

# Token-Wort-Konvertierung in der Text-To-Speech-Synthese

Studienarbeit Nr. 34

an der

Universität Stuttgart  
Institut für Maschinelle Sprachverarbeitung  
Azenbergstraße 12  
70174 Stuttgart

vorgelegt von

Christiane Schunk

Prüfer: PD Dr. phil. Bernd Möbius  
Prüfernummer: 01386

Betreuerin: Antje Schweitzer

04.06.2004 - 03.12.2004



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

Unterschrift



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Was ist Sprachsynthese? . . . . .	4
1.2	Was sind Non-Standard-Words? . . . . .	9
1.3	Wie funktioniert Festival? . . . . .	14
<b>2</b>	<b>Vorgehen bei der Suche nach Belegen in Korpora</b>	<b>17</b>
2.1	Abfragen . . . . .	17
2.2	Korpora . . . . .	19
2.2.1	Deutsche Abkürzungen . . . . .	19
2.2.2	European Languages News Corpus . . . . .	19
2.2.3	IMS Corpus Workbench Demo Corpus . . . . .	20
2.2.4	Handelsblatt . . . . .	20
2.2.5	Computerzeitung . . . . .	20
2.2.6	Internetadressen . . . . .	21
<b>3</b>	<b>Auswertung: Beschreibung der Problemklassen und Lösungen</b>	<b>22</b>
3.1	Akronyme . . . . .	22
3.1.1	Akronyme aus 2, 3 oder 4 Großbuchstaben . . . . .	22
3.1.2	Akronyme aus mindestens 5 Großbuchstaben / Wörter, die mehrere Großbuchstaben enthalten / Akronyme aus Kleinbuchstaben	23
3.2	sonstige Abkürzungen . . . . .	24
3.2.1	Abkürzungen mit Punkt . . . . .	24
3.2.2	Einzelbuchstaben bzw. mehrere Wortanfänge durch Punkte getrennt . . . . .	26
3.2.3	Kombinierte Abkürzungen . . . . .	27
3.3	Maßeinheiten . . . . .	27
3.4	Internetadressen . . . . .	28
3.5	eMail-Adressen . . . . .	29
3.6	Zahlen und Buchstaben gemischt . . . . .	29
3.7	Kardinalzahlen . . . . .	30
3.8	Ordinalzahlen . . . . .	31
3.9	Römische Zahlen . . . . .	32

3.10	Rationale Zahlen . . . . .	33
3.10.1	Kommazahlen . . . . .	33
3.10.2	Brüche . . . . .	33
3.11	Aufzählung . . . . .	34
3.12	Telefonnummern und Faxnummern . . . . .	35
3.13	Postfach und Postleitzahlen . . . . .	36
3.14	Bankleitzahlen und Kontonummern . . . . .	36
3.15	Zeitangaben . . . . .	37
3.16	Datum . . . . .	38
3.17	Jahreszahlen . . . . .	39
3.18	Geld . . . . .	39
3.19	Verhältniszahlen . . . . .	39
3.20	Sonderzeichen . . . . .	40
3.20.1	Bindestrich . . . . .	40
3.20.2	Prozent . . . . .	41
3.20.3	sonstige Sonderzeichen . . . . .	41
<b>4</b>	<b>Schlusswort</b>	<b>43</b>

# Kapitel 1

## Einleitung

Ziel dieser Arbeit ist die Evaluierung und Überarbeitung der Token-Wort-Konvertierung in IMS-Festival. Bei der Auswertung von synthetisierten Texten im Rahmen des Projektes Smartkom sind einige Probleme der Token-Wort-Konvertierung in IMS-Festival aufgefallen, z.B. wurde *CD-Rohling* bei der Synthese zu *römisch vierhundert Rohling* konvertiert, da alle Kombinationen von Buchstaben, die eine römische Zahl darstellen könnten, auch als solche expandiert wurden. Ein anderes Problem bestand darin, dass ein Schrägstrich zwischen zwei Zahlen ausnahmslos als Bruch erkannt wurde, was z.B. im Fall von Jahresangaben wie *91/92* falsch ist. Weitere Fehler werden in Kapitel 3 genauer erläutert. Ziel dieser Arbeit ist eine genaue Analyse der auftretenden Fehler, ihre Klassifizierung, sowie ihre Behebung, sofern möglich. Ausserdem soll die Token-Wort-Konvertierung um Algorithmen für bisher nicht behandelte Muster ergänzt werden.

Die Analyse soll anhand von gezielt aus einer größeren Datenmenge ausgewählten Sätzen erfolgen. Durch die systematische Extraktion von Daten aus unterschiedlichen Korpora kann besser abgeschätzt werden, wie häufig Probleme tatsächlich auftreten. Außerdem kann bei Ambiguitäten eher entschieden werden, welche Lesart häufiger ist.

In diesem Kapitel wird einführend auf Sprachsynthese, Non-Standard-Words und Token-Wort-Konvertierung, sowie die Funktionsweise von IMS-Festival eingegangen. In Kapitel 2 wird beschrieben, wie die Daten für die Auswertung aus Korpora gezielt extrahiert wurden. Kapitel 3 beschreibt die gefundenen Probleme bei der Behandlung von Non-Standard-Words und deren Lösung.

## 1.1 Was ist Sprachsynthese?

Sprachsynthese ist die künstliche Nachbildung natürlicher Sprache. Ihre Anwendungsmöglichkeiten sind vielseitig. Sie wird eingesetzt, wo kein oder kein geeignetes Display zur Verfügung steht um sich Sprache anzeigen zu lassen, z.B. für SMS ins Festnetz oder in Dialogsystemen. Auch in Situationen, in denen die Augen schon mit anderen Aufgaben beschäftigt sind, wie z.B. beim Führen eines Kraftfahrzeuges, ist Sprachsynthese hilfreich. In Fahrzeugen wird sie z.B. in Navigationssystemen eingesetzt. Sprachsynthese ist für Blinde von großem Nutzen, die sich Texte aus dem Internet oder von ihrem Computer vorlesen lassen können. Sprechbehinderte können sie nutzen um mit ihrer Umwelt zu kommunizieren (Möbius, "Sprachsynthesysteme" S.462).

Es existieren unterschiedliche Ansätze der Sprachsynthese, z.B.: Text-To-Speech- und Concept-To-Speech-Synthese. Die Concept-To-Speech-Synthese enthält eine Generierungskomponente, die eine Äußerung aus semantischem, pragmatischem und Diskurswissen generiert. Aus dieser Äußerung kann direkt das Sprachsignal erzeugt werden. Da das System seine Äußerungen selbst generiert, liegen alle morphologischen, syntaktischen, pragmatischen und semantischen Informationen vor, mit denen Wörter gleich in ihrer korrekten Form generiert werden können. Daher entsteht das Textvorverarbeitungsproblem von Text-To-Speech-Systemen in der Concept-To-Speech-Synthese nicht. Durch diesen Ansatz verspricht man sich außerdem eine deutliche Verbesserung der Prosodie und damit auch eine Verbesserung der Synthese (Möbius, "Sprachsynthesysteme" S.467-468). Die Concept-To-Speech-Synthese kann z.B. in Dialog- und in Navigationssystemen verwendet werden, überall da allerdings, wo Text die Eingabe ist, muss Text-To-Speech-Synthese (TTS) eingesetzt werden.

Die Text-To-Speech-Synthese erfolgt in mehreren Stufen. Das TTS-System erhält als Eingabe einen Text, der, bevor daraus ein Sprachsignal erzeugt werden kann, zunächst analysiert und in eine phonetische Beschreibung transformiert werden muss. Danach wird in einem weiteren Schritt die Prosodie generiert. Aus den dann vorliegenden Informationen kann das Sprachsignal entstehen. Eine Übersicht über den Aufbau der TTS-Synthese gibt Abbildung 1.1 (nach Dutoit, *An Introduction to Text-To-Speech Synthesis* S.63).

Nach Dutoit (*An Introduction to Text-To-Speech Synthesis* S.62) besteht das Sprachverarbeitungsmodul eines derzeitigen typischen TTS-Systems aus Textanalyse, Ausspracheregeln und Prosodiegenerierung. Die Textanalyse wiederum besteht aus den Kompo-

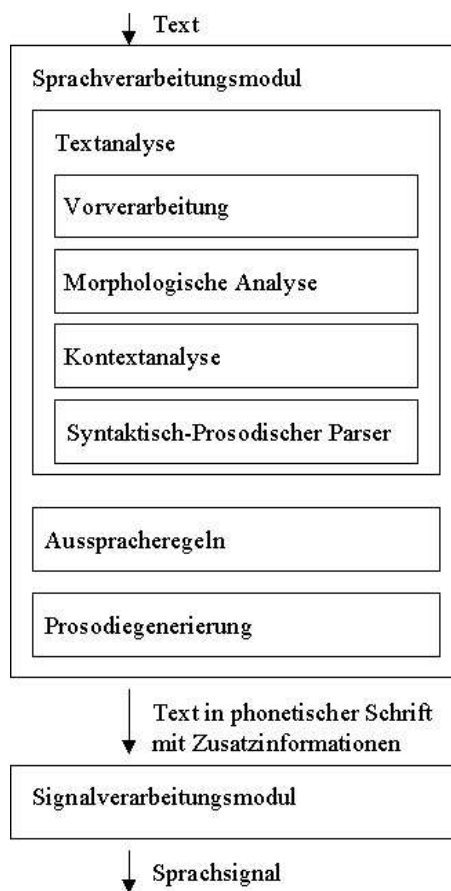


Abbildung 1.1: Stufen der TTS-Synthese

nenten Vorverarbeitung, morphologische Analyse, Kontextanalyse und dem syntaktisch-prosodischen Parser. Während der Vorverarbeitung wird der Text zunächst in Token segmentiert. Dieser Vorgang wird auch Tokenisierung genannt. Dabei gehört zu einem Token das, was durch ein Leerzeichen von seinem Umfeld getrennt ist. Häufig sind das Wörter, doch sind auch z.B. kombinierte Abkürzungen wie *Wohns.-Nr.* oder Zahlen wie *2004* jeweils ein Token. Der Satz *Das Haus mit der Nr. 12 wurde 1997 gebaut.* besteht aus den Token: *'Das' 'Haus' 'mit' 'der' 'Nr' '12' 'wurde' '1997' 'gebaut'*. In den meisten TTS-Systemen wird die Interpunktion abgetrennt und separat gespeichert, damit sie getrennt ausgewertet werden kann. In der Token-Typ-Erkennung wird dem Token ein Typ wie z.B. *Jahreszahl*, *URL* oder *Abkürzung* zugeordnet. Das Token *Nr.* aus unserem Beispielsatz bekommt den Token-Typ *Abkürzung*, das Token *12* den Typ *Kardinalzahl* und das Token *1997* bekommt den Typ *Jahreszahl*. Die Token-Wort-Konvertierung bildet die orthographische Form des Token. Hier wird für das Token *Nr.* durch Expansion die orthographische Form *Nummer* gebildet, das Token *12* erhält die orthographische Form *zwölf* und *1997* wird in *neunzehnhundertsiebenundneunzig* transformiert. Dass diese Expansion unter Umständen nicht so leicht ist, sieht man am Beispiel der Zahl *1*: Je nachdem, was sie bezeichnet, muss sie unterschiedlich expandiert werden, im Fall einer Hausnummer zu *eins*, in *1 Kilogramm* zu *ein*, oder in der Äußerung *1 Katze jagt 1 Hund*, sollte *1* zuerst zu *eine* und dann zu *einen* expandiert werden. All diese Token, deren orthographische Form erst berechnet werden muss, nennt man 'Non-Standard-Words' (NSWs). Es ist vorteilhaft, wenn die Satzendeerkennung parallel zur Token-Wort-Konvertierung läuft, da Interpunktion ambig sein kann, z.B. sind Punkte außer am Satzende auch in Ordinalzahlen und Abkürzungen zu finden, Doppelpunkte erscheinen auch in Zeitangaben.

