Ein automatisches Indexierungssystem für Fernsehnachrichtensendungen

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurswissenschaften
von der Fakultät für Informatik an der Universität Karlsruhe
genehmigte

Dissertation

von

Thomas Kemp

aus Brühl

Tag der mündlichen Prüfung: 13. Dezember 1999
Erster Gutachter: Prof. Dr. A. Waibel
Zweiter Gutachter: Prof. Dr. A. Schmitt

Zusammenfassung

Video-Archivierung und das Suchen in großen Beständen von Videodaten sind in den letzten Jahren immer mehr ins Zentrum des Interesses gerückt. Die Erstellung eines suchbaren Index für Videodaten erfordert jedoch sehr viel Handarbeit und ist für große Bestände prohibitiv aufwendig.

In der vorliegenden Arbeit wird der Prototyp des View4You-Systems vorgestellt, eines vollautomatischen Systems zur Erstellung einer Video-Datenbank. Als Videodaten werden Fernseh-Nachrichtensendungen verwendet. Diese wurden im Rahmen des View4You-Projektes täglich mit Hilfe eines Satellitenreceivers aufgenommen und in digitalisierter Form abgespeichert. Da jede Nachrichtensendung eine Vielzahl von unterschiedlichen Themen behandelt, von denen den Benutzer in der Regel nur eines interessiert, müssen die Aufnahmen zunächst segmentiert werden. Diese Aufgabe wird von einem audiosignalbasierten Segmentierer ausgeführt. Verschiedene Ansätze aus der Literatur wurden implementiert und mit einem neuentwickelten Ansatz verglichen. Dabei erzielte der neu entwickelte Ansatz mit einem F-Maß von 78% die besten Kennzahlen aller evaluierten Systeme.

Der eigentliche Index besteht aus einer Transkription (Verschriftung) der Sprache des Videofilms, die durch automatische Spracherkennung erzeugt wird. Der Spracherkenner stellt eine der Hauptkomponenten des Systems dar. Die Spracherkennung von Nachrichtensendungen ist eine relativ schwierige Aufgabe. Sie ist charakterisiert durch kontinuierliche, sprecherunabhängige Spracherkennung mit uneingeschränktem Vokabular, hohe Variabilität in den Aufnahmekanälen, eine Vielfalt von Störgeräuschen in den Korrespondentenberichten sowie häufige Sprecher im Hintergrund. Der Spracherkenner des View4You-Systems baut auf dem am Institut entwickelten Janus-Toolkit auf und erzielt auf Nachrichtensendungen eine Fehlerrate von 19,4%.
Zur Verbesserung der Wortakkuratheit des Spracherkenners wurden zwei neue Verfahren (zur Vokabularadaption und zur teilüberwachten Adaption) entwickelt und erfolgreich getestet. Dabei konnte durch beide Verfahren jeweils eine Fehlerratenreduktion um 5% (relativ) erzielt werden.

Die dritte Hauptkomponente neben dem Segmentierer und dem Spracherkenner ist das Informationssystem. Es hat die Aufgabe, die natürlichsprachlichen Anfragen des Benutzers zu verarbeiten und aus der Kollektion der verschrifteten Segmente diejenigen herauszusuchen, die im Hinblick auf die Anfrage relevant sind.

Einen wichtigen Beitrag dieser Arbeit stellt die Entwicklung eines Verfahrens zum unüberwachten Training eines Spracherkenners dar. Mit Hilfe der kontinuierlich anfallenden Audiodaten aus den Aufnahmen der Nachrichtensendungen wird ein vollautomatisch arbeitendes, inkrementell lernendes System erstellt, das seine Spracher-

kennungsleistung durch reines 'Zuhören' verbessert. Das vollautomatisch trainierte System erzielt eine Fehlerrate von 20,5% auf der Teststichprobe. Ein auf Adaption basierender Ansatz kann demgegenüber, aufgrund der niedrigen Anzahl an adaptierbaren Parametern, keine relevanten Verbesserungen des initialen - auf sehr wenig Daten trainierten - Systems erzielen.
Um das unüberwachte Training durchführen zu können, mußte zunächst ein Konfidenzmaß implementiert werden, mit dem der Spracherkenner seine eigene Ausgabe beurteilen kann. Dabei wurde ein neuer Algorithmus entwickelt und im Rahmen einer ausführlichen Untersuchung mit einer Vielzahl bereits aus der Literatur bekannter Verfahren erfolgreich evaluiert. Dieser Algorithmus beziehungsweise Varianten davon werden mittlerweile auch von anderen international agierenden Forschungsgruppen eingesetzt.

Die Arbeit wird abgeschlossen durch eine komplette Systemevaluation (End-to-End Evaluation) des gesamten View4You-Systems.

Danksagung

Diese Arbeit entstand am Institut für Logik, Komplexität und Deduktionssysteme in der *Interactive Systems Labs* Gruppe. Deren Leiter, Professor Dr. Alex Waibel, danke ich herzlich für die Betreuung dieser Arbeit. Durch sein unermüdliches, weltweites Engagement eröffnete er mir die Gelegenheit, in einer mit internationalen Kontakten und Geräten erstklassig ausgestatteten Forschungsumgebung arbeiten zu können. Professor Waibel war eine beständige Quelle guter Ideen und ehrgeiziger Ziele; er verstand es immer wieder, uns aus nah und fern mit viel guter Laune und Humor zu neuen Leistungen anzuspornen. Ebenfalls zu Dank verpflichtet bin ich Herrn Prof. Dr. Alfred Schmitt, der das Korreferat zu dieser Arbeit übernommen hat.

Erfolgreiches Arbeiten im komplexen, rechen- und datenintensiven Feld der Spracherkennung erfordert vielfältige Interaktion aller Beteiligten. Für eine gute und freundliche Zusammenarbeit möchte ich mich bei meinen jetzigen und ehemaligen Kollegen herzlich bedanken, darunter: Markus Baur, Uli Bodenhausen, Finn-Dag Buo, Noah Coccaro, Matthias Denecke, Frank Dreilich, Paul Duchnowski, Michael Finke, Jürgen Fritsch, Petra Geutner, Hermann Hild, Detlef Koll, Stefan Manke, Uwe Meier, Florian Metze, Wolfgang Minker, Jürgen Reichert, Klaus Ries, Ivica Rogina, Thomas Schaaf, Tanja Schultz, Rainer Stiefelhagen, Hagen Soltau, Bernhard Suhm, Martin Westphal und Monika Woszczyna. Besonders gewinnbringend war mir die in Wort und Tat erfahrene fachliche und persönliche Unterstützung von Hermann Hild, Michael Finke, Hagen Soltau, Thomas Schaaf und Monika Woszczyna.

Ein herzliches Dankeschön auch an die Mitglieder der View4You-Projektgruppe Jochen Gürtler, Peter Scheytt, Michael Schmidt, Borislav Tomaz und Manfred Weber, ohne deren tatkräftige Mitarbeit das View4You-Projekt nicht hätte realisiert werden können. Ich möchte mich besonders bei Manfred Weber für die stets zuverlässige Durchführung der Datensammlung und Transkription während der gesamten Laufzeit des Projektes bedanken.

Das seit Gründung des Lehrstuhls stetig wachsende Netz von Rechnern und Festplatten wurde und wird durch die intensive Betreuung von Stefan Manke, Markus Baur, Martin Klein und nunmehr Frank Dreilich am Leben gehalten. Für einen rei-

bungslosen organisatorischen Ablauf garantierten im Sekretariat immer freundlich und hilfsbereit Silke Dannenmaier, Ingrid Gemen, Evelyn Kimmich und Sonja Seitz.

Eine große Hilfe war mir die liebevolle Unterstützung durch meine Frau Anja, die mir geduldig die vielen langen Abende nachsah, die ich nicht ihr, sondern dieser Arbeit zukommen ließ. Danken möchte ich auch meiner Familie und meinen Verwandten, die ebenfalls mit viel Anteilnahme und Einsatz für mein Wohl sorgten, besonders meinen Eltern Manfred und Helga Kemp, deren warmherzige und liebevolle Erziehung mir die denkbar besten Startvoraussetzungen gab.

Ich widme diese Arbeit meinen beiden Töchtern Daniela und Judith, die an manchem Abend auf ihren Vater verzichten mußten. Unter allen und allem im Leben nehmen Kinder eine besondere Stellung ein, und ich bin dankbar, daß ich diese Erfahrung machen kann.

Karlsruhe, 28. Oktober 1999

Thomas Kemp

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Speicherung von Informationen hat in den letzten Jahrzehnten eine stürmische Entwicklung durchgemacht. Bis in die siebziger Jahre des zwanzigsten Jahrhunderts dominierten die Printmedien die Repräsentation von Wissen; im Wesentlichen als Buch, für Publikationen, die ein größeres Maß an Aktualität erforderten, als Zeitschrift oder Zeitung. Der Löwenanteil des neu entdeckten und publizierten Wissens wird auch nach wie vor über Printmedien abgedeckt. Durch den stetigen Preisverfall der magnetischen und optischen Speichermedien nimmt jedoch die Bedeutung von Video und Audio als Medium zum Transfer von Wissen zu. Videodaten sind im Gegensatz zu Büchern bimodal; sie bestehen aus Bild und Ton. Zahlreiche Anwendungen *erfordern* die Bimodalität, wie zum Beispiel ein Lehrfilm zur Vogelbestimmung, bei dem die Tiere in ihrem natürlichen Habitat gezeigt werden und die charakteristischen Rufe und Singstrophen abgespielt werden können, oder ein Lehrfilm über musikalische Themen, bei dem die Tondaten natürlich eine sehr wichtige Rolle spielen.

Des Weiteren können Filme *Abläufe* von Bewegungen oder Vorgängen deutlicher machen als gedrucktes Material, so daß zum Beispiel beim Eintrainieren von Bewegungsabläufen für Sportler oder Soldaten schon seit geraumer Zeit Lehrfilme anstelle von Lehrbüchern eingesetzt werden. Dasselbe gilt für Darstellungen von sich bewegenden Maschinen.

Bewegte Bilder mit den dazugehörenden Tönen entsprechen der normalen kognitiven Wahrnehmung des Menschen. Auch relativ lange und ausführliche verbale Beschreibungen sind daher in der Regel nicht in der Lage, dieselbe Plastizität und Authentizität wie ein Film zu liefern. Ein Beispiel ist die Beschreibung einer Landschaft wie des afrikanischen Regenwaldes, die kaum in der Lage ist, dem Leser einen Eindruck des Regenwalds zu verschaffen - im deutlichen Gegensatz zum Video. Hierbei muß jedoch zwischen Plastizität und Authentizität unterschieden werden. Während Beschreibungen in Büchern bei phantasiebegabten Personen durchaus in der Lage sind, sogar einen plastischeren Eindruck als ein Film hervorzurufen, leidet doch dar-

unter meist die Authentizität, d.h. die Übereinstimmung mit der Realität. Beides gemeinsam wird in hohen Maße nur vom Medium Video geliefert.

Die vordergründigere Plastizität führt auch dazu, daß Filme wesentlich leichter Emotionen wecken können als gedruckter Text. Manipulative Techniken, wie Werbung und Spendenaufrufe, bedienen sich daher wo immer möglich des Mediums Film.

Daß Videodaten trotz dieser Vorteile nicht früher und in stärkerem Maße Eingang in die Alltagswelt gefunden haben, läßt sich auf drei Ursachen zurückführen. Zum einen sind zu ihrer Wiedergabe aufwendige, oft sperrige und meist recht teure Wiedergabegeräte (z.B. Fernseher/Videorecorder, Computer) nötig. Zweitens sind die zu speichernden Datenmengen bei Videodaten sehr groß, was ebenfalls Kostennachteile nach sich zieht. Drittens ist das Nachschlagen und gezielte Suchen nach Informationen in Videodaten viel schwieriger als in gedruckten Medien. Das Problem der teuren und sperrigen Wiedergabegeräte und des hohen Speicheraufwandes verliert durch den technischen Fortschritt zunehmend an Bedeutung. Schon heute (1999) sind Laptops im Format eines Buches erhältlich. Daher muß das Augenmerk auf das Problem der Suche in Videodaten gerichtet werden.

In dieser Arbeit wird ein System vorgestellt, das genau dieses Problem adressiert, indem es eine schnelle und gezielte Suche in Videodaten ermöglicht. Ziel ist, einem Benutzer zu ermöglichen, in der Art eines Nachschlagewerkes zu einem gewünschten Thema passende Videosequenzen aufzufinden.

Kapitel 2

Übersicht über das System

In diesem Kapitel wird eine Übersicht über alle Komponenten des View4You-Systems, die Interaktion zwischen den Komponenten und die Integration zu einem Gesamtsystem gegeben.

Blockschaltbild

Eine schematische Übersicht über das realisierte System zeigt Abbildung 2.1.

Am Beginn der Verarbeitungskette steht die Aufnahme der Fernsehnachrichtensendungen. Da das View4You-Projekt von Anfang an auf die Erzeugung einer multilingualen Datenbank ausgerichtet war, wurde hierfür eine Satellitenempfangsanlage installiert, die schematisch im linken oberen Teil des Übersichtbildes dargestellt ist. Den eigentlichen Empfänger stellt eine 120 cm durchmessende Parabolantenne dar, die elektronisch auf verschiedene Satellitenpositionen ausgerichtet werden kann. Zur Aufnahme der 'tagesschau' am Empfangsort Karlsruhe ist ein derart großer Parabolspiegel unnötig; sie wurde im Hinblick auf die Aufnahme von Nachrichten in anderen Sprachen gewählt.

Das aufgenommene Signal wird von einem handelsüblichen Satelliten-Receiver decodiert und steht an seinem Ausgang als Videosignal zur Verfügung. In dieser Form ist die Bandbreite des Signals zur Verarbeitung zu hoch, so daß eine Kompression nach dem MPEG-Verfahren erfolgt[1]. MPEG ist ein asymmetrisches Kompressionsverfahren, das bei der Kodierung sehr aufwendig ist. Daher muß eine dedizierte MPEG-Coder-Hardware eingesetzt werden, um die Kompression in Echtzeit durchzuführen. Diese Hardware ist im View4You-System als PC-Einsteckkarte realisiert.

Die Ausgabe der komprimierten Nachrichtensendung erfolgt in zwei separaten Datenströmen: einem Videostrom und einem Audiostrom. Ersterer umfaßt - je nach

[1] Ausführliche Informationen über das MPEG-Verfahren finden sich im Anhang.

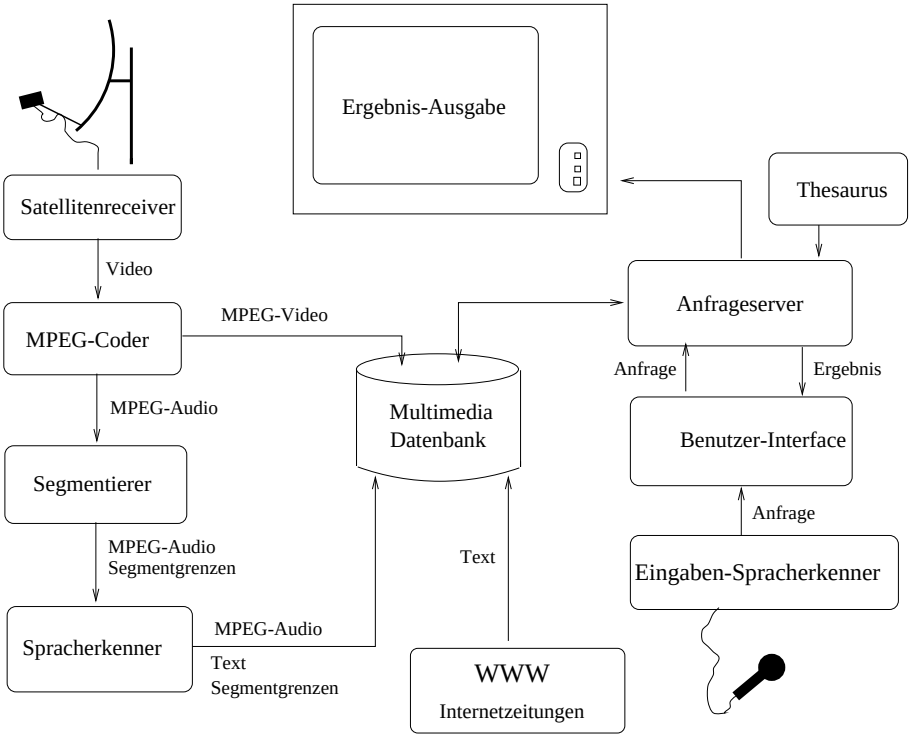


Abbildung 2.1: Aufbau des View4You-Systems

den gewählten Komprimierungseinstellungen - etwa 80% der Gesamtdatenmenge. Der Videostrom wird - mit dem Aufnahmedatum der Sendung versehen - ohne weitere Verarbeitung in der Datenbank abgelegt. Die eigentliche Indizierung der Aufnahmen basiert ausschließlich auf dem Audiostrom.

Die Audiodaten werden vom Segmentierer eingelesen, dessen Aufgabe es ist, die Grenzen der einzelnen Berichte innerhalb einer Nachrichtensendung zu identifizieren. Der Segmentierer ändert dabei die Audiodaten selber nicht, sondern erstellt lediglich eine Liste, in der die Zeitpunkte der gefundenen Segmentgrenzen aufgeführt sind. Es ist natürlich möglich, daß der Segmentierer Berichtgrenzen übersieht, oder daß an einer Stelle eine Berichtgrenze hypothetisiert wird, wo in Wirklichkeit keine vorliegt. Die Performanz des Segmentierers ist für das System kritisch. Werden zum Beispiel zu wenige Segmentgrenzen gefunden, enthält jedes gefundene Segment mehrere Berichte. Für den Benutzer bedeutet das einen ärgerlichen überflüssigen Zeitaufwand, weil für seine Anfrage irrelevante Daten gefunden und dargestellt werden. Wenn hingegen zu viele Segmentgrenzen gefunden werden, sind die Segmente kurz und der Benutzer wird mit abgeschnittenen Teil-Berichten konfrontiert.

Die nächste Systemkomponente stellt der Spracherkenner dar. Dieser erhält den

Audiostrom vom MPEG-Coder und die Segmentgrenzenliste vom Segmentierer. Für jedes vom Segmentierer gefundene Segment (d.h. für jeden Bericht getrennt), wird eine separate Spracherkennung durchgeführt. Es entstehen also genauso viele Hypothesen des Spracherkenners wie Segmente vom Segmentierer gefunden wurden; insbesondere entsteht bei idealer Segmentierung genau eine Hypothese pro Bericht. Die Ausgabe des Spracherkenners wird, zusammen mit der Segmentierung und dem Audiostrom selbst, an die Datenbank weitergeleitet.

Zusätzlich zu den aufgenommenen Fernsehsendungen werden noch weitere Wissensquellen automatisch in die Datenbank aufgenommen. Zu diesem Zweck fragt ein automatischer Prozess auf einer UNIX- Workstation jeden Morgen zwischen 4 Uhr und 5 Uhr mehrere Internet-Nachrichten-Server ab und speichert die erhaltenen Internetseiten in der Datenbank. Zu jeder so aufgenommenen Seite werden zusätzlich die Quelle (von welchem Server die Daten stammen) und das Aufnahmedatum gespeichert.

Die Schnittstelle der Datenbank zum Benutzer stellt der Anfrageserver dar. Als Eingabe in den Anfrageserver können beliebige Texte ohne jede syntaktische Einschränkung verwendet werden. Der Server bestimmt dann mit Techniken des *information retrieval* für alle Datensätze der Datenbank die Relevanz des Datensatzes in Bezug auf die Anfrage. Das Ergebnis einer Anfrage ist damit eine - nach absteigender Relevanz geordnete - Liste von Datensätzen der Datenbank. Optional könnte, zur Verbesserung der Findegenauigkeit, ein Thesaurus (siehe hierzu auch Seite 106) eingesetzt werden.

Der Benutzer des Systems sitzt vor einem Computerterminal, auf dem eine grafische Benutzeroberfläche implementiert ist. Die Benutzeroberfläche nimmt die Anfragen des Benutzers entgegen, leitet sie an den Anfrageserver weiter und stellt die Antwort des Anfrageservers in grafischer Form dar. Selektiert der Benutzer einen gefunden Datensatz, fragt die Benutzeroberfläche die entsprechenden Daten beim Anfrageserver an und stellt sie als Video bzw. als Text dar. Die Eingaben für die Benutzeroberfläche können dabei entweder über eine Tastatur oder über ein Spracherkennungssystem durch Spracheingabe erfolgen.

Technische Realisierung

Physikalischer Systemaufbau

Die technische Realisierung in Form einer Aufteilung auf Maschinen und Prozesse zeigt Bild 2.2. Insgesamt stellt das System ein Zusammenspiel von (mindestens) 8

Prozessen in fünf unterschiedlichen Programmiersprachen dar.

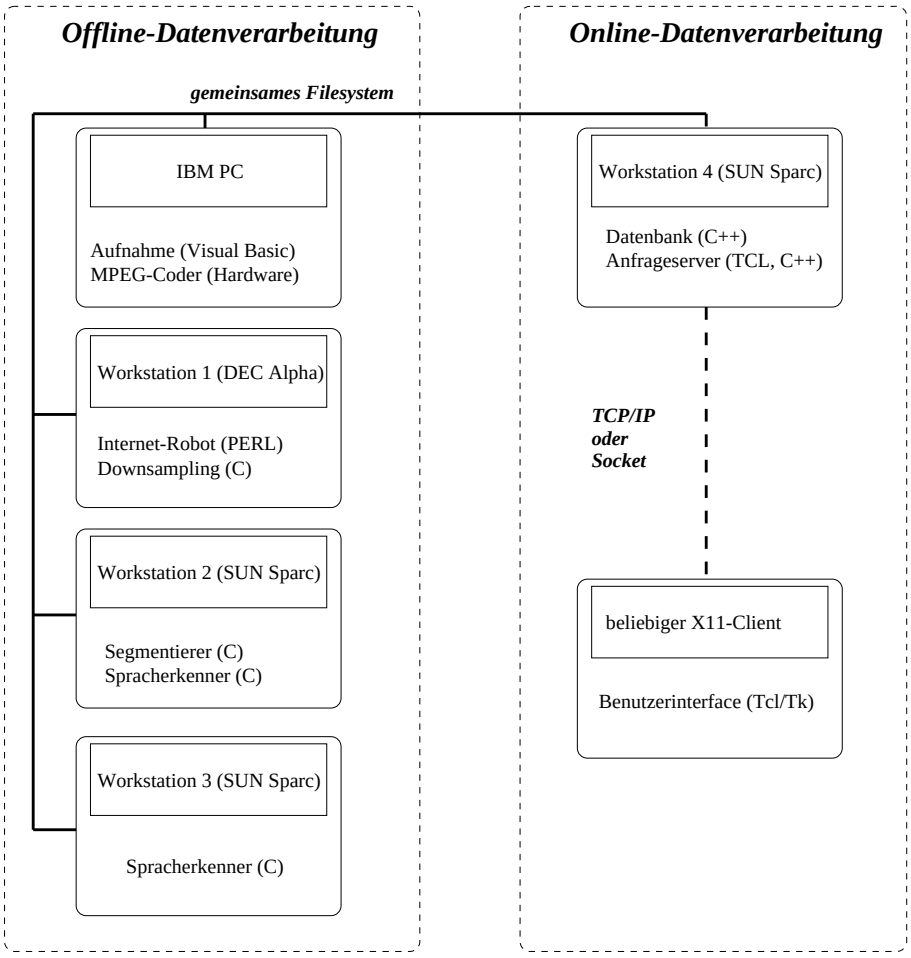


Abbildung 2.2: Realisierung des View4You-Systems

Die in Bild 2.2 gezeigte Anordnung ist die tatsächlich realisierte; sie stellt aber keineswegs die einzige mögliche Konfiguration dar. Insbesondere ist es möglich, alle Prozesse (außer der MPEG-Kompression, die dedizierte Hardware benötigt) auf nur einer Workstation laufen zu lassen.

Logischer Systemaufbau

Die *logische* Aufteilung kann in einen **online**- und einen **offline**-Teil erfolgen.

Für den **online**-Teil gilt

- Vorgänge für den Benutzer sichtbar

- Verarbeitung in Echtzeit muß angestrebt werden
- Varianz der Antwortzeiten von Anfrage zu Anfrage soll klein sein
- i.allg. keine Kontrolle über die eingesetzte Hardware (bei der Client-Maschine), d.h. es muß eine effiziente Implementierung angestrebt werden

Der online-Teil besteht im Wesentlichen aus dem Anfrageserver und dem Benutzerinterface. Die Verbindung zwischen beiden kann möglicherweise über eine langsame Modemleitung erfolgen, und der Client, auf dem das Benutzerinterface läuft, ist eventuell eine leistungsschwache Maschine. Diesen Umständen wird dadurch Rechnung getragen, daß die arbeitsintensiven Schritte soweit als möglich auf den Anfrageserver verlagert werden. Dieser kann bei entsprechender Dimensionierung auch mehrere Clients gleichzeitig bedienen. Die Hauptaufgaben des Anfrageservers sind die Bereitstellung von Daten aus der Datenbank und das eigentliche information retrieval, d.h. die Suche in der Datenbank. Dadurch sind die Aufgaben des Benutzerinterfaces auf das Verschicken der Anfrage sowie das Entgegennehmen und Darstellen der Antworten reduziert. Natürlich ist bei einem leistungsschwachen Rechner Spracherkennung als Eingabemedium nicht realistisch.

Der **offline**-Teil besteht hauptsächlich aus den Arbeitsschritten, die für das Auffüllen der Datenbank erforderlich sind. Charakteristisch für den offline-Teil ist

- Verarbeitung einer Sendung muß erst bei Aufnahme der nächsten Sendung abgeschlossen sein
- große Varianz der Verarbeitungszeiten der einzelnen Sendungen oder Segmente ist erlaubt
- 'Nacharbeiten' bereits verarbeiteter Daten (z.B. zur Qualitätsverbesserung) ist möglich
- keine für den Benutzer sichtbaren Vorgänge
- Einsatz leistungsstarker Hardware möglich

Im Fall des View4You-Systems bedeuten die ersten beiden Einschränkungen lediglich, daß alle Verarbeitungsschritte in jedem Fall innerhalb von 24 Stunden abgeschlossen sein müssen. Eventuell noch verbleibende Restzeit kann dazu verwendet werden, an früheren Tagen bereits verarbeitete Daten noch einmal zu sichten und zu verbessern, indem z.B. rechenzeitintensive Methoden angewendet werden, die bei der ersten Verarbeitung aus Zeitgründen noch nicht möglich waren. In der verbleibenden Restzeit kann das System auch versuchen, aufgrund der ihm vorliegenden Daten seine eigene Performanz autonom zu steigern. Mehrere Ansätze, wie eine solche 'intelligente' Systemverbesserung erreicht werden kann, werden in dieser Arbeit beschrieben.

Der rechenzeitintensivste Schritt im offline-Teil ist die Spracherkennung. Diese wurde routinemäßig auf zwei Rechner verteilt. Diese Parallelisierung ist trivial, da für jedes vom Segmentierer gefundene Segment ein Spracherkennerlauf durchgeführt wird und die Segmente voneinander unabhängig sind. Es mußte damit lediglich eine Liste von Segmenten erzeugt werden, von der alle (im Beispiel: alle beide) mit der Erkennung beschäftigten Maschinen den obersten Eintrag ablesen, von der Liste streichen und dann das entsprechende Segment bearbeiteten. Der Prozess endet, sobald die Liste leer ist. Der einzige locking-Mechanismus ist beim read/modify/write-Zyklus auf der Liste erforderlich. Experimentell wurde die Parallelisierung auf bis zu 8 Maschinen erfolgreich durchgeführt.

Kapitel 3

Daten

Zu Beginn der Arbeit mußte eine Auswahl der Sendungen, die indexiert werden sollten, getroffen werden. Maßgebliche Kriterien bei dieser Auswahl waren:

- die Verfügbarkeit (d.h. die Sendung sollte möglichst häufig, idealerweise täglich, ausgestrahlt werden)
- ein hoher Anteil an Sprache
- eine große Informationsdichte, damit die gefundenen Sendungsausschnitte für einen Benutzer möglichst interessant sind

Diese Kriterien werden von Nachrichtensendungen besonders gut erfüllt. Nachrichten werden (mindestens) täglich ausgestrahlt, weisen einen nur minimalen Prozentsatz an Musik- oder sonstigen sprachfreien Zeiten auf, und verfügen in aller Regel über eine große Informationsdichte.

3.1 Die Nachrichtensendung 'tagesschau'

Unter den im deutschen Fernsehen im Zeitraum der Erstellung dieser Arbeit (zweite Hälfte der neunziger Jahre) regelmäßig ausgestrahlten Nachrichtensendungen ist die **tagesschau** des öffentlich-rechtlichen Senderzusammenschlusses ARD diejenige mit dem größten Marktanteil (1996: 30,6%) und damit die bekannteste. Aus diesem Grund wurde diese Sendung als Grundlage der vorliegenden Arbeit ausgewählt. Andere, nicht betrachtete Nachrichtensendungen können mit den vorgestellten Prinzipien aber ebenfalls bearbeitet werden, ohne dass hierzu größere Modifikationen der Systemkomponenten erforderlich sind. Einzig der Segmentierer verfügt über eine Einstellung, die spezifisch auf die **tagesschau** zugeschnitten ist, nämlich über die erwartete Länge einer Sendung.

Geschichte

Die Nachrichtensendung 'tagesschau' wird seit Weihnachten 1952 von der Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland (ARD) produziert. Ab dem 1. Oktober 1956 wurde sie von montags bis samstags ausgestrahlt, am 3. September 1961 kam die Sonntagsausgabe hinzu. Die Tagesschau wird seitdem täglich um 20 Uhr gesendet. Das Erscheinungsbild der Sendung wurde den sich verändernden technischen Gegebenheiten und dem Zeitgeist kontinuierlich angepaßt. So wurde z.B. das Logo und die mit Musik unterlegte Anfangsmeldung zwischen 1952 und 1998 siebenmal geändert. Wichtige Einschnitte waren im März 1959 die Einführung des Sprechers, der die Nachrichten verliest, ab 1970 die Ausstrahlung in Farbe, und am 16.6.1976 das erstmalige Auftreten einer Frau (Dagmar Berghoff) als Sprecherin.
Da bis in die achtziger Jahre hinein das öffentlich-rechtliche Fernsehen keine Konkurrenz und die große Mehrzahl der Deutschen Zugriff auf genau drei Fernsehprogramme hatte, waren die Sprecher der tagesschau außerordentlich bekannte Figuren des öffentlichen Lebens, allen voran der fast 30 Jahre (vom 2.3.1959 bis zum 10.9.1987) 'amtierende' Chefsprecher Karl-Heinz Köpcke.
Die Sendungen werden stets von nur einem Sprecher moderiert. Um Monotonie zu vermeiden, werden in aufeinanderfolgenden Sendungen unterschiedliche Sprecher in mehr oder weniger zufälliger Abfolge eingesetzt. Im Zeitraum, in dem diese Arbeit entstand, waren insgesamt 8 Sprecher, vier Frauen und vier Männer, an der Entstehung der tagesschau beteiligt. Das Konzept ist erfolgreich; der Marktanteil 1996 betrug 8,44 Millionen Zuschauer oder 30,6% und die tagesschau war 1995 unter den 100 Sendungen mit den höchsten Einschaltquoten 62mal vertreten. Sie ist damit die erfolgreichste Sendung im deutschen Fernsehen überhaupt. Um diesen Erfolg auszubauen, wurden drei weitere Sendungen unter dem Label 'tagesschau' plaziert: ab dem 2.1.97 die 'tagesschau um fünf', ab dem 9.7.97 die 'tagesschau um drei', und schließlich - als bislang letzte neue tagesschau - ab 5.1.98 die 'tagesschau um zwölf'. Diese Sendungen folgen dem bewährten Konzept der 'tagesschau um acht'. In dieser Arbeit werden daher nur Sendungen der tagesschau um 20 Uhr betrachtet.

Aufbau

Die **tagesschau um acht** hat laut Programmheft eine Länge von 15 Minuten. Der tatsächliche Mittelwert liegt bei 15 Minuten und 49 Sekunden. 90% aller Sendungen weisen eine Länge zwischen knapp 15 und 17 Minuten auf. Die kürzeste der 60 transkribierten Sendungen in der View4You-Datenbank ist nur elf Minuten lang und wurde am 5.6.98 in der Halbzeitpause eines Fußball-Länderspiels ausgestrahlt. Die längste vom 3.6.98 bringt es auf knapp 20 Minuten; fast die Hälfte davon über das Zugunglück von Eschede an diesem Tag.

Es werden vier verschiedene Methoden eingesetzt, um eine Nachricht journalistisch aufzuarbeiten:

- Wortmeldung
- Nachricht im Film
- Berichte der Reporter
- Live-Berichte

Die **Wortmeldung** wird vom Sprecher verlesen und ist in der Regel 20 bis 30 Sekunden lang. Zur Illustration wird - vom Zuschauer aus gesehen - links vom Sprecher ein Foto, eine Karte oder eine Grafik eingeblendet, darüber steht eine kurze Schlagzeile, die das Thema der Meldung umreißt. Ihr zeitlicher Anteil an der 20.00-Uhr-Ausgabe beträgt normalerweise etwa ein Drittel. Alle Wortmeldungen einer Sendung werden vom gleichen Sprecher verlesen. Wortmeldungen haben die höchste akustische Qualität und weisen keinerlei Störgeräusche im Hintergrund auf.

Die **Nachricht im Film** ist üblicherweise etwa dreißig Sekunden lang. Zu einem Filmbeitrag vom Ort des Geschehens wird ein Kommentar von einem Off-Sprecher im Studio geliefert. Es können in einer Sendung unterschiedliche Off-Sprecher eingesetzt werden. Die akustische Qualität des Off-Sprechers selber ist sehr gut, da es sich um eine Studio-Aufnahme handelt. Allerdings werden häufig die Geräusche des Films dem Off-Sprecher unterlegt, so dass alle Arten von Hintergrundgeräuschen zu beobachten sind.

Die **Berichte der Reporter** sind die längsten Beiträge, mit einer Dauer von bis zu 2 Minuten. Im Schnitt werden vier bis fünf solche Berichte in jeder Sendung gezeigt. Sie werden via Überspielleitung oder Satellit von dem Korrespondenten an die Redaktion geliefert. Reporterberichte weisen stets mehr oder weniger laute Hintergrundgeräusche auf. Die akustische Qualität ist, bedingt durch das Überspielen, deutlich schlechter als bei Wortmeldungen oder Nachrichten im Film. Durch die Situation des Reporters am Rande des Geschehens enthält die Sprache zusätzlich noch Versprecher, Abbrüche und andere spontansprachliche Effekte, wie ungrammatikalische Konstrukte.

Live-Berichte sind in der tagesschau selten und werden nur eingesetzt, wenn die Aktualität es dringend erfordert; Live-Interviews werden in der tagesschau fast gar nicht geführt. Das ist ein großer Unterschied zu den amerikanischen News-Shows, die im ARPA-Hub4-Programm bearbeitet werden, wo etwa 30% der Sendezeit aus solchen Live-Berichten besteht. Die akustische Qualität von Live-Berichten liegt noch unter der von Reporterberichten, weil der Grad der Spontaneität höher ist. Die Qualität von Interviews ist in der Regel besser, weil der Interviewpartner meist in einem

auswärtigen Studio aufgenommen wird.

Den Beginn der Sendung macht stets die wichtigste Meldung des Tages. Gefolgt wird diese meist von einem Block mit Nachrichten aus dem Ausland, einem Block mit Nachrichten aus dem Inland, dem Sport, und dem (seit 1960) obligatorischen Wetterbericht. Im Schnitt werden pro Tagesschau 15 verschiedene Themen behandelt.

Als Nachrichtenquellen verwendet die **tagesschau** nach eigenen Angaben in erster Linie die Nachrichtenagenturen dpa, Adn/ddp, afp, ap, und Reuters. Ein weiterer Teil wird vom eigenen, umfassenden Korrespondentennetz geliefert. Die Textdaten der Agenturen, die von diesen mit einer gewissen zeitlichen Verzögerung der Allgemeinheit zur Verfügung gestellt werden, sollten sich daher gut zur Bestimmung eines statistischen Sprachmodells für die in der **tagesschau** verwendete Sprache eignen.

Bei der Abfolge der Beiträge wird auf Abwechslung zwischen Wortmeldungen und den anderen drei Berichtformen geachtet. In der ersten Hälfte der Sendung folgt auf eine Wortmeldung stets ein oder mehrere Reporterberichte bzw. Nachrichten im Film. In der zweiten Hälfte der Sendung werden dann oft zwei kurze Berichte hintereinander als Wortmeldungen verlesen, bevor wieder ein Film bzw. ein Reporterbericht gesendet wird. Zu Anfang und am Ende jeder Sendung wird eine Erkennungsmelodie gespielt. Exemplarisch wird im Anhang die zeitliche Abfolge der Beiträge in der Sendung vom 30.3.97 angegeben.
Von diesem Aufbau wird im Einzelfall zwar abgewichen, er stellt aber ein gutes Modell für die Mehrzahl der Sendungen dar. Der Segmentierer, der eine Sendung in Segmente zerlegt, macht Gebrauch von diesem Aufbauschema.

Die 'tagesschau' ist seit dem 1.8.1996 auch im Internet unter der URL http://www.tagesschau.de verfügbar. Auch der Aufbau der Internet-Seiten unterliegt stetigem Wandel; so wurde die Struktur des Servers in den ersten beiden Jahren des Betriebes zweimal geändert. Im Internet ist die ganze Tagesschau als Film mit Audio verfügbar. Als Konzession an die Bandbreitenprobleme im Internet wurde eine sehr starke Kompression gewählt, so dass die Qualität deutlich schlechter als die der empfangenen Sendungen ist. Zusätzlich liegen die Wortmeldungen und die Nachrichten im Film als Texte vor, von den Korrespondentenberichten gibt es hingegen keine Umschrift. Die Texte sind allerdings redaktionell gegenüber der gesendeten Form noch nach Themen geordnet und in der Reihenfolge geändert, so dass sie nicht direkt als Transkripte der Sendungen herangezogen werden können.

3.2 Multilinguale Daten

Die Überwindung der nationalen und der Sprachgrenzen ist im Zeitalter der globalen Vernetzung eine der größten und interessantesten Herausforderungen für die Informatik. Daher ist eines der Ziele des View4You-Projektes die Ergänzung der Videodatenbank durch Videos in anderen Sprachen als Deutsch.

Als Fremdsprache wurde das Serbokroatische gewählt.

3.3 Aufnahme der Daten

Die Aufnahme der Daten erfolgt mit handelsüblichen Satellitenempfangskomponenten und einer PC-gestützten MPEG Coder Hardware. Die verwendeten Satelliten zum Empfang der verschiedenen Sender sind in Tabelle 3.1 zusammengefaßt.

Sender	Satellit	Position	Frequenz	Polarisation	Tonunterträger
ARD	Astra 1B	19.2 O	11.494 GHz	H	7.02 MHz
HRT	Eutelsat II-F3	16 O	10.987 GHz	H	6.65 MHz
RTS	Eutelsat II-F2	10 O	11.598 GHz	H	6.60 MHz

Tabelle 3.1: Verwendete Satelliten

HRT steht für 'Hrvatska Radio Televizija' und strahlt die Nachrichtensendung 'Dnevnik' in kroatisch-serbischer Sprache aus. RTS bedeutet 'Radio Televizija Srbije'. Von dort kommt eine andere Nachrichtensendung in serbisch-kroatischer Sprache, die ebenfalls 'Dnevnik' (etwa 'täglich') heißt. Da die Zielsetzung der verfeindeten serbischen und kroatischen Führungseliten die möglichst weitgehende Trennung der beiden ehemaligen Landesteile Jugoslawiens ist, werden bestehende, eher dialektale Unterschiede zwischen serbisch-serbokroatisch und kroatisch-serbokroatisch in den öffentlichen Medien bestärkt.

Die niedrige Empfangsfeldstärke der Eutelsat-Satelliten macht die Verwendung der großen (1,20 Meter Durchmesser) Parabolantenne nötig. Die Qualität des empfangenen Signals ist bei guten Wetterbedingungen ausgezeichnet. Bei ungünstiger Witterung, wie Regen oder Nebel, ist allerdings eine merkliche Beeinträchtigung der Signalqualität zu beobachten, die sich natürlich auch negativ auf die Fehlerrate der Spracherkennung und Segmentierung auswirkt.

3.4 Umfang der Datenbasis

Für das Deutsche umfaßt die Datenbank insgesamt 212 aufgenommene Sendungen *tagesschau*, entsprechend 60 Stunden Sprache. 64 davon, entsprechend 16,5 h Sprache,

wurden transkribiert; der Rest wurde für das unüberwachte Training verwendet (vgl. S. 144 ff).

Eine Übersicht über die Datenbank enthält Tabelle 3.2.

Bezeichnung	Anzahl Sendungen	Stunden	Worte	Vokabular
gesammelt	212	60	?	?
Trainingsset	60	15.5	129889	18562
Testset	4	1	10653	3598
Mini-Trainingsset	2	0.5	3783	1691

Tabelle 3.2: Gesammelte Daten

In der Zeile 'gesammelt' ist die Gesamtgröße der Datenbank aufgeführt. Da für einen großen Teil der gesammelten Daten keine Transkription angefertigt wurde, ist eine Angabe der Zahl der Worte und der Vokabulargröße hier nicht möglich. Insgesamt wurden 64 Sendungen transkribiert, die in die beiden Stichproben 'Trainingsset' und 'Testset' aufgeteilt wurden. 'Mini-Trainingsset' ist eine Untermenge des 'Trainingsset'.

Kapitel 4

Maschinelle Spracherkennung

In diesem Abschnitt wird versucht, einen Überblick über den aktuellen Stand (1999) der maschinellen Spracherkennung und der verwendeten Algorithmen zu geben. Im Rahmen dieses Überblicks werden einige Begriffe eingeführt und erläutert, die in späteren Teilen der Arbeit als bekannt vorausgesetzt werden.

Maschinelle Spracherkennung ist ein weites Feld, das schon in den fünfziger Jahren bearbeitet worden ist. Es stellte sich jedoch rasch heraus, daß maschinelle Spracherkennung ein sehr viel schwierigeres Problemfeld darstellt als zunächst angenommen. Erst in den neunziger Jahren war die Technik weit genug fortgeschritten, um erste Applikationen in eingeschränkten Domänen (z.B. Menüsteuerungen über Telefon) zu ermöglichen. Seither hat das Interesse an Spracherkennung sehr stark zugenommen. Die Anzahl der weltweit tätigen Forscher auf diesem Gebiet geht in die tausende. Alleine in Deutschland wurden im Rahmen eines einzigen Verbundprojekts (VERBMOBIL) des Forschungsministeriums etwa 40 Wissenschaftler acht Jahre lang mit diesem Problem beschäftigt. Daher kann die hier vorliegende Zusammenfassung keinerlei Anspruch auf Vollständigkeit erheben. Eine gute Zusammenfassung über die wichtigsten Techniken geben [80] und [125].

Die maschinelle Spracherkennung hat die Aufgabe, zu einer gegebenen akustischen Observation O (z.B. die Aufnahme eines Sprachsignals) diejenige Wortfolge $\hat{W}$ zu finden, die die größte Wahrscheinlichkeit unter allen möglichen Wortfolgen W aufweist:

$$\hat{W} = \max_W p(W|O) = \max_W \frac{p(O|W)p(W)}{p(O)} \tag{4.1}$$

(4.1) bezeichnet man auch etwas emphatisch als *Grundgleichung der Spracherkennung*. Weil $p(O)$ nicht von W abhängt, gilt auch

$$\hat{W} = \max_{W} p(O|W)p(W) \tag{4.2}$$

Ein Spracherkenner benötigt also zwei Modelle, ein **akustisches Modell** für $p(O|W)$ und ein **Sprachmodell** für $p(W)$. Selbst wenn sehr effiziente Modelle für diese Verteilungen zur Verfügung stehen, ist eine direkte Auswertung von (4.2) durch Aufzählung der möglichen Wortfolgen und Maximumbestimmung nicht handhabbar. Daher muß in jedem Spracherkenner auch eine effiziente Suchmethode zur Lösung des Maximierungsproblems (4.2) implementiert sein.

Im Folgenden werden nacheinander Ansätze zur Modellierung von $p(W)$, von $p(O|W)$, und zur Lösung von (4.2) beschrieben.

4.1 Sprachmodelle

Die Aufgabe des Sprachmodells ist es, die Wahrscheinlichkeit für eine Wortfolge W abzuschätzen. Diese Wahrscheinlichkeit ist nicht abhängig von der akustischen Observation, also vom Sprachsignal, sondern stellt eine a-priori-Wahrscheinlichkeit für die Wortfolge W dar. Für $p(W)$ gilt natürlich die Normierungsbedingung

$$\sum_{\text{alle Wortfolgen W}} p(W) = 1 \tag{4.3}$$

Zur Modellierung der Wahrscheinlichkeitsfunktion $p(W)$ werden zwei prinzipiell unterschiedliche Verfahren angewendet: linguistisch motivierte und statistisch basierte. Die statistisch motivierten Verfahren bauen ein parametrisches Modell auf und versuchen, die Parameter des Modells direkt aus großen Textkorpora zu schätzen. Die linguistisch motivierten Verfahren versuchen hingegen, die der Sprache zugrundeliegende semantische und syntaktische Struktur zu analysieren und auf diese Weise eine Schätzung der Wahrscheinlichkeit einer gegebenen Wortfolge zu generieren.

Obwohl weitgehende Näherungen erforderlich sind, um ein parametrisches statistisches Sprachmodell trainieren zu können (d.h. bei gegebener Textmenge die Parameter robust schätzen zu können), haben sich bislang die statistischen Ansätze den linguistisch motivierten Ansätzen überlegen gezeigt. Daher werden im Folgenden nur die statistischen Modelle näher behandelt.

Die Wahrscheinlichkeit für eine Wortfolge $W = w_1 w_2 ... w_N$ läßt sich auch schreiben als

$$p(w_1 w_2 ... w_N) = \begin{array}{c} p(w_1) \\ *p(w_2|w_1) \\ *p(w_3|w_1 w_2) \\ *p(w_4|w_1 w_2 w_3) \\ ... * p(w_N|w_1 w_2 .. w_{N-1}) \end{array} \tag{4.4}$$

(4.5)

Die a-priori-Wahrscheinlichkeiten ($p(w_x)$) lassen sich sehr gut auf Textkorpora schätzen, weil für jedes Wort ausreichend Trainingsdaten verfügbar sind, um sie zu ermitteln. Sie werden auch als *Monogramme* bezeichnet. Schon der zweite Term in (4.4) ist in der Regel etwas problematisch: bei angenommenen 100.000 (10^5) verschiedene Wörtern tritt eine bestimmte Wortfolge $w_1 w_2$ in einem laufenden Text im Mittel nur alle 10^{10} Worte auf. Reale Texte sind in Größenordnungen bis etwa 10^9 Wörter verfügbar, so daß eine realistische Schätzung von $p(w_2|w_1)$ für seltene Wörter w_1, w_2 nicht mehr möglich ist. $p(w_2|w_1)$ bezeichnet man auch als *Bigramm*.

Für häufig auftretende Worte - das Pronomen 'der' tritt z.B. in Zeitungstexten mit einer relativen Häufigkeit von 3,96% auf - sind auch der dritte, vierte und fünfte Term (sogenannte *Trigramme*, *4-gramme* und *5-gramme*) in (4.4) noch gut zu schätzen. Bei den höheren Termen ist in der Regel selbst für häufigere Worte keine vernünftige Schätzung mehr möglich.

Daher wird als Näherung für die höheren Terme von (4.4) meist ein Modell niedrigerer Ordnung, häufig ein Trigramm, verwendet:

$$p(w_N|w_1 w_2 .. w_{N-2} w_{N-1}) \approx p(w_N|w_{N-2} w_{N-1}) \tag{4.6}$$

(4.6) läßt längerreichweitige Abhängigkeiten außer acht. Das wirkt sich im Deutschen z.B. bei Verben aus, deren Affix ans Satzende gestellt wird:

Erdbeeren lösen bei Dieter eine Allergie aus.

Ein deutscher Muttersprachler kann den Satz 'Erdbeeren lösen bei Dieter eine Allergie' leicht mit 'aus' ergänzen. Ein Trigrammsprachmodell hat zur Vorhersage des letzten Wortes nur die beiden vorhergehenden Worte 'eine Allergie' zur Verfügung. Daraus läßt sich 'aus' nicht mit derselben Sicherheit vorhersagen wie bei Kenntnis der gesamten Historie. Dennoch stellen N-gramme die derzeit beste bekannte Lösung des Problems der Modellierung von $p(W)$ dar.

Backing Off

Ein Trigrammsprachmodell für ein gegebenes Vokabular V besteht im Prinzip aus einer großen Tabelle, die für alle möglichen Kombinationen dreier Wörter des Voka-

bulars w_1, w_2, w_3 jeweils einen Eintrag aufweist. Dieser Eintrag stellt die Trigrammwahrscheinlichkeit $p(w_3|w_1w_2)$ dar. Die entsprechenden Wahrscheinlichkeiten können durch Abzählen auf einem großen Text ermittelt werden. Bezeichnet man die Anzahl der im Text gefundenen Wortfolgen $w_1w_2w_3$ mit $N(w_1w_2w_3)$, ist die maximum-likelihood-Lösung für die gewünschten Wahrscheinlichkeiten einfach zu erhalten:

$$p(w_3|w_1w_2) = \frac{N(w_1w_2w_3)}{\sum\limits_{w_i \in V} N(w_1w_2w_i)} = \frac{N(w_1w_2w_3)}{N(w_1w_2)} \tag{4.7}$$

Der Nachteil der maximum-likelihood-Lösung (4.7) ist, daß für alle im Trainingstext ungesehenen Wortfolgen $w_aw_bw_c$ die Wahrscheinlichkeit zu Null geschätzt wird. Dies schließt aus, daß die Grundgleichung der Spracherkennung (4.1) durch diese Wortfolge $w_aw_bw_c$ maximiert wird, und führt dazu, daß ein Spracherkenner mit einem solchen Sprachmodell ausschließlich Hypothesen erzeugen kann, bei denen jedes einzelne Worttripel im Trainingstext vorgekommen ist. Um auch ungesehenen Wortfolgen eine von Null verschiedene Wahrscheinlichkeit zuweisen zu können, wird daher die sogenannte backoff-Technik (engl.: to back off - zurückfallen) eingesetzt. Dabei wird im Prinzip für ein ungesehenes Trigramm $p(w_3|w_1w_2)$ das Bigramm $p(w_3|w_2)$ verwendet. Wurde auch die Wortfolge w_2w_3 nicht beobachtet, fällt man noch eine Stufe weiter auf das Monogramm $p(w_3)$ zurück.
Verwendet man jedoch für die gesehenen Worttripel $w_1w_2w_3$ die maximum-likelihood-Lösung (4.7) und für die ungesehenen Worttripel ein backoff auf das Bigramm $p(w_3|w_2)$, wird die Normierungsbedingung

$$\sum_{w_i \in V} p(w_i|w_1w_2) = 1 \tag{4.8}$$

verletzt, weil sich nach der Definition (4.7) bereits die Summe der Trigrammwahrscheinlichkeiten der *gesehenen* Worttripel zu eins summiert. Ein Teil der 'Wahrscheinlichkeitsmasse' $p(w_3|w_1w_2)$ muß daher bei der Berechnung der Wahrscheinlichkeiten der gesehenen Worttripel weggenommen werden. Dieser Teil stellt dann die Summe der Trigrammwahrscheinlichkeiten der ungesehenen Worttripel dar. Für das Wegnehmen (engl. discounting) von der Wahrscheinlichkeitsmasse hat sich das sog. *absolute discounting* in der Praxis am besten bewährt. Dabei wird von jeder Anzahl von Beobachtungen einer Wortfolge derselbe Wert d subtrahiert. Gängige Werte für d liegen in der Größenordnung von $0,5$. Mit dieser Methode wird die Trigrammwahrscheinlichkeit für gesehene Worttripel $w_1w_2w_3$ wie folgt berechnet:

$$p(w_3|w_1w_2) = \frac{N(w_1w_2w_3) - d}{N(w_1w_2)} \tag{4.9}$$

Für *ungesehene* Worttripel ergibt sich mit der Hilfsgröße D

$$D(w_1w_2w_3) = \begin{cases} d & : \quad w_1w_2w_3 \quad \text{wurde beobachtet} \\ 0 & : \quad \text{sonst} \end{cases} \tag{4.10}$$

$$p(w_3|w_1w_2) = p(w_3|w_2) * (\sum_{w_i \in V} \frac{D(w_1w_2w_i)}{N(w_1w_2)}) \tag{4.11}$$

Bei dieser Methode hängt die gesamte Wahrscheinlichkeitsmasse für die ungesehenen Modelle von der Zahl der beobachteten Nachfolger des Wortpaares w_1w_2 ab. Hat w_1w_2 viele verschiedene Nachfolger, ist die Wahrscheinlichkeit für einen unbeobachteten Übergang größer als wenn nach w_1w_2 nur wenige verschiedene Nachfolger beobachtet wurden.

Das backing off von Bigrammen auf Monogramme funktioniert völlig analog wie das beschriebene Zurückfallen von Trigrammen auf Bigramme. Die Wahrscheinlichkeit für eine sehr seltene Wortsequenz kann in einem realen statistischen Trigrammsprachmodell durchaus durch doppeltes Zurückfallen bis auf die Monogrammwahrscheinlichkeiten approximiert werden.

Perplexität

Hat man zwei verschiedene Sprachmodelle LM_1 und LM_2, so stellt sich die Frage, welches besser ist, d.h. welches der Modelle die tatsächlichen Wahrscheinlichkeiten $p(W)$ exakter vorhersagt. Allerdings sind die tatsächlichen Wahrscheinlichkeiten $p(W)$ nicht bekannt, so daß ein direkter Vergleich anhand dieser Definition nicht möglich ist. Man wählt daher einen etwas anderen Weg: Um die Vorhersagen des Sprachmodells mit der Wirklichkeit zu vergleichen, berechnet man, welche Wahrscheinlichkeit das Sprachmodell einem gegebenen, langen Text zuordnet. Ist der Text repräsentativ für die Sprache oder für das Aufgabenfeld, in dem das Sprachmodell eingesetzt werden soll, sollte seine Wahrscheinlichkeit groß sein. Im direkten Vergleich zweier Sprachmodelle wird somit dasjenige als 'besser' eingeschätzt, das auf einem ungesehenen, repräsentativen Testkorpus die höhere Gesamtwahrscheinlichkeit des Korpus $p(Korpus)$ vorhersagt. Natürlich muß für jedes der Sprachmodelle die Normierungsbedingung (4.3) gelten, weil ansonsten die Wahrscheinlichkeit beliebig groß werden kann.

Anstelle des Vergleichs der Wahrscheinlichkeiten $p_1(W)$ und $p_2(W)$ kann auch eine mittlere Wortwahrscheinlichkeit $\bar{p}(w)$ berechnet werden. Nimmt man an, daß die Wahrscheinlichkeit für jedes Wort gleich und unabhängig von den Nachbarn ist, so muß gelten

$$p(W) = p(w_1w_2...w_N) = \prod_{i=1}^{N} \bar{p}(w) \tag{4.12}$$

mit

$$\bar{p}(w) = \sqrt[N]{p(W)} \tag{4.13}$$

Hat die mittlere Wortwahrscheinlichkeit $\bar{p}(w)$ beispielsweise den Wert $0,01$, so bedeutet das, daß im Mittel $\frac{1}{0,01} = 100$ gleichwahrscheinliche Worte (genauer: Wortformen) auf einen vorgegebenen Kontext folgen. Welche Worte das im einzelnen sind,

darüber gibt das Sprachmodell *keinerlei* Auskunft - es ist in der Lage, auf (im Mittel) 100 gleichwahrscheinliche Alternativen zu reduzieren, aber nicht weiter. Die Auswahl unter den 100 Alternativen muß durch eine weitere Wissensquelle, nämlich die akustischen Modelle, vorgenommen werden. Aus dem Gesagten geht hervor, daß die Aufgabe der akustischen Modelle umso schwerer wird, je kleiner die mittlere Wortwahrscheinlichkeit $\bar{p}(w)$ bzw. je größer ihr Kehrwert $\frac{1}{\bar{p}(w)}$ wird. Man bezeichnet den Kehrwert der mittleren Wortwahrscheinlichkeit als *Perplexität PP* des Sprachmodells:

$$PP = \frac{1}{\bar{p}(w)} = \frac{1}{\sqrt[N]{p(W)}} \tag{4.14}$$

Die kleinste mögliche Perplexität ist 1. In diesem Fall sagt das Sprachmodell alle Worte mit Sicherheit voraus (und es sind keine akustischen Modelle mehr notwendig). Die größtmögliche Perplexität eines Sprachmodells mit einer Vokabulargröße von N Worten beträgt N und wird bei einer Gleichverteilung aller Wortwahrscheinlichkeiten erreicht. In diesem Fall liefert das Sprachmodell keinerlei Informationen über die Nachfolgeworte, gegeben einen Wortkontext. Zu beachten ist, daß die *gemessene* Perplexität auf einem Testkorpus durchaus Werte größer als N ergeben kann.

Der Wert der Perplexität dient als Maßzahl für die Effizienz eines Sprachmodells oder für die Schwierigkeit einer Erkennungsaufgabe. Allerdings hängt die Perplexität vom verwendeten Testkorpus ab, so daß direkte Vergleiche nur auf gleichen Korpora mit demselben Vokabular zulässig sind. Es gibt eine Korrelation zwischen Perplexität und Wortfehlerrate; allerdings sind Verbesserungen der Perplexität durchaus nicht immer mit einer Verringerung der Wortfehlerrate gekoppelt.

Die Perplexität des View4You-Sprachmodells auf zwei ungesehenen Sendungen *tagesschau* ist 381.

Auf US-amerikanischen Nachrichtensendungen konnten demgegenüber Perplexitäten von 120 erreicht werden. Der Unterschied in der Perplexität ist auf den viel kleineren Textkorpus, sowie auf das ungünstigere Vokabularwachstum des Deutschen - aufgrund der Flexionen und der Kompositabildungen - zurückzuführen.

4.2 Akustische Modelle

Die Aufgabe der akustischen Modelle ist es, eine Approximation für den Term $p(O|W)$ aus (4.1) zu liefern.

Diese Aufgabe wird zweistufig gelöst. Zunächst werden die vom Analog-Digital-Wandler aufgenommenen digitalisierten Daten O in eine zur Klassifikation geeignetere und vom Umfang her reduzierte Darstellung X umgewandelt. Die Berechnung von $p(O|W)$ erfolgt dann auf den so vorverarbeiteten Daten X. Wenn im Folgenden auf die Observation O Bezug genommen wird, ist damit immer die vorverarbeitete Version X gemeint.

Vorverarbeitung

Die Sprachdaten werden vom Mikrofon aufgenommen und in ein elektrisches analoges Signal umgewandelt. Dieses Signal wird von einem Analog-Digital-Wandler (A/D-Wandler) in kurzen, festen Zeitabständen gemessen. Die Messwerte werden digital abgespeichert. Die Auflösung der Messung hängt dabei von der verwendeten Elektronik ab; üblich sind Auflösungen zwischen 8 und 16 Bit. Bei einer Auflösung von N bit entsteht ein Quantisierungsfehler von $2^{-(N+1)}$, so daß ein 8-Bit-A/D-Wandler einen Quantisierungsfehler von 2^{-9} oder etwa 0,2% aufweist. Üblicherweise wählt man die Auflösung des A/D-Wandlers so, daß der Quantisierungsfehler in derselben Größenordnung wie die Varianz der Messwerte des Wandlers bei konstantem Eingangssignal liegt.

Der zweite wichtige Parameter bei der Abtastung ist der zeitliche Abstand zwischen zwei Messungen. Nach dem Abtasttheorem ist es für ein auf die Grenzfrequenz f_d bandbegrenztes Signal erforderlich, mindestens mit dem Doppelten der Grenzfrequenz abzutasten, um das Signal wieder rekonstruieren zu können. Da menschliche Sprache im Bereich oberhalb von ca. 8 kHz nur noch eine sehr geringe Leistungsdichte aufweist, genügt eine Abtastrate von 16kHz zu ihrer Aufnahme. Liegen Aufnahmen über das öffentliche Telefonnetz vor, so sind diese (in Deutschland) auf den Frequenzbereich von 300-3400 Hz bandbegrenzt. Hier würde nach dem Abtasttheorem eine Abtastrate von 6,8 kHz genügen. In der Praxis wählt man eine Abtastrate von 8kHz, um den nicht beliebig steilen Abfall der Tiefpassfilter der Telefongesellschaft zu kompensieren.

Die direkt vom Mikrofonsignal digitalisierten Daten eignen sich noch nicht gut für eine Erkennung. Ein- und dieselbe Äußerung kann beispielsweise etwas lauter oder leiser aufgenommen worden sein. Der A/D-Wandler weist oft einen geräteabhängigen Offset auf, so daß der Mittelwert der Daten von Null verschieden ist. Weitere Gründe sind die relativ hohe Datenrate und die Redundanz in den abgetasteten Daten. Diese Redundanz erkennt man schon daran, daß es mit einfachen Verfahren möglich ist, ein digitalisiertes Sprachsignal verlustfrei auf etwa 50% seiner Größe zu komprimieren [91].

Die Vorverarbeitung wird meist als Kurzzeitanalyse realisiert, bei der alle t_{fs} Zeiteinheiten ein für die ganze Zeiteinheit repräsentativer Datenvektor aus dem abgetasteten Signal berechnet wird. Der Datenvektor wird dabei häufig als *frame* und die Zeit t_{fs} als *frameshift* bezeichnet. Für die Umrechnung eines kurzen Datenabschnittes in einen repräsentativen Datenvektor sind verschiedene Verfahren gebräuchlich, von denen sich in den letzten Jahren vor allem die spektralanalytisch motivierten Ansätze durchgesetzt haben. Hierbei sind im besonderen zwei Verfahren zu nennen:

- Melscale-Spektralanalyse
- Cepstralanalyse

Bei der **Melscale-Spektralanalyse** handelt es sich um das Leistungsspektrum des betrachteten Zeitabschnitts mit einer nachfolgenden Zusammenfassung und Mittelung mehrerer Frequenzbänder in jeweils einen Melscale-Parameter. Die Zusammenfassung erfolgt dabei nach gehörphysiologischen Regeln: Es werden in Bereichen, in denen das menschliche Ohr eine niedrigere Frequenzauflösung hat, mehr Frequenzen in einen Parameter zusammengefasst als in den Bereichen, wo das Ohr eine hohe Frequenzauflösung aufweist (und in denen auch die wichtigsten Informationen für die Spracherkennung codiert ist). Die genaue Parametrisierung dieser sog. Bark-Skala findet sich auf Seite 76.

Cepstralparameter gehen von der Modellannahme aus, dass der resonanzfähige Raum 'Vokaltrakt' als lineares System aufgefaßt werden kann, das von der Grundfrequenz der Stimmbänder (und ihren Harmonischen) angeregt wird. Damit läßt sich Sprache im Frequenzbereich als Multiplikation der komplexen Übertragungsfunktion mit der Fouriertransformierten der Anregung darstellen. Die Grundidee bei der Cepstralanalyse ist nun, dieses gemischte Signal durch Logarithmieren in zwei additive Komponenten aufzuspalten und dann invers fourierzutransformieren. Die inverse Fouriertransformierte des Logarithmus des Leistungsspektrums bezeichnet man als Cepstrum. Im Cepstrum findet sich die Grundfrequenz mit ihren Harmonischen als ein scharfer 'Peak' am oberen Ende wieder, während die Fourierrücktransformierte der komplexen Übertragungsfunktion - also die Impulsantwort des Vokaltraktes - den Rest der Cepstralparameter ausmacht. Meist verwendet man von den Cepstralparametern $c_0 ... c_N$ nur die ersten 12-16, so dass der durch die Grundfrequenz der Stimmbänder erzeugte Peak nicht in die Parameter für die Erkennung eingeht. Dies ist insofern erwünscht, als dass die Grundfrequenz vom Sprecher und von der Prosodie (Sprachmelodie), aber nicht vom gesprochenen Phonem abhängt und somit - in erster Näherung - keine relevante Information beinhaltet.

Der Spracherkenner des View4You-Systems verwendet 12 Cepstralparameter, die aus 30 Melscale-Spektralparametern durch Logarithmierung und inverse Fouriertransformation gewonnen werden.

Bild 4.1 auf Seite 47 gibt den Verlauf der Vorverarbeitung in schematisierter Form wieder.

Lineare Diskriminanzanalyse (LDA)

Ein wichtiges Teilproblem der Mustererkennung ist die Gewinnung relevanter Merkmale. Die Verarbeitungsstufe, die dies zur Aufgabe hat, bekommt zu jedem (diskreten) Zeitpunkt t_k von der Vorverarbeitung einen Datenvektor $\vec{x}_k$ (z.B. Melscale-Spektralparameter oder Cepstralparameter) geliefert. Ihre Ausgabe ist wiederum ein Datenvektor, der zur Klassifikation besser geeignet ist als die direkte Ausgabe der Vorverarbeitung. Oft weist der neue Datenvektor eine geringere Dimensionalität auf als der alte, was die Aufgabe des Klassifikators erleichtert.

Am einfachsten ist es, die relevanten Komponenten aus dem Eingangsvektor auszuwählen und die anderen Komponenten zu unterdrücken. So könnten bei den Melscale-Parametern z.B. die Dimensionen 0-7, 11, 15 und 16 extrahiert und als neuer, in der Dimension reduzierter Eingabevektor verwendet werden. Die Leistungsfähigkeit dieses Verfahrens ist jedoch stark begrenzt. Besser ist es, den Eingangsvektor durch eine geeignet gewählte Transformation umzuformen.

Die gebräuchlichste Transformation in der Spracherkennung ist eine lineare Transformation, die sog. Lineare Diskriminanzanalyse. Diese geht von einer Lernstichprobe aus, für die jedem aus der Vorverarbeitung stammenden Datenvektor bereits die korrekte Klasse ω_i zugeordnet ist. Damit läßt sich für jede Klasse ω_i aus den Datenvektoren $\vec{x_k}$ und den Klassenmittelwerten $\vec{\mu_i}$ die **Klassen-Kovarianzmatrix** W_i berechnen:

$$W_i = \frac{1}{N(\omega_i)} \sum_{k,\vec{x}_k \in \omega_i} (\vec{x_k} - \vec{\mu_i})(\vec{x_k} - \vec{\mu_i})^T \tag{4.15}$$

Die über alle Klassen gemittelte Klassen-Kovarianzmatrix bezeichnet man auch als 'within-class-scatter' W:

$$W = \sum_i p(i) W_i \tag{4.16}$$

wobei $p(i)$ die a-priori-Wahrscheinlichkeit für Klasse i ist. Analog dazu läßt sich mit dem Mittelwert aller Daten $\vec{\mu}$ eine **totale Kovarianzmatrix** T definieren:

$$T = \frac{1}{N} \sum_k^N (\vec{x_k} - \vec{\mu})(\vec{x_k} - \vec{\mu})^T \tag{4.17}$$

Die lineare Diskriminanzanalyse sucht nun diejenige Transformationsmatrix A vom Eingangsraum x in den Merkmalsraum y,

$$\vec{y} = A\vec{x} \tag{4.18}$$

die das Kriterium

$$J = \frac{|T_y|}{|W_y|} \tag{4.19}$$

maximiert. Dies bedeutet anschaulich, dass bei konstanter totaler Kovarianzmatrix T die mittlere Klassen-Kovarianz, d.h. die Ausdehnung der einzelnen Klassen im Merkmalsraum, minimiert wird. Damit wird die Klassentrennung verbessert und die Klassifikationsaufgabe erleichtert. Die Lösung kann so gewählt werden, dass die totale Kovarianzmatrix T_y im Merkmalsraum y diagonal ist. Diese Dekorrelierung macht es möglich, in der akustischen Modellierung auf volle Kovarianzmatrizen zugunsten von rein diagonalen Kovarianzen zu verzichten. Allerdings gilt die Dekorrelierung nur im Mittel aller Klassen ω_i. Einzelne Klassen können durchaus - und haben in der Regel auch - nichtverschwindende Nichtdiagonalelemente ihrer Klassen-Kovarianzmatrix

W_i. Neuere Ansätze in der Spracherkennung [32] haben zum Ziel, diese verbleibende klassenabhängige Korrelation zu minimieren.

