Feature extraction and event detection for Automatic Speech Recognition

Kenmerkenextractie en eventdetectie voor Automatische Spraakherkenning

Frederik Stouten

Promotor: prof. dr. ir. Jean-Pierre Martens

Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. Jan Van Campenhout
Faculteit Ingenieurswetenschappen
Academiejaar 2007-2008

"It is the sheerest of coincidences"

(Isaac Asimov about the fact that the moon and the sun have the same size seen from earth, occasionally resulting in a total eclipse)

Abstract

This dissertation consists of two main parts. A first part describes work on *spontaneous* speech recognition, wile the second part focuses on the extraction of *phonological* features and their applicability for speech recognition.

Spontaneous speech

State-of-the art speech recognition systems typically work fine for well described tasks e.g. read speech presented under clean conditions (no noise or other distortions). However as soon as the speaking mode starts to deviate from these well prepared and well articulated conditions, the performance of the recognition system typically decreases significantly. Several fundamental reasons contribute to this fact. One of the important reasons is that unprepared or spontaneous speech is far more difficult to model. The modeling paradigms embedded in a speech recognition system seem no longer adequate for spontaneous speech. Spontaneous speech is characterized by severe reductions of syllables and word forms and by so-called disfluencies: points at which the sentence structure can be broken or interrupted

This dissertation will investigate solutions to the problem of disfluencies. It will turn out that many of the disfluencies are filled pauses and as such a better modeling of filled pauses could be a good step towards a better modeling of spontaneous speech.

The solution I propose is to implement an external detector of filled pauses that retrieves its information from features that are not necessarily available to the recognizer. Such a detector provides segmental probabilities for filled pauses and I have investigated how to use the external filled pause information in the recognizer. I propose two main strategies for doing this. The first strategy is to simply discard the segments which got a high filled pause probability. The second strategy consists of raising the language model probability of a 'uh'-hypothesis when it is generated in a time interval that shows a large overlap with the detected filled pause segment. Both strategies are compared to internally informed strategies advocated in the literature. In the latter strategies no external detector is involved but the language model probability is changed according to the occurrence of 'uh'-hypotheses in the word lattice produced by the recognizer. I found that my externally informed methods yield higher gains in word accuracy compared to the internally informed methods. Moreover, the small gain due to internally informed methods can be added to the gain of my method by simply combining the two strategies. The maximum attainable gain that could be reached with my methods is about 5.5% relative and is estimated to be half as large as the maximum gain that could have been obtained with an ideal filled pause detector that detects every possible filled pause with a probability of one.

Phonological features

In the second part of my dissertation the focus is shifted towards the investigation of using phonological features for speech recognition. One of the problems for the recognition of foreign names for instance is that they contain foreign phonemes that are not modeled by the baseline acoustic model set. The advantage of phonological features over classical acoustic features in such a cross-lingual situation is that these features can also model foreign phonemes. The reason for this is that phonological features are supposed to be language independent.

I first conducted a study of phonological features in general and I conceived a novel feature set and a hierarchical detector for extracting it from the speech signal. The detection accuracy was found to be comparable with that of related systems reported in the literature. On the basis of the phonological features I developed a phonetic segmentation and labeling tool that is intended to provide a segmental description of the speech, a description which can form a good basis for e.g. the assessment of the pronunciation proficiency of a speaker. By way of validation, I have used the tool to classify speakers into a native and a non-native class. From this validation experiment I got the confirmation that my aligner is as good as a much more complex aligner based on traditional triphone acoustic models. In the meantime, my aligner has been used with success for the automatic intelligibility assessment of pathological speakers (work done by a colleague).

My major goal was to investigate the usefulness of phonological features for speech recognition. First I note two important problems related to the use of these features: (1) the fact that they are locally correlated and (2) the fact that not all features are relevant for all phonemes. To find a solution to the first problem, I have fully elaborated a decorrelation technique that was initially proposed for speech recognition and speaker or environment adaptation on classical acoustic features. The choice for this method is motivated by the fact that the correlations between phonological features are dependent on the phoneme identity, and that it is therefore needed to utilize state dependent transformations to decorrelate the observations in such a state. In order to cope with the feature irrelevancy problem, I propose a novel scheme in which the emission distribution of each state is factorized into a relevant and an irrelevant part each working on the respective features. The findings were that the decorrelation technique helps a lot to improve the recognizer based on phonological features (26% relative improvement), but it also helps to improve the recognition based on acoustic features (14% relative improvement). The relevancy technique yields a small additional gain for the phonological based recognizer (8% relative improvement). However, I was not able to create a recognizer working with phonological features that could really compete with the best systems working with traditional features.

In the final part of my dissertation I have therefore thoroughly investigated the potential of combining phonological and acoustic features in the recognizer. Such a combination can be performed at several levels. I developed a word-level as well as a state-level combination. Word-level combination is performed on the word hypotheses encoded in two word graphs generated by the two systems. The technique I developed is based on the creation of a product graph, and on the rescoring of this graph. With this method I was unfortunately not able to improve on the best individual system. The state-level combination boils down to a two-stream approach in which the phonological features constitute a back-off stream. This approach did not allow me to improve the recognition of regular speech either, but it did offer a significant gain for the recognition of spoken person names and geographical names (22% relative improvement). This is owed to the fact that the morpho-syntax of names is strongly different from that of regular words. This means that the phonetic contexts are different on average and that the acoustic models do not always yield good estimates. The phonological stream then gets a good opportunity to take into account the phonological information that is not so much restricted to the abnormal context, as the information that is taken into account by a traditional model for this context.

