Diplomarbeit

Thema:

Spracherkennung von gesprochenen und buchstabierten Eigennamen

von

Michael Meyer

Bearbeitungszeitraum:

1. Oktober 1996 – 31. März 1997

Institut für Logik, Komplexität und Deduktionssysteme
der Universität Karlsruhe

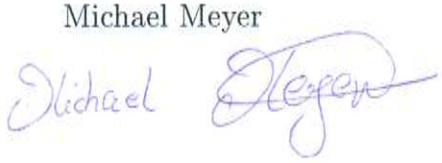
Betreuer:

Prof. Dr. Alexander Waibel
Dipl.-Inform. Hermann Hild

Diese Arbeit wurde von mir selbständig angefertigt. Alle verwendeten Literaturstellen sind im Literaturverzeichnis angeführt; eine Verwendung anderer Hilfsmittel erfolgte nicht. Ich versichere dies mit der nachstehenden Unterschrift.

Karlsruhe, 31. März 1997

Michael Meyer



Inhaltsverzeichnis

1 Einleitung

Aufgabe und Ziel dieser Arbeit ist die maschinelle Erkennung von kontinuierlich gesprochenen[1] und buchstabierten Eigennamen.

Die Erkennung von Eigennamen ist ein wichtiger Bestandteil für eine Vielzahl von Anwendungen, z.B. Telefonauskunftssysteme, oder Fahrplananfragen [ST95]. Besonders schwierig ist die Erkennung von großen Namenslisten. Neben den technischen Problemen der Haltung und Verarbeitung vieler Daten beinhalten große Vokabulare sehr ähnlich klingende Einträge, die leicht verwechselt werden können. Zur Reduktion von Verwechslungen werden Eigennamen oft buchstabiert oder gesprochen und buchstabiert. Der buchstabierte Name erweist sich, insbesondere bei Datenbankanfragen, bei denen es auf die Schriftform ankommt, als wichtiger Informationsträger.

In dieser Diplomarbeit soll untersucht werden, wie gut nur gesprochene, nur buchstabierte und gesprochene und buchstabierte Eigennamen erkannt werden können. Im letzteren Fall existieren zwei unterschiedliche akustische Repräsentationen für denselben Namen. Zur Ausnutzung dieser Redundanz werden verschiedene Verfahren vorgestellt, die auch bei großen Namenslisten noch eingesetzt werden können.

Im Anschluß an diese Einleitung wird ein kurzer Überblick über die maschinelle Spracherkennung gegeben. Dabei werden besonders die für diese Arbeit relevanten Verfahren und Algorithmen beschrieben.

Dann wird die Aufgabenstellung näher betrachtet und Basisexperimente durchgeführt, anhand derer sich die weiteren Ansätze messen lassen.

In den drei anschließenden Kapitel werden Verfahren zum Erkennen von Eigennamen vorgestellt. Von Kapitel zu Kapitel wächst dabei die Menge der Namen, die erkannt werden können. In den jeweiligen Unterkapiteln wird auf die unterschiedlichen Möglichkeiten der Spracheingabe eingegangen. Die beschriebenen Verfahren wurden implementiert und es wurden Experimente auf Sprachdaten durchgeführt.

In Kapitel 7 wird ein flexibler Ansatz vorgestellt. Der Anwender kann einen Namen buchstabieren, oder sprechen und buchstabieren. Es wird automatisch erkannt, um welche Variante es sich handelt, damit sie dann entsprechend weiter verarbeitet werden kann.

Die Ergebnisse werden in Kapitel 8 zusammengefaßt und verglichen. Da-

[1] Im folgenden Text wird von **gesprochenen** Namen die Rede sein, wenn der kontinuierlich gesprochene Name gemeint ist.

rauf aufbauend und aus den Erfahrungen bei der Durchführung der Arbeit, werden Vorschläge zur Verbesserung und Weiterführung im Kapitel "Ausblick" aufgezählt.

Im Anhang befinden sich die Beschreibungen der verwendeten Datensätze.

Danksagung:
Ich möchte mich bei den Mitarbeitern des Instituts, besonders bei Dipl.-Inform. Hermann Hild, für die geduldige Beantwortung meiner unzähligen Fragen und die konstruktiven Diskussionen bedanken, sowie für die Zurverfügungstellung enormer Rechen- und Speicherkapazitäten, ohne die diese Diplomarbeit nicht hätte durchgeführt werden können.

2 Grundlagen der maschinellen Spracherkennung

In diesem einführenden Kapitel wird die Problematik der automatisierten, maschinellen Sprachverarbeitung erläutert. Es werden einige Begriffe und Verfahren erklärt, die für das weitere Verständnis wichtig sind.

2.1 Spracherkennung allgemein

Die maschinelle Spracherkennung analysiert digitale Repräsentationen gesprochener Sprache. Gegenstand der Analyse kann der Sprecher, die Sprache oder der geäußerte Text sein. In dieser Diplomarbeit ist letzteres von Interesse. Es ist also die lautsprachliche Information einer gesprochenen Äußerung auszuwerten und das Ergebnis als Wort oder Folge von Wörtern auszugeben. Diese komplexe Aufgabe läßt sich in Teilschritte aufspalten. Die Gesamtaufgabe wird dadurch modularisiert und die einzelnen Komponenten werden austauschbar. In Abbildung 1 wird exemplarisch eine mögliche Struktur vorgestellt.

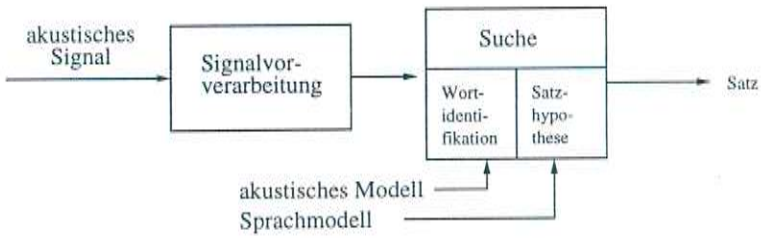


Abbildung 1: Grundstruktur eines Spracherkenners.

Im ersten Teilschritt wird die akustische Information in eine für die weitere Verarbeitung geeignete parametrische Darstellung transformiert. Diese nennen wir im weiteren *Merkmalsvektoren*. Die Transformation soll folgende Eigenschaften erfüllen:

- Datenreduktion ohne Verlust "wichtiger" Informationen, d.h. Abschwächen bzw. Unterdrücken irrelevanter Informationen oder Störgeräusche.

- Verstärkung der Merkmale, die zur Unterscheidung der gesprochenen Laute oder Wörter wichtig sind.

Eine ausführliche Beschreibung der Signalvorverarbeitung findet sich in [SR75].

Früher wurde als Zwischenschritt zwischen der Signalvorverarbeitung und der Suche eine Vektorquantisierung durchgeführt, um von einer numerischen zu einer symbolischen Präsentation zu gelangen. Es gibt eine Vielzahl von Klassifikationsverfahren [DH73], die in der angeführten Literatur beschrieben und auf ihre Tauglichkeit bei konkreten Problemstellungen untersucht wurden. Grundsätzlich läßt sich feststellen, daß jede Quantisierung fehlerbehaftet ist und somit Information verloren geht. Aus der symbolischen Präsentation des ursprünglichen Signals soll eine Wort- oder Satzhypothese gefunden werden. Diesen Vorgang nennen wir kurz *Suche* (oder Decoder).

Heute hat sich weitgehend der statistische Spracherkennungsansatz mit Hidden-Markov-Modellen (HMM) durchgesetzt. Im folgenden Unterkapitel werden Hidden Markov Modelle vorgestellt. Die *Suche* nach der besten Worthypothese wird durch die Modellierung durch HMMs zur Suche nach dem Modell, das die aktuellen Merkmalsvektoren am besten beschreibt.

Die Menge aller Wörter, die dem Erkenner zur Verfügung stehen, bezeichnen wir als *Vokabular*. Die Suche wird in unserem Schema über einem akustischen und einem Sprachmodell definiert. Während das akustische Modell für die Klassifikation von Wörtern verantwortlich ist, kann ein Sprachmodell die Häufigkeit und Abhängigkeit der Wörter untereinander berücksichtigen.

Die Verknüpfung dieser beiden Modelle läßt sich durch die *Bayes-Regel*, die oft auch als Fundamentalformel (1) der statistischen Spracherkennung bezeichnet wird, beschreiben.

$P(W|X)$: A posteriori Wahrscheinlichkeit
$P(X|W)$: Klassenbedingte Wahrscheinlichkeit
$P(W)$: A priori Wahrscheinlichkeit der Wortsqequenz W
$p(X)$: Gesamtwahrscheinlichkeit der Eingabe X

$$P(W|X) = \frac{P(X|W) \cdot P(W)}{p(X)} \tag{1}$$

2.2 Einzelworterkennung

2.2.1 Dynamic Time Warping

Für die Erkennung weniger einzelner Wörter kann für jedes Wort im Vokabular ein Repräsentant bestimmt werden. Für sprecherunabhängige Systeme wird ein Repräsentant nicht ausreichend sein. Deshalb können mehrere pro Wort zugelassen werden (z.B.: für Frauen und Männer). Um zu einer sprachlichen Äußerung die beste Hypothese zu ermitteln, wird jeder Repräsentant mit ihr verglichen. Der Abstand zwischen der Eingabe- und Referenzfolge kann als Summe lokaler Distanzen entlang eines geeigneten Zeitverzerrungspfades berechnet werden. Diese Aufgabe läßt sich durch dynamisches Programmieren realisieren. Der effiziente Algorithmus ist unter dem Begriff "dynamic time warping" [VZ70] bekannt.

Um ganze Sätze mit diesem musterbasierten Ansatz erkennen zu können, existieren zwei wesentliche Ansätze: "Two-level DP Matching"[SC90] und "One Stage Dynamic Programming[Ney84]. Das Auffinden der Wortgrenzen ist ein fehlerbehafteter Prozeß. Er muß zusammen mit der Satzhypothesenbildung optimiert werden.

2.2.2 Hidden-Markov-Modelle

Eine andere Methode, die sich in der Spracherkennung durchgesetzt hat, ist die Modellierung der Wörter durch stochastische Modelle. Die Beschreibung durch "hidden markov" Modelle war bereits 1975 von verschiedenen Forschungsgruppen vorgeschlagen und benutzt worden[Bak75, Jel76]. Dadurch war es möglich, den statistischen Zusammenhang zwischen Spracheinheiten und ihren akustischen Gegenstücken in eine geschlossene mathematische Theorie zu vereinigen, die freien Parameter aus vorgelegten Sprachproben zu schätzen und bei Bedarf anzupassen. Dies führte zu Fortschritten bezüglich Sprecheranpassung, Erweiterbarkeit und Erkennung großer Wortschätze.

Die akustische Modellierung des Wortes w_j wird durch den Term $P(x|w_j)$ ausgedrückt. Leider sind die dazugehörigen Verteilungsdichten nicht bekannt, sondern werden durch eine geeignete Funktionsfamilie mit einem zur Musterklasse w_j passenden Parametersatz λ_j angenähert. Dieser Parametersatz stellt somit ein Modell für ein Wort dar. Durch ein vorgeschaltetes statistisches Lernverfahren lassen sich mittels verschiedener Äußerungen eines Wortes die Parameter automatisch schätzen. In der Spracherkennung haben sich die Markov Modelle [LRS83] als Wortmodellierung von zeitlichen

Abfolgen akustischer Äußerungen durchgesetzt.

Definition
Ein HMM wird durch folgende Eigenschaften charakterisiert[Rab89]:

1. **N** sei die Anzahl der Zustände des Modells.

2. **M** sei die Anzahl verschiedener Symbole. Betrachtet man das Markov-Modell als stochastischen Automat, so produziert er die Symbole $O = O_1 O_2 \ldots O_T$.

3. Die **Transitionswahrscheinlichkeit** ist die Wahrscheinlichkeit, aus dem Zustand S_i in den Zustand S_j zu wechseln:

$$\begin{aligned} a_{ij} &= P(q_{t+1} = S_j | q_t = S_i) \qquad \text{mit} \qquad 1 \leq i, j \leq N \\ A &= [a_{ij}]_{N \times N} \end{aligned} \tag{2}$$

   dabei müssen die a_{ij} die *Stochastizitätsbedingungen*

   - $a_{ij} \geq 0$
   - $\sum_j a_{ij} = 1$

   erfüllen. Diese Parametermatrix wird auch als Transitionsmatrix bezeichnet.

4. Die **Emissionswahrscheinlichkeit** ist die Wahrscheinlichkeit, im Zustand S_j das Symbol O_k zu beobachten.

$$\begin{aligned} b_{jk} &= P(O_k | q_t = S_j) \qquad \text{mit} \qquad 1 \leq j \leq N \ , \ 1 \leq k \leq M \\ B &= [b_{jk}]_{NxM} \end{aligned} \tag{3}$$

5. Der **Anfangszustand** wird durch den N-dimensionalen Vektor π beschrieben.

$$\pi_i = P(q_1 = S_i) \qquad \text{mit} \qquad 1 \leq i \leq N \tag{4}$$

Damit ergibt sich als Definition für das Markov-Modell λ das Tripel:

$$\lambda = (\pi, A, B)$$

Dieses HMM läßt sich als Generator von Symbolsequenzen aus dem über V definierten Alphabet nutzen, oder als Modell, um zu analysieren, wie eine Symbolsequenz durch das HMM erzeugt werden könnte.

Damit Markov-Modelle als Modell für einen stochastischen Prozeß benutzt werden können, werden Algorithmen benötigt, die zum einen die Parameter der Modelle optimal einstellen (*Training*) und zum anderen das Modell finden, welches am besten zu einer Eingabesequenz paßt (*Test*). Für den Test existieren zwei Algorithmen:

1. **geg.:** die Symbolsequenz: $O = O_1 O_2 \ldots O_T$ und ein Modell $\lambda = (\pi, A, B)$

   ges.: die klassenbedingte Wahrscheinlichkeit $P(O|\lambda)$

   Lösung: *Forward-Backward-Algorithmus, (Forward-Alg.)*

2. **geg.:** die Symbolsequenz: $O = O_1 O_2 \ldots O_T$ und ein Modell $\lambda = (\pi, A, B)$

   ges.: die "beste" zur Beobachtungssequenz gehörende Zustandssequenz $P(O, Q|\lambda)$

   Lösung: *Viterbi-Algorithmus*

Das Trainingsproblem läßt sich wie folgt formalisieren:

geg.: die Symbolsequenz: $O = O_1 O_2 \ldots O_T$ und ein Modell $\lambda = (\pi, A, B)$

ges.: Wie lassen sich die Parameter $\lambda = (\pi, A, B)$ schätzen, um $P(O|\lambda)$ zu maximieren?

Lösung: Es gibt keine analytische Lösung dieses Problems, aber iterative Verfahren die sich einem lokalen Optimum annähern (Forward-Backward):

- *Baum-Welch*, ein EM-Verfahren
- Gradientenabstiegsverfahren

Kontinuierliche HMM

Wir haben die HMM über einer diskreten Zeichenfolge $O = O_1, \ldots, O_T$ definiert. Dieses diskretwertige HMM (Abb. 2) kann in der Spracherkennung

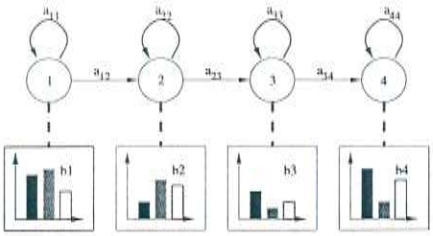


Abbildung 2: Diskretes HMM

durch einen vorgeschalteten Vektorquantisierer verwendet werden. Dann bezeichnen die O_i die Indizes der Referenzvektoren.

Wie bereits erläutert, ist eine Quantisierung mit einem Informationsverlust verbunden. Dieser kann durch kontinuierliche Ausgabeverteilungsdichten $b_j(x)$ umgangen werden, so daß in jedem Zustand zu jedem Eingabevektor die Emissions-Wahrscheinlichkeit berechnet werden kann. Die Verteilungsfunktionen $g_{jk}(x)$ werden in jedem Zustand j mit den Gewichten c_{jk} aufaddiert.

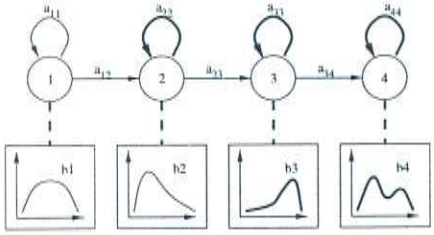


Abbildung 3: Kontinuierliches HMM

Bei diesem Verfahren ist die Wahl der Dichtefunktion fehlerbehaftet, denn die tatsächliche Dichtefunktion wird durch eine parametrische angenähert. Die Parameter einer solchen Funktion können z.B. bei Normal-, Binominal- oder Poissonverteilung geschätzt werden.

$$b_j(x) = \sum_{k=1}^{K} c_{jk} g_{jk}(x) \qquad \text{mit} \qquad \sum_{k=1}^{K} c_{jk} = 1 \tag{5}$$

Semikontinuierliche HMM

Die Kluft zwischen den diskreten und kontinuierlichen HMM schließen die semikontinuierlichen HMM[2].

$$b_j(x) = \sum_{k=1}^{K} c_{jk} g_k(x) \tag{6}$$

Die Dichtefunktionen werden in allen Zuständen wiederverwendet, so daß die Anzahl der benötigten Parameter erheblich reduziert wird. In Formel (6) drückt sich dies durch das Fehlen des Index j bei der Verteilungsfunktion $g_k(x)$ aus.

2.3 Wortuntereinheiten

Damit die Parameter des HMM gut geschätzt werden können, benötigt man eine große Anzahl von Mustern. Bei kleinen Vokabularien wie Zahlenwörtern oder Kommandosteuerungen ist es eventuell noch möglich, viele Äußerungen pro Wort zu erhalten. Je größer der Wortschatz wird, desto mehr Wörter werden statistisch ungenügend repräsentiert und deshalb schlecht modelliert. Ein weiterer Nachteil von wortbasierten Modellen ist die schlechte Erweiterbarkeit. Will man ein Wort zum Vokabular hinzufügen, müssen die Parameter dieses HMMs trainiert werden.

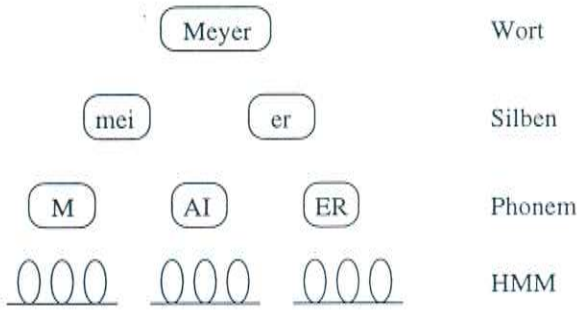


Abbildung 4: Von einem Wort zum HMM

Zur Beseitigung dieser Nachteile weicht man auf Wortuntereinheiten aus, aus denen sich ein Wort zusammensetzt. Dabei sollten möglichst wenige

[2]alternative Bezeichnung: *tied-mixture model*

Wortuntereinheiten benutzt werden, aus denen sich alle Wörter zusammensetzen. Für jede dieser Untereinheiten muß es genügend akustische Korrelate geben, damit die Modellierung möglichst exakt ist. Weiterhin sollte die Kontextabhängigkeit der Untereinheiten möglichst gering sein. Zu diesem Zweck existieren eine Vielzahl unterschiedlicher Phonemklassen. Welche Art unter welchen Bedingungen zu benutzen sind, kann in [ST95] nachgelesen werden.

2.4 Buchstabiererkennung

Damit wir mit den gleichen Begriffen arbeiten können wie bei anderen Spracherkennern auch, bezeichnen wir einen Buchstaben als Wort w_i und eine Buchstabensequenz[3] $\vec{w}$ als Satz.

