Lehrstuhl für Mensch-Maschine-Kommunikation

Technische Universität München

Modellierungstechniken und Adaptionsverfahren für die On- und Off-Line Schrifterkennung

Anja Brakensiek

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. G. Färber

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. habil. G. Rigoll
2. Univ.-Prof. Dr.-Ing. H.-M. Groß,
   Technische Universität Ilmenau

Die Dissertation wurde am 19.06.2002 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 12.11.2002 angenommen.

Vorwort

Die vorliegende Arbeit ist während einer gut vierjährigen Tätigkeit als wissenschaftliche Mitarbeiterin am Fachgebiet Technische Informatik des Fachbereiches Elektrotechnik (jetzt Fakultät für Ingenieurwissenschaften) der Gerhard-Mercator-Universität Duisburg entstanden. In dieser Zeit habe ich mich im Rahmen zweier Projekte mit verschiedenen Teilgebieten der automatischen Schrifterkennung beschäftigen können.
Mein besonderer Dank gilt dem Betreuer dieser Arbeit, Prof. Dr.-Ing. habil. Gerhard Rigoll, der mir das Arbeiten an diesen interessanten Forschungsthemen ermöglicht hat. Er hat diese Arbeit während der gesamten Dauer, auch nach seinem Wechsel von der Gerhard-Mercator-Universität Duisburg an die TU München (Lehrstuhl für Mensch-Maschine-Kommunikation), betreut und unterstützt.
Prof. Dr.-Ing. Horst-Michael Groß von der TU Ilmenau (Fachgebiet Neuroinformatik) danke ich für die Übernahme des Korreferates und dem damit verbundenen Aufwand.
Außerdem möchte ich mich bei allen Kollegen der Technischen Informatik bedanken, insbesondere bei Dr. Andreas Kosmala, Dr. Stefan Müller, Frank Wallhoff und Steffen Werner für das Korrekturlesen der Arbeit und für viele anregende Diskussionen zu meinem Dissertationsthema.

Anja Brakensiek

Inhaltsverzeichnis

Formelzeichen und Abkürzungen

Verwendete Formelzeichen

A	HMM-Übergangsmatrix
B	HMM-Emissionswahrscheinlichkeit (Matrix)
b_i	Emissionswahrscheinlichkeit im Zustand s_i
bo	Backoff-Faktor
$b(x, y)$	2-dimensionales Bild
C	explizite (zwanglose) Zeichensequenz
D	Dimension des Merkmalvektors
d	Discounting-Faktor
$d(\cdot, \cdot)$	Abstand
$f(x, y)$	Abtastpunkte
$g(\cdot)$	Gaußsche Normalverteilung
H	Entropie
I	Transinformation
J	Größe eines Codebuches
K	Modellanzahl, Symbole
$K(y), K(x)$	Histogramm
L	logarithmierte Likelihood
M	Matrix, Regressionsmatrix
N	Anzahl
O	diskrete Beobachtungsfolge
P	Wahrscheinlichkeit
p	Wahrscheinlichkeitsdichte
pp	Perplexität
$P(X\|\lambda)$	Likelihood
q	Folge von Zufallsvariablen
R	Clustergröße einer Regressionsmatrix
S	Sequenz von Zuständen
s_i	i-ter HMM Zustand

T	Anzahl der Merkmalvektoren
V	Ausgabealphabet (diskret)
W	Wort bzw. Zeichensequenz
w_i	Zeichen-Label ($\hat{=}$ HMM)
X	Sequenz von Merkmalvektoren
$\underline{x}$	Merkmalvektor
x_i	i-te Komponente des Merkmalvektors
Y	Sequenz von VQ-Labeln
y_j	Label der j-ten Codebuchpartition
α_t, β_t	Vorwärts-, Rückwärtswahrscheinlichkeit
α, Φ	Winkel
λ	HMM-Parameter
ω	Gewicht
$\underline{\pi}$	HMM-Anfangszustandswahrscheinlichkeit
$\underline{\mu}$	Mittelwertvektor
Σ	Kovarianzmatrix
σ_{ij}	Element der Kovarianzmatrix
τ	Schwelle

Verwendete Abkürzungen

ACC	Akkuratheit
ABR,ANK	verschiedene Schreiber
JMR,VDM	verschiedene Schreiber
AW	Adaptionsworte
BCC	Binary Connected Components
CCA	Connected Components Analyse
CMU-Cambridge -Toolkit	Software zur N-Gramm Generierung der Carnegie-Mellon- und der Cambridge-Universität
Conf	Konfidenzmaß
COR	Erkennungsrate
DCT	Diskrete Cosinus Transformation
DIS	Diskrete Modellierungstechnik
EM	Expectation Maximization
ERR	Fehlerrate
FAL	Anteil der falsch klassifizierten Daten
FAR	Anteil der falsch erkannten Daten, die nicht zurückgewiesen wurden
FRR	Anteil der korrekt erkannten Daten, die zurückgewiesen wurden
HAL	Postamt Halle

HAM	Postamt Hamburg
HMM	Hidden Markov Modell
HRO	Postamt Rostock
HTK	Hidden Markov Model Toolkit (Cambridge)
HYB	Hybride Modellierungstechnik
KON	Kontinuierliche Modellierungstechnik
LDA	Lineare Diskriminanz Analyse
Mix	max. Anzahl der Gaußverteilungen je Zustand
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information, maximale Transinformation
MMK	Mensch-Maschine-Kommunikation
MLP	Multi-Layer-Perceptron
NN	Neuronales Netzwerk
OCR	Optical Character Recognition
OOV	Out Of Vocabulary
PDA	Personal Digital Assistant
PLZ	Postleitzahl
RPROP	Resilient Backpropagation
REJ	Rückweisungsrate
SD	Siemens Dematic Konstanz
SEDAL	Systems Engineering and Design Automation Laboratory der Universität Sydney
SK	Semi-Kontinuierliche Modellierungstechnik
SLLR	Scaled Likelihood Linear Regression
STR	Postamt Stuttgart
TP	Tied Posterior
VQ	Vektor-Quantisierer
WB	Wörterbuch
diag	Diagonalmatrix
üb	überwacht
unüb	unüberwacht
voll	vollbesetzte Matrix
wi	schreiberunabhängig (writer-independent)
wd	schreiberabhängig (writer-dependent)

Kapitel 1

Einleitung

In den letzten Jahren hat die automatische Mustererkennung für die Mensch-Maschine-Kommunikation (MMK) deutlich an Bedeutung gewonnen. Die Ziele sind zum einen die benutzerfreundliche Interaktion zwischen Menschen und Computern und zum anderen die schnellere und effektivere Verarbeitung und Auswertung durch die Automatisierung oder Anwendungen im Multi-Media-Bereich. Die übliche MMK über die Tastatureingabe und eine bildschirmbasierte Textausgabe soll durch natürlichere Kommunikationsformen ersetzt werden. Dazu gehören die automatische Sprachsynthese und -erkennung (z.B. Telefonauskunft), die Symbol- und Dokumenterkennung (z.B. Formularleser und Bilddatenbanken, vgl. [Jun98, Mül01]), die Handschrifterkennung sowohl im on-line (elektronische Notizbücher) als auch im off-line Bereich (z.B. Postautomatisierung), die automatische Personen- und Gesichtserkennung (beispielsweise für Zugangsberechtigungen, siehe [Bra97, Wal01b]) und die Gestikerkennung (z.B. zur Interaktion mit autonomen Robotern [Bra98, Eic98]).

Das Thema dieser Arbeit sind verschiedene Aspekte zur automatischen Schrifterkennung – im wesentlichen von handschriftlichen Daten, aber auch von gedruckten Dokumenten – deren Motivation und Problematik in den nächsten Abschnitten näher erläutert wird.

1.1 Anwendungen und Ziele der Schrifterkennung

Das Anwendungsgebiet der automatischen Schrifterkennung ist weit gestreut und umfaßt sowohl die direkte MMK im on-line Bereich (Handschrifterkennung) als auch die Erfassung und Verarbeitung von Schriftstücken (z.B. Briefe) oder Dokumenten (off-line). On-line bedeutet in diesem Zusammenhang die Nutzung der zeitlichen Information, also die Trajektorie des Schriftzuges. Im Gegensatz dazu geht die off-line Erkennung von einem Bild aus. Die Ziele sind eine Steigerung der Benutzerfreundlichkeit durch Stift-basierte Eingabemedien und eine Erhöhung des Automatisierungsgrades für die schnelle und effiziente Bearbeitung und Erkennung von großen Mengen von Schriftstücken.

Neben der bekannten OCR (optical character recognition) von maschinengedruckten und digitalisierten Zeichen und der Erkennung von handgeschriebenen Einzelbuchstaben (Blockschrift) spielt neuerdings die Erkennung von kursiver Fließschrift eine immer größere Rolle. Die verschiedenen Anwendungsgebiete lassen sich abhängig vom Eingabemedium in die folgenden Kategorien einteilen:

- On-line Handschrifterkennung:
  Personal Digital Assistent (PDA: Termine, Notizen, etc.), Digitalisiertablett, Notebooks, elektronischer Schreibstift

- Off-line Handschrifterkennung:
  handschriftliche Notizen, Adreßerkennung (Postautomatisierung), ausgefüllte Formulare (Überweisungen, Anmeldungen)

- Dokumenterkennung (Maschinenschrift, OCR):
  Archivierung (Zeitungen, Rechnungen), Themenfindung in Datenbanken, Formulare, Adreßerkennung

Die heutigen kommerziellen OCR-Programme (z.B. Omnipage Pro, Textbridge Pro, Finereader) für maschinengedruckte und eingescannte Textseiten in guter Qualität liefern bereits sehr zufriedenstellende und somit auch nutzbare Erkennungsergebnisse von über 99.9% (unter der Voraussetzung, daß die Textlokalisierung, z.B. in Bildern und Tabellen, gelingt). Auch die Erkennung von handgeschriebenen Einzelzeichen bzw. Symbolen wird heute schon in PDAs und Formularlesern erfolgreich eingesetzt. Allerdings spielt hier häufig eher die Lernfähigkeit des Benutzers, der sich an eine bestimmte Schreibweise (die vom System erkannt wird) anpassen muß, eine wesentliche Rolle. Probleme ergeben sich, sobald Worte aus verbundenen Zeichen erkannt werden sollen. Hier ist eine Segmentierung der Worte in Einzelzeichen schwierig, was zu einer erhöhten Fehlerrate führt. Dies ist sowohl bei der kursiven Handschrift (Schreibschrift) als auch bei 'verschmierten' Dokumenten (Fotokopie, Fax) der Fall. Zwar gibt es auch hier erste kommerzielle Anwendungen (z.B. Briefsortierung von Siemens Dematic (SD)), die Erkennungsergebnisse sind jedoch deutlich geringer als die der OCR, sodaß hier noch ein hohes Potential an Verbesserungsmöglichkeiten liegt. Ein weiterer Punkt ist der mögliche Ausbau der Anwendungen für die Handschrifterkennung, für die bisher häufig nur die hardwaremäßigen Voraussetzungen geschaffen wurden (Digitalisiertablett: z.B. WACOM [WAC], elektronischer Schreibstift [FG01]), nicht jedoch die Software zur Verarbeitung (z.B. bei Notebooks: Thinkpad TransNote [IBM01]). Häufig können die handschriftlichen Notizen zwar als Bild abgespeichert und abgerufen werden, sie werden jedoch selten (nur bei Einzelbuchstaben) in maschinenlesbare Zeichen übersetzt, wodurch die weitere Handhabung der Information sich extrem vereinfachen würde. Eine Ausnahme bildet in diesem Bereich die Firma Paragraph [Par02], die neben der Erkennung von Druckschrift in natürlicher Schreibweise auch die Erkennung von Schreibschrift für Pocket- oder

Handheld-PCs anbieten (zur sichereren Erkennung wird jedoch die Druckschrift empfohlen, vgl. z.B. auch Apple Newton).

Bei den unterschiedlichen Anwendungsgebieten lassen sich jeweils verschiedene Problematiken unterscheiden:

- Lokalisierung, Vorverarbeitung und Merkmalextraktion der Schrift
- Erkennung von Einzelzeichen, Worten oder Sätzen
- Segmentierungseigenschaften (Blockschrift, Fließschrift, verbundene oder durchtrennte Schriftzeichen aufgrund geringer Qualität oder Auflösung)
- Anzahl verschiedener Fonts oder Schreiber (schreiberabhängig oder -unabhängig, Adaptionsmöglichkeiten)
- Auswahl eines Wörterbuches (Größe) oder Sprachmodells, Grammatik

Auf ausgewählte Teilprobleme und unterschiedliche Anwendungsbereiche soll in dieser Arbeit näher eingegangen werden (siehe Kap.1.2). Den Ausgangspunkt bildet jeweils die Erkennung verbundener Zeichen bzw. kontinuierlicher Fließschrift.

Gerade durch die Betrachtung kontinuierlicher Schrift, die sich nicht ohne weiteres in Einzelbuchstaben segmentieren läßt, ergibt sich die Ähnlichkeit zur Spracherkennung. In der Spracherkennung, aber auch in zunehmendem Maße in der Schrifterkennung, sind Hidden Markov Modelle (HMMs) die am meisten verwendete Methode zur Modellierung und Erkennung.

1.2 Beiträge und Gliederung der Arbeit

Die Beiträge dieser Arbeit beziehen sich schwerpunktmäßig auf die Handschrifterkennung (Fließschrift) von Worten im on- und off-line Bereich. Ergänzend werden aber auch maschinengedruckte Dokumente in geringer Qualität bzw. Auflösung in die Ausarbeitung mit einbezogen. Das gemeinsame Problem der hier untersuchten Anwendungsfelder ist eine Schrift, die sich nur schwer in Einzelzeichen segmentieren läßt und daher die Anwendung von Hidden Markov Modellen erforderlich macht.

In Kapitel 2 werden zunächst die allgemeinen Grundlagen und Prinzipien der automatischen Schrifterkennung, die auf HMMs beruht, erklärt. Dazu gehören die Methoden der Vorverarbeitung, Merkmalextraktion und Klassifikation. Kapitel 3 beschreibt die in dieser Arbeit verwendeten Datenbasen und die zugehörigen Merkmalextraktionsverfahren. Die Normierung und Auswahl der Merkmale wird dabei jeweils auf die speziellen Charakteristika des jeweiligen Schrift-Typs abgestimmt. Die untersuchten Anwendungsfelder betreffen die schreiberabhängige und schreiberunabhängige on-line Handschrifterkennung (am

Fachgebiet erstellte Datensammlung), die schreiberabhängige (vom Fachgebiet) und schreiberunabhängige off-line Handschrifterkennung (Adreßerkennung, Datenbank von Siemens Dematic (SD)) und die Erkennung gedruckter Dokumente (öffentlich zugängliche SEDAL-Datenbasis). Anhand dieser Datenbasen werden die verschiedenen Aspekte, wie die Modellierungstechniken mit HMMs, die Verwendung von Sprach- und Kontextmodellen und die Adaptionsverfahren, vorgestellt. Hybride Modellierungsverfahren, die eine Kombination von HMMs mit künstlichen Neuronalen Netzen darstellen, werden in Kapitel 4 erläutert. Kapitel 5 stellt die Grundlagen von Sprachmodellen, die aus der Spracherkennung bekannt sind, und ihre Verwendung für die Schrifterkennung dar. Der Schwerpunkt liegt hier auf der Erstellung von N-Grammen auf Buchstabenebene, die eine Alternative zur Wörterbuch-basierten Erkennung darstellen. Der Vergleich dieses Verfahrens mit unbegrenztem Vokabular mit der Erkennung anhand Wörterbüchern verschiedener Größe wird bei den Versuchsdurchführungen erörtert. Eine weiterer Aspekt in der HMM-basierten Schrifterkennung sind Kontextmodelle, die auf Merkmalebene den Kontext oder auch Übergang eines Zeichens zu seinen Nachbarn im Wort beschreiben (Kapitel 6). Die Sicherheit oder Vertrauenswürdigkeit des Erkennungsergebnisses ist bei HMM-basierten Systemen nicht direkt abzulesen. Dazu müssen Konfidenzmaße (Kapitel 7) bestimmt werden. In Kapitel 8 werden schließlich unterschiedliche Adaptionsverfahren vorgestellt mit deren Hilfe die schreiberunabhängigen Modelle auf einen bestimmten Schreiber (on-line Handschrift) oder auch auf eine lokale Gruppe von Schreibern (Adreßerkennung) angepaßt werden sollen um die Erkennungsleistung zu erhöhen. Für den sogenannten unüberwachten Modus sind Konfidenzmaße erforderlich. Die Versuchsdurchführungen und erzielten Ergebnisse mit den verschiedenen Datenbasen werden anschließend in Kapitel 9 vorgestellt.

Den Abschluß bildet Kapitel 10 mit einer Zusammenfassung und Bewertung der vorgestellten Arbeit. Im Anhang werden einige verwendete Formeln und zusätzliche Beispiele und Ergebnisse der Datenbanken gezeigt.

Kapitel 2

Grundlagen der automatischen Schrifterkennung

Kommerzielle OCR-Systeme zur Erkennung gedruckter und eingescannter Texte werden seit Jahren benutzt. Probleme stellen aber auch heute noch die Erkennung von Handschrift, insbesondere Schreib- bzw. Fließschrift, oder allgemein die Erkennung verbundener Zeichenketten dar. Wesentliche Techniken für die Handschrifterkennung sind an die Methoden der Spracherkennung [Lef01, ST95, Wil00a] angelehnt. Ein guter Überblick über den aktuellen Stand der Handschrifterkennung wird auch in [Pla00, Ste99] gegeben.

Die Prinzipien der automatischen Schrifterkennung sollen hier anhand von Abb. 2.1 erläutert werden. Der Erkennungsprozeß besteht, unabhängig von der Art der Eingabe, im wesentlichen aus der Vorverarbeitung, der Merkmalextraktion und der Klassifikation. Dies gilt nicht nur für die Schrifterkennung, sondern auch für die Spracherkennung oder allgemeine Mustererkennungsaufgaben.

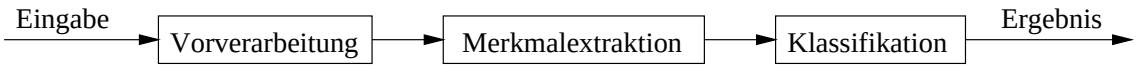


Abbildung 2.1: Prinzip der automatischen Mustererkennung

Die Eingabemedien bei der Schrifterkennung bestehen entweder aus Papier (off-line Schrifterkennung, siehe z.B. [Mar99]), welches durch Einscannen oder Aufnehmen per Kamera digitalisiert werden muß, oder aus einem elektronischen Grafiktablett bzw. Stift (on-line Erkennung, vgl. [Guy94, Kos00a, Man98]), wobei der Datenstrom des Schriftverlaufes direkt übertragen wird. Aus der Art des Eingabemediums wird sofort ersichtlich, daß es sich bei der on-line Schrifterkennung nur um Handschrift (Block- oder Fließschrift) handeln kann. Im off-line Bereich bestehen die Schriftproben hingegen aus handschriftlichen oder maschinengedruckten Worten. Der Unterschied zwischen on- und off-line Daten soll anhand von Abb. 2.2 erläutert werden.

Abbildung 2.2: On- und off-line Daten

Bei den on-line Daten liegen Abtastpunkte der Trajektorie des Schriftzuges vor. In diesem Datenstrom ist also ebenfalls die zeitliche Information gespeichert, die bei der Verwendung von off-line Daten verloren geht. Off-line Daten bestehen aus Binär- oder Grauwertbildern. Im allgemeinen sind gerade bei der Handschrifterkennung die Fehlerraten bei on-line Daten geringer als bei vergleichbaren off-line Daten, da die zeitliche Information hier genutzt werden kann. Prinzipiell läßt sich jede on-line Erkennung in eine off-line Erkennung überführen (siehe auch [Bra99b, Bra99a, Sei96]), indem die Stift-Trajektorie als Bild abgespeichert wird. Die Umkehrung, also den Versuch, aus einem eingescannten Bild den zeitlichen Ablauf des Schriftzuges wiederherzustellen, ist dagegen weitaus schwieriger (vgl. z.B. [Doe92, Lal00]).

Die Verfahren zur Vorverarbeitung und Merkmalextraktion der Daten sind stark abhängig sowohl vom Eingabemedium als auch der Art der Schrift. Verschiedene Varianten zur Abtastung, Bildverbesserung und Normalisierung, sowie unterschiedliche Arten von Merkmalen werden in Kap. 2.1 näher beschrieben. Ziel der Vorverarbeitung ist es, die großen Variationen in den Schriften und Schreibweisen zu reduzieren (siehe auch [Sch99, Sri01]). Als weiterer Punkt kommt bei der off-line Erkennung die Lokalisation und Segmentierung der zu erkennenden Textpassage hinzu (beispielsweise Zeitungsseiten mit Bildern, Tabellen, Textspalten), worauf in dieser Arbeit jedoch nicht näher eingegangen werden soll.

Der letzte Schritt des Erkennungsvorganges ist die Klassifikation. Zur Klassifikation gehören nicht nur die Zuordnung und Erkennung der Zeichen aufgrund der extrahierten Merkmale sondern auch die Einbeziehung von Grammatiken, Sprachmodellen und Wörterbüchern. Die analytischen Ansätze zur Schrifterkennung gliedern sich in Verfahren mit expliziter und impliziter Zeichensegmentierung. Standard OCR-Verfahren arbeiten in der Regel mit einer expliziten Segmentierung der Worte in Einzelzeichen, bevor die Erkennung der Zeichen über einen Vergleich mit Prototypen stattfindet. Diese Vorgehensweise ist bei gedruckten Zeichen oder auch handgeschriebenen Druckbuchstaben, die nicht fließend ineinander übergehen, sehr effektiv. Bei der Erkennung von Fließschrift sind jedoch implizite Verfahren, bei denen Segmentierung und Erkennung zur gleichen Zeit ablaufen (z.B. Hidden Markov Modelle), im Vorteil. Eine weitere Kategorie sind ganzheitliche Ansätze, die versuchen, ein Wort als Ganzes ohne Segmentierung zu erkennen. Diese Verfahren eignen sich zwar ebenfalls für Fließschrift (verbundene Zeichenketten), hier steigt allerdings im

Gegensatz zum analytischen Ansatz der Aufwand mit der Größe des Vokabulars extrem an. Die Klassifikation, die in der vorliegenden Arbeit auf der Verwendung von Hidden Markov Modellen und Sprachmodellen beruht, kann mit Hilfe des Satzes von Bayes durch die folgende Gleichung 2.1 beschrieben werden, die als Fundamentalgleichung für die Schrift- und Spracherkennung gilt:

$$W^* = \underset{W}{\operatorname{argmax}}\, P(W|X) = \frac{\underset{W}{\operatorname{argmax}}\, P(W) \cdot P(X|W)}{P(X)} = \underset{W}{\operatorname{argmax}}\, P(W) \cdot P(X|W) \quad (2.1)$$

Gesucht wird für die Eingabe X das wahrscheinlichste Wort (oder Zeichenkette, oder Satz) W^*. Dabei stellt $P(W)$ das Sprachmodell dar und $P(X|W)$ die bedingte Wahrscheinlichkeit, daß die Merkmalvektorfolge X von einer bestimmten Kette von Hidden Markov Modellen, die das Wort W repräsentieren, erzeugt wird. Zu beachten ist bei der HMM-basierten Erkennung, daß zwar das wahrscheinlichste Wort als Ergebnis ausgegeben wird, nicht jedoch unmittelbar die Wahrscheinlichkeit für eine korrekte Erkennung (Konfidenzmaß). Diese muß ggf. zusätzlich zur Erkennung berechnet werden. Die Verwendung von Hidden Markov Modellen und die Sprachmodellierung wird in den Kapiteln 2.2 und 2.3 näher erklärt. Die Ausgabe einer Erkennung ist jeweils eine maschinenlesbare Wort- oder Zeichenfolge (z.B. im ASCII-Format), die ggf. mit einer Wahrscheinlichkeit (Konfidenzmaß, Likelihood) versehen ist. Verschiedene Konfidenzmaße werden in Kap. 7 vorgestellt.

Neben der reinen Schrifterkennung ergeben sich darauf aufbauend weitere Anwendungsfelder, wie zum Beispiel ein handschriftlicher Formeleditor [Kos00b] (vgl. auch [Eis93]), die Themenfindung (‘Topic-Spotting’, ‘Retrieval’) in Bilddatenbanken oder Archiven [Kan00, Iur01], die Unterschriftenverifikation [Yan95, Rig98a] oder Multi-Media-Anwendungen [Hun00].

2.1 Vorverarbeitung und Merkmalextraktion

Grundlegendes Ziel der Vorverarbeitung ist es, Störungen der Eingabedaten zu verringern und die Variation der Schreibweisen oder verschiedener Fonts zu reduzieren. Die nachfolgende Merkmalextraktion soll möglichst die für die Erkennung relevanten Eigenschaften der Schrift bestimmen. Vorerst soll Abb. 2.3 die im weiteren verwendeten Begriffe anhand eines idealen Schriftbeispiels veranschaulichen.

Die untere Basislinie gibt die Lage der Grundlinie an, die obere Basislinie die obere Begrenzung der Kleinbuchstaben wie z.B. ‘a,e,m’. Als Kernhöhe wird der Bereich zwischen den Basislinien bezeichnet, die zusammen mit der Ober- und Unterlänge die gesamte Höhe eines Wortes beschreibt.

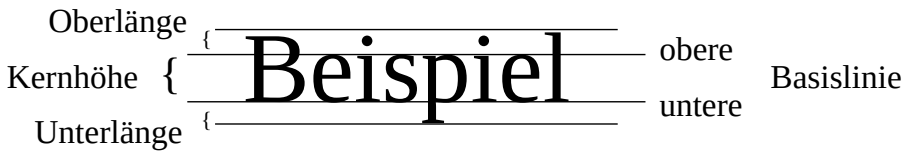


Abbildung 2.3: Definitionen

Vorverarbeitung

Zur Vorverarbeitung gehören im wesentlichen, abhängig von der Art der Eingabedaten, die folgenden Verfahren:

- Abtastung des Stift-Trajektorie bei on-line Daten
- Bearbeitung zeitverzögerter Striche (z.B. i-Punkte oder t-Striche, die evtl. erst am Ende des Wortes eingefügt werden) bei on-line Daten
- Beseitigung von Störungen (Rauschen, Dokument-Qualität, Schmutz), die durch das Scannen entstehen oder bereits durch die Vorlage verursacht werden (off-line)
- Skelettierung oder Kontur (nur bei off-line Daten sinnvoll)
- Bestimmung der unteren und oberen Basislinie (Grundlinie und Kernhöhe)
- Normierung der Größe, der Zeilenschräglage (skew) und der Zeichenneigung (slant)

Bei on-line Daten treten andere Effekte auf, als bei off-line Daten. So ist z.B. die Abtastung der Stift-Trajektorie notwendig, um unterschiedliche Schreibgeschwindigkeiten – die keinen Einfluß auf die Wortklasse haben – zu kompensieren. Auch der Zeitpunkt, wann abgesetzte Striche (t-Striche, i-, ü-Punkte, etc.) erfolgen, ist ggf. ein wichtiges Entscheidungsmerkmal für die Unterschriftenverifikation, spielt aber für die Erkennung keine Rolle. Andererseits treten Störungen, die das Schriftbild beeinflussen (Scanner, Kopierer, Faxgerät) nur bei off-line Daten auf. Auch eine Skelettierung (bzw. Verdünnung) oder Kontur-Ermittlung der Schriftdicke (abhängig vom Stift) ist nur bei off-line Daten eine sinnvolle Reduktion der Schreibvariationen. On-line Daten liegen häufig bereits skelettiert vor.

Die Normalisierungsverfahren hingegen betreffen sowohl on-line als auch off-line Daten. Hier ist eher die Art der Anwendung, also schreiberab- oder -unabhängig ausschlaggebend. Für ein schreiberabhängiges System sind die Schrifteigenschaften wie Größe und Zeichenneigung in der Regel gleich bleibend und müssen daher nicht notwendigerweise normiert werden. In einem schreiberunabhängigen System sind die Unterschiede zwischen den Schreibern zu groß, um eine Normierung zu umgehen. Die Normierungsschritte sind im allgemeinen von der Lage der Basislinien abhängig. Die obere und untere Basislinie kann oft nicht so eindeutig festgelegt werden, wie in Abb. 2.3 dargestellt wird. Bei der Handschrift,

aber auch bei kopierten und gefaxten Dokumenten bilden sich die Basislinien nicht parallel und häufig auch nicht als deutliche Gerade aus. Diese ‘Schlangenlinien’ können entweder direkt bestimmt werden (z.B. [Ben95]: Neuronales Netzwerk) oder durch Geraden approximiert werden [Cae93]. Die Zeilen- bzw. Wortschräglage kann anhand der unteren Basislinie durch eine Drehung korrigiert werden und die Größennormierung erfolgt anhand der Kernhöhe (oder auch implizit aufgrund der Art der Merkmalsextraktion). Ein Beispiel für die Normierung der Zeichenneigung zeigt Abb. 2.4.

Abbildung 2.4: Korrektur der Zeichenneigung

Die Zeichenneigung variiert stark zwischen verschiedenen Schreibern, insbesondere zwischen Rechts- und Linkshändern. Aber auch bei kursiv gedruckten (‘italic’) Zeichen tritt das Problem auf.

Merkmalextraktion

Bei der Wahl der Merkmale für die Schrifterkennung gibt es bisher – im Gegensatz zur Spracherkennung, bei der in der Regel Mel-Cepstrum-Koeffizienten [ST95] als Merkmale verwendet werden – kein dominierendes Verfahren. Auch hier muß, wie bei der Vorverarbeitung, zwischen on- und off-line sowie zwischen Wort- und Einzelzeichenerkennern unterschieden werden. Die Bearbeitung und Erkennung von Einzelzeichen soll in dieser Arbeit jedoch keine Rolle spielen, weshalb an dieser Stelle z.B. auf [Tri96, Bra96, Vuo01] verwiesen wird.

Eine Gemeinsamkeit bei der on- und off-line Erkennung ist die ‘sliding-window’ Technik, die für die Merkmalextraktion verwendet wird. Das heißt, ein lokales Fenster wird entlang des Schriftzuges (Trajektorie) oder über das Bild geschoben, innerhalb dessen dann die Merkmale extrahiert werden. Diese lokalen Merkmale können durch globale Merkmale, die das Wort als Ganzes betreffen (z.B. Worthöhe), ergänzt werden.

Mögliche Merkmale bei der on-line Handschrifterkennung, die in der Literatur vorgestellt werden, sind z.B. die folgenden (siehe [Ben95, Kos97a, Man98, Rig98c, Hu98, Jae01]):

- Lokale Höhe über der Basislinie
- Schreibrichtung von aufeinanderfolgenden Abtastpunkten
- Änderung der Schreibrichtung, Krümmung
- Stiftdruck

- Ober- und Unterlängen eines Wortes
- Bitmap des Fensters (unterabgetastet, DCT-Koeffizienten, etc.), das entlang der Trajektorie geschoben wird

Für die off-line Erkennung gibt es entsprechend die folgende Auswahl von Merkmalen (vgl. z.B. [Cae93, Bun95, Sao97, Côt98, Bip99, Bra00a, Wan00, Wan01]), die häufig verwendet werden:

- Bitmap des Fensters, das horizontal über die Schrift geschoben wird
- Lokale Höhe über und unter der Basislinie
- Bestimmung von Krümmungen, Schlaufen, Orientierungen, Scheitelpunkten
- DCT- oder Fourier-Koeffizienten, Momente
- Transformation oder Kompression von Merkmalen mittels Neuronaler Netze (NN)

Diese Merkmale (on- und off-line) werden in Merkmalvektoren $\underline{x}$ zusammengefaßt, deren Sequenz (zeitliche bzw. räumliche Vektorfolge) für das Training und die Erkennung der Hidden Markov Modelle verwendet wird. Alternativ können die Merkmalvektoren zusätzlich weiteren Transformationen (z.B. LDA: Lineare Diskriminanzanalyse, KLT: Karhunen-Loeve Transformation, Bildung von Differenzvektoren, Kombination benachbarter Merkmalvektoren) unterzogen werden, bevor sie zur Klassifikation herangezogen werden.

2.2 Hidden Markov Modelle

Hidden Markov Modelle (HMMs, siehe [Rab86, ST95, You00]) haben sich in den letzten Jahrzehnten hauptsächlich in der Spracherkennung, aber auch in der Handschrifterkennung etabliert. Ein wesentlicher Vorteil der Klassifikation über HMMs ist, daß die zu erkennende Sprach- oder Schriftsequenz nicht explizit segmentiert werden muß. Auch die Länge des Signals (Anzahl der Merkmalvektoren) kann variieren. Die statistische Modellierung mit HMMs setzt die extrahierte Merkmalvektorfolge $X = \underline{x}_1, ..., \underline{x}_T$ (bzw. im diskreten Modus deren Klassenindizes y_i) voraus. Das Ziel ist, die Verteilungsdichte $P(X|W)$ nach Gl. 2.1 für ein vorgegebenes Vokabular W durch das HMM λ optimal zu modellieren. Bei der Erkennung sollte dann $P(X|\lambda_i)$ für das Symbol w_i maximal sein.

In der Schrifterkennung wird in der Regel für jedes Zeichen (Buchstaben, Zahlen, etc.) ein HMM trainiert und in der Spracherkennung ein HMM für jedes Phonem. Durch die Zusammensetzung mehrerer HMMs werden dann Worte oder Sätze gebildet. Es gibt allerdings auch Ansätze, die ein HMM pro Wort trainieren, was bei einem großem Vokabular fast unmöglich wird, bzw. zu einem sehr hohen Aufwand führt.

Vorher festgelegt werden muß die Topologie der HMMs (z.B. ergodisch, linear), die Anzahl N der Zustände s und die Art der Ausgabeverteilung (diskret, kontinuierlich, semikontinuierlich). Kapitel 2.2.1 gibt die Definitionen der HMM-Parameter an. Die Art der Ausgabeverteilungen, die in dieser Arbeit ebenfalls eine Rolle spielen (vgl. Kap. 4 und 8) werden im Abschnitt 2.2.2 näher beschrieben. Das Training der HMMs nach dem Baum-Welch Verfahren wird in Kap. 2.2.3 und die Erkennung nach dem Viterbi-Algorithmus wird in Kap. 2.2.4 kurz erklärt. Eine detailliertere Beschreibung der wesentlichen Standard-Algorithmen ist in [Rab86, ST95] zu finden. Hier werden lediglich die Verfahren im Detail erläutert, die im Rahmen dieser Arbeit variiert wurden oder auf die explizit aufgesetzt wurde.

2.2.1 Grundlagen

Eine Hidden Markov Modell λ basierte Erkennung ist durch die Anzahl N der Zustände s_i (mit $i = 1, ..., N$), den Wortschatz W und den folgenden Parametersatz bestimmt:

$$\lambda = (\underline{\pi}, A, B) \tag{2.2}$$

Definitionen

Zur Erläuterung betrachte man vorerst einen diskreten stochastischen Prozeß q. Die Folge $q = q_1, ..., q_T$ von Zufallsvariablen kann verschiedene Zustände s_i einnehmen. Dabei beschreibt π_i die Wahrscheinlichkeit für die Einnahme des Anfangszustandes:

$$\pi_i = P(q_1 = s_i) \quad \text{mit} \quad \sum_{i=1}^{N} \pi_i = 1 \tag{2.3}$$

Die Übergangsmatrix A beschreibt die Wahrscheinlichkeiten für den Wechsel von einem Zustand in einen anderen mit

$$A = [a_{ij}]_{N \times N} \quad \text{mit} \quad a_{ij} = P(q_t = s_j | q_{t-1} = s_i) \tag{2.4}$$

wobei für a_{ij} die Stochastizitätsbedingungen $a_{ij} \geq 0$ und $\sum_j a_{ij} = 1$ gelten. Die Voraussetzung hierfür ist ein einfacher kausaler stationärer Prozeß, für dessen Übergangswahrscheinlichkeiten gilt (Markoveigenschaft 1. Ordnung):

$$P(q_t | q_1, ..., q_{t-1}) = P(q_t | q_{t-1}) \tag{2.5}$$

Das heißt, nur der letzte vorherige Zustand wird berücksichtigt. Dies ist für die Schrift- oder Spracherkennung eigentlich eine ungültige Annahme (wird dennoch standardmäßig verwendet), die die Aufgabe jedoch erheblich vereinfacht. Im allgemeinen Fall können alle Elemente der Übergangsmatrix A besetzt sein (ergodisches HMM). Für lineare Links-Rechts-Modelle, die in der Schrift- und Spracherkennung häufig verwendet werden, gilt:

$$a_{ij} = 0 \quad \text{falls:} \quad (j < i) \quad \text{oder} \quad (j > i + 1) \tag{2.6}$$

In einem Links-Rechts-Modell sind keine Übergänge in ältere Zustände erlaubt, d.h. es gibt definierte Anfangs- und Endzustände ($\underline{\pi} = (1, 0, ..., 0)^T$), die eine zeitliche (bzw. räumliche) Ordnung beinhalten. Linear bedeutet in diesem Zusammenhang, daß jeder Zustand mindestens einmal eingenommen werden muß. Eine weitere übliche HMM-Topologie ist das Bakis-Modell, ein Links-Rechts-Modell, bei dem bestimmte Zustände übersprungen werden dürfen.

Diskrete stochastische Prozesse, die nur mit Hilfe der Parameter π und A beschrieben werden, heißen Markovketten. Ergänzt man diesen ersten Prozeß um einen zweiten stochastischen Prozeß, der anhand eines Ausgabealphabetes $V = V_1, ..., V_K$ die diskrete Beobachtungsfolge $O = O_1, ..., O_T$ erzeugt, erhält man das Hidden Markov Modell λ mit den folgenden Ausgabeverteilungen, die nur vom aktuellen Zustand abhängen. Die Ausgabeverteilung B bzw. $\underline{b}$ unterscheidet sich nach der Modellierungsart (diskret oder kontinuierlich) und läßt sich für den diskreten Fall folgendermaßen darstellen:

$$B = [b_{jk}]_{N \times K} \qquad \text{mit} \quad b_{jk} = b_j(v_k) = P(O_t = v_k | q_t = s_j) \tag{2.7}$$

B muß wiederum die Stochastizitätsbedingungen erfüllen. Hier wird bei der Verwendung diskreter HMMs der Vektorquantisierer (VQ, Codebuch) für die kontinuierlichen Merkmalvektoren $\underline{x}$ so gewählt, daß die VQ-Label $Y = y_1, ..., y_J$ dem Ausgabealphabet V entsprechen (mit $J = K$).

Die verschiedenen Möglichkeiten für die Modellierung der Ausgabeverteilung verdeutlicht Abb. 2.5. Die kontinuierliche Ausgabeverteilungsdichte (siehe Kap. 2.2.2) wird entsprechend mit

$$b_j(\underline{x}) = P(O_t = \underline{x} | q_t = s_j) \tag{2.8}$$

definiert, wobei die Ausgabeverteilung der Normierungsbedingung $\int_x b_j(\underline{x}) dx = 1$ gehorchen muß. Abb. 2.5 zeigt ein lineares Links-Rechts-HMM mit drei Zuständen und unterschiedlichen Ausgabeverteilungen (diskret, kontinuierlich, semi-kontinuierlich).

Neben dieser Standardvariante eines eindimensionalen HMMs, gibt es auch Ansätze zweidimensionale bzw. Pseudo-zweidimensionale (Pseudo-2D, [Bip99, Bre01, Wal01a]) oder Pseudo-3D HMMs für Bildsequenzen zu verwenden (3 Dimensionen: x- und y-Ausdehnung des Bildes und die Zeit, siehe z.B. [Mül00]).

Produktionswahrscheinlichkeiten

Die Produktionswahrscheinlichkeit $P(O|\lambda)$ gibt die Wahrscheinlichkeit dafür an, daß die Beobachtungsfolge O vom HMM λ erzeugt wurde.
Sind die HMM-Parameter λ und die Beobachtungsfolge $O = O_1, ..., O_T$ bekannt, kann die Wahrscheinlichkeit für eine Zustandsfolge q mit

$$P(q|\lambda) = P(q_1, ..., q_T|\lambda) = \pi_{q_1} \cdot \prod_{t=2}^{T} a_{q_{t-1} q_t} \tag{2.9}$$

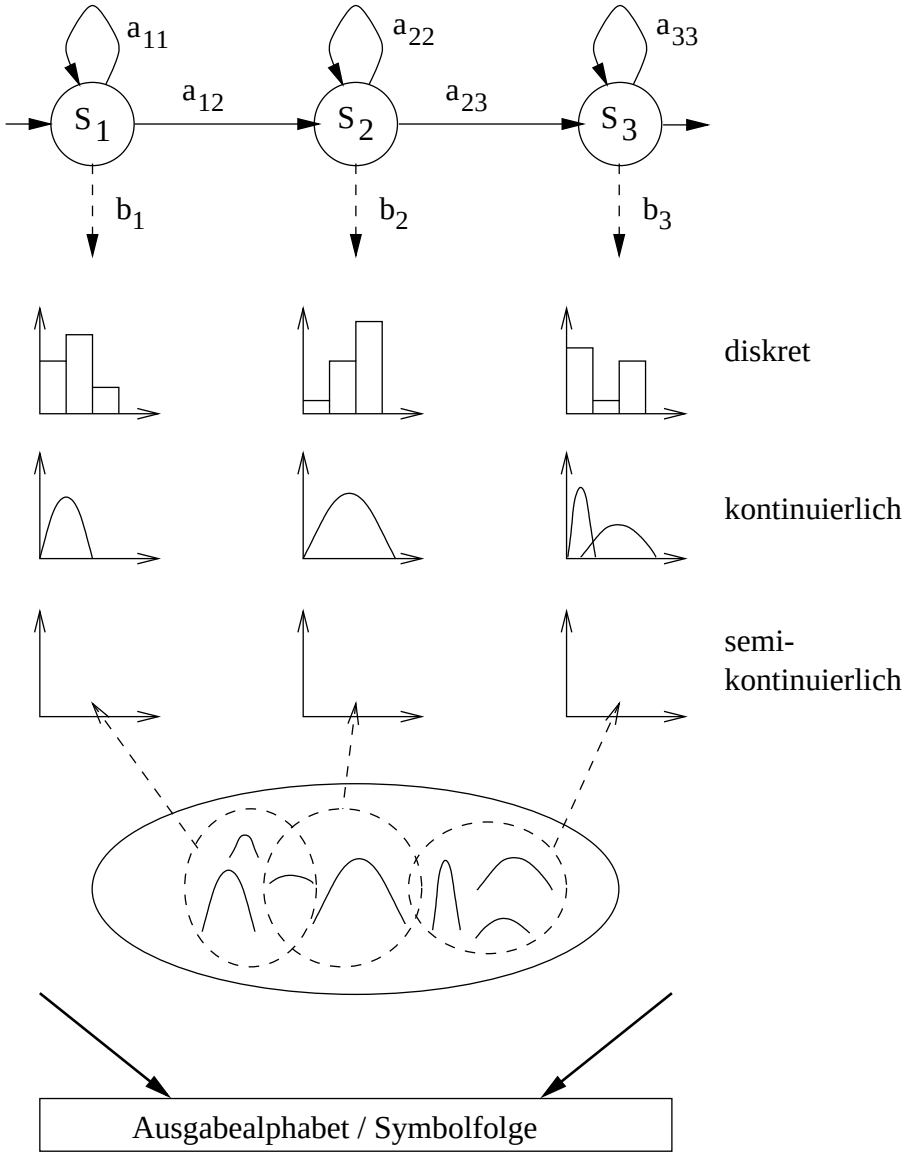


Abbildung 2.5: Lineares Links-Rechts-HMM mit unterschiedlichen Ausgabeverteilungen

beschrieben werden und die Wahrscheinlichkeit für O unter der Bedingung einer konkreten Zustandsfolge q als

$$P(O|q,\lambda) = P(O_1,...,O_T|q_1,...,q_T,\lambda) = \prod_{t=1}^{T} b_{q_t}(O_t) \tag{2.10}$$

Die Verbundwahrscheinlichkeit bei diskreter Modellierung ergibt sich dann zu:

$$P(O,q|\lambda) = P(O|q,\lambda) \cdot P(q|\lambda) \tag{2.11}$$

Die gesuchte Produktionswahrscheinlichkeit $P(O|\lambda)$ erhält man durch Summierung der Verbundwahrscheinlichkeiten aller (N^T) möglichen Zustandsfolgen:

$$P(O|\lambda) = \sum_{q} P(O, q|\lambda) = \sum_{q} \left(\pi_{q_1} b_{q_1}(O_1) \cdot \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(O_t) \right) \tag{2.12}$$

Da der Rechenaufwand zur Bestimmung von $P(O|\lambda)$ mit der Anzahl der Merkmalvektoren exponentiell wächst, werden zur effizienteren Berechnung *Vorwärts-* oder *Rückwärtswahrscheinlichkeiten* rekursiv bestimmt ('Forward-Backward'-Algorithmus [Rab86], siehe Kap. 2.2.3). Die Produktionswahrscheinlichkeit kann mit Hilfe der Vorwärtswahrscheinlichkeiten α_t folgendermaßen ausgedrückt werden:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j) \quad \text{mit:} \quad \alpha_t(j) = P(O_1, ..., O_t, q_t = s_j|\lambda) \tag{2.13}$$

Nach der Initialisierung mit $\alpha_1(j) = \pi_j b_j(O_1)$ kann rekursiv $\alpha_t(j)$ mit $t > 1$ folgendermaßen bestimmt werden (siehe Gl. 2.14):

$$\alpha_t(j) = \left(\sum_{i} \alpha_{t-1}(i) a_{ij} \right) b_j(O_t) \tag{2.14}$$

Die Rekursion wird beendet, wenn alle T Beobachtungen betrachtet wurden. Unter zu Hilfenahme der Rückwärtswahrscheinlichkeiten β_t ergeben sich folgende Gleichungen:

$$P(O|\lambda) = \sum_{j=1}^{N} \pi_j b_j(O_1) \beta_1(j) \quad \text{mit:} \quad \beta_t(i) = P(O_{t+1}, .., O_T|q_t = s_i, \lambda) \tag{2.15}$$

Hier verläuft die Rekursion in umgekehrter Richtung. Nach der Initialisierung mit $\beta_T(i) = 1$ wird $\beta_t(i)$ folgendermaßen bestimmt (Gl. 2.16):

$$\beta_t(i) = \sum_{j} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \tag{2.16}$$

Man kann zeigen, daß für die Wahrscheinlichkeiten die folgenden Beziehungen zu jedem Zeitpunkt t gelten, was für die in Kap. 2.2.3 erläuterte Trainingsmethode der HMMs eine wichtige Rolle spielt:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_t(j) \beta_t(j) \quad \text{und} \quad \alpha_t(j) \beta_t(j) = P(O, q_t = s_j|\lambda) \tag{2.17}$$

Für eine ausführliche Beschreibung der rekursiven Algorithmen zur Berechnung der Vorwärts- und Rückwärtswahrscheinlichkeiten sei auf [ST95] verwiesen. Die Komplexität der Berechnung verhält sich jetzt linear zur Anzahl T der Eingabevektoren.

2.2.2 Modellierung der Emissionswahrscheinlichkeiten

Die Ausgabe- oder Emissionswahrscheinlichkeiten (siehe Abb. 2.5) lassen sich prinzipiell als diskrete Ausgabeverteilung oder als kontinuierliche oder semi-kontinuierliche (tied-mixture) Verteilungsdichten darstellen. Die Merkmalvektorfolge X wird im (semi-) kontinuierlichen Fall direkt zur Modellierung der HMMs verwendet und für die diskrete Modellierung zunächst quantisiert. Weitere hybride Modellierungstechniken (Verknüpfung von HMM mit NN), die auf der diskreten oder semi-kontinuierlichen Modellierung basieren, werden in Kap. 4 erläutert.

Diskrete Modellierung

Diskrete HMMs modellieren Folgen diskreter Merkmale. Daher wird die Folge der Merkmalvektoren X mit einem Vektorquantisierer VQ in eine Folge von diskreten Symbolen Y abgebildet. Der Vektorquantisierer arbeitet in der Regel nach dem k-means- oder LBG-Verfahren (siehe [McQ67, Lin80]) und weist jedem Merkmalvektor $\underline{x}$ eine Codebuch-Partition y_j zu (mit $1 \leq j \leq J$), wobei J die Anzahl der Codebuch-Partitionen (Größe des Codebuchs) ist:

$$\underline{x} \stackrel{VQ}{\rightarrow} y_j \tag{2.18}$$

Die Codebuchgröße muß zuvor gewählt werden und die Zuweisung erfolgt anhand des kleinsten euklidischen Abstands des Merkmalvektors $\underline{x}$ zum Mittelpunktvektor einer Codebuch-Partition. Der Quantisierungsfehler, der entsteht, wird vernachlässigt und für die weiteren Berechnungen geht man von folgender Annahme aus:

$$p(\underline{x}|s) = p(y_j|s) \tag{2.19}$$

Somit ergibt sich die diskrete Ausgabeverteilung zu Gl. 2.20:

$$b_i(y_j) = P(y_j|s_i) = c_{ij} \quad \text{mit} \quad 1 \leq j \leq J \tag{2.20}$$

Ein Vorteil der diskreten Modellierung ist der geringe Berechnungsaufwand, der sich aus Gl. 2.20 ergibt, woraus die Möglichkeit einer schnellen Erkennung folgt. Der Nachteil besteht in der Quantisierung, wodurch ein Informationsverlust stattfindet. Dieser Quantisierungsfehler kann allerdings, wie in Kap. 4.1 beschrieben, durch den Einsatz Neuronaler Netze verringert werden.

Kontinuierliche Modellierung

Bei der kontinuierlichen Modellierungstechnik ist die Emissionswahrscheinlichkeit direkt durch kontinuierliche Verteilungsdichten $b_i(\underline{x})$ im Zustand s_i in Abhängigkeit vom Merkmalvektor $\underline{x}$ gegeben. In der Regel werden für die kontinuierlichen Ausgabeverteilungen

Gaußsche Normalverteilungen g_{ij} eingesetzt:

$$g_{ij}(\underline{x}) = p(\underline{x}|m_j, s_i) = \frac{1}{\sqrt{(2\pi)^D \cdot |\Sigma_{ji}|}} \cdot e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_{ji})^T \Sigma_{ji}^{-1} (\underline{x}-\underline{\mu}_{ji})} \tag{2.21}$$

Diese sind definiert durch den Mittelwert μ_{ji} und die Kovarianzmatrix Σ_{ji} der j-ten Gauß-Verteilung im Zustand s_i (D ist die Dimension des Merkmalvektors $\underline{x}$). Die gewichtete Überlagerung mehrerer Gauß-Funktionen (Anzahl J_i) führt zu Gaußschen Mischverteilungsdichten als Ausgabeverteilung:

$$b_i(\underline{x}) = p(\underline{x}|s_i) = \sum_{j=1}^{J_i} \omega_{ij} \cdot g_{ij}(\underline{x}) \tag{2.22}$$

Aus rechentechnischen Gründen wird häufig mit logarithmierten Wahrscheinlichkeiten gearbeitet. Für die Gewichte ω_{ij} muß gelten:

$$\omega_{ij} \geq 0 \quad \text{und} \quad \sum_j \omega_{ij} = 1 \tag{2.23}$$

Alle diese Parameter müssen aus den Merkmalvektoren, die für das Training zur Verfügung stehen, geschätzt werden.

Ein frei wählbarer Parameter bei dieser Modellierungstechnik ist die Art der Kovarianzmatrix: diagonale oder voll besetzte Matrix. Wird die Verteilung der Trainingsdaten durch eine diagonale Kovarianzmatrix angenähert, vereinfacht sich die Berechnung, da nach Gl. 2.21 die Inverse gebildet werden muß. Bei einer vollbesetzten Matrix spielt das Problem der Singularität eine große Rolle (große Menge an unabhängigen Trainingsvektoren erforderlich) und die große Anzahl zusätzlich zu schätzender Parameter. Die Art der Matrix kann sich aber auch nach der Art der Merkmalextraktion richten. So sind nach einer KLT die Merkmale unkorreliert und ergeben somit prinzipiell eine diagonale Kovarianzmatrix.

Semi-Kontinuierliche Modellierung

HMMs mit semi-kontinuierlicher Ausgabeverteilung schließen sowohl Aspekte der kontinuierlichen als auch der diskreten Modellierung mit ein. Ein Codebuch mit J Gaußfunktionen steht jedem Zustand zur Verfügung, wobei – wie bei der kontinuierlichen Modellierung – die Mischverteilungskomponenten über die Gewichte ω_{ij} bestimmt werden. Dies führt zu folgender Gleichung für die Ausgabedichte $b_i(\underline{x})$, die sich kaum von der kontinuierlichen Modellierung unterscheidet:

$$b_i(\underline{x}) = p(\underline{x}|s_i) = \sum_{j=1}^{J} \omega_{ij} \cdot g_j(\underline{x}) \tag{2.24}$$

Ein Vorteil der semi-kontinuierlichen Modellierung ist die im Vergleich zur kontinuierlichen Ausgabeverteilung kleinere Anzahl zu schätzender Parameter (Mittelwerte und Varianzen der Gaußfunktionen), da der Vorrat (‘Pool’) von Gaußfunktionen von allen HMMs gebildet bzw. benutzt wird.

2.2.3 Training

Training von HMMs bedeutet, die Modellparameter λ anhand von Trainingssequenzen O optimal zu schätzen. Die folgenden Formeln beziehen sich auf HMMs mit diskreter Ausgabeverteilung (mit N Zuständen und einem Ausgabealphabet mit K Symbolen), im Prinzip gelten die Überlegungen aber entsprechend auch für kontinuierliche Modellierungstechniken. Gesucht wird ein neuer Parametersatz λ^*, der die Maximum-Likelihood (ML) Zielfunktion $P(O|\lambda)$ maximiert:

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}}\, P(O|\lambda) \tag{2.25}$$

Dabei ist λ das Modell und O die Trainingssequenz für ein Wort W. Für das neue Modell λ^* soll dann gelten $P(O|\lambda^*) \geq P(O|\lambda)$. Die Wahrscheinlichkeit $P(O|\lambda)$ (manchmal auch Ähnlichkeitsmaß oder Plausibilität genannt), die jeweils als Ergebnis eines HMM-Trainings oder -Erkennung vorliegt, soll im weiteren mit dem üblichen englischen Begriff Likelihood bezeichnet werden.

