Universität Stuttgart
Lehrstuhl für Experimentelle Phonetik
Azenbergstr. 12, 70174 Stuttgart

Studienarbeit Nr. 120

Automatische Generierung von Aussprachevarianten für die automatische Annotation deutscher Spontansprache

Das Spontaneous Speech Tool für den IMS Aligner

von
Natali Mavrović
Matrikelnummer 2077413

Betreuer: Dr. Antje Schweitzer
Prüfer: Prof. Dr. Grzegorz Dogil

Anfang:18. März 2011
Ende: 26. Mai 2011

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.

Stuttgart, den 26. Mai 2011

Natali Mavrović

Zusammenfassung

Gesprochene Sprache kann in ihrer Realisierung große Unterschiede aufweisen. Ein Sprecher kann äußerst aufmerksam auf seine Aussprache achten, oder, im Gegenteil, er äußert seinen Gedankenfluss eher nachlässig, und lässt somit dialektale Färbungen sowie stärkere koartikulatorische Änderungen zu.

Für automatische Segmentierungssysteme ist die hohe Variabilität der spontanen Sprache eine besondere Herausforderung. Das System muss zu der eigentlichen Aussprache eines Wortes auch die im Redefluss erzeugten Varianten dieses Wortes segmentieren können. Aus diesem Grund werden automatische Segmentierungssysteme um Aussprachevarianten erweitert.

In der Literatur können zwei Grundansätze zur Erweiterung der Segmentierungssysteme um Varianten der spontanen Sprache ermittelt werden. Auf der einen Seite gibt es solche, welche die artikulatorischen Änderungen auf phonologische Regelmäßigkeiten zurückführen, d.h. die Änderung hängt von ihrem phonetisch-phonologischen Kontext ab. Auf der anderen Seite gibt es statistische Ansätze, welche einen Korpus nach den am häufigsten geänderten Wörtern durchsuchen, und diese Wörter im Lexikon des Systems um Varianten erweitern.

In dieser Arbeit präsentiere ich eine weitere Möglichkeit ein automatisches Segmentierungssystem um Aussprachevarianten zu erweitern. Der hier präsentierte Ansatz ist rein statistisch, berücksichtigt jedoch den phonetisch-phonologischen Kontext der sich verändernden Laute. Damit werden die bisherigen Ansätze in Frage gestellt sowie die Wahl der Grundeinheiten der bisherigen Ansätze (Phoneme, bzw. Wörter) diskutiert.

Als Resultat wird das Spontaneous Speech Tool vorgestellt, welches an das Segmentierungssystem des Instituts für maschinelle Sprachverarbeitung der Universität Stuttgart, den IMS Aligner, angeschlossen werden kann. Die Ergebnisse des Spontaneous Speech Tools werden anschließend evaluiert, sowie Verbesserungen zur Entwicklung solcher Tools vorgeschlagen.

Inhaltsverzeichnis

1 Einleitung **9**

1.1 Motivation 9

1.2 Gliederung der Arbeit 14

2 Grundlagen **17**

2.1 Automatische Segmentierung und Annotation von Sprachsignalen 17

2.2 IMS Aligner 18

2.2.1 Architektur des IMS Aligners 18

2.2.2 Grenzen des Aligners 20

2.3 Das Kiel Korpus 21

2.3.1 Das Kiel Korpus der Spontansprache 21

2.4 Methoden zur Ermittlung von Aussprachevarianten 26

2.4.1 Phonologische Prozesse als Regeln für Aussprachevarianten 26

2.4.2 Ermittlung von Aussprachevarianten durch Korpusanalyse 28

2.5 Diskussion der vorgestellten Ansätze und Präsentation eines neuen Ansatzes 29

2.5.1 Diskussion des phonologischen Ansatzes nach [Kemp, 1996] 29

2.5.2 Diskussion des statistischen Ansatzes nach [Sloboda et al., 1996] . 30

2.5.3 Statistischer Ansatz auf phonologischer Ebene 30

3 Aufbereitung der Daten **33**

3.1 Extraktion und Ergänzung der Informationen aus dem Korpus 34

3.2 Statistische Auswertung des Datensatzes 38

3.2.1 Analyse der Tilgung 39

3.2.2 Analyse der Glottalisierung 41

3.2.3 Analyse der Substitution 41

3.2.4 Analyse der Einfügungen von Lauten 42

3.2.5 Analyse der Nasalierung 44

3.3 Diskussion der Datenanalyse 44

4 Ausarbeitung des Spontaneous Speech Tools **47**
4.1 Ermittlung des besten Lernalgorithmus 47
4.1.1 WEKA: eine kompakte Übersicht 48
4.1.2 Experimente mit den Daten 50
4.1.2.1 Vorüberlegungen 50
4.1.2.2 Die Experimente 54
4.1.3 Zusammenfassung und Schlussfolgerungen der Experimente 59
4.2 Programmierung des Tools 60
4.2.1 Architektur des Spontaneous Speech Tools 61
4.2.2 Diskussion zur Funktionsweise des Spontaneous Speech Tools . . . 63
4.3 Anbindung an den IMS Aligner 66

5 Evaluierung **69**
5.1 Ergebnisse des Testens auf den Testdaten 69
5.2 Schlussfolgerungen 71

6 Zusammenfassung und Ausblick **73**

1 Einleitung

1.1 Motivation

Das Thema dieser Arbeit, „Automatische Generierung von Aussprachevarianten“, scheint auf den ersten Blick äußerst spezifisch zu sein. Erst im Laufe der Recherchen zum Thema habe ich die Erforschung der Generierung von Aussprachevarianten als elementaren Faktor der modernen Annotations- und Segmentierungstechnik erkannt. Um die Bedeutsamkeit der Erzeugung solcher Varianten zu veranschaulichen, möchte ich in der Einleitung den groben Weg eines Sprachsignals „vom Mikrofon zum Segmentierungssystem, und durch das Segmentierungssystem hindurch“ beschreiben.

Im Vorfeld werden ein paar wichtige Begriffe, wie *Annotation- und Segmentierungssystem*, *Spontansprache*, oder *Aussprachevarianten*, unter Berücksichtigung des Themas verdeutlicht.

Was ist ein Annotations- und Segmentierungssystem?

Ein Annotations- und Segmentierungssystem, meist nur Segmentierungssystem genannt, dient dazu, die Segmente eines Sprachsignals mit Etiketten, bzw. Markern oder Kennzeichnungen zu versehen. Ein solches System kann isoliert verwendet werden (es wird vom Nutzer direkt auf das Sprachsignal angewendet), oder es kann als Komponente eines größeren Systems implementiert werden.

Ein Beispiel für ein isoliertes Segmentierungssystem ist der IMS Aligner. Aus einem Sprachsignal und der orthografischen Transkription erzeugt es eine Liste mit den Endzeitpunkten der Segmente und, beispielsweise, ihren phonetischen Kennzeichnungen. Solche Systeme finden vor allem in der Forschung und der Untersuchung von sehr großen Sprachdatenbanken ihren Verwendungszweck.

Segmentierungssysteme als Komponente größerer Systeme sind dagegen auch für die kommerzielle Industrie von großem Interesse. Zu den einfachsten, zudem sehr erfolg-

reichen Produkten der Sprachtechnologie gehören beispielsweise Diktiersysteme. Diese Systeme nehmen kontinuierliche Sprache als Signal auf, und geben die orthografische Transkription als Resultat der Verarbeitung wieder. Um diese Aufgabe zu bewältigen hat ein Diktiersystem einen Eingebauten Spracherkenner, welcher die diktierte, bzw. vorgelesene Sprache verarbeiten kann. Damit das Diktiersystem auch spontan gesprochene Sprache erkennen kann, müsste es, zu den Grundaussprachen der Wörter, auch die durch den Redefluss abgewandelten Varianten verarbeiten können.

Ein weiteres, teilweise nicht ganz ausgereiftes, Beispiel aus der Sprachtechnologie sind Sprachdialogsysteme der künstlichen Intelligenz. Ein Nutzer des Systems stellt eine (gesprochene) Frage an das System, worauf das System inhaltlich auf die Frage in Form eines akustischen Sprachsignals antwortet. Das heißt, das System verfügt über einen Spracherkenner, eine semantische Repräsentation der Aussage, eine Dialog-Komponente und einen Sprachsynthetisator. Dialogsysteme müssen mit der gesprochenen bzw. spontanen Sprache des Nutzers zurecht kommen, und dies setzt die Erkennung von Varianten im Redefluss voraus.

Was ist Spontansprache, und wieso wird sie gesondert behandelt?

Die erste Instanz in der Verarbeitung gesprochener Sprache ist das Sprachsignal. Der Inhalt eines Sprachsignals kann sehr unterschiedlich sein. Datenbanken von Sprachsignalen dienen den verschiedensten sprachtechnologischen Anwendungen, und je nach Verwendungszweck der Aufnahme werden Sprecher und das Gesprochene vorab festgelegt oder nicht.

Betrachtet man, beispielsweise, das Erstellen eines Inventars für einen (Diphon-) Sprachsynthetisator, so müssen alle möglichen Parameter der Aufnahmen vorbestimmt und kontrolliert werden. Es fängt bereits beim Sprecher an: Er muss mit viel Bedacht ausgesucht werden, damit die Stimme auch nach der Synthese angenehm klingt. Der Sprecher muss alle Einheiten des Inventars mindestens einmal produzieren, meist in einem kontrolliertem phonetischen Kontext, und dies bedeutet, dass der Sprecher einen außerordentlich gut vorbereiteten Text vorlesen muss.

Allerdings ist es nicht immer möglich, oder sogar erwünscht, alle Parameter einer Sprachaufnahme im Voraus zu bestimmen. So unterscheidet sich ein Korpus für ein Sprachsyntheseinventar in großen Maßen von den Sprachsignalen, welche ein Dialogsystem verarbeitet. Würde ein Spracherkenner eines Dialogsystems auf Sprachdaten basieren, welche eine ebenso saubere Aussprache aufweisen wie die eines Diphonsynthesesystems, hätte er Schwierigkeiten eine Äußerung, die nicht mit der selben Genauigkeit artikuliert wurde,

korrekt zu erkennen.

Gerade am Beispiel eines Dialogsystems wird klar, dass es durchaus Situationen gibt, in welchen das zu erforschende Sprachsignal mehr Natürlichkeit aufweisen muss. Der natürliche, unvorbereitete und von außen[1] nicht beeinflusste Redefluss wird Spontansprache genannt.

Spontansprache unterscheidet sich deutlich von gelesener und vorbereiteter Sprache. Betrachtet man gesprochene Sprache unter dem mechanischen Aspekt, so ist Sprache eine Aneinanderreihung von artikulierten Lauten. Es liegt in der Natur des Menschen, jede Bewegung, so auch die artikulatorischen, mit minimalem Aufwand und maximaler Ergiebigkeit auszuführen. Folglich neigt der Sprecher dazu, seine Artikulatoren schnell und ungenau zu bewegen, jedoch ohne, dass die Aussage an Informationsverlust leidet [Lindblom, 1990].

Hat der Sprecher einen Dialogpartner, so wird er abhängig von der Situation versuchen, „so deutlich wie nötig, und so ungenau wie möglich“ zu sprechen. Hat der Dialogpartner Schwierigkeiten den Sprecher zu verstehen, so wird er es dem Sprecher mitteilen, und der Sprecher muss sich darauf einstellen (und z.B. deutlicher oder langsamer reden).

Spricht der Sprecher ohne einen Dialogpartner, oder liest einen Text vor, und ist sich der Sprecher bewusst, dass seine Äußerung später von jemandem angehört wird, so kann eine deutliche Änderung im Sprachverhalten beobachtet werden. Der Sprecher spricht deutlicher, langsamer, oder in sonst einer Weise kontrollierter als im Dialog. Man könnte spekulieren, dass solche VerhaltensÄnderungen den Grund im „unbekannten Zuhörer“ haben. Weiß der Sprecher nicht, wer der Hörer der Äußerung ist, so achtet er mehr auf die Korrektheit des Gesprochenen als im klassischen Dialog.

Als Folge dessen entstehen im natürlichen (Dialog-)Redefluss ungrammatische Sätze, die Sprache ist dialektal gefärbt, oder der Sprecher neigt zum schnellen Sprechen. Folglich weißt die Äußerung unzählige Koartikulationen auf. Dies stellt für Segmentierungssysteme und Spracherkenner eine Herausforderung dar, da die Varianten teilweise bis zur Unkenntlichkeit von einander abweichen, Beispiel: „siebenten“ wird von /`zi:b@nt@n`/ zu /`zi:mpm`/. Eine Möglichkeit dieses Problem anzugehen, ist die Verwendung von *Aussprachevarianten*.

[1] mit „außen“ sind hier äußere Einflüsse, wie z.B. bei der Herstellung des Sprachsyntheseinventars gemeint

Wie können Aussprachevarianten helfen?

Damit ein Segmentierungssystem VerÄnderungen im Redefluss verarbeiten kann, muss es lernen, mögliche Varianten einer Äußerung zu erkennen. Um zu verstehen, wo und wie Aussprachevarianten in einem Segmentierungssystem zum Einsatz kommen, wird als erstes ein grober Einblick in die Arbeitsweise eines solchen Systems gegeben (mehr über Segmentierungssysteme in 2.1).

Anhand des IMS Aligners kann der Prozess der Kennzeichnung von Segmenten der Signale veranschaulicht werden. Der IMS Aligner bekommt als Eingabe ein Sprachsignal und die zugehörige orthografische Transkription als Textdatei. Die Ausgabe ist eine Liste mit den Endzeitpunkten der Segmente und den zugehörigen Etiketten. Im folgenden wird die Entstehung einer Endzeitpunkt-Phonem-Liste beschrieben.

Aus der orthografischen Transkription werden die phonetischen Transkriptionen erstellt, indem die phonetische Schreibweise in einem Lexikon nachgeschlagen wird. Die Einträge eines phonetischen Lexikons, welche die realisierten Phoneme eines Wortes bei deutlichem Sprechen oder lesen beinhalten, werden auch *kanonische (Laut)-Form* genannt.

Die Liste der kanonischen Lautformen wird nun, durch zuvor erarbeitete und trainierte (mathematische) Phonemmodelle, als Netzwerk von Lauten dargestellt [URL: IMS]. In einem unabhängigen Prozess wird das Sprachsignal in Vektoren von Merkmalskoeffizienten zerlegt. Um die richtigen Endzeitpunkte der Segmente zu bestimmen, wird mit Hilfe des Netzwerks die wahrscheinlichste Folge von Segmenten mittels der Viterbi-Suche[2] ermittelt, die diese Merkmalsvektoren produziert haben könnte.

Eine Möglichkeit, Aussprachevarianten an ein Segmentierungssystem zu koppeln, ist, die Varianten nach dem Textverarbeitungsprozess zu generieren. Für jedes Wort der Äußerung kann zur gefundenen phonetischen Transkription im Lexikon eine zusätzliche Auswahl an phonetischen Varianten zur Verfügung gestellt werden. Der IMS Aligner entscheidet anschließend, welche Variante am besten mit dem Geäußerten übereinstimmt.

In Bild 1.1 ist diese Methode grob skizziert. Die schwarzen Pfeile stellen den Datenfluss eines konventionellen Segmentierungssystems dar. Die roten Pfeile zeigen den Datenfluss bei Einsatz eines Generators von Aussprachevarianten. Der Weg dargestellt durch den schwarzen Pfeil mit dem roten „X“ wird bei angeschlossenem Aussprachevariantengenerator nicht eingeschlagen.

[2]Viterbi-Suche: Suche nach dem optimalen Weg durch ein Netz von versteckten Zuständen

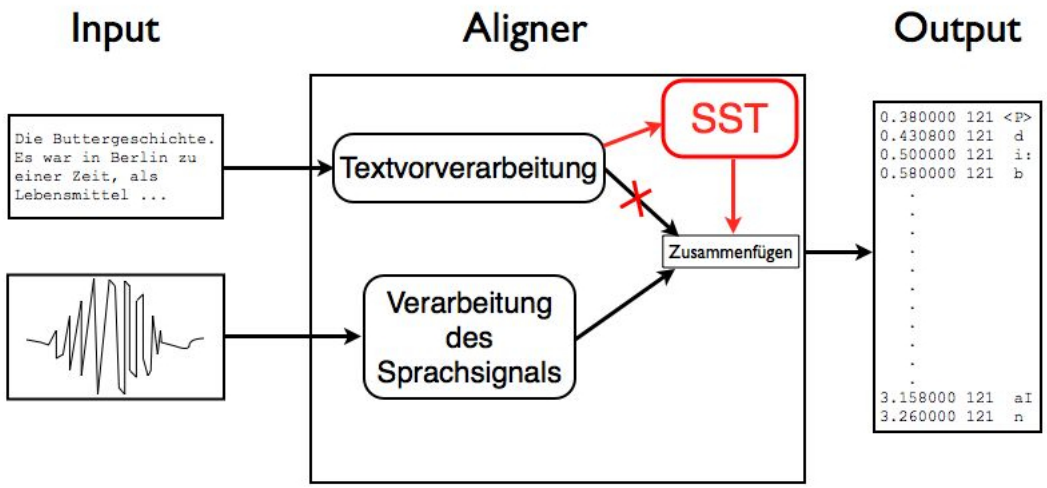

Abbildung 1.1: Architektur eines Segmentierungssystems mit eingebundenem Generator von Aussprachevarianten
(SST = Spontaneous Speech Tool)

Was ist das Spontaneous Speech Tool?

Das Spontaneous Speech Tool, in dieser Arbeit auch SST genannt, ist speziell für den IMS Aligner des Instituts für maschinelle Sprachverarbeitung der Universität Stuttgart entwickelt worden. Es wird eingesetzt, um dem Aligner zu den vorhandenen phonetischen Transkriptionen des eingebundenen Lexikons weitere Wortvarianten des freien Redeflusses zur Verfügung zu stellen.

Das Tool kommt zum Einsatz, nachdem der Aligner die Wörter im phonetischen Lexikon gefunden hat, und sie als Liste gespeichert hat. Diese Liste dient als Eingabe für das Tool. Es generiert aus der kanonischen Lautform alle möglichen spontansprachlichen Varianten, gibt die Liste mit den Varianten an den Aligner weiter, und der Aligner kann wie gewohnt den Prozessablauf aufnehmen (Aufbau des Netzwerks, Viterbi-Suche, usw.).

Die Abbildung 1.2 zeigt die grobe Architektur des SST. Ein- und Ausgabe sind jeweils Listen, der Kern des Tool ist ein trainiertes statistisches Modell.

Das Modell wurde auf einem von dem Kiel Korpus der Spontansprache [Kohler, 1995] hergeleiteten Datensatz trainiert. Der Inhalt des Datensatzes ist phonologischen Charakters, und enthält somit zu den Phonemen und ihrer lautlichen Umgebung auch Informationen über Artikulationsort, -art sowie Betonung. Das statistische Modell hat zur Aufgabe, die Änderung einer vorgegebenen Einheit, hier eines Phonems, vorherzusagen. Die vorhergesagten Laute ergeben somit eine Variante der eingelesenen Äußerung.

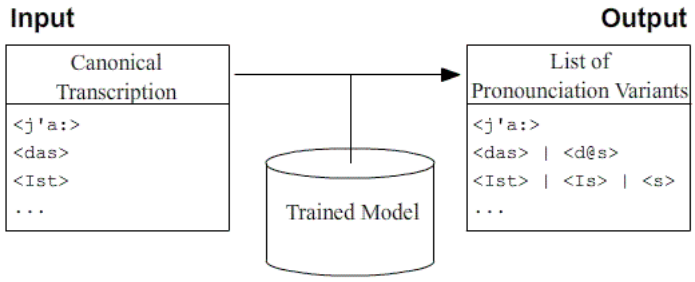

Abbildung 1.2: Architektur des Spontaneous Speech Tools

1.2 Gliederung der Arbeit

Die Arbeit in ist vier Hauptteile gegliedert: Grundlagen, Ausarbeitung des Spontaneous Speech Tools und Evaluierung.

