Lehrstuhl für Datenverarbeitung

Untersuchungen zur Sprecheradaption in Systemen zur automatischen Spracherkennung mit Hilfe stochastischer Modellierung

Florian Schiel

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

Vorsitzender:	Univ.-Prof. Dr.-Ing. K. Antreich
Prüfer der Dissertation:	1. Priv.-Doz. Dr.-Ing. G. Ruske
	2. Univ.-Prof. Dr.rer.nat. M. Lang

Die Dissertation wurde am 26.01.1993 bei der Technischen Universität eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 05.05.1993 angenommen.

Zusammenfassung

In dieser Arbeit werden verschiedene Verfahren der Sprecheradaption in Systemen der automatischen Spracherkennung untersucht. Ziel dieser Verfahren ist es, die Erkennungsleistung während der Benutzung durch einen unbekannten Sprecher automatisch zu steigern, indem die Systemparameter schrittweise an dessen typisches Sprechverhalten angepaßt werden (Adaption). Da es sich bei den hier untersuchten Verfahren um statistische Erkennungsalgorithmen handelt, geschieht die Adaption durch stochastische Modellierung der system-immanenten, statistischen Modellparameter. Da die Adaption möglichst rasch und unter realistischen Bedingungen zur Wirkung kommen soll, werden gemischt-statistisch-heuristische Verfahren angewandt. Bei den zur Sprecheradaption geeigneten Systemparametern handelt es sich um die statistischen Modellparameter der *semikontinuierlichen Vektorquantisierung* (Codebücher), der *Hidden Markov Modelle*, der *Verwechslungsstatistik von Lautsymbolen*, sowie um *phonetisch sinnvolle Aussprachevarianten* im Lexikon.

Die Adaption des *semikontinuierlichen Codebuchs* erfolgt durch Verlagerung der Prototypen aufgrund von Beobachtungen, die während des normalen Erkennungsprozesses gesammelt werden. Die Kovarianzmatrizen der Prototypen bleiben dabei unverändert. Ziel der Verlagerung ist eine Erhöhung der Likelihood unter der Annahme, daß die Äußerung korrekt erkannt wurde. Es werden 11 verschiedene Verfahren untersucht. Das beste Verfahren (SCHWAM) erzielt eine relative Reduktion der Erkennungsfehler um ca. 26.9 % nach nur 16 gesprochenen Wörtern.

Die Adaption der *Hidden Markov Modelle* (HMM) konzentriert sich auf eine Neuabschätzung der sog. Mixtur-Koeffizienten innerhalb der einzelnen Zustände. Durch eine einfache, rekursive Berechnungsvorschrift werden die Modellparameter der an der erkannten Äußerung beteiligten HMM an den unbekannten Sprecher adaptiert. Zusätzlich können durch ein spezielles Ähnlichkeitsmaß weitere Zustände in nicht beteiligten HMM gefunden werden, die mit derselben Beobachtung adaptiert werden. Es werden 8 verschiedene Algorithmen untersucht. Das beste Verfahren (SCHMIX) erreicht eine relative Reduktion des Erkennungsfehlers von ca. 43.3 % nach 200 gesprochenen Wörtern.

Ein ähnliches Verfahren wurde zur Nachschätzung der *Verwechslungsstatistik von Lautsymbolen* nach der Erkennung angewandt. Diese Statistik dient zur Bewertung verschiedener Worthypothesen in einem Spracherkennungssystem nach dem 'bottom up'-Prinzip. Durch Beobachtung der sprecher-typischen Lautverwechslungen während des Betriebs ist eine schrittweise Adaption dieser Statistik möglich, was mit einer Verbesserung der Erkennungsleistung des Spracherkenners verbunden ist. Da es sich hier um diskrete Wahrscheinlichkeiten handelt, ist eine automatische Anpassung der Lernrate mit Hilfe eines Entropiemaßes möglich.

Für die Adaption des Spracherkennungssystems an die *typische Aussprache* eines unbekannten Sprechers werden zunächst allgemein gültige Regeln für die häufigsten Abweichungen von der korrekten Aussprache im Deutschen in das Lexikon inkorporiert. Damit erhöht sich jedoch die Ambiguität des Lexikons, und die Erkennung von bestimmten Wörtern wird etwas erschwert. Der Adaptionsalgorithmus beobachtet während der Benutzung des Systems durch den unbekannten Sprecher, welche von diesen Regeln eine korrekten Erkennung ermöglichen. Mit Hilfe dieser Statistik kann die Ambiguität des Lexikons schrittweise eingeschränkt und damit die Erkennungsleistung für diesen speziellen Sprecher

gesteigert werden.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die nach wie vor ungebrochene, stürmische Entwicklung im Bereich der Datenverarbeitung hat in den letzten Jahren zu immer leistungsfähigeren und gleichzeitig kostengünstigeren Maschinen geführt, und in dieser Entwicklung ist derzeit kein Ende abzusehen. Die Folge ist, daß im Bereich der Mensch-Maschine-Kommunikation der Trend zu mehr Ergonomie nunmehr in die Phase eingetreten ist, wo auch die Industrie vermehrt mit benutzerfreundlichen Applikationen auf den Markt kommt.

Neben den vielfältigen anderen Aspekten der Mensch-Maschine-Kommunikation ist nach wie vor die automatische Spracherkennung (ASE) eine der vielversprechensten Verbesserungen für eine ergonomische Dateneingabe. Sie bietet zudem den Vorteil, bei einer weiteren Miniaturisierung der Hardware praktisch keine unteren Limits zu setzen: ein Mikrophon läßt sich fast beliebig verkleinern, eine Tastatur nicht. Auch die Tatsache, daß praktisch alle potentiellen Anwender lesen und sprechen, jedoch nur wenige maschinenschreiben können, spricht dafür, daß bereits in naher Zukunft die ersten sprachgesteuerten Betriebssysteme auf den Markt kommen werden. Weitere, bereits klassische Anwendungsgebiete der ASE sind die Hilfe für Behinderte, das automatische Diktieren bzw. Protokollieren sowie die Steuerung von Geräten, wenn der Benutzer Hände und/oder Augen für andere Tätigkeiten benötigt.

Im Zuge dieser Entwicklung wird von den meisten Applikationen *Sprecherunabhängigkeit* gefordert, d.h. das ASE System wird nicht auf einen bestimmten Benutzer trainiert, sondern kann von beliebigen Sprechern verwendet werden. Sprecherunabhängigkeit läßt sich auf zwei prinzipiell unterschiedlichen Wegen realisieren: entweder das ASE System wird so entworfen und trainiert, daß es ohne weitere Maßnahmen während des Betriebs die Sprache aller nur denkbaren Sprecher gleich gut erkennt, dann spricht man von einem *sprecherunabhängigen* System. Im anderen Falle wird während des Erkennungsprozesses aktiv versucht, das ASE System der charakteristischen Sprache des aktuellen Benutzers anzupassen, dann spricht man von *sprecheradaptiver* ASE.

Nach wie vor sind sprecherabhängige Verfahren den sprecherunabhängigen in der Erkennungsleistung deutlich überlegen. Algorithmen zur *Sprecheradaption*, wie sie in dieser Arbeit untersucht werden, versuchen diese Lücke während des Betriebs eines ASE Systems möglichst rasch zu schließen oder zumindest zu verkleinern.

Die verschiedenen Verfahren der ASE lassen sich grob in drei grundsätz-

lich verschiedene Ansätze gliedern: die *geometrischen Konzepte*, die *statistischen Verfahren* und *regelbasierte Ansätze*.

Die geometrischen Konzepte basieren sämtlich auf der Annahme, daß Mustererkennung durch Berechnung und Vergleich von geometrischen Abständen in einem meist hoch-dimensionalen Vektorraum möglich ist. Bekannte Beispiele sind *Nächster-Nachbar-Klassifikator*, *dynamische Interpolation* und *dynamische Programmierung ('Dynamic Time Warping')* (vgl. [Rus88]). Sie wurden, bis auf die konnektivistischen Ansätze ('Neuronale Netze') in den letzten Jahren von den statistischen Verfahren weitgehend abgelöst bzw. zu statistischen Verfahren erweitert. Trotzdem finden sich aber gerade in kommerziellen Systemen oft noch bewährte Algorithmen, welche auf geometrischen Konzepten basieren. Für diese Verfahren sind sind zahlreiche Ansätze für Sprecheradaption bekannt (z.B. [Cla90], [Jas82], [Käm90]), welche fast alle auf eine lineare oder nicht-lineare Transformation der Merkmalsvektoren vor der eigentlichen Mustererkennung zurückzuführen sind.

Die regelbasierte Erkennung von Sprache konnte bisher nur mit wenig Erfolg in der ASE eingesetzt werden: sie krankt vor allem an den nicht vorhandenen, automatischen Trainingsverfahren, um große Datenmengen verarbeiten zu können. Die meisten Ansätze für die regelbasierte ASE versuchen, eine Sprecherabhängigkeit durch geeignete Wahl der Merkmale und Regeln von Anfang an auszuschließen. Sprecheradaptive Verfahren sind daher auf diesem Gebiet bisher nicht bekannt geworden.

Die Erkennung von Sprache mit Hilfe von statistischen Verfahren gründet auf der Annahme, daß es sich bei der Erzeugung von Sprache um einen stochastischen Prozeß handelt, d.h. mehr oder weniger große Segmente des Sprachsignals – also entweder einzelne oder Folgen von Merkmalsvektoren – werden als (Zufalls-)Ereignisse interpretiert, deren Auftreten im statistischen Mittel bestimmten Gesetzmäßigkeiten folgt. Der zugrunde liegende stochastische Prozeß kann dann durch die Parameter eines geeigneten statistischen Modells beschrieben werden. Diese Parameter oder Modellparameter werden durch Beobachtung einer möglichst großen Anzahl von Ereignissen geschätzt und erlauben es anschließend, die Auftretenswahrscheinlichkeit für alle möglichen Ereignisse des stochastischen Prozesses anzugeben. Die Merkmalsvektoren eines unbekannten Sprachsegments werden dann anhand ihrer Auftretenswahrscheinlichkeit und einer Entscheidungsregel bestimmten Klassen zugeordnet (vgl. auch [Rus88]).

Für die Sprecheradaption bieten die statistischen Verfahren mehrere Vorteile:

- Statistische Verfahren kommen in fast allen Verarbeitungsstufen in Systemen der ASE zur Anwendung.

- Bestimmte Modellparameter der verwendeten statistischen Modelle repräsentieren das Sprechverhalten eines oder mehrere Sprecher, und sind daher geeignete Ansatzpunkte für eine Adaption an einen unbekannten Sprecher. Die gezielte Manipulation an diesen Parametern nennen wir *stochastische Modellierung*.

- Ein statistisches Modell und seine Modellparameter können in bestimmten Fällen als *Generalisierungen* einer großen Anzahl von ähnlichen Ereignissen betrachtet werden. Durch die Beobachtung von nur einigen wenigen Ereignissen kann somit auf das statistische Verhalten von anderen, nicht beobachteten Ereignissen geschlossen werden. Dieses wichtige Prinzip der

Generalisierung bildet eine der Grundvoraussetzungen für eine erfolgreiche Sprecheradaption.

- Die statistischen Verfahren werden derzeit zweifellos von der Fachwelt der ASE favorisiert. Alle bekannten großen Systeme aus Industrie und Forschung[1] basieren auf statistischen Algorithmen.

Aus den genannten Gründen konzentriert sich die vorliegende Arbeit auf Methoden der Sprecheradaption in Systemen der ASE, welche auf statistischen Verfahren beruhen.

Die folgenden Abschnitte dieses einführenden Kapitels behandeln einige grundlegende Aspekte der Sprecheradaption in statistischen Systemen der ASE unter realistischen Bedingungen, sowie die sich daraus ergebenden Folgerungen für die technische Realisierung.

1.1 Einsatzbereich der Sprecheradaption

Es ist zunächst zu klären, in welchen Systemen der ASE eine Sprecheradaption überhaupt sinnvoll ist. Ein guter Anhaltspunkt hierfür bildet naturgemäß die Anzahl und Fluktuation der Benutzer, für die das System entworfen wurde.

Eine Diktiermaschine ('hörende Schreibmaschine') wird in den meisten Fällen nur von einem Benutzer bedient werden. Für solche 1-Benutzer-Anwendungen ist ein sprecherabhängiges System sicher sinnvoller, da einem Dauerbenutzer eine einmalige Trainingsphase von einigen Stunden zugemutet werden kann (vgl. z.B. das *TANGORA* System von *IBM*).

Das andere Extrem stellt beispielsweise ein Auskunftssystem dar, welches nur wenige Worte mit einem Benutzer austauscht, wie z.B. *MAX* von der Europäischen Gemeinschaft. Bevor eine Sprecheradaption überhaupt zum Tragen kommt, wird in den meisten Fällen die *Sitzung*, d.h. der Dialog des ASE Systems mit einem bestimmten Benutzer, bereits beendet sein. Normalerweise arbeiten solche Systeme sprecherunabhängig und versuchen das dadurch bedingte Handikap durch Reduktion des Wortschatzes und sehr enge Dialogführung auszugleichen.

Es bleibt der Bereich zwischen den beiden Extremen: ASE Systeme, die zwar von mehreren Sprechern, aber vom jeweiligen Sprecher über einen längeren Zeitraum genutzt werden. Die typische Mindestdauer einer Sitzung liege zwischen 50 und 500 Wörtern. Anwendungsbeispiele sind z.B. Literaturdatenbank-Abfrage, Steuerung von Betriebssystemen, Gerätesteuerungen. In solchen Systemen lohnt sich der Einsatz von sprecheradaptiven Algorithmen, sofern sie den Benutzer nicht zu sehr behindern.

Die in den folgenden Kapiteln vorgestellten Verfahren zur Sprecheradaption werden daher speziell für den Einsatz in einem System der ASE mit den o.g. Voraussetzungen entworfen und auch anhand eines solchen Systems evaluiert (vgl. Kapitel 2).

[1] *SPICOS ([Pae89/1]), DRAGON ([Bak89]), TANGORA ([Rig91]), SPHINX ([Lee89]), EVAR ([Ehr89]), SILBOS ([Wei90])*

1.2 Sprecheradaption unter realistischen Bedingungen

In den meisten veröffentlichten Untersuchungen zur Sprecheradaption werden Systeme unter sog. 'Laborbedingungen' getestet. Meistens bedeutet dies, daß umfangreiches Sprachmaterial des unbekannten Sprechers bereits vorliegt und der Inhalt dieses Materials u.U. sogar bekannt oder gar vorgeschrieben ist (z.T. phonetisch ausbalanciert). Die Adaption wird dann in einem Schritt durchgeführt und das adaptierte System anschließend mit (hoffentlich) unabhängigen Daten des unbekannten Sprechers getestet[2].

Ein solches Szenario ist natürlich für den realen Einsatz völlig ungeeignet. Leider hat dies meistens auch zur Folge, daß die untersuchten Methoden auf diese Laborbedingungen abgestimmt sind. Beispielsweise benötigen manche Verfahren eine ausreichende Menge von Sprachdaten bzw. korrekt beschriftete Sprachdaten des unbekannten Sprechers, um überhaupt zu funktionieren.

Das im vorangegangenen Abschnitt grob skizzierte System der ASE soll unter realistischen, ergonomischen Randbedingungen arbeiten, d.h. der Benutzer der Applikation soll sowenig wie möglich bei der Ausübung seiner eigentlichen Aufgabe behindert werden. Die ASE soll schließlich kein Selbstzweck, sondern lediglich eine alternative Form der Dateneingabe für verschiedene Applikationen sein. Behinderungen können z.B. sein: vorgeschriebene Adaptionsphasen, verpflichtende Korrektur, explizite Anmeldung, etc.

Es ergeben sich die folgenden Randbedingungen (Maximalforderungen) für eine ergonomische Sprecheradaption:

- Die Sprecheradaption erfolgt weitgehend *unüberwacht*, d.h. es erfolgt keine explizite Korrektur durch den Benutzer. Eine Ablehnung einer falschen Erkennung ist dagegen zulässig, da der Benutzer solche mißverstandenen Eingaben sowieso zurückweisen muß, um seine Aufgabe erfüllen zu können. Optional kann statt dessen ein Mausklick auf das korrekte Ergebnis in einer Top-N Liste erfolgen (*halbüberwachte Adaption*, vgl. [Bak89]).
- Es gibt keine vorgeschriebene *Adaptionsphase*, innerhalb derer der unbekannte Sprecher gezwungen wird, bestimmte Äußerungen zu sprechen.
- Der Einsatz der Sprecheradaption muß so schnell wie möglich erfolgen. Im Idealfall bereits nach dem ersten gesprochenen Wort bzw. Satz.
- Ein Wechsel des Sprechers kann jederzeit und ohne explizite Ankündigung erfolgen.
- Die Benutzer des ASE Systems können
  - männlich oder weiblich sein.
  - allen Altersstufen angehören.
  - aus verschiedenen Gegenden stammen (Dialekt).
  - verschiedenen Gesellschaftsschichten angehören.

[2] z.B. [Hat90], [Jas82], [Cla90], [Scm91], [Shi86], [Käm90], [Hua91], [Bon91], [Kub90], [Rig89], [Lee89], [Shi91]. Ausnahmen von dieser Praxis sind z.B. [Fur89], [Fun91], teilweise [Bam91]

– ein verschiedenes Ausbildungsniveau besitzen.
– in unterschiedlicher körperlicher Verfassung sein.

Diese Maximalforderungen sollten beim Entwurf von Verfahren zur Sprecheradaption als Richtschnur dienen. In den meisten Fällen können nicht alle Forderungen gleichzeitig erfüllt werden.

1.3 Sprecheradaption mittels stochastischer Modellierung

Systeme der ASE, welche auf statistischen Methoden basieren, beinhalten immer ein mehr oder weniger genaues *statistisches Modell* des *Sprachprozesses* eines oder mehrerer Sprecher. Mit Hilfe dieses Modells errechnet das System Rückschlußwahrscheinlichkeiten vom beobachteten Sprachsignal auf beliebige Entitäten der Sprache, und versucht anhand dieser Wahrscheinlichkeiten eine korrekte Entscheidung über die gesprochene Äußerung zu fällen.

Das statistische Modell wird bestimmt durch seine *Struktur* und seine *Parameter*, welche eine *statistische Wissenbasis* bilden. Der Erfolg des ASE Systems hängt einerseits davon ab, wie genau die gewählte Struktur in der Lage ist, den Sprachprozeß zu modellieren, andererseits davon, wie gut die Parameter des Modells geschätzt werden können. Die Struktur wird meistens einmal bestimmt und bleibt danach unverändert. Die Abschätzung der Parameter wird *Training* des ASE Systems genannt. Die *Trainingsstichprobe*, d.h. das Sprachmaterial, mit Hilfe dessen die Modellparameter geschätzt werden, legt fest, welche Sprache das System zu erkennen in der Lage ist. Idealerweise ist die Trainingsstichprobe sehr groß und beinhaltet eine repräsentative Auswahl aller möglichen Sprachäußerungen, damit alle Parameter des Modells sicher geschätzt werden können. Beides ist aus technischen Gründen nicht durchzuhalten. Daher wird in den meisten Fällen darauf vertraut, daß auch Parameter, welche anhand von nicht vollständig repräsentativen Daten geschätzt wurden, die gesamte Sprache korrekt modellieren. In den meisten Fällen ist dies aber nicht der Fall, weshalb durch verschiedene *Glättungsverfahren* auch solche Parameter, die während des eigentlichen Trainings nie Information erhalten haben, mit einigermaßen vernünftigen Werten belegt werden.

Sprecheradaption in diesem Sinne bedeutet also, die dafür geeigneten Parameter eines gegebenen (und bereits trainierten) statistischen Modells so zu manipulieren, daß das Modell anschließend den Sprachprozeß des aktuellen Benutzers besser beschreibt.

Bei der Sprecheradaption unter realistischen Bedingungen (s.o.) steht jedoch im Gegensatz zum konventionellen Training keine große Sprachstichprobe des unbekannten Sprechers zur Verfügung. Das in Kapitel 2 beschriebene System enthält z.B. ungefähr 77000 Modellparameter. 10 Sekunden Sprache (12 - 20 Wörter) werden dagegen nach der Vorverarbeitung mit nur 41000 Merkmalskomponenten vollständig beschrieben (20 + 20 + 1 Merkmale alle 10 msec). Selbst unter der optimistischen Annahme, daß jeder dieser 41000 Werte jeweils zur Abschätzung von 10 verschiedenen Modellparametern herangezogen werden kann, bedeutet dies, daß ein Parameter aus nur etwa 5 beobachteten Werten geschätzt werden soll. Dies ist mit dem 'Gesetz der großen Zahlen', welches dem stochastischen Ansatz zugrunde liegt, nicht zu vereinbaren. Erschwerend kommt

noch hinzu, daß beim Training eines ASE Systems die Zuordnungen von Sprachsegmenten auf sprachliche Entitäten (Klassen) natürlich bekannt sind; bei der Sprecheradaption sind sie es nur mit einer gewissen Wahrscheinlichkeit.

Damit und mit dem oben Gesagten läßt sich das *Grundproblem der Sprecheradaption* formulieren:

> Mit Standardverfahren der Statistik ist es nicht möglich, die Parameter eines stochastischen Modells mit wenig Daten sicher abzuschätzen.

Alle Versuche, übliche Trainingsalgorithmen direkt für die Sprecheradaption einzusetzen, führen daher zu unrealistischen Bedingungen (vgl. Abschnitt 1.2 und Kapitel 7).

Für das oben genannte Grundproblem der Sprecheradaption in statistischen Systemen zur ASE gibt drei grundsätzlich verschiedene Lösungsansätze[3]: die *Interpolation*, die *Generalisierung*, sowie die *Sprechertypisierung*.

1.3.1 Interpolation

Die aus wenigen Daten des unbekannten Sprechers schlecht geschätzten Parameter werden mit gut geschätzten, aber sprecherunspezifischen Parametern *interpoliert*, z.B. nach dem Prinzip der *deleted interpolation*[4]. Diese Methode ist in vielerlei Variationen in den meisten Algorithmen der Sprecheradaption zu finden. Das Problem dabei ist, daß aus solchen Algorithmen in den meisten Fällen ein 'unendliches Gedächtnis' resultiert, d.h. das System verfügt nicht über die Möglichkeit, Daten wieder zu 'vergessen' und somit aktuellen Daten ein höheres Gewicht zu geben. In den Kapiteln 3 und 4 wird dieser entscheidende Punkt ausführlich behandelt.

1.3.2 Generalisierung

Die Parameter von so umfangreichen statistischen Modellen, wie sie in Systemen zur ASE typischerweise vorkommen, sind zumindest teilweise redundant. Das erklärt sich ganz einfach aus der Tatsache, daß die einzelnen Modelle sprachliche Entitäten modellieren, welche teilweise die gleichen oder sehr ähnliche Sprachereignisse beinhalten. Z.B. enthält ein System, welches *Ganzwortmodelle* zur Erkennung verwendet (vgl. Abschnitt 2.2.5), in mehreren Wortmodellen Parametersätze, die weitgehend die gleichen Laute modellieren, einfach deshalb, weil diese Laute praktisch identisch in den zugehörigen Wörtern vorkommen. Das gleiche gilt für Silben, Halbsilben, Konsonantenfolgen, Vokalcluster, sogar für ähnlich klingende Phoneme.

Mit Hilfe dieser Redundanz der Parameter können Beobachtungen, welche eigentlich nur einen bestimmten, eingegrenzten Bereich des Modells zugeordnet werden können, auf andere Parameter *generalisiert* werden. Man gewinnt dadurch eine Verteilung der gewonnenen Information auf größere Bereiche des Modells. In den Kapiteln 3, 4 und 9 werden Verfahren vorgestellt, die mit Hilfe von Generalisierungstechniken bessere Ergebnisse erzielen als mit Interpolation allein.

[3] In den meisten Fällen werden mehrere dieser Lösungsansätze kombiniert

[4] Lineare Interpolation, wobei die Faktoren der einzelnen Anteile von der Anzahl der Beobachtungen, die zu diesen Anteilen geführt haben, bestimmt werden.

Generalisierung wurde bisher in der Literatur nur wenig erwähnt. In [Shi91] wurde die Adaption der Mittelwerte von Gaußverteilungen in den Zuständen kontinuierlicher Hidden Markov Modelle (HMM) durch Generalisierung von beobachteten HMM auf nicht beobachtete HMM leicht verbessert.

1.3.3 Sprechertypisierung

Der dritte Lösungsansatz für das oben formulierte Grundproblem ist die *Sprechertypisierung*. Das ASE System wird dabei nicht an den unbekannten Sprecher adaptiert, sondern greift auf vorab berechnete, komplette Parametersätze zurück, die auf den Sprachprozeß dieses Benutzers am besten passen.

Auch dieser Ansatz wurde bereits in mehreren verschiedenen Variationen untersucht[5]. In den meisten Fällen wird der unbekannte Sprecher anhand einer Sprachstichprobe auf einen von S *Sprechertypen* klassifiziert. Diese Zuordnung kann auch schon während des normalen Erkennungsprozesses erfolgen, beispielsweise durch parallele HMM für jeden Typus ([Mat90]). Für jeden Sprechertypus in S existiert ein kompletter Satz von Modellparametern, auf den das ASE System nach erfolgter Klassifikation einfach 'umschaltet'. Erstaunlicherweise sind die mit solchen Verfahren erzielten Ergebnisse nicht so gut, wie man es erwarten würde. Die Gründe hierfür könnten sein:

- Verschiedene Sprecher sind schwer zu Sprechertypen zusammenzufassen[6].
- Die Parameter des einzelnen Sprechertypus müssen natürlich auch trainiert werden. Die Daten hierfür stammen aus einer großen Trainingsstichprobe mit vielen Sprechern. Durch die Einteilung in Typen entfallen auf den einzelnen Parametersatz natürlich viel weniger Trainingsdaten, als insgesamt vorhanden sind. Die Folge ist: statt eines gut geschätzten Parametersatzes verfügt das ASE System nunmehr über viele sehr schlecht geschätzte Parametersätze.

Weitere Nachteile der Verfahren der Sprechertypisierung sind:

- Ein gewisses Minimum an Sprachmaterial des unbekannten Sprechers muß bereits vor dem Beginn der Sitzung vorliegen.
- Der Speicher- und Trainingsaufwand ist um Größenordnungen höher als bei echten Adaptionsverfahren.

Vor allem aus dem zuletzt genanntem Grund wird in dieser Arbeit der Ansatz der Sprechertypisierung nicht verfolgt.

Ein naheliegender Gedanke ist die Kombination der Sprechertypisierung mit den beiden anderen Ansätzen auf folgende Weise:
Während der Sitzung mit einem unbekannten Sprecher werden die Modellparameter mit schritthaltenden Verfahren auf diesen adaptiert. Am Ende der Sitzung erhält er die Möglichkeit, diesen nun sprecherspezifischen Parametersatz unter einem Passwort bzw. seinem Benutzernamen abzuspeichern. Bei einer erneuten Verwendung des Systems kann (nach einer Identifikation) auf diese Parameter zurückgegriffen werden. Auf diese Weise wird lediglich der Aufwand an Speichermedien erhöht, was in den meisten Fällen kein Problem darstellt.

[5] Z.B. in [Mat90], [Lee89]

[6] In den meisten Untersuchungen werden zur Kategorisierung Standard-Cluster-Algorithmen verwendet

1.4 Sprecheradaption in verschiedenen Verarbeitungsstufen

Ein System der ASE besteht im allgemeinen aus mehreren *Verarbeitungsstufen*, welche in den meisten Fällen eine *Repräsentation* der eingegebenen Sprache in eine andere überführen (vgl. Bild 1.1). Z.B. transformiert die *Vorverarbeitung* das eindimensionale Sprachsignal in eine Folge von mehrdimensionalen Merkmalsvektoren, usw.

Beim Entwurf eines sprecheradaptiven Systems der ASE erhebt sich daher ganz zu Beginn die Frage, auf welcher Repräsentationsebene oder in welcher Verarbeitungsstufe die Sprecheradaption durchgeführt werden soll.

Grundsätzlich gibt es zwei verschiedene Verfahren der Sprecheradaption: Die Adaption der Modellparameter in einer bestimmten Verarbeitungsstufe oder die Transformation einer bestimmten Repräsentation des Sprachsignals (vgl. Bild 1.1 rechter bzw. linker Teil)

Die Methode der Transformation läßt sich in praktisch allen Typen der ASE einsetzen, nicht nur in statistischen Erkennern. Z.B. kann Sprecheradaption durch eine konstante Matrixmultiplikation der Merkmalsvektoren erreicht werden ([Cla90], [Jas82]), oder durch direkte Abbildung des vektorquantisierten Datenstroms mittels einer Tabelle (*codebook mapping*, [Fur89]). Solche Verfahren der Transformation werden in dieser Arbeit nicht untersucht.

Die Adaption von statistischen Modellparametern kann im Prinzip in allen Verarbeitungsstufen durchgeführt werden, welche über stochastische Modelle verfügen. In dem Beispiel eines 'bottom up' Systems in Bild 1.1 sind dies *Vektorquantisierung, Klassifikation, Lexikonstufe, Sprachmodell* und ev. noch weitere, 'höher' angeordnete Stufen.

In den Kapiteln 3, 4, 5 und 9 werden für diese Verarbeitungsstufen mehrere Algorithmen der Sprecheradaption vorgestellt und untersucht. Kapitel 7 beschäftigt sich mit den Problemen, die bei einer Kombination mehrerer solcher Adaptionsverfahren auftreten.

Abbildung 1.1: Vereinfachtes Schichtenmodell eines 'bottom up' System der ASE basierend auf stochastischer Modellierung

Kapitel 2

Spracherkennungssystem

Zusammenfassung

Das System zur automatischen Spracherkennung, welches als Grundlage für die meisten der in dieser Arbeit untersuchten Verfahren der Sprecheradaption dient, wird kurz beschrieben. Die Beschreibung umfaßt die komplette Verarbeitung vom Zeitsignal bis zur Wortklasse sowie einige Bemerkungen zum verwendeten Sprachmaterial.

2.1 Vorbemerkungen

Verfahren der Sprecheradaption sind in den meisten Fällen nur sehr schwer von den Methoden der Spracherkennung selbst, auf die sie wirken, zu abstrahieren. Auch bei der direkten Adaption von Merkmalsvektoren (z.B. [Cla90]) wird doch immer eine bestimmte Konfiguration vorausgesetzt, und die Verfahren sind auf diese optimiert. Eine erschöpfende Untersuchung aller denkbaren Verfahren der Adaption in verschiedenen Systemkonfigurationen zur Spracherkennung würde den Rahmen dieser Arbeit bei weitem sprengen.

Daher wurde ein relativ einfaches System entworfen, dessen Eigenschaften jedoch in den wesentlichen Punkten den derzeit erfolgreichsten Verfahren zur Erkennung gesprochener Sprache mittels stochastischer Modellierung entsprechen. Zahlreiche Untersuchungen (z.B. [Hua88]), auch am Lehrstuhl für Datenverarbeitung ([Pla91]) haben gezeigt, daß semikontinuierliche HMM sowohl den einfachen diskreten als auch den rein kontinuierlichen HMM in der Anwendung[1] deutlich überlegen sind. Aus diesem Grunde wurde ein ASE System zur Erkennung isoliert gesprochener Wörter auf der Basis semikontinuierlicher HMM entworfen.

Die prinzipielle Arbeitsweise des Erkenners sowie seine Randbedingungen werden in den folgenden Abschnitten beschrieben. Unter anderem wird auf seine zwei möglichen Arbeitsweisen näher eingegangen: Verarbeitung von Ganzwort-Modellen oder von verknüpften Phonem-Modellen.

[1] d.h. insbesondere, wenn kein unendlich großes Trainingsmaterial vorhanden ist

2.2 Aufbau des ASE Systems

2.2.1 Sprachmaterial

Als Sprachmaterial dienen die sog. *Berliner Sätze* nach [Sot84]. Die Sätze sind einfache Aussage und Fragesätze, welche den Tagesablauf einer Familie beschreiben, und enthalten alle im Deutschen vorkommenden Laute in typischer Häufigkeit. Sie wurden von 10 männlichen und weiblichen Sprechern meist sehr flüssig, zum Teil auch sehr langsam und leider sehr unterschiedlich laut gesprochen, analog aufgenommen und anschließend mit 16 kHz in 16 bzw. 14 Bit abgetastet.

Teilweise sind Rauschen und andere Störgeräusche beigemischt. Diese Daten wurden, um eine gerechte Beurteilung zu ermöglichen, ausgesondert.

Die stark unterschiedliche Aussteuerung der Signale führt zu erheblichen Problemen bei der Spracherkennung. Eine Notlösung ist die Normierung des Merkmals Gesamt-Lautheit *la* auf das absolute Maximum eines Satzes (vgl. Abschnitt 2.2.2). Dies kann in einem Echtzeitsystem nur annähernd durch gleitende Fenster während der Vorverarbeitung realisiert werden. Die beste Lösung ist eine vernünftige, analogseitige Aussteuerung des Signals.

Als Basiseinheit für Training, Adaption und Evaluierung dient das *einzelne Wort*. Die Wörter werden anhand einer von Hand vorgenommenen Segmentierung aus der fließenden Sprache extrahiert. In Folge der schnellen Sprechgeschwindigkeit sind vor allem die Segmente von *Funktionswörtern* extrem kurz (z.T. weniger als 10 msec), was die Erkennung solcher Wörter sehr erschwert (z.B. 'ist', 'die', 'an', etc.). Zusätzlich erhalten die initialen und finalen Laute der Wörter durch die verschiedenen Koartikulationseffekte über die Wortfugen hinweg eine weitaus höhere Variabilität als es bei isoliert gesprochenen Wörtern der Fall ist.

Da alle hier vorgestellten Verfahren von einer zunächst sprecherunabhängigen Erkennung ausgehen, wird eine gemischt-geschlechtliche *Trainingsdatenbasis* aus 6 Sprechern (3 männlich, 3 weiblich) zusammengestellt. Sie enthält insgesamt 1200 Sätze mit 341 verschiedenen Wörtern[2] , d.h. die *Berliner Sätze* in 12 Versionen, je 200 von einem Sprecher.

Zur Evaluierung des Spracherkenners und der verschiedenen Verfahren der Adaption werden die Daten von insgesamt 3 Testsprechern (2 männlich, 1 weiblich) zu einer *Testdatenbasis* zusammengestellt. Die Daten jedes dieser Sprecher werden wiederum unterteilt in drei verschiedene Sprachstichproben (*Adaption, Test1* und *Test2*). Die Sprachstichprobe *Adaption* (200 Wörter) ist sowohl mit *Test1* als auch mit *Test2* disjunkt. *Test2* mit 478 Wörtern ist eine Obermenge von *Test1* (200 Wörter). Der Wortschatz der drei Stichproben wurde per Zufall ausgewählt. Anhang B enthält die Wortinventare und eine Lautstatistik sowohl der drei Stichproben als auch der gesamten Sprachdatenbasis *Berliner Sätze*. Der Vergleich zeigt, daß in allen Sprachstichproben die Häufigkeiten der Phoneme etwa denen der gesamten *Berliner Sätze* entsprechen.

2.2.2 Vorverarbeitung

Das gefilterte und mit 16 kHz abgetastete Sprachsignal wird einer gehörgerechten Vorverarbeitung ([Rus91]) und Merkmalsextraktion unterworfen. Als Ergebnis liegen alle 10 msec folgende Merkmalsvektoren vor:

[2] Davon werden aus technischen Gründen nur 339 Wörter verwendet

- Auf die Gesamtlautheit normiertes Lautheitsspektrum $\underline{sh}(t)$ in 20 Barkstufen ([Zwi82]).
- Differenzspektrum $\underline{di}(t)$ analog zu Levinson ([Lev89]):

$$\underline{di}(t) = \sum_{j=-2}^{2} j \; \underline{sh}(t+j) \tag{2.1}$$

- Gesamt-Lautheit $la(t)$ nach Zwicker ([Zwi82]) normiert auf die maximale Lautheit des gesprochenen Satzes, aus welchem das Muster stammt.

Außerdem werden die Positionen der Silbenkerne durch Auswertung der modifizierten Lautheit automatisch angezeigt (siehe analog [Rus88], S. 117 ff). Auf Anfangs- und Ende-Detektion wird verzichtet, da die Trainings- und Test-Muster (Wörter) anhand einer vorgegebenen Segmentierung aus fließend gesprochener Sprache entnommen werden.

