Rheinisch Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik VI
Prof. Dr.-Ing. Hermann Ney

Seminar “Speech Recognition and Language Processing” im SS 2003

Discriminative Training

Lars Haferkamp

Matrikelnummer 225 913

7. August 2003

Betreuer: Wolfgang Macherey

Zusammenfassung

In dieser Ausarbeitung wird das Verfahren des diskriminativen Trainings als Methode zur Reduzierung der Erkennungsfehler eines automatischen Spracherkennungssystems betrachtet. Im Training im Allgemeinen werden die Parameter des akustischen Modells aufgrund gegebener Trainingsdaten so gewählt, dass eine gegeben Zielfunktion ein (lokales) Maximum erreicht. Die diskriminativen Zielfunktion berücksichtigt, im Gegensatz zum konventionellen Training, nicht nur die Wahrscheinlichkeit, dass ein akustisches Signal der richtige Klasse zugeordnet wird, sondern auch die Wahrscheinlichkeit, dass ein akustisches Signal der falschen Klasse zugeordnet wird. Somit werden die Klassengrenzen optimiert und aus „falschen“ Klassengrenzen resultierende Erkennungsfehler vermieden.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einführung

In automatischen Spracherkennungssystemen werden heute meist *Hidden Markov Modelle* (HMMs) eingesetzt, um gesprochene Sprache zu modellieren. Innerhalb der HMMs werden Wahrscheinlichkeitsverteilungen benutzt. Die Parameter dieser Verteilungen müssen im Training, das der eigentlichen Spracherkennung vorausgeht, geschätzt werden. Eine übliche Methode, um die Verteilungsparameter in HMMs zu schätzen, ist der *Maximum Likelihood (ML)* Schätzer. Das Problem dieser Methode ist jedoch, dass ML nur klassenindividuell trainiert. D.h. um die Parameter einer Klasse (z.B. einem Wort) zu trainieren, werden nur die zugehörigen Trainingsäußerungen berücksichtigt. Dadurch kann es passieren, dass sich Klassen überlappen und Äußerungen der falschen Klasse zugeordnet werden. Dieses Überlappungsproblem wird im diskriminativen Training angegangen; dort wird die Möglichkeit, dass eine Äußerung einer anderen Klasse zugeordnet wird, berücksichtigt und die Parameter so geschätzt, dass eine größtmögliche Klassentrennung vorliegt.

Wir behandeln hier das Training in der Wortkettenerkennung. Im Unterschied zur Einzelworterkennung werden Sätze bestehend aus verschiedenen aufeinanderfolgenden Äußerungen trainiert. So können die Übergänge zwischen den einzelnen Worten mittrainiert werden.

Da für jede Äußerung im Prinzip alle möglichen konkurrierenden Klassen in Betracht genommen werden, erhöht sich der Rechenaufwand enorm. In Systemen, die mit einem kleinen Vokabular arbeiten ist das noch machbar. Doch bei größerem Vokabular müssen neue Ansätze herangezogen werden und die Anzahl der konkurrierenden Klassen muss eingeschränkt werden. Eine Lösung dieses Problems, die mit Wortgraphen arbeitet, wird in dieser Ausarbeitung vorgestellt.

Im statistischen Ansatz der Spracherkennung wird bei der Suche nach der richtigen Wortfolge die *Bayessche Entscheidungsregel* angewandt (siehe Abbildung 1). Sie besagt, dass die Fehlerrate in der Erkennung minimal wird, wenn das Wort W gewählt wird, das die höchste a-posteriori Wahrscheinlichkeit liefert. X sei die aus dem digitalisierten Sprachsignal gewonnene Folge akustischer Merkmalsvektoren.

$$\begin{aligned}\hat{W} &= \arg\max_W \; pr(W|X) \\ &= \arg\max_W \; [\; pr(X|W)\; pr(W) \;/\; pr(X)]\end{aligned}$$

$pr(X|W)$ und $pr(W)$ bezeichnen hier die echten Wahrscheinlickeitsverteilungen. Da diese aber unbekannt sind, werden sie modelliert durch:

$$\hat{W} \approx \arg\max_W \; [\; p_\lambda(X|W)\; p(W) \;/\; p_\lambda(X)] \tag{1}$$

$p_\lambda(X|W)$ ist die klassenbedingte Wahrscheinlichkeit oder auch Emissionswahrscheinlichkeit von X gegeben W. Sie wird durch das akustische Modell, das HMM, bestimmt. λ ist die Menge der HMM-Parameter. $p(W)$ bezeichnet die a-priori Wahrscheinlichkeit der Wortfolge W. Diese wird durch das Sprachmodell erfasst. Da hier nur das $\arg\max$ betrachtet wird kann $p_\lambda(X)$ im Prinzip wegfallen. Wie wir aber sehen werden spielt $p_\lambda(X)$ eine wichtige Rolle im diskriminativen Training.

Die Frage im Training ist nun: Wie soll λ geschätzt werden, um die Erkennung zu optimieren?

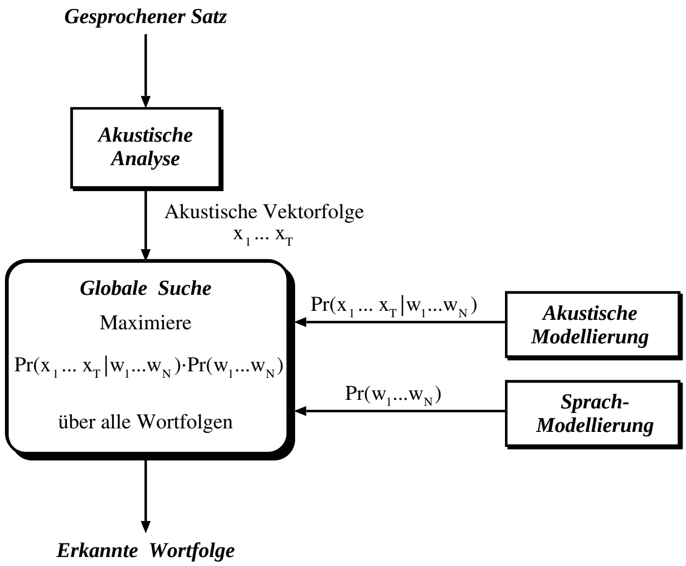

Abbildung 1: Architektur eines Spracherkennungssystems

1.1 Maximum-Likelihood-Training

Sei X_r die Folge akustischer Merkmalsvektoren $x_{r,1}, \cdots, x_{r,T_r}$ mit der zugehörigen Wortfolge W_r. Im Maximum Likelihood-Ansatz werden die Parameter so geschätzt, dass $p_\lambda(X_r|W_r)$ „im Mittel" maximal wird. D.h. für R Trainingsäußerungen maximiere:

$$p_\lambda(X|W) = \prod_{r=1}^{R} p_\lambda(X_r|W_r),$$

hierbei bleibt allerdings unberücksichtigt was $p_\lambda(X_r|W)$ für $W \neq W_r$ ist. Um eine fehlerfreie Erkennung durchzuführen, muss jedoch $p_\lambda(X_r|W_r) > p_\lambda(X_r|W)$, für alle $W \neq W_r$, sein. Dieses Problem wird im diskriminative Training behandelt. Es kann zum einen auf das in der Einführung erwähnte Überlappungsproblem zurückgeführt werden. Zum anderen kann es auch durch falsche oder schlechte Modelle verursacht werden. Z.B. benutzen wir in der akustischen Modellierung Gaussverteilungen, die echte Verteilung kann aber eine ganz andere sein.

1.2 Diskriminative Trainingskriterien

Ziel des diskriminativen Trainings ist es, so viele Trainingsäußerungen wie möglich richtig zu erkennen. Die diskriminativen Trainingskriterien berücksichtigen hierfür auch klassenfremde Trainingsdaten. Es gibt verschiedene diskriminative Kriterien, die in Tabelle 1 miteinander verglichen werden. In dieser Ausarbeitung soll aber verstärkt auf das häufig verwendete *Maximum Mutual Information* (MMI)-Kriterium eingegangen werden. Dieses Kriterium hat seinen Namen daher, dass hier die wechselseitige Information $I_\lambda(W, X)$ (auch Transinformation) zwischen der zu trainierenden Wortfolge und den zugehörigen

Merkmalsvektoren maximiert wird (vgl. [Nor96]):

$$I_\lambda(W,X) = \frac{p_\lambda(X|W)}{p_\lambda(X)\ pr(W)} = \prod_{r=1}^{R} \frac{p_\lambda(X_r,W_r)}{p_\lambda(X_r)\ p(W_r)}$$

Nehmen wir an, dass die Sprachmodellwahrscheinlichkeit $p(W_r)$ für jeden Satz konstant ist, dann erhalten wir als äquivalentes Kriterium die a-posteriori Wahrscheinlichkeit von W_r gegeben X_r:

$$p_\lambda(W_r|X_r) = \prod_{r=1}^{R} p_\lambda(W_r|X_r) = \prod_{r=1}^{R} \frac{p_\lambda(X_r,W_r)}{p_\lambda(X_r)}$$

Dies ist das *Conditional Maximum Likelihood* (CML)-Kriterium, das aber meist als MMI Kriterium bezeichnet wird. Im weiteren Verlauf werden wir nur noch den geläufigeren Begriff MMI-Kriterium verwenden. Mit dem Theorem über die totale Wahrscheinlichkeit

$$p_\lambda(X_r) = \sum_{W \in \mathcal{M}_r} p_\lambda(X_r|W)\ p(W) \tag{2}$$

folgt

$$p_\lambda(W_r|X_r) = \prod_{r=1}^{R} \frac{p_\lambda(X_r|W)\ p(W)}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda(X_r|W)\ p(W)}.$$

Bei korrekter Auswertung von Gleichung 2 müsste die Summe über alle möglichen Wortfolgen, d.h. unendlich viele, gebildet werden. Näherungsansätze für $\mathcal{M}_r$ werden in Abschnitt 2.2 dieser Ausarbeitung vorgestellt. Die MMI-Zielfunktion ergibt sich nun aus dem Logarithmus über die a-posteriori Wahrscheinlichkeit

$$\log p_\lambda(W_r|X_r) = \sum_{r=1}^{R} \log \frac{p_\lambda(X_r|W)\ p(W)}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda(X_r|W)\ p(W)}. \tag{3}$$

Allgemein wird die Zielfunktion für diskriminative Trainingskriterien wie folgt beschrieben:

$$F_D(\lambda) = \sum_{r=1}^{R} f\left(\log \frac{p_\lambda^\alpha(X_r|W)\ p(W)^\alpha}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda^\alpha(X_r|W)\ p(W)^\alpha}\right), \tag{4}$$

wobei f eine Glättungsfunktion ist, die den Beitrag der entsprechenden Trainingsäußerung wichtet. α wichtet die Verbundwahrscheinlichkeit $p_\lambda^\alpha(X_r,W) = p_\lambda^\alpha(X_r|W)\ p(W)^\alpha$, auch Satzwahrscheinlichkeit genannt. Außerdem unterscheiden sich die Kriterien noch durch die Wahl der konkurrierenden Wortfolgen $\mathcal{M}_r$. Tabelle 1 fasst die verschiedenen Kriterien zusammen (vgl. [Mac98]).

Tabelle 1: Diskriminative Kriterien

diskrim. Kriterium	Glättungs-funktion $f(z)$	Ableitung $f'(z)$	Wortfolgen in $\mathcal{M}_r$	Gewicht α
MMI	id	1	alle konk. Wortfolgen (einschließlich W_r)	1
CT			beste erkannte Wortfolge	(∞)
MCE	$\frac{1}{1+e^{2\beta z}}$	$-\frac{\beta}{2}\frac{1}{\cosh^2(\beta z)}$ $\beta = 4 \cdot 10^{-4}$	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

1.3 Optimierung der Zielfunktion

Um die Zielfunktion $F_D(\lambda)$ aus Gleichung 4 in Bezug zu den HMM-Parametern λ zu optimieren wird diese nach den einzelnen Parametern abgeleitet. Für eine genauere Beschreibung der Ableitung siehe [Mac98].

$$\frac{\partial F_D(\lambda)}{\partial \lambda} = \sum_{r=1}^{R} f'\left(\log \frac{p_\lambda^\alpha(X_r|W)\ p(W)^\alpha}{\sum\limits_{W \in \mathcal{M}_r} p_\lambda^\alpha(X_r|W)\ p(W)^\alpha}\right) \tag{5}$$

$$\cdot \left\{ \frac{\partial}{\partial \lambda} \log \left[p_\lambda^\alpha(X_r|W_r) p(W_r)^\alpha\right] \right. \tag{6}$$

$$\left. - \frac{\partial}{\partial \lambda} \log \left[\sum_{W \in \mathcal{M}_r} p_\lambda^\alpha(X_r|W)\ p(W)^\alpha \right] \right\} \tag{7}$$

