Graz University of Technology

Phonetic Similarity Matching of Non-Literal Transcripts in Automatic Speech Recognition

Doctoral Thesis

at

Graz University of Technology

submitted by

Stefan Petrik

Signal Processing and Speech Communication Laboratory,
Inffeldgasse 12, A-8010 Graz, Austria

December 2009

First Advisor: Univ.-Prof. Dr. Gernot Kubin, Graz University of Technology
Second Advisor: ao. Univ.-Prof. Dr. Harald Trost, Medical University of Vienna

Abstract

Large vocabulary continuous speech recognition (LVCSR) systems require large amounts of labelled audio data for training. While such literal transcriptions of audio recordings, i.e., highly accurate textual reproductions of the utterances are expensive and therefore only available in limited amounts, non-literal field data from commercial automatic dictation systems can be collected on large scale but with quality limitations. Automatic draft transcriptions from the dictation system contain misrecognitions and the manual corrections of the draft transcriptions produced by professional transcriptionists have been reformulated to comply with stylistic guidelines.

In this work, phonetic similarity matching is utilised to bridge this gap between literal and non-literal text resources such that large amounts of non-literal transcripts can be employed for the improvement of LVCSR systems. For the first time, a detailed analysis of the deviations between manual reference transcripts, automatically recognised transcripts, and final corrected documents of a medical transcription environment on orthographic and phonetic level is given. Based on these insights, a novel method for the alignment of recognised transcripts and final corrected documents on multiple levels of segmentation was developed. The alignment is calculated based on the similarity of two phone strings determined with a stochastic string edit distance function trained on task-specific data.

The proposed methods are applied for solving two exemplary application-driven problems. First, quasi-literal transcripts of medical dictations are reconstructed out of the non-literal automatically recognised and the final, corrected medical reports. Semantic and phonetic similarity measurements are defined for classifying aligned text chunks as either recognition errors or reformulations introduced by the medical transcriptionist. Language model retraining with a corpus of 50 million reconstructed words resulted in a relative word error rate reduction of 7.8% for a commercial medical transcription system. Second, speaker-specific pronunciation models for non-native speakers are generated from small amounts of available adaptation data. Phonetic similarity matching is utilised for measuring lexical confusability and the accuracy gain of a proposed pronunciation variant such that both effects are balanced for a given lexicon. Recognition tests with speaker-specifically adapted lexica resulted in an average relative word error rate reduction of 1% per speaker for the same commercial medical dictation system.

Kurzfassung

Spracherkennungssysteme mit großem Vokabular werden mit großen Mengen an annotierten Audiodaten trainiert. Solch genaue Transkriptionen der Audioaufnahmen, d.h. exakte Reproduktionen der Äußerungen sind teuer und daher nur in geringen Mengen verfügbar, während Felddaten von kommerziellen Diktiersystemen laufend in großem Stil, allerdings mit Qualitätseinschränkungen gesammelt werden können. Automatisch erkannte Transkripte des Diktiersystems beinhalten Erkennungsfehler und von professionellen Transkribenten händisch erstellte Dokumente beinhalten Umformulierungen um formalen Ansprüchen zu genügen.

In dieser Arbeit wird phonetische Ähnlichkeitsmessung dazu verwendet um diese Lücke zwischen genauen und näherungsweisen Transkriptionen zu überbrücken, damit große Felddatensammlungen zur Verbesserung von automatischen Diktiersystemen verwendet werden können. Dazu wurde erstmals eine detaillierte Analyse der Unterschiede zwischen exakten manuellen Referenztranskriptionen, automatisch erkannten Transkripten und professionell verfassten Befunden eines medizinischen Diktiersystems erstellt. Auf Basis dieser Erkenntnisse wurde eine neue Methode zur parallelen Ausrichtung von automatisch erkanntem und formatiertem Befund auf mehreren Segmentierungsebenen entwickelt. Die Ausrichtung wird berechnet indem die Ähnlichkeit zweier Phonemsequenzen durch eine stochastische Edit Distanz bestimmt wird, die mit anwendungsspezifischen Daten trainiert wurde.

Mit den vorgeschlagenen Methoden werden zwei Problemlösungen beschrieben. Zuerst wird ein quasi-exaktes Transkript eines medizinischen Diktats aus einem automatisch erkannten und einem manuell verschrifteten medizinischen Befund rekonstruiert. Mit Hilfe von semantischen und phonetischen Ähnlichkeitsmaßen werden dabei abweichende Textteile entweder als Erkennungsfehler oder als Umformulierung klassifiziert. Ein aus 50 mio. rekonstruierten Worten trainiertes Sprachmodell führte in Erkennungstests zu einer relativen Wortfehlerratenreduktion von 7.8% für ein kommerzielles medizinisches Diktiersystem. In einer zweiten Anwendung wurden sprecherspezifische Aussprachemodelle für nicht-muttersprachliche Sprecher aus kleinen Mengen an Adaptionsdaten erstellt. Mit phonetischer Ähnlichkeitsmessung wurde die Vertauschbarkeit innerhalb des Lexikons und der Gewinn an Genauigkeit bestimmt, den eine vorgeschlagenen Aussprachevariante bewirkt. Erkennungstests mit sprecherspezifisch adaptierten Lexika führten im Schnitt zu einer relativen Wortfehlerratenreduktion von 1% pro Sprecher für dasselbe kommerzielle medizinische Diktiersystem.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

date

(signature)

Acknowledgements

Although there is only one name on the title page, there are many others who contributed to this work. Their contributions shall be acknowledged at this position.

First I would like to express my sincere gratitude to my supervisor Gernot Kubin who gave me the chance of conducting my research and supported me over the years with advice and great patience. My thanks also go to my second supervisor Harald Trost who accompanied this work already from the beginning by giving ideas and pushing it into the right direction. Besides my official supervisors, I especially would like to thank my co-supervisor Franz Pernkopf for his continuous support, the many discussions we had, and the experiments we conducted together. In the same way, I would also like to thank Rudolf Muhr and his students for their efforts in transcribing the MEDTRANS corpus, and him personally for persistently renewing my motivation for conducting research in natural language processing.

I am particularly indebted to my scientific partners at OFAI, Vienna, Martin Huber, Jeremy Jancsary, Alexandra Klein, and Johannes Matiasek for the excellent cooperation in the development of the transcript alignment and reconstruction tools. Without their efforts this work would not have been possible. The same holds for my industrial partners at Nuance Communications Austria, Vienna (formerly Philips Speech Recognition Systems). In particular, I would like to thank Christina Drexel for providing tools and her constructive cooperation, Leo Fessler for the recognition experiments with retrained language models, Walter Müller for discussions and critically reviewing my ideas, and Zsolt Saffer for the many discussions and ideas in creating the speaker-specific pronunciation models.

In addition, many thanks go to my colleagues at the Signal Processing and Speech Communication Laboratory who accompanied me over the years, gave me a good time, and became friends to me. Thanks also to all my friends who pulled me through the tough times. And last, but by no means least very warm thanks to my family: without their support and understanding, this thesis would not have been possible.

This research has been carried out in the context of the projects SPARC and COAST (http://www.coast.at). I gratefully acknowledge funding by the Austrian FIT-IT and K-net programs, ZIT Zentrum für Innovation und Technologie, Vienna, Steirische Wirtschaftsförderungsgesellschaft mbH and Land Steiermark.

Stefan Petrik
Graz, Austria, December 2009

Contents

1 Introduction **1**

1.1 Large vocabulary continuous speech recognition . . . 1

1.2 Phonetic and phonologic knowledge in speech technology . . . 2

1.3 Speech and language resources for system development . . . 3

1.4 Outline of the thesis . . . 4

1.4.1 Problem statement and main hypothesis . . . 4

1.4.2 Scientific contributions . . . 5

1.4.3 Document structure . . . 7

I Fundamentals of Non-Literal Transcript Matching **8**

2 Literal and Non-Literal Text Resources in an ASR-Supported Medical Transcription Environment **9**

2.1 Medical transcription . . . 10

2.1.1 Transcription workflow . . . 10

2.1.2 ASR in medical transcription . . . 10

2.1.3 Text resources generated in the course of the workflow . . . 11

2.2 Literal text resources . . . 12

2.2.1 Manual orthographic transcriptions . . . 12

2.2.2 Manual phonetic transcriptions . . . 14

2.3 Non-literal text resources . . . 18

2.3.1 Recognised texts . . . 18

2.3.2 Final documents . . . 20

2.4 A paragraph from a sample medical report . . . 21

2.5 Medical Dictation text corpora . . . 23

2.5.1 MEDTRANS: Phonetically transcribed medical dictation corpus . . . 23

2.5.2 MEDALIGN: Multi-alignment medical dictation corpus . . . 28

2.6 Conclusion . . . 31

3 Similarity, Dissimilarity, and Confusability in ASR 32
3.1 General definition and notation . . . 33
3.2 State-based approach . . . 34
3.2.1 Continuous range . . . 35
3.2.2 Discrete range . . . 36
3.3 Symbol-based approach . . . 36
3.3.1 Deterministic measures . . . 38
3.3.2 Stochastic measures . . . 40
3.4 Evaluation: Pronunciation classification . . . 43
3.4.1 Experimental setup . . . 43
3.4.2 Data sets . . . 44
3.4.3 Training and decoding . . . 44
3.4.4 Quantitative results . . . 45
3.4.5 Qualitative results . . . 49
3.5 Suggestions for extensions . . . 50
3.5.1 The syllable as structuring unit . . . 50
3.5.2 Parameter-tying schemes . . . 51
3.6 Conclusion . . . 51

II Application-Driven Solutions 53

4 Automatic Reconstruction of Medical Dictations 54
4.1 SPARC approach . . . 55
4.2 Data description . . . 58
4.3 Text alignment . . . 59
4.4 Similarity measures . . . 60
4.4.1 Semantic similarity . . . 60
4.4.2 Phonetic similarity . . . 61
4.4.3 Combined similarity measurement . . . 65
4.5 Text reconstruction . . . 66
4.5.1 Rule engine . . . 67
4.5.2 Rule definitions . . . 68
4.6 Experiments . . . 70
4.6.1 Reconstruction quality . . . 70
4.6.2 Automatic speech recognition . . . 75
4.7 Conclusion . . . 78

5 Speaker-Specific Selective Pronunciation Modelling 80

5.1 Pronunciation modelling at the lexical level . . . 81

5.1.1 General definition . . . 81

5.1.2 Motivation . . . 81

5.1.3 State-of-the-art in lexical pronunciation modelling . . . 82

5.1.4 Alternative non-native speech adaptation approaches . . . 84

5.2 A model for minimisation of lexical confusability . . . 85

5.2.1 Learning speaker-specific phone confusions from data . . . 87

5.2.2 Selection of candidate words . . . 89

5.2.3 Pronunciation variants from Weighted Finite State Transducers . . . 90

5.2.4 Measuring confusability and the gain in modelling accuracy . . . 92

5.2.5 Optimisation strategies . . . 94

5.3 Experiments . . . 94

5.3.1 Experimental setup . . . 95

5.3.2 Comparison of optimisation strategies . . . 95

5.3.3 Comparison of candidate word sets . . . 96

5.3.4 Comparison of resources for speaker-specific confusions . . . 97

5.3.5 Adapted versus unadapted acoustic models . . . 98

5.4 Conclusion . . . 99

III Further Exploitation and Outlook 101

6 Outlook and Conclusion 102

6.1 Outlook . . . 102

6.1.1 Further corpus work . . . 102

6.1.2 Phonetic algorithm extensions . . . 103

6.1.3 More application-driven solutions . . . 103

6.2 Conclusion . . . 104

6.2.1 A critical discussion of the initial hypothesis . . . 104

6.2.2 Remarks on the generalisability of the approach . . . 105

IV Appendix 107

A Definitions 108

B MEDTRANS Corpus Details 112

Bibliography 117

List of Figures

2.1 Traditional workflow in medical transcription. 10

2.2 ASR-supported workflow in medical transcription. 11

2.3 Overview of documents created during the transcription workflow. 12

2.4 Number of pronunciation variants per word observed in the MEDTRANS corpus, sorted by word frequency. 15

2.5 Duration boxplots for 6 of the most frequent pronunciation variants (bottom to top) of the word `patient` from the MEDTRANS corpus. 17

2.6 A paragraph from a sample medical report as it is represented in the various medical transcription text types. 22

2.7 Multialignment process between manual transcriptions (SPK), draft transcripts (REC), and final corrected medical reports (WRI). 29

2.8 Deviations in tokenisation observed in multi-alignments induced by recognition errors. 30

2.9 Alignment mismatches in the multi-alignments induced by content-insensitive alignment. 30

3.1 Schematic illustration of the state-based similarity approach. 34

3.2 Schematic illustration of the symbol-based similarity approach 37

3.3 F1 performance versus training set size for SWBTRANS data set. 48

3.4 Number of phones per phone input sequence for syllable-based and word-based input sequences in the MEDTRANS training data set. 51

4.1 SPARC architecture. 57

4.2 A sample text passage with mismatch regions highlighted in boxes, aligned with standard Levenshtein alignment (left) and the advanced multi-alignment computed by SPARC (right). 60

4.3 Block scheme of phonetic similarity function: automatic phonetic transcription (APT), automatic syllabification, trainable string edit distance measure (SED), and Levenshtein measure (LevD). 62

4.4 A sample alignment containing two mismatch regions. 64

4.5 Learned probability distribution for edit operations z_i after 3 EM iterations. . 65

4.6 Reconstruction of a text passage with two mismatch regions. 66

4.7 Excerpt of aligned input sequences with sliding rule window. 67

4.8 A sample alignment containing two mismatch regions (dashed boxes), together with reconstruction rule results. 69

4.9 Recall/Precision diagram derived from the all$_{sem}$ system by varying the semantic similarity threshold for high, medium, and low WER texts. 72

4.10 Recall/Precision diagram derived from the all$_{pho}$ system by varying the phonetic similarity threshold for high, medium, and low WER texts. 73

4.11 WER in [%] for increasing re-training text corpus size. 78

5.1 CART trees of phone `/d/` for speakers a) #050678 b) #047722 c) #008107 from data set MEDALIGN-NNS102. 84

5.2 Schematic illustration of the lexical confusability optimisation approach. . . . 86

5.3 Reference automatic phonetic transcription generated from reference orthographic transcriptions vs. free automatic phonetic transcription. 88

5.4 Phone confusion matrices for speakers a) #050678 b) #047722 c) #008107 from data set MEDALIGN-NNS102 with ENROL alignment. 89

5.5 FST representation of the canonical pronunciation `/{ k s e s/` of the word `access`. 91

5.6 FST lexicon P comprising the words `access`, `axis`, `actual`, and `apple`. . . . 91

5.7 WFST confusion matrix C for substitutions and deletions of phones `/@/` and `/{/` with corresponding weights. 91

5.8 WFST composition of word `access` with confusion matrix C. 92

5.9 WFST composition of word `access` with confusion matrix C and lexicon P. . 93

5.10 Relative WER reduction per speaker in % (sorted in descending order) for WFST and CART framework with unadapted and adapted acoustic models. . 100

List of Tables

2.1 Speaker errors in manual orthographic transcriptions of the MEDALIGN-PROFILE data set, sorted by counts. 13

2.2 Pronunciation variants with syllabic structure for the words `milligrams` and `regular` from the MEDTRANS corpus. 16

2.3 Modifications observed in final medical reports of the MEDALIGN-PROFILE data set, sorted by counts. 20

2.4 Key figures for data subsets A, B, and C of the MEDTRANS corpus. 24

2.5 Annotation deviation categories of the MEDTRANS corpus with examples . 25

2.6 MEDTRANS transcription label statistics per subset. 26

2.7 MEDTRANS transcriber agreement measured in percent agreement and Cohen's κ for all labels and divided into phonetic classes in subset C. 28

2.8 Key figures for the MEDALIGN corpus data sets. 30

3.1 Key figures for pronunciation classification evaluation data sets SWBTRANS and MEDTRANS. 45

3.2 Recall, Precision, and F1 measures determined for minimum distance classification with data sets SWBTRANS and MEDTRANS. 46

3.3 Ranking results: percentage of data samples where the correct word appeared at rank k in the pronunciation classification results list. 47

3.4 Number of model parameters for the stochastic measures on the test data. . . 48

4.1 The SPARC approach to text reconstruction. 58

4.2 Reconstruction results in % for semantics-based rules in comparison to baseline systems. 71

4.3 Reconstruction results in % for phonetics-based rules in comparison to baseline systems. 71

4.4 Reconstruction results in % for combinations of the baseline identity, semantics-, and phonetics-based rules. 73

4.5 Rule matching counts and percentages for the combined rule systems. 74

4.6 Reconstruction results in % for baseline systems, the best rule-based system, and data-driven systems. 75

4.7 Language model details: Number of tokens and out-of-vocabulary (OOV) rate. 76

4.8 ASR results for the tested language models: Word error rate (WER) and relative difference (rel. △) to the Classic (baseline) model in [%]. 77

5.1 Ranked list of confusability measurements for base word `ACCESS` with pronunciation `/{ k s e s/`, confusion matrix C, and lexicon P (cf. figure 5.9). . . . 93

5.2 Native and non-native speaker accents annotated in the NNS102 data set. . . 95

5.3 WER in [%] and pronunciation variant (PV) counts for various optimisation strategies and evaluation setups: Fixed thresholds for all speakers (fix) and optimal thresholds per speaker (opt). 96

5.4 WER in [%] and pronunciation variant (PV) counts for strategy WFSTc and different candidate word sets: Fixed thresholds for all speakers (fix) and optimal thresholds per speaker (opt). 97

5.5 WER in [%] and pronunciation variant (PV) counts for strategy WFSTc and different sources for speaker-specific confusions: Fixed thresholds for all speakers (fix) and optimal thresholds per speaker (opt). 97

5.6 WER in [%] and pronunciation variant (PV) counts for unadapted and adapted acoustic models with WFST and CART approach: Optimal thresholds per speaker (opt). 98

5.7 WER in [%] for unadapted and adapted acoustical models with WFST and CART approach: Optimal thresholds per speaker, best results per group in boldface. 99

A.1 Phonetic symbol sets: International Phonetic Alphabet (IPA), Philips Speech Processing Phonetic Alphabet (PSPPA), Speech Assessment Methods Phonetic Alphabet (SAMPA), Advanced Research Projects Agency Alphabet (ARPABET). 108

A.2 Phonetic class assignment in ARPABET and SAMPA phonetic alphabet notations. 110

A.3 Lexical category tags from the Penn Treebank Tag Set [73] used for CART pronunciation modelling. 111

B.1 Word counts (WC) and pronunciation variant counts (VC) from the MEDTRANS corpus, subsets A and B. 112

B.2 List of the 100 most frequent phone confusions observed in the MEDTRANS corpus: canonical pronunciation (C) and observed pronunciation (O). 113

B.3 MEDTRANS annotation deviation category distribution – sorted by category. 114

B.4 MEDTRANS annotation deviation category distribution – sorted by count. . 115

B.5 Mapping between transcribers T_i and reports for subset C of the MEDTRANS corpus. 116

List of Notations and Acronyms

Typographic conventions

Function	Type	Example
word, orthographic representation	monospace	`word`
word, phonetic representation	monospace w. slashes	`/a k s @ s/`
sequence	angle brackets	⟨...⟩
tupel	parentheses	(...)
set	braces	{...}

Mathematical notational conventions

Meaning	Type	Example
symbol sequence of length N	lowercase + superscript	x^N
symbol at position i within sequence	subscript	x_i
symbol alphabet	calligraphic letters	$\mathcal{X},\mathcal{Y}$
alignment	capital greek letters	Λ,Π
estimate	'hat'	$\hat{p}$
parameter (set)	greek letters	α,θ

Statistical boxplots

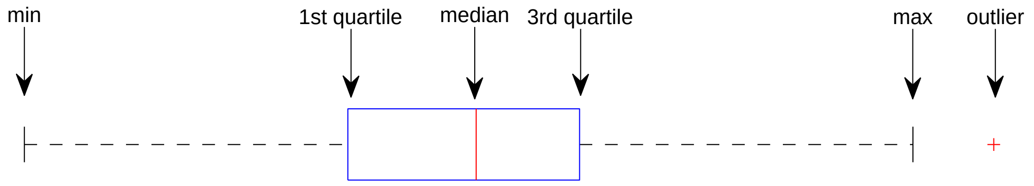


Figure 1. Statistical boxplot: The boxed region is defined by the 1st and 3rd quartiles of the underlying data distribution, i.e., the 25th and 75th percentiles, and the median is indicated by a red line. Points beyond the maximum whisker range (1.5 times box width for each whisker, i.e., dashed line) are marked as outliers.

List of abbreviations and acronyms

Acronym	Meaning
ASR	Automatic Speech Recognition
ARPABET	Advanced Research Projects Agency (ARPA) Phonetic Alphabet
APT	Automatic Phonetic Transcription
BNF	Backus-Naur Form
CART	Classification And Regression Tree
CFM	Confusion Matrix
DBN	Dynamic Bayesian Network
DTW	Dynamic Time Warping
EM	Expectation-Maximisation (algorithm)
FST	Finite State Transducer
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IPA	International Phonetic Alphabet
IR	Information Retrieval
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A-Posteriori
MLLR	Maximum Likelihood Linear Regression
MT	Medical Transcription
NLP	Natural Language Processing
NN	Neural Network
OOV	Out-Of-Vocabulary
OT	Optimality Theory
PM	Pronunciation Model
PFSM	Probabilistic Finite State Machine
PSPPA	Philips Speech Processing Phonetic Alphabet
PV	Pronunciation Variant
RBF	Radial Basis Function
RV	Random Variable
SAMPA	Speech Assessment Methods Phonetic Alphabet
SED	String Edit Distance
SPARC	Semantic and Phonetic Automatic Reconstruction of Dictations
SVM	Support Vector Machine
UMLS	Unified Medical Language System
WER	Word Error Rate
WFST	Weighted Finite State Transducer

Chapter 1

Introduction

Automatic speech recognition (ASR) has become one of the main applications in speech technology over the years. ASR systems have evolved from very simple speaker-dependent small-vocabulary toy applications to a greatly diversified class of complex applications serving a wide range of purposes within a large industrialised market. While the performance of single systems is respectable considering the complexity of the tasks, an all-in-one, multi-purpose, multilingual speech recognition system with human-like recognition performance is still not at hand. In a recent overview on the state-of-the art in ASR, O'Shaughnessy [86] lists performance evaluation results for a number of specific ASR tasks and databases as they have improved over the years. In comparison to human speech recognition capabilities, the current system performance is still rather poor. Even more so, in a study on the prospects of purely data-driven ASR system development, Moore [81] concludes that when extrapolating current system data requirements to recognition rates of 100%, an ASR system would have to be trained on 600,000 to 800,000 hours of speech data which in turn is equivalent to between 4 and 70 human lifetimes of speech exposure[1]. Considering the fact that even children are able to understand their parents, future ASR system development cannot rely on data alone.

Besides the mere performance figures, the user's expectations on ASR systems are particularly high. Biased by science fiction fantasies, people expect Star Trek whenever they use this technology, although it is at present not able to deliver the same results. Furthermore, input failures are not experienced as user errors like with keyboard input, but as system errors which cannot be influenced by the user [55]. Correct user input is wrongly interpreted by the system – a frustrating experience.

For this reason, the main challenge in ASR system development even before any technological problem solution is the design of systems that are accepted by their users. On the one hand, the best way to achieve this goal is to continuously improve the recognition performance. This is a feasible strategy as long as the efforts for further reducing the error rate are reasonable. On the other hand, it is equally important to design the human-system interaction appropriately to ensure the feeling of success for the user and at the same time avoid the annoyances that are associated with ASR usage. This thesis is a contribution to this endeavour.

1.1 Large vocabulary continuous speech recognition

Large vocabulary continuous speech recognition (LVCSR) is one of the most challenging specific applications in automatic speech recognition. Apart from the task of recognising the

[1] Depending on the actual lifetime and the language environment.

acoustics of spoken words correctly, the continuous nature of natural spoken speech has to be modelled as well with the help of a separate language model. For an appropriate coverage of a language lexicon sizes of 30,000 lexemes and above are necessary, including a wide range of phone contexts for the acoustic observations. Therefore, LVCSR is also a computationally demanding application.

The technological challenges in LVCSR are manifold. To account for more natural interaction, LVCSR systems have to move away from recognising read speech to spontaneous speech speaking style. Closely related to this problem is the recognition of non-native speech [36], a fact that is currently often ignored in today's system deployment. The same is true for fast speech, i.e., speakers talking at a much higher tempo than average speakers, which is characterised by phone duration changes and deletion of whole phones [79]. Other challenges are multi-lingual speech recognition [112] and, linked to that, the coverage of resource-limited languages.

One way of dealing with these issues would be to solve them with the help of specifically prepared data that contains the desired information. By broadening the data basis of an ASR system some of the above mentioned challenges can be solved. The benefit offered by this approach is, however, also its main problem. Collecting and annotating appropriate data bases is a tedious and expensive process and for handling a growing diversity of situations, the data collection approach is becoming more and more inefficient.

In this context, adaptation methods are a promising solution. The idea is to use only small amounts of data together with prior knowledge about the problem in question to tune a very general background model to a specific situation. The desired information, however, has to be part of the background model already, as the adaptation data is mainly used for re-weighting the existing model. For this reason, it is particularly suitable for tuning a system to e.g., a specific speaker, or a particular scenario for which it should be optimised. Adaptation techniques can be implemented at all modelling stages within an ASR system.

1.2 Phonetic and phonologic knowledge in speech technology: A way to meet the open challenges?

The integration of phonetic and phonologic knowledge in speech technology has been pursued for many years already. In several workshops this approach has been discussed from various aspects and many papers have been published on this subject. For motivating this thesis as a contribution to these efforts, some of the arguments from a panel discussion and printed in [7] shall be summarised here briefly.

The historical evolution of ASR systems brought a shift in technology away from linguistically motivated systems to almost purely statistically motivated systems in the late 1970s and early 1980s. As Ainsworth [1] states in his argument, statistical methods have proven to be superior in terms of the speech corpora and challenges available at this time. Now with today's challenges particularly in spontaneous speech and for ASR in noisy environments, the purely data-driven systems reach their limits. Phonetic knowledge may help out if it is employed for deriving more elaborate mathematical models of speech production and perception.

The fusion of phonetic sciences and speech technology could in fact be beneficial for both disciplines as Greenberg illustrates in his article [38]. In a study on automatic stress detection for conversational speech he compares a number of features to describe stress accent in English. Phonetic studies on controlled speech so far have indicated the fundamental frequency

f_0 to be the primary indicator for stress accent in English. In experiments on conversational speech, however, stress models based on amplitude and vocalic duration returned better results than those operating on f_0. Therefore, he concludes that speech technology is able to complement and extend the knowledge gained from phonetics.

Phonetic knowledge is, however, not the silver bullet of speech processing according to Strik [116]. In his review on various works on integration of phonetic knowledge in speech technology, he concludes that it is difficult in general to directly apply phonetic knowledge in speech processing. Phonetic studies operate on highly controlled recordings of so-called "lab-speech" while speech technology is founded on more realistic speech. Furthermore, the phonetic models are not adequately quantified to be employed directly into an ASR system. Nevertheless, in some tasks like, e.g., pronunciation modelling, phonetics provides powerful knowledge in the form of rules which only has to be complemented by appropriate quantifications derived from data.

This thesis follows the arguments of Greenberg and Strik by providing methods and analyses that supply knowledge to both, phonetics and speech technology, and by grounding them on real-world collections of data.

1.3 Speech and language resources for system development

Speech technology applications are intrinsically tied to the speech and language resources that they are based on. For ASR, these resources are audio recordings which are transcribed appropriately to obtain a ground truth for the statistical models of speech. A transcription is appropriate if it fully reproduces the relevant information of the source data sample in the terminology of the developed system. In a text-to-text scenario this kind of transcription could be termed *verbatim*. For a speech-to-text transcription an appropriate term would be *literal*, as the meaning of an audio segment remains the same in both, the audio recording and the textual transcription. The production of such resources is an expensive, time-consuming process, because it has to be done by trained human transcribers. For this reason it is not applicable to real-world use-cases like e.g., system adaptation to new speakers, or compilation of massive amounts of data.

For some speech technology applications, it is possible to resort to data resources which were not intended for system development originally. These may be automatically annotated field data collected for quality assurance purposes, or manually produced transcriptions in a different transcription style or level of detail, or possibly even very far related transcriptions such as lecture notes or presentation slides. Such resources are often available at low cost and in large amounts, but unfortunately not of the desired quality for system development. They exhibit quality deficits in various ways, depending on the initial purpose that they had been created for. Most of the time such texts are incomplete or partly incorrect. Transcriptions may also be imprecise, not properly reproducing the correspondence between a particular audio segment and its textual annotation. Sometimes, the text data is also only a derivative reproduction of the original wording, meaning that a text transformation process has been applied before. Due to the manifold deficits these resources will be simply termed *non-literal* throughout the thesis.

This data resource constraint must be considered for the development of an adaptation process. Phonetic and phonologic knowledge may help in dealing with the quality deficits of non-literal transcripts. The extend to which certain phenomena may be explained is, however, an open research issue that is addressed in this thesis.

1.4 Outline of the thesis

1.4.1 Problem statement and main hypothesis

The main hypothesis that will be discussed in this thesis is:

> *Are phonetic/phonologic algorithms suited to overcome the gap between literal and non-literal text resources, such that large amounts of non-literal transcripts can be employed for the development/improvement of medical dictation ASR systems?*

This hypothesis will be tested in terms of experimental evaluations for specific application-driven solutions to LVCSR challenges. For this reason, the scope of these investigations has to be defined appropriately in advance.

The experiments were all conducted against the background of an LVCSR system for the domain of medical dictation. This restriction to a single technology and domain appears reasonable as the thesis is intended to show exemplary results created from real-world data and under realistic conditions for implementation. However, the results are not primarily reported in the form of word error rates – the ultimate goal in speech recognition system development – but also in problem-specific performance measures that illustrate the performance of the method as such and not its effects within an LVCSR system. In fact, the strategy pursued in this thesis is that of minimum-invasive or even non-invasive system optimisation as it could be desired by ASR service providers which do not develop ASR technology themselves, but who have large amounts of data available for ensuring and improving their quality of service. Such a strategy was already proposed in works like [102] some time ago, and it is still meaningful these days, although the motivations for doing so might have changed. Today's highly integrated systems are not as easily tweakable as earlier systems used to be. Implementing pre- or post-processing steps is therefore easier than opening up a quasi-"black-box" system.

The challenges that will be addressed in the thesis are twofold: First, there is the problem of using non-literal transcripts for methods that require literal transcripts such as training or adaptation of probabilistic models. In this context, transcript means text data representing an annotation for the actual audio data. The "non-literalness" of the transcript is introduced by the speakers, the recognition process, and the transcriptionists reviewing the generated documents as detailed in chapter 2. The approach to meeting this challenge will be to make the deviations between literal and non-literal transcript types visible, relate them to each other and finally eliminate them to end up with a more literal annotation that is suitable for training. The second challenge comes from the application and application domain itself. The data is characterised by spontaneous speaking style, high rate realisations, and non-native speech. These difficulties will be addressed by an adaptation approach to better match the individual speaker characteristics of problematic speakers. Again the approach is designed to be minimum-invasive into an LVCSR system.

There are already existing solutions to these challenges. The technique of lightly supervised training [64] handles the problem of non-literal reference transcripts for, e.g., broadcast news transcription. Closed-captions of television broadcasts are used here as imprecise reference annotation for large amounts of unlabelled acoustic training data. In essence, lightly supervised training is a two-step procedure for re-estimating acoustic models. In a first step, the unlabelled acoustic data is automatically transcribed with a set of acoustic and language models that have been trained on a small amount of hand-labelled data. Then, the non-literal transcripts are aligned with the automatic transcripts and the portions of speech segments with matching transcriptions is used for re-training the acoustic models in an iterative fashion. Since the non-matching segments are discarded in each iteration there is a considerable loss of data which may result in the need for extra iterations or possible losses in terminal performance. The approach proposed here tries to remedy this issue and maximise the amount of data available at each iteration.

The challenge of speaker adaptation is addressed with acoustic adaptation methods such as Maximum-A-Posteriori (MAP) [65] or Maximum Likelihood Linear Regression (MLLR) [66] speaker adaptation. MAP adaptation is a Bayesian adaptation of the Gaussian acoustic model parameters. These parameters are assumed to be random and distributed according to a conjugate prior distribution, such that an MAP estimate can be obtained easily. As for each state the model means and variances are estimated separately, a large amount of adaptation data is necessary to improve the system performance. MLLR adaptation transforms only the mean vectors of a continuous Gaussian mixture acoustic model to end up with a speaker-dependent model. To reduce the amount of required training data for estimating the transformation matrices, similar states are grouped into regression equivalence classes for which the transformation is determined jointly. For this method, already small amounts of adaptation data on the order of a few seconds per speaker are enough to lower the word error rate significantly. Both methods, however, require direct access to the acoustic model parameters of the ASR system and are, therefore, highly invasive.

1.4.2 Scientific contributions

The scientific contributions of this work shall be summarised at this point to highlight the novel aspects of this thesis. With regard to the current scientific literature the following contributions can be mentioned – in the order of importance:

- A detailed analysis of the deviations between manual reference transcripts, recognised transcripts, and final corrected documents. To the best knowledge of the author, empirical studies on this kind of data are not available in the current scientific literature. From the scientific point of view this analysis is highly valuable as it enumerates and quantifies the phenomena that distinguish literal from non-literal language resources. The insights helped significantly in developing methods for finding correspondences between the different text types.

- A novel method for the alignment of recognised transcripts and final corrected documents at multiple levels of segmentation. This alignment procedure has been developed in cooperation with partners at the Austrian Research Institute for Artificial Intelligence (OFAI) and Philips Speech Recognition Systems, Vienna. Within this alignment framework the integration of phonetic matching and particularly matching on sub-word level have been suggested by the author of this thesis. The alignment procedure as a whole is a significant technological milestone in the processing of multiple literal and non-literal transcript types as it allows the compilation and analysis of large text databases – a task which has not even been mentioned in the scientific literature up to now.

- The idea of reconstructing a quasi-literal transcription from paired non-literal transcripts for the domain of medical dictations. Both the explicit reconstruction task and the idea of using literal and non-literal transcripts for this task is novel. The detailed evaluation of the specified reconstruction rules is added scientific value to the mere adaptation application. Furthermore, this idea has been implemented in a practical method.

- The extension of an existing framework for pronunciation modelling to an optimisation technique that generates speaker-specific pronunciation variants under the constraints of minimising confusability and maximising the gain in modelling accuracy from non-literal speaker adaptation material. Balancing confusability and accuracy explicitly for new pronunciation variants has not been implemented before. Furthermore, this pronunciation modelling framework is evaluated for the first time in terms of a LVCSR dictation system.
- The design, production, validation, and evaluation of a corpus of phonetically transcribed medical dictations recorded under field conditions. Up to now, a corpus of dictated speech in conjunction with related literal and non-literal text resources has not been available for research purposes.

Parts of this work have already been published and presented at international, peer-reviewed conferences:

- S. Petrik and G. Kubin: Reconstructing Medical Dictations from Automatically Recognized and Non-Literal Transcripts with Phonetic Similarity Matching, In *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2007, Honolulu, Hawaii, pp. 1125–1128 [90]
  In this paper, the medical dictation reconstruction task is presented and the phonetic similarity measure for this task is described. The theoretical framework was developed by both authors, while the experimental evaluation was done by the main author.
- S. Petrik and F. Pernkopf: Language Model Adaptation for Medical Dictations by Automatic Phonetics-Driven Transcript Reconstruction, In *Proceedings of the IASTED Intl. Conf. on Artificial Intelligence and Applications (AIA)*, 2008, Innsbruck, Austria, pp. 194–199 [92]
  Based on the previous findings on phonetic similarity measurement, its application to language model adaptation is described in this paper together with a small evaluation in terms of language model perplexity. The main author contributed the theoretical framework and the experimental evaluation. The automatic classification experiments were provided by the co-author.
- S. Petrik and F. Pernkopf: Automatic Phonetics-Driven Reconstruction of Medical Dictations on Multiple Levels of Segmentation, In *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2008, Las Vegas, Nevada, pp. 4317–4320 [91]
  The contribution of this paper is an extended text matching scheme that works on multiple levels of segmentation and solves segmentation error issues in alignments of non-literal transcripts. The main author contributed the theoretical framework and the experimental evaluation. The automatic classification experiments were provided by the co-author.

Chapter 4 is a verbatim reprint of the article *Semantic and Phonetic Automatic Reconstruction of Medical Dictations*, submitted to the journal *Computer Speech and Language* by Elsevier. It is co-authored by Christina Drexel and Leo Fessler from Philips Speech Recognition Systems Vienna, Jeremy Jancsary, Alexandra Klein, Johannes Matiasek, and Harald Trost from the Austrian Research Institute for Artificial Intelligence (OFAI), and Franz Pernkopf and Gernot Kubin from the Signal Processing and Speech Communication Laboratory at Graz University of Technology. The sole contributions of the author are the phonetic similarity matching function, the phonetic reconstruction rules, and the experimental evaluation of the reconstruction quality at the end of this chapter.

1.4.3 Document structure

The thesis is divided into three major parts according to the goals defined earlier on:

Part I gives an introduction to non-literal transcript matching within the scope of this thesis. As most of the natural language processing applications, also non-literal transcript matching is half engineering and half algorithmic science. Chapter 2 is devoted to the data resources available in an ASR-supported medical transcription system. The text types are analysed and related to each other to get insights into the deviations that have to be expected in similarity matching, particularly at the phonetic level. As additional outcome, two text corpora have been compiled that will be used throughout the rest of the thesis for experimental evaluations. Chapter 3 deals with the problem of similarity matching for phone symbol sequences which will be used to solve the application-driven problems later on.

Part II presents solutions for problems coming directly out of the application-domain of LVCSR. The first problem described in chapter 4 is the reconstruction of a literal transcript of a medical dictation out of the non-literal machine-recognised and final, human-corrected medical reports. The proposed solution utilises semantic and phonetic similarity measurement in conjunction to classify aligned text chunks as either recognition errors or reformulations introduced by the medical transcriptionist. The second exemplary problem from LVCSR – pronunciation modelling for non-native speakers – is presented in chapter 5. Phonetic similarity matching is applied here for generating speaker-specific pronunciation models from small amounts of available adaptation data. The framework for pronunciation generation allows the definition of measures for lexical confusability and accuracy gain measures that provide the means for an optimisation approach where accuracy and not confusability is increased.

The third part of the thesis tries to look beyond the proposed solutions. It gives an outlook on further corpus work – one of the original starting points for this thesis, discusses ideas how the algorithms can further be improved, and which other application-oriented problems might be solved with them. With the experimental evaluations from the previous parts and the potential analysis of the third part, the initial hypothesis will be revisited in the conclusion together with a critical review and recommendations for further research.

Part I

Fundamentals of Non-Literal Transcript Matching

Chapter 2

Literal and Non-Literal Text Resources in an ASR-Supported Medical Transcription Environment

The success of a Natural Language Processing (NLP) system cannot only be attributed to algorithms alone, but also to the data resources these algorithms are built upon. Particularly for applications using statistical methods such as ASR or speech synthesis, language resource engineering has become just as important as algorithm engineering. While algorithm engineering requires expertise in mathematics and computer science, language resource engineering is the tedious process of gathering prior knowledge by manual inspection and careful analysis of large amounts of data. Nevertheless, major efforts of the NLP community on structured and standardised analysis of large text corpora have led to substantial and sustained improvements in this field. [63], [41], [33], [72]

One of the main aims of this thesis is the utilisation of non-literal text resources as the basis for algorithm engineering in an NLP system. In contrast to literal text resources that are directly related to the speech data, non-literal transcripts are only indirectly derived from the users' utterances and, therefore, not directly useable for current algorithms. As a consequence, corpora of paired literal and non-literal transcripts as language resources are virtually not available. The preparation and design of such text corpora is one of the contributions of this thesis and will be described in this chapter.

The analyses and preparatory works are not only targeted on the orthographic domain, but also extended to the phonetic domain. From the literature on phonetics and phonology, it is well known that reduction phenomena are commonly observed in spontaneous speech (e.g., [21], [41], [53], [113]). Any algorithm that should exploit this inherent prior phonetic knowledge of the data requires annotation at this level of detail. Despite the focus on phonetic algorithms, the multi-alignment medical dictation text corpus presented at the end of this chapter was designed to be easily extensible with other kinds of text data.

Before the text resources are introduced, the terms *error*, *deviation*, and *mismatch* that will be used throughout this chapter shall be defined. An *error* is an isolated, but possibly systematically occurring negative event (a mistake) that has been observed within a single text source. *Deviations* in contrast are neutral observations of differences between two different text sources. *Mismatches* finally describe deviations that are perceived in a negative sense, usually when also the alignment of the texts is wrong.

This chapter starts with a brief introduction to medical transcription, how ASR is at present used to support the transcription workflow, and which text resources are generated in such large-scale document production environments. The individual types of literal and non-literal transcripts are explained in detail together with exemplified observations and rough quantifications. Based on these findings, two text corpora derived from medical transcription field data are presented that will be used in later chapters of this thesis.

2.1 Medical transcription

Medical transcription (MT) has evolved as a supporting health profession from the basic stenography-type task performed by single transcriptionists to a big industrialised market in the United States with distributed processing done by many thousands of homeworkers. This trend is not only driven by the pressure for cost reduction, but also by legal restraints on record keeping of patient data. The substantial progress in speech and language technology has made these growth figures possible.

2.1.1 Transcription workflow

The traditional medical transcription workflow is depicted in figure 2.1. The physician is examining a patient and dictates his findings to a recording device or online via telephone for distributed processing. The dictation recording is then transcribed by a professional medical transcriptionist (or 'medical editor') who compiles it into a written document – the medical report. The transcribed report needs to be approved by the physician again to ensure that critical passages like, e.g., dosages or medications were transcribed correctly. After approval, the medical report is handed over to the patient and filed for later reference [62].

The transcription task requires special skills from the transcriptionists which are trained in specific training programmes. This includes knowledge of medical domain vocabulary in terms of physiology, current medications, and also soft skills such as a consistent, rhythmic working pace, or keen perception to understand even recordings of bad quality. The physicians are required to speak clearly, slowly, and unambiguously, but in some working environments like, e.g., in emergency outpatient units these standards are often violated as there is only little time for dictating.