Eine Herleitung der Formeln zur Berechnung der Transformationsmatrix A findet sich z.B. in [120].

Vokaltraktlängennormierung (VTLN)

Bekanntlich haben Männer in der Regel eine tiefere Stimme als Frauen. Der Grund hierfür liegt zum einen in der niedrigeren Anregungsfrequenz der Stimmbänder, zum anderen in ihrer im Durchschnitt größeren Körpergröße, die auch zu einem größeren - und damit längeren - Hals-Rachenraum (Vokaltrakt) führt. Die Resonanzfrequenz eines (offenen) Rohres der Länge l beträgt

$$f_r = \frac{c}{4l} \tag{4.20}$$

Nimmt man die Schallgeschwindigkeit c als konstant an, folgt eine invers proportionale Beziehung zwischen der Länge des Vokaltraktes und seiner Resonanzfrequenz (der ersten Formante). Diese Variabilität zwischen Sprechern führt zu einer höheren Varianz der Sprachdaten bei gleicher gesprochener Wortfolge. Es ist das Ziel der Vokaltraktlängennormierung (VTLN, [48]), diese Variabilität in der Vorverarbeitung, d.h. vor der eigentlichen Klassifikation, auszugleichen.

Dazu wird zunächst anhand des Sprachsignals eine Schätzung der Vokaltraktlänge l des Sprechers durchgeführt. Der Schätzwert wird normiert, so dass ein Durchschnittssprecher die Vokaltraktlänge 1.0 aufweist. Dann werden die Koeffizienten des Leistungsspektrums (vgl. Bild 4.1) entsprechend der normierten Vokaltraktlänge verschoben. Hat ein Sprecher beispielsweise die Vokaltraktlänge $0,9$, so wird der 1000-Hz-Parameter des ursprünglichen Leistungsspektrums s_0 zum 900-Hz-Parameter des neuen Leistungsspektrums p', der 900-Hz-Parameter von s_0 zum 810-Hz-Parameter von p' und so weiter. Auf diese Weise entsteht ein neues Leistungsspektrum, das unabhängig von der Vokaltraktlänge des Sprechers ist. Die weiteren Schritte der Vorverarbeitung bauen dann auf dem modifizierten Leistungsspektrum auf.

Entscheidend für die VTLN ist die möglichst exakte Schätzung der Vokaltraktlänge l. Hierzu wird im View4You-System ein maximum-likelihood-Ansatz gewählt. Zunächst wird eine Hypothese des Gesagten ohne VTLN erzeugt und eine Zeitzuordnung zwischen jedem Datenframe und den Phonemen berechnet. Dann wird die VTLN für eine Matrix von möglichen Vokaltraktlängen l_i ($0,8 \leq l_i \leq 1,2$ in Schritten von $0,04$) durchgeführt und für jeden Wert von l_i die gesamte likelihood der Hypothese entlang der als konstant angenommenen Zeitzuordnung berechnet. Derjenige Wert von l_i, der die größte likelihood aufweist, wird als Schätzung für die normierte Vokaltraktlänge herangezogen [43].

Maximum Likelihood Linear Regression (MLLR)

Die Fehlerrate von sprecherabhängigen Spracherkennungssystemen ist - vergleichbare Mengen an Trainingsmaterial vorausgesetzt - nur etwa halb so groß wie die von sprecherunabhängigen Spracherkennern. Der Grund dafür ist die schlechtere Übereinstimmung der generischen Modelle des sprecherunabhängigen Erkenners mit der zu dekodierenden Sprache.
Es hat daher nicht an Versuchen gefehlt (vgl. z.B. [132], [115], [116], [127], [109]), anhand der vorliegenden Sprachprobe des bislang unbekannten Testsprechers die akustischen Modelle auf diesen speziellen Sprecher hin zu adaptieren, um durch dann besser passende Modelle eine niedrigere Fehlerrate zu erreichen. Die heute gebräuchlichste Methode wurde 1995 von Legetter und Woodland [74] unter dem Namen Maximum Likelihood Linear Regression (MLLR) vorgestellt. MLLR geht davon aus, dass die akustischen Modelle durch Normalverteilungen modelliert werden und transformiert in seiner einfachsten Form die Mittelwertvektoren der akustischen Modelle so, dass die log-likelihood L auf dem Adaptionssatz - bei fester Zeitzuordnung des Signals zu den akustischen Modellen - maximiert wird:

$$L = \sum_{t=1}^{T} (\vec{o_t} - (A\vec{\mu_s} + \vec{b}))^T \Sigma_s^{-1} (\vec{o_t} - (A\vec{\mu_s} + \vec{b})) \tag{4.21}$$

Dabei bezeichnet $\vec{o}$ den Observationsvektor zum Zeitpunkt t, s den diesem Zeitpunkt zugeordneten Zustand, und μ_s und Σ_s den Mittelwertvektor bzw. die Kovarianzmatrix des diesem Zeitpunkt zugeordneten Zustands. A ist die MLLR-Transformationsmatrix, $\vec{b}$ der MLLR-Verschiebungsvektor. Sie werden so gewählt, dass L maximal wird.

Zur Maximierungsaufgabe (4.21) läßt sich im Falle diagonaler Kovarianzmatrizen Σ_s eine geschlossene Lösung angeben [74].
Es ist möglich, anstelle einer globalen Transformationsmatrix A mehrere - z.B. eine pro Phonem - zu schätzen. Des Weiteren kann MLLR sowohl überwacht, also mit bekannter Referenz, als auch unüberwacht eingesetzt werden. In letzterem Fall ersetzt die aus der Hypothese des Erkenners resultierende Abfolge von Phonemzuständen $\hat{s}$ die wahre Folge s. An den entsprechenden Reestimierungsformeln ändert sich dadurch nichts. Reestimierungsformeln, die auch die Kovarianzmatrizen Σ transformieren, wurden von Gales und Woodland 1996 veröffentlicht [49].

Durch die Transformation $\vec{\mu} \rightarrow A\vec{\mu} + \vec{b}$ wird der Raum der Mittelwertvektoren gedreht, verzerrt und verschoben. Die Leistungsfähigkeit der MLLR gründet sich darauf, dass diese Transformation (im Falle einer einzigen Matrix A) auf *alle* Mittelwertvektoren angewendet wird, nicht nur auf solche, für die in der vorliegenden Sprachprobe ein Beispiel gesehen wurde. Auf diese Weise kann mit einer relativ kurzen Sprachprobe, die nur wenige Allophone des akustischen Modellraums abdeckt, dennoch eine

Adaption des kompletten Parametersatzes des Erkenners erzielt werden.

Im View4You-System kann durch eine unüberwachte MLLR-Adaption mit einer Adaptionsmatrix eine Verminderung der Fehlerrate von etwa 10% relativ erzielt werden (siehe hierzu Seite 85 ff). Dabei werden nur die Mittelwerte der Normalverteilungen adaptiert. Eine zusätzliche Varianzadaption brachte keine weitere Verringerung der Wortfehlerrate.

Akustische Modellierung

Die Aufgabe der akustischen Modellierung ist es, eine Schätzung für die Wahrscheinlichkeit $p(O|W) = p(\vec{o}_{1..T}|W)$ in (4.1) zu liefern. $\vec{o_t}$ ist dabei der Datenvektor des vorverarbeiteten Sprachsignals zum Zeitpunkt t.

Es wurden verschiedene Ansätze vorgeschlagen. Ist die mögliche Anzahl der Wortfolgen W klein, wie etwa bei einem Ziffernerkenner für isoliert gesprochene Ziffern, kann ein dediziertes Modell für jede mögliche Wortfolge verwendet werden. Für kontinuierlich gesprochene Sätze hingegen ist die Zahl der möglichen Wortfolgen für einen solchen Ansatz viel zu groß. Hier werden die Sätze in kleinere Untereinheiten zerlegt. Als mögliche Untereinheiten bieten sich Wörter, Silben, Phoneme und Subphoneme an. Dabei gilt, dass die Menge an verfügbarem Trainingsmaterial vom Wort zum Phonem zunimmt, da die Anzahl der unterschiedlichen Untereinheiten abnimmt - im Deutschen z.B. gibt es nur ca. 40 Phoneme, aber mindestens 300.000 Wörter.

Fast alle Spracherkennungssysteme verwenden daher kleine Untereinheiten, z.B. Phoneme. Um Koartikulationseffekte zu modellieren, werden Nachbarschaftsbeziehungen in die Modellierung integriert.

Bei der Verwendung von Untereinheiten wird zunächst eine Zerlegung der Verbundwahrscheinlichkeit $p(o_{1..T}|W)$ in ein Produkt von (voneinander als unabhängig angenommenen) Einzelwahrscheinlichkeiten vorgenommen:

$$p(o_{1..T}|W) \approx \prod_{t=0}^{T} p(o_t|s_t)p(s_t) \tag{4.22}$$

Dabei bezeichnet s_t dasjenige akustische Modell, das dem Beobachtungsvektor $\vec{o_t}$ zugeordnet wurde. Die Zeitzuordnung erfolgt meist durch dynamische Programmierung. Der Granularitätsgrad der akustischen Modelle kann sehr unterschiedlich sein. In dieser Betrachtung sei angenommen, jedes Phonem habe seinen eigenen Satz akustischer Modellparameter. Damit kann dem Term $p(s_t)$ die Bedeutung einer Phonemwahrscheinlichkeit zum Zeitpunkt t zugewiesen werden: Mit der Wahrscheinlichkeit $p(s_t)$ liegt zum Zeitpunkt t das Phonem s_t vor.

Es ist zu beachten, dass die Zerlegung von $p(O|W)$ in Gleichung (4.22) nur exakt ist, wenn die Observationen $\vec{o_t}$ statistisch unabhängig voneinander sind. Dies ist in der

Regel nicht der Fall. Dennoch hat die Zerlegung (4.22) so große praktische Vorteile, dass sie in fast allen Systemen explizit oder implizit durchgeführt wird.

Der Term $p(\vec{o}|s)$ definiert die unterste Ebene der akustischen Modellierung. Auf dieser Stufe wurde von zahlreichen Autoren eine Vielzahl von unterschiedlichen Ansätzen vorgeschlagen. So wurden beispielsweise Perzeptrone eingesetzt, wobei allerdings zu beachten ist, dass diese typischerweise $p(s|o)$ und nicht $p(o|s)$ approximieren, so dass noch entsprechende Umrechnungen vorzunehmen sind. Weitere, ebenfalls aus der Theorie der neuronalen Netze stammende Ansätze, die teilweise implizit die Unabhängigkeitsannahme aufweichen (TDNN, MS-TDNN), sind z.B. in [35] beschrieben.
In den meisten Systemen wird $p(\vec{o_t}|s)$ allerdings über Mixturen von Normalverteilungen modelliert. Dies hat praktische Vorteile, da die Normalverteilung über eine Vielzahl von Eigenschaften verfügt, die den Umgang mit ihr erleichtern. Zusätzlich dazu lässt sich der Einsatz auch theoretisch begründen, da sich durch eine genügende Anzahl von Normalverteilungen jede beliebige Verteilung beliebig genau approximieren läßt. Die Formel für die multivariate Normalverteilung im D-dimensionalen Raum lautet

$$N(\vec{o}|\vec{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} exp(-\frac{1}{2}(\vec{o}-\vec{\mu})^T\Sigma^{-1}(\vec{o}-\vec{\mu})) \tag{4.23}$$

Für eine Mixtur von M Normalverteilungen ergibt sich dann für $p(\vec{o_t}|s)$

$$p(\vec{o_t}|s) = \sum_{n=1}^{M} p(n)N(\vec{o_t}|\vec{\mu_n},\Sigma_n) \tag{4.24}$$

Die a-priori-Wahrscheinlichkeiten $p(n)$ für die einzelnen Normalverteilungen bezeichnet man auch als Mixturgewichte.

Das parametrische Modell (4.24) erfordert die Schätzung der Mittelwertvektoren und Kovarianzmatrizen $\vec{\mu_n}$ und Σ_n sowie der Mixturgewichte $p(n)$. Die Theorie der sogenannten Hidden-Markov-Modelle (HMM) liefert einen Algorithmus, der die Schätzung dieser Parameter und zusätzlich die damit konsistente Schätzung des Terms $p(s_t)$ aus (4.22) erlaubt. Die Existenz dieses, von Baum und Kollegen 1967 vorgestellten Algorithmus [138] ist ein wichtiger Grund dafür, dass HMM-basierte Spracherkenner heute (1999) den weitaus größten Anteil an allen implementierten Systemen haben.

4.3 Hidden Markov Modelle (HMMs)

Ein (diskretes) Hidden-Markov-Modell (HMM) ist definiert durch eine Menge $S := S_1, S_2, ..., S_N$ von Zuständen, durch Übergangswahrscheinlichkeiten $A = \{a_{ij}\}$ zwischen diesen Zuständen, durch eine initiale Konfiguration π, ein Ausgabealphabet V

und eine Verteilung $B = b_{ik} = p(V_k|S_i)$, die die Wahrscheinlichkeit der Emission des Symbols V_k im Zustand S_i angibt. Zu jedem Zeitpunkt befindet sich das HMM in einem der Zustände S_q. Zum Startzeitpunkt ($t = 0$) gibt π_i die Wahrscheinlichkeit an, dass sich das HMM im Zustand S_i befindet, wobei natürlich $\sum_i^N \pi_i = 1$ gelten muß.

Zu jedem Zeitpunkt t_q emittiert das HMM ein Zeichen des Ausgabealphabets und geht von einem Zustand in den nächsten über. Dieser Übergang ist ein Zufallsprozess, der durch die Übergangswahrscheinlichkeiten a_{ik} beschrieben wird. Häufig ist die Wahrscheinlichkeit für den Übergang $i \to i$ nicht Null, so dass das HMM auch von einem Zustand in denselben Zustand wechseln, d.h. in einem Zustand verbleiben kann. Der durch A und π charakterisierte Prozess heißt Markovkette.

Vor jedem Zustandsübergang emittiert das HMM ein Zeichen des Ausgabealphabets. Diese Emission ist wiederum ein stochastischer Prozess, dem die Wahrscheinlichkeitsverteilung B_q zugrundeliegt. B_q hängt nur vom aktuellen Zustand des HMM ab, nicht jedoch von den vorher evtl. bereits emittierten Zeichen. Das HMM hat also, abgesehen von seinen Zuständen, kein 'Gedächtnis'.

Bild 4.2 auf Seite 48 illustriert diese Zusammenhänge am Beispiel eines sehr vereinfachten Wettermodells. Es gibt in diesem Modell zwei Luftdruckzustände, 'Hoch' und 'Tief', und zwei mögliche Arten von Wetter, nämlich 'Sonne' und 'Regen'. Herrscht gerade ein Hochdruckgebiet, ist die Wahrscheinlichkeit für Sonne 80%; bei einem Tiefdruckgebiet ist die Wahrscheinlichkeit für schönes Wetter hingegen nur 30%. Hochdruckgebiete halten sich oft lange stabil, was in der hohen Wahrscheinlichkeit (P(Hoch—Hoch) = 75%) zum Ausdruck kommt, dass auf einen Tag mit hohem Luftdruck wieder ein Tag mit hohem Luftdruck folgt. Die Chance, dass ein Tiefdruckgebiet durch ein Hoch ersetzt wird, liegt demgegenüber bei 50%.

Im Unterschied zur Markovkette ist beim HMM der aktuelle Zustand des Systems nicht von außen beobachtbar - daher die Bezeichnung 'Hidden'. Lediglich die emittierten Symbole sind nach außen hin sichtbar. Im Beispiel des Bilds 4.2 gesprochen, steht bei einem HMM kein Barometer zur Messung des Luftdrucks zur Verfügung, sondern lediglich eine Wetterbeobachtung (Sonne oder Regen). Anhand dieser Beobachtung alleine ist es jedoch, von Sonderfällen entarteter Verteilungen A, B abgesehen, nicht möglich, eindeutig auf die durchlaufene Zustandsfolge zurückzuschließen. Allerdings kann man, bei Kenntnis der Verteilungen A, B, π, die *wahrscheinlichste* durchlaufene Zustandsfolge berechnen. Der Algorithmus, mit dem dies effizient möglich ist, heißt *Viterbi*-Algorithmus.

Desweiteren kann man die Frage stellen, mit welcher Wahrscheinlichkeit sich das HMM zum Zeitpunkt t in Zustand S befindet (gegeben die ausgegebenen Symbole). Die Antwort auf diese Frage gibt der forward-backward-Algorithmus.

Der Viterbi-Algorithmus

Gegeben sei das HMM aus Bild 4.2 und die Wetterbeobachtung an drei aufeinanderfolgenden Tagen: (Regen, Regen, Regen). Die Frage ist nun, welche Luftdruckabfolge während dieser drei Tage herrschte. Am ersten Tag lag - festgelegt durch π - ein Hoch vor. Die möglichen Abfolgen sind damit (H,H,H), (H,H,T), (H,T,H) und (H,T,T). Welche der vier Zustandsfolgen ist die wahrscheinlichste? Mathematisch formuliert liegt hier ein Optimierungsproblem vor, bei dem bei gegebener Observationssequenz O und gegebenen HMM-Parametern λ diejenige Zustands-Sequenz Q gesucht wird, die die größte a-posteriori-Wahrscheinlichkeit $p(O, Q|\lambda)$ aufweist:

$$\hat{Q} = \max_{Q} p(O, Q|\lambda) \tag{4.25}$$

Im Beispiel von Bild 4.2 läßt sich (4.25) durch Aufzählung der vier Möglichkeiten und explizite Berechnung der vier Wahrscheinlichkeiten bestimmen. So ergeben sich die Wahrscheinlichkeiten für die möglichen Abfolgen zu

$$\begin{aligned}
p(H,H,H) &= 1.0 * 0,2 * 0,75 * 0,2 * 0,75 * 0,2 &= 0,0045 \\
p(H,H,T) &= 1.0 * 0,2 * 0,75 * 0,2 * 0,25 * 0,7 &= 0,00525 \\
p(H,T,H) &= 1.0 * 0,2 * 0,25 * 0,7 * 0,5 * 0,2 &= 0,0035 \\
p(H,T,T) &= 1.0 * 0,2 * 0,25 * 0,7 * 0,5 * 0,7 &= 0,01225
\end{aligned}$$

Die wahrscheinlichste durchlaufene Zustandssequenz ist also (H,T,T).

Die Berechnung nach der direkten Methode führt bei größeren HMMs mit mehr Zuständen S und längeren Observationssequenzen O zu einer kombinatorischen Explosion: der Aufwand zur Berechnung ist (wenn keine Nullelemente in A auftauchen) $\mathbf{O}(S^{|O|})$.
Ein Algorithmus der dynamischen Programmierung, der Viterbi-Algorithmus [139][135], löst das Problem mit Aufwand $\mathbf{O}(|S| * |O|)$.

Dazu wird zunächst eine Matrix erstellt, bei der die Observationen O_t auf der x-Achse und die Zustände S auf der y-Achse eingetragen werden. Die Wahrscheinlichkeiten π_s für den Startzustand werden in die erste Spalte eingetragen. Danach schreitet der Algorithmus spaltenweise, beginnend bei $t = 1$, fort.

Im ersten Teilschritt werden die Zustandswahrscheinlichkeiten mit der Wahrscheinlichkeit multipliziert, dass der betrachtete Zustand die zu diesem Zeitpunkt erfolgte Beobachtung emittiert. Im zweiten Teilschritt geht man zur Spalte $t+1$ über und betrachtet dort jeden Zustand (State) separat. Für jeden dieser Zustände j der Spalte $t + 1$ kann die Wahrscheinlichkeit, dass er - gegeben die Observationssequenz O und die Modellparameter A, B, π des HMM - erreicht wird, **und** der Zustand q

der Spalte t sein Vorgänger ist, berechnet werden. Dies geschieht, indem die Wahrscheinlichkeit des Vorgängerzustands $p(s_q)$, die im ersten Teilschritt berechnet wurde, mit der Übergangswahrscheinlichkeit a_{qj} in den betrachteten Zustand j und der Wahrscheinlichkeit, daß der Zustand j die zu diesem Zeitpunkt erfolgte Beobachtung emittiert, multipliziert wird. Von allen möglichen Vorgängern q aus Spalte t wird derjenige gewählt, der zur größten Wahrscheinlichkeit für Zustand j zum Zeitpunkt $t+1$ führt und die entsprechende Zelle j der Spalte $t+1$ wird mit dieser Wahrscheinlichkeit gefüllt. Es wird also eine Maximierung über alle Vorgängerkandidaten durchgeführt:

$$q = \max_i p(s_i(t)) * a_{ij} * p(o_i|k) \tag{4.26}$$

Am Ende ist die gesamte Matrix mit Werten gefüllt, die die Wahrscheinlichkeit des günstigsten Weges bis zu diesem Punkt in der Matrix quantifizieren.

$$\text{viterbi}_{ik} = p_{\text{bester Weg}}(o_1..o_i, S(t=i) = S_k|\lambda) \tag{4.27}$$

In Bild 4.3 auf Seite 48 ist eine solche gefüllte Matrix dargestellt. Zum Zeitpunkt $t = 1$ ist $0,035$ die Wahrscheinlichkeit des besten Pfades in den Zustand 'Tief'. Es gibt keinen Weg durch die Matrix, der diesen Zustand mit einer größeren Wahrscheinlichkeit erreicht. Die größte Wahrscheinlichkeit in der letzten Spalte gibt den wahrscheinlichsten Zustand nach dem Durchlaufen der gesamten Matrix, d.h. der gesamten Observationssequenz, an.

Im zweiten Schritt des Viterbi-Algorithmus, dem sog. backtracking, wird ausgehend von diesem letzten Zustand rückwärts nach dessen wahrscheinlichsten Vorgänger gesucht, von diesem ausgehend wieder rückwärts nach dessen Vorgänger und so fort bis zum Anfang. Die so gefundene Zustandskette S ist diejenige mit der größten Wahrscheinlichkeit $p(O, S|\lambda)$.

Die Prozedur (ohne das Backtracking) ist in Bild 4.3 veranschaulicht. Man erkennt, dass der wahrscheinlichste Endzustand 'Tief' ist. Die Pfeile symbolisieren den jeweils bestmöglichen Vorgänger im Sinne des Maximierungsschrittes (4.26). Wüßte man beispielsweise zusätzlich, dass der Luftdruck am dritten Tag 'Hoch' war, so kann man die beste Zustandssequenz durch Rückwärtslaufen entlang der Pfeile, ausgehend vom Zustand 'Hoch' zum Zeitpunkt $t = 3$, zu (H,H,H) bestimmen.

Der forward-Algorithmus

Der forward-Algorithmus ähnelt in seiner Funktionsweise dem Viterbi-Algorithmus. Auch hier wird eine Matrix aufgebaut, wobei die Zeitschritte und Observationen auf der x-Achse und die Zustände des HMM auf der y-Achse liegen. Die einzelnen Zellen der Matrix bezeichnet man mit $\alpha_{i,k}$. Die Initialisierung der ersten Spalte erfolgt wie beim Viterbi-Algorithmus mit dem Produkt aus der initialen Zustandswahrscheinlichkeit π_i und der Emissionswahrscheinlichkeit, dass der Zustand i das Symbol o_0

ausgegeben hat. Der Algorithmus schreitet dann, genau wie der Viterbi-Algorithmus, spaltenweise fort. Der wesentliche Unterschied liegt im Maximierungsschritt (4.26), der beim forward-Algorithmus durch eine *Summation* über alle möglichen Vorgänger ersetzt wird:

$$\alpha_{i+1,k} = \left(\sum_{q=1}^{N} \alpha_{i,q} a_{qk} \right) p(o_i|k) \tag{4.28}$$

Die Interpretation der Wahrscheinlichkeiten $\alpha_{i,k}$ in der resultierenden forward-Matrix ist daher

$$\alpha_{i,k} = p(o_1 ... o_i, S(t = i) = S_k|\lambda) \tag{4.29}$$

also die Wahrscheinlichkeit, zum Zeitpunkt i im Zustand k zu sein.

Summiert man die $\alpha_{T,k}$ zum Zeitpunkt des letzten Zeitabschnitts, erhält man die Gesamtwahrscheinlichkeit $p(o_1..o_T|\lambda) = \sum_k p(o_1..o_T, S(t = T) = S_k|\lambda)$, mit der das HMM die Observation $O = o_1..o_T$ ausgibt:

$$P(O|\lambda) = \sum_{q=1}^{N} \alpha_{T,q} \tag{4.30}$$

Der forward-backward-Algorithmus

Eine wichtige Frage im Zusammenhang mit HMMs ist die nach der Wahrscheinlichkeit eines bestimmten Zustands k zu einer gegebenen Zeit $t = i$. Diese Wahrscheinlichkeit wird γ_{ik} genannt:

$$\gamma_{ik} = p(S(t = i) = S_k)|O, \lambda) \tag{4.31}$$

Durch Umformung läßt sich das schreiben als

$$\gamma_{ik} = p(S(t = i) = S_k)|\lambda) = \frac{p(O, S(t = i) = S_k)|\lambda)}{p(O|\lambda)} \tag{4.32}$$

Der Nenner in (4.32) läßt sich mit dem oben beschriebenen forward-Algorithmus durch Addition der α_{Tk} berechnen. Um den Zähler berechnen zu können, genügt der forward-Algorithmus alleine nur zum Zeitpunkt $t = T$, d.h. in der letzten Spalte der Matrix. Der Grund dafür ist, dass die α_{ik} die Wahrscheinlichkeit des Zustands k zum Zeitpunkt i gegeben die Beobachtungen von $t = 0$ bis $t = i$ angeben; der Zählerterm von (4.32) bedeutet jedoch die Wahrscheinlichkeit des Zustands k zum Zeitpunkt i gegeben die gesamte Observationssequenz. Nur zum Zeitpunkt $t = T$ sind also die forward-Wahrscheinlichkeiten mit den $p(O, S(t = i) = S_k)|\lambda)$ aus (4.32) identisch. Für die übrigen Zeitpunkte muß der forward-Algorithmus entsprechend um einen Teil, der den Rest der Observationen in Betracht zieht, ergänzt werden. Diese Aufgabe wird vom backward-Algorithmus gelöst. Man kann den backward-Algorithmus als eine zeitliche Umkehrung des forward-Algorithmus auffassen. Man definiert die Variablen β_{ik} als

$$\beta_{ik} = p(o_{i+1}...o_T|S(t=i) = S_k, \lambda) \tag{4.33}$$

Analog zum Viterbi- und zum forward-Algorithmus wird die Zeitschritt-Zustandsmatrix aufgebaut. Alle Elemente der letzten Spalte der Matrix werden mit 1 initialisiert. Dann wird die Matrix spaltenweise rückwärts durchlaufen und die Matrixelemente werden mit $\beta_{t,k}$ wie folgt gefüllt:

$$\beta_{t,k} = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1,j} \tag{4.34}$$

Die Wahrscheinlichkeit, sich zum Zeitpunkt $t = 0$ im Startzustand zu befinden, gegeben den Rest der Observationen (also alle Observationen), ist genau gleich der Wahrscheinlichkeit, sich zum Zeitpunkt $t = T$ im Endzustand zu befinden, gegeben alle Observationen. Es gilt also

$$P(O|\lambda) = \sum_{q=0}^{N} \beta_{0,q} = \sum_{q=0}^{N} \alpha_{T,q} \tag{4.35}$$

Mit Hilfe der $\beta_{i,k}$ läßt sich nun die gesuchte Wahrscheinlichkeit (4.31) errechnen:

$$\gamma_{ik} = p(S(t=i) = S_k)|\lambda) = \frac{\alpha_{ik}\beta_{ik}}{p(O|\lambda)} \tag{4.36}$$

Man beachte, dass der vom Viterbi-Algorithmus bestimmte Zustand i zum Zeitpunkt t durchaus nicht der zu diesem Zeitpunkt wahrscheinlichste, also derjenige mit dem größten γ_{ti} sein muß. Diesen auf den ersten Blick paradoxen Sachverhalt kann man sich am Beispiel von Verkehrsströmen zwischen zwei Städten klar machen.

Zwischen Stadt A und Stadt B existiert eine gut ausgebaute Bundesstraße, sowie eine bis zur Hälfte fertiggestellte Autobahn. Ab dem Ende der Autobahn gibt es drei schlecht ausgebauten Nebenstraßen zum Zielort. In der Summe benutzen 60% der Autofahrer die Autobahn und dann - gleichverteilt - eine der Nebenstraßen, und 40% der Fahrer benutzen die Bundesstraße. Dann ist der wahrscheinlichste Pfad der über die Bundesstraße, denn jeder konkrete Pfad über die Autobahn und dann über genau eine der Nebenstraßen wird von nur 20% der Fahrer benutzt - der Verkehr verteilt sich ja auf dem nicht ausgebauten Teil über drei Alternativen. Hingegen ist die Wahrscheinlichkeit, dass sich kurz nach seiner Abfahrt ein Auto auf der Autobahn befindet, größer als die Wahrscheinlichkeit, dass es sich auf der Bundesstraße befindet - obwohl die Bundesstraße den wahrscheinlichsten Pfad darstellt.

Wort-HMMs und Suche

Weil die robuste Schätzung von Ganzwortmodellen[1] für selten gesehene Wörter ein außerordentlich schwieriges Problem ist, verwenden fast alle modernen Spracherken-

[1] siehe hierzu das Glossar

ner für große Vokabularien Phonemmodelle, die zu Wörtern aneinandergehängt werden. Das Wissen, aus welchen Phonemen - und damit, aus welchen Phonemmodellen - ein Wort aufgebaut ist, ist dabei in einem phonetischen Wörterbuch abgelegt. Jedes Phonem hat ein eigenes akustisches Modell. Ein Phonemmodell kann im Prinzip aus einem einzigen HMM-Zustand bestehen. In der Regel wird jedoch eine Modellierung durch drei Zustände gewählt, wobei der erste Zustand den Anlaut, der zweite Zustand die konstante Phase und der dritte Zustand den Auslaut des Phonems beschreiben. Das Phonem-HMM erlaubt neben den Übergängen zum nachfolgenden Zustand (vom ersten in den zweiten, vom zweiten in den dritten und vom dritten in den ersten des Nachfolge-Phonems) fast immer auch Übergänge von einem Zustand in sich selbst, um Variationen in der Sprechgeschwindigkeit ausgleichen zu können. Manchmal existieren auch Übergänge, die einen Zustand überspringen. Rückwärtssprünge (Loops) sind hingegen in der Regel nicht möglich; wenn von seltenen Spezialfällen (Stottern) abgesehen wird, ist das auch nicht zweckmäßig.

Bild 4.4 auf Seite 49 zeigt den Aufbau des Wortmodells für das Wort 'der' aus den Phonem-HMMs für seine drei Phoneme **d**, **e:** und **6**.

Um bei einer gegebenen akustischen Observation die wahrscheinlichste Wortfolge zu ermitteln, wird der Viterbi-Algorithmus (s. Seite 40) verwendet. Dazu werden die Wort-HMMs für alle Wörter des Wörterbuchs aufgebaut und Übergänge zwischen dem Wortende von jedem Wort und den Wortanfängen jedes Wortes eingefügt. Die wahrscheinlichste Zustandsfolge wird mit dem Viterbi-Algorithmus ermittelt. Die Wörter, die auf der wahrscheinlichsten Zustandsfolge liegen, stellen die Ausgabe des Erkenners dar. Zur Erhöhung der Effizienz dieses Suchvorganges sind zahlreiche Heuristiken eingesetzt worden. Einen guten Überblick gibt Wosczczyna in [22].

Kontextabhängigkeit

Eine Modellierung wie in Bild 4.4 dargestellt ist jedoch zu grob. Der erste Zustand des Phonems 'e:' wird immer gleich modelliert, unabhängig vom vorhergehenden Phonem. Es ist jedoch bekannt, dass die Koartikulation für verschiedene Kontexte unterschiedlich ist, so dass es zweckmäßig ist, ein Phonem abhängig von seinen Nachbarn zu modellieren. Dieser Gedanke führt zum kontextabhängigen Modell. Im einfachsten Fall hängt das Phonemmodell von seinem linken (oder rechten) Nachbarn ab. Bei - angenommenen - 50 Basisphonemen ergeben sich 50*50 = 2500 solche Biphone, d.h. 2500 unterschiedliche Modelle. Üblich ist eine Modellierung von Triphonen. Hierbei hängt das Modell sowohl vom linken als auch vom rechten Nachbarn ab. Bei 50 Basisphonemen gibt es 50*50*50=125000 unterschiedliche Triphone. Es können natürlich auch größere Kontexte in Betracht gezogen werden. Allerdings ist der phonetische Einfluß eines mehr als 2 Phoneme nachfolgenden Lautes auf das aktuelle Phonem nur noch außerordentlich klein, so dass in der Praxis maximal 5-Phone zum Einsatz kommen.

Das Problem der Schätzung von 125000 unterschiedlichen Phonemmodellen ist

mit den üblicherweise zur Verfügung stehenden Datenmengen nicht zu lösen. Aus diesem Grund werden die 125000 theoretisch möglichen Modelle zu einer gut trainierbaren Menge von etwa 1000-10000 'generalisierten Triphonen' geballt. Alle Triphone in einem so entstehenden Cluster haben dasselbe Modell. Es sind unterschiedliche Techniken zur Ballung von Triphonen bekannt. In der vorliegenden Arbeit wurde ein top-down Ballungsverfahren eingesetzt, das zunächst alle Polyphone in einem einzigen Modell vereint und dann sukzessive eine Aufspaltung nach der anderen vornimmt. Die Aufspaltungen erfolgen anhand von linguistisch motivierten Fragen, um die Generalisierungsfähigkeit auf ungesehene Triphone zu gewährleisten. Als Kriterium für die Selektion der jeweils besten Frage wird ein Entropiekriterium (vgl. [122]) herangezogen. Eine detaillierte Beschreibung der linguistisch motivierten Fragen findet sich im Anhang.

4.4 Bewertung von Spracherkennern

Um Spracherkenner untereinander vergleichbar zu machen, haben sich als international akzeptierte De-facto-Standards die Maße Wortakkuratheit (engl. word accuracy, WA) und Wortkorrektrate (engl. word correct rate, WC) etabliert. Um sie zu berechnen, wird mittels dynamischer Programmierung die minimale Editierdistanz zwischen der Hypothese des Spracherkenners und der Referenz berechnet. Die Hypothese kann dabei zusätzlich zu den Substitutionsfehlern (anstelle eines Referenzwortes wird fälschlicherweise ein anderes Wort erkannt) noch Einfügungen (es wird ein Wort zu viel hypothetisiert) und Löschungen (einem gesprochenes Wort steht kein Wort der Hypothese gegenüber) enthalten. Ein Beispiel möge das veranschaulichen.

Die Referenz laute 'dann treffen wir uns am Freitag', und die Hypothese des Spracherkenners sei 'dann treten wir am Freitag ein'. Die Zuordnung zwischen Hypothese und Referenz sieht dann wie folgt aus:

Referenz	Hypothese	Fehlerart
dann	dann	Korrekt
treffen	treten	Substitution
wir	wir	Korrekt
uns	*	Löschung
am	am	Korrekt
Freitag	Freitag	Korrekt
	ein	Einfügung

Tabelle 4.1: Zuordnung zwischen Hypothese und Referenz

Die Editierdistanz zwischen Hypothese und Referenz beträgt in diesem Beispiel 3 Operationen (eine Substitution, eine Löschung und eine Einfügung). Mit den Be-

zeichnungen N_S für die Zahl der Substitutionsfehler, N_L für die Zahl der Löschfehler, N_E für die Zahl der Einfügefehler, N_C für die Zahl der korrekt identifizierten Worte und N für die Zahl der Worte in der Referenz, so sind die Maße Wortkorrektrate WC und Wortakkuratheit WA wie folgt definiert:

$$WC = \frac{N_C}{N} = \frac{N - N_S - N_L}{N} \tag{4.37}$$

$$WA = \frac{N - N_S - N_L - N_E}{N} = \frac{N_C - N_E}{N} = WC - \frac{N_E}{N} \tag{4.38}$$

Häufig wird anstatt der Wortakkuratheit die Wortfehlerrate (word error rate, WER) angegeben. Sie ist einfach als

$$WER = 1 - WA = \frac{N_S + N_E + N_L}{N} \tag{4.39}$$

definiert. Man beachte, dass die Wortfehlerrate, bedingt durch Einfügefehler, über 100% liegen kann.

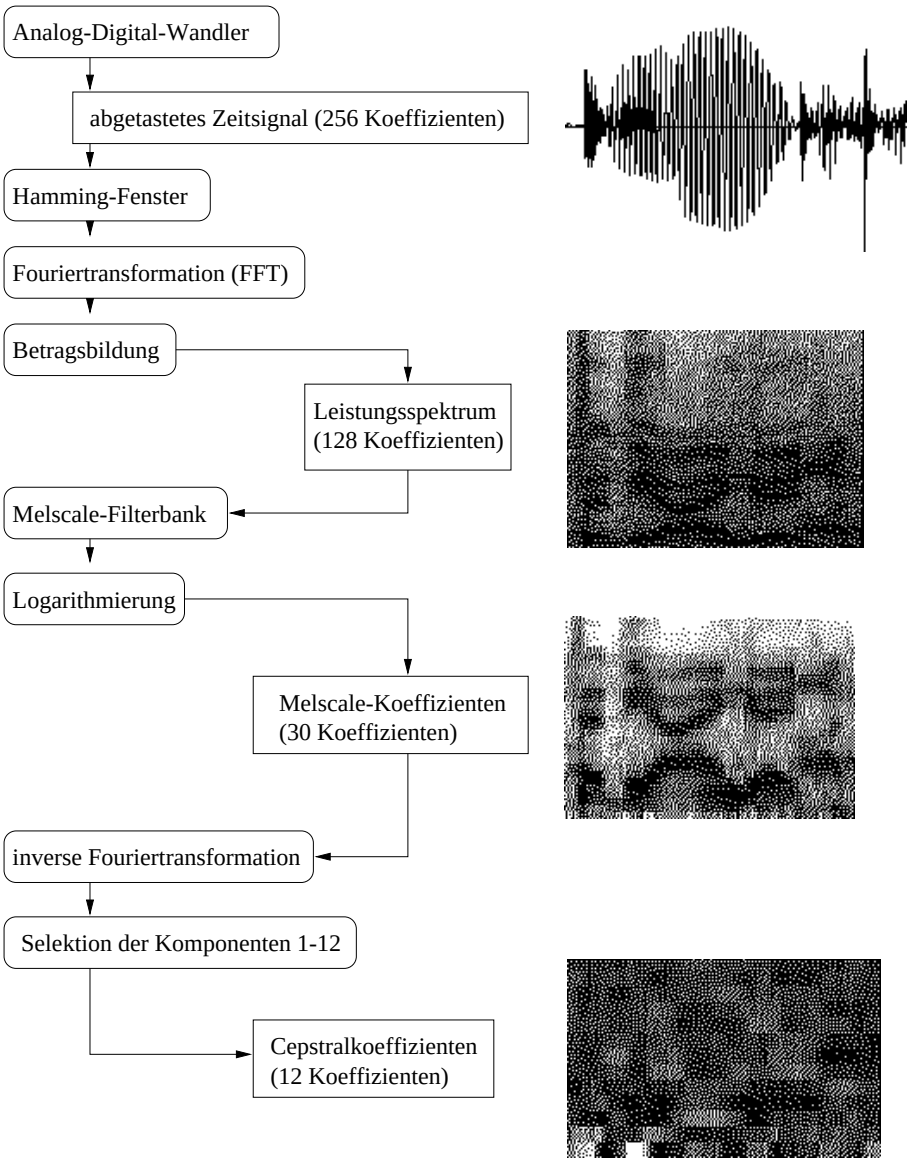


Abbildung 4.1: Schematische Funktionsweise der Vorverarbeitung

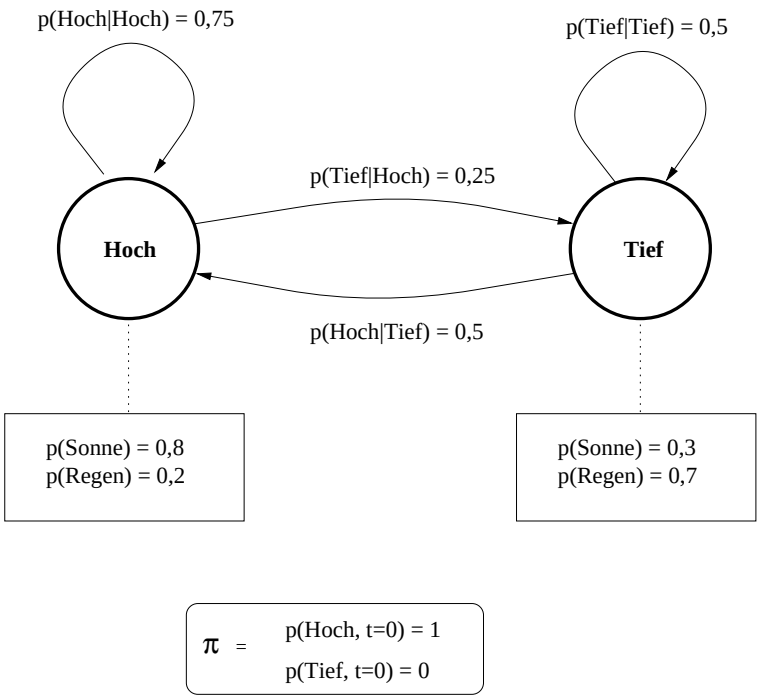


Abbildung 4.2: Ein Beispiel-HMM

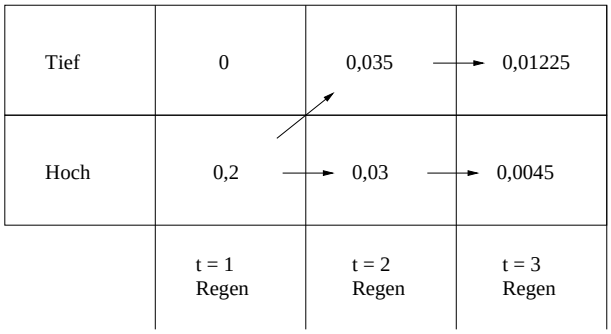


Abbildung 4.3: Viterbi-Matrix

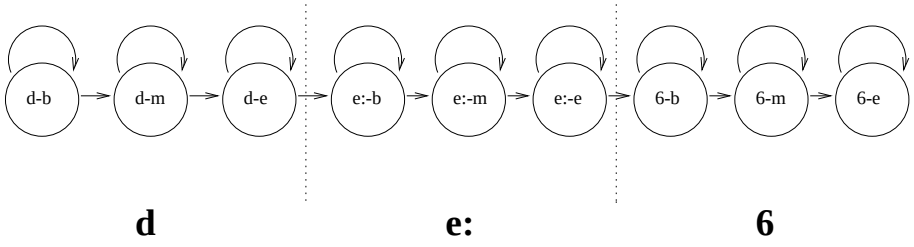


Abbildung 4.4: HMM des Wortes 'der'

Kapitel 5

Der Segmentierer des View4You-Systems

Einleitung

Wenn ein Benutzer das View4You-System nach einem bestimmten Thema befragt (z.B. 'was gibt es über den Nahen Osten?'), dann ist es nicht sehr hilfreich, als Antwort eine komplette Sendung zurückzuliefern. Diese müsste sich der Benutzer dann komplett ansehen, um den darin enthaltenen Beitrag über den Nahen Osten zu finden. Besser ist es, wenn jede Sendung bereits in Beiträge zerlegt ist und auf die Anfrage hin nur noch der passende ausgeschnittene Beitrag präsentiert wird.

Es ist die Aufgabe des Segmentierers, diese Zerlegung einer kompletten Sendung in ihre einzelnen Beiträge durchzuführen.

'Beitrag' ist ein semantisches Konzept. Ein Beitrag kann einfach darin bestehen, dass der Ansagesprecher eine Nachricht verliest; er kann auch aus dem Verlesen einer Nachricht, einem nachfolgenden Korrespondentenbericht vom Ort des Geschehens und einem darauffolgenden Interview zusammengesetzt sein. Um die Grenzen korrekt zu ziehen, ist eine *semantische* Analyse (im Hinblick auf das gerade behandelte Thema) der Sendung erforderlich. Sobald ein Themenwechsel stattfindet, muss eine Segmentgrenze eingefügt werden.

Die Realisierung der Segmentierung in Beiträge wird im View4You-System zweistufig durchgeführt:

1. Segmentierung der Sendung in akustisch homogene Segmente (Schnitte)
2. Zusammenfügen thematisch zusammengehörender, aneinandergrenzender Schnitte zu Beitrags-Segmenten

Im ersten Schritt werden Segmentgrenzen dort gesetzt, wo sich die **akustischen Charakteristika der Sendung ändern**, d.h. bei einem Sprecherwechsel oder beim Wechsel vom Ansagesprecher zu einem Korrespondenten.

Da häufig mehrere Nachrichten (zu unterschiedlichen Themen) hintereinander vom Ansagesprecher verlesen werden, wird eine Segmentgrenze auch gesetzt, wenn eine längere Pause innerhalb des Sprachbeitrags eines Sprechers stattfindet.

Die akustische Segmentierung führt systematisch zu einer Übersegmentierung: Ein Beitrag, der beispielsweise aus einem Ansagesprechertext gefolgt von einem Korrespondentenbericht besteht, wird durch sie in zwei Segmente zerlegt.

Abbildung 5.1 veranschaulicht nochmals den Unterschied zwischen dem erwünschten Ergebnis (obere Zeile, berichtorientierte Segmentierung) und dem Ergebnis des ersten Schrittes der Segmentierung, den akustisch motivierten Segmenten (mittlere Zeile). Die unterste Zeile zeigt das Ergebnis eines Segmentierungslaufes eines (fiktiven) akustischen Segmentierers.

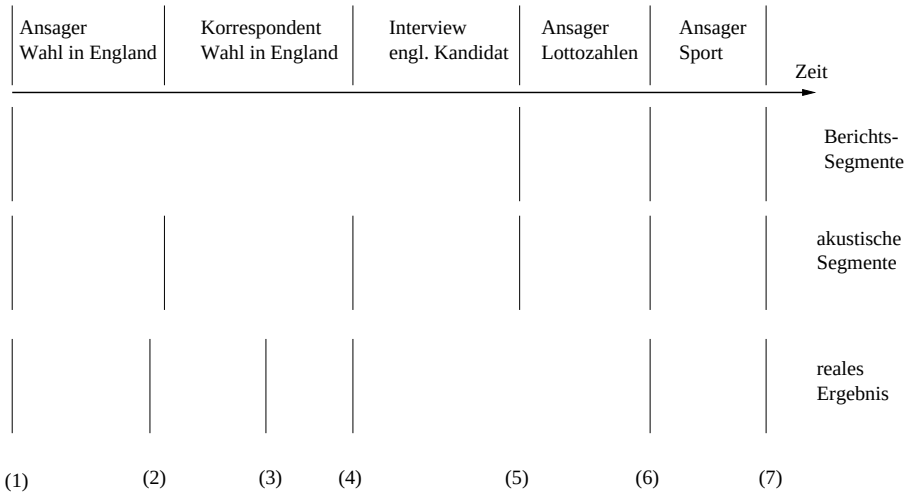


Abbildung 5.1: Segmentierung einer Sendung

Ausgehend von den Hypothesen für die akustischen Segmentgrenzen werden die Segmente im zweiten Schritt inhaltlich analysiert und benachbarte Segmente gleichen Inhalts miteinander verschmolzen. Dies geschieht, indem mit Hilfe der information-retrieval-Komponente des View4You-Systems (vgl. S. 102 ff.) Ähnlichkeiten von benachbarten Segmenten berechnet, und bei Unterschreiten einer vorher festgelegten Schranke die Segmente verschmolzen werden. Im Prototyp des View4You-Systems ist dieser Schritt jedoch nur rudimentär implementiert. Der Rest dieses Kapitels beschäftigt sich daher im Wesentlichen mit dem ersten Schritt, der akustischen Segmentierung.

Eine weitere, wichtige Aufgabe des Segmentierers ist es, Anfang und Ende der eigentlichen Sendung in der Aufnahme zu detektieren und die Teile davor und dahinter wegzuschneiden. Dies ist erforderlich, da aus Sicherheitsgründen die Aufnahme der 'tagesschau' bereits eine Minute zu früh (also um 19:59 Uhr) gestartet und - um

eventuelle Überlängen nicht abzuschneiden - zwei Minuten zu spät (um 20 Uhr 17) beendet wird.

Aufbau dieses Kapitels

Zunächst werden einige Kennzahlen und Methoden eingeführt, die bei der Bewertung von Segmentierungsstrategien eingesetzt werden. Danach wird eine Typisierung in drei grundlegende Algorithmenklassen vorgestellt, die sich an [17] orientiert, nämlich in energiebasierte, modellbasierte und abstandsbasierte Segmentierer. Eine Evaluation aller drei Ansätze auf realen **tagesschau**-Sendungen schließt sich an. Abschließend wird ein im Rahmen dieser Arbeit entwickeltes, hybrides Verfahren vorgestellt und es werden die Ergebnisse dieses Verfahrens mit den Ergebnissen der 'klassischen' Segmentierungsstrategien verglichen.

5.1 Evaluation eines Segmentierers

Um verschiedene Segmentierer miteinander vergleichen zu können, muss eine Kennzahl definiert werden, die sich für jeden Segmentierer berechnen lässt und anhand derer eine Aussage über die Nützlichkeit des Segmentierers getroffen werden kann: Je größer (bzw. kleiner) der Wert der Kennzahl ist, desto besser ist der Segmentierer. Wie aus Abbildung 5.1 hervorgeht, gibt es beim Segmentierungsproblem - analog zum Spracherkennungsproblem - falsche Einfügungen von Segmentgrenzen (z.B. Grenze (3) in der untersten Zeile in Bild 5.1), sowie Auslassungen von Segmentgrenzen (z.B. Grenze (5)). Da es nur einen einzigen Grenztyp gibt, sind Substitutionsfehler (Verwechslung von Segmentgrenzentypen) in diesem Szenario ausgeschlossen.

Die zeitliche Position einer Segmentgrenze ist eine nicht genau bestimmte Größe. Es ist zum Beispiel im Fall einer Sprechpause von einer Sekunde dem Transkribierenden freigestellt, an welchem Moment innerhalb der Sprechpause er eine Segmentgrenzenmarkierung einträgt. Eine *genaue* Übereinstimmung zwischen der Segmentgrenzenmarkierung, die manuell festgelegt wird, und der vom Segmentierer gefundenen Markierung ist daher die Ausnahme. Daher wird zur Evaluation ein Toleranzbereich Δt festgelegt. Eine Grenze gilt dann als korrekt gefunden, wenn sie sich innerhalb des Zeitintervalls $[t_0 - \Delta t, t_0 + \Delta t]$ einer manuell an die Stelle t_0 gesetzten Segmentgrenzenmarkierung befindet.

Die Evaluation selbst erfolgt anhand der folgenden Kriterien:

1. Precision
2. Recall
3. F-measure

Precision und **Recall** berechnen sich dabei als der Prozentsatz korrekt gefundener Grenzen unter allen gefundenen (Precision) bzw. unter allen vorhandenen (Recall) Grenzen. Die beiden Begriffe Precision und Recall werden ausführlich im Anhang auf Seite 200 erläutert.

Die meisten Segmentieralgorithmen können bei unterschiedlichen Arbeitspunkten betrieben werden. Dabei ergibt sich für jeden Arbeitspunkt ein Wertepaar für Precision und Recall. Die vollständige Charakterisierung des Segmentieralgorithmus wird durch einen Graph gegeben, in dem für jeden möglichen Arbeitspunkt Precision und Recall dargestellt werden. Die Abhängigkeit vom Arbeitspunkt kann eliminiert werden, indem nicht Precision und Recall über dem Arbeitspunktparameter, sondern Precision über Recall dargestellt wird. Dadurch wird der Graph auch übersichtlicher. Man bezeichnet einen solchen Graphen als **Charakteristik** des Segmentierers (engl. ROC - receiver operator characteristic). Aus einem solchen Graphen kann unter Kenntnis der relativen Kosten einer nicht gefundenen Grenze (niedriges Recall) sowie der Kosten einer falsch gefundenen Grenze (niedrige Precision) der optimale Arbeitspunkt leicht ermittelt werden.

In vielen Fällen wird die Tatsache, dass eine Charakterisierung des Algorithmus nur durch einen Graphen bzw. mindestens zwei Kennzahlen erfolgt, als störend empfunden. Daher gibt es verschiedene Methoden, die Charakterisierung auf eine einzige Kennzahl zurückzuführen. Die gebräuchlichste Methode ist das Effektivitätsmaß F (engl. F-measure), das von van Rijsbergen in [133] definiert wurde.

Es ist definiert als gewichtetes geometrisches Mittel aus Precision und Recall:

$$F = \frac{PRC * RCL * (1 + \beta^2)}{RCL + \beta^2 * PRC} \quad ; \quad 0 \leq \beta < \infty \tag{5.1}$$

β^2 ist ein freier Parameter des F-Maßes, der eine Gewichtung zwischen der Bedeutung von Precision und der von Recall erlaubt. Bei $\beta^2 = 1$ werden Precision und Recall gleich gewichtet. Für $\beta = 0$ entspricht das F-Maß Precision, für $\beta^2 \to \infty$ entspricht es Recall.

Natürlich hängt der Wert des Effektivitätsmaßes F vom gewählten Arbeitspunkt ab. Der Einfachheit halber wird in der Literatur häufig der größtmögliche Wert von F angegeben, der bei Variation des Arbeitspunktes erreicht werden kann.

Mittelung der Ergebnisse

Die Mittelung der Ergebnisse über mehrere zu segmentierende Sendungen kann auf zwei unterschiedliche Arten erfolgen: aus der **Benutzersicht** und der **Systemsicht**.

Bei der **Benutzersicht** werden für jede Sendung die relevanten Parameter (Precision, Recall, F-Maß) getrennt bestimmt. Diese Parameter werden dann gemittelt. Die so ermittelte Kennzahl gibt einen Erwartungswert für den jeweiligen Parameter

für eine neue, unbekannte Sendung an.

Bei der Evaluation aus **Systemsicht** werden alle Sendungen gemeinsam evaluiert, d.h. die N Sendungen werden wie eine einzige, lange Sendung behandelt und die relevanten Kennzahlen PRC, RCL und F auf dieser langen Sendung evaluiert. Das Ergebnis gibt einen Erwartungswert für den jeweiligen Parameter (PRC, RCL bzw. F) für eine neue, ungesehene Grenze an.

Der Unterschied zwischen den beiden Evaluierungsverfahren tritt bei unterschiedlicher Anzahl zu findender Grenzen pro Sendung zutage. Wenn z.B. die erste Sendung fünfmal mehr Grenzen aufweist als die anderen Sendungen zusammen, dominiert die Performanz des Segmentierers auf dieser Sendung die Ergebnisse aus Systemsicht. Aus Benutzersicht hingegen ist die Performanz des Segmentierers auf dieser Sendung genauso wichtig wie die auf jeder der anderen. Die Systemsicht betrachtet jede zu findende Grenze als gleich wichtig, während die Benutzersicht jede zu segmentierende Sendung als gleich wichtig behandelt, unabhängig von der Anzahl der darin enthaltenen Grenzen.

Welche der beiden Evaluierungsmethoden bevorzugt wird, hängt von der Applikation ab. Für die Evaluation im View4You-System wurde die Systemsicht gewählt, d.h. jede einzelne Grenze wird als gleich wichtig aufgefasst.

5.2 Algorithmen zur Segmentierung

In diesem Abschnitt werden unterschiedlichen Segmentierungsstrategien aus der Literatur vorgestellt. Diese lassen sich grob in drei Kategorien aufteilen:

1. Stillebasierte Segmentierung
2. Segmentierung basierend auf Modellen für die Segmente (modellbasierte Segmentierung)
3. Segmentierung basierend auf Erkennung von Wechseln der akustischen Eigenschaften (abstandsbasierte Segmentierung)

Stillebasierte Segmentierung

Der einfachste Ansatz zur Segmentierung ist das Schneiden bei Stille. Hierzu wird die Energie des Signals berechnet und zeitlich geglättet (letzteres, um Segmentierungen in sehr kurzen Stilleperioden zu vermeiden). Fällt die Energie des Signals unter eine vordefinierte Schranke, wird eine Segmentgrenze hypothetisiert.

Stillebasierte Verfahren sind aufgrund ihrer Einfachheit sehr beliebt. Ihr größter Nachteil liegt in mangelnder Robustheit gegenüber Schwankungen in der Grundlautstärke und Dynamik des Signals. Die Empfindlichkeitsschranke muss in jedem Fall dynamisch an die Datensituation adaptiert werden, wodurch ein Teil des Vorteils, der durch die einfache Implementierung entsteht, wieder verlorengeht.

Modellbasierte Segmentierung

Hat man *a priori* Kenntnisse über die unterschiedlichen Segmentklassen, können diese genutzt werden, um einen modellbasierten Segmentierer aufzubauen.

Dazu wird ein - vollständig verbundenes - HMM mit so vielen Zuständen aufgebaut, wie es unterschiedliche Segmentklassen gibt. Im Falle der 'tagesschau' könnte es z.B. vier verschiedene Segmentklassen (Ansager, Korrespondent, Stille, Musik) geben, die dann durch ein HMM mit 4 Zuständen symbolisiert werden. Durch Duplizieren von einzelnen Zuständen kann erreicht werden, dass sich das HMM mindestens eine vorgegebene Zahl von Zeitschritten in einem Zustand befindet. Auf diese Weise kann z.B. das Wissen genutzt werden, dass ein Ansagersegment nicht kürzer als drei Sekunden sein kann.

Das Prinzip eines so aufgebauten Segmentierers wird von Abbildung 5.2 illustriert.

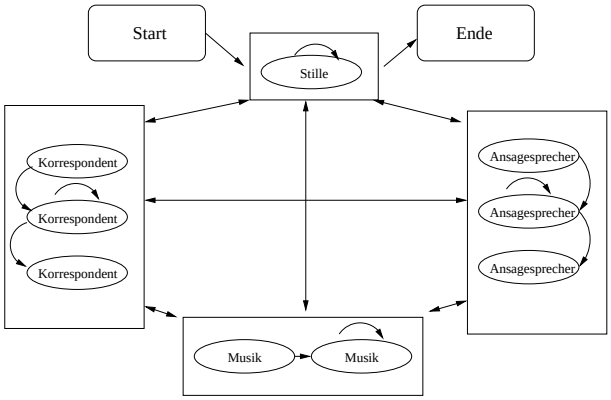


Abbildung 5.2: HMM bei modellbasierter Segmentierung

Dann wird auf der Trainingsstichprobe ein Modell für jede der unterschiedlichen Segmentklassen geschätzt. Hierbei werden parametrische Modelle analog denen der akustischen Modellierung für die Spracherkennung (also z.B. Mixturen von Normalverteilungen) herangezogen.

Um eine Sendung zu segmentieren, wird mit Hilfe des Viterbi-Algorithmus der wahrscheinlichste Pfad durch das HMM, gegeben die Daten der Sendung, berechnet. Jeder Zeitpunkt, an dem der Zustand des HMMs auf dem wahrscheinlichsten Pfad wechselt, ist eine Segmentgrenze.

Hain und Kollegen [18] schlugen 1998 vor, die initiale Segmentierung zur Verbesserung der Segmentmodelle heranzuziehen. Dazu werden die Segmentmodelle anhand der Segmentierungshypothese mit einer MLLR (s. S. 36) adaptiert und der wahrscheinlichste Pfad, und damit die Segmentierung, mit den adaptierten Modellen neu berechnet.

Abstandsbasierte Segmentierung

Abbildung 5.3 verdeutlicht die prinzipielle Funktionsweise der abstandsbasierten Segmentierung.

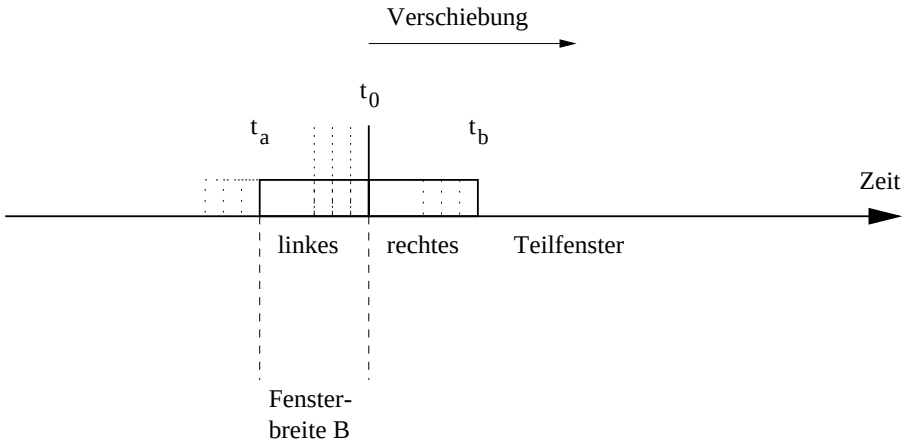


Abbildung 5.3: Funktionsweise der abstandsbasierten Segmentierung

Zwei aneinander angrenzende Fenster der Fensterbreite B werden über die zu segmentierende Sprachprobe geschoben. Für jeden Zeitpunkt wird für jedes der beiden Fenster anhand des im Fenster befindlichen Teils der Sprachprobe ein Modell geschätzt. Der Abstand (der Unterschied) zwischen den beiden Modellen wird berechnet. Je größer der Abstand (Unterschied) zwischen den beiden Modellen ist, desto unterschiedlicher ist der Inhalt der beiden Fenster. Trägt man den so berechneten Abstand zwischen den Modellen des linken und des rechten Teilfensters über der Zeit auf, erhält man eine Kurve ähnlich der in Abbildung 5.4 skizzierten.

Lokale Maxima der Kurve deuten große Unterschiede zwischen dem links angrenzenden und dem rechts angrenzenden Teil der Sprachprobe an. An den Stellen der lokalen Maxima liegt also wahrscheinlich ein Sprecherwechsel bzw. eine Segmentgrenze vor. Durch die Wahl von unterschiedlich hohen Schranken kann der Arbeitspunkt eines abstandsbasierten Segmentierers eingestellt werden: Der Zeitpunkt t_5 in Bild 5.4 wird je nach eingestellter Schranke als Grenze gefunden oder nicht.

Verschiedene abstandsbasierte Segmentierer unterscheiden sich durch die Wahl des Modells, dessen Parameter auf den beiden Teilfenstern geschätzt wird, und durch

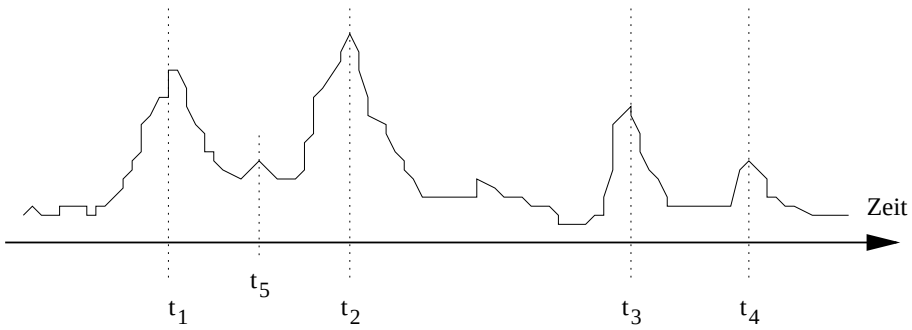


Abbildung 5.4: Ausgabe eines abstandsbasierten Segmentierers

die Abstandsfunktion, mit der der Abstand zwischen den beiden Modellen berechnet wird.

Als Modelle wurden in der Literatur überwiegend Mixturen von N Normalverteilungen, und hierbei wiederum überwiegend solche mit $N = 1$, also einzelne Normalverteilungen vorgeschlagen. Im Folgenden werden drei unterschiedliche Modelle und Abstandsfunktionen vorgestellt.

Kullback-Leibler-Distanz als Abstandsmaß

Die Kullback-Leibler-Distanz ist die zusätzliche Informationsmenge, die zum Codieren von Verteilung B erforderlich ist, wenn ein Code für Verteilung A existiert. Sie ist definiert als

$$d_{KL}(A;B) = E_A < log(P_A) - log(P_B) > \tag{5.2}$$

wobei E_A der Erwartungswert bezüglich der Verteilungsfunktion A ist. d_{KL} ist nicht symmetrisch bezüglich des Vertauschens von A und B und daher strenggenommen keine Distanzfunktion. Es ist aber möglich, d_{KL} zu symmetrisieren:

$$d_{\mathrm{KL,symm.}} = \frac{1}{2}(d_{\mathrm{KL}}(A;B) + d_{\mathrm{KL}}(B;A)) \tag{5.3}$$

Siegler und Stern schlugen in [40] die symmetrisierte Kullback-Leibler-Distanz zwischen zwei multivariaten Normalverteilungen als Abstandsmaß für die Segmentierung vor. Ein großer Vorteil der Kullback-Leibler-Distanz ist, dass es für normalverteilte Daten eine geschlossene Darstellung gibt.

Es gilt nämlich für eindimensionale Normalverteilungen N_A und N_B:

$$d_{\mathrm{KL}} = \int_{-\infty}^{\infty} N_A(x)(\log N_A(x) - \log N_B(x))dx \tag{5.4}$$

Das bestimmte Integral lässt sich nach einiger Rechnung wie folgt darstellen:

$$d_{\mathrm{KL,symm.}} = \frac{1}{2}\frac{\sigma_A^2}{\sigma_B^2} + \frac{1}{2}\frac{\sigma_B^2}{\sigma_A^2} + \frac{1}{2}(\mu_A - \mu_B)^2(\frac{1}{\sigma_A^2} + \frac{1}{\sigma_B^2}) - 1 \tag{5.5}$$

(5.5) ist die gewünschte geschlossene Darstellung.

Da multivariate Gaußverteilungen mit diagonalen Kovarianzen in das Produkt ihrer Komponentennormalverteilungen zerfallen, lässt sich (5.5) leicht auf diesen Fall verallgemeinern.

Siegler und Stern zeigten auch, dass die Kullback-Leibler-Distanz zweier Normalverteilungen eine bessere Segmentierungsperformanz liefert als ihre Mahalanobis-Distanz [40].

Likelihood ratio als Abstandsmaß

Chen [17] verwendete als Abstandsfunktion das Verhältnis der Likelihoods (engl. likelihood ratio). Die verwendete Abstandsfunktion wurde erstmals 1991 von Gish und Kollegen vorgeschlagen [113] und ist auch unter dem Namen Gish-Abstand bekannt.