2.2.3 Vektorquantisierung und Codebücher

Unter dem Begriff *Vektorquantisierung* versteht man im allgemeinen eine Methode, das Auftreten eines (Zufalls-)Ereignis im Vektorraum (Merkmalsvektor) auf eine begrenzte Anzahl von diskreten Ereignissen abzubilden. Im Gegensatz zur *Quantisierung*, bei der der Vektorraum in reguläre Teilräume unterteilt und lediglich das Auftreten eines Ereignisses in dem betreffenden Teilraum registriert wird, können bei der Vektorquantisierung sog. *Prototypen* oder *Centroiden* an beliebigen Stellen des Vektorraums verteilt werden. Die o.g. Abbildung besteht in letzterem Falle aus der Zuordnung des Merkmalsvektors zu einem bestimmten Prototypen (normalerweise zu dem, mit dem geringsten Abstand). Die Gesamtheit der Prototypen nennt man *Codebuch*. Man erhält sie durch ein geeignetes *Clusterverfahren*, welches aus einer möglichst großen und repräsentativen Stichprobe von Merkmalsvektoren eine vorgegebene Anzahl von Prototypen so berechnet, daß der Quantisierungsfehler minimiert wird. Dies ist sinnvoll, wenn man annimmt. daß die Merkmalsvektoren nicht gleichmäßig über den Vektorraum verteilt sind, sondern einzelne Häufungen (eng. 'cluster') bilden.

Das Ergebnis einer Vektorquantisierung ist also im einfachsten Falle eine Nummer pro Merkmalsvektor, nämlich die Ordnungszahl des Prototypen, auf den der Merkmalsvektor abgebildet wurde. Diese Abbildung ist irreversibel. Um eine genauere Beschreibung des beobachteten Merkmalsvektors zu erhalten, werden außer dem Protoypen auch die Varianzen des Clusters, aus welchem er während des Clusterverfahrens berechnet wurde, berücksichtigt. Ein Codebuch für einen N-dimensionalen Merkmalsvektor besteht in diesem Falle aus dem N-dimensionalen Prototypen (auch Mittelpunktsvektor genannt) und einer N x N großen *Kovarianzmatrix* ([Rus88]).

Zur Abschätzung der Codebuch-Prototypen und ihrer zugehörigen Varianzen in dem hier beschriebenen System werden aus der Trainingsdatenbasis über alle Sprecher gleichmäßig verteilte Merkmalsvektoren ausgewählt und mit dem LBG-Algorithmus nach [Lin80] in M Cluster aufgeteilt.

Die Größe der Codebücher M hat starken Einfluß auf die Leistung des Spracherkenners. Größere Codebücher bedeuten einerseits kleinere Quantisierungsfehler und damit eine bessere Modellierung, andererseits aber mehr Re-

chenaufwand in der Vektorquantisierung und Schwierigkeiten bei der Parameterabschätzung sowohl des Codebuchs als auch der HMM. Da demnach eine optimale Codebuchgröße nur schwer vorherzusagen ist, wird der beste Wert empirisch aus sprecherunabhängigen Erkennungsexperimenten ermittelt ([Str91]). Es ergibt sich ein optimaler Wert von $M = 64$, wobei zu bemerken ist, daß das eindimensionale Merkmal Lautheit natürlich auch mit einem kleineren Wert für M quantisiert werden könnte (z.B. [Wei90]). Zur Vereinfachung wird aber auch hier 64 gewählt.

In Voruntersuchungen mit den Daten eines einzelnen Sprechers wurde das Verhältnis von Distorsion[3] auf Testdaten fremder Sprecher und Distorsion auf unabhängige Daten des Trainingssprechers bei steigender Zahl von Trainingsvektoren bestimmt. Als Ergebnis läßt sich sagen, daß ca. 12000 Trainingsvektoren für das Codebuch ausreichen, um das Sprachmaterial eines Sprechers zu repräsentieren. Eine weitere Erhöhung der Anzahl von Trainingsvektoren bringt keine Verbesserung mehr.

Die Codebücher für die sprecherunabhängige Trainingsdatenbasis werden daher aus ca. 120000 Trainingsvektoren berechnet (6 Sprecher á 20000 Vektoren). Die Distorsion unabhängiger Daten der Trainingssprecher auf dieses Codebuch entspricht dem Wert der Distorsion auf ein Codebuch, das mit 12000 Trainingsvektoren eines einzelnen Sprecher erstellt wurde. Dies kann als Hinweis gelten, daß die Sprachdaten aller 6 Sprecher ausreichend genau repräsentiert werden.

Der Vergleich mit den Daten eines Sprechers der Testdatenbasis ergibt einen 147 % höheren Wert als auf die Sprecher der Trainingsdatenbasis.

Die diagonalisierten Kovarianzmatrizen werden aus den Clustern im letzten Iterationsschritt des LBG-Algorithmus geschätzt (analog zu [Rus88], S. 60).

2.2.4 Semikontinuierliche Vektorquantisierung

Jedes *semikontinuierliche Codebuch* für einen Merkmalsvektor k ($k = sh, di, la$) besteht aus M Prototypen $\underline{s}_k(m)$, $m = 1 \ldots M$ der gleichen Dimension wie der zugehörige Merkmalsvektor. Außerdem enthält das Codebuch zu jedem Prototypen die diagonalisierte Kovarianzmatrix $\underline{D}_k(m)$ zur Beschreibung der multivarianten, bedingten Wahrscheinlichkeitsdichte:

$$p(\underline{k}(t)|m) = \frac{1}{\sqrt{(2\pi)^N \ |\underline{D}_k(m)|}} \quad e^{-\frac{1}{2}\ (\underline{k}(t)-\underline{s}_k(m))' \underline{D}_k(m)^{-1} (\underline{k}(t)-\underline{s}_k(m))} \qquad (2.2)$$

Nach dem klassichen Maximum-Likelihood-Ansatz bilden obige bedingte Wahrscheinlichkeiten den Input für Training und Test von *semikontinuierlichen Hidden Markov Modellen* (SCHMM, z.B. [Hua88]). Aufgrund der z.T. hohen Dimensionalität der Merkmalsvektoren ergeben sich dabei nummerische Probleme (Größenordnung der Determinante im Bereich von 10^{80}, etc.). Infolgedessen ist oft die bedingte Wahrscheinlichkeit zu nur einem einzigen Prototypen nummerisch > 0 und das SCHMM reduziert sich wieder zu einem konventionellen diskreten HMM.

In Anlehnung an Verfahren der Abstandsklassifikatoren werden daher in diesem ASE System für die bedingte Wahrscheinlichkeit nach Gl. 2.2 die folgende Näherungen verwendet.

[3] mittlerer Fehler bei der Vektorquantisierung

Sei $d_G(\underline{k}(t), m)$ der gewichtete Euklidische Abstand vom Merkmalsvektor $\underline{k}(t)$ zum Prototypen $\underline{s}_k(m)$

$$d_G(\underline{k}(t), m) = (\underline{k}(t) - \underline{s}_k(m))' \underline{D}_k(m)^{-1} (\underline{k}(t) - \underline{s}_k(m)) \tag{2.3}$$

und $\underline{p}_k(t)$ ein M-dimensionaler Vektor mit den *Pseudorückschlußwahrscheinlichkeiten* für den Merkmalsvektor $\underline{k}(t)$ auf die M Prototypen $\underline{s}_k(m)$, $m = 1 \ldots M$. Dieser wird in zwei Varianten berechnet:

Variante 1:

$$\underline{p}_k(t) = [\ldots, p_k(t,m), \ldots]' \qquad m = 1 \ldots M \tag{2.4}$$

$$p_k(t,m) = \frac{1}{Y\ (W + d_G(\underline{k}_k, m))} \tag{2.5}$$

Y ist dabei lediglich ein Normierungsfaktor, der dafür sorgt, daß die Summe über alle Elemente von $\underline{p}_k(t)$ gleich 1 ist. Die konstante Größe $W = 10^{-10}$ verhindert unendliche Werte, falls der Abstand tatsächlich zufällig Null werden sollte.

$$Y = \sum_{m=1}^{M} \frac{1}{W + d_G(\underline{k}_k, m)} \tag{2.6}$$

Variante 2:

$$\underline{p}_k(t) = [\ldots, p_k(t,m), \ldots]' \qquad m = 1 \ldots M \tag{2.7}$$

$$p_k(t,m) = \frac{e^{-\frac{1}{2}\, d_G(\underline{k}_k, m)}}{Y} \tag{2.8}$$

mit

$$Y = \sum_{m=1}^{M} e^{-\frac{1}{2}\, d_G(\underline{k}_k, m)} \tag{2.9}$$

Auf diese Varianten, insbesondere ihre Auswirkung auf die Mixtur-Verteilungen in den Zuständen der SCHMM, wird in späteren Abschnitten eingegangen.

Ergebnis der Vektorquantisierung sind also die Pseudorückschlußwahrscheinlichkeiten der drei Merkmalsvektoren auf alle M Prototypen des zugeordneten Codebuchs in jedem Zeitschritt.

2.2.5 Semikontinuierliche Hidden Markov Modelle

Semikontinuierliche Hidden Markov Modelle sind in der Literatur oftmals ausführlich beschrieben worden (z.B. [Hua88], [Str91], etc.). Daher wird hier auf den Grundformalismus nicht näher eingegangen.

Als Struktur wird die gängige Links-rechts-Struktur mit einem möglichen Überspringer gewählt (vgl. Abb. 2.1). Die SCHMM werden mit einer Variante des sog. *segmental-k-means* ([Jua90]) trainiert und mit dem Viterbi-Algorithmus (z.B. [Rab86]) abgearbeitet. Die Formeln für Training und Test sind im Anhang C wiedergegeben.

Ganzwort-Modelle

Für jedes Wort der Datenbasis wird genau ein SCHMM trainiert. Für das Training analog zu *segmental-k-means* ist eine Vorbelegung der *Mixtur-Koeffizienten* in den einzelnen Zuständen notwendig (*seed models*). Dazu werden die Wörter der Trainingsstichprobe zunächst einer silbenorientierten Segmentierung in phonemähnliche Einheiten unterzogen und die entstehenden Partitionierungen auf eine festgelegte Grundstruktur abgebildet. Eine der untersuchten Grundstrukturen war z.B. pro Silbenkern 1 Zustand und max. 3 Zustände für dazwischen, davor oder dahinter liegende Konsonantenfolgen. Aus der Mittelung der in solche Grundzustände gefallenen Daten ergibt sich die Mixtur-Verteilung der *seed models*. Die genaue Vorgehensweise entnehme man [Str91].

Als Ausgangspunkt für das Training stehen dann pro Wort ein oder mehrere Ganzwortmodelle zur Verfügung, deren maximale Zustandszahl Z_{max} abhängig von der Zahl der detektierten Silben S in den Trainingsdaten ist: $Z_{max} = 4\ S+3$ D.h. einsilbige Wörter haben maximal 7, zweisilbige Wörter maximal 11 Zustände, usw. Die Übergangswahrscheinlichkeiten a_{ij} werden zunächst als gleichverteilt angenommen.

Alle diese Modelle werden mit dem gesamten zur Verfügung stehenden Trainingsmaterial trainiert und anschließend das Modell selektiert, dessen mittlere Emissionswahrscheinlichkeit in Bezug auf die Trainingsdaten maximal ist. Auf diese Weise entsteht pro Wort der Datenbasis genau ein Ganzwort-Modell (s. Bild 2.1 (a)).

Die Abarbeitung der Modelle in der Testphase geschieht mit Hilfe des Viterbi-Algorithmus. Die Entscheidung für eine Wortklasse geschieht durch die maximal erzielte Emissionswahrscheinlichkeit. Zur Berechnung der Emissionswahrscheinlichkeit eines Zustands werden nur die R größten Pseudorückschlußwahrscheinlichkeiten der VQ berücksichtigt (vgl. auch [Hua88]). Der optimale Wert für R in der sprecherunabhängigen Anwendung schwankt zwischen 2 und 5. Für alle Untersuchungen in dieser Arbeit wurde $R = 3$ gewählt.

Diese Variante des Spracherkenners soll die Vielzahl von Worterkennern mit kleinem Wortschatz repräsentieren, die zur Zeit auch schon kommerziell zum Einsatz kommen. Natürlich ist der Wortschatz eines solchen Systems gezwungenermaßen stark begrenzt. In dem oben beschriebenen System wurden die 339 Wörter der *Berliner Sätze* nach Sotschek ([Sot84]) trainiert.

Phonem-Modelle

Ein Phonem-Modell hat im Prinzip den gleichen Aufbau wie ein Ganzwort-Modell (s.o.). Es repräsentiert jeweils einen von 40 Phonemen der deutschen Sprache nach [Sam90] (s. Anhang A).

Die automatische Erstellung der *seed models* erfolgt allerdings ohne zusätzliche Orientierung an den Silbenkernen, da dies für so kurze Einheiten kaum sinnvoll erscheint. Deshalb ist die Anzahl der maximal möglichen Zustände Z_{max} eines Phonem-Modells konstant gleich 3. Die Trainingsdaten stammen aus einer von Hand vorgenommenen Segmentierung des Trainingsmaterials durch Studenten der Phonetik am Institut für Phonetik und sprachliche Kommunikation (IPK) der Ludwig Maximilian Universität München (s. dazu auch Abschnitt 2.2.1). Das Trainingsmaterial umfaßt im Mittel etwa nur 200 Phonem-Realisierungen pro Phonem-Modell, um realistische Bedingungen zu schaffen.

Abbildung 2.1: Die Strukturen eines Ganzwort-Modells (a) und eines virtuellen Wortmodells (b)

Die Abarbeitung erfolgt wiederum mit Hilfe des Viterbi-Algorithmus, diesmal jedoch angewandt auf ein sog. *virtuelles Wortmodell*, welches durch Verknüpfung von Phonem-Modellen entsteht (vgl. Abb. 2.1 (b)). Die Struktur eines solchen virtuellen Wortmodells entspricht der eines Ganzwort-Modells, bis auf die Verknüpfungsstellen der beteiligten Phonem-Modelle: Über diese dürfen keine Überspringer stattfinden. Die Verknüpfung selber richtet sich nach einem einfachen Aussprache-Graphen, welcher ausschließlich die *kanonische Form* ([Dud90]) des betreffenden Wortes auf die 40 SAM-Phoneme abbildet.

In den virtuellen Wortmodellen sind lediglich Zeiger eingetragen, welche auf die entsprechenden Phonem-Modelle deuten. Mit anderen Worten: auch wenn hunderte von virtuellen Wortmodellen das gleiche Phonem, etwa /a:/ enthalten, so existiert das eigentliche Phonem-Modell für /a:/ nur ein einziges Mal im Speicher; die virtuellen Wortmodelle dagegen enthalten nur Verweise darauf.

Diese Struktur bietet mehrere Vorteile:

1. Der Speicherplatz für virtuelle Wortmodelle ist extrem gering. Notfalls können bei sehr großen Lexica die virtuellen Wortmodelle on-line für die Klassifikation gebildet werden.

2. Der Wortschatz des Erkenners ist vom Training theoretisch völlig unabhängig, da lediglich die Aussprache-Beschreibung nach Duden ausgewechselt werden muß. In der Praxis ist allerdings der Diskursbereich des Trainingsmaterials sehr wohl für die Erkennungsleistung bedeutend. Tatsächlich wird ein solcher Wechsel des Diskursbereichs in dieser Untersuchung nicht vorgenommen. Für Demonstrationszwecke wurde als Beispiel ein Aussprache-Lexikon mit englischen UNIX-Befehlen zusammengestellt. Trotz des völlig veränderten Kontexts ergaben sich immer noch gute Erkennungsergebnisse.

3. Über die Pointer-Struktur der virtuellen Wortmodelle wird automatisch eine Generalisierung für die Adaption geleistet. D.h. wenn aus einem be-

liebigen Wort X Information für die Adaption der beteiligten Phonem-Modelle gewonnen wird, kommt dies automatisch allen Wörtern $A \ldots Z$ zugute, die ebenfalls eines der adaptierten Phonem-Modelle verwenden.

Mit Hilfe dieser Variante des Erkenners sollen Adaptionsverfahren untersucht werden, die kleinere Spracheinheiten zur Generalisierung von Information verwenden. Dies müssen nicht Phoneme sein. Die in dieser Arbeit vorgestellten Verfahren sind analog auf Diphone, Triphone, Konsonantenfolgen, Halbsilben oder Silben anwendbar. Generell aber gilt: je kleiner die gewählten Spracheinheiten, desto kleiner die Klassenzahl und desto größer die Fähigkeit zur Generalisierung. Das verwendete Aussprache-Lexikon umfaßt wieder die 339 Wörter der 100 Berliner Sätze.

Bild 2.2 zeigt noch einmal das Blockschaltbild des gesamten ASE Systems in den beiden Varianten Ganzwort-Modelle und Phonem-Modelle.

2.3 Simulation der Sprecheradaption

Der in den vorangegangenen Abschnitten beschriebene Spracherkenner dient zur Erkennung isoliert gesprochener Wörter. Durch das Training auf eine Sprechergruppe ist er zunächst weitgehend sprecherunabhängig. Mit Hilfe dieses Systems sollen nun verschiedene Verfahren der Sprecheradaption untersucht und beurteilt werden.

Dazu muß der Spracherkenner zusammen mit den jeweils untersuchten Adaptionsalgorithmen ausgestattet und in eine geeignete Testumgebung integriert werden, die es gestattet, die Anwendung eines solchen Systems in der realen Welt zu simulieren. Insbesondere müssen zusätzlich auch Verfahren zur Beurteilung der Adaptionsfähigkeit vorhanden sein. Mit Hilfe dieser Simulation soll der Ablauf einer Adaption des ASE Systems an einen unbekannten Sprecher nachgebildet werden (*Sitzung*). Der unbekannte Sprecher soll das System ohne vorherige Anmeldung oder Trainingsphase sofort benutzen. Der zeitliche Ablauf einer Sitzung wird auch *Prozeß* genannt.

Zu diesem Zweck wird eine im Hintergrund arbeitende Steuerung entworfen, welche bedeutend mehr Information über den Prozeß zur Verfügung hat als das eigentliche ASE System. In Bild 2.3 ist der Datenfluß zwischen den verschiedenen Komponenten des Gesamtsystems schematisch dargestellt. Parameter, Steuerungsbefehle und Ergebnisse sind mit normalen Pfeilen, Sprachdaten und Erkennungsergebnisse mit strukturierten Pfeilen dargestellt.

Die Steuerung verwaltet die Wortinventare, welche dem Spracherkenner im Laufe einer Simulation angeboten werden, prüft das Ergebnis, führt auf Wunsch Korrekturen durch, steuert den Zugriff auf die verschiedenen Datenbasen, führt zu bestimmten Zeitpunkten Tests mit unabhängigen Stichproben durch, um die geleistete Adaption zu beurteilen, registriert verschiedene, zeitlich sich ändernde Parameter des Prozesses und simuliert auch das Verhalten eines menschlichen Benutzers. Der Verlauf der Sitzung wird schritthaltend in eine Datei protokolliert.

Auf eine detailierte Beschreibung des Simulators muß hier aus Platzgründen verzichtet werden. Die wichtigsten Funktionen und Eigenschaften werden aus den Beschreibungen der einzelnen Versuche und Vorversuche ohnehin deutlich.

Abbildung 2.2: Blockschaltbild des ASE Systems

Abbildung 2.3: System zur Simulation der Sprecheradaption (siehe Text)

Kapitel 3

Adaption von Codebüchern

Zusammenfassung

Sprecheradaption der semi-kontinuierlichen Codebücher auf einen neuen Sprecher ohne die nachgeschalteten SCHMM (Ganzwortmodelle) zu verändern. Die in der Adaptionsphase beobachtete Datenmenge ist so klein, daß anstelle von rein statistischen Verfahren heuristische Adaptionsstrategien Anwendung finden. Hauptziel ist dabei die Vermeidung von Überadaption an die wenigen, beobachteten Daten, sowie die Erhaltung der inneren Struktur der Codebücher.

Die Versuchsergebnisse zeigen einen deutlichen Gewinn innerhalb weniger Wörter des neuen Sprechers. Folglich scheinen diese Verfahren besonders zur raschen Adaption geeignet. Für eine optimale Adaption wird ein 2-stufiges Verfahren vorgeschlagen.

3.1 Grundgedanke

Die Sprecheradaption durch Anpassung der bei der Vektorquantisierung verwendeten Codebücher geht davon aus, daß die nachgeordneten SCHMM das zeitliche Sprechverhalten eines jeden Sprechers ausreichend genau modellieren und daher im wesentlichen sprecherunabhängig sind. Aus dieser Annahme folgt, daß die vektorquantisierten Daten unabhängig von ihrer Abfolge stark mit dem jeweiligen Sprecher variieren und dadurch den Rückgang der erzielten Erkennungsraten bewirken. Damit ist ein spezielles Codebuch für jeden individuellen Sprecher denkbar, welches die Folge von Merkmalsvektoren eines unbekannten Sprechers auf eine Folge von Pseudorückschlußwahrscheinlichkeiten der Vektorquantisierung abbildet, wie sie ein Sprecher der Trainingsdatenbasis für die gleiche Äußerung erzeugt hätte. Dieses Codebuch für einen unbekannten Sprecher möglichst rasch zu finden, ist Ziel der nachfolgend beschriebenen Adaptionsalgorithmen.

Jeder Zustand eines SCHMM enthält einen kompletten Satz von M Mixtur-Koeffizienten, welche durch Gewichtung der im Codebuch enthaltenen multivariaten Gaußverteilungen eine Wahrscheinlichkeitsdichtefunktion im gesamten Merkmalsraum definieren. Durch Veränderung der Parameter des Codebuchs werden die Form und die Position der einzelnen Gaußglocken beeinflußt, nicht

aber ihre 'Bedeutung' (Mixtur-Koeffizienten) in den einzelnen Zuständen der SCHMM.

Die nachfolgend beschriebenen Algorithmen beschränken sich alle darauf, die *Prototypen*, also die Mittelwertsvektoren der Gaußglocken im Codebuch, aber nicht deren Kovarianzmatrizen zu manipulieren. Die Gründe hierfür sind zum einen, daß für die Neuabschätzung der Varianzen sehr viel mehr Daten notwendig sind, zum anderen, daß die Form der multivariaten Gaußglocken in initialen Codebuch bei der Neupositionierung der Prototypen berücksichtigt werden kann. Eine Sprecheradaption der Kovarianzen im Codebuch wurde z.B. in [Hua91] untersucht. Es ergab sich keinerlei Gewinn gegenüber der alleinigen Adaption der Prototypen.

Der hier verfolgte Grundgedanke ist folgender: aus der Beobachtung einiger weniger, gesprochener Wörter des neuen Sprechers muß genügend generalisierbare Information gewonnen werden, so daß das vorhandene Codebuch an den neuen Sprecher angepaßt werden kann. Das ist deshalb möglich, weil die Repräsentation des Sprachsignals in Codebuch-Symbolen noch sehr nahe an den eigentlichen Merkmalsvektoren liegt und damit im hohen Maße als generalisierbar gelten kann. Aus Untersuchungen über die Stationarität von einzelnen Lauten ([Str91]) ist bekannt, daß sich Laute durch stationäre Folgen bestimmter Codebuch-Symbole segmentieren lassen. Gelingt daher eine Veränderung des Codebuchs derart, daß die Merkmalsvektorfolge des unbekannten Sprechers auf eine Codebuch-Symbolfolge, die das betreffende HMM aufgrund seines Trainings mit mehreren Sprechern erwartet, abgebildet wird, so ist eine Verbesserung der Erkennungsrate zu erwarten.

Es sind zwei grundsätzliche Strategien denkbar: *Mittelwertsbildende* und *vergessende Strategie*.

Bei der mittelwertsbildenden Strategie werden laufend Beobachtungen gemacht, gemittelt und zum vorhandenen Codebuch in ein bestimmtes Verhältnis gesetzt. Dabei sind nur zwei Möglichkeiten gegeben: entweder 'vergißt' der Algorithmus die Information des initialen Codebuchs sofort (in der praktischen Anwendung nicht möglich) oder nie (der Algorithmus wird unflexibel).

Die vergessende Strategie dagegen, wie sie hier im wesentlichen verfolgt wird, ist in der Lage, je nach Umfang der erlangten Information, sowohl das initiale Codebuch als auch die bereits gelernten Daten je nach Bedarf geringer zu gewichten.

Nach einer Manipulation am Codebuch ist natürlich dessen Optimalität im Sinne der Informationstheorie (minimaler mittlerer Fehler bei der Quantisierung) nicht mehr gegeben. Dies ist aber weniger tragisch, als es im ersten Moment klingt, da hier keine Rekonstruktion des Signals, sondern die Extraktion des semantischen Gehalts an erster Stelle steht. Wenn letzteres mit Hilfe eines im Sinne der Informationstheorie 'schlechten' Codebuchs besser möglich ist, sollte dieser Einwand nicht weiter stören.

3.2 Übersicht der Versuche

3.2.1 Varianten

Folgende generelle Varianten sind für alle Versuche möglich:

- *Unüberwachte* oder *halbüberwachte* Adaption (UNUEB,HAUEB).
  Unüberwacht bedeutet, daß vom Benutzer keinerlei Rückmeldung erfolgt. Das System erhält seine Informationen allein aus dem vorhandenem Wissen innerhalb des Codebuchs und der HMM. In dieser Untersuchung geht das System immer 'optimistisch' vor, d.h. es nimmt immer an, korrekt gearbeitet zu haben.
  Halbüberwacht bedeutet (i.G. zu überwacht), daß vom Benutzer bei Mißerfolg eine Warnung erfolgt und das richtige Wort aus B ermittelten Alternativen ausgewählt wird (B ist in allen Versuchen gleich 5).

- *konstante* oder *dynamische* Adaptionsstärke.
  Die konstante Adaptionsstärke wird einmal empirisch ermittelt und bleibt während des gesamten Betriebs des Systems gleich.
  Die dynamische Adaptionsstärke richtet sich nach den aktuellen, bzw. in einem bestimmten Zeitfenster zurückliegenden Ereignissen im System, z.B. der Entwicklung der Erzeugungswahrscheinlichkeiten in den x zurückliegenden, erkannten Wörtern, der Zunahme der Ablehnungen durch den Benutzer, etc.

Des weiteren werden zwei grundsätzliche Varianten der Informationsgewinnung untersucht:

- Information nur aus der Vektorquantisierung selber (NOVITER).
  Der Adaption stehen in jedem Zeitschritt t der Merkmalsvektor $\underline{k}$ sowie die Pseudorückschlußwahrscheinlichkeiten $\underline{p}_k(t)$ zu allen M Prototypen zur Verfügung.

- Information zusätzlich aus dem Viterbipfad des SCHMM (VITER).
  Außer den o.g. Daten ist in jedem Zeitschritt t der *Mixturvektor* $\underline{q}_k$ bekannt. Dieser wird durch Rückverfolgung des Viterbi-Pfades im selektierten SCHMM den korrespondierenden Merkmalsvektoren zugeordnet.

Es werden sechs verschiedene Strategien der Adaption des Codebuchs untersucht. Außer dem Algorithmus LMITT sind dies alles vergessende Strategien:

- Zuordnungen von Merkmalsvektoren des unbekannten Sprechers zu jedem Prototypen sammeln, und diese im Codebuch verschieben (VERSCH).

- Zuordnungen von Merkmalsvektoren des unbekannten Sprechers zu bestimmtem Prototypen sammeln, und diese im Codebuch unter Berücksichtigung der Umgebung im Codebuch verschieben. Die innere Struktur des Codebuchs soll möglichst erhalten bleiben (*Schwamm-Prinzip*, SCHWAM).

- Feststellung von SOLL- und IST-Prototypen zu Merkmalsvektoren des unbekannten Sprechers. Verschiebung dieser SOLL- und IST-Prototypen jeweils konträr, so daß die Merkmalsvektoren nun 'korrekt' quantisiert würden (ISTSOLL) [1].

- Zum Vergleich mit den drei o.g. Strategien Versuche mit der aus der Literatur bekannten *LVQ1* (*Learning Vector Quantization*) von T. Kohonen.

[1] Ein ähnlicher Algorithmus wurde in [Gev91] zur Schätzung von Prototypen eines Nächsten-Nachbar-Klassifikators vorgeschlagen

Strategien		NOVITER	VITER
VERSCH	UNUEB	1	2
	HAUEB	-	3
SCHWAM	UNUEB	4	5
	HAUEB	-	6
ISTSOLL	UNUEB	-	7
	HAUEB	-	8
LVQ1	HAUEB	-	9
LVQ2	HAUEB	-	10
LMITT	HAUEB	-	11

Tabelle 3.1: Übersicht der Versuche zur Adaption von semi-kontinuierlichen Codebüchern

Ist der IST-Prototyp gleich dem SOLL-Prototyp, so wird dieser zu dem beobachteten Merkmalsvektor hin verschoben. Andernfalls wird der IST-Prototyp vom Merkmalsvektor weggeschoben (*LVQ1*).

- Die erweiterte Methode *LVQ2* von T. Kohonen. Zusätzlich zur *LVQ1* müssen bestimmte Nebenbedingungen erfüllt sein (*LVQ2*).
- Bildung von Langzeitmittelwerten aus den beobachteten Daten und Interpolation mit den vorhandenen Prototypen (LMITT).

3.2.2 Nummerierung der Versuche

Tab. 3.1 zeigt eine Übersicht der vorgenommenen Versuche und deren Nummerierung in Abhängigkeit der Versuchsbedingungen und Strategien.

Bemerkungen:

- Unter der Bedingung NOVITER haben die Strategien ISTSOLL, *LVQ1*, *LVQ2* keinen Sinn, da keine SOLL-Prototypen bestimmt werden können.
- Unter der Bedingung NOVITER hat eine halbüberwachte Adaption (HAUEB) keinen Sinn, da der Viterbi-Pfad gar nicht ausgewertet wird.
- Da die Versuche mit *LVQ1*, *LVQ2* und LMITT nur zum Vergleich mit den neu entworfenen Strategien durchgeführt werden, wird auf die Variante UNUEB verzichtet.

3.3 Vorversuche

Die Pseudorückschlußwahrscheinlichkeiten $p_k(t, m)$ der semikontinuierlichen Vektorquantisierung berechnen sich durch Inversion der gewichteten, Euklidischen Abstände zu den Prototypen und anschließende Normierung aller Werte auf Summe 1 (Variante 1 der semikontinuierlichen Vektorquantisierung, siehe Abschnitt 2.2.4). Dies führt zu relativ breiten Verteilungen innerhalb der Mixturvektoren und erzielt beim sprecherunabhängigen Test die besten Erkennungsraten.

3.3.1 Generelle Vorversuche

Qualitative Vorversuche mit kleinen Datenmengen ergaben folgende Ergebnisse:

- Jeder der drei Merkmalsvektoren (sh, di, la) trägt zur Adaption bei. Daher wird im folgenden die Verarbeitung aller drei Merkmalsvektoren adaptiert.
- Die optimale konstante Adaptionsstärke variiert bei den Strategien VERSCH und SCHWAM.
- Die Adaption erzielt bei bestimmter Adaptionsstärke schon nach wenigen Wörtern (ca. 10 - 20) sehr gute Verbesserungen, dann treten jedoch meistens Verschlechterungen auf. Dies ist kein Zufallsprodukt der gewählten Reihenfolge der Adaptionswörter, wie der Kreuzversuch mit zufallsverteilten Adaptionswörtern zeigt.
- Die Reihenfolge der Adaptionswörter kann jedoch ausschlaggebend für die Geschwindigkeit der Adaption sein. Ein Kreuzversuch mit inverser alphabetischer Liste ergab zu Beginn einen sehr mäßigen Anstieg, endet aber in einem noch besseren Ergebnis als die alphabetische Liste. Es werden daher zukünftig Listen mit zufällig angeordneten Wörtern verwendet.

3.3.2 Adaptionsstärke

Alle nachfolgend untersuchten Verfahren verwenden einen Parameter, die sog. *Adaptionsstärke* $g(w)$, welche ein globales Maß für die Änderung der Codebuch-Parameter in jedem Adaptionsschritt (Wort w) darstellt. Der optimale Wert von $g(w)$ ist analytisch nicht bestimmbar, weshalb er z.B. durch Variation in mehreren Versuchen empirisch ermittelt wird.

Ein solcher konstanter Wert für $g(w) = A_k$ ist jedoch für die Sprecheradaption nicht optimal. Besser wäre eine dynamische Einstellung, welche den Wert von $g(w)$ erhöht, wenn Bedarf nach starker Adaption herrscht, und erniedrigt, wenn Konvergenz erwünscht ist.

$$g(w) = A_k + A_d\, f(\ldots) \tag{3.1}$$

Nach obigem Ansatz setzt sich $g(w)$ aus einem konstanten Anteil A_k und einem dynamischen Anteil zusammen. A_d ist ein ebenfalls konstanter Faktor, $f(\ldots)$ eine Funktion von internen Systemparametern, welche mit dem Bedarf nach stärkerer Adaption korreliert.

Der Versuch, die Dynamik der Adaptionsstärke aus der mittleren Erfolgsrate des ASE Systems zu ermitteln, scheitert einfach an der Tatsache, daß dies viel zu lange dauert.

Ein weiterer Ansatz ist die Entwicklung der mittleren Erzeugungswahrscheinlichkeit (EWK) in den SCHMM. Der dynamische Anteil der Adaptionsstärke richtet sich dabei nach der mittleren, von der Framezahl unabhängigen EWK in den besten B SCHMM. Beim Absinken dieser Größe wird die Adaptionsstärke erhöht und vice versa. Die Motivation hierfür ist die Annahme, daß ein Sprecherwechsel durch ein Absinken der EWKen angezeigt wird. Und nach einem Sprecherwechsel besteht Bedarf nach stärkerer Adaption (vgl. auch Abschnitt 3.6.1).

Dem ASE System wird eine Folge von Wörtern w_i angeboten. Jedes Wort w_i enthält $F(w_i)$ Frames. Das System bestimmt die EWK für die B besten Modelle.

Dann ist die mittlere, logarithmierte und von der Framezahl unabhängige EWK dieser Modelle

$$\begin{aligned}\overline{log(E_U(w_i))} &= \frac{1}{B}\sum_{h=1}^{B} log(E_U(w_i,h)) = \\ &= \frac{1}{B\ F(w_i)}\sum_{h=1}^{B} log(E(w_i,h)) \end{aligned} \tag{3.2}$$

$E(w_i, h)$ sei dabei die EWK des SCHMM h auf die Daten des Wortes w_i berechnet nach Viterbi (vgl. Anhang C).

Die Differenz zwischen dem aktuellen Wort w_i und dem Mittelwert der L vorangegangenen Wörter $w_{i-L} \ldots w_{i-1}$ ist

$$\Delta\overline{log(E_U(w_i))} = \frac{1}{L}\sum_{l=1}^{L} \overline{log(E_U(w_{i-l}))} - \overline{log(E_U(w_i))} \tag{3.3}$$

Die Adaptionsstärke $g(w_i)$ für das Wort w_i berechnet sich daraus zu

$$g(w_i) = \begin{cases} A_k + A_d\ \Delta\overline{log(E_U(w_i))} & : \quad \Delta\overline{log(E_U(w_i))} > 0 \\ A_k & : \quad \text{sonst} \end{cases} \tag{3.4}$$

Zusätzlich wurden noch folgende interne Parameter des Systems untersucht: Framezahl des Adaptionswortes, Änderung der EWK des Adaptionswortes vor und nach Adaption, Änderung der auf Framezahl normierten EWK vor und nach Adaption und EWK selber.

Leider ergibt eine statistische Analyse aller obigen Parameter folgendes:
Es ist keinerlei Korrelation zwischen dem Erfolg eines Adaptionsschrittes (Test mit unabhängiger Stichprobe) und allen oben beschriebenen Parametern feststellbar. Zusätzlich wurde der Erfolg der Adaption in zwei Versuchen mit unterschiedlicher Reihenfolge der Adaptionswörter untersucht. Es ergeben sich keinerlei Korrelationen bei den gleichen Wörtern.

Das bedeutet: ob ein Wort positiv zur Adaption beiträgt oder nicht, ist mit den untersuchten, internen Parametern allein nicht vorhersehbar, sondern hängt wahrscheinlich vom Gesamtprozeß der Adaption (z.B. innerer Zustand des Codebuchs, etc.) ab.

Zwei Arten der Adaption von Codebüchern scheinen daher generell möglich:
Erstens die sichere, aber langsame Adaption mit kleiner konstanter Adaptionsstärke.
Zweitens die rasche, kurzfristige Adaption mit hoher Adaptionsstärke. Letztere muß jedoch rechtzeitig (nach ca. 10 - 14 Wörtern) beendet werden, da sich sonst die Erkennungsrate infolge von Überadaption wieder deutlich verschlechtert.

