

Universität Stuttgart  
Institut für Maschinelle Sprachverarbeitung (IMS)  
Azenbergstraße 12  
70174 Stuttgart

# Intonation und Wortgebung numerischer Ausdrücke

Ein anwendungsorientiertes Produktionsexperiment

Diplomarbeit Nr. 57

Katrin Schweitzer

angemeldet am: 15. Dezember 2005

abgegeben am: 15. Juni 2006

---

Betreuer und Erstgutachter: PD Dr. phil. Bernd Möbius  
Zweitgutachter: Prof. Dr. phil. Grzegorz Dogil



## **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.

Katrin Schweitzer

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Stand der Forschung</b>	<b>2</b>
<b>3</b>	<b>Produktionsexperiment</b>	<b>4</b>
3.1	Motivation und Ziel . . . . .	4
3.2	Stimulidesign . . . . .	4
3.2.1	Kategorien von Nummern . . . . .	5
3.2.1.1	Telefonnummern . . . . .	5
3.2.1.2	Kontonummern . . . . .	5
3.2.1.3	Bankleitzahlen . . . . .	5
3.2.1.4	Kreditkartennummern . . . . .	5
3.2.2	Kategorien von Phänomenen . . . . .	5
3.2.2.1	Abfolgen von Ziffern . . . . .	6
3.2.2.2	Nullen . . . . .	6
3.2.2.3	Wiederholungen von Ziffern . . . . .	6
3.3	Probanden . . . . .	7
3.4	Methoden . . . . .	7
3.4.1	Datenerhebung . . . . .	7
3.4.2	Automatische Vorverarbeitung . . . . .	7
3.4.3	Intonatorische Analyse . . . . .	8
3.4.3.1	Das ToBI-Modell . . . . .	8
3.4.3.2	vereinfachtes Modell . . . . .	9
3.4.4	Analyse der Wortgebung . . . . .	10
3.4.5	Datenaufbereitung . . . . .	11
3.4.5.1	Gruppierung des numerischen Ausdrucks . . . . .	11
3.4.5.2	Lokalisierung des Features . . . . .	12
3.4.5.3	Einordnung in Gruppierungskategorien . . . . .	13
3.4.5.4	Lokalisierung abweichender Wortgebung . . . . .	15
3.4.5.5	Lokalisierung der Pitchakzente . . . . .	15
3.4.5.6	Markierung von Akzentverschiebung . . . . .	15
3.4.5.7	Markierung der Featureposition . . . . .	15
<b>4</b>	<b>Modell der Prosodie numerischer Ausdrücke</b>	<b>17</b>
4.1	Motivation und Ziele der Dateninterpretation . . . . .	17
4.2	Einzeluntersuchungen . . . . .	17

4.2.1	Gruppierung . . . . .	17
4.2.1.1	Bankleitzahlen und Kreditkartennummern . . . . .	17
4.2.1.2	Kontonummern und Telefonnummern . . . . .	18
4.2.2	Akzentuierung . . . . .	18
4.2.3	Wortgebung . . . . .	18
4.3	Statistik . . . . .	18
4.3.1	Zu testende Variablen . . . . .	19
4.3.2	Testen auf statistische Unabhängigkeit . . . . .	19
4.3.2.1	$\chi^2$ -Test . . . . .	19
4.4	Ergebnisse . . . . .	20
4.4.1	Gruppierung in intonatorische Einheiten . . . . .	20
4.4.1.1	Kontonummern und Telefonnummern . . . . .	21
4.4.1.2	Kreditkartennummern . . . . .	27
4.4.2	Wortgebung . . . . .	31
4.4.2.1	Gleiche Einzelziffern . . . . .	34
4.4.2.2	Volle Hunderterzahlen . . . . .	35
4.4.2.3	Volle Zehnerzahlen . . . . .	36
4.4.3	Akzentuierung . . . . .	38
4.4.4	Korrelationen der untersuchten Parameter zu anderen Variablen . . . . .	38
4.4.4.1	Gruppierung . . . . .	39
4.4.4.2	Wortgebung . . . . .	41
4.5	Zusammenfassende Interpretation . . . . .	43
4.5.1	Gruppierung in intonatorische Einheiten . . . . .	44
4.5.1.1	Absteigende Ziffernfolgen . . . . .	44
4.5.1.2	Aufsteigende Ziffernfolgen . . . . .	45
4.5.1.3	Gleiche Einzelziffern . . . . .	45
4.5.1.4	Volle Hunderterzahlen . . . . .	46
4.5.1.5	Volle Zehnerzahlen . . . . .	46
4.5.1.6	Wiederholungen von Ziffern . . . . .	47
4.5.2	Wortgebung . . . . .	48
4.5.2.1	Gleiche Einzelziffern . . . . .	48
4.5.2.2	Volle Hunderterzahlen . . . . .	48
4.5.2.3	Volle Zehnerzahlen . . . . .	49
4.5.3	Akzentuierung . . . . .	49
<b>5</b>	<b>Praktische Anwendung des Modells</b>	<b>50</b>
5.1	Detektierung der Nummernfeatures . . . . .	50
5.1.1	Auf- bzw. absteigende Ziffernfolgen . . . . .	50
5.1.2	Gleiche Einzelziffern . . . . .	52
5.1.3	Volle Hunderterzahlen . . . . .	52
5.1.4	Volle Zehnerzahlen . . . . .	53
5.1.5	Wiederholung von nicht aufeinanderfolgenden Ziffern . . . . .	53
5.2	Bestimmung der Gruppierung . . . . .	54
5.2.1	Gleiche Einzelziffern . . . . .	56

5.2.2	Volle Hunderterzahlen . . . . .	56
5.2.3	Volle Zehnerzahlen . . . . .	57
5.2.4	Wiederholungen von Ziffern . . . . .	57
5.3	Bestimmung der Wortgebung . . . . .	58
5.3.1	Gleiche Einzelziffern . . . . .	59
5.3.2	Volle Hunderterzahlen . . . . .	59
5.3.3	Volle Zehnerzahlen . . . . .	60
<b>6</b>	<b>Fazit</b>	<b>61</b>
6.1	Diskussion . . . . .	61
6.2	Ausblick . . . . .	62
<b>A</b>	<b>Übersicht über die Stimuli im Produktionsexperiment</b>	<b>63</b>
A.1	Telefonnummern . . . . .	63
A.2	Kreditkartennummern . . . . .	65
A.3	Bankverbindungen . . . . .	66
A.3.1	Bankleitzahlen . . . . .	66
A.3.2	Kontonummern . . . . .	67
<b>B</b>	<b>R-Skripte</b>	<b>68</b>
<b>C</b>	<b>Alternativer Algorithmus für die Wortgebung von Hunderterzahlen</b>	<b>88</b>
	<b>Literatur</b>	<b>90</b>

# 1 Einleitung

Numerische Ausdrücke begegnen uns tagtäglich – zum Beispiel in Telefonnummern, Kontonummern, Bankleitzahlen und Kreditkartennummern.

Im Deutschen gibt es verschiedene Strategien, diese Ausdrücke zu strukturieren. Zum Teil sind diese so unterschiedlich, dass man die entsprechende Nummer nicht wiedererkennt, nur weil der Gegenüber sie anders gruppiert oder die Ziffern zu größeren, mehrstelligen Zahlen zusammensetzt.

Was jedoch nie passiert, ist, dass eine Nummer - um welchen Nummerntyp auch immer es sich handeln mag - völlig unstrukturiert und monoton gesprochen wird.

Für die maschinelle Sprachverarbeitung, im Speziellen die angewandte Phonetik, bedeutet dies, dass ein Sprachsynthesystem, das numerische Ausdrücke als monotone Sequenz von immer gleich betonten Ziffern realisiert, sehr unnatürlich wirkt und noch dazu die potentiellen Nutzer verärgern könnte, sind diese darauf angewiesen, sich die vorgelesene Nummer zu merken - eine Aufgabe, die durch fehlende Strukturierung deutlich erschwert wird. Eine Erklärung hierfür kann man in Miller (1956) finden. Er stellte fest, dass die Merkfähigkeit des menschlichen Gedächtnisses auf sieben Informationseinheiten beschränkt ist. Bestehen die Informationseinheiten nicht aus Einzelinformationen, sondern wurden diese zu sogenannten *Chunks* zusammengefasst, kann effektiv mehr Information gespeichert werden.

Daher ist es erstrebenswert, ein Modell zu entwickeln, das für verschiedene Arten von Nummern verschiedene Strategien zur Strukturierung und Wortgebung der Nummer ermittelt und das die Grundlage für eine Nummernkomponente eines Sprachsynthesystems bilden kann, so dass dieses numerische Ausdrücke gemäß der eigenen Prosodie eines jeden Ausdrucks, äußern kann.

Die vorliegende Arbeit stellt ein Produktionsexperiment vor, in dem die Intonation und die Wortgebung verschiedener Nummerntypen in Abhängigkeit bestimmter Eigenschaften der jeweiligen Ziffernfolge untersucht werden. Außerdem beschreibt sie, wie daraus ein Modell der Intonation und Wortgebung numerischer Ausdrücke entwickelt werden kann und schlägt eine mögliche Implementierung im Rahmen eines Synthesystems vor.

In Kapitel 2 werden bisherige Forschungsergebnisse zur Prosodie von Telefonnummern vorgestellt. Kapitel 3 beschreibt die Produktionsstudie und die Verarbeitung der akquirierten Daten. In Kapitel 4 wird aus den Ergebnissen der Produktionsstudie ein Modell für die Intonation und Wortgebung numerischer Ausdrücke entwickelt, das dann schließlich Grundlage für einen Vorschlag zur praktischen Anwendung ist, dessen Beschreibung sich in Kapitel 5 findet. Kapitel 6 fasst die Erkenntnisse zusammen und beschreibt mögliche weiterführende Arbeiten.

## 2 Stand der Forschung

Im Folgenden wird ein kurzer Überblick über die wenigen bisherigen Forschungsergebnisse zur Prosodie numerischer Ausdrücke gegeben.

Dass die prosodische Strukturierung von Telefonnummern ein sehr natürlicher, intuitiv notwendiger Vorgang ist, stellten Jacobi u. Reichel (2003) fest, die in ihrer Studie Probanden instruierten, Telefonnummern ungruppiert vorzulesen - was keiner der Versuchspersonen gelang.

Aufwändige Studien zur Prosodie numerischer Ausdrücke gibt es kaum. Simonsen u. a. (2004) untersuchten einen niederländischen Sprecher für eine Unit-Selection-Synthese und hielten am Rande ihrer Untersuchung fest, dass für die Datenbank drei hinsichtlich ihrer Akzentuierung verschiedene Formen für jede Ziffern aufgenommen werden sollten: eine neutrale Form, eine mit steigender, eine mit fallender Intonation.

Eine sehr relevante, weitaus aufwändigere und für die vorliegende Untersuchung grundlegende Studie wurde von Baumann u. Trouvain (2001) durchgeführt. Sie untersuchten, wie deutsche Sprecher Telefonnummern strukturieren, in dem sie die Äußerungen von zehn Sprechern hinsichtlich Gruppierung, Wortgebung und prosodischer Struktur analysierten.

Unter Gruppierung verstehen die Autoren das Einteilen der Nummer in zusammengehörige, semantisch-pragmatische Einheiten, also die Strukturierung einer Ziffernfolge in Kleingruppen von bis zu drei Ziffern.

Die prosodische Struktur fließt in diese Einteilung insofern ein, als die Gruppen durch Intonationsphrasen voneinander getrennt werden. Des Weiteren wird im Normalfall das jeweils letzte Element einer tonalen Phrase akzentuiert.

Mit der Wortgebung einer Äußerung ist dann Expansion der numerischen Ausdrücke in Wörter gemeint. Hierbei ist zu beachten, dass Ziffern innerhalb einer Gruppe sowohl als Einzelziffern als auch als zusammengefasste, längere Zahlwörter ausgesprochen werden können.

Baumann u. Trouvain (2001) berücksichtigten in ihrer Studie verschiedene Arten von Telefonnummern: Nummern mit Vorwahl, sechsstelligen Nummern, siebenstelligen Nummern, außerdem allgemein bekannte Nummern sowie auswendig gelernte. Die Nummern beinhalten bestimmte Phänomene von Ziffernfolgen, die daraufhin untersucht werden sollen, ob sie einen Einfluss auf die Struktur der Nummer haben. Dabei handelt es sich um aufeinanderfolgende und nicht-aufeinanderfolgende gleiche Einzelziffern, aufsteigende Ziffernfolgen sowie Nullen in verschiedenen Positionen.

Die Ergebnisse zeigen, dass die Gruppierung von Telefonnummern im Deutschen relativ uneinheitlich geschieht. Die Autoren stellten fest, dass jeder Sprecher eine bestimmte Gruppierungsstrategie bevorzugt und fanden für siebenstelligen Ziffern drei verschiedene



Strategien: die Gruppierung in drei Ziffern, gefolgt von zweimal zwei Ziffern (3-2-2), den umgekehrten Fall (2-2-3) und die Gruppierung Zweiergruppen gefolgt von einer einzelnen Ziffer (2-2-2-1).

Die Untersuchung von Gruppierung und Wortgebung, die die Autoren beschreiben, deutet darauf hin, dass die besonderen Phänomene in den Telefonnummern auch bestimmte Gruppierungsstrategien evozieren und auch die Wortgebung scheint zu einem kleinen Prozentteil von bestimmten Phänomenen beeinflusst zu sein. Allerdings werden die Ergebnisse nur deskriptiv statistisch dargestellt, nicht aber auf statistische Signifikanz untersucht.

Was die Akzentuierung der Ziffernfolgen anbelangt, so stellen Baumann u. Trouvain (2001) Standardakzentuierung für die Nummern fest: der nukleare Akzent wird - außer in der letzten Phrase - meist steigend realisiert, wohingegen die prä nuklearen Akzente meist lokale F<sub>0</sub>-Maxima sind, ohne einen unmittelbar folgenden Abfall der Tonhöhe.

Dem Produktionsexperiment lassen die Autoren noch eine Untersuchung zur Perzeption folgen. Sie vergleichen die drei im Experiment gefundenen Gruppierungsstrategien mit zwei Strategien von Telefonauskunftssystemen, wobei eines der Systeme ganz ohne prosodische Struktur in numerischen Ausdrücken arbeitet, das zweite eine der (inzwischen veralteten) deutschen Norm für die Schreibweise von Telefonnummern<sup>1</sup> entsprechende, rudimentäre prosodische Struktur verwendet. Die drei natürlichen Strategien werden gegenüber denen des Telefonsystems bevorzugt, jedoch kann man keine generelle Präferenz der rudimentär prosodisch strukturierten Nummern gegenüber den unstrukturierten feststellen.

Die Autoren schließen, dass ein Sprachsynthesystem Ziffernfolgen jeglicher Art prosodisch strukturieren sollte, wobei eine Default-Strategie empfohlen wird, die durch die besonderen Phänomene innerhalb der Nummer verändert wird.

Die Ergebnisse von Baumann u. Trouvain (2001) waren insofern richtungsweisend, als die Sensitivität für das Problem der Unnatürlichkeit numerischer Ausdrücke in Sprachsynthesystemen als tatsächliches Problem erkannt wurde. Breuer u. Abresch (2003) zum Beispiel berücksichtigten in ihre Unit-Selection-Sprachsynthesystem für eine Telefonauskunft die Ergebnisse insofern, als eine Gruppierungsstrategie eingeführt wurde, wenngleich es sich auch nicht um eine Umsetzung von Baumann u. Trouvains Ergebnissen handelt, sondern um eine Gruppierung im Sinne der DIN-Norm von 1996 - also dem Besseren der von Baumann u. Trouvain (2001) getesteten, nicht natürlichen Muster entsprechend.

Die vorliegende Arbeit baut auf Baumann u. Trouvain (2001) auf und erweitert sowohl die getesteten Phänomene als auch die untersuchten Nummerntypen. Des Weiteren werden die Ergebnisse mithilfe von statistischen Tests auf Signifikanz überprüft, um eine verlässliche Einschätzung des Einflusses der verschiedenen Phänomene auf die Intonation und Wortgebung der jeweiligen Nummer zu ermöglichen.

---

<sup>1</sup>Telefonnummern werden typografisch so angeordnet, dass die Nummer von hinten gesehen in Zweiergruppen aufgeteilt ist. Ungerade Ziffernfolgen beginnen also mit einer einzelnen Ziffer, gerade bestehen nur aus Zweiergruppen (vgl. Deutsches Institut für Normung e.V. (1996), zitiert in Baumann u. Trouvain (2001))

# 3 Produktionsexperiment

Im Folgenden wird ein Produktionsexperiment zur Intonation und Wortgebung von numerischen Ausdrücken im Deutschen vorgestellt. Das Design des Experiments orientiert sich stark an der Produktionsstudie von Baumann u. Trouvain (2001) und untersucht die verschiedenen Phänomene, die diese als relevant für Gruppierung und Wortgebung von numerischen Ausdrücken klassifiziert haben. Anders als in der Studie von Baumann u. Trouvain (2001) wurden im vorliegenden Experiment nicht nur Telefonnummern, sondern auch andere numerische Ausdrücke betrachtet. Außerdem wurden die untersuchten Phänomene innerhalb der Ziffernfolge, die Einfluss auf Gruppierung und Wortgebung zu haben scheinen, erweitert.

## 3.1 Motivation und Ziel

Diese Studie soll dazu dienen, die Annahmen von Baumann u. Trouvain (2001) zu überprüfen und ein explizites Modell für die Intonation und Wortgebung numerischer Ausdrücke zu erstellen, das auch als Richtlinie dienen kann, um eine Modul zur Generierung der Struktur solcher Ziffernfolgen im Rahmen eines Sprachsynthesystems zu implementieren.

## 3.2 Stimulidesign

Es handelt sich um eine klassische Produktionsstudie mit neun Versuchspersonen und fünfundsiebzehn vorzulesenden Stimuli. Die Stimuli waren in Trägersätze der Form

Die *<Nummerntyp>*nummer lautet *<Stimulus>*.

eingebettet und enthielten die von Baumann u. Trouvain (2001) für die Gruppierung der Ziffern und für die Wortgebung als relevant eingestuft Phänomene, sowie einige zusätzliche Abfolgen von Ziffern, die als mögliche Auslöser für Veränderungen in der von Baumann u. Trouvain (2001) festgestellten Default-Gruppierung in Frage kamen. Diese speziellen Eigenschaften einer Nummer, also die Phänomene, deren Auswirkung auf Gruppierung und Wortgebung der Ziffernfolge in der Vorliegenden Arbeit untersucht werden soll, werden im folgenden als *Features* bezeichnet. Die Stimuli des Experimentes lassen sich also auf zwei Arten kategorisieren: nach Nummerntyp und nach Features. Eine vollständige Übersicht über die Ziffernfolgen findet sich in Anhang A.3.2.

## **3.2.1 Kategorien von Nummern**

Anders als bei Baumann u. Trouvain (2001) wurden nicht nur Telefonnummern, sondern vier verschiedene Arten von Nummerntypen betrachtet. Neben Telefonnummern und Kontonummern, die - aufgrund ihrer Unabhängigkeit von der typografischen Darstellung - den Schwerpunkt der Untersuchung darstellten, wurden auch Kreditkartennummern und Bankleitzahlen untersucht.

### **3.2.1.1 Telefonnummern**

Bei den Telefonnummern wurden für jedes Feature vier Nummernfolgen präsentiert, wobei die Position des Features innerhalb der Nummer variierte. Der Großteil der Nummern erstreckte sich über sieben Zeichen, obwohl es auch kürzere Zahlenfolgen sowie Telefonnummern mit Vorwahl gab. Die Nummern wurden typografisch nicht gruppiert, sondern als eine zusammenhängende Zahlenfolge präsentiert. Lediglich die Stimuli, bei denen einen Vorwahl mit angegeben wurde, wurden in der typischen „Vorwahl-Notation“, also durch einen Bindestrich getrennt, gezeigt.

### **3.2.1.2 Kontonummern**

Die Kontonummern wurden alle in einer zusammenhängenden Zahlenfolge ohne Trennzeichen präsentiert. Es handelte sich um sieben- bis achtstellige Ziffernfolgen, innerhalb derer die Position des jeweiligen Features variierte.

### **3.2.1.3 Bankleitzahlen**

Die Bankleitzahlen wurden in der für Bankleitzahlen typischen 3-3-2-Notation präsentiert, das heißt, nach den ersten drei sowie den zweiten drei Ziffern wurde ein Leerzeichen gesetzt. Sie bestanden alle aus sieben Ziffern, wobei an verschiedenen Stellen in der Nummer das jeweilige Feature zu finden war.

### **3.2.1.4 Kreditkartennummern**

Das typografische Format der Kreditkartennummern entsprach der üblichen Notation, das heißt, sie wurden in vier durch ein Leerzeichen getrennten Viererblöcken präsentiert. Wie bei den Kontonummern war jedes Feature mehrmals innerhalb der langen Zahlenfolge zu finden, so dass die Featureposition variiert werden konnte, ohne den Versuchspersonen zu lange Aufnahmezeiten zuzumuten.

## **3.2.2 Kategorien von Phänomenen**

Die Einteilung der Phänomene und somit die Festlegung derjenigen Features, die im Experiment getestet werden sollten, erfolgte in Anlehnung an Baumann u. Trouvain (2001). Es handelt sich weitestgehend um die gleichen Phänomene, das heißt, im Wesentlichen wurden Baumann u. Trouvains Kategorien übernommen, es wurden jedoch noch einige wenige Phänomene hinzugenommen.

<i>ab</i>
<u>6783914</u> • <u>5437</u> <u>1987</u> <u>2564</u> <u>8769</u> • <u>154</u> <u>987</u> <u>56</u>
<i>auf</i>
<u>9786234</u> • <u>6783</u> <u>9234</u> <u>1567</u> <u>7895</u> • <u>678</u> <u>923</u> <u>67</u>

Abbildung 3.1: Beispiele für die Nummernfeatures *ab* und *auf*

<i>zehn</i>
<u>6307520</u> • <u>7039452</u> • <u>6307</u> <u>2052</u> <u>5790</u> <u>6492</u> • <u>630</u> <u>205</u> <u>90</u>
<i>hundert</i>
<u>2005896</u> • <u>8690031</u> • <u>6300</u> <u>5004</u> <u>6517</u> <u>1005</u> • <u>300</u> <u>501</u> <u>07</u>

Abbildung 3.2: Beispiele für die Nummernfeatures *zehn* und *hundert*

### 3.2.2.1 Abfolgen von Ziffern

Die Effekte von Abfolgen von Ziffern wurden anders als bei Baumann u. Trouvain (2001) nicht nur für aufsteigende, sondern auch für absteigende Ziffernfolgen untersucht. Es handelte sich um unterschiedliche Folgen, die sich jeweils über zwei oder drei Ziffern erstreckten. Für die Klassifizierung der Nummern zur späteren Analyse ergaben sich somit zwei verschiedene Features für die entsprechenden Nummern, die mit *ab* und *auf* markiert wurden (vgl. Abb. 3.1, S. 6).

### 3.2.2.2 Nullen

Da in vorherigen, informellen Tests festgestellt wurde, dass die Nullen in der Ziffer häufig eine Wortgebung, die von der Standardwortgebung abweicht, evozieren, das heißt, dass die vorherige Ziffer und die Null als Zehnerzahl ausgesprochen werden, sollte auch untersucht werden, wie es sich mit zwei aufeinanderfolgenden Nullen, also der Bildung von vollen Hunderterzahlen, verhält. Daher wurde dieses Feature auch als eigenständiges Phänomen abgetestet, weswegen sich auch hier zwei Kategorien von Merkmalen der Nummer unterscheiden ließen, sie wurden mit *zehn* und *hundert* markiert (vgl. Abb. 3.2, S. 6).

### 3.2.2.3 Wiederholungen von Ziffern

Die Effekte, die sich wiederholende Ziffern auf die Struktur der numerischen Ausdrücke haben, wurden hinsichtlich verschiedener Kombinationen von sich wiederholenden Ziffern getestet. Zum einen wurden gleiche Einzelziffern getestet. Diese wurden unter zwei verschiedenen Bedingungen abgeprüft: aufeinanderfolgende und getrennt auftretende Einzelziffern sowie gleiche Ziffernpaare. Nummern mit aufeinanderfolgenden identischen Ziffern wurde mit *gleiche*, die anderen Nummern mit *wdh* markiert (vgl. Abb. 3.3, S. 7). Eine Ziffernfolge wie 6236284 wurde also mit dem Feature *wdh* markiert, da die Ziffernfolge 62 zweimal in der Nummer vorkommt. Es wurden sowohl die wiederholten Ziffern

<i>gleiche</i>
6335841 • 6332 2772 5573 8622 • 6358441
<i>wdh</i>
5393278 • 63664695 • 6236284 • 6367 5852 9584 7584 •

Abbildung 3.3: Beispiele für die Nummernfeatures *gleiche* und *wdh*

als auch die Folgeziffern als Teile des Features markiert. Auch die Ziffernfolge 5393278 fiel in die Kategorie von sich wiederholenden, aber nicht aufeinanderfolgenden Ziffern, da die Einzelziffer 3 zweimal in ihr vorkommt. In diesem Fall wurden die wiederholten Ziffern sowie die Vorgängerziffern markiert.

### 3.3 Probanden

Das Experiment wurde mit neun Probanden durchgeführt, davon zwei weibliche. Alle Versuchspersonen waren Studenten der Computerlinguistik und wurden vorher nicht auf die zu untersuchenden Phänomene hingewiesen. Sie wurden instruiert, die Nummern, die ihnen präsentiert wurden, still zu lesen und dann so vorzulesen, dass ein potentieller Zuhörer die Nummer gut aufschreiben oder auswendig behalten kann.

### 3.4 Methoden

#### 3.4.1 Datenerhebung

Die Aufnahmen fanden im schallarmen Raum des IMS (Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart) statt. Die Stimuli wurden einmal zufällig sortiert, so dass alle Probanden ein und die selbe Abfolge von Stimuli, nicht sortiert nach Nummerntyp oder Feature, zu lesen hatten. Jeder Stimulus erschien erst wenn er aufgenommen werden sollte, auf dem Bildschirm im Aufnahmezimmer, das heißt, die Versuchspersonen sahen immer genau eine Ziffernfolge, die dann auch sofort gelesen und diktiert werden sollte. Die Aufnahmen wurden mit einer Abtastrate von 48 kHz gemacht. Jeder Stimulus wurde in einer einzelnen Datei gespeichert, so dass später fehlgeschlagene Aufnahmen ohne großen Aufwand aussortiert werden konnte.