Grundlagen. Bevor man sich in die technische Umsetzung des Spontaneous -Speech-Tool vertieft, werden die theoretischen Grundlagen zum Thema erörtert. Als erstes werden automatische Segmentierungssysteme, und, in ihrem Zusammenhang der IMS Aligner, im Detail beschrieben. Anschließend wird ein erster Einblick in das Kiel Korpus, bzw. das Kiel Korpus der Spontansprache, als verwendeter Trainings- und Testdatensatz gegeben. Zuletzt wird eine Zusammenfassung der bereits erforschten und entwickelten Methoden zur Generierung von Aussprachevarianten gegeben.

Aufbereitung der Daten. Als erstes wird das Kiel Korpus der Spontansprache ausführlich untersucht. Anhand der so gewonnenen Erkenntnisse wird ein Datensatz zum Trainieren eines statistischen Modells herausgearbeitet, und anschließend wird über die Vollständigkeit und Tauglichkeit des so konzipierten Datensatzes diskutiert.

Ausarbeitung des Spontaneous Speech Tools. In diesem Abschnitt wird die Entwicklung des Tools eingehend spezifiziert. Zu Beginn wird erklärt, was ein trainiertes Modell ist, und ein Überblick des dazu verwendeten Hilfswerkzeuges WEKA gegeben. Als nächstes werden die Methoden und Experimente zur Ermittlung des besten lern Algorithmus präsentiert, sowie über die Entscheidung für einen der Algorithmen und Methoden diskutiert. Im Anschluss wird die eigentliche Implementierung des Tools und die Anbindung an den IMS Aligner beschrieben.

Evaluierung. Hier werden die Resultate des Aligners ohne das Tool und mit Einbindung des Spontaneous Speech Tools diskutiert. Anfangs wird der Hilfsalgorithmus zur Evaluierung vorgestellt, und daraufhin die erlangten Ergebnisse diskutiert.

2 Grundlagen

2.1 Automatische Segmentierung und Annotation von Sprachsignalen

Wie in der Einleitung bereits erwähnt, dient ein Annotations- und Segmentierungssystem dazu, die Segmente eines Sprachsignals zu annotieren, also den Segmenten bestimmte Kennzeichnungen zuzuordnen. Das verwendete Annotationsinventar kann sehr unterschiedliche Kennzeichnungen, sog. *Labels*, beinhalten: Beispielsweise phonetische, prosodische oder morphologische Ereignisse. Das Ergebnis der Segmentierung ist eine Datei mit einer Liste mit den Elementen des Annotationsinventars und den jeweiligen (Anfangs- oder End-) Zeitpunkten der Segmente, welche auch Labelfile genannt wird.

Man kann das Sprachsignal von Hand oder automatisch annotieren. Die manuelle Annotation hat den Vorteil, dass man genauer arbeiten und sprecherspezifische Ungenauigkeiten berücksichtigen kann. Manuelles Annotieren ist jedoch sehr ermüdend und langsam. Ein unerfahrener Labeler kann für eine Minute Sprachaufnahme bis zu einer Stunde an Arbeitszeit aufbringen. Ausserdem muss mit Inkonsistenzen in den Annotationen gerechnet werden, wenn mehrere Personen am Labeln beteiligt sind. Außerdem ist durch den hohen Zeitaufwand das Labeln von großen Datenbanken unvermeidbar mit hohen Kosten verbunden.

Die Automatisierung des Prozesses der Segmentierung bietet da eine praktische Alternative. Ein Sprachsignal und (je nach System) der Inhalt der Äußerung, werden an ein Segmentierungssystem gegeben, welches das resultierende Labelfile in kurzer Zeit fertig stellt.

Andererseits sind auch diese Systeme nicht einwandfrei. Ein automatisches Segmentierungs- und Annotationssystem kann eine Fehlerrate von 5% - 25% haben, abhängig von Sprache und Annotationsschema [Carstensen et al., 2010, Seite 493-494]. Dennoch ist der Einsatz von automatischen Annotationssystemen nicht aus der Sprachtechnologie weg zu denken. Die Sprachdatenbanken wachsen immer weiter an, und manuelles Segmentieren kann

mit diesen Mengen niemals zurecht kommen. Dies bedeutet auch, dass die Weiterentwicklung und Verbesserung der automatischen Segmentierungssysteme für die moderne Sprachtechnologie von großer Bedeutung ist.

Die meisten automatischen Segmentierungssysteme verwenden als Eingabe nicht nur das Sprachsignal, sondern auch eine orthografische Transkription mit den vom Sprecher geäußerten Sätzen [Demuynck et al., 2002]. Diese Segmentierung besteht im Grunde, aus zwei voneinander unabhängigen Vorgängen.

In einem der Vorgänge wird die phonetische Transkription aus der Textdatei gewonnen, indem aus der orthografischen Transkription ein Netzwerk mit allen möglichen phonetischen Realisationen gebildet wird. Im zweiten Vorgang werden aus dem Audiosignal Merkmalsvektoren extrahiert. Durch eine Viterbi-Suche wird anschließend der günstigste Weg durch das Phonemnetz gesucht. Zum Schluss wird die endgültige Zeit-Ereignis-Liste ausgegeben.

2.2 IMS Aligner

Der IMS Aligner [Rapp, 1995] ist ein auf Hidden Markov Modellen (HMMs) basiertes Segmentierungs- und Annotationssystem. Das System verwendet das Entropic Hidden Markov Toolkit [HTK, 1994] und das CELEX Lexikon der deutschen Sprache mit kanonischen Lautformen [Baayen et al., 1995]. Als Eingabe dienen das Sprachsignal und die orthographische Transkription der Äußerung. Die Ausgabe sind drei Labelfiles, welche die Annotation auf Phonem- , Silben- oder Wortebene enthält. Im folgenden wird der Prozess der Gewinnung eines Labelfiles auf Phonembasis nach [Rapp, 1995] beschrieben.

2.2.1 Architektur des IMS Aligners

Die Architektur des Tools wird in Abbildung 2.1 dargestellt. Die Eingaben, das Sprachsignal sowie die orthographische Transkription, werden unabhängig von einander bearbeitet.

Die Verarbeitung der orthografischen Transkription beginnt mit einer Graphem-zu-Phonem-Konvertierung (im Bild: linke Spalte, oben). Die im Text enthaltenen Wörter werden im CELEX Lexikon nachgeschlagen, und die gefundene phonetische Transkription als Liste aneinander gehängt. Wird eine Vokabel nicht im Lexikon gefunden (Namen, Fremdwörter, usw.), können diese Wörter durch vorhandene Regeln konvertiert werden.

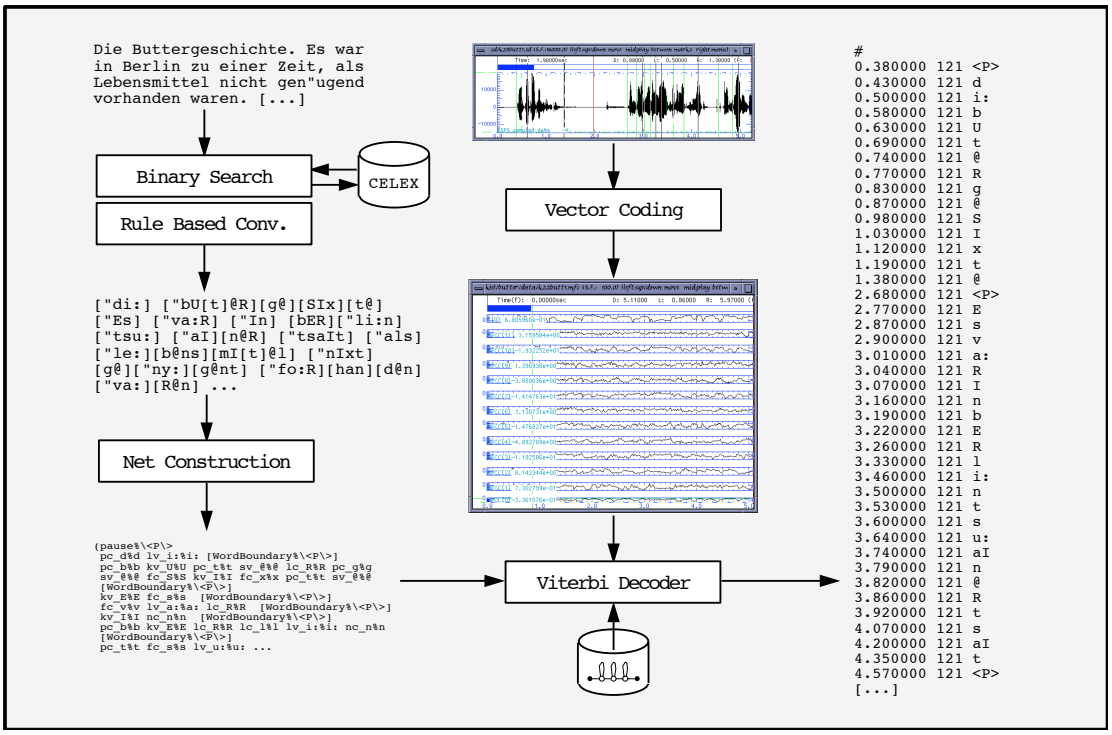

Abbildung 2.1: aus [Rapp, 1998]. **Linke Spalte:** Vorverarbeitung des Textes, **mittlere Spalte:** Vorverarbeitung des Sprachsignals, **unten:** trainierte Phonemmodelle, **rechte Spalte:** Ausgabe Labelfile

Eine weitere Komponente des Aligners sind die trainierten Phonemmodelle (im Bild als Zylinder dargestellt). Mittels der gewonnen Liste aus der Textvorverarbeitung, werden die Phonemmodelle als Netzwerk zusammengefügt (im Bild: linke Spalte, unten).

Separat von diesem Prozess wird das Audiosignal in Vektoren umgewandelt (im Bild: mittlere Spalte, oben). Das System bestimmt zunächst 12 mel-Frequenz Koeffizienten und die Gesamtenergie (also 13 Koeffizienten), aus welchen die Delta[1]- und die Delta-Delta-Werte berechnet werden. Dies ergibt am Ende 39 Parameter pro Vektor.

Die so gewonnenen Vektoren und das Phonemmodellnetz werden an den Viterbi-Decoder weiter gegeben (im Bild: mittlere Spalte, unten). Hier wird durch die Viterbi-Suche der optimalste Weg durch das Netz bestimmt. Als Resultat wird ein Labelfile mit den Phonemen und den jeweiligen Endzeitpunkten ausgegeben (im Bild: rechte Spalte).

Für das Training der Phonemmodelle wurde in der erste Trainingsphase das Kiel Korpus der gelesenen Sprache verwendet (vgl. Abschnitt 2.3), aber auch die Einträge des CELEX Lexikons.

[1]Delta: mathematischer Differenzoperator

2.2.2 Grenzen des Aligners

Der IMS Aligner hat zum Ziel, die Zeitpunkte der sprachlichen Einheiten, sprich Phoneme, Silben oder Wörter, so genau wie nur möglich zu bestimmen. In diesem Sinne können die Ergebnisse des Aligners als zufrieden stellend angesehen werden. Genauer gesagt, kann ein solches Segment mit einer 76.82%-igen Wahrscheinlichkeit, auf bis zu 20 ms ungenau sein, oder wie es [Rapp, 1995] ausgedrückt hat:

> „[...] about one fourth of the automatically found segment boudaries mismatch the manual labelling by 20ms or more [...]“

Da der IMS Aligner anfangs nur für gelesene Sprache eingesetzt werden sollte, hat sich auch der Zugriff auf das CELEX Lexikon als ausreichende Datenquelle erwiesen. Es bietet eine gute Abdeckung linguistischer Formen, und außerdem bietet es die Möglichkeit, Annotationen nicht nur auf Wort- und Phonemebene zu erstellen, sondern auch auf Silbenebene, welche morphologische und morphosyntaktische Markierungen enthält.

Andererseits ist das Zurückgreifen auf ein phonologisch orientiertes Lexikon ungemein einschränkend. In CELEX sind nur die kanonischen Ausspracheformen zu finden, also keine Varianten der Aussprache, welche nicht nur in unkontrolliertem spontanen Redefluss produziert werden. Eine der häufigsten phonologischen Änderungen im Redefluss jeder Art, ist die Tilgung oder Änderung von Lauten in unbetonten Endsilben (Bsp. „haben“ wird von /ha:b@n/ oft zu /ha:bm/, siehe Kapitel 2.4.1). Solche Modifikationen beeinflussen selbstverständlich das Endergebnis des Segmentierens, da Segmente nicht gefunden werden (im Beispiel /@/), oder durch andere ersetzt wurden (im Beispiel /m/). Aussprachen, welche von der kanonischen Form deutlich abweichen, können im Sprachsignal teilweise nicht richtig vom IMS Aligner lokalisiert werden.

Des Problems war sich der Konstrukteur des Aligners, Stefan Rapp, in der technischen Umsetzung bewusst. Daher wurde die Möglichkeit offen gelassen, eine einfache Lösung für das Aussprachevariantenproblem zu gestallten: das Lexikon kann ersetzt oder um Varianten erweitert werden.

Wie in der Einleitung zuvor erklärt, greift das Spontaneous Speech Tool genau an diesem Punkt ein. Von den beiden Ergänzungsmöglichkeiten des Aligners wird die zweite, Erweiterung des Lexikons um Varianten, bevorzugt. Eine Ersetzung des umfangreichen CELEX Lexikons ist nicht notwendig, da Aussprachevarianten stark vom Kontext abhängen. Beispielsweise ist es nicht notwendig, die Variante /UN/ von „und“ im Kontext „...und dann...“ zu generieren, da die Modifikation von /Unt/ zu /UN/ einen velaren Folgelaut voraussetzt. Um dem IMS Aligner nur die relevanten Varianten von Aussprachen

zur Verfügung zu stellen, werden diese aus der Liste von phonetischen Transkriptionen generiert, welche nach dem CELEX-Lexikon-Lookup ausgegeben wird.

2.3 Das Kiel Korpus

Als Teilnehmer des PHONDAT-Verbundsprojektes[2] übernahm das Institut für Phonetik und Digitale Sprachverarbeitung (IPDS) der Universität Kiel die Erarbeitung einer sehr großen gesprochenen Datenbank. Ziel dieses Projektes war „[...] die Schaffung einer repräsentativen Materialgrundlage auf Diphonem- sowie Silbenbasis in gelesener Sprache zum Trainieren der Spracherkennungssysteme [...]“ [Kohler, 1992].

Im Rahmen dieses Projektes wurde nicht nur eine Datenbank gelesener, sondern auch spontan gesprochener Sprache zusammengetragen, was zur Unterscheidung zwischen dem Kiel Korpus der gelesenen und dem Kiel Korpus der spontanen Sprache führt. Alle Sprachaufnahmen sind manuell annotiert worden. Das verwendete Annotationssystem wurde speziell für das PHONDAT-Projekt entwickelt und ist unter dem Terminus „Kieler Konventionen“ bekannt. Die leicht abgewandelte SAMPA-Notation (Speech Assessment Methods Phonetic Alphabet [URL: SAMPA]), beinhaltet 22 Konsonanten, 20 verschiedene Vokalsymbole und 4 Nasalvokale. Obwohl hauptsächlich auf phonetischer Ebene annotiert, gibt es auch Markierungen für Informationen zu Satzakzenten, Wortgrenzen (auch in Komposita) und der allgemeinen Klasse der Funktionswörter.

Da für die Ausarbeitung des Spontaneous Speech Tools Daten aus dem Kiel Korpus der Spontansprache verwendet wurden, wird nun detailliert auf dieses Korpus, besonders auf die Labelfiles des Korpus, eingegangen.

2.3.1 Das Kiel Korpus der Spontansprache

Das Kiel Korpus der Spontansprache [Kohler, 1995] umfasst drei CDs, auf welchen insgesamt 117 Dialoge von 52 Sprechern aufgenommen wurden, welche gesamt 1.984 Audiosignale ausmachen. Bei den Dialogen handelt es sich um Terminvereinbarungsaufgaben. Die Sprechpartner befanden sich in unterschiedlichen Zimmern, hatten vom Aufnahmeteam vorbereitete Terminkalender vorliegen, und versuchten via Headset einen Termin zu vereinbaren. Beide Sprechpartner wurden auf zwei Kanälen (links und rechts) mit 16-bit und 16kHz aufgenommen.

[2] Das PHONDAT-Verbundsprojekt ist eine Zusammenfassung von Projekten der Universitäten Braunschweig, Kiel und LMU München mit den Unternehmen Siemens, Philips, Daimler und SEL

Als nächstes wurden die Aufnahmen transliteriert und die zugehörigen Labelfiles gemäß den Kieler Konventionen erstellt. Da es sich hier um Spontansprache handelt, wurden zu den 46 Lautmarkierungen auch Markierungen zu nichtsprachlichen Ereignissen, wie Husten, Lachen oder Zögern, verwendet.

In den Tabellen 2.1 und 2.2 sind die verwendeten SAMPA-Markierungen der Kieler Konventionen aufgelistet. Der einzige Unterschied zwischen den üblichen SAMPA-Labels und den Kiel-SAMPA-Lables, ist die Markierung des glottalen Verschlusslautes. Die Tabellen 2.3.1 und 2.5 geben eine Erklärung der nichtsprachlichen Ereignisse. Die Labels in Tabelle 2.4 stehen für phonologische Modifikationen, welche der Sprecher im Redefluss erzeugt. Eine weitere Besonderheit der Kieler Konventionen ist die zusätzliche Markierung jedes Labels mit einem Etikett für wortinterne oder -externe Position, Tabelle 2.6.

SAMPA	IPA	Beispiel
a:, a	aː, a	Kahn, Kamm
e:, E	eː, ɛ	Beet, Bett
i:, I	iː, ɪ	riet, ritt
o:, O	oː, ɔ	bog, Bock
u:, U	uː, ʊ	Buße, Busse
y:, Y	yː, ʏ	Hüte, Hütte
E:	ɛː	Käse
2:, 9	øː, œ	Höhle, Hölle
a~	ã	Restaurant
E~	ɛ̃	Teint
O~	ɔ̃	Saison
9~	œ̃	Parfum
aI	aɪ	zwei
aU	aʊ	Bauch
OY	ɔʏ	neun
@	ə	lesen
6	ɐ	Leser, her
’	Betonung	’a:
’’	Zweitbetonung	’’a

Tabelle 2.1: [Kohler, 1995] Lautinventar: 24 Vokale und ihre Betonungen

SAMPA	IPA	Beispiel
b	b	Bein
d	d	Dusche
f	f	frei
g	g	Gast
h	h	Hast
j	j	ja
k	k	Kahn
l	l	Licht
m	m	Mann
n	n	neun
p	p	Platz
r	r	rauch
s	s	las
t	t	Torte
v	v	Vase
z	z	lesen
C	ç	Sicht
x	x	Dach
S	ʃ	Stadt
Z	ʒ	Loge
N	ŋ	Junge
Q	ʔ	acht

Tabelle 2.2: [Kohler, 1995] Lautinventar: 22 Konsonanten

morphologische und syntaktische Marker	
#	Morphemgrenzen bei Komposita
+	Kennzeichnet das Ende eines Funktionswortes
#c:	Satzanfang

Tabelle 2.3: [Kohler, 1995] Markierungen der groben morphologischen Analyse

Modifikationen im Redefluss	
$-h	Aspiration eines Plosivs
$-q	Knarren
$-~	Nasalierung
$-MA	Ansatz eines eigentlich gelöschten Lautes
$-t	Einfügen von Laut (hier /t/)
$n-m	Substitution (hier /n/ wird durch /m/ ersetzt)
$@-	Löschen von Laut (hier wurde /@/ gelöscht)

Tabelle 2.4: [Kohler, 1995] phonologische Kennzeichnungen

nichtsprachliche Ettiketierungen	
/+ und =/+	falscher Start beim Sprechen
/- und =/-	Abbrechen des Wortes
_	wortinterne Störungen
z:	zögerungsbedingte Längung des Wortes
v:	vor Zögerungen
n:	Neologismen
p:	Pause
h:	Atemgeräusche
l:	Lachen
q:	Husten
r:	Räuspern
s:	Schmatzen
w:	Schlucken
g:	Klicken, Kilngeln, Klpofen, ...
:k	technische Störung
$%	Unsicherheit des Labelrs

Tabelle 2.5: [Kohler, 1995] Markierungen non-verbaler Ereignisse

Für den in dieser Arbeit verwendeten Datensatz wurden allerdings nicht alle Labels der Kieler Konventionen übernommen. Die Vokale und Konsonanten aus den Tabelle 2.1 und 2.2 werden unverändert verwendet. Aus Tabelle 2.3.1 bleiben nur Morphemgrenzen „#“ erhalten. Die Kennzeichnungen der Tabelle 2.5 werden nicht für weitere Untersuchungen gebraucht. Die Modifikationen aus Tabelle 2.4 haben sich als sehr nützlich erwiesen (siehe Kapitel 3), die Kennzeichnung der Aspiration oder die Unsicherheit des Erstellers der Labelfiles wurden nicht beachtet. Auch die zusätzlichen Labels aus Tabelle 2.6 wurden nicht in den verwendeten Datensatz aufgenommen, jedoch wurde die Etikette $ für

Wortgrenzen (und leere Wörter, siehe Tabelle 3.3 Abschnitt 3.1) eingeführt.