Mit Hilfe des *Baum-Welch*- und *Forward-Backward-Algorithmus* werden die neuen Parameter λ^* iterativ nach dem EM-Verfahren (*expectation maximization*) geschätzt, wobei die Vorwärts- und Rückwärtswahrscheinlichkeiten verwendet werden. Der Baum-Welch Algorithmus soll im folgenden kurz erläutert werden: Es sei $\xi_t(i,j)$ die a posteriori Wahrscheinlichkeit eines Überganges vom Zustand s_i nach s_j entsprechend Gl. 2.26:

$$\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j|O,\lambda) = \frac{P(q_t = s_i, q_{t+1} = s_j, O|\lambda)}{P(O|\lambda)} \tag{2.26}$$

Der Ausdruck $P(O|\lambda)$ läßt sich nach Gl. 2.17 durch die Vorwärts- und Rückwärtswahrscheinlichkeiten ausdrücken, sodaß gilt:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \tag{2.27}$$

Zusammen mit der Zustandswahrscheinlichkeit γ_t mit

$$\gamma_t(i) = P(q_t = s_i|O,\lambda) = \sum_j \xi_t(i,j) \tag{2.28}$$

ergeben sich die Schätzformeln für den Baum-Welch [Bau68] Algorithmus (χ ist eine charakteristische Funktion (Kroneckerdelta), welche nur dann gleich 1 ist, wenn gilt $O_t = v_k$):

$$\pi_i^* = \gamma_1(i) \tag{2.29}$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{2.30}$$

$$b_{jk}^* = \frac{\sum_{t=1}^{T}\gamma_t(j)\cdot\chi_{[O_t=v_k]}}{\sum_{t=1}^{T}\gamma_t(i)} \tag{2.31}$$

Eine detaillierte Beschreibung des Baum-Welch-Verfahrens ist z.B. in [ST95, Rab86] zu finden. Alternativ kann das Training auch mit dem Viterbi-Algorithmus (siehe Kap. 2.2.4) erfolgen.

Um den Zusammenhang mit den Adaptionsverfahren (siehe Kap. 8) zu verdeutlichen, werden im folgenden auch die Berechnungsvorschriften nach dem Baum-Welch Verfahren für kontinuierliche HMMs mit einer multivariaten Gaußdichte $g_j(\underline{x})$ dargestellt ($1 \leq t \leq T$):

$$\underline{\mu}_j^* = \frac{\sum_t \gamma_t(j)\underline{x}_t}{\sum_t \gamma_t(j)} \tag{2.32}$$

$$\Sigma_j^* = \frac{\sum_t \gamma_t(j)(\underline{x}_t - \underline{\mu}_j^*)(\underline{x}_t - \underline{\mu}_j^*)^T}{\sum_t \gamma_t(j)} \tag{2.33}$$

Eine entsprechende Erweiterung der Formeln auf Gaußsche Mischverteilungsdichten (einschließlich der Schätzung der Gewichte) ist in [ST95] zu finden.

2.2.4 Erkennung

Zur Erkennung bzw. Klassifikation mit HMMs wird die wahrscheinlichste Zustandsfolge q^* gesucht, mit der die Beobachtungsfolge O vom HMM λ erzeugt wurde. Hier wird der *Viterbi*-Algorithmus verwendet. Für die a posteriori Wahrscheinlichkeit gilt:

$$P(q|O,\lambda) = \frac{P(O,q|\lambda)}{P(O|\lambda)} \tag{2.34}$$

Die Viterbi-Wahrscheinlichkeit $P^*(O|\lambda)$ wird als Klassifikationsmaß gewertet und kann, da der Term $P(O|\lambda)$ nicht von der Zustandsfolge abhängt, folgendermaßen definiert werden:

$$P^*(O|\lambda) = \max_q P(O,q|\lambda) = P(O,q^*|\lambda) \tag{2.35}$$

Die gesuchten Wahrscheinlichkeiten werden – ähnlich der Bestimmung der Vorwärtswahrscheinlichkeiten – rekursiv bestimmt (vgl. [ST95]):

$$P^*(O|\lambda) = \max_j \vartheta_T(j) \quad \text{mit:} \quad \vartheta_t(j) = \max_q \{P(O_1,...,O_t,q_1,...,q_t|\lambda) \mid q_t = s_j\} \tag{2.36}$$

Neben der maximalen Likelihood, mit der die Beobachtungsfolge von einem bestimmten Modell erzeugt wurde, kann durch Rückverfolgung auch die wahrscheinlichste Zustandsfolge bestimmt werden.

Für die Schrifterkennung können die einzelnen Zeichen-HMMs zu Wörtern oder Sätzen verbunden werden (siehe Abb. 2.6). Die Zusammensetzung der Wortmodelle ergibt sich dabei aus der Schreibweise, die durch ein Lexikon vorgegeben ist, oder – wie in Kap. 2.3 erläutert – durch statistische Zeichenfolgewahrscheinlichkeiten. Die Modellierung von Sätzen kann mit Hilfe einer Grammatik erfolgen. In weiterführenden Arbeiten (siehe z.B. [Wan02a, Wan02b]) können auch verschiedene HMM-basierte Worterkenner kombiniert werden, um die Erkennungsrate zu steigern.

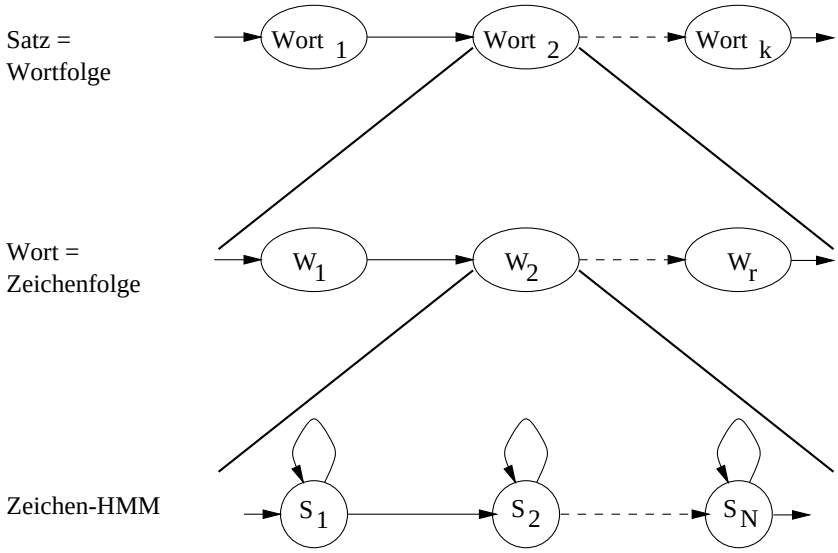


Abbildung 2.6: Modellierung von Worten und Sätzen

2.3 Sprachmodellierung

Neben der reinen merkmalsbasierten Klassifikation mit Hilfe von Hidden Markov Modellen liefert die Sprachmodellierung oder Grammatik einen weiteren wesentlichen Beitrag zur Erkennung, wie in Gl. 2.1 bereits dargelegt wurde.

In der Regel wird zur (Einzel-) Worterkennung ein bestimmtes Wörterbuch (WB) verwendet, in dem das zulässige Vokabular und die korrekte Schreibweise – also die Zusammensetzung der Worte aus den Buchstabenmodellen, die als HMM trainiert wurden – definiert wird (vgl. Abb. 2.7 links; Wörterbuch: ‘hallo, und, 30, ...’). In diesem Fall kann die Größe des Wörterbuches je nach Anwendung (Adreßerkennung, Terminplanung, allgemeiner Text, etc.) eingeschränkt werden, wodurch die Erkennungsleistung deutlich beeinflußt wird: je kleiner das Vokabular, desto höher die Erkennungsrate. Diese Aussage gilt jedoch nur dann, falls die zu erkennenden Worte auch im verwendeten Wörterbuch vorkommen. Gerade bei allgemeinen Texten unbekannten Themas (oder technisch/ wissenschaftlich/ medizinische Abhandlungen mit Spezialbegriffen, Nachrichten oder Namen) kann häufig kein vollständiges Wörterbuch zur Erkennung gestellt werden. Testworte, die nicht im WB vorkommen (OOV - out of vocabulary) , werden automatisch einem falschen Eintrag zugeordnet. Bei einer Erkennung ohne Vokabular werden wie in Abb. 2.7 rechts dargestellt, die Worte durch beliebige Buchstabensequenzen gebildet. Diese Erkennung ohne Kontextwissen führt – wie auch beim Menschen zu beobachten ist (vgl. [Sch98]) – zu einer sehr hohen Wortfehlerrate. Dieses Problem führt zur Anwendung statistischer Sprachmodelle, wie sie auch in der vorliegenden Arbeit verwendet werden (siehe z.B. [Bra00c]). Sprachmodelle beschreiben den Sprachgebrauch oder die Grammatik und sind unabhängig von den akustischen Signalen (Spracherkennung) oder dem Schriftbild (Schrifterkennung). Das Sprachmodell gibt die a

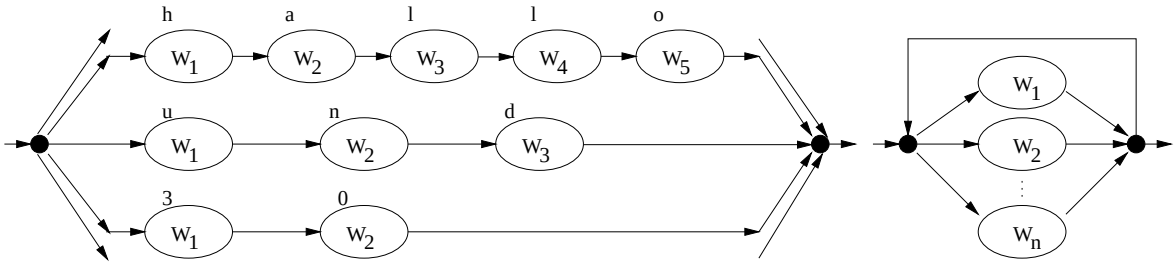


Abbildung 2.7: Modellierung von Worten mit (links) und ohne (rechts) Vokabular

priori Wahrscheinlichkeit $P(W) = P(w_1, w_2, .., w_N)$ dafür an, daß ein Satz (Wort) W aus den Wortfolgen (Zeichenfolgen) w_i besteht. Die übliche Anwendung bezieht sich allerdings auf die Satzerkennung von fließend gesprochener Sprache (z.B. [Coc98]), d.h. es werden Sprachmodelle benötigt, die Wortfolgen modellieren. In dieser Arbeit wird das Sprachmodell als Alternative zum Wörterbuch eingesetzt. Hier werden Buchstabenfolgen statistisch modelliert. Das bedeutet, aufbauend auf dem in Abb. 2.7 rechts dargestellten Verfahren werden zuvor geschätzte Wahrscheinlichkeiten für bestimmte Buchstabensequenzen in die Erkennung mit einbezogen. In der Regel werden sogenannte (Backoff-) N-Gramme als Sprachmodell verwendet, die abhängig von $N - 1$ vorherigen Buchstaben die Auftrittswahrscheinlichkeit des aktuellen Buchstabens schätzen. Der wesentliche Vorteil dieser Buchstaben N-Gramme, die in Kap. 5 ausführlich beschrieben werden, ist die Verbesserung der Erkennungsrate bei einer Erkennung ohne (bzw. mit unbegrenztem) Vokabular.

2.4 Kapitelzusammenfassung

In diesem Kapitel wurden die Grundlagen der automatischen Schrifterkennung anhand unterschiedlicher Eingabemedien und der einzelnen Stufen des Erkennungsprozesses, bestehend aus Vorverarbeitung, Merkmalextraktion und Klassifikation beschrieben.

Es wurden Charakteristika, die sich bei der Erkennung von Block- oder Fließschrift, von Druck- oder Handschrift und von on-line oder off-line Schriftdaten ergeben, dargestellt. Die Probleme, die sich bei den in dieser Arbeit behandelten Erkennungsaufgaben stellen, lassen sich durch ein entscheidendes Merkmal beschreiben: die einzelnen Buchstaben eines Wortes lassen sich nicht ohne weiteres separieren; sie sind untereinander verbunden. Darauf aufbauend wurden unterschiedliche Vorverarbeitungs- und Merkmalextraktionsmethoden im Prinzip vorgestellt und grundlegende Begriffe definiert. Als Klassifikator wurden Hidden Markov Modelle und Sprachmodelle vorgestellt. Hidden Markov Modelle setzen auf den extrahierten Merkmalvektoren an, hingegen wird mit den Sprachmodellen die Grammatik oder der linguistische Kontext berücksichtigt. Die Theorie der Hidden Markov Modelle wurde mit einem besonderen Schwerpunkt auf die Art der Emissionswahrscheinlichkeiten erläutert.

Kapitel 3

Datenbasen und Merkmalextraktionsverfahren

Die in dieser Arbeit entwickelten Verfahren zur Schrifterkennung wurden anhand unterschiedlicher Datenbasen untersucht. Allgemein kann man das Problemfeld der Schrifterkennung in verschiedene Kategorien unterteilen, wie z.B. die Erkennung von Einzelzeichen oder Worten, von Fließschrift oder Blockbuchstaben, von on-line oder off-line Daten, von Handschrift oder Druckschrift und die Unterscheidung zwischen einem und mehreren Schreibern bzw. Fonts. Abhängig vom Anwendungsbereich der Schrifterkennung wird die Art der Merkmalextraktion und des Klassifikators gewählt.

Den Schwerpunkt hier bildet die Handschrifterkennung von fließend geschriebenen Worten sowohl für den on-line als auch für den off-line Bereich. Eine weitere verwendete Datenbasis besteht aus gedruckten Dokumenten in geringer Qualität. Allen diesen Datenbasen ist gemein, daß eine separate Segmentierung der Worte in Einzelzeichen nur schwer möglich ist, woraus sich der entscheidende Vorteil für die Verwendung von Hidden Markov Modellen als Klassifikator ergibt. Die Verwendung unterschiedlicher Datenbasen zur Evaluierung der in dieser Arbeit entwickelten Verfahren zeigt zum einen das breite Anwendungsspektrum auf und ermöglicht zum anderen auch einen direkten Vergleich (beispielsweise on- und off-line Erkennung). Im folgenden werden die verschiedenen verwendeten Datenbasen mit den zugehörigen Merkmalextraktionsverfahren näher beschrieben.

3.1 On-Line Handschrifterkennung

Das Erkennungssystem für handgeschriebene on-line Daten wurde mit einer am Fachgebiet Technische Informatik erstellten Datenbasis entwickelt. Diese on-line Datenbasis besteht aus handschriftlichen Wörtern oder Sätzen, die von verschiedenen Personen in Fließschrift (kursiv) mit einem Digitalisiertablett (WACOM-Schreibtablett) aufgezeichnet wurden. Für die

on-line Datenbasis werden die Sequenzen der zeitlich geordneten Koordinaten des Schriftzuges gespeichert. Die Aufnahme der Daten erfolgt mit einer konstanten Abtastperiode von 5 ms. Dies hat jedoch zur Folge, daß ein völlig identisches Zeichen bei unterschiedlichen Schreibgeschwindigkeiten zu einer anderen Datensequenz führt. Die Normierung der Schreibgeschwindigkeit erfolgt durch eine räumliche Abtastung mit einer konstanten Vektorlänge (siehe [Rig96, Kos97a, Kos00a]). Ein weiteres Problem, welches nur in der on-line Erkennung auftaucht, ist die Bearbeitung zeitlich verzögerter Striche, wie z.B. t-Striche und i-Punkte. Diese werden je nach Schreiber direkt nach Vollendung des Buchstabens eingefügt oder auch erst am Ende des Wortes. Diese kurzen Striche werden, wenn sie erst am Ende des Wortes geschrieben werden, durch ein zusätzliches Modell (‘Rücksprung’) definiert, wobei jedoch der Bezug zum zugehörigen Buchstaben verloren geht. Im schreiberabhängigen Modus ist dieses Problem nicht so markant, da ein Schreiber für gewöhnlich die gleiche Reihenfolge beibehält. Die Vorverarbeitungsmethoden wurden bereits in [Kos00a] vorgestellt und werden hier im wesentlichen unverändert übernommen.

In dieser Arbeit wird zwischen einem schreiberabhängigen und einem schreiberunabhängigen System unterschieden, die sich in erster Linie durch die Menge der Beispieldaten je Schreiber und die Art der Vorverarbeitung der Daten unterscheiden. Die Aufnahmemethode der Daten ist in beiden Fällen identisch. Die Experimente zur schreiberabhängigen on- und off-line Erkennung (siehe Kap. 9.2.1 und 9.3.1) beziehen sich auf den gleichen Datensatz, sodaß hier ein direkter Vergleich stattfinden kann.

3.1.1 Schreiberabhängiges System

Im schreiberabhängigen Modus wird für jeden Schreiber ein eigenes Erkennungssystem trainiert, woraus folgt, daß die Anzahl der Trainingsworte je Schreiber entsprechend hoch sein muß im Vergleich zum schreiberunabhängigen System (vgl. auch [Bra99b, Kos00a]).

Datenbasis

Die schreiberabhängige Datenbasis besteht aus vier Personen (ABR, ANK, JMR, VDM), die jeweils einen Trainingsdatensatz von mehreren Sätzen (etwa 2000 Wörter in Groß- und Kleinschreibung) und einen Testdatensatz von je fast 200 Einzelwörtern geschrieben haben. Der zu schreibende deutsche Text ist für jede Person identisch. Beispiele der Test-Datenbasis sind in Abb. 3.1 dargestellt. Bis auf den Hinweis, ‘normal und waagerecht’ zu schreiben, wurden keine Einschränkungen vorgegeben. Der Datensatz enthält etwa 80 verschiedene Zeichen: Buchstaben, Zahlen und Sonderzeichen wie z.B. Klammern oder Satzendezeichen. Zusätzlich haben alle vier Schreiber die zugelassenen Zeichen auch mehrfach als Einzelzeichen (vgl. Abb. B.1) geschrieben. Diese Daten werden zur Initialisierung der HMMs benutzt.

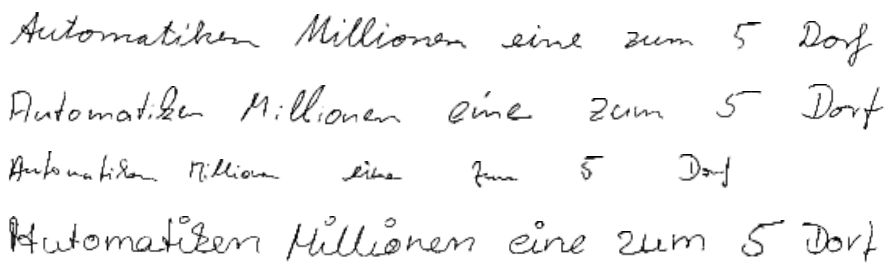


Abbildung 3.1: Beispiele von 4 Schreibern (ABR, ANK, JMR, VDM) der schreiberabhängigen on-line Datenbasis

Die gleichen Daten (Einzelzeichen und Worte) werden auch für das schreiberabhängige off-line Erkennungssystem verwendet (siehe Kap. 3.2.1), indem die on-line Daten in ein Binärbild überführt werden.

Merkmalextraktion

Bis auf die Abtastung des Schriftzuges um unterschiedliche Schreibgeschwindigkeiten zu kompensieren, wird keine weitere Vorverarbeitung der Daten durchgeführt (vgl. auch [Ben95]). Erst bei einer Erweiterung des Systems auf schreiberunabhängige Schrifterkennung würde eine aufwendigere Vorverarbeitung erforderlich sein (vgl. Kap. 3.1.2). Hier werden, wie in Abb. 3.2 dargestellt, die folgenden Merkmale aus dem Schriftzug ermittelt (siehe auch [Rig98c, Rig98b, Kos00a]):

- der Winkel der abgetasteten Striche (Segmente) des Schriftzuges ($\sin\alpha$, $\cos\alpha$)
- die Winkeldifferenz aufeinanderfolgender Striche ($\sin\Delta\alpha$, $\cos\Delta\alpha$)
- der Stiftdruck (binär)
- eine unterabgetastete Bitmap (9-dimensionaler Vektor); diese enthält die aktuelle Bildinformation in einem 30×30 Fenster, das entlang des Schriftzuges geschoben wird

Dieser 14-dimensionale Merkmalvektor $\underline{x}$ (5 dynamische Merkmale und 9 Elemente der Bitmap) bildet die Eingabeinformation der Hidden Markov Modelle.

Abbildung 3.2: Handschrift-Merkmalextraktion (on-line)

Die zugehörigen Versuchsreihen und Ergebnisse werden in Kap. 9.2.1 vorgestellt.

3.1.2 Schreiberunabhängiges System

Um eine schreiberunabhängige on-line Handschrifterkennung durchführen zu können, wird eine Datenbasis von insgesamt 166 Schreibern zur Verfügung gestellt, die im Vergleich zum schreiberabhängigen Modus jedoch jeweils deutlich weniger Trainingsbeispiele schreiben (vgl. [Kos00a, Bra01a, Bra02a]). Mit Hilfe dieser Datenbasis, in der auch die vier Schreiber aus Kap. 3.1.1 vorkommen, wird auch der Einfluß von Adaptionsverfahren (siehe Kap. 8) untersucht, sodaß ein direkter Vergleich zwischen dem schreiberab- und schreiberunabhängigen Erkennungsmodus erfolgen kann.

Datenbasis

Die Datenbasis besteht aus Schriftproben (Fließschrift) von insgesamt 166 verschiedenen Schreibern, die jeweils ca. 200 Worte auf ein Digitalisiertablett geschrieben haben. Beispiele der Datenbasis sind in Abb. 3.3 dargestellt. Die Trainingsdatenbasis besteht aus etwa 24400 Worten von 145 Schreibern, für den Testdatensatz werden 4153 Worte von 21 Schreibern verwendet. Keiner der Testschreiber kommt in der Trainingsmenge vor.

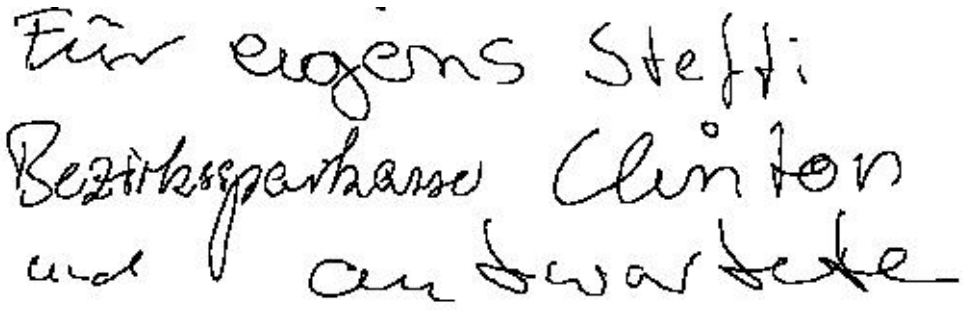


Abbildung 3.3: Beispiele der schreiberunabhängigen on-line Datenbasis (hier: 7 Schreiber)

Um Adaptionsverfahren testen zu können, wird dieser Testdatensatz der 21 Schreiber gegebenenfalls in eine Adaptions- und eine Testmenge unterteilt, die somit jeweils etwa 100 Worte von 21 Personen enthalten.

Merkmalextraktion

Nach der Abtastung des Schriftzuges (Abtastpunkte $f(x, y)$) zur Kompensation unterschiedlicher Schreibgeschwindigkeiten, werden die Schriftproben normiert (vgl. auch Kap. 3.1.1). Die Normierung bezieht sich auf die Zeichenschräglage und die Größe. Beide Schrifteigenschaften variieren zwischen einzelnen Schreibern stark und müssen für ein schreiberunabhängiges Erkennungssystem angeglichen werden.

Die Normierung der Zeichenschräglage (slant) erfolgt über eine Scherung des Schriftbildes gemäß einem Entropie-Kriterium (vgl. Gl. 3.1), sodaß der Anteil der vertikalen Striche

überwiegt (siehe auch [Bra02a]). Die Neigung des Schriftzuges wird durch eine Scherung (siehe Anhang A) zuerst in verschiedenen Richtungen und unterschiedlichem Neigungswinkel Φ verändert. Die Projektion des Schriftzuges auf die x-Achse $K(x)$ und die daraus ermittelte Entropie H gibt Aufschluß über den tatsächlichen Neigungswinkel Φ^* (minimale Entropie, vgl. [Bra02a]).

$$H(\Phi) = -\sum_x K_\Phi(x) \cdot \log K_\Phi(x) \quad \rightarrow \quad \Phi^* = \underset{\Phi}{\text{argmin}}\, H(\Phi) \tag{3.1}$$

$$\text{mit:} \quad K(x) = \frac{n_x}{\sum_x n_x} \quad \text{und} \quad n_x = \sum_y f(x,y) \tag{3.2}$$

Die Größennormierung richtet sich nur nach der Kernhöhe, also der Höhe der kleinen Buchstaben wie ‘m, o, a’, und ist unabhängig von der Gesamtschrifthöhe. Die Basislinien zur Bestimmung der Kernhöhe werden durch horizontale Geraden approximiert, die durch die Häufung von lokalen Minima (Grundlinie) bzw. Maxima (Kernhöhe) bestimmt werden. Die anschließend extrahierten Merkmale sind identisch mit denen des schreiberabhängigen Systems (vgl. Abb. 3.2):

- der Winkel der abgetasteten Striche (Segmente) des Schriftzuges ($\sin\alpha$, $\cos\alpha$)
- die Winkeldifferenz aufeinanderfolgender Striche ($\sin\Delta\alpha$, $\cos\Delta\alpha$)
- der Stiftdruck (binär)
- eine unterabgetastete Bitmap (9-dimensionaler Vektor); diese enthält die aktuelle Bildinformation in einem 30×30 Fenster, das entlang des Schriftzuges geschoben wird

Durch eine ungenaue Schätzung der Basislinien (krumme Schreibweise, zum Wortende kleiner) können die Merkmale auch verfälscht werden. Kap. 9.2.2 beschreibt die zugehörigen Versuche und Ergebnisse. Dort wird ebenfalls der Einfluß der Normierung untersucht, indem zwei schreiberunabhängige Systeme (mit und ohne Normierung der Schrift) verglichen werden.

3.2 Off-Line Handschrifterkennung

Für die off-line Erkennung stehen zwei prinzipiell verschiedene Datenbasen zur Verfügung. Dies ist zum einen eine am Fachgebiet erstellte schreiberabhängige Datenbasis (konvertierte on-line Daten) und zum anderen eine Adreßdatenbasis, die innerhalb eines BMBF-Projektes (Adaptive READ) von Siemens Dematic (SD) zur Verfügung gestellt wurde.

3.2.1 Konvertierte On-Line Daten (schreiberabhängig)

Ein dem on-line System entsprechendes schreiberabhängiges System wird für jeden der vier Schreiber (ABR, ANK, JMR, VDM) erstellt (vgl. Kap. 3.1.1).

Datenbasis

Als off-line Datenbasis wird hier die konvertierte on-line Datenbasis bestehend aus je 2000 Trainings- und 200 Testworten je Schreiber (siehe Abb. 3.1) verwendet. Die Koordinaten des Schriftzuges werden in ein Binärbild überführt, die dynamischen Merkmale der Dateninformation werden nicht benutzt. Somit kann für den Vergleich von on- und off-line Handschrifterkennung die gleiche Qualität der Schriftdaten garantiert werden. Störungen des off-line Datensatzes, die z.B. durch das Einscannen entstehen können, werden so vermieden. Aufgrund dieser Datengewinnung ist auch die Vorverarbeitung der Schriftbilder sehr einfach, da eine Bildaufbereitung (z.B. Entfernung von Rauschen) und Skelettierung hinfällig wird (siehe z.B. [Bra99b, Bra00a]).

Merkmalextraktion

Die Normierung der off-line Daten beinhaltet nur die Korrektur der Zeilenschräglage, wohingegen eine Scherung (Aufrichtung der Buchstaben) oder eine Größennormierung in einem schreiberabhängigen System nicht unbedingt notwendig sind. Im off-line Handschrifterkennungssystem werden im Rahmen dieser Arbeit dann die folgenden Merkmale von den normierten Bild-Daten extrahiert (vgl. Abb. 3.4):

- eine unterabgetastete Bitmap (9-dimensionaler Vektor), die in horizontaler Richtung über das Textbild geschoben wird (sliding window)
- einige zusätzliche Merkmale wie den Abstand zur Grundlinie, die Dicke des Wortes an der aktuellen Position und die Anzahl der schwarz-weiß-Übergänge

Abbildung 3.4: Handschrift-Merkmalextraktion (off-line)

In horizontaler Richtung wird ein schmales Fenster (konstante Breite von zwei Pixeln) über das binäre Textbild geschoben, wobei die jeweilige Umgebung von ca. 20 Pixeln die

aktuelle Höhe und y-Position bestimmt. Dieses Fenster wird in neun gleich große horizontale Abschnitte unterteilt, die jeweils gemittelt den ersten Merkmalvektor (9-dimensional) für das System bilden. Als Alternative zur Unterabtastung des Fensters wurden auch Experimente mit den DCT-Koeffizienten dieses Fensters (Diskrete Cosinus-Transformation) durchgeführt, welche gerade bei kontinuierlichen HMMs zu höheren Erkennungsraten führen, wie in Kap. 9.3.1 gezeigt wird.

Eine Eigenschaft dieser Bitmap ist die Unabhängigkeit von der Größe (näherungsweise) und der Grundlinie der Schrift. Im Gegensatz dazu basieren die drei weiteren Merkmale teilweise sowohl auf der Lage der Grundlinie als auch auf der Texthöhe. Anhand der Anzahl der schwarz-weiß-Übergänge (Wechsel Schrift-Hintergrund) kann man beispielsweise auf Schlaufen schließen. Auf die Bestimmung der Kernhöhe wurde hier verzichtet, da, wie in Abb. 3.4 zu sehen ist, bereits die Grundlinie relativ ungenau als Gerade approximiert werden kann. Die Schätzung der Grundlinie eines Wortes (oder Satzes) beruht auf der Auswertung der lokalen Minima des Schriftzuges (Faltung mit entsprechenden Masken). Diese werden rekursiv in einem schmaler werdenden Bereich ausgewertet und durch eine horizontale Linie approximiert. Auf eine exaktere Bestimmung der Grundlinie wie z.B. in [Ben95] oder [Sei96], wobei die Grundlinie quasi jedem Buchstaben eines Wortes angepaßt wird (evtl. kurvenförmige Linie), wurde hier verzichtet, da dieser zweite Merkmalvektor in erster Linie der Unterscheidung von Groß- und Kleinbuchstaben bzw. Ober- und Unterlängen dient und die einfache Grundlinienbestimmung für diesen Zweck hinreichend genau ist. Im Lauf der Arbeit zeichnete sich ab, daß entweder die Basislinie genau und robust geschätzt werden muß, oder aber die Merkmale relativ unabhängig von dieser sein müssen. Hier fiel die Entscheidung auf die zweite Methode. Eine andere Möglichkeit, Fehler durch die ungenaue Basislinienbestimmung auszugleichen, wird in [Wan00, Wan01] verfolgt. Hier wird die geschätzte Grundlinie absichtlich parallel nach oben und unten verschoben, sodaß sich drei verschiedene Modelle aufgrund dreier Basislinien ergeben.

Die so ermittelten 12-dimensionalen Merkmalvektoren $\underline{x}$ werden zum Training genutzt. Der Vergleich der unterschiedlichen Merkmalextraktionen und weitere Versuche zu Modellierungstechniken sind in Kap. 9.3.1 dargestellt.

3.2.2 Handgeschriebene Adressen (schreiberunabhängig)

Diese zweite off-line Datenbasis enthält handgeschriebene Adressen in Fließschrift oder auch in Blockbuchstaben (siehe [Bra01d, Bra02c]). Ziel bei der – verständlicherweise schreiberunabhängigen – Adreßerkennung ist die automatisierte Postzustellung. Die Adressen stammen von verschiedenen Postverteilzentren in Deutschland und den USA und wurden von Siemens Dematic (SD) bereitgestellt. Briefe enthalten neben der Anschrift und dem Absender noch weitere ‘Störfaktoren’ wie Stempel und Briefmarken. Die Segmentierung und Lokalisation der Adresse ist nicht Thema dieser Arbeit und wurde ebenfalls von SD

ausgeführt. Auch die Segmentierung der Anschrift in Name, Straße und Ort wird hier vorausgesetzt, da die vollständigen Adressen (bis auf Einzelbeispiele, siehe Abb. 3.6) aus Datenschutzgründen nicht gespeichert bzw. verwertet werden dürfen. Der Name spielt bei der automatischen Postverteilung in der Regel (außer z.B. Nachsende-Aufträge) keine Rolle, sodaß sich die Erkennungsaufgabe und somit die Datenbasis auf Straßen (oder auch Postfächer) und Städte beschränkt. Auch die Erkennung der Postleitzahl (PLZ) ist eher ein Einzelzeichenerkennungsproblem und wird hier nicht extra berücksichtigt. In der Praxis wird anhand der erkannten PLZ das Wörterbuch zur Erkennung des Städte-Namens eingeschränkt.

Im Rahmen dieser Arbeit wurden für die Adreßerkennung zwei getrennte Systeme entwickelt, für den deutschen und den amerikanischen Datensatz. Die Merkmalextraktion ist in beiden Fällen identisch.

Deutsche Adreß-Datenbasis

Beispiele der deutschen Adreß-Datenbasis sind in Abb. 3.5 dargestellt. Dieser Datensatz beinhaltet zwei verschiedene Typen:

- der *Basisdatensatz* besteht aus ca. 22000 Worten, von denen fast 20000 zum Trainieren und 2027 (935 Städte und 1092 Straßen) zum Testen verwendet werden. Die Aufteilung in Test- und Trainingsdatenbasis erfolgt zufällig. Diese Daten stammen aus verschiedenen Postämtern in Deutschland (Dresden, Essen, Offenburg, Hamm, Frankfurt, Hamburg und andere) und enthalten weder Postleitzahl noch Hausnummer. Lediglich für die Adaptionsversuche gibt es zum Training der Ziffern-HMM einen separaten Trainingsdatensatz bestehend aus 5200 Postleitzahlen.

- vier spezielle Datensätze enthalten nur Adressen (getrennt nach Stadt und Straße) von einem bestimmten Postamt und werden für Adaptionsversuche genutzt. Diese Beispiele enthalten in der Regel auch Postleitzahlen (ggf. mit Zusatz ‘D-’) und Hausnummern (ein- oder mehrstellige Zahlen und Buchstaben). Der Grund für diese unterschiedliche Datenstruktur ist jedoch nur der Zeit/Kosten-Faktor zur Erstellung der Daten bei SD.
  Der *HRO-Datensatz* (Postamt Rostock) besteht aus etwa 1700 Trainingsworten und 1550 Testworten.
  Der *STR-Datensatz* (Postamt Stuttgart) besteht aus etwa 1500 Trainingsworten und 1500 Testworten.
  Der *HAL-Datensatz* (Postamt Halle/Saale) besteht aus etwa 1580 Trainingsworten und 1460 Testworten.
  Der *HAM-Datensatz* (Postamt Hamburg) besteht aus etwa 1510 Trainingsworten und 1500 Testworten.

Etwa 77 verschiedene Einzelzeichen (Buchstaben, Zahlen und Sonderzeichen wie z.B. ‘. - /’) kommen in den Daten vor.

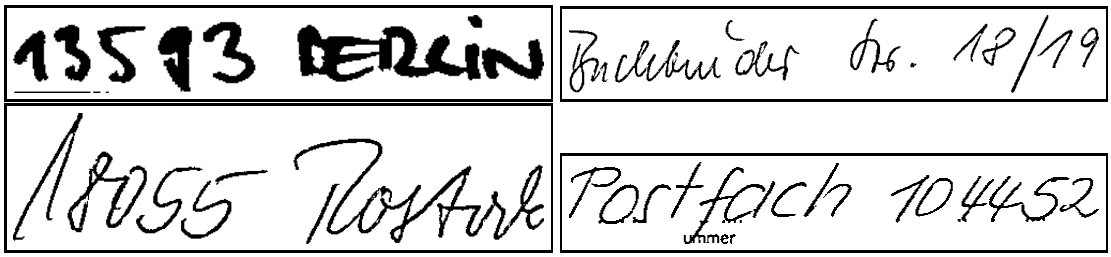


Abbildung 3.5: Beispiele der deutschen Adreß-Datenbasis (SD)

Amerikanische Adreß-Datenbasis

Dieser Datensatz besteht aus etwa 16600 Einzelworten. Einzelworte sind in diesem Fall Städte- oder Straßen-Namen, auch wenn diese aus zwei oder mehreren Teilausdrücken (z.B. New York) bestehen. Postleitzahlen oder Hausnummern wurden zuvor bereits abgetrennt und kommen nicht vor. Für das Training werden 15100 Beispiele (Städte und Straßen) verwendet und der Testdatensatz besteht aus 958 Städten und 551 Straßen. Ein Beispiel einer vollständigen Adresse zeigt Abb. 3.6.

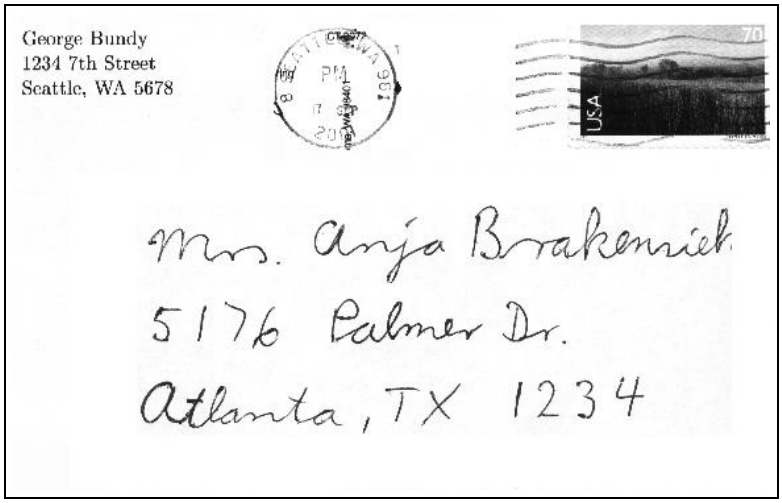


Abbildung 3.6: Beispiel einer amerikanischen (virtuellen) Adresse

Die Daten enthalten etwa 70 verschiedene Einzelzeichen (Buchstaben, Zahlen, Sonderzeichen wie z.B. ‘. - &’), denn anders als in Deutschland, bestehen viele amerikanische Straßen-Namen auch aus Zahlen (z.B. ‘83rd St’).

Merkmalextraktion

Die Vorverarbeitung und Merkmalextraktion der Adreß-Daten wurde von Siemens Dematic durchgeführt (siehe z.B. [Cae93, Fra97]) und nur die extrahierten Merkmalvektoren wurden für die Untersuchungen hier zur Verfügung gestellt. Trotzdem sollen die Merkmale im weiteren kurz erläutert werden.

Nach der Aufnahme (Scannen) des Briefes wird das Bild binarisiert und die Anschrift lokalisiert und in Zeilen bzw. Worte segmentiert. Fehler oder Störungen, die durch das Scannen entstehen, werden mit Hilfe eines Zebra-Filters eliminiert. So werden auch Lücken im Schriftzug geschlossen. In einem weiteren Schritt werden verbundene Strukturen, die *Binary Connected Components (BCC)* (siehe [Man90]) im Bild ermittelt, auf denen die weiteren Normierungs- und Merkmalextraktionsverfahren aufsetzen. Ein BCC-Objekt ist eine hierarchische Struktur und besteht aus einem Konturpolygon (der Farben schwarz oder weiß), einem umgebenden Rechteck und dem Bezug zu den inneren Bereichen.

Zur Größen- und Rotationsnormierung (Korrektur der Zeilenschräglage) ist die Lage der Grundlinie und die Kernhöhenbestimmung notwendig. Die Bestimmung der Basislinien erfolgt durch die Ermittlung lokaler Minima bzw. Maxima des Wortes. Diese Basislinien werden als Geraden approximiert (lineare Regression), die nicht notwendigerweise parallel verlaufen müssen. Dabei spielt auch die Art der Extrema (flach, spitz) eine Rolle. Ein Beispiel für die Lage der unteren und oberen Basislinie ist in Abb. 3.7 dargestellt.

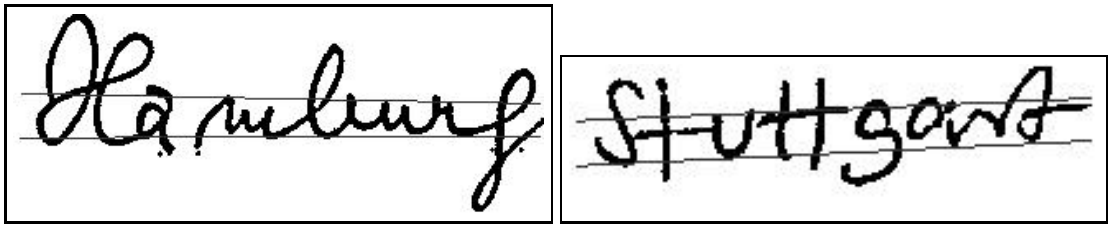


Abbildung 3.7: Bestimmung der Basislinien

Die Größennormierung erfolgt implizit durch die Auswahl der Merkmale, die sich an den Basislinien orientiert. Die Normierung der Zeilenschräglage wird durch Rotation anhand der Lage der Grundlinie erreicht. Die Zeichenschräglage wird durch eine Scherung des Schriftzuges korrigiert, wobei der Scherungswinkel innerhalb eines Wortes konstant ist.

Für die Merkmalextraktion wird ein Fenster von links nach rechts über das normierte BCC-Bild geschoben, wobei die Fenster einander überlappen. Die Fensterbreite wird so gewählt, daß ein durchschnittlicher Buchstabe mit fünf überlappenden Fenstern abgedeckt werden kann. Jedes Fenster wird anhand der beiden Basislinien in fünf Abschnitte eingeteilt, die sich wiederum überlappen. Innerhalb dieser Abschnitte werden die folgenden Merkmale ermittelt und mit Werten zwischen Null und Eins kodiert:

- unter der Grundlinie: Querstriche
- im Bereich der Grundlinie: Scheitelpunkte und Spitzen
- zwischen unterer und oberer Basislinie: Kurven, Anstiege und horizontale Linien
- im Bereich der Kernhöhe: vertikale und horizontale Striche

- über der oberen Basislinie: Punkte, Anstiege und horizontale Linien

So wird ein 20-dimensionaler Merkmalvektor $\underline{x}$ gebildet. Wahlweise werden mehrere benachbarte 20-dimensionale Merkmalvektoren anschließend LDA-transformiert (Lineare Diskriminanzanalyse), sodaß nach einer Dimensionsreduzierung in der Regel 30-elementige Merkmalvektoren entstehen. Die so gebildeten Vektoren haben den Vorteil, daß sie auch Informationen über die aktuelle Nachbarschaft (also vorausgehende und nachfolgende Merkmale) beinhalten.

Die Beschreibung der Experimente und Ergebnisse zur deutschen und amerikanischen Adreß-Datenbasis ist in Kap. 9.3.2 zu finden.

3.3 Dokumenterkennung

Für die Erkennung gedruckter Schrift wird die SEDAL-Datenbasis[1] verwendet. Diese Datenbasis ist öffentlich zugänglich, wodurch ein Vergleich der mit den eigenen Verfahren erzielten Ergebnisse mit denen anderer Institute ermöglicht wird (vgl. [Sch98, Bra01c]).

Datenbasis

Die SEDAL-Datenbasis besteht aus eingescannten Dokumenten in geringer Auflösung (200 dpi) und schlechter Qualität, wie sie beispielsweise durch häufiges Kopieren oder Faxen entsteht. Auf Grund dessen sind die einzelnen Schriftzeichen häufig verschmiert und somit mit den Nachbarzeichen verbunden. Das andere Extrem bilden sehr dünne Zeichen, die in mehrere Teile aufgespalten sind. Beides erschwert eine Zeichensegmentierung, wie sie für die konventionelle Schrifterkennung (OCR) notwendig wäre. Neben diesen echten Dokumenten enthält die SEDAL-Datenbasis auch künstlich erzeugte Texte in verschiedenen Fonts (Schriftgröße jeweils 10 pt), deren Erscheinungsbild nachträglich durch ungünstige Einstellungen (zu hell oder zu dunkel) beim Kopieren oder Faxen verschlechtert wurde. Beispiele dieser Datenbasis sind auch im Anhang D zu finden und verdeutlichen auch für den menschlichen Betrachter die Erkennungsproblematik, da es sich hier um willkürliche Zeichenketten ohne Sinn (keine bekannten Worte) handelt. Die Dokumente enthalten englischsprachigen Text in verschieden Fonts und Fontgrößen zu unterschiedlichen Themengebieten (Briefe, Artikel). Zu jeder binären Bilddatei sind außerdem die Positionen der Einzelworte im Bild angegeben, so daß eine Segmentierung der Dokumente in Abschnitte (Briefkopf, Überschrift, Symbole) bzw. Zeilen und Spalten hier entfallen kann. Die im Rahmen dieser Arbeit präsentierten Ergebnisse beziehen sich auf die Worterkennung und schließen Fehler bei der Dokumentsegmentierung aus.

[1] Systems Engineering and Design Automation Laboratory at the University of Sydney, Australia, http://www.sedal.usyd.edu.au

Für das Training des Schrifterkennungssystems werden ca. 63000 Einzelzeichen (80 verschiedene Klassen: Groß- und Kleinbuchstaben, Zahlen und einige Sonderzeichen) aus verschiedenen Dokumenten der SEDAL-Datenbasis verwendet. Die Testdaten bestehen aus sechs nicht in der Trainingsmenge enthaltenen echten Dokumenten (Ausschnitte siehe Abb. 3.8) und enthalten ca. 12600 Einzelzeichen bzw. 2200 Worte.

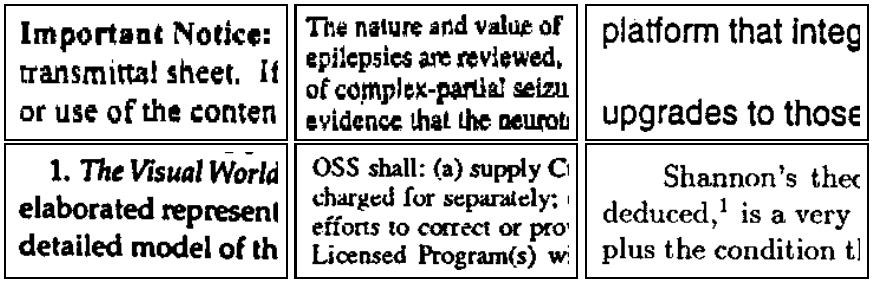


Abbildung 3.8: Ausschnitte aus der SEDAL-Testdatenbasis (oben: intel, neurofax, precept; unten: vision, maintenance, james)

Merkmalextraktion

Bei maschinengedruckter Schrift vereinfacht sich die Vorverarbeitung im Vergleich zur Handschrifterkennung (vgl. Kap. 3.2). Hier besteht die Vorverarbeitung und Merkmalextraktion in einer Zeilenlagekorrektur, der Bestimmung der Grundlinie und der Kernhöhe und einer impliziten Größennormierung aufgrund der Merkmale, die im wesentlichen aus DCT-Koeffizienten eines gleitenden Fensters bestehen.

Obwohl die Dokumente (und deren eingescannte Binärbilder $b(x, y)$) prinzipiell horizontale Zeilen enthalten, wird hier für die Erkennung jedes Wort bzgl. seiner Zeilenneigung korrigiert. Diese geringfügige Korrektur ist notwendig, da durch Kopieren und Faxen häufig Schlangenlinien im Dokument entstehen (vgl. Abb. 3.9, die Hilfslinien sind nur zur Verdeutlichung eingezeichnet). Diese Korrektur der Zeilenneigung ('skew') wird durch eine Rotation anhand eines Entropie-Maßes bestimmt (vgl. auch Kap. 3.1.2).

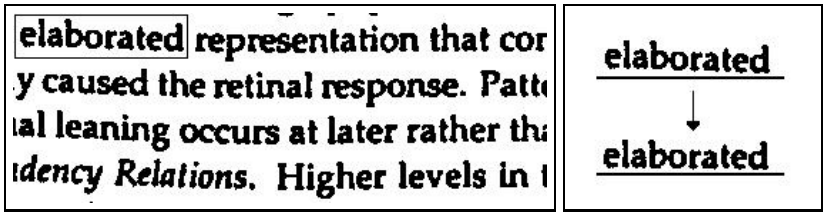


Abbildung 3.9: Beispiel mit Schlangenlinien (links: Ausschnitt aus 'vision') vor (rechts oben) und nach (rechts unten) der Korrektur der Zeilenneigung (Drehung um 1.5°)

Das Bild wird in einem bestimmten Bereich um jeweils 0.5° gedreht. Die Entropie, basierend

auf dem Histogramm (horizontale Projektion der Schriftpunkte) jedes gedrehten Wort-Bildes wird bestimmt und der endgültige Drehwinkel ergibt sich aus den Werten der minimalen Entropie. Diese Korrektur wird bei sehr kurzen Worten (z.B. ‘a’) nicht durchgeführt.

In einem zweiten Schritt werden die Basislinien (Grundlinie und Kernhöhe) bestimmt. Dazu wird durch horizontale Projektion des Wortes ein Histogramm $K(y)$ gemäß Gl. 3.3 gebildet, welches zum einen von der Anzahl der schwarzen Bildpunkte (Schrift, $K_p(y)$) je Zeile und zum anderen auch von der Anzahl der schwarz-weiß Übergänge ($bz_{wechsel}(y)$) abhängt (vgl. Abb. 3.10). Wertet man nur die Anzahl der Schriftpunkte pro Zeile aus, ergeben sich Probleme bei horizontalen Strichen (z.B. beim ‘T’ oder ‘e’), die durch die Kombination mit der Anzahl der Schrift-Hintergrund Wechsel allerdings wieder ausgeglichen werden können. Dieses Merkmal wiederum ist anfälliger gegenüber Störpunkten. Gerade für kurze Worte hat sich in Experimenten die Basislinienbestimmung anhand $K(y)$ als geeigneter erwiesen.

$$K(y) = bz_{wechsel}(y) \cdot K_p(y) \qquad \text{mit:} \quad K_p(y) = \sum_x b(x,y) \tag{3.3}$$

Dieses Histogramm (vgl. Abb. 3.10) weist in der Regel zwei deutliche sprunghafte Anstiege auf, anhand derer die untere und obere Basislinie bestimmt werden können (vgl. auch Abb. 3.11). Dies kann vorausgesetzt werden, da die Häufigkeit der Zeichen mit Ober- oder Unterlänge im Wort in der Regel eher gering ist. Ausgehend von der unteren und der oberen Begrenzung wird nach einer deutlichen Differenz $K(y') - K(y'')$ im Histogramm gesucht. Dabei wird allerdings auch das Wissen über ungefähre Proportionen der Schrift (z.B. Verhältnis von Kernhöhe zur Gesamthöhe) berücksichtigt. Je länger das Wort, umso deutlicher wird das Histogramm. Diese Basislinien können ohne großen Näherungsfehler als horizontale Geraden angesehen werden (im Gegensatz zur Handschrift).

Abbildung 3.10: Histogramme zweier Beispiele (mitte): links $K(y)$, rechts $K_p(y)$

Die korrekte Bestimmung dieser Basislinien ist wesentlich für die Erkennungsaufgabe, da die folgende Merkmalextraktion darauf beruht. Probleme und Fehler bei der Bestimmung der Kernhöhe treten immer dann auf, wenn ein Wort nur aus Zeichen mit oder ohne Oberlänge besteht (z.B. ‘come’, ‘OSS’). Hier kann die Entscheidung über die Wahl der Kernhöhe nur im Zeilen- oder Dokumentzusammenhang fallen. Dies setzt jedoch eine einheitliche Fontgröße innerhalb des Vergleichsmusters voraus, was bei Überschriften oder Anschriften nicht der Fall ist.

Abbildung 3.11: Merkmalextraktion für gedruckte Schrift

Für die Merkmalextraktion wird nun ein schmales Fenster (sliding window-Technik) in horizontaler Richtung über das Wort geschoben. Dieses Fenster wird zentriert auf der Kernhöhe positioniert und ist doppelt so hoch wie diese. Der Fensterinhalt bzw. die extrahierten Merkmale sind deshalb unabhängig von Ober- und Unterlängen. Die folgenden Merkmale werden je Fenster ermittelt und bilden den Merkmalvektor $\underline{x}$:

- 10 DCT-Koeffizienten (Diskrete Cosinus Transformation) des Fensterinhaltes; dabei wird das Fenster zuvor in einen 1-dimensionalen senkrechten Mittelwertvektor überführt
- 3 zusätzliche Merkmale je Fenster, die auf die Worthöhe normiert werden: Anzahl der gesetzten Schriftpunkte, lokale Höhe über der Grundlinie und die Größe der Unterlänge

Diese 13-dimensionalen Merkmalvektoren $\underline{x}$ werden zum Training und Testen des Dokument-Schrifterkennungssystems verwendet. Versuche und Ergebnisse mit dieser Datenbasis sind im Kap. 9.4 beschrieben.

Ergebnisse kommerzieller OCR-Systeme

Die Autoren von [Sch98], die die SEDAL-Datenbasis erstellt und veröffentlicht haben, geben in ihrem Artikel auch Ergebnisse kommerzieller OCR-Software (Stand 1997) auf dieser Datenbasis an. Diese Ergebnisse werden im folgenden (Tab. 3.1) zusammenfassend dargestellt, um die Verwendung dieser Datenbasis für den Einsatz von HMMs und Sprachmodellen zu motivieren. Die Testbeispiele sind einige ‘real-world’-Dokumente und ähnlich den in dieser Arbeit verwendeten Testdokumenten.

Tabelle 3.1: Ergebnisse (Zeichen-Fehlerrate in %) kommerzieller Systeme auf der SEDAL-Datenbasis (6 Dokumente), wie in [Sch98] präsentiert

OCR	System 1		System 2	
	ohne WB	mit WB	ohne WB	mit WB
Durchschnitt	24.5	24.2	42.8	33.7

Es soll an dieser Stelle ausdrücklich darauf hingewiesen werden, daß ein direkter Vergleich mit den hier erzielten Ergebnissen nur größenordnungsmäßig erfolgen kann, da zum einen die Dokumente nicht 100-prozentig übereinstimmen und sich zum anderen die Leistung kommerzieller OCR-Systeme seit 1997 verbessert hat. Auch über die Hersteller der OCR-Software (Produktname) wurde in [Sch98] nicht berichtet. Anhand der Ergebnisse wird jedoch deutlich, daß auch für gedruckte Dokumente (in schlechter Qualität) Handlungsbedarf besteht. Die gleichen Produkte liefern auf 'guten' Dokumenten fast keinen Fehler [Sch98]. Ein anderer Aspekt, der hier deutlich wird, ist der Einfluß des Wörterbuches (WB).

