

Auszug aus der Diplomarbeit

**Entwicklung eines automatischen
datenbasierten Tools zur Graphem-Phonem-
Konvertierung**

Veronika Boiko

Diplomarbeit Nr. 94

Prüfer:	Prof. Dr. Grzegorz Dogil
Betreuer:	Dr. Antje Schweitzer
Beginn der Arbeit	17. Mai 2010
Ende der Arbeit	17. November 2010

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung
Azenbergstraße 12
70174 Stuttgart

Erklärung

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.

Ort, Datum

Veronika Boiko

Danksagung

Meine Diplomarbeit ist dem Bereich Sprachsynthese gewidmet. Sie entstand am Institut für Maschinelle Sprachverarbeitung. Mein Dank gilt meiner Betreuerin Frau Dr. Antje Schweitzer und meinem Prüfer Herrn Prof. Dr. Grzegorz Dogil.

In meiner Diplomarbeit beschäftige ich mich mit der Entwicklung eines automatischen datenbasierten Tools zur Graphem-Phonem-Konvertierung für ein deutschsprachiges Text-to-Speech-System. Es wurde für die Firma SpeechConcept in Heidelberg entwickelt. Ich bedanke mich bei dem Geschäftsführer Herrn Michael Mende und meinen Betreuern Frau Anne Schönwandt und Herrn Marek Ivan.

Vorwort

Im Rahmen dieser Diplomarbeit wurde ein automatisches datenbasiertes Tool zur Graphem-Phonem-Konvertierung für ein deutschsprachiges Text-to-Speech-System entwickelt. Das Tool wurde als ein Offline-Verfahren konzipiert, das zur automatischen Transkription von großen Datenmengen offline dienen soll, z.B. zur Erstellung oder Aktualisierung eines Lexikons. Bei dem entwickelten Verfahren handelt es sich um ein hybrides Verfahren im weiteren Sinne, da verschiedene Ansätze und Lösungen in einem Verfahren interagieren.

Wegen meiner Verschwiegenheitsverpflichtung findet man hier nur einen Auszug aus der Diplomarbeit. Er enthält eine Zusammenfassung von Grundlagen zur Graphem-Konvertierung und gibt einen Überblick über den aktuellen Forschungsstand. Im Mittelpunkt der Auseinandersetzung mit diversen Methoden zur Graphem-Phonem-Konvertierung stehen folgende Fragen: auf welchen Prinzipien basieren die Methoden; was setzen sie voraus; durch welche Vor- und Nachteile zeichnen sie sich aus; welche Ergebnisse liefern sie.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Begriffserläuterung	1
1.2	Allgemeines zu Sprachsynthese-Systemen	1
1.3	Übersicht über das TTS-System CereVoice	3
1.4	Bedeutung der Graphem-Phonem-Konvertierung für Sprachsynthese und Spracherkennung	7
1.5	Gliederung der Arbeit	7
2	Grundlagen der Graphem-Phonem Konvertierung	9
2.1	Graphem-Phonem-Zuordnung	9
2.2	Einflussfaktoren der Graphem-Phonem-Abbildung	11
3	Theoretische Ansätze und Verfahren zur Graphem-Phonem-Konvertierung	13
3.1	Strategien zur Graphem-Phonem-Konvertierung	13
3.2	Regelbasierte Verfahren	15
3.3	Datenbasierte Verfahren	17
3.3.1	Pronunciation by Analogy	17
3.3.2	Instance-based learning	24
3.3.3	Neuronale Netze	30
3.3.4	Entscheidungsbäume	35
3.3.5	Statistische Verfahren: HMMs und joint n-gram-Modelle	41
3.4	Evaluierung von verschiedenen Verfahren zur Graphem-Phonem- Konvertierung	45
3.5	Fazit	50
	Anhang	52
A	IPA Symbole für das Deutsche	52
B	SAMPA für das Deutsche	53
	Literaturverzeichnis	55

1 Einleitung

1.1 Begriffserläuterung

Ein orthographischer String kann in Schriftzeichen, *Graphe*, zerlegt werden, die die kleinsten bedeutungsunterscheidenden Einheiten sind. Alle Graphe mit derselben bedeutungsunterscheidenden Funktion werden zu einer abstrakten Klasse zusammengefasst, die man als *Graphem* bezeichnet. Das Graphem ist das kleinste bedeutungsunterscheidende graphische Symbol, das ein oder mehrere Phoneme wiedergibt (*Duden* 1990). Im Deutschen sind graphische Symbole Buchstaben, deren Menge das Alphabet der deutschen Sprache bildet. Das Alphabet mit 30 Buchstaben dient zur schriftlichen Darstellung der gesprochenen Sprache. Grapheme und Graphemfolgen werden durch spitze Klammern gekennzeichnet, z.B. <s>, <synthese>.

Analog zu Graphen in der schriftlichen Sprache, gibt es *Phone* in der gesprochenen Sprache. Das Phon ist als die kleinste nicht mehr unterteilbare lautliche Einheit definiert. Phone, die in der gleichen lautlichen Umgebung vorkommen und mit dem phonologischen Unterschied auch den Bedeutungsunterschied markieren, gehören zu verschiedenen *Phonemen*. Phoneme sind die kleinsten abstrakten bedeutungsunterscheidenden Einheiten einer Sprache (*Rook* 1987). Phoneme werden mit Lautschriftsymbolen (IPA, SAMPA) zwischen zwei Schrägstrichen dargestellt, z.B. /s/, /z o n @/.

Im Mittelpunkt dieser Diplomarbeit steht die Graphem-Phonem-Konvertierung. Das Ziel der Graphem-Phonem-Konvertierung ist die Bestimmung der Aussprache eines Wortes. Die Aussprache ist mit phonetischer Lautschrift, *Transkription*, gegeben. Damit dient die Graphem-Zu-Phonem-Konversion zur Überführung von Graphemen (Buchstaben) in die Lautschrift (Phoneme).

1.2 Allgemeines zu Sprachsynthese-Systemen

Das Thema Graphem-Zu-Phonem-Konversion tritt in erster Linie im Kontext von Sprachsynthese und Text-to-Speech-Systemen (TTS) auf. Sprachsynthese ist als Erzeugung gesprochener Sprache durch Computer oder andere Maschinen definiert. Unter dem Begriff *Sprachsynthese* werden zwei Arten von Systemen vereint: Voice-Response-Systeme (IVR) und eigentliche Sprachsynthesizer.

Voice-Response-Systeme hantieren mit einem begrenzten Vokabular (ca. 100 Wörter). Typischerweise werden sie überall da eingesetzt, wo Sätze und Phrasen nach einem genauen vordefinierten Muster gebildet werden. Entscheidend für die Zuordnung ist ein recht kleiner Umfang von Informationen und ein fester Satz von Mustern, in denen diese Informationen übermittelt werden, z.B.

Verkehrsinformationen, Durchsagen über Ankunft und Abfahrt, Übermittlung von bestimmten Informationen über das Telefon bei Finanzinstituten, Öffnungszeiten bei öffentlichen Institutionen.