3.4 Beschreibung der Versuche

Sprachmaterial:	Sprecher TM BR
Sprachstichprobe für Beurteilung:	*Test1*
Sprachstichprobe für Adaption:	*Adaption*

Adaption enthält die Sprachstichprobe des unbekannten Sprechers, anhand derer das System adaptieren soll. *Test1* enthält die Wörter, anhand derer die Erkennungsleistung des Systems nach einem oder mehreren Adaptionsschritt(en)

beurteilt wird. Die beiden Wortmengen sind disjunkt, d.h. nicht die Adaptionsfähigkeit an eine bestimmte Stichprobe sondern an die Sprache des unbekannten Sprechers wird beurteilt. Das Lexikon enthält genau die Vereinigungsmenge der beiden Wortlisten.

Äußerer Versuchsablauf für alle Versuche:

1. Die Erkennungsrate mit unadaptierten Codebüchern wird ermittelt.
2. Das nächste Wort aus *Adaption* wird vom ASE System verarbeitet.
3. Codebücher werden adaptiert.
4. Alle Wörter aus *Test1* werden verarbeitet, um die Adaption zu beurteilen.
5. Weiter bei 1.), bis kein Wort mehr in *Adaption*.

Die Versuche werden jeweils in zwei Varianten durchgeführt: Adaption mit schwacher Adaptionsstärke über längeren Zeitraum (100 Wörter) und Adaption mit starker Adaptionsstärke über kurzen Zeitraum (20 Wörter). Aus letzteren Versuchen läßt sich auch die optimale Dauer der starken Adaptionsphase abschätzen.

3.4.1 Versuch 1 — VERSCH,NOVITER,UNUEB

1. In jedem Zeitschritt t sind die unbekannten Merkmalsvektoren $\underline{sh}(t)$, $\underline{di}(t)$ und $\underline{la}(t)$ sowie deren Pseudorückschlußwahrscheinlichkeiten auf die M jeweiligen Prototypen $\underline{p}_{sh}(t)$, $\underline{p}_{di}(t)$, und $\underline{p}_{la}(t)$ bekannt. Der Merkmalsvektor sei im folgenden mit k abgekürzt.

2. Zu demjenigen Prototypen $\underline{s}_k(m_{max})$ mit

$$p_k(t, m_{max}) = \max_{m=1\ldots M}(p_k(t, m)) \tag{3.5}$$

wird der Differenzvektor zwischen diesem und dem Merkmalsvektor $\underline{k}(t)$ berechnet.

$$\begin{aligned} \Delta\underline{k}(t, m) &= [\underline{s}_k(m) - \underline{k}(t)]\, u(\underline{s}_k(m), \underline{k}(t)) \quad \text{für} \quad m = m_{max} \\ \Delta\underline{k}(t, m) &= \underline{0} \quad \text{sonst} \end{aligned} \tag{3.6}$$

$u(\underline{s}_k(m), \underline{k}(t))$ ist dabei eine Berücksichtigung der jeweiligen Varianzen des Prototypen $\underline{s}_k(m)$ und ist somit von der Art des in der Vektorquantisierung verwendeten Abstandsmaßes abhängig. Im Vorversuchen wurde der gewichtete Euklidische Abstand d_G als geeignetes Abstandsmaß ausgewählt.

$$d_G(\underline{s}_k(m), \underline{k}(t), \underline{D}_k(m)) = (\underline{k}(t) - \underline{s}_k(m))' \underline{D}_k(m)(\underline{k}(t) - \underline{s}_k(m)) \tag{3.7}$$

wobei $\underline{D}_k(m)$ eine Matrix mit den Kehrwerten der Varianzen auf den Komponenten der Vektoren im Cluster m in der Diagonale ist:

$$\underline{D}_k(m) = \begin{cases} [1/\sigma_{ij}] & : \quad i = j \\ 0 & : \quad i \neq j \end{cases} \tag{3.8}$$

Abbildung 3.1: Berücksichtigung der Varianzen der Cluster

Entsprechend läßt sich eine Matrix $\overline{\underline{D}}_k(m)$ definieren, deren Hauptdiagonalelemente alle gleich dem arithmetischen Mittelwert der inversen Streuungen auf den Komponenten sind:

$$\overline{\underline{D}}_k(m) = \begin{cases} [1/\sigma_{ij}] = [1/\overline{\sigma}] & : \quad i = j \\ 0 & : \quad i \neq j \end{cases} \tag{3.9}$$

mit

$$1/\overline{\sigma} = 1/N \sum_{i=1}^{N} 1/\sigma_{ii} \tag{3.10}$$

Dann ergibt sich $u(\underline{s}_k(m), \underline{k}(t))$ zu

$$u(\underline{s}_k(m), \underline{k}(t)) = \frac{d_G(\underline{s}_k(m), \underline{k}(t), \underline{D}_k(m))}{d_G(\underline{s}_k(m), \underline{k}(t), \overline{\underline{D}}_k(m))} \tag{3.11}$$

D.h. der Betrag des Differenzvektors $\Delta\underline{k}(t, m)$ wird um so größer, je größer der Abstand von $\underline{k}(t)$ zum Prototypen $\underline{s}_k(m)$ unter Berücksichtigung der Streuungen der Komponenten im Cluster m gegenüber dem Abstand bei Berücksichtigung einer nur kugelförmigen Verteilung ist (s. Abb. 3.1). Eine andere Alternative wäre es, den gewichteten Euklidischen Abstand direkt als Gewichtung zu verwenden. Dabei kann es jedoch zu Normierungsproblemen kommen. Deshalb wurde der einfachere Weg gewählt und auf den mittleren gewichteten Euklidischen Abstand normiert.

3. Die Differenzvektoren werden über alle T Zeitschritte des Wortes für jeden Prototypen m rekursiv gemittelt.

$$\overline{\Delta\underline{k}}(0, m) \quad = \quad \underline{0}$$

Abbildung 3.2: Strategie VERSCH für einen Zeitschritt (a), für ein Wort mit 6 Zeitschritten (b)

$$\overline{\Delta \underline{k}}(t,m) \quad = \quad (1-\frac{1}{t})\ \overline{\Delta \underline{k}}(t-1,m) + \frac{1}{t}\ \Delta \underline{k}(t,m) \tag{3.12}$$

4. Die resultierenden mittleren Differenzvektoren $\overline{\Delta \underline{k}}(t,m)$ werden mit der Adaptionsstärke $g(w)$ gewichtet von den Prototypen $\underline{s}_k(m)$ subtrahiert. Fand keine Zuordnung eines Merkmalsvektors auf einen Prototypen statt, ergibt sich für den Differenzvektor der Null-Vektor, d.h. der Prototyp wird nicht verschoben.

$$\begin{aligned} \underline{s}'_k(m) &= \underline{s}_k(m) - g(w)\overline{\Delta \underline{k}}(T,m) \\ g(w) &= A_k \quad \text{(vgl. dazu 3.3.2)} \end{aligned} \tag{3.13}$$

   Prototypen werden also immer in Richtung des nächstliegenden Merkmalsvektor verschoben. Durch die rekursive Mittelung heben sich statistische gleichverteilte Verschiebungen auf, die globale Tendenz der Beobachtungen wird aber im Laufe der Zeit den Prototypen verlagern.

Dieses 'blinde' Adaptieren setzt voraus, daß ein überwiegender Teil der Vektorquantisierung auf 'korrekte' Weise geschieht, d.h. daß im Mittel mehr Merkmalsvektoren auf Prototypen fallen, die zu einer richtigen Erkennung des nachfolgenden HMMs führen (s. Abb. 3.2).

3.4.2 Versuch 2 — VITER,VERSCH,UNUEB

1. In jedem Zeitschritt t sind die unbekannten Merkmalsvektoren $\underline{sh}(t)$, $\underline{di}(t)$ und $\underline{la}(t)$, sowie deren Pseudo-Rückschlußwahrscheinlichkeiten auf die M

jeweiligen Prototypen $\underline{p}_{sh}(t)$, $\underline{p}_{di}(t)$, und $\underline{p}_{la}(t)$ bekannt. Außerdem ist die Folge der auf dem Viterbi-Pfad durchlaufenen Zustände im Modell mit der höchsten Erzeugungswahrscheinlichkeit aus der vorangegangenen Erkennung durch Backtracking bestimmbar. Dadurch kann für jeden Zeitschritt t eine Verteilung der Mixtur-Koeffizienten über die drei Codebücher $\underline{q}_{sh}(t)$, $\underline{q}_{di}(t)$, und $\underline{q}_{la}(t)$ angegeben werden. Der Merkmalsvektor sei im folgenden mit k abgekürzt.

2. Zu demjenigen Prototypen $\underline{s}_k(m_{max})$ mit

$$p_{kq}(t, m_{max}) = \max_{m=1\ldots M}(p_k(t,m)\; q_k(t,m)) \tag{3.14}$$

wird der Differenzvektor zwischen diesem und dem Merkmalsvektor $\underline{k}(t)$ berechnet.

$$\begin{aligned} \Delta\underline{k}(t,m) &= [\underline{s}_k(m) - \underline{k}(t)]\; u(\underline{s}_k(m), \underline{k}(t)) \quad \text{für} \quad m = m_{max} \\ \Delta\underline{k}(t,m) &= \underline{0} \quad \text{sonst} \end{aligned} \tag{3.15}$$

Zur Herleitung von $u(\underline{s}_k(m), \underline{k}(t))$ siehe Versuch 1.

Die übrigen Punkte entsprechen Versuch 1.

Dies ist im Prinzip die gleiche Strategie wie unter Versuch 1, jedoch wird hier nur der Prototyp nachadaptiert, der im jeweiligen Zustand den wesentlichen Beitrag hatte. Dies muß nicht der nächstliegende Prototyp sein, sondern ist von den Mixtur-Koeffizienten des Zustands auf dem Viterbi-Pfad abhängig (s. Abb. 3.3). Indem das Maximum über das Produkt von Pseudorückschlußwahrscheinlichkeit und Mixtur-Koeffizient gebildet wird, soll vermieden werden, daß Prototypen, die sehr weit entfernt vom beobachteten Merkmalsvektor liegen, zur Adaption verwendet werden.

3.4.3 Versuch 3 — VITER,VERSCH,HAUEB

Im Gegensatz zu Versuch 2 erfolgt hier die Adaption halbüberwacht, d.h. der Benutzer informiert bei Mißerfolg das System über das tatsächlich gesprochene Wort, sofern es sich unter den besten B Wörtern befindet. Befindet es sich nicht darunter, wird nicht adaptiert. Dadurch ist die Folge der auf dem Viterbi-Pfad durchlaufenen Zustände im Modell des tatsächlich gesprochenen Wortes und damit in jedem Zeitschritt t die Verteilung der Mixtur-Koeffizienten über die drei Codebücher $\underline{q}_{sh}(t)$, $\underline{q}_{di}(t)$, und $\underline{q}_{la}(t)$ bestimmbar. Die übrigen Punkte entsprechen Versuch 2.

Anders als in Versuch 2 wird dem System mitgeteilt, welches HMM die höchste Erzeugungswahrscheinlichkeit haben *sollte*. Durch die Adaption der Prototypen soll diese für das eben gesprochene Wort erhöht werden, in der Hoffnung, daß dadurch die Prototypen generell besser zwischen dem unbekannten Sprecher und den fest trainierten HMM vermitteln. Ist das tatsächlich gesprochene Wort nicht unter den ersten B Ergebnissen, war die Äußerung so schlecht, daß sich eine Adaption nicht lohnt, voraussichtlich sogar zu einer Verschlechterung führt.

Abbildung 3.3: Strategie VERSCH mit Berücksichtigung des Viterbi-Pfades

3.4.4 Ergebnisse der Versuche 1,2,3

Kurze Adaptionsphase (20 Wörter)

In Vorversuchen mit variierter Adaptionsstärke wird der optimale Wert für A_k ermittelt: $A_k = 0.15$.

Abb. 3.4 zeigt einen raschen Anstieg der Erkennungsrate nach wenigen Wörtern mit Strategie VERSCH. Deutlich besser erscheinen die Versuche 2 und 3, welche die Information des Viterbi-Pfades nutzen. Eine rasche Adaptionsphase mit den Versuchsbedingungen von 3 sollte nach ca. 13 Wörtern beendet werden, bevor Überadaption einsetzt.

Lange Adaptionsphase (100 Wörter)

In Vorversuchen mit variierter Adaptionsstärke wird der optimale Wert für A_k ermittelt: $A_k = 0.10$.

Abb. 3.5 zeigt zunächst einen langsamen Anstieg der Erkennungsrate für alle 3 Versuche in Strategie VERSCH. Ab ca. 40 Adaptionswörtern wird jedoch die Strategie ohne Berücksichtigung des Viterbi-Pfades instabil und verschlechtert sich im weiteren Verlauf sogar. Versuch 2 und 3 dagegen steigen weiter an, führen jedoch zu starken Schwankungen in der Erkennungsrate.

Abbildung 3.4: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 1, 2 und 3, $A_k = 0.15$

Abbildung 3.5: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 1, 2 und 3, $A_k = 0.10$

Abbildung 3.6: Strategie VERSCH (a) und SCHWAM (b) in einem Zeitschritt

3.4.5 Versuch 4 — NOVITER,SCHWAM,UNUEB

Anders als in Versuch 1, Punkt 2 wird in diesem Versuch zu jedem Prototypen $\underline{s}_k(m)$, $m = 1 \ldots M$ ein Differenzvektor berechnet

$$\Delta\underline{k}(t,m) = \frac{p_k(t,m)}{p_k(t,m_{max})} \quad [\underline{s}_k(m_{max}) - \underline{k}(t)] \quad u(\underline{s}_k(m_{max}), \underline{k}(t)) \tag{3.16}$$

Die übrigen Punkte entsprechen denen von Versuch 1.

Im Gegensatz zur Strategie VERSCH (Versuch 1) wird hier die Umgebung des Codebuchs mit berücksichtigt. Fällt z.B. ein Merkmalsvektor $\underline{k}$ etwa in die Mitte von drei Prototypen des Codebuchs m_1, m_2 und m_3, und m_1 ist der nächstliegende, so wird m_1 in der Strategie VERSCH in Richtung $\underline{k}$ verschoben und nähert sich dadurch eventuell zu sehr m_2 oder m_3. Bei der Strategie SCHWAM werden jedoch m_2 und m_3 in der *gleichen* Richtung nur mit etwas niedrigerem Betrag wie m_1 verschoben (s. Abb. 3.6). Die innere, lokale Struktur des Codebuchs bleibt dadurch besser erhalten, wenn auch nur nahe $\underline{k}$. Weit entfernte Prototypen bleiben von der Verschiebung ev. ganz ausgeschlossen, weil ihre entsprechenden Pseudorückschlußwahrscheinlichkeiten zu klein sind.

3.4.6 Versuch 5 — VITER,SCHWAM,UNUEB

Dieser Versuch ist eine Kombination der Versuche 2 und 4. Der Übersichtlichkeit halber seien die ersten Rechenschritte hier vollständig wiedergegeben.

1. In jedem Zeitschritt t sind die unbekannten Merkmalsvektoren $\underline{sh}(t)$ (sh), $\underline{di}(t)$ (di) und $\underline{la}(t)$ (la), sowie deren Pseudo-Rückschlußwahrscheinlichkeiten auf die M jeweiligen Prototypen $\underline{p}_{sh}(t)$, $\underline{p}_{di}(t)$, und $\underline{p}_{la}(t)$ bekannt.

Außerdem ist die Folge der auf dem Viterbi-Pfad durchlaufenen Zustände im Modell mit der höchsten Erzeugungswahrscheinlichkeit aus der vorangegangenen Erkennung durch Backtracking bestimmbar. Dadurch kann für jeden Zeitschritt t eine Verteilung der Mixtur-Koeffizienten über die drei Codebücher $\underline{q}_{sh}(t)$, $\underline{q}_{di}(t)$, und $\underline{q}_{la}(t)$ angegeben werden. Der Merkmalsvektor sei im folgenden mit k abgekürzt.

2. Zu demjenigen Prototypen $\underline{s}_k(m_{max})$ mit

$$p_{kq}(t, m_{max}) = \max_{m=1\ldots M}(p_k(t, m)\ q_k(t, m)) \tag{3.17}$$

wird der Differenzvektor zwischen diesem und dem Merkmalsvektor $\underline{k}(t)$, und aus diesem die gewichteten Differenzvektoren für alle Prototypen berechnet.

$$\Delta\underline{k}(t, m) = \frac{p_k(t, m)}{p_k(t, m_{max})} \quad [\underline{s}_k(m_{max}) - \underline{k}(t)] \quad u(\underline{s}_k(m_{max}), \underline{k}(t)) \tag{3.18}$$

Zur Herleitung von $u(\underline{s}_k(m), \underline{k}(t))$ siehe Versuch 1.

Die übrigen Punkte (3 und 4) entsprechen denen von Versuch 1.

3.4.7 Versuch 6 — VITER,SCHWAM,HAUEB

Dieses Experiment entspricht Versuch 5 jedoch mit halbüberwachter Adaption. Siehe sinngemäß Punkt 1 in Versuch 3.

3.4.8 Ergebnisse der Versuche 4,5,6

Kurze Adaptionsphase (20 Wörter)

In Vorversuchen mit variierter Adaptionsstärke wird der optimale Wert für A_k ermittelt: $A_k = 0.50$.

Abb. 3.7 zeigt einen raschen Anstieg der Erkennungsrate nach wenigen Wörtern in den Versuchen mit Auswertung des Viterbi-Pfades (5 und 6). Dagegen erbringt Versuch 4 nur eine Verbesserung von ca. 2.5 %. Eine rasche Adaptionsphase mit den Versuchsbedingungen von 6 sollte nach ca. 16 Wörtern beendet werden, bevor Überadaption einsetzt.

Lange Adaptionsphase (100 Wörter)

In Vorversuchen mit variierter Adaptionsstärke wird der optimale Wert für A_k ermittelt: $A_k = 0.05$.

Abb. 3.8 zeigt ebenfalls in Versuch 4 nur eine Verbesserung von ca 1 %. Überraschend gut entwickeln sich die Erkennungsraten in den Versuchen 5 und 6, welche sich bei einer Verbesserung von 5.0 % bzw. 5.5 % stabilisieren und auch eine kleinere Schwankungsbreite zeigen als in Strategie VERSCH.

3.4.9 Versuch 7 — VITER,ISTSOLL,UNUEB

1. In jedem Zeitschritt t sind die unbekannten Merkmalsvektoren $\underline{sh}(t)$, $\underline{di}(t)$ und $\underline{la}(t)$, sowie deren Pseudorückschlußwahrscheinlichkeiten auf die M jeweiligen Prototypen $\underline{p}_{sh}(t)$, $\underline{p}_{di}(t)$, und $\underline{p}_{la}(t)$ bekannt. Außerdem ist die

Abbildung 3.7: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 4, 5 und 6, $A_k = 0.50$

Abbildung 3.8: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 4, 5 und 6, $A_k = 0.05$

Folge der auf dem Viterbi-Pfad durchlaufenen Zustände im Modell mit der höchsten Erzeugungswahrscheinlichkeit aus der vorangegangenen Erkennung durch Backtracking bestimmbar. Dadurch kann für jeden Zeitschritt t eine Verteilung der Mixtur-Koeffizienten über die drei Codebücher $\underline{q}_{sh}(t)$, $\underline{q}_{di}(t)$, und $\underline{q}_{la}(t)$ angegeben werden. Der Merkmalsvektor sei im folgenden mit k abgekürzt.

2. Zu demjenigen Prototypen $\underline{s}_k(m_{soll})$ (SOLL-Prototyp) mit

$$p_k(t, m_{soll}) = \max_{m=1...M}(p_k(t, m)\ q_k(t, m)) \tag{3.19}$$

wird der Differenzvektor zwischen diesem und dem Merkmalsvektor $\underline{k}(t)$ berechnet.

$$\Delta\underline{k}'(t, m_{soll}) \quad = \quad \underline{s}_k(m_{soll}) - \underline{k}(t) \tag{3.20}$$

und zu dem Prototypen $\underline{s}_k(m_{ist})$ (IST-Prototyp) mit

$$p_k(t, m_{ist}) = \max_{m=1...M}(p_k(t, m)) \tag{3.21}$$

der Differenzvektor zwischen diesem und dem Merkmalsvektor $\underline{k}(t)$:

$$\Delta\underline{k}'(t, m_{ist}) \quad = \quad \underline{s}_k(m_{ist}) - \underline{k}(t) \tag{3.22}$$

Für SOLL- und IST-Prototypen wird der Differenzvektor so gewichtet, daß beide um den gleichen Betrag d im Raum verschoben werden und der Merkmalsvektor $\underline{k}(t)$ um den Faktor Δ_{IS} näher am SOLL-Prototypen liegt als am IST-Prototypen. Ist Δ_{IS} gleich eins, so ergeben sich nach der Verschiebung gleiche Abstände von IST- und SOLL-Prototypen zum Merkmalsvektor $\underline{k}(t)$.

$$\begin{aligned} \Delta\underline{k}(t, m_{soll}) &= d\ \frac{\Delta\underline{k}'(t, m_{soll})}{\mid \Delta\underline{k}'(t, m_{soll}) \mid} \\ \Delta\underline{k}(t, m_{ist}) &= -d\ \frac{\Delta\underline{k}'(t, m_{ist})}{\mid \Delta\underline{k}'(t, m_{ist}) \mid} \end{aligned} \tag{3.23}$$

d berechnet sich dabei zu

$$d = \frac{1}{1 + \Delta_{IS}}\ (\Delta_{IS} \mid \Delta\underline{k}'(t, m_{soll}) \mid - \mid \Delta\underline{k}'(t, m_{ist}) \mid) \tag{3.24}$$

Die übrigen Punkte entsprechen den Punkten 3 und 4 in Versuch 1. Aus jeder Beobachtung von SOLL- und IST-Prototypen werden Verschiebungsvektoren so berechnet, daß bei einer erneuten Quantisierung auf den 'korrekten' Prototyp[2] quantisiert würde. Durch die Mittelung über alle Beobachtungen ergibt sich entweder eine Aufhebung statistisch nicht eindeutiger Verschiebungen oder eine Tendenz, mit deren Hilfe die Prototypen des Codebuchs adaptiert werden können (s. Abb. 3.9). Die Verschiebung erfolgt längs der Verbindung des jeweiligen Prototypen zum Merkmalsvektor um die Länge d. Der IST-Prototyp wird vom Merkmalsvektor weg-, der SOLL-Prototyp zu diesem hingeschoben.

[2]korrekt i.S. des Viterbi-Pfades

Abbildung 3.9: Strategie ISTSOLL - Verschiebung aus einem Zeitschritt

Herleitung der Größen d und Δ_{IS}:

Der Faktor Δ_{IS} soll ausdrücken, um wieviel näher der Merkmalsvektor $\underline{k}(t)$ nach der Adaption des Codebuchs am SOLL-Prototypen liegt als am IST-Prototypen, d.h.

$$\Delta_{IS} = \frac{\mid \underline{s}'(m_{ist}) - \underline{k}(t) \mid}{\mid \underline{s}'(m_{soll}) - \underline{k}(t) \mid} \tag{3.25}$$

Gestrichene Prototypen seien Prototypen nach der Adaption. Außerdem soll gelten, daß sowohl SOLL- als auch IST-Prototyp um den gleichen Betrag d verschoben werden, nur in entgegengesetzter Richtung.

$$\begin{aligned} \mid \underline{s}'(m_{soll}) - \underline{k}(t) \mid &= \mid \underline{s}(m_{soll}) - \underline{k}(t) \mid - d \\ \mid \underline{s}'(m_{ist}) - \underline{k}(t) \mid &= \mid \underline{s}(m_{ist}) - \underline{k}(t) \mid + d \end{aligned} \tag{3.26}$$

In Gl. 3.25 eingesetzt und nach d aufgelöst ergibt dies

$$d = \frac{1}{1 + \Delta_{IS}} \left(\Delta_{IS} \mid \underline{s}(m_{soll}) - \underline{k}(t) \mid - \mid \underline{s}(m_{ist}) - \underline{k}(t) \mid\right) \tag{3.27}$$

was gleichbedeutend ist mit

$$d = \frac{1}{1 + \Delta_{IS}} \left(\Delta_{IS} \mid \Delta\underline{k}'(t, m_{soll}) \mid - \mid \Delta\underline{k}'(t, m_{ist}) \mid\right) \tag{3.28}$$

Abbildung 3.10: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 7, 8 und 10, $\Delta_{IS} = 1.6$

3.4.10 Versuch 8 — VITER,ISTSOLL,HAUEB

Dieser Versuch entspricht Versuch 7, nur stammt die Verteilung der Mixtur-Koeffizienten aus dem Viterbi-Pfad durch das Modell des tatsächlich gesprochenen Wortes, statt des Modells mit der höchsten Erzeugungswahrscheinlichkeit. D.h. die Adaption erfolgt halbüberwacht (siehe sinngemäß Versuch 3).

3.4.11 Ergebnisse der Versuche 7 und 8

Kurze Adaptionsphase (20 Wörter)

In Vorversuchen mit variierter Adaptionsstärke wird der optimale Wert für Δ_{IS} ermittelt: $\Delta_{IS} = 1.6$.

Abb. 3.10 zeigt die Entwicklung der Erkennungsraten in den Versuchen 7 und 8. In beiden Versuchen zeigt sich eine starke Schwankung der Werte. Dadurch ist eine Abschätzung der Anzahl von Adaptionswörtern für eine kurze Adaptionsphase nicht möglich. Da in den entsprechenden Vorversuchen für praktisch alle Werte von $\Delta_{IS} = 0.2 \ldots 2.0$ nach 100 Adaptionswörtern keine konvergente Verbesserung der Erkennungsrate zu beobachten war, wurde auf den Versuch mit 100 Wörtern verzichtet. Die Ergebnisse geben Anlaß zu dem Verdacht, daß die Strategie ISTSOLL mit den gleichen Problemen zu kämpfen hat wie die bekannte *LVQ* von Kohonen ([Koh88]). Dort wurde beobachtet, daß sich die Ergebnisse verschlechtern, wenn die Auswahl der IST- und SOLL-Prototypen nicht strengeren Kriterien unterworfen werden (siehe auch Abschnitt 3.4.13).

Abbildung 3.11: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 9, $A_k = 0.15$ (durchgezogen), $A_k = 0.20$ (gestrichelt)

3.4.12 Versuch 9 — VITER,*LVQ1*,HAUEB

Analog zu Versuch 7 wird ein IST- und SOLL-Prototyp zu dem beobachteten Merkmalsvektor bestimmt und der Differenzvektor zum IST-Prototypen nach Gl. 3.22 $\Delta\underline{k}'(t, m_{ist})$ berechnet. Der resultierende Differenzvektor $\Delta\underline{k}(t, m_{ist})$ berechnet sich zu

$$\Delta\underline{k}(t, m_{ist}) = \begin{cases} \Delta\underline{k}'(t, m_{ist}) & : \quad m_{ist} = m_{soll} \\ -\Delta\underline{k}'(t, m_{ist}) & : \quad \text{sonst} \end{cases} \tag{3.29}$$

Die übrigen Punkte entsprechen den Punkten 3 und 4 in Versuch 1. Dies entspricht der *LVQ1* von T. Kohonen ([Koh88]). Wird ein Merkmalsvektor 'korrekt' quantisiert, so wird der betreffende Prototyp in Richtung dieses Merkmalsvektors verschoben, anderenfalls wird dieser in entgegengesetzter Richtung verschoben. Die Stärke der Verschiebung wird durch $g(w)$ bestimmt.

Kurze Adaptionsphase (20 Wörter)

Die *LVQ1* wurde unter den gleichen Versuchsbedingungen wie in Versuch 3 getestet. Abb. 3.11 zeigt den Verlauf der Erkennungsrate. Für beide Werte von A_k ergibt sich keine Verbesserung gegenüber der einfacheren Strategie VERSCH (vgl. Bild 3.4), wenn man davon absieht, daß die Überadaption etwas später einsetzt.

Abbildung 3.12: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern in Versuch 9, $A_k = 0.05$ (durchgezogen), $A_k = 0.10$ (gestrichelt)

Lange Adaptionsphase (100 Wörter)

Abb. 3.12 zeigt den Verlauf der Erkennungsrate über 100 Adaptionswörter mit *LVQ1*. Auch hier zeigen sich ungefähr die gleichen Ergebnisse wie mit der einfachen Strategie VERSCH (vgl. Bild 3.5).

3.4.13 Versuch 10 — VITER,*LVQ2*,HAUEB

Dieser Versuch entspricht im wesentlichen dem Versuch 8, Strategie ISTSOLL. Entsprechend von T. Kohonens *LVQ2* wird jedoch die Auswahl der IST-SOLL-Paare für die Adaption eingeschränkt. Es müssen folgende Bedingungen erfüllt sein, damit ein Prototyp adaptiert wird ([Koh88]):

- der IST-Prototyp ist nicht der SOLL-Prototyp.
- der am zweit-nächsten liegende IST-Prototyp ist der SOLL-Prototyp.
- der beobachtete Merkmalsvektor liegt innerhalb eines kleinen 'Fensters' zwischen IST- und SOLL-Prototyp.

Nur wenn alle genannten Bedingungen erfüllt sind, werden die ermittelten IST- und SOLL-Prototypen anhand des beobachteten Merkmalsvektors analog zur Strategie ISTSOLL (s.o.) nachadaptiert. Die dritte Bedingung wurde nicht berücksichtigt, da sonst nur noch in den seltensten Fällen überhaupt IST-SOLL-Pärchen detektiert werden.

Kurze Adaptionsphase (20 Wörter)

Das Ergebnis zeigt der gepunktete Graph in Abb. 3.10. Die Verschärfung der Auswahl nach *LVQ2* hat keinen positiven Effekt. Im Gegenteil sinkt sogar die maximal erreichbare Erkennungsrate gegenüber der Strategie ISTSOLL. Offensichtlich sind die harten Bedingungen der *LVQ2* besser für die Verfeinerung eines vorhandenen Codebuchs mit vielen Daten geeignet, da dort explizit bekannt ist, welchem Prototypen (SOLL-Typ) ein Merkmalsvektor zugeordnet wird. In der Simulation kommen alle Informationen über den SOLL-Prototypen selbst aus einem stochastischen Prozess (der Erkennung mit den HMM) und sind daher gewissermaßen nur in der Tendenz richtig. Offensichtlich wird diese Tendenz von Strategien wie SCHWAM oder auch *LVQ1* besser berücksichtigt als der *LVQ2*.

3.4.14 Versuch 11 — VITER,LMITT,HAUEB

Die bisher durchgeführten Versuche adaptieren immer das aktuelle Codebuch, das zur Klassifikation des Adaptionswortes zur Verfügung steht. Dies führt zu den in Abschnitt 3.5 beschriebenen Fensterfunktionen zur Gewichtung von Start-Prototypen und Beobachtungen. Außerdem hat es den Vorteil, daß nur eine Variante der Codebücher zur Verfügung gehalten werden müssen.

Werden dagegen die beobachteten Verschiebevektoren $\Delta\underline{k}(t,m)$ aus Gl. 3.6 in Bezug auf den *initialen Prototypen* $\underline{s}_k(m,0)$ berechnet, über alle bisherigen Adaptionswörter gemittelt und jeweils nach jedem Adaptionswort mit g gewichtet vom initialen Prototypen subtrahiert, so ergibt sich eine rechteckige Fensterfunktion, deren konstante Höhe (Gewicht) von g abhängt. Gl. 3.32 wird dann zu

$$\begin{aligned}\overline{\Delta\underline{k}}(T,m,w) &= \underline{s}_k(m,0) - \frac{1}{w}\sum_{i=1}^{w}\frac{1}{T(i)}\sum_{t=1}^{T}\underline{k}(t,i) = \\ &= \underline{s}_k(m,0) - \frac{1}{w}\sum_{i=1}^{w}\overline{\underline{k}}(i) \qquad (3.30)\end{aligned}$$

Damit ergibt sich der nach dem Adaptionswort w berechnete Prototyp nach Gl. 3.13 zu

$$\begin{aligned}\underline{s}_k(m,w) &= \underline{s}_k(m,0) - g\,[\underline{s}_k(m,0) - \frac{1}{w}\sum_{i=1}^{w}\overline{\underline{k}}(i)] = \\ &= (1-g)\,\underline{s}_k(m,0) + g\,\frac{1}{w}\sum_{i=1}^{w}\overline{\underline{k}}(i) \qquad (3.31)\end{aligned}$$

D.h. der initiale Prototyp $\underline{s}_k(m,0)$ erhält immer das Gewicht $(1-g)$ und der Mittelwert aus allen Beobachtungen (Langzeit-Mittelwert) das Gewicht g. Die Folgen sind:

- Je größer w wird, desto weniger Gewicht erhält eine neue Beobachtung.
- Das System kann den initialen Prototypen niemals vergessen.
- Die Erkennungsraten werden auf jeden Fall mit steigendem w konvergieren.

Abbildung 3.13: Verlauf der Erkennungsraten mit Langzeit-Mittelwerten (LMITT)

Es werden unter denselben Versuchsbedingungen der Versuche 3 und 6 (vgl. Abschnitte 3.4.3 und 3.4.7) mehrere Versuche mit verschiedenen Werten für A_k durchgeführt. Abb. 3.13 zeigt den Verlauf der Erkennungsraten für 100 Adaptionswörter unter den Bedingungen des Versuchs 3 mit $A_k = 1.0$ (bestes Ergebnis).

Die hier erzielten Verbesserungen erreichen vergleichbare Werte wie in Versuchen mit Potentialfunktionen als Fenster, wenn über einen längeren Zeitraum (100 Wörter) adaptiert wird. Allerdings läßt sich der Adaptionsvorgang hier nicht weiter beschleunigen, da mit $A_k = 1.0$ bereits nach dem ersten Wort die initialen Prototypen durch gemittelte, beobachtete Merkmalsvektoren ersetzt werden. Die Folge ist, daß Strategien wie SCHWAM innerhalb der ersten 10 - 20 Wörter bedeutend besser adaptieren (vgl. in Abb. 3.7 Erkennungsrate bei 78.5 % nach 10 Wörtern). Der Grund liegt wahrscheinlich darin, daß die Verschiebevektoren immer in Bezug auf den initialen Prototypen gebildet werden und somit keine iterative Verbesserung des Prototypen möglich ist. Außerdem bleibt ein einmal schlecht geschätzter Verschiebevektor zu jeder neuen Beobachtung gleich gewichtet und kann nie mehr vergessen werden. Da auch der Rechenaufwand deutlich höher liegt als bei den übrigen Versuchen, wird die Strategie LMITT hier nicht weiter verfolgt.

Ein ähnliches Verfahren wurde bereits in [Hua91] getestet. Allerdings wurde dabei vom unbekannten Sprecher ein Adaptionstext von 40 Sätzen (entspricht ungefähr 240 Wörtern) gesprochen und die Adaption in einem einzigen Schritt durchgeführt.