#### 3.4.2 Automatische Vorverarbeitung

Die Laut- bzw. Silben- und Wortgrenzen der einzelnen Segmente wurden mittels des am IMS verfügbaren Aligners (Rapp 1995) automatisch aligniert. Neben einem 16 kHz Soundfile benötigt der Aligner für jedes Aufnahmen-Segment ein Textfile, welches eine orthographische Transkription des im Segment gesprochenen Textes enthält. Diese wurden aus der Präsentationsliste der Stimuli automatisch erstellt und in einem zweiten Schritt mittels eines interaktiven Perl-Skriptes auf die jeweiligen Aufnahmen der

Versuchspersonen angepasst, da häufig während der Aufnahme nicht vorhergesehene metasprachliche Einschübe der Probanden eine automatische Alignierung mithilfe der Trägersätze unmöglich machten.

Die F0-Kontur wurde mittels des in Wavesurfer (Sjölander u. Beskow 2000) implementierten esps-Algorithmus erzeugt.

### 3.4.3 Intonatorische Analyse

Die Intonation jeder Äußerung wurde manuell annotiert. Den Annotationen lag das Stuttgarter GToBI-Modell von Mayer (1995) zugrunde. Das Modell basiert auf dem ToBI-Annotationssystem (Pierrehumbert 1980), einem phonologischen Intonationsmodell, bzw. dessen Modifikation für das Deutsche (Grice u. Benz Müller 1995), und somit auf der autosegmental-metrischen Theorie der Intonation (vgl. Ladd 1996). Das bedeutet, dass Intonation als lineare Struktur angesehen wird, die sich aus verschiedenen intonatorischen Ereignissen – Tönen – zusammensetzt. Die tonale Struktur wird also als vom Phonem-Segment unabhängig, aber selbst aus Segmenten bestehend verstanden. Diese Ereignisse werden als kategorial angesehen, wenngleich es dazu bislang keine umfassende empirische Evidenz gibt. Für verschiedene Sprachen wurde zwar teilweise kategoriale Wahrnehmung von Grenztönen (Falé u. Faria 2006; Schneider u. Lintfert 2003) oder Pitch-Bewegungen (House 1996; Kohler 1987) nachgewiesen, jedoch gibt es noch keinen Konsens darüber, ob tonale Kontraste kontinuierlich oder kategorial wahrgenommen werden.

#### 3.4.3.1 Das ToBI-Modell

Im ToBI-Modell gibt es neben verschiedenen Markierungen für die Töne phonetische Realisierungsregeln, die beschreiben, wie diese auf konkrete Grundfrequenzwerte abgebildet werden sollen. Zwischen diesen Werten wird interpoliert.

Die tonalen Ereignisse lassen sich aufgrund ihrer Eigenschaften in zwei Gruppen unterteilen: *Pitchakzente* markieren Prominenz, sind also diejenigen Ereignisse, die die tonale Bewegung an den akzentuierten Silben eines Satzes (den Satzakkzenten) bzw. in der Nähe dieser Silben, ausmachen.

*Phrasenakzente und Grenztöne* markieren Grenzen von tonalen Phrasen, also diejenigen Ereignisse, die die tonale Bewegung am Phrasenende, und somit eher die globale Struktur der tonalen Phrase bestimmen. Sowohl im ursprünglichen ToBI-System als auch in seinen Adaptionen für das Deutsche gibt es zwei verschiedene Arten von tonalen Phrasen und somit auch von Labels für die Phrasenenden. Es wird zwischen intermediären Phrasen (markiert durch Phrasenakzente) und Intonationsphrasen (markiert durch Grenztöne) unterschieden, wobei die Intonationsphrasen durch deutlich wahrnehmbare und intuitiv erkennbare Einschnitte gekennzeichnet sind, wohingegen die intermediären Phrasen kleinere, schwerer wahrnehmbare tonale Einheiten sind, die durch ein Zögern innerhalb der Äußerung, einen höheren Pitchakzent oder auch eine Unstetigkeit in der F0-Kontur erkennbar sind.

Die Einheiten jeder dieser Gruppen bedienen sich zweier verschiedener Ebenen, mithilfe derer der Tonhöhenverlauf möglichst genau beschrieben werden soll. **H** bezeichnet einen Ton im oberen Drittel des Registers des Sprechers, **L** steht für einen tiefen Ton (im unteren Drittel). Im amerikanischen ToBI-System wird davon ausgegangen, dass sowohl intermediäre als auch Intonationsphrasen mit einem Endton realisiert werden. Es gibt also für jede Art von tonaler Phrasengrenze ein hohes und ein tiefes Label, jeweils kombiniert mit der Information über die Art der Phrasengrenze: % für Intonations- und – für intermediäre Phrasen. Das Stuttgarter System macht diese Annahmen nicht, da im Deutschen der Tonhöhenverlauf sehr stark durch den nuklearen Akzent bestimmt sein kann. Daher besteht in diesem System nicht nur die Möglichkeit, hohe und tiefe Grenztöne zu annotieren, sondern auch, eine „neutrale“ Phrasengrenze für Intonationsphrasen. Bei den intermediären Phrasen wird gänzlich auf die Möglichkeit einer Ebenenangabe verzichtet, da davon ausgegangen wird, dass die F0-Bewegung am Ende der Phrase durch den nuklearen Akzent bestimmt wird.

Für Akzente gibt es verschiedene Möglichkeiten. Sowohl mono- als auch bitonale Akzente können markiert werden. Grundsätzlich erkennt man Labels für Pitchakzente an dem Asterisk (\*), der signalisiert, dass es sich um einen Pitchakzent handelt. Der **H**- oder **L**-Ton, der vor dem Asterisk steht, kennzeichnet die Tonhöhe auf der akzentuierten Silbe. Der nachfolgende Ton bei bitonalen Akzenten bezeichnet die Tonhöhe des Verlaufstons (*trailing tone*), beschreibt also, wohin sich die F0-Kontur an den postakzentuierten Silben bewegt. Im originalen ToBI-System wird die zeitliche Alignierung des Akzentes durch den Stern markiert, das heißt, es gibt verschiedene Arten von steigenden und fallenden Akzenten. Im Stuttgarter System entfällt dies.

Die tonalen Ereignisse folgen einer hierarchischen Struktur (vgl. Abb. 3.4, S. 10). Jede Äußerung besteht aus einer oder mehreren Intonationsphrasen, die ihrerseits aus einer oder mehreren intermediären Phrasen bestehen. In jeder intermediären Phrase gibt es mindestens einen Pitchakzent. Der letzte Pitchakzent wird als nuklearer Akzent bezeichnet und am stärksten wahrgenommen.

Im amerikanischen System ergeben sich dadurch für Phrasengrenzen, bei denen Phrasenakzent und Grenzton zusammenfallen, zwei Labels. Im Stuttgarter System, in dem ja die Phrasenakzente keinen Ton tragen, wird der Phrasenakzent in solchen nicht markiert, um Redundanz zu vermeiden.

### 3.4.3.2 vereinfachtes Modell

Für die Beschreibung der Intonation numerischer Ausdrücke wurde das Stuttgarter ToBI-System in stark vereinfachter Weise verwendet. Pitchakzente wurden zwar steigend (**LH**) und fallend (**HL**) markiert, da jedoch für eine Sprachsynthesekomponente, die numerische Ausdrücke verarbeitet, ein Standardakzentmuster<sup>1</sup> als wünschenswert

---

<sup>1</sup>Es wird von einem Muster /**LH**+ **HL**/ ausgegangen, da eine Abweichung von dieser „Hutkontur“ den numerischen Ausdruck unverständlicher machen würde: ein früher fallender Akzent signalisiert eine Finalität, die in diesem Fall irreführend wäre, ein steigender Akzent am Äußerungsende würde beim Hörer eine unangebrachte Erwartungshaltung evozieren.

### ToBI Domänen

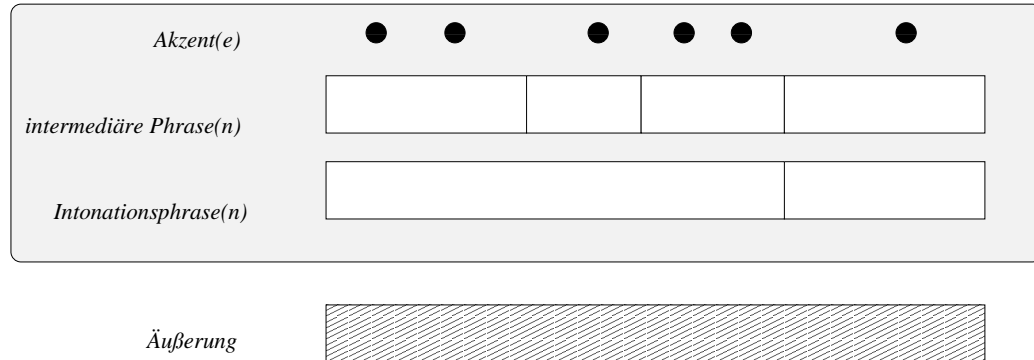


Abbildung 3.4: Hierarchie der intonatorischen Domänen in ToBI-Modellen

erachtet wurde, wurde in der vorliegenden Studie nur auf die Position des Pitch-Akzentes geachtet.

Das Hauptaugenmerk der Untersuchung wurde auf die Gruppierung der numerischen Ausdrücke in kleinere, zusammengehörige Zifferngruppen gerichtet. Daher wurden Labels gesetzt, kleineren ( $\sim$ ) und größeren ( $/$ ) Zäsuren entsprechen, die sich also, sollte dies für die Implementierung nötig sein, in die ip- und IP-ToBI-Labels – und % übersetzen lassen würden. Hohe Phrasengrenzen wurden testweise für drei Sprecher annotiert, da diese jedoch extrem selten sind und auch für die Anwendung nicht relevant und da sie durch keines der Features bedingt zu sein schienen, wurde bei den restlichen Sprechern auf die Markierung verzichtet. Tiefe Phrasengrenzen, die im Stuttgarter ToBI-Dialekt extrem selten vorkommen, traten nicht auf.

Die Annotationen wurden in zwei verschiedenen Labeldateien gemacht.

#### 3.4.4 Analyse der Wortgebung

Die Wortgebung der Äußerungen wurde in einer separaten Annotationsebene markiert, sofern sie von der Standardwortgebung abwich. Als Standardwortgebung wurde diejenige Wortgebung definiert, bei der jede Ziffer als ein Zahlwort ausgesprochen wurde. Bei einer ersten informellen Untersuchung der Wortgebung wurden verschiedene Arten von abweichenden Wortgebungen festgestellt: die Probanden verwendeten längere als einstellige Zahlen, um die Nummer zu beschreiben – es traten Zehner- und Hunderterzahlen auf. Für beide Kategorien wurde je ein Label verwendet (*als.10* und *als.100*). Des Weiteren verwendeten die Sprecher für die gleichen Zahlen teilweise die Formulierung



„dreimal die ...“ oder auch „zweimal die“, für alle diese Fälle wurde ein weiteres Label eingeführt (*als.mal*). Somit gab es, sofern die Wortgebung von der Standardwortgebung abwich, drei verschiedene Kategorien von abweichender Wortgebung.

### 3.4.5 Datenaufbereitung

Hauptgegenstand der Untersuchungen sollte die Gruppierung und Wortgebung der numerischen Ausdrücke sein. Es musste also eine Repräsentation gefunden werden, die die Gruppierung und die Wortgebung in Relation zu den im numerischen Ausdruck enthaltenen Features setzt. Um eine automatische Auswertung möglich zu machen, sollte diese Repräsentation relativ leicht mit Shell- oder Perlskripten weiterverarbeitet werden. Daher wurde jede Äußerung als ASCII-Repräsentation kodiert. Dies geschah in mehreren Verarbeitungsschritten (vgl. Abb. 3.7, S. 14).

#### 3.4.5.1 Gruppierung des numerischen Ausdrucks

Die vorgelesenen Ziffern sollten in der ASCII-Repräsentation der Realisierung gemäß der vom Sprecher verwendeten Intonation gruppiert werden. Um dies zu erreichen, wurde auf die automatisch alignierten Wortgrenzen sowie auf die manuell gesetzten Phrasengrenzen zurück gegriffen.

Beide Labeldateien lagen im waves-Labelformat (vgl. Abb. 3.5, S. 12) vor, das heißt, die Angabe des genauen Zeitpunkt des Labels (in der ersten Spalte des Files) konnte genutzt werden um die beiden zu kombinieren.

Dafür musste jedoch sichergestellt sein, dass die Phrasengrenzen millisekundengenau an der Wortgrenze annotiert sind. Dies war aufgrund der manuellen Annotation nicht der Fall (vgl. Abb. 3.6, S. 13), weswegen die Phrasengrenzen diesbezüglich abgeprüft und gegebenenfalls mithilfe eines Perlskripts an die Position der Wortgrenzen verschoben wurden.

Danach wurden die beiden Labelfiles kombiniert und die verschiedenen Labels mithilfe der Zeitpunkte in die richtige Reihenfolge gebracht. Aus dieser Abfolge wurde dann eine flache Repräsentation erzeugt. Für eine der Realisierungen der Kreditkartennummer 2444 8887 4856 2596 beispielsweise, wurde die Zeichenkette 2~444-8887-4856-25~96 gespeichert, da die Sprecherin nach der ersten und der 14. Ziffer je eine intermediäre, nach der vierten, achten und zwölften Ziffer hingegen eine Intonationsphrasengrenze<sup>2</sup> gesetzt hatte.

Aus dieser flachen, ausführlichen Repräsentation wiederum wurde eine Übersicht über das Gruppierungsschema generiert, aus der abzulesen war, wie viele Ziffern in einer tonalen Phrase realisiert wurden. Im Fall des diskutierten Beispiels wurde also die Repräsentation 1~3-4-4-2~2 erzeugt.

---

<sup>2</sup>Die Tilde stand weiterhin für die ip-Grenze, der Slash, der die IP-Grenze markierte wurde aus Gründen der besseren Verarbeitbarkeit hier durch einen Gedankenstrich ersetzt

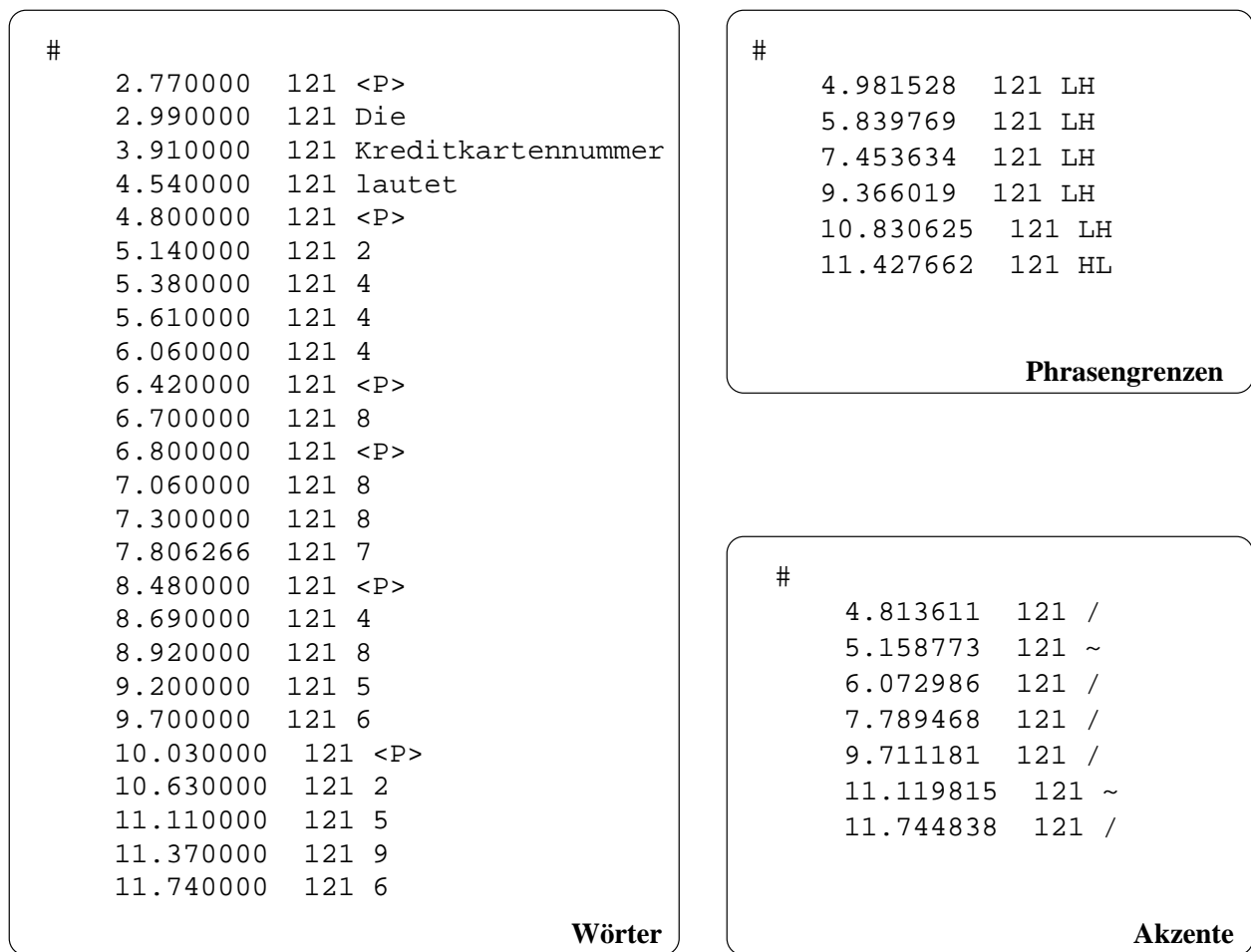


Abbildung 3.5: waves-Labeldateien für Wörter (automatisch annotiert), Phrasengrenzen und Akzente (beides manuell annotiert)

### 3.4.5.2 Lokalisierung des Features

Um das Feature nun mit dieser Repräsentation zu assoziieren, musste die Ziffernfolge auf das entsprechende Feature abgeprüft werden, weswegen sie mithilfe eines Perlskripts eingelesen und systematisch auf gleiche, aufeinanderfolgende und sich wiederholende Ziffernfolgen, sowie volle Zehner- und Hunderterzahlen durchsucht wurde. Die Ausgabe erfolgte als Zeichenkette, wobei die Features durch einen Buchstaben und die Ziffern, die nicht Teil des Features waren, durch eine Null repräsentiert wurden. Für das obige Beispiel erhält man also die Repräsentation `0~bbb-ccc0-0000-00~00`, wobei `b` für das erste Vorkommen des Features „gleiche Ziffern“ stand und `c` für das zweite. Für diejenigen Ziffernfolgen, in welchen das Feature mehr als einmal vorkam, wurde die Information für jedes Feature distribuiert, wobei mit vermerkt wurde, um das wievielte Vorkommen des Features innerhalb der Ziffernfolge es sich handelt. Für das obige Beispiel wurden also zwei Datensätze abgespeichert, wovon einer durch die ASCII-Kette

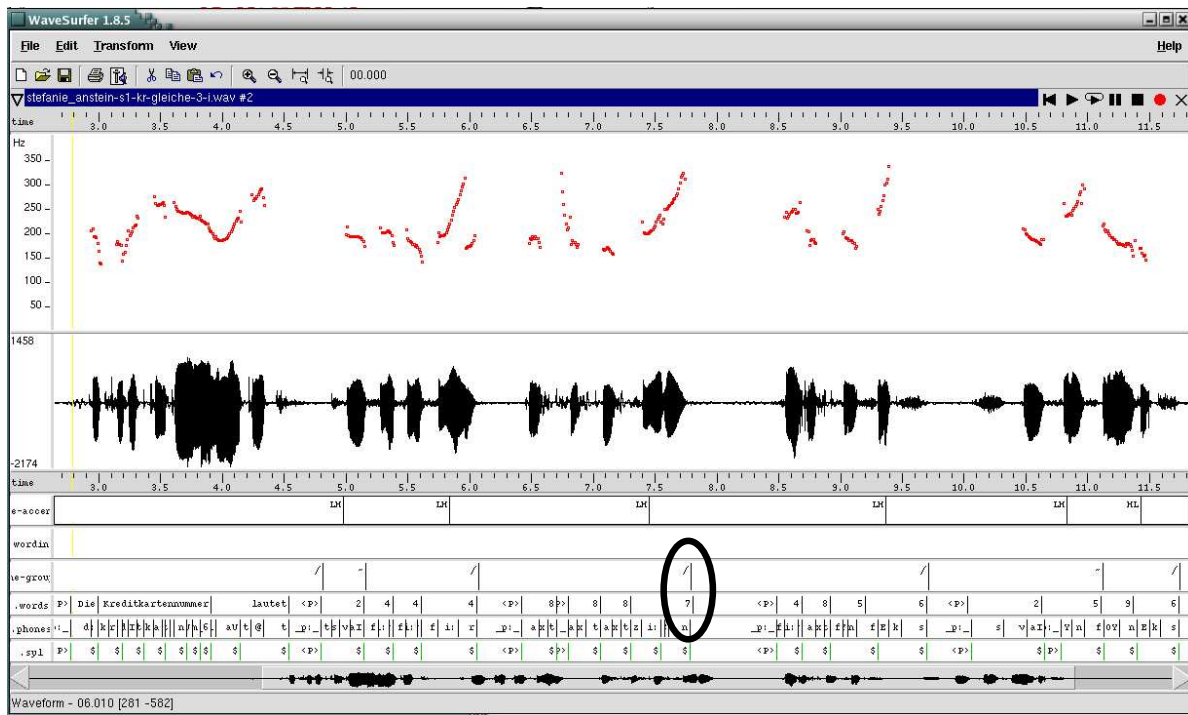


Abbildung 3.6: Gelabelte Äußerung eines numerischen Ausdrucks. Die zeitlich falsch gesetzte Phrasengrenze ist markiert.

0~bbb-0000-0000-00~00, der andere durch 0~000-ccc0-0000-00~00 repräsentiert wurde und jeweils einen Vermerk „erstes“ bzw. „zweites“ Vorkommen erhielten.

### 3.4.5.3 Einordnung in Gruppierungskategorien

Diese Repräsentation war die Basis für die Einordnung der Gruppierung in eine oder mehrere Gruppierungskategorien. Anhand ihrer wurde überprüft, ob

1. direkt vor oder nach dem Feature eine IP- oder ip-Grenze realisiert worden war (*IP.before.F*, *ip.before.F*, *IP.after.F*, *ip.after.F*).
2. eine ip- oder IP-Grenze innerhalb des Features gesetzt wurde (*IP.cuts.F*, *ip.cuts.F*)
3. nur das Feature und sonst keine weiteren Ziffern innerhalb einer ip oder IP zu finden waren, also ob sowohl direkt vor, als auch direkt nach dem Feature eine Phrasengrenze realisiert worden war (*only.F.in.IP*, *only.F.in.ip*)
4. keine Phrasengrenze vor, innerhalb oder nach dem Feature gesetzt wurde (*no.boundary*)

Für jede dieser Bedingungen wurde festgehalten, ob sie in der jeweiligen Ziffernfolge zutraf oder nicht.

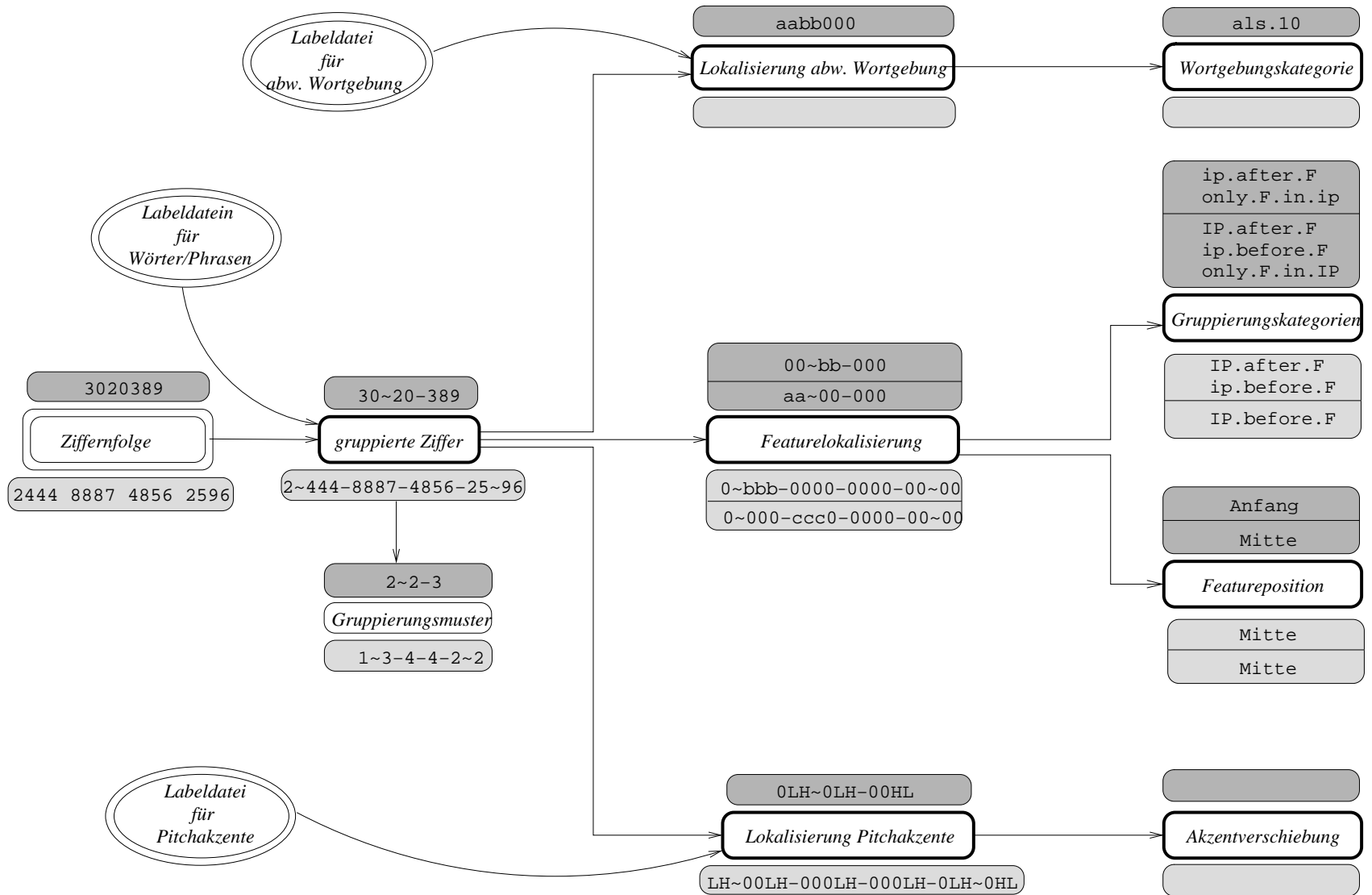


Abbildung 3.7: Verarbeitungsschritte zur Ermittlung der Gruppierung des numerischen Ausdrucks mit konkreten Beispielen. Im Fall des hellgrau hinterlegten Beispiels handelt es sich um eine Kreditkartennummer, im dunkelgrau hinterlegten um eine Kontonummer.

