

Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Azenbergstraße 12
70174 Stuttgart

SS 2006

Studienarbeit

Eine autoritative Diphonliste für die deutsche Sprachsynthese

Manuel Weiß

6. Februar – 6. Mai 2006

Studienarbeit Nr. 51
Betreuerin: Antje Schweitzer
Prüfer: PD Dr. phil. Bernd Möbius

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe.

Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet.

Ziel dieser Studienarbeit ist es, eine möglichst umfassende Diphonliste für die deutsche Sprachsynthese zu erstellen, die Phonotaktik und Kontexteffekte berücksichtigt. Dabei sollen auch fremdsprachliche Laute in das Phonem-Inventar aufgenommen werden, die nicht auf Laute des Deutschen abgebildet werden können. Nachdem ein Phoneset definiert ist, soll die Erstellung der Diphonliste sowie die Generierung der Trägerwörter weitgehend automatisch erfolgen.

Inhaltsverzeichnis

1. Einleitung	6
2. Theorie	7
2.1. Warum benötigt man ein Diphon-Inventar?	7
2.2. Grundlagen: Phone und Diphone	7
2.3. Diphone in der Sprachsynthese	7
2.4. Phonotaktische Beschränkungen	9
2.5. Minimale Koartikulation	9
2.6. Koartikulatorische Effekte	10
2.7. Regelbasierte vs. lexikonbasierte Liste	11
2.8. Fremdsprachliche Laute	11
2.9. Trägerwort-Generierung	11
3. Praktische Durchführung	13
3.1. Das Phoneset	13
3.2. Regelbasierte Generierung der Diphonliste	14
3.2.1. Zu viele Diphone...	14
3.2.2. Reduktion durch Phonotaktik und minimale Koartikulation	16
3.2.3. Berücksichtigung koartikulatorischer Effekte	17
3.3. Empirische Untersuchung zu Diphonvorkommen	19
3.3.1. Deutsches Aussprachelexikon: gcelex-onomastica	19
3.3.2. Englisches Aussprachelexikon: cmudict	20
3.4. Vergleich der regelbasierten und empirischen Diphonliste	24
4. Diskussion der Ergebnisse	25
4.1. Systematische Lücken	25
4.2. Diphon-Generierung durch Regeln?	25
4.3. Berücksichtigung koartikulatorischer Effekte	25
4.4. Das endgültige Diphon-Inventar	26
5. Zusammenfassung	27
A. Formatdefinitionen	28
A.1. Das Phon-Inventar	28
A.2. Die Ausschlussregeln	30
A.3. Die Kontextregeln	31
A.4. Ausschnitt aus der Diphonliste	31

A.5. Ausschnitt aus der Liste der kontextsensitiven Diphone	32
B. Die Perl-Skripten	33
B.1. Generierung der Diphonliste	33
B.2. Generierung von kontextsensitiven Diphonen	36
B.3. Diphone im Aussprachelexikon	39
B.4. Mapping Arpabet nach SAMPA	41
B.5. Deutsche Lautmodellierung	42

1. Einleitung

Personally I do not believe that concatenative synthesis of whatever type, unless highly adaptable, will ultimately be sufficiently flexible to serve the needs for many future applications even if collecting the required concatenative units becomes easier and easier.

Louis Pols, im Vorwort von [SPROAT 1998]

Trotz dieser sehr pessimistischen Worte basieren alle heutigen Sprachsynthesysteme auf der Verkettung von zuvor aufgenommenen Einheiten und es hat sich gezeigt, dass man durch ein sorgfältiges Design des Einheiteninventars eine beeindruckend gute Sprachqualität erreichen kann. In den folgenden Kapiteln soll dargestellt werden, was dabei beachtet werden muss und es soll der Versuch unternommen werden, die Erstellung des Inventars – ausgehend von einem zuvor festgelegten Phoneset – weitgehend zu automatisieren.

Seit den frühen Tagen der auf aufgenommenen Einheiten basierenden Sprachsynthese (s. z.B. [KÜPFMÜLLER und WARNS 1956]) war man auf der Suche nach der idealen akustischen Einheit, aus der sich beliebige komplexe Äußerungen aufbauen lassen. Einerseits sollte die einzelne Einheit möglichst lang sein, um möglichst wenig Verkettungsstellen zu haben und der Koartikulation der Laute Rechnung zu tragen. Andererseits sollte die Einheit möglichst kurz sein, um die Anzahl der Einheiten in einem vernünftigen Rahmen zu halten (sie steigt exponentiell mit der Einheitenlänge).

Viele Sprachsynthesysteme verwenden deshalb Diphone als Kompromiss. Es gibt aber auch Sprachsynthesysteme auf Halbsilben- oder Silbenbasis und mit gemischtem Einheiteninventar.¹ Moderne Systeme arbeiten aber meist mit Unit Selection, d.h. aus einem großen Corpus werden zur Laufzeit die optimalen (und möglichst langen) Einheiten gesucht. Im Rahmen dieser Studienarbeit soll untersucht werden, wie die Erstellung eines Diphoninventars automatisiert werden kann und es sollen Software-Tools erstellt werden, die es erlauben, aus einem definierten Phoneset – unter Berücksichtigung von Koartikulation und Phonotaktik – eine Liste aller für die Synthese notwendigen Diphone zu generieren.

In Kapitel 2 werde ich zunächst die theoretischen Hintergründe darstellen, um dann in Kapitel 3 zu erläutern, wie ich im Einzelnen vorgegangen bin. In Kapitel 4 findet sich schließlich eine Diskussion der Ergebnisse. Der Anhang enthält einige Formatdefinitionen sowie die erstellten Perl-Skripten.

¹eine umfangreiche Aufstellung der verschiedenen Systeme findet sich in [PORTELE 1996], S. 7-13

2. Theorie

2.1. Warum benötigt man ein Diphon-Inventar?

Auch wenn moderne Sprachsynthesysteme zunehmend auf Unit Selection basieren, so ist es doch wichtig, zumindest alle Diphone einer Sprache abzudecken, damit im worst case zumindest die Sprachqualität einer Diphonsynthese erreicht wird. Dafür ist es unumgänglich, eine vollständige, autoritative Diphonliste zu haben, damit die Abdeckung der Diphone (für ein gegebenes Phoneteset) durch das Aufnahmecorpus gewährleistet ist.

Es gab immer wieder Stimmen, die die Synthese mit Diphonen für trivial hielten und für nicht sehr ergiebig, was die Weiterentwicklung von artikulatorischer und akustischer Theorie betrifft ([ALLEN 1992], zitiert nach [MÖBIUS 2001]). Allerdings wird sich im Folgenden zeigen, dass man in die sorgfältige Konstruktion eines Diphon-Inventars sehr viel Arbeit und phonetisches Wissen stecken kann.

2.2. Grundlagen: Phone und Diphone

Ein *Phon* ist die grundlegende lautliche Einheit, aus der höhere Einheiten wie Silben und Wörter zusammengesetzt werden. In meiner Studienarbeit werde ich durchgehend die SAMPA-Notation verwenden, da ich diese auch in den Perl-Skripten benutzt habe; als Konversionstabelle von und nach der IPA-Notation mag Tabelle 2.1 dienen.

Es kommt vor, dass zwei (oder mehr) Phone Allophone eines Phonems sind, d.h. zwischen ihnen wird regelgeleitet je nach Kontext ausgewählt und sie stehen zueinander in konträrer Verteilung. Ein klassisches Beispiel im Deutschen sind z.B. die beiden Phone $[x]$ und $[C]$, die Allophone sind und gemeinsam das Phonem $/x/$ bilden. Hier wird die Verteilung durch den vorangehenden Vokal bestimmt: auf einen Hinterzungenvokal folgt $[x]$, auf einen Vorderzungenvokal folgt $[C]$: $[I C]$ vs. $[b u: x]$.

Unter einem *Diphon* versteht man eine akustische Einheit, die die Transitionsstelle zwischen zwei Phonem umfasst; Diphone sind eine wichtige Einheit in der Sprachsynthese (s. 2.3).

2.3. Diphone in der Sprachsynthese

Wie oben (2.2) bereits geschildert, hat eine jede Sprache eine Menge von Phonem, aus denen alle Wörter dieser Sprache aufgebaut sind. In der Sprachsynthese kann man sich dies zunutze machen und aus einer begrenzten Menge von Einheiten beliebig

2. Theorie

	IPA	SAMPA		IPA	SAMPA		IPA	SAMPA	
Vokale	ɪ	I	Diphtonge	aɪ	aI	Plosive	p	p	
	i:	i:		aʊ	aU		b	b	
	ʊ	U		ɔɻ	OY		t	t	
	u:	u:		(englisch)	eɪ		eI	d	d
	ɻ	Y		(englisch)	ɔɪ		OI	k	k
	y:	y:		(englisch)	əʊ		@U	g	g
	ɛ:	E:		(englisch)	ɪə	I@	Frikative	f	f
	e:	e:		(englisch)	eə	e@		v	v
	o:	o:		(englisch)	ʊə	U@		s	s
	œ	9		(englisch)	oʊ	oU		z	z
	ø:	2:				ʃ		S	
	a:	a:	Schwa	ɐ	6	ʒ		Z	
	ɔ	O		ə	@	x		x	
	(französisch) œ̃	9~				ç		C	
	(französisch) ê	e~	Halbvokal	j	j	h		h	
	(französisch) ã	a~				(englisch) w		w	
	(französisch) õ	o~				(englisch) θ	T		
	(englisch) æ	{	Nasale	ŋ	N	(englisch) ð	D		
	(englisch) ɑ:	Ä:		m	m	Liquide	ʀ	R	
	(englisch) ɒ	Q		n	n		l	l	
(englisch) ʌ	V				(englisch) r		r		
(englisch) ɜ:	3:	Glottaler Stop	ʔ	ʔ	(englisch) L		L		

Aus Gründen der Übersichtlichkeit wurden die kurzen Vokalvarianten weggelassen.
Die Affrikaten sind als zwei einzelne Phone transkribiert.

Tabelle 2.1.: IPA-SAMPA-Konversionstabelle (Quellen: [Edinburgh] u.a.)

viele Wörter produzieren. Allerdings bemerkte man schnell, dass eine Aneinanderreihung von einzelnen Phonemen zu einem sehr unnatürlichen Klang führt, da es an den Verkettungsstellen starke Sprünge im Spektrogramm gibt. Dies liegt daran, dass gerade der Grenzbereich zwischen zwei Phonemen die Transitionen von den spektralen Eigenschaften des einen Lautes hin zu denen des anderen Lautes enthält und man somit an einer Stelle schneidet, die sehr schnelle Veränderungen enthält.

Sehr viel besser eignen sich Diphthonge, Kombinationen aus jeweils zwei Phonemen, wobei die zweite Hälfte vom ersten Phonem und die erste Hälfte vom zweiten Phonem verwendet wird, man die Diphthong also in der Mitte schneidet, wo sie eine relativ stationäre Phase haben. Dadurch erreicht man zum Einen, dass die Verkettungsstellen weniger starke Sprünge aufweisen, zum Anderen enthalten Diphthonge die wichtige Transition zwischen zwei Phonemen und decken damit zumindest ein gewisses Maß an Koartikulation ab (dazu später mehr).

Der Nachteil dabei ist, dass man nun ein sehr viel größeres Inventar an Einheiten braucht, da jedes Phonem mit jedem anderen kombiniert werden muss. Für n Phoneme ergeben sich also $n * (n - 1)$ Diphthonge.

2.4. Phonotaktische Beschränkungen

Die soeben gemachte Aussage über die Größe des Diphthong-Inventars relativiert sich jedoch, wenn man bedenkt, dass es phonotaktische Beschränkungen gibt, d.h. nicht jede Kombination von Phonemen auch tatsächlich erlaubt ist. Diese Beschränkungen sind natürlich sprachspezifisch. So kann im Deutschen z.B. $[h]$ nur silbeninitial vor einem Vokal vorkommen, d.h. die Folge $[h] + \text{Konsonant}$ ist ausgeschlossen. In der Literatur werden meist nur die phonotaktischen Beschränkungen innerhalb der Silbe betrachtet; für die Definition der Diphthongliste hilft das jedoch wenig, da natürlich an Silben- und Wortgrenzen viele Kombinationen auftauchen, die innerhalb einer Silbe nicht legal wären. Um das Beispiel aus 2.2 wieder aufzugreifen: innerhalb einer Silbe folgt auf einen Hinterzungenvokal ein $[x]$, an einer Silbengrenze kann aber durchaus auch ein $[C]$ folgen (Kuhchen – $[k u: C @ n]$, Beispiel aus [WALTHER 2001]).