Betrachten wie nun die Ableitung der einzelnen Terme:
Term 5 ist abhängig von der Wahl des diskriminativen Kriteriums und fällt für das MMI Kriterium weg, da die äußere Ableitung der *id* Funktion 1 ergibt.
Die Ableitung von Term 6 und 7 aus obiger Funktion reduziert sich auf die Ableitung der Emissionswahrscheinlichkeiten $p_\lambda^\alpha(X_r|W)$ nach einem zustandsabhängigen Parameter $\theta_s \in \lambda$. Der Teilterm $p(W)^\alpha$ fällt in der Ableitung weg, da hier kein λ vorkommt. Aufgrund der verwendeten akustischen Modellierung ergibt sich dann:

$$\frac{\partial \log p(x_1^T|W)}{\partial \theta_s} = \cdots$$

$$= \sum_{t=1}^{T} p(s_t = s | x_1^T, W) \tag{8}$$

$$\cdot \frac{\partial \log p(x_t|s)}{\partial \theta_s} \tag{9}$$

An dieser Stelle führen wir die *Forward-Backward (FB) Wahrscheinlichkeiten* ein. Der Summand in 8 wird als *spezielle* FB-Wahrscheinlichkeit bezeichnet:

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

Die spezielle FB-Wahrscheinlichkeit beschreibt die Wahrscheinlichkeit, dass der Zeitanpassungspfad für Wortfolge W bei gegebener Merkmalsvektorfolge X_r zum Zeitpunkt t durch den Zustand s führt [Mac98]. Summieren wir noch zusätzlich über alle konkurrierenden Wortfolgen, wie in Term 7, dann sprechen wir von der *generalisierten* FB-Wahrscheinlichkeit:

$$\begin{aligned}\gamma_{r,t}(s) &:= \sum_{W\in\mathcal{M}_r} \frac{p_\lambda^\alpha(X_r|W)\ p(W)^\alpha}{\sum\limits_{V\in\mathcal{M}_r} p_\lambda^\alpha(X_r|V)\ p(V)^\alpha}\ \gamma_{r,t}(s|W) \\ &= p_\lambda(s_t = s|X_r) \quad \text{falls } \alpha = 1\end{aligned}$$

Der Teilterm

$$\begin{aligned}&\frac{p_\lambda^\alpha(X_r|W)\ p(W)^\alpha}{\sum\limits_{V\in\mathcal{M}_r} p_\lambda^\alpha(X_r|V)\ p(V)^\alpha} \\ =\ & p_\lambda(X_r, W)/\,p_\lambda(X_r) \quad \text{falls } \alpha = 1\end{aligned}$$

bestimmt dabei das Gewicht der jeweiligen speziellen FB-Wahrscheinlichkeit. Die generalisierte FB-Wahrscheinlichkeit beschreibt also die (gewichtete) Wahrscheinlichkeit, dass Zustand s zum Zeitpunkt t unabhängig von einer Wortfolge hypothetisiert wird. Die Differenz aus spezieller und generalisierter FB-Wahrscheinlichkeit bildet somit das Gewicht der Ableitung des Logarithmus der Emissionswahrscheinlichkeit (Term 9). Die effiziente Berechnung der FB-Wahrscheinlichkeiten ist Inhalt von Kapitel 2.

Aus Term 9 wird, da wir hier Mischverteilungen verwenden, nach Maximum Approximation

$$\delta(l, l(x_{r,t}, s)) \cdot \frac{\partial}{\partial\theta_{s,l}} \log p(x_{r,t}|\theta_{s,l})$$

mit

$$l(x, s) := \arg\max_l \{c_{s,l} \cdot p(x|\theta_{s,l})\},$$

s, l bezeichnet hier Mischverteilung l von Zustand s und $c_{s,l}$ die Wichtung der Verteilung.

Insgesamt ergibt sich dann für die Ableitung der MMI-Zielfunktion:

$$\frac{\partial F_{MMI}(\lambda)}{\partial\theta_{s,l}} = \sum_{r=1}^{R}\sum_{t=1}^{T_r}[\gamma_{r,t}(s|W) - \gamma_{r,t}(s)]\ \delta(l, l(x_{r,t}, s))\ \frac{\partial}{\partial\theta_{s,l}} \log p(x_{r,t}|\theta_{s,l}) \qquad (10)$$

Durch Nullsetzen der Ableitung und Auflösen nach einem Parameter kann dann die optimale Zielfunktion bestimmt werden, was aber i.A. analytisch zu keiner geschlossenen Lösung führt. Mit welchen Verfahren die optimale Zielfunktion angenähert wird ist Bestandteil von Kapitel 3.

1.4 Der Trainingsalgorithmus

Der verallgemeinerte diskriminative Trainingsalgorithmus ist hier dargestellt.

1. Initialisierung:
   - Bestimme die Startwerte λ_0 der Parameter mittels linearer Segmentierung.
   - Initialisiere Anzahl $L(s, w)$ der Einzelverteilungen einer Mischverteilung.
2. Führe eine gewisse Anzahl von Iterationen ML-Training durch.
3. Generiere die konkurrierenden Wortfolgen, unter der Annahme, dass sich diese in der Folge kaum noch ändern (siehe [VOWY97]).
4. Wiederhole bis Konvergenz der diskriminativen Zielfunktion erreicht ist:
   (a) Für jede Trainingsäußerung $r = 1, \cdots, R$ aus den Trainingsdaten
      i. Führe Forced Time Alignment für die richtige Wortfolge durch
      ($x_{r,t} \rightarrow$ Mischverteilung, nächste Einzelverteilung).
      Akkumuliere die sogenannten Zählerstatistiken (da aus dem Zähler der diskriminative Zielfunktion stammend):
      $\sum_{t=1}^{T_r} \gamma_{r,t}(s|W),\ \sum_{t=1}^{T_r} \gamma_{r,t}(s|W)x_{r,t},\ \sum_{t=1}^{T_r} \gamma_{r,t}(s|W)x_{r,t}^2.$
      ii. Führe Forced Time Alignment für alle konkurrierenden Wortfolgen durch.
      Bewerte konkurrierende Wortfolgen neu.
      Akkumuliere Nennerstatistiken (da aus dem Nenner stammend):
      $\sum_{t=1}^{T_r} \gamma_{r,t}(s),\ \sum_{t=1}^{T_r} \gamma_{r,t}(s)x_{r,t},\ \sum_{t=1}^{T_r} \gamma_{r,t}(s)x_{r,t}^2$
   (b) Schätze die Parameter neu
   Mittelwerte, Varianzen, Mischverteilungsgewichte
   (EB-Algorithmus oder Gradientenabstiegsverfahren)
   (c) Splitten der Dichten (optional). Hierdurch kann die Anzahl der Zentren der Mischverteilung erhöht werden.