3.5 Gewichtung von initialen Prototypen und Beobachtungen

Die folgende Betrachtung will die Gewichtung des Ursprungs-Prototypen sowie der beobachteten Merkmalsvektoren in Abhängigkeit der Größe $g(w)$ und den Adaptionswörtern w anschaulich machen. Betrachtet wird nur ein Prototyp $\underline{s}_k(m)$ in einem der drei Codebücher. Der Faktor u in Gl. 3.6 sei hier vernachlässigt. Aus der Beobachtung des Adaptionswortes w ($w = 1 \ldots W$) werden jeweils T Differenzvektoren $\Delta\underline{k}(t)$, $t = 1 \ldots T$ berechnet. Diese werden nach rekursiver Vorschrift gemittelt (Gl. 3.12), was dem arithmetischen Mittelwert entspricht:

$$\begin{aligned}
\overline{\Delta\underline{k}}(T,m,w) &= 1/T \sum_{t=1}^{T} \underline{s}_k(m,w-1) - \underline{k}(t,w) = \\
&= \underline{s}_k(m,w-1) - 1/T \sum_{t=1}^{T} \underline{k}(t,w) = \\
&= \underline{s}_k(m,w-1) - \overline{\underline{k}}(w) \qquad (3.32)
\end{aligned}$$

Die Adaptionsstärke $g(w)$ sei in dieser Herleitung konstant $g(w) = g$. Nach Gl. 3.13 berechnet sich dann der Prototyp nach dem w-ten Adaptionswort zu

$$\begin{aligned}
\underline{s}_k(m,w) &= \underline{s}_k(m,w-1) - g\ \overline{\Delta\underline{k}}(T,m,w) = \\
&= \underline{s}_k(m,w-1) - g\ (\underline{s}_k(m,w-1) - \overline{\underline{k}}(w)) = \\
&= (1-g)\ \underline{s}_k(m,w-1) + g\ \overline{\underline{k}}(w) \qquad (3.33)
\end{aligned}$$

Diese rekursive Vorschrift läßt sich umformen in die endliche Reihenformel

$$\underline{s}_k(m,w) = (1-g)^w\ \underline{s}_k(m,0) + \sum_{j=1}^{w} (1-g)^{w-j}\ g\ \overline{\underline{k}}(j) \qquad (3.34)$$

bezogen auf den ursprünglichen Prototypen $\underline{s}_k(m,0)$. Aus dieser Formel ist ersichtlich, daß der ursprüngliche oder Start-Prototyp mit dem Gewicht $(1-g)^w$, sowie die gemachten Beobachtungen $\overline{\underline{k}}(j)$ mit den Gewichten $g(1-g)^{w-j}$ zum neu geschätzten Prototypen beitragen. Es läßt sich zeigen, daß für jedes w und g die Summe über diese Gewichte gleich 1 ist und somit eine echte Mittelwertsbildung mit einer Potenzfunktion als zeitliches Fenster vorliegt.

$$(1-g)^w + \sum_{j=1}^{w} g\ (1-g)^{w-j} \equiv 1 \qquad (3.35)$$

Beweis durch Herleitung:

$$\begin{aligned}
(1-g)^w + \sum_{j=1}^{w} g\ (1-g)^{w-j} &= (1-g)^w + g\ (1-g)^w \sum_{j=1}^{w} \left(\frac{1}{(1-g)}\right)^j = \\
&= (1-g)^w + g\ (1-g)^w \left(\frac{1 - \left(\frac{1}{1-g}\right)^{w+1}}{1 - \frac{1}{1-g}} - 1\right) = \\
&= (1-g)^w\ (1 - g - (1 - g - (1-g)^{-w})) = \\
&= 1 \quad \text{(w.z.b.w.)} \qquad (3.36)
\end{aligned}$$

Abbildung 3.14: Gewichtung von Start-Prototypen (s) und Beobachtungen (k) für $w = 5$

Die Gewichtung des ursprünglichen Prototypen in Abhängigkeit von w, sowie die Gewichtung der Beobachtungen für $w = 5$ und $g = 0.2$ bzw. $g = 0.4$ zeigt Abb. 3.14.

3.6 Weitere Versuche

3.6.1 Kombination von kurzer und langer Adaptionsphase

Mit Hilfe der in Abschnitt 3.4 gewonnenen Daten wird für die beiden Strategien VERSCH und SCHWAM ein kombinierter Versuch in zwei Adaptionsphasen I und II durchgeführt (KOMB). Es zeigt sich, daß die dort ermittelten Werte für A_k in der Strategie SCHWAM auch in kombinierter Weise zum Erfolg führen, wogegen in der Strategie VERSCH nach der Phase I eine deutliche Verschlechterung zu beobachten ist. Offensichtlich ist die Verbesserung von Phase I hier nicht zu halten, wenn weiter mit $A_k = 0.10$ adaptiert wird. Der Wert für A_k wurde daher in Phase II auf 0.005 reduziert, um Konvergenzverhalten zu bewirken.

Abbildung 3.15: Verlauf der Erkennungsrate in Abhängigkeit von Adaptionswörtern im kombinierten Versuch für Strategie VERSCH (durchgezogen) und SCHWAM (gestrichelt)

Es wurden folgende Parameter gewählt:

Strategie	Dauer A.phase I	A_k in I	A_k in II
VERSCH	12 Wörter	0.15	0.005
SCHWAM	16 Wörter	0.50	0.05

Abbildung 3.15 zeigt den dabei gemessenen Verlauf der Erkennungsrate über 80 Adaptionswörter. Die Versuchsbedingungen entsprechen den Versuchen 7 bzw. 11.

3.6.2 Versuche mit phonetisch motiviertem Codebuch

Im Rahmen eines weiteren Projekts am Lehrstuhl für Datenverarbeitung der TU München (ADDCB) wurde ein *phonetisch motiviertes Codebuch* mit 64 Prototypen für den Merkmalsvektor *sh* erstellt, welches gegenüber den Standard-Codebuch leichte Verbesserungen der Distorsion aufweist.

Die Versuche 7 und 11 werden jeweils mit diesem Codebuch wiederholt. Die Codebücher für die Merkmalsvektoren *di* und *la* bleiben unverändert. Die Erkennungsrate für nicht adaptierte Codebücher steigt dabei von 72.08 % auf 76.65 %.

Es zeigt sich, daß das phonetische Codebuch zwar einen deutlichen Gewinn bei unadaptierten Codebüchern erbringt, dieser sich jedoch nicht auch auf die Adaption erstreckt, sondern nach der Adaption fast die gleichen Raten erzielt werden wie mit konventionellen Codebüchern. Der gleiche Effekt ist auch bei

Abbildung 3.16: Verlauf der Erkennungsraten mit weiblichem Sprecher im Kombinationsversuch

Sprechern zu beobachten, die ebenfalls ohne Adaption bereits bessere Erkennungsraten erzielen.

3.6.3 Versuch mit weiblicher Sprecherin

Da die bisherigen Versuche nur mit einem männlichen Sprecher durchgeführt wurden, sollen die Versuchsvarianten mit den besten Ergebnissen mit den Daten eines weiblichen Sprechers wiederholt werden. Abb. 3.16 zeigt den Verlauf der Erkennungsrate im Kombinationsversuch unter gleichen Bedingungen wie in Abschnitt 3.6.1.

3.7 Zusammenfassung der Ergebnisse

In Tabelle 3.2 sind die maximal erzielten Verbesserungen in % bezogen auf die Erkennungsrate ohne Adaption (72.08 %) eingetragen. Bei Versuchen mit kurzer Adaptionsphase (20 W.) ist zusätzlich in Klammern die optimale Anzahl von Adaptionswörtern eingetragen, soweit diese bestimmbar ist. Ist bei Versuchen mit langer Adaptionsphase (100 W.) Konvergenz beobachtet worden, wird ein (+) eingetragen, bei fehlender Konvergenz ein (-).

Beispielsweise erzielt das Verfahren SCHWAM mit halbüberwachter Adaption bereits nach 16 Wörtern einen relativen Gewinn von 10.42 %. Dies entspricht einer Verminderung der Erkennungsfehlerrate von 27.92 % auf 20.41 %, d.h. die Erkennungsfehlerrate wird um 26.9 % reduziert.

Strategien		NOVITER		VITER			
VERSCH		20 W.	100 W.	20 W.	100 W.	KOMB	LMITT
	UNUEB	4.86 (10)	6.25 (-)	8.33 (10)	6.25 (+)	-	-
	HAUEB	-	-	9.03 (12)	6.94 (-)	8.33 (+)	8.33 (+)
SCHWAM		20 W.	100 W.	20 W.	100 W.	KOMB	LMITT
	UNUEB	4.16 (?)	2.08 (+)	9.72 (13)	6.94 (+)	-	-
	HAUEB	-	-	10.42 (16)	7.64 (+)	11.11 (+)	2.8 (+)
ISTSOLL				20 W.	100 W.	KOMB	LMITT
	UNUEB	-	-	7.64 (10)	-	-	-
	HAUEB	-	-	6.94 (10)	-	-	-
LVQ1				20 W.	100 W.	KOMB	LMITT
	HAUEB	-	-	6.90	6.90	-	-
LVQ2				20 W.	100 W.	KOMB	LMITT
	HAUEB	-	-	6.25	-	-	-

Tabelle 3.2: Übersicht der Ergebnisse der Adaption von semikontinuierlichen Codebüchern

Kapitel 4

Adaption von Hidden Markov Modellen

Zusammenfassung

Sprecheradaption von semikontinuierlichen HMM (SCHMM) durch Adaption der Mixtur-Koeffizienten in den Zuständen der HMM auf einen neuen Sprecher ohne Veränderung der Vorverarbeitung und semikontinuierlichen Vektorquantisierung (siehe Kapitel 3). Aus wenigen Äußerungen des neuen Sprechers wird versucht, möglichst viel generalisierte Information für möglichst viele Modelle zu gewinnen. Dabei werden die Viterbi-Pfade durch das aktuell beste virtuelle Wortmodell (unüberwacht) bzw. durch das virtuelle Wortmodell des tatsächlich gesprochenen Wortes (halbüberwacht) ausgewertet. Die Generalisierung erfolgt einerseits über die Phonem-HMM, andererseits durch die Mixtur-Koeffizienten in den Zuständen der SCHMM.

4.1 Grundgedanke

Die Adaption der statistischen Modelle, die explizit die Lautbildung der Sprache nachbilden, auf die der Erkenner trainiert wurde, ist theoretisch die mächtigste Möglichkeit der Anpassung eines Systems an einen neuen Sprecher. Leider läuft dies in den meisten Fällen darauf hinaus, die statistischen Modelle auf den unbekannten Sprecher neu zu trainieren (vgl. z.B. [Rig89], [Ken90], [Jar87]). Dazu ist jedoch ausreichend, d.h. eigentlich genauso viel Sprachmaterial des neuen Sprechers nötig, wie im sprecherabhängigen Betrieb des Erkenners. Im allgemeinen ist ein solches Training wegen des hohen Aufwands in der Praxis nicht durchführbar, weshalb durch verschiedene Verfahren der Interpolation entweder das wenige Sprachmaterial des neuen Sprechers künstlich vergrößert wird, oder die Parameter des bereits (gut) trainierten Erkenners genutzt werden.

Unter realistischen Bedingungen besteht das Problem darin, aus wenig Material (Größenordnung: einige Wörter) des neuen Sprechers soviel generalisierte Information zu gewinnen, daß damit auch Modelle, die an den gemachten Äußerungen des neuen Sprechers nicht direkt beteiligt sind, auf diesen adaptiert werden können. Anschaulicherweise muß dies mit einer Adaption in vorgeordneten

Verarbeitungsstufen besser gelingen; trotzdem soll hier der Versuch gemacht werden, unter den gleichen Randbedingungen wie in Kapitel 3 eine Adaption des ASE Systems aus Kapitel 2, Abschnitt 2.2.5 vorzunehmen. Hierbei erfolgt die Generalisierung durch möglichst kleine Untereinheiten, durch Phonem-HMM bzw. durch deren Zustände.

4.2 Adaption von Phonem-HMM

Die Vorverarbeitung und semikontinuierliche Vektorquantisierung erfolgt analog zu den Abschnitten 2.2.2 und 2.2.4, jedoch ohne Adaption der Codebuch-Prototypen. In Vorversuchen wurde festgestellt, daß schärfere Mixtur-Verteilungen[1] bei der Verwendung von kontextfreien, kleinen Spracheinheiten günstiger sind. Daher erfolgt die Berechnung der Pseudorückschlußwahrscheinlichkeiten $p_k(t, m)$ durch Erheben des gewichteten, Euklidischen Abstandes in den Exponenten von e und anschließende Normierung aller Werte auf Summe 1 (vgl. Abschnitt 2.2.4, Variante 2 der semikontinuierlichen Vektorquantisierung).

Die Abarbeitung erfolgt in Worteinheiten. Zu jedem Wort des Lexikons existiert ein Aussprachegraph, der lediglich die Dudenaussprache nach [Dud90] in SAMPA-Notation enthält (vgl. Anhang A). Zur Erkennung eines unbekannten Wortes werden die diesem Aussprachegraphen entsprechenden Phonem-HMM verknüpft, wobei nur innerhalb der einzelnen Phoneme Übersprünge erlaubt werden. Das auf diese Weise erhaltene, *virtuelle Wortmodell* wird mit Viterbi abgearbeitet. Der Wortschatz dieses Einzelwort-Erkenners ist somit theoretisch unbegrenzt; in den folgenden Versuchen ist jedoch das Lexikon auf das der Sotschek-Datenbasis von 341[2] Wörtern beschränkt.

Ein Wort gilt als erkannt, wenn seine Erzeugungswahrscheinlichkeit für die beobachtete Merkmalsvektorfolge die aller anderen virtuellen Wortmodelle übertrifft. Die Beurteilung der Anpassung der Phonemmodelle an den neuen Sprecher erfolgt durch den Test von Wörtern aus einer unabhängigen Stichprobe des neuen Sprechers (Sprachstichproben *Test1* und *Test2*).

Die Information für die Adaption der Phonem-HMM stammt implizit aus den Aussprache-Graphen des Lexikons. Denn nur dadurch ist es möglich, daß eine Teilfolge von Merkmalsvektoren auch auf ein HMM abgebildet wird, auf die sie – infolge des sprecherunabhängigen Tests – akustisch schlecht paßt, obwohl sie semantisch richtig zugeordnet ist. Durch den Viterbi-Pfad wird immer eine Folge von Pseudo-Rückschluß-Verteilungen auf eine Folge von Mixtur-Verteilungen abgebildet. Mit Hilfe dieser Abbildung werden die Mixturen an den neuen Sprecher adaptiert.

4.3 Übersicht der Versuche

Tabelle 4.1 zeigt als Übersicht die Nummerierung der Versuche.

Analog zu Kapitel 3 gibt es folgende zwei generelle Varianten, die für alle Versuche möglich sind:

- Unüberwachte Adaption (UNUEB).
  Unüberwacht bedeutet, daß vom Benutzer keinerlei Rückmeldung erfolgt.

[1] Im folgenden wird auch der Begriff 'Mixturen' anstatt 'Mixtur-Verteilung' verwendet
[2] Davon werden aus technischen Gründen nur 339 Wörter verwendet

Strategien	ADDMIX	REDMIX	INCMIX	SCHMIX
UNUEB	1	3	5	7
HAUEB	2	4	6	8

Tabelle 4.1: Übersicht der Versuche zur Adaption von Phonem-HMM

Das System nimmt seine Informationen allein aus dem vorhandenem apriori-Wissen der Aussprachegraphen im Lexikon, d.h. de facto aus dem Viterbi-Pfad des virtuellen Wortmodells mit der höchsten Erzeugungswahrscheinlichkeit. In dieser Variante geht das System immer 'optimistisch' vor, d.h. es nimmt immer an, korrekt entschieden zu haben.

- Halbüberwachte Adaption (HAUEB).
  Halbüberwacht bedeutet, daß vom Benutzer bei Fehlerkennung eine Warnung erfolgt und das richtige Wort aus B ermittelten Alternativen ausgewählt wird ($B = 5$). D.h. es wird der Viterbi-Pfad des vom Benutzer bezeichneten Modells ausgewertet oder gar nicht adaptiert.

Da bereits in Kapitel 3 keine Möglichkeit für eine dynamische Adaption ermittelt werden konnte, werden auch die folgenden Versuche mit konstanten Adaptionsstärken durchgeführt.

$$g(w) = A_k \tag{4.1}$$

Folgende Strategien zur Adaption der Mixtur-Verteilungen in den Zuständen der Phonem-HMM werden untersucht:

- Die jeweils in einem Zustand eines Phonem-HMMs längs des Viterbi-Pfades beobachteten Pseudo-Rückschlüsse auf die Codebuch-Prototypen (pro frame eine Verteilung über alle Codebuch-Einträge) werden gewichtet mit einem konstanten Faktor A_k (Adaptionsstärke) auf die Mixtur-Verteilung dieses Zustandes addiert und anschließend die Mixturen wieder auf Summe 1 normiert (ADDMIX).

- Nur die R höchsten in einem Zustand eines Phonem-HMMs längs des Viterbi-Pfades beobachteten Pseudo-Rückschlüsse werden gewichtet mit einem konstanten Faktor A_k auf die Mixtur-Verteilung dieses Zustandes addiert und anschließend die Mixturen wieder auf Summe 1 normiert (REDMIX).

- Der Mixtur-Koeffizient, dessen Pseudo-Rückschluß in einem Frame maximal ist, wird in dem Zustand, in welchen der Frame auf dem Viterbi-Pfad gefallen ist, um einem konstanten Betrag (Adaptionsstärke) erhöht, unabhängig davon, wie hoch der Pseudo-Rückschluß war. Anschließend werden die Mixturen wieder auf Summe 1 normiert (INCMIX).

- Die Mixtur-Koeffizienten werden nicht nur im zugeordneten Zustand, sondern auch in den Zuständen aller Phonem-Modelle, welche eine ähnliche Mixtur-Verteilung enthalten, mit den gleichen Pseudorückschlüssen adaptiert. Das Gewicht der Adaption wird dabei einerseits durch die Adaptionsstärke wie unter Strategie ADDMIX, andererseits durch den Euklidischen Abstand der Mixtur-Verteilungen bestimmt (SCHMIX).

4.4 Beschreibung der Versuche

Sprachmaterial:	Sprecher TM ZN BR
Sprachstichprobe für Beurteilung:	*Test1*
Sprachstichprobe für Adaption:	*Adaption*

Adaption enthält die Sprachstichprobe des unbekannten Sprechers, anhand derer das System adaptieren soll. *Test1* enthält die Wörter, anhand derer die Erkennungsleistung des Systems nach einem oder mehreren Adaptionsschritt(en) beurteilt wird. Die beiden Wortmengen sind disjunkt, d.h. nicht die Adaptionsfähigkeit an eine bestimmte Stichprobe sondern an die Sprache des unbekannten Sprechers wird beurteilt. Das Lexikon enthält genau die Vereinigungsmenge der beiden Wortlisten.

Äußerer Versuchsablauf für alle Versuche:

1. Die Erkennungsrate mit unadaptierten Phonem-HMM wird ermittelt.
2. Das nächste Wort aus *Adaption* wird verarbeitet.
3. Die Phonem-HMM des erkannten Wortes werden adaptiert.
4. Alle Wörter aus *Test1* werden verarbeitet und die Erkennungsrate bestimmt.
5. Weiter bei 2.), bis kein Wort mehr in *Adaption*.

4.4.1 Versuch 1 — ADDMIX,UNUEB

1. In jedem Zeitschritt t des Adaptionswortes sind die Pseudo-Rückschlußwahrscheinlichkeiten der Merkmalsvektoren auf die M jeweiligen CB-Prototypen $\underline{p}_{sh}(t)$, $\underline{p}_{di}(t)$, und $\underline{p}_{la}(t)$ bekannt. Der Merkmalsvektor sei im folgenden mit k abgekürzt.
2. Außerdem ist die Folge der auf dem Viterbi-Pfad des virtuellen Modells mit der höchsten Erzeugungswahrscheinlichkeit durchlaufenen Zustände $\tau(t)$ durch Backtracking bestimmbar. Damit ergibt sich eine eindeutige Zuordnung von beobachteten Rückschlußwahrscheinlichkeiten auf Zustände der am Adaptionswort beteiligten Phoneme.
3. Die adaptierte Verteilung der Mixturen $\underline{q}'_k(z)$ nach der Beobachtung genau einer Pseudorückschlußwahrscheinlichkeit $\underline{p}_k(t)$ im Zustand $z = \tau(t)$ des virtuellen Modells berechnet sich zu

$$\underline{q}'_k(z) \quad = \quad \frac{\underline{q}_k(z) + A_k \underline{p}_k(t)}{\sum_{m=1}^{M}(q_k(z,m) + A_k p_k(t,m))} \tag{4.2}$$

   D.h., die beobachtete Verteilung der Pseudorückschlüsse wird mit A_k gewichtet auf die korrespondierende Verteilung der Mixturen addiert und anschließend auf Summe 1 normiert.
4. Die normierten, adaptierten Mixturen $\underline{q}'_k(z)$ ersetzen die bisherigen Mixturen $\underline{q}_k(z)$ in den beteiligten Phonem-Modellen.
5. Die zwei letzten Schritte werden für jede in z beobachtete Verteilung der Pseudorückschlußwahrscheinlichkeiten ($z = \tau(t)$) und für jeden Zustand z durchgeführt.

Wie leicht zu zeigen ist, entspricht diese Vorschrift der gleitenden Mittelwertsbildung über die beobachteten Pseudorückschlüsse mit konstanten Gewichten, deren Wert von A_k abhängt:

$$\underline{q}'_k(z) = \frac{\underline{q}_k(z) + A_k \underline{p}_k(t)}{1 + \sum_{m=1}^{M} A_k p_k(t,m)} \tag{4.3}$$

$$= \frac{\underline{q}_k(z) + A_k \underline{p}_k(t)}{1 + A_k \sum_{m=1}^{M} p_k(t,m)} \tag{4.4}$$

$$= \frac{\underline{q}_k(z) + A_k \underline{p}_k(t)}{1 + A_k} \tag{4.5}$$

$$= \frac{1}{1 + A_k} \underline{q}_k(z) + \frac{A_k}{1 + A_k} \underline{p}_k(t) \tag{4.6}$$

Dieser Kurzzeitmittelwert bewirkt ein exponentiell abklingendes Beobachtungszeitfenster, dessen Zeitkonstante nur von A_k abhängig ist ([Rus88], S. 61). Die Form der Gewichtungsfunktionen für die initiale Mixtur-Verteilung und die Beobachtungen entsprechen ungefähr denen aus Abb. 3.14 auf S. 43.

4.4.2 Versuch 2 — ADDMIX,HAUEB

Im Gegensatz zu Versuch 1 wird hier der Viterbi-Pfad durch das Modell des tatsächlich gesprochenen Wortes ausgewertet, sofern es sich unter den besten B Modellen befindet. Das bedeutet einen aktiven Eingriff des Benutzers (Markierung) in allen Fällen, in denen das tatsächlich gesprochene Wort nicht an erster Stelle erkannt wurde. Ist das tatsächlich gesprochene Wort nicht einmal unter den besten B Modellen aufgetreten, erfolgt keine Adaption. In allen anderen Punkten entspricht das Vorgehen Versuch 1 (s.o.).

4.4.3 Ergebnisse zu Versuch 1 und 2

Unter den Versuchsbedingungen 2 wird der Parameter A_k zwischen 0.001 und 1.0 variiert. Es werden für jeden Wert von A_k zwei Versuchreihen auf die gleiche Folge von Adaptionswörtern durchgeführt. Die erste erstreckt sich über 100 Wörter mit Test der Erkennungsleistung nach je 20 Wörtern, die zweite über nur die ersten 20 mit Test nach je 4 Wörtern. Dadurch sollen eventuell vorhandene rasche Anstiege nach den ersten Adaptionswörtern sichtbar gemacht werden. Eine generell feinere Auflösung der Tests war leider infolge der langen Rechenzeiten unmöglich (pro Test ca 6 Std.).

Bei niedrigen Werten von A_k ($0.001 \ldots 0.009$) ergibt sich langfristig eine gute Tendenz. Konvergenz ist allerdings nur für niedrige Werte von A_k zu beobachten. Höhere Werte (> 0.01) führen teilweise zu instabilen Zuständen, was darauf schließen läßt, daß die Information der sprecherunabhängigen Trainingsstichprobe wesentlich zum Erkennungsprozeß beiträgt und nicht leichtfertig durch Daten des neuen Sprechers unterdrückt werden darf. Die fein aufgelösten Verläufe innerhalb der ersten 20 Adaptionswörter sind sehr unruhig. Ein rascher Anstieg ist für keinen Wert zu beobachten. Dies war zu erwarten, da eine rasche Adaption der HMM kaum praktikabel ist. Ein Einbruch nach 4 Wörtern zeigt deutlich, daß es lokal durchaus zu Fehladaptionen kommen kann; nur im Mittel über genügend Material zeigt sich die Tendenz zur Verbesserung. Dies deckt

Abbildung 4.1: Verlauf der Erkennungsrate in Versuch 1 und 2 mit $A_k = 0.02$

sich mit Beobachtungen in [Bam91] (*DRAGON System*), wo ebenfalls nur im langfristigen Mittel eine Verbesserung der Erkennungsrate erreicht wurde.

Aufgrund dieser Ergebnisse scheint der Bereich $A_k = 0.01 \ldots 0.02$ für eine Adaption geeignet zu sein.

Bild 4.1 zeigt den Verlauf der Erkennungsrate in den Versuchen 1 und 2. Es werden 200 Wörter des neuen Sprechers angeboten (Sprachstichprobe *Adaption*) und die Phonem-HMM mit $A_k = 0.02$ nach Strategie ADDMIX adaptiert. Nach jeweils 10 Adaptionswörtern wird die Erkennungsrate innnerhalb einer unabhängigen Sprachstichprobe des neuen Sprechers (100 Worte) bestimmt. In beiden Versuchen ist zunächst ein starker Einbruch der Erkennungsrate zu beobachten. Dies kann mehrere Gründe haben (vgl. auch Abschnitt 4.5.1):

- Zufällig treten zu Beginn der Sprachstichprobe *Adaption* gehäuft Wörter auf, deren Verarbeitung zu Fehladaptionen führt.
- Durch die Adaption werden die Mixturen aus dem sprecher-unabhängigen Training zu rasch 'vergessen' (A_k zu groß).
- Die Teststichprobe *Test1* ist nicht repräsentativ genug für den neuen Sprecher. Infolgedessen kommt es zu starken Schwankungen der Erkennungsrate.

Deutlich ist zu sehen, daß die halbüberwachte (2) der unüberwachten Strategie (1) überlegen ist. Dennoch ist erstaunlich, daß auch ohne jeglichen Eingriff des Benutzers relative Gewinne von max. 11.4 % in der Erkennungsrate möglich sind. Die halbüberwachte Strategie erreicht max. 15.7 %.

4.4.4 Versuch 3 und 4 — REDMIX

Die Versuche 3 und 4 entsprechen im wesentlichen den Versuchen 1 und 2 mit folgendem Unterschied:
Es werden nicht alle beobachteten Pseudorückschlußwahrscheinlichkeiten $\underline{p}_k(t)$ auf die Mixturen eines zugeordneten Zustands addiert, sondern lediglich die R höchsten. Die davon nicht betroffenen Mixturen werden nur indirekt durch die anschließende Normierung auf Summe 1 verändert.

Die Motivation für diese Strategie ist die Folgende:
In der Testphase wird die Menge der berechneten Pseudorückschlüsse für jeden Merkmalsvektor auf die R höchsten beschränkt. Versuche im sprecherunabhängigen Test haben gezeigt, daß durch diese Reduktion die wesentlichen Informationen für die Erkennung besser ausgewertet werden ([Str91]). Dies ist eine direkte Folge der relativ weichen Verteilungen der Mixturen in den HMM. Erfolgt keine Reduktion, werden selbst charakteristische Spitzenwerte aus der VQ durch die Summation über alle Codebuch-Einträge 'weggebügelt'. Robuste Modelle mit weichen Mixtur-Verteilungen sind aber notwendig, um sprecherunabhängig arbeiten zu können; jedes adaptionsfähige System startet zwangsweise zunächst sprecherunabhängig. Zu scharfe Verteilungen bedeuten meist eine Überadaption an die Trainingsstichprobe. Enthält die Trainingsstichprobe genügend viele Sprecher, entstehen weiche Verteilungen automatisch dadurch, daß die Modelle während des Trainings eben mehrere Sprecher 'gesehen' haben, und außerdem beim Training keinerlei Reduktion vorgenommen wird (siehe auch Abschnitt 2.2.4).

Aus diesen Beobachtungen erhebt sich die Frage, ob die niedrigen Pseudorückschlüsse, die also nicht in den R besten enthalten sind, für die Adaption überhaupt eine relevante Information enthalten oder eventuell sogar störend wirken. Beim Training des Systems wurde keine Reduktion vorgenommen, um robuste Mixtur-Verteilungen zu bekommen. Bei der Adaption ist das Ziel im Gegenteil gerade die 'Überadaption' an den neuen Sprecher und kann eventuell durch Reduktion der VQ auch bei der Adaption erreicht werden.

Für die Versuche 3 und 4 wird $R = 3$ gewählt. Dieser Wert zeigte für den konventionellen, sprecherunabhängigen Test die besten Ergebnisse.

4.4.5 Ergebnisse zu Versuch 3 und 4

Bereits in den Vorversuchen werden bis auf Rechenungenauigkeiten die gleichen Werte erzielt wie in Versuch 1 und 2. Die These, daß die Reduktion der adaptierten Mixturen eine Verschärfung und damit eine Verbesserung der Erkennungsraten bewirken werde, hat sich nicht bestätigt. Auf eine Darstellung der Verläufe wird daher verzichtet.

4.4.6 Versuch 5 — INCMIX, UNUEB

1. In jedem Zeitschritt t des Adaptionswortes sind die Pseudo-Rückschlußwahrscheinlichkeiten der Merkmalsvektoren auf die M jeweiligen CB-Prototypen $\underline{p}_{sh}(t)$, $\underline{p}_{di}(t)$, und $\underline{p}_{la}(t)$ bekannt. Der Merkmalsvektor sei im folgenden mit k abgekürzt.

2. Außerdem ist die Folge der auf dem Viterbi-Pfad des virtuellen Modells mit der höchsten Erzeugungswahrscheinlichkeit durchlaufenen Zustände

$\tau(t)$ durch Backtracking bestimmbar. Damit ergibt sich eine eindeutige Zuordnung von beobachteten Rückschlußwahrscheinlichkeiten auf Zustände der am Adaptionswort beteiligten Phoneme.

3. Die adaptierte Verteilung der Mixturen $\underline{q'_k}(z)$ nach der Beobachtung genau einer Pseudorückschlußwahrscheinlichkeit $\underline{p}_k(t)$ im Zustand $z = \tau(t)$ des virtuellen Modells berechnet sich zu

$$q'_k(z,m) = \begin{cases} q_k(z,m) + A_k & : \quad m = m_{max} \\ q_k(z,m) & : \quad \text{sonst} \end{cases} \tag{4.7}$$

wobei gilt:

$$p_k(t, m_{max}) = \max_{m=1\ldots M}(p_k(t,m)) \tag{4.8}$$

D.h. Auf denjenigen Mixtur-Koeffizienten $q_k(z, m_{max})$, dessen korrespondierende Pseudorückschlußwahrscheinlichkeit in der Beobachtung maximal ist, wird der konstante Anteil A_k addiert.

4. Die Verteilung der Mixturen wird anschließend auf Summe 1 normiert:

$$\underline{q''_k}(z) \quad = \quad \frac{\underline{q'_k}(z)}{\sum_{m=1}^{M} q'_k(z,m)} \tag{4.9}$$

5. Die normierten, adaptierten Mixturen $\underline{q''_k}(z)$ ersetzen die bisherigen Mixturen $\underline{q_k}(z)$ in den beteiligten Phonem-Modellen.

6. Die zwei letzten Schritte werden für jede in z beobachtete Verteilung der Pseudorückschlußwahrscheinlichkeiten ($z = \tau(t)$) und für jeden Zustand z durchgeführt.

Dieses Verfahren stellt gewissermaßen eine weitere Verschärfung der oben geschilderten Strategien dar. Durch die Erhöhung nur desjenigen Mixtur-Koeffizienten mit der höchsten Pseudorückschlußwahrscheinlichkeit – unabhängig von seinem Wert – sollen die Modelle noch schärfer auf den neuen Sprecher adaptiert werden. Algorithmisch entspricht diese Strategie einem Training von diskreten HMM.

4.4.7 Versuch 6 — INCMIX,HAUEB

Im Gegensatz zu Versuch 5 wird hier der Viterbi-Pfad durch das Modell des tatsächlich gesprochenen Wortes ausgewertet, sofern es sich unter den besten B Modellen befindet (vgl. analog Versuch 2).

4.4.8 Ergebnisse zu Versuch 5 und 6

Unter den gleichen Versuchsbedingungen wie für Strategie ADDMIX werden auch die Vorversuche für Strategie INCMIX durchgeführt. Allerdings wird auf die fein aufgelösten Versuche innerhalb der ersten 20 Adaptionswörter verzichtet, da nach den negativen Ergebnissen in der Strategie ADMMIX kaum eine Verbesserung zu erwarten ist.

Abbildung 4.2: Verlauf der Erkennungsrate in Versuch 5 und 6 mit $A_k = 0.02$

Insgesamt läßt sich sagen, daß die Verläufe meistens nur schlechtere Werte erreichen als in vergleichbaren Versuchen mit Strategie ADDMIX. Die Annahme, daß durch künstliche Verschärfung nur des Mixtur-Koeffizienten mit der höchsten Pseudorückschlußwahrscheinlichkeit eine schnellere bzw. bessere Anpassung an den neuen Sprecher möglich wäre, hat sich also nicht bestätigt. Da aber dennoch ungefähr vergleichbare Ergebnisse mit der Strategie INCMIX erzielt werden, werden hoch auflösende Versuche mit $A_k = 0.02$ durchgeführt.

Bild 4.2 zeigt den Verlauf der Erkennungsrate in den Versuchen 5 und 6. Es werden 200 Wörter des neuen Sprechers angeboten (Sprachstichprobe *Adaption*) und die Phonem-HMM mit $A_k = 0.02$ nach Strategie INCMIX adaptiert. Nach jeweils 10 Adaptionswörtern wird die Erkennungsrate innnerhalb einer unabhängigen Sprachstichprobe des neuen Sprechers (100 Worte) bestimmt. Der Verlauf der Erkennungsrate streut stärker als in den Versuchen 1 und 2. Der maximale, relative Gewinn liegt bei beiden Versuchen bei ca. 12.5 %. Allerdings ist eine Konvergenz innerhalb der 200 Wörter nicht zu beobachten, so daß diese Werte nur Anhaltspunkte sein können.

4.4.9 Versuch 7 — SCHMIX,UNUEB

Die Generalisierung über die phonetischen HMM wirkt sich naturgemäß erst nach einer gewissen Zeit aus, nämlich dann, wenn ein großer Anteil der Modelle aus dem Inventar von 40 SAMPA-Phonemen mindestens einmal dem Adaptionsverfahren unterzogen wurde. Hier soll der Versuch unternommen werden, die Generalisierung dadurch zu beschleunigen, daß auch in nicht beobachteten Phonem-HMM die Mixtur-Verteilungen adaptiert werden.

Die Grundidee dabei ist die Vorstellung, daß ähnliche Mixtur-Verteilungen

auch ähnliche Signal-Statistiken haben müssen. Ähnliche Signal-Statistiken aber bedeuten, daß auch die Signal-Segmente, die von ähnlichen Mixtur-Verteilungen modelliert werden, einander ähnlich sind und daher mit den gleichen Parametern adaptiert werden können. Zur Beurteilung der Ähnlichkeit zweier Mixtur-Verteilungen $\underline{q}_k(h_i, z_k)$ und $\underline{q}_k(h_j, z_l)$ in den Zuständen z_k bzw. z_l der Phonem-HMM h_i bzw. h_j bietet sich folgende Gewichtung f an:

$$f(\underline{q}_k(h_i, z_k), \underline{q}_k(h_j, z_l)) = e^{-F \left| \underline{q}_k(h_i, z_k) - \underline{q}_k(h_j, z_l) \right|} \tag{4.10}$$

Der Parameter F bestimmt dabei die Steilheit der exponentiellen Abhängigkeit des Gewichts f vom Euklidischen Abstand der beiden Mixtur-Verteilungen. f kann jedoch nie größer als 1 werden, da der Euklidische Abstand minimal zu Null werden kann.

Vom Prinzip her ist dieses Verfahren der Strategie SCHWAM aus Abschnitt 3.4.5 sehr ähnlich (daher auch die Abkürzung SCHMIX, wie *Schwamm-Mixturen*). Wie dort soll aus der Beobachtung eines einzelnen Ereignisses (hier die Zugehörigkeits-Verteilung der Vektor-Quantisierung, dort der Merkmalsvektor) generalisierte Information für weitere Parameter des Systems gewonnen werden, obwohl diese Parameter nicht direkt mit dem beobachteten Ereignis zu tun haben.

Ein ähnliches Verfahren wurde in [Shi91] beschrieben. Dort sollten kontinuierliche HMM für Halbsilben adaptiert werden, indem die Mittelwerte der Verteilungen von HMM, die in der Adaptionsphase nicht berücksichtigt wurden, in Abhängigkeit von der Distanz zu adaptierten Mittelwerten verschoben wurden.

Der Algorithmus für die Strategie SCHMIX ist bis auf Punkt 3 identisch mit der Strategie ADDMIX. Sei h_τ das Phonem-HMM mit der höchsten Erzeugungswahrscheinlichkeit und $z_\tau = \tau(t)$ der vom Viterbi-Pfad zugeordnete Zustand. Dann lautet Punkt 3 nun wie folgt:

> Die adaptierten Verteilungen aller Zustände in allen Phonem-HMM $\underline{q}'_k(h_i, z_k)$ berechnen sich zu
>
> $$\underline{q}'_k(h_i, z_k) = \frac{\underline{q}_k(h_i, z_k) + A_k \; f(\underline{q}_k(h_i, z_k), \underline{q}_k(h_\tau, z_\tau)) \; \underline{p}_k(t)}{\sum_{m=1}^{M} (q_k(h_i, z_k, m) + A_k \; f(\underline{q}_k(h_i, z_k), \underline{q}_k(h_\tau, z_\tau)) \; p_k(t, m))} \tag{4.11}$$
>
> D.h., die beobachtete Verteilung der Pseudorückschlüsse $\underline{p}_k(t)$ wird mit A_k und dem Gewicht f multipliziert und auf alle Mixtur-Verteilungen addiert, und diese werden anschließend wieder auf Summe 1 normiert.