Dennoch gibt es, neben der oben genannten, einige Beschränkungen für Diphthong-Kombinationen. Neben $[h]$ kann auch $[j]$ nur vor Vokalen vorkommen; für das Englische gilt dasselbe für $[w]$. Vokal-Vokal-Diphthonge könnten teilweise durch Vokal/Glottal Stop- und Glottal Stop/Vokal-Diphthonge ersetzt werden, allerdings tritt zwischen Langvokalen oder Diphtongen und Schwa kein Glottal Stop auf, ebenso nicht bei Fremdwörtern ([PORTELE 1996]), es ist also besser, sie durch Einheiten zu repräsentieren.

2.5. Minimale Koartikulation

Es gibt eine ganze Reihe von Konsonant-Konsonant-Kombinationen, bei denen die gegenseitige artikulatorische Beeinflussung relativ gering ist. So trennt z.B. die Verschlussphase eines Plosivs weitgehend von Einflüssen des vorhergehenden Lautes. Eine Reihe von Beispielen mit graphischer Darstellung findet sich in [MÖBIUS 2001],

2. Theorie

S. 166-7. Eine weitere Reduktion der Diphonliste kann also erreicht werden, indem solche Diphone, bei denen die beiden Phone nur minimal miteinander interagieren, weggelassen werden. Diese Diphone können dann bei Bedarf aus Phonemen zusammengesetzt werden, die an den Segmentgrenzen geschnitten wurden. Tabelle 2.2 enthält eine Liste von Konsonant-Kombinationen, die minimale Koartikulation zeigen.

Konsonant 1	Konsonant 2
Plosiv	Plosiv
Plosiv	Nasal
Frikativ	Frikativ
Frikativ	Plosiv
Frikativ	Nasal
Nasal	Plosiv
Nasal	Frikativ
Lateral	Plosiv
Lateral	Frikativ

Tabelle 2.2.: Diphone aus Konsonanten mit minimaler Koartikulation
(Quelle: [MÖBIUS 2001])

2.6. Koartikulatorische Effekte

„Concatenative synthesis assumes that all perceptually significant coarticulatory phenomena can be captured by using units that span regions of heavy coarticulation.“ ([OLIVE et al. 1998], S. 194) Die Frage ist nur, wie groß diese Einheiten sein müssen.

Aufgrund der Trägheit der Artikulatoren gibt es mehr oder minder starke Interaktionen zwischen den einzelnen Lauten einer Sequenz; dies war ja auch der Grund, warum man für die Synthese Diphone (und nicht einzelne Phone) als grundlegende Einheit nimmt (s.o.). Allerdings haben diese Effekte Auswirkungen weit über ihre direkt angrenzenden Nachbarn hinaus. Nun könnte man natürlich Triphone oder noch größere Einheiten verwenden, dies würde aber zu einer explosiven Vergrößerung des Einheiten-Inventars führen und ist somit nicht praktikabel.

Ein Ausweg aus diesem Problem ist die mehrfache Aufnahme bestimmter Diphon-Einheiten mit unterschiedlichen Kontexten.

Ein Beispiel für ein Phonem, das abhängig von seinem Kontext starken Schwankungen in der akustischen Realisierung unterliegt, ist das /r/, das im Deutschen sehr unterschiedlich realisiert wird: vokalisiert zu Schwa zwischen Vokal und Konsonant/Morphemgrenze, als velarer Approximant zwischen zwei Vokalen und als velarer Frikativ nach stimmlosem Obstruenten ([MÖBIUS 2001]).

So kann das Diphon [e: R] aus „Pferd“ schlecht verwendet werden, um „Amerika“ zu synthetisieren.

2. Theorie

Ein weiteres Beispiel ist /v/, das zwischen Vokalen stimmhaft auftritt (z.B. in „Aktivität“), nach einem stimmlosen Obstruenten jedoch stimmlos (z.B. in „Quelle“).

Auch bei Vokalen treten koartikulatorische Effekte auf: vor allem bei kurzen, unbetonten Vokalen kommt es häufig zu einem sogenannten *target undershoot* ([LINDBLÖM 1963], zitiert nach [MÖBIUS 2001]), d.h. aufgrund der Trägheit der Artikulatoren erreichen diese nicht ihre Sollposition, bevor der nächste Laut geformt wird. Akustisch wirkt sich das ebenfalls dahingehend aus, dass die eigentlich zu erwartenden Formantwerte nicht erreicht werden.

Bei einem Wort wie „Skrupel“ ist die Lippenrundung schon beim [s] beobachtbar, da sie für die Lautfolge [s k R] keine Rolle spielt, die Artikulatoren also schon früh in Stellung gebracht werden können, um das [u] dann auszusprechen (Beispiel aus [PORTELE 1996]). Dies zeigt, dass Koartikulation weit über die direkt angrenzenden Phone hinaus reichen kann.

Die Berücksichtigung koartikulatorischer Effekte führt natürlich wieder zu einer Vergrößerung des Diphon-Inventars.

2.7. Regelbasierte vs. lexikonbasierte Liste

Es gibt nun zwei Möglichkeiten, eine Diphonliste zu erstellen: entweder versucht man, phonotaktische Regeln zu finden, die die kombinatorisch mögliche Anzahl an Diphonen auf diejenigen beschränken, die tatsächlich in der jeweiligen Sprache vorkommen, oder man erstellt sie empirisch mit Hilfe eines großen Aussprachelexikons (unter der Annahme, dass das Lexikon nahezu alle tatsächlich möglichen Diphone abdeckt).

Allerdings ist es nicht einfach, Regeln aufzustellen, die weder zu restriktiv sind noch zu viele unnötige Diphone zulassen. Und auch das Aussprachelexikon kann noch so groß sein – es wird meist nicht alle möglichen Diphone abdecken.

2.8. Fremdsprachliche Laute

Bei dieser Studienarbeit ging es insbesondere auch darum, gängige fremdsprachliche Laute zu berücksichtigen, die schlecht durch deutsche Laute nachgebildet werden können, wie z.B. englisch [T, r, ...] oder die französischen Nasale. Als Grundlage für die Auswahl wurde die Technische Dokumentation des Smartkom-Projektes ([SMARTKOM 2000]) verwendet, die dann um englische und französische Laute, soweit nicht schon vorhanden, erweitert wurde. In Tabelle 2.1 sind diese Laute entsprechend markiert, allerdings wurden nicht alle in das endgültige Phoneset übernommen, sondern (soweit sinnvoll) auf ähnliche deutsche Laute gemappt (mehr dazu in Abschnitt 3.1).

2.9. Trägerwort-Generierung

Für die Aufnahme müssen die Diphone in geeignete Trägerwörter eingebettet werden. Dabei hat es sich als sinnvoll erwiesen (s. [MÖBIUS 2001]), Kunstwörter zu verwenden.

2. Theorie

Lautkombination	Trägerwort
Vokal-Konsonant	adVCa'dei
Schwa-Konsonant	ad@-Ca'dei
Konsonant-Vokal	aCVda'nei
Konsonant-Schwa	aC@-da'nei
Konsonant-Konsonant	aCCa'dei
Vokal-Vokal	adVV'dei
Vokal-aSchwa	adV6'dei
Vokal-Glottaler Stop	adV-ada'nei
Glottaler Stop-Vokal	ada-Vda'nei
Stille-Konsonant	Cada'nei
Konsonant-Stille	anadaC
Stille-Vokal	Vda'nei
Vokal-Stille	anadV

Tabelle 2.3.: Schema für die Trägerwort-Generierung

den, da diese einerseits automatisch generierbar sind und man andererseits die volle Kontrolle über die Position des Diphons innerhalb des Wortes und dessen Silbenbetonung hat. Um eine möglichst neutrale Aufnahme zu erreichen, sollte das Diphon weder Teil der hauptbetonten noch einer ganz unbetonten Silbe (Gefahr von *target undershoot*) sein.

Für die Generierung der Trägerwörter habe ich das in Tabelle 2.3 zu findende Schema verwendet, das mir Antje Schweitzer zur Verfügung gestellt hat.

3. Praktische Durchführung

In den folgenden Abschnitten möchte ich darstellen, wie ich konkret vorgegangen bin, um das Phoneset zu erstellen, welche empirischen Untersuchungen zu Diphonen ich gemacht habe und wie ich dann zu einer endgültigen Liste von Diphonen gekommen bin.

Zunächst überlegte ich, wie ich aus dem einmal definierten Phon-Inventar die Diphon-Liste generieren sollte. Prinzipiell gab es zwei Ansatzmöglichkeiten:

- *kombinatorisch*: aus dem Phoneset werden durch Regeln alle möglichen Diphone generiert
- *empirisch*: aus einem (möglichst umfangreichen) Aussprachelexikon wird eine Liste aller vorkommenden Diphone extrahiert

Beide Vorgehensweisen haben Vor- und Nachteile. Da aus dieser Studienarbeit ein Tool hervorgehen soll, das die Erzeugung der Diphonliste aus einem definierten Phon-Inventar weitgehend automatisiert, wäre ein kompakter Satz an Regeln sehr viel praktischer als ein über 50MB großes Aussprachelexikon. Allerdings ist es schwierig, diese Regeln so festzulegen, dass weder über- noch untergeneriert wird.

Ich bin deswegen zweigleisig vorgegangen: zum Einen habe ich untersucht, welche Diphone in einem großen deutschen und einem (weit kleineren) englischen Aussprachelexikon vorkommen, die beide in elektronischer Form vorlagen (mehr dazu unter [3.3](#)). Zum Anderen habe ich versucht, ein Regelwerk zu definieren, das die produzierten Diphone weitgehend auf die phonotaktisch legalen begrenzt, so dass die Diphonliste dann tatsächlich ohne Verwendung weiterer Ressourcen aus dem Phoneset erzeugt werden kann.

3.1. Das Phoneset

Zunächst einmal galt es, eine Menge von Phonemen zu definieren, die als Grundlage für die Diphonliste dienen soll. Neben allen deutschen Phonemen sollten auch die wichtigsten englischen und französischen Phoneme abgedeckt sein, damit später eine halbwegs natürliche Aussprache von Fremdwörtern aus diesen beiden Sprachen unter Verwendung der produzierten Diphone möglich sein würde. Für die Zusammenstellung der Phoneme orientierte ich mich an der Dokumentation des Smartkom-Projektes ([[SMARTKOM 2000](#)]); allerdings wurden eine Reihe französischer und englischer Phoneme aufgenommen, die bei Smartkom auf deutsche Phoneme gemappt werden: /ẽ, ã, õ, T, D, w, O:, eI, oU, L, r/; alle diese Laute können schlecht durch deutsche Laute

3. Praktische Durchführung

ersetzt werden. Andererseits wurden mit einer Ausnahme (/a/) alle kurzen, gespannten Vokale weggelassen (/i, u, y, e, o, ɔ/), d.h. auf die entsprechenden langen Vokale gemappt (/i:, u:, y:, e:, o:, ɔ:/), da sie sich nur durch die Länge unterscheiden, nicht jedoch durch spektrale Eigenschaften.¹ Die Länge kann dann durch Signalverarbeitung nach Bedarf verändert werden; sie variiert auch beim menschlichen Sprecher, abhängig vom Betonung-Status der Silbe usw., beträchtlich.

In Tabelle 3.1 ist das endgültige Phon-Inventar zu finden; diejenigen Phone, die nicht ins Inventar aufgenommen wurden, sind mit Mapping angegeben.²

3.2. Regelbasierte Generierung der Diphonliste

Aus dem Phon-Inventar (Tabelle 3.1) soll nun mit Hilfe eines Perl-Skripts (s. Anhang B.1) die Diphonliste erzeugt werden. Dabei werden zunächst alle Phone mit jedem anderen kombiniert und dann durch Regeln überprüft, ob es sich um eine ‘legale’ Kombination handelt.