Im anschließenden Beispiel 1 ist ein Ausschnitt aus dem Kieler Labelfile `g071a000.s1h` gegeben. Es veranschaulicht die Verwendung der Kennzeichnungen aus jeder der obigen Tabellen.

`#`	non-verbale Markierung (Achtung: doppelte Belegung von `#`)
`##`	Wortanfang
`$`	Wortinternes Label.

Tabelle 2.6: Zusätzliche (Kiel-spezifische) Markierungen

BEISPIEL 1: „Ja, guten Tag.“

Zeit	Label	Erklärung
5935	`#c:`	Satzanfang, wortextern
5935	`#-s:`	Schmatzen, wortextern
6265	`#-h:`	Atmung, wortextern
12695	`##j`	Laut am Wortanfang
1416	`$'a:`	Laut, wortintern
15406	`#,`	Komma, wortextern
15406	`##g`	Laut am Wortanfang
16222	`$'u:`	Laut, wortintern
16728	`$t-n`	`/t/` wird durch `/n/` substituiert, wortintern
18048	`$@-`	`/@/` wird getilgt, wortintern
18048	`$n-`	`/n/` wird getilgt, wortintern
18048	`##t`	Laut am Wortanfang
18282	`$-h`	Aspiration des Plosivs, wortintern
18682	`$'a:-'a`	`/a:/` wird durch `/a/` substituiert, wortintern
19646	`$k-x`	`/k/` wird durch `/x/` substituiert, wortintern
20264	`#.`	Punkt, wortextern

2.4 Methoden zur Ermittlung von Aussprachevarianten

Bevor ich auf die die Diskussion zu der in dieser Arbeit vorgestellten Methode zur Generierung von Aussprachevarianten eingehe, werden zunächst andere Arbeiten zu diesem Thema vorgestellt.

Im Allgemeinen können die meisten Methoden in zwei Grundarten geteilt werden: Generierung von Aussprachevarianten durch Anwendung phonologischer Regeln, und Ermittlung der Varianten durch statistische Auswertung von Korpora. Die im Folgenden präsentierten Ansätze sind gute Repräsentanten dieser Grundansätze.

Zu diesen Methoden wird später Bezug genommen, da der in dieser Arbeit präsentierte Ansatz sich von den unten beschrieben unterscheidet.

2.4.1 Phonologische Prozesse als Regeln für Aussprachevarianten

Bereits in der Einleitung wurde auf die Unterschiede zwischen gelesener und im freien Redefluss erzeugter Sprache hingewiesen. Betrachtet man die gelesene Sprache als Ausgangspunkt (deshalb auch die Bezeichnung *kanonische* Lautform), so können Varianten im Redefluss als Abwandlung der „Grundsprache“ gesehen werden. In kontinuierlicher Spontansprache kommt es zu Änderungen im Redefluss: Lautkontexte beeinflussen sich, es kommt zum Ersetzen, Löschen oder gar Einfügen von Lauten.

Solche Modifikationen der kanonischen Lautformen zu Aussprachevarianten unterliegen (meist) einer Regelmäßigkeit. Daher werden die Modifikationen auch phonologische Prozesse oder Regeln genannt. Sind die phonologischen Regeln einer Sprachen bekannt, so können aus den kanonischen Lautformen die Varianten abgeleitet werden. Solche Regeln können manuell oder automatisch erstellt werden. Bei der (halb-) automatischen Ermittlung solcher Regeln aus einem Korpus wird das Korpus statistisch untersucht und die zahlreichsten Abweichungen von der kanonischen Aussprache als Regeln herausgearbeitet.

Genau diese Methode ist die Grundlage der regelbasierten Ermittlung von Aussprachevarianten wie sie in [Kemp, 1996] beschrieben wird. Die verwendeten Regeln sind im Anschluss aufgelistet (jedoch ohne die erste und häufigste Regel: keine Änderung). Sie wurden durch zwei unabhängige halbautomatische Korpusanalysen ermittelt. Die Regeln werden auf die kanonische Lautform angewendet, und als Resultat werden Varianten generiert, welche an den Spracherkenner weitergegeben werden.

Im Aufsatz von [Kemp, 1996] wird auch über die Anzahl der Anwendungen der Regeln auf ein Wort diskutiert. Da sich phonologische Prozesse gegenseitig beeinflussen, wie es aus den Regeln unten ersichtlich wird, ist es nicht ausreichend, eine Regel pro Wort anzuwenden. Im nächsten Schritt müsste die nächste Regel auf zwei Wörter angewandt werden, im dritten Schritt auf drei Wörter, usw. Eine solche Vorgehensweise führt zur Übergenerierung. Um eine Übergenerierung zu vermeiden, wurden „[...] für ein gegebenes Wort alle anwendbaren Regeln an allen anwendbaren Stellen angewendet. Die so entstandenen Varianten wurden jedoch [...] nicht weiter modifiziert [...]“ [Kemp, 1996].

Das Ergebnis der Anwendung der so entstandenen Varianten im Spracherkenner ist eine Verbesserung der Fehlerrate um 1.5% [Kemp, 1996].

Liste der phonologischen Regeln (nach [Kemp, 1996]):

1. **Tilgung des glottalen Verschlusses**
   Beispiel: „*könnte ich*“
   `k'9nt@ QIC` → `k'9nd_ _'IC`
   Erklärung: als Folge der Tilgung des Glottalverschluss `/Q/` vor `/I/`, wird der Laut `/t/` in seiner Stimmhaftigkeit geändert
2. **Tilgung des Schwa-Lautes**
   Beispiel: „*wollen*“
   `v'Ol@n` → `v'Ol_n`
   Erklärung: Tilgung von `/@/` in der letzten Silbe `/l@n/`
3. **Nasalassimilation nach Schwa-Elision**
   Beispiel: „*Abend*“
   `Q'ab@nt` → `Q'ab_mt`
   Erklärung: Nach Tilgung von `/@/` in der letzten Silbe, kommt es zur Substitution von `/n/` durch `/m/`
4. **Reduktion von `/r/` bei Vokal-`/r/`-Verbindungen**
   Beispiel: „*wäre*“
   `vE:r@` → `vE:6_`
   Erklärung: Substitution von `/r/` durch `/6/`, danach Tilgung von `/@/` am Wortende
5. **Änderungen der Vokaldauer**
   Beispiel: „*Montag*“
   `m'o:nta:k` → `m'o:ntak`
   Erklärung: Die Ersetzung des `/a:/` durch `/a/` könnte hier eine Folge schnellen Redens sein
6. **Änderungen der Vokalqualität**

Beispiel: „*es*“
`QEs` → `_@s`
Erklärung: Der vordere halboffene Vokal `/E/` wurde durch `/@/` ersetzt

7. Stimmhaft-stimmlos Änderungen

Beispiel: „*guten*“
`g'u:t@n` → `g'u:d_n`
Erklärung: siehe Erklärung unter 1.

8. Lautverschmelzung bei gleichem Silbenaus- und Anlaut

Beispiel: „*hat der*“
`hat d@6` → `ha_ d@6`
Erklärung: Das `/t/` am Wortende und das `d` am Wortanfang verschmelzen zu einem stimmhaften Laut (wegen der Vokalumgebung)

9.Monophtoniering von Diphtongen

Beispiel: „*auch*“
`QaUx` → `Qo:x`
Erklärung: Diese Modifikation ist rein dialektal, d.h. der Sprecher produziert solche Variationen bewusst, und nicht als Folge von Ungenauigkeit

10. Glottalisierung eines Plosivs in einem Kontext von Nasalkonsonanten

Beispiel: „*kuck mal*“
`k'Uk$ma:l` → `k'UQ#ma:l`
Erklärung: Der wortfinale Plosiv `/k/` wird durch den glottalen Verschlusslaut ersetzt, und die Worte verschmelzen zu einem

11. Nasalierung der Endsilbe „-nden“

Beispiel: „*finden*“
`f'Ind@n` → `f'In__n`
Erklärung: Die komplette letzte Silbe wird auf ein `/n/` reduziert

2.4.2 Ermittlung von Aussprachevarianten durch Korpusanalyse

Auch diese Verfahren basieren auf einem phonetisch annotiertem Korpus. Aus diesen Daten werden durch statistisches Auswerten, und je nach Verfahren anschließendes Aussortieren, die phonetischen Varianten herausgefiltert und dem Lexikon hinzugefügt, ohne dabei phonologische Regelmäßigkeiten zu beachten.

Eine mögliche Umsetzung dieser Methode wird in [Sloboda et al., 1996] präsentiert. Dieser Ansatz soll die Erkennungsrate des Spracherkenners JANUS[3] erhöhen.

[3]JANUS: Spracherkenner für mehrere Sprachen, auch im VERBMOBIL-Projekt verwendet

Bevor der eigentliche Algorithmus wie in [Sloboda et al., 1996] angewandt werden kann, muss einiges an Vorarbeit geleistet werden. Als erstes werden mittels des Spracherkenners Labelfiles auf Wortebene erstellt. Anschließend wird eine Konfusionsmatrix[4] der Phoneme, und die jeweiligen Phonemmodelle erstellt. Nach der Analyse häufigster Fehlerkennungen wird eine Liste generiert, mit Wörtern, welche im Wörterbuch um Varianten ergänzt werden sollen.

Der eigentliche Algorithmus beginnt mit der Extrahierung aller Vorkommen der Wörter der oben genannten Wortliste aus dem Korpus. Mit Hilfe eines Segmentierungssystems werden die jeweiligen phonetischen Transkriptionen generiert, und diese statistisch ausgewertet. Die statistisch irrelevanten Varianten eines Wortes sowie Homophone[5] werden aussortiert. Anhand der generierten Konfusionsmatrix werden auch diejenigen Varianten ausgeschlossen, welche sich nur in stark verwechselbaren Lauten unterscheiden (z.B. die Variante `/dam/` für das Wort „dann“ wegen der hohen Verwechslungsrate von `/n/` und `/m/`). Das Lexikon wird um die übrigen Varianten erweitert. Der Spracherkenner kann nun neu trainiert und getestet werden. Diese Methode verringert die Fehlerrate des Erkenners um bis zu 6.3% [Sloboda et al., 1996].

2.5 Diskussion der vorgestellten Ansätze und Präsentation eines neuen Ansatzes

2.5.1 Diskussion des phonologischen Ansatzes nach [Kemp, 1996]

Vorteil dieses Verfahrens ist, dass die Regeln auf der Phonologie der Äußerungen, und nicht auf der der Wörter basieren. Das bedeutet, dass auch wortübergreifende spontansprachliche Änderungen berücksichtigt werden. So kann die Aussprache von „und“ im Kontext „und dann“ zu /`Un tan`/ mutieren, während im Kontext „und ganz“ die Aussprache /`UN gants`/ bevorzugt wird.

Außerdem können bei diesem Ansatz auch statistisch seltenere Regeln herausgearbeitet werden, was letztendlich eine gute Abdeckung aller Änderungen im Redefluss garantiert.

Nachteilig ist dagegen, dass es sehr mühselig ist, gute Regeln aufzustellen. möchte man eine gute Abdeckung der Änderungen, so hat man schnell über tausend Regeln, welche einzeln auf ihre Effizienz geprüft werden müssen, wie beispielsweise in [Kipp, 1997] be-

[4] Konfusionsmatrix: tabellarische Darstellung der Häufigkeiten des Auftretens für alle möglichen Kombinationen ermittelter Phoneme und tatsächlicher Phoneme

[5] Homophon: gleichklingendes Wort unterschiedlicher Schreibung, z.B. *„Küste“* und *„(er) küsste“*

schrieben. Verzichtet man auf eine hohe Abdeckungsrate, so muss die Vielzahl der Regeln immer noch auf einen minimalen Satz von Konventionen herunter gearbeitet werden, was auch hier mit einem hohen Zeitaufwand einhergeht.

Zudem bleibt bei einem regelbasiertem Ansatz immer das Problem der potenziellen Übergenerierung bestehen, völlig unabhängig von der Anzahl der Regeln.

2.5.2 Diskussion des statistischen Ansatzes nach [Sloboda et al., 1996]

Da bei diesem Verfahren das Lexikon des Segmentierungssystems nur um statistisch relevante Varianten einer Aussprache erweitert wird, hat das System weit weniger unpassende Varianten zu verarbeiten.

Außerdem basiert dieser Ansatz ausdrücklich auf der Statistik der Daten. Somit kann nach der Erstumsetzung des Algorithmus das Verfahren mit weit geringerem Zeitaufwand auf weitere Datensätze angewandt werden.

Unvorteilhaft allerdings ist, dass für eine repräsentative statistische Auswertung das veränderte Wort oft genug im Korpus enthalten sein muss. Ist dies nicht der Fall, so wird die Aussprachevariante aussortiert und damit nicht ins Lexikon aufgenommen. Da jedoch viele phonologische Prozesse sehr selten vorkommen (siehe [Baayen, 2001]), kann hier nicht von einer großflächigen Abdeckung der spontansprachlichen Änderungen die Rede sein.

2.5.3 Statistischer Ansatz auf phonologischer Ebene

Alle statistisch basierten Ansätze haben den Vorteil des höheren Automatisierungsgrades. Wurde ein statistischer Algorithmus einmal implementiert, so kann er mindestens halbautomatisch auf weitere Datensätze angewandt werden. Unter Berücksichtigung des Themas dieser Arbeit bedeutet dies, dass je nach Wahl des Datensatzes Aussprachevarianten entweder gezielt regional, oder gar für weitere Sprachen generiert werden können.

Dieser deutliche Vorteil dem regelbasierten Ansatz gegenüber hat mich davon überzeugt, eine statistische Vorhergehensweise zu entwickeln. Was ich jedoch an der statistischen Methode von [Sloboda et al., 1996] bemängele, ist die Änderungsanalyse auf Wortebene. In einer spontansprachlichen Äußerung verschmelzen Wörter regelmäßig zu untrennbaren Einheiten.

Beispielsweise wird die Wortfolge „können wir“ bei schnellem Reden, gern zu /k'9m@/ verkürzt. Mann könnte hier zwischen den Wörtern /k'9/ und /m@/ unterscheiden, jedoch wird „können“ in kaum einem anderen Kontext dermaßen verkürzt, und die Änderung von „wir“ zu /m@/ kann nur nach unbetonten Endsilben /t@n/, /b@n/, usw. beobachtet werden. Eindeutig handelt es sich bei diesem Phänomen um einen wortübergreifenden Prozess. Das statistische Verfahren nach [Sloboda et al., 1996] wird wohl kaum die Änderung der einzelnen Wörter „können“ und „wir“ zu ihren Varianten /k'9/ und /m@/ als statistisch relevant anerkennen. Hier sind die phonologischen Regeln tatsächlich von Vorteil. Da der Prozess der Herausarbeitung von Regeln eine phonologisch Datenanalyse voraussetzt, können auch solche Besonderheiten durch die Analyse aufgedeckt werden.

Da dabei die Regelaufstellung das Potenzial des Ansatzes stark einschränkt, habe ich einen Ansatz entwickelt, der die Vorteile der Statistik, sowie auch die der phonologischen Regeln, kombiniert. Die Grundidee des *statistischen Ansatzes auf phonologischer Ebene* ist, die Anwendung von Regeln durch die Vorhersage eines statistischen, trainierten Modells zu simulieren.

Grundlegend für Ansätze dieser Art ist ein gut aufbereiteter Datensatz, auf welchem ein statistisches Modell trainiert wird. Durch das Training erkennt das Modell, dass sich Einheiten in bestimmten Kontexten zu anderen Einheiten abändern. Das trainierte Modell kann anschließend auf neue Einheiten angewandt werden und als Resultat eine Variante zu der eingegebenen Einheit generieren.

Damit ein maschinelles Verfahren Änderungen erkennen kann und diese zu einem Modell entwickelt, benötigt es zweierlei Daten: die unveränderten, grundliegenden Daten, sowie veränderte, variierte Daten. Beide Klassen von Daten werden Einheit für Einheit in einem neuen Datensatz nebeneinander platziert. So kann ein statistisches Verfahren die häufigsten Änderungen der Einheiten deutlich erkennen. Soll das Modell Änderungen in Abhängigkeit von Bedingungen erlernen, so müssen weitere Informationen zu jeder Einheit zur Verfügung gestellt werden.

Angewandt auf die Terminologie des Themas Generierung von Aussprachevarianten, ist der Ansatz wie folgt umzusetzen:

> **Erstens:** Es wird ein Korpus benötigt, welches zu der spontansprachlichen Aussprache auch die kanonische beinhaltet. Alternativ kann die Information über die kanonische Aussprache aus einem Lexikon herangezogen werden, dafür wird allerdings die Transliteration[6] der Äußerungen benötigt.

[6]Transliteration bezeichnet hier die Übertragung von Wörtern aus der gesprochenen Sprache in die geschriebene

Zweitens: Aus den kanonischen und variierten Transkriptionen muss ein Datensatz für die Zwecke des maschinellen Lernens aufgebaut werden. Dazu werden nicht nur die Transkriptionen nebeneinander aufgelistet, sondern auch weitere Informationen, die zur Erkennung von Regelmäßigkeiten beitragen. In dieser Arbeit wird ein lautlicher Kontext (zwei Einheiten vor und nach jeder Einheit) vorgeschlagen, sowie die Betonung und phonetische Eigenschaften der Einheiten.
Drittens: Ein Modell wird auf dem fertigen Datensatz trainiert. Es wird empfohlen, mehrere maschinelle Lernverfahren anzuwenden, um dasjenige mit der höchsten Genauigkeitsrate zu ermitteln. Dafür sollte jedes Modell nach dem Training getestet werden.
Viertens: Das gewählte Modell wird in ein „*Tool*“, bzw. Hilfsprogramm, eingearbeitet, welches wiederum an ein Segmentierungssystem angeschlossen werden kann. Das Tool bekommt als Eingabe die vom Segmentierungssystem vorgeschlagene Aussprache der Äußerung, generiert alternative Aussprachen, und gibt diese an das System weiter.

Die Vorteile dieses Ansatzes sind ziemlich nahe liegend: Das Verfahren ist statistisch und kann ohne größere Schwierigkeiten auf verschiedene Datensätze wie oben beschrieben angewandt werden, und es berücksichtigt bei der Generierung von Varianten allerlei zusätzliche Informationen.

