

**Institut für Maschinelle Sprachverarbeitung  
Universität Stuttgart  
Azenbergstr. 12  
70174 Stuttgart**

## **Hybride Unit Selection für ein Sprachsynthesystem**

**Tanja Klankert  
Diplomarbeit Nr. 19**

<b>Prüfer:</b>	<b>Prof. Dr. Grzegorz Dogil PD Dr. Bernd Möbius</b>
<b>Betreuer:</b>	<b>Dipl. Ling. Antje Schweitzer PD Dr. Bernd Möbius</b>
<b>Begin der Arbeit:</b>	<b>1. November 2002</b>
<b>Ende der Arbeit:</b>	<b>30. April 2003</b>



# Kurzfassung

Für das SmartKom-Dialogsystem wurde eine hybride *Unit Selection* Strategie entwickelt, die den spezifischen Anforderungen der Projektdomäne entspricht. Die Domäne ist auf verschiedene Anwendungsbereiche eingeschränkt, umfasst jedoch aufgrund von Eigennamen und fremdsprachlichen Ausdrücken eine unbegrenzte Vokabulargröße. Die Synthesekomponente des Dialogsystems muss daher sowohl domänenspezifisches als auch unbeschränktes Textmaterial verarbeiten. Aufgrund dieser Dichotomie der Anwendungsdomäne werden zwei unterschiedliche *Unit Selection* Ansätze miteinander kombiniert. Der *Phonological Structure Matching*-Algorithmus bietet eine hohe Sprachqualität für domänenspezifische Äußerungen und Phrasen und reduziert den Aufwand der Einheitenwahl. Der *Cluster*-Algorithmus zeichnet sich durch Robustheit gegenüber unbeschränktem Textmaterial aus. Damit die Algorithmen größtmöglichen Nutzen von den Daten machen können, wurde für das Korpusdesign ebenfalls eine zweigeteilte Strategie verfolgt. Sowohl domänenspezifisches als auch unbeschränktes Material wurde für die Aufnahmen verwendet. Für die Auswahl von unbeschränktem Text wurde die Phonem- und Diphon-Abdeckung in verschiedenen phonetischen und prosodischen Kontexten kontrolliert. Für den domänenspezifischen Teil des Korpus wurden Phrasen und Wörter in verschiedene prosodische Kontexte eingebettet. Informelle Hörtests mit domänenspezifischen Beispieldialogen und domänenunabhängigen Testsätzen ergaben, dass die perzeptuelle Qualität der Sprachausgabe insbesondere im Hinblick auf Natürlichkeit und Stimmqualität als sehr gut bewertet wird.

# Abstract

For the Smartkom dialog system a hybrid unit selection strategy was developed which accounts for the specific requirements of the project domain. The domain is restricted, but it comprises names and foreign words and phrases, and is therefore unlimited in terms of vocabulary size. The synthesis module has to deal with domain specific and open-domain material. To handle this dichotomy of the application domain, two different unit selection approaches are combined. The *phonological structure matching* algorithm serves high speech quality for domain specific material and reduces the computational effort of the selection procedure. The cluster algorithm ensures robustness for unrestricted text material. The corpus design also follows a biased strategy to ensure that the algorithms make maximum use of the available data. For the open-domain part the coverage of diphones and phonemes in different phonetic and prosodic contexts is carefully controlled. The domain specific part contains names and phrases embedded in different prosodic contexts. Informal listening tests which included domain specific dialogs as well as open-domain sentences showed that the perceptual quality of the synthesized speech is regarded to be very good, particularly with respect to naturalness and voice quality.

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>iii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Mensch-Maschine-Kommunikation . . . . .	1
1.2 Aufbau von Sprachsynthesystemen . . . . .	3
1.3 Syntheseverfahren für Sprachsynthese . . . . .	4
1.4 Korpusbasierte Sprachsynthese . . . . .	8
1.5 Übersicht . . . . .	9
<b>2 Grundlagen der Unit Selection</b>	<b>11</b>
2.1 Prinzip der Unit Selection . . . . .	12
2.2 Unit Selection Algorithmen . . . . .	13
2.2.1 Markov-Modelle . . . . .	14
2.2.2 Kontextuelle Klassifikation . . . . .	23
2.3 Distanzmaße und Gewichte . . . . .	25
2.4 Phonetische und phonologische Bäume . . . . .	28
2.5 Korpusdesign . . . . .	29
<b>3 Unit Selection Systeme</b>	<b>33</b>
3.1 Unbeschränkte Anwendungsdomänen . . . . .	34
3.1.1 Das $\mu$ -Talk Sprachsynthesystem . . . . .	35
3.1.2 Das CHATR Sprachsynthesystem . . . . .	36
3.1.3 Das Festival-Sprachsynthesystem . . . . .	39
3.1.4 Das Laureate TTS-System . . . . .	43
3.2 Eingeschränkte Anwendungsdomänen . . . . .	46
3.2.1 Der CMU DARPA Communicator . . . . .	46
3.2.2 Das MIT Jupiter System und der ILEX Museumsführer . . . . .	48

<b>4</b>	<b>Hybride Unit Selection</b>	<b>55</b>
4.1	Korpusdesign . . . . .	55
4.2	Unit Selection Verfahren . . . . .	57
4.2.1	Phonologische Baumsuche . . . . .	57
4.2.2	Kontextuelle Klassifikation . . . . .	61
4.2.3	Auswahlalgorithmus . . . . .	63
4.2.4	Systemaufbau . . . . .	64
4.3	Ergebnisse und Diskussion . . . . .	66
<b>5</b>	<b>Resumée</b>	<b>71</b>
	<b>Literaturverzeichnis</b>	<b>75</b>
	<b>Abkürzungen</b>	<b>81</b>

# Abbildungsverzeichnis

1.1	Interaktion mit dem SmartKom-System . . . . .	2
1.2	Aufbau eines Sprachsynthesystems . . . . .	5
1.3	Aufbau der Synthesekomponente . . . . .	6
2.1	<i>Unit Selection</i> nach Hunt und Black . . . . .	12
2.2	Modelltopologien für <i>Unit Selection</i> . . . . .	16
3.1	Entwicklungsrichtungen in der Sprachsynthese . . . . .	34
3.2	Pfadauswahl beim Laureate TTS-System . . . . .	45
3.3	Ausschnitt eines phonologischen Baums . . . . .	50
3.4	Binärer Targetbaum mit Kandidaten . . . . .	51
3.5	PSM-Systemaufbau mit <i>back-off</i> Strategie . . . . .	54
4.1	Targetspezifikation für eine Beispieläußerung . . . . .	59
4.2	Targetbaum mit Kandidaten . . . . .	60
4.3	Viterbi-Suche . . . . .	63
4.4	<i>Optimal coupling</i> von PSM-Einheiten . . . . .	64
4.5	Aufbau des hybriden <i>Unit Selection</i> Systems . . . . .	65
4.6	Einfluss primärer Merkmale . . . . .	68
4.7	Vergleich der Ansätze . . . . .	70
5.1	Einordnung des hybriden Syntheseansatzes . . . . .	72



# 1 Einleitung

## 1.1 Mensch-Maschine-Kommunikation

Sprachliche Benutzerschnittstellen sind durch die zunehmende Verbreitung mobiler Geräte wie Laptops, PDAs und Handys im Alltag immer häufiger anzutreffen. Für die Kommunikation von Mensch und Maschine gewinnen multimediale Benutzerschnittstellen daher an Bedeutung. Die Darstellung von Information kann dabei visuell oder akustisch erfolgen. Der Vorteil sprachlicher Kommunikation liegt hauptsächlich in der hohen Übertragungsrate von 120-250 Wörtern pro Minute [ST95]. Dagegen liegt der Maximaldurchsatz bei Eingabe über die Tastatur bei 100-150 Wörtern pro Minute und setzt intensive Schulung voraus. Ein weiterer Vorteil ist die Bewegungsfreiheit des Benutzers. Vor allem Hände und Augen des Benutzers sind für weitere Aktivitäten frei. Einsatzgebiete für sprachliche Kommunikationsschnittstellen finden sich bei Auskunftssystemen wie Navigationssystemen, Reiseauskünften oder Wettervorhersage, bei Multimedia-Anwendungen im Unterhaltungsbereich, sowie im Büro und im Haushalt als interaktiver Anrufbeantworter, Email-Vorlesegerät oder Gerätebedienung. Der zunehmende Bedarf an sprachlichen Benutzerschnittstellen hat der Forschung und Entwicklung im Bereich Mensch-Maschine-Kommunikation weitere Impulse gegeben.

Aktuelle Forschungsprojekte wie das SmartKom-Projekt<sup>1</sup>, die sich mit der Interaktion von Mensch und Maschine befassen, verbinden die Darbietung textueller und grafischer Information mit sprachlicher Kommunikation, die durch paralinguistische Kommunikationsmittel wie Gestik und Mimik unterstützt wird. Ein Ziel des Projektes ist es, verschiedene Codesysteme wie Sprache, Gestik und Mimik in einem multimodalen Dialogsystem zu koordinieren. Die Information wird von dem System durch den Benutzereingabeagenten "Smartakus", ein freundliches blaues Männchen, visuell und akustisch präsentiert. Es gibt drei Versionen von SmartKom für verschiedene

---

<sup>1</sup> Dieses Projekt wird gefördert durch das Bundesministerium für Bildung und Forschung (BMBF), Nr. 01IL905E/6.

Anwendungsszenarien: *SmartKom-Public* ist ein multimodales Kommunikationsterminal an Flughäfen, Bahnhöfen oder Hotels, um Reisenden und Touristen ein personalisierbares Informationsportal zu liefern. *SmartKom-Mobil* ist eine PDA-Version, die als Navigationssystem im Auto oder als mobiler Internetzugang für Fußgänger dienen kann. *SmartKom-Home/Office* kann als Zugang für Informationsdienste und Kommunikationsdienste wie Telefon, Email, als elektronischer Programmführer fürs Fernsehen (*Electronic Programming Guide*, EPG), als Steuerung für Haushaltsgeräte und Unterhaltungselektronik genutzt werden. Das SmartKom-Dialogsystem kann Anfragen aus verschiedenen Domänen behandeln. Die Domäne beinhaltet Kino- und Fernsehprogramminformation, Touristeninformation, Routenplanung, Telefonbuch- und Adressbuchverwaltung. Abbildung 1.1 zeigt einen Ausschnitt aus einem Dialogszenario. Der Benutzer möchte eine Platzreservierung für einen Kinofilm machen. Der Benutzereingabeagent "Smartakus" zeigt die Platzordnung im Kino an. Das System fragt den Benutzer:

*Wo möchten Sie gerne sitzen?*

Der Benutzer zeigt auf die gewünschten Plätze und antwortet:

*Ich möchte diese beiden Plätze.*

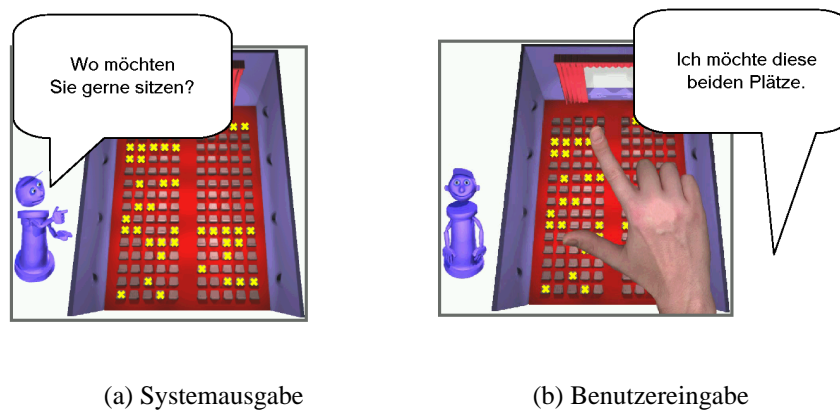


Abbildung 1.1: Interaktion mit dem SmartKom-System [WRB01]

Das System versucht die gesprochene Äußerung und die Geste des Benutzers zu erkennen und in eine interne Repräsentation zu überführen. Dabei müssen die sprachliche Äußerung und die Geste desambiguiert werden. Die Anfrage wird durch

Gesten- und Sprachanalyse unter der Hinzunahme von diskurs- und domänenspezifischem Wissen sowie Handlungswissen semantisch interpretiert. Mit Hilfe einer Datenbank wird eine entsprechende Antwort generiert. Diese wird wiederum in eine gesprochene Äußerung umgewandelt, die von “Smartakus” durch Mimik und Gestik begleitet wiedergegeben wird.

Die maschinelle Verarbeitung gesprochener Sprache umfasst die Analyse und Synthese sprachlicher Äußerungen. Spracherkennung sowie Sprecheridentifikation und -verifikation beruhen auf der Analyse gesprochener Äußerungen. Bei der Spracherkennung wird als Resultat der Analyse eine rechnerinterne Darstellung erzeugt. Sprachsynthese ist die Umwandlung einer rechnerinternen, z.B. textuellen Repräsentation natürlicher Sprache in eine lautsprachliche Äußerung. Auskunft- und Dialogsysteme wie das SmartKom-System beinhalten sowohl eine Spracherkennungs- als auch eine Sprachsynthesekomponente.

## 1.2 Aufbau von Sprachsynthesystemen

Sprachsynthesesysteme werden in *Text-to-speech* (TTS) und *Concept-to-speech* (CTS) Systeme unterteilt. CTS-Systeme erzeugen gesprochene Sprache auf der Basis einer internen linguistischen Repräsentation der Äußerung. Benutzerschnittstellen für Auskunft- und Dialogsysteme werden meist als CTS-Systeme realisiert. Sie verfügen über ein kontextabhängiges Modell, mit dem sie Anfragen interpretieren und Antworten generieren können. Aufgrund der Komplexität der zugrundeliegenden Kontextmodelle sind CTS-Systeme auf bestimmte Anwendungsdomänen begrenzt. Da die Sprachgenerierung von CTS-Systemen nicht auf einer textuellen Repräsentation basiert, die die Struktur der lautsprachlichen Äußerung nur rudimentär wiedergibt, ist eine Verbesserung der prosodischen Qualität synthetisierter Sprache von CTS-Systemen zu erwarten [Möb01]. TTS-Systeme verlangen im Gegensatz zu CTS-Systemen als Eingabe eine textuelle Repräsentation der Äußerung. Sie sind nicht notwendig auf eine begrenzte Domäne eingeschränkt und bieten daher eine höhere Flexibilität als CTS-Systeme.

Bei TTS-Systemen muss der Eingabetext zunächst linguistisch analysiert werden. Durch die syntaktische Analyse wird der Text in eine linguistische Repräsentation überführt, die mit Wortart- und Satzgliedinformation annotiert ist. *Part-of-speech* (POS) Tagger liefern Information über die Wortart, Chunker zerlegen Sätze in oberflächliche syntaktische Phrasen (*Chunks*). Die syntaktischen Phrasen korrespondie-

ren prosodischen Phrasen der lautsprachlichen Äußerung. Durch Graphem-Phonem-Konversion wird eine phonetische Transkription des Textes erstellt. Diese kann durch Suche in einem Lexikon (*Lexicon-look-up*) oder durch Ausspracheregeln (*Letter-to-sound Rules*) erfolgen. Schwierig ist die Transkription von Eigennamen und Fremdwörtern sowie Homographen. Hinzu kommen sprachspezifische Probleme. Das Englische benötigt beispielsweise eine große Anzahl von verschiedenen Ausspracheregeln. Hierbei zeigt sich der Vorteil von CTS-Systemen, bei denen die von der *Natural Language Generation*-Komponente (NLG) erzeugte linguistische Repräsentation bereits größtenteils relevante Informationen für die Weiterverarbeitung enthält.

Auf der Grundlage der linguistischen Repräsentation bestimmt die akustisch-prosodische Komponente die Phrasierung, den Intonationsverlauf sowie die Lautdauer der zu synthetisierenden Äußerung. Dies ist aufgrund einer Korrelation zwischen der syntaktischen Struktur und der prosodischen Struktur möglich. Ausgabe der prosodischen Komponente ist eine phonetische Repräsentation, die aus einer Folge von Merkmalsvektoren besteht. Die Merkmalsvektoren beinhalten symbolische Merkmale wie Segmentidentität und numerische akustische Parameter wie  $f_0$ -Werte, Lautdauer und Lautstärke. Aus der phonetischen Repräsentation erzeugt die Synthesekomponente ein digitales Sprachsignal. Abbildung 1.2 zeigt die Architektur eines Sprachsynthesesystems mit konkatenativer Synthese. Das System beinhaltet Komponenten für linguistische Textanalyse, akustische Prosodiegenerierung und Synthese der akustischen Sprachausgabe.

### 1.3 Syntheseverfahren für Sprachsynthese

Für die Synthese des Sprachsignals gibt es verschiedene Verfahren. Zu diesen zählen die konkatenative und die parametrische Synthese. Die akustisch-parametrische (Formant-) Synthese beruht auf dem *Quelle-Filter-Modell* der Spracherzeugung von Gunnar Fant [Fan60]. Bei der Spracherzeugung wird das stimmhafte oder stimmlose Anregungssignal entsprechend den Filtereigenschaften des Vokaltraktmodells und der Abstrahlung der Lippen spektral verändert. Die parametrische Synthese simuliert die Spracherzeugung anhand von Kontrollparametern, die die Modifikation des Ausgangssignals bestimmen. Vorteil dieses Syntheseverfahrens ist die gute Nachbildung der Formantstruktur und die Flexibilität. Die Anzahl der Regeln zur Steuerung der Formantstruktur und der Amplitude sowie der Charakteristika des Anregungssignals ist jedoch sehr groß. Problematisch ist der mangelnde Eindruck an Natürlichkeit der

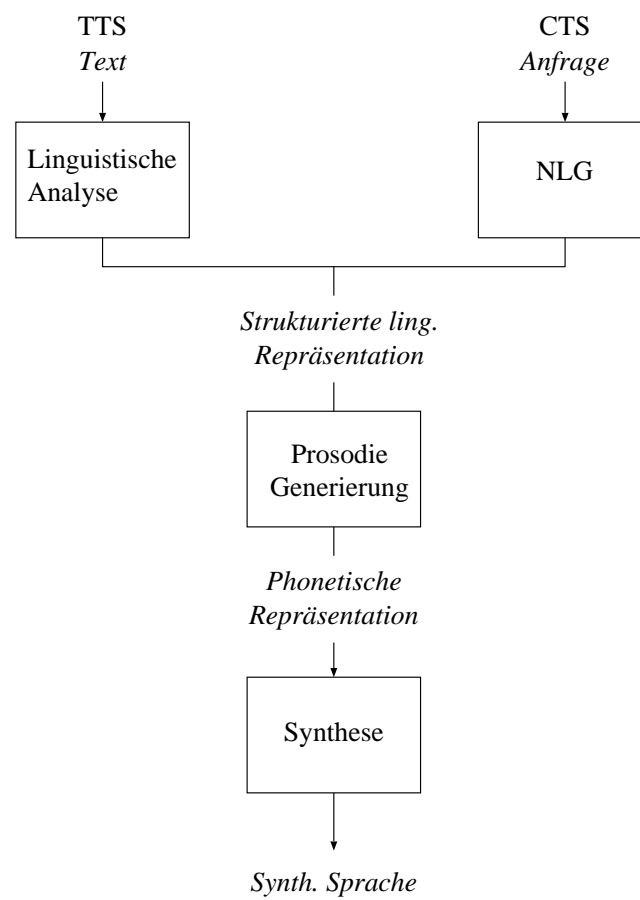


Abbildung 1.2: Aufbau eines Sprachsynthesystems

Sprachausgabe.

Das gängige Verfahren, um eine verständliche und natürliche Sprachausgabe zu produzieren, ist die Konkatenierung und Wiedergabe von zuvor aufgenommenen Sprachbausteinen. Die konkatenative Synthese verfügt derzeit über die beste Sprachqualität und ist das am häufigsten eingesetzte Syntheseverfahren. Bei der Synthese werden aus einer Datenbank die passenden Spracheinheiten ausgewählt und verknüpft. Die Auswahl der Spracheinheiten erfolgt anhand der phonetischen Repräsentation, die durch die Komponenten für die linguistische Analyse und die Prosodiegenerierung erzeugt wurde. Wenn keine Einheiten mit den gewünschten prosodischen Eigenschaften zur Verfügung stehen, müssen  $f_0$ -Kontur, Lautdauer und Lautstärke der Spracheinheiten durch Signalverarbeitungstechniken verändert werden. Signalmodifikationen führen jedoch zu einer geringeren Qualität des Sprachsignals. Um beispielsweise die Lautdauer eines Segments anzupassen, werden beim PSOLA-Verfahren [MC90] Teile des Sprachsignals wiederholt oder gelöscht. Dies hat eine Minderung der Natürlichkeit der Sprachausgabe zur Folge. Nach der Verknüpfung der Einheiten wird das Sprachsignal an den Konkatenationsstellen durch den Einsatz von Filtern geglättet. Abbildung 1.3 zeigt die Synthesekomponente eines konkatenativen Sprachsynthesesystems.

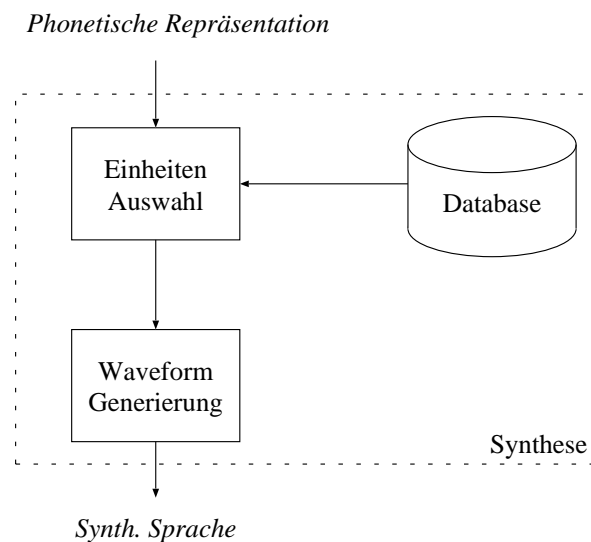


Abbildung 1.3: Aufbau der Synthesekomponente

Die verschiedenen konkatenativen Synthesesysteme unterscheiden sich im Wesentlichen durch den Aufbau der Sprachdatenbank. Ein wichtiger Aspekt ist die

Länge der Spracheinheiten und die Größe des Inventars. Im einfachsten Fall wird ein kleines Inventar von Sprachbausteinen aufgenommen und bei der Synthese re-kombiniert. Das Verfahren wird als *Canned Speech* bezeichnet und ist nur für Anwendungsgebiete mit kleinem, abgeschlossenem Vokabular und fester Äußerungsstruktur einsetzbar. Anwendungsbeispiele finden sich für die automatische Ansage in Aufzügen oder die Ansage von Haltestellen in Bussen und Bahnen. Auch Zeitan-sagen<sup>2</sup>, Wetterberichte und Navigationssysteme können durch einen ähnlichen An-satz realisiert werden. Ein Teil des Inventars besteht aus längeren Spracheinheiten wie Phrasen oder Sätzen. Diese fungieren als Trägersätze und werden mit kleineren Spracheinheiten nach dem *Slot-Filler*-Prinzip mit geringfügigen prosodischen Modi-fikationen ergänzt. Ein Beispiel für ein template-basiertes Wetteransagesystem findet sich in [BL00b]. Die Templates haben die Form

*The weather at HOUR, on DAY DATE, outlook OUTLOOK, TEMPERA-TURE degrees, winds WINDDIRECTION, WINDSPEED (with gusts to WIND-SPEED).*

Für das Wetteransagesystem genügt ein Sprachinventar von etwa 100 Sätzen. Problematisch für diesen Ansatz ist eine nachträglichen Erweiterung des Sprachinven-tars, wenn beispielsweise der Sprecher oder die Sprecherin nicht mehr zur Verfügung stehen. Für viele Anwendungen ist die Re-Synthese natürlicher Sprache bzw. die Wiedergabe von größeren Sprachbausteinen mit geringfügigen prosodischen Modi-fikationen eine einfache und effiziente Lösung, die über eine realistische Komplexität, exzellente Sprachsignalrepräsentation und Sprachsignalmodifikation verfügt [ST95].

Eine größere Herausforderung stellt die Verarbeitung und Synthese unbeschränk-ter Texte dar. Im Gegensatz zu *Canned Speech* besteht das Inventar aus kleinen Ein-heiten wie Diphonen, Triphonen, Halbsilben etc. Diese können flexibel kombiniert werden. Die Synthese von längeren Einheiten verfügt über eine bessere Steuerung von koartikulatorischen Effekten, weniger Konkatenierungsstellen und damit über eine höhere Natürlichkeit. Verbreitet sind diphonbasierte Inventare für die konkate-native Synthese. Diphone sind Spracheinheiten, die Lautübergänge zwischen zwei Phonemen enthalten. Die Grenzen liegen im spektral stabilen Bereich in der Mitte eines Phonems. Mikrokoartikulationseffekte sind durch die Lautübergänge bereits in den Signalabschnitten enthalten. Koartikulationseffekte, die über Silben- oder Wort-grenzen hinaus auftreten, werden nicht modelliert. Es gibt Ansätze, die Halbsilben

<sup>2</sup> Ein Beispiel für eine Zeitan-sage (*Talking Clock*) in verschiedenen Sprachen wie Englisch, Chinesisch und Nepali findet sich unter [http://festvox.org/ldom/ldom\\_time.html](http://festvox.org/ldom/ldom_time.html).

oder Silben als Syntheseeinheiten verwenden. Die Inventarerstellung ist jedoch aufgrund der großen Anzahl an silbenbasierten Einheiten sehr aufwendig.<sup>3</sup> Da die Speicheranforderungen mit der Inventargröße und mit der Länge der Einheiten zunehmen, ist eine vollständige Abdeckung in der Regel nicht möglich. Moderne Systeme verwenden Mischinventare zur Abdeckung koartikulatorischer Effekte [Sag88]. Bei der Inventarerstellung muss häufig ein Kompromiss zwischen guter Abdeckung und Inventargröße gemacht werden.