3.2.1. Zu viele Diphone...

Wie bereits in 2.3 geschildert, werden unnötig viele Diphone erzeugt, wenn man rein kombinatorisch vorgeht – bei den 54 in Tabelle 3.1 gelisteten Phonen wären es 2862 Diphone, von denen jedoch um die 600 eingespart werden können.

Nach [OLIVE et al. 1998] sind für das Deutsche rund 1100 Diphone notwendig, plus 75 spezielle Einheiten zur Abdeckung von Kontexteffekten, allerdings bei einem Inventar von nur 43 Phonen.

Das Skript, das ich geschrieben habe, liest eine Datei mit dem Phon-Inventar ein und erzeugt dann alle Kombinationen. Dabei wird schon automatisch ein eventuelles Mapping berücksichtigt, d.h. für Phone, die auf ein anderes Phon abgebildet werden können, wird dieses, wie in der Datei definiert, verwendet. Das genaue Format der Definition des Phon-Inventars ist dem Anhang zu entnehmen (Abschnitt A.1); jede Zeile besteht aus Phon, Mapping und Feature-Vektor, wobei die drei Einträge durch Kommata getrennt werden. Hier zwei Beispiele:

```
T, , Konsonant Frikativ englisch
i, i:, Vokal
```

In der ersten Zeile ist ein Laut zu sehen, der nicht gemappt wird (deshalb ein leerer Eintrag fürs Mapping) und mehrere Features hat, in der zweiten Zeile ein Laut mit Mapping und einem Feature.

¹im Gegensatz zu den kurzen, laxen Vokalen /I, U, Y, E, O/, die sich auch spektral deutlich unterscheiden

²d.h. mit demjenigen Phon, durch das sie ersetzt werden, um das Inventar nicht zu groß werden zu lassen (die Anzahl der Diphone wächst ja quadratisch mit der Anzahl der Phone!)

3. Praktische Durchführung

Laut	Mapping	Laut	Mapping	Laut	Mapping
Vokale		Schwa		Plosive	
I		@		p	
i	i:	6		b	
i:				t	
U				d	
u	u:	Diphthonge		k	
u:		aI		g	
Y		aU		?	
y	y:	OY			
y:		eI		Frikative	
E		OI	OY	f	
e	e:	@U	o:	v	
e:		I@	i: 6	s	
E:		e@	E: 6	z	
O		U@	u: 6	S	
o	o:	oU		Z	
o:				C	
9				x	
2	2:	Nasale		h	
2:		n		T	
a		m		D	
a:		N		w	
{	E	9~	e~	Liquide	
Q	O	e~		l	
V	a	a~		L	
3:	2: 6	o~		R	
A:	a:			r	
O:					
3'	2: 6	Semivokal		Stille	
e'	6	j		*	

Tabelle 3.1.: Das Phon-Inventar mit Mapping (basierend auf: [SMARTKOM 2000])

3. Praktische Durchführung

Dann habe ich iterativ Regeln hinzugefügt, die, basierend auf den Feature-Vektoren der Phone, ungültige bzw. unnötige Kombinationen unterdrücken. Im nächsten Abschnitt ist dies näher beschrieben. Auf Diphone, die wegen koartikulatorischer Effekte in mehreren Kontext-Varianten vorliegen sollten, wird später noch eingegangen.

3.2.2. Reduktion durch Phonotaktik und minimale Koartikulation

Neben den bereits in Abschnitt 2.4 beschriebenen phonotaktischen Beschränkungen des Deutschen habe ich durch Vergleich mit der empirischen Diphonliste (s. 3.3) versucht, phonotaktische Regeln zu definieren, die die Ausgabe von allen nicht erwünschten Diphonen verhindern.

Die Regeln werden als einfache Nennung derjenigen Kombination von Features bzw. Lauten, die ausgeschlossen werden soll, formuliert, wobei einfache reguläre Ausdrücke möglich sind:

`Frikativ, Frikativ|Plosiv|Nasal`

würde alle Diphone ausschliessen, die eine Kombination aus Frikativ und Frikativ, Plosiv oder Nasal darstellen, während

`Konsonant, "N"`

die Kombination von Konsonanten mit $[N]$ verhindert. Die Features, wie sie im Phonetset definiert sind, erlauben also die Zusammenfassung von Lauten zu Gruppen und das intuitive Schreiben von Regeln. Diese Regeln werden vom Skript aus der Datei `ausschlussregeln.txt` (s. Anhang A.2) eingelesen. Zu beachten ist, dass Phone in Anführungszeichen eingeschlossen werden müssen, damit sie von den Features unterschieden werden können, und dass die beiden Teile durch ein Komma getrennt werden.

Die Aufstellung der Regeln ist gar nicht so einfach, da man sehr vorsichtig sein muss, nicht zu strikte Regeln zu definieren. Im Zweifelsfall wurden auch Diphone erlaubt, die eher unwahrscheinlich sind, aber nicht völlig ausgeschlossen werden können. So kann z.B nicht davon ausgegangen werden, dass $[C]$ nur nach Vorderzungen- und $[x]$ nur nach Hinterzungenvokalen auftaucht (s. dazu auch Abschnitt 2.4).

Im Einzelnen handelt es sich um die folgenden Regeln:

1. wegen minimaler Koartikulation weglassen
 - Plosiv-Plosiv, Plosiv-Nasal
 - Frikativ-Frikativ, Frikativ-Plosiv, Frikativ-Nasal
 - Nasal-Frikativ, Nasal-Plosiv
 - Lateral-Frikativ, Lateral-Plosiv

3. Praktische Durchführung

2. Kombinationen weglassen, die aus zwei sprachspezifischen Lauten bestehen³

- englisch-französisch, französisch-englisch
- deutsch-englisch, englisch-deutsch

3. echte phonotaktische Beschränkungen

- ‘h’-Konsonant, ‘h’-Stille
- ‘j’-Konsonant, ‘j’-Stille
- ‘D’-Konsonant, ‘D’-Stille⁴
- ‘w’-Konsonant, ‘w’-Stille
- Konsonant-‘N’, Stille-‘N’
- ‘?’-Konsonant, ‘?’-Stille

Die Regeln unter 1. schließen Diphone aus, die, wie in 2.5 beschrieben, wegen geringer Interaktion ohne Beeinträchtigung der Sprachqualität aus einzelnen Phonen zusammengesetzt werden können. Unter 2. sind einfache Regeln gelistet, die verhindern, dass die (vermutlich unnötigen) Diphone gebildet werden, die aus zwei sprachspezifischen Phonen bestehen. Die echten phonotaktischen Restriktionen finden sich unter 3.: [*h, j, D, w, ?*] können nur vor Vokalen auftreten, werden also vor Konsonanten und wortfinal ausgeschlossen; [*N*] hingegen kann nur nach Vokalen auftreten, wird also nach Konsonanten und wortinitial ausgeschlossen. Diese Regeln sind natürlich sprachspezifisch; für andere Sprachen gelten andere Restriktionen.

3.2.3. Berücksichtigung koartikulatorischer Effekte

Wie in Abschnitt 2.6 bereits geschildert, gibt es eine ganze Reihe von koartikulatorischen Effekten, die sich durch Diphoneinheiten nicht abdecken lassen. Nun gibt es verschiedene Möglichkeiten, diesem Problem zu begegnen. Man könnte es ignorieren und eine deutlich schlechtere Sprachqualität in Kauf nehmen (das kann aber nicht das Ziel unserer Bemühungen sein) oder man könnte zusätzliche, größere Einheiten verwenden, z.B. Triphone für Konsonant-Vokal-Konsonant-Übergänge. Diesen Weg sind tatsächlich einige Entwickler gegangen.

Eine andere und elegantere Möglichkeit ist die Verwendung spezieller, kontextsensitiver Diphoneinheiten, wie sie z.B. Olive ([OLIVE 1990]) vorgeschlagen hat. Im Folgenden soll dies verwendet werden, um einige der wichtigsten koartikulatorischen Phänomene abzudecken.

Bei den Liquiden sollte zwischen einerseits prä- oder intervokalischem und andererseits postvokalischem Auftreten unterschieden werden ([PORTELE 1996], S. 88).

³Kombinationen wie [*T y:*] oder [*r a ~*]; die hier verwendeten Regeln sind sehr rudimentär und auf die wenigen sprachspezifisch markierten Phone in der Liste abgestimmt; eigentlich wäre ein aufwändigerer Mechanismus notwendig

⁴zwar kommt im Englischen [*D*] hin und wieder vor wortfinalen [*s, t*] vor, in allen diesen Fällen wurde es aber bei der Modellierung der Auslautverhärtung auf [*T*] gemappt

3. Praktische Durchführung

Während die Liquide bei prä- und intervokalischem Vorkommen deutlich artikuliert sind, ist ein postvokalisches /l/ meist nur als Transition am Ende des Vokals erkennbar, ein /R/ ist in diesem Fall fast immer vokalisiert (als /6/ realisiert). Für alle Vokal-Liquid-Kombinationen sollten also zwei Varianten aufgenommen werden: solche mit einem Konsonanten als rechtem Kontext und solche mit einem Vokal. Während man für die beiden Liquide /l/ und /R/ jeweils $52 \times 52 = 2704$ Triphone bräuchte (und entsprechend viele für die Varianten /r/ und /L/), braucht man nur jeweils $2 \times 21 = 42$ kontextsensitive Diphoneinheiten.⁵

Nach [MÖBIUS 2001], S. 172, lässt sich dies auf alle Sonoranten und stimmhaften Frikative (s. Beispiel in 2.6) verallgemeinern – ihre Stimmhaftigkeit hängt von den benachbarten Lauten ab, es ist also sinnvoll, kontextsensitive Einheiten zu verwenden.⁶

Ähnlich verhält es sich bei den kurzen Vokalen, die häufig von *target undershoot* betroffen sind; hier ist es allerdings der Artikulationsort der angrenzenden Konsonanten, der sich stark auf die Formant-Transitionen (im Besonderen auf F_2) auswirkt. Es reicht also aus, jeweils einen typischen Vertreter der drei Artikulationsorte velar, dental und labial als linken oder rechten Kontext aufzunehmen. Dies bewirkt wiederum eine deutliche Reduktion der benötigten Einheiten: von $27 \times 27 = 729$ Triphonen pro Vokal⁷ auf nur noch $3 \times 27 + 3 \times 27 = 162$ Diphone.

Alle diese genannten Phänomene habe ich durch relativ simple Regeln in der Datei `kontextregeln.txt` (s. Anhang A.3) beschrieben, aus denen dann durch ein Perl-Skript (s. Anhang B.2) die zusätzlichen, kontextsensitiven Diphone erzeugt werden. Dabei habe ich auf eine automatische Generierung von Trägerwörtern verzichtet, da es sehr kompliziert wäre, sie allgemeingültig mit dem jeweils korrekten Kontext zu erzeugen. Es kann aber, für ein konkretes Beispiel, ausgehend vom in Tabelle 2.3 dargestellten Schema, ohne Probleme jeweils ein passendes Trägerwort zum kontextsensitiven Diphon von Hand erstellt werden und dann per regulärem Ausdruck auf die Liste angewandt werden.

Die Regeln zur Erzeugung von kontextsensitiven Diphonen sehen folgendermaßen aus:

```
Vokal, "l", [a]
[a], Sonorant|voicedFrikativ, Vokal
```

Dabei wird der Kontext in eckige Klammern gesetzt, Laute werden in Anführungszeichen eingeschlossen. Es können Features oder Disjunktionen von Features verwendet werden, um Gruppen von Lauten zu beschreiben. Der Kontext muss entweder ganz links oder ganz rechts stehen; jede Zeile darf genau drei mit Kommata getrennte Einträge enthalten. Kommentare sind erlaubt (mit '#'). Die Regeln im Anhang veranschaulichen, wie verschiedene koartikulatorische Effekte berücksichtigt werden können.