3.4 Kapitelzusammenfassung

Zur Evaluierung und zum Vergleich der in dieser Arbeit untersuchten Modellierungstechniken und Adaptionsverfahren werden die folgenden Datenbasen verwendet, die in diesem Kapitel erläutert wurden:

- kleine on-line Handschrift-Datenbasis für ein schreiberabhängiges Erkennungssystem für deutsche Schrift (kursiv)
- on-line Handschrift-Datenbasis für ein schreiberunabhängiges Erkennungssystem für deutsche Fließschrift (166 Schreiber)
- kleine off-line Handschrift-Datenbasis für ein schreiberabhängiges Erkennungssystem für deutsche Schrift (konvertierte on-line Daten)
- deutsche handgeschriebene Adreß-Datenbasis (off-line, schreiberunabhängig)
- amerikanische handgeschriebene Adreß-Datenbasis (off-line, schreiberunabhängig)
- Datenbasis bestehend aus maschinengedruckten Dokumenten in schlechter Qualität und verschiedenen Fonts

Im Kap. 3 wurden neben der Spezifikation der Trainings- und Testdaten auch die jeweilige Vorverarbeitung und Merkmalextraktion je Datenbasis vorgestellt. Diese wurden an die Gegebenheiten und Charakteristika der jeweiligen Datenbasis angepaßt. Allgemeingültig ist jedoch der segmentierungsfreie Ansatz (keine Zeichentrennung) und die 'sliding window' Technik für die Merkmalextraktion. Zur Vorverarbeitung gehören gegebenenfalls die Korrektur der Zeilen- und Zeichenneigung, die Bestimmung der Basislinien und die Größennormierung. Die extrahierten Merkmale setzen sich aus Winkelmerkmalen und Bitmaps, DCT-Koeffizienten oder schriftspezifischen Merkmalen wie Krümmungen und Punkten zusammen. Als Resultat ergibt sich jeweils eine Sequenz von Merkmalvektoren $\underline{x}$, die direkt oder nach einer weiteren Transformation (LDA, Quantisierung) zum Training und Testen der HMM-basierten Erkennungssysteme zur Verfügung stehen.

Kapitel 4

Hybride Schrifterkennungssysteme

Als eine weitere Variante zu den Standard-Modellierungstechniken für Hidden Markov Modelle, wie sie in Kapitel 2.2 beschrieben sind, werden hier zwei verschiedene hybride Erkennungssysteme vorgestellt. Hybrid bedeutet in diesem Zusammenhang die Kombination von HMMs und Neuronalen Netzen (NN), wobei die Anwendung der NN sehr unterschiedlich sein kann. In [Ben95] wird beispielsweise ein SDNN (Space Displacement NN) zur Einzelzeichenerkennung in Kombination mit HMMs verwendet, in [Sch94] wird die Verwendung eines Time Delay NN (TDNN) zur Zeichenerkennung in Verbindung mit HMMs zur Nachbearbeitung vorgestellt, und in [Sch97b] ein NN zur Wahrscheinlichkeitsschätzung für kontinuierliche HMMs. Das Ziel ist, die Vorteile der Hidden Markov Modelle – die sich insbesondere für die Erkennung von Merkmalsequenzen (ohne Segmentierung) eignen – mit denen der Neuronalen Netze – die sehr gute Schätzungen der Posterior-Wahrscheinlichkeit liefern – zu verbinden.

In Kap. 4.1 wird ein hybrides System beschrieben, welches auf einer diskreten HMM-Struktur beruht. Hier wird das NN als Vektorquantisierer eingesetzt (vgl. [Neu99, Rot99, Neu01, Neu97a, Rig98b, Bra99b, Bra00b, Rot00a]). Kap. 4.2 beschreibt ein Erkennungssystem mit einer semi-kontinuierlichen Struktur, wobei das NN zur Schätzung der Emissionswahrscheinlichkeiten dient (vgl. auch [Bou94, Rot00b, Bra00a]).

4.1 Maximum-Mutual-Information (MMI)

Dieses hybride Erkennungssystem basiert auf einem neuronalen Vektorquantisierer (VQ), der nach dem Maximum-Mutual-Information (MMI, maximale Transinformation) Kriterium trainiert wurde, und beruht infolge dessen auf diskreten HMMs.

Ausgangspunkt für dieses MMI-hybride System ist ein mit dem k-means Vektorquantisierer trainiertes diskretes System, dessen VQ-Partitionen zur Initialisierung des NN dienen. Hier wird der übliche k-means Vektorquantisierer durch ein neuronales Netzwerk (winner-takes-all) ersetzt. Das Training des neuronalen Netzes mit allen Trainingsvektoren $\underline{x}$ resul-

tiert in einer Sequenz Y der VQ-Label y_j (Gl. 4.1).

$$\underline{x} \overset{NN-VQ}{\longrightarrow} y_j \tag{4.1}$$

Im Unterschied zur k-means Vektorquantisierung, welches ein unüberwachtes Verfahren darstellt, wird der MMI-VQ im überwachten Modus trainiert. Die Codebuchgröße J wird wie bei der üblichen diskreten Technik zuvor gewählt. Die Label der Merkmalvektoren, also die Zustandssequenz S die beim Training eines Wortes W durchlaufen wird, muß bekannt sein. Diese Information wird aus den bekannten Wort-Trainingslabeln und dem üblichen diskreten HMM-System gewonnen. Als Struktur des neuronalen VQs, der hier verwendet wird, wird wiederum ein euklidischer Abstandsklassifikator gewählt.

Die Zielfunktion des NN-VQ ist die Maximierung der Transinformation $I(Y, W)$ zwischen den VQ-Partitionen y_j und den entsprechenden zugehörigen Klassen-Labeln w_i nach dem informationstheoretischen Kanalmodell (siehe Abb. 4.1, Bergersches Diagramm):

$$I(Y, W) = H(Y) - H(Y|W) = H(W) - H(W|Y) \tag{4.2}$$

$H(Y)$ ist dabei die Entropie der neuronalen Feuerungssequenz Y, $H(Y|W)$ die Äquivokation und $H(W)$ die Entropie der verschiedenen Klassen (hier: Buchstaben), die mit der Folge der Trainingsvektoren $X = \underline{x}(1), ..., \underline{x}(t)$ korrespondiert.

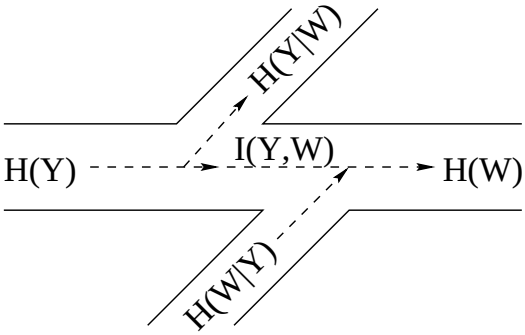


Abbildung 4.1: Informationstheoretisches Kanalmodell

Diese Entropiebegriffe lassen sich aus Sicht der Informationsübertragung anhand eines gestörten Nachrichtenkanals nach Abb. 4.1 veranschaulichen. Die VQ-Label entsprechen dabei der Sendeinformation und die erkannten Worte der Empfangsinformation.

Mit Hilfe von Gradientenabstiegsverfahren (RPROP: resilient backpropagation, siehe z.B. [Zel97]) wird das neuronale Netz so trainiert, daß die Transinformation $I(Y, W)$ in Gleichung 4.2 maximiert wird (siehe [Neu99] für eine ausführliche Herleitung). Das Ziel ist, die Irrelevanz (oder Dissipation) $H(W|Y)$ durch die Bestimmung der VQ-Partitionen zu minimieren. Das lokale Optimierungsverfahren (RPROP) minimiert schrittweise die Dissipation $H(W|Y)$ bzw. $H(S|Y)$, mit

$$H(S|Y) = -\sum_i \sum_j P(s_i, y_j) \cdot \log P(s_i|y_j) \tag{4.3}$$

wobei für das Verfahren nur die Ableitung eine Rolle spielt, die aus den Trainingsdaten ermittelt werden kann (siehe [Neu99]):

$$\frac{\partial H(S|Y)}{\partial P(s_i, y_j)} = -\log \frac{P(s_i, y_j)}{\sum_r P(s_r|y_j)} = -\log P(s_i|y_j) \tag{4.4}$$

Ein typischer Verlauf der Transinformation nach mehreren RPROP-Iterationsschritten ist in Bezug auf die Experimente zur Adreßerkennung in Abb. 9.4 dargestellt.
Der Quantisierungsfehler, der bei der üblichen k-means Quantisierung durch die Annahme

$$p(\underline{x}|s) = p(y_n|s) \tag{4.5}$$

entsteht, kann so verringert werden. Das MMI-hybride Erkennungssystem läßt sich als eine Art semi-kontinuierliche Modellierungsstruktur auffassen, wobei die Wahrscheinlichkeitsdichten besser als beim diskreten System berücksichtigt werden:

$$p(\underline{x}|s) = \frac{P(\underline{x})}{P(y_n)} \cdot p(y_n|s) \tag{4.6}$$

Wie in [Rig98c, Neu99] gezeigt wird, führt das konsequente Training der HMMs in Verbindung mit den Gewichtsvektoren des neuronalen Netzes nach dem Maximum Likelihood (ML) - Kriterium zu dem Maximum Mutual Information (MMI, maximale Transinformation) - Trainingskriterium für das neuronale Netz (vgl. auch [Van02]).

4.2 Tied-Posterior (TP)

Im Gegensatz zum MMI-hybriden Erkennungssystem wird hier das Neuronale Netz als Approximator für die Wahrscheinlichkeitsdichtefunktion semi-kontinuierlicher HMMs verwendet (vgl. [Neu97b, Rot00b, Sta00, Hut94] für Anwendungen in der Spracherkennung). Es gilt für die Emissionswahrscheinlichkeit b_i im Zustand s_i der folgende Zusammenhang:

$$b_i(\underline{x}) = p(\underline{x}|s_i) = \sum_{j=1}^{J} \omega_{ij} \cdot p(\underline{x}|m_j) = \sum_{j=1}^{J} \omega_{ij} \cdot \frac{P(m_j|\underline{x})}{P(m_j)} \tag{4.7}$$

Bei diesem Verfahren werden die bedingten Wahrscheinlichkeitsdichten $p(\underline{x}|m_j)$ (Gauß-Funktionen) eines üblichen semi-kontinuierlichen Systems (vgl. Gl. 2.24) durch die Posterior-Wahrscheinlichkeiten $P(m_j|\underline{x})$ ersetzt, welche vom NN geschätzt werden. Nach Bayes Regel

$$\frac{p(\underline{x}|m_j)}{p(\underline{x})} = \frac{P(m_j|\underline{x})}{P(m_j)} \tag{4.8}$$

wird außerdem die a priori Wahrscheinlichkeit $P(m_j)$ benötigt, die aus den Trainingsdaten ermittelt werden kann (J ist ist die Anzahl der Mischverteilungen m_j, bzw. die Größe

des Codebuches oder die Größe der Ausgangsschicht des neuronalen Netzwerkes; die Wahrscheinlichkeitsdichte $p(\underline{x})$ ist unabhängig vom Zustand s_i).

Das NN ist ein Multi-Layer-Perceptron (MLP) mit einer verdeckten Schicht, welches nach der Backpropagation-Methode (mit der Kreuz-Entropie als Fehlerfunktion) trainiert wird. Die Eingabeinformation bilden die Merkmalvektoren $\underline{x}$, wobei in der Regel auch mehrere benachbarte Vektoren bzw. Frames berücksichtigt werden, sodaß im Durchschnitt alle (oder ein großer Anteil) Merkmalvektoren eines Zeichens gleichzeitig am Eingang liegen. Der Netzausgang wird im Training auf das entsprechende Zeichen-Label gesetzt. Die Zugehörigkeit von bestimmten Frames zu den entsprechenden Zeichen-Labeln muß zuvor von einem initialen System als 'Alignment' (nur die Wort-Label sind im Training bekannt) ermittelt werden. $P(m_j)$ beschreibt dabei die Häufigkeit, mit der ein zu einer bestimmten Klasse (J verschiedene Zeichen) gehörender Vektor im Training vorkommt.

Die Gewichte ω_{ij} werden wie im semi-kontinuierlichen System nach dem Baum-Welch Verfahren geschätzt. Werden die Gewichte so gewählt, daß jeweils nur ein Ausgang des NN aktiv ist,

$$\omega_{ij} = 1 \quad \text{falls: } i = j \qquad \text{und} \qquad \omega_{ij} = 0 \quad \text{sonst} \tag{4.9}$$

ergibt sich der hybride Ansatz nach Bourlard, der in [Bou94] vorgestellt wurde und gerade in der Spracherkennung häufig als hybrides Standardverfahren angesehen wird.

Der wesentliche Vorteil des hier vorgestellten Ansatzes ist jedoch die Unabhängigkeit der Posterior-Wahrscheinlichkeiten von der Anzahl der HMM-Zustände durch die Wahl der Gewichte. Das bedeutet, daß in diesem TP-hybriden System die Anzahl der Zustände pro Zeichen größer sein kann als eins (im System nach [Bou94] können nur Modelle mit einem Zustand modelliert werden). Die Anzahl der Netzausgänge hängt von der Anzahl der Modelle (Zeichen oder Phoneme) ab, woraus folgt, daß die Codebuchgröße der Modellanzahl entsprechen muß. Dies ist in diesem TP-hybriden System auf den ersten Blick ein Nachteil, da für entsprechende Standard tied-mixture Schrifterkennungssysteme – wie in Kap. 9.3.2 gezeigt wird – die Codebuchgröße für eine gute Erkennungsrate deutlich höher ist als die Modellanzahl. Ein weiterer Vorteil im Vergleich zum hybriden Standardverfahren ist die Eingabe von multi-frame (mehrere benachbarte Merkmalvektoren) Eingaben, die, wie auch die Anpassung an Kontext-abhängige Modellierungsstrukturen (vgl. auch [Fri97]), mit diesem TP-hybriden System einfach möglich ist. Werden kontextabhängige Modelle (Triphone oder sog. Trigrapheme) eingeführt, bleibt das ursprüngliche NN erhalten und nur die Gewichte werden nach dem ML-Verfahren neu bestimmt.

4.3 Kapitelzusammenfassung

Im vorliegenden Kapitel wurden zwei unterschiedliche Kombinationsmethoden von neuronalen Netzen mit Hidden Markov Modellen vorgestellt, woraus sich jeweils ein sogenanntes

hybrides Erkennungssystem ergibt. Die MMI-hybride Methode (Maximum Mutual Information) beruht auf einem neuronalen Vektorquantisierer, was bedeutet, daß es sich um eine diskrete Modellstruktur handelt. Bei dem TP-hybriden Verfahren (Tied Posterior) wird das neuronale Netz zum Schätzen der Emissionswahrscheinlichkeiten genutzt. Die HMM-Struktur dieses Verfahrens basiert auf einer semi-kontinuierlichen Modellierung.

Vergleichende Versuchsreihen zu der MMI-hybriden Technik wurden sowohl anhand der on-line als auch der off-line Datenbasen durchgeführt und sind in den entsprechenden Abschnitten des Kap. 9 näher beschrieben. Die Versuche zu den TP-hybriden Systemen beschränken sich auf die off-line Datenbasen (siehe Kap. 9.3).

Kapitel 5

Sprachmodelle

Für die Schrifterkennung ist nicht nur die Erkennung einzelner Buchstaben anhand von extrahierten Signalmerkmalen erforderlich (z.B. Modellierung durch HMMs), sondern auch die Verwendung von Kontextinformation (durch Sprachmodelle) in Form von Lexika, linguistischem oder stochastischem Wissen oder Grammatiken. Die Notwendigkeit von Sprachmodellen bzw. Kontextwissen kann man leicht nachvollziehen, wenn man selbst versucht undeutliche Schrift in einer unbekannten Sprache (kein Wissen über das Vokabular und die Satz-Grammatik) oder Einzelzeichen ohne Wortzusammenhang (vgl. Abb. D.1) zu lesen.

Statistische Sprachmodelle (siehe auch [Jel98, Ros00]) schätzen die Auftrittswahrscheinlichkeiten verschiedener linguistischer Einheiten wie Buchstaben, Wörter oder Sätze. Dies bedeutet, daß Sprachmodelle unabhängig von der Signaldatenbasis (Handschrift, Dokumente, Spracheingabe) sind und nur von der Anwendung (Thema: allgemein, technisch, Korrespondenz, etc.) und der Sprache (deutsch, englisch, etc.) abhängen.
Die folgenden Techniken zur Ermittlung statistischer Sprachmodelle haben sich in den letzten Jahren in erster Linie für die automatische Erkennung natürlicher Sprache etabliert (vgl. [Ros00]):

- N-Gramme auf der Basis von Buchstaben oder Worten, die die Wahrscheinlichkeit von aufeinanderfolgenden Einheiten angeben
- Entscheidungsbäume (anhand von Fragen basierend auf dem Vokabular)
- linguistische Modelle (Kontext-freie Grammatiken (CFG)), die aus Symbolen und Regeln bestehen
- sogenannte ’dependency grammars‘ (DG), die Wortabhängigkeiten mittels eines Graphen modellieren; Vorteil zum N-Gramm: Abhängigkeiten von nicht zwingend aufeinanderfolgenden Worten
- semantische Analysen (’latent semantic analysis’ (LSA)), mit deren Hilfe das wahrscheinliche Vokabular eingeschränkt werden kann

Zur Bestimmung von Sprachmodellen ist jeweils ein großer Text-Corpus erforderlich (ASCII-Text, keine Bilddaten). Am weitesten verbreitet sind die N-Gramm-Statistiken (siehe auch Kap. 5.2). Diese beziehen sich in der Regel auf Übergangswahrscheinlichkeiten von zwei ('bigram') oder drei ('trigram') aufeinanderfolgenden Worten. Sie können aber auch, wie in dieser Arbeit, auf Buchstabenebene (vgl. [Elm98, Baz99, Wil00b, Bra00c, Bra01c, Bra02c]) mit hoher Kontexttiefe (N>3) als Alternative zum fest vorgegebenen (begrenzten) Vokabular verwendet werden.

Die weiteren oben aufgeführten Sprachmodelle (siehe z.B. [Bel98, Coc98]) basieren auf einem Lexikon und werden auf Sätze oder ganze Textpassagen angewandt. Dies ist jedoch nicht das Thema dieser Arbeit, die sich mit der Erkennung von Einzelworten (oder auch kurzen Wortsequenzen) befaßt. Weitere Anwendungsbereiche von Sprachmodellen sind die Satzerkennung, die Dokumentklassifikation (z.B. [Jun98]), die Themenerkennung oder auch die Erkennung von (Fremd-) Sprachen.

Die Anwendung von Sprachmodellen im Zusammenhang mit der automatischen Schrifterkennung (vgl. auch [Baz99, Pit00, Sri93]) wird im folgenden Kapitel 5.1 erläutert und die Berechnung von Backoff Buchstaben N-Grammen in Kap. 5.2. In Kap. 5.3 werden die Trainingstexte der hier verwendeten deutschen und englischen N-Gramme beschrieben.

5.1 Anwendung von N-Grammen

Sprachmodelle können entweder als Alternative, wie in dieser Arbeit, oder als Ergänzung zur Wörterbuch-basierten Erkennung eingesetzt werden. Das beste Erkennungsergebnis, die Zeichenfolge W^* mit maximaler Wahrscheinlichkeit $P(W|X)$, hängt nach Bayes Regel (vgl. Gl. 5.1) sowohl vom Merkmalmodell (HMM) als auch vom Sprachmodell (N-Gramm) ab.

$$W^* = \underset{W}{\text{argmax}}\ P(W|X) = \frac{\underset{W}{\text{argmax}}\ P(W) \cdot P(X|W)}{P(X)} = \underset{W}{\text{argmax}}\ P(W)\ P(X|W) \quad (5.1)$$

Dabei ist die Auftrittswahrscheinlichkeit der Beobachtung der Merkmalvektoren $P(X)$ unabhängig von der Zeichenfolge $W = w_1, w_2, .., w_n$. Die bedingte Wahrscheinlichkeit, oder auch Likelihood $P(X|W)$ wird durch die Hidden Markov Modelle bestimmt und $P(W)$ gibt nach Gl. 5.2 die Wahrscheinlichkeit der Zeichenfolge an. Beispielsweise ist die Zeichenkette 'qu' in der deutschen Sprache sehr viel wahrscheinlicher als 'qa'. Diese Wahrscheinlichkeit aufeinanderfolgender Buchstaben kann mit einem Backoff Zeichen N-Gramm (Kontexttiefe N) beschrieben werden (siehe Kap. 5.2), wobei die jeweils letzten (N-1) Zeichen berücksichtigt werden:

$$P(W) = \prod_{i=1}^{n} P(w_i|w_1, w_2, .., w_{i-1}) \approx \prod_{i=1}^{n} P(w_i|w_{i-N+1}, .., w_{i-1}) \quad (5.2)$$

Ein wesentlicher Vorteil eines Sprachmodells auf Zeichenebene ist, daß das Vokabular bzw. das Wörterbuch nicht bekannt sein muß. Lediglich die erlaubten und zu erkennenden Zeichen bzw. Buchstaben sollten bekannt sein. Aber auch hier besteht die Möglichkeit, unbekannte (nicht zu beachtende) Zeichen auf ein separates Modell (OOV) abzubilden. Gerade bei unbekannten Texten zu verschiedenen speziellen Themen kann auch ein allgemeines (großes) Wörterbuch das Vokabular nicht abdecken, sodaß auf diese Weise zusätzliche Fehler entstehen (OOV: out of vocabulary). Ein buchstabenbasiertes Sprachmodell hingegen kann auch zuvor ungesehene Zeichenketten erkennen, indem auf das N-Gramm mit der nächst kleineren Kontexttiefe zurückgegriffen wird.

Nicht nur bei der SEDAL-Datenbasis, die beliebige Texte enthält, sondern auch bei der Erkennung von Adressen kann das Problem eines unbekannten Vokabulars auftreten, wenn die PLZ nicht erkannt wurde oder die Reihenfolge der Adresse (Name, Straße, Stadt, Land) vertauscht oder nicht strukturiert ist. Auch beliebige Adreß-Zusätze oder ausländische Schreibweisen (‘München - Munich’) gehen über das übliche Wörterbuch hinaus. Eine weitere Anwendungsmöglichkeit bei der Postsortierung betrifft Nachsendeaufträge, bei denen der Name erkannt werden muß.

Perplexität

Ein übliches Qualitätsmerkmal von Sprachmodellen ist die Entropie H nach Gl. 5.3 bzw. die Perplexität pp. Die Perplexität ist abhängig vom Sprachmodell und dem zu erkennenden Text und ist ein Maß für die Überraschung bzw. Wahrscheinlichkeit genau diese Zeichenfolgen des Testtextes vorzufinden bzw. zu erkennen (siehe z.B. [Cla97, Man99]). Im allgemeinen bedeutet eine geringe Perplexität eine gute Übereinstimmung zwischen trainiertem Sprachmodell und dem Testtext (ASCII) und damit in der Regel auch eine höhere Erkennungsgenauigkeit.

$$H \approx -\frac{1}{N_d} \cdot \sum_{i=1}^{N_d} \log_2 P(w_i|w_{i-N+1}, ..w_{i-1}) \qquad \text{und} \qquad pp = 2^H \tag{5.3}$$

In Gl. 5.3 beschreibt N_d die Anzahl der Einzelzeichen im Testtext.

Deutlich wird die Aussagefähigkeit der Perplexität vor allem bei Wort-basierten Sprachmodellen (vergleiche z.B. [Mar00]). So wird beispielsweise für Diktiersysteme mit großem Wortschatz eine Perplexität von etwa 240 (oder besser) auf einem allgemeinen englischen Text (Brown Corpus: eine Millionen Worte) und eine Perplexität von etwa 20 für das spezielle Themengebiet der Radiologie angegeben bei Verwendung eines Trigrammes. Diese Werte sollen jedoch nur die Größenordnung bzw. das Verhältnis untereinander verdeutlichen und nicht als absoluter Wert angesehen werden. Die Perplexität von Buchstaben N-Grammen liegt in der Größenordnung von zehn (vgl. auch [Baz99, Bra01c]). Näheres zu den hier verwendeten N-Grammen findet sich bei den Ergebnissen in Kap. 9.

Dekodierung

Für die Dekodierung eines Schriftzuges unter Einbeziehung eines Sprachmodells kann entweder ein modifizierter (zeitsynchroner) Viterbi-Algorithmus verwendet werden (vgl. Kap. 2.2) oder wie in dieser Arbeit, ein Stackverfahren (siehe [Wil00b, Pau92]). Vorteil des zeitasynchronen Stackdekoders ist der einfachere und schnelle Umgang mit Sprachmodellen hoher Kontexttiefe. Anhand des HMM-basierten Merkmalmodells und des Sprachmodells werden Hypothesen für Teildekodierungen gebildet. Nach dem sogenannten A*-Verfahren wird jeweils der Stack mit der besten Hypothese ausgewählt. Im Vergleich zur Viterbi-Dekodierung richtet sich der Stackdekoder nach Wortgrenzen. Eine genauere Beschreibung dieses Stackdekoders, der am Fachgebiet entwickelt wurde, ist in [Wil00b, Wil00a] zu finden. Ein weiterer wichtiger Faktor bei der Dekodierung ist der Anteil des Einflusses des Sprachmodells auf die Erkennung.

5.2 Bestimmung der Backoff Buchstaben N-Gramme

Die Buchstaben N-Gramme verschiedener Kontexttiefe wurden mit dem CMU-Cambridge Toolkit [Cla97] auf verschiedenen Testtexten ermittelt. Ein Buchstaben N-Gramm (oder auch Zeichen N-Gramm) ist ein statistisches Modell, welches die Wahrscheinlichkeit aufeinanderfolgender Buchstaben beschreibt. N-Gramme werden geschätzt, indem die Häufigkeit von Zeichen in einem Text bestimmt wird:

$$P(w_i) = \frac{N(w_i)}{N_g} \tag{5.4}$$

Hier ist N_g die Anzahl aller Zeichen im Trainingstext. Bei der Betrachtung von Einzelzeichen kann man auch bei einem relativ kleinen Trainingstext davon ausgehen, daß jedes der erlaubten Zeichen vorkommt. Ist dies nicht der Fall müssen die Wahrscheinlichkeiten korrigiert werden, wie es anhand der Bestimmung eines Bigrammes dargestellt wird. Prinzipiell gibt es zwei Standard-Methoden zur Beachtung von wenigen oder ungesehenen Daten bzw. Zeichenfolgen: die Backoff-Methode (siehe z.B. [Kat87]), die hier im folgenden verwendet wird, oder eine Interpolation von Sprachmodellen verschiedener Kontexttiefe, die gewichtet werden.

Ein Backoff Bigramm wird nach folgender Gleichung berechnet:

$$P(w_i|w_{i-1}) = \begin{cases} \frac{N(w_{i-1},w_i)\cdot d}{N(w_{i-1})} & : \; N(w_{i-1},w_i) > \tau \\ P(w_i)\cdot bo(w_{i-1}) & : \; \text{sonst} \end{cases} \tag{5.5}$$

Dabei ist $N(w_{i-1}, w_i)$ die Anzahl, wie häufig das Zeichen w_i dem Zeichen w_{i-1} im Trainingstext folgt. Der Discounting-Faktor d und der Backoff-Faktor bo sind notwendig um

die Wahrscheinlichkeiten für im Text vorhandene und ungesehene Zeichenfolgen zu korrigieren. Das heißt, die Wahrscheinlichkeit von häufigen Zeichenketten wird zugunsten von seltenen und nicht vorkommenden Zeichensequenzen leicht reduziert. Hier wurde das 'linear discounting'-Verfahren (vgl. z.B. [Nie97]) zur Bestimmung verwendet. Der Discounting-Faktor d mit

$$d = 1 - \delta \tag{5.6}$$

reduziert die im Text ermittelte Häufigkeit um den Betrag δ, der die Häufigkeit von Zeichenketten beschreibt, die nur einmal vorkommen. Dabei wird hier für jedes Zeichen der gleiche Discounting-Faktor verwendet. Kommt eine Zeichenfolge weniger als τ mal vor, so wird unter Berücksichtigung des Backoff-Faktors auf das Unigramm zurückgegriffen. Der Backoff-Faktor bo muß unter Berücksichtigung von Gl. 5.5 so bestimmt werden, daß die folgende Gleichung zutrifft, wobei N_l die Anzahl der verschiedenen Label bzw. Buchstaben angibt:

$$\sum_{w_i=1}^{N_l} P(w_i|w_{i-1}) = 1 \tag{5.7}$$

N-Gramme höherer Kontexttiefe werden analog berechnet (siehe Gl. 5.8 zur Bestimmung eines Trigrammes).

$$P(w_i|w_{i-2}, w_{i-1}) = \begin{cases} \frac{N(w_{i-2},w_{i-1},w_i)\cdot d}{N(w_{i-2},w_{i-1})} & : \; N(w_{i-2}, w_{i-1}, w_i) > \tau \\ P(w_i|w_{i-1}) \cdot bo(w_{i-2}, w_{i-1}) & : \text{ Bigramm } w_{i-2}, w_{i-1} \text{ existiert} \\ P'(w_i|w_{i-1}) & : \text{ sonst} \end{cases} \tag{5.8}$$

$P'(w_i|w_{i-1})$ führt dabei auf das Bigramm ohne Berücksichtigung des Discounting-Faktors zurück. Auch hier muß der Backoff-Faktor so bestimmt werden, daß wiederum gilt:

$$\sum_{w_i=1}^{N_l} P(w_i|w_{i-2}, w_{i-1}) = 1 \tag{5.9}$$

Würde diese Korrektur mittels Discounting und Backoff nicht durchgeführt werden, hätte eine im Training nicht vorkommende Zeichensequenz die Wahrscheinlichkeit Null und könnte somit niemals in einem Testtext erkannt werden. Aufgrund dieser 'Smoothing'-Technik ist deren Wahrscheinlichkeit zwar immer noch sehr gering (dies ist ja auch das Ziel eines statistischen Sprachmodells), aber nicht mehr unmöglich.

In der Erkennungsphase wird bei einem Backoff N-Gramm rekursiv auf ein (N-1)-Gramm zurückgegriffen – unter Beachtung des Backoff-Faktors – wenn eine Zeichenkette im Training selten oder unbekannt war. Durch dieses schrittweise Zurückgehen in der Kontexttiefe, welches auch bei der Bestimmung der N-Gramme berücksichtigt wird (vgl. Gl. 5.8), ist gewährleistet, daß auch ein unbekanntes und somit unbegrenztes Vokabular bearbeitet und erkannt werden kann. Aufgrund der Schätzung der Buchstaben N-Gramme wird deutlich,

daß für unterschiedliche Sprachen (deutsch, englisch) oder auch unterschiedliche Themen (allgemeiner Text, nur Adressen) spezielle Sprachmodelle verwendet werden müssen. So ist im englischen die Zeichenfolge ‘the’ im Vergleich zum deutschen sehr viel wahrscheinlicher. Und Wortabschnitte wie ‘str’ oder ‘burg’ kommen in Adressen deutlich häufiger vor als in allgemeinen deutschen Texten.

5.3 Verwendete Sprachmodelle

In dieser Arbeit kommen Zeichen N-Gramme unterschiedlicher Kontexttiefe (N=2,3,5,7) als Alternative zum Lexikon zum Einsatz. Diese Sprachmodelle wurden auf englischen oder deutschen Texten (je nach Anwendung) mit Hilfe des CMU-Cambridge-Toolkits erstellt. Die verwendete Texttrainingsmenge ist im folgenden näher erläutert. Standardmäßig wurde die ‘Linear discounting’ Methode verwendet. Die Anzahl der zugelassenen Zeichen, die im Sprachmodell berücksichtigt werden (also nur Buchstaben oder auch Sonderzeichen) hängt von der späteren Anwendung ab. Kommen im Trainingstext unbekannte oder nicht erlaubte Zeichen vor (hier z.B. ‘# {’), so werden diese auf ein OOV-Modell abgebildet.

Deutscher Text-Datensatz

Je nach Anwendungsbereich wurden verschiedene deutsche Sprachmodelle auf unterschiedlichen Texten erstellt. Für das *allgemeine* deutsche Buchstaben N-Gramm wurden ca. vier Millionen Wörter von zufällig ausgewählten deutschsprachigen Web-Seiten ausgewertet (Textausschnitte siehe Kap. E, ein großer Anteil des Textes besteht aus Nachrichten). Dies führt zu 19505 verschiedenen Trigramm-, 195877 verschiedenen 5-Gramm- und 610899 verschiedenen 7-Gramm-Buchstabensequenzen, wenn 87 verschiedene Einzelzeichen (Groß- und Kleinbuchstaben, Zahlen, einige Sonderzeichen) zugelassen werden.

Eine weitere N-Gramm-Kategorie ist für den direkten Vergleich Lexikon - Sprachmodell vorgesehen und basiert nur auf dem Text des entsprechenden Wörterbuches. Hier ist der Trainingstext zur Bestimmung der N-Gramme zwar extrem klein, aber sehr spezifisch.

Des weiteren wurden für das Adreß-Erkennungsproblem gezielt *spezielle* Buchstaben N-Gramme trainiert. Der zugehörige Text besteht entweder aus einer Liste aller deutschen Städte- bzw. Straßen-Namen, oder auch aus deren Kombination mit den am häufigsten vorkommenden Namen (je Postamt oder nach Einwohnerzahl der Großstädte). Um spezifischer auf das Anwendungsproblem eingehen zu können, wurden jeweils getrennte Sprachmodelle zur Erkennung von Städten und Straßen ermittelt. Hier läßt sich ein allgemeingültiger (für alle Postämter) Trainingstext zur N-Gramm Modellierung von Städten relativ einfach bestimmen (im Vergleich zu den Straßen-Namen), indem die Namen der Großstädte relativ zur Einwohnerzahl häufiger vorkommen. Eine Art Adaption des Sprachmodells kann erfolgen, indem nicht nur die Großstädte überproportional im Trainingstext vertreten sind, sondern

auch die im jeweiligen Postamt am häufigsten vorkommenden Adressen. Dieses Verfahren läßt sich auch leicht auf den Straßen-Trainingstext übertragen. Diese Methode ist zulässig und auch erfolgreich, da man festgestellt hat, daß die in einem Postverteilzentrum eintreffenden Briefe zu einem großen Teil in die nähere Umgebung verschickt werden.

Englischer Text-Datensatz

Die englischen Zeichen N-Gramme wurden anhand von etwa vier Millionen Wörtern von zufällig ausgewählten englischsprachigen Web-Seiten generiert, die sich auf beliebige zufällig ausgewählte Themen beziehen (Textausschnitte siehe Kap. E). Aus 79 möglichen erlaubten Einzelzeichen (Buchstaben: ‘A-Z’, ‘a-z’; Zahlen: ‘0-9’; Sonderzeichen wie z.B. ‘.,:;?/-+()´ “ & %’, Start- und Endsymbole) ergeben sich 3711 Bigramm-, 26129 Trigramm-, 110342 verschiedene 5-Gramm- und 198049 verschiedene 7-Gramm-Buchstabensequenzen.

5.4 Kapitelzusammenfassung

Sprachmodelle beeinflussen neben den Merkmalmodellen (HMMs) den Erkennungsvorgang und sind unabhängig von Bild- oder Signalmerkmalen. Neben einer allgemeinen Einführung in Sprachmodelle wird im Kap. 5 die Bestimmung und Anwendung von N-Gramm-Statistiken erläutert.

In dieser Arbeit werden als Alternative zur Wörterbuch-basierten Erkennung mit Hidden Markov Modellen Backoff Buchstaben N-Gramme verwendet. Diese Sprachmodelle werden anhand von Beispieltexten (ASCII) bestimmt und schätzen die Wahrscheinlichkeit aufeinander folgender Zeichen. Da Sprachmodelle deshalb sowohl von der Sprache (deutsch, englisch) als auch vom Thema (allgemeiner Text, Nachrichten, Adressen) abhängen, wurden für die Anwendungsbeispiele unterschiedliche N-Gramme gebildet. Die Kontexttiefe variiert zwischen Bigrammen (N=2) und sehr großen Modellen mit N=7. Als Qualitätsmaß von N-Grammen wurde die Perplexität eingeführt. Die Dekodierung einer Schriftprobe erfolgt mit einem Stackdekoder, der die Wahrscheinlichkeiten der HMMs mit denen der N-Gramme kombiniert. Die für die Versuche in Kap. 9 verwendeten Sprachmodelle wurden einschließlich der benötigten Trainingstexte in Kap. 5.3 beschrieben.

Kapitel 6

Kontextmodelle

In den vorherigen Kapiteln wurde als Basiseinheit für die HMM-Modellierung jeweils ein Zeichen bzw. Buchstabe (Monophon/Monographem) angesehen, sodaß je nach Anwendung und Anzahl zugelassener Sonderzeichen (Satzendezeichen, mathematische Symbole, etc.) etwa 80 verschiedene HMMs und deren Verkettung für Training und Dekodierung benutzt wurden. Die Verwendung von Kontextmodellen (kontextabhängige Modelle: Bi- oder Triphone in der Spracherkennung bzw. Bi- oder Trigrapheme in der Schrifterkennung) beinhaltet nun die Berücksichtigung des linken und/oder rechten Nachbarn – also des Kontextes – des jeweiligen Zeichens im Wort. Dadurch wird die Anzahl der Modelle stark erhöht, was im weiteren noch ausführlicher dargestellt wird.

Kontextmodelle auf der Basis von Triphonen sind in der Spracherkennung allgemein üblich, da sich hier eine deutliche Steigerung der Erkennungsrate durch den Übergang von Monophonen zu Triphonen beobachten läßt (siehe z.B. [Bah80, ST95, Wil99]). Wie man leicht selbst nachvollziehen kann, ändert sich die Aussprache eines Phonems – und somit auch die Merkmalvektoren, die für das Erkennungssystem verwendet werden – in Abhängigkeit vom Kontext.

Hier stellt sich nun die Frage, ob dieses Prinzip der kontextbezogenen Phoneme auf die Schrifterkennung (vgl. z.B. [Kos97b, Sch97a, Kos00a]) übertragen werden kann. Kontextmodelle in der Schrifterkennung sollen mit Hilfe von Abb. 6.1 eingeführt werden. Hier wird der Buchstabe ‘n’ im Kontext von ‘o/n/e’ und ‘e/n/#’ im Wort ‘Millionen’ und im Kontext ‘i/n/e’ im Wort ‘eine’ betrachtet (# soll das Fehlen eines Nachbarzeichens symbolisieren).
Betrachtet man die Schriftproben von Schreiber A (Abb. 6.1 links), ist festzustellen, daß der Buchstabe ‘n’ unterschiedlich geschrieben wird, abhängig davon, ob der vorherige Buchstabe auf Höhe der unteren Basislinie (z.B. ‘i’,‘e’) oder der oberen Basislinie (z.B. ‘o’) endet. Diese Regel trifft allerdings nicht auf die Schreibweise von Schreiber B zu. Änderungen in der Schreibweise betreffen also in erster Linie die Übergänge zu den Nachbarzeichen. In diesem Beispiel könnte es für den Schreiber A Sinn machen, verschiedene HMMs für den Buchstaben ‘n’ zu trainieren: nämlich z.B. die Trigrapheme ‘o-n+e’ und ‘i-n+e’. Dies führt jedoch

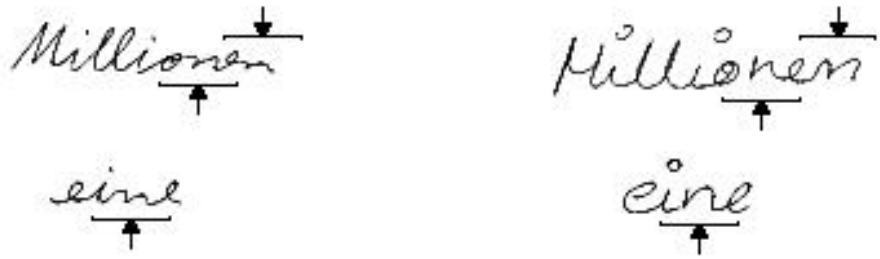


Abbildung 6.1: Beispiele für einen unterschiedlichen Kontext des Buchstaben ‘n’ von zwei Schreibern A (links) und B (rechts)

zu einer deutlichen Erhöhung der benötigten Trainingsdaten. Außerdem stellt sich speziell für ein schreiberunabhängiges System das Problem, inwieweit bestimmte Charakteristika in der Schreibweise für viele Schreiber zutreffen und inwieweit die Schreibweise wirklich in erster Linie von den Nachbarzeichen abhängt. Betrachtet man beispielsweise die Adressen in Abb. C.1, scheint die Varianz in der Schreibweise von einem Schreiber zum anderen so stark zu variieren (Druck- und/oder Schreibschrift), daß der Kontext der Nachbarzeichen nur eine untergeordnete Rolle spielt. Trigrapheme werden, wie oben schon verwendet, folgendermaßen notiert:

$Vorgänger - Zeichen_{aktuell} + Nachfolger$: ‘L’-Zeichen+‘R’

für Bigrapheme gilt entsprechend:

$Vorgänger - Zeichen_{aktuell}$ oder $Zeichen_{aktuell} + Nachfolger$.

Ausgehend von den guten Ergebnissen in der Spracherkennung befaßt sich diese Arbeit mit den allgemeineren Trigraphemen statt Bigraphemen. Die folgenden Kapitel beschreiben das Verfahren zur Erstellung der Trigrapheme und auch verschiedene Clustermethoden, um die Modellanzahl einzuschränken.

6.1 Erstellung und Verwendung von Trigraphemen

Geht man von einer Modellanzahl (bzw. Zeichenanzahl) von $K = 80$ aus, ergeben sich, wenn man alle möglichen rechten und linken Nachbarzeichen berücksichtigt, $K \cdot K \cdot K = 512000$ verschiedene Trigrapheme. Werden entsprechend nur (rechts- oder links-kontextabhängige) Bigrapheme erstellt, führt dies immer noch zu $K^2 = 6400$ verschiedenen Modellen.
Beispielsweise ändert sich das Lexikon (und auch die Label der Trainingsdaten) bezüglich der Modelle beim Übergang von Mono- zu Trigraphemen wie folgt:

- Hamburg: H a m b u r g → #-H+a H-a+m a-m+b m-b+u b-u+r u-r+g r-g+#

Will man Monographeme in Trigrapheme überführen, werden die entsprechenden HMMs des zentralen Monographems ($Zeichen_{aktuell}$) vervielfacht und mit den neuen Trigraphem-

Bezeichnungen versehen. Diese Modelle werden wie in Kap. 2.2.3 beschrieben, mit dem Standard Baum-Welch-Verfahren und den neuen Trigraphem-Labeln trainiert. Die sich theoretisch ergebende Modellanzahl kann kaum trainiert werden, da zum einen eine bestimmte Datenmenge je Modell verfügbar sein muß und zum anderen nicht unbedingt alle Trigrapheme (und auch nicht alle Trigrapheme der Testdaten) in den Trainingsdaten vorkommen. Deshalb werden verschiedene Cluster-Verfahren angewandt (siehe Kap. 6.2) um die Parameterzahl wieder einzuschränken.

Von allen Trigraphemen eines Monographems wird in der Regel ein und dieselbe Übergangsmatrix A verwendet ('tying'). Nur die Parameter der Emissionswahrscheinlichkeiten b (Gaußfunktion und Gewichte oder auch diskrete Verteilung, siehe Kap. 2.2) werden für jedes einzelne Trigraphem neu geschätzt. Bei der Verwendung von Trigraphemen bei der Dekodierung stellt sich das Problem, daß im allgemeinen nicht alle Trigrapheme, die in der Testmenge vorkommen, auch in der Trainingsdatenmenge enthalten sind. Diese können folglich nicht entsprechend trainiert werden. Da bei Berücksichtigung aller möglichen Trigrapheme die Modellanzahl nicht mehr praktikabel ist, werden bestimmte Modelle wieder zusammengefaßt. Die Clustermethoden sind im folgenden Kapitel 6.2 beschrieben.

Zu beachten ist, daß die kontextabhängigen Modelle im Gegensatz zu den in Kap. 5 vorgestellten Sprachmodellen auf den Merkmalen des Schriftbildes basieren und nicht auf der Grammatik.

6.2 Clustermethoden für Trigrapheme

Zum Zusammenfassen von Trigraphemen sind mehrere Methoden möglich. Beispielsweise kann man manuell bestimmte Cluster definieren, die logisch erscheinen: z.B. kommt bei Adressen die Endung 'str' häufig vor, welche als Trigraphem separat modelliert werden könnte; eine andere Möglichkeit besteht darin, die zentralen Zustände der zueinander gehörenden Trigrapheme jeweils gleich zu modellieren, da die kontextabhängigen Änderungen im wesentlichen die Zeichenübergänge betreffen. Eine automatisierte Clusterung von Modellen kann mit Hilfe folgender Methoden erfolgen: anhand der Merkmale der Trainingsdaten kann man Modelle zusammenfassen (Datengetriebene Clusterung, 'data-driven') oder anhand von Fragen, die sich auf die Nachbarschaft beziehen (Entscheidungsbaum basierte Clusterung, 'Tree-Based'). Die beiden zuletzt genannten Methoden führen zur Bestimmung von sogenannten Zustands-verknüpften ('Tied-State') Trigraphemen, welche auch in der Spracherkennung (z.B. [You00, ST95]) stark verbreitet sind. Diese sollen in den folgenden Kapiteln bezogen auf die Probleme der Schrifterkennung näher beschrieben werden. Technisch betrachtet ist hier eine deutliche Ähnlichkeit zu den semi-kontinuierlichen bzw. tied-mixture HMMs (siehe Kap. 2.2.2) vorhanden. Bei den tied-mixture HMMs gibt es einen Pool an Gauß-Verteilungen, den sich alle Zustände aller Modelle teilen. Die unterschiedli-

che Zusammensetzung der Gaußschen Mischverteilungen erfolgt über die Gewichte. Bei den tied-state Trigraphemen werden ganze Zustände (mit ihren Mischverteilungen) von bestimmten Trigraphemen mit gleichem zentralen Monographem geteilt, wie in Abb. 6.2 dargestellt. Die Kriterien zur Clusterung, welche Zustände von welchen Trigraphemen zusammen benutzt werden, werden in den folgenden Abschnitten erläutert.

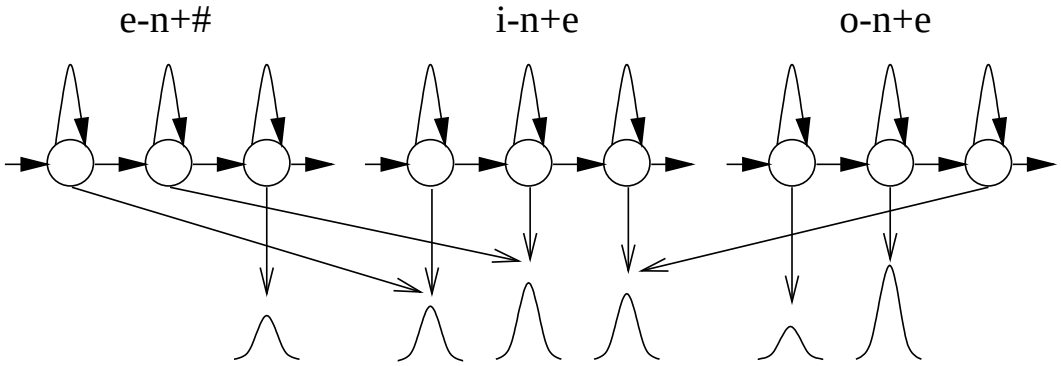


Abbildung 6.2: Tied-State Trigrapheme (zum Beispiel für $*$-n+$*$)

6.2.1 Datengetriebene Clusterung

Die datengetriebene Clusterung (‘data-driven’) beruht auf einem bottom-up Verfahren, wobei nur Trigrapheme mit gleichem zentralen Monographem geclustert werden. Zur Initialisierung wird jedem Zustand ein eigener Cluster zugewiesen. Diese Cluster werden nun so lange mit dem jeweils ‘nächsten Cluster’ verschmolzen, bis eine Abbruchbedingung, die vom Benutzer vorgegeben wird, erfüllt ist. Diese Abbruchbedingung kann z.B. die Größe oder die Anzahl der Cluster angeben. Die Bestimmung des ‘nächsten Clusters’ ist durch den minimalen (gewichteten) euklidischen Cluster-Abstand definiert, der sich wiederum als maximaler Zustands-Abstand d aller Zustände zweier Cluster zueinander ergibt. Für kontinuierliche HMMs mit nur einer Gaußfunktion (Mittelwertvektor $\underline{\mu}$ und Kovarianzmatrix Σ mit den Varianzelementen σ) je Zustand wird der Abstand $d(s_i, s_j)$ zwischen zwei Zuständen s_i und s_j folgendermaßen berechnet (D ist die Dimension der Vektoren):

$$d(s_i, s_j) = \sqrt{\frac{1}{D} \cdot \sum_{k=1}^{D} \frac{(\mu_{ik} - \mu_{jk})^2}{\sigma_{ik} - \sigma_{jk}}} \tag{6.1}$$

Bei semi-kontinuierlichen HMMs wird entsprechend der gewichtete euklidische Abstand der Gewichte der Zustände herangezogen (vgl. auch Kap. 2.2). Sollen Trigrapheme geclustert werden, deren Modelle aus J Gaußschen Mischverteilungen bestehen, wird der Abstand über

die Emissionswahrscheinlichkeit der Mittelwertvektoren $b(\underline{\mu})$ berechnet:

$$d(s_i, s_j) = -\frac{1}{J} \cdot \sum_{k=1}^{J} \log b_j(\underline{\mu}_{ik}) + \log b_i(\underline{\mu}_{jk}) \tag{6.2}$$

Werden von verschiedenen Trigraphemen des gleichen Zentralgraphems jeweils die gleichen Zustandscluster benutzt, ergeben sich sog. 'tied-models'. Im Extremfall entsteht somit wieder genau ein Monographem, falls alle zugehörigen Trigrapheme jeweils über alle Zustände verknüpft sind.

Voraussetzung für dieses datengetriebene Verfahren ist, daß vor dem Clustern alle einzelnen Trigrapheme separat trainiert werden müssen, da sonst kein am nächsten liegendes Cluster bestimmt werden kann. Dies ist auch gleichzeitig ein Nachteil dieses Clusterverfahrens, da Trigrapheme, die im Training nicht vorkommen, nicht richtig zugeordnet werden können (das Cluster-Kriterium basiert auf den Gaußverteilungen, die in diesem Fall nicht modelliert werden konnten).

In der Spracherkennung geht man häufig von sehr viel größeren Datenmengen aus, als das hier bei der Schrifterkennung der Fall ist. Außerdem wird dort nur zwischen etwa 47 Monophonen unterschieden (bei der Schrifterkennung etwa 80 Monographeme), sodaß bei wortinternen Triphonen dieses Problem der nicht im Training vorkommenden Trigrapheme (sog. 'unseen' Zeichensequenzen) selten vorkommt. Erst bei wortübergreifenden Triphonen (oder auch Allophonemen mit mehr als zwei zu berücksichtigenden Nachbarzeichen) ergeben sich ggf. Zeichen- oder Wortsequenzen, die nicht im Trainingsmaterial vorkommen. Dieses Problem wird mit dem zweiten hier vorgestellten Verfahren behoben (siehe Kap. 6.2.2).

6.2.2 Entscheidungsbaum basierte Clusterung

Im Gegensatz zur datengetriebenen Clusterung ist diese Entscheidungsbaum basierte ('tree-based') Methode ein top-down Verfahren. Das Ziel ist wiederum die Zusammenfassung von Zuständen, wobei prinzipiell auch ganze Modelle geclustert werden können. Zu Beginn werden alle zueinander gehörenden Trigrapheme (also Trigrapheme mit gleichem Zentralgraphem) in einem einzelnen Cluster versammelt, der dann anhand von ja/nein-Fragen aufgeteilt wird (vgl. Abb. 6.3). Das Abbruchkriterium bestimmt die Feinheit der Clusterung. Dieses entscheidet anhand der Likelihood der jeweiligen Cluster (und nicht anhand eines Abstandsmaßes wie beim datengetriebenen Clustern) über eine weitere Aufteilung. Anhand der Fragen zum Kontext werden die Cluster sukzessiv aufgespalten, wobei jeweils die Frage die Aufteilung bewirkt, die die Likelihood am besten maximieren kann. Betrachten wir als Beispiel wieder die Trigrapheme 'o-n+e', 'e-n+#' und 'i-n+e', die am Anfang von Kap. 6 vorgestellt wurden, so könnte die Entscheidungsbaum basierte Aufteilung wie in Abb. 6.3 dargestellt, verlaufen. An jedem Knoten des Baumes wird eine Frage bzgl. des linken ('L') oder rechten ('R') Nachbarzeichens gestellt, z.B. "Ist der linke Nachbar ein Zeichen, dessen

Übergang der Schrifttrajektorie auf der oberen Basislinie verläuft?" Diese Frage wäre für das Trigraphem 'o-n+e' mit ja zu beantworten (aber beispielsweise auch für die Trigrapheme 'b-n+∗', 'A-n+∗', usw. wenn man von einer Schreibvariante ausgeht, die in der Schule als Schreibschrift gelehrt wird). Diese Fragen werden für jeden Zustand der Modelle separat gestellt. Da die Verknüpfung bei dieser Methode auf Fragen beruht und nicht auf den Trainingsdaten, können auch Trigrapheme, die in den Trainingsdaten nicht vorkommen, korrekt zugeordnet werden.