Bei der Beschreibung dieses Ansatzes habe ich beabsichtigt den Begriff „Einheit“ verwendet. Je nach Beschaffenheit des Korpus kann anstelle von Phonemen auf Silben- oder Morphembasis gearbeitet werden. Aufgrund der Struktur des Kieler Korpus der Spontansprache, habe ich mich für einen phonembasierten Datensatz entschieden.

3 Aufbereitung der Daten

Bevor ein statistisches Lernmodell zur Vorhersage von Änderungen im Redefluss trainiert werden kann, muss ein Datensatz mit durchdachtem Inhalt aufgestellt werden.

Wie in der Einleitung bereits angesprochen, wird das Kiel Korpus der Spontansprache als Grunddatensatz, sowie zu Testzwecken verwendet. Auf 2/3 der Daten wird trainiert, auf dem restlichen Drittel getestet. Da der IMS Aligner auf dem Kiel Korpus der gelesenen Sprache trainiert und getestet wurde, ist das Kiel Korpus der Spontansprache eine sinnvolle Datenbasis zur Weiterentwicklung des Aligners. Da beide Korpora im Rahmen des selben Projektes entstanden, gibt die Ähnlichkeit der Annotationen der Korpora einen klaren Anhaltspunkt zum Unterschied zwischen den Ergebnissen des Aligners, angewandt auf gelesener und spontaner Sprache.

In Abschnitt 2.3 wurde der grobe Aufbau des Kiel Korpus der Spontansprache sowie eine Erläuterung des verwendeten Etiketteninventars gegeben. Ergänzend dazu werden nun die Daten und ihr Aufbau im einzelnen diskutiert.

Der Inhalt jeder der drei CDs des spontansprachlichen Korpus kann in drei Typen von Daten aufgeteilt werden:

1. Sprachsignale (*file*.r16 und *file*.l16) mit den jeweiligen Labelfiles (*file*.s1h),
2. vier Arten von Lexika:
   a) Lexikon der kanonischen Transkriptionen (*kielcdNUM*.lxc),
   b) Lexikon der tatsächlichen Realisierungen (*kielcdNUM*.lxv),
   c) Lexikon der kanonischen Transkriptionen und Anzahl ihrer Vorkommen (*kielcdNUM*.lsc),
   d) Lexikon der tatsächlichen Realisierungen und Anzahl ihrer Vorkommen (*keilcdNUM*.lsv),
3. Transliterationen (*file*.trl).

Da sich die Untersuchungen auf Varianten im Redefluss beziehen, rückt das Lexikon der tatsächlichen Realisierungen (2.b) als erstes ins Blickfeld. In diesem Wörterbuch ist jeweils das Wort in seiner kanonischen Form sowie alle realisierten Variationen des Wortes und die Anzahl derer Vorkommen aufgelistet. möchte man die Variabilität der

Spontansprache auf Wortebene untersuchen, diente dieses Lexikon als hervorragende Wissensquelle.

Für unsere Untersuchungen reichen die Einträge dieses Lexikons allerdings nicht aus. Lautbeeinflussung und -veränderung kommen auch wortübergreifend in beträchtlicher Menge vor (z.B. Assimilation bei gleichem An- und Ablaut, Abschnitt 2.4.1). Folglich werden Daten benötigt, welche ganze Äußerungen, und nicht nur sortierte Wörter, enthalten. Diese Art von Information ist in den Labeldateien des Korpus kodiert.

3.1 Extraktion und Ergänzung der Informationen aus dem Korpus

Eine Labeldatei des Kiel Korpus beinhaltet sehr viel Information zu den jeweiligen Äußerungen und ist in vier Teilen zu lesen: 1. die Transliteration, 2. die kanonische Transkription, 3. die variierte Realisierung der Äußerung, und 4. die Zeit-Ereignis-Liste (vgl. Tabelle 3.1).

BEISPIEL 2:

g071a013.s1h	*Name des Files*
HAH013: doch , auf jeden Fall . wunderbar .	*Teil 1*
oend d 'O x , Q aU f+ j 'e: d @ n f 'a l . v 'U n d 6 b a: 6 .	*Teil 2*
kend c: %d -h 'O x , Q- aU f+ j 'e: d @- n f 'a l . c: -p: v 'U n %d-n 6 b -h a:6 .	*Teil 3*
hend 2 #c: 2 ##%d 15 $-h 211 $'O ... 24686 $-h 24821 $a:6 27480 #.	*Teil 4*

Tabelle 3.1: Labelfile g071a013.s1h von Keil Korpus CD2

Aus einem solchen Labelfile wird der zweite Teil (kanonische Transkription) und dritte Teil (Transkription mit Variationen) extrahiert sowie alle Markierungen gemäß der Tabelle 2.5 aus Abschnitt 2.3.1 entfernt. Die restlichen Labels werden in zwei Listen zwischengespeichert: eine Liste mit der kanonischen Aussprache und eine Liste mit der tatsächlich realisierten Aussprache. Diese beiden Listen sind ab nun unter dem Begriff kanonische bzw. variierte Liste zu verstehen.

Als nächstes werden die Listen mit der kanonischen und variierten Aussprache Laut für Laut miteinander verglichen. Die zwei Laute, jeweils aus einer der Listen, und das Ergebnis des Vergleichs werden in einer Tabelle gespeichert. Stimmen die Laute der Listen überein, so wird in die Tabelle zu den Lauten die Kennzeichnung „`same`“ eingetragen. Stimmen die Laute der beiden Aussprachen nicht überein, so kann durch die besondere Annotation der Kieler Konventionen, der für die Änderung der Laute verantwortliche phonologische Prozess bestimmt und in die Tabelle, neben den beiden Lauten, eingetragen werden. Entsprechend den Kieler Konventionen aus Tabelle 2.4 im Abschnitt 2.3.1 wird zwischen folgenden phonologischen Prozessen unterschieden: „`substitution`“, „`deletion`“, „`glottalisation`“, „`nasalisation`“, und „`insertion`“.

Die bisher zusammengestellte Tabelle, welche später als Datensatz zum maschinellen Lernen dienen wird, ist mit den obigen Einträgen noch nicht vollständig. Da das maschinelle Verfahren so genau wie nur möglich die Änderungen der kanonischen zu variierten Aussprachen vorhersagen soll, ist es nur sinnvoll, mehr Information über die lautliche Umgebung und die phonetischen Eigenschaften der Laute mit einzubeziehen. Daher wird die Tabelle um die beiden Vorgänger des kanonischen und um die beider Vorgänger des variierten Lautes und zwei Nachfolger des kanonischen Lautes erweitert. Die Hinzunahme dieser Attribute soll die mangelnde morphologische Analyse der Wörter ergänzen, genauer gesagt soll auch ohne eine genaue morphologische Analyse erkannt werden, in welcher Silbe sich der aktuell betrachtete Laut befindet. Leider kann die Position der Silbe im Wort nicht ermittelt werden.

Zu den verschiedenen Lauten (kanonischer Laut, seine zwei Vorgänger und Nachfolger, sowie der variierte Laut und seine zwei Vorgänger) und dem phonologischen Prozess (gleich geblieben, Tilgung, Einfügung, Substitution, Glottalisierung und Nasalierung), habe ich mich entschieden, die Betonung der Silbe des Lautes und Artikulationsort und -art aller kanonischen Laute in die Tabelle mit aufzunehmen. Damit sollen eventuelle phonologische Regelmäßigkeiten einfacher erkannt werden. Mehr dazu in Abschnitt 3.2.

Natürlich können weitere Merkmale hinzu genommen (beispielsweise Position in der Silbe, falls die Information vorhanden ist), oder weggelassen werden (z.B. Betonung), doch um erste Experimente durchzuführen, habe ich mich auf die obigen festgelegt.

Da die zusammengestellte Tabelle sehr viele Merkmale der Laute enthält, wird sie im Folgenden Merkmalstabelle genannt. Die Merkmalstabelle dient als Datensatz für das statistische Lernen in Kapitel 4.

Der Übersicht wegen ist unten eine tabellarische Auflistung aller Einträge der Merkmalsliste abgebildet, Tabelle 3.3.

	Attribut	**Erläuterung**	**Werte**
1.	`can-2`	Vorvorgänger von `can0`	Instanzen der Tabelle 2.1 und 2.2
2.	`can-1`	Vorgänger von `can0`	(wie 1.)
3.	`can0`	**aktueller kanonischer Laut**	(wie 1.)
4.	`var-2`	Vorvorgänger von `var0`	(wie 1.)
5.	`var-1`	Vorgänger von `var0`	(wie 1.)
6.	`var0`	**aktueller variierter Laut**	(wie 1.)
7.	`can+1`	Nachfolger von `con0`	(wie 1.)
8.	`can+2`	Nachfolger von `can+1`	(wie 1.)
9.	`str_can`	Betonung von `can0`	numerische Werte `0`, `1`, `2`
10.	`place-2`	Artikulationsort von `can-2`	`vowel`, `bilabiar`, `labiodental`, `alveolar`, `palatalalveolar`, `palatal`, `velar`, `glottal`
11.	`manner-2`	Artikulationsart von `can-2`	`vowel`, `nasal`, `plosive`, `fircative`, `approximant`, `trill`, `lateral`
12.	`place-1`	Artikulationsort von `can-1`	(wie 10.)
13.	`manner-1`	Artikulationsart von `can-1`	(wie 11.)
14.	`place0`	Artikulationsort von `can0`	(wie 10.)
15.	`manner0`	Artikulationsart von `can0`	(wie 11.)
16.	`place+1`	Artikulationsort von `can+1`	(wie 10.)
17.	`manner+1`	Artikulationsart von `can+1`	(wie 11.)
18.	`place+2`	Artikulationsort von `can+2`	(wie 10.)
19.	`manner+2`	Artikulationsart von `can+2`	(wie 11.)
20.	`mod`	phonologischer Prozess	`same`, `deletion`, `substitution`, `nasalisation`, `insertion`, `glottalisation`

Tabelle 3.3: Übersicht aller Einträge der Merkmalstabelle

can-2	can-1	can0	var-2	var-1	var0	can+1	can+2	str	plc-2	mnr-2	...
z	'i:	t	z	'i:	t	$	Q	0	alv	fric	...
t	$	Q	t	$	Q-	E	s	0	alv	plos	...
$	Q	E	$	_	E-	s	$	0	$	$	...
Q	E	s	_	_	s	$	Q	0	glt	plos	...
s	$	Q	s	$	Q	'aU	s	0	alv	fric	...
s	$	Q	$	Q	-q	'aU	s	0	alv	fric	...

...	plc-1	mnr-1	pl0	mnr0	plc+1	mnr+1	plc+2	mnr+2	mod
...	vow	vow	alv	plos	$	$	glt	plos	same
...	$	$	glt	plos	vow	vow	alv	fric	tilg
...	glt	plos	vow	vow	alv	fric	$	$	tilg
...	vow	vow	alv	fric	$	$	glt	plos	same
...	$	$	glt	plos	vow	vow	alv	fric	same
...	$	$	glt	plos	vow	vow	alv	fric	glott

Tabelle 3.4: Kleiner Ausschnitt aus der Merkmalstabelle - dem Datensatz, welcher zum Trainieren eines statistischen Modells verwendet wurde.

3.2 Statistische Auswertung des Datensatzes

In diesem Abschnitt wird der erstellte Datensatz, die Merkmalstabelle, analysiert, um einerseits einen besseren Einblick in die Prozesse der spontanen Sprache zu gewinnen, und andererseits die Wahl der Attribute der Merkmalstabelle zu rechtfertigen. Die statistische Auswertung der Merkmalstabelle entspricht ungefähr dem Verfahren, um phonologische Regeln aus Daten zu extrahieren, jedoch wird anstelle der Ausarbeitung der einzelnen phonologischen Regeln gezeigt, dass die Statistik und ein geschickt gewählter Datensatz zum selben Endergebnis führen.

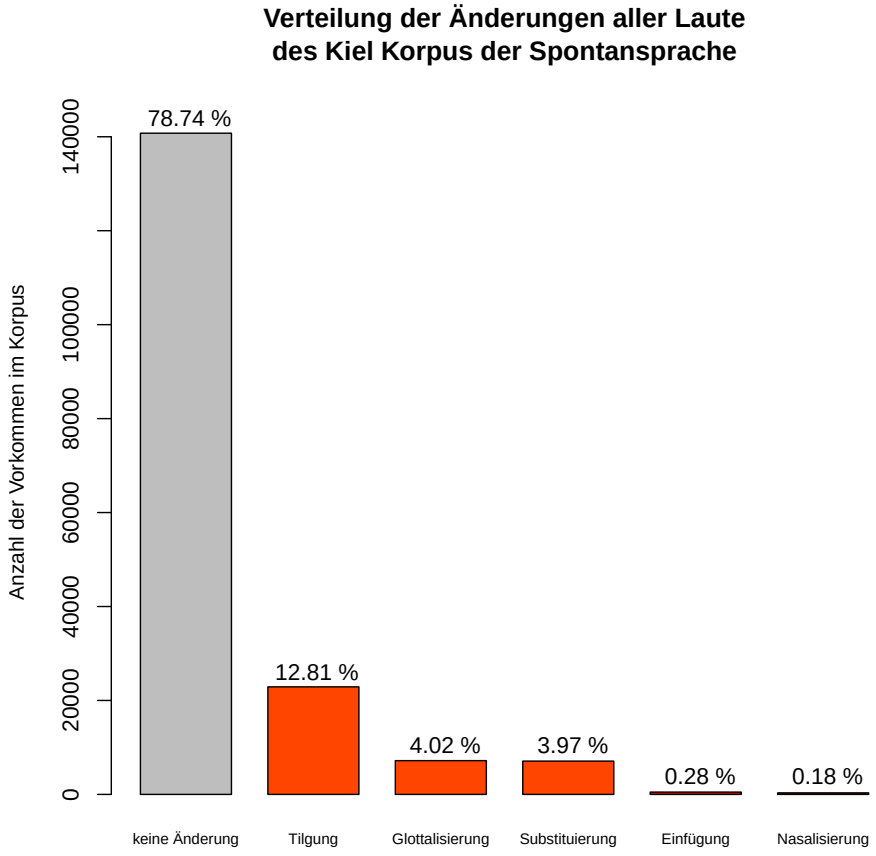

Tabelle 3.5: **Überblick über die Verteilung von Änderungen im Redefluss.** Den größten Anteil aller Änderungen, 60.3%, nimmt die Tilgung ein. Glottalisierung und Substitution sind mit fast gleichem Anteil von circa 18.8% vertreten, während die Einfügung von Lauten sowie die Nasalierung vergleichsweise selten vorkommen.

Da die Grundeinheit der Merkmalstabelle ein Laut ist, wird der Datensatz auf Lautebene analysiert. In Abbildung 3.5 wird die Verteilung aller Änderungen der Laute dargestellt, einschließlich der Laute, die keiner Änderung unterliegen. Von insgesamt 178.579 Lauten

des Kiel Korpus der Spontansprache werden 78,74% im Redefluss nicht verändert, die restlichen 21,26% unterliegen phonologischen Änderungen wie in Tabelle 2.4, Abschnitt 2.3.1, aufgelistet.

3.2.1 Analyse der Tilgung

In Abbildung 3.1 ist die prozentuale Verteilung der getilgten Laute dargestellt.

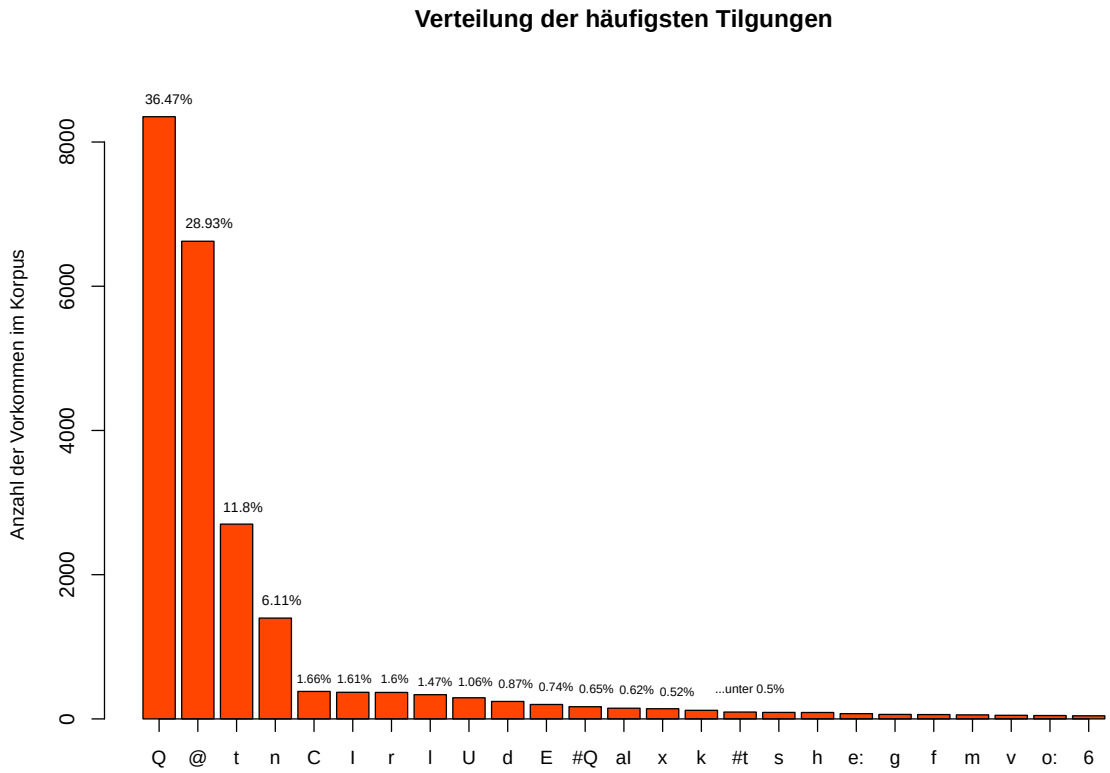

Abbildung 3.1: Verteilung der häufigsten 25 getilgten Laute.
Bereits die fünfthäufigste Tilgung (die /C/-Tilgung) ist mit gerade mal 1.66% vertreten. Ab der zehnthäufigsten Tilgung (/d/), erreichen die Vorkommen nicht mal 1%.

Die häufigsten drei Tilgungen, die des glottalen Verschlusslautes /Q/, des Schwa /@/ und des Plosivs /t/, werden nun genauer untersucht.

Nach Sortieren und Zählen der /Q/-Tilgungen im Korpus wird /Q/ am häufigsten am Wortanfang vor einem (unbetonten) Vokal getilgt. Dieses Ergebnis stimmt mit den Ergebnissen aus [Kemp, 1996] überein: im Redefluss wird meist keine Pause zwischen den

Wörtern eingesetzt und die Anfangsvokale werden eher „flüssig“ als „plötzlich“ realisiert. Einen Zusammenhang zwischen Tilgung von /Q/ und dessen Vorgänger oder Vorvorgänger, sowie zweitem Nachfolger, konnte nicht festgestellt werden.