4.4.10 Versuch 8 — SCHMIX,HAUEB

Versuch 8 unterscheidet sich zu Versuch 7 nur dadurch, daß h_τ nicht das HMM mit der höchsten Erzeugungswahrscheinlichkeit ist, sondern das vom Benutzer markierte Modell (siehe sinngemäß Versuch 2, Abschnitt 4.4.2).

4.4.11 Ergebnisse zu Versuch 7 und 8

In Vorversuchen wird mit einer festen Adaptionsstärke $A_k = 0.02$ der Faktor F aus Gl. 4.10 zwischen 1 und 5000 variiert. Ein optimaler Verlauf der Erken-

Abbildung 4.3: Verlauf der Erkennungsrate in Versuch 7 und 8

nungsrate ergibt sich bei einem Wert von $F = 300$. Bild 4.3 zeigt den Verlauf der Erkennungsraten in Versuch 7 und 8 unter den gleichen Versuchsbedingungen wie in Versuch 1 respektive 2.

Der genaue Vergleich mit den Ergebnissen aus Versuch 1 und 2 ergibt eine mittlere Verbesserung der Erkennungsraten[3] von 0.2 % und 0.8 %. Außerdem sind die Einbrüche zum Beginn der Sitzung nicht mehr so stark. Ob sich der recht erhebliche Rechenaufwand der Strategie SCHMIX dafür in der Praxis lohnt, ist fraglich. Prinzipiell jedoch ist die Generalisierung über Mixtur-Verteilungen erfolgreich.

4.5 Weitere Versuche

Zusätzlich zu den oben beschriebenen Versuchen werden Varianten des Versuchs 2 durchgeführt.

Es handelt sich dabei um:

- Vertauschung der Sprachstichproben. Adaption auf Sprachstichprobe *Test1*, Test mit Sprachstichprobe *Adaption*.
- Größere Teststichprobe, Test mit Sprachstichprobe *Test2*.
- Tests mit weiteren Sprechern.

[3] Arithmetischer Mittelwert aller Ergebnisse einer Sitzung

Abbildung 4.4: Verlauf der Erkennungsrate bei Vertauschung der Adaptions- und Teststichprobe in Versuch 2 (s. Text)

4.5.1 Vertauschung der Sprachstichproben

Dieser Versuch soll die Annahme überprüfen, bei dem anfänglichen Einbruch der Erkennungsrate in den bisherigen Versuchen handele es sich um ein Zufallsprodukt, hervorgerufen durch eine Häufung mehrerer 'ungünstiger' Adaptionswörter zu Beginn der Sprachstichprobe *Adaption*.

Bild 4.4 zeigt den Verlauf der Erkennungsrate bei vertauschten Sprachmaterialien. D.h. eine Adaption erfolgt anhand von *Test1* und getestet wird mit *Adaption*. Der durchgezogene Graph zeigt noch einmal die Ergebnisse von Versuch 2, der gestrichelte Graph die Ergebnisse bei Vertauschung der Sprachstichproben. Der Vergleich mit dem Versuch 2 zeigt, daß die obige Annahme richtig ist. Eine generelle anfängliche Verschlechterung infolge von zu geringer Gewichtung der initialen Mixturen ist also nicht anzunehmen.

4.5.2 Größere Teststichprobe

Die Ergebnisse zeigen meist starke Schwankungen der Erkennungsrate bei der Beurteilung der Adaption anhand der unabhängigen Sprachstichprobe *Test1*. Da diese sich aus nur 100 verschiedenen Wörtern zusammensetzt, besteht der Verdacht, daß sie die Sprache des neuen Sprechers nicht in genügendem Maße repräsentiert und deshalb die Ergebnisse stark streuen. Eine Wiederholung des Versuchs 2 mit einer größeren Teststichprobe soll zeigen, ob die Daten der Sprachstichprobe *Test1* für die Sprache des neuen Sprechers repräsentativ genug sind. Bild 4.5 zeigt den Verlauf der Erkennungsrate in Versuch 2 mit *Test1* und *Test2* als Teststichproben. Der Verlauf der Erkennungsrate mit *Test2* ist deutlich ruhiger als mit *Test1*. In beiden Fällen erfolgt jedoch eine Steigerung der

Abbildung 4.5: Verlauf der Erkennungsrate in Versuch 2 mit Teststichprobe *Test1* und *Test2*

Erkennungsrate um ungefähr den gleichen Betrag. Die Sprachstichprobe *Test1* scheint folglich, zumindest für die Gesamtbeurteilung der Adaption die Sprache des neuen Sprechers ausreichend zu repräsentieren.

4.5.3 Tests mit weiteren Sprechern

Um die gewonnenen Erkenntnisse abzusichern und sicherzustellen, daß die ermittelten Parameter der Adaption nicht nur für einen bestimmten Testsprecher Gültigkeit besitzen, wird der Versuch 2 mit den Daten eines weiteren Sprechers (ZN) und einer Sprecherin (BR) wiederholt. Beide sind natürlich ebenso wie der Sprecher TM nicht in der Trainingsstichprobe enthalten. Bild 4.6 zeigt den Verlauf der Erkennungsraten für die Sprecher ZN und BR. In beiden Versuchen ist ein Gewinn zu beobachten, der etwa dem des Sprechers aus Abschnitt 4.4 entspricht. Die ermittelten Parameter sind also durchaus verallgemeinerbar.

4.6 Zusammenfassung der Ergebnisse

Die Ergebnisse zeigen, daß mit relativ wenig Aufwand an Speicher und Rechenleistung auch innerhalb kurzer Sitzungen eine unüberwachte Adaption von SCHMM unter realistischen Bedingungen möglich ist. Sie dauert länger als die Adaption von Codebüchern, erreicht aber auch deutlich bessere Werte als diese. Tabelle 4.2 zeigt jeweils die maximal erzielten, relativen Verbesserungen der Erkennungsrate bezogen auf die Erkennungsrate ohne Adaption (70.0 %).

Beispielsweise erzielt das Verfahren SCHMIX bei halbüberwachter Adaption eine relative Verbesserung der Erkennungsrate von 18.6 %. Dies entspricht einer

Abbildung 4.6: Verlauf der Erkennungsrate in Versuch 2 mit Sprecher ZN (männl.) und BR (weibl.)

Strategie	ADDMIX	INCMIX	SCHMIX
UNUEB	11.4 %	12.5 %	17.2 %
HAUEB	15.7 %	12.5 %	18.6 %

Tabelle 4.2: Übersicht der Ergebnisse der Adaption von Phonem-Modellen

Reduktion der Erkennungsfehlerrate von 30 % auf 17 %, d.h. die Fehlerrate wird um 43.3 % vermindert.

Kapitel 5

Adaption der Bewertung von Lautsymbolen

Zusammenfassung

Lauthypothesen vom Klassifikator werden durch a priori geschätzte Rückschlußwahrscheinlichkeiten (RSW) bewertet. Diese sollten der Quellenstatistik des Gesamtsystems Sprecher – Klassifikator entsprechen. In der Praxis gilt dies nur für das Trainingsmaterial, da sich die Quellenstatistik bei einem Sprecherwechsel ändert. Durch Beobachtung der Verwechslungen während der Sitzung mit einem neuen Sprecher wird eine kontinuierliche Aktualisierung der RSW erreicht. Diese kann überwacht oder unüberwacht, mit konstanter oder dynamischer Lernrate erfolgen. Die dynamische Lernrate wird dabei mit Hilfe eines Entropiemaßes bestimmt. Da für den Nachweis der Wirksamkeit der vorgestellten Algorithmen sehr große Datenmengen erforderlich wären, wird eine Simulation durchgeführt, in welcher die Quellenstatistik genau definiert werden kann. Ein einfacher Maximum Likelihood Klassifikator zur Bewertung von Kunstwörtern wird auf diese Weise auf die neue Quellenstatistik adaptiert. Es zeigt sich, daß sowohl überwachte als auch unüberwachte Algorithmen die Leistung des Erkenners bis zur theoretisch möglichen Obergrenze steigern.

5.1 Grundgedanke

Systeme der ASE, die nach dem 'bottom up'-Prinzip arbeiten, werten keine anderen Wissensquellen aus, als diejenigen, die auf der jeweiligen Verarbeitungsstufe zur Verfügung stehen. Man sagt auch, es bestehen keinerlei Einschränkungen ('constraints'), welche das Erkennen bestimmter Äußerungen des Lexikons von vorn herein ausschließen. Eine Möglichkeit, ein solches ASE System für die Erkennung fließend gesprochener Sprache zu entwerfen, ist die Einführung einer phonetischen (bzw. allophonischen, o.a.) Zwischenrepräsentation des Sprachsignals und eine anschließende, rein symbolische Weiterverarbeitung nur dieser

Transkription.

Ein konkretes Beispiel bildet das am Lehrstuhl für Datenverarbeitung, TU München entwickelte System *SILBOS 3.0* ([Rus88], [Wei90], [Sch91/2]). Dort werden Worthypothesen an bestimmten Positionen des Sprachsignals (Silbenkerne) bewertet, indem die a priori geschätzten RSW der an der jeweiligen Worthypothese beteiligten Phonemgruppen multipliziert werden. Das Gesamtsystem vom Sprecher bis hin zum Klassifikator wird somit als eine stochastische Quelle betrachtet, welche in Abhängigkeit der gesprochenen Äußerung A mit bestimmter Wahrscheinlichkeit $P(T|A)$ die Transkription T erzeugt. In *SILBOS 3.0* werden diese diskreten Wahrscheinlichkeiten aus der Trainingstichprobe einmal geschätzt und für die Bewertung der Worthypothesen nach dem Prinzip der maximalen Likelihood verwendet. In einem sprecherunabhängigen System ist aber zu erwarten, daß sich die Quellenstatistik bei jedem Sprecher ändert und somit zu falschen Bewertungen von Worthypothesen führt.

In diesem Kapitel wird ein einfaches Verfahren vorgestellt, mit Hilfe dessen die Quellenstatistik für die Bewertung von Lauthypothesen (und damit auch für die Bewertung beliebiger größerer Komplexe) an den aktuellen Sprecher schritthaltend adaptiert werden kann. Da für die Beurteilung der entworfenen Methoden weder ein geeignetes System noch ausreichende Datenmengen zur Verfügung standen, wurde ein weiteres Simulationssystem entworfen. Dieses erlaubt es, sowohl die Quelle (Sprecher – Klassifikator) als auch die Lexikonstufe (Bewertung von Worthypothesen) eines 'bottom up' ASE Systems zu simulieren. Damit ist es möglich, sehr große Wortmengen in relativ kurzer Zeit zu verarbeiten und zuverlässige Statistiken über das Verhalten der verschiedenen Methoden bei unterschiedlichen Bedingungen zu erstellen. Da es sich dabei ausschließlich um diskrete Wahrscheinlichkeiten handelt, ist die Erzeugung einer künstlichen Quellenstatistik relativ einfach und unkritisch.

5.2 Simulation

5.2.1 Modell zur Simulation

Das Modell zur Simulation der Adaption auf Symbolebene besteht im wesentlichen aus vier Teilen (vgl. Bild 5.1): Die *Worterzeugung* erzeugt einen Strom von Worten w_i aus dem verwendeten Lexikon. Die Auswahl erfolgt zufallsgesteuert, wobei jedes Wort gleichwahrscheinlich auftritt $p(w_i) = 1/\lambda$. Die Worte des Lexikons werden so gewählt, daß die Redundanz der natürlichen Sprache weit unterschritten wird (ca. 1:40), d.h. das Lexikon enthält vor allem ähnlich transkribierte Worte, die dadurch schwer zu unterscheiden sind ("dann","denn","dach", ...). Das Lexikon enthält keine Aussprachevarianten sondern nur je eine kanonische Form. Das Auftreten der η verschiedenen Phoneme im Lexikon ist gleichverteilt, d.h. alle verwendeten Phoneme kommen gleich häufig vor $P(X_i) = 1/\eta$.

Die erzeugten Worte sind vorerst korrekt (nach dem Lexikon) transkribiert. In der anschließenden *Verfälschungs*-Stufe werden die phonetischen Symbole, aus welchen die Worte bestehen, anhand eines Zufallsprozesses vertauscht. Diese Operation modelliert die Verfälschung durch das Gesamtsystem von der Artikulation bis hin zum Output des Klassifikators. Dabei gelten drei Annahmen: Erstens, die Verwechslungsstatistik für ein bestimmtes Lautsymbol ist unabhängig von seinem Kontext. Zweitens, die Quellenstatistik ändert sich im Laufe einer

Abbildung 5.1: Modell zur Simulation der Adaption von Rückschlußwahrscheinlichkeiten

Sitzung nicht mehr, ist also stationär. Drittens, diese stationäre Quellenstatistik ist vom jeweiligen Sprecher abhängig. d.h. bei einem Sprecherwechsel ändert sich auch die Verwechslungsstatistik der Lautsymbole.

Die Quellenstatistik wird durch sog. Vorwärtswahrscheinlichkeitsmatrizen (VWKM) bestimmt, in welchen für jedes phonetische Symbol X die diskrete Wahrscheinlichkeit $P(Y|X)$ seiner Vertauschung mit einem beliebigen anderen phonetischen Symbol Y angegeben ist. In einem optimal arbeitenden System[1] ist die VWKM die Einheitsmatrix $\underline{E}$, d.h.

$$P(Y|X) = \begin{cases} 1 & : \quad X = Y \\ 0 & : \quad \text{sonst} \end{cases} \tag{5.1}$$

Die auf diese Weise erzeugte, verfälschte Transkription eines Wortes w_i wird an die nächste Einheit des Modells, die *Worterkennung* übergeben. Dort erfolgt eine Bewertung aller λ Worthypothesen in Lexikon anhand des Klassifikator-Outputs, indem die RSW der beobachteten Symbole $Y_j(w_i)$ in der Transkription von w_i auf die Laute $X_j(w_l)$ der lexikalen Einträge w_l aufmultipliziert werden.

$$P(w_l) = \prod_j P(X_j(w_l)|Y_j(w_i)) \qquad l = 1 \ldots L \tag{5.2}$$

Als erkannt gilt dann dasjenige Wort w_e, welches die niedrigste Gesamtwahrscheinlichkeit $P(w_e)$ aufweist.

$$w_e = \underset{l=1\ldots\lambda}{\operatorname{argmin}} P(w_l) \tag{5.3}$$

Ist $w_e = w_i$, war die Entscheidung des Klassifikator richtig, im anderen Falle falsch.

Die bedingten Wahrscheinlichkeiten $P(X|Y)$ sind in sog. Rückschlußwahrscheinlichkeitsmatrizen (RWKM) abgespeichert. Diese können für den Idealfall aus den entsprechenden Erzeugungswahrscheinlichkeiten in der VWKM berechnet werden, wobei die $P(X)$ gleichverteilt angenommen werden (Maximum Likelihood Prinzip).

$$P(X_i|Y_j) = \frac{P(Y_j|X_i)}{\sum_k P(Y_j|X_k)} \tag{5.4}$$

Technisch entspricht dies der transponierten VWKM, in der zeilenweise auf 1 normiert wurde.

In der Realität weichen die tatsächliche RWKM und die beobachtete RWKM natürlich mehr oder weniger voneinander ab und führen zu Erkennungsleistungen, die unter der theoretisch erreichbaren liegen.

Der vierte Teil des Modells, die eigentliche *Adaption*, hat Zugang zu folgenden Informationsquellen: Die erzeugte Transkription vom Klassifikator, im unüberwachten Falle das erkannte Wort, im überwachten Falle das tatsächlich gesprochene Wort. Die verschiedenen Adaptionsverfahren verändern nur die Einträge der RWKM. Dies kann nach jedem gesprochenen Wort (bei Einzelworterkennung) oder nach jedem gesprochenen Satz (Erkennung fließend gesprochener Sprache) erfolgen. In den nachfolgend beschriebenen Versuchen wurden jeweils 6 Wörter zu einem Satz zusammengefaßt und danach adaptiert.

[1] d.h. genau das, was der Sprecher zu sagen beabsichtigte, wird vom Klassifikator ausgegeben

5.2.2 Maß für die Verfälschung

Die VWKM bestimmt durch ihre diskreten Wahrscheinlichkeiten, inwieweit die erzeugten Wörter verfälscht und somit schwerer erkannt werden können. Gewünscht wäre ein allgemeines Maß dafür, wieviel Information durch diesen Verfälschungsprozeß verloren geht.

Zieht man eine Analogie zum Kanalmodell der klassischen Informationstheorie, so kann das oben beschriebene Modell als 'discrete memoryless channel' (DMC) nach [Hag89] bezeichnet werden. Ein DMC hat eine wohldefinierte Kanalkapazität C, die die maximal zu übertragende Informationsmenge pro Symbol ohne Störung wiedergibt. Ist η die Anzahl der phonetischen Symbole und treten alle Symbole gleich wahrscheinlich auf (s.o.), so ist C definiert als

$$\begin{aligned} C &= \sum_{\eta} \frac{1}{\eta} \quad \mathrm{ld} \frac{1}{1/\eta} \\ &= \mathrm{ld}\, \eta \end{aligned} \tag{5.5}$$

Die mittlere Transinformation H_T dagegen berücksichtigt die tatsächlichen Eigenschaften des Kanals, ist also bestimmt durch die Werte der VWKM. Als ein geeignetes Maß für die Verfälschung V durch die VWKM sei daher die prozentuelle Abweichung der Transinformation H_T von der Kanalkapazität C definiert:

$$V = 1 - \frac{H_T}{C} \quad 100\% \tag{5.6}$$

Die mittlere Transinformation H_T ist definiert als

$$\begin{aligned} H_T &= H(Y) - H(Y|X) \\ &= \sum_j \left[P(Y_j)\, \mathrm{ld} \frac{1}{P(Y_j)} \right] - \sum_i \sum_j \left[[P(X_i, Y_i)\, \mathrm{ld} \frac{1}{P(Y_j|X_i)} \right] \end{aligned} \tag{5.7}$$

Da $P(X_i) = \frac{1}{\eta}$ (s.o.), läßt sich $P(Y_j)$ schreiben als

$$\begin{aligned} P(Y_j) &= \sum_i [P(X_i) P(Y_j|X_i)] \\ &= \frac{1}{\eta} \sum_i P(Y_j|X_i) \end{aligned} \tag{5.8}$$

Außerdem gilt

$$P(X_i, Y_j) = P(X_i) P(Y_j|X_i) = \frac{1}{\eta}\, P(Y_j|X_i) \tag{5.9}$$

Damit wird Gl. 5.7 zu

$$\begin{aligned} H_T &= \sum_j \left[\frac{1}{\eta} \sum_i P(Y_j|X_i)\, \mathrm{ld} \frac{1}{\frac{1}{\eta} \sum_k P(Y_j|X_k)} - \sum_i \left[\frac{1}{\eta} P(Y_j|X_i)\, \mathrm{ld} \frac{1}{P(Y_j|X_i)} \right] \right] \\ &= \frac{1}{\eta} \sum_j \left[\sum_i \left[P(Y_j|X_i)\, \mathrm{ld} \frac{\eta}{\sum_k P(Y_j|X_k)} \right] - \sum_i \left[P(Y_j|X_i)\, \mathrm{ld} \frac{1}{P(Y_j|X_i)} \right] \right] \\ &= \frac{1}{\eta} \sum_i \sum_j \left[P(Y_j|X_i)\, \mathrm{ld} \frac{\eta\, P(Y_j|X_i)}{\sum_k P(Y_j|X_k)} \right] \end{aligned} \tag{5.10}$$

Abbildung 5.2: Absinken des Anteils unverfälschter Wörter W bei steigender Verfälschung V der VWKM

Damit ergibt sich für das Verfälschungsmaß V

$$V = 1 - \frac{\sum_i \sum_j P(Y_j|X_i) \operatorname{ld} \frac{\eta\, P(Y_j|X_I)}{\sum_k P(Y_j|X_k)}}{\eta \operatorname{ld} \eta} \quad 100\% \tag{5.11}$$

Um die Zweckmäßigkeit des Verfälschungsmaßes V zu überprüfen, wird experimentell der Anteil von gestörten Wörtern in Abhängigkeit von V bestimmt. Bild 5.2 zeigt eine fast lineare Abnahme der unverfälschten Wörter mit zunehmenden V. Somit scheint V gut geeignet, die Zuverlässigkeit des Systems Sprecher – Klassifikator zu beurteilen.

5.2.3 Erkennung im optimalen und realen Fall

Der optimale Fall aus der Sicht der *Worterkennung* ist gegeben, wenn die zur Bewertung benutzten RWKM nach Gl. 5.4 aus den VWKM der *Verfälschung* berechnet werden. Das bedeutet aber nicht, daß in diesem Falle eine Worterkennungsrate von 100 % erreicht wird, sondern lediglich, daß die Fehlerrate durch optimale Anpassung minimiert wird. Eine Erkennungsrate von 100 % wird nur erreicht, wenn sowohl RWKM als auch VWKM an den korrespondierenden Positionen nur 1-Werte oder 0-Werte enthalten ($V = 0$). Weicht nun die tatsächlich verwendete RWKM von der nach Gl. 5.4 berechneten RWKM ab, so ergibt sich eine Worterkennungsrate, die deutlich unter dem optimalen Fall liegt. Ziel einer Adaption ist es, möglichst nahe an den theoretisch möglichen Wert heranzukommen. Er kann aber niemals überschritten werden. Die Differenz der Worterkennungsraten im optimalen Fall und im realen Fall wird im folgenden

Anpassung genannt und von 0 bis 100 % skaliert. 100 % Anpassung bedeutet, der optimale Fall ist erreicht.

5.2.4 Verlauf einer Adaptionssitzung

Die VWKM der *Verfälschung* werden mit festen Werten in Anlehnung an reale Werte aus dem System *SILBOS 3.0* vorbelegt. Durch geeignete Verfahren können beliebige Verfälschungsmaße erzeugt werden. Mit Gl. 5.4 werden die optimalen Werte der RWKM aus der VWKM bestimmt. Damit kann die Worterkennungsrate bei optimaler Anpassung von 100 % in einer Simulation über 5000 Wörter bestimmt werden.

Die VWKM wird nun verändert. Dies geschieht z.B. durch Aufaddieren von 'Störmatrizen', welche Zufallswerte enthalten, und anschließende zeilenweise Normierung. In einer zweiten Simulation kann nun die Worterkennungsrate ohne Adaption (Anpassung 0 %) bestimmt werden. Die 'Störmatrizen' wurden so festgelegt, daß sich ein Absinken der Worterkennungsrate von 10 - 15 % ergab. Dies entspricht ungefähr dem Einbruch der Erkennungsrate in realen ASE Systemen nach einem Sprecherwechsel. Vollständige Symbolvertauschungen sind nicht enthalten.

Zur Beurteilung verschiedener Adaptionsstrategien wird nun die Erzeugung und Erkennung von bis zu 200 Sätzen simuliert, während die RWKM schritthaltend adaptiert werden, um die Störung der Quelle (Sprecherwechsel) auszugleichen. Die Erkennungsleistung des *Worterkenners* (Anpassung) wird anhand einer unabhängigen Stichprobe von 5000 Wörtern nach jedem Adaptionsschritt evaluiert.

5.3 Versuche

Alle nachfolgend beschriebenen Methoden entstanden unter der Berücksichtigung zweier Gesichtspunkte:

Erstens sollten die Algorithmen möglichst rekursiv arbeiten und möglichst wenig Rechen- und Speicheraufwand verursachen, damit sie in realen Echtzeitsystemen einsetzbar sind.

Zweitens soll der Benutzer das System sofort und ohne jegliche Einschränkung für seine Zwecke benutzen können. Letzteres bedeutet vor allem,

- die Adaption setzt bereits nach der ersten vollständig verarbeiteten Äußerung (Satz) ein.
- keine vorgeschriebenen Adaptionssätze oder -wörter.
- keine Identifikation oder Anmeldung des neuen Benutzers.
- keine Überwachung durch den Benutzer.

5.3.1 Unüberwachte Adaption

Algorithmus

Dem Algorithmus stehen die vom Klassifikator erzeugte (unsichere) Transkription des gesprochenen Wortes w_i, sowie die Transkription des Lexikonwortes w_e

Abbildung 5.3: Verlauf der Anpassung bei unüberwachter Adaption mit konstanter Adaptionsstärke $A_k = 0.1$

mit maximaler Bewertung aus der *Worterkennung* zur Verfügung. Durch eine 1:1 Zuordnung (alle Wörter haben gleiche Länge) werden Symbolverwechslungen $(X(w_i), Y(w_e))$ ermittelt und in einer entsprechenden Matrix $\underline{K}$ gezählt. Nach jeweils 6 Wörtern (= 1 Satz,s_i) wird die Matrix $\underline{K}$ gewichtet mit der Adaptionsstärke $g(s_i)$ auf die RWKM addiert und diese anschließend wieder zeilenweise normiert. Die Adaptionsstärke $g(s_i)$ ist in diesem Versuch konstant.

$$g(s_i) = A_k \tag{5.12}$$

Ergebnisse

In Vorversuchen wurde die optimale, konstante Adaptionsstärke $A_k = 0.1$ ermittelt.

Bild 5.3 zeigt den Verlauf der Anpassung bei unüberwachter Adaption über 200 Sätze (Adaptionsschritte). Die Worterkennungsraten für 0 % bzw. 100 % Anpassung liegen in diesem Versuch bei 77.7 % bzw. 89.0 %. D.h. durch die Verfälschungen der VWKM ($V = 35$ %) ist bei optimaler Anpassung des simulierten Spracherkenners eine Erkennungsrate von 89.0 % erreichbar. Durch den simulierten Sprecherwechsel weicht die tatsächlich verwendete RWKM von der optimalen RWKM soweit ab, daß sich eine gemessene Erkennungsrate von nur 77.7 % ergibt.

Der Test zur Beurteilung der Erkennungsrate wurde mit 5000 zufällig ausgewählten Wörtern durchgeführt.

Die Worterkennungsrate steigt zwar nicht streng monoton, aber tendenziell deutlich an und konvergiert bei ca. 90 % Anpassung, d.h. knapp unterhalb der theoretisch erreichbaren Erkennungsrate. Die zeitweisen Einbrüche im Verlauf

sind darauf zurückzuführen, daß es sich sowohl bei der Simulation als auch bei der Ermittlung der Erkennungsrate um einen echten stochastischen Prozeß handelt: Auch die Wörter zur Beurteilung der Erkennungsrate werden durch einen Zufallsprozeß erzeugt.

Insgesamt läßt sich sagen, daß eine vollkommen unüberwachte, schritthaltende Adaption auf symbolischer Ebene durchaus praktikabel ist.

5.3.2 Überwachte Adaption

Algorithmus

Im Gegensatz zum vorherigen Abschnitt verwendet der Adaptionsalgorithmus als Referenz nunmehr die Transkription des Wortes, das der Sprecher wirklich zu sprechen beabsichtigte (vgl. Bild 5.1). Das bedeutet, zu Beginn der Sitzung wird in 22.3 % der Fälle die Entscheidung des Klassifikators korrigiert (Anpassung 0 % entspricht 77.7 % Worterkennungsrate).

Eine solche Korrektur muß in einem realen System nicht unbedingt durch den Benutzer erfolgen. Ein ASE System verfügt normalerweise über weitere Verarbeitungsstufen, z.B. Syntax, Semantik, Pragmatik, Prosodie-Auswertung, etc., welche zur Erkennung eines ganzen Satzes bestimmte Worte als gesprochen erkennen, obwohl deren rein akustische Likelihood unter der eines anderen Wortes liegt. Solche Informationen können dann vom Adaptionsalgorithmus als Korrektur verwendet werden. In dieser Simulation erfolgt allerdings eine hundertprozentige Korrektur, d.h. es werden überhaupt keine Fehler zugelassen.

Mit dem gleichen Prinzip beschafft sich der Adaptionsalgorithmus in [Rig90] Information für eine unüberwachte Adaption der Wortmodelle im *TANGORA* System.

Alle übrigen Versuchsbedingungen sind identisch zum Versuch der unüberwachten Adaption (s.o.).

Ergebnisse

Der Graph 'statisch' in Bild 5.4 zeigt den Verlauf der Worterkennungsrate bei überwachter Adaption und konstanter Adaptionsstärke $A_k = 0.1$. Ein Vergleich mit Bild 5.3 zeigt, was zu erwarten war, nämlich einen deutlich rascheren Anstieg der Anpassung und ein etwas höheres Konvergenzniveau als bei unüberwachter Adaption. Bereits nach ca. 80 Sätzen wird ein Wert der Anpassung erreicht, auf dem der unüberwachte Algorithmus erst nach ca. 150 Sätzen konvergiert.

Es ist also zu erwarten, daß in einem realen System, in welchem weitere Wissensquellen zur Erkennung ganzer Sätze zur Verfügung stehen, auch ohne jeglichen Eingriff des Benutzers und innerhalb relativ kurzer Zeit spürbare Verbesserungen in der Erkennungsleistung zu erreichen sind.

5.3.3 Dynamische Adaptionsstärke

Algorithmus

Die bisher beschriebenen Verfahren arbeiten beide mit konstanter Adaptionsstärke $g(s_i) = A_k$. Der Wert für A_k ist ein Kompromiß zwischen möglichst rascher Adaption und dem Risiko Instabilitäten herbeizuführen. Wird $g(s_i)$ zu groß gewählt, werden u.U. bestimmte Beobachtungen überbewertet und dies

führt zu starken Einbrüchen der Worterkennungsraten, im unüberwachten Fall sogar bis hin zum Zusammenbruch des Systems. Andererseits ist gerade nach einem Sprecherwechsel eine kräftige Veränderung der RWKM erwünscht, um möglichst rasch die getätigten Beobachtungen in die RWKM zu übertragen.

Gewünscht wäre also ein dynamisches Einstellen der Adaptionsstärke $g(s_i)$, welche sich automatisch dem aktuellen Bedarf anpaßt.

Versuche, die Worterkennungsrate direkt als Entscheidungskriterium heranzuziehen, scheiterten aus ähnlichen Gründen wie schon in Kapitel 3, Abschnitt 3.3.2.

Einen guten Anhaltspunkt, wann stärkere Adaption notwendig wird, bietet die RWKM selber: findet ein Sprecherwechsel statt und ändert sich dadurch die Quellenstatistik, so ist auch bei kleiner Adaptionsstärke $g(w_i)$ eine Änderung der Entropie der RWKM zu beobachten. Es wird daher vorgeschlagen, diese Änderung als Entscheidungsgröße zu benutzen. Gl. 5.13 für die Adaptionsstärke im Satz s_k wird um einen dynamischen Anteil erweitert:

$$g(s_k) = A_k + A_d \, |H_{RWKM}(s_{k-1}) - H_{RWKM}(s_{k-2})| \tag{5.13}$$

A_d ist ein konstanter Faktor, der den Einfluß der Dynamik bestimmt und geeignet festgelegt werden muß. $H_{RWKM}(s_k)$ sei die Entropie der RWKM nach der Adaption anhand des Satzes s_k.

$$H_{RWKM}(s_k) \quad = \quad \sum_i \sum_j \left[P(X_i|Y_j, s_k) \, \mathrm{ld} \frac{1}{P(X_i|Y_j, s_k)} \right] \tag{5.14}$$

Die Änderung dieser Entropie aus den beiden vorangegangenen Sätzen bestimmt somit die Adaptionsstärke im aktuellen Satz.

Die Parameter A_k und A_d müssen so gewählt werden, daß der Algorithmus auf einen Wechsel der Quellenstatistik rasch reagiert (A_k nicht zu klein), andererseits darf das System nicht instabil werden und nach erfolgter Adaption die Adaptionsstärke wieder zurück regeln (A_d nicht zu groß).

Die übrigen Versuchsbedingungen entsprechen den bisher beschriebenen Versuchen.

Ergebnisse

Der Einsatz der dynamischen Adaptionsstärke mit unüberwachter Adaption erbrachte keinerlei Verbesserung gegenüber der konstanten Adaptionsstärke. Auf eine Darstellung der Ergebnisse wird daher verzichtet.

In Bild 5.4 zeigt der Graph 'dynamisch' die Entwicklung der Erkennungsrate bei überwachter Adaption, wenn die Adaptionsstärke nach Gl. 5.13 dynamisch eingestellt wird. Die Parameter sind $A_k = 0.05$ und $A_d = 3.0$.

Der Gewinn gegenüber der 'statischen' Adaption ist offensichtlich. Bereits nach 20 Sätzen konvergiert die Worterkennungsrate auf etwa dem gleichen End-Niveau. Damit erweist sich eine dynamische Einstellung der Adaptionsstärke zumindest im überwachten Falle als sehr wirkungsvoll.

5.3.4 Zusammenhang Adaption – Verfälschung

In den bisherigen Versuchen wurde eine konstante Verfälschung V von 35 % angenommen.

Abbildung 5.4: Verlauf der Anpassung bei überwachter Adaption mit konstanter und dynamischer Adaptionsstärke

Dies kann so interpretiert werden, daß das Gesamtsystem in der Simulation im optimalen Fall (Anpassung bei 100 %) eine Worterkennungsrate von 89 % erreicht. V beeinflußt also die Güte des simulierten Spracherkenners. Interessant ist nun die Frage: Innerhalb welcher Werte für V erbringt die überwachte oder unüberwachte Adaption einen Gewinn ?

Zu diesem Zweck wird die Verfälschung V in einem Bereich von 27 - 90 % variiert und für jeden Wert von V die Versuche aus Abschnitt 5.3.1 und 5.3.2 durchgeführt. Bild 5.5 zeigt jeweils die Worterkennungsrate nach 200 verarbeiteten Sätzen für verschiedene Werte von V. Die Kurve 'Optimale Erkennung' gibt die Erkennungsrate wieder, die mit optimal angepaßten RWKM (vgl. Gl. 5.4) erreicht werden (Anpassung 100 %). Die Kurve 'Unadaptierte Erkennung' dagegen zeigt die Worterkennungsrate nach einem Sprecherwechsel, sprich nach Verändern der RWKM (Anpassung 0 %).

Der Graph 'Überwachte Adaption' enthält die Resultate nach einer überwachten Adaption mit konstanter Adaptionsstärke. Wie erwartet schmiegt sich diese Kurve eng an die optimal erreichbare Grenze an. Es ergibt sich für alle Werte von V ein erheblicher Gewinn.

Die Kurve 'Unüberwachte Adaption' zeigt die Ergebnisse für den Fall der unüberwachten Adaption. Sie erreicht für Werte von V bis ca. 40 % ebenfalls eine sehr hohe Anpassung, fällt dann aber ab, und unterschreitet bei $V = 67\%$ sogar die Werte für Anpassung 0 %. Dies ist nicht weiter verwunderlich, da sich der Algorithmus quasi „an seinem eigenen Zopf aus dem Sumpf ziehen“ muß. Je unsicherer die Entscheidung für ein gesprochenes Wort wird (steigendes V), desto öfters wird der Algorithmus eine Fehladaption nicht vermeiden können. Diese führen wieder zu vermehrt neuen Fehlentscheidungen, etc.: das System

Abbildung 5.5: Worterkennungsraten bei variierter Verfälschung V

wird instabil.

Interessant ist die Beobachtung, daß diese Instabilität bei einer Worterkennungsrate im unadaptierten Fall (Anpassung 0 %) von ziemlich genau 50 % erfolgt. Zunächst drängt sich der Verdacht eines Zufallsereignisses auf. Da es sich um eine echte Simulation mit stochastischen Werten handelt, sind die Verläufe der Erkennungsraten nicht exakt reproduzierbar. Die Genauigkeit hängt ab von der verwendeten Anzahl von Testwörtern und der Zahl der verarbeiteten Sätze. Bei 5000 Testwörtern sind die Erkennungsraten an sich zwar sehr genau reproduzierbar, der Adaptionsprozeß nach 200 Sätzen kann jedoch infolge der zufälligen Auswahl der Wörter deutliche Unterschiede aufweisen. Eine Erhöhung der Anzahl verarbeiteter Sätze erschien jedoch nicht sinnvoll, da in realen Systemen auch nicht Tausende Sätzen zur Adaption verarbeitet werden können. Der Versuch wird daher einfach mehrfach wiederholt.