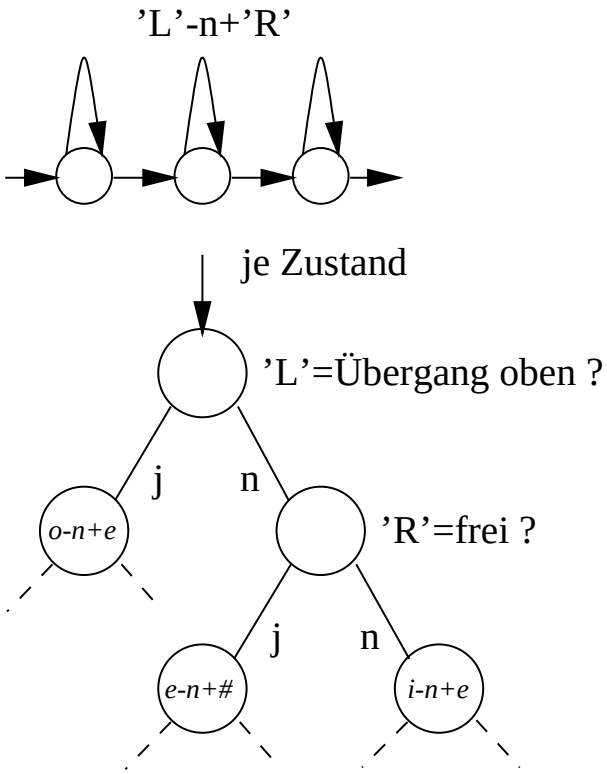


Abbildung 6.3: Entscheidungsbaum zur tree-based Clusterung (zum Beispiel für ∗-n+∗)

Das Problem dieser Cluster-Methode liegt in erster Linie in der Festlegung der Fragen. Prinzipiell besteht die Möglichkeit nach jedem möglichen Zeichen als Kontext (linker und rechter Nachbar) zu fragen und/oder aber nach Zeichengruppen mit bestimmten Merkmalen, anhand derer eine sinnvolle Verknüpfung erfolgen kann.

In der Spracherkennung sind diese phonetischen Gruppen für die Clusterung z.B. Nasale, Vokale, Konsonanten, Frikative, Plosive, Stimmlose, etc. Für die off-line Handschrifterkennung wurden im Rahmen dieser Arbeit die folgenden Gruppen gebildet (für die Experimente wurden entweder alle oder nur eine Auswahl von Fragen gestellt, vgl. Kap. 9):
Ist der rechte (linke) Nachbar ein

- A, B, C, ..., Z, a, b, ...,z, Ä, ä, ..., 0, 1, ..., 9, #, -, ... (separat für jedes Zeichen) ?
- Großbuchstabe, Kleinbuchstabe, Zahl, Sonderzeichen ?

- Buchstabe mit Schriftübergang auf der oberen Basislinie, unteren Basislinie ?
- Zeichen mit Oberlänge, mit Unterlänge ?

Anhand dieser Kriterien kann das Aussehen eines einzelnen Zeichens beurteilt werden: nach Zahlen oder Sonderzeichen wird in der Regel kein fließender Übergang zum nächsten Zeichen durchgeführt; wenn die Nachbarzeichen Großbuchstaben sind, handelt es sich wahrscheinlich um Druckschrift; die Ober- oder Unterlängen (z.B. auch 'T'-Striche) können je nach Merkmalextraktionsverfahren die Merkmalvektoren des zentralen Monographems beeinflussen. Kontextmodelle und entsprechende Entscheidungsfragen für die on-line Handschrifterkennung sind in [Kos00a] erklärt.

Die Voraussetzung für eine gelungene Verknüpfung ist jedoch, wie schon erwähnt, eine kontextabhängige Schreibweise. So wechseln manche Schreiber innerhalb eines Wortes von Schreib- auf Druckschrift oder benutzen andere Schreibstile, die die Varianz der einzelnen Zeichen stark vergrößern. Üblich ist es auch, mit einem großen Druckbuchstaben zu beginnen und das Wort in Schreibschrift zu beenden. Die Fragestellung lautet nun, ob sich diese Vielfalt der Schreibarten bei einem schreiberunabhängigen Erkennungssystem durch bestimmte Fragen geeignet verknüpfen lassen. Beispielsweise funktioniert die Frage nach dem Schriftübergang bei einigen Buchstaben – A, b, o, r, v, w (Übergang auf der oberen Basislinie) – nur, wenn der Schreiber in Norm-Schreibschrift schreibt.

Die Auswirkung der Kontextmodelle auf die Erkennungsrate für die schreiberab- und schreiberunabhängige Handschrifterkennung wird in den Kapiteln 9.3.1 und 9.3.2 erläutert und ausgewertet.

6.3 Kapitelzusammenfassung

Die Modellierung von kontextabhängigen Modellen (Trigrapheme) ist der Spracherkennung entnommen. Kontextmodelle berücksichtigen auf Merkmalebene den rechten und linken Nachbarn des zu modellierenden Zeichens, um auf spezielle Schreibweisen, die durch diesen Kontext entstehen, eingehen zu können. Da bei der Nutzung aller möglichen Trigrapheme die Anzahl der zu schätzenden HMM-Parameter extrem ansteigt und ein robustes Training nicht mehr möglich ist, wurden zwei unterschiedliche Clusterverfahren – die datengetriebene Clusterung und die Entscheidungsbaum basierte Clusterung – zur Verknüpfung von Zuständen beschrieben.

Kapitel 7

Konfidenzmaße

Konfidenzmaße oder Vertrauensmaße sollen die Korrektheit des Erkennungsergebnisses bewerten. Betrachtet man die Einzelzeichenerkennung (oder die Erkennung vorsegmentierter Worte) mit Hilfe von Neuronalen Netzen oder Abstandsmaßen zu bestimmten Prototypen (z.B. KNN-Klassifikator: K-Nearest-Neighbor), scheint diese Forderung selbstverständlich und einfach zu sein. Im allgemeinen liefern diese Techniken bei der Erkennung automatisch eine Art Konfidenzmaß, bzw. ein Ergebnis, das als Wahrscheinlichkeit für eine korrekte Erkennung gewertet werden kann. Für ein HMM-basiertes Erkennungssystem trifft das nicht zu. Wie bereits in Kapitel 2 erläutert wurde, kann das Erkennungsproblem eines statistischen Systems durch folgende Gleichung 7.1 nach Bayes Regel beschrieben werden:

$$P(W|X) = \frac{P(W) \cdot P(X|W)}{P(X)} \tag{7.1}$$

Wird lediglich nur eine Erkennung des besten Wortes W aus einem vorgegebenen Wörterbuch gefordert, so ist der Term $P(X)$ (die a priori Wahrscheinlichkeit der Merkmalvektoren) irrelevant und es ergibt sich Gl. 2.1. Das Erkennungsergebnis an sich und auch die Reihenfolge der N besten Ergebnisse bleibt unabhängig von $P(X)$ unverändert. Das Problem liegt in der Bewertung dieser Erkennungsergebnisse, die sich im wesentlichen auf die Wahrscheinlichkeit $P(X|W)$ stützt. Betrachtet man die Einzelworterkennung mit Hilfe eines Wörterbuches, dessen Einträge jeweils gleich wahrscheinlich sind, spielt auch die Grammatik $P(W)$ keine Rolle mehr. Dies bedeutet, daß das Ergebnis nur das wahrscheinlichste Wort unter Berücksichtigung des Lexikons (mit $P(X|W)$) widerspiegelt und die Erkennungswahrscheinlichkeit kein absolutes Maß für die Korrektheit darstellt, sondern ein relatives. Zur Bewertung muß die Erkennungs-Likelihood also normiert werden.

Ein optimales Konfidenzmaß würde der Posterior-Wahrscheinlichkeit $P(W|X)$ entsprechen. Gehen wir wiederum von einer Einzelworterkennung mit gleichverteilter a priori Wahrscheinlichkeit der Worte $P(W)$ aus, ergibt sich Gl. 7.2 zur Bestimmung des Konfidenzmaßes

$Conf$ mit

$$Conf := P(W|X) = \frac{P(X|W)}{P(X)} \tag{7.2}$$

unter der Voraussetzung, daß es möglich ist, $P(X)$ zu schätzen. Da eine genaue Berechnung von $P(X)$ quasi nicht möglich ist, wird diese Größe entweder mit Hilfe der Likelihoods einer Zwei-Best Erkennung (siehe Kapitel 7.3) oder eines sogenannten Garbage-Modells (siehe Kapitel 7.4) angenähert.

Weitere Möglichkeiten zur Konfidenzmaßbestimmung beziehen sich immer auf eine Normierung. In Kap. 7.2 wird die Häufigkeit der erkannten Worte einer N-Best Liste (großes N erforderlich) zur Berechnung herangezogen, in Kap. 7.5 die Erkennungs-Likelihood eines einfachen Zeichenerkenners ohne Wörterbuch. Als einfach zu berechnendes Konfidenzmaß soll die Frame-normierte Likelihood $P(X|W)$ dienen (siehe Kap. 7.1), um Referenzwerte zu erhalten. Eine weitere Möglichkeit besteht in der Kombination verschiedener Konfidenzmaße (siehe [Dol98]). In der Literatur, speziell für die Erkennung fließend gesprochener Sprache, sind viele andere Definitionen für Konfidenzmaße beschrieben (z.B. [Wil98]), die sich jedoch in der Regel nicht auf die Einzelworterkennung beziehen, sondern ebenfalls die Satz-Grammatik berücksichtigen. Eine völlig andere Methode ('Two-Pass') zur Bewertung der Sicherheit wird in [Wan01] dargestellt. Hier werden nach einer 'normalen' Erkennung (vertikale Merkmalvektoren) in einem zweiten Schritt neue Merkmale (horizontale Merkmalvektoren) in Abhängigkeit der Erkennung und der damit verbundenen Segmentierung gebildet, und mit einem zweiten Set von HMMs werden die Ergebnisse des ersten Durchlaufes verifiziert. Auch diese Übereinstimmung der Ergebnisse kann als Konfidenzmaß gewertet werden.

Unabhängig von der Berechnungsvorschrift für das Konfidenzmaß $Conf$, gilt für die Entscheidung Gleichung 7.3 mit

$$Conf \begin{cases} < \tau \rightarrow zurückweisen \\ \geq \tau \rightarrow klassifizieren \end{cases} \tag{7.3}$$

wobei τ eine festzulegende Schwelle darstellt. Mit Hilfe dieser Schwelle kann der Anteil der zurückgewiesenen Daten – und somit auch der Anteil der Fehler in den restlichen zu klassifizierenden Daten – eingestellt werden. Das Ziel besteht darin, möglichst viele der falsch erkannten und möglichst wenige der korrekt erkannten Worte zurückzuweisen. Als Ergebnis von Versuchen zur Evaluierung von Konfidenzmaßen wird deshalb häufig eine Kurve von Rückweisungsquote in Abhängigkeit der Fehlerquote angegeben, die durch Variieren der Schwelle τ berechnet werden kann (vgl. auch Kap. 9.2.2 und 9.3.2).

Anwendungsbereiche für Konfidenzmaße sind sowohl im Erkennungsmodus als auch beim Training zu finden. Beispielsweise ist es bei einem interaktiven Mensch-Maschine-Dialog (entweder über Sprache oder auch per Handschrifteingabe, PDA) sinnvoll und benutzerfreundlich, wenn die Rückfragen bzgl. der Eingabe begrenzt werden. Häufig wird nach

jeder menschlichen Eingabe das Erkennungsergebnis vom Computer wiederholt, um sicherzugehen, daß das Erkannte stimmt (z.B. automatische Fahrplanauskunft per Telefon). Dieses Ergebnis muß vom Benutzer bestätigt werden. Sind diese Rückfragen aber zu häufig, ist der Anwender leicht genervt. Mit Hilfe von Konfidenzmaßen kann eine benutzerfreundliche Dialogsteuerung erstellt werden.

Aber auch im off-line Bereich gibt es Anwendungsbeispiele. In der Postautomatisierung wäre eine Briefzustellung aufgrund einer unsicheren Erkennung ggf. zeit- und kostenintensiv [Glo97]. Es ist günstiger die Post, deren Adresse mit einem geringen Konfidenzmaß erkannt wurde, zurückzuweisen und per Hand zu sortieren als falsch zuzustellen. Eng verbunden mit der Rückweisung von unsicher erkannten Ergebnissen ist auch die Detektion von Wörtern, die nicht im Vokabular vorkommen (OOV, siehe z.B. [You94]). Ziel ist es, diese Worte auszusondern.

Eine weitere Anwendung ist beim Training des Systems gegeben. Soll ein Erkennungssystem im unüberwachten Modus (keine Label der Trainingsdaten vorhanden; vgl. Kap. 8) weiter trainiert oder adaptiert werden, müssen die Wort-Label vom System selbst erzeugt werden. Dies führt dazu, daß entweder alle Trainingsdaten (richtig und falsch erkannte) zum Nachtrainieren benutzt werden oder aber nur eine Auswahl der Daten, die anhand der Konfidenz erfolgt (vgl. z.B. auch [Wal00]). Einsatzmöglichkeiten liegen hier in der Schreiber-Adaption eines schreiberunabhängigen Handschrifterkennungssystems (z.B. PDA) oder aber im unüberwachten Nachtrainieren auf spezielle Besonderheiten oder Änderungen in der Schrift (Postämter: verschiedene typische Schreibweisen in unterschiedlichen Ländern/ Regionen). In der Spracherkennung werden gerade Broadcast-Erkennungssysteme (automatisches ‘Hören’ von Nachrichtensendungen) im unüberwachten Modus weiter trainiert, da das manuelle Labeln von enormen Datenmengen einfach zu kostspielig ist.

Die detaillierte Beschreibung der in dieser Arbeit untersuchten Konfidenzmaße erfolgt in den anschließenden Kapiteln (vgl. auch [Bra02b]). In Abb. 7.1 werden die verschiedenen Möglichkeiten zuvor grafisch dargestellt. Aufgrund des großen Wertebereiches, der bei der HMM-basierten Erkennung auftritt, werden in der Regel logarithmierte Größen der Wahrscheinlichkeiten verwendet. Dies spielt jedoch prinzipiell bei den folgenden Betrachtungen keine Rolle.

7.1 Normierte Likelihood

Als einfachstes Konfidenzmaß wird die vom HMM ausgegebene logarithmierte Wortwahrscheinlichkeit, die Likelihood $P(X|W)$ gewählt, die auf die Anzahl der zum Wort gehörenden Frames normiert wird. Häufig wird diese Größe als Referenzwert oder auch in Kombination mit anderen Maßen (siehe z.B. [Dol98]) verwendet.

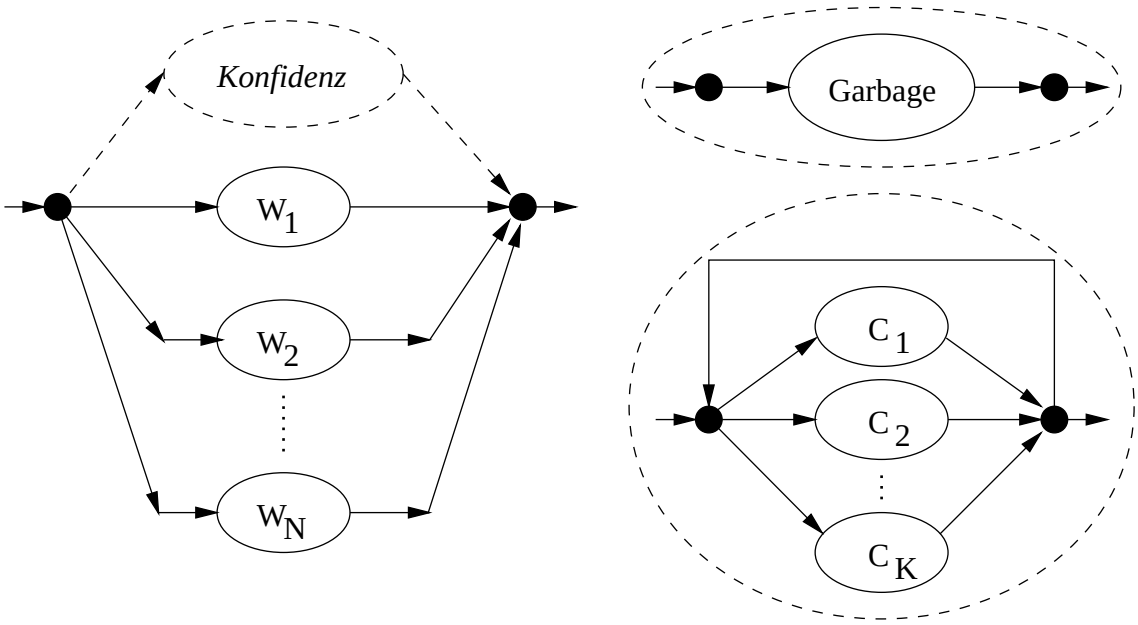


Abbildung 7.1: Konfiguration der Dekodierung zur Bestimmung von Konfidenzmaßen: die wörterbuchbasierte Erkennung (Vokabular $W_1, .., W_N$) kann durch einen weiteren Pfad ergänzt werden um die Konfidenz zu ermitteln: Garbage-Modell (oben rechts) oder zwanglose Zeichenerkennung c_i (unten rechts)

Das Konfidenzmaß wird als Quotient gemäß folgender Gl. 7.4 gebildet:

$$Conf := \frac{P(X|W)}{Anzahl(Frames)} \tag{7.4}$$

Vorteil dieser Definition ist, daß die zur Berechnung notwendigen Größen bei einer Erkennung sowieso vorliegen und somit kein zusätzlicher Rechenaufwand entsteht.

Die Normierung ist notwendig, da die Likelihood von der Länge des Wortes bzw. der Anzahl der Merkmalvektoren abhängt (vgl. Kap. 2.2). Ohne Normierung ergibt sich damit prinzipiell eine unsichere Erkennung von längeren Worten. Dies ist allerdings nicht der Fall. Bei langen Worten gibt es im Wörterbuch sehr viel weniger Verwechslungsmöglichkeiten. Die Rückweisung aller kurzen Worte ist jedoch nicht der Sinn eines Erkennungssystems.

7.2 N-Best Listen

Konfidenzmaße, die auf den erkannten Wort-Labeln der N-Best Listen basieren, sind unabhängig von der Posterior Wahrscheinlichkeit. Sie berücksichtigen die Worthäufigkeiten zur Normierung. Statt einer üblichen Ein-Best Erkennung werden die besten N Ergebnisse (gemäß der Viterbi-Wahrscheinlichkeiten $P(X|W)$) ermittelt (vgl. auch [Kem97]). Diese Top-N Erkennung kann theoretisch N unterschiedliche Lösungen (bzw. Einzelworte) bein-

halten, oder aber auch mehrfach das gleiche Wort gemäß dem Wörterbuch, wobei der Unterschied nur in den Frame-Grenzen der einzelnen Buchstaben zu finden ist (Wortanfang, -ende) oder in der unterschiedlichen Erkennung nicht relevanter Zeichen. Bei einer Satzerkennung oder Wortsequenzerkennung (hier z.B. 'PLZ Ort Zusatz') wird entsprechend die Häufigkeit der Worte an der gleichen oder ähnlichen Position (bzgl. der Frames) untersucht.
Hier wird das Konfidenzmaß folgendermaßen definiert:

$$Conf := \frac{Anzahl(W_{best})}{N} = Häufigkeit(W_{best}) \tag{7.5}$$

Die Likelihoods der Erkennungsergebnisse selbst spielen dabei keine Rolle. Kommt dasselbe Wort häufig unter den am besten passenden Ergebnissen vor, ist die Sicherheit für eine korrekte Erkennung relativ groß. Folglich ist dieses Konfidenzmaß aber auch abhängig von den im Wörterbuch vorkommenden Worten. Da diese Auswertung nur bei einer N-Best Erkennung mit großem N erfolgreich ist (hier z.B. $N = 50$), ist der Rechenaufwand entsprechend hoch. Betrachtet man Abb. 7.1, so kommt dieses Konfidenzmaß ohne den dort eingezeichneten 'Konfidenz-Pfad' aus.

7.3 Zwei-Best Abstand

Dieses Konfidenzmaß basiert auf der Schätzung der Posterior Wahrscheinlichkeit $P(W|X)$, wie in der Einleitung des Kap. 7 bereits erläutert wurde:

$$Conf := \frac{P(X|W)}{P(X)} \tag{7.6}$$

Im Gegensatz zur Verwendung der Worthäufigkeiten einer N-Best Liste nach Gl. 7.5 ist hier nur eine Zwei-Best Erkennung notwendig. Dabei spielt nicht das zweitbeste Ergebnis an sich eine Rolle, sondern nur dessen Likelihood (siehe z.B. auch [Dol98, Eic00] für Anwendungen in der Sprach- und Gesichtserkennung). Die a priori Wahrscheinlichkeit $P(X)$ wird nach folgender Gleichung 7.7 durch die Erkennungs-Likelihood des zweitbesten Ergebnisses W_{2best} angenähert (N beschreibt hier die Größe des Wörterbuches):

$$P(X) \approx \sum_{k=1}^{N} P(X|W_k) \cdot P(W_k) \quad \Rightarrow \quad P(X) \approx P(X|W_{best}) + P(X|W_{2best}) \tag{7.7}$$

Diese Vereinfachung ist möglich, wenn die Wahrscheinlichkeiten der zwei besten Ergebnisse $P(X|W_{best})$ und $P(X|W_{2best})$ als deutlich größer angenommen werden, als die nachfolgenden N-Best Resultate (die a priori Wahrscheinlichkeit $P(W)$ ist wiederum gleichverteilt). Um den großen Dynamikbereich der Werte zu handhaben, kann dieser Ausdruck durch Wahl eines transformierten Konfidenzmaßes (bzw. einer anders definierten Schwelle τ) wie folgt

umgestellt werden:

$$\tau = \frac{P(X|W_{best})}{P(X|W_{best}) + P(X|W_{2best})} \quad \Rightarrow \tag{7.8}$$

$$\tau^* = \frac{P(X|W_{best})}{P(X|W_{2best})} \qquad \text{mit:} \qquad Conf := \tau^* = \frac{\tau}{1-\tau}$$

Dies führt zur Auswertung der Differenz der logarithmierten Wahrscheinlichkeiten der beiden besten Ergebnisse. Somit ist dieses Konfidenzmaß, wie auch die Auswertung der N-Best Liste nach Kap. 7.2 sehr stark vom Wörterbuch abhängig; und zwar nicht nur von der Größe, sondern auch von der Ähnlichkeit der Einträge. Ist der Unterschied der beiden Likelihoods $P(X|W_{best})$ und $P(X|W_{2best})$ groß, erfolgte die Erkennung sehr sicher. Ist der Abstand klein, so kann es dafür zwei Gründe geben: das erkannte Wort ist falsch und kann somit berechtigterweise zurückgewiesen werden, oder die beiden Worte unterscheiden sich nur geringfügig (abhängig vom Vokabular), was zu einer ähnlichen Likelihood führt. Betrachtet man bezogen auf den zweiten Fall beispielsweise die logarithmierten Likelihoods von ‘Hamburg’ und ‘Homburg’, die sich jeweils als Summe der logarithmierten Likelihoods der Einzelbuchstaben ergeben, so ist dort zwangsläufig nur ein geringer Unterschied auszumachen. Wörterbuch-Einträge, deren Levenshtein-Distanz klein ist, werden folglich häufiger zurückgewiesen.

7.4 Garbage-Modelle

Auch dieses Konfidenzmaß, welches durch eine Normierung auf die Likelihood von Garbage-Modellen bestimmt wird, basiert auf der Schätzung der Posterior Wahrscheinlichkeit (siehe z.B. [Ros90, Eic00]). Dabei wird $P(X)$ folgendermaßen angenähert:

$$P(X) \approx P(X|W_{garb}) \tag{7.9}$$

Das Garbage- oder Filler-Modell W_{garb} wird auf allen zur Verfügung stehenden Trainingsdaten trainiert (ohne Berücksichtigung der Label-Information). Es bildet sich also eine Art Durchschnittsmodell. Ein Training auf wirklichem ‘Garbage’ bzw. falschen Daten ist in vielen Fällen sinnvoller (z.B. in [Eic00] zur Unterscheidung: Gesicht/ Nicht-Gesicht), in dieser Anwendung der Schrifterkennung jedoch schwierig zu verwirklichen, da keine konkrete Menge existiert, von der die Schriftdaten zu unterscheiden sind. Das Konfidenzmaß ergibt sich nach Gl. 7.2. Bei der Dekodierung wird nur eine normale Ein-Best Erkennung ausgeführt, allerdings muß, wie in Abb. 7.1 verdeutlicht, ein weiterer paralleler Pfad mit dem Garbage-Modell berücksichtigt werden. Ist der Quotient aus Wort-Likelihood und Garbage-Likelihood groß, passen die Merkmale sehr spezifisch zu einem bestimmten Wörterbucheintrag und eine korrekte Erkennung wird wahrscheinlicher.

7.5 Zwanglose Zeichenerkennung

Wie beim Garbage-Modell ist auch bei diesem Konfidenzmaß ein zusätzlicher Pfad bei der Dekodierung notwendig (siehe Abb. 7.1). Hier erfolgt die Normierung durch die Likelihood einer Zeichenkette C, die anhand einer zwanglosen Erkennung ohne WB nach Gl. 7.10 ermittelt wurde (vgl. [You94, Haz01]):

$$P(X|C) = P(X|c_1, ...c_k) = \prod P(X_{f_i}|c_i) \quad \Rightarrow \quad Conf := \frac{P(X|W)}{P(X|C)} \tag{7.10}$$

Parallel zur Wörterbuch basierten Erkennung wird eine Erkennung auf Zeichenebene mit $P(X|C)$ ohne Vokabular (K verschiedene Zeichen/HMMs erlaubt) durchgeführt, wobei die jeweils verwendeten Frames f für die Zeichenkette und das Wort identisch sein müssen. Die Länge der Zeichenkette muß lediglich größer als ein Zeichen sein; in der Literatur spricht man bei dieser Art von Erkennung manchmal auch von '2+'. Stimmen Zeichenkette und Wort gut überein, bzw. sind die Erkennungswahrscheinlichkeiten sehr ähnlich, ist das Sicherheitsmaß hoch. In der Regel ist $P(X|C)$ größer oder gleich $P(X|W)$. In dem zuvor betrachteten Fallbeispiel der Erkennung von 'Hamburg' oder 'Homburg' wirkt sich dieses Konfidenzmaß jedoch nicht unbedingt positiv aus. So kann es ebenfalls vorkommen (siehe Kap. 9.2.2), daß die WB-Erkennung und die Zeichenerkennung genau das gleiche, aber falsche Ergebnis liefern.

Zusätzlich zum bloßen Vergleich der Wahrscheinlichkeiten kann auch die Levenshtein-Distanz (siehe Kap. A) der beiden Strings (Wort und Zeichenkette) bestimmt und berücksichtigt werden. Sind sowohl die Levenshtein-Distanz als auch der Abstand der entsprechenden Likelihoods groß, kann man daraus auch auf OOV schließen. Außerdem besteht die Möglichkeit, die Zeichenerkennung zwar ohne Vokabular, aber stattdessen mit einem Sprachmodell durchzuführen (vgl. Kap. 5). Wie die meisten hier vorgestellten Konfidenzmaße (außer den N-/Zwei-Best Listen) ist auch dieses Maß unabhängig von der Wahl des Wörterbuches.

7.6 Kapitelzusammenfassung

Im vorliegenden Kapitel wurden fünf verschiedene Konfidenzmaße für die Einzelworterkennung eingeführt, auf Grund derer die Sicherheit/Korrektheit eines Erkennungsergebnisses bewertet werden kann. Dabei dient die Frame-normierte Worterkennungs-Likelihood als Referenz. Die Techniken, die auf der Zwei-Best Erkennung oder dem Garbage-Modell beruhen, bilden das Konfidenzmaß als Schätzung der Posterior Wahrscheinlichkeit. Die N-Best Liste wertet Worthäufigkeiten als Vertrauensmaß und bei der Zeichenerkennung wird das Konfidenzmaß durch die Normierung auf die Likelihood beliebiger Zeichensequenzen ermittelt. Ein weiteres Unterscheidungsmerkmal ist neben dem verursachten Rechenaufwand die Art der Abhängigkeit vom Wörterbuch. Sowohl das Konfidenzmaß basierend auf der N-Best

Liste als auch der Zwei-Best Erkennung hängt von der Wahl des verwendeten Wörterbuches ab. Werden einzelne Einträge des Vokabulars geändert, kann die Sicherheitsbewertung völlig anders ausfallen. Daraus folgt jedoch auch, daß besser auf Verwechslungsprobleme bei der Erkennung eingegangen werden kann. Die anderen Konfidenzmaße sind Wörterbuchunabhängig. Experimente und Ergebnisse zu den in diesem Kapitel erläuterten Konfidenzmaßen sind im Kap. 9.2.2 anhand der on-line Handschrifterkennung und im Kap. 9.3.2 anhand der Adreßerkennung beschrieben.

Kapitel 8

Adaptionsverfahren

Adaptionsverfahren bilden eine weitere Möglichkeit, die Erkennungsleistung für bestimmte Anwendungen zu steigern. Das Ziel besteht darin, bestehende Modelle (HMMs oder repräsentative Prototypen einer Klasse, etc.) durch Adaption an neue, spezifische Daten (Schrift oder Sprache) anzupassen. Notwendig für eine Adaption sind, wie beim Training, die Merkmale der Daten und die entsprechenden Label, wobei hier ggf. auch die Zuordnung der einzelnen Zeichen-Modelle zu den Merkmalvektoren gegeben sein muß.
In der Regel handelt es sich dabei um eine Adaption eines allgemeinen (schreiber- oder sprecherunabhängigen) Erkennungssystems auf einen bestimmten Schreiber (bzw. Font) oder Sprecher (siehe Kap. 9.2.2). Prinzipiell sind aber auch Adaptionen beispielsweise auf spezielle Schreibweisen (mehrere Schreiber, vgl. Experimente zur Postamt-Adaption in Kap. 9.3.2) möglich. In der automatischen Spracherkennung ist neben der reinen Sprecher-Adaption entsprechend eine Adaption auf bestimmte Dialekte oder beispielsweise auf bestimmte Radio-Nachrichtensendungen möglich. Durch eine Adaption verbessert sich in der Regel die Erkennungsrate für die neuen Daten, die ursprünglichen Basisdaten werden jedoch schlechter erkannt.
Für Adaptionsverfahren in der Schrifterkennung lassen sich verschiedene Kategorien definieren:

- Art der Klassifikatoren bzw. der Struktur des Erkennungssystems: Adaption von Prototypen oder Hidden Markov Modellen
- Anwendungsbereich: Adaption auf einen speziellen Schreiber/ Font oder auf Schreibweisen bestimmter festzulegender Gruppen
- Modus: überwachte oder unüberwachte Adaption
- Adaptionstechnik: ML, MLLR, SLLR, MAP, etc.

Unabhängig von der Art der Klassifikatoren lassen sich Adaptionsverfahren einsetzen. Adaptionstechniken für isolierte handschriftliche Einzelzeichen (Prototypen-basierte Er-

kennung) werden z.B. in [Mat93, Con99, Vuo01] beschrieben, sie sollen in dieser Arbeit aber nicht weiter betrachtet werden. Das Thema hier ist die Adaption von Erkennungssystemen basierend auf Hidden Markov Modellen, wozu es nur wenige Literaturbeiträge gibt. In [Nat99, Lu99] werden Ergebnisse zweier Techniken für die Font-Adaption qualitativ schlechter Dokumente vorgestellt. Erste Ansätze mit dem MLLR-Verfahren für die on-line Handschrifterkennung zeigt der Beitrag [Sen97], weiterführende und vergleichende Untersuchungen wurden größtenteils im Rahmen dieser Arbeit durchgeführt (vgl. [Bra01b, Bra01d, Bra01a, Bra02a, Vin02]).

Die Anwendungsbereiche, die sich in dieser Kategorie der HMM-basierten Systeme ergeben, werden im nächsten Kapitel 8.1 detaillierter vorgestellt. Die Entscheidung zur überwachten oder unüberwachten Adaption ist eng mit der Verwendung von Konfidenzmaßen verknüpft (siehe auch Kap. 7). Im Anschluß werden vier untersuchte Techniken näher erläutert. Diese sind das Nachtrainieren nach dem Maximum Likelihood (ML) Verfahren (Kap. 8.2, siehe [Rab86]), die Adaption nach dem Maximum Likelihood Linear Regression (MLLR) Verfahren (Kap. 8.3, vgl. [Leg94]), die Adaption nach dem Scaled Likelihood Linear Regression (SLLR) Verfahren (Kap. 8.4, vgl. auch [Wal00]) und die Adaption nach dem Maximum A Posteriori (MAP) Verfahren (Kap. 8.5, vgl. [Gau94]). Andere in der Literatur für die Spracherkennung beschriebene Verfahren sind häufig eine Kombination aus den hier vorgestellten Techniken (z.B. Maximum A Posteriori Linear Regression (MAPLR) von [Che99]). Diese Adaptionsverfahren arbeiten nach einem gemeinsamen Schema: eine optimale Anpassung der HMM-Parameter unter Verwendung weniger Daten. Dies kann entweder durch eine Einschränkung der Parameteranzahl (ML: Adaption bestimmter Parameter; MLLR/SLLR: Adaption von Gruppen bzw. Clustern) oder durch das Wissen der a priori Wahrscheinlichkeit (MAP) erfolgen.

Alle diese Verfahren haben eine Anpassung der Modelle auf bestimmte Eigenheiten in der Schreibweise zum Ziel. Eine Normierung auf Merkmalebene (vgl. Kap. 2.1) hat im Gegensatz dazu das Ziel, möglichst viele spezielle Eigenheiten zu vereinheitlichen. Der Einfluß dieser beiden Methoden – Adaption und Vorverarbeitung –, auf die Erkennungsleistung wird ebenfalls bei den hier durchgeführten Versuchen untersucht (Kap. 9.2.2, vgl. auch [Bra02a]).

8.1 Problematiken und Anwendungsbereiche

Warum werden überhaupt Adaptionsverfahren eingesetzt und kein ‘normal’ trainiertes Erkennungssystem? Dafür sind zwei Gründe ausschlaggebend: zum einen spielt die benötigte Datenmenge eine große Rolle und damit verbunden auch die zum Training benötigte Zeit; zum anderen besteht bei Adaptionsverfahren die Möglichkeit einer unüberwachten Anpassung der Modelle.

Generell ist die Datenmenge, die für eine Adaption benötigt wird, deutlich geringer als die notwendige Trainingsmenge für ein völlig neues System. Bei der Adaption können bereits bestehende Modelle genutzt werden, sodaß auch bei wenigen Daten die Parameter robust geschätzt werden können. Damit ergibt sich gleich die Voraussetzung für eine Adaption: die neuen Daten müssen prinzipiell bzgl. ihrer Struktur den alten Daten entsprechen. Ein Nebeneffekt ist die kürzere Zeitdauer, die sich bedingt durch die kleinere Datenmenge ergibt. Der zweite Grund, eine Adaption im unüberwachten Modus zu ermöglichen, bringt weitere Zeit- und Kosten-Vorteile. Unüberwacht bedeutet, daß die Wort-Label, die für die Adaption notwendig sind, nicht manuell sondern automatisch mit dem Basissystem erzeugt werden. Im überwachten Modus werden, wenn nötig (abhängig von der Adaptionstechnik), nur die Zeichengrenzen eines vorgegebenen Wortes mit dem Basissystem ermittelt. Aus diesen Vorteilen heraus ergeben sich mehrere Anwendungsbereiche.

Die Standard-Anwendung von Adaptionsverfahren ist sicherlich die Anpassung eines bestehenden Modells an einen bestimmten Schreiber (z.B. in [Sen97, Bra01b, Bra01a, Bra02a, Vin02]). Diese Schreiber-Adaption eines schreiberunabhängigen Handschrifterkennungssystems ist in erster Linie in der on-line Erkennung (z.B. bei PDAs) sinnvoll. Handschrifterkennungssoftware wird für elektronische Notizbücher im schreiberunabhängigen Modus erstellt. Obwohl die Erkennungsraten in den letzten Jahren ständig zugenommen haben, ist eine benutzerfreundliche sichere Erkennung von fließend geschriebenen Worten heute noch nicht möglich. Hier bieten Schreiber-Adaptionsverfahren die Möglichkeit, die Erkennungsrate – ein schreiberabhängiges System ist i.a. stets besser als ein schreiberunabhängiges – und somit auch die Benutzerakzeptanz zu steigern. Und gerade PDAs werden in der Regel nur von einer Person benutzt, sodaß die Adaption wirkungsvoll ist und eine Steigerung der Fehlerrate für andere fremde Personen kein Problem darstellt. In diesem Szenario ist sowohl eine überwachte als auch eine unüberwachte Adaption denkbar. Die Effizienz einer überwachten Adaption ist höher, da alle Worte korrekt gelabelt sind. Das bedeutet, daß weniger aber dafür manuell gelabelte Adaptionsbeispiele vom Benutzer eingegeben werden müssen. Bei einer unüberwachten Adaption kann sich das System quasi während der Benutzung ständig weiter adaptieren, ohne daß diese Anpassung für den Schreiber mit Arbeit verbunden ist. Die zur Verfügung stehende Adaptionsdatenmenge ist somit gewissermaßen unendlich, allerdings setzt dieses Verfahren eine bestimmte Basis-Erkennungsrate voraus. Sind die Erkennungsergebnisse des schreiberunabhängigen Systems zu schlecht, driftet das adaptierte System in die ‘falsche’ Richtung. Eine Möglichkeit, den Anteil der korrekten Label in der Adaptionsmenge zu erhöhen, ist die Reduzierung der verwendeten Daten mittels Konfidenzmaßen.

Eine weitere Anwendung, die eng mit der Schreiber-Adaption verknüpft ist, ist die Font-Adaption im off-line Bereich. Heutige OCR-Software ist multi-Font fähig, da Dokumente und Briefe in vielen verschiedenen Fonts gedruckt werden. Diese Forderung überträgt sich auch auf Dokumente geringer Qualität (eingescannte und gefaxte Textseiten, verbun-

dene oder getrennte Zeichen, siehe Kap. 3.3: SEDAL-Datenbasis), die mit einem HMM-basierten System erkannt werden sollen. Hier könnte, wie bei der Schreiber-Adaption, ein Font-unabhängiges (multi-Font) OCR-System entweder mit Hilfe weniger Zeilen eines Dokumentes auf eben dieses adaptiert werden, oder aber mit Hilfe eines vollständigen Beispieldokuments angepaßt werden, falls sich die folgenden Dokumente im Typ nicht unterscheiden (vgl. z.B. [Nat99, Lu99]).

Als dritter Anwendungsbereich sind Adaptionen auf die Schreibweisen bestimmter Gruppen (mehrere Schreiber) zu nennen. Einsatzmöglichkeiten ergeben sich hier in erster Linie im off-line Bereich der Erkennung von schreiberunabhängiger Handschrift (z.B. handgeschriebene Adressen, vgl. Kap. 3.2.2). Adreß-Erkennungssysteme, die in verschiedenen Postämtern eingesetzt werden, können auf lokale Änderungen in der Schreibweise adaptiert werden (z.B. [Bra01d]). Einsatzgebiete wären hier z.B. auch länderspezifische Schreibweisen, also die Anpassung von Einzelzeichen, die bei der gleichen Schriftart im Durchschnitt in abgewandelter Form auftreten (Vergleich amerikanisch - deutsch: z.B. hat die '1' im amerikanischen häufig einen 'Standfuß'). Aber auch innerhalb einer Region kann sich die Schreibweise mit der Zeit (über Generationen) ändern, abhängig vom Stil, der in der Schule gelehrt wird (z.B. vereinfachte Ausgangsschrift).

8.2 Maximum Likelihood (ML)

Die Adaption – oder besser gesagt das Nachtrainieren – nach dem Maximum Likelihood Kriterium [Rab86, You00] basiert, wie das normale Training von HMMs auf dem EM-Verfahren (vgl. Kapitel 2.2.3). Das Ziel besteht darin, die Parameter der HMMs λ, die bereits auf einer Basisdatenmenge (schreiberunabhängige Schriftproben, allgemeine Dokumente bzw. Fonts, etc.) trainiert wurden, so an die neuen (z.B. schreiberabhängige Daten) Merkmalvektoren X anzupassen, daß die Likelihood (das Ähnlichkeitsmaß oder auch die Plausibilität) $P(X|\lambda)$ maximiert wird:

$$\lambda_{ML} = \underset{\lambda}{\text{argmax}}\ P(X|\lambda) = \underset{\lambda}{\text{argmax}}\ P(X|W,\lambda) \quad \Rightarrow \quad \begin{cases} \underline{\mu}_{wi} \overset{EM}{\rightarrow} \underline{\mu}_{ML} \\ \Sigma_{wi} \overset{EM}{\rightarrow} \Sigma_{ML} \\ \omega_{wi} \overset{EM}{\rightarrow} \omega_{ML} \\ A_{wi} \overset{EM}{\rightarrow} A_{ML} \end{cases} \tag{8.1}$$

Der Baum-Welch Algorithmus wird auch hier zur Lösung eingesetzt. Es werden, wie beim Training, jeweils die HMMs λ des zugehörigen Trainings- bzw. Adaptionswortes W angepaßt. Dieses Adaptionsverfahren kann wahlweise alle oder nur einige der HMM-Parameter beeinflussen, die sich je nach Modellierungstechnik in der Regel aus den Gauß-Funktionen (Mittelwertvektor $\underline{\mu}$, Kovarianzmatrix Σ), den Gewichten ω und der Transitionsmatrix A zusammensetzen. Der Index wi in Gl. 8.1 steht hier für das schreiberunabhängige (writer inde-

pendent) System, wobei die Verfahren selbstverständlich, wie oben erörtert, auch für Fonts, etc. gelten. Je nach verfügbarer Adaptionsdatenmenge kann die Anzahl der zu adaptierenden Parameter eingeschränkt werden. Häufig werden lediglich die Mittelwerte und/oder Kovarianzen der Gaußverteilungen angepaßt, wie es auch bei den expliziten Adaptionsverfahren MLLR, SLLR und MAP üblich ist.

8.3 Maximum Likelihood Linear Regression (MLLR)

Bei dem Maximum Likelihood Linear Regression Adaptionsverfahren [Leg94] wird das Problem der geringen Adaptionsmenge durch Clustern von Modellen bzw. Gaußverteilungen und einer anschließenden gemeinsamen Transformation mit einer Regressionsmatrix M zur Anpassung gelöst. Das MLLR-Verfahren wurde ursprünglich von Leggetter [Leg94] für die Spracherkennung eingeführt. Wie beim ML-Verfahren soll auch diese Methode anhand kontinuierlicher HMMs erläutert werden, das Prinzip kann aber auf die semi-kontinuierliche Modellierungstechnik übertragen werden. Für diskrete HMM-Strukturen ist dieses Verfahren nicht direkt übertragbar, wie sich aus dem Algorithmus ersehen läßt. In [Rot00a] sind ergänzende Adaptionsverfahren auch für diskrete HMM-Strukturen erklärt.

Die Zielfunktion ist wie beim ML-Verfahren, die Maximierung der Likelihood $P(X|\lambda)$ durch Anpassung der HMM-Parameter. Standardmäßig ist damit gemeint, daß zum Training des Modells λ nur die passenden Beispiele W gewählt werden. Lediglich zur Unterscheidung von MLLR- und SLLR-Verfahren, wird hier explizit von $P(X|W,\lambda)$ gesprochen. Allerdings wird hier die Anpassung durch eine Regressionsmatrix M durchgeführt, mit der die ursprünglichen HMM-Parameter wie Mittelwertvektor $\underline{\mu}$ und Kovarianzmatrix Σ multipliziert werden:

$$\lambda_{MLLR} = \underset{\lambda}{\operatorname{argmax}}\, P(X|W,\lambda) = \underset{M}{\operatorname{argmax}}\, P(X|W,\lambda,M) \tag{8.2}$$

Die folgenden Betrachtungen beziehen sich auf die Adaption des Mittelwertvektors $\underline{\mu}$. Für die Berechnungen wird ein erweiterter Mittelwertvektor $\underline{\hat{\mu}}$ eingeführt:

$$\underline{\hat{\mu}}_{wi} = \begin{pmatrix} v \\ \underline{\mu}_{wi} \end{pmatrix} \tag{8.3}$$

Der Offset v ist notwendig, um über die Matrixmultiplikation auch eine Verschiebung des Vektors erzielen zu können:

$$\underline{\mu}_{MLLR} = M_{MLLR} \cdot \underline{\hat{\mu}}_{wi} \tag{8.4}$$

Die Matrix M hat die Dimension $(D+1)\times D$. Abhängig von der Clusterung der Gaußdichten auf die im nächsten Abschnitt noch näher eingegangen werden soll, wird jeder Mittelwert einzeln oder alle Mittelwerte eines Clusters mit derselben Matrix M multipliziert. Das heißt,

für jedes Cluster wird eine Matrix M ermittelt. Sind alle Gaußfunktionen in einem Cluster vereint, spricht man von einer globalen Adaption mit nur einer Matrix M_{global}.

Das Problem ist die Bestimmung dieser Matrix, die die MLLR-Zielfunktion maximieren soll. Zur Lösung werden zwei Alternativen aufgezeigt: die Standard-Lösung über den Baum-Welch Algorithmus oder eine Variante, die auf einem Gradientenabstieg beruht. Hier soll zunächst die Herleitung der Baum-Welch Gleichungen in Hinblick auf die MLLR-Adaption kurz erläutert werden. Eine ausführliche Beschreibung wird in [Leg94] angegeben.
Betrachtet man Gl. 8.4 für jeden Zustand s, so ergibt sich:

$$\underline{\mu}_{MLLR,s} = M_s \cdot \underline{\hat{\mu}}_s \tag{8.5}$$

Eingesetzt ergibt sich die folgende Gleichung 8.6 für Gaußsche Normalverteilungen g:

$$g(\underline{x}) = \frac{1}{\sqrt{(2\pi)^D \cdot |\Sigma|}} \cdot e^{-\frac{1}{2}(\underline{x}-M_s\cdot\underline{\hat{\mu}}_s)^T\Sigma^{-1}(\underline{x}-M_s\cdot\underline{\hat{\mu}}_s)} \tag{8.6}$$

Im weiteren wird formal von nur einer Gaußverteilungsdichte je Zustand ausgegangen, eine Erweiterung auf Gaußsche Mischverteilungen kann nach dem gleichen Prinzip erfolgen.

Die Herleitung des Baum-Welch Verfahrens nach der EM-Methode basiert auf der Definition der ML-Zielfunktion (vgl. Gl. 2.12) und einer Hilfsfunktion Q (Kullback-Leibler-Statistik, λ^* sind die neuen Modellparameter):

$$P(X|\lambda) = \sum_q P(X,q|\lambda) \quad \text{und} \quad Q(\lambda,\lambda^*) = \sum_q P(X,q|\lambda)\log P(X,q|\lambda^*) \tag{8.7}$$

Wenn $Q(\lambda,\lambda^*)$ maximiert wird, wird auch $P(X|\lambda^*)$ maximiert. Die Maximierung der Parameter erfolgt durch die Ableitung von Q:

$$\frac{dQ(\lambda,\lambda^*)}{dM_s} = 0 \tag{8.8}$$

Daraus folgt Gl. 8.9:

$$\sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}\underline{x}_t\underline{\hat{\mu}}_s^T = \sum_{t=1}^{T}\gamma_s(t)\Sigma_s^{-1}M_s\underline{\hat{\mu}}_s\underline{\hat{\mu}}_s^T \tag{8.9}$$

mit (vgl. Gl. 2.28):

$$\gamma_s(t) = \frac{1}{P(X|\lambda)}\sum_q P(X,q_t=s|\lambda) \tag{8.10}$$

Für den Spezialfall, daß jede Gaußverteilung mit einer eigenen Matrix adaptiert wird (vgl. Gl. 2.32), folgt dann:

$$\underline{\mu}_s = \frac{\sum_t \gamma_s(t)\underline{x}_t}{\sum_t \gamma_s(t)} \tag{8.11}$$

Dies ist für die kontinuierliche Modellierungstechnik die Standardformel zum Training des Mittelwertes nach dem Baum-Welch Verfahren.

Erst wenn die Matrix M_s für mehrere Gaußverteilungen benutzt wird, ergibt sich das typische MLLR-Verfahren. Wenn R Gaußverteilungen der Zustände s_r geclustert sind, ergibt sich Gl. 8.12:

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}\underline{x}_t\underline{\hat{\mu}}_{s_r}^{T} = \sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\Sigma_{s_r}^{-1}M_s\underline{\hat{\mu}}_{s_r}\underline{\hat{\mu}}_{s_r}^{T} \tag{8.12}$$

Diese Gleichung kann, wie in [Leg94] gezeigt wird, mit Hilfe eines linearen Gleichungssystems nach M_s aufgelöst werden. Abhängig von der Wahl der Matrix M und des Offsets v kann die zu schätzende Parameterzahl weiterhin eingeschränkt werden. Wird v zu Null gesetzt (oder die erste Spalte von M), wird eine Transformation der Mittelwertvektoren $\underline{\mu}$ ohne Verschiebung durchgeführt. Eine weitere Möglichkeit besteht darin, eine Diagonalmatrix M zu wählen (mit oder ohne Verschiebungsvektor). Bei einer vollbesetzten Matrix kann eine Verschiebung, Rotation und Skalierung der ursprünglichen Mittelwerte erzielt werden. Auf analoge Weise (siehe [Gal96]) kann auch die Transformationsmatrix für die Varianzen ermittelt werden.

Bildung der Cluster

Die Voraussetzung für eine gemeinsame Transformation von mehreren Modellen bzw. Gaußverteilungen ist, daß sich diese Modelle bei verschiedenen Schreibern auch prinzipiell ähnlich verhalten. Leggetter [Leg94] nimmt für die Spracherkennung an, daß folgendes für den akustischen Merkmalraum gilt: Sind für Sprecher A zwei akustische Klassen im Merkmalraum benachbart, müssen diese Klassen auch für einen anderen Sprecher B benachbart sein bzw. in der gleichen 'Merkmalregion' liegen. Wobei diese Regionen von Sprecher A und B nicht übereinstimmen müssen. Die Experimente zeigen jedoch, daß eine solche Annahme für die Schrift weit weniger zutrifft, als bei der Sprache.

Die Clusterbildung erfolgt nach zwei unterschiedlichen Ansätzen. Als kleinste Einheit, die geclustert bzw. zusammengefaßt werden kann, sind die einzelnen Gaußverteilungsdichten g_{ij} jedes Zustandes zu nennen (vgl. auch [Gal96]). Alternativ können auch jeweils nur ganze Modelle – inklusive aller Zustände mit allen Gaußfunktionen – mit anderen geclustert werden. Wird keine Gaußfunktion mit einer anderen zusammen gefaßt, ergibt sich das normale ML-Training.

Es wurden im Rahmen der Arbeit zwei unterschiedliche Clustermethoden verwendet: ein Top-Down Verfahren basierend auf der Clusterung einzelner Gaußverteilungsdichten und ein k-means Verfahren basierend auf der Clusterung ganzer Modelle.
Wenn zur Clusterung das Top-Down Verfahren verwendet wird, wird ein Binärbaum ('binary tree') gebildet. Ausgehend von einem Knoten, dem alle Gaußdichten zugeordnet sind, wird der Baum aufgespannt, indem die verschiedenen Gaußdichten anhand eines euklidischen Abstandsmaßes (bezogen auf den Mittelwertvektor der Gaußverteilung) auf verschiedene Knoten aufgeteilt werden (vgl. auch Abb. 8.1). Die Anzahl der endgültigen Knoten, bzw. die

Anzahl der Cluster muß vom Benutzer festgelegt werden und sollte sich nach der Menge der zur Verfügung stehenden Adaptionsdaten richten. Dabei ist jedoch zu beachten, daß die Clusterung nur auf den Ähnlichkeiten der allgemeinen schreiberunabhängigen Modellen beruht und nicht die Adaptionsdaten berücksichtigt. Sind genügend (vom Benutzer zu definierende Mindest-Aufenthaltswahrscheinlichkeit der Adaptionsmerkmalvektoren) Adaptionsdaten je Cluster vorhanden, wird je eine Regressionsmatrix M_r je Cluster ermittelt. Bei zu wenigen Adaptionsbeispielen eines Clusters wird im Baum der nächst 'höhere' Knoten ermittelt und die Regressionsmatrix M_r bezieht sich auf alle Gaußverteilungsdichten die zu diesem Knoten gehören. Das heißt, hier wird die Anzahl der Daten, die je Cluster zur Verfügung stehen, und insbesondere die Aufenthaltswahrscheinlichkeit der Merkmalvektoren in einem Zustand berücksichtigt. Daraus ergibt sich ein wesentlicher Vorteil der MLLR-Adaption: auch Modelle (Zeichen), die in der Adaptionsmenge nicht vorkommen, werden automatisch einem Cluster zugeordnet, der adaptiert wird.

Der Vorteil dieses Top-Down Verfahrens ist im wesentlichen programmtechnischer Art. Die Cluster (maximale Anzahl) können unabhängig von den Adaptionsdaten ermittelt werden. Wenn nun auf unterschiedliche Schreiber mit ggf. unterschiedlichen Datenmengen adaptiert werden soll, kann die gleiche Grundstruktur der Cluster verwendet werden. Je nach Adaptionsdatenmenge kann die Clusteranzahl eingeschränkt werden, indem auf 'höhere' Knoten im Baum zurückgegriffen wird. Andererseits werden bei diesem Verfahren Cluster, die sich nahe des globalen Mittelwertes aller Modelle befinden, in der Regel in unterschiedliche Cluster aufgeteilt, da der Baum jeweils von der Mitte verzweigt. Anhand des vereinfachten Beispiels in Abb. 8.1 bedeutet dies, daß bei einer Schwelle von mindestens 10 Beispielen je Cluster, für die Cluster A, C und D eigene Regressionsmatrizen ermittelt werden (laut Beispiel existieren für diese Cluster 10 bzw. 20 Beispiele). Das Cluster B (hier nur 2 Beispiele) wird mit einer Regressionsmatrix transformiert, die auf dem Cluster AB beruht. Die Lage (x-Richtung) der Cluster im Bild soll deren Mittelwerte beschreiben.