Damit hat ein Buchstabiererkenner ein relativ kleines Vokabular mit sehr ähnlichen Wörtern, deren lautsprachliche Äußerungen eher kurz sind. Diese zu unterscheiden stellt das eigentliche Problem dar. Ansonsten gibt es keine elementaren Unterschiede zwischen einem Spracherkenner und einem speziellen Buchstabiererkenner.

2.5 Sprachmodelle

Überläßt man die Satzausgabe eines automatischen Spracherkenners ausschließlich der Entscheidung der akustischen Analyse, so ist jedes Folgewort eines erkannten Wortes möglich und muß in der Suche berücksichtigt werden. Der Satzaufbau ist jedoch bestimmten Regeln unterworfen, die sich formalisieren und in die akustische Analyse integrieren lassen. Leider neigen Menschen in einem natürlich-sprachlichen Dialog dazu, viele grammatikalische Gesetzmäßigkeiten zu vergessen. Des weiteren ist nicht nur die Syntax eines Satzes zu berücksichtigen, sondern ebenso die Semantik, Plausibilität hinsichtlich des Anwendungskontextes etc. Aus diesen Gründen ist ein Sprachmodell von Vorteil, das sich automatisch aus prototypischen Dialogen extrahieren läßt. Da sich die natürliche, spontane Sprache nicht an grammatikalische Regeln hält und aufgrund der geschlossenen mathematischen Theorie (siehe Formel (1)), erwiesen sich stochastische Modelle als besonders geeignet.

[3] Es wird in diesem Zusammenhang die Vektorschreibweise verwendet um darauf aufmerksam zu machen, daß es sich um eine Buchstabensequenz, also um ein Tupel handelt. Dieses Tupel ist jedoch nicht von fester Länge.

Stochastisches Modell

In einem stochastischen Modell wird die Auftrittswahrscheinlichkeit eines Wortes in Abhängigkeit von seinem Kontext betrachtet. Die Gesamtwahrscheinlichkeit einer Wortsequenz wird aus dem Produkt

$$P(\vec{w}) = P(w_1, ..., w_n) = P(w_1)P(w_2|w_1)...P(w_n|w_1...w_{n-1}) \quad (7)$$

gebildet. Damit hängt die Identität eines Wortes prinzipiell von der gesamten Historie ab. Aufgrund der begrenzten Rechnenkapazitäten und der beschränkten Korpora zum Bestimmen der Wahrscheinlichkeiten ist die Berücksichtigung des gesamten Kontextes jedoch nicht möglich. Die Beobachtung, daß der unmittelbare Kontext entscheidender ist als der weit entfernte, führt zu Grammatiken, die nur $n-1$ Wörter der Vergangenheit berücksichtigen. Diese nennen wir *N-Gramm-Grammatik.*

$$P(w_i|w_1...w_{i-1}) \approx P(w_i|w_{i-n+1}...w_{i-1}) \quad (8)$$

Die Häufigkeitsanalyse wird über einem endlichen Text durchgeführt. Einige Wortkombinationen werden in diesem Text nicht oder nur einmal vorkommen und deshalb sehr kleine Wahrscheinlichkeitswerte haben. Eine Anhebung der kleinen Wahrscheinlichkeitswerte auf Kosten der großen nennt man *Glättung.*

In dieser Diplomarbeit wird hauptsächlich Einzelworterkennung betrieben, bei der keine Historie berücksichtigt werden kann. Jedoch kann eine Häufigkeitsanalyse in Form von Monogrammen verwendet werden. Für die Buchstabiererkennung, bei der die Buchstaben in diesem Zusammenhang Wortsequenzen darstellen, kann die Grammatik auch ein Graph sein.

Im Gegensatz zu stochastischen Modellen legt ein Graph die Syntax des Satzes fest. Die einleitende Diskussion zu Sprachmodellen ließ diesen Ansatz unbrauchbar erscheinen. Anders verhält es sich jedoch bei der Buchstabierung von Namen. Hierbei ist die Buchstabensequenz eindeutig durch den zu buchstabierenden Eigennamen vorgegeben. Natürlich kann man davon ausgehen, daß Menschen nie ganz fehlerfrei buchstabieren, aber die Fehlerrate ist relativ klein. Von 2900 buchstabierten Namen, die vom Blatt abgelesen wurden, sind 115 fehlerhaft buchstabiert worden ($\approx 4\%$).

Eine Grammatik, die aus Buchstabensequenzen gültige Eigennamen erzeugt, läßt sich durch einen endlichen Automaten, einen Graphen oder einen

Baum realisieren. Der MS-TDNN-Buchstabiererkenner kann u.a. eine Baumgrammatik verwenden. Die Baumstruktur hat den Vorteil gegenüber einem minimalen Graphen, daß keine Rückwärtsverweise benötigt werden. Sobald ein Blatt erreicht wurde, ist der gesamte Pfad bekannt.

2.6 Güte der Erkennungsleistung

Um die Erkennungsleistung messen zu können, bedarf es eines Abstandmaßes zwischen dem korrekten Wort oder Satz und dem erkannten. Ein allgemein anerkanntes Gütekriterium für Spracherkennung sind die Wort- und Satzakkuratheit. Es wurde vorgeschlagen von NIST[4], basierend auf dynamischer Programmierung: [PGMF86]

N_{all} : Anzahl der korrekten Wörter im Referenzsatz
N_{sub} : Anzahl der substituierten Wörter
N_{del} : Anzahl der gelöschten Wörter
N_{ins} : Anzahl der eingefügten Wörter
N_{kor} : Anzahl der vollständig korrekt erkannten Sätze
N_{sall} : Anzahl der korrekten Sätze

$$\begin{aligned} WA &= 1 - \frac{N_{sub} + N_{del} + N_{ins}}{N_{all}} \\ SA &= \frac{N_{kor}}{N_{sall}} \end{aligned} \tag{9}$$

Bei der Einzelworterkennung besteht ein Satz nur aus einem Wort. Somit ist die Wortgenauigkeit gleich der Satzgenauigkeit und wird im Laufe dieser Arbeit auch als *Namensgenauigkeit* bezeichnet. Bei der Buchstabierung eines Namens kann die Sequenz der Buchstaben als Satz aufgefaßt werden. Die Namensgenauigkeit entspricht dann der Satzgenauigkeit, die Wortgenauigkeit wird dann als *Buchstabengenauigkeit* bezeichnet.

[4]The United States National Institute of Standards and Technology

2.7 Verwendete Erkenner

2.7.1 JANUS

Der Name "Janus", als Sinnbild eines Mannes mit zwei Gesichtern, stand Pate für ein Spracherkennungs- und Übersetzungsprojekt. Das Ziel dieses Projektes ist die Übersetzung von gesprochener Sprache in eine andere Sprache. In dieser Diplomarbeit wurde nur der Teil des JANUS benötigt, der für die Spracherkennung verantwortlich ist. Der Begriff *JANUS* bezeichnet im weiteren ausschließlich diesen Teil des Gesamtsystems.

Der Spracherkenner JANUS wurde mit dem Interpreter Tcl und dem Toolkit Tk [Ous94] kompiliert. Die Experimente, die im Rahmen dieser Diplomarbeit durchgeführt wurden, konnten hauptsächlich durch Tcl-Skripte implementiert werden. Zur Anbindung der Gradient-Box, einer Hardware zur Aufnahme und Digitalisierung und zur Beschleunigung des Viterbi-Algorithmus, wurden einige neue Funktionen in C programmiert.

Die Grundkonfiguration, die für alle Versuche gleich war, betrifft die Signalvorverarbeitung und die akustische Modellierung. Diese werden zu Beginn eines Erkennungsprozesses vom Erkenner geladen. Während der Signalvorverarbeitungsphase werden alle Merkmale aus dem Sprachsignal extrahiert, die für den Erkennungsprozeß relevant sind: FFT, MEL, IMEL, LPC, etc. Es werden dann schon trainierte Emissions- und Übergangswahrscheinlichkeiten für die HMM geladen. Diese entstanden aus den offiziellen "Verbmobil"-Evaluationsdaten[FGH+97] von 1996. Dies waren Sprachdaten bezüglich einer spontansprachlichen Terminabsprache mit ca. 5000 Wörtern Vokabular und einer erzielten Wortgenauigkeit von 86.2%. Die Suche in JANUS wurde besonders auf kontinuierliche Spracherkennung abgestimmt. Eine schnelle Einzelworterkennung stand deshalb nicht im Vordergrund. Folgende Suchverfahren wurden benutzt:

- *Viterbi:* Die Funktion des Viterbi-Algorithmus wurde bereits im Kapitel 2.2.2 beschrieben. Um die beste Hypothese aufgrund des akustischen Modells zu bestimmen, wird die Gesamtbewertung aller Wortmodelle berechnet und das mit der besten Bewertung ausgewählt. Auf der Suche nach dieser Hypothese sind viele Operationen überflüssig. Sollte bereits eine bessere Bewertung gefunden worden sein als die aktuelle, kann vorzeitig abgebrochen werden. Der Zeitgewinn wird bei der Gegenüberstellung des normalen mit dem schnellen Viterbi-Algorithmus in Abbildung 5 deutlich.

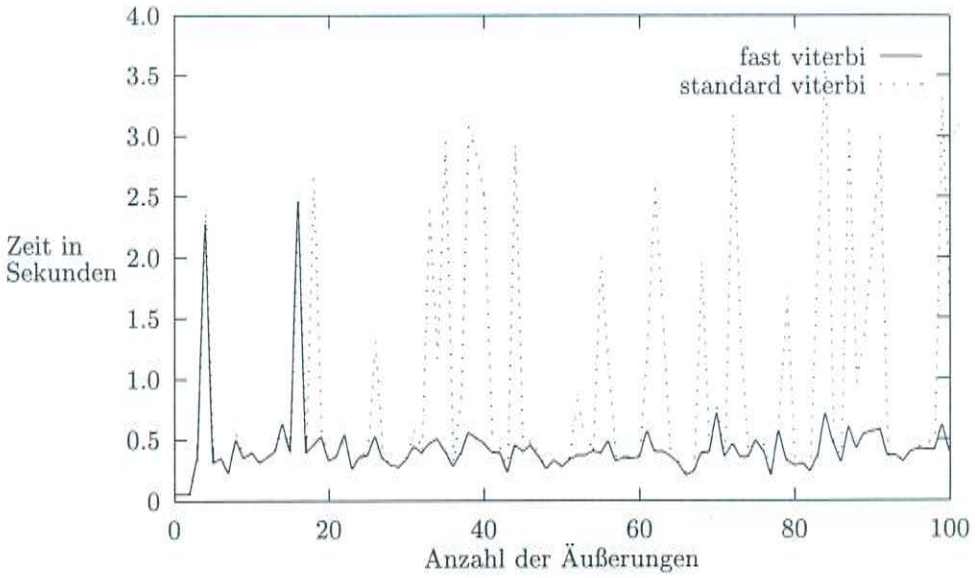


Abbildung 5: Gegenüberstellung des normalen mit dem schnellen Viterbi.

- *Viterbi mit Varianten:* Eine andere Möglichkeit, die Suche zu beschleunigen, ist die gleichzeitige Berechnung aller Bewertungen. Zu der momentan besten wird ein Offset (auch Beam genannt) addiert und alle Hypothesen, die innerhalb dieses Beams liegen, werden zur weiteren Suche benutzt. Die anderen werden nicht weiter betrachtet. Es können Fehlentscheidungen entstehen, wenn der Beam zu gering gewählt wurde und die richtige Hypothese aufgrund einer schlechten akustischen Übereinstimmung früh aus dem Suchbereich fällt.

  Eigentlich ist für die Bestimmung der besten Hypothese der Forward-Algorithmus zu verwenden, da dieser die klassenbedingte Wahrscheinlichkeit bestimmt. Da jedoch die Zustandssequenz wertvolle Informationen liefern kann, wird der Viterbi-Algorithmus verwendet.

- *Suche:* Der eigentliche Decoder ist für spontane, kontinuierliche Sprache ausgelegt und optimiert. Dadurch entstehen bei der Einzelworterkennung Einfügefehler. Der Übergang von einem Wort in ein anderes kann mit einer sogenannten Wortübergangsstrafe belegt werden, ganz vermeiden lassen sich Einfügefehler dadurch jedoch nicht.

2.7.2 MS-TDNN

Hinter der Abkürzung MS-TDNN verbirgt sich ein konnektionistischer Erkenner, der ein *Multi-State Time Delay Neural Network*[HW93] verwendet. Der schematische Aufbau des Erkenners ist in Abbildung 6 dargestellt.

Er nutzt das zeitinvariante Verhalten des TDNN[WHH+89] zum Berechnen der Bewertung für Phoneme. Da jeder Buchstabe über eine Phonemsequenz moduliert wird, kann mittels dynamischer Programmierung ein optimaler Pfad für jeden Buchstaben gefunden werden. Die Aktivierungen entlang des Pfades werden in einem Buchstabenausgabeknoten aufsummiert.

Um aus einer hypothetisierten Buchstabensequenz einen Namen zu ermitteln, wird nach dem akustischen Modell ein Sprachmodell geschaltet. Dieses kann syntaktisch oder statistisch sein. Ein syntaktisches Sprachmodell besteht aus einer möglicherweise auch unendlichen Menge von gültigen Sätzen. Nur diese können von dem Erkenner akzeptiert werden. Die gültigen Sätze können durch einen endlichen Graphen modelliert werden. Betrachten wir jeden Knoten als ein Wort und die Wortübergänge als Kanten, so stellen wir fest, daß es zur Modellierung der gültigen Sätze notwendig ist, Knoten mit demselben akustischen Modellen mehrfach zu erzeugen, da sie unterschiedliche Vorgängerknoten haben. Da aber jeder Knoten Speicherresourcen benötigt, sollte der Graph möglichst gering sein und dabei noch eine effiziente Suche ermöglichen. Dazu kann man einen minimalen Graphen erzeugen. Ein minimaler Graph benötigt weniger Knoten als ein Baum. Trotzdem bietet die Baumstruktur einen entscheidenden Vorteil: Es werden keine Rückwärtszeiger benötigt, denn sobald man einen finalen Knoten erreicht hat, ist auch die beste Wortsequenz bekannt.

Durch die Modellierung als Suchbaum hat man zwar alle gültigen Sätze modelliert, aber die Auftrittswahrscheinlichkeiten der einzelnen Sätze nicht berücksichtigt. Sollten diese nicht bekannt sein, so hat man nichts verloren, anderenfalls können Gesamtwahrscheinlichkeiten im finalen Knoten oder Übergangswahrscheinlichkeiten an den Kanten diese Information mitberücksichtigen.

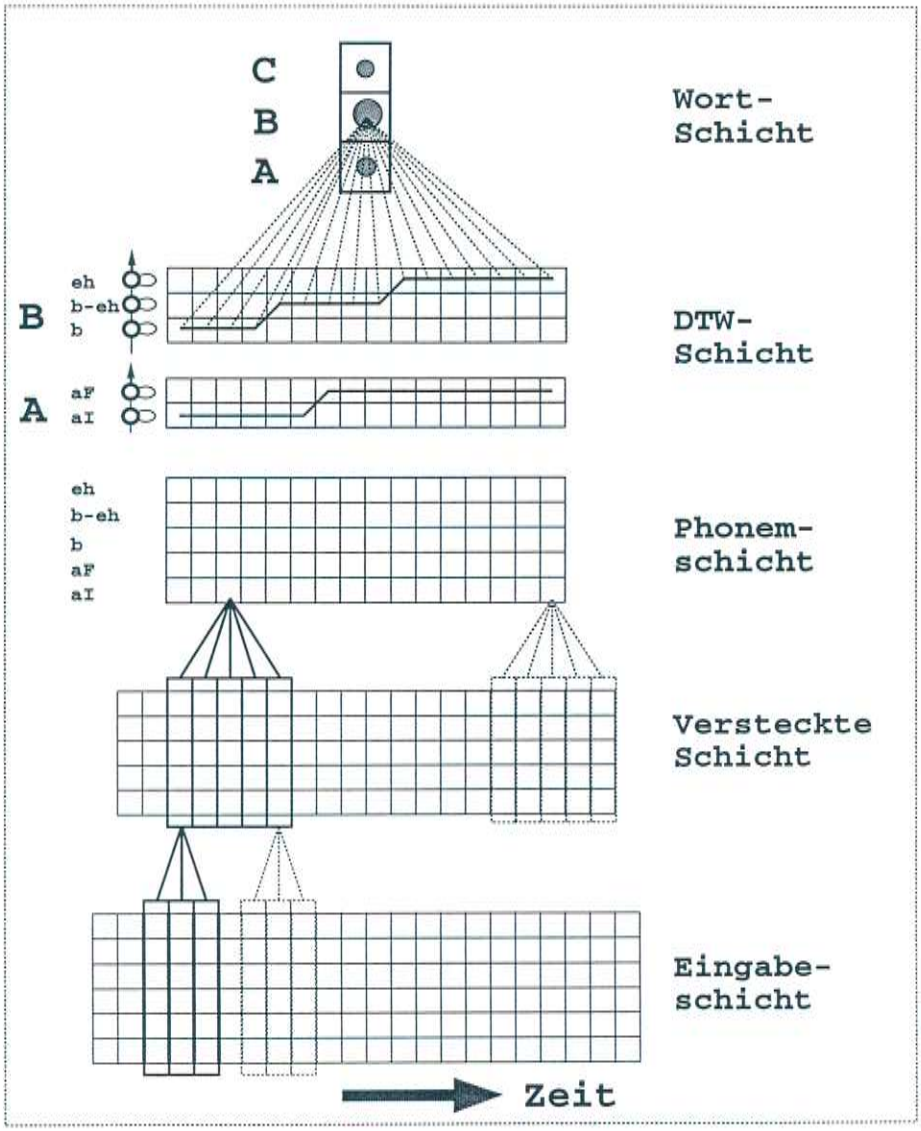


Abbildung 6: Schematischer Aufbau des MS-TDNN.

3 Nähere Betrachtung der Aufgabenstellung

3.1 Überblick

Es wurde bisher beleuchtet, wie Spracherkennung grundsätzlich funktioniert. Im weiteren wird die Erkennung von gesprochenen bzw. buchstabierten Eigennamen besprochen. Ein Eigenname bezeichnet "*eine einzelne, als Individuum oder individuelles Kollektiv gedachte Person oder Sache zum Zweck der eindeutigen Identifizierung und Benennung...Eigennamen bezeichnen ein Objekt unabhängig von der Bedeutung des Wortes nur aufgrund des Lautkomplexes...*"[5][Gü91]. In dieser Arbeit werden als Eigennamen Nachnamen benutzt. Dabei handelt es sich hauptsächlich um Nachnamen des deutschsprachigen Raumes. Einige Vertreter nicht typisch deutscher Nachnamen, die ebenfalls benutzt wurden sind z.B.: Delgado, Rocha, Soto, Garces, Rodriguez, Sanchez, etc. Ebenfalls zu berücksichtigen sind Doppelnamen, die sich aus mehreren Einzelnamen zusammensetzen können: "Rodriguez Gomez". Einer Erweiterung auf Vornamen, Straßennamen oder eine andere Sprache steht nichts im Wege, da in keiner Weise auf typische Eigenschaften deutscher Nachnamen eingegangen wird.

Weshalb lohnt sich eine Beschäftigung mit diesem Thema? Betrachten wir einmal folgendes Szenario: Jemand ruft bei der Sekretärin eines Instituts an und möchte die Telefonnummer von Hermann Hild. Diese Anfrage kann folgende Formen haben:

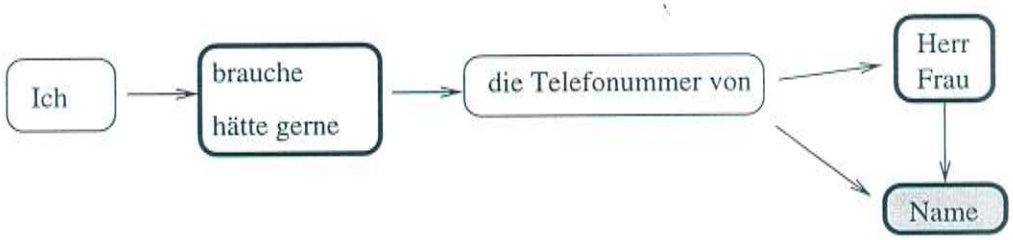


Abbildung 7: Ein einfacher Wortgraph.