Die Trainingsdaten sollten die gesamte Variationsbreite des akustischen Signals abdecken [Ney02]. D.h. durch

- verschiedene Sprecher
- „natürliche“ Sprache (leichte Dialekte, unterschiedliche Sprechgeschwindigkeiten)
- Trainingsdaten über größeren Zeitraum verteilen
- verschiedene Übertragungskanäle (Mikrofon, Telefon,...)

Zum Test der Fehlerrate nie die Trainingsdaten verwenden $\rightarrow$ eigene Testdaten.

2 Berechnung der FB-Wahrscheinlichkeiten

Die Berechnung der FB-Wahrscheinlichkeiten bildet einen wichtigen Punkt im diskriminativen Training, wie aus dem Trainingsalgorithmus hervorgeht. Besonders die effektive Berechnung der generalisierten FB-Wahrscheinlichkeit für eine grosse Anzahl von konkurrierenden Wortfolgen ist ein wichtiges Thema. Im Gegensatz zur generalisierten FB-Wahrscheinlichkeit wird die spezielle FB-Wahrscheinlichkeit, auf die wir als nächstes eingehen wollen, auch im ML-Training gebraucht. Die Definitionen der speziellen und der generalisierten FB-Wahrscheinlichkeit wurden schon in Kapitel 1.3 gegeben.

2.1 Berechnung der speziellen FB-Wahrscheinlichkeit

Nach [Nor96] und [Mac98] gibt die spezielle FB-Wahrscheinlichkeit $\gamma_{r,t}(s|W)$ die a-posteriori Wahrscheinlichkeit einzelner Zustände eines Wortes zu einem bestimmten Zeitpunkt an. Wir betrachten hier die exakte Berechnung (Baum-Welch-Kriterium) und die Maximum-Approximation (Viterbi-Näherung).

2.1.1 Das Baum-Welch-Kriterium

Die exakte Berechnung der speziellen FB-Wahrscheinlichkeit sieht mittels der Baum-Welch-Rekursion wie folgt aus:

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

$a_{r,t-1}(\sigma, W)$ ist die sogenannte *Forward*-Wahrscheinlichkeit und ist als die Wahrscheinlichkeit definiert, mit welcher der Pfad der in σ endet $x_{r,1}, \cdots, x_{r,t-1}$ produziert hat. $b_{r,t}(\sigma, W)$ ist die sogenannte *Backward*-Wahrscheinlichkeit, die besagt mit welcher Wahrscheinlichkeit ein Pfad der in s beginnt $x_{r,t}, \cdots, x_{r,T_r}$ produziert hat. Die Vorgängerzustände σ werden bei Verwendung des 0-1-2 Standardmodells durch die Zustände eingeschränkt, die durch einen *Loop-*, *Forward-* oder *Skip*-Übergang in Zustand s führen. Zur effizienten Berechnung der Forward- und der Backward-Wahrscheinlichkeit werden nach [Mac98] folgende Rekursionsgleichungen herangezogen:

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

und

$$b_{r,t}(s, w) = \sum_{\sigma} b_{r,t+1}(\sigma, w)\ p_\lambda(x_{r,t+1}|\sigma)\ p(\sigma|s)$$

Das Prinzip zur Berechnung der speziellen FB-Wahrscheinlichkeit kann Abbildung 2 entnommen werden.

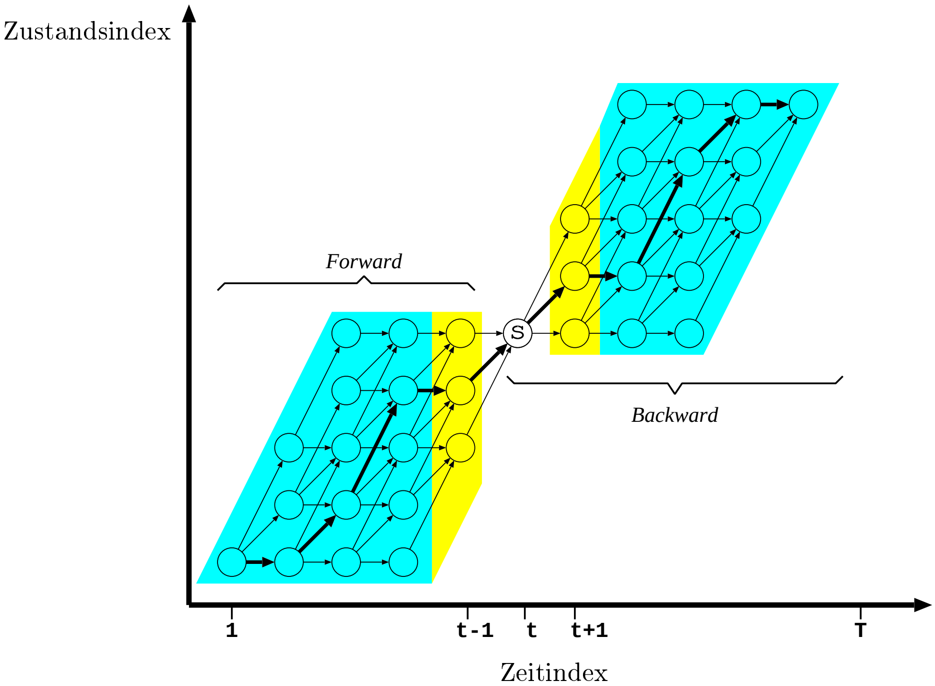

Abbildung 2: Prinzip zur Berechnung der speziellen FB-Wahrscheinlichkeit

2.1.2 Maximum-Approximation

In der Praxis wird meist eine Maximum-Aproximation verwendet, die sogenannte Viterbi Näherung. Hier wird zur Berechnung der speziellen FB-Wahrscheinlichkeit, nur der Zeitanpassungspfad mit der höchsten Satzwahrscheinlichkeit betrachtet. D.h. für Zustände, die auf diesem Pfad liegen ist die FB-Wahrscheinlichkeit 1 und für alle anderen Zustände ist die FB-Wahrscheinlichkeit 0 (siehe auch [Mac98]):

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

$s_t(X_r, W)$ bezeichnet den Zustand, der zum Zeitpunkt t auf dem besten Zeitanpassungspfad liegt. Da die Satzwahrscheinlichkeit nur für einen Pfad berechnet werden muss, genügt die Berechnung der Forward-Wahrscheinlichkeit zur Bestimmung des besten Zeitanpassungspfades.