Die Inventarerstellung erfolgt in drei Schritten: Bei der Aufnahme von natürlicher Sprache müssen alle Phoneme einer Sprache sowie gegebenenfalls auch fremdsprachliche Phoneme für Fremdwörter in verschiedenen phonetischen und prosodischen Kontexten<sup>4</sup> abgedeckt werden. Für die klassische Diphonsynthese werden Phoneme in Trägerwörtern mit kontrollierter Prosodie aufgenommen. Die Sprachdaten werden automatisch oder manuell gelabelt und segmentiert und die Einheiten für das Inventar ausgewählt. Insbesondere das Labeln der Sprachdaten ist sehr zeitaufwendig. Automatische Werkzeuge wie *Aligner* können den Prozess der Inventarerstellung unterstützen. Da die Sprachqualität der konkatenativen Synthese sehr stark von der Qualität des zur Verfügung stehenden Inventars abhängt, muss zumindest eine manuelle Kontrolle erfolgen.

## 1.4 Korpusbasierte Sprachsynthese

Verständlichkeit und Natürlichkeit spielen bei der Bewertung von Sprachsynthese eine wichtige Rolle. Während die Verständlichkeit von konkatenativen Sprachsystemen als sehr gut und für viele Anwendungen ausreichend beurteilt wird, benötigt ein Hörer kaum mehr als 500 ms, um festzustellen, ob eine Äußerung von einem TTS-System erzeugt wurde [vS97b]. Hierfür wird die mangelnde Natürlichkeit der Sprachausgabe verantwortlich gemacht. Zur einer verbesserten Verständlichkeit trägt neben einer hohen Lautqualität eine sinnvolle Prosodie bei, die auf den Anwendungsbereich abgestimmt ist. Grund für die mangelnde Natürlichkeit und den Eindruck einer monotonen Prosodie ist die Begrenztheit des Synthese-Inventars. Eine Sprachausgabe mit natürlicher Prosodie und guter Signalqualität kann durch den Einsatz

<sup>3</sup> Eine Diskussion silben- und wortbasierter Ansätze für die konkatenative Synthese findet sich in [Möb00]. Aufgrund der LNRE-Charakteristik der Sprache folgert Möbius, dass solche Ansätze nur in abgeschlossenen Domänen anwendbar sind.

<sup>4</sup> Die Realisierung eines Phonems in einem spezifischen phonetischen und prosodischen Kontext wird als Allophon bezeichnet.

datenbasierter Verfahren bei der Auswahl der Synthesebausteine erreicht werden. Für datenbasierte Verfahren werden möglichst umfangreiche Datenbanken und effiziente Auswahlverfahren verwendet. Bei dem von ATR entwickelten und als *Unit Selection* bekannten korpusbasierten Syntheseverfahren werden die Spracheinheiten *online* zur Laufzeit des Systems ausgewählt [BC95, HB96]. Die Einheiten stammen aus einer größeren Sprachdatenbank, die viele Lautkombinationen in verschiedenen phonetischen und prosodischen Kontexten enthält. Damit wird die Limitierung des Syntheseinventars relativiert: die Länge der Spracheinheiten, die Klassifikation der Einheiten und die Anzahl der Vorkommen in der Datenbank sind variabel. Ein für die Diphonsynthese übliches fixes Inventar enthält für jedes Phonem nur ein oder einige wenige Token, die aus prosodisch neutralen Sprachaufnahmen stammen. Im Gegensatz dazu können bei der *Unit Selection* die Einheiten aus einer Sprachdatenbank mit prosodisch "reichhaltigen" Sätzen ausgewählt werden. Durch die Auswahl der Einheiten aus einer großen Datenbank während der Laufzeit kann der phonetische und prosodische Kontext optimal approximiert und Koartikulationseffekte besser modelliert werden. Dies trägt zur Natürlichkeit der Sprachausgabe bei. Die Einheiten bei der *Unit Selection* sind durchschnittlich länger als bei der konkatentativen Synthese mit fixem Inventar. Längere Spracheinheiten haben den Vorteil, dass sich die Anzahl der Konkatenierungsstellen reduziert und damit weniger Signalmodifikationen nötig sind. Da nachträgliche Signalmanipulationen Störungen des Sprachsignals bewirken können, führt die Verminderung an Signalverarbeitung ebenfalls zu einer natürlicheren Sprachausgabe.

## 1.5 Übersicht

*Unit Selection* hat sich als ein Standardverfahren für Sprachsynthesysteme etabliert. In den letzten 10 Jahren wurde eine Vielzahl unterschiedlicher *Unit Selection* Ansätze entwickelt und in verschiedenen Systemen eingesetzt. Die grundlegenden Verfahren und Techniken für *Unit Selection* werden in Abschnitt 2 vorgestellt. Die Prinzipien und die Algorithmen für die Einheitenauswahl werden erläutert und die Verwendung von Distanzmaßen und Gewichten sowie der Einsatz phonologischer Bäume als Merkmalsbeschreibung diskutiert. Auf die Entwicklung eines speziellen Korpusdesigns wird eingegangen.

Abschnitt 3 stellt ausgewählte Ansätze für *Unit Selection* anhand verschiedener Sprachsynthesysteme vor, auf denen die hier vorgestellte Arbeit aufbaut. Die Syn-

theseverfahren werden anhand verschiedener Entwurfskriterien bewertet. Da die Anwendungsdomäne den Schwierigkeitsgrad der Synthese bestimmt, werden die vorgestellten Systeme in unbeschränkte TTS-Systeme und solche für eingeschränkte Domänen unterteilt. Zu der ersten Gruppe gehören das bei ATR entwickelte CHATR-TTS-System, das als erstes Synthesystem *Unit Selection* einsetzt und als einschlägigen Formalismus für die Einheitenauswahl *Hidden Markov Modelle* verwendet, dessen Vorgängersystem  $\mu$ -Talk, das bereits wichtige Prinzipien der *Unit Selection* umsetzt und die Einheiten durch kontextuelle Klassifikation auswählt, das Festival Synthesystem der Universität Edinburg, das den Ansatz von CHATR mit einem Verfahren zur kontextuellen Klassifikation (*Acoustic Clustering*) verbindet, und das Laureate TTS-Systems von British Telecom, dessen Sprachinventar aus phonologischen Bäumen besteht. Die für den CMU DARPA Communicator modifizierte Version von Festival und das für das MIT Jupiter System und den ILEX Museumsführer eingesetzte *Phonological Structure Matching* Verfahren, dessen Einheitensuche auf der Verwendung phonologischer Bäume basiert, gehören zu den Systemen für eingeschränkte Anwendungsdomänen.

Das SmartKom-Projekt stellt für Sprachsynthesysteme ein schwieriges Anwendungsgebiet dar, da es über eine eingeschränkte Domäne mit offenem Vokabular verfügt. Für die Applikation wurde eine hybride *Unit Selection* Strategie entwickelt, die in Abschnitt 4 erläutert wird. Die *Unit Selection* Komponente verbindet den in dem Sprachsynthesystem Festival verwendeten Ansatz des *Acoustic Clusterings* für beschränktes Textmaterial mit dem *Phonological Structure Matching* Verfahren für domänenspezifisches Material. Für die Synthese wurde ein an die Gegebenheiten der Anwendungsdomäne angepasstes zweigeteiltes Korpusdesign verfolgt. Das hybride *Unit Selection* System besteht aus folgenden Komponenten: Für die Einheitensuche werden phonologische Bäume und kontextuelle Klassifikation verwendet, als Auswahlalgorithmus wird eine *Hidden Markov Modell*-basierte Viterbi-Suche eingesetzt. Eine Übersicht über den Systemaufbau wird angegeben, an die sich eine Diskussion der Probleme und Ergebnisse anschließt.

Abschnitt 5 bewertet die Ergebnisse der Arbeit und ordnet die Syntheseleistung des hybriden Ansatzes ein.

## 2 Grundlagen der Unit Selection

*Unit Selection* ist ein Verfahren, um die Probleme, die durch die Limitierungen eines fixen Sprachinventars entstehen, zu lösen. Als akustisches Inventar steht bei diesem Verfahren ein vollständiges Sprachkorpus zur Verfügung. Viele Einheiten kommen in dem Korpus mehrfach in unterschiedlichen phonetischen und prosodischen Kontexten vor. Die Auswahl der passenden Einheiten geschieht anhand kontextueller Merkmale während der Laufzeit durch einen geeigneten Auswahlalgorithmus. Um eine Äußerung zu synthetisieren, werden die längsten verfügbaren Verkettungen von Einheiten aus dem Sprachkorpus ausgewählt, die den Einheiten der Äußerung entsprechen. Die aus dem Korpus ausgewählten Einheiten werden konkateniert und die Äußerung synthetisiert. Idealerweise würde die Äußerung als Ganzes im Korpus gefunden werden und der Aufwand zur Synthese auf ein "Play back" reduziert. Aufgrund der Komplexität und Kombinatorik der Sprache ist dies bei unbeschränkten Anwendungsdomänen unwahrscheinlich. Dennoch ist die Wahrscheinlichkeit sehr hoch, dass in dem Sprachkorpus Einheiten aus mehreren Segmenten, ganzen Silben oder Wörtern gefunden werden, aus denen die Äußerung synthetisiert werden kann. Die durchschnittliche Länge der ausgewählten Einheiten ist größer als die Länge von Diphonen oder Halbsilben [Möb00]. Makrokoartikulatorische Effekte, die sehr schwer zu steuern sind und erheblich zum mangelnden perzeptiven Eindruck der Natürlichkeit bei der Diphonsynthese beitragen, können durch die Berücksichtigung des phonetischen und prosodischen Kontextes modelliert werden. Die Anzahl der Konkatenierungspunkte sowie das Maß an notwendigen Signalmodifikationen ist geringer als bei der Diphonsynthese, was zu einer verbesserten Natürlichkeit der Sprachausgabe beiträgt. Bei dem von ATR entwickelten TTS-System CHATR [BT94] wurde gänzlich auf Signalverarbeitung verzichtet unter der Annahme, dass der Hörer gelegentlich auftretende spektrale Diskontinuitäten und prosodische "Mismatches" toleriert, wenn die Sprachqualität des Systems insgesamt sehr gut ist.

## 2.1 Prinzip der Unit Selection

Bei der *Unit Selection* muss für die Auswahl der Einheiten, aus denen eine Äußerung synthetisiert wird, ein Maß definiert werden, das einen Vergleich der Einheiten zulässt. Sagisaka [Sag88], der als erster einen Ansatz zur Verwendung von Mischinventaren für konkatentative Synthese vorschlägt, unterscheidet zwischen *unit distortion* und *continuity distortion*. *Unit distortion* bezeichnet den spektralen Abstand zwischen den Einheiten der zu synthetisierenden Äußerung (*target units*) und den Inventareinheiten (*candidate units*), die als Kandidaten für die Synthese ausgewählt werden. *Continuity distortion* bezeichnet den spektralen Abstand zweier aufeinanderfolgender Inventareinheiten an der Konkatenierungsstelle. Sie ist ein Maß für die bei der Konkatenierung auftretenden spektralen Diskontinuitäten und den erforderlichen Grad an nachträglichen Signalmodifikationen. Analog unterscheiden Hunt und Black [HB96] zwei Arten von Kosten, *target costs* und *concatenation costs*, die der Auswahlalgorithmus bei der Suche nach den passenden Einheiten zu minimieren versucht (Abbildung 2.1).

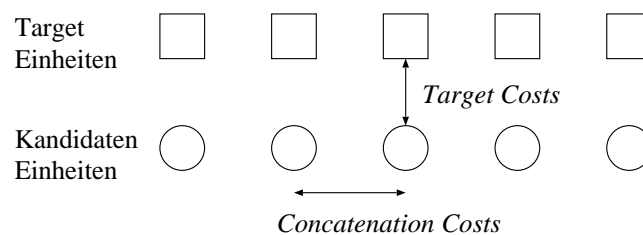


Abbildung 2.1: *Unit Selection* nach Hunt und Black [HB96]

Hunt und Black betrachten die Einheiten der Sprachdatenbank als ein Zustandsübergangnetzwerk, dessen Zustandskosten die Targetkosten und dessen Übergangskosten die Konkatenierungskosten sind. Die Einheiten der Sprachdatenbank sind mit multidimensionalen Merkmalsvektoren annotiert, die segmentelle Eigenschaften und den phonetischen und prosodischen Kontext beschreiben. Die Merkmalsvektoren können sowohl symbolische kategorielle als auch numerische Werte enthalten. Die Aufbereitung der Sprachdatenbank erfolgt *offline* mittels automatischer Werkzeuge oder manuell. Während der Einheitensuche werden die Merkmalsvektoren der Targetäußerung berechnet. Bei *Text-to-speech* Systemen können nur solche Merkmale berücksichtigt werden, die ausgehend von der textuellen Repräsentation der Targetäußerung berechnet werden können. Bei *Concept-to-speech* Systemen kann

die Repräsentation auch zusätzliche Merkmale enthalten, die die Aussprache oder Betonung festlegen. Die Targetkosten drücken aus, wie gut die im Sprachkorpus zur Verfügung stehenden Einheiten die Spezifikation der Einheiten der Targetäußerung approximieren. Für die Berechnung der Konkatenierungskosten, die die Signaleigenschaften aufeinanderfolgender Kandidateneinheiten an der Konkatenierungsstelle reflektieren, können alle *offline* oder *online* berechneten Merkmale verwendet werden.

## 2.2 Unit Selection Algorithmen

Die Modellierung der Sprachdatenbank als Zustandsübergangnetzwerk nach Hunt und Black [HB96] genügt der formalen Definition eines Markov-Modells, das im folgenden Abschnitt näher erläutert wird. Das Netzwerk repräsentiert das Suchproblem: bei der Einheitenauswahl wird der optimale Pfad durch das Zustandsübergangnetzwerk unter Minimierung der Kosten gesucht. Ein *Hidden Markov Modell* (HMM) ist ein stochastisches Modell. Der Ansatz von Hunt und Black unterscheidet sich von einem Markov-Modell darin, dass statt Wahrscheinlichkeitsverteilungen Kostenfunktionen verwendet werden.

Es werden drei Algorithmen für die Lösung des Suchproblems<sup>1</sup> vorgestellt: Der Viterbi-Algorithmus ist das klassische Suchverfahren zur Bestimmung des optimalen Pfades in einem HMM. Der Ressourcen- und Zeitaufwand für die Viterbi-Suche ist jedoch erheblich, so dass häufig keine vollständige Suche durchgeführt werden kann und der Suchraum eingeschränkt werden muss. Die Effizienz des Netzwerkes und das Maß, in dem die Kosten für die Bestimmung des optimalen Pfades vorausberechnet werden können, beeinflussen den Zeitaufwand für die *Unit Selection* entscheidend. Die Strahlsuche und der A\*-Algorithmus sind heuristische Graphsuchverfahren, die den Aufwand der Einheitenauswahl auf unterschiedliche Weise reduzieren.

Eine weitere Möglichkeit, das Suchproblem der *Unit Selection* zu lösen, ist die kontextuelle Klassifikation des Einheiteninventars. Unter Klassifikation wird die automatische Kategorisierung von Merkmalsvektoren verstanden. Die Einheiten der Sprachdatenbank, die durch phonetische und prosodische Merkmale spezifiziert sind, können in Äquivalenzklassen (*cluster*) zusammengefasst werden. Die Dimensionie-

---

<sup>1</sup> Das Suchproblem kann allgemein als Optimierungsproblem behandelt werden, das mittels *Dynamischer Programmierung* gelöst werden kann. Es kann gezeigt werden, dass der Viterbi-Algorithmus eine Variante der *Dynamischen Programmierung* ist [Jel97].

zung des Merkmalsraums erfolgt auf der Grundlage statistischer Eigenschaften des Einheiteninventars. Bei der Einheitenauswahl wird die optimale Einheitenfolge in Abhängigkeit des Kontextes bestimmt. Die kontextuelle Klassifikation erfolgt mit Hilfe des Entscheidungsbaumkonzeptes (*decision tree*), das im zweiten Abschnitt erläutert wird.

### 2.2.1 Markov-Modelle

Markov-Modelle sind im Bereich der statistischen Modellierung von Sprachaspekten in der Spracherkennungs- und Sprachsyntheseforschung die am weitesten verbreitete und bislang erfolgreichste Technik [ST95]. Es handelt sich dabei um einen Formalismus, der einen zweistufigen Zufallsprozess modelliert, welcher in einem festen Zeitraster sukzessiv eine Folge von Zuständen durchläuft und dabei Beobachtungsfolgen emittiert bzw. konsumiert. Die Ausgabe hängt statistisch nur vom aktuellen Zustand ab.

Ein HMM wird durch seine Struktur, d.h. die Anzahl der Zustände und deren Übergänge, sowie durch seine statistischen Parameter, d.h. die Übergangswahrscheinlichkeiten und die Ausgabeverteilung, vollständig beschrieben. Formal kann ein HMM definiert werden als ein 5-Tupel  $\lambda = (Q, K, \Pi, A, B)$ , wobei  $Q$  eine endliche Menge der Zustände,  $K$  ein endliches Ausgabealphabet und  $\Pi, A, B$  Wahrscheinlichkeiten für die Einnahme des Anfangszustandes, die Zustandsübergangswahrscheinlichkeiten und die Ausgabeverteilung sind.  $O$  denotiert die Beobachtungsfolge, die beim Durchlaufen der nicht bekannten Zustandsfolge  $q$  erzeugt wird.  $q$  wird dabei durch einen stochastischen Prozess erzeugt und ist eine Folge von Zufallsvariablen. Die diskrete Symbolfolge  $O$  ist das Resultat eines zweiten stochastischen Prozesses, der in Abhängigkeit von der Zustandsfolge  $q$  aus dem Ausgabealphabet  $K$  Symbole generiert.

Menge der Zustände	$Q = \{s_1, \dots, s_N\}$
Zustandsfolge	$q = q_1, \dots, q_T, \quad q_t \in Q$
Ausgabealphabet	$K = \{v_1, \dots, v_K\}$
Ausgabefolge	$O = o_1, \dots, o_T, \quad o_t \in K$
Anfangswahrscheinlichkeiten	$\pi_i = P(q_1 = s_i), \quad \sum_{i=1}^N \pi = 1$
Übergangswahrscheinlichkeiten	$A = [a_{ij}]_{N \times N}, \quad a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i)$
Emissionswahrscheinlichkeiten	$B = [b_{jk}]_{N \times K}, \quad b_{jk} = P(o_t = v_k \mid q_t = s_j)$

Die Übergangswahrscheinlichkeiten, die einen einfachen, kausalen und stationären Prozess charakterisieren, erfüllen die Bedingung

$$P(q_t | q_1 \dots q_{t-1}) = P(q_t | q_{t-1})$$

Diese Eigenschaft wird auch als Markov-Eigenschaft bezeichnet. Sie besagt, dass ein Zustand nur vom vorangehenden Zustand abhängt. Die Übergangswahrscheinlichkeiten  $A$  definieren zusammen mit den Anfangsbedingungen  $\pi$  diskrete Prozesse, die als Markov-Ketten bezeichnet werden. Die Ausgabeverteilung ist nur durch den aktuellen Zustand bedingt.

$$P(o_t | o_1 \dots o_{t-1}, q_1 \dots q_t) = P(o_t | q_t)$$

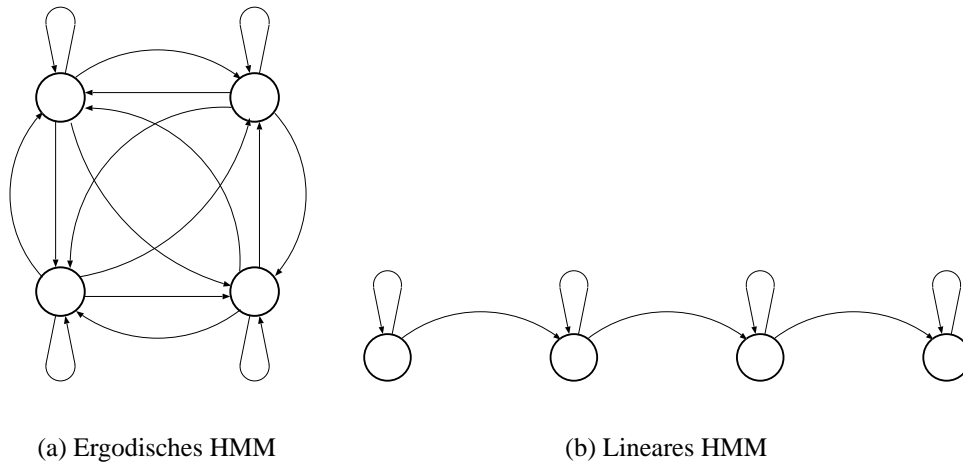
Das Suchproblem der *Unit Selection* in der Sprachsynthese kann wie folgt als ein Markov-Modell formalisiert werden: Die bei der Synthese generierte Targetspezifikation der Eingabeäußerung wird durch die diskrete Folge  $O$  von Targeteinheiten beschrieben. Das Symbolalphabet  $K$  entspricht den möglichen Merkmalsvektoren für eine Targeteinheit, die von den linguistischen und prosodischen Analysekomponenten des Systems generiert werden. Das Einheiteninventar der Sprachdatenbank ist die Menge der möglichen Merkmalsvektoren  $Q$ . Die Zustandsfolge  $q$  ist eine Folge von Einheiten aus der Sprachdatenbank. Die statistischen Parameter werden durch gewichtete Kostenfunktionen ersetzt. Zur Berechnung der Kosten wird eine Distanzmetrik über den Merkmalsvektoren definiert. Die Übergangswahrscheinlichkeiten entsprechen den gewichteten Konkatenierungskosten  $A \simeq C^c$ , die Emissionswahrscheinlichkeiten den gewichteten Targetkosten  $B \simeq C^t$ .<sup>2</sup>

Für HMMs werden verschiedene Modelltopologien unterschieden, darunter das *Lineare Modell*, das *Bakis-Modell* und das *Links-Rechts-Modell* [ST95]. Für *Unit Selection* Verfahren wird vor allem das *Lineare Modell* verwendet (unter anderem in [Con99, BCS<sup>+</sup>99]). Hunt und Black [HB96] betrachten die Datenbank als *Ergodisches HMM* erster Ordnung. Abbildung 2.2 zeigt geeignete Modelltopologien für *Unit Selection*.

Mit Hilfe von Markov-Modellen lassen sich drei grundsätzliche Probleme behandeln:

1. das Evaluierungsproblem: Wie kann die Wahrscheinlichkeit einer Beobachtungsfolge  $P(O | \lambda)$  für ein gegebenes Modell  $\lambda = (\Pi, A, B)$  und eine Beobachtungsfolge  $O$  effizient berechnet werden?

<sup>2</sup> Als Anfangswahrscheinlichkeiten können die Übergangskosten  $C_{S q_1}^c$  von Stille  $S$  zu der ersten Spracheinheit  $q_1$  betrachtet werden.

Abbildung 2.2: Modelltopologien für *Unit Selection*

2. das Dekodierungsproblem: Wie kann für eine gegebene Beobachtungsfolge  $O$  und ein Modell  $\lambda = (\Pi, A, B)$  die wahrscheinlichste Zustandsfolge  $q$  bestimmt werden?
3. das Lern- und Optimierungsproblem: Wie kann für eine Beobachtungsfolge  $O$  und eine Menge möglicher Modelle durch Variieren der Modellparameter  $\lambda = (\Pi, A, B)$  das Modell  $\tilde{\lambda}$  gefunden werden, das die höchste Beobachtungswahrscheinlichkeit  $P(O \mid \tilde{\lambda}) > P(O \mid \lambda)$  hat und damit die Daten am besten beschreibt?

Für die *Unit Selection* kann das erste Problem betrachtet werden als Bewertung, wie gut die Parameter eines Modells eine gegebene Folge  $O$  von Targeteinheiten beschreiben.<sup>3</sup> Das Modell der Sprachdatenbank beinhaltet die gewichteten Kostenfunktionen  $C^c$  und  $C^t$ . Aus mehreren zur Verfügung stehenden Modellen wird dasjenige Modell ausgewählt, das die Daten der Targetäußerung am besten modelliert. Durch Variieren der Gewichte für die einzelnen Parameter der Kostenfunktionen kann ein Modell gefunden werden, das die geringsten Kosten für eine Targetäußerung  $O$  liefert. Durch iterative Berechnung der Kosten von Targetäußerungen für ein (zufällig gewähltes) Modell werden diese minimiert, indem die Modellparameter modifiziert werden. Die Modellparameter, die am häufigsten verwendet werden, können anhand

<sup>3</sup> Dieses Problem wird in der Statistik durch die Berechnung der Produktionswahrscheinlichkeit aus den Vorwärts- bzw. Rückwärtswahrscheinlichkeiten mit Hilfe des *Forward*- bzw. *Backward*-Algorithmus gelöst.

der Berechnung identifiziert werden. Durch Minimieren dieser Kosten erhält man ein neues Modell, das bei der Berechnung geringere Kosten für eine gegebene Targetäußerung erzielt. Die Methode entspricht der *Expectation Maximization* (EM) Methode zur Lösung des dritten Problems.<sup>4</sup> Sie wird als Trainieren eines Modells verstanden und auf Trainingsdaten ausgeführt. Das zweite Problem beschreibt das zentrale Suchproblem der *Unit Selection*: Wie kann für eine gegebene Targetspezifikation  $O$  einer Äußerung und ein gegebenes Modell der Sprachdatenbank die kostengünstigste Einheitenfolge  $q$  gefunden werden, die die Targetbeschreibung am besten approximiert?

### Viterbi-Suche

Das zweite Problem, die beste Kandidatenfolge für ein gegebenes Modell  $\lambda$  und eine bestimmte Folge  $O$  von Targeteinheiten zu finden, lässt sich mit Hilfe des Viterbi-Algorithmus lösen.

Die Kosten  $C$ , die für ein gegebenes Modell  $\lambda$  beim Durchlaufen einer bestimmten Zustandsfolge  $q$  anfallen, können mit Hilfe der Markov-Eigenschaft beschrieben werden als

$$C(q | \lambda) = C(q_1 \dots q_T | \lambda) = C_{S q_1}^c + \sum_{t=2}^T C_{q_{t-1} q_t}^c$$

wobei  $C_{S q_1}^c$  die Kosten für den Übergang von Stille  $S$  zum ersten Zustand  $q_1$  bezeichnen und  $C_{q_{t-1} q_t}^c$  die Kosten für den Übergang von dem Zustand  $q_{t-1}$  zum Zustand  $q_t$ .