Eigentliche Sprachsynthesizer wandeln symbolische Darstellung von Information in ihre akustische Repräsentation um. Je nach ihrem Input werden Synthesizers/Sprachgeneratoren in Text-to-Speech-Systeme (TTS) und Concept-To-Speech (CTS) Systeme unterteilt. CTS erzeugen gesprochene Sprache aus linguistischen Repräsentationen, Konzepten, abstrakten Daten, z.B. Tabellen. In dieser Diplomarbeit sind TTS-Systeme von Interesse. Als Eingabe für ein TTS-System dient ein Text. Das kann ein einfacher Text sein oder ein Text, der zusätzlich mit den für seine Generierung wichtigen Informationen versehen ist, z.B. emphatische Betonung, Sprechstil, Pause, Hinweise auf ein Zahlenformat (Datum, Währung, Nummer). Die TTS-Systeme werden verwendet, wenn Texte mit einem beliebigen Wortschatz ausgegeben werden sollen. Typische Einsatzgebiete sind:

- **Telekommunikationsdienste**
TTS-Systeme ermöglichen den Zugang zu Informationen in Textform über das Telefon, z.B. in Auskunftssystemen.
- **Web und Radio**
Webseiten profitieren von der Vertonung mit TTS-Systemen. Einige Radiosender verwenden TTS-Systeme zum Vorlesen von Verkehrsnachrichten.
- **E-learning**
Hochwertige TTS-Synthese kann in E-learning-Systemen eingesetzt werden, z.B. in Online-Wörterbüchern.
- **Für Behinderte**
Stimmen-Handicaps können mit Hilfe einer speziellen Tastatur Texte zur Synthese eingeben. Blinde Menschen können auch an TTS-Systemen mit gekoppelten optischen Erkennern, die den Zugang zur geschriebenen Sprache ermöglichen, profitieren.
- **Multimedia, „Sprechende“ Hörbücher und Spiele**
- **Integrierte Systeme**
Alle Situationen, wo Audio-Informationen effizienter als Texte sind, wo man sich auf anderen visuellen Quellen konzentrieren muss, z.B. im Cockpit, beim Autofahren, sind potentielle Einsatzorte für TTS-Systeme. Schiffsnavigation, Baukräne, Mobiltelefone, Messinstrumente oder Überwachungssysteme, Warenhaus Management Systeme, Produktionslinien gehören zu den

1.3 Übersicht über das TTS-System CereVoice

Die Entwicklung des in dieser Diplomarbeit vorgestellten Verfahrens setzte eine intensive Auseinandersetzung mit dem Text-to-Speech-System CereVoice voraus. In diesem Kapitel wird die Aufbau und die Arbeitsweise eines TTS-Systems am Beispiel von CereVoice illustriert. CereVoice® ist ein Sprachsynthese-SDK, das von CereProc Ltd hergestellt wird. Das Unternehmen wurde 2005 gegründet mit dem Ziel, eine charaktervolle und effiziente Unit-Selection-Sprachsynthese anzubieten. CereProc arbeitet eng mit CSTR (Center of Speech Technology Research) der Universität Edinburgh zusammen. Das TTS-System CereVoice ist sowohl für kommerzielle als auch für Forschungszwecke verfügbar. Abbildung 1 gibt einen Überblick über das TTS-System CereVoice.

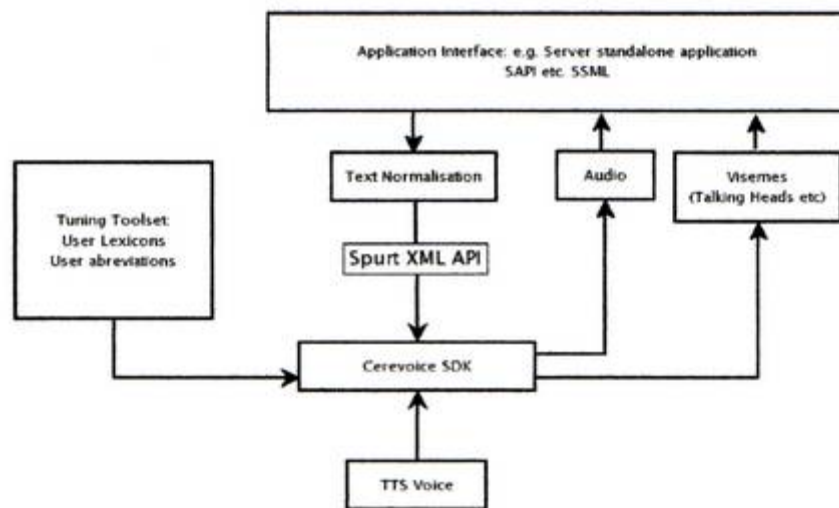


Abbildung 1: Überblick über das TTS-System CereVoice (Andersson et al. 2008)

Ein XML-API bestimmt das Input für das SDK. Das API basiert auf den sogenannten „Spurts“. Ein Spurt ist ein Abschnitt der Sprache zwischen zwei Pausen. CereVoice verwendet für die Textprozessierung ein modulares Python-System. Für die Generierung von Spurts wird ein Normalisierer verwendet. Alle Satzzeichen werden geparkt und in eine Ergebnisstruktur (Token-Buffer) in einen reservierten Platzhalter verschoben. Alle Abkürzungen werden aufgelöst. Alle Großbuchstaben werden durch Kleinbuchstaben ersetzt. Datum, Uhrzeit, Währung, Email, URL, Zahlen und Sonderzeichen werden mittels spezieller Regeln expandiert. Der Normalisierer baut als Ergebnis Token-Strukturen (Token-Buffer) auf, die Token und wichtige Informationen aus der Analyse enthält, z.B.

Index im Text, Normalisierungsregel, Satzzeichen vor und nach dem Token. Nach der Normalisierung werden alle Satzzeichen in SSML-break-Tags (Speech Synthesis Markup Language) übersetzt. Anhand der break-Tags werden Spurts erstellt. Anschließend werden die Spurts durch Reduktion und Homographen-Tags korrigiert. In Abbildung 2 ist in der ersten Spalte ein Text aus zwei Sätzen angegeben. In der zweiten Spalte ist er in Spurts aufgeteilt und mit Tags versehen.

<p>Research into expressive characters, for example embodied conversational agents, is a growing field, while new work in human-robot interaction (HRI) has also focused on issues of expressive behaviour. With recent developments in computer graphics, natural language engineering and speech processing, much of the technological platform for expressive characters both graphical and robotic is in place.</p>	<p>Research <lex phonemes='ih2 n t uw1'> into¹ </lex> <usel variant='1'> expressive² </usel> characters, for example embodied conversational agents, is a growing field, <break type='4' />³ while new work in <sig rate='0.8'> human-robot⁴ </sig> <lex phonemes='ih1 n t er0 ael k sh ax0 n'> interaction⁵ </lex> <break type='0' />⁶ (HR <usel variant='3'> I⁷ </usel>) has also focused on issues of expressive <lex phonemes='b ax0 hh eyl v y er0'> behaviour⁸ </lex>. With recent developments in computer graphics, natural language engineering and speech processing, <break type='4' />⁹ much of the technological <usel variant='1'> platform¹⁰ </usel> for expressive characters both graphical and robotic is in <usel variant='2'> place¹¹ </usel>.</p>
---	---