Samenvatting

Deze verhandeling bestaat uit twee delen. Het eerste deel beschrijft werk i.v.m. *spontane* spraakherkenning, terwijl het tweede deel handelt over de extractie van *fonologische* kenmerken en hun toepassingen voor spraakherkenning.

Spontane spraak

Hedendaagse spraakherkenningssystemen werken goed voor welomschreven taken zoals voorgelezen spraak zonder noemenswaardige achtergrondruis noch andere distorties. Van zodra de spreekstijl minder voorbereid en goed gearticuleerd begint te worden, vermindert de nauwkeurigheid van de herkenner significant. Verschillende oorzaken kunnen hiervoor aangewezen worden. Eén van de belangrijke redenen is dat weinig voorbereide of spontane spraak veel moeilijker te modelleren is. De modelleringsparadigma's die in de spraakherkenner ingebed zijn, blijken dikwijls voor spontane spraak niet goed meer te werken. Spontane spraak wordt gekenmerkt door noemenswaardige verkortingen van lettergrepen en woordvormen en door zogenaamde haperingen: plaatsen waar de zinsstructuur afgebroken of onderbroken wordt.

Deze verhandeling zoekt naar oplossingen voor het probleem van de haperingen. Het zal duidelijk blijken dat veel van deze haperingen eigenlijk gevulde pauzes zijn en als zodanig kan een betere modellering van gevulde pauzes een eerste stap vormen tot een betere modellering van spontane spraak.

De oplossing die ik voorstel bestaat erin om een uitwendige detector voor gevulde pauzes in te schakelen die zijn nodige informatie betrekt uit kenmerken die niet noodzakelijk toegankelijk zijn voor de spraakherkenner. Zo'n detector genereert kansen dat welbepaalde segmenten gevulde pauzes zijn. Ik heb dan onderzocht hoe deze uitwendige gevulde pauze informatie kan gebruikt worden in de herkenner. Ik stel voor om daarbij gebruik te maken van twee verschillende strategieën. De eerste strategie laat de gevulde pauze segmenten waaraan een hoge kans werd toegekend gewoon weg. De tweede strategie zal de taalkundige kans van de 'uh'-hypothese verhogen indien deze optreedt in een tijdsinterval dat een grote overlap vertoont met de gedetecteerde gevulde pauze. Beide strategieën worden vergeleken met inwendig geïnformeerde strategieën die ook in de literatuur worden voorgesteld. Hierbij wordt geen gebruik gemaakt van een uitwendige detector, maar de taalkundige kans wordt nu afhankelijk gemaakt van het voorkomen van 'uh'-hypotheses in de woordgraaf gegenereerd door de herkenner. Ik kwam tot de vaststelling dat de uitwendig geïnformeerde methode tot een hogere verbetering in woordnauwkeurigheid leidde dan de inwendig geïnformeerde methode. De eerder kleine winst die bekomen wordt met de inwendige strategie kan eenvoudig toegevoegd worden aan de winst die bekomen werd met mijn methode door beide te combineren. Verder bleek de maximale winst die gehaald werd met mijn methodes ongeveer 5.5% relatief te zijn en gelijk te zijn aan de helft van de winst die zou kunnen gehaald worden met een ideale gevulde pauze detector die elke mogelijke gevulde pauze detecteert met een kans gelijk aan één.

Fonologische kenmerken

In het tweede gedeelte van mijn verhandeling wordt de focus verlegd en onderzoek ik de toepasbaarheid van fonologische kenmerken in een spraakherkenner. Eén van de problemen bij de herkenning van vreemde namen bv. is dat deze vreemde fonemen bevatten die niet behoren tot de modellenset van de herkenner. Het voordeel van fonologische kenmerken t.o.v. klassieke akoestische kenmerken in zo'n cross-linguale situatie is dat ze gebruikt kunnen worden om deze vreemde fonemen te modelleren. De reden hiervoor is dat de fonologische kenmerken taalonafhankelijk verondersteld worden.

Eerst heb ik fonologische kenmerken algemeen bestudeerd om vervolgens zowel een nieuwe kenmerkenset als een hiërarchische detector voor de extractie ervan, te ontwerpen. De detectienauwkeurigheid bleek vergelijkbaar te zijn met die die ik in de literatuur gevonden heb. Gebaseerd op deze fonologische kenmerken heb ik dan een systeem voor fonetische segmentatie en labeling gebouwd die bedoeld is om een segmentele beschrijving van de spraak op te leveren die dan een goede basis vormt voor bv. de beoordeling van de uitspraakkwaliteit van een spreker. Bij wijze van validatie heb ik deze tool gebruikt om sprekers in een native of een non-native klasse te klasseren. Dit validatie-experiment heeft uitgewezen dat mijn systeem even goed werkt als een veel complexer systeem dat gebaseerd is op een set van traditionele trifoon modellen. Ondertussen wordt mijn systeem met succes gebruikt voor de automatische beoordeling van de verstaanbaarheid van sprekers met een spraakgebrek (werk uitgevoerd door een collega).

