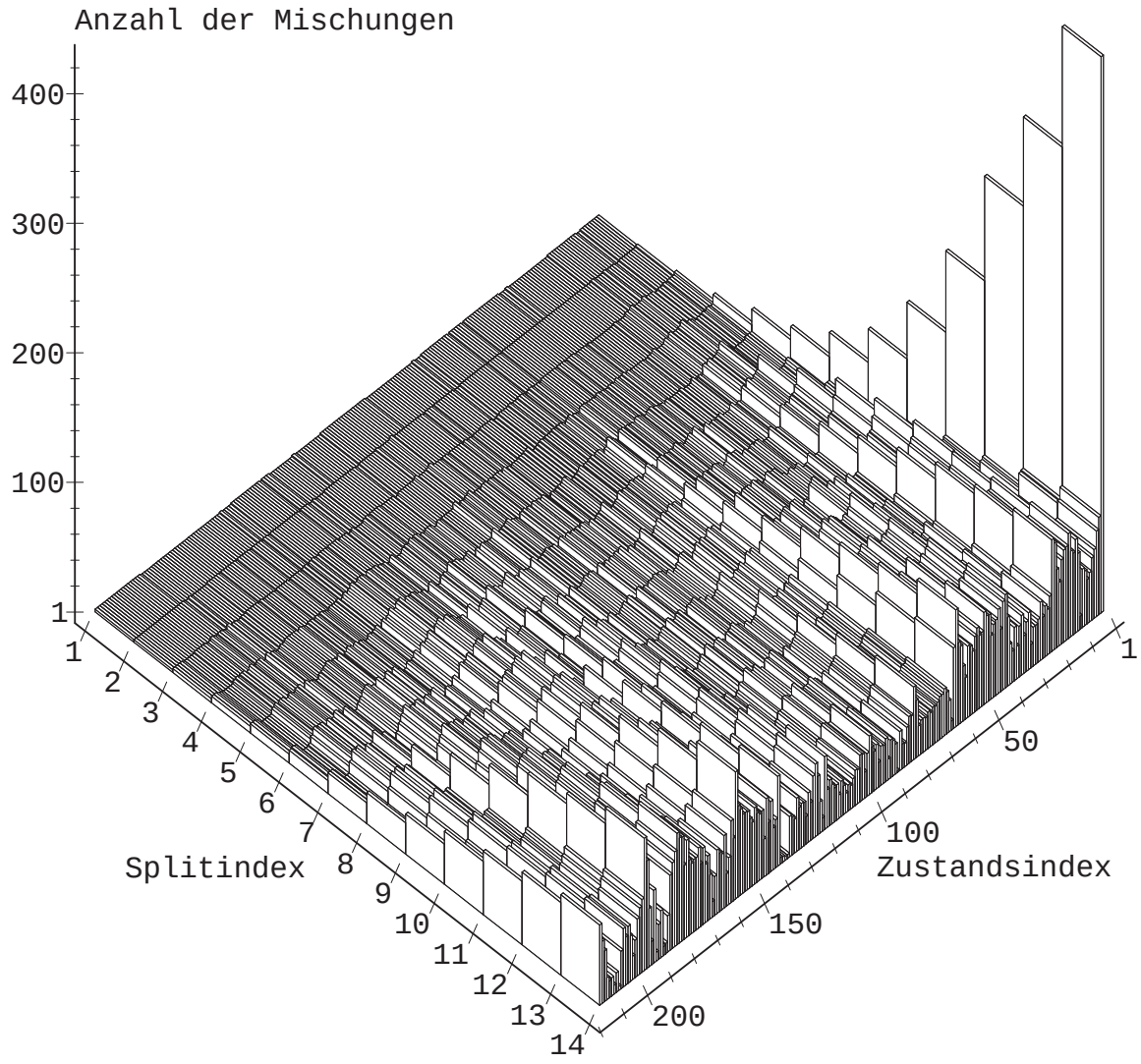


Abbildung 7.3: Histogramm über die Anzahl der jeder Mischverteilung zugeordenten Mischungskomponeten als Funktion des Splitindex und des Zustandsindex (SieTill-Korpus).

Kapitel 8

Experimente und Ergebnisse

In diesem Kapitel werden die Experimente und Messungen zu den in dieser Arbeit behandelten Verfahren vorgestellt. Den Schwerpunkt der Untersuchungen bildet hierbei ein Vergleich der diskrminativen Kriterien *Corrective Training* (CT), *Maximum Mutual Information* (MMI), *falsifizierendes Training* (FT) und *Minimum Classification Error* (MCE). Der Vergleich wird sowohl für die diskriminativen Kriterien untereinander durchgeführt als auch gegenüber dem konventionellen Maximum-Likelihood (ML) Training. Zusätzlich werden Ergebnisse für ein Baum-Welch (BW)-Training unter Verwendung des ML- und des MMI-Kriteriums vorgestellt. Im Rahmen der Untersuchungen über lineare Merkmalstransformationen wird die *lineare Diskriminanzanalyse* (LDA) mit der *linearen MMI Analyse* (LMA) verglichen. Ferner werden die Ergebnisse zum Splitting im diskriminativen Training vorgestellt. Eine Diskussion der Ergebnisse sowie ein Vergleich der erzielten Resultate mit den Ergebnissen anderer Forschungsgruppen schließen das Kapitel ab.

8.1 Beschreibung der Korpora und des Erkennungssystems

Die in dieser Arbeit durchgeführten Messungen wurden stets auf dem englischsprachigen TI-Digits-Korpus bzw. dem deutschsprachigen SieTill-Korpus durchgeführt. Bei beiden Korpora handelt es sich um Datensammlungen für die sprecherunabhängige Erkennung kontinuierlich gesprochener Ziffern. Das Vokabular der TI-Digits-Datensammlung umfaßt 11 englische Ziffern, inklusive „oh“. Training und Erkennung beschränken sich jeweils auf die Sprachproben der erwachsenen Sprecher und wurden direkt über ein Mikrofon aufgenommen.

Das SieTill-Korpus wurde für die Erkennung kontinuierlich gesprochener deutscher Ziffernketten zusammengestellt. Im Unterschied zum TI-Digits-Korpus wurden sowohl die Trainings- als auch die Testäußerungen über verschiedene Telefonkanäle aufgenommen. Das Vokabular umfaßt wie das TI-Digits-Korpus 11 Ziffern, darunter „zwo“.

Tabelle 8.1 enthält für beide Korpora eine Übersicht über die Aufteilung der Datensammlungen in Trainings- und Testsätze. Die Erkennnungssysteme für beide Korpora basieren auf Ganzwortmodellen und verwenden kontinuierliche Emissionsverteilungsdichten. Die folgende Aufstellung stellt die wichtigsten Eigenschaften beider Erkennungssysteme zusammen:

Tabelle 8.1: Korpus-Statistiken für den SieTill- und den TI-Digits Korpus

Korpus-Statistiken								
Korpus	Aufnahmekanal/ Sprache	Train./ Test	male		female		gesamt	
			Sätze	Ziffern	Sätze	Ziffern	Sätze	Ziffern
SieTill	Telefon/ deutsch	Train.	6886	22631	6150	20226	13036	42857
		Test	6938	22881	6176	20205	13114	43086
TI-Digits	Mikrofon/ englisch	Train.	4235	13915	4388	14414	8623	28329
		Test	4311	14159	4389	14424	8700	28583

SieTill Erkennungssystem

- 11 deutsche Ziffern, inklusive „zwo“
- geschlechtsabhängige Ganzwortmodelle, bestehend aus insgesamt 214 Zuständen; zusätzlich je ein Zustand für geschlechtsabhängiges Pausen-Modell (*Silence*)
- Gaußsche Mischverteilungen mit einer über alle Zustände gepoolten diagonalen Kovarianzmatrix bzw. mit zustandsabhängigen diagonalen Kovarianzmatrizen
- 12 cepstrale Merkmale mit ersten Ableitungen und der Ableitung der Energie (d.h. 25-komponentiger Merkmalsvektor)
- Frameshift: 10 ms

TI-Digits Erkennungssystem

- 11 englische Ziffern, inklusive „oh“
- geschlechtsabhängige Ganzwortmodelle, bestehend aus insgesamt 357 Zuständen; zusätzlich je ein Zustand für geschlechtsabhängiges Pausen-Modell (*Silence*)
- Gaußsche Unimodal-Verteilungen mit zustandsabhängiger diagonaler Kovarianzmatrix
- 16 cepstrale Merkmale mit je 16 ersten und zweiten Ableitungen (d.h. 48-komponentiger Merkmalsvektor)
- Frameshift: 10 ms

Die im Rahmen der folgenden Untersuchungen produzierten Parametermengen wurden stets geschlechtsabhängig trainiert. Für beide Parametersätze wurde anschließend eine Erkennung auf dem gesamten Korpus durchgeführt, so daß für jede Äußerung zwei (unter Umständen verschiedene) Dekodierungen vorlagen. Die Entscheidung für eine der beiden Satzhypothesen wurde stets zugunsten derjenigen Wortfolge getroffen, die die größere Satzwahrscheinlichkeit produzierte.

8.2 Konventioneller und diskriminativer Ansatz in der Viterbi-Näherung

In diesem Abschnitt werden die Ergebnisse zu den Untersuchungen bezüglich des konventionellen ML-Ansatzes sowie der diskriminativen Verfahren CT, MMI, FT und MCE vorgestellt. Training und Erkennung erfolgten dabei jeweils in der Viterbi-Näherung. Bei Verwendung von Mischverteilungen wurde stets die Maximum-Approximation verwendet.

8.2.1 Training der Initialreferenzen

Für die diskriminativen Verfahren wurde – sofern nichts anderes angegeben wird – stets eine ML-trainierte Referenz als Aufsatzpunkt verwendet. Im Rahmen der auf dem TI-Digits Korpus durchgeführten Untersuchungen wurde hierzu ein ML-Training für Gaußsche Mischverteilungen und zustandsabhängige Varianzen durchgeführt. Lineare Merkmalstransformationen wurden nicht verwendet. Die erzielten Erkennungsergebnisse sind der Tabelle 8.2 zu entnehmen.

Tabelle 8.2: Initialergebnis auf dem TI-Digits Korpus für ein ML-Training bei Verwendung Gaußscher Einzelverteilungen mit zustandsabhängigen diagonalen Kovarianzmatrizen.

TI-Digits: ML-Kriterium					
Korpus	Kriterium	Methode	del-ins-sub	WER[%]	SER[%]
Train.	ML	–	79 - 11 - 68	0.56	1.69
Test	ML	–	56 - 31 - 120	0.72	2.00

Das Kürzel WER[%] steht hierbei für die prozentuale Wortfehlerrate (*word error rate*). Die Wortfehlerrate wurde mittels der Levenshtein-Distanz bestimmt und setzt sich aus der minimalen Anzahl benötigter Wortauslassungen (del), Worteinfügungen (ins) sowie Wortersetzungen (sub) zusammen, die erforderlich sind, um die gesprochene Wortfolge in die erkannte Sequenz umzuwandeln. Die Spalte SER[%] gibt die prozentuale Satzfehlerrate (*string error rate*) an.

Die ML-Ergebnisse für die auf dem Sietill-Korpus durchgeführten Untersuchungen sind Tabelle 8.3 zu entnehmen. Dargestellt sind jeweils die erzielten Fehlerraten auf dem Testkorpus für Gaußsche Einzel- und Mischverteilungen unter Verwendung zustandsabhängiger diagonaler Kovarianzmatrizen (SV) ohne eine zusätzliche Transformation, sowie die Resultate für gepoolte diagonale Kovarianzmatrizen (PV) unter optionaler Verwendung einer LDA. Die Spalte „Dichten“ gibt jeweils die Anzahl verwendeter Mischungskomponenten an, die auf die beiden geschlechtsabhängig trainierten Parametersätze entfallen. Hierbei steht „m“ für die männlichen und „f“ für die weiblichen Sprecher. Die besten ML-Resultate wurden für gepoolte Varianzen unter Verwendung einer LDA erzielt. Für alle folgenden Untersuchungen auf dem SieTill-Korpus wurden daher stets, soweit nicht anders angegeben, die ML-Initialreferenzen für gepoolte diagonale Kovarianzmatrizen unter Verwendung einer LDA als Aufsatzpunkt für ein weiteres diskriminatives Training herangezogen.

8.2.2 Konvergenzverhalten und Vergleich der Optimierungsmethoden

Da es bei Verwendung kontinuierlicher Verteilungsdichten bislang keinen Konvergenzbeweis für die Reestimation der Mischverteilungsparameter im erweiterten Baum (EB)-Algorithmus gibt, wurde zunächst das Konvergenzverhalten der diskriminativen Kriterien untersucht. In einem ersten Experiment wurde hierzu das Corrective Training (CT)-Kriterium mit dem EB-Algorithmus sowie einem Gradientenverfahren (GD) optimiert. Für die spezielle Wahl der Schrittweiten im Gradientenverfahren gemäß Abschnitt 3.5.2 ergaben Experimente auf dem TI-Digits-Korpus erwartungsgemäß ein nahezu identisches

Tabelle 8.3: ML-Initialergebnisse auf dem SieTill-Test-Korpus für Gaußsche Einzel- und Mischverteilungen mit zustandsabhängigen diagonalen Kovarianzmatrizen (SV) ohne zusätzliche lineare Transformation, sowie gepoolten diagonalen Kovarianzmatrizen (PV) unter optionaler Verwendung einer LDA.