Die Gesamtkosten für eine Targetäußerung  $O$  beim Durchlaufen einer bestimmten Kandidatenfolge  $q$  ist dann

$$C(O | q, \lambda) = C(o_1 \dots o_T | q_1 \dots q_T, \lambda) = \sum_{t=1}^T C_{q_t}^t(o_t)$$

Analog zur *Bayes-Regel* kann man die Gesamtkosten berechnen durch

$$C(O, q | \lambda) = C(O | q, \lambda) + C(q | \lambda) = C_{S q_1}^c + C_{q_1}^t(o_1) + \sum_{t=2}^T [C_{q_{t-1} q_t}^c + C_{q_t}^t(o_t)]$$

Die optimale Kandidatenfolge  $q^*$  mit den geringsten Kosten kann beschrieben werden als

<sup>4</sup> Zur Lösung des dritten Problems wird in der Statistik der auf der EM-Methode beruhende *Forward-Backward-* oder *Baum-Welch-*Algorithmus eingesetzt.

$$C(O, q^* | \lambda) = \min_{q \in Q^T} C(O, q | \lambda)$$

Es können mehrere optimale Zustandsfolgen existieren. Der Viterbi-Algorithmus berechnet die minimalen Kosten für jeden Zustand. Er ist im Folgenden angegeben (Algorithmus 2.1).

$$\vartheta_t(j) = \min\{C(o_1 \dots o_t, q_1 \dots q_t | \lambda) \mid q \in Q^T, q_t = j\}$$

Über allen berechneten Kosten der vorangegangenen Zustände wird das Minimum in einer Matrix  $\Psi = [\psi_t(j)]$  gespeichert. Durch *Backtracking* kann man die kostengünstigste Zustandsfolge  $q^*$  ausgehend vom Endzustand mit den geringsten Kosten finden.

---

```

for all  $j = 1, \dots, N$  do {Initialisierung}
     $\vartheta_1(j) \leftarrow C_{S_j}^c$ 
     $\psi_1(j) \leftarrow 0$ 
end for
for all  $j = 1, \dots, N$  do {Rekursion}
     $\vartheta_t(j) \leftarrow \min_i (\vartheta_{t-1}(i) + C_{ij}^c) + C_j^t(o_t)$ 
     $\psi_t(j) \leftarrow \operatorname{argmin}_i \vartheta_{t-1}(i) + C_{ij}^c$ 
end for
 $P^*(O | \lambda) \leftarrow \min_j \vartheta_T(j)$  {Terminierung}
 $q_T^* \leftarrow \operatorname{argmin}_j \vartheta_T(j)$ 
for  $t = T - 1, \dots, 1$  do {Rückverfolgung}
    {ergibt sich eine optimale Folge durch}
     $q_t^* \leftarrow \psi_{t+1}(q_{t+1}^*)$ 
end for

```

---

Algorithmus 2.1: Viterbi-Algorithmus

Die Auswahl der Einheiten ist ein Ressourcenproblem, da enorme Anforderungen an die Rechenleistung und Speicherkapazität der eingesetzten Hardware gestellt werden. Die Anzahl der verfügbaren Einheiten im Sprachkorpus bedingt einen erheblichen Speicherbedarf für die Merkmalsvektoren. Während der Suche ist eine große Anzahl an Berechnungen für die Target- und Konkatinationskosten durchzuführen. Je größer das Korpus ist, desto umfangreicher ist der Suchraum.

## Strahlsuche

Der Aufwand der Viterbi-Dekodierung hängt von der Anzahl der Zustände und der Topologie des Modells ab. Für Netzwerke von HMMs, wie sie in der Spracherkennung und Sprachsynthese eingesetzt werden, sind die benötigte Rechenzeit und die Speicheranforderungen häufig unrealistisch. Die Strahlsuche (*beam search*) reduziert den Dekodierungsaufwand durch geeignete Einschränkung des Suchraums. Das Prinzip der Strahlsuche besteht darin, bei jedem Zeittakt nur ein Bündel von geeigneten Zuständen weiterzuverfolgen; die übrigen Zustände werden von der Suche ausgeschlossen. Das Verfahren liefert nicht immer eine korrekte Lösung, da der optimale Lösungspfad durch das Netzwerk abgeschnitten werden kann. Es ist jedoch ein bewährter Kompromiss zwischen Korrektheit und Effizienz [ST95].

Um eine geeignete Menge an vielversprechenden Zuständen festzulegen, muss ein Kriterium für die Suchbescheidung definiert werden. In der Literatur sind verschiedene Kriterien anzutreffen [ST95]. Für die Kostenberechnung bei der *Unit Selection* kann ein Kriterium definiert werden als die Menge der *aktiven* Zustände  $Q$  zum Zeitpunkt  $t$

$$Q_t = \{i \mid \vartheta_t(i) \leq \min_j \vartheta_t(j) \cdot \kappa\}$$

wobei  $\kappa$  ein geeigneter Wert ist, der für verschiedene Anwendungen experimentell bestimmt werden muss. Das Kostenlimit ist zeitabhängig und passt sich während der Suche an. Die Effizienz des Verfahrens hängt von der Strahlbreite ab. Die Korrektheit wird dadurch bedingt, dass das Kostenlimit eine obere Schranke ist. Der unten angegebene strahlgesteuerte Viterbi-Algorithmus (Algorithmus 2.2) ist eine für die *Unit Selection* modifizierte Variante des in [ST95] angegebenen Algorithmus. Die Menge der *aktiven* Zustände  $Q_t$  ist als kostensortierte Liste implementiert, deren bestbewertete Elemente als erste bearbeitet werden. Als Schätzwert für das Kostenlimit wird das partielle Minimum  $\Gamma_t$  verwendet.

Bei dem vorgegangenen Dekodierungsverfahren wird der Suchraum bei der Suche mit jedem Schritt eingeschränkt. Bei der asynchronen Dekodierung wird diese Beschränkung gelockert. Der implizite Suchraum wird Schritt für Schritt expandiert, bis ein vollständiger Pfad mit maximaler Bewertung - hier mit minimalen Kosten - gefunden wird. Zu den asynchronen Dekodierungsverfahren zählt der A\*-Algorithmus.

---

```

 $\mathcal{Q}_0 \leftarrow \{1\}$  {Initialisierung}
for all  $t$  do
   $\Gamma_t \leftarrow MAX$ 
end for
repeat {Rekursion}
  for all  $t = 0, \dots, T - 1$  do
    for all  $i \in \mathcal{O}_t$  do {in Listensortierung}
      for all  $j$  and  $C_{ij}^c < MAX$  do
         $\vartheta \leftarrow \vartheta_t(i) + C_{ij}^c + C_j^t(o_{t+1})$ 
        if  $\vartheta \geq \Gamma_{t+1} \cdot \kappa$  then
          weiter
        end if
        if  $j \in \mathcal{Q}_{t+1}$  and  $\vartheta > \vartheta_{t+1}(j)$  then
          weiter
        end if
         $\vartheta_{t+1}(j) \leftarrow \vartheta$ 
         $\psi_{t+1}(j) \leftarrow i$ 
        if  $\vartheta \leq \Gamma_{t+1}$  then
           $\Gamma_{t+1} = \vartheta$ 
        end if
        if  $j \notin \mathcal{Q}_{t+1}$  then
          if  $\vartheta \leq \Gamma_{t+1}$  then
            Füge  $j$  am Kopf der Liste  $\mathcal{Q}$  ein
          else
            Füge  $j$  am Ende der Liste  $\mathcal{Q}$  ein
          end if
        end if
      end for
    end for
  end for
until
  {Terminierung wie Viterbi-Algorithmus}

```

---

Algorithmus 2.2: Strahlgesteuerte Viterbi-Suche

### A\*-Algorithmus

Der A\*-Algorithmus ist ein Graphsuchverfahren, das eine heuristisch informierte Suche in einem bewerteten gerichteten Graphen realisiert. Die Zustände des Übergangnetzwerkes sind jetzt die Knoten in einem bewerteten gerichteten Graphen. Ein bewerteter gerichteter Graph ist ein Tripel  $(K, T, d)$  mit

$$\begin{aligned} \text{Menge an Knoten} & \quad K = \{k_1, k_2, k_3, \dots\} \\ \text{Menge an Kanten} & \quad T \subseteq K \times K \\ \text{Kostenfunktion} & \quad d: T \longrightarrow \mathbb{R}^+ \end{aligned}$$

Die Menge der Knoten ist geordnet bezüglich der Präzedenzrelation ' $\prec$ ' und es gilt  $k_1 \prec k_2$ , falls es eine Kante  $(k_1, k_2) \in T$  von  $k_1$  nach  $k_2$  gibt. Ein Knoten  $k \in K$  heißt Startknoten (Zielknoten), wenn er keinen Vorgänger (Nachfolger) hat. Die Menge der Startknoten sei  $K_\alpha$ , die Menge der Zielknoten  $K_\omega$ . Ein Pfad ist eine Knotenfolge  $k \in K^m$  mit  $k_1 \prec k_2 \prec \dots \prec k_m$ , wobei die Kosten des Pfades berechnet werden durch die Summe aller Teilpfade

$$D(k) = d(k_1, k_2) + \dots + d(k_{m-1}, k_m)$$

Das Graphsuchproblem besteht darin, einen Pfad von einem Startknoten zu einem Zielknoten mit minimalen Kosten zu finden. Zur Steuerung der Suche wird eine *heuristische Funktion*  $\hat{f}$  definiert, die eine Schätzung der Kosten  $f(k)$  des günstigsten Lösungspfades durch  $k$  liefert. Die Kosten  $f(k)$  setzen sich zusammen aus der Summe der minimalen Kosten  $g(k)$  eines Pfades vom Startknoten zum Knoten  $k$  und den minimalen Kosten  $h(k)$  eines Pfades von  $k$  zu einem Zielknoten. Die geschätzten Kosten  $\hat{f}(k)$  werden entsprechend als Summe der Schätzwerte definiert

$$\hat{f}(k) = \hat{g}(k) + \hat{h}(k)$$

wobei  $\hat{g}(k)$  die vorläufigen Kosten und  $\hat{h}(k)$  die für die zukünftige Suche geschätzten Restkosten bezeichnet. Die geschätzten vorläufigen Kosten  $\hat{g}(k)$  für einen Knoten  $k$  ergeben sich durch Addition der Kosten  $d(k', k)$  zu den vorläufigen Kosten des Vorgängerknotens  $k'$ .  $\hat{h}(k)$  wird auch Restschätzung genannt und repräsentiert die heuristische Information über die zukünftige Suche. Die Restschätzung hat folgende Eigenschaften:

1. Informativität:  $\hat{h}_1(k)$  ist *informierter* als  $\hat{h}_2(k)$ , falls für alle Knoten  $k$   $\hat{h}_1(k) \geq \hat{h}_2(k)$  gilt.

2. Monotonie:  $\hat{h}(k)$  ist monoton, falls für alle benachbarten Knoten  $k \prec k'$  stets  $\hat{h}(k) - \hat{h}(k') \leq d(k, k')$  gilt.
3. Optimismus:  $\hat{h}(k)$  ist optimistisch, falls  $\hat{h}(k)$  immer eine untere Schranke der tatsächlichen Restkosten  $h(k)$  ist.

Die Restschätzung bedingt die Effizienz des Algorithmus. Je informierter die Restschätzung ist, desto weniger Knoten müssen während der Suche expandiert werden. Ist  $\hat{h}(k) = 0$ , so ist die Suche *uninformiert*, und es müssen alle Knoten während der Suche expandiert werden. Da bei der Suche auch suboptimale Pfade expandiert werden können, werden Pfade mit gleichem Endknoten während des Suchvorgangs rekombiniert. Wie bei der *Dynamischen Programmierung* kommen nur kostenoptimale Teilpfade als Teile des Lösungspfades in Frage, wodurch der Suchraum erheblich verkleinert wird. Die geordnete Suche mit der heuristischen Funktion  $\hat{h}(k)$  und einem Mechanismus zur Rekombination konvergierender Pfade wird als A\*-Algorithmus bezeichnet. Der Algorithmus ist im Folgenden dargestellt (Algorithmus 2.3).

---

```

repeat
   $Q \leftarrow K_\alpha$ 
   $\mathcal{G} \leftarrow \emptyset$ 
  Entferne aus  $Q$  den bestbewerteten Knoten  $k$  und bringe ihn nach  $\mathcal{G}$ 
  if  $k \in K_\omega$  then
    gib  $k$  als Lösung aus
  end if
  for all  $k' \in K$  and  $k \prec k'$  do
    Berechne  $\hat{g}(k'), \hat{h}(k')$ 
  end for
  if  $k' \notin Q \cup \mathcal{G}$  then
    sortiere  $k'$  in  $Q$  ein
  else
    korrigiere die vorläufigen Kosten  $\hat{g}(k^n)$  aller betroffenen Knoten  $K^n \in Q$ 
  end if
until

```

---

Algorithmus 2.3: A\*-Algorithmus

Verfügt der A\*-Algorithmus über eine monotone und optimistische Restschätzung  $\hat{h}(k)$ , so findet er zuverlässig und effizient den global kostengünstigsten Pfad. Die

Bestimmung einer geeigneten Schätzfunktion für die Restkosten ist im Bereich der Spracherkennung und für die *Unit Selection* im Bereich der Sprachsynthese kein triviales Problem.

### 2.2.2 Kontextuelle Klassifikation

Eine auf kontextueller Klassifikation basierende Methode zur Einheitenwahl wurde von [NH88, Nak94, WCIS93] vorgeschlagen. Die Datenbank wird in Äquivalenzklassen (*Cluster*) von Phonemrealisierungen (Allophonen) unterteilt, wobei für jedes Phonem ein Entscheidungsbaum aufgebaut wird. Die Definition der Äquivalenzklassen im Entscheidungsbaum erfolgt durch Merkmale des phonetischen und prosodischen Kontextes, wodurch kontextuelle und koartikulatorische Effekte modelliert werden. Sprachsynthesysteme auf der Basis von kontextueller Klassifikation wurden bei IBM [DE98] und an der Cambridge Universität [DW99] entwickelt. Das Festival Sprachsynthesystem [BT97a] verwendet ebenfalls kontextuelle Klassifikation für die *Unit Selection* Komponente.

Das Verfahren besteht darin, zu den entsprechenden Zeitpunkten durch Entscheidungen eine Klassifikation des Einheiteninventars  $Q = \{s_1, \dots, s_N\}$  vorzunehmen. Als Entscheidungsgrundlage dient die Symbolfolge  $O$  zum Zeitpunkt  $t$ . Der Zustand  $q_t$  kann dann klassifiziert werden anhand der Regel<sup>5</sup>

$$P(q_t = j \mid O, \lambda) = \frac{P(O, q_t = j \mid \lambda)}{P(O \mid \lambda)}$$

Die kontextuelle Klassifikation erfolgt mit Hilfe von Entscheidungsbäumen (*decision trees*). Ein Entscheidungsbaum ist ein Baum, dessen Knoten kontextuelle Fragen repräsentieren, die eine Dimensionierung des Merkmalsraums in Äquivalenzklassen ermöglichen. Zur Klassifikation können symbolische kategorielle Merkmale oder numerische Werte verwendet werden. Klassifikationsbäume verwenden kategorielle Attribute, Regressionsbäume verwenden Mittelwerte und Standardabweichungen, um Äquivalenzklassen zu bilden. Eine Menge von Fragen oder Merkmalstests, die die Daten in Äquivalenzklassen unterteilen, wird als Klassifikator bezeichnet. Um einen Entscheidungsbaum zu generieren, wird eine Methode zur Bewertung des Klassifikators benötigt. Wie beim Trainieren von *Hidden Markov Modellen* wird als Methode *Estimation Maximization* verwendet. Ein Maß für den Informationsgehalt

<sup>5</sup> Häufig kann die Klassifikation nicht den globalen Kontext berücksichtigen, sondern nur die Beobachtungsfolge bis zum Zeitpunkt  $t$  als Entscheidungsgrundlage verwenden.

des Klassifikators ist die Entropie. Sie dient als Kriterium zur Unterteilung des Merkmalsraums und zur Bestimmung des geeignetsten Merkmalstests (*splitting criterion*). Ein Merkmalstest ist geeignet, wenn die Differenz zwischen der Entropie des Klassifikators und der Entropie des Klassifikators bei Hinzunahme des Tests am größten ist. Es wird also eine Minimierung der verbleibenden Entropie angestrebt. Für die *Unit Selection* Komponente des Festival Sprachsynthesystems wurde als *splitting criterion* zur Klassifikation des akustischen Sprachinventars ein akustisches Distanzmaß definiert, das die “Unreinheit” (*Impurity*) eines *Clusters* bestimmt. Durch Partitionierung des *Clusters* wird eine Minimierung der verbleibenden *Impurity* angestrebt ([BT97a], vgl. Abschnitt 3.1.3).

Es wird eine Menge von Trainingsdaten ausgewählt, die durch ein geeignetes Repräsentationsmodell den Merkmalsraum optimal abdecken.<sup>6</sup> Eine iterative Trainingsprozedur wird definiert, die “gierig” (*greedy*) den geeignetsten Klassifikator auswählt, um den Merkmalsraum optimal zu dimensionieren. Eine optimale Dimensionierung maximiert den Informationsgewinn. Entscheidungsbäume werden in der Regel generiert, indem zunächst ein großer Baum aufgebaut wird. Wird die Trainingsprozedur unkontrolliert auf den Trainingsdaten ausgeführt, resultiert dies in einer Unterteilung der Daten bei geringem oder keinem Informationsgewinn. Beim sogenannten *Overfitting* beruht die Klassifikation der Trainingsdaten auf zufälligen Merkmalen. Es werden Besonderheiten gelernt statt zu generalisieren. Dies führt bei neuen ungesehenen Testdaten zu Fehlern. Um das Wachstum des Entscheidungsbaums zu kontrollieren, wird ein Abbruchkriterium (*stopping criterion*) benötigt. Es gibt viele verschiedene Abbruchkriterien in der Literatur [Jel97].<sup>7</sup>

Eine effektive Methode, um *Overtraining* zu vermeiden, ist die Cross-Validierung. Cross-Validierung ist eine Evaluierungsmethode, die auf unabhängigen Testdaten die durch Generalisierung des Modells bedingte Fehlerrate überprüft. Sie kann verwendet werden, um ein optimales Modell aus verschiedenen Modellen auszuwählen. Bei der Generierung von Entscheidungsbäumen kann sie dazu verwendet werden, anzuzeigen, ob ein weiteres Wachstum “hilfreich” ist. Knoten des Baums werden abgeschnitten (*Pruning*), wenn dadurch die Fehlerrate auf unabhängigen Testdaten sinkt. Macon, Cronk und Wouters [MCW98] verwenden für das bei OGI entwickelte Sprachsynthesystem Cross-Validierung bei der Baumgenerierung zur Optimierung

<sup>6</sup> Die Auswahl der Trainingsdaten und des Repräsentationsmodells ist eine schwierige Aufgabe. Häufig kann eine optimale Abdeckung des Merkmalsraums nur approximiert werden.

<sup>7</sup> Das Fehlen eines theoretisch fundierten *Stopping Criterion*s ist eines der Nachteile des Entscheidungsbaum-Konzeptes [Jel97].

des *Stopping Criteria*. Sie beobachten, dass ihre Methode größere Cluster mit geringerer objektiver Distanz hervorbringt.

## 2.3 Distanzmaße und Gewichte

Im vorangehenden Abschnitt wurden die klassischen Suchalgorithmen für *Unit Selection* vorgestellt. Im Folgenden sollen die Kostenfunktionen betrachtet werden, die die Suche steuern. Um geeignete Kostenfunktionen für die Suche zu bestimmen, müssen folgenden Fragen beantwortet werden:

- Welche kontextuellen Merkmale sollen für die Berechnung der Target- und Konkatenierungskosten verwendet werden?
- Wie sollen die Kosten relativ gewichtet werden? Welche Verfahren stehen für ein automatisches Training der Gewichte zur Verfügung?
- Welche Distanzmaße sollen beim Training verwendet werden? Wie ist das Verhältnis von objektiver und perzeptueller Distanz?

Welche Merkmale für die Einheitenwahl verwendet werden sollen, hängt von verschiedenen theoretischen und praktischen Aspekten ab. Das zugrundeliegende Sprachmodell beeinflusst die Auswahl der Merkmale, wie die Diskussion um die Verwendung akustischer vs. phonologischer Merkmalsbeschreibungen zeigt [BJ98, TB97, Tay00] (Abschnitt 2.4). Ebenso spielen die Art und die Aufbereitung der Sprachdaten und der eingesetzte Auswahlalgorithmus für die Definition eines Merkmalssets eine Rolle. Generell sollte ein Merkmalsset möglichst viele variable Eigenschaften des Sprachsignals beschreiben und vorhersagen. Es sollte darüber hinaus robust und kompakt sein und keine redundanten Merkmale beinhalten.

Da die Datenbank nicht alle Realisierungen von Sprachlauten beinhalten kann, führt jede Abstraktion durch eine Merkmalsbeschreibung zu einer Unterspezifikation. Daher müssen die Merkmale relativ gewichtet werden. Kriterien für die Gewichtung von Merkmalen können durch Klassifikation akustischer Daten, durch empirische Beobachtung und linguistische Theorien sowie durch Beschränkungen auf Signalverarbeitungsebene gewonnen werden.

Die relative Gewichtung der akustischen Merkmale ist für die Auswahl der geeigneten Kandidaten während der Suche von großer Bedeutung. Die schlecht ausbalancierte Gewichtung segmenteller und prosodischer Merkmale führte bei einer frühen

Version des CHATR Systems (Abschnitt 3.1.2) zu partiell unverständlicher Sprachausgabe. Der Auswahlalgorithmus gab die segmentelle Identität zugunsten prosodischer Merkmale auf, wenn letztere die Targetbeschreibung sehr gut approximierten.

### **Gewichtetraining**

Automatisch trainierte Gewichte liefern bei der Synthese eine bessere Sprachqualität als von Hand gesetzte [HB96]. Für das automatische Training der Gewichte gibt es in der Literatur zwei verschiedene Ansätze: *weight space search* und *multiple linear regression training*. Bei der *weight space search* [BC95] wird eine limitierte Suche zur Bestimmung der optimalen Gewichte durch ein *analysis-by-synthesis* Verfahren durchgeführt. Beim Training werden die optimalen Gewichte für die Kostenfunktionen bestimmt, indem der Unterschied zwischen dem natürlichen und dem synthetisierten Sprachsignal für eine gegebene Targetäußerung minimiert wird. Die Targetäußerung wird aus den vom Auswahlalgorithmus gefundenen Einheiten synthetisiert und die akustische Distanz zur Originaläußerung gemessen. Dieser Prozess wird anhand der Äußerungen im Trainingsset über variierende Gewichte iteriert, bis das global beste Set an Gewichten für die Merkmale gefunden wird. Nachteil dieses Verfahrens ist der sehr hohe Rechenaufwand.

Multiple lineare Regression kann eingesetzt werden, um die Gewichte für die Target- und Konkatenierungskosten zu bestimmen. [HB96] betrachten die Kostenfunktionen getrennt. Während für die Konkatenierungskosten *weight space search* eingesetzt wird, werden die Targetkosten durch *multiple linear regression* optimiert. Bei dem Verfahren wird ein exhaustiver Vergleich der Einheiten in der Datenbank anhand eines objektiven Distanzmaßes durchgeführt. Die objektive Distanz wird durch eine lineare Gewichtung der vorberechneten Kosten vorhergesagt. Der Vorteil dieses Verfahrens liegt in der Effizienz und der Flexibilität. Es erlaubt für verschiedene Phonetypen oder -klassen wie beispielsweise die Gruppe der Nasale die Verwendung unterschiedlicher Gewichte.

### **Distanzmaße**

Der Unterschied zwischen der natürlichsprachlichen und der synthetisierten Äußerung wird durch ein objektives Distanzmaß bestimmt. Black und Campbell [BC95] verwenden als objektives Distanzmaß die euklidische cepstrale Distanz. Dieses Distanzmaß weist eine höhere Priorität für *unit distortion* gegenüber *continuity distortion* auf. Mit Hilfe von Perzeptionstests zeigen [BC95], dass Hörer fließende Über-

gänge an Konkatenierungsstellen bevorzugen. Das cepstrale Distanzmaß berechnet den mittleren Fehler für jeden Zeitpunkt des Signals. Diskontinuitäten an der Verkettungsstelle fallen bei der Berechnung der Konkatenierungskosten nicht ins Gewicht. Hörer empfinden jedoch solche Diskontinuitäten ebenso störend wie prosodische Fehler. Ein objektives Distanzmaß muss daher die perzeptuelle Abnahme der Sprachqualität durch hörbare Diskontinuitäten entsprechend reflektieren. Eine Gewichtung der Konkatenierungskosten relativ zu den Targetkosten muss durch Perzeptionstests bestimmt werden.

Das Verhältnis der objektiven zur perzeptuellen Distanz wird vorwiegend im Bereich Sprachcodierung thematisiert. In der Sprach- und Sprechererkennung wird die spektrale Distanz beim Vergleich von Sprachmustern gemessen. Aktuelle Ansätze in der Spracherkennung verwenden häufig die euklidische Distanz von *Mel-Frequency Cepstral Coefficients* (MFCC) als objektives Distanzmaß. Es gibt weitere Distanzmaße wie die gewichtete euklidische cepstrale Distanz oder die Kullback-Leibler Distanz. Untersuchungen zeigen, dass die euklidische Distanz von MFCCs und die Kullback-Leibler Distanz von LPC-basierten Energiespektren eine hohe Korrelation zur perzeptuellen Distanz aufweisen [SS01]. Jedoch gibt es bislang keine ausreichend zuverlässigen Prädiktoren für die perzeptuelle Qualität einer synthetisierten Äußerung [SS01, Möb00].

Für die Berechnung der Konkatenierungskosten anhand eines akustischen Distanzmaßes wird häufig eine *optimal coupling*-Technik eingesetzt [CI97]. Bei dieser Technik wird die optimale Konkatenierungsstelle zwischen den Einheiten durch Minimierung der akustischen Distanz berechnet. Die verschiedenen Methoden unterscheiden sich durch den Einsatz des akustischen Distanzmaßes und des Vergleichsverfahrens wie der Regressionsanalyse oder der *linear fit*-Methode. Insbesondere für die Verkettung von phonembasierten Einheiten, deren Segmentgrenzen in den spektral instabilen Signalabschnitten liegen, ist diese Methode von Vorteil. Durch das *optimal coupling* kann eine Position in einem stabilen Signalabschnitt zur Mitte des Phonems hin als Konkatenierungspunkt gewählt werden. Der Suchbereich wird beim *optimal coupling* in das vorhergehende Segment ausgedehnt. Wie bei der Diphonsynthese können dadurch die Lautübergänge zwischen zwei Phonemen implizit modelliert werden.