Das von Gish und Chen verwendete Kriterium lässt sich als Hypothesentest auffassen. Die Nullhypothese lautet, dass beide Teilfenster des Signals derselben stochastischen Quelle entstammen, d.h. Teile desselben Segments sind. Die Alternativhypothese besagt, dass die beiden Teilfenster des Signals unterschiedlichen stochastischen Quellen entstammen, d.h. dass eine Segmentgrenze zwischen den beiden Teilfenstern lokalisiert ist. Der Hypothesentest wird durchgeführt, indem die Wahrscheinlichkeit (likelihood) gegeben die Nullhypothese durch die Wahrscheinlichkeit der Alternativhypothese dividiert wird:

$$\lambda = \frac{L(\vec{x_0}; \vec{\mu_0}, \mathbf{\Sigma_0})}{L(\vec{x_1}; \vec{\mu_1}, \mathbf{\Sigma_1}) L(\vec{x_2}; \vec{\mu_2}, \mathbf{\Sigma_2})} \tag{5.6}$$

wobei die Indizes 1 und 2 für die beiden Teilfenster und der Index 0 für die Vereinigung der beiden Teilfenster stehen. Gl. 5.6 liegt die Annahme zugrunde, dass die stochastischen Prozesse, die das Signal generieren, als multivariate Normalverteilung mit einem Mittelwert und einer (vollen) Kovarianzmatrix modelliert werden können.

Ersetzt man die $\vec{\mu_i}$ und $\mathbf{\Sigma_i}$ durch ihre maximum-likelihood-Schätzungen auf den jeweiligen Signalabschnitten, erhält man

$$-\log(\lambda) = N_0 \log|\mathbf{\Sigma_0}| - N_1 \log|\mathbf{\Sigma_1}| - N_2 \log|\mathbf{\Sigma_2}| \tag{5.7}$$

Der Übergang von (5.6) nach (5.7) basiert auf der Abhängigkeit der Parameter der Normalverteilungen von den Datenpunkten, auf denen die Likelihood bestimmt wird. Das lässt sich bei eindimensionalen Normalverteilungen besonders leicht sehen:

$$
\begin{aligned}
-\log(L) = & \quad -\log \prod_{i=0}^{N} L(x_i, \mu, \sigma) \\
= & \quad -\sum_{i=0}^{N} \log L(x_i, \mu, \sigma) \\
= & \quad -\sum_{i=0}^{N} \log(\tfrac{1}{\sqrt{2\pi}\sigma} exp(-\tfrac{1}{2}(x_i - \mu)\sigma^{-1}(x_i - \mu)) \\
= & \quad N \log(\sqrt{2\pi}\sigma) + \tfrac{1}{2} \sum_{i=0}^{N} (x_i - \mu)\sigma^{-1}(x_i - \mu)) \\
= & \quad N \log(\sqrt{2\pi}\sigma) + \tfrac{1}{2}\sigma^{-1} \sum_{i=0}^{N} (x_i - \mu)^2 \\
= & \quad N \log(\sqrt{2\pi}\sigma) + \tfrac{1}{2}\sigma^{-1} N\sigma \\
= & \quad \tfrac{N}{2}(1 + \log(2\pi)) + N \log(\sigma)
\end{aligned}
$$

Der von σ unabhängige erste Term kürzt sich beim Übergang von (5.6) nach (5.7) durch $N = N_1 + N_2$ heraus.

Für die praktische Berechnung von (5.7) ist es erforderlich, die charakteristischen Parameter von drei Segmenten ($\mathbf{\Sigma_1}$, $\mathbf{\Sigma_2}$, $\mathbf{\Sigma_0}$, $\vec{\mu_1}$, $\vec{\mu_2}$, $\vec{\mu}$) zu schätzen. Das kann natürlich über die Definition auf den Urdaten erfolgen. Es ist aber auch möglich, aus den beiden Teilfenster-Verteilungen $\mathbf{\Sigma_1}, \vec{\mu_1}, \mathbf{\Sigma_2}, \vec{\mu_2}$ die Summenverteilung $\mathbf{\Sigma_0}, \vec{\mu_0}$ direkt zu berechnen.

Der Mittelwert $\vec{\mu_0}$ ergibt sich einfach zu

$$\vec{\mu_0} = E(\vec{x_i}) = \frac{N_1}{N_1 + N_2}\vec{\mu_1} + \frac{N_2}{N_1 + N_2}\vec{\mu_2} \tag{5.8}$$

Die einzelnen Werte $\Sigma_0(x, y)$ der Kovarianzmatrix $\mathbf{\Sigma_0}$ ergeben sich nach der Definition der Kovarianz

$$\Sigma(x, y) = E(xy) - E(x)E(y) = \mu(xy) - \mu(x) - \mu(y) \tag{5.9}$$

Hierin sind $\mu(x)$ und $\mu(y)$, die Erwartungswerte der einzelnen Komponenten der Datenvektoren, bereits aus (5.8) bekannt. Für $E(xy)$ gilt

$$E_0(xy) = \frac{N_1}{N_1 + N_2}E_1(xy) + \frac{N_2}{N_1 + N_2}E_2(xy) \tag{5.10}$$

Die beiden Werte $E_1(xy)$ und $E_2(xy)$ lassen sich aus den (bekannten) Verteilungsparametern $\mu_1(.), \mu_2(.)$ und $\Sigma_1(.,.), \Sigma_2(.,.)$ über die Definition der Kovarianz (5.9) berechnen:

$$E_1(xy) = \Sigma_1(x, y) + \mu_1(x)\mu_1(y) \tag{5.11}$$

Einsetzen ergibt

$$N_0\Sigma_0(x, y) = N_1\Sigma_1(x, y) + N_2\Sigma_2(x, y) + N_1\mu_1(x)\mu_1(y) + N_2\mu_2(x)\mu_2(y) - N_0\mu_0(x)\mu_0(y) \tag{5.12}$$

Die Berechnung nach dieser Vorschrift erspart die Akkumulation für die Gesamtverteilung.

Entropie auf einem VQ-Codebuch als Abstandsmaß

Eine weitere Möglichkeit zur Modellierung der Daten der beiden Fenster beim abstandsbasierten Ansatz ist die Verwendung eines durch Vektorquantisierung vorberechneten Codebuchs. Bei diesem Verfahren werden zunächst ('offline') aus einer großen Menge von Trainingsdaten durch ein Ballungsverfahren eine Anzahl M von Referenzvektoren erzeugt. Das Ensemble der M Referenzvektoren wird als VQ-Codebuch bezeichnet.

Während der eigentlichen Segmentierung wird jedem Datenvektor, der in einem Fenster enthalten ist, der jeweils nächstgelegenen Vektor des VQ-Codebuchs zugeordnet. Damit entsteht für jedes Fenster eine diskrete Wahrscheinlichkeitsverteilung mit M Werten.

Die M-dimensionale Verteilung wird als Beschreibung eines Zufallsexperiments mit M möglichen Ausgängen angesehen und als vollständige Charakterisierung des Fensters aufgefasst. Unähnliche Wahrscheinlichkeitsverteilungen der beiden Fenster implizieren dabei eine Segmentgrenze zwischen den beiden Fenstern.

Bezeichnet man die absolute Häufigkeit, mit der der VQ-Codebuchvektor i einem Datenvektor des Fensters $k(k \in [1,2])$ am nächsten lag, mit $n_k(i)$, lässt sich die gemeinsame Häufigkeitsverteilung $n_0(i)$ wie folgt berechnen:

$$n_0(i) = n_1(i) + n_2(i) \tag{5.13}$$

Bezeichnet man die Gesamtzahl der Datenvektoren des Fensters k mit N_k, so berechnet sich die Entropie der Häufigkeitsverteilung k mit $p_k(i) = \frac{n_k(i)}{N_k}$ zu

$$H_k = \sum_{i=1}^{i=M} p_k(i) \log(p_k(i)) \tag{5.14}$$

Die Abstandsfunktion ist definiert als der Gewinn an Entropie, der beim Codieren der Gesamtverteilung 0 durch eine einzige Verteilung anstelle der separaten Codierung durch zwei Verteilungen 1 und 2 eintritt:

$$d = N_0 H_0 - N_1 H_1 - N_2 H_2 \tag{5.15}$$

Dieses Kriterium wurde bereits 1988 von Lee [122] zur Berechnung der Ähnlichkeit zweier kontextabhängiger Allophonmodelle vorgeschlagen. Es wird im Rahmen dieser Arbeit erstmals zu Segmentierungszwecken eingesetzt.

5.3 Experimente

Alle im vorhergehenden Abschnitt beschriebenen Ansätze zum Segmentieren wurden auf derselben Teststichprobe von 4 Sendungen 'tagesschau' evaluiert. Dabei wurden

für jedes untersuchte Verfahren mehrere Arbeitspunkte jeweils auf allen 4 Aufnahmen untersucht. Die Schwellenparameter, die die Arbeitspunkte definierten, waren dabei jeweils auf allen 4 Aufnahmen gleich.

Evaluation des stillebasierten Ansatzes

Für die Evaluierung des stillebasierten Ansatzes wurden zunächst in einem Abstand von $t = 10$ ms Energiewerte $E(t)$ für jeweils 16 ms breite Fenster berechnet. Das Maximum $\hat{E}$ dieser Energiewerte wurde bestimmt und dann für jedes Fenster die logarithmierte Energie $lE(t)$

$$\mathrm{lE(t)} = \log(\frac{\hat{E}}{10000} + E(t)) \tag{5.16}$$

berechnet. Die Addition eines $\frac{1}{10000}$ der Energie des höchstenergetischen Fensterns vor dem Logarithmieren bewirkt zweierlei: Zunächst wird der Wertebereich des Logarithmus eingeschränkt und die Divergenz bei Null vermieden. Zum Zweiten werden Energien, die klein im Vergleich zu einem Zehntausendstel der maximalen Energie sind, in der Auflösung stark vermindert. Verkürzt kann man sagen, dass der Unterschied zwischen 'sehr leise' und 'sehr sehr leise' durch diese Addition verkleinert wird.

Nach der Logarithmierung wird das gesamte Signal, d.h. jedes Fenster, linear in einen Wertebereich von 0..1 skaliert und geglättet. Die Glättung wird durch eine zweimalige Anwendung der IIR-Filteroperation

$$lE(t) := \frac{1}{9}lE(t-2) + \frac{2}{9}lE(t-1) + \frac{3}{9}lE(t) + \frac{2}{9}lE(t+1) + \frac{1}{9}lE(t+2) \tag{5.17}$$

realisiert.

Die resultierenden, geglätteten logarithmierten Leistungsspektrums-Werte werden als Merkmale des stillebasierten Segmentierers verwendet. Eine Segmentgrenze wird genau dann hypothetisiert, wenn ein Datenpunkt ein lokales Minimum der Energie darstellt und unterhalb der voreingestellten Schranke liegt. Durch Variation der Schranke kann der Arbeitspunkt modifiziert werden.

Evaluation des modellbasierten Ansatzes

Im Falle der Segmentierung der 'tagesschau' wurden vier Klassen unterschieden: Ansagesprecher, Korrespondent, Stille, und Musik. In die Klasse 'Musik' wurden nur rein instrumentale Musiksegmente aufgenommen; Singen sowie Musik als Hintergrundgeräusch wurden als 'Korrespondent' klassifiziert. Die vier Klassen wurden als Mixturen von multivariaten Normalverteilungen modelliert. Als Eingangsdaten wurden sowohl die (auch vom Erkenner verwendeten) Cepstralparameter, als auch einfache logarithmierte Melscale-Koeffizienten verwendet. Vorexperimente ergaben, dass

sich mit Melscale-Koeffizienten durchweg bessere Kennzahlen erzielen ließen als mit Cepstren. Der modellbasierte Segmentierer verwendete aus diesem Grund 16 logarithmierte Mel-Spektralparameter als Eingangsdaten.

Der in der Spracherkennung übliche zeitliche Versatz (engl. frameshift) zwischen zwei Datenvektoren von 10 Millisekunden ist für die Segmentierung zu klein und führt zu unnötig langen Rechenzeiten. Mehrere durchgeführte Kontrastexperimente ergaben, dass ein frameshift von 50 Millisekunden einen guten Kompromiss zwischen Geschwindigkeit und Genauigkeit darstellt. Die Fensterbreite von 16 Millisekunden bei der Berechnung der Spektralparameter wurde dabei beibehalten.

Für jede der vier Klassen (Ansagesprecher, Korrespondent, Stille, Musik) wurde auf der Trainingsstichprobe eine Mixtur von multivariaten Normalverteilungen (mit diagonalen Kovarianzen) geschätzt. Da beispielsweise die Klasse 'Stille' nur eine sehr kleine Klassenvarianz aufweist, wurde die Anzahl der Mixturparameter für die vier Klassen unterschiedlich gewählt:

Klasse	Anzahl Mixturverteilungen
Ansagesprecher	128
Korrespondent	128
Musik	32
Stille	2

In der praktischen Realisierung wurde ein Spracherkenner zum Segmentieren verwendet. Hierzu wurden vier Phoneme entsprechend der vier Klassen verwendet, die jeweils durch die entsprechende Mixtur von Normalverteilungen akustisch modelliert wurden. Das Wörterbuch bestand aus vier Wörtern (Ansager, Korrespondent, Stille, Musik), die durch eine Aneinanderreihung der jeweiligen Phoneme modelliert wurden. Da ein Ansagersegment eine Mindestlänge von etwa 5 Sekunden aufweist, ein Stillesegment aber sehr kurz sein kann, wurden unterschiedliche Mindestlängen für die Worte definiert:

Klasse	Mindestlänge
Ansagesprecher	5 s
Korrespondent	5 s
Musik	2.5 s
Stille	0.2 s

Die Mindestlänge wurde dabei durch Aneinanderreihung von unterschiedlich vielen Basisphonemen realisiert. Da jedes Phonem mindestens ein Datenframe (hier:

50 ms) lang sein muss, kann beispielsweise das Wort 'Musik', das aus 50 hintereinandergereihten Phonemen 'Musik' besteht, nicht kürzer als 2.5 Sekunden (50 mal 0.05 Sekunden) sein. Jedes Phonem wurde durch ein 1-State-HMM modelliert. Das statistische Sprachmodell wurde durch eine Gleichverteilung approximiert.

Die eigentliche Segmentierung erfolgte durch einen Erkennerlauf des so aufgebauten 4-Wort-Erkenners. An allen Stellen, an denen in der Hypothese ein Wortanfang (also der Anfang von 'Stille', 'Ansager', 'Korrespondent' bzw. 'Musik') detektiert worden war, wurde eine Segmentgrenze hypothetisiert.

Durch die Wahl unterschiedlicher Worteingangsstrafen des Sprachmodells konnten unterschiedliche Arbeitspunkte des Segmentierers realisiert werden.

Evaluation des abstandsbasierten Ansatzes

Für alle evaluierten abstandsbasierten Verfahren wurde eine Fensterbreite von jeweils 1.5 Sekunden und ein Versatz von 0.5 Sekunden verwendet. Auf den 1.5 Sekunden breiten Fenstern wurde jeweils ein Modell geschätzt: Eine Normalverteilung mit diagonalen Kovarianzen im Fall des Kullback-Leibler-Abstandes (5.5), sowie eine Normalverteilung mit voller Kovarianzmatrix für den Gish-Abstand (5.6). Die Merkmale sind in allen Fällen 13 nicht LDA-transformierte Mel-Cepstralparameter.

Für den Entropieabstand (5.15) wurde zunächst auf allen Trainingsdaten eine Mixtur von 32 Normalverteilungen auf den 13 Cepstralparametern geschätzt. Für jedes Fenster wurde dann eine Vektorquantisierung auf diese 32 Normalverteilungen berechnet. Als Abstandsfunktion diente dabei die Mahalanobisdistanz

$$d_{Mahalanobis}(\vec{x};\vec{\mu},\Sigma) = (\vec{x}-\vec{\mu})\mathbf{\Sigma}^{-1}(\vec{x}-\vec{\mu})^T \tag{5.18}$$

zwischen dem Datenvektor $\vec{x}$ aus dem Fenster und den Mittelwertvektoren $\vec{\mu_i}$.

Mit Hilfe der unterschiedlichen Abstandsfunktionen wurden dann die Abstände zwischen den Modellen für das linke und das rechte Teilfenster berechnet. Ähnlich wie bei der stillebasierten Segmentierung wurde eine Segmentgrenze genau dort hypothetisiert, wo ein lokales Minimum der Abstände eine vordefinierte Schranke t unterschritt. Durch Variation der Schranke t konnten verschiedene Arbeitspunkte eingestellt werden.

Ergebnisse

Die Evaluation wurde **aus Systemsicht** (siehe Seite 53) auf 4 Tagesschauen (vom 30.3.1997, 13.4.1997, 28.5.1997 und 30.6.1997) durchgeführt. Als Referenzen wurden die 141 von den Transkribierenden markierten Segmentgrenzen verwendet. Da diese angewiesen worden waren, bei jedem Wechsel der akustischen Bedingung eine Segmentgrenze zu setzen, entspricht dies einer Evaluation auf den akustischen Segmenten

(nicht auf den berichtbasierten Segmenten).

Der modellbasierte Segmentierer wurde mit und ohne nachfolgende MLLR-Adaption evaluiert. Durch die Adaption konnten jedoch die Ergebnisse nicht verbessert werden, so dass in den hier vorgestellten Daten die Resultate ohne Adaption dargestellt werden.

Die Ergebnisse der Evaluation sind in Bild 5.5 und in Tabelle 5.1 zusammengefaßt.

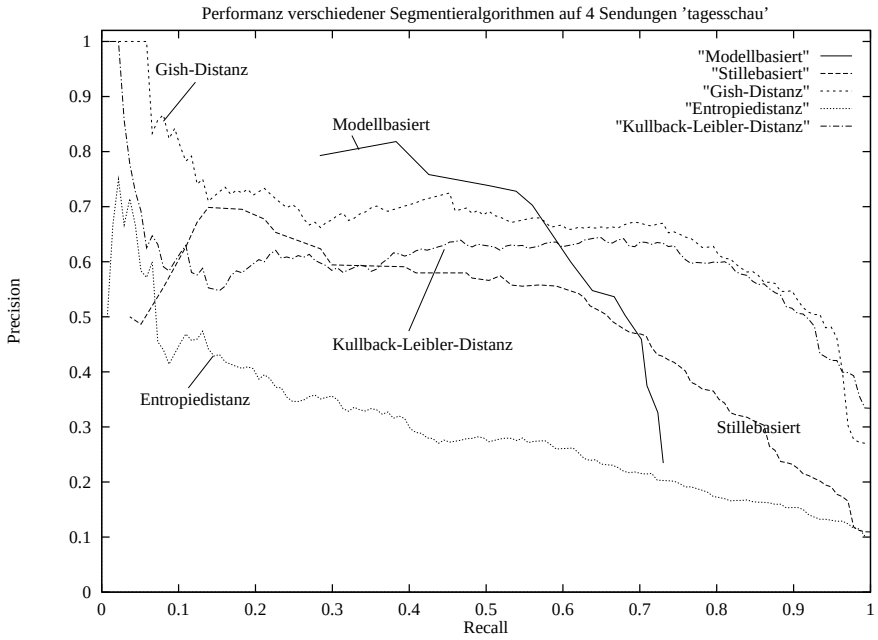


Abbildung 5.5: ROC-Kurven für verschiedene Segmentierungsansätze

Methode	F-Maß
stillebasiert	0.579
Entropiemaß	0.372
Kullback-Leibler-Distanz	0.689
Gish-Distanz	0.702
modellbasiert	0.623

Tabelle 5.1: Maxima des F-Maßes

Man erkennt, dass der schnelle modellbasierte Segmentierer die größten Werte für Precision aufweist. Allerdings fällt die Precision mit steigendem Recall sehr stark ab, so dass ab einer bestimmten Recall-Rate die abstandsbasierten Segmentierer überlegen sind. Der modellbasierte Segmentierer liefert überhaupt keine Ergebnisse mit einem größeren Recall als 73%. Sowohl der maximal erzielbare Recall, als auch die

bei großem Recall erreichbaren Werte für Precision sind bei den abstandsbasierten Verfahren besser.

Unter den abstandsbasierten Verfahren funktioniert die Gish-Distanz am besten, gefolgt von der Kullback-Leibler-Distanz. Die Entropiedifferenz eignet sich als Distanzmaß offenbar weniger gut.

Hybrider Ansatz

Abstandsbasierte Verfahren erzielen hohe Recall-Werte bei relativ guter Precision. Der modellbasierte Ansatz andererseits erzielt hervorragende Precision-Werte bei mittlerem Recall. M. Westphal schlug vor, die Stärken der jeweiligen Verfahren durch einen hybriden Algorithmus zu kombinieren. Ein solcher Algorithmus wird in dieser Arbeit erstmals beschrieben und evaluiert.

Die Grundidee ist, die Modelle für einen modellbasierten Ansatz auf den Ergebnissen der Segmentierung durch einen abstandsbasierten Ansatz aufzubauen. Als Abstandsfunktion wurde dabei nach den Ergebnissen der Evaluation der Gish-Abstand gewählt.

Der hybride Ansatz: Algorithmus

Im ersten Schritt werden in einem zeitlichen Abstand von einer Sekunde auf Fenstern von einer Sekunde Breite die Parameter jeweils einer Normalverteilung mit voller Kovarianzmatrix estimiert. Als Eingabemerkmale dienen dabei 13-dimensionale Mel-Cepstralvektoren, die alle 10 Millisekunden berechnet werden. Als Ergebnis dieses Schrittes stehen für jede Sekunde der zu segmentierenden Sendung ein Mittelwertvektor und eine (volle) Kovarianzmatrix zur Verfügung.

Bei einer durchschnittlichen Länge der 'tagesschau' von gut 15 Minuten oder etwa 930 Sekunden (vgl. Seite 21) entstehen auf diese Weise gut 900 initiale Segmente. Im folgenden Schritt werden die initialen Segmente agglomerativ geballt.

Hierzu werden zunächst die Gish-Abstände jedes Segments mit jedem anderen Segment berechnet. Die beiden Segmente mit dem geringsten Abstand werden dann zu einem Cluster verschmolzen. Diese beiden Segmente müssen dabei nicht nebeneinander liegen. Nach der Verschmelzung wird für den neu entstandenen Cluster ein neuer Mittelwertvektor und eine neue Kovarianzmatrix berechnet. Dies kann zweckmäßigerweise durch Anwendung von Gl. (5.12) geschehen. Danach werden alle Abstände zwischen dem neuen Cluster und allen anderen Segmenten bzw. Clustern berechnet. Da die Abstände zwischen Segmenten bzw. Clustern, die nicht an der Verschmelzung beteiligt waren, durch die Verschmelzung nicht geändert werden, müssen diese nicht neu berechnet werden. Die beschriebene Ballung wird wiederholt, bis nur noch ein einziger Cluster vorhanden ist, der alle 900 Segmente enthält.

Das Endergebnis der agglomerativen Ballung ist natürlich trivial. Die Zwischenergebnisse, bei denen die Ballung weit fortgeschritten ist, sind hingegen interessant. Sind beispielsweise noch 8 Cluster übrig, so sind in jedem Cluster eine Menge von untereinander ähnlichen Sekunden-Segmenten enthalten. Die 8 Cluster können daher als gute *Repräsentanten* der vorliegenden Tagesschau-Sendung aufgefasst werden - bessere Repräsentanten, als die vier statisch definierten Gruppen 'Ansagesprecher', 'Korrespondent', 'Musik' und 'Stille' des modellbasierten Ansatzes sein können.

Im nächsten Schritt werden daher für jeden der 8 Cluster Mixturen von multivariaten Gaußverteilungen - analog zur Vorgehensweise bei der modellbasierten Segmentierung - geschätzt. Mit diesen Modellen wird ein modellbasierter Segmentierer (mit 8 Phonemen und 8 Wörtern) analog zur oben beschriebenen Vorgehensweise konstruiert. Dabei muss allerdings die Mindestlänge aller Worte gleich gewählt werden, da kein a-priori-Wissen über die zu erwartende Mindestlänge der jeweiligen Segmente mehr bekannt ist: es ist beispielsweise nicht bekannt, welcher Cluster (wenn diese überhaupt alle in einen einzigen Cluster geballt wurden) die Stille-Segmente enthält.

Aus demselben Grund wird auch die Zahl der Referenzvektoren pro Phonemmodell für alle Modelle gleich (zu $N = 64$) gewählt.

Das Ergebnis der Segmentierung mit dem hybriden Ansatz ist in Bild 5.6 und Tabelle 5.2 zusammengefaßt. Man erkennt, dass der hybride Segmentierungsansatz die Vorteile der modellbasierten Segmentierung (hohe Präzision bei mittleren Recall-Werten) und der abstandsbasierten Segmentierung (hohes Recall bei mittleren Präzisionswerten) miteinander vereint. Hervorzuheben ist vor allem die außerordentlich hohe Präzision von über 95%, die bei einem Recall von 65% erreicht werden kann. Allerdings ist es nicht möglich, mit dem hybriden Verfahren Recall-Werte von mehr als 87% zu erzielen. Wenn ein solcher Arbeitspunkt erforderlich ist, müssen rein abstandsbasierte Algorithmen eingesetzt werden.

Methode	F-Maß
stillebasiert	0.579
modellbasiert	0.623
Gish-Distanz	0.702
hybrid (6 Cluster)	0.782

Tabelle 5.2: Maxima des F-Maßes

Zusammenfassung

In diesem Kapitel wurden die drei prinzipiellen Ansätze zur akustischen Segmentierung - stillebasierte, modellbasierte und abstandsbasierte Verfahren - vorgestellt. Es wurden für jeden Ansatz ein oder mehrere Segmentiersysteme implementiert, und

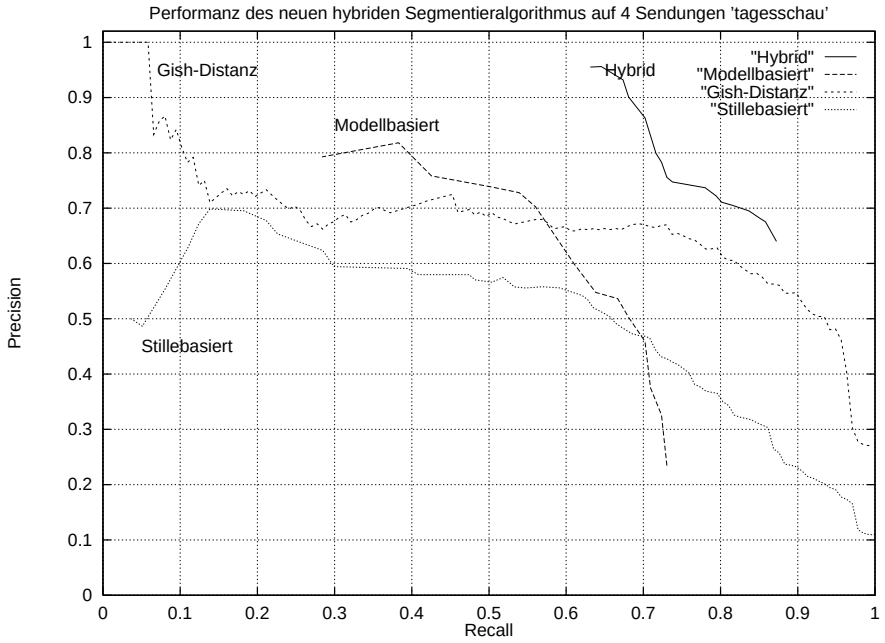


Abbildung 5.6: ROC-Kurven für verschiedene Segmentierungsansätze

alle Segmentierer auf derselben Teststichprobe von 4 Sendungen 'tagesschau' gegeneinander getestet.

Bei den abstandsbasierten Verfahren wurden drei unterschiedliche Abstandsfunktionen evaluiert: der von Gish vorgeschlagene Gish-Abstand [113], die von Siegler verwendete Kullback-Leibler-Distanz [40] und der Entropiedifferenz auf VQ-Codebüchern. Von diesen dreien erwies sich der Gish-Abstand als der beste, gefolgt von der Kullback-Leibler-Distanz.

Der Gish-Abstand-basierte Segmentierer erwies sich im Vergleich mit den beiden anderen Segmentierungsstrategien dem stillebasierten Ansatz als überlegen. Nur im Bereich kleiner Ausschöpfungswerte (Recall, RCL) erzielte der modellbasierte Segmentierer bessere Ergebnisse der Präzision (PRC) als der Gish-Abstand-basierte.

Um die Vorteile der hohen Präzision des modellbasierten Ansatzes mit der hohen Ausschöpfung des abstandsbasierten Ansatzes zu kombinieren, wurde ein neuer hybrider Ansatz entwickelt. Dieser erzielte deutlich bessere Werte der Kennzahlen PRC und RCL als alle anderen Ansätze.

Kapitel 6

Der Spracherkenner des View4You-Systems

6.1 Einleitung

Das Herzstück der Indexerstellung im View4You-System ist der Spracherkenner. Seine Aufgabe ist es, für jedes vom Segmentierer gelieferte Stück einer Sendung eine Verschriftung zu erzeugen.

Der Spracherkenner des View4You-Systems baut auf dem am ISL erstellten JRTk-Spracherkennungs-Toolkit auf. Dieses Toolkit stellt alle im Abschnitt 'Maschinelle Spracherkennung' beschriebenen Standardtechniken zur Verfügung und geht teilweise darüber hinaus. Das JRTk-Toolkit wurde mit einem Aufwand von mehreren Mannjahren angefertigt. Ohne diese Vorarbeiten wäre es nicht möglich gewesen, im gesteckten Zeitrahmen das in dieser Arbeit beschriebene System zu implementieren.

Der Aufbau des Spracherkenners des View4You-Systems entspricht dem momentan (1999) gängigen Paradigma der Spracherkennungsforschung: Ein HMM-basierter Erkenner mit Mixturen von Gaußverteilungen als akustischen Modellen, cepstralen Eingangsvektoren im 10-Millisekunden-Raster, und einem statistischen Trigramm-Sprachmodell. Der prinzipielle Aufbau wurde bereits im Kapitel 'Maschinelle Spracherkennung' beschrieben. Im nun folgenden Kapitel werden die Komponenten des Erkenners einzeln spezifiziert.

Der beschriebene Erkenneraufbau ist nicht der einzig mögliche. Im Rahmen der Entwicklung wurden mehrere Dutzend unterschiedliche Erkenner trainiert und evaluiert, von denen hier nur der endgültige, beste Aufbau beschrieben wird.

6.2 Phonemsatz

Wörter werden im View4You-System als eine Aneinanderreihung von Phonemen modelliert. Der Wahl des Phoneminventars kommt aus diesem Grund entscheidende Bedeutung zu. Wählt man das Inventar zu klein, werden unterschiedliche Wörter auf identische Phonemketten abgebildet. Bei einem zu großen Phonemsatz wiederum entstehen die Probleme bei der Konsistenz im Wörterbuch, der schwierigeren Trainierbarkeit und der fehlenden Generalisierungsfähigkeit der resultierenden Modelle. Der im View4You-System eingesetzte Phonemsatz orientiert sich an den im international standardisierten SAMPA-Alphabet festgelegten Phonemen.

Da der Spracherkennung die Erkennung der **Laute** zugrundeliegt, ist es zweckmäßig, für jeden unterschiedlichen Basislaut ein eigenes Modell einzuführen. Insofern kann strenggenommen nicht von Phonemmodellen, sondern von Phonmodellen gesprochen werden, wie die folgenden Definitionen zeigen:

> Phonem: Kleinste lautliche Einheit der Sprache mit bedeutungsunterscheidender Wirkung, aber nicht für sich alleine bedeutungstragend. Ein System von Phonemen ist charakteristisch für jede Einzelsprache. Dabei ist wesentlich, dass der Wortschatz der Sprache mit möglichst wenig Phonemen eindeutig dargestellt wird. Aus diesem Grund können einem Phonem mehr als nur ein Sprachlaut entsprechen (...)
>
> Phon: Lautsprachliches, von einem Normalhörenden einer Sprachgemeinschaft wahrnehmbares und von anderen Phonen unterscheidbares Element. Phone werden mit Hilfe von Hörtests nachgewiesen und einer endlichen Anzahl von Valenzklassen zugeordnet (...)
>
> *Terminologie der Sprachakustik, Vorlage der Fachgruppe Sprachakustik der Informationstechnischen Gesellschaft im VDE*

Im Deutschen sind die Begriffe Phon und Phonem weitgehend synonym. Die einzige Ausnahme stellt das Phonem 'ch' dar. 'ch' ist ein Phonem mit zwei Aussprachen, der velaren (SAMPA-Symbol 'x', Realisierung wie in 'ach') und der palatalen Variante (SAMPA-Symbol 'C', Realisierung wie in 'ich'). Es gibt kein Wortpaar im Deutschen, das sich nur in der Realisierung des 'ch' unterscheidet, so dass trotz unterschiedlicher Aussprache nach Definition nur ein einziges Phonem vorliegt. Dennoch verwendet der View4You-Erkenner zwei unterschiedliche Modelle für das velare und das palatale 'ch'.

Der verwendete Phonsatz besteht aus 38 Phonen, die aus dem SAMPA-Alphabet abgeleitet sind. Zusätzlich werden 4 Diphthonge explizit, d.h. als Phonemmodelle, modelliert. Ein weiteres Modell für Stille und 6 Geräuschmodelle komplettieren den 49 Basismodelle umfassenden Phonemsatz.

PHONDAT	IPA	Beispiel	PHONDAT	IPA	Beispiel
9	œ	H**ö**lle	2:	ø:	H**öh**le
N	η	Ju**ng**e	S	$\int$	Ta**sch**e
a	a	k**a**nn	a:	a:	K**ah**n
b	b	**b**ei	d	d	**d**u
e	e	M**e**teor	e:	e:	B**ee**t
E	E	B**e**tt	E:	ϵ :	K**ä**se
@	turned-e	lach**e**n	6	turned-a	Les**er**
f	f	**f**rei	g	g	**G**ast
h	h	**H**ast	I	I	r**i**tt
i:	i:	r**ie**t	j	j	**j**a
k	k	**K**ahn	l	l	**L**icht
m	m	**M**auer	n	n	**n**eun
O	open-o	B**o**ck	o:	o:	b**o**g
p	p	**P**latz	r	r	**R**auch
s	s	la**s**	t	t	**T**orte
U	υ	m**u**ß	u:	u:	M**u**s
v	v	**w**ann	x	x	a**ch**
C	ç	di**ch**	Y	Y	H**ü**tte
y:	y:	H**ü**te	z	z	le**s**en

Tabelle 6.1: Monophoneme des View4You-Spracherkenners

Die vier Diphthonge aus Tabelle 6.2 sind sehr häufig und auch stark koartikuliert. Aus diesem Grund wurden explizite Modelle für sie eingeführt.

PHONDAT	IPA	Beispiel
ts	ts	pu**tz**en
aI	aI	r**ei**ten
aU	aU	R**au**ch
OY	open-o Y	h**eu**te

Tabelle 6.2: Explizit modellierte Diphthonge des View4You-Spracherkenners

Die Vokale - mit Ausnahme von 'a' - können im Deutschen sowohl gespannt als auch ungespannt artikuliert werden, wobei die gespannte Version vor allem in Lehnwörtern vorkommt (P**o**litik vs B**o**ck). In der sog. PHONDAT-Notation wird der gespannte Vokal mit einem entsprechenden Kleinbuchstaben ('o'), der ungespannte mit einem Großbuchstaben ('O') dargestellt. In einem der zur Verfügung stehenden Wörterbüchern waren jedoch keine gespannten Vokale enthalten. Da die gespannten Vokale ohnehin relativ selten sind, wurde im View4You- Spracherkenner auf die gespannten Vokale mit Ausnahme des 'e' verzichtet und diese wurden auf die unge-

Symbol	Bedeutung
QK	jedes Phonem
+hBR	Atemgeräusch
+hEH	Häsitation 'Ähh'
+hEM	Häsitation 'Ähm'
+hGH	Sprach-Artefakte (Schmatzen usw)
+hGN	Geräusch (Telefonklingeln usw)
SIL	Stille

Tabelle 6.3: Besondere Phoneme des View4You-Spracherkenners

spannten Versionen abgebildet. In Tabelle 6.4 sind die durchgeführten Abbildungen explizit aufgeführt.

PHONDAT	IPA	Beispiel		PHONDAT	IPA	Beispiel
i	i	Polit**i**k	→	I	I	r**i**tt
o	o	P**o**litik	→	O	open-o	B**o**ck
u	u	K**u**lisse	→	U	υ	m**u**ß
y	y	k**y**rillisch	→	Y	Y	H**ü**tte
2	ø	**Ö**konom	→	9	œ	H**ö**lle

Tabelle 6.4: Abbildung gespannter Vokale auf ungespannte

Nasallaute kommen im Deutschen ursprünglich nicht vor. In französischen Lehnwörtern ('Restaurant') haben sie jedoch Einzug in die Umgangssprache gefunden. Eine explizite Modellierung auf deutschen Daten ist aus zwei Gründen jedoch schwierig. Zum Ersten werden die Nasale von Sprecher zu Sprecher - je nach der Vertrautheit des Sprechers mit dem Französischen - sehr unterschiedlich artikuliert, zum Zweiten sind die Nasale sehr selten, so dass es nicht möglich ist, robuste Modelle für sie zu schätzen. Aus diesem Grund enthält der Phonemsatz des View4You keine Nasale, sondern bildet diese auf deutsche Phoneme anhand der Abbildungsvorschrift in Tabelle 6.5 ab.

PHONDAT	IPA	Beispiel		PHONDAT	IPA	Beispiel
Z	Yogh	Lo**g**e	→	S	$\int$	Ta**sch**e
9	$\tilde{\text{œ}}$	Parf**u**m	→	9	œ	H**ö**lle
a	$\tilde{\text{a}}$	Restaur**ant**	→	aN	aη	R**ang**
O	$\tilde{\text{O}}$	Sais**on**	→	ON	Oη	K**on**gress
E	$\tilde{\epsilon}$	T**eint**	→	E:	ϵ :	K**ä**se

Tabelle 6.5: Benutzte Abbildungen von Nasalen auf Konsonanten des View4You-Erkenners

Die beiden Lautpaare 'ON' und 'aN' in Tabelle 6.5 werden im View4You-System durch Konkatenation der Basismodelle 'O' und 'N' bzw. 'a' und 'N' modelliert.

6.3 Phonetische Äquivalenzklassen

Sämtliche Modelle im View4You-Spracherkenner sind Entscheidungsbaum-geballte Subphoneme. Um die Ballung (engl. clustering) der Modelle durchführen zu können, sind phonetische Äquivalenzklassen erforderlich, die die einzelnen Phonemmodelle in linguistisch motivierte Gruppen einordnen. Diese Gruppeneinteilung erfordert entweder phonetisches a-priori-Wissen oder sie kann automatisch aus einer Trainingsstichprobe berechnet werden. Obwohl neuere Ergebnisse ([10], [6]) die automatische Fragenerstellung favorisieren, sind diese keinesfalls zwingend. In [6] wird auf einem Phonemset eine Verschlechterung und auf einem anderen eine Verbesserung erzielt, wobei die Verbesserung nur 2% relativ beträgt, also statistisch insignifikant ist. Zudem liegen die Basisfehlerraten mit 25,9% auf den Ansagesprechern von Nachrichtensendungen um mindestens einen Faktor zwei über den besten zur Zeit erreichbaren Werten. Dasselbe Problem stellt sich bei [10], wo zwar eine Verbesserung erzielt wurde, jedoch ausgehend von einer Fehlerrate von 12,04% auf einer Teststichprobe, auf der Mei-Yuh Hwang [99] schon 1993 eine Fehlerrate von nur 7,1% erzielte. In der Arbeit von Hwang wurde auch aufgezeigt, dass die Ballung auf Grund von phonetisch motivierten Fragen dieselbe Performanz erzielte wie ein hybrides System aus automatisch durchgeführten Ballungen für gesehene, und Entscheidungsbaum-basierten Ballungen für ungesehene Triphone.

Aus diesem Grund wurde für das endgültige System auf die automatische Ballung verzichtet. Insgesamt wurden 72 manuell generierte Gruppen eingesetzt, denen noch 18 stellungsabhängige Gruppen für die Diphthonge hinzugefügt wurden. Die genaue Einteilung ist im Anhang dargestellt.

6.4 Sprachmodell, Vokabular und Wörterbuch

Bei kaum einer Aufgabe in der maschinellen Spracherkennung gilt das Bonmot 'there is no data like more data' so sehr wie bei der Erstellung eines Trigramm-Sprachmodells. Zur Erstellung des Sprachmodells wurden daher Textdaten aus fünf verschiedenen Quellen zusammengenommen: Süddeutsche Zeitung, Frankfurter Allgemeine Zeitung, Bayern 5 Radio, der Internet-Zeitung 'Germany Live', sowie Serverabzüge des *tagesschau*-Servers mit Transkripten der Ansagesprecher einiger Tagesschau- und Tagesthemen-Sendungen. In all diesen Texten wurde die Groß-Kleinschreibung am Satzanfang anhand eines Wörterbuchs normalisiert, es wurden die Satzzeichen getilgt und die offensichtlichen Fehler korrigiert. Weil die Sendungen der Kreuzvalidierungsstichprobe aus dem Jahr 1997 stammten, wurden nur Textdaten

verwendet, die spätestens am 28.2.1997 erzeugt wurden. Der 'Sicherheitsabstand' zur frühesten Testsendung, der vom 30.03.1997, beträgt damit einen Monat. Die Textgrößen der fünf Korpora sind in Tabelle 6.6 zusammengefasst.

Korpus	Sätze	Wörter	Bytes
Tagesschau	10.442	113.328	805.652
Germany Live	244.681	3.821.552	27.273.628
Bayern 5	150.689	2.119.375	15.653.515
FAZ	2.100.603	39.668.914	286.067.729
SZ 1995	1.839.732	28.582.448	201.901.782
SZ 1996	1.862.219	28.531.746	200.880.145
Summe	6.208.366	102.837.363	732.582.451

Tabelle 6.6: Verwendete Texte für das statistische Sprachmodell

Das Vokabular wurde auf die häufigsten 60789 Wörter dieses Korpus festgelegt. Das seltenste Wort im Korpus, das auf diese Weise noch in das Vokabular aufgenommen wird, ist 36mal belegt.

Wenn alle im Korpus beobachteten Bigramme und Trigramme explizit ins Sprachmodell aufgenommen werden, wird dieses zu groß für den Speicher einer typischen Workstation. Aus diesem Grunde wurden bei der Erstellung des Sprachmodells Bigramme und Trigramme, die nur einmal oder zweimal beobachtet wurden, lediglich zur Berechnung der Backoff-Wahrscheinlichkeiten herangezogen, aber nicht explizit modelliert. Mit diesem sogenannten *cutoff* ergeben sich explizit noch 2.468.775 Bigramme und 4.613.739 Trigramme. Bei der Erstellung des Modells wurde zur Berechnung der Backoffs ein von Kneser und Ney vorgeschlagenes Verfahren [73] zusammen mit *absolute discounting* (4.9) eingesetzt. Das so erstellte Sprachmodell weist eine Perplexität von 383 (gemessen auf den 4 Sendungen vom 30.3.97, 13.4.97, 28.5.97 und 30.6.97) auf. Die Vokabularabdeckung beträgt 95,57% (379 von 8554 Wörtern sind unbekannt).

Andere Sprachmodelle

Um die Auswirkung der verwendeten Datenmenge auf die Qualität des Sprachmodells zu sehen, wurden die Texte der 'Süddeutschen Zeitung' aus dem Trainingsmaterial entfernt und mit der verbleibenden knappen Hälfte des Materials ein zweites Sprachmodell trainiert. Dieses wurde in Erkennungsexperimenten mit dem ursprünglichen Sprachmodell verglichen. Die Ergebnisse zeigt Tabelle 6.7. Die Fehlerrate steigt um zwei Prozent absolut an, wenn der Sprachmodell-Trainingskorpus halbiert wird.

Dieses Ergebnis lässt vermuten, dass eine weitere Vergrößerung des Sprachmodell-Trainingskorpus die Wortfehlerrate reduzieren könnte. Daher wurden in einem weiteren Experiment Textmeldungen der Nachrichtenagenturen dpa, ap und afp der Jahre

Datenmenge im Sprachmodell	Wortfehlerrate (%)
44 MWorte	28,9%
102 MWorte	26,9%

Tabelle 6.7: Wortfehlerrate bei weniger Sprachmodell-Trainingsdaten

1993-1995 zum Sprachmodelltext hinzugenommen. Weil die 'Süddeutsche Zeitung' diese Informationen in ihren Artikeln verwertet, entsteht dadurch ein Überlapp und die neuen Daten können die Wortstatistiken verfälschen. Die Größe der hinzugenommenen Korpora ist in Tabelle 6.8 dargestellt. Für dieses Experiment wurde ein anderes Basissystem mit einer niedrigeren Fehlerrate als in Tabelle 6.7 verwendet.

Das Ergebnis in Tabelle 6.8 zeigt, dass die Wortfehlerrate um (statistisch insignifikante) 1.3% relativ ansteigt. Da die Sprachmodelle gleichzeitig größer sind, also mehr Speicher und Dekodierzeit beanspruchen, wurde auf die Daten der Nachrichtenagenturen in allen weiteren Experimenten verzichtet.

Korpus	Sätze	Wörter	Bytes
ap 1993-1995	3.131.498	55.741.705	405.972.091
dpa 1992-1995	2.317.953	41.451.258	294.662.683
afp 1993-1995	1.679.877	28.886.669	208.587.720

Tabelle 6.8: Nachrichtenagentur-Korpora

Sprachmodell	Wortfehlerrate
Ausgangspunkt (102 Millionen Wörter)	21,3%
plus Agenturen	21,6%

Tabelle 6.9: Erkennungsergebnisse mit/ohne Texte der Agenturen

Wörterbuch

Das Wörterbuch ist die zentrale Wissensquelle eines Spracherkenners. Eine falsche phonetische Umschrift der Wörter im Wörterbuch führt nicht nur zu einer Fehlerkennung infolge der von der Umschrift abweichenden Realisierung, sondern auch zu einem fehlerhaften Training der Phonemmodelle und damit zu schlechter Erkennungsleistung auch auf den korrekt umschriebenen Wörtern.

Von den 60.789 Wörtern des Vokabulars waren bei Beginn der Arbeiten nur gut 25.000 im Wörterbuch enthalten. Die restlichen mehr als 30.000 Einträge wurden während der Entwicklung des View4You-Systems manuell erstellt.

Viele Wörter, vor allem die häufigeren, aber auch viele Lehnwörter, werden je nach Dialekt mit verschiedenen Betonungen und Aussprachen realisiert. Ein gutes

Wörterbuch muss daher nicht nur die 'hochdeutsche', sondern auch die gebräuchlichsten umgangssprachlichen Ausspracheformen jeden Wörters enthalten. Sofern gebräuchliche Aussprachevarianten bekannt waren, wurden diese bei der Erstellung des Wörterbuches mit eingegeben.

Im Mittel hat jedes Wort im Wörterbuch 1,1 Aussprachevarianten. Wenn alle im Wörterbuch vorhandenen Aussprachevarianten der 60789 Wörter des Erkennerwortschatzes in das Erkennerwörterbuch aufgenommen würden, ergäben sich rund 66.000 (genau 66417) Einträge. Zu viele Aussprachevarianten verschlechtern allerdings das Erkennungsergebnis. In einer Reihe von Experimenten wurde ermittelt, dass die Aufnahme der Aussprachevarianten der häufigsten 10000 Wörter des Vokabulars die beste Performanz ergibt. Die endgültige Größe des Erkennerwörterbuchs, inklusive der Aussprachevarianten, liegt damit bei 61333 Einträgen.

6.5 Vorverarbeitung

Die Vorverarbeitung folgt im Wesentlichen dem Schema von Bild 4.1. Im Folgenden werden die verwendeten Faktoren spezifiziert und die einzelnen Algorithmen genauer beschrieben.

Die MPEG-dekomprimierten, auf 16 kHz heruntergesampelten Werte - die Originaldaten sind mit 44,1 kHz abgetastet - werden zunächst mittelwertfrei gemacht, um einen eventuellen Offset des verwendeten A/D-Wandlers auszugleichen. Das Sprachsignal wird dann in Segmente ('Frames') von jeweils 16 Millisekunden Dauer (256 Abtastwerte) unterteilt, wobei die Segmente im Abstand von 10 Millisekunden aufeinander folgen. Jedes Segment überlappt also um 6 Millisekunden mit den Nachbarsegmenten.

Jedes Segment wird für sich mit einer Hamming-Fensterfunktion [134] multipliziert. Das Hammingfenster verbessert gegenüber dem Rechteckfenster die Frequenzcharakteristik des resultierenden Spektrums, indem es eine breitere, aber weniger intensive 'Verschmierung' der Frequenzbänder bewirkt. Es ist definiert als

$$h(t) = 0.54 - 0.46 \cos(\frac{2\pi t}{T}) \tag{6.1}$$

Die 256 mit dem Hammingfenster multiplizierten Abtastwerte werden danach fouriertransformiert, und es wird der Betrag der entstehenden Fourierkomponenten berechnet. Im nächsten Schritt wird das resultierende Leistungsspektrum vokaltraktlängennormiert (VTLN). Dazu wird zunächst eine Schätzung des relativen Längenparameters α vorgenommen ($0,8 \leq \alpha \leq 1,2$). Das Spektrum wird stückweise linear transformiert. Dazu wird eine Eckfrequenz e von 0,8 mal der halben Abtastrate

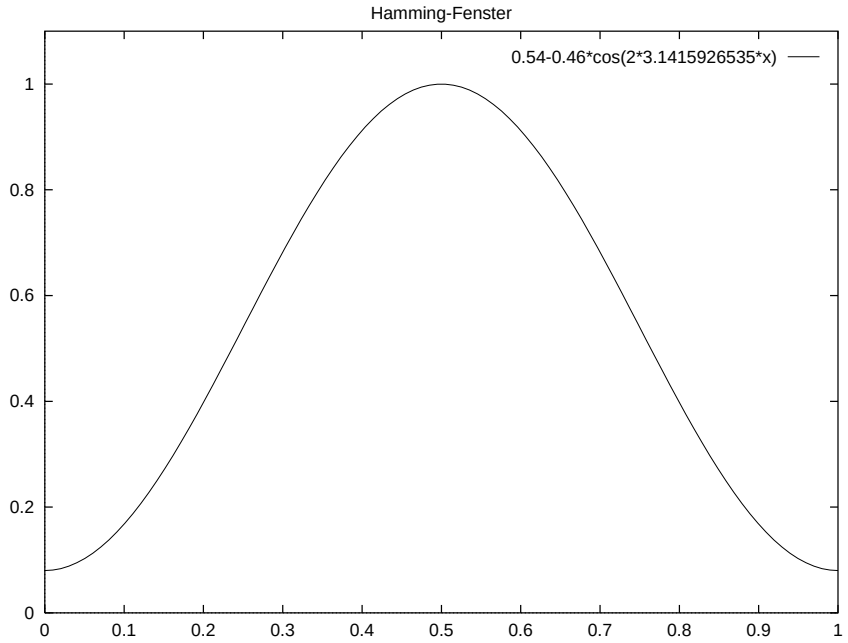


Abbildung 6.1: Hamming-Fenster

definiert. Die Abbildungsvorschrift lautet dann

$$y(f) = \begin{cases} \frac{f}{\alpha} & : \quad f \leq e \\ \frac{\frac{e}{\alpha} - e + (1 - \frac{e}{\alpha})f}{1-e} & : \quad 1 \geq f > e \end{cases} \tag{6.2}$$

Das so entstandene Leistungsspektrum wird mit einer Mel-Filterbank von 129 auf 30 Dimensionen reduziert. Die Mel-Filterbank fasst dabei jeweils mehrere Frequenzkomponenten des Leistungsspektrums in einer neuen Komponente zusammen, wobei tiefe Frequenzen weniger stark zusammengefaßt werden als hohe. Die Frequenzauflösung wird auf diese Weise im hohen Bereich des Spektrums stärker reduziert als im tiefen Bereich. Die Transformation vom Frequenz- in den Mel-Bereich wird nach der sog. Bark-Skala

$$\text{mel(hz)} = 2595 * \log_{10}(1 + \text{hz}/700) \tag{6.3}$$

berechnet, so dass das 1. Frequenzband mit f = 62.5 Hz zu 96,4 mel und das 127. Frequenzband mit f = 8000 Hz zu 2840 mel transformiert wird. Die eigentliche Filterbank ist im Mel-Raum äquidistant und verwendet überlappende Dreieckfilter mit einer Breite von 2. Das heißt, der zweite Koeffizient beginnt bei 1, hat bei 2 sein Maximum und endet bei 3, während der dritte Koeffizient von 2 bis 4 reicht. Auf diese Weise wird eine Glättung im Frequenzbereich erzielt. Alle Filterbänke sind normalisiert, so dass weißes Rauschen im Frequenzbereich sich auf weißes Rauschen im Mel-Bereich abbildet.

Zu den 30 Werten des Mel-Leistungsspektrums wird eine Konstante 1 addiert und der resultierende Wert logarithmiert. Das so logarithmierte Mel-Leistungsspektrum wird dann invers fouriertransformiert, um 30 Cepstralkoeffizienten zu erhalten. Von diesen werden nur die ersten 13 verwendet, die übrigen 17 Koeffizienten werden ignoriert.

Die resultierenden Cepstren werden mittelwertfrei gemacht (CMS - cepstral mean subtraction), wobei zur Berechnung des Mittelwerts nur Daten aus Nicht-Stille-Bereichen verwendet werden (definiert durch eine Leistung von weniger als $\frac{1}{100000}$ der Leistung des lautesten Frames). Daraufhin werden die ersten und zweiten Ableitungen dieser Koeffizienten durch $d(t) = x(t+2) - x(t-2)$ und $dd(t) = d(t+2) - d(t-2)$ approximiert und die Cepstren zusammen mit diesen Ableitungen zu einem 39-dimensionalen Merkmalsvektor zusammengefasst. Dieser wird durch eine LDA-Transformation, bei dem jedes akustische Modell eine Klasse definiert, auf den 16-dimensionalen endgültigen Merkmalsvektor transformiert.

6.6 Akustische Modellierung

Ein Wortmodell besteht aus einer Aneinanderreihung von Phonemmodellen. Dabei wird jedes Phonem durch drei HMM-Zustände modelliert, von denen der erste den Anlaut samt Koartikulation mit dem vorhergehenden Phonem, der zweite die zeitlich konstante Phase der Artikulation und der dritte Zustand den Auslaut und die Koartikulation mit dem nachfolgenden Phonem modelliert. Jeder HMM-Zustand kann in sich selbst oder in seinen Nachfolgezustand übergehen; das 'Überspringen' von Zuständen ist also nicht erlaubt und jedes Phonem hat eine Mindestlänge von 3 Zuständen, entsprechend 3 Frames oder 30 ms. Alle Übergangswahrscheinlichkeiten werden einheitlich auf 1 gesetzt. Auf ein Training der Übergangswahrscheinlichkeiten wurde verzichtet, da Vorexperimente keine Steigerung der Erkennungsleistung ergeben hatten.

Der Aufbau des HMM für das Wort 'der' wurde bereits in Bild 4.4 auf Seite 49 skizziert.

Akustische Modellierung der HMM-Zustände

Die eigentliche Modellierung der Emissionsverteilungen der HMM-Zustände erfolgt durch Mixturen von Normalverteilungen (engl. *mixtures of gaussians*, vgl. Seite 38). Dabei wurde die Zahl der Normalverteilungen pro Mixtur auf 30 und die Dimensionalität auf 16 festgelegt.

Da die Vorverarbeitung eine lineare Diskriminanzanalyse enthält, sind die einzelnen Dimensionen der Merkmalsvektoren im Mittel über alle Klassen dekorreliert. Aus diesem Grund wurde nicht die vollständige Kovarianzmatrix, sondern nur ihre

Hauptdiagonale modelliert (diagonale Kovarianzen), so dass die multivariate Normalverteilung in das Produkt der univariaten Komponentennormalverteilungen zerfällt.

Jeder HMM-Zustand verfügt über ein eigenes Modell, bestehend aus 30 Normalverteilungen mit diagonalen Varianzen - Codebuch genannt - und 30 Werten für die apriori-Wahrscheinlichkeiten der einzelnen Normalverteilungen. Weder die Codebücher noch die Varianzen werden über verschiedene Modelle hinweg geteilt. Damit realisiert der View4You-Erkenner ein sogenanntes *fully continuous HMM*.

Im View4You-Erkenner kommt je nach Größe des Trainingskorpus eine unterschiedliche Anzahl von kontextabhängigen Triphonen zum Einsatz. Jedes Triphon verfügt dabei über drei Zustände, von denen wiederum jeder separat durch eine Mixtur von 30 Normalverteilungen modelliert wird. Bei Verwendung des gesamten Trainingskorpus werden insgesamt maximal 5000 Subtriphone modelliert. Insgesamt sind damit 5000 (Anzahl der Subtriphone) * 30 (Normalverteilungen pro Zustand) = 150.000 Normalverteilungen zu schätzen.

Die Ballung der kontextabhängigen Triphone erfolgt dabei divisiv wie auf Seite 44 beschrieben. Die ersten fünf verwendeten Fragen (die die ausgeprägtesten kontextabhängigen Laut-Modifikationen anzeigen) sind in Tabelle 6.10 zusammengefasst. Die linke Spalte zeigt dabei das Subphonem in der Schreibweise 'Phonem - Stellungsmarker', wobei der 'Stellungsmarker' die drei Werte 'b', 'm' und 'e' (für 'Beginn', 'Mitte' und 'Ende') annehmen kann. 'D-b' bedeutet also den Anfang ('-b') des Phonems 'D'.

In der rechten Spalte ist diejenige Frage aus dem im Anhang D ab Seite 192 beschriebenen Fragensatz angegeben, die die Menge aller beobachteten Subphoneme dieses Typs (also im Beispiel alle Beginn-Zustände von 'D', unabhängig von den benachbarten Phonemen) in zwei möglichst unterschiedliche Gruppen aufteilt. Der Entropieverlust bei dieser Aufteilung ist das Sortierkriterium in Tabelle 6.10.

N-e	rechts = SILENCE?
D-b	links = SILENCE?
6-e	rechts = NASAL?
T-b	links = KONS-PARTVOICED?
R-e	rechts = VOKAL?

Tabelle 6.10: Die wichtigsten 5 Kontextfragen

Die stärkste Kontextabhängigkeit weist also der hintere Teil des Phonems 'N' mit seinem rechten Nachbarn auf. Es ändert sich dabei vor allem dann, wenn Stille als Nachbar folgt.

Die Modelle für Stille sowie die Geräuschmodelle wurden kontextunabhängig modelliert. 'Stille' wurde abweichend von allen anderen Phonemen nur durch einen einzigen Zustand (anstelle von dreien) dargestellt.

Zeitaufwand für das Training

Es ist interessant, den Zeitaufwand für das Training des Erkenners I39, der auf 60 transkribierten **tagesschau**-Sendungen trainiert wurde, im Detail zu analysieren. Tabelle 6.11 fasst den Zeitaufwand jedes Arbeitsschrittes zusammen.

Schritt	Zeitaufwand absolut	Zeitaufwand relativ
LDA-Matrix auf Phonemklassen berechnen	1h24'	3%
Codebücher extrahieren und clustern	1h39'	3.6%
3 Iterationen Viterbi-Training	5h55'	12.9%
phonetische Kontexte extrahieren	42'	1.5%
Training einer Verteilung pro phon. Kontext	2h34'	5.6%
Berechnen des Polyphon-Clusterbaumes	19h01'	41.3%
LDA-Matrix auf Polyphonklassen berechnen	1h25'	3.1%
Codebücher extrahieren und clustern	4h44'	10.3%
5 Iterationen Viterbi-Training	9h16'	20.1%
Summe	46h40'	100%

Tabelle 6.11: Zeitaufwand für die einzelnen Trainingsschritte

Man erkennt, dass fast die Hälfte der Trainingszeit bei der Ballung (Clustering) der Polyphone zu generalisierten Polyphonen beansprucht wird. Dieser Trainingsschritt wurde daher parallel auf vier Workstations gerechnet.

6.7 Optimierung der Parameterraumgröße

Eine wichtige Frage beim Design eines Spracherkenners, wenn nicht sogar die für die Performanz wichtigste Frage überhaupt, ist die nach der Anzahl und Allokation der Parameter für die akustischen Modelle.

Die 'richtige' Modellgröße stellt dabei einen Kompromiss dar zwischen der Feinheit der Modellierung durch mehr Parameter einerseits und der abnehmenden Robustheit der Schätzung dieser Parameter andererseits. In der Praxis wird das Problem meist dadurch gelöst, dass durch einen Systemingenieur eine 'vernünftige' Parameterraumgröße ad-hoc definiert wird, wobei diese Entscheidung auf dem Erfahrungsschatz des Systemingenieurs basiert. Grundlagen seiner Entscheidung sind

- die Menge der Trainingsdaten,
- die Art der Erkennungsaufgabe im Hinblick auf die Varianz der akustischen Bedingungen,
- der Grad der Abweichung der Trainingsdaten von den zu erwartenden Daten im Einsatz des Systems,

- die Qualität der Trainingsdaten,
- die Anforderungen hinsichtlich Echtzeit und Speicherbedarf,
- die Menge und Qualität der Textdaten und damit des Sprachmodells.

Ausgehend von dem so konstruierten Basissystem werden dann größere und kleinere Parameterräume evaluiert. Dasjenige System, das dabei die niedrigsten Fehlerraten auf einer Kreuzvalidierungsstichprobe erzielt, wird schließlich eingesetzt.

Eines der Ziele dieser Arbeit war es, ein vollautomatisch lernendes System zu konstruieren, das anhand der kontinuierlich eintreffenden Aufnahmen von immer neuen Fernsehnachrichtensendungen seine Performanz permanent verbessert. Für ein solches System ist die geschilderte Vorgehensweise zur Bestimmung der Parameteranzahl nicht wünschenswert. Das Wissen und die Erfahrung des Systemingenieurs sollten bei einem solchen System bereits zur Zeit des Systemstarts in Form von Formeln und Heuristiken implementiert sein. Um die Ermittlung und Kodierung des erforderlichen Wissens geht es im folgenden Abschnitt. Dabei kann die Aufgabenstellung gegenüber der oben skizzierten, allgemeineren Aufgabenstellung vereinfacht werden: Weder der Grad der Abweichung der Trainingsdaten von den Testdaten, noch die Qualität der Trainingsdaten, noch die Art der Erkennungsaufgabe ist innerhalb des Kontextes des View4You-Systems variabel. Auch die Anforderungen hinsichtlich Echtzeit und Speicherbedarf ändern sich nicht. Es müssen also im Wesentlichen die Abhängigkeiten von der *Menge* der Trainingsdaten untersucht werden.

Parameterallokation

Die Frage nach der optimalen Allozierung der Parameter in einem parametrischen Modell hängt von der Art des verwendeten Modells ab. Das im View4You-System verwendete Modell wurde bereits ausführlich dargestellt. Seine für die Parameterallokation relevanten Eigenschaften lassen sich in fünf Punkten zusammenfassen:

- Sätze werden als Folge von Wörtern modelliert.
- Wörter werden als Folge von Phonemen modelliert.
- Phoneme werden als Folge von drei kontextabhängigen Subphonemen modelliert.
- Modellierung des Satzes als HMM, wobei die Zustände den kontextabhängigen Subphonemen entsprechen.
- Modellierung der Emissionswahrscheinlichkeiten der HMM-Zustände durch Mixturen von Normalverteilungen.

Die freien Parameter dieses parametrischen Modells sind also

1. das Wörterbuch,
2. der Phonemsatz,
3. die Anzahl der kontextabhängigen Subphoneme,
4. die Zustandsübergangswahrscheinlichkeiten,
5. die Anzahl der Parameter pro Subphonem.

Der Phonemsatz als solcher ist dabei nur von der Sprache abhängig, nicht jedoch von der Menge der verfügbaren Trainingdaten.

Es hat sich gezeigt, dass die Leistung von Spracherkennern nicht oder kaum durch die Zustandsübergangswahrscheinlichkeiten beeinflusst wird. Im View4You-System wurde daher auf deren Modellierung komplett verzichtet.

Die verbleibenden Parameter, die einer Optimierung offenstehen, sind

- das Wörterbuch,
- die Anzahl der kontextabhängigen Subphoneme,
- die Anzahl der Parameter pro Subphonem.

Das Wörterbuch

Das (statische) Wörterbuch wurde bereits auf Seite 72 behandelt. Die dynamische Adaption des Vokabulars an neue Namen und Begriffe sowie an neue Themengebiete wird auf Seite 87 ff. besprochen.

Anzahl der kontextabhängigen Subphoneme

Hält man die Anzahl der Normalverteilungen pro Subphonem konstant (im View4You-System bei 30 Normalverteilungen pro Subphonem), nimmt die Anzahl der Parameter im System mit der Anzahl der unterschiedlich modellierten kontextabhängigen Subphoneme zu.
Die Anzahl der Subphonemmodelle wird dabei nach *unten* durch die Anzahl der Phoneme eingeschränkt - jedes Phonem hat mindestens 3 Subphonemmodelle.

Nach *oben* wird die mögliche Zahl der unterschiedlichen Subphonemmodelle von der Zahl der unterschiedlichen Phonemkontexte begrenzt, die im Trainingsset vorkommen - es ist nicht möglich, für einen nicht beobachteten Phonemkontext ein Modell zu konstruieren.

Die Zahl der unterschiedlichen beobachteten Phonemkontexte ist allerdings sehr groß. Im Falle des View4You-Systems werden beispielsweise auf einer Trainingsmenge von 60 Sendungen 'tagesschau' mehr als 190.000 unterschiedliche Phonemkontexte gesehen, die zu mehr als 570.000 verschiedenen Subphonemmodellen führen könnten. Die meisten dieser Kontexte werden allerdings nur einmal beobachtet, so dass die eigentlichen Subphonemmodelle als Modelle für ganze Klassen von Kontexten trainiert werden. Die Klassenbildung geschieht dabei divisiv (ausgehend vom Monophon) mit Hilfe von Entscheidungsbäumen (siehe Seite 44).

Je weiter das divisive Aufspalten betrieben wird, d.h. je tiefer der Entscheidungsbaum ist, desto mehr unterschiedliche Modelle ergeben sich. Durch ein unterschiedliches Beschneiden eines vollständigen Baumes können also auf recht einfache Weise verschiedene Systeme trainiert werden, die unterschiedlich viele Subphonemmodelle aufweisen.

Experimente

In einer Serie von Experimenten wurde die Abhängigkeit der Wortfehlerrate von der Zahl der akustischen Parameter evaluiert. Dabei wurde die Zahl der Normalverteilungen pro Subphonemmodell konstant gehalten und die Anzahl der unterschiedlichen Subphonemmodelle durch verschiedene Tiefen des Entscheidungsbaumes variiert.

Als Trainingsstichprobe dienten 8 Sendungen 'tagesschau' aus dem Jahr 1996. Insgesamt wurden fünf unterschiedliche Systeme trainiert. Dabei wurde stets derselbe Entscheidungsbaum verwendet, der nur an unterschiedlichen Stellen abgeschnitten wurde. Das Beschneiden erfolgte dabei abhängig von der Anzahl der gesehenen Beispiele für das jeweilige Modell: Wurde eine bestimmte Zahl von Trainingsbeispielen nicht erreicht, wurde der Baum weiter beschnitten.