Nach der Vorverarbeitung folgt die morphologische Analyse. Dutoit unterscheidet (*An Introduction to Text-To-Speech Synthesis* S.77) zwischen Funktionswörtern und Inhaltswörtern, wobei Wörter wie z.B. *und*, *auf*, *in*, *oder*, *er* Funktionswörter sind. Die Anzahl der Funktionswörter einer Sprache ist endlich, d.h. es ist möglich sie im Lexikon zu speichern. Da ihre Aussprache nicht immer regelmäßig ist, ist die Speicherung im Lexikon ein weiterer Vorteil. Anders verhält es sich bei den Inhaltswörtern. Hiervon gibt es unendlich viele, daher ist die Speicherung aller in einer Sprache vorkommenden Wörter unmöglich. Zum einen entstehen laufend neue Wörter, ob durch Komposition von Wörtern (vor allem im Deutschen: z.B.: *Donaudampfschiffahrtskapitän*, wo beliebig viele Nomina aneinandergehängt werden können), durch den Einfluss anderer Sprachen

oder durch Neuentwicklungen. Zum anderen existieren in manchen Sprachen aufgrund ihrer Flexion und Derivation sehr viele Wortformen. Durch die morphologische Analyse werden Informationen über ein Wort gewonnen, durch die Aussprache, Silbentrennung und die Betonungsberechnung verbessert werden. Wenn nur noch Wortstämme anstatt sehr vieler Vollformen gespeichert werden müssen, kann dadurch viel Speicherplatz gespart werden.

Während der Kontextanalyse (Tagging) wird die Umgebung eines Token analysiert, da es nach der morphologischen Analyse immer noch Ambiguitäten gibt, die nur durch die Betrachtung des syntaktischen Kontexts aufgelöst werden können: Expansion von Abkürzungen wie z.B. *tgl.*, (Dutoit, *An Introduction to Text-To-Speech Synthesis* S.75) von der man ohne Kontextanalyse nicht weiß, ob sie zu *täglich*, *tägliche*, *täglichem*, *täglichen*, *täglicher* oder *tägliches* expandiert werden soll. Die Kontextanalyse wird im Deutschen auch benötigt, um heterophone Homographen zu disambiguieren: z.B. *'modern* und *mo'dern*, die anhand ihrer Schreibweise nicht unterschieden werden können.

Syntaktisch-prosodisches Parsing: Auch nach der Kontextanalyse sind immer noch Ambiguitäten vorhanden. Dies tritt z.B. bei Wörtern, die in sich ambig sind, und Wörtern, denen mehrere Kategorien zugewiesen werden können, auf. Außerdem erscheinen solche Ambiguitäten bei Wörtern, die von der morphologischen Analyse nicht zerlegt werden konnten, z.B. durch Schreibfehler oder durch ein unvollständiges Lexikon. Hier kann syntaktisches Parsing einen Satz in Phrasen zerlegen und seinen Wörtern Part-of-Speech-Kategorien zuweisen. Manche Sätze erhalten durch syntaktisches Parsing mehrere mögliche Strukturen (z.B. *Ich sah den Mann mit dem Teleskop*). Um ihnen eine Struktur davon eindeutig zuzuweisen, ist zusätzliches semantisches Wissen nötig.

Nachdem die Textanalyse abgeschlossen ist, können Ausspracheregeln angewendet werden. Buchstaben können nicht 1:1 in Phoneme überführt werden, da die Entsprechung nicht immer parallel verläuft. Ein einzelner Buchstabe kann in bestimmten Umgebungen keinem (z.B. *h* in *geht*) oder mehreren Phonemen (*x* in *Fixkosten*) entsprechen. Außerdem können mehrere Buchstaben einem Phonem entsprechen (*ch* in *ich*). Buchstaben können in unterschiedlichen Umgebungen unterschiedlich ausgesprochen werden (*s* in *Stadt* vs. *Sachen*). Und ein Phonem kann durch unterschiedliche Buchstaben zustande kommen (*Rat* vs. *Rad*) (Dutoit, *An Introduction to Text-To-Speech Synthesis* S.106). Um die Aussprache von Wörtern zu bestimmen gibt es drei Strategien: In wörterbuchbasierten Lösungen mit morphologischer Komponente werden so viele Morpheme wie

möglich in einem Lexikon gespeichert. Vollformen werden durch Flexions-, Derivations- und Kompositionsregeln erfasst. In einer regelbasierten Lösung werden aus dem phonologischen Wissen von Wörterbüchern Ausspracheregeln generiert. Nur Wörter, die in ihrer Aussprache eine absolute Ausnahme bilden, werden in ein Ausnahmewörterbuch aufgenommen. Die beiden Ansätze unterscheiden sich sehr in der Größe ihrer Lexika, dasjenige der wörterbuchbasierten Lösungen ist um ein Vielfaches größer als das Ausnahmewörterbuch der regelbasierten Lösungen. Wörterbuchbasierte Lösungen können eventuell genauer sein als regelbasierte Lösungen, wenn sie ein genügend großes phonetisches Wörterbuch zur Verfügung haben (Dutoit, *An Introduction to Text-To-Speech Synthesis* S.112). Bei der dritten Strategie wird ein sogenanntes Vollformlexikon aufgebaut, wobei so viele Vollformen wie möglich gespeichert werden (z.B. in IMS-Festival). Die Aussprache derjenigen Wörter, die nicht im Lexikon eingetragen sind, wird durch Ausspracheregeln bestimmt.

Wenn die Aussprache der Wörter bestimmt ist, folgt die Prosodiegenerierung. Der Grad der Natürlichkeit eines TTS-Systems ist abhängig von prosodischen Faktoren wie der Intonationsmodellierung (Phrasierung und Akzentuierung), Amplitudenmodellierung und der Dauermodellierung (dazu gehören Lautdauer und Pausendauer, wodurch sich die Silbendauer und das Sprechtempo ergeben). Prosodische Merkmale haben verschiedene Funktionen: Durch sie kann z.B. der Fokus eines Satzes erkannt werden, d.h. dass eine Konstituente als wichtig oder neu hervorgehoben wird. Außerdem sind sie zuständig für die Segmentierung eines Satzes. Sie können Beziehungen zwischen Satzteilen oder Sätzen herstellen und bestimmen den Satzmodus (Aussagesatz - Fragesatz). Syntaktische Informationen sind von besonderer Bedeutung für die Prosodie-Generierung. Durch das Wissen über die syntaktische Struktur eines Satzes kann für die meisten Sätze die Prosodie berechnet werden. Für einige Sätze jedoch ist semantische und pragmatische Information wichtig: Sätze, die in ihrer syntaktischen Struktur ambig sind, erhalten oft je nach betonter Komponente eine andere Bedeutung. Die Position des Fokus ist vor allem in verneinten Sätzen wichtig: die Komponente, auf die sich die Verneinung bezieht, sollte durch Betonung hervorgehoben werden (z.B. in *Maria ist nicht mit dem Auto nach Hamburg gefahren.*). Welche das ist, kann nur durch pragmatisches Wissen berechnet werden (Dutoit, *An Introduction to Text-To-Speech Synthesis* S.146). Die meisten derzeitigen TTS-Systeme beschränken sich darauf, die Prosodie zu benutzen, um Äußerungen in Teile zu segmentieren, damit sie für den Hörer verständlicher werden (Dutoit, *An Introduction to Text-To-Speech Synthesis* S.148). Dadurch wird die prosodische Struktur

etwas flacher als in natürlicher Sprache. Semantisches und pragmatisches Wissen steht jedoch nur wenigen TTS-Systemen zur Verfügung.

Die Daten aus dem Sprachverarbeitungsmodul werden an das Signalverarbeitungsmodul übergeben. Hier passiert die eigentliche Synthese, bei der ein Audiosignal generiert wird. Bei der konkatenativen Synthese erfolgt hier Einheitenauswahl und -verkettung. Dazu gehören z.B. Diphonsynthese und Unit-Selection. Für die Diphonsynthese werden aus einer Datenbank mit Diphonen - das sind Einheiten, die in der Mitte eines Lautes beginnen und in der Mitte des darauffolgenden Lautes enden - die passendsten Kandidaten (falls mehrere geeignete Kandidaten vorhanden sind) ausgewählt und aneinandergelinkt. Unit-Selection zeichnet sich dadurch aus, dass ganze Sätze in einer Datenbank gespeichert werden, aus denen möglichst lange zusammenhängende Stücke ausgewählt und verkettet werden. Außer der konkatenativen Synthese gibt es die regelbasierte Synthese, die, obwohl sie mittlerweile unüblich ist, der Vollständigkeit halber erwähnt werden soll. Die regelbasierte Synthese basiert auf dem Quelle-Filter-Modell. Durch mathematische Regeln werden Formanten und artikulatorische Parameter modelliert. Die Vorteile von regelbasierter Synthese im Vergleich zu konkatenativer Synthese waren bis vor einiger Zeit, dass sie zum einen weit weniger Speicherplatz benötigt und außerdem viel glatter ist als konkatenative Synthese. Da Speicherplatz inzwischen viel günstiger geworden ist und sich die Einheitenverkettung bei konkatenativer Synthese verbessert hat, sind die Argumente, die für regelbasierte Synthese sprachen, nicht mehr relevant (Sproat, *Multilingual Text-To-Speech Synthesis* S.194).

## 1.2 Was sind Non-Standard-Words?

Non-Standard-Words (NSWs) sind Token, die vor der Synthese in eine entsprechende orthographische Form überführt (d.h. expandiert) werden müssen, wie z.B. Zahlen, Abkürzungen und Sonderzeichen. Tabelle 1 enthält eine Einteilung von NSWs. Hierbei kann die gleiche Klasse von NSWs unterschiedliche Muster aufweisen, die in der Textvorverarbeitungskomponente getrennt abgefangen werden müssen. Es kommt auch vor, dass unterschiedliche Abkürzungsklassen das gleiche Muster aufweisen, was dann, wenn eine Unterscheidung nötig ist, eventuell durch Eigenschaften der Nachbar-Token aufgelöst werden kann. Ein Beispiel dafür ist die rationale Zahl *10,99*, die, wenn sie alleine erscheint, zu *zehn komma neun neun* expandiert wird. Folgt allerdings darauf das Wort

*Euro*, wird diese Kombination zu *zehn Euro neunundneunzig* expandiert.

*Akronyme* sind Abkürzungen, die aus den Anfangsbuchstaben mehrerer Wörter zusammengesetzt sind. Sie werden in der Literatur unterschiedlich definiert: Manchmal werden sie nicht von Abkürzungen unterschieden (Bußmann, *Lexikon der Sprachwissenschaft* S.42), in anderen Quellen müssen sie als Wort gesprochen werden können (z.B. bei adlexikon) und dürfen nicht mit Punkt abgekürzt sein. In dieser Arbeit wird die Definition aus Glück übernommen, nach der Akronyme nicht als Wort gesprochen werden können müssen:

“Akronym (auch Initialwort): Aus den Anfangsbuchstaben oder -silben einer Wortgruppe oder eines Kompositums gebildete Abkürzung, die als Wort verwendet wird, z.B. AT 'Altes Testament', SPD 'Sozialdemokratische Partei Deutschlands', Fewa 'Feinwaschmittel', AKüSpra 'Abkürzungssprache’ (Glück, *Metzler Lexikon Sprache* S.21).