#### 3.4.5.4 Lokalisierung abweichender Wortgebung

Um die abweichende Wortgebung (falls vorhanden) zu lokalisieren, wurde die manuelle Annotation der Wortgebung mit den Labels für die Wortgrenzen kombiniert. Dann wurde, ähnlich der Featurelokalisierung, eine ASCII-Repräsentation für die abweichende Wortgebung erstellt, in der ein Buchstabe für die abweichende Wortgebung, eine Null für die restlichen Ziffern stand. Im diskutierten Beispiel trat keine abweichende Wortgebung auf, in anderen Datensätzen jedoch konnte man viele hintereinander auftretende längere Zahlen beobachten, so wurde zB die Kreditkartennummer 6307 2052 5790 6492 als „*dreiundsechzig null sieben zwanzig zweiundfünfzig siebenundfünfzig neunzig vierundsechzig zweiundneunzig*“ realisiert, was als `aa00bbccddeeffgg` kodiert wurde.

#### 3.4.5.5 Lokalisierung der Pitchakzente

Die Pitchakzente wurden lokalisiert, indem das Labelfile für die Akzente mit dem für die Wörter kombiniert wurde, eine Ziffern-Akzentabfolge erzeugt und schließlich die Ziffer, auf der der Akzent realisiert worden war, durch den Akzent ersetzt wurde. In einem weiteren Schritt wurde dann, ähnlich der Featurelokalisierung, eine Repräsentation erzeugt, in der für die Ziffern eine Null und für die Akzente das entsprechende Label steht. Die Akzentlokalisierung für die beiden obigen Datensätze gab die Zeichenkette `LH~00LH-000LH-000LH-0LH~0HL` aus.

Die Realisierungen mit abweichender Wortgebung wurden getrennt bearbeitet, hier wurde die Länge der abweichenden Wortgebung (aus dem manuell annotierten File) mit berücksichtigt, so dass in der ASCII-Kette keine Verschiebungen der Akzente auftraten. Das Ausgabeformat entsprach dem für die Daten, in denen keine abweichende Wortgebung auftrat.

#### 3.4.5.6 Markierung von Akzentverschiebung

Mithilfe der Repräsentation für die Akzentlokalisierung wurde geprüft, ob der Akzent von der Default-Position (auf der letzten Ziffer des numerischen Ausdrucks) auf eine andere Position innerhalb der Intonationseinheit verschoben wurde. Dies wurde für das Feature „Wiederholung von Ziffern,, (*wdh*) erwartet. Für jede Ziffer in der Intonationseinheit wurde festgehalten, ob sich der Akzent auf diese Position verschiebt und wenn ja, ob an der selben Stelle ein Feature zu finden war (dies geschah durch gleichzeitiges Überprüfen der Featurelokalisierung). Für eine spätere Analyse wurde eine Variable *move.accent.F* erstellt, in der festgehalten wurde, ob sich der Akzent am Feature verschiebt. Im diskutierten Beispiel trat keine Akzentverschiebung auf.

#### 3.4.5.7 Markierung der Featureposition

Für jedes Feature wurde mithilfe der Repräsentation zur Lokalisierung des Features (vgl. Abschnitt 3.4.5.2, S. 12) und eines R-Skriptes (R Development Core Team 2006) markiert, ob das Feature am Anfang der Nummer, in ihrer Mitte oder am Ende der

Nummer steht. Jede Repräsentation einer Realisierung wurde also noch mit *Anfang*, *Mitte* oder *Ende* gekennzeichnet.

Alle Informationen über jedes Vorkommen eines Features wurden in einer ASCII-Tabelle gespeichert um eine effiziente statistische Analyse der Daten mithilfe der R-Software (R Development Core Team 2006) zu ermöglichen.

Da für die verschiedenen Nummerntypen unterschiedliche Ergebnisse erwartet wurden und die Variablenanzahl der statistischen Analyse so gering wie möglich gehalten werden sollte, wurde der Nummerntyp nicht als Variable im Datensatz markiert, sondern es wurden separate Analysedatensätze für die Nummerntypen konstruiert, um evtl. getrennte Analysen zu ermöglichen.

# 4 Modell der Prosodie numerischer Ausdrücke

## 4.1 Motivation und Ziele der Dateninterpretation

Bei der statistischen Analyse sollten verschiedene Fragen beantwortet werden. Das Hauptaugenmerk sollte hier zum einen darauf gelegt werden, ob die Anwesenheit eines bestimmten Features eine bestimmte Gruppierung der Ziffernfolge in Kleingruppen evokiert, und zum andern sollte überprüft werden, ob bestimmte Features das Auftreten einer vom Standard abweichenden Wortgebung hervorrufen. Neben diesen zwei Hauptpunkten sollte außerdem untersucht werden, ob eine bestimmte Wortgebung eine bestimmte Gruppierung auslöst und ob sich bei dem Phänomen „Wiederholung von Einzelziffern“ der Akzent verschiebt.

Zu beantworten sind also folgende Fragen:

1. Gibt es einen Zusammenhang zwischen dem Auftreten eines (bestimmten) Features und
  - a) der Gruppierung der Ziffernfolge?
  - b) der Wortgebung innerhalb der Ziffernfolge?
2. Gibt es Zusammenhänge zwischen einer vom ein-Ziffer-pro-Wort-Standard abweichenden Wortgebung und einer bestimmten Gruppierungsstrategie?
3. Ruft das Feature *Wiederholung* eine Akzentverschiebung hervor?

## 4.2 Einzeluntersuchungen

Um die Daten in Bezug auf Gruppierung, Wortgebung und Akzentuierung zu analysieren, wurden sie, aufgrund ihrer Beschaffenheit bzw. aufgrund der unterschiedlichen Arten der typografischen Stimulipräsentation, je nach Nummerntyp in verschiedene Datensätze aufgeteilt.

### 4.2.1 Gruppierung

#### 4.2.1.1 Bankleitzahlen und Kreditkartennummern

Bankleitzahlen und Kreditkartennummern ist schon aufgrund ihrer durch Leerzeichen vorgegebenen Gliederung in Ziffernblöcke ein bestimmtes Gruppierungsmuster inhärent.

In der vorliegenden Arbeit wurde davon ausgegangen, dass durch diese Vorgabe die Gruppierungsgrenzen zwischen den einzelnen Blöcken so stark sind, dass sie nicht vom Vorhandensein eines Features überschrieben werden können. Daher wurde insbesondere betrachtet, wie und ob die Setzung der intermediären Phrasengrenzen innerhalb der vorgegeben Ziffernblöcke durch das etwaige Vorhandensein eines Features beeinflusst wird. Dies wiederum macht nur für Kreditkartennummern Sinn, da diese Viererblöcke enthalten, in welchen es plausibel ist, eine intermediäre Phrasengrenze zu finden. Bankleitzahlen, für die durch die Kürze der Ziffernblöcke keine Phrasengrenze innerhalb eines Ziffernblockes erwartet wurde, wurden daher nicht in Bezug auf die tonale Gruppierung der Ziffernfolge untersucht.

Bankleitzahlen und Kreditkartennummern wurden daher in zwei separaten Datensätzen (BLZ bzw KR) gespeichert.

#### **4.2.1.2 Kontonummern und Telefonnummern**

Kontonummern und Telefonnummern wurden in keiner bestimmten typografisch vorgegebenen Gruppierung präsentiert. Daher wurden bei diesen nicht nur die intermediären Phrasengrenzen betrachtet, sondern es wurde auch beobachtet, wie die Nummer insgesamt strukturiert war, d.h. wie die Intonationsphrasen gesetzt worden waren.

Da sich die Nummern in ihrer grafischen Präsentation glichen (beide wurden ohne Leerzeichen präsentiert), wurden sie in einem gemeinsamen Datensatz KTOT gespeichert.

#### **4.2.2 Akzentuierung**

Für die Akzentuierung wurden Abweichungen vom Standard-Akzentuierungsmuster, das heißt von einer Akzentuierung der letzten Ziffer in der Phrase betrachtet. Da eine Verschiebung des Akzents nur bei dem Nummernfeature *Wiederholung einer Ziffer* erwartet wurde (und auch nur dort für eine spätere, praktische Anwendung Sinn macht), wurden nur diejenigen Fälle analysiert, bei denen eine Akzentverschiebung an diesem Nummernfeature auftritt.

#### **4.2.3 Wortgebung**

Zur Beantwortung der Frage, ob das Vorhandensein eines Features eine vom Standard (ein Wort pro Ziffer) abweichende Wortgebung auslöst, wurden alle Nummerntypen betrachtet. Die Analyse erfolgte für jeden Datensatz einzeln.

### **4.3 Statistik**

Im Folgenden werden die statistischen Verfahren beschrieben, auf die sich das Modell und somit die später vorgeschlagene Implementierung stützt. Es wird sowohl auf die zu untersuchenden Parameter als auch auf den verwendeten Test eingegangen.



### 4.3.1 Zu testende Variablen

Formal sind die Variablen, deren Abhängigkeit untersucht werden soll, die *Gruppierung*, d.h. die Platzierung der Intonations-, bzw intermediären Phrasengrenzen, sowie eine etwaige *Akzentverschiebung*. Des Weiteren soll der Einfluss der Nummernfeatures auf die Wortgebung, bzw auf die Abweichung der Wortgebung von der Standardwortgebung<sup>1</sup> berücksichtigt werden, das heißt, auch *Wortgebung* muss als abhängige Variable betrachtet werden.

Da davon ausgegangen wurde, dass das Vorhandensein eines Nummernfeatures bestimmte Arten von Gruppierung auslösen, wurde auf Unabhängigkeit der Gruppierung (bzw der Wortgebung) vom Nummernfeature getestet.

Um Interaktionen mit anderen, aufgrund des experimentellen Designs variierenden Parametern auszuschließen, wurde nicht nur der Einfluss des Nummernfeatures, sondern auch derjenige von *Sprecheridentität* (bevorzugen einzelne Sprecher „eigene“ Gruppierungsmuster?), der *Position des Features* in der Ziffernfolge (*Anfang, Mitte, Ende*) und etwaigem *Mehrfachvorkommen* eines Features innerhalb der gleichen Ziffernfolge getestet.

### 4.3.2 Testen auf statistische Unabhängigkeit

Alle zu testenden Variablen waren nominal skaliert<sup>2</sup>, das heißt, jede Ausprägung stellt ein qualitatives Merkmal der jeweiligen Beobachtung dar; es handelte sich also um kategoriale Variablen.

#### 4.3.2.1 $\chi^2$ -Test

Die Unabhängigkeit der jeweiligen Variablen wurde mithilfe des  $\chi^2$ -Testes überprüft. Diese Art der Kontingenzanalyse ermöglicht das Untersuchen nominaler Variablen, weswegen sie in vielen Fällen anwendbar ist. Der Kontingenzanalyse liegen Häufigkeiten, meist dargestellt in Kreuztabellen, zu Grunde, es wird also gezählt, auf wie viele Beobachtungen einer Stichprobe ein Merkmal zutrifft. Das  $\chi^2$ -Verfahren beruht auf der Idee, dass für jede Ausprägung einer gegebenen Variable die Verteilung einer abhängigen Variable gleich ist wie ihre Verteilung in der gesamten Stichprobe.

Bei zwei zweidimensionalen Merkmalen (Tabelle 4.1) bedeutet dies, dass im Falle einer Unabhängigkeit der beiden Variablen das Verhältnis der Anzahl der Beobachtungen, auf die die erste Ausprägung des einen Merkmals zutrifft, und der Anzahl Beobachtungen, auf die die zweite Ausprägung des selben Merkmals zutrifft, in der gesamten Stichprobe ähnlich dem Verhältnis der zwei Ausprägungen des zweiten Merkmals ist.

Das heißt, im Beispiel aus Tabelle 4.1 wird erwartet, dass  $\frac{n_{11}}{n_{21}} \approx \frac{n_{12}}{n_{22}} \approx \frac{n_{1\bullet}}{n_{2\bullet}}$  sowie  $\frac{n_{11}}{n_{12}} \approx \frac{n_{21}}{n_{22}} \approx \frac{n_{\bullet 1}}{n_{\bullet 2}}$ .

---

<sup>1</sup>ein Wort pro Ziffer

<sup>2</sup>Im Falle der Variable *Mehrfachvorkommen* handelt es sich streng genommen um eine Ordinalskala; da jedoch nur eine begrenzte Anzahl von Werten vorhanden war (erstes bis viertes Vorkommen eines Features), wurde auch diese Skala als kategorial betrachtet

Merkmal a	Merkmal b		$\Sigma$
	1	2	
1	$n_{11}$	$n_{12}$	$n_{11} + n_{12} = n_{1\bullet}$
2	$n_{21}$	$n_{22}$	$n_{21} + n_{22} = n_{2\bullet}$
$\Sigma$	$n_{11} + n_{21} = n_{\bullet 1}$	$n_{12} + n_{22} = n_{\bullet 2}$	n

Tabelle 4.1: Kreuztabellierung für die Kontingenzanalyse

Für jede Zelle einer Kreuztabelle mit  $i$  Zeilen und  $j$  Spalten (das heißt mit  $i$  Ausprägungen der einen und  $j$  Ausprägungen der anderen Variable) lässt sich also ein Erwartungswert  $e_{ij}$  bilden, indem die jeweilige Zeilensumme mit der jeweiligen Spaltensumme multipliziert und durch die Stichprobengröße dividiert wird:  $e_{ij} = \frac{n_{i\bullet} \times n_{\bullet j}}{n}$  (vgl. Backhaus u. a. 2006). Die eigentliche Testgröße des  $\chi^2$ -Tests wird dann mithilfe der quadrierten (um Vorzeichenwechsel auszugleichen) Differenz zwischen dem Erwartungswert für jede Zelle der Kreuztabelle und der tatsächlichen Anzahl gebildet:

$$\chi^2 = \sum_{j=1}^j \sum_{i=1}^i \frac{(n_{ij} - e_{ij})^2}{e_{ij}} \quad (\text{Backhaus u. a. 2006}).$$

## 4.4 Ergebnisse

In der vorliegenden Studie wurde jede Variable selbst auf ihre Unabhängigkeit vom Nummernfeature getestet, danach, um ein genaueres Bild der Zusammenhänge zu bekommen, die wechselseitige Abhängigkeit einzelner, aus der Variable erstellter Dummy-Variablen.<sup>3</sup>

Im Folgenden finden sich die Ergebnisse der  $\chi^2$ -Tests, sortiert nach den zu untersuchenden Phänomenen und Datensätzen. Es wurde ein Signifikanzniveau von 5% gewählt, weswegen in der tabellarischen Übersicht über die p-Werte diejenigen Werte unter 0.05 als signifikant (sig) markiert und höhere p-Werte sind durch einen Strich gekennzeichnet sind.

In Abschnitt 4.5 (S. 43) werden die Ergebnisse interpretiert.

### 4.4.1 Gruppierung in intonatorische Einheiten

Die Analyse der Gruppierung erfolgte mithilfe der in 3.4.5.3 vorgestellten Kategorien.

Bei der Analyse von Phrasengrenzen vor bzw. nach einem Feature musste beachtet werden, dass bestimmte Stimuli für bestimmte statistische Auswertungen ausgeschlossen werden mussten, je nach der Position des Features innerhalb der Nummer. Stand das Feature am Ende der Nummer, so sollte diese Beobachtung nicht in die Analyse der Abhängigkeit bzw. Unabhängigkeit von einer Phrasengrenze direkt nach einem Feature (*IP/ip.after.F*) und einem bestimmten Nummernfeature mit aufgenommen werden

<sup>3</sup>Als Dummy-Variablen werden binäre Variablen bezeichnet, die aus den Ausprägungen einer kategorialen Variable erzeugt werden und in denen die Info „Ausprägung trifft zu“ oder „Ausprägung trifft nicht zu“ kodiert ist.

- denn diese Phrasengrenze ist obligatorisch! Der umgekehrte Fall gilt für die Featureposition am Anfang der Nummer und die Gruppierungsstrategien, die Phrasengrenzen nach dem Feature widerspiegeln (*IP/ip.before.F*). Die entsprechenden Beobachtungen wurden daher von der jeweiligen Analyse ausgeschlossen.

Außerdem wurden als zusätzliche zu testende Dummy-Variablen noch die Gruppierungsstrategien *ip.before.F* und *IP.before.F* sowie die entsprechenden Markierungen für nachfolgende Phrasengrenzen zu *bound.before/after.F* zusammengefasst, um etwaige Effekte, ausgelöst durch eine tonale Phrase beliebiger Art, zu erfassen.

#### 4.4.1.1 Kontonummern und Telefonnummern

Für den Datensatz der Kreditkartennummern und Telefonnummern (im Folgenden als KTOT abgekürzt) wurde die Unabhängigkeit der Gruppierung vom Sprecher nachgewiesen, für die anderen getesteten Variablen (*Nummernfeature*, *Featureposition* und *ntes.Vorkommen*) ergaben sich signifikante p-Werte, was auf eine Abhängigkeit der beiden Variablen hindeutet (Tabelle 4.2).

$\chi^2$	Sprecher	Nummernfeature	Featureposition	ntes.Vorkommen
Gruppierung	—	sig	sig	sig
Wortgebung	sig	sig	—	—

Tabelle 4.2: Signifikanzmatrix des  $\chi^2$ -Tests für alle Variablen beim Datensatz KTOT

Die Einzeluntersuchungen der Korrelationen ergaben Hinweise auf Zusammenhänge zwischen jedem der Nummernfeatures (*ab*, *auf*, *gleiche*, *hundert*, *wdh*, *zehn*) und der Gruppierung (Tabelle 4.3).

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
Gruppierung	sig	sig	sig	sig	sig	sig
Wortgebung	sig	sig	sig	sig	—	sig

Tabelle 4.3: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Nummernfeature mit den Response-Variablen beim Datensatz KTOT

Eine genauere Untersuchung der Zusammenhänge ergibt Korrelationen verschiedener Gruppierungsstrategien mit verschiedenen Nummernfeatures. Tabelle 4.4 gibt einen Überblick über die Zusammenhänge, die im Folgenden dargestellt werden.

Da verhindert werden sollte, dass eine etwaige abweichende Wortgebung die Ergebnisse für die verschiedenen Arten der Gruppierung beeinflusst (da innerhalb eines Features, das in einer Phrase („dreimal die ...“) oder in einem Einzelwort („zweihundert“) zusammengefasst wird, normalerweise keine Phrasengrenze realisiert wird (*ip/IP.cuts.F*) und dieses Feature auch häufig innerhalb einer eigenen tonalen Phrase steht (*only.F.in.IP*),

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
ip.after.F	—	—	—	—	—	—
IP.after.F	—	—	—	—	—	—
ip.before.F	—	—	—	—	—	—
IP.before.F	—	—	—	—	—	—
ip.cuts.F	sig	—	sig	—	—	sig
IP.cuts.F	sig	sig	sig	—	sig	sig
no.boundary	—	—	—	—	—	sig
only.F.in.ip	—	—	—	—	—	—
only.F.in.IP	sig	—	sig	sig	sig	—
bound.before.F	—	—	—	—	—	—
bound.after.F	—	—	—	—	—	—

Tabelle 4.4: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit den Ausprägungen von Nummernfeature beim Datensatz KTOT

wurde der Datensatz geteilt und auch Signifikanzmatritzen für nur diejenigen Beobachtungen erstellt, bei denen keine abweichende Wortgebung auftrat (Datensatz KTOT.N, Tabelle 4.5). Sofern dies für die Interpretation relevant ist, das heißt, sofern sich die

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
ip.after.F	—	—	—	—	—	—
IP.after.F	—	—	—	—	—	—
ip.before.F	—	—	—	—	—	—
IP.before.F	—	—	—	—	—	—
ip.cuts.F	sig	—	sig	sig	—	sig
IP.cuts.F	sig	sig	sig	—	sig	sig
no.boundary	—	—	—	—	—	sig
only.F.in.ip	—	—	—	—	—	—
only.F.in.IP	sig	—	sig	—	sig	—
bound.before.F	—	—	—	—	—	—
bound.after.F	—	—	—	—	—	—

Tabelle 4.5: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit den Ausprägungen von Nummernfeature beim Datensatz KTOT.N

Korrelationen unterscheiden, wird im Folgenden auf diese Signifikanzen eingegangen, ansonsten wird der ganze Datensatz betrachtet, da so mehr Beobachtungen in die Analyse eingehen.

**4.4.1.1.1 Absteigende Ziffernfolgen** Aus Tabelle 4.4 und Tabelle 4.5 ist ersichtlich, dass das absteigende Ziffernfolgen (Nummernfeature *ab*) einerseits damit korrelieren, dass nur das Feature innerhalb der jeweiligen intermediären Phrase steht (Gruppierung *only.F.in.ip*) und andererseits damit, dass eine tonale Phrasengrenze innerhalb des Features gesetzt wird (Gruppierung *ip/IP.cuts.F*).

Diese auf den ersten Blick widersprüchlichen Ergebnisse müssen nun daraufhin untersucht werden, ob es sich um eine positive oder eine negative Korrelation der jeweiligen Variablen handelt. Um dies zu ermitteln, müssen Häufigkeiten, das heißt die entsprechenden Kreuztabellen betrachtet werden (Tabelle 4.6). Vergleicht man nun die relative Häufigkeit der Gruppierungsstrategien, also das Verhältnis von Anzahl der Beobachtungen, in denen die jeweilige Gruppierung aufgetreten ist, zu der Anzahl der Beobachtungen, in welchen dies nicht der Fall ist, beim Vorhandensein des Nummernfeatures mit dem bei seinem Nicht-Vorhandensein, so kann man sehen, dass für die beiden Gruppierungsmarkierungen, die anzeigen, dass eine tonale Phrase innerhalb des Features gesetzt wurde (*ip/IP.cuts.F*), die Korrelation positiv ist: in beiden Fällen tritt die Gruppierungsstrategie relativ gesehen häufiger auf, wenn das Nummernfeature *ab* vorhanden ist, als wenn es nicht vorhanden ist ( $7/62 > 13/470$  für *ip.cuts.F*, sowie  $29/40 > 88/395$  für *IP.cuts.F*). Umgekehrt verhält es sich für die Gruppierung *only.F.in.IP*: sie tritt seltener auf, wenn das Nummernfeature *ab* vorhanden ist ( $18/51 < 231/252$ ), es handelt sich also um eine negative Korrelation der beiden Variablen (vgl. zu den Häufigkeiten Tabelle 4.6).

Das bedeutet, dass das Nummernfeature *ab* die Wahrscheinlichkeit einer Phrasengrenze innerhalb des Features erhöht, während es die Wahrscheinlichkeit dafür, dass das Feature allein innerhalb einer Intonationsphrase steht, verringert.

Allerdings sollte man bedenken, dass sich zwar das Verhältnis von Häufigkeit des Auftretens der Gruppierungsstrategie zu ihrem Nicht-Auftreten verändert, wenn das Nummernfeature *ab* in der Nummer vorhanden ist, dass jedoch trotzdem die absoluten Werte zeigen, dass von 69 Fällen, in welchen das Nummernfeature auftrat, in immerhin 62 Fällen keine intermediäre Phrasengrenze und in 40 Fällen keine Intonationsphrasengrenze innerhalb des Features gesetzt wurde.

Dies liegt im experimentellen Design begründet: Da von Default-Gruppierungsmustern, wie sie Baumann u. Trouvain (2001) festgestellt haben, ausgegangen wurde, wurden in der vorliegenden Studie bereits die Stimuli darauf ausgelegt, Effekte, die durch das Vorhandensein der Features evoziert werden, offenzulegen, weswegen keine neutralen Ziffernfolgen ohne eines der Features aufgenommen wurden. Die Features wurden daher in der statistischen Analyse nur untereinander verglichen, nicht jedoch mit einem neutralen Default.

**4.4.1.1.2 Aufsteigende Ziffernfolgen** Das Nummernfeature *auf*, also das Vorhandensein von aufsteigenden Ziffernfolgen in der Nummer, korreliert für die Konto- und Telefonnummern nur mit der Gruppierungsstrategie *IP.cuts.F* (Tabelle 4.4), also damit, dass innerhalb des Features eine Intonationsphrase gesetzt wird. Betrachtet man die Häufigkeiten (Tabelle 4.7), so stellt man fest, dass es sich wiederum um eine positi-

<b>ab</b>	<b>ip.cuts.F</b>		<b>ab</b>	<b>IP.cuts.F</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	470	13	nicht vorhanden	395	88
vorhanden	62	7	vorhanden	40	29

<b>ab</b>	<b>only.F.in.IP</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	252	231
vorhanden	51	18

Tabelle 4.6: Kreuztabellen für das Nummernfeature *ab* und die Gruppierungsstrategien *only.F.in.IP*, *IP.cuts.F* und *ip.cuts.F* im Datensatz KTOT

<b>auf</b>	<b>IP.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	378	78
vorhanden	57	39

Tabelle 4.7: Kreuztabelle für das Nummernfeature *auf* und die Gruppierung *IP.cuts.F* im Datensatz KTOT

ve Korrelation handelt: relativ gesehen tritt die Gruppierung *IP.cuts.F* häufiger auf, wenn das Feature *auf* vorhanden ist ( $39/57 > 78/378$ ), als wenn ein anderes Feature in der Nummer vorkommt. Das Vorhandensein aufeinanderfolgender aufsteigender Ziffernfolgen erhöht also im Vergleich zum Vorhandensein anderer Nummernfeatures die Wahrscheinlichkeit dafür, dass eine Phrasengrenze innerhalb des Features gesetzt wird.

Auch hier jedoch zeigt die isolierte Betrachtung derjenigen Fälle, in welchen das Feature vorkommt, die Tendenz, dass keine Phrasengrenze gesetzt wird: in nur 29 aller 96 Beobachtungen für das Nummernfeature *ab* wird die Gruppierungsstrategie *IP.cuts.F* eingesetzt.