Abbildung 2: Ein in Spurts aufgeteilter Text (Aylett und Pidcock)

Erklärung für die SSML-Tags:

<p>2,7,10,11 <usel variant='... '></p>	<p>Innerhalb eines XML-Spurts kann ein Wort in „usel“-Tag eingeschlossen sein. In CereVoice es ist möglich, einen Teil vom</p>
--	--

	besten Pfad, der vom Viterbi-Algorithmus vorgeschlagen wurde, auszuschließen und eine weniger optimale Alternative, eine Variante (variant), zu bekommen. Dazu dient der „usel“-Tag. Im Fall von ‚expressive‘, wird aus der Datenbank eine Variante vorgeschlagen, die mehr nach ‚ixpressive‘ (Variante 0) als nach ‚expressive‘ (Variante 1) klingt. Durch den Tag wird statt Variante 0 Variante 1 verwendet.
1, 5, 8 <lex phonemes=' ...' >	Dieser Tag überschreibt die Aussprache, z.B. ändert die Betonung in der Aussprache.
3, 6, 9 <break type=' ...' >	Der „break“-Tag kontrolliert Pausen. Damit können Pausen entfernt, verkürzt oder verlängert werden.
4 <sig rate=' 0.8' >	Der „sig“-Tag steuert die Sprechrate. In diesem Beispiel wird sie gesenkt, weil <i>human-robot</i> kein gewöhnliches Kompositum ist.

Das Ergebnis der Textanalyse-Komponente ist ein normalisierter, aufgeteilter, mit allen Tags versehener Text, der im nächsten Schritt durch die Graphem-Phonem-Konvertierung in die Lautschrift überführt wird. In Abbildung 3 wird das TTS auf eine einfache Pipeline-Architektur zurückgeführt.

Die zwischen der Textprozessierung/Textvorverarbeitung und der Signalgenerierung/Synthese positionierte Graphem-Phonem-Konvertierung bildet den Schwerpunkt der Arbeit. Es gibt unterschiedliche Ansätze zur Graphem-Phonem-Konvertierung, die im weiteren Verlauf dieser Arbeit erläutert werden.

Die Graphem-Phonem-Konvertierung erfolgt anhand der Ausspracheregeln (regelbasierter Ansatz) oder mit Hilfe eines Lexikons (datenbasierter Ansatz). Diese Arbeit befasst sich mit einer datenbasierten Methode zur Graphem-Phonem-Konvertierung. Sie besteht aus drei Komponenten: Lexikon-Look-Up, Letter-to-Sound-Rules (LTS, Fragmente) und Back-Off. Das eingegebene Wort wird erst im Lexikon nachgeschlagen. Falls das Wort nicht im Lexikon eingetragen ist, wird seine Transkription aus Fragmenten zusammengesetzt. Für Teilstrings (meistens einzelne Buchstaben), die von Fragmenten nicht abgedeckt wurde, wird das Back-

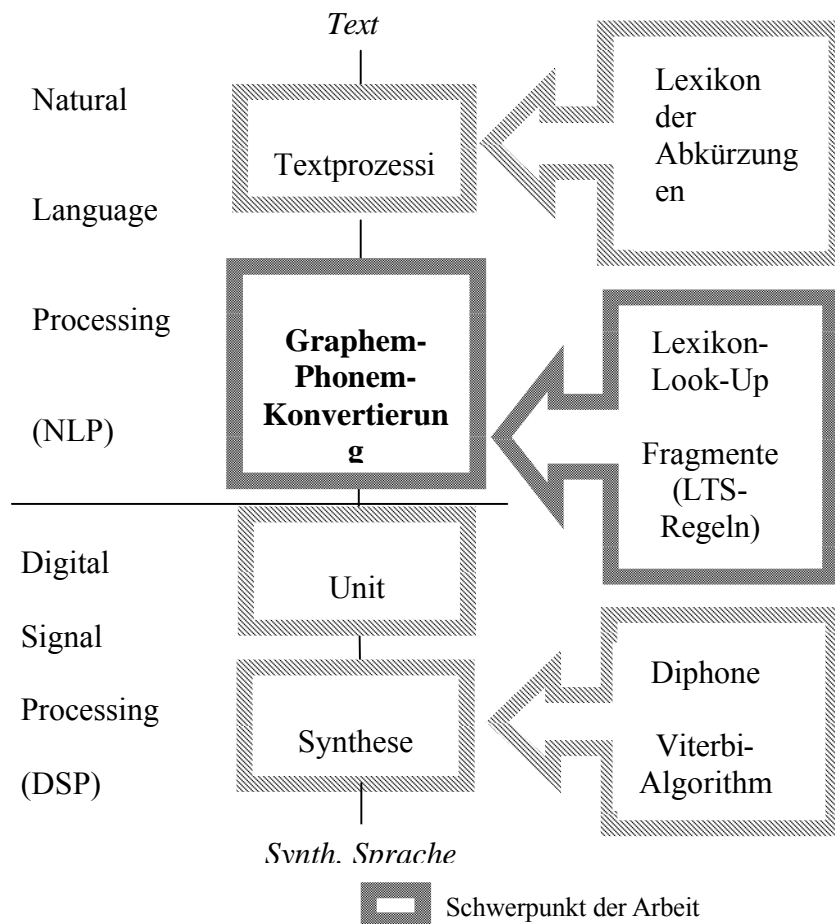


Abbildung 3: TTS als eine vereinfachte Pipeline –Architektur

Off-Verfahren herangezogen, das jedem übrig gebliebenen Buchstaben ein Phonem zuordnet und damit die Lücken in der Transkription schließt.

Das DSP-Modul erzeugt aus der symbolischen Information (Phonem-Strings) natürliche Sprache. Zur Generierung der Sprachausgabe gehört auch die Bestimmung der Zusammensetzung des Audiosignals. Dazu dient der Viterbi-Algorithmus, der aus der Datenbank geeignete Einheiten auswählt (Unit Selection) und diese miteinander verkettet (konkatenative Synthese). CereVoice verwendet Diphone als Einheiten. Ein Diphon ist ein Abschnitt von der Mitte eines Lautes bis zur Mitte des benachbarten Lautes. Das Inventar besteht damit aus einer Sammlung von allen in einer Sprache vorhandenen Übergängen zwischen zwei Lauten. Die Entscheidung für die Diphone ist durch die Tatsache, dass spektrale Eigenschaften in der Mitte eines Lautes stabil sind, begründet. Die Verkettungsstelle in der Mitte des Lautes wirkt perzeptiv nicht so störend wie im

Lautübergang. Da der Lautübergang erhalten bleibt, werden gegenseitige Einflüsse von benachbarten Lauten (koartikulatorische Effekte wie Assimilation, Reduktion, Verlust an Stimmhaftigkeit) erfasst. Ein großer Vorteil von Diphon-Systemen ist, dass die Qualität bei unterschiedlichen Texten konstant bleibt.