2.1.2 ASR in medical transcription

The modified workflow with ASR technology is shown in figure 2.2. In contrast to figure 2.1, the dictation recording is first processed by an ASR system that creates a draft transcript of the dictation. This transcript is then passed on to the medical transcriptionist and just corrected instead of transcribed from scratch [62]. Depending on the transcript requirements, this process leads to major savings and increased productivity in the document creation process.

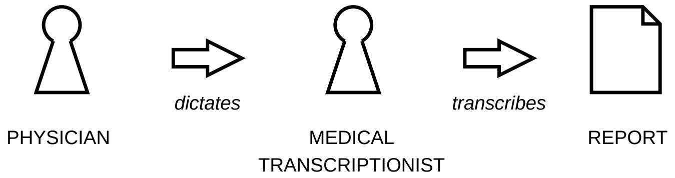


Figure 2.1. Traditional workflow in medical transcription.

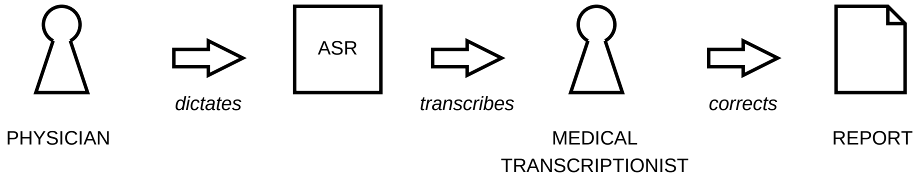


Figure 2.2. ASR-supported workflow in medical transcription.

The transcription task for the ASR system can be specified as follows. Medical dictation is a large vocabulary continuous speech recognition task with highly domain-specific vocabulary that is combined with standard text vocabulary for freely dictated passages. For some investigations, a further division according to the medical discipline (e.g., cardiology, emergency medicine, ...) or the type of final document (e.g., operational summary report, radiology report, ...) is helpful. Dictated speech as speaking mode can be described as freely articulated, but by professional users[1]. The ASR system has to handle a large number of users in a broad range of language proficiency levels (native and non-native speakers). The recordings are done indoors over a band-limited telephone channel in a possibly reverberant acoustic environment with a certain level of background noise (e.g., in emergency outpatient units). As an additional requirement, the desired output should conform to certain formal standards in terms of formatting and style that need to be imposed on the draft transcript.

2.1.3 Text resources generated in the course of the workflow

The overall goal of this document creation process is the production of a medical report from dictated speech. If the physicians spoke ready-for-print, the ASR system worked perfectly, and the transcriptionists were not creative in their reviewing process, the textual output would be perfect at each processing stage. Unfortunately, this is not the case, and there are two kinds of imperfections in the output texts that will be distinguished:

- System errors
  Each actor introduces errors into the workflow. Some errors may be automatically corrected in post-processing if there is reliable knowledge about them. They become visible, however, only in comparison to a true reference for each processing stage, which is not always available.

- Non-literal output
  With respect to the audio reference, a degradation of the transcription accuracy can be observed after each processing step. In section 1.3, the term *literal* was introduced for transcriptions which fully reproduce the relevant information of the reference data in the terminology of the developed system. In contrast to that, transcripts become *non-literal* as new errors are introduced or previous errors are compensated.

Studying the system errors is difficult, as most of the generated transcripts are non-literal and not suited as a reference for evaluating the processing stages separately. Figure 2.3 illustrates the text resources that are generated in the course of the transcription process:

[1] Although professional, it is not possible to assume the users to be cooperative, as many professional dictation systems operate in *batch mode* where automatic speech recognition is done offline and not during dictation. In this scenario the speakers are not aware of the existence of an ASR system in the background.

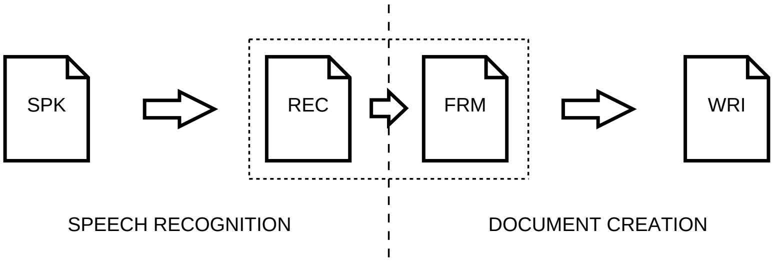


Figure 2.3. Overview of documents created during the transcription workflow: manual transcriptions (SPK), draft transcripts (REC), formatted draft transcripts (FRM), and final corrected medical reports (WRI).

Manual orthographic and phonetic transcriptions (SPK) are still literal transcripts, while the draft transcripts or recognised texts (REC, FRM) and the final medical reports (WRI) are already non-literal transcripts. The workflow implies two tasks that need to be solved consecutively as shown in figure 2.3: the speech recognition task which aims at accurately reproducing the speakers utterances, and the document creation step that creates a formal report from an informal dictation transcript. According to these two tasks text types can either be assigned to the *spoken* or the *written* level.

At present, text corpora from the medical dictation document production workflow are not publicly available due to legal constraints concerning patient information and commercial interests of ASR-technology providers. The analyses and evaluations presented in this thesis were done on a subset of data that Philips Speech Recognition Systems uses for benchmarking and monitoring their operational LVCSR systems for American English medical dictation. These are realistic data from the medical domain, comprising all presented text types and providing enough material for training and testing of statistical measures. From these data two speech corpora were compiled: the MEDTRANS corpus of phonetically transcribed medical dictations (cf. section 2.5.1) and the MEDALIGN corpus of multi-alignment medical dictations (cf. section 2.5.2). The specification and compilation of these corpora is also a contribution of this thesis.

In the following discussion of literal and non-literal text resources the statistics and examples presented are directly derived from the two corpora. A detailed description of the production of these corpora together with key data figures will be given later in section 2.5.

2.2 Literal text resources

2.2.1 Manual orthographic transcriptions

Manual orthographic transcriptions are literal transcripts of dictations produced by human transcriptionists for offline documentation purposes. Assuming that the literal transcription process is error-free, all insights on this type of texts reveal information about speaker errors. The orthographic notation is by definition incomplete for this task and therefore usually extended by text markups indicating non-speech events. Furthermore, transcription guidelines define how unclear or incomplete utterances should be documented.

Table 2.1 gives a brief summary on a number of speaker errors that were automatically extracted from the manual orthographic transcriptions in the PROFILE set of the MEDALIGN corpus (see section 2.5.2). Unconsciously produced errors are listed in the first part of table

Table 2.1. Speaker errors in manual orthographic transcriptions of the MEDALIGN-PROFILE data set, sorted by counts.

Error category	**Example**	**Count**	[%]
hesitations, filled pauses	‘ahm’, ‘uhm’, ‘hmm’	22,652	28.87
non-speech	breathing, laughter, coughing	4,967	6.33
self-corrections	‘digi* digits’	2,128	2.71
repetitions	‘again again’	270	0.34
syntactic structure omitted	‘’ → ‘period’, ‘’ → ‘comma’	37,816	48.20
textual structure omitted	‘’ → ‘new line’, ‘’ → ‘new paragraph’	9,543	12.16
abbreviations, acronyms	‘EOMI’ → ‘extraocular movements intact’	705	0.90
short forms	‘meds’ → ‘medications’	336	0.43
instructions	‘scratch that’, ‘all capital’, ‘please add’	37	0.05
TOTAL		78,454	100.00

2.1, and consciously produced speaker errors that are clearly opposing the guidelines for medical dictation are listed in the second part of the table. From a total of approximately 312,000 spoken words in the PROFILE set, 78,454 could be identified as speaker errors according to the above error categorisation.

- Hesitations & filled pauses
  The most prominent speaker errors are hesitations and filled pauses. On average about one in fifteen spoken words is actually such a “non-word”, which makes it by far the most frequent word in a medical dictation[2]. Hesitations and filled pauses serve various purposes, for instance, structuring the speakers’ utterances and thoughts [117], or providing time for finding the right word. As artifacts that do not convey meaning of the dictated text as such they are undesired in the final medical report and need to be removed. Considering the ASR task, it is important to correctly recognise them (even just for removing them) and to avoid a misrecognition as an ordinary short word.
  E.g., `‘the patient [ahm] had ...’` ↔ `The patient had ...`

- Self-corrections & repetitions
  Interruptions of the speaker’s train of thought are not only indicated by hesitations, but also by self-corrections or repetitions. Usually, a self-correction is characterised by only partially realised words that are immediately followed by either the intended word or another, correct word. In the manual transcriptions a broad spectrum of self-corrections can be observed, ranging from short slips (`be*`, `mis*`, ...) over nearly complete repetitions (`*lieves`, `assessmen*`, ...) to nonsense words (`*lumpempt*`, `regurtition*`, ...)[3]. Such word fragments are critical for the speech recognition step, because they are not contained in the ASR lexicon and are easily confused with other similar sounding words. Repetitions are similar to self-corrections in that the same word or phrase is uttered more than once in direct succession. They can harm the recognition quality as they are usually not represented in the ASR language model. Both kinds of speaker errors are unwanted in a final medical report and need to be corrected.

- Document structuring
  In a dictation scenario the speaker assumes that the utterances are transcribed literally and entirely and no additions are to be made by the transcriptionist. This implies

[2] The second most frequent word in the PROFILE set – `and` – occurred 6,961 times.

[3] It is not clear how accurately these orthographic labels really represent the audio segments.

that the utterances themselves are meant to be complete or "ready for press". It is, however, a common observation that syntactic as well as textual document structuring is generally omitted or inconsistently dictated. Syntactic structuring refers to punctuation while textual structuring comprises formatting, paragraphing, or formats like, e.g., enumerations. Enumerations are a particular source of confusion, since a consistent numbering is hardly maintained by the speakers due to counting errors, or the term "next number" which occurs mostly for enumerations with more than four to five items as they are typical for lists of medications.

E.g., `'past medical history unremarkable surgical history a vasectomy'`
↔ `</PAST MEDICAL HISTORY/> Unremarkable.`
`</PAST SURGICAL HISTORY/> Vasectomy.`

- Acronyms & abbreviations
  Due to time limitations and the high level of standardisation, many common medical terms are uttered in abbreviated form or as acronyms. Although not being a speaker error, these units need to be expanded by the medical transcriptionist in the final report in order to be understandable for the patient.
  E.g., `'HIV'` ↔ `Human immunodeficiency virus`
  `'PERRLA'` ↔ `pupils equal, round & reactive to light & accommodation`

- Meta-commands & instructions
  Finally, dictations also contain instructions to the transcriptionist which are a problem on their own. These short phrases like `'scratch that'`, `'please add'`, or `'go back to'`, for instance, need not only to be edited out of the final document whether they have been correctly recognised or not, but they also cause changes in the text that are hard to reproduce by an ASR system without semantic analysis[4]. There are even extreme cases, where the physician at the beginning of the dictation requests a standard template of his own to be used such that the rest of the dictation does not match the final medical report at all.

2.2.2 Manual phonetic transcriptions

In addition to orthographic manual transcriptions, a substantial number of dictations were re-transcribed on phonetic level to obtain insights into the articulation characteristics of speech produced in medical dictation and to have a reference for algorithm testing. At this point, a brief overview from the literature on phonologic variation in speech production is given together with selected examples from the MEDTRANS corpus (cf. section 2.5.1).

Phonologic variation in spoken language

Dictated speech is a special speaking style that does not directly correspond to any of the 'classical' styles read, spontaneous, or conversational speech. The utterances are planned, as the speaker is supposed to carefully select his words before speaking them as if he would be reading them off his mind. In practice, however, this is hardly the case, and the frequent interruptions in the train of thoughts result in disfluencies that often give dictations a more

[4] Longer and more complex instructions like `'oops I guess I am going back sorry going back to current medical problems when it said [ahm] left hip pain [ahm] say parentheses with reported per* fracture of the left hip in the nineteen eighties end of parentheses sorry about that one'` are rare, but not impossible.

spontaneous than read character. The observed surface pronunciations for this speaking style are (among other factors) mainly the result of high rate (fast) speech and the physical condition of the speaker.

The pronunciation of native American English speakers in professional dictation is only slightly to moderately accented, as the speakers usually conform to General (or Network) American English pronunciation. This quasi-standard which is perceived as accentless for most middle-class white Americans has its roots in the massive immigration from various countries (and hence different languages), the continuing trend for urbanisation, the high mobility of Americans in general, and the unifying American school system [77]. Most importantly, the introduction of radio and television in the 20$^{\text{th}}$ century then made this pronunciation of the originally Mid-Western regions predominant all over the United States as many broadcasting organisations and their broadcasters originated from that area [60], [61].

Phonetic variation in spontaneous speech can often be understood in terms of underlying phonologic processes [84],[113]. This allophonic variation allows the definition of rules that explain how specific phonetic realisations emerge from canonical baseforms. Phonologic processes result in assimilation, deletion, addition, substitution, or reduction of phones, depending on the actual phone context. For this reason, it is hard to represent a word with a single canonical pronunciation and a static pronunciation lexicon.

The variations are, however, not only of phonologic but also of phonetic nature. For fast speech it is clear that due to the reduced amount of time for conveying the same information, at some point losses have to occur. These losses can of course partially be attributed to mechanical constraints imposed by the vocal apparatus [68]. Still, further studies indicate that they are also driven by cognitive processes, claiming that phonetic realisations are determined rather "by habit than speed or inertia" as stated by Shockey [113]. The latter claim is supported by a list of factors for phonologic speech reduction ranked in a vulnerability hierarchy (cf. [113], p. 15). The following list of factors is inspired by the vulnerability hierarchy, but reduced to well-observable deviations in medical dictations.

- Word frequency
  Word frequency is a main factor for phonologic reduction. High frequency words such as content or filler words are usually more redundant than low frequency words like nouns or adjectives. The more information is conveyed in a word, the more it is likely that it will not be reduced or altered, simply for the reason that the recipient ought to understand the message. Previous studies on the SWITCHBOARD corpus support this observation [40],[32].

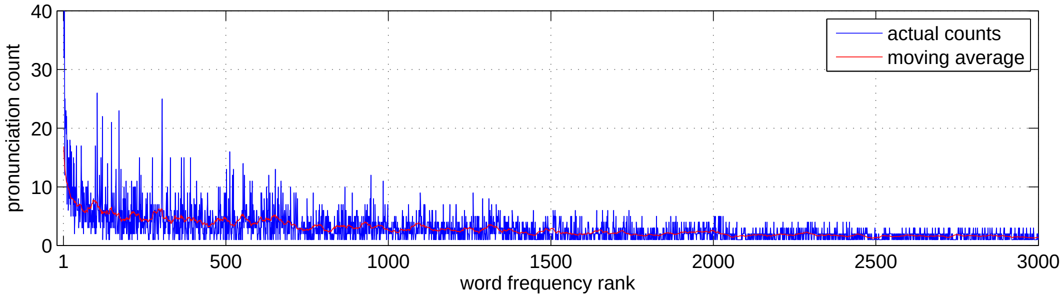


Figure 2.4. Number of pronunciation variants per word observed in the MEDTRANS corpus, sorted by word frequency.

Table 2.2. Pronunciation variants with syllabic structure for the words milligrams and regular from the MEDTRANS corpus.

Pronunciation variant	**Count**	**Pronunciation variant**	**Count**
/m I · l I · g r & m z/	198	/r e g · l R/	42
/m I · l @ · g r & m z/	29	/r e g · j @ · l R/	40
/m @ · l @ · g r & m z/	21	/r e g · @ · l R/	14
/m I · l I · g r @ m z/	6	/r e g · j @ · l 3 r/	7
/m I · l · g r & m z/	3	/r e g · j U · l @ r/	5
/m I · l I · g r & m s/	3	/r e g · j @ R/	5
/m @ · l I · g r & m z/	3	/r e g · @ · l @ r/	5
/m I · l I · g r & m/	2	/r e g · j @ · l @ r/	4
/m I · l @ · g r @ m z/	1	/r e g · U · l R/	4
/m 3 · l · g r & m z/	1	/r e g · U · l @ r/	4
/m 2 · g r @ m z/	1	/r e g · @/	3

This assumption is confirmed by the phonetic transcriptions of the MEDTRANS corpus. With decreasing word frequency, i.e., with increasing word frequency rank, the number of observed pronunciation variants decreases as well as shown in figure 2.4 and table B.1 in appendix B. While for the most frequent word – the hesitation [hes] – 60 different realisations were found, for the word physical at rank 100 the pronunciation variant count reduces to 9, and for the word discuss at rank 1,000 to only 3.

- Syllable structure
  Not all phonologic processes can be meaningfully described within the local scope of a phone, but instead within the larger context of syllables. The structure of syllables with an onset, a nucleus, and the coda gives some implications for explaining pronunciation variation. While onsets are very unlikely to be reduced, the nucleus may undergo a substitution, and the final coda is likely to be omitted. One explanation for this observation may be the high responsiveness of the auditory cortex to beginnings of sounds [37].
  Consider the word milligrams from the MEDTRANS corpus (268 realisations, rank 58 in the word frequency ranking). The syllabic structure is a good indicator for the observed phonologic variation as shown in table 2.2. While the onsets are never affected, the vowel nuclei /I/ and /&/ tend to be reduced or even deleted completely. The syllable structure of the word regular[5] (152 realisations, rank 104 in the word frequency ranking) is less stable. The first syllable remains constant, but the second and third syllable are very likely to be reduced or even merged into a single artifact as in /r e g · j @ R/ or /r e g · @/.

- Speaking rate
  For the English language the effects of speaking rate are manifested in the reduction or deletion of unstressed vowels. According to an experimental evaluation in [21] the conditions for these reductions are determined by the position within the word and the position related to the word stress. While in a slow rate mode post-stress vowels are more likely to be deleted preferably in word-medial positions, for fast rate mainly the word-medial vowels are affected by deletions, no matter if they are before or after the

[5] The word regular is part of the phrase 'heart regular rate and rhythm' used in the physical examination section of a medical dictation. This section is governed by such standardised phrases and therefore often uttered very fast and sloppily.

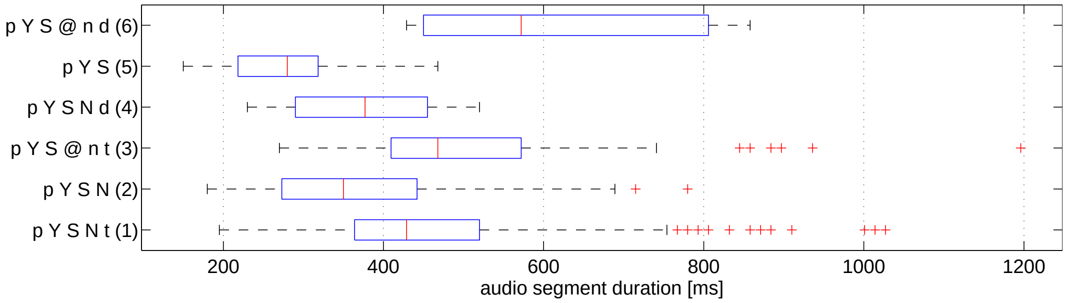


FIGURE 2.5. Duration boxplots for 6 of the most frequent pronunciation variants (bottom to top) of the word `patient` from the MEDTRANS corpus.

stressed vowel. Furthermore, there is evidence that the deletion of the schwa sound is governed by the syllable structure and also related to accompanying consonant changes in the environment [21]. Whenever the resyllabification of surrounding consonants as onsets and codas of wellformed syllables in careful speech is possible, the reduction of the syllable is highly likely to occur in fast speech.

An example from the MEDTRANS corpus illustrates these influences of speaking rate. The word `patient` is a highly frequent word in the MEDTRANS corpus (866 realisations, rank 17 in the word frequency ranking) with a total of 15 different observed pronunciation variants. For 6 of the most frequent pronunciation variants, the audio segment durations were collected. Figure 2.5 summarises the measures in boxplots. For longer audio segments, the realisations correspond to the canonical pronunciation `/p Y S N t/` or the properly articulated variants including an explicit schwa in the second syllable. With decreasing segment length, however, the word-final consonants (`/p Y S/`, `/p Y S N/`) are more likely to be deleted as well as the second-syllable schwas.

- Phonetic/Phonologic

  It can be observed that not all phones of a language are equally likely to undergo changes. The type of phone and the immediate environment of a particular realisation also determines whether it will be articulated canonically or not. Miller and Nicely were among the first to investigate the confusion of English consonants systematically [78]. For English, the alveolars `/t,d,n,l/` and the fricatives `/s,z/` are particularly vulnerable [113]. The result does, however, not always have to be a deletion, but may also be an assimilation with the subsequent phone.

  The MEDTRANS transcriptions support these assumptions. Table B.2 in appendix B lists the 100 most frequently observed phone substitutions calculated by performing a Levenshtein alignment between canonical and manual phonetic transcription (cf. chapter 3, section 3.3.1). The alveolars `/t,d/` are in fact those phones that are most easily confused or reduced. The phones `/n,l,r/` appear at the top of this ranking mainly due to their similarity with their syllabic variants `/N,L,R/` – a direct consequence of the Philips phonetic alphabet (cf. appendix A, table A.1). The fricatives `/f,v,s,z/` are less affected than the alveolars, but still among the main confusions. Vowel reductions to schwa or even full deletions occur often in this context.

- Non-nativeness

  A considerable amount of physicians in U.S. health institutions does not speak English as a first language. Therefore, non-native speech is another important observation in medical dictations that is reflected in the phonetic transcriptions. In contrast to the reduction phenomena discussed before, non-native speech results in phone addition, reduced coarticulation, and phone substitutions motivated by the primary language phone inventory. A more detailed account on non-native speech is given in chapter 5.

The deviations described so far have only involved single phones at most. Massive reductions of more than one phone or even whole syllables can also be observed. Studies suggest that this is a common phenomenon for colloquial speech [53], [94]. A detailed quantitative analysis of deviations in manual phonetic transcriptions is given in section 2.5.1 in terms of the MEDTRANS corpus description.

2.3 Non-literal text resources

2.3.1 Recognised texts

As shown earlier in figure 2.3, the output of the ASR system can be separated into an output of the speech recognition stage – the draft transcript – and an intermediate result of the document production stage – the formatted draft transcript. For analysis purposes, these two texts are discussed separately, although for correction and editing, only the formatted draft transcript is processed within the transcription process.

Draft transcripts

The goal of the draft transcription is to obtain a transcript which is as close to the manual orthographic transcript as possible. This means that the ideal draft transcript is an exact literal transcript of the audio recording including also non-speech and unusable speech, possibly labelled such that it can be automatically removed in the subsequent formatting stage. The draft transcript is the output of a speaker-independent LVCSR system trained on medical dictation training utterances. The types of errors produced by LVCSR systems are well known [55], [39]. The actual errors that may occur in the draft transcripts are, however, highly system-dependent. For this reason, the following error categorisation is meant to be descriptive and exemplary only.

- Word substitutions
  Similar words are easily confused by ASR systems which results in substitution errors. The confusability of two word depends on their phonetic similarity (i.e. the similarity between the phone sequences in the dictionary) and their chance of occurring in similar word contexts. Short words and, in particular, function words exhibit low discriminative power and are thus highly vulnerable.
  E.g., `‘one’` ↔ `none`

- Segmentation errors
  Depending on lexicon and language model, long words may be split into shorter words or vice versa. These errors are summarised under the term segmentation errors. Again, short words with high frequency of occurrence (e.g., function words) get easily deleted, or merged with neighbouring words.
  E.g., `‘rudimentary’` → `room ventrally`

- Insertions due to background noise/speech
  The quality of the draft transcript also strongly depends on the cleanness of the audio input. Background noise or non-speech utterances produced by the speaker may be misrecognised as short words. The same holds for background speech which can be commonly observed in dictations made in a hospital environment.
  E.g., `‘period [bg_noise] new paragraph ...’` → `period the new paragraph ...`

- Out-of-vocabulary words
  Out-of vocabulary (OOV) words and, in particular, proper names are a systematic shortcoming of lexicon-based ASR systems as they are by default misrecognised and may lead to further errors in their immediate neighbourhood due to language modelling side-effects.
  E.g., `‘Maverick’` → `Aimee the ER I seek a`

Formatted draft transcripts

After the speech recognition stage, the goal of the document creation step is to format the draft transcript according to formal and stylistic guidelines. The ideal formatted draft transcript is a final document that does not need any additional correction. The draft transcript is post-processed to compensate some types of speaker errors and to apply formatting as it is desired for the final medical report. In general, only those errors are handled which can be reliably detected and corrected after the complete draft transcript has been recognised.

- Compensation of speaker errors
  Hesitations, non-speech, and self-corrections are speaker errors that can be easily compensated by simply removing them from the draft transcript. If these events have not been correctly recognised the speaker error is propagated from the spoken to the written level beyond this processing stage.
  E.g., `‘myo* [ahm] myocardial infarction’` → `myocardial infarction`

- Punctuation
  Since few speakers dictate punctuation consistently and correctly, punctuation marks can be added automatically with the help of a stochastic punctuation model which inserts the marks tentatively at the desired positions. Some punctuation may be integrated into the lexicon as well, like e.g., colons or hyphens.
  E.g., `‘lungs clear’` → `LUNGS: clear.`

- Formatting
  The main focus of the post-processing stage is on formatting the draft transcript. Numerical expressions like e.g., dates, times, laboratory values, or medications have to be transferred from their spoken wording to a written digitised form. These deterministic mappings can be implemented with context-free grammars.
  E.g., `‘December 6’` → `12/06`

- Document structuring
  The document is structured according to formatting commands given by the speaker or by a document formatting model. Formatting in this respect means division into paragraphs, highlighting of headlines, and creation of enumerations.
  E.g., `‘history of present illness’` → `</HISTORY OF PRESENT ILLNESS/>`

TABLE 2.3. Modifications observed in final medical reports of the MEDALIGN-PROFILE data set, sorted by counts.

Error category	**Example**	**Count**	[%]
headings	‘allergies’ → {ALLERGIES}	5,547	44.91
contractions	‘I’ll’ → ‘I will’ ‘I’d’ → ‘I would’	1,600	12.95
capitalisation	‘Aspirin’ → ‘aspirin’	1,434	11.61
abbreviations, acronyms	‘A and O’ → ‘alert and oriented’	1,301	10.53
concatenations	‘intraoral’ ↔ ‘intra-oral’	653	5.28
numerus	‘respiration’ → ‘respirations’	566	4.58
tempus, genus	‘is’ ↔ ‘was’, ‘are’ ↔ ‘were’, ...	476	3.85
short forms	‘meds’ → ‘medications’	336	2.72
’the patient’	‘he’ → ‘the patient’	312	2.52
spellings	B U N → BUN	126	1.02
TOTAL		12,351	100.00

2.3.2 Final documents

According to specifications given by health authorities, medical reports need to fulfil standards concerning the document structure, the coverage and level of detail for particular kinds of medical reports, and other formal requirements related to writing style.

To get an overview of corrections and reformulations that occur between spoken and written text in medical dictations, an exploratory data analysis was performed on the PROFILE subset of the MEDALIGN multi-alignment medical dictation corpus (cf. section 2.5.2). The categories for deviations were chosen without regard to syntactic, semantic, text-analytic or other respects, only the subjective frequencies determined the primary selection. Since the counting was done manually the results collected in table 2.3 should be interpreted just as a ranking of categories for highlighting the relative differences between them. The most frequent categories are discussed in the following.

- Headings
  The most frequent deviation in the alignments are text parts which were formatted into a paragraph heading in the written form (44% of classified deviations). Whenever the text part only comprises a single word, the formatting is straightforward. As soon as several words are affected, this transformation can become quite complex. In that case, the formatting is often combined with a reformulation as in `‘he has no allergies’` → `{ALLERGIES} none`.

- Contractions, capitalisation, abbreviations & acronyms
  The correction of contractions, capitalisation, abbreviations, and acronyms are also frequently observed. The expansion of contractions in particular turns out to be highly ambiguous. In some cases, when there is also a temporal change involved, the result of the transformation process can be completely different from the original text. Again, formatting may be combined with reformulation.
  E.g., `‘he’s’` → `he has, he is, he was, ...`

- Concatenations
  Deviations in the writing style of concatenated words were also noted to be frequent. The interchanging usage of hyphens, spaces, and also the omission of a separating character occurs arbitrarily and can only be explained by the manual nature of the text generation process for both, the manual reference transcripts and the written reports.

- Numerus
  For nouns, a change in numerus can be observed frequently. In most of the cases, singular in the spoken text is changed to plural in the written text. Changes from plural to singular occur much less frequently. These deviations are clearly due to speaker errors in comparison to the previous errors which were of syntactic nature.
  E.g., `‘palpitation’` ↔ `palpitations`, `’murmurs’` ↔ `murmur`

- Tempus & Genus
  For verbs deviations in tempus and genus can be observed in about the same order of magnitude as the changes in numerus for the nouns. The deviations, however, do not seem to be systematic. Changes from past to present tense are about equally frequent as changes from present to past tense. Other changes are also possible, but rather rare in comparison to the above mentioned, just like deviations in modus(active/passive).
  E.g., `‘is’` ↔ `was`, `‘are’` ↔ `were`

- Short forms
  Short forms are often used to speed up the dictation process. They act as a kind of code between dictating person and transcriber (cf. speaker errors in section 2.2.1). In the final report these short forms have to expanded again according to official transcription style guidelines to ensure “clarity of communication” (cf. [15], p. 2). The number of used short forms is limited, but each of them is used frequently.
  E.g., `‘O two sat’` ↔ `oxygen saturation`
  `‘C section’` ↔ `cesarean section`
  `‘Afib’` ↔ `atrial fibrillation`

- Personal pronoun → ‘the patient’
  Another frequent observation is the reformulation of the personal pronouns `he` and `she` into the term “the patient” in the final report. Medical transcription style guidelines suggest that “within the body of a medical report, care should be taken to avoid mentioning personally identifying information.” (cf. [15], p.103). However, if proper names appear in a dictation, they are mostly not patient’s names but primary care physicians’ or hospital’s names. Automatic correction is, therefore, not trivial.

- Spellings
  Spellings account for approx. 1% of the classified deviations. Usually, the spelled term is pronounced regularly before the actual spelling (e.g., `‘clindamycin C L I N D A M Y C I N’` ↔ `clindamycin`). During post-processing, the spelled letters are mostly just removed, unless recognition errors were involved. As expected, the spelled entities are almost exclusively singletons, meaning that they only occur once in the data.

2.4 A paragraph from a sample medical report

To illustrate the differences between the text sources, a paragraph from a sample medical report is printed verbatim in figure 2.6. The selected paragraph is the description of the physical examination of a patient in a hospital’s emergency outpatient unit. This part of the medical report is typically highly standardised in the information it has to give and, therefore, often corrected by the medical transcriptionists to conform to these standards. At the same time, it is often uttered with high speaking rate, which makes it difficult to recognise automatically.

Manual orthographic transcription

on physical exam he was alert afebrile pulse ox was ninety four percent head ears eyes nose and throat EOMI PERRLA ??? clear TMs normal oropharyngeal mucosa is clear neck is supple no ??? JVD heart regular rate rhythm chest clear to auscultation percussion any wheezes rubs or rhonchi abdomen is soft without any masses pain guarding rebound bowel sounds are normal extremities showed no cyanosis clubbing or edema neurological exam is intact with no obvious deficits period [ahm] repeat [ahm] repeat auscultation of the chest [ahm] revealed wheezes throughout the lung bases period [ahm] patient under use accessory muscles auscultation

Recognised text (draft transcript)

of physical exam is alert comma afebrile period pulse ox ninety-four percent as of thirty-one apparently since her TMs normal period mucosa is clear period negative abscess JVD or regular rhythm period CHEST: past period discussion was resolved rhonchi period ABDOMEN: softly masses period no bowel sounds normal period duration cyanosis comma clubbing or edema period neurological exam is intact with no obvious tests period [ahm] repeat [ahm] repeat auscultation of the chest reveals wheezes throughout [ahm] lung bases period [ahm] the patient had a cystocele source period duration

Final medical report

</PHYSICAL EXAMINATION/>
He was alert, afebrile, pulse ox was 94%. HEENT: EOMI. PERRLA. TMS normal. Oropharyngeal mucosa is clear. Negative nodes or masses, JVD. Heart is regular rate and rhythm. Chest is clear to auscultation and percussion. No wheezes, rales or rhonchi. Abdomen is soft without masses, pain, guarding or rebound. Bowel sounds are normal. Extremities showed no clubbing, cyanosis or edema. Neurological exam is intact with no obvious deficits. Repeat auscultation of the chest revealed wheezes throughout the lung bases. The patient did not use any accessory muscles for auscultation.

Differences between recognised text and final medical report

~~of~~ *</*PHYSICAL EXAM*INATION/>*
~~is~~ *He was* alert, afebrile, pulse ox was 94%. ~~as of thirty-one apparently since her~~ *HEENT: EOMI. PERRLA.* TMS normal. *Oropharyngeal* mucosa is clear. Negative ~~abscess~~ *nodes or masses,* JVD. *Heart is*~~or~~ regular *rate and* rhythm. Chest ~~past period discussion was resolved~~*is clear to auscultation and percussion. No wheezes, rales or* rhonchi. Abdomen *is* soft~~ly~~ *without* masses*, pain, guarding or rebound*. Bowel sounds *are* normal. *Extremities showed no clubbing,* ~~duration~~ cyanosis ~~clubbing~~ or edema. Neurological exam is intact with no obvious *deficits*~~tests~~. Repeat auscultation of the chest revealed wheezes throughout *the* lung bases. The patient ~~had a cystocele source period duration~~*did not use any accessory muscles for auscultation*.

FIGURE 2.6. A paragraph from a sample medical report as it is represented in the various medical transcription text types: red parts were deleted from the recognised text and blue parts were inserted to obtain the final medical report.

The manual orthographic transcription contains a number of acronyms (`EOMI`, `PERRLA`, `TMs`, `JVD`), hesitations (`[ahm]`) and parts that were not understandable by the transcriptionist (`???`). Punctuation is missing completely, as well as accompanying document formatting.

The draft transcript contains recognition errors, particularly for short words or longer phrases along with segmentation errors. Still, it is possible to recover almost all information that was given in the dictation. The capitalised words `CHEST:` and `ABDOMEN:` are recognition lexicon entries with integrated formatting and not the result of an automatic formatting attempt by the ASR system.

The final medical report has changed notably compared to the original orthographic transcript. The paragraph heading was reformatted, and punctuation was inserted by the medical transcriptionist. Numbers were digitised and hesitations removed from the draft transcript. Apart from these obvious changes, the text was also considerably reformatted which can be seen in the last part of figure 2.6.

2.5 Medical Dictation text corpora

2.5.1 MEDTRANS: Phonetically transcribed medical dictation corpus

The MEDTRANS corpus is a set of orthographically and phonetically transcribed dictations of medical reports with corresponding audio recordings. The phonetic transcriptions of these recordings were produced in terms of this thesis to gain insights into particular problems of ASR, since plain orthographic transcriptions are not capable of delivering this information. Furthermore, the transcribed material provides a valuable reference for the development and evaluation of the phonetic edit distance measures in chapter 3. Form and level of detail of the phonetic transcription was defined according to these goals.

Corpus structure

The structure of the corpus is determined by the data resources from which the corpus was compiled and the transcription process itself. The first part of the corpus (subset A) is composed of real world recordings from an installed ASR-supported medical transcription environment, i.e., documents produced according to the ASR-supported workflow (cf. figure 2.2). The second part of the corpus (subset B) consists of medical reports and recordings that were produced according to the traditional workflow (cf. figure 2.1) with draft transcripts produced in a separate offline post-production process. It was an essential requirement to have both, manual orthographic transcription and an already recognised text at hand for generating automatic phonetic transcriptions (APTs). The APTs were intended to minimise transcription efforts and ensure consistency among transcribers. Speakers and reports were selected according to the subjective quality of dictation in terms of speed, intelligibility, acoustic quality, and usage of hesitations.

Each report was transcribed by a single phonetic transcriber except for 18 reports from two speakers which had accidentally been transcribed by at least two transcribers independently from each other. These transcriptions were collected in a third subcorpus (subset C) which was used for calculation of transcriber labelling agreement. Table 2.4 shows the key figures for each of the resulting 3 data subsets.

Table 2.4. Key figures for data subsets A, B, and C of the MEDTRANS corpus.

	A	**B**	**C ⊂ B**	**A ∪ B**
speakers (male/female)	11 (11/0)	19 (15/4)	2 (0/2)	30 (26/4)
reports	102	170	40	312
total length (h:min:s)	5:26:41	8:34:32	1:47:52	15:49:05

Transcription process

The reports were transcribed by 9 English students with experience in English phonetics under supervision of an expert phonetician. Training included basic instruction on how to use the transcription software *ELAN* [130], introduction to the domain of dictated speech, and a joint transcription of a sample report to establish a common transcription style. During transcription, remaining ambiguous cases were discussed in groups together with the supervisor. Apart from the phonetic transcription, also phonetic deviations observed by the transcribers were annotated according to a pre-defined set of annotation deviation categories. The phonetic symbol inventory and annotation deviation category set are shown in table 2.5 and in appendix A, table A.1.

From the manual orthographic transcriptions and the triphone acoustic models of the ASR engine, APTs were created with an HMM-based forced alignment procedure beforehand which then only had to be corrected by the transcribers. In addition to a draft phonetic transcription the APTs also provided an audio segmentation on word level that facilitated counting of deviations in the transcriptions. The automatically determined word boundaries remained untouched during transcription, only segment labels were corrected.

After subset A had been finished, two major changes were implemented. First, the annotation deviation category set was extended by two categories (#34, #45) that were missing before. And second, the acoustical recordings from subset B were stretched using the software *Praat* [10] with the PSOLA method [82] by a factor of 1.3 to enhance their intelligibility. This step dramatically increased the efficiency of the transcribers: While for subset A one minute of audio material took about 60 minutes to transcribe, the transcription time of the same amount of audio data reduced to 35 minutes for subset B.

Evaluation

The transcriptions were evaluated with respect to transcription labels and annotation deviation category labels. In this part, the differences between automatic and manual phonetic transcriptions are studied and summarised. Furthermore, the transcription quality was validated by determining transcriber labelling agreement.

Transcription labels & annotation deviation categories

The overall statistics are similar for each of the data subsets as shown in table 2.6. Subset A accounts for about 40% of the data, and subset B for about 60%. Around 82% of the total number of audio segments had a speech transcription, the rest were segments containing non-speech like silences or parts marked acoustically not useful. A small number of segments were transcribed without having an initial orthographic form. This number is probably even higher, as it does not take into account those segments that already had an (incomplete) orthographic form.

TABLE 2.5. Annotation deviation categories of the MEDTRANS corpus with examples in SAMPA notation. Categories #34 and #45 were introduced after the transcription of subset A.

#	Description	Automatic	Corrected
11	schwa deletion	`/n oU t @ d/`	`/n oU t @ d @/`
12	vowel deletion	`/s i t/`	`/s i t i/`
13	consonant deletion	`/m oU s l i/`	`/m oU s d l i/`
14	schwa insertion	`/p A s @ b l=/`	`/p A s b l=/`
15	vowel insertion	`/f A l oU V p/`	`/f A l oU p/`
16	consonant insertion	`/p eI S n= t/`	`/p eI S n=/`
21	diphthong instead of monophthong	`/k oU l d/`	`/k O l d/`
22	monophthong instead of diphthong	`/Q l s O/`	`/Q l s oU/`
23	schwa instead of full vowel	`/h e r oU @ n/`	`/h e r oU I n/`
24	full vowel instead of schwa	`/m e d I k l=/`	`/m e d @ k l=/`
25	voiced	`/l e v l= z/`	`/l e v l= s/`
26	unvoiced	`/p r e z n= t/`	`/p r e z n= d/`
27	vowel lengthening marker added	`/n O: r m l=/`	`/n O r m l=/`
28	vowel lengthening marker removed	`/I m p r u v d/`	`/I m p r u: v d/`
29	wrong vowel	`/p r A b l @ m z/`	`/p r V b l @ m z/`
31	American English (AE) transcription	`/t O: n s @ l/`	`/t A n s L/`
32	British English (RP) transcription	`/s d & d @ s/`	`/s t Y t @ s/`
33	error in proper noun transcription	`/b r U k f i l= d/`	`/b r U k f @ l= d/`
34	consonant substitution	`/f O l Y t/`	`/f O l Y k/`
41	wrong orthographic transcription	`/T 3 r t i/`	`/T r i/`
42	orthographic w.o. phonetic transcription	---	---
43	undefined transcription symbol used	`/o r T @ p i d I k s/`	`/O r T @ p i d I k s/`
44	unintelligible – word partially audible	`/d @ v e l @ p I N/`	`/v e l f I n/`
45	unintelligible – nothing audible	`/k @ n t I n j u d/`	---

Out of the segments having a transcription, about 23% deviated from the automatically generated form in at least one symbol. Most of those segments contained just one deviation, while only a very limited number (> 0.5%) contained three or more deviations. The maximum number of observed deviations was five.

The highest number of deviations were removed lengthening markers (around 25%). Consonant insertions account for the second largest number of deviations, following with 12%. The same holds for schwa deletion (cat. #11). Another prominent observation is the conversion from voiced to unvoiced phones (cat. #26). Together, these four observations sum up to already more than 50% of all annotations in all data sets. The detailed statistics for each annotation deviation category are shown in appendix B, tables B.3 and B.4.

The ranking of the remaining categories differs significantly for each of the data set. While, e.g., full vowels instead of schwa (cat. #24) are very prominent with 9.87% in subset A, the same category was observed in only 2.14% of the cases in subset B. Also worth mentioning are the differences for categories #31 and #32, concerning British and American English transcriptions (4.69/4.22% vs. 0.68/0.09%). Finally, there is also a notable difference with categories #44 and #45, describing acoustically unintelligible segments (4.76/0.82% vs. 2.17/1.28%).

Table 2.6. MEDTRANS transcription label statistics per subset.

	A		**B**		**C ⊂ B**		**A ∪ B**	
	Count	[%]	Count	[%]	Count	[%]	Count	[%]
Total segments	62,772	100.00	89,010	100.00	20,249	100.00	151,782	100.00
... w. initial transcription	51,662	82.30	73,686	82.78	16,765	82.79	125,348	82.58
... w.o. initial transcription	11,110	17.70	15,324	17.22	3,484	17.21	26,434	17.42
Deviating segments	15,097	29.22	19,269	26.15	3,858	23.01	34,366	22.64
1 deviation	12,639	24.46	15,904	21.58	3,299	19.68	28,543	18.80
2 deviations	2,145	4.15	2,842	3.85	495	2.95	4,987	3.29
3 deviations	259	0.50	417	0.56	56	0.33	676	0.44
>3 deviations	54	0.10	105	0.14	8	0.05	159	0.10
Different deviations	17,927		23,286		4,490		41,213	

Discussion

The following list of observations highlights some interesting figures that were found in the results and is not meant to be a complete listing.