Daher gibt es in der Klasse Akronyme zwei Gruppen: Die Abkürzungen der ersten Gruppe werden buchstabiert und für gewöhnlich nicht expandiert. Sie stellen die Mehrzahl der Akronyme dar. Die zweite Gruppe sollte als Wort gesprochen werden. Die beiden Gruppen können von einem TTS-System nicht unterschieden werden, da sie sich in ihrer Schreibweise nicht unterscheiden, z.B. *ARD* und *ESA*. Diese Aufgabe kann meist nur gelöst werden, indem diejenigen Abkürzungen, die als Wort gesprochen werden sollen, in eine Abkürzungstabelle oder in das Lexikon eingetragen werden.

*Kurzwörter* sind Wortteile, wie z.B. *Foto* für *Fotographie*, *Auto* für *Automobil*, *Trockner* für *Wäschetrockner* oder *Motel* für *Motorhotel* (aus Glück, *Metzler Lexikon Sprache* S.349). Im Gegensatz zu Akronymen sind sie meist aus der mündlichen Sprache entstanden und so alltäglich, dass sie nicht mehr ohne weiteres als Abkürzungen erkannt werden und auch in der Sprachsynthese nicht gesondert behandelt werden müssen.

Zur Klasse *sonstige Abkürzungen* gehören Abkürzungen, die mit einem Punkt enden. Auch hiervon gibt es zwei Gruppen: Die erste Gruppe kann aufgrund ihrer Eindeutigkeit mithilfe einer Abkürzungstabelle expandiert werden. Leider gibt es nur wenige Regelmäßigkeiten, nach denen Algorithmen implementiert werden können. In die zweite Gruppe gehören diejenigen Abkürzungen, die nicht eindeutig sind, weil sie fachspezifisch sind und je nach Kontext eine unterschiedliche Bedeutung haben, oder Namen darstellen, und deshalb nicht oder nur mit großem Aufwand expandiert werden können. Das Beispiel aus Tabelle 1 *J.S. Bach* stellt zwei Token dar: *J.S.* und *Bach*. *J.S.* kann viele Bedeutungen besitzen und wird im Allgemeinen nur im Zusammenhang mit dem Nachnamen

Art	Beispiele
Akronyme	ARD; GmbH; CD; USA; ESA; UFO; DAX
Kurzwörter	Auto; Foto; Motel
sonstige Abkürzungen	z.B.; u.a.; evtl.; Nr.; S.; J.S. Bach, Wohns.-Nr.
Maßeinheiten	km; cm; l; MHz; km/h; T/min
URL	www.ims.uni-stuttgart.de; ftp.ims.unterlagen.de; http://www.ims.uni-stuttgart.de
eMail-Adressen	studio@swr3.de
Zahlen und Buchstaben gemischt	S2000; 60er
Kardinalzahlen	12; 10.000; 100 000 000
Ordinalzahlen	am 7. Mai; der 7. Mai
römische Zahlen	Kapitel II; Henry V; Rocky IV
rationale Zahlen	0,3; 1/2
Aufzählung	1.; 2); (3)
Telefon- / Faxnummern	0711/9255443; 0711-9255443; (040) 41132511
Postfach / PLZ	Postfach 7111; 70569 Stuttgart
BLZ / Kontonummern	35060190; 6894551
Zeitangaben	8.45 Uhr; 8:45 h; 13 h 30; 2h; 15 min; 5s; 5 sek; 03:15:02
Datum	19.04.04; 19.04.'04; 19.4.2004; 19-4-04; vom 21. bis 25. Mai; 21. - 25. Mai
Jahreszahlen	04; '04; 2004
Geld	50 Euro; 10,11 Euro; 10 c
Verhältniszahlen	2:1; 04:03
Sonderzeichen	\$ + :- ) 75% ~

*Tabelle1: Einteilung NSWs (nach Sproat et al. (2001))*

*Bach* zu *Johann Sebastian* expandiert. Es wäre ein großer Aufwand, *J.S.* mit verschiedenen Expansionsvarianten in eine Tabelle einzutragen, um je nach nachfolgendem Token unterschiedlich zu expandieren. Es ist unmöglich, alle Namen von Komponisten, Schriftstellern, Sängern u.s.w. in Tabellen einzutragen, wenn aber nur einige eingetragen würden - wo müsste die Grenze gezogen werden?

Auch für die Expansion von *Maßeinheiten* werden Tabellen benötigt. Allerdings existieren hier Muster, nach denen Maßeinheiten gebildet werden. Die Abkürzungen der Wörter *Millimeter*, *Zentimeter* und *Kilometer* werden durch die Kombination des jeweiligen Anfangsbuchstabens und einem *m*, das für *Meter* steht, gebildet. In den Abkürzungen *Kilogramm*, *Kilometer* und *Kilojoule* wird *Kilo* stets durch *k*, das zweite Teilwort wiederum durch seinen Anfangsbuchstaben angegeben. Eine zusätzliche Schwierigkeit besteht darin, dass Maßeinheiten manchmal direkt hinter die vorangehende Zahl geschrieben wird, das Leerzeichen dazwischen also fehlt. Dann muss das Token vor der weiteren Verarbeitung in Zahl und Maßeinheit getrennt werden.

Auch *URLs* sind für ein TTS-System nicht einfach zu verarbeiten. Zwar können Punkte, Doppelpunkt, Bindestriche, '/' und '\' als Trennzeichen verwendet werden, an denen die URL auseinander genommen werden kann, die Behandlung der Teile dazwischen wie z.B. *ims* oder *uni* ist jedoch eine Schwierigkeit für sich. Es ist für das System unmöglich zu erkennen, ob es sich um Wörter oder Abkürzungen handelt. Die Tatsache, dass in Internetadressen zwischen Groß- und Kleinschreibung nicht unterschieden wird, und daher meist alles klein geschrieben wird, erschwert die Entscheidung noch zusätzlich, ob ein Wort oder eine Abkürzung gesprochen oder buchstabiert werden soll.

*eMail-Adressen* beinhalten häufig Namen, bzw. Teile von Namen, die von einem TTS-System nicht aufgelöst werden können. Für eine eMail-Adresse wie z.B. *muellermn@ims.uni-stuttgart.de* würde ein menschlicher Leser den Namen *mueller* abtrennen und als Wort aussprechen, *m* und *n* dagegen buchstabieren. Diese Fähigkeit steht einem TTS-System nicht zur Verfügung.

*Zahlen und Buchstaben gemischt* wie z.B. *S2000* oder *60er* müssen vor der Verarbeitung aufgesplittet und getrennt behandelt werden.

*Kardinalzahlen* können mit einem Algorithmus expandiert werden. Die einzige Schwierigkeit stellt die Expansion der Zahl '1' dar. Wenn sie die Funktion eines Zahl-Adjektivs besitzt, sind für eine korrekte Expansion Informationen über den entsprechenden Genus

und Kasus nötig, da sie mit dem zugehörigen Nomen kongruieren muss: z.B. (*eine Katze* vs. *ein Hund* oder *einen Hund*).

Um *Ordinalzahlen* korrekt aussprechen zu können, wird die dementsprechende Genus-, Kasus- und Numerusinformation benötigt, wie an den folgenden Beispielen deutlich wird: *am siebten Mai* vs. *der siebte Mai* und *Ich erkenne den siebten Mann von rechts und die siebte Frau von links wieder*.

*Römische Zahlen* können sowohl die Funktion von Kardinalzahlen als auch die von Ordinalzahlen übernehmen.

*Rationale Zahlen* können ohne Probleme mit einem Algorithmus expandiert werden.

*Aufzählungen* haben, wie aus Tabelle 1 ersichtlich, mehrere Notationsmöglichkeiten. Sie werden wie Ordinalzahlen ausgesprochen, ihre Endung lautet stets *-ens*, (z.B.: *erstens*, *zweitens*, *drittens*...). Sie kommen nur am Beginn eines Abschnittes vor.

Ziffern aus *Telefon- / Faxnummern* werden manchmal in Blöcke gruppiert und manchmal einzeln ausgesprochen. Deutsche Telefonnummern beginnen - wenn sie mit Vorwahl notiert sind - mit einer Null. Ein Bindestrich, Schrägstrich oder Klammern trennt die Vorwahl von der Rufnummer. Es ist nicht einfach zu entscheiden, ob die Ziffern gruppiert werden sollen, und wenn sie gruppiert werden sollen, wie.

Auch bei *Postfach* und *Postleitzahl* werden Ziffern manchmal in Blöcke gruppiert. Dann stellt sich auch hier die Frage, wie sie gruppiert werden sollen.

*Bankleitzahlen* und *Kontonummern* müssen genauso wie Postfächer und Postleitzahlen ziffernweise gesprochen oder sinnvoll gruppiert werden.

In *Zeitangaben* können Stunden und Minuten entweder durch Punkt oder Doppelpunkt getrennt werden. 8.45 Uhr wird zu *acht Uhr fünfundvierzig* expandiert, d.h. das Wort *Uhr* wird zwischen der Stunden- und der Minutenangabe gesprochen. Wie bei den Maßeinheiten kann es auch hier sein, dass die Einheit von der Zahl nicht durch ein Leerzeichen getrennt ist. Dann muss das Token wieder aufgespalten und getrennt verarbeitet werden.

Wie aus Tabelle 1 ersichtlich, existieren auch für *Datumsangaben* viele unterschiedliche Schreibweisen, die vom TTS-System getrennt abgefangen und verarbeitet werden müssen.

Auch *Jahreszahlen* können unterschiedlich geschrieben werden. Vor allem ist aber wich-

tig, dass bestimmte Jahreszahlen anders ausgesprochen werden als gewöhnliche Kardinalzahlen. Das betrifft die Zahlen von 1100 bis 1999: die Zahl 1997 z.B. wird *neunzehnhundertsiebenundneunzig* ausgesprochen wenn sie eine Jahreszahl darstellt, als Kardinalzahl ausgesprochen *eintausendneuhundertsiebenundneunzig*. Schwierigkeiten entstehen jedoch auch bei anderen Jahreszahlen. Dies betrifft nicht die Aussprache der Zahlen, wenn aber z.B. 2002 - 2004 bei der Synthese nicht als Jahreszahlen erkannt werden, kann es sein, dass *zweitausendzwei minus zweitausendvier* anstatt *zweitausendzwei bis zweitausendvier* synthetisiert wird, da es sich genauso um eine Rechnung handeln könnte.

Genau wie bei Zeitangaben muss bei *Geldangaben* die Währung anstatt des Punktes eingefügt und hinten gelöscht werden. Auch hier kann es sein, dass Betrag und Einheit nicht durch ein Leerzeichen getrennt sind. Dann wird genauso wie bei Zeitangaben verfahren.

Bei *Verhältniszahlen* wird der Doppelpunkt zu 'zu' expandiert. Um sie von einer Teilung (z.B.  $4:2 = 2$ ) unterscheiden zu können muss der Kontext beachtet werden.

Für in einem Text vorkommende *Sonderzeichen* muss entschieden werden, ob sie gesprochen werden sollen oder nicht. Wenn ja, werden sie in einer Tabelle nachgeschlagen und expandiert. Der Bindestrich in *21. - 25. Mai* muss je nach Kontext entweder zu *bis* oder *bis zum* expandiert werden und wird in anderen Kontexten häufig nicht gesprochen. Bei *Prozentzahlen* müssen das Prozentzeichen und eventuelle Anhänge (wie z.B. bei *25%ige Chance*) von der Zahl getrennt werden. Alles muss getrennt in eine orthographische Form gebracht und anschließend wieder zusammengefügt werden.