Mijn belangrijkste doelstelling is om het nut van fonologische kenmerken voor spraakherkenning te onderzoeken. Eerst wijs ik op twee belangrijke tekortkomingen van de fonologische kenmerken, namelijk (1) het feit dat ze lokaal gecorreleerd zijn en (2) het feit dat niet alle kenmerken relevant zijn voor alle fonemen. Aan het eerste probleem kom ik tegemoet door middel van een decorrelatietechniek die ik overgenomen heb uit de literatuur waar ze oorspronkelijk werd voorgesteld voor spraakherkenning en spreker- of omgevingsaanpassing op akoestische kenmerken. De keuze voor deze techniek wordt mee bepaald door het feit dat de correlaties tussen de fonologische kenmerken sterk afhankelijk zijn van de foneemidentititeit, en dat het dus nuttig is om toestandsafhankelijke transformaties te gebruiken om de kenmerkenvectoren in elke toestand te decorreleren. Het probleem van het niet altijd relevant zijn van de kenmerken kan verholpen worden door de emissiefunctie in een welbepaalde toestand van de herkenner te factoriseren in een gedeelte dat inwerkt op de relevante kenmerken en een gedeelte dat enkel met de irrelevante kenmerken rekening houdt. De bevindingen zijn dat de decorrelatietechniek de nauwkeurigheid van de herkenner gebaseerd op fonologische kenmerken sterk doet toenemen (26% relatieve verbetering), maar dat de herkenning gebaseerd op akoestische kenmerken eveneens hierdoor verbetert (14% relatieve verbetering). De relevantie techniek leidt tot een eerder bescheiden extra verbetering voor de fonologische herkenner (8% relatieve verbetering). Ik ben er niet in geslaagd om een herkenner te bouwen die met fonologische kenmerken werkt en die het beter deed dan een systeem dat werkt met traditionele akoestische kenmerken.

Omwille daarvan heb ik in het laatste gedeelte van deze verhandeling de verschillende mogelijkheden onderzocht om fonologische en akoestische kenmerken in de herkenner te combineren. Zo'n combinatie kan op verschillende niveau's gebeuren. Ik koos ervoor om zowel een woordniveau als een toestandsniveau combinatie uit te testen. De woordniveau combinatie wordt uitgevoerd op de woordhypotheses die geëncodeerd zitten in de woordgrafen gegenereerd door de beide systemen. De techniek die ik ontwikkelde, maakt gebruik van een zogenaamde productgraaf, en een herscoring daarvan. Met deze methode kon ik het beste individuele systeem niet verbeteren. De toestandscombinatie daarentegen is in essentie een tweestromenmodel waarbij de tweede informatiestroom aangeduid wordt als een fonologische "back-off" stroom. Deze aanpak liet mij evenmin toe om de herkenning van gewone spraak te verbeteren, maar het leidde wel tot een significante verbetering voor de herkenning van gesproken persoons- en geografische namen (22% relatieve verbetering). Dit kan verklaard worden doordat de morfo-syntaxis van namen nogal sterk verschilt van die van gewone woorden. Dit betekent verder dat de fonetische contexten gemiddeld genomen afwijken en dat de akoestische modellen dus niet altijd goede schattingen zullen opleveren. Daardoor wordt de mogelijkheid gegeven aan de tweede fonologische stroom om de fonologische informatie te gebruiken die niet zo sterk gebonden is aan de context, daar waar dit wel het geval is bij de informatie die een traditioneel model zou gebruiken voor deze context.

Acknowledgments

I would like to express my sincere gratitude to several people without whom this work would not have been possible. First of all my thanks go to my thesis supervisor, Jean-Pierre Martens. He guided me on the sometimes difficult path that ultimately led to this dissertation. I felt myself privileged that I could work together with a person so energetic who never ceased to raise new possible research issues.

My next thanks go to my colleagues of the Speech Lab: Catherine Middag and Bert Réveil, who are currently still working on their project or Ph.D. I also would like to thank some former colleagues who found their way into the industry: Wim Goedertier for answering many technical questions related to computer systems and programming or scripting languages, Tom Demulder and An Vandecatseye, fellow students, who joined the Speech Lab together with me after graduating. Particular thanks go to Qian Yang, who took the time to answer many of my questions related to speech recognition systems. I also like to express thanks to the employees of TNI, a spin-off company, located near the lab. Particular thanks go to Jean Ryckebosch, Merijn Vandenabeele and Franky Maes for many enjoyable badminton hours.

During the course of my thesis I had the opportunity to work together with the people of ESAT at the university of Leuven. I would like to express my thanks to Jacques Duchateau for the fruitful collaboration and many discussions and to Kris Demuynck for many good advice and support.

My next thanks goes to Ronny Blomme, system manager at ELIS, for helping me each time my computer system was turning against me.

I also like to thank the members of the Ph.D. examining board: Christian Wellekens, Louis ten Bosch, Hugo Van hamme, Peter Lambert, Bert Van Coile, Sabine Wittevrongel and the dean, Daniël De Zutter. They

all kindly accepted to review my dissertation despite their heavy responsibilities and busy schedules.

This work has brought me to exotic places. Some of them I had never heard of before: The Virgin Islands, Jeju Island, Pittsburgh, Kyoto and Toulouse. I am grateful for having had the opportunity to attend the exciting scientific conferences that were held at those (even more) exciting places.