- Schwa deletion (#11)
  The deletion of the schwa sound may be explained to a large extent with the Philips phonetic transcription guidelines which prescribe the omission of schwa in case of syllabic `/l/`, `/n/`, `/m/`, and `/r/` [93]. Schwa insertion (#14) only plays a minor role.

- Schwa substituted by full vowel (#24)
  Interestingly, this vowel substitution occurs very often in subset A (9.87%), but much less frequently in subset B (2.14%). This observation is confirmed by the exactly opposite configuration for the inverse category #23 (2.13% vs. 6.92%). Therefore, it is very likely that this observation is related to the stretching of the recordings in subset B and, thereby, the increased vowel quality perception in fast speech passages.

- Voiced – unvoiced conversion (#25, #26)
  Conversion of a voiced to an unvoiced part of a segment occurs quite frequently in all data sets (7.7%), while the inverse conversion happens rather rarely (1.52%).

- Vowel lengthening markers removed (#28)
  This occurs due to canonical pronunciations in the lexicon or the imprecise conversion between the phonetic alphabets. Interestingly, the inverse observation (#27: vowel lengthening marker added) occurs much less frequently (3.76%) which is a strong indicator for the conversion problem.

- Deviations due to British or American English (#31, #32)
  The phonetic lexicon used for APT is built from American English and British English resources such that it can be used in both language environments. Therefore, the APT may be ambiguous in some cases and confusions may occur easily. Interestingly, these deviations have mostly been observed in subset A, i.e., in the beginning of the transcription process. Since these deviations are also covered by other categories it seems that the phonetic transcribers did not use these labels consistently afterwards.

- Wrong orthographic transcription (#41)
  At first sight, the quality of the orthographic transcriptions appears to be poor, considering the relatively high frequency of this category (4.77%). A closer look, however,

revealed that a lot of hesitations are involved here. Furthermore, some transcribers seemed to use this category inflationary even for correct segments. Further analysis would be necessary to determine the real causes. Mismatches like the one given in table 2.5, however, are not as frequent.

- Unintelligible (#44, #45)
  In terms of intelligibility the quality of subset B was better than that of subset A due to the automatic stretching of the recordings and a more careful pre-selection of speakers for transcription. Therefore, the frequency of unintelligible segments could be significantly reduced which is reflected in these measures.

Transcriber labelling agreement

Consistency in labelling is an important indicator for the quality of a phonetic transcription [25], [20], [94]. Since the phonetic transcriptions were made by more than a single transcriber, it is essential to determine the transcriber labelling agreement. Otherwise, it is not clear whether the observed deviations in the transcriptions represent phonologic variation or just 'transcription noise'.

Usually, labelling consistency is measured by determining the pairwise agreement of transcribers for a certain subset of the corpus that has been transcribed by all transcribers. This procedure is not directly applicable to this corpus, as there is no report which has been transcribed by all transcribers. Still, subset C allows the determination of pairwise transcriber agreement on different texts. Table B.5 in appendix B shows the mapping between transcribers and reports for subset C. Altogether, 9 transcribers annotated 18 different reports such that there were 26 transcriber pairings with 10 of them being distinct.

Transcriber agreement itself was calculated in two ways. First, by counting the number of common phone labels divided by the total number of labels for each pairing (% agreement or $P(a)$). And second, by determining Cohen's κ coefficient[19] which measures the agreement between two raters who each classify N items into C mutually exclusive categories. This measure is more robust than simple percent agreement calculation since it takes into account the agreement occurring by chance. Assuming that the labelling events are collected in a $C \times C$ confusion matrix $G = \{g_{ij};\ i, j = 1..C\}$, the κ-coefficient is determined by

$$\kappa = \frac{P(a) - P(e)}{1 - P(e)} \tag{2.1}$$

where $P(a)$ is the relative observed agreement among raters, and $P(e)$ is the probability that agreement is due to chance, defined by

$$P(a) = \frac{1}{N}\sum_{i=1}^{C} g_{ii} \quad \text{and} \quad P(e) = \sum_{c=1}^{C} \frac{\sum_{j=1}^{C} g_{cj}}{N} \frac{\sum_{i=1}^{C} g_{ic}}{N}\,. \tag{2.2}$$

These agreement scores were calculated for the whole symbol set and for five phonetic classes of symbols (stops, fricatives, liquids, nasals, and vowels). Since vowels have an optional lengthening marker, agreement was determined with and without regard of the lengthening marker. The resulting agreement scores are listed in table 2.7.

The transcriber agreement scores are comparable to those of other phonetic corpora transcribed at similar level of detail [127], [94]. The percentage agreement is significantly higher than the previously measured correspondences between automatic phonetic transcription and

Table 2.7. MEDTRANS transcriber agreement measured in percent agreement and Cohen's κ for all labels and divided into phonetic classes in subset C.

Phonetic class	% agreement	κ	N
stops	92.35	0.9102	12,933
fricatives	97.06	0.9650	7,214
liquids	91.60	0.8902	5,298
nasals	93.12	0.8823	5,013
vowels (w.o. lengthening)	89.33	0.8812	15,271
vowels (w. lengthening)	81.69	0.7981	15,271
overall	89.11	0.8870	40,758

manual phonetic transcription (89% in table 2.7 vs. 77% in table 2.6). For this reason, the deviations observed in the corpus are not only the result of the manual transcription process, but also of actual phonologic variations.

2.5.2 MEDALIGN: Multi-alignment medical dictation corpus

For analysis and processing of the various text types, a text corpus is needed which integrates all text formats in such a way that relations between sub-units are established flexibly. Assuming that correspondences and deviations are local, data analysis algorithms can access relevant portions of each text format without extensive search. The integration task is therefore a text alignment step. In this case, however, not only two independent text sequences need to be aligned, but at least three which poses a problem for representation. Instead of a single alignment label, two alignment labels need to be calculated for documenting the relation between entities. The resulting alignment will be referred to as *multi-alignment*.

Multi-alignment procedure

For the medical dictation data described in this chapter a multi-alignment procedure was provided by Philips Speech Recognition Systems. This procedure is based on the idea that the text types can be separated into time-stamped data where text segments are linked to a specific segment in the audio recording and continuous text data that is neither segmented nor directly related to the audio. Note that this distinction is not the same as for literal and non-literal transcripts. Time-stamped texts are the recognised text and the manual phonetic transcription, while continuous texts are the manual orthographic transcription and the final medical report. This difference in tokenisation and synchronicity holds some implications for the multi-alignment. First, an atomic tokenisation with respect to the recognition lexicon may help resolving hard cases for a tokeniser within the final medical report. And second, a kind of consistency constraint ensures that segments of one text cannot "overtake" the other text that has hard evidence from the audio.

Starting from these assumptions, the multi-alignment is done from two elementary two-sided alignments as shown in figure 2.7. Alignment A is a semi-automatic alignment of the manual orthographic transcription with the final medical report, while alignment B is an automatic alignment of the manual orthographic transcription and the draft recognised transcript. The alignments are then merged based on the common manual orthographic transcription with respect to the two previously mentioned constraints.

This way of aligning multiple texts has several advantages. It is fast, as the simple alignments are based on word level and the merging step operates more locally than globally.

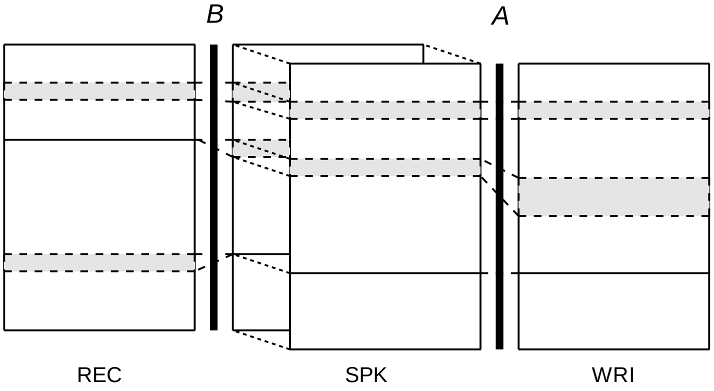


FIGURE 2.7. Multialignment process between manual transcriptions (SPK), draft transcripts (REC), and final corrected medical reports (WRI). Two alignments (A: SPK ↔ WRI and B: REC ↔ SPK) are merged based on the common SPK transcript, such that REC, SPK, and WRI are synchronised. Therefore, gaps must be opened in alignments A and B wherever necessary.

Furthermore, it allows for simple extension by integrating further text types, if they are directly related to one of the primary text types. For the application presented in chapter 4, up to 17 different text types were aligned, including information such as word type information or different representations of text formatting.

Challenges in multi-alignment

The alignment scheme is too simple for the complex input containing formatting and major differences in representation of the various text deviations described in sections 2.2 and 2.3. Two prominent challenges in multi-alignment shall be highlighted in the following.

The major shortcoming of this multi-alignment approach is the inconsistent tokenisation that results in segmentation problems and local alignment inconsistencies. Figure 2.8 illustrates how recognition errors induce such mismatches and break the correspondences between the individual text types. A solution to this problem would need to hypothesise at which position longer words may be split into shorter fragments such that sub-word correspondences may be established between the compared strings.

Another shortcoming of the multi-alignment procedure are obvious mismatches in the alignment as shown in figure 2.9 with the words `auscultation` and `consultation`. The multi-alignment procedure between non-timed text types is based on Levenshtein alignment (cf. chapter 3, section 3.3.1) which assigns equal costs for substitutions, no matter how similar or dissimilar they are. An improved multi-alignment must establish correspondences between non-identical strings in a content-sensitive way. In section 4.3 of chapter 4, a more elaborate alignment procedure is presented that remedies these problems.

Data sets

With the help of the multi-alignment procedure a number of data sets were compiled. The label MEDALIGN only refers to the common method for creation, because each of the derived data sets was compiled for a specific purpose. Table 2.8 gives an overview on the key figures of each set.

REC	↔	*SPK*	↔	*WRI*
`Atrovent`	`COR`	`Atrovent`	`COR`	`Atrovent`
			`INS`	`,`
`neuro`	`SUB`	`spironolactone`	`COR`	`spironolactone`
`lactone`	`DEL`			
`<hes>`	`DEL`			
			`INS`	`,`
`and`	`COR`	`and`	`COR`	`and`
`lipids`	`SUB`	`Lipitor`	`COR`	`Lipitor`
`for`	`DEL`			
			`INS`	`.`

Figure 2.8. Deviations in tokenisation observed in multi-alignments induced by recognition errors: matches(COR), substitutions (SUB), deletions (DEL), and insertions (INS).

REC	↔	*SPK*	↔	*WRI*
`dietary`	`COR`	`dietary`	`COR`	`dietary`
`<hes>`	`SUB`	`-AM`	`DEL`	
`<hes>`	`SUB`	`consultation`	`COR`	`consultation`
`auscultation`	`SUB`	`he`	`COR`	`he`
`with`	`SUB`	`received`	`COR`	`received`
`some`	`SUB`	`while`	`COR`	`while`
`blood`	`DEL`			
`in`	`COR`	`in`	`COR`	`in`
`the`	`COR`	`the`	`COR`	`the`
`hospital`	`COR`	`hospital`	`COR`	`hospital`

Figure 2.9. Alignment mismatches in the multi-alignments induced by content-insensitive alignment: matches (COR), substitutions (SUB), deletions (DEL).

Table 2.8. Key figures for the MEDALIGN corpus data sets: PROFILE for manual inspection and text resource analysis, WERBAL for experimentation with balanced word error rates, NNS102 for evaluation of non-native speakers, and INSPECT for adaptation. Note that for some of the dictations in the PROFILE and NNS102 data sets, recognised texts were not available.

	PROFILE	**WERBAL**	**NNS102**	**INSPECT**
speakers (male/female)	60 (48/12)	283 (224/59)	102 (102/0)	434 (434/0)
reports	630	735	758	3,453
total length [h:min:s]	36:32:55	47:54:06	60:36:16	221:33:54
SPK words	312,424	335,474	445,798	1,573,024
REC words	216,037	381,738	182,995	1,779,300
WRI words	270,937	328,193	347,199	1,523,161

The PROFILE data set was meant for the quantitative analysis of text resources presented in this chapter. The WERBAL set was composed as an evaluation data set covering a wide range of speakers. The reports were selected to fit three balanced word error rate ranges (low, medium, and high) and to avoid outliers. The NNS102 data set comprises only non-native or accented speakers and was designed for training and testing of the pronunciation modelling approach in chapter 5. The INSPECT set is used for acoustic adaptation and for large-scale analysis of potential ASR errors. NNS102 and INSPECT are fully disjoint to ensure the proper separation of acoustic and pronunciation model adaptation data.

2.6 Conclusion

Within the ASR-supported medical transcription workflow several different text resources are generated that may be utilised for improving an ASR system. There are major differences between these text types, and each type adds special benefits that can be exploited. Manual transcriptions are not only the true reference for training the acoustic models of the ASR system, but also document speaker errors that need to be corrected somewhen in the transcription workflow. Draft transcripts from the ASR system contain speech recognition errors which become visible in direct comparison to the manual transcriptions. The final reports give insights into the editing operations done by the medical transcriptionists for either correcting speaker errors, or ASR errors, or for formatting the dictation into a standardised medical report.

Working with many text types with significant differences among each other turned out to be a challenging task. To exploit the full range of information that is given in a medical dictation, a method for alignment of multiple texts must be developed. Pairwise comparison of the texts reveal deviations, but only the combination of multiple alignments into a single multi-alignment synchronises the individual text types and provides a systematic approach on text analysis. The analysis of deviations presented in this chapter indicates that the deviations range from whole words over syllables down to single phones. The first attempt for multi-alignment based on time-synchronisation used for compilation of the MEDALIGN corpus only provides rough correspondences between the texts. The level of detail in the segmentation is too coarse to account for a proper matching. The main conclusion of this chapter is therefore that an accurate alignment must establish correspondences on multiple levels of segmentation. An algorithm that implements such a multi-level alignment is presented in chapter 4.

The findings of this chapter suggest the phonetic domain for processing the various literal and non-literal transcript types. The analyses hold the following implications:

- The comparison of short phone sequences is sufficient as variation usually occurs within units not larger than a syllable.
- Some phone positions in a word are less or not at all affected by variation like syllable onsets in contrast to the syllable nucleus and coda which are highly vulnerable. This is another strong argument in favour of the syllable as structuring unit for phonetic similarity matching.
- Not all phones are equally affected by pronunciation variation, thus phone-specific weights appear promising to model the phonologic variation.
- Some of the available cues like syntactic or semantic information cannot be covered by phonetic measures. Nevertheless, the integration of such a priori knowledge sources may be beneficial for disambiguating difficult cases.

Chapter 3

Similarity, Dissimilarity, and Confusability in ASR

In everyday language, the terms *similar* and *dissimilar* are used to characterise the relation between two objects in an informal way. Whenever either only a quick, approximate judgement is desired, or whenever the object properties are difficult to describe and measure, people refer to such vague descriptions. Although the exact transition between 'similar' and 'dissimilar' is undefined, the meaning of this judgement is usually clear. In contrast to the subjective human judgement, an objective comparison of two objects must clearly define the ranges for similarity and dissimilarity, and if clear, deterministic decisions are to be made. Setting the threshold itself is just an art as defining an appropriate similarity measure.

Directly related to the problem of similarity measurement is the problem of confusability. Classification methods are based on the assumption that the similarity of an actual data sample and a previously recorded pattern can be determined and appropriately expressed in a single figure. The difficulty of finding the similarity threshold is solved for classification tasks such that data samples that are closely related to the prototyping pattern will be assigned the same label, for each prototype. This way, dissimilarity automatically starts whenever the similarity to a different prototype is higher, thus the similarity threshold is implicitly set. This concept is a good application for objective similarity measurement, but while it allows for a certain amount of variation (or generalisation), there are limits to its performance. The effectiveness of this approach strongly depends on how well the similarity function is suited to distinguish between the prototypes, or in other words, how much the prototyping patterns are alike themselves. Ideally, their distances should be equal among each other and much larger than the distances to their various realisations to allow for a clear comparison. Otherwise the blessing turns into a curse as data samples are incorrectly classified or *confused*.

The goal of this chapter is to find an appropriate similarity measure for automatically assessing the deviations between literal and non-literal transcripts that were described in chapter 2. The findings there suggest the phonetic domain for processing the various text types. Such a *phonetic similarity measure* would be beneficial in two respects: First, it could lead to an enhanced alignment in comparison to the multi-alignment presented in section 2.5.2. And second, it would allow to interpret the deviations with respect to the common source to which all text sources are related: the audio recording of the dictation. In other words, a phonetic similarity measure could estimate how plausible it is that two compared text parts originate in the same audio segment of a dictation.

Rabiner and Juang already argue in their description of pattern comparison techniques that "correct time alignment between two utterances of different words is not a well-defined linguistic concept" (cf. [98], section 4.7, p. 225). While this a legitimate concern for time-aligning acoustic audio signals, it is less of a problem for aligning literal and non-literal phone symbol strings, which are used throughout this chapter. Furthermore, the chosen pronunciation classification task interprets the similarity problem as a confusability problem, which makes it ideal for comparing algorithms.

The chapter is structured as follows: The first part is a formal definition of the problem together with a review of selected state-of-the art similarity measurement algorithms in language processing. With respect to the current literature, the methods are divided into state-based and symbol-based approaches. The second part of this chapter presents a short evaluation of these methods on a publicly available data set and application-specific data. Based on this evaluation, arguments for an appropriate phonetic similarity measure of literal and non-literal transcripts will be given together with suggestions for application-specific extensions.

3.1 General definition and notation

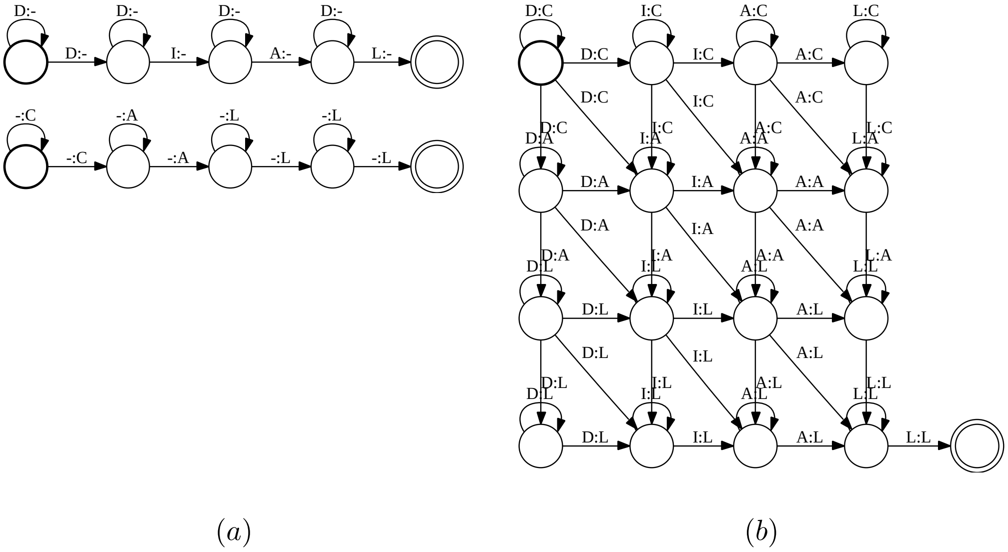
In the literature on sequence matching various method-specific definitions and illustrations have been given for describing the problem (e.g., traces in [125], or paths in [122]). At this point, however, more general definitions will be introduced that even fundamentally different sequence comparison approaches and algorithms can be described within the same notational framework.

For determining similarity two interlinked problems need to be solved. First, an *alignment* between the input sequences has to be established. With the proper correspondences between the single symbols within the sequences, a *scoring scheme* can be applied either as a by-product of the alignment procedure, or in a separate step. Formally, this requires two definitions:

Definition 1 *Let $x_1^N = \langle x_1, x_2, ..., x_N \rangle$ (short: x^N) be a sequence of symbols $x_i \in \mathcal{X}$ of length N, and $y_1^M = \langle y_1, y_2, ..., y_M \rangle$ (short: y^M) a second sequence of symbols $y_j \in \mathcal{Y}$ of length M, where in general $M \neq N$. An $\underline{\text{alignment}}$ Λ of x^N and y^M is a sequence of paired symbols $\Lambda = \langle (x_i, y_j) \rangle$ for all $i = 1..N$, $j = 1..M$, such that each symbol pairing occurs only once.*

Definition 2 *A $\underline{\text{score}}$ is the real-valued cost c assigned to a single input symbol pair: $\mathcal{X} \times \mathcal{Y} \to \mathbb{R} : (x_i, y_j) \to c$. A $\underline{\text{scoring scheme}}$ takes all the scores of an alignment $\Lambda = \langle (x_i, y_j) \rangle$ to produce a total score $d(\Lambda) = d(\langle (x_i, y_j) \rangle)$.*

Note that these definitions do not give any indication to how the actual alignment procedure should work or to the type of input symbols. Depending on the understanding of the alignment procedure and the symbol alphabets $\mathcal{X}$ and $\mathcal{Y}$ (continuous or discrete, uni- or multivariate), *state-based* models and *symbol-based* models for sequence alignment can be dichotomised. Although the foundations are similar for both approaches, they are fundamentally incompatible, but there are attempts for combining the two ideas [24]. In the following sections, the approaches will be introduced in more detail and for each of them a selection of state-of-the art algorithms will be presented.



(*a*)

(*b*)

FIGURE 3.1. State-based similarity approach: Input sequence representation as (a) finite state transducers and (b) similarity graph as formal composition $x^N \circ y^M$ of the transducers.

3.2 State-based approach

The state-based approach to sequence alignment is best viewed from the perspective of a deterministic finite state transducer (FST) (cf. section 3.4 in [54]). The input symbol sequences are each represented as an FST with states $S = \{s_1, s_2, ..., s_{\{N,M\}+1}\}$, labelled state transitions E, and input alphabets $\mathcal{X}, \mathcal{Y}$ as depicted in figure 3.1. One of the transducers acts as an acceptor with state transitions $E = \{(s_i \rightarrow s_i, x_i : \epsilon), (s_i \rightarrow s_{i+1}, x_i : \epsilon) \mid i = 1..N\}$ that consumes the symbol sequence as its input without producing output ($\langle$D,I,A,L$\rangle$ in figure 3.1), while the other transducer acts as a generator with state transitions $E = \{(s_j \rightarrow s_j, \epsilon : y_j), (s_j \rightarrow s_{j+1}, \epsilon : y_j) \mid j = 1..M\}$ which generates the symbol sequence at the output from no input ($\langle$C,A,L,L$\rangle$ in figure 3.1). The formal composition $x^N \circ y^M$ of acceptor x^N and generator y^M results in a trellis-structure that describes all possible alignments of the input sequences. By traversing the trellis-structure, the sequences are processed synchronously, as for each time step, the model either remains in the current state, or it changes to one of the neighbouring states. The alignment is then the sequence of traversed edges. In the state-based approach the information about the current position within the alignment is defined by the *states* of the model and not by the input symbols. In figure 3.1, the symbol sequences $\langle$D,I,I,I,I,A,A,L,L,L$\rangle$ and $\langle$C,C,C,A,A,A,L,L,L,L$\rangle$ would just as well be accepted. As a consequence of the separation of sequence transduction and underlying data representation, the state model allows for discrete, continuous, and even multivariate data ranges.

Due to the formal FST composition of generating and accepting edges, there are no more empty symbols in the resulting alignment. For this reason, this approach is suited whenever there is no intuitive notion of an empty symbol, e.g., when processing synchronous feature sequences obtained from frame-wise processing of audio signals with dynamic time warping.

3.2.1 Continuous range

Dynamic Time Warping (DTW)

The dynamic time warping algorithm was originally introduced to meaningfully compare two realisations of the same utterance with possibly varying speaking rate or speech sound deviations [98]. For this task, the input signals are windowed and usually short-time spectra are calculated such that a spectral distortion measure can be applied as scoring scheme. The symbol alphabets are therefore $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^d$ with d being the number of spectral features. The spectral distortion measure incorporates heuristics such as monotonicity, local continuity, endpoints, global path, or slope weighting constraints, according to the domain of the investigated signals. Signal alignment and time-normalisation is achieved by a dynamic programming scheme which uses the local spectral distortion measure to achieve a globally optimal alignment.

DTW is useful whenever audio signals are directly compared to each other on a frame-by-frame basis and no indirect symbolic representation is available. Applications are, e.g., template-based automatic speech recognition [124] or example-based automatic phonetic transcription [67].

Acoustic confusability

The theory of acoustic confusability developed in [96] and refined in [46], [47], and [16] is another good example for the state-based approach to sequence comparison. The basic idea is to derive the acoustic similarity between two words directly from already existing, well-trained parameter distributions, i.e., the acoustic model of an ASR system which is capable of recognising these words. Current acoustic models are based on Hidden Markov Models (HMMs), where each HMM represents a context-independent phone or context-dependent triphone with three or more states, that follow a non-ergodic left-to-right sequence topology (i.e., diagonal state transition matrix) [54], [131]. The observations are modelled with multivariate Gaussian Mixture Models (GMMs). To synthesise whole words from individual phone models, the similarity graph shown in figure 3.1 is extended to a weighted finite state transducer (WFST) by assigning the distance of the specified HMM pairs to the corresponding edges. Since there are no ϵ-edges in the similarity graph, but always symbol pair edges, the state-based approach is optimally suited for this task.

The computation of a distance or similarity between two HMMs is then a completely separated problem. To compare continuous probability densities, the measure of relative entropy or Kullback-Leibler divergence is an appropriate scoring scheme. For distributions x and y representing symbols in $\mathcal{X}, \mathcal{Y}$, it can be defined as follows:

$$h(x \in \mathcal{X} \| y \in \mathcal{Y}) = \int x(\xi) \log \frac{x(\xi)}{y(\xi)} \, d\xi. \tag{3.1}$$

For single Gaussian distributions there exist closed-form evaluations of equation 3.1 whereas for GMMs, there is no closed form or analytical solution to this problem and, usually, numerical methods like Monte-Carlo sampling have to be applied to approximate the solution. The authors test various alternative divergence measures for this task including:

$$\begin{array}{rcll} D_B(x,y) &=& -\log \int \sqrt{x(\xi)y(\xi)}d\xi & \text{Bhattacharyya divergence} \\ D_{MC}(x,y) &=& \frac{1}{K}\sum\limits_{k=1}^{K} \log \frac{x(\xi_k)}{y(\xi_k)} \rightarrow D(x \parallel y) & \text{Monte-Carlo sampling} \\ D_G(x,y) &=& \sum\limits_{a} \pi_a (D(x_a \parallel y_{m(a)}) + \log \frac{\pi_a}{\omega_{m(a)}}) & \text{Goldberger approximation [35]} \\ D_{min}(x,y) &=& \min\limits_{a,b} D(x_a \parallel y_b) & \text{Gaussian approximation} \end{array}$$

for Monte-Carlo samples k, GMM components x_a and y_b with weights π_a and ω_b, and a matching function $m(a)$ that relates components x_a, y_b to each other according to their similarity. The advantage of this model of acoustic similarity is its direct relation to the ASR system by evaluation of the acoustic model. This means that no extra data and no extra training procedures are necessary. Current acoustic models, however, are too complex for accurate and fast computation of phone similarities. Even with the proposed approximations, the model is only computable in reasonable time for monophone acoustic models and not for the usually implemented tied-state triphone acoustic models.

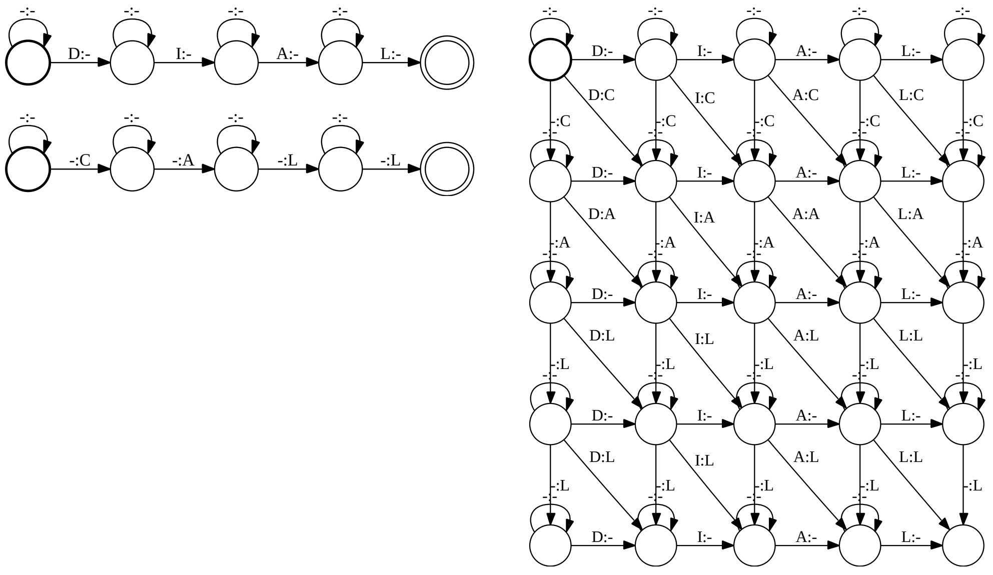
3.2.2 Discrete range

Confusion extraction from time-synchronously aligned transcriptions

Another application of the state-based approach is presented in [11]. There, the authors introduce a system for fully automated recognition of non-native speech by modification of the acoustic model with phone confusion information. For extracting this information, an alignment of the spoken language transcription and the speaker's native language transcription is necessary. In this case, the symbol alphabets are the target language phoneme set for $\mathcal{X}$ and the foreign language phoneme set for $\mathcal{Y}$. From this synchronous alignment, language specific confusion rules are extracted from the data, such that overlapping time segments are collected and related to each other in the form of maximum likelihood estimates. The strict time-alignment prevents the definition of an empty time interval and hence requires the application of the state-based approach.

3.3 Symbol-based approach

Staying with the FST representation, the input sequences in the symbol-based approach again follow the generator/acceptor paradigm. After each symbol, however, an arbitrary number of empty symbols may be processed as shown in figure 3.2. The composition of these input transducers therefore results in a similar lattice structure but with different state transitions. These transitions can be described as operations that edit the generator input sequence such that the acceptor sequence is produced. Thus, an alignment of two strings is a sequence of these so-called *edit operations* which in sum describe how one string is transformed into the other. In contrast to the state-based approach, the information about the current position in the alignment is encoded in the input *symbols* and not the states. Only the specified sequence of symbols is accepted by the input FSTs, so gaps may appear in the alignment that are



(*a*) (*b*)

FIGURE 3.2. Symbol-based similarity approach: Input sequence representation as (a) finite state transducer and (b) similarity graph as formal composition $x^N \circ y^M$ of the transducers.

labelled with the empty symbol ϵ which makes the symbol-based approach asynchronous. For this reason the symbol alphabets can only be discrete, which directly implies the term *string alignment*.

Originally, the symbol-based approach was developed for applications in communications to describe how a message changes when transmitted over a noisy channel (e.g., [69], [6]). Further domains of applications include bioinformatics (DNA sequencing), text processing (approximate string matching, spelling correction [22], [18]), or image processing (editing of contour paths). A thorough overview on string matching algorithms is given in [83].

The concept of string edit distance can be formalised in the following way: Starting from the general definitions given earlier, the string transformation process is broken down into a sequence of *elementary edit operations* $Z^l = \langle z_1, z_2, ..., z_l \rangle$. The commonly defined operations are *substitution* $z_{sub} = (x_i, y_j)$ (with *identity* $x_i = y_j$ as special case), *deletion* $z_{del} = (x_i, \epsilon)$, and *insertion* $z_{ins} = (\epsilon, y_j)$. Depending on the application, further operations can be defined as well. In speech processing, e.g., *compression* $z_{comp} = (\langle x_i, x_{i+1} \rangle, y_j)$ and *expansion* $z_{exp} = (x_i, \langle y_j, y_{j+1} \rangle)$ are often-observed phenomena [105], while *transpositions* $z_{trans} = (y_j, x_i)$, for instance, were introduced in [125] to model common typing errors in typed texts.

Mathematically, a proper edit distance measure has to fulfil the axioms of a metric:

$$\begin{array}{ll} d(x^N, y^M) > 0 & \text{non-negativity} \\ d(x^N, x^N) = 0 & \text{zero property} \\ d(x^N, y^M) = d(y^M, x^N) & \text{symmetry} \\ d(x^N, y^M) + d(y^M, z^L) \geq d(x^N, z^L) & \text{triangle inequality} \end{array} \tag{3.2}$$

In practice, however, many measures violate one or more of these axioms – in particular the symmetry and triangle inequality properties – while still returning useful results. Generally, two types of edit distance measures can be distinguished. *Deterministic measures* have fixed scores for each edit operation and hence produce a deterministic result for a given source/target string pair. *Stochastic measures* model the transformation from source into target string probabilistically. Each edit operation has a probability of occurrence assigned, and the edit distance is then defined as the sum over the probabilities of all possible edit sequences.

3.3.1 Deterministic measures

Levenshtein distance (*LEV*)

Levenshtein supposedly was the first to define a simple edit distance measure together with an algorithm for computing it [69]. Many other authors presented similar ideas and algorithms of comparable complexity (cf. [105], [125], [75]). The Levenshtein distance handles 4 elementary edit operations: *identity*, *substitution*, *deletion*, and *insertion.* Given a cost function c which assigns costs to each edit operation, the Levenshtein distance is calculated with the following recursive formula:

$$d_{lev}(x^n, y^m) = \min \begin{cases} c_{del}(x_n, \epsilon) + d_{lev}(x^{n-1}, y^m) \\ c_{ins}(\epsilon, y_m) + d_{lev}(x^n, y^{m-1}) \\ c_{sub}(x_n, y_m) + d_{lev}(x^{n-1}, y^{m-1}) \\ c_{ident}(x_n, y_m) + d_{lev}(x^{n-1}, y^{m-1})\,. \end{cases} \tag{3.3}$$

This means that for a given pair of subsequences $x^n \leq x^N$, $y^m \leq y^M$, the Levenshtein distance is the minimum cost of either:

- (x^{n-1}, y^m) plus the cost of a deletion,
- (x^n, y^{m-1}) plus the cost of an insertion, or
- (x^{n-1}, y^{m-1}) plus the cost of either an identity or a substitution operation.

Since the total distance only depends on the distance calculations of the input string prefixes, the overall distance can be calculated with a dynamic programming algorithm in $O(N \cdot M)$ time, the product of the sequence lengths. A faster algorithm is presented in [75] which runs in $O(N^2/\log(N))$ time for sequences which are both of length N. In [119] an exact algorithm is proposed which runs even in $O(d \cdot N)$ time, where d is the edit distance of the string pair.

In case of the standard Levenshtein distance, the cost for a substitution, deletion, or insertion operation is $c_{sub} = c_{del} = c_{ins} = 1$, while it is zero for the identity operation ($c_{id} = 0$). Therefore, the Levenshtein distance becomes the minimum number of substitutions, deletions, and insertions. For the so-called *Generalised Levenshtein Distance*, the cost function is not only dependent on the type of edit operation, but also on the input symbol sequences x^N and y^M (cf. [125]). An even more general scoring scheme was proposed in [13], by replacing the substitution cost with a single parameter r. This way, the computed distance becomes a function of the parameter r and hence postpones the exact definition of an optimal substitution cost for a subsequent classification task to the following classifier. Such an optimal classification approach is investigated in [85].

Normalised Levenshtein distance (*NLEV*)

One of the main problems of the Levenshtein distance is that the interpretation of the total cost depends on the lengths of the compared strings. Two short sequences related by a single correcting edit operation may be interpreted as being rather dissimilar, while two long sequences related by the same single operation may be considered very much alike. To simplify the interpretation, one could normalise the calculated distance by the sum of the input sequence lengths:

$$d_{nlev}(x^N, y^M) = \frac{d_{lev}}{N+M} \,. \tag{3.4}$$

This method is not optimal for the generalised Levenshtein distance as the range of output values is different for strings of approximately equal length and strings of unequal length. Therefore, a different normalisation method is proposed in [74] where only the edit path is considered for normalisation. A better way of normalising the Levenshtein distance would be based on the edit path Π as proposed in [74]. In a straight forward manner – the so-called *post-normalisation* – the edit distance result could be normalised by the length of the edit path Π leading to a minimum distance of

$$d_{nlev_{post}}(x^N, y^M) = \frac{\operatorname{argmin}_{\Pi} W(\Pi)}{L(\Pi)} \,. \tag{3.5}$$

The edit path Π is defined as the sequence of edit operations needed to transform the source into the target string. Its weight $W(\Pi)$ is then the sum of costs along the path and $L(\Pi)$ is the length of this path. Post-normalisation, however, is still not the most accurate way of normalisation as minimisation of the weights prior to normalisation is not necessarily equal to the minimum after normalisation. Therefore, the *path-normalisation* is based on the normalised path weights:

$$d_{nlev_{path}}(x^N, y^M) = \operatorname*{argmin}_{\Pi} \frac{W(\Pi)}{L(\Pi)} \,. \tag{3.6}$$

This normalised distance can be computed in $O(N \cdot M^2)$ time, as the length of the edit paths has to be considered in the calculation. A similar algorithm implementing the same idea, but running in $O(N \cdot M)$ is presented in [122].

While the previous algorithms do not fulfil the metric axioms, a true normalised Levenshtein distance metric was proposed in [132] together with a proof of the triangle equality:

$$d_{nlev_{metric}}(x^N, y^M) = \frac{2 \cdot d_{lev}(x^N, y^M)}{\alpha \cdot (N+M) + d_{lev}(x^N, y^M)} \,, \tag{3.7}$$

where $\alpha = \max\{c(x,\epsilon), c(\epsilon, y); x \in \mathcal{X}, y \in \mathcal{Y}\}$. The complexity of this algorithm is $O(N \cdot M)$ and comparable to the previous methods. Each of the discussed normalisation methods has its particular advantage. The choice for a specific method, however, depends on the application.

Generalised Levenshtein distance with phonetic feature weights

The Levenshtein distance was introduced as a robust and flexible method for expressing the differences between two symbol strings. The Generalised Levenshtein distance allows the incorporation of a priori information into the distance calculation, as the costs not just depend on the type of edit operation, but also on the actual symbols [125]. Therefore, the distance calculation becomes domain-dependent and more precise.

For the domain of phonetic distance calculation, several studies successfully proposed the usage of phonetic features or phonetic classes as a priori information [59], [108], [31], [45]. This information can either be obtained from confusability charts, e.g. from [78], or from the IPA chart [50] which categorises consonants according to their attributes place of articulation and manner of articulation, and vowels according to their attributes height and backness as shown in the vowel quadrangle. The actual phone feature class assignments for the various phonetic alphabets used in this work are listed in appendix A, table A.2. In the experimental evaluation presented in section 3.4, the best results were achieved with the substitution cost function

$$c_{sub}(x_n, y_m) = 1 - \frac{|\mathcal{F}\{x_n\} \cap \mathcal{F}\{y_m\}|}{|\mathcal{F}\{x_n\} \cup \mathcal{F}\{y_m\}|} \cdot \alpha \,, \tag{3.8}$$

where $\mathcal{F}\{x_n\}$ is the set of phonetic features for symbol x_n and α is an independent weighting or scaling factor. This means that phone pairs that share many features and are therefore easily confusable get lower substitution cost than phone pairs sharing only few phonetic features.

3.3.2 Stochastic measures

Several definitions for probabilistically modelling String Edit Distance have been given, ranging from generative models [6],[103], over Markov Random Fields [126], to discriminatively-trained Conditional Random Fields [76]. Common to all these approaches is a probabilistic scoring scheme where the scores are expressed as probabilities of occurrence. The following overview is based upon the ideas from [103] and extensions presented in [27]. In their works, the authors propose a generative model inspired by a stochastic transducer. Their idea is that a string pair can be represented by all sequences of edit operations that explain how the one string is transformed into the other. Assuming that each string pair is generated by at least one edit sequence, the probability of the string pair is then the sum of the probabilities of all edit sequences for that string pair.

More formally this means that one needs to determine the joint probability $P(x^N, y^M \mid \theta)$ of the source and target symbol sequences pair (x^N, y^M) given model parameters θ. The edit operations are modelled with a hidden two-dimensional random variable $Z_i : (\mathcal{X} \cup \epsilon \times \mathcal{Y} \cup \epsilon)$ indexed by $i = 1..\{max(N, M) \leq l \leq N + M\}$ to reflect the position within the edit sequence. The desired joint probability is now defined as

$$P(x^N, y^M \mid \theta) = \sum_{\{max(N,M) \leq l \leq N+M\}} \sum_{\{z^l \# : v(z^l \#) = (x^N, y^M)\}} P(Z^l = z^l, x^N, y^M \mid \theta) \,, \tag{3.9}$$

where $v(z^l\#)$ is the so-called *yield* of the terminated edit sequence $z^l\#$, which is the set of all terminated edit sequences of length l that produced (x^N, y^M). The special termination symbol # is introduced to make sure that the set $Z^*\#$ of all terminated, arbitrary-length edit sequences is prefix free and the above defined probability function is valid. In short, the joint probability is the sum over all edit sequences z^l of all possible lengths l that result in the given sequence pair.

From this joint probability two edit distance measures are derived. The so-called *Viterbi Edit Distance*

$$d_{MCI_{vit}} = -\log \underset{\{z^l : v(z^l) = (x^N, y^M)\}}{\operatorname{argmax}} P(Z^l = z^l, x^N, y^M \mid \theta) \tag{3.10}$$

only considers the most likely edit sequence for the given string pair, while the *Stochastic Edit Distance*

$$d_{MCI_{sto}} = -\log\ P(x^N, y^M | \theta) \tag{3.11}$$

accumulates the probabilities for all possible edit sequences. For both measures the calculated distance decreases exponentially with the edit operation sequence length and is never equal to zero for any string pair.

So far there has not been any comment on the probability of the single edit operations $P(Z_i = z_i, x^N, y^M \mid \theta)$ and on the interdependencies between the z_i's. By defining these relations different model topologies can be realised. Selected topologies are presented in the following.