Für jeden der fünf unterschiedlich großen, resultierenden Modellsätze wurde danach eine LDA-Matrix auf den Trainingsdaten geschätzt (siehe Seite 33), die Parameter der Normalverteilungen durch eine kmeans-Ballung initialisiert und dann mit 5 Iterationen Viterbi-Training geschätzt. Tabelle 6.12 und Bild 6.2 zeigen die von den fünf resultierenden Erkennern erzielten Erkennungsergebnisse auf einer unabhängigen Teststichprobe, sowie die Anzahl der Datenframes, die zur Schätzung einer einzelnen Normalverteilung zur Verfügung standen.

Man erkennt, dass ein Optimum bei ca. 15-20 Datenframes pro Normalverteilung erreicht wird.

Anzahl der Parameter pro Subphonem

Aus Bild 6.2 geht hervor, dass mindestens 15-20 Datenvektoren benötigt werden, um eine Gaußverteilung robust schätzen zu können. Hält man die Gesamtzahl der Normalverteilungen konstant, existiert noch ein weiterer Freiheitsgrad: Die Anzahl der Normalverteilungen pro Mixtur, also pro Subphonem. Die Frage ist, ob eher mehr

Minimaler Count	Anzahl der Subphonemmodelle	Datenframes pro Normalverteilung	Fehlerrate
1500	334	62.2	24,5%
800	578	36.0	21,3%
400	1086	19.1	20,2%
200	2086	10.0	20,6%
100	3795	5.5	21,9%

Tabelle 6.12: Abhängigkeit der Fehlerrate von der Zahl der Parameter

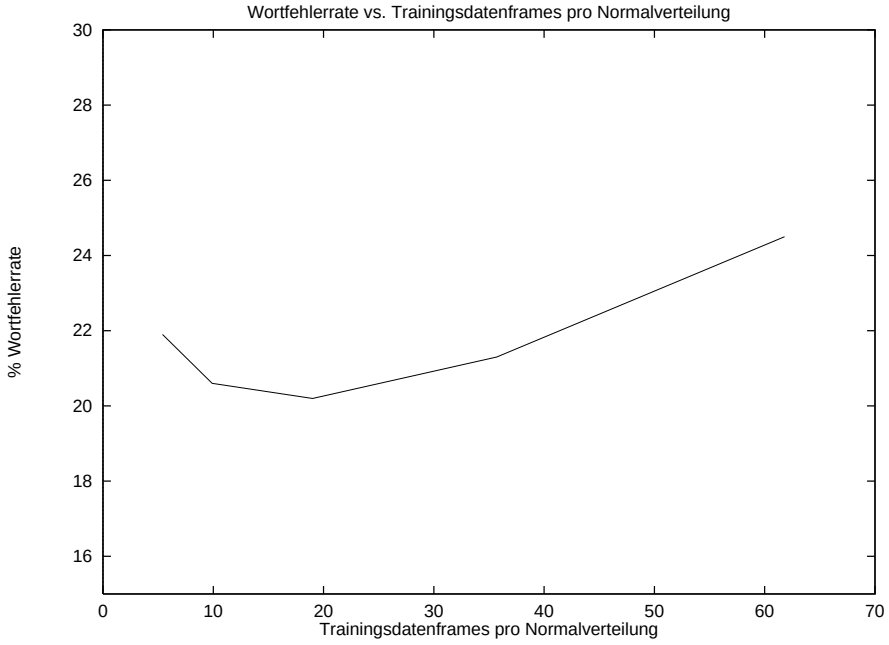


Abbildung 6.2: Abhängigkeit der Fehlerrate von der Zahl der Parameter

Subphonemmodelle mit weniger Normalverteilungen pro Mixtur, oder eher weniger, dafür akkurater modellierte Subphonemmodelle die bessere Performanz ergeben.

Um diese Frage zu klären, wurde ein weiterer Erkenner nach dem oben beschriebenen Schema trainiert. Dieser Erkenner verwendete genausoviele Subphonemmodelle wie das beste System in Tabelle 6.12 (1086), die Anzahl der Normalverteilungen pro Subphonemmodell war jedoch reduziert, so dass sich dieselbe Anzahl an Normalverteilungen ergab wie im System mit 578 Subphonemmodellen in Tabelle 6.12. Das Ergebnis ist in Tabelle 6.13 zusammengefaßt.
Die Fehlerraten der beiden Systeme sind im Rahmen statistischer Schwankungen identisch. Daraus kann geschlossen werden, dass der Spracherkenner nicht empfindlich auf unterschiedliche Allokation der Normalverteilungen reagiert.

In allen weiteren Experimenten wurde daher die Anzahl von 30 Normalverteilun-

Datenframes pro Normalverteilung	Anzahl Subphonemmodelle	Fehlerrate
36.0	578	21,3%
38.1	1086	21,4%

Tabelle 6.13: Abhängigkeit der Fehlerrate vom Ort der Parameterallokation

gen pro Mixtur festgehalten.

Obere Schranke der Zahl der Parameter

Die Zahl von 15 Datenframes pro Normalverteilung führt bei großen Trainingsdatenmengen rasch zu sehr großen Systemen. Diese sind für praktische Anwendungen zu langsam und zu speicherintensiv. Oberhalb einer bestimmten, durch die verfügbare Hardware gegebenen Schranke kann also die Größe des Parameterraumes aus technischen Gründen nicht mehr erhöht werden.

Die maximale Zahl an Normalverteilungen wurde für diese Arbeit zu 150000 festgelegt. Das entspricht gut 5000 Subphonemmodellen zu je 30 Normalverteilungen. Die Größe des Parameterraumes entspricht dabei derjenigen, die auch von (englischsprachigen) Gruppen für Evaluationserkennersysteme verwendet wird [19].

Zusammenfassung

Zusammenfassend läßt sich folgende Heuristik zur automatischen Parameterallokation eines autonom lernenden HMM-basierten Spracherkenners in der Domäne 'Nachrichtensendungen' angeben:

- Die Anzahl der Normalverteilungen pro Mixtur ist relativ unkritisch und kann einen konstanten Wert haben. Für das View4You-System wird dabei $N = 30$ gewählt.
- Die optimale Zahl der Subphonemmodelle hängt von der zur Verfügung stehenden Menge an Trainingsdaten ab. Solange keine Beschränkungen durch den zunehmenden Bedarf an Speicher und Rechenzeit bestehen, sollte sie so gewählt werden, dass für jede Normalverteilung 15-20 Datenframes zur Verfügung stehen (entsprechend 5-7 Normalverteilungen pro Sekunde Sprache).
- Die maximale Zahl der Subphonemmodelle hängt von der zur Verfügung stehenden Hardware ab. Im View4You-System wurden maximal 5000 unterschiedliche Subphonemmodelle trainiert.

6.8 Teilüberwachte Adaption

Der Unterschied in der Wortfehlerrate zwischen sprecherunabhängiger und sprecherspezifischer Spracherkennung ist deutlich. Ein System, das sorgfältig auf einen bestimmten Sprecher trainiert und abgestimmt ist, erzielt im Vergleich zu einem sprecherunabhängigen System eine um einen Faktor 2 niedrigere Fehlerrate. Es hat daher nicht an Versuchen gefehlt, durch Adaption des Erkenners an die gerade vorliegende Eingabe ein sprecherabhängiges System 'on demand' zu erzeugen und so die Fehlerrate zu reduzieren. Die erfolgreichsten Adaptionskonzepte hierbei sind VTLN (s. Seite 35) und MLLR (s. Seite 36).

Bei der MLLR werden die Modelle des Erkenners mit Hilfe einer vorliegenden Sprachprobe der Zielperson adaptiert. Dazu ist eine Referenz erforderlich, es muß also bekannt sein, was die Zielperson in der vorliegenden Sprachprobe tatsächlich gesagt hat. Stehen ausreichend viele solche Sprachproben zur Verfügung (mehrere Minuten bis wenige Stunden), kann ein sprecherunabhängiger Spracherkenner mit Hilfe von MLLR nahezu die Performanz eines sprecherabhängigen Systems erreichen [74]. Man spricht in diesem Fall von überwachter Adaption.

In der Praxis ist diese Situation allerdings selten. In der Regel liegt keine transliterierte Sprachprobe derjenigen Person vor, die gerade in das Mikrofon etwa eines Auskunftssystems spricht. Dennoch ist eine Adaption mit Hilfe eines Kunstgriffs möglich: An Stelle der Transkription des Gesagten wird eine Hypothese des (unadaptierten)

Spracherkenners verwendet. Im Gegensatz zur überwachten Adaption ist hierbei mit Fehlern in der Referenz zu rechnen, so dass die unüberwachte Adaption - je nach Anzahl der Fehler - die Worterkennungsrate sogar reduzieren kann. Unüberwachte Adaption wird daher häufig mit einem Konfidenzmaß kombiniert. Dieses beurteilt die Hypothese des Erkenners und zeigt vermutliche Erkennungsfehler an. Die Adaption wird dann auf den als 'vermutlich korrekt' bewerteten Bereichen berechnet.

Auch im View4You-Systems wird eine unüberwachte Adaption durchgeführt. Dabei werden Wörter, die vom Konfidenzmaß *gamma* (siehe Seite 127) eine niedrigere a-posteriori-Wahrscheinlichkeit als 0.5 zugewiesen bekommen, nicht berücksichtigt. Den Einfluss der (unüberwachten) Adaption auf die Fehlerrate summiert Tabelle 6.14.

System	Ansagesprecher	Korrespondent	gesamt
nicht adaptiert	12.4%	28.1%	21.8%
unüberwachte MLLR	11.9%	24.6%	19.5%

Tabelle 6.14: Einfluß der unüberwachten MLLR

Die Wortfehlerrate sinkt auf den Ansagesprechern kaum (0.5% absolut), hingegen deutlich bei den Korrespondenten (3.5% absolut).

Wie im Kapitel 'Segmentierer' gezeigt werden konnte, liegt mit dem Gish-Abstandsmaß eine sehr zuverlässige und schnelle Methode zur Berechnung der Ähnlichkeit zweier Sprachproben vor. Mit Hilfe dieses Abstandsmaßes ist es möglich, eine teilüberwachte Adaption durchzuführen. Die Grundidee der teilüberwachten Adaption ist es, wie bei der überwachten Adaption die transkribierten Aufnahmen der Trainingsstichprobe zu verwenden.

Allerdings kann dabei kein Wissen über die Identität des Sprechers der vorliegenden Sprachprobe bzw. der Identität des Sprechers in der Trainingsstichprobe vorausgesetzt werden. Dieses Wissen - beziehungsweise eine Hypothese darüber - wird mit Hilfe des Gish-Abstandsmaßes gewonnen.

Der Algorithmus der teilüberwachten Adaption, der im Rahmen dieser Arbeit entwickelt wurde, läßt sich wie folgt zusammenfassen:

- für alle Segmente der Teststichprobe:
    1. bestimme Abstand des Test-Segments zu allen Segmenten der Trainingsstichprobe
    2. bestimme die N kleinsten Abstände
    3. führe eine überwachte MLLR-Adaption auf den N ähnlichsten Segmenten der Trainingsstichprobe aus

4. benutze die adaptierten akustischen Modelle zu einem 'normalen' Erkennungslauf:
   - berechne VTLN-Adaption
   - berechne initiale Hypothese
   - führe eine unüberwachte MLLR-Adaption auf der Hypothese durch
   - berechne die finale Hypothese
5. restauriere die ursprünglichen akustischen Modelle für das nächste Segment

Bei der teilüberwachten Adaption nach diesem Schema wird also zwei Mal adaptiert. Zunächst werden die akustischen Modelle mit Hilfe der ähnlichen Segmente des Trainingssets verschoben, und danach werden die bereits adaptierten Modelle noch einmal mit einer normalen, unüberwachten MLLR zu den endgültigen Modellen transformiert. Zusätzlich findet eine VTLN statt, so dass das Schema insgesamt eine dreimalige Adaption darstellt.

Experimente

Der Algorithmus zur teilüberwachten Adaption wurde mit $N = 25$ auf der Teststichprobe evaluiert. Das Ergebnis ist in Tabelle 6.15 zusammengefasst. Die teilüberwachte Adaption wirkt besonders gut auf den Ansagesprechersegmenten, auf denen die unüberwachte Adaption nur wenig Verbesserung erzielen konnte. Hingegen konnte die teilüberwachte Adaption die Performanz auf den Korrespondentenberichten nicht mehr erhöhen.

System	Ansagesprecher	Korrespondent
nur VTLN	12,4%	28,1%
VTLN + normal (unüberwacht)	11,9%	24,6%
VTLN + teilüberwacht + unüberwacht	11,1%	24,6%

Tabelle 6.15: Auswirkung der teilüberwachten Adaption auf die Wortfehlerrate

6.9 Vokabularadaption

Die meisten kommerziellen Applikationen, die Spracherkennung verwenden, beschränken sich auf eine relativ überschaubare Domäne. Beispiele für solche Applikationen sind 'hands-free' Anwendungen im medizinischen Bereich, Sprachsteuerung für elektronische Haushaltsgeräte oder solche für KFZ-Anwendungen zur Steuerung des Telefons und des Radios. In diesen Applikationen ist der Umfang des verwendeten Vokabulars klein. Die zu erwartenden akustischen Bedingungen sind jedoch starken

Schwankungen unterworfen und die Qualität der Aufnahme kann sehr schlecht sein (z.B. im KFZ). In einem anderen Typ von Applikation, wie in einem Diktiersystem, das Radiologen das Diktieren von Befunden erlaubt, ist das Vokabular groß und die Qualität der Aufnahme sehr gut. Obwohl das beim Diktieren solcher Befunde verwendete Vokabular groß ist und eine Menge von Fachwörtern umfasst, ist es kleiner als der Wortumfang der Sprache insgesamt (mit ca. 20000 Wörtern ist in der Radiologie-Domäne bereits eine sehr gute Abdeckung erreicht), und es ist *statisch*. Man kann daher mit einem fixen Vokabular nicht nur die Wörter der Radiologiebefunde von heute, sondern auch die Befunde des nächsten Jahres abdecken.

Bei der Indexierung von Nachrichtensendungen unterliegt das vom Sprecher verwendete Vokabular keinen Einschränkungen. Eine gute Vokabularabdeckung ist damit erst bei sehr großen Vokabulargrößen zu erwarten. Diesem Problem kann man mit adaptiven Vokabularien [2] [11] begegnen.

Um das Problem zu quantifizieren, wurden die Sprachmodelldaten des View4You-Erkenners verwendet, um eine Liste der Wörter mit den absoluten Häufigkeiten ihres Vorkommens zu berechnen. Die häufigsten N Wörter aus dieser Liste wurden dann extrahiert und für jedes N die Vokabularabdeckung auf fünf nicht im Sprachmodelltext enthaltenen Tagesschauen berechnet. Das Ergebnis ist in Tabelle 6.16 und Bild 6.3 in zwei verschiedenen Darstellungsweisen zusammengefaßt. Man sieht, dass die Vokabularabdeckung bei einer Vokabulargröße von 20000 Wörtern lediglich 89% beträgt, d.h. jedes neunte Wort unbekannt ist. Bei 60000 Wörtern Vokabular liegt die Rate der unbekannten Wörter immer noch über 4%. Um eine Vokabularabdeckung von mehr als 99% zu erreichen, ist ein Vokabular von mehr als 800.000 Wörtern erforderlich.

Unbekannte Wörter der Tagesschau vom 28.5.1997

Zur Illustration sind die unbekannten Wörter aus einer Tagesschau im Folgenden angegeben. Von den nicht abgedeckten Wörtern lassen sich zwei große Klassen abspalten: Die offene Klasse der Zahlworte, sowie die Klasse der Namen und Ortsbezeichnungen.

- Zahlwörter (19)
  dreiundzwanzig fünfundachtzig achtzehnstündige zweitausendfünfzig zweihundertzweiundneunzig dreiundzwanzig vierundfünfzigjährige neunzehnhundertachtzig neunundzwanzig hundertachtzig dreiunddreißig neununddreißig vierundvierzig siebenundzwanzig dreiundvierzig achtundvierzig siebenundsiebzig neunundzwanzigsten dreiundzwanzig

- Namen und Ortsbezeichnungen (11)
  Masar Scharif Malik Maliks Tshisekedi Kabilas Rajlovac Garoll Mauresmo Kandarr Knippschild

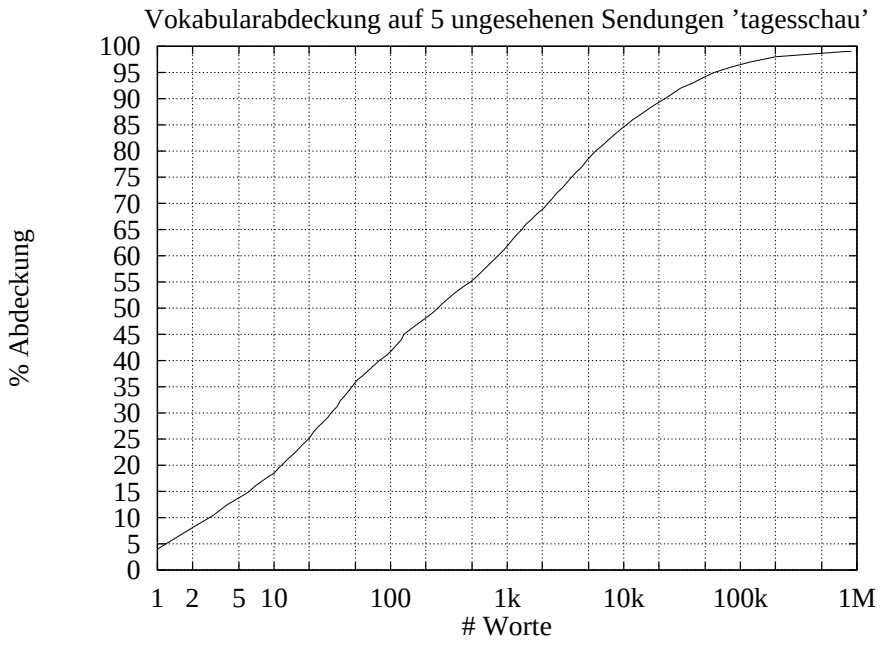


Abbildung 6.3: Vokabularabdeckung in der Domäne Fernsehnachrichtensendungen

- Andere (48)
  Weltwirtschaftsordnung Milliardenkredit Kredites Erdgasleitung Islamisierungspolitik Massenkundgebung Weltbevölkerungsbericht Gesundheitsbereich Trauernde Hauptgefreite Feldlager Bordkanone Panzerwagens Ehrenzeremonie Lösegeldes umzutauschen verbüßte Reisebusses Verkaufswert Rauschgiftes Durchleuchten Fahrzeuges Freihafen Wirbelstürmen pflückte Betonfundamenten saugte sauste braust Windwalze glättete angsterfüllte Luftstrom gesonnenen zerriss Suchtrupps Leichenwagen Rettungshelfer Zentral Superzahlen Gewinnzahl ziehend Osthälfte Quellwolken Nordwind Südhälfte frischt wärmsten

Kanonisierung

Ein signifikanter Teil der unbekannten Wörter läßt sich automatisch durch eine Textnormalisierung in bekannte Worte verwandeln: Zahlen können kanonisiert und Nominalkomposita in ihre Einzelteile zerlegt werden. Dies kann automatisch anhand eines großen Wörterbuchs erfolgen.

Kanonisierung von Zahlen

Die gesprochene Zahl 'einhunderteinundsechzig' kann vom Erkenner - vorausgesetzt, die entsprechenden Einheiten sind im Vokabular enthalten - auf vielerlei Weisen erkannt werden:

Abdeckung	nötige Vokabulargröße
10%	3
20%	12
30%	31
40%	81
50%	254
60%	838
70%	2242
80%	5729
85%	10532
90%	22588
92%	30763
94%	47997
95%	59635
96%	82729
97%	122364
98%	201552
99%	807468

Tabelle 6.16: Für eine gegebene Abdeckung erforderliche Vokabulargröße

- ein hundert ein und sechzig
- einhundert ein und sechzig
- einhundert 61
- 100 einundsechzig
- 161
- 100 ein und 60

sind nur einige der Möglichkeiten. Daher wird in der Kanonisierung jede Zahl in ihre kleinstmöglichen Einheiten aufgebrochen. '161' wird damit zu 'ein hundert ein und sechzig'.

Auf den 4 Tagesschauen der Teststichprobe senkt diese Maßnahme die OOV-Rate von 4,44% auf 3,61% (19% relativ). Die Gesamtzahl der Wörter erhöht sich bei der Kanonisierung, da Zahlworte in kleinere Einheiten zerlegt werden.

Insgesamt steigt die Zahl der Wörter um 2% von 8554 auf 8741.

Kompositazerlegung

Es ist eine Eigentümlichkeit der deutschen Sprache, dass Komposita, namentlich Nominalkomposita, quasi beliebig neu gebildet werden können. Das wohl bekannteste

Beispiel für dieses Phänomen ist der 'Donaudampfschifffahrtsgesellschaftskapitän'. Die Nominalkomposita stellen damit im Prinzip eine offene Klasse von Wörtern dar, so dass es nicht möglich ist, eine vollständige Abdeckung der Nominalkomposita mit einem statischen Wörterbuch zu erhalten. Eine weitere Komplikation entsteht durch die orthographische Freiheit, ein Kompositum mit Bindestrich oder als ein Wort zu schreiben. So sind 'Palästinenser-Präsident' und 'Palästinenserpräsident' zwei legale Schreibweisen für dasselbe Kompositum. Diese Freiheit wirkt sich bei seltenen Komposita negativ auf die Performanz des Sprachmodells aus. Bei Komposita aus häufigen Konstituenten gewinnt das Sprachmodell hingegen an Diskriminierungskapazität, sofern auch das Kompositum selbst häufig genug auftritt.

Zwangsläufig sind aber viele Nominalkomposita (z.B. 'Leichenwagen') nicht im Wörterbuch enthalten. Viele von ihnen bestehen aber aus bekannten - d.h. im Wörterbuch enthaltenen - Konstituenten. Es liegt nahe, in solchen Fällen das Kompositum in seine Konstituenten zu zerlegen. Sowohl aus 'Palästinenser-Präsident' als auch aus 'Palästinenserpräsident' wird auf diese Weise 'Palästinenser Präsident'. Diese Zerlegung hilft, die Vokabularabdeckung zu verbessern, und sie reduziert auch die gemessene Wortfehlerrate deutlich. Lautet zum Beispiel die Referenz 'Palästinenser-Präsident' und die Hypothese 'Palästinenser Präsident', so werden ohne Kompositazerlegung zwei Fehler ermittelt (eine Substitution und eine Einfügung). Nach der Kompositazerlegung von Referenz und Hypothese liegt kein Fehler mehr vor.

Es ist zu beachten, daß die Kompositazerlegung auf der textuellen Ebene der Hypothesen durchgeführt wird und damit eine reine Nachbearbeitung der Erkennungsergebnisse darstellt. Am Sprachmodell oder an den akustischen Modellen des Erkenners wird keine Änderung vorgenommen.

Auf 4 Tagesschauen der Teststichprobe senkt die Nominalkompositazerlegung die OOV-Rate von 3,61% auf 2,32% (36% relativ) bei einer gleichzeitigen Zunahme der Zahl der Wörter um 8% (von 8741 auf 9485).

Einfluss der Kanonisierung auf die gemessene Wortfehlerrate

Die Kompositazerlegung und die Kanonisierung der Zahlen reduzieren insgesamt die OOV-Rate auf der Teststichprobe von 4,44% auf 2,32%. Es ist daher ein deutlicher Effekt auf die Wortfehlerrate zu erwarten.

Tabelle 6.17 fasst den Einfluss der Kanonisierung auf die gemessene Wortfehlerrate zusammen. Die Kanonisierung reduziert die gemessene Wortfehlerrate um fast 5% absolut. Dabei ist festzuhalten, dass diese Verbesserung ausschließlich durch Normalisierung der fertigen Erkennerhypothese, d.h. ohne Änderung des eigentlichen Erkenners - inklusive des Sprachmodells - erreicht wird.

Im Rahmen dieser Arbeit werden daher alle Ergebnisse, soweit nicht explizit anders gekennzeichnet, auf kanonisierten Referenzen und Hypothesen angegeben.

Kanonisierung	OOV-Rate	Fehlerrate Ansager	Fehlerrate Korrespondent	Fehlerrate insgesamt
ohne	4,44%	17,2%	29,3%	24.4%
mit	2,32%	11,9%	24,6%	19.5%

Tabelle 6.17: Einfluss der Kanonisierung

Ursachen der mangelhaften Vokabularabdeckung

Ein wichtiges Problem, das die Verwendung von automatischer Spracherkennung für Fernsehnachrichtensendungen schwierig macht, ist der tägliche Themenwechsel und die damit verbundene Variation des aktiven Vokabulars. Diese Änderung besteht aus zwei Komponenten; einer sehr kurzfristigen und einer langfristigen.

Die langfristige Komponente ergibt sich aus Veränderungen auf der politischen und weltanschaulichen Landkarte. So wird das Wort 'Chruschtschow', das in den fünfziger und sechziger Jahren recht häufig vorkam, heute (beinahe) nicht mehr benötigt. Auf einer ähnlich langen Zeitskala wechseln Wörter wie 'NATO-Doppelbeschluss' oder 'Elvis Presley' ihre Häufigkeitsverteilung. Rascher - aber immer noch in Zeiträumen von mehreren Jahren - wechseln Namen von demokratisch gewählten Staatsoberhäuptern ('Ronald Reagan').

Die kurzfristige Komponente ergibt sich aus tagesaktuellen Nachrichten. Es wird aus Orten berichtet, an denen Erwähnenswertes vorgefallen ist, wobei Ortsnamen und Namen von Lokalpolitikern relevant sind. Solche Wörter sind nur wenige Tage, viele sogar nur einen einzigen Tag lang prominent.

Um den Einfluss der langfristigen wie der kurzfristigen Komponente zu ermitteln, wurde folgende Untersuchung durchgeführt.

Aus dem Sprachmodellkorpus (vgl. Seite 73), der Zeitungstexte der Jahre 1992 bis 1996 sowie geringere Textmengen aus dem Internet aus dem Jahr 1996 und den Monaten Januar und Februar 1997 enthält, wurden die häufigsten 60782 Wörter als Vokabular gewählt. Auf Transkripten von 'tagesschau'-Sendungen kann mit Hilfe dieses Vokabulars die jeweilige OOV-Rate ermittelt werden. Die für verschiedene Ausstrahlungsdaten von Sendungen ermittelten OOV-Raten (jeweils nach der Kanonisierung) sind in Tabelle 6.18 und Bild 6.4 zusammengefasst.

Der Verlauf der Kurve ist überraschend. Im Rahmen der statistischen Schwankungen zeigt die Vokabularabdeckung keine erkennbare Abhängigkeit vom Ausstrahlungsdatum. Das impliziert, dass die langfristige Komponente der Vokabularschwankungen im beobachteten Zeitraum von knapp über zwei Jahren noch keine merkliche Rolle zu spielen scheint. In welcher Größenordnung sich die langfristige Komponente nach längeren Zeiträumen als zwei Jahren bemerkbar macht, lässt sich mit dem zur Zeit verfügbaren Datenmaterial leider (noch) nicht klären.

Die OOV-Rate im 4. Quartal 1996, ein noch von den Texten des Sprachmodelles

Datum	OOV-Rate
Q4/96	2.77%
Q1/97	2.59%
Q3/97	2.35%
Q1/98	2.44%
Q3/98	2.54%
Q1/99	2.45%

Tabelle 6.18: Abhängigkeit der OOV-Rate vom Sendedatum (Vokabular berechnet auf Daten 1992-1996)

abgedeckter Zeitraum, liegt nicht niedriger als die OOV-Rate in den Zeiträumen, von denen keine Texte vorhanden sind. Das lässt sich so interpretieren, dass die kurzfristigen Variationen in der Verwendung des Vokabulars die ganz überwiegende Ursache für die OOV-Rate darstellen. Diese kurzfristigen Variationen setzen sich ihrerseits zusammen aus den Variationen der deutschen Sprache als solcher und aus der angesprochenen kurzfristigen Komponente des themenabhängigen Teils des Vokabulars. Demgegenüber scheint der Unterschied zwischen den zum Training verwendeten Texten und den später ausgestrahlten Sendungen, sowie die langfristige Komponente der Vokabularschwankung keine oder keine signifikante Rolle zu spielen.

Dies wird auch durch einen Blick auf die - exemplarische - Liste der unbekannten Wörter der Tagesschau vom 28.5.97 (s. S. 88) bestätigt. Die meisten unbekannten Wörter sind 'normale' Wörter des Deutschen. Der Anteil dieser Wörter ist nicht (stark) vom Inhalt der Sendung, und damit vom Ausstrahlungsdatum, abhängig. An zweiter Stelle folgen Namen von Politikern und Orten, die nur an diesem einen Tag von Bedeutung waren, die also der obengenannten extrem kurzfristigen Variation zuzurechnen sind.

Adaptive Vokabularien

Eine Faustregel der maschinellen Spracherkennung besagt, dass jedes unbekannte Wort in der Teststichprobe $1 - 1,5$ Fehler in der Hypothese verursacht. Bei etwa 2,5% unbekannten Wörtern (nach der Kanonisierung) sind also etwa 3% (absolut) Wortfehlerrate auf die unbekannten Wörter zurückzuführen. Bei einer Gesamtfehlerrate von etwa 12% auf den Ansagesprechern ist also ein Viertel aller Fehler durch die ungenügende Vokabularabdeckung verursacht.

Es hat nicht an Versuchen gefehlt, die Vokabularabdeckung von Spracherkennern zu verbessern. Die einfachste Lösung wäre, die Vokabulargröße zu erweitern. Auf den 4 Tagesschauen der Teststichprobe wird die OOV-Rate durch eine Verdopplung der Vokabulargröße von 4,44% auf 2,82% oder um 37% relativ gesenkt. Es sind - abgese-

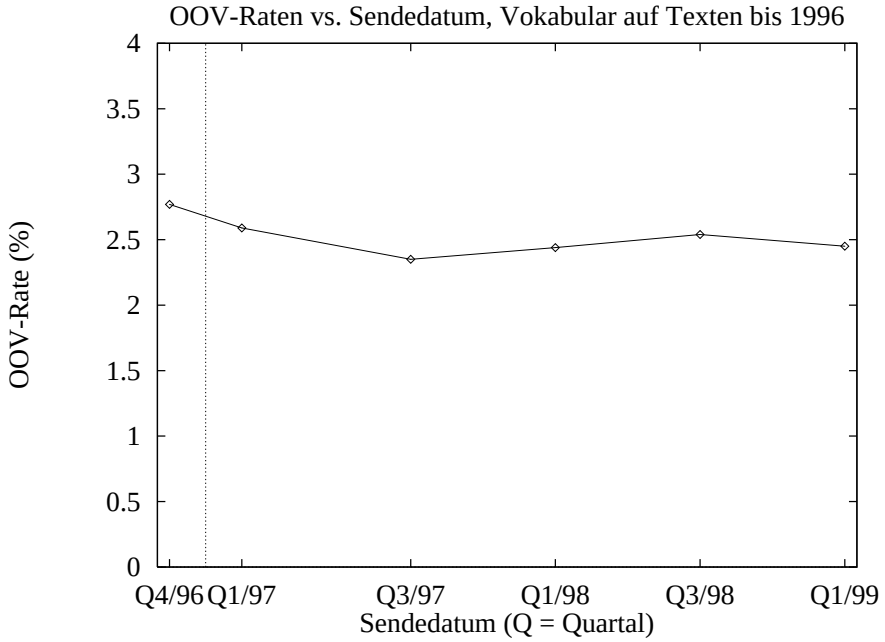


Abbildung 6.4: OOV-Raten über Ausstrahlungsdatum

hen von der Schwierigkeit, für die neuen, selten gesehenen Wörter noch verlässliche Sprachmodellwahrscheinlichkeiten zu schätzen - vor allem technische Gründe, die dem entgegenstehen. Die Indizierung eines Vokabulars von 60000 Wörtern ist noch mit zwei Byte möglich ($2^{16} = 65536$), und nahezu alle wichtigen Compiler für Programmiersprachen stellen Datentypen mit zwei Byte zur Verfügung. Wird diese Grenze überschritten, ist der nächstgrößere Datentyp typischerweise mit 4 Byte doppelt so speicheraufwendig, so dass der Gesamtspeicherbedarf des Erkenners an dieser Stelle sprunghaft anwächst.

Mit zunehmender Leistungsfähigkeit der Rechner wird das Speicherproblem, das ein vergrößerter Wortschatz mit sich bringt, an Bedeutung verlieren. Das Problem der robusten Schätzung der Sprachmodellparameter wird sich allerdings weiter verschärfen, da die Zahl der zu schätzenden Bigramme quadratisch und die der Trigramme sogar kubisch mit der Vokabulargröße ansteigt.

Im Rahmen der vorgestellten Arbeit war durch die Architektur des Spracherkenners sowie durch die Begrenzung des Hauptspeichers der verwendeten Workstations die Vokabulargröße auf $2^{16} - 2$ (65534) Worte begrenzt.

Vorarbeiten anderer Autoren

Geutner befasst sich in [11] und [2] ausführlich mit dem Thema Vokabularadaption. Die Grundidee bei ihren Arbeiten ist, anhand der Hypothese eines Erkenners mit generischem Vokabular ein neues, adaptiertes Vokabular zu bestimmen, mit dem

dann ein zweiter Erkennerdurchlauf gestartet wird. Die Wörter des neuen Vokabulars werden dabei einem großen, bestehenden Hintergrundwörterbuch entnommen. Dabei wird angenommen, dass für die Wörter des Hintergrundwörterbuchs ausreichend viele passende Textdaten zur Berechnung eines Sprachmodells zur Verfügung stehen.

Um die Selektion des Vokabulars für den zweiten Suchdurchlauf durchzuführen, geht Geutner von den im ersten Suchdurchlauf gefundenen Wörtern aus und schließt von diesen auf das geeignete Vokabular für den zweiten Suchdurchlauf. In [2] untersucht sie sowohl einen morphologischen Ansatz, bei dem die Stämme der Wörter der Hypothese der initialen Hypothese mit allen möglichen Endungen ergänzt werden, als auch einen ähnlichkeitsbasierten Ansatz, bei dem alle Wörter des Hintergrundwörterbuchs mit den Wörtern der Hypothese verglichen werden und die phonetisch ähnlichsten das neue Vokabular bilden. Die Ergebnisse, die mit diesen Ansätzen erzielt werden, sind je nach Sprache und Domäne unterschiedlich. Insgesamt konnte jedoch eine signifikante Reduktion der OOV- und der Wortfehlerraten erreicht werden.

Internetbasierte Ansätze

Im View4You-System werden zur Ergänzung der Datenbasis täglich Zeitungstexte von verschiedenen Quellen aus dem Internet gesammelt. Diese Zeitungstexte umfassen zum Großteil tagesaktuelle Nachrichten. Es ist also sehr wahrscheinlich, dass sich eine Nachricht, die abends in der 'tagesschau' gesendet wird, am selben Tag oder am nächsten Tag in der Zeitung wiederfindet. Dabei werden in den Zeitungstexten mit relativ hoher Wahrscheinlichkeit dieselben Orts- und Personennamen verwendet wie in der 'tagesschau'. Es sollte mithin möglich sein, mit Hilfe dieser Zeitungstexte die Vokabularabdeckung zu verbessern - und das speziell bei den Wörtern, die durch die kurzfristige Variation des verwendeten Vokabulars hinzukommen.

Das ist die Grundidee der folgenden beiden, im Rahmen dieser Arbeit entwickelten Verfahren. Bei beiden Ansätzen werden aus der Gesamtmenge der vom Internet geladenen Zeitungstexte 'vermutlich relevante' Artikel herausgesucht. Aus diesen wird ein neues Vokabular berechnet, das dann in einem zweiten, endgültigen Erkennungsdurchlauf eingesetzt wird.

Vokabularaustausch

Ausgangspunkt der Berechnung des neuen Vokabulars ist der große Hintergrundkorpus $K_{Hintergrund}$, der auch zur Berechnung des Sprachmodells herangezogen wurde, und eine Menge von - in der Summe viel weniger Text enthaltenden - Artikeln A_n, die noch mit einer Relevanz α_n versehen sein können. Diese Artikel stammen aus den aktuellen Nachrichtentexten aus dem Internet. Sie bilden zusammen den Adaptionskorpus $K_{aktuell}$.

Die Aufgabe ist es nun, aus dem Hintergrundkorpus und den neuen Artikeln ein gemeinsames Vokabular zu berechnen.

Der hierzu verwendete Algorithmus läßt sich wie folgt zusammenfassen.

1. Zähle für alle Wörter i ($i \in I$) aus $K_{Hintergrund}$ die absolute Häufigkeit ihres Auftretens $N_h(i)$ in $K_{Hintergrund}$.

2. Zähle für alle Wörter j ($j \in J$) aus $K_{aktuell}$ die absolute Häufigkeit ihres Auftretens $N_a(j)$ in $K_{aktuell}$. Gewichte dabei jedes Auftreten mit der Relevanz α_n des Artikels, in dem das Wort j auftrat.

3. Berechne für alle Wörter w ($w \in I \cup J$) ihre gewichtete Häufigkeit $N_w(w) = N_h(w) + \gamma * N_a(w)$.

4. Sortiere $N_w(w)$ und selektiere die (z.B.) 60000 Wörter mit der höchsten gewichteten Häufigkeit.

Durch Anwendung dieses Algorithmus' wird das - auf dem Hintergrundkorpus berechnete - Grundvokabular adaptiert, indem Wörter des Hintergrundkorpus durch solche des Adaptionskorpus ersetzt werden. Natürlich kann durch diese Vorgehensweise die OOV-Rate auf einer Teststichprobe auch ansteigen: Wenn relevante Wörter des Hintergrundkorpus durch irrelevante Wörter des Adaptionskorpus ersetzt werden. Außerdem berücksichtigt der Algorithmus bei der Berechnung des neuen Vokabulars die Häufigkeiten eines Wortes in beiden Korpora. Es ist also durchaus nicht so, dass zwangsläufig das seltenste Wort im Grundvokabular durch das häufigste, nicht im Grundvokabular enthaltene Wort des Adaptionskorpus ersetzt wird.

Im einfachsten Fall sind die Relevanzen der Artikel α_n alle gleich 1. In diesem Fall bleibt als freier Parameter γ, ein Gewichtungsfaktor für den Adaptionskorpus.

Wählt man $\gamma = 0$, so entspricht das Vokabular genau dem des Hintergrundkorpus, der Adaptionskorpus spielt keine Rolle. Für $\gamma \to \infty$ wird das neue Vokabular aus den Wörtern des Adaptionskorpus gebildet. Nur wenn dieser weniger als 60000 unterschiedliche Wörter enthält, werden noch Wörter aus dem Hintergrundkorpus hinzugenommen.

Mit Hilfe von γ kann also eine stufenlose Gewichtung des Adaptionskorpus vorgenommen werden. Verfügt man über weitere Informationen zur Bestimmung der Relevanz eines einzelnen Artikel n, kann die Relevanz durch Verändern des Gewichtungsfaktors α_n angepasst werden.

Ausnutzen der zeitlichen Nähe

Es kann angenommen werden, dass Texte, die in geringer zeitlicher Distanz zu einer gegebenen Nachrichtensendung entstanden, eine größere Relevanz im Hinblick auf das verwendete Vokabular aufweisen. In einer Reihe von Experimenten wurden daher die Internet-Zeitungstexte, die im zeitlichen Abstand von weniger als einem Monat zu

einer gegebenen Sendung entstanden, als Adaptionskorpus für das Vokabular dieser Sendung verwendet.

In einer ersten Testreihe wurde festgestellt, wie groß der Interpolationsfaktor γ sein sollte. Um von der konkreten Größe der beiden Korpora etwas zu abstrahieren, werden die Ergebnisse nicht über γ, sondern über der Zahl der durch die Adaption ausgetauschten Wörter angegeben.

0	1k	2k	3k	4k	5k	6k	7k	8k	10k	15k	60k
4.44	4.09	3.96	3.76	3.75	3.69	3.53	3.42	3.48	3.53	3.65	4.09

Tabelle 6.19: OOV-Rate über Zahl der ersetzten Wörter

Daraus geht hervor, dass ein Austausch von 10% des Vokabulars die besten Ergebnisse erbringt.

Der optimale Interpolationsparameter γ und auch die Zahl der auszutauschenden Wörter ist nicht für jede Sendung gleich. Es ist besser, γ auf jeder einzelnen Sendung neu zu bestimmen. Ein Algorithmus, der das automatisch leistet, ist jedoch nicht leicht anzugeben. Daher wurde in einem Kontrollexperiment berechnet, wie stark die OOV-Rate reduziert werden könnte, wenn es einen optimal funktionierenden solchen Algorithmus gäbe - ein Experiment, das nur mit Kenntnis der Referenz möglich ist (*cheating experiment*). Das Ergebnis ist in Tabelle 6.20 angegeben. Die OOV-Rate sinkt im Vergleich zum Ersetzen nur noch um weniger als 0,2% absolut, so dass sich der Entwicklungsaufwand für den angesprochenen Algorithmus nicht auszahlt.

Algorithmus	OOV-Rate
6k Wörter ersetzen	3,53%
optimales γ (cheating!)	3,34%

Tabelle 6.20: OOV-Rate bei optimalem Interpolationsfaktor

Information-retrieval (IR) basierte Vokabularadaption

Im View4You-System steht mit der Datenbank-Komponente ein Modul zur Verfügung, das Ähnlichkeiten zwischen zwei Texten ermitteln kann und auf eine Anfrage (Text 1) eine nach Ähnlichkeit sortierte Liste von Texten zurückliefert (Details zur Datenbank finden sich ab Seite 102).
Die Grundidee der Information-retrieval (IR) basierten Vokabularadaption ist es nun, die initiale Hypothese des Erkenners, die mit einem generischen Vokabular berechnet wurde, als Anfrage an die Datenbank des View4You-Systems selbst zu richten. Dazu werden die vom Internet bezogenen Nachrichten-Artikel zuvor in die Datenbank eingespeist.

Die Datenbank liefert einen Satz von ähnlichen Artikeln zurück, von denen jeder mit einer Relevanz annotiert ist. Je größer die Relevanz ist, desto ähnlicher ist der

entsprechende Artikel. Die eigentliche Vokabularadaption geschieht wieder mit dem Algorithmus von Seite 96.

Die von der Datenbank berechneten Relevanzen der einzelnen Artikel dienen dabei als Gewichtungsfaktoren α_i. Der freie Parameter γ wird so eingestellt, dass insgesamt 10% aller Wörter des Basisvokabulars ersetzt werden. Die Vokabularadaption durch Information-retrieval (IR) läßt sich also summarisch wie folgt darstellen:

1. Alle verfügbaren Artikel aus dem Internet werden in die Datenbank eingespeichert.
2. Die zu erkennende Sendung wird segmentiert.
3. Ein initialer Erkennungslauf auf allen Segmenten der zu erkennenden Sendung 'tagesschau' wird durchgeführt, wobei das Hintergrundvokabular verwendet wird.
4. Die Hypothesen werden als Anfragen an die Datenbank gerichtet.
5. Sämtliche von der Datenbank gelieferten Artikel werden als Adaptionskorpus K_{neu} definiert. Jeder Artikel erhält ein Gewicht α_i, das der von der Datenbank gelieferten Relevanz entspricht.
6. Das neue Vokabular wird mit Hilfe des Algorithmus zur Vokabularadaption (s. Seite 96) berechnet. Der Parameter γ wird so eingestellt, dass 10% aller Wörter des Hintergrundvokabulars ausgetauscht werden.

Dieser Algorithmus weist gegenüber dem im vorhergehenden Abschnitt beschriebenen drei prinzipielle Vorteile auf:

- Relevante Artikel, die nicht in den betrachteten Zeitraum fallen, werden berücksichtigt.
- Irrelevante Artikel des betrachteten Zeitraumes werden nicht berücksichtigt.
- Jeder Artikel wird abhängig von seiner Relevanz unterschiedlich stark berücksichtigt.

Die Vorteile werden allerdings durch die Notwendigkeit eines zusätzlichen Erkennerlaufes 'erkauft'.

Algorithmus	OOV-Rate
keiner	4,44%
zeitnah	3,53%
zeitnah optimal (cheating!)	3,34%
information retrieval	3,16%

Tabelle 6.21: OOV-Raten bei verschiedenen Algorithmen

Experimente

Die Vokabularabdeckung auf der Teststichprobe von 4 Sendungen 'tagesschau' (vom 30.3., 13.4., 28.5. und 30.6.97) ist in Tabelle 6.21 zusammengefaßt. Dabei wurde das Ergebnis der Zeile 'information retrieval' mit dem im vorangehenden Abschnitt beschriebenen Algorithmus erzielt.

Das auf information retrieval basierende Verfahren erzielt die größte Reduktion der OOV-Rate.

Bild 6.5 zeigt die Abhängigkeit der erzielten OOV-Reduktion von der Zahl der ersetzten Wörter für beide konkurrierende Verfahren exemplarisch auf. Man erkennt, dass für beide Verfahren ein zu starkes Gewichten des Adaptionskorpus schädlich ist. Wie zu erwarten war, enthält der Adaptionskorpus bei der Methode 'zeitnah' mehr irrelevante Wörter, so dass der Abfall der Abdeckung bei diesem Verfahren früher beginnt und drastischer ausfällt.

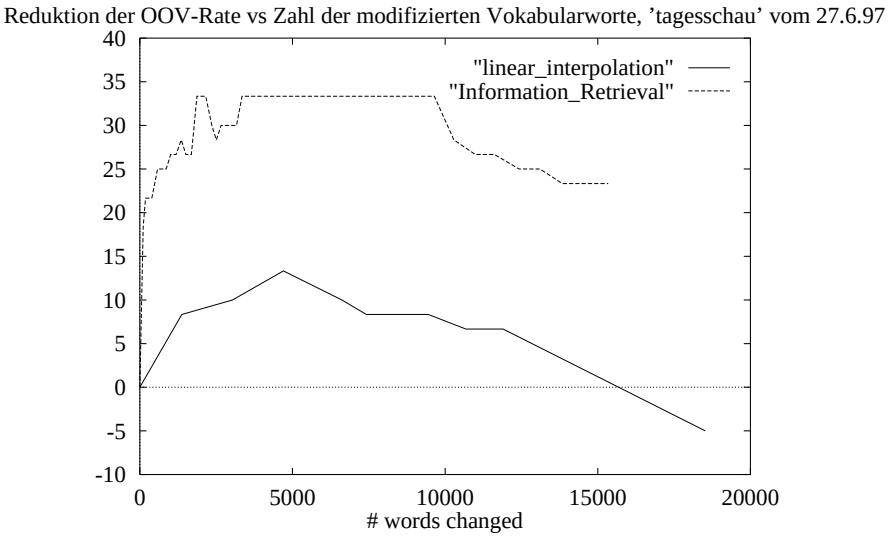


Abbildung 6.5: OOV-Raten über Zahl der ausgetauschten Wörter

Mehrere Vokabularien pro Sendung

Durch den Segmentierer wird jede Sendung in durchschnittlich etwa 40 Segmente aufgeteilt. Es ist zu erwarten, dass das Berechnen von unterschiedlichen Vokabularien für jedes der Segmente zu einer besseren durchschnittlichen Vokabularabdeckung führt, als ein globales Vokabular für alle Segmente zusammengenommen.

Ein Kontrollexperiment auf einer Tagesschau ergab jedoch, dass die OOV-Rate auch bei Verwendung eines Vokabulars pro Segment (anstelle eines Vokabulars pro Sendung) nur um 0,14% absolut abnimmt. Da der Aufwand für die Erkennung jedoch durch ein solches Vorgehen um einen Faktor 40 zunimmt, kann geschlossen werden, dass die Verwendung eines Vokabulars pro Sendung für praktische Belange ausreichend genau ist.

Erkennungsergebnisse

Hat man mit Hilfe eines der beschriebenen Algorithmen ein neues Vokabular und - auf dem Hintergrundkorpus - ein neues Sprachmodell berechnet, kann der zweite Erkennerdurchlauf erfolgen. In diesem zweiten Durchlauf ist die OOV-Rate niedriger als beim ersten Durchlauf mit dem generischen Vokabular, so dass eine niedrigere Fehlerrate resultieren sollte.

Es ist zusätzlich noch möglich, ein zweites Sprachmodell auf dem Adaptionskorpus zu bestimmen. Dieses wird zwar auf deutlich weniger Daten berechnet als das hintergrundkorpusbasierte Standard-Sprachmodell, es enthält aber die Kontexte der neuen Wörter, die im Hintergrundkorpus ja schlecht (oder gar nicht) abgedeckt sind. Das zweite Sprachmodell kann mit dem Standard-Sprachmodell interpoliert werden und ergibt so ein Sprachmodell für die Erkennung. Aus Geschwindigkeitsgründen wurde dabei das interpolierte Sprachmodell nur zum Bewerten der Worthypothesengraphen des Erkenners verwendet.

Die besten Ergebnisse sind in Tabelle 6.22 aufgeführt, die Fehlerrate über dem Interpolationsparameter des Sprachmodells zeigt Bild 6.6. Ein Interpolationsparameter von 1.0 entspricht dabei der (alleinigen) Verwendung des Hintergrund-Sprachmodells; ein Interpolationsparameter von 0 entspricht der alleinigen Verwendung des adaptierten Sprachmodells.

Algorithmus	OOV-Rate	WER
keiner	4,44%	19,47%
information retrieval (IR)	3,16%	19,00%
IR plus LM-Interpolation	3,16%	18,41%

Tabelle 6.22: Wortfehlerraten bei Vokabularadaption

Insgesamt senkt die Vokabularadaption die Fehlerrate um knapp über 5% relativ

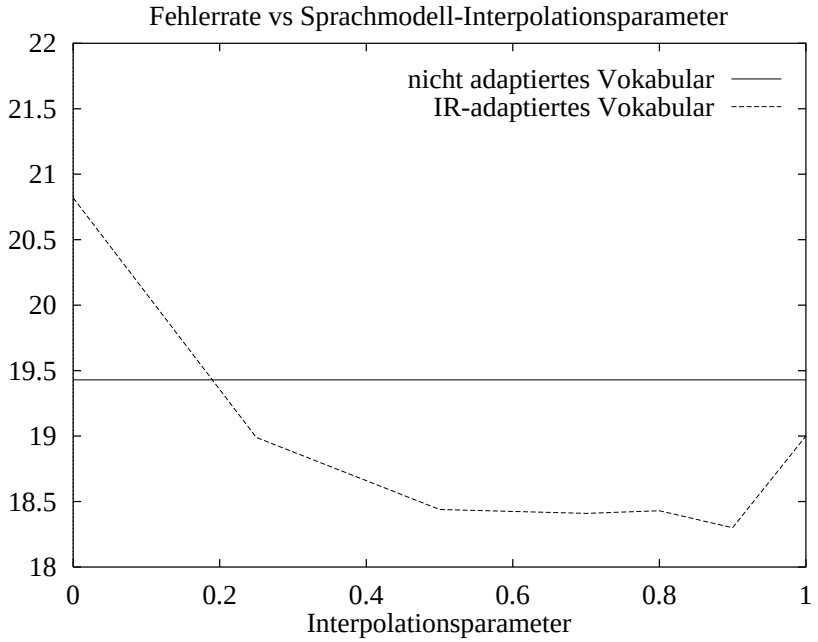


Abbildung 6.6: Wortfehlerrate über LM-Interpolationsgewicht

(1% absolut). Etwa die Hälfte der Verbesserung ist auf die verbesserte Vokabularabdeckung und die andere Hälfte auf das verbesserte Sprachmodell zurückzuführen.

Kapitel 7

Datenbank und Informationssystem des View4You-Systems

Die Aufgabe des Informationssystems ist es, die aufgenommenen und vom Spracherkenner transkribierten 'tagesschau'-Sendungen segmentweise zu archivieren und eine Abfragemöglichkeit zur Verfügung zu stellen. Dies geht auch aus der folgenden Definition hervor:

> **Informationssystem:**
>
> System zur Speicherung, Wiedergewinnung (engl. information retrieval), Verknüpfung und Auswertung von Informationen. Ein Informationssystem besteht aus einer Datenverarbeitungsanlage, einem Datenbanksystem und den Auswertungsprogrammen.
>
> *Duden Informatik, Dudenverlag Mannheim/Wien/Zürich, korr. Nachdruck 1989*

Um das View4You-System einem möglichst breiten Personenkreis zur Verfügung stellen zu können, wurde eine natürlichsprachliche Eingabe bei der Entwicklung favorisiert. Ziel ist es, auf Benutzerangaben der Art

- 'Ich will alles über Prinzessin Diana wissen'
- 'Gibt es Informationen über den Nahost-Friedensprozeß?'

adäquat zu reagieren.

Liegt eine solche Anfrage vor, so müssen aus den archivierten Segmenten diejenigen bestimmt werden, die in Bezug auf die Anfrage relevant sind. Dieses Problem - aus einer Kollektion von Texten solche zu extrahieren, die in Bezug auf eine Anfrage semantisch relevant sind - bezeichnet man als *information retrieval*.

Die bekannteste Klasse von Information-Retrieval-Systemen sind die Internet-Suchmaschinen. Diese Programme durchsuchen weite Bereiche des World Wide Web (WWW) und nehmen die gefundenen Internetseiten in ihre Datenbank auf. Stellt ein Benutzer - in der Regel über ein Browserinterface - eine Anfrage, so wird eine nach Relevanz sortierte Liste von Internetseiten ausgegeben, die zu dieser Anfrage passen. Abgesehen von Implementierungsdetails, die vor allem der Abwehr unerwünschter Relevanz-Steigerungsmaßnahmen von Seiten der Internetseiten-Ersteller dienen, entsprechen die eingesetzten Algorithmen der Suchmaschinen den in diesem Kapitel vorgestellten Algorithmen.

In diesem Kapitel wird zunächst ein Abriss der Geschichte des Information Retrieval gegeben. Danach wird der Okapi-Algorithmus, der im View4You-System implementiert wurde, erläutert und motiviert. Abschließend werden einige wichtige Verfahren, die der Steigerung der Retrieval-Leistung dienen, besprochen.

Information Retrieval

Pionierarbeit für den Bereich Information Retrieval wurde in den späten sechziger Jahren von Salton und Kollegen durch den Aufbau des SMART-Systems [136] geleistet.

In jüngerer Zeit wurde die Forschung auf diesem Gebiet vor allem durch das TIPSTER-Programm des amerikanischen National Institute of Standards (NIST) vorangetrieben. Das Ziel von TIPSTER ist, die Forschung in den Gebieten

- Suche relevanter Dokumente in großen Datenbanken, und
- Informationsextraktion und Generierung von Zusammenfassungen

zu fördern und diese beiden Gebiete in einem System zu vereinen. Dafür wurde eine große Datenbank mit englischen Texten (etwa 1 Million Dokumente aus 'Wall Street Journal', 'AP News Wire' und anderen Quellen) mit zusammen 4 GByte Größe gesammelt. Aus dem TIPSTER-Programm gingen zwei Konferenzen hervor, die sich mit den beiden Hauptzielen des Projekts beschäftigen: TREC (für **T**ext **Re**trieval **C**onference) und MUC (für **M**essage **U**nderstanding **C**onference). Im November 1992 fand in Gaithersburg (Maryland, USA) die erste Text Retrieval Conference (TREC-1) statt. Die Konferenz wurde ein großer Erfolg und wird bis heute (1999) fortgesetzt.

Algorithmen des Information Retrieval

Das Herz jedes Information-Retrieval-Systems ist ein Algorithmus, der Ähnlichkeiten zwischen zwei Texten a und d (der Anfrage und dem Datensatz) berechnet. Dabei wird Ähnlichkeit meist über das gemeinsame Auftreten von Schlüsselwörtern in beiden

Texten festgestellt: Je mehr (semantisch relevante) Wörter in beiden Texten a und d vorkommen, desto ähnlicher sind die beiden Texte.

Ein semantisch relevantes Wort wird auch als 'Term' bezeichnet. Prinzipiell werden - im einfachsten Fall - aus Anfrage und Text die Terme extrahiert und als (hochdimensionale) Termvektoren abgespeichert. Jede Dimension der Vektoren gibt an, wie häufig der Term, der dieser Dimension zugeordnet ist, in dem jeweiligen Dokument enthalten ist. Die Ähnlichkeit zwischen Anfrage und Text ergibt sich aus dem Skalarprodukt der beiden Termvektoren, normiert mit den Textlängen. Dieser Algorithmus wird auch als Vektorraummodell bezeichnet.

Der einfache Ansatz des Vektorraummodells wird in der Praxis noch verfeinert. So sind nicht alle Terme gleich bedeutsam. Konjunktionen z.B. tragen nahezu keine semantische Bedeutung. Dieser Tatsache wird durch die Einführung von Termgewichten Rechnung getragen. Eine Standardmethode zur Berechnung von Termgewichten ist die sogenannte *tf-idf*-Methode (nach englisch: term frequency - inverse document frequency), wobei das Termgewicht g_k für das Wort k definiert wird als

$$g_k = \mathrm{tf}_k * \log(\frac{N}{f_k}) = \mathrm{tf}_k * \mathrm{idf}_k \tag{7.1}$$

wobei tf_k die Häufigkeit des Auftretens des Termes k im betrachteten Dokument ist, N die Anzahl aller Dokumente der Datenbank und f_k die Anzahl derjenigen Dokumente, die den Term k enthalten. Das Termgewicht g_k wird gerade dann groß, wenn ein Term in einem Dokument häufig vorkommt, aber nur in wenigen Dokumenten der Kollektion vertreten ist. Eine übliche Normierung auf die Dokumentlänge ist

$$w_k = \frac{\mathrm{tf}_k * \mathrm{idf}_k}{\sqrt{\sum_k (\mathrm{tf}_k \mathrm{idf}_k)^2}} \tag{7.2}$$

In [79] wurden verschiedene Vektorraummodelle und das Okapi-Modell der City University London [28] gegeneinander evaluiert. Dabei übertraf die Performanz des Okapi-Modells alle Parametrisierungen der Vektorraummodelle. Bei dieser Evaluation wurden relativ lange Anfragen an das System (im Mittel 77 Terme) verwendet.

Im Okapi-Verfahren wird an Stelle der *inverse document frequency* die Robertson/Sparck-Jones Gewichtung eingesetzt. Dabei werden die Dokumente noch in die Klassen `[ relevant / irrelevant ]` unterteilt und das Wissen über die Wahrscheinlichkeit eines Terms innerhalb der beiden Klassen verwendet. Die Robertson/Sparck-Jones Gewichtung ist definiert als

$$\mathrm{rsj} = \log(\frac{(r+0.5)/(R-r+0.5)}{(n_k-r+0.5)/(N-R+0.5)}) \tag{7.3}$$

Verfügt man über kein Wissen über die Verteilung relevanter und irrelevanter Dokumente, reduziert sich die Robertson/Sparck-Jones Gewichtung auf

$$\mathrm{rsj} = \log \frac{N+0.5}{n_k+0.5} \tag{7.4}$$

Der Grenzfall der Robertson/Sparck-Jones Gewichtung für einen niedrigen Anteil relevanter Dokumente an der Gesamtdatenbank sowie für unbekannte Relevanz der Dokumente ist also gerade die *inverse document frequency*.

In [85] wird auf das Problem kurzer Anfragen (im Mittel 4 Terme) eingegangen, die bei der Abfrage einer multimedialen Datenbank besonders wichtig sind. Das Ergebnis ist, dass auch hier eine Variante des Okapi-Modells die besten Ergebnisse zeigt. Interessanterweise wurde eine sehr einfache Okapi-Variante gewählt, die ohne Wissen über die Relevanz vorauszusetzen ($R = r = 0$) trotzdem sehr gute Ergebnisse erzielt. Sie läßt sich schreiben als

$$\begin{aligned} d(q,d) &= \sum_{t \in q \wedge t \in d} \log\left(\frac{N - f_t}{f_t}\right)\left(\frac{f_{d,t}}{f_{d,t} + \sqrt{\frac{f_d}{E(f_d)}}}\right) \\ &= \mathrm{OKA}(k_1 = 1, k_2 = 0, k_3 = 0, b = 1, r = 0, R = 0) \end{aligned} \tag{7.5}$$

Dabei ist N die Zahl der Dokumente in der Datenbank, f_t die Anzahl der Dokumente, die den Term t enthalten, $f_{d,t}$ ist die Häufigkeit des Vorkommens von Term t in Dokument d (also die *term frequency* des Terms t in Dokument d), und f_d ist die Anzahl der unterschiedlichen Terme im Dokument d (eine Approximation an die Dokumentlänge).

Bevor die Vorschrift nach Gl. 7.5 angewendet wird, wird jeder Text noch einer Vorverarbeitung unterworfen. Diese dient dazu, irrelevante Wörter zu identifizieren und zu tilgen und Flektionsformen in die Grundformen zu kanonisieren.

Die Berechnung der Ähnlichkeit zweier Texte lässt sich im folgenden Schema zusammenfassen:

1. Die Worte beider Texte d und q werden identifiziert und komplett in Kleinbuchstaben gewandelt.
2. Alle Worte, die in einem Wörterbuch von semantisch irrelevanten Worten (die sog. *Stopwortliste*) enthalten sind, werden gelöscht (äquivalent: ihre Termgewichte werden auf Null gesetzt).
3. Eine morphologische Suffix-Tilgung wird eingesetzt, um Flektionsformen von Worten in die Stämme zu verwandeln.
4. Die Okapi-Distanz $d(q,d)$ nach Gl. 7.5 wird berechnet.

Bei einer Anfrage q an die Datenbank werden diese vier Schritte für alle Dokumente d in der Datenbank durchgeführt und die resultierenden Ähnlichkeitswerte sortiert. Es wird eine nach Ähnlichkeit sortierte Liste aller Dokumente zurückgeliefert, die eine Ähnlichkeit größer als Null zur Anfrage q aufweisen.

Stopwords

Die sogenannte Stopwortliste enthält häufige Wörter der deutschen Sprache, die üblicherweise keine semantische Bedeutung tragen, wie z.B. Konjunktionen ('und', 'oder'). Würde man diese Wörter als Terme behandeln, würde die Berechnung der Ähnlichkeit zwischen zwei Dokumenten dadurch stark beeinflusst und damit verfälscht.

Die Stopwortliste des View4You-Systems enthält insgesamt 508 Worte. Es handelt sich dabei um die hundert häufigsten Wörter (Nomina und Nicht-Nomina) sowie um die 400 nächsthäufigeren Nicht-Nomina des Sprachmodellkorpus. Nach einer Analyse der typischen Anfragen wurden noch 8 'anfragetypische' Wörter (*Berichte, Informationen, möchte, interessiert, Tagen, Ergebnisse, über, Zahlen*) in die Stopwortliste aufgenommen.

Morphologische Dekomposition

Die morphologische Zerlegung dient der Kanonisierung der Terme. So ist es nicht sinnvoll, die beiden Terme 'Experte' und 'Experten' als unterschiedlich zu behandeln. Ziel der morphologischen Dekomposition ist es, alle Wörter auf ihre Stämme zurückzuführen und damit die Abhängigkeit vom im Text vorliegenden Kasus aufzuheben.

Im View4You-System wird nur eine sehr einfache Morphologie eingesetzt. Im Wesentlichen werden typische Wortendungen wie 'en', 'es', 'e' und 's' und bei einer Wortlänge von mehr als 7 Zeichen auch die Endsilben 'heit' und 'keit' getilgt.

Verwendung eines Thesaurus

Zur Leistungssteigerung von IR-Systemen kann ein Thesaurus eingesetzt werden. Dieser stellt Relationen zwischen den Termen des IR-Systems her. Bei einer Anfrage werden die Relationen vom IR-System in geeigneter Weise genutzt, um eine höhere Ausschöpfung und/oder eine höhere Präzision zu erreichen.
Ein Beispiel für eine solche Relation kann z.B. die Relation 'ist Flektionsform von' sein. Diese Relation - die implizit durch die morphologische Dekomposition realisiert ist - würde dann die beiden Terme 'Israel' und 'Israels' verknüpfen. Sucht der Benutzer nach 'Israel', kann das IR-System auch Vorkommen von 'Israels' berücksichtigen

und, ggf. mit einer kleineren Gewichtung, dem Benutzer zurückliefern. Die folgende Liste von Thesaurus-Relationen lehnt sich an die Abhandlung von Juettner [130] an.

1. Synonyme Begriffe
   Hierunter sind unterschiedliche Bezeichnungen für das gleiche Objekt zu verstehen. Dies schließt Übersetzungen in verschiedene Fremdsprachen sowie Abkürzungen mit ein.

2. Homonyme Begriffe
   Homonyme Begriffe haben im Kontext eines Informationssystems unterschiedliche Bedeutungen. Zur Desambiguierung enthält der Thesaurus semantische Informationen zu den Begriffen, z.B.

   - Bank (Sitzmöbel)
   - Bank (Kreditinstitut)

3. Verwandte Begriffe
   Begriffe, die miteinander in einer anderen Art des Zusammenhangs stehen als einer der bisher genannten, werden in diese Gruppe eingeordnet. Beispiele sind Gegensatzpaare (*bottom up* versus *top down*) oder funktionale Zusammenhänge (*Schiff* versus *Anker*)

4. Ober/Unterbegriffe
   Begriffe, die eine Verallgemeinerung bzw. Spezialisierung voneinander darstellen, wie *Baum* und *Fichte*, *Getränk* und *Sprudel*

5. Flektionsform von
   Diese Relation besteht zwischen verschiedenen Flexionen desselben Wortes, wie z.B. *Baum* und *Bäume*

Experimente

Das Informationssystem wurde im Rahmen der Systemevaluation evaluiert. Die Experimente hierzu finden sich im Kapitel 'Systemevaluation' ab Seite 170.

Kapitel 8

Konfidenzmaße

Spracherkenner erhalten als Eingabe eine digitalisierte Sprachprobe und produzieren als Ausgabe eine Sequenz von Wörtern, die im Idealfall dem Gesagten entspricht. Allerdings arbeiten weder Maschinen noch Menschen als fehlerfreie Spracherkenner: Oft wird etwas anderes verstanden als gesagt wurde. Die Fehlerraten der besten Spracherkennungssysteme liegen dabei für klare, wohlformulierte vorgelesene Eingaben eine Größenordnung über der des Menschen [33], und das Verhältnis wird noch wesentlich günstiger für menschliche 'Spracherkenner', sobald Hintergrundgeräusche, wechselnde akustische Kanäle oder undeutliche Aussprache hinzukommen.

Hört ein Mensch Sprache, so tritt bisweilen die Situation auf, dass eine Unsicherheit in Bezug auf das Gesagte entsteht. Man glaubt zu wissen, was gesagt wurde, ist sich aber nicht sicher. In anderen Worten, das Vertrauen in die Akkuratheit der eigenen Spracherkennung ist niedrig.

Es wäre nun für viele Aufgaben nützlich, wenn auch ein maschineller Spracherkenner eine solche Beurteilung der eigenen Erkennungsleistung vornehmen könnte. Dies ist die Aufgabe eines Vertrauensmessers.

8.1 Eine Taxonomie der Vertrauensmesser

Vertrauensmesser können eingeteilt werden nach den *Basiseinheiten*, deren Konfidenz bestimmt wird, und nach der *Art*, wie bei diesen Einheiten die Angabe der Konfidenz erfolgt.