Da die Berechnung der Konkatenierungskosten während der Laufzeit sehr rechenintensiv ist und erheblich zum Zeitaufwand der Synthese beiträgt, ist es wünschenswert die Anzahl und Komplexität der Berechnungen zu reduzieren. Ansätze, die Laufzeit der *Unit Selection* zu reduzieren, konzentrieren sich auf zwei Aspek-

te: Durch die Limitierung der Anzahl der Kandidaten, die während der Einheitenauswahl zur Verfügung stehen, kann die Anzahl der Berechnungen verringert werden. Die Komplexität kann durch eine *offline*-Kostenberechnung vermindert werden. Beutnagel, Mohri und Riley [BMR99] schlagen die Konstruktion eines *offline*-Caches für die Konkatenierungskosten vor. Aufgrund der hohen Anzahl aller Einheitenkombinationen wird bei der Konstruktion des Caches nur eine Teilmenge von Einheitenpaaren berücksichtigt. Es konnte eine Komplexitätsreduktion um einen Faktor 4 erzielt werden ohne signifikante Minderung der Sprachqualität [CBSB00]. Bei einem Cache mit 1.2 Mio. Einheitenpaaren konnte eine Abdeckung von 99% erreicht werden. Anhand von Experimenten konnte gezeigt werden, dass die synthetisierten Äußerungen zu 98.2% identisch zu Äußerungen sind, die unter Verwendung des vollständigen Einheiteninventars erzeugt wurden [BMR99].

Ein Verfahren, bei dem die Targetkosten *offline* berechnet werden können, wurde in dem Festival Sprachsynthesystem [BT97a] (Abschnitt 3.1.3) implementiert. Durch kontextuelle Klassifikation mit Entscheidungsbäumen wird die Sprachdatenbank strukturiert und durch *Pruning* die Anzahl der Einheiten (optional) reduziert, um die Laufzeit für die Einheitenauswahl während der Synthese zu verringern. Beide Ansätze bringen eine Minderung der Sprachqualität mit sich, wenn die Anzahl der für die Einheitenauswahl zur Verfügung stehenden Kandidaten zu stark limitiert wird. Die Größe des Einheiteninventars muss daher beim Korpusdesign sorgfältig abgewogen werden.

## 2.4 Phonetische und phonologische Bäume

Frühe Systemrealisierungen wie das  $\mu$ -Talk System (Abschnitt 3.1.1) verfügen über ein hohes Maß an notwendiger Signalverarbeitung. Grund hierfür ist die Verwendung von akustischen Merkmalen für die Einheitenauswahl. Für das Nachfolgesystem CHATR (Abschnitt 3.1.2) wurde neben phonetischen Merkmalen auch der prosodische Kontext als Auswahlkriterium berücksichtigt, um das Maß an Signalverarbeitung zu verringern und die Sprachqualität zu verbessern.

Breen und Jackson [BJ98] vertreten den theoretischen Standpunkt, dass Syntheseansätze, die auf der akustischen Beschreibung von Sprache beruhen, nicht ausreichend berücksichtigen, dass das Sprachsignal Manifestation eines strukturierten kommunikativen Prozesses ist. Dieser Prozess unterliegt einer vorhersagbaren Variabilität, die es ermöglicht ein Sprachsignal durch ein geeignetes Set an abstrakten

Merkmale ausreichend zu beschreiben.

Als abstrakte strukturelle Repräsentation auf symbolischer Ebene können phonologische Bäume [BJ98, TB97, Tay00] dienen. Diese beinhalten linguistische und phonologische Merkmale sowie strukturelle Informationen, durch die wesentliche linguistische Relationen dargestellt werden können. Phonologische Repräsentationen sind nach Taylor und Black [TB97, Tay00] aus folgenden Gründen vorzuziehen:

- Phonologische Repräsentationen sind kompakter und beinhalten keine redundanten Merkmale. Die Größe des Merkmalsraums ist geringer. Durch die Verwendung kompakter Merkmale kann daher der Suchraum bei der *Unit Selection* stärker eingeschränkt werden.
- Phonologische Repräsentationen beinhalten weniger Fehler aufgrund der geringeren Fehleranfälligkeit der Module. Bei Sprachsynthesystemen liegt häufig eine *Pipeline*-Architektur vor. Fehlanalysen lösen Folgefehler aus, die sich während der Verarbeitung potenzieren und eine stark abnehmende Signalqualität zur Folge haben [Möb01].
- Im Gegensatz zu phonologischen Repräsentationen beinhalten phonetische Darstellungen häufig perzeptuell unnötig Spezifikationen.

Die strukturelle und inhaltliche Repräsentation phonologischer Bäume hängt von dem zugrundeliegenden Sprachmodell und der Verarbeitungseffizienz ab. [TB97, Tay00] verwenden metrische Binärbäume, räumen jedoch ein, dass andere Repräsentationen geeigneter sein können.

## 2.5 Korpusdesign

Frühe konkatenative Synthesysteme verwendeten eine kleine Anzahl an Einheiten als Syntheseinventar, üblicherweise ein Token für jedes Phonem. Solche Systeme sind sehr effizient im Hinblick auf Speicherplatz und Rechenzeit. Durch die Entwicklungen im Speicher- und Prozessorbereich und die zunehmende Verfügbarkeit von großen Sprachkorpora können ganze Sprachdatenbanken als Inventar für die konkatenative Synthese verwendet werden. Da das Sprachinventar sehr großen Einfluss auf die Qualität der synthetisierten Sprachausgabe hat, muss bei der Verwendung eines Sprachkorpora eine sorgfältige Auswahl und Aufbereitung der Daten vorgenommen werden. Dabei spielen folgende Aspekte für das Korpusdesign eine Rolle:

- Welche (maximale) Größe kann / darf die Sprachdatenbank haben?
- Wie kann der Abdeckungsgrad definiert und bestimmt werden? Welchen Abdeckungsgrad muss die Sprachdatenbank erreichen?
- Welche Sprachkorpora kommen als Inventar in Frage? Welche Sprechstile soll die Sprachdatenbank beinhalten?

Die Größe von Sprachdatenbanken war lange Zeit durch die maximal verfügbare Speicherkapazität beschränkt. Dies ist heute beim Einsatz sprachlicher Kommunikationsschnittstellen in mobilen Kommunikationsgeräten wie PDAs und Handys der Fall; im PC- und Server-Bereich spielen die Anforderungen an den Speicher durch große Datenbanken keine Rolle mehr. Große Sprachdatenbanken sind jedoch schwer aufzubereiten und zu warten. Beim Erstellen einer Sprachdatenbank für *Unit Selection* muss die optimale Größe des Korpus bestimmt werden. Bei der Entwicklung der ersten Systeme wurde die erforderliche Größe einer *Unit Selection* Datenbank auf 40 Minuten für das Englische bzw. 20 Minuten für das Japanische geschätzt [Möb00]. Das im Abschnitt 3.1.2 dargestellte Sprachsynthesystem CHATR [BT94] verfügt über verschiedene japanisch- und englischsprachige Datenbanken, die zwischen 10 und 150 Minuten Sprachaufnahmen von isolierten Wörtern und Radionachrichten enthalten. Das IBM Sprachsynthesystem [DE98] wurde anhand von einem 45 Minuten großen Sprachkorpus trainiert. Das Synthesystem der Cambridge Universität [DW99] verwendet ein Korpus von 60 Minuten gesprochener Sprache. Conkie und die Forschergruppe bei AT&T [Con99] beobachten, dass bei Verwendung einer größeren Anzahl von Spracheinheiten in der Datenbank die Qualität der Sprachausgabe deutlich verbessert werden konnte. Ein Bedarf an größeren Sprachdatenbanken wird mehrfach in der Fachliteratur konstatiert [Möb00].

Die Frage nach der Größe einer Sprachdatenbank muss im Zusammenhang mit dem Abdeckungsgrad betrachtet werden. Wieviele und welche Phoneme, Silben, Wörter und Sätze muss ein Sprachkorpus enthalten, um eine ausreichende Abdeckung der Lautstruktur einer Sprache zu erreichen? Van Santen [vS97a] hat eine systematische Untersuchung des Abdeckungsgrades einer Sprachdatenbank vorgenommen. Er legt für Diphon-Einheiten einen Kontextmerkmalsvektor fest, der prosodische Merkmale beinhaltet wie Wortakzent und Position innerhalb der Äußerung. Der Abdeckungsgrad (*coverage index*) wird definiert als die Wahrscheinlichkeit, dass die Diphon-Merkmalvektoren eines zufällig ausgewählten Testsatzes in einem Trainingsset eines Korpus auftreten. Als Ergebnis erhält van Santen für ein Trainingsset

von 25.000 Merkmalsvektoren einen Abdeckungsgrad von 0.03; für ein Trainingsset von 150.000 Merkmalsvektoren einen Abdeckungsgrad von 0.75. Die Wahrscheinlichkeit, dass ein Trainingsset alle in einem Testsatz vorkommenden Diphon-Merkmalsvektoren abdeckt, ist somit sehr gering. Eine Ursache dafür ist die LNRE-Charakteristik<sup>8</sup> der Sprache, die den Abdeckungsgrad von akustischen Inventaren beeinflusst [Möb03]. Für Sprachkorpora, deren Genre oder Domäne von der des Textsatzes abweichen, ergeben sich nach [vS97a] noch schlechtere Werte. Dies legt nahe, dass eine ausreichende Abdeckung durch ein Sprachkorpus aufgrund der Komplexität und Kombinatorik der Sprache sowie der LNRE-Charakteristik insbesondere für unbeschränkte Anwendungsdomänen nicht möglich ist. Um eine möglichst optimale Abdeckung zu erzielen, muss die Sprachdatenbank sorgfältig anhand linguistischer und phonetischer Kriterien erstellt werden und an den jeweiligen Anwendungsbereich angepasst sein.

Bei der Entwicklung moderner *Unit Selection* Systeme wird ein spezielles Datenbank-Design verfolgt. Die Sprachdatenbank soll alle relevanten akustischen Realisierungen von Phonemen (Allophone) abdecken und “phonetisch reichhaltiges” domänenspezifisches Sprachmaterial enthalten. Bei der Erstellung der Datenbank für das AT&T Next-Gen TTS-System [BC99] wurde die Diphon-Abdeckung sorgfältig kontrolliert. [BC99] verwenden für die Sprachaufnahmen unterschiedliches domänenspezifisches Textmaterial (sowohl Zeitungstext als auch spontan gesprochene Sprache), um neben phonetischen Kontextmerkmalen auch prosodische Kontexte im Sprachkorpus abzudecken. Van Santen und Buchsbaum [vSB97] schlagen als statistisches Analysewerkzeug zur Bestimmung der phonetischen und prosodischen Abdeckung die Verwendung eines *Greedy*-Algorithmus vor. Die ausgewählten Sätze sollen möglichst viele verschiedene akustische Phonemrealisierungen enthalten. Sie sollen außerdem eine reguläre Struktur und eine angemessene Länge aufweisen, um beim Sprechen die erwarteten prosodischen Muster zu erhalten. Der *Greedy*-Algorithmus ermöglicht es, Sätze und Textabschnitte mit optimalem Abdeckungsgrad anhand phonetischer und prosodischer Merkmale zu selektieren. Redundantes Textmaterial kann entfernt werden, um die Datenbank kompakt und klein zu halten. Da kleine Datenbanken verlässlicher segmentiert und annotiert werden können, liefern sie eine bessere Synthesequalität. Die Performanz des *Unit Selection* Algorithmus hängt stark von der Auswahl der linguistischen und phonetischen Merkmale zur An-

---

<sup>8</sup> LNRE-Verteilungen (*large number of rare events*) weisen Klassen auf, die sehr viele Elemente mit geringer Wahrscheinlichkeit beinhalten.

notation der Datenbank und von der Verlässlichkeit der Korpusaufbereitung ab.

Die Auswahl des Sprechstils beim Korpusdesign hat ebenso wie das Textgenre und die Textdomäne großen Einfluss auf die Sprachsynthese. Verschiedene Sprechstile in der Datenbank können bei der Synthese in einem “Patchwork” von Einheiten mit unterschiedlichen prosodischen Eigenschaften resultieren [BJ98]. Dies ist möglich, wenn die verschiedenen Sprechstile nicht ausreichend spezifiziert und annotiert sind. Eine Kontrolle des Sprechstils ist vor allem bei großen Sprachdatenbanken sehr schwierig. Die meisten Synthesysteme verwenden Sprachdaten mit neutralem Sprechstil. Unterschiedliche Sprechstile können jedoch für beschränkte Anwendungsdomänen mit bekannten linguistischen Merkmalsverteilungen sinnvoll sein, wenn die Datenbank entsprechend spezifiziert und annotiert ist und der Auswahlalgorithmus Sprechstil als Auswahlkriterium verwendet [BJ98, Möb00].<sup>9</sup>

Auf *Unit Selection* basierende Sprachsynthesysteme sind in der Lage, von natürlicher Sprache kaum unterscheidbare gesprochene Sprachausgaben zu produzieren. Ein Problem dieses Verfahrens ist, dass natürlich klingende Sprachabschnitte durch meist kleinere, nicht passende Einheiten unterbrochen werden. Diese entsprechen nicht den Targetanforderungen und verursachen Signalstörungen an den Konkatenierungsstellen. Ziel der *Unit Selection* Forschung ist es daher, eine konsistent hohe Sprachqualität bei der Synthese zu erzielen. Die Forschungsansätze konzentrieren sich auf die Entwicklung von optimierten Auswahlalgorithmen, die Definition von geeigneten linguistischen und phonetischen Auswahlkriterien, die Verwendung von perzeptuell relevanten Distanzmaßen und die Verbesserung des Korpusdesigns. Im Folgenden sollen einige Forschungsansätze für *Unit Selection* anhand verschiedener Sprachsynthesysteme vorgestellt und diskutiert werden, auf denen die hier vorgestellte Arbeit aufbaut.

---

<sup>9</sup> Im Bereich emotionale Sprachsynthese werden anhand des Sprechstils verschiedene Emotionen wie Angst, Freude, Trauer oder Langeweile simuliert.

## 3 Unit Selection Systeme

Eine Vielzahl von unterschiedlichen *Unit Selection* Verfahren werden derzeit in verschiedenen Systemen eingesetzt. Obwohl sich die Systeme hinsichtlich der verwendeten Techniken unterscheiden, können allgemeine Kriterien formuliert werden, die einen Vergleich und eine Einschätzung der Verfahren erlauben.<sup>1</sup> Für den Entwurf eines *Unit Selection* Systems spielen folgende Aspekte eine Rolle:

- Das Verfahren muss effizient bezüglich Rechenzeit und Speicheranforderungen sein. Der verfügbare Speicherplatz ist abhängig von der einsetzbaren Hardware. Die Anforderungen an die Rechenzeit ergeben sich aus der nötigen Responsezeit des Systems, die beispielsweise im Fall von Dialogsystemen nur wenige Millisekunden bis Sekunden beträgt.
- Das Verfahren muss möglichst großen Nutzen von den zur Verfügung stehenden Sprachdaten machen und die optimale Kombination von Spracheinheiten aus dem Korpus finden. Datenbank und Auswahlalgorithmus müssen aufeinander und auf die Anwendungsdomäne abgestimmt sein.
- Das Verfahren muss eine optimale Sprachqualität für die Anwendungsdomäne liefern. Es sollte eine skalierbare Synthesequalität aufweisen und eine graduelle Reduktion der Synthesequalität ermöglichen, wenn die gewünschten Sprachdaten nicht im Korpus vorhanden sind.

Eine hohe Sprachqualität kann nur erreicht werden, wenn der Auswahlalgorithmus einen optimalen Nutzen von den Sprachdaten macht. Ein spezielles Korpusdesign und die Abstimmung von Auswahlalgorithmus und Sprachkorpus aufeinander führt bei Synthesystemen für eingeschränkte Anwendungsdomänen zu einer erheblichen Verbesserung der Sprachqualität. Synthesysteme für unbeschränkte Anwendungsdomänen verfügen i.d.R. über eine geringere und stärker variierende Synthesequalität, sind jedoch robuster gegenüber unbekanntem Textmaterial.

---

<sup>1</sup> Eine Evaluation der Synthesequalität der Systeme kann nicht erfolgen, da keine vergleichbaren Daten zur Bewertung der Sprachqualität vorliegen.

### 3.1 Unbeschränkte Anwendungsdomänen

Obwohl die meisten Einsatzgebiet für Sprachsynthesysteme über eine eingeschränkte Anwendungsdomäne verfügen<sup>2</sup>, wurde seit Beginn der Sprachsyntheseforschung die Synthese von unbeschränktem Text als einziges Forschungsziel betrachtet [Tay00]. Während die Entwicklung in der Spracherkennung von einer stark eingeschränkten Anwendungsdomäne (sprecherabhängige Einzel-Wort-Erkennung) zu schwierigeren Anwendungen (sprecherunabhängige kontinuierliche Spracherkennung mit großen Vokabular) übergang, wurde in der Sprachsynthese versucht, die Performanz für eine unbeschränkte Synthese kontinuierlich zu verbessern. Taylor [Tay00] stellt daher die Frage, ob durch eine anfängliche Einschränkung der Domäne und eine inkrementelle Steigerung des Schwierigkeitsgrades der Anwendung bei gleichbleibender Performanz die Entwicklung im Bereich der Sprachsynthese beschleunigt werden kann. Abbildung 3.1 zeigt die beiden Entwicklungsrichtungen der Sprachsyntheseforschung.

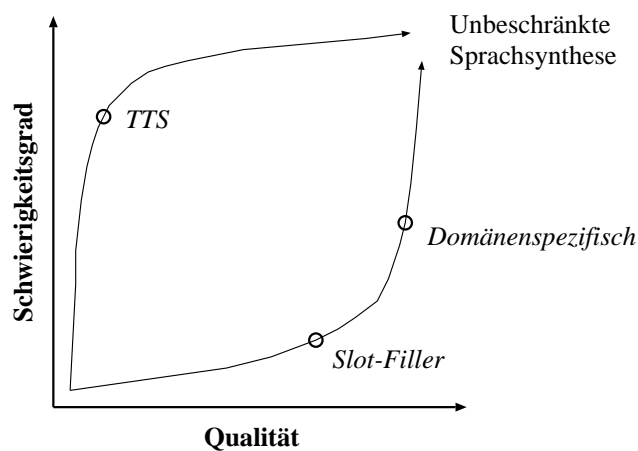


Abbildung 3.1: Entwicklungsrichtungen in der Sprachsynthese [Tay00]

Die im diesem Abschnitt vorgestellten Systeme folgen dem traditionellen Forschungsziel der unbeschränkten TTS-Synthese. In Abschnitt 3.2 werden Synthesysteme für eingeschränkte Anwendungsdomänen diskutiert.

<sup>2</sup> Eine unbeschränkte Domäne stellen Nachrichtentexte und E-Mails dar, für die Synthesysteme als Vorlesegeräte eingesetzt werden können.

### 3.1.1 Das $\mu$ -Talk Sprachsynthesystem

Das erste auf *Unit Selection* basierende Synthesesystem ist das bei ATR entwickelte CHATR-TTS-System [BT94] (Abschnitt 3.1.2). Bei dem Entwurf des Vorgängersystems  $\mu$ -Talk wurden bereits wichtige Prinzipien wie die Unterscheidung von *unit distortion* und *continuity distortion* berücksichtigt, die von Sagisaka und anderen in früheren Arbeiten vorgeschlagen wurden [Sag88, TKS90]. Die Einheitenwahl des  $\mu$ -Talk Systems erfolgt durch Minimierung der Kosten, die sich durch *unit distortion* und durch *continuity distortion* ergeben. Die intersegmentelle spektrale Distanz wird zwischen zwei aufeinanderfolgenden Segmenten an der Konkatenierungsstelle berechnet. Das Maß der spektralen Signaländerungen an den Segmentgrenzen dient als Prädiktor für die Abnahme der Signalqualität an der Konkatenierungsstelle und liefert den idealen Schnittpunkt der ausgewählten Einheiten. Das Maß an *unit distortion* wird durch die Prototypizität einer ausgewählten Einheit bestimmt. Diese wird durch kontextuelle Klassifikation anhand der euklidischen Distanz zwischen dem Zentroiden eines Segmentclusters in verschiedenen Triphonkontexten berechnet. Die kontextuelle Klassifikation ist auf Vokalspektren eingeschränkt.<sup>3</sup> Die Minimierung der Kostenfunktionen erfolgt durch Dynamische Programmierung. Die optimale Einheitenfolge wird durch Minimierung der globalen Kosten der Targetäußerung bestimmt. Die ausgewählten Segmente werden an der vorher berechneten Schnittstelle konkateniert, wobei zur Glättung von spektralen Diskontinuitäten Signalverarbeitungstechniken eingesetzt werden.

Das  $\mu$ -Talk System verfügt über ein hohes Maß an notwendiger Signalverarbeitung. Dies ist zum Teil bedingt durch die Spezifikation von Target- und Kandidatensegmenten auf akustischer (cepstraler) Ebene. Es werden nur phonetische, keine prosodischen Merkmale bei der Einheitenwahl betrachtet. Ein weiterer Nachteil des in  $\mu$ -Talk eingesetzten Verfahrens ist der hohe Rechenaufwand, der sich bei der Berechnung der optimalen Einheitensequenz durch Dynamische Programmierung ergibt.

---

<sup>3</sup> Die Beschränkung der kontextuellen Klassifikation auf Vokale in Triphonkontexten ist aufgrund der phonotaktischen Struktur des Japanischen möglich.

### 3.1.2 Das CHATR Sprachsynthesystem

Wie bei dem Vorgängersystem  $\mu$ -Talk (Abschnitt 3.1.1) beruht die Synthesekomponente von CHATR [BT94] auf einem *Unit Selection* Verfahren. Der Ansatz von  $\mu$ -Talk wurde erweitert, indem für die Einheitenwahl neben dem phonetischen auch der prosodische Kontext berücksichtigt wird. Die Hinzunahme von prosodischer Information als Auswahlkriterium dient dazu, das Maß an Signalverarbeitung zu verringern und damit die Qualität der Sprachausgabe zu verbessern. Die Sprachdatenbank ist mit multidimensionalen Merkmalsvektoren annotiert, die die phonetischen und prosodischen Kontexteigenschaften beschreiben. Hunt und Black [HB96] betrachten die Einheiten der Sprachdatenbank als ein Zustandsübergangsnetzwerk, dessen Zustandskosten die Targetkosten (*target cost*) und dessen Übergangskosten die Konkatenierungskosten (*concatenation cost*) sind. Die Auswahl der Einheiten erfolgt durch eine HMM-basierte Viterbi-Dekodierung unter Minimierung der beiden Kostenfunktionen.

#### Einheitenwahl

Während der Synthese erzeugt CHATR aus der Texteingabe eine Targetspezifikation, die aus einer mit Merkmalsvektoren annotierten Sequenz von Segmenten besteht. Die Merkmalsvektoren beschreiben wie die der Segmente in der Datenbank den phonetischen und prosodischen Kontext. Sie enthalten numerische akustische Werte, die durch Signalverarbeitungstechniken berechnet werden, sowie diskrete kategoriale Kontextmerkmale. Als phonetischer Kontext wird das vorangehende und das nachfolgende Phonem betrachtet und anhand folgender Merkmale beschrieben: Vokal vs. Konsonant, Stimmhaftigkeit, Konsonanttyp, Artikulationsort, Vokalhöhe, Vokallänge, Vokalkrümmung. Die Vergleichsfunktion (*matching function*) für die diskreten Merkmale zweier Segmente liefert den Wert 0 bei Übereinstimmung und 1, wenn die Merkmale verschieden sind. Die Targetkosten berechnen sich als gewichtete Summe der Unterschiede zwischen den Elementen des Targetvektors  $t_i$  und des Kandidatenvektors  $u_i$ . Für  $p$  Elemente eines Merkmalsvektors berechnen sich die Targetkosten für das  $i$ -te Segment als

$$C^t(t_i, u_i) = \sum_{j=1}^p w_j^t C_j^t(t_i, u_i)$$

wobei  $w_j^t$  die Gewichte für die Elemente  $j = 1, \dots, p$  des Merkmalsvektors sind. Die Konkatenierungskosten werden entsprechend als die gewichtete Summe der  $q$

Subkosten zwischen dem jeweiligen Segment  $u_i$  und dem vorangehenden Segment  $u_{i-1}$  berechnet. Als akustische Merkmale für die Berechnung der Subkosten werden die cepstrale Distanz an der Konkatenierungsstelle und die absolute Differenz im Energiespektrum sowie die absolute Differenz der Grundfrequenz  $f_0$  verwendet. Die Konkatenierungskosten für das  $i$ -te Segment ergeben sich aus

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^q w_j^c C_j^c(u_{i-1}, u_i)$$

Folgen die beiden Segmente  $u_{i-1}$  und  $u_i$  in der Datenbank aufeinander, so sind die Konkatenierungskosten gleich 0.

Die optimale Einheitensequenz in der Datenbank ergibt sich durch Minimierung der Kosten für die  $n$  Segmente der Targetäußerung.

$$\min C(t_1^n, u_1^n) = \sum_{i=1}^n \sum_{j=1}^p w_j^t C_j^t(t_i, u_i) + \sum_{i=2}^n \sum_{j=1}^q w_j^c C_j^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S)$$

$S$  ist Stille.  $C^c(S, u_1)$  und  $C^c(u_n, S)$  sind die Kosten für die Verkettung des ersten und des letzten Segmentes mit dem Stille-Segment.