1.4 Bedeutung der Graphem-Phonem-Konvertierung für Sprachsynthese und Spracherkennung

Die Graphem-Zu-Phonem-Konversion (G2P) von isolierten Wörtern ist sowohl für die Sprachsynthese als auch für die Spracherkennung von Bedeutung. In der Sprachsynthese trägt ein leistungsstarker Graphem-Zu-Phonem-Modul zur Verbesserung der Sprachsynthese bei, da die Gesamtqualität eines TTS-Systems von seinen einzelnen Komponenten abhängig ist. Moderne TTS-Systeme verfügen über große Lexika, um die Aussprache von eingegebenen Wörtern im Lexikon nachzuschlagen. Diese Strategie stößt sehr bald auf ihre Grenzen. Zum einen sorgen produktive Wortbildungsprozesse und Neologismen für zahlreiche neue Wörter, die im Lexikon noch keinen Platz gefunden haben. Das Aktualisieren von Lexika ist ressourcenintensiv, zumal die Lexikoneinträge in der Regel manuell von einem Experten gemacht werden. Außerdem kann das Lexikon nicht unendlich erweitert werden. Dies gilt sowohl für Vollformen- als auch für Stammlexika. Aus diesen Gründen muss eine weitere Strategie zur Generierung der Aussprache in Betracht gezogen werden, die den oben genannten Herausforderungen gewachsen ist. Die Entwicklung von Werkzeugen zur automatischen Transkription stellt damit eine wichtige Aufgabe dar.

Für die Spracherkennung bringt ein automatisches Transkriptionsverfahren auch einige Vorteile mit sich. Mit Hilfe von G2P können neue Wörter ins Vokabular eines Spracherkenners eingefügt werden. Außerdem ist eine umgekehrte Abbildung von Phonemen auf Grapheme in der Spracherkennung gefragt, um die Konversion von Phonemen zum Text zu ermöglichen. Möglicherweise können Methoden zur Graphem-Phonem-Konvertierung in der Sprachsynthese auch zur Phonem-Graphem-Konvertierung in der Spracherkennung angewandt werden.

1.5 Gliederung der Arbeit

Der Schwerpunkt der Arbeit liegt in der zweiten Verarbeitungsstufe der Sprachsynthese, der Graphem-Phonem-Konversion (s. Abb. 3).

Die Arbeit ist wie folgt strukturiert. Nach einer Zusammenfassung von Grundlagen zur Graphem-Konvertierung in Kapitel 2 wird ein Überblick über den

aktuellen Forschungsstand und gängige Methoden zur Graphem-Zu-Phonem-Konversion in Kapitel 3 präsentiert. Im Mittelpunkt der Auseinandersetzung mit diversen Methoden zur Graphem-Phonem-Konvertierung stehen folgende Fragen: auf welchen Prinzipien basieren die Methoden; was setzen sie voraus; durch welche Vor- und Nachteile zeichnen sie sich aus; welche Ergebnisse liefern sie.

2 Grundlagen der Graphem-Phonem Konvertierung

2.1 Graphem-Phonem-Zuordnung

Die Aufgabe der Graphem-Zu-Phonem-Konversion ist die Abbildung einer Menge der Buchstaben des Eingabewortes auf eine Menge der korrespondierenden Phoneme. Diese Abbildung ist im mathematischen Sinne keine Funktion, da die Abbildung nicht eindeutig ist. Es ist nicht der Fall, dass jedem Element der ersten Menge genau ein Element der zweiten Menge zugeordnet wird.

Die Abbildung von Graphemen auf Phoneme ist keine Eins-Zu-Eins Abbildung, d.h. die Anzahl von Graphemen ist nicht gleich der Anzahl von Phonemen. Abbildung 4 stellt typische Graphem-Phonem-Abbildungsprobleme dar:

- Ein Graphem kann auf verschiedene Phoneme abgebildet werden, s. Abb. 4 (b).
- Ein Graphem wird auf mehrere Phoneme abgebildet, s. Abb. 4 (c).
- Mehrere Grapheme können auf ein Phonem abgebildet werden, s. Abb. 4 (d-e). In Abb. 4 (d) wird das Digraph <ee> dem Phonem /e:/ zugeordnet. In Abb. 4 (e) ist das Trigraph <sch> zu sehen.

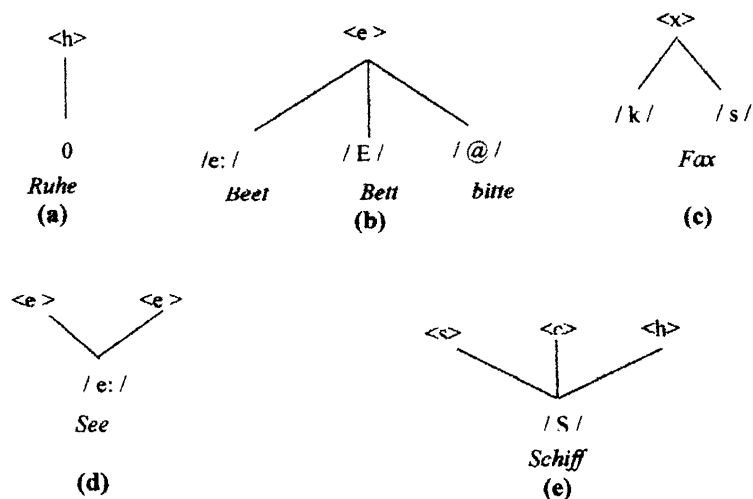


Abbildung 4: Graphem-Phonem-Zuordnungen. Grapheme stehen in Spitzklammern, während Phoneme (in SAMPA) zwischen den beiden Schrägstrichen positioniert werden.

Zur Erstellung von Graphem-Phonem-Mustern können Aussprachewörterbücher verwendet werden. Ein Aussprachewörterbuch bzw. ein TTS-Lexikon (auch in CereVoice) enthält in der Regel nur eine Zuordnung der orthographischen Folge zur Phonemfolge, z.B.

Sprache S_p_r_a:l_x_@0 (SAMPA, s. Anlage).

Bei der Erstellung von Graphem-Phonem-Mustern müssen einzelne Elemente der Graphem- und Phonemfolgen einander zugeordnet werden, z.B.

S p r a c+h e
S p r a:l x @0.

Hain (*Hain 2005*) beschreibt einige Verfahren zur Graphem-Phonem-Zuordnung. Das erste wurde in einem regelbasierten System verwendet (*Lawrence und Kaye 1986*). Laut Hain wird der Abbildungsprozess mit Hilfe von manuell erstellten Zuordnungslisten durchgeführt. In dem Fall, dass ein Graphem kein Phonem generiert, wird ein Null-Phonem eingeführt. Die Abbildung von einem Graphem auf ein Nullphonem wird mit fünf Sonderregeln behandelt. Das Wort und die Phonemfolge werden mit der Zuordnungstabelle verglichen.