Die Möglichkeiten, wie eine Anfrage nach einer Telefonnummer formuliert werden, scheinen endlos. Viele könnten durch einen Graphen wie in Ab-

[5]Eine interessante Eigenschaft der Eigennamen ist die prinzipielle Unübersetzbarkeit. Wenn jedoch eine durchsichtige Bedeutung, wie zum Beispiel bei "Schwarzwald" vorliegt, wird von dieser Regel abgewichen: Black-Forest, Fôret Noir.

bildung 7 beschrieben werden. Allgemein lassen sich solche Anfragemuster durch eine Grammatik beschreiben und für eine zielgerichtete Suche nutzen. Die Problematik verbirgt sich jedoch in dem Platzhalter für den Namen, denn wird die Frage an die Sekretärin des Instituts gestellt so sind vielleicht 20 Namen möglich (Abbildung 8).

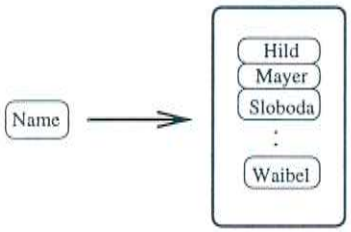


Abbildung 8: Perplexität

Die Perplexität[6] des Sprachmodells steigt durch den Platzhalter "Name" an. Der Kontext hilft nur sehr bedingt weiter. Wenn "Herr" oder "Frau" vorher gesagt wurde, lassen sich einige Namen ausschließen, manche Namen werden häufiger nachgefragt, oder es wird über eine Stimmenanalyse und eine Statistik ermittelt, wer wen oft sprechen will. Durch die Forderung, ein Objekt möglichst eindeutig zu identifizieren, gibt es eine Vielzahl von Nachnamen. Bei der Telefonauskunft der Stadt Karlsruhe mit ca. 270.000 Einwohner müssen weitere Techniken verwendet werden.

Eine weitere Problematik stellt die akustische Verwechselbarkeit der Eigennamen dar. Zum Beispiel gibt es vom Eigennamen "Meier" die Varianten "Mayer", "Maier" und "Meyer". Diese Varianten werden alle ähnlich ausgesprochen. Hiermit ist eine Anfrage schon allein aus diesem Grund mehrdeutig. Die Sekretärin müßte nachfragen, um welche Variante es sich in diesem Fall handelt. Die Antwort wird eine buchstabierte Teilmenge des Namens sein. Entweder der Form "Meyer mit E Y", "Martha Emil Ypsilon Emil Richard" oder "Meyer, M E Y E R".

Im wesentlichen werden in dieser Diplomarbeit zwei Aspekte der Erkennung gesprochener Eigennamen näher beleuchtet:

- Wie erfolgte die Spracheingabe des Eigennamens. Im nächsten Unterkapitel wird auf diesen Aspekt näher eingegangen.

[6]Die Perplexität gibt den mittleren Verzweigungsgrad der Sprachmodells an.

(1)	Bitte sagen Sie Ihren Namen:	"Schmidt"
	Bitte buchstabieren Sie ihn:	"S C H M I D T"
(2)	Bitte sprechen und buchstabieren Sie ihren Namen:	"Schmidt, S C H M I D T"
(3)	Wie lautet Ihr Name:	"Ich heiße **Schmidt**, das buchstabiert man **S C H M I D T**"

Tabelle 1: Drei Szenarien zum Sprechen und Buchstabieren von Eigennamen.

- Aus wievielen Namen soll der richtige Name ermittelt werden. Ist die Aussprache der Namen bekannt?

Die Lösungsansätze werden anhand dieser zwei Punkte gegliedert. Sukzessive wird daraus eine Tabelle aufgebaut, deren Inhalte im Folgenden näher erklärt werden.

3.1.1 Beschreibung der Anfrage

Hier geht es um die Freiheit des Benutzer beim Formulieren seiner Anfrage. Je freier der Benutzer seine Anfrage formulieren kann, desto flexibler muß der Erkenner sein. Drei Szenarien mit steigender Komplexität sind in Tabelle 1 dargestellt. Es sind jedoch mehr Spielarten möglich und wir werden uns in Kapitel 7 einer Anfrage zuwenden, die eine Erweiterung von Szenario 2 ist.

1. Will man den gesprochenen Namen und den buchstabierten Namen getrennt vorliegen haben, kann man den Benutzer durch einen restriktiven Dialog dazu bewegen, zunächst nur seinen Namen zu sagen und ihn anschließend buchstabieren zu lassen. Eine weitere Möglichkeit besteht darin, daß der Benutzer explizit angibt an welcher Stelle er den Namen spricht und wann er buchstabiert.

   Durch die explizite Trennung der beiden Varianten kann man die Erkennung unabhängig voneinander durchführen und die Ergebnisse später wieder vereinigen.

Zum einen stellt diese Vorgehensweise eine große Hilfe für den Erkennungsprozeß dar, die bessere Erkennungsleistungen erwarten läßt. Auf der anderen Seite erkauft man sie sich durch Einschränkung des Benutzungskomforts.

2. Wird der Benutzer wie in Szenario 2 aufgefordert den Eigennamen zu sprechen und zu buchstabieren, so verliert man die Information, an welcher Stelle im Sprachsignal der gesprochene Name aufhört und der buchstabierte anfängt.

   Man hat jetzt die Möglichkeit die Trennlinie zwischen den beiden zu ermitteln und dann die Erkennung wie im ersten Fall durchzuführen.

   Im allgemeinen sollte versucht werden, nicht schon mit einem Eingangsfehler den Erkennungsprozeß zu beginnen. Dieser Fehler würde sich durch alle Phasen fortpflanzen und ließe sich nur schwer im weiteren Verlauf beheben. Was auf die Sprachvorverarbeitung zutrifft, gilt im Besonderen auch für das Auffinden der Trennlinie. Diese kann nur zu einem gewissen Maße korrekt bestimmt werden und eine falsche Trennlinie verschlechtert das Erkennungsergebnis. Erfolgsversprechender scheint auf den ersten Blick der Versuch das Wörterbuch des Erkenner so zu ändern, daß der gesprochene und buchstabierte Namen zusammen den Eigennamen beschreibt. Das Modell für einen Namen wird also aus dem gesprochenen und dem buchstabierten Modell zusammengesetzt.

3. Der gesprochene und buchstabierte Namen ist eingebunden in einen Satz. Für den Benutzer ist dies die natürlichste Art und Weise, wie er seine Anfrage formulieren kann. Der Erkennungsprozeß muß zunächst mögliche Positionen für Eigennamen finden [Thi93]. Dabei kann ein Sprachmodell sehr hilfreich sein. Durch dieses ist es z.B. möglich, Kontextwissen zu dem Namen zu extrahieren, wie zum Beispiel, ob es sich um einen Mann, oder Frau handelt, ob der Name buchstabiert oder gesprochen wurde, etc. Wurden mögliche Stellen gefunden[7], dann kann das Problem wieder auf den ersten Fall reduziert werden, wobei auch hier mit einem Segmentierungsfehler zu rechnen ist.

[7]Solche Platzhalter, denen zunächst kein Wort aus dem Vokabular zugeordnet werden können, werden als "out of vocabulary" Wörter, oder kurz OOV bezeichnet.

3.1.2 Informationen über die Eigennamen

Ein Spracherkenner benötigt grundsätzlich ein Vokabular. Die Elemente dieses Vokabulars nennen wir Wörter. In Abbildung 4 war der Zusammenhang von Wörtern zu Wortuntereinheiten ersichtlich. Bei den Wörtern handelt es sich jetzt um Eigennamen. Zu den gesprochenen Eigennamen werden die passenden Modelle zusammengesetzt. Dabei ist zu unterscheiden, ob es sich um buchstabierte oder gesprochene Varianten handelt. In sofern brauchen wir die Information, wie die Eigennamen ausgesprochen werden. Bei einem buchstabierten Namen hat man den Vorteil, daß schon Wortuntereinheiten zur Verfügung stehen, nämlich die Buchstaben, deren Aussprachewörterbuch ca. 30 Einträge beinhaltet und aus dem sich dann jeder Name extrahieren läßt. Für die kontinuierlich gesprochenen Namen ist dies nicht so einfach möglich. Hierzu existieren Aussprachedatenbanken, die im Anhang beschrieben sind. Kann aufgrund einer speziellen Anwendung die Anzahl der Eigennamen eingeschränkt werden, so läßt sich damit auch das Vokabular und damit auch das Wörterbuch reduzieren.

Eine weitere Information, die für den Erkennungsprozeß von Nutzen sein kann, ist die a priori Wahrscheinlichkeit. Soweit vorhanden kann man die Häufigkeit der Namen nutzen, um die Entscheidung für einen Namen zu beeinflussen. So kann über eine repräsentative Menge die Häufigkeit der Namen bestimmt werden, um diese Information als Sprachmodell zu verwenden (Monogramm Wahrscheinlichkeiten). Die Bestimmung der Häufigkeiten sollte möglichst nahe an den tatsächlichen Daten liegen. Das kann in manchen Anwendungen nicht leicht sein. In dem anschaulichen Beispiel mit der Institutssekretärin, kann über mehrere Tage (besser Wochen) diese Häufigkeiten ermittelt werden. Die Problematik von nie nachgefragten Namen, die in diesem Fall eine Nullwahrscheinlichkeit erhalten würden, wurde im Unterkapitel *Sprachmodelle* diskutiert.

Die Experimente wurden deshalb auch in Abhängigkeit des zur Verfügung stehenden Wörterbuchs durchgeführt:

1. **Liste der Eigennamen:** In allen Experimenten gehen wir davon aus, daß wir über eine Liste der Eigennamen verfügen, die erkannt werden sollen. Ein Auskunftsystem kann dann in seiner Datenbank den entsprechenden Namen finden.

2. **Transkriptionen der gesprochenen Namen:** Die Beschreibung der Aussprache eines Eigennamens ist nur für die gesprochenen Namen not-

Information über die Eigennamen		Anfrage Szenarien	
Stichwort	Beschreibung	Szenario 1	Szenario 2
klein	Namensliste bekannt Anzahl der Namen gering Transkription bekannt	Kapitel 4	
mittel	Namensliste bekannt Anzahl der Namen groß Transkriptionen bekannt	Kapitel 5	
groß	Namensliste bekannt Anzahl Namen groß Transkriptionen nicht bekannt	Kapitel 6	

Tabelle 2: Überblick und Struktur über die folgenden Kapitel.

wendig. Für die buchstabierten kann man sie relativ einfach automatisch erzeugen. Bei den gesprochenen Namen ist man auf eine Datenbasis, oder Handarbeit angewiesen. Da man nicht mit der Vollständigkeit dieser Datenbasis rechnen kann, wird ein Ansatz diskutiert, der ohne diese Information auskommt.

3. **Größe des Wörterbuches:** Diese Größe hängt in erster Linie von der Anzahl der Namen ab, die erkannt werden sollen also dem Vokabular. Dies ist aber nur eine untere Grenze, denn zum einen können Namen Aussprachevarianten haben, zum anderen kann gefordert sein, daß der Erkenner auch noch andere Wörter erkennen soll. Aus technischer Sicht begrenzt die Leistungsfähigkeit des Erkenners die Größe des Wörterbuches. Wir werden hier keine konkrete Zahl angeben, da die Rechenkapazitäten und Leistungsfähigkeit der Erkenner ständig steigen. Die Grundsätzliche Problematik bleibt jedoch erhalten.

3.1.3 Struktur der Lösungsansätze

Aus den Überlegungen die zu den Anfrage Szenarien und dem Wissen über die Eigennamen diskutiert wurden, ergibt sich die in Tabelle 3.1.3 dargestellte Struktur.

Jeder Zeile wird in einem eigenen Kapitel behandelt. Dabei werden Lösungsansätze diskutiert und durch Experimente auf Eignung untersucht.

Die Stichwörter *klein, mittel* und *groß* werden in den folgenden Kapitel wiederverwendet, damit auf die ausführliche Beschreibung verzichtet werden kann und die Tabelle dadurch übersichtlicher wird. Die mittlere Zeile (*mittel*) bedeutet: Die Namensliste ist so umfangreich, daß sie als Vokabular für den Erkenner zu groß ist, jedoch für alle Namen die Transkription noch zur Verfügung stehen.

3.2 Verwandte Arbeiten

Einige Forschungsarbeiten haben sich bereits mit der Erkennung von gesprochenen oder buchstabierten Eigennamen beschäftigt. In den meisten Fällen mit dem Ziel, ein Auskunftsystem zu erstellen. Da dabei eine hohe Erkennungsrate unabdingbar ist, wird versucht über Sprachmodelle oder Interaktion mit dem Benutzer zusätzliche Informationen zu gewinnen. In [KSS95] werden die Wortgenauigkeiten von gesprochen und buchstabierten Eigennamen bei unterschiedlich großen Namenslisten gegenübergestellt. Dabei wird sehr deutlich, daß der buchstabierte Name besser erkannt werden kann als der gesprochene. Bei 1200 Namen wurde der buchstabierte mit ca. 92%, der gesprochene dagegen nur mit ca. 70% erkannt. Aus diesem Grund ist es nicht verwunderlich, daß sich die Mehrheit der Forschungsarbeiten in diesem Gebiet mit Buchstabiererkennung beschäftigen. Zur Verbesserung der Erkennungsleistung wird jedoch nicht der gesprochene Name verwendet, sondern in Dialog mit dem Benutzer getreten.

Ein Buchstabiererkenner, der in 4 Schritten das Sprachsignal analysiert, Buchstabenhypothesen erstellt und ein Vergleich der Hypothese mit den Namen in der entsprechenden Namensliste durchführt wird in [JVFM95] beschrieben. Dieser Erkenner erreicht bei 3388 Namen im Vokabular eine Namensgenauigkeit von 95.3%.

In [YTTK89] werden Ergebnisse für "sehr große Vokabularien" vorgestellt. Hierbei handelt es sich um eine Vokabulargröße von 80000 Wörtern, davon 71251 Namen. Die Eigennamen werden in einem kontinuierlichen Satz gesprochen und von einem sprecherunabhängigen HMM-Erkenner erkannt. Die Erkennungsrate von 88% Wortgenauigkeit ist auf das Sprachmodell zurückzuführen, welches nicht nur die Häufigkeiten der Namen, sondern auch den Kontext des Satzes analysieren kann. Damit sinkt die Perplexität des Sprachmodells und die Ergebnisse lassen sich nur bedingt mit denen in meiner Ar-

beit vergleichen.

Als Ausgangspunkt wird untersucht, wie gut die mir zur Verfügung stehenden Erkenner auf den gesammelten Sprachdaten sind. Diese Ergebnisse werden im nächsten Unterkapitel präsentiert.

3.3 Einführende Experimente (Baselines)

3.3.1 Isoliert kontinuierlich gesprochener Name

Zur Erkennung des isoliert gesprochenen Namens wurden die Daten, wie im Anhang beschrieben, verwendet. Die Resultate werden in Abhängigkeit der verwendeten Suche präsentiert. Aufgrund der Auslegung der *Suche* auf kontinuierlich gesprochener Sprache und damit auch auf ganze Sätze, werden leider viele Einfügefehler gemacht, die das Erkennungsergebnis verschlechtern. Auch die Erhöhung einer Wortübergangsstrafe kann dies nur bedingt verhindern. Bei Verwendung der *Viterbi-Suche* können solche Einfügefehler nicht entstehen. Es wird bei dieser Suche die Häufigkeit der Namen nicht berücksichtigt.

JANUS	
Kommentar	**Liste 1**
Suche ohne Häufigkeiten	57.8%
s.o. mit Häufigkeit	60.0%
viterbi ohne Häufigkeiten	59.2%

Tabelle 3: Erkennunsergebnisse der kontinuierlich gesprochenen Namen

Die Abhängigkeit der Namensgenauigkeit in Bezug auf die Vokabulargröße ist in Abbildung 9 ersichtlich. Bei einer Vokabulargröße von über 3000 Namen sank die Namensgenauigkeit auf unter 50%. Solange die Vokabulargröße kleiner gleich der Anzahl der Namen war, zu der Transkriptionen vorhanden sind und zu denen Testäußerungen existieren, wurden aus dieser Liste die Namen zufällig gezogen. Wird diese Grenze überschritten, so werden beliebige neue Namen aus der Transkriptionstabelle dazu genommen. Die Versuche wurden mit 57 Sprechern der Qualitätsstufen[8] Null und Eins durchgeführt (siehe Liste 1), zu denen Transkriptionen vorhanden sind.

[8]Die Qualitätsstufen werden im Anhang beschrieben.

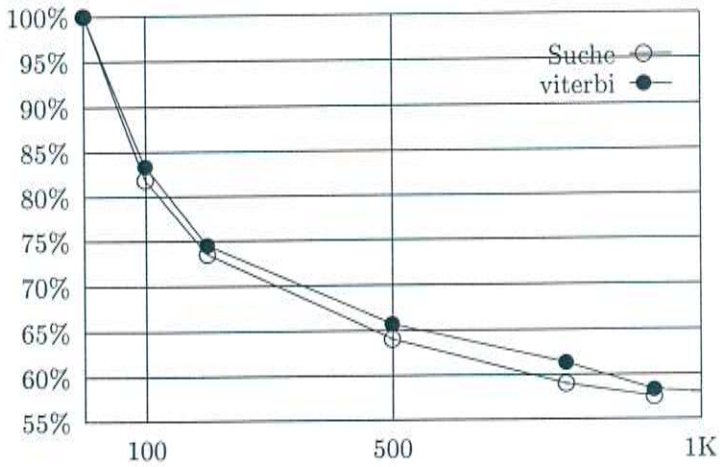


Abbildung 9: Verhalten bei wachsendem Vokabular bis 1000 Namen. Ergebnisse in Prozent Namen korrekt.

Die schlechteren Ergebnisse von 60% Namensgenauigkeit, gegenüber der Wortgenauigkeit des JANUS während der Verbmobil-Evaluation, haben mehrere Ursachen:

1. Die Perplexität der in etwa gleichwahrscheinlichen Eigennamen liegt bei über 900 und ist damit um Faktor 20 größer als bei der Verbmobil-Evaluation. Aus Abbildung 9 ist ersichtlich, daß bei ca. 90 verschiedenen Eigennamen, mit einer Namensgenauigkeit von 86% zu rechnen ist. Die Versuche wurden auch mit der "Suche", die für spontane Sprache optimiert ist, durchgeführt. Da dabei einige Einfügefehler entstehen, fällt die Namensgenauigkeit stets etwas schlechter aus, als wenn der Viterbi-Algorithmus verwendet wird.

2. Der JANUS-Erkenner ist nicht auf Einzelwörter, sondern auf kontinuierlicher und spontaner Sprache trainiert. Erschwerend kommt hinzu, daß er ausschließlich Wörter erkennen muß, auf denen er niemals trainiert wurde.

3. Das Wörterbuch der Eigennamen wurde maschinell erstellt und ist fehlerbehaftet. Die Datenbank der Eigennamentranskriptionen (im SAMPA-Format) wurde in das JANUS eigene Format transformiert. Bei dieser Transformation entstanden einige Ambiguitäten, die anhand

von ähnlichen Einträgen im Verbmobil-Wörterbuch gelöst wurden. Damit ist unklar inwieweit die ONOMASTICA-Transkriptionen, deren Konventionen und Herkunft im Anhang beschrieben werden, mit den Konventionen des JANUS Wörterbuches konsistent sind.