### 1.3 Wie funktioniert Festival?

Die Äußerungsstruktur in Festival besteht aus Items, die zu einer oder mehreren Relationen gehören können. Items können Objekte wie z.B. Wörter, Laute oder Silben darstellen (Black et al., *The Festival Speech Synthesis System*. S.64). Relationen sind Listen oder Bäume, die Items enthalten. Die Äußerungsstruktur (Black et al., *The Festival Speech Synthesis System*. S.63) wird aufgebaut, indem verschiedene Module durchlaufen werden, die der Äußerungsstruktur nach und nach neue Informationen hinzufügen. Welches Modul durchlaufen wird, hängt von der Wahl der Stimme und des Äußerungstyps ab (Black et al., *The Festival Speech Synthesis System*. S.65). In IMS-Festival ruft der Äußerungstyp Text zur Zeit folgende Module der Reihe nach auf: Initialize, Text, Token-

Pos, Token, POS, Phrasify, Word, Pauses, Intonation, PostLex, Duration, Int-Targets, Wave-Synth.

Bei Aufruf des Moduls *Initialize* werden alle bisher bestehenden Relationen gelöscht und alle nötigen Relationen aus der Eingabe geladen.

Im Modul *Text* wird die Tokenisierung vorgenommen. In Festival sind Token diejenigen Zeichen, die durch Leerzeichen, Zeilensprünge und Tabs von anderen Zeichen getrennt sind. Bei der Tokenisierung wird Interpunktion ("" ' . , ; ! ? ()[]) vor und nach einem Token abgetrennt und als Feature des Tokens gespeichert.

Das Modul *Token-Pos* ordnet jedem Token einen Typ zu (Tokentyperkennung).

Im Modul *Token* geschieht die Token-Wort-Konvertierung. In IMS-Festival wird zuerst abgefragt, ob das Token im Lexikon steht, denn dann muss es die Token-Wort-Konvertierung nicht durchlaufen. Diese Abfrage ist auch für Tokenteile, die nach einer ersten Verarbeitung (z.B. Aufsplittung von Kombinierten Abkürzungen oder gemischten Token) noch einmal die Token-Wort-Konvertierung durchlaufen, wichtig. Bei der Token-Wort-Konvertierung wird die Expansion von Kombinierten Abkürzungen, eMail-Adressen, Geldbeträgen, Verhältniszahlen, Rechnungen (+ - \* :), Telefonnummern, gemischten Token, römischen Zahlen, Sonderzeichen, Maßeinheiten, Abkürzungen, Jahreszahlen, Datumsangaben, Kardinalzahlen, Ordinalzahlen und rationalen Zahlen vorgenommen.

Im Modul *POS* wird der Part-Of-Speech-Tagger eingesetzt, hier bekommt jedes Wort ein Part-Of-Speech-Tag zugewiesen.

Das Modul *Phrasify* bestimmt die Phrasengrenzen (Black et al., *The Festival Speech Synthesis System*. S.81).

Im Modul *Word* findet der Lexikonnachschlag statt, hier werden die Relationen *SylStructure*, *Segment* und *Syllable* erzeugt. Die Lexikon-Einheit in Festival besteht aus drei Teilen (Black et al., *The Festival Speech Synthesis System*. S.49): Beim Lexikon-Lookup wird zuerst in einem kleinen Lexikon mit von Hand ergänzten Wörtern nachgeschlagen, wenn der Eintrag dort nicht gefunden wird, wird in einem großen kompilierten Vollformlexikon mit mehreren tausend Wörtern nachgeschlagen. Falls auch hier kein Eintrag existiert, werden Ausspracheregeln auf das zu sprechende Wort angewendet (Black et al., *The Festival Speech Synthesis System*. S.52). Ein Lexikoneintrag besteht aus einem Token, einer Part-Of-Speech-Kategorie und der zugehörigen Aussprache. Die Ausspra-

che beschreibt außer den Lauten auch Silbenstruktur und -betonung. Es können mehrere gleiche Token mit unterschiedlichem POS im Lexikon stehen. Falls sowohl das gesuchte Wort, als auch die POS-Kategorie auf einen Eintrag im Lexikon passen, wird dieser Eintrag verwendet, ansonsten wird der erste Eintrag ausgewählt, der mit dem gesuchten Wort übereinstimmt.

Im Modul *Pauses* werden an den durch Phrasify bestimmten Phrasengrenzen Pausen eingefügt.

Im Modul *Intonation* geschieht die Intonationsgenerierung (Black et al., *The Festival Speech Synthesis System*. S.83).

Im Modul *PostLex* werden die postlexikalischen Regeln angewendet. Durch sie wird die Koartikulation modelliert, welche die Natürlichkeit der Synthese steigert (Black et al., *The Festival Speech Synthesis System*. S.61).

Im Modul *Duration* wird die Lautdauer bestimmt (Black et al., *The Festival Speech Synthesis System*. S.87). Auch hier gibt es verschiedene vordefinierte Methoden.

Das Modul *IntTargets* bestimmt durch die Labels des Moduls Intonation die Grundfrequenzkontur.

Im Modul *Wave-Synth* wird das Sprachsignal generiert. Mit der Wahl der Stimme wird der Synthese-Typ gewählt, je nach Stimme ist das Unit-Selection oder Diphonsynthese (Black et al., *The Festival Speech Synthesis System*. S.65/66).

# Kapitel 2

## Vorgehen bei der Suche nach Belegen in Korpora

Um feststellen zu können, welche Klassen von NSWs bereits jetzt korrekt von IMS-Festival behandelt werden und wo noch Schwierigkeiten bei der Token-Wort-Konvertierung liegen, wurden Daten aus verschiedenen Korpora extrahiert. Dabei wurde darauf geachtet, dass die Daten möglichst verschiedene Bereiche abdecken, um eine Vielzahl von NSWs zu erhalten. Für die erhaltenen Token wurde die jeweilige orthographische Form bestimmt und überprüft.

### 2.1 Abfragen

Mit den in Tabelle 2 aufgelisteten Suchmustern wurde in den unten beschriebenen Korpora nach NSWs gesucht. Je nachdem, ob der Kontext für die Token-Wort-Konvertierung wichtig ist, wurden einzelne Token oder ganze Sätze extrahiert.

Art	Beispiele
Akronyme aus 2 Großbuchstaben:	"^[A-Z]{2}\$";
Akronyme aus 3 Großbuchstaben:	"^[A-Z]{3}\$";
Akronyme aus 4 Großbuchstaben:	"^[A-Z]{4}\$";
Akronyme aus mindestens 5 Großbuchstaben:	"^[A-Z]{5,}\$";
Akronyme aus Kleinbuchstaben:	"^[a-z]{3}\$";
Wörter, die 2 Großbuchstaben enthalten:	"[a-z]*[A-Z][a-z]*[A-Z][a-z]+"; "[a-z]*[A-Z][a-z]+[A-Z][a-z]*"; "[a-z]+[A-Z][a-z]*[A-Z][a-z]*";
Wörter, die 3 Großbuchstaben enthalten:	"[a-z]*[A-Z][a-z]*[A-Z][a-z]*[A-Z][a-z]+"; "[a-z]*[A-Z][a-z]*[A-Z][a-z]+[A-Z][a-z]*"; "[a-z]*[A-Z][a-z]+[A-Z][a-z]*[A-Z][a-z]*"; "[a-z]+[A-Z][a-z]*[A-Z][a-z]*[A-Z][a-z]*";
Abkürzungen mit Punkt:	"[a-zA-Z]+\.";
einzelne Kleinbuchstaben mit Punkten:	"([a-z]\.){2}"; "([a-z]\.){3}";
einzelne Buchstaben mit Punkten:	"([a-zA-Z]\.){2}"; "([a-zA-Z]\.){3}";
Kleinbuchstaben mit Punkt:	"[a-z]+\.";
Maßeinheiten:	"[a-z]{1,5}\/[a-z]{1,5}";
Internetadressen:	"www"
eMail:	".@.";
Zahlen und Buchstaben gemischt:	"[a-z]*[0-9]+[a-z]+[0-9]*[a-z]*"; "[a-z]*[0-9]*[a-z]+[0-9]+[a-z]*";
Kardinalzahl:	"[0-9]{1,3}(\.[0-9]{3})+";
Ordinalzahlen:	"[0-9]{1,2}\."; "[0-9]{3,4}\."; "[0-9]{5,}\.";
rationale Zahlen:	"[0-9]{1,4}\,[0-9]{1}"; "[0-9]{1,4}\,[0-9]{3,4}";
Brüche:	"[0-9]{1,4}\/[0-9]{1,4}";
Telefon:	"Telefonnummer" "[0-9].+"; "[0-9]{3,8}\-[0-9]{5,7}";
Postfach:	"Postfach" "[0-9]{3,5}";
Postleitzahlen:	"[0-9]{5}" "[A-Z][a-z]{2,}";
Uhrzeit:	".+[0-9]" "Uhr"; ".+[0-9]" "h";
Datum:	"am" "[0-9].+"; "den" "[0-9]{2}\.[0-9]{2}\."; "vom" "[0-9].+";
Jahreszahl:	"[0-9]{4}";
Geldbeträge:	"[0-9]{3,5}\,[0-9]{2}";
Prozent:	"%";
Verhältnis:	"[0-9]+:[0-9]+";
Sonderzeichen:	"\&"; "\\$";...

Tabelle2: Muster für die Extraktion von Daten aus den Korpora

## 2.2 Korpora

### 2.2.1 Deutsche Abkürzungen

Dieses Korpus wurde ausgewählt, da es nur Abkürzungen mit Punkt enthält. Die Anzahl der enthaltenen Token liegt bei 5778. Das Korpus wurde 1994 erstellt. Mit der Abfrage für Abkürzungen mit Punkt erhält man alle 5778 Token.

### 2.2.2 European Languages News Corpus

Es wurde der deutsche Teil des Korpus verwendet. Die Anzahl der enthaltenen Token ist sehr hoch und liegt bei ca. 104 Millionen. Es wurde 1997 erstellt und ist damit noch relativ neu. Im Folgenden ist die Anzahl der jeweiligen durch die oben beschriebenen Abfragen erhaltenen Token und Sätze aufgelistet.

Akronyme: 41458 Token

Abkürzungen mit Punkt: 473 Token

Wörter, die Großbuchstaben enthalten: 2908 Token

Zahlen und Buchstaben gemischt: 9061 Token

Internetadressen: 57 Token

Jahreszahlen: 1279629

einzelne Buchstaben mit Punkten: 606 Token

Brüche: 356 Sätze

Datum: 1509 Sätze

Geld: 191 Sätze

Kardinalzahlen: 713 Sätze

kleine Akronyme: 1197 Token

Maßeinheiten: 5378 Token

Ordinalzahlen: 629 Sätze

PLZ: 68 Sätze

Postfach: 8 Sätze

Prozent: 122 Sätze

rationale Zahlen: 1303 Sätze

Sonderzeichen: 224 Sätze

Telefonnummern: 54 Sätze  
Uhrzeit: 130 Sätze  
Verhältniszahlen: 803 Sätze

### **2.2.3 IMS Corpus Workbench Demo Corpus**

Dieses Korpus wurde ausgewählt, um NSWs aus dem Bereich Recht zu erhalten. Es besteht aus frei verfügbaren Texten. Es wurde der deutsche Teil des Korpus verwendet, der ca. 816000 Token enthält. Das Jahr der Erstellung konnte nicht festgestellt werden. Aus diesem Korpus wurde die folgende Anzahl Token extrahiert:

Akronyme: 36 Token  
Abkürzungen mit Punkt: 30 Token  
Wörter, die Großbuchstaben enthalten: 24 Token

### **2.2.4 Handelsblatt**

Auch dieses Korpus ist sehr groß. Es enthält ca. 36 Millionen Token aus verschiedenen Bereichen. Es wurde 1986-1988 erstellt.