Die Suche der optimalen Einheitensequenz in dem Zustandsübergangsnetzwerk erfolgt durch eine schrittweise eingeschränkte Strahlsuche (Abschnitt 2.2.1). Das schrittweise *Pruning* der Datenbank ist nötig, um den Rechenaufwand für Echtzeitbedingungen zu reduzieren. Im ersten Schritt werden diejenigen Einheiten in der Datenbank ausgewählt, deren phonetischer Kontext mit dem der Targeteinheit weitestgehend übereinstimmt. Danach wird die Menge der verbleibenden Einheiten anhand der Targetkosten und schließlich anhand der Konkatenierungskosten eingeschränkt. [HB96] zufolge hat das *Pruning* der Datenbank kaum Auswirkungen auf die Qualität der Sprachausgabe.

### Gewichtstraining

Die Kostenfunktionen, die die Einheitenauswahl bei der Viterbi-Suche steuern, können mit statistischen Verfahren trainiert werden. Hunt und Black [HB96] implementierten zwei Verfahren, *weight space search* und *linear regression training* (Abschnitt 2.3).

Bei der *weight space search* wird für eine gegebene Menge von Gewichten die optimale Einheitensequenz für eine Äußerung ausgewählt und das Sprachsignal synthetisiert. Der Abstand der zeitalignierten, re-synthetisierten Äußerung von der natürlichsprachlichen wird mit einer euklidischen cepstralen Distanzfunktion berechnet.

Die Prozedur wird über die Gewichtemenge und mehrere Äußerungen iteriert, bis eine Gewichtekombination gefunden ist, die eine maximale spektrale Ähnlichkeit zur Targetäußerung und eine hohe Signalqualität erzielt. Die Komplexität des Trainingsalgorithmus hängt exponentiell von der Anzahl der Gewichte und den möglichen Werten, die die Gewichte annehmen können, ab. Hunt und Black [HB96] verwenden 3 – 5 verschiedene Werte für die Gewichtung der prosodischen und phonetischen Kontextmerkmale. Bei 10 Testsätzen wurden bis zu mehrere 100.000 Sprachsignale synthetisiert und verglichen.

Meron und Hirose [MH99] schlagen eine effizientere Strategie für das Gewichte-training vor. Sie spalten die iterative Prozedur in drei Schritte für Auswahl, Synthese und Vergleich auf. Die Auswahlprozedur erfolgt für alle Gewichte-Kombinationen, während die Schritte für Synthese und Vergleich nur für unterschiedlich ausgewählte Einheitenfolgen durchgeführt werden. Zerlegt man die Äußerungen in kürzere Einheitenfolgen wie Paare oder Triplets, so ist die Anzahl der unterschiedlich ausgewählten Einheitenfolgen noch geringer und es kann weitere Rechenzeit eingespart werden. Diese kann verwendet werden, um eine größere Anzahl von Gewichten und möglichen Werten zu trainieren und damit eine robustere Sprachqualität zu erzeugen.

Neben der *weight space search* setzen Hunt und Black [HB96] lineare Regression als effizientes Trainingsverfahren ein. Die Bestimmung der Gewichte für die beiden Kostenfunktionen erfolgt getrennt. Für die Konkatenierungskosten werden anhand der iterativen Suchmethode eine Linearkombination von cepstraler Distanz und absoluter Differenz der Energie an der Konkatenierungsstelle als Prädiktor für die Qualität der Verkettung automatisch trainiert. Die Gewichtung der absoluten Differenz der Grundfrequenz erfolgt manuell. Die Targetkosten, die sich aus den phonetischen und prosodischen Kontextmerkmalen berechnen, werden durch mehrfache lineare Regression automatisch gewichtet. Das Regressionstraining besitzt eine größere Flexibilität, indem es eine separate Gewichtung verschiedener Phonemtypen oder -klassen erlaubt, für die der Einfluss der phonetischen und prosodischen Kontextfaktoren unterschiedlich sein kann. Dennoch beobachten Hunt und Black nur eine geringe Verbesserung der Synthesequalität. Der enorme Effizienzvorteil im Vergleich zum *weight space search* Verfahren<sup>4</sup> erlaubt es jedoch, eine größere Anzahl an Gewichten zu trainieren.<sup>5</sup> Dies sowie die Verwendung robusterer Verfahren wie schrittweise lineare Regression können zu einer weiteren Verbesserung der Sprachausgabe füh-

<sup>4</sup> Die Trainingszeit konnte um einen Faktor 100 reduziert werden [HB96].

<sup>5</sup> Die Anzahl der Gewichte beeinflusst die Komplexität des Regressionsverfahrens nur linear.

ren.

Ein alternatives Verfahren für das Regressionstraining wird in [MH99] vorgeschlagen, bei dem die Target- und Konkatenierungskosten simultan trainiert werden können. Ein separates Training der Target- und Konkatenierungskosten ist nach Auffassung von Meron und Hirose nicht optimal, da die beiden Kostenfunktionen korreliert sind. Bei dem eingesetzten Verfahren werden Paare von konkatenierten Einheiten mit natürlichsprachlichen Einheitenpaaren verglichen. Da kein separates Training für unterschiedliche Phonemtypen oder -klassen möglich ist, wird eine datenbasierte phonetische Klassifikation mit Hilfe von Entscheidungsbäumen vorgenommen, um die Kontextsensitivität der Gewichte zu erhöhen. Das Verfahren erlaubt auch, den Einfluss prosodischer Modifikationen während der Synthese zu berücksichtigen. Perzeptuelle Experimente zeigen, dass die Kontextklassifikation der Gewichte die Synthesequalität deutlich verbessert.

Das CHATR TTS-System setzt als erstes Sprachsynthesystem *Unit Selection* ein. Die Einheitenauswahl erfolgt anhand phonetischer und prosodischer Kontextmerkmale. Durch die Hinzunahme von prosodischer Information als Auswahlkriterium wird das Maß an Sprachverarbeitung verringert und die Sprachqualität der Synthese verbessert. Für die Gewichtung der Auswahlparameter werden zwei automatische Trainingsverfahren vorgeschlagen. Das automatische Training ist wie auch der Auswahlalgorithmus aufgrund der anfallenden Kostenberechnungen sehr rechenzeitintensiv.

### 3.1.3 Das Festival-Sprachsynthesystem

Die *Unit Selection* Komponente des modularen Sprachsynthesystems Festival [BT97b], das an der Universität Edinburg entwickelt wurde, verbindet den Ansatz für kontextuelle Klassifikation (Abschnitt 2.2.2) mit dem Auswahlalgorithmus nach Hunt und Black [HB96]. Die kontextuelle Klassifikation wird als *offline*-Verfahren zur Reduzierung und Strukturierung der Datenbank genutzt, um die Laufzeit für die Einheitenauswahl während der Synthese zu verringern. Die Segmente der Datenbank werden anhand eines objektiven Distanzmaßes in Äquivalenzklassen (*cluster*) partitioniert. Während der Einheitensuche wird diejenige Äquivalenzklasse von Kandidaten für ein bestimmtes Phonem ausgewählt, deren kontextuelle Information mit der des gewünschten Targetsegments übereinstimmt. Die Bestimmung des optimalen Pfades durch das Übergangsnetzwerk von Kandidaten erfolgt anhand des Abstands des Kan-

didaten vom Zentroiden des *Clusters* und den Konkatenierungskosten der Kandidaten. Im Folgenden wird die von Black und Taylor [BT97a] vorgestellte Realisierung des *Cluster*- und des Auswahlalgorithmus erläutert. Die Ergebnisse für das *Pruning* der Datenbank werden anschließend diskutiert.

### Cluster-Algorithmus

Die Klassifikation der Einheiten erfolgt anhand eines akustischen Distanzmaßes, das den Abstand zwischen Einheiten desselben Phonemtyps bestimmt. Black und Taylor [BT97a] verwenden eine gewichtete Mahalanobische<sup>6</sup> Distanzmetrik. Der Abstand zwischen zwei Einheiten wird definiert als die mittlere Distanz der Vektoren über allen Frames der Einheiten sowie eines Teils der jeweils vorangehenden Einheiten, um Information über den segmentellen Kontext zu berücksichtigen. Die Vektoren enthalten folgende Parameter: *Mel-Frequency Cepstral Coefficients (MFCC)*,  $f_0$ , Energie, sowie die entsprechenden Delta-Werte. Die akustische Distanz  $D(U, V)$  zwischen zwei Einheiten  $U$  und  $V$  desselben Phonemtyps wird wie folgt berechnet

$$D(U, V) = P \cdot \frac{|U|}{|V|} \sum_{i=1}^{|U|} \sum_{j=1}^n \frac{W_j}{n\sigma_j|U|} \text{abs}(F_{ij}(U) - F_{(i \cdot |V|/|U|)j}(V)) \quad (3.1)$$

wobei  $|V| > |U|$  ist und die kürzere Einheit  $U$  linear interpoliert wird. Der Parameter  $P$  (*duration penalty*) gewichtet die unterschiedliche Länge der Segmente.  $|U|$  ist die Anzahl der Frames des Segments  $U$  und  $n$  die Anzahl der Parameter eines Vektors.  $W_j$  ist das Gewicht für den Parameter  $j$ , und  $\sigma_j$  ist die Standardabweichung des Parameters  $j$ .  $F_{xy}(U)$  bezeichnet den Parameter  $y$  des Frames  $x$  des Segments  $U$ . Entsprechend steht der Ausdruck  $F_{(i \cdot |V|/|U|)j}$  für den Parameter  $F_{xy}$ , wobei  $x$  als  $i \cdot |V|/|U|$  berechnet wird und  $y = j$  ist.

Die akustische Distanz wird als Maß für die “Unreinheit” (*Impurity*) eines *Clusters* verwendet.<sup>7</sup> Durch Partitionierung des *Clusters* wird eine Minimierung der verbleibenden *Impurity* angestrebt. Die *Impurity*  $I$  eines *Clusters*  $C$  wird bestimmt durch

$$I(C) = \frac{1}{|C|^2} \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} D(C_i, C_j) \quad (3.2)$$

Die Unterteilung des *Clusters* erfolgt anhand kontextueller Information. Durch

<sup>6</sup> Mahalanobische Distanz ist ein euklidisches Abstandsmaß, bei dem jedes Element eines Vektors mit Hilfe der Varianz oder Standardabweichung normalisiert ist.

<sup>7</sup> Die *Impurity* entspricht der Entropie bei Verwendung von Wahrscheinlichkeitsverteilungen.

Verwendung der CART-Technik wird ein Entscheidungsbaum aufgebaut, dessen Knoten Fragen enthalten, die die *Impurity* der *Sub-Cluster* minimieren. Der Entscheidungsbaums wird durch Verwendung eines *Greedy*-Algorithmus generiert. Die minimale *Cluster*-Größe wird auf 10 – 20 Einheiten festgelegt.

Die Klassifikation von Segmenten eines Typs (Phonems) erfolgt aufgrund des phonetischen und prosodischen Kontextes. Für jedes Phonem in der Datenbank wird ein Entscheidungsbaum (CART) aufgebaut. Für die Generierung des Baumes werden folgende Merkmale verwendet: Phonemtyp sowie phonetische Merkmale, prosodische Merkmale wie  $f_0$ , Lautdauer, Wortakzent (*stress*), Position innerhalb der Silbe und Position innerhalb der Phrase.<sup>8</sup> Für die phonetischen und prosodischen Merkmale wird ein Fenster zur Beschreibung des segmentellen Kontextes verwendet. Für die verschiedenen Phonemklassen sind unterschiedliche Merkmale relevant. Der lexikalischer Akzent (*stress*) wird beispielsweise nur für die Phoneme schwa, i, a und n verwendet.

Vorteil des *Cluster*-Algorithmus gegenüber dem von Hunt und Black [HB96] vorgeschlagenen Verfahren ist, dass ein Gewichtetraining für die Merkmale, mit denen die Targetkosten geschätzt werden, nicht notwendig ist. Die Targetkosten müssen nicht während der Synthese berechnet werden, sondern werden bei der Generierung des Entscheidungsbaums als Abstand vom *Cluster*-Zentroiden gemessen. Die eingesparte Rechenzeit kann für den Einsatz besserer Optimierungsalgorithmen und für nachträgliche Signalmodifikationen genutzt werden.

### Auswahlalgorithmus

Während der Synthese wird für jedes Targetsegment ein *Cluster* aus dem entsprechenden Entscheidungsbaum für den jeweiligen Phonemtyp ausgewählt. Die Auswahl des *Clusters* erfolgt anhand der phonetischen und prosodischen Merkmale im Entscheidungsbaum. Die Segmente des ausgewählten *Clusters* sind die Kandidaten, die die Targetspezifikation des Segments am besten approximieren. Die Kandidaten stellen zusammen mit den Targetsegmenten ein Übergangsnetzwerk dar. Ein Viterbi-Algorithmus sucht den optimalen Pfad durch das Kandidatennetzwerk unter Minimierung der Target- und Konkatenierungskosten. Die Targetkosten ergeben sich aus der akustischen Distanz eines Segments zum Zentroiden des *Clusters*, die durch

---

<sup>8</sup> In einer früheren Version wurden zusätzliche Merkmale wie Delta- $f_0$  zwischen einem Segment und dem vorhergehenden Segment verwendet, die jedoch keinen Einfluss auf die Segmentauswahl hatten.

den *Cluster-Algorithmus offline* berechnet wurde. Die Funktion für die Berechnung der Targetkosten ist in Gleichung (3.2) angegeben. Die Konkatenierungskosten werden durch eine *optimal coupling*-Technik berechnet. Beim *optimal coupling* (Abschnitt 2.3) wird durch Vergleich der Spektren zweier aufeinanderfolgender Segmente und Minimierung der akustischen Distanz die optimale Verkettungsstelle bestimmt. Die Suchregion des *optimal couplings* ist innerhalb der Segmentgrenzen beschränkt. Bei einer Übereinstimmung des Phonemtyps des vorangehenden Segments in der Datenbank mit dem ausgewählten Segment wird eine Ausdehnung der Suchregion bis zu 60% in das vorangehende Segment erlaubt. Dabei wird die optimale Verkettungsstelle der beiden Segmente anhand einer Distanzfunktion bestimmt, die die Kosten für die Verknüpfung der Einheiten definiert. Als Kostenfunktion verwenden Black und Taylor [BT97a] eine Frame-basierte gewichtete euklidische Distanz. Die Vektoren enthalten dieselben Parameter wie bei der Berechnung der akustischen Distanz für das *Clustering* der Einheiten: *Mel-Frequency Cepstral Coefficients (MFCC)*,  $f_0$ , Energie und die entsprechenden Delta-Werte. Der Parameter  $f_0$  wird höher gewichtet, um lokale Diskontinuitäten festzustellen, die bei einer Konkatenierung die Signalqualität stark beeinträchtigen können. Die Gesamtkosten ergeben sich dann als Summe der Targetkosten  $Tdist(U_i)$  und der Konkatenierungskosten  $Jdist(U_i, U_{i-1})$  für zwei aufeinanderfolgende Segmente  $U_i$  und  $U_{i-1}$  über allen  $N$  Segmenten der Targetäußerung

$$\sum_{i=1}^N Tdist(U_i) + W * Jdist(U_i, U_{i-1})$$

$W$  ist ein unabhängiger Parameter, der die relative Gewichtung der Target- und Konkatenierungskosten ermöglicht. Da Frame-basierte objektive Distanzmaße Diskontinuitäten an Konkatenierungsstellen nicht ausreichend berücksichtigen, ist ein automatisches Training schwierig und perzeptuelle Experimente unabdingbar.

### Pruning der Datenbank

Ein wichtiger Aspekt der kontextuellen Klassifikation ist die Möglichkeit, die Anzahl der Einheiten der Datenbank durch *Pruning* zu reduzieren. Dies erlaubt einerseits das Entfernen von atypischen Einheiten, die beispielsweise durch verschliffene Artikulation oder falsche Aussprache bei den Sprachaufnahmen und durch Labelfehler bei der Korpusaufbereitung entstehen. Zum anderen können häufige Einheiten, die ausreichend in dem Korpus repräsentiert sind, von der Einheitenauswahl ausgeschlossen werden. Dazu werden die Einheiten eines *Clusters* getilgt, die am weitesten vom

Zentroiden entfernt sind. Bei der Tilgung häufiger Einheiten muss jedoch die Verteilung der Distanzen vom *Cluster*-Zentroiden berücksichtigt werden. Black und Taylor [BT97a] beobachten, dass das *Pruning* der Datenbank um 20% keinen signifikanten Einfluss auf die Sprachqualität hat, während bei 50% eine deutliche Abnahme der Sprachqualität bemerkbar ist. Mit Hilfe von Perzeptionstests ermitteln sie die Anzahl der maximal entfernbaren Einheiten in Abhängigkeit von der *Cluster*-Größe. Für eine *Cluster*-Größe von 10 ergibt sich keine Änderung der Sprachqualität bei einem *Pruning* von bis zu 2 Einheiten, für eine *Cluster*-Größe von 15 bzw. 20 ist ein *Pruning* von 3 bzw. 3 – 4 Einheiten möglich.

Die kontextuelle Klassifikation der Sprachdaten und die Reduktion des Synthesinventars nach Black und Taylor beruhen auf der Verwendung eines akustischen Distanzmaßes. Das Merkmalsset, das die Einheiten eines *Clusters* beschreibt, wird durch die akustische Analyse der Sprachdaten generiert und beinhaltet akustische Merkmale wie Lautdauer. Nach Breen und Jackson [BJ98] ist dieses Verfahren zur Generierung von Merkmalssets zu komplex und rechenzeitaufwendig. Die Verwendung akustischer Merkmale ist fehleranfällig und häufig nicht zur diskriminativen Beschreibung sprachlicher Syntheseeinheiten geeignet. Das Laureate TTS-System von British Telecom [BJ98] verwendet für die *Unit Selection* eine ausschließlich auf phonologischen Kriterien basierende Distanzmetrik.

### 3.1.4 Das Laureate TTS-System

Das Laureate TTS-System von British Telecom (BT) verwendet für die *Unit Selection* ein gemischtes Inventar von  $N$ -Phon Einheiten, die eine Länge von bis zu drei Segmenten (Triphonen) haben können.<sup>9</sup> Die Auswahl der Einheiten erfolgt in zwei Schritten: Die Datenbanksuche identifiziert die besten Einheiten anhand einer Distanzmetrik, die auf phonologischen Kriterien beruht. Der Auswahlalgorithmus findet dynamisch die optimale Kandidatensequenz mit Hilfe einer globalen Kostenfunktion. Für die Kostenberechnung wird die bei der Korpusssuche gewonnene Information über den phonetischen und phonologischen Kontext verwendet sowie Kriterien (*constraints*) auf Signalverarbeitungsebene. Die Kandidatensuche und -auswahl basiert ausschließlich auf phonologischer Information. Damit hängt die *Unit Selecti-*

<sup>9</sup> Die Einheiten können theoretisch beliebige Länge haben; aus Effizienzgründen wird die Einheitenlänge für den auf Dynamischer Programmierung basierenden Auswahlalgorithmus eingeschränkt.

on in sehr hohem Maß von der Qualität der Korpusaufbereitung, d.h. der Korrektheit der Annotationen und dem Informationsgehalt der verwendeten Merkmale ab. Im Folgenden wird die Datenbanksuche und der Auswahlalgorithmus des Systems beschrieben.

### Datenbanksuche

Bei der Korpussuche werden die Einheiten der Targetäußerung mit denen der Datenbank anhand einer *matching function* verglichen. Zwei Einheiten sind identisch, wenn sie dieselben Attribute haben. Als Attribute werden klassische linguistische Merkmale, artikulatorische Merkmale und suprasegmentale Merkmale verwendet. Die phonetischen und phonologischen Kontextmerkmale werden in einer vorverarbeiteten Struktur, dem *phoneme context tree*, gespeichert. Aus Effizienzgründen wird die Suche auf den Baumstrukturen ausgeführt. Während der *context tree search* wird auf der aufbereiteten Datenbank eine exhaustive Suche nach den Einheiten der Targetäußerung durchgeführt. Als Einheit wird eine Sequenz von Segmenten definiert, deren maximale Länge durch ein gegebenes symmetrisches Kontextfenster (*context tree window*) bestimmt ist (z.B. 3 oder 5 Positionen). Bei einer Fensterlänge von 3 werden alle Triphone, Diphone und Phoneme der Targetäußerung betrachtet.<sup>10</sup> Eine Distanzmetrik wird verwendet, um die Ähnlichkeit zwischen den Symbolen eines Kontextfensters und den Symbolen, mit denen die aufbereitete Baumstruktur annotiert ist, festzustellen. Anhand der *matching function* werden die Merkmale, die die segmentelle Identität beschreiben, und die segmentellen und suprasegmentellen Kontextmerkmale verglichen. Die Darstellung der Merkmale in einer Baumstruktur erlaubt, Information über die relative Gewichtung der Merkmale zu geben, was eine euklidische Distanzmetrik nicht leisten kann. Beispielsweise befindet sich die Segmentidentität auf der obersten Ebene im Baum und wird höher gewichtet, um längere Einheiten und damit eine geringere Anzahl an Konkatenationsstellen zu erhalten. Die Menge der Pfade durch den Baum definiert alle möglichen Phoneme und suprasegmentalen Kontexte im Korpus. Während der Suche wird für jeden Kandidaten ein Merkmalsvektor generiert, der den Pfad durch den Baum und die entsprechenden Merkmale eindeutig bestimmt. Die gefundenen Kandidaten für jedes Kontextfenster werden in einem *work space* zwischengespeichert.

---

<sup>10</sup> Aus Symmetriegründen werden fehlende Positionen im *context tree window* durch Stillesegmente aufgefüllt.

## Auswahlalgorithmus

Während der Einheitenauswahl wird ein Targetpfad generiert und jeder Knoten mit dem entsprechenden Kandidatenknoten verglichen. Der Wert der Distanzmetrik wird durch eine *scoring function* berechnet, die die identischen Pfadentscheidungen angibt. Durch die Verwendung des *context tree windows* können die Einheiten überlappen. Die Kosten werden für jede Verknüpfung berechnet und der optimale Pfad durch Dynamische Programmierung bestimmt. Der Algorithmus favorisiert in der Regel überlappende Sequenzen von längeren Einheiten. Die durchschnittliche Einheitenlänge hängt jedoch von der Anzahl der Kandidaten ab, die während der Korpusuche gefunden werden. Abbildung 3.2 zeigt die Suche des optimalen Pfades durch die Einheiten im *work space*.

Mittleres Phonem	#	t	Q	m	@	s	#
Linkes Diphon		#t_	tQ_		m@_	@s_	s#_
Rechtes Diphon	_#t	_tQ	_Qm		_@s	_s#	
Phon	_t_			_m_	_@_	_s_	_#_
Triphon		#tQ	tQm				

Abbildung 3.2: Pfadauswahl beim Laureate TTS-System [BJ98]

Die Ausgabe des *Unit Selection* Prozesses ist eine Folge von Referenzen zu Einheiten im Sprachkorpus, aus denen das Sprachsignal zusammengesetzt wird.

Der Vorteil des Ansatzes von Breen und Jackson [BJ98] ist, dass keine akustischen Merkmale generiert und zur Einheitenauswahl herangezogen werden müssen. Die Korpusuche und Einheitenauswahl erfolgt durch eine auf phonetischen und prosodischen Kriterien basierende Distanzmetrik. Eine Verbesserung des Ansatzes versprechen sich die Autoren durch ein präziseres phonologisches Merkmalsmodell. Ein weiterer Ansatz, der auf phonologischen Bäumen basiert, wird im nächsten Abschnitt 3.2.2 vorgestellt.

## 3.2 Eingeschränkte Anwendungsdomänen

Für viele Einsatzgebiete von Sprachsynthesystemen ist eine eingeschränkte Anwendungsdomäne charakteristisch. Diese bestimmt auch den Schwierigkeitsgrad der Anwendung. Einige Domänen lassen sich durch ein festes Inventar abdecken, das aus wenigen Äußerungen und einem festgelegten Vokabular besteht. Es handelt sich dabei um abgeschlossene Domänen. Eine auf dem *Slot-Filler*-Prinzip basierende Synthese ist für solche Anwendungsgebiete meist ausreichend und liefert eine sehr hohe Sprachqualität. Dialogsysteme verfügen häufig über eine eingeschränkte Domäne mit offenem Vokabular. Neue Wörter können beispielsweise als Eigennamen auftreten. Solche Domänen stellen größere Anforderungen an die Sprachsynthese. Die Frequenz von *out-of-vocabulary* Wörtern beeinflusst den Schwierigkeitsgrad der Domäne. Bei geringem Vorkommen von neuen Wörtern kann eine einfache *back-off* Strategie wie das Umschalten auf eine Diphonsynthesestimme ausreichend sein. Jedoch resultiert eine solche Vorgehensweise in einer stark variierenden Synthesequalität. Häufiges Vorkommen unbekannter Wörter erfordert dagegen einen grundlegenden Ansatz, beispielsweise durch die Kombination verschiedener *Unit Selection* Verfahren.

### 3.2.1 Der CMU DARPA Communicator

Der an der Carnegie Mellon University (CMU) entworfene *Communicator* ist ein telefonbasiertes Dialogsystem für Reiseplanung, Flug-, Auto- und Hotelreservierungen. Die Domäne ist nicht abgeschlossen, da registrierte Benutzer mit ihrem Namen begrüßt werden. Weitere Eigennamen für Flughäfen, Fluglinien etc. können auftreten.

Für die Anwendung wurde ein domänenspezifisches Sprachkorpus mit Hilfe der Festvox-Werkzeuge [BL00a] erstellt. Obwohl es sich um eine eingeschränkte Anwendungsdomäne handelt, ist ein *Slot-Filler*-Ansatz aufgrund der Variabilität der Äußerungsstruktur nicht geeignet. Daher wurde auf eine allgemeinere *Unit Selection* Technik zurückgegriffen. Der Ansatz des Festival Sprachsynthesystems [HB96, BT97a] (Abschnitt 3.1.3) wurde modifiziert, um für eine anwendungsspezifische Domäne die Effizienz der Einheitenwahl zu steigern und die Synthesequalität zu verbessern.