Die dynamische Zeitanpassung (DTW, dynamic time warping) kann auch für die Graphem-Phonem-Zuordnung benutzt werden (*Pagel, Lenzo und Black 1998*). DTW ist ein Spezialfall der dynamischen Programmierung. DTW wird verwendet, um ein Muster aufgrund einer Folge von Merkmalen zu klassifizieren. Hain erklärt, dass das Prinzip der dynamischen Zeitanpassung darin besteht, ein unbekanntes Muster in Komponenten zu zerlegen und diese mit denen der typischen Vertreter der einzelnen Klassen zu vergleichen. Dazu wird eine Matrix aufgespannt, deren eine Achse durch die Komponenten der unbekanntes Folge und deren andere Achse durch die Komponenten der Referenzfolge aufgespannt wird. Laut Hain werden in die Felder der Matrix die Abstände zwischen jeweiligen Komponenten (lokale Bewertungen) eingetragen. Danach wird ein Pfad durch die Matrix gesucht, bei dem die Verknüpfung der einzelnen Bewertungen entlang des Pfades ein Optimum (ein minimaler Abstand oder maximale Ähnlichkeit) ergibt. Bei der Graphem-Phonem-Zuordnung ist die Aufgabe zu einer Graphemfolge nicht die ähnlichste sondern die wahrscheinlichste Phonem-Referenzfolge zu finden. Für die Suche des besten Pfades verwendet man Übergangshäufigkeiten bzw. Kostenfunktionen. Abbildung 5 zeigt einen Ausschnitt der Matrix für das Wort <schrieen> mit Übergangshäufigkeiten und dem vorläufigen besten Pfad.

n	0	0	0	0.052
e	0	0.005	0.052	0
e	0	0.017	0.052	0
i	0	0.052	0	0
r	0.078	0	0	0
	r	i:	@	n

Abbildung 5: Ausschnitt der Matrix für das Wort <schreien> mit dem vorläufigen besten Pfad (Hain 2005, S. 55)

2.2 Einflussfaktoren der Graphem-Phonem-Abbildung

Die Graphem-Phonem-Abbildung wird von einer Reihe von Faktoren beeinflusst. Dazu zählen Graphemumgebung, Silbenstruktur und Morphologie. Das folgende Beispiel zeigt vier korrespondierende Phoneme des Graphems <s>:

- (a) s → /s/ *Los*
- (b) s → /z/ *Rose*
- (c) s → /S/ *Straße*
- (d) s → 0 *Tasse*

In (d) hängt die Graphem-Phonem-Abbildung ganz von dem Graphemkontext ab. Das zweite <s> wird nicht ausgesprochen. In (a)-(c) reicht die Begründung der Graphem-Phonem-Abbildung alleine durch den Graphemkontext nicht aus. Der Versuch, die Graphem-Phonem-Abbildungen alleine vom Graphemkontext abhängig zu machen, resultiert in den folgenden misslungenen Graphem-Phonem-Abbildungsregeln:

- (e) s → /z/ | V_V *Lose **aber** Loserwerb*
- (f) s → /S/ | _t *Straße **aber** Gerüst*

Offensichtlich werden Graphem-Phonem-Abbildungen durch eine gemeinsame Wirkung von Graphemkontext, Morphologie und Silbenstruktur bestimmt. So ist Regel (f) gültig unter der Voraussetzung, dass <st> am Morphemangfang vorkommt, z.B. *Stuhl* oder *abstreiten*. Im Fall von *Loserwerb* in (e) ist die

Entscheidung schwieriger. Zwar tritt <s> in einer intervokalischen Position auf, aber diese Tatsache ist für die Graphem-Phonem-Abbildung irrelevant. Ausschlaggebend ist in diesem Fall die morphologische Struktur. Das Wort *Loserwerb* ist ein Kompositum, das sich aus zwei Wörtern *Los* und *Erwerb* zusammensetzt. Das Präfix *er-* im *Erwerb* verlangt eine vorangehende Silbengrenze, was zur Auslautverhärtung und zum Einfügen von einem glottalen Verschlusslaut führt.

Die morphologische Struktur manifestiert sich in der Silbenstruktur und bestimmt damit indirekt die Phonemidentität. Folglich wird <s> in *Loserwerb* trotz der vokalischen Umgebung auf ein stimmloses /s/ abgebildet. Sowohl in (e) als auch in (f) ist die morphologische Struktur des Wortes für die Graphem-Phonem-Abbildung essentiell. Betrachtet man *Los* in (a) und *Lose* in (e), stellt man fest, dass die Graphem-Phonem-Abbildung in *Los* vs. *Lose* auch durch den Unterschied in der Silbenstruktur erklärt werden kann. Durch das Einfügen eines Pluralmorphems *e* in *Lose* wird eine neue Silbe gebildet. So wird <s> zu /s/ in der Koda (s. Abb. 6 (a)) und zu /z/ im Onset (s. Abb. 6 (b)). Ähnlich ist die Situation in *täuschen* und *Häuschen*. Die Aussprache von <sch> hängt davon ab, wo sich die Morphemgrenze in dieser Graphemfolge befindet. Zwei unterschiedliche morphologische Strukturen erklären zwei unterschiedliche Silbenstrukturen und zwei verschiedene Aussprachen der gleichen Graphemfolge.

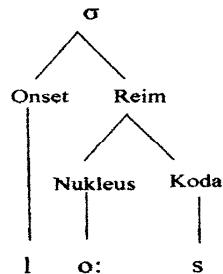


Abbildung 6(a):
Silbenstruktur für *Los*

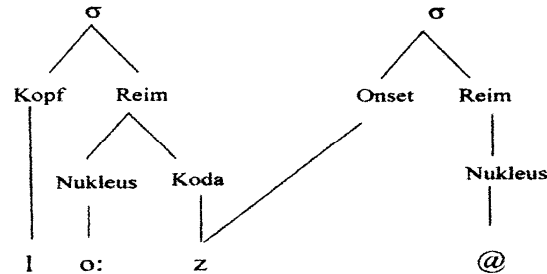


Abbildung 6 (b):
Silbenstruktur für *Lose*

3 Theoretische Ansätze und Verfahren zur Graphem-Phonem-Konvertierung

3.1 Strategien zur Graphem-Phonem-Konvertierung

TTS-Systeme werden zur Vertonung von beliebigen Texten verwendet. Ihre Funktion als „Vorleser“ wird, wie bereits in der Einführung erwähnt, im Radio und Web gerne genutzt. Dabei übernehmen die TTS-Systeme die Rolle eines Menschen, der beim Vorlesen die Konvertierung der Buchstaben in die Sprechlaute durchführt. Um diesen Vorgang in einem TTS-System zu automatisieren, muss man sich mögliche Strategien vorstellen, welche ein Mensch beim Vorlesen verfolgt. In der Fachliteratur findet man zwei verschiedene Strategien: Ein-Weg-Strategie und Zwei-Weg-Strategie.

Die erste Strategie (*Coltheart 1978*) geht davon aus, dass Menschen für die Aussprache von bekannten Wörtern das Lexikon benutzen, während sie für die Aussprache von unbekanntem Wörtern die Regeln verwendet. Diese Hypothese wird damit begründet, dass jeder imstande ist, sogenannte Pseudowörter auszusprechen, die es in der Sprache nicht gibt, und dass man mehr Zeit für die Aussprache von solchen Wörtern benötigt als für die Aussprache von regulären Wörtern. Diese Strategie wird oft in TTS-Systemen eingesetzt, wo ein Inputwort erst im Lexikon nachgeschlagen wird und unbekannte Wörter mit Graphem-Phonem-Regeln transkribiert werden. Die Entwicklung von solchen Regeln erfolgt von einem Experten, der ein großes Korpus analysiert und sein Wissen zusammen mit seiner Intuition in den Konvertierungsregeln formalisiert. Es ist ein komplexer Prozess mit einem großen Nachteil, dass zahlreiche Regeln sehr schnell unübersichtlich werden und viele Fehler verursachen können. Außerdem wird diese Strategie nicht allgemein anerkannt.