Memoryless, context-independent model (*MCI* model)

Whenever there are no dependencies between the edit operations z_i the model is considered to be *memoryless*. In this case, the probability of the sequence of edit operations is simply the product of the single edit operations:

$$P(Z^l = z^l, x^N, y^M) = \prod_{i=1}^{l} P(Z_i = z_i, x^N, y^M)\,. \tag{3.12}$$

Furthermore, if the single edit operations do not depend on the source and/or target symbols at position i but are the same throughout the edit sequence, the model is *context-independent*:

$$P(Z_i = z_i, x^N, y^M) = \begin{cases} c_{del}(x_{a_i}) & z_i = (x_{a_i}, \epsilon) \\ c_{ins}(y_{b_j}) & z_i = (\epsilon, y_{b_j}) \\ c_{sub}(x_{a_i}, y_{b_j}) & z_i = (x_{a_i}, y_{b_j}) \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.13}$$

a_i, b_j are the indices within the source and target sequences up to z_i. On the whole, the edit operations need to define a valid probability distribution, so $\sum_z P(z) = 1$.

The advantage of this memoryless, context-independent model as proposed in [103] is the small number of parameters θ that must be specified for decoding. Its capabilities are, however, limited for modelling effects that go beyond interactions of single symbols.

Memoryless, context-dependent model (*MCD* model)

The memoryless, context-dependent model was developed already much earlier [6]. Inspired by a problem in communications, the authors construct this probabilistic model for a noisy channel which produces output of varying length for a given input sequence. Furthermore, they propose an algorithm for decoding information transmitted through this channel. The original idea and notation is reproduced here briefly to motivate context-dependency.

The channel model is defined as follows: for each transmittable symbol $x_k \in \mathcal{X}$ a Markov chain $F(x_k)$ is constructed with states $S = \{s_1, s_2, ..., s_{I_k}\}$, where I_k is the number of states for symbol x_k. The chain has transitions producing output symbols $y_j \in \mathcal{Y}$ according to the conditional probability distribution $Q(y_j|s_{i_1} \rightarrow s_{i_2})$ where the basic state transition probability is denoted by $P(s_{i_1}|s_{i_2})$ or $P_\phi(s_{i_1}|s_{i_2})$ if no output symbol is produced (null transition). The probabilities of all outgoing transitions for a state sum up to one. By concatenating the Markov chains of the input symbols, a Probabilistic Finite State Machine (PFSM) for the whole input sequence is generated. Since the emission probabilities Q are dependent on the state transition $s_{i_1} \rightarrow s_{i_2}$ which is directly related to the input symbols in $F(x_k)$, the model is *context-dependent.*

Coming back to the previous notation this means that the edit operation probabilities are now dependent either on the source symbol x_{a_i} at edit sequence position i when conditioned on the source string, or on target symbol y_{b_j} when conditioned on the target string, respectively:

$$P(Z_i = z_i, x^N, y^M) = \begin{cases} c_{del,x_{a_i}}(x_{a_i}) & z_i = (x_{a_i}, \epsilon) \\ c_{ins,x_{a_i}}(y_{b_j}) & z_i = (\epsilon, y_{b_j}) \\ c_{sub,x_{a_i}}(x_{a_i}, y_{b_j}) & z_i = (x_{a_i}, y_{b_j}) \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.14}$$

Memory model (*MEM* model)

The memory model is an extension to the context-independent model that takes the last edit operation z_{i-1} into account when choosing the current edit operation $z_i = (x_n, y_m)$. The computation is thus extended from the simple product of the individual edit operation probabilities to

$$P(Z^l = z^l, x^N, y^M) = P(z_1) \cdot \prod_{i=2}^{l} P(z_i \mid z_{i-1}, x^N, y^M) \cdot P(\#|z_l)\,, \tag{3.15}$$

which results in a much larger number of model parameters, but allows to take care of edit operation sequences containing, e.g., consecutive deletions or operations such as transpositions. The combination with context-dependent edit operations is not recommendable due to the further increase in model parameters.

HMM-like model

The *HMM*-like model (cf. [27], [100]) is a completely different interpretation of the stochastic edit distance problem. So far, all discussed models had a hidden random variable (RV) Z representing the edit operations. The *HMM*-like model does without this implicit description of the transduction process and instead forces the consumption of one target symbol at each time instant. For this reason, insertions and deletions have to be modelled by separate hidden RVs I and D that control the propagation of the source symbol sequence. While I is of cardinality 2 representing the source index being either incremented or not, the cardinality of D is constrained to a fixed value m, allowing only m consecutive deletions in the source string.

The joint distribution therefore becomes the sum over all pairs of insertion and deletion sequences (i^M, d^M) which make up the difference between input sequence lengths N and M:

$$P(x^N, y^M) = \sum_{\{(i^M, d^M): \sum_{j=1}^{M} i_j - d_j = N-M\}} P(I^M = i^M, D^M = d^M, x^N, y^M)\,. \tag{3.16}$$

Note the fixed length M of the sequence of insertions and deletions. Apart from this restriction, this HMM-like model is similar to the MCD model as the conditional probabilities $P(y_j|x_i)$ are modelled – in this case explicitly:

$$P(I^M = i^M, D^M = d^M, x^N, y^M) = \prod_{j=1}^{M} P(i_j)P(d_j|i_j)P(x_j|d_j)P(x_j|i_j)P(y_j|x_j)\,. \tag{3.17}$$

3.4 Evaluation: Pronunciation classification

In a pronunciation classification task, surface pronunciations obtained from manual phonetic transcriptions are compared to canonical pronunciations taken from a phonetic lexicon. For each surface form, a minimum distance classifier then selects the word with the most similar canonical form. The application of phonetic similarity is clearly defined in this task and phonetic distance measures can meaningfully be compared with each other. This evaluation method was already chosen in previous studies on this topic ([103], [27]). It should be noted, however, that the results strongly depend on the test data sets and the provided lexicons which makes comparisons between studies difficult. For this reason, the experimental evaluation comprises a similar setup to the studies in [103], [27] to verify the reference results on conversational speech, and an application-specific setup based on the MEDTRANS corpus (cf. chapter 2, section 2.5.1) to evaluate the phonetic distance measures on dictation speech.

3.4.1 Experimental setup

The classification problem is defined as follows: For an observed target string t (surface pronunciation) the corresponding source string s (canonical pronunciation) shall be found. Let $C(w) = s$ denote the mapping from an orthographic word w to a pronunciation s and $C^{-1}(s) = w$ the inverse mapping. Then $W_t = \{C^{-1}(t)\}$ is the set of all words corresponding to the observed surface pronunciation t. Furthermore, let W be the orthographic reference vocabulary of all words w and $S = \bigcup C(w \in W)$ be the set of all canonical pronunciations from W. Then, a minimum distance classifier is defined as

$$\hat{W}_t = C^{-1}(\operatorname*{argmin}_{s\in S} d(t,s))\,. \tag{3.18}$$

$\hat{W}_t$ contains all those words of vocabulary W whose canonical pronunciations have the minimum distance to the target pronunciation t. Since $C(w)$ is not unique due to homophony, the error rate has to be expressed in terms of the information retrieval measures *Recall* and *Precision.* Recall describes how many of the relevant words were found, while Precision expresses how many of the found words were actually relevant. The *F1-measure* is the harmonic mean of Recall and Precision [120]. In the previous notation, these measures are defined as

$$\text{Recall} = \frac{|\hat{W}_t \cap W_t|}{|W_t|}\,, \quad \text{Precision} = \frac{|\hat{W}_t \cap W_t|}{|\hat{W}_t|}\,, \text{and} \quad \text{F1} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{3.19}$$

To give an example, for a given pronunciation $t =$ `/W n d/`, $\hat{W}_t = \{$`pound`, `found`, `sound`, `round`, `wound`$\}$ was hypothesized. The actual $W_t = \{$`pound`, `pounds`$\}$, which means that

Recall = 1 / 2 * 100% = 50%, and Precision = 1 / 5 * 100% = 20%. The results presented in [103] and [27] are equivalent to Precision values.

It must be noted that these measures favour the deterministic distance measures, as words with equal distance are accumulated in the set of hypotheses. Therefore, Recall values tend to be high, while Precision values are very low. To provide a fair comparison, the k-best word hypotheses $\hat{W}_{t,k}$, $k = 1..5$ were calculated for each target string, and the rank of the correct word within this sorted list was determined. This means that the best matching canonical pronunciations $s_{j_1}...s_{j_k} \in S$ are sorted such that $d(t, s_{j_1}) < d(t, s_{j_2}) < ... < d(t, s_{j_k})$. The resulting rank is the smallest k for which $\hat{W}_{t,k} \cap W_t \neq \{\}$.

3.4.2 Data sets

The phonetic similarity measures were tested on two different data sets. The first data set was compiled from manual phonetic transcriptions of the SWITCHBOARD corpus ([34], [41]) performed at the International Computer Science Institute (ICSI), University of California, Berkeley. These transcripts of conversational speech exhibit a substantial amount of variation in pronunciation and are, therefore, well-suited for the pronunciation classification task. This data has also been used in previous studies on pronunciation modelling ([103], [27]), but it was not possible to reproduce the exact division into training and test data sets for these experiments. The canonical reference pronunciations were taken from the pronunciation dictionary created in terms of the SWITCHBOARD resegmentation project [23] at the Institute for Signal and Information Processing (ISIP), Missisippi State University. Both, transcriptions and dictionary are in ARPABET phonetic notation. In the following, this data set will be called SWBTRANS.

The second data set was compiled from the MEDTRANS corpus of narrow phonetic transcriptions of medical dictations (cf. chapter 2, section 2.5.1). In contrast to the SWITCHBOARD data, the domain fits the target application of medical dictation, as the transcriptions were done from actual real world recordings. The pronunciation dictionary was provided by Philips Speech Recognition Systems.

Both data sets were post-processed in a similar fashion before the experimental evaluation was done. The following steps were performed on the raw data to obtain the final data sets:

- removal of non-speech, unintelligible segments, incompletely uttered words (e.g., false starts), and hesitations,
- correction of misspelt orthographic words,
- validation of the source and target phoneme inventories,
- removal of utterances with phone sequence length < 3.

In the SWBTRANS set, the target phone inventory has been reduced to those 51 symbols which occurred at least 10 times in the corpus to avoid conditioning problems during training of the stochastic distance measures. A few key figures for both sets are listed in table 3.1.

3.4.3 Training and decoding

So far, the model parameters for the stochastic measures have been described but not set to specific values. It is one of the major advantages of these measures that the model parameters

Table 3.1. Key figures for pronunciation classification evaluation data sets SWBTRANS and MEDTRANS: number of training and test word pairs, number of unique orthographic words, number of unique phonetic forms (pronunciations), and number of forms per word for source (canonical reference transcription) and target string (observed surface pronunciation). The number of forms per word may be less than one, as one pronunciation can be represented by many orthographic words.

	SWBTRANS		**MEDTRANS**	
	Count	[%]	Count	[%]
sample pairs (x^N, y^M)	19,607	100.0	74,044	100.0
training pairs	18,607	94.9	73,044	98.6
... identical pairs $(x^N = y^M)$	6,982	35.6	58,327	78.8
... differing pairs $(x^N \neq y^M)$	11,625	59.3	14,717	19.8
test pairs	1,000	5.1	1,000	1.4
... identical pairs $(x^N = y^M)$	405	2.1	831	1.1
... differing pairs $(x^N \neq y^M)$	595	3.0	169	0.3

	SWBTRANS		**MEDTRANS**	
	Source	Target	Source	Target
words	3,622	3,622	6,346	6,346
forms	3,575	8,744	6,568	11,461
forms / word	0.987	2.414	1.035	1.806
alphabet size	42	51	46	46

can directly be estimated from data. This way, the similarity measures become adaptable to specific tasks or domains.

The stochastic measures were implemented as Dynamic Bayesian Networks (DBN) with the Graphical Models Toolkit GMTK [8] following the approach in [27]. The DBN representation allows for easy modification of model topologies and provides ready-made solutions for parameter learning via probabilistic inference. In this implementation the parameters were estimated with the Expectation-Maximisation (EM) algorithm optimising the Maximum Likelihood criterion. For all models the algorithm converged after three EM iterations. Decoding was done with the Viterbi algorithm for Viterbi edit distance (cf. eqn. 3.10) and the Junction Tree algorithm for stochastic edit distance (cf. eqn. 3.11). As the differences were marginal, results are reported on stochastic edit distance only.

The memoryless, context-independent model was re-implemented in C++ following the reference implementation in [103]. This implementation was much faster than the GMTK model: Training time on the SWBTRANS corpus could be reduced from 10 hours with the GMTK model to 15 seconds in the C++ model, giving exactly the same results. Similar speed-ups were observed for decoding.

3.4.4 Quantitative results

Classification

The Recall/Precision/F1 measures determined in the pronunciation classification experiments are collected in table 3.2 and the statistics for the 5-best ranking of word hypotheses in 3.3.

The stochastic measures perform superior to the deterministic ones with both data sets as indicated by the F1 measure and the ranking results. For the SWBTRANS data set, the best results were achieved by the memory model (*MEM*) with 82.69%, followed by the

Table 3.2. Recall, Precision, and F1 measures determined for minimum distance classification with data sets SWBTRANS and MEDTRANS: standard symbol alphabets (std), mapped symbol alphabets (map), and extensions with phonetic class information (cla). The models are: Levenshtein distance (LEV), Normalised Levenshtein Distance (NLEV) with path normalisation ($NLEV_{path}$) and metric-compliant normalisation ($NLEV_{metric}$), Memoryless context-independent model (MCI), Memoryless context-dependent model (MCD), Memory model (MEM), and HMM-like model (HMM).

	Model	**Eqn.**		**SWBTRANS**			**MEDTRANS**		
				Rec.	Prec.	F1	Rec.	Prec.	F1
deterministic measures	*LEV*	(3.3)	std	78.33	67.65	72.60	90.08	94.38	92.18
			map	77.80	68.41	72.80	90.08	94.38	92.18
		(3.8)	cla	67.53	74.47	70.83	87.78	95.22	91.35
	NLEV	(3.4)	std	74.33	71.60	72.94	88.03	94.57	91.19
			map	77.80	68.41	72,80	88.03	94.57	91.19
		(3.8)	cla	69.87	76.64	73.10	87.58	95.29	91.28
	$NLEV_{path}$	(3.6)	std	72.58	69.83	71.18	87.03	93.97	90.37
			map	70.73	70.15	70.26	87.03	93.97	90.37
	$NLEV_{metric}$	(3.7)	std	74.33	71.60	72.94	88.03	94.57	91.19
			map	73.07	71.99	72.52	88.03	94.57	91.19
stochastic measures	MCI_{sto}	(3.13)		77.92	85.60	81.58	**90.13**	**98.40**	**94.09**
	MCD_{s_i}	(3.14)		70.60	77.62	73.94	89.53	97.80	93.48
	MCD_{t_i}			75.19	82.60	78.72	89.93	98.15	93.86
	$MCD_{s_i,s_{i-1}}$			74.47	82.28	78.18	89.63	97.85	93.56
	$MCD_{t_i,t_{i-1}}$			76.54	84.72	80.42	**90.13**	98.35	94.06
	MEM	(3.15)		**78.72**	**87.07**	**82.69**	89.88	97.85	93.70
	HMM_{s_i}	(3.17)		71.49	79.25	75.17	89.43	97.75	93.41
	HMM_{t_i}			71.81	79.55	75.48	88.63	96.95	92.61

memoryless, context-independent model (*MCI*) with 81.58%. For the MEDTRANS data set, the situation is reversed as the *MCI* model performed slightly better than the memory model (94.09% compared to 93.70%). The result rankings in table 3.3 confirm these findings. Comparing the stochastic measures, the context-independent measures (*MCI* and *MEM*) returned the best results. Context-dependency works better when applied to the target string than to the source string. This finding is intuitive as there is much more variation in the target strings than in the source strings. Extending the context-range to the previous symbols further pushes the performance to a significant degree. The *HMM* model cannot keep up with the rest and only performs marginally better than the deterministic measures. Interestingly, there is not much difference between the source-sequence and target-sequence oriented models.

Compared to the stochastic measures the basic deterministic measures achieve higher Recall, but lower Precision scores with both data sets. This tendency is easily explained as the distance scores are independent of the actual symbols involved in the edit operation. Therefore, many word pairs can be found that have the same score, meaning that the set $\hat{W}_t$ of hypothesised words is large and the correct word thus likely to be found. The stochastic measures on the other hand either hit or miss, and there are only very few cases where there is more than one best match. Normalisation of the Levenshtein distance (*NLEV*) mainly improves the Precision as it reduces the number of best matching candidates, sometimes, unfortunately, by removing the correct word as well. The alternative normalisation methods based on the editing path length ($NLEV_{path}$) and compliant with the metric axioms

Table 3.3. Ranking results: percentage of data samples where the correct word appeared at rank k in the pronunciation classification results list.

SWBTRANS

Rank	*LEV*		*NLEV*				*MCI*	*MCD*				*MEM*	*HMM*	
k	*std*	*cla*	*std*	*cla*	*path*	*metric*		s_i	t_i	s_i, s_{i-1}	t_i, t_{i-1}		s_i	t_i
1	67.7	75.4	71.4	77.2	69.6	71.4	86.7	78.5	83.5	83.2	82.9	**87.9**	80.1	80.6
2	19.8	5.7	14.2	6.5	14.3	14.2	5.4	5.6	4.9	5.6	5.6	4.4	5.4	4.8
3	10.0	3.2	5.6	3.1	4.6	5.6	1.5	3.2	2.7	2.5	2.3	2.2	2.3	2.7
4	2.1	2.1	4.2	1.8	3.1	4.2	1.1	1.4	1.1	1.1	1.3	1.3	1.5	2.3
5	0.4	0.9	1.1	1.0	1.8	1.1	0.5	1.3	1.0	0.7	0.6	0.5	0.8	0.6
>5	0	12.7	3.5	10.4	6.6	3.5	4.8	10.0	6.8	6.9	7.3	3.7	9.9	9.0

MEDTRANS

Rank	*LEV*		*NLEV*				*MCI*	*MCD*				*MEM*	*HMM*	
k	*std*	*cla*	*std*	*cla*	*path*	*metric*		s_i	t_i	s_i, s_{i-1}	t_i, t_{i-1}		s_i	t_i
1	94.2	95.7	94.7	95.7	94.2	94.7	**98.8**	98.2	98.6	98.3	**98.8**	98.4	98.1	97.3
2	4.6	1.9	3.0	1.3	2.7	3.0	0.6	1.1	1.1	0.8	0.4	1.0	0.9	1.2
3	0.9	1.1	1.0	1.4	1.7	1.0	0.3	0.2	0.1	0.3	0.3	0.1	0.3	0.6
4	0.2	0	0.8	0.3	0.7	0.8	0.1	0	0.1	0	0	0	0	0.2
5	0.1	0.3	0.2	0.3	0.2	0.2	0	0.2	0	0.2	0	0	0.1	0
>5	0	1.0	0.3	1.0	0.5	0.3	0.2	0.3	0.1	0.4	0.5	0.5	0.6	0.7

($NLEV_{metric}$) did not bring better results for any of the two data sets and are, therefore, not suited for this type of input data.

Inclusion of phonetic feature weights (cla rows in table 3.2) improved the Precision scores at the expense of the Recall for both data sets. In contrast to the standard implementations, the Recall/Precision ratio is almost inverted as the distance scores are highly diversified due to their symbol-dependency. The gains are, however, small and could only be measured when the weighting factor α was low (SWBTRANS: $\alpha = 0.2$, MEDTRANS: $\alpha = 0.3$) and the dependency therefore low as well. This is an interesting result, because it suggests that the discriminating effect is more beneficial than the simple accumulation of an absolute score. For the latter, it would be necessary to re-define the costs for insertions and deletions as well, which would require further phonetic knowledge that goes beyond the phonetic feature labels such as articulatory constraints or phonologic rules.

There are several reasons for the significant differences between the results for the two data sets. First, the number of samples with identical source and target string in the test set of MEDTRANS is about twice as high as in SWBTRANS. Furthermore, SWBTRANS has different phoneme inventories for source and target pronunciation which makes comparisons with the deterministic measures difficult. Second, SWBTRANS is a corpus of conversational speaking style in contrast to the dictation speaking style in the MEDTRANS corpus. As a consequence, there are fewer and less prominent deviations in pronunciation in the MEDTRANS data set. For these reasons, the SWBTRANS set is more challenging and hence more interesting for evaluation in contrast to the less variable MEDTRANS data set. MEDTRANS, on the other hand, comprises a larger vocabulary and thus potentially higher confusability among the words.

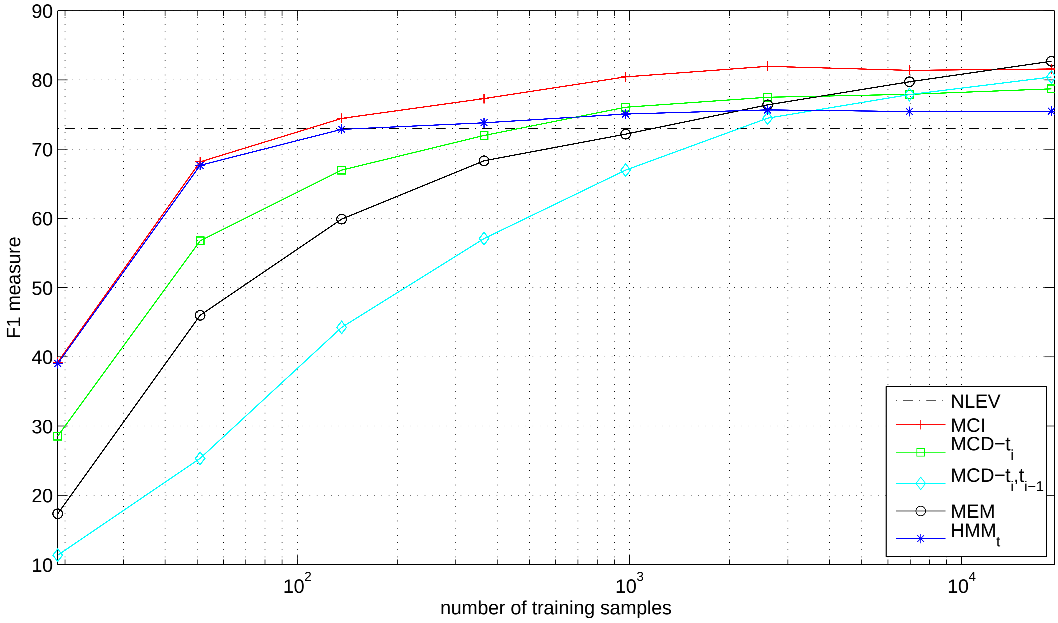


FIGURE 3.3. F1 performance versus training set size for SWBTRANS data set. Note: For the *MCD* and *HMM* models only the best performing configurations are plotted.

Dependency on training data for the stochastic measures

For the stochastic measures, the number of training samples needed to achieve a reasonable performance is essential. Figure 3.3 shows the relation between the F1 measure and the amount of training data for the SWBTRANS data set. With increasing number of model parameters, the need for training samples increases as well as shown in table 3.4. To achieve the same performance as the best deterministic measure in terms of F1 – the *NLEV* measure – the *MCI* model requires only around 100 training sample pairs. The same holds for the *HMM* model, which has almost reached its maximum there. The *MCD* model with single symbol context already needs 400 samples to measure up with *NLEV*. The *MEM* model requires one order of a magnitude more data ($\sim$ 1,000 samples). And finally, the *MCD* model with two symbols of context requires even around 4,000 data samples.

TABLE 3.4. Number of model parameters for the stochastic measures on the test data.

Model	Eqn.	SWBTRANS	MEDTRANS
MCI_{sto}	(3.13)	2,236	2,209
MCD_{s_i}	(3.14)	4,940	4,418
MCD_{t_i}		4,940	4,418
$MCD_{s_i,s_{i-1}}$		140,608	103,823
$MCD_{t_i,t_{i-1}}$		96,148	103,823
MEM	(3.15)	4,999,696	4,879,681
HMM_{s_i}	(3.17)	2,236	2,209
HMM_{t_i}		2,236	2,209

3.4.5 Qualitative results

Apart from the quantitative analysis, also a manual inspection of the experimental results was made to get a better insight on how the different methods work. For this inspection, a subset of samples was selected from both test data sets that showed a Levenshtein distance $d_{lev} > 1$. The subset was then visually analysed to find categories that describe the observed deviations. After the sample pairs had been assigned to the categories the returned candidate word lists of each tested measure were compared with each other.

The following main categories of deviations could be determined, in order of frequency:

- Moderate reductions: `/T AX M EY DX AX Z/` → `/T AX M AA T OW Z/`: `tomatoes`

  Moderate reductions are minor deviations between the strings that do not affect the number of syllables per word. They can occur due to speaking style variation or small disfluencies in fluent speech.

  The deterministic measures sensitively react on moderate reductions, as they fail in about one third of all cases. The stochastic measures get on better with this type of deviation. The *MCI* model excels as the only method which was able to correctly recognise all moderate reductions.

- Massive reductions: `/EY M OW/` → `/EH N IY M OW R/`: `anymore`

  Massive reductions are caused by fast speech or sloppy pronunciation of the speaker. The canonical phone sequence is strongly distorted, by phone substitutions and particularly phone deletions, usually at the word end. For these experiments, an utterance was labelled as massively reduced if at least one of the syllables was missing. In contrast to word artifacts, however, the uttered phone sequence can still be correctly understood by an attentive and experienced human listener.

  All measures were substantially affected by these deviations. The Levenshtein and *HMM*-like models suffer from almost complete breakdown. The best results were achieved by the memory model which was able to correctly recognise more than half of the massively reduced pronunciations. In case of utterances such as `/Z P R AA LG IH/` → `/P R AA B AX B L IY/` for the word `probably`, this is remarkable.

- Segmentation errors: `/L AE S T SIL N/` → `/L AE S T/`: `last`

  Segmentation errors were introduced already at the production of the phonetic corpora used in the experiments. In case of the SWBTRANS data set, the word level segmentation of the phonetic transcription was done automatically based on time-stamp information, and in case of the MEDTRANS data set, the automatic orthographic segmentation was not changed during the phonetic transcription process. Therefore, segmentation errors are unfortunately inevitable for some samples.

  The effect on the experimental results, however, is low. The *MCI* model again is most robust against segmentation errors, as well as the deterministic measures. The worst effect can be observed with the *MCD* and *HMM* models, most probably because the contexts are broken for this type of deviation.

- Short, functional word artifacts: `/V AX JH/` → `/B AH T/`: `but`

  The classification task becomes harder the shorter the compared pronunciations are. Mostly, this happens for functional words which convey little meaning and usually only serve syntactic purposes within a sentence. Therefore, they also tend to be heavily reduced and appear as almost random. A sample was labelled as short artifact if it was not possible to infer the orthographic word from the observed phone sequence manually.

None of the methods could appropriately deal with short functional word artifacts. Even the *MEM* model as the best method in this respect was not able to recover half of the words. The *HMM* models failed completely, as did the *MCI* model. The Levenshtein model performed not much worse than the *MEM* model in terms of recall, as there are many candidates with short pronunciations in the lexicon.

- Transcription errors: `/IH D V AE N Z/` → `/AE D V AE N T IH JH/`: `advantage`

  A data sample was labelled as transcription error if the phonetic transcription suggested a different word from the lexicon than what had actually been transcribed. Such mistakes often happen with inflected words where the baseform was transcribed instead of the actual inflected form or vice versa. Another type of transcription errors were phone repetitions.

  Phone repetitions were easily handled by all measures while the rest of the transcription errors were not resolvable for any measure. Therefore, these samples are best ignored in the analysis.

- Heterophones: `/L IH V/` → `/L AY V/`: `live`

  Heterophones are words that have the same orthographic representation, but different phonetic representations. By definition, it is impossible to disambiguate such words without additional semantic cues. A few words in the test corpora were heterophones where only one of the two possible phonetic realisations was present in the lexicon, but the other one was uttered.

  Almost all methods fail for heterophones, except for the memory model which was able to find the correct word for the above example. This may, however, have happened incidentally.

3.5 Suggestions for extensions

The analysis of the pronunciation classification results showed that the remaining errors are mostly artifacts that cannot be classified meaningfully. Therefore, there is hardly any room for improvement on this data. Pronunciation classification, however, is only one of many possible applications in text matching, and as such it does not directly address those observations from chapter 2 that require a context wider than just a single word within a text alignment. At this point, a few promising extensions to the phonetic similarity measures shall be discussed briefly.

3.5.1 The syllable as structuring unit

In section 2.2.2 the syllable structure was identified as one of the major factors for describing phonologic variation in spoken language. Assuming that the syllable structure of the input phone sequences can be determined automatically, this knowledge can be integrated into phonetic similarity measurement.

One way of exploiting this knowledge would be to split the input phone sequences into individual syllables and perform syllable matching instead of whole-word matching. Since the variation in syllable length is much smaller than the variation in word length (cf. figure 3.4),

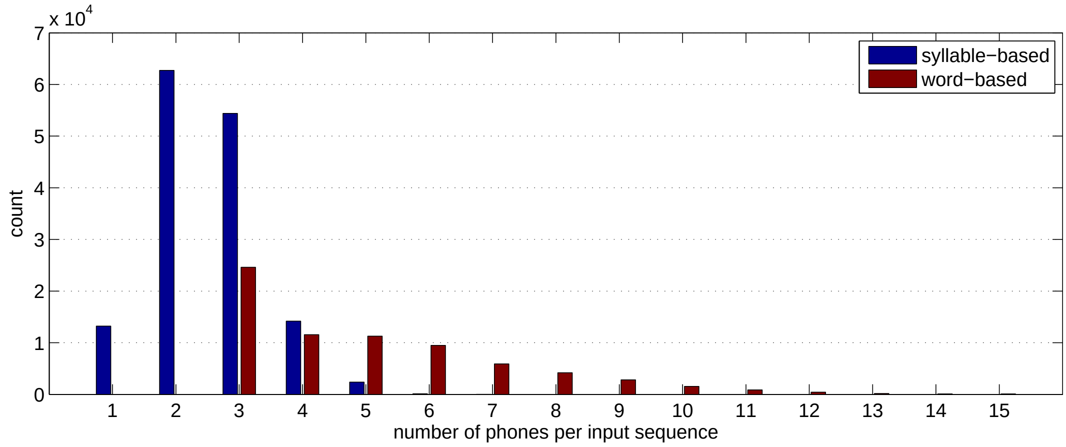


FIGURE 3.4. Number of phones per phone input sequence for syllable-based and word-based input sequences in the MEDTRANS training data set.

this partitioning step has a normalisation effect that would particularly be beneficial for the stochastic measures whose distance scores increase exponentially with edit sequence length. Furthermore, the syllable boundaries may give good guesses for potential segmentation errors in the alignment as discussed in section 2.5.2.

The syllable structure may also be integrated by means of position-dependent weighting factors. This way, edit operations are weighted with respect to their function in a syllable. For instance, the deletion of a consonant like `/d/` could be assigned a lower weight in coda position than in onset position. At the same time, such weighting factors would enforce the correct alignment of the phone strings with respect to the syllable structure.

3.5.2 Parameter-tying schemes

The application of stochastic edit distance measures is limited by the relatively high number of model parameters that must be specified or trained from data. Particularly for complex models such as the context-dependent model or the memory model, a reduction of the number of model parameters could reduce the amount of required training data. Ristad and Yianilos already gave this indication in their work [103]. The dependence on the previous phone x_{i-1} or the previous edit operation z_{i-1} could be weakened to the previous phone class (vowel, fricative, plosive, nasal, stop, ...) or the previous type of edit operation (substitution, deletion, or insertion). Similarly, parameter-tying could also be conditioned on the syllable structure.

3.6 Conclusion

Phonetic similarity matching is a process of joint alignment and local scoring. While the alignment procedure usually follows a dynamic programming scheme, the various algorithms mainly differ in their scoring models. The main contribution of this chapter is an experimental evaluation of both, deterministic and stochastic edit distance measures, as in the literature, so far only either stochastic measures or normalisation methods of the Levenshtein distance have been benchmarked against the Levenshtein distance. This comparative evaluation was set up as a pronunciation classification task based on phonetic transcriptions of conversational speech recordings and – as a further contribution – of dictation speech. Furthermore, this evaluation is the first one that relates the pronunciation classification performance to the amount of data used for training the stochastic models.

In direct comparison the stochastic edit distance models outperform the deterministic distance models in the pronunciation classification task. The performance gain comes from the prior knowledge obtained from data during the training phase which allows for better differentiation between similar sequences. Their abilities go even as far as correctly assigning pronunciations suffering from massive reductions, where the deterministic models fail completely. For applications this means that as soon as domain and task-specific real world training data is available, a stochastic edit distance model should be the primary choice for similarity measurement.

The choice for a specific stochastic edit distance model basically depends on the amount of available training data. The experiments show that for the memoryless, context-independent model already around 100 training pairs are enough to achieve higher F1 scores than the best performing deterministic measure. With such little training data, it could be possible to develop highly specific models, e.g., speaker-, domain-, or maybe even section-specific models. The memory model shows the highest potential, but requires about an order of magnitude more data to draw levels with the deterministic measure and another order of magnitude to reach the performance of the memoryless model. The learning curve, however, is then still not bounded, which makes this model interesting for long-term adaptation scenarios where large amounts of data are available.

Part II

Application-Driven Solutions

Chapter 4

Automatic Reconstruction of Medical Dictations*

After decades of research, speech recognition technology has reached a level where it can be successfully integrated into products for everyday use. In particular, this applies to dictation systems with integrated speech recognition which help reduce the amount of manual transcriptions. In the medical domain, where dictation traditionally plays an important role, speech recognition systems have contributed to a more efficient report creation process since medical transcriptionists no longer have to type whole documents, instead they only do the post-processing to create the final reports. This way, highly skilled medical transcriptionists make better use of their expertise.

In many cases, this post-processing step unfortunately still involves a lot of tedious editing: recognition errors have to be corrected, and the style and formatting of the document have to be adapted to the standards applied to written reports. Particularly for dictations by unexperienced users, post-processing can become time consuming, and thus may lead to many and various deviations between the recognition results and the final reports.

While recognition results and final reports are usually available in abundance, manual transcriptions of the actual spoken words without recognition errors (i.e., assuming *perfect recognition*) are costly and scarce. For training automatic speech recognition systems, however, literal transcriptions of the actual words are needed.

A standard methodology to overcome the problem of non-literal transcriptions in ASR training is unsupervised or lightly supervised training [64], [56]. These approaches allow the generation of statistical models from only small amounts of literal transcriptions together with large amounts of non-literal transcriptions in an iterative fashion. For language model training, methods like linear model interpolation [17] or transformation-based learning [89] are used to cope with non-literal transcriptions. Although these methodologies lead to reductions in word error rate for as various domains as news broadcasts or transcriptions of class lectures [44], they do not give explanations for the mismatches between the non-literal data and the actual wording in the training utterances. Furthermore, problematic segments like disfluencies, hesitations, or speaker corrections cannot be modelled without proper annotation. For these reasons, literal transcriptions are still valuable.

[*] This chapter is a verbatim reprint of the article *Semantic and Phonetic Automatic Reconstruction of Medical Dictations*, submitted to the journal *Computer Speech and Language* by Elsevier. It is co-authored by Christina Drexel, Leo Fessler, Jeremy Jancsary, Alexandra Klein, Gernot Kubin, Johannes Matiasek, Franz Pernkopf, and Harald Trost. The sole contributions of the author are the phonetic similarity matching function, the phonetic reconstruction rules, and the experimental evaluation of the reconstruction quality at the end of this chapter.

These motivations lead us to the definition of the problem of how a literal transcription can be automatically reconstructed from non-literal transcripts of different information sources. This problem has already been addressed by [87] for modelling disfluencies and hesitations in medical dictations. However, a comprehensive model for automatic reconstruction needs to go beyond the scope of specific phenomena and provide a generic framework for exploiting the full potential of the analysed documents.

In this article, we propose such a reconstruction framework and describe a system which has been developed for automatically reconstructing the actual spoken words from the recognition result and the final medical reports. These two different input sources are complementary for the task of reconstructing literal transcripts. The resulting reconstructions can be used the same way as manual transcriptions for training speech recognition systems.

The base for reconstruction is an alignment between the written report and the recognition result. The alignment takes into account semantic information (for explaining reformulations) and phonetic information (for explaining recognition errors) as well as syntactic information in terms of document formatting. From the interpretation of the deviations between the written report and the recognition result, the words which are considered to have actually been spoken are reconstructed. According to the proposed methods, we name our approach Semantic and Phonetic Automatic ReConstruction (SPARC). The main innovative aspect of our method is the optimal interplay between two independent knowledge sources, namely semantics and acoustics/phonetics in the categorisation of differences between automatic transcript and final document, as well as in the reconstruction of the original utterance from these two data sources.

Qualitative and quantitative evaluations based on manual transcriptions have shown that, in many cases, the alignment leads to a correct reconstruction. The resulting reconstructed text can serve not only as a base for training and improving the speech recognition system; a deeper understanding of the typical reformulations and reformatting may eventually also support a shift from mere speech recognition to document production in dictation applications.

In the following sections, we will continue with a more detailed account of the SPARC approach in section 4.1 and a description of the available text corpora in section 4.2. Following this introductory part, in sections 4.3, 4.4, and 4.5 the three main units of the approach – text alignment, similarity measurement, and text reconstruction – are then elucidated. In section 4.6, we report experimental results in terms of the quality of the reconstructed text and an automatic speech recognition experiment with retrained language model. We conclude the paper with a discussion of the results and an outlook for further applications.

4.1 SPARC approach

The SPARC approach is a method for the automatic production of literal transcriptions from available data sources in large document production environments using speech recognition. Three types of data are currently available in such systems:

- *Audio files (AF)*, comprising the original utterances;
- *Draft transcriptions (DT)* – or more simply: *recognised texts* -, produced by the dictation system (containing the recognition errors);
- *Final documents (FD)* – or more simply: *written texts* -, produced by the typist (where recognition errors are corrected but where also some parts are re-formulated in a way different from the original utterances).

Error-free *literal transcriptions (LT)* – or more simply: *reference texts* – of the audio files, however, are usually not available, or only to a certain degree if some manual transcriptions have been made. Yet, literal transcriptions of the original spoken utterances are needed for advancing the accuracy and efficiency of automated dictation:

- Aligned corpora of LT and FD can be used to automatically learn recurrent reformulations, thus allowing automated dictation to be augmented by an automatic text reformulation module which provides a draft that is closer to the intended final document.
- Large quantities of literal transcriptions and audio files can serve as data for training of the acoustic and language models to decrease the word error rate of speech recognition.

For medical dictations, the reconstruction task was already described by [87]. There, the authors propose an augmented probabilistic finite-state model for generating semi-literal transcriptions. This probabilistic model handles so-called 'out-of-transcription expressions' like greetings, false starts and repairs, and filled pauses as the only sources of mismatches between recognised and written texts. For the same task, SPARC provides added value by also explaining and categorising such mismatches. For hypothesising the reconstructed text mismatches are not only detected, but also interpreted. The interpretation of a mismatching token pair as e.g., a recognition error, or a reformulation of the typist helps in designing more accurate models for the differences between spoken and written form of medical dictations. [123] describes a hybrid method for detecting speech recognition errors in radiology reports based on semantic knowledge, constraint rules and statistical modelling (i.e., pointwise mutual information and co-occurrence analysis). In [44], transcription generation was presented for recorded academic lectures with a finite-state transducer approach.

Semantic relatedness and similarity measures have mostly been developed to improve the recall of Information Retrieval (IR) systems. There are two main established ways of measuring the semantic similarity between two terms: on the one hand, relatedness can be measured in terms of the distance between two words or multiword expressions in a knowledge base, e.g., WordNet (see [26]). On the other hand, relatedness can be derived from a corpus by determining co-occurrence and context features with IR methods. Often, corpus- and knowledge-based measures are combined. Due to the many available knowledge sources, the medical domain lends itself well to knowledge-based measures for semantic relatedness and similarity (for an overview, cf. [88]).

Similarly, phonetic similarity measurement has been used for addressing many topics in ASR: modelling pronunciation variation (e.g., [103], [27]), predicting ASR errors [31], measuring acoustic confusability [96], discriminative language model training and OOV detection [99], or IR [133]. In many of these applications, confusion matrices are used to measure the phonetic similarity of phone sequences or phone confusion networks. These matrices are either handcrafted, e.g., from phonetic class information, or estimated from data.

The technological goal is to automatically construct an error-free literal written transcription of the user's original utterances. Methodologically, the basis for this reconstruction is formed by an analysis of the semantic and acoustic differences between DT and FD. Scien-

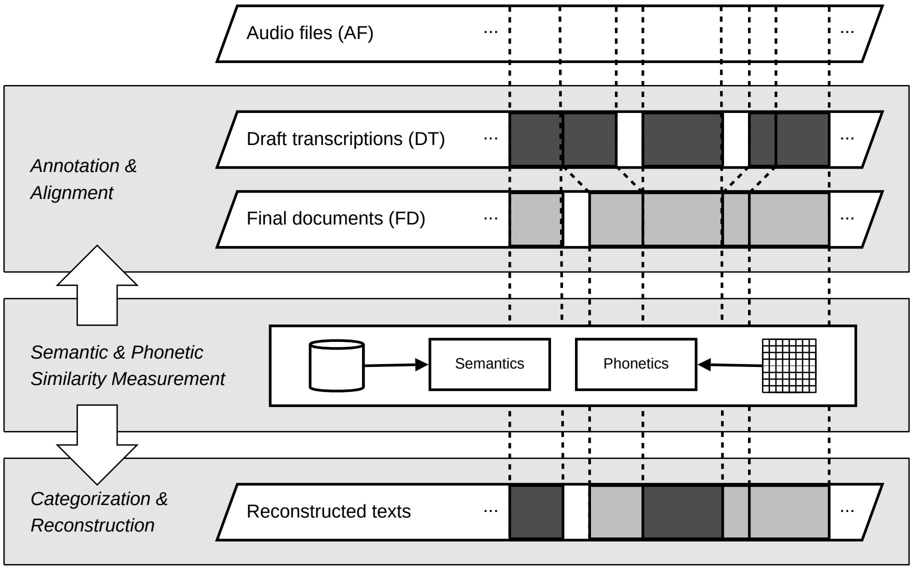

Audio files (AF)
Annotation & Alignment
Draft transcriptions (DT)
Final documents (FD)
Semantic & Phonetic Similarity Measurement
Semantics
Phonetics
Categorization & Reconstruction
Reconstructed texts

Figure 4.1. SPARC architecture: draft transcriptions and final documents are first annotated, then properly matched based on semantic and phonetic similarities, and finally categorised and selectively combined into a reconstruction hypothesis.

tifically, SPARC requires solutions for the following problems:

- Automatic semantic annotation of text corpora with the help of a domain-specific ontology.
- Accurate text alignment and chunking for the available draft transcriptions and final documents.
- Methods for comparing aligned text chunks for semantic and phonetic similarity.
- Classification of text chunks based on the similarity measures (text reconstruction).