Art der Angabe der Konfidenzen

Bezüglich der *Art* der Ausgabe der Konfidenz kann man zwischen binären Vertrauensmessern und kontinuierlichen Vertrauensmessern unterscheiden. Ein kontinuierlicher Vertrauensmesser gibt für jede Basiseinheit eine Schätzung der a-posteriori-Wahrscheinlichkeit aus, mit der diese Basiseinheit korrekt erkannt worden ist. Für

diese Wahrscheinlichkeit gilt

$$0 < p \leq 1 \tag{8.1}$$

Betrachtet man als Applikation beispielsweise einen Ziffernerkenner für die Ziffern null bis neun, so könnte für eine gegebene Spracheingabe die Ausgabe des Erkenners selbst 'zwei' und die Ausgabe des Vertrauensmessers '0.9' sein. Dieses Ergebnis wäre dann wie folgt zu interpretieren:
Mit einer Wahrscheinlichkeit von 90% handelt es sich bei der Eingabe um das gesprochene Wort 'zwei'. Mit einer Wahrscheinlichkeit von insgesamt 10% wurde nicht 'zwei', sondern eine andere Ziffer gesprochen.

Über die Verteilung dieser 10% auf die neun verbleibenden möglichen Ziffern wird nichts ausgesagt.

Für eine andere Eingabe könnte die Ausgabe des Erkenners 'sieben' und die Ausgabe des Vertrauensmessers '0.4' sein. In diesem Falle wäre die Wahrscheinlichkeit dafür, dass 'sieben' gesagt wurde, kleiner als die, dass eine andere Ziffer vorlag. Die 'Restwahrscheinlichkeit' von 60% verteilt sich allerdings so auf die verbleibenden neun Ziffern, dass keine davon eine größere Wahrscheinlichkeit als 0.4 aufweist - sonst hätte der Spracherkenner nicht 'sieben', sondern diese andere Ziffer hypothetisiert. Die Situation lässt sich etwa so umschreiben:
Der wahrscheinlichste Kandidat ist 'sieben', aber es ist wahrscheinlich eher nicht die 'sieben' sondern eine andere Ziffer gesprochen worden'.

Bei *binären* Vertrauensmessern kann die Ausgabe nur die beiden Werte 'richtig' und 'falsch' annehmen. Im obigen, ersten Beispiel des Ziffernerkenners wäre die Ausgabe des Erkenners wieder 'zwei' und die Ausgabe des Vertrauensmessers 'richtig'. Im zweiten Beispiel würden die Ausgaben dementsprechend 'sieben' und 'falsch' lauten. Offensichtlich ist es möglich, jeden kontinuierlichen Vertrauensmesser durch die Definition einer Schwelle ϵ in einen binären Vertrauensmesser umzuwandeln. Durch die geeignete Festlegung des Schwellwertes kann dabei eine Anpassung an die Anforderungen der jeweiligen Anwendung erfordern. Wünscht man beispielsweise hohe Sicherheit, zum Beispiel für eine Maschinensteuerung für medizinische Eingriffe, kann der Schwellwert ϵ hoch, etwa bei $\epsilon = 0.99$, angesetzt werden. Das System reagiert dann nur auf Eingaben, die es mit 99%-iger Sicherheit korrekt klassifizieren konnte.

Im Beispiel des Ziffernerkenners wäre die niedrigste theoretisch mögliche Konfidenz 0.1, weil es zehn unterschiedliche Ziffern gibt. Sind nach Ausnutzen der internen Wissensquellen des Ziffernerkenners immer noch alle zehn Ziffern gleich wahrscheinlich, liegt die maximal mögliche Unsicherheit vor und der Erkenner gibt - zufällig ausgewählt - eine der Ziffern, zum Beispiel 'eins', als Hypothese aus und belegt diese mit der Konfidenz 0.1.

Eine kleinere Wahrscheinlichkeit als 0.1 ist bei einem Ziffernerkenner nicht

möglich, weil sich die Wahrscheinlichkeiten der zehn möglichen Ausgaben des Erkenners zu 1 addieren müssen - unter der Annahme, die Eingabe enthielte auch tatsächlich genau eine Ziffer. Ist also die Wahrscheinlichkeit einer bestimmten Ziffer kleiner als 0.1, muss die einer anderen größer als 0.1 sein und wäre somit eine wahrscheinlichere Hypothese, die vom Erkenner auch ausgegeben werden müsste. Ganz allgemein gilt für jeden Klassifikator in einem N-Klassen-Problem, dass die kleinste Konfidenz in eine Klassifikation durch $\frac{1}{N}$ gegeben ist. Für eine endliche Zahl von Klassen folgt hieraus (8.1). Es folgt auch, dass ein binäres Konfidenzmaß für ein Zwei-Klassen-Problem sinnlos ist: die Ausgabe müsste stets 'richtig' lauten.

Definition der Basiseinheit

Im Falle eines Ziffernerkenners für einzelne Ziffern ist die Definition der Basiseinheit einfach: Die ganze Ausgabe besteht aus einem einzigen Wort und dieses wird bewertet. Bei kontinuierlicher Sprache kann man ebenfalls als Basiseinheit die ganze Äußerung wählen. In diesem Fall würde die Korrektheit der gesamten Hypothese bewertet. 'Richtig' ist eine Hypothese dann, wenn alle Worte in ihr richtig erkannt worden sind. Dieser Ansatz ist brauchbar, wenn die Wortfehlerrate des Systems sehr klein ist. Aus den Erfahrungen der US-amerikanischen DARPA Hub-4 Evaluationen [19] jedoch ist bekannt, dass selbst auf den relativ leicht zu erkennenden Ansagesprechern die besten Systeme noch Wortfehlerraten von etwa 10% absolut aufweisen. Da ein durchschnittliches Segment der tagesschau 48 Worte enthält, ergibt sich daraus eine a-priori-Wahrscheinlichkeit von $0.9^{48} = 0.006 = \frac{1}{158}$ dafür, dass ein Segment fehlerfrei erkannt wird. Würde man also ganze Sätze als Basiseinheiten benutzen, würde ein (ideales) Vertrauensmaß in 157 von 158 Fällen die Hypothese mit 'fehlerhaft' bewerten. Der Nutzen eines solchen Maßes ist zumindest fragwürdig.

Daher wählt man als Basiseinheit anstelle der gesamten Hypothese meistens ihre einzelnen Worte. Dieser Ansatz hat den weiteren Vorteil, dass differenzierte Reaktionen auf das Ergebnis der Vertrauensmessung möglich sind. So kann z.B. eine niedrige Konfidenz eines Wortes einer Begrüßungsfloskel ignoriert werden, während eine niedrige Konfidenz bei einem diktierten Überweisungsbetrag zu einer Rückfrage beim Benutzer führt.
Obwohl es einfach und elegant scheint, die einzelnen Wörter der Hypothese als Basiseinheiten der Vertrauensmessung zu verwenden, gibt es bei diesem Ansatz ein Problem durch die Möglichkeit von Löschungen und Einfügungen. Ein Beispiel möge das veranschaulichen.

Bild 8.1 zeigt eine reale Sprachprobe, die Hypothese des Erkenners und die tatsächlich gesprochene Äußerung. In dieser Sprachprobe hatte die Sprecherin ein technisches Problem ('Hallo, ich verstehe dich schon nicht mehr') mitteilen wollen. Die tatsächliche Äußerung lautete 'Hallo ich versteh' ich schon nicht mehr' (das 'dich' wurde ohne 'd' realisiert). Dabei ist die zeitliche Zuordnung der Wörter zum Signal in der oberen Zeile in Bild 8.1 angegeben.

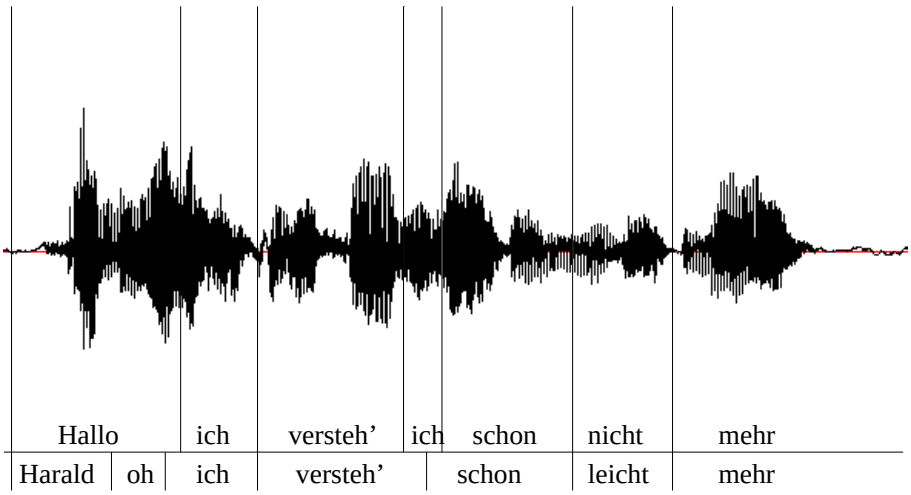


Abbildung 8.1: Real gesprochener Satz (oben) und Hypothese des Erkenners (unten)

Die (fehlerhafte) Hypothese des Spracherkenners lautete ’Harald oh ich versteh’ schon leicht mehr’. Die vom Spracherkenner gefundene Zeitzuordnung findet sich in der unteren Zeile in Bild 8.1. Wie groß *sollte* nun das Vertrauen des Erkenners in die einzelnen Wörter dieser Hypothese sein?

Das Wort ’mehr’ ist ohne jeden Zweifel richtig erkannt worden und sollte daher eine möglichst hohe Konfidenz aufweisen. ’Harald’, ’oh’ und ’leicht’ sind - ebenfalls ohne Zweifel - falsch erkannt worden und sollten daher eine möglichste kleine Konfidenz aufweisen. Problematischer ist das erste ’ich’, das zwar richtig erkannt, aber zeitlich (am Anfang) falsch zugeordnet wurde. Welche Konfidenz in einem solchen Fall erwünscht ist, hängt von der Anwendung ab und kann nicht allgemein festgelegt werden.

Das eigentliche Problem mit Wörtern als Basiseinheiten besteht jedoch bei Löschfehlern, wie beim zweiten ’ich’ der Referenz. Sowohl die Worte ’versteh’ als auch ’schon’ sind im Sinne der Wortfehlerrate *richtig* erkannt worden. Daher sollte der Vertrauensmesser ihnen die Konfidenz 1.0 zuweisen. Eine solche Zuweisung ignoriert jedoch die Tatsache, dass zwischen den beiden Wörtern ein Löschfehler stattgefunden hat. Weil in der Hypothese an dieser Stelle aber kein Wort vorhanden ist, ist es bei wortbasierten Vertrauensmessern nicht möglich, Löschfehler adäquat zu behandeln.

Ein Ausweg aus diesem Dilemma ist die *Zeit* als Basiseinheit. Ein zeitbasierter Vertrauensmesser weist Zeitintervallen Vertrauenswerte zu. Dadurch sind prinzipiell die Probleme bei der Verwendung von Wörtern als Basiseinheiten gelöst: Sowohl Löschungen als auch falsche Zeitzuordnungen der Hypothese können behandelt werden. Als Zeiteinheit kommt zum Beispiel der Zeitbereich eines Merkmalsvektors der

Vorverarbeitung (i.a. 10 Millisekunden) in Frage.

Trotz der theoretischen Vorteile der Zeit als Basiseinheit dominieren Worte als Basiseinheiten die derzeitige Forschung. Das liegt zum einen daran, dass zahlreiche, zur Konfidenzschätzung nützliche Wissensquellen, wie zum Beispiel das statistische Sprachmodell, explizit auf Wörtern aufbauen und für eine zeitbasierte Vertrauensmessung nur schwer nutzbar zu machen sind. Auf der anderen Seite fallen die durch Löschfehler verursachten Probleme bei Wörtern als Basiseinheiten in vielen Anwendungen deshalb nicht ins Gewicht, weil Löschfehler im Vergleich zu Substitutionsfehlern sehr viel seltener auftreten.

Der in dieser Arbeit entwickelte Vertrauensmesser arbeitet entsprechend der dargelegten Taxonomie *wortbasiert* und *kontinuierlich*.

8.2 Bewertung von Vertrauensmessern

Wenn für einen Spracherkenner zwei verschiedene Vertrauensmesser vorliegen, stellt sich die Frage, welcher der Vertrauensmesser der Bessere ist. Um diese Frage allgemein zu beantworten, wäre eine Bewertungszahl nützlich, anhand derer man - in Analogie zur Wortfehlerrate bei Spracherkennern - einen Vergleich vornehmen könnte. Der Bessere der beiden Vertrauensmesser ist dann derjenige, der den höheren (bzw. niedrigeren) Wert der Bewertungszahl aufweist.
Im Gegensatz zu der Situation bei Spracherkennern, wo die Wortfehlerrate (4.39) eine weltweit anerkannte Bewertungszahl ist, gibt es bei Vertrauensmessern mit einem kontinuierlichen Wertebereich der ausgegebenen Konfidenzen eine Vielzahl miteinander konkurrierender Ansätze zu ihrer Bewertung. Vergleichsweise einfach stellt sich die Situation hingegen bei digital arbeitenden Vertrauensmessern dar, also bei solchen, die jedes Wort der Hypothese des Spracherkenners entweder mit 1 (für 'richtig') oder mit 0 (für 'falsch') markieren.

Bewertung von digitalen Vertrauensmessern

Ein digitaler Vertrauensmesser weist jedem Wort der Hypothese des Spracherkenners eine Marke (engl. *tag*) 1 oder 0 (für 'richtig erkannt' bzw. für 'Erkennungsfehler') zu. Für diese Situation lässt sich analog zur Wortkorrektheit WC des Spracherkenners (siehe Gl. (4.37) auf Seite 46) eine Konfidenzakkuratheit (CA, confidence accuracy) wie folgt definieren:

$$CA := \frac{\text{Anzahl korrekter Marken}}{\text{Gesamtzahl der Wörter in der Hypothese}} \tag{8.2}$$

Die CA liegt stets zwischen 0 und 1. Den Wert 1 nimmt sie genau dann an, wenn jedes Wort der Hypothese korrekt bewertet wurde. Ein trivialer Vertrauensmesser,

der stets alle Wörter mit '1' markiert, erreicht für die Konfidenzakkuratheit CA genau den Wert der Wortkorrektheit WC. In diesem Fall ist nämlich der Anteil der richtig markierten Wörter an allen Wörtern der Hypothese gerade der Anteil der richtig erkannten Wörter an allen Wörtern der Hypothese.

Mit Hilfe einer Entscheidungsschwelle ϵ kann ein kontinuierlich arbeitender Vertrauensmesser in einen digitalen umgewandelt werden. Für den so entstandenen digitalen Vertrauensmesser kann die CA nach (8.2) berechnet werden. Auf diese Weise kann man auch für einen kontinuierlichen Vertrauensmesser eine CA (als Funktion von ϵ) angeben und ihn so bewerten.

Bewertung von kontinuierlichen Vertrauensmessern

Zur Bewertung von kontinuierlichen Vertrauensmessern wurden verschiedene Kriterien vorgeschlagen [42] [47] [92]. In dieser Arbeit wird zusätzlich zur *confidence accuracy*, die durch ihre Analogie zur Wortakkuratheit sehr anschaulich ist, die normalisierte Kreuzinformation, oder 'normalized cross entropy' NCREP [42] verwendet. Dieses Maß wurde auch von der NIST quasi offiziell zum Standardmaß der Performanz von Vertrauensmessern deklariert [36].

NCREP ist definiert als

$$NCREP = \frac{CREP_0 - CREP}{CREP_0} \tag{8.3}$$

$$CREP = \frac{1}{N}\Big(\sum_{w\in \text{ korrekte Worte}} \log(c_w) + \sum_{w\in \text{ fehlerhafte Worte}} \log(1-c_w)\Big) \tag{8.4}$$

wobei c_w die Ausgabewahrscheinlichkeit des Vertrauensmessers für das jeweilige Wort ist. $CREP_0$ ist $CREP$ für den besten a-priori-Vertrauensmesser, also für denjenigen, der alle Wörter gleich (mit der Wort-Korrekt-Rate c_0) markiert:

$$CREP_0 = c_0 \log(c_0) + (1-c_0)\log(1-c_0) \tag{8.5}$$

Für den perfekten Vertrauensmesser ist $CREP = 0$ und damit $NCREP = 1$. Je weiter die Resultate des Vertrauensmessers von den richtigen Werten (1 für richtig erkannte und 0 für falsch erkannte Worte) abweichen, desto kleiner wird $NCREP$. Der a-priori-Vertrauensmesser erzielt nach der Definition eine $NCREP$ von 0.

$NCREP$ misst also auf einer Skala von 0 bis 1, um wieviel ein Vertrauensmesser besser ist als der einfache a-priori-Vertrauensmesser.

8.3 Aufbau eines Vertrauensmessers

Der Vertrauensmesser erhält als Eingabe die Hypothese des Erkenners. Seine Ausgabe ist eine a-posteriori-Wahrscheinlichkeit für jedes Wort dieser Hypothese. Um diese be-

rechnen zu können, benötigt der Vertrauensmesser natürlich zusätzlich zur Hypothese weitere Wissensquellen. Es sind viele Wissensquellen denkbar, die Anhaltspunkte über die Konfidenz in ein Wort in der Hypothese eines Spracherkenners liefern. Wenn die Wissensquellen nicht sehr stark miteinander korreliert sind, ist anzunehmen, dass eine geeignete Kombination mehrerer Wissensquellen eine bessere Klassifikationsleistung erbringt als jede der verwendeten Wissensquellen für sich alleine. Aus diesem Grund bietet sich für den Vertrauensmesser ein Vektorklassifikator-Ansatz an. Den prinzipiellen Aufbau eines Vertrauensmessers mit diesem Ansatz zeigt Bild 8.2.

Eingabedaten für den Vertrauensmesser sind

- die Hypothese, die bewertet werden soll,
- ein Satz von Merkmalen.

Die Merkmale müssen zwei Anforderungen genügen:

- Sie müssen für jedes Wort einzeln berechenbar sein.
- Sie dürfen die Kenntnis des tatsächlich Gesagten nicht voraussetzen.

Über Art und Anzahl der Merkmale ist hiermit noch nichts ausgesagt. Generell kann gesagt werden, dass jedes verwendete Merkmal zur Wahrscheinlichkeit des Auftretens eines Erkennungsfehlers korreliert sein muss, um zur Klassifikation nützlich zu sein.

Die aus diesen Informationsquellen gewonnenen Hinweise auf die Richtigkeit des zu beurteilenden Wortes müssen zu einer a-posteriori-Wahrscheinlichkeit (oder, im Falle eines digital arbeitenden Vertrauensmessers, zu einer richtig/falsch-Entscheidung) geeignet verrechnet werden. Diese Aufgabe erfüllt der Regressor, bzw. im Falle des digitalen Vertrauensmessers der Klassifikator. Aus der Literatur sind verschiedene Ansätze für Vektorklassifikatoren und -regressoren bekannt. Welcher davon für die gegebene Aufgabe am besten geeignet ist, kann nur experimentell bestimmt werden. In den meisten Fällen wird ein Regressor auf einer Trainingsstichprobe *trainiert*. Dabei werden Hypothesen für eine Menge von Sprachproben generiert, bei denen die Referenz, also das tatsächlich Gesagte, bekannt ist. Für jedes Wort der Hypothese wird dann bestimmt, ob es sich um einen Erkennungsfehler (=0) oder um korrekte Erkennung (=1) handelt. Diese Werte stellen die Zielausgabe des Vertrauensmessers dar, d.h. ein idealer Vertrauensmesser generiert für jedes richtig erkannte Wort eine 1.0 und für jedes falsch erkannte Wort eine 0.0 als Ausgabe, und zwar selbstverständlich ohne die Kenntnis des tatsächlich Gesagten. Mit dieser Zielausgabe und den Eingabemerkmalen für das betrachtete Wort wird die Transferfunktion von den Eingabemerkmalen zur Zielausgabe gelernt. Das Lernverfahren hängt natürlich von der Art des eingesetzten Regressors ab.

Die Erstellung eines Vertrauensmessers gliedert sich somit in die Schritte

1. Finden eines geeigneten Satzes brauchbarer Merkmale
2. Konstruktion eines Klassifikators

Den prinzipiellen Aufbau eines Vertrauensmessers zeigt Abbildung 8.2.

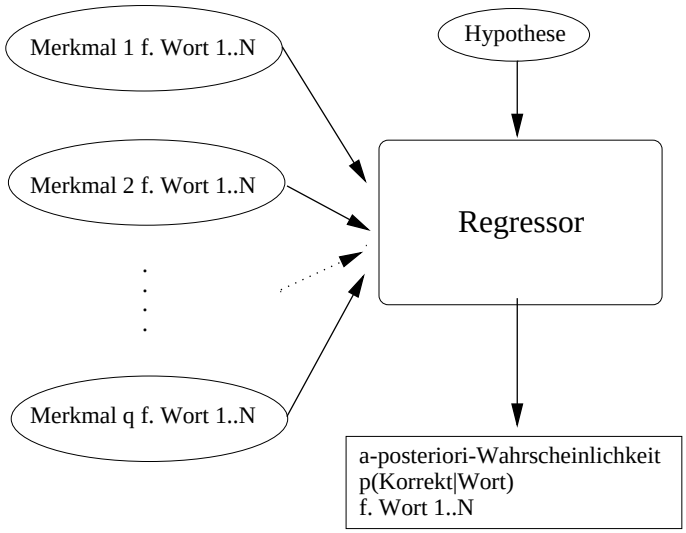


Abbildung 8.2: Prinzipieller Aufbau eines Vertrauensmessers

Zunächst sollen nun die Vorarbeiten anderer Gruppen beschrieben werden, die auch den Ausgangspunkt zu den in dieser Arbeit durchgeführten Experimenten darstellen.

8.4 Ansätze aus der Literatur

Die Forschung auf dem Gebiet der Vertrauensmesser führte bis zum Anfang der neunziger Jahre ein Schattendasein neben der Forschung an Spracherkennung als solcher. Anfang der neunziger Jahre war jedoch die Entwicklung von Spracherkennern an einem Punkt angelangt, an dem sich erste reale Systeme zum Einsatz außerhalb des Labors implementieren ließen. Diese Systeme beschränkten sich meist auf die Erkennung von einfachen, nicht kontinuierlich gesprochenen Schlüsselwörtern, wie zum Beispiel die Ziffern 0 bis 9, und wurden häufig in Telefonauskunftssystemen eingesetzt. Im Gegensatz zu den kontrollierten Laborbedingungen wurden diese Systeme mit naiven Benutzern konfrontiert, die die Systeme nicht kannten:

> *Hier ist die Telefonzentrale der X-Versicherung. Wenn Sie eine neue Versicherung abschließen wollen, dann sagen Sie bitte 'eins'. Haben Sie eine Frage zu einer bestehenden Versicherung, dann sagen Sie bitte 'zwei'. Bitte sprechen Sie jetzt.*

Wie bitte? äh... bestehende Versicherung, ja genau. Also **ZWEI**. *Hallo?*

Das System erwartet an dieser Stelle genau ein Wort als Eingabe, nämlich entweder 'eins' oder 'zwei'. Es berechnet daher, ob die Eingabe - der ganze Satz - sich eher wie 'eins' oder wie 'zwei' anhörte. Das Ergebnis ist dann natürlich rein zufällig.
Um dem Problem abzuhelfen, wurden *Wordspotter* eingesetzt, deren Aufgabe das Herausfiltern der Schlüsselwörter aus einer möglicherweise Schlüsselwörter enthaltenden Eingabe war. Die von den Wordspottern gefundenen potentiellen Schlüsselwörter mußten danach noch verifiziert werden, ob sie

1. überhaupt ein Schlüsselwort waren, und
2. ob das Schlüsselwort mit hinreichender Sicherheit korrekt identifiziert werden konnte.

Die zweite Aufgabe schließt die erste mit ein.
Die meisten Arbeiten bis etwa 1995 behandeln folgerichtig die Verifikation von Schlüsselwörtern, die in Sprache eingebettet sind. Der beinahe ausschließlich verwendete Ansatz war die Berechnung einer *likelihood ratio* L. Diese ist definiert als das Verhältnis der Wahrscheinlichkeit dafür, dass es sich um das Schlüsselwort W_s handelt, zu der Wahrscheinlichkeit dafür, dass es sich um ein anderes Schlüsselwort W_q oder um überhaupt kein Schlüsselwort handelt. Je größer L ist, desto sicherer ist die Identifikation des Schlüsselwortes X.

$$L = \frac{p(W_s)}{\sum_{W \neq W_s} p(W)} \tag{8.6}$$

Für die Modellierung von $p(W_s)$, also der Schlüsselwörter selbst, können die akustischen Modelle des Wordspotters verwendet werden. Das eigentliche Problem besteht in der Modellierung des Nennerterms, der die Wahrscheinlichkeit dafür angibt, dass irgendein anderes Wort ausser W_s gesprochen wurde. Weil die Modelle für den Nennerterm alle akustischen Ereignisse ausser den Schlüsselwörtern abdecken müssen, werden sie auch als *filler models* (engl. Füll-Modelle) bezeichnet. Zur Problematik, welche *filler models* am besten geeignet sind, wurden von zahlreichen Autoren Vorschläge gemacht. Mehrere Autoren ([110], [105]) verwendeten kontextunabhängige Monophonmodelle ohne Wörterbuch oder Grammatik. In [102] wurde die gleichzeitige Verwendung von Phonemmodellen, Silben und Wortmodellen vorgeschlagen. Jeanrenaud [100] verglich die Verwendung von Phonemmodellen mit Ganzwortmodellen sowie mit Ganzwortmodellen mit Sprachmodell und stellte fest, dass bei einer guten Vokabularabdeckung der Ganzwortmodelle diese eine deutlich bessere Leistung erbrachten als Phonemmodelle. Diese Entdeckung war ein Vorgriff auf die spätere Entwicklung der Wordspotter.

Als ab etwa 1995 durch verbesserte Maschinenleistung Vokabularien von 60000 Wörtern beherrschbar wurden, stellte man fest, dass solche Spracherkenner bessere Wordspotter waren als alle spezialisierten Systeme. Damit hatte sich eine Synthese der beiden Entwicklungen vollzogen, denn wie bereits prinzipiell in [100] dargestellt, kann ein Spracherkenner als ein Wordspotter mit einem komplexen *filler model* aufgefasst werden. Dieser Synthese folgend wurden ab 1996 Arbeiten veröffentlicht, die die Bewertung jedes Wortes der Hypothese zum Ziel hatten. Dabei wurden zunehmend mehr Merkmale zur Klassifikation herangezogen.

Young [105], eine der ersten Autorinnen, die einen Vertrauensmesser in dieser modernen Typisierung vorstellten, schlug die Wortidentität als Merkmal vor. Darin wird ausgenutzt, dass manche Worte *a priori* eine geringere Wahrscheinlichkeit der Fehlerkennung haben als andere. Aus der Arbeit von Lleida und Kollegen [102] stammt der Vorschlag, die Länge des zu beurteilenden Wortes (gemessen in Phonemen) als Klassifikationsmerkmal heranzuziehen. Weintraub [84] schlug vor, die Anzahl der Vorkommen eines Wortes in der N-Besten-Liste der Hypothesen des Dekoders als Merkmal zu verwenden und demonstrierte die hervorragende Performanz dieses Merkmals.

Eide [65] analysierte mit Hilfe von Entscheidungsbäumen den Einfluss verschiedener Merkmale auf die Wahrscheinlichkeit der Fehlerkennung. In dieser Arbeit wurde neben schon bekannten Merkmalen die Häufigkeit des Hypothesenwortes im akustischen Trainingsmaterial, der Sprachmodell-Score, das Signal-Rauschverhältnis und der Schätzwert für die Sprechgeschwindigkeit als Prädiktoren analysiert. Cox und Rose [47] schlugen 1996 vor, die Anzahl der konkurrierenden Hypothesen während der Suche als Merkmal heranzuziehen.

Die erste systematische Zusammenfassung und Evaluierung einer Vielzahl von Merkmalen erfolgte im Rahmen der vorliegenden Arbeit in [24]. Die Ergebnisse sind in den folgenden Abschnitten zusammengefasst.

8.5 Auswahl geeigneter Wissensquellen

Die Aufgabe, Merkmale für den Konfidenzmesser zu selektieren, kann in zwei Unteraufgaben unterteilt werden:

- Bestimmen eines möglichst großen Satzes von möglichen Merkmalen
- Selektion der relevanten Merkmale daraus

Dabei ist der erste Schritt der eigentlich kreative Teil. Aus der (beinahe) unendlichen Mannigfaltigkeit von möglichen Merkmalen muß hier durch Expertenwissen und Intuition ein vielversprechender Satz von Merkmalen extrahiert werden. Welche sich davon in der Praxis bewähren, wird im zweiten Schritt evaluiert.

Bewertung von Merkmalen für Vertrauensmesser

Für den zweiten Schritt wird ein Qualitätsmaß für die Merkmale benötigt. Das beste, ergebnisorientierte Maß ist die Nützlichkeit des betrachteten Merkmals für die Akkuratheit des Vertrauensmessers. Dieses Maß erfordert allerdings den Bau eines separaten Klassifikators und das Testen desselben für jedes einzelne neue Merkmal, was sehr aufwendig ist.

Ein alternatives Maß ist die Berechnung des (linearen) Korrelationskoeffizienten r zwischen den Merkmalswerten und der Fehlerwahrscheinlichkeit aller Worte, die diese Merkmalswerte aufweisen. Dieses alternative Maß ist leicht zu berechnen. Weist ein Merkmal eine hohe lineare Korrelation mit der Fehlerwahrscheinlichkeit auf, ist es in jedem Fall zur Prädiktion von Fehlern geeignet. Der Umkehrschluß ist jedoch nicht richtig, wie die beiden Bilder 8.3 zeigen.

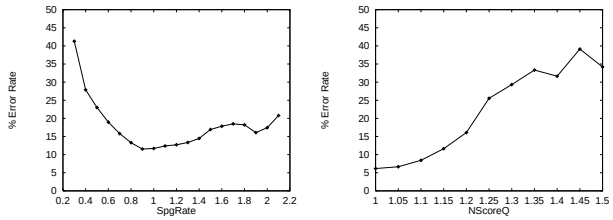


Abbildung 8.3: Fehlerrate über Merkmal **Sprechgeschwindigkeit** (links) und **akustischer Score** (rechts)

Bild 8.3 rechts zeigt das Merkmal *akustischer Score*, das im Folgenden beschrieben werden wird, und die Fehlerwahrscheinlichkeit über den Merkmalswerten, die in diesem Fall zwischen 1.0 und 1.5 liegen. Man erkennt eine lineare Korrelation, die sich auch durch den Korrelationskoeffizienten von 0.17 ausdrückt. Im Gegensatz dazu ist der lineare Korrelationskoeffizient für das Merkmal Sprechgeschwindigkeit (Bild 8.3) nahe null. Dennoch ist das Merkmal offensichtlich zur Bestimmung von Erkennungsfehlern nützlich: Je mehr die Sprechgeschwindigkeit von der Norm abweicht, desto größer ist die Wahrscheinlichkeit eines Erkennungsfehlers. Da der zugrundeliegende funktionale Zusammenhang jedoch nicht linear ist, weist r einen kleinen Wert auf.

Der lineare Korrelationskoeffizient zwischen zwei Merkmalen X und Y ist nur sinnvoll, wenn beide Merkmale kardinalskaliert vorliegen. Das ist für die Merkmalswerte in den meisten Fällen der Fall, für die Fehlerwahrscheinlichkeit Y sind jedoch nur die Werte 0 (Wort ist falsch) und 1 (Wort ist richtig) möglich. Man kann daher Y auch als nominalskaliert auffassen, die Merkmalswertskala X auf eine Nominalskala abwerten und dann den Kontingenzkoeffizienten K [114] berechnen. Dieser hat den Vorteil, dass auch Fälle nichtlinearen Zusammenhangs zwischen den Merkmalswerten und der Fehlerwahrscheinlichkeit erfasst werden können.

Der Kontingenzkoeffizient ist wie folgt definiert:

$$K = \sqrt{\frac{\chi^2}{N + \chi^2}} * K_{norm} \tag{8.7}$$

K_{norm} ist ein Normierungsfaktor, der von der Anzahl der möglichen Ausprägungen der beiden nominalskalierten Größen abhängt. Hat - wie im Falle der Konfidenzen - eine der beiden Größen nur zwei mögliche Ausprägungen, gilt $K_{norm} = 2$.

Die Größe χ^2 ist definiert als

$$\chi^2 = \sum_{i=1}^{k} \sum_{j=1}^{l} \frac{(N(i,j) - N'(i,j))^2}{N'(i,j)} \tag{8.8}$$

wobei i und j die verschiedenen Ausprägungen der nominalskalierten Größen numerieren, $N(i,j)$ die beobachtete Zahl der i,j Wertepaare ist und $N'(i,j)$ der Erwartungswert von $N(i,j)$ im Falle der statistischen Unabhängigkeit der beiden Größen voneinander.

Der Kontingenzkoeffizient K ist also zur Bewertung von Merkmalen besser geeignet als der Korrelationskoeffizient r. Aufgrund des großen Bekanntheitsgrades des linearen Korrelationskoeffizienten wurden jedoch für sämtliche untersuchten Merkmale beide Koeffizienten berechnet und angegeben.

Die Abwertung einer Kardinalskala - die im Falle der meisten Merkmale vorliegt - auf eine Nominalskala ist nicht eindeutig und muß in irgend einer sinnvollen Weise festgelegt werden. Für die Experimente in dieser Arbeit wurde willkürlich eine Einteilung der Kardinalskala in 25 Klassen definiert. Dabei wurden die Klassengrenzen so gesetzt, dass die Daten über alle Klassen gleichverteilt waren.

8.6 Untersuchte Merkmale

In diesem Abschnitt wird eine Vielzahl von unterschiedlichen Merkmalen zur Konfidenzschätzung vorgestellt. Einige dieser Merkmale wurden in der Literatur bereits vorgeschlagen; andere wurden im Rahmen dieser Experimente erstmals systematisch untersucht.

Merkmale basierend auf den akustischen Modellen

Es bietet sich an, die akustischen Modelle des Spracherkenners zur Schätzung der Konfidenz heranzuziehen. Diese Modelle berechnen die Wahrscheinlichkeit der Beobachtung des akustischen Signals, gegeben ein Allophon:

$$score = -\log(p(\text{akust. Signal}|\text{Allophon})) \tag{8.9}$$

Je kleiner also der über die Dauer eines Wortes gemittelte durchschnittliche 'score' ist, desto größer ist die Wahrscheinlichkeit der beobachteten akustischen Sequenz gegeben das hypothetisierte Wort. Im Umkehrschluß bedeutet das eine große Wahrscheinlichkeit des hypothetisierten Worts.
Allerdings ist der durchschnittliche Score nicht robust. An Stellen, an denen zum Beispiel ein Hintergrundgeräusch auftritt, passt keines der akustischen Modelle zu der Aufnahme, und dort entsteht dann automatisch eine schlechte Konfidenzbewertung. Um dies zu vermeiden, wird der durschnittliche Score des zu beurteilenden Wortes in Verhältnis zu den akustischen Scores von alternativen Wortmodellen gesetzt. Dieses Verhältnis (die sogenannte *likelihood ratio* wurde als - meist alleiniges - Merkmal in der Frühzeit der Entwicklung von Vertrauensmessern nahezu ausschließlich verwendet (vgl. S. 116).

Durch Umformung von 8.9 erhält man einen Ausdruck für die gesuchte Wahrscheinlichkeit $p(Allophon)$:

$$p(\text{Allophon}) = \frac{p(A|\text{Allophon})}{p(A)} \approx \frac{p(A|\text{Allophon})}{\sum_{\text{alle Allophone}} p(A|\text{Allophon})} \tag{8.10}$$

Unter Annahme der Unabhängigkeit voneinander kann daraus durch Mittelung über alle Frames innerhalb des Zeitbereiches des Wortes die Wahrscheinlichkeit des ganzen Wortes, gegeben die akustischen Modelle und die Aufnahme berechnet werden. Der Logarithmus dieser Wahrscheinlichkeit wurde als Merkmal 'NScore' (für 'normalisierte Score') untersucht.

$$\text{NScore} = E_t(\log(p(A|\text{Allophon})) - \log(\sum_{alle Allophone} p(A|\text{Allophone}))) \tag{8.11}$$

Um den Aufwand der Berechnung des Nennerterms in (8.10) zu reduzieren, kann dieser durch Verwendung z.B. eines freilaufenden Phonemerkenners [110] approximiert werden. Die a-priori-Wahrscheinlichkeit jedes Datenframes wird dann durch den akustischen Score des lokal besten Phonems angenähert. Experimente ergaben jedoch einen um 10% schlechteren linearen Korrelationskoeffizienten im Vergleich zur exakten Berechnung (8.11), so dass in dieser Untersuchung diese Approximation nicht angewendet wurde.

Ein weiteres Merkmal entsteht durch Division der logarithmierten Scores, das heißt durch Verhältnisbildung zwischen dem Score des Wortmodells und dem des alternativen Modells. Es wird mit *NScoreQ* bezeichnet und ist wie folgt definiert:

$$\text{NScoreQ} = E_t(\frac{\log(p(A|\text{Allophone}))}{\log(\sum_{alle Allophone} p(A|\text{Allophone}))}) \tag{8.12}$$

$E_t(.)$ symbolisiert den Erwartungswert gemittelt über die Zeit.

Signalabhängige Merkmale

Signal-Rausch-Verhältnis

Es ist anzunehmen, dass stark gestörte (verrauschte) Daten eine niedrigere Erkennungsrate aufweisen als solche ohne Hintergrundgeräusche. Das Signal-Rausch-Verhältnis liefert einen Meßwert für den Grad der Überlagerung eines Signals der Leistung $P(S)$ durch Rauschen der Leistung $P(N)$:

$$SNR = 10\frac{P(S)}{P(N)}[dB] \tag{8.13}$$

Der Schätzwert für die Rauschleistung $P(N)$ in (8.13) wird meist in den Sprechpausen geschätzt. Kommen keine Sprechpausen vor, kann die SNR in der Praxis nicht mehr zuverlässig berechnet werden.

Ein SNR von 0 dB bedeutet, dass das Signal und das Rauschen dieselbe Leistung aufweisen. Eine negative SNR bedeutet, dass die Energie des Rauschens diejenige des Signals übersteigt und zeigt eine stark gestörte Aufnahme an. Klare, wenig gestörte Aufnahmen weisen eine SNR von mehr als 25 dB auf; selbst unter guten Aufnahmebedingungen sind aber in einem normalen Raum mehr als 40-50 dB in der Regel nicht zu erreichen. Eine Faustregel besagt, dass die Worterkennungsrate bis zu einer SNR von 20 dB kaum, bei kleineren Signal-Rauschabständen dann aber drastisch absinkt.

Da die Leistung des Sprachsignals über die Dauer eines Wortes nicht zeitlich konstant ist, ist die SNR in Gl. 8.13 eine Funktion der Zeit. Um aus dieser zeitkontinuierlichen Funktion ein einzelnes, skalares Merkmal zu gewinnen, kann der Mittelwert der SNR über die Dauer des Wortes berechnet werden, aber auch beispielsweise das *Minimum* oder das *Maximum*. Untersuchungen ergaben, dass das (frameweise berechnete) Maximum des Signal-Rauschabstandes am stärksten mit der Fehlerrate korreliert. Dieses Maximum wurde daher als Merkmal mit in die Analyse aufgenommen.

$$SNR = \max_t(10\frac{P(S)}{P(N)}) \tag{8.14}$$

Akustische Entropie

Die akustischen Modelle erlauben die Berechnung einer a-posteriori-Wahrscheinlichkeit für jedes Allophon. Für jedes Datenframe einer Eingabe können diese Wahrscheinlichkeiten berechnet und als diskrete Wahrscheinlichkeitsverteilung aufgefasst werden. Die Gestalt dieser Verteilung erlaubt Rückschlüsse auf die Klassifikationsleistung der Modelle. Ähnelt die Verteilung beispielsweise der Gleichverteilung, dann sind die Modelle nicht in der Lage, der Eingabe ein Allophon eindeutig zuzuordnen. Ein Zeitbereich, bei dem alle (oder die meisten) Datenframes eine solche Gleichverteilung aufweisen, hat vermutlich eine geringere Chance, dass das darin enthaltene gesprochene Wort korrekt klassifiziert

wird, weil die akustischen Modelle keine klare Entscheidung zulassen.
Ein Maß, das die Ähnlichkeit einer Verteilung mit einer Gleichverteilung misst, ist die Entropie. Daraus läßt sich das Merkmal *akustische Entropie* ableiten:

$$\text{AEntropy} = E_t(\sum_{\text{alle Allophone a}} p_a \log(p_a)) \tag{8.15}$$

mit

$$p_a = p(\text{Observation}|\text{Allophon a}) \tag{8.16}$$

wobei die akustischen Emissionswahrscheinlichkeiten p_a für die Allophone normiert sind:

$$\sum_{\text{alle Allophone a}} p_a = 1 \tag{8.17}$$

Sprachmodellbasierte Merkmale

Bei der Schätzung der Wahrscheinlichkeiten für statistische Trigramm-Sprachmodelle müssen Terme der Form

$$p(Wort_3|Wort_1, Wort_2) \tag{8.18}$$

geschätzt werden. Bei (mindestens) 500.000 verschiedenen Wörtern im Deutschen müsste ein Text, der jedes dieser Ereignisse im Mittel auch nur einmal gesehen hätte, bereits $500000^3 = 1,25 * 10^{17}$ Worte lang sein. In der Praxis sind die verfügbaren Texte jedoch kleiner als 10^9 Worte (mehr zu statistischen Sprachmodellen siehe Seite 27 ff.).
Daraus folgt, dass eine auf gesehenen Daten beruhende Schätzung der Trigrammwahrscheinlichkeiten nur für einen Bruchteil der Trigramme möglich ist. Das Nichtvorhandensein eines Trigramms beinhaltet jedoch, wenn die zugrundeliegende Textmenge relativ groß ist, die Information, dass das entsprechende Worttripel nicht sehr gebräuchlich ist. Dasselbe gilt in noch stärkerem Maße für nicht beobachtete Bigramme, also Wortpaare. Taucht ein solches, nicht beobachtetes Bigramm oder Trigramm in der Hypothese des Spracherkenners auf, dann liegt entweder ein ungebräuchliches Konstrukt vor, oder es handelt sich um einen Erkennerfehler. In Anlehnung an [65] läßt sich hieraus ein ternäres Merkmal *LM-NGram* wie folgt definieren:

$$\text{LM-Ngram} = \begin{cases} 0 & : \text{Trigramm im Trainingstext enthalten} \\ 1 & : \text{Bigramm, jedoch nicht Trigramm im Trainingstext enthalten} \\ 2 & : \text{Bigramm im Trainingstext nicht enthalten} \end{cases} \tag{8.19}$$

Wortabhängige Merkmale

Wortlänge

Einige Merkmale lassen sich ohne vorheriges Wissen allein aus der Wortidentität ableiten. So ist beispielsweise bekannt, dass kurze Worte häufiger falsch erkannt werden als lange. Der Grund dafür ist, dass bei kurzen Wörtern in der Regel nur wenige Datenpunkte zur Klassifikation zur Verfügung stehen. Als Indikator für die Länge eines Wortes kann die Anzahl N_p seiner Phoneme herangezogen werden. Vorexperimente ergaben, daß $\log(N_p)$ stärker mit der Fehlerrate korreliert ist als N_p selbst. Daher wurde das Merkmal *LogNPhones* wie folgt definiert:

$$LogNPhones = \log(\text{Anzahl Phoneme des Wortes}) \tag{8.20}$$

Aussprachevariante

Ein weiteres nützliches Merkmal ist die Tatsache, ob es sich bei dem gefundenen Wort um die kanonische Variante oder eine - möglicherweise exotische - Aussprachevariante handelt. Eine Abweichung der gefundenen Aussprache von der gewöhnlichen Aussprache kann ein Hinweis auf einen Erkennungsfehler sein. Daraus leitet sich das binäre Merkmal *PronVar* ab:

$$PronVar = \begin{cases} 0 & : \text{ kanonische Aussprache} \\ 1 & : \text{ Aussprachevariante} \end{cases} \tag{8.21}$$

Andere, nicht suchraumbasierte Merkmale

Anzahl Vorkommen im Trainingskorpus

In [65] wurde eine negative Korrelation zwischen der Anzahl der Vorkommen eines Wortes in den akustischen Trainingsdaten und der Wahrscheinlichkeit der Fehlerkennung festgestellt. Das Merkmal *LogAnzImTraining* wird wie folgt definiert:

$$LogAnzImTraining = \log(\text{Anzahl Vorkommen des Wortes im Trainingsmaterial}) \tag{8.22}$$

Geräusch als Nachbar

Im Fall von spontaner Sprache kann ein Spracherkenner an Stelle eines Wortes auch ein Geräusch detektieren. Dabei kann es sich um eine Häsitation (Ähm und ähnliche Artefakte), um ein Atemgeräusch, ein Schmatzen, Husten oder Ähnliches handeln. Es ist anzunehmen, dass das Vorhandensein von solchen Ereignissen einen Einfluss auf die Erkennungsgenauigkeit der benachbarten Wort hat. Um diesen Effekt zu quantifizieren, wurde das Merkmal *NDisfluent* untersucht. Es ist als die Anzahl der an das betrachtete Wort angrenzenden gefundenen Geräusche definiert.

Sprechgeschwindigkeit

Das Merkmal *LogNPhones* misst die Länge des Wortes in Phonemen. Ein Wort mit einer gegebenen Anzahl von Phonemen kann aber mit unterschiedlicher Sprechgeschwindigkeit ausgesprochen werden. Bei schneller Sprechweise wird einerseits die Artikulationsgenauigkeit reduziert [45], und andererseits wird die Zahl der Eingabevektoren, die zur Klassifikation herangezogen werden können, verkleinert. Beide Effekte sorgen für eine Erhöhung der Wortfehlerrate bei steigender Sprechgeschwindigkeit. Es wurden insgesamt drei Merkmale untersucht, die diesen Effekt auszunutzen versuchen: *SprechGeschWPBQ*, *Duration*, und *AnzFrames*.

$$SprechGeschWPBQ = \frac{\text{tatsächliche Länge des Wortes in msec}}{\text{erwartete Länge des Wortes in msec}} \tag{8.23}$$

Die erwartete Länge eines Wortes wird dabei aus der Summe der mittleren Länge seiner Phoneme berechnet. Die mittleren Phonemlängen wurden auf dem Trainingsset durch Berechnung der Zeitzuordnung geschätzt.

Duration ist einfach als das Inverse der Sprechgeschwindigkeit definiert:

$$Duration = \frac{1}{\text{SprechGeschWPBQ}} \tag{8.24}$$

Das Merkmal *NFrames* mißt die Dauer des hypothetisierten Wortes in 10-Millisekunden-Inkrementen und ist damit sowohl von der Sprechgeschwindigkeit als auch von der Wortlänge abhängig:

$$AnzFrames = \text{Länge des Wortes in 10-msec-Frames} \tag{8.25}$$

Suchraumbasierte Merkmale

Anzahl der Kanten im Worthypothesengraph

Die Aufgabe der Suche in einem Spracherkenner ist es, aus der ausserordentlich großen Zahl an theoretisch möglichen Wortkombinationen diejenige mit der größten *a-posteriori*-Wahrscheinlichkeit $p(W_1..W_n|Sprachsignal)$ zu finden. Das zeitsynchrone Suchverfahren, das hierbei eingesetzt wird, verfolgt eine Vielzahl von konkurrierenden Hypothesen gleichzeitig und entscheidet sich erst am Ende der Äußerung für die wahrscheinlichste Wortfolge. Um den Rechenzeitaufwand klein zu halten, werden Methoden der Beschneidung des Suchraumes ('Pruning') eingesetzt. Die gebräuchlichste Strategie hierzu ist, alle Teilhypothesen, deren Wahrscheinlichkeit um eine vorher eingestellte Pruningschranke kleiner ist als die Wahrscheinlichkeit der lokal besten Teilhypothese, zu eliminieren.

Am Ende der Suche steht mithin eine große Zahl von möglichen Hypothesen zur Verfügung, wobei die Zahl dieser Hypothesen durch die Beschneidung des Suchraums

um viele Größenordnungen unter der Zahl der theoretisch möglichen Wortkombinationen liegt. Es ist nun zweckmäßig, diesen reduzierten Suchraum in kompakter Form als Ausgabe des Suchprozesses anzusehen. Die Darstellung des Suchraums kann dabei sehr effizient in Form eines Graphen erfolgen, dessen Knoten mit Zeitpunkten identifiziert werden, und dessen Kanten mit den hypothetisierten Wörtern markiert sind. Der Nachteil liegt darin, dass die beste Hypothese durch eine weitere Suche durch den Graphen bestimmt werden muß. Dem stehen aber so viele Vorteile gegenüber, dass die meisten neueren Spracherkennungssysteme Wortgraphen als ihre Standard-Ausgabeform wählen.

Einer der wichtigsten Vorteile ist es, dass der Suchraum, verglichen mit dem des ursprünglichen Problems, drastisch reduziert ist. Dadurch können zur Suche im Worthypothesengraphen Verfahren eingesetzt werden, die im ersten Suchdurchlauf zu aufwendig sind. Solche Verfahren können z.B. interpolierte Sprachmodelle, Sprachmodelle mit größerer Reichweite als Trigramme, oder auch aufwendig zu berechnende akustische Modelle sein.

Durch einen Kunstgriff kann die Vielzahl der Hypothesen, die in einem Worthypothesengraphen implizit kodiert sind, weiter gesteigert werden. Verschmilzt man Knoten, die zu verschiedenen Hypothesen gehören, jedoch zufällig auf denselben Zeitpunkt fallen, miteinander, so entstehen durch diese Verschmelzung neue Hypothesen, die vorher nicht im Graphen enthalten waren. Abb. 8.4 illustriert das Verfahren. In diesem Fall enthält der Wortgraph, der durch Verschmelzung von zwei Hypothesen entstand, acht Hypothesen. Die zusätzlichen Hypothesen entstehen durch die Verschmelzung der ersten beiden zeitgleichen Wortübergänge in jeweils einen gemeinsamen Knoten.

Bedingt durch die Strategie der Beschneidung des Suchraums, ist die Anzahl der konkurrierenden Worte, und damit von Kanten im Graphen, abhängig von der Ähnlichkeit der Bewertung der Teilhypothesen bis zu diesen Zeitpunkt. Liefern die akustischen und Sprachmodelle keine starke Diskriminanz, ist die Zahl der konkurrierenden Wörter groß. Wird im anderen Extremfall ein Wort A lokal deutlich besser bewertet als alle konkurrierenden Wörter, werden diese aus dem Suchraum entfernt, und der entsprechende Zeitbereich wird im Worthypothesengraphen nur noch durch eine einzige Kante - markiert mit dem Wort A - überspannt. Die Zahl der Kanten, die einen Zeitbereich überspannen, ist also eine Kennzahl für das Maß an Verwechselbarkeit der lokalen Hypothesen untereinander, und damit indirekt auch für die Wahrscheinlichkeit einer Fehlerkennung.

Die Zahl der Kanten im Worthypothesengraphen, die den Zeitbereich eines Wortes überspannen, lässt sich auf mehrere Arten definieren. In dieser Arbeit wurden die folgenden sechs Berechnungsvorschriften untersucht:

$$N(t_a + 1) = \text{Anzahl Kanten bei t} = 1 \text{ Frame nach Wortanfang} \tag{8.26}$$

$$N(t_a - 1) = \text{Anzahl Kanten bei t} = 1 \text{ Frame vor Wortanfang} \tag{8.27}$$

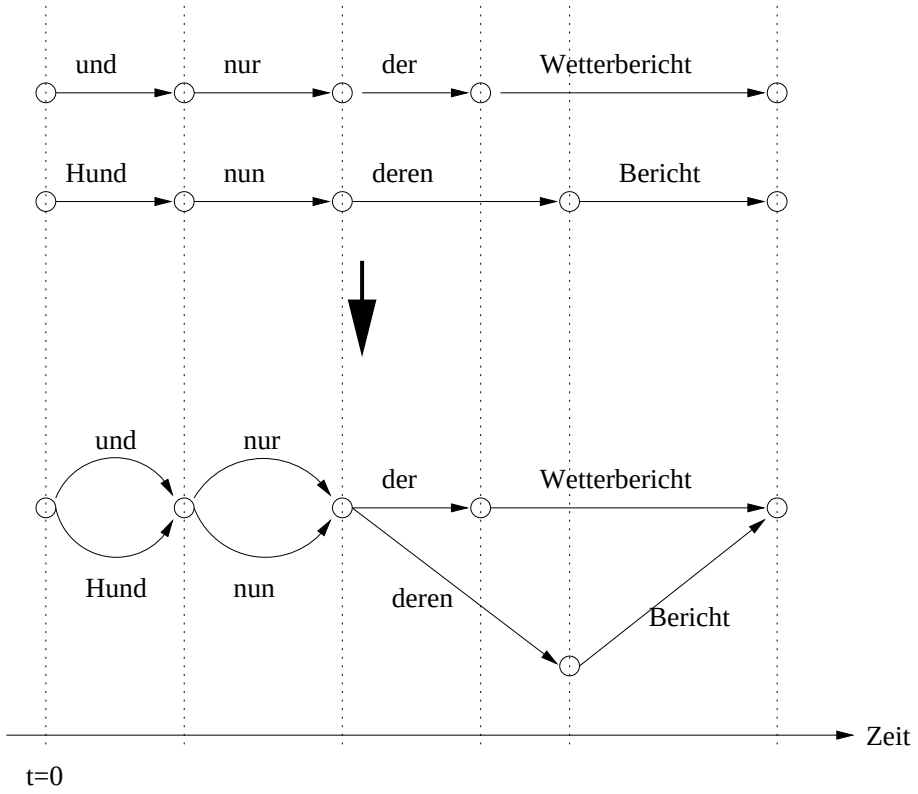


Abbildung 8.4: Aufbau eines Wortgraphen aus zwei Hypothesen

$$N(t_e - 1) = \text{Anzahl Kanten bei t = 1 Frame vor Wortende} \tag{8.28}$$

$$N(t_e + 1) = \text{Anzahl Kanten bei t = 1 Frame nach Wortende} \tag{8.29}$$

$$E(\log N) = \frac{1}{Anz.Frames} \sum_{t \in \text{Zeitbereich des Wortes}} \log(\text{Anzahl Kanten(t)}) \tag{8.30}$$

$$\log E(N) = \log(\frac{1}{Anz.Frames} \sum_{t \in \text{Zeitbereich des Wortes}} \text{Anzahl Kanten(t)}) \tag{8.31}$$

Anzahl aktiver Hypothesen während der Suche

Während der eigentlichen Suche, also der erstmaligen Einschränkung des Suchraumes, lässt sich die Zahl der aktiven Wortenden zu jedem Zeitpunkt leicht ermitteln.

Der Logarithmus dieser Anzahl, gemessen am Anfang und am Ende des betrachteten Wortes (jeweils gemittelt über 3 Frames), wurde unter den Bezeichnungen *LogAWEbeg-3* und *LogAWE-End* in den untersuchten Merkmalssatz aufgenommen.

Akustische Stabilität

Finke und Zeppenfeld [52] schlugen 1996 ein Verfahren zur Konfidenzberechnung vor, das sie als *akustische Stabilität* oder *AStabil* bezeichneten. Das Verfahren geht vom Worthypothesengraphen aus. Dabei werden aus dem Graphen mehrere (in der Größenordnung $N = 10$ bis 100) Hypothesen extrahiert, indem dem statistischen Sprachmodell jeweils ein unterschiedliches Gewicht verliehen wird und danach die wahrscheinlichste Hypothese für dieses Sprachmodellgewicht aus dem Worthypothesengraphen gesucht wird. Die Sprachmodellgewichte variieren dabei um das - auf der Kreuzvalidierungsstichprobe bestimmte - optimale Sprachmodellgewicht. Die N Hypothesen werden einzeln mit dynamischer Programmierung mit der besten Wortfolge (also der bei optimalem Sprachmodellgewicht aus dem Worthypothesengraphen extrahierten) verglichen. Für jedes Wort $W_i (i = 1, 2, ..., M)$ der zu beurteilenden Hypothese wird gezählt, in wievielen der N Hypothesen es enthalten ist - minimal in keiner einzigen davon, höchstens in allen N. Diese Maßzahl wird durch Division durch N auf den Bereich 0..1 normiert und stellt das Merkmal akustische Stabilität dar.

Die akustische Stabilität ist bemerkenswert gut mit der Fehlerwahrscheinlichkeit korreliert. Der Grund dafür ist, dass in die Bewertung beide Wissensquellen des Spracherkenners, die akustischen Modelle und das Sprachmodell, eingehen. Nachteilig ist der relativ große Berechnungsaufwand, wenn N groß gewählt wird.

8.7 Das Merkmal gamma

Im Rahmen dieser Arbeit wurde ein Verfahren zur Konfidenzberechnung erarbeitet, das auf dem Worthypothesengraphen basiert.

Die Grundidee dabei ist, die akustischen und die Sprachmodell-Bewertungen des Worthypothesengraphen in Wahrscheinlichkeiten umzurechnen und die Gesamtwahrscheinlichkeit des Worthypothesengraphen auf 1 zu normieren. Das impliziert die Annahme, dass der Worthypothesengraph sämtliche möglichen Hypothesen enthält - nur dann ist diese Normierungsbedingung per Definition erfüllt:

$$\sum_{\text{alle Wortfolgen H}} p(H|O) = 1 \tag{8.32}$$

Ein Worthypothesengraph endlicher Größe enthält natürlich nicht alle möglichen Sätze, die in einer Sprache gebildet werden können. Die auf die fehlenden Wortfolgen entfallende Wahrscheinlichkeitsmasse fehlt damit in Gl. (8.32). Betrachtet man die

Menge H aller möglichen Wortfolgen, trägt jede einzelne davon zur Summe (8.32) bei.

Bei adäquater Modellierung ist die Wahrscheinlichkeit einer beliebigen Wortfolge, die dem tatsächlich Gesagten wenig ähnelt, allerdings ausserordentlich klein. Der Fehler, der durch Weglassen dieser Wortfolge in der Summe in Gl. 8.32 gemacht wird, ist dann vernachlässigbar.

Nimmt man an, dass der durch die endliche Graphengröße verursachte Fehler tatsächlich klein ist, d.h. dass die Gesamtwahrscheinlichkeit aller Hypothesen im Worthypothesengraphen zusammengenommen (nahezu) 1 ist, lässt sich die Wahrscheinlichkeit für jedes einzelne Wort im Graphen mit dem forward-backward Algorithmus [124] berechnen. Dazu wird der Worthypothesengraph in ein HMM umgewandelt. Den Zuständen des HMM entsprechen die Worte. Sequenzen von Zuständen entsprechen Folgen von Wörtern. Die Übergangswahrscheinlichkeiten von einem Zustand zum nächsten, und damit von einem Wort zum nächsten, werden durch das statistische Sprachmodell gegeben. Die Emissionswahrscheinlichkeiten des HMM entsprechen den akustischen Bewertungen. Im folgenden Beispiel wird dieser Prozess an einem einfachen Worthypothesengraphen detailliert erläutert.

Der Worthypothesengraph in Bild 8.5 sei die Ausgabe des Spracherkenners.

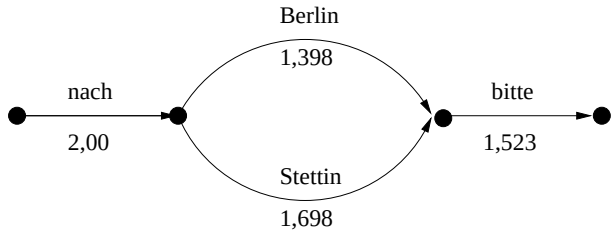


Abbildung 8.5: Einfacher Worthypothesengraph

Über den Kanten sind die Worte der Hypothese notiert, unter den Kanten die Bewertung ('score') durch den Spracherkenner, wie üblich als negativer Logarithmus der Wahrscheinlichkeit. Der Graph kodiert lediglich 2 Hypothesen, 'nach Berlin bitte' und 'nach Stettin bitte', die z.B. aus einem Zug-Auskunft-Szenario stammen könnten. Das verwendete (Bigramm-) Sprachmodell weise folgende Wahrscheinlichkeiten auf:

$$
\begin{array}{lcl}
\text{p(nach)} & = & 0{,}20 \\
\text{p(Berlin|nach)} & = & 0{,}25 \\
\text{p(Stettin|nach)} & = & 0{,}40 \\
\text{p(bitte|Berlin)} & = & 0{,}50 \\
\text{p(bitte|Stettin)} & = & 0{,}50
\end{array}
$$

Bei der Umwandlung in ein HMM wird nun jede Kante, also jedes Wort, in

einen Zustand des HMMs verwandelt. Die zugehörige Observationswahrscheinlichkeit $p(O|W)$ des HMM-Zustandes wird durch Exponentiation des 'score' des Erkenners berechnet. Nach der Emission der Observablen (also des Sprachsignals bis zum nächsten Wort) geht das HMM in den nächsten Zustand über.
Im Falle des ersten Wortes ('nach') gibt es zwei mögliche Nachfolger ('Berlin' und 'Stettin'). Die Übergangswahrscheinlichkeiten für den Übergang von einem Wort in ein anderes werden vom statistischen Sprachmodell geliefert. Die Wahrscheinlichkeit, nach 'nach' das Wort 'Stettin' zu sehen, ist 40%. Analog wird mit allen anderen Wortübergängen verfahren. Das vollständig umgewandelte HMM des Wortgraphen (8.5) sieht damit wie in Bild 8.6 dargestellt aus (die Wahrscheinlichkeiten an den HMM-Zuständen entstehen durch delogarithmieren der Scores in Abb. 8.5).

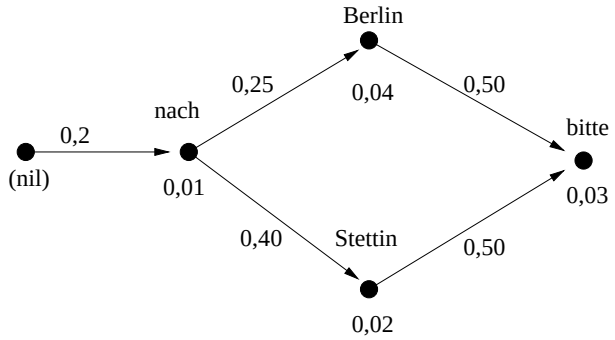


Abbildung 8.6: HMM, aus Worthypothesengraph erstellt

Aus dem entstandenen HMM kann nun die a-posteriori-Wahrscheinlichkeit für jedes Wort darin berechnet werden. Dies geschieht durch den Einsatz des forward-backward-Algorithmus, der im Folgenden exemplarisch dargestellt wird. Der forward-backward-Algorithmus besteht aus einer Anwendung des **Forward**-Algorithmus, einer Anwendung des **Backward**-Algorithmus und einer anschließenden Verrechnung der Zwischenresultate dieser beiden Algorithmen. Eine genaue Erläuterung des forward-backward Algorithmus und der verwendeten Bezeichnungen findet sich auf Seite 41.
Der *Forward-Algorithmus* hat das Ziel, zu jedem Zeitpunkt - gegeben die Observation - die Wahrscheinlichkeit zu berechnen, dass das HMM einen bestimmten Zustand erreicht. Diese Wahrscheinlichkeiten werden forward-Variablen oder *alphas* genannt. Die Wahrscheinlichkeit des ersten (initialen) Zustands wird dabei mit 1.0 initialisiert: Jede mögliche Wortfolge, also jede mögliche Zustandsfolge, beginnt mit dem Nullzustand am Anfang. Nach der Initialisierung wird die Wahrscheinlichkeit der möglichen Folgezustände entsprechend der Vorschrift in (4.28) berechnet. Dabei werden die Wahrscheinlichkeiten der Zustandsübergänge (die a_{ik}) durch die Sprachmodell-Wahrscheinlichkeiten gegeben. Die vollständig gefüllte Matrix der *alphas* ist in Bild 8.7 dargestellt.

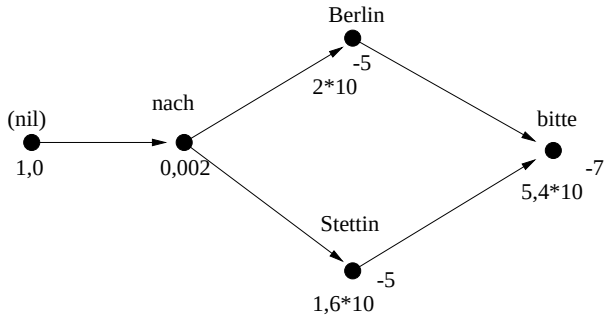


Abbildung 8.7: Berechnung der Alphas

Der *Backward-Algorithmus* hat zum Ziel, zu jedem Zeitpunkt - gegeben die Observation *ab diesem Zeitpunkt* - die Wahrscheinlichkeit zu berechnen, dass das HMM in einem bestimmten Zustand ist. Analog zum forward-Algorithmus wird der Endzustand mit der Wahrscheinlichkeit 1.0 initialisiert. Danach wird der Graph von hinten nach vorne analog zum Forward-Algorithmus durchlaufen. Das Endergebnis zeigt Bild 8.8.

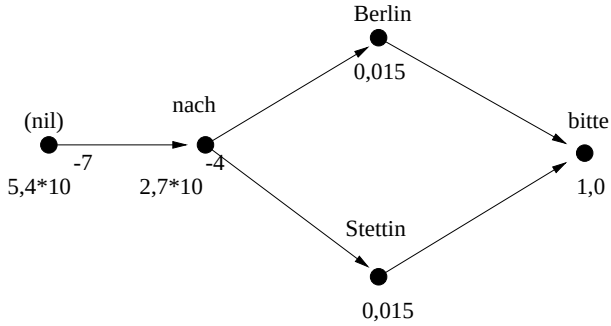


Abbildung 8.8: Berechnung der Betas

Wie in Gleichung 4.35 auf Seite 43 verlangt, ist β_0 gleich α_T (gleich $5,4 * 10^{-7}$). Die α_i geben die Wahrscheinlichkeit an, sich im Zustand i zu befinden, gegeben die Observationen bis dorthin, während die β_i die Wahrscheinlichkeit angeben, sich im Zustand i zu befinden, gegeben die Observationen ab diesem Moment. Multipliziert man die α_i mit den β_i, ergibt sich die Wahrscheinlichkeit, in diesem Zustand zu sein, gegeben die gesamte Observation; teilt man diesen Wert durch die Wahrscheinlichkeit der Observation $p(O|\lambda)$, ergibt sich die Wahrscheinlichkeit des Zustandes zu diesem Zeitpunkt. Dies entspricht der Definition der γ_i auf Seite 43. Bild 8.9 zeigt das HMM mit den γ_i von jedem Zustand.

Die Wahrscheinlichkeit für das Wort 'bitte' beträgt nach dieser Berechnung 1,0. Gegeben die Modellierung des Spracherkenners, ist also an dieser Stelle der Observation das Wort 'bitte' mit 100%iger Wahrscheinlichkeit gesprochen worden. Dies

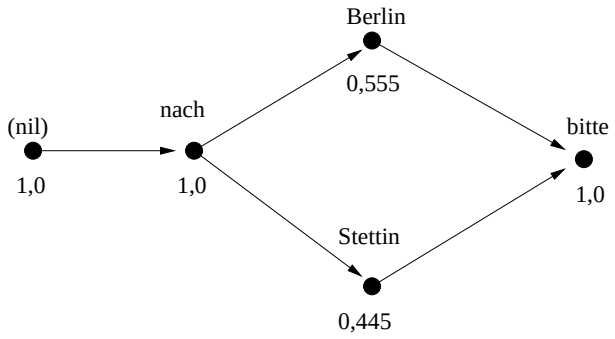


Abbildung 8.9: Berechnung der Gammas

entspricht auch der Intuition, weil beide möglichen Hypothesen an dieser Stelle übereinstimmen.