<b>gleiche</b>	<b>ip.cuts.F</b>		<b>gleiche</b>	<b>IP.cuts.F</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	390	20	nicht vorhanden	301	109
vorhanden	142	0	vorhanden	134	8

<b>gleiche</b>	<b>only.F.in.IP</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	246	164
vorhanden	57	85

Tabelle 4.8: Kreuztabellen für das Nummernfeature *gleiche* und die Gruppierungsstrategien *only.F.in.IP*, *ip.cuts.F* und *IP.cuts.F* im Datensatz KTOT

hundert	only.F.in.IP	
	trifft nicht zu	trifft zu
nicht vorhanden	283	215
vorhanden	20	34

Tabelle 4.9: Kreuztabelle für das Nummernfeature *hundert* und die Gruppierung *only.F.in.IP*

**4.4.1.1.3 Gleiche Einzelziffern** Bei gleichen Ziffern sind ebenfalls Korrelationen des Nummernfeatures einerseits mit den Gruppierungsstrategien, die eine Phrasengrenze innerhalb des Features markieren (*ip/IP.cuts.F*) und andererseits damit, dass nur das Feature innerhalb der jeweiligen Intonationsphrase steht (*only.F.in.IP*), zu beobachten (Tabelle 4.4), allerdings stellen sich hier bei einem Vergleich der Verhältnisse von Fällen in denen gleiche Einzelziffern auftreten (Feature *gleiche*), zu Fällen, in denen dies nicht der Fall ist, genau umgekehrte Korrelationen als bei den aufeinanderfolgenden Ziffern heraus: Das Vorhandensein gleicher aufeinanderfolgender Ziffern korreliert positiv damit, dass nur das jeweilige Feature innerhalb der Intonationsphrase steht (85/57 beim Fehlen des Features *gleiche* > 164/246 beim Vorhandensein des Features), wohingegen zu den Gruppierungsmarkierungen, die anzeigen, dass das Feature durch eine tonale Phrasengrenze in zwei verschiedene Einheiten aufgeteilt wurde, eine negative Korrelation besteht (8/134 < 109/301 bzw. 0/142 < 20/390). Hier ist anzumerken, dass der  $\chi^2$ -Test für Fälle von unter fünf Beobachtungen unverlässliche Ergebnisse liefert. Daher wurde die Korrelation mit dem exakten Fisher-Test (vgl. Backhaus u. a. 2006, S. 243) überprüft und mit einem p-Wert < 0.005 bestätigt. Folgen also gleiche Ziffern aufeinander, so erhöht sich die Wahrscheinlichkeit dafür, dass erstens die Ziffern innerhalb ein und der selben Intonationseinheit zu finden sind (negative Korrelation mit *ip/IP.cuts.F*) und dass zweitens sogar keine weitere Ziffer innerhalb der jeweiligen Intonationsphrase steht - dass also die gleichen Ziffern allein eine tonale Einheit darstellen (positive Korrelation mit *only.F.in.IP*).

**4.4.1.1.4 Volle Hunderterzahlen** Das Nummernfeature *hundert*, also eine beliebige Ziffer gefolgt von zwei Nullen, korreliert signifikant mit der Gruppierung *only.F.in.IP*, die anzeigt, dass nur das jeweilige Feature und keine weiteren Ziffern innerhalb der Intonationsphrase stand (Tabelle 4.4). Eine Betrachtung der Häufigkeiten zeigt, dass diese Korrelation positiver Art ist (34/20 > 215/283). Das Nummernfeature *hundert* in der Nummer erhöht also das Verhältnis von Fällen, in denen nur das Feature in der IP steht (Gruppierung *only.F.in.IP*) und Fällen, in welchen dies nicht der Fall ist.

Allerdings ist diese Korrelation nur für den gesamten Datensatz relevant. Betrachtet man den Datensatz reduziert auf diejenigen Fälle, in welchen Standardwortgebung verwendet wurde (Datensatz KTOT.N), so findet man eine Korrelation des Features damit, dass eine intermediäre Phrasengrenze innerhalb des Features gesetzt wird (Gruppierung *ip.cuts.F*, vgl. Tabelle 4.5).

<b>hundert</b>	<b>ip.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	483	15
vorhanden	49	5

Tabelle 4.10: Kreuztabelle fuer das Nummernfeature *hundert* und die Gruppierung *ip.cuts.F* im Datensatz KTOT.N

<b>zehn</b>	<b>ip.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	385	20
vorhanden	147	0

<b>zehn</b>	<b>IP.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	289	116
vorhanden	146	1

Tabelle 4.11: Kreuztabellen für das Nummernfeature *zehn* und die Gruppierungsstrategien *ip.cuts.F* und *IP.cuts.F* beim Datensatz KTOT

In diesem Fall zeigt die Betrachtung der Häufigkeiten (Tabelle 4.10), dass es sich um eine positive Korrelation handelt ( $5/49 > 15/483$ ), das heißt, wenn das Feature *hundert* auftritt und nicht als Hunderterzahl sondern als mehrere Einzelziffern ausgesprochen wird, erhöht sich die Wahrscheinlichkeit dafür, dass zwischen diesen Einzelziffern eine Phrasengrenze realisiert wird.

**4.4.1.1.5 Volle Zehnerzahlen** Wie in Tabelle 4.4 und Tabelle 4.5 ersichtlich ist, korrelieren die vollen Zehnerzahlen, also einzelne Nullen innerhalb der Nummer mit den beiden Gruppierungsstrategien, die eine Phrasengrenze innerhalb des Features anzeigen (*ip/IP.cuts.F*) und damit, dass keine Grenze vor oder nach dem Feature gesetzt wurde (*no.boundary*). Eine genaue Betrachtung der Ziffernfolgen und ihrer Phrasierung zeigt jedoch, dass die Fälle, die vom R-Skript (vgl. Anhang B, S. 68) als *no.boundary* eingestuft wurden, stets am Ziffernende liegen - also doch von einer Phrasengrenze angeschlossen werden. Daher wird dieses Ergebnis im Folgenden nicht berücksichtigt.

Betrachtet man die Häufigkeitstabellen für die beiden Gruppierungsstrategien, die eine Phrasengrenze innerhalb des Features markieren (Tabelle 4.11), so stellt man fest, dass es sich in beiden Fällen um eine negative Korrelation handeln muss: in keinem Fall, in dem das Nummernfeature *zehn* vorkam, wurden diese Gruppierungsstrategien gewählt. Da der für die Signifikanzmatrix angewandte  $\chi^2$ -Test in diesem Fall nicht verlässlich ist (Backhaus u. a. 2006) wurde auch hier das Testergebnis für beide Gruppierungsstrategien mit dem exakten Fisher-Test (vgl. Backhaus u. a. 2006, S. 243) verifiziert. In beiden Fällen ergab sich eine signifikante negative Korrelation mit  $p < 0,005$ .



wdh	IP.cuts.F	
	trifft nicht zu	trifft zu
nicht vorhanden	420	88
vorhanden	15	29

wdh	only.F.in.IP	
	trifft nicht zu	trifft zu
nicht vorhanden	265	243
vorhanden	38	6

Tabelle 4.12: Kreuztabelle für das Nummernfeature *wdh* und die Gruppierungskategorien *IP.cuts.F* und *only.F.in.IP*

**4.4.1.1.6 Wiederholungen von Ziffern** Für das Nummernfeature *wdh*, also für gleiche, nicht aufeinanderfolgende Einzelziffern innerhalb der Nummer, wurden Korrelationen mit dem Auftreten einer Intonationsphrase innerhalb des Features (*IP.cuts.F*) einerseits und mit der alleinigen Gruppierung des Features innerhalb einer Intonationsphrase (*only.F.in.IP*) andererseits gefunden.

Eine Betrachtung der Häufigkeiten (Tabelle 4.12) ergab, dass es sich im Falle vom Auftreten einer IP innerhalb des Features (*IP.cuts.F*) um eine positive, im Falle der alleinigen Gruppierung des Features innerhalb einer Intonationsphrase (*only.F.in.IP*) um eine negative Korrelation handelte ( $29/15 > 88/420$  und  $6/38 < 243/265$ ).

Gleiche, nicht aufeinanderfolgende Einzelziffern innerhalb einer Nummer erhöhen also die Wahrscheinlichkeit, dass zwischen den jeweiligen zum Feature gehörigen Ziffern eine Intonationsphrasengrenze realisiert wird. Außerdem verringern sie die Wahrscheinlichkeit dafür, dass das komplette Feature allein innerhalb einer einzelnen Intonationsphrase steht.

#### 4.4.1.2 Kreditkartennummern

Bei den Kreditkartennummern sind bezüglich der Abhängigkeit bzw Unabhängigkeit der kategorialen Variablen die gleichen Zusammenhänge zu beobachten wie für die Kontonummern und Telefonnummern: die Sprecheridentität spielt keine Rolle, während die anderen Variablen Zusammenhänge aufweisen (Tabelle 4.13).

$\chi^2$	Sprecher	Nummernfeature	Featureposition	ntes.Vorkommen
Gruppierung	—	sig	sig	sig
Wortgebung	sig	sig	—	—

Tabelle 4.13: Signifikanzmatrix des  $\chi^2$ -Tests für alle Variablen beim Datensatz KR

Eine genauere Betrachtung der Korrelationen von Ausprägungen der Variable *Nummernfeature* mit den Ausprägungen von *Gruppierung* zeigt Zusammenhänge von Gruppierungsstrategien mit den Nummernfeatures *gleiche*, *hundert*, *wdh* und *zehn* (Tabelle

4.14). Wie bei den Konto- und Telefonnummern wurde im Falle der Kreditkartennum-

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
ip.after.F	—	—	—	—	—	—
IP.after.F	—	—	—	—	—	—
ip.before.F	—	—	—	—	—	—
IP.before.F	—	—	—	—	—	—
ip.cuts.F	—	—	—	sig	—	sig
IP.cuts.F	—	—	—	—	sig	—
no.boundary	—	—	—	—	—	—
only.F.in.ip	—	—	sig	—	—	sig
only.F.in.IP	—	—	—	—	—	sig
bound.before.F	—	—	—	—	—	—
bound.after.F	—	—	—	—	—	—

Tabelle 4.14: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit den Ausprägungen von Nummernfeature beim Datensatz KR

mern ebenfalls abgeprüft, ob im Datensatz KR.N, also in demjenigen Datensatz, in dem nur die Fälle enthalten sind, in dem die Standard-Wortgebung verwendet wurde, die Korrelationen gleich sind wie im Gesamtdatensatz. Dies war für alle Nummernfeatures bis auf das Nummernfeature *gleiche* der Fall (Tabelle 4.15).

Im Folgenden werden die Ergebnisse der einzelnen  $\chi^2$ -Tests für die Kreditkartennummern vorgestellt, die, wie diejenigen für die Konto- und Telefonnummern, in Abschnitt 4.5 (S. 43) interpretiert werden.

**4.4.1.2.1 Gleiche Einzelziffern** Bei den gleichen Einzelziffern findet man in denjenigen Fällen, in welchen die Nummernfolge auch tatsächlich als einzelne Ziffer ausgesprochen wurde, eine negative Korrelation vom Nummernfeature und der Gruppierung *ip.cuts.F* ( $3/20 < 55/98$ ). Das heißt also, aufeinanderfolgende identische Ziffern verringern die Wahrscheinlichkeit für eine Phrasengrenze zwischen ebendiesen Ziffern im Vergleich zu anderen Nummernfeatures.

**4.4.1.2.2 Volle Zehnerzahlen** Das Vorkommen voller Zehnerzahlen in einer Nummer korrelierte signifikant mit den Gruppierungsstrategien *ip.cuts.F*, *only.F.in.ip* und *only.F.in.IP*. Im ersten Fall handelte es sich um eine negative Korrelation ( $6/36 < 49/69$ ), in den anderen beiden Fällen um eine positive Korrelation. Das heißt, das Nummernfeature *zehn* führt dazu, dass, im Vergleich zu anderen Nummernfeatures, das Feature häufiger allein innerhalb einer tonalen Einheit steht (positive Korrelation zu *only.F.in.ip/IP*) und dass seltener eine intermediäre Phrasengrenze innerhalb des Features gesetzt wird (negative Korrelation zu *ip.cuts.F*).

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
ip.after.F	—	—	—	—	—	—
IP.after.F	—	—	—	—	—	—
ip.before.F	—	—	—	—	—	—
IP.before.F	—	—	—	—	—	—
ip.cuts.F	—	—	sig	sig	—	sig
IP.cuts.F	—	—	—	—	sig	—
no.boundary	—	—	—	—	—	—
only.F.in.ip	—	—	—	—	—	sig
only.F.in.IP	—	—	—	—	—	sig
bound.before.F	—	—	—	—	—	—
bound.after.F	—	—	—	—	—	—

Tabelle 4.15: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit den Ausprägungen von Nummernfeature beim Datensatz KR.N

<b>gleiche</b>	<b>ip.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	98	55
vorhanden	20	3

Tabelle 4.16: Kreuztabelle mit Häufigkeiten für die Gruppierungskategorie *ip.cuts.F* und das Nummernfeature *gleiche* im Datensatz KR

<b>zehn</b>	<b>ip.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	67	49
vorhanden	36	6

<b>zehn</b>	<b>only.F.in.ip</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	111	5
vorhanden	33	9

<b>zehn</b>	<b>only.F.in.IP</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	111	5
vorhanden	31	11

Tabelle 4.17: Kreuztabelle mit Häufigkeiten für die Gruppierungsstrategien *ip.cuts.F*, *only.F.in.IP* und *only.F.in.ip* und das Nummernfeature *zehn* im Datensatz KR.N

Es sei betont, dass die betrachteten Häufigkeiten aus dem Datensatz KR.N stammen, also nur diejenigen Realisierungen beinhalten, in welchen die Ziffern einzeln ausgesprochen wurden. Die beobachteten Effekte entstehen also nicht durch die Wortgebung.

<b>hundert</b>	<b>ip.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	95	42
vorhanden	8	13

Tabelle 4.18: Kreuztabelle mit Häufigkeiten für die Gruppierungskategorie *ip.cuts.F* und das Nummernfeature *hundert* im Datensatz KR.N

**4.4.1.2.3 Volle Hunderterzahlen** Für volle Hunderterzahlen, die innerhalb der Vierergruppen einer Kreditkartennummer zu finden waren, wurde, anders als bei den gleichen aufeinanderfolgenden Ziffern, eine positive Korrelation zwischen dem Nummernfeature (*hundert*) und dem Setzen einer intermediären Phrasengrenze innerhalb des Features (Gruppierung *ip.cuts.F*) festgestellt ( $13/8 > 42/95$ , vgl. Tabelle 4.18, S. 30). Gegenüber den anderen Nummernfeatures wird also häufiger eine intermediäre Phrasengrenze innerhalb des Features gesetzt, wenn es sich bei dem Feature um das Vorkommen einer vollen Hunderterzahl handelt.

<b>wdh</b>	<b>IP.cuts.F</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	137	12
vorhanden	0	9

Tabelle 4.19: Kreuztabelle mit Häufigkeiten für die Gruppierungskategorie *IP.cuts.F* und das Nummernfeature *wdh* im Datensatz KR.N

**4.4.1.2.4 Wiederholung gleicher Einzelziffern** Betrachtet man die Ergebnisse des  $\chi^2$ -Tests für gleiche Einzelziffern, die nicht aufeinander folgen, so stellt man eine Korrelation des Nummernfeatures (also der Variable *wdh*) mit der Gruppierungsstrategie, eine Intonationsphrasengrenze innerhalb des Features zu setzen (Gruppierung *IP.cuts.F*), fest. Eine genauere Betrachtung der Häufigkeiten (vgl. Tabelle 4.19, S. 30) zeigte wiederum, dass der  $\chi^2$ -Test für diese beiden Variablen nicht verlässlich war, da die Zellohäufigkeit für die Zelle „*IP.cuts.F* trifft nicht zu/*wdh* vorhanden“ zu gering war. Daher wurde auch hier der exakte Fisher-Test (vgl. Backhaus u. a. 2006, S. 243) durchgeführt. Er ergab einen sehr kleinen p-Wert, weit unter dem Signifikanzniveau von 0.05.

Die Analyse des Verhältnisses vom Zutreffen von *IP.cuts.F* und dem Nicht-Zutreffen in Abhängigkeit vom Nummernfeature *wdh* zeigt, dass es sich um eine positive Korrelation handelt: während sich das Auftreten von Fällen, in denen eine Intonationsphrase innerhalb des Features realisiert wurde (*IP.cuts.F*) verringert (genauer gesagt auf Null reduziert), wenn das Nummernfeature vorhanden ist, steigt es an, wenn ein anderes Feature als *wdh* vorkommt. Es handelt sich also um eine negative Korrelation.

## 4.4.2 Wortgebung

Um die Korrelation der abweichenden Wortgebung mit bestimmten Features zu überprüfen, wurde zunächst für alle drei Datensätze (Kontonummern und Telefonnummern, Kreditkartennummern, Bankleitzahlen) überprüft, ob eine Abhängigkeit zwischen *Nummernfeature* und *Wortgebung* besteht und ob es weitere Abhängigkeiten geben könnte. Wie in Tabelle 4.2, Tabelle 4.13 und Tabelle 4.20 ersichtlich ist, ist in allen Datensätzen eine signifikante Korrelation zwischen den beiden Variablen zu beobachten.

$\chi^2$	Sprecher	Nummernfeature	Featureposition	ntes.Vorkommen
Gruppierung	—	sig	sig	sig
Wortgebung	sig	sig	sig	sig

Tabelle 4.20: Signifikanzmatrix des  $\chi^2$ -Tests für alle Variablen beim Datensatz BLZ

Um diesen Zusammenhang näher zu beleuchten, wurde für jede Ausprägung der Variablen *Wortgebung* getestet, ob sie mit einem oder mehreren Nummernfeatures korreliert (vgl. Tabelle 4.21, Tabelle 4.22 und Tabelle 4.23).

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
als.10	—	—	sig	—	—	sig
als.100	—	—	—	sig	—	sig
als.mal	—	—	sig	—	—	—
keine.W	sig	sig	—	sig	—	sig

Tabelle 4.21: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Nummernfeature beim Datensatz BLZ

Um bei einer eingehenderen Untersuchung nur die Korrelationen zu betrachten, die tatsächlich aufgrund einer abweichenden Wortgebung entstehen<sup>4</sup>, wurden alle Datensätze aufgeteilt, und nur die Beobachtungen, in welchen eine abweichende Wortgebung auftrat

<sup>4</sup>Viele der signifikanten  $\chi^2$ -Test erklären sich dadurch, dass negative Korrelationen auftreten, weil bestimmte Nummernfeatures nie mit einer bestimmten Wortgebung auftreten. Den Großteil dieser Korrelationen verliert man, wenn man nur die Fälle betrachtet, in denen auch tatsächlich eine abweichende Wortgebung geäußert wurde.

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
als.10	sig	sig	—	—	—	sig
als.100	—	sig	—	sig	—	sig
als.mal	—	—	sig	—	—	—
keine.W	sig	sig	—	sig	sig	—

Tabelle 4.22: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Nummernfeature beim Datensatz KTOT

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
als.10	—	—	—	—	—	sig
als.100	—	—	—	sig	—	—
keine.W	—	sig	—	—	—	sig

Tabelle 4.23: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Nummernfeature beim Datensatz KR

(Datensätze BLZ.W, KTOT.W und KR.W) wurden in die Analyse der Zusammenhänge zwischen verschiedenen Nummernfeatures und den in 3.4.4 beschriebenen Wortgebungs-kategorien *als.10*, *als.100* und *als.mal* einbezogen. Die Signifikanzmatritzen für diese  $\chi^2$ -Tests finden sich in Tabelle 4.24, Tabelle 4.25 und Tabelle 4.26.<sup>5</sup>

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
Gruppierung	—	—	sig	—	sig	sig
Wortgebung	—	—	sig	sig	—	sig

Tabelle 4.24: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Nummernfeature mit den Response-Variablen beim Datensatz BLZ.W

Die Tabellen für die Ausprägungen der kategorialen Variablen *Nummernfeature* und *Wortgebung* weisen in den verschiedenen Datensätzen ein relativ homogenes Bild auf: Tabelle 4.29, Tabelle 4.27 und Tabelle 4.28 zeigen, dass für alle Datensätze signifikante Korrelationen zwischen der verwendeten Wortgebung und vollen Hunderter- und Zehnerzahlen innerhalb der Ziffernfolge gefunden werden (*hundert* bzw. *zehn*), und - außer bei den Kreditkartennummern - auch signifikante p-Werte für eine wahrscheinliche Abhängigkeit zwischen der Wortgebung und gleichen aufeinanderfolgenden Einzelziffern (*gleiche*).

<sup>5</sup> Bei Tabelle 4.25 und Tabelle 4.26 fällt auf, dass nicht alle Dummy-Variablen getestet wurden. Dies liegt daran, dass der  $\chi^2$ -Test nur sinnvoll ist, wenn die Werte der Variable mindestens zwei verschiedene Werte annehmen. War dies nicht der Fall, was gerade bei den Datensätzen, die nur abweichende

$\chi^2$	ab	gleiche	hundert	wdh	zehn
Gruppierung	—	—	sig	—	—
Wortgebung	—	sig	sig	—	sig

Tabelle 4.25: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Nummernfeature mit den Response-Variablen beim Datensatz KTOT.W

$\chi^2$	hundert	zehn
Gruppierung	—	—
Wortgebung	sig	sig

Tabelle 4.26: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Nummernfeature mit den Response-Variablen beim Datensatz KR.W

$\chi^2$	ab	gleiche	hundert	wdh	zehn
als.10	—	—	sig	—	sig
als.100	—	—	sig	—	sig
als.mal	—	sig	—	—	sig

Tabelle 4.27: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Nummernfeature beim Datensatz KTOT.W

$\chi^2$	hundert	zehn
als.10	sig	sig
als.100	sig	sig

Tabelle 4.28: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Nummernfeature beim Datensatz KR.W

$\chi^2$	ab	auf	gleiche	hundert	wdh	zehn
als.10	—	—	sig	sig	—	sig
als.100	—	—	—	sig	—	sig
als.mal	—	—	sig	—	—	—

Tabelle 4.29: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Nummernfeature beim Datensatz BLZ.W

**BLZ:**

gleiche	als.10		gleiche	als.mal	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	93	33	nicht vorhanden	123	3
vorhanden	34	2	vorhanden	28	8

**KTOT:**

gleiche	als.mal	
	trifft nicht zu	trifft zu
nicht vorhanden	410	0
vorhanden	132	10

Tabelle 4.30: Kreuztabelle mit Häufigkeiten für das Nummernfeature *gleiche*

Im Folgenden werden die Ergebnisse der  $\chi^2$ -Tests für die verschiedenen Nummernfeatures und die Ausprägungen der Variable *Wortgebung* in den verschiedenen Datensätzen, in denen die Wortgebung vom Standard abwich, dargestellt. Da die Ergebnisse relativ homogen waren, werden für jedes Feature alle Datensätze auf einmal präsentiert.

**4.4.2.1 Gleiche Einzelziffern**

Für die *gleiche*, aufeinanderfolgende Einzelziffern wurde nur bei den Kontonummern und den Bankleitzahlen ein Effekt beobachtet (vgl. Tabelle 4.25, Tabelle 4.24, Tabelle 4.26). Eine genauere Analyse der Zusammenhänge zeigt, dass in beiden Datensätzen die Korrelation dazu, dass das Feature nicht als mehrere Einzelziffern sondern als Phrase ausgesprochen wurde („x-mal die ...“, also Wortgebungsstrategie *als.mal*) positiv war ( $8/28 > 3/123$  und  $10/132 > 0/410$  vgl. (Tabelle 4.30)), während die im Datensatz BLZ zu beobachtende Korrelation von gleichen, aufeinanderfolgenden Einzelziffern mit mit denjenigen Fällen, in welchen die Ziffern als Zehnerzahl („dreiunddreißig“, Wortgebungskategorie *als.10*) ausgesprochen wurden, negativ ist ( $2/34 < 33/93$  vgl. Tabelle 4.30).

Das heißt also, durch das Vorhandensein des Nummernfeatures *gleiche* in einer Bankleitzahl, Telefon- oder Kontonummer, wurde das Verhältnis von Ziffernfolgen, die als Phrase ausgesprochen wurden, zu als Einzelziffern ausgesprochenen Ziffernfolgen erhöht.

Ein Blick auf die Häufigkeiten in Tabelle 4.30 zeigt, dass im Falle der Bankleitzahlen in acht von 36 Fällen, in welchen das Nummernfeature *gleiche* vorkam, die entsprechenden Ziffern als Phrase (*als.mal*) realisiert wurden. Das heißt in 22% aller Vorkommnisse wurden aufeinanderfolgende gleiche Einzelziffern nicht in der Standardwortgebung, sondern als Phrase realisiert. Bei den Konto und Telefonnummern sind es - trotz der statistischen Signifikanz - nur 7% der Fälle.

---

Wortgebung enthalten, häufig vorkommt, da es sich bei ihnen um relativ kleine Stichproben handelt, wurde die Variable nicht abgetestet.



**BLZ.W:**

hundert	als.10		hundert	als.100	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	13	35	nicht vorhanden	46	2
vorhanden	9	0	vorhanden	0	9

**KTOT.W:**

hundert	als.10		hundert	als.100	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	17	45	nicht vorhanden	55	7
vorhanden	20	7	vorhanden	7	20

**KR.W:**

hundert	als.10		hundert	als.100	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	0	12	nicht vorhanden	12	0
vorhanden	3	3	vorhanden	3	3

Tabelle 4.31: Kreuztabellen mit Häufigkeiten für das Nummernfeature *hundert* und die Wortgebungskategorien *als.10* und *als.100*

**4.4.2.2 Volle Hunderterzahlen**

Für volle Hunderterzahlen wurde in allen drei Datensätzen eine Korrelation des Features mit der Wortgebungsstrategie, die Ziffern als Zehnerzahl und mit der, die Ziffern als Hunderterzahl auszusprechen (*als.10* bzw. *als.100*) beobachtet Tabelle 4.32.

Im ersten Fall handelt es sich in allen Datensätzen mit ausschließlich abweichender Wortgebung um eine negative Korrelation, da die Nummern im Vergleich zu anderen Arten der Wortgebung (außer der Standardwortgebung) seltener bzw. nie als volle Zehnerzahl ausgesprochen werden, wenn die Ziffern der Nummernfolge eine volle Hunderterzahl bilden und die Nummer in abweichender Wortgebung realisiert wurde (Tabelle 4.31).

Allerdings tritt hier ein interessanter Gegensatz auf, wenn man die Korrelationen in den Gesamtdatensätzen KTOT, BLZ und KR, also in jeweils der kompletten Stichprobe betrachtet. In diesen Fällen stellt man fest, dass dann für Konto- und Telefonnummern sowie für Kreditkarten eine positive Korrelation zu beobachten ist: das heißt, das Nummernfeature löste oft genug eine Aussprache in Hunderterzahl aus, um den Unterschied zu anderen Nummernfeatures signifikant werden zu lassen (Tabelle 4.32). Dies erklärt sich durch die Fälle, in welchen der Sprecher eine Aussprache der Art „zwanzig null acht“ wählte.