⁵bei 21 Vokalen; im Deutschen können Liquide in der Coda nur direkt nach dem Vokal vorkommen

⁶für eine ausführliche Diskussion s. [MÖBIUS 2004]

⁷bei 27 Konsonanten

3.3. Empirische Untersuchung zu Diphonvorkommen

Zum Vergleich, aber auch um zu überprüfen, inwieweit die oben aufgestellten phonotaktischen Regeln tatsächlich mit der Realität übereinstimmen, habe ich zwei umfangreiche Aussprachelexika untersucht: das deutschsprachige *gcelex-onomastica* mit ca. 800.000 Einträgen sowie das englischsprachige *cmudict* mit ca. 106.000 Einträgen.

3.3.1. Deutsches Aussprachelexikon: *gcelex-onomastica*

Das *gcelex-onomastica* ist ein sehr umfangreiches Lexikon in SAMPA-Notation, das auch viele Eigennamen enthält (es handelt sich dabei um eine Kombination von *onomastica* [Onomastica 1995] und deutschem *celex* [BAAYEN et al. 1995]). Ich habe ein Perl-Skript (s. Anhang B.3) geschrieben, das ein Aussprachelexikon im Format von *gcelex-onomastica* einliest und eine Liste sowohl aller vorkommenden Phone als auch Diphone ausgibt. Zudem werden einige statistische Daten wie Häufigkeit und Gesamtanzahl ausgegeben. Ein weiteres Skript erstellte eine Liste aller am Wortanfang bzw. -ende vorkommenden Phone.

Ausschlussliste

Im Laufe meiner Untersuchung stellte ich wiederholt fest, dass Einträge im Lexikon fehlerhaft waren (s. Tabelle 3.2). Zum Einen waren Einträge falsch transkribiert, zum Anderen wurden Phoneme benutzt, die nicht in unserem Phon-Inventar enthalten waren. Um nicht die Originaldatei verändern zu müssen, erstellte ich eine Ausschlussliste, die diejenigen Zeilen enthält, die vom Skript ignoriert werden sollen:

Die ersten drei Zeilen enthalten das Zeichen ‘*’, die nächste ein ‘#’, dann folgen Zeilen mit ‘A’ und ‘E’, die beide nicht in SAMPA definiert sind, sowie mehrere Einträge mit Phonemen, die nicht in unserem Inventar enthalten sind (bzw. auf andere Phoneme gemappt wurden). Einige Einträge habe ich ausgeschlossen, weil sie nicht die im Deutschen obligatorische Auslautverhärtung berücksichtigen (dazu mehr unter 3.3.2).

Phone in *gcelex*

Tabelle 3.3 zeigt die in *gcelex-onomastica* vorkommenden Phone nach ihrer Häufigkeit sortiert. Fünf der Phone in unserem Inventar kommen überhaupt nicht im Aussprachelexikon vor: [T, D, oU, L, r] – allesamt englische Phone. Andersherum deckt unser Inventar alle im Lexikon auftretenden Phone ab. Interessant ist auch, dass 43 Phone am Wortanfang vorkommen, aber nur 36 am Wortende. Mit großem Abstand die häufigsten wortfinalen Phone sind [@, t, n 6, s] (in dieser Reihenfolge). Bei den wortinitialen Phonen ist die Verteilung sehr viel gleichmäßiger.

Bei all diesen Daten darf nicht vergessen werden, dass, auch wenn es sich um ein recht großes Lexikon handelt, die Werte nicht unbedingt repräsentativ sind. Ein normaler Text besteht ja zu einem großen Teil aus einer recht kleinen Menge an

3. Praktische Durchführung

```
("Aprikosensteig" ZNEonoSTRE1 ((a p) 0) ((R i:) 0) ((k o:) 1) ((z @ n) 0) ((S t aI k *) 0))
("Aprikosenweg" ZNEonoSTRE1 ((a p) 0) ((R i:) 0) ((k o:) 1) ((z @ n) 0) ((v e: k *) 0))
("Horstfelde" ZNEonoTOWN1 ((h 0 6 s t) 0) ((f E l) 1) ((d @ *) 0))
("Jr." ZNEonoCOMP1 ((# j u: n) 0) ((j 0 6) 0))
("Ferdin." ZNEonoCOMP1 ((F E 6) 1) ((d i:) 0) ((n a n t) 0))
("Kalevala" N ((k A) 1) ((l E) 0) ((v A) 0) ((l A) 0))
("Friedr." ZNEonoCOMP1 ((f r i: d) 1) ((R I C) 0))
("Nachsaisons" N ((N a: x) 1) ((z E) 0) ((z o~ s) 0))
("Rouge" N ((R u: Z) 1))
("Friedj" ZNEonoSTRE1 ((f R i: d) 1))
("Handj" ZNEonoSTRE1 ((h a n d) 1))
("Reverend" N ((R E) 1) ((v @) 0) ((R @ n d) 0))
("Plumpudding" N ((p l V m) 1) ((p U) 1) ((d I N) 0))
("Plumpuddings" N ((p l V m) 1) ((p U) 1) ((d I N s) 0))
("Advantage" N ((@ d) 0) ((v A: n) 1) ((t I d Z) 0))
("Advantages" N ((@ d) 0) ((v A: n) 1) ((t I d Z s) 0))
("Trademark" N ((t R eI d) 1) ((m A: k) 0))
("Trademarks" N ((t R eI d) 1) ((m A: k s) 0))
("Parfum" N ((p a 6) 0) ((f 9~) 1))
("Parfums" N ((p a 6) 0) ((f 9~ s) 1))
```

Tabelle 3.2.: Die Ausschlussliste

Funktionswörtern und anderen häufigen Worten, die Laut-Häufigkeiten könnten dadurch deutlich anders verteilt sein. Ein großer Vorteil des Lexikons ist aber, dass es auch sehr viele seltene Wörter (wie z.B. Eigennamen) enthält und damit auch eher exotische Lautkombinationen.

Diphone in gcelex

Insgesamt fanden sich 1376 verschiedene Diphone; dabei wurden die Kombinationen Phon-Stille bzw. Stille-Phon nicht berücksichtigt. Wie Bild 3.1 zeigt, handelt es sich um eine typische LNRE-Verteilung (‘Large Number of Rare Events’): einige wenige Diphone sind sehr häufig, während sehr viele Diphone relativ bis sehr selten sind, insgesamt aber eine große Gruppe ausmachen. Wenn man die durch Kombination der Phone am Anfang bzw. Ende der Einträge gewonnenen Diphone noch hinzunimmt (da Wörter ja beliebig kombiniert werden können), erhält man 1782 verschiedene Diphone.

Die 10 häufigsten Diphone im Aussprachelexikon sind in Tabelle 3.4 zu finden.

3.3.2. Englischsprachiges Aussprachelexikon: *cmudict*

Das englische⁸ Aussprachelexikon *cmudict* lag in einer anderen Notation vor: Arpabet, einer Umschrift, die ebenfalls die IPA-Symbole auf ASCII-Zeichen abbildet. Deshalb musste ich für die weitere Untersuchung das Lexikon zunächst per Perl-Skript (s. Anhang B.4) in SAMPA umwandeln, anschliessend modellierte ich mit

⁸amerikanisches Englisch

3. Praktische Durchführung

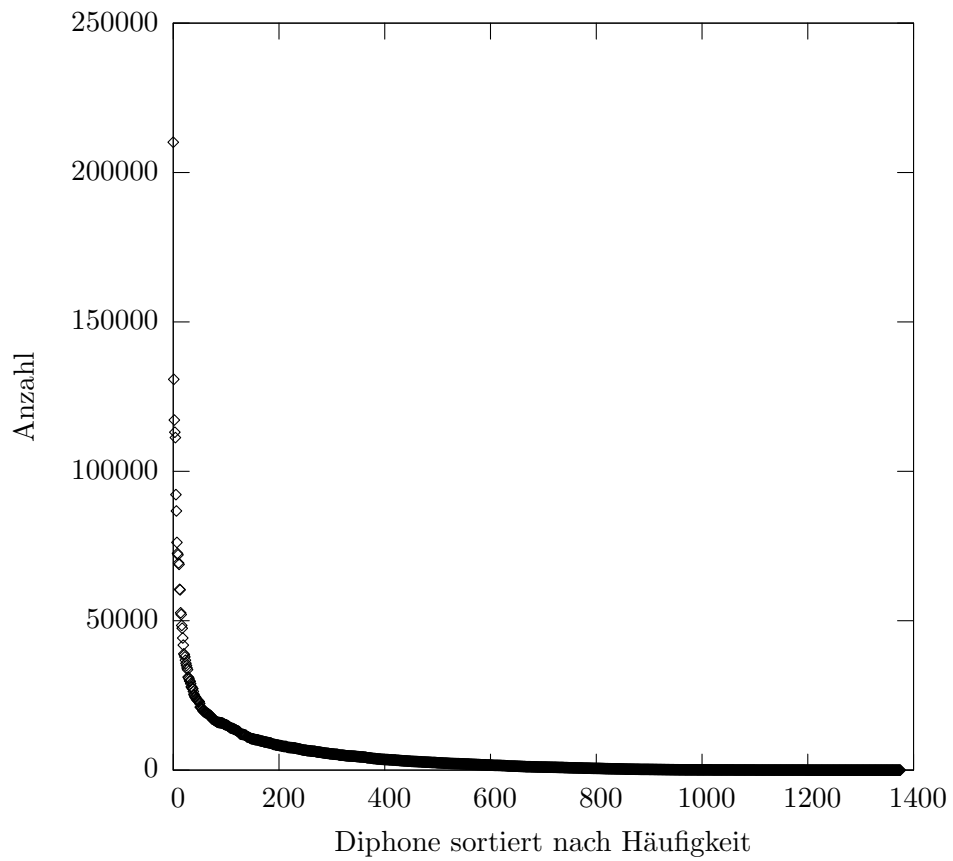


Abbildung 3.1.: Die Diphone in gcelex: eine typische LNRE-Verteilung

3. Praktische Durchführung

Phon	t	@	s	n	l	R	6	a
Anzahl	568770	488653	374504	367735	336821	336743	309837	233127
Phon	E	S	I	k	b	f	m	g
Anzahl	213793	209830	208146	205929	193657	178281	164768	162974
Phon	a:	d	v	i:	h	p	aI	z
Anzahl	146937	133375	123616	123298	117871	117512	103760	97535
Phon	o:	U	O	e:	C	aU	Y	y:
Anzahl	96160	94916	92191	89297	73332	58327	52180	51539
Phon	u:	N	2:	E:	9	x	OY	j
Anzahl	50462	49901	36728	33395	30224	26265	25968	15664
Phon	Z	a~	o~	e~	O:	eI	w	
Anzahl	1121	623	243	79	9	8	3	

Tabelle 3.3.: Phone in *gcelex-onomastica*, sortiert nach Häufigkeit

Diphon	@ n	t @	s t	S t	t s	s @	t R	g @	n t	R a:
Anzahl	210151	130794	117152	113090	111294	92191	86739	76193	72533	72057

Tabelle 3.4.: Die 10 häufigsten Diphone in *gcelex-onomastica*

einem weiteren Skript das in unserem Phon-Inventar verwendete Mapping sowie die deutsche Aussprache.

Als „Abfallprodukt“ entstand somit ein umfangreiches englisches Aussprachelexikon, das auf deutsches SAMPA gemappt ist und auch eine weitgehend deutsche Aussprache berücksichtigt. Es könnte z.B. dazu benutzt werden, einem deutschen Sprachsynthesystem die rudimentäre Synthese von englischen Wörtern zu ermöglichen, wenn auch mit vermutlich starkem deutschem Akzent.

Mapping nach SAMPA

Für das Mapping von Arpabet nach SAMPA wurde eine Zusammenstellung mehrerer verschiedener Notationen auf

<http://www.ling.ed.ac.uk/facilities/howto/ipa/ipatable.html>

verwendet. Eine Gegenüberstellung der beiden Notationen findet sich in Tabelle 3.5.

Deutsche Lautmodellierung

Als nächstes wurde durch ein Skript (s. Anhang B.5) das Mapping wie in Tabelle 3.1 durchgeführt, sowie die Affrikaten in zwei Phone zerlegt, Auslautverhärtung realisiert und ‘er’-Schwa am Wortende als [ɐ] transkribiert. Das Phänomen der Auslautverhärtung tritt sowohl am Wort- wie auch am Morphemende auf, wurde hier allerdings nur am Wortende modelliert, da die im Lexikon vorhandene Silbeninformation für diese Zwecke nicht brauchbar ist (Stichproben ergaben sehr viele falsche Syllabifizierungen).