Wäre die Modellierung im Spracherkenner perfekt, dann wären die auf diese Art berechneten γ_i direkte Indikatoren für die Wahrscheinlichkeit, an der jeweiligen Stelle im Eingangssignal das Wort i zu beobachten. Die Modelle des Erkenners sind jedoch niemals perfekt, so dass die so errechneten Wahrscheinlichkeiten mehr oder weniger von den tatsächlichen Wahrscheinlichkeiten abweichen. Wie bereits oben erwähnt, ergibt sich durch die Beschneidung des Suchraums während der Suche (das *Pruning*) eine weitere Fehlerquelle. Nach der Beschneidung des Suchraumes wurde z.B. die Hypothese 'Berlin Berlin bitte' nicht mehr in den Ausgabegraphen übernommen, weil der entsprechende Pfad im Suchprozess eliminiert wurde. Die Hypothese 'Berlin Berlin bitte' hat zwar eine sehr viel kleinere Wahrscheinlichkeit als die beiden im Graphen verbleibenden Hypothesen; allerdings ist ihre Wahrscheinlichkeit größer als Null, so dass durch die Beschneidung des Suchraumes ein Fehler verursacht wird. Wie die Experimente auf Seite 138 zeigen, ist der Einfluss dieses Fehlers jedoch bei geeigneter Wahl der Pruning-Schranken klein.

Experimente

Alle Experimente in diesem Abschnitt wurden mit dem Erkenner 'I39' und einer Teststichprobe von 4 Sendungen 'tagesschau' (30.3.97, 13.4.97, 28.5.97 und 30.6.97) durchgeführt. Keinerlei Maßnahmen zur Beschleunigung des Erkenners wurden eingesetzt. Die Ergebnisse werden jeweils als NCREP (siehe Seite 113) und Konfidenzfehlerrate CER angegeben (die Konfidenzfehlerrate ist definiert als eins minus Konfidenzakkuratheit CA (siehe Seite 112)). Das Baseline-Ergebnis mit dem Erkenner ohne Konfidenzmaß ist in Tabelle 8.1 dargestellt. Die Wortkorrektrate (siehe Seite 46) liegt bei diesem Erkenner bei 76,9%. A-priori wird daher jedes Wort als 'korrekt' markiert. Dabei werden 23,1% (100% minus 76,9%) der Worte falsch markiert. Die Entropie eines Zufallsprozesses, der zwei Zeichen mit einer Wahrscheinlichkeit von 0,231 und 0,769 ausgibt, liegt bei 0,54 bit. Diese Zahl wird zur Berechnung der

normalisierten Kreuzentropie (Seite 113) benötigt.

CER	Entropie H0
23.1%	0.54

Tabelle 8.1: Ergebnisse des Erkenners ohne Konfidenzmaße

In sämtlichen im Folgenden beschriebenen Experimenten wurde das Merkmal *gamma* auf den kompletten Ausgabegraphen des Erkenners berechnet. Für die Worte der besten Hypothese wurden die gamma-Werte extrahiert und jedes Wort mit seinem gamma-Wert konfidenzannotiert. Konfidenzwerte über 0,5 wurden als Entscheidung 'dieses Wort wurde richtig erkannt' gewertet, während Konfidenzwerte unter 0,5 als Entscheidung 'dieses Wort wurde fehlerhaft erkannt' interpretiert wurden. Mit Hilfe der - manuell erstellten - Referenz wurden dann die Entscheidungen bewertet und die Maße Konfidenzakkuratheit CA (bzw. die Konfidenzfehlerrate CER = 1-CA) und NCREP berechnet.

Skalierung

Berechnet man mit dem beschriebenen Verfahren die Werte für *gamma* in den Ausgabe-Worthypothesengraphen, ergeben sich die folgenden Leistungskennzahlen für das Merkmal *gamma*:

Merkmal	CER	Entropie H0
keines	23.1%	0.54
gamma	30.5%	-0.74

Tabelle 8.2: Gamma ohne Skalierung

Die Performanz des Merkmals ist sehr schlecht. Der Grund hierfür liegt darin, dass die akustischen 'Scores' des Erkenners nicht ohne weiteres als logarithmierte Wahrscheinlichkeiten zu interpretieren sind, sondern zusätzlich eine Skalierung im logarithmischen Bereich aufweisen. Im Forward-Algorithmus werden die Scores jedoch mit den Sprachmodellwahrscheinlichkeiten multipliziert und dann zusammenaddiert, um die Wahrscheinlichkeit zu berechnen, daß ein gegebener Knoten erreicht wird. Diese Addition ist aber nur dann möglich, wenn es sich bei den zu addierenden Werten tatsächlich um unskalierte Wahrscheinlichkeiten handelt.

Auch während der eigentlichen Erkennung muß die Skalierung der akustischen Scores berücksichtigt werden, damit die Sprachmodellwahrscheinlichkeiten und die akustischen Wahrscheinlichkeiten richtig miteinander verrechnet werden können. Hierzu wird in aller Regel ein Sprachmodellgewicht eingeführt, mit dem die (logarithmierten) Sprachmodellwahrscheinlichkeiten multipliziert werden. Dieses Sprachmodellgewicht wird üblicherweise auf einer Kreuzvalidierungsstichprobe ermittelt. Für

das Sprachmodell des View4You-Erkenners ergab sich ein optimaler Wert von 24. Anstatt das Sprachmodell mit 24 zu multiplizieren, können die akustischen Scores auch mit $\frac{1}{24} = 0,042$ multipliziert werden.

Tabelle 8.3 fasst die Ergebnisse mit unterschiedlichen Skalierungen der akustischen Scores zusammen.

Skalierungsfaktor	CER	NCREP
baseline	23,1%	0
0,01	27,3%	-0,01
0,015	22,6%	0,09
0,02	19,6%	0,15
0,025	17,7%	0,175
0,03	17,3%	0,174
0,04	18,2%	0,137
0,05	19,5%	0.083

Tabelle 8.3: Einfluss des Skalierungsfaktors

Zusammenfassend kann festgestellt werden, dass die richtige Skalierung der akustischen Scores für das Funktionieren des Merkmals *gamma* entscheidend ist.

Konkurrierende Kanten

Bei dem Worthypothesengraphen, der in Bild 8.10 dargestellt ist, konkurrieren im letzten Zeitsegment zwei Kanten 'bitte' miteinander. Diese beiden können nicht verschmolzen werden, da der Anfangszeitpunkt unterschiedlich ist.

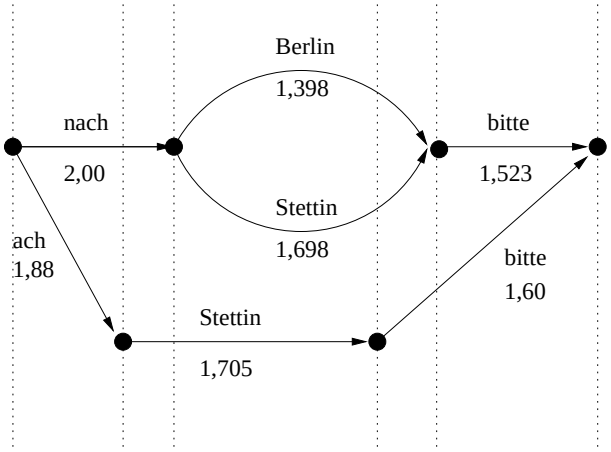


Abbildung 8.10: Worthypothesengraph mit überlappenden Kanten

Berücksichtigt man die zweite 'bitte'-Kante bei der Berechnung der Wahrscheinlichkeit des Wortes 'bitte' als letztes Wort der Hypothese 'nach Berlin bitte' nicht, so

ergibt sich eine Wahrscheinlichkeit von weniger als 1.0, obwohl die mit dem finalen 'bitte' konkurrierende Hypothese auch 'bitte' lautet. Daher sollten die a-posteriori-Wahrscheinlichkeiten der 'bitte'-Kanten addiert werden. Mit der Addition ergibt sich als a-posteriori-Wahrscheinlichkeit für 'bitte' wieder das erwünschte Ergebnis 1,0.

Dabei entsteht allerdings das Problem der Addition von nur teilweise überlappenden Kanten. Sollte beispielsweise im Falle des Graphen in Bild 8.11 die Wahrscheinlichkeit der unteren 'bitte'-Kante zu der der oberen hinzuaddiert werden, obwohl die beiden nur in einem kleinen Bereich überlappen?

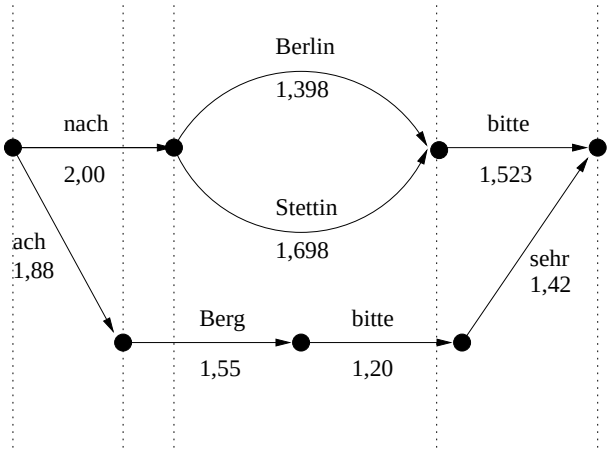


Abbildung 8.11: Worthypothesengraph mit überlappenden Kanten

Zur experimentellen Klärung dieser Frage wurden mehrere Experimente durchgeführt. Dabei wurden ein konkurrierendes Wort genau dann addiert, wenn

- es in einem Zeitbereich von $\pm t_a$ Frames am Wortanfang des betrachteten Wortes anfing, und
- es in einem Zeitbereich von $\pm t_e$ Frames am Wortende des betrachteten Wortes aufhört.

Ein negativer Zeitbereich bedeutet dabei, dass der Zeitpunkt des Anfangs bzw. Endes des konkurrierenden Wortes nicht beachtet wurde.

Eine andere untersuchte Möglichkeit ist die des Schwerpunkts. Als Schwerpunkt wird derjenige Zeitpunkt aufgefasst, der genau in der Mitte des betrachteten Wortes liegt. Wird beispielsweise das Wort 'bitte' von der Suche im Zeitbereich $4,0s < t < 6,0s$ gefunden, liegt der Schwerpunkt bei $t = 5,0s$. Andere Wortkanten werden dann als konkurrierend aufgefasst, wenn sie den Zeitpunkt des Schwerpunktes überspannen, d.h. wenn sie zum Zeitpunkt $t = 5,0s$ aktiv sind.

Die Ergebnisse - jeweils mit einem Skalierungsfaktor von 0,025 berechnet - sind in Tabelle 8.4 zusammengefasst. Man erkennt, dass vor allem die normalisierte Kreu-

zentropie NCREP durch das Addieren paralleler Kanten mit gleicher Wortidentität stark ansteigt.

Die niedrigste Konfidenzfehlerrate ergibt sich bei Anwendung der Schwerpunktmethode; die beste NCREP bei einer $t_a = t_b = 30ms$.

t_a	t_e	CER	NCREP
0	0	18.0%	0.233
-1	0	17,9%	0,241
0	-1	16,4%	0,252
1	1	17.2%	0,255
3	3	16,5%	0,267
6	6	16,5%	0,265
3	-1	16,4%	0,260
-1	-1	17,2%	0,154
Schwerpunkt	-	16,3%	0,261

Tabelle 8.4: Ergebnisse mit unterschiedlicher Berechnung paralleler Kanten

Einfluss des Prunings

Um den Einfluss der endlichen Größe der Worthypothesengraphen auf die Qualität des Merkmals *gamma* zu evaluieren, können Ergebnisse auf verschieden großen Graphen ermittelt und diese Ergebnisse dann ggf. extrapoliert werden. Zu diesem Zwecke werden zunächst möglichst große Worthypothesengraphen erzeugt. Diese werden dann sukzessive verkleinert, und auf jeder entstehenden Graphengröße wird das Merkmal *gamma* evaluiert. Die Verkleinerung der Graphen geschieht dabei auf folgende Weise. Zunächst wird der beste Pfad durch den Graphen gesucht und dessen a-posteriori-Wahrscheinlichkeit ('Score') berechnet. Danach wird für jedes Wort im Graphen der beste Pfad berechnet, der dieses Wort enthält. Die a-posteriori-Wahrscheinlichkeit dieses Pfades wird berechnet und mit dem Score des besten Pfades durch den gesamten Graphen verglichen. Liegt der Pfadscore um mehr als eine vordefinierte Pruning-Schwelle schlechter als der beste Score des global besten Pfades, so wird das entsprechende Wort gelöscht. Diese Löschung zieht in der Regel offene Enden im Graphen nach sich, weswegen in einem weiteren Schritt alle Knoten und Kanten getilgt werden, die auf keinem Pfad vom Anfang zum Ende liegen. Bei diesem Verfahren ist es durch Wahl der Pruning-Schwelle möglich, Wortgraphen kontinuierlich kleiner zu machen, ohne die beste Hypothese durch den Graphen dadurch zu ändern.

Als Maß für die Dichte eines Worthypothesengraphen sind mehrere Merkmale denkbar. In dieser Arbeit wird als Worthypothesengraphdichte der Quotient aus der Anzahl der unterschiedlichen Worte im Graphen und der Anzahl der Worte in der Referenz verstanden. Ein Worthypothesengraph mit 50 Wortknoten, der die Hypothese für einen 10 Worte langen Satz zusammenfasst, hätte also eine Dichte von 5.

Bild 8.12 und Bild 8.13 zeigen die Konfidenzakkuratheit und das Maß NCREP über der Graphendichte. Die zugrunde liegenden Daten sind auch in Tabelle 8.5 zusammengefasst.

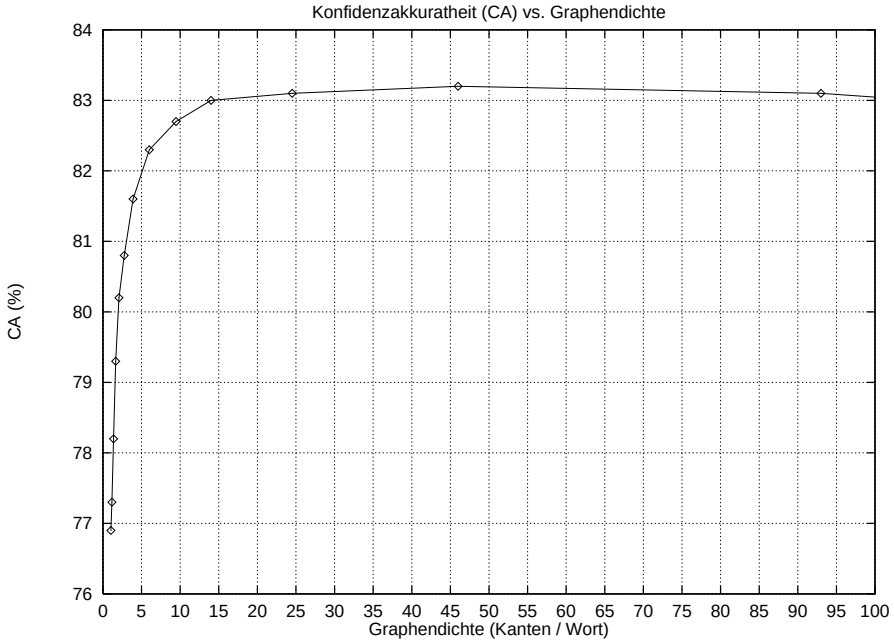


Abbildung 8.12: Konfidenzakkuratheit über Worthypothesengraph-Dichte

Die Konfidenzakkuratheit ohne Verwendung eines Konfidenzmaßes beträgt für dieses Testset 76,9%.

Man stellt fest, dass sowohl Konfidenzakkuratheit als auch NCREP ab einer Graphendichte von etwa 15 Wörtern pro Referenzwort nicht mehr ansteigen. Daraus kann geschlossen werden, dass die Näherung (8.32), die der Verwendung des Merkmals *gamma* zugrundeliegt, oberhalb dieser Graphendichte sehr gut erfüllt ist. Auffällig ist der Abfall der NCREP bei kleinen Graphendichten bis in den negativen Bereich. Dies ist dadurch zu erklären, dass bei abnehmender Graphendichte immer mehr Kanten *keine* parallel laufenden, konkurrierenden Kanten mehr aufweisen. Allen diesen Kanten muß das Konfidenzmaß den Wert 1.0, bzw. den oberen Grenzwert, der in diesem Experiment auf 0.95 gesetzt wurde, zuweisen. Jeder Fehler in dieser Zuweisung bewirkt eine starke Reduktion der Kreuzentropie, so dass ihr Gesamtwert negativ wird.

Arbeiten anderer Gruppen

Das hier vorgeschlagene Konfidenzmaß wurde vom Autor erstmals 1997 [26] vorgestellt. Es wurde danach auch von anderen Forschungsgruppen eingesetzt ([20], [9]). Wendemuth und Kollegen versuchten in [9], die Performanz eines Konfidenztaggers durch Kombination mehrerer Eingangsmerkmale zu verbessern. Ihre Analyse ergab,

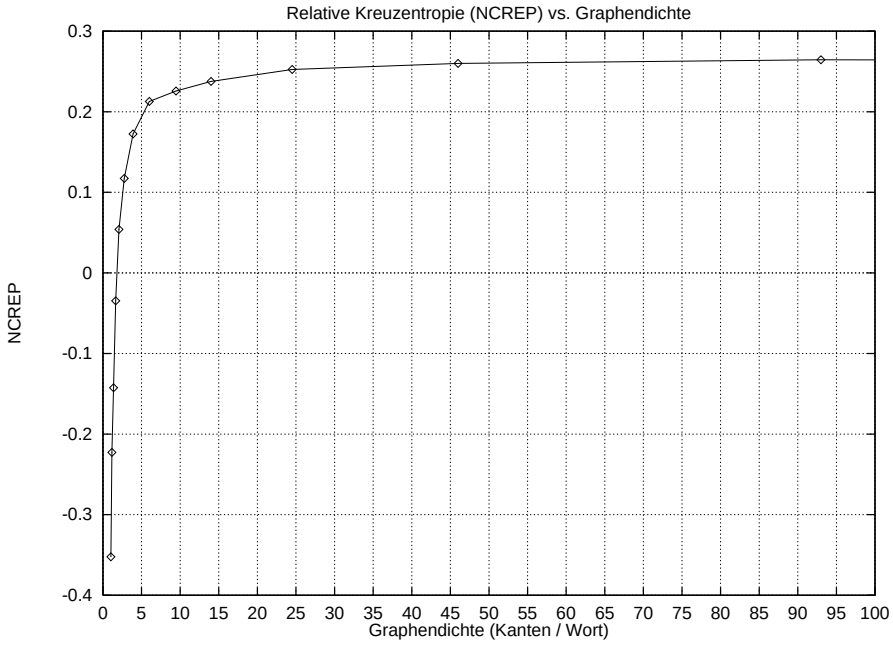


Abbildung 8.13: Relative Kreuzentropie über Worthypothesengraph-Dichte

dass die Kombination anderer Wissensquellen mit *gamma* nur noch eine geringe Steigerung der Konfidenzakkuratheit bewirkte.

[..] the best single large vocabulary confidence measure [..] is already very efficient since it uses the word history, a further improvement is therefore more difficult.

8.8 Experimente

Die im vorhergehenden Abschnitt beschriebenen potentiellen Merkmale wurden auf ihre Wirksamkeit hin untersucht. Dazu wurde auf einer unabhängigen Teststichprobe von 5940 Wörtern (39 Minuten Sprache) ein Erkennerlauf durchgeführt und jedes der Merkmale für jedes Wort der Erkennerhypothesen berechnet. Mit Hilfe der Referenz wurden die linearen Korrelationskoeffizienten und die Kontingenzkoeffizienten jedes Merkmals mit der Fehlerrate berechnet. Die Ergebnisse sind in der Tabelle 8.6 zusammengefasst.

Graphendichte	NCREP	CA
145	0.263	83.0
133	0.264	83.0
125	0.264	83.0
124	0.264	83.0
119	0.264	82.9
93	0.264	83.1
46	0.260	83.2
24.5	0.253	83.1
14	0.238	83.0
9.46	0.226	82.7
6	0.213	82.3
3.9	0.173	81.6
2.75	0.117	80.8
2.07	0.054	80.2
1.65	-0.035	79.3
1.37	-0.143	78.2
1.16	-0.223	77.3
1.02	-0.352	76.9

Tabelle 8.5: Abhängigkeit der Performanz von gamma von der Graphengröße

Man erkennt, dass die wortgraphbasierten Merkmale die größte Korrelation zur Fehlerwahrscheinlichkeit aufweisen, für eine Klassifikation also die nützlichsten sind. Das Merkmal mit der stärksten Korrelation ist *gamma*.

8.9 Untersuchte Klassifikatoren

In den vorangegangenen Abschnitten wurden Merkmale auf ihre Korrelation mit der Wortfehlerrate und damit auf ihre Eignung für den Aufbau eines Vertrauensmessers untersucht. Im folgenden Abschnitt geht es um den tatsächlichen Aufbau eines Vertrauensmessers, also um die Kombination der Merkmalswerte in einem Klassifikator (im Fall einer diskreten Ausgabe falsch/richtig) bzw. einem Regressor.

Eine besonders einfache Art von Klassifikatoren sind die *linearen Klassifikatoren*. Sie erzielen dann optimale Ergebnisse, wenn die Klassen im Merkmalsraum linear separierbar sind. Wie aber bereits in Bild 8.3 auf Seite 118 dargestellt wurde, ist die lineare Separierbarkeit nicht für alle Merkmale gegeben. Aus diesem Grund sollte ein nichtlinearer Klassifikator bessere Ergebnisse bei der Bestimmung der Wortkonfidenz liefern können, als ein linearer.

Im Rahmen dieser Arbeit wurden sowohl ein linearer, transformationsbasierter Klassifikator als auch ein nichtlinearer Klassifikator auf Basis eines neuronalen Netzes evaluiert. Die beiden Ansätze werden im Folgenden beschrieben.

Lineare Klassifikatoren

Eine einfache, aber für viele Anwendungen ausreichende Entscheidung zwischen zwei Klassen kann durch

$$g(\vec{x}) = \vec{w}\vec{x} \tag{8.33}$$

realisiert werden, auch *linearer Klassifikator* oder *Perzeptron* genannt. Man entscheidet sich dabei für Klasse ω_1, falls $g(\vec{x}) < t$ gilt, sonst für Klasse ω_2. Die Bestimmung des Gewichtsvektors $\vec{w}$ bezeichnet man als *Training* des Perzeptrons. Ein Perzeptron lässt sich beispielsweise durch ein neuronales Netz simulieren. Das Training eines solchen Netzes erfordert allerdings zahlreiche Iterationen, und das Ergebnis ist abhängig vom Startwert. Daher wurde zur Bestimmung des Gewichtsvektors ein anderer Weg gewählt.

Ausgangspunkt ist eine Menge von Trainingsbeispielen, bei denen die korrekte Klassenzugehörigkeit (richtig/falsch erkannt) bekannt ist. Die Trainingsmenge wird, entsprechend der Klassenzugehörigkeit richtig/falsch, in zwei Teilmengen aufgeteilt. Auf diesen beiden Klassen wird eine lineare Diskriminanzanalyse (LDA, siehe Seite 33) berechnet. Diese transformiert den Eingangsvektor $\vec{x}$ linear in einen Zielmerkmalsraum Y: $\vec{y} = A\vec{x}$ so, dass bei gleichbleibender Gesamtkovarianz die Klassenkovarianz minimiert wird. Ergebnis der LDA ist die Transformationsmatrix A, die im Fall eines Zweiklassenproblems nur den Rang 1 hat. Die erste Zeile von A stellt gerade den gesuchten Vektor $\vec{w}$ dar.

In einem zweiten Schritt wird (8.33) auf alle Werte der Trainingsstichprobe angewendet und diejenige Grenze t bestimmt, bei der die resultierende Fehlerrate minimiert wird.

Neuronale Netze

Mehrschichtige neuronale Netze, sogenannte *multilayer perceptrons*, sind die heute gebräuchlichsten nichtlinearen Klassifikatoren. Ein mehrschichtiges neuronales Netz besteht aus einem oder mehreren Knoten pro Schicht, wobei die Ausgabe der ersten Schicht die Eingabe der zweiten Schicht darstellt und so weiter. Theoretische Untersuchungen haben gezeigt, dass bereits ein Netz mit einer einzigen Zwischenschicht (dem sog. 'hidden layer') zwischen Eingabewerten und Ausgabe-Knoten beliebig komplexe Entscheidungsebenen finden kann [121].

Jeder Knoten eines neuronalen Netzes hat N ($N \geq 1$) Eingänge e_i, die mit Eingangsgewichten w_i belegt sind, und einen Ausgang, dessen Ausgabewert über die sogenannte *Aktivierungsfunktion* f von den Eingabewerten abhängt: $y =$

$f(e_1, e_2, e_3..., e_N)$. Für die Aktivierungsfunktion wird sehr häufig die *Sigmoidfunktion* gewählt:

$$y = \frac{1}{1 + e^{-x}} \tag{8.34}$$

$$x = \sum_{i=1}^{N} e_i w_i \tag{8.35}$$

Hat man eine Topologie für das neuronale Netz gewählt, werden die Gewichte w_i jedes Knotens mit Hilfe der Trainingsstichprobe optimiert. Diese Optimierung erfolgt mit einem Gradienten-Abstiegsverfahren, dem sogenannten *Backpropagation*-Algorithmus. Details über diesen Algorithmus, sowie weitere Literaturverweise auf die umfangreiche Literatur zum Thema Neuronale Netze, findet sich z.B. in [35].

Für den Vertrauensmesser wurden unterschiedliche Topologien von neuronalen Netzen trainiert. Dabei stellte sich heraus, dass mit einer sehr einfachen Struktur bereits hervorragende Ergebnisse zu erzielen waren. Die Topologie des verwendeten Netzes - beispielhaft für 2 Eingabemerkmale dargestellt - zeigt Bild 8.14.

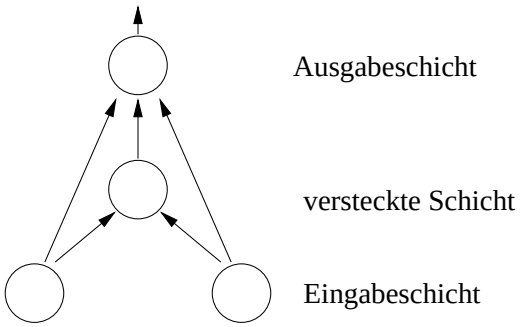


Abbildung 8.14: Struktur des eingesetzten neuronalen Netzes

Durch die direkten Verbindungen von der Eingabe- zur Ausgabeschicht kann dieses Netz bereits das XOR-Problem lösen und ist damit kein linearer Klassifikator mehr.

8.10 Evaluation

Auf einer Stichprobe von 39 Minuten spontaner deutscher Sprache aus der Terminabsprachedomäne wurden die verschiedenen Klassifikatoren evaluiert. Die Testmenge enthielt 5940 Wörter, die Wort-Korrekt-Rate des eingesetzten Erkenners betrug 85.3%.

Zunächst wurden die beiden Merkmale mit dem größten Kontingenzkoeffizienten in Tabelle 8.6, *gamma* und *A-Stabil*, einzeln evaluiert. In diesem Fall ist kein Klassifikator notwendig, es muss lediglich die optimale Grenze t festgelegt werden. Die resultierenden Konfidenzakkuratheiten lagen bei 88.0% für *gamma* und bei 87.4% für *A-Stabil*.

In einer weiteren Versuchsreihe wurden verschiedene Kombinationen von Merkmalen mit dem linearen Klassifikator zusammengefasst und evaluiert. Das Ergebnis ist in Tabelle 8.7 zusammengefasst.
Wie erwartet, wird das beste Ergebnis dann erzielt, wenn alle Merkmale miteinander kombiniert werden.
In Tabelle 8.8 sind die Ergebnisse des linearen Klassifikators und des neuronalen Netzes gegenübergestellt.
Das neuronale Netz erzielt nur marginal bessere Kennzahlen als der lineare Klassifikator.

Kontext

Die Konfidenz des benachbarten Wortes, also des unmittelbaren Vorgängers und Nachfolgers, hat einen Einfluss auf die Konfidenz des gerade betrachteten Wortes. Das hat zwei Hauptursachen. Zum ersten erzeugt ein falsches Nachbarwort eine falsche Sprachmodellwahrscheinlichkeit für das betrachtete Wort, da von einem falschen Kontext ausgegangen wird. Zum zweiten ist die Wahrscheinlichkeit groß, dass ein falsches Nachbarwort zeitlich auch falsch zugeordnet wird und so einen Teil des betrachteten Wortes 'überdeckt' bzw. das betrachtete Wort um nicht dazugehörige Bereiche verlängert. Aus diesem Grund macht es Sinn, in den Merkmalssatz beim Training eines Vertrauensmessers auch die Merkmale der benachbarten Wörter aufzunehmen. Die Ergebnisse, berechnet mit dem neuronalen Netz als Klassifikator, zeigt Tabelle 8.9.

Es ist also möglich, durch die Verwendung von Kontext die Leistung des Vertrauensmessers gegenüber einem System ohne Kontext zu verbessern.

Merkmal	Kontingenz-koeffizient	Korrelations-koeffizient r
gamma	0.638	0.502
A-Stabil	0.628	0.482
$N(t_a - 1)$	0.599	-0.402
$N(t_a + 1)$	0.588	-0.389
A-Stabil-noAdapt	0.574	0.432
$\frac{d}{d\mu}$gamma	0.571	-0.367
$E(\log N)$	0.568	-0.378
$\log E(N)$	0.564	-0.378
gamma (bei $\mu = 0$)	0.521	0.370
$N(t_e - 1)$	0.478	-0.336
LM-Ngram	0.392	0.279
$N(t_e + 1)$	0.369	-0.232
NScoreQ	0.331	-0.173
NScore	0.323	-0.161
LogAWE-End	0.304	-0.213
PronVar	0.217	-0.113
Score-per-Frame	0.211	-0.137
Duration	0.191	-0.096
SprechGeschWPBQ	0.191	0.026
LogAnzImTraining	0.183	0.010
LogNPhones	0.174	0.092
NDisfluent	0.158	-0.043
AnzFrames	0.125	0.047
LogAWEbeg-3	0.115	-0.068
MaxSNR	0.111	0.064
AEntropy	0.104	-0.030

Tabelle 8.6: Korrelations- und Kontingenzkoeffizienten (μ ist das Sprachmodellgewicht)

Merkmale	CA	Fehlerreduktion
keines	85.3%	-
N(t_e-1),N(t_a+1),log($E(N)$)	87.5%	14.9%
plus N(t_e+1), N(t_a-1)	87.9%	17.7%
plus gamma	88.4%	21.1%
plus A-Stabil	88.9%	24.5%
LM-Ngram, LogAWE-End, NScoreQ, NScore, PronVar, Score-per-Frame Duration, LogNPhones, LogAWEbeg-3, MaxSNR	87.3%	13.6%
alle	90.0%	29.9%

Tabelle 8.7: Konfidenzakkuratheit des linearen Klassifikators mit verschiedenen Eingabemerkmalen

Merkmale	CA linear	CA neuronales Netz
nur wortgraphbasierte	88.9%	88.9%
alle	90.0%	90.1%

Tabelle 8.8: Vergleich des linearen Klassifikators mit dem neuronalen Netz

Merkmale	Kontext	CA	NCREP
nur wortgraphbasierte	nein	88.9%	0.326
nur wortgraphbasierte	ja	89.1%	0.340
alle	nein	90.1%	0.398
alle	ja	90.6%	0.416

Tabelle 8.9: Ergebnisse mit und ohne Kontext

Kapitel 9

Lernen ohne Transkripte

9.1 Motivation

Zu den größten Kostenfaktoren - sowohl zeitlich als auch finanziell - beim Bau eines Spracherkennungssystems gehört die Bereitstellung der Trainingsstichprobe für das akustische Training. Die Trainingsstichprobe sollte möglichst groß sein, damit viele Parameter robust geschätzt werden können. Sie wird meist in Form von geschnittenen, nicht allzu langen Sprachaufnahmen zusammen mit einer Transkription des Gesagten ausgeliefert.

Geht man bei der Erstellung der Trainingsstichprobe von einer existierenden Sprachaufnahme aus der Zieldomäne aus, muss zu der Sprachaufnahme eine Transkription, d.h. die Verschriftung des Gesagten, erstellt werden. Das Erstellen einer solchen Transkription erfordert wiederholtes Anhören der Sprachprobe und ist daher aufwendig und teuer. Um zum Beispiel eine Sendung 'tagesschau' zu transliterieren, sind zwischen 4 und 6 Stunden Zeitaufwand, also etwa 20-fache Echtzeit, erforderlich. Die in dieser Zeit produzierte Transliteration umfasst dann neben den eigentlichen Wörtern auch grobe Versprecher, Atemgeräusche und laute andere Geräusche. Wünscht man eine genauere Transliteration, wächst der Zeitaufwand noch einmal deutlich an. D. Oppermann gibt beispielsweise an [5], dass für die Transliteration der spontansprachlichen Sprachaufnahmen aus dem VERBMOBIL-Projekt insgesamt 112-fache Echtzeit erforderlich war. Die in dieser Zeit erstellten Transkripte umfassen allerdings eine sehr genaue Erfassung der realisierten Aussprachen aller Worte und eine Feintranskription aller Hintergrund- und Störgeräusche.

Heutige Spracherkennersysteme sind häufig auf dutzenden bis mehreren hundert Stunden Sprachmaterial trainiert. Der Aufwand zur Bereitstellung des Trainingsmaterials ist dann entsprechend groß.

Prinzipiell sind zwei Auswege aus diesem Dilemma denkbar. Der eine ist das Sammeln von vorgelesenen Textdaten. In diesem Szenario ist bereits a-priori bekannt, was der Inhalt des Sprachsignals ist. Die Notwendigkeit einer Transkription entfällt;

die Aufnahmen müssen allerdings noch manuell validiert werden, um technische Abbrüche und grobe Versprecher zu detektieren.

Der andere Weg wäre, das Transkribieren zu automatisieren. Wenn es möglich wäre, durch Training auf automatisch transkribierten Daten eine Verbesserung bestehender akustischer Modelle zu erzielen, reduzierte sich das Datensammelproblem auf das reine Aufnehmen von domänenspezifischen Daten.

Der erste Lösungsansatz - das Sammeln von vorgelesener Sprache - hat zwei bedeutende Nachteile:

- Der Grad der Spontaneität der aufgenommenen Sprache weicht von der Spontaneität in den Testdaten ab.
- Die Geräuschsituation (und generell der akustische Kanal) ist in den Trainingsdaten deutlich anders als in den Testdaten.

Der **Grad der Spontaneität** hat einen sehr starken Einfluss auf die Erkennungsleistung. Auf Diktierdaten, das heißt vorgelesenen Sätzen, sind Fehlerraten von 5-7% erreichbar [58]. Bei ungeplanter, aber sorgfältig artikulierter Sprache liegen die Fehlerraten bei 10-15% [25]. Ungeplante und nicht sorgfältig artikulierte Sprache, wie sie in der Mensch-zu-Mensch-Kommunikation verwendet wird, führt bereits zu Fehlerraten von 20-30% [31].

Die **Geräuschsituation** hat ebenfalls einen sehr starken Einfluß auf die Erkennungsleistung. So weist ein auf Diktierdaten trainierter Spracherkenner auf Daten mit Fahrgeräuschen im Hintergrund eine doppelt bis dreifach höhere Fehlerrate auf als ein auf Daten mit Fahrgeräuschen trainierter Erkenner [8].

Trotz dieser beiden Nachteile wurde in der Praxis bislang immer der erste Weg gewählt. Um dabei den negativen Einfluß der schlecht passenden akustischen Modelle zu reduzieren, wird häufig eine kleine Menge von Daten aus der entsprechenden Domäne transkribiert und die auf vorgelesenen Daten trainierten Modelle werden mit diesen Daten (z.B. mit MLLR) adaptiert. Dennoch erzielen die so trainierten System bei weitem nicht dieselben Erkennungsleistungen wie solche, die mit ausreichend Daten aus der richtigen Domäne trainiert werden.

In diesem Kapitel werden die vom Autor durchgeführten Experimente zur Verwendung von untranskribierten Daten zum Training eines Spracherkenners beschrieben. Untranskribierte Audio-Daten fallen im View4You-System täglich an und liegen damit in quasi unbegrenzter Menge vor. Die Verwendung von untranskribierten Daten zum Training hat zahlreiche Vorteile:

- Die Datensammlung entfällt, bzw. reduziert sich auf das Mitschneiden der Sendungen
- Die Daten entstammen derselben Domäne, weisen dieselben Hintergrundgeräusche und Aufnahmebedingungen auf wie die Daten, auf denen das System später eingesetzt wird

- Der Datenbestand kann sehr einfach kontinuierlich erweitert und aktualisiert werden.

Die Grundidee des Trainings mit untranskribierten Daten ist, dass ein initialer, schlechter Spracherkenner dazu verwendet wird, eine (schlechte) Transkription der Daten zu erzeugen. Auf dieser Transkription wird ein besserer Spracherkenner trainiert. Dieser kann seinerseits die Daten noch einmal transkribieren, auf den neuen Transkripten kann ein weiterer Spracherkenner trainiert werden usf.

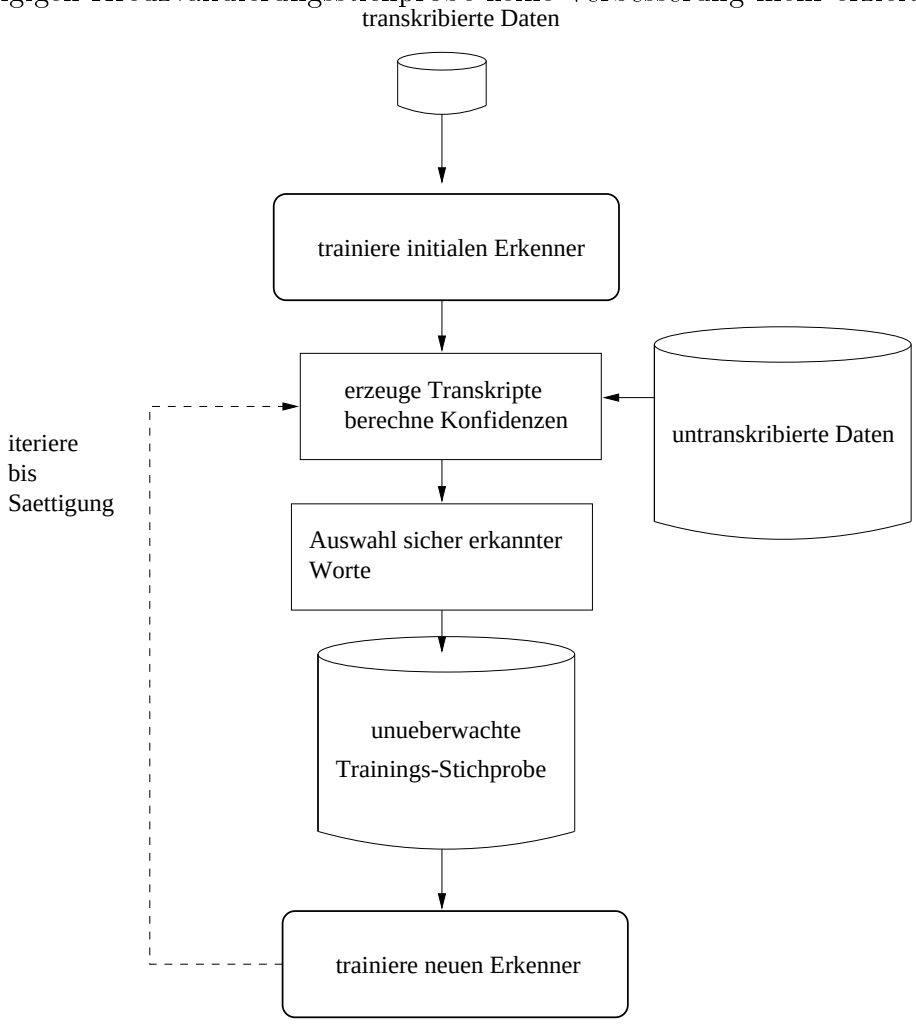
Der Versuch, einen Klassifikator auf unklassifizierten Stichproben lernen zu lassen, erscheint bei erster Betrachtung aussichtslos. Der Klassifikator kann nur diejenigen Beispiele korrekt lernen, die er von Anfang an richtig klassifizieren konnte, denn nur diese sind korrekt transkribiert. An den Stellen, an denen er fehlerhaft klassifiziert, wo also die selbsterstellte Transkription falsch ist, wird er nicht von einer übergeordneten Instanz korrigiert. Im Gegenteil, die fehlerhafte Transkription wird zum Lernen verwendet, was die Erkennungsleistung reduziert. Es ist also nicht anzunehmen, dass die Klassifikationsleistung ohne die Verwendung von zusätzlichem, nicht im Klassifikator bereits vorhandenem Wissen gesteigert werden kann. Das Trainieren auf untranskribierten Daten muss daher versagen, mehr noch, die Erkennungsleistung sollte durch ein solches Vorgehen sogar abfallen, da die Modelle auf falschen Daten trainiert werden.

Im Falle des autonom lernenden View4You-Systems ist die Ausgangslage jedoch nicht ganz so hoffnungslos, denn es steht externes Wissen in Form des Sprachmodells zur Verfügung. Das statistische Sprachmodell kann bei der Erkennung kleinere Fehler des akustischen Modells korrigieren. Die Transkription der Aufnahme ist damit besser als diejenige, die bei alleiniger Verwendung des akustischen Modells zu erzielen wäre. Das bedeutet, dass durch ein Neuschätzen der akustischen Modelle neue, korrekte, bisher (akustisch) falsch klassifizierte Daten zur Verfügung stehen. Diese steigern die Erkennungsleistung. Andererseits ist zu erwarten, dass die fehlerhaft erkannten Worte, die es natürlich trotz Verwendung des Sprachmodells weiterhin gibt, die Erkennungsleistung wiederum absenken.

9.2 Algorithmus zum unüberwachten Lernen

Der Algorithmus zum unüberwachten Lernen ist in Bild 9.2 zusammengefasst. Das Funktionsprinzip ist einfach: zunächst werden mit Hilfe eines initialen Spracherkenners automatisch Referenzen (Transkripte) angefertigt. Von diesen Transkripten werden Teile, die mit einer hohen Wahrscheinlichkeit fehlerhaft sind, eliminiert. Mit den restlichen Transkripten werden dieselben Trainingsalgorithmen angewendet wie beim 'normalen' Lernen auf transkribierten Daten. Das eigentliche akustische Training des Spracherkenners muss also nur minimal verändert werden, um auf lückenhaft transkribierten Daten zu funktionieren (wobei die Lücken durch das Eliminieren von wahrscheinlich inkorrekten Wörtern in den Transkripten entstehen).

Steht ein verbesserter Erkenner zur Verfügung, können mit diesem neue, verbesserte Transkripte angefertigt werden. Mit den nun verbesserten Trainingsdaten kann das Training wiederholt werden. Diese Vorgehensweise wird so lange iteriert, bis auf einer unabhängigen Kreuzvalidierungsstichprobe keine Verbesserung mehr erzielt wird



Das Initialisierungproblem

Um die initialen Hypothesen zu erzeugen, ist ein funktionsfähiger Spracherkenner erforderlich. Einer der wichtigsten Einsatzgebiete für den Algorithmus des unüberwachten Lernens ist der Wechsel der Domäne bzw. der Sprache, so dass in vielen Fällen kein solcher initialer Spracherkenner zur Verfügung stehen dürfte. Für dieses Problem gibt es zwei mögliche Lösungen:

- Die ersten Hypothesen werden durch einen Erkenner erzeugt, der auf einer anderen Domäne trainiert und mit Hilfe von wenigen Adaptionssätzen auf die Zieldomäne adaptiert wurde.

- Es wird mit Hilfe von wenig transkribierten Daten der Zieldomäne ein minimaler neuer Erkenner auf der Zieldomäne trainiert.

Welche der Vorgehensweisen die praktikablere ist, hängt vom Einzelfall ab. In beiden Fällen ist es erforderlich, eine kleine Menge von Daten der Zieldomäne von Hand zu transkribieren.

9.3 Vorarbeiten anderer Gruppen

BBN 1997

Auf dem 'Hub-5 Conversational Speech Recognition Workshop', der vom 4. bis zum 6.11. 1997 in Linthicum Heights, Maryland, USA von der amerikanischen Standardisierungsbehörde NIST organisiert wurde, stellte eine Forschungsgruppe der Firma BBN ihre Experimente zum unüberwachten Trainieren von Spracherkennern vor [27]. In dieser Arbeit sollte explizit das Problem, einen Spracherkenner auf eine neue Sprache anzupassen, bearbeitet werden. Dabei wurde angenommen, dass in der neuen Sprache

- ein Textkorpus, nicht notwendigerweise aus der Zieldomäne, vorhanden sein soll
- eine geringe Menge (wenige Stunden) transkribierte Sprache von vielen Sprechern zur Verfügung steht
- eine sehr große Menge von untranskribiertem Sprachmaterial für *dieselben* Sprecher zusätzlich zur Verfügung steht

Das Experiment wurde auf dem sog. CallHome/CallFriend Spanish Corpus durchgeführt. Dieser Korpus besteht aus Aufnahmen von Telefongesprächen, die von spanischen Muttersprachlern mit ihren Familien geführt worden waren. Die Gespräche wurden fast durchweg auf interkontinentalen Leitungen geführt. Die Domäne ist nicht eingeschränkt.

Von 356 verschiedenen Sprechern wurden jeweils 30 Sekunden transkribiert und als Trainingsset verwendet. Die Gesamtgröße des Trainingskorpus betrug somit ungefähr 3 Stunden Sprache oder 42000 laufende Worte. Zur Erzeugung des Sprachmodells wurden diesem Text noch 800.000 Worte laufender spanischer Zeitungstext hinzugefügt.
Als untranskribiertes Material standen weitere jeweils 4 Minuten derselben 356 Sprecher zur Verfügung (zusammen ungefähr 25 Stunden Sprache). Getestet wurde auf weiteren Äußerungen derselben Sprecher und auch auf ungesehenen, neuen Sprechern. Die Wortfehlerrate des Basissystems, trainiert auf 3 Stunden Sprache, lag bei 78,5%, das entspricht einer Wortakkuratheit von 21,5%. Zum Vergleich erzielte das beste

System der BBN-Gruppe, trainiert auf 58 Stunden Sprache, eine Wortfehlerrate von 59,3%.

In dem eigentlichen Experiment wurde zunächst das untranskribierte Material mit dem Basis-Erkenner dekodiert. Dann wurde ein Konfidenztagger eingesetzt, um in den Hypothesen die korrekt erkannten Worte zu finden. Der Konfidenztagger bestand aus der Kombination von mehreren Merkmalen wie der n-best-frequency (entsprechend A-stabil), Sprachmodellcounts, Sprachmodell-Scores und akustischen Scores, die mit generalisierten linearen Modellen zusammengefaßt wurden. Das Ergebnis des Konfidenztaggers ist in Tabelle 9.1 dargestellt.

% als 'korrekt' markiert	% davon tatsächlich korrekt
100	21.5
6	70
4	75
3	80
1	87.5

Tabelle 9.1: Performanz des Konfidenz-Taggers im BBN-Experiment

Bei einer erwünschten Präzision (Precision, PRC) des Konfidenztaggers von 80% ergab sich noch eine Ausschöpfung (Recall, RCL) von 3%. Da 3% nur 36 Minuten Sprache entsprachen, also bei der erwünschten Präzision die Menge der Trainingsdaten nur um etwa 10% vermehrt worden wäre, entschied man sich für die Simulation eines Konfidenztaggers: es wurden bei einer Präzision von 80% insgesamt 10% der gesamten Worte als korrekt markiert. So wurde die Datenmenge im Vergleich zum Basiserkenner knapp verdoppelt, wobei 20% der zusätzlichen Trainingsworte falsch waren. Mit dem neuen Trainingsset (bestehend aus den 42000 Wörtern der ursprünglichen und 40000 Wörtern der unüberwacht erzeugten Menge) wurden die akustischen Modelle und auch die Sprachmodell-Wahrscheinlichkeiten des Erkenners nachtrainiert. Dabei ergab sich eine insignifikante Verbesserung von 2.2% relativer Fehlerreduktion auf den bekannten Sprechern und von 0.4% auf unbekannten Sprechern.

Unter der Annahme eines perfekten Konfidenztaggers, der bei einer Ausschöpfung von 10% eine Präzision von 100% erreicht, wurde das Experiment wiederholt. Dabei ergab sich eine Verbesserung von 4.1% relativ auf den Trainingssprechern und von 0.8% auf den Testsprechern.

Die - doch recht pessimistischen - Schlussfolgerungen aus diesem Experiment lauteten

- eine Präzision von 80% erzielt etwa den halben Gewinn, der sich mit einer Präzision von 100% erzielen ließe

- um überhaupt eine signifikante Verbesserung des Systems zu erzielen, sind etwa 2 Größenordnungen mehr untranskribiertes Material als transkribiertes Material erforderlich

Zu bemerken ist natürlich die hohe Basisfehlerrate von über 70%. Bei einer niedrigeren Fehlerrate wird eine hohe Präzision des Konfidenztaggers schon bei weit größeren Ausschöpfungen erzielt. Allerdings erfordert das Training eines ohnehin schon guten Systems eine deutlich höhere Präzision als das Training eines recht schlechten Systems. Um zu klären, ob auf besseren Systemen das beschriebene Verfahren eventuell besser funktioniert, gibt BBN selbst die erforderliche Datenmenge mit 15000 Stunden an. Solche Datenmengen lassen sich mit den heute (1999) zur Verfügung stehenden Techniken jedoch noch nicht effektiv verarbeiten.

9.4 Methodik

In den in diesem Kapitel beschriebenen Experimenten werden komplette Spracherkennungssysteme miteinander verglichen.

Ein Spracherkenner ist ein komplexes System mit zahlreichen freien Parametern. Das Einstellen und Schätzen dieser Parameter ist eine schwierige und zeitaufwendige Aufgabe. Um zum Beispiel einen fertigen Erkenner zu erzeugen, mussten zehn hintereinanderfolgende Schritte durchgeführt werden. Bei jedem dieser Schritte sind zahlreiche Variationen denkbar und möglich. Kleine Abweichungen in einem einzigen dieser Schritte können zu signifikanten Änderungen in der Leistung des fertigen Systems führen.

Um einen aussagekräftigen Vergleich verschiedener Systeme zu ermöglichen, müssen also alle Schritte des Trainings der Systeme identisch gehandhabt werden. Es müssen auch dieselben Methoden (LDA, Adaption, VTLN, MLLR...) in allen Fällen auf dieselbe Art angewendet werden. Auch die verwendeten Wissensquellen (Sprachmodell, Wörterbuch, Phonemsatz, ...) müssen gleich sein.

Ein weiteres, grundlegendes Problem sei an dieser Stelle nicht verschwiegen. Durch das 'Spielen' an freien Parametern der eingesetzten Trainings- und Testmethoden ist es möglich, die Erkennungsleistung auf einem gegebenen Testset deutlich zu erhöhen. Ein Experimentator, der ein neues Verfahren evaluiert und ein Interesse an guten Ergebnissen des neuen Verfahrens hat, ist in der Gefahr, das System mit dem neuen Verfahren mehr zu 'tunen' als das Ausgangssystem. Der Vergleich der Systeme ist dann nicht mehr aussagekräftig.

Um die aufgezeigten potentiellen methodischen Schwierigkeiten und Mängel zu umgehen, wurde für alle in diesem Kapitel beschriebenen Experimente ein **vollautomatisiertes Trainingsskript** eingesetzt. Das bedeutet, dass das gesamte Training - beginnend beim Punkt Null - mitsamt der Auswertung auf der Kreuzvalidierungsstichprobe durch den Aufruf eines einzigen Trainingsskriptes realisiert wurde. Dieses

Trainingsskript ruft hintereinander zehn Unterskripten auf, die die verschiedenen Stufen des Trainings durchführen. Der gesamte Prozess ist vollautomatisch und läuft ohne die Intervention des Experimentators ab. Dessen Rolle beschränkt sich ausschließlich auf das Starten und das Interpretieren der Ergebnisse.

Im folgenden Abschnitt wird kurz die automatische Trainingsprozedur umrissen. Da die einzelnen Algorithmen und ihre Funktionsweise an anderer Stelle (in den Kapiteln 'Maschinelle Spracherkennung' ab Seite 26 und 'Der Spracherkenner des View4You-Systems' ab Seite 68) ausführlich erläutert sind, wird auf eine detaillierte Beschreibung an dieser Stelle verzichtet.

Die Trainingsprozedur

Die zehn Schritte der automatischen Trainingsprozedur sind:

1. Berechnung der LDA-Matrix mit Sub-Monophonen als Klassen
2. Extrahieren von Beispielvektoren und kmeans-Ballen derselben zur Erzeugung der initialen Mittelwert- und Kovarianzvektoren der Submonophonmodelle
3. Drei Iterationen Viterbi-Training der Submonophonmodelle
4. Bestimmung aller in der Trainingsstichprobe vorkommenden unterschiedlichen phonetischen Kontexte und Erzeugung von (leeren) a-priori-Verteilungen für die bestehenden Submonophonem-Codebücher
5. Schätzung der a-priori-Verteilungen durch eine Iteration Viterbi-Training
6. Berechnung des divisiven Allophonclusterbaumes bis zu einer vorgegebenen Tiefe
7. Berechnung der LDA-Matrix mit kontextabhängigen Allophonmodellen als Klassen
8. Extrahieren von Beispielvektoren und kmeans-Ballen derselben zur Erzeugung der initialen Mittelwert- und Kovarianzvektoren der kontextabhängigen Allophonmodelle
9. Fünf Iterationen Viterbi-Training der kontextabhängigen Allophonmodelle
10. Test auf der Kreuzvalidierungsstichprobe

Natürlich legt diese Abfolge noch nicht die genauen Parameter des fertigen Systems fest. Beim Extrahieren von Beispielvektoren ist die Anzahl der extrahierten Beispielvektoren pro Codebuch zum Beispiel eine frei einstellbare Größe; beim Testen ist das Sprachmodell einstellbar, und so weiter. Alle diese Parameter werden aus einer zentralen Steuerdatei gelesen. Durch den festen Ablauf des Trainings, die Steuerdatei und die vorberechneten Zeitzuordnungen (Labelfiles) wird ein System vollständig definiert.

Die Anzahl der Parameter im fertigen System hängt von der zur Verfügung stehenden Menge an Daten ab. Je mehr Trainingsdaten zur Verfügung stehen, desto mehr Parameter können geschätzt werden. Im Kapitel 'Optimierung der Parameterraumgröße' ab Seite 79 wurde eine Formel zur Berechnung der optimalen Zahl von Parametern bei einer gegebenen Größe des Trainingskorpus hergeleitet: etwa 15-20 Datenvektoren pro Normalverteilung mit einer oberen Grenze von 5000 Mixturen. Der Allophonclusterbaum (Schritt 6) wurde beim Training bis zu derjenigen Tiefe berechnet, bei der die optimale Größe des Parameterraumes erreicht war. Zwei Systeme A und B, die auf unterschiedlichen Datenmengen berechnet wurden, weisen daher unterschiedlich große Parameterräume auf.

9.5 Experimente

Daten

Für die in diesem Kapitel beschriebenen Experimente standen insgesamt 208 Sendungen 'tagesschau', aus dem Zeitraum vom 25.11.1996 bis zum 2.9.1998 zur Verfügung. Für insgesamt 64 dieser Sendungen existierten Transkriptionen. 4 transkribierte Sendungen (30.3.97, 13.4.97, 28.5.97, 30.6.97) wurden als Kreuzvalidierungs- bzw. Teststichprobe verwendet; die verbleibenden 60 transkribierten und 144 untranskribierten Sendungen wurden zum Training der unterschiedlichen Erkenner eingesetzt.

Beim unüberwachten Training nach dem Algorithmus auf Seite 147 ist es erforderlich, eine kleine Menge an Transkriptionen zum Training des initialen Erkenners zur Verfügung zu haben (vgl. Seite 147).

Für die beschriebenen Experimente wurden die beiden frühesten Aufnahmen (vom 25.11.1996 und 26.11.1996, mit den Ansagesprechern Jan Hofer und Dagmar Berghoff) als die angesprochene kleine Menge an transkribierten Daten verwendet. Zwei Sendungen entsprechen ungefähr 30 Minuten Daten.

Selbstverständlich wird der Algorithmus zum unüberwachten Lernen in jeder realistischen Anwendungssituation sämtliche verfügbaren transkribierten - also besonders wertvollen - Daten nutzen und diese zu seiner Datenbasis hinzufügen. Dies wurde daher auch in allen Experimenten, auch wenn es nicht gesondert aufgeführt ist, so durchgeführt. Den beiden Sendungen vom 25. und 26.11.96 kam daher eine besondere Bedeutung zu: sie sind in den Trainingsdatenmengen von *allen* im Rahmen dieser Arbeit untersuchten Systemen vorhanden.

Tabelle 9.11 fasst alle Daten zusammen.

Initiale Spracherkenner

Mit der beschriebenen automatischen Technik wurde auf 2 Sendungen 'tagesschau' ein initialer Erkenner trainiert. 2 Sendungen entsprechen lediglich 30 Minuten Spra-

che. Um die kleine, handtranskribierte Datenmenge optimal zu nutzen, wurde beim Training des initialen Spracherkenners zweimal iteriert. Zunächst wurde eine Zeitzuordnung mit Hilfe eines anderen deutschen Spracherkenners, der auf spontanen Dialogen trainiert worden war, durchgeführt. Mit Hilfe der Zeitzuordnung - also von Labelfiles - wurde der existierende Spracherkenner auf die Trainingsstichprobe adaptiert (MLLR). Danach konnte eine verbesserte Zeitzuordnung berechnet werden. Auf dieser besseren Zeitzuordnung wurde dann mit Hilfe des automatischen Trainings-Skriptes der initiale Erkenner mit der Bezeichnung 'I41' trainiert.

Um eine obere Grenze für die Leistung eines unüberwachten Trainingsalgorithmus zu erhalten, muss ein Erkenner auf denselben, aber handtranskribierten Daten trainiert werden. Ein solches Training wurde auf den 60 zur Verfügung stehenden transkribierten Aufnahmen durchgeführt. Die Fehlerraten beider Systeme sind in Tabelle 9.2 zusammengefaßt.

Trainingsmenge	WER Ansager	WER Korrespondent	WER total
2 Sendungen	22.0%	38,8%	32,1%
60 Sendungen	11,9%	24,6%	19,5%

Tabelle 9.2: Performanz des initialen Spracherkenners (2 Sendungen) und Performanz bei Verwendung aller transkribierter Daten (60 Sendungen)

Konfidenzmaß

Ein sehr wichtiger Schritt im Algorithmus zum unüberwachten Lernen ist die Identifizierung von falschen Referenzen durch die Anwendung eines Konfidenzmaßes. Für alle durchgeführten Experimente wurde das im Rahmen dieser Arbeit entwickelte Konfidenzmaß *gamma* verwendet (siehe Seite 127). *gamma* berechnet die a-posteriori-Wahrscheinlichkeit, mit der ein Wort aus der Hypothese korrekt erkannt wurde, als einen Wert p mit $0 < p \leq 1$. Eine sichere Identifikationsmöglichkeit von korrekt erkannten Wörtern liefert *gamma* - selbst bei fehlerfreier Funktion - also nur für den Sonderfall $p = 1$. Es ist aber nicht praktikabel, nur solche Worte zum Training zuzulassen, bei denen die a-posteriori-Wahrscheinlichkeit des Konfidenzmaßes 1 ist, da prozentual nur sehr wenige Worte mit $p = 1$ markiert werden. Damit wäre die Größe des verwendbaren Trainingskorpus zu klein, bzw. die Menge an untranskribierten Daten, die benötigt werden, zu groß.

Ein weiterer Grund wurde bereits kurz angesprochen.

Worte mit $p = 1$ weisen eine so viel bessere Bewertung durch die akustischen Modelle des Erkenners als alle konkurrierenden Worte auf, dass die konkurrierenden Worte im Verlauf der Suche aus dem Suchraum entfernt (geprunt) wurden. Eine

gute Bewertung durch die akustischen Modelle ist gleichbedeutend mit einer hohen Übereinstimmung zwischen Modell und Testdaten. Wählt man nur solche Worte, für die die akustischen Modelle bereits sehr gut 'passen', ist die zusätzliche Varianz durch die neuen Trainingsdaten klein. In anderen Worten: der Erkenner kann nichts Neues lernen, wenn er im Training nur Dinge sieht, die er bereits gut kennt.

Auf der Gegenseite darf natürlich auch nicht jedes Wort der Hypothese zum Training herangezogen werden. Ein fehlerhaftes Wort in der Referenz bewirkt zwangsläufig, dass die entsprechenden Modelle falsch adaptiert werden. Das wiederum führt zu einem Rückgang der Erkennungsleistung.

Aus dem Gesagten geht hervor, dass zum Training alle Worte herangezogen werden sollten, deren a-posteriori- Wahrscheinlichkeit größer ist als ein Schwellwert t.

kleiner Schwellwert t		großer Schwellwert t	
Vorteil	Nachteil	Vorteil	Nachteil
große Menge			kleine Menge
	Transkriptionsfehler	Korrektheit	
viel Varianz			wenig Varianz

Tabelle 9.3: Auswirkung des Schwellwerts t

Die vom Konfidenzmaß berechneten Werte sind natürlich teilweise fehlerhaft. Der genaue Zusammenhang zwischen Datenqualität und Datenmenge auf der einen Seite und dem Schwellwert t auf der anderen Seite kann nur experimentell bestimmt werden.

Hierzu wurden zunächst die 4 Sendungen der Teststichprobe mit einem Spracherkenner (System I33_0_5) dekodiert und mit *gamma* konfidenzannotiert - das heißt, für jedes Wort der Hypothesen wird eine Konfidenz berechnet und dem Wort zugeordnet. Da für die Teststichprobe die Referenz bekannt ist, läßt sich die Performanz des Konfidenztaggers bestimmen.

Von den vier möglichen Kennzahlen (Precision und Recall jeweils für die richtig erkannten und die fehlerhaft erkannten Worte) sind für das unüberwachte Training vor allem die Recalls wichtig. Ein hohes Recall der richtig erkannten Worte bedeutet eine gute Ausnutzung der Trainingsdaten; ein hohes Recall bei den Erkennungsfehlern - die vom Training dann ausgeschlossen werden - bedeutet eine weniger fehlerbehaftete Trainingsstichprobe. Ideal wäre ein Recall von 1,0 sowohl bei den richtig erkannten Wörtern als auch bei den Erkennungsfehlern. Die Schaubilder 9.1 und 9.2 zeigen das Ergebnis der Evaluation. Die Basisfehlerrate - wenn jedes Wort mit 'korrekt' getaggt wird - beträgt 30% (Startwert der Kurve 'Precision korrekt erkannter Worte' in Bild 9.2). Bei kleiner Schranke ($t = 0.2$) werden nahezu alle Worte mit 'korrekt' getaggt. Die Ausnutzung der richtig erkannten Worte ist fast vollständig ($> 98\%$); allerdings

werden nur gut 22% der Erkennungsfehler als solche markiert. Bei einer mittleren Schranke ($t = 0.5$) werden immer noch über 90% der richtig erkannten Worte verwendet, und bereits mehr als die Hälfte der Erkennungsfehler korrekt identifiziert. Wählt man eine hohe Schranke ($t = 0.9$), so werden 83% der Erkennungsfehler identifiziert, aber nur noch 62% der richtig erkannten Worte verwendet.

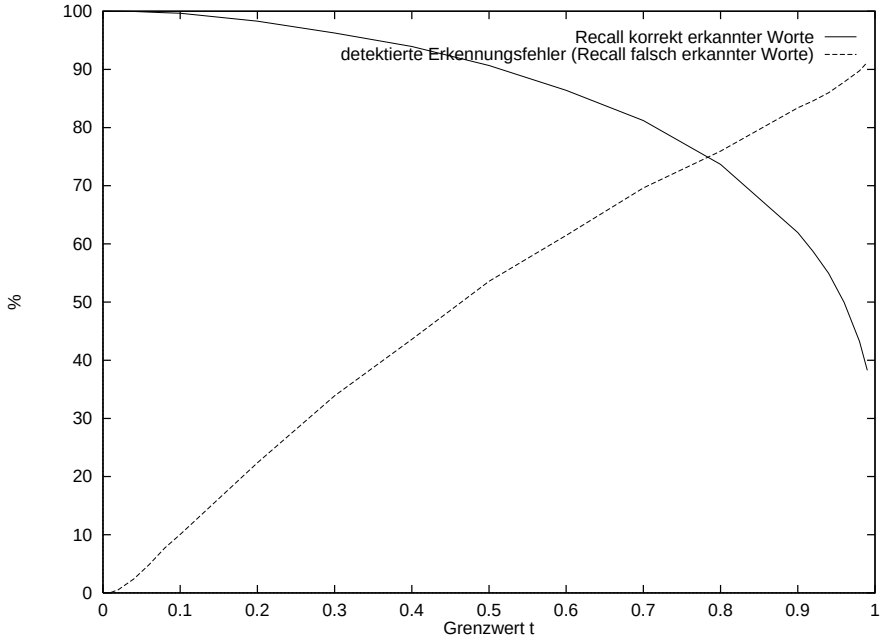


Abbildung 9.1: Recall korrekt / fehlerhaft erkannte Worte

Bei einer Schranke von $t = 0.1$ liegt die Präzision der als richtig markierten Worte bei 71%, bei $t = 0.5$ bei 81% und bei $t = 0.9$ bei 89%. Die im Experiment von BBN (siehe Seite 148 ff) geforderten 80% Präzision werden also ab einer Schranke von 0.5 erreicht.

Intermediäres System

Die Performanz des initialen Erkennersystems I41 ist relativ schlecht. Aus diesem Grund wurde der auf Seite 147 dargestellte Algorithmus einmal auf einer intermediären Trainingsstichprobe von 24 Tagesschau-Sendungen (entsprechend etwa 6 Stunden Sprache) iteriert. Die Konfidenzschranke wurde willkürlich auf $t = 0.5$ festgelegt. Zum Training des Systems wurden zunächst Hypothesen der 24 Sendungen mit Hilfe des initialen (I41) Erkenners generiert und konfidenzannotiert. Zugleich wurden Zeitzuordnungen zwischen den Hypothesen und den Aufnahmen ('Labelfiles') berechnet und abgespeichert. Diese Zeitzuordnungen wurden während des gesamten Trainings verwendet. Alle Worte mit einer Konfidenz größer als 0.5 wurden zur Schätzung der neuen akustischen Modelle herangezogen.