Auch die Untersuchungen der /@/-Tilgungen bestätigen die Ergebnisse der Tabelle aus [Kemp, 1996]. Schwa-Tilgungen finden am häufigsten am Wortende, bzw. in der letzten Silbe im Wort statt, daher konnte ein direkter Zusammenhang zum Vorvorgänger des kanonischen Lautes oder des variierten Lautes nicht nachgewiesen werden. Allerdings ist eine starke Verbindung zum Nachfolger sowie zum zweiten Nachfolger des kanonischen Lautes zu erkennen: Liegt die Kombination can0 = /@/, can+1 = /n/ und can+2 = /$/ (Wortende) vor, so wird das Schwa getilgt. Genauer genommen, sind die häufigsten /@/-Tilgungen:

Tilgung	Beispiel	
st@n$ → st_n$	*Pfingsten*:	pf’INst@n → pf’INst_n
nt@n$ → nt_n$	*könnten*:	k’9nt@n → k’9nt_n
i:n@n$ → i:_n$	*ihnen*:	Qi:n@n → _i:_n
as@n$ → as_n$	*passen*:	p’as@n → p’as_n
Ef@n$ → Ef_n$	*treffen*:	tr’Ef@n → tr’Ef_n

Als nächstes werden die Tilgungen des Plosivs /t/ untersucht. Hier fällt auf, dass /t/ am häufigsten in einsilbigen Wörtern am Silbenende getilgt wird. Eine weitere Auffälligkeit ist die Tilgung von /t/ in der Affrikate /ts/ nach einem vorderen hohen Vokal (z.B. /I/). Entsprechende Regeln zu diesen Prozessen werden allerdings nicht in den phonologischen Regeln in [Kemp, 1996] aufgelistet, lediglich die Lautverschmelzung bei gleichem Silben Ab- und Anlaut. Hier die häufigsten /t/-Tilgungen:

Tilgung	Beispiel	
Unt#ts → Un_#ts	*Einundzwanzig*:	Q’aInunt#tsv’’antsIC → Q’aInun_#tsv’’antsIC
Unt$d → Un_$d	*und dann*:	QUnt$dan → QUn_$dan
Ist$Q → Is_$_	*ist in*:	QIst$QIn → Qis_$_In
antsI → an_sI	*Einundzwanzig*:	Q’aInunt#tsv’’antsIC → Q’aInun_#tsv’’an_sIC
ICt$Q → IC_$_	*nicht ab*:	nICt$Qap → nIC_$_ap

3.2.2 Analyse der Glottalisierung

96.52% der Glottalisierungen beziehen sich auf wortinitiale Vokale. Meist, sogar in 73.23% der Glottalisierungen, wird der Glottalverschlusslaut /Q/, welcher dem Vokal vorangeht, getilgt, dafür der Vokal glottalisiert. Ist dies nicht der Fall, wird der Vokal als Folge von Koartikulation glottalisiert.

Betrachtet man die Betonung der glottalisierten Vokale, so werden in unbetonten Silben Vokale öfter glottalisiert als in betonten Silben (67.96% zu 29.41%), die Glottalisierung in einer zweitbetonten Silben kann grundsätzlich ausgeschlossen werden.

Im Grunde kann dieser phonologische Prozess als eine Abschwächung des Glottalverschlusslautes im Falle der Tilgung von /Q/, oder als Koartikulationseffekt im Falle der Nicht-Tilgung von /Q/ verstanden werden.

Auch dieser Prozess wird in den Regeln in [Kemp, 1996] nicht gelistet.

3.2.3 Analyse der Substitution

Bei der Tilgung und Glottalisierung konnten gewisse phonologische Änderungen in sehr großer Häufigkeit beobachtet und damit ihre Regelmäßigkeit klar nachgewiesen werden. Die Substitution ist in diesem Sinne nicht so einfach zu analysieren. In Abbildung 3.2 sind die 20 häufigsten Substitutionen dargestellt. Die meisten Substitutionen machen weniger als 1% aller Ersetzungen aus.

Es gibt zwar auch allgemeine Substitutionsmuster (beispielsweise finden 36.7% aller Substitutionen am Wortende statt, 20.92% finden vor einem /@/ statt), um jedoch die genaue lautliche Umgebung der Prozesse zu untersuchen, müssen die einzelnen Substitutionen sehr detailliert analysiert werden. Meist kommen Substitutionen von Lauten in Verbindung mit vorangegangener Tilgung vor, welche komplexere phonologische Prozesse wie z.B. Assimilation bilden.

Die frequentesten /n/-zu-/m/-Substitutionen finden in den wortfinalen Silben /b@n/, /m@n/, /f@n/ mit nachfolgendem labiodentalen oder bilabialen Laut und getilgtem /@/ statt:

Substitution	Beispiel	
b@n$ → b_m$	*haben (wir)*:	ha:b@n → ha:b_m
m@n$ → m_m$	*kommen (wir)*:	kOm@n → kOm_m
f@n$ → f_m$	*schaffen (wir)*:	Saf@n → Saf_m

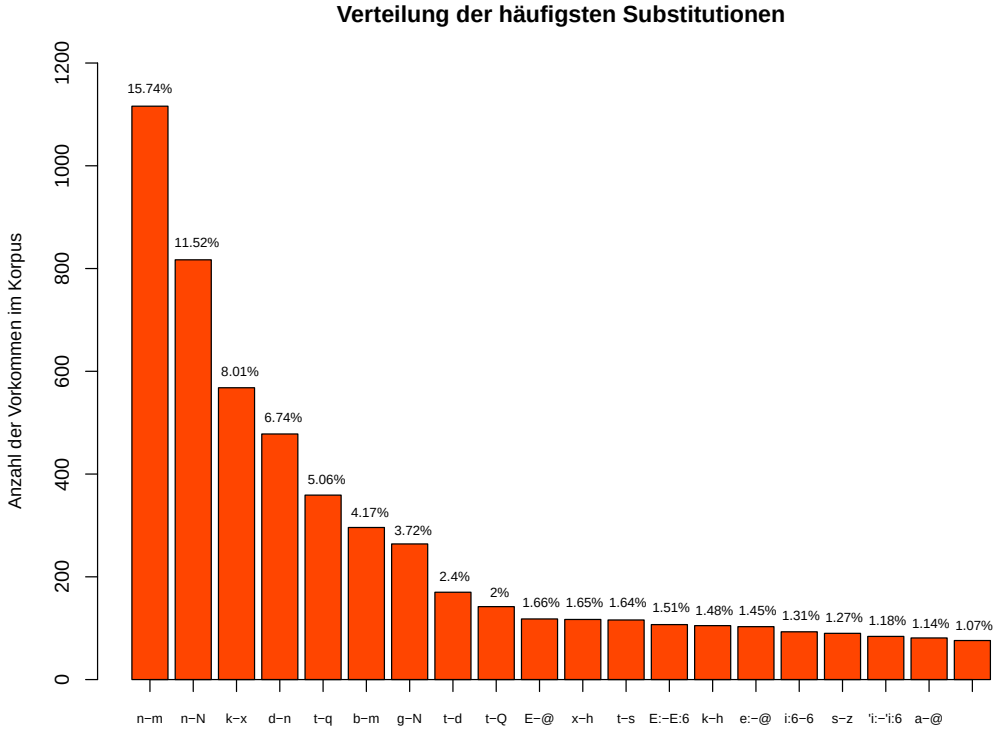

Abbildung 3.2: Verteilung der 20 häufigsten Substitutionen.
Wie auch bei den Tilgungen kommen alle nicht aufgeführten Substitutionen in unter 1% vor.

Bei der Substitution von `/n/` durch `/N/` können ähnliche Regelmäßigkeiten herausgearbeitet werden. Diese Substitution kommt meist in wortfinalen Silben /`g@n`/ und /`x@n`/ vor, wobei der nachfolgende Laut keine Rolle spielt:

Substitution	Beispiel	
`g@n$` → `g_N$`	*fragen*:	`fr'a:g@n` → `fr'a:g_N`
`x@n$` → `x_N$`	*machen*:	`max@n` → `max_N`

Auffallend ist, dass die meisten Regeln aus [Kemp, 1996] unter den Substitutionen zu finden sind, unter anderem auch die Änderung der Vokaldauer oder -qualität, stimmhaft-stimmlos-Änderungen oder die Nasalierung der Endsilbe `/-nd@n/`.

3.2.4 Analyse der Einfügungen von Lauten

Einfügungen von Lauten nehmen gerade mal 1.3% aller Variationen im Korpus ein. Trotzdem ist ihre Analyse mindestens so komplex wie die der Substitutionen. Die Abbildung 3.3 zeigt die Verteilung der häufigsten Einfügungen.

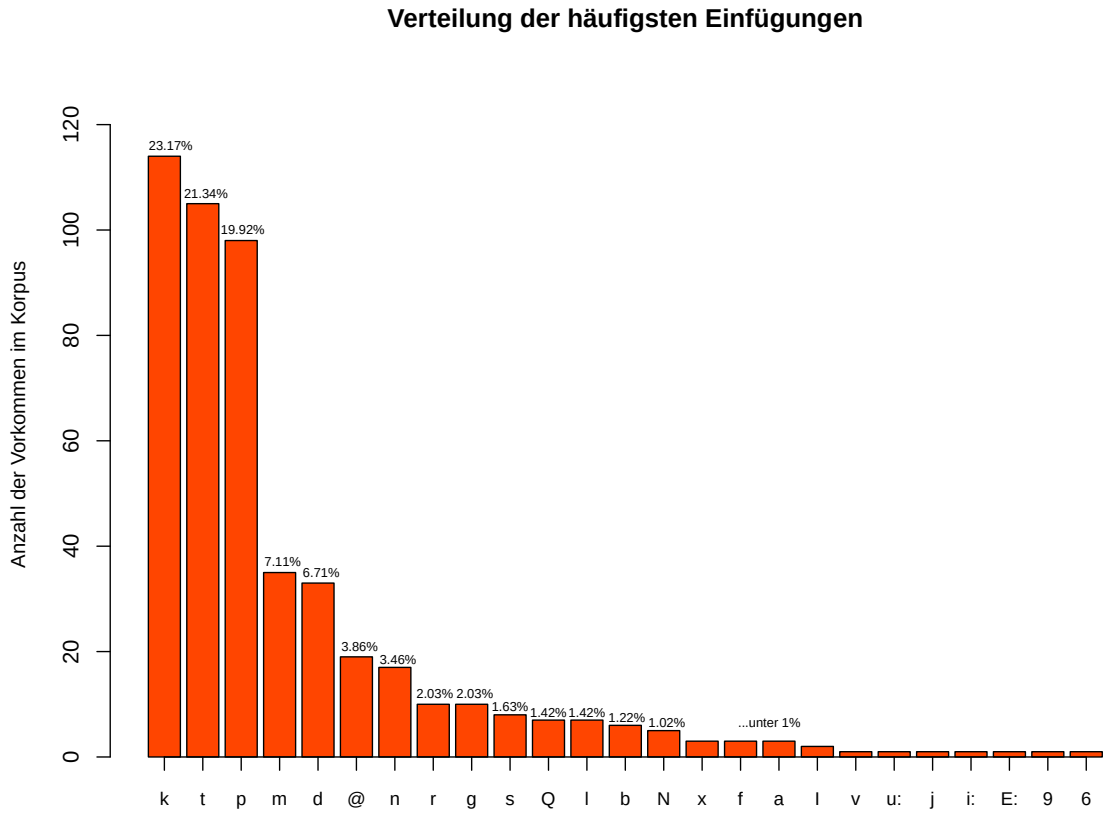

Abbildung 3.3: Verteilung der 25 häufigsten Einfügungen.
Gerade mal die ersten drei Einfügungen weisen ein höheres Vorkommen auf. Nur die ersten 14 Einfügungen haben ein Vorkommen von über 1%.

Auch diese Modifikation ist häufiger am Wortende zu beobachten, jedoch spielt hier der erste Laut des nachfolgenden Wortes ein große Rolle: Einfügungen treten häufiger am Wortende auf, wenn das darauf folgende Wort mit einem Plosiv anfängt. Trotzdem, wie auch bei der Substitution, müssen die Statistiken der Einfügung für jeden einzelnen Laut aufs genaueste herausgefiltert werden.

Beispielsweise kann für die Einfügung von `/k/` beobachtet werden, dass es überwiegend an eine Silbe, bestehend aus `/n/`,`/f/` oder `/ts/`, den Vokalen `/U/` oder `/a/` und dem Nasal `/N/`, gehängt wird:

Einfügung	Beispiel	
`nUN$` → `nUNk$`	*Ordnung*:	`Q'O6dnUN` → `Q'O6dnUNk`
`faN$` → `faNk$`	*Anfang*:	`Q'an#f''aN` → `Q'an#f''aNk`
`tsUN$` → `tsUNk$`	*Sitzung*:	`z'ItsUN` → `z'ItsUNk`

Die Einfügung von `/t/` kommt zwar auch in wortfinalen Silben vor, jedoch kann dieses

Phänomen genau so oft bei wortinternen Silben beobachtet werden. Meist wird /t/ zwischen zwei Silben eingefügt. Auffallend ist dabei die Artikulationsart des kanonischen Lautes und seines Nachfolgers, denn /t/ wird meist zwischen zwei alveolaren Lauten eingefügt (zwischen /n/ und /s/, oder /l/ und /s/):

Einfügung	Beispiel	
ns → nts	*günstig*:	g'YnstIC → g'YntstIC
ls → lts	*als*:	Qals → qalts
nst → nts	*Dienstag*:	d'i:nsta:k → d'i:ntsta:k

Diese Phänomene sind als Assimilation bezüglich der Artikulationsart [Wesenick, 1996] zu verstehen. Leider werden solche Regeln in [Kemp, 1996] nicht angesprochen, da Allgemein auf Einfügungen basierte Regeln außer Acht gelassen wurden.

3.2.5 Analyse der Nasalierung

Nasalierung ist an einen strengen Lautkontext gebunden: nasaliert werden Vokale vor einem Nasallaut. Im Gegensatz zu den bisherigen Modifikationen im Redefluss findet die Nasalierung sehr oft in der ersten Silbe im Wort statt, sogar in 33.13% aller Nasalierungen, oder in Einsilblern, was 21.58% der Nasalierungen ausmacht. Sogar die Nasalierung wortinterner Silben kommt öfter vor als die der wortfinalen (14.58% zu 11.55%).

Da die Nasalierung eine Abschwächung des Nasalkonsonanten darstellt, wird eine Nasalierung regelmäßig von der Tilgung des nachfolgenden Nasals begleitet. Die zahlreichste und naheliegendste Nasalierung ist die des Vokals zwischen zwei Nasallauten, dicht gefolgt von Nasalierung in einsilbigen Funktionswörtern. Untersucht man die Art der nasalierten Vokale, so sind die Vokale /U/, /o:/ und /a/ in unwesentlicher Mehrzahl vorhanden.

Nasalierung	Beispiel	
$mo:n → $mo:~	*Montag*:	m'o:nta:k → m'o:nta:~_k
$So:n → $So:~	*schon*:	So:n → So:~_
Yn → Y~	*Fünfzehn*:	f'Ynfts''e:n → f'Y~_fts''e:n

3.3 Diskussion der Datenanalyse

Die Analyse konnte die Qualifikation der gewählten Merkmale für das statistische Lernen bestätigen. Alle in der Merkmalstabelle vorkommenden Attribute wurden verwendet und

benötigt um auf Regelmäßigkeiten von phonologischen Änderungen zu schließen. Obwohl keine genaue morphologische Analyse zu Verfügung stand, konnten die Modifikationen den entsprechenden Silben und den Positionen in der Silbe zugeordnet werden. Trotzdem wäre eine morphologische Analyse als weiteres Attribut eine nützliche Ergänzung.

Außerdem konnte eine deutliche Verbindung der statistischen Analyse des Datensatzes zu phonologischen Regeln nachgewiesen werden. Fast alle Regeln aus [Kemp, 1996] konnten durch statistisches Suchen ermittelt werden, auch Regelmäßigkeiten die nicht in [Kemp, 1996] gelistet sind. Es könnte verschiedene Gründe für das Nichtvorhandensein bestimmter Regelmäßigkeiten geben, da jedoch in [Kemp, 1996] nicht genauer über den Hintergrund der Regelfindung berichtet wird, und die Unterschiede eher gering sind, wird nicht weiter darauf eingegangen.

Dazu ist noch zu beachten, dass das Kiel Korpus der Spontansprache nur eingeschränkt Spontansprache darstellt. Das Thema (Terminvereinbarung) und die vorhandenen Hilfsmittel (Kalender) sind vorgegeben, und folglich wurden bestimmte Phrasen und Wörter deutlich häufiger gesprochen, was natürlich auch die Statistik der Modifikationen beeinflusst. So kommen Zahlen, Daten, Wochentage und das Wort Pfingsten in auffallender Häufigkeit vor, was als Folge z.B. die Frequenz der `/k/`-Einfügungen durch `pf'INst@n` → `pf'INkst_n` ansteigen lässt.

Aus den in diesem Abschnitt vorgetragenen Untersuchungen können bereits die ersten Mutmaßungen zu dem Ergebnis der Vorhersage von Lauten oder Änderungen aufgestellt werden. Da die Nasalierung und Einfügung von Lauten äußerst selten vertreten sind, sollten Vorhersagen zu diesen Ereignissen nicht erwartet werden. Andererseits kann mit einer hohen Genauigkeit der Vorhersage der Tilgung von `/@/` sowie Glottalisierung gerechnet werden, da diese nicht nur zahlreich vorhanden sind, sondern auch von wenig Bedingungen abhängen. Lediglich die Position im Wort, damit auch die Betonung, und der nachfolgende Laut beeinflussen dieses Phänomen.

4 Ausarbeitung des Spontaneous Speech Tools

In diesem Abschnitt wird der in Abschnitt 2.5.3 präsentierte Ansatz in die Praxis umgesetzt. Das Leitmotiv des Ansatzes ist die gänzlich automatische Vorhersage von phonologischen Änderungen im Redefluss, basierend auf einem phonologischen Datensatz. Aus einem annotierten Spontansprachenkorpus soll ein Data-Mining-Verfahren[1] lernen ob und wie sich ein Laut verändert, und dieses Wissen als Modell festhalten. Das so entstandene Modell wird anschließend in das Spontaneous Speech Tool eingearbeitet.

4.1 Ermittlung des besten Lernalgorithmus

Nachdem ein sinnvoller Datensatz aus dem Korpus herausgearbeitet wurde, wird untersucht, welcher Lernalgorithmus die besten Ergebnisse erzielt, um gewünschte Attribute, wie zum Beispiel den variierten Laut, vorher zu sagen.

Als Hilfswerkzeug dient das Data-Mining-Programm WEKA (Version 3.6.1), welches eine große Auswahl an maschinellen Lernalgorithmen zur Verfügung stellt [Witten et al., 2005]. Es wurden nicht nur Versuche zu allerhand diversen Lernalgorithmen vorgestellt, sondern auch Experimente zur Vorhersage unterschiedlicher Attribute durchgeführt.

[1] Data Mining (“ Datenbergbau“): Oberbegriff für datenbasiertes Lernen, bzw. Problemlösung durch Analyse großer Datenmengen

4.1.1 WEKA: eine kompakte Übersicht

Eine ausführliche Dokumentation zur Benutzung von WEKA ist in [URL: WEKA] gegeben. Hier wird lediglich ein Überblick des WEKA-Interfaces „Explorer“ gegeben, da dieser Teil des Programms für die Experimente verwendet wurde.

Die in WEKA zur Verfügung stehenden Lernalgorithmen werden in sechs Kategorien aufgeteilt: „`bayes`“, „`functions`“, „`lazy`“, „`meta`“, „`mi`“, „`misc`“, „`trees`“ und „`rules`“. In Tabelle 4.1 werden die Kategorien der Verfahren kurz beschrieben.

Kategorie	**Beschreibung**
`bayes`	Verfahren, welche die Instanzen mittels der Wahrscheinlichkeiten des Bayes-Theorems den Klassen zuordnen
`functions`	Verschiedene Algorithmen, welche auf linearer Regression, neuronalen Netzen oder Support Vector Machines basieren
`lazy`	Basieren auf der „Lazy-Learning“-Methode (das LernModell wird erst zum Zeitpunkt der Anfrage gebildet)
`meta`	Verwenden die Ausgabe anderer Verfahren und erhöhen damit die Leistungsfähigkeit mancher Grundalgorithmen
`mi`	Klassifizierer für mehrfach-Instanz-Daten
`misc`	Algorithmen, welche keiner der obigen Kategorien zuzuordnen sind
`trees`	Klassifizierung durch eine herausgearbeitete Baumstruktur (z.B. von Bedingungen), am häufigsten verwendete Verfahren
`rules`	Verfahren, welche mit aus den Daten gewonnenen Regeln klassifizieren

Tabelle 4.1: Überblick der Lernverfahren in WEKA

Ein WEKA-Lernalgorithmus wird auf einen Datensatz angewandt und das Resultat als Modell gespeichert. Dieser Prozess wird *Trainieren* genannt. Wendet man ein Modell auf Daten an, die nicht Teil der Trainingsdaten waren, so spricht man vom *Testen* des Modells.