Last but certainly not least, I thank my family and friends for their kind support throughout all those years. I guess they sometimes felt closely connected with the ups and downs of scientific investigation. Now I am glad to say that it was worth the effort.

Contents

Nomenclature

ACF	Acoustic Feature
AM	Acoustic Model
ANN	Artificial Neural Network
ARF	Articulatory Feature
ARPA	Advanced Research Projects Agency
ARPABET	Phonetic alphabet developed by ARPA
ASR	Automatic Speech Recognition
BN	Broadcast News
BS	Baseline System
CALL	Computer Assisted Language Learning
CAPT	Computer Assisted Pronunciation Training
CD	Context-Dependent
CDF	Cumulative Distribution Function
CGN	(Dutch) Corpus Gesproken Nederlands
CI	Context-Independent
CMS	Cepstral Mean Subtraction
CSR	Continuous Speech Recognition
DT	Decision Tree
EBP	Error Back-Propagation
EBPT	Error Back-Propagation through Time
EM	Estimate-Maximize
ESAT	Electronica Systemen Automatisatie en Technologie
FIR	Finite Impuls Response
FP	Filled Pause
G2P	Grapheme to Phoneme converter

GMM	Gaussian Mixture Model
GOP	Goodness Of Pronunciation
HMM	Hidden Markov Model
IPA	International Phonetic Association
IWR	Isolated Word Recognition
LL	Log Likelihood
LM	Language Model
LMA	Language Model Adaptation
MFCCs	Mel Frequency Cepstral Coefficients
MIDA	Mutual Information Discriminant Analysis
MI	Mutual Information
ML	Maximum Likelihood
MLLT	Maximum Likelihood Linear Transform
MLE	Maximum Likelihood Estimation
MLP	Multi-Layer Perceptron
NFP	Non Filled Pause (segment)
OOV	Out-Of-Vocabulary
PCA	Principal Component Analysis
PDF	Probability Density Function
PHF	Phonological Feature
RNN	Recurrent Neural Network
RI	Relative Improvement of the WER
RMSE	Root Mean Squared Error
ROVER	Recognizer Output Voting Error Reduction
SAMPA	Speech Assessment Methods Phonemic Alphabet
SCM	Standard Context Manipulation
SD	Speaker Dependent
SI	Speaker Independent
SIM	Speaker Independent Model
SLF	HTK Standard Lattice Format
SPE	Sound Pattern of English
SR	Sentence Restart
SSR	Spontaneous Speech Recognition
SWB	Switchboard

TIMIT	Speech corpus for phonemic research designed by Texas Instruments (TI) and the Massachussetts Institute of Technology (MIT)
WER	Word Error Rate
WR	Word Repetition
WSJ	Wall Street Journal corpus

List of symbols

T	number of observations in one training utterance
$\alpha_t(i)$	forward probability to arrive at time t in state i starting at time 0 at state 0
a_{ij}	transition probability between state i and state j
$\beta_t(j)$	backward probability to reach state j at time t starting from state N at time T
$b_j(\mathbf{x})$	emission function in state j
$b_{jk}(\mathbf{x})$	k-th component of $b_j(\mathbf{x})$
c_{jk}	weight of the k-th mixture component of the GMM attached to state j
$c_0(t)$	log-energy (zero’th cepstral coefficient)
$d_{i,j}$	Euclidean distance between feature vectors at times i and j
D_{stab}	minimal mean distance between cepstral vector and neighbours
f_i	component i of the phonological feature vector
k_t	mixture component at time t
λ	total set of system parameters
$\overline{\lambda}$	new estimate of λ
μ_{jk}	mean vector of k-th mixture component of the GMM attached to state j
$p(\mathbf{x} \mid q)$	likelihood of $\mathbf{x}$ in state q
s_t	state at time t
t_{PPR}	Posterior Probability Ratio Threshold
t_{PP}	Posterior Probability Threshold
$\zeta_t(j,k)$	probability of being in state j and mixture component k at time t

A_p	transformation matrix for partition p
K	hidden mixture component random variable
$\mathrm{LL}(\mathbf{x}\|q)$	log likelihood in state q
M	total number of mixture components
N_{pm}	number of physical models
N_q	set of negative features attached to state q
$\mathcal{N}(\mathbf{x}, \mu, \Sigma)$	multivariate Gaussian PDF with mean vector μ and covariance matrix Σ
$P(q \mid \mathbf{x})$	posterior probability in state q
P_q	set of positive features attached to state q
$Q(\lambda, \overline{\lambda})$	auxiliary-function for the EM-algorithm calculated on the current estimate of λ and the new estimate $\overline{\lambda}$
$\tilde{S}(m)$	log signal power in the m^{th} sub-band of the Mel-scale filterbank
Σ_{jk}	covariance matrix of Gaussian PDF attached to state j, mixture component k
S	hidden state random variable - state space
S_p	p-th state partition
$\mathbf{W}$	sequence of words
$\mathbf{x}$	observation vector
$\mathbf{X}$	sequence of observation vectors

1
Introduction

This dissertation is about Automatic Speech Recognition, abbreviated as ASR. The ultimate goal of an ASR-system is to convert *natural speech* into a sequence of words.