Es zeigt sich jedoch, daß der Punkt, an dem das System instabil wird, bis auf leichte Abweichungen immer bei etwa 50 % zu liegen kommt. Man kann dies so interpretieren, daß nach einem Sprecherwechsel noch mindestens 50 % der Wörter richtig erkannt werden muß, damit sich das System unüberwacht noch adaptieren kann. Je höher über diesem kritischen Punkt die Erkennungsrate liegt, desto schneller erfolgt die Adaption.

5.3.5 Einsatz im sprecherabhängigen ASE System

Die oben beschriebenen Algorithmen wurden in das am Lehrstuhl für Datenverarbeitung, TU München entwickelte ASE System *SILBOS 3.0* integriert. *SILBOS 3.0* ist sprecherabhängig trainiert und eignet sich daher nicht zur Überprüfung von Algorithmen zur Sprecheradaption. Allerdings taucht auch bei spre-

cherabhängigen Systemen das Problem auf, daß sich die sprechertypische Artikulation im Laufe der Zeit ändert. Die Folge ist ein allmähliches Absinken der mittleren Erkennungsraten. Die Trainingsdaten für *SILBOS 3.0* waren zur Zeit der Versuche bereits mehrere Jahre alt. Deshalb ist zu erwarten, daß die beschriebenen Algorithmen solche Verluste durch Anpassung der Wortbewertung zumindest teilweise wieder ausgleichen.

Ein Versuch mit Daten des Referenz-Sprechers ergab einen mittleren Zuwachs der Worterkennungsrate von 76.8 % auf 78 % bei unüberwachter Adaption, und auf 81.3 % bei überwachter Adaption nach 46 verarbeiteten Sätzen. In beiden Versuchen wurde eine konstante Adaptionsstärke verwendet.

Kapitel 6

Kontrolle der Adaption

Zusammenfassung

Bei den meisten bisher beschriebenen Adaptionsverfahren ist zu beobachten, daß nicht nach jedem Adaptionsschritt eine Verbesserung des Systems eintritt, wenn die Beurteilung dieser Verbesserung anhand einer unabhängigen Sprachstichprobe erfolgt. In dem folgenden Abschnitt soll ein Versuch einer Selbstkontrolle der Adaption unternommen werden, die den Adaptionsverlauf optimiert. Bei der Beurteilung solcher Verfahren tritt wiederum das Problem der vernünftigen Beurteilung anhand unabhängiger Stichproben auf.

6.1 Adaption – mit welchem Ziel ?

Es wurde z.B. in [Roz91] beobachtet, daß bei bestimmten Sprechern das Nachtraining von Codebuch-Prototypen eine Verbesserung, bei anderen aber sogar eine Verschlechterung ergab. Es wurde daher vorgeschlagen, nach einer Nachtrainingsphase mittels eines Intraset-Abstandsmaßes die vorgenommene Adaption zu beurteilen und nur bei erfolgversprechenden Werten die adaptierten Codebücher zu verwenden. Die Argumentation lautete, daß eine Vergrößerung des Intraset-Abstandes mit einer Expansion der Prototypen im Raum gleichzusetzen sei und damit auch die Trennbarkeit von Klassen verbessert würde. Die experimentelle Untersuchung zeigte allerdings nur eine sehr schwache Korrelation zwischen der Vergrößerung des Intraset-Abstandes und den erzielten Verbesserungen. Außerdem hat das Verfahren den Nachteil, daß erst nach einer langen Nachtrainingsphase (40 Sätze, also ca. 200 Wörter) entschieden werden kann, ob sich dieser Aufwand denn überhaupt lohnt.

Bei Verfahren der Sprecheradaption, welche nur die ad hoc anfallenden Daten des neuen Benutzers des Systems zur Verbesserung der Erkennungsleistung nutzen, kann man das Problem anders formulieren: Es kann nicht garantiert werden, daß die Adaption anhand eines so kleinen Materials (in dieser Untersuchung meist nur ein Wort) eine generelle Verbesserung der Erkennungsleistung mit dem Sprachmaterial des neuen Sprechers erzielt. Meistens ist dies zwar der Fall – sonst ließe sich über längere Zeit gar kein Gewinn erzielen – in einzelnen Fällen kommt es aber vor allem infolge von Fehlzuweisungen von Referenzdaten

Abbildung 6.1: Verlauf der Erkennungsrate in Versuch 2 zur Adaption von Phonem-HMM mit Sprachstichprobe *Adaption* als Teststichprobe

(Daten des noch nicht adaptierten Systems) und Zieldaten (Daten des neuen Sprechers) zu *lokalen Fehladaptionen*.

Man steht hier vor zwei Problemen: Erstens sollen lokale Fehladaptionen verhindert werden und so die Adaption an den neuen Benutzer beschleunigt werden. Zweitens ist es aber schwer zu entscheiden, was eine lokale Fehladaption ist. Denn jede Beurteilung einer Adaption erfolgt natürlich mit Datenmaterial des neuen Sprechers, welches aus technischen Gründen selbst in der Simulation nicht unendlich groß sein kann.

Streng genommen wird in allen bisherigen Versuchen nicht die Adaptionsfähigkeit des Systems an einen neuen Sprecher, sondern die Adaption an eine bestimmte Sprachstichprobe (meist *Test1* oder *Test2*) gemessen, wogegen das Material zur Adaption aus der disjunkten Stichprobe *Adaption* stammt (vgl. Abschnitt 2.2.1). Um eine gerechte Beurteilung der Adaptionsfähigkeit vorzunehmen, müßte das Testmaterial unendlich groß sein und somit auch die Daten der Stichprobe *Adaption* umfassen. Da dies nicht möglich ist, wurde eine technische vertretbare Größe gewählt, dafür aber die Daten der Adaption ausgeschlossen. Dies bedeutet einen sehr harten Test der Adaptionsfähigkeit eines Systems, was sich auch darin wiederspiegelt, daß in den meisten Veröffentlichungen zu diesem Thema die gleichen (oder zumindest überlappende) Sprachstichproben zur Adaption und zum Test Verwendung finden (z.B. [Bam91] u.a.).

Bild 6.1 zeigt den Verlauf der Erkennungsrate in Versuch 2 aus Kapitel 4, wenn zur Beurteilung die Sprachstichprobe *Adaption* verwendet wird. Wie erwartet, erreichen die Erkennungsraten in diesem Versuch ungefähr die Werte (96.5 %) eines Sprechers, dessen Daten zum Training des sprecherunabhängigen Systems Verwendung fanden (vgl. [Str91], S. 54). Der Vergleich mit Versuch 2

(Bild 4.1, S. 52) zeigt, daß die Sprachstichproben *Adaption* und *Test1* trotz ähnlicher Phonem-Statistik (vgl. Anhang B.2.4), infolge des unterschiedlichen Kontexts nur bedingt die Sprache des neuen Benutzers repräsentieren. Die Erkennungsraten nach 200 Adaptionwörtern weichen um ca. 15 % voneinander ab.

Eine Vergrößerung der Teststichprobe (*Test2*) führt zu keiner wesentlichen Verbesserung. Auch hier ist der Unterschied zwischen Test mit unabhängiger Stichprobe *Test2* und der Adaptionsstichprobe *Adaption* sehr groß (ca. 16 % nach 200 Wörtern).

Man könnte nun folgendermaßen argumentieren: Gewünscht ist, daß das System nicht an jedes nur denkbare Sprachmaterial des neuen Benutzers adaptiert, sondern möglichst schnell den Diskurs-Bereich erlernt, über den dieser gerade spricht. Bei einem Wechsel des Diskursbereichs kann dann schritthaltend eine erneute Adaption erfolgen. In diesem Falle wäre eine *Kontrolle der Adaption* wünschenswert, die nicht mit einer unabhängigen, endlichen Sprachstichprobe arbeitet (wie in den bisherigen Versuchen), welche in der Realität sowieso nicht zur Verfügung steht, sondern mit Hilfe der Daten des neuen Benutzers, soweit sie schon verfügbar sind.

Die logische Folge ist allerdings, daß nunmehr die *Beurteilung der Kontrolle* nicht mehr möglich ist. Darauf wird im letzten Abschnitt dieses Kapitels noch gesondert eingegangen.

6.2 Kontrolle durch Datensammlung

Die folgenden Darstellungen beziehen sich alle direkt oder indirekt auf die Untersuchungen in Kapitel 4, *Adaption von Hidden Markov Modellen.*

Die Adaption wird analog zu Versuch 1 bzw. 2 (Strategie ADDMIX) durchgeführt. Bisher wurden die Daten von jedem gesprochenen Wort des neuen Benutzers zur Adaption verwendet. Ziel des folgenden Versuchs ist es, solche Worte auszuschließen, deren Daten zu Fehladaptionen führen.

Zu diesem Zweck wird das System um einen Mechanismus zur Datensammlung erweitert. Ein einfacher FIFO-Buffer (*first in, first out*) erlaubt es, die C zuletzt gesprochenen Wörter $w_{i-C} \ldots w_{i-1}$ des neuen Benutzers zusammen mit ihren zugeordneten virtuellen Wortmodellen zu speichern. Natürlich ist der FIFO zu Beginn der Sitzung noch leer und erst nach C gesprochenen Wörtern gefüllt. Dann enthält jedes Speicherelement $c, \quad c = 1 \ldots C$ des FIFO zum darin gespeicherten Adaptionswort w_{i-C+c} folgende Daten:

- Die Ergebnisse der semikontinuierlichen Vektorquantisierung des gesprochenen Wortes $\underline{p}_k(t, w_{i-C+c})$, $t = 1 \ldots T(w_{i-C+c})$, $k = sh, di, la$.
- Den Zeiger auf das virtuelle Wortmodell $h(w_{i-C+c})$, welches während der Verarbeitung des Wortes zur Adaption gewählt wurde. Bei unüberwachter Adaption ist dies das Modell mit der höchsten Erzeugungswahrscheinlichkeit (EWK). Bei halbüberwachter Adaption das vom Benutzer markierte Modell, falls das tatsächlich gesprochene Wort unter den B besten Wörtern war. Andernfalls werden die Daten nicht in den FIFO aufgenommen.

Vor und *nach* der Adaption mit Wort w_i wird anhand der Daten im FIFO die *mittlere, logarithmierte EWK* $\overline{\log E}_{FIFO}$ der zugeordneten Modelle auf die

gespeicherten Adaptionswörter bestimmt[1].

$$\overline{\log E}_{FIFO} = \frac{1}{C}\sum_{c=1}^{C}\log E(w_c, h(w_c)) \tag{6.1}$$

Da inzwischen bestimmte Phonem-Modelle anhand der Daten aus Wort w_i adaptiert wurden und diese mit hoher Wahrscheinlichkeit in einem der gespeicherten Modelle im FIFO enthalten sind, werden diese beiden Werte in den meisten Fällen nicht gleich sein.

Die Differenz dieser beiden Werte

$$\Delta\overline{\log E}_{FIFO} = \overline{\log E}_{FIFOvor} - \overline{\log E}_{FIFOnach} \tag{6.2}$$

dient nun als Entscheidungsgröße, ob eine Adaption erfolgreich war oder nicht. Da es sich bei den logarithmierten EWKen immer um negative Werte handelt, ist $\Delta\overline{\log E}_{FIFO}$ negativ, falls die mittlere EWK innerhalb des FIFO infolge der Adaption betragsmäßig gestiegen ist und umgekehrt. Im letzteren Fall hat die Adaption die Likelihood der gespeicherten Daten auf ihre Modelle verschlechtert und der Adaptionsalgorithmus macht die vorgenommene Adaption rückgängig (Kontroll-Schwelle $S = 0$). Durch Werte ungleich Null für die Kontroll-Schwelle S kann die Kontrolle verschärft ($S < 0$) bzw. abgemildert ($S > 0$) werden.

Durch diesen Kontroll-Mechanismus werden also Wörter von der Adaption ausgeschlossen, wenn ihre Daten zu einer Verschlechterung der Likelihood der Modelle auf die C letzten beobachteten Wörter führen. Die Folge ist eine noch raschere Adaption auf die beobachteten Daten, jedoch *nicht unbedingt auch auf das generelle Sprachmaterial des neuen Sprechers.* Bild 6.2 zeigt den Verlauf der Erkennungsraten in Versuch 1 ohne und mit Kontrolle (gestrichelt). Obwohl durch den Kontroll-Algorithmus eine schärfere Anpassung an die Adaptionsstichprobe *Adaption* erfolgt, ergibt sich auch beim Test mit der Sprachstichprobe *Test1* ein qualitativ besserer Verlauf als ohne Kontrolle. Insbesondere die starken Einbrüche sind nicht mehr zu beobachten. Die Adaption nach 200 Wörtern erreicht ungefähr die gleichen Werte wie im Versuch ohne Kontrolle.

6.3 Problem: Beurteilung der Kontrolle

Es erhebt sich die Frage, wie die Effektivität des oben beschriebenen Kontroll-Algorithmus sinnvoll getestet und beurteilt werden kann. Das Beispiel des vorangegangenen Abschnitts (Bild 6.2) gibt keinerlei Hinweis darauf, ob das System sich tatsächlich an den aktuellen Diskursbereich des Sprechers besser adaptiert hat als in den regulären Versuchen, da es sich um die Beurteilung mit einer völlig unabhängigen Sprachstichprobe (*Test1*) handelt. Die Beurteilung anhand der Adaptionsstichprobe *Adaption* wie in Bild 6.1 ist andererseits ebenfalls trivial, da ja gerade aus dieser Stichprobe die C Referenzwörter stammen, mit denen die Kontrolle der Adaption durchgeführt wird.

An sich wären zwei Stichproben (für Adaption und Test) nötig, die beide aus einem sehr engen Diskursbereich stammen (z.B. irgendein Auskunftssystem), aber dennoch disjunkt sind. Solche Stichproben standen leider nicht zur Verfügung.

[1] Die Mittelwertsbildung wird lediglich durchgeführt, um Werte unabhängig von C zu erhalten.

Abbildung 6.2: Verlauf der Erkennungsrate in Versuch 1 zur Adaption von Phonem-HMM ohne und mit Kontrolle (gestrichelt), $S = -0.5 \quad C = 20$

Als Notlösung werden die zwei Versionen der Sprachstichprobe *Adaption* gewählt. Beide Versionen beinhalten die gleichen gesprochenen Sätze eines Sprechers aus zwei Aufnahmesitzungen. Allerdings reduziert sich damit die Wortmenge der zur Adaption angebotenen Wörter auf 100. Zum Vergleich werden die entsprechenden regulären Versuche (vgl. Kapitel 4) mit dem gleichen Stichprobenpaar wiederholt.

Bild 6.3 zeigt den Verlauf der Erkennungsrate in Versuch 2 aus Kapitel 4 ohne und mit Kontrolle der Adaption. Die Adaption erfolgt anhand der Daten der 1. Version von *Adaption*, der Test mit den Daten der 2. Version. Die Adaptionstärke ist $A_k = 0.02$, die Kontroll-Schwelle ist $S = 0.75$. Der Anteil der durch die Kontrolle zurückgewiesenen Adaptionwörter beträgt ca. 10 %. Die erwartete, stärkere Adaption an das Material des aktuellen Diskurses ist nicht zu erkennen. Wie in Abb. 6.2 ergibt sich zwar ein ruhigerer Anstieg der Erkennungsrate, jedoch keine sichtbare Beschleunigung der Adaption.

6.4 Zusammenfassung

Durch den beschriebenen Kontroll-Algorithmus lassen sich keine nachweislichen Verbesserungen an den Eckdaten der Adaption (Geschwindigkeit, absoluter Gewinn) erzielen. Die gemessenen Verläufe der Erkennungsrate schwanken jedoch deutlich weniger, wenn Wörter zurückgewiesen werden, deren Auswertung zur Adaption zu einer drastischen Verminderung der mittleren EWKen im FIFO führt. Da diese Schwankungen aber lediglich während der Adaptionsphase auftreten, bis das System auf den neuen Sprecher konvergiert ist, lohnt sich der

Abbildung 6.3: Verlauf der Erkennungsrate in Versuch 2 zur Adaption von Phonem-HMM mit (gestrichelt) und ohne Kontrolle (durchgezogen)

relativ hohe technische Aufwand des Kontroll-Algorithmus wahrscheinlich nicht oder nur in speziellen Fällen, in denen ein 'glatter' Adaptionsverlauf wichtig ist.

Kapitel 7

Kombinierte Adaption

Zusammenfassung

Eine Verbesserung der Adaption durch Kombination verschiedener Verfahren ist nicht garantiert, da die meisten Verfahren sich auf die Systemparameter anderer Verarbeitungsstufen stützen. Exemplarisch wird die gleichzeitige Adaption von Codebuch Prototypen und Mixturverteilungen experimentell untersucht. Es zeigt sich, daß bei gleichzeitiger Anwendung der Verfahren kein, bei stufenweiser Anwendung ein leichter Gewinn zu erzielen ist.

7.1 Zulässigkeit der Kombination

Eine gleichzeitige Adaption in mehreren Verarbeitungsstufen eines Systems zur ASE wirft zunächst die Frage auf, ob sich daraus keine negativen Beeinflussungen zwischen den zwar verschiedenen, aber voneinander abhängigen Parametern ergeben. Ein gleichzeitiges Nachtraining von Klassifikator und Lexikonstufe z.B. könnte dazu führen, daß Modelle ihre semantische Bedeutung vertauschen, was zwar zunächst keinen spürbaren Effekt auf die Erkennungsraten hat, aber eventuell nicht auflösbare Ambiguitäten im Aussprache-Lexikon zur Folge hat. Im folgenden soll exemplarisch der Versuch unternommen werden, die Verfahren der Kapitel 3 und 4 gleichzeitig anzuwenden.

In [Pla92] und [Pla91] in Verbindung mit [Jua85] wird nachgewiesen, daß ein gleichzeitiges Nachtraining von Codebuch- und Mixtur-Koeffizienten zulässig ist und konvergiert. Die entsprechenden Formeln (z.B. [Pla92]) für die Nachschätzung der Parameter sind mittelwertsbildend und nur von den Rückschlußwahrscheinlichkeiten $p(\underline{s}_k(m)|\underline{k})$ des beobachteten Merkmalsvektors $\underline{k}$ auf die Codebuch-Prototypen $\underline{s}_k(m)$ abhängig, wobei diese mit den 'alten' Parametern, also den Parametern vor der Neuabschätzung berechnet werden. Wie in den Abschnitten 3.5 und 4.4.1 gezeigt, lassen sich sowohl die Strategie SCHWAM als auch die Strategie ADDMIX auf mittelwertsbildende Formeln zurückführen, wobei die Gewichte der gemittelten Daten allerdings in Abhängigkeit der Adaptionsstärke $g(w)$ mit der Zeit exponentiell abnehmen.

Das Nachtraining in obigen Literaturstellen bezieht sich jedoch auf eine große Datenmenge, die iterativ zur Maximierung der Likelihood des Gesamtsystems

Codebuch/SCHMM verarbeitet wird.

Bei der raschen, unüberwachten Sprecheradaption sind die Randbedingungen etwas andere: Eine größere Datenmenge von Sprachmaterial des neuen Sprechers ist nicht vorhanden. Dennoch soll bereits nach einem gesprochenen Wort oder Satz mit der Adaption begonnen werden. Eine Sammlung von Sprachdaten des neuen Sprechers ist mit Hilfe der Mechanismen aus Kapitel 6 (FIFO) zwar möglich, allerdings werden dann gerade die zur raschen Adaption entwickelten Algorithmen, welche ihre Information direkt aus dem Erkennungsprozeß beziehen, hinfällig, wenn nach jeder gesprochenen Äusserung ein konventionelles, iteratives Nachtraining gestartet wird.

Darüber hinaus ist bei der unüberwachten Adaption das tatsächlich gesprochene Wort und damit die Folge der semantisch richtigen Phonem–SCHMM nicht bekannt. Dies ist aber eigentlich eine der Voraussetzungen in obigen Literaturstellen, damit das Nachtraining zulässig ist.

Aus diesen Betrachtungen folgt, daß für eine kombinierte Adaption von Codebuch-Parametern und Mixturen unter den harten Bedingungen der raschen, unüberwachten Adaption der Erfolg *nicht* garantiert werden kann.

7.2 Kombination von VERSCH und ADDMIX

7.2.1 Gleichzeitige Adaption

Das Simulationssystem zur Adaption von Phonem-HMM aus Kapitel 4 wird so erweitert, daß gleichzeitig mit der Nachschätzung der Mixturen auch die Prototypen der Codebücher adaptiert werden.

Für die Adaption der Codebücher wird die Strategie VERSCH (siehe Abschnitt 3.4.3) in zwei Adaptionsphasen (vgl. analog Abschnitt 3.6.1) gewählt. In der ersten Adaptionsphase (Phase I) ist die Adaptionsstärke $A_{kCB} = 0.15$, danach ist $A_{kCB} = 0.005$. Die erste Adaptionsphase endet in diesem Versuch bereits nach den ersten 5 Adaptionswörtern. Der Grund dafür war die Beobachtung, daß bei Codebuch-Adaption in Verbindung mit Phonem-Modellen eine Sättigung sehr viel rascher eintritt als in Verbindung mit Ganzwort-Modellen.

Die Nachschätzung der Mixturen in den beteiligten Phonem-Modellen erfolgt nach der Strategie ADDMIX (siehe Abschnitt 4.4.1). Die Adaptionstärke ist dabei konstant $A_{kMIX} = 0.02$.

Die genannten Parameter sind das Ergebnis mehrerer Optimierungen der verschiedenen Parameter. Sie stellen nicht garantiert die bestmögliche Kombination von Parametern dar, aber gewiß ein lokales Optimum. Die übrigen Versuchsbedingungen entsprechen den regulären Versuchen in Kapitel 4. Es erfolgt keine Kontrolle der Adaption nach Kapitel 6.

Abbildung 7.1 zeigt den Verlauf der Erkennungsrate mit kombinierter Adaption (gestrichelter Graph), im Vergleich dazu den Verlauf unter gleichen Bedingungen mit Adaption der Mixturen allein (ADDMIX, durchgezogener Graph). Deutlich ist zu erkennen, daß sich keine Verbesserung durch Kombination der beiden Verfahren erzielen läßt. Im Gegenteil ist der mittlere Gewinn der Erkennungsrate bei Adaption der Mixturen allein größer als bei kombinierter Adaption. Der gleiche Effekt zeigt sich auch gegenüber der alleinigen Adaption von Codebüchern, sowie bei anderen Adaptionsstrategien. Auf die Darstellung der Ergebnisse wird hier verzichtet.

Abbildung 7.1: Erkennungsrate in Abhängigkeit von Adaptionswörtern mit kombinierter Adaption

Anschaulich läßt sich dieses Ergebnis folgendermaßen erklären: Sowohl die Adaption des Codebuchs als auch die Adaption der SCHMM sind Operationen, die sich auf die Parameter des jeweils anderen Systemteils stützen. Um Protoypen des Codebuchs in günstigere Positionen zu bewegen, muß zunächst mit Hilfe der Mixtur-Koeffizienten des zugeordneten SCHMM genau diese Position bestimmt werden. Im umgekehrten Fall werden Mixturen mit Zugehörigkeitsfunktionen bezogen auf das Codebuch nachtrainiert. Die gleichzeitige, nichtiterative[1] Manipulation von Codebuch und Mixturen entzieht jedoch den beiden Teilsystemen ihre jeweilige Orientierung. Anders liegt der Fall, wenn mit sehr großer Datenmenge und iterativ trainiert wird. Die Folge ist eine sehr kleine Adaptionsstärke und gleiche Gewichtung für alle Beobachtungen in jeder Iteration. Dann spricht man aber besser von einem Neu-Training des Gesamtsystems und nicht von Sprecheradaption.

7.2.2 Stufenweise Adaption

Um obige Vermutung zu verifizieren, wird das Simulationsprogramm so modifiziert, daß die Adaption von Codebüchern und Mixturen nicht mehr gleichzeitig, sondern getrennt hintereinander ablaufen.

Bild 7.2 zeigt den Verlauf der Erkennungsrate (gestrichelter Graph) im Vergleich mit der alleinigen Adaption der Mixtur-Koeffizienten (durchgezogener Graph). Die beobachteten Erkennungsraten verhalten sich nun deutlich besser als in Bild 7.1. Gegenüber der alleinigen Adaption der Mixturen ist ein leichter Gewinn und ein etwas ruhigerer Verlauf zu beobachten.

[1] d.h. die Iteration besteht nur aus einem Schritt

Abbildung 7.2: Erkennungsrate in Abhängigkeit von Adaptionswörtern mit stufenweiser kombinierter Adaption (gestrichelt)

Der gegenseitige Einfluß der beiden Adaptionsalgorithmen ist zwar verhindert. Eine gemeinsame Optimierung ist aber dadurch nicht mehr möglich. Ein weiterer Nachteil ist die Tatsache, daß das System über einen Sprecherwechsel informiert werden muß (vgl. Kapitel 8).

Kapitel 8

Sprecherwechsel

Zusammenfassung

Es sind drei mögliche Szenarios für einen Wechsel des Sprechers denkbar: *obligatorische Anmeldung, automatische Detektion* oder *Unabhängigkeit* vom Sprecherwechsel. Es wird anhand von Simulationen gezeigt, daß – abgesehen von einer Ausnahme – alle untersuchten Algorithmen unabhängig von einem jederzeit möglichen Wechsel des Benutzers arbeiten. Es scheint daher nicht sinnvoll, das System mit einer automatischen Detektion des Sprecherwechsels auszustatten.

8.1 Problem des Sprecherwechsels

Jedes Spracherkennungssystem wird in der Praxis – bis auf ganz wenige Ausnahmen – von mehr als einem Sprecher genutzt werden. Um das dabei entstehende technische Problem des Sprecherwechsels zu beurteilen, müssen eine Vielzahl von Randbedingungen berücksichtigt werden, z.B.:

- Welche Sprechergruppen (Geschlecht, Alter, Ausbildungsstand, etc.) werden das System benutzen ?
- Ist eine explizite Anmeldung zumutbar ?
  (z.B.: Auskunftsystem: nein, Diktiergerät: ja)
- Wie lange wird ein Sprecher das System mindestens benutzen ? (Adaptionsdauer)
- Steht zu jeder Zeit genügend Rechenleistung für die Adaption zur Verfügung ?
- Soll das System vor jedem Sprecherwechsel initialisiert werden ?

In diesem Kapitel sollen drei der wichtigsten Szenarios diskutiert werden, die *obligatorische Anmeldung* eines neuen Sprechers, die *automatische Detektion* bzw. Wiedererkennung eines Sprechers und die völlige *Unabhängigkeit* des Systems von einem Sprecherwechsel.

8.2 Obligatorische Anmeldung

Ein neuer Sprecher gibt seine Identität explizit bekannt, z.B. durch Einloggen am System. Dies bedeutet den optimalen Fall für das Spracherkennungssystem. Es kann die Information über den Sprecher nutzen, indem es z.B.

- die Systemparameter (Codebücher, HMM, RWS-Matrizen, etc.) initialisiert.
- eine Adaptionsphase einleitet.
- den Sprecher 'wiedererkennt' und dessen bereits abgespeicherte Parametersätze verwendet.
- den neuen Benutzer für eine bestimmte Zeit um Unterstützung bittet (z.B. durch halbüberwachte Adaption).

Wann immer die äußeren Bedingungen eine solche Anmeldung bzw. Verwaltung von Benutzern zuläßt, sollte sie auch durchgeführt werden. Andererseits sind viele Realisierungen denkbar, bei welchen eine obligatorische Anmeldung unmöglich oder zumindest benutzerfeindlich wäre, z.B. in Auskunftsystemen, Systemen mit begrenzter Speichermöglichkeit und/oder Rechenleistung.

8.3 Automatische Detektion

Eine einigermaßen sichere, automatische Detektion eines Sprecherwechsels bzw. auch eine Sprechererkennung könnte die gleichen Aufgaben erfüllen wie die obligatorische Anmeldung. Allerdings ergibt sich dabei das folgende Problem.

Um einen Sprecherwechsel zu detektieren (wobei das Nicht-Anzeigen eines solchen Wechsels praktisch ausgeschlossen sein muß), braucht das System ein gewisses Minimum an Material des neuen Sprechers. Mit anderen Worten, die Detektion wird ziemlich stark verspätet einsetzen. Das ist aber sicher nicht benutzerfreundlich, da in dieser Zeit mit schlechten Erkennungsraten zu rechnen ist. Eine explizite Anmeldung, z.B. auch nur durch einen Tastendruck, ist daher der automatischen Detektion vorzuziehen.

Dennoch werden in der Literatur einige Vorschläge für die Klassifikation oder Erkennung von bestimmten Sprechern oder Sprecher-Typen gemacht (z.B. [Lee89],[Mat90]). Der Hauptgrund dafür ist der Wunsch, nicht Rechenzeit für Adaptionen verschwenden zu müssen, die man schon einmal vorgenommen hat.

8.4 Unabhängigkeit vom Sprecherwechsel

Die eleganteste Lösung des Problems wären Adaptionsalgorithmen, die auf die Information, daß ein neuer Sprecher das System benutzen will, vollkommen verzichten können. In diesem Fall konzentriert sich auch der Rechenaufwand auf die eigentliche Adaption und wird nicht für Verwaltungsaufgaben bzw. Sprecherdetektion verwendet. Ein weiterer Vorteil ist, daß auch sprecherspezifische Veränderungen der Sprachquelle mit berücksichtigt werden, ohne daß diese angezeigt werden müssen.

Abbildung 8.1: Adaption von Codebüchern: Sprecherwechsel von Sprecher ZN auf TM nach 100 Adaptionswörtern

Im folgenden soll gezeigt werden, daß die wichtigsten in den Kapiteln 3 und 4 beschriebenen Adaptionsalgorithmen unabhängig von einem Sprecherwechsel zufriedenstellend arbeiten. Die Situation sei dabei folgende: ein bestimmter Sprecher oder eine Sprecherin benutzt das System bereits längere Zeit (100 - 200 Adaptionswörter), d.h. die System-Parameter (Codebuch-Prototypen, Mixtur-Koeffizienten) sind auf ihn/sie adaptiert. Dann wechselt der Sprecher, und zwar einmal auf einen gleichgeschlechtlichen, einmal auf einen Sprecher des anderen Geschlechts. Das System erhält keine Informationen über diesen Wechsel. Das heißt insbesondere, daß es sich nicht neu initialisieren, also z.B. sprecherunabhängige System-Parameter laden kann. Die Adaptionsalgorithmen arbeiten mit gleicher Adaptionsstärke, etc. weiter wie bisher. Das System adaptiert nun vom ursprünglichen Sprecher auf den neuen Sprecher, ohne sich dessen aber 'bewußt' zu werden.

8.4.1 Sprecherwechsel bei Codebuch-Adaption

Das Simulationssystem aus Abschnitt 3 wird dahingehend erweitert, daß nach 100 verarbeiteten Adaptionswörtern eines ersten Sprechers die Sprachdatenbasis auf einen weiteren Sprecher umgeschaltet werden kann. Die Umschaltung betrifft nur die Sprachdatenbasis, d.h. sämtliche übrigen Parameter des Spracherkenners bleiben unverändert. Die Versuchsbedingungen entsprechen denen von Versuch 6 in Abschnitt 3 (vgl. 3.4.7). Alle folgenden Versuche werden mit Parametern für lange Adaptionsphasen (100 Wörter) durchgeführt.

Bild 8.1 zeigt die Sprecherkombination männlich-männlich, d.h. nach 100 Adaptionsworten des Sprechers ZN werden dem System Wörter des Sprechers

Abbildung 8.2: Adaption von Codebüchern: Sprecherwechsel von Sprecherin BR auf TM nach 100 Adaptionswörtern

TM angeboten. Im Moment des Sprecherwechsels (senkrechte, gestrichelte Linie) kommt es zu einem drastischen Einbruch in der Erkennungsrate. Allerdings liegt das Minimum immer noch deutlich über dem Wert, welchen der Sprecher TM mit unadaptierten Codebüchern erzielt (72 %). Dann erfolgt die Adaption an den neuen Sprecher bis Werte um etwa 79 % erreicht werden. Auch dies ist ein besserer Wert als bei der Adaption des Sprechers TM allein (77.5 %). Offensichtlich sind die Codebücher durch die vorangegangene Adaption an den Sprecher ZN bereits 'männlich vorgeprägt', weshalb der Übergang auf den ebenfalls männlichen Sprecher TM leicht fällt.

Bild 8.2 zeigt den Sprecherwechsel in der Kombination weiblich-männlich. Nach 100 Adaptionswörtern der Sprecherin BR wird auf den Sprecher TM gewechselt. Da die Sprecherin BR sich nur um einige Prozent verbessern kann, kommt es beim Wechsel zu einem Sprung der Erkennungsraten in umgekehrter Richtung. Interessant ist, daß bereits der erste gemessene Wert des Sprechers TM nicht schlechter liegt als beim Test mit unadaptierten Codebüchern (72 %). Die Codebücher sind also durch die Adaptionsphase mit der Sprecherin BR für einen männlichen Sprecher nicht 'verschlechtert' worden. Es ergeben sich in diesem Fall keine Einbußen beim Übergang auf einen fremdgeschlechtlichen Sprecher. Dies ist allerdings nicht immer der Fall: beim Wechsel von Sprecher TM auf Sprecherin BR ist z.B. eine anfängliche Einbuße von ca. 2 % zu beobachten, der nach längerer Adaptionsphase wieder ausgeglichen werden kann.

Die Kombination von zwei Adaptionsphasen wie in Abschnitt 3.6.1 erfordert natürlich Information über einen Sprecherwechsel, um die 'starke Adaptionsphase' einzuleiten. Auf die entsprechenden Simulationen wird daher verzichtet.

Abbildung 8.3: Adaption von SCHMM: Sprecherwechsel von Sprecher TM auf ZN (durchgezogen) und BR (gestrichelt) nach 200 Adaptionswörtern

8.4.2 Sprecherwechsel bei Adaption der SCHMM

Das Simulationssystem aus Abschnitt 4 wird dahingehend erweitert, daß nach 200 verarbeiteten Adaptionswörtern eines ersten Sprechers die Sprachdatenbasis auf einen weiteren Sprecher umgeschaltet werden kann. Die Umschaltung betrifft nur die Sprachdatenbasis, d.h. die sämtlichen übrigen Parameter des Spracherkenners bleiben unverändert. Bild 8.3 zeigt den Verlauf der Erkennungsraten für die Kombination männlich-männlich (durchgezogener Graph) und männlich-weiblich (gestrichelter Graph). Zu Beginn ist das System wie üblich mit sprecherunabhängigen Parametern initialisiert. Die Versuchsbedingungen entsprechen denen von Versuch 8 in Abschnitt 4 (vgl. 4.4.9). Der Sprecherwechsel erfolgt jeweils nach 200 Wörtern des Sprechers TM (senkrechte Linie).

In beiden Fällen kommt es zunächst zu einem scharfen Einbruch in den Erkennungsraten. Beim Wechsel auf den männlichen Sprecher (ZN) entspricht dies ungefähr der Erkennungsrate des nicht adaptierten, sprecherunabhängigen Spracherkenners (vgl. Bild 4.6). Danach adaptiert sich das System relativ rasch an den Sprecher ZN und erreicht sogar deutlich bessere Werte als bei der Adaption an den sprecherunabhängigen Erkenner (vgl. Bild 4.6). Beim Wechsel auf die Sprecherin (BR) sinken die Erkennungsraten noch stärker ab und erreichen nach 200 Wörtern nicht ganz die Resultate der Adaption eines sprecherunabhängigen Systems.

Diese Beobachtungen decken sich mit den Erwartungen: Das System ist nach 200 Adaptionswörtern an den männlichen Sprecher TM adaptiert, daher gelingt der Wechsel auf einen anderen männlichen Sprecher natürlich leichter als auf einen weiblichen. Darüber hinaus zeigt sich, daß die Adaption an den ersten

männlichen Sprecher sich positiv auf die Werte des folgenden Sprechers auswirken, denn er erreicht bessere Werte als bei Adaption an das sprecherunabhängige System. Aber auch der schwierigere Übergang von männlich auf weiblich zeigt schon nach ca. 100 Wörtern einen Gewinn von ca. 15 %.

8.5 Zusammenfassung

Sowohl die Adaption der Codebücher als auch der Mixtur-Verteilungen in den SCHMM läßt sich ohne automatische oder explizite Anmeldung eines neuen Benutzers durchführen, ausgenommen die zweistufige Adaption von Codebüchern aus Abschnitt 3.6.1. Die Adaption der Bewertung von Lautsymbolen (s. Kapitel 5) benötigt ebenfalls keinerlei Information über den aktuellen Benutzer. Eine automatische Detektion eines Sprecherwechsels erübrigt sich demnach für die vorgestellten Verfahren.