Akronyme: 22191 Token  
Abkürzungen mit Punkt: 841 Token  
Wörter, die Großbuchstaben enthalten: 3397 Token  
Zahlen und Buchstaben gemischt: 6118 Token

### **2.2.5 Computerzeitung**

Dieses Korpus wurde ausgewählt, um NSWs aus dem Bereich Computer bzw. Internet-typische Ausdrücke zu erhalten. Es enthält ca. 2 Millionen Token und wurde 1993/1994 erstellt.

Akronyme: 1615 Token  
Abkürzungen mit Punkt: 1097 Token  
Wörter, die Großbuchstaben enthalten: 488 Token

Zahlen und Buchstaben gemischt: 248 Token  
eMail-Adressen: 17 Token

## **2.2.6 Internetadressen**

Um eine große Menge an Internetadressen zu erhalten, wurden 1935 Adressen aus dem lokalen Netz extrahiert.

# Kapitel 3

## Auswertung: Beschreibung der Problemklassen und Lösungen

Durch die Auswertung der gesammelten Daten ergaben sich die in den folgenden Abschnitten beschriebenen Auffälligkeiten.

### 3.1 Akronyme

Während der Auswertung fielen verschiedene Arten von Akronymen auf: Akronyme die nur aus Großbuchstaben bestehen (z.B. *UEFA*), Akronyme die nur aus Kleinbuchstaben bestehen (wie z.B. *dpa*) und solche mit gemischter Schreibweise (z.B. *AdW*).

#### 3.1.1 Akronyme aus 2, 3 oder 4 Großbuchstaben

##### Situationsbeschreibung und Probleme

Die Auswertung ergab, dass die meisten Akronyme, die aus 2, 3 oder 4 Großbuchstaben bestehen, tatsächlich Akronyme, also keine regulären Wörter, sind. In der Regel müssen Akronyme nicht expandiert werden, da sie genauso gut oder sogar besser verstanden werden als ihre expandierte Form, z.B. *AIDS*, *DAX*, *SPD*, *FIFA*, *NASA*. Eine Ausnahme bildet unter anderem das Akronym *SS*, das vor allem in Studentenkreisen schnell

mit dem Wort *Sommersemester* verknüpft und in dieser Bedeutung nie als *SS* ausgesprochen wird. Leider ist dieses Akronym aus der Zeit von vor 1945 negativ vorbelastet und hat auch sonst mehrere Bedeutungen. Bei der Auswertung trat die Expansion von *SS-Massenhinrichtung* zu *Sommersemester Massenhinrichtung* und von *SS-Offizier* zu *Sommersemester Offizier* auf. Dieses Beispiel zeigt, dass man sehr genau prüfen muss, wieviele Bedeutungen ein Akronym besitzt, bevor man es expandiert. Bisher ist es so, dass Akronyme, die nur aus Konsonanten bestehen, buchstabiert werden, ansonsten werden sie als Wort gesprochen, außer sie sind im Lexikon oder einer Abkürzungstabelle eingetragen.

## **Lösung**

Da die größere Gruppe der Akronyme buchstabiert werden sollte, werden nun nur noch diejenigen als Wort gesprochen, die in der Abkürzungstabelle eingetragen sind. Expandiert werden nur solche, für die im DUDEN (Wörterbuch der Abkürzungen) nur eine Bedeutung eingetragen ist, bzw. deren Bedeutung durch die Betrachtung des Kontexts auf eine bestimmte eingeschränkt werden kann. Für das Beispiel *SS* bedeutet das, dass es nur noch zu *Sommersemester* expandiert wird, wenn darauf eine Jahreszahl folgt und außerdem eines der Wörter *das*, *im*, *nächste*, *nächstes* oder *nächsten* vorangeht.

### **3.1.2 Akronyme aus mindestens 5 Großbuchstaben / Wörter, die mehrere Großbuchstaben enthalten / Akronyme aus Kleinbuchstaben**

#### **Situationsbeschreibung und Probleme**

Diese Abkürzungen gehören zwar von ihrer Schreibweise ausgehend nicht in dieselbe Klasse und sind außerdem zum Großteil keine Akronyme. Bisher wurden sie jedoch wie Akronyme behandelt und außer den Abkürzungen die nur aus Kleinbuchstaben bestehen, wurden sie komplett buchstabiert. Dadurch war ihr Sinn nicht mehr gut verständlich, obwohl sie meist Wörter des Deutschen darstellen (z.B. *FRIEDENSPROZESS*, *ORIGINALDREHBUCH*, *StadtExpress*, *TrabiClubs*, *InformatikerInnen*, *BUERobedarf*, *EDV-maessigen*). Akronyme aus Kleinbuchstaben wurden prinzipiell als Wort gesprochen. Ein

Akronym wie z.B. *dpa* oder *mbar* ist im Deutschen aber nicht als Wort aussprechbar, da *d* und *p* und auch *m* und *b* nicht in einer Silbe aufeinander folgen können. Bisher fehlte eine Überprüfung, ob es sich um ein Akronym handelt.

## **Lösung**

Da Akronyme aus fünf und mehr Buchstaben und Wörter, die mehrere Großbuchstaben enthalten, zum Großteil reguläre deutsche Wörter darstellen wurde anhand von Daten aus den oben genannten Korpora ein Silbenmodell des Deutschen erstellt. Die Silbenstruktur dieser Wörter wird im Lexikon nachgeschlagen und mithilfe des Silbenmodells auf Aussprechbarkeit überprüft. Falls sich herausstellt, dass eine der Silben nicht aussprechbar ist wird das Wort buchstabiert, ansonsten ausgesprochen. Akronyme wie z.B. *dpa*, die nur aus Kleinbuchstaben bestehen, wurden bisher grundsätzlich ausgesprochen. Mit dem Silbenmodell werden auch sie auf Aussprechbarkeit überprüft. Da im Deutschen weder *d* und *p*, noch *m* und *b* im Onset einer Silbe aufeinander folgen können, wird sowohl *dpa* als auch *mbar* buchstabiert.

## **3.2 sonstige Abkürzungen**

### **3.2.1 Abkürzungen mit Punkt**

#### **Situationsbeschreibung und Probleme**

Nur wenige Abkürzungen mit Punkt werden expandiert, da bisher nur diejenigen Abkürzungen expandiert werden, die mit ihrer expandierten Form in eine Tabelle eingetragen sind. Bei der Auswertung sind jedoch einige Regelmässigkeiten bei der Expansion von Nomina aufgefallen:

Im Deutschen gibt es viele Nomina, die mit *-ung* enden. Sie werden oft so abgekürzt, dass Stamm und 'Anhänge' voll ausgeschrieben werden, das Postfix *-ung* jedoch nur durch *g*. angedeutet wird, wie z.B. bei *Forderg.*, *Geldforderg.*, *Zahlungsaufforderg.* Der menschliche Leser erkennt sofort, was gemeint ist, und expandiert dementsprechend zu *Forderung*, *Geldforderung* und *Zahlungsaufforderung*.

Eine andere Gruppe Nomina endet mit *-keit*. Für diejenigen, die mit *-möglk.*, *-lichk.* und *-bark.* abgekürzt werden, besteht die Möglichkeit sie zu expandieren. Auch hier ist es für den Menschen einfach z.B. *Verständlichk.* zu *Verständlichkeit*, *Beschäftigungsmöglk.* zu *Beschäftigungsmöglichkeit* und *Verwendbark.* zu *Verwendbarkeit* zu expandieren.

Auch Straßennamen, die häufig mit *-str.* abgekürzt werden, sind für den Menschen einfach zu expandieren. Er erkennt, dass es sich um einen Straßennamen handelt und expandiert *-str.* zu *-straße*. Für ein Sprachsynthesystem ist es allerdings nicht ganz einfach, diese Wörter zu expandieren. Bisher werden keine Abkürzungen nach einem bestimmten Muster expandiert, sondern nur diejenigen Abkürzungen, die in einer Abkürzungstabelle erfasst sind. Die Anzahl der oben genannten Nomina ist aufgrund ihrer Produktivität unendlich. Da Tabellen nur eine endliche Zahl von Wörtern enthalten können, können diese Nomen nicht alle in Tabellen geschrieben werden.

Durch die Auswertung wurden außerdem einige weitere Abkürzungen gefunden, die eindeutig sind und daher mithilfe der Abkürzungstabelle expandiert werden können, z.B.: *fortl.* (*fortlaufend*), *insbes.* (*insbesondere*) oder *oblig.* (*obligatorisch*), und auch solche, die expandiert werden, aber nicht eindeutig sind, z.B. heißt *Ges.* nicht nur *Gesellschaft*, sondern kann unter anderem auch *Gesamtheit*, *Gesandter* oder *Gesang* bedeuten. Abgekürzte Wörter wie z.B. *zuzgl.* oder *erf* werden weder expandiert noch buchstabiert sondern als Wort gesprochen.

## Lösung

Die oben beschriebenen Nomina, die auf *-ung* enden und deren Endung nur durch *g.* angedeutet wird, wie z.B. bei *Forderg.*, *Geldforderg.* oder *Zahlungsaufforderg.* können expandiert werden. Dafür wird der Anfang abgeschnitten und in einer Variablen gespeichert. Der abgekürzte Stamm des Wortes wird in einer Tabelle nachgeschlagen. Zurückgeliefert wird seine Übersetzung, die mit dem zu Beginn abgeschnittenen und gespeicherten Anfang zusammengesetzt wird.

Betrachten wir diejenige Gruppe von Nomina, die auf *-keit* endet. Für diejenigen, die mit *-möglk.*, *-lichk.* und *-bark.* abgekürzt werden, besteht die Möglichkeit sie zu expandieren. Auch hier wird der Anfang abgeschnitten und gespeichert, *-möglk.* wird zu *-möglichkeit*, *-lichk.* zu *-lichkeit* und *-bark.* zu *-barkeit* expandiert. Dann wird die jeweilige expandierte Endung mit dem Anfang zusammengefügt.

Straßennamen werden häufig mit *-str.* abgekürzt. Da *-str.* aber auch z.B. zu *Strecke* oder *String* expandiert werden kann, wird zuerst überprüft, ob dahinter eine Zahl steht. Wenn ja, wird genauso wie in den vorigen Fällen verfahren. Der Anfang wird abgeschnitten und gespeichert, die Endung zu *-straße* expandiert, dann werden beide Teile zusammengefügt und anschließend ausgegeben. Für alle hier erwähnten Fälle wird außerdem überprüft, ob die Abkürzung groß geschrieben wurde, da man nur dann sicher sein kann, dass es sich um ein Nomen handelt.

Die Abkürzungen, die im DUDEN (Wörterbuch der Abkürzungen) nur eine Übersetzung haben, werden wiederum in eine Abkürzungstabelle eingetragen. Abkürzungen mit mehreren Bedeutungen dürfen nicht expandiert werden. Eine Abkürzung die nicht expandiert werden darf, ist z.B. *erf.* da sie *erfahren, erfassen, erfinden, erfolgen, erforderlich, erforschen* oder *erfüllen* bedeuten kann. Abkürzungen, die nicht expandiert werden können, werden nun buchstabiert und ihr Punkt gesprochen, im Fall von *erf.* also *e r f Punkt*.

### **3.2.2 Einzelbuchstaben bzw. mehrere Wortanfänge durch Punkte getrennt**

#### **Situationsbeschreibung und Probleme**

Auch hier besteht das Problem, dass eine Abkürzung mehrere Bedeutungen besitzen kann. Zum Beispiel ist die Abkürzung *u.a.* sehr häufig und bedeutet in den allermeisten Fällen *unter anderem*, sie ist aber nicht eindeutig und kann auch *und andere* heißen.