3. Praktische Durchführung

Arpabet	SAMPA	Arpabet	SAMPA	Arpabet	SAMPA
ng	N	ae	{	ao	O:
th	T	uh	U	ey	eI
dh	D	oh	Q	ea	e@
sh	S	ah	V	ay	aI
zh	Z	ax	@	aw	aU
hh	h	ua	U@	ua	U@
ch	tS	iy	i:	ow	@U
jh	dZ	aa	A:	oy	OI
ih	I	uw	u:	ia	I@
eh	E	er	3:	y	j

Alle nicht aufgeführten Symbole sind in beiden Notationen identisch.

Tabelle 3.5.: Mapping von Arpabet nach SAMPA

Somit hatte ich ein englisches Aussprachelexikon vorliegen, das ich mit den selben Skripten verarbeiten konnte, die ich auch für das deutsche Lexikon verwendet habe. Diese Version von *cmudict* wird im Folgenden als *cmudict-SAMPA-German* bezeichnet.

Phone in cmudict

Zum Vergleich mit Tabelle 3.3 sind in Tabelle 3.6 die Phone mit ihrer Häufigkeit in *cmudict-SAMPA-German* angegeben. Es fällt auf, dass es nur 38 Phone sind im Vergleich zu den 47 in *gcelex-onomastica*, allerdings ist dabei zu bedenken, dass in letzterem sehr viele Eigennamen und Fremdwörter enthalten sind. In *cmudict-SAMPA-German* wird durchgehend [r] statt [R] verwendet. Alle im Lexikon vorkommenden Phone werden von unserem Inventar abgedeckt.

Phon	@	I	E	t	r	n	l	s
Anzahl	44370	38003	37991	35225	34556	34411	33170	30254
Phon	k	d	m	a:	b	i:	p	2:
Anzahl	29365	21428	21077	19498	16244	15398	14590	14510
Phon	6	o:	f	g	S	eI	O:	aI
Anzahl	14425	10486	9948	9801	9613	9288	8513	8191
Phon	v	h	u:	w	a	z	Z	j
Anzahl	8005	7202	6754	6521	6242	5213	4415	3661
Phon	N	aU	U	T	OY	D		
Anzahl	3219	2494	1787	1724	843	413		

Tabelle 3.6.: Phone in *cmudict-SAMPA-German*, sortiert nach Häufigkeit

3. Praktische Durchführung

Diphone in *cmudict*

In *cmudict-SAMPA-German* fanden sich 1154 verschiedene Diphone, relativ gesehen mehr als im deutschen Aussprachelexikon (ca. 82% vs. ca. 65% der rechnerisch möglichen Diphone). Dies liegt wohl vor allem daran, dass im deutschen Lexikon ein größerer Anteil von selten verwendeten Phonen (franz. Nasale, engl. Laute) vorhanden ist. Durch Hinzunahme der durch Wortkombinationen möglichen Diphone erhöht sich die Zahl auf 1336 verschiedene Diphone.

Wie im deutschen Lexikon zeigt sich eine LNRE-Verteilung. Die 10 häufigsten Diphone sind in Tabelle 3.7 gelistet.

Diphon	@ n	2: 6	@ l	s t	E n	n t	I N	I n	m @	a: r
Anzahl	15844	14510	8804	8021	7689	6480	6069	5908	5339	5117

Tabelle 3.7.: Die 10 häufigsten Diphone in *cmudict-SAMPA-German*

3.4. Vergleich der regelbasierten und empirischen Diphonliste

Wie unter Abschnitt 3.2 beschrieben, wurden bei der Erstellung der regelbasierten Diphonliste zwei verschiedene Gesichtspunkte berücksichtigt – zum Einen wurden phonotaktische Regeln berücksichtigt, die unmögliche Diphone verhindern, zum Anderen wurden Regeln angewandt, die auf rein akustischen Beobachtungen basierend die Anzahl der für die Synthese notwendigerweise aufzunehmenden Diphone deutlich reduzieren. Letzteres ist für die empirische Diphonliste natürlich nicht gegeben.

Zählt man die Diphone, die generiert werden, wenn nur die phonotaktischen Beschränkungen angewandt werden, kommt man auf 2569 Diphone. Im Vergleich zu den rechnerisch möglichen 2862 Diphonen (bei 54 Phonen) ergibt sich also nur eine Reduktion um rund 300 Diphone. Fasst man alle in den beiden Lexika vorkommenden Diphone zusammen, ergibt sich eine Liste von 2204 Diphonen.

Es zeigt sich also, dass noch eine Lücke von 365 Diphonen bleibt, die entweder nicht von den Lexika abgedeckt werden oder aber übergeneriert werden. Vermutlich ist ersteres der Fall, da z.B. viele mögliche Kombinationen mit den französischen Nasalen nicht im Lexikon auftauchen. Ausserdem ist zu bedenken, dass es aufgrund der LNRE-Verteilung der Diphone sehr wahrscheinlich ist, dass es eine recht ansehnliche Menge von Diphonen gibt, die sehr selten sind, also mit einiger Wahrscheinlichkeit nicht im Lexikon vorkommen.

4. Diskussion der Ergebnisse

4.1. Systematische Lücken

Beim Aufstellen der phonotaktischen Regeln für die Generierung habe ich mir die Diphone angeschaut, die in der empirischen Diphonliste fehlen, um systematische Lücken zu finden. Meist waren diese Diphone aber nicht auszuschliessen. Es konnten also keine systematischen Lücken gefunden werden, die nicht schon durch die der Literatur entnommenen Regeln abgedeckt waren.

4.2. Diphon-Generierung durch Regeln?

Berücksichtigt man phonotaktische Einschränkungen und Kombinationen minimaler Koartikulation, so lässt sich die Liste der Diphone auf rund 2250 Einträge reduzieren, was (bei 54 Phonen im Inventar) immerhin einer Reduktion um ca. 20% entspricht.

Angesichts des Aufwandes, der für Aufnahme und Nachverarbeitung notwendig ist, sollte man sich also auf jeden Fall Gedanken über die Konstruktion des Einheiteninventars machen.

Es reicht nicht aus, ein großes Aussprachelexikon als Ausgangspunkt zu verwenden, da dann viele seltene Diphone fehlen werden (LNRE-Verteilung!).

Damit bleibt nur die Generierung der benötigten Diphone aus einer Phon-Liste, was sich zumindest teilweise, wie in dieser Studienarbeit aufgezeigt, automatisieren lässt.

Man wird aber auf jeden Fall die erzeugte Diphonliste sorgfältig überprüfen und bei Bedarf korrigieren bzw. erweitern müssen.

4.3. Berücksichtigung koartikulatorischer Effekte

Um eine qualitativ hochwertige Diphon-Synthese zu erreichen, kommt man nicht umhin, für bestimmte koartikulatorische Effekte kontextsensitive Einheiten zu verwenden. Mit den in 3.2.3 aufgestellten Regeln werden 2004 Diphone erzeugt.¹ Davon entfallen allein auf die speziellen Einheiten für die neun kurzen Vokale 1458 Diphone, 462 werden benötigt, um die Entstimmung von Sonoranten und stimmhaften Frikativen zu berücksichtigen. Dagegen sind für die verschiedenen Varianten von /R/ und /l/ nur 84 Einheiten notwendig.² Allerdings darf man dabei nicht vergessen, dass

¹das stimmt so nicht ganz: für die koartikulatorischen Effekte wurden die Diphthonge ignoriert, die im Phoneset aber auch als Vokal markiert sind

²bzw. 168, wenn man /r/ und /L/ entsprechend behandelt

4. Diskussion der Ergebnisse

diese speziellen Diphone zum Teil die bereits vorhandenen generischen Diphone ersetzen, es sind also nicht wirklich alles „zusätzliche“ Einheiten. Ausserdem wird ein Teil von ihnen durch die phonotaktischen Regeln wieder ausgeschlossen, was ich hier jetzt nicht berücksichtigt habe.

Die Aufnahme aller hier vorgestellten potentiellen kontextsensitiven Diphone würde beinahe zu einer Verdoppelung der Inventargröße führen, ist also nicht unbedingt praktikabel. Am ehesten können wohl die Einheiten für die kurzen Vokale weggelassen werden – zum Einen sind es sehr viele, zum Anderen sind die Auswirkungen nicht so gravierend wie bei den anderen koartikulatorischen Effekten.

4.4. Das endgültige Diphon-Inventar

Die endgültige Liste der Diphone umfasst 2248 Diphone, von denen 673 in verschiedenen Kontexten aufgenommen werden müssen, um die geschilderten koartikulatorischen Effekte angemessen zu erfassen. Damit würden dann 2050 kontextsensitive Diphone dazu kommen, was insgesamt ein Inventar von 3625 Einheiten ergäbe.

Dies erscheint sehr viel, lässt sich aber dramatisch reduzieren, wenn man nur einen Teil der kontextsensitiven Einheiten berücksichtigt – die Einheiten für *target undershoot* bei kurzen Vokalen könnten z.B. weggelassen werden. Dann wären nur noch rund 300 zusätzliche, kontextsensitive Diphone notwendig. Dies erscheint mir ein sinnvoller Kompromiss. Damit ergibt sich eine Inventargröße von ca. 2550 Diphonen.

Aus Platzgründen habe ich mich entschieden, nicht die gesamte Liste von Diphonen im Anhang (A.4) abzudrucken, sondern nur einen Querschnitt von zufällig ausgewählten Einträgen (inkl. der Trägerwörter), der einen guten Eindruck gibt, wie die Liste aussieht. Entsprechend bin ich mit der Liste der kontextsensitiven Diphone (A.5) verfahren: sie zeigt beispielhaft für ein paar ausgewählte Diphone jeweils die beiden verschiedenen Kontexte, in denen sie aufgenommen werden müssen.

5. Zusammenfassung

Nachdem ich zunächst sehr viel Zeit und Aufwand in die Untersuchung der Aussprachelexika gesteckt hatte (was mir aber eine gute Einführung in das Thema gab), war die Implementierung der Diphonlistengenerierung basierend auf einem festgelegten Phoneset und phonotaktischen Regeln eigentlich relativ einfach. Selbst die Generierung der passenden Trägerwörter mit Hilfe des Schemas war kein großes Problem.

Die Idee dabei war, alles so anzulegen, dass es einfach an andere Bedingungen angepasst werden kann, z.B. ein größeres oder kleineres Phonem-Inventar, andere phonotaktische Regeln usw. Entsprechend sind Phonem-Inventar und phonotaktische Regeln nicht im Quellcode enthalten, sondern werden aus einfach strukturierten Text-Dateien eingelesen.

Deutlich schwieriger war es, die koartikulatorischen Phänomene adäquat zu behandeln. Ich habe mich auf die drei prominentesten Beispiele aus der einschlägigen Literatur beschränkt und anhand von ihnen aufgezeigt, wie sich dieses Problem durch automatisch generierte kontextsensitive Diphon-Einheiten lösen lässt und wie diese in das Diphon-Inventar integriert werden können. Auch hier werden die Einheiten aus Kontextregeln erzeugt, die in einer Text-Datei festgelegt werden.

Es hat sich gezeigt, dass in die sorgfältige Konstruktion eines Diphon-Inventars sehr viel Arbeit fließt, die sich auch nicht völlig automatisieren lässt. Ich hoffe aber, dass die im Rahmen dieser Studienarbeit entwickelten Skripten die Erstellung deutlich beschleunigen und vereinfachen können.

Im Vergleich zu den Inventargrößen anderer Systeme (z.B. German Bell Labs TTS-System mit 1177 Einheiten, [MÖBIUS 2001]) scheinen ca. 2500 Einheiten für unser Inventar recht viel, allerdings enthält es auch deutlich mehr Phoneme als die meisten anderen. Es ist aber eine Größe, mit der man noch hantieren kann.