Kapitel 9

Adaption in symbolischen Verarbeitungsstufen

Zusammenfassung

Kurzer Ausblick auf die weitere Entwicklung im Gebiet der Sprecheradaption: Adaption von sprechertypischen Aussprachevarianten, Dialekterkennung, Adaption von Sprachmodellen, sprecheradaptive Dialogführung. Für die Adaption von sprechertypischen Aussprachevarianten wird ein detailierter Vorschlag unterbreitet.

9.1 Ausblick

Es ist zu erwarten, daß sich die Aktivitäten auf dem Gebiet der Sprecheradaption in den nächsten Jahren auf 'höhere', meist symbolische Verarbeitungsstufen in Systemen zur ASE verlagern werden, nachdem in den klassischen Bereichen der Signalverarbeitung und Mustererkennung schon außerordentlich viele erfolgreiche Methoden eingesetzt werden. Vor allem im Bereich der spontansprachlichen Erkennung, z.B. Dialogsysteme zur Datenbankabfrage etc., treten zahlreiche Probleme auf, weil fast jeder Sprecher über die rein akustische Diversität hinaus auch andere sprechertypische Eigenschaften aufweist, wie z.B. die Aussprache bestimmter Wörter bzw. Wortformen, syntaktische Irregularitäten, Satzbildung, Dialekt, Dialogstrategie, usw. Als Beispiel sei auf die Untersuchungen in [Kra91] verwiesen.

Bisher war der Bereich der ASE der reinen Hochsprache vorbehalten, d.h. es wurde immer eine korrekt formulierte, syntaktisch einwandfreie Spracheingabe vorausgesetzt. Daher sind bis dato Anstrengungen in Richtung der o.g. Probleme vernachläßigt worden. Für den kommerziellen Einsatz solcher ASE Systeme kann jedoch nicht mit Laborbedingungen gerechnet werden, daher müssen für den Feldeinsatz wahrscheinlich schon sehr bald erweiterte Lösungen bereitgestellt werden.

Im folgenden werden Vorschläge für die Lösung zumindest einiger der o.g. Probleme vorgestellt. Die Darstellung kann naturgemäß nicht umfassend sein, da zum einen eine gründliche Behandlung den Rahmen dieser Arbeit bei weitem sprengen würde, zum anderen die davon betroffenen Forschungsgebiete am

Lehrstuhl für Datenverarbeitung der Technischen Universität München nicht bearbeitet werden. Für die Adaption von sprechertypischen Aussprachevarianten wird jedoch ein detailierter Vorschlag unterbreitet, da diese Verarbeitungsstufe noch direkt die Bewertung von Worthypothesen betrifft.

9.2 Sprechertypische Aussprachevarianten

Unter einer Aussprachevariante eines Wortes soll hier keine allophonische Variante verstanden werden, sondern die Vertauschung, Auslassung oder Einfügung von bedeutungstragenden Einheiten, also Phonemen (analog zu [Jek89]). Die Grenze zwischen allophonischer Variation eines Phonems und einer Aussprachevariante sei der Einfachheit halber durch das verwendete ASE System gegeben: Sobald der Klassifikator aufgrund einer Verschleifung im Sprachsignal eine von der kanonischen Form abweichende Transkription ermittelt, bezeichnen wir dies als Aussprachevariante; der Rest seien allophonische Variationen.

9.2.1 Problematik

Jedes System zur ASE mit großem Wortschatz (2000 - 20000 Wörter) benötigt Informationen über die Aussprache von lexikalen Einheiten. Im allgemeinen handelt es sich dabei entweder um Vollformen oder Stammformen mit entsprechenden Suffices, die meist in Form einer phonetischen Umschrift in einem sog. Aussprache-Lexikon[1] gesammelt vorliegen. Während des Erkennungsprozesses werden anhand dieser Einträge bestimmten Segmenten des Sprachsignals Worthypothesen mit einer entsprechenden akustischen Bewertung zugewiesen.

Meistens geschieht dies, indem anhand der phonetischen Umschrift im Lexikon statistische Modelle für Wortuntereinheiten (z.B. Phoneme, Phonem-Cluster, Halbsilben, Silben) verkettet und deren gesamte Rückschlußwahrscheinlichkeit auf den sprachlichen Input als Bewertung berechnet wird (sog. 'top down' Ansatz). Der andere Weg ist der Vergleich einer ohne Restriktionen ermittelten phonetischen Transkription des Sprachsignals mit der phonetischen Umschrift eines lexikalen Eintrags im Lexikon und Bewertung anhand der Rückschlußwahrscheinlichkeiten der dabei gebildeten Einheiten-Paare ('bottom up' Ansatz, vgl. Kapitel 5). Für beide Ansätze ist die korrekte phonetische Umschrift im Lexikon essentiell. Das Lexikon enthält entweder nur die kanonische Form, wie z.B. im ASE System *SPICOS* (z.B. [Pae89/1]) oder mehrere, ev. auch gewichtete Ausprache-Varianten für jedes Wort, wie z.B. im ASE System *SILBOS 3.0* ([Wei90]).

Im allgemeinen sprechen Menschen keine Hochsprache (ausgenommen geschulte Sprecher, wie z.B. in Rundfunk und Fernsehen). Daher kann in fast allen Fällen beobachtet werden, daß ein Sprecher bestimmte Wörter auf eine ihm mehr oder weniger charakteristische Weise ausspricht. Eine solche, von der kanonischen Form abweichende Aussprache ist durchaus reproduzierbar und somit nicht schwerer zu erkennen als die 'korrekte' Aussprache[2].

Dies läßt sich z.B. mit Beobachtungen beim Einsatz von sog. selbsttrainierbaren Systemen zur Erkennung von Einzelwörtern belegen: Ein sprachgesteuertes Lagerhaltungssystem der Computer Gesellschaft Konstanz (CGK) verlangt von

[1] Im folgenden wird nur noch der in diesem Kontext eindeutige Begriff Lexikon verwendet
[2] 'korrekt' im Sinne z.B. des Aussprache-Wörterbuchs [Dud90]

jedem Benutzer zunächst die mehrfache Eingabe aller erkennbaren Kommandos. Anhand der dabei erstellten Muster werden sprecherabhängig Befehle zur Lagerbuchhaltung erkannt, während der Angestellte mit beiden Händen manuelle Tätigkeiten ausführen kann. Es wurde nun beobachtet, daß die vorgesehenen Kommandos von den verschiedenen Angestellten je nach Dialekt, sozialem Status oder auch Akzent sehr unterschiedlich ausgesprochen wurden. Dies minderte die Erkennungsleistung dieses Systems natürlich nicht, da es auf solche Charakteristika durch das vorangegangene Training eingestellt war. Ein sprecherunabhängiges System mit einem kanonischen Lexikon hätte in diesem Fall mit hoher Wahrscheinlichkeit völlig versagt.

Das o.g. Beispiel stellt natürlich einen Grenzfall dar. Dennoch wird ein ASE System, das nur über die kanonische Form in seinem Lexikon verfügt, zu teils erheblichen Fehlbewertungen kommen, die u.U. die Erkennung der Äußerung vereiteln. Sogar dann, wenn man davon ausgeht, daß der Benutzer kooperativ ist und sich bemüht, deutlich zu sprechen.

In [Wei90], [Wik90] wurde ein sprecherspezifisches Training von Aussprache-Modellen untersucht, welches anhand der Trainingsdaten eines sprecherabhängigen ASE Systems (*SILBOS 3.0*) das Lexikon auf den Referenzsprecher trainiert. Dabei werden die Aussprache-Varianten, welche das System bei der Erkennung der Trainingsdaten erzeugt, explizit beobachtet und in einfache Finite-State-Graphen eingetragen. Diese werden dann bei der Erkennung mit Hilfe der 1-stufigen dynamischen Programmierung abgearbeitet, so daß immer die günstigste Aussprache-Variante zur Bewertung herangezogen wird.

Auch in [Rig90] wird ein Verfahren der Adaption von Wortmodellen vorgestellt, in welchem durch explizite Auswertung des Viterbi-Pfades während der Erkennung Aussprache-Varianten[3] gelernt werden.

Leider sind solche Verfahren nur praktikabel, wenn entweder das verwendete Lexikon sehr klein ist oder bereits so viel Material eines Sprechers vorliegt, daß jeder lexikale Eintrag mindestens 5 - 10mal beobachtet werden kann. Dies ist für große Lexika natürlich nicht durchführbar.

9.2.2 Adaption mit Hilfe von Verschleifungsregeln

Es wird daher vorgeschlagen, die sprechertypischen Aussprachevarianten des aktuellen Benutzers nicht durch explizite Beobachtung, sondern durch das Lernen von Regeln zu erfassen. Die Regeln sollten so allgemein definiert sein, daß mit ihrer Hilfe ein kanonisches Lexikon in ein sprechertypisches Lexikon transformiert werden kann.

Das Lernen von Regeln findet grundsätzlich in mindestens zwei Schritten statt. Der erste Schritt ist die Beobachtung eines oder mehrerer gleicher Phänomene; der zweite Schritt die Formulierung einer Regel, welche das Phänomen in einer abstrakten, deskriptiven Form beschreibt und für irgendeine bestimmte Anwendung einsetzbar ist, also meist einer festen Syntax gehorcht ([Beh91], [Zue86]).

Ein Beispiel dafür wäre die Beobachtung, daß ein bestimmter Sprecher fast immer für das Wort 'gehen' die Ausspracheform /ge:N/ statt /ge:h@n/ verwen-

[3] Allerdings nicht in Form von phonetischen Symbolen, sondern als Folge von sog. 'phenonic baseforms'

det, und die Formulierung einer Regel[4]

$$/ge:h@n/ \Rightarrow /ge:N/ \tag{9.1}$$

Ein solcher Fall ist völlig unproblematisch, aber auch recht nutzlos, da die formulierte Regel keinerlei Generalisierung enthält. Gerade die Generalisierung auf nicht beobachtete Fälle ist es aber, was man von einer 'echten' Regel erwartet. Man könnte daher die Regel 9.1 umformulieren in

$$/h@n\#/ \Rightarrow /N\#/ \tag{9.2}$$

Wobei die gewählte Notation bedeuten soll:

> „Ersetze in jedem Wort den Phonem-Komplex /h@n/ am Wortende (#) in das Phonem /N/ am Wortende."

Die Frage ist nur: Ist eine solche Generalisierung auf alle Wörter, die auf /h@n/ enden, korrekt ? Spätestens jetzt kommt ein dritter Schritt des Lernvorgangs hinzu, nämlich die Verifikation der erlernten Regeln.

Für die Verifikation der formulierten Regel gibt es zwei Möglichkeiten. Entweder die Regel wird durch eine umfassende statistische Analyse einer repräsentativen Stichprobe belegt, oder man läßt einen Experten entscheiden, ob die Regel sinnvoll ist oder nicht. Beide Fälle sind für die automatische Sprecheradaption gleichermaßen ungeeignet. Weder eine repräsentative Stichprobe des Sprachmaterials des unbekannten Sprechers noch ein Experte steht zur Verfügung.

Der einzig sinnvolle Ausweg aus diesem Dilemma ist die Vorformulierung potenziell sprechertypischer Regeln, welche durch eine der o.g. Methoden anhand von umfangreichem Sprachmaterial verifiziert worden sind. Unter 'potenziell sprechertypischer Regel' sei eine Regel gemeint, die für einen bestimmten Sprecher typisch sein *könnte*. Die Aufgabe beschränkt sich dann nur noch auf die Zuordnung dieser Regeln zum Sprechverhalten des aktuellen Benutzers.

9.2.3 Potenziell sprechertypische Verschleifungsregeln

Ausgehend vom einem Korpus von 83 Regeln aus [Jek89] werden insgesamt 137 allgemeine Verschleifungsregeln formuliert. Eine detailierte Zusammenstellung und Beschreibung entnehme man [Wof91]. Eine Untersuchung der 996 häufigsten deutschen Wörter nach [Kae98] ergibt für einen lexikalen Eintrag im Mittel 2.3 Aussprachevarianten; die maximale Anzahl von Varianten für ein Wort beträgt 36 (siehe auch [Sch91/1]). Die Trennfähigkeit des Lexikons wird durch das Hinzufügen aller erzeugten Aussprachevarianten zwar vermindert, aber es treten keine gleichlautenden Transkriptionen verschiedener lexikaler Einträge auf. D.h., theoretisch sind noch alle Einträge anhand der phonetischen Umschrift voneinander trennbar.

Die erzeugten Aussprachevarianten wurden von Hand überprüft. Es läßt sich feststellen, daß die überwiegende Anzahl der Varianten sinnvolle Aussprachemöglichkeiten darstellen. Nur in vereinzelten Fällen entstehen extreme, nicht nachvollziehbare Verschleifungen. wie z.B. im lexikalen Eintrag /g@n@Ra:l/, für das aufgrund der Regel /@n@/ ⇒ /n@/ die Variante /gn@Ra:l/ erzeugt wurde.

[4]Zur Notation vgl. Anhang A

9.2.4 Identifikation von sprechertypischen Regeln

Das ASE System beginnt die Sitzung mit einem unbekannten Sprecher zunächst unter Verwendung eines Lexikons, das alle durch die o.g. Regeln erzeugbaren Aussprachevarianten enthält. Im Lexikon sind zu jeder dieser Varianten alle Regeln eingetragen, welche für die Erzeugung aus der kanonischen Form verantwortlich sind. Während der Sitzung wird durch Backtracking oder ein ähnliches Verfahren die Kette von Aussprachevarianten ermittelt, die letztlich zur Erkennung der gesprochenen Äußerung geführt hat. Dadurch kann eine schritthaltende Statistik aufgebaut werden, welche besagt, wie häufig eine Regel vom aktuellen Sprecher 'benutzt'[5] wird.

Anhand dieser Statistik kann nunmehr entschieden werden, ob und wie das Lexikon dem Sprachgebrauch des Benutzers angepaßt wird. Z.B. kann mit dem Satz beobachteter Regeln ein Lexikon mit sprechertypischen Aussprachevarianten erzeugt werden, wenn nach einer bestimmten Zeit keine neuen Regeln mehr beobachtet werden. Die Trennbarkeit des Lexikons wird dadurch wieder erhöht und die Erkennungsleistung des ASE Systems steigt (vgl. dazu auch [Sch92/2]).

Das hier vorgeschlagene Konzept steht und fällt mit der geeigneten Formulierung des initialen Regelsatzes. Die in den bisherigen Untersuchungen verwendeten Regeln stellen sicher nicht den Idealfall dar, da sie aus einem Korpus von Regeln hervorgingen, welcher z.B. keine Dialektbildung berücksichtigte. Dennoch läßt sich an ihnen die Leistungsfähigkeit des Ansatzes demonstrieren.

9.2.5 Adaption auf Dialekte

Ein ähnliches Konzept wie im vorangegangenen Abschnitt ist auch für die Erkennung und Berücksichtigung von dialektalen Verschleifungen denkbar. Dazu wären jedoch umfangreiche Untersuchungen anhand von dialektgefärbtem Material nötig, um Regelsätze für bestimmte Dialekte zu erstellen. Anhand solcher dialektspezifischer Regelsätze könnte nach der o.g. Methode die Zugehörigkeit des beobachteten Sprechers zu einem bestimmten Dialekt ermittelt und das Lexikon mit Hilfe des gleichen Regelsatzes adaptiert werden.

9.3 Sprecheradaptives Sprachmodell

Fast alle derzeit entwickelten Systeme zur ASE verwenden zur Erkennung von Wortketten ein sog. Sprachmodell ('language model'). Es kann sich dabei um ein statistisches oder deterministisches Modell handeln, welches Bewertungen für das Auftreten von Einzelworten (Unigramm-Modell), Paaren von Wörtern (Bigramm-Modell), etc. ermöglicht (z.B. [Pae89/2], [Ney91]).

Statistische Sprachmodelle werden i.a. aus einer möglichst großen und für die gesprochene Sprache repräsentativen Stichprobe durch Auszählen ermittelt. Allerdings berücksichtigt ein solches Modell nicht die Tatsache, daß in einem bestimmten Gesprächskontext diese Statistik höchstwahrscheinlich völlig falsch ist.

Nehmen wir an, wir hätten ein ASE System, das mit einem sehr umfassenden Lexikon ausgestattet ist und ein statistisches Bigramm-Modell verwendet,

[5] Ob der Mensch tatsächlich Regeln für die Sprachbildung verwendet, ist natürlich nicht bekannt. Näheres dazu findet sich in [Hof91]

welches tatsächlich aus einer so umfassenden Menge gesprochener Sprache ermittelt wurde, daß jedes Auftreten eines Wortes vernünftig bewertet werden kann. In diesem Sprachmodell wird das Wort 'Rechtsstreit' wahrscheinlich eine sehr niedrige Auftretenswahrscheinlichkeit haben, da es im überwiegenden Teil der Sprache praktisch nie vorkommt. Demzufolge wird dieses Wort bei der Erkennung immer sehr niedrig bewertet werden, auch dann, wenn sich der Dialog mit dem ASE System bereits seit einiger Zeit mit juristischen Problemen befaßt.

Der Mensch paßt sich mit seiner Erkennungsleistung dem jeweiligen Kontext sehr stark an. Das geht so weit, daß teilweise grobe Fehlerkennungen auftreten, weil der Betreffende infolge des Kontexts seinen Erkennungswortschatz eingeschränkt hat bzw. erwartungsgesteuert vorgeht.

Wünschenswert wäre also ein Sprachmodell, welches sich an den aktuellen Diskurs anpaßt. An der McGill University, Montreal wurden Untersuchungen vorgenommen, in welchen mit Hilfe eines sog. 'cache memory' eine feste Anzahl der zuletzt gesprochenen Wörter gespeichert wird ([Kuh90]). Ähnliche Ansätze wurden auch in [Jel91], [Mat92] und [Del92] veröffentlicht. Daraus läßt sich dann ein adaptives Sprachmodell des aktuellen Diskurses berechnen und mit einem allgemeinen Sprachmodell linear interpolieren. Natürlich erfaßt ein solches 'cache memory' nicht nur den aktuellen Diskursbereich, sondern auch die sprechertypische Wortwahl des Sprechers und ist somit im weitesten Sinne auch ein Mittel zur Sprecheradaption.

9.4 Sprecheradaptive Dialogführung

Nicht jeder Benutzer ist den Umgang mit einem streng logisch aufgebauten Informationssystem, wie z.B. einer Datenbankabfrage, so vertraut, daß er auf zahlreiche Hilfestellungen während des Dialogs mit dem System verzichten kann. Applikationen, die für einen Dialog über Terminal (getippte Eingabe) entworfen sind, werden daher heutzutage meistens mit sehr ausführlichen und für den erfahrenen Benutzer ermüdenden Hilfestellungen bzw. Dialogführungen ausgestattet. Einige dieser Systeme ermöglichen es dem Benutzer, diese Hilfestellungen nach und nach einzuschränken und lernen auf diese Weise, welche Hilfestellungen während des Dialogs noch angeboten werden sollen und welche nicht.

Applikationen, die mit automatischer Spracheingabe und -ausgabe arbeiten, müssen dem Benutzer natürlich ebenfalls ausführliche Hilfestellungen anbieten. Solche erklärenden Texte – meistens infolge der Sprachsynthese mit sehr deutlicher und langsamer Sprache vorgetragen – führen bei den meisten Benutzern rasch zu Ungeduld und Langeweile.

Für eine ergonomische Dialogführung ist es daher unumgänglich, daß das System sich allmählich dem Benutzer anpaßt und dessen Erfahrung mit dem System berücksichtigt. Ein erster Schritt in die richtige Richtung ist die inzwischen weit verbreitete Gewohnheit, den Benutzer zu Beginn der Sitzung zu fragen, ob er überhaupt Erklärungen wünscht. Wird dies verneint, verzichtet das Programm auf alle Erklärungen während des Dialogs. An jeder Stelle sollte dann aber die Möglichkeit bestehen, mittels eines bestimmten Schlüsselwortes eine Hilfestellung anzufordern.

Einige Vorschläge für eine sprecheradaptive Dialogführung:

- Adaption auf das Vorwissen.
  Das Programm protokolliert sämtliche bereits gegebenen Erklärungen und

den Zeitpunkt der Erklärung für einen bestimmten Benutzer. Angenommen, der Dialog kommt nun an eine Stelle, an der eine Hilfestellung notwendig ist. Hat der Benutzer diese noch nie gehört, wird sie unaufgefordert gegeben. Im anderen Falle sollte sich die Strategie an der verstrichenen Zeit orientieren. Dabei sind z.B. drei Abstufungen denkbar:

1. Die letzte Erklärung zur Dialogsituation ist sehr lange her (mehr als 1 - 2 Monate): Die Hilfestellung wird unaufgefordert gegeben.
2. Die letzte Erklärung lag im einem mittleren Zeitraum (1 Tag bis 1 Monat zurück): Das System fragt, ob es eine Erklärung geben soll.
3. Die letzte Erklärung erfolgte am selben Tag: Auf eine weitere Hilfestellung wird verzichtet.

Zur weiteren Feinabstimmung der Entscheidung kann auch die Häufigkeit der bereits gegebenen Hilfestellungen dienen.

- Adaption des 'Hilfe-Levels'.
  Zu jedem Dialogpunkt, der einer Erläuterung bedarf, werden mehrere – z.B. drei – verschiedene Hilfestellungen formuliert. Die erste (E1) sei ziemlich ausführlich und erfordert keinerlei Vorkenntnisse, die zweite etwas knapper (E2) und die dritte (E3) bestehe aus einem einzelnen Satz, der nur das Wichtigste enthält, aber z.B. keine Erklärungen über Eingabemodus, etc. Der Benutzer kann jederzeit, auch während laufender Sprachausgabe mittels zweier Schlüsselwörter mehr Hilfestellung anfordern (z.B. „Hilfe“) bzw. frühzeitig abbrechen (z.B. „Danke“). Bei einem unbekannten Benutzer beginnt das System grundsätzlich mit der kürzesten Erklärung E3. Schon nach einigen Dialogschritten wird erkennbar, welcher Hilfe-Level E1, E2 oder E3 für den Benutzer optimal ist. Die Information über den gewünschten Hilfe-Level kann gespeichert und bei zukünftigen Sitzungen berücksichtigt werden. Die weitere Dialogführung kann auf diese Weise optimal auf den jeweiligen Benutzer abgestimmt werden. Dies kann sich auch auf die Antworten oder Rückfragen des Programms erstrecken.

  Ein Beispiel:
  Das Programm *Christina* (Fa. Philips, Erlangen) gibt Auskunft über die neuen Postleitzahlen, indem es die bisherige Postleitzahl in einzelnen Ziffern abfragt. Zur Kontrolle erfolgt nach jeder Zifferneingabe des Benutzers die Ausgabe: „Sagten Sie die Zahlen: acht null null null ?“, worauf der Benutzer mit „ja“ oder „nein“ quittieren muß. Der erfahrene Benutzer kann auf den Zusatz „Sagten Sie die Zahlen:...“ gerne verzichten, da er gelernt hat, daß alle Eingabe bestätigt werden müssen, und solche unnötigen Ausgaben den Dialog nur in die Länge ziehen (und dabei auch erhöhte Telefongebühren verursachen).

- Adaption der Sprechgeschwindigkeit.
  Die meisten Systeme zur Sprachsynthese verwenden eine ungewöhnlich langsame Sprechgeschwindigkeit (z.B. SPRAUS der Fa. AEG, VoxPC 200 der Fa. Infovox), um die Verständlichkeit der synthetischen Stimme zu erhöhen. Andererseits wäre eine höhere Sprechgeschwindigkeit für den Benutzer angenehmer, um den Dialog zu beschleunigen.

Es ist nun oft zu beobachten, daß der Teilnehmer in einem Dialog dazu tendiert, selber langsamer und deutlicher zu sprechen, wenn er den Partner nicht verstanden hat[6] (vgl. z.B. die Versuchsaufnahmen in [Kra91]).

Eine einfache Möglichkeit, diese implizite Information auszunutzen, ist die Adaption der Sprechgeschwindigkeit der Sprachausgabe an die Sprechgeschwindigkeit des Benutzers. Dadurch wird dem Benutzer das Verstehen der Sprachausgabe bei Rückfragen erleichtert und gleichzeitig das angenehme Gefühl vermittelt, das Programm sei für seine Reaktionen sensibel.

Algorithmen zur Ermittlung der Sprechgeschwindigkeit sind relativ einfach zu realisieren. Meistens können dazu Merkmale ausgewertet werden, die bei der Erkennung fließend gesprochener Sprache sowieso anfallen, z.B. eine Auswertung des Energieverlaufs.

Leider sind es oft solche scheinbar unwichtigen Details, die den Dialog mit automatischen Auskunftsystem für viele Menschen lästig machen und sich daher negativ auf die Akzeptanz auswirken.

[6] natürlich auch dann, wenn er das Gefühl hat, der Partner verstehe ihn nicht.

Kapitel 10

Schlußbetrachtung

In der vorliegenden Arbeit wurden mehrere Untersuchungen vorgestellt, die den Einsatz von Methoden der stochastischen Modellierung für die Sprecheradaption in der automatischen Spracherkennung zum Ziel haben. Anhand eines einfachen, aber repräsentativen Spracherkenners basierend auf einem derzeit weit verbreiteten Ansatz mit Hidden Markov Modellen wurde gezeigt, daß das Problem der Sprecheradaption in verschiedenen Repräsentationsstufen auftritt. Dadurch entstand quasi automatisch eine Hierarchie von Methoden und Algorithmen, welche auf die Struktur eines Sprache erkennenden Systems abbildbar ist. Diese Hierarchie spiegelt sich wieder in den Untersuchungen der Kapitel 3, 4, 5 und 9.

Die einzelnen grundlegenden Untersuchungen wurden in der Weise konzipiert und durchgeführt, daß unüberschaubare Seiteneffekte von anderen Verarbeitungsstufen weitgehend ausgeschlossen waren. In den durchgeführten Experimenten und Simulationen sowie bei der Gesamtkonzeption des Spracherkenners wurde großer Wert auf realistische Randbedingungen gelegt, d.h. die 'künstliche' Situation des Laborversuchs sollte vermieden werden.

Als Gesamtergebnis der Untersuchungen läßt sich folgendes sagen:

1. In praktisch allen Verarbeitungsstufen des verwendeten Spracherkenners lassen sich einfache Verfahren angeben, welche die Erkennungsleistung des Gesamtsystems schon nach kurzer Verwendung durch den unbekannten Sprecher deutlich ansteigen lassen.

2. Die Verfahren sind praktisch alle heuristischer Natur, basieren jedoch auf statistischen Grundannahmen. Die verwendete Heuristik entsteht zwangsweise aus der Tatsache, daß bei der speziellen Aufgabe der Sprecheradaption mit Daten gearbeitet werden muß, welche im Sinne der Statistik nicht signifikant sind.

3. Die Leistung dieser Verfahren ist in praktisch allen Fällen abhängig von Parametern, die empirisch ermittelt wurden. Es wurde Wert darauf gelegt, die Anzahl dieser Parameter so klein wie möglich zu halten. In den meisten Verfahren sind die ermittelten Parameter jedoch auf andere Versuchsbedingungen übertragbar.

4. Der Bedarf an Rechenleistung und Speicher der beschriebenen Verfahren

ist bis auf wenige Ausnahmen weniger als 1 % des entsprechenden Bedarfs der Verarbeitungsstufe, welcher das Verfahren zugeordnet ist.

Außerdem wurden folgende Beobachtungen gemacht:

- Der maximal erzielbare Gewinn (Reduktion der relativen Fehlerrate) steht in umgekehrten Verhältnis zur Abnahme des relativen Fehlers mit der Zeit. D.h. je schneller ein Verfahren adaptiert und konvergiert, desto geringer ist der insgeamt erzielte Gewinn durch dieses Verfahren. Dieser Effekt gilt vor allem bei Veränderung der o.g. empirisch ermittelten Parameter.
- Der durch Adaption erzielte Gewinn streut stark unter verschiedenen Testsprechern.
- Die Kombination mehrerer Verfahren ist problematisch und garantiert keineswegs eine kumulative Verbesserung der Erkennungsleistung.

Aus dem bisher Gesagten ergibt sich, daß eine eindeutige Bewertung der Verfahren nicht möglich ist. Es kommt wie in den meisten Fällen darauf an, unter welchen Randbedingungen ein Verfahren eingesetzt werden soll. Als wichtigstes Kriterium sei hier die *Nutzungsdauer* genannt, anders gesagt: die akzeptable Dauer derjenigen Phase, innerhalb derer die Sprecheradaption bereits eine deutliche Verbesserung der Erkennungsleistung bewirkt.

Für einen Spracherkenner mit *kurzer Nutzungsdauer* (20 - 100 Wörter, z.B. Telefonauskunft) sollten die in Kapitel 3 beschriebenen Verfahren zur Adaption von Codebuch-Prototypen eingesetzt werden. Unter diesen hat das kombinierte Verfahren mit Anpassung der Adaptionsstärke die besten Ergebnisse erzielt (vgl. Abschnitt 3.6.1).

Ist der Spracherkenner für eine *mittlere Benutzungsdauer* (100 - 1000 Wörter) konzipiert, bieten sich die Verfahren aus Kapitel 4 an, d.h. die direkte Adaption der Hidden Markov Modelle. Das Verfahren SCHMIX, also die generalisierte Adaption von Mixtur-Verteilungen schnitt dabei am besten ab (vgl. Abschnitt 4.4.9). Handelt es sich um einen Spracherkenner mit 'bottom-up'-Ansatz, kann zusätzlich eines der Verfahren aus Kapitel 5 zur Adaption der Bewertung von Lautsymbolen eingesetzt werden.

Für eine *lange Benutzungsdauer* (Dauerbenutzung) wäre eine sequentielle Anwendung aller drei genannten Verfahren ein guter Ansatz zur Adaption an einen neuen Benutzer. D.h. ähnlich wie in Abschnitt 7.2.2 werden nacheinander Codebuch, HMM und Bewertung von Lautsymbolen adaptiert.

Wie bereits in Kapitel 9 angedeutet, eröffnet sich vor allem in den Bereichen der symbolischen Sprachverarbeitung wie Sprachmodell, Syntax, Semantik, Dialogführung, etc. ein weites Feld von bisher nicht untersuchten Möglichkeiten für die Sprecheradaption. Interessant ist dabei, daß in zunehmendem Maße regelbasierte Verfahren an Bedeutung gewinnen, je weiter man sich dabei von der Signalverarbeitung im engeren Sinne entfernt. Als Beispiel kann auf die in Abschnitt 9.2 vorgeschlagene Adaption von Aussprachemodellen verwiesen werden. Das dort vorgeschlagene Konzept basiert im wesentlichen auf den allgemeinen Verschleifungsregeln der deutschen Sprache. Eine deutliche Verbesserung dieses und ähnlicher regelbasierter Ansätze wäre ein Verfahren, welches die automatische und gleichzeitig sichere Formulierung von Verschleifungsregeln durch Beobachtung von Sprachmaterial leistet.

Die Entwicklung sprecheradaptiver Verfahren in Bereichen wie Semantik, Dialogführung, Ergonomie setzt nach Meinung des Autors eine weite interdisziplinäre Zusammenarbeit mehrere Forschungsgebiete voraus. Dabei wären vor allem die Bereiche Linguistik, Psychologie und natürlich die Ingenieurwissenschaften zu nennen.

Zum Abschluß möchte ich mich bei allen ganz herzlichst bedanken, die in irgendeiner Weise zum Zustandekommen der vorliegenden Arbeit beigetragen haben. An erster Stelle ist hierbei mein langjähriger Mentor und Betreuer Priv. Doz. Dr.-Ing. G. Ruske zu nennen. Ich hatte das große Glück, in seiner Gruppe die richtige, fruchtbare Mischung aus Freiheit und konstruktiver Kritik zu finden. Des weiteren bin ich H. Prof. Dr.techn. J. Swoboda und allen Mitarbeitern seines Lehrstuhls für Datenverarbeitung an der Technischen Universität München zu großem Dank verpflichtet. Dabei muß ich die Mitglieder der Forschungsgruppe Sprachverarbeitung besonders erwähnen: viele Ideen und Methoden dieser Arbeit entsprangen den vielfältigen und teils leidenschaftlich geführten Diskussionen mit ihnen. Für die zahlreichen Hilfestellungen, die Bereitstellung von Rechenleistung und Speicher danke ich dem Systemverwalter Dr.-Ing. K. Centmayer. Für die hier beschriebenen Untersuchungen wurden ca. 6238 CPU-Stunden auf verschiedenen Workstations verbraucht. An letzter und damit bevorzugter Stelle möchte ich meiner Frau für ihre unendliche Geduld danken, ohne die zweifellos ein Ingenieurwissenschaftler kaum zu ertragen ist.

Literaturverzeichnis

[Bak89] Janet M. Baker: *DRAGONDICTATEtm-30K:* Natural Language Speech Recognition with 30000 Words, Proc. of the EUROSPEECH 1989 in Paris, p. 161, 1989.

[Bam91] P.G. Bamberg, M.A. Mandel: Adaptable phoneme-based models, Speech communication, Vol. 10, No. 5 - 6, Dez 1991, pp. 437 - 451.

[Beh91] M. Beham: Untersuchungen von Merkmalen für die automatische Erkennung deutscher Sprachlaute, Abschlußbericht zum gemeinsamen Forschungsvorhaben der Technischen Universität München und der Deutschen Bundespost, Lehrstuhl für Datenverarbeitung, Technische Universität München, S. 14 - 15, Okt 1991.

[Bon91] H. Bonneau-Maynard: Vector Quantization for Speaker Adaptation: Results on a 5000-Word Database, IEEE Speech Communication, Vol. 10, No. 5 - 6, pp. 463 - 469, Dez 1991.

[Cla90] F. Class: Standardisierung von Sprachmustern durch vokabularinvariante Abbildungen zur Anpassung an Spracherkennungssysteme, Fortschrittsberichte VDI, Reihe 10: Informatik/Kommunikationstechnik, Nr. 131, VDI Verlag, Düsseldorf, 1990.

[Del92] S. Della Pietra, V. Della Pietra, R.L. Mercer, S. Roukos: Adaptive Language Modelling Using Minimum Discriminant Estimation, Proc. of the International Conference on Acoustics, Speech and Signal Processing, San Franzisco, California, p. I-633, Apr 1992.

[Dud90] Duden – Band 6, Aussprachewörterbuch, hrsg. vom Wiss. Rat d. Dudenred.: G. Drosdowski u.a., Dudenverlag, Mannheim Wien Zürich, 1990.

[Ehr89] U. Ehrlich: Bedeutungsanalyse und Interpretation von Zeitangaben im Spracherkennungs- und Dialogsystem EVAR, Informationstechnik it31 (1989)6, Oldenbourg Verlag, S. 373, 1989.

[Fun91] P. Fung, T. Kawahara, S. Doshita: Unsupervised Speaker Normalization by Speaker Markov Model Converter for Speaker-Independent Speech Recognition, Proc. of the EUROSPEECH in Genua, Italy, pp. 1111 - 1114, Oct 1991.

[Fur89] S. Furui: Unsupervised Speaker Adaptation Based on Hierarchical Spectral Clustering, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Glasgow, U.K., pp. 286 - 289, May 1989.

[Gev91] S. Geva, J. Sitte: Adaptive Nearest Neighbor Pattern Classification, IEEE Transactions on Neuronal Networks, Vol. 2, No. 2, pp. 318 - 322, Mar 1991.

[Hag89] J. Hagenauer: Einführung in die Codierungstheorie, Vorlesungsskript SS 1989, Technische Universität München, DLR Institut für Nachrichtentechnik, Oberpfaffenhofen, 1989.

[Hat90] H. Hattori: Speaker Adaptation Based on Markov Modelling of Speakers in Speaker Independent Speech Recognition, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 845 - 848, May 1991.

[Hof91] U. Hofmann: Automatische Modellierung von Aussprachevarianten in der Spracherkennung, Diplomarbeit am Lehrstuhl für Datenverarbeitung, Technische Universität München, Dez 1991.

[Hua88] X.D. Huang, M.A. Jack: Hidden Markov Modelling of Speech Based on Semicontinuous Model, Electronics Letters, Vol. 24, No. 1, pp. 6 - 7, 7th Jan 1988.

[Hua91] X.D. Huang, K.F. Lee: On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 877 - 880, May 1991.

[Jar87] A. Jarre, R. Pieraccini: Some Experiments on HMM Speaker Adaptation, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Dallas, Texas, pp. 1273 - 1276, Apr 1987.

[Jas82] J. Jaschul: Adaption vorverarbeiteter Sprachsignale zum Erreichen der Sprecherunabhängigkeit automatischer Spracherkennungssysteme, Dissertation am Lehrstuhl für Datenverarbeitung Technische Universität München, 1982.