SieTill: ML-Kriterium						
Korpus	LDA	Var.	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**nein**	SV	215+215	422 - 391 - 990	4.18	10.48
			430+430	391 - 379 - 887	3.85	9.57
			860+860	386 - 348 - 797	3.56	9.02
			1720+1720	365 - 268 - 713	3.13	7.88
			3435+3439	356 - 246 - 634	2.87	7.28
			6759+6806	**341 - 228 - 587**	**2.68**	**6.79**
			13K + 13K	411 - 220 - 1108	4.04	10.20
			23K + 23K	414 - 211 - 1116	4.04	10.18
		PV	215+215	503 - 357 - 1117	4.59	11.34
			430+430	374 - 380 - 872	3.77	9.46
			860+860	295 - 405 - 808	3.50	8.91
			1720+1720	287 - 324 - 686	3.01	7.69
			3440+3440	255 - 267 - 629	2.67	6.92
			6787+6824	249 - 242 - 573	2.47	6.35
			13K + 13K	236 - 239 - 529	2.33	5.91
			23.7K+23.7K	**226 - 231 - 507**	**2.24**	**5.87**
	ja	PV	215+215	304 - 270 - 1053	3.78	9.74
			430+430	234 - 318 - 824	3.19	8.31
			860+860	198 - 329 - 652	2.74	7.18
			1719+1720	190 - 298 - 579	2.48	6.47
			3435+3436	191 - 294 - 499	2.28	5.92
			6862+6839	198 - 201 - 450	1.97	5.31
			13K+13K	**198 - 162 - 419**	**1.81**	**4.93**
			25K+24K	193 - 169 - 431	1.85	4.94

Iterationsverhalten der Zielfunktion $F_{CT}(\lambda)$ für beide Optimierungsmethoden (siehe Abbildung 8.1 und Tabelle 8.4)

Das gleiche Verhalten konnte ebenso für den Verlauf der Wortfehlerrate auf den Trainings- und Testdaten des TI-Digits Korpus beobachtet werden (siehe Abbildung 8.2). In ähnlicher Weise wurden die Experimente nun auf dem SieTill-Korpus für Gaußsche Mischverteilungen unter Verwendung einer gepoolten diagonalen Kovarianzmatrix sowie einer LDA wiederholt. Hierbei konnte die Aussage, daß der Verlauf der CT-Zielfunktion sowie der Wortfehlerrate auf den Trainingsdaten für die Optimierungsmethoden EB und GD nahezu identisch sind, bestätigt werden (vgl. hierzu die Abbildungen 8.3 (a) und 8.3 (b)). Abweichungen ergaben sich jedoch für die höheren Iterationen, was aber auf eine zu große Schrittweite zurückzuführen ist (vgl. Abschnitt 8.2.4). Die auf den Trainings- und Testdaten des SieTill-Korpus ermittelten Fehlerraten sind für beide Optimierungsmethoden in Tabelle 8.5 aufgeführt.

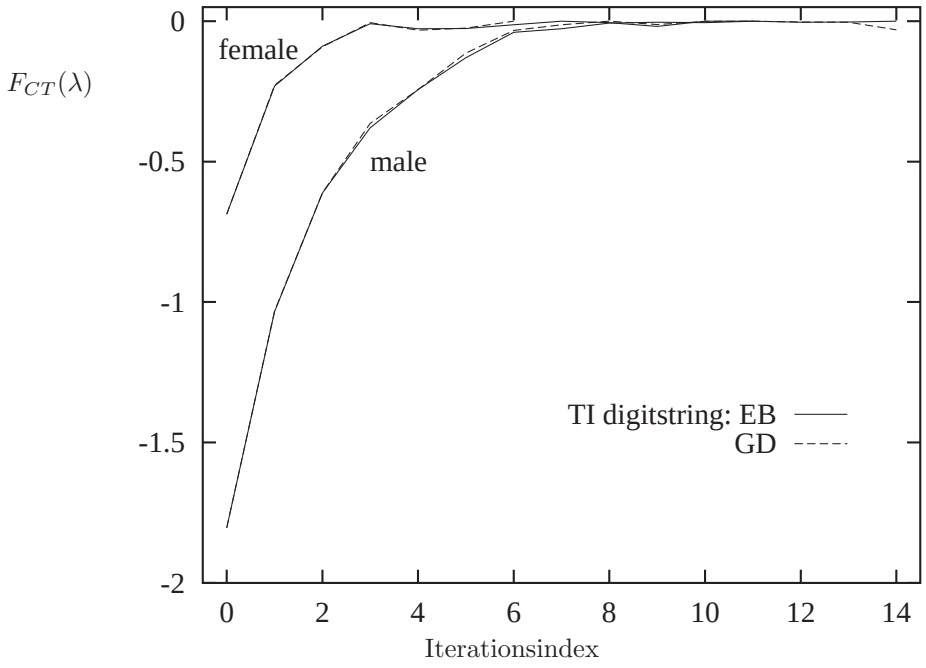


Abbildung 8.1: CT-Kriterium als Funktion des Iterationsindex auf den Trainingsdaten des TI-Digits-Korpus für Gaußsche Einzelverteilungen mit zustandsabhängiger diagonaler Kovarianzmatrix.

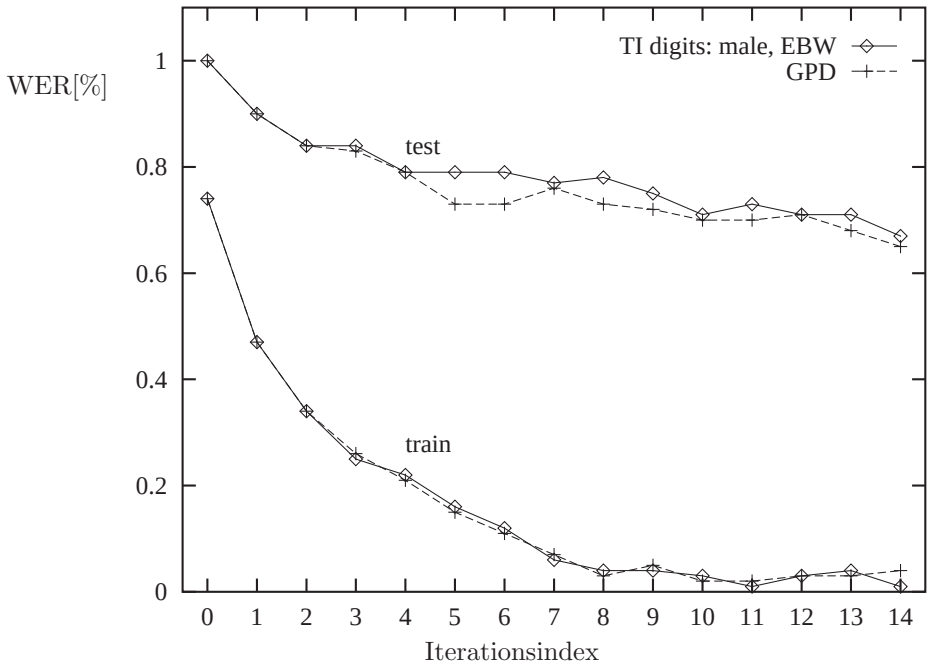


Abbildung 8.2: Wortfehlerrate (WER) als Funktion des Iterationsindex für das CT-Kriterium bei Verwendung Gaußscher Emissionsverteilungen mit zustandsabhängiger diagonaler Kovarianzmatrix (männliche Sprecher des TI-Digits Korpus).

Tabelle 8.4: Fehlerraten auf den Trainings- und Testdaten des TI-Digits Korpus für das *Corrective-Training* (CT) Kriterium unter Verwendung Gaußscher Einzelverteilungen mit zustandsabhängigen diagonalen Kovarianzmatrizen. Die Optimierung des CT-Kriteriums wurde jeweils mit dem *erweiterten Baum* (EB) Algorithmus sowie mit dem in Abschnitt 3.5.2 beschriebenen Gradientenverfahren (GD) durchgeführt.

TI-Digits: EB vs. GD					
Korpus	Kriterium	Methode	del-ins-sub	WER[%]	SER[%]
Train.	ML	–	79 - 11 - 68	0.56	1.69
	CT	EB	0 - 0 - 0	0.00	0.00
		GD	2 - 2 - 2	0.02	0.06
Test	ML	–	56 - 31 - 120	0.72	2.00
	CT	EB	35 - 24 - 83	0.50	1.38
		GD	36 - 24 - 75	0.47	1.32

Tabelle 8.5: Vergleich der Optimierungsmethoden *erweiterter Baum* (EB) Algorithmus und *Gradientenverfahren* (GD) auf dem SieTill-Korpus für Gaußsche Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und LDA.

SieTill: EB vs. GD							
Korpus	LDA	Kriterium	Methode	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Train.	**ja**	CT	EB	6857 + 6821	18 - 8 - 3	0.07	0.21
			GD	6858 + 6825	18 - 15 - 6	0.09	0.27
Test	**ja**	CT	EB	6857 + 6821	152-208-417	1.80	4.96
			GD	6858 + 6825	128-223-433	1.82	4.97

8.2.3 Auswahl des Parametersatzes

Da das Konvergenzverhalten auf Trainings- und Testdaten recht gut übereinstimmt, wurde die Konvergenz der Wortfehlerrate auf den Trainingsdaten als Kriterium für den Abbruch der Trainingsiterationen verwendet. Der für die Erkennung auf den Testdaten verwendete Parametersatz wurde für die folgenden Experimente daher stets zur Iteration mit der kleinsten Wortfehlerrate auf den Trainingsdaten gewählt. Die Tabelle 8.4 faßt die besten erzielten Resultate auf dem TI-Digits-Korpus zusammen, die mit dem CT-Kriterium unter Verwendung Gaußscher Einzelverteilungen sowie einer über alle Zustände gepoolten diagonalen Kovarianzmatrix erreicht worden sind. Insbesondere konnte der Trainingskorpus für Einzelverteilungen mit Hilfe des EB-Algorithmus vollständig, d.h. fehlerfrei abgedeckt werden.

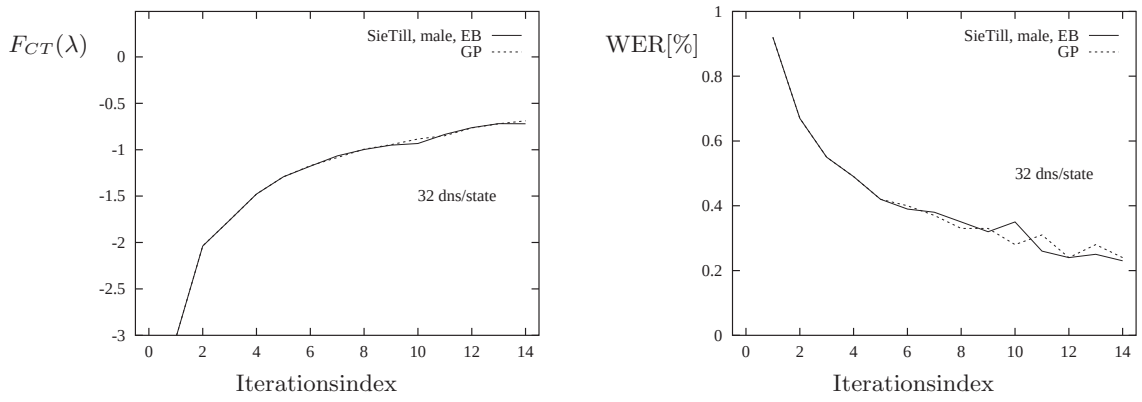


Abbildung 8.3: Verlauf der Zielfunktion (a) und der Wortfehlerrate (b) als Funktion des Iterationsindex für das CT-Kriterium auf den Trainingsdaten des SieTill-Korpus (männliche Sprecher) bei Verwendung Gaußscher Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA.

Aufgrund der großen Homogenität des TI-Digits-Korpus sowie der niedrigen, bereits für Einzelverteilungen erzielten Fehlerrate, wurden alle weiteren Experimente nur noch auf dem SieTill-Korpus durchgeführt.

8.2.4 Einfluß unterschiedlicher Schrittweiten

Trotz einer insgesamt über die Iterationen erkennbaren Konvergenz traten für das SieTill-Korpus vereinzelt Sprünge im Verlauf der Zielfunktion sowie der Wortfehlerrate für das CT-Kriterium auf (siehe Abbildung 8.4 und 8.5). Wie in Abschnitt 3.6.2 beschrieben wurde, liegt die Ursache hierfür in einer zu kleinen Wahl der die Schrittweite bestimmenden Iterationskonstanten h. Wird der Parameter h zu klein gewählt, nimmt der Einfluß der alten Mischverteilunggsparameter in der Reestimation ab und die neuen Mischverteilungsparameter konzentrieren dann zu stark auf die zuvor falsch klassifizierten Beobachtungen[1]. Eine Erhöhung der Iterationskonstanten von $h = 1.1$ auf den Wert $h = 2$ vermochte nun den Verlauf von Zielfunktion und Wortfehlerrate für die höheren Iterationen zu glätten. Das durch die größere Schrittweite verursachte Sprungverhalten führte jedoch in der Regel bei gleicher Zahl an Iterationen zu kleineren Fehlerraten auf den Trainingsdaten. Kleinere Sprünge wurden deshalb während des Trainings toleriert. So zeigt Abbildung 8.5, daß sich trotz einer deutlich ausgeprägten Oszillation ab der 28. Iteration für die Wahl der Schrittweite $h = 1.1$ dennoch geringfügig bessere Fehlerraten auf den Trainingsdaten erzielen lassen. Ähnliches gilt für den Verlauf des Trainingskriteriums (siehe Abbildung 8.4).

[1]Da der Parameter h die Beimischung der alten Mischverteilungsparameter während der Reestimation steuert, entspricht eine große Schrittweite einem kleinen h. Umgekehrt wird eine kleine Schrittweite durch eine größere Wahl des Parameters h erreicht, da dann der Einfluß der alten Mischverteilungsparameter in der Reestimation zunimmt.

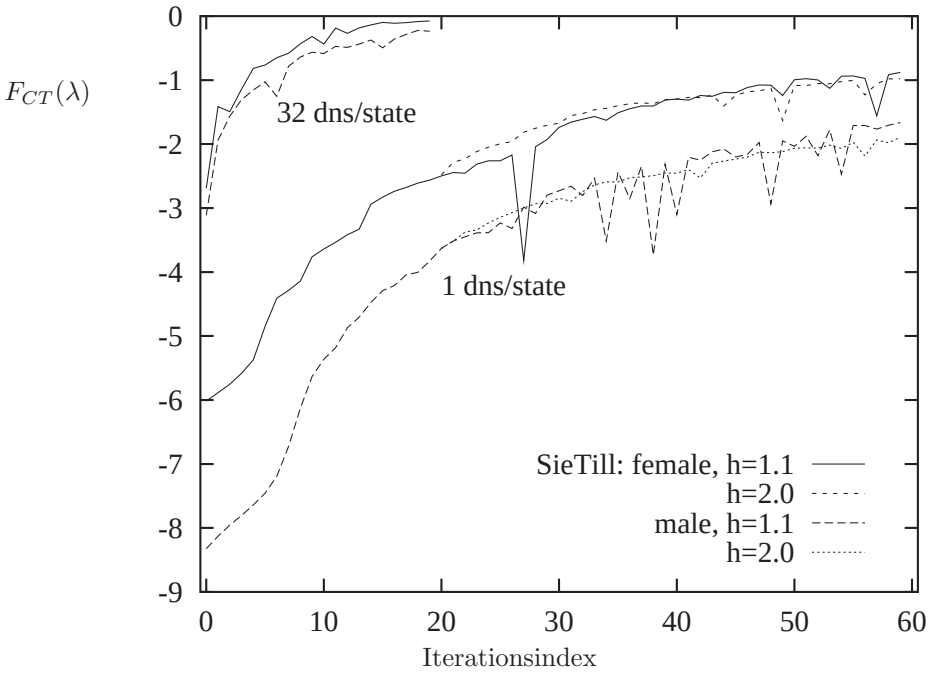


Abbildung 8.4: CT-Kriterium als Funktion des Iterationsindex für Gaußsche Einzel- und Mischverteilungen auf dem SieTill Trainings-Korpus.