Die zweite Strategie wurde von Glushko (*Glushko 1979, Glushko 1981*) angenommen. In dieser Ein-Weg-Strategie ist keine Generierung von Regeln erforderlich. Die Aussprache von unbekanntem Wörtern wird von der Aussprache bereits bekannter Wörter durch Analogien abgeleitet. Wörter, die in ihrer Schreibweise einem unbekanntem Wort ähneln, werden im mentalen Lexikon aktiviert. Die aktivierten phonologischen Repräsentationen aus dem Lexikon werden zur Aussprache des unbekanntem Wortes kombiniert. Diese Theorie wird mit Ergebnissen von verschiedenen Experimenten belegt. In einem Experiment wurden zwei Typen von Pseudowörtern getestet:

- „Ausnahmen-Pseudowörter“, die Ähnlichkeit mit Wörtern mit inkonsistenter Aussprache haben, und
- „reguläre Pseudowörter“, die Wörtern mit konsistenter Aussprache ähnlich sind.

Ein Beispiel von einem regulären und einem Ausnahme-Pseudowort ist in Tabelle 1 zu sehen. Das Ausnahme-Pseudowort TAVE hat Nachbarn mit inkonsistenter Aussprache und kann deswegen als /tev/ oder /taev/ ausgesprochen werden. Das reguläre Pseudowort TAZE wird dagegen eindeutig als /tez/ ausgesprochen, da seine Nachbarn konsistente Aussprache haben.

	<i>Orthographische Repräsentation</i>	<i>Aussprache</i>
<i>Ausnahme:</i>	TAVE	/taev/ oder /tev/
<i>Nachbarn:</i>	HAVE	/haev/
	GAVE	/gev/
<i>Regulär:</i>	TAZE	/tez/
<i>Nachbarn:</i>	DAZE	/dez/
	GAZE	/gez/

Tabelle 1: Ausnahme-Pseudowort TAVE und reguläres Pseudowort TAZE mit ihren Aussprachen und ihren Nachbarn (*Dedina und Nusbaum 1991, S. 57*)

Glushko stellte fest, dass TAVE, das inkonsistente Nachbarn hat, eine längere Analysezeit erfordert als TAZE, dessen Nachbarn in ihrer Aussprache konsistent sind. Darüber hinaus werden Ausnahme-Pseudowörter wie TAVE langsamer ausgesprochen. Diese Ergebnisse sprechen dafür, dass die Aussprachen von beiden Wörtern und anderen Pseudowörtern mindestens zu einem gewissen Grad von der Aussprache anderer Wörter mit einer ähnlichen Schreibweise abhängig sind.

Die Ein- und Zwei-Weg-Strategien zur Graphem-Phonem-Konvertierung bilden ein theoretisches Fundament für die Methoden der Graphem-Phonem-Konvertierung. Die Zwei-Weg-Strategie findet ihren Ausdruck in den regelbasierten Methoden. Die Ein-Weg-Strategie liegt den datengetriebenen Methoden zugrunde. Damit lassen sich alle gängigen Methoden zur G2P in zwei Gruppen einteilen, wobei die zweite Gruppe sehr unterschiedliche Methoden einschließt:

- Regelbasierter Ansatz
- Datenbasierter Ansatz
 - Pronunciation by analogy, kurz PbA
 - Instance-based learning, kurz IBL
 - Neuronale Netze
 - Entscheidungsbäume

- Hidden Markov Modelle (HMM)

3.2 Regelbasierte Verfahren

Der regelbasierte Ansatz ist eine seit langer Zeit bekannte Methode, die Expertenwissen über die Sprache voraussetzt. In einem regelbasierten System wird Sprache als ein hierarchisches System dargestellt. Ergebnisse syntaktischer und morphologischer Analyse sowie Informationen über die Silbentrennung und Betonung werden in die Erstellung von Graphem-Phonem-Abbildungen einbezogen. Dies erfordert einen hohen Aufwand an Sprachverarbeitung, der in einem multilingualen System mit jeder neuen Sprache wächst, da die gewonnenen Erkenntnisse sprachspezifisch sind. Die größte Schwierigkeit ist das optimale Zusammenspiel aller Regeln miteinander. In einem regelbasierten System mit gewöhnlich Hunderten Regeln stellt eine genaue Abstimmung der Regeln aufeinander die größte Herausforderung dar.

Hain (*Hain 2005*) stellt drei regelbasierte Systeme vor. Das erste wurde von Wothke (*Wothke 1993*) vorgeschlagen. Wie in vielen regelbasierten Systemen erfolgt auch hier die morphologische Zerlegung vor der Graphem-Zu-Phonem-Konversion. Im ersten Verarbeitungsschritt werden Eingabewörter in Präfixe, Stämme und Suffixe zerlegt. Als Wissensquellen für die morphologische Zerlegung dienen ein Morphemlexikon und eine Wortsyntax. Im Morphemlexikon werden alle Morpheme mit ihren Klassen (bis zu 6 Klassen) eingetragen. Morpheme werden in (1) Präfixe, Stämme, Suffixe und (2) Verb-, Adjektiv- und Nomenmorpheme eingeteilt. Die Einträge enthalten auch zusätzliche Eigenschaften wie Numerus, Kasus, Zeit, Modus, Komparation und Ablaut. Die Aufgabe der Wortsyntax besteht darin, für Deutsch gültige Sequenzen von Morphemen zu bestimmen. Der zweite Verarbeitungsschritt beinhaltet die Graphem-Phonem-Konvertierung mittels Regeln. Das folgende Beispiel zeigt, wie durch die Verwendung der Graphem- und Phonemgruppen die Anzahl von Regeln verringert wird. Im Regelsystem werden folgende drei Einträge in (a) zu einer Regel in (b) zusammengefasst (*Hain 2005*, S. 9).

(a)	(b)
[b] t -> p	
[b] s -> p	[b]/VLCONS/ -> p
[b] k -> p	

VLCONS ist eine Gruppe von stimmlosen Konsonanten (voiceless consonants). Es gibt 31 solcher Gruppen. Die Konvertierungsregeln haben die Form:

links [Zentrum] rechts → Phonem1 Phonem2

Die Graphemfolge in den eckigen Klammern auf der linken Seite der Regel wird

im gegebenen Kontext der Phonemfolge auf der rechten Seite der Regel zugeordnet. Die morphologische Struktur des Wortes wird berücksichtigt, indem man bei der Angabe auf der linken Seite der Regel Bezug auf die verschiedenen Morphemgrenzen nimmt.