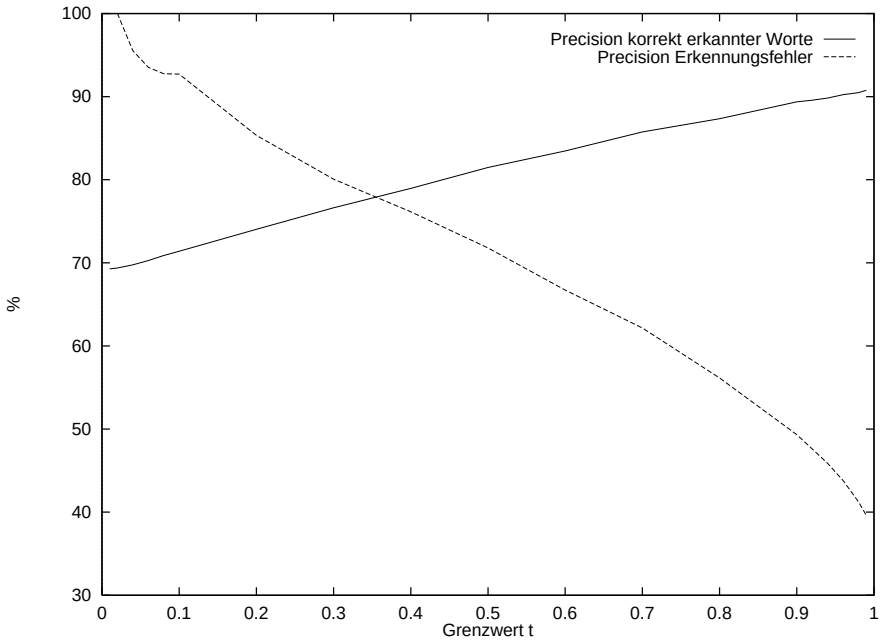


Abbildung 9.2: Precision korrekt / fehlerhaft erkannte Worte

Das System I41 weist aufgrund der geringen Datenmenge nur 314 unterschiedliche Subphonemmodelle auf. Durch die erhöhte Menge an Trainingsmaterial wurde die Anzahl der unterschiedlichen Subphonemmodelle für das neue System durch Anwendung der auf Seite 84 hergeleiteten Beziehung zu 2396 festgelegt.

Das eigentliche Training erfolgte mit Hilfe der automatischen, auf Seite 151 dargestellten Prozedur. Nach 5 Iterationen Viterbi-Training der neuen Modelle wurde die Performanz des fertigen Systems (genannt I33_0_5) mit der des initialen Systems (I41) auf derselben Teststichprobe verglichen. Das Ergebnis ist in Tabelle 9.4 zusammengefaßt. Die Fehlerrate sinkt um fast 8 Prozent absolut (25% relativ).

System	WER Ansager	WER Korrespondent	WER total
I41	22.0%	38.8%	32.1%
I33_0_5	14.0%	31.0%	24.2%

Tabelle 9.4: Fehlerrate des intermediären Systems I33_0_5

Es ist nun die Frage, inwieweit die schlechte Qualität der initialen Hypothesen und Zeitzuordnungen den Ausgang des Trainingsexperiments beeinflußt. Um diese Frage zu klären, wurde ein weiteres Experiment durchgeführt.

Mit Hilfe des intermediären Systems I33_0_5 wurden die 24 Sendungen der intermediären Trainingsstichprobe transkribiert und konfidenzannotiert, und daraufhin neue, verbesserte Zeitzuordnungen zwischen den neuen Hypothesen und den Aufnahmen berechnet. Die Konfidenzschranke wurde wie beim System I33_0_5 auf $t = 0.5$

festgelegt. Auf den neuen, verbesserten Hypothesen und Zeitzuordnungen wurden mit Hilfe des automatischen Trainingsverfahrens neue akustische Modelle (System 'I53') berechnet. Beide Systeme (I53 und I33_0_5) wurden also auf derselben Trainingsstichprobe mit exakt denselben Algorithmen und Wissensquellen trainiert. Das Ergebnis ist in Tabelle 9.5 angegeben. Die Performanz des - auf besseren Hypothesen und Zeitzuordnungen berechneten - Systems ist nur insignifikant besser als die des intermediären Systems I33_0_5. Auf eine weitere Iterierung kann also verzichtet werden.

In einem weiteren Versuch wurden insgesamt 120 Sendungen 'tagesschau' (entsprechend 32 Stunden Sprachdaten) durch das intermediäre System I33_0_5 transkribiert und konfidenzannotiert sowie Zeitzuordnungen berechnet. Die Anwendung der Regel von Seite 84 ergibt bei dieser Datenmenge eine nicht mehr handhabbare Größe des akustischen Parameterraumes. Daher wurde die Anzahl der unterschiedlichen Subphonemmodelle auf die Obergrenze von 5000 festgelegt. Auf der vergrößerten Trainingsstichprobe wurde mit Hilfe des automatischen Trainingsverfahrens ein weiteres System (I56) trainiert. Die Testergebnisse für dieses System finden sich ebenfalls in Tabelle 9.5. Man erkennt eine deutliche Fehlerreduktion im Vergleich sowohl zum initialen, als auch zum intermediären System. Die Fehlerrate fällt insgesamt um 10.7% (absolut).

System	WER Ansager	WER Korrespondent	WER total
I41	22.0%	38.8%	32.1%
I33_0_5	14.0%	31.0%	24.2%
I53	14.9%	29.8%	23.8%
I56	12.6%	27.3%	21.5%

Tabelle 9.5: Fehlerrate des neutrainierten intermediären Systems I53 und des auf 120 Sendungen trainierten I56

Arbeitspunkt des Konfidenzmaßes

Der gewählte Arbeitspunkt $t = 0.5$ für das Konfidenzmaß ist möglicherweise nicht optimal im Hinblick auf die erzielbare Erkennungsleistung. Aus diesem Grund wurden Experimente zur Bestimmung des besten Schwellwertes t durchgeführt.

Für die Arbeitspunkte $t = 0$ (kein Konfidenzmaß), $t = 0.2$ (maximale Ausschöpfung bei gleichzeitiger Eliminierung der schlimmsten Fehler), $t = 0.5$ (siehe oben) und $t = 0.9$ (hochqualitative Daten) wurden insgesamt 4 Systeme auf derselben Trainingsstichprobe trainiert. Sämtliche im Training verwendeten Wissensquellen waren identisch. Die Zeitzuordnung für alle vier Systeme stammte vom intermediären System I33_0_5. Die Zahl der akustischen Modelle wurde in allen Fällen auf 5000 begrenzt. Das eigentliche Training erfolgte mit dem bereits beschriebenen automatischen Trainingsverfahren (siehe Seite 151). Die resultierenden Systeme wurden auf

derselben Teststichprobe getestet. Dabei ergaben sich bei den Ansagesprechern nur insignifikante Unterschiede in der Performanz. Die Ergebnisse auf den Korrespondentenberichten sind in Tabelle 9.6 dargestellt. Man erkennt, dass die niedrigste Fehlerrate für eine Schranke $t = 0.5$ erzielt wird.

Schranke t	WER Korrespondent
0	29.2%
0.2	28.7%
0.5	27.3%
0.9	28.3%

Tabelle 9.6: Performanz über Konfidenzschwelle t (120 Sendungen Training)

Lineare Konfidenzen

Für die bisher beschriebenen Experimente wurden stets alle Worte aus dem Trainingsset eliminiert, die eine Konfidenz von weniger als t aufwiesen. Das ist natürlich nicht die einzige Methode, die Informationen des Konfidenzmaßes zu nutzen. Außer der Wahl anderer Arbeitspunkte können die Worte auch nach ihrer Konfidenz gewichtet werden [30]. Ein Wort mit einer Konfidenz von beispielsweise 0.4 würde dann nur halb so großen Einfluß auf die Schätzung der Parameter haben wie ein Wort mit einer Konfidenz von 0.8. Eine solche Gewichtung ist bei der Berechnung der relevanten Parameter der zur Modellierung verwendeten Mixturen von Normalverteilungen leicht realisierbar. Hierzu wird sämtlichen Datenvektoren, die zeitlich (durch den Viterbi-Algorithmus) dem Wort i mit der Wort-Konfidenz c_i zugeordnet werden, die Konfidenz c_i zugeordnet. Berechnet man nun die Parameter einer beliebigen Normalverteilung eines Allophones aus den zugehörigen Datenvektoren, so gilt für den Mittelwertvektor $\vec{\mu}$

$$\vec{\mu} = \frac{1}{\sum_k c_k} \sum_k c_k \vec{x_k} \tag{9.1}$$

und analog für die Varianz. Für die Berechnung der Scattermatrizen für die lineare Diskriminanzanalyse (Gln. 4.2 und 4.2 auf Seite 34) kann ebenso verfahren werden.

Um dieses Verfahren zu evaluieren, wurde ein Erkenner auf denselben 120 Sendungen trainiert, die auch für die Systeme aus Tabelle 9.6 verwendet wurden. Das Ergebnis zeigt Tabelle 9.7.

Verfahren	WER Korrespondent	WER total
digital, t=0.5	27.3%	21.5%
linear	27.9%	21.7%

Tabelle 9.7: Performanz bei linearen Konfidenzen

Das Endergebnis bei linearer Verwendung der Konfidenzen ist nur wenig schlechter als das Ergebnis mit digitalen Konfidenzen und einer Grenze von $t = 0.5$. Lineare Konfidenzen stellen damit eine gangbare Alternative zum Eliminieren von niedrigkonfidenten Worten dar.

Obere Grenze des Verfahrens

Es ist interessant, die obere Grenze der potentiellen Leistungsfähigkeit des verwendeten Trainingsalgorithmus' zu kennen. Andersherum gefragt: wie gut könnte das System werden, wenn es sich selber perfekt beurteilen könnte, wenn also das Konfidenzmaß fehlerfrei wäre?

Um diese Frage zu klären, muss ein fehlerfreies Konfidenzmaß simuliert werden. Dies ist nur möglich, wenn das tatsächlich Gesagte bekannt ist. Die für das unüberwachte Trainingsverfahren verwendete Trainingsstichprobe musste also aus dem transkribierten Trainingsset entstammen. Natürlich durfte die Transkription dabei nur zur Simulation des fehlerfreien Konfidenzmaßes, keinesfalls jedoch als Ersatz der Hypothese herangezogen werden.

Zunächst wurden mit Hilfe des intermediären Erkenners I33_0_5 Hypothesen derjenigen 58 Sendungen 'tagesschau' angefertigt, für die Transkriptionen vorlagen und die nicht zum Training des intermediären Erkenners selbst herangezogen worden waren (also diejenigen der zweiten Reihe in Tabelle 9.11). Diese 58 Sendungen waren für den Erkenner ungesehene Testdaten. Unter Ausnutzung der vorhandenen Transkription wurde dann jedes falsch erkannte Wort mit '0', jedes richtig erkannte Wort mit '1' annotiert. Diese Annotationen entsprachen damit denen eines perfekten, fehlerfreien Konfidenzmaßes. Alle Worte mit Konfidenz '1' wurden danach zum Training herangezogen. Das so erzeugte System stellt eine obere Schranke der möglichen Leistung des Trainingsverfahrens dar.

Um diese Zahl vergleichen zu können, wurden die Hypothesen des intermediären Systems zusätzlich noch - ohne Ausnutzung der Transkriptionen - mit dem *gamma*-Konfidenzmaß konfidenzannotiert und ein weiteres System mit einer Schwelle von $t = 0.5$ trainiert.

Die Ergebnisse sind in Tabelle 9.8 zusammengefaßt. Die unterste Zeile beschreibt das I39-System, das auf allen 60 Sendungen der transkribierten Trainingsstichprobe trainiert wurde. Man erkennt, dass die Fehlerrate durch das unüberwachte Training drastisch (um 10% absolut) sinkt. Selbst ein perfektes Konfidenzmaß bewirkt dabei nur noch eine unwesentliche Verbesserung um 1.1% absolut. Der Abstand zwischen dem perfekten Konfidenzmaß und den Transkriptionen ist mit fast 3% noch recht groß. Der Grund für diesen großen Abstand ist die mangelnde korrigierende Komponente des automatischen Verfahrens: ein falsch erkanntes Wort wird nicht dem Klassifikator nicht in korrigierter Form vorgelegt, sondern lediglich aus der Trainingsstichprobe eliminiert - entsprechend einem menschlichen Lehrer, der beim Vokabelabfragen nur 'falsch' sagt, ohne die korrekte Bedeutung des abgefragten Wortes zu

nennen.

Training	WER Korrespondent
initialer Erkenner (2 TGS)	38.8%
60 TGS unüberwacht, reales Konfidenzmaß	28.5%
60 TGS unüberwacht, perfektes Konfidenzmaß	27.4%
60 TGS überwacht (I39)	24.5%

Tabelle 9.8: Auswirkung eines perfekten Konfidenzmaßes

Konvergenz des unüberwachten Trainingsverfahrens

Bild 9.3 zeigt Erkennungsergebnisse von insgesamt elf Spracherkennern, die unter Verwendung des automatischen Trainingsskriptes auf verschiedenen Datenmengen trainiert wurden. Die untere Kurve markiert die Resultate des 'normalen' Trainings auf transkribierten Daten, die obere Kurve die Ergebnisse des hier vorgestellten Trainings auf untranskribierten Daten, jeweils mit $t = 0.5$. Die Anzahl der Parameter ist in jedem System unterschiedlich und wurde so eingestellt, dass 15 Datenframes pro Mittelwertvektor, maximal aber 150000 Mittelwertvektoren insgesamt (entsprechend 5000 kontextabhängigen Subpolyphonmodellen) geschätzt wurden.

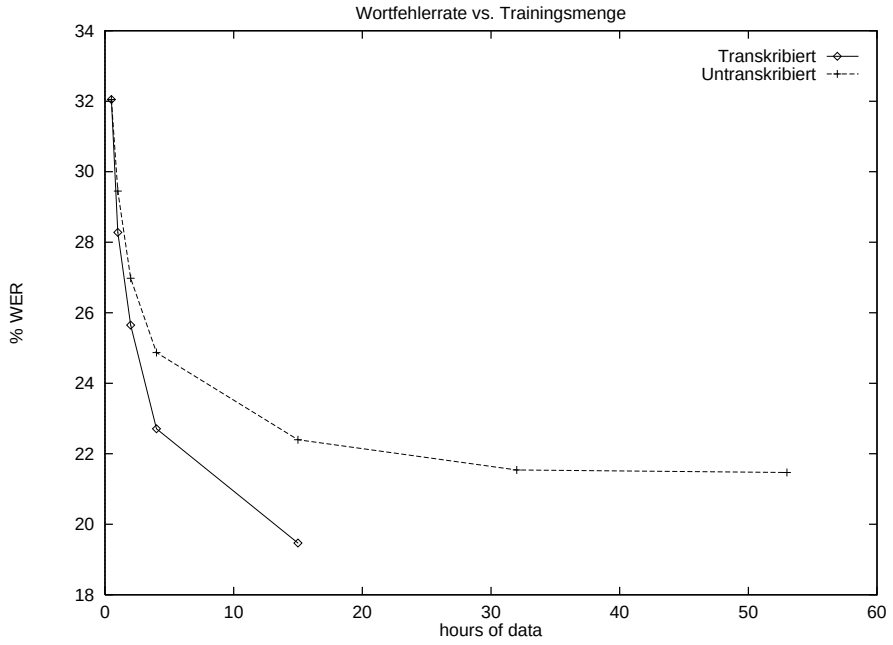


Abbildung 9.3: Wortfehlerrate über Menge des Trainingsmaterials

Das unüberwachte Trainingsverfahren konvergiert rasch bis zu einem Wert von ca. 21,4% Gesamtfehlerrate, die durch Hinzunahme von weiteren Trainingsdaten nicht

mehr unterschritten werden können. Die Sättigung wird bereits bei etwa 30 Stunden Trainingsmaterial erreicht.

Ob eine ähnliche Sättigung auch beim konventionellen Training auf transkribierten Daten eintritt, läßt sich mit den zur Verfügung stehenden Datenmengen nicht beurteilen. Bei den amerikanischen DARPA-Evaluationen konnten Woodland et al. die Fehlerrate um 2.1% absolut (entsprechend 7% relativ) senken [21], wenn die Trainingsdatenmenge von 35 auf 70 Stunden verdoppelt wurde. Chen et al. beobachteten auf denselben Daten nur noch eine insignifikante Änderung ($< 0.5\%$) [16]. In jedem Fall ist mit einer Abflachung der Lernkurve zu rechnen.

ROVER

Obwohl die Performanz des nach dem vorgeschlagenen Algorithmus unüberwacht trainierten Systems durch einfache Vergrößerung der Trainingsmenge nicht mehr verbessert werden konnte, können weitere Trainingsdaten dennoch nutzbringend eingesetzt werden. Dies geschieht auf dem Wege der Hypothesenkombination.

Die Grundidee hierzu wurde von Fiscus 1997 für die Spracherkennung vorgeschlagen [31]. Voraussetzung sind mehrere unabhängig voneinander entstandene Hypothesen für dieselbe Sprachäußerung, von denen eine als Referenzhypothese herangezogen wird. Für jedes Wort der Referenzhypothese kann nun nachgesehen werden, ob die Mehrheit der anderen Hypothesen an dieser Stelle dasselbe Wort beinhaltet. Ist das der Fall, gilt das Wort als akzeptiert und wird in die endgültige Hypothese übernommen. Wenn die anderen Hypothesen mehrheitlich ein anderes Wort als die Referenzhypothese an der betrachteten Position enthalten, wird dieses in die endgültige Hypothese übernommen. Sind alle Hypothesen an der betrachteten Position unterschiedlich, wird das Wort aus der Referenzhypothese gewählt.

Im Prinzip findet also - Wort für Wort - eine Mehrheitsentscheidung unter den unterschiedlichen Hypothesen statt. Wenn die Hypothesen unterschiedlich sind und die Worterkennungsrate innerhalb der einzelnen Hypothesen größer ist als purer Zufall, führt das Verfahren der Mehrheitsentscheidung im Mittel dazu, dass die Fehlerrate der kombinierten Hypothese niedriger ist als die jeder einzelnen daran beteiligten Hypothese. Dieses auf den ersten Blick etwas paradox anmutende Verhalten läßt sich leicht begründen. Angenommen, ein Ensemble von N unabhängigen Klassifikatoren (Spracherkennern) klassifiziert eine Eingabe (Sprachprobe). Der Einfachheit halber bestehe diese aus nur einem einzigen gesprochenen Wort. Jeder Klassifikator erzeuge mit einer Wahrscheinlichkeit p_i die richtige Ausgabe und mit einer Wahrscheinlichkeit $\frac{1-p_i}{V-1}$ eine beliebige falsche Ausgabe aus seinem Ausgabealphabet des Umfangs V, wobei die falschen Ausgaben statistisch gleichverteilt seien. Dann ist die Wahrscheinlichkeit, dass q oder mehr ($1 < q < N$) Klassifikatoren die richtige Ausgabe machen, gleich

$$p(\text{richtig}) = \sum_{r=q}^{r=N} \binom{N}{r} p_i^r (1-p_i)^{N-r} \tag{9.2}$$

Für falsche Ausgaben gilt

$$p(\text{falsch}) = (V-1) \sum_{r=q}^{r=N} \binom{N}{r} (\frac{1-p_i}{V-1})^r p_i^{N-r} \tag{9.3}$$

(9.3) ist kleiner als (9.2), solange $p_i > \frac{1}{V}$ gilt. Damit ist bewiesen, dass ein Ensemble von Klassifikatoren, die jeweils besser sind als Zufall und deren Fehler gleichverteilt sind, immer besser ist als jeder Klassifikator für sich. Für $N \to \infty$ fällt die kumulierte Fehlerrate auf Null.

Die Einschränkung, dass die Fehler gleichverteilt sind, ist in den meisten praktischen Fällen nicht gegeben. Ein Wort wie 'Baum' wird z.B. häufiger mit 'Raum' als mit 'Computer' verwechselt. Läßt man die Einschränkung fallen, hängt die 'Nützlichkeit' eines Klassifikators sowohl von seiner Klassifikationsrate p_i als auch von der Verteilung der von ihm gemachten Fehler ab. Für $p_i > 0.5$ ist (9.2) immer größer als (9.3), unabhängig von der Verteilung der Falschklassifikationen.

In der Herleitung wurde stillschweigend vorausgesetzt, dass nicht nur jeder Klassifikator im Mittel über alle Samples eine Klassifikationsrate von p_i erzielen muss, sondern dass auch im Mittel aller Klassifikatoren ein Sample j eine mittlere Klassifikationsrate p_i erzielt. Gilt dies nicht, gibt es also 'schwere' Samples, bei denen alle Klassifikatoren versagen, so sinkt die kumulierte Fehlerrate auch im Grenzfall unendlich vieler Klassifikatoren nicht auf Null.

In der Praxis liegen niemals unendlich viele Klassifikatoren vor. Werte von $2 \leq N < 10$ sind eher realistisch. Die Entscheidung für ein bestimmtes Ergebnis ist bei endlich vielen Klassifikatoren deutlich schwieriger als im theoretischen Fall unendlich vieler Klassifikatoren. Hier wirken sich Korrelationen zwischen den einzelnen Systemen schwerwiegend aus. Machen beispielsweise zwei von drei verwendeten Klassifikatoren an denselben Stellen die gleichen Fehler - etwa weil sie ähnlich aufgebaut sind - so wird das Gesamtergebnis stets diesen Fehler enthalten. Es ist also wichtig, möglichst unterschiedliche Systeme, die unterschiedliche Fehler machen, zu konstruieren. Dies ist jedoch in den meisten Fällen nur unter großem Aufwand möglich.

Fiscus [31] kombinierte die Hypothesen von insgesamt 5 Erkennern, um eine verbesserte Ausgabe zu erhalten. Er verwendete dabei ein gewichtetes Mittel aus der Wortkonfidenz und der Worthäufigkeit in den unterschiedlichen Hypothesen, wobei der Gewichtungsfaktor auf einer Kreuzvalidierungsstichprobe gelernt wurde. Durch die Hypothesenkombination konnte die Gesamtfehlerrate um 12% relativ gesenkt werden. Die Bedingungen für das Experiment waren sehr günstig: die fünf Erkenner stammten von unterschiedlichen Forschergruppen in unterschiedlichen Labors und waren unabhängig voneinander entwickelt worden; wiesen also große Unterschiede zueinander auf.

Im View4You-System stehen nach dem unüberwachten Training zwei unterschiedliche Systeme zur Verfügung, von denen das eine mit 32 Stunden, und das andere mit 53 Stunden Daten trainiert wurde (vgl. Bild 9.3). Die beiden Systeme weisen eine vergleichbare Fehlerrate auf. Es liegt daher nahe, die Ergebnisse zu kombinieren, um eventuell eine niedrigere Gesamtfehlerrate zu erzielen. Da es mit nur zwei Hypothesen nicht möglich ist, eine 'Mehrheitsentscheidung' herbeizuführen, mussten andere Entscheidungskriterien verwendet werden. Es wurden zwei unterschiedliche Kriterien untersucht: das Sprachmodellkriterium und die Wortkonfidenz.

Zur Illustration der beiden Kriterien soll folgendes Beispiel dienen. Gegeben seien die beiden Hypothesen der Spracherkenner A und B, von denen bekannt ist - z.B. durch Messung auf einer unabhängigen Stichprobe - dass A im Mittel eine niedrigere Fehlerrate aufweist als B:

```
A: Guten am meine Damen und Herren Bundeskanzler Schroeder und der
B: Guten am meine Damen und Lehrer Bundeskanzler Schroeder     der
       (1)                  (2)                           (3)

A: chinesische Parteichef    Peng
B: chinesische Parteichef Li Peng
                         (4)
```

Fehler (1) kann durch Hypothesenkombination nicht korrigiert werden, da er in beiden Hypothesen gleichzeitig auftritt. Fehler (2) ist eine Substitution; hier muss entschieden werden, ob das Wort 'Herren' aus Hypothese A oder 'Lehrer' aus Hypothese B verwendet wird. Fehler (3) und (4) sind Löschungen bzw. Einfügungen, hier muss entschieden werden, ob die entsprechenden Worte 'und' bzw. 'Li' in der endgültigen Ausgabe auftreten sollen oder nicht.

Konfidenzkriterium

Für das Konfidenzkriterium wird für jedes Wort der beiden Hypothesen mit dem *gamma*-Konfidenzmaß die a-posteriori-Wahrscheinlichkeit der korrekten Erkennung berechnet. Die drei möglichen Unterschiede zwischen Hypothese A (die a-priori vermutlich bessere, d.h. mit der niedrigeren Fehlerrate) und B werden wie folgt behandelt:

Substitutionen (Fehler (2))	wähle A wenn $\gamma_s c_A > c_B$, sonst B
Löschungen (Fehler (3))	wähle A wenn $\gamma_d c_A > 0.5$, sonst keine Ausgabe
Einfügungen (Fehler (4)	wähle B wenn $\gamma_i c_B > 0.5$, sonst keine Ausgabe

c_A und c_B sind dabei die Konfidenzen des betrachteten Wortes in Hypothese A bzw. B. Die Faktoren γ_s, γ_d und γ_i sind freie Parameter. Sie müssen auf einer unabhängigen Stichprobe eingestellt werden. Steht keine solche zur Verfügung, wählt man $\gamma_s = \gamma_d = \gamma_i = 1.0$.

Sprachmodellkriterium

Die häufigsten korrigierbaren Fehler sind Substitutionsfehler (Fehlertyp 2 im obigen Beispiel). Es muss entschieden werden, ob die gemeinsame Hypothese an dieser Stelle Wort X oder Wort Y lauten soll. Da das statistischen Sprachmodell die Aufgabe hat, die Wahrscheinlichkeit von Wortfolgen zu modellieren, kann es zur Entscheidung gut herangezogen werden. Das Sprachmodell stellt Wahrscheinlichkeiten der Form

$$p(W|H) = p(w_N|w_1 w_2 ... w_{N-1}) \tag{9.4}$$

(typischerweise mit $N = 3$) zur Verfügung. Die Entscheidung zwischen Wort X und Wort Y ('Herren' und 'Lehrer' im obigen Beispiel) erfolgt, indem die beiden Wahrscheinlichkeiten p('Lehrer'|'guten am meine Damen und') und p('Herren'|'guten am meine Damen und') berechnet und miteinander verglichen werden. Das Wort mit der höheren vom Sprachmodell vorausgesagten Wahrscheinlichkeit wird selektiert.

Experimente

Die Hypothesen des auf 204 Sendungen 'tagesschau' trainierten Erkenners I57 wiesen die niedrigste Fehlerrate aller unüberwacht trainierten Systeme (21,4%) auf. Das Ziel der Hypothesenkombination war es, diese Fehlerrate noch weiter zu verringern. Hierzu wurden die Hypothesen des auf 120 Sendungen unüberwacht trainierten Systems (I56) mit einer Fehlerrate von 21,5% verwendet, und sowohl mit dem Sprachmodellkriterium, als auch mit dem Konfidenzkriterium evaluiert. Die freien Parameter γ_s, γ_d und γ_i wurden dabei auf die Werte $\gamma_s = 1.05$, $\gamma_d = 1.0$ und $\gamma_i = 1.0$ festgelegt.

System	Fehlerrate
I56	21.5%
I57	21.4%
kombiniert (Sprachmodellkriterium)	21.0%
kombiniert (Konfidenzmaß)	20.5%

Tabelle 9.9: Verbesserung des besten unüberwachten Systems I56 durch Hypothesenkombination

Das Konfidenzmaß als Kriterium zur Hypothesenkombination führt zu einer deutlich besseren Fehlerreduktion als das Sprachmodell.

Es ist auch möglich, mit Hilfe des unüberwacht trainierten Erkenners die Fehlerrate des besten Systems I39, das mit Hilfe aller verfügbaren Transkripte trainiert wurde, zu verbessern. Hierzu wurden die Hypothesen des Systems I39 mit denen des - auf 120 untranskribierten Sendungen trainierten - Erkenners I56 kombiniert. Die Ergebnisse zeigt Tabelle 9.10.

System	Fehlerrate
I39	19.5%
I56	21.5%
kombiniert (Sprachmodellkriterium)	19.2%
kombiniert (Konfidenzmass)	18.5%

Tabelle 9.10: Verbesserung des besten Systems I39 durch Hypothesenkombination

Zusammenfassend lässt sich sagen, dass die Fehlerrate sowohl des besten unüberwachten als auch des besten überwacht trainierten Systems durch Hypothesenkombination mit einem anderen, jeweils unüberwacht trainierten Erkenner um 1% absolut reduzieren lässt. Als Kriterium für die Selektion konkurrierender Worte in zwei Hypothesen wurde sowohl die Wortkonfidenz als auch der Sprachmodellscore untersucht. Zwar führten beide Selektionskriterien zu einer reduzierten Fehlerrate, aber die Verbesserungen waren bei Verwendung der Wortkonfidenz deutlich größer.

9.6 Zusammenfassung

In diesem Kapitel wurde ein Algorithmus zum unüberwachten Lernen eines Spracherkenners vorgestellt und evaluiert. Ausgehend von einem auf nur zwei Sendungen 'tagesschau' trainierten initialen Erkenner mit einer Fehlerrate von über 32%, konnte der Trainingsalgorithmus durch Verwendung von weiteren 204 Aufnahmen (entsprechend 7 Monaten täglichem Ansehen der 'tagesschau') eine Fehlerrate von 20,5% erzielen. Das beste auf transkribiertem Material trainierte System erzielte demgegenüber eine Fehlerrate von 19,5%.

Durch Hypothesenkombination des unüberwachten mit dem überwacht trainierten System war es weiterhin möglich, die Fehlerrate des besten Erkenners von 19,5% auf 18,5% zu senken.

Damit konnte gezeigt werden, dass mit unüberwachtem Lernen nach dem vorgestellten Algorithmus ähnliche Ergebnisse erzielbar sind wie beim Training auf einer mittelgroßen Menge transkribierter Daten. Der Aufwand an Handarbeit, der zum unüberwachten Lernen erforderlich ist, ist dabei jedoch um ein Vielfaches geringer, da der Transkriptionsaufwand entfällt. Der vorgestellte Algorithmus eignet sich daher besonders, um

- sehr schnell Spracherkenner für neue Domänen oder Sprachen zu erzeugen (*rapid prototyping*)
- effizient Bootstrap-Systeme in neuen Domänen oder Sprachen zu erzeugen

- Spracherkennungsapplikationen, die nicht die höchstmögliche Erkennungsakkuratheit benötigen, preiswert und schnell zu konstruieren
- unabhängige Erkenner zu produzieren, um durch Hypothesenkombination die Fehlerrate eines state-of-the-art-Systems noch weiter zu senken

9.7 Aufstellung der Trainingsdaten und Systeme

Die folgenden beiden Tabellen führen die Aufnahmen, die den beschriebenen Experimenten zugrunde lagen, sowie die im vorangegangenen Abschnitt beschriebenen Erkenner noch einmal zusammenfassend auf. Eine Darstellung der Größe der Trainingskorpora, ausgewertet nach laufenden Worten, Zeit und Vokabular, wurde bereits in Tabelle 3.2 auf Seite 25 gegeben.

Datenart	Sendungen vom (Datum)	Anzahl
transkribierte Trainingsdaten (immer benutzt)	25.11.96, 26.11.96	2
transkribierte Trainingsdaten	27.11.96, 28.11.96, 02.12.96, 05.12.96, 06.12.96, 09.12.96, 10.12.96, 16.12.96, 17.12.96, 21.01.97, 24.01.97, 27.01.97, 28.01.97, 25.02.97, 03.04.97, 04.04.97, 05.04.97, 06.04.97, 07.04.97, 08.04.97, 09.04.97, 10.04.97, 11.04.97, 12.04.97, 27.06.97, 27.08.97, 26.11.97, 27.11.97, 01.06.98, 02.06.98, 03.06.98, 04.06.98, 05.06.98, 08.06.98, 09.06.98, 10.06.98, 11.06.98, 12.06.98, 13.06.98, 14.06.98, 15.06.98, 16.06.98, 17.06.98, 18.06.98, 19.06.98, 20.06.98, 21.06.98, 22.06.98, 24.06.98, 25.06.98, 01.07.98, 03.07.98, 07.07.98, 08.07.98, 09.07.98, 10.07.98, 11.07.98, 12.07.98	58
untranskribierte Trainingsdaten	01.03.97, 01.03.98, 01.04.98, 01.09.97, 01.09.98, 02.03.98, 02.04.98, 02.05.98, 02.09.98, 03.02.98, 03.03.97, 03.03.98, 03.04.98, 03.09.97, 04.03.97, 04.04.98, 04.05.98, 05.02.98, 05.03.97, 05.04.98, 05.05.98, 05.09.97, 06.04.98, 06.05.98, 06.09.97, 07.04.98, 07.05.98, 07.09.97, 08.04.98, 08.05.98, 08.06.97, 09.04.98, 09.05.98, 09.09.97, 10.02.97, 10.03.98, 10.05.98, 10.06.97, 11.03.98, 11.04.98, 11.05.98, 11.06.97, 11.08.98, 11.09.97, 12.05.98, 12.06.97, 12.08.98, 13.02.97, 13.04.97, 13.05.98, 14.02.97, 14.03.98, 14.05.98, 15.02.97, 15.02.98, 15.03.98, 15.05.98, 16.01.98, 16.02.97, 16.03.98, 16.05.98, 16.06.97, 16.09.97, 17.02.98, 17.03.98, 17.04.98, 17.05.98, 17.08.97, 17.09.97, 18.03.97, 18.03.98, 18.04.98, 18.05.98, 18.06.97, 19.01.98, 19.03.97, 19.03.98, 19.04.98, 19.05.98, 19.06.97, 19.07.98, 19.08.97, 20.01.98, 20.03.97, 20.03.98, 20.04.98, 20.05.98, 20.06.97, 20.07.98, 20.08.97, 20.09.97, 21.01.98, 21.03.98, 21.04.98, 21.05.98, 21.06.97, 21.07.98, 21.08.97, 21.09.97, 22.02.98, 22.03.98, 22.04.98, 22.05.98, 22.06.97, 22.07.98, 22.08.97, 22.08.98, 23.04.98, 23.05.98, 23.07.98, 23.08.97, 23.08.98, 24.02.97, 24.03.98, 24.04.98, 24.07.98, 24.08.97, 24.08.98, 25.03.98, 25.04.98, 25.08.97, 25.08.98, 26.03.98, 26.04.98, 27.01.98, 27.02.98, 27.03.98, 27.04.98, 28.02.97, 28.02.98, 28.03.98, 28.04.98, 28.05.97, 28.05.98, 28.08.98, 29.04.98, 29.05.98, 29.08.97, 29.08.98, 30.03.97, 30.03.98, 30.04.98, 30.05.98, 30.06.97, 30.08.98, 31.03.98, 31.08.97, 31.08.98	148
Testdaten	30.03.97, 13.04.97, 28.05.97, 30.06.97	4

Tabelle 9.11: Daten für das akustische Training

System	Trainingsdaten	Fehlerrate
I41	2 Sendungen transkribiert	32,1%
I39	60 Sendungen transkribiert	19,5%
I33_0_5	2 transkribiert 24 untranskribiert	24,2%
I56	2 transkribiert + 120 untranskribiert	21,5%
I57	2 transkribiert + 204 untranskribiert	21,4%
I56+I57	2 transkribiert + 204 untranskribiert	20,5%

Tabelle 9.12: Übersicht über die verschiedenen Systeme

Kapitel 10

Systemevaluation

Das View4You-System stellt einen voll funktionsfähigen Prototypen für eine automatisch erstellte Videodatenbank dar. Obwohl an verschiedenen Orten an den einzelnen Komponenten geforscht wird, die im View4You-System eingesetzt werden, gibt es in ganz Deutschland kein einziges anderes vollständiges System dieser Art. Insofern sind Untersuchungen über das Zusammenspiel der einzelnen Komponenten und die Performanz des Gesamtsystems von besonderem Interesse.
Insbesondere sind die folgenden Fragestellungen relevant:

1. Wie gut ist das Gesamtsystem? Lässt die Performanz überhaupt einen praktischen Einsatz zu?
2. Wie groß ist der Einfluß der Fehler, die der *Spracherkenner* macht, auf die Systemeigenschaften?
3. Welchen Einfluß hat der *Segmentierer* auf die Systemeigenschaften?
4. Wie gut eignet sich der gewählte *information-retrieval*-Ansatz für die Datenbank, d.h. wie gut wäre das System bei perfekter Segmentierung und perfekter Spracherkennung?

10.1 Methodik

Die Untersuchung der Systemperformanz erfolgte nach dem folgenden Schema:

1. Naive Benutzer werden gebeten, einen Satz von Fragen an das System zu formulieren.
2. in der Datenbank des Systems wird manuell bestimmt, welche Zeitausschnitte welcher Sendungen relevant in Bezug auf die einzelnen Fragen sind.
3. die Fragen werden dem System zur Beantwortung vorgelegt.

4. durch Vergleich der Systemantwort mit der (in Schritt 2 ermittelten) erwünschten Systemantwort werden Kennzahlen ermittelt, die über die Systemleistung Aufschluß geben.

Die Fragen an das System wurden von drei unterschiedlichen potentiellen Benutzern formuliert. Den Testpersonen wurde dabei zunächst das System anhand eines Schaubildes erläutert. Sie wurden dann aufgefordert, jeweils Fragen aufzuschreiben, die sie - ihrer Meinung nach - in dieser Formulierung an ein solches System richten würden. Von diesen Fragen wurden zehn für die endgültige Evaluation ausgesucht:

1. Gibt es Berichte über Jerusalem?
2. Wird Helmut Kohl noch einmal kandidieren?
3. Ich möchte die Lottozahlen wissen!
4. Gibt es etwas zu Benjamin Netanjahu?
5. Mich interessiert, was in den letzten Tagen alles in Afrika geschehen ist!
6. Mich interessiert, wie die Lage in Albanien ist.
7. Mich interessieren die Ergebnisse der Bundesliga!
8. Gibt es Berichte über Flüchtlinge?
9. Ich möchte die Berichte über den Besuch des Bundespräsidenten Herzog in Japan sehen!
10. Gibt es etwas Neues im Mykonos-Prozess?

Interessant ist die geringe Varianz der Formulierungen. Bis auf die Frage 2 beginnt jede Formulierung entweder mit 'Ich möchte.... wissen, sehen', mit 'Mich interessiert...' oder mit 'Gibt es etwas über...'. Dies läßt eine semantische Auswertung der Anfrage mit Hilfe eines relativ einfachen Parsers realisierbar erscheinen.

Für große Datenbanken ist der zweite Schritt der Evaluation, das manuelle Durchsehen der gesamten Datenbank im Hinblick auf die gestellten Fragen, prohibitiv zeitaufwendig. Daher wurde eine kleine Datenbank mit insgesamt 10 Sendungen 'tagesschau' ausgewählt, die mit moderatem Aufwand manuell durchgesehen werden kann. Die 10 Sendungen in dieser Datenbank wurden zwischen dem 3.4.97 und dem 12.4.97 ausgestrahlt. Die auf dieser kleinen Datenbank erzielten Ergebnisse konnten mittlerweile auch auf einer wesentlich größeren Datenmenge validiert werden.

Bei der Auswahl der als 'korrekt' markierten Segmente wurde von der wahrscheinlichen Intention des Fragenden ausgegangen. Mehrere Fragen schränken den zeitlichen Bereich, in dem gesucht werden soll, explizit (Frage 5) oder implizit (Fragen 3 und 7)

ein. Für die Auswertung wurde diese Einschränkung ignoriert. Für Frage 7 wurden daher sämtliche Beiträge über Spiele der Bundesliga, für Frage 5 alle Berichte aus irgendeinem Land Afrikas ohne zeitliche Beschränkung, und für Frage 3 alle Lotto-Ausspielungen als relevant angesehen - auch wenn der Fragende bei der Frage nach den Lottozahlen vermutlich nur an der zeitlich nächsten Ausstrahlung interessiert war.

In Tabelle 10.1 ist zusammengefaßt, wieviele Berichte es zu jeder der zehn Fragen in der Testdatenbank tatsächlich gibt. Die beiden Spalten 'Anzahl Berichte' und 'Anzahl Segmente' beziehen sich auf unterschiedliche Segmentierungen. Ein 'Segment' ist hierbei ein akustisch homogener Teil einer Sendung, beispielsweise ein Korrespondentenbericht. Eine Segmentgrenze wird dort angenommen, wo ein Wechsel der akustischen Parameter eintritt, wo also z.B. ein Sprecherwechsel oder ein Wechsel zwischen Ansagesprecher und Korrespondent vorliegt. Die Aufteilung in 'Segmente' nach dieser Definition ist auch diejenige, die vom Segmentierer des View4You-Systems angestrebt wird. Die 'Segmente' in Tabelle 10.1 sind die tatsächlichen akustischen Segmente, die von Hand bestimmt wurden. Ein 'Bericht' ist eine semantische Einheit, die ein oder mehrere direkt aufeinander folgende 'Segmente' desselben Themas beinhaltet. Da 'Segmente' und 'Berichte' unterschiedlich lang sind, ist auch die Gesamtlänge des relevanten Videomaterials zu jeder Frage angegeben.

Frage	Länge (s)	Anzahl Berichte	Anzahl Segmente
1 (Jerusalem)	699.1	10	25
2 (Kohl)	330.6	4	22
3 (Lotto)	108.9	3	3
4 (Netanjahu)	678.1	10	24
5 (Afrika)	597.9	8	23
6 (Albanien)	414.3	7	16
7 (Bundesliga)	388.6	6	10
8 (Flüchstlinge)	246.7	6	11
9 (Herzog)	168.7	3	8
10 (Mykonos)	584.7	5	21
zusammen	2264.9	62	163
ganze Datenbank	9207.9	174	365

Tabelle 10.1: Tatsächliches Auftreten der relevanten Ereignisse in den 10 Tagesschauen der Testdatenbank

In der Zeile 'ganze Datenbank' ist die Größe der gesamten Datenbank, in der gesucht wird, dargestellt. Die 10 Fragen beziehen sich in ihrer Summe auf etwa ein Viertel der Datenbank; die restlichen drei Viertel sind 'Rauschen'.

10.2 Evaluation bei manueller Segmentierung und manueller Transkription

In einer ersten Evaluation wurden die zehn Sendungen (vom 3.4.97 bis 12.4.97) manuell segmentiert und transkribiert in die Test-Datenbank eingespeist. Die zehn oben definierten Fragen wurden an die Datenbank des View4You-Systems gerichtet und die von der Datenbank zurückgelieferten Segmente wurden evaluiert, ob sie (jeweils) für die Frage relevant waren oder nicht.

Das Experiment wurde auf zwei unterschiedlichen manuellen Segmentierungen durchgeführt: auf einer Segmentierung in semantisch definierte 'Berichte', und einer Segmentierung in akustisch motivierte 'Segmente'. Die Ergebnisse der Evaluation sind in den Tabellen 10.2 und 10.3 zusammengefaßt.

Frage	tatsächlich vorhanden	insgesamt gefunden	gefunden (richtig)	gefunden (falsch)	PRC	RCL
1 (Jerusalem)	10	7	7	0	1.0	0.7
2 (Kohl)	4	6	4	2	0.67	1.0
3 (Lotto)	3	2	2	0	1.0	0.67
4 (Netanjahu)	10	6	6	0	1.0	0.6
5 (Afrika)	8	2	1	1	0.5	0.12
6 (Albanien)	7	8	7	1	0.87	1.0
7 (Bundesliga)	6	3	3	0	1.0	0.5
8 (Flüchtlinge)	6	7	6	1	0.86	1.0
9 (Herzog)	3	6	3	3	0.5	1.0
10 (Mykonos)	5	8	4	4	0.5	0.8
zusammen	62	55	43	12	0.78	0.69

Tabelle 10.2: Endergebnis bei manueller Segmentierung in Berichte und manueller Transkription

Legt man eine perfekte Segmentierung in Berichte zugrunde, werden maximal 69% aller relevanten Berichte gefunden, und nur 22% aller gefundenen Berichte sind irrelevant. Bei einer perfekten Segmentierung in akustische Segmente werden nur 44% aller relevanten Segmente gefunden; die gefundenen Segmente sind aber zu 85% korrekt.

Die berichtorientierte Segmentierung weist eine deutlich bessere Finderate (Recall, RCL) auf. Dies ist zwangsläufig deshalb der Fall, weil akustische Segmente im Fall der manuellen Segmentierung eine reine Binnensegmentierung der Berichte darstellen: ein Bericht besteht aus einem oder mehreren Segmenten. Wird ein Segment (bei der segmentbasierten Evaluation) korrekt gefunden, wird automatisch - da das Segment Teil des Berichts ist - bei der berichtbasierten Evaluation auch der Bericht korrekt gefunden. Umgekehrt jedoch wird, wenn ein Bericht gefunden wurde, deshalb nicht automatisch jedes Segment des Berichts gefunden. Die Recall (RCL) der seg-

Frage	tatsächlich vorhanden	insgesamt gefunden	gefunden (richtig)	gefunden (falsch)	PRC	RCL
1 (Jerusalem)	25	7	7	0	1.0	0.28
2 (Kohl)	22	12	10	2	0.83	0.45
3 (Lotto)	3	2	2	0	1.0	0.67
4 (Netanjahu)	24	12	12	0	1.0	0.5
5 (Afrika)	23	2	1	1	0.5	0.04
6 (Albanien)	16	14	13	1	0.93	0.81
7 (Bundesliga)	10	3	3	0	1.0	0.3
8 (Flüchtlinge)	11	7	6	1	0.86	0.55
9 (Herzog)	8	11	7	4	0.64	0.88
10 (Mykonos)	21	14	10	4	0.71	0.48
zusammen	163	84	71	13	0.85	0.44

Tabelle 10.3: Endergebnis bei manueller Segmentierung in akustische Segmente und manueller Transkription

mentbasierten Evaluation muß daher immer schlechter oder kann höchstens gleich gut im Vergleich zur Recall der berichtbasierten Evaluation sein. Nimmt man an, dass die fehlerhaften Ergebnisse auf eine Anfrage sich mehr oder weniger zufällig in der Datenbank verteilen, werden diese bei der segmentbasierten Evaluation im Mittel relativ weit auseinanderliegen, d.h. in verschiedene Berichte fallen. Die Anzahl der fehlerhaft gefundenen *Berichte* wird daher nur wenig niedriger liegen als die Anzahl der fehlerhaft gefundenen *Segmente*. Da insgesamt die Anzahl der zu findenden Entitäten bei der berichtbasierten Evaluation kleiner ist, sinkt die Präzision (tendenziell) beim Übergang von der segmentbasierten zur berichtbasierten Evaluation.

Betrachtet man die Ergebnisse der (berichtbasierten) Evaluation in Tabelle 10.2, stellt man trotz perfekter Segmentierung und 'Erkennung' fest, dass ein Drittel der Berichte nicht gefunden wird und dass jeder fünfte präsentierte Bericht mit der Anfrage nichts zu tun hatte. Worauf läßt sich diese mangelnde Performanz zurückführen?

Gründe für niedrige Präzision (fälschlicherweise gefundene Berichte)

Praktisch alle falsch gefundenen Berichte wurden deshalb gefunden, weil eine semantische Einschränkung der Anfrage nicht beachtet wurde. Bei der Frage nach der erneuten Kanzlerkandidatur von Helmut Kohl wurden 2 Berichte über Helmut Kohl gefunden, in denen es nicht um die Kanzlerkandidatur ging (sondern um die Steuerreform). Bei der Frage nach der Lage in Albanien wurde ein Wetterbericht zurückgeliefert ('ein Tief über **Albanien** zieht nordwestwärts und beherrscht in den nächsten Tagen unser Wetter'). Die Frage nach dem Besuch des Bundespräsidenten Herzog in Japan produzierte Berichte über den österreichischen **Bundespräsidenten** Klestil, über den

Besuch des Papstes in Sarajewo und den **Besuch** des Bundesverteidigungsministers ebenfalls in Sarajewo. Bei der Frage nach dem Mykonos-Prozess wurden insgesamt 4 andere Berichte über **Prozesse** (gegen Giftmüllsünder, Skinheads usw.) gefunden.

Bei der Behandlung der Frage 'Mich interessiert, was in den letzten Tagen alles in Afrika geschehen ist' sorgte das Schlüsselwort 'geschehen' für eine Fehlzuordnung.

Mögliche Abhilfen

Die folgenden Verbesserungen am View4You-System könnten dazu beitragen, die Präzision der zurückgelieferten Antworten signifikant zu erhöhen:

1. Parsen der Eingabe mit dem Ziel, die Schlüsselwörter zu extrahieren und nur diese der Datenbank zu präsentieren
2. Einführung von Boole'schen Ausdrücken ('Bundespräsident **AND** Herzog **AND** Japan'
3. Negationen ('Albanien aber nicht Wetter')
4. explizite Selektion eines relevanten Zeitbereiches mit Hilfe der Benutzerschnittstelle

Idealerweise sollte ein Parser in der Lage sein, die Informationen, die zur Boole'schen Verknüpfung der Suchbegriffe relevant sind, aus der natürlichsprachlich formulierten Anfrage zu extrahieren.

Gründe für niedriges Recall (nicht gefundene Berichte)

Der Grund für die Mehrzahl der nicht gefundenen Berichte ist mangelndes Weltwissen.

Die mit Abstand niedrigste RCL (12%) weist die Frage nach den Ereignissen 'in Afrika' auf. Hierbei wurden Berichte über Ruanda, Algerien und Tansania nicht gefunden, weil dem System nicht bekannt ist, dass diese Staaten geographisch in Afrika liegen. Die zweitniedrigste RCL (50%) bei der Frage nach den Ergebnissen der Bundesliga entsteht aus demselben Grund: ein Bericht über den 'VfL Bochum' hat für das System nichts mit dem Begriff 'Bundesliga' zu tun.

Das Problem mangelnden Weltwissens ließe sich durch die Verwendung geeignet konstruierter Thesauri und Synonymwörterbücher reduzieren. Eine Übersicht über verschiedene Verfahren zur Erzeugung von Thesauri findet sich in [41].

10.3 Evaluation bei manueller Segmentierung und maschineller Spracherkennung

Eine sehr interessante Frage ist die nach dem Einfluß der Fehler, die der Spracherkenner macht, auf die Performanz des Gesamtsystems. Um diese zu bestimmen, wurden

insgesamt 3 Auswertungen mit unterschiedlichen Spracherkennern durchgeführt, wobei sich die verwendeten Erkenner in Fehlerrate und Echtzeitfaktor unterschieden:

1. ein Erkenner entsprechend dem Bootstrap-Erkenner für das unüberwachte Training, etwas geschwindigkeits-optimiert (30-40 Echtzeitfaktoren), mit einer Fehlerrate von 33,7%

2. der auf 53 Stunden 'tagesschau' trainierte Erkenner I57 mit einer Fehlerrate von 21,4% und etwa 90-120 Echtzeitfaktoren Zeitaufwand

3. der im Prototyp integrierte, geschwindigkeits-optimierte Erkenner I63bbi, mit einer Fehlerrate von 22,7% und etwa 15-20 Echtzeitfaktoren Zeitaufwand

System	Fehlerrate Spracherkenner	PRC	RCL
I14bbi	33,7%	0.73	0.58
I63bbi	22,7%	0.75	0.65
I57	21,5%	0.75	0.66
Transkripte	0%	0.78	0.69

Tabelle 10.4: Endergebnis bei manueller Segmentierung in Berichte und maschineller Spracherkennung

Wie man an den Ergebnissen sieht, erzielt das unüberwacht trainierte System I57 mit 75% PRC bzw. 66% PRC nur um jeweils 3% schlechtere Kennzahlen als die Transkriptionen. Das einen Faktor 6 schnellere, im Prototyp integrierte System I63bbi liegt unwesentlich schlechter. Die erhöhte Fehlerrate des Systems I14bbi macht sich vor allem in einer deutlichen Verringerung der Finderate (RCL) bemerkbar.

Ein Vergleich mit Werten aus der Literatur ist schwierig, da weder die Größe der Datenbank, noch die Art der Fragen vergleichbar ist. Vergleichbare Untersuchungen wurden von Wactlar et al im Informedia-Projekt [7], [57], sowie in neuerer Zeit innerhalb der TREC-Konferenzen [23] durchgeführt. Hierbei werden eine Vielzahl unterschiedlicher Metriken zur Evaluierung herangezogen, so daß zum Vergleich häufig Umrechnungen erfolgen müssen. Ein direkter Vergleich der erzielten Kennzahlen führt aus den beschriebenen Gründen in die Irre. Qualitativ wird jedoch durch Wactlars Untersuchungen das Ergebnis bestätigt, daß die Wortfehlerrate des Spracherkenners, sofern sie im Bereich von 0% bis etwa 30% liegt, nur einen sehr geringen Einfluß auf die Performanz des Gesamtsystems hat.

10.4 Stopwords und morphologische Zerlegung

Die Liste der semantisch irrelevanten Worte (der sog. stopwords, siehe hierzu auch das Kapitel 'Die Datenbank des View4You Systems') besteht aus den 500 Worten des

Sprachmodellkorpus. Diese setzen sich zusammen aus

1. die 100 häufigsten Worte des Sprachmodellkorpus (inklusive Nomina)

2. die 400 häufigsten Nichtnomina des Sprachmodellkorpus

Durch die unterschiedliche Behandlung von Nomina wird erreicht, dass nur wenige der potentiell eher bedeutungsrelevanten Nomina in die Liste der semantisch irrelevanten Worte aufgenommen werden.

In einer Evaluation wurde die Systemperformanz mit und ohne Stopword-Liste gemessen. Zur Segmentierung wurde die manuelle Segmentierung in Berichte und als Spracherkenner der im Prototyp integrierte Erkenner I63bbi herangezogen. Die Ergebnisse mit und ohne Stopwortliste zeigt Tabelle 10.5. Aufgrund der sowohl in der Anfrage als auch in den Berichten auftauchenden semantisch irrelevanten Wörtern ('und', 'er', 'den') wird eine sehr große Zahl von Berichten fälschlicherweise gefunden. Das Verfahren ist in dieser Form nicht brauchbar.

System	PRC	RCL
mit Stopwordliste	0.75	0.65
ohne Stopwordliste	0.07	0.81

Tabelle 10.5: Ergebnis mit und ohne Stopwordliste

In einem weiteren Experiment wurde die in die Datenbank eingebaute morphologische Stammbildung (engl. stemming) abgeschaltet. Das Resultat ist in Tabelle 10.6 zusammengefasst. Offensichtlich ist das stemming der Systemperformanz zuträglich.

System	PRC	RCL
mit stemming	0.75	0.65
ohne stemming	0.66	0.63

Tabelle 10.6: Ergebnis mit und ohne stemming

Es ist auch möglich, die Normierung auf Kleinbuchstaben in der Datenbank ab- und anzuschalten. Normalerweise werden alle Worte auf Kleinschreibung kanonisiert. Ohne diese Kanonisierung sinkt die Performanz des Systems auf PRC 0.68 und RCL 0.65 im Vergleich zur PRC 0.75 und RCL 0.65 mit Kanonisierung (Tabelle 10.7).

System	PRC	RCL
mit Groß- und Kleinbuchstaben	0.68	0.65
nur Kleinbuchstaben	0.75	0.65

Tabelle 10.7: Ergebnis mit und ohne Dekapitalisierung

10.5 Evaluation bei automatischer Segmentierung und maschineller Spracherkennung

Die bisher stets getroffene Annahme von fehlerfreier Segmentierung liegt im realen View4You-System natürlich nicht vor. Daher wurde eine Evaluation auf der Segmentierung des (modellbasierten, s. Seite 55) Segmentierers durchgeführt. Das Ergebnis - wieder mit dem integrierten Erkenner I63bbi - zeigt Tabelle 10.8.

Segmentierung	PRC	RCL
perfekt (Beiträge)	0.75	0.65
perfekt (akust. Segmente)	0.83	0.40
Standardsegmentierer	0.85	0.30

Tabelle 10.8: Ergebnis mit verschiedenen Segmentierungen

Durch die Übersegmentierung des Standardsegmentierers wird jeder Beitrag in mehrere Segmente aufgeteilt. Diese kurzen Segmente enthalten zu einem relativ großen Teil die relevanten Schlüsselwörter nicht mehr, so dass die Finderate - auf diesen Segmenten gerechnet - relativ niedrig ist.

Die Evaluation nach der *Anzahl* der gefundenen Segmente sagt nichts über die Länge der Segmente aus. Es ist interessant, eine Auswertung nach der *Zeit* durchzuführen. Hierbei wird gemessen, wieviele Sekunden Video vom System korrekt gefunden werden, und wieviele Sekunden irrelevanter Daten der Benutzer über sich ergehen lassen muß. Das Ergebnis ist in der folgenden Tabelle dargestellt.

Frage	tatsächlich vorhanden	gefunden (richtig)	gefunden (falsch)	PRC	RCL
1 (Jerusalem)	699	73	0	1.0	0.10
2 (Kohl)	330	220	80	0.73	0.66
3 (Lotto)	109	6	0	1.0	0.06
4 (Netanjahu)	678	231	0.2	1.0	0.34
5 (Afrika)	598	14	73	0.16	0.02
6 (Albanien)	414	324	21	0.94	0.78
7 (Bundesliga)	389	42	4	0.92	0.11
8 (Flüchtlinge)	246	148	1.5	0.99	0.60
9 (Herzog)	169	108	127	0.46	0.64
10 (Mykonos)	585	276	89	0.76	0.47
zusammen	4218	1441	396	0.78	0.34

Tabelle 10.9: Endergebnis: automatische Segmentierung und Spracherkenner I63bbi; nach Zeit ausgewertet

Ausgesprochen auffällig ist die schlechte Performanz bei der Frage nach Jerusalem. Diese ist durch einige lange Berichte über die Demonstrationen palästinensi-

scher Jugendlicher gegen die israelische Siedlungspolitik, u.a. am Berg Bar Homa in Ostjerusalem, mitverursacht. Diese Berichte wurden als relevant zum Thema Jerusalem klassifiziert. Werden diese Berichte als irrelevant eingestuft, steigt RCL bei der Jerusalem-Frage auf 0.17 und insgesamt von 0.342 auf 0.365.

10.6 Evaluation bei verschiedenen Segmentierungen

Vergleicht man die Ergebnisse bei verschiedenen Segmentierungen miteinander, muß die Auswertung zeitorientiert erfolgen. Eine Auswertung nach der Anzahl der Segmente hat bei unterschiedlicher Anzahl und Länge der Segmente möglicherweise wenig Bedeutung.

Tabelle 10.10 faßt einige der Ergebnisse der bereits vorgestellten Evaluationen, diesmal nach Zeit ausgewertet, zusammen.

Segmentierer	Spracherkenner (Wortfehlerrate)	PRC	RCL	F
manuell (Berichte)	Transkripte	0.72	0.79	0.75
manuell (Segmente)	Transkripte	0.84	0.46	0.60
manuell (Segmente)	I63bbi (22.7%)	0.81	0.43	0.56
modellbasiert	I63bbi (22.7%)	0.78	0.34	0.48
modellbasiert	I57 (21.5%)	0.78	0.37	0.50
modellbasiert	I14 (33.7%)	0.77	0.29	0.42

Tabelle 10.10: Endergebis: Auswertung nach Zeit

Beim Vergleich der F-Maße miteinander fällt ins Auge, dass der Spracherkenner eine Verschlechterung um 0.04, der Segmentierer eine Verschlechterung von 0.08 gegenüber der Referenz verursacht. Es liegt daher nahe, den - besseren - hybriden Segmentierer einzusetzen, um die Gesamtperformanz zu optimieren. Die Frage, welcher Arbeitspunkt dabei für den hybriden Segmentierer zu wählen ist, läßt sich dabei nur experimentell entscheiden. Drei verschiedene Arbeitspunkte des Segmentierers wurden ausgewählt. Die Arbeitspunkte werden durch die Werte in Tabelle 10.11 charakterisiert.

Grenze	PRC	RCL	F
20	0.687	0.851	0.760
60	0.722	0.738	0.730
300	0.947	0.674	0.787

Tabelle 10.11: Untersuchte Arbeitspunkte des hybriden Segmentierers

Dabei ist zu beachten, dass sich Precision und Recall in Tabelle 10.11 nicht auf die Gesamtperformanz des Systems, sondern ausschließlich auf die Performanz des Segmentierers alleine beziehen.

Für die drei Arbeitspunkte in Tabelle 10.11 wurden drei komplette Systemevaluationen durchgeführt. Die Resultate finden sich in Tabelle 10.12.

Segmentierer (Grenze)	Spracherkenner	PRC	RCL	F
modellbasiert	I63bbi	0.78	0.34	0.48
hybrid (20)	I63bbi	0.68	0.44	0.54
hybrid (60)	I63bbi	0.58	0.48	0.53
hybrid (300)	I63bbi	0.57	0.54	0.55

Tabelle 10.12: Gesamtergebnisse bei unterschiedlichen Arbeitspunkten des Segmentierers

Die Gesamtperformanz - ausgedrückt durch das F-Maß - wird durch den hybriden Segmentierer deutlich gesteigert (von 0.48 auf 0.55). Dies wird vor allem durch eine deutliche Steigerung der Findequote (Recall) von 0.34 auf 0.54 verursacht. Allerdings fällt gleichzeitig die Präzision der gefundenen Zeitabschnitte von 0.78 auf 0.57.

Insgesamt scheint die Gesamtperformanz um so besser zu sein, je größer das F-Maß des Segmentierers ist. Dieses Ergebnis rechtfertigt im Nachhinein die Verwendung des F-Maßes als Indikator für die Güte eines Segmentierers.

10.7 Zusammenfassung

Da es sehr schwierig ist, die Leistung zweier Systeme miteinander zu vergleichen, die jeweils durch zwei Kennzahlen (Precision und Recall) charakterisiert werden, wurde das gewichtete geometrische Mittel dieser beiden Kennzahlen, das sog. F-Maß, zur Evaluierung herangezogen. Das optimale F-Maß eines perfekten Systems ist 1.0, das schlechtestmögliche F-Maß ist 0.

Das View4You-System erzielt bei Verwendung des Spracherkenners I63bbi (mit einer Wortfehlerrate von 22,7%) einen maximalen Wert von F=0,55.

Die einzelnen Ursachen für die nicht optimale Leistung sind in Tabelle 10.13 aufgeführt.

Die größte Fehlerquelle ist das Informationssystem, gefolgt von der Segmentierungsstrategie. Um das Informationssystem signifikant zu verbessern, wäre eine Analyse der Semantik sowohl der Anfrage als auch des untersuchten Segments erforderlich. Bei der Segmentierungsstrategie liegt der Fall ähnlich. Hier könnte durch eine semantische Analyse versucht werden, akustische Segmente, die zu ein und demselben Thema gehören, miteinander zu verschmelzen. Alternativ könnte auch eine Ähnlichkeitsklassifikation benachbarter Segmente durch die *information retrieval* Komponente des

Grund	ΔF	F-Maß
perfektes System	-	1.00
Informationssystem anstelle menschlicher Auswertung	-0.25	0.75
Segmentierung in akustische Segmente anstelle von themenbasierten Segmenten	-0.15	0.60
Spracherkenner I63 statt Transkripten	-0.04	0.56
hybrider Segmentierer statt Handsegmentierung	-0.01	0.55
reales System	-	0.55

Tabelle 10.13: Gründe für die nicht perfekte Systemleistung

Systems selber erfolgen, wobei dann benachbarte ähnliche Segmente verschmolzen werden würden.

Knapp 90% der Fehler, die das System macht, lassen sich auf eine dieser beiden Ursachen zurückführen. Demgegenüber spielen Fehler des Spracherkenners und des Segmentierers keine große Rolle mehr.

10.8 Diskussion

Die Ergebnisse der Evaluation erscheinen zunächst überraschend. *A-priori* würden die meisten Beobachter den Fehlern des Spracherkenners einen hohen Einfluß auf die Gesamtperformanz zuordnen. Es zeigt sich jedoch, daß eine Fehlerrate von knapp 25% noch keinen starken Einbruch der Performanz nach sich zieht. Dieses Ergebnis wird auch durch jüngste Resultate der Text REtrieval Conference (TREC-8) gestützt, nach denen die Findeleistung der evaluierten Systeme (auf dem sog. 'spoken document retrieval track') erst ab einer Fehlerrate von etwa 30% des Spracherkenners deutlich absinkt.

Den größten Einfluß auf die Gesamtperformanz hat nicht der Spracherkenner, sondern das Informationssystem. Grundlegende Optimierungen in diesem Bereich sind allerdings vermutlich schwierig, da dieses Forschungsgebiet aufgrund des großen kommerziellen Interesses (Internet-Suchmaschinen) bereits stark bearbeitet wurde. Ein interessanter Ansatz, vor allem zur Lösung des Problems mangelnden Weltwissens, stellt *Latent Semantic Indexing* (LSI, [119]) dar. LSI komprimiert große Term-Dokument-Matrizen, die auf Hintergrundkorpora sowie dem zu indizierenden Korpus berechnet worden sind, durch Weglassen von Dimensionen in einer SVD-Zerlegung. Dadurch entstehen implizit Verknüpfungen zwischen Termen, die häufig gemeinsam in einem Dokument vorkommen. LSI kann somit als ein Verfahren zur automati-

schen Erstellung eines Thesaurus angesehen werden. Erste Ergebnisse mit LSI, die gegen Ende des Abschlusses dieser Arbeit durchgeführt wurden, verliefen erfolgversprechend.

Eine weitere Technik, die zur Verbesserung des Informationssystems eingesetzt werden könnte, ist die sog. *query expansion*. 'Query Expansion' bezeichnet eine Technik, bei der die Antworten des Systems auf eine Benutzeranfrage verwendet werden, um eine erweiterte Anfrage zu generieren. Es wird dabei von der Annahme ausgegangen, daß in den zunächst zurückgelieferten, relevanten Artikeln gehäuft auftretende Schlüsselworte relevant in Bezug auf das angefragte Thema sind. Auch 'query expansion' fällt damit in die Kategorie der Techniken, die implizit einen Thesaurus erstellen.

Die morphologische Zerlegung, die im View4You-System verwendet wird, ist nicht optimal. Durch den Einsatz eines wörterbuchbasierten Zerlegers, der z.B. auch 'stand' auf die Verbform 'stehen' zurückführen kann, könnte eine Leistungssteigerung des Gesamtsystems erzielt werden.

Der Segmentierer hat einen recht starken Einfluß auf die Gesamtperformanz. Durch den neuen hybriden Segmentierungsalgorithmus ist es zwar möglich, nahezu die Performanz von manuell gesetzten akustischen Segmentgrenzen zu erreichen. Allerdings ist dieses Ergebnis unter dem starken Vorbehalt zu sehen, daß eine Segmentierung nach akustischen Segmentgrenzen ganz deutlich schlechter ist als eine Segmentierung nach semantischen Grenzen. Es bietet sich an, die akustischen Segmente, die ja (meist) eine reine Binnensegmentierung der thematischen Segmente darstellen, geeignet zusammenzufassen. Dies könnte im einfachsten Fall dadurch erfolgen, daß der Okapi-Abstand zwischen einem Segment und seinem direkten Nachbarn berechnet und mit dem durchschnittlichen Okapi-Abstand des Segments zu allen anderen Segmenten verglichen wird. Bei Unterschreitung eines vorher bestimmten Schwellwerts würde das Segment mit seinem Nachbar verschmolzen.

Eine interessante Frage ist die nach der *ausreichenden* Segmentierungsqualität, d.h. wie gut eine Segmentierung sein müßte, um nur noch einen geringen Einfluß auf die Systemperformanz zu haben. Um diese Frage zu klären, müßten noch weitergehenden Experimente durchgeführt werden. Dazu müßte entweder ein gutes Verfahren zur Simulation eines Segmentierers entwickelt werden, oder ein es müßte ein Segmentierungsalgorithmus gefunden werden, der deutlich besser als die bisher existierenden und im Rahmen dieser Arbeit untersuchten Algorithmen funktioniert.