1.1 Automatic Speech recognition

Speech recognition has advanced considerably since the first machines which could convert human speech into symbolic form (i.e. transcribe it) were conceived in the 1950s. Still, humans are much better than machines at deciphering speech under changing acoustic conditions, in unknown domains, and at describing somebody’s speech characteristics as "sloppy", "nasal" or similar, which allows them to rapidly adjust to a particular speaking style. This results in a human speech transcription performance unmatched by machines. The main reasons why machines are not able to match the human performance are the many sources of variability inherent to speech. With inter-speaker variability, the difference between speakers is indicated. Several speakers do not pronounce the same word in the same way. Factors like age, gender and voice timbre are contributing to a significant degree of inter-speaker variability. Intra-speaker variability means that one speaker does not always pronounce the same word in the same way. Speaking rate and speaking style, psychological conditions (e.g. stress) and lexical context are important causes of intra-speaker variation. Probably one of the most important sources of variability is the environment. The current speech recognition systems are much more sensitive to noise than human listeners are. Two types of noise can degrade the quality of the input speech signal. One is environmental noise (background noise), being defined as any sound from sources other than the target speaker. Street noise at a public telephone boot is a typical example. Another type of noise is the distortion caused by the channel over which the speech was recorded. This can be caused by the microphone transfer characteristic or by digital encoding or decoding errors, e.g. with GSMs. Environmental noise is supposed to be additive whereas channel noise is often convolutional in nature. Moreover, when environmental noise is clearly present, speakers will tend to raise their voice so as to be understandable. In such a situation speech has different spectral characteristics when compared to normal conditions (no background noise). This is called the Lombard effect.

In order to cope with all the mentioned speech variabilities the recognition task has to be constrained. The more constrained a task is, the easier it is for the recognizer. Constraints can take the form of a restricted vocabulary size, a low grammatical complexity or a restriction on the speaking style. There are two dimensions in the speaking style which are directly related to the task. One is the dimension distinguishing between isolated words and continuous speech, the other is the dimension distinguishing between read and spontaneous speech (also called conversational speech).

The ultimate goal of the research in ASR is to build a large vocabulary, speaker-independent, conversational speech recognition system that can work properly, even in noisy circumstances. Present day systems can only reach an acceptable accuracy if one or more of the mentioned constraints are imposed. A lot more applications would be possible if one were successful in deploying speech recognition systems without such constraints. In this dissertation the emphasis is on speaker-independent *continuous speech recognition (CSR)*, where the task is to convert a continuous speech signal into a sequence of words. The first part (chapters 3 and 4) describes work on spontaneous speech, whereas the second part (chapters 5 till 7) describes work on read speech. In the second part I will also concentrate on isolated word recognition (IWR), where the recognizer's output is restricted to a single word.

1.2 Definitions

Before discussing my work in more detail, I will now introduce some basic terms which are frequently used in the ASR research field.

Phone This is simply a sound that belongs to a language. Other sounds like laughter, coughing, etc. are not considered as phones. The notation for a phone will be between two slashes, like in /p/.

Phoneme The set of phonemes is the set of symbols that is needed to describe the pronunciations of all words in a language. Two symbols are phonemes if there exist two words whose pronunciation only differs in this symbol. The existence of *paard* and *baard* in Dutch, implies that /p/ and /b/ are two Dutch phonemes. A phoneme is thus the minimal information bearing distinctive unit. Most languages need 30-50 phonemes (e.g. English has 40 phonemes, Spanish only 24, see Appendix B for more information about language dependent phonemes). A phoneme will occur between slashes.

Allophone All the acoustic realizations (physical objects) of the same phoneme (symbolic unit) are called *allophones* of that phoneme. They all carry the same phonological meaning, but they may sound very distinct to the human ear. Especially the phonemic context (preceding and subsequent phonemes) can have a large impact on the way an allophone is perceived. Notice that two allophones in a certain language may be no allophones in another (sounds /l/ and /r/ are allophones in Chinese but not in English or Dutch).

Some allophones have a complex structure: they can appear as a sequence of two or three sub-phonemic parts with more homogeneous acoustic properties. These atoms of the acoustic realization of a phoneme are sometimes called *sub-phonemic units.* I will identify these units as synonyms of phones.

Grapheme The set of graphemes is the set of symbols that is needed to describe the spelling of all words in a language.

Orthographic transcription An orthographic transcription is a sequence of graphemes associated with a word sequence.

Phonemic transcription A phonemic transcription is a sequence of phonemes describing the pronunciation of a word sequence.

Phonetic transcription A phonetic transcription is the phone sequence describing the pronunciation of a word sequence.

Speech production refers to the complex process of articulation that is responsible for the generation of speech. Every phone is characterized by a typical configuration of the *vocal tract*, which is the physiological structure being responsible for the speech production. More on this in chapter 5.

Native A person who is speaking the language he has learned as a child (mother tongue) is called a *native* speaker of that language. In contrast, someone who expresses himself in another language than his mother tongue will be called a *non-native* speaker of that language.

Accent Nonnative speakers will almost always exhibit a certain accent. An *accent* can be defined as the ensemble of allophonic variations that are not commonly observed in the speech of native speakers. The age at which someone starts to learn a second language strongly influences the gravity of the accent he will have. An accent is something that is most likely being perceived differently according to the listener.

Articulators are all organs that can take part in the speech production, e.g. the lips, the tongue, . . .