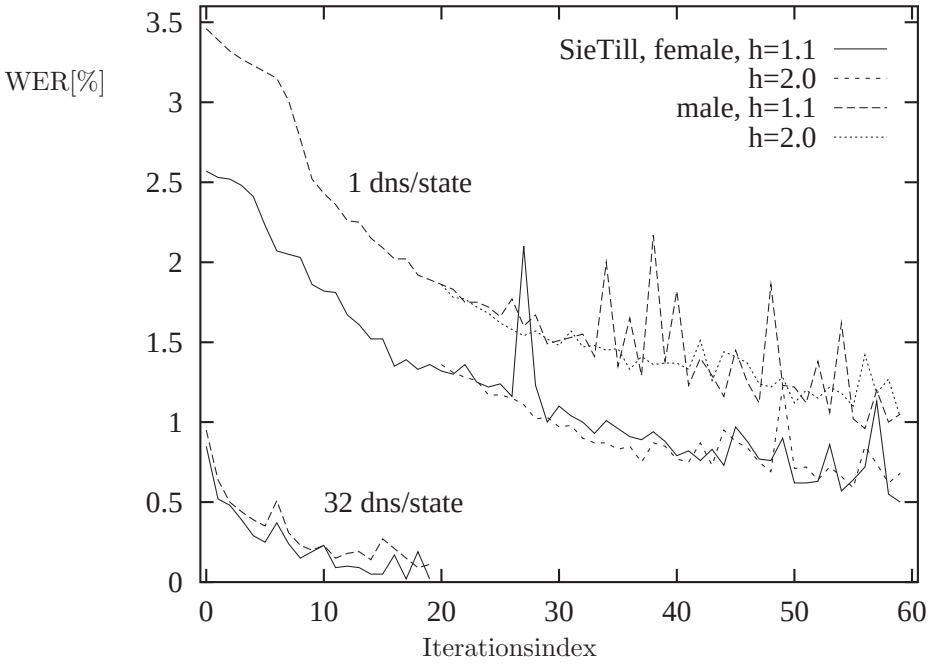


Abbildung 8.5: Wortfehlerrate (WER[%]) auf den Trainingsdaten des SieTill-Korpus als Funktion des Iterationsindex für das *Corrective Training* (CT)-Kriterium unter Verwendung Gaußscher Einzel- und Mischverteilungen.

8.2.5 Vergleich diskriminativer Kriterien

Im folgenden werden die Ergebnisse zur Untersuchung der Kriterien *Corrective-Training* (CT), *Maximum Mutual Information* (MMI), *falsifizierendes Training* (FT) sowie *Minimum-Classification Error* (MCE) vorgestellt. Für die Experimente wurde der in Kapitel 3 beschriebene verallgemeinerte Ansatz für diskriminative Kriterien verwendet. Hierdurch ist ein direkter Vergleich der erzielten Performanz sowie des Verhaltens der Kriterien im Training möglich.

Corrective Training (CT)

Das *Corrective Training* bildet unter den hier betrachteten Kriterien das einfachste Verfahren, da die Menge der konkurrierenden Hypothesen allein auf die beste erkannte Wortfolge eingeschränkt wird. Weil beim CT-Kriterium nur fehlerhaft erkannte Äußerungen zur Diskrimination beitragen, ergeben sich hieraus auch die Grenzen des Verfahrens. So ist eine weitere Optimierung des Trainingskriteriums für den Fall, daß keine oder nur sehr wenige Fehler auf den Trainingsdaten auftreten, nicht mehr möglich.

Für die im Rahmen dieser Arbeit durchgeführten Untersuchungen zum CT-Kriterium wurde der Exponent α zur Umwichtung der Satzwahrscheinlichkeit gemäß der Definition auf 1 gesetzt. Die Bestimmung der besten erkannten Satzhypothese erfolgte unter Verwendung eines *one-pass*-Erkenners, wobei der Suchraum mittels der in Abschnitt 4.3.1 beschriebenen Weise fokussiert wurde. Die folgende Tabelle enthält die gemittelten Laufzeitmessungen für das CT-Kriterium bei Verwendung Gaußscher Einzel- und Mischverteilungen. Die Laufzeitmessungen wurden auf einem ALPHA 5000 PC durchgeführt.

Tabelle 8.6: Echtzeitfaktoren (RTF) und Wortgraphdichten (WGD) bei verschiedener Parameterzahl für die diskriminativen Kriterien *Corrective Training* (CT), *Maximum Mutual Information* (MMI), *Minimum Classification Error* (MCE) und *falsifizierendes Training* (FT).

SieTill: Laufzeitmessungen					
Korpus	LDA	Kriterium	Dichten m+f	WGD	RTF
Train.	**ja**	CT	215 + 215	–	0.1
			6.8K + 6.8K	–	1.1
		MMI, MCE, FT	215 + 215	7.3	0.2
			6.8K + 6.8K	5.4	1.1

Für Einzelverteilungen ergab sich hierbei ein Echtzeitfaktor (RTF) von 0.1 je Iteration. Bei Mischverteilungen stieg der Faktor auf 1.1 an.

Die Erkennungsergebnisse für verschiedene Iterationszahlen bei Verwendung Gaußscher Einzel- und Mischverteilungen sind in Tabelle 8.7 zusammengefaßt. Gegenüber einem konventionellen ML-Training erzielte das CT-Kriterium bei gleicher Anzahl der Parameter

nach 20 Iterationen eine Relativverbesserung von 22% für Einzelverteilungen und 7.6% für Mischverteilungen (32 Dichten pro Zustand) nach 10 Iterationen. Für die Einzelverteilungen konnte die nach 20 Iterationen erzielte Fehlerrate durch weitere 40 Iterationen auf 2.69% gesenkt werden, was einer Relativverbesserung von insgesamt 29% entspricht.

Tabelle 8.7: Fehlerraten für das *Corrective Training* (CT) Kriterium bei verschiedenen Iterationszahlen (Iter.) auf den Trainings- und Testdaten des SieTill-Korpus für Gaußsche Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA. Das CT-Training wurde jeweils auf einem initialen Maximum-Likelihood (ML) Training aufgesetzt

SieTill: CT-Kriterium							
Korpus	LDA	Kriterium	Dichten m+f	Iter.	del - ins - sub	WER[%]	SER[%]
Train.	**ja**	ML	215 + 215	20	236-162- 904	3.04	8.16
			6787+6824	50	106-120- 160	0.90	2.59
		CT	215 + 215	20	218- 83 - 396	1.63	4.60
				30	181- 72 - 287	1.26	3.79
				60	110- 70 - 139	0.56	2.12
			6858+6825	10	28 - 19 - 9	0.11	0.40
			6856+6817	20	13 - 9 - 4	0.06	0.12
Test	**ja**	ML	215 + 215	20	304-270-1053	3.78	9.74
			6787+6824	50	198-201- 450	1.97	5.31
		CT	215 + 215	20	348-191- 726	2.95	7.56
				30	329-201- 695	2.85	7.27
				60	257-223- 679	2.69	6.95
			6858+6825	10	223-128- 433	1.82	4.97
			6856+6817	20	185-214- 405	1.87	5.15

Maximum Mutual Information (MMI)

Im Unterschied zum CT-Kriterium, bei dem nur die beste erkannte Wortfolge in Konkurrenz zur gesprochenen Sequenz gesetzt wird, werden beim MMI-Kriterium sämtliche in einem Wortgraphen enthaltene Satzhypothesen zur Diskrimination herangezogen. Die Wortgraphen wurden hierzu gemäß dem in Abschnitt 4.2 beschriebenen Verfahren synchron zu einer zeitabhängigen Suche generiert, wobei zur Vermeidung von Pfadvervielfältigungen die Wortpaarapproximation sowie das modifizierte Silence-Modell verwendet wurden. Die Bestimmung der Wortwahrscheinlichkeiten erfolgte unter Verwendung des in Abschnitt 3.4 beschriebenen FB-Algorithmus auf Wortgraphen. Abbildung 8.6 zeigt, wie sich die Posterior-Wahrscheinlichkeiten prozentual auf die Kanten des Wortgraphen verteilen. Unter Berücksichtigung der dem Experiment zugrunde liegenden Wortgraphdichte von 5.4 ergibt sich hieraus, daß die beste erkannte Wortfolge im allgemeinen eine Posterior-Wahrscheinlichkeit nahe 1 erhält, während die Posterior-Wahrscheinlichkeiten vieler schlechter bewerteter Hypothesen in der Regel nahe 0 liegen. Obwohl das MMI-Kriterium hierdurch näher an das CT-Kriterium heranrückt, zeigte sich im Verlauf der

Untersuchungen, daß sich gegenüber dem Corrective Training für das MMI-Kriterium in der Regel eine schnellere Konvergenz ergibt. Unter Umständen besteht hier die Möglichkeit, für das MMI-Kriterium einen weiteren Performanzgewinn dadurch zu erzielen, daß die akustischen Wahrscheinlichkeiten im diskriminativen Training zusätzlich mit einem Exponenten gewichtet werden. Dieser Exponent würde die Abstände zwischen den konkurrierenden Klassen verkleinern und somit zu einer gleichmäßigeren Verteilung der Posterior-Wahrscheinlichkeitsmasse auf die konkurrierenden Hypothesen beitragen.

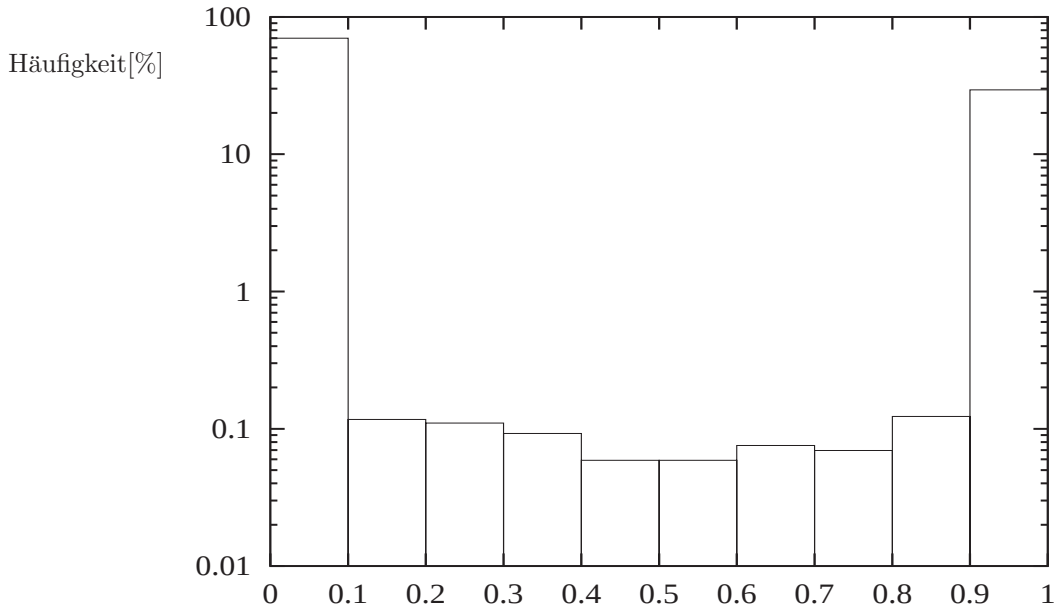


Abbildung 8.6: Prozentuale Verteilung der Posterior-Wahrscheinlichkeiten auf die Kanten eines Wortgraphen der Dichte 5.4 für das MMI-Kriterium unter Verwendung Gaußscher Einzelverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA (männliche Sprecher des SieTill-Trainingskorpus).

Die Laufzeitmessungen für Einzel- und Mischverteilungen sind in Tabelle 8.6 aufgeführt. Gegenüber dem *Corrective Training* ergab sich für das MMI-Kriterium ein Anstieg des Echtzeitfaktors auf 0.2 für Einzelverteilungen aufgrund des zusätzlichen Overheads durch die Bestimmung der Posterior-Wahrscheinlichkeiten für die im Wortgraphen enthaltenen Hypothesen. Für Mischverteilungen ergab sich kein signifikanter Anstieg der Laufzeit, da hier die Abstandsberechnungen den geschwindigkeitsbestimmenden Schritt bildeten.

In Tabelle 8.8 sind die auf den Trainings- und Testdaten des SieTill-Korpus erzielten Fehlerraten für das MMI-Kriterium aufgeführt. Es ist zu beachten, daß das MMI-Training für die Messungen mit „215+215“ und „6858+6825“ Dichten jeweils auf ein über 20 bzw. 10 Iterationen durchgeführtes CT-Training aufgesetzt wurde. Die Messung für „13445+13381“ wurde dagegen unmittelbar auf eine ML-traininerte Referenz aufgesetzt. Gegenüber dem CT-Vergleichsergebnis nach 30 Iterationen ergab sich hier nur eine marginale Relativverbesserung von 1.5% für Einzelverteilungen. Für die Mischverteilungen konnten Verbesserungen von 7% relativ erzielt werden.

Tabelle 8.8: Fehlerraten für das *Maximum Mutual Information* (MMI) Kriterium bei verschiedenen Iterationszahlen (Iter.) auf den Trainings- und Testdaten des SieTill-Korpus für Gaußsche Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA.