Figure 4.1 illustrates the architecture of the SPARC approach. Our method starts with the automatic semantic annotation of both DTs and FDs. Pairs of documents are then aligned to identify chunks where texts display differences. Semantic similarity is measured based on the semantic annotation, while phonetic similarity is determined online with a parameterised stochastic similarity measure. This way, the difference between a specific chunk in DT and FD can be categorised as correction of a speech recognition error or a reformulation by the human typist – or a combination of both (cf. table 4.1). Reconstruction of the originally dictated words is based on this analysis. Note that semantic and phonetic similarity measurement are used for both alignment and reconstruction.

SPARC can be adapted to any domain and to any language as long as the basis for training/learning – namely adequately sized parallel corpora of DT and FD – as well as the necessary linguistic resources – lexical, morphology, thesauri, etc. – are available. We implemented SPARC for English medical reporting, due to the fact that very large collections of medical corpora in English can be obtained, and medical reporting is at the moment by far the most important application of speech recognition in professional dictation.

Table 4.1. The SPARC approach to text reconstruction. Based on semantic and phonetic similarity measurements, chunks of written and recognised text can be classified as either matches, reformulations, corrections or a combination of both.

		PHONETICS	
		similar	*dissimilar*
SEMANTICS	*similar*	MATCH	REFORMULATION
	dissimilar	CORRECTION	REFORMULATION & CORRECTION

4.2 Data description

For reconstruction, we distinguish between matching (i.e., identical) and mismatching parts of the aligned texts. As this task is trivial for matching parts, only the mismatching parts will be of interest. Generally, we describe mismatches between texts on word level in terms of the mismatch edit operations insertion, deletion, and substitution. This way, a word error rate can be determined easily, but mismatch interpretation is difficult since actual mismatches can be composed of several adjacent mismatch edit operations. For this reason, we define a *mismatch region* as a contiguous sequence of mismatch edit operations in order to establish correspondences between matched words.

A statistical study of a corpus of 80,000 medical reports with 38 million words revealed an average length of 2.3 words for a mismatch region and an average occurrence of 3.6 times for this region within the corpus. Regions occurring only once account already for 60% of all mismatches while frequent regions occurring $\geq$ 10,000 times only account for about 11% of all mismatches. Such highly frequent mismatches are, e.g., insertions or deletions of punctuations and short words. On the other hand, regions of length 1 cover around 20% of all mismatches, and 75% of all mismatches occur in regions of length ≤ 5. For the reconstruction task, this means that only relatively short symbol sequences have to be processed.

Mismatches can be traced back to the human dictation process, the automatic recognition process, and the human transcription process. In general, the dictating person speaks freely, thus hesitations, self-corrections, and repetitions can be observed often in the recordings, but of course not in the final documents. ASR is error-prone, resulting in the confusion of words which are phonetically similar. The transcription process completes the range of mismatch sources by adding formatting to the text according to certain well-defined standards. Formatting affects the text in two ways: first, by additional structure like inserted punctuations, paragraph breaks, or capitalisation of words, and second, by formatting of particular document entities like headings, grammatical units (dates, quantities, etc.), or enumerations out of continuous text. The latter formatting step makes reconstruction difficult, as different speaking variants are mapped onto a standardised written form. Furthermore, the structure and style of the text can be altered by reformulations of the typist as well. These alterations include expansion of abbreviations, acronyms, and short forms, or grammatical corrections like changes in genus, tempus, or numerus so as to put the final written text into a proper stylistic and grammatical form.

4.3 Text alignment

Establishing proper alignment of the final report (FD) and the recognised text (DT) is an important prerequisite for all further steps [49].

During alignment, both input documents are viewed as sequences of tokens. A generalised Levenshtein alignment algorithm is then applied to these sequences (cf. [69]). The Levenshtein algorithm views alignment as a minimisation problem, where a number of actions with associated costs can be performed to navigate through the search space:

- If **substitution** is performed for two elements x_i and y_j of sequences x_1^N and y_1^M, then these two elements will be mapped to each other in the final alignment and labelled with [=]. This action includes the special case of identity where $x_i = y_j$ with zero cost (unlike 'true' substitutions).
- **Deletion**, on the other hand, results in element x_i of sequence x_1^N being mapped to the empty element, i.e., it will not have a corresponding element of sequence y_1^M in the alignment. Deletions are labelled with [<].
- **Insertion** is symmetric to deletion and as such leads to y_j being mapped to the empty element. Insertions are labelled with [>].

For each pairing (x_i, y_j) out of $x_1^N \times y_1^M$, a scoring function is invoked that evaluates the respective costs for each of the three available actions. Dynamic programming is applied to find the cheapest path (i.e., the cheapest sequence of actions) through the search space in $O(NM)$ time, where N and M are the length of x_1^N and y_1^M, respectively. This approach allows to factor out all domain-specific aspects to the scoring function by, e.g., assigning special scores to formatting marks while the dynamic programming scheme for cost minimisation remains untouched.

A common phenomenon that can be observed in such alignments are mismatches caused by recognition errors involving splitting or merging of words (segmentation errors) within the recognised texts or massive reductions due to fast speech (cf. figure 4.2 and 4.4). To account for these problems, the alignment has been extended to handle multiple levels of segmentation. Since the alignment procedure operates on sequences of tokens, it can be applied recursively to any pair of tokens that has been further split to a finer level of segmentation. Multi-word expressions or grammatical units can thus be reduced to sequences of single words which in turn can be broken down to sequences of syllables. The sequence of alignment labels obtained from these alignment processes are concatenated into a single alignment label, expressing the amount of overlap on submatching level between the parent tokens.

For the purpose of creating a literal transcript, it is crucial that all *corresponding* passages of the two input documents are mapped to each other. *Corresponding* means that two passages denote the same section in the actual dictation. Naturally, the two passages need not necessarily consist of the same tokens like, e.g., in a mismatch region. Figure 4.2 illustrates this problem for a sample text passage. The standard Levenshtein algorithm with equal costs for all edit operations calculates the minimum cost alignment based on the orthographic spelling, however, at the expense of proper word correspondences. Furthermore, the mismatch region is even split improperly at the wrong comma, such that the semantic correspondence between written and recognised text is lost. The SPARC alignment re-establishes the proper word correspondences even for segmentation mismatches and preserves a singular contiguous mismatch region.

written text	↔	*recognised text*
.	COR	.
\n\n/DIET/ \n	=	\n\n
Low-fat	=	Diet
,	=	is
low-cholesterol	=	a
	>	low
	>	fat
,	COR	,
two-gram	=	low
	>	cholesterol
	>	2 grams
sodium	COR	sodium

written text	↔	*recognised text*
.	COR	.
	>	\n \n
\n\n/DIET/ \n	=	Diet
	>	is
	>	a
Low-fat	=<	low
	<=	fat
,	=	,
low-cholesterol	=<	low
	<=	cholesterol
,	<	
two-gram	=	2 grams
sodium	COR	sodium

FIGURE 4.2. A sample text passage with mismatch regions highlighted in boxes, aligned with standard Levenshtein alignment (left) and the advanced multi-alignment computed by SPARC (right). Labels are: COR for identical words, [=] for corresponding/substituted words, [<] for deletions, and [>] for insertions.

Hence, two scoring mechanisms have been developed that compare token pairs for semantic and for phonetic similarity, respectively, and these have then been united in a single scoring function. Naturally, it would be desirable to not only compare token pairs, but whole passages for similarity in the scoring functions. However, the restriction to token pairs is a necessary concession to the already unfavourable computational complexity of alignment problems. Less local comparisons can be performed at the reconstruction stage.

4.4 Similarity measures

Similarity measurement of tokens is used in both text alignment (cf. section 4.3) and reconstruction (cf. section 4.5). For text alignment, the similarity measures are consulted by scoring functions of the generalised Levenshtein alignment algorithm to improve accuracy in contrast to plain orthographic matching. In text reconstruction, the measures are used to condition reconstruction rules and perform the classification of text chunks as either matches, corrections, reformulations, or a combination thereof (cf. section 4.1, table 4.1). The basic methods, however, are the same in alignment and in reconstruction.

4.4.1 Semantic similarity

In order to measure semantic similarity, words are first assigned a semantic representation. Since our primary application domain is medical reports, specialised medical terminology has to be incorporated into the knowledge sources. The resource we employ for that purpose is the Unified Medical Language System (UMLS, [70]), which includes a metathesaurus, a semantic network, and a lexicon (SPECIALIST). The morphosyntactic information from the lexicon was worked into the finite-state transducer that is used as a morphological lexicon.

The metathesaurus is a very large, multi-purpose, and multi-lingual terminology database that contains information about biomedical and health related concepts, their various names, and the relationships among them. Unfortunately, the relations between UMLS concepts appear to depend on the particular knowledge source the concept comes from, and the depth it is modeled solely within that knowledge source. Nevertheless, for analysing synonymity of two words or determining a rough degree of semantic relatedness, these relations appear to be sufficient. In addition, all concepts in the metathesaurus are assigned to at least one semantic type from the UMLS semantic network.

Furthermore, a high coverage resource for general vocabulary, the WordNet lexical database [26] is available for English. In Wordnet, nouns, verbs, adjectives and adverbs are organised into synonym sets, each representing one underlying lexical concept; the relations connecting WordNet synsets are quite different from the relations between UMLS concepts. For our purpose, the hypernym relation is the most important synset relation.[2].

The following ordinal scale has been defined in order to obtain a rough measure of semantic similarity of two words:

7	identical (modulo case)
6	same root (only inflection)
5	synonymous
4	morphologically derived
3	conceptual siblings
2	same UMLS semantic type or `parent(word1,word2)` or `parent(word2,word1)`
1	direct hierarchical relation between semantic types
0	no similarity at all

In the above context, `parent(word1, word2)` means that `word1` maps to a concept/synset (inter alia) that is a direct UMLS superconcept or hypernym synset of one of the concepts/synsets `word2` maps to. Two words are siblings if they share at least one direct UMLS superconcept or hypernym synset. The intuition behind this was to use a measure which is available in both WordNet and UMLS, which has a finer granularity than the (rather crude) UMLS semantic type and which assures that both concepts have something in common (the "supertype").

Based on the similarity value of its two argument tokens on the ordinal scale, costs for substitution, insertion and deletion are determined by the semantic scoring function and returned to the invoking alignment framework.

4.4.2 Phonetic similarity

Phonetic similarity measurement [90] requires three sources of information for comparison: the phonetic symbol sequence from the recognised text, the orthographic word sequence from the recognised text and the word sequence from the written text. The basic similarity measurement process is depicted in figure 4.3, and its main components are explained in more detail below.

Automatic phonetic transcription (APT)

In a first step, the written text is transferred to the phonetic domain with automatic phonetic transcription (APT). This is done by a simple lexicon lookup. The phonetic lexicon we used contains 160,000 words with 197,000 pronunciations. It includes common as well as domain-dependent vocabulary and was compiled from customary and publicly available resources like CMUdict[3]. To improve coverage on formatted text parts, a de-formatting grammar is applied to formatted text units. The de-formatting grammar is an inverted version of a formatting

[2] For a study which compares WordNet and UMLS in greater detail, see [14]

[3] See `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

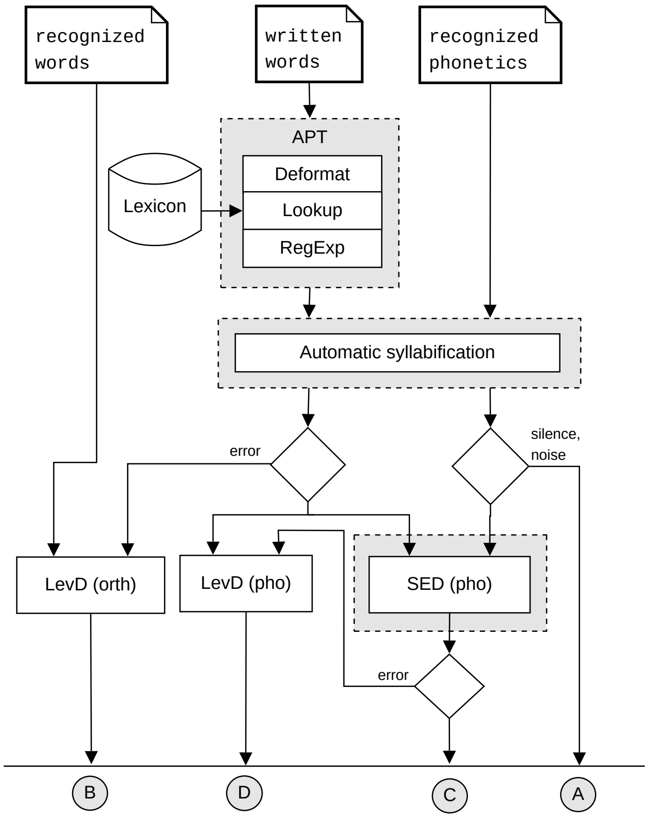


FIGURE 4.3. Block scheme of phonetic similarity function: automatic phonetic transcription (APT), automatic syllabification, trainable string edit distance measure (SED), and Levenshtein measure (LevD).

grammar used in the speech recogniser which now produces speaking variants for a given formatted entity as shown in the following example:

```
December 6  →  December the sixth
               December O six
               sixth of December
               ...
```

Furthermore, a simple regular expression syntax was defined to encode the possibly many speaking and pronunciation variants in a single string. The extended syntax allows grouping and alternation ("|") of expressions as described in the corresponding BNF grammar:

$$\begin{aligned} expr &:= group^{+} \\ group &:= \text{“(”}\ word^{+}\ (\ \text{“|”}\ word^{*})^{*}\ \text{“)”} \\ word &:= [A, .., Z, a, .., z] \end{aligned}$$

Since the word after the alternation-operator | is optional, whole words may be omitted. This is particularly useful for dealing with hesitations or dictated formatting instructions which do not appear in the written text by definition.

The recognised text still contains non-speech events like silence or noise markers which do not have a phonetic transcription and which are not contained in the written text either. These parts get scores assigned which automatically force them to be marked as insertions (path A in figure 4.3). After that, it is certain that the remaining string pairs are valid phonetic strings that can be handled by the phonetic similarity measurement model. Whenever the APT fails, phonetic matching is impossible, so the string pair can only be matched in the orthographic domain with the Levenshtein measure (path B in figure 4.3).

Automatic syllabification

Syllable boundaries are usually best assigned by expert phoneticians or can be inferred from stress markers stored offline in the lexicon. For the highly specific vocabulary used in the medical domain, such annotated expert phonetic lexica were not available to us. Furthermore, the vocabulary is subject to change over time, as new medication may be prescribed or medical treatments and measures may change. Therefore, an online automatic syllabification algorithm was implemented to determine syllables directly from the texts. The algorithm introduced by [43] is based on Optimality Theory [95], where phonological processes are modelled by applying ranked constraints on base forms to obtain surface forms. For syllabification, this means that a number of competing syllabification constraints are applied to the input words. In contrast to [43], the 'noonset' constraint had to be removed, as primary stress information was not available in the phonetic lexicon. The modified algorithm was tested on a sample set of 100 randomly selected words which were manually compared to a reference syllabification provided by Merriam-Webster's online dictionary[4]. The modification degraded the performance of the algorithm in terms of accuracy of the syllable boundaries, but not the number of detected syllables, and still returned correct results in around 80% of all cases.

With this algorithm, the word level units for recognised and written text are split into sequences of syllables. The alignment algorithm is then applied recursively on the syllable sequences. Adjacent words are not only aligned, but also tested for overlap on syllable level. The word-level alignment label is therefore replaced by an overlap symbol string. The resulting alignment expresses both word and syllable level correspondences. Consider the sample alignment in figure 4.4. Within the first mismatch region, the word `Charcot` was incorrectly recognised and split into `sharp` and `cold`. The syllable level alignment, however, shows that `sharp` corresponds to the first, and `cold` to the second syllable of `Charcot`. As syllable alignment is determined based on phonetic similarity, the alignment may sometimes look confusing. The short words `of` and `in` are not aligned with each other, since `in` is phonetically more similar to the last syllable of `ulceration` than to `of`.

Training a string edit distance measure (SED)

The main component of the phonetic scoring function is a trainable string edit distance measure based on the stochastic model presented by [103]. In this model, a string pair (x, y) is represented by all sequences of edit operations z_i which produce that pair. Assuming that each pair can be produced by at least one edit sequence, the probability of the pair is the sum of the probabilities of all edit sequences for that pair:

$$p(x, y|\theta) = \sum_{\{z^n\#:v(z^n\#)=\langle x,y\rangle\}} p(z^n\#|\theta) \ , \tag{4.1}$$

where $\#$ is the sequence termination symbol and $v(z^n\#)$ defines the set of all terminated edit sequences producing $\langle x, y\rangle$. Since every z_i has a probability $p(z_i)$ assigned and the model

[4] See `http://www.merriam-webster.com/`

written text		↔	*recognised text*	
a		COR		a
Charcot	SAr·k@t	=<	SArp	sharp
--	--	>=	koUld	cold
foot	fUt	=	fUt	foot
,	kAm@	<	--	--
though	DoU	=	nO	no
there is	Der= Iz	COR	Der= Iz	there is
no	nO	COR	nO	no
ulceration	Vl·s@·reI·S@n	====	Ql·t@·reI·S@n	alteration
--	--	<<<=	In	in
of	Vv	<	--	--
skin	skIn	COR	skIn	skin

FIGURE 4.4. A sample alignment containing two mismatch regions. The re-aligned mismatch regions are highlighted in boxes while identical words are labelled with COR. Phonetic strings are in SAMPA notation and syllable boundaries are marked with dots [·]. Note that the [=]-overlap symbol just indicates correspondence, not equality of syllables, in contrast to the insertion [<] and deletion [>] symbols which label non-matching syllables.

is memoryless, $p(z^n\#|\theta)$ is the product of the probabilities of the single edit operations. These probabilities $p(z_i)$ are learned from a corpus of predefined, similar string pairs with an EM algorithm ([103]). Accumulating the probabilities for all edit sequences, a similarity measure can now be defined as

$$d(x,y) = -\log\ p(x,y|\theta)\ . \tag{4.2}$$

Two issues should be noted at this point. First, the similarity value decreases exponentially with the input string length due to the usage of the distinct termination symbol #. Therefore, the similarity value needs to be normalised – in this case by the sum of the input string lengths. Furthermore, the similarity measure is never zero since each edit operation has assigned a probability $0 < z_i < 1$. To still be able to detect exact matches, the systematic bias is subtracted symmetrically to normalise the measure to zero according to the following formula:

$$d_0(x,y) = d(x,y) - \frac{1}{2} \cdot [d(x,x) + d(y,y)] \tag{4.3}$$

Prior to matching, the regular expressions generated by the automatic phonetic transcription have to be expanded again, as only the minimum score for all possible realizations is returned (path C in figure 4.3). Finally, in case the stochastic model fails, another fallback to the Levenshtein measure is done, this time with phonetic strings (path D in figure 4.3).

The model was trained in 3 EM iterations with a set of 13,383 string pairs obtained from manual narrow phonetic transcriptions of a domain-specific corpus of 272 medical reports. The transcriptions were done by English students with specific training in phonetics, ensuring quality in the transcription process. For each word in the transcription, a string pair consisting of the canonical transcription obtained from the phonetic lexicon and the actual phonetic transcription was compiled. This way, phonetic similarity is clearly defined, and frequent phoneme confusions can be learned easily from real-world data.

Figure 4.5 displays the learned probability distribution for each edit operation defined on a phonetic symbol pair. As expected, most of the probability mass was assigned to

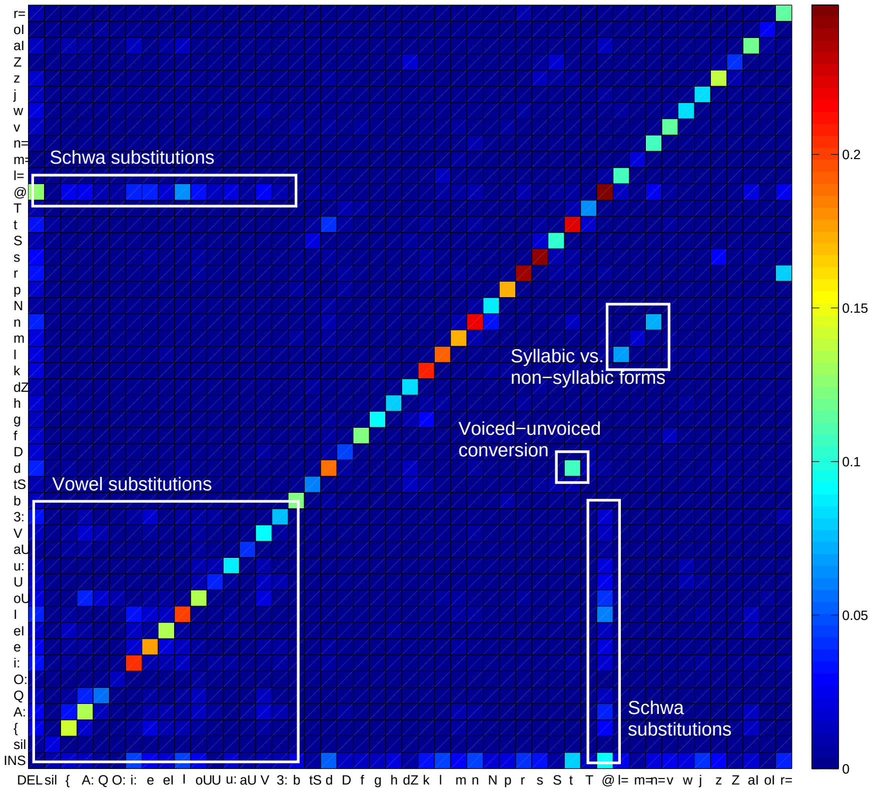


Figure 4.5. Learned probability distribution for edit operations z_i after 3 EM iterations. Phonetic symbols are in SAMPA notation.

identity operations (main diagonal). Furthermore, vowels were likely to be substituted by *schwa* (/@/) and vice versa. Voiced-unvoiced substitutions between /t/ and /d/ were also quite prominent, just like substitutions between the syllabic (/n=/, /m=/, /l=/) and non-syllabic forms (/n/, /m/, /l/) of the semi-vowels. The learned probability distribution clearly reflects the phonetic knowledge that can be observed in dictated speech.

4.4.3 Combined similarity measurement

The semantic and phonetic scoring functions are used as building blocks for a combined scoring function that best exhibits the behaviour that is required for further processing.

The goal is to align any two sequences of elements for which phonetic or semantic similarity can be assigned. Distinguishing between phonetic and semantic similarity is postponed to the reconstruction process since it is the single aim of this processing stage to put related elements into proper correspondences.

Combining the two sets of scores for substitution, deletion and insertion into a single set of scores is somewhat subtle, because contradictory actions might be suggested by semantic and phonetic similarity scores. As an example, phonetic scoring might vote for substituting two elements, while semantic scoring might want to substitute one of these elements with a

written text	↔	*recognised text*	*deformatted*	*reconstructed*	*rule*
He	=	he	he	he	sim_bigram
says	=	said	said	said	sim_bigram
he	COR	he	he	he	identical
did not	COR	did not	did not	did not	identical
have	COR	have	have	have	identical
any	COR	any	any	any	identical
cardiac	COR	cardiac	cardiac	cardiac	identical
	>	,	comma	--	–
	>	residual	residual	residual	repetition
residuals	COR	residuals	residuals	residuals	identical

FIGURE 4.6. Reconstruction of a text passage with two mismatch regions (dashed boxes): Written text, alignment labels, and recognised text are given as input. Deformatted recognised text, reconstructed text, and the matching rule for each alignment line are generated by the system.

different one. Such contradictions need to be resolved while still following the overall goal of performing substitution (mapping between two elements) when either the phonetic or semantic measure indicate similarity.

The combined scoring function for alignment was developed and tuned heuristically by manual inspection of a small number of alignments. In general, the phonetic similarity function analyses the tokens on a high level of detail and thus establishes correspondences in a greedy fashion which sometimes results in alignments that cannot be interpreted meaningfully any more. The semantic similarity scoring function on the other hand, is more robust against "over-correspondencing" but at the same time not capable of properly detecting fine matches. For these reasons, semantic matching is applied in the first place to filter out clear cases and avoid overstretched regions of correspondence, before phonetic matching is used to find detailed matches.

4.5 Text reconstruction

Based on the alignment, a reconstruction hypothesis for a literal transcription can be computed. In general, this process can be seen as a classification task, as already outlined in section 4.1 (cf. table 4.1). A classifier is used to select the recognised or the written text for each alignment token. For optimal control and fine-tuning, we implemented a rule-based reconstruction system that allows generic and context-dependent analysis of the alignment. This approach is also compared to state-of-the-art automatic classification approaches using the same input features.

The rule-based reconstruction process, which is described in [51], operates on the established alignment. The steps performed for reconstructing the actually spoken words are the following (cf. figure 4.6):

- **Deformatting**:
  First, a column containing the completely deformatted variant of the recognised words is created (cf. section 4.4.2). In particular, formatted items and punctuation are replaced by the most likely spoken variant based on the phonetic representation and the measures for phonetic and semantic similarity.

written text	↔	*recognised text*
HEART:	=	Heart
Examination	=	examination
is	<	
normal	COR	normal
.	COR	.
LUNGS:	=	Lungs
	>	are
Clear	=	clear
.	COR	.

FIGURE 4.7. Excerpt of aligned input sequences with sliding rule window indicated by solid frame.

- **Identifying and retracing moved blocks**:
  Then, moved blocks are identified if there are any, and within the written text the identified text blocks are actually moved to the place where the corresponding text is assumed to have been dictated. The moved regions are then realigned, such that the result of this (and the previous) step is a new alignment column.

- **Application of reconstruction rules**:
  Reconstruction rules specified by the user are applied to this alignment, and two additional columns are created: one containing the reconstructed words and another one consisting of a justification (i.e., the responsible rule) for that reconstruction.

- **Reconstructing moved blocks**:
  Finally, the moved parts of the report are reinserted in order to resemble the original input.

4.5.1 Rule engine

Once a stable alignment has been established, knowledge about corresponding passages can be used for inspecting tokens and their contexts both in the edited document and the output of the speech recognition system.

For this purpose, a rule engine has been developed. The reconstruction rules that are interpreted by this engine provide a mechanism for inspecting a sliding window that is moved over multiple columns according to their alignment. In addition to columns for the edited document (cf. figure 4.7, left side) and the output of the speech recogniser (cf. figure 4.7, right side), a so-called "alignment" column is available that indicates the correspondence between the left and the right side at the current element: "=" indicates that some kind of similarity has been found between the left and the right side, and therefore a substitution has been performed, whereas "<" indicates a deletion and ">" indicates an insertion. In the case of deletion, there is no element on the right side corresponding to that on the left side. Symmetrically, there is no element on the left side if the alignment column contains an insertion label.

Figure 4.7 depicts aligned columns and the sliding window of a rule that is used to inspect the column elements and their context at a certain position in the input. For each rule, a regular expression is applied to the alignment column, which specifies a dynamic sliding window size. In the example above, the regular expression might have been formulated in such a way that the sliding window iterates over instances of consecutive lines labelled with

"=", with the intention of inspecting only whole blocks of elements for which some kind of similarity has been found.

Each rule adheres to the following skeleton:

```
rule rulename
match -w/window regexp/
 # inspect sliding window
do
 # specify reconstruction result
done
```

As explained above, the "window regular expression" works on the string of labels in the alignment column and specifies for which lines the rule window is set up. The `match` block can then be used to inspect all columns within the borders of the window. If the rule finds that the lines inside the window exhibit a phenomenon that this rule can handle, a non-zero value is returned in the `match` block, which causes the `do` block to be triggered. The `do` block is then responsible for building a literal transcription of the matching lines and writing it to a result column.

The advantage of this approach is that each phenomenon (like recogniser errors, repetitions, etc.) can be handled by a separate rule which encapsulates both the detection of such cases as well as the required knowledge to decide which column should be used or which transformations have to be applied to build an appropriate literal transcription for the current window.

The bodies of the `match` and `do` blocks can be freely expressed in regular Perl code. In addition, some special built-in functions for measuring phonetic and semantic similarity between two strings, and for converting formatted expressions into their most likely spoken variant (e.g.: `500 mg` → `five hundred milligrams`, cf. section 4.4.2) are available in these blocks.

Since more than one rule can match for a certain sequence of alignment labels, rules match on a first-come first-serve basis, meaning that rule precedence influences the result. In the experiments (cf. section 4.6), the effect of rule ordering is investigated explicitly.

4.5.2 Rule definitions

To test the effects of the previously described techniques, we specified reconstruction rules, where an alignment label is either the identity edit operation (COR) or a sequence of alignment labels $[=, <, >]$ (cf. figure 4.8). The rules can be grouped into three categories: baseline rules, semantics-based rules, and phonetics-based rules.

Baseline rules are the three simple starting points for the hypothesized reconstruction that do not require any advanced processing:

- **Baseline**: only identical words in the alignment (COR) are reconstructed, mismatch regions are ignored.
- **Recognised-only (REC)**: for each alignment label, always select the *recognised text* for reconstruction.
- **Written-only (WRI)**: for each alignment label, always select the *written text* for reconstruction.

written text	↔	*recognised text*	**CTX**$_{pho}$	**OVG**$_{pho}$	**OVS**$_{pho}$	*reference text*
a	COR	a	a	a	a	a
Char·cot	=<	sharp	Charcot	Charcot	Charcot	Charcot
	>=	cold				
foot	=	foot	foot	foot	-	foot
,	<		-	,	-	
though	=	no	though	-	-	though
there is	COR	there is	there is	there is	there is	there is
no	COR	no	no	no	no	no
ul·ce·ra·tion	====	al·te·ra·tion	ulceration	ulceration	-	ulceration
	<<<=	in			-	
of	<		-	of	-	of
skin	COR	skin	skin	skin	skin	skin

FIGURE 4.8. A sample alignment containing two mismatch regions (dashed boxes), together with reconstruction rule results. Syllable boundaries are marked with dots [·]. Note that the [=]-overlap symbol just indicates correspondence, not equality of syllables, in contrast to the insertion [<] and deletion [>] symbols which label non-matching syllables. The solid boxes highlight lines affected by each rule, dashes [−] mark parts not covered by the rule.

Semantics-based rules implement semantic knowledge in the reconstruction process. With regard to the initial assumption that reformulations are semantically similar, semantic rules select the *recognised text* for reconstruction, as soon as the rule matches.

Phonetics-based rules on the other hand try to detect corrections of speech recognition errors in the alignments. Therefore, they select the *written text* for reconstruction whenever the rule matches. As recognition errors are more likely to occur than reformulations, these rules should match more often than the semantic rules.

The following types of rules were defined for both semantic and phonetic similarity separately as indicated by subscripts in section 4.6:

- **Context (CTX)**: matches sequences of 1, 2, or 3 alignment labels containing at least one submatching label (=), if similarity is higher than threshold t. The idea behind this rule is that longer corresponding regions in the alignment are more likely to be real correspondences.
- **Overlap, greedy (OVG)**: matches sequences of 2 or 3 alignment labels, where inserted or deleted submatching labels (</>) are either preceded or succeeded by at least one matching label (=), if similarity is higher than threshold t. This rule collects all word sequences showing any possible overlap at submatching level without regard of the matching order.
- **Overlap, selective (OVS)**: matches sequences of 2, 3, or 4 alignment labels, where submatching labels (=) are first succeeded by insertion (<), and then preceded by deletion (>) labels if similarity is higher than threshold t. This pattern is typical for segmentation errors in the recognised text.

Figure 4.8 illustrates the effect of each rule on a sample alignment for the phonetic similarity case. The **context** rule is activated whenever a group of matching syllables appears. Still, it is not enough as it does not handle stand-alone insertions or deletions appropriately. The **greedy overlap** rule can handle insertions and deletions whenever they appear in terms of a syllable overlap. However, it is not activated when there is a direct match (`though` ↔

`no`). The **selective overlap** rule, finally, matches only the precise first segmentation error, where the syllable counts exactly match. Accidental matches are therefore impossible. This example indicates that combination of rules may be beneficial.

4.6 Experiments

The text reconstruction process was evaluated for two different tasks to examine the performance of the SPARC method. First, the quality of the reconstruction was tested. For this test, a literal transcription was reconstructed and compared to a manual reference transcription for a set of medical reports. We define the evaluation as a text retrieval task, because the results reflect how much of the original text can be reconstructed and how much of the reconstructed text is actually part of the original text. This test is a true performance measure of the system, without considering any particular application. The performance of the main components – semantic and phonetic similarity measurement, and text reconstruction – will be evaluated separately in section 4.6.1.

Second, the speech recognition performance using reconstructed texts is measured (cf. section 4.6.2). In this test, the language model of the speech recogniser producing the recognised texts was re-trained with the reconstructed texts and tested on an independent test set. This test is only an indirect performance measure and is intended to demonstrate the applicability and impact on the speech recognition process. For this reason, we decided to test with a commercially available ASR system instead of an academic one and did not perform specific parameter tuning to keep the results more independent from the actually used ASR system.

4.6.1 Reconstruction quality

For measuring reconstruction quality, we report results in terms of the metrics $Recall = \frac{|COR|}{|COR|+|MISS|}$, $Precision = \frac{|COR|}{|COR|+|WRONG|}$, and their harmonic mean $F1$, where |COR| is the number of reconstructed words with perfect correspondence in the reference text, |MISS| is the number of words in the reference text without correspondence in the reconstructed text, and |WRONG| is the number of reconstructed words without correspondence in the reference text [120].

The evaluation corpus consisted of 735 written and recognised texts of about 335,000 tokens, as well as manually transcribed reference texts for validation of the hypothesized reconstruction. The texts were selected such that they equally represent three ranges of average word error rates (WER) for the recognised text compared to a manual reference transcription. Hesitations and incomplete words were removed beforehand to avoid biased results.

Semantic and phonetic similarity measurement

The impact of semantic and phonetic similarity measurement is studied by evaluating semantic and phonetic reconstruction rules separately before they are joined in a single system. For the phonetic rules, previous results from [91] are summarised here, while for the semantic rules and the joint system, entirely new results are presented.

We start with the evaluation of the semantic rules in table 4.2. The first group covers the baseline rules (Baseline, REC, WRI), while the CTX$_{sem}$, OVG$_{sem}$, and OVS$_{sem}$ systems

TABLE 4.2. Reconstruction results in % for semantics-based rules (second block) in comparison to baseline systems (first block). Best results for each row grouping are boldface.

	5-13% WER			**20-25% WER**			**40-45% WER**		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	**100**	78.9	88.2	**99.9**	64.6	78.4	**99.5**	46.0	62.9
REC	83.3	**93.4**	88.1	79.0	85.7	82.2	66.7	71.9	69.2
WRI	92.8	93.1	**92.9**	89.9	**89.6**	**89.8**	85.9	**85.4**	**85.6**
CTX$_{sem}$	98.6	87.6	**92.8**	97.3	76.9	85.9	95.6	60.6	74.2
OVG$_{sem}$	99.7	80.2	88.9	99.4	66.6	79.8	98.7	47.9	64.5
OVS$_{sem}$	**99.8**	79.2	88.4	**99.7**	65.0	78.7	**99.0**	46.5	63.3
all$_{sem}$	98.6	**87.7**	**92.8**	97.2	**77.0**	**86.0**	95.5	**60.8**	**74.3**

TABLE 4.3. Reconstruction results in % for phonetics-based rules (second block) in comparison to baseline systems (first block). Best results for each row grouping are boldface.

	5-13% WER			**20-25% WER**			**40-45% WER**		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	**100**	78.9	88.2	**99.9**	64.6	78.4	**99.5**	46.0	62.9
REC	83.3	**93.4**	88.1	79.0	85.7	82.2	66.7	71.9	69.2
WRI	92.8	93.1	**92.9**	89.9	**89.6**	**89.8**	85.9	**85.4**	**85.6**
CTX$_{pho}$	97.6	90.4	93.8	95.4	82.8	88.7	93.1	69.8	79.8
OVG$_{pho}$	97.9	86.4	91.8	95.8	78.3	86.2	93.1	65.7	77.0
OVS$_{pho}$	**99.8**	79.5	88.5	**99.6**	65.6	79.1	**98.8**	47.3	64.0
all$_{pho}$	97.0	**91.1**	**94.0**	94.7	**84.3**	**89.2**	92.1	**72.6**	**81.2**

of the second group represent semantic context and overlap. The combination of all rules is denoted by all$_{sem}$.

The recognised text (REC) is not a good starting point for reconstructing a literal transcription. Although the recall scores are comparable to the other methods, many errors stem from the recognition process, resulting in poor precision. The written text (WRI) is more reliable for the domain of medical dictations.

Using semantic context (CTX$_{sem}$) for reconstruction returns accurate results with higher precision than recognised-only (REC) or written-only (WRI) reconstruction and significantly higher recall than the baseline system. This holds even more for the overlap rules (OVG$_{sem}$, OVS$_{sem}$): whenever semantic overlap is detected, it is almost always correct. Unfortunately, the recall scores are only 0.4% - 2.0% absolute higher than the baseline scores, indicating a low number of matches for these rules. In sum, neither the separate semantic rule systems nor their combination is able to exceed the baseline systems for any of the WER ranges.

The threshold value for semantic similarity measurement can take values between $t = 0$ (no similarity) and $t = 7$ (identity) and was varied from $t = 1$ to $t = 7$ in the experiments. The resulting curves are plotted in a Recall/Precision diagram (cf. figure 4.9). Adjusting the semantic similarity threshold does not contribute much to the overall performance. The trade-off between recall and precision is almost linear, as is shown by the graphs in figure 4.9. The best recall/precision value pairs were obtained for a similarity threshold value $t = 5$ for all WER ranges.

Likewise, we evaluated the phonetic rules separately and in combination compared to the baseline systems. Table 4.3 summarises the results.

In the phonetically controlled reconstruction contextual information (CTX$_{pho}$) returned better F1 scores than in the semantically-controlled reconstruction. Only for the low WER

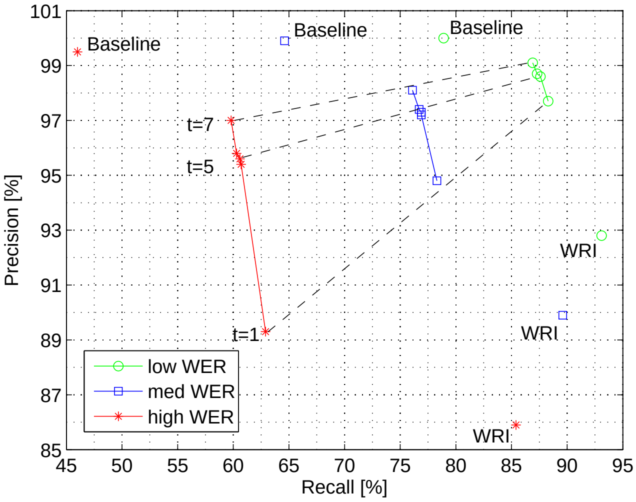


FIGURE 4.9. Recall/Precision diagram derived from the all$_{sem}$ system by varying the semantic similarity threshold t between $t = 1$ and $t = 7$ for high, medium, and low WER texts.

case, however, a gain of 0.9% absolute can be observed in contrast to the written-only (WRI) reconstruction. The greedy exploration of overlap on syllable level (OVG$_{pho}$) returned surprisingly precise results which are absolutely comparable to using contextual information. This applies even more to the selective overlap rule (OVS$_{pho}$), which has only very little gain in recall in comparison to the baseline, but almost maximum precision. These findings indicate that the combination of these rules could be beneficial. The combination of all rules shows the best performance for all WER ranges.

The threshold value for phonetic similarity measurement can be adjusted between $t = 0.0$ (no similarity) and 10.0 (identity) and was varied from $t = 5.0$ to 10.0 in the experiments. Like for semantic similarity measurement, the resulting curves are plotted in a Recall/Precision diagram, shown in figure 4.10. Optimising the threshold value for phonetic similarity also contributes to the overall performance. The trade-off between recall and precision is not linear, as the graphs in figure 4.10 show. The best recall/precision value pairs were obtained for a similarity threshold value $t = 8.0$ for all WER ranges.

The SPARC method tries to combine knowledge about semantic and phonetic similarity to detect matches, corrections, and reformulations in the data (cf. section 4.1, figure 4.1). For this reason, the best semantics- and phonetics-based systems were combined into a single system. As mentioned before, the rule engine is sensitive to rule precedence, so there are several possible combinations. Thus, the impact of semantic and phonetic knowledge in the reconstruction process can be estimated. Table 4.4 lists the results for the given combinations: the I+S and I+P systems are combinations of the baseline and the all$_{sem}$/all$_{pho}$ systems, where the results are taken from tables 4.2 and 4.3, respectively. The I+S+P and I+P+S systems are combinations of the baseline, semantic and phonetic systems with the given rule precedence.

In terms of reconstruction performance, the combination of semantic and phonetic rules leads to improvements in recall without major losses in precision, resulting in gains in F1. The semantic system improves significantly (1.4% to 7.4% relative) while the phonetic system improves only slightly (0.1% to 0.25% relative). The best results are obtained when phonetic rules are given precedence over semantic rules. The detailed statistics on rule matching

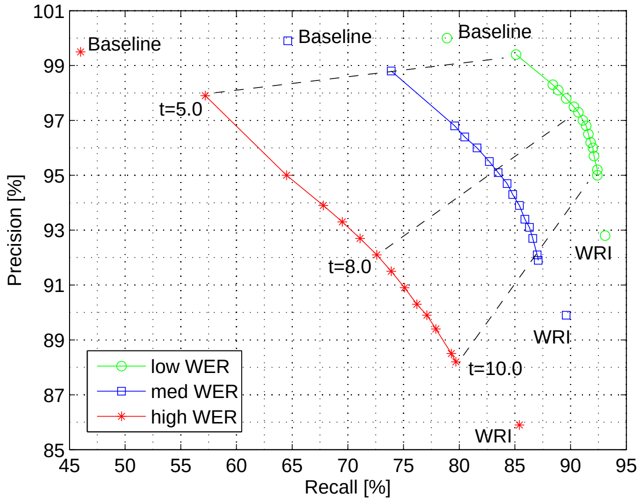


Figure 4.10. Recall/Precision diagram derived from the all$_{pho}$ system by varying the phonetic similarity threshold t between t = 5.0 and t = 10.0 for high, medium, and low WER texts.