2.2 Berechnung der generalisierten FB-Wahrscheinlichkeit

Die generalisierte FB-Wahrscheinlichkeit $\gamma_{r,t}(s)$ gibt nach [Nor96] an, mit welcher Wahrscheinlichkeit der Merkmalsvektor $x_{r,t}$ durch den Zustand s, unabhängig von einer Wortfolge, generiert wurde. Wie schon im Abschnitt 1.2 über die diskriminativen Trainingskriterien gezeigt, können wir bei der Berechnung der generalisierten FB-Wahrscheinlichkeit nicht *alle* (da unendlich viele) konkurrierenden Wortfolgen betrachten. D.h. wir müssen die Menge $\mathcal{M}_r$ der konkurrierenden Wortfolgen einschränken.

2.2.1 Verwendung von N-Besten-Listen

Eine Idee, um dieses Problem zu lösen, ist die Verwendung von N-Besten Listen. N-Besten Listen enthalten die N besten Wortfolgen, die zu einer Äußerung die größten Satzwahrscheinlichkeiten produzieren. Diese konkurrierenden Wortfolgen werden in einem vorangehenden Erkennungsschritt gewonnen. Für jede Wortfolge muss dann die spezielle FB-Wahrscheinlichkeit mit dem Gewicht $p_\lambda(X_r, W)/p_\lambda(X_r)$ berechnet werden. Ein Problem hierbei ist, dass sich manche Hypothesen in den N-Besten Listen nur minimal unterscheiden (durch ein Wort oder eine Silbe). Trotzdem wird das Gewicht $p_\lambda(X_r, W)/p_\lambda(X_r)$ übereinstimmender Zustandsfolgen immer wieder neu berechnet. Außerdem sind aus diesem Grund große Listen notwendig, um die Trainingsdaten gut auszuschöpfen. D.h.die Verwendung von N-Besten-Listen resultiert in einem hohen Rechenzeitaufkommen.

2.2.2 Verwendung von Wortgraphen

Eine bessere Lösung des oben genannten Problems bietet die Verwendung von *Wortgraphen* (Word Lattice, siehe auch Abbildung 3). Ein Wortgraph ist ein gerichteter Graph in dem die Kanten durch Wörter beschriftet sind und die Knoten den Start- und Endzeitpunkten der Wörter entsprechen. Ein Wortgraph hat einen eindeutigen Anfangs- und einen Endknoten. Jeder Pfad vom Anfangs- bis zum Endknoten stellt einen Satz dar. Hier werden nun die konkurrierenden Wortfolgen auf die in einem Wortgraphen enthaltenen Sätze eingeschränkt. Der Wortgraph muss in einem vorangehenden Erkennungsschritt gewonnen werden.

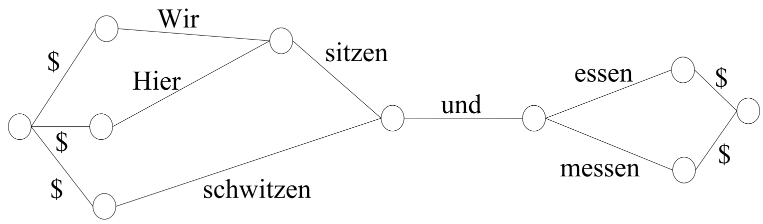

Abbildung 3: Beispiel eines Wortgraphen

Dann lässt sich die generalisierte FB-Wahrscheinlichkeit folgendermaßen berechnen:

$$\begin{aligned}\gamma_{r,t}(s) &= \sum_{W \in \mathcal{M}_r} \frac{p_\lambda(X_r, W)}{p_\lambda(X_r)} \gamma_{r,t}(s|W) \\ &= \sum_{\{w | t_a(w) \le t \le t_e(w)\}} \frac{a_{r,t}(s, W_a\ w\ W_e) \cdot b_{r,t}(s, W_a\ w\ W_e)}{p_\lambda(X_r)}\end{aligned}$$

Zur Berechnung von $\gamma_{r,t}(s)$ werden also nur die Wörter w betrachtet, für die der Zeitpunkt t zwischen Anfangs- $t_a(w)$ und Endzeit $t_e(w)$ von Wort w liegt. W_a bezeichnet die Menge der Wortfolgen, die in $t_a(w)$ enden und W_e die Menge der Wortfolgen, die in $t_e(w)$ beginnen. Des weiteren müssen die Rekursionsformeln für Forward- und Backward-Wahrscheinlichkeit erweitert werden. Für die Forward-Wahrscheinlichkeit müssen zusätzlich alle Endzustände σ_e von Wörtern v berücksichtigt werden, die den Zustand s in Wort

w erreichen können:

$$a_{r,t}(s, W_a\, w\, W_e) = p_\lambda(x_{r,t}|s, w) \cdot \left(\sum_\sigma a_{r,t-1}(\sigma, w)\; p(s|\sigma) + \underbrace{\sum_v \sum_{\sigma_e} a_{r,t-1}(\sigma_e, v)\; p(s|\sigma)\; p(w|v)}_{\text{zusätzlicher Term}} \right)$$

Entsprechendes gilt für die Backward-Wahrscheinlichkeit, hier werden zusätzlich die Anfangszustände σ_a der von Zustand s in Wort w aus erreichbaren Wörter (gemäß Wortgraph und akustischem Modell) in Betracht gezogen:

$$b_{r,t}(s, W_a\, w\, W_e) = \sum_\sigma b_{r,t+1}(\sigma, w)\; p_\lambda(x_{r,t+1}|\sigma)\; p(\sigma|s) + \underbrace{\sum_u \sum_{\sigma_a} b_{r,t+1}(\sigma_a, u)\; p_\lambda(x_{r,t+1}|\sigma_a)\; p(u|w)}_{\text{zusätzlicher Term}}$$

Hier wurde jetzt gezeigt wie die generalisierte FB-Wahrscheinlichkeit nach dem Baum-Welch Training aus Abschnitt 2.1.1 bestimmt wird. Betrachten wir nun die Viterbi-Näherung aus Abschnitt 2.1.2, d.h. wir betrachten nur den besten Zeitanpassungspfad. Somit können wir Anfangs- und Endzeit für jedes Wort genau bestimmen. Sei $p_{[t_a,t_e]}(w|X_r)$ die Wahrscheinlichkeit, dass Wort w in den Zeitraum $[t_a(w), t_e(w)]$ fällt, auch *Wortwahrscheinlichkeit* genannt, dann ergibt sich für die generalisierte FB-Wahrscheinlichkeiten:

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

$p_\lambda(s_t = s|w, x_{r\,t_a}^{\;t_e})$ entspricht der speziellen FB-Wahrscheinlichkeit eingeschränkt auf Wörter, die im Zeitintervall $[t_a, t_e]$ beginnen oder enden. Die Berechnung der Wortwahrscheinlichkeit $p_{[t_a,t_e]}$ kann nun durch einen FB-Algorithmus erfolgen, wobei wieder auf eine rekursive (und effiziente) Berechnung der Forward- und der Backwardwahrscheinlichkeit zurückgegriffen wird. Bei Verwendung eines m-Gramm Sprachmodells folgt:

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

Dabei ist die Forward Wahrscheinlichkeit $\Phi_{r,t_e}(h_1, \cdots, h_{m-2}, w)$ als die Wahrscheinlichkeit definiert, dass die letzten $m-1$ Kanten (Wörter) eines im Endknoten von Wort $w_{t_a}^{t_e}$ endenden Pfades der Wortfolge $h_1, \cdots, h_{m-2}, w$ entsprechen.

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

Die Backward Wahrscheinlichkeit $\tilde{\Psi}_{r,t_a}(w, f_1, \cdots, f_{m-2})$ ist entsprechend als die Wahrscheinlichkeit definiert, dass die folgenden $m-1$ Kanten eines im Startknoten von Wort $w_{t_a}^{t_e}$ beginnenden Pfades der Wortfolge $w, f_1, \cdots, f_{m-2}$ entsprechen.

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

Aus der Forderung, dass die Backward Wahrscheinlichkeit Ψ in $t = 1$ gleich der Forward Wahrscheinlichkeit in $t = T_r$ sein muss folgt, dass bei der Berechnung der Backward Wahrscheinlichkeit die gesamte Historie im Sprachmodell berücksichtigt werden muss. Daher wurde Ψ zu $\tilde{\Psi}$ modifiziert und der fehlende Sprachmodellfaktor erst durch den Term

$$\prod_{i=1}^{m-2} p^{\alpha}(f_i | h_{i+1}^{m-1}, w, f_1^{i-1})$$

in der Berechnung von $p_{[t_a,t_e]}(w|X_r)$ berücksichtigt. Die Bestimmung von $\bar{p}_{\lambda}(X_r)$ geschieht durch die Summierung der Forward-Wahrscheinlichkeiten aller aus dem Wortgraphen herausführenden Kanten bzw. durch die Summierung der Backward-Wahrscheinlickeiten aller Startwörter. Für eine ausführliche Beschreibung der Berechnungen siehe [Mac98].

3 Neuschätzung der Parameter

Um die Zielfunktion in Bezug zu ihren Verteilungsparametern zu optimieren, muss die Zielfunktion nach einem Parameter abgeleitet, gleich Null gesetzt und nach dem Parameter aufgelöst werden. Da dies aber für solche Funktionen, wie die diskriminative Zielfunktion i.A. analytisch nicht möglich ist, wird die Zielfunktion iterativ optimiert. D.h. die Paramter werden schrittweise neugeschätzt bis ein (lokales) Maximum erreicht wird. Für die Neuschätzung der Verteilungsparameter existieren bislang zwei Methoden. Der in der Praxis verwendete *erweiterte Baum (EB)* Algorithmus und das *Gradientenabstiegsverfahren*. Das Gradientenabstiegsverfahren ist eine einigermaßen sichere Parameterabschätzungstechnik. Mit einer infinitesimalen Schrittweite, die nie realisiert werden kann, konvergiert es gegen ein (lokales) Maximum. Das Problem ist, dass eine zu kleine Schrittweite in eine zu langsame Konvergenz resultiert. D.h. die Schwierigkeit liegt nicht darin, konvergierende Verfahren zu finden, sondern auch effektive.

3.1 Der EB-Algorithmus

In [GKNN89] wird eine Neuschätzungsformel verwendet, die für eine hinreichend kleine Schrittweite gegen ein lokales Maximum von $F_{MMI}(\lambda)$ konvergiert. Allerdings wird dies nur für diskreter Verteilungen gezeigt. In [Nor96] wird eine diskrete Approximation an die Gauss'schen Verteilungen verwendet und folgende Neuschätzungsformeln für Mittelwert μ und Kovarianzmatrix σ^2 aufgestellt:

$$\hat{\mu}_{s,l} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} [\gamma_{r,t}(s|W) - \gamma_{r,t}(s)] x_{r,t} + D\mu_{s,l}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} [\gamma_{r,t}(s|W) - \gamma_{r,t}(s)] + D}$$

$$\hat{\sigma}_{s,l}^2 = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} [\gamma_{r,t}(s|W) - \gamma_{r,t}(s)] x_{r,t}^2 + D(\sigma_{s,l}^2 + \mu_{s,l}^2)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} [\gamma_{r,t}(s|W) - \gamma_{r,t}(s)] + D} - \hat{\mu}_{s,l}^2$$

Auf die Optimierung der Mischverteilungsgewichte wird in [Mac98] näher eingegangen. Dort wird folgende Neuschätzungsformel aufgestellt:

$$\hat{c}_{s,l} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} [\gamma_{r,t}(s|W) - \gamma_{r,t}(s)] + Dc_{s,l}}{\sum_{\bar{l}=1}^{L(s)} \sum_{r=1}^{R} \sum_{t=1}^{T_r} [\gamma_{r,t}(s|W) - \gamma_{r,t}(s)] + D}$$

$L(s)$ ist hier die Anzahl der Einzelverteilungen der Mischverteilung von Zustand s.

Auch wenn kein Beweis dafür existiert, dass die Neuschätzungsformeln konvergieren, so wurde aber bei der Wahl von D wie in [Nor96] in der Praxis erfolgreich mit ihnen gearbeitet.