## Korpusdesign

Für das Projekt wurde eine Datenbank mit den häufigsten Äußerungen und Phrasen, die die Anwendungsdomäne beinhaltet, erstellt. Die Datenbank beinhaltet ca. 100 feste Sätze und Phrasen ohne variable Teile, z.B. Begrüßungsformeln wie “*Welcome to the CMU Communicator*” oder häufige Ausgaben des Systems wie “*I’m sorry, I don’t understand that*”. Die festen Äußerungen wurden vollständig in das Sprachkorpus übernommen und für die Sprachaufnahmen verwendet. In einem weiteren Schritt wurde aus der Datenbank eine Menge von Templates und möglichen *Slot-Fillern* wie Städtenamen, Bezeichnungen für Flughäfen und -linien sowie Zahlen, Preis-, Datums- und Zeitangaben extrahiert. Letztere stellen eine abgeschlossene Klasse dar, aus denen eine verkleinerte Liste unter Berücksichtigung der Wortabdeckung konstruiert wurde. Für die offene Klasse der Städtenamen und Bezeichnungen für Flughäfen und -linien wurden die Häufigkeitsverteilungen festgestellt. Häufige Namen und Bezeichnungen wurden in verschiedene prosodische Kontexte eingebettet, während seltenere Namen und Bezeichner nur einmal in prosodisch neutralen Kontexten in das Sprachkorpus aufgenommen wurden. Es wurden insgesamt 300 Namen für Städte und Flughäfen extrahiert. Aus dem Templates und möglichen *Slot-Fillern* wurden ca. 500 Äußerungen konstruiert und aufgenommen. Die Sprachaufnahmen wurden automatisch segmentiert und gelabelt und teilweise manuell korrigiert.

## Einheitenauswahl

Im Gegensatz zu Sagisakas Vorschlag [Sag88] kann der in Abschnitt 3.1.3 beschriebene Ansatz des Festival Sprachsynthesystems [BT97a] als *uniform Unit Selection* bezeichnet werden. Black und Lenzo [BL00b] verfolgen durch Modifikation des Algorithmus in [BT97a] die Strategie einer *non-uniform Unit Selection* für beschränkte Anwendungsdomänen. Bei der Einheitenauswahl werden zusätzlich zu den Einheiten eines ausgewählten *Clusters* alle Segmente berücksichtigt, die in dem Sprachkorpus auf ein bereits ausgewähltes Segment folgen und denselben Phonemtyp haben. Dadurch können längere Einheiten ausgewählt werden, und die Einheiten verfügen über unterschiedliche Länge. Bei der Auswahl der Einheiten wird statt dem Merkmal Phonemtyp ein Konstrukt Phonemtyp + Wort verwendet. Dadurch werden Einheiten aus der Datenbank ausgewählt, die Teil eines Wortes sind, das mit dem zu synthetisierenden Wort übereinstimmt. Obwohl das zu synthetisierende Wort als Auswahlkriterium dient, kann der Ansatz jedoch nicht mit einem Ansatz zur Wortkonkatenation verglichen werden. Es werden häufig Phoneme von verschiedenen In-

stanzen eines Wortes aus dem Korpus ausgewählt, und der Konkatenierungspunkt wird dynamisch durch das *optimal coupling* bestimmt [BL00b]. Durch die Verwendung des Wortes als Auswahlkriterium stehen weniger Kandidaten für die *Unit Selection* zur Verfügung. Dies führt zu einer enormen Verbesserung der Laufzeit des *Unit Selection* Algorithmus. Problematisch ist die Rigidität des Kriteriums für die Synthese von Sätzen, die nicht der Anwendungsdomäne angehören. Bei der Synthese von Wörtern, die nicht im Korpus vorkommen (*out-of-vocabulary*), beobachten Black und Lenzo eine sehr starke Abnahme der Sprachqualität. Der Ansatz liefert keine skalierbare Sprachqualität für nicht-domänenspezifisches Textmaterial. Als *back-off* Strategie wird die Diphonsynthese verwendet. Da die perzeptuelle Qualität der Sprachausgabe zwischen der Diphonsynthese und der Synthese mit *Unit Selection* sehr unterschiedlich ist, kann das System die Synthesestimme nicht innerhalb einer Äußerung umschalten. Wenn eine Äußerung ein Wort enthält, das nicht vom Korpus abgedeckt ist, wird daher die ganze Äußerung mit der Diphonstimme erzeugt. Eine anwendungsorientierte Evaluation ergab, dass von 18.276 synthetisierten Phrasen 459 (2.5%) Wörter enthielten, die nicht vom Korpus abgedeckt sind.

Black und Lenzo [BL00b] konnten durch das sorgfältige Korpusdesign und durch die vorgenommenen Modifikationen des Syntheseverfahrens für anwendungsspezifische Domänen eine Verbesserung der Synthesequalität erreichen. Für Sätze, deren Wörter im Korpus vorkommen, ist die Qualität der Sprachausgabe sehr hoch und kaum von natürlicher Sprache unterscheidbar. Für eine weniger eingeschränkte Sprachsynthese mit einem größerem Vokabular, variierender Äußerungsstruktur und Prosodie ist der modifizierte Syntheseansatz aufgrund des erforderlichen Abdeckungsgrades des Sprachkorpus nicht ausreichend.

### 3.2.2 Das MIT Jupiter System und der ILEX Museumsführer

Das *Phonological Structure Matching* (PSM) Verfahren von Taylor und Black [TB97, Tay00] wurde ursprünglich für unbeschränkte TTS-Systeme entwickelt. Da der Ansatz sehr großen Nutzen von domänenspezifischem Textmaterial macht, ist er jedoch eher für eingeschränkte Anwendungsgebiete geeignet [SBK<sup>+</sup>03]. Er wurde in zwei Anwendungen eingesetzt, die über eine eingeschränkte Domäne verfügen: das MIT Jupiter Wettervorhersagesystem und der ILEX Museumsführer. Die Domänen beinhalten häufig vorkommende feste Phrasen wie *“In Bosten today”* und *“partly cloudy*

skies". Ein *Slot-Filler*-Ansatz ist jedoch insbesondere für die Domäne des ILEX Museumsführers nicht ausreichend, da neue Wörter und Phrasen auftreten können. Der PSM-Algorithmus verbindet die nötige Flexibilität eines *Unit Selection* Verfahrens für unbeschränkte Domänen mit der Sprachqualität eines *Slot-Filler*-basierten Re-Syntheseverfahrens.

Wie das Laureate TTS-System [BJ98] (Abschnitt 3.1.4) verwendet der PSM-Algorithmus phonologische Bäume. Bei der Synthese wird die Targetäußerung als phonologische Baumstruktur spezifiziert. Das Korpus ist eine *offline* vorverarbeitete Struktur aus phonologischen Bäumen, auf der die Einheitensuche zur Laufzeit ausgeführt wird. Der PSM-Algorithmus sucht in dem Korpus die längsten Einheiten, die Teilen des Targetbaumes entsprechen. Die Suche erfolgt dabei ausschließlich anhand phonologischer und struktureller Information, die durch die Spezifikation der Targetäußerung und des Korpus als phonologische Bäume gegeben ist. Dadurch können Probleme, die bei fehlerhafter Repräsentation der Ziel- oder Kandidatenspezifikation durch prosodische Module entstehen, vermieden werden. Der Suchraum wird erheblich reduziert, was eine Effizienzsteigerung des Auswahlalgorithmus bewirkt.

### Phonologische Bäume

Jeder Knoten im Baum ist mit einem Merkmalsvektor annotiert, der kategorielle linguistische und phonologische Merkmale beinhaltet. Merkmalsvektoren auf unterschiedlichen linguistischen Ebenen im Baum können verschiedene Attribut-Werte-Paare als Elemente haben. Knoten auf Wortebene enthalten *Part-of-speech* und syntaktische Information. Intonation wird auf Silbenebene durch das Merkmal *akzentuiert* beschrieben. Auf Phonemebene werden Artikulationsort und -art, Stimmhaftigkeit etc. angegeben. Der Aufbau der phonologischen Bäume als binäre metrische Baumstruktur liefert weitere Informationen: Die Gewichtung der Tochterknoten eines Knotens wird durch die Relation *strong* - *weak* angegeben.

Abbildung 3.3 stellt einen Ausschnitt einer phonologischen Baumstruktur für eine Äußerung dar.

Die Syntheseeinheiten des Korpus sind Knoten innerhalb der phonologischen Bäume. In einer Äußerung kann eine Einheit eine Phrase (z.B. *around nineteen twenty*), ein Wort (z.B. *nineteen*), eine Silbe (z.B. *nine*), ein Teil einer Silbe wie der *Onset* (z.B. *tw*) oder ein einzelnes Phonem sein. Der PSM-Algorithmus sucht in der Datenbank die Einheiten, die den Knoten im Targetbaum entsprechen und weist sie diesen als Kandidaten zu. Aus den Kandidatenlisten wird die optimale

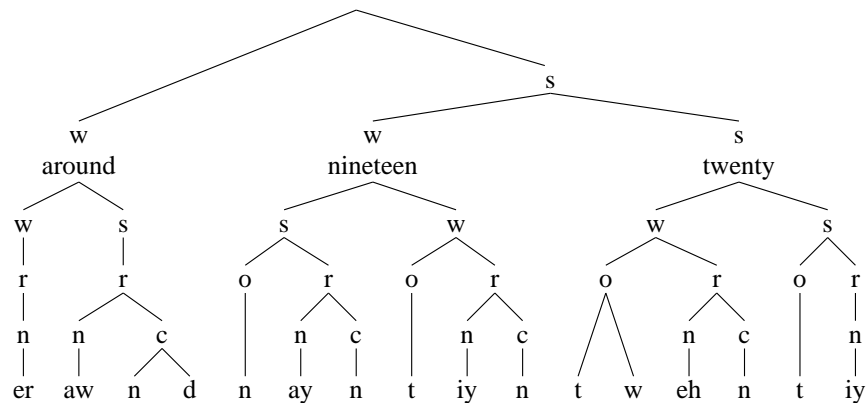


Abbildung 3.3: Ausschnitt eines phonologischen Baums [TB97, Tay00]

Einheitensequenz ausgewählt und die Einheiten konkateniert. Im Folgenden werden die drei Schritte, in die das *Unit Selection* Verfahren nach Taylor und Black unterteilt ist, beschrieben: die Kandidatensuche, die Kandidatenauswahl und die Konkatenation der Einheiten mit (optionaler) Signalmodifikation.

### PSM-Suche

Anhand einer *matching function* werden die Einheiten als Kandidaten für die Einheitenauswahl durch Vergleich der Knoten des Targetbaums mit denen der phonologischen Bäume in der Datenbank gesucht. Zwei Einheiten stimmen überein, wenn die Teilbäume unterhalb der entsprechenden Knoten übereinstimmen und die Blätter mit denselben Phonemen gelabelt sind. Der Algorithmus sucht *top-down* ausgehend vom Wurzelknoten des Targetbaums nach den passenden Einheiten im Korpus. Wird eine geeignete Einheit als Kandidat gefunden, wird sie dem entsprechenden Knoten im Targetbaum zugewiesen. Die Suche bricht auf dieser Ebene ab und wird für den nächsten Targetknoten fortgeführt. Wenn keine passende Einheit im Korpus gefunden wird, wird die Suche für alle Tochterknoten des aktuellen Knotens fortgesetzt. Im ungünstigsten Fall muss der Algorithmus die Suche auf Segmentebene durchführen. Dieser Fall tritt ein, wenn keine größeren Konstituenten des Targetbaums im Korpus gefunden werden können. Der Algorithmus terminiert, wenn alle Knoten des Targetbaums abgearbeitet wurden oder auf einer höheren Ebene bereits Kandidaten gefunden wurden. Im Folgenden wird der PSM-Algorithmus aus [TB97] angegeben (Algorithmus 3.1):

Die Kandidaten sind *non-uniform*, da sie eine unterschiedliche Anzahl von Seg-

---

**Benötigt:** Target  
finde passenden Kandidaten in der Datenbank  
**if** |Kandidaten| > 0 **then**  
    weise Kandidaten Target zu  
**else**  
    **for all** Tochterknoten von Target **do**  
        PSM(Tochterknoten)  
    **end for**  
**end if**

---

Algorithmus 3.1: PSM-Algorithmus

menten haben können. Im Gegensatz zu dem Ansatz in [BJ98] können die Einheiten unterschiedlichen linguistischen Kategorien angehören. Es kann sich um Segmente, Silben, Wörter oder Phrasen handeln. Abbildung 3.4 zeigt einen Targetbaum, dem Kandidaten auf verschiedenen Ebenen zugewiesen sind. Die Kandidaten eines Knotens sind in einer Liste enthalten. Jedes Element der Kandidatenliste referiert auf eine Sprachsequenz in der Datenbank, die durch einen Knoten eines phonologischen Baums repräsentiert wird. Die Kandidaten einer Liste stimmen untereinander und mit den entsprechenden Knoten des Targetbaums überein.

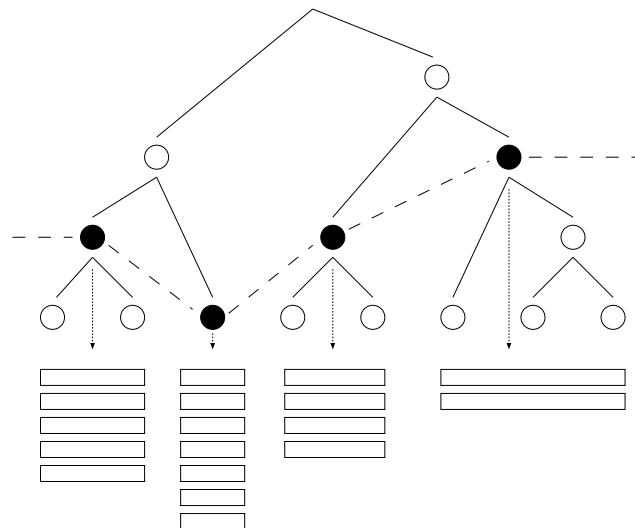


Abbildung 3.4: Binärer Targetbaum mit Kandidaten [Tay00]

Ein Vorteil des PSM-Algorithmus ist, dass die Kandidaten nicht überlappen. Dies

kann bei der *non-uniform* Synthese ein Problem darstellen, da die Anzahl der möglichen Konkatenationsstellen und damit die Anzahl der Berechnungen während der Einheitenauswahl sehr viel größer ist.

### Auswahlalgorithmus

Der Auswahlalgorithmus sucht die optimale Sequenz von Einheiten aus den Kandidatenlisten anhand einer *scoring function* aus. Bei der Kandidatensuche wird nur die Struktur der Teilbäume unterhalb des jeweiligen Knotens und die phonetischen Label an den Blättern des Teilbaums berücksichtigt. Die Gewichtung der Tochterknoten (*strong - weak*) spielt bei der Suche keine Rolle. Phonologische Merkmale wie Wortakzent und Intonation, mit denen die Knoten annotiert sind, werden ebenfalls nicht verwendet. Bei der Kandidatenauswahl hingegen wird die Übereinstimmung der Kandidaten mit dem entsprechenden Targetknoten anhand phonologischer und kontextueller Merkmale wie *phrase initial* und *phrase final* festgestellt. Die *scoring function* berechnet die Targetkosten für jeden Kandidaten. Der Wert 0 entspricht einer Übereinstimmung aller Elemente der phonologischen Merkmalsvektoren. Die Konkatenierungskosten werden anhand phonologischer Kontextmerkmale und akustischer Information berechnet [Tay00]. Als akustisches Distanzmaß wurde eine Frame-basierte Mahalanobische Distanz (Gleichung (3.1)) zwischen den Parametern der akustischen Merkmalsvektoren gewählt. Folgende Parameter werden verwendet: *Mel-Frequency Cepstral Coefficients (MFCC)*,  $f_0$  und Energie. Die Auswahl der optimalen Kandidatensequenz erfolgt durch einen Viterbi-Dekoder. Die Kandidatenlisten stellen einen gerichteten Graph dar, dessen Knoten die Kandidaten sind. Die Kandidaten eines Targetknotens sind mit allen Kandidaten des nachfolgenden Targetknotens verbunden. Startknoten ist das erste Element der Kandidatenliste des ersten Targetknotens. Der Viterbi-Algorithmus berechnet die partiellen Pfadkosten, die sich aus den Target- und Konkatenierungskosten ergeben. Aufgrund der linearen Netzwerktopologie wird für jeden Kandidaten nur der Pfad mit den niedrigsten Kosten zum Zeitpunkt  $t$  gespeichert. Die Kandidaten, die den global kostengünstigsten Pfad durch das Netzwerk darstellen, werden vom Algorithmus ausgewählt.

### Synthese

Die Sprachsignale der ausgewählten Kandidaten werden aus dem Sprachkorpus extrahiert und miteinander konkateniert. Der Ansatz in [TB97, Tay00] verwendet Signalverarbeitungstechniken zur Modifikation der prosodischen Eigenschaften der

Einheiten wie Intonation und Lautdauer, wenn diese zu stark von der Targetspezifikation abweichen. Dazu wird eine *back-off* Strategie eingesetzt:

Die Targetspezifikation enthält neben phonologischen Merkmalen auch akustische Werte wie  $f_0$  und Lautdauer, die von den Modulen für die Prosodiegenerierung vorhergesagt werden. Bei der Berechnung der Targetkosten anhand der *scoring function* wird ein normalisierender Faktor  $\alpha$  bestimmt, der ein Maß für den Abstand zwischen der phonologischen Spezifikation des Targets und des jeweiligen Kandidaten darstellt. Der Faktor dient dazu, eine optimale Balance zwischen Target und Syntheseinheit zu finden. Die resultierende Lautdauer und  $f_0$ -Kurve ergeben sich als gewichtete Linearkombination zwischen den akustischen Parametern des Targets und der ausgewählten Syntheseinheit. Die Adaption der Lautdauer eines Segments wird wie folgt berechnet:

$$d_{final} = \alpha d_{target} + (1 - \alpha) d_{source}$$

Die Parameter für die Intonation können analog bestimmt werden. Mit Hilfe von Signalverarbeitungstechniken werden die Sprachsignale entsprechend der berechneten akustischen Parameter modifiziert und konkateniert.

Abbildung 3.5 zeigt den Systemaufbau des PSM-Verfahrens mit Signalmodifikation und Synthese.

Das PSM-Verfahren wählt die Einheiten für die Synthese anhand einer phonologischen Repräsentation aus, deren Merkmalsbeschreibungen auf der linguistischen Ebene angesiedelt sind. Die phonetische Repräsentation wird ausschließlich für nachfolgende Signalbearbeitung verwendet. Ein Vorteil von phonologischen Bäumen ist, dass die Anzahl der Parameter für den Auswahlalgorithmus reduziert ist. Die niedrige Anzahl an phonologischen Merkmalen vereinfacht das automatische bzw. manuelle Gewichtetraining. In [TB97, Tay00] werden die Gewichte für die Merkmale Wortakzent (*stress*) und Intonation sowie für einige kontextuelle Merkmale von Hand gesetzt und durch Perzeptionstests überprüft.

Der PSM-Algorithmus liefert durch die *top-down* Suche auf linguistischen Ebenen längere Syntheseinheiten, wodurch die Anzahl der Konkatenierungsstellen sowie das notwendige Maß an Signalverarbeitung reduziert werden. Die geringere Anzahl an Signalstörungen führt zu einer besseren Signal- und Sprachqualität. Durch längere Einheiten können koartikulatorische Effekte implizit modelliert werden. Bei der Modellierung von Rhythmus, Lautdauer, phrasenfinaler Längung und Intonation liefert das PSM-Verfahren sehr gute Ergebnisse [Tay00]. Die Sprachqualität ist häufig sehr

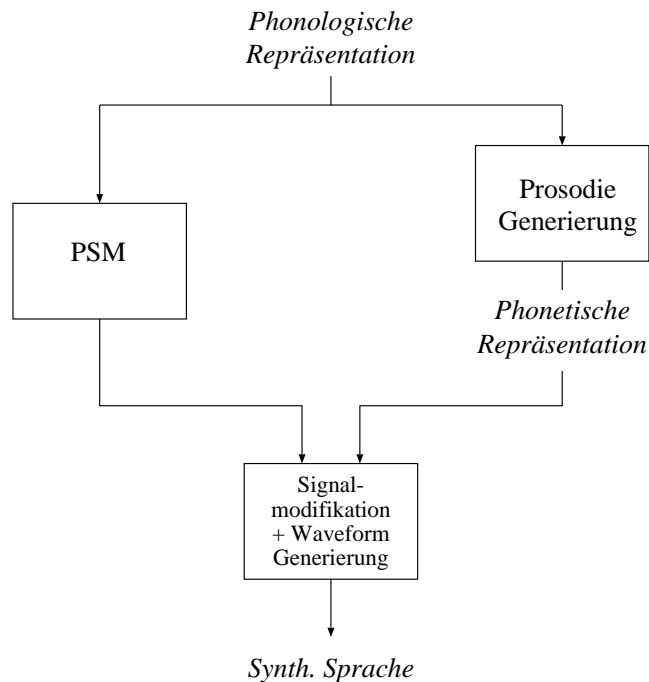


Abbildung 3.5: PSM-Systemaufbau mit *back-off* Strategie

gut und kaum von natürlicher Sprache unterscheidbar. Durch längere Synhteseeinheiten ist die Anzahl der Kandidaten geringer. Die Komplexität des Auswahlverfahrens wird reduziert und die Effizienz bezüglich der Rechenzeit gesteigert. Wenn der PSM-Algorithmus keine linguistischen Einheiten oberhalb der Segmentebene findet, entspricht die Komplexität des Auswahlverfahrens der des Ansatzes in [HB96]. Dies ist bei schlechter Anpassung der Targetdomäne an die Anwendungsdomäne der Fall, wenn viele Phrasen und Wörter vorkommen, die nicht im Sprachkorpus abgedeckt sind. Für *out-of-vocabulary* Wörter und Phrasen ist die Sprachqualität sehr viel geringer, da die Vorteile des PSM-Algorithmus sich vor allem bei anwendungsspezifischem Textmaterial zeigen.

## 4 Hybride Unit Selection

Für das SmartKom-Dialogsystem wurde eine hybride *Unit Selection* Strategie entwickelt, die an die spezifische Anwendungsdomäne des Projektes angepasst ist. Die Domäne ist auf verschiedene Anwendungsbereiche eingeschränkt und umfasst Bereiche wie Kino- und Fernsehprogramminformation, Touristeninformation, Routenplanung, Telefon- und Adressbuchverwaltung. Da in den Dialogen unbekannte Eigennamen, Filmtitel und insbesondere fremdsprachliche Ausdrücke auftreten können, ist die Vokabulargröße praktisch unbeschränkt. Die Synthesekomponente des Dialogsystems muss also domänenspezifisches und unbeschränktes Textmaterial verarbeiten. Aufgrund dieser Dichotomie der Anwendungsdomäne werden zwei unterschiedliche *Unit Selection* Ansätze miteinander kombiniert. Das *Phonological Structure Matching* Verfahren bietet eine hohe Sprachqualität für domänenspezifische Äußerungen und Phrasen und reduziert den Aufwand der Einheitenauswahl. Der *Cluster*-Algorithmus zeichnet sich durch Robustheit gegenüber unbeschränktem Textmaterial aus. Für das Korpusdesign wurde ebenfalls eine zweigeteilte Strategie verfolgt. Sowohl domänenspezifisches als auch unbeschränktes Material wurden für die Aufnahmen verwendet. Häufig vorkommende domänenspezifische Phrasen und Wörter wurden mehrfach in verschiedenen prosodischen Kontexten aufgenommen. Für die Auswahl von unbeschränktem Text wurde die Phonem- und Diphon-Abdeckung in verschiedenen phonetischen und prosodischen Kontexten kontrolliert.

### 4.1 Korpusdesign

Für den Entwurf eines *Unit Selection* Verfahrens ist es essentiell, dass Korpus und Auswahlalgorithmus aufeinander und auf die Anwendungsdomäne abgestimmt sind, damit der Algorithmus größtmöglichen Nutzen von den Daten machen kann. Die Verwendung zweier unterschiedlicher *Unit Selection* Algorithmen macht daher auch ein zweigeteiltes Korpusdesign nötig. Während der PSM-Algorithmus bevorzugt längere domänenspezifische Einheiten auswählt, sucht der *Cluster*-Algorithmus Ein-

heiten auf Segmentebene aus. Für den *Cluster*-Algorithmus muss daher die Phonem- und Diphon-Abdeckung der Datenbank kontrolliert werden, um eine optimale Abdeckung zu erreichen. Für den PSM-Algorithmus wurde domänenspezifisches Material, wie häufige Phrasen und Wörter, in verschiedenen prosodischen Kontexten eingebettet.

Um die Einheiten-Abdeckung des Korpus zu optimieren, wurde ein *Greedy*-Algorithmus verwendet wie in [vSB97] vorgeschlagen. Dazu wurde ein deutsches Zeitungskorpus von etwa 170.000 Sätzen automatisch mit phonetischer Transkription und prosodischen Merkmalen gelabelt. Jedes Segment wurde dabei mit einem Merkmalsvektor annotiert, der folgende Merkmale beinhaltet (Tabelle 4.1):

<b>Ebene</b>	<b>Merkmal</b>
Segmentebene	Segmentidentität
	Position innerhalb der Silbe (onset, coda)
Silbenebene	Wortakzent ( <i>stress</i> )
	Akzentuierung
	Phrasengrenze
	Position innerhalb des Wortes ( <i>initial, medial, final</i> )
Wortebene	Wortklasse (Funktionswort, Inhaltswort)

Tabelle 4.1: Übersicht über die Merkmale für das Korpusdesign

Jedem Satz wurde eine mit den in Tabelle 4.1 aufgelisteten Merkmalen annotierte Phonsequenz und eine entsprechende Diphonsequenz zugeordnet. Mit Hilfe des *Greedy*-Algorithmus wurden iterativ diejenigen Sätze ausgewählt, die die Anzahl der neuen Merkmalsvektoren und Diphonkombinationen maximieren. Diphone, die nicht im Korpus auftraten, wurden in Trägersätze eingebettet und dem Korpus hinzugefügt. Problematisch für die Methode ist das häufige Vorkommen von deutschen Komposita, Abkürzungen und fremdsprachlichen (v.a. englischen) Wörtern und Phrasen, die fehlerhaft annotiert sind. Diese mussten teils automatisch, teils manuell korrigiert werden. Fremdsprachliche Diphone wurden auf deutsche Diphone und Diphonkombinationen abgebildet. Das Einheiteninventar wurde partiell durch englische und französische Inventare ergänzt, für diejenigen Diphone, die nicht durch einen deutschen Laut ersetzbar sind (z.B. das englische Phonem *T* wie in “through”). Durch das eingesetzte Verfahren konnten 90% der Diphonkombinationen abgedeckt werden. Von 4.600 relevanten Merkmalsvektoren wurden etwa 60% abgedeckt. Der

geringe Abdeckungsgrad für Merkmalsvektoren ist u.a. auf die LNRE-Charakteristik der Merkmale zurückzuführen.