Table 4.4. Reconstruction results in % for combinations of the baseline identity (I), semantics- (S), and phonetics-based (P) rules. Rule precedence is indicated by the order of the rule addition terms. Best results for each column are boldface.

	5-13% WER			**20-25% WER**			**40-45% WER**		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
I+P	97.0	91.1	94.0	94.6	84.2	89.1	92.1	72.5	81.1
I+S	**98.6**	87.6	92.8	**97.3**	76.9	85.9	**95.6**	60.6	74.2
I+P+S	97.0	**91.3**	**94.1**	94.5	**84.6**	**89.3**	91.9	**72.8**	**81.3**
I+S+P	97.9	90.5	**94.1**	95.9	82.7	88.8	93.4	69.5	79.7

counts in table 4.5 explain this observation. In about 70 to 80% of all cases, identical items are detected which are matched by the baseline identity rule. Semantic rule matches account for about 18% of all matches and phonetic matches for about 28%, when applied separately to the alignments. In combination, however, phonetic rules still match in about 8% of all cases after semantic matching, while semantic rules only match in 0.5% after phonetic rules have been applied. Therefore, it can be concluded that 8% of mismatches are of pure phonetic nature, only 0.5% of pure semantic nature, and the rest of about 17 to 18% can be explained in both semantic and phonetic terms.

Rule-based versus data-driven reconstruction

The rule-based reconstruction approach was compared to a data-driven approach to evaluate the classification performance. For data-driven text reconstruction, we use different classifiers to produce the hypothesized literal transcription which is the 2-class output of a classifier, i.e., either *written text* or *recognised text*. For classifier training, the class labels are produced by aligning the reference text with the written text. The features are derived from the automatic alignment and the phonetic similarity score (see section 3.1) computed for the aligned written and recognised phoneme strings. In addition, this score is derived for 3 consecutive phoneme

Table 4.5. Rule matching counts and percentages for the combined rule systems.

	TOTAL	**identical (I)**		**semantic (S)**		**phonetic (P)**	
	[1]	[1]	[%]	[1]	[%]	[1]	[%]
I+S	79,525	64,911	81.6	14,614	18.4	0	0
I+P	89,592	64,923	72.5	0	0	24,669	27.5
I+S+P	86,741	64,916	74.8	14,621	16.9	7,204	8.3
I+P+S	90,033	64,923	72.1	441	0.5	24,669	27.4

strings to model the dependency of adjacent words in the classifier. The remaining features are computed from the sequence of submatching alignment labels. Therefore, the sequence is split into 3 equal parts. After assigning values to the labels, i.e. $[=]\ldots 0, [<]\ldots -1, [>]\ldots 1$, the mean and standard deviation of each part serve as feature. The last feature denotes the length of the syllable symbol sequence. Hence, 9 features are used for the classifiers. The following classification approaches are used [9]:

- ***k*-NN**: k-nearest neighbour classifier. For the presented results $k = 9$.
- **NN**: Neural network (Multilayer Perceptron) with 3 layers. The number of neurons in the input and output layer is set to the number of features and the number of classes, respectively. The number of neurons in the hidden layer is set to 70. We use Levenberg-Marquardt backpropagation for training, a hyperbolic tangent sigmoid transfer function for the neurons in the input and hidden layer, and a linear transfer function in the output layer.
- **SVM**: The support vector machine with the radial basis function (RBF) kernel uses two parameters C^* and σ, where C^* is the penalty parameter for the errors of the non-separable case and σ is the parameter for the RBF kernel. We set the values for these parameters to $C^* = 1$ and $\sigma = 1.5$.

The optimal choice of the parameters, kernel function, number of neighbours, and transfer functions of the above mentioned classifiers has been established during extensive experiments. Five-fold cross-validation is used to produce the results with the classifiers. Throughout our experiments, we use exactly the same data partitioning for each training procedure.

Table 4.6 lists the data-driven systems k-NN, NN, and SVM in comparison to the best combined rule-based system I+P+S. Both, rule-based and data-driven reconstruction use the same input features derived from the alignment labels, semantic, and phonetic similarity scores.

The data-driven systems are closer to the written text only (WRI) reconstruction than the rule-based system, showing improvement in precision for all WER ranges. The rather simple k-NN classifier consistently produces the highest precision while the more complex NN and SVM classifiers achieve higher recall scores. The rule-based system outperforms the data-driven system only for low error rates.

The selection of either rule-based or data-driven reconstruction framework depends on the intended application. The definition of rules allows the precise control of the reconstruction process and specific fine tuning for either high precision or high recall. Furthermore, it can be used efficiently for "labelling" a corpus of parallel recognised and written texts by applying specific rule configurations. The data-driven system, however, is better when the amount of reconstructed data needs to be maximised, particularly for the high WER condition. The main benefit is then that no handcrafting of rules and no tuning of similarity thresholds is required.

Table 4.6. Reconstruction results in % for baseline systems (first block), the best rule-based system (second block), and data-driven systems (third block). Best results for each column are boldface.

	5-13% WER			**20-25% WER**			**40-45% WER**		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	**100**	78.9	88.2	**99.9**	64.6	78.4	**99.5**	46.0	62.9
REC	83.3	**93.4**	88.1	79.0	85.7	82.2	66.7	71.9	69.2
WRI	92.8	93.1	92.9	89.9	**89.6**	89.8	85.9	**85.4**	**85.6**
I+P+S	97.0	91.3	**94.1**	94.5	84.6	89.3	91.9	72.8	81.3
k-NN	94.9	92.8	93.9	91.6	88.0	89.8	87.1	83.4	85.2
NN	94.9	93.0	93.9	91.4	88.5	**89.9**	86.6	84.0	85.3
SVM	94.8	93.0	93.9	91.3	88.6	**89.9**	86.6	84.4	85.5

4.6.2 Automatic speech recognition

The effect of using reconstructed texts for language model training was evaluated in the environment of a commercial backend speech-recognition system for telephone channel audio. The SPARC method was compared to the standard method for language model training and to a random generation of reconstructed text.

Language modelling approach

The standard approach of this ASR system for creating language models is to segment large corpora of written text into lexicon entries and to train trigram models on them. The mapping of written text onto lexicon entries is not a trivial process since formatted items like numbers, quantities, dates etc. cannot be directly represented as lexicon entries and give only little clue as to what a speaker would say to dictate such items. Written text further contains additions (including punctuation marks) and reformulations by transcriptionists and, therefore, does not represent what actually has been or will be dictated.

To handle the common cases of formatted numeric items ("grammar items") and inserted punctuation marks, a 2-stage decoding-rescoring strategy is applied:

- **Decoding:** An initial language model is trained on regular words and classes of grammar items. This language model cannot cover spoken forms of grammar items. Therefore, at decoding time, it is interpolated with an additional language model derived from grammars representing spoken forms of grammar items ("grammar language model").
- **Rescoring:** The emerging wordgraph is parsed for grammar items, and enriched with edges marked with the corresponding grammar classes. This wordgraph then is rescored using the initial language model. At the same time, punctuation marks are hypothesized.

The setup is robust in language-model adaptation as only the class of a grammar item needs to be determined, without guessing a word sequence that might have been spoken. Its weakness, however, lies in the imprecise grammar language model used at decoding time.

Table 4.7. Language model details: Number of tokens and out-of-vocabulary (OOV) rate.

Language model	Size	OOV rate
MultiMed	287M	0.9%
Classic (baseline)	51.92M	1.15%
Random as-spoken	56.37M	1.15%
Reconstructed	49.64M	1.34%

Experimental setup

We chose the domain of Clinical Reports as test case since there was enough data available, both to create reconstructed text (requiring written and recognised text) and to evaluate the performance (requiring reference transcriptions of what has actually been dictated). The baseline word error rate indicated a low-to-medium WER condition, thus the rule-based system with I+S+P rules was used for text reconstruction. This setup reconstructed 101,607 reports from 504 authors (between 45 and 507 reports per author) running in parallel on four standard personal computers for three weeks and produced a text corpus of 52 million words (cf. table 4.7), the equivalent of about 9,000 hours of sound.

The speaker-independent acoustic models of the ASR system have been trained on 200 hours (female speakers) and 300 hours (male speakers) of acoustic material recorded on a telephone channel with 4kHz bandwidth, Acoustic speaker-adaptation was performed using Maximum-Likelihood Linear Regression (MLLR) for the first 15 minutes of sound, and Maximum-A-Posteriori (MAP) adaptation for the rest of the available data, which was 10 hours of sound for each speaker in the test set.

For this domain, a large medical lexicon of 58,103 words was used, giving a high coverage on both the test set and the training corpus (OOV rates $< 1.5\%$).

Recognition tests were performed on a set of 239 reports from 2 female and 3 male authors (3 hours of sound per user), all in the domain of Clinical Reports[5], and all recorded through a telephone channel with 4kHz bandwidth. The best available acoustic references for these speakers were used. The baseline word error rate on our test set was 11.77% (cf. table 4.8).

Language models

The reconstruction of spoken words allows to avoid the use of language model classes for grammar items. To measure this effect and the quality of the resulting language models, we perform recognition tests with four different trigram language models:

1. **MultiMed:** A language model used in commercial applications, created from 287 million words of general medical reports. This language model requires an interpolated grammar language model at decoding time, since it trains grammar items as classes.
2. **Classic:** This language model is built from the corrected text of the reconstruction corpus (52 million words), and is otherwise consistent with initial language models used in commercial applications, requiring an interpolated grammar language model at decoding time.

[5] The available speakers for the domain of Clinical Reports do not cover all word error rate ranges of the previous experiments in section 4.6.1. For this reason, results are reported per speaker and on average, but not according to the previous separation into low, medium, and high word error rate conditions.

3. **Random as-spoken:** Same as classic, but instead of grammar classes, a randomly chosen spoken representation of that class is trained. This is a standard technique to get closer to what has been spoken, at least in the case of grammar items, and can be seen as a simple case of reconstruction. It does not require a grammar language model at decoding time any more.

4. **Reconstructed:** Reconstructed text produced by SPARC was slightly post-processed to match the lexicon: Phrases (i.e., multi-word expressions handled as single lexicon entry like "she is", "he had" etc.) are handled by SPARC as word sequences and were mapped back to single lexicon entries. Special words for punctuation marks were reintroduced (SPARC reconstructs dictated periods and commas as lexicon entries "period" and "comma", while the lexicon and the rescoring language model use special symbols).

The chosen evaluation method is biased against the SPARC reconstruction approach and leads to slightly worse results for two reasons:

- SPARC reconstructs the so called "demographic header", this is demographic patient information at the beginning of the dictation, which is not a part of the final report. Recognition performance on the demographic header is ruled out in all tests since recognition accuracy in this section is of no benefit for the user.

- The lexicon (and rescoring language model) makes a distinction between special words like "Lungs:" and "Heart:" versus "lungs" and "heart", respectively to be able to produce appropriately formatted output. SPARC always reconstructs the regular words; therefore, recognition accuracy is expected to be lower on these special words.

Results

The results are summarised in table 4.8. Using the reconstructed text language model reduced the overall word error rate from 11.77% to 10.86% which is a relative reduction of 7.74% compared to the baseline classic language model. The randomly generated as-spoken variants only lead to an overall relative reduction of 4.38%. Table 4.8 also shows that these improvements are consistent for all speakers. Both, the random as-spoken and reconstructed text language models even outperform the MultiMed language model which was created from substantially more data. Hypothesizing the spoken forms of grammar items is therefore beneficial for the applied 2-stage decoding-rescoring strategy.

Based on the findings from this first experiment, we conducted a second experiment where we gradually increased the corpus size for language model training from 1 million tokens up

Table 4.8. ASR results for the tested language models: Word error rate (WER) and relative difference (rel. △) to the Classic (baseline) model in [%].

Speaker	MultiMed	Classic	Random		Reconstructed	
	WER	WER	WER	rel. △	WER	rel. △
F1	7.68	8.25	7.93	-3.91	7.81	-5.36
F2	16.80	17.64	16.37	-7.18	15.42	-12.56
M1	16.79	17.37	16.76	-3.50	16.56	-4.67
M2	6.86	7.13	7.12	-0.28	6.57	-7.98
M3	9.33	8.87	8.46	-4.60	8.45	-4.80
Total	11.34	11.77	11.25	-4.38	10.86	-7.74

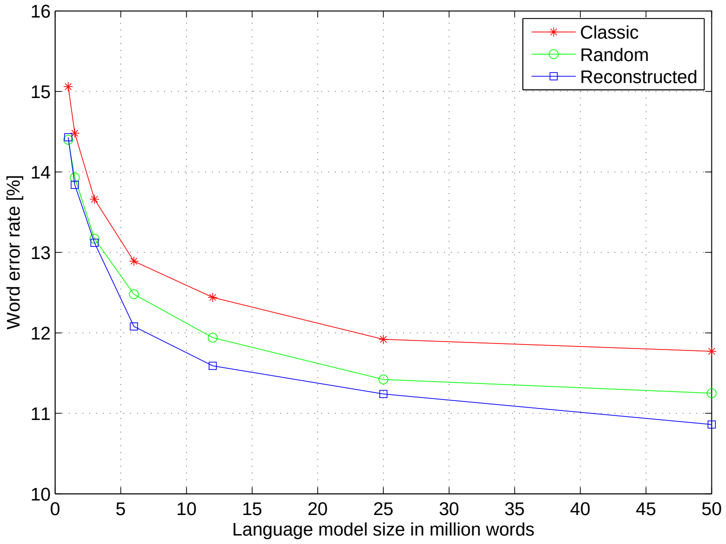


FIGURE 4.11. WER in [%] for increasing re-training text corpus size.

to the maximum size of about 50 million tokens. Figure 4.11 illustrates the evolution of the word-error rate with respect to the size of the language model training corpus. Up to a corpus size of 3 million words, there is not much difference between the models built on randomly generated as-spoken variants and reconstructed text. For a corpus size of 6 million tokens or more, the SPARC method performs consistently better. At 50 million words, the word-error rate begins to go into saturation, so incorporating more data will only have minor effects on the word-error rate.

Apart from the mentioned adjustments, the reconstructed texts were not further optimised or tuned for ASR purposes. Using the SPARC method without any further tuning immediately resulted in the reported improvements. Additional fine-tuning in terms of, e.g., the interpretation of punctuation or the exclusion of leading and trailing irrelevant text blocks in recognised texts may even further improve the performance.

4.7 Conclusion

We have described the SPARC method of semantics and phonetics based similarity measurement for the automatic reconstruction of medical dictations from draft recognised texts and final written reports. The resulting reconstructed texts can be used for various applications in language technology, including but not limited to acoustic and language model adaptation for automatic speech recognition, computer-aided document production in medical transcription, or generally for the development of parallel text corpora of non-literal text resources.

The method is based on an alignment between a draft speech recognition transcript containing errors and a formatted, corrected medical report that may have been paraphrased during the transcription process. The text alignment uses a model of semantic and phonetic similarity to detect corresponding (matching) regions in the texts and to properly align them on multiple levels of segmentation. For this purpose, semantic and phonetic similarity measures were developed for the matching procedure. The resulting alignment is interpreted with a newly developed rule engine which allows precise control over the reconstruction process with context-sensitive reconstruction rules.

The experimental evaluation showed that the text quality improved for the reconstructed text in comparison to both recognised and written text. For recognised texts with a low word error rate, the best reconstruction system improved the F1-score of the best baseline system from 92.9 to 94.1%. In general, phonetics-based rules proved to be more effective than semantic-based rules while semantics-based rules turned out to be more precise. Combining phonetic and semantic knowledge for text reconstruction improved the reconstruction quality. A more detailed analysis revealed that 8% of the resolved mismatches are of pure phonetic nature, only 0.5% of pure semantic nature, and about 17-18% are detectable with both semantic and phonetic measures together. The rule engine for reconstruction proved to have comparable performance to a data-driven classification system for the low word error rate condition, while for medium and high word error rates, the automatic classifiers returned better results.

Concerning the overall benefit for the speech recognition system, an experiment with a retrained language model based on reconstructed texts yielded a word error rate reduction of 7.74% relative in comparison to a standard retraining, and of 4.38% relative for a language model based on randomly reconstructed text. As no specific optimisation has been performed yet, further improvements by parameter tuning are still possible.

The focus of the SPARC approach on the assignment of phonetic and semantic similarity between aligned speech recognition results and final reports has turned out to be useful and suitable for the reconstruction of literal transcripts. Three main aspects of the approach have already turned out to be beneficial: 1) Reconstructed texts reduce the required amount of manually transcribed texts for training of speech recognition systems. 2) Retraining with reconstructions leads to slightly lower word-error rates in speech recognition. 3) Since the reconstruction and alignment are knowledge based, our methods may also be used as annotation tools for semantic and phonetic information; these methods may serve as a starting point for automatic document creation.

In future work, we plan to evaluate the usage of reconstructed text for re-training or re-scoring of the acoustic models. We expect the gains for this task, however, to be minor, since the amount of material that the SPARC method makes available for acoustic training in addition to the material that is already there is relatively small. Furthermore, we want to compare our results to different approaches of including semantic information into the language model, e.g., as classes or embedded grammars.

Chapter 5

Speaker-Specific Selective Pronunciation Modelling

Within the data processing chain of large vocabulary continuous speech recognition (LVCSR), the pronunciation model (PM) is the important link between the acoustic model and the language model. Although this mapping from a sequence of phones to words significantly contributes to the performance of the ASR system, often only little effort is invested in optimisation of the PM. Both, implementation in the form of a lexicon and design with the help of phonologic rules have not changed considerably over the years [52], [98], [54]. A PM should be capable of modelling variation in pronunciation as it occurs within a larger group of speakers and of resolving phonologic ambiguities such as homophony or oronymy. It is right at this processing stage that automatically extracted phonetic knowledge is best integrated into the ASR process. This way, confidence in the decision process is gathered early and the disambiguation of similar-sounding words is not just postponed to a later processing stage (e.g., the language model or semantic rescoring of the n-best recognition results).

For systems with a very large number of users, a universally optimal PM is difficult to find. Particularly non-native speakers or fast talkers exhibit deviations in speaking style that are often either too strong for complying with the existing models or simply too rare to be generalised at all. In such situations, a speaker- or speaker group-specific adaptation of the LVCSR system is a promising solution for dealing with the variety. Apart from acoustic model adaptation, also pronunciation model and – whenever the degree of language proficiency is very low – language model adaptation is possible. Model adaptation, however, always presumes that a well-conditioned model already exists, and only parameter transformations have to better match the underlying model with the observed data. Ideally, this transformation works with only small amounts of adaptation data in an unsupervised fashion for full automation. For ASR service providers that have data of many users at their disposal, but only buy in black-box ASR technology, such automatic adaptation techniques operating in a non-invasive fashion are an interesting option for improving their quality of service.

Many methods have been proposed for modelling pronunciation variation at the lexical level (cf. [115], [128], [30] for an overview). These can basically be divided into data-driven and knowledge-based methods. Recently, more attention has been drawn to the data-driven approaches due to the availability of large annotated speech corpora. In contrast to the knowledge-based approaches, data-driven pronunciation modelling is more amenable to speaker-specific styles and can be semi-automated requiring only little supervision. These advantages are counterbalanced by higher sensitivity to modelling data noise and the dependency on large amounts of collected speech data.

Usually, these pronunciation model adaptation methods propose the extension of an initial general ASR lexicon with new pronunciation variants. Inserting new pronunciation variants, however, implies two opposing effects: improvement (gain) in pronunciation model accuracy and losses due to higher confusability among similar words within the lexicon. For this reason, a new approach is presented here that is exactly built on optimising this conflict of objectives. It is an extension to the works in [71] and [31], adapted to the task of pronunciation modelling at the lexical level with the help of speaker-specific phone confusions extracted from aligned recognised texts and final reports. The optimisation model incorporates the derived joint probability distribution of phoneme substitutions in a weighted finite state transducer (WFST) to allow for efficient computation.

In the beginning of this chapter, the pronunciation modelling and adaptation problem is defined, reasons for variation in pronunciation are discussed and an overview over related work is given. These introductory reflections are the basis for the definition of a new PM adaptation approach that is presented in section 5.2. In a series of experiments, the new approach is extensively evaluated and the results are discussed, such that directions can be given for further work on this idea.

5.1 Pronunciation modelling at the lexical level

5.1.1 General definition

A pronunciation model is usually realised in the form of a lexicon or dictionary. Each lexeme represents an orthographic word that is related to at least one phone symbol sequence – the word's pronunciation. The first pronunciation per word is referred to as the *canonical* pronunciation. Additional pronunciation alternatives are included to allow for a certain degree of variation. The final PM is the result of the process of finding pronunciation alternatives. These can be either directly collected from a large corpus of phonetically transcribed utterances based on their probability of occurrence or predicted with a rule-based model.

The pronunciation prediction task can be formalised as follows: Given a canonical source pronunciation $s^N(w) = \langle s_1, ..., s_N \rangle$ of a word w in an utterance, a prediction function $f(s^N, w) \rightarrow \hat{t}^M$ defines the transformations that transform s^N into the observed target pronunciation $\hat{t}^M = \langle \hat{t}_1, ..., \hat{t}_M \rangle$ obtained by manual phonetic transcription of the utterance. In general, $M \neq N$ and the symbol inventory is the same for both, source and target pronunciation. Canonical source pronunciations are derived from a phonetic lexicon, based on word w taken from the orthographic transcription of the utterance.

5.1.2 Motivation

The one-to-many mapping of the pronunciation model is primarily helpful for dealing with linguistic ambiguities that cannot be resolved otherwise. Pronunciation modelling can, however, also be applied to account for variations in speaking style over a larger group of speakers (e.g., non-native or accented speech, speaking rate variations). In some cases, it can also be "abused" for modelling syntactical constraints without a separate post-processing stage (e.g., the expansion of acronyms).

Linguistic ambiguities

The term linguistic effects relates to those words which cannot be disambiguated just based on their orthography. *Heterophones* are words that have identical orthographic spelling, but different pronunciation. The grapheme `read`, for instance, is used to denote the present tense (`/r i: d/`) and past tense (`/r e d/`) form of the same verb. *Heteronyms* as a special case of heterophones do not only differ in pronunciation, but also in their meaning. Consider the word `does` which can be either the 3rd person singular of the verb do (`/d V z/`), or the noun describing female animals (`/d oU z/`).

Non-native speech

Non-native speech is often characterised by serious deviations from native speech resulting in reduced acoustic model performance [129]. In contrast to regional varieties of a language, non-native speech is restricted to a smaller group of speakers which is usually inhomogeneous. Individual biographies have a large influence on the degree of deviation from the standard language. Factors such as birthplace, age, and social integration are directly related to a speaker's proficiency of a foreign language. This premise makes a systematic investigation difficult. For this reason, non-native accents are best studied with second language learners. These are already assessed according to their proficiency and their output is streamlined to a certain degree due to the teacher's instruction.

Non-native speech can be studied with respect to frequency domain, time domain, and at the phonetic symbol level [129]. Frequency-domain features include deviations in the formant structure, particularly with F2 and F3, even if the speech is well-articulated and understandable [5]. Time domain features are, e.g., differences in voice onset time and word final stop release time [4]. Word rate and the ratio between speech and silence are significantly different as well. Non-native speakers tend to speak slower and with longer pauses which leads to a reduction of coarticulation effects [118]. Differences on phonetic level are most easily explained with the differences between the phoneme set of mother-tongue and foreign language for a speaker [28]. During the acquisition of the foreign language, the speaker learns to transform his source phoneme inventory to the new target language phoneme inventory and possibly to extend it with new phonemes that are foreign to him. The latter is often difficult and results in systematically wrong pronunciations.

5.1.3 State-of-the-art in lexical pronunciation modelling

Modifications of the lexicon can be divided into rule-based, data-driven and combined approaches. Rule-based approaches try to model non-native speech with the help of explicit pronunciation rules for non-native speech. New pronunciation variants are derived by applying these rules to the canonical pronunciation (e.g., [57], [107]). Adapting a lexicon this way is much faster compared to handcrafting, although at the expense of accuracy, as rules usually only cover phenomena in a broad sense. Data-driven approaches extract the information directly from a training corpus of phonetically transcribed non-native speech by means of a phoneme recogniser (cf. [3], [101]). This leads to higher precision for specific applications, but requires large amounts of adaptation data for finding robust estimates. At the same time, generalisation to new words is difficult. Combined methods avoid this problem by learning the pronunciation rules from a training corpus [29]. As a reference for a combined pronunciation modelling technique, the classification and regression tree method will be explained in more detail.

Classification and regression trees (CART)

Classification and Regression Trees (CART) were introduced by Breiman [12] as a solution for automatic classification of samples based on distinctive features. CART are a nonparametric and nonlinear method that is best used whenever only little a priori knowledge about the underlying structure is available like, e.g., in data mining. In such cases, CART provide results which are easy to interpret and to implement in optimisation tasks. CART have already been successfully used by Fosler [29] and others (e.g., [2], [101]) for modelling pronunciation variation in spontaneous speech.

A CART tree is a compact description of how a data set may be partitioned into disjoint subsets along a binary tree, such that each leaf of the tree represents a data subset and the tree nodes define splitting criteria. The tree growing algorithm maximises the "purity" of each immediate subset with respect to the current node's splitting criterion. Therefore, the method requires a set of binary questions as possible splitting criteria, a goodness of split criterion that selects the best criterion for a certain split, a rule that defines when to stop splitting, and an assignment of leaf nodes to classes [12]. For the case of PM, the set of questions comprises contextual information combined with phonetic (cf. appendix A, table A.2) and linguistic a priori knowledge (cf. appendix A, table A.3).

Pronunciation variant generation

In the training phase, the input phone sequences are first aligned and then for each source phone symbol all instances are extracted together with their immediate right and left context phone. From these list of realisations a separate tree is learned for each phone which expresses the transformation in terms of the contextual information and additional phonetic and linguistic class information. An example tree for the phone `/d/` realised by three different speakers is depicted in figure 5.1. The binary questions are the internal nodes of the tree. By traversing the tree the transformation rules are read off and the final leaf symbol defines the actual output symbol of the transformation.

For creating a new pronunciation variant the training procedure is reversed. For each phone of the canonical pronunciation a target phone is predicted by traversing the specified CART tree. The resulting phone sequence is then only concatenated to retrieve the new pronunciation variant.

Modelling deletions and insertions

If source and target pronunciation are of different length (i.e., $M \neq N$), the modelling effort increases. Deletion of phones from the source string is modelled easily by indicating the deletion with a special (empty) symbol in the target pronunciation. Modelling insertions is a much bigger problem. In contrast to deletions, an insertion of a phone in the target pronunciation would require the generation out of nothing at an arbitrary position within the symbol string. There are three options how to handle this problem in a sequential generation approach:

A. By allowing a generative function that produces output of more than one target symbol per source symbol. This is the most accurate and intuitive strategy, but it requires non-trivial modifications of the learning algorithm.

`source:`	`s`	`{`	`l`	`@`	`d`
`target:`	`s`	`{`	`l`	`@`	`d @`

B. By introducing new symbols into the symbol inventory which represent inserted phones as double-phones. This model is easy to implement, but produces an exponentially growing symbol inventory which is difficult to handle.

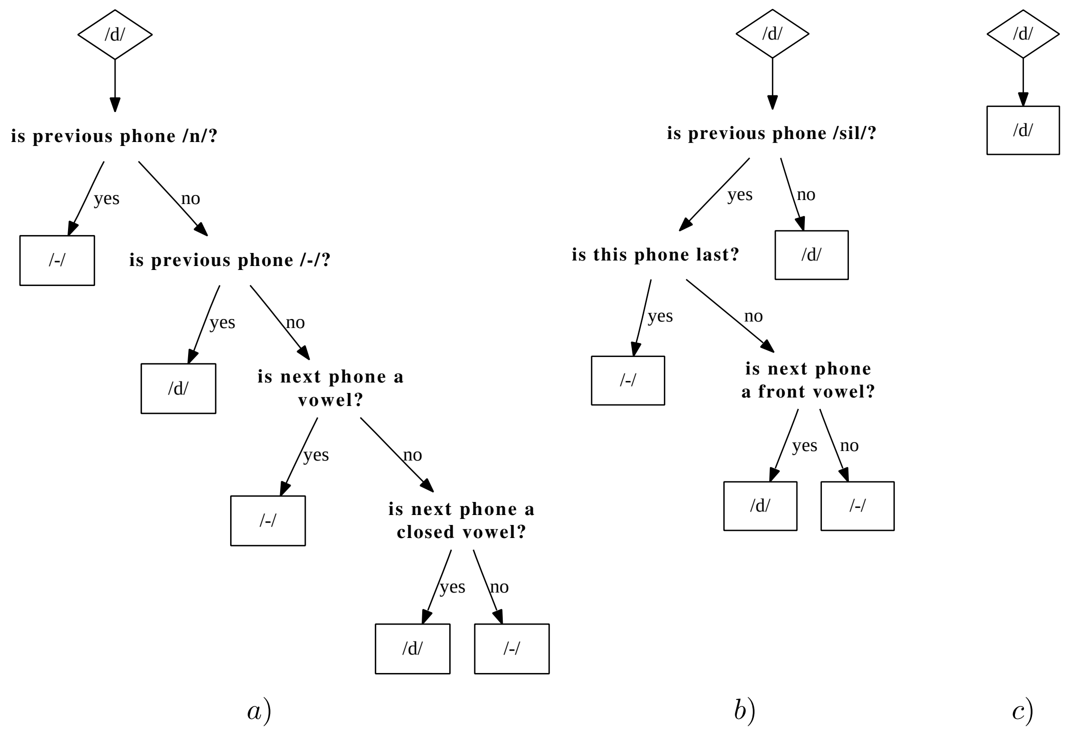


FIGURE 5.1. CART trees of phone `/d/` for speakers a) #050678 b) #047722 c) #008107 from data set MEDALIGN-NNS102. Inner nodes represent splitting criteria with regard to current, previous, and subsequent phone.

source:	s	{	l	@	d
target:	s	{	l	@	d+@

C. By inserting seed symbols into the source symbol string at each possible position where an insertion may occur as shown below. If the seed symbol is chosen to be the empty symbol which has been used for modelling deletions, this approach is intuitive and easy to implement. However, it is only possible to model single insertions.

source:	s	_	{	_	l	_	@	_	d	_
target:	s	_	{	_	l	_	@	_	d	@

In the experiments, insertions were modelled with model C. To avoid the loss of contextual information due to the additional seed symbols, the phones in context of the current source phone were selected such that they do not represent the immediate seed symbols but the actual context phone instead, unless the seed symbol covers a real insertion.

5.1.4 Alternative non-native speech adaptation approaches

Several methods have been proposed to account for non-native speech in automatic speech recognition. They can be divided into methods for acoustic model adaptation or specific pronunciation modelling. In terms of pronunciation modelling, modifications to the lexicon have been proposed as well as modified decoding procedures. The latter avoid the trade-off between high coverage and accuracy in the lexicon due to the increased number of pronunciation variants including increased confusability.

Acoustic model adaptation

Due to the inhomogeneity of the group of non-native speakers, methods for speaker-specific acoustic model adaptation appear promising. These methods include Maximum-A-Posteriori (MAP) adaptation [65] or Maximum Likelihood Linear Regression (MLLR) [66]. Some of these methods, however, require large amounts of adaptation data for modification of the acoustic models for each speaker. Another approach is to model the second language acquisition process by transforming acoustic models from the non-native speaker's source language to the target language. Actual merging of source and target acoustic models has proven to be the most effective method [129]. In contrast to MLLR adaptation, less data is needed to provide the same improvement on the word recognition rate. The combination of MLLR adaptation and lexical pronunciation modelling has proven to be beneficial as well [36].

Alternative pronunciation modelling techniques

Pronunciation modelling is not restricted to lexicon modifications. In [111], priors for pronunciation variants are estimated during training with forced alignment in maximum likelihood or discriminative fashion [110]. These (unigram) priors are then explicitly integrated into the decoding procedure, thus reducing the impact of less likely pronunciation variants. Similarly, pronunciation modelling may also be incorporated at the parameter-tying level of the acoustic model as proposed by Saraçlar et al. [106] or Hain [42]. In these methods, a 'soft' parameter tying scheme replaces the deterministic mapping between phoneme and HMM or HMM state. Instead of explicitly enforcing phone substitutions at the symbol or HMM level, HMM state sharing between the substitutable phones are allowed during acoustic model training which implicitly models variation in pronunciation. A more radical approach is taken by Ristad and Yianilos [104]. Instead of guessing appropriate underlying pronunciation forms during lexicon design they propose to model the observed surface pronunciations directly with a stochastic transducer. In contrast to the standard generative pronunciation model their surficial pronunciation model encodes the variability across pronunciations and not the phonologic processes to derive a surface form from an underlying canonical form.

5.2 A model for minimisation of lexical confusability

Many lexical pronunciation modelling methods propose the extension of an existing LVCSR lexicon with new pronunciation variants. Pronunciation modelling, however, also means to find the optimal balance between two opposing effects, namely the achieved gain in modelling accuracy and the losses due to higher confusability within the lexicon. For increasing the goodness of fit of a specific word, it is preferable to have many possible pronunciations assigned to it. These additional pronunciations are, however, not only similar to the word that they represent, but also to other words in the lexicon. The higher the number of alternatives, the higher the confusability. Reducing the number of pronunciation alternatives on the other hand, also decreases the accuracy.

Consider a very simple lexicon consisting of the three words ACCESS, AXIS, and EXCESS:

Word	**Canonical pronunciation**	**Alternative pronunciations**
ACCESS	/{ k s e s/	*/e k s e s/*
AXIS	/{ k s @ s/	
EXCESS	*/e k s e s/*	/I k s e s/

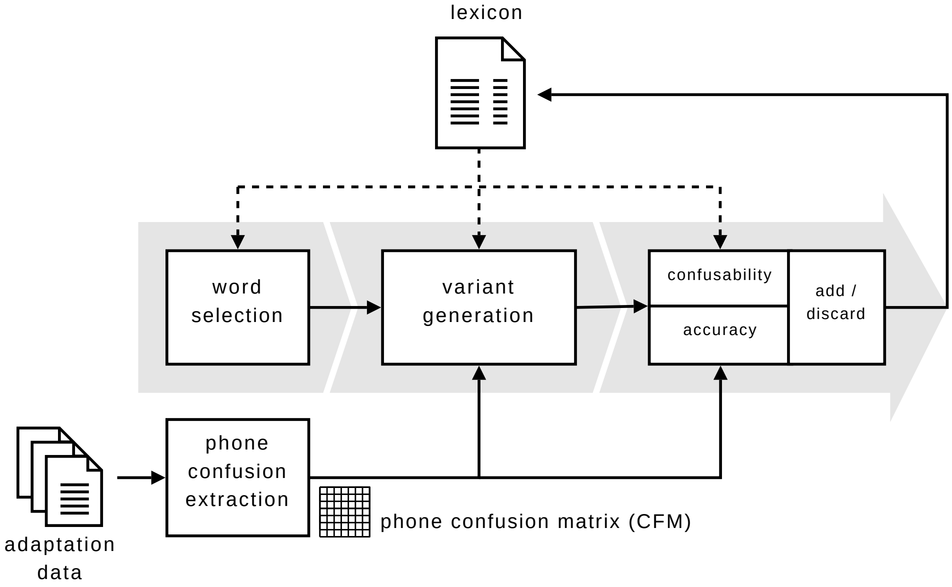


FIGURE 5.2. Schematic illustration of the lexical confusability optimisation approach.

The canonical pronunciations already show a high degree of similarity among each other. If variants are added arbitrarily, highly confusable pairings within the lexicon may appear easily as indicated between the words `ACCESS` and `EXCESS`. Ideally, the gain in model accuracy by adding the variant `/e k s e s/` should be opposed to the increase in confusability that goes with it. As a result this variant would most probably be discarded.

Various criteria have been proposed for optimal pronunciation variant selection. Most simply, pronunciation variants may be included based on their frequency of occurrence in a training corpus [109]. In [48], a maximum likelihood criterion is proposed for selecting an optimal subset of pronunciation baseforms. Similarly, a confidence measure can be used for rejecting statistically irrelevant variants during lexicon generation [114].

Based on this observation, a pronunciation modelling approach can also be defined as an optimisation problem for finding the right balance between these effects. If there are measures for confusability and the gain in accuracy, a method can be defined that modifies an existing pronunciation lexicon with regard to these criteria. Figure 5.2 illustrates this approach schematically. Starting with a pronunciation lexicon, a candidate word is selected from it for optimisation. From this word, a new, speaker-specific pronunciation variant is generated. The new pronunciation is then tested for both, confusability with other words and similarity to already existing variants of the same word. A classification stage decides based on the measurements whether the variant should be integrated or not. From the updated lexicon, the next word is selected for optimisation. Speaker-specific phonetic confusion information is extracted offline from a pool of data and provided as an input to the variant generation routine and the measurement procedures.

This scheme presents two essential challenges: First, the development of appropriate measures for accuracy and confusability. While a measure for accuracy has to express the gain achieved by the model between a speaker-independent and its speaker-specific variant for each word, the confusability measure has to relate a new pronunciation variant for a word to the rest of the lexicon. Second, it is vital to find an efficient implementation that allows for fast processing of large lexicons. Consider an LVCSR lexicon with 50,000 entries. Determining the similarity of a word to all other words in the lexicon requires $W \cdot W$ measurements with W being the number of words in the lexicon. For each word pair the similarity calculation again requires $O(N \cdot M)$ tests with N, M being the number of phones per word. Assuming an average word length of 4 phones, a total of $4 \cdot 10^{10}$ tests must be performed to complete one iteration of optimisation. Even for a single iteration, it turns out that this approach is computationally too expensive. Besides an efficient representation, the integration of approximations and heuristics becomes necessary for speeding up the computation.

In general, this greedy algorithm will not find a global optimum for the optimisation problem. Nevertheless, the algorithm could be iterated as well or combined with a dedicated mathematical global optimisation method such as, e.g., Simulated Annealing [58]. Despite the efficient computational representation and various approximations, an iterative global optimisation is at the moment not computable in reasonable time for large vocabulary pronunciation lexica.

5.2.1 Learning speaker-specific phone confusions from data

For the optimisation framework, we assume that speaker-specific information is represented in form of a phone confusion matrix (CFM) as it is produced by the methods reviewed in chapter 3. The calculation of a speaker-specific CFM consists of two steps: In the first one, two independent phonetic transcriptions of the same utterance representing *reference* and *hypothesised* transcriptions are aligned and segmented on word level. In the second step, the CFM is trained from these phone sequence pairs.

Alignment

Depending on the input data sources, two competing approaches were defined for the experiments. The first one is intended for classical ASR application scenarios with speaker enrolment, while the second represents a rather automatic training in absence of manual orthographic transcriptions of the adaptation data.

- The **ENROL** alignment requires audio recordings and corresponding manual orthographic transcriptions of the adaptation data. From these basic data sources two different phone sequences were generated (cf. figure 5.3). The *reference automatic phonetic transcription (APT)* is the result of a forced alignment of the manual orthographic transcription with the audio data using speaker-adapted triphone acoustic models. In these transcriptions, recognition errors on word level can be ruled out at the expense of reduced accuracy on phoneme level. For a phonetic transcription closer to the actual pronunciation a *free automatic phonetic transcription* was produced by using speaker-adapted monophone acoustic models and unigram phone language model. This transcription now allows for more variation as it is more local and obtained with less contextual constraints, but at the same time it contains more inserted or substituted phones. This "transcription noise" should hopefully cancel out during CFM training. Finally, the two transcriptions were acoustically realigned at the frame level with the adapted acoustic models to ensure synchrony for the whole recording.

  The phoneme sequences were then paired based on time stamp information, as both transcriptions are time synchronous. The alignment was done on word level, where overlapping phones on word boundaries were assigned to the word that contained the larger fraction of the phone. The result of the alignment process is thus a sequence

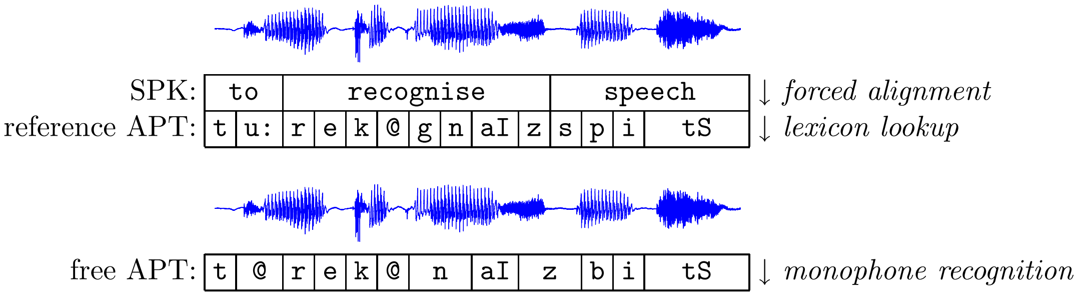


FIGURE 5.3. Reference automatic phonetic transcription generated from reference orthographic transcriptions by forced alignment and lexicon lookup vs. free automatic phonetic transcription directly generated from monophone acoustic model recognition.

of word pairs in phonetic transcription, where the canonical reference transcription from the ASR lexicon is paired with a monophone recognition result, reflecting a closer transcription of the actually observed pronunciation.

- In the **AUTO** alignment, the speaker-specific information is extracted in absence of manual orthographic transcriptions from automatically recognised transcripts of medical dictations and human post-processed final medical reports as utilised by the SPARC method (cf. chapter 4). The *reference* transcription phone sequence is an automatic phonetic transcription of the final medical report (generated during reconstruction, cf. section 4.4.2). The hypothesised transcription phone sequence is the most likely phone sequence from the speech recogniser. After alignment on word level, only those word pairs showing high phonetic similarity were selected as reliable matches that indicate potentially corrected recognition errors. In sum, this data set enables learning from available non-literal transcripts in a fully automatic way.

Confusion matrix training

A memoryless, context-independent (MCI) stochastic edit distance model was trained with the paired phoneme sequences for each speaker (cf. section 3.3.2). Training was done in three iterations with a minimum probability of 10^{-4} for each edit operation and a uniform initial parameter distribution. The learned parameter distributions represent a speaker-specific CFM. Figure 5.4 shows the CFMs for three speakers of the evaluation set as calculated with the ENROL alignment. Note the evident differences in substitutions (off-diagonal elements), deletions (leftmost column), and insertions (bottommost row).