Konvention	Phoneme		
SAMPA	ZZ	PF	OEHR
JANUS	TSCH	P F	OEH R

4. Eigennamen enthalten viele untypische Phonemsequenzen und insbesondere ausländische Familiennamen, die in unserer Sprachsammlung häufig vorkommen, können sehr unterschiedlich ausgesprochen werden. Ihre automatisch erstellte Transkription ist deshalb fragwürdig.

3.3.2 Isoliert buchstabierter Name

Buchstabierte Namen können sowohl von JANUS als auch von MS-TDNN erkannt werden. Es werden die unterschiedlichen Ergebnisse der beide Erkenner in Bezug auf die benutzten Sprachmodelle in Tabelle 4 zusammengefaßt. Da es sich bei einem buchstabierten Eigennamen um eine Sequenz von Buchstaben handelt, wird die Erkennungsgenauigkeit in Wortgenauigkeit[9] und Satzgenauigkeit[10] unterschieden.

Die Tests wurden über zwei Testmengen durchgeführt. Zum einen über der Liste der Äußerungen, von denen auch Transkriptionen vorhanden waren, damit man die Ergebnisse mit denen der gesprochenen Test vergleichen kann (Liste 1). Zum anderen über alle Äußerungen von 57 Sprechern (Liste 2).

Das Verhalten bei wachsenden Vokabularien ist für den Buchstabiererkenner in [BH95] ersichtlich. Bei 800000 verschiedenen Namen wurde eine Namensgenauigkeit von fast 90% erzielt.

Da es in dieser Arbeit um die Erkennung der Eigennamen geht, ist das wichtige Ergebnis die Satz- bzw. Namensgenauigkeit. Wie aus der Tabelle 4 abgelesen werden kann, ist die Namensgenauigkeit des MS-TDNN besser als die des JANUS-Erkenners. Da wir aber für die weiteren Versuche den Zusammenhang zwischen dem gesprochenen und buchstabierten Namen untersuchen wollen, ist es für einige Experimente nötig, die Buchstabenerkennung ebenfalls mit JANUS durchzuführen. Deshalb wurden folgende Bemühungen

[9] hier: Buchstabengenauigkeit
[10] hier: Namensgenauigkeit

MS-TDNN			
Kommentar	**Liste 1**		**Liste 2**
	Wort	Satz	Satz
ohne LM	88.7%	50.9%	50.9%
TREE	97.7%	**95.6%**	95.5%
TREE (lp=4)	98.1%	**96.5%**	94.9%
TREE (r)	98.4%	96.9%	96.9%
JANUS			
Suche ohne LM	67.7%	13.1%	13.1%
Suche mit trigram	81.3%	31.0%	31.1%
Suche mit Namen	90.7%		89.4%
s.o. mit Häufigkeiten	92.3%		88.8%
viterbi	**93.3%**		89.9%

Tabelle 4: Vergleich der Buchstabier- und Namensgenauigkeit von Janus und MS-TDNN

unternommen, die zu einer Namensgenauigkeit von über 93% führten: Eine Häufigkeitsanalyse der Buchstabensequenzen wurde über 3000 Namen berechnet. Dabei wurden Wörter wie "doppelt", "scharfes" und "Bindestrich" mitberücksichtigt. Damit wurde die Namensgenauigkeit verbessert, war aber immer noch deutlich unter der Erkennungsleistung des MS-TDNN. Es muß der Vorteil ausgenutzt werden, daß die Namensliste bekannt ist und nur diese Namen erkannt werden können. D.h. für jeden Namen wird ein buchstabiertes Modell konstruiert. Damit steigt die Anzahl der Einträge im Wörterbuch. Die Angabe der Wortgenauigkeit bezieht sich in diesem Zusammenhang auf den ganzen Namen und ist damit identisch mit der Satzgenauigkeit, ansonsten würde die Buchstabengenauigkeit bei der Viterbi-Suche 96.3% betragen. In Tabelle 5 werden die Größe der Wörterbücher und Vokabularien aufgeführt. Das Absinken der Erkennungsleistung von Liste 1 auf Liste 2, ist durch die Zunahme des Vokabulars zu erklären. Die Diskrepanz zwischen Vokabular- und Wörterbuchgröße ist auf eine Vielzahl von Aussprachevarianten bei der Buchstabierung zurückzuführen. Betrachten wir zum Beispiel die Möglichkeiten, die sich bei der Buchstabierung des Namen "Großmann" (siehe Abb.10) ergeben. Für den Buchstaben "ß" gibt es drei Aussprachevarianten und für Doppelbuchstaben zwei. Die kombinatorischen Möglichkeiten lassen somit

Liste	Vokabular	Wörterbuch
Liste 1	925	1177
Liste 2	1927	2451

Tabelle 5: Vokabular- und Wörterbuchgröße für Buchstabierung

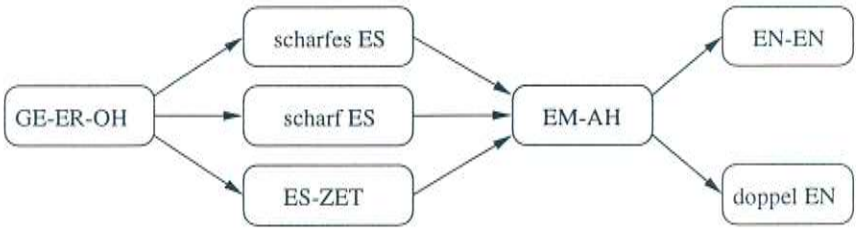


Abbildung 10: Varianten von "Großmann"

sechs Aussprachevarianten des Namens "Großmann" entstehen, die alle im Wörterbuch aufgenommen werden.

4 Verfahren bei kleinem Vokabular

In diesem Kapitel werden Verfahren vorgeschlagen, um nur wenige Namen zu erkennen. Der Begriff kleines Vokabular wird dabei nicht weiter konkretisiert, da dies von der Leistungsfähigkeit der Erkenner abhängt. Für die Experimente wurden alle Namen ins Vokabular aufgenommen, von denen Sprachdaten vorhanden waren (siehe Liste 2). Da die Zusammenhänge von gesprochenen und buchstabierten Namen untersucht wurden, wurde die Schnittmenge der Sprachdaten mit den zur Verfügungstehenden Transkriptionen gebildet, woraus Liste 1 mit 925 verschiedenen Namen entstand.

Bei kleinen Vokabularien kann die Aussprache jedes Namens erzeugt und für den Erkennungsprozeß benutzt werden. Sollte keine Transkriptionsdatenbank vorhanden sein, geht man davon aus, daß die Transkriptionen der gesprochenen Namen von Hand erstellt werden können.

Vokabulargröße	Anfrage Szenarien	
	Szenario 1	Szenario 2
klein	λ-NBest	Phonemrescoring und NBest-Ansatz mit Schätzen der Grenze

4.1 Getrennte Erkennung von gesprochenen und buchstabierten Namen

4.1.1 Ansatz mit Bestenliste

Geht man zunächst davon aus, daß die Wortgrenze zwischen dem gesprochenen und dem buchstabierten Namen bekannt ist, so lassen sich die beiden Namen getrennt voneinander erkennen.

Die Erkenner liefern den Namen mit der geringsten Fehlerwahrscheinlichkeit. Die Bewertungen, nach denen die einzelnen Namen sortiert wurden, sind jedoch nicht direkt als Wahrscheinlichkeitswerte interpretierbar. Sie wurden aus Effizienzgründen logarithmiert, mit Wortübergangstrafen versehen und mit einem gewichteten Sprachmodell aufaddiert.

Wird nicht nur die beste, sondern die N-besten Hypothesen berechnet, so kann man bestimmen, ob sich die richtige Hypothese unter den N-Besten befindet (siehe Tabelle 6). Die Namensgenauigkeit gibt an, wie oft sich die richtige Hypothese unter den ersten N Name befindet. Diese Erkennungsleistung wird in Zukunft als TOP-N Erkennungsleistung bezeichnet. Würde

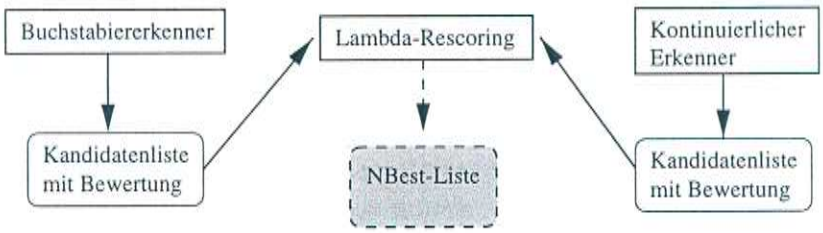


Abbildung 11: Struktur des λ-NBest Erkenners.

man eine TOP-N Namensgenauigkeit berechnen und N wäre die Anzahl aller möglichen Namen aus der Namensliste, dann wäre das Ergebnis 100% TOP-N Namensgenauigkeit. Da sich ab der 5. Hypothese die TOP-N Namensgenauigkeit nur sehr langsam verbessert, wurde die Bestenliste auf 10 Namen beschränkt.

Die Bewertung pro Namen für den jeweiligen Erkenner Y_L und Y_C lassen sich über eine einfache Formel gewichtet addieren und neu sortieren.

$$Y(Name) = \lambda \cdot Y_L(Name) + (1 - \lambda) \cdot Y_C(Name) \qquad 0 \leq \lambda \leq 1 \qquad (10)$$

Damit entsteht eine vereinigte neue Liste, deren beste Hypothese sowohl die gesprochene als auch die buchstabierte Information berücksichtigt. Es bleibt die Frage nach der Gewichtung der Information. Aufgrund der Tatsache, daß buchstabierte Namen besser erkannt werden können als gesprochene, wird die Gewichtung zugunsten des Buchstabiererkenners ausfallen. Experimente mit Variation des Gewichtungsfaktors λ bestätigten diese Vermutung (siehe Abbildung 12). In Abbildung 11 ist die Gesamtstruktur des λ-NBest Erkenners dargestellt.

Mit dieser Struktur [KTT95] konnte die Namensgenauigkeit gegenüber der Buchstabiererkennung um 0.5% auf 96.1% verbessert werden. Diese Verbesserung stellte sich bei Namen ein, bei denen sich die Bewertungen der Hypothesen des Buchstabiererkenners nur wenig differieren und die erste Hypothese falsch war, wogegen die richtige Hypothese in der gesprochenen Liste als eine der besten vorkam. Tauscht man die buchstabierten Bestenliste von JANUS mit der des MS-TDNN aus, dann ergibt sich ein λ mit ähnlicher Größenordnung.

Der optimale Wert von λ kann über eine Trainingsmenge bestimmt werden. Dazu wurden aus der Gesamtmenge 600 Äußerungen zufällig

	buchstabiert		gesprochen
	MS-TDNN	JANUS	JANUS
1	95.6%	90.9%	57.8%
2	97.3%	94.4%	67.3%
3	97.5%	95.1%	69.7%
4	97.6%	95.4%	71.1%
5	97.7%	95.7%	72.4%
6	97.7%	96.0%	73.1%
7	97.7%	96.4%	73.9%
8	97.8%	96.6%	74.4%
9	97.8%	96.7%	74.5%
10	97.8%	96.8%	75.1%

Tabelle 6: TOP-N Namensgenauigkeit der Bestlisten der jeweiligen Erkenner

ausgewählt, die zum Einstellen von λ dienten. Es wurde eine maximale Erkennungsleistung von 96.3% bei $0.9994 \leq \lambda \leq 0.9996$ ermittelt, mit einer Verbesserung von 0.6% gegenüber der Buchstabiererkennung. Für die Evaluation wurde mit einem $\lambda = 0.9995$ die verbleibenden 737 Äußerungen gewichtet und eine Namensgenauigkeit von 95.9% erzielt. Das war ebenfalls die maximal erreichbare Erkennungsleistung und eine Verbesserung um 0.5% gegenüber der Buchstabiererkennung. Diese Experimente zeigen, daß sich der Gewichtungsfaktor einstellen läßt. Bei einem Wechsel zu einem anderen Erkenner, oder bei einem anderen Verfahren zur Berechnung der Bewertung, ist der Gewichtungsfaktor anzupassen.

Durch eine Gewichtung für den buchstabieren Namen wird kein gesprochener Name, der nicht in der Bestenliste vom Buchstabiererkenner vorkommt, als bester den Neugewichtungsprozeß verlassen. Deshalb müßte JANUS nur die gesprochenen Namen der Kandidatenliste des MS-TDNN bewerten. Da es sich nur um wenige Namen handelt, würde dies sehr schnell gehen. Eine parallele Erkennung ist allerdings nicht mehr möglich. Da das MS-TDNN die Bewertungen schneller ermittelt als JANUS, bietet sich diese Vorgehensweise jedoch an. Dabei ist der λ-Wert neu zu bestimmen: $\lambda = 0.96$. Bei dieser Vorgehensweise ist eine Namensgenauigkeit auf Liste 1 mit 97.7% erreicht worden.

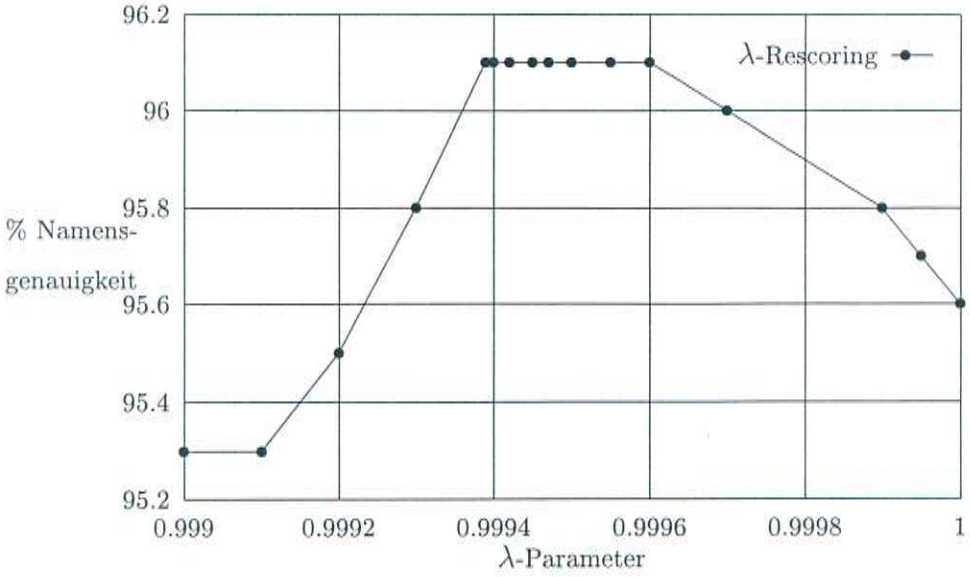


Abbildung 12: Namensgenauigkeit mit λ-NBest Erkenner und variablem λ.

Von Vorteil wäre eine geschlossene Theorie, die die Gewichtung automatisch bestimmt, oder ganz überflüssig macht.

Dies führt zu folgenden, allgemeineren Überlegungen: Der Erkennungsprozeß liefert im Fall des JANUS-Erkenners einen logarithmischen Wahrscheinlichkeitswert als Bewertung. Durch Exponentieren und Normieren der Bewertung pro Hypothese läßt sich dieser in eine Wahrscheinlichkeit für diese Hypothese zurücktransformieren.

Sei Y die Funktion, die die Wahrscheinlichkeit zu einer Hypothese ermittelt. Erkenner C liefert N Hypothesen $C_1 ... C_N$, dabei soll gelten:

$$1 \geq Y(C_1) \geq Y(C_2) \geq ... \geq Y(C_N) \geq 0 \qquad \text{mit} \qquad \sum_{i=1}^{N} Y(C_i) = 1 \qquad (11)$$

Ebenso für Erkenner L. Wahrscheinlichkeitswerte die dicht beieinander liegen, deuten eine unsichere Entscheidung an. Erwartet wird dieser Effekt besonders für die gesprochenen Namen.

Anstrengungen in dieser Richtung führten aus mehreren Gründen nicht zum Erfolg:

- Die Bewertungen, die vom Erkenner geliefert werden, stehen durch den Viterbi-Algorithmus, für die logarithmische Wahrscheinlichkeit des be-

sten Pfades und nicht für die Wahrscheinlichkeit der Hypothese. Trotzdem könnte man sich einen Erkenner vorstellen, der Wahrscheinlichkeiten liefert.

- Es werden nur N-Beste Hypothesen geliefert. Würde man die Wahrscheinlichkeiten dieser Besten so normieren, daß ihre Summe eins ergibt, dann würde man explizit davon ausgehen, daß die restlichen Hypothesen eine Nullwahrscheinlichkeit aufweisen. Diese Annahme ist nicht korrekt, wären aber näherungsweise akzeptabel. Wichtig ist in diesem Zusammenhang auch, daß für beide Erkenner die gleiche Anzahl an Hypothesen berechnet werden, da durch die Normierung sonst Inkonsistenzen entstehen[11]

- Versucht man aus den Bewertungen Wahrscheinlichkeiten zu berechnen, so stößt man aufgrund des großen Wertebereiches auf Probleme der mathematischen Ungenauigkeit der Zahlendarstellung auf dem Rechner. Diese sind mit verstärktem Programmieraufwand zu überwinden, die angeführten Punkte rechtfertigen diesen Aufwand jedoch nicht.

4.2 Gesprochen-Buchstabiert zusammen erkennen

4.2.1 Modellierung der Namen im Wörterbuch

Im folgenden wird ein Lösungsansatz für Szenario 2 bei kleinem Vokabular diskutiert. In Szenario 2 wird der Eigenname nacheinander gesprochen und buchstabiert. Die Stelle, an der der gesprochene Name aufhört und der buchstabierte anfängt, wird im weiteren als *Grenze* bezeichnet. Diese Grenze ist, im Gegensatz zu den Experimenten im vorigen Unterkapitel, nicht gegeben. Damit fehlt eine Information, die benötigt wird, um die Buchstabiersequenz unabhängig vom gesprochenen Namen zu bearbeiten.

In einem ersten Verfahren wird untersucht, ob auf die explizite Bestimmung der Grenze verzichtet werden kann, wenn der gesprochene und buchstabierte Name zusammen ins Wörterbuch aufgenommen wird.

```
{Erb_E_R_B} {{E WB} R P SIL EH SIL ER SIL B {EH WB}}
```

Da längere Äußerungen i.d.R. auch besser erkannt werden, erwartet man durch diese Modellierung ein gutes Erkennungsergebnis.

[11] Die Wahrscheinlichkeiten von vielen Hypothesen liegen dichter zusammen, als die von wenigen.

JANUS	
Kommentar	**Liste 1**
Suche	86.0%
s.o. mit Wahrscheinlichkeit	85.9%
viterbi	86.1%

Tabelle 7: Gesprochene und buchstabierte Namen ohne Berücksichtigung der Grenze. Angaben in Prozent Namen korrekt.

Als überraschendes Ergebnis verschlechterte sich jedoch die Erkennungsrate, im Vergleich zur Buchstabiererkennung um etwa 10% (vergl. Tabelle 4 mit 7). Die zusätzliche Information des gesprochenen Namens am Anfang der Äußerung bringt also keine Verbesserung. Sie hat vielmehr einen störenden Einfluß auf den Erkennungsprozeß. Aus diesem Grund wollen wir den Einfluß des gesprochenen Namens auf die Erkennung reduzieren.