Coarticulation is the phenomenon of neighboring phones affecting the acoustic properties of an examined phone. It is an important source of allophonic variation, but one that can be explained by the fact that articulators are changing continuously from one configuration to another. Preceding configurations affect the current articulatory configuration because of the inertia of the articulators (regressive coarticulation). This effect is most obvious at the onset of the phone. Upcoming articulatory configurations will also have an effect on the offset of the phone, because of anticipation (progressive coarticulation). Other sources of allophonic variation cannot be explained by a theory like this.

Spontaneous speech is an unprepared form of speech ranging from interviews, over talks to day-to-day conversations. This kind of speech poses serious problems for state-of-the-art recognition systems. Some of the main characteristics of spontaneous speech which are responsible for these problems can be summarized as follows:

1. Sloppy pronunciations
   Typically, words will be pronounced more swiftly, resulting in a shortening of the word. In highly spontaneous speech whole syllables can be deleted. A typical Dutch example could be the reduction of the word /natuurlijk/ to forms such as /'tuurlijk/, /'tuurl'k/ and even /'t'rl'k/.
2. Grammatically less strict sentences
   The sentence structure is far less evident in spontaneous speech than in read speech.
3. Disfluencies
   Spontaneous speech is also characterized by the occurrence

of disfluencies such as: repetitions of words or word groups, restarts, filled or unfilled pauses, repairs (see further).

We are still lacking good ASR methodologies for handling these problems well. No speech recognizer is able to reach a recognition level that is even remotely comparable with that of humans.

Cross-lingual A situation in which only one language is involved is called monolingual. If however a system is trained on one language, but tested on another then this is called a cross-lingual situation. Multilingual means that the system was trained on more than one language.

Word vs. word token In the course of this dissertation I will often speak of the total number of words or word tokens. The former means the number of *different* words, whereas the latter simply refers to *all* words that occurred in the text.

1.3 Structure of a CSR system

All modern CSR systems use some kind of pattern recognition paradigm to retrieve the most likely word sequence, given the acoustic input and some background knowledge of the recognition task (vocabulary, grammar, speaking style). Figure 1.1 shows the typical architecture of a CSR system. The input speech waveform $s(n)$ carries a lot of information

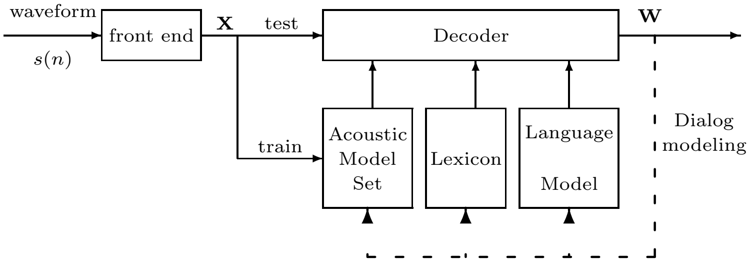


Fig. 1.1: General architecture of a CSR system [74]

that is redundant for the recognition process. Therefore, the recognizer first extracts from that waveform a sequence of acoustic features which represent the most important acoustic information carried by the signal. The feature extraction is performed by the so-called *front end*. The individual feature vectors are denoted as $\mathbf{x}_t$ with t representing a time index with each time unit corresponding to a multiple of 10 ms. This 10 ms is usually called the *frame shift* or *frame rate*.

The heart of the recognizer is the *pattern matcher*, usually called the *decoder*. Its objective is to search for the most likely word sequence $\mathbf{W}$, given the acoustic feature stream and given all available knowledge sources. The main knowledge sources are the *acoustic model set*, the *lexicon* and the *language model* (LM) (see section 1.4 for their definitions). The quality of a CSR system will to a large extent depend on the quality of its knowledge sources, with the acoustic model set possibly being the most critical one.

In modern CSR systems, acoustic models and language models are both based on a statistical analysis of acoustic and text data respectively. In other words, they are optimized with respect to objective criteria. The lexicon on the other hand is usually created on the basis of readily available resources (e.g. phonemic dictionaries).

In Figure 1.1 feedback loops from the recognized word string to all knowledge sources are depicted. This feedback symbolizes the adaptation of the knowledge sources to the task. For example when spontaneous speech has to be recognized, the user of the ASR-system may opt for an acoustic model set trained on spontaneous speech. Similarly when prior knowledge about the vocabulary says that it will be restricted to some words, the user can adapt the lexicon by leaving out all other words.

1.4 The knowledge sources of a CSR system

In this section, I provide a bit more information on the content and the purpose of the different knowledge sources:

Acoustic model set The acoustic models are statistical models capturing the acoustic variation in the acoustic realizations of a phoneme. More in particular, each model must be be able to determine how likely it is that a sequence of acoustic feature vectors is an instantiation of an allophone of a particular phoneme. It is possible to work with one model per phoneme but it is also possible to construct several models for each phoneme, one model for each context in which a special model is needed. In the former case one talks about *context-independent* (CI) phoneme models or *monophones*, in the latter case about *context-dependent* (CD) phoneme models.

Lexicon The lexicon comprises all the words that can be hypothesized by the recognizer[1]. This word set is also called the *vocabulary*. In addition, the lexicon describes how the production of each word can be considered as a sequence of phonemes. Such a phoneme sequence is called a *phonemic transcription* or a *pronunciation* of the word. If more than one pronunciation is given for a certain word, these pronunciations are called *pronunciation variants*.