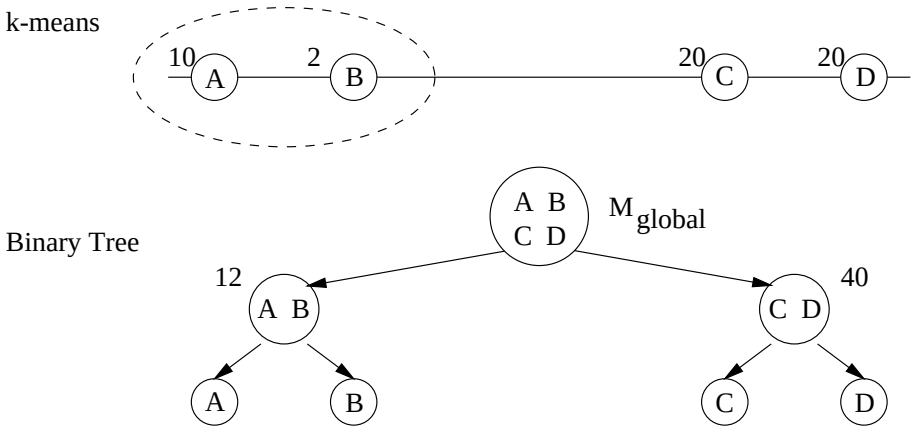


Abbildung 8.1: Clusterbildung zur MLLR-Adaption: k-means oder Binärbaum

Als zweite Variante wurden mit einem k-means Verfahren jeweils alle Gaußverteilungsdichten eines Modells mit anderen Modellen zusammengefaßt, bis die gewünschte Anzahl von Clustern unter Berücksichtigung der Mindest-Aufenthaltswahrscheinlichkeit entstanden ist (Bottom-Up). Auch hier basiert die Ähnlichkeit von Clustern, die zusammengefaßt werden sollen, auf den Mittelwertvektoren (gemittelt über alle zugehörigen Gaußfunktionen) der HMMs. Wesentlicher Unterschied zum ersten Verfahren ist, daß hier alle Gauß-Funktionen aller Zustände eines Modells auf jeden Fall mit der gleichen Regressionsmatrix transformiert werden. Ein weiterer Unterschied in der Clustermethode wird in Abb. 8.1 deutlich. Hier wird das Cluster A und B mit der gleichen Regressionsmatrix transformiert.

Alternative zum EM-Verfahren: Gradientenabstieg

Eine alternative Möglichkeit zur Berechnung der Elemente der Regressionsmatrix M setzt auf Gl. 8.2 und Gl. 8.6 auf. Hier wird mittels des RPROP-Verfahrens (Resilient Propagation) durch einen Gradientenabstieg der Ausdruck $P(X|W,\lambda)$ maximiert (siehe [Wal00]).

Vorausgesetzt wird dabei die Zustandsfolge Q, die sich mit den Beobachtungssequenzen bzw. Merkmalvektoren $X = \underline{x}_1, ..., \underline{x}_T$ und einem Alignment (Zuordnung von Vektoren und HMM-Zuständen) der bekannten Wortsequenzen W auf dem Basissystem ergibt. Damit ergibt sich der folgende Ausdruck:

$$P(X|W,\lambda) \approx P(X|Q,\lambda) \approx \prod_t p(\underline{x}_t|q_t = s_i, \lambda) \tag{8.13}$$

L_{MLLR} wird als logarithmierter Ausdruck dieser Likelihood definiert, die anhand der Ausgabeverteilungsdichte b_i im Zustand $q_t = s_i$ so ausgedrückt werden kann:

$$L_{MLLR} = \log \prod_t p(\underline{x}_t|q_t,\lambda) = \log \prod_t b_i(\underline{x}_t) = \sum_t \log b_i(\underline{x}_t) \tag{8.14}$$

Für kontinuierliche HMMs mit J Gaußschen Mischverteilungen und einer diagonal besetzten Kovarianzmatrix ($\Sigma = diag(\sigma_d) : d = 1, ..., D$), kann die Ausgabeverteilungsdichte b folgendermaßen unter Berücksichtigung der Regressionsmatrix M dargestellt werden ($\underline{x}_t$ besteht aus den Elementen $x_{t1}, ..., x_{tD}$):

$$b_i(\underline{x}_t) = \sum_{j=1}^{J} \omega_{ij} g_{ij}(\underline{x}_t) = \sum_{j=1}^{J} \frac{\omega_{ij}}{\sqrt{(2\pi)^D \cdot \prod_d \sigma_{ijd}}} \cdot e^{-\frac{1}{2}\sum_d \frac{(x_{td} - (M\hat{\underline{\mu}}_{ij})_d)^2}{\sigma_{ijd}}} \tag{8.15}$$

Die Elemente der Regressionsmatrix M können nun mit Hilfe des RPROP-Verfahrens (vgl. auch Kap. 4.1) ermittelt werden, indem die partielle Ableitung nach allen Matrixelementen m_{rs} gebildet wird:

$$\frac{\partial L_{MLLR}(m)}{\partial m_{rs}} = \sum_t \frac{\partial \log b_i(\underline{x}_t)}{\partial m_{rs}} = \sum_t \frac{\partial \log p(\underline{x}_t|q_t)}{\partial m_{rs}} \tag{8.16}$$

RPROP

Das RPROP (resilient propagation) Verfahren ist ein Lernverfahren, welches anhand des Vorzeichens des Gradienten $\partial E(w)/\partial w$ einer Funktion $E(w)$ die Schrittweite Δw ändert und somit lokale Maxima dieser Funktion findet (siehe z.B. [Zel97]). Übertragen auf die Aufgabenstellung der MLLR-Adaption ist die Funktion $E(w)$ mit $L(m)$ gleichzusetzen. Das Verfahren berücksichtigt die Tatsache, daß die Ableitung einer Funktion bei $w > w_{max}$ negativ und bei $w < w_{max}$ positiv sein muß. So wird w iterativ an den Parameter des Maximums $E_{max} = E(w_{max})$ angepaßt, wobei der Betrag der Ableitung für die Wahl der Schrittweite keine Rolle spielt.

8.4 Scaled Likelihood Linear Regression (SLLR)

Das Scaled Likelihood Linear Regression (SLLR) Adaptionsverfahren ist gewissermaßen eine diskriminative MLLR-Adaption (vgl. z.B. [Wal00, Val96]). Die Zielfunktion ist die Maximierung der Likelihood $P(X|W,\lambda)$ für die zur korrekten Wortsequenz W gehörenden Modelle λ und gleichzeitiger Minimierung von $P(X|\lambda)$ durch Anpassung der HMM-Parameter:

$$\lambda_{SLLR} = \underset{\lambda}{\text{argmax}} \frac{P(X|W,\lambda)}{P(X|\lambda)} \qquad \text{mit:} \quad P(X|\lambda) \approx \sum_{\text{besten W}} P(X|W,\lambda)P(W) \tag{8.17}$$

Im Unterschied zur MLLR-Adaption muß auch der Term $P(X|\lambda)$ geschätzt werden. Dies könnte, wie in Gl. 8.17 angedeutet, durch die Auswertung von N-Best Listen erfolgen (vgl. auch Kap. 7.3 zur Bestimmung von Konfidenzmaßen). Da die Adaptionsmenge und somit auch die Anzahl der Verwechslungen zum diskriminativen Training eher gering ist, wird hier ein frame-basierter Ansatz verfolgt (siehe [Wal00]):

$$P(X|\lambda) \approx \prod_t p(\underline{x}_t) \quad \text{mit:} \quad p(\underline{x}_t) \approx \sum_q p(q_t)p(\underline{x}_t|q_t) \tag{8.18}$$

Die Zustandsfolge Q wurde aus dem Viterbi-Pfad entnommen. $p(\underline{x}_t|q_t)$ kann wiederum durch die Ausgabeverteilungsdichte b, die von M_{SLLR} abhängt, dargestellt werden.

Auch hier wird, wie beim MLLR-Verfahren, die Adaption der Mittelwerte $\underline{\mu}_{wi}$ der Gaußfunktionen durchgeführt, indem diese mit einer Regressionsmatrix M_{SLLR} transformiert werden.

$$\underline{\mu}_{SLLR} = M_{SLLR} \cdot \hat{\underline{\mu}}_{wi} \tag{8.19}$$

Zusammenfassend ergibt sich dann:

$$\lambda_{SLLR} = \underset{\lambda}{\text{argmax}} \prod_t \frac{p(\underline{x}_t|q_t,\lambda)}{\sum_q p(q_t|\lambda)p(\underline{x}_t|q_t,\lambda)} \tag{8.20}$$

Definiert man L_{SLLR} als logarithmierte Likelihood, so ergibt sich die mit dem RPROP-Verfahren zu lösende Gleichung zu (siehe [Wal99]):

$$\frac{\partial L_{SLLR}(m)}{\partial m_{rs}} = \sum_t \left(\frac{\partial \log p(\underline{x}_t|q_t)}{\partial m_{rs}} - \frac{\partial \log \left(\sum_q p(q_t) p(\underline{x}_t|q_t) \right)}{\partial m_{rs}} \right) \tag{8.21}$$

Betrachtet man Gl. 8.21, wird deutlich, daß dieses Verfahren aufgrund des rechten Terms in der Summe sehr viel rechenzeitaufwendiger ist als das MLLR-Verfahren (vgl. Gl. 8.16).

Die SLLR-Adaption wird in der Regel mit einer globalen Regressionsmatrix durchgeführt. Problematisch wird das Verfahren, wenn nicht alle Zeichen in den Adaptionsdaten vorkommen. Die Modelle der nicht vorkommenden Zeichen werden zwar mit anderen ähnlichen Modellen geclustert, können aber nicht von anderen Zeichen diskriminativ abgespalten werden, da keine Daten $p(\underline{x}|q)$ für das richtige Modell λ zur Verfügung stehen.

8.5 Maximum A Posteriori (MAP)

Das Maximum A Posteriori (MAP) Verfahren (siehe [Dud73, Gau91, Gau94]) handhabt das Problem der geringen Adaptionsdatenmengen nicht durch Zusammenfassung von Modellen, sondern durch Berücksichtigung der a priori Wahrscheinlichkeit. Diese Adaptionstechnik wird häufig auch als Bayes-Schätzung bezeichnet. Mit Hilfe der MAP-Adaption werden hier die Mittelwertvektoren und Kovarianzmatrizen der kontinuierlichen HMMs angepaßt.

Im Gegensatz zur ML-Adaption wird hier die a posteriori Wahrscheinlichkeit maximiert. Dazu ergibt sich nach Bayes Formel der folgende Zusammenhang, wie in Gl. 8.22 dargestellt. Die a priori Wahrscheinlichkeit $P(\lambda)$ der Modelle bzw. Zeichen wird anhand der Basistrainingsdaten bestimmt.

$$\lambda_{MAP} = \underset{\lambda}{\text{argmax}}\ P(\lambda|X) \approx \underset{\lambda}{\text{argmax}}\ P(X|\lambda)P(\lambda) \tag{8.22}$$

Wird die a priori Wahrscheinlichkeit $P(\lambda)$ als gleichverteilt angenommen, ergibt sich die in Kap. 8.2 beschriebene ML-Schätzung. Für eine analytische Optimierung der MAP-Zielfunktion werden die a priori Wahrscheinlichkeiten $P(\lambda_k)$ der einzelnen Klassen einer Dirichlet-Verteilung angepaßt (*konjugierte Familien*). Somit folgen Gl. 8.23 und Gl. 8.24, wie in [Dud73, Gau91] gezeigt wird:

$$\underline{\mu}_{MAP_{ij}} = f_L \cdot \underline{\mu}_{wd_{ij}} + (1 - f_L) \cdot \underline{\mu}_{wi_{ij}} \qquad \text{mit:} \quad f_L = \frac{\sum_t l_{ij}(t)}{(\sum_t l_{ij}(t)) + \tau} \tag{8.23}$$

Für diesen Fall kann die Schätzung nach Gl. 8.22 wiederum mit dem EM-Algorithmus erfolgen. Diese Schätzung der neuen Parameter kann iterativ wiederholt werden. $\underline{\mu}_{wi_{ij}}$ beschreibt

den Mittelwert der ursprünglichen (schreiberunabhängigen) Daten, $\underline{\mu}_{wd_{ij}}$ beschreibt entsprechend den Mittelwert der neuen beobachteten Datenmenge. Analog gilt die Gleichung 8.24 ebenfalls für das ML-Training:

$$\underline{\mu}_{wd_{ij}} = \frac{\sum_t l_{ij}(t) \cdot \underline{x}_t}{\sum_t l_{ij}(t)} \tag{8.24}$$

Dabei steht $l_{ij}(t)$ für die Aufenthaltswahrscheinlichkeit im Zustand $q_t = s_i$ bei der Gauß-Funktion m_j und kann mit Hilfe des Vorwärts-Rückwärts-Algorithmus (Wahrscheinlichkeiten α_t und β_t) bestimmt werden:

$$l_{ij}(t) = P(q_t = s_i, m_j | X, \lambda) \tag{8.25}$$

Für den Fall einer multivariaten Gaußverteilungsdichte je Zustand s_i ergibt sich $l_{ij} = \gamma_i$ (siehe Gl. 8.10). Der Parameter τ in Gl. 8.23 bezieht sich auf die a priori Wahrscheinlichkeitsverteilung und wird empirisch gesetzt. Anhand dieses Parameters läßt sich der Einfluß der Adaptionsdaten regeln. Ist die Aufenthaltswahrscheinlichkeit in einer bestimmten Gaußfunktion sehr klein, so ist der Faktor f_L klein und somit ergibt die MAP-Schätzung einen Wert, der nahe dem schreiberunabhängigen System liegt. Für große Stichproben strebt die MAP-Schätzung gegen die ML-Schätzung.

Dies bedeutet, daß bei der MAP-Adaption im Gegensatz zur MLLR- oder SLLR- Adaption der Mittelwert jeder Gaußverteilung separat geschätzt werden muß (und kann), abhängig vom ursprünglichen Mittelwert, den eingestellten Gewichten und den Adaptionsdaten. Theoretisch folgt daraus – wie es sich in der Spracherkennung auch bestätigt hat – ein größerer Bedarf an Adaptionsdaten als beispielsweise bei der MLLR-Adaption, bei der Cluster gebildet werden. Die in Kap. 9 beschriebenen Ergebnisse zur Schreiber-Adaption zeigen jedoch, daß die zur erfolgreichen MAP-Adaption benötigte Datenmenge sehr gering ausfallen kann. Der Vorteil dieser Adaptionsmethode liegt gerade in der separaten Schätzung der Parameter. Hier kann, wenn die a priori-Verteilung aussagekräftig ist und die Datenmenge ausreicht, eine sehr spezifische Anpassung je Modell erfolgen. Damit ergibt sich gleichzeitig das Problem der Schätzung von Klassen (bzw. Zeichen), die nur sehr selten oder gar nicht in den Adaptionsdaten vorkommen. Diese haben eine sehr geringe Aufenthaltswahrscheinlichkeit und werden somit nicht (oder nur wenig) angepaßt.

Es läßt sich zeigen, daß auch die anderen Parameter (Kovarianzmatrix Σ, Transitionsmatrix A und Gewichte ω) durch eine Interpolation zwischen alten Modell-Parametern und neuen, nach dem ML-Verfahren geschätzten Werten bestimmt werden können. Diese Eigenschaft, daß die Modellparameter durch Interpolation in Abhängigkeit der Aufenthaltswahrscheinlichkeit verbessert werden können, wurde beim MAP-Verfahren aufgrund der a priori Wahrscheinlichkeitsbetrachtung ermittelt. In [Bau68] wurde jedoch auch nachgewiesen, daß die Baum-Welch-Gleichungen ebenfalls über die Annahme hergeleitet werden können, daß die Produktionswahrscheinlichkeit $P(O|\lambda)$ ein homogenes Polynom darstellt, woraus folgt,

daß auch alle interpolierenden Parameter zwischen altem und neuem Modell eine Verbesserung der ML-Zielgröße darstellen. Diese theoretische Betrachtung soll hier nicht weiter vertieft werden, das Ergebnis kann jedoch auf weitere Adaptionsexperimente – insbesondere die Adaption semi-kontinuierlichen HMMs – angewandt werden.

Bei der Adaption von semi-kontinuierlichen HMMs wurden nicht die Mittelwerte und/ oder Varianzen, sondern die Gewichte adaptiert. Das Codebuch bestehend aus dem Pool von Gaußverteilungsdichten wird konstant gehalten und nur die Gewichte ω_{ij} in den einzelnen Zuständen werden zwischen dem ursprünglichen und dem nach der ML-Methode geschätzten neuen Gewicht interpoliert.

Eine weitere Adaptionsvariante ist die Kombination aus MLLR und MAP, die die Vorteile beider Verfahren beinhaltet. Mit einer MLLR-Adaption werden zuerst die ermittelten Cluster transformiert (also alle Zeichen, auch die, die in den Daten selbst nicht vorkommen) und anschließend kann eine Art Feinjustierung mittels des MAP-Verfahrens erfolgen.

8.6 Kapitelzusammenfassung

Kap. 8 beschreibt Anwendungsgebiete und Voraussetzungen für Adaptionsverfahren, wie sie in dieser Arbeit verwendet werden, und vier unterschiedliche Adaptionstechniken: das ML-Nachtraining, die MLLR- und die SLLR-Adaption und die MAP-Adaption.

Ausgehend von einem HMM-Basissystem, welches für Schreiber- oder Font-unabhängige Anwendungen trainiert wurde, können verschiedene HMM-Parameter – im wesentlichen Mittelwertvektoren und Kovarianzmatrizen – auf spezielle (z.B. schreiberabhängige) Daten angepaßt werden. Das ML- und MLLR-Verfahren verfolgen die gleiche Zielfunktion, die Maximierung der Likelihood $P(X|W)$, wobei das MLLR-Verfahren das Problem der neuen Parameter-Schätzung bei wenig Adaptionstrainingsdaten durch eine Clusterung von ähnlichen Gaußdichtefunktionen angeht, die mit einer Regressionsmatrix transformiert werden. Ebenfalls auf der Basis von Regressionsmatrizen arbeitet das SLLR-Verfahren, welches eine diskriminative Variante der MLLR-Methode darstellt. Die MAP-Adaption hingegen benötigt keine Zusammenfassung von Modellen, hier wird die a priori Wahrscheinlichkeit der Zeichen-Modelle berücksichtigt. Grenzfälle der expliziten Adaptionsverfahren münden wieder im ML-Training, wobei je nach Adaptionsmenge nur bestimmte HMM-Parameter adaptiert werden. Wird für die MLLR-Adaption je ein Cluster je Gaußdichtefunktion gewählt, oder ist die Adaptionsmenge für das MAP-Verfahren sehr groß (bzw. wird der Einfluß der Adaptionsdaten hoch bewertet), ergibt sich die ML-Adaption.

Versuchsdurchführungen und Ergebnisse zur Adaption sind im Kap. 9.2.2 zur Erkennung von on-line Handschriftdaten und im Kap. 9.3.2 zur Erkennung von Adressen (off-line Handschrift) beschrieben.

Kapitel 9

Experimente und Ergebnisse

Dieses Kapitel beschreibt die Versuchsanordnungen und Ergebnisse, denen die zuvor beschriebenen Datenbasen zugrunde liegen. Im Kap. 9.1 werden für alle Experimente gültige Grundlagen und Definitionen beschrieben, bevor in den Kapiteln 9.2, 9.3 und 9.4 auf die Auswirkungen der unterschiedlichen Modellierungstechniken und Adaptionsverfahren eingegangen wird. Die Auswertung der Ergebnisse erfolgt in Kap. 9.5.

9.1 Systembeschreibung

Die grundlegenden Algorithmen (Baum-Welch, Viterbi) zum Training und Testen der HMM-basierten Erkennungssysteme basieren auf dem HTK-Toolkit [You00]. Abb. 9.1 veranschaulicht den prinzipiellen Aufbau der Erkennungssysteme und deren Unterschiede (die gestrichelten Pfeile stellen mögliche Alternativen dar).

Die Trainings- bzw. Testdaten werden mit den in Kap. 3 beschriebenen Verfahren entsprechend vorverarbeitet und die je Datenbasis spezifischen Merkmale werden extrahiert. Entweder bilden diese Merkmalvektoren direkt den zu modellierenden Signaleingang der HMMs, oder sie werden einer weiteren Transformation (z.B. LDA, Ergänzung um Differenz- oder Nachbarvektoren) unterzogen. Eine weitere Verarbeitungsvariante besteht in der Quantisierung der Merkmalvektoren mit einem k-means oder MMI basierten Codebuch. Dies führt zur Verwendung von diskreten HMMs. Beim TP-hybriden Verfahren werden neben den Merkmalvektoren auch deren a posteriori Wahrscheinlichkeiten verwendet.

Ein entscheidender Vorteil einer zusätzlichen Transformation der Merkmalvektoren ist die Berücksichtigung der Nachbarschaft. Dies wird in der Spracherkennung in der Regel durch Δ und $\Delta\Delta$ -Merkmale (Differenzvektoren zu den benachbarten Merkmalvektoren) erreicht oder wie hier durch eine LDA-Transformation (siehe Anhang A) mit der Eigenvek-

tormatrix $M_{lda,f3}$ nach Gl. 9.1

$$\underline{x}_{i,lda} = M_{lda,f3} \cdot \begin{pmatrix} \underline{x}_{i-1} \\ \underline{x}_i \\ \underline{x}_{i+1} \end{pmatrix} \tag{9.1}$$

oder Quantisierung nach Gleichung 9.2:

$$\begin{pmatrix} \underline{x}_{i-1} \\ \underline{x}_i \\ \underline{x}_{i+1} \end{pmatrix} \stackrel{VQ}{\Longrightarrow} y_{i,n} \tag{9.2}$$

In der Regel werden in dieser Arbeit jeweils 3 benachbarte Merkmalvektoren $\underline{x}_{i-1}, \underline{x}_i, \underline{x}_{i+1}$ zusammengefaßt und anschließend LDA-transformiert oder quantisiert, womit jeweils eine Dimensionsreduktion verbunden ist. Die LDA-Transformation bietet außerdem den Vorteil einer besseren Trennbarkeit der einzelnen Klassen.

Für die Erkennung werden die zuvor trainierten oder adaptierten Modelle zusammen mit einem Lexikon oder einem Sprachmodell (N-Gramm) herangezogen.

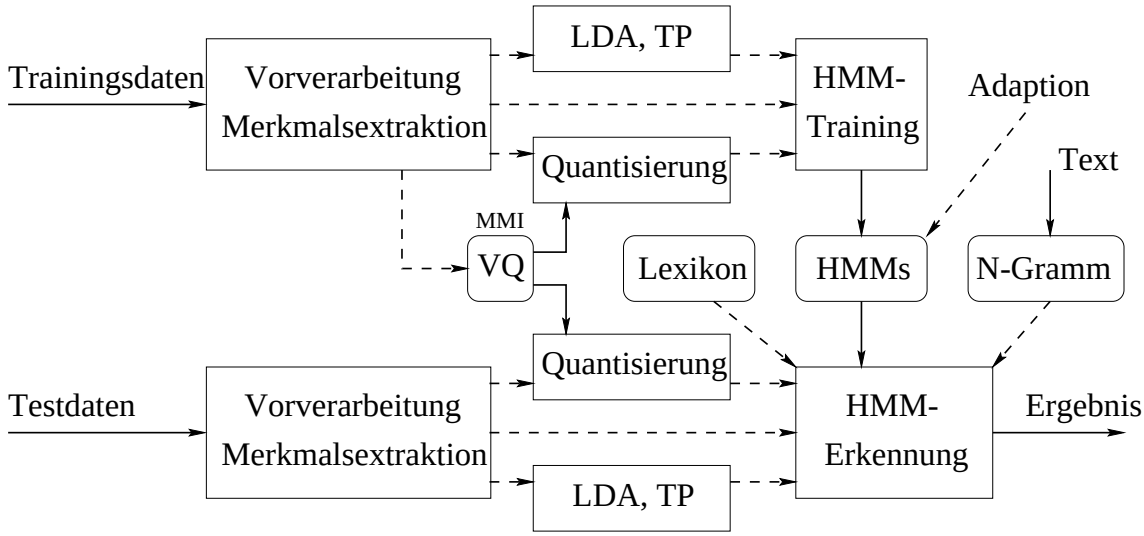


Abbildung 9.1: Prinzipbild der Erkennungssysteme

Definitionen

Die später aufgeführten Erkennungsergebnisse beziehen sich entweder auf die *Erkennungsrate*, bzw. die *Korrektheit* COR nach Gl. 9.3 oder auf die *Akkuratheit* ACC nach Gl. 9.4. Hier ist N_{all} die Gesamtzahl der zu erkennenden Worte oder Zeichen und N_{ok} die Anzahl aller richtig erkannten Worte bzw. Zeichen. Bei einem Einzelwort-Erkennungsproblem spielt

nur die Anzahl der vertauschten Worte N_{subst} eine Rolle, woraus folgt, daß die Erkennungsrate gleich der Akkuratheit ist.

$$COR = \frac{N_{ok}}{N_{all}} = \frac{N_{all} - N_{del} - N_{subst}}{N_{all}} \tag{9.3}$$

$$ACC = \frac{N_{all} - N_{del} - N_{subst} - N_{ins}}{N_{all}} \tag{9.4}$$

Können während des Erkennungsvorganges jedoch auch zusätzliche Zeichen (Zeichen-Erkennung mit einem Sprachmodell) oder Worte (Erkennung ganzer Sätze oder mehrerer Worte) eingefügt (N_{ins}) oder gelöscht (N_{del}) werden, ergibt sich ein Unterschied zwischen ACC und COR. Ein Beispiel für das Löschen von Zeichen ist die Erkennung eines 'm' statt der korrekten Zeichenfolge 'rn'. Für das Einfügen gilt entsprechend das umgekehrte Beispiel. Das hier in der Regel verwendete Maß ist die Akkuratheit ACC. Das ***Fehlermaß*** ERR ergibt sich bei N_{err} Fehlern entsprechend zu

$$ERR = (1 - ACC) \cdot 100\% \ = \frac{N_{err}}{N_{all}} \tag{9.5}$$

falls eine Erkennung ohne Rückweisung vorgenommen wird.
Die ***Rückweisungsrate*** REJ gibt bei Verwendung eines Konfidenzmaßes $Conf$ den Anteil der Daten N_{rej} an, deren Erkennungssicherheit unter einer zu definierenden Schwelle τ liegt:

$$Conf \begin{cases} < \tau \rightarrow zurückweisen \ \ (N_{rej}) \\ \geq \tau \rightarrow klassifizieren \ \ (N_{err} + N_{ok} = N_{all} - N_{rej}) \end{cases} \tag{9.6}$$

Dann gilt:

$$ERR + ACC + REJ = 1 \quad \text{mit:} \quad REJ = \frac{N_{rej}}{N_{all}} \quad \text{und} \quad FAL = \frac{N_{err}}{N_{all} - N_{rej}} \tag{9.7}$$

Der Ausdruck FAL bezeichnet den Anteil der falsch klassifizierten Daten, d.h. der Daten, die nicht zurückgewiesen werden konnten aber trotzdem zu einem Fehler führen. Bei den Testreihen werden die Angaben zur Fehlerquote ERR in Abhängigkeit von der Rückweisungsquote REJ bei variierender Schwelle τ angegeben. Wird kein Konfidenzmaß benutzt, ist REJ gleich Null. Zu beachten ist, daß sich sowohl N_{err} als auch N_{ok} in diesen Definitionen jeweils nur auf die Menge bezieht, die nicht zurückgewiesen wurde.

In der Spracherkennung oder der Biometrie wird bei der Untersuchung von Konfidenzmaßen häufig eine sogenannte ROC-Kurve ('Receiver operating characteristics') als Ergebnis angegeben. Diese gibt das Verhältnis von FAR ('False Accept Rate') zu FRR ('False Reject Rate') an. FAR ist dabei der Anteil der falsch erkannten Daten, die nicht zurückgewiesen wurden. FRR ist der Anteil der korrekt erkannten Daten, die zurückgewiesen wurden. Zur besseren Übersicht sind bei einigen Konfidenz-Untersuchungen auch diese Fehlerraten angegeben. Da jedoch gerade in Hinblick auf eine unüberwachte Adaption im wesentlichen die Fehler interessieren, die sich nur auf die klassifizierte Datenmenge (also die

mögliche Adaptionsmenge) beziehen, werden in dieser Arbeit standardmäßig Angaben zu REJ und ERR bzw. FAL gemacht.

9.2 On-Line Handschrifterkennung

Die folgenden Aspekte wurden anhand der on-line Handschrift-Datenbasis untersucht: verschiedene *Modellierungstechniken* und *Konfidenzmaße*, die *Adaptionsverfahren* und im Vergleich dazu unterschiedliche *Normierungsmethoden*, der Vergleich eines *schreiberabhängigen* zu einem *schreiberunabhängigen System* und der Vergleich von *on-line* und *off-line Systemen* bei Verwendung gleicher Daten.

Bei den verwendeten Modellen handelt es sich jeweils um Monographeme, d.h. für jedes Zeichen existiert genau ein HMM. Tests zu Trigraphemen im on-line Bereich wurden für den schreiberabhängigen Modus in [Kos00a] durchgeführt. Hier konnte zwar eine Steigerung der Erkennungsrate erzielt werden, jedoch mußten die Parameter je Schreiber optimiert werden. Die Wahl der Merkmale (siehe Kap. 3.1) für das on-line Erkennungssystem wurde aus [Kos00a] übernommen und nicht weiter ausgetestet. Das Training wird standardmäßig mit dem Baum-Welch Algorithmus durchgeführt, die Erkennung mit dem Viterbi-Verfahren.

9.2.1 Schreiberabhängiges System

Es wurden vier eigene separate schreiberabhängige Systeme für die Schreiber ABR, ANK, JMR und VDM trainiert und getestet. Die Versuche beziehen sich in erster Linie auf den Vergleich der *Modellierungstechnik* (kontinuierlich, diskret, MMI-hybrid). Hier wurden die Ergebnisse aus [Kos00a] evaluiert und um einen weiteren Schreiber erweitert.

Aufgrund der Daten, die für das off-line System in Kap. 9.3.1 verwendet werden (transformierte on-line Daten) können mit diesem System direkte Vergleiche der Effizienz von *on-* und *off-line* Merkmalen durchgeführt werden (vgl. [Bra99a, Bra99b]). Desweiteren kann ein Vergleich mit adaptierten schreiberunabhängigen Systemen aus Kap. 9.2.2 erfolgen. Ein zusätzlicher Aspekt betrifft hier die Untersuchung der automatischen Auswahl der Anzahl der Zustände von HMMs.

Systemspezifikation

Das Erkennungssystem besteht aus 88 verschiedenen linearen HMMs, von denen im Testdatensatz (Einzelworte) nur etwa 80 verschiedene Zeichen vorkommen. Es wird ein HMM je Zeichen (Buchstaben, Zahlen, Sonderzeichen, siehe Kap. B) gebildet. In der Regel werden 12 Zustände für die Zahlen und Buchstaben verwendet, bis auf kurze Sonderzeichen (drei Zustände). Die Initialisierung des Systems erfolgt auf der Basis von Einzelzeichen (siehe Abb. B.1), das weitere Training anhand von Sätzen bzw. Wortsequenzen, wie in Kap. 3.1.1

beschrieben. Die Erkennung wird jeweils mit einem sehr großen, 30000 Worte umfassenden deutschen Wörterbuch durchgeführt, in dem alle Worte des Testdatensatzes enthalten sind.

Experimentelle Untersuchungen

Bei der Modellierung mit kontinuierlichen HMMs bildet der 14-dimensionale Merkmalvektor $\underline{x}$ (5 dynamische Merkmale und 9 Elemente der Bitmap), wie in Kap. 3.1.1 beschrieben, die Eingabeinformation der HMMs. Werden diskrete oder MMI-hybride HMMs verwendet, wird dieser Merkmalvektor zuerst mit einem oder mehreren Codebüchern quantisiert, wobei das Codebuch entweder nach dem k-means Verfahren erstellt bzw. nach dem MMI-Kriterium optimiert wurde.

Tab. 9.1 zeigt Worterkennungsraten bei den verschiedenen Schreibern bei unterschiedlichen *Modellierungstechniken*. In [Kos00a] wurde gezeigt, daß die Erkennungsraten steigen, wenn nicht der ganze Merkmalvektor mit einem Codebuch quantisiert wird. Auf diese Ergebnisse wird hier zurückgegriffen. Werden nur die 9 Elemente der Bitmap mit einem Codebuch (Größe 300) quantisiert, steigt die Erkennungsquote im Durchschnitt von 92.6% (DIS1bm) bei einem k-means VQ auf 95.7% (HYB1bm-MMI) bei einem MMI-VQ. Dabei werden jeweils fünf benachbarte Frames quantisiert. Bei Verwendung des ganzen Merkmalvektors, der mit zwei Codebüchern quantisiert wurde (Größe 300 für die Bitmap und Größe 200 für die fünf dynamischen Merkmale: Winkel, Winkeldifferenz, Stiftdruck), kann nach der MMI-Optimierung der VQs die Erkennungsrate auf 97.2% (HYB2bmonl-MMI) erhöht werden. Bei den dynamischen Merkmalen werden hier sogar neun benachbarte Frames berücksichtigt. Die Ergebnisse der kontinuierlichen Modellierungstechnik (bis zu vier Gaußschen Mischverteilungen je Zustand) fallen dagegen schlechter aus. 95.3% (KON1bmonl) der Worte konnten richtig erkannt werden. Allerdings ist hier zu beachten, daß zwar der vollständige Merkmalvektor berücksichtigt wurde, nicht jedoch die Nachbarschaft.

Tabelle 9.1: Ergebnisse der schreiberabhängigen on-line Handschrifterkennung (Worterkennungsraten in %) bei unterschiedlichen Modellierungstechniken und einem 30k-WB

Methode	ABR	ANK	JMR	VDM	Durchschnitt
DIS1bm	95.7	95.7	86.6	92.5	92.6
HYB1bm-MMI	97.3	96.8	90.9	97.8	95.7
HYB2bmonl-MMI	98.4	98.4	93.0	98.9	97.2
KON1bmonl	97.3	96.8	91.4	95.7	95.3

Der Vorteil der diskreten bzw. MMI-hybriden Technik besteht in der geringeren Trainingsdatenmenge, die notwendig ist. Wenn man die Anzahl der Gaußschen Mischverteilungen beim

kontinuierlichen System erhöht, sinkt die Erkennungsrate wieder, da zu viele Parameter mit zu wenigen Daten geschätzt werden müssen.

Die Schwankung der Erkennungsrate zwischen den einzelnen Schreibern wird verständlich, wenn man sich die Beispiele in Abb. 3.1 betrachtet. Häufige Fehler betreffen nur Verwechslungen einzelner Buchstaben oder das Wortende, welches oft undeutlich geschrieben wird (z.B. ‘eine’, ‘einer’, ‘einen’).

Als zweiter Aspekt wurde eine automatische *Anpassung der HMM-Struktur* untersucht, die sich auf die Anzahl der Zustände bezieht. Die Anzahl der Zustände je Zeichen richtet sich dabei nach der durchschnittlichen Framelänge (Anzahl der Merkmalvektoren) in den Trainingsdaten. Tabelle 9.2 zeigt die Ergebnisse anhand des Schreibers VDM. Für die Versuchsreihe wurden jeweils diskrete HMMs verwendet, wobei der Merkmalvektor nur aus der Bitmap besteht. Werden für die kurzen Sonderzeichen drei und für alle anderen Buchstaben und Zahlen je 12 Zustände verwendet (Standard-Modellstruktur, vgl. Tab. 9.1), ergibt sich eine Wort-Erkennungsrate von 92.5%. Ein zusätzlicher Test auf 87 Einzelzeichen ergibt eine Erkennungsrate von 86.2%. Wird die Anzahl der Zustände von 12 auf 10 oder 14 geändert, sinkt die Erkennungsrate leicht (siehe Tab. 9.2: Spalte 2 und 3). Die letzten drei Spalten der Tab. 9.2 zeigen die Ergebnisse bei einer automatischen Anpassung der HMM-Struktur bzgl. der Zustände. Abhängig von der minimalen und der maximalen durchschnittlichen Frameanzahl je Zeichen wird die Anzahl der Zustände der zugehörigen Zeichen-Modelle linear zwischen 3 und 12, bzw. 10 und 14, oder 12 und 14 aufgeteilt (bei den letzten beiden Versuchen sind die HMMs für kurze Sonderzeichen unverändert geblieben). Beispielsweise ist bei diesem Schreiber das ‘m’ im Durchschnitt 1.5 mal so lang wie das ‘n’, woraus folgt, daß für das ‘m’ je nach Interpolationsvorschrift deutlich mehr Zustände verwendet werden.

Tabelle 9.2: Ergebnisse der on-line Handschrifterkennung für VDM: Worterkennungsraten in % (Einzelzeichenerkennungsraten in %) bei unterschiedlichen Zustandsanzahl; DIS1bm

Zustände	3/12	3/10	3/14	3-12 lin	3/10-14 lin	3/12-14 lin
VDM	92.5 (86.2)	91.4	92.0	81.3 (89.7)	90.9	92.0

Es zeigt sich, daß größere Unterschiede in der Zustandsanzahl zu deutlich geringeren Worterkennungsraten (81.3% bei einer linearen Aufteilung zwischen 3 und 12 Zuständen) führen, wobei allerdings der Test auf Einzelzeichen eine Steigerung erkennen läßt. Bei der Verknüpfung von Zeichen-Modellen unterschiedlicher Zustandsanzahl zu Worten kommt es (abhängig vom Wörterbuch) leichter zu Verwechslungen. Einfach veranschaulichen kann man sich das Problem bei Betrachtung der Buchstaben ‘m’, ‘n’ und ‘r’. Längenmäßig betrachtet ergibt sich in etwa ‘m=r+n’, was auch unter dem Gesichtspunkt der Merkmalvektoren stimmt. Wenn diese Gleichung nun auch noch für die Anzahl der Zustände der Zeichen

zutrifft auf die sich die Merkmalvektoren abbilden müssen, wird klar, daß eine Unterscheidung von ‘m’ und ‘rn’ unsicherer wird. Aufgrund dieser Untersuchung ist im weiteren die Anzahl der Zustände je Zeichen fest vorgegeben. Mit Ausnahme kurzer Sonderzeichen werden jeweils gleich viele Zustände verwendet.

9.2.2 Schreiberunabhängiges System

Für die zwei verschiedenen schreiberunabhängigen on-line Erkennungssysteme wurden jeweils die Daten von 166 Schreibern verwendet. Der Unterschied liegt lediglich in der *Normierung*. Anhand dieser Daten wurden außerdem Versuche zur *Schreiber-Adaption* und zu *Konfidenzmaßen* durchgeführt.

Systemspezifikation

Das Erkennungssystem besteht, wie beim schreiberabhängigen System, aus 88 verschiedenen linearen HMMs. Es wird ein lineares HMM je Zeichen (siehe Kap. B) gebildet, wobei in der Regel 12 Zustände verwendet werden. Die Initialisierung des Systems erfolgt auf der Basis von Einzelzeichen (siehe Abb. B.1), das weitere Training anhand von Sätzen bzw. Wortsequenzen, wie es in Kap. 3.1.1 beschrieben wurde. Die Erkennung wird jeweils mit einem 2200 Worte umfassenden deutschen Wörterbuch durchgeführt, in dem alle Worte des Testdatensatzes enthalten sind.

Hier wird in der Regel die kontinuierliche Modellierungstechnik (bis zu 15 Gaußschen Mischverteilungen mit diagonaler Kovarianzmatrix) verwendet, da die Trainingsdatenmenge größer ist als beim schreiberabhängigen System. Die Anzahl der Mischverteilungen je Zustand hängt dabei von der Anzahl der zugehörigen Trainingsdaten ab. Außerdem eignet sich die kontinuierliche Modellierungstechnik besonders für den Vergleich der Adaptionsmethoden, auf denen hier der Schwerpunkt liegt. Einige Adaptionsmethoden, wie MAP und MLLR/ SLLR basieren auf der Anpassung der Gaußverteilungen.

Experimentelle Untersuchungen

Die erste Versuchsreihe evaluiert den Einfluß verschiedener *Schreiber-Adaptionsverfahren* und *Normalisierungsmethoden* für das schreiberunabhängige Erkennungssystem anhand von 21 Schreibern (vgl. [Bra01a, Bra02a]). Tab. 9.3 stellt die ermittelten Wort-Erkennungsraten dar, wobei auch der Einfluß der Adaptionsdatenmenge untersucht wird. Es wurden zwei getrennte schreiberunabhängige Basis-Systeme (Tab. 9.3: wi) trainiert, das eine mit normalisierten (Normierung bzgl. Schreibgeschwindigkeit, Größe und Zeichenneigung) Merkmalvektoren, das andere auf Basis der Original-Vektoren, bei denen lediglich die Schreibgeschwindigkeit normiert wurde. Werden alle 4153 Test-Worte getestet, ergibt sich eine Erkennungsrate von 85.8% für die Original-Daten und von 86.7% für das normalisierte Sy-

stem. Dieser Test-Satz wird für die weiteren Versuche je Schreiber halbiert, sodaß nur noch 2071 Testworte zur Verfügung stehen und 2082 Worte für die Adaption. Die Normierung der Merkmalvektoren führt dann zu einer Erkennungsrate von durchschnittlich 87.0% im Vergleich zu 85.7% ohne Normierung. Dies bedeutet für die einzelnen Schreiber Erkennungsraten von 38.0% bis 98.1% bzw. von 64.1% bis 96.1% (siehe Tab. 9.3). Die Unterschiede in der Erkennungsrate zwischen den einzelnen Schreibern sind nach einer Normalisierung deutlich geringer.

Die weiteren Zeilen in Tab. 9.3 zeigen Ergebnisse nach einer überwachten ML-Adaption, einer MAP- und einer MLLR-Adaption, wobei die Anzahl der verwendeten Adaptionsworte (AW) zwischen 6 und 100 variiert wird. Bei einer Adaption mit nur 6 Worten werden zwei verschiedene (zufällig ausgewählte) Adaptionsdatenbasen (AW1 und AW2) gebildet und getestet, um die Ergebnisse, die bei so geringen Datenmengen stark vom Zufall abhängen, zu evaluieren. Die Adaption erfolgt auf jeden Schreiber separat. Es wurden wahlweise die Mittelwertvektoren $\underline{\mu}$, die Kovarianzmatrizen Σ, die Gewichte ω oder die Übergangsmatrix A adaptiert. Bei der MLLR-Adaption wurde außerdem die Anzahl der Cluster bzw. der verwendeten Regressionsmatrizen variiert (Top-Down Verfahren), die sich hier jeweils auf einzelne Zustände (nicht ganze Modelle) beziehen.

Tabelle 9.3: Wort-Erkennungsraten (%) für 21 Schreiber bei unterschiedlichen Adaptionsverfahren (überwacht) und unterschiedlichen Normierungsmethoden; 2.2k-WB

Adaptionsmethode	Original	Normalisiert
Basissystem: **wi** (von-bis)	**85.7** (38.0-98.1)	**87.0** (64.1-96.1)
ML, $\underline{\mu}$, 100 Adaptionsworte	93.7	93.3
ML, ω, 100 AW	91.0	91.9
ML, Σ, 100 AW	90.8	91.7
ML, $\underline{\mu}\omega$, 100 AW (von-bis)	**94.0** (57.6-100.0)	**93.6** (68.5-99.0)
ML, $\underline{\mu}\Sigma$, 100 AW	89.1	88.1
ML, $\underline{\mu}\omega\Sigma A$, 100 AW	89.5	87.6
ML, $\underline{\mu}$, 6 AW (AW1 / AW2)	86.1 / 86.9	86.8 / 86.2
ML, ω, 6 AW (AW1 / AW2)	87.2 / 86.1	87.6 / 87.4
MAP, $\underline{\mu}$, 100 AW	91.1	92.2
MAP, $\underline{\mu}\Sigma$, 100 AW	91.0	92.2
MAP, $\underline{\mu}$, 6 AW (AW1 / AW2)	**87.0 / 87.2**	**88.4 / 88.2**
MLLR, $\underline{\mu}$, 100 AW, 1 Cluster	86.1	87.5
MLLR, $\underline{\mu}$, 100 AW, 16 Cluster	87.2	87.7
MLLR, $\underline{\mu}$, 100 AW, 128 Cluster	88.1	87.1

Wenn die Adaptionsdatenmenge mit ca. 100 Worten relativ groß ist, werden die besten Ergebnisse bei der ML-Adaption erzielt, wenn sowohl die Mittelwerte als auch die Gewichte der HMMs angepaßt werden. Die Worterkennungsrate von 94.0% ist dann bei Verwendung von Original-Daten sogar leicht höher als die der normierten Daten mit 93.6%. Wird neben dem Mittelwert auch die Varianz (oder alle HMM-Parameter) angepaßt, steigt die Anzahl der erkannten Worte nur leicht. Dies deutet auf eine zu große Anzahl zu schätzender Parameter bei zu wenig Trainingsdaten hin. Werden nur 6 Adaptionsworte benutzt, zeigt die MAP-Adaption die beste Steigerung der Erkennungsleistung (durchschnittlich 87.1% bzw. 88.3%). Bei der MLLR-Adaption wird unabhängig von der Clusteranzahl im Durchschnitt nur eine geringe Verbesserung erzielt. Aus diesem Grund wurde hier auch auf die SLLR-Adaption verzichtet, da diese prinzipiell auf der MLLR-Methode aufsetzt. Die deutliche Verringerung der Fehlerquote aufgrund der MLLR-Adaption bei sehr wenigen einzelnen Schreibern konnte nicht verallgemeinert werden.

Tab. 9.4 zeigt die Ergebnisse nach einer unüberwachten Adaption. Da in diesem Modus quasi beliebig viele Daten zur Verfügung stehen würden, wurden jeweils ca. 100 Worte zum Adaptieren verwendet. Auch im unüberwachten Modus kann die Erkennungsrate mit nur 100 Worten deutlich auf 92.0% bzw. 92.3% gesteigert werden, wenn Mittelwerte und Gewichte nach dem ML-Verfahren angepaßt werden. Wie erwartet fallen diese Ergebnisse etwas geringer aus als im überwachten Modus (vgl. Tab. 9.3). Die Wort-Erkennungsrate auf dem Adaptionsdatensatz liegt ebenfalls bei etwa 86%. Dies bedeutet, daß ca. 14% der Wort-Label, mit denen adaptiert wird, falsch sind. Trotzdem führt die unüberwachte Adaption zu einer deutlichen Fehlerreduktion.

Tabelle 9.4: Wort-Erkennungsraten (%) für 21 Schreiber bei unterschiedlichen Adaptionsverfahren (unüberwacht) und unterschiedlichen Normierungsmethoden

Adaptionsmethode	Original	Normalisiert
Basissystem (**wi**)	**85.7**	**87.0**
ML, $\underline{\mu}$, 100 Adaptionsworte	91.6	91.8
ML, ω, 100 AW	89.9	90.8
ML, $\underline{\mu}\omega$, 100 AW	**92.0**	**92.3**
MAP, $\underline{\mu}$, 100 AW	89.9	91.4
MLLR, $\underline{\mu}$, 100 AW, 16 Cluster	87.1	88.1

Im *Vergleich* des *schreiberunabhängigen* zum *schreiberabhängigen* (wd, vgl. Tab. 9.1) System, sind die Erkennungsraten auch nach einer Adaption noch geringer. Dabei ist zu beachten, daß die Ergebnisse in Tab. 9.1 sogar mit einem 30k-WB erzielt wurden. Tab. 9.5 schlüsselt die Ergebnisse für zwei Schreiber (ABR und JMR) auf, die sowohl im schreiberabhängigen als auch im schreiberunabhängigen Test vorkommen.

Tabelle 9.5: Wort-Erkennungsraten (%) für ABR und JMR: Vergleich von wd, wi und Adaption; Original-Daten, KON1bmonl, i.d.R. 2200er WB

System	wd	wi	wi adaptiert: ML, $\mu\omega$, 100 AW
ABR	98.9 (97.3_{30k-WB})	95.7	97.9
JMR	(91.4_{30k-WB})	38.0	57.6

Die folgenden Ergebnisse beziehen sich auf die Auswertung von *Konfidenzmaßen* (siehe auch [Bra02b]). Bisher wurden keine Daten zurückgewiesen, es galt $REJ = 0$. Nun soll untersucht werden, inwieweit die Fehlerrate reduziert werden kann, wenn unsichere Ergebnisse nicht klassifiziert werden. Die Ergebnisse in Abb. 9.2 beziehen sich auf das Basissystem aus Tab. 9.3 mit normalisierten Schriftdaten.

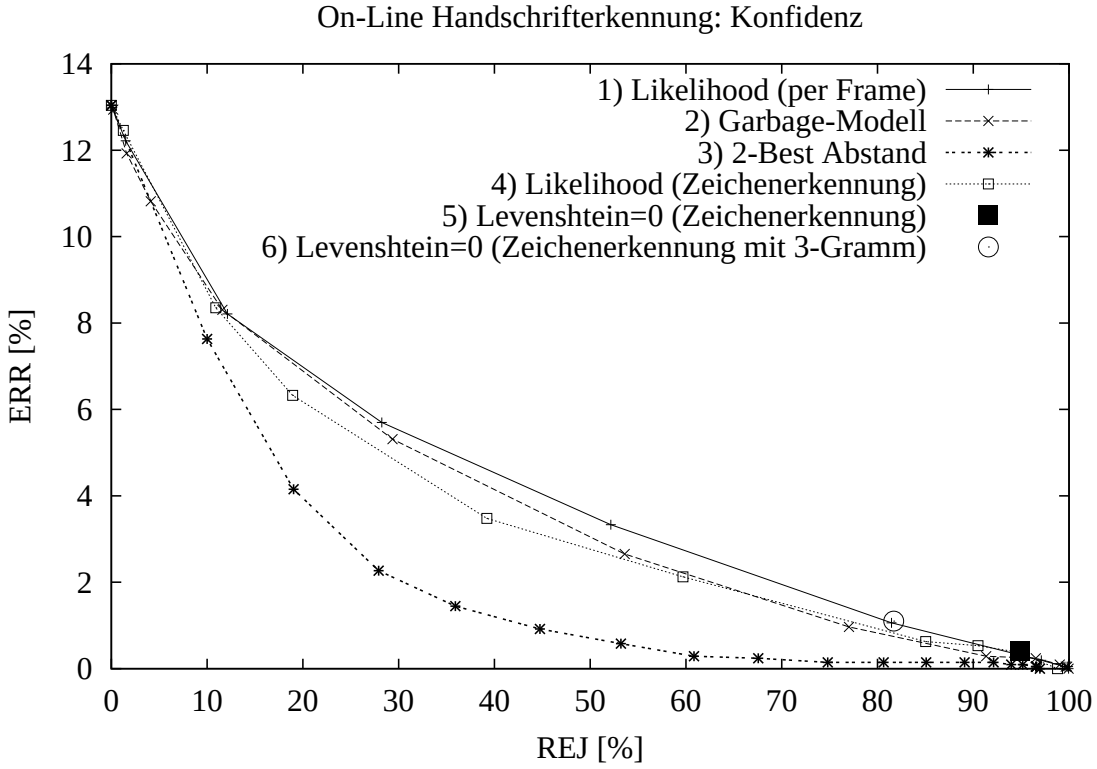


Abbildung 9.2: Rückweisungs- und Fehlerquote bei Verwendung unterschiedlicher Konfidenzmaße und variierender Schwelle τ (21 Schreiber, normalisierte Daten)

Ohne Rückweisung beträgt die Fehlerquote 13.0%. Die Abb. 9.2 zeigt das Verhältnis von Fehlerquote zur Rückweisungsquote bei vier verschiedenen Konfidenzmaßen (vgl. Kap. 7) und unterschiedlicher Schwelle τ: 1) die Frame-normierte Likelihood, 2) die auf ein Garbage-Modell normierte Likelihood, 3) die Auswertung des 2-Best-Abstandes, 4) die auf eine Zeichenerkennung normierte Likelihood. Zusätzlich sind zwei Punkte 5) und 6) mar-

kiert, die auf einer Erkennung ohne Wörterbuch basieren. Die Erkennung von Zeichensequenzen ohne WB wurde bei 6) mit einem allgemeinen deutschen Sprachmodell (Backoff 3-Gramm) durchgeführt. Die eingezeichneten Punkte markieren nach einem Abgleich (Levenshtein-Distanz=0) der erkannten Zeichensequenzen (ohne Nachverarbeitung) mit dem WB den Anteil der Daten, die nicht mit einem WB-Eintrag übereinstimmen, und somit zurückgewiesen werden, und den Anteil der Worte, die trotz Übereinstimmung mit dem WB falsch erkannt wurden.

Wie in Abb. 9.2 zu sehen ist, ist, wie erwartet, das Konfidenzmaß basierend auf der Frame-normierten Likelihood am schlechtesten. Selbst wenn 52.1% der Worte zurückgewiesen werden, sinkt die Wort-Fehlerrate nur auf 3.3%. Dies bedeutet 7.0% falsch klassifizierte Worte (siehe Tab. 9.6), bzw. 2.4% falsch klassifizierte Zeichen. Die Konfidenzmaße, die vom Garbage-Modell oder der Zeichenerkennung abhängen, sind nur etwas effektiver. Das beste Konfidenzmaß basiert auf dem 2-Best Abstand. So kann der Fehler auf 0.9% reduziert werden, wenn 44.8% der Worte zurückgewiesen werden. Dies entspricht einem Anteil von 1.7% falsch klassifizierten Beispielen.

Tabelle 9.6: Ausgewählte Punkte (vgl. Abb. 9.2) der Worterkennung (%) für 21 Schreiber

Konfidenzmaß	Rückweisung	Fehler	falsch klassifiziert		FAR	FRR
	Wort (REJ)	Wort (ERR)	Wort (FAL)	Zeichen	Wort	Wort
Basissystem	0.0	13.0	13.0	6.4	100	0.0
1) Likelihood per Frame	52.1	3.3	7.0	2.4	25.6	48.8
3) 2-Best Abstand	44.8	0.9	1.7	0.6	7.0	37.5

Die Erkennungsrate ist nicht nur stark abhängig von der Größe des verwendeten Wörterbuches, sondern auch von der Ähnlichkeit der WB-Einträge. Diese Tatsache wird vom 2-Best Abstand berücksichtigt. Sind die ersten beiden erkannten Ergebnisse sehr ähnlich (kleine Levenshtein-Distanz), sind auch die zugehörigen Likelihoods sehr ähnlich. Daraus folgt, daß das Konfidenzmaß eher gering ausfällt und das Wort zurückgewiesen wird. Bei einem anderen Wörterbuch, bei dem beispielsweise die zweitbeste Hypothese nicht vorkommt, ändert sich das Konfidenzmaß und das Verhältnis von Fehler- und Rückweisungsquote kann völlig anders ausfallen. Die anderen Konfidenzmaße sind WB-unabhängig und können von daher allerdings auch Fehler, die auf Verwechslungen von nur einem oder wenigen Buchstaben basieren, nicht ausgleichen.

Dieser Effekt wird auch deutlich, wenn die Ergebnisse betrachtet werden, die ohne WB erkannt wurden. Diese Ergebnisse sind deutlich schlechter. Hier werden ohne Sprachmodell 94.9% zurückgewiesen bei einer Fehlerquote von 0.4% (vgl. Abb. 9.2: 5)). Wird ein 3-Gramm zur Erkennung verwendet, ergibt sich der Wort-Fehler zu 1.1% bei 81.7%

Rückweisungen (Abb. 9.2: 6)). Ohne WB-Abgleich würden alle Rückweisungen zu einem Fehler führen. Aber auch hier zeigt sich, daß die Zeichensequenz-Erkennung zu 18.3% mit einem WB-Eintrag übereinstimmt (vgl. Konfidenzmaß 4). Allerdings sind dann ca. 6% dieser Übereinstimmungen falsch. Auch diese Fehler beruhen häufig auf einer Verwechslung von nur einem oder wenigen Buchstaben.