Für den domänenspezifischen Teil des Korpus wurden 2.643 SmartKom-spezifische Wörter und Phrasen aufgenommen, darunter Auszüge von Demodialogen und hochfrequente Wörter und Phrasen, wie sie für *Slot-Filler*-Anwendungen üblich sind (z.B. Zahlen, Wochentage, Zeitangaben, häufige Eigennamen). Englische und deutsche Filmtitel stellen den größten Anteil des Textmaterials dar.

Die Aufnahmen wurden mit einem professionellen männlichen Sprecher gemacht. Das Korpus umfasst 160 Minuten Sprachaufnahmen, die automatisch segmentiert, gelabelt und von Hand korrigiert sind. Die Daten sind automatisch mit intonatorischen Labels wie Akzent und Grenzton annotiert und teilweise manuell korrigiert.

## 4.2 Unit Selection Verfahren

Im Folgenden werden die einzelnen Komponenten des hybriden Synthesystems erläutert und ein Überblick über die Systemarchitektur gegeben.

### 4.2.1 Phonologische Baumsuche

Während der Synthese wird von den linguistischen Analysekomponenten und der Prosodiekomponente des Systems eine Targetspezifikation in Form eines phonologischen Baumes erzeugt. Im Gegensatz zu dem Ansatz von Taylor und Black [TB97, Tay00] werden keine binären metrischen Bäume verwendet. Die phonologischen Bäume haben die Struktur von Festival-Bäumen (*utterances*) [BT97b], die mit zusätzlichen Merkmalen für die Suche und die Einheitenwahl annotiert sind. Die Ebenen des Baumes entsprechen den linguistischen Ebenen von Phrasen, Wörtern, Silben und Segmenten. Die Knoten auf den unterschiedlichen Ebenen können verschiedene Merkmale beinhalten. Wie bei dem Ansatz in [TB97, Tay00] werden zwei Mengen von Merkmalen definiert: *primäre* Merkmale, anhand denen der PSM-Algorithmus die Kandidaten aussucht, und *sekundäre* Merkmale, die für die Einheitenwahl relevant sind. Die Unterscheidung ist dadurch motiviert, dass die Synthese von prosodisch suboptimalen, längeren Einheitensequenzen eine perzeptuell höhere Signalqualität liefert als die Konkatenierung von mehreren kürzeren Einheiten, die den prosodischen Kontext gut approximieren. Bei der Konkatenation von längeren Einheitensequenzen treten weniger spektrale Diskontinuitäten auf, die von Hörern als

störender bewertet werden als prosodische “Mismatches” [BC95]. *Primäre* Merkmale sind wesentliche Auswahlkriterien für die Kandidatensuche, während die *sekundären* Merkmale die Einheiten der Targetäußerung hinsichtlich phonetischer und prosodischer Kontexteigenschaften spezifizieren. In [TB97, Tay00] erfolgt die Einheitensuche durch den PSM-Algorithmus anhand der Struktur des zu einem Knoten gehörenden Teilbaumes und der phonetischen Transkription. Für die Einheitenauswahl werden weitere Merkmale wie Wortakzent und Positionsmerkmale verwendet. Die Merkmalsunterscheidung des hier vorgestellten Ansatzes ist flexibler, da sie die Angabe weiterer Merkmale als Suchkriterien für den PSM-Algorithmus erlaubt. Tabelle 4.2 gibt eine Übersicht über die verwendeten *primären* und *sekundären* Merkmale:

	<b>Ebene</b>	<b>Merkmal</b>
<i>Primäre</i> Merkmale	alle	phon. Transkription (Segmentidentität)
		Silben-, Wort- und Phrasengrenzen
		Wortakzent ( <i>stress</i> )
<i>Sekundäre</i> Merkmale	Phrasenebene	keine
	Wortebene	Akzent ( <i>ToBI light</i> )
		Grenzton ( <i>ToBI light</i> )
		Position innerhalb der Phrase
		Position innerhalb des Satzes
		POS <sup>1</sup>
	Silbenebene	Akzent ( <i>ToBI light</i> )
		Grenzton ( <i>ToBI light</i> )
		Position innerhalb des Wortes
		Position bezüglich Phrase und Satz

Tabelle 4.2: Übersicht der *primären* und *sekundären* Merkmale

Abbildung 4.1 gibt die Spezifikation der Äußerung “*Ich habe diese Informationen zu 'Die Innere Sicherheit'*” als Targetbaum an. Die *primären* Merkmale sind in Fettschrift wiedergegeben; die *sekundären* Merkmale sind darunter angegeben. Zu den *primären* Merkmalen gehört auch die Baumstruktur, die Information über Phrasen-, Wort- und Silbengrenzen enthält.

<sup>1</sup> Part-of-speech Information wird in der derzeitigen Implementierung nicht mehr verwendet.

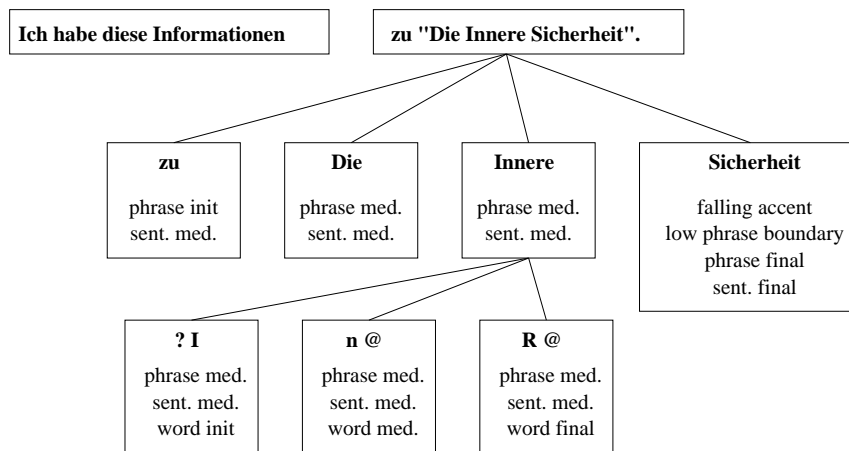


Abbildung 4.1: Targetspezifikation für eine Beispieläußerung

Die Datenbank ist eine *offline* vorverarbeitete Struktur, die mit denselben Merkmalen annotiert ist wie die Targetäußerung.<sup>2</sup> Für die Korpusuche wird eine *matching function* definiert, die die Struktur der Teilbäume zweier Knoten und die *primären* Merkmale vergleicht. Der PSM-Algorithmus arbeitet wie in Abschnitt 3.2.2 dargestellt. Während der Suche nach passenden Einheiten wird der Targetbaum *top-down* durchlaufen. Abweichend von [TB97, Tay00] erfolgt die PSM-Suche nur für Einheiten im Targetbaum oberhalb der Segmentebene. Die Kandidaten sind also Phrasen, Wörter oder Silben. Wenn kein passender Kandidat gefunden wird, bricht der Algorithmus auf Segmentebene ab. Die fehlenden Segmente werden durch den *Cluster*-Algorithmus (Abschnitt 4.2.2) ergänzt. Wird ein Kandidat gefunden, so wird er an den entsprechenden Knoten der Targetstruktur angehängt. Abbildung 4.2 zeigt einen Targetbaum, der nach Abschluss der Suche Kandidatenlisten für Einheiten auf verschiedenen Ebenen enthält.<sup>3</sup>

Da der PSM-Algorithmus bevorzugt größere linguistische Einheiten im Korpus findet, wird der Suchraum erheblich reduziert und die Effizienz des nachfolgenden Auswahlalgorithmus gesteigert. Aufgrund der unterschiedlichen Häufigkeitsverteilungen von Phrasen, Wörtern und Silben variiert die Anzahl der gefundenen Kandidaten für die Targeteinheiten stark. Dies ist u.a. auf die LNRE-Charakteristik der Sprache zurückzuführen, die sich auf verschiedenen linguistischen Ebenen nieder-

<sup>2</sup> Aus Effizienzgründen werden die Festival-Baumstrukturen der Datenbank in Listen transformiert, auf denen die Suche erfolgt.

<sup>3</sup> Im Unterschied zu [TB97, Tay00] ist der Targetbaum nicht binär (vgl. Abbildung 3.4).

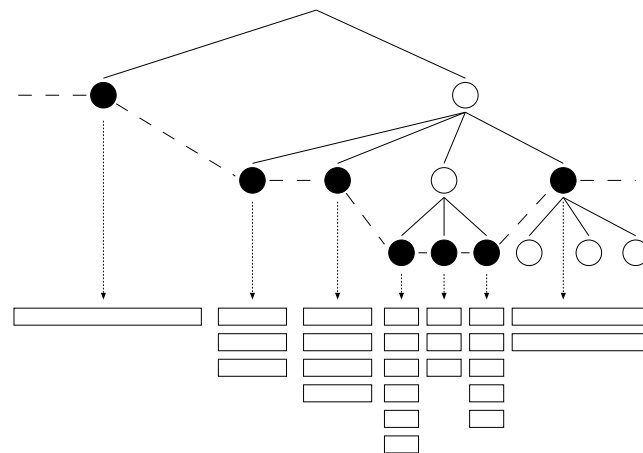


Abbildung 4.2: Targetbaum mit Kandidaten

schlägt [Möb03]. Insbesondere für Funktionswörter wie Artikel und für häufige Silben wie das Verbpräfix “ge” können sehr viele Kandidaten auftreten, wodurch die Effizienz der Einheitenauswahl beeinträchtigt wird. Um den Auswahlalgorithmus und damit die Syntheselaufzeit zu beschleunigen, erfolgt eine Vorauswahl der Kandidaten, wenn deren Anzahl einen definierbaren Wert überschreitet. Die Methode der Vorauswahl ist dem in dem AT&T Next-Gen TTS-System [CBSB00] verwendeten *Preselection filtering* der Kandidaten ähnlich. In [CBSB00] werden die aus Halbphonen bestehenden Einheiten anhand kontextueller Information gefiltert, die durch einfache Targetkostenberechnungen vor dem Auswahlalgorithmus zur Verfügung gestellt wird. Die Auswahlprozedur wird anschließend mit der reduzierten Kandidatenliste durchgeführt. In dem hier vorgestellten System erfolgt die Vorauswahl der Einheiten anhand der *sekundären* Merkmale, die auch für die Einheitenauswahl verwendet werden. Die Kandidatenlisten werden schrittweise für jedes Merkmal oder jede Merkmalskombination durchsucht. Liegt die Anzahl der Kandidaten noch immer über dem Maximalwert, wird die Suche iterativ für das Merkmalsset fortgeführt. Wenn nach einem Suchschritt die Kandidatenliste unter einem definierbaren Mindestwert liegt, wird der letzte Suchschritt rückgängig gemacht. Die Suche kann dann mit den verbliebenen Merkmalen oder Merkmalskombinationen weitergeführt werden. Als oberes Limit werden in der aktuellen Implementierung 10 Einheiten und als unteres Limit 2 Einheiten verwendet. Die Reihenfolge der *sekundären* Merkmale spielt für die Reduktion der Kandidatenlisten eine Rolle und kann daher variabel festgelegt werden. Die *sekundären* Merkmale in Tabelle 4.2 sind in der Reihenfol-

ge von oben nach unten angegeben, wie sie für die Suche verwendet werden. Der Suchalgorithmus ist im Folgenden angegeben (Algorithmus 4.1).

---

```
for all Merkmale do
  while |Kandidaten| > MAX_LIMIT do
    finde passenden Kandidaten
    erzeuge neue Liste Kandidaten*
  end while
  if |Kandidaten*| ≥ MIN_LIMIT then
    übernehme neue Liste Kandidaten*
  end if
end for
```

---

Algorithmus 4.1: *Preselection*-Algorithmus

Die Suche wird also in zwei Prozeduren aufgeteilt: eine *basic search* und eine *refined search*. Damit wird die Strategie verfolgt, zunächst anhand der *primären* Merkmale durch den PSM-Algorithmus eine Liste von Kandidaten zu erstellen. Aus dieser wird dann durch eine zweite Suchprozedur anhand der *sekundären* Merkmale ein optimales Kandidatenset generiert. Überschreitet zum Ende der Suche die Anzahl der Kandidaten den Maximalwert, kann ein optionales *Pruning* der Liste erfolgen, um den Aufwand der Kandidatenauswahl niedrig zu halten.

### 4.2.2 Kontextuelle Klassifikation

Der PSM-Algorithmus wird im Unterschied zu [TB97, Tay00] nur für Einheiten oberhalb der Segmentebene ausgeführt, da für Segmente die Kandidatenlisten zu groß sind und die Definition eines Merkmalssets zur Einschränkung der Kandidatenmenge zu komplex ist. Für die Auswahl der Segmente wird daher auf den im Festival Sprachsynthesystem zur Verfügung stehenden *Cluster*-Algorithmus zurückgegriffen (Abschnitt 3.1.3). Findet der PSM-Algorithmus ein Wort oder eine Silbe nicht in der Datenbank, ergänzt der *Cluster*-Algorithmus die fehlenden Segmente in der Targetstruktur. Dies ist häufig bei Eigennamen oder Fremdwörtern bzw. bei fremdsprachlichen Wörtern und Phrasen der Fall, die nicht domänenspezifisch (*out-of-vocabulary*) und daher nicht vom Korpus abgedeckt sind.

Für das *Clustering* der Segmente wird wie in [BT97a] eine gewichtete Frame-basierte Mahalanobische Distanz (Gleichung (3.1)) als akustisches Abstandsmaß ver-

wendet. Die Merkmalsvektoren enthalten als Parameter *Mel-Frequency Cepstral Coefficients (MFCC)*,  $f_0$ , Energie und die entsprechenden Delta-Werte. Das *Clustering* erfolgt mit Hilfe der CART-Technik. Für jedes Phonem wird anhand des linguistischen und phonetischen Kontextes ein Entscheidungsbaum generiert. Um den segmentellen Kontext zu berücksichtigen, wird für die Klassifikation ein symmetrisches Fenster verwendet, das das vorangehende und das nachfolgende Segment enthält. Tabelle 4.3 gibt eine Übersicht über die verwendeten Merkmale für die Segmentklassifikation.

<b>Ebene</b>	<b>Merkmal</b>
Segmentebene	Segmentidentität
	Vokal: Höhe, Länge, Position, Rundung
	Konsonant: Artikulationsort, Stimmhaftigkeit
	Lautdauer
	$f_0$ ( <i>pitch</i> )
	Silbenstruktur ( <i>onset, coda</i> )
	Position innerhalb der Silbe
Silbenebene	Wortakzent ( <i>stress</i> )
	Akzentuierung
	Phrasengrenze
	Position innerhalb des Wortes ( <i>initial, final</i> )
Wortebene	Phrasengrenze

Tabelle 4.3: Übersicht über die Merkmale für das *Clustering*

Optional können die Segmente in den *Clustern* durch *Pruning* reduziert werden. In der aktuellen Implementierung wird kein *Pruning* der *Cluster* durchgeführt.

Bei der Korpusuche wird anhand der linguistischen und phonetischen Information der Targetspezifikation ein *Cluster* für das jeweilige Phonem aus dem Entscheidungsbaum ausgewählt. Die Kandidaten des *Clusters* werden als Liste an die Targetstruktur angehängt.

Die Verwendung des *Cluster*-Algorithmus ist dadurch motiviert, dass dieser für uneingeschränkte Sprachsynthese konzipiert ist. Der PSM-Algorithmus ist hingegen eher für domänenspezifisches Textmaterial geeignet. Die Verbindung beider Verfahren reflektiert die Dichotomie der Anwendungsdomäne des SmartKom-Projektes.

### 4.2.3 Auswahlalgorithmus

Die Auswahl der optimalen Einheitensequenz erfolgt durch einen Viterbi-Algorithmus. Die Kandidaten der beiden Suchstrategien werden in ein Zustandsübergangnetzwerk transformiert. Der Viterbi-Algorithmus bestimmt den optimalen Pfad durch Minimierung der Target- und Konkatenierungskosten. Abbildung 4.3 zeigt die Viterbi-Suche für den obigen Beispielsatz:

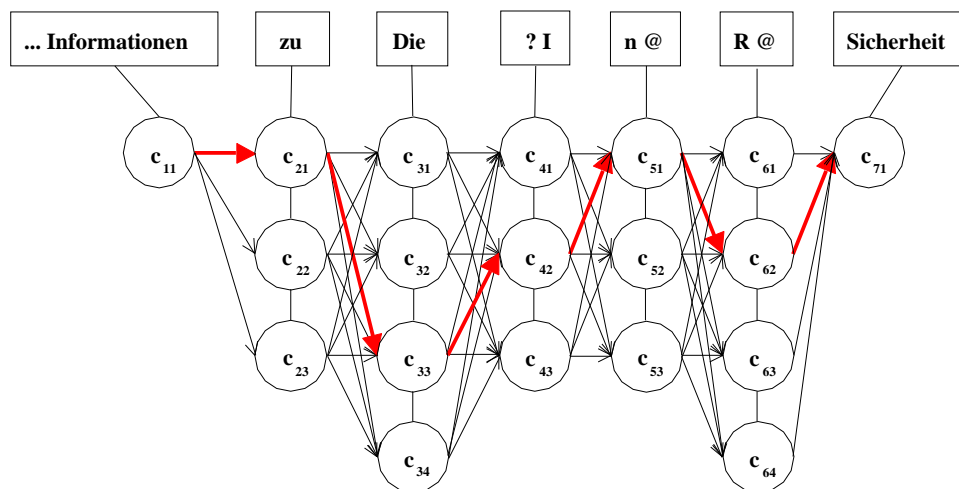


Abbildung 4.3: Viterbi-Suche

Die Konkatenierungskosten werden durch eine *optimal coupling*-Technik berechnet. Als akustisches Distanzmaß wird eine gewichtete euklidische Distanz verwendet. Die Parameter der Merkmalsvektoren sind dieselben wie beim *Clustering*. Die Suchregion des *optimal couplings* ist sowohl bei PSM-Einheiten als auch bei den *Cluster*-Einheiten innerhalb der Segmentgrenzen beschränkt. Für linguistische Einheiten oberhalb der Segmentebene wird die optimale Konkatenierungsstelle für das erste oder das letzte Segment bestimmt. Abbildung 4.4 zeigt schematisch das *optimal coupling* der beiden aufeinanderfolgenden Wörter "zu" und "die" aus dem obigen Beispiel.

Ob für längere Einheiten eine andere *optimal coupling*-Technik geeigneter ist [CI97], muss für diesen Ansatz experimentell überprüft werden.

Die Targetkosten der *Cluster*-Kandidaten werden durch den *Cluster*-Algorithmus *offline* berechnet (Gleichung (3.2)). Sie ergeben sich aus der akustischen Distanz eines Segments zum *Clusters*-Zentroiden. Für die Berechnung der Targetkosten der PSM-Kandidaten wird eine *scoring function* definiert, die den gewichteten Abstand

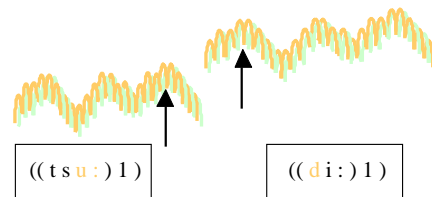


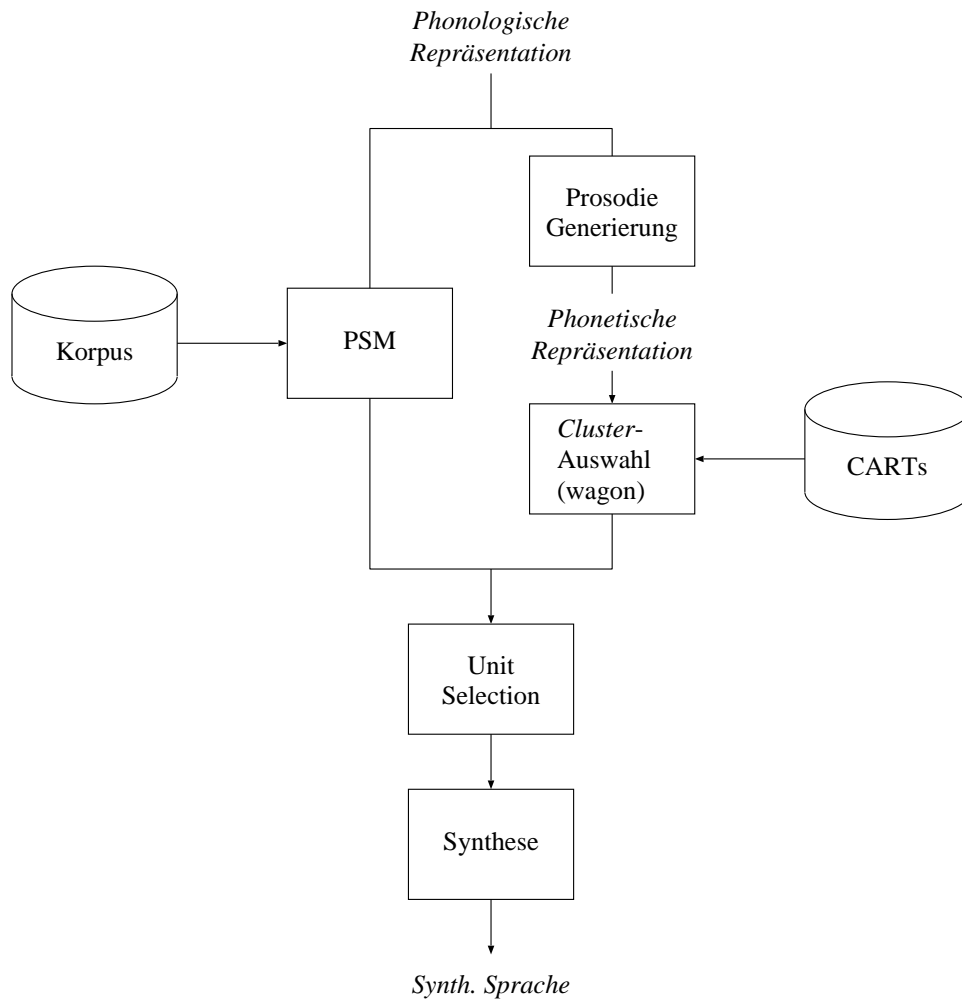
Abbildung 4.4: *Optimal coupling* von PSM-Einheiten

des Targets zum Kandidaten anhand der *sekundären* Merkmale bestimmt. Die Funktion liefert den Wert 0, wenn alle Merkmale übereinstimmen. Die Gewichtung der Merkmale erfolgt von Hand. Durch einen globalen Parameter kann eine relative Gewichtung der Target- und Konkatenierungskosten erfolgen. Die Ausgabe des *Unit Selection* Prozesses ist eine Folge von Referenzen zu Einheiten im Sprachkorpus, aus denen das Sprachsignal zusammengesetzt wird.

#### 4.2.4 Systemaufbau

Als Plattform für die Synthesekomponente des Smartkom-Projektes wird das Festival Sprachsynthesensystem [BT97b] (Abschnitt 3.1.3) verwendet. Die Modularität der Festival-Architektur bietet ein hohes Maß an Flexibilität, die es erlaubt, verschiedene Komponenten für Forschungs- und Testzwecke einfach zu integrieren. Für die am Institut für Maschinelle Sprachverarbeitung (IMS), Stuttgart, entwickelte deutsche Version des Festival Sprachsynthesensystems steht eine deutsche Diphonsynthesekomponente zur Verfügung. Diese wurde für das Smartkom-Projekt durch die hybride Unit Selection Komponente ersetzt. Abbildung 4.5 gibt einen Überblick über die Systemarchitektur.

Die linguistischen Module des Festival Sprachsynthesensystems erzeugen eine strukturierte linguistische Repräsentation in Form eines phonologischen Targetbaumes. Dieser stellt die Eingabe für die Prosodiekomponente dar, die eine phonetische Repräsentation mit akustischen Parametern wie Lautdauer und  $f_0$  generiert. Für die Datenbanksuche mit Hilfe des PSM-Verfahrens werden linguistische und phonologische Merkmale verwendet. Auf Segmentebene erfolgt die Suche von *Cluster*-Kandidaten anhand phonologischer und phonetischer Kontextinformation. Die Auswahl der Kandidaten erfolgt durch den Viterbi-Algorithmus, der eine Sequenz von Referenzen zu unterschiedlichen linguistischen Einheiten im Sprachkorpus liefert, aus denen das Sprachsignal zusammengesetzt wird. Die den ausgewählten Einhei-

Abbildung 4.5: Aufbau des hybriden *Unit Selection* Systems

ten entsprechenden Sprachsignale werden verknüpft und die Konkatenierungsstellen (optional) durch eine Fensterfunktion geglättet. Es werden keine prosodischen Modifikationen durch Signalverarbeitungstechniken eingesetzt. Die von den Prosodiemodulen erzeugte phonetische Repräsentation kann jedoch wie in [TB97, Tay00] für anschließende Signalmodifikationen genutzt werden.

### 4.3 Ergebnisse und Diskussion

Eine formale empirische Evaluation des Synthesystems wurde bislang nicht durchgeführt. Anhand einer statistischen Erhebung in [SBK<sup>+</sup>03] wurde die mittlere Einheitenlänge festgestellt, die ein Prädiktor für die Anzahl der Konkatenationsstellen und damit für die Synthesequalität ist. Bei einem Testset von 31 verschiedenen Sätzen mit z.T. domänenspezifischem Material<sup>4</sup> wurde für die vom PSM-Algorithmus gefundenen Einheiten eine durchschnittliche Einheitenlänge von 5.5 Segmenten berechnet. Die mittlere Einheitenlänge nach der Konkatenation der Einheiten ergab einen höheren Wert von 6 Segmenten. Dies ist darauf zurückzuführen, dass Einheiten ausgewählt wurden, die im Korpus aufeinanderfolgen. Die längste Einheit ist Teil der SmartKom-Begrüßungsformel “*Herzlich Willkommen beim SmartKom-Informationssystem*” und hat eine Länge von 46 Segmenten. Die Ergebnisse zeigen, dass durch den Einsatz des PSM-Verfahrens für domänenspezifisches Material wesentliche längere Einheiten gefunden werden als bei üblichen *Unit Selection* Systemen. Damit kann die Effizienz der Einheitenauswahl und die Synthesezeit reduziert werden. Die perzeptuelle Qualität der Sprachausgabe ist durch die geringere Anzahl an Konkatenierungsstellen und die implizit modellierte natürliche Prosodie sehr gut. Informelle Hörtests mit domänenspezifischen Beispieldialogen und einigen domänenunabhängigen Testsätzen ergaben, dass die perzeptuelle Qualität der Sprachausgabe als sehr gut bewertet wird.