4.2.2 Phonem-Neubewertung

Wir haben festgestellt, daß der buchstabierte Name mehr zur korrekten Erkennung beiträgt als der gesprochene. Aus diesem Grund wollen wir seinen Einfluß verstärken. Dazu wird eine Phonem-Neubewertung eingeführt, bei der die Bewertung der Phoneme, die zu dem gesprochenen Namen gehören, abgeschwächt, während die Buchstabenphoneme verstärkt werden. Durch den Viterbi-Algorithmus läßt sich die versteckte optimale Zustandssequenz wiederherstellen und damit ein mögliches Grenzintervall[12] bestimmen. Dadurch ist auch eine Bestimmung der Frames und der Bewertung pro Phonem möglich. In Abbildung 13 sind zwei Namen (Wort1 und Wort2) mit ihrer Phonemsequenz dargestellt. Der Index i bezeichnet im folgenden einen beliebigen Namen. Das erste Frame des Namens i, dessen Phonem zu einem Buchstaben gehört, wird mit m_i bezeichnet.

Mit dieser Zuordnung läßt sich die Bewertung gewichten, je nachdem welchem Bereich sie zugeordnet wurde.

F bezeichnet die Anzahl der Frames und ist für eine Äußerung konstant. Wir wollen nun, pro Namen i, die Gesamtbewertung S_i, die Bewertung über den Buchstaben b_i und die Bewertung im Bereich des gesprochenen Namens

[12]Bereich, in dem der gesprochene Namen aufhört und der buchstabierte anfängt.

k_i nennen. Da die Gesamtbewertung über eine feste Framelänge berechnet wird, aber die Framelänge des gesprochenen bzw. buchstabierten Namens für jeden Namen unterschiedlich sein kann, lassen sich k_i und k_j nicht vergleichen, wenn $m_i \neq m_j$ gilt. Denn würde man sie miteinander vergleichen, wäre der mit der kürzeren Framelänge bevorzugt. Um dieses Problem zu beseitigen, werden wir eine Längennormierung durchführen.

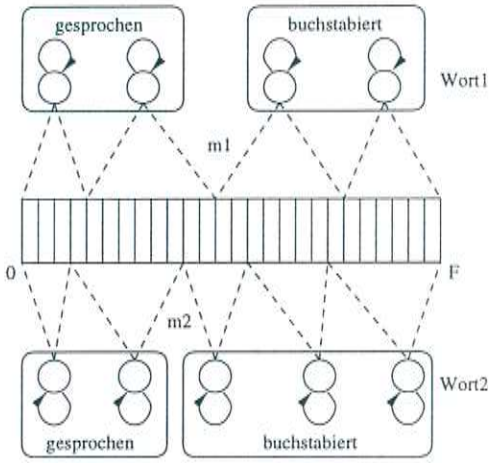


Abbildung 13: Zwei HMM bezüglich der Frames einer Äußerung

Dadurch wird die durchschnittliche Bewertung pro Frame ermittelt. Eine hohe Bewertung, die über einem großen Bereich berechnet wurde, erhält die gleiche Chance wie eine kleine Bewertung über einen kurzen Bereich.

$$\widetilde{S_i} = \frac{S_i}{F} \tag{12}$$

$$\widetilde{k_i} = \frac{k_i}{m_i} \tag{13}$$

$$\widetilde{b_i} = \frac{b_i}{F - m_i} \tag{14}$$

Zu beachten ist, daß bezüglich der Sortierung nach der normierten Gesamtbewertung $\widetilde{S_i}$ im Vergleich zur nicht normierten keine Veränderung aufgetreten ist, da über allen i mit F normiert wurde. Dies trifft nicht auf $\widetilde{k}$ und $\widetilde{b}$ zu.

Wir wollen, wie bereits erwähnt, eine Gewichtung einführen. Betrachten wir zunächst den naheliegendsten Ansatz:

$$Y_i = \lambda \cdot \tilde{k_i} + (1 - \lambda) \cdot \tilde{b_i} \tag{15}$$

Diese Formel hat aber den Nachteil, daß die Gleichgewichtung des gesprochenen und buchstabierten Namens ($\lambda = \frac{1}{2}$) nicht der Sortierung nach dem Gesamtbewertung entspricht, da die Summe der beiden normierten Teilbewertungen nicht gleich der normierten Gesamtbewertung ist:

$$\begin{aligned} \widetilde{S_i} &= \frac{S_i}{F} \\ &= \frac{k_i + b_i}{m_i + (F - m_i)} \\ &\neq \frac{k_i}{m_i} + \frac{b_i}{F - m_i} \quad \text{, falls} \quad m_i \neq (F - m_i) \end{aligned}$$

Wenn die Trennung zwischen gesprochenem und buchstabiertem Namen zu einer Äußerung nicht genau in der Mitte ($m_i = \frac{F}{2}$) erfolgt, dann läßt sich nicht bestimmen für welchen Lambdawert die Sortierung der Sortierung nach der Gesamtbewertung entspricht.

Deshalb bestimmen wir einen "Korrekturfaktor" X_i wie folgt:

$$\begin{aligned} \widetilde{S_i} &= X_i(\tilde{k_i} + \tilde{b_i}) \quad \Longleftrightarrow \\ X_i &= \frac{\widetilde{S_i}}{\tilde{k_i} + \tilde{b_i}} \quad \text{, wobei} \quad \tilde{k_i} + \tilde{b_i} > 0 \end{aligned} \tag{16}$$

Damit läßt sich Formel 15 so angeben, daß bei einer Gleichgewichtung der beiden Bewertungen mit $\lambda = \frac{1}{2}$ gerade die Sortierung wie über der Gesamtbewertung entsteht. Wir könnten X_i noch mit zwei multiplizieren um die normierte Gesamtbewertung zu erhalten. Für die Sortierung ist ein konstanter Faktor jedoch bedeutungslos:

$$Y_i = X_i \cdot (\lambda \cdot \tilde{k_i} + (1 - \lambda) \cdot \tilde{b_i}) \tag{17}$$

Eine andere Möglichkeit besteht darin, über allen Namen eine mittlere Grenze zu berechnen in der Hoffnung, daß diese von den tatsächlichen Grenzen einen möglichst geringen Abstand hat. Einige der Bewertungen

würden somit im falschen Bereich gewichtet werden, aber man hätte den Vorteil, Formel (15) benutzen zu können.

Ergebnisse

Von 1100 Äußerungen wurden mit dieser Modellierungstechnik jeweils 20 Kandidaten bestimmt. Die beste Hypothese erzielte auf dieser Liste eine Namensgenauigkeit von 88.5% (93% TOP-20 Namensgenauigkeit). Eine Neugewichtung hätte also maximal 93% Namensgenauigkeit erzielen können. Wie aus der Abbildung 14 ersichtlich, stellt sich eine maximale Erkennungsleistung bei $\lambda = \frac{1}{2}$ ein. Bei diesem Wert für λ entspricht das Ergebnis einer Sortierung nach der Gesamtbewertung. Die Sortierung nach der Buchstabenbewertung befindet sich im Bereich $0 \leq \lambda \leq \frac{1}{2}$. Es konnte also durch eine Neuberwertung keine Verbesserung erzielt werden.

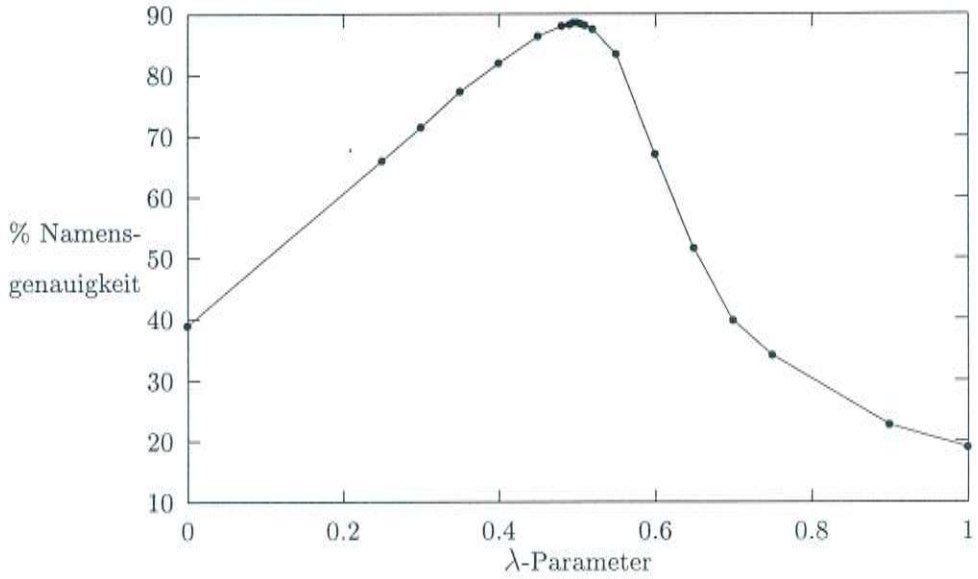


Abbildung 14: Neugewichtung der Phonembewertung über eine Besten-Liste.

Erfolgversprechend scheint dagegen die Benutzung der gewonnenen Grenze, um den Buchstabier- und kontinuierlichen Erkenner unabhängig voneinander zu benutzen. Damit erreicht man das Verhalten wie bei Szenario 1.

4.2.3 Schätzen der Grenze durch Viterbi

Im letzten Abschnitt wurde vorgestellt, wie in einer Äußerung die Bereiche des gesprochen und buchstabierten Namens gefunden werden können (siehe Abbildung 13). Das dort mit m_i bezeichnete Frame kennzeichnet dabei die Stelle, an der der gesprochene Name aufhört und der buchstabierte beginnt. Es handelt sich dabei genau um die Information, welche durch eine halbautomatische und von Hand nachbearbeitete Grenze für alle gesammelten Sprachdaten zur Verfügung gestellt wurde. In Abbildung 15 wurden die durch den Viterbi gefundenen Grenzen mit den von Hand bestimmten verglichen. Es liegen über 75% der gefundenen Grenzen nicht weiter als 10ms von den "tatsächlichen" Grenzen entfernt.

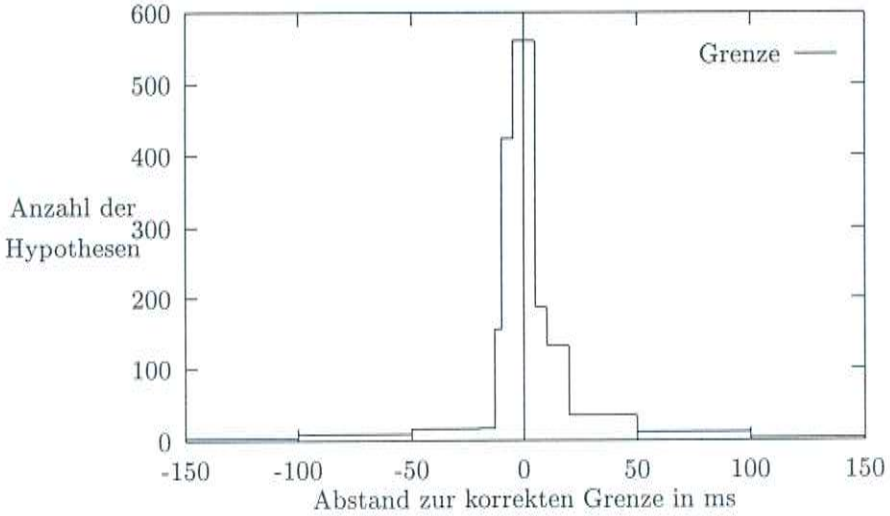


Abbildung 15: Abweichung der geschätzten zur handbestimmten Grenze in Millisekunden. Die Nullinie bedeutet, daß die handsegmentierte Grenze mit der gefundenen übereinstimmt.

Mit dieser geschätzten Grenze kann der bereits beschriebene λ-NBest Ansatz benutzt werden.

4.2.4 NBest-Ansatz mit geschätzter Grenze

Die Erkennungsleistung der Lösungsansätze zu Szenario 2, bei denen die *Grenze* zwischen dem gesprochenen und buchstabierten Namen nicht expli-

zit ermittelt wurde, waren schlechter als eine reine Buchstabiererkennung. Um eine Buchstabiererkennung unabhängig vom gesprochenen Namen durchzuführen, muß zunächst der Anfang des buchstabierten Namens bestimmt werden. Wenn diese Grenze gut geschätzt werden kann, dann müßte, wie bei der Buchstabiererkennung in Tabelle 4, die Namensgenauigkeit bei etwa 95% liegen.

Wie im vorigen Ansatz erläutert wurde, kann die Zustandssequenz durch den Viterbi-Algorithmus ermittelt werden. Daraus läßt sich der Übergang vom gesprochenen zum buchstabierten Namen bestimmen. Da wir aber a priori nicht wissen, welcher Name gesprochen wurde, wird als erstes eine Hypothese, wie in Abschnitt 4.2.1 beschrieben, berechnet. Mit Hilfe dieser Hypothese wird dann der Anfang des buchstabierten Namens berechnet. In Abbildung 15 sieht man, daß diese Grenze in vielen Fällen nur geringfügig von der handbestimmten Grenze abweicht.

Mit der Bestimmung der Grenze haben wir das Problem auf Szenario 1 reduziert. Man kann den dort beschriebenen Ansatz benutzen und den buchstabierten und gesprochenen Namen getrennt erkennen. Vergleicht man das Verhalten des λ-Parameters, wenn die Grenze bekannt bzw. geschätzt wurde (Abbildung 16), sieht man, daß sich das Maximum für beide Kurven bei $\lambda = 0.9995$ einstellt.

Durch die Vereinigung und Neusortierung der Bestenlisten konnte ca. 0.5% Erkennungsleistung gewonnen werden. Die Erkennungsleistung mit geschätzter Grenze im Vergleich zur bekannten ist um ca. 1% gesunken. Dies ist demnach eindeutig auf die Bestimmung der Bereiche des gesprochenen und buchstabierten Namens zurückzuführen.

Eine weitere Verbesserung der Erkennungsleistung wurde durch folgende Überlegung erzielt: Die Bewertung des buchstabierten Namen wird durch die Neubewertung mit λ bevorzugt. Die Namen, die in der Bestenliste des Buchstabiererkenners vorkommen, erhalten eine so gute Bewertung, daß nur sie zur besten Hypothese werden können. Die gesprochenen Namen können somit nur noch die Sortierung beeinflussen. Aus diesem Grund brauchen nur die gesprochenen Namen bewertet werden, die der Buchstabiererkenner als Hypothesen ausgibt. Dies hat den Vorteil, daß Namen, die zwar in der einen aber nicht in der anderen Bestenliste vorkommen, nicht mit einer festen, niedrigen Bewertung versehen werden müssen. Die Struktur dieses Erkenners

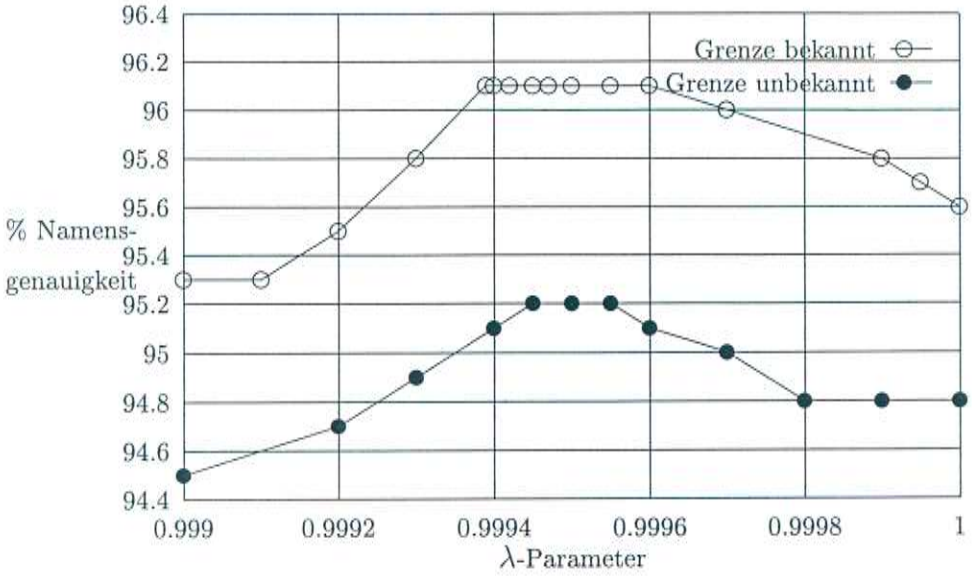


Abbildung 16: Vergleich der Erkennungsleistung mit bekannter und geschätzter Grenze.

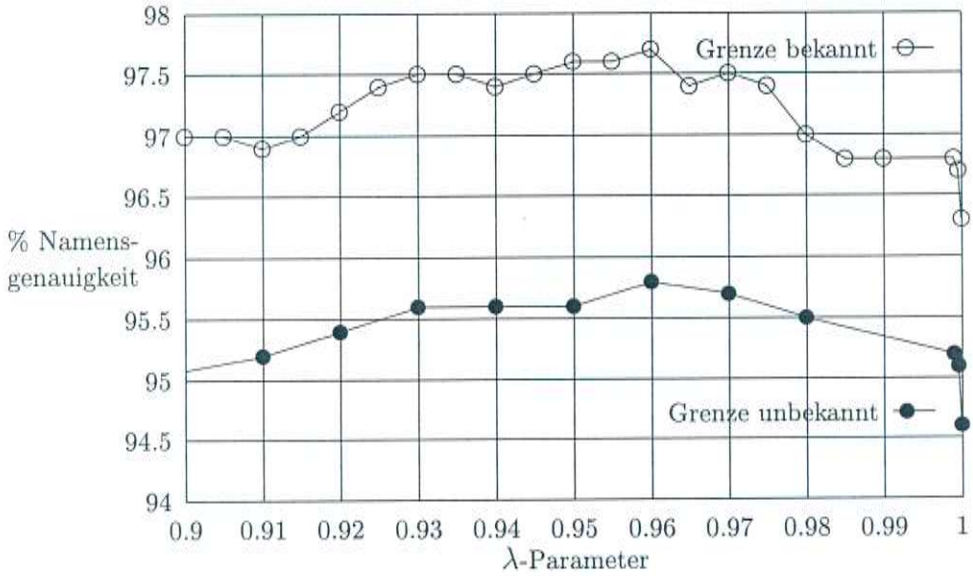


Abbildung 17: Wie in Abbildung 16 mit Kandidaten vom MS-TDNN-Buchstabiererkenner.

wird im folgenden Kapitel aus einem weiteren Gesichtspunkt motiviert und in Abbildung 18 dargestellt. In Abbildung 17 wird mit dieser Verbesserung die Erkennungsleistung bei bekannter und geschätzter Grenze gegenübergestellt. Die Namensgenauigkeit des Buchstabiererkenners bei bekannter Grenze liegt in diesem Fall bei 96.5%. Es kann also durch die Neusortierung, sowohl bei bekannter als auch bei geschätzter Grenze, eine Verbesserung um 1% erzielt werden.

5 Verfahren bei mittlerer Vokabulargröße

Unter einer mittleren Vokabulargröße verstehen wir ein Vokabular, das so umfangreich ist, daß die Kapazität des Erkenners nicht ausreicht, alle Namen aufzunehmen und für den Erkennungsprozeß zu nutzen. Jedoch kann noch für jeden Namen seine Aussprache ermittelt werden. Sie kann jedoch nicht wie bisher direkt benutzt werden. Aus diesem Grund wird die Erkennung in zwei Phasen aufgeteilt:

1. Phase: Reduktion des Vokabulars auf eine möglichst kleine Liste (Kandidatenliste), in der die richtige Hypothese mit großer Sicherheit enthalten ist.

2. Phase: Mit der Kandidatenliste können die Verfahren bei kleinem Vokabular benutzt werden.

Information über die Eigennamen	Anfrage Szenarien	
	Szenario 1	Szenario 2
mittel	erweiterter NBest-Erkenner	Zwei-Phasen Erkenner

5.1 Reduktion des Vokabulars

Zur Reduktion des Vokabulars modelliert man statt Namen nur Wortuntereinheiten (Kapitel 2.3). Bei einem buchstabierten Namen bieten sich Buchstaben als natürliche Wortuntereinheiten an. Aus den Buchstabensequenzen können über eine (Baum-) Grammatik mögliche Namen erzeugt werden. Diese hypothetische Namensliste bezeichnen wir im weiteren als *Kandidatenliste*.