Language model The language model is intended to assign a priori probabilities to word sequences that are investigated as potential solutions by the decoder. By properly integrating the language model probabilities in the decoding process one can dramatically reduce the number of hypotheses to explore. This leads to a large speed-up of the decoding process, as well as to a significant reduction of the number of recognition errors being made.

In order to attain a CSR system with an acceptable performance, all the knowledge sources must be properly optimized for the envisaged task.

1.5 The basic equations of CSR

In this section I briefly introduce the basic equation of CSR with the intention to give the reader a first idea of how the decoder operates and how the knowledge sources fit into the decoding process.

Assuming static knowledge sources (this means that all sources are fixed before the recognition starts), the task of the decoding process can formally be described as a search for the word sequence that maximizes the *a posteriori probability* $P(\mathbf{W}|\mathbf{X})$ of $\mathbf{W}$ given the sequence $\mathbf{X}$ of acoustic feature vectors and given the knowledge captured in the knowledge sources. By applying Bayes' law, it follows that the system is searching for

$$\hat{\mathbf{W}} = \operatorname*{argmax}_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) = \operatorname*{argmax}_{\mathbf{W}} \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})} \tag{1.1}$$

Since $P(\mathbf{X})$ does not depend on the selected word sequence, the above equation can be simplified to

$$\hat{\mathbf{W}} = \operatorname*{argmax}_{\mathbf{W}} P(\mathbf{X}|\mathbf{W})P(\mathbf{W}) \tag{1.2}$$

[1] Words that do not occur in the language model cannot be recognized even if they are in the lexicon.

If **F** represents an arbitrary phonemic sequence, one can finally rewrite this equation as

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} \sum_{\mathbf{F}} P(\mathbf{X}|\mathbf{F})\ P(\mathbf{F}|\mathbf{W})\ P(\mathbf{W}) \tag{1.3}$$

The first factor in the right hand side of expression (1.3) is computed by means of the acoustic model set. The second factor follows from pronunciation knowledge encoded in the lexicon. The third factor is computed on the basis of the language model. More details on how to compute these factors on the basis of the knowledge sources will be given in chapter 2.

1.6 Acoustic modeling

Much of the popularity of the current ASR approach is due to the existence of an algorithm for the automatic training of acoustic models, usually Hidden Markov Models or HMMs. More on HMMs in chapter 2. The training uses an orthographically or phonemically transcribed speech corpus. An efficient *Maximum Likelihood* (ML) criterion [77] is commonly adopted, giving rise to the so-called EM-algorithm to train the model parameters. EM stands for Expectation Maximization and offers an elegant solution to the training problem.

1.7 Pronunciation modeling

A lexicon is basically a pronunciation dictionary containing the pronunciations of the words of the vocabulary. In most CSR systems the lexicon comprises one pronunciation per word, called the *typical* or *canonical* pronunciation of this word. Such a lexicon is called a *baseline lexicon.* Usually, the pronunciations presented in the baseline lexicon originate from phonological knowledge (electronic pronunciation dictionaries like CELEX [16] and grapheme-to-phoneme modules [4] developed in the context of speech synthesis), but often they have been manually checked by the lexicon designer. However, there exist pronunciation modeling methods aiming at improving the recognition accuracy by automatically discovering and introducing in the lexicon, alternative pronunciations of the words [125].

1.8 Language modeling

The language model (LM) must provide the probabilities $P(\mathbf{W})$ in equation (1.3). These probabilities are independent of the acoustic observations and describe the lexical constraints which are revealed by constraints in word ordering of the language and to a lesser extent of the task and domain. Since it is impossible to model all probabilities $P(\mathbf{W})$, because this would require too much parameters to be trained, speech scientists have proposed to use N-grams. N-grams are sequences of N words

$$\{w_{i-N+1}, \ldots, w_i\} \tag{1.4}$$

for which conditional probabilities

$$P(w_i \mid w_{i-N+1}, \ldots, w_{i-1}) = P(w_i \mid w_{i-N+1}^{i-1}) \tag{1.5}$$

are estimated on the basis of text data by simply counting the number of times a word sequence appears in the training data. For (1.5) this becomes,

$$\hat{P}(w_i \mid w_{i-N+1}^{i-1}) = \frac{\text{Count}(w_{i-N+1}, \ldots, w_i)}{\text{Count}(w_{i-N+1}, \ldots, w_{i-1})} \tag{1.6}$$

However, in practice no text corpus is large enough to yield reliable estimates for all possible N-grams. Should one apply (1.6), many estimated probabilities would be based on few or no examples.

A typical solution to this problem is to *smooth* the N-gram probability distributions by discounting a small portion of the total probability mass for a certain context w_{i-N+1}^{i-1} to unseen or rarely seen events. One of the best known discounting strategies is called back-off [59]. Back-off uses the most complex LM if it offers a reliable probability estimate for the requested event, otherwise a lower order model is used instead. A typical value for N is three, in which case one speaks of a *trigram* LM.

One often speaks of the *perplexity* of a language model. This is a measure of the mean uncertainty about the next word given the N-1 previous words, according to the LM. The perplexity is a function of the entropy of the LM. Sometimes the perplexity of the test set is given. This is the same definition but only measured as a mean over the words occurring in the test set.