Im Fall von abgekürzten Vornamen, bzw. nicht expandierten Abkürzungen mit Punkt, werden die Punkte, bis auf den letzten, bisher mit ausgesprochen. *J.S. Bach* wird *J Punkt S Bach* und *i.d.G.* (*in der Gegend* oder *in dem Gesetz*) wird *i Punkt d Punkt G* ausgesprochen. Die Punkte werden als Trennzeichen nicht benötigt, da man bei so kurzen Abkürzungen davon ausgehen kann, dass die einzelnen Buchstaben jeweils Wortanfänge repräsentieren.

#### **Lösung**

Die Abkürzungen, die im DUDEN (Wörterbuch der Abkürzungen) nur eine Übersetzung haben, werden wiederum in eine Abkürzungstabelle eingetragen. Die Punkte zwischen

den einzelnen Buchstaben der Abkürzungen, die nicht expandiert werden, werden nicht mehr gesprochen.

### 3.2.3 Kombinierte Abkürzungen

#### Situationsbeschreibung und Probleme

Kombinierte Abkürzungen wie z.B. 'öffentl.-rechtl.' werden zuerst aufgesplittet. Dann wird für jede Abkürzung nach einer Expansion gesucht. Falls eine Expansion gefunden wird, wird diese gesprochen, wenn für die Abkürzung aber kein Eintrag existiert, wird sie buchstabiert. Bisher werden die vorher gesplitteten Teile einfach wieder aneinander gehängt. Leider geht dabei die Information verloren, dass es sich um eine kombinierte Abkürzung handelt, die aus mehreren Abkürzungen besteht. Diese Information ist aber wichtig, wenn für mehrere aneinander hängende Abkürzungen keine Expansion vorhanden ist und sie daher buchstabiert werden. Die kombinierte Abkürzung *Anz.-Ann.* z.B. kann vom Menschen in einem bestimmten Kontext als Anzeigen-Annahme erkannt werden. Wenn jedoch die einzelnen Buchstaben *A n z A n n* direkt aneinander gehängt werden, wird es schwieriger, der kombinierten Abkürzung eine Bedeutung zuzuordnen.

#### Lösung

Damit man die einzelnen Teile kombinierter Abkürzungen, die buchstabiert werden, erkennen kann, wird nun nach jedem buchstabierten Teil ein Punkt gesprochen.

## 3.3 Maßeinheiten

#### Situationsbeschreibung und Probleme

Maßeinheiten wie z.B. *km* oder *ha* werden korrekt expandiert. Maßeinheiten wie *km/h* oder *min/km* werden bisher nicht korrekt expandiert: *50 km/h* wurde zu *fünfzig kilo Meter h* und *5 min/km* zu *fünf min kilo Meter* expandiert.

## Lösung

Ein Algorithmus, der Maßeinheiten nach diesem Muster behandelt, expandiert nun *50 km/h* zu *fünfzig kilo Meter pro Stunde* und *5 min/km* zu *fünf Minuten pro kilo Meter*.

## 3.4 Internetadressen

### Situationsbeschreibung und Probleme

Bisher werden nur Internetadressen der Form *http://www.uni-stuttgart.de* als Internetadressen erkannt. Inzwischen ist es allerdings üblich, Internetadressen nur mit *www.uni-stuttgart.de* anzugeben, da die heutigen Browser das Übertragungsprotokoll selbst ergänzen. Eine andere Art von Internetadressen sind Adressen, die mit *ftp* beginnen. Internetadressen, die mit *www* oder *ftp* beginnen, werden von IMS-Festival bisher nicht als Internetadressen erkannt. Die darin vorkommenden Wörter werden zum Grossteil buchstabiert und Sonderzeichen wie z.B. *'/'* werden ignoriert. Ein anderes Problem stellt die fehlende Unterscheidung in der Groß- und Kleinschreibung dar, z.B. werden Akronyme nicht mehr als solche erkannt und daher - falls sie aussprechbar sind - ausgesprochen. Das ist ein entscheidender Nachteil, da viele Firmennamen und damit auch viele Internetadressen Akronyme enthalten. Schwierig ist außerdem die Aussprache der Länderkennungen, die manchmal buchstabiert (z.B. *de*), manchmal gesprochen werden sollten (z.B. *com*).

### Lösung

Durch eine Veränderung der Abfrage werden inzwischen auch *ftp*- und *www*-Adressen richtig erkannt. Klein geschriebene Akronyme werden durch das Silbenmodell auf Aussprechbarkeit überprüft. Länderkennungen wie *de* und kleine Abkürzungen können nun in der Abkürzungstabelle nachgeschlagen werden.

## 3.5 eMail-Adressen

### Situationsbeschreibung und Probleme

Wie in Kapitel 1.2 am Beispiel *muellermn* beschrieben, ist die Unterscheidung von Wörtern und Buchstabenaneinanderreihung oder ihrer Kombination nicht möglich. Daher werden die Buchstaben einer eMail-Adresse als Wörter behandelt. Da bisher keine Überprüfung auf Aussprechbarkeit stattfindet, werden sie ausgesprochen, es sei denn, sie bestehen nur aus Konsonanten.

### Lösung

Auch die Wörter und Buchstabenaneinanderreihungen von eMail-Adressen werden durch das Silbenmodell auf Aussprechbarkeit überprüft.

## 3.6 Zahlen und Buchstaben gemischt

### Situationsbeschreibung und Probleme

Es gibt Abkürzungen wie z.B. *60er*, die aus Zahlen und Buchstaben bestehen. Bisher wird die Zahl in einen Buchstabenstring übersetzt und beide Buchstabenstrings getrennt voneinander ausgesprochen. Gerade für dieses Beispiel ist das nicht optimal, da */sECtsIC6/* nicht korrekt ist - es muss */sECtsIlg6/* heissen.

Token wie z.B. *1.OG*, die normalerweise 2 Token darstellen würden, da sie eigentlich durch ein Leerzeichen getrennt werden sollten, sind für Festival schwer zu verarbeiten. Bei der Synthese von z.B. *im 1.OG* durch IMS-Festival, erhält man bisher *im eins Punkt O G*. Nur bei korrekter Eingabe: *im 2.OG*, erhält man bei der Synthese *im zweiten O G*.

### Lösung

Inzwischen werden die beiden Buchstabenstrings *sechzig* und *er* durch einen String-append aneinandergefügt und als ein Wort ausgesprochen. Dadurch erscheint die Zahl nicht mehr am Wortende, sondern wird korrekt ausgesprochen. Token wie z.B. *1.OG*,

die aus einer Zahl und einer Abkürzung bestehen, werden am Punkt getrennt und beides wird getrennt verarbeitet.

## 3.7 Kardinalzahlen

### Situationsbeschreibung und Probleme

Die korrekte Aussprache der Zahl 1 erfordert Genus- und Kasus-Informationen über das nachfolgende Nomen. In IMS-Festival wird die Zahl 1 bisher immer zu *eins* expandiert, z.B. *1km* zu *eins kilometer*, *2 1/2* zu *zwei eins halbe*, *Vor 1 Jahr* zu *Vor eins Jahr* und *1h 30min* zu *eins Stunden dreißig Minuten*. Die besondere Schwierigkeit hierbei ist, dass weder Genus- noch Kasus-Informationen zur Verfügung stehen.

Bei Kardinalzahlen mit führender Null wird bisher zuerst die Null gesprochen, dann die Kardinalzahl, z.B.: *0332* wird expandiert zu *null drei hundert zwei und dreißig*. Da Kardinalzahlen normalerweise nicht mit führender Null geschrieben werden, ist diese Zahl wahrscheinlich eine Nummer und sollte ziffernweise gesprochen werden.

Kardinalzahlen, die in Dreierblöcken notiert werden, die jeweils durch ein Leerzeichen getrennt sind, werden zu einer Kardinalzahl gruppiert. Bisher wird Klammerung dabei nicht beachtet. Auch Bankleitzahlen (z.B. *550 660 20*), rationale Zahlen (z.B. *Rückstoß eines Atomkerns: 0,000 000 000 000 000 001*) und Teile von Telefonnummern (z.B. *0711/22 555 489*) sind davon betroffen. Die Gruppierung von Kardinalzahlen darf nicht stattfinden wenn eines der Token in Klammern steht: *377,50 (379 Dollar)* darf nicht expandiert werden zu *siebenunddreißig Million siebenhundertfünfzigtausenddreihundertneunundsiebzig Dollar*. Auch Teilblöcke von Bankleitzahlen und Telefonnummern dürfen nicht gruppiert werden.

### Lösung

Die korrekte Behandlung der Zahl 1 hätte den Rahmen dieser Studienarbeit gesprengt und wird daher nicht behandelt.

Kardinalzahlen mit führender Null werden durch eine Abfrage abgefangen und die Ziffern einzeln ausgesprochen.

Die Gruppierung von Dreierblöcken, die zu Bankleitzahlen, rationalen Zahlen oder Telefonnummern gehören, wird nicht mehr durchgeführt. Auch geklammerte Dreierblöcke werden nicht mehr gruppiert.

## 3.8 Ordinalzahlen

### Situationsbeschreibung und Probleme

Bei der Expansion von Ordinalzahlen besteht das Problem, dass sie in Genus, Kasus und Numerus mit dem bezeichneten Nomen übereinstimmen müssen. Bisher wird das von Festival nicht beachtet: Die Ordinalzahl in *seinen 88. Geburtstag* wird expandiert zu *achtundachtzigster*, die in *an seinem 62. Geburtstag* zu *zweiundsechzigster*, und *haben gestern abend die 47. internationalen Filmfestspiele begonnen* zu *siebenundvierzigste*.

Eine Zahl ab vier Stellen ist nur sehr selten eine Ordinalzahl, sondern meist eine Kardinalzahl am Satzende. Dies wird nicht immer erkannt, z.B. wird die Zahl in *Es war eines der härtesten Urteile gegen die Dissidenten von 1989* zu *eintausendneunhundertneunundachtzigster* expandiert, die Zahl in *ueber die telefonische Hotline 39700571* zu *neununddreißig Millionen siebenhunderttausendfünfhunderteinundsiebzigster*.

Wenn die Ordinalzahl am Satzanfang steht, ist der Kasus Nominativ. Auch für Ordinalzahlen im Nominativ werden jedoch noch Genus-Informationen benötigt, um eine Expansion korrekt durchführen zu können. Man sieht das an den folgenden beiden Beispielen: *1. Runde* wird expandiert zu *erster Runde*, *1. Bundesliga* zu *erster Bundesliga*.

### Lösung

Da bei der Token-Wort-Konvertierung keine Kasus- und Genus-Informationen zur Verfügung stehen, muss die Endung einer Ordinalzahl durch vorhergehende Artikel oder Pronomen bestimmt werden. Auch die Bearbeitung dieser Aufgabe bleibt noch offen.

Da Zahlen ab vier Stellen nur sehr selten wirklich Ordinalzahlen sind, werden nur noch 1-3stellige Zahlen als Ordinalzahlen erkannt.

## 3.9 Römische Zahlen

### Situationsbeschreibung und Probleme

Auffallend ist hier, dass bisher oft Buchstabenkombinationen als römische Zahl erkannt werden, die zwar in bestimmten Kontexten römische Zahlen sind, in diesem Zusammenhang aber Akronyme darstellen. Die Buchstabenkombination *CD* ist hier aufgefallen, da sie in den meisten Fällen eine Abkürzung für *Compact Disc* darstellt, von IMS-Festival aber zu *römisch vierhundert* expandiert wurde. Auch das Akronym *DVD* wurde während der Auswertung zu einer römischen Zahl expandiert, was jedoch weder richtig ist noch Sinn macht, da nicht 100 von 5 abgezogen werden, um sofort wieder aufaddiert zu werden. Auch im folgenden Satz wird deutlich, dass keine römische Zahl gemeint ist: das Akronym in *Der Verkehrsclub Deutschland VCD...* wurde zu *römisch fünfhundertfünfundneunzig* expandiert.