Die Ausgabe der Trainings- und Testprozesse berechnet eine Vielzahl statistischer Werte, wie zum Beispiel die Kappa-Statistik, den statistischen Gesamtfehler (“ mean absolute error“) oder die Anzahl korrekt und inkorrekt klassifizierter Instanzen. Für die Ermittlung des besten Lernalgorithmus sind für die Zwecke dieser Arbeit hauptsächlich die Werte *Recall* und *Precision* relevant.

Unter Recall, oder *Trefferquote*, wird das Verhältnis der Anzahl der richtig klassifizierten Instanzen einer Klasse, zur Gesamtanzahl aller Instanzen dieser Klasse verstanden. Mit anderen Worten, man kann den Anteil der korrekt vorhergesagten Instanzen zu allen Instanzen dieser Klasse beurteilen.

Die Precision, oder *Genauigkeit*, beschreibt das Verhältnis von richtig klassifizierten Instanzen einer Klasse zur Anzahl aller zu dieser Klasse klassifizierten Instanzen. Dieser Wert beschreibt den Anteil der korrekt klassifizierten Instanzen aus allen Vorhersagen für diese Klasse.

Genau genommen ist für die Generierung von Aussprachevarianten die Precision-Rate von größerer Bedeutung. Es ist wichtiger, dass die Vorhersagen genau sind, als das alle Instanzen einer Klasse der richtigen Klasse zugeordnet werden.

Um die Überlegungen dazu etwas zu verdeutlichen, werden zwei Beispiele präsentiert. Im ersten Beispiel, Tabelle 4.2, erreicht die Trefferquote sogar 100%. Betrachtet man jedoch ihre Vorhersagen, so sind die Ergebnisse völlig unbrauchbar: Das Verfahren klassifiziert jede Instanz zu der am weitesten verbreiteten Klasse (im Beispiel unten, Klasse „`same`“). Das zweite Beispiel zeigt eine akzeptable Vorhersagequalität.

BEISPIEL „ZeroR“:

Vorhersage:	gehört zur Klasse `same`	gehört nicht zur Klasse `same`
	119.184	0
Korrekte Vorhersage	93.998	0
Falsche Vorhersage	25.186	0
Tatsächliche Verteilung:	Anzahl der Laute der Klasse „`same`“:	Anzahl der Laute der Klasse „nicht `same`“
	93.998	25.186

$$\rightarrow \textsc{Trefferquote} = 93.998/93.998 = 100\%$$
$$\rightarrow \textsc{Genauigkeit} = 93.998/119.184 = 78,86\%$$
$$(\rightarrow \text{Trefferquote und Genauigkeit für alle anderen Klassen beträgt } 0\%)$$

Tabelle 4.2: Herleitung der Trefferquote (Recall) und Genauigkeit (Precision) für die Vorhersage „Laut wird nicht verändert“ des WEKA-Verfahrens „ZeroR“

BEISPIEL „J48“:

Vorhersage:	gehört zur Klasse `same`	gehört nicht zur Klasse `same`
	96.192	25.192
Korrekte Vorhersage	90.435	19.490
Falsche Vorhersage	5.757	5.702
Tatsächliche Verteilung:	Anzahl der Laute der Klasse „`same`“:	Anzahl der Laute der Klasse „nicht `same`“
	93.998	25.186

$$\rightarrow \textsc{Trefferquote} = 90.435/93.998 = 96,21\%$$
$$\rightarrow \textsc{Genauigkeit} = 90.435/96.192 = 94,06\%$$

Tabelle 4.3: Herleitung der Trefferquote (Recall) und Genauigkeit (Precision) für die Vorhersage „Laut wird nicht verändert“ des WEKA-Verfahrens „J48“

4.1.2 Experimente mit den Daten

4.1.2.1 Vorüberlegungen

Das in das Spontaneous Speech Tool eingebettete trainierte Modell soll eine Vorhersage treffen, ob oder wie sich ein Laut verändert.

In der Terminologie der verwendeten Merkmalstabelle bedeutet dies, dass eine Vorhersage über den aktuellen kanonischen Laut, in der Merkmalstabelle in der Spalte „`can0`“, getroffen werden soll, zu welchem variiertem Laut, „`var0`“, er modifiziert wird. Alternativ kann vorhergesagt werden, welche Art von Änderung stattgefunden hat, und in einer weiteren Vorhersage der genaue Laut bestimmt werden (d.h. im ersten Schritt werden die Werte der Spalte „`modification`“ der Merkmalstabelle vorhergesagt, im zweiten Schritt die Werte der Spalte „`var0`“).

In diesem Zusammenhang müssen folgende Fragen beantwortet werden:

1. Welcher Lernalgorithmus liefert die beste Genauigkeitsrate?

2. Soll der Laut direkt vorhergesagt werden, oder ist es sinnvoller zuerst eine Vorhersage über den phonologischen Prozess zu ermitteln (bleibt der Laut gleich, wird er getilgt, ersetzt, nasaliert, glottalisiert, oder ein neuer Laut eingefügt), und erst dann den genauen Laut zu bestimmen?

3. Sollen Informationen über die variierten Vorgänger des zu vorhersagenden Lau-

tes in das Modell mit einbezogen werden (die Spalten „`var-2`“ und „`var-1`“ der Merkmalstabelle)?

Weshalb die erste Frage gestellt wurde, ist plausibel, doch weshalb man sich mit den Fragen zwei und drei auseinander setzen muss, muss vorerst erklärt werden.

Zur Frage eins. Um den besten Lernalgorithmus zu finden, wurden alle von WEKA zur Verfügung stehenden Algorithmen auf den Datensatz angewandt. Zwei Drittel der Daten wurden zum Trainieren eines Modell verwendet, welches auf dem übrigen Drittel getestet wurde.

Zur Frage zwei. Die einfachste Art, ein Modell mit den Daten der Merkmalstabelle zu trainieren, ist, den variierten Laut direkt vorherzusagen. Kann bei dieser Art zu trainieren eine hohe Genauigkeit erzielt werden, so könnte man ein solches Modell direkt, d.h. ohne Zwischenschritte in das Spontaneous Speech Tool einbinden. Da die vorhandenen Daten eine sehr ungleichmäßige Verteilung der einzelnen Modifikationen aufweist, könnten seltene Regelmäßigkeiten eine zu geringe Wahrscheinlichkeit zugewiesen bekommen, so dass sie gar nicht vorhergesagt würden. Nasalierung kommt beispielsweise nur 141 mal im gesamten Korpus vor, folglich kommt Nasalierung eines konkreten Vokals noch seltener vor. Ein statistisches Verfahren würde somit die Nasalierung eines bestimmten Vokals mit hoher Wahrscheinlichkeit überhaupt nicht vorhersagen. Daher ist es eine Überlegung wert, zuerst die Modifikation vorherzusagen und anschließend den variierten Laut.

Dies bedeutet, dass im Falle der zwei-Schritt-Vorhersage der zweite Schritt ausschließlich für die Vorhersage der variierten Laute der Kategorien „`substitution`“ und „`insertion`“ benötigt wird. Alle anderen Elemente sind nach Schritt eins, Vorhersage der Modifikation, bereits vollständig bestimmt.

Daher werden unter diesem Punkt zwei Vorhersagemethoden untersucht:

1. Direkte Vorhersage des variierten Lautes
2. Zwei-Schritte-Methode:
   - Schritt 1: Vorhersage der Änderung (Klassifikation der Laute zu den Kategorien „`same`“, „`substitution`“, „`insertion`“, „`deletion`“, „`glottalisation`“, „`nasalisation`“)
   - Schritt 2: Vorhersage des genauen Variationslautes „`var0`“

Nebenbemerkung: die variierten Laute der Kategorien „`same`“, „`deletion`“, „`nasalisation`“ und „`glottalisation`“ müssen nicht durch die Vorhersage in Schritt 2 bestimmt werden!
Denn: Die Laute der Kategorie „`same`“ werden nicht modifiziert und einfach übernommen, die Laute der Kategorie „`deletion`“ werden getilgt, d.h. durch ein Leerzeichen ersetzt, und die Laute in „`nasalisation`“ und „`glottalisation`“ werden jeweils um ein Nasalierungs- (/∼/) oder Glottalisierunssymbol (/q/) ergänzt.

Wie anfangs geschildert, möchte man den besten Lernalgorithmus anhand der Precision-Werte aussuchen. Bei einer zwei-Schritte-Methode (Schritt 1: Vorhersage der Modifikationsklasse, Schritt 2: zu welchem Laut wird der kanonische Laut modifiziert?) muss die Precision-Rate durch eine zusammengesetzte Gleichung errechnet werden.

Herleitung der Gleichung zur Berechnung der zusammengesetzten Genauigkeitsrate der Zwei-Schritte-Methode:
Betrachte folgende Variablenbelegung:

A:	Datensatz
#A:	Anzahl aller Instanzen des Datensatzes A
A_i für $i \in 1...n$:	eine Partition von A wobei $A_l \cap A_k = 0 \; \forall l, k \in 1...n$
$\#A_i$:	Anzahl der Instanzen in A_i
$P_1(A_i)$:	die resultierende Genauigkeitsrate nach Anwendung eines geeigneten Modells auf die Teilmenge A_i aus Schritt 1
$P_2(A_i)$:	die resultierende Genauigkeitsrate nach Anwendung eines geeigneten Modells auf die Teilmenge A_i aus Schritt 2

Da die Teilmengen A_i keine gemeinsamen Elemente haben, kann die Precision der Zwei-Schritte-Methode wie folgt dargestellt werden:

$$\text{2-Step-Precision} = \sum_{i=1}^{n} \frac{\#A_i}{\#A} \cdot P_1(A_i) \cdot P_2(A_i)$$

Für die Daten der Merkmalstabelle kann nun folgende Formel aufgestellt werden:

$$\begin{aligned} \text{2-step-Precision} &= \frac{\#A_{same}}{\#A} \cdot P_1(A_{same}) \\ &+ \frac{\#A_{deletion}}{\#A} \cdot P_1(A_{deletion}) \\ &+ \frac{\#A_{nasalisation}}{\#A} \cdot P_1(A_{nasalisation}) \end{aligned}$$

$$
\begin{aligned}
&+ \quad \frac{\#A_{glottalisation}}{\#A} \cdot P_1(A_{glottalisation}) \\
&+ \quad \frac{\#A_{insertion}}{\#A} \cdot P_1(A_{insertion}) \cdot P_2(A_{insertion}) \\
&+ \quad \frac{\#A_{substitution}}{\#A} \cdot P_1(A_{substitution}) \cdot P_2(A_{substitution})
\end{aligned}
$$

Beachte auch hier: nur die Teilmengen $A_{substitution}$ und $A_{insertion}$ müssen durch Schritt zwei bestimmt werden, folglich gibt es nur für diese Kategorien eine Precision in Schritt zwei.

Zur Frage drei. In Abschnitt 3.1 wurde, unter anderem, darüber diskutiert, weshalb es sinnvoll ist, die zwei Vorgänger des aktuellen kanonischen Lautes „`can0`“ und variierten Lautes „`var0`“ in die Merkmalstabelle mit aufzunehmen. Auch wenn sich in Abschnitt 3.1 die Verwendung dieser Laute als ausgesprochen nützlich erwiesen hat (da der Datensatz über keine vollständige Morphologie verfügt), so bereitet die Verwendung der beiden Vorgänger des variierten Lautes bei der Programmierung des Spontaneous Speech Tools Schwierigkeiten.

Das Tool soll aus der phonetischen, kanonischen Grundform Varianten erzeugen. Das heißt, dass das Tool anfangs nur die kanonische Transkription zur Verfügung stehen hat und keinerlei variierte Laute kennt, also auch keine der variierten Vorgänger. Dies stimmt jedoch nicht mit den Trainingsdaten des Modells überein, da die Merkmalstabelle die Spalten „`var-2`“ und „`var-1`“ enthält.

Dieses Problem kann auf zwei Wegen gelöst werden: (a) man implementiert das Tool so, dass die kanonische Transkription der Äußerung schrittweise von links nach rechts durchlaufen wird, so dass das Ergebnis der Vorhersage der vorangegangenen Laute für die Spalten „`var-2`“ und „`var-1`“ der Merkmalstabelle verwendet werden kann, oder (b) man trainiert das Modell von vornherein ohne diese Attribute.

Um sich für eine der vorgeschlagenen Lösungen zu entscheiden, soll für bestimmte Algorithmen, und zwar diejenigen, welche die besten Ergebnisse auf den Daten mit Vorgängern der Variationen erzielen, auch ein Durchlauf ohne die genannten Attribute gemacht werden. Das Ergebnis der Genauigkeitsrate soll zur Entscheidung beitragen. Unterscheiden sich die Ergebnisse nur minimal, soll Methode (b) bevorzugt werden, sind die Unterschiede merkbar größer, wird Methode (a) gewählt.

4.1.2.2 Die Experimente

Die Merkmalstabelle wurde in Trainings- und Testdaten aufgeteilt. Zwei Drittel (ca. 66%) dienten als Trainingsdaten, auf dem Rest (ca. 33%) wurde getestet. Die Ergebnisse sind in den Tabellen 4.8, 4.9, 4.10 und 4.11 am Ende dieses Abschnittes dargestellt.

1. Direkte Vorhersage des aktuellen variierten Lautes:

1. (a) Experimente mit Daten, welche die Attribute „`var-2`“ und „`var-1`“ beinhalten
In der Tabelle 4.8 werden die Ergebnisse der direkten Vorhersage des Variationslautes „`var0`“ gezeigt. Der beste Algorithmus ist das SimpleCart-Verfahren, welches eine Genauigkeit von 91.5% erreicht.

1. (b) Wiederholung der besten Algorithmen aus 1. (a), diesmal mit Daten ohne die Attribute „`var-2`“ und „`var-1`“
In Tabelle 4.5 sind die Ergebnisse zu diesen Experimenten zu sehen.

Verfahren	mit „`var-2`“ und „`var-1`“	ohne „`var-2`“ und „`var-1`“	Unterschied (in %)
SimpleCart	91.5	85.3	6.77
PART	91.3	85.5	6.35
REPTree	91.2	85.6	6.14
BFTree	91.2	84.8	7.01

Tabelle 4.5: **Direkte Vorhersage des variierten Lautes:** Übersicht der Precision der besten vier WEKA-Algorithmen mit und ohne die zwei Vorgänger des variierten Lautes, sowie der errechnete prozentuale Unterschied zwischen den jeweiligen Methoden.

Die Genauigkeit bei der Vorhersage des Variationslautes ist ohne die Attribute des Vor- und Vorvorgängers im Durchschnitt 6.5% schlechter als die Genauigkeit der Vorhersage mit den diskutierten Attributen. Dies bedeutet, dass die Attribute des Vorvorgängers „`var-2`“ und Vorgängers „`var-1`“ für die Vorhersage des aktuellen variierten Lautes „`var0`“ von Bedeutung sind und nicht ausgelassen werden können.

2. Zwei-Schritte-Methode:

2. (a) Experimente zur Zwei-Schritte-Methode, wobei die Datensätze die Attribute „`var-2`“ und „`var-1`“ beinhalten
In der Tabelle 4.9 ist die Genauigkeitsrate der Algorithmen im ersten Schritt, Vorhersage der Modifikation, aufgelistet. Das Verfahren mit der höchsten Precision ist der END-Algorithmus mit einer Genauigkeit von 92.2%. In den Tabellen 4.10 und 4.11 sind die

Ergebnisse des zweiten Schrittes, Vorhersage des variierten Lautes, dargestellt: Für die Vorhersage von Einfügungen erzielt der Algorithmus IB1 die besten Ergebnisse (73.3%) und für die Vorhersage von Substitutionen ist es der Algorithmus Decorate (80.6%).

Nun muss die Genauigkeit für die Kombination dieser Algorithmen berechnet werden. Als erstes werden die genauen Werte der Variablen festgelegt (siehe oben „Herleitung der Gleichung zur Berechnung der zusammengesetzten Genauigkeitsrate der Zwei-Schritte-Methode“). Dafür wird das Ergebnis des SimpleCart-Verfahrens für die Vorhersage der Modifikation genauer analysiert. Die WEKA-Ausgabedatei gibt Aufschluss über die Genauigkeit der Vorhersage einzelner Kategorien:

Kategorie	Precision in %
`same`	94.5
`glottalisation`	99.9
`deletion`	81.6
`nasalisation`	97.2
`insertion`	100.0
`substitution`	67.7

Daraus ergibt sich folgende Variablenbelegung:

$$
\begin{array}{llll}
\#A & = 59\ 586 & P_1(A_{same}) & = 94.5 \\
\#A_{same} & = 46\ 723 & P_1(A_{glottalisation}) & = 99.9 \\
\#A_{glottalisation} & = 2\ 691 & P_1(A_{deletion}) & = 81.6 \\
\#A_{deletion} & = 7\ 731 & P_1(A_{nasalisation}) & = 97.2 \\
\#A_{nasalisation} & = 141 & P_1(A_{insertion}) & = 100.0 \\
\#A_{insertion} & = 2\ 085 & P_1(A_{substitution}) & = 67.7 \\
\#A_{substitution} & = 215 & P_2(A_{insertion}) & = 80.6 \\
 & & P_2(A_{substitution}) & = 73.3
\end{array}
$$

Setzt man die Werte in die 2-Step-Precision-Formel ein, so ergibt sich eine endgültige Genauigkeitsrate für die Zwei-Schritte-Methode von 91.58%.

2. (b) Wiederholung der besten zwei-Schritte-Experimente mit den Attributen „`var-2`“ und „`var-1`“

Die Tabelle 4.7 zeigt den prozentualen Unterscheid der Precision des ersten Schrittes der Zwei-Schritte-Methode mit und ohne die die variierten Vorgänger.

Verfahren	mit „`var-2`“ und „`var-1`“	ohne „`var-2`“ und „`var-1`“	Unterschied (in %)
END	92.2	86.7	5.96
ND	92.1	86.7	5.86
PART	91.9	86.6	5.76
J48	91.9	86.7	5.65

Tabelle 4.7: **Zwei-Schritte-Methode:** Übersicht der Precision der besten vier WEKA-Algorithmen mit und ohne die zwei Vorgänger des variierten Lautes, sowie der errechnete prozentuale Unterschied zwischen den jeweiligen Methoden.

Die Unterschiede der Genauigkeit der Methoden mit und ohne die Attribute des Vor- und Vorvorgängers des variierten Lautes sind erheblich. Im Durchschnitt ist ein Modell um 6% schlechter, verwendet man einen Datensatz ohne die genannten Attribute (vergleichbar mit der Abweichung der Methode der direkten Vorhersage, siehe Punkt 1. (b)). Dieser Wert reicht aus, um einen Datensatz ohne die erwähnten Attribute auszuschliessen. Eine Berechnung der genauen 2-Step-Precision muss für diese Verfahren daher nicht aufgestellt werden.