Im zweiten Fall hingegen, also im Fall der Korrelation von Hunderterzahlen innerhalb der Nummer mit als Hunderterzahlen ausgesprochenen Realisierungen (*als.100*), lässt sich in allen Datensätzen beobachten, dass sich das Verhältnis von Fällen, in welchen die Ziffernfolge als Hunderterzahl ausgesprochen wird zu denen, in welchen dies nicht der Fall ist, deutlich erhöht, sobald das Nummernfeature *hundert* in der Nummer vorhanden ist ( $9/0 > 2/151$ ,  $20/34 > 7/461$  sowie  $3/24 > 0/149$ , vgl. Tabelle 4.32). Allerdings muss an dieser Stelle bemerkt werden, dass alle Tests für Häufigkeitstabellen, in denen die

**BLZ:**

<b>hundert</b>	<b>als.10</b>		<b>hundert</b>	<b>als.100</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	118	35	nicht vorhanden	151	2
vorhanden	9	0	vorhanden	0	9

**KTOT:**

<b>hundert</b>	<b>als.10</b>		<b>hundert</b>	<b>als.100</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	453	45	nicht vorhanden	491	7
vorhanden	47	7	vorhanden	34	20

**KR:**

<b>hundert</b>	<b>als.10</b>		<b>hundert</b>	<b>als.100</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	137	12	nicht vorhanden	149	0
vorhanden	24	3	vorhanden	24	3

Tabelle 4.32: Kreuztabellen mit Häufigkeiten für das Nummernfeature *hundert* und die Wortgebungskategorien *als.10* und *als.100*

Zellhäufigkeit von mindestens einer Zelle kleiner fünf war, der jeweilige  $\chi^2$ -Test mithilfe des exakten Fisher-Tests (vgl. Backhaus u. a. 2006, S. 243) überprüft wurde und dass nur für die Bankleitzahlen und die Konto- und Telefonnummern die Ergebnisse bezüglich des p-Werts unter dem angelegten Signifikanzniveau bestätigt wurden. Das heißt also, dass wiederum bei den Kreditkartennummern kein signifikanter Effekt gefunden werden konnte, wengleich auch die eine zu den anderen beiden Datensätzen analoge Tendenz in den Häufigkeitstabellen zu beobachten ist.

Was die prozentuale Vorkommenshäufigkeit von Fällen, in welchen die vollen Hunderterzahlen in der Nummer auch als Hunderterzahlen ausgesprochen wurden im jeweils ganzen Datensatz betrifft, so beobachtet man für die Konto- und Telefonnummern, dass in 37% aller Fälle von Nummern, in denen das Nummernfeature *hundert* vorhanden war, die Ziffern als Hunderterzahl ausgesprochen wurden; bei den Bankleitzahlen war dies sogar für alle Vorkommen des Nummernfeatures der Fall - allerdings handelte es sich hier absolut betrachtet auch um weniger Fälle, da jeder Sprecher nur eine Bankleitzahl, die das Nummernfeature *hundert* trug, realisieren mussten, bei den Konto- und Telefonnummern hingegen waren es je drei Fälle.

#### 4.4.2.3 Volle Zehnerzahlen

Bei der Korrelationen von vollen Zehnerzahlen innerhalb der Nummer (Nummernfeature *zehn*) mit einer Aussprache als Hunderterzahl (Feature *hundert*) handelt es sich um eine negative Korrelation: das Nummernfeature verringert die Wahrscheinlichkeit dafür, dass die Ziffern als Hunderterzahl ausgesprochen werden ( $0/54 < 11/97, 0/54 < 3/119$  und  $2/145 < 24/380$  vgl. Tabelle 4.33).

**BLZ:**

<b>zehn</b>	<b>als.100</b>		<b>zehn</b>	<b>als.10</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	97	11	nicht vorhanden	97	11
vorhanden	54	0	vorhanden	30	24

**KR:**

<b>zehn</b>	<b>als.100</b>		<b>zehn</b>	<b>als.10</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	119	3	nicht vorhanden	119	3
vorhanden	54	0	vorhanden	42	12

**KTOT:**

<b>zehn</b>	<b>als.100</b>		<b>zehn</b>	<b>als.10</b>	
	trifft nicht zu	trifft zu		trifft nicht zu	trifft zu
nicht vorhanden	380	25	nicht vorhanden	382	23
vorhanden	145	2	vorhanden	118	29

<b>zehn</b>	<b>als.mal</b>	
	trifft nicht zu	trifft zu
nicht vorhanden	395	10
vorhanden	147	0

Tabelle 4.33: Kreuztabellen mit Häufigkeiten für das Nummernfeature *zehn* in den kompletten Datensätzen

zehn	move.accent.F	
	trifft nicht zu	trifft zu
nicht vorhanden	112	10
vorhanden	51	3

Tabelle 4.34: Kreuztabelle für das Nummernfeature *zehn* und die Variable *move.accent.F* beim Datensatz KTOT.N

Gleiches gilt auch für die Korrelation zur „Phrasenaussprache“ (*als.mal*) im Datensatz KTOT.

Korrelationen zu einer Aussprache als Zehnerzahl hingegen sind in allen Fällen positiv ( $24/30 > 11/97$ ,  $12/42 > 3/119$  und  $29/118 > 23/382$  vgl. Tabelle 4.33). Ein Vorhandensein des Nummernfeatures *zehn* begünstigt also eine Aussprache als Zehnerzahl - bei allen Nummerntypen.

### 4.4.3 Akzentuierung

Um eine Akzentverschiebung zu untersuchen, wurden die Daten anfangs nur hinsichtlich des Nummernfeatures *wdh* ausgewertet, da nur hier eine Verschiebung des Akzentes erwartet wurde (vgl. Abschnitt 3.4.5.6, S. 15).

Allerdings ergab der  $\chi^2$ -Test keine signifikante Korrelation, obwohl es durchaus Fälle in den Daten gab, in welchen der Akzent an der Position des Features verschoben wurde. Eine detaillierte Betrachtung der Daten zeigte, dass die meisten Fälle von Akzentverschiebung beim Nummernfeature *zehn* auftraten. Der  $\chi^2$ -Test ergab eine signifikante Korrelation des Features mit der Variable *move.accent.F*, sowohl im Datensatz KTOT ( $p = 0.01566$ ), als auch im Datensatz KTOT.N ( $p = 0.004097$ ), also bei denjenigen Fällen in welchen die Akzentverschiebung nicht dadurch entstanden sein kann, dass mehrere Ziffern zu einem Wort zusammengefasst wurden.

In den anderen Datensätzen (KR und BLZ) traten für eine Akzentverschiebung weder mit *zehn* noch mit *wdh* signifikante Korrelationen auf.

Die Untersuchung der Häufigkeiten (Tabelle 4.34) ergab einen positiven Zusammenhang zwischen dem Nummernfeature *zehn* und der Variable *move.accent.F* ( $16/100 > 18/329$ ). Das heißt also, durch das Vorhandensein des Nummernfeatures *zehn* wird innerhalb der entsprechenden Ziffernfolge häufiger der an der Akzent verschoben, als dies für die anderen Nummernfeatures der Fall ist.

### 4.4.4 Korrelationen der untersuchten Parameter zu anderen Variablen

Im folgenden Abschnitt finden sich die Korrelationen zwischen den Response-Variablen *Gruppierung* und *Wortgebung* und anderen, sie evtl. beeinflussenden, Parametern. Die aufgelisteten Untersuchungen soll nur einen Überblick über die möglichen Einflussfaktoren geben, nicht aber eine vollständige Analyse dieser Einflüsse. Es werden nur die Korrelationen innerhalb aller Datensätze angegeben, nicht die Analyse der einzelnen

Datensätze, da eine detaillierte Untersuchung der Korrelationen den Rahmen dieser Arbeit sprengen würde.

#### 4.4.4.1 Gruppierung

Was die Korrelation der Gruppierung zu anderen Variablen anbelangt, so wurden für diese Überlegung wieder nur die Datensätze betrachtet, die die Fälle enthalten, in welchen keine abweichende Wortgebung verwendet wurde. Der Hintergrund hierfür ist, dass diejenigen tonalen Phrasengrenzen ausgeschlossen werden sollten, die „automatisch“ entstehen, weil lange Wörter wie „dreihundert“ oder Phrasen wie „dreimal die Sechs“ sehr wahrscheinlich allein in einer tonalen Phrase stehen.

**4.4.4.1.1 Korrelation mit Featureposition** Die Featureposition scheint ein relevanter Faktor zu sein, wenn es um die Gruppierung der Nummer geht. Vor allem die Gruppierungsstrategien, die angeben, ob das Feature allein innerhalb eine Phrase steht, oder ob es durch eine Phrasengrenze in zwei Gruppen aufgeteilt wird, stehen in Abhängigkeit zu den verschiedenen Positionen des Features (Tabelle 4.35).

$\chi^2$	Anfang	Ende	Mitte
ip.after.F	—	—	—
IP.after.F	—	—	—
ip.before.F	—	—	—
IP.before.F	—	—	—
ip.cuts.F	sig	—	sig
IP.cuts.F	—	sig	—
no.boundary	sig	sig	sig
only.F.in.ip	—	sig	—
only.F.in.IP	sig	sig	sig
bound.before.F	—	—	—
bound.after.F	—	—	—

Tabelle 4.35: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit den Ausprägungen von Featureposition beim Datensatz ALL.N

**4.4.4.1.2 Korrelation mit einzelnen Sprechern** Wie in Tabelle 4.36 ersichtlich ist, gibt es Sprecher, die deutlich bestimmte Gruppierungsstrategien bevorzugen. Diese Erkenntnis deckt sich mit den Beobachtungen von Baumann u. Trouvain (2001). Eine detailliertere Untersuchung zeigte, dass dies bei den Konto- und Telefonnummern für zwei Sprecher (SP-ba und SP-sn), bei den Kreditkartennummern für drei Sprecher der Fall war (Sp-hm, SP-sa und SP-sn).

$\chi^2$	SP-ba	SP-cs	SP-fw	SP-gk	SP-hm	SP-ks	SP-mw	SP-sa	SP-sn
ip.after.F	sig	—	—	—	—	—	—	—	—
IP.after.F	—	—	—	—	—	—	—	—	sig
ip.before.F	—	—	—	—	—	—	—	—	—
IP.before.F	—	—	—	—	—	—	—	—	—
ip.cuts.F	—	—	—	—	sig	—	—	—	—
IP.cuts.F	—	—	—	—	—	—	—	—	sig
no.boundary	—	—	—	—	—	—	—	sig	—
only.F.in.ip	sig	—	—	—	—	—	—	—	—
only.F.in.IP	sig	—	—	—	—	—	—	—	—
bound.before.F	—	—	—	—	—	—	—	—	—
bound.after.F	—	—	—	—	—	—	—	—	—

Tabelle 4.36: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit den verschiedenen Sprechern beim Datensatz ALL.N

**4.4.4.1.3 Korrelation mit Mehrfachvorkommen** Eine deutliche Sprache spricht die Signifikanzmatrix für die Abhängigkeit der Gruppierung von etwaigen Mehrfachvorkommen des gleichen Nummernfeatures innerhalb einer Nummer: fast alle Gruppierungsstrategien weisen Korrelationen zu jeder der Ausprägungen von *Mehrfachvorkommen* auf. Eine genauere Analyse der Zusammenhänge zeigt, dass nur die Gruppierungen *IP.after.F*

$\chi^2$	erstes	zweites	drittes	viertes
ip.after.F	—	—	—	—
IP.after.F	sig	sig	sig	sig
ip.before.F	sig	sig	sig	sig
IP.before.F	sig	sig	sig	sig
ip.cuts.F	sig	sig	sig	sig
IP.cuts.F	sig	sig	sig	sig
no.boundary	—	—	—	—
only.F.in.ip	sig	sig	sig	sig
only.F.in.IP	—	—	—	—
bound.before.F	sig	sig	sig	sig
bound.after.F	sig	sig	sig	sig

Tabelle 4.37: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Gruppierung mit verschiedenen Mehrfachvorkommen beim Datensatz ALL.N

und *IP.cuts.F* positiv mit dem Mehrfachvorkommen korrelieren, alle anderen negativ.

Eine genaue Kontrolle über diese Zusammenhänge sollte in einem potentiellen Folgeexperiment schon beim experimentellen Design angestrebt werden.

#### 4.4.4.2 Wortgebung

Für die Wortgebung wurde, anders als bei der Betrachtung der zusätzlichen Korrelationen der Variable Gruppierung, jeweils der komplette Datensatz betrachtet.

**4.4.4.2.1 Korrelation mit Featureposition** Die Korrelationen zu einer bestimmten Featureposition zeigten, dass bei Kontonummern und Telefonnummern die Wortgebung unabhängig von der Featureposition gewählt wird (Tabelle 4.38), wohingegen bei den Bankleitzahlen und den Kreditkartennummern Korrelationen zu beobachten sind (vgl. Tabellen 4.39 und 4.43, 43).

$\chi^2$	Anfang	Ende	Mitte
als.10	—	—	—
als.100	—	—	—
als.mal	—	—	—
keine.W	—	—	—

Tabelle 4.38: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Featureposition beim Datensatz KTOT

$\chi^2$	Anfang	Ende	Mitte
als.10	—	—	—
als.100	sig	—	sig
als.mal	—	—	—
keine.W	sig	—	sig

Tabelle 4.39: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Featureposition beim Datensatz BLZ

$\chi^2$	Anfang	Ende	Mitte
als.10	—	—	—
als.100	—	sig	—
keine.W	—	—	—

Tabelle 4.40: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den Ausprägungen von Featureposition beim Datensatz KR

Eine genauere Analyse der Zusammenhänge zeigt bei den Bankleitzahlen positive Korrelationen zwischen *Mitte* und *keine.W*, sowie zwischen *Anfang* und *als.100*. Die

anderen beiden Korrelationen sind negativ. Das heißt also, dass Hunderterzahlen vor allem am Anfang der Nummer als Hunderterzahlen ausgesprochen werden, wohingegen in der Mitte der Nummer eher keine abweichende Wortgebung stattfindet. Grund hierfür könnte sein, dass am Anfang der Nummer Features leichter erkannt werden als in der Mitte.

Bei den Kreditkartennummern hingegen handelt es sich bei der Korrelation zwischen *Ende* und *als.100* um eine negative Korrelation. Dies könnte allerdings darin begründet liegen, dass ein dreistelliges Feature innerhalb einer Kreditkartennummer die den Kreditkartennummern inhärente Default-Gruppierung verletzen würde (vgl. Abschnitt 4.5.2.2, S. 48).

**4.4.4.2.2 Korrelation mit einzelnen Sprechern** Was die Korrelationen zu einzelnen Sprechern anbelangt, so zeigte sich ganz deutlich, dass bestimmte Sprecher Präferenzen für eine bestimmte Wortgebungsstrategie zeigen.

$\chi^2$	SP-ba	SP-cs	SP-fw	SP-gk	SP-hm	SP-ks	SP-mw	SP-sa	SP-sn
als.10	—	sig	sig	—	—	sig	—	—	—
als.100	—	—	—	—	—	—	—	—	—
als.mal	—	—	—	—	—	—	sig	—	—
keine.W	—	sig	—	—	—	—	—	—	—

Tabelle 4.41: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den verschiedenen Sprechern beim Datensatz BLZ

Vor allem im Datensatz KTOT, also bei den Konto- und Telefonnummern, waren Korrelationen bestimmter Wortgebungsstrategien zu bestimmten Sprechern zu beobachten (Tabelle 4.42), doch auch die anderen beiden Datensätze wiesen mehrere Korrelationen zu einzelnen Sprechern auf (vgl. Tabelle 4.41 und Tabelle 4.43).

$\chi^2$	SP-ba	SP-cs	SP-fw	SP-gk	SP-hm	SP-ks	SP-mw	SP-sa	SP-sn
als.10	—	sig	—	sig	—	sig	sig	—	—
als.100	—	—	—	—	—	—	—	—	—
als.mal	sig	—	—	—	sig	—	—	—	—
keine.W	sig	sig	—	—	—	—	sig	sig	—

Tabelle 4.42: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den verschiedenen Sprechern beim Datensatz KTOT

**4.4.4.2.3 Korrelation mit Mehrfachvorkommen** Für etwaige Korrelationen der Wortgebung zu einem Mehrfachvorkommen des jeweiligen Features, also dazu, ob das Feature



$\chi^2$	SP-ba	SP-cs	SP-fw	SP-gk	SP-hm	SP-ks	SP-mw	SP-sa	SP-sn
als.10	—	—	—	sig	—	—	sig	—	—
als.100	—	—	—	—	—	—	—	sig	—
keine.W	—	—	—	sig	—	—	sig	—	—

Tabelle 4.43: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit den verschiedenen Sprechern beim Datensatz KR

das erste, zweite oder gar dritte Feature dieser Art innerhalb der Nummer war, ließ sich beobachten, dass nur bei den Bankleitzahlen und den Konto- und Telefonnummern, nicht aber bei den Kreditkartennummern Korrelationen zu beobachten waren (vgl. Tabelle 4.44, Tabelle 4.45 und Tabelle 4.46). Im Falle der Bankleitzahlen handelt es sich um positive Korrelationen, bei den Kreditkartennummern sind die Korrelationen negativer Natur.

$\chi^2$	erstes	zweites	drittes	viertes
als.10	—	—	—	—
als.100	sig	sig	sig	sig
als.mal	—	—	—	—
keine.W	—	—	—	—

Tabelle 4.44: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit verschiedenen Mehrfachvorkommen beim Datensatz BLZ

$\chi^2$	erstes	zweites	drittes	viertes
als.10	sig	sig	sig	sig
als.100	—	—	—	—
als.mal	—	—	—	—
keine.W	—	—	—	—

Tabelle 4.45: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit verschiedenen Mehrfachvorkommen beim Datensatz KTOT

## 4.5 Zusammenfassende Interpretation

Im Folgenden werden die Ergebnisse aus Abschnitt 4.4 interpretiert. Hierbei werden die Resultate für alle Nummerntypen (Bankleitzahlen, Telefon- und Kontonummern, Kredit-

$\chi^2$	erstes	zweites	drittes	viertes
als.10	—	—	—	—
als.100	—	—	—	—
keine.W	—	—	—	—

Tabelle 4.46: Signifikanzmatrix des  $\chi^2$ -Tests für die Ausprägungen von Wortgebung mit verschiedenen Mehrfachvorkommen beim Datensatz KR

kartennummern) gemeinsam besprochen, um einen Überblick über den Gesamteinfluss der Nummernfeatures auf die Realisierung der jeweiligen Ziffernfolge zu geben.

## 4.5.1 Gruppierung in intonatorische Einheiten

Der folgende Abschnitt befasst sich mit dem Einfluss der verschiedenen Nummernfeatures auf die tonale Gruppierung der Ziffernfolge. Diejenigen Features, deren Einfluss sich als signifikant erwiesen hat (vgl. Abschnitt 4.4, S. 20), werden einzeln dargestellt.

### 4.5.1.1 Absteigende Ziffernfolgen

Eine absteigende Ziffernfolge innerhalb einer **Telefon- oder Kontonummer** beeinflusst die Gruppierung der Nummer dahingehend, dass die absteigenden Ziffern seltener als andere Features in eine tonalen Phrase gruppiert werden: Phrasengrenzen innerhalb des Features treten signifikant häufiger auf, wenn es sich um eine absteigende Ziffernfolge handelt, als wenn ein anderes Nummernfeature in der Ziffernfolge zu finden ist (positive Korrelation vom Feature *ab* zu den Gruppierungsstrategien *ip.cuts.F* und *IP.cuts.F*, vgl. Abschnitt 4.4.1.1.1, S. 22).

Des Weiteren bilden absteigende Ziffern, die aufeinander folgen, seltener allein eine tonale Phrase als andere Nummernfeatures (negative Korrelation vom Feature *ab* zur Gruppierung *only.F.in.IP*, vgl. Abschnitt 4.4.1.1.1, S. 22).

Das heißt für absteigende Ziffernfolgen sind die Ergebnisse unerwartet, da sie genau gegensätzlich zu den durch Baumann u. Trouvains (2001) Studie erwarteten Ergebnisse sind: absteigende Ziffern werden nicht nur nicht häufiger, sondern sogar seltener als andere Nummernfeatures in einer einzelnen Intonationseinheit allein realisiert und es wird häufiger eine Phrasengrenze (*ip* oder *IP*) innerhalb der Folge gesetzt.

Es sei aber nochmals an die absoluten Werte erinnert: betrachtet man nur diejenigen Beobachtungen, in welchen eine absteigende Folge auftritt, so wurden häufiger keine Phrasengrenzen gesetzt (in 89% bzw. 57% aller Fälle von absteigenden Ziffernfolgen), als Phrasengrenzen auftraten. Dieses Ergebnis entspricht genau dem Ergebnis von Baumann u. Trouvain (2001), die ebenfalls nur deskriptiv prozentuale Häufigkeiten angeben und feststellen, dass in 60% aller Fälle von Nummern mit aufeinanderfolgenden Ziffern diese innerhalb der gleichen Intonationseinheit zu finden sind.

Für **Kreditkartennummern** zeigte sich keine Korrelation von absteigenden Ziffernfolgen und der Gruppierung der Nummer, in der diese vorkamen.

### 4.5.1.2 Aufsteigende Ziffernfolgen

Auch für aufsteigende Ziffernfolgen innerhalb einer **Konto- oder Telefonnummer** lässt sich ein unerwarteter Zusammenhang beobachten: treten aufeinanderfolgende aufsteigende Ziffern innerhalb der Nummer auf, wird häufiger eine Intonationsphrase innerhalb der Folge gesetzt als bei anderen Nummernfeatures (positive Korrelation des Features *auf* mit der Gruppierung *IP.cuts.F*, vgl. Abschnitt 4.4.1.1.2, S. 23).

Allerdings ist der Effekt, den aufsteigende Ziffernfolgen auslösen, insofern schwächer als der von den absteigenden Folgen ausgelöste, als er nur die Intonationsphrasen betrifft, während bei den absteigenden Ziffern auch intermediäre Phrasen häufiger gesetzt sind. Da jedoch eine Intonationsphrase eine stärkere Zäsur innerhalb der Ziffernfolge bedeutet, ist dies kein wesentlicher Unterschied für eine spätere praktische Anwendung der Ergebnisse.

Auch hier gilt bezüglich der Vergleichbarkeit zu Baumann u. Trouvains Studie, dass in immerhin 60% derjenigen Fälle, in welchen eine aufsteigende Ziffernfolge vorkam, keine Phrasengrenze gesetzt wurde, das heißt, das Feature innerhalb einer einzigen Intonationseinheit zu finden war. Gleiches berichten auch Baumann u. Trouvain (2001).

Es gibt keine Hinweise auf Zusammenhänge einer aufsteigenden Ziffernfolge mit weiteren Gruppierungsstrategien.

Wie auch bei den absteigenden Ziffernfolgen konnte für **Kreditkartennummern** keine Korrelation zwischen einer oder mehreren Gruppierungsstrategien und dem Nummernfeature gefunden werden.

### 4.5.1.3 Gleiche Einzelziffern

Für gleiche Ziffern entsprechen im Datensatz der **Konto- und Telefonnummern** die Ergebnisse der statistischen Tests den Erwartungen: die Ziffern werden seltener als bei anderen Nummernfeatures durch eine Phrasengrenze voneinander getrennt. Außerdem machen sie häufiger die Phrase allein aus, es ist also häufiger genau vor und nach dem Feature eine Phrasengrenze zu finden.

Um die Ergebnisse mit den Ergebnissen von Baumann u. Trouvain (2001) vergleichen zu können, sei hier dennoch ein Blick auf den prozentualen Anteil der jeweiligen Gruppierungsstrategien an der Gesamtzahl der Nummern mit gleiche Ziffern geworfen. Es wurde in keinem der 144 Fälle von gleichen aufeinanderfolgenden Ziffern eine intermediäre Phrasengrenze und in nur acht Fällen (5%) eine Intonationsphrasengrenze zwischen die jeweiligen Ziffern gesetzt. Außerdem waren die gleichen Ziffern in 69% der Fälle die einzigen Ziffern innerhalb der jeweiligen Intonationphrase.

Wie bei den Konto- und Telefonnummern ist auch bei den **Kreditkartennummern** der Effekt zu beobachten, dass gleiche, aufeinanderfolgende Ziffern in der Nummer dazu führen, dass an dieser Stelle die Nummernfolge seltener durch eine Phrasengrenze getrennt wird. Dieses Ergebnis entspricht auch insofern den Erwartungen, als der Effekt sich für intermediäre Phrasengrenzen zeigt, da diese innerhalb der typografisch vorgegebenen Vierergruppen (die durch Intonationsphrasengrenzen und deutliche Pausen markiert wurden) auftraten.

#### 4.5.1.4 Volle Hunderterzahlen

Betrachtet man **Kontonummern** oder **Telefonnummern**, die volle Hunderterzahlen enthalten, so stellt man fest, dass das Vorhandensein zweier Nullen in der Ziffernfolge die Wahrscheinlichkeit erhöht, dass diese beiden Nullen und die vorangehende Zahl gemeinsam in einer Intonationsphrase realisiert werden.

Schränkt man die Analyse auf diejenigen Beobachtungen ein, in welchen die Ziffern nicht als Hunderterzahl ausgesprochen wurden, so findet man eine Tendenz zur intermediären Phrasengrenze vor den beiden Nullen (positive Korrelation von *hundert* und *only.F.in.,IP* in KTOT bzw. *ip.cuts.F* in KTOT.N, vgl. Abschnitt 4.4.1.1.4, S. 25). Eine Betrachtung der Beobachtungen, in welchen dies der Fall war, zeigt, dass die intermediäre Phrasengrenze stets vor den beiden Nullen gesetzt wurde, die Nullen fungierten also gewissermaßen als Phrasen-Trigger, eine Beobachtung die auch Baumann u. Trouvain (2001) gemacht haben - sie untersuchten allerdings nur einzelne Nullen.

Trotz der Verschiebung der Signifikanzen ist zu beobachten, dass die am häufigsten angewandte Gruppierungsstrategie für dieses Nummernfeature diejenige ist, bei der das komplette Feature, also die beiden Nullen sowie die vorangehende Ziffer, allein innerhalb einer Phrase steht. Dies passiert in 40% der Fälle. Verglichen mit den anderen Features ist die Veränderung, die durch eine Hunderterzahl hervorgerufen wird, jedoch statistisch nicht signifikant.