[Jek89] U. Jekosch, T. Becker: Maschinelle Generierung von Aussprachevarianten: Perspektiven für Sprachsynthese- und Spracherkennungssysteme, Informationstechnik it (1989)6, S. 400, Oldenbourg Verlag, 1989.

[Jel91] F. Jelinek, B. Merialdo, S. Roukos, M. Strauss: A Dynamic Language Model for Speech Recognition, Proc. of the Fourth DARPA Workshop Speech Natural Language, Pacific Grove, California, pp. 293 - 295, Feb 1991.

[Jua85] B.H. Juang: Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains, AT&T Technical Journal, Vol. 64, No. 6, pp. 1235 - 1249, Jul - Aug 1985.

[Jua90] B.H. Juang, L.R. Rabiner: The Segmential K-Means Algorithm for Estimating Parameters of Hidden Markov Models, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, No. 9, pp. 1639 - 1641, Sep 1990.

[Käm90] B.R. Kämmerer: Sprecherunabhängigkeit und Sprecheradaption, Informatik Fachberichte 244, Springer Berlin - Heidelberg - New York, 1990.

[Kae98] F.W. Kaeding: Häufigkeitswörterbuch der deutschen Sprache, Selbstverlag des Herausgebers, Steglitz bei Berlin, 1898.

[Ken90] P. Kenny, M. Lenning, P. Mermelstein: Speaker Adaptation in a Large-Vocabulary Gaussian HMM Recognizer, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 9, pp. 917 - 920, Sept 1990.

[Koh88] T. Kohonen: Self-organization and Associative Memory (2nd Ed.), Springer, Berlin - Heidelberg - New York - Tokyo, pp. 199 - 202, 1988.

[Kra91] J. Krause, L. Hitzenberger: Endbericht zum Projekt „Sprachverstehende Systeme, Teilprojekt: Simulation einer multimedialen Dialog-Benutzer-Schnittstelle“, Universität Regensburg, Ling. Informationswissenschaft, März 1991.

[Kub90] F. Kubala, R. Schwartz, C. Barry: Speaker Adaptation from a Speaker-Independent Traing Corpus, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, pp. 137 - 140, Apr 1990.

[Kuh90] R. Kuhn, R. de Mori: A Cache Based Natural Language Model for Speech Recognition, IEEE Trans. on Pattern Analysis and Mashine Intelligence, Vol. 12, No. 6, pp. 570 - 583, Jun 1990.

[Lee89] Kai-Fu Lee: Automatic Speech Recognition, The Development of the SPHINX System, Chap. 7, Learning and Adaptation, pp. 115 - 127, Kluwer Academic Publishers, Boston Dordrecht London, 1989.

[Lev89] S.E. Levinson, M.Y. Liberman, A. Ljolje, L.G. Miller: Speaker Independent Transcription of Fluent Speech for Large Vocabulary, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Glasgow, U.K., pp. 441 - 444, May 1989.

[Lin80] Y. Linde, A. Buzo, R.M. Gray: An Algorithm for Vector Quantizer Design, IEEE Trans. on Communication, Vol. COM-28, No. 1, pp. 84 - 95, Jan 1980.

[Mat90] L. Mathan, L. Miclet: Speaker Hierarchical Clustering for Improving Speaker Independent HMM Word Recognition, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, pp. 149 - 152, Apr 1989.

[Mat92] S. Matsunaga, T. Yamada, K. Shikano: Task Adaptation in Stochastic Models for Continuous Speech Recognition, Proc. of the International Conference on Acoustics, Speech and Signal Processing, San Franzisco, California, p. I-165, Mar 1992.

[Ney91] H. Ney, U. Essen: On Smoothing Techniques for Bigram-Based Natural Language Modelling, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 825 - 828, May 1991.

[Pae89/1] A. Paeseler, V. Steinbiss, A. Noll: Phonem-Based Continuous-Speech Recognition in the SPICOS-II System, Informationstechnik it31 (1989)6, Oldenbourg Verlag, München Wien, S. 392, 1989.

[Pae89/2] A. Paeseler, H. Ney: Continuous Speech Recognition Using a Stochastic Language Model, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Glasgow, U.K., pp. 719 - 722, May 1989.

[Pla91] B. Plannerer: Verallgemeinerte Beschreibung von „Hidden–Markov"–Modellen für die Spracherkennung, Diplomarbeit am Lehrstuhl für Datenverarbeitung, Technische Universität München, 1991.

[Pla92] B. Plannerer: Recognition of Demisyllable Based Units Using Semicontinuous Hidden Markov Modells, Proc. of the International Conference on Acoustics, Speech and Signal Processing, San Franzisco, California, pp. 581 - 584, Mar 1992.

[Rab86] L.R. Rabiner, B.H. Juang: An Introduktion to Hidden Markow Modells, IEEE ASSP Magazine, p. 4, Jan 1986.

[Rig89] G. Rigoll: Speaker Adaptation for Large Vokabulary Speech Recognition Systems Using Speaker Markow Modells, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Glasgow, U.K., pp. 5 - 8, May 1989.

[Rig90] G. Rigoll: Baseform Adaptation for Large Vocabulary HMM Based Speech Recognition Systems, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, pp. 144 - 147, Apr 1990.

[Rig91] G. Rigoll: Sprache erkennen, MC, Feb 1991, S. 70 - 75, 1991.

[Roz91] W.A. Rozzi, R.M. Stern: Speaker Adaptation in Continuous Speech Recognition via Estimation of Correlated Mean Vectors, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 865 - 868, May 1991.

[Rus88] G. Ruske: Automatische Spracherkennung: Methoden der Klassifikation und Merkmalsextraktion, Oldenbourg Verlag, München Wien, 1988.

[Rus91] G. Ruske, M. Beham: Gehörbezogene automatische Spracherkennung, in „Sprachliche Mensch-Maschine-Kommunikation", Hrsg. Mangold, Oldenbourg Verlag, München Wien, S. 33 - 48, 1991.

[Sam90] Technical Report SAM, ESPRIT Projekt 2589, ohne Ort, 1990.

[Sch91/1] F. Schiel, F. Wolfertstetter: Regelbasierte Erzeugung von robusten Aussprachemodellen und deren Darstellung im Silbenraster, Studientexte zur Sprachkommunikation, Heft 8 (ISSN 0940-6832), S. 173 - 182, 1991.

[Sch91/2] F. Schiel: Modifizierter A*-Algorithmus zur Erkennung fließend gesprochener Sätze, Informatik Fachberichte 290, Mustererkennung 1991, Springer-Verlag, S. 244 - 250.

[Sch92/1] F. Schiel: Rapid Non-Supervised Speaker Adaptation of Semincontinuous Hidden Markov Modells, Proc. of the International Conference on Speech and Language Processing, Banff, Alberta, pp. 1463 - 1466, 1992.

[Sch92/2] F. Schiel: A new Approach to Speaker Adaptation by Modelling the Pronunciation in Automatic Speech Recognition, Proc. of the Fourth Australian International Conference in Speech Science and Technology, Brisbane, Australia, pp. 425 - 430, Dez 1992.

[Sch92/3] F. Schiel: Phonetically Seeded SCHMM of Variable Length for Speaker Independent Recognition of Isolated Words, Proc. of the Fourth Australian International Conference in Speech Science and Technology, Brisbane, Australia, pp. 92 - 97, Dez 1992.

[Scm91] O. Schmidbauer, H. Höge: Speaker Adaptation Based on Articulatory Features, Proc. of the EUROSPEECH 1991 in Genua, Italy, pp. 1099 - 1102, Oct 1992.

[Shi86] Kiyoshiro Shikano, Kai-Fu Lee, Raj Reddy: Speaker Adaptation through Vector Quantization, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Tokyo, Japan, pp. 2643 - 2646, Apr 1986.

[Shi91] K. Shinoda, K. Iso, T. Watanabe: Speaker Adaptation for Demi-Syllable Based Continuous Density HMM, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 857 - 860, May 1991.

[Sot84] J. Sotschek: Sätze für Sprachgütemessungen und ihre phonologische Anpassung an die Deutsche Sprache, Tagungsband DAGA: Fortschritte der Akustik, S. 873 - 876, 1984.

[Str91] M. Streicher: Implementation eines Ganzwort-Erkenners mit semikontinuierlichen Hidden Markov Modellen, Diplomarbeit am Lehrstuhl für Datenverarbeitung, Technische Universität München, 1991.

[Wei90] W. Weigel: Silbenorientierte Erkennung fließender Sprache mittels diskreter stochastischer Modellierung, Dissertation, Lehrstuhl für Datenverarbeitung, Technische Universität München, 1990.

[Wik90] P. Winkler: Lernverfahren für Wortmodelle zur automatischen Erkennung gesprochener Sätze, Diplomarbeit am Lehrstuhl für Datenverarbeitung, Technische Universität München, 1990.

[Win91] M. Winter: Untersuchungen zur Sprecheradaption auf symbolischer Ebene in der automatischen Spracherkennung, Diplomarbeit am Lehrstuhl für Datenverarbeitung, Technische Universität München, 1991.

[Wof91] F. Wolfertstetter: Regelbasierte Generierung und Modellierung von Aussprachevarianten in einem silbenteilorientierten Spracherkennungssystem, Diplomarbeit am Lehrstuhl für Datenverarbeitung, Technische Universität München, 1991.

[Zue86] V.W. Zue, et al.: The Developement of the MIT Lisp-Machine Based Speech Research Workstation, Proc. of the International Conference on Acoustics, Speech and Signal Processing, Tokyo, Japan, pp. 329 - 332, Apr 1986.

[Zwi82] E. Zwicker: Psychoakustik, Springer Verlag, Berlin Heidelberg New York, 1982.

Anhang A

Phonem-Inventar

Lautinventar der deutschen Sprache nach [Sam90].

SAM	*Aussprache*	*SAM*	*Aussprache*	*SAM*	*Aussprache*	*SAM*	*Aussprache*
6	besṣer	Y	Tüte	tS	deuṭsch	N	Diṇg
a	Saṭz	y:	süss	dZ	Ḍschungel	l	Ḷeim
a:	Taṭ	@	bittẹ	f	ḟast	R	Ṛhein
e:	Bẹet	aI	Ẹis	v	ṿas		
I	Siṭz	aU	Haụs	s	Taṣse		
i:	Liẹd	OY	Kreụz	z	Haṣe		
O	Troṭz	p	Ṗein	S	waṣchen		
o:	roṭ	b	Ḅein	Z	Ǧenie		
U	Schụtz	t	Ṭeich	C	siçher		
u:	Blụt	d	Ḍeich	j	Ja̰hr		
E	Geseṭz	k	Ḳunst	x	Buḵh		
E:	spät	g	Ǧunst	h	Ḥand		
9	plötzlich	pf	Ṗfahl	m	ṃein		
2:	blöd	ts	Ẓahl	n	ṇein		

Bemerkung:
Die Laute /2:/, /Z/ und /dZ/ werden in diesem System nicht verwendet, da sie im Sprachmaterial nicht vorhanden sind. Außerdem wird das Material für die Laute /a/ und /6/ zusammengefaßt, da anhand der Segmentierung kein Unterschied zwischen diesen Lauten festzustellen ist. Diese Gruppe erhält daher das Label /6/. Gleiches gilt für die Laute /E:/ und /E/, da für /E:/ nicht genügend Samples vorhanden sind. Das Material dieser beiden Laute wurde unter /E/ zusammengefaßt.

Anhang B

Sprachmaterial

B.1 *Berliner Sätze*

Die Sprachdatenbasis *Berliner Sätze* nach [Sot84] enthält die auf der folgenden Seite wiedergegebenen Wörter in fließender Rede, gesprochen von 12 männlichen und weiblichen Sprechern.

aber	abfahrt	abteil	acht	achte	acker
aerzte	alle	am	an	angesagt	ansage
anzuenden	arbeitet	auch	auf	aufbruch	aus
autos	bach	bahn	bahnhof	bahnsteig	bald
bauer	begeistert	beim	besuchen	bewohnt	bitte
blauen	bluehen	bremsen	brett	broetchenkorb	bunte
butter	da	dach	daemmerung	dahinter	damit
dampft	danach	daneben	darf	das	dazu
dein	dem	den	denn	der	dickicht
die	doch	dort	draussen	drei	drin
drueben	du	durch	eilen	eiligen	ein
eine	einen	einer	einkaufen	einverstanden	endet
entfernt	er	erbsen	erna	es	essen
extra	fahrkarten	fahrt	fehlte	feld	felder
feldscheune	fenster	festen	figur	fliegt	frau
freie	frische	fruehlingswetter	fruehstueck	fuehrt	fuer
furchen	fuss	gabel	gar	gardinen	gedeckt
gehen	gehoeren	gehoert	gehts	gelbe	gerne
geschaeft	geschlafen	gestern	gestillt	gewinnt	gib
gibt	gleich	glueck	graesslich	gruent	gut
haben	haengen	haeuschen	hans	hasen	hat
hause	heim	heute	hier	himmel	hoeren
hunger	ich	ihn	ihr	im	in
ins	isst	ist	ja	jetzt	junge
kaffee	karten	kartoffeln	kasse	kaufen	keinen
kleinen	klingt	kocht	koennen	kommen	konnte
konserven	kuchen	kuechenofen	kurz	lacht	laenger
laeuft	landschaft	leicht	leise	lesen	leuchten
leute	liegen	liegt	links	loest	luft
machen	macht	maechtig	maenner	maiers	mal
manche	mehr	messer	milch	minuten	mir
mischt	mit	mittagessen	mitte	moechte	montag
morgens	muessen	muesste	musik	muss	mutter
nach	nacheinander	nacht	nahrungsmittel	neben	nem
nettes	nicht	noch	nun	ob	obstbaeume
oel	pfeife	pfeift	pflug	plaetschernden	praechtig
quietschen	radfahrer	radio	rauchen	rechnen	regentonne
regnerisch	richten	riecht	rinder	rollen	rosengarten
rueckt	salat	sausen	schalter	schlafen	schluss
schnell	schnitzel	schoener	schoenes	schon	schuhe
schularbeiten	sechs	seine	seinem	sich	sie
sieben	sieglinde	sind	sitzen	skat	so
sofa	solltest	sonne	sonst	sorgt	spazieren
spiele	spuere	staerken	station	stehen	steht
steigen	strasse	stuehle	stuermte	suche	suessigkeiten
tag	tante	tassen	teller	tiefe	tisch
topf	trauriges	trinkt	tulpen	tut	ueber
ueberfahren	ueberquere	uhr	und	uns	unser
vater	verkuendet	verschwinden	verstauchter	viele	vielleicht
voller	vorbei	vorsichtig	waerme	waers	wagen
wald	wand	wanderung	war	was	wasser
weg	weht	weide	weissbrot	weniger	wer
wie	will	wind	wintersaat	wir	wird
wirst	wohl	wolken	wollen	wurst	zaun
zeichnet	zeit	zeitung	ziehen	zieht	ziel
zimmerleute	zu	zucker	zug	zugbegleiter	zum
zur	zurueck	zuvor			

B.2 Sprachstichproben

Für die Simulationen der Sprecheradaption wurden folgende drei Sprachstichproben definiert. Eine vergleichende Lautstatistik ist in B.2.4 enthalten.

B.2.1 Sprachstichprobe Adaption

acht	achte	aerzte	ansage	aufbruch	bahnhof	begeistert
bewohnt	dach	damit	danach	darf	den	denn
dickicht	doch	drin	eilen	einen	einer	er
erbsen	extra	fahrt	fehlte	fenster	festen	figur
freie	fruehlingswetter	gehts	gerne	geschlafen	gestillt	hause
heute	hier	hoeren	im	in	kaffee	karten
kasse	keinen	konserven	lacht	leuchten	leute	links
loest	luft	machen	mehr	mitte	moechte	muesste
mutter	nacht	nahrungsmittel	nettes	nun	obstbaeume	plaetschernden
quietschen	richten	rinder	sausen	schalter	schluss	schoenes
schon	schuhe	schularbeiten	sechs	sich	skat	sofa
sorgt	spuere	staerken	steigen	suessigkeiten	tag	tulpen
tut	ueberfahren	ueberquere	unser	voller	weissbrot	wie
will	wird	zeit	zeitung	ziehen	ziel	zu
zum	zuvor					

B.2.2 Sprachstichprobe Test1

acker	am	angesagt	anzuenden	arbeitet	auf	bach
bahn	bauer	besuchen	bitte	blauen	bunte	butter
da	dampft	daneben	das	der	die	draussen
drueben	ein	einkaufen	einverstanden	entfernt	feld	feldscheune
fliegt	fuer	furchen	gabel	gedeckt	gehoert	gewinnt
gleich	glueck	gruent	haben	haengen	haeuschen	hans
heim	himmel	ins	ja	kartoffeln	kleinen	klingt
kocht	konnte	kurz	landschaft	leise	macht	maiers
manche	messer	milch	mir	mittagessen	morgens	nach
ob	pfeift	pflug	radio	rauchen	rosengarten	schoener
seine	sie	sieben	sieglinde	sitzen	so	solltest
spiele	stehen	strasse	stuermte	teller	tiefe	tisch
trauriges	ueber	vater	vielleicht	vorsichtig	waers	wald
wanderung	was	weg	weide	wintersaat	wohl	wollen
zimmerleute	zur					

B.2.3 Sprachstichprobe Test2

aber	abfahrt	abteil	acht	achte	acker
aerzte	am	an	angesagt	ansage	anzuenden
arbeitet	auch	auf	aufbruch	bach	bahn
bahnhof	bald	bauer	begeistert	beim	besuchen
bewohnt	bitte	blauen	bunte	butter	da
dach	dahinter	damit	dampft	danach	daneben
darf	das	dem	den	denn	der
dickicht	die	doch	dort	draussen	drin
drueben	eilen	ein	eine	einen	einer
einkaufen	einverstanden	entfernt	er	erbsen	extra
fahrt	fehlte	feld	felder	feldscheune	fenster
festen	figur	fliegt	frau	freie	frische
fruehlingswetter	fuer	furchen	gabel	gar	gedeckt
gehoert	gehts	gerne	geschlafen	gestillt	gewinnt
gleich	glueck	gruent	haben	haengen	haeuschen
hans	hasen	hause	heim	heute	hier
himmel	hoeren	ihn	im	in	ins
ja	kaffee	karten	kartoffeln	kasse	keinen
kleinen	klingt	kocht	konnte	konserven	kurz
lacht	landschaft	leise	leuchten	leute	liegt
links	loest	luft	machen	macht	maechtig
maiers	manche	mehr	messer	milch	minuten
mir	mit	mittagessen	mitte	moechte	morgens
muesste	muss	mutter	nach	nacht	nahrungsmittel
neben	nettes	noch	nun	ob	obstbaeume
pfeift	pflug	plaetschernden	praechtig	quietschen	radfahrer
radio	rauchen	richten	rinder	rollen	rosengarten
sausen	schalter	schlafen	schluss	schnitzel	schoener
schoenes	schon	schuhe	schularbeiten	sechs	seine
seinem	sich	sie	sieben	sieglinde	sitzen
skat	so	sofa	solltest	sonne	sorgt
spazieren	spiele	spuere	staerken	stehen	steigen
strasse	stuermte	suessigkeiten	tag	teller	tiefe
tisch	trauriges	tulpen	tut	ueber	ueberfahren
ueberquere	uhr	unser	vater	vielleicht	voller
vorsichtig	waers	wald	wanderung	war	was
wasser	weg	weide	weissbrot	weniger	wie
will	wintersaat	wird	wirst	wohl	wollen
zeit	zeitung	ziehen	ziel	zimmerleute	zu
zugbegleiter	zum	zur	zurueck	zuvor	

B.2.4 Vergleichende Lautstatistik

Laut	Adaption		Test1		Test2		Berliner Sätze	
6	9	2 %	9	2 %	23	2 %	28	2 %
a:	17	3 %	23	5 %	50	4 %	64	4 %
e:	7	1 %	4	1 %	14	1 %	21	1 %
I	18	4 %	16	3 %	43	4 %	56	3 %
i:	5	1 %	9	2 %	17	1 %	27	2 %
O	4	1 %	8	2 %	16	1 %	24	1 %
o:	7	1 %	4	1 %	11	1 %	14	1 %
U	10	2 %	6	1 %	18	2 %	27	2 %
u:	6	1 %	2	0 %	11	1 %	18	1 %
E	13	3 %	12	2 %	28	2 %	49	3 %
@	58	12 %	57	10 %	134	11 %	182	11 %
9	4	1 %	2	0 %	6	1 %	10	1 %
Y	1	0 %	4	1 %	6	1 %	11	1 %
y:	5	1 %	3	1 %	8	1 %	12	1 %
aI	12	2 %	13	3 %	30	3 %	39	2 %
aU	3	1 %	7	1 %	12	1 %	17	1 %
OY	4	1 %	3	1 %	7	1 %	8	0 %
l	20	4 %	24	5 %	52	4 %	69	4 %
m	11	2 %	14	3 %	32	3 %	43	3 %
n	46	9 %	50	10 %	113	10 %	167	10 %
N	4	1 %	3	1 %	7	1 %	12	1 %
R	41	8 %	37	7 %	99	8 %	137	8 %
z	10	2 %	13	3 %	26	2 %	32	2 %
s	20	4 %	13	3 %	36	3 %	51	3 %
x	9	2 %	6	1 %	17	1 %	21	1 %
C	6	1 %	9	2 %	19	2 %	29	2 %
f	17	3 %	16	3 %	39	3 %	53	3 %
S	11	2 %	8	2 %	23	2 %	34	2 %
v	9	2 %	10	2 %	23	2 %	33	2 %
j	0	0 %	1	0 %	1	0 %	3	0 %
h	5	1 %	8	2 %	15	1 %	18	1 %
b	9	2 %	15	3 %	29	2 %	36	2 %
d	11	2 %	15	3 %	30	3 %	41	2 %
g	8	2 %	10	2 %	21	2 %	36	2 %
p	5	1 %	2	0 %	11	1 %	14	1 %
t	47	9 %	43	9 %	103	9 %	150	9 %
k	16	3 %	16	3 %	35	3 %	52	3 %
pf	0	0 %	3	1 %	3	0 %	5	0 %
ts	9	2 %	5	1 %	18	2 %	26	2 %
tS	2	0 %	1	0 %	3	0 %	3	0 %

Anhang C

Viterbi-Algorithmus

C.1 Viterbi-Test

C.1.1 Allgemeiner Viterbi-Algorithmus

Der Viterbi-Algorithmus geht von der vereinfachenden Annahme aus, da die Erzeugungswahrscheinlichkeit bereits durch die alleinige Auswertung des besten Pfades anstelle der Summe über alle Pfade, ausreichend angenähert wird, d.h. die Anteile der anderen Pfade sind – auch in ihrer Summe (!) – vernachlässigbar klein. Es wird also sowohl der beste Pfad durch das Modell, als auch die Erzeugungswahrscheinlichkeit der Symbolfolge, wenn dieser Pfad durchlaufen wird, ermittelt. Definitionen, die nicht in der allgemeinen Nomenklatur (s. Anhang D) enthalten sind:

e_z	Wahrscheinlichkeit dafür, da sich das Modell zum Zeitpunkt $t = 1$ im Zustand z befindet (Einsprungvektor)
O_t	Zum Zeitpunkt t auftretendes Symbol
$b_z(O_t)$	Emissionswkt. des Symbols O_t im Zustand z
$c_y(t)$	Maximale Wahrscheinlichkeit für eine Zustandsfolge, die die ersten t Symbole der der Symbolfolge generiert und im Zustand y endet.
$\psi_z(t)$	Backtracking-Information (*WENN* der beste Pfad im Zeitpunkt t durch Zustand z geht, *DANN* geht er im Zeitpunkt $t-1$ durch Zustand $\psi_z(t)$.)
$y^*(t)$	Zustand zum Zeitpunkt t im besten Pfad

Formal lät sich der Viterbi-Algorithmus in vier Schritte gliedern (vgl. auch [Rab86]):

1. Initialisierung:

$$c_z(1) = e_z b_z(O_1) \qquad 1 \leq z \leq Z \tag{C.1}$$

$$\psi_z(1) = 0 \tag{C.2}$$

2. Rekursion:
   Durch eine schrittweise Abarbeitung der Trellis wird die Erzeugungswahrscheinlichkeit für den besten Pfad ermittelt.

für $2 \leq t \leq T$, für $1 \leq z \leq Z$

$$c_z(t) = \max_{1 \leq y \leq Z} [c_y(t-1) \cdot a_{yz}] \cdot b_z(O_t) \tag{C.3}$$

$$\psi_z(t) = \operatorname*{argmax}_{1 \leq y \leq Z} [c_y(t-1) \cdot a_{yz}] \tag{C.4}$$

3. Terminierung:

$$p_{viterbi} = \max_{1 \leq y \leq Z} [c_y(T)] \tag{C.5}$$

$$y^*(T) = \operatorname*{argmax}_{1 \leq y \leq Z} [c_y(T)] \tag{C.6}$$

4. Backtracking:
für $t = T-1, \quad T-2, \quad T-3, \ \ldots\ldots, 1$

$$y^*(t) = \psi_{y^*(t+1)}(t+1) \tag{C.7}$$

Dies ist die allgemeine Form des Viterbi-Algorithmus, wie er für beliebige Modellstrukturen verwendet werden kann. Aufgrund der genannten Einschränkungen der Trellis bei Phonem- bzw. Wortmodellen, ergeben sich jedoch noch einige Modifikationen, die den Berechnungsaufwand zum Teil deutlich reduzieren.

C.1.2 Vereinfachungen

Da der Einsprung nur in Zustand 1 erfolgen kann gilt $e_1 = 1$ und $e_z = 0$ für $2 \leq z \leq Z$. Damit wird aus Gl. C.1:

$$c_z(1) = \begin{cases} b_1(O_1) & für: \quad z = 1 \\ 0 & für: \quad 2 \leq z \leq Z \end{cases} \tag{C.8}$$

In der Rekursion indiziert der Parameter y die Vorgänger des Zustandes z. In Links-Rechts-Modellen kommen aber nur die Zustände $y \leq z$ als Vorgänger des Zustandes z in Frage. Damit wird aus den Gleichungen C.3 und C.4:

$$c_z(t) = \max_{1 \leq y \leq z} [c_y(t-1) \cdot a_{yz}] \cdot b_z(O_t) \tag{C.9}$$

$$\psi_n(t) = \operatorname*{argmax}_{1 \leq y \leq z} [c_y(t-1) \cdot a_{yz}] \tag{C.10}$$

Da sich das Modell im Zeitschritt T im Zustand Z befinden mu, kann die Maximumsbildung in der Terminierung entfallen, so da sich hierfür anstelle der Gl. C.5 und Gl. C.6 folgende Ausdrücke ergeben:

$$p_{viterbi} = c_Z(T) \tag{C.11}$$

$$y^*(T) = Z \tag{C.12}$$

C.1.3 Semikontinuierlicher Ansatz und Produkt-Code

Aufgrund der Verwendung von drei Merkmalen und semikontinuierlicher Quantisierung, ist es nicht möglich, die Emissionswahrscheinlichkeit einfach dem Modell zu entnehmen. Die 'Emissionswahrscheinlichkeit der Objekte zum Zeitpunkt t' muß durch eine geeignete Verknüpfung der drei Merkmale berechnet werden.

Zur Verknüpfung der Merkmale *sh*, *di* und *la* (vgl. Abschnitt 2.2.2) wird ein sogenannter *Produkt-Code* berechnet, wobei vereinfachend angenommen wird, da die drei Merkmale statistisch unabhängig sind.

Die entsprechende Gleichung lautet:

$$e(z,t) = e_{sh}(z,t) \cdot e_{di}(z,t) \cdot e_{la}(z,t) \tag{C.13}$$

Die Einzel-Emissionswahrscheinlichkeiten $e_k(z,t)$ der drei Merkmale im Zustand z berechnen sich nach dem semikontinuierlichen Ansatz[1] wie folgt:

$$e_k(z,t) = \sum_{m=1}^{M} p_k(t,m) q_k(z,m) \tag{C.14}$$

Es wird für alle drei Merkmale eine Summe über alle Symbole des jeweiligen Codebuchs gebildet, in der die Produkte aus Pseudo-Rückschluß auf das Symbol und dem entsprechenden Mixtur-Koeffizienten im Zustand z aufsummiert werden. Zur Berechnung der Erzeugungswahrscheinlichkeit einer Merkmalsvektorfolge durch ein Modell mu nun anstelle der Emissionswahrscheinlichkeit $b_z(O_t)$ der Produkt-Code $e(z,t)$ in den Viterbi-Algorithmus (Gl. C.9) eingesetzt werden.

C.2 Viterbi-Training (*segmental-k-means*)

Für das Training existieren Verfahren, die auf dem Forward-Backward-Algorithmus (*Baum-Welch-Training*) bzw. auf dem Viterbi-Algorithmus (*segmental-k-means*, [Jua90]) basieren. Dabei wird ein Modell anhand einer Lernstichprobe so lange iterativ verbessert, bis sich keine (nennenswerte) Veränderung mehr ergibt.

Der Trainingsprozess für ein HMM läuft wie folgt ab:

1. Für alle Referenzdaten zu einem HMM wird der Viterbi-Algorithmus durchgeführt und dabei alle Pfade gespeichert, sowie das mittlere Bewertungsma ermittelt. Unterschreitet die Änderung des mittleren Bewertungsmaes eine gewählte Grenze, so wird das Training beendet.

2. Die Übergangswahrscheinlichkeiten werden mit Hilfe der gespeicherten Beobachtungen neu berechnet. Dabei ist zu beachten, da am Ende des Modells ein Aussprung angenommen wird, so da die Übergangswahrscheinlichkeit $a_{ZZ}(h) < 1$ ist.

$$a_{ij} \quad = \quad \frac{trans_{ij}}{trans_{ix}} \tag{C.15}$$

[1] Zu beachten ist jedoch, da es sich beim Ausdruck $p_{\underline{k}}(t)$ um die Pseudo-Rückschluwkt. der Codebuch-Symbole $m = 1 \ldots M$ auf den Vektor $\underline{k}(t)$ handelt, und nicht wie in [Hua88] um die Vorwärtswkt.

mit: $trans_{ij}$ Anzahl der Übergänge von Zustand i nach Zustand j
$trans_{ix}$ Anzahl der Übergänge aus Zustand i heraus

3. Neuberechnung der Mixturkoeffizienten.
Für jeden Wahrscheinlichkeitsvektor $\underline{p}_k(t)$ kann anhand des Viterbi-Pfades entschieden werden, in welchem Zustand sich das Modell zum entsprechenden Zeitpunkt befunden hat. Auf die Mixturkoeffizienten in diesem Zustand wird der Wahrscheinlichkeitsvektor aufsummiert. Anschlieend wird wieder auf $\sum \equiv 1$ normiert. Da die Pseudorückschlußwahrscheinlichkeiten auf 1 normiert sind, genügt die Division durch die Anzahl der in den Zustand z gefallenen Beobachtungen $J(z)$. Die Neuabschätzung der Mixturen $\underline{q}_k(z)$ im Zustand z erfolgt also durch

$$\underline{q}_k(z) = \frac{\sum_{\tau(t)=z} \underline{p}_k(t)}{J(z)} \tag{C.16}$$

mit: J Anzahl der Beobachtungen im Zustand z
$\tau(t)$ Zustandsfolge auf dem Viterbi-Pfad

4. Zurück zu Schritt 1

Anhang D

Nomenklatur

$\overline{X}$	:	arithmetischer Mittelwert von X
$\underline{a}_k(t)$	:	Abstandsvektor zum Merkmal k im Zeitpunkt t (M-dim.), m-te Komponente: $a_k(t,m)$
$a(w,h,f)$	:	Übergangswahrscheinlichkeit vom f-ten Frame zum $f+1$-ten Frame des Wortes w im HMM h
A_k	:	konstanter Anteil der Adaptionsstärke
A_d	:	Faktor der dynamische Adaptionsstärke
B	:	Anzahl der besten Wörter einer Klassifikation, innerhalb derer nach Information für die Adaption gesucht wird.
c	:	Index im FIFO bei kontrollierter Adaption, $c = 1 \ldots C$
C	:	Anzahl der Elemente des FIFOs bei kontrollierter Adaption bzw. Kanalkapazität
d	:	Betrag der Verschiebung im Verfahren ISTSOLL
$d_G(\underline{k}(t), \underline{s}_k(m))$	:	Gewichteter Euklid-Abstand vom Merkmalsvektor $\underline{k}(t)$ zum Prototypen $\underline{s}_k(m)$
di	:	Merkmal Differenzspektrum
$\underline{D}_k(m)$	:	Matrix mit inversen Varianzen des Klusters m im Merkmal k
$\underline{D}'_k(m)$	:	Matrix mit mittlerer inverser Varianz des Klusters m im Merkmal k
Δ_{IS}	:	Faktor, um wieviel der SOLL-Prototyp näher am Merkmalsvektor liegen soll, als der IST-Prototyp im Verfahren ISTSOLL.
$\Delta\underline{k}(t,m)$	:	Differenzvektor für Symbol m aus der Beobachtung des Merkmalsvektors $\underline{k}(t)$ zum Zeitpunkt t
$e(w,h,t)$	:	Emissionswahrscheinlichkeit des t-ten Frames des Wortes w im HMM h
$E(w,h)$	:	Erzeugungswahrscheinlichkeit (EWK) des Wortes w im HMM h
$\overline{\log E}_{FIFO}$	:	mittlere, logarithmierte Erzeugungswahrscheinlichkeit (EWK) der Daten im FIFO bei kontrollierter Adaption

$f(\underline{q}_k(h_i, z_k), \underline{q}_k(h_j, z_l))$	:	Gewichtung der Mixtur $\underline{q}_k(h_i, z_k)$ in Bezug auf die Mixtur $\underline{q}_k(h_j, z_l)$ oder vice versa
F	:	Parameter, welcher die Abh. des Gewichts f vom euklidschen Abstand der beteiligten Mixtur-Verteilungen bestimmt
$g(w)$	:	Adaptionsstärke im Wort w
h	:	Index für HMMs
H	:	Anzahl von HMMs
$H(X)$	:	Entropie
H_T	:	Transinformation
k	:	Merkmal (sh,di,la)
$\underline{k}(t)$	:	Merkmalsvektor zum Merkmal k im Zeitpunkt t (N-dim.), n-te Komponente: $k(t, n)$
L	:	Anzahl der zur Berechnung der Adaptionsstärke $g(w)$ berücksichtigten, vorangegangenen Wörter
l	:	Index über Wörter (s.o.)
la	:	Merkmal Lautheit
λ	:	Anzahl der Wörter im Lexikon
m	:	Index der Prototypen innerhalb eines Codebuchs
M	:	Codebuchgröße
n	:	Index der Komponenten eines Merkmalsvektors
N	:	Merkmalsdimension
η	:	Anzahl der Phoneme
$\underline{p}_k(t)$	:	Normierte Pseudo-Rückschlußwahrscheinlichkeit zum Merkmal k im Zeitpunkt t (M-dim.), m-te Komponente: $p_k(t, m)$
$\underline{q}_k(t)$	:	Mixtur-Koeff. eines HMMs in dem Zustand, der auf dem Viterbi-Pfad zum Zeitpunkt t durchlaufen wird (= $\underline{q}_k(\tau(t))$), m-te Komponente: $q_k(t, m)$
$\underline{q}_k(h, z)$	:	Mixtur-Koeff. eines HMMs h im Zustand z (M-dim.) m-te Komponente: $q_k(h, z, m)$
R	:	Reduktion der Vektorquantisierung in der Test-Phase
$\underline{s}_k(m)$	:	m-ter Prototyp zum Merkmal k (N-dim.), n-te Komponente: $s_k(m, n)$
S	:	Schwellwert für Kontrolle der Adaption bzw. Silbenzahl
sh	:	Merkmal Gehörgerechtes Lautheitsspektrum (shape)
σ_{ij}	:	Kovarianz der Komponente i zur Komponente j
t	:	Frame-Nr., laufender Index der Frames eines Wortes $t = 1 \ldots T$
T, $T(w)$	:	Anzahl der Frames eines Wortes/im Wort w
$\tau(t)$	:	Zustandsfolge auf dem Viterbi-Pfad durch ein Modell
$u(\underline{s}(m), \underline{k}(t), \underline{D}_k(m))$	:	Gewichtsfaktor zur Berücksichtigung der Varianzen des Klusters m bei der Berechnung von $\Delta\underline{k}(t, m)$
V	:	Verfälschungsmaß, prozentuale Abweichung der Transinformation von der Kanalkapazität

w_i	:	Adaptionswort in einer Kette von Wörtern zur Adaption
z	:	Index der Zustände innerhalb eines HMMs
Z	:	Anzahl der Zustände eines HMMs