Tabelle 4.8: Direkte Vorhersage des variierten Lautes `var0`

Verfahren	**Precision (%)**
BayesNet	82.0
NaiveBayes	81.7
NaiveBayesUpd.	81.7
IB1	85.5
IBk	88.8
KStar	89.2
LWL	29.1
HyperPipes	40.3
VFI	86.3
AttributeSel.Class.	77.3
Bagging	91.3
Class.ViaClust.	2.7
END	90.4
FilteredClassifier	91.1
RandomSubSpace	91.3
ClassBalancedND	90.5
DataNearBND	90.7
ND	90.0
ConjunctiveRule	2.8
DecisionTable	89.5
DTNB	89.9
JRip	90.4
NNge	87.3
OneR	77.3
PART	91.3
Ridor	88.4
ZeroR	1.7
BFTree	91.2
DecisionStump	2.8
J48	91.1
J48graft	91.0
LADTree	43.1
RandomTree	90.7
REPTree	91.2
SimpleCart	**91.5**

Tabelle 4.9: Vorhersage von Änderungen (Tilgung, Glottalisierung, Einfügung, Nasalierung, Substitution oder keine Änderung)

Verfahren	**Precision (%)**
BayesNet	86.0
NaiveBayes	85.6
NaiveBayesUpd.	85.6
IB1	89.1
IBk	91.2
KStar	91.5
LWL	81.9
HyperPipes	75.0
VFI	81.6
AttributeSel.Class.	78.9
Bagging	91.9
Class.ViaClust.	59.4
END	**92.2**
FilteredClassifier	91.9
RandomSubSpace	91.2
ClassBalancedND	92.1
DataNearBND	92.0
ND	92.1
ConjunctiveRule	73.6
DecisionTable	91.5
DTNB	90.7
JRip	91.2
OneR	78.9
PART	91.9
Ridor	91.0
ZeroR	61.5
DecisionStump	73.6
J48	91.9
J48graft	91.9
NBTree	92.0
RandomForest	91.5
RandomTree	90.9
REPTree	91.8
SimpleCart	91.7

Tabelle 4.10: Vorhersage der Klasse „`insertion`“

Verfahren	**Precision (%)**
BayesNet	66.5
NaiveBayes	58.9
NaiveBayesUpd.	58.9
IB1	**73.3**
IBk	69.9
KStar	70.7
LWL	58.1
AttributeSel.Class.	70.4
Bagging	69.3
Class.ViaClust.	16.5
Dagging	56.7
Decorate	71.9
END	68.5
FilteredClassifier	71.0
RandomSubSpace	67.8
RotationForest	70.7
ClassBalancedND	67.5
DataNearBND	63.0
ND	59.9
ConjunctiveRule	23.3
DecisionTable	62.7
DTNB	66.5
JRip	59.9
NNge	69.6
OneR	64.4
PART	69.6
Ridor	71.1
ZeroR	3.0
BFTree	64.1
DecisionStump	24.0
FT	73.0
J48	71.0
J48graft	70.7
NBTree	69.7
RandomForest	70.3
RandomTree	55.1
REPTree	70.4
SimpleCart	71.3

Tabelle 4.11: Vorhersage der Klasse „`substitution`“

Verfahren	**Precision (%)**
BayesNet	74.5
NaiveBayes	73.1
NaiveBayesUpd.	73.1
IB1	76.6
IBk	77.4
KStar	78.4
LWL	51.7
AttributeSel.Class.	61.4
Bagging	77.8
Class.ViaClust.	7.2
Dagging	70.7
Decorate	**80.6**
END	77.9
FilteredClassifier	79.4
RandomSubSpace	78.4
RotationForest	79.9
ClassBalancedND	76.2
DataNearBND	74.9
ND	68.9
ConjunctiveRule	7.9
DecisionTable	73.2
DTNB	73.1
JRip	76.6
NNge	75.4
OneR	61.7
PART	74.1
Ridor	74.1
ZeroR	5.8
BFTree	78.2
DecisionStump	7.9
FT	78.3
J48	79.4
J48graft	78.9
LADTree	6.0
RandomForest	77.8
RandomTree	75.6
REPTree	77.7
SimpleCart	78.2

4.1.3 Zusammenfassung und Schlussfolgerungen der Experimente

Nachdem alle Experimente abgeschlossen sind, können die drei anfänglichen Fragen erneut erörtert werden. Diesmal werden die Fragen in umgekehrter Reihenfolge betrachtet, da die Antwort auf die letzte Frage die Antwort auf die zweite Frage beeinflusst, und genau so die zweite die erste beeinflusst.

3. **Sollen Informationen über die variierten Vorgänger des zu vorhersagenden Lautes in das Modell mit einbezogen werden (die Attribute „`var-2`“ und „`var-1`“)?**
   Betrachtet man die Zahlen aus den Tabellen 4.5 und 4.7, so kann eine eindeutige Antwort gegeben werden:
   → Die Precision ist ohne die Attribute des Vorgängers und des Vorvorgängers (`var-1` und`var-2`) deutlich niedriger (um die 20%).
   → Die Instanzen der Attribute „`var-2`“ und „`var-1`“ müssen zu den Trainingsdaten hinzugenommen werden.

2. **Soll der Laut direkt vorhergesagt werden, oder zuerst eine Vorhersage über die Modifikationsklasse und im zweiten Schritt über den Laut gemacht werden?**
   Die Antwort auf diese Frage ist nicht so eindeutig wie die vorherige. Die Genauigkeit der Zwei-Schritte-Methode ist gerade mal um 0,0875% höher als die Genauigkeit der direkten Vorhersage (91.58% zu 91.5%). Bei einem dermassen geringem Unterschied muss man sich genau überlegen, ob ein höherer Programmieraufwand gerechtfertigt ist.
   Ist eine höhere Genauigkeitsrate die einzige Priorität, so sollte man zweifelsohne die Zwei-Schritte-Methode Implementieren. Im Rahmen dieser Studienarbeit wird jedoch auf den höheren Programmieraufwand verzichtet, da dadurch der zeitliche Rahmen einer Studienarbeit überschritten würde.

1. **Welcher Algorithmus liefert die beste Genauigkeit?**
   Unter Berücksichtigung der Antworten auf die Fragen drei und zwei kann der eindeutige Schluss gezogen werden, dass das Modell des Algorithmus SimpleCart mit einer Genauigkeit von 91.5% das beste Ergebnis liefert.

Wie aus den Antworten auf die Fragen drei, zwei und eins hervorgeht, fällt die Entscheidung auf die Implementierung der direkten Vorhersage der aktuellen variierten Lautes

`var0` mittels des Modells des SimpleCart Algorithmus.

4.2 Programmierung des Tools

Aus dem vorangegangenen Kapitel geht hervor, dass die Vorhersagemethode mit der höchsten Wahrscheinlichkeit die direkte Vorhersage des variierten Lautes mittels des Modells SimpleCart ist. Folglich bildet ein entsprechend trainiertes SimpleCart-Modell das Herzstück des Spontaneous Speech Tools.

In der Einleitung wurde bereits ein Ausblick auf die Funktionsweise des Spontaneous Speech Tools gegeben. Das SST dient der Erweiterung des vorhandenen IMS Aligners. Das bedeutet, dass das SST eine Eingabe vom Aligner bekommt, diese verarbeitet und sie anschließend an den Aligner wieder ausgibt. Die Abbildung 4.1, welche auch in der Einleitung gezeigt wurde, veranschaulicht diese Vorgehensweise. Die roten Pfeile zeigen den Datenfluss des Aligners bei eingebundenem SST.

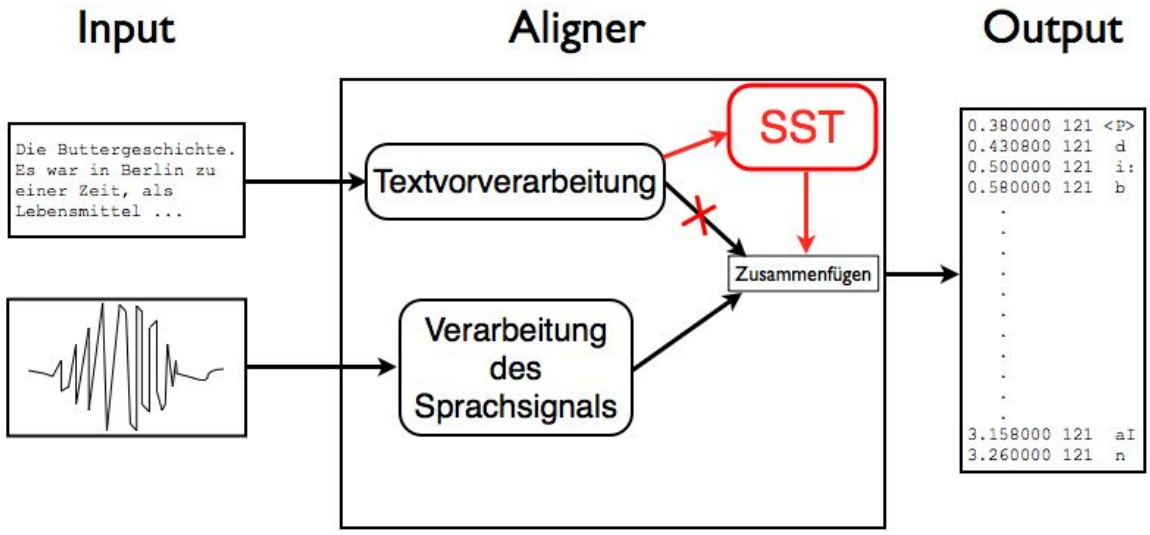

Abbildung 4.1: Architektur eines Segmentierungssystems mit eingebundenem Aussprachevarianten-Generator
(SST = Spontaneous Speech Tool)

4.2.1 Architektur des Spontaneous Speech Tools

Dem Spontaneous Speech Tool liegt eine einfache Pipeline-Struktur[2] zu Grunde. Die Architektur des SST ist durch den Datenfluss innerhalb des Tools und deren Verarbeitung festgelegt, siehe Abbildung 4.2. Die Eingabe für das SST ist die Ausgabe der Vorverarbeitung des Textes durch den Aligner. Im SST durchläuft die Eingabe drei Module:

Modul 1: Vorverarbeitung der Eingabe, bzw. Anpassung der Eingabe an die Trainingsdaten des Modells
Modul 2: Vorhersage der variierten Laute (das Herzstück des SST)
Modul 3: Anpassung der Ausgabe

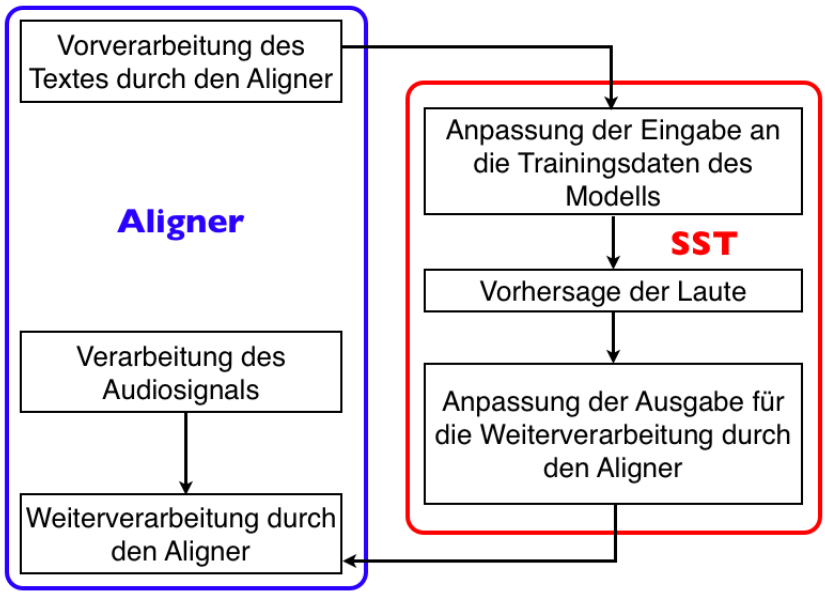

Abbildung 4.2: **Datenfluss im Aligner mit eingebundenem SST:**
blau umrandet - Datenfluss im Aligner
rot umrandet - Datenfluss im SST

Modul 1: Vorverarbeitung der Eingabe

Die Aufgabe des ersten Moduls ist die Anpassung der Ausgabe der Textvorverarbeitung des Aligners an die Trainingsdaten des Modells. Dieser Schritt wird benötigt, weil die Ausgabe der Textvorverarbeitung des Aligners SAMPA-kodiert ist, während die Trainingsdaten des Modells im SST auf Daten basieren, welche den Kieler Konventionen entsprechen.

[2]Bei einer „Pipeline-Struktur“ eines Programms werden die einzelnen Module nacheinander, „wie am Fließband“, abgearbeitet

Modul 2: Vorhersage der variierten Laute

Die Ausgabe des ersten Moduls des SST dient als Eingabe für das zweite Modul. In diesem Schritt wird für jeden Laut vorhergesagt, zu welchem variierten Laut er mutiert, oder ob er sogar unverändert bleibt.

Um eine solche Vorhersage zu treffen, wird das trainierte SimpleCart-Modell eingesetzt. Das Input für das Modell muss die gleiche Strukur aufweisen wie die Daten, auf welchen es trainiert wurde. Zur Erinnerung, die Trainingsdaten haben folgende Struktur:

can-2	can-1	can0	var-2	var-1	var0	can+1	can+2	str	plc-2	mnr-2	...
z	'i:	t	z	'i:	t	$	Q	0	alv	fric	...
t	$	Q	t	$	Q-	E	s	0	alv	plos	...
$	Q	E	$	_	E-	s	$	0	$	$	...
Q	E	s	_	_	s	$	Q	0	glt	plos	...
s	$	Q	s	$	Q	'aU	s	0	alv	fric	...
s	$	Q	$	Q	-q	'aU	s	0	alv	fric	...

Die Vor- und Vorvorgänger der kanonischen Laute sind von Anfang an bekannt. Was sich jedoch durch die sukzessive Verarbeitung der Phonemkette ändert, sind die Vor- und Vorvorgänger der variierten Laute. Das bedeutet, dass die beiden Vorgänger für jedes Phonem der Eingabezeichenkette aufs Neue bestimmt werden müssen. Aus diesem Grund werden die Werte des letzten vorhergesagten Lautes gespeichert, und in die Daten für die Vorhersage des nächsten Lautes als Vorgänger von „`var0`“ eingearbeitet.

Eben dieses Ineinandergreifen der Werte des variierten Vorgängers und Vorvorgängers lässt dem SST keine Andere Wahl, als das Modell für die Vorhersage bei jeder Änderung des Datensatzes auf Neue aufzurufen. Die Folgen können verheerend für die Nutzbarkeit eines solchen Systems sein, doch dazu mehr im Abschnitt 4.2.2.

Modul 3: Anpassung der Ausgabe

Nachdem für jeden Laut eine entsprechende Vorhersage getroffen worden ist, kommen die Daten am letzten Verarbeitungspunkt im Tool an: Dem Modul der Anpassung der Ausgabe für die Weiterverarbeitung durch den Aligner. Bevor das SST die vorhergesagte Zeichenkette ausgibt, muss die Zeichenkette wieder an das Format der Eingabe, also die SAMPA-Notation, angepasst werden.

Das SST gibt aber nicht nur die abgeänderte kanonische Zeichenkette aus. Es kombiniert die kanonische Eingabezeichenkette mit der variierten und generiert somit als Ausgabe mehrere Aussprachevarianten.

Um eine solche Ausgabe zu generieren, werden beide Phonemketten parallel Phonem für Phonem miteinander verglichen. Sind alle Phoneme gleich, wird nur eine, die kanonische, Zeichenkette ausgegeben. Stößt der Algorithmus auf den ersten Unterschied, so wird die kanonische Aussprache ausgegeben, und zusätzlich wird in der kanonischen Zeichenkette das erste zum kanonischen verschiedene Phonem eingearbeitet. Auch diese Phonemkette wird ausgegeben. Der Algorithmus sucht anschließend nach weiteren Unterschieden. Findet er den nächsten Unterschied, wird der abgeänderte Laut in die zuletzt ausgegebene Zeichenkette eingearbeitet und ausgegeben, findet er keinen Unterschied, ist die Ausgabe abgeschlossen.

4.2.2 Diskussion zur Funktionsweise des Spontaneous Speech Tools

Grundsätzlich wirkt sich die Modularisierung von komplexen (Programm-)Systemen positiv auf deren Entwicklung aus. Fehler können leichter ermittelt und der Datenfluss im System leichter verfolgt werden. Der für mich größte Vorteil der modularen Pipline-Architektur war die Möglichkeit, die Module unabhängig von einander zu entwickeln und zu testen. So waren die Module 1 und 3 lange vor Modul 2 fertig und einsatzbereit.

Zu Modul 1. Bei der Entwicklung dieses Moduls ist mir aufgefallen, dass die Eingabe für den SST, also die Ausgabe der Textvorverarbeitung des IMS Aligners, nicht alle gewünschten morphosyntaktischen Informationen enthält. Der Aligner gibt zwar eine morphologische Zerlegung aus, jedoch fehlen ihm Angaben zur Zweitbetonung oder Kompositagrenzen.

Dennoch stellt dieses Manko keine erheblichen Schwierigkeiten bei der Generierung von Aussprachevarianten dar, da Zweitbetonung sowie Kompositagrenzen für die Erzeugung phonologischer Variationen im Redefluss nicht von großer Relevanz sind (siehe Analysen in Abschnitt 3.2).

Zu Modul 2. Wie im vorherigen Abschnitt bereits angedeutet, weist dieses Modul die höchste Komplexität unter den Modulen auf. Es ist nicht möglich, eine Vorhersage über den aktuellen Laut zu treffen, wenn die Vorhersagen der beiden Vorgänger nicht bekannt sind. Dies ist der Grund dafür, dass für jeden Laut einzeln eine Vorhersage gemacht

werden muss, was heißt, dass das zu Grunde liegende Modell für jeden Laut geladen und der Vorhersageprozess aufs Neue gestartet werden muss.

Bei dieser Methode kann sich die Größe des Modells oder die Kapazität des Rechners sehr schnell zum Problem entwickeln. Je größer das Modell, desto langsamer findet der Vorhersageprozess statt. Bereits bei einer Größe des Modells von circa 85 MB kann die Vorhersage für nur einen Laut bis zu 50 Sekunden dauern. Rechnet man diesen Wert auf einen Satz mit 10 Wörtern und durchschnittlich 3-4 Phonemen pro Wort hoch, so muss der Nutzer der Programms 30 Minuten auf die Ergebnisse warten.

Die Nutzbarkeit eines Tools, welches ein trainiertes Modell so häufig aufrufen muss, hängt somit stark von der Größe des Modells ab, sowie auch von der Größe des Arbeitsspeichers des Rechners. Aus diesem Grund musste auf das Modell mit der höchsten Genauigkeit verzichtet werden. Das SimpleCart-Modell ist bis zu 55% langsamer als ein schwächeres, aber dennoch ausreichend gutes Modell. Das endgültige Modell für die Vorhersage von geänderten Lauten ist das Modell des WEKA-Verfahrens J48.

Zu Modul 3. Wie im Abschnitt „Modul 3: Anpassung der Ausgabe“ bereits erklärt, werden die kanonische und die vorhergesagte variierte Aussprache sukzessive mit einander verglichen und bei gefundener Abweichung von der kanonischen Aussprache wird eine neue Phonemkette generiert, in welcher die Abweichung aufgenommen worden ist. Siehe dazu das Beispiel unten.

Beispiel: Aussprachevarianten des Wortes „zweitägigen“:

kanonische Lautform:	`tsvaItEgIg@n`	
Vom SST generierte Varianten:	1. `tsvaItEgIg@n`	(kanonisch)
	2. `tsvaItEgIgn`	(Tilgung von `/@/`)
	3. `tsvaItEgIgN`	(Substitution von `/n/` durch `/N/`)
Vom Aligner mit angeschlossenem SST ausgewählte Variante:	`tsvaItEgIgN`	

Ein Vorteil dieses Verfahrens ist, dass zu der vorhergesagten Variante auch zusätzliche Varianten generiert werden. Der Aligner entscheidet später, welche dieser Varianten zu dem tatsächlich Geäußerten am besten passt.