Zur Modellierung des gesprochen Namens werden als Wortuntereinheiten etwa 60 Phoneme benutzt. Das Vokabular besteht also nur aus 60 Einträgen. Durch den Erkennungsprozeß erhalten wir eine Menge von Transkriptionshypothesen. Mit Hilfe der Transkriptionsdatenbank können Namen gesucht werden, deren Aussprache sehr ähnlich der Transkriptionshypothese sind. Dabei könnten leicht verwechselbare, ähnlich klingende Phoneme über eine Verwechselbarkeitsmatrix mitberücksichtigt werden. Die daraus resultierende Namensliste bezeichnen wir wieder als *Kandidatenliste*. Die Anzahl der Kandidaten kann über die Anzahl der Transkriptionshypothesen und den Übereinstimmungsgrad der Hypothesen zur Datenbanktranskription festgelegt werden. Damit lassen sich sowohl für buchstabierte sowie für gesprochene

Namen eine Kandidatenliste bestimmen. Mit dieser können die für Szenario 1 beschrieben Verfahren benutzt werden.

Wurde der Name gesprochen und buchstabiert, so werden Phoneme und Buchstaben als Vokabular benutzt. Das Vokabular enthält somit etwa 90 sehr kleine Worte. Kandidaten können damit sowohl über die Phoneme, als auch über die Buchstaben ermittelt werden.

5.2 Gesprochen-Buchstabiert getrennt erkennen

5.2.1 Erweiterter NBest-Erkenner

In diesem Unterkapitel wird das Vorgehen bei Szenario 1 beschrieben. Die Erkennung wie sie bisher durchgeführt wurde, ist für die gesprochenen Namen durch die Vergrößerung des Suchraumes nicht mehr möglich. Für die Buchstabiererkennung werden als Vokabular die Buchstaben benutzt. Über eine entsprechende Grammatik werden Eigennamen ausgegeben. Dies ist auch bei einer Million Eigennamen noch effizient möglich.

Um die Information der gesprochenen Namen zu integrieren, wird das Modell in Abbildung 18 vorgeschlagen:

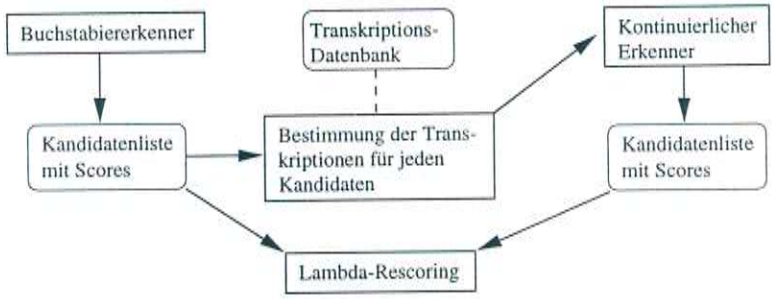


Abbildung 18: Erweiterter λ-NBest Erkenner bei großen Vokabularien

Für die durch den Buchstabiererkenner erzeugten Namen werden die Transkriptionen ermittelt. Die Kandidaten sind, wie folgende Listen zeigen, sehr ähnlich:

Bsp. 1: *Vannerum, Lanner, Panner, Danner, Tanner, Zanner, Gannert, Kannert*

Bsp. 2: *Reeckers, Becker, Baecker, Beckeer, Wecker, Biecker, Beckner, Pecker, Bencker*

Anzahl der Namen	Namensgenauigkeit
(Liste 1) 925	97.7%
10000	94.8%
100000	89.5%

Tabelle 8: Erkennungsleistung des erweiterten λ-NBest Erkenners bei unterschiedlicher Vokabulargröße.

Der kontinuierliche Erkenner kann auf wenigen Namen Bewertungen erzeugen, die durch das λ-NBest Neubewertungsverfahren mit den buchstabierten vereinigt werden. Die experimentellen Ergebnisse sind in Tabelle 8 zusammengestellt.

5.3 Gesprochen und Buchstabiert zusammen erkennen

Für Szenario 2 steht der buchstabierte Name nicht direkt zur Verfügung. Im vorigen Kapitel haben wir den Bereich des gesprochenen und buchstabierten Namen durch den Viterbi-Algorithmus bestimmt. Dies ist zunächst wegen der Größe des Wörterbuchs nicht möglich.

5.3.1 Phonem- und Buchstabenerkenner

Es kann nicht für jeden Namen ein Markov-Modell erstellt werden. Die Anzahl der Namen ist hierfür zu groß. Es werden deshalb Wortuntereinheiten als Vokabular gewählt, aus denen Namen extrahiert werden können. Für die Buchstabiersequenz sind Buchstaben geradezu ideale Worte. Für die gesprochenen Namen müssen kleinere Wortuntereinheiten, nämlich Phoneme gewählt werden.

Da der Namen zuerst gesprochen und dann buchstabiert wurde, sollen auch zunächst nur Phoneme und dann Buchstaben erkannt werden. Dies wurde mit einem Trigramm-Sprachmodell erreicht. Es kann weiterhin dazu benutzt werden, die Häufigkeit der Phonem- und Buchstabenübergänge zu berücksichtigen. Die Häufigkeitsanalyse wurde über einer möglichst großen Namensmenge durchgeführt. Dazu diente die Namensliste aus dem Lexikon Deutscher Eigennamen (siehe Anhang). Die 288464 Namen aus der Transkriptionsdatenbank, wurden als Phonem-Buchstabensequenz dargestellt. Es werden folgende Faktoren berücksichtigt:

```
H AU S $R $H $A $U $S $E $R  (hypo 0 | score = 13513.71)
H AU S $R $H $A $U $F $E $R  (hypo 1 | score = 13516.33)
H AU S $R $K $A $U $S $E $R  (hypo 2 | score = 13521.20)
H AU S $R $K $A $U $F $E $R  (hypo 3 | score = 13523.83)
H AU Z E2 $H $A $U $S $E $R  (hypo 4 | score = 13531.89)
```

Abbildung 19: Ausgabe von Phase 1

```
REFERENCE: a t r A  sch $a $t $t *** $r $a $s $c $h **
HYPOTHESE: a t r AH sch $a $t $t $~A $r $a $s $c $h $A
```

Tabelle 9: Vergleich der Hypothese mit der Referenz.

- Von 30% der Namen, die Doppelbuchstaben beinhalten, wird die Aussprachevariante "doppel" verwendet.
- Der Buchstabe "ß" wurde mit drei Aussprachevarianten versehen, die in der Grammatik gleichwahrscheinlich modelliert wurden.
- Der "Bindestrich" wurde nicht berücksichtigt.

Über diese Menge werden Trigramme berechnet. Damit wird erzwungen, daß nach der Erkennung des ersten Buchstaben nur noch Buchstaben folgen können.

In Abbildung 19 sind die fünf besten Hypothesen einer Äußerung mit ihrer Bewertung exemplarisch dargestellt. Die Phoneme "E2" und "R" werden bei den ersten 4 Hypothesen fälschlicherweise als erster Buchstabe "$R" erkannt. Eine Liste der besten Hypothesen, mit maximal N Einträgen, wird ausgegeben.

Es wurde eine Phonem-Buchstabengenauigkeit von 64.3% erzielt. In Tabelle 9 wurde der Name "Attrasch" als Hypothese und als Referenz gegenübergestellt. Beim gesprochenen Namen wurde ein gedehntes A mit einem kurzen verwechselt und als Fehler gezählt.

5.3.2 Bestimmung von Kandidaten

Im nächsten Schritt werden aus der Phonem-Buchstabensequenz die Namen ermittelt, die am besten auf diese Sequenz passen. Diese Namen nennen wir Kandidaten. Dazu wird mittels dynamischer Programmierung nach Formel 9

der Abstand zwischen der Hypothese und allen Referenzen berechnet (Tabelle 9). Der Abstand wird iterativ so verändert, daß die gewünschte maximale Anzahl der Kandidaten erzeugt wird. Damit nicht unnötigerweise viele unähnliche Namen benutzt werden, kann auch der maximale Abstand, der noch berücksichtigt werden soll, gewählt werden. Pro Äußerung haben sich etwa 90 Kandidaten als maximale Kandidatenzahl als ausreichend erwiesen, um eine TOP-90 Namensgenauigkeit von über 90% zu erzielen. Da die Phonemgenauigkeit schlechter ist als die Genauigkeit der Buchstaben, ergibt dasselbe Verfahren nur über die Buchstaben gleich gute Ergebnisse bei durchschnittlich 60 Kandidaten. Dazu ist die Berechnung deutlich schneller, da weniger Zeichen verglichen werden müssen (Kurve "Buchstaben" in Abbildung 20).

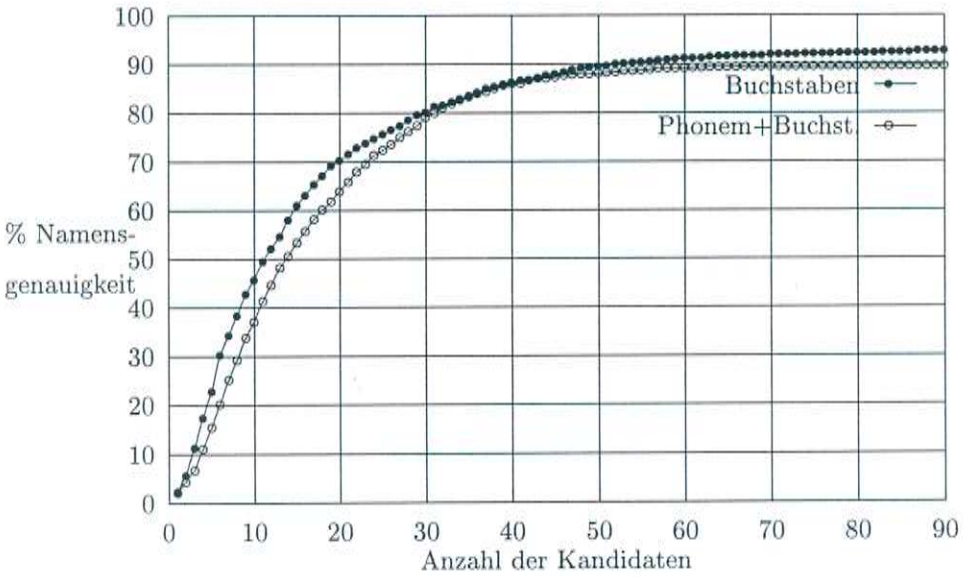


Abbildung 20: Vergleich der TOP-N Namensgenauigkeit mit und ohne Phonemsequenz.

5.3.3 Zwei-Phasen Erkennung

Als Ausgangspunkt haben wir ein großes Vokabular und die Transkriptionen der Eigennamen. Die Erkennung wird deshalb in mehrere Phasen (siehe Abbildung 21) unterteilt. Dabei wird Phase für Phase eine Kandidatenliste

erzeugt. Über diese relativ schlanke Liste ist im letzten Schritt eine Suche nach der besten Hypothese möglich.

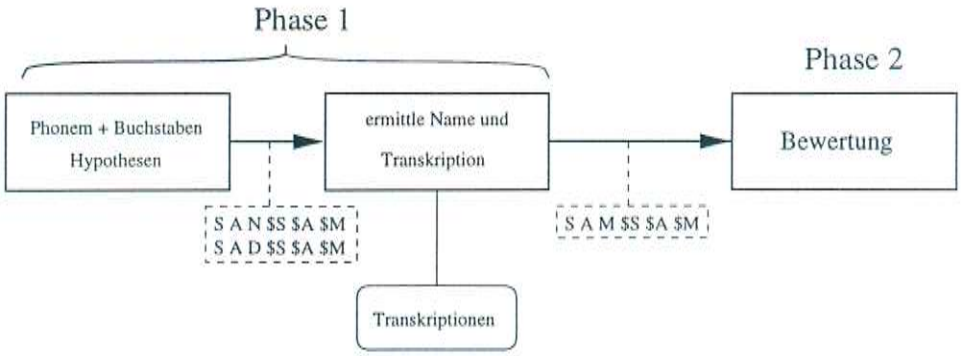


Abbildung 21: Zweiphasige Erkennung: Reduktion des Vokabulars und anschließende Bewertung.

In Szenario 2 ist die Grenze zwischen buchstabierten und gesprochenen Namen nicht bekannt. Die Erzeugung der Kandidatenliste wurde bereits vorgestellt.

In Phase 2 wird für jeden Kandidaten wie in Abschnitt 4.2.1 ein HMM erzeugt und dieses bewertet. Die Sortierung nach der Bewertung liefert die beste Hypothese. Der Vergleich der TOP-N Namensgenauigkeit nach Phase 1 und nach Phase 2 ist in Abbildung 22 dargestellt.

Zusammenfassung

Ausgehend von der Annahme, daß die Trennung zwischen buchstabierten und gesprochenen Namen nicht bekannt ist, werden Phonem- und Buchstabensequenzen erkannt. Die Bestimmung einer Liste der Namen, unter denen sich der tatsächlich gesprochene Namen befindet, wird mittels Formel (9) bestimmt. Es hat sich erwiesen, daß dazu die Buchstabensequenz ausreicht. Jeder Namen (Kandidat) dieser Liste wird mit einem HMM modelliert und bewertet. Werden die Namen nach der Bewertung sortiert, erhält man die beste Hypothese mit einer Namensgenauigkeit von 89% bei Liste 1.

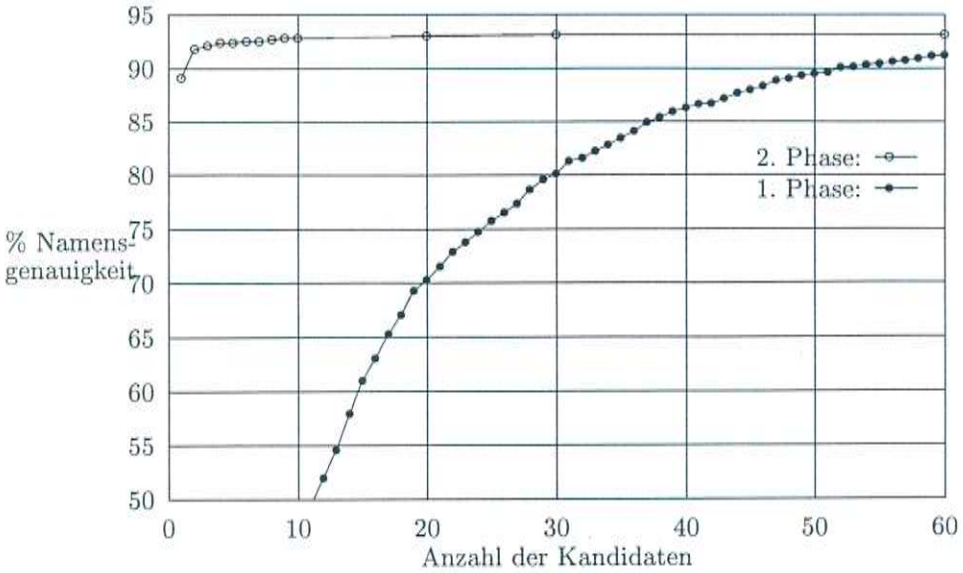


Abbildung 22: Vergleich der TOP-N Namensgenauigkeit nach Phase 1 und Phase 2 für Liste 1.

5.3.4 Erweiterter NBest-Erkenner mit geschätzter Grenze

In den bisherigen Ansätzen haben wir festgestellt, daß es sinnvoll ist, eine Erkennung des buchstabierten Namens, wenn möglich unabhängig von dem gesprochenen durchzuführen. Dazu ist aber die Kenntnis der Bereiche notwendig, in der die Namen gesprochen bzw. buchstabiert wurden. Der Phonem-Buchstabiererkenner kann diese Bereiche auch liefern. Mit Hilfe des Viterbi-Algorithmus wird die Zustandssequenz und daraus die Frames bestimmt (siehe Abbildung 13).

Nach der Bestimmung der Grenze kann wie in Szenario 1 vorgegangen werden. Für 100.000 Namen wurde mit dieser Methode eine Namensgenauigkeit von 71.8% erzielt.

Die Bestimmung der Grenze mit dem Phonem-Buchstabiererkenner hat den Vorteil, daß sie unabhängig von einer Namensliste und deren Transkriptionen durchgeführt werden kann. Sie ist aber nicht sehr genau und für die relativ niedrige Namensgenauigkeit von 71.8% verantwortlich. Mit demselben Erkenner, aber korrekter Grenze wurde eine Namensgenauigkeit von 89.5%

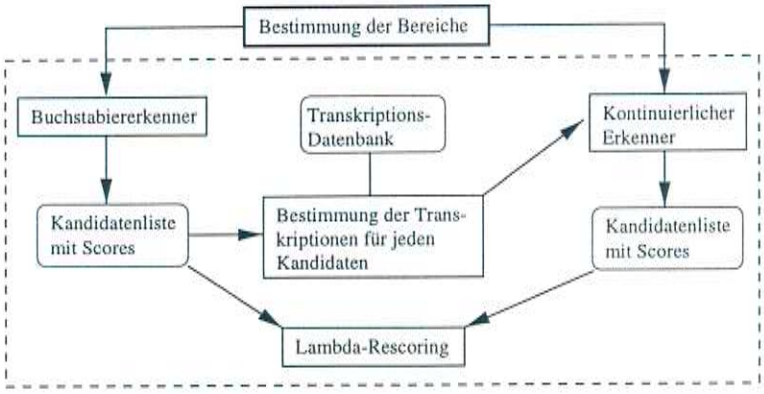


Abbildung 23: Struktur des erweiterten λ-NBest Erkenners mit geschätzter Grenze.

erreicht.

Die Bestimmung der Bereiche kann nur mit größerem Aufwand genauer ermittelt werden:

1. Aus der Buchstabenhypothese des Phonem-Buchstabenerkenners wird der Name ermittelt, dessen Abstand nach Formel 9 am geringsten ist.

2. Von diesem Kandidaten wird die Transkription ermittelt und eine Modellierung des gesprochenen und buchstabierten Namens erstellt.

3. Mit dem Viterbi-Algorithmus und mit dem Modell aus dem letzten Schritt kann dann eine genauere Grenze ermittelt werden.

Mit diesem Verfahren wurde die Namensgenauigkeit bei 100.000 Namen auf **88.1%** verbessert. In Abbildung 24 ist ersichtlich, wie sich die Erkennungsgenauigkeit nach jedem Verarbeitungsschritt verbessert. Theoretisch könnte nun iterativ die Grenze angepaßt werden, indem die Ausgabe des NBest-Erkenners als Eingabe für die Bestimmung der Grenze genommen wird. Wenn die neu ermittelte Grenze stark von der bisherigen abweicht, könnte sich ein zweiter Durchgang lohnen. Ob der damit wachsende Rechenaufwand im Verhältnis zur Verbesserung der Erkennungsleistung stehen, wurde im Rahmen dieser Diplomarbeit nicht erarbeitet.