Anhand dieser Ergebnisse mit Konfidenzmaßen kann auch eine *unüberwachte Adaption* durchgeführt werden, bei der nur die sicher erkannten Label benutzt werden. Der folgende Test benutzt von den 2082 Adaptionsworten nur noch 1444 für alle 21 Schreiber, indem das 2-Best Konfidenzmaß bei einer Rückweisungsquote von 44.8% (siehe Tabelle 9.6) die Adaptionsdaten bestimmt.

Tabelle 9.7: Wort-Erkennungsraten (%) für 21 Schreiber bei einer unüberwachten Adaption mit Konfidenzmaßen (normalisierte Daten, Adaptionsdaten: $FAL = 1.7\%\ ,\ REJ = 44.8\%$)

Basissystem (wi)	Adaption: ML, $\underline{\mu}\omega$	Adaption: MAP, $\underline{\mu}$
87.0	91.3	89.5

Tab. 9.7 zeigt, wie sich die Verwendung weniger Label-Fehler aber auch weniger Adaptionsdaten auf die Erkennungsergebnisse auswirkt. Im überwachten Modus konnten mit je 100 Adaptionsworten 93.6% korrekt erkannte Worte nach der ML-Methode erzielt werden (vgl. Tab. 9.3), im unüberwachten Modus immerhin noch 92.3% (siehe Tab. 9.4). Im Vergleich dazu ist die Erkennungsrate hier mit 91.3% geringer, allerdings deutlich höher als bei einer Adaption mit sechs Worten. Die Ergebnisse zeigen jedoch, daß eine unüberwachte Adaption mit Labelfehlern (hier 6.4% der Zeichen) bessere Ergebnisse erzeugt als eine korrektere aber deutlich reduzierte Datenmenge. Wenn dieses Verfahren allerdings später im laufenden Betrieb eingesetzt werden kann, können die Adaptionsdatenmengen beliebig erweitert werden.

Die hier entwickelten Algorithmen sind in keiner Weise zeitoptimiert, außerdem ist die Erkennungsdauer u.a. von der WB-Größe, der Anzahl der Mischverteilungen und den eingestellten Parametern abhängig. Größenordnungsmäßig dauert die Erkennung hier etwa 0.5 bis 2 Sekunden je Wort (Pentium II, 400 MHz). Diese Zeitspanne läßt sich aber extrem verkürzen, wie z.B. die Praxisanwendung für die Postsortierung zeigt.

9.3 Off-Line Handschrifterkennung

Die beiden off-line Handschrift-Datenbasen – für den schreiberabhängigen und den schreiberunabhängigen Modus – unterscheiden sich hinsichtlich der Datengewinnung und Merkmalextraktion so stark (vgl. Kap. 3.2), daß ein direkter Vergleich der Ergebnisse hier nicht

möglich ist. Für die off-line Erkennung wurden alle thematischen Schwerpunkte dieser Arbeit untersucht: verschiedene HMM-*Modellierungstechniken*, die Verwendung von *Kontext-* und *Sprachmodellen*, *Konfidenzmaße*, (Postamt-) *Adaptionsverfahren* und der Vergleich von *on- und off-line* Erkennung. Für das Training wird wiederum der Baum-Welch-Algorithmus verwendet und für die Wörterbuch-basierte Erkennung der Viterbi-Algorithmus. Die Erkennung unter Berücksichtigung von Sprachmodellen erfolgt über das Stack-Dekoder Verfahren, wie in Kap. 5.1 beschrieben.

9.3.1 Konvertierte On-Line Daten (schreiberabhängig)

Analog zu dem on-line Erkennungssystem in Kap. 9.2.1 wurden vier schreiberabhängige Erkennungssysteme (Einzelworte in deutscher Sprache) für die Schreiber ABR, ANK, JMR und VDM gebildet. Der Schwerpunkt liegt im Vergleich der *Modellierungstechniken*, wobei insbesondere die hybriden Verfahren berücksichtigt werden, und der Verwendung von *Backoff Zeichen N-Grammen* (siehe [Bra00a, Bra00c, Bra01c]) im Vergleich zu unterschiedlich großen Wörterbüchern. Außerdem werden *Trigrapheme* für die Erkennung untersucht. Aufgrund der Datengewinnung wird hier außerdem ein direkter Vergleich zwischen on- und off-line Erkennunssystemen möglich (vgl. auch [Bra99a, Bra99b]).

Systemspezifikation

Das Erkennungssystem besteht aus 88 verschiedenen linearen HMMs, von denen im Testdatensatz (Einzelworte) nur etwa 80 verschiedene Zeichen vorkommen (keine Sonderzeichen wie z.B. Satzende-Symbole). Es wird ein HMM je Zeichen (Buchstaben, Zahlen, Sonderzeichen, siehe Anhang B) gebildet. In der Regel werden 8 Zustände für die Zahlen und Buchstaben verwendet, bei kurzen Sonderzeichen (‘.,’ etc.) nur zwei Zustände. Die Sonderzeichen werden zwar auf einem Einzelzeichentest überprüft, spielen jedoch bei der Worterkennung keine Rolle. Die Initialisierung erfolgt wiederum auf der Basis von Einzelzeichen (je Schreiber), das weitere Training anhand von Sätzen. Die Erkennung wird i.d.R. mit dem gleichen 30000 Worte umfassenden deutschen Wörterbuch durchgeführt (kein OOV), welches auch für die on-line Erkennung verwendet wird. Zusätzlich werden Testreihen mit einem 200000er WB und einem auf dem Text-Corpus der Sprachmodelle erstellten 50000er WB (mit OOV) durchgeführt, um die Einführung von Zeichen N-Grammen zu begründen. Die Zeichensequenz-Erkennung ohne WB erfolgt anhand des in Kap. 5.3 beschriebenen allgemeinen deutschen Sprachmodells.

Experimentelle Untersuchungen

Die Ergebnisse in den folgenden Tabellen 9.8 und 9.9 ermöglichen einen direkten Vergleich der untersuchten *Modellierungstechniken* bei Verwendung des üblichen 30000er

Wörterbuches. Die Erkennungsraten in Tab. 9.8 und Tab. 9.9 unterscheiden sich lediglich in der Wahl der Merkmalvektoren: unterabgetastete Bitmap mit 9 Elementen bzw. 8 DCT-Koeffizienten des jeweiligen Fensters (*bm*), ergänzt um die gleichen 3 zusätzlichen Merkmale (*zus*). Bei den Versuchsreihen mit diskreten (k-means VQ: DIS2bmzus) und MMI-hybriden (HYB2bmzus-MMI) HMMs werden die Bitmap-Merkmale und die zusätzlichen Merkmale mit zwei verschiedenen Codebüchern quantisiert (Größe 200 und 100), wobei jeweils drei benachbarte Frames berücksichtigt werden. Bei der kontinuierlichen Modellierungstechnik wird der vollständige Merkmalvektor $\underline{x}$ verwendet, eine Aufteilung in zwei Teilvektoren – wie bei den diskreten HMMs – führte zu geringeren Erkennungsraten. Die Anzahl der Gaußschen Mischverteilungen je Zustand wurde dabei für jeden Schreiber individuell optimiert und variiert bei diesen Tests zwischen eins und acht.

Tabelle 9.8: Ergebnisse der off-line Handschrifterkennung (Wort-Erkennungsraten in % bei einem 30k-WB) bei unterschiedlichen Modellierungstechniken; der Merkmalvektor besteht aus der quantisierten Bitmap und den zusätzlichen Merkmalen

Methode	ABR	ANK	JMR	VDM	Durchschnitt
DIS2bmzus	99.5	90.7	77.7	80.4	87.1
HYB2bmzus-MMI	98.4	95.1	80.4	83.1	89.3
KON1bmzus	95.6	82.5	79.7	75.0	79.7

Im Durchschnitt kann der relative Fehler um fast 20% reduziert werden, wenn MMI-hybride anstatt diskrete HMMs verwendet werden (Vergleich von DIS2bmzus und HYB2bmzus-MMI). Die Versuche mit kontinuierlichen HMMs ergaben geringere Erkennungsquoten, wobei hier ein deutlicher Unterschied deutlich wird, wenn statt der unterabgetasteten Bitmap die DCT-Koeffizienten des Fensters verwendet werden (Tab. 9.8 und 9.9: KON1bmzus). Bei diskreten HMMs spielt diese Änderung der Merkmale fast keine Rolle.

Wie wichtig die drei zusätzlichen Merkmale für die Erkennung sind, zeigt das Ergebnis, wenn diese Merkmale nicht beachtet werden. Dann sinkt die Worterkennungsrate im Durchschnitt von 89.2% auf 78.9% bei MMI-hybriden HMMs (siehe Tab. 9.9: HYB1bm-MMI).
Auch die zweite hybride Modellierungstechnik, die Tied Posteriors, werden hier untersucht. Die besten Ergebnisse – im Durchschnitt 86.3% (Tab. 9.9: HYB1bmzus-TP15) – mit dieser Technik ergeben sich, wenn 15 benachbarte Frames am Eingang des Neuronalen Netzes (MLP mit 300 Neuronen in der verdeckten Schicht) angelegt werden um die Ausgabewahrscheinlichkeit der HMMs zu ermitteln. Bei 7 Frames sinkt die Quote bereits auf 84.5%.

Tabelle 9.9: Ergebnisse der off-line Handschrifterkennung (Wort-Erkennungsraten in % bei einem 30k-WB) bei unterschiedlichen Modellierungstechniken; der Merkmalvektor besteht aus den DCT-Koeffizienten und den zusätzlichen Merkmalen

Methode	ABR	ANK	JMR	VDM	Durchschnitt
DIS2bmzus	98.4	92.3	78.3	77.7	86.7
HYB2bmzus-MMI	98.9	93.4	80.4	84.2	89.2
KON1bmzus	96.7	89.6	70.1	78.6	83.7
HYB1bm-MMI	93.6	81.2	74.3	66.3	78.9
HYB1bmzus-TP15	96.2	92.9	78.3	77.7	86.3
HYB1bmzus-TP7	95.1	86.9	78.3	77.7	84.5

Da diese Versuche gezeigt haben, daß die MMI-hybride Modellierung die besten Ergebnisse aufweist, werden weitere Tests zur *Sprachmodell-basierten Erkennung* in dieser Technik erprobt.
Tab. 9.10 zeigt zum einen Ergebnisse, die ohne Wörterbuch erzielt wurden und zum anderen Ergebnisse bei verschiedenen sehr großen Wörterbüchern. Ohne Wörterbuch und ohne Sprachmodell ergibt sich eine Zeichen-Erkennungsrate unter Berücksichtigung von eingefügten und gelöschten Zeichen von 75.4% im Durchschnitt. Dies entspricht einer Worterkennungsrate von nur 36%.

Tabelle 9.10: Ergebnisse der off-line Handschrifterkennung; Verwendung von N-Grammen ohne Wörterbuch: Zeichen-Akkuratheit ACC in %; Verwendung verschiedener WB: Wort-COR in %; der Merkmalvektor besteht u.a. aus den DCT-Koeffizienten (HYB2bmzus-MMI)

Methode	ABR	ANK	JMR	VDM	Durchschnitt Zeichen	Durchschnitt Worte
	Zeichen-Erkennungsraten ACC					
HYB2bmzus-MMI	90.7	79.2	65.2	66.6	75.4	36
3-Gramm, pp=11.9	94.2	87.7	75.1	75.2	83.1	52
5-Gramm, pp=8.9	96.0	91.9	81.4	82.4	87.9	65
7-Gramm, pp=9.7	95.4	91.7	82.5	82.5	88.0	66
	Wort-Erkennungsraten COR					
30k-WB	98.9	93.4	80.4	84.2	–	89.2
200k-WB	96.3	87.6	73.3	73.8	–	82.7
50k-WB (zufällig) (OOV: 34%)	65.8	59.7	48.1	54.6	–	57.1

Wird eine Erkennung mit Zeichen N-Grammen durchgeführt, steigt die Zeichen-Erkennungsrate deutlich auf bis zu 87.9% bei einer Kontexttiefe von $N = 5$. Diese Fehlerreduktion wird bei einem 7-Gramm nicht weiter fortgesetzt (die durchschnittliche Wortlänge beträgt sechs Zeichen), sodaß sich eine Wort-Erkennungsrate von etwa 66% ergibt. Im Vergleich zu 89.2% korrekten Worten, die mit einem 30000er WB erzielt werden, ist dieser Wert gering. Allerdings stellt die N-Gramm Erkennung quasi eine Erkennung mit unbegrenztem Vokabular dar. So werden auch nur noch 82.7% mit einem 200000er WB (kein OOV) richtig erkannt.
Wird gar ein Wörterbuch aus dem Text-Corpus (zufällig ausgewählte 4 Millionen Worte), der zum Training des Sprachmodells dient, bestimmt, ergeben sich nur etwa 50000 verschiedene WB-Einträge und die Wort-Erkennungsrate sinkt auf 57.1% (vgl. Tab. 9.10: 50k-WB). Dies liegt allerdings daran, daß 34% der Testworte in diesem 50k-WB nicht vorkommen. Wenn also der gleiche unbekannte Text entweder als Sprachmodell oder als Wörterbuch genutzt werden kann, ist in diesem Fall die Erkennung mit N-Grammen überlegen. Dieses Beispiel verdeutlicht, wie schwierig es ist, in der Praxis ein WB ohne OOV zu bestimmen, wenn keine Vorgaben (Thema, Anwendungsbereich, etc.) zum System gemacht werden sollen. Wenn hingegen ein 7-Gramm auf den Worten des 30000er WB statt auf den 4 Millionen Worten der Web-Seiten trainiert wird, ändert sich die Erkennungsrate kaum. Die Perplexität sinkt von 9.7 auf 9.2, da das 30k-WB alle Testworte beinhaltet.

Eine weitere Versuchsreihe betrifft die Untersuchung von *Kontextmodellen*. Die bisherigen Ergebnisse beziehen sich jeweils auf Monographeme. Trigrapheme wurden sowohl bei der MMI-hybriden als auch der kontinuierlichen Modellierungstechnik getestet.
Tab. 9.11 zeigt für beide Modellierungstechniken die besten Ergebnisse, die erzielt wurden. Die Ergebnisse der Monographeme in Tab. 9.11 entsprechen denen von Tab. 9.9 und sind hier nur zum besseren Vergleich noch einmal angegeben. In den Klammern sind jeweils die Anzahl der verschiedenen HMMs und die Anzahl der Ausgabeverteilungen dargestellt. Im diskreten Fall (MMI-hybrid) gibt es 88 Monographeme (Anzahl der erlaubten Zeichen) mit insgesamt 628 verschiedenen Ausgabeverteilungen bzw. 628 Zuständen. Bei einer Erweiterung auf Trigrapheme und anschließender datengetriebener Clusterung erhöht sich sowohl die Anzahl der Modelle als auch die der Ausgabeverteilungen. Entsprechend gibt es bei der kontinuierlichen Modellierungstechnik ebenfalls 88 Monographeme und je Schreiber eine unterschiedliche Anzahl von Ausgabeverteilungen, die hier die Anzahl der Gaußfunktionen (optimierte Anzahl je Schreiber; z.B. bei ABR: 628, ANK: 1480) insgesamt beschreibt. Hier konnten die besten Trigraphem-Ergebnisse mit dem Entscheidungsbaum-basierten Clusterverfahren erzielt werden.
Es kann zwar in der Regel eine Steigerung der Erkennungsrate erzielt werden, allerdings müssen die Cluster-Parameter für jeden Schreiber individuell angepaßt werden. Dies wird

auch in Tab. 9.11 deutlich, wenn man die optimale Anzahl der Trigraphem-HMMs je Schreiber vergleicht (z.B. 228 HMMs bei ABR, 483 HMMs bei JMR). Die Verwendung von Trigraphemen bedeutet eine enorme Erhöhung der zu schätzenden Parameter, was bei dieser relativ kleinen Trainingsdatenbasis von 2000 Worten je Schreiber nicht zu einem robusten System führt.

Tabelle 9.11: Ergebnisse mit dem 30k-WB bei unterschiedlichen Trigraphemen: Wort-Erkennungsraten in % (Anzahl HMMs/ Anzahl Ausgabeverteilungen); der Merkmalvektor besteht aus den DCT-Koeffizienten und den zusätzlichen Merkmalen

Methode	ABR	ANK	JMR	VDM	Durchschnitt
	HYB2bmzus-MMI				
Monographeme	98.9 (88/628)	93.4 (88/628)	80.4 (88/628)	84.2 (88/628)	89.2 (88/628)
Trigrapheme Datengetrieben	98.4 (228/740)	96.2 (239/653)	85.9 (483/798)	83.7 (351/789)	91.0 (325/745)
	KON1bmzus				
Monographeme	96.7 (88/628)	89.6 (88/1460)	70.1 (88/1394)	78.6 (88/1657)	83.7 (88/1285)
Trigrapheme Entscheidungsbaum	98.4 (247/694)	90.7 (360/2015)	73.9 (285/1441)	82.6 (403/2196)	86.4 (324/1587)

9.3.2 Handgeschriebene Adressen (schreiberunabhängig)

Die zwei vorhandenen Adreß-Datenbasen für deutsche und amerikanische Adressen werden im folgenden separat behandelt, da die Wörterbücher und auch die zulässigen Einzelzeichen, wie beispielsweise Umlaute und Sonderzeichen, variieren. Prinzipielle Verfahren und Modellierungstechniken gelten bzgl. der Systemspezifikation jedoch für beide Datenbasen.

Systemspezifikation

Es wird in der Regel für jedes Zeichen (77 für das deutsche und 69 für das amerikanische System) ein lineares HMM mit drei Zuständen trainiert. Für kurze Sonderzeichen werden HMMs mit einem Zustand verwendet. Als Standard wird die Semi-Kontinuierliche Modellierungstechnik mit 300 Gaußschen Mischverteilungen und vollbesetzter Kovarianzmatrix verwendet. Somit kann bei diesen SD-Daten ein direkter Praxisbezug hergestellt werden, da SD eine ähnliche Erkennungsmethode verwendet. Die gültigen Zeichen bestehen aus Buchstaben und Zahlen und einigen Sonderzeichen (siehe Anhang C), wobei Verwechslungen von einigen Sonderzeichen, die die Bedeutung nicht verändern, bei der Erkennung nicht als

Fehler gewertet werden (z.B. ‘Dorfstr.’ = ‘Dorfstr’). Auch Verwechslungen von Groß- und Kleinschreibung werden nicht berücksichtigt. Das zugehörige Wörterbuch sieht dann prinzipiell folgendermaßen aus (# symbolisiert ein Leerzeichen):

- HAMBURG: H A M B U R G *oder* H a m b u r g
- DORFSTR: D O R F S T R *od.* D O R F S T R . *od.* D o r f s t r *od.* D o r f s t r . *od.* D O R F # S T R *od.* D o r f # S t r *od.* ...

Allerdings werden verschiedene Erkennungsergebnisse wie z.B. ‘Frankfurt aM’ und ‘Frankfurt am Main’ trotz gleicher Bedeutung als Fehler gewertet. Die Erkennung von Zahlen als PLZ oder Hausnummer spielt nur insofern eine Rolle, als daß bei der Adresse die Abgrenzung zur Stadt bzw. zum Straßennamen ggf. automatisch erfolgen muß. Erkennungsfehler bei der PLZ oder Hausnummer werden nicht gewertet. Bei den amerikanischen Daten hingegen bestehen viele Straßennamen aus Zahlen (‘2nd St’), welche dann selbstverständlich bei der Erkennung berücksichtigt werden. Die Wörterbücher verschiedener Größe enthalten kein OOV und werden in der Regel separat für Straßen und Städte gebildet. Die verwendeten Sprachmodelle wurden anhand des Text-Corpus gemäß Kap. 5.3 bestimmt. Im Regelfall wird eine LDA auf drei benachbarten Merkmalframes gebildet, so daß die Dimension der benachbarten Merkmalvektoren $\underline{x}_{i-1}, \underline{x}_i, \underline{x}_{i+1}$ von 60 auf $D = 30$ reduziert wird.

Experimentelle Untersuchungen

Während beim amerikanischen Adreß-Erkennungssystem der Schwerpunkt der Experimente auf der *Modellierungstechnik* und dem Einfluß der *Wörterbuch-Größe* liegt, werden anhand des deutschen Systems *alle Aspekte* dieser Arbeit untersucht.
Die automatische Erkennung handschriftlicher Adressen ist keine Neuheit, wie die Anwendung von SD in der Praxis zeigt. Hier werden jedoch neue Methoden (‘Postamt-Adaption’, Adreß-Sprachmodelle) vorgestellt, um die bestehenden Erkennungssysteme zu optimieren und somit wirtschaftlicher zu gestalten. Dabei spielt die Erkennungssicherheit, aber auch die automatische Anpassung der Systeme an neue Gegebenheiten eine entscheidende Rolle.

Deutsches Adreß-Erkennungssystem

Für die Basisdaten aus verschiedenen Postämtern Deutschlands wird ein 421er WB für die Städte bestimmt und ein 901er WB für die Straßen (jew. ohne OOV), wobei jeweils die Label der Testdaten berücksichtigt werden. Entsprechende Wörterbücher werden für die speziellen Postämter – deren Daten zur Postamt-Adaption verwendet werden – gebildet, indem alle Label der von diesem Postamt stammenden Daten aufgenommen werden. Weitere sehr große (je ca. 20k Einträge oder 50k) wurden zu Vergleichszwecken aus möglichen gültigen Adreß-Listen erstellt. Ziel dieser Vorgehensweise ist die Bestimmung von vollständigen

Wörterbüchern (kein OOV), wobei durch die Wahl der WB-Größe unterschiedliche Szenarien der realen Praxisanwendung simuliert werden: entweder kann die PLZ zuvor relativ sicher erkannt werden und das WB für die nachfolgende Erkennung der kursiven Schrift (Stadt oder Straße) anhand dieses Ergebnisses eingeschränkt werden, oder die PLZ ist nicht vorhanden oder unleserlich, was dazu führt, daß das WB alle gültigen Worte enthalten muß.
Folgende Standard-Wörterbücher werden verwendet:

- Basisdaten: 421er Städte-WB, 901er Straßen-WB
- Adaptionsdaten HRO: 716er Städte-WB, 1240er Straßen-WB
- Adaptionsdaten STR: 637er Städte-WB, 1227er Straßen-WB
- Adaptionsdaten HAL: 736er Städte-WB, 1154er Straßen-WB
- Adaptionsdaten HAM: 562er Städte-WB, 1220er Straßen-WB

Tabelle 9.12 zeigt Wort-Erkennungsraten der deutschen Basisdaten bei unterschiedlichen *Modellierungstechniken*.

Tabelle 9.12: Wort-Erkennungsraten (%) der Basisdaten (deutsche Städte- und Straßen-Namen) bei unterschiedlichen Modellierungstechniken: jeweils LDA, $D = 30$

System	KON diag, 18 Mix	SK diag, 300 Gauß	SK voll, 300 Gauß	TP 7 Frames
Stadt (421er WB)	87.5	83.6	86.8	92.1
Straße (901er WB)	89.7	85.3	91.0	94.1

Die mit Abstand besten Ergebnisse werden mit dem hybriden Tied-Posterior System erzielt. Die Erkennungsraten von 92.1% für die Städte und 94.1% für die Straßen basieren allerdings auf einem MLP mit einer Eingangsschicht von 7×30 Neuronen, wodurch sich die Rechenzeit extrem verlängert. Deshalb wird aufgrund der für die Anwendungspraxis relevanten Vorgaben, das Semi-Kontinuierliche System mit 300 Gaußschen Mischverteilungen (vollbesetzte Kovarianzmatrix) als Standard eingesetzt. Die Erkennungsrate von 86.8% für die Städte und 91.0% für die Straßen bei Verwendung der Standard-Wörterbücher dient deshalb im weiteren als Referenzsystem. Bei der kontinuierlichen Technik (diagonale Kovarianzmatrix) mit bis zu 18 Mischverteilungen je Zustand ergeben sich ähnlich hohe Erkennungsraten.

Tab. 9.13 zeigt im Vergleich dazu die Ergebnisse mit einem offenen Vokabular auf der Basis von *Sprachmodellen*. Da Zeichen N-Gramme gewissermaßen eine Erkennung mit unendlich großem Wörterbuch beschreiben, sind zum Vergleich auch die Wort- und Zeichenerkennungsraten sehr großer WB (20000 Einträge) angegeben. Wie man in Tab. 9.13 sieht,

sinkt die Wort-Erkennungsrate deutlich von 86.8% auf 63.9% bei den Städtenamen und von 91.0% auf 86.0% bei den Straßennamen, wenn die Anzahl der WB-Einträge zunimmt. So werden bei einem 30k-WB nur noch 82.4% der Straßen korrekt erkannt. Diese relativ hohen Erkennungsraten beruhen allerdings jeweils auf vollständigen WB ohne OOV.
Wird weder ein WB noch ein N-Gramm benutzt, werden nur 2.5% der Städte und 0.4% der Straßen korrekt erkannt (Tab. 9.13: c). Diese Akkuratheit kann allerdings durch eine 7-Gramm basierte Erkennung deutlich auf 40.0% bzw. 35.1% gesteigert werden (Tab. 9.13: f). Der wesentliche Unterschied der verwendeten Sprachmodelle liegt im Trainingstext. So sind die präsentierten Ergebnisse abhängig von der Übereinstimmung zwischen Testdaten und Sprachmodell (vgl. auch die Perplexitätsbetrachtungen in Tab. 9.14). Ein allgemeines Adreß N-Gramm, welches auf einer Liste aller gültigen Städte- und Straßennamen erstellt wird (*Städte+Straßen-Liste*) kann verwendet werden, wenn die Adreß-Struktur (Reihenfolge von Stadt und Straße) unbekannt ist. N-Gramme, die nur auf Städte- oder Straßennamen trainiert werden (Tab. 9.13: g,h), liefern bei den Städten eine höhere Erkennungsrate. Hier konnte, im Gegensatz zur Straßenerkennung, allerdings auch die Häufigkeit der Städtenamen in Abhängigkeit der Einwohnerzahl berücksichtigt werden. Werden gar die Trainingsdaten bei der Bildung des N-Grammes berücksichtigt, was einer Art Adaption des Sprachmodells auf bestimmte Postämter gleichkommt, steigt die Erkennungsrate weiter (Tab. 9.13: i,j). Ein allgemeiner deutscher Text hingegen ist nicht repräsentativ und führt zu geringen Erkennungsraten.

Tabelle 9.13: Wort-Erkennungsraten in % der Basisdaten (deutsche Adressen) bei unterschiedlichen Sprachmodellen: Wort-ACC (Zeichen-ACC); SK-Modellierung mit 300 Gauß

Methode	Stadt	Straße
a) Standard-WB (421 Städte/ 901 Straßen)	86.8 (92.2)	91.0 (95.2)
b) große WB (je 20k)	63.9 (81.8)	86.0 (92.3)
c) ohne WB, ohne N-Gramm	2.5 (45.2)	0.4 (45.8)
d) ohne WB, 3-Gramm: Städte+Straßen-Liste	14.8 (58.5)	11.7 (65.9)
e) ohne WB, 5-Gramm: Städte+Straßen-Liste	35.8 (67.4)	27.5 (72.9)
f) ohne WB, 7-Gramm: Städte+Straßen-Liste	40.0 (68.4)	35.1 (75.2)
g) ohne WB, 5-Gramm: Städte-Liste (rel. Häufigkeit)	42.9 (72.7)	—
h) ohne WB, 5-Gramm: Straßen-Liste	—	27.5 (72.8)
i) ohne WB, 5-Gramm: Städte-Datenbasis	59.4 (79.8)	—
j) ohne WB, 5-Gramm: Straßen-Datenbasis	—	33.3 (74.4)
k) ohne WB, 3-Gramm: allg. deutscher Text	8.0 (51.3)	2.5 (47.6)

Diese Ergebnisse lassen sich auch anhand der Perplexität der Testdaten auf dem jeweiligen Sprachmodell verifizieren (Tab. 9.14). Je spezifischer das N-Gramm zur Adreß-Datenbasis

paßt, je geringer ist die Perplexität. Auch mit höherer Kontexttiefe nimmt die Perplexität ab, wobei die Änderung vom 5-Gramm zum 7-Gramm allerdings nur noch gering ausfällt (vgl. Tab. 9.14: e,f).

Tabelle 9.14: Perplexität pp der Basis-Testdaten (Stadt oder Straße) bei verschiedenen Sprachmodellen (vgl. Tab. 9.13)

Test	d)	e)	f)	g)	h)	i)	j)	k)
Stadt	10.1	6.3	6.0	5.1	6.6	3.5	10.8	21.7
Straße	13.8	10.7	10.1	16.5	10.8	28.6	6.6	23.8

Die Wort-Erkennungsraten der N-Gramme im Vergleich zum korrekten vollständigen Wörterbuch fallen zwar gering aus, allerdings wurde hier noch keine Nachbearbeitung der Ergebnisse vorgenommen.
Die Zeichenerkennungsrate von 68.4% bzw. 75.2% beim 7-Gramm ist relativ hoch. Einige Wort-Fehler entstehen nur durch Verwechslungen einzelner Buchstaben. Ein wesentlicher Vorteil einer Sprachmodell-basierten Erkennung ist die gute Möglichkeit zur Nachbearbeitung, da sozusagen schlüssige Fehler entstehen. So können die Methoden der normalen OCR von Einzelzeichen auf die erkannten Zeichensequenzen angewandt werden. Bei einer WB-basierten Erkennung wird das wahrscheinlichste Wort des WB erkannt. Wenn das korrekte Wort jedoch nicht im WB vorkommt (OOV aufgrund falscher PLZ oder Verwechslung von Land/ Stadt/ Straße/ Name), ist häufig keine Ähnlichkeit (z.B. hohe Levenshtein-Distanz) zwischen erkanntem und korrektem Label zu sehen. Eine Nachbearbeitung wird somit quasi unmöglich. Das Gegenteil ist bei der N-Gramm basierten Erkennung der Fall. Hier können Fehler einfacher korrigiert werden.

Das Problem, mit welcher Sicherheit ein Erkennungsergebnis korrekt ist, wird im folgenden anhand der *Konfidenzmaße* betrachtet. Der Nachteil einer WB-basierten Erkennung mit OOV wird deutlich, wenn man eine Erkennung mit ‘falschem WB’ durchführt. Wird die Testdatenbasis der Städte mit dem 20k Straßen-WB erkannt und umgekehrt, so liegt die Fehlerquote verständlicherweise jeweils bei 100%. Betrachtet man allerdings die durchschnittliche Likelihood per Frame auf den jeweiligen Testdaten, so ist diese bei einem falschen WB nur um etwa 2.7% kleiner als bei Verwendung des korrekten WB ohne OOV. Daraus wird zum einen ersichtlich, daß die Entscheidung, ob es sich um ein OOV-Problem handelt, nicht ohne weiteres zu treffen ist. Zum anderen wird klar, daß die Frame-normierte Likelihood als Konfidenzmaß (siehe Kap. 7.1) keine sicheren Ergebnisse liefert.

Abb. 9.3 zeigt das Verhältnis von Rückweisungsrate zu Fehlerrate bei verschiedenen Konfidenzmaßen, wenn die Schwelle τ variiert wird. Die Ergebnisse beziehen sich auf die

SD-Basisdaten der Städte und Straßen bei Verwendung des normalen Standard-Wörterbuches. Werden keine Daten als unsicher zurückgewiesen, werden 13.2% der Städtenamen und 9.0% der Straßennamen falsch erkannt (vgl. Tab. 9.13: a). Verglichen werden die folgenden Konfidenzmaße in Abb. 9.3: Likelihood (Kap. 7.1: Normierung der Likelihood auf die Anzahl der Frames), Garbage (Kap. 7.4: Normierung der Wort-Likelihood auf die Likelihood eines Garbage-Modells), N-Best (Kap. 7.2: Auswertung der Häufigkeit der gleichen Erkennungsergebnisse einer 50-Best Liste).

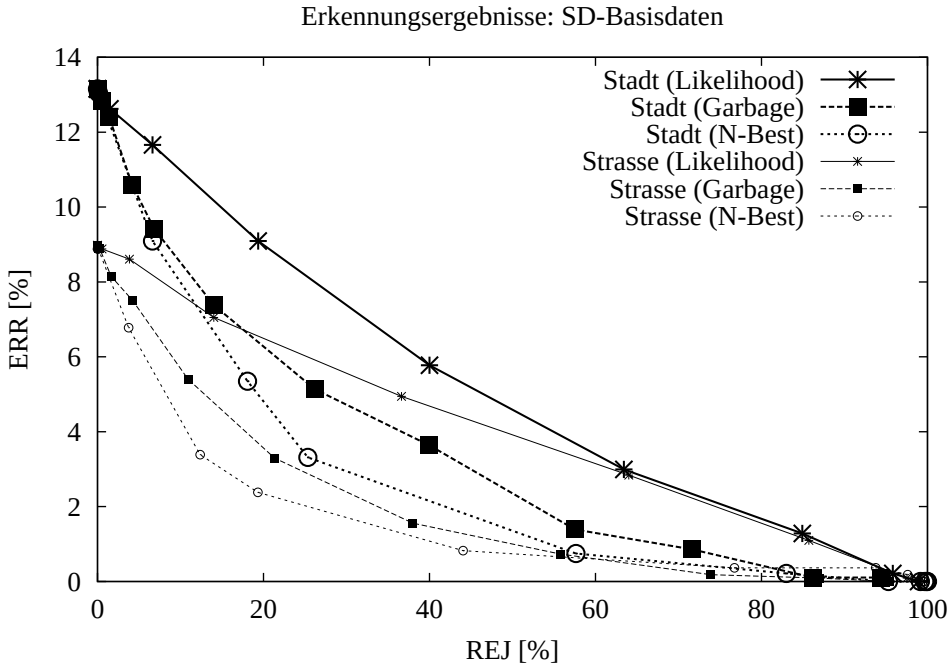


Abbildung 9.3: Rückweisungsmanagement der deutschen Basis-Adreßdaten bei unterschiedlichen Konfidenzmaßen und variierender Schwelle τ

Wie auch schon die Experimente mit den on-line Handschriftdaten (siehe Abb. 9.2) zeigten, ist die Frame-normierte Likelihood und auch das Garbage-Modell weniger als Konfidenzmaß geeignet. Die besten Ergebnisse konnten hier mit der N-Best Liste erzielt werden. So sinkt die Wort-Fehlerrate auf etwa 1%, wenn fast 60% der Daten anhand der Konfidenz-Schwelle zurückgewiesen werden. Die Unterschiede zwischen N-Best Liste und der Auswertung des 2-Best Abstandes (siehe Kap. 7.3) sind gering. Ein ausführlicherer Vergleich der Konfidenzmaße, auch für die Daten der sog. Adaptionspostämter, ist im Anhang in den Abbildungen C.2, C.3 und C.4 dargestellt. Dort ist auch zu sehen, daß sich die Rückweisungskurven der Konfidenzmaße N-Best Liste und 2-Best Abstand kaum unterscheiden.

Vergleicht man nun die Ergebnisse bei Verwendung eines vollständigen Wörterbuches und eines Zeichen-Sprachmodells unter Berücksichtigung der Zuverlässigkeit, ergibt sich ein anderes Bild. Werden die mit dem 7-Gramm erkannten Zeichensequenzen (vgl. Tab. 9.13: f)

mit dem entsprechenden Standard-Wörterbuch abgeglichen (ohne Nachbearbeitung), sinkt die Fehlerrate jew. auf etwa 1%, wenn 59.1% der Test-Städte und 63.8% der Test-Straßen zurückgewiesen werden. Die Rückweisung erfolgt hier, sobald die erkannte Zeichensequenz nicht zu 100% mit einem WB-Eintrag übereinstimmt. Dieses Ergebnis (ERR/REJ) stimmt in etwa mit der WB-Erkennung bei Verwendung der N-Best Liste überein (vgl. auch Abb. C.2). Erfolgt ein Abgleich der erkannten Zeichensequenzen mit dem falschen WB (Vertauschung von Stadt und Straße), können 99.5% der Daten als falsch zurückgewiesen werden. Das heißt, wenn eine mittels Sprachmodell erkannte Zeichensequenz mit einem gültigen Wort übereinstimmt, ist dieses Ergebnis sehr zuverlässig. In Abb. C.5 wird auch deutlich, daß mit zunehmender Größe des Wörterbuches die Leistungsfähigkeit der Konfidenzmaße, insbesondere auch die des 2-Best Abstandes, abnimmt.

Die folgenden Untersuchungen beziehen sich auf die Modellierung von *Trigraphemen* statt – wie bisher – Monographemen. Tab. 9.15 zeigt ausgewählte Ergebnisse bei semi-kontinuierlicher und kontinuierlicher Modellierung von Monographemen im Vergleich zu geclusterten Trigraphemen. Dabei wird die datengetriebene und die Entscheidungsbaum basierte Clusterung untersucht. Bei der semi-kontinuierlichen Technik kann die Erkennungsrate durch die Erhöhung der Modell-Anzahl von 77 auf 154 (bei gleichbleibender Anzahl der Gauß-Verteilungen) zwar leicht erhöht werden, diese Steigerung ist statistisch jedoch nicht signifikant. Auch bei der kontinuierlichen Technik sind die Ergebnisse der Trigrapheme bei vergleichbarer Anzahl der Gaußverteilungen nicht besser als die der Monographeme (Tab. 9.15: Mix = max. Anzahl der Gaußschen Mischverteilungen je Zustand).

Tabelle 9.15: Ergebnisse (Standard-WB) der deutschen Basis-Adressen bei unterschiedlichen Trigraphemen: Wort-Erkennungsraten in %

Methode	Stadt	Straße
Semi-Kontinuierlich (vollbesetzte Kovarianzmatrix)		
Monographeme (77 HMM, 300 Gauß)	86.8	91.0
Trigrapheme: Datengetrieben (154 HMM)	87.3	91.3
Trigrapheme: Datengetrieben (504 HMM)	87.4	90.7
Trigrapheme: Datengetrieben (729 HMM)	86.5	90.8
Kontinuierlich (diagonale Kovarianzmatrix)		
Monographeme (77 HMM, 216 Gauß, 1 Mix)	73.5	75.2
Trigrapheme: Entscheidungsbaum (513 HMM, 432 Gauß,1 Mix)	77.7	81.9
Monographeme (77 HMM, 3172 Gauß, 18 Mix)	87.5	89.7
Trigrapheme: Entscheidungsbaum (513 HMM, 7160 Gauß, 21 Mix)	86.2	89.5
Trigrapheme: Datengetrieben (467 HMM, 5613 Gauß, 21 Mix)	87.4	89.5

Aus diesen Beobachtungen läßt sich schließen, daß die Schreibweise einzelner Buchstaben im Wort nur geringfügig von seinen Nachbarn abhängt. Die Variabilität der Buchstaben bei verschiedenen Schreibern ist dagegen deutlicher ausgeprägt.

Aufgrund dieser Ergebnisse beziehen sich die weiteren Untersuchungen zur *Postamt-Adaption* wieder auf die Modellierung von Monographemen.
Bei den Adaptionsversuchen, bei denen das allgemeine Basis-Adreßerkennungssystem auf Daten spezieller Postämter (nicht spezielle Schreiber!) adaptiert wird, wird neben der Adaptionsmethode auch die benötigte Adaptionsdatenmenge untersucht. Außerdem wird die Anpassung unterschiedlicher HMM-Parameter bei den Adaptionsmethoden ML, MLLR, SLLR und MAP untersucht. Durch diese Postamt-Adaption sollen die Adreßleser die unterschiedlichen lokalen Schreibweisen berücksichtigen können, die in den spezifischen Postämtern verstärkt auftreten (man kann zeigen, daß ein großer Teil der Post, die in einem Postamt bearbeitet wird, sowohl aus der näheren Umgebung des Postamtes stammt, als auch in die nähere Umgebung geschickt wird). Da die Menge der benötigten Adaptionsdaten einen wesentlichen Kosten-Faktor darstellt, werden unterschiedliche Adaptionsvarianten getestet. Die folgende Auflistung erläutert die Datenmengen der vier Adaptions-Postämter. Die große Menge (*g*) beinhaltet dabei alle (außer den Testdaten) zur Verfügung stehenden Daten (Städte und Straßen) des jew. Postamtes, die kleine (*k*) und sehr kleine (*sk*) Adaptionsmenge betragen jeweils in etwa ein Viertel dieser Menge. Mit diesen Adaptionsmengen wird sowohl der überwachte (*üb*) als auch der unüberwachte (*unüb*: Label, die vom Basissystem erkannt wurden) Modus getestet. Eine weitere Variante ist die Einschränkung der großen Adaptionsmenge mit Hilfe von Konfidenzmaßen (*konf*: N-Best Liste, gleiche Schwelle τ je Postamt), wobei nur die sichereren Ergebnisse verwendet werden. Hier werden die automatisch erkannten Label genutzt, wie im unüberwachten Modus.

- Adaptionsmenge HRO:
  groß (g): 1733 Worte, klein (k): 434 Worte, sehr klein (sk): 111 Worte, über die Konfidenz reduziert (konf): 1495 Worte

- Adaptionsmenge STR:
  groß (g): 1504 Worte, klein (k): 376 Worte, sehr klein (sk): 94 Worte, über die Konfidenz reduziert (konf): 1313 Worte

- Adaptionsmenge HAL:
  groß (g): 1588 Worte, klein (k): 397 Worte, sehr klein (sk): 100 Worte, über die Konfidenz reduziert (konf): 1270 Worte

- Adaptionsmenge HAM:
  groß (g): 1512 Worte, klein (k): 378 Worte, sehr klein (sk): 95 Worte, über die Konfidenz reduziert (konf): 1276 Worte

In Tab. 9.16 sind die wichtigsten Adaptionsergebnisse zusammengefaßt (vgl. auch [Bra01d]). Eine detaillierte Auflistung aller durchgeführten Experimente für die vier verschiedenen Postämter zeigt Tab. C.1 im Anhang. In Tab. C.1 wird deutlich, daß sich die Adaptionsmethoden bei den verschiedenen Postämtern prinzipiell gleich auswirken.

In der Regel kann durch eine Adaption eine Steigerung der Erkennungsrate erfolgen. Die besten Ergebnisse werden erzielt, wenn mit der großen Datenmenge im überwachten Modus die HMM-Parameter $\underline{\mu}\omega A$ nach der ML-Methode adaptiert werden (Tab. 9.16: 83.1% korrekt erkannte Städte, 86.6% korrekt erkannte Straßen). Die Anpassung einzelner Parameter führt zu geringeren Erkennungsraten. Steht nur eine kleine oder sehr kleine Datenmenge zur Adaption bereit, ist die MAP-Anpassung der Gewichte oder die globale MLLR- oder SLLR-Adaption dem ML-Verfahren überlegen. Werden für die MLLR-Adaption mehrere Klassen gebildet, ändert sich die Erkennungsrate nur geringfügig und für die vier Postämter nicht in einheitlicher Weise. Beispielsweise sinkt die Erkennungsrate für HAL von 80.6% für die Städte und 86.0% für die Straßen bei einer globalen MLLR (siehe Tab. C.1:17) auf 80.4% bzw. 85.7% bei Verwendung von 11 Regressionsmatrizen (bei 26 Clustern: 80.9% bzw. 85.6%). Auffallend ist auch, daß sich die Adaption auf die Erkennung der Städte deutlicher auswirkt als auf die der Straßen.

Tabelle 9.16: Zusammenfassung der Wort-Erkennungsraten (%) der Adaptionsdaten (deutsche Adressen, SK: voll); siehe Tab. C.1 für Details

Verfahren	Durchschnitt: 4 Postämter	
	Stadt	Straße
Basissystem	**80.9**	**85.4**
SLLR, global, $\underline{\mu}$,sk,üb	81.7	85.4
MLLR, global,$\underline{\mu}$,sk,üb	81.0	85.5
MLLR, global, $\underline{\mu}$,k,üb	81.3	85.6
MAP, ω,k,üb	81.5	85.9
ML, $\underline{\mu}$,k,üb	81.2	85.2
ML, ω,k,üb	80.0	84.2
ML, $\underline{\mu}$,g,üb	81.3	85.4
ML, $\underline{\mu}\omega A$, g, üb	**83.1**	**86.6**
ML, $\underline{\mu}\omega A$, sk, üb	79.1	83.4
ML, $\underline{\mu}\omega A$, g, unüb	**82.4**	**86.1**
ML, $\underline{\mu}\omega A$, konf	**82.7**	**85.9**

Die durchschnittliche Erkennungsrate auf dem Basissystem von 80.9% für die Städte und 85.4% für die Straßen ist deutlich geringer als die der allgemeinen Basis-Testdaten (vgl.

Tab. 9.13). Dies liegt allerdings daran, daß bei den speziellen Postämtern zum einen die verwendeten Wörterbücher größer sind, und zum anderen die Daten nicht nur aus den Einzelworten (Stadt oder Straße) bestehen, sondern evtl. mit PLZ oder Hausnummer umgeben sind. Eine falsche Erkennung dieser Zusätze wird zwar nicht als Fehler gewertet, aber die automatisch vorzunehmende Abtrennung zum Stadt- bzw. Straßennamen erhöht die Fehlerquote. Beispielsweise lautet der WB-Eintrag für HAMBURG entweder [D-][*PLZ*] HAMBURG oder [D-][*PLZ*] Hamburg, wobei '[]' für wahlweise vorkommende Zeichen in den Testdaten steht.
Der Fehler kann im Durchschnitt um bis zu 11.5% relativ für die Städte und um 8.2% für die Straßen reduziert werden. Wenn diese ML-Adaption im unüberwachten Modus angewendet wird, fällt die Verringerung der Fehlerrate geringer aus (82.4% bzw. 86.1%). Die Einführung von Konfidenzmaßen führt nicht zu einer wesentlichen Verbesserung im Vergleich zur unüberwachten Adaption (siehe Tab. 9.16: letzte Zeile; vgl. auch Tab. C.1: 1ab, 8ab). Dabei muß man jedoch beachten, daß die Adaptionsmengen beim Vergleich *überwacht/ unüberwacht – Konfidenz* nicht mehr identisch sind.

Um sicherzustellen, daß die Reduktion der Fehlerquote nicht lediglich auf die größere Datenmenge zurückzuführen ist, zeigt Tab. 9.17 die Ergebnisse, die man erhält, wenn auf die Daten eines anderen Postamtes adaptiert wird.

Tabelle 9.17: Wort-Erkennungsraten (%) der Adaptionsdaten bei Adaption auf fremde Postämter (ML-Adaption: $\underline{\mu\omega A}$, üb, große Adaptionsmenge, vgl. Tab. C.1: 8)

System	STR Test-Daten					HAM Test-Daten				
	Basis-system	Adapt. STR	Adapt. HRO	Adapt. HAL	Adapt. HAM	Basis-system	Adapt. HAM	Adapt. HRO	Adapt. HAL	Adapt. STR
Stadt	81.5	84.4	81.5	83.3	81.2	79.6	81.4	79.1	80.6	79.8
Straße	85.3	85.9	84.8	85.3	85.3	85.6	87.5	86.9	87.0	86.0

Die Erkennungsraten steigen zwar in der Regel auch leicht bei einer Adaption auf die 'falschen' Daten, die höchste Erkennungsleistung wird aber bei den zum Postamt gehörigen Daten ermittelt. Bei dieser Postamt-Adaption spielt somit nicht die Schreibweise die alleinige Rolle, sondern auch das Vokabular und die Häufigkeit der vorkommenden Einträge – und somit die Zeichen der Trainigsworte in einem bestimmten Kontext – die für jedes Postamt unterschiedlich sind.
Die Unterschiede bei den erzielten Erkennungsraten sind allerdings relativ gering. So können die durchschnittlichen Erkennungsraten der vier Postämter nach einer überwachten ML-Adaption mit allen zusammengefaßten Adaptionsdaten der vier Postämter ebenfalls auf 82.8% bzw. 86.7% (vgl. Tab. C.1: 10) erhöht werden. Dieses Vorgehen ist jedoch keine

Adaption mehr, sondern eher ein lernfähiges Training. Damit zeigt sich aber, daß ein 'Weiterlernen' eines Adreß-Lesesystems während des Betriebes durchaus sinnvoll ist.

Amerikanisches Adreß-Erkennungssystem

Anhand der amerikanischen Adreß-Datenbasis (siehe Kap. 3.2.2) wurden in erster Linie verschiedene *Modellierungstechniken* miteinander verglichen und der Einfluß der *Wörterbuchgröße* (und verschiedener Schreibweisen) untersucht. Ein zweiter Aspekt ist die *Adaption* des deutschen Adreßerkennungssystems auf die amerikanischen Daten. Städte und Straßen werden wahlweise mit einem 1000er oder 6000er WB – das jeweils sowohl Städte- als auch Straßen-Namen enthält – erkannt, oder aber mit einem spezifischen 100er WB. Bei keinem dieser WB tritt ein OOV-Problem auf, da bei den 100er WB jew. das aktuell zu erkennende Wort dem WB beigefügt wurde. Dies soll den Praxisfall simulieren, in dem eine (hoffentlich korrekte) Vorauswahl des WB anhand der erkannten PLZ erfolgt.

Die ersten Experimente zeigen die Ergebnisse einer diskreten (DIS: k-means VQ; HYB-MMI: neuronaler VQ) Modellierungsstruktur (siehe Tab. 9.18) bei Verwendung der 20-elementigen Original-Merkmalvektoren. In Abb. 9.4 ist der zugehörige Verlauf der Transinformation I in Abhängigkeit der Anzahl der Iterationen des MMI-Trainings dargestellt (vgl. Kap. 4.1).

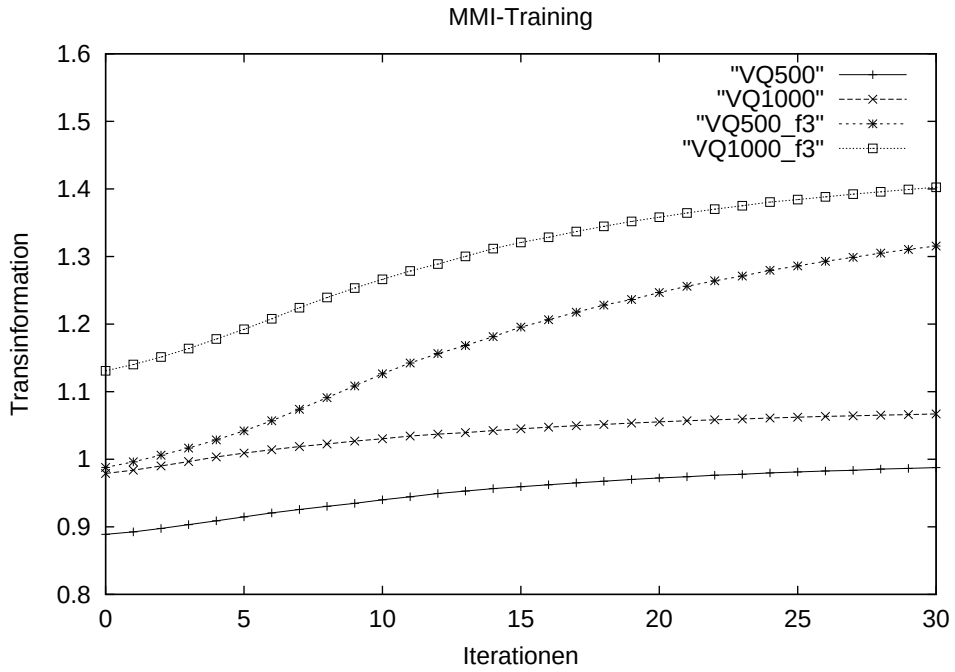


Abbildung 9.4: Iterative Optimierung der Transinformation bei unterschiedlichen Codebuchgrößen (VQ= 500 oder 1000) und unterschiedlicher Anzahl zu quantisierender Merkmalvektoren (1 oder 3 benachbarte Frames); amerikanische Adreßdatenbasis

Aus der Grafik ist abzulesen, daß die Transinformation im Verlauf stetig ansteigt bis zur Sättigung. Außerdem ist festzustellen, daß die Transinformation von der Codebuchgröße (VQ) und der Frame-Anzahl ($f3$: 3 benachbarte Frames) abhängt und die Zunahme der Transinformation bei Quantisierung von drei Frames deutlich größer ausfällt. Diese Eigenschaften können als typisch angesehen werden.
Die Abhängigkeiten sind teilweise auch bei den Wort-Erkennungsraten in Tab. 9.18 abzulesen, wobei eine höhere Transinformation nicht gleichbedeutend ist mit einer höheren Erkennungsrate; ausschlaggebend ist in erster Linie die Zunahme der Transinformation, was auch durch andere Versuche bestätigt wurde. Vergleicht man das diskrete mit dem MMI-hybriden System, so ist die Fehlerreduktion durch das MMI-Training bei der Verwendung von drei Frames am größten. Hier kann mit einer Codebuchgröße von 500 eine Erkennungsrate von 82.9% (bzw. 84.0%) erreicht werden (1000er WB, siehe Tab. 9.18).

Tabelle 9.18: Wort-Erkennungsraten (%) mit dem 1000er WB: diskrete HMM-Struktur (ein VQ); 20-elementige Merkmalvektoren (ohne LDA), Transkription $_{1,2}$ s.u., vgl. Abb. 9.4

	VQ 500		VQ 1000	
Methode	DIS	HYB-MMI	DIS	HYB-MMI
1 Frame	73.5_1	75.6_1	71.9_1	73.8_1
3 Frames	74.0_1 / 75.7_2	82.9_1 / 84.0_2	76.9_1	82.0_1

Anders als beim deutschen System ist hier die Art der Transkription der Wort-Label anders zu wählen, wie sich in den Experimenten herausstellte. Deshalb wurden hier drei Varianten untersucht, die anhand eines Beispiels (Label: NEW YORK) erläutert werden sollen:

1. NEW YORK: N E W Y O R K oder N e w Y o r k
2. **NEW YORK:** N E W Y O R K oder N e w Y o r k oder n e w y o r k
3. NEW YORK: N E W Y O R K oder n E w y O R k oder etc.