A. Formatdefinitionen

A.1. Das Phon-Inventar

```
# Phoneset in SAMPA-Notation, mit Mapping in der zweiten Spalte,  
# Features in der dritten Spalte (CSV-Format)  
# im Moment 54 (inkl. Pause und Glottalem Stop) nicht gemappte, 19 gemappte Phone  
  
? , , Konsonant # Glottaler Stop  
* , , Pause  
  
# Vokale  
I , , Vokal kurzerVokal  
i , i: , Vokal  
i: , , Vokal  
U , , Vokal kurzerVokal  
u , u: , Vokal  
u: , , Vokal  
Y , , Vokal kurzerVokal deutsch # verhindert Kombination mit englischen Lauten  
y , y: , Vokal  
y: , , Vokal deutsch  
E , , Vokal kurzerVokal  
e , e: , Vokal  
e: , , Vokal  
E: , , Vokal  
O , , Vokal kurzerVokal  
o , o: , Vokal  
o: , , Vokal  
9 , , Vokal kurzerVokal  
2 , 2: , Vokal  
2: , , Vokal  
a , , Vokal kurzerVokal  
a: , , Vokal  
{ , E , Vokal englisch  
Q , O , Vokal englisch  
V , a , Vokal englisch  
3: , 2:6 , Vokal englisch  
A: , a: , Vokal englisch  
O: , , Vokal englisch  
3' , 2:6 , Vokal englisch  
e' , 6 , Vokal englisch  
  
# Nasalvokale
```

A. Formatdefinitionen

```
9~ , e~ , Vokal franzoesisch
e~ , , Vokal franzoesisch
a~ , , Vokal franzoesisch
o~ , , Vokal franzoesisch

# Schwa
@ , , Vokal kurzerVokal
6 , , Vokal kurzerVokal

# Diphtonge
aI , , Vokal
aU , , Vokal
OY , , Vokal
eI , , Vokal englisch
OI , OY , Vokal englisch
@U , o: , Vokal englisch
I@ , i:6 , Vokal englisch
e@ , E:6 , Vokal englisch
U@ , u:6 , Vokal englisch
oU , , Vokal englisch

# Semivokal
j , , Konsonant

# Plosive
p , , Konsonant Plosiv
b , , Konsonant Plosiv
t , , Konsonant Plosiv
d , , Konsonant Plosiv
k , , Konsonant Plosiv
g , , Konsonant Plosiv

# Frikative
f , , Konsonant Frikativ
v , , Konsonant Frikativ voicedFrikativ
s , , Konsonant Frikativ
z , , Konsonant Frikativ voicedFrikativ
S , , Konsonant Frikativ
Z , , Konsonant Frikativ voicedFrikativ
C , , Konsonant Frikativ deutsch
x , , Konsonant Frikativ deutsch
h , , Konsonant Frikativ
T , , Konsonant Frikativ englisch
D , , Konsonant Frikativ englisch
w , , Konsonant Frikativ englisch voicedFrikativ
```

A. Formatdefinitionen

```
# Liquide
l , , Konsonant Sonorant Lateral deutsch
L , , Konsonant Sonorant Lateral englisch
R , , Konsonant Sonorant Lateral deutsch
r , , Konsonant Sonorant Lateral englisch

# Nasale
n , , Konsonant Sonorant Nasal
m , , Konsonant Sonorant Nasal
N , , Konsonant Sonorant Nasal
```

A.2. Die Ausschlussregeln

```
# Ausschlussregeln fuer die Diphon-Generierung
# (c) Manuel Weiss 2006
#
# es koennen regulaere Ausdruecke fuer die Features verwendet werden
#
# wenn beide Ausdruecke auf die Featureliste ihres Phons
# (s. Phonetset-Definition) matchen, wird das entsprechende
# Diphon unterdrueckt.
# Alternativ kann, in Anfuehrungszeichen eingeschlossen,
# ein Phon direkt angegeben werden

# koennen wegen minimaler Koartikulation eingespart wegen:
Plosiv, Plosiv|Nasal
Frikativ, Frikativ|Plosiv|Nasal
Nasal, Frikativ|Plosiv
Lateral, Frikativ|Plosiv

# sprachspezifische Laute brauchen vermutlich nicht
# mit einander gemischt werden:
englisch, franzoesisch
franzoesisch, englisch
englisch, deutsch
deutsch, englisch

# phonotaktische Beschraenkungen
"h", Konsonant|Pause
"j", Konsonant|Pause
"D", Konsonant|Pause
"w", Konsonant|Pause
Konsonant|Pause, "N"

"?", Konsonant|Pause # Glottaler Stop nur vor Vokal
```

A.3. Die Kontextregeln

```
# Kontextregeln
# [] markiert Kontext
# Literale muessen in "" eingeschlossen werden
# Gruppen können durch Feature angegeben werden (aus Phonetset.txt)

# Einheiten fuer Vokal-Liquid-Diphone mit unterschiedlichem rechtem Kontext
Vokal, "l", [a]
Vokal, "l", [t]
Vokal, "R", [a]
Vokal, "R", [t]

# Einheiten fuer unterschiedliche Stimmhaftigkeit
# von Sonoranten/Frikativen je nach linkem Kontext
[k], Sonorant|voicedFrikativ, Vokal
[a], Sonorant|voicedFrikativ, Vokal

# Einheiten fuer target undershoot bei kurzen Vokalen
#Konsonant, kurzerVokal, [p]
#Konsonant, kurzerVokal, [t]
#Konsonant, kurzerVokal, [k]
#[p], kurzerVokal, Konsonant
#[t], kurzerVokal, Konsonant
#[k], kurzerVokal, Konsonant
```

A.4. Ausschnitt aus der Diphonliste

```
Diphon  Trägerwort (unbetonte, nebenbetonte Silbe)
a~ f    ada~fa'dei, da~fa'da
N e~    aNe~da'nei
N e:    aNe:da'nei
N U     aNUda'nei
d S     adSa'dei
E z     adEza'dei, dEza'da
E oU    adEoU'dei, adEoUd
y: i:   ady:i:'dei, ady:i:d
y: x    ady:xa'dei, dy:xa'da
k E:    akE:da'nei
k l     akla'dei
k eI    akeIda'nei
e: r    ade:ra'dei, de:ra'da
t E:    atE:da'nei
i: k    adi:ka'dei, di:ka'da
i: o:   adi:o:'dei, adi:o:d
v 2:    av2:da'nei
s aI    asaIda'nei
aI b    adaIba'dei, daIba'da
I d     adIda'dei, dIda'da
```

A. Formatdefinitionen

I t adIta'dei, dIta'da
I w adIwa'dei, dIwa'da
z a~ aza~da'nei
2: o~ ad2:o~'dei, ad2:o~d
w 9 aw9da'nei
x E axEda'nei
* aI aIda'nei
* l lada'nei
C aU aCaUda'nei
@ t ad@-ta'dei, ad@ta'da
aU 9 adaU9'dei, adaU9d
l aU alaUda'nei
eI a adeIa'dei, adeIad
p T apTa'dei
p eI apeIda'nei
R E: aRE:da'nei

A.5. Ausschnitt aus der Liste der kontextsensitiven Diphone

Diphone [Kontext]
a l [a]
a l [t]
a R [a]
a R [t]
E l [a]
E l [t]
E R [a]
E R [t]
[k] Z e:
[a] Z e:
[k] v o:
[a] v o:
[k] m y:
[a] m y:

B. Die Perl-Skripten

B.1. Generierung der Diphonliste

../GeneriereDiphonListeAusPhonset.pl:

```
#!/usr/bin/perl -w
use strict;

# Manus Diphon-Studienarbeit-Perl-Skript-Kollektion
# Dieses Skript liest eine Phoneme-Liste im Manuel-Standard-Format
# (Phoneme, Mapping, Features) ein und generiert eine Liste von
# Diphonen mit Traegerwort

my ($Phonem, $Mapping, $Features);
my $gemappt = 0;
my $gelesen = 0;
my (%Phonemliste, %Featureliste);
my $datei = "ausschlussregeln.txt";
my (@regeln1, @regeln2);

# diese Funktion generiert die Traegerwoerter
# wird mit zwei Phonem als Argument aufgerufen
sub traegerwort($$)
{
    my ($i, $j) = @_;
    if ($i eq "@" and $Featureliste{$j} =~ /Konsonant/) # Schwa - Konsonant
    {
        return "ad\@-$j)a\'dei, ad\@${j})a\'da";
    }
    if ($Featureliste{$i} =~ /Vokal/ and $j eq "?") # Vokal - Glottaler Stop
    {
        return "ad$i-ada\'nei, ad$i-adana";
    }
    if ($Featureliste{$i} =~ /Vokal/ and $Featureliste{$j} =~ /Konsonant/) #
        Vokal - Konsonant
    {
        return "ad${i}${j})a\'dei, d${i}${j})a\'da";
    }
    if ($Featureliste{$i} =~ /Konsonant/ and $j eq "@") # Konsonant - Schwa
    {
        return "a${i})\@-da\'nei";
    }
    if ($i eq "?" and $Featureliste{$j} =~ /Vokal/) # Glottaler Stop - Vokal
    {
        return "ada-${j})da\'nei, ada-${j})dana";
    }
    if ($Featureliste{$i} =~ /Konsonant/ and $Featureliste{$j} =~ /Vokal/) #
        Konsonant - Vokal
    {
        return "a${i}${j})da\'nei";
    }
}
```

B. Die Perl-Skripten

```
if ($Featureliste{$i} =~ /Konsonant/ and $Featureliste{$j} =~ /Konsonant/)
    # Konsonant - Konsonant
{
    return "a${i}${j}a\`dei";
}
if ($Featureliste{$i} =~ /Vokal/ and $j eq "6") # Vokal - a-Schwa
{
    return "ad${i}6\`dei, ad${i}erd";
}
if ($Featureliste{$i} =~ /Vokal/ and $Featureliste{$j} =~ /Vokal/) # Vokal
    - Vokal
{
    return "ad${i}${j}\`dei, ad${i}${j}d";
}
if ($Featureliste{$i} =~ /Pause/ and $Featureliste{$j} =~ /Konsonant/) #
    Pause - Konsonant
{
    return "${j}ada\`nei";
}
if ($Featureliste{$i} =~ /Pause/ and $Featureliste{$j} =~ /Vokal/) # Pause
    - Vokal
{
    return "${j}da\`nei";
}
if ($Featureliste{$i} =~ /Konsonant/ and $Featureliste{$j} =~ /Pause/) #
    Konsonant - Konsonant
{
    return "anada${i}";
}
if ($Featureliste{$i} =~ /Vokal/ and $Featureliste{$j} =~ /Pause/) #
    Konsonant - Konsonant
{
    return "anad${i}";
}

return "Schema fehlt";
}

# Regel-Datei oeffnen und einlesen:
open FH, $datei or die "Fehler beim Oeffnen von $datei\n";

while(<FH>)
{
    chomp;
    s/#.*$//; # Kommentare loeschen
    s/\s+$//; # Spaces am Zeilenende loeschen
    next if (/^\s*$/); # Leerzeilen ueberspringen
    my ($i, $j) = split /,/;
    $i =~ tr/ //d; # Leerzeichen loeschen
    $j =~ tr/ //d;
    push @regeln1, $i;
    push @regeln2, $j;
}

die "Fehler in Regeln\n" if ($#regeln1 != $#regeln2);
close FH;

while(<>)
{
    chomp;
    s/#.*$//; # Kommentare loeschen
    s/\s+$//; # Spaces am Zeilenende loeschen
```