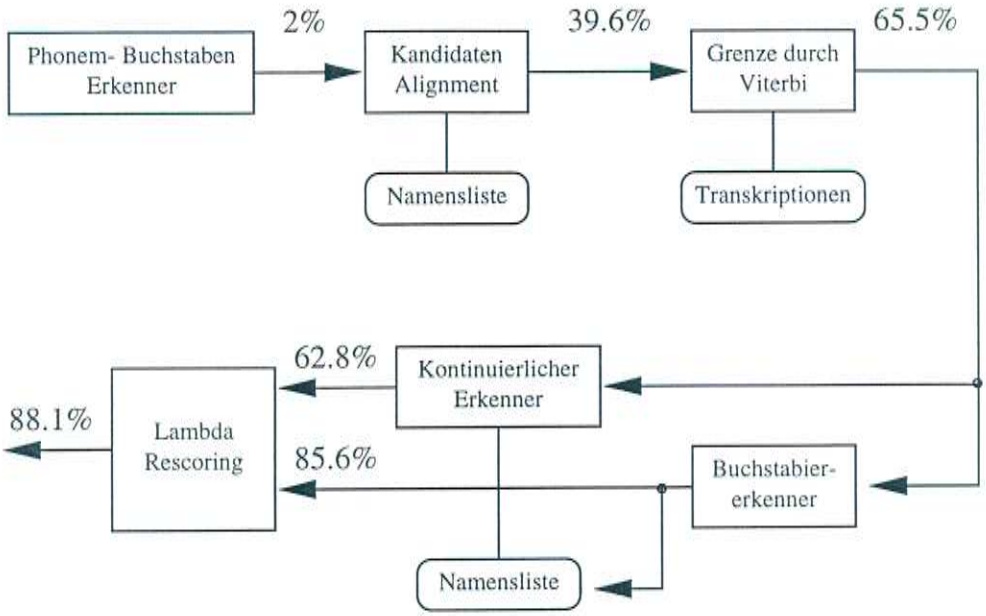


Abbildung 24: Datenfluß des erweiterten λ-NBest Erkenners mit geschätzter Grenze. Die Zahlen geben die Namensgenauigkeit nach jedem Modul an, bei einer Namensliste von 100.000 Namen.

Anzahl der Eigennamen	Kandidaten Alignment	Viterbi	Buchstabier-Erkenner	kontinuier-licher Erkenner	NBest-Erkenner
(Liste 1) 925	69.5%	91.5%	95.7%	89.8%	96.9%
14000	49.8%	80.2%	91.2%	74.0%	92.7%
100000	39.6%	65.5%	85.6%	62.8%	88.1%

Tabelle 10: Namensgenauigkeit des Erweiterten λ-NBest Erkenners nach jedem Schritt.

6 Verfahren bei großem Vokabular

Die Namensliste ist so groß, daß zu den meisten Namen keine Transkriptionen vorliegen. Demnach ist eine Erkennung des gesprochenen Namens nicht möglich und die Verarbeitung der sprachlichen Äußerung beschränkt sich auf die Buchstabierung.

Vokabulargröße	Anfrage Szenarien	
	Szenario 1	Szenario 2
groß	Buchstabier-erkennung	Buchstabiererkennung mit Bereichbestimmung

Man könnte jedoch den gesprochenen Namen in den Erkennungsprozeß integrieren, auch wenn seine Transkription a priori nicht vorhanden ist. Dazu sollen folgende Punkte Anregung geben:

- Anhand des erkannten buchstabierten Namens kann durch ein automatisches Transkriptionsverfahren auch die gesprochene Variante bewertet werden. Es könnten dann alle bisherigen Verfahren verwendet werden.
- Aus der Buchstabensequenz könnte man auf bestimmte Merkmale in dem gesprochenen Namen schließen, wie zum Beispiel Anzahl der Vokale, Zischlaute oder ähnlichem und damit eine ähnliche Bewertung wie mit einem kontinuierlichen Erkenner durchführen.

Diese Möglichkeiten wurden in dieser Diplomarbeit nicht untersucht.

6.1 Gesprochen und Buchstabiert getrennt erkennen

6.1.1 Buchstabiererkennung

In Szenario 1, das wir zunächst betrachten wollen, kann der buchstabierte Namen unabhängig vom gesprochenen erkannt werden. Die Namensliste ist so groß, daß die meisten Transkriptionen für die gesprochenen Namen nicht zur Verfügung stehen. Somit kann nur der buchstabierte zur Erkennung herangezogen werden.

Da automatische Transkriptionsverfahren existieren, ist die Nutzung eines solchen Verfahrens nach der Buchstabiererkennung möglich. Damit könnte ein erweiterter NBest-Erkenner benutzt werden. Da die bisher benutzten Transkriptionen ebenfalls maschinell bestimmt wurden, ist damit eine kleine Erkennungsverbesserung zu erwarten.

Anzahl der Namen	Namensgenauigkeit
10000	93.4%
100000	86.4%

Tabelle 11: Namensgenauigkeit des Buchstabiererkenners bei großem Vokabular

6.2 Gesprochen und Buchstabiert zusammen erkennen

6.2.1 Buchstabiererkennung mit geschätzter Grenze

Als ersten Erkennungsschritt liefert der Phonem-Buchstabiererkenner mehrere Hypothesen für mögliche Phonem-Buchstabensequenzen und mögliche Grenzen zwischen gesprochenem und buchstabiertem Namen. Da die Grenze ein wesentlicher Bestandteil der weiteren Erkennung ist, sollte sie möglichst genau geschätzt werden.

Durch das Auffinden der Grenze kennt man den Bereich des buchstabierten Namens. Für die Erzeugung von Hypothesen kann man nur den Buchstabiererkenner verwenden, da keine Transkriptionen der gesprochenen Namen vorhanden sind.

Anzahl der Namen	Namensgenauigkeit
(Liste 1) 925	78.0%
10000	73.7%
100000	69.8%

Tabelle 12: Namensgenauigkeit des MS-TDNN bei geschätztem Bereich

7 Flexible Erkennung

Ein buchstabierter Name läßt sich generell besser erkennen als ein gesprochener Name. Bei der Sammlung der Sprachdaten hat sich jedoch gezeigt, daß es den Menschen leichter fällt, ein Wort zu buchstabieren, wenn sie es vorher einmal kontinuierlich gesprochen haben. Um auf dieses Bedürfnis einzugehen wird ein flexibler Ansatz entwickelt, der sowohl nur buchstabierte, als auch gesprochene und buchstabierte Eigennamen als Spracheingabe akzeptieren kann.

7.1 Ansatz mit Modellierung

Mit der Modellierung, wie sie in Kapitel 4.2.1 vorgestellt wurde, kann für ein kleines Vokabular folgende Informationen bestimmt werden:

- **Klassifikation:** Bestimmung, um welches Szenario es sich handelt. Die Klassengenauigkeit gibt an, wie genau sich buchstabierte Namen von gesprochenen und buchstabierten Namen unterscheiden lassen. Die Ergebnisse werden als Verwechselbarkeitsmatrix dargestellt. Somit läßt sich ablesen, welche Klassen untereinander am häufigsten verwechselt werden.
- **Bereiche:** Mit dem Viterbi-Algorithmus kann der Bereich des buchstabierten Namens bestimmen werden.
- **Hypothese:** Es kann eine Hypothese ausgegeben werden. Bei der Modellierung mit HMM wird die Genauigkeit jedoch nicht hoch sein.

In einem ersten Versuch wurde der gesprochene Namen noch hinzugenommen, um einen Überblick über die Verwechslungen der einzelnen Klassen zu erhalten.

Je besser eine Klasse erkannt wird, desto näher wird die Namensgenauigkeit dieser Klasse, an den Baseline Ergebnissen liegen. Es traten kaum Verwechslungen der gesprochen und buchstabierten (beides) mit einer der anderen Klassen auf. Durch die Modellierung des gesprochenen und buchstabierten Namens mit einem HMM wurde als Baseline 86.1% Namensgenauigkeit erzielt. Dieses Ergebnis hat sich bei der flexiblen Erkennung nur um 0.1% verschlechtert. Bei der Buchstabiererkennung mit JANUS traten hingegen mehr Klassenverwechslungen auf, was sich auf die Erkennungsleistung auswirkt.

Liste 1	es wurde erkannt			Namens-
	gesprochen	buchstabiert	beides	genauigkeit
gesprochen	98.5%	0.5%	1.0%	62.0%
buchstabiert	0.5%	96.5%	3.0%	89.5%
beides	0.0%	1.0%	99.0%	86.0%

Tabelle 13: Die Verwechselbarkeitsmatrix gibt an, welche Klassen untereinander wie häufig verwechselt werden. Die Bezeichnung "beides" bedeutet, daß gesprochen und buchstabiert wurde.

7.2 Ansatz mit getrennter Erkennung

Die gute Klassengenauigkeit der Modellierung kann als Vorverarbeitung genutzt werden, um in einem zweiten Durchgang den λ-NBest Erkenner mit geschätzter Grenze zu benutzen. Von der besten Hypothese werden mittels Viterbi-Algorithmus die Bereiche des buchstabierten und wenn vorhanden des gesprochenen Namens bestimmt. Der buchstabierte Name wird dann mit dem MS-TDNN-Buchstabiererkenner erkannt. Theoretisch sollte für diesen eine Namensgenauigkeit von $96.5\% \cdot 0.984 = 95.0\%$ erreicht werden können. Für den gesprochenen und buchstabierten Namen erwartet man $95.8\% \cdot 0.99 = 94.8\%$ Namensgenauigkeit, bei Verwendung des λ-NBest Ansatz mit geschätzter Grenze (siehe Kapitel 4.2.4). Eine falsche Klassifizierung führt zu einem falschen Bereich, in dem die Buchstabensequenz vermutet wird. Dies hat aber nicht unbedingt eine fehlerhafte Erkennung zur Folge. Die theoretische Bestimmung der Namensgenauigkeit ist demnach eine untere Grenze.

Über Liste 1 (925 verschiedene Namen) wurde klassifiziert, ob buchstabiert oder gesprochen und buchstabiert wurde. Wurde nur buchstabiert erkannt, dann wird über die gesamte Äußerung mittels MS-TDNN der buchstabierte Name ermittelt. Ansonsten wird mit der geschätzten Grenze der λ-NBest Ansatz benutzt. Es wurden nur die gesprochenen Namen bewertet, die in der Bestenliste des Buchstabiererkenners vorkamen. Die Namensgenauigkeit und die Klassenverwechslungen sind in Tabelle 14 zusammengefaßt, Im Mittel ist eine Namensgenauigkeit von **95.7%** erzielt worden.

Bei nur buchstabierten Eigennamen hat der Klassifikator 22 Namen als gesprochen und buchstabiert erkannt, also falsch klassifiziert. Der λ-NBest Algorithmus konnte von diesen noch 7 Namen richtig erkennen.

Liste 1	es wurde erkannt		Namens-
	buchstabiert	beides	genauigkeit
buchstabiert	98.4%	1.6%	95.7%
beides	1.0%	99.0%	95.7%

Tabelle 14: Klassenverwechslungen und Namensgenauigkeit bei flexibler Erkennung über Liste 1. Die Bezeichnung "beides" bedeutet, daß gesprochen und buchstabiert wurde.

7.3 Verfahren bei mittlerer Vokabulargröße

Der Phonem-Buchstabiererkenner kann auf die neue Aufgabenstellung angepaßt werden, indem ein Sprachmodell erzeugt wird, bei dem neben gesprochenen und buchstabierten Namen auch nur Buchstabensequenzen zugelassen werden. Werden nur Buchstaben erkannt, dann handelt es sich bei der Äußerung um einen buchstabierten Namen, anderenfalls wurde der Name zusätzlich noch gesprochen. In Tabelle 15 wurde gemessen wie gut der Phonem-Buchstabenerkenner die Klassen unterscheiden kann.

Liste 1	buchstabiert	gesprochen + buchstabiert
buchstabiert	96.0%	4.0%
gesprochen + buchstabiert	23.2%	76.8%

Tabelle 15: Verwechslungsmatrix des Phonem-Buchstabiererkenners bei der Zuordnung von buchstabierten bzw. gesprochen und buchstabierten Namen

Nach der Klassifikation kann für eine nur buchstabierte Äußerung der Buchstabiererkenner benutzt werden. Dabei wird eine theoretische Namensgenauigkeit von $96.5\% \cdot 0.96 = 92.6\%$ erwartet. Für den gesprochenen und buchstabierten Namen sinkt die Erkennungsrate aufgrund der fehlerhaften Klassifikation auf $96.9\% \cdot 0.768 = 74.4\%$ bei 1000 Namen. Dieser Wert deutet darauf hin, daß zur Klassifikation ein anderes, verbessertes Verfahren zu wählen ist.

8 Zusammenfassung

Ziel dieser Arbeit war die maschinelle Erkennung von gesprochenen und buchstabierten Eigennamen. Dabei sollte festgestellt werden, ob die redundante Information beider Äußerungen zur Verbesserung der Erkennungsleistung beitragen kann.

8.1 Baseline

Aus den einführenden Experimenten, bei denen die Erkennungsleistung der gesprochenen und buchstabierten Eigennamen getrennt betrachtet wurde, zeichnete sich ab, daß der gesprochene Name nur eine unterstützende Funktion haben kann. Weiter wurde festgestellt, daß die Hinzunahme des gesprochenen Namens, also "gesprochen und buchstabiert", die Ergebnisse verschlechtern kann.

	buchstabiert	gesprochen	gesprochen+buchstabiert
Liste 1	96.5%	60.0%	86.1%

8.2 Vereinigung der getrennten Erkennung

Um eine Verschlechterung der Erkennung durch Hinzunahme des gesprochenen Namens zu vermeiden, wurde ein Verfahren entwickelt, das wir den λ-NBest Erkenner nennen. Der Parameter λ ist hierbei ein Gewichtungsfaktor. Die Bewertungen der Hypothesen, die von den beiden Erkennern geliefert werden, werden mit diesem Faktor gewichtet und neu sortiert. Über ein Cross-Validation-Set kann dieser Faktor eingestellt werden. Diese Vorgehensweise hat jedoch folgende Nachteile:

1. Die gesprochenen und buchstabierten Namen müssen getrennt voneinander vorliegen, d.h. ihre Bereiche in einer Äußerung bekannt sein. Sollten die Bereiche des gesprochen und buchstabierten Namens nicht bekannt sein, so können sie in einem Vorverarbeitungsschritt ermittelt werden.

2. Die Einstellung des Parameters λ geschieht heuristisch und muß bei Änderung der äußeren Bedingungen neu erfolgen.

Die getrennte Verarbeitung bietet jedoch den Vorteil, spezialisierte Erkenner zu verwenden. Diese können getrennt voneinander trainiert und gegebenenfalls ausgetauscht werden. Die Vereinigung mit dem λ-NBest Erkenner bewirkt bei bekannten Bereichen immer eine Verbesserung der Namensgenauigkeit gegenüber der ausschließlichen Erkennung des buchstabierten Namens.

8.3 Unterschiedlich große Namenslisten

Es werden nun die wichtigsten Ergebnisse in Tabelle 16 präsentiert. In den Zeilen wird die Namensliste größer und die Informationsmenge, die a priori zur Verfügung steht, kleiner. In den Spalten werden Anfrageszenario 1 und 2 gegenübergestellt. Zum Vergleich sind ebenfalls die Ergebnisse der reinen Buchstabenerkennung angegeben. Alle Versuche wurden über Liste 1 mit 1337 Äußerungen durchgeführt. Die Ergebnisse geben die Namensgenauigkeit an. Eine Namensgenauigkeit von 95.9% bedeutet also, daß von 1337 Namen 1282 richtig erkannt wurden.

Größe des Vokabulars			Kombination	
		Buchstabiert	Szenario 1	Szenario 2
klein	1k	95.6%	96.1%	95.8%
mittel	1k	96.5%	97.7%	96.9%
	10k	93.2%	94.8%	92.7%
	100k	86.5%	89.5%	88.1%
groß	10k	93.2%	93.2%	73.7%
	100k	86.5%	86.5%	69.8%

Tabelle 16: Die Ergebnisse auf einen Blick.

Vergleichen wir die Buchstabiererkennung mit Szenario 1, so kann mit dem λ-NBest Erkenner bei festem λ immer eine Verbesserung erzielt werden. (Bei großen Namenslisten wurde der gesprochene Name nicht berücksichtigt.)

Buchstabiert: Hier steht die Namensgenauigkeit der reinen Buchstabiererkennung. Die Sprachdaten wurden wie in Szenario 2 aufgenommen. Damit dennoch Szenario 1 und nur buchstabiert untersucht werden konnte, wurde die Grenze zwischen den beiden Varianten von Hand bestimmt. Es können trotzdem Koartikulationseffekte auftreten, wodurch die reine Buchstabiererkennung beeinträchtigt wird.

Szenario 1: Der Bereich des buchstabierten und gesprochenen Namens ist bekannt und sie können getrennt voneinander untersucht werden.

Szenario 2: Der Eigenname wurde innerhalb einer Äußerung gesprochen und buchstabiert.

Der Unterschied zwischen Szenario 1 und Szenario 2 liegt einzig in dem Vorhandensein der Bereiche des jeweiligen Repräsentanten des Namens. Durch die große Dominanz der Buchstabierung konnten die besten Resultate für Szenario 2 durch eine automatische Bestimmung der Bereiche und damit Reduktion des Problems auf Szenario 1, erreicht werden. Da die Bestimmung dieser Bereiche jedoch fehlerhaft ist, sind die Ergebnisse schlechter als die von Szenario 1. Die Erkennungsleistung wurde im Vergleich zur reinen Buchstabiererkennung, außer bei großen Namenslisten, verbessert.

klein: Die Namensliste enthält nur wenige Namen. Von jedem Namen ist seine Transkription bekannt und er kann als Wort ins Wörterbuch eines Erkenners aufgenommen werden. Für Szenario 1 werden beide Erkenner getrennt voneinander verwendet. Die jeweils besten Hypothesen werden gewichtet und neu sortiert. Die Schätzung der Grenze für Szenario 2 kann durch den Viterbi-Algorithmus bestimmt werden. Damit kann der richtige Kandidat in 86% aller Fälle richtig ermittelt werden. Von diesem werden die Bereiche bestimmt und dann der λ-NBest Ansatz benutzt.

mittel: Die Größe der Namensliste übersteigt die Kapazität des Erkenners, die Namen als Vokabular zu benutzen. Jedoch steht immer noch für jeden Namen seine Transkription zur Verfügung. Deshalb wird eine zweiphasige Erkennung benutzt, bei der zunächst eine Reduktion des Vokabulars durchgeführt wird.

Nach dieser Phase erhalten wir eine reduzierte Namensliste, wir nennen sie Kandidaten. Mit diesen Kandidaten kann wiederum wie bei kleinen Vokabularien verfahren werden. Für Szenario 1 werden die Kandidaten mit dem Buchstabiererkenner erzeugt.

Als Vokabular werden nur die Buchstaben benutzt und über eine Baumgrammatik Eigennamen als Kandidaten bestimmt. Von diesen Kandidaten wird die Bewertung der gesprochenen Namen ermittelt und anschließend wieder der λ-NBest Erkenner verwendet.

Für Szenario 2 muß zunächst die Grenze berechnet werden. Dazu wurde ein Phonem-Buchstabenerkenner entwickelt, der den gesprochenen Namen am Anfang einer Äußerung "überlesen" kann und schließlich den buchstabierten Namen ausgibt. Mit diesem Erkenner kann die Grenze bestimmt werden. Diese Grenze erweist sich als zu ungenau. Deshalb werden aus den Buchstabensequenzen mögliche Namen ermittelt und diese modelliert. Von dem Kandidaten mit der besten Bewertung wird die Grenze ausgegeben. Der beste Kandidat war, bei der kleinen Liste, zu 91.5% korrekt. Damit war die Schätzung der Grenze robuster, als bei den Verfahren mit kleinem Vokabular und die Namensgenauigkeit besser. Ein weiterer Grund für die bessere Namensgenauigkeit bei mittlerer Vokabulargröße ist, daß nur die gesprochenen Namen bewertet wurden, die in der Bestenliste des Buchstabiererkenners vorkommen.

groß: Wie zuvor ist die Namensliste zu groß für den Erkenner, jedoch muß jetzt auf die Transkription des Eigennamens verzichtet werden. In Szenario 1 wurde deshalb nur die Buchstabenerkennung durchgeführt. In Szenario 2 wurde die Grenze durch den Phonem-Buchstabenerkenner relativ ungenau ermittelt. Auf dem Bereich des buchstabierten Namens wird der Buchstabenerkenner benutzt. Um hier eine Verbesserung zu erreichen, könnte ein automatisches Transkriptionsverfahren oder ein wissensbasierter Ansatz benutzt werden.