Die erste Variante entspricht dem deutschen System und wird bei den amerikanischen Erkennungstests mit dem Index 1 gekennzeichnet. Die zweite Variante wird bei diesen Versuchsreihen als Standard angesehen, da sie, wie auch Tab. 9.18 zeigt, die besseren Ergebnisse liefert. Diese Art der Schreibweise – evtl. keine Großbuchstaben am Wortanfang – hat sich bei der Durchsicht der Daten bestätigt. Die dritte Variante – beliebige Schreibweise bzgl. Groß- und Kleinbuchstaben – führte zu geringeren Erkennungsquoten, da die Anzahl der möglichen Alternativen bei der Dekodierung stark ansteigt.

Die Berücksichtigung der lokalen Umgebung durch Verwendung von benachbarten Frames führt zu eindeutig höheren Erkennungsraten. Folglich ist eine Merkmaltransformation,

die die Nachbarschaft berücksichtigt durchaus sinnvoll. Bei einem Blick in die Spracherkennung zeigt sich die Berechnung von Δ- und $\Delta\Delta$-Merkmalen (Differenz des aktuellen Vektors zu seinen Nachbarn) als häufig benutzte Merkmalvektoren. So kann beispielsweise auch hier die Erkennungsrate des 1000er WB (NN-VQ 500, 3 Frames) von 84.0% auf 85.8% erhöht werden, wenn die Δ- und $\Delta\Delta$-Merkmale hinzugezogen werden und diese getrennt mit drei separaten 500er NN-VQs quantisiert werden. Eine andere Möglichkeit, die lokale Umgebung stärker zu berücksichtigen, ist die LDA-Transformation von mehreren benachbarten Frames. Dieses Verfahren wurde hier, wie auch beim deutschen Erkennungssystem, bevorzugt, da zum einen die Erkennungsergebnisse weiter gesteigert werden konnten (vgl. Tab. 9.19 und 9.20) und zum anderen diese Datentransformation auch in der Praxis eingesetzt wird.
Tabelle 9.19 zeigt den Vergleich zwischen der hybriden MMI-Modellierung und einer semikontinuierlichen Modellierungstechnik (Gaußsche Mischverteilungen mit diagonaler oder voll besetzter Kovarianzmatrix), die jeweils auf den LDA-Merkmalen beruhen. Zusätzlich wurde hier der Einfluß der WB-Größe untersucht. Die Ergebnisse mit der semikontinuierlichen (SK) Modellierung und diagonaler Kovarianzmatrix sind deutlich schlechter als die der beiden Vergleichssysteme. Allerdings ist auch die Parameteranzahl im Vergleich zur Modellierung mit voller Kovarianzmatrix geringer. Überzeugend arbeitet das MMI-diskrete System, welches eine Erkennungsrate von 89.2% für das 1000er WB erzielt und damit geringfügig besser ist als das hier vorgestellte SK-System mit voller Kovarianzmatrix. Jedoch sind für dieses geringfügig bessere Ergebnis trotz LDA-Transformation mit Dimensionsreduktion auf 40 Elemente wiederum drei Frames für die Quantisierung erforderlich, wodurch der Berechnungsaufwand steigt.

Tabelle 9.19: Wort-Erkennungsraten (%) bei unterschiedlichen Modellierungstechniken: jew. LDA auf 3 benachbarten Frames mit Dimensionsreduktion auf D Elemente

Methode	HYB-MMI D=40	SK: diag D=30		SK: voll D=30
	VQ 500 (3 Frames)	200 Gauß	400 Gauß	400 Gauß
1000er WB	89.2	80.6	83.8	88.9
100er Städte-WB	95.5	–	–	95.0
100er Straßen-WB	96.0	–	–	96.2
6000er WB	81.6	–	–	80.4

Für Adreß-Erkennungssysteme mag die Erkennungsrate von knapp 90% relativ gering erscheinen, in der Praxis wird jedoch ein wesentlich kleineres WB verwendet. So steigt die Erkennungsrate der Städte-Namen auf 95.5% und die der Straßen-Namen auf 96.0%

bei einem 100er WB (HYB-MMI). Auch die Nachbearbeitung der Erkennungsergebnisse, z.B. eine Verknüpfung der Wahrscheinlichkeiten für die erkannte PLZ und die erkannte Stadt, konnte hier aus praktischen Gründen (keine vollständigen Daten, Datenschutz) nicht erfolgen. Außerdem wurde auch die gleiche Bedeutung einer Erkennung (z.B. 'PO' oder 'P.O. Box') nicht berücksichtigt, sondern Verwechslungen ggf. als Fehler gewertet. Diese Nachbearbeitungen sind zwar stark praxisrelevant, erfordern aber auch spezielles Wissen über bestimmte Parameter (z.B. was sind unterschiedliche Schreibweisen für ein und dieselbe Adresse und was sind wirklich zwei unterschiedliche Adressen?), die an die jeweiligen (kommerziellen) Systeme angepaßt werden müssen. Der Vergleich mit einem 6000er WB (Simulation der Erkennung ohne PLZ-Wissen) zeigt eine Erkennungsrate von nur noch 81.6%. Diese Fehlerraten begründen auch den Einsatz von Rückweisungsmöglichkeiten mittels Konfidenzmaßen (siehe deutsches Adreß-Erkennungssystem).

Um den Einfluß *länderspezifischer Schreibvarianten* zu testen, wurde ein auf deutschen (vgl. Ergebnisse in Tab. 9.12) und ein auf amerikanischen Daten trainiertes Basissystem mit den gleichen Parametern erstellt (SK, 300 Gauß, volle Kovarianzmatrix). Das deutsche Basissystem wurde dann mit Hilfe der *ML-Adaption* an die amerikanischen Daten angepaßt. Die erzielten Ergebnisse für die amerikanischen Testdaten sind in Tab. 9.20 beschrieben.
Hinsichtlich der Erkennungsrate von 87.9% bzw. 63.0% wird ein deutlicher Unterschied zwischen amerikanischen und deutschen Adreßdaten sichtbar. Diese Differenz kann nach der Adaption deutlich reduziert werden. 1886 Adaptions-Worte entsprechen einem Achtel der amerikanischen Trainingsdatenbasis, 7544 Worte entsprechen der Hälfte.

Tabelle 9.20: Wort-Erkennungsraten (%) amerikanischer Adressen vor und nach Adaption des deutschen Systems: SK (volle Kovarianzmatrix, 300 Gauß); jew. LDA auf 3 benachbarten Frames mit Dimensionsreduktion auf 30 Elemente

Methode	amerikan. System	deutsches System	ML-Adaption mit 1886 Worten	ML-Adaption mit 7544 Worten
1000er WB	87.9	63.0	76.2	79.1

Für ein endgültiges Fazit aus diesem Versuch sind die Test- und Trainings-Datenmengen von nur zwei verschiedenen Ländern zu gering. Die Ergebnisse lassen jedoch vermuten, daß es möglich und wirtschaftlich ist, ein Basis-Erkennungssystem für lateinische Schrift durch Adaption an Schreibweisen verschiedener Länder anzupassen.

9.4 Dokumenterkennung

Anhand der SEDAL-Dokumentendatenbasis sollen zum einen verschiedene Methoden zur *Vorverarbeitung* und *Merkmalextraktion* untersucht werden, deren Ergebnisse mit veröffentlichten Ergebnissen anderer Institute verglichen werden können (siehe auch [Bra00c, Bra00b, Bra01c]). Einen Schwerpunkt bildet die Untersuchung von diskreten *Modellierungstechniken*. Eine deutliche Verbesserung des Erkennungssystem durch die Verwendung eines neuronalen Vektorquantisierers (hybrid-MMI) im Vergleich zum Standard k-means VQ wird dabei ersichtlich. Ein weiterer Schwerpunkt ist die Erkennung mit Hilfe von Backoff Zeichen *N-Grammen* unterschiedlicher Kontexttiefe, da bei dieser Datenbasis das verwendete Vokabular völlig unbekannt ist.

Systemspezifikation

Das Erkennungssystem besteht aus etwa 80 verschiedenen HMMs, ein HMM je Zeichen (Buchstaben, Zahlen, Sonderzeichen, siehe Anhang D). Dabei werden in der Regel sieben Zustände je Zeichen verwendet mit Ausnahme schmaler Buchstaben wie ‘i’ (fünf Zustände; ggf. weniger als sieben Merkmalvektoren vorhanden) und kurzer Sonderzeichen wie ‘.,’ etc. (drei Zustände). Ein zusätzliches ‘Short Space’-Modell mit einem Zustand wird bei diesem System für eventuelle Buchstabenzwischenräume benutzt. Dieses HMM kann wahlweise übersprungen werden (nicht lineare Modellstruktur). Das Training wird standardmäßig mit dem Baum-Welch-Algorithmus durchgeführt, die Erkennung mit einem Viterbi- oder Stack-Dekoder-Verfahren.
Die Versuche beziehen sich jeweils auf eine diskrete HMM-Struktur (vgl. Kap. 2.2.2), was bedeutet, daß die Merkmalvektoren zuvor quantisiert werden. Diese Quantisierung erfolgt auf jeweils drei benachbarten Frames mit einem k-means- oder MMI-VQ.
Die Test- und Trainingsdaten bestehen aus qualitativ schlechten vollständigen Dokumenten (vgl. Kap. 3.3) in englischer Sprache, deren Wortgrenzen (Umrandung jedes einzelnen Wortes und die Position im Dokument) allerdings bekannt sind. Um den Vergleich mit den in [Sch98] präsentierten Ergebnissen durchführen zu können und Fehler, die durch eine falsche Segmentierung entstehen, ausschließen zu können, wurden diese vorgegebenen Wortgrenzen (nicht die Zeichengrenzen) hier für die Schrifterkennung benutzt. Vorkommende Sonderzeichen wie Anführungsstriche und Satzendezeichen werden jew. dem nächsten Wort zugeordnet (z.B. ‘discussed.’). Da die Dokumente sehr spezielle Themen behandeln, wird in der Regel eine Erkennung ohne Wörterbuch durchgeführt. Die meisten Ergebnisse beziehen sich auf die Zeichen-Akkuratheit. So kann auch ein Vergleich mit kommerzieller OCR-Software erfolgen, bei der ein WB, wenn man es verwenden würde, einen unbekannten Faktor darstellt.

Experimentelle Untersuchungen

Eine erste Untersuchungsreihe betrifft die *Vorverarbeitung* und *Merkmalextraktion*, die hier jedoch keinen Anspruch auf Vollständigkeit erheben soll. Es geht lediglich um ein 'relativ gutes' Basissystem, auf dem die Modellierungsaspekte getestet werden können. Als Referenzsystem werden die von [Sch98] auf der SEDAL-Datenbasis erzielten und präsentierten Ergebnisse herangezogen. Bei Verwendung von HMMs und NN konnte dort auf einem ähnlichen Testdatensatz (11655 Zeichen) eine Zeichenerkennungsrate von durchschnittlich 86.3% erreicht werden. Zum Vergleich erzielten kommerzielle OCR-Systeme, wie in [Sch98] dokumentiert, Zeichenerkennungsraten zwischen 57% und 75% (siehe Tab. 3.1).

Im wesentlichen wurden in dieser Arbeit die folgenden Aspekte zur Merkmalgewinnung untersucht (vgl. auch [Tri96]), die zu den klassischen Mustererkennungsmethoden gehören: Vorverarbeitung mit oder ohne Skelettierung; Merkmale der sliding-window Technik basierend auf einer unterabgetasteten Bitmap, zweidimensionalen Zentralmomenten oder einer DCT-Transformation (siehe Anhang A); zusätzliche Merkmale wie z.B. die aktuelle Höhe der Schrift. Die in Kap. 3.3 beschriebenen Merkmale sind die, mit denen im Rahmen der Experimente die kleinsten Fehlerraten erzielt wurden.
Die Skelettierung der Dokumente führte im Durchschnitt zu jeweils deutlich schlechteren Erkennungs-Ergebnissen als eine Vorverarbeitung ohne Skelettierung. Dieses Ergebnis kann man bei Betrachtung der Abb. 9.5 nachvollziehen. Bei vielen Fonts (verschnörkelt, Comic-Font, etc.) trägt die unterschiedliche Strichdicke die entscheidende Information (siehe auch [Tri96]). Nach einer Skelettierung kann dies zu verbundenen Linien führt, die das Zeichen verfälschen. Der gleiche Effekt tritt auch bei verschmierten Dokumenten (Kopie, Fax) auf, weshalb i.d.R. keine Skelettierung bei dieser Datenbasis durchgeführt wurde.

Abbildung 9.5: Originalbild und Bild nach Skelettierung

Der Vergleich von Bitmap (ggf. überlappend), Momenten und DCT-Koeffizienten (unterschiedliche Anzahl) ergab bei gleicher Anzahl von Merkmalen (Größe des Merkmalvektors) eine leicht bessere Erkennungsrate bei der Wahl der DCT mit den ersten 10 Koeffizienten (jew. normiert auf den Nullten Koeffizienten). Eine weitere Steigerung der Erkennungsrate konnte in Anlehnung an die Merkmale der on-line Erkennung durch einen zusätzlichen Merkmalvektor (siehe Kap. 3.3, z.B. aktuelle Höhe, etc.) erzielt werden. Die anhand dieser

Merkmale erzielte Zeichenerkennungsrate beträgt 86.1% bei Modellierung mit diskreten (k-means VQ) HMMs, wie in Tabelle 9.21 dargestellt wird, und ist somit bereits dem in [Sch98] präsentierten System gleichwertig.

Die folgenden Ergebnisse basieren auf der in Kap. 3.3 beschriebenen Merkmalextraktion, wobei jeweils drei benachbarte Teil-Merkmalvektoren $\underline{x}$ (entsprechend den zwei Merkmaltypen: DCT der Bitmap und zusätzliche Merkmale) von zwei separaten Codebüchern der Größe 400 und 200 quantisiert werden. Tabelle 9.21 zeigt die Zeichen-Erkennungsraten ohne Verwendung eines Wörterbuches und ohne Sprachmodell. Dabei werden die diskrete und die MMI-hybride Modellierungstechnik verglichen. Die Erkennungsergebnisse der Dokumente 'neurofax' und 'maintenance' sind am schlechtesten, was auch dem optischen Eindruck nach Abb. 3.8 entspricht. Hinzu kommt, daß das Dokument 'maintenance' viele unterstrichene Worte enthält. Die Fehlerrate konnte bei der MMI-hybriden Technik um 28% relativ von 13.9% auf 10.0% gesenkt werden. Dies entspricht einer Wort-Erkennungsrate des hybriden Systems von 68.2% (siehe Tab. 9.21 unten). Häufige Fehler, die auftreten, sind Verwechslungen sehr ähnlicher Zeichen und insbesondere von 'I;l;1' ('großes I; kleines l; eins'), 'O;0' ('O; Null') und '.;,' ('Punkt; Komma'). Diese Zeichen können jedoch von Font zu Font so stark variieren, daß auch der menschliche Betrachter Probleme hat, diese Zeichen ohne Kontextwissen (kein Vokabular, keine Satzgrammatik) richtig zu lesen. Werden diese Fehler nicht berücksichtigt, steigt die Zeichen-Erkennungsrate auf 91.5% im Durchschnitt und die Wort-Erkennungsrate auf 72.8%.

Tabelle 9.21: Zeichen-Erkennungsraten (ACC in %) bei unterschiedlichen Modellierungstechniken ohne Wörterbuch und ohne Sprachmodell

Dokument	Anzahl der Zeichen	ACC diskret	ACC hybrid-MMI	ACC mit: I=l=1,O=0,.=, hybrid-MMI
intel	396	91.4	95.5	96.5
neurofax	2061	75.9	83.3	84.8
precept	1923	87.2	90.2	93.1
vision	2643	92.6	95.5	96.4
maintenance	2979	81.2	85.7	87.7
james	2592	91.6	93.9	94.1
insgesamt	12594	86.1	90.0	91.5
Worte insgesamt	2218	60.0	68.2	72.8

Eine andere Fehlerquelle, die die Verwechslung ganzer Worte bzw. Zeichensequenzen zur Folge hat, gründet auf einer falschen Bestimmung der oberen und / oder unteren Basislinie. Dieses Problem tritt insbesondere dann auf, wenn einzelne Worte nur aus Zeichen ohne

Ober- und Unterlänge bestehen (‘come, a, name’ im Vergleich zu ‘US, 10’) und die Fontgröße unbekannt ist. Mittels Histogramm wird festgestellt, daß die obere Wortgrenze mit der geschätzten Kernhöhe übereinstimmt. Die Entscheidung, um welche Kennlinie es sich handelt, kann nur aus dem Kontext des Dokumentes bestimmt werden. Deshalb wurden hier im Zweifelsfall die Nachbarwörter in der gleichen Zeile berücksichtigt. Diese Methode erreicht jedoch bei unterschiedlichen Fontgrößen je Dokument und Briefköpfen ihre Grenzen. Wenn, wie bei dieser Dokumentdatenbank, die Position der einzelnen Worte vorgegeben ist, kann leicht die zugehörige Zeile eines Wortes ermittelt werden. Im realen Fall müssen jedoch sowohl die Einzelworte als auch die Zeilen im Dokument automatisch bestimmt werden. Dies kann beispielsweise über ein Histogramm (erst Zeilen und dann Spalten) oder auch über die Connected Components Analyse (CCA, siehe z.B. [Gon93]) erfolgen, wobei die verbundenen Komponenten (Connected Components, benachbarte Pixel gleicher Farbe) anhand ihrer Lage und Ausdehnung zu Worten und Zeilen zusammengefaßt werden können. Die CCA-Methode wurde hier verwendet um bei Zweifelsfällen, wie oben erläutert, die Kernhöhe korrekt bestimmen zu können. Dazu wurden die geschätzten Basislinien der zugehörigen Zeile berücksichtigt.

Die Effizienz von *Sprachmodellen* auf der Basis von Zeichen N-Grammen wird in Tab. 9.22 deutlich. Hier kann eine Zeichen-Erkennungsrate von 97.1% bei Verwendung eines 5-Grammes erzielt werden. Dies entspricht einer Fehlerreduktion von 70% relativ im Vergleich zur Zeichenerkennung ohne Sprachmodell.

Tabelle 9.22: Zeichen-Erkennungsraten (ACC in %) mit MMI-hybriden HMMs und Sprachmodellen unterschiedlicher Kontexttiefe (ohne WB)

Dokument	2-Gramm pp=18.3	3-Gramm pp=11.7	5-Gramm pp=5.8	7-Gramm pp=5.4
intel	97.5	97.7	98.7	98.2
neurofax	92.0	93.6	95.5	95.7
precept	95.3	96.4	97.1	97.0
vision	96.7	97.5	97.9	98.1
maintenance	92.1	93.6	96.5	96.5
james	95.9	96.3	97.9	97.7
insgesamt	94.5	95.5	97.1	97.1

Wie aus Tabelle 9.22 ersichtlich wird, hat eine weitere Erhöhung der Kontexttiefe des Sprachmodells keine weitere Steigerung der Erkennungsleistung zur Folge. Auch mit einem 7-Gramm beträgt die Zeichen-Erkennungsrate 97.1% im Durchschnitt. Der Grund dafür liegt

ganz einfach in der durchschnittlichen Wortlänge von 5.7 Zeichen pro Wort. Dieser Zusammenhang läßt sich auch von der Perplexität pp dieser Testdokumente auf dem jeweiligen Sprachmodell ablesen. Die Perplexität dieser englischen Dokumente auf dem englischen N-Gramm sinkt bis zur Kontexttiefe von $N = 5$ deutlich auf $pp = 5.8$ und verändert sich beim 7-Gramm nur noch unwesentlich. Betrachtet man die Perplexitätsbestimmung im Detail, bedeutet dies, daß beim 5-Gramm ca. 94% der Zeichensequenzen von den im Sprachmodell enthaltenen 5er Sequenzen direkt vorkommen, beim 7-Gramm sind es nur noch 61% der 7er Sequenzen und die restlichen Worte/ Buchstabenfolgen werden auf N-Gramme mit kleinerer Kontexttiefe abgebildet (Backoff N-Gramm). Diese Untersuchungen zeigen auch, daß trotz der relativ großen Text-Datenmenge von ca. 4 Millionen zufällig ausgewählten englischen Worten, mit denen die Sprachmodelle trainiert wurden (vgl. Kap. 5.3), bei dieser Dokumentdatenbasis bereits 61 mal auf ein Unigramm zurückgegriffen werden mußte; d.h. im Test treten 2er-Zeichensequenzen auf, die in den Sprachmodell-Trainingsdaten nicht vorkamen. Daraus lassen sich zwei Schlußfolgerungen ziehen: 1.) für eine weitere Verbesserung der Erkennungsrate muß die Trainingsdatenmenge für das Sprachmodell erhöht werden. 2.) auch ein Wörterbuch ohne OOV gibt es quasi nicht, vor allem dann nicht, wenn das Thema des zu erkennenden Dokumentes unbekannt und sehr speziell ist (womit sich wiederum die Notwendigkeit von Sprachmodellen bestätigt, ggf. in Kombination mit Wörterbüchern).

Trotzdem soll zu Vergleichszwecken eine WB-basierte Erkennung dieser Dokumente auf einem speziell erstellten Wörterbuch (1115 Einträge, kein OOV) stattfinden. Dabei wurden alle Wörter der Testdokumente einschließlich der Sonderzeichen (vorgegebene Wortgrenzen) zu einem Wörterbuch zusammengefügt, um einen direkten Vergleich mit der N-Gramm-Erkennung zu ermöglichen. So ergeben sich beispielsweise WB-Einträge als ‘notice:’ oder ‘Glutamate,’ (Kombination von Wort und Sonderzeichen). Wie erwartet, ist die Erkennungsleistung mit kleinem WB deutlich besser als bei Verwendung eines 5-Grammes (siehe Tab. 9.23).

Tabelle 9.23: Wort-Erkennungsraten (ACC in %) mit MMI-hybriden HMMs

Dokument	Anzahl Worte	5-Gramm	1115er WB
intel	79	96.2	98.7
neurofax	340	84.7	97.7
precept	295	89.2	97.6
vision	476	93.5	98.5
maintenance	519	91.9	97.3
james	505	94.6	97.2
insgesamt	2218	91.6	97.7

Die Wort-Erkennungsrate des 5-Grammes liegt bei 91.6%, die des 1115er WB bei 97.7%. Berücksichtigt man jedoch, daß eine N-Gramm basierte Erkennung gewissermaßen eine Erkennung mit unbegrenztem Vokabular darstellt und einige Worte dieser Dokumentdatenbank (Abkürzungen, Namen, Adressen, medizinische und technische Fachausdrücke) in einem Standard-WB in der Regel nicht vorkommen, ergibt sich ein positives Bild für die Sprachmodelle. Der Einfluß des Sprachmodells wird auch deutlich, wenn man die Ergebnisse mit einem 'falschen' 5-Gramm, einem auf deutschen Worten trainiertem Modell, betrachtet. Hier sinkt die Zeichen-Akkuratheit auf 82.4%, was einer Wort-Erkennungsrate von nur 50.6% entspricht.

9.5 Vergleich und Auswertung

In den vorangegangenen Abschnitten wurden Aspekte der Vorverarbeitung und Merkmalextraktion (bzw. Transformation), der Modellierungstechnik für HMM-basierte Erkennungssysteme, der Sprachmodellierung mit Zeichen N-Grammen, der Modellierung von Kontextmodellen, der Bestimmung von Konfidenzmaßen, und der Adaptionsmethoden experimentell anhand verschiedener Datenbasen untersucht. Zusätzlich können die Ergebnisse von on- und off-line Handschrifterkennungssystemen und von schreiberab- und schreiberunabhängigen Systemen miteinander verglichen werden.
Zusammenfassend können die folgenden Ergebnisse festgehalten werden:

- die Erkennungsraten eines on-line Handschrifterkennungssystems sind bei identischen Daten höher als die des entsprechenden schreiberabhängigen off-line Systems
- aufwendige Vorverarbeitungsverfahren und Normierungsmethoden sind bei schreiberabhängigen Systemen nicht zwingend notwendig; eine robuste Bestimmung der oberen und unteren Basislinie ist allerdings notwendig, wenn die Merkmale darauf beruhen
- bei schreiberunabhängigen Systemen sind die Erkennungsraten im Durchschnitt nach einer Normierung höher; die Varianz zwischen den Schreibern wird kleiner; nach einer Schreiber-Adaption läßt der Einfluß der Normierung auf die Erkennungsrate nach
- erwartungsgemäß ist die Fehlerquote eines schreiberunabhängigen Systems höher als die eines schreiberabhängigen Systems
- die Wahl der besten Modellierungstechnik hängt stark von der Art der extrahierten Merkmale ab und von der Anzahl der verfügbaren Trainingsdaten: bei der diskreten Technik sind weniger Parameter zu schätzen und somit weniger Daten erforderlich
- hier konnte für die schreiberabhängigen Systeme der Wort-Fehler um 40% relativ für die on-line Daten und um 20% relativ für die off-line Handschriftdaten gesenkt wer-

den, wenn statt des k-means VQ die MMI-hybride Technik eingesetzt wird; die TP-hybride Modellierungstechnik wirkt sich hier nicht positiv aus; bei den SEDAL-Daten reduziert sich der Zeichen-Fehler um 28% relativ bei der MMI-hybriden Technik

- die kontinuierliche bzw. semi-kontinuierliche (und TP-hybride) Modellierungstechnik war nur bei den Adreß-Daten überlegen, bei denen im Vergleich weniger, aber dafür 'breitere' Merkmale extrahiert werden, die einen Buchstaben abdecken; die kontinuierliche Ausgabeverteilung bietet Vorteile bei den Adaptionsverfahren

- unabhängig von der Art der Daten ist die Berücksichtigung mehrerer benachbarter Merkmalvektoren (LDA oder Quantisierung) von Vorteil

- die Verwendung von Zeichen N-Grammen anstatt vollständigen Wörterbüchern führt zu einer deutlichen Erhöhung der Fehlerrate; wenn das Vokabular der Testdaten nicht bekannt ist, sind N-Gramme gegenüber den OOV-Wörterbüchern im Vorteil

- die Verwendung von Backoff Zeichen N-Grammen führt bis zu einer Kontexttiefe von 5-7 zu einer deutlichen Verringerung des Fehlers; je spezifischer die Sprachmodelle trainiert werden können, je höher ist die Fehlerreduktion

- bei den SEDAL-Dokumenten kann der Zeichen-Fehler (ohne WB) um 70% relativ mit Hilfe von 5-Grammen reduziert werden, bei den schreiberabhängigen off-line Daten und den handgeschriebenen Adressen um bis zu 50% relativ

- es besteht eine gute Möglichkeit der Nachbearbeitung der Erkennungsfehler einer N-Gramm Erkennung, da häufig nachvollziehbare Fehler entstehen

- die Modellierung von Kontextmodellen (Trigraphemen) für die off-line Erkennung ist nur im schreiberabhängigen Modus von leichtem Vorteil (stark parameterabhängig); eine deutliche Verbesserung, wie sie in der Spracherkennung häufig erfolgt, kann nicht festgestellt werden; bei den Adreßdaten konnte mittels Trigraphemen keine Verbesserung erzielt werden

- als Konfidenzmaß hat sich die Auswertung einer N-Best Liste oder die Verwendung eines 2-Best Abstandes der Likelihoods bewährt; diese Methoden können bei einem vollständigen Wörterbuch die Verwechslungsmöglichkeiten innerhalb des WB besser berücksichtigen

- die erfolgreichste Adaptionsmethode, sowohl für die Schreiber-Adaption (on-line) als auch für die Postamt-Adaption (off-line) ist das Nachtrainieren mittels des ML-Verfahrens; es sollten je nach Adaptionsdatenmenge nicht nur die Mittelwertvektoren sondern auch die Gewichte und evtl. die Übergangsmatrizen angepaßt werden

- eine Schreiber-Adaption (on-line) mit nur 6 Worten führt mittels des MAP-Verfahrens bereits auf eine Fehlerreduktion von 10% relativ, bei 100 Worten reduziert sich der Fehler um etwa 50% nach dem ML-Verfahren; somit ist eine Anwendung von Adaptionsverfahren z.B. für PDAs auch im Hinblick auf die Benutzerfreundlichkeit sinnvoll
- die Fehlerquote eines schreiberabhängigen Systems ist geringer als die eines adaptierten schreiberunabhängigen Systems (allerdings spielt hierbei auch die Datenmenge eine Rolle)
- bei einer Postamt-Adaption kann der rel. Fehler hier um bis zu 16% reduziert werden; auch länderspezifische Schreibweisen können nachtrainiert werden
- die Adaptionsmethoden wie MLLR und SLLR, die in der Spracherkennung zu deutlichen Verbesserungen führen, bewirken hier im Durchschnitt nur eine minimale Fehlerreduktion; die Bedingung, daß zwei Cluster mit Gaußverteilungen, die bei einem allgemeinen System benachbart sind, auch bei einem speziellen (schreiberabhängigen) System benachbart sein müssen, ist hier nicht zwangsläufig erfüllt
- auch eine fehlerhafte unüberwachte Adaption führt zu einer Verringerung der Fehlerquote
- eine unüberwachte Adaption mit mittels Konfidenzmaßen eingeschränkter Trainingsmenge erzielt nur eine relativ geringe Verringerung der Fehlerrate; die zur Verfügung stehenden Datenmengen reichen für eine weitergehende Untersuchung nicht aus

Viele Resultate sind auch in den Veröffentlichungen zu dieser Arbeit (siehe [Bra99a] bis [Bra02c]) nachzulesen.

9.6 Kapitelzusammenfassung

Neben der Systembeschreibung wurden im vorliegenden Kapitel die Versuchsreihen zu den unterschiedlichen hier vorgestellten Datenbasen beschrieben: die schreiberab- und schreiberunabhängige on-line Handschrift-Datenbasis, die schreiberab- und schreiberunabhängige (Adressen) off-line Handschrift-Datenbasis und die öffentliche SEDAL-Dokumentendatenbasis. Es wurden die in dieser Arbeit vorgestellten Modellierungstechniken und Adaptionsverfahren aus den Kap. 4 bis 8 untersucht, wobei die Charakteristika der jeweiligen Daten berücksichtigt wurden. Abschließend erfolgte eine Auswertung der erzielten Ergebnisse.
Als konkrete praxisrelevante Erkenntnis läßt sich aufgrund der Experimente festhalten, daß eine Schreiber-Adaption schon mit sehr wenigen Worten erfolgreich ist (siehe PDA), daß

ein unüberwachtes Nachtrainieren auch für Adreßlesesysteme (Postautomatisierung) sinnvoll sein kann, und daß Zeichen N-Gramme (ggf. in Kombination mit Wörterbüchern) die Fehlerrate bei thematisch unbekannten Texten (wozu auch unstrukturierte Adressen gehören können) deutlich verringern können.

Kapitel 10

Zusammenfassung

In dieser Arbeit wurden verschiedene Aspekte der automatischen on- und off-line Handschrifterkennung sowie der Erkennung von gedruckten Dokumenten vorgestellt und untersucht. Eine Gemeinsamkeit der verschiedenen Erkennungssysteme ist die Verwendung von Hidden Markov Modellen (HMM), da diese sich für die Erkennung kursiver zusammenhängender Schrift – wozu in gewisser Weise auch gedruckte Dokumente, die in schlechter Qualität vorliegen und somit häufig ineinander übergehende Zeichenfolgen besitzen, gehören – eignen. Das Thema dieser Arbeit ist die Einzelworterkennung, wobei die Schwerpunkte auf verschiedenen Modellierungstechniken und Adaptionsverfahren liegen.

Nachdem in Kapitel 1 ein Überblick über diese Arbeit gegeben wurde und die Anwendungsbereiche der Schrifterkennung erläutert wurden, befaßt sich Kapitel 2 mit den Grundlagen der automatischen Schrifterkennung. Dazu gehören neben der Vorverarbeitung und der Merkmalextraktion auch die Klassifikation, die hier auf Hidden Markov Modellen und Sprachmodellen basiert. Die Theorie der Hidden Markov Modelle mit einem Schwerpunkt auf der Modellierung der Emissionswahrscheinlichkeiten wurde erläutert. Diese können eine diskrete, kontinuierliche oder semi-kontinuierliche Struktur besitzen. Ebenfalls wird ein Einblick in die Sprachmodellierung gegeben, die im Gegensatz zu den Hidden Markov Modellen nicht die extrahierten Merkmale der Schrift berücksichtigt, sondern die Grammatik bzw. linguistisches Wissen.

Ausgehend von diesen grundlegenden Betrachtungen zur Schrifterkennung wurden in Kapitel 3 die in dieser Arbeit verwendeten Datenbasen zur schreiberab- und -unabhängigen on- und off-line Handschrifterkennung und die öffentlich zugängliche SEDAL-Dokumentendatenbasis vorgestellt. Der Schwerpunkt in diesem Kapitel liegt auf der für die jeweiligen Daten charakteristischen Vorverarbeitung und den Merkmalextraktionsverfahren. Diese Datenbasen wurden gewählt um neben den Modellierungstechniken und Adaptionsverfahren auch konkrete Vergleiche zwischen identischen on- und off-line Daten und adaptierten schreiberunabhängigen und den entsprechenden schreiberabhängigen Systemen durchführen zu können. Die Adreß-Datenbasis von Siemens Dematic bietet außerdem den

Vorteil von praxisrelevanten Experimenten. Bei der SEDAL-Datenbasis können als Vergleich Ergebnisse anderer Institute herangezogen werden. Eine Standard-Handschrift-Datenbasis zur Evaluierung, wie es sie in der Spracherkennung gibt, war zu Beginn dieser Arbeit nicht verfügbar. Die Vorverarbeitung der Daten beschränkt sich im wesentlichen auf die Normierung der Zeilen- und Zeichenneigung und die Größennormierung. Für die Merkmalextraktion wird die sogenannte Sliding-Window Technik verwendet.

Aufbauend auf den Grundlagen der Hidden Markov Modelle wurden in Kapitel 4 zwei hybride Modellierungstechniken vorgestellt. Hybrid bedeutet in diesem Zusammenhang die Kombination von Hidden Markov Modellen mit neuronalen Netzwerken. Die Vorteile beider Verfahren, den segmentierungsfreien Erkennungsansatz bzw. die gute Separierbarkeit von Zeichen, können genutzt werden. Das MMI-hybride Verfahren basiert auf der Maximierung der Transinformation und verwendet einen neuronalen Vektorquantisierer. Dies führt zu einer diskreten HMM-Struktur. Das zweite hybride Verfahren nutzt ein neuronales Netzwerk zur Schätzung der Posterior Wahrscheinlichkeit der Zeichen. Die HMM-Struktur ist dann näherungsweise semi-kontinuierlich.

Ein wesentlicher Schwerpunkt dieser Arbeit ist die Modellierung und Erkennung mit Sprachmodellen. Im Gegensatz zum üblichen Verständnis von Sprachmodellen – der statistischen Modellierung von Wortfolgen – wurde in Kapitel 5 die Bestimmung von Sprachmodellen auf Zeichenebene vorgestellt. Hier wurden statistische Modelle, die Backoff Zeichen N-Gramme, als Alternative zur Wörterbuch basierten Erkennung beschrieben. Diese modellieren nicht mehr die Satz-Grammatik, sondern ermöglichen eine Erkennung mit quasi unbegrenztem Vokabular. Dies ist immer dann wichtig, wenn das Thema und somit das verwendete Vokabular des Test-Textes unbekannt ist. Zeichen N-Gramme berücksichtigen die Häufigkeit von bestimmten Zeichensequenzen. Daraus folgt, daß abhängig von der Sprache (deutsch, englisch) und vom Thema (z.B. Adressen, allgemeiner Text) unterschiedliche Sprachmodelle trainiert werden müssen. Neben der Bestimmung und Verwendung der Zeichen N-Gramme mit sehr hoher Kontexttiefe wurden in dieser Arbeit auch die zugehörigen ASCII-Trainingstexte vorgestellt.

Kontextmodelle, die Trigrapheme, wurden in Kapitel 6 als Analogon zu den Triphonen der Spracherkennung eingeführt. In der Theorie geht man davon aus, daß sich die Schreibweise von Zeichen innerhalb eines Wortes abhängig von den aktuellen Nachbarzeichen ändert. Die daraus resultierende Änderung der Merkmalvektoren wird von Trigraphemen berücksichtigt. Werden Trigrapheme anstatt der üblichen Monographeme verwendet, erhöht sich die Anzahl der Hidden Markov Modelle extrem, da zu jedem zu modellierenden Zeichen auch der linke und rechte Nachbar berücksichtigt wird. Um diese Modelle trotzdem robust trainieren zu können, wurden zwei Cluster-Methoden beschrieben, mit denen verschiedene Trigrapheme des gleichen zentralen Monographems zusammengefaßt werden können: die datengetriebene Clusterung und die Entscheidungsbaum basierte Clusterung.

In Kapitel 7 wurden fünf verschiedene Konfidenzmaße für eine HMM-Erkennung vorgestellt: die normierte Likelihood je Frame, die Auswertung von N-Best Listen, der Zwei-Best Abstand bezogen auf die Likelihood der Ergebnisse, eine Normierung der Likelihood mittels Garbage-Modellen und die Normierung auf die Likelihood einer zwanglosen Zeichenerkennung ohne Wörterbuch. Bei einer HMM-basierten Erkennung sind Konfidenzmaße, die die Sicherheit oder Vertrauenswürdigkeit eines erkannten Ergebnisses angeben, notwendig, da das Erkennungsergebnis selbst keine Auskunft darüber gibt. Die Likelihood je Frame dient als Referenzmaß, die anderen Konfidenzmaße sind sicherer aber auch aufwendiger zu berechnen. Sie unterscheiden sich im wesentlichen dadurch, daß nur die N-Best Liste und der Zwei-Best Abstand von der Wahl des Wörterbuches abhängt. Konfidenzmaße wurden in dieser Arbeit sowohl zur Sicherheitsbestimmung mit der Möglichkeit zur Rückweisung von Daten genutzt als auch zur Einschränkung von Adaptionstrainingsdaten im unüberwachten Modus.

Der zweite Schwerpunkt dieser Arbeit ist die Anwendung von Adaptionsverfahren, die in Kapitel 8 vorgestellt wurden. Die Anpassung der HMM-Parameter mittels des ML-, MLLR-, SLLR- oder MAP-Verfahrens erfolgt anhand weniger Adaptionsdaten als für ein normales Training benötigt werden würde. Das MLLR- und SLLR-Verfahren handhabt das Problem der geringen Adaptionsdaten mittels Clusterung, das MAP-Verfahren berücksichtigt die a priori Wahrscheinlichkeit der Zeichen. In dieser Arbeit wurde die Adaption eines schreiberunabhängigen on-line Systems auf je einen speziellen Schreiber vorgestellt. Die erforderliche Datenmenge liegt dabei bei nur 6 bis 100 Worten, womit diese Verfahren beispielsweise für PDAs besonders geeignet sind. Außerdem wurden die Adaptionsverfahren auch in einem eher unüblichen Zusammenhang erprobt: der Postamt-Adaption. Es wurde gezeigt, daß sich durch adaptives Lernen der aktuell vorkommenden Adreßdaten die Erkennungsleistung eines Lesesystems in speziellen Postämtern steigern läßt. Die Variabilität in der Schreibweise, die sowohl von Schreiber zu Schreiber, aber auch je nach Region (Postämter, Länder) schwankt, kann automatisch berücksichtigt werden. Dabei spielt bei der Postamt-Adaption jedoch nicht nur die Schreibweise, sondern auch eine jeweils andere, spezielle Häufigkeitsverteilung der Adressen eine Rolle.

Im anschließenden Kapitel 9 wurden die in dieser Arbeit allgemeingültigen Systemspezifikationen und die Ergebnisse, die mit den verschiedenen Datenbasen erzielt wurden, präsentiert. Dabei wurden alle zuvor genannten Modellierungstechniken und Adaptionsverfahren untersucht und die Experimente ausgewertet.

Die maximal erzielte Wort-Erkennungsrate von 97.2% (30k-Wörterbuch) für das schreiberabhängige on-line Erkennungssystem liegt damit deutlich höher als die vergleichbaren Ergebnisse der off-line Handschrifterkennung mit 89.3% (30k WB) im schreiberabhängigen Modus. Für das schreiberunabhängige on-line Erkennungssystem wurde eine maximale Wort-Erkennungsrate von 87.0% (2.2k WB) erreicht. Die Wort-Erkennungsrate für die hand-

schriftlichen Adressen liegt bei ca. 88.9% (WB $<$ 1k), wobei berücksichtigt werden muß, daß in der Praxis kleinere Wörterbücher verwendet werden. Auf der SEDAL-Datenbasis konnte ohne WB und ohne Sprachmodell eine Zeichen-Erkennungsrate von 90.0% erzielt werden. Die hybride Modellierungstechnik ist bei einer relativ kleinen Datenmenge den kontinuierlichen Systemen überlegen. Im Vergleich zum üblichen diskreten HMM mit einem k-means Vektorquantisierer konnte die MMI-hybride Technik deutlich höhere Erkennungsraten erzielen. Die Verwendung von Backoff Zeichen N-Grammen konnte sich bei unbekannten Texten (evtl. OOV) gegenüber den fest vorgegebenen Wörterbüchern durchsetzen. Im Vergleich zur reinen Zeichen-Erkennung ohne Wörterbuch konnte so der Fehler um bis zu 70% relativ reduziert werden. In Zukunft wäre eine Kombination beider Verfahren wünschenswert. Die Kontextmodelle konnten in dieser Arbeit nicht überzeugen, da die erzielte Fehlerreduktion gering und stark parameterabhängig war. Nur bei schreiberabhängigen Systemen konnte eine Verbesserung gegenüber dem Einsatz von Monographemen erzielt werden. Bei vollständigen Wörterbüchern ergaben die Konfidenzmaße, die mehrere Wörterbuch-Einträge berücksichtigt haben (N-Best Liste, 2-Best Abstand), die höchste Zuverlässigkeit. Bei den Adaptionsverfahren hat sich das ML-Verfahren durchgesetzt, bei sehr kleinen Datenmengen das MAP-Verfahren. Die maximale Fehlerreduktion lag hier bei ca. 50% relativ nach einer Schreiber-Adaption. Auch bei der Postamt-Adaption oder einem unüberwachten Nachtrainieren der Modelle konnte der Fehler reduziert werden. Die MLLR-Adaption führte im Durchschnitt nur zu geringfügig besseren Ergebnissen.

Weitere Aufgaben für die Zukunft betreffen die Verbesserung der Sprachmodelle – auch für Wortsequenzen bzw. Sätze – durch Berücksichtung linguistischen Wissens und deren Kombination mit Wörterbüchern, und eine längerfristige Untersuchung von ständigen unüberwachten Adaptionen während des Betriebes (hier ist aber eine größere Datenmenge erforderlich). Zu vielen Zielstellungen können analoge Aufgaben in der Spracherkennung herangezogen werden, wobei jedoch die spezifischen Charakteristika der Schrift beachtet werden müssen. Nicht alle Ideen, die in der Spracherkennung erfolgreich sind (z.B. Triphone, MLLR-Adaption), lassen sich für die Schrifterkennung ebenso erfolgreich umsetzen, da sich die Eigenschaften von Schrift und Sprache doch deutlich unterscheiden. Nur wenn die Erkennungsraten für Handschrift noch weiter erhöht werden können und die Anwendung benutzerfreundlicher wird, kann die Akzeptanz von automatischen Erkennungssystemen für die Mensch-Maschine-Kommunikation steigen.

Eine wirklich holistische Erkennung unter Berücksichtigung des inhaltlichen Zusammenhangs und Deutung von Schrift, wie der Mensch sie ganz selbstverständlich durchführt, steht noch aus.

Anhang A

Verwendete Formeln

Im folgenden sind einige Standardverfahren und -begriffe, die in dieser Arbeit verwendet werden, zusammenfassend dargestellt.

Scherung

Scherung eines Bildes $b(x, y)$ um einen Winkel α:

$$b(x, y) \rightarrow b(x_n, y_n): \quad y_n = y, \quad x_n = x - y \tan \alpha \tag{A.1}$$

Entropie

Entropie einer diskreten Zufallsvariablen V (v_i statistisch unabhängig):

$$H(V) = -\sum_i P(v_i) \cdot logP(v_i) \tag{A.2}$$

Verbundentropie:

$$H(V, U) = -\sum_i \sum_j P(v_i, u_j) \cdot logP(v_i, u_j) \leq H(V) + H(U) \tag{A.3}$$

Bedingte Entropie:

$$H(V|U) = H(V, U) - H(U) \tag{A.4}$$

Levenshtein-Distanz

Die Levenshtein-Distanz $d_{lev}(string1, string2)$ gibt an, wieviele Vorgänge (austauschen, einfügen, löschen) mindestens notwendig sind, um $string1$ in $string2$ zu überführen.

DCT

DCT-Koeffizienten $S(u)$ für einen D-dimensionalen Merkmalvektor $\underline{x}$, dessen Elemente die Werte $f(x)$ annehmen:

$$S(u) = \frac{C(u)}{2} \sum_{x=0}^{D} f(x) \cos \frac{(2x+1)u\pi}{2D} \tag{A.5}$$

$$\text{mit:} \quad C(u) = \frac{1}{\sqrt{(2)}} \quad \text{falls} \quad u = 0, \qquad C(u) = 1 \quad \text{falls} \quad u > 0$$

LDA

Lineare Diskriminanzanalyse (siehe auch [Fuk90]):
Gegeben: insgesamt N Merkmalvektoren $\underline{x}$ zu K verschiedenen Klassen (N_k Vektoren in Klasse k)
Gesucht: LDA-Transformationsmatrix M_{lda};
mit: $\underline{\mu}_k$ - Mittelwertvektor der Klasse k; $\underline{\mu}_{ges}$ - gesamter Mittelwertvektor; Λ - Eigenwertmatrix

$$\underline{x}_{neu} = M_{lda} \cdot \underline{x} \qquad \text{mit Eigenwertproblem:} \quad S_W^{-1} S_B M_{lda}^T = M_{lda}^T \Lambda \tag{A.6}$$

Within-Class Streumatrix S_W:

$$S_W = \sum_{k=1}^{K} \frac{N_k}{N} \cdot S_k \qquad \text{mit:} \quad S_k = \frac{1}{N_k} \sum_{i=1}^{N_k} (\underline{x}_{i,k} - \underline{\mu}_k)(\underline{x}_{i,k} - \underline{\mu}_k)^T \tag{A.7}$$

Between-Class Streumatrix S_B:

$$S_B = \sum_{k=1}^{K} \frac{N_k}{N} \cdot (\underline{\mu}_k - \underline{\mu}_{ges})(\underline{\mu}_k - \underline{\mu_{ges}})^T \tag{A.8}$$

Lösung (falls S_W singulär): EV(·) - Eigenvektormatrix, EW(·) - Eigenwertmatrix

$$M_{lda} = M_1 \cdot M_2 \tag{A.9}$$

$$M_1 = (\text{EW}(S_W))^{-0.5} \cdot \text{EV}(S_W) \tag{A.10}$$

$$M_2 = \text{EV}(M_1 \cdot S_B \cdot M_1^T) \tag{A.11}$$

Anhang B

Online-Datenbasis

Dieser Abschnitt zeigt weitere Einzelheiten für die on-line Handschrifterkennungssysteme: Beispiele von Einzelzeichen zur Initialisierung der Hidden Markov Modelle und eine vollständige Liste der verwendeten und zugelassenen Zeichen.

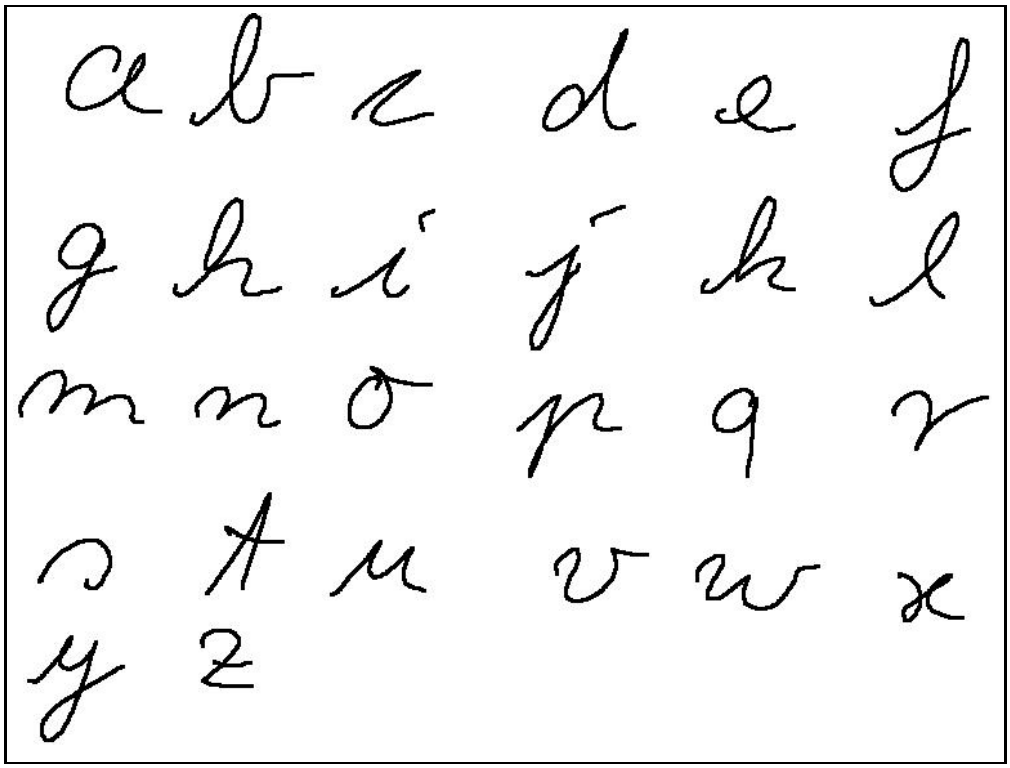


Abbildung B.1: Einige Einzelzeichen eines Schreibers (on-line)

Vollständige Zeichenliste:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z Ä Ö Ü ä ö ü ß 0 1 2 3 4 5 6 7 8 9 . , : ; “ ! ? ( ) % - + & ´ / =<> *Blank*

Anhang C

SD-Adreß-Datenbasis

Neben weiteren Beispielen dieser Adreß-Datenbasis von Siemens Dematic, werden hier auch die für das deutsche und amerikanische Erkennungssystem zugelassenen Zeichen aufgelistet und detaillierte Ergebnisse zu den Adaptionsversuchen und der Effizienz verschiedener Konfidenzmaße angegeben.

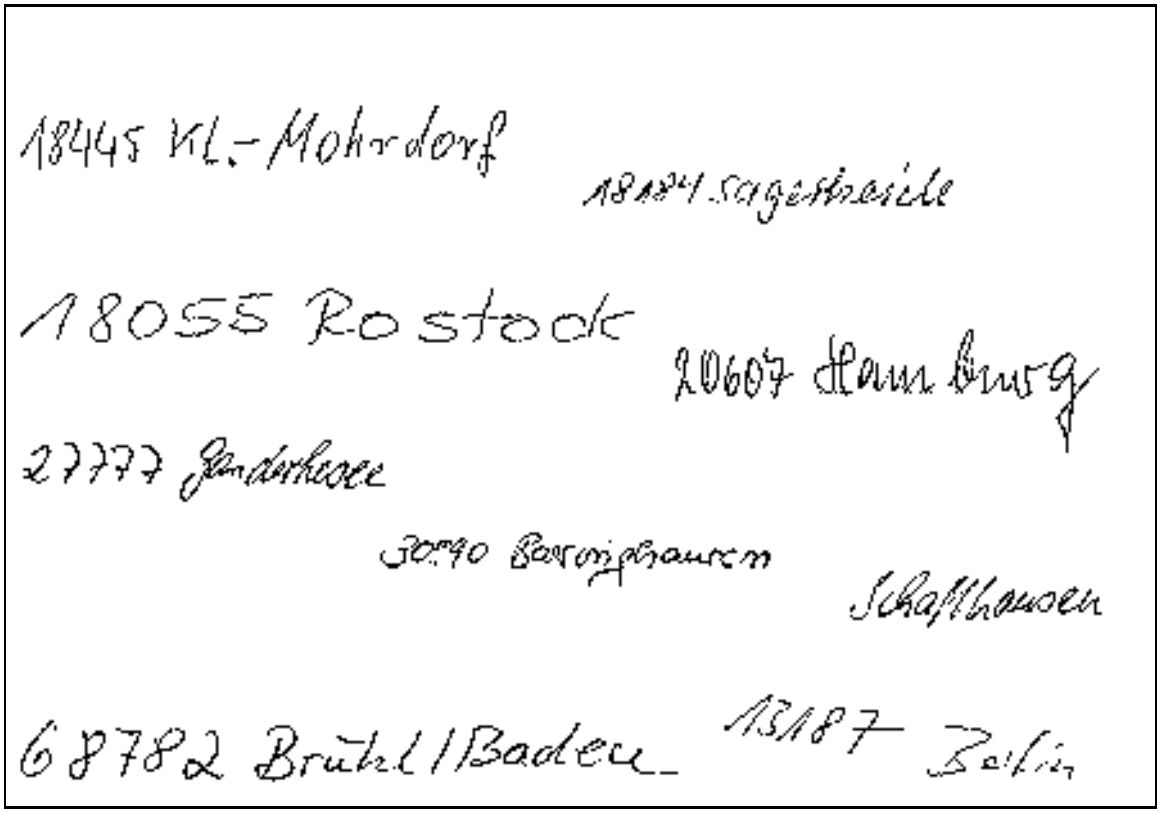


Abbildung C.1: Deutsche Adressen (SD): Beispiele der Städtenamen (i.d.R. mit PLZ)

Vollständige Zeichenliste der deutschen Datenbasis:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z Ä Ö Ü ä ö ü ß 0 1 2 3 4 5 6 7 8 9 . , - / ( ) ´ *Blank*

Vollständige Zeichenliste der amerikanischen Datenbasis:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 . , - + & ´ *Blank*

Die Graphen in Abb. C.2 zeigen Rückweisungskurven für die deutschen Basisdaten:

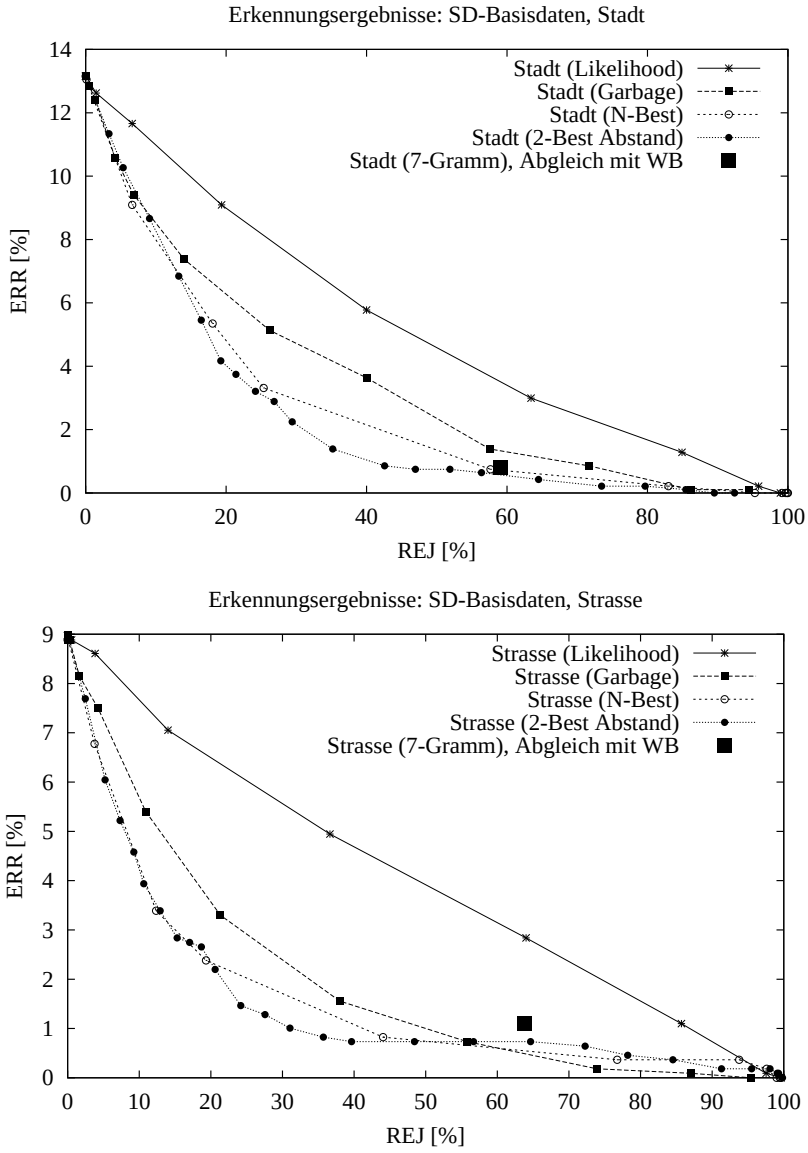


Abbildung C.2: Rückweisungsmanagement der deutschen Adreßdaten bei unterschiedlichen Konfidenzmaßen: Erkennung der Städte und der Straßen mit dem Standard-WB

Die Auswirkung der Konfidenzmaße für die Adaptionspostämter HRO und STR wird in Abb. C.3 deutlich.