Eine andere Möglichkeit, mehrere Varianten zwischen der kanonischen und der vorhergesagten Aussprache zu erzeugen, wäre die Generierung aller Kombinationen dieser beiden Aussprachen. Bei diesem Vorgehen hätte man im obigen Beispiel zu den drei vorgestellten Varianten eine vierte Variante, nämlich `/tsvaItEgIg@N/`. Die Generierung einer solchen

Variante im natürlichen Redefluss ist jedoch eher unwahrscheinlich. Phonologische Prozesse im Redefluss beeinflussen sich gegenseitig. Das heißt, kam es zu einem Zeitpunkt in einer Äußerung zu einer Modifikation der Aussprache durch einen phonologischen Prozess, so darf diese Modifikation zu einem späteren Zeitpunkt nicht ignoriert werden. Im Beispiel vorhin heißt das, dass die Tilgung von dem Laut `/@/` in der Silbe `/g@n/` als Folge die Glottalisierung von `/n/` hat, da nach der Tilgung von `/@/` die Laute `/g/` und `/n/` direkt aufeinander folgen.

In diesem Modul werden auch alle glottalen Verschlüsse vor der Weitergabe an den Aligner herausgefiltert. Grund dafür ist, dass der IMS Aligner ohne glottale Verschlusslaute trainiert worden ist, und er für diese keine Modelle vorliegen hat.

4.3 Anbindung an den IMS Aligner

Durch die ursprüngliche Konstruktion des IMS Aligners wurden die notwendigen Bedingungen für die Anbindung weiterer Komponenten bereits geschaffen. Wie auch die Architektur des SST ist auch die Struktur des IMS Aligners modular. Ein solcher Aufbau des Systems ermöglicht es, weitere Module an den IMS Aligner anzuknüpfen, oder diese durch andere Module zu ersetzen. Entsprechende Schnittstellen zum Anschluss neuer Module sind ebenfalls im Aligner bereits vorhanden. Die Anbindung des Spontaneous Speech Tools an den IMS Aligner ist also lediglich ein Anschließen des SST an die entsprechenden Schnittstellen.

Insgesamt werden für die Anbindung des SST an den Aligner zwei Module des Systems geändert. Das erste Modul ist der so genannte *Tokenizer*, welches eine Äußerung in ihre Grundeinheiten, sprich Wörter, zerlegt. Der im Aligner vorhandene Tokenizer wird durch den Tokenizer des Systems *Festival* ersetzt.

Das zweite Eingreifen in das Modulsystem des IMS Aligners ist die eigentliche Einbindung des SST-Moduls. Dieses Modul wird nach der Erstellung der Wortliste einer Äußerung (Tokeniseirung) und vor der Konstruktion des Phonemnetzes (siehe Abschnitt 2.2.1, Architektur des IMS Aligners) angeschlossen.

Die Manipulation des Modulsystems wird durch die Definition zweier Umgebungsvariablen kontrolliert: Die Variablen „ORTHOGRAPHICINPUT2WORDLIST“ und „WORDLIST2PHONEMES“. Die genaue Parameterbelegung dieser und fünf weiterer Variablen ist in Tabelle 4.13 aufgelistet. Der technische Hintergrund der Umgebungsvariablen und ihrer Belegungen werden hier nicht erläutert, siehe dazu [URL: IMS Aligner] . Sind alle sieben Parameter wie in Tabelle 4.13 definiert worden, kann der Aligner wie gewohnt mit dem Befehl „`Alignphones` *file*`.wav`“, bzw. „`Alignwords` *file*`.wav`“, gestartet werden.

Bitte beachten: Die üblichen Bedingungen, um mit dem IMS Aligner zu arbeiten, müssen auch im Falle des angeschlossenen SST erfüllt sein. Dazu gehören das Vorhandensein der Transkription zum Sprachsignal, die Latin-1 Kodierung der Textdatei, sowie die Definition der Standardparameter (siehe Tabelle 4.13).

Anbindung des SST:

Umgebungsvariable	Belegung
ORTHOGRAPHICINPUT2WORDLIST	SST-v3.tokenize
WORDLIST2PHONEMES	SST-v6.predict

Standardparameter, unabhängig vom SST:

Umgebungsvariable	Belegung
ALIGNERBIN	/usr/local/Aligner/bin
ALIGNERHOME	/usr/local/Aligner
PATH	${PATH}:/usr/local/htk/bin.linux
ALANG	deu
LC_ALL	de_DE

Tabelle 4.13: **Belegung der Umgebungsvariablen**, um den IMS Aligner mit angeschlossenem SST bedienen zu können. Die ersten zwei Belegungen schliessen das SST an den Aligner an, die restlichen fünf werden benötigt, um den Aligner zu konfigurieren.

Unter den Linux Fedora Systemen der Rechner des Institutes der maschinellen Sprachverarbeitung der Universität Stuttgart werden Umgebungsvariablen auf einer tc-Shell durch den Befehl „`setenv` *Variable Belegung*“ umgesetzt.

5 Evaluierung

5.1 Ergebnisse des Testens auf den Testdaten

Wie bereits des Öfteren erwähnt, wurden die vorhandenen Daten in Trainings- und Testdaten aufgeteilt. Zwei Drittel der Daten dienten als Wissensquelle für das Trainieren des Modells, das restliche Drittel wird nun in diesem Kapitel untersucht. Das letzte Drittel der Daten ergibt 597 Labelfiles, bzw. 597 entsprechende Audiodateien. Die Beurteilung der Ergebnisse des Aligners mit und ohne SST wird anhand dieser 597 Dateien festgelegt.

Um die Ergebnisse des Aligners mit eingebundenem Spontaneous Speech Tool zu evaluieren, wurde der Aligner einmal mit, und einmal ohne SST auf die Sprachsignale angewandt. So gelangt man zu drei Arten von Labeldateien pro Audiosignal:

1. vom Aligner ohne SST generiertes Labelfile („Alignerlabelfile“)
2. vom Aligner mit SST generiertes Labelfile („SST-Labelfile“)
3. manuell erstelltes Labelfile aus dem Kiel Korpus („Kiellabelfile“)

Als nächstes werden die erzeugten Labeldateien mit den Kiellabelfiles verglichen. In Tabelle 5.1 ist eine erste Übersicht der Ergebnisse dargestellt.

	Lautänderungen in den Kiellabelfiles	davon korrekt in Alignerlabels	davon korrekt in SST-Labels
Labels gesamt	54.020	46.644	47.973
davon Tilgungen	4.833	466	1.919
davon Einfügungen	414	42	46
davon Substitutionen	1.589	24	285

Tabelle 5.1: Übersicht der korrekten Labels der vom Aligner erzeugten Labeldateien.
„Alignerlabel“= vom Aligner ohne SST erzeugtes Labelfile
„SST-Label“= vom Aligner mit angeschlossenem SST erzeugtes Labelfile

Hier ist zu beachten, dass der Aligner ohne das Spontaneous Speech Tool zwar keine Einfügungen, Substitutionen oder Einfügungen vorhersagen kann, in der Tabelle dennoch

Zahlen bei den genannten Änderungen stehen. Diese Auffälligkeit lässt darauf schließen, dass sich die kanonische Lautform des Aligners von der in den Kiellabeldateien unterscheidet.

Als nächstes werden die Werte der Tabelle 5.1 zueinander in Verhältnis gesetzt und daraus die prozentuale Verbesserung der korrekten Labels errechnet, siehe Tabelle 5.2.

	Anteil korrekter Labels der Alignerlabelfiles	Anteil korrekter Labels der SST-Labelfiles	Verbesseung
gesamt	86,34%	88,8%	2,85%
Tilgungen	9,64%	39,7%	311,88%
Einfügeungen	10,14%	11,11%	9,56%
Substitutionen	1,51%	17,93%	1.087,41%

Tabelle 5.2: Anteil der Labels der Aligner- und SST-Labels, welche mit den Kiellabels übereinstimmen.

Die Werte der Tabelle 5.2 sind beeindruckend. Auf der einen Seite hat sich die Übereinstimmung der Kiel- und vom Aligner erzeugten Labels nach dem Einbinden des SST um nur 2,85% verbessert. Betrachtet man andererseits die Übereinstimmung der Labels welche Änderungen unterliegen, so kann eine außerordentlich hohe Verbesserung beobachtet werden.

Aus diesen Zahlen kann die Schlussfolgerung gezogen werden, dass das Spontaneous Speech Tool außer korrekten Vorhersagen auch einige falsche Vorhersagen trifft. In Tabelle 5.3 können die falschen sowie korrekte Vorhersagen und ihre prozentuale Verteilung eingesehen werden.

Änderung	Anzahl korrekter Vorhersagen	Anzahl falscher Vorhersagen	Anteil falscher Vorhersagen
Tilgungen	1.919	410	17,6%
Substitutionen	285	278	49.37%
Einfügungen	46	0	0%

Tabelle 5.3: Übersicht aller vom SST getroffenen Vorhersagen und der Anteil der falschen Vorhersagen.

Genau diese Zahlen sind ausschlaggebend für die Beurteilung des SST. Das SST macht zu viele falsche Vorhersagen. In Tabelle 5.4 sind die häufigsten Fehler bei der Vorhersage von Tilgung und Substitution aufgelistet. Vor allem die Substitutionen werden mit einer hohen Fehlerrate vorhergesagt.

Fehlerhafte Tilgungen		Fehlerhafte Substitutionen	
1. /E:/ und /m/	in „ähm“	1. /a/→/a:/	in „zwanzig“
2. /e:/	in „der“ und „Herr“	2. /b/→/p/	in „glaub’ Ich“
3. /C/	in „zwanzigster“	3. /C/→/k/	in „zwanzigster“

Tabelle 5.4: Übersicht der Häufigsten Vorhersagefehler des Spontaneous Speech Tools

Zur Erinnerung: Aus Effizienzgründen wurde auf das Modell mit der höchsten Genauigkeit verzichtet. Es wurde ein etwas schwächeres, jedoch deutlich schnelleres Verfahren verwendet, das WEKA-Verfahren J48. Diese Abweichung von dem ursprünglichen Ansatz wie in Abschnitt 2.5.3 beschrieben kann ein Grund für die relativ schlechte Vorhersage sein.

5.2 Schlussfolgerungen

Auch wenn die Ergebnisse des SST nicht die selbe Verbesserung der Fehlerrate wie der statistische Ansatz von [Sloboda et al., 1996] aufweisen kann (bei welchem die Fehlerrate des Spracherkenners JANUS um bis zu 6,3% gesteigert wurde), so ist eine deutliche Verbesserung im Vergleich zu dem auf phonologischen Regeln basierten Verfahren von [Kemp, 1996] zu erkennen (bei [Kemp, 1996] Verbesserung um 1.5%).

Aus diesen Zahlen kann der Schluss gezogen werden, dass der zugrunde liegende Theorieansatz, „Statistischer Ansatz auf phonologischer Ebene“ (siehe Abschnitt 2.5.3), erfolgversprechend ist. Zwar sind nicht alle Vorhersagen des statistischen Modells korrekt, aber die Gründe dafür liegen vielmehr in der praktischen Umsetzung.

Der erste Stolperstein in der Umsetzung des neuen Ansatzes ist die gezwungene Einschränkung auf die phonologische Ebene der Äußerungen. Das Kiel Korpus der Spontansprache bietet keine Information über die Silbenstruktur und kaum Aufschluss über die morphologische Beschaffenheit der Äußerungen.

Beide diese Aspekte des Geäußerten spielen in der Theorie von Aussprachevarianten eine bedeutende Rolle. Vor allem in Abschnitt 3.2, in welchem das Kiel Korpus der Spontansprache das erste Mal statistisch untersucht wurde, wurde deutlich, dass eine reine phonologische Struktur nicht ausreicht. In Abschnitt 3.2 konnte nachgewiesen werden, dass beispielsweise die meisten Tilgungen in unbetonten Endsilben stattfinden. Folglich könnte man spekulieren, dass hauptsächlich Suffixmorpheme Tilgungen unterliegen. Ebenso konnte beobachtet werden, dass Einfügungen niemals *in* einer Silbe, sondern *zwischen* Silben produziert werden. Des Weiteren hat die Untersuchung des Korpus eine

Tendenz zur Beständigkeit der Stämme der Wörter aufgedeckt. Einen Zusammenhang von Substitution und Morphologie konnte nicht festgestellt werden, jedoch bietet das Korpus keine Möglichkeit, nach solchen Abhängigkeiten zu suchen.

Da ich mir der fehlenden morphologischen Analyse bewusst war, habe ich versucht, durch die Hinzunahme von Umgebungslauten zu kompensieren. Diese Vorgehensweise war insofern erfolgreich, dass Laute tatsächlich nur in Endsilben geändert wurden.

Der nächste Umsetzungsfehler ist das Zurückgreifen auf ein anderes statistisches Modell als die Theorie es verlangt. Da der Datensatz sehr viele Attribute beinhaltet (20 Spalten in der Merkmalstabelle, siehe Tabelle 3.3, Abschnitt 3.1), die Merkmale eine große Vielfalt an Werten annehmen können (das Merkmal des variierten Lautes „`var0`“ kann 146 verschiedene Werte annehmen), und die Verteilung vieler Änderungen gleichmäßig ist, werden die statistischen Modelle außerordentlich groß. Wie in Abschnitt 4.2.2 bereits diskutiert, kann ein zu groß geratenes Modell die Verwendung verhindern. Dauert die Generierung von Aussprachevarianten zu lange und ist dabei die Fehlerrate nicht viel geringer als ohne Vorhersage von Aussprachevarianten, wird das Tool kaum verwendet werden.

6 Zusammenfassung und Ausblick

Zusammenfassend kann die Verwendung des Spontaneous Speech Tools als erfolgreich betrachtet werden. Der Aligner mit eingebundenem SST erzeugt um 2,85% mehr korrekte Labels als ohne SST. Somit ist der präsentierte Ansatz, der „statistische Ansatz auf phonologischer Ebene“, durchaus haltbar.

Der Ansatz kann in vier Phasen aufgeteilt werden. Als Erstes wird ein geeigneter Korpus gewählt, und damit die Grundeinheiten des Datensatzes (Phoneme, Silben oder Morpheme) festgelegt. In der nächsten Phase wird der (Trainings- und Test-) Datensatz ausgearbeitet. Um die Vorhersage von Einheiten zu unterstützen, wird empfohlen neben der kanonischen und variierten Form der Einheiten auch weitere Merkmale der Einheiten, wie Betonung, Position im Wort, etc. in den Datensatz mit aufzunehmen, sowie den Datensatz wortübergreifend zu gestallten. In der dritten Phase wird ein statistisches Modell auf dem Datensatz trainiert und getestet. Hier ist es sinnvoll mehrere Verfahren zu testen und anhand der Precision-Rate ein Verfahren auszuwählen. In der vierten und letzten Phase wird das Modell, als Herzstück eines kleineren Programms, in dieser Arbeit das Spontaneous Speech Tool, an das vorhandene Annotations- und Segmentierungssystem angeschlossen.

Die Vorteile dieses Ansatzes liegen hauptsächlich in der Variabilität der Grundeinheiten des Datensatzes, d.h. der Möglichkeit den Ansatz auf Phonem-, Silben- oder Morphemebene zu gestalten. Des Weiteren hat sich die wortübergreifende Struktur des Datensatzes als sinnvoll erwiesen, da das nachfolgende Wort, oder möglicherweise das Satzende, die Aussprache des aktuellen Wortes nachweislich beeinflusst.

Um eine gute Verbesserungsrate zu erzielen ist vor allem die Struktur des zugrunde liegenden Korpus entscheidend, da sie die Ebene der Einheiten und deren Informationsgehalt festlegt. In dieser Arbeit wurde das Kiel Korpus der Spontansprache gewählt, womit die Einheiten unvermeidbar auf Phonemebene eingeschränkt wurden.

Wie in Abschnitt 5.2 bereits diskutiert, stellt die Silben- oder Morphemebenenvariante des Ansatzes eine sinnvolle Weiterführung der Untersuchungen dieser Arbeit dar. Die Hauptursache für die Anregung weitere Untersuchungen anzustellen ist die Feststellung,

dass spontansprachliche Änderungen nicht ausschließlich an Phonemen festgemacht werden können.

Vielmehr haben die Ergebnisse dieser Arbeit die Spekulation bestärkt, dass der Wortstamm bei Aussprachevarianten deutlich seltener Änderungen unterliegt als (wortfinale) Affixe. Um diese Spekulationen zu untersuchen, sollte der Ansatz auf einem Korpus getestet werden, welcher die gewünschte Silben- oder Morphemstruktur aufweist. Dabei sollte der Algorithmus des Ansatzes auf mindestens zwei Ebenen durchgeführt werden (z.B. phonologische und morphologische Ebene), um einen sinnvollen Vergleich der Strukturebenen zu erhalten.

Literaturverzeichnis

[Carstensen et al., 2010, Seite 493-494] K.-U. Carstensen, C. Ebert, C. Ebert, S. J. Jekat, R. Klabunde, H. Langer, 2010,
Computerlinguistik und Sprachtechnologie: Eine Einführung, 3. Auflage;

[Baayen et al., 1995] H. Baayen, R. Piepenbrock, L. Gulikers, L., 1995,
The CELEX lexical database—Release 2, CD-ROM, Centre for Lexical Information, Max Planck Institute for Psycholinguistics, Nijmegen;

[Baayen, 2001] H. Baayen 2001,
Word Frequency Distributions, Kluwer Academic Publisher;

[Demuynck et al., 2002] K. Demuynck, T. Laureys, 2002
A Comparison of Different Approaches to Automatic Speech Segmentation, International Conference Text, Speech and Dialouge, s. 277-284, Brno, Czech Republic;

[HTK, 1994] Entropic Research Laoratory, Inc, 1994,
HTK - Hidden Markov Modell Toolkit, Washington DC;

[Kemp, 1996] T. Kemp, 1996,
Regelbasiert generierte Aussprachevarianten für Spontansprache, Interactive System Labs, ILKD, Universität Karlsruhe;

[Kipp, 1997] A. Kipp, M.-B. Wesenick, F. Schiel 1997,
Pronunciation Modelling Applied to Automatic Segmentation of Spontaneous Speech, IPSK, Universität München;

[Kohler, 1992] K. Kohler, 1992,
Kieler Arbeiten zu den PHONDAT-Projekten 1989-1992, Arbeitsberichte Nr. 26, Institut für Phonetik und digitale Sprachverarbeitung, Universität Kiel;

[Kohler, 1995] K. Kohler, 1995
AIPUK, Arbeitsberichte Nr. 29, Institut für Phonetik und digitale Sprachverarbeitung, Universität Kiel;

[Lindblom, 1990] B. Lindblom, 1990,
Explaining phonetic vatiation: A sketch of the H & H theory, s. 403-439;

[Rapp, 1998] S. Rapp, 1998
Automatisierte Erstellung von Korpora für die Prosodieforschung, Dissertation Universität Stuttgart, Arbeiten des Instituts für Maschinelle Sprachverarbeitung;

[Rapp, 1995] S. Rapp, 1995
Automatic phonemic transcription and linguistic annotation from known text with Hidden Markov Modells, Universität Stuttgart;

[Sloboda et al., 1996] T. Sloboda, A. Waibel, 1996,
Dictionary Learning for Spontaneous Speech Recognition, Interactive Systems Laboratories, Universität Karlsruhe und Carnegie Mellon University, Pittsburgh, USA;

[Wesenick, 1996] M.-B. Wesenick, 1996,
Automatic generation of german pronunciation variants, Institut für Phonetik und Sprachliche Kommunikation, Ludwig-Maximilians-Universität München;

[Witten et al., 2005] I. H. Witten, E. Frank, 2005,
Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), The University of Waikato;

[URL: IMS] http://www.ims.uni-stuttgart.de/phonetik/projekte/IDS/antrag2.html/
letzter Aufruf der Siete: 17.09.2010, 17:13;

[URL: IMS Aligner] http://www.ims.uni-stuttgart.de/phonetik/helps/aligner.html/
letzter Aufruf der Seite: 26.04.2011, 12:17;

[URL: SAMPA] http://www.phon.ucl.ac.uk/home/sampa/index.html/
letzter Aufruf der Seite: 24.08.2010, 20:23;

[URL: WEKA] http://www.cs.waikato.ac.nz/∼ml/weka/index.html/
letzter Aufruf der Seite: 25.01.2011, 11:39;