SieTill: MMI-Kriterium Iter. (X,Y): $X\times$CT + $Y\times$MMI							
Korpus	LDA	Kriterium	Dichten m+f	Iter.	del - ins - sub	WER[%]	SER[%]
Train.	**ja**	ML	13K + 13K	(0, 0)	54 -118- 76	0.58	1.67
		CT	215 + 215	(30, 0)	181- 72 -287	1.26	3.79
			6856 +6817	(20, 0)	13 - 9 - 4	0.06	0.12
		MMI	215 + 215	(20,10)	349-175-684	2.81	7.13
			6858 + 6825	(10,10)	182-161-407	1.74	4.80
			13K + 13K	(0,15)	7 - 3 - 0	0.02	0.08
Test	**ja**	ML	13K + 13K	(0, 0)	198-162-419	1.81	4.93
		CT	215 + 215	(30, 0)	329-201-695	2.85	7.27
			6856 +6817	(20, 0)	185-214-405	1.87	5.15
		MMI	215 + 215	(20,10)	349-175-684	2.81	7.13
			6858 + 6825	(10,10)	182-161-407	1.74	4.80
			13K + 13K	(0,15)	188-191-393	1.79	4.90

Minimum Classification Error (MCE)

Für die Untersuchungen zum MCE-Kriterium wurden die Wortgraphen in der gleichen Weise wie beim MMI-Kriterium erzeugt. Da für das MCE-Kriterium jedoch die gesprochene Wortfolge aus der Menge der konkurrierenden Hypothesen entfernt werden muß, wurde zusätzlich zu jedem Wortgraphen eine N-Besten-Liste generiert, um all diejenigen Zeitanpassungen der gesprochenen Sequenz im Wortgraphen aufzufinden, deren Satzwahrscheinlichkeit hinreichend nahe bei der besten erkannten Satzhypothese liegen. Der Parameter β aus der für das MCE-Kriterium verwendeten Sigomoidfunktion $1/(1+e^{2\beta z})$ wurde zuvor auf den Trainingsdaten optimiert, wobei sich das Minimum in der Wortfehlerrate für die Einstellung $\beta = 4 \cdot 10^{-5}$ ergab (vgl. Abbildung 8.8). Alle weiteren Experimente zum MCE-Kriterium wurden daher stets mit dieser Wahl des Parameters β durchgeführt. Die Laufzeitmessungen zum MCE-Training entsprachen denen des MMI-Trainings, da die zusätzliche Generierung der N-Besten-Liste in linearer Zeit erfolgt und somit keinen signifikanten Zeitverlust bedeuten (vgl. Tabelle 8.6).

Die auf den Trainings- und Testdaten des SieTill-Korpus ermittelten Fehlerraten sind der Tabelle 8.9 zu entnehmen. Wie beim MMI-Training wurden Trainingsläufe für die Messungen mit je „215+215“ sowie „6858+6825“ Dichten auf ein initiales CT-Training aufgesetzt. Für „13K+13K“ und „25K+24K“ Dichten wurde das MCE-Training unmittelbar zur Fortsetzung eines vorab durchgeführten ML-Trainings herangezogen. Gegenüber dem CT-Vergleichsergebnis nach 30 Iterationen ergab sich bei Verwendung des MCE-Kriteriums

eine Reduktion der Wortfehlerrate von 9.1% relativ für Einzel- und 6.4% relativ für Mischverteilungen (6.8K Dichten) bei gleicher Anzahl der Iterationen. Für 25K Dichten konnte im Vergleich zum ML-Kriterium ein Performanzgewinn von 8.6% relativ erzielt werden.

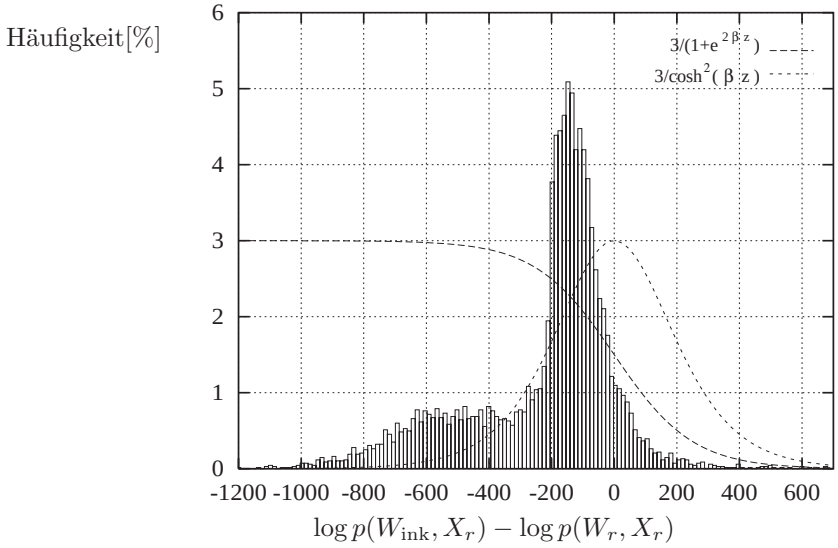


Abbildung 8.7: Histogramm über die Score-Differenzen zwischen gesprochener und bester, falsch erkannter Wortfolge für die männlichen Sprecher des SieTill-Trainingskorpus (Gaußsche Einzelverteilungen mit gepoolter diagonaler Kovarianzmatrix unter Verwendung einer LDA). Auf der Ordinate ist die prozentuale Häufigkeit abgetragen, mit der die entsprechende Score-Differenz für den gegebenen Parametersatz im Trainingskorpus auftrat.

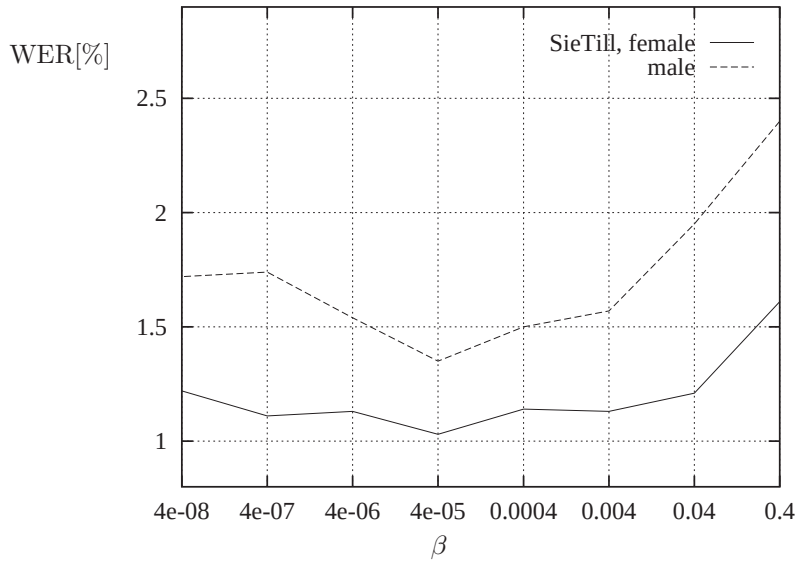


Abbildung 8.8: β-Optimierung auf Trainingsdaten

Tabelle 8.9: Fehlerraten für das *Minimum Classification Error* (MCE) Kriterium bei verschiedenen Iterationszahlen (Iter.) auf den Trainings- und Testdaten des SieTill-Korpus für Gaußsche Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA.

SieTill: MCE-Kriterium Iter. (X,Y): $X\times$CT + $Y\times$MCE							
Korpus	LDA	Kriterium	Dichten m+f	Iter.	del - ins - sub	WER[%]	SER[%]
Train.	**ja**	ML	13K + 13K	(0, 0)	54 -118- 76	0.58	1.67
			25K + 24K	(0, 0)	38 - 99 - 34	0.40	1.17
		CT	215 + 215	(30, 0)	181- 72 -287	1.26	3.79
			6856 +6817	(20, 0)	13 - 9 - 4	0.06	0.12
		MCE	215 + 215	(20,10)	246- 75 -302	1.45	3.97
			6858 + 6825	(10,10)	18 - 19 - 8	0.11	0.35
			13K + 13K	(0,15)	37 - 16 - 32	0.20	0.59
			25K + 24K	(0, 7)	34 - 19 - 20	0.17	0.51
Test	**ja**	ML	13K + 13K	(0, 0)	198-162-419	1.81	4.93
			25K + 24K	(0, 0)	193-169-431	1.85	4.94
		CT	215 + 215	(30, 0)	329-201-695	2.85	7.27
			6856 +6817	(20, 0)	185-214-405	1.87	5.15
		MCE	215 + 215	(20,10)	281-194-643	2.59	6.70
			6858 + 6825	(10,10)	177-157-417	1.75	4.80
			13K + 13K	(0,15)	183-148-398	1.69	4.64
			25K + 24K	(0, 7)	161-160-406	1.69	4.64

Falsifizierendes Training (FT)

Das *falsifizierende Training* beruht auf einer Maximum-Approximation des MCE-Kriteriums, bei der nur die beste, inkorrekt erkannte Wortfolge zur Diskrimination herangezogen wird. Das FT-Kriterium ist somit das Analogon zum CT-Kriterium. Für die verwendete Sigmoidfunktion $1/(1 + e^{2\beta z})$ wurde nach einer Optimierung auf den Trainingsdaten $\beta = 0.07$ gewählt. Die Bestimmung der besten, inkorrekten Satzhypothese erfolgte mittels einer N-Besten-Listengenerierung, über die das erste Auftreten einer zur gesprochenen Wortsequenz abweichenden Wortfolge ermittelt wurde. Die N-Besten-Listen wurden hierzu wie beim MCE-Training auf einem vorab generierten Wortgraphen produziert.

Die Laufzeitmessungen (siehe Tabelle 8.6) entsprechen wiederum denen zum MCE-Training, da die erforderlichen Berechnungsschritte bis auf die Bestimmung der Wortwahrscheinlichkeiten für beide Verfahren im wesentlichen übereinstimmen. Die auf den Trainings- und Testdaten ermittelten Fehlerraten sind in Tabelle 8.10 aufgeführt. Wie beim CT-Kriterium wurde das falsifizierende Training für alle Messungen unmittelbar auf ein ML-Training aufgesetzt. Gegenüber dem konventionellen ML-Kriterium erbrachte das FT-Training hierbei eine Relativverbesserung von 26% für Einzel- und 12.7% für Mischverteilungen (6.8K Dichten).

Tabelle 8.10: Fehlerraten zum *falsifizierenden Training* (FT) bei verschiedenen Iterationszahlen (Iter.) auf den Trainings- und Testdaten des SieTill-Korpus für Gaußsche Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA. Das FT-Training wurde jeweils auf einem initialen Maximum-Likelihood (ML) Training aufgesetzt

SieTill: FT-Kriterium							
Korpus	LDA	Kriterium	Dichten m+f	Iter.	del - ins - sub	WER[%]	SER[%]
Train.	**ja**	ML	215 + 215	20	236-162- 904	3.04	8.16
			6787+6824	50	106-120- 160	0.90	2.59
		FT	215 + 215	20	236- 96 - 371	1.64	4.25
				30	230- 82 - 321	1.48	3.82
			6858+6825	10	70 - 46 - 52	0.39	1.10
Test	**ja**	ML	215 + 215	20	304-270-1053	3.78	9.74
			6787+6824	50	198-201- 450	1.97	5.31
		FT	215 + 215	20	301-288- 718	3.03	7.76
				30	281-274- 650	2.80	7.27
			6858+6825	10	176-158- 409	1.72	4.73

8.2.6 Resümee

In Tabelle 8.11 sind die erzielten Erkennungsergebnisse für die einzelnen Kriterien nochmals zusammengefaßt. Für Gaußsche Einzelverteilungen mit gepoolter diagonaler Kovarianzmatrix sowie der Verwendung einer LDA konnte gegenüber dem konventionellen ML-Ansatz mit Hilfe diskriminativer Verfahren ein Performanzgewinn von 26% relativ erreicht werden. Für Mischverteilungen mit je 6.8K Dichten erzielten die diskriminativen Verfahren eine Relativverbesserung von 15.2%. Bei der Verwendung von 13K Dichten betrug die Relativverbesserung gegenüber dem initialen ML-Ergebnis nur noch 6.6%.

Die Messungen zeigen somit deutlich, daß bei wachsender Zahl der Parameter und der hierdurch bedingten zunehmend detaillierteren Modellierung die Performanz des diskriminativen Trainings abnimmt. Die Ursache hierfür liegt darin, daß mit steigender Parameterzahl bereits durch konventionelle Methoden eine immer bessere Adaption der Trainingsdaten möglich wird (siehe hierzu die Initialfehlerraten der diskriminativen Verfahren auf den Trainingsdaten aus Abbildung 8.10). Dies bestätigt auch das Steigungsverhalten der einzelnen diskriminativen Zielfunktionen (über die Iterationen betrachtet). So ergeben sich mit zunehmender Parameterzahl deutlich flachere Konvergenzkurven (vgl. Abbildung 8.9).

Unter den diskriminativen Verfahren konnten mit Hilfe des MCE-Kriteriums sowie dem falsifizierenden Training (soweit) die besten Resultate erzielt werden. Im Vergleich zum FT- und MCE-Kriterium ist der geringere Performanzgewinn des CT- sowie des MMI-Kriteriums auf die zu hohe Sensibilität gegenüber Ausreißern in den Trainingsdaten zurückzuführen. Während beim FT- und MCE-Kriterium die schlecht erklärbaren Ausreißer durch die Sigmoidfunktion nur mit einem sehr kleinen Gewicht in die Zielgröße eingehen und damit unberücksichtigt bleiben, versuchen sowohl das MMI- als auch das CT-

Kriterium, diese Ausreißer korrekt zu klassifizieren. Dies führt jedoch aufgrund der geringen Generalisierbarkeit solcher Beobachtungen zu Verlusten in der Performanz für die Klassifikation bislang ungesehener Daten.

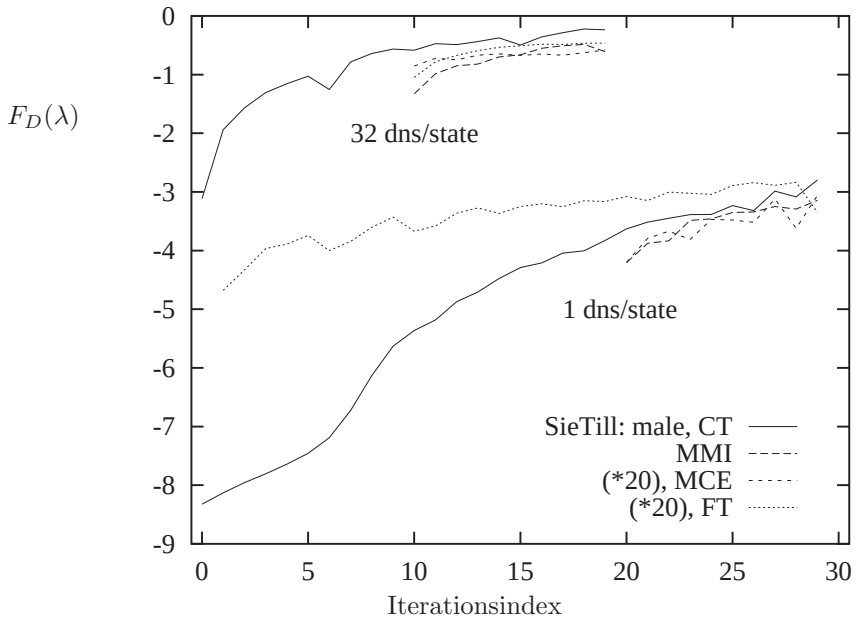


Abbildung 8.9: Zielgrößen verschiedener diskriminativer Kriterien als Funktionen des Iterationsindex auf den Trainingsdaten des SieTill-Korpus für Gaußsche Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und unter Verwendung einer LDA.