Besides the speaker-specific CFMs, a speaker-independent CFM was trained as well from the MEDTRANS corpus (cf. section 2.5.1). This speaker-independent reference can be used for studying and evaluation of speaker-dependent CFMs. The training data used for the speaker-independent CFM differs from the speaker-dependent CFM in two important aspects. First, the MEDTRANS corpus contains actual manual phonetic transcriptions in contrast to the free automatic phonetic transcriptions of the MEDALIGN-NNS102 data set. Therefore, it can be seen as true reference data. And second, the manual transcriptions are derived from the canonical reference transcriptions such that there is no overlap of phones at word boundaries.

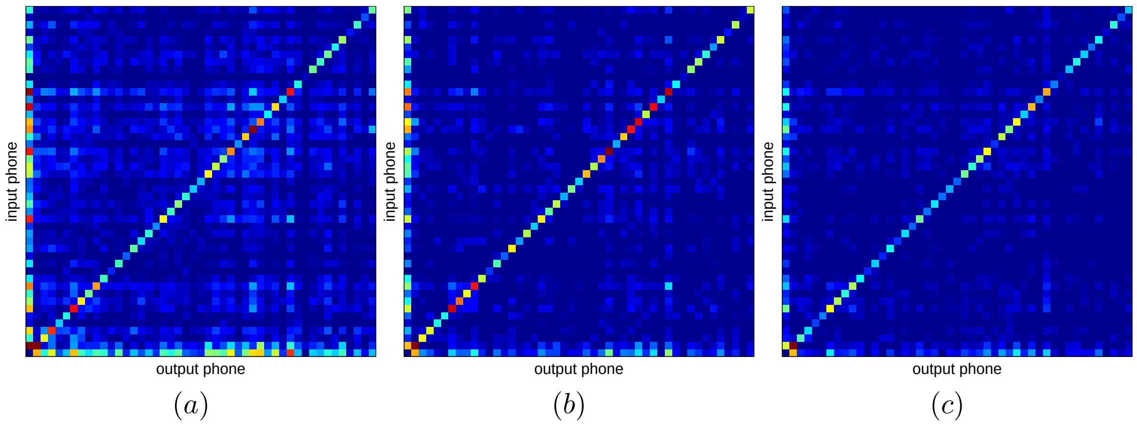


FIGURE 5.4. Phone confusion matrices for speakers a) #050678 b) #047722 c) #008107 from data set MEDALIGN-NNS102 with ENROL alignment.

5.2.2 Selection of candidate words

The optimisation procedure is accelerated if only a subset of lexicon entries is inspected. These candidate words should be representative for the lexicon and show potential for improvement. Words with high recognition error rates appear to be suitable candidates. For compiling a list of recognition errors again the SPARC reconstruction method presented in chapter 4 is applied. Requiring only recognised and corrected transcripts, the whole candidate selection procedure is fully automated and, therefore, applicable in a real-word adaptation scenario.

Like for the calculation of the confusion matrices, again an alignment of the two texts on word level is necessary, but only in the orthographic domain. The alignment process consists of the following steps:

1. **Selection of data**: The coverage of the candidate word list increases with the amount of analysed data. Since for many speakers only little material is available, the analysis was performed in a speaker-independent fashion.

2. **Definition of reconstruction rules**: Two reconstruction rules were defined for the framework. The first rule filters out identical words in the alignment to reduce the amount of data for postprocessing and counting. The second rule labels those parts in the alignment which show high phonetic similarity ($d_{th} \geq 8.0$). These word pairs are potential recognition errors that have been corrected during the transcription process. In addition to the labelling, the most likely spoken variant was returned instead of the actually written word.

3. **Realignment and reconstruction**: The paired texts were realigned and reconstructed according to this setup. Whenever the reconstruction rules returned more than one word for a certain pairing – which is possible due to spoken variant generation – the reconstruction result was split into single words. This means that the obtained error frequencies are upper bounds for the actual error frequencies.

4. **Ranking of the most frequent errors**: The word pairs of recognised and reconstructed texts were scanned for the error labels, sorted, and ranked according to their frequency of occurrence.

5. **Matching words with the ASR lexicon**: To ensure correspondence with the ASR lexicon for confusability analysis, the list of frequently confused words was matched with the lexicon in terms of hyphenation and case.

In the experimental evaluation, this set of recognition error candidate words is compared to the set of words with multiple pronunciation variants and a set containing all lexicon words.

5.2.3 Pronunciation variants from Weighted Finite State Transducers

Weighted finite state transducers (WFSTs) were proposed and already successfully demonstrated for ASR applications [80]. The processing chain from feature extraction down to language modelling can fully be expressed within this framework and provides a transparent, factored view on the complex task of ASR. The methods for pronunciation variant generation are an extension to the work of [31], where a similar task – modelling ASR errors by phonetic confusions – is described.

Definition

A WFST T is a tuple $T = (\mathcal{X}, \mathcal{Y}, Q, E, i, F, \lambda, \rho)$ over the semiring $\mathbb{K}$, where $\mathcal{X}$ is the input alphabet, $\mathcal{Y}$ is the output alphabet, Q is a finite set of states, $E \subseteq Q \times Q \times (\mathcal{X} \cup \varepsilon) \times (\mathcal{Y} \cup \varepsilon) \times \mathbb{K})$ is a finite set of transitions, $i \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, λ is an initial weight, and ρ is a final weight function. A state transition $t = (p[t] \to n[t], l_i[t] : l_o[t], w[t])$ is then an edge from source state $p[t]$ to destination state $n[t]$, with input label $l_i[t]$, output label $l_o[t]$, and weight $w[t] \in \mathbb{K}$. ε denotes the empty symbol. More details can be found in [80].

A path in T is a consecutive sequence of transitions $t = \langle t_1 ... t_n \rangle$. It becomes a successful path Π if it starts from the start state i and ends in a final state $f \in F$. The output label of Π is the concatenation of the labels along its transitions: $l[\Pi] = \langle l[t_1] ... l[t_n] \rangle$. For pronunciation modelling, $\mathbb{K}$ will be a *tropical semiring* $(\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$ over the positive real numbers together with min and addition (+) as algebraic operations, negative log likelihoods as weights, and a Viterbi (best path) approximation. In this case, the path's weight is the sum of the initial weight, the transition weights, and the final weight: $w[\pi] = \lambda + w[t_1] + ... + w[t_n] + \rho(n[t_n])$ [80].

The algorithmic operations on WFSTs that will be used in the framework are: *union*, *minimisation*, *composition*, *inversion*, and *decoding*. The *Union* operation combines two WFSTs in parallel. *Minimisation* removes redundant information by determinisation (ε-removal) and combination of states. *Composition* means the relational composition of two WFSTs $T = R \circ S$, such that there is exactly one mapping sequence u to sequence w for each pair of paths that is created by $R : u \to v$ and $S : v \to w$. *Decoding* finally is the method that returns the n-best successful paths sorted according to their path weights [80].

Lexicon and CFM representation

A pronunciation lexicon $P : w_i \to \{p_{i,1} ... p_{i,n_i}\}$ is a mapping from orthographical words w_i to pronunciations $p_{i,j}$, where n_i is the number of pronunciations per word and W is the total number of words. Each $p_{i,j}$ is represented by a finite state transducer as a transformation of a graphemic input symbol w_i to a sequence of phonetic output symbols (cf. figure 5.5). The

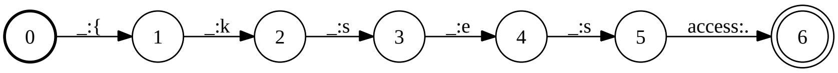


FIGURE 5.5. FST representation of the canonical pronunciation /{ k s e s/ of the word access.

empty symbol is denoted by "_" in the graphemic domain and "." in the phonetic domain respectively.

P is created by forming the *union* of the individual pronunciation FSTs. As a consequence, the resulting directed graph is huge, containing many epsilon transitions and redundant information. For larger lexicons, it turns out that many pronunciations share common prefixes. The words access → /{ k s e s/ and axis → /{ k s @ s/ for example share the first three phones and altogether only differ in a single phone. Since the graphemic input symbol that assigns the phone sequence to a word is only added at the very end of the path, common prefixes can be combined into a single path, significantly reducing the size of the lexicon graph (cf. figure 5.6).

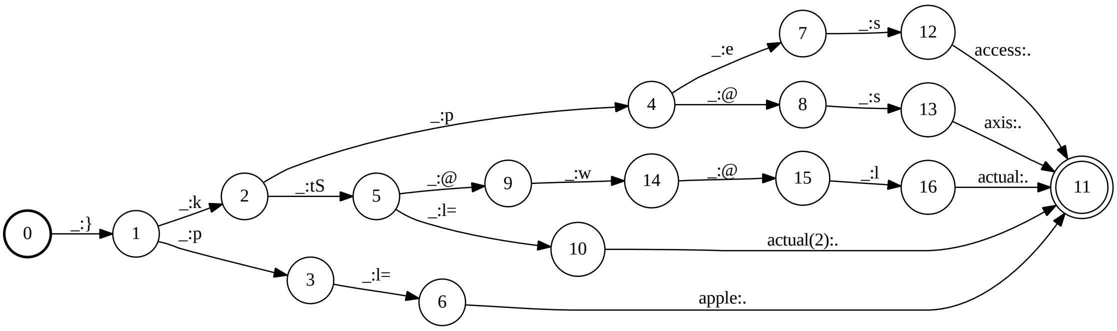


FIGURE 5.6. FST lexicon P comprising the words access, axis, actual, and apple.

A confusion matrix $C = \{c_{xy}\}$ between input phones $x = 1..|\mathcal{X}|$ and output phones $y = 1..|\mathcal{Y}|$ is also expressed as a WFST with identical input and output alphabets. Every matrix element c_{xy} is a transduction from symbol $x \in \mathcal{X}$ to $y \in \mathcal{Y}$ with weight c_{xy}. The WFST of C is thus a single state with self-transitions for each of its entries (cf. figure 5.7).

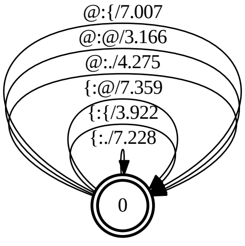


FIGURE 5.7. WFST confusion matrix C for substitutions and deletions of phones /@/ and /{/ with corresponding weights.

Pronunciation variant generation

From the speaker-specific confusion matrix C and the speaker-independent lexicon P it is possible to generate new pronunciation variants $\hat{p}_i$ for a specific word w_i by applying the composition operation:

$$\hat{p}_i = \text{argmax}\{w_i \circ P^{-1} \circ C\}. \tag{5.1}$$

The first composition selects the canonical pronunciation for word w_i from the pronunciation lexicon. The second composition replaces the phone transitions with those edges from confusion matrix C that have the corresponding input symbol. To reduce the number of transitions, the graph is pruned such that only transitions falling below a threshold d_{cfm} are included. The resulting graph depicted in figure 5.8 contains all paths that result in possible pronunciation variants. The decoding operation finally returns the list of best pronunciations from the graph.

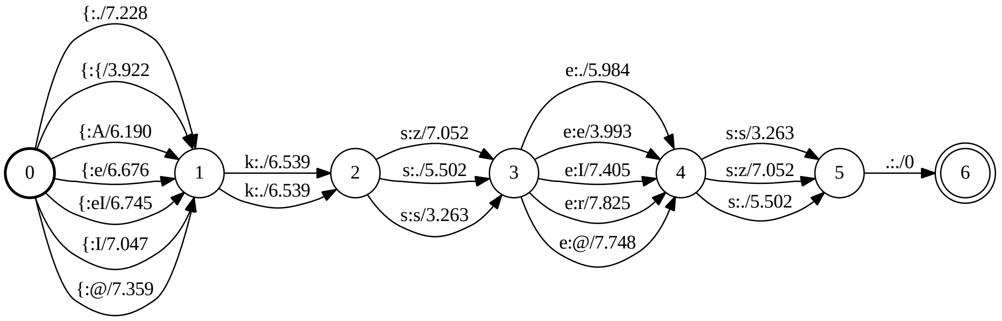


FIGURE 5.8. WFST composition of the word `access` with confusion matrix C.

Since phone identities have minimal negative log likelihoods, the best path – by definition – is the canonical pronunciation and, therefore, discarded. The remaining variants are then filtered such that they fulfil two constraints: First, variants with more than one deletion are discarded to avoid artifacts, and second, phone repetitions are merged into single phones.

5.2.4 Measuring confusability and the gain in modelling accuracy

Confusability measure

The measure for lexical confusability is an extension to the pronunciation variant generation procedure. By applying an additional composition operation to the WFST of figure 5.8, the phone confusions are transformed back to words that are already in the lexicon:

$$\hat{w}_i = \text{argmax}\{w_i \circ P^{-1} \circ C \circ P\}. \tag{5.2}$$

The operation simply feeds the output symbol sequences of the pronunciation generator as input sequences to the lexicon transducer which then returns the corresponding words (cf. figure 5.9). Again the number of paths to investigate has to be reduced by pruning. This way, the most similar words of a lexicon given speaker-specific confusions are found very efficiently. To arrive at the confusability measure, the argmax operator needs to be relaxed such that $\hat{w}_i$ becomes a list of words that is sorted according to their similarity to the base word w_i using the MCI model of section 3.3.2, eqn. 3.11:

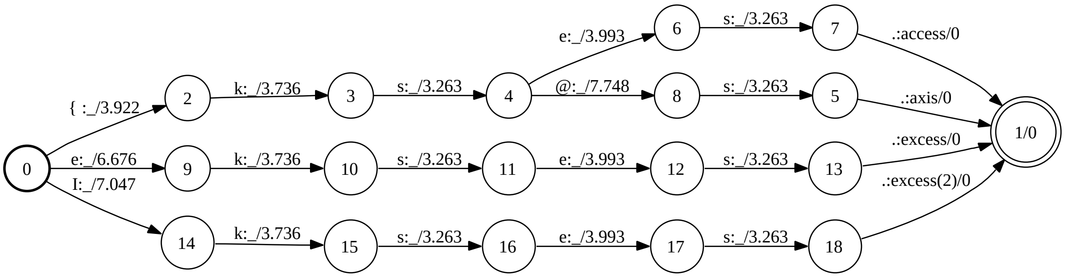


FIGURE 5.9. WFST composition of the word `access` with confusion matrix C and lexicon P.

TABLE 5.1. Ranked list of confusability measurements for base word `ACCESS` with pronunciation `/{ k s e s/`, confusion matrix C, and lexicon P (cf. figure 5.9).

Word	**Pronunciation**	**Score**	**Rank**
`ACCESS`	`/{ k s e s/`	18.177	1
`EXCESS`	`/e k s e s/`	20.931	2
`EXCESS(2)`	`/I k s e s/`	21.302	3
`AXIS`	`/{ k s @ s/`	21.932	4

$$\hat{w}_i = \left\{ \hat{w}_{i,1}, \hat{w}_{i,2}, ..., \hat{w}_{i,m} \right\}_{d_{MCI}(w_i,\hat{w}_{i,1}) \leq ... \leq d_{MCI}(w_i,\hat{w}_{i,m})}. \tag{5.3}$$

This list is in fact the m-best list of the decoding operation. For practical considerations, m will typically be much small than W (e.g., $m = 20$). From this ranked list of confusable words, the confusability measure d_{cfb} is derived as the rank of the base word w_i, where equally weighted words due to homophones are assigned the same rank:

$$d_{cfb}(w_i, \hat{w}_i) = \left\{ k \mid \hat{w}_{i,k} = w_i \right\}. \tag{5.4}$$

The measure is low, if there are only few words whose pronunciation variants are more similar to the pronunciation variants of the base word, i.e., which have a lower rank. If there are many highly confusable words, then the returned rank and hence the confusability measure will be higher. Table 5.1 illustrates confusability measurement for the example graph in figure 5.9. Starting from the base word `access` with pronunciation `/{ k s e s/`, the list of similar words is ranked according to the similarity score. In this example, $d_{cfb} = 1$, as the base word already heads the m-best list.

Accuracy gain measure

The measure for the gain in modelling accuracy is meant to counterbalance the effects of adding new pronunciation variants with low confusability. Keeping confusability low is of no meaning if the gain in accuracy achieved by the new pronunciation is negligible. For this reason, a measure that defines the gain in accuracy is needed.

For the definition of this measure the phonetic similarity measures presented in chapter 3 can be re-used. The memoryless, context-independent model of a stochastic edit distance between phone sequences was defined as (cf. eqn. 3.11):

$$d_{MCI_{sto}} = -\log\ p(x^N, y^M|\theta) \tag{5.5}$$

meaning the negative log likelihood of the joint probability of phone sequences x^N and y^M given the edit operation probabilities θ. So far, differences in the distance values have only been achieved by altering either of the two input sequences, e.g., by comparing two realisations $\tilde{y}^M$ and $\tilde{\tilde{y}}^M$ of a word to the canonical pronunciation x^N. By exchanging the underlying parameters θ, the similarity of the input sequences with respect to different parameterisations can be computed instead. Defining θ_{spk} as the speaker-specific CFM and θ_{ref} as the speaker-independent reference CFM determined at the end of section 5.2.1, a measure is defined that is directly related to the gain in modelling accuracy:

$$d_{acg}(p_{i,1}, \hat{p}_i) = \mid d_{MCI_{sto}}(p_{i,1}, \hat{p}_i \mid \theta_{ref}) - d_{MCI_{sto}}(p_{i,1}, \hat{p}_i \mid \theta_{spk}) \mid \tag{5.6}$$

where $p_{i,1}$ is the canonical pronunciation and $\hat{p}_i$ is the newly generated pronunciation to be tested. d_{acg} is low for pronunciations of approximately equal similarity with respect to speaker-dependent and reference parameters, and high if the similarities are different.

5.2.5 Optimisation strategies

The proposed optimisation approach assumes an existing initial pronunciation lexicon that already contains pronunciation variants (PVs) for a number of words. Therefore, it has to deal with existing variants as well as with possible new variants. There are several strategies for optimising an existing pronunciation lexicon to a specific speaker, i.e., to minimise the confusability while retaining high accuracy:

a. **Test existing variants, do not add new ones (WFSTa)**: If the lexicon already contains the right PVs for a given speaker, then only the redundant alternative PVs need to be removed to minimise the confusability. This approach is the most simple, as there is no need for the generation of new, speaker-specific PVs.

b. **Test existing variants and add new ones (WFSTb)**: The lexicon may in some cases already contain the most suitable PVs for a given speaker, but maybe not for every candidate word. This means that the existing variants have to be evaluated together with newly generated ones, and only the best ones will be included in the new lexicon.

c. **Keep existing variants and add new ones (WFSTc)**: The initial lexicon gives already good coverage with its variants, but in some cases, the speaker's pronunciation is too far away from the available pronunciations. By adding new variants, the shortcomings will be remedied.

d. **Discard existing variants and add new ones (WFSTd)**: The initial lexicon does not represent the speaker at all. For this reason, all existing variants have to be removed and a new lexicon is built from scratch. This step simulates the creation of an initial lexicon.

5.3 Experiments

The WFST-based PM optimisation approach was evaluated in an experiment by creating new, speaker-specific lexica for non-native and accented native speakers. For these speakers, the standard pronunciation model may not fit perfectly, thus, there is potential for improvement. The models were trained with a moderate amount of adaptation data and evaluations were performed on a held-out test set. As the ultimate goal is to improve the ASR recognition rate, the pronunciation models were directly tested with an operational ASR system.

Table 5.2. Native and non-native speaker accents annotated in the NNS102 data set.

Native accents	Speakers	Non-native accents	Speakers
North	1	African	1
Northeast	19	Asian	6
Midland	7	British	1
Other	6	Hispanic	13
South	14	Indian	16
West	8	Iranian	10

5.3.1 Experimental setup

The ASR experiments were performed with a Philips commercial speech-recognition system for professional medical dictation. Recognition was performed in batch dictation mode with adapted and unadapted acoustic models, and a specific context model comprising multiple medical domains. The initial lexicon contained 13,830 words with 16,917 pronunciations, covering the evaluation data completely. The performance was measured in terms of word error rate (WER) between recognised text and manual reference transcription where partial words, hesitations, and non-speech words in the reference transcription are ignored. For illustration purposes, the resulting number of newly generated pronunciations is included in the form of statistical boxplots as well.

All investigations are based on data from the MEDALIGN corpus (cf. section 2.5.2). The so-called inspection subset (INSPECT) covers data which are used for acoustic adaptation. These data must not be taken for PM training, but can be used for the selection of candidate words as a large number of reports is available for each speaker. In total, this set comprises 3,453 reports from 434 speakers.

The actual speaker-specific data is collected in the NNS102 data set. This set consists of 102 speakers which are marked as either non-native or accented. Speakers are assigned to one out of 12 accent groups (cf. table 5.2), where 6 are non-native speech and 6 are native American accents. For 102 speakers, more than 25min of recordings were available which were divided into a *Train/Dev set* of 15min and an *Eval set* of ≥10min of speech.

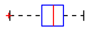
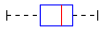
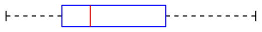
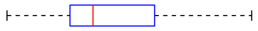
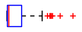
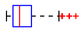
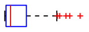
In the course of the evaluation a number of factors were identified which influence the system performance. These are the optimisation strategy for building the lexicon, the set of candidate words, the data source from which the speaker-specific confusions are extracted, and the usage of speaker-specific adapted models. The following presentation of the experimental results is structured according to these factors.

5.3.2 Comparison of optimisation strategies

The first series of experiments was aimed at finding the best optimisation strategy together with its optimal threshold settings for the confusability and accuracy gain measures. For this reason, candidate word set and phonetic confusion data was fixed to potentially optimal values, while the thresholds were varied between $d_{cfb} = \{1, 2, .., 6\}$ and $d_{acg} = \{0.5, 1.0, .., 3.0\}$ for confusability and accuracy gain, respectively. For all optimisation strategies presented in section 5.2.5 the optimal values were determined. For each strategy, results are reported with the best fixed threshold settings over all speakers (fix) and speaker-specific optimal thresholds (opt). The resulting figures are collected in table 5.3.

All systems including the baseline system exhibit a relatively high standard deviation in word error rate. This is a consequence of the high diversity among speakers of the NNS102

Table 5.3. WER in [%] and pronunciation variant (PV) counts for various optimisation strategies and evaluation setups: Fixed thresholds for all speakers (fix) and optimal thresholds per speaker (opt).

		WER	**Cand.**	**New PVs**	**Total number of PVs**
		[%]	Count	Count	14,000 15,000 16,000 17,000 18,000 19,000
base		28.45 ± 13.35			
WFSTa	fix	29.73 ± 13.39	2,409	0	
	opt	29.39 ± 13.03	2,409	0	
WFSTb	fix	29.78 ± 13.52	2,409	4,132	
	opt	29.23 ± 13.18	2,409	4,132	
WFSTc	fix	28.46 ± 13.36	2,409	1,071	
	opt	28.19 ± 13.27	2,409	1,071	
WFSTd	fix	33.19 ± 12.18	2,409	1,071	
	opt	32.67 ± 12.02	2,409	1,071	

data set. As shown later in table 5.7 non-native speakers show worse word error rates on average than the native accented ones. With respect to the high standard deviation the differences between the systems seem to be diminishing at first, but become more visible on a per speaker basis as will be shown in figure 5.10 later on.

Overall, the WFSTc approach of keeping existing pronunciation variants and adding new ones returned the best results. This finding is probably best explained by the high quality of the initial lexicon and the fact that the existing pronunciation variants had been considered in the training phase as well. This is why the related WFSTa and WFSTb strategies return comparable figures, while the WFSTd strategy of creating new alternatives from scratch is far behind. The boxplots illustrate the clear differences in lexicon size and also the variability that goes with each method.

Table 5.3 also makes clear that individual threshold optimisation per speaker is essential for achieving best results. The variation in optimal threshold settings between speakers is very high and it was not possible to find any correlation with factors such as speaker group or baseline word error rate. If only a fixed threshold setting over all speakers is determined then it turns out that only very low threshold settings (e.g., $d_{cfb} = 1.0$ and $d_{acg} = 0.5$ for WFSTc) are selected and the pronunciation modelling effect is thus minimal. As the whole lexicon optimisation is done offline, tuning the thresholds for each speaker is a doable procedure in a real-world system.

5.3.3 Comparison of candidate word sets

The set of candidate words is another main factor in the PM approach. In the previous experiments, the set of candidate words was set to those words which already had more than one pronunciation in the lexicon. In this experimental series, this choice (MULTIVAR) was compared to a setup where candidate words were determined as those words with frequent recognition errors (RECERR) and another setup, where all lexicon words were defined as candidates (FULL). Since the WFSTc strategy turned out to be the best, all further experiments were done with this strategy. The results for each candidate word setup are collected in table 5.4.

Optimising the full lexicon gives the best results, and there is a general trend that the word error rate goes down with higher number of candidates. The step between the largest

Table 5.4. WER in [%] and pronunciation variant (PV) counts for strategy WFSTc and different candidate word sets: Fixed thresholds for all speakers (fix) and optimal thresholds per speaker (opt).

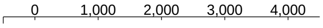
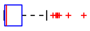
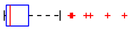
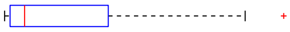
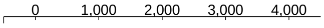
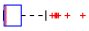
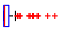
		WER	**Cand.**	**New PVs**	**Total number of PVs**
		[%]	Count	Count	
base		28.45 ± 13.35			
MULTIVAR	fix	28.46 ± 13.36	2,409	1,071	
	opt	28.19 ± 13.27	2,409	1,071	
RECERR	fix	28.50 ± 13.43	3,749	2,426	
	opt	28.14 ± 13.28	3,749	2,426	
FULL	fix	28.49 ± 13.40	13,830	8,069	
	opt	28.13 ± 13.29	13,830	8,069	

Table 5.5. WER in [%] and pronunciation variant (PV) counts for strategy WFSTc and different sources for speaker-specific confusions: Fixed thresholds for all speakers (fix) and optimal thresholds per speaker (opt).

		WER	**Cand.**	**New PVs**	**Total number of PVs**
		[%]	Count	Count	
base		28.45 ± 13.35			
ENROL	fix	28.46 ± 13.36	2,409	1,071	
	opt	28.19 ± 13.27	2,409	1,071	
AUTO	fix	28.47 ± 13.37	2,409	2,380	
	opt	28.35 ± 13.32	2,409	2,380	

candidate set (FULL) and the second-largest set (RECERR) is, however, diminishing. Considering the computation time as well, the RECERR set composed of frequent recognition errors should be the first choice for optimisation.

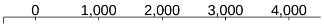
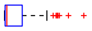
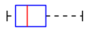
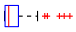
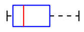
The ratio between proposed and actually included PVs does not scale linearly either. This ratio is not equal to one if there are PVs generated by the optimisation method that are not included into the lexicon due to high confusability or low modelling gains. At the same time, it is also an indicator for already existing PVs of the candidate words as the RECERR candidate set does not contain many words with existing PVs and does not overlap much with the multivar set.

5.3.4 Comparison of resources for speaker-specific confusions

It is a benefit of the proposed PM approach that the speaker-specific confusion information is completely separated from the rest of the framework and may therefore be obtained from any kind of source data with any kind of method. A preferable way of obtaining this information would be by automatic processing of already collected non-literal transcripts such that no manual intervention is needed to come up with a PM for a problematic speaker. This idea was tested as described in section 5.2.1 with data retrieved by the SPARC method. Table 5.5 presents the results compared to the baseline and the CFMs trained from literal transcripts (ENROL).

The AUTO CFMs with optimal threshold settings beat the baseline word error rates, but the gains are much reduced in comparison to the ENROL CFMs. At the same time, the

TABLE 5.6. WER in [%] and pronunciation variant (PV) counts for unadapted and adapted acoustic models with WFST and CART approach: Optimal thresholds per speaker (opt).

			WER	New PVs	Total number of PVs
			[%]	Count	
unadapted	base		28.45 ± 13.35		
acoustic	WFSTc	opt	28.19 ± 13.27	1,071	
models	CARTc	opt	28.23 ± 13.31	2,409	
adapted	base		17.02 ± 10.03		
acoustic	WFSTc	opt	16.89 ± 9.99	1,071	
models	CARTc	opt	16.90 ± 9.97	2,409	

number of proposed PVs is much higher, in fact almost equal to the number of candidate words, while the number of actually included PVs is much less. A closer inspection of the optimisation process revealed that most of the proposed PVs had been rejected due to high phonetic distance scores in the accuracy gain measure. In combination with the high number of proposed PVs, it must be concluded that the AUTO CFMs show a high mismatch to the CFMs trained with the ENROL data. Therefore, many generated variants are definitely new, but they do not resemble the speaker-independent distance model at all which means that the variant is rejected. Unfortunately, this means that for some speakers the AUTO CFMs provided only a very poor description of the actual speaker-specific confusions. These defects could be remedied by either more training data or a more elaborate confusion extraction from the non-literal transcripts.

5.3.5 Adapted versus unadapted acoustic models

In this final experiment the performance achieved with the WFST lexical PM optimisation system is compared to a benchmark CART system with respect to acoustic speaker adaptation. The experiments so far have been conducted with speaker-independent acoustic models to provide a realistic starting point for modelling without any speaker-specific adjustments. In practice, however, a pronunciation model will be applied in conjunction with an acoustic adaptation procedure. In the experiments acoustic adaptation was performed by applying a combination of MLLR and MAP adaptation prior to the actual pronunciation modelling step. Table 5.6 shows a comparison of the baseline word error rates with the best WFST results and results achieved with the CART method that has been optimised accordingly in terms of threshold settings. The CART results were created by first training a speaker-specific CART tree for each phone with the NNS102 training set, and then deriving pronunciation variants for the lexicon by applying the CART trees to each phone of the canonical pronunciation.

The figures in table 5.6 clearly show that acoustic model parameter adaptation has much higher effects on the word error rate than lexical PM adaptation. Although the number of adaptable parameters is much higher in acoustic modelling the potential gains are higher as the classification occurs at the high-dimensional acoustic feature level that provides more evidence for statistic modelling than the mere symbolic phone strings handled in lexical pronunciation modelling. Despite this fundamentally different initial situation it is noteworthy that there are still small, but measureable improvements, even for the case with acoustic adaptation. In this initial experiment only a single iteration was conducted between acoustic adaptation and PM adaptation. As the pronunciation modelling step leads to an improved

Table 5.7. WER in [%] for unadapted and adapted acoustical models with WFST and CART approach: Optimal thresholds per speaker, best results per group in boldface.

Speaker	**Unadapted**			**Adapted**		
group	base	WFSTc	CARTc	base	WFSTc	CARTc
all	28.45	**28.19**	28.23	17.02	**16.89**	16.90
African	40.39	**38.43**	40.05	25.32	**24.63**	24.97
Asian	31.95	**31.69**	31.79	23.27	23.12	**22.94**
British	23.28	23.10	**22.91**	10.86	**10.70**	10.82
Hispanic	37.64	**37.18**	37.21	21.62	**21.44**	21.48
Indian	37.02	**36.62**	36.81	22.54	**22.37**	22.40
Iranian	32.71	32.61	**32.53**	13.41	**13.35**	13.39
North	18.06	**17.90**	18.00	14.16	**14.05**	14.07
Northeast	24.95	24.77	**24.45**	20.68	**20.61**	20.73
Midland	20.86	20.77	**20.73**	15.16	15.03	**15.02**
Other	41.29	41.01	**40.93**	18.02	**17.85**	17.89
South	20.48	**20.27**	20.28	13.65	**13.58**	13.62
West	19.48	19.30	**19.28**	9.67	**9.54**	9.58

annotation of the acoustic reference for adaptation, consecutive iterations of acoustic adaptation and lexical adaptation might lead to additional improvements.

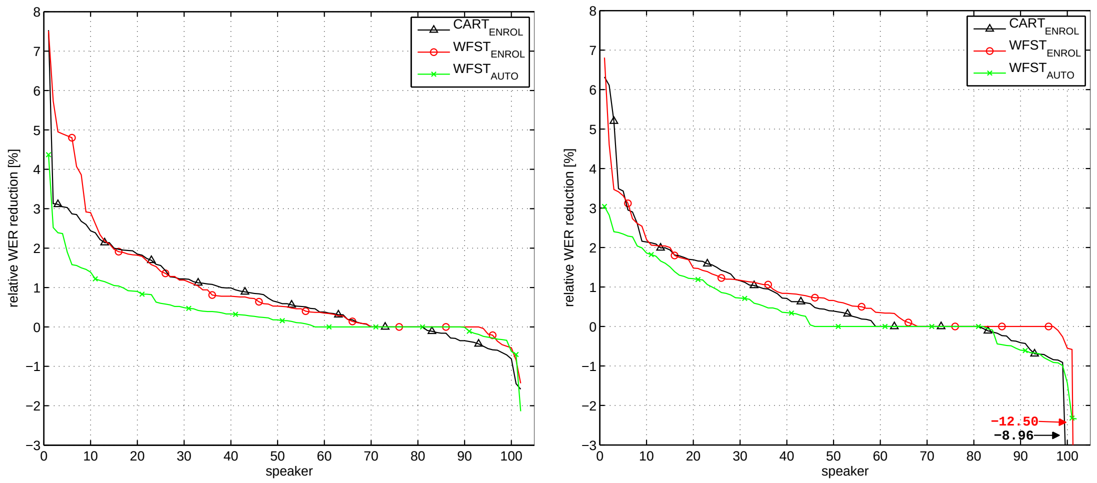
The WFST system performs marginally better than the CART system. The reasons for that are to be found in the explicit insertion model used for the CART system, while in the WFST system insertions are modelled implicitly. Therefore, the generated variants are substantially different which is reflected in the number of suggested pronunciations.

Table 5.7 gives a breakdown of the results reported in table 5.6 according to the initial speaker group definitions (cf. table 5.2). At first, the striking difference in baseline word error rates between non-native and accented native speakers is visible, not so much for the adapted case, but definitely for the unadapted acoustic models. With PM, the baseline rates slightly improve for all groups, apart from the solitary African speaker, who clearly benefits from the adapted lexicon. In absolute figures, the native accents profit less from PM than the non-native ones. While with unadapted acoustic models the best results are balanced between the WFST and CART systems, in the adapted model case, the WFST system performs superior to the CART system.

The PM effects on individual speakers are depicted in figure 5.10 for the results from table 5.6. It is interesting to note that in both cases the WFST system causes a very slight WER degradation for only very few speakers while the recognition rates improve for the majority of speakers. The CART system returns higher improvement rates for a few speakers, but also leads to noteable losses for about 20% of all speakers. From this point of view, the WFST lexicon optimisation approach can be applied without much concerns for performance loss.

5.4 Conclusion

Lexical pronunciation modelling allows for ASR system adaptation without major interventions into the training and decoding processes. For this reason, adaptations at this processing stage may as well be done by ASR service providers for fine-tuning the system to their users



(*a*)

(*b*)

FIGURE 5.10. Relative WER reduction per speaker in [%] (sorted in descending order) for WFST and CART framework with a) unadapted and b) adapted acoustic models.

and task domains. Non-native speakers proved to be relevant candidate speakers for speaker-specific adaptation of the lexical pronunciation model in an experimental evaluation, even more relevant than just accented native speakers of the target language. In contrast to earlier attempts of pronunciation modelling, the newly proposed approach aims at creating speaker-specific pronunciation variants for selected words such that the confusability within the lexicon and the obtained gain in model accuracy are balanced.

The careful extension of an existing lexicon with new pronunciation variants turned out to be the best optimisation strategy compared to building up a completely new lexicon from scratch. Nevertheless, the word error rate reductions accomplished by the optimisation vary a lot between speakers and an automatic, speaker-specific tuning of the optimisation thresholds is essential. In general, the effects of lexicon adaptation are not comparable to acoustic model adaptation, but in conjunction small improvements are still measureable. It would be interesting to use the adapted lexicon already for the training or acoustic adaptation process to maybe further increase the effects.

As far as the initial assumption on the balance between confusability and accuracy within the lexicon is concerned, the experimental evaluation has shown that there are significant differences in the number of proposed pronunciation variants and actually inserted variants. In conjunction with the high variability of the actual inclusion criteria among speakers, it can be concluded that pronunciation model adaptation based on these two measures is directly related to the performance of the system.

In direct comparison to the non-parametric CART method for lexicon adaptation, the WFST method shows comparable performance together with higher robustness towards the modelling of insertions. On a per-speaker basis, the WFST method also does not deteriorate the overall ASR system performance for almost all speakers.

Part III

Further Exploitation and Outlook

Chapter 6

Outlook and Conclusion

6.1 Outlook

This thesis was set up as a first exploration of working with non-literal text resources and phonetic algorithms in speech technology. The previous chapters gave some insights into the specialities of non-literal transcripts via experimental evaluations on well-designed text corpora, and provided methods for solving application-specific problems within a specified non-literal transcript processing framework. Nevertheless, the thesis does not claim to be complete in terms of the methods presented and the full range of applications exploited. The experimental studies rather provide a valuable potential analysis for the chosen approach that allows for further ideas to be realised in a similar fashion.

This outlook is intended to give the reader an idea in which respects the developed algorithms may be extended and to show how versatile the phonetic approach is in its application. The outlook on further research options is divided into three main sections: First, ideas concerning corpora creation with phonetic methods are listed, based on the experiences that were gained during the creation of the text corpus in chapter 2, section 2.5. Second, extensions to the phonetic similarity matching algorithms from chapter 3 are proposed. And finally, further areas of application besides the covered topics in chapter 4 and 5 are discussed.

6.1.1 Further corpus work

In chapter 2, literal and non-literal transcript types were introduced along with text corpora compiled from a pool of available text databases (cf. section 2.5.2, MEDALIGN corpus). The simple text alignment methods used for their creation were not able to solve all problematic issues related to text format discrepancies and tokenisation. With the help of the semantics- and phonetics-driven alignment step of the SPARC method (cf. section 4.3), the MEDALIGN corpus could already be refined in terms of alignment accuracy as shown in chapter 4.

The rule-based reconstruction engine of the SPARC method provides a framework for further and much more precise annotation of such parallel text corpora. Specifically tuned reconstruction rules could label particular phenomena within the alignment automatically and thus help building up knowledge bases of text deviations. Consider as an example the problem of headings within a medical dictation: Only in few cases a heading is dictated as such within the continuous text, but usually, headings are created from the first words of the subsequent paragraph by the transcriptionist. This results in a reformulation of the original dictation and a measurable mismatch between recognised and written text. A specific reconstruction rule which searches for a heading style in the written text and a low phonetic similarity score in conjunction with the recognised text could retrieve all potential heading reformulations. These annotations may provide a deeper understanding of the underlying processes that could be used in the document creation stage. The reconstruction problem could then be inverted into an automatic dictation reformulation system.

Another application of the SPARC method has already been given in chapter 5, where a list of misrecognised words was extracted from the MEDALIGN-INSPECT corpus with the help of two specialised reconstruction rules. The rule framework is not only limited to the semantic and phonetic similarity features, but may be extended with other measures as well that may provide additional information for corpus annotation. The re-aligned and extra-annotated corpora may then in turn be used for re-training the phonetic and possible other similarity measures in a reinforcement learning fashion.

6.1.2 Phonetic algorithm extensions

The experimental evaluation in chapter 3 revealed that for phonetic similarity matching a stochastic string edit distance model returns the best results in a pronunciation classification task. The addition of prior knowledge in terms of phone frequencies provided significant improvements to the similarity estimation. Therefore, it may be beneficial to include further knowledge into the stochastic similarity model.

The most natural extension would be to include syllable structure information. As already mentioned in chapter 2, several studies indicated that pronunciation reduction can directly be explained by the underlying rhythm of the English language. One way for integration could be in the form of a parameter tying scheme in the parameter-rich context dependent model for reducing the number of trainable parameters while still exploiting the benefits of phone context. The syllabic structure of the compared words could also be modelled by a separate random variable on top of the current context-independent edit distance model which would allow the distinction between word-initial and word-final positions.

Apart from additional sources of prior knowledge, the convergence of the state-based and symbol-based similarity measurement approaches is another interesting topic for further research. Although basically incompatible due to the inherent model difference caused by the empty symbol, it would be very helpful to learn the phone confusions directly from the acoustic ASR models, but apply them in a symbol-based comparison framework. This way, the similarity model would be defined by the acoustic properties only and would not suffer from any other recogniser side effects.

6.1.3 More application-driven solutions

With the reconstruction of literal transcriptions and the development of speaker-specific pronunciation models, two solutions to ASR-related problems were proposed in chapters 4 and 5. The methods developed in this thesis can, however, be applied for creating many further applications and supporting tools.

At the acoustic modelling stage, current parameter tying schemes are based upon hand-crafted pronunciation rules that are evaluated in decision trees [131]. Instead of decision trees, learned phone confusion probability distributions may be used as well for motivating parameter reductions which would not require any additional rule definitions. Acoustic modelling in this fashion could on the one hand become more language independent and on the other hand possibly more speaker-dependent as well, assuming that speaker-specific phone confusion matrices are employed. Another application in terms of acoustic modelling could be accent classification of speakers based on small amounts of ASR-generated texts. A confusion matrix trained on these data would be compared to a set of accent-specific confusion matrices and the new speaker assigned to the best-matching model. Such a classification could be used for the pre-selection of accent-specific acoustic models or other accent-specific resources like specialised lexica or language models.

Apart from these online applications phonetic similarity matching could also be applied meaningfully during the design phase of an ASR system. Consider a system with limited, context-dependent vocabulary and controlled input language, such as a speech-controlled form filling system: In this scenario the recognition rate will strongly depend on the confusability inherent in the local grammar rules for each form field, since language modelling effects are minimised due to the limited number of words per utterances. The optimisation framework for confusability within the ASR lexicon (cf. chapter 5, section 5.2) could directly be applied here as well. When designing the grammar rules and contextual vocabularies, this framework allows for immediate control of the acoustic confusability and hence the exclusion or reformulation of critical words.

In a similar fashion the same technology may be utilised in the post-processing phase during transcription correction as well. Knowledge about highly confusable words and frequent ASR errors allows the design of tools which could further accelerate and simplify the work of medical transcriptionists. It is impossible today to imagine document editing without the automatic correction of typing errors and spell checkers which are both built upon orthographic and phonetic similarity measurement. For the case of medical transcription, these tools could be further tuned towards recognition errors for re-ranking the list of word alternatives during editing. Another tool could be built upon the n-best list results from recognition, re-ordered according to phonetic similarity given speaker-specific confusion information. Recognition n-best lists have already been exploited in several works, e.g., with semantic similarity measurements [97], or in specific user interfaces designed for people with special needs [121].