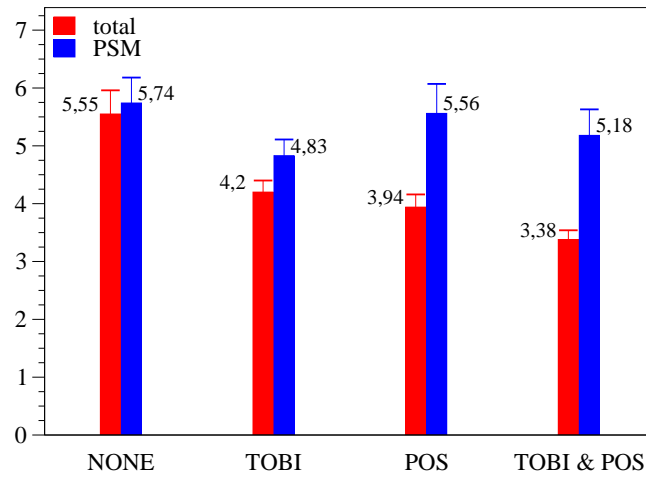
Ein bislang nicht gelöstes Problem ist die Unterteilung der linguistischen Merkmale in *primäre* Merkmale für die Einheitensuche und *sekundäre* Merkmale für die *Preselection* und die Einheitenauswahl. Eine Erweiterung der *primären* Merkmale für die Einheitensuche ist sinnvoll, um Einheiten mit geringem Vorkommen im Korpus von der Einheitensuche auszuschließen, wenn sie der Targetbeschreibung nicht ausreichend entsprechen. Findet der PSM-Algorithmus auf einer Ebene keine passende

---

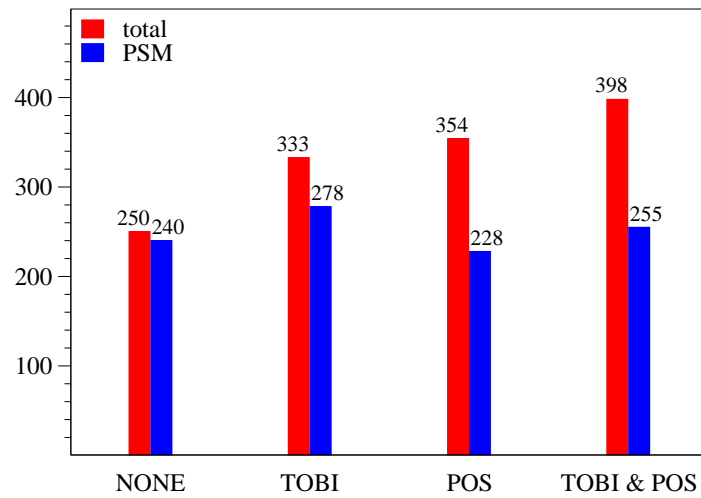
<sup>4</sup> Die Beispielsätze für die Synthese finden sich unter <http://www.ims.uni-stuttgart.de/phonetik/unitselection>.

Einheit, wird die Suche auf der darunterliegenden Ebene fortgesetzt. Ein Beispiel hierfür sind Wörter oder Silben mit geringem Vorkommen wie Eigennamen oder Komposita, die nur in einem prosodischen Kontext (z.B. mit bestimmtem Akzent) im Korpus vorkommen. In diesem Fall könnte eine Synthese von kleineren Einheiten, die die Targetspezifikation gut approximieren, bessere Resultate erzeugen. Die Verwendung eines rigideren Sets *primärer* Merkmale führt jedoch umgekehrt dazu, dass sehr häufig auf höheren Ebenen keine Kandidaten gefunden werden. Da die Synthese von längeren Einheiten mit geringeren Konkatenierungsstellen und Signalmodifikationen von Hörern bevorzugt wird, wirkt sich die Verwendung zusätzlicher *primärer* Merkmale nachteilig auf die Sprachqualität aus. Die Einheitenlänge nimmt durch Verwendung eines größeren Sets an *primären* Merkmalen stark ab. Abbildung 4.6(a) zeigt den Einfluss *primärer* Merkmale auf die mittlere Länge der Einheiten. Angegeben ist die mittlere Einheitenlänge für die PSM-Einheiten (*psm*) sowie für alle Einheiten (*total*) nach der Konkatenation. Für das oben genannte Testset mit 31 Sätzen wurden als *primäre* Merkmale zusätzlich Akzent und Grenzton (*TOBI*), sowie positionelle Merkmale (z.B. Position innerhalb des Wortes, *POS*) verwendet. Die zusätzlichen Merkmale werden für alle Ebenen ausgenommen der Phrasenebene berücksichtigt. Die Abnahme der Einheitenlänge insgesamt (*total*) bei der Verwendung zusätzlicher *primärer* Merkmale ist deutlich erkennbar. Die mittlere Einheitenlänge der PSM-Einheiten nimmt ebenfalls ab (vgl. die Werte 5.74 für *NONE* vs. 4.83 für *TOBI* bei der Hinzunahme von Akzent und Grenzton als *primäre* Merkmale). Es werden mehr PSM-Einheiten auf den unteren Ebenen gefunden. Die höheren PSM-Werte für die zusätzlichen Merkmalskombinationen *POS* und *TOBI & POS* sind darauf zurückzuführen, dass die Einheitenanzahl der PSM-Einheiten insbesondere auf der Wortebene und Silbenebene stark abnimmt. Der PSM-Algorithmus findet hauptsächlich noch feste Phrasen und Prompts. Abbildung 4.6(b) zeigt entsprechend, dass die mittlere Einheitenanzahl bei der Verwendung zusätzlicher *primärer* Merkmale für die Einheiten insgesamt (*total*) zunimmt.

Die Verwendung zusätzlicher *sekundärer* Merkmale wirkt sich nachteilig auf die Effizienz der *Preselection* und der Viterbi-Suche aus. Die Komplexität des Viterbi-Algorithmus hängt von der Anzahl der Kandidaten und der zu berechnenden Kostenfunktionen ab. Aufgabe des *Preselection*-Algorithmus ist es, anhand der *sekundären* Merkmale ein optimales Kandidatenset zu generieren und damit die Komplexität der Einheitenauswahl zu reduzieren. Dies ist vor allem für linguistische Einheiten, die sehr häufig im Korpus vorkommen, nötig. Beispielsweise werden für häufige Funktionswörter, Prä- und Suffixe bis zu mehrere hundert Kandidaten gefunden. Für



(a) Einheitenlänge



(b) Einheitenanzahl

Abbildung 4.6: Einfluss primärer Merkmale

diese hochfrequenten Einheiten ist eine Spezifikation anhand der bisher implementierten Merkmale nicht ausreichend, um eine genügend kleine Kandidatenliste zu erstellen. Da der *Preselection*-Algorithmus eine iterative Merkmalsuche auf den Kandidatenlisten ausführt, erhöht eine Erweiterung des Merkmalssets die Rechenzeit der Suche. In der aktuellen Implementierung kann die Kandidatenliste durch (optionales) *Pruning* für die Viterbi-Suche reduziert werden. Hierbei besteht die Gefahr, dass optimale Kandidaten mit niedrigen Konkatenierungskosten, die den phonetischen Kontext sehr gut approximieren, abgeschnitten werden. Um den Aufwand der Einheitenauswahl bei gleichbleibender Synthesequalität zu reduzieren, könnte eine strahlgesteuerte Viterbi-Suche (*beam search*) oder der A\*-Algorithmus eingesetzt werden (Abschnitt 2.2.1). Bei der *beam search* muss der Suchraum durch ein geeignetes Kostenlimit eingeschränkt werden. Aufgrund der Heterogenität der Einheiten variiert der Wertebereich der anfallenden Kosten stark. Die Bestimmung eines geeigneten Schätzwertes für eine obere Schranke der Kosten ist daher keine triviale Aufgabe. Analoges gilt für den A\*-Algorithmus, bei dem die gesteuerte Suche auf der Schätzung der kostengünstigsten Einheitenfolge basiert. Für die Berechnung der verbleibenden Restkosten müsste hierbei zwischen den verschiedenen Einheiten unterschieden werden, da die Berechnungsgrundlage der Targetkosten für die *PSM*-Einheiten und die *Cluster*-Einheiten unterschiedlich ist. Ob diese Verfahren eine Verbesserung der Effizienz und der Qualität der Sprachsynthese bewirken, muss noch überprüft werden.

Durch die Verwendung kompakter phonologischer Merkmalsbeschreibungen ist die Anzahl der Parameter der *Unit Selection* vergleichsweise gering. Gewichtet werden die *sekundären* Merkmale für die Berechnung der Targetkosten und die Parameter der akustischen Merkmalsvektoren für die Berechnung der Konkatenierungskosten und für die kontextuelle Klassifikation der Segmente. Die relative Gewichtung von Target- und Konkatenierungskosten wird anhand eines globalen Parameters bestimmt. Aufgrund der geringen Anzahl an Parametern können die Gewichte einfach von Hand gesetzt werden. Eine automatisches Training der Gewichte ist bislang nicht implementiert. Problematisch für ein automatisches Training ist die Heterogenität der Merkmale und das Fehlen eines geeigneten objektiven Distanzmaßes. Ein Vergleichsexperiment, bei dem der Abstand des synthetisierten Sprachsignals zur Originaläußerung anhand des in [BT97a] verwendeten akustischen Distanzmaßes (Gleichung (3.1)) berechnet wurde, zeigt, dass dieses als objektives Distanzmaß für ein automatisches Training nicht geeignet ist. Bei dem Experiment wurde die Signalqualität des hybriden *Unit Selection* Verfahrens (*psm*) mit der der *Unit Selection*

Komponente des Festival Sprachsynthesystems (*cl*) (Abschnitt 3.1.3) verglichen. Abbildung 4.7 gibt die Werte für 11 Beispieläußerungen an.

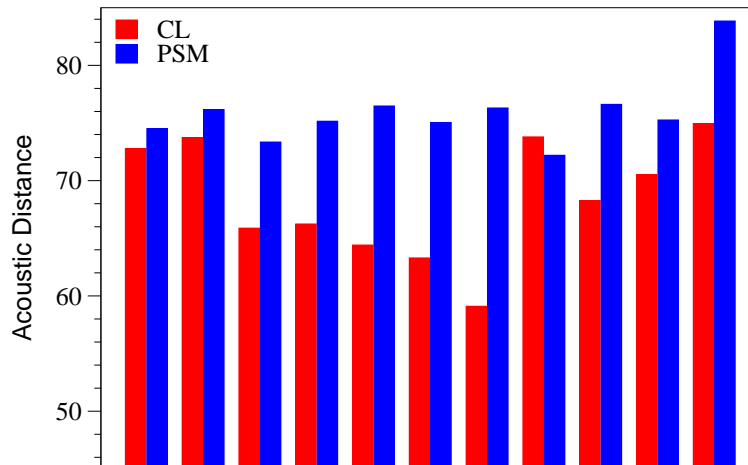


Abbildung 4.7: Vergleich der Ansätze

Das Diagramm zeigt eine deutliche Präferenz des objektiven Distanzmaßes für die synthetisierten Äußerungen der *Unit Selection* des Festival Systems. Dies ist z.T. darauf zurückzuführen, dass das akustische Distanzmaß (Gleichung (3.1)) ebenfalls für das automatische Training beim *Cluster*-Algorithmus verwendet wurde. Eine Bewertung der Sprachqualität der Äußerungen durch Hörer ergab jedoch ein sehr viel differenzierteres Bild und eine stärkere Präferenz für die synthetisierten Äußerungen des hybriden *Unit Selection* Verfahrens.

## 5 Resumée

Für die Bewertung der Performanz eines Synthesystems ist eine Einschätzung des Schwierigkeitsgrades der Anwendung erforderlich. Ein Maß für den Schwierigkeitsgrad der Domäne kann als Indikator für die zu erwartende Synthesequalität eines Sprachsynthesystems dienen. Taylor [Tay00] schlägt als ein Maß die Vokabulargröße und die Perplexität, die die Wortentropie eines Korpus angibt und den Grad der Regularität eines Korpus beschreibt, vor. Die Perplexität kann als Prädiktor für die Anzahl der Konkatenationsstellen zwischen Einheiten fungieren. Eine niedrige Perplexität einer Domäne gibt an, dass die Anzahl der möglichen Wort- $N$ -Gramme höher ist und damit die Wahrscheinlichkeit für das Vorkommen von Einheiten in verschiedenen phonetischen und prosodischen Kontexten steigt. Dieses Maß berücksichtigt jedoch nicht ausreichend die LNRE-Eigenschaften linguistischer Einheiten. Das SmartKom-Dialogsystem stellt hohe Anforderungen an die Sprachsyntheseleistung, da die eingeschränkte Anwendungsdomäne aufgrund von Eigennamen und fremdsprachlichen Ausdrücken ein offenes Vokabular umfasst. Durch die hohe Frequenz von unbekanntem Vokabular ist der Schwierigkeitsgrad der Domäne zwischen einer Synthese mit eingeschränkter Domäne, wie sie für das *Phonological Structure Matching* Verfahren von Taylor und Black [TB97, Tay00] vorliegt, und einer unbeschränkten Sprachsynthese einzuordnen. Abbildung 5.1 zeigt schematisch eine Einordnung der Syntheseleistung des Systems.

Der Schwierigkeitsgrad und die spezifische Struktur der Anwendungsdomäne verlangt eine Kombination verschiedener *Unit Selection* Verfahren. Es wurden zwei unterschiedliche *Unit Selection* Ansätze miteinander kombiniert, um die Dichotomie einer eingeschränkten Anwendungsdomäne mit unbeschränktem Vokabular zu reflektieren. Der *Phonological Structure Matching* Algorithmus bietet eine hohe Sprachqualität für domänenspezifische Äußerungen und Phrasen und reduziert den Aufwand der Einheitenauswahl. Der *Cluster*-Algorithmus zeichnet sich durch Robustheit gegenüber unbeschränktem Textmaterial aus. Für die Verknüpfung der beiden Algorithmen waren verschiedene Modifikationen notwendig. Durch den Einsatz einer *Preselection*-Komponente bei der Einheitensuche konnte die Effizienz und die

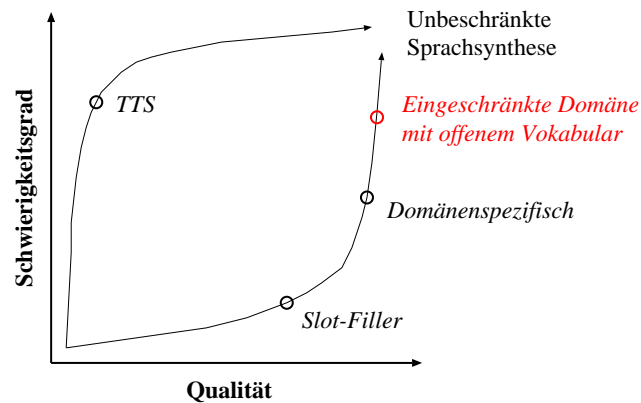


Abbildung 5.1: Einordnung des hybriden Syntheseansatzes

Sprachqualität der Synthese verbessert werden. Die Ergebnisse zeigen, dass durch den Einsatz des PSM-Verfahrens für domänenspezifisches Material wesentliche längere Einheiten gefunden werden als bei üblichen *Unit Selection* Systemen. Damit kann die Effizienz der Einheitenwahl und die Synthesezeit im Vergleich zu einem ausschließlich auf kontextueller Klassifikation beruhendem Verfahren wie dem Festival Sprachsynthesystem (Abschnitt 3.1.3) deutlich reduziert werden. Die perzeptuelle Qualität der Sprachausgabe ist durch die geringere Anzahl an Konkatenierungsstellen und die implizit modellierte natürliche Prosodie sehr gut. Durch den Einsatz des *Cluster*-Algorithmus konnte eine skalierbare Synthesequalität erreicht werden, die für unbekanntes Textmaterial eine graduell reduzierte Sprachqualität aufweist. Informelle Hörtests mit domänenspezifischen Beispieldialogen und einigen domänenunabhängigen Testsätzen ergaben, dass die perzeptuelle Qualität der Sprachausgabe als sehr gut bewertet wird.

# Danksagung

Ich danke allen, die zum Gelingen dieser Arbeit beigetragen und mich unterstützt haben. Mein besonderer Dank gilt



**Prof. Dr. Grzegorz Dogil**

für die Realisierung dieser Arbeit  
an seinem Lehrstuhl,



**PD Dr. Bernd Möbius**

für anregende Diskussionen und  
Vorschläge,



**Dipl.-Ling. Antje Schweitzer**

für intensive fachliche Betreuung  
und Hilfestellung,

**Peter Dillinger**

für technische Unterstützung und  
viel Geduld.



# Literaturverzeichnis

- [BC95] BLACK, ALAN W. und W. NICK CAMPBELL: *Optimising selection of units from speech databases for concatenative synthesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 1, Seiten 581–584, Madrid, Spanien, 1995.
- [BC99] BEUTNAGEL, MARK und ALISTAIR D. CONKIE: *Interaction of Units in a Unit Selection Database*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 3, Seiten 1063–1066, Budapest, Ungarn, 1999.
- [BCS<sup>+</sup>99] BEUTNAGEL, MARK, ALISTAIR D. CONKIE, JÜRGEN SCHROETER, YANNIS STYLIANOU und ANN K. SYRDAL: *The AT&T Next Generation TTS System*. In: *Proceedings of the Joint Acoustical Society of America (ASA)*, Berlin, 1999.
- [BJ98] BREEN, ANDREW P. und PETER JACKSON: *Non-uniform unit selection and the similarity metric within BT's Laureate TTS system*. In: *Proceedings of the 3<sup>rd</sup> ESCA/COCOSDA Workshop on Speech Synthesis*, Jenolan Caves, NSW, Australien, 1998.
- [BL00a] BLACK, ALAN W. und KEVIN A. LENZO: *Building voices in the Festival speech synthesis system*. Veröffentlicht unter <http://festvox.org>, 2000.
- [BL00b] BLACK, ALAN W. und KEVIN A. LENZO: *Limited Domain Synthesis*. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, 2000.
- [BMR99] BEUTNAGEL, MARK, MEHRYAR MOHRI und MICHAEL RILEY: *Rapid unit selection from a large speech corpus for concatenative speech syn-*

- thesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 2, Seiten 607–610, Budapest, Ungarn, 1999.
- [BT94] BLACK, ALAN W. und PAUL TAYLOR: *CHATR: a generic speech synthesis system*. In: *Proceedings of the International Conference on Computational Linguistics*, Band 2, Seiten 983–986, Kyoto, Japan, 1994.
- [BT97a] BLACK, ALAN W. und PAUL TAYLOR: *Automatically clustering similar units for unit selection in speech synthesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 2, Seiten 601–604, Rhodos, Griechenland, 1997.
- [BT97b] BLACK, ALAN W. und PAUL TAYLOR: *The Festival Speech Synthesis System: system documentation*. Technischer Bericht HCRC/TR-83, Human Communication Research Centre. University of Edinburgh, Edinburgh, Schottland, GB, 1997. Veröffentlicht unter <http://www.cstr.ed.ac.uk/projects/festival.html>.
- [CBSB00] CONKIE, ALISTAIR D., MARK C. BEUTNAGEL, ANN K. SYRDAL und PHILIP E. BROWN: *Preselection of candidate units in a unit selection-based text-to-speech synthesis system*. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, 2000.
- [CI97] CONKIE, ALISTAIR D. und STEPHEN ISARD: *Optimal Coupling of Diphones*. In: SANTEN, J. VAN, R. SPROAT, J. OLIVE und J. HIRSCHBERG (Herausgeber): *Progress in Speech Synthesis*, Seiten 293–305. Springer-Verlag, New York, NY, USA, 1997.
- [Con99] CONKIE, ALISTAIR D.: *Robust unit selection for speech synthesis*. In: *Proceedings of the Joint Acoustical Society of America (ASA)*, Berlin, 1999.
- [DE98] DONOVAN, ROBERT E. und ELLEN M. EIDE: *The IBM trainable speech synthesis system*. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australien, 1998.

- [DW99] DONOVAN, ROBERT E. und PHIL C. WOODLAND: *A hidden Markov-model-based trainable speech synthesizer*. *Computer, Speech and Language*, 13:223–241, 1999.
- [Fan60] FANT, GUNNAR: *The Acoustic Theory of Speech Production*. Mouton & Co., The Hague, 1960.
- [HB96] HUNT, ANDREW J. und ALAN W. BLACK: *Unit selection in a concatenative speech synthesis system using a large speech database*. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Atlanta, GA, USA, 1996.
- [Jel97] JELINEK, FREDERICK: *Statistical Methods for Speech Recognition*. Language, Speech, and Communication. MIT Press, London, 1997.
- [MC90] MOULINES, ERIC und FRANCIS CHARPENTIER: *Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones*. *Speech Communication*, 9(5-6):453–467, 1990.
- [MCW98] MACON, MICHAEL W., ANDREW E. CRONK und JOHAN WOUTERS: *Generalization and discrimination in tree-structured unit selection*. In: *Proceedings of the 3<sup>rd</sup> ESCA/COCOSDA Workshop on Speech Synthesis*, Seiten 195–200, Jenolan Caves, NSW, Australien, 1998.
- [MH99] MERON, YORAM und KEIKICHI HIROSE: *Efficient weight training for selection based synthesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 5, Seiten 2319–2322, Budapest, Ungarn, 1999.
- [Möb00] MÖBIUS, BERND: *Corpus-Based Speech Synthesis: Methods and Challenges*. Band 6(4) der Reihe *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS)*, Universität Stuttgart, Seiten 87–116. 2000.
- [Möb01] MÖBIUS, BERND: *Sprachsynthesysteme*. In: KAI-UWE CARSTENSEN ET AL. (Herausgeber): *Computerlinguistik und Sprachtechnologie: Eine Einführung*, Seiten 462–468. Spektrum Akademischer Verlag, Heidelberg, 2001.

- [Möb03] MÖBIUS, BERND: *Rare Events and Closed Domains: Two Delicate Concepts in Speech Synthesis*. International Journal of Speech Technology, 6(1):57–71, 2003.
- [Nak94] NAKAJIMA, SHIN-YA: *Automatic synthesis unit generation for English speech synthesis based on multi-layered context oriented clustering*. Speech Communication, 14:313–324, 1994.
- [NH88] NAKAJIMA, SHIN-YA und HIROSHI HAMADA: *Automatic generation of synthesis units based on context oriented clustering*. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 659–662, New York, NY, USA, 1988.
- [Sag88] SAGISAKA, YOSHINORI: *Speech synthesis by rule using an optimal selection of non-uniform synthesis units*. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seiten 679–682, New York, NY, USA, 1988.
- [SBK<sup>+</sup>03] SCHWEITZER, ANTJE, NORBERT BRAUNSCHWEILER, TANJA KLANKERT, BERND MÖBIUS und BETTINA SÄUBERLICH: *Restricted Unlimited Domain Synthesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Genf, Schweiz, 2003.
- [SS01] STYLIANOU, YANNIS und ANN K. SYRDAL: *Perceptual and objective detection of discontinuities in concatenative speech synthesis*. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, USA, 2001.
- [ST95] SCHUKAT-TALAMAZZINI, ERNST GÜNTER: *Automatische Spracherkennung*. Vieweg, 1995.
- [Tay00] TAYLOR, PAUL: *Concept-to-Speech Synthesis by Phonological Structure Matching*. In: *Philosophical Transactions of the Royal Society*, Band 356 (1769), Seiten 1403–1416, 2000.
- [TB97] TAYLOR, PAUL und ALAN W. BLACK: *Speech synthesis by phonological structure matching*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 2, Seiten 623–626, Budapest, Ungarn, 1997.

- [TKS90] TAKEDA, KAZUYA, ABE KATSUO und YOSHINORI SAGISAKA: *On unit selection algorithms and their evaluation in non-uniform speech synthesis*. In: *Proceedings of the ESCA Workshop on Speech Synthesis*, Seiten 35–38, Autrans, Frankreich, 1990.
- [vS97a] SANTEN, JAN P.H. VAN: *Combinatorial issues in text-to-speech synthesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 5, Seiten 2511–2514, Rhodos, Griechenland, 1997.
- [vS97b] SANTEN, JAN P.H. VAN: *Prosodic Modeling in Text-to-Speech Synthesis*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Rhodos, Griechenland, 1997.
- [vSB97] SANTEN, JAN P.H. VAN und ADAM L. BUCHSBAUM: *Methods for optimal text selection*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 2, Seiten 553–556, Rhodos, Griechenland, 1997.
- [WCIS93] WANG, W. J., W. NICK CAMPBELL, NAOTO IWAHASHI und YOSHINORI SAGISAKA: *Tree-based unit selection for English speech synthesis*. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Band 2, Seiten 191–194, Minneapolis, MN, USA, 1993.
- [WRB01] WAHLSTER, WOLFGANG, NORBERT REITHINGER und ANSELM BLOCHER: *SmartKom: Multimodal communication with a life-like character*. In: *Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech)*, Band 3, Seiten 1547–1550, Aalborg, Dänemark, 2001.



# Abkürzungen

CART	<i>Classification and Regression Tree</i>
CTS	<i>Context-to-speech</i>
EM	<i>Estimation Maximization</i>
EPG	<i>Electronic Programming Guide</i>
HMM	<i>Hidden Markov Modell</i>
LPC	<i>Linear Predictive Coding</i>
LNRE	<i>Large Number of Rare Events</i>
MFCC	<i>Mel-Frequency Cepstral Coefficients</i>
NLG	<i>Natural Language Generation</i>
POS	<i>Part-of-speech</i>
PSM	<i>Phonological Structure Matching</i>
PSOLA	<i>Pitch Synchronous Overlap Add</i>
TTS	<i>Text-to-speech</i>