Für **Kreditkartennummern** lässt sich ebenfalls beobachten, dass das Vorhandensein einer vollen Hunderterzahl in der Nummer dazu führt, dass eher eine intermediäre Phrasengrenze gesetzt wird, als innerhalb eines anderen Features. Allerdings zeigte hier die genaue Betrachtung der Beobachtungen, dass – anders als bei den Telefon- und Kreditkartennummern – die intermediäre Phrasengrenze nicht immer vor der Doppelnull gesetzt wird, sondern dass die Platzierung der Phrasengrenze davon abhängt, an welcher Stelle der Vierergruppe das Feature steht. In den meisten Fällen wurde eine intermediäre Phrase nach zwei Ziffern gesetzt, also genau in der Mitte der Vierergruppe. Nur in seltenen Fällen wurde dieser anscheinend bestehende Default „überschrieben“ und die Phrasengrenze nach der ersten Ziffer der Vierergruppe (was dann in allen Fällen vor der Doppelnull war) gesetzt.

#### 4.5.1.5 Volle Zehnerzahlen

Bei den Zehnerzahlen innerhalb von **Telefon- oder Kontonummern** wurde in keinem der Fälle eine Phrasengrenze zwischen die erste Ziffer und die Null gesetzt. Das heißt, dieses Feature wird als eng zusammengehörig empfunden und die Nummer dementsprechend phrasiert.

Auffällig ist hier der Unterschied zu den vollen Hunderterzahlen: während hier die Anwesenheit der zwei Nullen eine Phrasengrenze vor den Nullen evoziert, ist dies im Falle von Zehnerzahlen bei keiner der getesteten Nummernfolgen der Fall. Hierin unterscheiden sich die Ergebnisse von denen von Baumann u. Trouvain (2001), die in 28% aller Fälle, in welchen eine Null in der Ziffernfolge vorkam, eine Phrasengrenze vor der Null fanden. Baumann u. Trouvain (2001) stellten außerdem in 57% ihrer getesteten Nummern mit

Nullen fest, dass nach der Null eine Phrasengrenze realisiert wurde. Dieses Ergebnis kann in vorliegender Studie bestätigt werden, da in 59% aller getesteten Ziffernfolgen, in welchen eine Null innerhalb der Ziffernfolge zu finden war, eine Phrasengrenze (ip oder IP) nach der Null realisiert wurde. Allerdings ist dieses Ergebnis im Sinne des angewandten  $\chi^2$ -Test nicht signifikant, da sich keine Änderung der Wahrscheinlichkeit von einer nachfolgenden Phrasengrenze im Vergleich zu den anderen Nummernfeatures ergibt, die groß genug ist für statistische Signifikanz.

Die Ergebnisse für **Kreditkartennummern** sind noch deutlicher als die für die Telefon-/Kontonummern: tauchen Nullen in der Nummer auf, so wird das gesamte Feature eher in einer Phrase als in zwei Phrasen realisiert (negative Korrelation mit *ip.cuts.F*, vgl. Abschnitt 4.4.1.2.2, S.28) und steht sogar meist allein innerhalb der Phrase (positive Korrelation mit *only.F.in.IP/ip*, gleicher Abschnitt).

Die beobachteten Effekte gelten auch für die Realisierungen, in welchen das Feature als Folge von Einzelziffern ausgesprochen wurden, sind also nicht auf die Wortgebung zurückzuführen und sollten in einer praktischen Anwendung auf jeden Fall modelliert werden.

#### 4.5.1.6 Wiederholungen von Ziffern

Gleiche Ziffern innerhalb einer Nummer, die nicht nacheinander auftreten (z. B. 6236284), sondern durch eine oder zwei weitere Ziffern getrennt wurden, lösten einen deutlichen Effekt für die Gruppierung von **Telefon- oder Kontonummern** aus: sie sind starke Indikatoren für eine Phrasengrenze innerhalb des Features. Dementsprechend selten werden sie innerhalb der selben Intonationsphrase realisiert (positive Korrelation von *wdh* mit *IP.cuts.F* und negative mit *only.F.in.IP*, vgl. Abschnitt 4.4.1.1.6, S.26).

Auf den ersten Blick war selbst bei den **Kreditkartennummern** ein deutlicher Hinweis auf das Setzen einer Intonationsphrasengrenze zu beobachten, obwohl hier nicht mit Effekten auf die Intonationsphrasen gerechnet worden war. Eine genauere Betrachtung der Daten zeigte allerdings, dass dies nur in denjenigen Fällen zutrifft, in welchen das Feature sich über zwei Vierergruppen erstreckt - der Effekt liegt also mit großer Wahrscheinlichkeit an der typografisch vorgegebenen Gruppierung. Für eine praktische Anwendung ist er daher vernachlässigbar.

Allerdings kann man beobachten, dass in zwei Dritteln aller Fälle von Kreditkartennummern, in denen Ziffern an späterer Position in der Nummer nochmals wiederholt wurden, eine intermediäre Phrasengrenze gesetzt wurde, wohingegen nur in einem Drittel der Fälle von Kreditkartennummern, denen dies nicht der Fall war, eine intermediäre Phrasengrenze an der Stelle des Features gesetzt wurde. Wenngleich auch dies nicht statistisch signifikant ist, so bestätigt es dennoch die Ergebnisse für die Telefon- und Kontonummern: wiederholte Ziffern sollten in einer praktischen Anwendung in zwei verschiedene tonale Einheiten gruppiert werden.

## 4.5.2 Wortgebung

Im Folgenden findet sich ein Überblick über diejenigen Features, die einen Einfluss auf die Realisierung der Ziffernfolge in Bezug auf die verwendete Wortgebung hatten. Wie bei den Ergebnissen zur Gruppierung werden nur diejenigen Features besprochen für die signifikante Effekte gefunden wurden.

### 4.5.2.1 Gleiche Einzelziffern

Für gleiche aufeinanderfolgende Ziffern entsprachen die beobachteten Effekte für **Bankleitzahlen-, Konto- und Telefonnummern** den Erwartungen: Es zeigte sich, dass die jeweiligen Ziffern signifikant häufiger als Phrasen („dreimal die...“) als als Einzelziffern ausgesprochen wurden (vgl. Abschnitt 4.4.2.1, S. 34). Auch der negative Zusammenhang zwischen der Strategie, zwei Ziffern als Zehnerzahl auszusprechen und dem Nummernfeature im Datensatz der Bankleitzahlen widersprach diesen Erwartungen nicht. Die prozentuale Betrachtung der Datensätze zeigte, dass für die Bankleitzahlen ein Fünftel der Fälle, in welchen gleiche Einzelziffern aufeinanderfolgend auftraten, in der Form „dreimal die“ bzw. „zweimal die“ realisiert wurde. Bei den Konto- und Telefonnummern war es nur ein geringer Prozentsatz der aller Fälle, die das Nummernfeature aufwiesen, der auf diese Art realisiert wurde. Dieser schwächere Effekt könnte sich durch die unterschiedliche typografische Präsentation der Stimuli erklären: gleiche Ziffern in Bankleitzahlen waren immer innerhalb einer typografischen Gruppe zu finden (444 888 45) und wurden so eventuell stärker als Einheit wahrgenommen als dies bei den Konto- und Telefonnummern der Fall war.

Obwohl bei den Bankleitzahlen die Hälfte der vorkommenden Features das zweite Vorkommen des Features in der Nummer war, wohingegen bei den Kontonummern die für nicht mal 10% der Fall war, spielt das Mehrfachvorkommen des Features, was eine weitere mögliche Erklärung für den schwächeren Effekt bei den Kontonummern sein könnte, keine Rolle (vgl. Abschnitt 4.4.4, S. 38).

Überraschend war, dass die Tendenz, gleiche Ziffern als Phrase auszusprechen, im Datensatz der **Kreditkartennummern** nicht zu beobachten war. Ein möglicher Erklärungsansatz hierfür wäre wiederum, dass die Default-Phrasierung für Kreditkartennummern, die ja bereits in 4.5.1.4 angenommen wurde, so stark ist, dass in denjenigen Fällen, in welchen die gleichen Ziffern über die Default-ip-Grenze zwischen erster und zweiter Ziffer der Vierergruppe reichen, zugunsten der Standardphrasierung auf eine einprägsamere Wortgebung verzichtet wird.

### 4.5.2.2 Volle Hunderterzahlen

Die Daten zeigen, dass Ziffern, die eine volle Hunderterzahl innerhalb einer **Telefon- oder Kontonummer** oder einer **Bankleitzahl** bilden, tendenziell auch eher als solche ausgesprochen werden: im Falle der Bankleitzahlen galt dies für alle Fälle, bei den Konto- oder Telefonnummern für immerhin ungefähr ein Drittel der Vorkommnisse. Es ist zu beachten, dass das Nummernfeature anscheinend generell eine abweichende Wortgebung evoziert, da für den kompletten Datensatz auch eine signifikante positive Korrelation mit

der Aussprache als Zehnerzahl zu finden war. Präferierter war allerdings die Aussprache als Hunderterzahl, was die signifikante negative Korrelation des Nummernfeatures mit der Aussprache als Zehnerzahl innerhalb derjenigen Daten, die alle nicht in der Standardwortgebung geäußert wurden, erklärt.

Obwohl bei den **Kreditkartennummern** eine Tendenz zur „...hundert“-Aussprache beobachtet wurde, war diese Tendenz nicht signifikant. Dies könnte wiederum an der Stärke des Default-Phrasierungsmusters von Kreditkarten sein (vgl. Abschnitte 4.5.2.1, S 48 und 4.5.1.4, S 46), da ja jedes Vorkommen des sich über drei Ziffern erstreckenden Nummernfeatures über die Default-ip-Grenze hinweg reicht und eine Phrasierung an dieser Stelle nicht möglich wäre, würde das Feature als Hunderterzahl ausgesprochen.

#### 4.5.2.3 Volle Zehnerzahlen

Bei den Zehnerzahlen war das Ergebnis klar und genau wie erwartet: eine volle Zehnerzahl innerhalb der Nummer (das heißt, eine Ziffer gefolgt von einer Null) löst einen signifikanten Effekt auf die verwendete Wortgebung aus: es ist wahrscheinlicher, dass die beiden Zahlen als Zehnerzahl ausgesprochen werden als dies für andere Nummernfeatures der Fall ist. Bei Bankleitzahlen war das für 55% aller Vorkommnisse, bei den anderen Nummerntypen nur für ca. 20% der Fall.

### 4.5.3 Akzentuierung

Wie in Abschnitt 4.4.3 (S. 38) beschrieben, war unerwarteterweise kein signifikanter Zusammenhang zwischen einer Akzentverschiebung und sich wiederholenden Ziffern innerhalb der Nummer zu beobachten. Demnach scheint die starke Zäsur durch eine Intonationsphrasengrenze (vgl. Abschnitt 4.5.1.6, S. 47) einzige Konsequenz dererlei Ziffernkombinationen zu sein. Allerdings fiel hier bei der detaillierteren Analyse der Ziffern auf, dass bei Fällen, in welchen mehr als zwei gleiche Einzelziffern vorkamen und zwei der Wiederholungen aufeinander folgten (z. B. 63664696), die beiden gleichen aufeinanderfolgenden Einzelziffern den Effekt der Wiederholung an anderer Stelle überdeckten. Die Nummer wurde also von den Probanden in die Feature-Kategorie *gleiche* eingeordnet.

Eine überraschender Zusammenhang war zwischen einer Akzentverschiebung und vollen Zehnerzahlen innerhalb der Nummer zu beobachten. Der Akzent wurde signifikant häufiger innerhalb eines Features verschoben, wenn es sich bei dem Feature um eine volle Zehnerzahl handelte. Da dieser jedoch unintuitiv, in einigen Beispielen sogar falsch erscheint, wurde für eine praktische Anwendung auf die Modellierung dieses Zusammenhangs verzichtet.

# 5 Praktische Anwendung des Modells

Im Folgenden soll skizziert werden, wie die Erkenntnisse des Produktionsexperiments praktisch angewendet werden können.

Erst wird dargestellt, wie die relevanten Phänomene in der Nummer ermittelt werden könnten. Da dieser Schritt im Rahmen der vorliegenden Arbeit ebenfalls stattfinden musste (vgl. Abschnitt 3.4.5.2, S. 12), werden die dafür vorgestellten Algorithmen in der für die Verarbeitung der Daten des Produktionsexperiments verwendeten Programmiersprache Perl angegeben.

Da keine Implementierung des Modells innerhalb eines bestimmten Sprachsynthesystems erfolgte, wird für den zweiten Schritt, die Modellierung der Auswirkungen eines jeden Nummernfeatures auf die Phrasierung und Wortgebung der jeweiligen Nummer, eine Programmierempfehlung im Pseudo-Code gegeben.

## 5.1 Detektierung der Nummernfeatures

Die Ermittlung der Nummernfeatures erfolgte im vorgestellten Experiment mithilfe von Perl-Skripten. Die Ziffernabfolge wurde eingelesen und dann linear auf die verschiedenen Features abgetestet.

Im Folgenden finden sich Fragmente des Codes, die zur Ermittlung der Features dienen.

Es wird empfohlen, nicht alle in (vgl. Abschnitt 3.2.2, S. 5) vorgestellten Features zu berücksichtigen, da einige der Features keinen signifikanten Effekt auf die Phrasierung oder die Wortgebung der Ziffernfolge hatten, daher sind auch nur die relevanten Teile des Codes gelistet.

Da im vorliegenden Experiment nur eine Art von Features innerhalb der Nummern ermittelt werden sollte, erfolgte der Programmaufruf mit einer Kommandozeilenoption, in der das gewünschte Feature angegeben werden konnte (vgl. Algorithmus 1). Die erforderliche Eingabe für das Programm war ein Textfile, welches die zu untersuchende Ziffernabfolge enthielt. Die Ausgabe des Skriptes war ein Textfile, das eine Repräsentation der Ziffernfolge wie in Abschnitt 3.4.5.2 (S. 12) beschrieben, enthielt.

### 5.1.1 Auf- bzw. absteigende Ziffernfolgen

Da die Wortgebung nicht durch das Vorhandensein von auf- bzw. absteigenden Ziffernfolgen beeinflusst wird und die Effekte, die für die Gruppierung auf- bzw. absteigender Ziffernfolgen gefunden wurden, nur das Setzen von Phrasengrenzen innerhalb des Features betrafen - ein Effekt, der nur zeigt, dass das Feature nicht als zusammengehörende



---

**Algorithmus 1** Fragment des Perlcodes zur Lokalisierung der Features

---

```
#!/usr/local/bin/perl
use strict;
if ($#ARGV < 1)
  {print "... "
  exit;
  }

# Einlesen der Kommandozeilenargumente:
my $feature = shift @ARGV;
my $numberfile = shift @ARGV;

my $digits = ();

#Öffnen des Nummern-(text)-files
open NRFILE, $numberfile
or die "\n $0: \n Cannot open $numberfile. \n $!\n";
chomp($digits=<NRFILE>);
close NRFILE;

#Array: jede Ziffer ein Element
my @digits_sep=split(//,$digits);

#Aufrufen der Subroutinen
if ($feature eq "-gleiche"){&gleiche_ziffern(@digits_sep)}
elsif($feature eq "-auf"){&folge_auf(@digits_sep)}
elsif($feature eq "-ab"){&folge_ab(@digits_sep)}
elsif($feature eq "-hundert"){&hundert(@digits_sep)}
elsif($feature eq "-zehn"){&zehn(@digits_sep)}
elsif($feature eq "-wdh"){&wdh(@digits_sep)}
else{print"Gebrauch: ... ''
  }
```

---

Einheit empfunden wird, wird empfohlen, für diese Features weder Wortgebung noch Phrasierung in besondere Art und Weise zu implementieren. Daher entfällt auch die Lokalisierung des Features.

### 5.1.2 Gleiche Einzelziffern

Gleiche Ziffern werden dadurch definiert, dass die aktuelle Ziffer gleich der Folgeziffer ist. Im vorgestellten Algorithmus wurde jedes Feature mit einem Buchstaben, Ziffern, die nicht Teil des Features waren, mit einer Null markiert. Die Markierung erfolgte in einem Ausgabearray, das so viele Elemente enthielt, wie die eingelesene Ziffer Zahlen. Die Markierung des Features begann dem dem Zeichen „a“ zugehörigen Wert, der dann, nach Abschluss eines Features (wenn die aktuelle Ziffer nicht mehr gleich der Folgeziffer war), hochgezählt wurde, so dass eventuelle Mehrfachvorkommen des Features mit unterschiedlichen Buchstaben markiert wurden (vgl. Algorithmus 2).

---

#### Algorithmus 2 Programmcode zur Ermittlung gleicher Einzelziffern in der Nummer

---

```
sub gleiche_ziffern {  
  
    #Array von der Länge des Eingabearrays ,  
    #Belegung mit Nullen  
    my @abstract=split (//, "0"x@_);  
  
    #Anfangszeichen zur Markierung des Features :  
    #Ascii-Code von 'a'  
    my $featmark=ord('a');  
  
    #Schleife über Arraylänge  
    for(my $i=0; $i<@_-1; $i++){  
        if ($_[ $i]==$_[ $i+1]){  
            $abstract[ $i]=chr($featmark);           #aktuelle Ziffer = Folgeziffer  
            $abstract[ $i+1]=chr($featmark)         #Markieren der Ziffern  
                                                    #mit dem entsprechenden Zeichen  
        }  
        else{  
            $featmark+=1                            #Ascii-Code-Markierung hochzählen  
        }  
    }  
    print "@abstract\n";  
}
```

---

### 5.1.3 Volle Hunderterzahlen

Volle Hunderterzahlen werden dadurch definiert, dass der aktuellen Ziffer zwei Nullen folgen. In Perl musste darauf geachtet werden, dass es tatsächlich zwei Folgeziffern gibt

(vgl. Algorithmus 3, Zeile 5). Wurde ein Hunderterfeature gefunden, wurden die entsprechenden Stellen im Ausgabearray mit dem selben Buchstaben belegt (Zeile 6-8). Wurde kein Feature festgestellt, wurde der Feature-Marker hochgezählt (Zeile 10).

---

**Algorithmus 3** perl-Programmcode zur Ermittlung voller Hunderterzahlen in der Nummernfolge

---

```
sub hundert { 1
  my @abstract=split (//,"0"x@_); 2
  my $featmark=ord('a'); 3
  for(my $i=0; $i<@_-1; $i++){ 4
    if ($_[ $i+1]==0 && defined($_[ $i+2])&& $_[ $i+2]==0 ){ 5
      $abstract[ $i]=chr($featmark); 6
      $abstract[ $i+1]=chr($featmark); 7
      $abstract[ $i+2]=chr($featmark); 8
    } else { 9
      $featmark+=1 10
    } 11
  } 12
  print "@abstract\n"; 13
} 14
```

---

#### 5.1.4 Volle Zehnerzahlen

Die vollen Zehnerzahlen wurde analog zu den Hunderterzahlen ermittelt, nur dass gefordert wurde, dass die zweite Folgeziffer ungleich Null bzw. nicht definiert sein sollte (vgl. Algorithmus 4, Zeile 5-7). Analog zu den anderen Subroutinen wurde dann das Ausgabearray mit Buchstaben belegt, im anderen Fall die Featuremarkierung hochgezählt (Zeile 8-12).

#### 5.1.5 Wiederholung von nicht aufeinanderfolgenden Ziffern

Für nicht aufeinanderfolgende identische Ziffern wurden verschiedene Muster abgetestet.

Es wurden gleiche Ziffern innerhalb von Zweiergruppen abgetestet, also gleiche Ziffern, denen zwei unterschiedliche Ziffern vorangehen (6353). Im Algorithmus wurden sie ermittelt, indem die erste Folgeziffer der aktuellen Ziffer und die dritte Folgeziffer auf Gleichheit abgeprüft wurden und gegebenenfalls alle vier Ziffern mit einem Buchstaben markiert (vgl. Algorithmus 5, Zeile 7-15) und dann das Markierungszeichen für ein Feature hochgezählt wurde (Zeile 16).

Außerdem wurden gleiche Ziffern innerhalb von Dreiergruppen ermittelt. Sie wurden definiert als vier gleiche Ziffern, die zwei unterschiedliche Ziffern einschließen (636646). Es wurde abgetestet, ob die aktuelle Ziffer gleich ihrer zweiten, dritten und fünften Folgeziffer ist (Zeile 17-24). In diesem Fall wurden alle sechs Ziffern markiert und das Markierungszeichen hochgezählt (Zeile 25-31). Für eine praktische Anwendung müsste

---

**Algorithmus 4** perl-Programmcode zur Ermittlung voller Zehnerzahlen in der Ziffernfolge

---

```
sub zehn { 1
my @abstract=split (//, "0"x@_); 2
my $featmark=ord( 'a' ); 3
for (my $i=0; $i<@_-1; $i++){ 4
    if ( $_[$i+1]==0 5
        && ( $_[$i+2]!=0 || $_[$i+2]== undef) 6
        && $_[$i]!=0 7
    ){ 8
        $abstract[$i]=chr($featmark); 9
        $abstract[$i+1]=chr($featmark); 10
    } 11
    else {$featmark+=1} 12
} 13
print "@abstract\n"; 14
```

---

aber geprüft werden, ob dieses Muster tatsächlich Sinn macht, da bei der Sichtung der Daten der perzeptuelle Eindruck entstand, dass es meist von anderen Schemata überlagert wurde (vgl. Abschnitt 6, S. 61).

Die zwei weiteren Muster, die abgeprüft wurden, bestehen aus gleichen Paaren von Ziffern, in einem Fall gefolgt von zwei unterschiedlichen Ziffern, im anderen Fall den zwei unterschiedlichen Ziffern folgend (636736 bzw. 623628). Die Implementierung findet sich in Zeile 32-56 von Algorithmus 5 (S. 55) und geschah analog zu den anderen Mustern.

## 5.2 Bestimmung der Gruppierung

Es wird empfohlen, eine Default-Gruppierung für jeden Nummerentyp zu etablieren.

Hierbei sollte für Konto- und Telefonnummern eine der in Baumann u. Trouvain (2001) gelisteten Default-Gruppierungsstrategien gewählt werden. Im vorliegenden Fall waren die häufigsten Gruppierungsmuster das Muster 3-2-2 (44% aller siebenstelligen Nummern) und 2-2-3 (30%), weswegen im Folgenden von einer Default-Gruppierung, die dem Muster 3-2-2 entspricht, ausgegangen wird.

Kreditkartennummern sollten, aufgrund der in der vorliegenden Studie festgestellten „Resistenz“ gegenüber Änderungen der Default-Phrasierungs-Struktur mithilfe von Intonationsphrasen in Vierergruppen, diesen dann wiederum durch intermediäre Phrasengrenzen in Zweiergruppen geordnet werden. Eine andere Möglichkeit, die den Kreditkartennummern eigene Rhythmik zu erreichen, könnte es sein, bei der Akzentuierung der Vierergruppe (*unbetont - betont - unbetont - betont*) den ersten Akzent etwas „stärker“ (in Range und Steigung) zu wählen, um so die perzeptuelle Zäsur nach den ersten beiden Ziffern zu erzeugen. Bankleitzahlen sollten in die, durch die Typographie vorgegebenen, Gruppen (3-3-2) eingeteilt werden.

---

**Algorithmus 5** perl-Programmcode zur Ermittlung von Wiederholungsphänomenen in der Ziffernfolge

---

```
sub wdh { 1
  my @abstract=split(//,"0"x@_); 2
  my $featmark=ord('a'); 3
  for(my $i=0; $i<@_-1; $i++){ 4
    if (#Muster x1x1 - 63 53 5
      defined($_[$i+2]) && 6
      defined($_[$i+3]) && 7
      defined($_[$i+4]) && 8
      $_[$i+3]==$_[$i+1] 9
    ){ $abstract[$i]=chr($featmark); 10
      $abstract[$i+1]=chr($featmark); 11
      $abstract[$i+2]=chr($featmark); 12
      $abstract[$i+3]=chr($featmark); 13
      $featmark+=1} 14
    elsif (#Muster 1x11x1 - 646 656 15
      defined($_[$i+2])&& defined($_[$i+3]) 16
      && defined($_[$i+4]) && defined($_[$i+5]) 17
      && $_[$i]==$_[$i+2] && $_[$i+2] == $_[$i+3] 18
      && $_[$i+3]==$_[$i+5] 19
    ){ $abstract[$i]=chr($featmark); 20
      $abstract[$i+1]=chr($featmark); 21
      $abstract[$i+2]=chr($featmark); 22
      $abstract[$i+3]=chr($featmark); 23
      $abstract[$i+4]=chr($featmark); 24
      $abstract[$i+5]=chr($featmark); 25
      $featmark+=1} 26
    elsif(#Muster x12x12 - 636 736 27
      defined($_[$i+2]) 28
      && defined($_[$i+3]) && defined($_[$i+4]) 29
      && defined($_[$i+5]) && $_[$i+1] == $_[$i+4] 30
      && $_[$i+2] == $_[$i+5] 31
    ){ $abstract[$i+1]=chr($featmark); 32
      $abstract[$i+2]=chr($featmark); 33
      $abstract[$i+4]=chr($featmark); 34
      $abstract[$i+5]=chr($featmark); 35
      $featmark+=1} 36
    elsif(#Muster 12x12x - 623 628 37
      defined($_[$i+1]) 38
      && defined($_[$i+2])&& defined($_[$i+3]) 39
      && defined($_[$i+4])&& $_[$i+3] == $_[$i] 40
      && $_[$i+1] == $_[$i+4] 41
    ){ $abstract[$i]=chr($featmark); 42
      $abstract[$i+1]=chr($featmark); 43
      $abstract[$i+3]=chr($featmark); 44
      $abstract[$i+4]=chr($featmark); 45
      $featmark+=1} 46
  } 47
  print "@abstract\n"; 48
} 49
```

---

Bei der Implementierung, die im Folgenden vorgeschlagen wird, handelt es sich um eine Constraint-Regelung. Ausgehend von einer Featurerepräsentation der Nummer wie in Abschnitt 3.4.5.2 (S.12) beschrieben, wird der Featureteil eingelesen und für jeden Zwischenraum zwischen den Ziffern, wenn nötig, ein Constraint gesetzt, der die Default-Gruppierungsstrategie manipulieren kann. Das heißt, zuerst erfolgt ein Setzen der Constraints für die Zwischenräume zwischen den Ziffern, die gemeinsam ein Feature bilden, danach wird die Default-Gruppierung aktiviert, die dann die Phrasen gemäß der Default-Regel unter Berücksichtigung der Constraints setzt. Bei den Constraints handelt es sich entweder um Marker, die erzwingen, dass eine Phrase realisiert wird, bzw. an dieser Stelle keine Phrase Realisiert wird (*ip*, *IP*, *no-boundary*), oder um Anweisungen, die zur Folge haben, dass die geplante tonale Kategorie transformiert wird, die also nur wirken, wenn von der Default-Gruppierung ein bestimmtes Ereignis an dieser Stelle geplant ist (*IP-to-ip*, *no-boundary-to-ip*). Die Constraints werden in einer Liste *constraints* der Länge *Featurelänge+1* gespeichert, die an die Defaultgruppierung zurückgegeben wird.