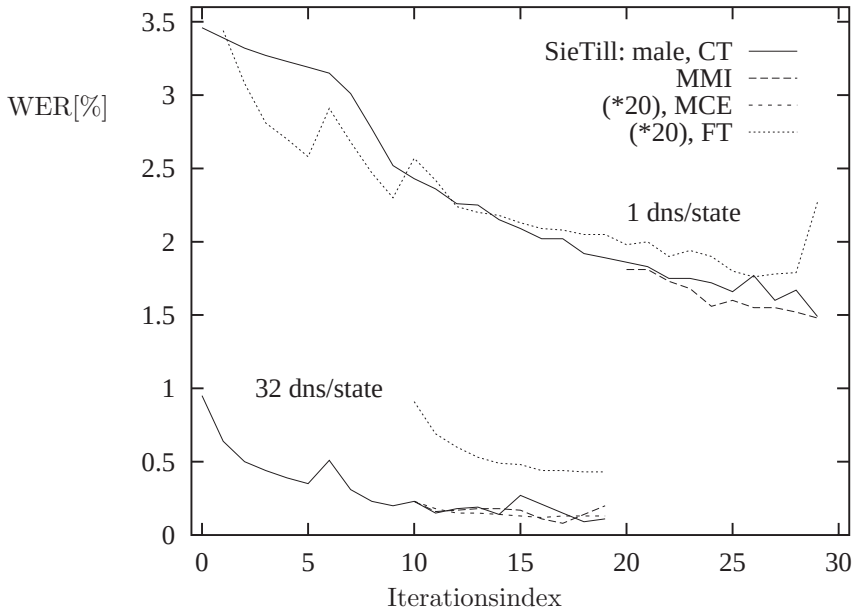


Abbildung 8.10: Wortfehlerraten (WER[%]) für verschiedene diskriminative Kriterien auf den Trainingsdaten des SieTill-Korpus als Funktion des Iterationsindex (Gaußsche Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix unter Verwendung einer LDA).

Tabelle 8.11: Die verschiedenen Trainingskriterien im Vergleich

SieTill: Kriterien						
Korpus	LDA	Kriterium	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**nein**	ML	215+215	503 - 357 - 1117	4.59	11.34
			23.7K+23.7K	**226 - 231 - 507**	**2.24**	**5.87**
	ja	ML	215+215	304 - 270 - 1053	3.78	9.74
			6862+6839	198 - 201 - 450	1.97	5.31
			13K + 13K	**198 - 162 - 419**	**1.81**	**4.93**
		CT	215+215	329 - 201 - 695	2.85	7.27
			6858+6825	**223 - 128 - 433**	**1.82**	**4.97**
		MMI	215+215	349 - 175 - 684	2.81	7.13
			6858+6825	**182 - 161 - 407**	**1.74**	**4.80**
		FT	215+215	281 - 274 - 650	2.80	7.27
			6858+6838	**176 - 158 - 409**	**1.67**	**4.50**
		MCE	215+215	315 - 176 - 657	2.60	6.73
			13472+13397	**183 - 148 - 398**	**1.69**	**4.64**

8.3 Baum-Welch Training im konventionellen und diskriminativen Ansatz

In diesem Abschnitt werden die Ergebnisse für das Baum-Welch (BW) Training im ML-Ansatz vorgestellt. Hierbei steht die Fragestellung im Vordergrund, ob die zuvor durch diskriminative Verfahren erzielte Performanz mit Hilfe des BW-Trainings eingestellt werden kann. Zudem werden erste Ergebnisse für ein diskriminatives Baum-Welch Training vorgestellt und mit den Resultaten des in Abschnitt 8.2.5 diskutierten MMI-Trainings verglichen. Die Erkennungen auf den Testdaten erfolgten stets unter Verwendung eines Viterbi-Dekoders[2] sowie einem Zerogramm-Sprachmodell. Um die Vergleichbarkeit zu den bisherigen Ergebnissen zu gewährleisten, wurden im Baum-Welch Training wie in den vorangehenden Experimenten Gaußsche Emissionsverteilungen mit einer gepoolten diagonalen Kovarianzmatrix verwendet. Für die in einigen Experimenten eingesetzte LDA wurde dieselbe Transformationsmatrix wie in den Messungen zum Abschnitt 8.2 verwendet.

8.3.1 Baum-Welch Training im ML-Ansatz

Im Rahmen der Untersuchungen zum BW-Training wurde zunächst die Performanz eines ML-Trainings unter Verwendung Gaußscher Einzel- und Mischverteilungen mit einer gepoolten diagonalen Kovarianzmatrix sowie einer optionalen LDA ermittelt. Dabei wurde die Frage untersucht, welchen Einfluß die Maximum-Approximation für Mischverteilungen bzw. die exakte Berechnung auf das Erkennungsergebnis haben. Die Fehlerraten zu dieser

[2]Der in Abschnitt 4.4 vorgestellte Suchalgorithmus zum diskriminativen Baum-Welch Training kann konzeptionell sehr einfach auf einen vollwertigen BW-Dekoder erweitert werden. In dieser Arbeit wurde jedoch auf die Erweiterung verzichtet, da der dann erforderliche Vergleich des Baum-Welch-Dekoders mit dem Viterbi-Dekoder den Rahmen dieser Arbeit sprengen würde.

Untersuchung sind in Tabelle 8.12 aufgeführt. Hierbei bedeutet der Eintrag „BW/MAX“ die Verwendung der Maximum-Approximation, während das Kürzel „BW/SUM“ für die exakte Berechnung der Mischverteilungen steht.

Tabelle 8.12: Erkennungsergebnisse auf den Testdaten des SieTill-Korpus für ein Baum-Welch Training (BW) unter Verwendung des ML-Kriteriums. Die Dekodierung der gesprochenen Wortfolge wurde für die Experimente ohne LDA jeweils mit der Maximum-Approximation für Mischverteilungen (BW/MAX) sowie mit der exakten Berechnung (BW/SUM) durchgeführt.

SieTill: Baum-Welch (ML)							
Korpus	LDA	Krit.	Methode	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**nein**	ML	Viterbi	215+215	503 - 357 - 1117	4.59	11.34
				1720+1720	**287 - 324 - 686**	**3.01**	**7.69**
			BW/MAX	215+215	400 - 415 - 1106	4.46	11.19
				1720+1720	**218 - 398 - 655**	**2.95**	**7.54**
			BW/SUM	215+215	400 - 415 - 1106	4.46	11.19
				1720+1720	**215 - 393 - 652**	**2.92**	**7.50**
	ja	ML	Viterbi	215+215	304 - 270 - 1053	3.78	9.74
				1719+1720	**190 - 298 - 579**	**2.48**	**6.47**
			BW/SUM	215+215	330 - 221 - 1059	3.74	9.64
				1720+1720	**198 - 301 - 528**	**2.39**	**6.39**

Wie aus Tabelle 8.12 hervorgeht, ist die exakte Berechnung marginal besser als die Verwendung der Maximum-Approximation. Für die weiteren Untersuchungen zum BW-Training wurde daher stets die exakte Berechnung verwendet[3]. Die aus dem konventionellen ML-Training für die Viterbi-Näherung erzielten Ergebnisse konnten sämtlich mit Hilfe des BW-Trainings eingestellt werden. Die Verbesserungen gegenüber dem ML-Training in der Viterbi-Näherung sind jedoch so gering, daß hier die Aussage legitim erscheint, daß die durch diskriminative Verfahren erzielte Performanz bei Verwendung Gaußscher Emissionsverteilungen mit gepoolter Kovarianzmatrix nicht durch ein Baum-Welch Training im ML-Ansatz eingestellt werden kann. Die durchgeführten Messungen beschränken sich jedoch ausschließlich auf den Fall gepoolter Kovarianzmatrizen, so daß für die gleiche Fragestellung hinsichtlich der Verwendung zustandsabhängiger oder gar dichtespezifischer Kovarianzmatrizen weitere Untersuchungen erforderlich sind.

8.3.2 Baum-Welch Training im diskriminativen Ansatz

Neben dem BW-Training im ML-Ansatz wurden erste Experimente zu einem diskriminativen BW-Training für das MMI-Kriterium durchgeführt. Für die Bestimmung der generalisierten FB-Wahrscheinlichkeiten wurde der in Abschnitt 4.4 beschriebene Suchalgorithmus

[3]Dennoch bestätigt das Ergebnis, daß die in der Praxis häufig verwendete Maximum-Approximation für Mischverteilungen eine legitime Näherung darstellt, bei deren Verwendung ein signifikanter Performanzverlust in der Regel nicht zu erwarten ist.

zur Produktion eines wortbasierten Zustandsgraphen verwendet. Die Suche erfolgte unter Einbeziehung eines Zerogramm-Sprachmodells, wobei die Wortstrafe über die Zustände eines jeden Wortautomaten „verschmiert“ wurde, um zu große Score-Differenzen zwischen den Bewertungen der einzelnen Zeitanpassungspfade bei der Interwort-Rekombination zu vermeiden. Die (unter Verwendung der exakten Berechnung für Mischverteilungen) erzielten Fehlerraten auf den Testdaten sind in Tabelle 8.13 aufgeführt.

Tabelle 8.13: Erkennungsergebnisse auf den Testdaten des SieTill-Korpus für ein diskriminatives Baum-Welch Training unter Verwendung Gaußscher Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer optionalen LDA.

SieTill: Baum-Welch (MMI)							
Korpus	Methode	Krit.	LDA	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**BW/SUM**	ML	nein	215+215	400-415-1106	4.46	11.19
				1720+1720	215-393- 652	2.92	7.50
			ja	215+215	330-221-1059	3.74	9.64
				1720+1720	198-301- 528	2.39	6.39
		MMI	nein	215+215	314-257- 732	3.03	7.91
				1720+1720	285- 92 - 546	2.15	5.75
			ja	215+215	261-252- 695	2.81	7.31
				1720+1720	170-152- 497	1.91	5.21

Gegenüber dem BW-Training im ML-Ansatz erzielte das diskriminative BW-Training unter Verwendung des MMI-Kriteriums eine Relativverbesserung von 32% für Gaußsche Einzelverteilungen mit diagonaler Kovarianzmatrix ohne LDA sowie 25% relativ unter Einsatz einer LDA. Für Mischverteilungen ergab sich ohne LDA ein Performanzgewinn von 26.4% relativ sowie 20.1% relativ mit LDA. Um den Vergleich zu den diskriminativen Verfahren in der Viterbi-Näherung herzustellen, wurde für die Mischverteilungen zusätzlich ein MMI-Training in der Viterbi-Näherung durchgeführt. Die Verläufe der Zielfunktionen sind in Abbildung 8.11 dargestellt. Die Tabelle 8.14 enthält die auf den Testdaten ermittelten Fehlerraten.

Tabelle 8.14: Erkennungsergebnisse auf dem SieTill-Testkorpus für den Vergleich eines diskriminativen MMI-Trainings in der Viterbi-Näherung und dem BW-Training (Gaußsche Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und optionaler LDA). Das Training wurde jeweils mit 20 Iterationen auf ein initiales ML-Training aufgesetzt.

SieTill: Viterbi vs. Baum-Welch (MMI)							
Korpus	Krit.	Dichten m+f	LDA	Meth.	del - ins - sub	WER[%]	SER[%]
Test	MMI	1720+1720	nein	Viterbi	283-200-679	2.70	6.91
				BW	285- 92 -546	2.15	5.75
			ja	Viterbi	250-213-512	2.28	5.97
				BW	170-152-497	1.91	5.21

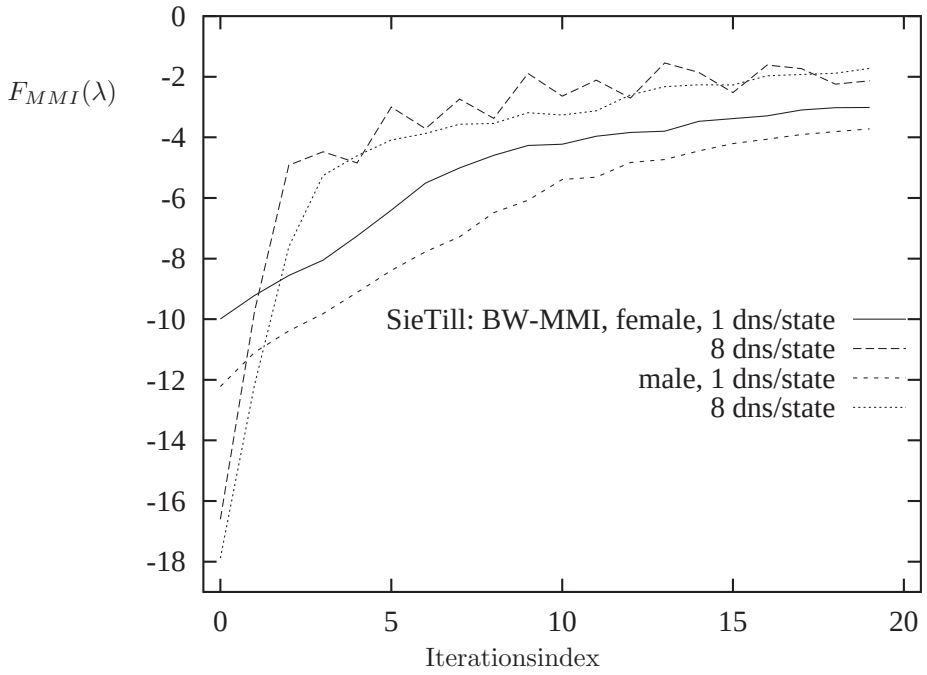


Abbildung 8.11: MMI-Kriterium als Funktion des Iterationsindex für ein diskriminatives Baum-Welch Training auf dem SieTill Trainings-Korpus unter Verwendung Gaußscher Einzel- und Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix unter Verwendung einer LDA.