Kapitel 11

Zusammenfassung

Die Gegenwart ist charakterisiert von einem geradezu explosionsartigen Anstieg von Wissen. Dieses Wissen wird zwar nach wie vor überwiegend über Printmedien wie Bücher und Zeitschriften verbreitet; der Anteil von Videodaten nimmt aber ständig zu.

Damit steht das klassische Bibliothekswesen vor einer neuen Herausforderung: es müssen multimediale Datenbanken eingerichtet, betrieben und gewartet werden. Der Benutzer einer multimedialen Datenbank erwartet natürlich mindestens denselben Komfort, den er bereits von den klassischen Print-Datenbanken her kennt. Insbesondere muß eine Suche nach Stichwörtern oder Inhalten zur Verfügung stehen. Weil ausführliche Inhaltsangaben von Videofilmen nur in wenigen Fällen vorliegen, muss ein Index für solche Filmdaten mühevoll von Hand angelegt werden. Um die relevanten Inhalte zusammenzufassen, müsste sich ein Mitarbeiter der Bibliothek das entsprechende Material ansehen, was eine sehr zeitintensive Tätigkeit ist und den schnellen Aufbau von großen Videodatenbanken stark behindert. Initiativen der öffentlichen Hand, wie zum Beispiel die 'Baden-Württemberg digital library', versuchen dem Mißstand durch Erhöhung der Ressourcen abzuhelfen. Es ist jedoch absehbar, dass in Zeiten leerer Kassen der öffentlichen Hände und gleichzeitig zunehmenden Bedarfs die flächendeckende Einführung von Videodatenbanken nur sehr schleppend voranschreiten wird.

Das in dieser Arbeit vorgestellte System versucht einen Weg aufzuzeigen, um diese Lücke zu schließen. Die Grundidee ist es, einen Index vollautomatisch durch maschinelle Spracherkennung zu generieren. Dieser Index wird von einem Informationssystem (engl. *information retrieval system*) genutzt, um die Benutzeranfragen zu beantworten.

Als zu indexierende Videodaten wurden Fernsehnachrichtensendungen ausgewählt, da diese eine hohe Informationsdichte bei einer Vielfalt von verschiedenen Hintergrundgeräuschen aufweisen, und daher sowohl vom wissenschaftlichen als auch vom praktischen Gesichtspunkt aus interessant für eine Videodatenbank sind. Fernsehnachrichtensendungen weisen, wie zahlreiche andere Videodaten auch, das Pro-

blem der Binnensegmentierung auf: In einer Nachrichtensendung wird nicht nur ein einzelnes Thema, sondern eine ganze Vielfalt von Themen behandelt. Es ist daher erforderlich, die Aufnahme zu segmentieren, damit die einzelnen Themenberichte getrennt voneinander gefunden werden können.

Diese Aufgabe übernimmt der Segmentierer. In dieser Arbeit wurden verschiedene aus der Literatur bekannte Segmentierungsstrategien erstmals systematisch auf derselben Teststichprobe miteinander verglichen. Zusätzlich wurde im Rahmen dieser Arbeit ein völlig **neuer Segmentierungsansatz** entwickelt. Bei der Evaluation erzielte dieser neue Segmentierungsansatz für die meisten Arbeitspunkte mit einer Leistungszahl F von 0.78 deutlich bessere Resultate als alle bisher bekannten Algorithmen ($F = 0.58...0.70$). Der Segmentierer wird ausführlich im Kapitel 'Der Segmentierer des View4You-Systems' ab Seite 50 behandelt, der neue Segmentierungsalgorithmus findet sich ab Seite 65.

Von den US-amerikanischen ARPA-Evaluationen ist bekannt, dass mit dem heutigen Stand der Technik kontinuierliche, wohlartikulierte Sprache, wie sie etwa in Nachrichtensendungen vorkommt, mit Fehlerraten von weniger als 20% maschinell erkannt werden kann. Da für die deutsche Sprache noch kein solches Erkennersystem existiert, wurde im Rahmen dieser Arbeit ein Spracherkenner trainiert, der eine Fehlerrate von 19,5% auf ungesehenen Testdaten aufweist. Die Komponenten dieses Erkenners sind ausführlich im Kapitel 'Der Spracherkenner des View4You-Systems' ab Seite 68 dargestellt. Der Spracherkenner verwendet dabei größtenteils Methoden und Algorithmen, die in den letzten Jahren von verschiedenen Autoren und Gruppen für die Spracherkennung vorgeschlagen wurden und die einen allgemein akzeptierten Standard für Spracherkenner darstellen. Im Verlauf der Arbeit wurden zwei neue Algorithmen entwickelt, die **Vokabularadaption aus Internettexten** und die **halbüberwachte Adaption**. Bei ersterer wird die initiale Hypothese dazu verwendet, aus beliebigem, aktuellem Textmaterial aus dem Internet ein neues Vokabular und ein neues Sprachmodell zu bestimmen. Dieses Vokabular und Sprachmodell decken dann die vorliegende Sendung besser ab als das statische Vokabular und Sprachmodell. Durch diese Maßnahme sinkt die Fehlerrate um 5% relativ.

Bei der halbüberwachten Adaption werden die aktuellen Sprecher jedes Segments mit den Sprechern in der Trainingsdatenbank verglichen. Die ähnlichsten Aufnahmen aus der Trainingsdatenbank werden zur Adaption der akustischen Modelle vor dem eigentlichen Erkennungslauf verwendet. Der Algorithmus reduziert die Fehlerrate auf den Ansagesprechersegmenten um 8% relativ.

Das Informationssystem des View4You-Systems hat die Aufgabe, zu einer gegebenen Benutzeranfrage alle relevanten Segmente herauszusuchen und nach Relevanz zu sortieren. Das Informationssystem wurde innerhalb des View4You-Projektes vom Autor entwickelt. Es verwendet für die eigentliche Berechnung der Relevanzen den

Okapi-Ansatz der City University of London. Bei einer Evaluation des Informationssystems erzielte dieses ein Qualitätsmaß $F = 0,732$.

Das Informationssystem wird im Kapitel 'Datenbank und Informationssystem des View4You-Prototypen' ausführlich dargestellt. Die Evaluation findet sich im Kapitel 'Systemevaluation' ab Seite 170.

Um einen Spracherkenner zur Videoindexierung zu trainieren, sind große Mengen transkribiertes Trainingsmaterial erforderlich. Diese Sprachdaten sind aufwendig zu beschaffen und daher teuer. Ein wesentlicher Beitrag der Arbeit liegt in der Entwicklung eines **Algorithmus zum unüberwachten Training eines Spracherkenners**. Dabei erstellt sich der Spracherkenner die zum Training erforderlichen Referenzen einfach selber, indem er auf seine Ausgabe ein Konfidenzmaß anwendet. Es konnte gezeigt werden, dass unter Einsatz der Aufnahmen von etwa sieben Monaten (gut 200 Sendungen) nahezu die Performanz des auf 'normale' Weise, also mit manuell erstellten Transkriptionen trainierten, Systems erzielt werden kann (20,5% Fehlerrate).

Der Algorithmus sowie die Beschreibung der durchgeführten Experimente zum unüberwachten Lernen findet sich ab Seite 144.

Eine Systemevaluation mit dem unüberwacht trainierten System wurde durchgeführt und resultierte in einem Qualitätsmaß von $F = 0,702$. Im Vergleich zur Performanz auf Transkriptionen wird das System also nur um $\Delta F = 0,03$ (5% relativ) schlechter. Dieser Verlust kann im Vergleich zum Verlust von $\Delta F = 0,27$ durch das Informationssystem vernachlässigt werden. Dadurch konnte gezeigt werden, dass der vollständig unüberwacht trainierte Erkenner ausreichend gut funktioniert, um in einem Video-Indexierungssystem eingesetzt zu werden.

Zum unüberwachten Lernen ist eine Selbstbeurteilung des Spracherkenners, ein **Konfidenzmaß**, erforderlich. Im Abschnitt 'Konfidenzmaße' ab Seite 108 werden umfangreiche Experimente zu verschiedenen Konfidenzmaßen beschrieben. In einer Versuchsreihe wurden zahlreiche Merkmale evaluiert, die teilweise in der Literatur beschrieben und teilweise neu entwickelt worden waren. Eines der neu entwickelten Merkmale, die graphbasierte a-posteriori-Wahrscheinlichkeit **gamma**, erwies sich als mindestens genauso leistungsstark wie das beste bisher bekannte Merkmal, bei einem um einen Faktor zehn geringeren Aufwand. Es wurde nach seiner Veröffentlichung in [26] auch von anderen Forschungsgruppen übernommen und von diesen mit Erfolg eingesetzt. Neben einem Vergleich aller Merkmale enthält das Kapitel 'Konfidenzmaße' auch Ergebnisse zur Kombination der Merkmale, den Einsatz von kontextuellen Merkmalen sowie über die Eignung verschiedener Klassifikatoren. Dabei wurde gezeigt, daß durch die Kombination von Merkmalen deutlich bessere Kennzahlen erzielbar sind ist durch jedes Merkmal für sich, daß kontextuelle Merkmale die Klassifikationsleistung

konsistent erhöhen, und daß ein einfacher linearer Klassifikator zur Kombination der Merkmale nur unwesentlich schlechtere Resultate als ein mehrschichtiges neuronales Netz liefert.

Im Abschnitt 'Systemevaluation' ab Seite 170 wird das gesamte System anhand eines Fragensatzes evaluiert, der von naiven Benutzern aufgestellt wurde. Verschiedene Standardtechniken des information retrieval, sowie der Einfluß der Segmentierung auf das Gesamtergebnis, werden ausgewertet. Dabei ergibt sich, daß das Informationssystem den größten Anteil der fehlerhaften Antworten des Systems verursacht, gefolgt von Segmentierer und Spracherkenner. Weitere Optimierungen des Systems sollten daher in den Teilkomponenten Informationssystem und Segmentierer erfolgen. Einige solche Optimierungen werden ab Seite 181 diskutiert.

Anhang A

Aufbau der 'tagesschau' vom 30.03.1997

Zeit (s)	TGS-Kategorie	Inhalt	View4You Kategorie
0	-	Werbung	-
53.2	Studio	Gong und Anfangsmeldung	DISTORTED
58.1	Studio	Musik	MUSIC
65.3	Studio	Intifada in den palästinensischen Autonomiegebieten	CLEAN
96.4	Korrespondent	Intifada	DISTORTED
123.6	Korrespondent	israel. Verteidigungsminister spricht	DISTORTED
135.4	Korrespondent	Beerdigung eines Palästinensers	DISTORTED
187.6	Korrespondent	der Korrespondent kommentiert direkt	DISTORTED
204.4	Studio	Arafat beim Treffen der arabischen Liga	CLEAN
227.9	Korrespondent	Treffen der arabischen Liga	DISTORTED
302.5	Studio	Flüchtlingsdrama in der Adria	CLEAN
335.9	Korrespondent	Trauerfeier der albanischen Flüchtlinge	DISTORTED
387.0	Studio	Le Pen wiedergewählt	CLEAN
412.0	Korrespondent	Treffen der front national in Straßburg	DISTORTED
428.5	Korrespondent	Rede Le Pens	DISTORTED
441.4	NiF	Wiederwahl Le Pens	DISTORTED
462.0	Korrespondent	Gegendemonstration gegen Le Pen	DISTORTED
472.6	Korrespondent	Gegendemonstrationen gegen Le Pen	DISTORTED
488.3	Korrespondent	der Korrespondent kommentiert direkt	DISTORTED
511.3	Studio	Anschlag in Kambodscha	CLEAN
526.8	Korrespondent	Bericht aus Kambodscha	DISTORTED
615.5	Studio	Regierungskrise in Indien	CLEAN
641.7	NiF	Flüchtlingselend in Zaire	DISTORTED
675.5	Studio	Rußland unterzeichnet NATO-Abkommen	CLEAN
704.6	NiF	Ostermärsche in Wittstock	DISTORTED
731.4	Studio	Neuregelung der Sozialhilfe	CLEAN
761.8	Studio	DAG-Vorsitzender zur Arbeitslosigkeit	CLEAN
789.9	Studio	Osterbotschaften der großen Kirchen	CLEAN
824.6	Korrespondent	Papst-Segen in Rom	DISTORTED
843.1	Korrespondent	Papst-Segen live	DISTORTED
859.2	Korrespondent	Papst-Segen in Rom	DISTORTED
873.8	Korrespondent	Urbi et Orbi live (lateinisch)	MUSIC
891.4	Studio	Intendant Reinhold V oth gestorben	CLEAN
914.0	Studio	Jacques Villeneuve gewinnt großen Preis von Brasilien	CLEAN
926.1	Korrespondent	großer Preis von Brasilien	DISTORTED
988.4	Studio	Wetter-Ansage	CLEAN
994.0	Studio	Wetterbericht	CLEAN
1017.4	-	(Aufnahme endet etwas zu früh)	-

Tabelle A.1: Aufnahme vom 30.3.97

Anhang B

MPEG

MPEG steht für Motion Picture Expert Group, ein 1988 gegründetes internationales Kommitee der ISO. Das MPEG-Konsortium hat mittlerweile bereits zwei Normensystem verabschiedet (MPEG-1 und MPEG-2 genannt). Der MPEG-1 Standard wurde im November 1992 als ISO/IEC 11172 'Coding of Motion Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s' verabschiedet. Der Standard besteht aus drei Dokumenten: 11172-1 behandelt die Synchronisierung und das Multiplexen von Audio und Video, 11172-2 Video und 11172-3 Audio. MPEG-1 ist ein stark asymmetrisches Verfahren. Die Codierung ist einen Faktor 100 bis 1000 rechenaufwendiger als die Decodierung.

Schon nach kurzer Zeit stellten sich die Datenraten in MPEG-1 und auch die erreichbare Videoqualität als nicht ausreichend für Anwendungen wie digitales Fernsehen heraus. Aus diesem Grund wurde MPEG-2 definiert, und als ISO/IEC 13818 normiert. MPEG-2 existiert zusätzlich zu MPEG-1, es löst MPEG-1 nicht ab. Insbesondere werden damit keine besseren Kompressionsraten als mit MPEG-1 erzielt.

Die Leistungsmerkmale des ursprünglich geplanten MPEG-3 wurden im Laufe des Jahres 1995 in MPEG-2 übernommen, so daß als nächste Version im Oktober 1998 MPEG-4 (Version 1, ISO/IEC 14496) standardisiert wurde. MPEG-4 Version 2 ist für Ende 1999 geplant; danach wird (für Mitte 2001) MPEG-7 angestrebt. Neu in MPEG-4 sind unter anderem sehr niedrige Bitraten, QoS (Quality of Service)-Unterstützung und ein verbesserter Copyrightschutz.

Die Videocodierung von MPEG basiert auf Fouriertransformationen von 8*8 Pixel großen Blöcken und der Übertragung der niederfrequenten Bereiche der transformierten Blöcke. Die so codierten Daten werden in sogenannten I-Frames abgelegt. Das Kompressionsverfahren ist dem JPEG-Verfahren sehr ähnlich. Zusätzlich zu den I-Frames existieren die P-Frames (prediction frames), die relativ zum letzten I-Frame oder P-Frame Bewegungsvektoren enthalten und so ein Bild als Unterschied zum letzten I-Frame oder P-Frame codieren. Zwischen P-Frames sind häufig jeweils zwei B-Frames (bidirectional predicted frames) angeordnet. B-Frames dienen nicht zur Referenz für weitere B- oder P-Frames. Hingegen wächst der Fehler durch die Kom-

pression für jedes P-Frame, das auf einem vorhergehenden P-Frame aufbaut, so daß in regelmäßigen Abständen (meist 2 pro Sekunde) I-Frames eingefügt werden müssen.

D-Frames mit sehr starker Kompression sollen schnellen Vorlauf ermöglichen. CD-i verwendet unter dem Markennamen 'full motion video' das MPEG-1 Kompressionsverfahren.

Die Unterschiede zwischen MPEG-1 und MPEG-2 betreffen hauptsächlich das Videosignal und sind in Tabelle B.1 zusammengefaßt.

Parameter	MPEG-1	MPEG-2
Größe/Pixel (Standard, 50%)	PAL:360*288 / NTSC: 352*240	360*288/NTSC: 352*240 lowlevel
Größe/Pixel (Standard, 100%)	nicht möglich	720*596/NTSC: 704*480 mainlevel
Größe/Pixel (HDTV, 4:3)	nicht möglich	1440*1152 (high1440-Level)
Größe/Pixel (HDTV, 16:9)	nicht möglich	1920*1080 (high Level)
Größe (Scanvorlage)	$\leq$ 720*576 / NTSC: 704*480	$\leq$ 16384*16384 pixel
Farbformat (YC_bC_r)	4:2:0 (Scan 4:2:2)	4:2:2 (Scan bis 4:4:4)
Zahl Makroblöcke / Bild	$\leq$ 396	levelabhängig
Zahl Makroblöcke / Sekunde	$\leq$ 396*25 (NTSC: 330*30)	levelabhängig
Bildwiederholrate	$\leq$ 30 pps	$\leq$ 30 pps
max. Bit-Rate	$\leq$ 1.862 MBit/s	2-15 MBit/s
Interlace-Unterstützung	nein	ja
Audio-Verfahren	Standard (2-Kanal)	Surround-Sound

Tabelle B.1: Unterschiede zwischen MPEG-1 und MPEG-2

MPEG-1 Audio

MPEG-1 Audio erreicht Kompressionsraten bis zu 1:22 ohne größere hörbare Verluste. Üblich sind Werte von 1:6 oder 1:7. Es werden 3 Abtastraten unterstützt: 44.1, 48 und 32 kHz. Es existieren 3 Level von MPEG-1 Audio. Level I verwendet den einfachsten Algorithmus, bietet jedoch auch eine schlechtere Qualität als Level 3. In jedem Fall wird eine Frequenzanalyse der Audiodaten durchgeführt. Die Spektralkomponenten werden quantisiert und so kodiert abgelegt, wobei der erlaubte Quantisierungsfehler von der Signalamplitude (auch) in den anderen Frequenzbändern abhängt. Der erlaubte Quantisierungsfehler wird über psychoakustische Regeln vorgegeben. In Layer 1 wird ein vereinfachtes Musicam-Codierschema verwendet [1]. In Layer 2 wird das Original-Musicam Schema verwendet. Layer 3 verwendet eine Kombination von zwei Kodierschemata, Musicam und ASPEC (adaptive spectral perceptual entropy coding).

Die Audiodaten bestehen aus einer Folge von frames, die eine feste Anzahl von Abtastwerten enthalten. Bei Level 1 sind das 384, bei Level 2 und 3 1152 Abtastwerte.

[1] Musicam steht für Masking-pattern adapted Universal Subband Integrated Coding and Multiplexing, entwickelt vom CCETT, Philips, und dem IRT in München

Sie werden aus Effizienzgründen nicht byteweise, sondern als Bitstrom ohne Füllbits abgespeichert. Die Bitrate kann zwischen 32 kbit/s und maximal 448 kbit/s liegen, wobei die Bitrate im Fall von Audio Layer 3 in jedem Frame unterschiedlich sein kann. Es werden drei unterschiedliche Abtastraten (44.1 kHz, 48 kHz und 32 kHz) unterstützt.

MPEG-2 Audio

Die Audiokompression von MPEG-2 ist im wesentlichen gleich zu der von MPEG-1. Es wurden Änderungen in zwei Richtungen durchgeführt: die Qualität bei sehr niedrigen Bitraten wurde verbessert, und es wurden Vielkanal-Stereoaufnahmen eingeführt ('Dolby-Surround'). Die Qualität bei niedrigen Bitraten wurde durch die Einführung von zusätzlichen niedrigen Abtastraten verbessert. Namentlich sind das die halben Bitraten von MPEG-1, also 22.05, 24 und 16 kHz.

Anhang C

Worthypothesengraphen

Der folgende Worthypothesengraph wurde auf der Anfangsmeldung einer **tagesschau** errechnet. Die Referenz lautet 'Hier ist das Erste Deutsche Fernsehen mit der Tagesschau'.

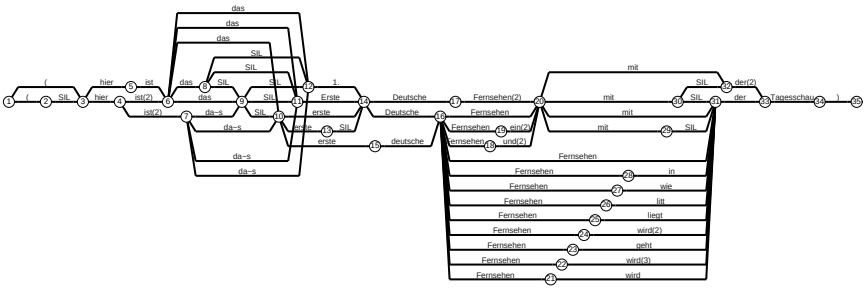


Abbildung C.1: Worthypothesengraph

Anhang D

Phonemklassen

Es ist das Ziel der entscheidungsbaumbasierten Ballung von Allophonen, auch für nicht beobachtete Phonemkontexte ein möglichst gut geeignetes Modell zu prädizieren. Dies wird durch die Verwendung von linguistisch motivierten Äquivalenzklassen ermöglicht, die als Fragen für einen binären Entscheidungsbaum verwendet werden. Die Generalisierung wird dadurch gewährleistet, daß ein ungesehener Nachbar so behandelt wird wie ein gesehener Nachbar, der mit dem ungesehenen Nachbar in ein und derselben Äquivalenzklasse ist.

Die Äquivalenzklassenliste, die im View4You-System verwendet wurde, wurden vom Autor aus einer englischen Liste [39] abgeleitet und dann von Prof. Klaus Kohler von der Universität Kiel und Frau Anja Petzold von der Universität Bonn durchgesehen und überarbeitet. Herrn Professor Kohler und Frau Petzold sei an dieser Stelle noch einmal ausdrücklich gedankt.

Die grundlegendste Einteilung ist die in Vokale und Konsonanten.

```
VOKALE:
a E: a: E @ e e: 6 I i i: O 9 2 2: a~ a~: o: U Y y y: u u: aI aU
```

Vokale

Die Gruppe der Vokale läßt sich in gelängte und kurze Vokale aufspalten:

```
VOK_KURZ: a E @ e 6 I i O 9 2 a~ U Y y u aI aU

VOK_LANG: E: a: e: i: 2: a~: o: y: u:
```

Die unterschiedlichen Vokale werden durch unterschiedliche Lippen- und Zungenstellungen realisiert. Bei den Lippenstellungen unterscheidet man zwischen gerundeten ('Kußmund') und ungerundeten Lippen.

```
RUND: O 9 2 2: o o: U Y y y: u u: OY
```

Die Gruppe UNRUND enthält dementsprechend alle Vokale, die nicht in RUND enthalten sind. Bei der Zungenstellung sind zwei Kriterien maßgeblich:

- die Stelle im Mund, an der die Zunge ihren höchsten Punkt erreicht
- wie hoch der höchste Punkt ist, d.h. wie eng der Luftspalt wird

Bei beiden Parametern (Höhe der Zunge an ihrem höchsten Punkt im Mund und Stellung des höchsten Punktes) unterscheidet man drei Klassen:

```
VOK_Z_UNTEN: a: a 6 a~ a~:
VOK_Z_MITTE: E E: 2 2: @ e e: o o: 9 O
VOK_Z_OBEN:  Y U I y y: i i: u u:

VOK_Z_VORNE: i: e e: y y: I Y E: E 2 2: 9
VOK_Z_MITTE: @ 6 a a: a~ a~:
VOK_Z_HINTEN: u: U o O o:
```

Ist das Gaumensegel geöffnet, spricht man von Nasalen.

```
VOK_NASAL: a~ a~:
```

Eine weitere wichtige, stets unbetonte Gruppe sind die Schwa.

```
VOK_SCHWA: 6 @
```

Schließlich wurden im View4You-System noch die Basisvokale des Deutschen als Gruppen verwendet:

```
VOK_A: a a:
VOK_E: e e: E
VOK_I: i i: I
VOK_O: o o: O
VOK_U: u u: U
VOK_Y: y y: Y
VOK_OE: 2 2: 9
```

Konsonanten

Die große Gruppe der Konsonanten läßt sich grob aufgliedern in zwei Subklassen, die stimmhaften (Stimmbänder schwingen) und die stimmlosen (Stimmbänder schwingen nicht), wobei noch eine Untergruppe der schwach stimmhaften definiert werden kann:

```
KONS_STIMMHAFT: m n N l r
KONS_STIMMLOS: p t k f s Q S C x h
KONS_PARTVOICED: b C x d f g Z h j k p s S t v z
```

Eine etwas andere Aufteilung anhand der Stimmhaftigkeit ist durch die sog. Fortes und Lenes gegeben:

```
KONS_FORTES:  p t k f s Q S C x
KONS_LENES:   b d g v z Z j
```

Die stimmhaften Konsonanten werden weiter in Nasale, Liquide, Laterale und Vibranten aufgeteilt:

```
KONS_NASAL: m n N
LIQUIDE: l r
LATERALE: l
VIBRANTEN: r
```

Die stimmlosen Konsonanten teilen sich in Verschlußlaute (oder Plosive, engl. 'stops'), Reibelaute (Frikative) und Kombinationen aus Verschluß- und Reibelauten (Affrikate) auf. Bei den Frikativen reibt sich der Luftstrom an einer Engstelle, beim Plosiv gibt es einen vollständigen Verschluss, der dann geräuschartig gesprengt wird. Bei stimmhaften Frikativen und Plosiven schwingen die Stimmbänder durchaus, wenn auch nicht unbedingt während des gesamten Artikulationsprozesses.

```
PLOSIVE: p b t d k g
FRIKATIVE: f v s z S Z C j x r h
AFFRIKATE: pf ts tS
PLOSIV_STIMMLOS: p t k
FRIK_STIMMLOS: C S f h s x
```

Auch die Konsonanten können nach Artikulationsort gruppiert werden. Dies kann entweder getrennt für die Gruppen Plosive, Frikative und stimmhafte Konsonanten erfolgen, oder aber gemeinsam für alle Konsonanten. Beide Gruppenaufteilungen werden im View4You-Spracherkenner vorgesehen. Die verschiedenen Artikulationsorte sind, sortiert von vorne nach hinten:

- *labial*, die Lippen gegeneinander
- *labiodental*, Unterlippe und obere Schneidezähne
- *dental-alveolar*, Zungenspitze gegen obere Schneidezähne oder Alveolen
- *palatal-alveolar*, Zungenspitze gegen Gaumen oder Alveolen
- *palatal*, Zungenspitze gegen Gaumen
- *velar*, Zungenspitze gegen den weichen Hintergaumen
- *uvular*, Zäpfchenlaut

- *glottal*, Stimmritzenlaute

Nimmt man die Gruppeneinteilung *aller* Konsonanten nach Erzeugungsort in der dargestellten Granularität vor, so entstehen die folgenden Gruppen:

```
LABIAL: p b m
LAB_DENTAL: f v
DENTAL_ALV: t d n l r s z ts
PALAT_ALV: S Z tS
PALATAL: C j
VELAR: k g x N
GLOTTAL: h Q
```

Im View4You-System sind drei weitere Gruppeneinteilungen mit geringerer Granularität vorhanden. Sie lauten

```
CONS_3VORNE: LABIAL LAB_DENTAL
CONS_3MITTE: DENTAL_ALV
CONS_3HINTEN: VELAR PALATAL PALAT_ALV
```

und

```
CONS_2VORNE: LABIAL LAB_DENTAL DENTAL_ALV
CONS_2HINTEN: PALAT_ALV PALATAL VELAR GLOTTAL
```

und

```
CONS_2MITTE: DENTAL_ALV PALAT_ALV
CONS_2AUSSEN: LABIAL LAB_DENTAL PALATAL VELAR GLOTTAL
```

Man kann auch die Plosive und die Frikative getrennt nach Artikulationsort aufteilen. Diese Aufteilungen bringen nichts grundlegend Neues und seien hier der Vollständigkeit halber wiedergegeben.

```
PLOSIV_VORNE: b p
PLOSIV_MITTE: d t
PLOSIV_HINTEN: g k
```

```
FRIK_LAB_DENTAL: f v
FRIK_DENTAL_ALVEOLAR: s z
FRIK_PALAT_ALV: S Z
FRIK_PALATAL: C j
FRIK_VELAR: x r
FRIK_GLOTTAL: h
FRIK_3_VORNE: FRIK_LAB_DENTAL
```

```
FRIK_3_MITTE: FRIK_DENTAL_ALVEOLAR
FRIK_3_HINTEN: FRIK_PALAT_ALV FRIK_PALATAL FRIK_VELAR FRIK_GLOTTA
FRIK_2_VORNE: FRIK_LAB_DENTAL FRIK_DENTAL_ALV FRIK_PALAT_ALV
FRIK_2_HINTEN: FRIK_PALATAL FRIK_VELAR FRIK_GLOTTAL
```

Weitere Gruppen, die sich nicht in die vorgestellten Ordnungsschemata einfügen lassen, sind:

```
FRIK_SIBILANTEN: s z S Z
STIMMHAFT: KONS_STIMMHAFT VOKALE
NASAL: KONS_NASAL VOK_NASAL
APICAL: y n d s t
LAX: O @ I Y E 9:
SILBISCH: n m l N
UNFORTLENES: KONS_STIMMHAFT h
Z_VORNE: CONS_3VORNE VOK_Z_VORNE
Z_MITTE: CONS_3MITTE VOK_Z_MITTE
Z_HINTEN: CONS_3HINTEN VOK_Z_HINTEN
```

Durch die Aufnahme der Diphthonge in den Basis-Phonemsatz entsteht das Problem, daß die Konstituenten des Diphthongs zu unterschiedlichen Gruppen gehören können. Unter der Annahme, daß nachfolgende Phoneme durch den zweiten Laut des Diphthongs, vorhergehende Phoneme hingegen durch den ersten Laut des Diphthongs beeinflußt werden, kann dieses Problem durch *richtungsabhängige* Gruppen gelöst werden. Ein Diphthong wird in verschiedene Gruppen eingeordnet, je nachdem, ob es links oder rechts vom gerade betrachteten Phonem steht. Durch diese Richtungsabhängigkeit erhöht sich die Anzahl der Gruppen um weitere 18.

```
R-CONS_2AUSSEN C N g h j k x
R-PLOSIVE b d g k p t ts
R-VOK_A a a: aI aU
R-VOK_O OY o:
R-VOK_Z_HINTEN OY o o: u:
R-VOK_Z_LOW ER2 a a: aI aU
R-VOK_Z_MID 2 2: @ E E: OY e e: o o:
R-VOK_Z_MITTE 6 @ a a: aI aU
L-FRIKATIVE C S f h j r s ts v x z
L-FRIK_DENTAL_ALV s ts z
L-FRIK_SIBILANTEN S s ts z
L-VOK_I AI i:
L-VOK_RUND 2 2: OY aU o o: u u: y y:
L-VOK_U aU u:
L-VOK_UNRUND 6 @ E E: a a: aI e e: i:
```

```
L-VOK_Z_HIGH OY aI aU i i: u u: y y:
L-VOK_Z_HINTEN aU o o: u:
L-VOK_Z_VORNE 2 2: E E: OY aI e e: i: y y:
```

Anhang E

Glossar

Allophone

'Allophon' ist die Bezeichnung für eine bestimmte Kontextausprägung eines Phonems. Man spricht z.B. vom 'links-d rechts-s Allophon des Phonems a', wenn das 'a' in 'das' gemeint ist.

Frames

Ein *frame* ist die kleinste zeitliche Einheit in der maschinellen Spracherkennung. Er bezeichnet keine genormte Länge in Sekunden, sondern ist abhängig von der Implementation. In dieser Arbeit entspricht ein *frame* 10 ms (0.01 Sekunden).

Ganzwortmodelle

Für die Spracherkennung können verschiedene lange Basiseinheiten der Modellierung verwendet werden. Die längste noch einigermaßen gebräuchliche Basiseinheit ist das Wort, die dafür erforderlichen Modelle bezeichnet man als *Ganzwortmodelle*. Bei ihrer Verwendung hat jedes Wort im Vokabular des Erkenners ein eigenes Modell. Der Vorteil von Ganzwortmodellen ist, daß sie die wortabhängige Realisierung aller ihrer Phoneme sowie deren Koartikulation implizit modellieren. Ihr Nachteil liegt darin, daß zur robusten Schätzung der Modellparameter eines Ganzwortmodells eine große Zahl von Trainingsbeispielen erforderlich ist, die normalerweise für seltene Worte nicht erreicht werden kann. Ganzwortmodelle werden daher beinahe ausschließlich in Erkennern mit sehr kleinen Wortschätzen eingesetzt.

Phoneme

Ein *Phonem* ist die kleinste für eine gegebene Sprache bedeutungstragende akustische Einheit. Eine akustische Einheit ist dann bedeutungstragend, wenn es zwei Worte A und B gibt, die sich nur in diesem einen Laut unterscheiden. Ein Beispiel für ein sog. Minimalpaar wären *Hut* und *gut*, die sich nur im ersten Laut unterscheiden. An diesem Beispiel erkennt man, daß 'h' und 'g' Phoneme sein müssen.
Diese Definition des Phonems deckt sich gut mit der intuitiven Vorstellung, daß Phoneme kurze, voneinander unterscheidbare Laute seien. Im Deutschen gibt es allerdings für den velaren Frikativ 'ch' (wie in *Bach*) und dem palatalen Frikativ 'ch' (wie in *ich*) kein Wortpaar, das sich nur in diesem einen Laut unterscheidet. Die beiden Laute werden daher der Definition entsprechend nicht als unterschiedliche Phoneme behandelt. In dieser Arbeit werden, abweichend von der Definition, das velare 'ch' und das palatale 'ch' als eigenständige, unterscheidbare Phoneme behandelt.
Die phonetisch-linguistische Theorie bezeichnet akustisch unterscheidbare Laute, die keine Phoneme sind, als *Phone* (Einzahl: Phon).

Monophone

Ein *Monophon* entspricht einem Phonem. Gemeint ist damit eine bestimmte Art der akustischen Modellierung, die jedem Phonem ein einziges akustisches Modell zuweist. Monophon-Modelle können durch die Vorgänger- und Nachfolgerphoneme verursachte Variationen in der Aussprache eines Phonems nicht modellieren und erreichen daher nur vergleichsweise niedrige Erkennungsleistungen. Allerdings lassen sich Monophonmodelle sehr gut statistisch schätzen, weil sich die ganze Datenbasis auf wenige (für die meisten Sprachen deutlich unter 100) Monophone verteilt, so daß jedes einzelne mit sehr vielen Daten robust estimiert werden kann. Daher werden Monophonmodelle oft zum Glätten von feineren Modellen herangezogen.

Triphone

Unter Triphonen [122] versteht man eine bestimmte Art der akustischen Modellierung, die jedem Phonem für jede Kombination von seinen Vorgänger- und Nachfolgerphonemen jeweils ein eigenes akustisches Modell zuweist. Triphone erlauben eine sehr differenzierte Modellierung. Durch ihre hohe Zahl (bei 50 Basisphonemen schon $50^3 = 125000$) ist eine robuste Schätzung ihrer Parameter aber meist kaum möglich. In vielen Fällen werden daher Gruppen von Triphonen gebildet, die gemeinsam ein akustisches Modell haben und einen Kompromiß zwischen Feinheit der Modellierung und Robustheit der Parameterschätzung darstellen.

Precision (PRC) und Recall (RCL)

Versucht man, aus einer großen Menge von Entitäten diejenigen herauszufinden, die eine besondere Eigenschaft haben - zum Beispiel Birnen in einem Sack voll Äpfel - dann benötigt man zwei Kennzahlen, um die Qualität der Suche zu beurteilen. Diese beiden Kennzahlen sind einerseits die *Findequote, engl. recall oder RCL*, also der Prozentanteil der Birnen, die gefunden wurden, und andererseits die *Findepräzision, engl. precision oder PRC*, also der Prozentanteil von Birnen aus allen gefundenen Früchten - es ist ja durchaus möglich, daß eine als Birne gefundene Frucht in Wirklichkeit ein Apfel ist und ein Fehler des Klassifikators Birne/Apfel vorliegt. Würde man z.B. nur die Findequote angeben, könnte diese leicht auf 100% dadurch gebracht werden, daß man alle Früchte im Sack als Birnen findet - und schon hat man 100% der tatsächlich vorhandenen Birnen, egal wieviele das sind, gefunden. Andererseits kann man die Findepräzision dadurch optimieren, daß man gar keine Frucht als Birne klassifiziert. In diesem Fall sind 100% der gefundenen Früchte Birnen, aber die Findequote ist Null.

Precision und Recall sind eng mit Typ-I und Typ-II-Fehlern verknüpft (siehe dort).

ROC curve; Typ-I und Typ-II-Fehler

Versucht man, aus einer großen Menge von Entitäten einige herauszufinden, die eine besondere Eigenschaft haben - ein Beispiel hierfür ist ein Wordspotter - dann kann man dabei zwei grundsätzlich verschiedene Fehlertypen machen. Typ-I-Fehler sind solche, bei denen eine Entität nicht gefunden wurde, obwohl sie - der Problemdefinition nach - hätte gefunden werden sollen. Typ-II-Fehler sind solche, bei denen eine Entität gefunden wurde, die aber die gesuchte Eigenschaft nicht aufweist. Typ-II-Fehler bezeichnet man auch als *false alarms*. Der Zusammenhang der Typ-I und Typ-II-Fehler mit den Maßen Precision und Recall (siehe dort) ist wie folgt:

$$PRC = \frac{H_{f,c}}{H} = 1.0 - \frac{Typ - II - Fehler}{H} \tag{E.1}$$

$$RCL = \frac{H_{f,c}}{N_f} = 1.0 - \frac{Typ - I - Fehler}{N_f} \tag{E.2}$$

Dabei ist N die Zahl der Entitäten insgesamt, N_f die Zahl der zu findenden Entitäten daraus, H die Zahl der gefundenen und $H_{f,c}$ die Zahl der gefundenen und auch tatsächlich gesuchten Entitäten.

Typ-I-Fehler und Typ-II-Fehler sind nicht unabhängig voneinander. Es ist zum Beispiel möglich, die Typ-I-Fehler auszuschalten, indem einfach sämtliche Entitäten als gefunden markiert werden. Umgekehrt können die Typ-II-Fehler eliminiert werden, indem keinerlei Entitäten markiert werden. Durch Einstellung einer Entscheidungsschwelle innerhalb des Wordspotters ist es möglich, den Arbeitspunkt zwischen

diesen beiden Extremen einzustellen. Auf diesen gewählten Arbeitspunkt beziehen sich dann die Angaben zur Performanz, wie Precision und Recall oder Typ-I und Typ-II-Fehlerrate. Es ist jedoch in der Regel nicht möglich, anhand dieser Kennzahlen zwei Wordspotter miteinander zu vergleichen, da die Kennzahlen sich meist auf unterschiedliche Arbeitspunkte beziehen. Eine vollständige Charakterisierung der Performanz eines Wordspotters ist daher erst durch die Angabe von Precision und Recall über der internen Schwelle, oder zweckmäßiger durch die Angabe von Precision über Recall möglich. Eine solche Charakteristik (Precision über Recall, oder häufiger Precision über der 'false alarm rate') wird als ROC (receiver operator characteristic)-Kurve bezeichnet.

Literaturverzeichnis

[1] T. Kemp, A. Waibel, *Unsupervised training of a speech recognizer: recent Experiments*, in Proc. EUROSPEECH 99, Budapest, September 1999

[2] P. Geutner, M. Finke, A. Waibel, *Selection criteria for Hypothesis Driven Lexical Adaptation*, in Proc. of the IEEE conference for acoustics, speech and signal processing (ICASSP), 15.-19. März 1999, Phoenix, AZ, USA

[3] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, *Performance measures for information extraction*, Proceedings of the DARPA Broadcast News Workshop, Hilton at Washington Dulles Airport, Herndon, VA, 28. Februar - 3. März 1999

[4] K. Ohtsuki, S. Furui, A. Iwasaki, N. Sakurai, *Message-driven speech recognition and topic word extraction*, in Proc of the IEEE conference for acoustics, speech and signal processing (ICASSP), März 1999, Phoenix, AZ, USA

[5] D. Oppermann, TU München: persönliche Kommunikation

[6] R. Singh, B. Raj, R. Stern, *Automatic clustering and generation of contextual questions for tied states in hidden markov models*, in Proc. of the IEEE conference for acoustics, speech and signal processing ICASSP), März 1999, Phoenix, AZ, USA

[7] H. Wactlar, M. Christel, Y. Gong, A. Hauptmann, *Lessons Learned from Building a Terabyte Digital Video Library*, **Computer**, Februar 1999, S. 66-73

[8] M. Westphal, A. Waibel, *Towards spontaneous speech recognition for on-board car navigation and information systems*, in Proc. EUROSPEECH 99, Budapest, September 1999

[9] A. Wendemuth, G. Rose, J. Dolfing, *Advances in confidence measures for large vocabulary*, in Proc. of the IEEE conference for acoustics, speech and signal processing ICASSP), 15.-19. März 1999, Phoenix, AZ, USA

[10] D. Willett, C. Neukirchen, J. Rotland, G. Rigoll, *Refining tree-based state clustering by means of formal concept analysis, balanced decision trees and automatically generated model-sets*, in Proc. of the IEEE conference for acoustics, speech and signal processing ICASSP), 15.-19. März 1999, Phoenix, AZ, USA

[11] P. Geutner, *Adaptive Vocabularies in Large Vocabulary Speech Recognition*, Ph. D. thesis, University of Karlsruhe, Germany, February 1999

[12] T. Kemp, A. Waibel, *Reducing the OOV rate in broadcast news speech recognition*, in Proc. of ICSLP 98, Vol 5, S. 1839 ff, Sydney, Australia, Dezember 1998

[13] T. Kemp, P. Geutner, M. Schmidt, B. Tomaz, M. Weber, M. Westphal, A. Waibel, *The Interactive Systems Labs View4You video indexing system*, in Proc. of ICSLP 98, Vol 4, S. 1639 ff, Sydney, Australia, Dezember 1998

[14] T. Kemp, A. Waibel, *Unsupervised training of a speech recognizer using TV broadcasts*, in Proc. of ICSLP 98, Vol 5, S. 2207 ff, Sydney, Australia, Dezember 1998

[15] T. Kemp, M. Weber, P. Geutner, J. Guertler, P. Scheytt, M. Schmidt, B. Tomaz, M. Westphal, A. Waibel, *Automatische Erstellung einer Video-Datenbank: das View4You-System*, in Proc. of the 4th Conference on Natural Language Processing KONVENS-98, Vol. 1, S. 347 ff., Bonn, Germany, Oktober 1998

[16] S. Chen, M.J.F. Gales, P.S. Gopalakrishnan, R.A. Gopinath, D. Kanevsky, P. Olsen, L. Polymenakos, *IBM's LVCSR system for transcription of broadcast news used in the 1997 HUB4 english evaluation*, in Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, S. 127 ff, Landsdowne Conference Resort, Landsdowne, VA, 8.-11. Februar 1998

[17] S.S. Chen, P.S. Gopalakrishnan, *Speaker, environment and channel change detection and clustering via the bayesian information criterion*, in Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, S. 127 ff, Landsdowne Conference Resort, Landsdowne, VA, 8.-11. Februar 1998

[18] T. Hain, S.E. Johnson, A. Tuerk, P.C. Woodland, S.J. Young, *Segment Generation and Clustering in the HTK Broadcast News Transcription System*, in Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, S. 133 ff, Landsdowne Conference Resort, Landsdowne, VA, 8.-11. Februar 1998

[19] National Institute of Standards (NIST), *Proceedings of the DARPA Broadcast News transcription and understanding workshop*, Lansdowne, VA, 8.-11. Februar 1998

[20] F. Wessel, K. Macherey, R. Schlüter, *Using word probabilities as confidence measures*, in Proc. of the IEEE ICASSP 98, Seattle, Washington, USA, 12.-15. Mai 1998

[21] P.C. Woodland, T. Hain, S.E. Johnson, T.R. Niesler, A. Tuerk, E.W.D. Whittaker, S. Young, *The 1997 HTK Broadcast News Transcription System*, in Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, S. 133 ff, Landsdowne Conference Resort, Landsdowne, VA, 8.-11. Februar 1998

[22] M. Woszczyna, *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*, Dissertation an der Universität Karlsruhe, Fakultät für Informatik, Februar 1998

[23] http://www-nlpir.nist.gov

[24] T. Schaaf, T. Kemp, *Confidence measures for spontaneous speech*, in Proc. ICASSP-97, Vol 2, pp 875 ff, Müchen, April 1997

[25] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, M. Westphal, *The Karlsruhe-Verbmobil Speech Recognition System*, in Proc. ICASSP-97, Vol 1, S. 83 ff, München, April 1997

[26] T. Kemp, T. Schaaf, *Estimating confidence using word lattices*, in Proc. EUROSPEECH-97, Band 2, S. 827 ff, Rhodos, Griechenland, September 1997

[27] J. Billa, K. Ma, M. Siu, G. Zavaliagkos, *Acoustic modeling work at BBN*, in Proc. of the Hub-5 Conversational Speech Recognition workshop, NIST, Linthicum Heights, Maryland, November 1997

[28] M.M. Beaulieu, M. Gatford, X. Huang, S.E. Robertson, S. Walker, P. Williams, *Okapi at TREC-5*, Proc. of the 5th Text Retrieval Conference, NIST, Gaithersburg, MD, Januar 1997

[29] J. Caminero, L. Hernandez, C. de la Torre, C. Martin, *Improving utterance verification using hierarchical confidence measures in continuous natural numbers recognition*, Proc. ICASSP 97, S. 891 ff., München, April 1997

[30] M. Finke, L. Chase, J. Fritsch, P. Geutner, K. Ries, T. Zeppenfeld, A. Waibel, *JanusRTk Switchboard/CallHome Evaluation System*, Proc. of the ARPA Hub-5E evaluation, Linthicum Heights, Maryland, Mai 1997

[31] J. G. Fiscus, *A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER)*, in Proc. of the Hub-5E conversational speech recognition workshop, ARPA, 1997

[32] M.J.F. Gales, *Semi-tied full-covariance matrices for hidden markov models*, Cambridge University Technical Report TR 287, CUED/F-INFENG/TR 287, Cambridge, England, April 1997

[33] J. Garofolo, J. Fiscus, W. Fisher, *Design and preparation of the 1996 Hub-4 broadcast news benchmark corpora*, in Proc. of the DARPA Speech Recognition Workshop held in the Westfields Conference Center, Chantilly, VA, Februar 1997

[34] L. Gillick, Y. Ito, J. Young, *A probabilistic approach to confidence estimation and evaluation*, Proc. ICASSP 97, S. 879 ff., München, April 1997

[35] H. Hild, *Buchstabiererkennung mit neuronalen Netzen in Auskunftssystemen*, Dissertation, Universität Karlsruhe, Fakultät für Informatik, Shaker Verlag, Aachen, ISBN 3-8265-3155-8, 1997

[36] National Institute of Standards (NIST), *The 1997 Hub-5NE evaluation plan for recognition of conversational speech over the telephone, in non-english languages*, in Proc. of the Hub-5 conversational speech recognition workshop, Maritime Institute of Technology (MITAGS), Linthicum Heights, Maryland, November 1997

[37] A. Manos, V. Zue, *A segment-based wordspotter using phonetic filler models*, Proc. ICASSP 97, S. 899 ff., Müchen, April 1997

[38] C. Neti, S. Roukos, E. Eide, *Word-based confidence measures as a guide for stack search in speech recognition*, Proc. ICASSP 97, S. 883 ff., München, April 1997

[39] I. Rogina, *Parameterraumoptimierung für Diktiersysteme mit unbeschränktem Vokabular*, Dissertation, Universität Karlsruhe, Fakultät für Informatik, 1997

[40] M. Siegler, U. Jain, B. Ray, R. Stern, *Automatic segmentation, classification and clustering of broadcast news audio*, Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, S. 97ff, The Westfields Conference Center, Chantilly, VA, 2.-5. Februar 1997

[41] J. Viegener, *Inkrementelle, domänenunabhängige Thesauruserstellung in dokumentbasierten Informationssystemen durch Kombination von Konstruktionsverfahren*, Ph. D. Thesis, Fakultät für Informatik der Universität Karlsruhe, 1997, ISBN 3-89601-428-5, ISSN 0941-5769, infix Verlag, Sankt Augustin, 1997

[42] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, A. Stolcke, *Neural-Network based measures of confidence for word recognition*, Proc. ICASSP 97, S. 887 ff., München, April 1997

[43] P. Zhan, M. Westphal, *Speaker normalization based on frequency warping*, in Proc. ICASSP-97, München, April 1997

[44] T.Kemp, A.Jusek: 'Modelling unknown words in spontaneous speech', in Proc. ICASSP-96, pp 530 ff, Atlanta, Mai 1996, ISBN 0-7803-3192-3

[45] T. Kemp, *Regelbasiert generierte Aussprachevarianten fuer Spontansprache*, in: Natural Language Processing and Speech Technology, Dafydd Gibbon (Ed.), Mouton de Gruyter, Berlin, 1996, ISBN 3-11-015449-8

[46] M. Woszczyna, M.Finke, D.Gates, M.Gavalda, T.Kemp, A.Lavie, A.McNair, L.Mayfield, M.Maier, I.Rogina, K.Shima, T.Sloboda, A.Waibel, P.Zhan, T.Zeppenfeld, *Janus II - advances in spontaneous speech translation*, in Proc. ICASSP-96, pp 409 ff, Atlanta, Mai 1996, ISBN 0-7803-3192-3

[47] S. Cox, R. Rose: *Confidence Measures for the Switchboard Database*, in Proc. ICASSP-96, pp 511 ff, Atlanta, Georgia, Mai 1996

[48] E. Eide, H. Gish: *A parametric approach to vocal tract length normalization*, in Proc. ICASSP-96, pp 346 ff, Atlanta, Georgia, Mai 1996

[49] M.J.F. Gales, P.C. Woodland, *Mean and variance adaptation within the MLLR framework*, Computer Speech and Language **10** (1996), pp 249-264

[50] Haitao Qiu: *Confidence Measure for Speech Recognition Systems*, Masters Thesis, Carnegie Mellon University Computational Linguistics Philosophy Department, Pittsburgh, PA, April 1996

[51] E. Lleida, R.C. Rose: *Efficient decoding and training procedures for utterance verification in continuous speech recognition*, in Proc. ICASSP-96, pp 507 ff, Atlanta, Georgia, Mai 1996

[52] M. Finke, T. Zeppenfeld, M. Maier, L. Mayfield, K. Ries, P. Zhan, J. Lafferty, A. Waibel: *Switchboard April 1996 Evaluation Report*, DARPA, April 1996

[53] Summary of the NIST Workshop on confidence measures, NIST, Januar 1996

[54] P. Placeway, J. Lafferty: *Cheating with imperfect transcripts*, in Proc. ICSLP 96, Philadelphia, September 1996

[55] Z. Rivlin, M. Cohen, V. Abrash, T. Chung, *A phone-dependent confidence measure for utterance rejection*, in Proc. ICASSP-96, S. 515 ff, Atlanta, Georgia, Mai 1996

[56] R. Sukkar, A. Setlur, M. Rahim, C.H. Lee, *Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training*, in Proc. ICASSP 96, S. 518 ff, Atlanta, Georgia, Mai 1996

[57] H. Wactlar, A. Hauptmann, M. Witbrock, *Informedia: news-on-demand experiments in speech recognition*, Proc. of the ARPA SLT workshop, 1996.

[58] P.C. Woodland, M.J.F. Gales, D. Pye, *Improving environmental robustness in large vocabulary speech recognition*, in Proc. ICASSP 96, S. 65 ff, Atlanta, Georgia, Mai 1996

[59] T. Kemp, *Data-driven codebook adaptation in phonetically tied SCHMMs*, in Proc. ICASSP-95, vol 1, pp 477ff, Detroit, Mai 1995

[60] B. Suhm, P. Geutner, T. Kemp, A. Lavie, L. Mayfield, A.E.McNair, I. Rogina, T. Sloboda, W. Ward, M. Woszczyna, A. Waibel. *Janus: towards multilingual spoken language translation*, in Proc. of the ARPA Spoken Language Systems Technology Workshop held at the Barton Creek Resort Conference Center, Austin, TX, 22.-25. Januar 1995; Morgan Kaufman Publishers Inc., 340 Pine Street, 6th floor, San Francisco, CA 94104, ISBN 1-55860-374-3

[61] P.Geutner, B.Suhm, F.D.Buo, T.Kemp, L.Mayfield, A.E.McNair, I.Rogina, T.Schultz, T.Sloboda, W.Ward, M.Woszczyna, A.Waibel. *Integrating different learning approaches into a multilingual spoken language translation system*, in Proc. of the IJCAI workshop on New Approaches to Learning for Natural Language Processing, pp 33 ff, Montreal, Canada, August 1995

[62] Simon Arnfield, Peter Roach, Jane Setter, Peter Greasley, Dave Horton, *Emotional Stress and Speech Tempo Variation*, Proceedings of the ESCA-NATO Tutorial and Research Workshop on Speech under Stress, Lisbon, Portugal, September 1995, ISSN 1018-4554

[63] A. Anastasakos, R. Schwartz, H. Shu, *Duration modeling in large vocabulary speech recognition*, Proc. ICASSP 95, pp 628 ff., Detroit, Mai 1995

[64] P. Jeanrenaud, E. Eide, U. Chaudhari, J. McDonough, K. Ng, M. Siu, H. Gish, *Reducing word error rate on conversational speech from the switchboard corpus*, Proc. ICASSP 95, S. 53 ff., Detroit, Mai 1995

[65] E. Eide, H. Gish, P. Jeanrenaud, A. Mielke: *Understanding and improving speech recognition performance through the use of diagnostic tools*, in Proc. ICASSP-95, S. 221 ff., Detroit, Mai 1995

[66] P. Fetter, F. Class, U. Haiber, A. Kaltenmeier, U. Kilian, P. Regel-Brietzmann, *Detection of unknown words in spontaneous speech*, in Proc. EUROSPEECH 95, vol. 3, S. 1637 ff, Madrid, September 1995

[67] R. Haeb-Umbach, P. Beyerlein, E. Thelen, *Automatic transcription of unknown words in a speech recognition system*, in Proc. ICASSP 95, S. 840 ff., Madrid, September 1995

[68] Michael D. Monkowski, Michael A. Picheny, P. Srinivasa Rao, *Context dependent phonetic duration models for decoding conversational speech*, in Proc. ICASSP 95, vol. 1, S. 528 ff., Detroit, Mai 1995

[69] Nikki Mirghafori, Eric Fosler, Nelson Morgan, *Fast speakers in large vocabulary continuous speech recognition: analysis and antidotes*, Proc. EUROSPEECH 95, vol. 1, S. 491 ff., Madrid, September 1995

[70] Douglas O'Shaughnessy, *Timing patterns in fluent and disfluent spontaneous speech*, Proc. ICASSP 95, S. 600 ff., Detroit, Mai 1995

[71] P. Dumouchel, D. O'Shaughnessy, *Segmental duration and HMM modeling*, Proc. EUROSPEECH 95, vol. 1, S. 803 ff., Madrid, September 1995

[72] P. Jeanrenaud, M. Siu, H. Gish, *Large vocabulary word scoring as a basis for transcription generation*, Proc. EUROSPEECH 95, S. 2149 ff., Madrid, September 1995

[73] R. Kneser, H. Ney, *Improved backing-off for m-gram language modeling*, Proc. ICASSP 95, vol 1, pp 181 ff, Detroit, Mai 1995

[74] C.J. Legetter, P.C. Woodland, *Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models*, Computer Speech and Language **9** (1995), pp 171 ff

[75] E. Paulus, M. Lehning, *Die Evaluierung von Spracherkennungssystemen in Deutschland*, Verbmobil Report 70/95, Juli 1995; auch im Tagungsband *Fortschritte der Akustik*, DAGA 1994, Dresden 1994, S. 147 ff.

[76] Yukihiro Osaka, Shozo Makino, Toshio Sone, *Spoken Word recognition using phoneme duration information estimated from speaking rate of input speech*, Proc. ICSLP 95, S. 191 ff.

[77] M. Rahim, C.H. Lee, B.H. Juang, *Robust utterance verification for connected digits recognition*, Proc. ICASSP-95, S. 285 ff, Detroit, Mai 1995

[78] R.C. Rose, B.H.Juang, C.H. Lee: *A training procedure for verifying string hypotheses in continuous speech recognition*, Proc. ICASSP-95, S. 281 ff, Detroit, Mai 1995

[79] J. Savoy, M. Ndarugendamwo, D. Vrajitoru, *Report on the TREC-4 Experiment: Combining Probabilistic and Vector-Space Schemes*, Proc. of the 4th Text Retrieval Conference, NIST, Gaithersburg, MD, November 1995

[80] E.G. Schukat-Talamazzini, *Automatische Spracherkennung*. Vieweg Verlag, Braunschweig, 1995.

[81] Matthew A. Siegler, Richard M. Stern, *On the effects of speech rate in large vocabulary speech recognition systems*, Proc. ICASSP 95, S. 612 ff., IEEE, Detroit, Mai 1995

[82] T. Sloboda, *Dictionary learning: performance through consistency*, Proc. ICASSP 95, pp 453 ff., IEEE, Detroit, Mai 1995

[83] David Burshtein, *Robust parametric modeling of durations in HMMs*, in Proc. ICASSP 95, vol. 1, S. 548 ff., IEEE, Detroit, Mai 1995

[84] M. Weintraub: *LVCSR Log-Likelihood ratio scoring for keyword spotting*, in Proc. ICASSP 95, vol. 1, S. 297 ff., IEEE, Detroit, Mai 1995

[85] R. Wilkinson, J. Zobel, R. Sacks-Davis, *Similarity Measures for Short Queries*, Proc. of the 4th Text Retrieval Conference, NIST, Gaithersburg, MD, November 1995

[86] M. Woszczyna, N.Aoki-Waibel, F.D.Buo, N. Coccaro, K. Horigushi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C.P. Rose, T. Schultz, B. Suhm, M. Tomita, A. Waibel, *Janus 93: Towards Spontaneous Speech Translation*, Proc. ICASSP-94, S. 345-349, IEEE, Adelaide, April 1994

[87] H. Bourlard, B. D'hoore, J.M. Boite: *Optimizing recognition and rejection performance in wordspotting systems*, Proc. ICASSP-94, S. I-373 ff., IEEE, Adelaide, April 1994

[88] Gudrun Flach, *Beschreibung von Aussprachevarianten*, Tagungsband 'Elektronische Sprachsignalverarbeitung', 1994

[89] K. Kohler, M. Pätzold, A. Simpson, *Handbuch zur Segmentation und Etikettierung von Spontansprache 2.3*, IPDS Kiel, Verbmobil Technisches Dokument Nr. 16, Dezember 1994

[90] D.S.Pallett et al, *Be Sure to Read the Fine Print: II*, Proc. ARPA Spoken Language Systems and Technology Workshop, ARPA, März 1994

[91] T. Robinson, *SHORTEN: simple lossless and near-lossless waveform compression*, Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR.156, Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK, Dezember 1994

[92] Sheryl Young, *Detecting misrecognitions and out-of-vocabulary words*, in Proc. ICASSP-94, S. II-21 ff., Adelaide, Australia, April 1994

[93] R. Sukkar, *Rejection for connected digit recognition based on GPD segmental discrimination*, in Proc. ICASSP-94, S. I-393 ff., Adelaide, Australia, April 1994

[94] T. Otsuki, A. Ito, S. Makino, T. Otomo, *The performance prediction method on sentence recogntion system using a finite state automaton*, in Proc. ICASSP-94, S. I-397 ff., Adelaide, Australia, April 1994

[95] M. Jones, P.C. Woodland, *Using relative duration in large vocabulary speech recognition*, in Proc. EUROSPEECH 93, S. I-311 ff.

[96] M. Jones, P.C.Woodland, *Exploiting variable width features in large vocabulary speech recognition*, Proc. ICASSP 93, Vol. 2, S. 323 ff.

[97] Michael M. Hochberg, Harvey F. Silverman, *Constraining model duration variance in HMM-based connected-speech recognition*, Proc. EUROSPEECH 93, S. 323 ff.

[98] Yifan Gong, William C. Treurniet, *Duration of phones as function of utterance length and its use in automatic speech recognition*, Proc. EUROSPEECH 93, S. 315 ff.

[99] Mei-Yuh Hwang, *Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition*, Ph.D. thesis, CMU-CS-93-230, Carnegie Mellon University, Pittsburgh, PA 15213, Dezember 1993

[100] P. Jeanrenaud, K. Ng, M. Siu, J. Rohlicek, H. Gish, *Phonetic-based word spotter: Various configurations and application to event-spotting*, in Proc. EUROSPEECH 1993, S. 1057 ff., ESCA, Berlin, 1993

[101] Nelly Suaudeau, Regine Andre-Obrecht, *Sound duration modelling and time-variable speaking rate in a speech recognition system*, Proc. EUROSPEECH 93, S. 307 ff.

[102] E. Lleida, J. Marino, J. Salavedra, A. Bonafonte, E. Monte, A. Martinez, *Out-of-vocabulary word modelling and rejection for keyword spotting*, in Proc. EUROSPEECH 1993, S. 1265 ff., ESCA, Berlin 1993

[103] M. Woszczyna, N. Coccaro, A. Eisele, A. Lavie, A. McNair, T. Polzin, I. Rogina, C.P. Rose, T. Sloboda, M. Tomita, J. Tsutsumi, N. Aoki-Waibel, A. Waibel, W. Ward, *Recent Advances in Janus, a Speech-to-Speech Translation System*, Proc. EUROSPEECH 1993, S. 1295-1298

[104] J. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, M. Siu, *Phonetic Training and language modeling for word spotting*, Proc. ICASSP 93, S. II-459 ff., IEEE Signal Processing Society, Minneapolis, April 1993

[105] S.R. Young, W. Ward: *Recognition confidence measures for spontaneous spoken dialog*, in Proc. EUROSPEECH 93, S. 1177 ff, Berlin, September 1993

[106] R. Sukkar, J. Wilpon: *A two pass classifier for utterance rejection in keyword spotting*, Proc. ICASSP 1993, S. II-451 ff., IEEE Signal Processing Society, Minneapolis, April 1993

[107] L. Villarrubia, A. Acero: *Rejection techniques for digit recognition in telecommunication applications*, Proc. ICASSP 1993, S. II-455 ff., IEEE Signal Processing Society, Minneapolis, April 1993

[108] M. Adda-Decker, G. Adda, *Experiments on stress dependent phone modelling for continuous speech recognition systems*, Proc. ICASSP 1992, S. 561 ff.

[109] J. Bellegarda, P. de Souza, A. Nadas, D. Nahamoo, M. Picheny, L. Bahl, *Robust speaker adaptation using a piecewise linear acoustic mapping*, Proc. ICASSP-92, pp 445 ff

[110] R. Rose, *Discriminant Wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech*, Proc. ICASSP 1992, S. II-105 ff., IEEE Signal Processing Society, San Francisco, März 1992

[111] Colin Wills Wightman, *Automatic detection of prosodic constituents for parsing*, Ph.D. thesis, Boston University Graduate School, 1992

[112] D. Paul, *New results with the Lincoln tied mixture HMM CSR-System*, Proc. DARPA Speech and Natural Language Workshop 1991, S. 65-70.

[113] H. Gish, M.H. Siu, R. Rohlicek, *Segregation of Speakers for Speech Recognition and Speaker Identification*, in Proc. ICASSP-91, S. 873 ff

[114] G. Bamberg, F. Baur, *Statistik*, R. Oldenbourg Verlag, Munich/Vienna, ISBN 3-486-21325-3, 1989

[115] F. Class, A. Kaltenmeier, P. Regel, K. Trottler, *Fast speaker adaptation for speech recognition systems*, Proc. ICASSP-90, vol 1, pp 133 ff

[116] S. J. Cox, J. Bridle, *Unsupervised speaker adaptation by probabilistic spectrum fitting*, Proceedings ICASSP-89, vol 1, pp 294 ff

[117] T.H. Crystal, A.S. House, *Articulation rate and the duration of syllables and stress groups in connected speech*, Journal of the Acoustical Society of America, **88**(1):101-112, 1990

[118] T.H. Crystal, A.S. House, *Segmental durations in connected-speech signals: Current results*, Journal of the Acoustical Society of America, **83**(4), April 1988, S. 1553 ff.

[119] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, *Indexing by latent semantic analysis*, Journal of the Society for Information Science, 1990

[120] K. Fukunaga: *Introduction to statistical pattern recognition*, Academic Press Inc., San Diego, CA 92101, ISBN 0-12-269851-7, San Diego, 1990

[121] R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley Publishing Company, Reading, MA, USA, 1990

[122] Kai-Fu Lee, *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*, Ph. D. thesis, CMU-CS-88-148, Carnegie Mellon University, Pittsburgh, PA 15213, April 1988

[123] P.J.Price, C.W. Wightman, M. Ostendorf, J. Bear, *The use of relative duration in syntactic desambiguation*, Proc. ICSLP 90, 1.4.1-1.4.4.

[124] L.R. Rabiner, *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, in Proc. of the IEEE, 77(2):257-285, Februar 1989

[125] A. Waibel, K.F. Lee (editors), *Readings in Speech Recognition*, Morgan Kaufman Publishers, San Mateo, CA, 1990. ISBN 1-558-60124-4

[126] G. Yu, R.Schwartz: *Discriminant analysis and supervised vector quantization for continuous speech recognition*, Proc. ICASSP 1990, S. 685 ff.

[127] K. Choukri, G. Chollet, Y. Grenier, *Spectral transformations through canonical correlation analysis for speaker adaptation in ASR*, Proc. ICASSP-86, vol 4, pp 2659 ff

[128] S.E. Levinson, *Continuously variable duration Hidden Markov models for automatic speech recognition*, Computer Speech and Language **1**, No. 1, S. 29-45

[129] M.J.Russell, R.K.Moore, *Explicit modeling of state occupancy in hidden Markov model for automatic speech recognition*, Proc. ICASSP 85, S. 5 ff.

[130] G. Juettner, *Entwicklung eines wissensbasierten Lernsystems zum Aufbau eines Thesaurus für Information Retrieval Systeme*, Dissertation, TU München, Fakultät für Mathematik und Informatik, Mai 1987

[131] T.H. Crystal, A.S. House, *Segmental durations in connected speech signals: Preliminary results*, Journal of the Acoustical Society of America, **72**(3), Sept. 1982, S. 705 ff.

[132] J. Jaschul, *Speaker adaptation by a linear transformation with optimised parameters*, Proceedings of the ICASSP **3** (1982), 1657-1670

[133] C.J. van Rijsbergen, *Information Retrieval*, London: Butterworth, 1979, S. 174 ff

[134] F.J. Harris, *On the use of windows for harmonic analysis with the discrete Fourier transform*, Proc. IEEE, **66** (1978), 51-83

[135] G.D. Forney, *The Viterbi algorithm*, Proc. IEEE, vol 61, S. 268-278, März 1973

[136] G. Salton (Ed.), *The SMART Retrieval System / Experiments in Automatic Document Processing*, Prentice-Hall Inc., Englewood Cliffs, New Jersey 1971

[137] L. Baum, T. Petrie, G. Soules, N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains*, The Annals of Mathematical Statistics 1970, Vol 41, No 1, pp 164-171

[138] L. Baum, T. Petrie, *Statistical inference for probabilistic functions of finite state markov chains*, The Annals of Mathematical Statistics 1966, Vol 37, No 6, pp 1554-1563

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

[140] Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, 1938, ISBN 0-471-06259-6