Die Implementierung wird nur für die Konto- und Telefonnummern im Pseudocode dargestellt, das Verarbeiten der Kreditkartennummern erfolgt analog, der jeweils vorgeschlagenen Strategie entsprechend.

### 5.2.1 Gleiche Einzelziffern

**Kontonummern und Telefonnummern** sollten beim Auftreten von gleichen, aufeinanderfolgenden Einzelziffern so gruppiert werden, dass erstens keine Intonationsphrase oder intermediäre Phrase innerhalb der zusammengehörigen Ziffern steht, und dass zweitens die Ziffern, wenn möglich innerhalb ein und der selben Intonationseinheit stehen. Bei zwei- und dreistelligen Features wird empfohlen alle Ziffern in eine Intonationsphrase zu gruppieren, bei längeren Abfolgen von gleichen Ziffern sollte die Default-Gruppierung vorgenommen werden, jedoch nicht mithilfe von Intonationsphrasen, sondern innerhalb des Features nur durch intermediäre Phrasen strukturiert werden. Nach dem Featureende, sollte wieder die Default-Strategie (mit Intonationsphrasen) verfolgt werden (vgl. Algorithmus 6).

Auch für **Kreditkartennummern** wird vorgeschlagen, die Default-Gruppierung derartig abzuwandeln. Allerdings sollte das nur für bis zu vierstellige Features gelten. Die „natürliche“ Phrasengrenze zwischen den Vierergruppen einer Kreditkartennummer sollte nicht manipuliert werden.

### 5.2.2 Volle Hunderterzahlen

Volle Hunderterzahlen in **Konto- und Telefonnummern** sollten (bei Standardwortgebung) dazu führen, dass vor der Doppelnull eine intermediäre Phrasengrenze gesetzt wird. Das heißt, in dem Zwischenraum zwischen erster und zweiter Ziffer des Features, soll eine intermediäre Phrase gesetzt werden. Der Mini-Algorithmus 7 (S. 57) zeigt die entsprechenden Programmzeilen.

---

**Algorithmus 6** Pseudocode zur Gruppierung von Konto- und Telefonnummern bei gleichen Einzelziffern

---

```
read feature                                ▷ Input: flache Repräsentation des Features
if feature.length < 4 then                  ▷ bis zu dreistelliges Feature
  for i=0 to feature.length-1 do
    constraints[i]=no-boundary              ▷ keine Phrasengrenze innerhalb des Features
  end for
  constraints[feature.length]=IP           ▷ danach IP-Grenze
  return constraints
else                                        ▷ längeres Feature
  for i=0 to feature.length-1 do
    constraints[i]= IP-to-ip                ▷ Default Gruppierung, aber ip anstatt IP
  end for
  return constraints
end if
```

---

---

**Algorithmus 7** Pseudocode zur Gruppierung von Konto- und Telefonnummern bei vollen Hunderterzahlen

---

```
constraints[1]= ip-boundary
return constraints
```

---

Für **Kreditkartennummern** sollte aufgrund der Abhängigkeit von der Default-Gruppierung (vgl. Abschnitt 4.5.1.4, S. 46) keine Manipulation von ebendieser stattfinden.

### 5.2.3 Volle Zehnerzahlen

Sowohl bei **Kreditkarten-**, also auch bei **Konto- und Telefonnummern** wird empfohlen, keine Phrasengrenze innerhalb des Features zu erlauben und nach dem Feature eine Intonationsphrasengrenze zu setzen (vgl. Algorithmus 8).

---

**Algorithmus 8** Pseudocode zur Gruppierung von Kreditkarten-, Konto- und Telefonnummern bei vollen Zehnerzahlen

---

```
constraints[1]=no-boundary
constraints[feature.length]=IP
return constraints
```

---

### 5.2.4 Wiederholungen von Ziffern

Was das Phänomen der wiederholten Ziffern, die durch andere Ziffern voneinander getrennt sind, anbelangt, so gilt auch hier, dass die Gruppierung für **Kreditkarten-, Konto-, und Telefonnummern** gleichartig erfolgt. Die Ziffern, die das erste Vorkommen der Ziffern enthalten und die, die die Wiederholung enthalten, sollten in zwei

verschiedenen tonalen Einheiten gruppiert werden. Bei den Kreditkartennummern, sollte dies allerdings, wie bei den gleichen Einzelziffern, nur für Features gelten, die nicht höchstens vierstellig sind. Außerdem sollte hier die Phrasengrenze, die gesetzt wird, eine intermediäre Phrasengrenze sein, im Falle der Telefon- und Kontonummern jedoch eine Intonationsphrasengrenze.

Es wird vorgeschlagen, falls die Default-Gruppierung keine Phrasengrenze vor und nach dem Feature vorschlägt, zumindest eine intermediäre Phrasengrenze zu setzen. Soll an der entsprechenden Stelle eine Intonationsphrase stehen, so sollte diese realisiert werden.

Algorithmus 8 (S. 57) beschreibt die Implementierung für Konto- und Telefonnummern. Es muss dabei darauf geachtet werden, dass die Phrasengrenze an der richtigen Stelle steht. Bei Ziffernfolgen, die ein sechsstelliges Feature beschreiben (636-736, 623-628), sollte die Phrasengrenze nach drei, bei denen die ein zweistelliges Feature markieren (64-65), nach zwei Ziffern gesetzt werden.

---

**Algorithmus 9** Pseudocode zur Gruppierung von Konto- und Telefonnummern bei wiederholten, voneinander getrennt auftretenden, Ziffern

---

```
read feature
if feature.length = 4 then
    constraints[0] = no-boundary-to-ip
    constraints[2] = IP
    constraints[4] = no-boundary-to-ip
    return constraints
else
    constraints[0] = no-boundary-to-ip
    constraints[3] = IP
    constraints[4] = no-boundary-to-ip
    return constraints
end if
```

---

## 5.3 Bestimmung der Wortgebung

Bei der Bestimmung der Wortgebung sollte daran gedacht werden, dass die Wortgebung sehr stark sprecherabhängig ist (vgl. Abschnitt 4.4.4.2.2, S. 42) und die in dieser Untersuchung ermittelten Ergebnisse nur den „Durchschnittssprecher“ repräsentieren. Daher könnte man zur Implementierung einen bestimmten Sprecher heranziehen und dessen Präferenzen mit in die Implementierung einfließen lassen. Im Folgenden wird dies nicht getan, sondern es wird versucht, einen „globalen“ Ansatz zu beschreiben.

Generell ist die Bestimmung der Wortgebung, bzw. die Bestimmung von abweichender Wortgebung, etwas aufwändiger als die Bestimmung der Gruppierung, da jeweils noch erkannt werden muss, um welche Ziffern es sich handelt (um z. B. zu entscheiden, ob „dreimal die *drei*“ oder „dreimal die *vier*“ gesagt werden soll). Daher sollte hier nicht



die abstrakte Repräsentation des Features, sondern die entsprechenden Ziffern eingelesen werden. In der vorgeschlagenen Implementierung werden diese in der Liste *numbers* gespeichert und zur Bestimmung der Wortgebung herangezogen. Die Ausgabeliste *words* gibt an, wo die Wortgebung verändert werden soll. Hat das Array keinen Eintrag, soll die Standardwortgebung verwendet werden.

### 5.3.1 Gleiche Einzelziffern

Gleiche Einzelziffern könnten für **Konto- und Telefonnummern** sowie für **Bankleitzahlen** in abweichender Wortgebung realisiert werden. **Kreditkartennummern** sollten in Standardwortgebung realisiert werden.

Eventuell wäre es sinnvoll, die abweichenden Wortgebung Prozentsatz der Fälle so zu implementieren. Diese Frage muss in Abhängigkeit von der konkreten Anwendung geklärt werden. Wird die Synthese in einem System gebraucht, in dem häufig Ziffernfolgen vorgelesen werden sollen, so könnte dies Sinn machen. Wird jeweils ein Benutzer nur eine Ziffernfolge abfragen, so besteht für diese Art von Variation kein Bedarf. In diesem Fall muss für die konkrete Anwendung abgewägt werden, ob eine Aussprache der Ziffernfolge als Phrase für das System einen Vorteil bringt (zum Beispiel eine leichterere Erinnerbarkeit der vorgelesenen Ziffernfolge oder einen kürzere Ansagezeit).

Bei gleichen Einzelziffern wird die Identität einer beliebigen Ziffer des Features, ein String, sowie die Featurelänge dazu verwendet die richtige Wortgebung zu erzeugen: Aus 222 (Featurelänge 3, Identität einer beliebigen Ziffer: 2) wird „3mal die 2“ (vgl. Algorithmus 10). Danach wird die korrekte Aussprache „dreimal die Zwei“ vom Graphem-zu-Phonemsystem der Sprachsynthese erzeugt.

---

**Algorithmus 10** Bestimmung der Wortgebung von Bankleitzahlen, sowie Konto- und Telefonnummern beim Auftreten von gleichen Einzelziffern

---

```
read numbers
number = numbers[1]           ▷ Identität einer beliebigen Ziffer
length = feature.length      ▷ Featurelänge
wording = append(length, "mal die", number)  ▷ Zusammensetzen des Strings
words[0] = wording
for i=1 to feature.length-1 do  ▷ die restlichen Positionen bleiben leer
    words[i]=no-wording
end for
return words
```

---

### 5.3.2 Volle Hunderterzahlen

Bei vollen Hunderterzahlen verhält es sich wie bei den gleichen Einzelziffern: **Bankleitzahlen** sowie **Kontonummern und Telefonnummern** sollten mit abweichender Wortgebung realisiert werden, die letzteren beiden eventuell nur in ca. einem Drittel

der Fälle. **Kreditkartennummern** hingegen sollten in Standardwortgebung geäußert werden.

Der Vorschlag für die Implementierung geht davon aus, dass das Sprachsynthesystem zusammengeschrriebene Ziffern als volle Zahlen ausspricht, wie es bei System Festival (Black 1997) der Fall ist. Sollte dies nicht der Fall sein, findet sich ein Alternativvorschlag findet sich in Anhang C (S.88).

Im hier vorgestellten Algorithmus werden alle Ziffern des Features zu einem String zusammengesetzt und dieser als zusammenhängende Zahl an das Synthesystem weitergegeben (vgl. Algorithmus 11).

---

**Algorithmus 11** Bestimmung der Wortgebung von Bankleitzahlen, sowie Konto- und Telefonnummern beim Auftreten von vollen Zehner- oder Hunderterzahlen

---

```
read numbers
wording = " " ▷ leerer String
for i= 0 to feature.length-1 do
    paste(wording, numbers[i]) ▷ Zusammensetzen des Strings
    words[0] = wording
end for
for i=1 to feature.length-1 do ▷ die restlichen Positionen bleiben leer
    words[i]=no-wording
end for
return words
```

---

### 5.3.3 Volle Zehnerzahlen

Volle Zehnerzahlen sollten für alle Nummerntypen in abweichender Wortgebung implementiert werden. Wird eine Implementierung gewählt, die nur einen Teil der Realisierungen abweichend von der Standardwortgebung ausgibt, so sollten **Bankleitzahlen** in der Hälfte der Fälle, die anderen Nummerntypen nur in einem Fünftel auf diese Art und Weise realisiert werden. Algorithmus 11 (S.60) kann exakt übernommen werden, um diese Wortgebung zu implementieren, da das Zusammensetzen des Strings in Abhängigkeit von der Featurelänge passiert. Bei Zehnerzahlen besteht der String also aus zwei, bei Hunderterzahlen aus drei Ziffern.

# 6 Fazit

## 6.1 Diskussion

Die vorliegende Studie beschäftigte sich mit den Auswirkungen bestimmter Ziffernabfolgen auf die Gruppierung, Wortgebung und Akzentuierung von numerischen Ausdrücken und stellt einen Vorschlag zur Implementierung dieser Effekte im Rahmen einer Sprachsynthesekomponente vor.

Es zeigte sich, dass es sinnvoll ist, die verschiedenen Nummerntypen (Bankleitzahlen, Kreditkartennummern, Telefonnummern sowie Kontonummern) unterschiedlich zu behandeln, da diese unterschiedlich auf das Vorhandensein eines Nummernfeatures in der Ziffernfolge reagieren.

Des Weiteren wurde festgestellt, dass in der vorliegenden Studie nicht für alle in Baumann u. Trouvain (2001) vorgestellten Phänomene eine Auswirkung auf Wortgebung und Gruppierung der Ziffernfolge nachgewiesen werden konnte. Dies kann zum einen daran liegen, dass die Autoren keine statistischen Tests durchführten, sondern nur relative Häufigkeiten angeben. Diese entsprechen denen der vorliegenden Studie. Zum anderen könnte das experimentelle Design dafür verantwortlich sein: zwar glich sich das Aufnahmeverfahren, jedoch könnte es sein, dass in Baumann u. Trouvain (2001) das Hauptaugenmerk auf die Ermittlung der verschiedenen präferierten Gruppierungsmuster gerichtet wurde und daher die Stimuli anders geartet waren, als in vorliegendem Experiment, in dem kein Default ermittelt wurde, sondern ausschließlich die verschiedenen Auswirkungen der zu untersuchenden Phänomene betrachtet wurden.

Interessanterweise zeigte sich, dass für die Wortgebung und die Gruppierung die jeweils gleichen Features Auswirkungen hatten. Dabei handelte es sich um folgende Phänomene:

- gleiche aufeinanderfolgende Einzelziffern
- volle Hunderterzahlen
- volle Zehnerzahlen
- gleiche nicht aufeinanderfolgende Einzelziffern

Aus diesem Grund beschränkte sich der Vorschlag für eine Implementierung des Modells im Rahmen eines Sprachsynthesystems auf diese Features. Es wurde eine Implementierung vorgeschlagen, die auf einer Default-Gruppierungsstrategie basiert und diese an bestimmten Stellen des jeweiligen Features manipulieren kann. Dies sollte bei jedem Auftreten eines solchen Features passieren. Anders hingegen verhält es sich bei der Wortgebung: hier wurde vorgeschlagen, je nach Anwendung zu entscheiden, ob überhaupt

eine von der Standardwortgebung abweichende Wortgebung verwendet werden soll. Der Hintergrund hierfür ist die Abhängigkeit der Wortgebung vom Einzelsprecher. Was die Gruppierungsstrategien anbelangt, so waren waren weitaus seltener Korrelationen von bestimmten Sprecher zu bestimmten Strategien zu finden, als im Falle der Wortgebung.

Für die Akzentuierung ergab sich ein unerwartetes Ergebnis: das Wiederholen gleicher Einzelziffern führte nicht dazu, dass der Akzent verschoben wurde. Daher wurde für die Implementierung keine Modifizierung der Default-Akzentuierung vorgeschlagen.

## 6.2 Ausblick

Der Schwerpunkt der vorliegenden Arbeit lag auf der detaillierten Untersuchung der Auswirkung verschiedener Eigenschaften von Ziffernfolgen auf ihre Wortgebung, Gruppierung und Akzentuierung, nicht auf der Implementierung der Ergebnisse.

Generell ist zu sagen, dass das vorgestellte Experiment deutlich zeigt, dass eine praktische Anwendung des Modells im Rahmen eines Sprachsynthesystems sinnvoll ist. Aufgrund der Komplexität der Zusammenhänge zwischen bestimmten Eigenschaften einer Nummernfolge, dem Typ der Nummer und ihrer Gruppierung bzw. Wortgebung, ist es keinesfalls sinnvoll, ein Synthesystem für das Deutsche ohne eine Nummernkomponente, die die Ziffernfolgen detailliert analysiert und die Phrasierung und Wortgebung in Abhängigkeit der Ziffernidentitäten bestimmt, bereitzustellen.

Daher wäre eine Umsetzung der vorgeschlagenen Implementierung und eine anschließende Evaluierung der Komponente erstrebenswert. Der vorgeschlagene Ansatz wäre für die Unit-Selection insofern vielversprechend, als er auf Default-Regelungen basiert, die, je nach Eigenschaften der jeweiligen Nummer, abgeändert werden können, jedoch nicht müssen. Das heißt, dass bei der Einheitenauswahl mehrere Varianten generiert werden können, die sich sowohl hinsichtlich ihrer Phrasierung als auch hinsichtlich der zu verwendenden Wortgebung voneinander unterscheiden. Werden daher für die präferierte Realisierung keine geeigneten Kandidaten gefunden, könnte auf andere Realisierungen zurückgegriffen werden, um zum Beispiel hohe Konkatenierungskosten oder anderweitig ungeeignete Kandidaten zu vermeiden.

In der vorliegenden Arbeit sollten die Auswirkung einzelner Features isoliert untersucht werden. Ein weiterer Schritt für eine mögliche Folgeuntersuchung wäre nun die Ermittlung einer Feature-Hierarchie, die beschreibt, welche Features einen stärkeren Zusammenhalt aufweisen. Im hier vorgestellten Experiment fiel bei der Sichtung der Daten auf, dass bei dem Feature *wdh*, also bei der Wiederholung von Einzelziffern, wie z. B. in der Telefonnummer 636646, die aufeinanderfolgenden Ziffern einen stärkeren Zusammenhalt aufweisen zu schienen - die Ziffer wurde meist als 63-66-46 gruppiert, die erwartete Akzentverschiebung blieb aus.

Außerdem könnten die Störparameter im Experiment, vor allem das etwaige Mehrfachvorkommen des selben Features innerhalb einer Nummer, detaillierter untersucht werden, als es im Rahmen dieser Arbeit möglich war.

# A Übersicht über die Stimuli im Produktionsexperiment

## A.1 Telefonnummern

### 1. aufeinanderfolgende gleiche Ziffern

#### a) Zweierblock

- i. 6332272 (zweimal zwei Mitte)
- ii. 447358 (einmal zwei Anfang)
- iii. 53911 (einmal zwei Ende)
- iv. 0711-2364487 (einmal zwei Vorwahl, einmal zwei Nummer)

#### a) Dreierblock

- i. 2144487 (einmal drei Mitte)
- ii. 5552317 (einmal drei Anfang)
- iii. 4219222 (einmal drei Ende)
- iv. 07333-9657771 (einmal drei Vorwahl, einmal drei Nummer)

### 2. aufeinanderfolgende Ziffernfolgen

#### a) aufsteigend

- i. 6783914 (Anfang)
- ii. 3145692 (Mitte)
- iii. 9786234 (Ende)
- iv. 02456-85376 (Vorwahl)

#### b) absteigend

- i. 5437162 (Anfang)
- ii. 3165492 (Mitte)
- iii. 9786432 (Ende)
- iv. 02654-85376 (Vorwahl)

### 3. volle größere Zahlen

#### a) ganze Zehner

- i. 6307520 (zweimal verteilt)
- ii. 3020389 (zweimal Anfang)
- iii. 7394630 (einmal Ende)
- iv. 7039452 (einmal Anfang)
- v. 03024-675381 (Vorwahl)

b) ganze Hunderter

- i. 6300500 (zweimal)
- ii. 2005896 (einmal Anfang)
- iii. 8693200 (einmal Ende)
- iv. 8690031 (einmal Mitte)

4. wiederholte Ziffern

- a) 5393278
- b) 6236284
- c) 4623628
- d) 07161-675381

5. Nullen

- a) 6703016 (zweimal)
- b) 9089310

6. allgemein bekannte Nummern

- a) 11833
- b) 01013-675381
- c) 0800-675381
- d) 0190-675381

7. auswendig gelernte Nummern

8. Nummern ohne spezielles Phänomen

- a) 6205497
- b) 2364286
- c) 4637415
- d) 4612745

## A.2 Kreditkartennummern

- Format: vier vier vier vier

### 1. aufeinanderfolgende gleiche Ziffern

#### a) Zweierblock

- i. 6332 2772 5573 8622

#### a) Dreierblock

- i. 2444 8887 4856 2596 (einmal drei Mitte)

### 2. aufeinanderfolgende Ziffernfolgen

#### a) aufsteigend

- i. 6783 9234 1567 7895

#### b) absteigend

- i. 5437 1987 2564 8769 (Anfang)

### 3. volle größere Zahlen

#### a) ganze Zehner

- i. 6307 2052 5790 6492

#### b) ganze Hunderter

- i. 6300 5004 6507 1005

### 4. wiederholte Ziffern

- a) 6367 5852 9584 7584

### 5. Nullen

- a) 5026 6308 9520 7584

### 6. auswendig gelernte Nummern

### 7. Nummern ohne spezielles Phänomen

- a) 6205 4975 3428 9642

- b) 2364 2865 1426 8359

## A.3 Bankverbindungen

### A.3.1 Bankleitzahlen

1. aufeinanderfolgende gleiche Ziffern
  - a) Zweierblock
    - i. 633 277 89
  - a) Dreierblock
    - i. 444 888 45
2. aufeinanderfolgende Ziffernfolgen
  - a) aufsteigend
    - i. 678 923 67
  - b) absteigend
    - i. 154 987 56
3. volle größere Zahlen
  - a) ganze Zehner
    - i. 630 205 90
  - b) ganze Hunderter
    - i. 300 501 07
4. wiederholte Ziffern
  - i. 636 646 95
5. Nullen
  - i. 502 630 20
6. auswendig gelernte Nummern
7. Nummern ohne spezielles Phänomen
  - i. 625 497 42
  - ii. 264 286 14



## A.3.2 Kontonummern

### 1. aufeinanderfolgende gleiche Ziffern

#### a) Zweierblock

- i. 6335841
- ii. 6635841
- iii. 6355841
- iv. 6358841
- v. 6358441
- vi. 6358411

#### a) Dreierblock

- i. 6663584
- ii. 6333584
- iii. 6358884
- iv. 6358444

### 2. aufeinanderfolgende Ziffernfolgen

#### a) aufsteigend

- i. 6782367
- ii. 6781345

#### b) absteigend

- i. 15498756
- ii. 5437946

### 3. volle größere Zahlen

#### a) ganze Zehner

- i. 63020590

#### b) ganze Hunderter

- i. 30050107

### 4. wiederholte Ziffern

- i. 63664695

### 5. Nullen

- i. 50263020

## B R-Skripte

```
#####  
#           #Daten einlesen #           #  
#####  
  
#Einzeltabellen einlesen  
BLZ ← read.csv("ne-blz.csv", header = TRUE)  
KTOT ← read.csv("ne-ktot.csv", header = TRUE)  
KR ← read.csv("ne-kr.csv", header = TRUE)  
  
#in Variable Dats speichern  
Dats ← c("BLZ", "KTOT", "KR")  
  
#####  
#           Daten ergaenzen #           #  
#####  
  
library(gdata) #um die fkt drop.levels zu haben  
  
# Entfernen der Beobachtungen ohne Nummernfeature  
# aus allen Datensatzen  
for (d in Dats){  
  assign(gsub("$", ".copy", d), get(d))  
  assign(d, subset(get(d), get(d)[["Nummernfeature"]] != "ohne"))  
  assign(d, drop.levels(get(d)))  
}  
  
# Hinzufuegen eines Vektor, der das Vorkommen  
# von abweichender Wortebung markiert  
for(d in Dats) {  
  assign(d,  
    cbind(get(d),  
          Wortvorhandenheit=  
          as.factor(ifelse(is.na(  
            get(d)  
            [["abw.wording.an.feature"]]),  
            "no.wording", "wording"  
          ))  
    )))  
}
```

```
#####
#           Kodierung der Dummy-Variablen           #
#####
```

```
#fuer Nummernfeature
for (d in Dats){
  for (k in levels(get(d)[["Nummernfeature"]])){
    eval(parse(text=paste("assign(d, cbind(get(d),",
                          k,
                          "=as.factor(ifelse(
                              get(d)[['Nummernfeature']]=k,
                              1,
                              0))))",
                          sep="")
          )
    )
  }
}
```

```
#fuer Wortgebung
for(d in Dats) {
  eval(parse(text=
    paste(d,
          "$Wortgebung—as.character(",
          d,
          "$Wording)",
          sep="")
    )
  )
  eval(parse(text=
    paste(d,
          "$Wortgebung—ifelse(",
          d,
          "$Wortgebung==',',keine.W)",
          d,
          "$Wortgebung)",
          sep="")
    )
  )
}
```

```

#####
#           weitere Datenergaenzungen           #
#####

#aus den Labels "hund", "thous", "tens" und "x"
#die jeweiligen Wortgebungskategorien erzeugen
for(d in Dats) {
  #in Characters umwandeln, so dass gegrept werden kann
eval(parse(text=paste(
    d,
    "$Wortgebung—as.character(",
    d,
    "$Wortgebung)",
    sep="")
  )
)
  #hunderter:
eval(parse(text=paste(
    d,
    "$Wortgebung[grep('hund',as.character(",
    d,
    "$Wording))]<- 'als.100'",
    sep="")
  )
)
  #tausender:
eval(parse(text=paste(
    d,
    "$Wortgebung[grep('thous',as.character(",
    d,
    "$Wording))]<- 'als.1000'",
    sep="")
  )
)
  #zehner:
eval(parse(text=paste(
    d,
    "$Wortgebung[grep('tens',as.character(",
    d,
    "$Wording))]<- 'als.10'",
    sep="")
  )
))

```

```

    #x mal die y:
eval(parse(text=paste(
    d,
    "$Wortgebung[ grep ('x', as.character(",
    d,
    "$Wording)])← 'als.mal'",
    sep=""
    )
    )
)
}

#Dummy-Variablen fuer Wortgebung
for(d in Dats){
    eval(parse(text=
        paste(
            "for (k in levels(as.factor(",
            d,
            "BLZ",
            "$Wortgebung))){" ,
            d,
            "[[k]]← as.factor( ifelse(",
            d,
            "$Wortgebung==k,1,0))}" ,
            sep=""
        )
    )
)
}

### Ermittlung der Featureposition
### aus der Repraesentation fuer die Featurelokalisierung
### (grep nach Buchstaben am Featureanfang/-mitte/-ende)
for(d in Dats){
    eval(parse(text=
        paste(d,
            "$Featureposition←as.character(",
            d,
            "$Featurelokalisierung)",
            sep=""
        )
    )
)
eval(parse(text=
    paste(d,
        "$Featureposition [ grep ( '^ [a-z] ', as.character (",

```

```

        d,
        "$Featurelokalisierung))]<'Anfang'",
        sep=""
    )
)
)

eval(parse(text=
    paste(d,
        "$Featureposition [grep ('[a-z]$', as.character(",
        d,
        "$Featurelokalisierung))]<'Ende'",
        sep=""
    )
)
)
eval(parse(text=
    paste(d,
        "$Featureposition [grep ('^[^a-z].*[a-z]+.*[^a-z]$',
        as.character(",
        d,
        "$Featurelokalisierung))]<'Mitte'",
        sep=""
    )
)
)
}

#Dummyvariablen fuer "Featureposition"
for (d in Dats){
  for (k in levels(as.factor(get(d)[["Featureposition"]]))){
    eval(parse(text=paste("assign(d, cbind(get(d),",
        k,
        "=as.factor(ifelse (
            get(d)[['Featureposition']]==k,
            1,
            0))))",
        sep=""
    )
    )
  }
}