6.2 Conclusion

6.2.1 A critical discussion of the initial hypothesis

The introduction included a formulation of a main hypothesis of this thesis that shall be revisited here. This work was specified to find an answer to the question:

> *Are phonetic/phonologic algorithms suited to overcome the gap between literal and non-literal text resources, such that large amounts of non-literal transcripts can be employed for the development/improvement of medical dictation ASR systems?*

Since this a precise and complex question a critical discussion has to deal with its various aspects. There are three main points of interest. First, whether phonetic/phonologic algorithms were the right choice for reaching the proposed goal. From the literature and the characteristics of the processed data, this approach appeared very promising. For the literal transcript reconstruction task (cf. chapter 4), phonetic similarity matching was directly compared to semantic similarity matching. Phonetic similarity shows its strength with matching on subword-level while for coarser alignment of whole mismatch regions semantic similarity returns more plausible results. From the experiments it can be concluded that phonetic similarity is the stronger of both concepts, although the best results were achieved when they were applied in combination. For pronunciation modelling phonetic algorithms are without alternative anyway.

The second question was whether it was possible to bridge the gap between literal and non-literal transcripts. In this respect, the investigations in chapter 4 proved that a combination of automatically recognised draft transcript and a final corrected medical report is closer to a literal transcription than any of the two original texts. However, it is not possible to fully reconstruct a literal transcript from a non-literal one as missing or heavily deviating passages cannot be recovered. For practical considerations, the improvements in transcript quality are still good enough to show an effect on the ASR system performance.

This directly leads to the third question whether the methods lead to measurable ASR system improvements. The conducted ASR experiments in chapters 4 and 5 are only exemplary as direct comparisons to existing methods could not be implemented due to the minimum invasiveness constraint. For this reason, a general conclusion to this question is difficult. Still, the experiments indicated that there are small, but measurable word error rate reductions for both applications: language model retraining with reconstructed literal transcripts and speaker-specific pronunciation variants. This is a result worth mentioning, as the used ASR system is a professional production type system developed and optimised over many years and not an academic prototype.

Altogether, the methods proved their effectiveness in various tasks, from pronunciation classification to transcript reconstruction and speaker-specific pronunciation prediction. For ultimate system optimisation, it must be noted that phonetic similarity matching cannot be the first choice as it is not competitive enough to outperform existing methods of acoustic model adaptation and language model training. Under the constraint of minimum-invasive system optimisation, the available options are nevertheless exploited as much as possible. And in combination with, e.g., lightly supervised acoustic model training, further positive effects may still occur since the amount and quality of the training material is improved. For this reason, we conclude that the initial hypothesis is confirmed to a large degree.

6.2.2 Remarks on the generalisability of the approach

In the beginning of this work, the scope for the investigations was set tightly to LVCSR systems in the domain of medical dictation for English. This strong restriction was justified by the exemplary character of the evaluations for demonstrating the functionality of the developed methods under real-world conditions. In the light of the results, the question arises how the findings would generalise in a wider context.

One particular benefit of the stochastic phonetic similarity methods presented in chapter 3 is that although they model phonetic and phonologic (language-dependent) knowledge they are still language-independent. Given a training data set of a different language, the relevant information will still be captured in the derived confusion matrices. Therefore, the restriction to English as singular language can easily be disregarded.

Relaxing the domain context is also possible as long as the data resources are not too different from the investigated ones. Basically, the methods should be applicable to all professional dictation scenarios with a similar workflow to medical dictation (cf. figure 2.2), e.g., also in the legal or financial domains. Whenever collections of non-literal transcripts are available for such systems they can be analysed in the same fashion. Another possible domain could be broadcast news transcription. In that case, closed-captions could be utilised as non-literal text resource if the amount of similarity to the draft transcription from the recogniser is about the same as in the medical scenario.

For ASR tasks other than LVCSR, the applicability is at least questionable. The proposed alignment techniques rely on continuous text data and will surely lose some of their potential in comparison to simpler alignment methods on shorter inputs. Instead, applications in text processing and data mining are definitely imaginable, e.g., the direct comparison of different versions of the same document, or plagiarism detection.

Altogether, the presented algorithms of this thesis are probably applicable within a wider scope than it was set in the beginning. For definite statements on particular scope extensions, however, more research has to be conducted.

Part IV

Appendix

Appendix A

Definitions

Phonetic symbol alphabets

Table A.1. Phonetic symbol sets: International Phonetic Alphabet (IPA), Philips Speech Processing Phonetic Alphabet (PSPPA), Speech Assessment Methods Phonetic Alphabet (SAMPA), Advanced Research Projects Agency Alphabet (ARPABET). Note: Symbols #47 to #53 are custom extensions by ICSI to standard ARPABET.

#	IPA	PSPPA	SAMPA	ARPABET	Example
1		si	#sil#	SIL	pre-and-post
2	æ	&	{	AE	bad
3	ɑ	A	A:	AA	car
4	ɒ	Q	Q		law
5	ɔ	o	O:	AO	north
6	i	i	i:	IY	free
7	e	e	e	EH	America
8	eɪ	Y	eI	EY	eight
9	ɪ	I	I	IH	kit
10	oʊ	O	oU	OW	low
11	ʊ	U	U	UH	sure
12	u	u	u:	UW	fool
13	aʊ	W	aU	AW	town
14	ʌ	V	V	AH	cup
15	ɜ	3	3:	ER	bird
16	b	b	b	B	black
17	tʃ	C	tS	CH	choose
18	d	d	d	D	do
19	ð	D	D	DH	other
20	f	f	f	F	fat
21	g	g	g	G	get
22	h	h	h	HH	hot

Continued on next page...

TABLE A.1. Phonetic symbol sets: International Phonetic Alphabet (IPA), Philips Speech Processing Phonetic Alphabet (PSPPA), Speech Assessment Methods Phonetic Alphabet (SAMPA), Advanced Research Projects Agency Alphabet (ARPABET). Note: Symbols #47 to #53 are custom extensions by ICSI to standard ARPABET.

#	IPA	PSPPA	SAMPA	ARPABET	Example
23	dʒ	J	dZ	JH	jar
24	k	k	k	K	key
25	l	l	l	L	light
26	m	m	m	M	more
27	n	n	n	N	now
28	ŋ	G	N	NG	ring
29	p	p	p	P	pen
30	r (ʁ,ʀ)	r	r	R	right
31	s	s	s	S	soon
32	ʃ	S	S	SH	share
33	t	t	t	T	ten
34	θ	T	T	TH	thumb
35	ə	@	@	AX	afford
36	^{ə}l	L	l=	EL	little
37	^{ə}m	M	m=	EM	socialism
38	^{ə}n	N	n=	EN	cotton
39	v	v	v	V	move
40	w	w	w	W	wet
41	j	j	j	Y	yet
42	z	z	z	Z	zero
43	ʒ	Z	Z	ZH	vision
44	aɪ	2	aI	AY	pride
45	ɔɪ	9	oI	OY	joy
46	^{ə}r	R	r=		baker
47	ɾ			DX	butter
48	ɨ			IX	roses
49				LG	*lateral glide*
50	ɾ̃			NX	*nasal flap*
51				PV	*filled pause*
52	ʔ			Q	uh-oh
53	ʉ			UX	suit

Phone classes

Table A.2. Phonetic class assignment in ARPABET and SAMPA phonetic alphabet notations.

Phonetic class	ARPABET	SAMPA
vowel	/AA AE AH AO AW AX AY EH ER EY IH IY OW OY UH UW/	/A { V O aU @ aI e 3 eI I i oU oI U u/
front vowel	/AE AY AW EH EY IH IY OY/	/{ aI aU e eI I i oI/
central vowel	/ER AW AY OW UH/	/3 aU aI oU U/
back vowel	/AA AW OY AO OW UH AH UW/	/A aU oI O oU U V u/
closed vowel	/AW EY IH OY OW UH UW W Y AY/	/aU eI I oI oU U u w j aI/
closed mid vowel	/AY ER AW EY IH OY OW UH/	/aI 3 aU eI I oI oU U/
open vowel	/AE AA AY AW OH OW/	/{ A aI aU Q oU/
open mid vowel	/AE EH EY OY AO AH AX/	/{ e eI oI O V @/
diphtong	/AY AW EY OY OW/	/aI aU eI oI oU/
voiced	/ER AE AA AY AW EH EY IH IY OH OY AO OW UH AH UW B D DH G L M N NG R V Y Z/	/3 { A aI aU e eI I i Q oI O oU U V u b d D g N v j z l= m= n= r=/
labial	/B F M P V/	/b f m p v m=/
fricative	/CH DH F JH HH S SH TH V Z ZH/	/tS D f dZ h s S T v z Z/
anterior+	/D DH L N R S T TH Z/	/d D l n r s t T z n= r=/
anterior−	/AE AY AW EH EY IH IY OY CH JH NG SH Y ZH/	/{ aI aU e eI I i oI tS dZ N S j Z/
dorsal	/AA AW OH OY AO OW UH AH UW G K N NG W/	/A aU Q oI O oU U V u g k n N w n=/
stop	/B CH D G JH K M N NG P T/	/b tS d g dZ k m n N p t m= n=/
approximant	/ER AE AA AY AW EH EY IH IY OH OY AO OW UH AH UW L R W Y/	/3 { A aI aU e eI I i Q oI O oU U V u l r w j l= r=/
sibilant	/CH JH S SH Z ZH/	/tS dZ s S z Z ts/
nasal	/M N NG/	/m n N m= n=/
plosive	/B D G K P T/	/b d g k p t/
liquid	/L R W Y/	/l r w j l= r=/
silence	/SIL/	/#sil#/

Part-of-speech tags from the Penn Treebank corpus

TABLE A.3. Lexical category tags from the Penn Treebank Tag Set [73] used for CART pronunciation modelling.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition/subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NP	Proper noun, singular
NPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PP	Personal pronoun
PP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Appendix B

MEDTRANS Corpus Details

Word and phone frequency statistics

Table B.1. Word counts (WC) and pronunciation variant counts (VC) from the MEDTRANS corpus, subsets A and B.

Rank	WC	Word	VC	Examples
1	4,276	[hes]	65	/&/, /& m/, /A m/, /d A/, /@ m/, /A & m/, /m A/, /e h/, /m/
2	3,109	the	32	/D/, /D &/, /D @/, /D I/, /D e/, /D i/, /T @/, /d @/, /d i/
3	2,735	and	46	/&/, /& G/, /& n/, /& n d/, /& n t/, /@ n/, /@ n d/, /@ n t/, /n/, /n d/
4	2,347	is	12	/@ z/, /I s/, /I z/, /i z/, /z/
5	2,300	of	25	/@ f/, /@ v/, /A v/, /O f/, /Q v/, /V v/, /Q f/, /V f/
6	2,271	a	20	/&/, /2/, /@/, /A/, /Y/, /A Y/, /d 2/, /s m Q l si Y/
7	2,075	-A	21	/&/, /@/, /A/, /A A/, /A h A/, /A m/, /A m A/, /D @/, /Y/
8	1,803	to	23	/T @/, /d @/, /d u/, /t @/, /t O/, /t U/, /t u/
9	1,761	period	19	/p @ r @ d/, /p I r @ d/, /p I r i & d/, /p I r i @ d/, /p I r i @ t/
10	1,590	was	22	/w @/, /w @ s/, /w @ z/, /w A z/, /w O Z/, /w Q z/, /w V z/
...				
99	170	physical	9	/f I z @ k L/, /f I z I g @ l/, /f I z I g L/, /f I z I k @ l/, /f I z I k L/
100	169	who	2	/A h u/, /h u/
101	164	back	5	/b & d/, /b & g/, /b & k/, /l & k/, /v & k/
102	163	S	3	/@ s/, /I s/, /e s/
103	161	other	8	/A D R/, /O D R/, /Q D R/, /V D 3 r/, /V D @/, /V D @ r/
104	152	regular	26	/r e g @ l @ r/, /r e g @ l R/, /r e g j @ R/, /r e g j @ l 3 r/
...				
999	12	discuss	3	/d I s g @ s/, /d I s g V s/, /d I s k V s/
1,000	12	diagnosed	1	/d 2 @ g n O z d/
1,001	12	deficits	3	/d e f @ s @ t s/, /d e f @ s I t s/, /d e f s @ t s/
1,002	12	culture	2	/k @ l C R/, /k V l C R/
1,003	12	consider	2	/k @ n s I d @ r/, /k @ n s I d R/
1,004	12	consciousness	3	/k A n S @ s n @ s/, /k A n S n @ s/

Table B.2. List of the 100 most frequent phone confusions observed in the MEDTRANS corpus: canonical pronunciation (C) and observed pronunciation (O).

#	Count	C	O	#	Count	C	O	#	Count	C	O	#	Count	C	O
1	3,025	d	t	26	192	.	l	51	101	O	A	76	58	V	A
2	2,779	@	.	27	192	A	@	52	100	e	.	77	56	.	b
3	1,388	r	R	28	188	.	h	53	94	@	N	78	55	&	A
4	1,342	.	d	29	185	.	&	54	92	.	z	79	55	@	Y
5	1,183	.	t	30	165	Q	A	55	91	.	e	80	50	.	L
6	1,172	n	N	31	161	.	n	56	91	t	.	81	50	U	V
7	966	.	@	32	160	.	v	57	91	u	@	82	49	O	.
8	915	@	I	33	156	n	G	58	84	r	.	83	48	@	2
9	752	I	@	34	155	U	@	59	80	.	N	84	47	.	A
10	633	l	L	35	147	A	m	60	79	A	.	85	47	A	V
11	631	.	m	36	139	.	R	61	78	@	o	86	46	3	e
12	478	O	V	37	139	I	.	62	73	Q	V	87	46	O	Q
13	476	@	V	38	137	@	&	63	72	O	o	88	45	k	.
14	384	O	@	39	137	@	t	64	72	m	M	89	43	e	i
15	353	d	.	40	137	n	.	65	68	@	u	90	40	e	Y
16	314	A	&	41	135	I	i	66	67	.	O	91	39	V	O
17	288	.	I	42	125	@	R	67	67	.	w	92	39	j	.
18	286	.	i	43	124	g	k	68	67	e	@	93	39	t	T
19	251	f	v	44	121	.	s	69	65	S	C	94	38	.	u
20	229	.	j	45	119	@	A	70	65	s	.	95	38	@	r
21	223	@	i	46	119	@	O	71	64	A	O	96	37	.	p
22	218	@	e	47	114	s	z	72	64	n	t	97	37	@	3
23	200	3	.	48	112	.	k	73	63	3	@	98	36	.	V
24	199	&	@	49	112	.	r	74	62	&	.	99	36	I	e
25	194	t	d	50	112	i	.	75	60	&	e	100	36	w	.

Transcription label & annotation deviation category statistics

Table B.3. MEDTRANS annotation deviation category distribution – sorted by category.

Cat.	**A**		**B**		$\mathbf{C} \in \mathbf{B}$		$\mathbf{A} \cup \mathbf{B}$	
#	Count	[%]	Count	[%]	Count	[%]	Count	[%]
11	1,466	8.18	1,977	8.49	407	9.06	3,850	8.63
12	123	0.69	655	2.81	155	3.45	933	2.09
13	206	1.15	333	1.43	35	0.77	574	1.29
14	376	2.10	288	1.24	40	0.89	704	1.58
15	279	1.56	289	1.24	49	1.09	617	1.38
16	2,011	11.22	3,070	13.18	659	14.67	5,740	12.87
21	323	1.80	327	1.40	67	1.49	717	1.61
22	461	2.57	180	0.77	27	0.60	668	1.50
23	381	2.13	1,612	6.92	166	3.69	2,159	4.84
24	1,770	9.87	498	2.14	127	2.82	2,395	5.37
25	262	1.46	363	1.56	52	1.15	677	1.52
26	1,157	6.45	1,967	8.45	311	6.92	3,435	7.70
27	239	1.33	1,288	5.53	155	3.45	1,682	3.77
28	3,729	20.80	6,254	26.86	1,442	32.11	11,425	25.61
29	1,403	7.83	1,342	5.76	360	8.01	3,105	6.96
31	757	4.22	159	0.68	35	0.77	951	2.13
32	840	4.69	21	0.09	2	0.04	863	1.93
33	83	0.46	79	0.34	12	0.26	174	0.39
34	88	0.49	705	3.03	72	1.60	865	1.94
41	872	4.86	1,076	4.62	179	3.98	2,127	4.77
42	0	0.00	0	0	0	0	0	0
43	101	0.56	0	0	0	0	101	0.23
44	853	4.76	505	2.17	84	1.87	1,442	3.23
45	147	0.82	298	1.28	54	1.20	499	1.12

Table B.4. MEDTRANS annotation deviation category distribution – sorted by count.

Cat.	**A**		**Cat.**	**B**		**Cat.**	**C ∈ B**		**Cat.**	**A ∪ B**	
#	Count	[%]	#	Count	[%]	#	Count	[%]	#	Count	[%]
28	3,729	20.80	28	6,254	26.86	28	1,442	32.11	28	11,425	25.61
16	2,011	11.22	16	3,070	13.18	16	659	14.67	16	5,740	12.87
24	1,770	9.87	11	1,977	8.49	11	407	9.06	11	3,850	8.63
11	1,466	8.18	26	1,967	8.45	29	360	8.01	26	3,435	7.70
29	1,403	7.83	23	1,612	6.92	26	311	6.92	29	3,105	6.96
26	1,157	6.45	29	1,342	5.76	41	179	3.98	24	2,395	5.37
41	872	4.86	27	1,288	5.53	23	166	3.69	23	2,159	4.84
44	853	4.76	41	1,076	4.62	27	155	3.45	41	2,127	4.77
32	840	4.69	34	705	3.03	12	155	3.45	27	1,682	3.77
31	757	4.22	12	655	2.81	24	127	2.82	44	1,442	3.23
22	461	2.57	44	505	2.17	44	84	1.87	31	951	2.13
23	381	2.13	24	498	2.14	34	72	1.60	12	933	2.09
14	376	2.10	25	363	1.56	21	67	1.49	34	865	1.94
21	323	1.80	13	333	1.43	45	54	1.20	32	863	1.93
15	279	1.56	21	327	1.40	25	52	1.15	21	717	1.61
25	262	1.46	45	298	1.28	15	49	1.09	14	704	1.58
27	239	1.33	14	288	1.24	14	40	0.89	25	677	1.52
13	206	1.15	15	289	1.24	31	35	0.77	22	668	1.50
45	147	0.82	22	180	0.77	13	35	0.77	15	617	1.38
12	123	0.69	31	159	0.68	22	27	0.60	13	574	1.29
43	101	0.56	33	79	0.34	33	12	0.26	45	499	1.12
34	88	0.49	32	21	0.09	32	2	0.04	33	174	0.39
33	83	0.46	43	0	0.00	42	0	0.00	43	101	0.23
42	0	0.00	42	0	0.00	43	0	0.00	42	0	0.00

Transcriber labelling agreement

Table B.5. Mapping between transcribers T_i and reports for subset C of the MEDTRANS corpus.

Report ID	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{T}_3$	$\mathbf{T}_4$	$\mathbf{T}_5$	$\mathbf{T}_6$	$\mathbf{T}_7$	$\mathbf{T}_8$	$\mathbf{T}_9$
A092054F00037ER	X						X		
A092054F00049ER	X				X				
A092054F00054ER	X				X				
A092054F00066ER	X				X				
A092054F00068ER	X				X				
A092054F93436ER	X					X	X		
A092054F93440ER	X					X			
A092054F93443ER	X		X						
A092054F93456AD	X			X					
A092054F93513ER	X		X	X					
A092054F93606AD	X		X						
A092054F93613AD	X	X	X						
A092054F95170AD	X	X	X						
A092054F95212AD	X		X						
G403501F92221CL								X	X
G403501F94484CL								X	X
G403501F97030CL								X	X
G403501F97628CL								X	X

Bibliography

[1] W. A. Ainsworth, "Can phonetic knowledge be used to improve the performance of speech recognisers and synthesisers," in *The Integration of Phonetic Knowledge in Speech Technology*, W. J. Barry and W. A. van Dommelen, Eds. Dordrecht, The Netherlands: Springer, 2005, pp. 13–19.

[2] I. Amdal, "Learning pronunciation variation - a data-driven approach to rule-based lexicon adaptation for automatic speech recognition," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Trondheim, Norway, 2002.

[3] I. Amdal, F. Korkmazskiy, and A. C. Surendran, "Data-driven pronunciation modelling for non-native speakers using association strength between phones," in *Proc. ISCA Tutorial & Research Workshop (ITRW) 'Automatic Speech Recognition: Challenges for the new Millennium'*, Paris, France, 2000, pp. 85–90.

[4] L. M. Arslan, "Foreign accent classification in American English." Ph.D. dissertation, Duke University, Durham, North Carolina, 1996.

[5] L. M. Arslan and J. H. L. Hansen, "Frequency characteristics of foreign accented speech," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, Germany, 1997, pp. 1123–1126.

[6] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Transactions on Information Theory*, vol. 21, pp. 404–411, 1975.

[7] W. J. Barry and W. A. van Dommelen, *The Integration of Phonetic Knowledge in Speech Technology*, W. J. Barry and W. A. van Dommelen, Eds. Dordrecht, The Netherlands: Springer, 2005.

[8] J. Bilmes and G. Zweig, "The Graphical Models Toolkit: An open source software system for speech and time-series processing," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, Florida, 2002, pp. 3916–3919.

[9] C. Bishop, *Pattern Recognition and Machine Learning.* New York City, New York: Springer, 2006.

[10] P. Boersma and D. Weenink, "PRAAT: Doing phonetics by computer," Computer Program, 2009. [Online]. Available: http://www.praat.org

[11] G. Bouselmi, D. Fohr, I. Illina, and J. P. Haton, "Fully automated non-native speech recognition using confusion-based acoustic model integration," in *Proceedings of INTERSPEECH*, Lisbon, Portugal, 2005, pp. 1369–1372.

[12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* Belmont, California: Wadsworth International Group, 1984.

[13] H. Bunke and J. Csirik, "Parametric string edit distance and its application to pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, pp. 202–206, 1995.

[14] A. Burgun and O. Bodenreider, "Comparing terms, concepts and semantic classes in WordNet and the Unified Medical Language System," in *Proceedings of NAACL'2001 Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburgh, Pennsylvania, 2001, pp. 77–82.

[15] L. A. Byrne, D. S. Heath, B. J. Hurley, K. Rockel, and C. Tessier, *The AAMT Book of Style for Medical Transcription*, P. Hughes, Ed. American Association for Medical Transcription, 1995.

[16] J.-Y. Chen, P. A. Olsen, and J. R. Hershey, "Word confusability - measuring Hidden Markov similarity," in *Proceedings of INTERSPEECH*, Antwerp, Belgium, 2007, pp. 2089–2092.

[17] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modelling," in *Proceedings of the 34th annual meeting of the Association for Computational Linguistics (ACL)*, Santa Cruz, California, 1996, pp. 310–318.

[18] K. W. Church and W. A. Gale, "Probability Scoring for Spelling Correction," *Statistics and Computing*, vol. 1, no. 2, pp. 93–103, 1991.

[19] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 1960.

[20] E. Coussé, S. Gillis, H. Kloots, and M. Swerts, "The Influence of the Labeller's Regional Background on Phonetic Transcriptions:Implications for the Evaluation of Spoken Language Resources," in *Proceedings of the Intl. Conf. on Language Resources and Evaluation (LREC)*, Lissabon, Portugal, May 2004, pp. 1447–1450.

[21] J. M. Dalby, "Phonetic structure of fast speech in American English," Ph.D. dissertation, Indiana University, Bloomington, Indiana, 1986.

[22] F. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, vol. 7, pp. 171–176, 1964.

[23] N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker, and J. Picone, "Resegmentation of SWITCHBOARD," in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998, pp. 1543–1546.

[24] S. Dobrišek, J. Žibert, N. Pavešić, and F. Mihelič, "An edit-distance model for the approximate matching of timed strings," *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 31, no. 4, pp. 736–741, 2009.

[25] B. Eisen, H.-G. Tillman, and C. Draxler, "Consistency of judgements in manual labelling of phonetic segments: The distinction between clear and unclear cases," in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Banff, Canada, 1992, pp. 871–874.

[26] C. Fellbaum, *WordNet: An Electronic Lexical Database.* Cambridge, Massachusetts: MIT Press, 1998.

[27] K. Filali and J. Bilmes, "A dynamic Bayesian framework to model context and memory in edit distance learning: An application to pronunciation classification," in *Proceedings of the ACL*, Ann Arbor, Michigan, 2005, pp. 338–345.

[28] J. Flege, "Effects of equivalence classification on the production of foreign language speech sounds." in *Sound Patterns in Second Language Acquisition*, A. James and J. Leather, Eds. Dordrecht, The Netherlands: Foris publications, 1988, pp. 9–39.

[29] E. Fosler-Lussier, "Dynamic pronunciation models for automatic speech recognition," Ph.D. dissertation, Univ. of California Berkeley, 1999.

[30] ——, "A tutorial on pronunciation modelling for large vocabulary speech recognition," in *Text- and Speech-Triggered Information Access*, ser. Lecture Notes in Computer Science, G. Grefenstette and S. Renals, Eds. Berlin, Germany: Springer, 2003, no. 2705, pp. 38–77.

[31] E. Fosler-Lussier, I. Amdal, and H.-K. J. Kuo, "A framework for predicting speech recognition errors," *Speech Communication*, vol. 46, pp. 153–170, 2005.

[32] E. Fosler-Lussier and N. Morgan, "Effects of speaking rate and word frequency on conversational pronunciations," in *Proc. of the ESCA Workshop 'Modeling Pronunciation Variation For Automatic Speech Recognition'*, Rolduc, The Netherlands, 1998, pp. 35–40.

[33] D. Gibbon, R. Moore, and R. Winski, *Handbook of Standards and Resources for Spoken Language Systems.* Walter de Gruyter Publishers, Berlin & New York: Mouton de Gruyter, 1997.

[34] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, California, 1992, pp. 517–520.

[35] J. Goldberger, S. Gordon, and H. Greenspan, "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures," in *Proceedings of IEEE Intl. Conf. on Computer Vision (ICCV)*, Nice, France, 2003, pp. 487–493.

[36] S. Goronzy, *Robust Adaptation to Non-Native Accents in Automatic Speech Recognition*, ser. Lecture Notes in Artificial Intelligence, J. Carbonell and J. Siekmann, Eds. Berlin, Germany: Springer, 2002, no. 2560.

[37] S. Greenberg, "Speaking in shorthand: A syllable-centric perspective for understanding pronunciation variation," in *Proc. of the ESCA Workshop 'Modeling Pronunciation Variation For Automatic Speech Recognition'*, Rolduc, The Netherlands, 1998, pp. 47–56.

[38] ——, "From here to utility," in *The Integration of Phonetic Knowledge in Speech Technology*, W. J. Barry and W. A. van Dommelen, Eds. Dordrecht, The Netherlands: Springer, 2005, pp. 107–131.

[39] S. Greenberg, S. Chang, and J. Hollenback, "An introduction to the diagnostic evaluation of the SWITCHBOARD-corpus automatic speech recognition systems," in *Proceedings of the NIST Speech Transcription Workshop*, College Park, Maryland, 2000.

[40] S. Greenberg and E. Fosler-Lussier, "The uninvited guest: Information's role in guiding the production of spontaneous speech," in *Proc. CREST Workshop on Models of Speech Production: Motor Planning and Articulatory Modelling*, Kloster Seeon, Germany, 2000.

[41] S. Greenberg, J. Hollenback, and D. Ellis, "Insights into spoken language gleaned from phonetic transcription of the SWITCHBOARD corpus," in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, Pennsylvania, 1996, pp. 24–27.

[42] T. Hain, "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech Communication*, vol. 46, pp. 171–188, 2005.

[43] M. Hammond, "Syllable parsing in English and French," Rutgers Optimality Archive, 1995, http://roa.rutgers.edu/.

[44] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings," in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Pittsburgh, Pennsylvania, 2006, pp. 1606–1609.

[45] W. Heeringa, "Measuring dialect pronunciation differences using Levenshtein distance," Ph.D. dissertation, University of Groningen, Groningen, The Netherlands, 2004.

[46] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007, pp. 317–320.

[47] ——, "Variational Bhattacharyya divergence for Hidden Markov models," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, Nevada, 2008, pp. 4557–4560.

[48] T. Holter and T. Svendsen, "Maximum likelihood modelling of pronunciation variation," *Speech Communication*, vol. 29, pp. 177–191, 1999.

[49] M. Huber, J. Jancsary, A. Klein, J. Matiasek, and H. Trost, "Mismatch interpretation by semantics-driven alignment," in *Proceedings of KONVENS*, Konstanz, Germany, 2006.

[50] *The International Phonetic Alphabet*, The International Phonetic Association, 2005, http://www.langsci.ucl.ac.uk/ipa/ipachart.html.

[51] J. Janscary, A. Klein, J. Matiasek, and H. Trost, "Semantics-based automatic literal reconstruction of dictations," in *Proc. Workshop on the Semantic Representation of Spoken Language (SRSL07), CAEPIA - TTIA*, Salamanca, Spain, 2007.

[52] F. Jelinek, L. Bahl, and R. Mercer, "Design of a linguistic statistical decoder for the recogition of continuous speech," *IEEE Transactions on Information Theory*, vol. 21, pp. 250–256, 1975.

[53] K. Johnson, “Massive reduction in conversational American English,” in *Proceedings of the Workshop on Spontaneous Speech: Data and Analysis*, Tokyo, Japan, 2002, pp. 29–54.

[54] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Pearson Education, 2009.

[55] C.-M. Karat, C. Halverson, D. Horn, and J. Karat, “Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, Pittsburgh, Pennsylvania, 1999, pp. 568–575.

[56] T. Kemp and A. Waibel, “Unsupervised training of a speech recognizer: recent experiments,” in *Proceedings of the European Conf. on Speech Communication and Technology (EUROSPEECH)*, Budapest, Hungary, 1999, pp. 2725–2728.

[57] A. Kipp, B. Wesenick, and F. Schiel, “Pronunciation modeling applied to automatic segmentation of spontaneous speech.” in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Rhodes, Greece, 1997, pp. 1023–1026.

[58] S. Kirkpatrick, C. G. Jr., and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[59] G. Kondrak, “Phonetic alignment and similarity,” *Computers and the Humanities*, vol. 3, no. 37, pp. 273–291, August 2003.

[60] Z. Kövecses, *American English: An Introduction.* Peterborough, Ontario, Canada: Broadview Press, 2000.

[61] W. A. Kretzschmar, “Standard American English pronunciation,” in *A Handbook of Varieties of English: a multimedia reference tool*, E. W. Schneider, K. Burridge, B. Kortmann, R. Mesthrie, and C. Upton, Eds. Berlin, Germany: Mouton de Gruyter, 2004, vol. 1, pp. 257–269.

[62] J. Lai and J. Vergo, “MedSpeak: Report creation with continuous speech recognition,” in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, Atlanta, Georgia, 1997, pp. 431–438.

[63] L. Lamel and G. Adda, “On designing pronunciation lexicons for large vocabulary, continuous speech recognition,” in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, PA, 1996, pp. 6–9.

[64] L. Lamel, J.-L. Gauvain, and G. Adda, “Lightly supervised and unsupervised acoustic model training,” *Computer Speech and Language*, vol. 16, pp. 115–129, 2002.

[65] C.-H. Lee, C.-H. Lin, and B.-H. Juang, “A study on speaker adaptation of continuous density HMM parameters,” in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Albuquerque, New Mexico, 1990, pp. 145–148.

[66] C. Leggetter and P. Woodland, “Speaker adaptation of continuous density HMMs using multivariate linear regression,” in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Yokohama, Japan, 1994, pp. 451–454.

[67] C. Leitner, “Data-based automatic phonetic transcription,” Master’s thesis, Graz University of Technology, Graz, Austria, 2008.

[68] W. Levelt, *Speaking: From intention to articulation.* Cambridge, Massachusetts: MIT Press, 1989.

[69] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals." *Soviet Physics - Doklady*, vol. 10, pp. 707–710, 1966.

[70] D. Lindberg, B. Humphreys, and A. McCray, "The Unified Medical Language System," *Methods of Information in Medicine*, vol. 32, pp. 281–291, 1993.

[71] K. Livescu and J. Glass, "Lexical modeling of non-native speech for automatic speech recognition," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, 2000, pp. 1683–1686.

[72] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* Cambridge, Massachusetts: MIT Press, 1999.

[73] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn treebank," *Computational Linguistics*, vol. 19, pp. 313–330, 1993.

[74] A. Marzal and E. Vidal, "Computation of Normalized Edit Distance and Applications," *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 15, no. 9, pp. 926–932, 1993.

[75] W. J. Masek and M. S. Paterson, "A faster algorithm computing string edit distances," *Journal of Computer and System Sciences*, vol. 20, pp. 18–31, 1980.

[76] A. McCallum, K. Bellare, and F. Pereira, "A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance," in *Proceedings of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, Arlington, Virginia, 2005, pp. 388–395.

[77] R. I. McDavid, *Varieties of American English*, A. S. Dil, Ed. Stanford, California: Stanford University Press, 1980.

[78] G. A. Miller and P. E. Nicely, "An analysis of perceptual confusions among some english consonants," *Journal of the Acoustical Society of America*, vol. 27, pp. 338–352, 1955.

[79] N. Mirghafori, E. Fosler-Lussier, and N. Morgan, "Towards robustness to fast speech in ASR," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Atlanta, Georgia, 1996, pp. 335–338.

[80] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, pp. 69–88, 2002.

[81] R. K. Moore, "A comparison of the data requirements of automatic speech recognition systems and human listeners," in *Proceedings of the European Conf. on Speech Communication and Technology (EUROSPEECH)*, Geneva, Switzerland, 2003, pp. 2581–2584.

[82] E. Moulines and J. Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech," *Speech Communication*, vol. 16, no. 2, pp. 175–205, 1995.

[83] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, pp. 31–88, 2001.

[84] J. P. Olive, A. Greenwood, and J. Coleman, *Acoustics of American English Speech: A Dynamic Approach.* New York City, New York: Springer, 1993.

[85] B. J. Oomen and R. K. S. Loke, “On using parametric string distances and vector quantization in designing syntactic pattern recognition systems,” in *Proc. IEEE Intl. Conf. Systems, Man, and Cybernetics*, Orlando, Florida, 1997, pp. 511–517.

[86] D. O’Shaughnessy, “Automatic speech recognition: History, methods, and challenges,” *Pattern Recognition*, vol. 41, pp. 2665–2679, 2008.

[87] S. Pakhomov, M. Schonwetter, and J. Bachenko, “Generating training data for medical dictations,” in *Proceedings of the NAACL*, Pittsburgh, Pennsylvania, 2001, pp. 1–8.

[88] T. Pedersen, S. Pakhomov, S. Patwardhan, and C. G. Chute, “Measures of semantic similarity and relatedness in the biomedical domain,” *Journal of Biomedical Informatics*, vol. 40, pp. 288–299, 2007.

[89] J. Peters and C. Drexel, “Transformation-Based Error Correction for Speech-to-Text Systems,” in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Jeju Island, Korea, 2004, pp. 1449–1452.

[90] S. Petrik and G. Kubin, “Reconstructing medical dictations from automatically recognized and non-literal transcripts with phonetic similarity matching,” in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007, pp. 1125–1128.

[91] S. Petrik and F. Pernkopf, “Automatic phonetics-driven reconstruction of medical dictations on multiple levels of segmentation,” in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 4317–4320.

[92] ——, “Language model adaptation for medical dictations by automatic phonetics-driven transcript reconstruction,” in *Proceedings of the IASTED Intl. Conf. on Artificial Intelligence*, Innsbruck, Austria, 2008, pp. 194–199.

[93] *Phonetic Transcription Guideline - US English V2*, Philips Speech Processing, Vienna, Austria, 1996.

[94] M. A. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, “The Buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability,” *Speech Communication*, vol. 45, pp. 89–95, 2005.

[95] A. Prince and P. Smolensky, *Optimality Theory: Constraint Interaction in Generative Grammar.* Oxford, England: Blackwell, 2004.

[96] H. Printz and P. A. Olsen, “Theory and practice of acoustic confusability,” *Computer Speech and Language*, vol. 16, pp. 131–164, 2002.

[97] M. Pucher, “Semantic similarity in automatic speech recognition for meetings,” Ph.D. dissertation, Graz University of Technology, Graz, Austria, 2007.

[98] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, A. V. Oppenheim, Ed. Upper Saddle River, New Jersey: Prentice-Hall, 1993.

[99] A. Rastrow, A. Sethy, and B. Ramabhadran, “A new method for OOV detection using hybrid word/fragment system,” in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipeh, Taiwan, 2009, pp. 3953–3956.

[100] P. Rentzepopoulos and G. Kokkinakis, “Efficient multilingual phoneme-to-grapheme conversion based on HMM,” *Computational Linguistics*, vol. 22, pp. 351–376, 1996.

[101] B. Resch, “Data driven pronunciation modelling for large vocabulary spontaneous speech recognition,” Master’s thesis, Graz University of Technology, Graz, Austria, 2002.

[102] E. K. Ringger and J. F. Allen, “A fertility channel model for post-correction of continuous speech recognition,” in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, Pennsylvania, 1996, pp. 897–900.

[103] E. S. Ristad and P. N. Yianilos, “Learning String-Edit Distance,” *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[104] ——, “A surficial pronunciation model,” in *Proc. of the ESCA Workshop ’Modeling Pronunciation Variation For Automatic Speech Recognition’*, Rolduc, The Netherlands, 1998, pp. 117–119.

[105] D. Sankoff and J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Reading, Massachusetts: Addison-Wesley, 1983.

[106] M. Saraçlar, H. Nock, and S. Khudanpur, “Pronunciation modelling by sharing gaussian densities across phonetic models,” *Computer Speech and Language*, vol. 14, pp. 137–160, 2000.

[107] S. Schaden, “Generating non-native pronunciation lexicons by phonological rules,” in *Proceedings of the Intl. Conf. on Phonetic Sciences (ICPhS)*, Barcelona, Spain, 2003, pp. 2545–2548.

[108] ——, “Evaluation of automatically generated transcriptions of non-native pronunciations using a phonetic distance measure,” in *Proceedings of the Intl. Conf. on Language Resources and Evaluation (LREC)*, Genoa, Italy, 2006, pp. 2441–2446.

[109] F. Schiel, A. Kipp, and H. Tillmann, “Statistical modelling of pronunciation: It’s not the model, it’s the data,” in *Proc. of the ESCA Workshop ’Modeling Pronunciation Variation for Automatic Speech Recognition’*, Rolduc, The Netherlands, 1998, pp. 131–136.

[110] H. Schramm, “Modelling spontaneous speech variability for large vocabulary continuous speech recognition,” Ph.D. dissertation, RWTH Aachen, Aachen, Germany, 2006.

[111] H. Schramm and X. Aubert, “Efficient integration of multiple pronunciations in a large vocabulary decoder,” in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, 2000, pp. 1659–1662.

[112] T. Schultz and K. Kirchhoff, Eds., *Multilingual Speech Processing.* Burlington, Massachusetts: Academic Press, 2006.

[113] L. Shockey, *Sound Patterns of Spoken English*, L. Shockey, Ed. Oxford, England: Blackwell Publishing, 2003.

[114] T. Sloboda and A. Waibel, “Dictionary learning for spontaneous speech recognition,” in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, Pennsylvania, 1996, pp. 2328–2331.

[115] H. Strik, “Pronunciation adaptation at the lexical level,” in *Proc. of the ISCA Tutorial & Research Workshop (ITRW) 'Adaptation Methods For Speech Recognition'*, Sophia-Antipolis, France, 2001, pp. 123–131.

[116] ——, “Is phonetic knowledge of any use for speech technology?” in *The Integration of Phonetic Knowledge in Speech Technology*, W. J. Barry and W. A. van Dommelen, Eds. Dordrecht, The Netherlands: Springer, 2005, pp. 167–180.

[117] M. Swerts, “Filled pauses as markers of discourse structure,” *Journal of Pragmatics*, vol. 30, pp. 485–496, 1998.

[118] L. M. Tomokiyo, “Recognizing non-native speech: Characterizing and adapting to non-native usage in LVCSR,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2001.

[119] E. Ukkonen, “Algorithms for approximate string matching,” *Information and Control*, vol. 64, pp. 100–118, 1985.

[120] C. J. van Rijsbergen, *Information Retrieval.* London, England: Butterworths, 1979.

[121] K. Vertanen, “Efficient computer interfaces using continuous gestures, language models, and speech,” Master’s thesis, University of Cambridge, Cambridge, England, 2004.

[122] E. Vidal, A. Marzal, and P. Aibar, “Fast computation of normalized edit distances,” *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 17, pp. 899–902, 1995.

[123] K. D. Voll, “A methodology of error detection - Improving speech recognition in radiology,” Ph.D. dissertation, School of Computing Science, Simon Fraser University, Burnaby, Canada, 2006.

[124] M. D. Wachter, M. Matton, K. Demuynck, P. Wambacq, and D. V. Compernolle, “Template-based continuous speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1377–1390, 2007.

[125] R. A. Wagner and M. J. Fischer, “The String-to-String Correction Problem,” *Journal of the ACM*, vol. 21, no. 1, pp. 168–173, 1974.

[126] J. Wei, “Markov Edit Distance,” *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 26, no. 3, pp. 311–321, March 2004.

[127] M.-B. Wesenick and A. Kipp, “Estimating the Quality of Phonetic Transcriptions and Segmentations of Speech Signals,” in *Proceedings of the Intl. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, Pennsylvania, October 1996, pp. 129–132.

[128] M. Wester, “Pronunciation modeling for ASR - Knowledge-based and data-derived methods,” *Computer Speech and Language*, vol. 17, pp. 69–85, 2003.

[129] S. M. Witt, “Use of speech recognition in computer-assisted language lerning,” Ph.D. dissertation, Cambridge University, Cambridge, England, 1999.

[130] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "ELAN: A professional framework for multimodality research," in *Proceedings of the Intl. Conf. on Language Resources and Evaluation (LREC)*, 2006, pp. 1556–1559. [Online]. Available: http://www.lat-mpi.eu/tools/elan/

[131] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.3)*. Cambridge, England: Cambridge University Engineering Department, April 2005.

[132] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 29, pp. 1091–1095, 2007.

[133] J. Zobel and P. W. Dart, "Phonetic string matching: Lessons from information retrieval," in *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, 1996, pp. 166–172.