B. Die Perl-Skripten

```
next if (/^\s*$/); # Leerzeilen ueberspringen
if (not /\S\S?\s*,\s*.{1,4}\s*,\s*(.*)$/ )
{
    print STDERR "Fehlerhafte Zeile: $_\nEintrag ignoriert.\n";
    next;
}

$Phonem = ($1 ? $1 : ""); # enthaelt 1. Spalte aus Phoneme-Liste (= Phonem
)
$Mapping = ($2 ? $2 : ""); # optional, enthaelt 2. Spalte aus Phoneme-
Liste (= Phonem, auf das es gemappt wird)
$Features = ($3 ? $3 : ""); # optional, enthaelt 3. Spalte aus Phoneme-
Liste (= Features)

# Space am Anfang und Ende entfernen
$Phonem =~ s/^\s+//;
$Phonem =~ s/\s+$//;
$Mapping =~ s/^\s+//;
$Mapping =~ s/\s+$//;
$Features =~ s/^\s+//;
$Features =~ s/\s+$//;

print STDERR "$Phonem\t$Mapping\t$Features\n";
$gelesen++;

if (not $Mapping)
{
    $Phonemliste{$Phonem} = 1;
    $Featureliste{$Phonem} = $Features;
}
else
{
    $gemappt++;
    print STDERR "gemappt: $Phonem nach $Mapping\n";
}
}

print STDERR "$gelesen Phoneme eingelesen, davon $gemappt gemappt.\nIm
Phonemset: ", scalar(keys %Phonemliste), "\n";

foreach my $Phonem1 (keys %Phonemliste)
{
    phonem: foreach my $Phonem2 (keys %Phonemliste)
    {
        if ($Phonem1 ne $Phonem2)
        {
            for (my $i = 0; $i <= $#regeln1; $i++)
            {
                # Fall 1: zwei Phone
                if($regeln1[$i] =~ "/" and $regeln2[$i] =~ "/")
                {
                    if ("\"$Phonem1\" eq $regeln1[$i] and "\"$Phonem2\" eq $regeln2[$i
                    ])
                    {
                        next phonem;
                    }
                }
                # Fall 2: Phon und Feature
                elsif($regeln1[$i] =~ "/")
                {
                    if ("\"$Phonem1\" eq $regeln1[$i] and $Featureliste{$Phonem2} =~ /
                    $regeln2[$i]/ )
```

B. Die Perl-Skripten

```
        {
            next phonem;
        }
    }
    # Fall 3: Feature und Phon
    elsif($regeln2[$i] =~ /"/)
    {
        if ($Featureliste{$Phonem1} =~ /$regeln1[$i]/ and "\"$Phonem2\" eq
            $regeln2[$i] )
        {
            next phonem;
        }
    }
    # Fall 4: zwei Features
    elsif ($Featureliste{$Phonem1} =~ /$regeln1[$i]/ and $Featureliste{
        $Phonem2} =~ /$regeln2[$i]/ )
    {
        next phonem;
    }
}
my $Traegerwort = traegerwort($Phonem1, $Phonem2);
print "$Phonem1 $Phonem2\t$Traegerwort\n";
}
}
```

B.2. Generierung von kontextsensitiven Diphonen

../GeneriereSpezielleDiphone.pl:

```
#!/usr/bin/perl -w
use strict;

# Manus Diphon-Studienarbeit-Perl-Skript-Kollektion
# Dieses Skript liest eine Phonem-Liste im Manuel-Standard-Format
# (Phonem, Mapping, Features) sowie Kontextregeln ein und
# generiert eine Liste von kontextabhaengigen Diphonen

my ($Phonem, $Mapping, $Features);
my $gemappt = 0;
my $gelesen = 0;
my (%Phonemliste, %Featureliste);
my $datei = "ausschlussregeln.txt";
my $datei2 = "kontextregeln.txt";
my (@ausschl_regeln1, @ausschl_regeln2);
my (@kontext_regeln1, @kontext_regeln2, @kontext_regeln3);

# Regel-Datei oeffnen und einlesen:
open FH, $datei or die "Fehler beim Oeffnen von $datei\n";

while(<FH>)
{
    chomp;
    s/#.*$//; # Kommentare loeschen
    s/\s+$//; # Spaces am Zeilenende loeschen
    next if (/^\s*$/); # Leerzeilen ueberspringen
    my ($i, $j) = split /,/;
    $i =~ tr//d; # Leerzeichen loeschen
    $j =~ tr//d;
}
```

B. Die Perl-Skripten

```
    push @ausschl_regeln1, $i;
    push @ausschl_regeln2, $j;
}

die "Fehler in Ausschluss-Regeln\n" if ($#ausschl_regeln1 != $#ausschl_regeln2
    );
close FH;

open FH, $datei2 or die "Fehler beim Oeffnen von $datei2\n";

while(<FH>)
{
    chomp;
    s/#!/$/; # Kommentare loeschen
    s/\s+$/; # Spaces am Zeilenende loeschen
    next if (/^\s*$/); # Leerzeilen ueberspringen
    my ($i, $j, $k) = split /,/;
    $i =~ tr/ //d; # Leerzeichen loeschen
    $j =~ tr/ //d;
    $k =~ tr/ //d;
    push @kontext_regeln1, $i;
    push @kontext_regeln2, $j;
    push @kontext_regeln3, $k;
}

die "Fehler in Kontext-Regeln\n" if ($#kontext_regeln1 != $#kontext_regeln2 or
    $#kontext_regeln1 != $#kontext_regeln3);
close FH;

while(<>)
{
    chomp;
    s/#!/$/; # Kommentare loeschen
    s/\s+$/; # Spaces am Zeilenende loeschen
    next if (/^\s*$/); # Leerzeilen ueberspringen
    if (not /^(\S\S?)\s*,\s*{1,4}\s*,\s*(.*)$/ )
    {
        print STDERR "Fehlerhafte Zeile: $_\nEintrag ignoriert.\n";
        next;
    }

    $Phonem = ($1 ? $1 : ""); # enthaelt 1. Spalte aus Phoneme-Liste (= Phonem
    )
    $Mapping = ($2 ? $2 : ""); # optional, enthaelt 2. Spalte aus Phoneme-
    Liste (= Phonem, auf das es gemappt wird)
    $Features = ($3 ? $3 : ""); # optional, enthaelt 3. Spalte aus Phoneme-
    Liste (= Features)

    # Space am Anfang und Ende entfernen
    $Phonem =~ s/^\s+//;
    $Phonem =~ s/\s+$/;
    $Mapping =~ s/^\s+//;
    $Mapping =~ s/\s+$/;
    $Features =~ s/^\s+//;
    $Features =~ s/\s+$/;

    print STDERR "$Phonem\t$Mapping\t$Features\n";
    $gelesen++;

    if (not $Mapping)
    {
        $Phonemliste{$Phonem} = 1;
    }
}
```

B. Die Perl-Skripten

```
$Featureliste{$Phonem} = $Features;
}
else
{
$gemappt++;
print STDERR "gemappt: $Phonem nach $Mapping\n";
}
}

print STDERR "$gelesen Phoneme eingelesen, davon $gemappt gemappt.\nIm
Phonemset: ", scalar(keys %Phonemliste), "\n";

foreach my $Phonem1 (keys %Phonemliste)
{
phonem: foreach my $Phonem2 (keys %Phonemliste)
{
if ($Phonem1 ne $Phonem2)
{
for (my $i = 0; $i <= $#ausschl_regeln1; $i++)
{
# Fall 1: zwei Phone
if ($ausschl_regeln1[$i] =~ "/" and $ausschl_regeln2[$i] =~ "/")
{
if ("\"$Phonem1\"" eq $ausschl_regeln1[$i] and "\"$Phonem2\"" eq
$ausschl_regeln2[$i])
{
next phonem;
}
}
# Fall 2: Phon und Feature
elsif ($ausschl_regeln1[$i] =~ "/")
{
if ("\"$Phonem1\"" eq $ausschl_regeln1[$i] and $Featureliste{
$Phonem2} =~ /$ausschl_regeln2[$i]/ )
{
next phonem;
}
}
# Fall 3: Feature und Phon
elsif ($ausschl_regeln2[$i] =~ "/")
{
if ($Featureliste{$Phonem1} =~ /$ausschl_regeln1[$i]/ and "\"
$Phonem2\"" eq $ausschl_regeln2[$i] )
{
next phonem;
}
}
# Fall 4: zwei Features
elsif ($Featureliste{$Phonem1} =~ /$ausschl_regeln1[$i]/ and
$Featureliste{$Phonem2} =~ /$ausschl_regeln2[$i]/ )
{
next phonem;
}
}
}
for (my $i = 0; $i <= $#kontext_regeln1; $i++)
{
# linker Kontext
if ($kontext_regeln1[$i] =~ /\[.+\\]/)
{
if ("\"$Phonem1\"" eq $kontext_regeln2[$i] and $Featureliste{
$Phonem2} =~ /$kontext_regeln3[$i]/
or $Featureliste{$Phonem1} =~ /$kontext_regeln2[$i]/ and "\"
```

B. Die Perl-Skripten

```
        $Phonem2\"" eq $kontext_regeln3[$i]
    or $Featureliste{$Phonem1} =~ /$kontext_regeln2[$i]/ and
        $Featureliste{$Phonem2} =~ /$kontext_regeln3[$i]/)
    {
        print "$kontext_regeln1[$i] $Phonem1 $Phonem2\n";
    }
}
# rechter Kontext
elsif ($kontext_regeln3[$i] =~ /\[.+\/])
{
    if ("\"$Phonem1\" eq $kontext_regeln1[$i] and $Featureliste{
        $Phonem2} =~ /$kontext_regeln2[$i]/
        or $Featureliste{$Phonem1} =~ /$kontext_regeln1[$i]/ and "\"
        $Phonem2\" eq $kontext_regeln2[$i]
        or $Featureliste{$Phonem1} =~ /$kontext_regeln1[$i]/ and
        $Featureliste{$Phonem2} =~ /$kontext_regeln2[$i]/)
    {
        print "$Phonem1 $Phonem2 $kontext_regeln3[$i]\n";
    }
}
else
{
    print "Fehlender Kontext!\n";
}
}
}
}
```

B.3. Diphone im Aussprachelexikon

../DiphoneSuchen.pl:

```
#!/usr/bin/perl -w
use strict;

# Manus Diphon-Studienarbeit-Perl-Skript-Kollektion
# dieses Skript liest Zeilen im gcelex-Format von STDIN und
# gibt eine Liste der darin vorkommenden Diphone aus

my $falscherEintrag = 0;
my $tokens = 0;
my $skipped = 0;
my ($Wort, $POS, $Umschrift, $reineUmschrift, $eineSilbe);
my $datei = "Ausschlussliste.gcelex.txt";
my (@Silben, @Phoneme, %Phonemliste, %Diphonliste, @Ausschlussliste);

# Hier kann man ein- und ausschalten, ob ueber die Silbengrenzen hinweg
# gesucht werden soll:
my $nur_innerhalb_der_Silbe = 0;

# Ausschlussliste oeffnen und einlesen:
open FH, $datei or die "Fehler beim Oeffnen von $datei\n";

while(<FH>)
{
    chomp;
    push @Ausschlussliste, $_;
}
}
```

B. Die Perl-Skripten

```
close FH;

ausсен:
while(<>)
{
    chomp;

    # die Zeilen ueberspringen, die in der Ausschlussliste enthalten sind
    foreach my $i (@Ausschlussliste)
    {
        $skipped++ and next ausсен if ($i eq $_);
    }

    # dieser reg. Ausdruck zerlegt eine Zeile in ihre drei Spalten; \x[...] =
    # Unicode fuer ae, oe, ue, sz, e mit Akzent
    if (not /\^\("[A-Za-z.\-\x{00E4}\x{00C4}\x{00F6}\x{00D6}\x{00FC}\x{00DC}\x
        {00DF}\x{00E9}]+)" (\w+) (\(.+\))$/ )
    {
        $falscherEintrag++;
        print STDERR "nicht ge-parse-te Zeile: $_\n";
        next;
    }

    $Wort = $1;    # enthaelt 1. Spalte aus gcelex = Originalwort
    $POS = $2;    # enthaelt 2. Spalte aus gcelex = POS-Annotation
    $Umschrift = $3; # enthaelt 3. Spalte aus gcelex = phonetische Umschrift in
        SAMPA

    $reineUmschrift = $Umschrift;
    $reineUmschrift =~ s/\) \(/ - /g if ($nur_innerhalb_der_Silbe); #
        Silbengrenzen mit "-" markieren
    $reineUmschrift =~ tr/[(\)01]//d; # Klammern und Betonungsinformation
        (0/1) entfernen

    # Mapping fuer Phoneme, die nicht im Phoneteset sind:
    $reineUmschrift =~ s/9~/e~/g; # Wichtig: bei diesen Phonemen das
        Leerzeichen nicht vergessen!
    $reineUmschrift =~ s/3[:'] /2: 6 /g; # s.o.
    $reineUmschrift =~ s/2 /2: /g; # s.o.
    $reineUmschrift =~ s/y /y: /g; # s.o.
    $reineUmschrift =~ s/e /e: /g; # s.o.
    $reineUmschrift =~ s/o /o: /g; # s.o.
    $reineUmschrift =~ s/u /u: /g; # s.o.
    $reineUmschrift =~ s/i /i: /g; # s.o.
    $reineUmschrift =~ s/A /a: /g; # s.o.

    @Silben = split / - /, $reineUmschrift; # an den Silbengrenzen (s.o.)
        splitten

    foreach $eineSilbe (@Silben)
    {
        $eineSilbe =~ tr/ / /s; # mehrfache Leerzeichen durch eins ersetzen
            ... sicher ist sicher.
        @Phoneme = split / /, $eineSilbe; # an den Leerzeichen in Phoneme
            splitten
        for(my $i = 0; $i < (@Phoneme - 1); $i++)
        {
            $Phonemliste[$Phoneme[$i]]++; # Hier zaehlen wir die einzelnen Phoneme
            if ($Phoneme[$i] ne $Phoneme[$i + 1]) # Diphone aus zwei gleichen
                Phonemen ausschliessen (kommt an Silbengrenzen oft vor)
            {