3.2 Das Gradientenabstiegsverfahren

Um die Zielfunktion in Bezug auf einen Parameter $\theta \in \lambda$ zu optimieren wird diese im *Gradientenabstiegsverfahren (GD)* nach θ abgeleitet. Durch Einsetzen der aktuellen Parameterwerte erhalten wir dann die Richtung und Stärke der Steigung der Zielfunktion in Bezug auf θ. Der neue Parameterwert ergibt sich dann aus dem alten, der abhängig von Stärke der Steigung und Wahl der Schrittweite ϵ in Richtung der Steigung verschoben wird. In [Nor96] wird das Gradientenabstiegsverfahren (GD) auf diskreten Verteilungen angewandt. Wir wollen hier aber kontinuierliche Verteilungen betrachten. Die Neuschätzungsformeln sind dann folgende (siehe auch [Mac98]):

$$\text{Für die Mittelwerte ergibt sich:} \quad \hat{\mu}_{s,l} = \mu_{s,l} + \epsilon_{\mu_{s,l}} \frac{\partial F(\lambda)}{\partial \mu_{s,l}}$$

$$\text{Für die Kovarianzmatrizen:} \quad \hat{\sigma}^2_{s,l} = \sigma^2_{s,l} + \epsilon_{\sigma^2_{s,l}} \frac{\partial F(\lambda)}{\partial \sigma^2_{s,l}}$$

$$\text{Für die Mischverteilungsgewichte:} \quad \hat{c}_{s,l} = \left(c_{s,l} + \epsilon_{c_{s,l}} \frac{\partial F(\lambda)}{\partial c_{s,l}} \right) / \left(1 + \sum_l \epsilon_{c_{s,l}} \frac{\partial F(\lambda)}{\partial c_{s,l}} \right)$$

Bei der richtigen Wahl der Schrittweiten wird in [SMK+97] gezeigt, dass Gradientenabstiegsverfahren und Baum-Welch Algorithmus äquivalent sind. Leichte Unterschiede ergeben sich nur in den Iterationsgleichungen der Kovarianzmatrizen.

3.3 Bestimmung der Schrittweite/Iterationskonstanten

Ein Hauptproblem bei der Benutzung der Neuschätzungsformeln ist die richtige Wahl der Iterationskonstanten. Wird D sehr groß gewählt, d.h. ist die Schrittweite sehr klein, so konvergiert die Formel zwar relativ sicher, aber sehr langsam. Ist die Schrittweite hingegen zu groß, dann kann es sein, dass sich die Zielfunktion nicht mit jedem Schritt verbessert und oszilliert. Eine sinnvolle untere Grenze für D ist ein Wert der sicherstellt, dass alle Varianzen positiv bleiben [VOWY97]. In [VOWY97] wurde D auf den doppelten Wert gesetzt. Ausserdem wurden dort Experimente durchgeführt, um den Nutzen einer globalen Iterationskonstante mit dem von phonemspezifischen Konstanten (47 verschiedene insgesamt) zu vergleichen. Dabei verbesserte die Version mit den phonemspezifischen Konstanten die Konvergenzgeschwindigkeit um den Faktor 2 und mehr.

4 Experimentelle Ergebnisse

Wir wollen hier nun die Ergebnisse verschiedener Forschungsgruppen zusammenstellen. Verwendet wird immer das MMI-Trainingskriterium, welches dann mit dem ML-Training verglichen wird. Wir betrachten hier der Übersichtlichkeit halber nur die Wortfehlerrate.

In [Nor96] werden Experimente auf dem TI digit strings Korpus (sprecherunabhängige Ziffernketten Datenbank) mit 8.5 Stunden gesprochener Sprache durchgeführt. Hier wird der Corrective Training Algorithmus nach 3 Iterationen ML-Training angewandt. Bei Verwendung von diskreten Verteilungen ergibt sich eine Reduzierung der Wortfehlerrate von 1.36% auf 0.75% gegenüber dem reinen ML-Training. Bei Verwendung von kontinuierlichen Verteilungen reduziert sich die Fehlerrate von 1.22% auf 0.66%, also um 46%. Allerdings fällt die Reduzierung der Fehlerrate bei Verwendung von geschlechtsspezifischen Modellen nicht so hoch aus (32%).
In [Ney02] werden Ergebnisse auf dem gleichen Korpus vorgestellt. Hier reduziert sich die Wortfehlerrate bei Verwendung von Gaußsschen Einzelverteilungen von 0.72% im ML-Training auf 0.47% im MMI Training.

In [VOWY97] werden Experimente auf verschiedenen Testdaten aus der Wall Street Journal Datenbank und mit einer Lexikongröße von 65k Wörtern durchgeführt. Trainiert wird ein System mit Gaußschen Mischverteilungen, Cross-Word Kontext, Triphon Modell und einem Bigram Language Model (LM) auf 66 Stunden aus der Wall Street Journal Datenbank. Das System wird mit den Trainingsdaten (SI-284) selber und einem Bigram-, Trigram- und Fourgram LM getestet. Auf den Trainingsdaten reduziert sich die Wortfehlerrate von 8.1% auf 3.5%. Auf den Testdaten lässt sich feststellen, dass sich die Fehlerrate im Bigram-Fall um rund 2% reduziert, mit einem Trigram-LM ergibt sich jedoch ein leicht schlechteres Ergebniss als im ML-Fall. Außerdem wird die Verwendung einer unterschiedlichen Anzahl von Mischverteilungskomponenten pro Zustand getestet. Hier ergibt sich bei der Verwendung von einer Komponente eine Verbesserung der Fehlerrate von rund 2.5%, bei der Verwendung von 12 Komponenten ergibt sich nur noch eine Verbesserung von rund 0.5%. Bei einem Vergleich der Fehlerrate auf den Trainingsdaten und den Testdaten sieht man hier das die Fehlerrate auf den Testdaten mit jeder Iteration geringer wird, während sie auf den Testdaten nach der 6ten Iteration wieder ansteigt. Dies lässt auf Übertraining schließen. Ähnliches wurde auch in [Mac98] festgestellt und in Abbildung 4 verdeutlicht.