### Lösung

In der Auswertung wurde deutlich, dass römische Zahlen vor allem für Abschnittsnummierungen in Texten und Gesetzen und in Namen von Königen, Zaren und Päpsten benutzt werden. Dabei sind nur Kombinationen von V, I und X aufgetreten, womit man über den Wert '38' nicht hinauskommt. Daher wurde die Abfrage nun insofern verändert, dass jetzt nur noch Zahlen bis zum Wert 38 als römische Zahlen erkannt werden. Außerdem wird eine römische Zahl nun nicht mehr mit *römisch* angekündigt. Dies ist zwar ein geringer Informationsverlust, doch auch beim Vorlesen einer römischen Zahl wird das *römisch* häufig weglassen. In den folgenden Beispielsätzen wäre es wohl eher verwirrend als informativ: *Seine erste grosse Rolle bekam er in Rocky IV. Vor 3 Tagen begann in Havanna das IV. Zigarrenfestival.*

## 3.10 Rationale Zahlen

### 3.10.1 Kommazahlen

#### Situationsbeschreibung und Probleme

Kardinalzahlen, die zur besseren Lesbarkeit in Dreierblöcke eingeteilt sind, und deren Blöcke durch Punkte getrennt werden, werden schon bisher richtig verarbeitet. Rationale Zahlen, die in der gleichen Weise notiert sind, werden bisher nicht korrekt expandiert: *18.614,26* wird zu *eins acht Punkt sechs eins vier Komma zwei sechs* expandiert.

Nur der Block der folgenden rationalen Zahl, der das Komma enthält, wird als solche erkannt: *Rückstoß eines Atomkerns: 0,000 000 000 000 001*.

#### Lösung

Rationale Zahlen, die in Dreierblöcke eingeteilt sind, und deren Blöcke durch Punkte getrennt werden, werden abgefangen und korrekt expandiert.

Rationale Zahlen, die in Dreierblöcke eingeteilt sind, und deren Blöcke durch Leerzeichen getrennt werden, werden inzwischen komplett als rationale Zahl erkannt.

### 3.10.2 Brüche

#### Situationsbeschreibung und Probleme

Die wenigsten der durch die Auswertung untersuchten vermeintlichen Brüche waren tatsächlich Brüche. Zum Beispiel wurden Jahreszahlen wie *1991/92* als Bruch erkannt und zu *neunzehnhunderteinundneunzig zweiundneunzigstel* expandiert, *ICE 2/2* und *20/20 Initiative* wurden als Bruch gesehen, wobei es sich bei gleichen Zahlen nur unwahrscheinlich um einen Bruch handelt und in diesem Fall der Bruch eine Typbezeichnung darstellt. In *mit drei Kochmützen und 17/20 Punkten ausgezeichnet* sollte es *17 von 20 Punkten* heißen, und in *mit einem Sonderfreibetrag von 1200/2400 Euro möglich* sollte der Schrägstrich in *bzw.* expandiert werden.

## Lösung

Da die Auswertung gezeigt hat, dass die wenigsten der als Bruch erkannten Token tatsächlich Brüche sind, mussten die verschiedenen Muster genauer differenziert werden. Da Brüche mit vierstelligem Zähler und zweistelligem Nenner aufeinander folgende Jahreszahlen sein können, werden sie daraufhin überprüft, ob die letzten beiden Stellen des Zählers um eins kleiner sind als der Nenner. Falls diese Bedingung zutrifft, werden beide Zahlen als Kardinalzahlen ohne Trennzeichen gesprochen, z.B. (*neunzehn Hundert ein und neunzig zwei und neunzig*). Falls diese Bedingung nicht zutrifft, werden beide Zahlen als Kardinalzahlen und dazwischen ein Strich als Trennzeichen gesprochen, z.B.: *3562/78* wird zu *drei tausend fünf hundert zwei und sechzig strich acht und siebenzig*. Der Sonderfall *08/15* wird zu *null acht fünfzehn* expandiert. Wenn auf den Bruch das Wort *Punkten* folgt, werden Zähler und Nenner als Kardinalzahlen gesprochen und der Schrägstrich zu *von* expandiert, z.B. *neun und neunzig von ein hundert Punkten*. Wenn sowohl Zähler als auch Nenner zweistellig sind, wird wiederum für beide überprüft, ob sie aufeinanderfolgende Jahreszahlen sein können. Wenn ja, wird genau wie bei den Brüchen mit vierstelligem Zähler und zweistelligem Nenner verfahren, z.B. *neun und achtzig neunzig*. Zähler und Nenner werden auch als Kardinalzahlen gesprochen wenn sie identisch sind, z.B. *neunzig neunzig*. Wenn beide Bedingungen nicht zutreffen, werden Zähler und Nenner als Kardinalzahlen mit einem Strich als Trennzeichen gesprochen. Zähler und Nenner von einstelligen Brüchen werden wiederum daraufhin überprüft, ob sie identisch sind. Wenn ja, werden sie als Kardinalzahlen ohne Trennzeichen gesprochen. Wenn die Bedingung nicht zutrifft, werden sie als Bruch gesprochen, z.B. *drei viertel*. Nur in diesem Fall ist die Wahrscheinlichkeit sehr hoch, dass es sich tatsächlich um einen Bruch handelt. Alle anderen Brüche werden als Kardinalzahlen mit einem Strich als Trennzeichen gesprochen.

## 3.11 Aufzählung

### Situationsbeschreibung und Probleme

Es gibt unterschiedliche Notationen: *1.*, *1)*, oder *(1)*. Bisher wurden nur Aufzählungen der Form *1. 2. 3...* behandelt. Für diese Aufzählungen allerdings wird durch das

nachfolgende Wort entschieden, ob z.B. *1.* zu *erstens* oder *eins* expandiert wird. Folgt z.B. das (großgeschriebene) Wort *Der*, wird zwar richtig entschieden, dass es sich um einen Satzanfang handeln muss, Ordinalzahlen vor Satzanfängen werden in IMS-Festival aber als Kardinalzahlen gesprochen. Da eine Aufzählung für gewöhnlich am Beginn einer Äußerung erscheint, kann sie bisher nur richtig expandiert werden, wenn kein nur am Satzanfang großgeschriebenes Wort folgt.

### **Lösung**

Aufzählungen der oben beschriebenen Notationsweisen werden, wenn sie am Beginn einer Äußerung erscheinen, richtig expandiert.

## **3.12 Telefonnummern und Faxnummern**

### **Situationsbeschreibung und Probleme**

Telefonnummern werden auf sehr unterschiedliche Weise notiert. Diejenigen, die von IMS-Festival als Telefonnummer erkannt werden, werden ziffernweise gesprochen. Bisher werden nur Telefonnummern erkannt, deren Vorwahl von der Rufnummer durch Bindestrich oder '/' getrennt wird. Zwischen den Ziffern und dem Trennzeichen dürfen keine Leerzeichen stehen. Ansonsten werden sie jeweils als Kardinalzahlen gesprochen: *040 - 6724944: null vierzig sechs Millionen siebenhundertvierundzwanzigtausendneunhundertvierundvierzig*. Telefonnummern ohne Vorwahl (*Telefonnummer 01 242 42 29* wird expandiert zu *eintausendzweihundertzweiundvierzig zweiundvierzig neunundzwanzig*) und Telefonnummern, deren Vorwahl in Klammern steht, werden nicht erkannt. Von Telefonnummern, deren Rufnummer in Blöcke geteilt ist, wird nur der Anfang als Telefonnummer erkannt (*Telefonnummer 0180/23 23 424* wird expandiert zu *null eins acht null zwei drei dreiundzwanzigtausend vierhundertvierundzwanzig*).

### **Lösung**

Sowohl Telefonnummern mit Leerzeichen dazwischen, als auch solche, deren Vorwahl in Klammern steht, werden als Telefonnummern erkannt. Nummern ohne Vorwahl werden

erkannt, wenn eines der Wörter *Telefonnummer, Faxnummer, Tel., Fax* oder *Hotline* vorangeht.

### 3.13 Postfach und Postleitzahlen

#### Situationsbeschreibung und Probleme

Postfachnummern und Postleitzahlen werden bisher als Kardinalzahlen gesprochen: *Postfach 201551 in 53145 Bonn: zweihunderteintausendfünfhunderteinundfünfzig in dreiundfünfzigtausendeinhundertfünfundvierzig Bonn.* Auch sie sollten ziffernweise gesprochen werden.

#### Lösung

Ein Postfach ist einfach zu erkennen, da das Wort *Postfach* immer vorangeht. Die Ziffern dieser Zahl werden manchmal gruppiert, z.B. *Postfach 14 02 80, 53107 Bonn*, ansonsten werden sie zusammen geschrieben. Zahlen nach diesem Muster mit dem Wort 'Postfach' als Vorgänger werden ziffernweise gesprochen.

Postleitzahlen kann man daran erkennen, dass sie 5-stellig sind und nach dem Muster *Hausnummer, PLZ Stadt* angegeben werden. Nach diesem Muster werden sie abgefangen und ziffernweise gesprochen.

### 3.14 Bankleitzahlen und Kontonummern

#### Situationsbeschreibung und Probleme

Auch Bankleitzahlen und Kontonummern werden bisher als Kardinalzahl gesprochen, z.B. *Bankleitzahl 35060190: fünfunddreißig Millionen sechzigtausendeinhundertneunzig* oder *Konto 1015: eintausendfünfzehn.*

## Lösung

Sowohl Bankleitzahlen als auch Kontonummern sind einfach zu erkennen, da sie immer durch das Wort *Bankleitzahl* bzw. *Kontonummer* angekündigt werden und Bankleitzahlen außerdem immer 8-stellig sind. Eine Gruppierung von Kardinalzahlen, z.B. *550 620 80* wird von IMS-Festival nun auch als Bankleitzahl erkannt, wenn entweder *Bankleitzahl* oder *BLZ* vorangeht. Die einzelnen Blöcke werden jeweils als Kardinalzahlen gesprochen, z.B. *fünfhundertfünfzig sechshundertzwanzig achtzig*. Auch wenn die Ziffern einer Bankleitzahl nicht in Blöcke aufgeteilt sind, werden sie trotzdem blockweise gesprochen. Eine Kardinalzahl wird als Kontonummer erkannt, wenn ihr eines der Wörter *Kontonummer*, *Konto-Nr.*, *Kto.-Nr.* oder *Konto-Nummer* vorangeht. Sie werden nun ziffernweise gesprochen.

## 3.15 Zeitangaben

### Situationsbeschreibung und Probleme

Uhrzeiten werden erkannt, wenn die Stunden von den Minuten entweder durch Punkt oder Doppelpunkt getrennt sind und auf die Zahl entweder *Uhr* oder *h* folgt. In diesem Satz: *Die Wahllokale sollten von 8.00 bis 16.00 Uhr Ortszeit (15.00 bis 23.00 Uhr MESZ) geöffnet bleiben.*, wird jeweils die erste Uhrzeit nicht als solche erkannt, da das Wort *Uhr* nicht direkt darauf folgt: *von acht Punkt null null bis sechzehn Uhr Ortszeit fünfzehn Punkt null null bis dreiundzwanzig Uhr M E S Z geöffnet bleiben*. Auch die französische Schreibweise *fevrier 1996 a 13 h 00 sur FR3* wird nicht erkannt: *dreizehn h null*

Zeitangaben im Format Stunde:Minute:Sekunde werden nur erkannt, wenn auch die Stundenzahl zweistellig ist: *Endstand: 1. Prudnikowa 9:26:48 h* wird als *neun Doppelpunkt sechszwanzig Uhr achtundvierzig* gesprochen.