Für den Vergleich wurde das MMI-Training in beiden Fällen mit je 20 Iterationen bei einer Schrittweite von $h = 1.1$ auf das zugehörige initiale ML-Training aufgesetzt. Beide Messungen wurden sowohl unter Verwendung einer LDA als auch ohne zusätzliche Merkmalstransformation durchgeführt. Gegenüber dem MMI-Training in der Viterbi-Näherung ließen sich unter Verwendung des diskriminativen BW-Trainings für das MMI-Kriterium signifikante Relativverbesserungen in der Wortfehlerrate von 20.3% ohne LDA, sowie 16.2% mit LDA bei Verwendung von Mischverteilungen erzielen. Diese deutliche Verbesserung ist umso höher zu bewerten, da zusätzlich berücksichtigt werden muß, daß für das BW-Training exakt dieselbe LDA-Transformationsmatrix wie für die Messungen zur Viterbi-Näherung verwendet wurden. Für zukünftige Experimente wäre hier ein weiterer Performanzgewinn denkbar, wenn die LDA unmittelbar aus dem BW-Training bestimmt werden würde. Zudem wurde mit der Anzahl verwendeter Mischverteilungsparameter voraussichtlich noch nicht das Minimum in der Wortfehlerrate für das initiale ML-trainierte BW-Ergebnis erreicht, so daß hier ein weiterer Performanzgewinn durch die Erhöhung der Parameter möglich erscheint.

8.4 Experimente zur linearen MMI-Analyse (LMA)

Im Rahmen der Untersuchungen über lineare Merkmalstransformationen wurde die lineare MMI-Analyse (LMA) mit der konventionellen LDA verglichen. Beide Transformationsmatrizen wurden gemäß der in Abschnitt 6.1 bzw. 6.2 beschriebenen Weise im Anschluß an ein ML-Training bzw. ein MMI-Training für Gaußsche Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix bestimmt. Zu jeder Transformationsmatrix wurde anschließend

ein ML-Training durchgeführt. Die erzielten Fehlerraten sind in Tabelle 8.15 aufgeführt.

Tabelle 8.15: Erkennungsergebnisse auf den Testdaten des SieTill-Korpus für den Vergleich der linearen Merkmalstransformationen (LT) *lineare MMI-Analyse* (LMA) und *lineare Diskriminanzanalyse* (LDA) nach einem ML-Training (Gaußsche Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix).

SieTill: LDA vs. LMA						
Korpus	Kriterium	LT	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**ML**	LDA	215+215	304-270-1053	3.78	9.74
			430+430	234-318- 824	3.19	8.31
			860+860	198-329- 652	2.74	7.18
			1719+1720	198-298- 579	2.48	6.47
			3435+3436	191-294- 499	2.28	5.92
		LMA	215+215	320-214- 985	3.53	9.18
			430+430	236-298- 754	2.99	7.87
			860+860	212-343- 622	2.73	7.15
			1720+1719	218-317- 531	2.48	6.48
			3428+3435	197-304- 464	2.24	5.83

Verbesserungen gegenüber der konventionellen LDA waren hier jedoch nur für kleine Parameterzahlen zu beobachten. So beträgt die Relativverbesserung für Einzelverteilungen 6.6%, geht jedoch dann bei zunehmender Parameterzahl zurück. Daß die Performanz trotz der auf den Trainingsdaten gemessenen Relativverbesserung von 16.8% für die männlichen Sprecher und 9% für die weiblichen Sprecher (vgl. hierzu die Abbildungen in Abschnitt 6.2.2) nicht größer ausfällt, ist auf eine zu geringe Generalisierbarkeit der Beobachtungen im Training zurückzuführen, bzw. darauf, daß die lineare MMI-Analyse zu lokal operiert. Hier sollte für zukünftige Untersuchungen eine iterative Form zur Optimierung der LMA in Betracht gezogen werden.

8.5 Splitting im diskriminativen Training

Das Aufspalten der Mischverteilungen im diskriminativen Training erfolgte gemäß dem in Kapitel 7 beschriebenen Algorithmus. Hierzu wurden zunächst die Gaußschen Einzelverteilungen mit einer gepoolten diagonalen Kovarianzmatrix unter Verwendung einer LDA mit Hilfe des ML-Kriteriums trainiert, wobei vor jedem Aufspalten der Mischungskomponenten ein zusätzlicher MMI-Schritt durchgeführt wurde. Die zu splittenden Komponenten wurden gemäß der größten auftretenden diskriminativen Mittelwerte $\Gamma_{s,l}(1)$ bestimmt. Nach jedem MMI-Schritt wurde das Training der neuen Mischverteilungsparameter mit je 6 weiteren Iterationen eines konventionellen ML-Trainings fortgesetzt. Die auf den Testdaten des SieTill-Korpus erzielten Fehlerraten sind in Tabelle 8.16 aufgeführt.

Tabelle 8.16: Erkennungsergebnisse auf den Testdaten des SieTill-Korpus zum konventionellen und diskriminativen Split-Kriterium für Gaußsche Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix unter Verwendung einer LDA.

SieTill: Splitting							
Korpus	LDA	Krit.	Split-Krit.	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**ja**	ML	konv.	215+215	304 - 270 - 1053	3.78	9.74
				430+430	234 - 318 - 824	3.19	8.31
				860+860	198 - 329 - 652	2.74	7.18
				1719+1720	190 - 298 - 579	2.48	6.47
				3435+3436	191 - 294 - 499	2.28	5.92
				6862+6839	198 - 201 - 450	1.97	5.31
				13K+13K	**198 - 162 - 419**	**1.81**	**4.93**
				25K+24K	193 - 169 - 431	1.85	4.94
			diskr.	215+215	304 - 270 - 1053	3.78	9.74
				323+323	274 - 266 - 851	3.23	8.42
				404+485	266 - 204 - 754	2.84	7.50
				605+724	243 - 172 - 623	2.41	6.42
				901+1075	262 - 155 - 562	2.28	6.04
				1328+1604	236 - 126 - 506	2.02	5.41
				1924+2270	225 - 131 - 477	1.94	5.26
				2767+3117	182 - 133 - 448	1.77	4.77
				3840+4007	187 - 128 - 447	1.77	4.77
				5069+5059	199 - 113 - 458	1.79	4.82
				6314+6027	187 - 99 - 429	1.66	4.49
				7314+6777	**176 - 101 - 418**	**1.61**	**4.42**
				8449+7402	179 - 100 - 427	1.61	4.50
				9339+7976	179 - 99 - 415	1.61	4.44

Gegenüber dem konventionellen Split-Kriterium ergab sich hierbei eine Relativverbesserung von 11%. Zu beachten ist, daß für dieses Resultat schon die halbe Parameterzahl ausreichte. Entsprechend ergibt sich bei vergleichbarer Parameterzahl eine Relativverbesserung von 18.3%.

Der Erfolg des neuen Split-Kriteriums beruht im wesentlichen auf einer sehr viel differenzierteren Auswahl der aufzuspaltenden Mischungskomponenten. Während das konventionelle Split-Kriterium viele Mischungskomponenten auch dann aufspaltet, wenn sie die Beobachtungen bereits adäquat erklären können, selektiert das neue Kriterium gezielt diejenigen Mischungskomponenten, die auch nach vielen Iterationen die Beobachtungen nur mit einer großen Fehlerwahrscheinlichkeit erklären können. Damit führt das neue Split-Kriterium nicht nur zu besseren Fehlerraten, sondern benötigt darüber hinaus auch weniger Parameter. Die Abbildung 8.12 zeigt, wieviele Mischungskomponenten nach dem neuen Kriterium auf jeden Zustand der Ganzwortmodelle entfielen. Zum Vergleich ist die Zahl der Mischungskomponenten, die sich nach dem konventionellen Split-Kriterium ergab, als gestrichelte Linie eingezeichnet.

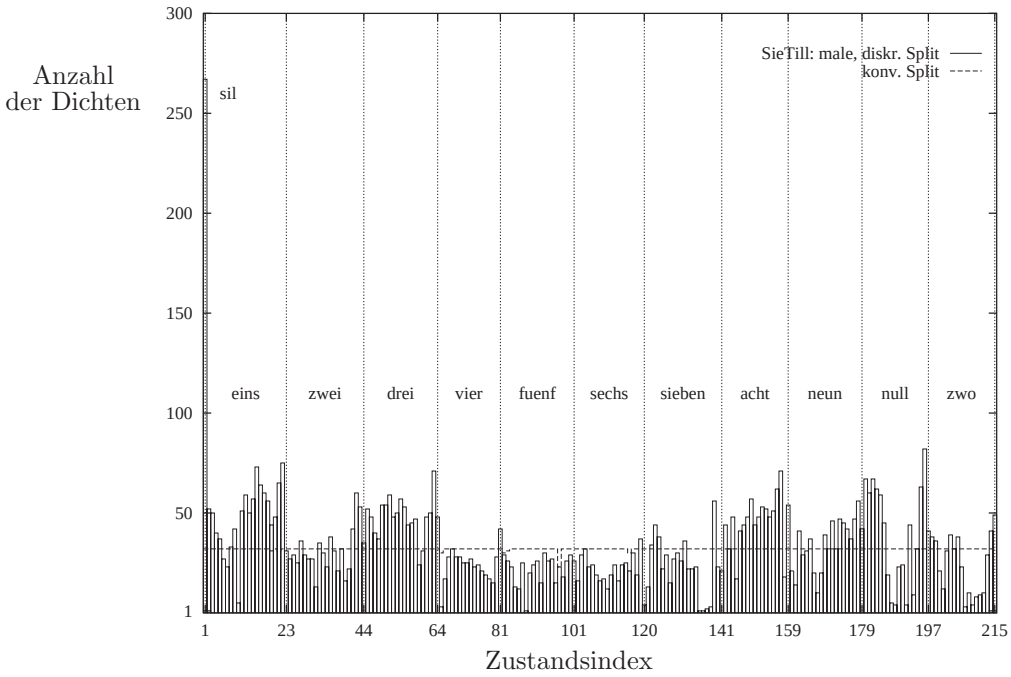


Abbildung 8.12: Anzahl der Mischungskomponenten, die nach dem diskriminativen Splitting auf jeden einzelnen Zustand des Lexikons für den SieTill-Korpus entfallen. Zum Vergleich ist die Zahl der Mischungskomponenten aus dem konventionellen Split-Kriterium als gestrichelte Linie eingezeichnet

Die erzielte Wortfehlerrate von 1.61% ließ sich durch ein zusätzliches diskrmininatives Training unter Verwendung des MMI-Kriteriums nicht weiter verbessern:

Tabelle 8.17: MMI-Training auf bestem diskriminativen Split-Ergebnis (10 Iterationen)

SieTill: Diskriminativer Split + MMI						
Korpus	LDA	Kriterium	Dichten m+f	del - ins - sub	WER[%]	SER[%]
Test	**ja**	MMI-Spl.-ML	7314+6777	176-101-418	1.61	4.42
		MMI	7304+6792	172-103-417	1.61	4.46

8.6 Resultate anderer Forschungsgruppen

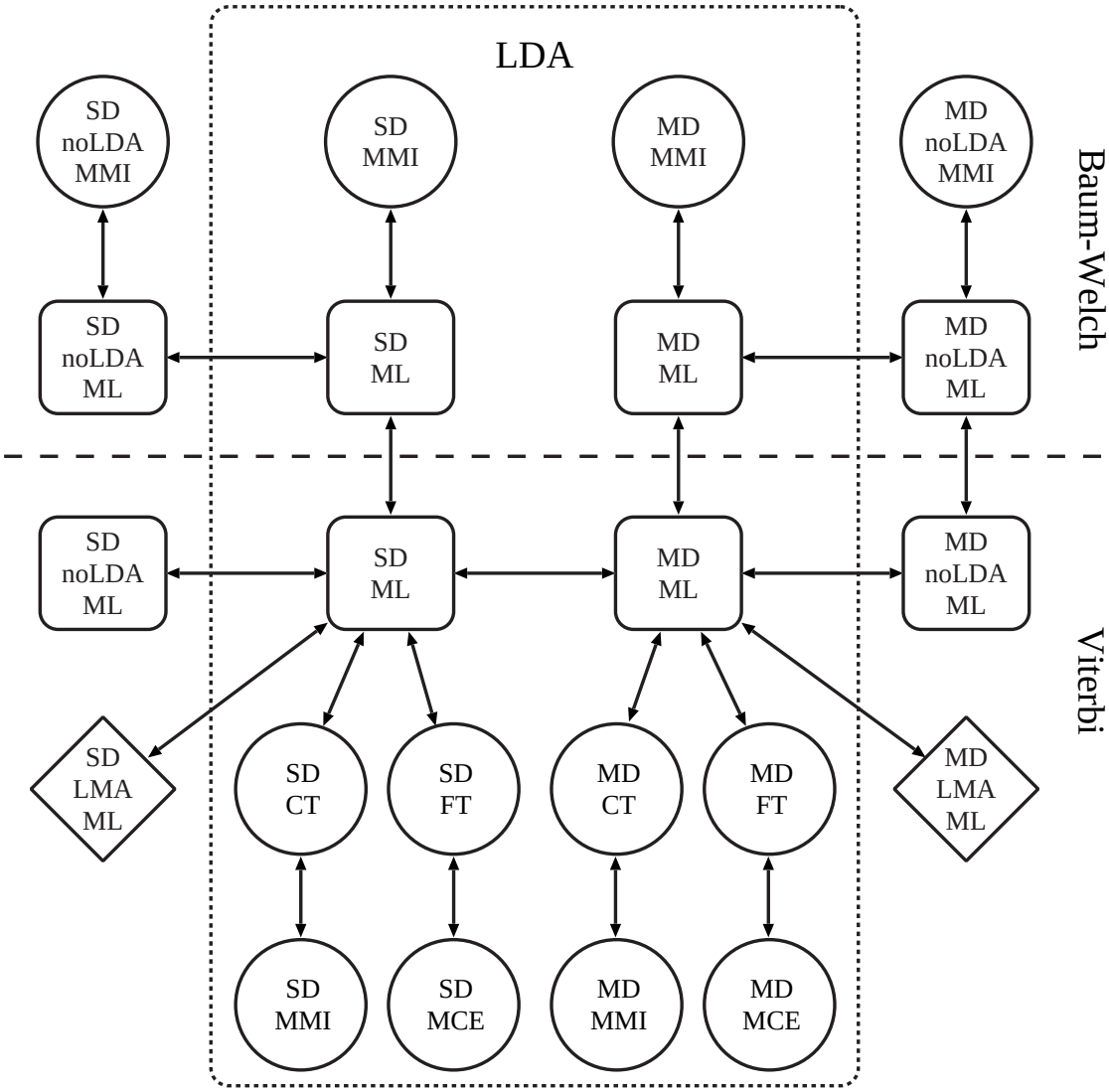
In [Eisele et al. 96] wird für ein konventionelles ML-Training unter Verwendung Gaußscher Mischverteilungen mit gepoolter diagonaler Kovarianzmatrix und einer LDA auf den Testdaten des SieTill-Korpus eine Wortfehlerrate von 2.5% erzielt. Für ein diskriminatives Training unter Verwendung des CT-Kriteriums wird von einer zur Siemens AG München gehörenden Forschungsgruppe eine Fehlerrate von 2.4% berichtet. Diese Angabe ist jedoch noch nicht publiziert und kann somit nur als Anhaltspunkt für den gegenwärtigen Stand gesehen werden.