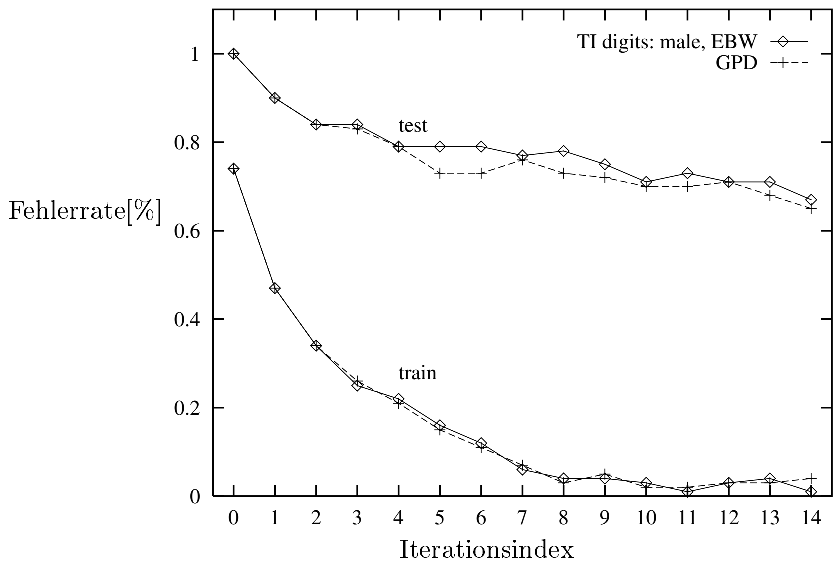

Abbildung 4: Ein Vergleich zwischen der Fehlerrate auf den Trainingsdaten und der Fehlerrate auf den Testdaten

In [WP02] wird ein System mit kontinuierlichen Verteilungen, Cross-Word Kontext, Triphon oder Quinphon Modellen und einem Trigram-LM auf 18, 68 und 265 Stunden Trainingsdaten aus dem Hub5 Korpus (Daten aus Telefongesprächen) trainiert. Die Testdaten sind eine Teilmenge aus dem Hub5 Korpus. Hier wird u.a. der Nutzen einer unterschiedlich großen Menge von Trainingsdaten untersucht. In den Experimenten mit 18 Stunden Training werden die besten Ergebnisse nach zwei Iterationen festgestellt. Hier ist der Gewinn in der Wortfehlerrate 1.7% gegenüber dem ML-Training bei Verwendung eines Bigram-LMs und 1.9% bei Verwendung eines Unigram-LMs. Dabei ist das System mit 6 Mischverteilungskomponenten leicht besser als das mit 12 Komponenten. Bei Verwendung des Systems, das mit 68 Stunden trainiert wurde, und Triphon-Modellen lassen sich die besten Ergebnisse auch nach 2 Iterationen feststellen. Für 6 Mischverteilungskomponenten ergibt sich eine noch grössere Reduzierung der Fehlerrate. Im Vergleich zum ML-Training reduziert sich die Fehlerrate von 27.4% auf 21.2%. Bei Nutzung von Quinphon-Modellen ergibt sich nur noch eine absolute Reduktion der Fehlerrate von 2.9%. Das beste Ergebniss wird hier nach 3 Iterationen erreicht. ...

5 Abschließende Betrachtung

In dieser Ausarbeitung haben wir nun gesehen wie diskriminatives Training benutzt werden kann, um die Erkennung gesprochener Sprache zu verbessern. Angefangen von der Idee das Problem des Klassenüberlapps im ML-Training zu lösen, über den Trainingsalgorithmus bis hin zu den mathematischen Formeln. Ein wichtiger Punkt im diskriminativen Training ist die effiziente Berechnung der generalisierten FB-Wahrscheinlichkeit. Hier bietet die Verwendung von Wortgraphen die beste Lösung. Bei Experimenten ließ sich eine Verbesserung der Erkennung auf ungesehenen Testdaten von 5%-10% ([WP02]) bzw. 5%-16% ([VOWY97]) feststellen. An verschiedenen Stellen ([Nor96] und [VOWY97]) wurde angemerkt das diskriminatives Training wenige Parameter effektiver nutzt als ML-Training. Bei einer größeren Anzahl von Parametern schwindet der Vorteil des diskriminativen Trainings. Dies kommt daher, dass im diskriminativen Training die Erkennung auf den Trainingsdaten optimiert wird und so die Gefahr des Übertrainings (siehe auch [VOWY97]) stärker gegeben ist als um ML-Training. Dieses Problem wird meist als Generalisierungsproblem bezeichnet. Es ist daher nicht unbedingt vorteilhaft das globale Optimum der Zielfunktion in Bezug auf einen Parameter zu finden (was im Baum-Welch Training für kontinuierliche Dichten auch nicht garantiert ist). Die Wahl einer geeigneten Glättungsfunktion kann außerdem die Gefahr des Übertrainings verringern?! In [WP02] wird jedoch gezeigt, dass diskriminatives Training bei geeigneten Generalisierungsmaßnahmen, wie Interpolation von ML-und MMI Zielfunktion, vielen Trainingsdaten und geeigneter Skalierung des akustischen Modells, auch in komplexen Erkennungsaufgaben auf großem Vokabular besser als ML-Training ist. In [WP02] wird aufgrund von Experimenten vermutet, dass der Vorteil des diskriminativen Trainings verschwindet, wenn in der Erkennung ein komplexeres Sprachmodell als im Training benutzt wird. Nicht näher eingegangen sind wir hier auf das diskriminative Splitten der Dichten, was aber in [Nor96], [VOWY97] und [WP02] nachgelesen werden kann.

Diskriminatives Training ist nur eine Methode um die Spracherkennung zu verbessern, doch insgesamt bietet es uns ein Verfahren durch die sich die Erkennung gegenüber ML trainierten Systemen sichtlich verbessert.

Literatur

[GKNN89] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo. A Generalization of the Baum Algorithm to Rational Objective Functions. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, page paper S12.9, Glasgow, GB, 1989.

[Mac98] W. Macherey. Implementierung und Vergleich diskriminativer Verfahren für Spracherkennung bei kleinem Vokabular. Master's thesis, RWTH Aachen, 1998.

[Ney02] H. Ney. Spracherkennung. Ausarbeitung zur Vorlesung an der RWTH Aachen. Ausgearbeitet von: Margit Generet und Sonja Nießen, 2002.

[Nor96] Y. Normandin. Maximum Mutual Information Estimation of Hidden Markov Models. In K. K. Paliwal C.H. Lee, F. K. Soong, editor, *Automatic Speech and Speaker Recognition*, pages 57–81. Kluwer Academic Publishers, Norwell, MA, 1996.

[SMK+97] R. Schlüter, W. Macherey, S. Kanthak, H. Ney, and L.Welling. Comparison of Optimization Methods for Discriminative Training Criteria. In *ESCA 5th European Conference On Speech Communication And Technology*, volume 1, pages 15–18, Rhodes, Greece, 1997.

[VOWY97] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young. MMIE Training of Large Vocabulary Recognition Systems. In *Speech Communication*, volume 22, No. 4, pages 303–314, September 1997.

[WP02] P. C. Woodland and D. Povey. Large Scale Discrimnative Training of Hidden Markov Models for Speech Recognition. In *Computer Speech and Language*, volume 16, No. 1, pages 25–47, Januar 2002.