Laut Hain wurde der regelbasierte Ansatz auch im TTS SVOX (Traber 1995) für die Graphem-Konvertierung eingesetzt. Das eingegebene Wort wird erst im Lexikon nachgeschlagen. Außer der Aussprache enthält das Lexikon auch syntaktische Eigenschaften, die bei dem Homographenproblem hilfreich sind. Die Aussprachevarianten sind mit Strafwerten versehen, die von den syntaktischen Eigenschaften abhängig sind. Unbekannte Wörter werden morphologisch zerlegt, bevor die Regeln zur Graphem-Phonem-Konvertierung starten. Analog zu den Lexikoneinträgen enthalten auch die Regeln einen Strafterm, der bei mehreren Möglichkeiten die Auswahl der richtigen steuert. Das Einstellen von den Straftermen der Grammatikregeln und der Lexikoneinträge ist das Hauptproblem des Verfahrens.

Morphologische Zerlegung und die anschließende Graphem-Phonem-Konvertierung bilden den Kern des Verfahrens von Kommenda (Kommenda 1991). Die morphologische Zerlegung geschieht anhand eines morphologischen Lexikons mit ca. 2500 Morph-Einträgen. Viele Stämme sind in einer reduzierten Form gespeichert, z.B. *Klapp* statt *Klappe*. Dies ermöglicht eine kompakte Darstellung sowohl von Derivaten als auch von Komposita (*klappen*, *Klappstuhl*). Bei jedem eingetragenen Morph sind seine phonologischen, morphologischen und syntaktischen Eigenschaften angegeben. Bei der Angabe der Konjugations- und Derivationsmustern werden auch Informationen eingetragen, die nicht nur das Morph selbst, sondern das folgende Morph betreffen. Endungen und Fugen werden sehr detailliert beschrieben und klassifiziert, damit man feststellen kann, ob ein Stamm oder ein Derivationsuffix verlangt wird. Es wird versucht, das eingegebene Wort so in Teilstrings zu zerlegen, dass diese im Lexikon vorhanden sind und bei ihrem Aneinanderreihen das gesamte Wort entsteht. Danach wird mit einem Klassifikationsschema und einer Strukturformel überprüft, ob die gefundene Zerlegung gültig ist. In den Fällen, wo Wörter nicht vollständig zerlegt werden können, kommt das „Joker“-Verfahren ins Spiel. Dieses bestimmt die Zerlegung von unbekanntem Wörtern durch die Strukturformel (Hain 2005, S.11):

$$[K_i +] V[[+K_m] +V][+ K_f]$$

mit

+	Verkettung
K _i	initialer Konsonantencluster
V	Vokalcluster
K _m	medialer Konsonantencluster
K _f	finaler Konsonantencluster
[]	optionale Elemente

Die Transkription wird aus den Transkriptionen von Morphen zusammengesetzt und in Abhängigkeit vom Kontext modifiziert, z.B. durch die Auslautverhärtung.

In allen diesen Systemen hängt die Korrektheit der Aussprache von den Ergebnissen der morphologischen Zerlegung ab. Dies erfordert einen großen Apparat an Klassen und Regeln, die sehr sorgfältig erstellt werden müssen. Sie gelten nur für die Sprache, für die sie spezifiziert sind. Die datenbasierten Methoden können im Gegensatz an unterschiedlichen Daten (Sprachen oder Dialekten) angewandt werden.

3.3 Datenbasierte Verfahren

3.3.1 Pronunciation by Analogy

Die Tatsache, dass viele Wörter, die nicht im Lexikon stehen und transkribiert werden müssen, in ihrer Orthographie und Aussprache bereits vorhandenen Einträgen im Lexikon ähneln, wird von datenbasierten Methoden genutzt. Im Unterschied zum regelbasierten Ansatz werden in datenbasierten Techniken meist keine expliziten kontextsensitiven Regeln erstellt. Sie sind aber implizit in ganz anderer Form vorhanden. So kann eine von einem Entscheidungsbaum gestellte Frage als eine Art kontextsensitive Regel betrachtet werden. Mit Recht sieht Taylor (*Taylor 2009*) den Unterschied zwischen daten- und regelbasierten Methoden in der Art und Weise, wie diese Regeln entwickelt werden: per Hand mit sprachspezifischem Wissen oder durch maschinelles Lernen.

Das Prinzip

In diesem Ansatz wird die Aussprache von unbekanntem Wörtern mit Hilfe von implizitem phonologischem Wissen aus dem Lexikon abgeleitet. Der Algorithmus sucht nach den Übereinstimmungen zwischen den Teilfolgen eines Eingabewortes und den Teilfolgen der Lexikoneinträge, indem er die Buchstaben im Input-Wort mit jedem Lexikoneintrag vergleicht. Die Transkriptionen von den gefundenen Übereinstimmungen bilden partielle Transkriptionen, die zum Schluss zur kompletten Transkription zusammengestellt werden. Für jede übereinstimmende Teilfolge wird ein so genanntes Aussprachenetz aus Input-Buchstaben und Output-Phonemen erzeugt.

Dieses Netz kann als ein gerichteter Graph mit einer Menge von Knoten und Kanten dargestellt werden (s. Abb. 7). Jeder Knoten repräsentiert einen Buchstaben und sein korrespondierendes Phonem. Die Kanten verbinden einzelne Knoten, erstrecken sich über Knotenfolgen und bilden Pfade. Alle im Lexikon

beobachteten Graphem-Phonem-Abbildungen werden gezählt und zur Gewichtung der Kanten verwendet.

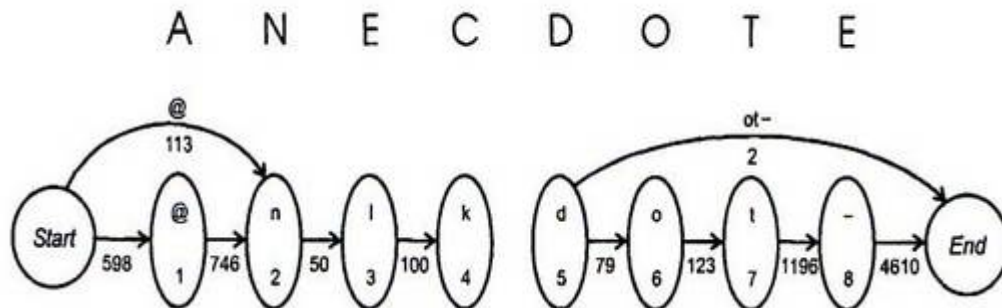


Abbildung 7: Ein Teil des Aussprachenetzes für das Wort „anecdote“. Zur Vereinfachung wird nur ein Teil der Kanten dargestellt (*Marchand und Damper 2000, S. 2000*)

Das Netz repräsentiert alle möglichen, im Lexikon gelernten Aussprachen des Eingabewortes. Eine mögliche Aussprache entspricht einem kompletten Pfad, einem Weg durch das Aussprachenetz. Sie setzt sich aus den Phonem-Labels der Knoten, die durchlaufen wurden. In einem Aussprachenetz gibt es in der Regel mehrere Aussprachen. Zum Schluss wird eine Entscheidungsfunktion angewandt, die die beste Aussprache bestimmt.