8.7 Zusammenstellung der wichtigsten Resultate

Die folgende Tabelle 8.18 stellt nochmals die wichtigsten Resultate zusammen, die im Rahmen dieser Arbeit auf dem SieTill-Korpus erzielt worden sind. Die nachfolgende Abbildung veranschaulicht hierzu das Schema des zugrundegelegten Vergleichs.

Tabelle 8.18: Übersicht über die wichtigsten Ergebnisse

SieTill: Kriterien							
Korpus	Dichten m+f	Krit.	LT	Meth.	del - ins - sub	WER[%]	SER[%]
Test	215+215	ML	–	Vit.	503 - 357 - 1117	4.59	11.34
				BW	400 - 415 - 1106	4.46	11.19
			LDA	Vit.	304 - 270 - 1053	3.78	9.74
				BW	330 - 221 - 1059	3.74	9.64
			LMA	Vit.	320 - 214 - 985	3.53	9.18
		CT	LDA	Vit	329 - 201 - 695	2.85	7.27
		MMI	LDA	Vit	349 - 175 - 684	2.81	7.13
				BW	261 - 252 - 695	2.81	7.31
		FT	LDA	Vit	281 - 274 - 650	2.80	7.27
		MCE	LDA	Vit	**315 - 176 - 657**	**2.60**	**6.73**
	1720+1720	ML	–	Vit.	287 - 324 - 686	3.01	7.69
				BW	215 - 393 - 652	2.92	7.50
			LDA	Vit.	190 - 298 - 573	2.48	6.47
				BW	198 - 301 - 528	2.39	6.39
		MMI	–	Vit.	283 - 200 - 679	2.70	6.91
				BW	285 - 92 - 546	2.15	5.75
			LDA	Vit.	250 - 213 - 512	2.28	5.97
				BW	**170 - 152 - 497**	**1.91**	**5.21**
	23.7K+23.7K	ML	–	Vit.	226 - 231 - 507	2.24	5.87
	6.8K+6.8K	ML	LDA	Vit.	198 - 201 - 450	1.97	5.31
	13.4K+13.4K	ML	LDA	Vit.	198 - 162 - 419	1.81	4.93
	6.8K+6.8K	CT	LDA	Vit.	223 - 128 - 433	1.82	4.97
		MMI	LDA	Vit.	182 - 161 - 407	1.74	4.80
		FT	LDA	Vit.	**176 - 158 - 409**	**1.67**	**4.50**
	13.4K+13.3K	MCE	LDA	Vit.	183 - 148 - 398	1.69	4.64
	7.3K+6.7KK	MMI-Spl.-ML	LDA	Vit.	**176 - 101 - 418**	**1.61**	**4.42**



SD:	Einzelverteilungen	ML:	Maximum Likelihood
MD:	Mischverteilungen	MMI:	Maximum Mutual Information
noLDA:	keine LDA	CT:	Corrective Training
LDA:	lineare Diskriminanzanalyse	MCE:	Minimum Classification Error
LMA:	lineare MMI-Analyse	FT:	Falsifizierendes Training

Kapitel 9

Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Fülle verschiedenster diskriminativer Verfahren für das Anwendungsgebiet der automatischen Spracherkennung untersucht. Den Schwerpunkt der Untersuchungen bildete hierbei ein Vergleich der diskriminativen Kriterien *Corrective Training* (CT), *Maximum Mutual Information* (MMI), *falsifizierendes Training* (FT) und *Minimum Classification Error* (MCE). Die Untersuchungen wurden auf der Basis eines verallgemeinernden diskriminativen Ansatzes geführt, um einen direkten Vergleich der durch die einzelnen Verfahren erzielten Performanz zu ermöglichen. Für die Parameteroptimierung wurden der *erweiterte Baum* (EB) Algorithmus sowie ein *Gradientenverfahren* (GD) verwendet. Hierbei ließen sich für eine spezielle Wahl der Schrittweiten im Gradientenverfahren formal nahezu identische Reestimationsgleichungen zu denen aus dem EB-Algorithmus herleiten. Die formale Übereinstimmung konnte experimentell durch das Verhalten der CT-Zielfunktion sowie der Wortfehlerrate auf den Trainings- und Testdaten für Einzel- und Mischverteilungen bestätigt werden.

Im Rahmen der Untersuchungen zum MMI- und MCE-Kriterium wurde zur Bestimmung der generalisierten *Forward-Backward* (FB) Wahrscheinlichkeiten ein FB-Algorithmus auf Wortgraphen implementiert und erfolgreich angewendet. Für die Wortgraphproduktion wurde hierzu eigens eine zeitabhängig organisierte Suche implementiert, die bereits während des Suchprozesses die Bildung von Pfadvervielfältigungen im Wortgraphen durch geeignete Modellierungen (Wortpaarapproximation und modifiziertes Silence-Modell) einzuschränken vermag. Der Einsatz eines Wortgraphen zur Berechnung der Wortwahrscheinlichkeiten im Fall des MCE-Kriteriums erfolgte in dieser Form sogar erstmals in der Spracherkennung. Einen wichtigen Aspekt dieser Arbeit bildeten in diesem Zusammenhang auch die neu entwickelten Techniken, um die Satzwahrscheinlichkeit der gesprochenen Wortfolge aus der Berechnung der generalisierten FB-Wahrscheinlichkeiten nachträglich wieder herausrechnen zu können.

Um den Einfluß der Viterbi-Näherung auf die Performanz des diskriminativen Trainings zu untersuchen, wurden verschiedene Experimente für ein Baum-Welch (BW) Training im ML-Ansatz durchgeführt. Zusätzlich konnten erste Ergebnisse für ein diskriminatives BW-Training unter Verwendung des MMI-Kriteriums präsentiert werden. Einen Schwerpunkt der Arbeit bildete im Rahmen dieser Untersuchung die Entwicklung und Implementierung eines Algorithmus zur Generierung wortbasierter Zustandsgraphen.

Im Rahmen der Experimente über lineare Merkmalstransformationen wurde ein Vergleich der *linearen Diskriminanzanalyse* (LDA) mit der *linearen MMI-Analyse* (LMA) durchgeführt. Abschließend wurde eine neue Methode zur Aufspaltung der Mischverteilungen im diskriminativen Training entwickelt und untersucht.

Experimente wurden auf den beiden Korpora TI-Digits und SieTill für die sprecherunabhängige Erkennung kontinuierlich gesprochener Ziffernketten amerikanischer bzw. deutscher Sprache durchgeführt. Für die diskriminativen Verfahren konnten bei Verwendung der Viterbi-Näherung gegenüber den initialen ML-Ergebnissen Relativverbesserungen in der Wortfehlerrate bis zu 31.2% für Einzelverteilungen, sowie 15.2% für Mischverteilungen (6.8K Dichten) erzielt. werden. Tendenziell führte die Verwendung des MCE- und des FT-Kriteriums gegenüber dem MMI- und CT-Kriterium stets zu höheren Performanzgewinnen (bis zu 7.4% im Fall der Einzelverteilungen), was auf eine zu große Sensibilität des MMI- bzw. CT-Kriteriums auf Ausreißer in den Trainingsdaten zurückgeführt werden konnte. Ein weiteres wichtiges Ergebnis bildete der experimentelle Nachweis, daß der überwiegende Anteil der Wahrscheinlichkeitsmasse bei der Berechnung der generalisierten FB-Wahrscheinlichkeiten auf diejenigen Hypothesen im Wortgraphen entfällt, die zur besten erkannten Wortfolge gehören. Hier sollte für zukünftige Untersuchungen eine Wichtung der akustischen Wahrscheinlichkeiten mit einem geeignet zu wählenden Exponenten aus $(0, 1]$ in Betracht gezogen werden. Dieser Wichtungsexponent verringert die Abstände zwischen den konkurrierenden Hypothesen, was in der Konsequenz zu einer Umverteilung der FB-Wahrscheinlichkeitsmasse zugunsten einer größeren Zahl alternativer Hypothesen im Wortgraphen führt.

Die Verwendung der LMA erbrachte gegenüber der konventionellen LDA nur für kleine Parameterzahlen (Einzelverteilungen) signifikante Gewinne in der Performanz der Wortfehlerrate (5.6% relativ). Da die in die Berechnung der Transformationsmatrizen eingehenden diskriminativen Gewichte hier nur aus einem einzelnen MMI-Schritt gewonnen wurden, legt dies für zukünftige Untersuchungen ein iteratives Training der LMA nahe.

Für die Untersuchungen zum Vergleich des Baum-Welch Trainings mit der Viterbi-Näherung ergaben sich im Unterschied zu den eher geringfügigen Performanzgewinnen beim ML-Kriterium deutliche Verbesserungen im erstmals in der Spracherkennung eingesetzten diskriminativen BW-Training. Obwohl die für das BW-Training verwendete LDA-Transformationsmatrix zur Wahrung der Vergleichbarkeit identisch zu der aus dem Viterbi-Training gewählt wurde, ließen sich bei gleicher Parameter- und Schrittzahl deutliche Relativverbesserungen in der Wortfehlerrate von 16.2% für Mischverteilungen gegenüber dem MMI-Training in Viterbi-Näherung erzielen (ohne LDA betrug die Relativverbesserung in der Wortfehlerrate zum entsprechenden Viterbi-Vergleichsergebnis sogar 20.3%.). In diesem Zusammenhang ist für zukünftige Untersuchungen vor allem die Frage interessant, ob und wenn ja, in welcher Größenordnung die Relativverbesserung liegt, wenn die LDA unmittelbar aus dem BW-Training gewonnen wird. Zudem sind weitere Messungen für größere Parameterzahlen sinnvoll, da mit der hier verwendeten Zahl an Mischungskomponenten das Minimum in der Wortfehlerrate für das ML-Initialergebnis voraussichtlich noch nicht erreicht worden ist. Ferner können sich aus der Fragestellung bezüglich des Performanzgewinnes bei Verwendung des MCE-Kriteriums im diskriminativen BW-Training weitere interessante Einblicke in das diskriminative Training gewinnen lassen. Die praktische Umsetzung dürfte sich hierbei aufgrund des in dieser Arbeit konsequent verwendeten

verallgemeinerten Ansatzes für diskriminative Kriterien nahtlos an die nun verfügbare Software anschließen lassen.

Die Ergebnisse zum Splitting im diskriminativen Training führten auch im Vergleich zum diskriminativen Training zu dem bislang besten bekannten Resultat auf dem SieTill-Korpus (Relativverbesserung in der Wortfehlerrate von 15.7% gegenüber dem ML-Ergebnis mit vergleichbarer Parameterzahl). Der Erfolg dieses Verfahrens beruht hauptsächlich auf einer sehr viel differenzierteren, durch das MMI-Kriterium bestimmten Auswahl der zu splittenden Parameter, sowie der Reestimation der neuen Verteilungsparameter unter Verwendung des konventionellen ML-Kriteriums. Hier ist die Übertragung der Methode auf andere Korpora, insbesondere für großes Vokabular von besonderem Interesse.

Die erzielten Verbesserungen legen beim Entwurf eines Spracherkennungssystem für die Lernphase somit ein diskriminatives BW-Training nahe, das unter Verwendung des MMI- oder des MCE-Kriteriums arbeitet und die Methode des diskriminativen Splittens zur Erhöhung der Verteilungsparameter einsetzt.

Anhang A

Detaillierte Rechnungen

A.1 Differentiation von Emissionswahrscheinlichkeiten

$$
\begin{aligned}
\frac{\partial \log p(x_1^T|W)}{\partial \theta_s} &= \frac{1}{p(x_1^T|W)} \frac{\partial}{\partial \theta_s} \sum_{[s_1,\ldots,s_T]|W} p(x_1^T, s_1^T) \\
&= \frac{1}{p(x_1^T|W)} \frac{\partial}{\partial \theta_s} \sum_{[s_1,\ldots,s_T]|W} \prod_{\tau=1}^{T} p(x_\tau, s_\tau | x_1^{\tau-1}, s_1^{\tau-1}) \\
&= \frac{1}{p(x_1^T|W)} \frac{\partial}{\partial \theta_s} \sum_{[s_1,\ldots,s_T]|W} \prod_{\tau=1}^{T} p(x_\tau|s_\tau)\, p(s_\tau|s_{\tau-1}) \\
&= \frac{1}{p(x_1^T|W)} \sum_{[s_1,\ldots,s_T]|W} \sum_{t=1}^{T} \left[\frac{\partial}{\partial p(x_t|s)} \prod_{\tau=1}^{T} p(x_\tau|s_\tau)\, p(s_\tau|s_{\tau-1}) \right] \frac{\partial p(x_t|s)}{\partial \theta_s} \\
&= \frac{1}{p(x_1^T|W)} \sum_{t=1}^{T} \frac{\partial p(x_t|s)}{\partial \theta_s} \cdot \sum_{[s_1^{t-1}, s_{t+1}^T]|W} \delta_{s,s_t} \left[p(s_t|s_{t-1}) \prod_{\tau\neq t,\tau=1}^{T} p(x_\tau|s_\tau) p(s_\tau|s_{\tau-1}) \right] \\
&= \frac{1}{p(x_1^T|W)} \sum_{t=1}^{T} \frac{1}{p(x_t|s)} \frac{\partial p(x_t|s)}{\partial \theta_s} \cdot \sum_{[s_1^{t-1}, s_t=s, s_{t+1}^T]|W} \left[\prod_{\tau=1}^{T} p(x_\tau|s_\tau) p(s_\tau|s_{\tau-1}) \right] \\
&= \sum_{t=1}^{T} \frac{\sum_{[s_1^{t-1}, s_{t+1}^T]|W} p(x_1^T, s_1^T, s_t = s)}{p(x_1^T|W)} \frac{\partial \log p(x_t|s)}{\partial \theta_s} \\
&= \sum_{t=1}^{T} \frac{p(x_1^T, s_t = s)}{p(x_1^T|W)} \frac{\partial \log p(x_t|s)}{\partial \theta_s} \\
&= \sum_{t=1}^{T} p(s_t = s|x_1^T, W) \frac{\partial \log p(x_t|s)}{\partial \theta_s}
\end{aligned}
$$