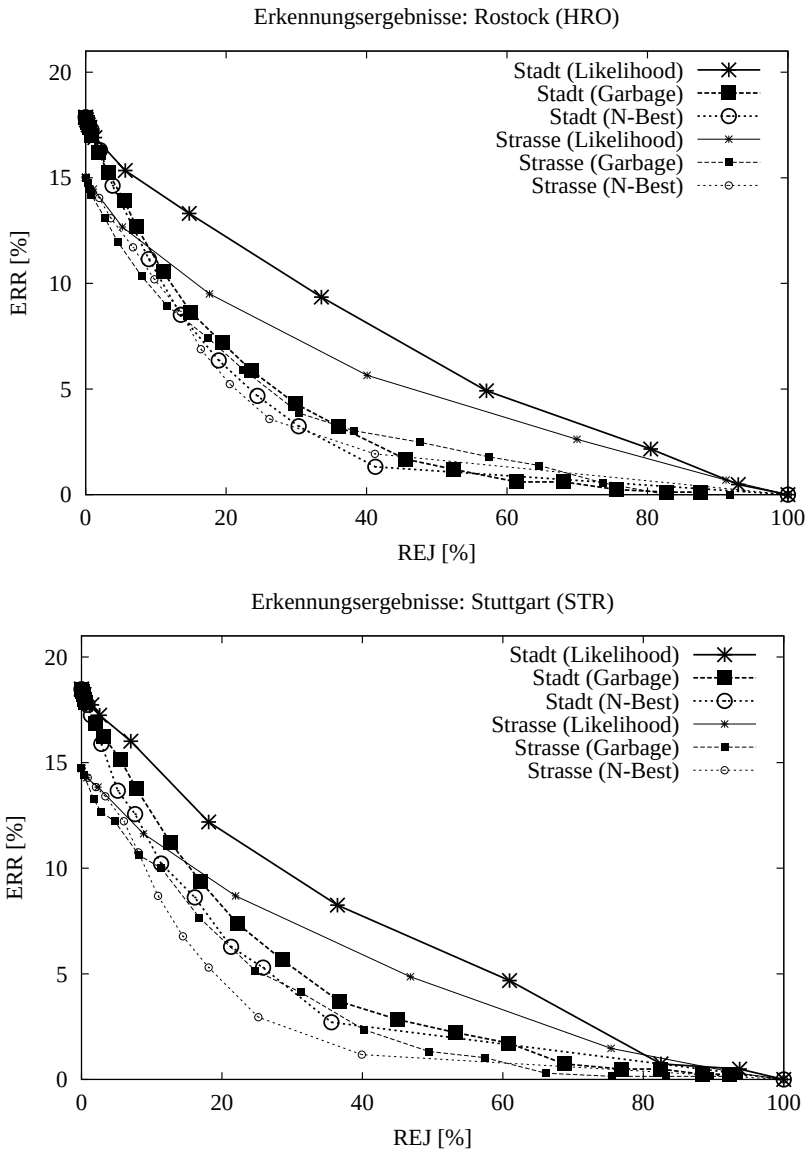


Abbildung C.3: Rückweisungsmanagement der deutschen Adreßdaten bei unterschiedlichen Konfidenzmaßen: ‘Adaptionspostämter’ HRO, STR

Die Ergebnisse zur Untersuchung der Konfidenzmaße für die Adaptionspostämter HAM und HAL sind in Abb. C.4 dargestellt.

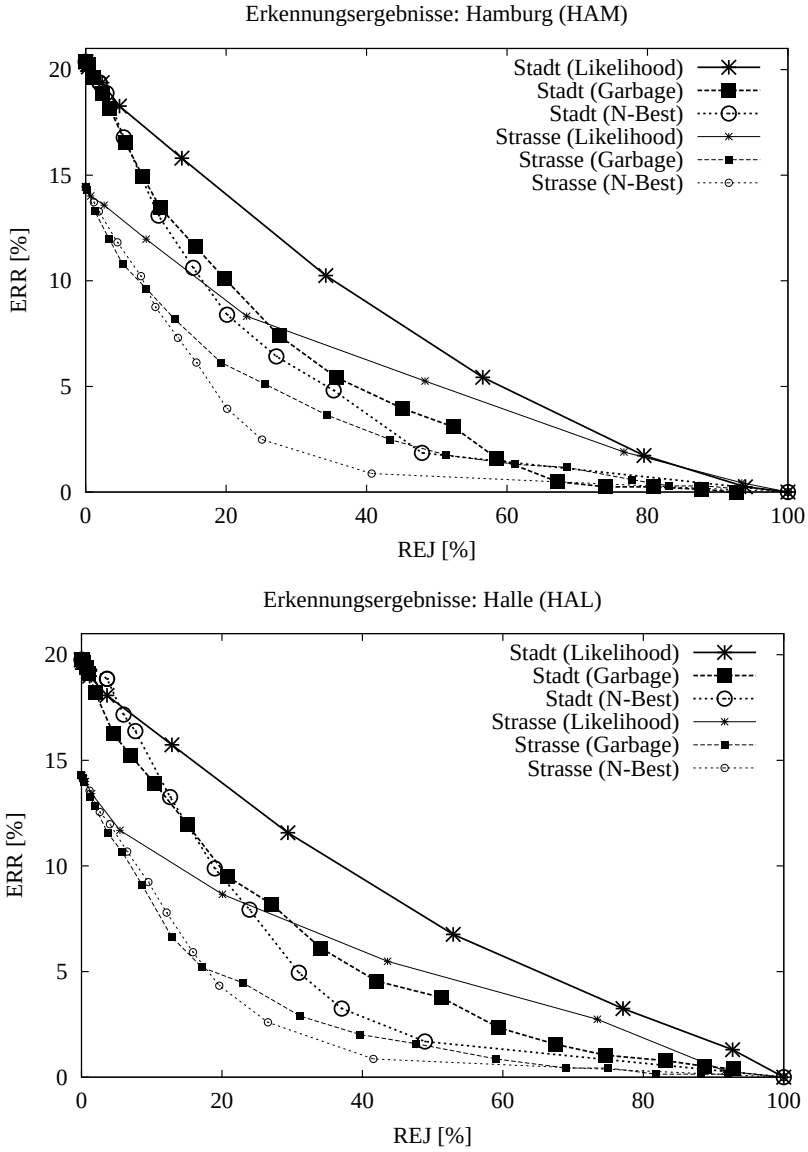


Abbildung C.4: Rückweisungsmanagement der deutschen Adreßdaten bei unterschiedlichen Konfidenzmaßen: ‘Adaptionspostämter’ HAM, HAL

Abbildung C.5 zeigt Rückweisungs- und Fehlerrate der deutschen Adressen bei einer Erkennung mit dem 20000 Worte umfassenden Wörterbuch.

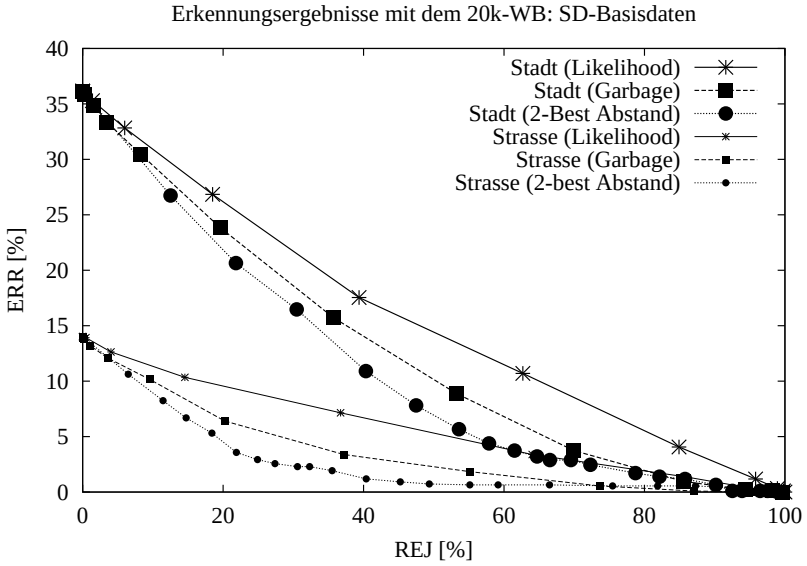


Abbildung C.5: Rückweisungsmanagement der deutschen Adreßdaten bei unterschiedlichen Konfidenzmaßen: Erkennung der Städte und der Straßen mit dem 20k-WB

Die folgende Tabelle C.1 zeigt die Ergebnisse nach Adaptionsversuchen für jedes der 4 Postämter HRO, STR, HAL und HAM getrennt. Mit $TOPN$ werden jeweils die Erkennungsraten angegeben, die die besten N Ergebnisse berücksichtigen. Der Zusatz ‘4Post’ bedeutet, daß die Daten aller Postämter zusammen für den entsprechenden Versuch zum Nachtrainieren benutzt wurden (vgl. Kap. 9.3.2).

Tabelle C.1: Wort-Erkennungsraten (%) der Adaptionsdaten bei verschiedenen Adaptionsverfahren (deutsche Adressen, SK: volle Kovarianzmatrix)

	Daten HRO		Daten STR		Daten HAL		Daten HAM	
Verfahren	Stadt	Straße	Stadt	Straße	Stadt	Straße	Stadt	Straße
Basissystem	**82.1**	**85.0**	**81.5**	**85.3**	**80.2**	**85.7**	**79.6**	**85.6**
Basissystem TOP 5	*86.1*	*89.1*	*85.6*	*89.1*	*85.4*	*88.9*	*84.4*	*89.3*
Basissystem TOP 50	*91.5*	*93.4*	*91.0*	*93.7*	*91.9*	*95.8*	*91.2*	*93.7*
1 ML,$\underline{\mu}$,g,üb	83.0	85.4	82.1	85.4	80.0	85.3	79.9	85.3
1a ML,$\underline{\mu}$,g,unüb	82.7	85.1	82.0	85.3	80.6	85.3	79.8	85.1
1b ML,$\underline{\mu}$,konf	83.0	85.0	82.3	85.4	80.1	85.6	79.8	85.1
2 ML, $\underline{\mu}$,k,üb	82.4	85.3	82.1	84.8	79.8	85.0	80.3	85.8
3 ML, ω,g,üb	83.8	85.7	82.8	86.3	81.3	86.6	80.4	86.3
3a ML,ω,g,unüb	83.1	85.5	82.6	84.8	80.6	85.9	79.6	85.4
4 2xML,ω,g,üb	83.2	85.4	82.8	85.4	80.9	85.7	80.5	85.7
5 ML,ω,k,üb	82.6	83.3	81.3	84.5	77.2	84.7	78.8	84.4
6 ML,A,g,üb	82.4	85.7	83.3	86.5	80.2	87.2	81.0	86.7
7 ML,$\underline{\mu\omega}$,g,üb	83.7	86.0	83.4	85.1	81.5	86.7	81.2	86.6
7a ML,$\underline{\mu\omega}$,g,unüb	83.0	85.4	82.8	84.5	82.1	85.1	79.6	86.4
8 **ML,$\underline{\mu\omega A}$,g,üb**	**84.2**	**85.5**	**84.4**	**85.9**	**82.2**	**87.5**	**81.4**	**87.5**
8) TOP 5	*87.9*	*90.4*	*87.7*	*89.0*	*87.3*	*91.2*	*86.2*	*91.1*
8a ML,$\underline{\mu\omega A}$,g,unüb	83.2	85.8	83.7	85.0	82.1	86.3	80.4	87.3
8b ML,$\underline{\mu\omega A}$,konf	84.1	85.3	84.0	85.6	81.9	86.2	80.6	86.6
9 ML,$\underline{\mu\omega A}$, sk, üb	80.6	83.9	81.3	83.5	77.5	83.7	76.8	82.6
10 **ML,4Post,$\underline{\mu\omega A}$,4g,üb**	83.8	86.2	83.6	85.7	82.2	87.5	81.4	87.5
10) TOP 5	*88.0*	*90.1*	*86.8*	*90.3*	*87.3*	*91.1*	*87.3*	*91.4*
11 2xML,4Post,$\underline{\mu\omega A}$,4g,üb	84.1	86.5	83.6	86.0	81.4	87.6	81.1	87.7
12 ML,$\underline{\mu\omega A}$,g(Stadt),üb	82.6	85.7	84.6	85.9	81.8	86.2	81.5	86.4
13 ML,4Post,$\underline{\mu\omega A}$,4g(Stadt),üb	83.9	86.0	83.9	86.6	81.7	86.3	81.5	86.9
14 **ML,4Post,$\underline{\mu\omega A}$,4k,üb**	84.1	86.1	83.1	86.2	80.1	86.3	81.2	86.9
15 MAP:ω,Basis+3,g,üb	83.6	86.0	83.0	86.3	81.3	86.3	80.5	86.0
16 MAP:ω,Basis+5,k,üb	83.8	85.3	82.4	85.7	79.5	86.0	80.3	86.4
17 MLLR, global, $\underline{\mu}$,g,üb	82.4	85.3	82.3	85.9	80.6	86.0	80.5	85.8
18 MLLR, global, $\underline{\mu}$,k,üb	82.5	85.4	82.3	85.6	80.1	85.7	80.3	85.8
19 MLLR, global,$\underline{\mu}$,sk,üb	82.3	85.3	81.8	85.0	79.8	85.9	80.0	85.8
20 SLLR, global, $\underline{\mu}$,sk,üb	84.1	85.5	81.9	85.4	80.5	85.7	80.1	85.1

Anhang D

SEDAL-Datenbasis

Für die Dokumenterkennung sind im folgenden die trainierten und für das System zugelassenen Zeichen aufgelistet. Weitere Beispiele aus den Bereichen ‘künstlich erzeugter Text’ und ‘echtes Dokument’ der verwendeten öffentlich zugänglichen SEDAL-Datenbasis (siehe [Sch98]) sind hier dargestellt.

Vollständige Zeichenliste:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 . , : ; “ ! ? ( ) % - + & ´ / =<> $ [ ] # * *Blank*

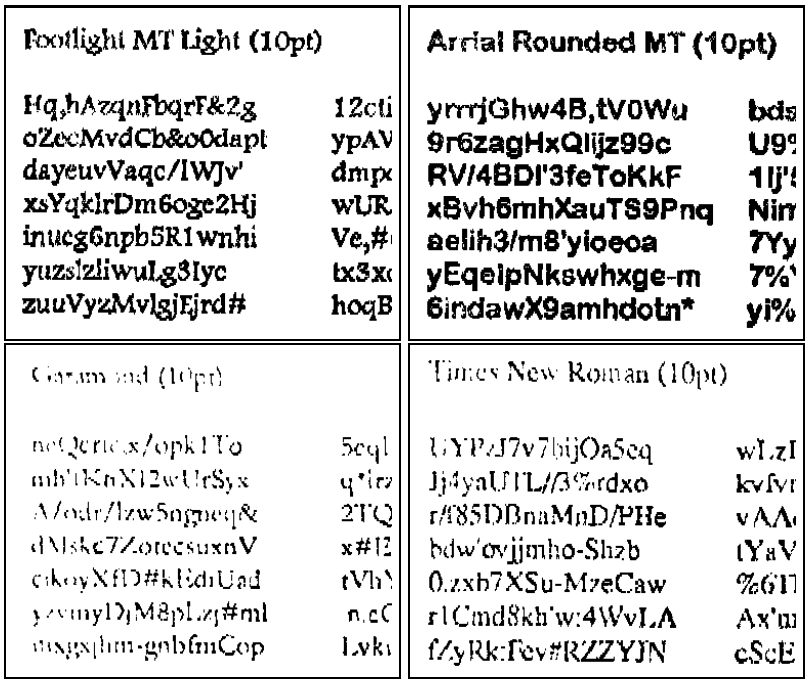
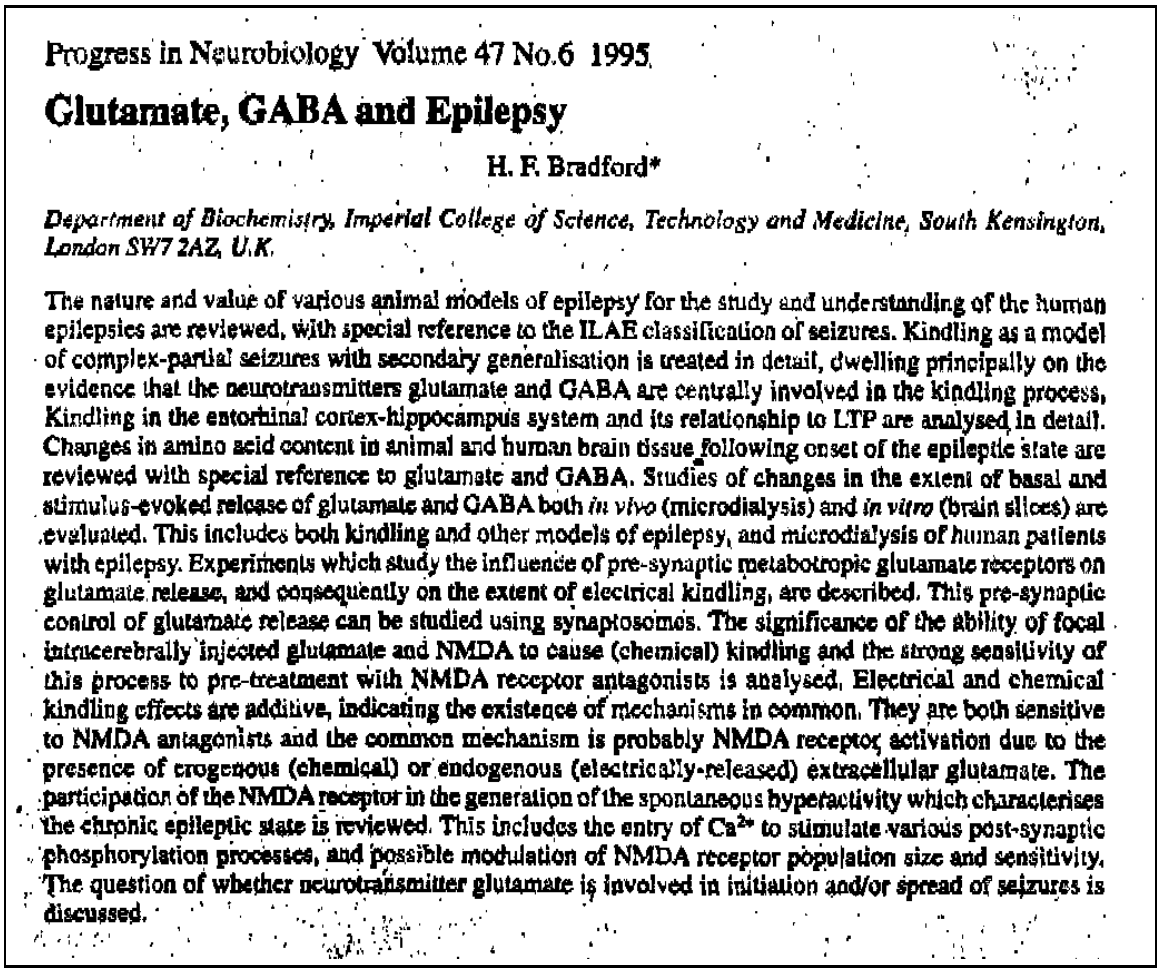


Abbildung D.1: SEDAL-Beispiele: künstlich erzeugter und verschlechterter Textauszug



Progress in Neurobiology Volume 47 No.6 1995

Glutamate, GABA and Epilepsy

H. F. Bradford*

Department of Biochemistry, Imperial College of Science, Technology and Medicine, South Kensington, London SW7 2AZ, U.K.

The nature and value of various animal models of epilepsy for the study and understanding of the human epilepsies are reviewed, with special reference to the ILAE classification of seizures. Kindling as a model of complex-partial seizures with secondary generalisation is treated in detail, dwelling principally on the evidence that the neurotransmitters glutamate and GABA are centrally involved in the kindling process. Kindling in the entorhinal cortex-hippocampus system and its relationship to LTP are analysed in detail. Changes in amino acid content in animal and human brain tissue following onset of the epileptic state are reviewed with special reference to glutamate and GABA. Studies of changes in the extent of basal and stimulus-evoked release of glutamate and GABA both *in vivo* (microdialysis) and *in vitro* (brain slices) are evaluated. This includes both kindling and other models of epilepsy, and microdialysis of human patients with epilepsy. Experiments which study the influence of pre-synaptic metabotropic glutamate receptors on glutamate release, and consequently on the extent of electrical kindling, are described. This pre-synaptic control of glutamate release can be studied using synaptosomes. The significance of the ability of focal intracerebrally injected glutamate and NMDA to cause (chemical) kindling and the strong sensitivity of this process to pre-treatment with NMDA receptor antagonists is analysed. Electrical and chemical kindling effects are additive, indicating the existence of mechanisms in common. They are both sensitive to NMDA antagonists and the common mechanism is probably NMDA receptor activation due to the presence of exogenous (chemical) or endogenous (electrically-released) extracellular glutamate. The participation of the NMDA receptor in the generation of the spontaneous hyperactivity which characterises the chronic epileptic state is reviewed. This includes the entry of Ca^{2+} to stimulate various post-synaptic phosphorylation processes, and possible modulation of NMDA receptor population size and sensitivity. The question of whether neurotransmitter glutamate is involved in initiation and/or spread of seizures is discussed.

Abbildung D.2: SEDAL-Beispiel: echtes Dokument ‘neurofax’

Anhang E

Text-Corpus

Zur besseren Veranschaulichung sind hier Textausschnitte abgedruckt, mit denen bestimmte Zeichen N-Gramme trainiert wurden.

Deutscher ASCII-Text

Beispiele der deutschen allgemeinen Textdatenbasis zur Bestimmung von Sprachmodellen:

- Herzlich willkommen allerseits! Verpaßt im Radio - nachlesen im Internet. Radiofeature Total Global. Klicken Sie in das Herz der Lindenstrasse. Wir suchen interessante Öko-Ideen! Täglich von 16.05 Uhr bis 17.00 Uhr: Der WDR Verkehrsservice im Internet - damit Sie besser vorwärts kommen. Mit der aktuellen NRW Verkehrslage, Autobahnbaustellen und Flughafeninfos. Die Job und Lehrstellenbörse beim WDR: Die Rheinische Post schreibt am Montag nach der Sendung: Redakteur Hans-Georg Kellner: Leitungen waren vor allem in der Zeit von 0: Die WDR online Redaktion liefert Stoff für die Statistik. Nur jeder 36te konnte eine Zugriffschance nutzen. Wieviel Internet-Teilnehmer haben auf die Ecape-Seiten zugegriffen? Neben der Escape-Homepage hatte die Redaktion über 30 weitere Unterseiten ins Netz gestellt. Spezialisten in der call-in Ecke weitergeleitet. Ja, und da war doch noch der Chat: ...

Englischer ASCII-Text

Beispiele der englischen allgemeinen Textdatenbasis zur Bestimmung von Sprachmodellen:

- Search the Carnegie Mellon Web: Founded by industrialist Andrew Carnegie as the Carnegie Technical Schools in 1900, Carnegie Mellon University has emerged as one of the nation’s top private research institutions. Today, its internationally recognized programs encompass the areas of engineering, computer science, technology, science, liberal arts, fine arts and public and private management. ...

- Watch Pam Surano, Len Rome, Meteorologist Stan Boney’s First Weather at 6: Join Len Rome and Pam Surano for a wrap-up of the day’s events. Meteorologist Stan Boney has tomorrow’s forecast at 11: Saturdays and Sundays at 6 p.m.and 11 p.m., keep up with what is going on. Watch Steve Chenevey, Heather Weber with the weather, and Bill Castrovince’s Big Board Sports. Just click to go! At News Channel 33, we need your help. We want to do the best job we can for you, so we need to understand your concerns about our community. This will help us produce better newscasts and help us stay in better touch with you. ...

Literaturverzeichnis

[Bah80] L. Bahl, R. Bakis, P. Cohen, A. Cole, F. Jelinek, B. Lewis und R. Mercer. "Further results on the recognition of a continuously read natural corpus." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 872–875. Denver, 1980.

[Bau68] L. Baum und G. Sell. "Growth transformations for functions on manifolds." *Pac. J. of Math.*, 27(2), Seiten 211–227, 1968.

[Baz99] I. Bazzi, R. Schwartz und J. Makhoul. "An Omnifont Open-Vocabulary OCR System for English and Arabic." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, Nr. 6, Seiten 495–504, Juni 1999.

[Bel98] J. Bellegarda. "Exploiting both local and global constraints for multi-span statistical language modeling." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 677–680. Seattle, Mai 1998.

[Ben95] Y. Bengio, Y. LeCun, C. Nohl und C. Burges. "A NN/HMM Hybrid for On-line Handwriting Recognition." *Neural Computation*, 7, Seiten 1289–1303, 1995.

[Bip99] R.-D. Bippus und V. Maergner. "Cursive Script Recognition Using Inhomogeneous P2DHMM and Hierarchical Search Space Reduction." In *5th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 773–776. Bangalore, India, September 1999.

[Bou94] H. Bourlard und N. Morgan. "Connectionist Speech Recognition - A Hybrid Approach." *Kluwer Academic Press*, 1994.

[Bra96] A. Brakensiek. *Ziffernerkennung auf der Basis komplexer Konturrepräsentationen im Vergleich zu klassischen Erkennungsmethoden*. Diplomarbeit, Fachgebiet Grundlagen der Elektrotechnik, Universität-GH Paderborn, Januar 1996.

[Bra97] A. Brakensiek, U.-D. Braumann, H.-J. Böhme, C. Rieck und H.-M. Groß. "Farb- und strukturbasierte neuronale Verfahren zur Lokalisierung von Gesichtern in Real-World-Szenen." In *19. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 113–120. Braunschweig, Germany, September 1997.

[Bra98] A. Brakensiek, U.-D. Braumann, A. Corradini, H.-J. Böhme und H.-M. Groß. "Neuronale Verfahren zur Lokalisation und Bewertung von Handgesten in Real-World-Szenen." In *Proc. Neuronale Netze in der Anwendung (NN98)*, Seiten 211–218. Magdeburg, Germany, Februar 1998.

[Bra99a] A. Brakensiek, A. Kosmala, D. Willett und G. Rigoll. "Vergleich verschiedener statistischer Modellierungsverfahren für die On- und Off-Line Handschrifterkennung." In *21. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 70–77. Bonn, Germany, September 1999.

[Bra99b] A. Brakensiek, A. Kosmala, D. Willett, W. Wang und G. Rigoll. "Performance Evaluation of a New Hybrid Modeling Technique for Handwriting Recognition Using Identical On-Line and Off-Line Data." In *5th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 446–449. Bangalore, India, September 1999.

[Bra00a] A. Brakensiek, J. Rottland, A. Kosmala und G. Rigoll. "Off-Line Handwriting Recognition Using Various Hybrid Modeling Techniques and Character N-Grams." In *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 343–352. Amsterdam, Netherlands, September 2000.

[Bra00b] A. Brakensiek, D. Willett und G. Rigoll. "Improved Degraded Document Recognition with Hybrid Modeling Techniques and Character N-Grams." In *15th Int. Conference on Pattern Recognition (ICPR)*, Band 4, Seiten 438–441. Barcelona, Spain, September 2000.

[Bra00c] A. Brakensiek, D. Willett und G. Rigoll. "Unlimited Vocabulary Script Recognition Using Character N-Grams." In *22. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 436–443. Kiel, Germany, September 2000.

[Bra01a] A. Brakensiek, A. Kosmala und G. Rigoll. "Comparing Adaptation Techniques for On-Line Handwriting Recognition." In *6th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 486–490. Seattle, USA, September 2001.

[Bra01b] A. Brakensiek, A. Kosmala und G. Rigoll. "Writer Adaptation for On-Line Handwriting Recognition." In *23. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 32–37. Munich, Germany, September 2001.

[Bra01c] A. Brakensiek und G. Rigoll. "A Comparison of Character N-Grams and Dictionaries Used for Script Recognition." In *6th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 241–245. Seattle, USA, September 2001.

[Bra01d] A. Brakensiek, J. Rottland, F. Wallhoff und G. Rigoll. "Adaptation of an Address Reading System to Local Mail Streams." In *6th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 872–876. Seattle, USA, September 2001.

[Bra02a] A. Brakensiek, A. Kosmala und G. Rigoll. "Comparing Normalization and Adaptation Techniques for On-Line Handwriting Recognition." In *16th Int. Conference on Pattern Recognition (ICPR)*. Quebec, Canada, August 2002.

[Bra02b] A. Brakensiek, A. Kosmala und G. Rigoll. "Evaluation of Confidence Measures for On-Line Handwriting Recognition." In *Pattern Recognition, 24. DAGM-Symposium, Springer-Verlag*, Seiten 507–514. Zurich, Switzerland, September 2002.

[Bra02c] A. Brakensiek, J. Rottland und G. Rigoll. "Handwritten Address Recognition with Open Vocabulary Using Charcter N-Grams." In *8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 357–362. Niagara-on-the-Lake, Canada, August 2002.

[Bre01] H. Breit und G. Rigoll. "Improved Person Tracking Using a Combined Pseudo-2D-HMM and Kalman Filter Approach with Automatic Background State Adaptation." In *IEEE Int. Conference on Image Processing (ICIP)*. Thessaloniki, Greece, Oktober 2001.

[Bun95] H. Bunke, M. Roth und E. Schukat-Talamazzini. "Off-Line Cursive Handwriting Recognition Using Hidden Markov Models." *Proc. Pattern Recognition*, 28, Seiten 1399–1413, 1995.

[Cae93] T. Caesar, J. Gloger und E. Mandler. "Preprocessing and Feature Extraction for a Handwriting Recognition System." In *Proc. Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 408–411. Tsukuba, Japan, Oktober 1993.

[Che99] C. Chesta, O. Siohan und C.-H. Lee. "Maximum A Posteriori Linear Regression for Hidden Markov Model Adaptation." In *6th European Conference on Speech Communication and Technology (Eurospeech)*, Seiten 211–214. Budapest, Hungary, September 1999.

[Cla97] P. Clarkson und R. Rosenfeld. "Statistical Language Modeling Using the CMU-Cambridge Toolkit." In *5th European Conference on Speech Communication and Technology (Eurospeech)*, Seiten 2707–2710. Rhodes, Greece, September 1997.

[Coc98] N. Coccaro und D. Jurafsky. "Towards better integration of semantic predictors in statistical language modeling." In *5th Int. Conference on Spoken Language Processsing (ICSLP)*, Seiten 2403–2406. Sydney, Australia, Dezember 1998.

[Con99] S. Connell und A. Jain. "Writer Adaptation of Online Handwriting Models." In *5th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 434–437. Bangalore, India, September 1999.

[Côt98] M. Côté, E. Lecolinet, M. Cheriet und C. Suen. "Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts: the PERCEPTO system." *Int. Journal on Document Analysis and Recognition (IJDAR)*, 1, 1998.

[Doe92] D. Doermann und A. Rosenfeld. "Recovery of temporal information from static images of handwriting." In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Seiten 162–168. Champaign, IL, Juni 1992.

[Dol98] J. Dolfing und A. Wendemuth. “Combination of Confidence Measures in Isolated Word Recognition.” In *5th Int. Conference on Spoken Language Processsing (ICSLP)*, Seiten 3237–3240. Sydney, Australia, Dezember 1998.

[Dud73] R. Duda und P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[Eic98] S. Eickeler, A. Kosmala und G. Rigoll. “Hidden Markov Model Based Continuous Online Gesture Recognition.” In *Int. Conference on Pattern Recognition (ICPR)*, Seiten 1206–1208. Brisbane, Australia, August 1998.

[Eic00] S. Eickeler, M. Jabs und G. Rigoll. “Comparison of Confidence Measures for Face Recognition.” In *IEEE Int. Conference on Automatic Face and Gesture Recognition*, Seiten 257–262. Grenoble, France, März 2000.

[Eis93] G. Eisenack. “Erkennen grafischer mathematischer Ausdrücke.” *ct-Magazin für Computertechnik*, 9, Seiten 188–191, 1993.

[Elm98] A. Elms, S. Procter und J. Illingworth. “The advantage of using an HMM-based approach for faxed word recognition.” *Int. Journal on Document Analysis and Recognition (IJDAR)*, 1, Seiten 18–38, 1998.

[FG01] Fraunhofer-Gesellschaft. *Elektronischer Schreibstift*. http://www.fraunhofer.de (TEG), 2001.

[Fra97] J. Franke, J. Gloger, A. Kaltenmeier und E. Mandler. “A Comparison of Gaussian Distribution and Polynomial Classifiers in a Hidden Markov Model Based System for the Recognition of Cursive Script.” In *Proc. Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 515–518. Ulm, Germany, August 1997.

[Fri97] J. Fritsch, M. Finke und A. Waibel. “Context-Dependent Hybrid HME/HMM Speech Recognition using Polyphone Clustering Decision Trees.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 1759–1762. Munich, Germany, April 1997.

[Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[Gal96] M. Gales und P. Woodland. “Variance Compensation within the MLLR Framework for Robust Speech Recognition and Speaker Adaptation.” In *Int. Conference on Spoken Language Processing (ICSLP)*, Seiten 1832–1835. Philadelphia, PA, September 1996.

[Gau91] J.-L. Gauvain und C.-H. Lee. “Bayesian Learning of Gaussian Mixture Densities for Hidden Markov Models.” In *Proc. of the DARPA Speech and Natural Language Workshop*, Seiten 272–277. Pacific Grove, CA, Februar 1991.

[Gau94] J.-L. Gauvain und C.-H. Lee. “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observation of Markov Chains.” *IEEE Transactions on Speech and Audio Processing*, 2, Nr. 2, Seiten 291–298, April 1994.

[Glo97] J. Gloger, A. Kaltenmaier, E. Mandler und L. Andrews. “Reject Management in a Handwriting Recognition System.” In *Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 556–559. Ulm, Germany, August 1997.

[Gon93] R. Gonzales und R. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.

[Guy94] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman und S. Janet. “UNIPEN project of on-line data exchange and benchmarks.” In *Int. Conference on Pattern Recognition (ICPR)*, Seiten 29–33. Jerusalem, Israel, 1994.

[Haz01] T. Hazen und I. Bazzi. “A Comparison and Combination of Methods for OOV Word Detection and Word Confidence Scoring.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Salt Lake City, Utah, Mai 2001.

[Hu98] J. Hu, S. Lim und M. Brown. “HMM Based Writer Independent On-line Handwritten Character and Word Recognition.” In *6th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 143–155. Taejon, Korea, 1998.

[Hun00] J. Hunsinger und M. Lang. “A Single-Stage Top-Down Probabilistic Approach towards Understanding Spoken and Handwritten Mathematical Formulas.” In *6th Int. Conference on Spoken Language Processing (ICSLP)*, Seiten 386–389. Beijing, China, Oktober 2000.

[Hut94] H.-P. Hutter und B. Pfister. “Neuartiger Hybrider SKHMM/KNN-Ansatz für die Spracherkennung.” In *Elektronische Sprachsignalverarbeitung, Tagungsband*, Seiten 90–97. Berlin, Germany, Oktober 1994.

[IBM01] IBM. *Thinkpad TransNote*. http://www.ibm.com/pc/de, 2001.

[Iur01] U. Iurgel, R. Meermeier, S. Eickeler und G. Rigoll. “New Approaches to Audio-Visual Segmentation of TV News for Automatic Topic Retrieval.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten ?–? Salt Lake City, Utah, Mai 2001.

[Jae01] S. Jaeger, S. Manke, J. Reichert und A. Waibel. “On-Line Handwriting Recognition: The NPen++ Recognizer.” *Int. Journal on Document Analysis and Recognition (IJDAR)*, 3, Seiten 169–180, 2001.

[Jel98] F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts, 1998.

[Jun98] M. Junker und R. Hoch. “An experimental evaluation of OCR text representations for learning document classifiers.” *Int. Journal on Document Analysis and Recognition (IJDAR)*, 1, Seiten 116–122, 1998.

[Kan00] P. Kantor und E. Voorhees. “The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text.” *Information Retrieval*, 2, Seiten 165–176, 2000.

[Kat87] S. Katz. “Estimation of probabilities from sparse data for the language model component of a speech recognizer.” *IEEE Transactions on Acoustic, Speech and Signal Processing*, 35(3), Seiten 400–401, 1987.

[Kem97] T. Kemp und T. Schaaf. “Estimating Confidence using Word lattices.” In *5th European Conference on Speech Communication and Technology (Eurospeech)*, Seiten 827–830. Rhodes, Greece, September 1997.

[Kos97a] A. Kosmala, J. Rottland und G. Rigoll. “Improved On-Line Handwriting Recognition Using Context Dependent Hidden Markov Models.” In *Int. Conference on Document Analysis and Recognition (ICDAR)*, Band 2, Seiten 641–644. Ulm, Germany, August 1997.

[Kos97b] A. Kosmala, J. Rottland und G. Rigoll. “Large Vocabulary On-Line Handwriting Recognition with Context Dependent Hidden Markov Models.” In *19. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 254–261. Braunschweig, Germany, September 1997.

[Kos00a] A. Kosmala. *HMM-basierte Online Handschrifterkennung - ein integrierter Ansatz zur Text- und Formelerkennung*. Dissertation, Fachbereich Elektrotechnik, Gerhard-Mercator-Universität Duisburg, Dezember 2000.

[Kos00b] A. Kosmala und G. Rigoll. “On-Line Handwritten Formula Recognition with Integrated Correction Recognition and Execution.” In *15th Int. Conference on Pattern Recognition (ICPR)*, Seiten 590–593. Barcelona, Spain, September 2000.

[Lal00] P. Lallican, C. Viard-Gaudin und S. Knerr. “From Off-Line to On-Line Handwriting Recognition.” In *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 303–312. Amsterdam, Netherlands, September 2000.

[Lef01] F. Lefevre, J.-L. Gauvain und L. Lamel. “Towards Task-Independent Speech Recognition.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Salt Lake City, Utah, Mai 2001.

[Leg94] C. Leggetter und P. Woodland. “Speaker Adaptation of Continuous Density HMMs using Multivariate Linear Regression.” In *Int. Conference on Spoken Language Processing (ICSLP)*, Seiten 451–454. Yokohama, Japan, September 1994.

[Lin80] Y. Linde, A. Buzo und R. Gray. “An algorithm for vector quantizer design.” *IEEE Transactions on Communication*, COM-28, 1, Seiten 84–95, 1980.

[Lu99] Z. Lu, I. Bazzi, A. Kornai, J. Makhoul, P. Natarajan und R. Schwartz. “A Robust, Language-Independent OCR System.” In *Proc. 27th AIPR Workshop: Advances in Computer-Assisted Recognition (SPIE)*, Seiten 96–105, 1999.

[Man90] E. Mandler und M. Oberländer. “A single pass algorithm for fast contour coding of binary images.” In *12. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 248–255. Oberkochen-Aalen, Germany, September 1990.

[Man98] S. Manke. *On-line Erkennung kursiver Handschrift bei großen Vokabularen*. Dissertation, Fakultät für Informatik, Universität Karlsruhe, Februar 1998.

[Man99] C. Manning und H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, Mai 1999.

[Mar99] U.-V. Marti und H. Bunke. "A full English sentence database for off-line handwriting recognition." In *5th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 705–708. Bangalore, India, September 1999.

[Mar00] U.-V. Marti und H. Bunke. "Unconstrained Handwriting Recognition: Language Models, Perplexity, and System Performance." In *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 463–468. Amsterdam, Netherlands, September 2000.

[Mat93] N. Matic, I. Guyon, J. Denker, und V. Vapnik. "Writer adaptation for on-line handwritten character recognition." In *Int. Conference on Pattern Recognition and Document Analysis*, Seiten 187–191. Tsukuba, Japan, 1993.

[McQ67] J. McQueen. "Some Methods for Classification and Analysis of Multivariate Observations." In *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Seiten 281–297. California, Berkeley, 1967.

[Mül00] S. Müller, S. Eickeler und G. Rigoll. "Crane Gesture Recognition using Pseudo 3-D Hidden Markov Models." In *IEEE Int. Conference on Automatic Face and Gesture Recognition*, Seiten 398–402. Grenoble, France, März 2000.

[Mül01] S. Müller, S. Eickeler und G. Rigoll. "An Integrated Approach to Shape and Color-Based Image Retrieval of Rotated Objects Using Hidden Markov Models." *Int. Journal on Pattern Recognition and Artificial Intelligence*, 15, Nr. 1, Februar 2001.

[Nat99] P. Natarajan, I. Bazzi, Z. Lu, J. Makhoul und R. Schwartz. "Robust OCR of Degraded Documents." In *5th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 357–361. Bangalore, India, September 1999.

[Neu97a] C. Neukirchen und G. Rigoll. "Advanced Training Methods and New Network Topologies for Hybrid MMI-Connectionist/HMM Speech Recognition Systems." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 3257–3260. Munich, Germany, April 1997.

[Neu97b] C. Neukirchen und G. Rigoll. "Time Series Classification using Hidden Markov Models and Neural Networks." In *Proc. of the IAR Annual Meeting*. Duisburg, Germany, November 1997.

[Neu99] C. Neukirchen. *Integration neuronaler Vektorquantisierer in ein Hidden-Markov-Modell-basiertes System zur automatischen Spracherkennung*. Dissertation, Fachbereich Elektrotechnik, Gerhard-Mercator-Universität Duisburg, Juni 1999.

[Neu01] C. Neukirchen, J. Rottland, D. Willett und G. Rigoll. "A Continuous Density Interpretation of Discrete HMM Systems and MMI-Neural Networks." In *IEEE Transactions on Speech and Audio Processing*, Mai 2001.

[Nie97] T. Niesler. *Category-based statistical language models*. Dissertation, Dept. of Engineering, University of Cambridge, U.K., Juni 1997.

[Par02] Paragraph. *CalliGrapher*. http://www.paragraph.com/calligrapher.html, 2002.

[Pau92] D. Paul. "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 25–28. San Francisco, CA, März 1992.

[Pit00] J. Pitrelli und E. Ratzlaff. "Quantifying the Contribution of Language Modeling to Writer-Independent On-Line Handwriting Recognition." In *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 383–392. Amsterdam, Netherlands, September 2000.

[Pla00] R. Plamondon und S. Srihari. "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22, Nr. 1, Seiten 63–84, Januar 2000.

[Rab86] L. Rabiner und B. Juang. "An Introduction to Hidden Markov Models." *IEEE ASSP Magazine*, Seiten 4–16, 1986.

[Rig96] G. Rigoll, A. Kosmala, J. Rottland und C. Neukirchen. "A Comparison Between Continuous and Discrete Density Hidden Markov Models for Cursive Handwriting Recognition." In *Int. Conference on Pattern Recognition (ICPR)*, Band 2, Seiten 205–209. Vienna, Austria, August 1996.

[Rig98a] G. Rigoll und A. Kosmala. "A Systematic Comparison Between On-Line and Off-Line Methods for Signature Verification with Hidden Markov Models." In *Int. Conference on Pattern Recognition (ICPR)*, Seiten 1755–1757. Brisbane, Australia, August 1998.

[Rig98b] G. Rigoll, A. Kosmala und D. Willett. "A New Hybrid Approach to Large Vocabulary Cursive Handwriting Recognition." In *Int. Conference on Pattern Recognition (ICPR)*, Seiten 1512–1514. Brisbane, Australia, August 1998.

[Rig98c] G. Rigoll, A. Kosmala und D. Willett. "An Investigation of Context-Dependent and Hybrid Modeling Techniques for Very Large Vocabulary On-Line Cursive Handwriting Recognition." In *6th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*. Taejon, Korea, August 1998.

[Ros90] R. Rose und D. Paul. "A Hidden Markov Model based Keyword Recognition System." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 129–132. Albuquerque, New Mexico, 1990.

[Ros00] R. Rosenfeld. “Two decades of statistical language modeling: Where do we go from here?” *Proceedings of the IEEE*, 88(8), 2000.

[Rot99] J. Rottland, C. Neukirchen, D. Willett und G. Rigoll. “Speaker Adaptation Using Regularization and Network Adaptation for Hybrid MMI-NN/HMM Speech Recognition.” In *6th European Conference on Speech Communication and Technology (Eurospeech)*, Seiten 219–222. Budapest, Hungary, September 1999.

[Rot00a] J. Rottland. *Ein hybrider Ansatz zur automatischen Spracherkennung und Sprecheradaption für große Wortschätze*. Dissertation, Fachbereich Elektrotechnik, Gerhard-Mercator-Universität Duisburg, Februar 2000.

[Rot00b] J. Rottland und G. Rigoll. “Tied Posteriors: An Approach for Effective Introduction of Context Dependency in Hybrid NN/HMM LVCSR.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Istanbul, Turkey, Juni 2000.

[Sao97] G. Saon und A. Belaid. “Off-line Handwritten Word Recognition Using a Mixed HMM-MRF Approach.” In *Proc. Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 118–122. Ulm, Germany, August 1997.

[Sch94] M. Schenkel, I. Guyon und D. Henderson. “On-Line Cursive Script Recognition Using Time Delay Neural Networks And Hidden Markov Models.” In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 637–640. Adelaide, Australia, April 1994.

[Sch97a] M. Schüßler und H. Niemann. “Die Verwendung von Kontextmodellen bei der Verwendung handgeschriebener Wörter.” In *19. DAGM-Symposium, Tagungsband Springer-Verlag*, Seiten 262–269. Braunschweig, Germany, September 1997.

[Sch97b] M. Schuster. “Incorporation of HMM Output Constraints in Hybrid NN/HMM Systems During Training.” In *5th European Conference on Speech Communication and Technology (Eurospeech)*, Seiten 2843–2846. Rhodes, Greece, 1997.

[Sch98] M. Schenkel und M. Jabri. “Low resolution, degraded document recognition using neural networks and hidden Markov models.” *Pattern Recognition Letters*, 19, Seiten 365–371, 1998.

[Sch99] L. Schomaker und E. Segers. “Finding features used in the human reading of cursive handwriting.” *Int. Journal on Document Analysis and Recognition (IJDAR)*, 2, Seiten 13–18, 1999.

[Sei96] R. Seiler, M. Schenkel und F. Eggiman. “Off-Line Cursive Handwriting Recognition Compared with On-Line Recognition.” In *Proc. Int. Conference on Pattern Recognition (ICPR)*, Seiten 505–509. Vienna, Austria, August 1996.

[Sen97] A. Senior und K. Nathan. “Writer adaptation of a HMM handwriting recognition system.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 1447–1450. Munich, Germany, April 1997.

[Sri93] R. Srihari, C. Ng, C. Baltus und J. Kud. "Use of Language models in On-line Recognition of Handwritten Sentences." In *3rd Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 284–294. Buffalo, NY, 1993.

[Sri01] S. Srihari, S.-H. Cha und S. Lee. "Establishing Handwriting Individuality Using Pattern Recognition Techniques." In *6th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 1195–1204. Seattle, USA, September 2001.

[ST95] E. Schukat-Talamazzini. *Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Braunschweig, 1995.

[Sta00] J. Stadermann, J. Rottland und G. Rigoll. "Tied-Posteriors: A New Hybrid Speech Recognition Technology with Generic Capabilities and High Portability." In *ISCA Tutorial and Research Workshop ASR2000*. Paris, September 2000.

[Ste99] T. Steinherz, E. Rivlin und N. Intrator. "Offline cursive sript word recognition - a survey." *Int. Journal on Document Analysis and Recognition (IJDAR)*, 2, Seiten 90–110, 1999.

[Tri96] O. Trier, A. Jain und T. Taxt. "Feature extraction methods for character recognition - A survey." *Proc. Pattern Recognition*, 29, Seiten 641–662, 1996.

[Val96] V. Valtchev, J. Odell, P. Woodland und S. Young. "Lattice-Based Discriminative Training for Large Vocabulary Speech Recognition Systems." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 605–608. Atlanta, GA, Mai 1996.

[Van02] K. VanHorn. "Commentary on Continuous Density Interpretation of Discrete HMMs." *IEEE Transactions on Speech and Audio Processing, submitted*, 2002.

[Vin02] A. Vinciarelli und S. Bengio. "Writer adaptation techniques in HMM based Off-Line Cursive Script Recognition." *Pattern Recognition Letters*, 23, Nr. 8, Seiten 905–916, Juni 2002.

[Vuo01] V. Vuori, J. Laaksonen, E. Oja und J. Kangas. "Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters." *Int. Journal on Document Analysis and Recognition (IJDAR)*, 3, Seiten 150–159, 2001.

[WAC] WACOM. *Elektronische Grafiktabletts*. http://www.wacom.de.

[Wal99] F. Wallhoff. *Überwachte und unüberwachte Sprecheradaption mit verschiedenen Trainingskriterien für die automatische Spracherkennung*. Diplomarbeit, Faculty of Electrical Engineering - Computer Science, Gerhard-Mercator-University Duisburg, Dezember 1999. In German.

[Wal00] F. Wallhoff, D. Willett und G. Rigoll. "Frame Discriminative and Confidence-Driven Adaptation for LVCSR." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 1835–1838. Istanbul, Turkey, Juni 2000.

[Wal01a] F. Wallhoff, S. Eickeler und G. Rigoll. "A Comparison of Discrete and Continuous Output Modeling Techniques for a Pseudo-2D Hidden Markov Model Face Recognition System." In *IEEE Int. Conference on Image Processing (ICIP)*. Thessaloniki, Greece, Oktober 2001.

[Wal01b] F. Wallhoff und G. Rigoll. "A Novel Hybrid Face Profile Recognition System Using The FERET And MUGSHOT Databases." In *IEEE Int. Conference on Image Processing (ICIP)*. Thessaloniki, Greece, Oktober 2001.

[Wan00] W. Wang, A. Brakensiek, A. Kosmala und G. Rigoll. "HMM based High Accuracy Off-line Cursive Handwriting Recognition by a Baseline Detection Error Tolerant Feature Extraction Approach." In *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Seiten 209–218. Amsterdam, Netherlands, September 2000.

[Wan01] W. Wang, A. Brakensiek, A. Kosmala und G. Rigoll. "Multi-Branch and Two-Pass HMM Modeling Approaches for Off-Line Cursive Handwriting Recognition." In *6th Int. Conference on Document Analysis and Recognition (ICDAR)*, Seiten 231–235. Seattle, USA, September 2001.

[Wan02a] W. Wang, A. Brakensiek und G. Rigoll. "Combination of Multiple Classifiers for Handwritten Word Recognition." In *8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*. Niagara-on-the-Lake, Canada, August 2002.

[Wan02b] W. Wang, A. Brakensiek und G. Rigoll. "Combining HMM-Based Two-Pass Classifiers for Off-Line Word Recognition." In *16th Int. Conference on Pattern Recognition (ICPR)*. Quebec, Canada, August 2002.

[Wil98] D. Willett, A. Worm, C. Neukirchen und G. Rigoll. "Confidence Measures for HMM-based Speech Recognition." In *5th Int. Conference on Spoken Language Processsing (ICSLP)*, Seiten 3241–3244. Sydney, Australia, Dezember 1998.

[Wil99] D. Willett, C. Neukirchen, J. Rottland und G. Rigoll. "Refining Tree-Based Clustering by Means of Formal Concept Analysis, Balanced Decision Trees and Automatically Generated Model-Sets." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 565–568. Phoenix, AZ, März 1999.

[Wil00a] D. Willett. *Beitraege zur statistischen Modellierung und effizienten Dekodierung in der automatischen Spracherkennung*. Dissertation, Fachbereich Elektrotechnik, Gerhard-Mercator-Universität Duisburg, November 2000.

[Wil00b] D. Willett, C. Neukirchen und G. Rigoll. "DUCODER - The Duisburg University LVCSR Stackdecoder." In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 1555–1558. Istanbul, Turkey, Juni 2000.

[Yan95] L. Yang, B. Widjaja und R. Prasad. "Application of Hidden Markov Models for signature verification." *Proc. Pattern Recognition*, 28, 1995.

[You94] S. Young. “Detecting Misrecognitions and Out-Of Vocabulary Words.” In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 21–24. Adelaide, Australia, April 1994.

[You00] S. Young, J. Jansen, J. Odell, D. Ollason und P. Woodland. *The HTK Book*. University of Cambridge, 2000.

[Zel97] A. Zell. *Simulation neuronaler Netze*. Oldenbourg Verlag, München, 1997.