```

B. Die Perl-Skripten

```
        my $Diphon = $Phoneme[$i] . " " . $Phoneme[$i + 1];
        $tokens++;
        $Diphonliste{$Diphon}++;      # Hier zaehlen wir die Diphone
    }
}

@Silben = ();      # Wichtig: Array wieder leeren

}

foreach (keys %Diphonliste)
{
    print $Diphonliste{$_}, "\t", $_, "\n";
}

print STDERR scalar(keys %Phonemliste) . " Phoneme gefunden:\n";
foreach (keys %Phonemliste)
{
    print STDERR $Phonemliste{$_}, "\t", $_, "\n";
}
print STDERR "insgesamt $tokens Diphone gefunden\n";
print STDERR scalar(keys %Diphonliste) . " verschiedene Diphone gefunden\n";
print STDERR "$falscherEintrag nicht verarbeitbare Eintraege im Input\n";
print STDERR "$skipped Zeilen ignoriert, da in $datei\n";
```

B.4. Mapping Arpabet nach SAMPA

../Arpabet2SAMPA.pl:

```
#!/usr/bin/perl -w

# ein simples Skript zur Konvertierung
# von Arpabet in SAMPA-Notation

use strict;
use locale;

my ($Wort, $POS, $Umschrift);

while(<>)
{
    chomp;

    if (not /^\[A-Za-z.\-]+\ (\w+) (\(.+)\)$/)
    {
        print STDERR "nicht geparsete Zeile: $_\n";
        next;
    }

    $Wort = $1;
    $POS = $2;
    $Umschrift = $3;

    $Umschrift =~ s/ng/N/g;
    $Umschrift =~ s/th/T/g;
    $Umschrift =~ s/dh/D/g;
    $Umschrift =~ s/sh/S/g;
    $Umschrift =~ s/zh/Z/g;
```

B. Die Perl-Skripten

```
$Umschrift =~ s/hh/h/g;
$Umschrift =~ s/ch/tS/g;
$Umschrift =~ s/jh/dZ/g;
$Umschrift =~ s/ih/I/g;
$Umschrift =~ s/eh/E/g;
$Umschrift =~ s/ae/ɨ/g;
$Umschrift =~ s/uh/U/g;
$Umschrift =~ s/oh/Q/g;
$Umschrift =~ s/ah/V/g;
$Umschrift =~ s/ax/\@/g;
$Umschrift =~ s/ua/U\@/g;
$Umschrift =~ s/iy/i:/g;
$Umschrift =~ s/aa/A:/g;
$Umschrift =~ s/uw/u:/g;
$Umschrift =~ s/er/ɜ:/g;
$Umschrift =~ s/ao/O:/g;
$Umschrift =~ s/ey/eI/g;
$Umschrift =~ s/ea/e\@/g;
$Umschrift =~ s/ay/aI/g;
$Umschrift =~ s/aw/aU/g;
$Umschrift =~ s/ua/U\@/g;
$Umschrift =~ s/ow/\@U/g;
$Umschrift =~ s/oy/OI/g;
$Umschrift =~ s/ia/I\@/g;
$Umschrift =~ s/y/j/g;

print "(\"$Wort\" $POS $Umschrift)\n";
}
```

B.5. Deutsche Lautmodellierung

../DeutscheLautmodellierung.pl:

```
#!/usr/bin/perl -w

# Dieses Skript modelliert die Aussprache englischer Woerter
# mit deutschem Sampa und versucht die engl. Phoneme soweit
# moeglich auf deutsche zu mappen, inkl. Auslautverhaertung
# (C) Manuel Weiss 2006

use strict;
use locale;

my ($Wort, $POS, $Umschrift, $test);

while(<>)
{
    chomp;

    # Zeile in ihre drei Spalten zerlegen:
    if (not /\^(("[A-Za-z.-]+)" (\w+) (\(.+\)\$) /)
    {
        print STDERR "nicht geparsete Zeile: $_\n";
        next;
    }

    $Wort = $1;
    $POS = $2;
    $Umschrift = $test = $3;
}
```

B. Die Perl-Skripten

```
# Affrikate zerlegen
$Umschrift =~ s/dZ/d Z/g;
$Umschrift =~ s/tS/t S/g;

# Auslautverhaertung:
if ($Umschrift =~ /([ktpsSTgdbzZD](?: [ktpsSTgdbzZD]){0,3}\) [01]\)\)\$/) #
    stimmhafte Konsonantengruppe am Wortende
{
    my $mapping = $1;
    # Konsonant-Gruppen aus bdgzZ muessen am Wort- und Morphemende nach ptksS
    # gemappt werden
    # hier werden nur solche am Wortende beruecksichtigt
    $mapping =~ tr/gdbzZD/ktpsST/;
    # Sonderzeichen escapen:
    my $quoted = quotemeta $1;
    $Umschrift =~ s/$quoted/$mapping/ if ($1 ne $mapping);
}

# "er" -> Schwa am Wortende
$Umschrift =~ s/3:\) 0\)\)\$/6) 0))/g;

# andere Mappings
$Umschrift =~ s/i /i: /g;
$Umschrift =~ s/u /u: /g;
$Umschrift =~ s/y /y: /g;
$Umschrift =~ s/e /e: /g;
$Umschrift =~ s/o /o: /g;
$Umschrift =~ s/2 /2: /g;
$Umschrift =~ s/3[':]/2: 6/g;
$Umschrift =~ s/e'/6/g;
$Umschrift =~ s/{/E/g;
$Umschrift =~ s/Q/O/g;
$Umschrift =~ s/I\@/i: 6/g;
$Umschrift =~ s/e\@/E: 6/g;
$Umschrift =~ s/U\@/u: 6/g;
$Umschrift =~ s/9~/e~/g;

$Umschrift =~ s/\@U/o:/g;
$Umschrift =~ s/OI/OY/g;
$Umschrift =~ s/A:/a:/g;
$Umschrift =~ s/V/a/g;
$Umschrift =~ s/\(s w/(s v/g; # Dieses Mapping nur wort-/silbeninitial??

print "(\"$Wort\" $POS $Umschrift)\n";
}
```

Literaturverzeichnis

- [ALLEN 1992] ALLEN, JONATHAN (1992). *Overview of text-to-speech systems*. In: FURUI, SADAOKI und M. M. SONDEHI, Hrsg.: *Advances in Speech Signal Processing*, S. 741–790. Marcel Dekker, New York. **2.1**
- [BAAYEN et al. 1995] BAAYEN, HARALD, R. PIEPENBROCK und L. GULIKERS (1995). *The CELEX lexical database – Release 2*. CD-ROM. Centre for Lexical Information, Max Planck Institute for Psycholinguistics, Nijmegen; Linguistic Data Consortium, University of Pennsylvania. **3.3.1**
- [CMU 1998] CMU (1998). *Carnegie Mellon Pronouncing Dictionary*. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>. frei verfügbares Aussprachelexikon für nordamerikanisches Englisch.
- [Edinburgh] EDINBURGH. *Mappingtabelle IPA-Arrabet-SAMPA u.a.* <http://www.ling.ed.ac.uk/facilities/howto/ipa/ipatable.html>. zusammengestellt vom Dept. of Linguistics, University of Edinburgh, letzter Zugriff 1. Mai 2006. **2.1**
- [KÜPFMÜLLER und WARNS 1956] KÜPFMÜLLER, KARL und O. WARNS (1956). *Sprachsynthese aus Lauten*. Nachrichtentechnische Fachberichte, 3:28–31. **1**
- [LINDBLOM 1963] LINDBLOM, BJÖRN (1963). *A spectrographic study of vowel reduction*. Journal of the Acoustical Society of America, 35:1773–1781. **2.6**
- [MÖBIUS 2001] MÖBIUS, BERND (2001). *German and Multilingual Speech Synthesis (Habilitationsschrift)*, Bd. 7 d. Reihe *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung, Uni Stuttgart*. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart. **2.1, 2.5, 2.2, 2.6, 2.9, 3.2.3, 5**
- [MÖBIUS 2004] MÖBIUS, BERND (2004). *Corpus-Based Investigations on the Phonetics of Consonant Voicing*. Folia Linguistica, 38(1-2):5–26. **6**
- [OLIVE 1990] OLIVE, JOSEPH (1990). *A new algorithm for a concatenative speech synthesis system using an augmented acoustic inventory of speech sounds*. In: *Proceedings of the ESCA Workshop on speech synthesis*, S. 25–30, Autrans. ESCA. **3.2.3**
- [OLIVE et al. 1998] OLIVE, JOSEPH, J. VAN SANTEN, B. MÖBIUS und C. SHIH (1998). *Synthesis*. In: SPROAT, RICHARD, Hrsg.: *Multilingual Text-to-Speech Synthesis – The Bell Labs Approach*, Kap. 7, S. 191–228. Kluwer Academic Publishers, Dordrecht. **2.6, 3.2.1**

Literaturverzeichnis

- [OLIVE et al. 1993] OLIVE, JOSEPH P., A. GREENWOOD und J. COLEMAN (1993). *Acoustics of American English Speech: A Dynamic Approach*. Springer-Verlag.
- [Onomastica 1995] ONOMASTICA (1995). *Multi-language pronunciation dictionary of proper names and place names*. Final Report, 30 May 1995, European Community, Linguistic Research and Engineering Programme, European Commission, DG XIII. Project No. LRE-61004. [3.3.1](#)
- [PORTELE 1996] PORTELE, THOMAS (1996). *Ein phonetisch-akustisch motiviertes Inventar zur Sprachsynthese deutscher Äußerungen*, Bd. 32 d. Reihe *Sprache und Information*. Max Niemeyer Verlag, Tübingen. [1](#), [2.4](#), [2.6](#), [3.2.3](#)
- [SMARTKOM 2000] SMARTKOM (2000). *Das Leitprojekt SmartKom: Dialogische Mensch-Technik-Interaktion durch koordinierte Analyse und Generierung multipler Modalitäten*. <http://smartkom.dfki.de/>. [2.8](#), [3.1](#), [3.1](#)
- [SPROAT 1998] SPROAT, RICHARD (1998). *Multilingual Text-to-Speech Synthesis – The Bell Labs Approach*. Kluwer Academic Publishers. [1](#)
- [WALTHER 2001] WALTHER, MARKUS (2001). *Phonologie*. In: CARSTENSEN, KAI-UWE, C. EBERT, C. ENDRISS, S. JEKAT, R. KLABUNDE und H. LANGER, Hrsg.: *Computerlinguistik und Sprachtechnologie: eine Einführung*, Kap. 3.1, S. 136 ff. Spektrum Akademischer Verlag, Heidelberg. [2.4](#)
- [WIESE 2000] WIESE, RICHARD (2000). *The Phonology of German*. The phonology of the world's languages. Oxford University Press.