A.2 Variante zur Kullback-Leibler-Statistik

$$
\begin{aligned}
Q(\lambda,\overline{\lambda}) &= \sum_{r=1}^{R} \sum_{[s_1^{T_r}]|W_r} p_\lambda(s_1^{T_r}|X_r)\cdot \log p_{\overline{\lambda}}(X_r, s_1^{T_r}) \\
&= \sum_{r=1}^{R} \frac{1}{p_\lambda(X_r)} \cdot \sum_{[s_1^{T_r}]|W_r} p_\lambda(s_1^{T_r}, X_r)\cdot \log p_{\overline{\lambda}}(X_r, s_1^{T_r}) \\
&= \sum_{r=1}^{R} \frac{1}{p_\lambda(X_r)} \cdot \sum_{[s_1^{T_r}]|W_r} p_\lambda(s_1^{T_r}, X_r)\cdot \log \prod_{t=1}^{T_r} p_{\overline{\lambda}}(x_{r,t}|s_t)\cdot p(s_t|s_{t-1}) \\
&= \sum_{r=1}^{R} \frac{1}{p_\lambda(X_r)} \cdot \sum_{[s_1^{T_r}]|W_r} p_\lambda(s_1^{T_r}, X_r)\cdot \sum_{t=1}^{T_r} \log \Big[p_{\overline{\lambda}}(x_{r,t}|s_t)\cdot p(s_t|s_{t-1})\Big] \\
&= \sum_{r=1}^{R}\sum_{t=1}^{T_r} \Big[p_{\overline{\lambda}}(x_{r,t}|s_t) + \log p(s_t|s_{t-1})\Big] \sum_{\sigma,s} \sum_{[s_1^{t-2},s_{t-1}=\sigma,s_t=s,s_{t+1}^{T_r}]|W_r} \frac{p_\lambda(X_r, s_1^{T_r})}{p_\lambda(X_r)} \\
&= \sum_{r=1}^{R}\sum_{t=1}^{T_r} \log p_{\overline{\lambda}}(x_{r,t}|s_t) \sum_{\sigma,s} p_\lambda(s_{t-1}=\sigma, s_t=s|X_r,W_r) \\
&\quad + \sum_{r=1}^{R}\sum_{t=1}^{T_r} \log p(s_t|s_{t-1}) \sum_{\sigma,s} p_\lambda(s_{t-1}=\sigma, s_t=s|X_r,W_r) \\
&= \sum_{r=1}^{R}\sum_{t=1}^{T_r} \log p_{\overline{\lambda}}(x_{r,t}|s_t) \sum_{s} p_\lambda(s_t=s|X_r,W_r) \\
&\quad + \sum_{r=1}^{R}\sum_{t=1}^{T_r} \log p(s_t|s_{t-1}) \sum_{\sigma,s} p_\lambda(s_{t-1}=\sigma, s_t=s|X_r,W_r) \\
&= \sum_{s}\sum_{r=1}^{R}\sum_{t=1}^{T_r} p_\lambda(s_t=s|X_r,W_r)\cdot \log p_{\overline{\lambda}}(x_{r,t}|s_t) \\
&\quad + \sum_{\sigma,s}\sum_{s}\sum_{r=1}^{R}\sum_{t=1}^{T_r} p_\lambda(s_{t-1}=\sigma, s_t=s|X_r,W_r)\cdot \log p(s_t|s_{t-1}) \\
&= \sum_{s}\sum_{r=1}^{R}\sum_{t=1}^{T_r} \Big[\underbrace{\gamma_{r,t}(s|W_r)\cdot \log p_{\overline{\lambda}}(x_{r,t}|\overline{\theta}_{s_t})}_{=:q_1} + \underbrace{\sum_{\sigma} \gamma_{r,t}(\sigma, s|W_r)\cdot \log p(s|\sigma)}_{=:q_2}\Big]
\end{aligned}
$$

Literaturverzeichnis

[Ayer 92] Colin Malcolm Ayer, *A Discriminatively Derived Linear Transform Capable of Improving Speech Recognition Accuracy*, Ph.D. dissertation, Universiy of London, London, Juni 1992.

[Ayer et al. 93] C. M. Ayer, M. J. Hunt und D. M. Brookes, *A Discriminative Derived Transform for Improved Speech Recognition*, ESCA European Conference On Speech Communications And Technology (Berlin, Germany), Bd. 1, September 1993, S. 583–586.

[Bahl et al. 86] L. R. Bahl, P. F. Brown, P. V. Souza und R. L. Mercer, *Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (Tokyo, Japan), 1986.

[Bauer 97] Josef G. Bauer, *Enhanced Control and Estimation of Parameters for a Telephone Based Isolated Digit Recognizer*, IEEE International Conference on Acoustics, Speech and Signal Processing (Munich, Germany), Bd. 2, April 1997, S. 1531–1534.

[Baum und Eagon 67] L. E. Baum und J. A. Eagon, *An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology*, Bulletin of the American Mathematical Society **73** (1967), S. 360–363.

[Bellman 61] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[Biem und Katagiri 93] A. Biem und S. Katagiri, *Feature Extraction Based on Minimum Classification Error/Generalized Probabilistic Descent Method*, IEEE International Conference on Acoustics, Speech and Signal Processing (Minneapolis, Minnesota, USA), Bd. 2, April 1993, S. 275–278.

[Bridle 90] John S. Bridle, *Alpha-Nets: A Recurrent 'Neural' Network Architecture with a Hidden Markov Model Interpretation*, Speech Communication (1990), Nr. 9, S. 83–92.

[Bridle et al. 82] J. Bridle, M. Brown und R. Chamberlain, *An Algorithm for Connected Word Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (Paris, France), 1982, S. 899–902.

[Cardin et al. 92] R. Cardin, Y. Normandin und R. De Mori, *High Performance Connected Digit Recognition Using Codebook Exponents*, IEEE International Conference on Acoustics, Speech and Signal Processing (San Francisco, California, USA), Bd. 1, März 1992, S. 505–508.

[Cardin et al. 93] R. Cardin, Y. Normandin und E. Millien, *Inter-Word Coarticulation Modeling and MMIE Training for Improved Connected Digit Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (Minneapolis, Minnesota, USA), Bd. 2, April 1993, S. 243–246.

[Chou et al. 92] W. Chou, B.-H. Juang und C.-H. Lee, *Segmental GPD Training of HMM Based Speech Recognizer*, IEEE International Conference on Acoustics, Speech and Signal Processing (San Francisco, California, USA), Bd. 1, März 1992, S. 473–476.

[Chou et al. 93] W. Chou, C.-H. Lee und B.-H. Juang, *Minimum Error Rate Training based on N-Best String Models*, IEEE International Conference on Acoustics, Speech and Signal Processing (Minneapolis, Minnesota, USA), Bd. 2, April 1993, S. 652–655.

[Chow 90] Yen-Lu Chow, *Maximum Mutual Information Estimation of HMM Parameters for Continuous Speech Recognition using The N-Best Algorithm*, IEEE International Conference on Acoustics, Speech and Signal Processing (Albuquerque, New Mexico, USA), April 1990.

[Eisele et al. 96] T. Eisele, R. Haeb-Umbach und D. Langmann, *A Comparative Study of Linear Feature Transformation Techniques for Automatic Speech Recognition*, Fourth International Conference on Spoken Language Processing (Philadelphia, PA, USA), Bd. 1, Oktober 1996, S. 252–255.

[Fukunaga 90] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1990.

[Gopalakrishnan et al. 89] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas und D. Nahamoo, *A Generalization of the Baum Algorithm to Rational Objective Functions*, IEEE International Conference on Acoustics, Speech and Signal Processing (Glasgow, GB), 1989.

[Gopalakrishnan et al. 91] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas und D. Nahamoo, *An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems*, IEEE Transactions on Information Theory **37** (1991), S. 107–113.

[Haeb-Umbach und Ney 92] R. Haeb-Umbach und H. Ney, *Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (San Francisco, California, USA), Bd. 1, März 1992, S. 13–16.

[Kapadia et al. 93] S. Kapadia, V. Valtchev und S. J. Young, *MMI Training for Continuous Phoneme Recognition on the TIMIT Database*, IEEE International Conference on Acoustics, Speech and Signal Processing (Minneapolis, Minnesota, USA), Bd. 2, April 1993, S. 494–494.

[Ljolje et al.] A. Ljolje, Y. Ephraim und L. R. Rabiner, *Estimation of Hidden Markov Model Parameters by Minimizing Empirical Error Rate*, IEEE International Conference on Acoustics, Speech and Signal Processing.

[Merialdo 88] B. Merialdo, *Phonetic Recognition using Hidden Markov Models and Maximum Mutual Information Training*, IEEE International Conference on Acoustics, Speech and Signal Processing (New-York, USA), 1988.

[Ney 84] H. Ney, *The Use of a One-stage Dynamic Programming Algorithm for Connected Word Recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing **32** (1984), S. 263–271.

[Ney 90] Hermann Ney, *Acoustic Modeling of Phoneme Units for Continuous Speech Recognition*, Fifth Europ. Signal Processing Conference (Barcelona, Spanien), September 1990, S. 65–72.

[Ney 93] H. Ney, *Architecture and Search Strategies for Large Vocabulary Continous Speech Recognition*, NATO Advanced Study Institute (Bubion, Spain), Juni 1993, S. 59–84.

[Ney und Aubert 96] H. Ney und X. Aubert, *Dynamic Programming Search Strategies: From Digit Strings to Large Vocabulary Word Graphs*, Automatic Speech and Speaker Recognition (Norwell, Massachusetts, USA) (C.-H. Lee, F. K. Soong und K. K. Paliwal, eds.), Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1996, S. 385–411.

[Ney und Eiden 95] H. Ney und A. Eiden, *Musterekennung und neuronale Netze*, September 1995, Vorlesungsmitschrift.

[Ney und Nießen 95] H. Ney und S. Nießen, *Sprachmodellierung und Suche*, April 1995, Vorlesungsmitschrift.

[Normandin 91] Yves Normandin, *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem*, Ph.D. dissertation, McGill University, Montreal, Juni 1991.

[Normandin 95] Yves Normandin, *Optimal Splitting of HMM Gaussian Mixture Components with MMIE Training*, IEEE International Conference on Acoustics, Speech and Signal Processing (Detroit, Michigan, USA), Bd. 1, Mai 1995, S. 449–452.

[Normandin 96] Yves Normandin, *Maximum Mutual Information Estimation of Hidden Markov Models*, Automatic Speech and Speaker Recognition (Norwell, Massachusetts, USA) (C.-H. Lee, F. K. Soong und K. K. Paliwal, eds.), Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1996, S. 57–81.

[Normandin et al.] Y. Normandin, R. Lacouture und R. Cardin, *MMIE Training for Large Vocabulary Continuous Speech Recognition*, International Conference On Spoken Language Processing.

[Normandin und Morgera 91] Yves Normandin und S. D. Morgera, *An Improved MMIE Training Algorithm for Speaker-Independent, Small Vocabulary, Continuous Speech Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (Toronto, Canada), Bd. 1, Mai 1991, S. 537–540.

[Paliwal et al. 95] K. K. Paliwal, M. Bacchiani und Y. Sagisaka, *Minimum Classification Error Training Algorithm for Feature Extractor and Pattern Classifier in Speech Recognition*, ESCA 3rd European Conference On Speech Communication And Technology (Rhodes, Greece), Bd. 1, September 1995, S. 541–544.

[Reichl und Ruske 95] W. Reichl und G. Ruske, *Discriminative Training for Continuous Speech Recognition*, ESCA 3rd European Conference On Speech Communication And Technology (Madrid, Spanien), Bd. 1, September 1995, S. 537–540.

[Schlüter 96] R. Schlüter, *Diskriminatives Training: Detaillierte Rechnungen und Erläuterungen*, Interner Bericht, 1996, Lehrstuhl für Informatik VI, RWTH-Aachen.

[Schlüter et al. 97] R. Schlüter, W. Macherey, S. Kanthak, H. Ney und L. Welling, *Comparison of Optimization Methods for Discriminative Training Criteria*, ESCA 5th European Conference On Speech Communication And Technology (Rhodes, Greece), Bd. 1, 1997, S. 15–18.

[Schlüter et al. 98] R. Schlüter, W. Macherey, B. Müller und H. Ney, *Comparison of Discriminative Training Criteria and Optimization Methods for Small Vocabulary and Applications to Large Vocabulary Speech Recognition*, 1998, eingereicht bei Speech Communication.

[Schlüter und Macherey 98] R. Schlüter und W. Macherey, *Comparison of Discriminative Training Criteria*, IEEE International Conference on Acoustics, Speech and Signal Processing, Bd. 1, Mai 1998, S. 493–496.

[Schukat-Talamazzini] E. G. Schukat-Talamazzini, Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen.

[Valtchev 95] Valtcho Valtchev, *Discriminative Methods in HMM-based Speech Recognition*, Ph.D. promotion, St. John's College, University of Cambridge, GB, März 1995.

[Valtchev et al. 96] V. Valtchev, J. J. Odell, P. C. Woodland und S. J. Young, *Lattice-Based Discriminative Training For Large Vocabulary Speech Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (Atlanta, Georgia, USA), Bd. 2, Mai 1996, S. 605–608.

[Vintsyuk 71] T. Vintsyuk, *Element-Wise Recognition of Continous Speech Composed of Words from a Specified Dictionary*, Cybernetics **7** (1971), S. 361–372.

[Viterbi 67] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimal decoding algorithm*, IEEE Transactions on Information Theory **IT-13** (1967), S. 260–269.

[Welling et al. 95] L. Welling, H. Ney, A. Eiden und C. Forbrig, *Connected Digit Recognition using Statistical Template Matching*, ESCA 3rd European Conference On Speech Communication And Technology (Madrid, Spanien), Bd. 2, September 1995, S. 1483–1486.

[Wolfertstetter 96] F. Wolfertstetter, *Verallgemeinerte stochastische Modellierung für die automatische Spracherkennung*, Ph.D. promotion, Lehrstuhl für Mensch-Maschine-Kommunikation, Technische Universität München, Deutschland, 1996.