Die sorgfältige Gestaltung einer Entscheidungsfunktion ist eine wichtige Aufgabe, die zur Optimierung des ganzen Algorithmus beitragen kann. Der Ursprungsansatz war den kürzesten Pfad (einen Pfad mit längsten einzelnen Spannweiten) durch das Netz zu finden. Im Fall von mehreren kürzesten Pfaden kann der Pfad mit der größten Summe der Häufigkeiten der Kanten bevorzugt werden. Alternativ können das Produkt oder das Minimum von Häufigkeiten der Kanten berechnet werden. Die Anzahl der gleichen oder unterschiedlichen Aussprachen sowie die Standardabweichung von der Pfadstruktur können ebenso in die Entscheidung einbezogen werden. Diese Strategien werden später betrachtet.

PRONOUNCE

PRONOUNCE ist ein Programm, das zu einem Eingabewort seine Aussprache automatisch nach dem Pronunciation-by-Analogy-Prinzip berechnet. Es wurde von Michael J. Dedina und Howard C. Nusbaum (*Dedina und Nusbaum 1991*) entwickelt. Eine ausführliche Beschreibung von PRONOUNCE findet man bei Marchand und Damper (*Marchand und Damper 2000*). Das System besteht aus 4 Komponenten (s. Abb. 8):

- einem Lexikon;
- einem Modul, das das Eingabewort mit allen Lexikoneinträgen vergleicht und Übereinstimmungen findet;
- einem Aussprachenetz, das alle möglichen Aussprachen repräsentiert, und
- einer Entscheidungsfunktion, die unter allen möglichen die beste Aussprache auswählt.

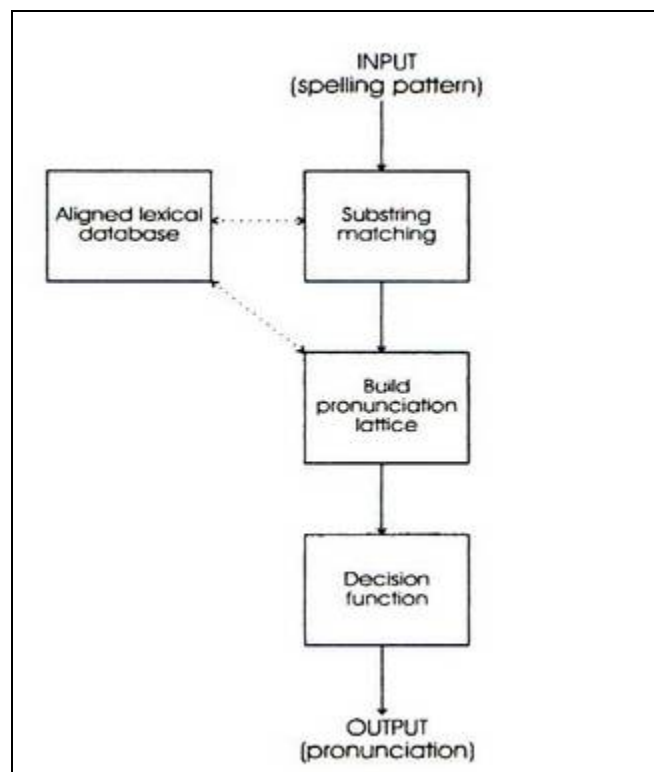


Abbildung 8: Blockdiagramm von PRONOUNCE von Dedina und Nusbaum (*Dedina und Nusbaum 1991*)

1) Das Lexikon

Das Lexikon von PRONOUNCE besteht aus ca. 20000 Wörtern aus Webster's Pocket Dictionary. Jeder Eintrag enthält eine Abbildung von den Buchstaben des Wortes auf die korrespondierenden Phoneme. Diese Abbildung ist das Output eines Lisp-Programms, das nur wenige Informationen benötigt. Das Programm weiß, ob ein Graphem bzw. ein Phonem zu Konsonanten oder zu Vokalen gehört. Es zerlegt orthographische und phonetische Strings in zwei Gruppen: Konsonanten vs. Vokale. Dann werden konsonantische Grapheme auf konsonantische Phoneme und vokalische Grapheme auf vokalische Phoneme abgebildet.

2) Suche nach Übereinstimmungen

PRONOUNCE nutzt das Prinzip der Analogie. Das Kriterium für die Analogie ist einfach gestaltet. Wenn das Eingabewort und ein Lexikoneintrag mindestens einen gemeinsamen Buchstaben haben, kann die phonetische Information aus diesem Eintrag für die Generierung der Aussprache des Eingabewortes nützlich sein. Das eingegebene Wort wird mit allen Wörtern im Lexikon verglichen und übereinstimmende Teilstrings werden identifiziert. Die Voraussetzung ist, dass die Graphem-Phonem-Abbildung im Lexikon eine Eins-Zu-Eins-Abbildung ist.

Die Suche nach übereinstimmenden Teilfolgen beginnt mit dem ganz links stehenden Buchstaben des Eingabewortes. Dann wird der kürzere String nach rechts um einen Buchstaben verschoben. Die Verschiebung endet am Ende des längeren Strings. In jeder Position bei jeder Verschiebung werden die beiden Strings miteinander verglichen und gemeinsame Teilstring werden in das resultierende Aussprachenetz aufgenommen. Abbildung 9 illustriert die Suche nach den Übereinstimmungen zwischen dem eingegebenen Pseudowort *blope* und dem Lexikoneintrag *sloping*.

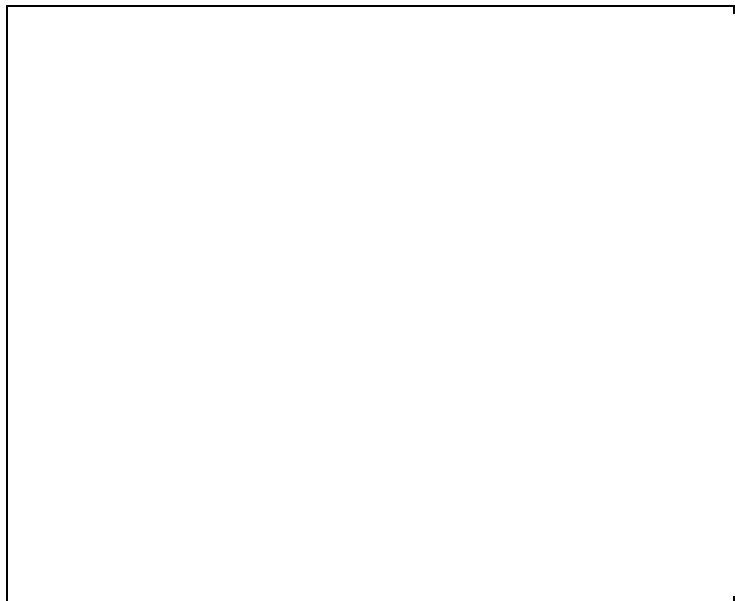


Abbildung 9: Ein Beispiel für die Suche nach Übereinstimmungen in PRONOUNCE. Das Eingabewort *<blope>* wird mit dem Lexikoneintrag *<sloping>* in drei Positionen verglichen (*Dedina und Nusbaum 1991, S.59*)

Das Ergebnis ist ein Aussprachenetz. Ein Knoten im Aussprachenetz repräsentiert einen übereinstimmenden Buchstaben L_i , wobei i die Position im Input-String ist. Der Knoten selbst ist mit einem Label mit Index i und Phonem