8.4 Flexible Erkennung

Bei der flexiblen Erkennung kann die Spracheingabe des Eigennamens sowohl buchstabiert als auch gesprochen erfolgen. Der Erkennungsprozeß soll automatisch feststellen, um welche Eingabeklasse es sich handelt und entsprechend die Äußerung verarbeiten. Es wurde bisher betrachtet, wie gut

Beschreibung	Buchstabiert	Beides
festgelegt	96.5%	95.8%
Flexibel	95.7%	95.7%

Tabelle 17: Der flexible Ansatz im Vergleich zu den bisherigen Ergebnissen. Die Experimente wurden mit Liste 1 (925 verschiedenen Eigennamen) durchgeführt. Die Zahlen geben die Namensgenauigkeit an.

die Erkennungsleistung bei "festgelegter" Eingabeklasse sein kann. Bei der flexiblen Erkennung, die im Rahmen dieser Diplomarbeit auf kleinen Listen durchgeführt wurde, werden diese Werte aufgrund des Klassifizierungsfehlers nicht erreicht.

Man beachte jedoch folgendes: Wenn bisher auf kleinen Listen nur gesprochen und buchstabiert werden konnte, hat man mit dem flexiblen Ansatz nahezu die gleiche Erkennungsleistung, aber die Möglichkeit einen Namen auch ausschließlich zu buchstabieren und dabei nicht mit Leistungseinbußen bei der Erkennungsleistung zu rechnen.

Dieses Ergebnis motiviert die Fortsetzung im Bereich flexibler und damit anwendungsfreundlicher Ansätze.

9 Ausblick

9.1 Finden der Grenze

Von zentraler Bedeutung hat sich das genaue Auffinden des Bereichs der buchstabierten Namen erwiesen. Hat man diesen Bereich gefunden, so kann ein spezialisierter Buchstabenerkenner (MS-TDNN) sehr gute Ergebnisse erzielen.

Eine Suche nach der ersten Pause in einem Sprachsignal, in dem der gesprochene und der buchstabierte Name erwartet wird, kann nur als erste Abschätzung dienen. Wenn auch Doppelnamen berücksichtigt werden sollen, wie es in dieser Diplomarbeit der Fall war, ist diese Methode gänzlich unbrauchbar.

Mit dem Phonem-Buchstabenerkenner konnte zwar eine Schätzung des Bereichs durchgeführt werden. Sehr viel genauer ist es jedoch, den gesprochenen und buchstabierten Namen als ein HMM zu modellieren und anhand von diesem die Bereiche zu bestimmen. Insgesamt ist diese Vorgehensweise sehr aufwendig. Man ist auf eine gute Schätzung der Bereiche angewiesen, denn auf ihr bauen die weiteren Erkenner auf. Es sollte untersucht werden, ob es schnelle, genaue Verfahren gibt, die bessere Ergebnisse liefern.

9.2 Wahl der Eigennamen

In dieser Diplomarbeit wurden hauptsächlich deutsche Eigennamen als Spracheingabe verwendet. Interessant wäre ein Vergleich der Ergebnisse mit anderen Eigennamen oder einer anderen Sprache.

9.3 Flexible Erkennung

Es wurde im Kapitel 7 für kleine Vokabularien untersucht, inwieweit die vorgestellten Verfahren benutzt werden können, um eine flexiblere Eingabe für den Anwender zu ermöglichen. Interessant wären auch Ergebnisse für größere Vokabularien.

9.4 Eingebunden in natürliche Sprache

Für den Benutzer am angenehmsten ist ein System, das Sätze akzeptiert und aus diesen die Anfrage extrahiert. Darunter würde dann auch das Auffinden

des gesprochenen und buchstabierten Namens fallen. Hat man diese ermittelt, lassen sich die in dieser Diplomarbeit beschriebenen Methoden verwenden.

9.5 Telefonauskunftssystem

Der Bedarf an Telefonauskunftsystemen ist groß und es existieren bereits eine Vielzahl unterschiedlichster Systeme [KSS95, MWK96, YTTK89]. Es wurde im Rahmen dieser Diplomarbeit eine Demonstration erstellt, die für eine kleine Namensliste eine flexible Erkennung ermöglicht. Eine Erweiterung auf größere Namenslisten und die Integrierung des spezialisierten Buchstabiererkenners wären für ein brauchbares Auskunftssystem dringend erforderlich.

A Anhang

In diesem Kapitel werden die Datensätze beschrieben, mit denen die Experimente durchgeführt wurden.

A.1 Transkriptionen

Zur Erkennung gesprochener Namen werden ihre Transkriptionen benötigt. Von der TU Berlin und der TELEKOM wurde, innerhalb des ONOMASTICA-Projekts, ein Lexikon Deutscher Eigennamen aufgebaut, welches nicht nur Nachnamen, sondern auch Vor-, Straßen-, Städte- und Firmennamen entält.

Kontaktadressen:

Andreas Mengel	Antje Wirth (FZ131b)
Technische Universitaet Berlin	Deutsche Telekom
Institut fuer Kommunikationswissenschaft	Technologiezentrum
Einsteinufer 17	Am Kavalleriesand 3
10587 Berlin	64295 Darmstadt
Tel.: 030 - 314 26675	Tel.: 0 61 51 - 83 3542
Fax: 030 - 314 21143	Fax: 0 61 51 - 89 5234
Email: mengel@kgw.tu-berlin.de	Email: wirth@fz.telekom.de
www: http://www.kgw.tu-berlin.de/~mengel	

Die Hälfte ihrer Datenbank wurde uns freundlicherweise zur Verfügung gestellt. Da diese im SAMPA[13] Format gespeichert sind mußte eine Anpassung an die Umschrift der verwendeten Software ins IPA[14] Format vorgenommen werden. Es konnte nicht sichergestellt werden, daß die daraus resultierenden Transkriptionen konsistent mit den Konventionen des JANUS-Erkenners sind.

Laut Angaben des ONOMASTICA Projekts werden folgende Phoneme benutzt. Die korrespondierende JANUS Transkription wurde in einer weiteren Spalte angegeben:

[13] Speech Assessment Methods Phonetic Alphabet
[14] International Phonetic Alphabet

Vokale

IPA	SAMPA	Beispiel	JANUS
304	a	Deutschl(a)nd	A
303	E	B(e)lgien	E
301	i	Hait(i)	I
319	I	F(i)nnland	I
307	o	Kyot(o)	O
306	O	Sch(o)ttland	O
308	u	Korf(u)	U
321	U	Edinb(u)rgh	U
320	Y	T(ue)rkei	UE
311	9	K(oe)ln	OE
304,503	a:	It(a)lien	AH
302,503	e:	Br(e)men	EH
303,503	E:	D(ae)nemark	AEH
301,503	i:	(I)sland	IE
307,503	o:	P(o)len	OH
308,503	u:	Toul(ou)se	UH
309,603	y:	L(ue)neburg	UEH
310,503	2:	Gr(oe)nland	OEH
324	6	Brem(er)haven	ER2
322	@	Spani(e)n	E2

Diphthonge

IPA	SAMPA	Beispiel	JANUS
306,320	OY	N(eu)stadt	EU
304,319	aI	W(ei)terstadt	AI
304,321	aU	L(au)terbach	AU
303,319	EI	h(ey)	E I

Konsonanten

IPA	SAMPA	Beispiel	JANUS
101	p	(P)aris	P
102	b	(B)onn	B
103	t	Por(t)ugal	T
104	d	(D)eutschland	D
109	k	(K)openhagen	K
110	g	(G)riechenland	G
128	f	(F)rankfur	F
129	v	(W)ien	V
132	s	Am(s)terdam	S
133	z	(S)ophia	Z
134	S	(Sch)weiz	SCH
135	Z	Gara(g)e	SCH
138	C	Frankrei(ch)	CH
153	j	(J)ugoslawien	J
140	x	Kasa(ch)stan	X
114	m	Daene(m)ark	M
116	n	Norwege(n)	N
119	N	E(ng)land	NG
155	l	(L)uxemburg	L
123	R	(R)om	R
113	?	()Amsterdam	?
146	h	(H)elsinki	H

Sonstige

SAMPA	Verwendung	JANUS
'	Hauptbetonung	
,	Nebenbetonung	
-	Silbengrenze	-

Es existierte bereits ein "PERL"-Skript, welches SAMPA Format in die JANUS-Erkenner spezifische Phonemumschrift überführt. Dieses Skript wur-

Format	Konvention für 3 Zeichen								
SAMPA	a:6	i:6	y:6	2:6	E:6	u:6	o:6	t+s	t+S
JANUS	AHR	IHR	UEHR	**OEH R**	AEHR	UHR	OHR	TS	TSCH

Format	Konvention für 2 Zeichen								
SAMPA	pf	ts	tS	i:	96	e:	2:	E:	u:
JANUS	**P F**	TS	TSCH	IE	**OE ER2**	EH	OEH	AEH	UH
SAMPA	aI	aU	OY	I6	Y6	y:	E6	U6	O6
JANUS	AI	AU	EU	IR	**UE ER2**	UEH	ER	UR	OR
SAMPA	o:	a:	a6	a~	ZZ				
JANUS	OH	AH	AR	ANG	**SCH**				

Format	Konvention für 1 Zeichen												
SAMPA	Q	?	#	.	+	–	'	"	;	9	2	e	E
JANUS	?	?				–				OE	OE	E	E
SAMPA	:	%	,	p	b	t	d	k	g	a	@	6	l
JANUS				P	B	T	D	K	G	A	E2	ER2	L
SAMPA	f	v	s	z	S	Z	C	x	r	R	h	u	U
JANUS	F	V	S	Z	SCH	SCH	CH	X	R	R	H	U	U
SAMPA	n	N	o	O	j	i	I	m	y	Y			
JANUS	N	NG	O	O	J	I	I	M	UE	UE			

Tabelle 18: Transformationstabelle von SAMPA in das JANUS-eigene Format.

de an den hervorgehobenen Stellen angepaßt, bzw. erweitert. Zunächst wird versucht, 3 Zeichen auf ein passendes Phonem abzubilden. Kann kein passendes gefunden werden, wird versucht zwei Zeichen zu finden, die transformiert werden können. Bleibt auch dies ohne Erfolg, wird Zeichen für Zeichen transformiert.

A.2 Sprachdaten

Zur Erkennung von deutschen Nachnamen existiert eine an der Universität Karlsruhe gesammelte Sprachdatenbank. Diese enthält 2900 Sätze von 60 Sprechern. Jeder Satz enthält einen gesprochenen und buchstabierten Nachnamen: "Meyer M E Y E R" . Die Grenze zwischen dem gesprochenen und buchstabierten Namen wurde automatisch ermittelt und dann manuell nach-

verarbeitet, um den Segmentierungsfehler möglichst klein zu halten. Zu jeder Aussage wurde festgehalten, was tatsächlich gesagt wurde. Die Daten wurden über Mikrophon mit 16000Hz Abtastfrequenz aufgezeichnet. Mit dieser Segmentierung lassen sich buchstabierte und gesprochene Namen getrennt voneinander erkennen. Jedoch können Koartikulationseffekte auftreten, da den Sprechern mittgeteilt wurde, keine explizite Pause zwischen dem gesprochenen und buchstabierten Namen machen zu müssen.

Erste Durchläufe mit den Erkennern und ein Testhören der Sprachdaten lieferte drei Sprecher, deren Aufnahme stark von denen der anderen abweicht:

fhh1 Hierbei handelte es sich um eine Sprecherin mit sehr starkem Akzent, die außerdem noch Schwierigkeiten hatte, korrekt zu buchstabieren.

mal3 Ähnlich wie fhh1 hat mal3 einen sehr starken Akzent.

mrg3 Bei der Aufnahme dieses Sprechers ist ein technischer Fehler unterlaufen. Die Lautstärke der Aufnahme ist zu gering. Eine problemlose Verarbeitung war somit nicht möglich.

Damit in den folgenden Versuchen ein Maß für die Erkennungsleistung gefunden werden kann, muß bekannt sein, was tatsächlich gesprochen wurde. Die Sprecher hatten für die Versuche eine Namensliste mit jeweils 50 Namen. Diese sollten gesprochen und buchstabiert werden. Damit ist bekannt, was die Sprecher in der jeweiligen Äußerung hätten sagen sollen (idealisiert). Beim Durchhören der Daten wurden allerdings zum Teil erhebliche Abweichungen zwischen dem geschriebenen und dem gesprochenen festgestellt. Besonders gravierend waren die Buchstabierungsfehler, Stotterer und Ausdrücke der Art "aeh" , "also", usw. Zu jeder Äußerung wurde notiert was tatsächlich (real) gesprochen wurde. Grundsätzlich wurde die Namensgenauigkeit anhand der idealen Namen gemessen. Bei den Baselines für den spezialisierten Buchstabiererkenner wurde die reale Liste genommen, damit die Buchstabengenauigkeit mit bisherigen Veröffentlichungen verglichen werden kann.

Für die Erkennung der gesprochenen Namen werden deren Transkriptionen benötigt. Mit anderen Worten: Es können nur die kontinuierlich gesprochenen Namen erkannt werden, deren Transkription bekannt ist.

Aus diesen Gegebenheiten wurden die Daten folgendermaßen aufgeteilt:

- **Sprecherbezogene Daten:**

  s60 eine Liste der Äußerungen aller Sprecher

s60p eine Liste der Äußerungen aller Sprecher von denen eine Transkription vorhanden ist.

s57 57 Sprecher, deren Aufnahme und Akzent passabel ist.

s57p Die Äußerungen der 57 Sprecher mit passablem Akzent, für deren gesprochene Namen auch eine Transkription vorhanden ist.

- **Qualitätsmerkmale:**

  0 Der Name wurde korrekt gesprochen und buchstabiert. Es wurden keine Worte zusätzlich gesprochen.

  1 Der Namen wurde korrekt gesprochen, aber beim Buchstabieren wurden Buchstaben ausgelassen oder hinzugefügt.

  2 Es wurden Worte gesprochen, die nicht zum Namen gehören. (z.B.: "aeh", "also")

Kennung	Qualität 0	Qualität 1	Qualität 2	Qualität 012
s57	2650	94	36	2780
s57p	1301	36	9	1346

Liste	Namen	Äußerungen
Liste 1 (s57p in Qualität 0+1)	925	1337
Liste 2 (s57 in Qualität 0+1)	1927	2774

Tabelle 19: Anzahl der Namen und Äußerungen.

A.3 Liste der Eigennamen

Für die Versuche mit großen Namenslisten wurde eine Liste der in Karlsruhe wohnhaften Personen verwendet, von denen Transkriptionen vorhanden sind.

Beschreibung	Anzahl Namen	Anzahl ohne doppelte
ONOMASTICA	288464	288464
Karlsruher Namen (KN)	111865	32250
KNp	53721	13974
KNp mit s57p	55066	13988

Für die Liste mit 100.000 Namen wurden die Namen aus *KNp mit s57p* genommen (ohne mehrfache) und die restlichen mit Namen aus der ONOMASTICA Liste aufgefüllt. Damit sind alle gesprochenen Namen in der Liste repräsentiert und für jeden dieser Namen existiert die Transkription. Die Häufigkeit der Namen hätte nur über die Namen der Karlsruher Liste und der Sprachdaten bestimmt werden können. Das hätte sich vorteilhaft für die Experimente ausgewirkt, aber wenig mit einer realen Namenshäufigkeit zu tun gehabt. Deshalb wurden alle Namen als gleichwahrscheinlich angenommen.

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[Bak75] J. Baker. The DRAGON System - an Overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23:24–29, 1975.

[BH95] Martin Betz and Hermann Hild. Language Models for a Spelled Letter Recognizer. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 856–859. IEEE, Mai 1995.

[DH73] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York,Chichester, Brisbane, Toronto, Signapore, 1973.

[FBC95] Mark Fanty, E. Barnard, and Ronald Cole. *Alphabet Recognition*, pages F2.1:1–8. IOP Publishing Ltd and Oxford University Press, 1995.

[FGH+97] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal. The Karlsruhe-Verbmobil Speech Recognition Engine. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1997.

[Gü91] Drosdowski Günther. *Brockhaus Enzyklopädie.* Brockhaus, 1991.

[HW93] Hermann Hild and Alex Waibel. Connected Letter Recognition with a Multi-State Time Delay Neural Network. In *Advances in Neural Network Information Processing Systems (NIPS-5-)*, pages 712–719. Morgan Kaufmann, 1993.

[HW96] Hermann Hild and Alex Waibel. Recognition of Spelled Names over the Telephone. In *Proceedings Fourth International Conference on Speech and Language Processing*, pages 346–349, Oktober 1996.

[Jel76] Frederick Jelinek. Continuous Speech Recognition by Statistical Methods. *Proceedings of the IEEE*, 64(4):532–556, 1976.

[JLMG93] D. Jouvet, A. Laine, J. Monnè, and C. Gagnoulet. Speaker-Independent Spelling Recognition over the Telephone. In *Proc.*

IEEE International Conference on Acoustics, Speech and Signal Processing, pages 235–238. IEEE, 1993.

[JVFM95] Jean-Claude Junqua, Stephane Valente, Dominique Fohr, and Jean-Francois Mari. An N-Best Strategy, Dynamic Grammars and Selectively Trained Neural Networks for Real-Time Recognition of Continuously Spelled Names over the Telephone. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 852–855. IEEE, Mai 1995.

[KSS95] C.A. Kamm, C.R. Shamieh, and S. Singhal. Speech Recognition Issues for Directory Assistance Applications. *Speech Communication*, 17:303–311, 1995.

[KTT95] H. Kanazawa, M. Tachimori, and Y. Takebayashi. A Hybrid Wordspotting Method for Spontaneous Speech Understanding using Word-Based Pattern Matching and Phonem-Based HMM. In *EUROSPEECH'95 (4th European Conference on Speech Communication and Technology)*, pages 289–292, September 1995.

[LRS83] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. *Bell System Technical Journal*, 62(4):1035–1074, April 1983.

[MWK96] Eric McDermot, Eric A. Woudenberg, and Shigeru Katagiri. A Telephone-Based Directory Assistance System adaptively trained using Minimum Classification Error / Generalized Probabilistic Descent. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3346–3349. IEEE, März 1996.

[Ney84] Hermann Ney. The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 263–271. IEEE, April 1984.

[Ous94] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, 1994.

[PGMF86] J. Picone, G. Goudie-Marshall, Doddington, and W. Fisher. Automatic Text Alignment for Speech System Evaluation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 780–784. IEEE, 1986.

[Rab89] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE.* IEEE, 1989.

[SC90] Hiroaki Sakoe and Seibi Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*, chapter 4.3, pages 159–165. Morgan Kaufmann, 1990.

[SR75] Ronald W. Schafer and Lawrence R. Rabiner. Digital Representation of Speech Signals. *Proceedings of the IEEE*, 63(4):662–667, 1975.

[ST95] E.G. Schukat-Talamazzini. *Automatische Spracherkennung.* Vieweg, Braunschweig, Germany, 1995.

[Thi93] Christine Thielen. Eigennamen in Texten. Ein inkrementelles Verfahren zum Eigennamen-Tagging für das Deutsche. Master's thesis, Universität Trier, 1993.

[VZ70] V. Velicho and N. Zagaryko. Automatic Recognition of 200 words. In *Int. J. Man-Machine Studies*, pages 223–235, 1970.

[WHH+89] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-37:328–339, März 1989.

[YTTK89] Minami Yasuhiro, Matasuoka Tatsuo, Yamada Tomokazu, and Shikano Kiyohiro. Very Large Vocabulary Continous Speech Recognition Algorithm for Telephone Directory Assistance. *Computer, Speech and Language*, 3(2), 1989.