```

```

#Dummyvariablen fuer "Sprecher"
for (d in Dats){
  for (k in levels(as.factor(get(d)[["Sprecher"]]))){
    eval(parse(text=paste("assign(d, cbind(get(d),",
      k,
      "=as.factor(ifelse(get(d)[['Sprecher']]==k,1,0)))",
      sep="")
    )
  )
}
}

```

```

#Dummyvariablen fuer "ntes.Vorkommen"
for (d in Dats){
  assign(d,
    cbind(get(d),
      "erstes" = as.factor(ifelse(
        get(d)[['ntes.Vorkommen']=="1",
        1,
        0
      )
    )
  )
  assign(d,
    cbind(get(d),
      "zweites" = as.factor(ifelse(
        get(d)[['ntes.Vorkommen']=="1",
        1,
        0
      )
    )
  )
  assign(d, cbind(get(d),
    "drittes" = as.factor(ifelse(
      get(d)[['ntes.Vorkommen']=="1",
      1,
      0))))
  assign(d, cbind(get(d),
    "viertes" = as.factor(
      ifelse(get(d)[['ntes.Vorkommen']=="1",
      1,
      0
    ))))
}

```

```

## ip- und IP-Phrasengrenzen zusammenfassen
## zu bound.after.F/bound.before.F
for(d in Dats){
  eval(parse(text=
    paste(d,
      "$bound.after.F—ifelse(",
      d,
      "$ip.after.F==1]",
      d,
      "$IP.after.F==1,1,0)",
      sep=""
    )
  )
  eval(parse(text=
    paste(d,
      "$bound.before.F—ifelse(",
      d,
      "$ip.before.F==1]",
      d,
      "$IP.before.F==1,1,0)",
      sep=""
    )
  )
}

# in allen Variablen eventuell auftretende Leerzeichen ersetzen
# und: Speicherung als Faktor, nicht als numerischen Vektor
for(d in Dats){
  for (n in names(get(d))){
    eval(parse(text=paste(d,
      "$",
      n,
      "←gsub(' ',',',as.character(get(d)[[n]]))",
      sep=""
    )
  )
  eval(parse(text=paste(d,
    "$",
    n,
    "←as.factor(get(d)[[n]])"
  )))}}

```



```

# Speicherung aller Datensaeetze in einem grossen Datensatz:
temp ← rbind(BLZ,KTOT)
ALL ← rbind(temp,KR)

# Hinzufuegen zur Liste der Datensaeetze
Dats ← c(Dats,"ALL")

### Datensaeetze aufspalten in diejenigen Beobachtungen mit
### und diejenigen ohne abweichende Wortgebung
AllDats←vector()
for (d in Dats){
  AllDats ← c(AllDats,d)
  w ← paste(d,"W",sep=".")
  eval(parse(text=paste(
    w,
    "←subset(",
    d,
    ",",
    d,
    "$Wortgebung!= 'keine.W'",
    sep="")
  )
  )
  AllDats ← c(AllDats,w)
  n ← paste(d,
    "N",
    sep=".")
  )
  eval(parse(text=paste(
    n,
    "←subset(",
    d,
    ",",
    d,
    "$Wortgebung== 'keine.W'",
    sep="")
  )
  )
  AllDats ← c(AllDats,n)
}

```

```

## Funktion zum Entfernen des letzten Elements aus einem Vektor
pop← function(vektor){
  return(vektor[1:length(vektor)-1])}

#Fallenlassen der nicht vorkommenden Auspraegungen
for(d in AllDats){
eval(parse(text=paste(d,"←drop.levels(",d,")",sep=""))
)
}

#####
###           weitere Funktionen           ###
#####

## Chi-Quadrat-Test auf eine ganze Matrix anwenden
chi.dat← function(matr, dat){
  for (r in rownames(matr)){
    print(r) #DEBUG
    for (c in colnames(matr)){
      print(c) #DEBUG
      matr[r,c]←chisq.test(dat[[r]], dat[[c]])$p.value}}
  return(matr)}

## Funktion fuer diejenigen Matritzen,
## die bound.before, oder after enthalten
## -> Features am Anfang/Ende muessen ausgeschlossen werden
chi.dat.part← function(matr, dat){
for (r in 1:nrow(matr)){
  for (c in 1:ncol(matr)){
    if(length(grep("after",rownames(matr)[r]))>0){
      TRY←"NODAT"
      try(TRY,
        chisq.test(dat[[rownames(matr)[r]]
          [dat$Featureposition!="Ende"],
          dat[[colnames(matr)[c]]
          [dat$Featureposition!="Ende"]])$p.value
        )
      matr[r,c] ← TRY
    }else{
      if(length(grep("before",rownames(matr)[r]))>0){
        TRY←"NODAT"
        try(TRY,
          chisq.test(
            dat[[rownames(matr)[r]][dat$Featureposition!="Anfang"],
            dat[[colnames(matr)[c]]
          )
        )
      }
    }
  }
}

```

```

                                [dat$Featureposition !=" Anfang" ])$p.value
        )
        matr[r,c] ← TRY
    }else{
        matr[r,c]←
            chisq.test(
                dat[[rownames(matr)[r]]],
                dat[[colnames(matr)[c]]])$p.value
    }
}
}
}
}
return(matr)
}

# Funktion zum Testen einer Matrix auf Signifikanz
# output: Signifikanzmatrix
# "sig" falls signifikant, "—" falls nicht
# Das Signiveau kann mit p=ZAHL angegeben werden, default ist 0.05.
# Gebrauch der fkt: sig(matr=MATRIX,p=ZAHL) oder sig(matr=MATRIX)
sig ←
    function(p=0.05,matr){
s←matr
for(r in rownames(s)){
for(c in colnames(s)){
    if(matr[r,c]=="NODAT"){
        #keine Daten --> kein numerischer Vergleich moeglich
        s[r,c] ← "—"
    }else{
        #numerischer Vergleich
        if (as.double(matr[r,c])<p) { #als Zahl
            #Wert ist kleiner Signifikanzniveau
            s[r,c] ← "sig"
        } else {
            #Wert ist groesser Signifikanzniveau
            s[r,c]← "—"
        }
    }
}
}
}
return(s)
}

```

```

# Funktion , zum genaueren Einordnen der p-Werte
#"sig" falls signifikant , "<0.07", falls ... ,
# "<0.01", falls .... , "-----" falls nicht
p.wert ←
  function(m)      #matrix
  {p.w ← m        #kopie fuer ergebnisse der ueberpruefung
  for(r in rownames(p.w)){
  for(c in colnames(p.w)){
    if (m[r,c]<0.05) {#bei <0.05 melde signifikant
      p.w[r,c]←paste("sig",m[r,c])
    } else {
      if (m[r,c]<0.07) {
        p.w[r,c]←paste("<0.07",m[r,c]) #kleiner p.Wert
      } else {
        if (m[r,c]<0.1){
          p.w[r,c]←"<0.1" #bisschen groesserer p-Wert
        } else {
          p.w[r,c]←paste("-----",m[r,c])
        }
      }
    }
  }
  }
}
return(p.w)
}

```

```

# Funktion zum schnellen Erzeugen einer Matrix
matr← function(rows,cols) {
  matrix(
    nrow=length(rows),
    ncol=length(cols),
    dimnames=list(
      rows,
      cols
    )
  )
}

```

```

#### Funktion, die testet, ob Matrizen fuer Datensatz Sinn machen
#Problem: In manchen Datensatzen gibt es weniger Daten,
#         so dass bei manchen Variablen nur 1 Auspraegung vorkommt
#         -> Chiquad-Test kann nicht angewendet werden!
#Diese Variablen werden aus der Matrix fuer den konkreten
#Datensatz entfernt

```

```

check.matr.dat ← function(matr, dat){
  cutrows←vector()
  cutcols←vector()
  newcolnames←colnames(matr)
  newrownames←rownames(matr)
  for (r in rownames(matr)){
    if (length(levels(dat[[r]])) < 2){
      cutrows←c(cutrows, r)
    }
  }
  for(c in colnames(matr)){
    if (length(levels(dat[[c]])) < 2){
      cutcols←c(cutcols, c)
    }
  }
  if(length(cutrows)>0){
    newrownames←setdiff(rownames(matr), cutrows)
  }
  if(length(cutcols)>0){
    newcolnames←setdiff(colnames(matr), cutcols)
  }
  return(matr(newrownames, newcolnames))
}

```

```

#####
# Erstellung der Matrizen, die getestet werden #####
#####

```

```

#loeschen aller anderen Objekte, die noch mit Chi anfangen:
rm(list = ls(pattern="^CHI"))

```

```

#Allgemein: welche Variable korreliert mit welcher anderen?
#-> alle relevanten Var
Rows=c("Gruppierung", "Wortgebung")
Cols=c("Sprecher",
        "Nummernfeature",
        "Featurevorhandenheit",
        "Featureposition",
        "ntes.Vorkommen")

```

```

CHI←
matr(rows=Rows, cols=Cols)

# Welches Nummernfeature korreliert mit
# Gruppierungskategorie & Wortgebung?
# -> Auspraegungen von Nummernfeature mit
# Gruppierungskategorie und W.
Rows=c("Gruppierung", "Wortgebung")
CHI.LEV.F ←
  matrix(
    nrow=length(Rows),
    ncol=length(levels(KTOT$Nummernfeature)),
    dimnames=list(Rows, levels(KTOT$Nummernfeature))
  )

# Welche Wortgebung korreliert mit welchem Nummernfeature?
# -> Auspraegungen von Wortgebung und Nummernfeature
CHI.LEV.WF ←
  matrix(
    nrow=length(levels(KTOT$Wortgebung)),
    ncol=length(levels(KTOT$Nummernfeature)),
    dimnames=list(
      levels(KTOT$Wortgebung),
      levels(KTOT$Nummernfeature)
    )
  )

# Welche Gruppierungskategorien korrelieren zu welchen Nummernfeatures?
# -> Auspraegungen von Gruppierungskategorie und Nummernfeature
# -> zusaetzlich noch 2 neue Kategorien
# fuer Phrasengrenzen allegemein (ip&IP)
CHI.LEV.GF ←
matrix(
  nrow=length(levels(KTOT$Gruppierungskategorie))+2,
  ncol=length(levels(KTOT$Nummernfeature)),
  dimnames=list(
    c(gsub(" ", "", levels(KTOT$Gruppierungskategorie)),
      "bound. before .F",
      "bound. after .F"
    ),
    levels(KTOT$Nummernfeature))
  )

```

```

# Welche Gruppierungskategorien korrelieren zu welcher Featureposition?
CHI.LEV.GFP ←
  matrix(
    nrow=length(levels(KTOT$Gruppierungskategorie))+2,
    ncol=length(levels(KTOT$Featureposition)),
    dimnames=list(
      c(gsub(" ", "", levels(KTOT$Gruppierungskategorie)),
        "bound.before.F",
        "bound.after.F"
      )
    ,
    levels(KTOT$Featureposition)
  ))

# Welche Wortgebungekategorien korrelieren zu welcher Featureposition?
Rows=levels(KTOT$Wortgebung)
Cols=levels(KTOT$Featureposition)
CHI.LEV.WFP ← matr(Rows, Cols)

# Welche Auspraegungen von Worgebung korrelieren
# zu den anderen Variablen?
Rows=levels(BLZ$Wortgebung)
Cols=c(" Sprecher", " Featureposition", " ntes.Vorkommen")
CHI.LEV.W.AK ←
  matr(rows=Rows, cols=Cols)
# Welche Auspraegungen von Gruppierungskategorie
# korrelieren zu den anderen Variablen?
Rows=levels(KTOT$Gruppierungskategorie)
CHI.LEV.G.AK ←
  matr(Rows, Cols)

# Welche Auspraegungen von Gruppierungskategorie
# korrelieren zu welcher Auspraegung von Sprecher?
CHI.LEV.GS ←
  matrix(
    nrow=length(levels(KTOT$Gruppierungskategorie))+2,
    ncol=length(levels(KTOT$Sprecher)),
    dimnames=list(
      c(gsub(" ", "", levels(KTOT$Gruppierungskategorie)),
        "bound.before.F",
        "bound.after.F"),
      levels(KTOT$Sprecher))
  )

```

```

# Welche Auspraegungen von Wortgebung korrelieren
# zu welcher Auspraegung von Sprecher?
Rows=levels(KTOT$Wortgebung)
Cols=levels(KTOT$Sprecher)
CHI.LEV.WS ←
  matr(Rows, Cols)

#Welche Auspraegungen von Gruppierungskategorie
#korrelieren zu welcher Auspraegung von ntes.Vorkommen?
CHI.LEV.GV ←
  matrix(
    nrow=length(levels(KTOT$Gruppierungskategorie))+2,
    ncol=4,
    dimnames=list(
      c(gsub(" ", "", levels(KTOT$Gruppierungskategorie)),
        "bound.before.F",
        "bound.after.F"
      )
    ,
      c("erstes", "zweites", "drittes", "viertes")
    )
  )

#Welche Auspraegungen von Wortgebung korrelieren zu
# welchen Auspraegungen von ntes.Vorkommen?
Rows=levels(KTOT$Wortgebung)
Cols=c("erstes", "zweites", "drittes", "viertes")
CHI.LEV.WW ←
  matr(Rows, Cols)

```



```

#####
### Testen der Matrizen ###
#####

# Globbing: alle Objekte, die Pattern entsprechen,
# hier: die mit Chi anfangen sammeln:
Chis ← ls(pattern="^CHI")

#Liste erzeugen, in der die Ergebnisse der Tests gespeichert werden
# Struktur der Ergebnisse: Erg$Datensatz$Ergebnistabelle

#pWerte:
pErg←list()
for (d in AllDats){
for (c in Chis){
  if(ncol(check.matr.dat(get(c),get(d)))==0
    ||
    nrow(check.matr.dat(get(c),get(d)))==0
  ){
    pErg[[d]][[c]]←"NODAT"
  }else{
    if (c=="CHI.LEV.GF" | c=="CHI.LEV.GFP" | c=="CHI.LEV.GS"){
      pErg[[d]][[c]]←chi.dat.part(check.matr.dat(get(c),get(d)),get(d))
    }else{
      pErg[[d]][[c]]←chi.dat(check.matr.dat(get(c),get(d)),get(d))}}}}

#signifikanzen:
sErg←list()
for (d in AllDats){
  print(d)
for (c in Chis){
  print(c)
  if(ncol(check.matr.dat(get(c),get(d)))==0
    || row(check.matr.dat(get(c),get(d)))==0
  ){
    sErg[[d]][[c]]←"NODAT"
  }else{
    if (c=="CHI.LEV.GF" | c=="CHI.LEV.GFP"){
      sErg[[d]][[c]]←
      sig(p=0.05,matr=chi.dat.part(check.matr.dat(get(c),get(d)),get(d)))
    }else{
      sErg[[d]][[c]]←
      sig(p=0.05,matr=chi.dat(check.matr.dat(get(c),get(d)),get(d))}}}}

```

```

#unterschiedliche signifikanzniveaus/p-werte
spErg←list ()
for (d in AllDats){
for (c in Chis){
  if(ncol(check.matr.dat(get(c),get(d)))==0
  ||
  nrow(check.matr.dat(get(c),get(d)))==0
  ){
  spErg[[d]][[c]]←"NODAT"
} else {
  if(c=="CHI.LEV.GF" | c=="CHI.LEV.GFP"){
  spErg[[d]][[c]]←
  p.wert(chi.dat.part(check.matr.dat(get(c),get(d)),get(d)))
} else {
  spErg[[d]][[c]]←
  p.wert(chi.dat(check.matr.dat(get(c),get(d)),get(d)))
}
}
}
}
}

```

```

#####
# Ergebnisse in Latex-File schreiben #
#####
#ins Ergebnisverzeichnis wechseln,
setwd("Ergebnisse")
library(quantreg) #latex.table

```

```

##### Ergebnisse in Latex-Tabellen schreiben #####
Text←vector ()
Text["CHI"]=" alle Variablen"
Text["CHI.LEV.F"]=" die Auspr\`agungen von
Nummernfeature mit den Response-Variablen"
Text["CHI.LEV.GF"]=" die Auspr\`agungen von
Gruppierung mit den Auspr\`agungen
von Nummernfeature"
Text["CHI.LEV.WF"]=" die Auspr\`agungen von Wortgebung
mit den Auspr\`agungen von Nummernfeature"
Text["CHI.LEV.GFP"]=" die Auspr\`agungen von Gruppierung
mit den Auspr\`agungen von Featureposition"
Text["CHI.LEV.WFP"]=" die Auspr\`agungen von Wortgebung
mit den Auspr\`agungen von Featureposition"
Text["CHI.LEV.W.AK"]=" die Auspr\`agungen von Wortgebung
mit weiteren Variablen"
Text["CHI.LEV.G.AK"]=" die Auspr\`agungen von Gruppierung
mit weiteren Variablen"

```

```

Text["CHI.LEV.GS"]="die Auspr\`agungen von Gruppierung
                    mit den verschiedenen Sprechern"
Text["CHI.LEV.WS"]="die Auspr\`agungen von Wortgebung
                    mit den verschiedenen Sprechern"
Text["CHI.LEV.GV"]="die Auspr\`agungen von Gruppierung
                    mit verschiedenen Mehrfachvorkommen"
Text["CHI.LEV.WV"]="die Auspr\`agungen von Wortgebung
                    mit verschiedenen Mehrfachvorkommen"

```

```

## Anonymisierung der Sprecher: Vorname_Nachname -> SP-VN
SubsChi ← c("CHI.LEV.GS", "CHI.LEV.WS")
for ( s in SubsChi){
  for(d in AllDats){
    eval(parse(text=paste(" if (sErg$",
                          d,
                          "[s] != 'NODAT' ){ colnames(sErg$",
                          d,
                          "$",
                          s,
                          ") ← gsub( ' ([a-z])[a-z]*_([a-z])[a-z]* ',
                          'SP-\\\\1\\\\2 ', colnames(sErg$",
                          d,
                          "$",
                          s,
                          "))) }",
                          sep="" )
    )
  }
}

```

```

#Ergebnisfile fuer alle Ergebnisse
if(file.exists("alle.tex")){file.remove("alle.tex")}

#tex files speichern
for(d in AllDats){
for(c in Chis){
  #nur belegte Matritzen drucken
  if(is.matrix(sErg[[d]][[c]])){
    if(nrow(sErg[[d]][[c]]) > 0 && ncol(sErg[[d]][[c]]) > 0){
      latex.table(sErg[[d]][[c]],
        caption=
        paste("Signifikanzmatrix des \\chiquad -Tests f\"ur",
          Text[c], "beim Datensatz", d),
        file=paste(d,c,sep="-"),
        rowlabel="\\chiquad",
        label= paste("tab",paste(d,c,sep="."),sep=":")
        ) #einzelfiles
      latex.table(sErg[[d]][[c]],
        caption=
        paste("Signifikanzmatrix des \\chiquad -Tests f\"ur",
          Text[c], "beim Datensatz", d),
        file="alle",
        rowlabel="\\chiquad",
        append=TRUE) #ein grosses file
    }else{
      print(paste("nrow oder ncol kleiner Null bei",d,"und",c))
    }
  }else{print(paste("keine Ergebnisse fuer",d,"und",c))
  }
}
}}

##### weitere Tests #####

# jede Art von Akzentverschiebungen am Feature in einem Faktor:
KTOT.N$move.accent.R←
  as.factor(ifelse(KTOT.N$akzentverschiebung.feature.auf1 ==1
    | KTOT.N$akzentverschiebung.feature.auf4 ==1,1,0))
KTOT$move.accent.F ←
  as.factor(ifelse(KTOT$akzentverschiebung.feature.auf1 ==1
    | KTOT$akzentverschiebung.feature.auf4 ==1,1,0))

```

```

KR.N$move.accent.F ←
  as.factor( ifelse (KR.N$akzentverschiebung.feature.auf1 ==1
                    | KR.N$akzentverschiebung.feature.auf4 ==1,1,0))
KR$move.accent.F ←
  as.factor( ifelse (KR$akzentverschiebung.feature.auf1 ==1
                    | KR$akzentverschiebung.feature.auf4 ==1,1,0))
BLZ.N$move.accent.F ←
  as.factor( ifelse (BLZ.N$akzentverschiebung.feature.auf1 ==1 |
                    BLZ.N$akzentverschiebung.feature.auf4 ==1,1,0))
BLZ$move.accent.F ←
  as.factor( ifelse (BLZ$akzentverschiebung.feature.auf1 ==1
                    | BLZ$akzentverschiebung.feature.auf4 ==1,1,0))

```

```

# Chiquad-Test fuer neuen Faktor mit Feature zehn

```

```

chisq.test(KTOT.N$zehn, KTOT.N$move.accent.F)
chisq.test(KTOT$zehn, KTOT$move.accent.F)
chisq.test(KR.N$zehn, KR.N$move.accent.F)
chisq.test(KR$zehn, KR$move.accent.F)
chisq.test(BLZ.N$zehn, BLZ.N$move.accent.F)
chisq.test(BLZ$zehn, BLZ$move.accent.F)

```

```

#####
# Funktion zum testen, ob positive oder negative Korrelation #
#####

```

```

check.corr ← function (var1, var2){
  t ← table(var1, var2)
  if (t[4]/t[2] > t[3]/t[1]){
    print("+")
  } else if (t[4]/t[2] < t[3]/t[1]){
    return("-")
  } else {
    return("?")
  }
}
check.corr.dat ← function (var1, var2, dat){
  attach(dat)
  t ← table(var1, var2)
  if (t[4]/t[2] > t[3]/t[1]){
    print("+")
  } else if (t[4]/t[2] < t[3]/t[1]){
    print("-")
  } else {
    print("?")
  }
  detach(dat)
}

```

# C Alternativer Algorithmus für die Wortgebung von Hunderterzahlen

---

**Algorithmus 12** Bestimmung der Wortgebung von Bankleitzahlen, sowie Konto- und Telefonnummern beim Auftreten von vollen Hunderterzahlen

---

```
read numbers
number = numbers[1]                                ▷ Identität der ersten Ziffer
wording = append(number, "hundert")
words[0] = wording
for i=1 to feature.length-1 do                   ▷ die restlichen Positionen bleiben leer
    words[i]=no-wording
end for
return words
```

---

# Literaturverzeichnis

## **Backhaus u. a. 2006**

BACKHAUS, K. ; ERICHSON, B. ; PLINKE, W. ; WEIBER, R.: *Multivariate Analysemethoden*. 11. Berlin, Heidelberg, New York : Springer, 2006. – Statistik Lehrbuch

## **Baumann u. Trouvain 2001**

BAUMANN, S. ; TROUVAIN, J.: On the Prosody of German Telephone Numbers. In: DALSGAARD, P. (Hrsg.) ; LINDBERG, B. (Hrsg.) ; BENNER, H. (Hrsg.): *7th European Conference on Speech Communication and Technology (EUROSPEECH '01)*, September 3–7. Aalborg, Denmark, 2001, S. 557–560

## **Black 1997**

BLACK, A. W.: *The Festival speech synthesis system*.  
<http://www.cstr.ed.ac.uk/projects/festival.html>, November 1997

## **Breuer u. Abresch 2003**

BREUER, S. ; ABRESCH, J.: Unit selection speech synthesis for a directory enquiries service. In: *Proceedings of the 15th International Congress of Phonetic Sciences (Barcelona)*, 2003, S. 2573–2576

## **Deutsches Institut für Normung e.V. 1996**

DEUTSCHES INSTITUT FÜR NORMUNG E.V. (Hrsg.): *Sonderdruck DIN 5008*. Berlin, Wien, Zürich, 1996

## **Falé u. Faria 2006**

FALÉ, I. ; FARIA, I. H.: Categorical perception of intonational contrasts in European Portuguese. In: *Proceedings of Speech Prosody 2006 (Dresden)*, 2006, S. 69–72

## **Grice u. Benz Müller 1995**

GRICE, M. ; BENZMÜLLER, R.: Transcription of German Intonation using ToBI tones; The Saarbrücken System. In: *Phonus 1* (1995), S. 33–51

## **House 1996**

HOUSE, D.: Differential perception of tonal contours through the syllable. In: *Proceedings of the International Conference on Spoken Language Processing (Philadelphia, PA)* Bd. 1, 1996, S. 2048–2051

## **Jacobi u. Reichel 2003**

JACOBI, I. ; REICHEL, U. D.: A grammar of intonational units in German digit numbers. In: *Proceedings of the 15th International Congress of Phonetic Sciences (Barcelona)*, 2003, S. 591–594

**Kohler 1987**

KOHLER, K. J.: Categorical pitch perception. In: *Proceedings of the 11th International Congress of Phonetic Sciences* Bd. 5. Tallinn, 1987, S. 331–333

**Ladd 1996**

LADD, R.: *Intonational phonology*. Cambridge University Press, 1996

**Mayer 1995**

MAYER, J.: Transcribing German Intonation - The Stuttgart System / Universität Stuttgart. 1995. – Forschungsbericht

**Miller 1956**

MILLER, G. A.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. In: *The Psychological Review* 63 (1956), S. 81–97

**Pierrehumbert 1980**

PIERREHUMBERT, J. B.: *The Phonology and Phonetics of English Intonation*, Massachusetts Institute of Technology, Diss., September 1980

**R Development Core Team 2006**

R DEVELOPMENT CORE TEAM: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2006. – ISBN 3-900051-07-0. <http://www.R-project.org>

**Rapp 1995**

RAPP, S.: Automatic phonemic transcription and linguistic annotation from known text with Hidden Markov models—An aligner for German. In: *Proceedings of ELS-NET Goes East and IMACS Workshop "Integration of Language and Speech in Academia and Industry"* (Moskau, Russland), 1995

**Schneider u. Lintfert 2003**

SCHNEIDER, K. ; LINTFERT, B.: Categorical perception of boundary tones in German. In: *Proceedings of the 15th International Congress of Phonetic Sciences (Barcelona)*, 2003, S. 631–634

**Simonsen u. a. 2004**

SIMONSEN, B. ; KLABBERS, E. ; POLS, L.: Synthesised Speech with Unit Selection. In: *Proceedings* Bd. 24. University of Amsterdam, 2004, S. 57–63

**Sjölander u. Beskow 2000**

SJÖLANDER, K. ; BESKOW, J.: WaveSurfer - an open source speech tool. In: *Proceedings of ICSLP* Bd. 4, 2000, S. 464–467