## Lösung

Uhrzeiten wie im oben genannten Beispielsatz, auf die ein *bis* oder *-* und eine weitere Uhrzeit folgt, werden nun auch als Uhrzeiten erkannt. Auch Uhrzeiten, auf die *MEZ* oder

MESZ folgt, werden als Uhrzeiten erkannt und korrekt expandiert. Uhrzeiten in Französischer Schreibweise werden von IMS-Festival nicht abgefangen, da sie in deutschen Texten einen Sonderfall darstellen und daher nicht besonders häufig auftreten.

Durch geringfügige Veränderung der Abfrage werden nun auch Zeitangaben mit einstelliger Stundenzahl erkannt.

## 3.16 Datum

### Situationsbeschreibung und Probleme

Da es sehr viele Datumsformate gibt, ist es kompliziert, ein Datum zu erkennen und korrekt auszusprechen. Es kann nach Jahr Monat Tag, Tag Monat Jahr oder sogar Monat Tag Jahr angegeben sein. Außerdem werden Punkte, Schrägstriche oder Bindestriche als Trennzeichen verwendet. In Deutschland ist allerdings vor allem die Reihenfolge Tag Monat Jahr in Gebrauch. IMS-Festival erkennt bisher Datumsangaben der Form *16.10.99*, *16.10.1999*, *5.10.87* und *10.5.87*. Datumsangaben nach dem Muster *16-10-2004* werden als kombinierte Abkürzung erkannt und somit nicht als Datum gesprochen. Bei der Auswertung sind ausserdem Datumsangaben in folgendem Format vorgekommen: JJ/MM/TT, die einmal als Telefonnummer (*95/04/13: fünfundneunzig null vier eins drei*) und einmal als Bruch (*95/11/19 bruch: fünfundneunzig elf neunzehntel*) analysiert worden sind. Zweistellige Jahreszahlen ab dem Jahr 2000 wurden bisher auch als Kardinalzahl gesprochen, d.h., dass der *15.04.04* als *fünfzehnte vierte vier* anstatt *fünfzehnte vierte null vier* gesprochen wird.

Wie bei den Zeitangaben besteht auch hier das Problem, dass der betrachtete Kontext nicht ausreichende Informationen liefert. Zum Beispiel *Am 24. und 25. April gehen wir baden.* wird expandiert zu *Am vierundzwanzigsten und fünfundzwanzigster April gehen wir baden.*

### Lösung

Die Abfrage für kombinierte Abkürzungen wurde soweit eingeschränkt, dass Datumsangaben nach dem Muster *16-10-2004* als Datum erkannt und korrekt expandiert wer-

den. Datumsangaben, die Schrägstriche als Trennzeichen verwenden, werden nun auch erkannt. Da die Lesart TT/MM/JJ in Deutschland eher üblich ist, wird diese Lesart erwartet. Nur wenn der vermeintliche 'Tag' höher als 31 ist, wird automatisch von der umgekehrten Lesart ausgegangen. Zweistellige Jahreszahlen, die eine Null enthalten, werden nun korrekterweise ziffernweise gesprochen.

Das Problem der fehlenden Kasus-Information bei der Expansion von bestimmten Datumsangaben konnte nicht im Rahmen dieser Studienarbeit behandelt werden.

### **3.17 Jahreszahlen**

Das einzige Problem, das während der Auswertung bei der Verarbeitung von Jahreszahlen auffiel, ist, dass Jahreszahlangaben im Format 1991/92 als Bruch erkannt wurden. Sie werden nun gemeinsam mit Brüchen behandelt und korrekt expandiert (siehe Kapitel 3.10.2).

### **3.18 Geld**

Während der Auswertung sind keine Probleme bei der Expansion von Geldbeträgen aufgetreten.

### **3.19 Verhältniszahlen**

Im Laufe der Auswertung sind keine Probleme bei der Verarbeitung von Verhältniszahlen aufgetreten.

## 3.20 Sonderzeichen

### 3.20.1 Bindestrich

#### Situationsbeschreibung und Probleme

Es ist nicht ganz einfach das Token '-' zu behandeln. Je nach Kontext kann es z.B. einen Gedankenstrich darstellen, als *minus* oder *bis* ausgesprochen werden. Ein Bindestrich, der zwischen zwei Jahreszahlen steht, wird von IMS-Festival bisher ignoriert, sollte jedoch als *bis* ausgesprochen werden. Auch wenn die Jahreszahlen und der Bindestrich nicht durch Leerzeichen getrennt eingegeben werden, sollte man das gleiche Ergebnis erhalten. Bisher erhält man für *1977-1999* die Expansion *ein tausend neun hundert sieben und siebenzig ein tausend neun hundert neun und neunzig*.

Steht ein Bindestrich zwischen zwei Datumsangaben, erwartet man für seine expandierte Form *bis zum*, falls das Wort vor dem ersten Datum *vom* lautet, ansonsten *bis*. Die Datumsangabe *Vom 1.1.2004 - 30.3.2004* wird bisher zu *Vom ersten ersten zwei tausend vier dreißigster dritter zwei tausend vier* expandiert - auch in diesem Fall wird der Bindestrich ignoriert. Außerdem sollte das zweite Datum, genauso wie das erste, im Dativ stehen.

Erscheint ein Bindestrich zwischen zwei Zahlen, und das Wort vor der ersten oder nach der letzten Zahl ist groß geschrieben, kann man davon ausgehen, dass es sich um eine Einheit handelt. Die expandierte Form von *2 - 4 TL* sollte *zwei bis vier Teelöffel* lauten. Auch dieser Bindestrich wird bisher nicht gesprochen: *zwei vier Teelöffel*. Bei Synthese eines Token, das aus einer Abkürzung und zwei durch Bindestrich getrennten Zahlen besteht, wie z.B. *S.20-25*, erhält man bisher: *S Punkt zwei null fünf und zwanzig*.

Ein Bindestrich direkt vor einer Zahl wird nicht gesprochen: Die Zahlen des Satzes *Aber auch aus den USA (-14 %), Belgien (-18 %) und Frankreich (-11 %) wurden erheblich weniger Gäste in den Schweizer Hotels empfangen.* wurden zu *vierzehn, achtzehn und elf* anstatt zu *minus vierzehn, minus achtzehn und minus elf* expandiert.

## Lösung

Ein Bindestrich zwischen zwei Jahreszahlen wird *bis* ausgesprochen. Auch wenn die Zahlen und der Bindestrich nicht durch Leerzeichen getrennt eingegeben werden, erhält man das gleiche Ergebnis.

Steht ein Bindestrich zwischen zwei Datumsangaben, wird er zu *bis zum* expandiert, falls dem ersten Datum das Wort *vom* vorangeht, ansonsten *bis*. Erscheint ein Bindestrich zwischen zwei Zahlen, wird er zu *bis* expandiert, wenn das Wort vor der ersten oder nach der letzten Zahl groß geschrieben wird, da es sich dann um eine Einheit handelt.

Token wie z.B. 'S.1-300', die aus einer Abkürzung und zwei durch Bindestrich getrennten Zahlen bestehen, werden am Punkt getrennt, für die Abkürzung wird nach einer Expansion gesucht. Die Zahlen werden getrennt verarbeitet und der Bindestrich wird zu *bis* expandiert.

Ein Bindestrich direkt vor einer Zahl darf nicht ignoriert werden. Er wird inzwischen als *minus* gesprochen.

### 3.20.2 Prozent

Während der Auswertung sind keine Schwierigkeiten bei der Expansion von % aufgetreten.

### 3.20.3 sonstige Sonderzeichen

#### Situationsbeschreibung und Probleme

Ein '+' wird bisher generell als plus gesprochen, z.B. im Fall von *Gruner + Jahr* sollte es jedoch zu *und* expandiert werden. Die Zeichen « und » werden bisher nicht erkannt und geben ein *unknown* aus.

## Lösung

Ein '+' zwischen Wörtern wird inzwischen als *und* gesprochen. Die Zeichen « und » wurden als Sonderzeichen eingetragen und werden nun als *Anführungszeichen* gesprochen.

# Kapitel 4

## Schlusswort

Durch die Auswertung der systematisch aus verschiedenen Korpora extrahierten Daten konnten deutlich mehr Probleme festgestellt werden als erwartet, wobei ein Großteil davon gelöst werden konnte. Besonders für die extrahierten Zahlen war die Betrachtung des Kontexts wichtig, um entscheiden zu können ob sie falsch expandiert wurden und wie sie korrekt expandiert werden müssen. Durch die systematische Extraktion von Daten konnte außerdem festgestellt werden, wie häufig Probleme in der Praxis auftreten.

Die Probleme, deren Lösung noch offen ist, sollen im Folgenden kurz zusammengefasst werden: Bei der Expansion der Zahl 1 entstehen Fehler aufgrund fehlender Genus- und Kasusinformationen. Auch die Fehler bei der Expansion von Ordinalzahlen, die durch fehlende Genus-, Kasus- und Numerusinformationen entstehen, und das Problem der fehlenden Kasusinformation bei der Expansion von bestimmten Datumsangaben sind noch nicht gelöst. Alle diese Probleme wären schnell gelöst, wenn Genus-, Kasus- und Numerusinformationen zur Verfügung stehen würden. Leider ist das bisher nicht der Fall. Die Beschaffung dieser Informationen würde außerdem durch auftretende Ambiguitäten erschwert.

Sicherlich können immer noch Probleme gefunden werden und neue Probleme entstehen, jedoch sind viele Fehler mit dem Abschluss dieser Studienarbeit behoben.

# Literaturverzeichnis

- [1] Black, Alan et al. *The Festival Speech Synthesis System. System documentation, Edition 1.4 for Festival Version 1.4.1.* Stand November 1999.
- [2] Breitenbücher, Mark. *Textvorverarbeitung zur deutschen Version des Festival Text-To-Speech Synthese Systems.* Studienarbeit. Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, 1997.
- [3] Bußmann, Hadumod. *Lexikon der Sprachwissenschaft.* Stuttgart: Alfred Kröner Verlag, 1990.
- [4] Dutoit, Thierry. *An Introduction to Text-To-Speech Synthesis.* Dordrecht: Kluwer Academic Publishers, 1997.
- [5] Glück, Helmut, Hrsg. *Metzler Lexikon Sprache.* Stuttgart; Weimar: Metzler, 1993.
- [6] Möbius, Bernd. *German and Multilingual Speech Synthesis.* AIMS Vol.7 Nr.4. Stuttgart: phonetik AIMS, 2001.
- [7] Möbius, Bernd. "Sprachsynthesysteme". *Computerlinguistik und Sprachtechnologie: Eine Einführung.* Hrsg. Carstensen, Kai-Uwe et al. Heidelberg; Berlin: Spektrum Akademischer Verlag, 2001: 462-468.
- [8] Sproat, Richard, Hrsg. *Multilingual Text-To-Speech Synthesis.* Dordrecht: Kluwer Academic Publishers, 1998.
- [9] Sproat, Richard et al. "Normalization of non-standard words". *Computer Speech and Language* (2001) 15: 287-333.
- [10] Werlin, Josef, Hrsg. *Wörterbuch der Abkürzungen DUDEN 11., 3. Neubearb. u. erw. Aufl.* Mannheim; Wien; Zürich: Dudenverlag, 1987.

[11] <http://de.wikipedia.org/wiki/Akronym>