1.9 Topics covered in this dissertation

The first part of the dissertation is concerned with the recognition of spontaneous speech and in particular with the development of a method for coping with the disfluencies occurring in this speech. This research was also chronologically the first I carried out. I succeeded in building a disfluency detector that is able to detect filled pause segments in running spontaneous speech and to integrate the filled pause (FP) hypotheses successfully in the decoding process: an improvement of the recognition accuracy is obtained. The integration is accomplished in two ways. The first strategy boils down to a discarding of the filled pause segments during the decoding. The second strategy aims to make it easier for the decoder to hypothesize filled pauses (as semi-words) during detected pause segments. Classical approaches that are situated at the level of the language model were tried as well, and were compared to the results obtained with the FP detector information.

Since my research suggested that it would be difficult to obtain significant additional improvements by pursuing a special treatment of other disfluency types, and since the research project in which I performed my research had terminated, I had to contemplate a reorientation of my research. At the time of this reorientation the DSSP group (in which I work) was engaged in a project that aimed at using ASR technology in medical and educational applications. In these applications one has to deal with disordered, pathological speakers that cannot properly articulate the sounds, as well as children and non-natives that differ in their articulation from native adult speakers. In particular the speech of pathological speakers may be difficult to describe in terms of the modal phonemes of the language.

Therefore, it was decided to investigate whether a better characterization would be possible by using a phonologically inspired symbolic representation as an alternative to the phonemically based representation adopted by all commercial recognizers.

The full exploitation of this new representation turned out to necessitate an adaptation of the traditional schemes for the training and utilization of the new acoustic models.

The phonologically based methodology I developed has proven to be useful for the objective characterization of the intelligibility of disordered speech [81] as well as for the automatic recognition of foreign names by native speakers.

The new representation did not allow me however to improve the ASR of common speech by native speakers. I have hopes however, that it can be helpful for the ASR of speech of non-native speakers, but this is a subject of future research.

1.10 Main contributions of this dissertation

The main contributions of my work can be summarized as follows.

Disfluency detection I was able to develop a detector for filled pauses. It uses a discriminative pattern classifier and segmental speech features, and it is able to detect filled pauses in spontaneous speech with a very reasonable accuracy. The development and evaluation of the detector is described in [100].

Spontaneous speech recognition with disfluency information I supplied the output of the filled pause detector to the decoder of the recognizer. I have investigated two strategies: one consists of discarding the detector's output segments during the decoding, the other consists of favoring the hypotheses of disfluencies in segments overlapping with the detected filled pause segments. These strategies are reported in [102; 101; 99].

Phonologically inspired speech features I proposed a new speech feature set with a phonological interpretation. In order to extract these features I conceived a novel discriminative pattern classifier. This work is described in [103].

ASR with phonological features I performed recognition tests on representative benchmark tasks with acoustic models that make use of the new features. My contributions here lie in the proposed adaptations to the model training scheme, and in the search for strategies to combine the new and the traditional features in one acoustic model set. This led to publication [104]. I also applied the new features with success in a spoken name recognition task. In such a task one is confronted with names comprising foreign phonemes that do not exist in the native language. These phonemes are thus not covered by the native phoneme models. It was expected that phonological features could offer benefits for the description of the foreign phonemes. The results of this work are described in [106; 105].

1.11 Outline of this dissertation

The rest of this dissertation is structured as follows.

Chapter 2 defines and explains all the basic technical definitions and tools that I used throughout my research. This chapter also provides an overview of the speech databases I have used for training and evaluation.

Chapter 3 gives an overview of the kind of disfluencies that can occur in spontaneous speech and it also introduces a disfluency syntax model.

Chapter 4 provides the details about the filled pause detector I conceived. In particular I motivate the choice for segmental speech features and I discuss the detection results that were obtained. Then the methods for applying disfluency information during spontaneous speech recognition are being described.

Chapter 5 introduces concepts from phonetics and phonology. In this chapter I first introduce the interesting new speech features which will turn out to represent phonological events. Then I discuss the choice of the feature set and the application of neural networks to detect the features I am interested in.

Chapter 6 outlines an exploratory study that is intended to demonstrate the capability of the features to separate native and non-native speakers of American English. In order to do so, a segmentation and labeling method is explained and results are compared with the literature.

All algorithms and experiments concerning ASR with phonological features are described in chapter 7. More in particular, the adaptations of the training algorithm, the combination of classical and new features in the search and the use of phonological features to deal with cross-lingual phenomena as they appear frequently in spoken names.

Finally this dissertation ends with the main conclusions of my work and with some suggestions for further research directions.

2
Tools and Databases

This chapter explains all the technical tools which are needed in the following chapters. First I explain the standard feature extraction and the HMM framework that is used by the baseline recognizers. The appropriate notation will also be introduced here. Then follows a description of an alternative way to store the output hypotheses of a recognizer and an overview of important concepts about neural networks. Finally I describe all speech databases that were used for benchmarking. This description is followed by a discussion on the evaluation metric. Finally the results of the baseline systems on the benchmarks are presented.

2.1 Acoustic feature extraction

The function of the front end is to convert the speech signal into a parametric representation that effectively and efficiently represents the information that is needed by the recognizer. Considering the front end as a black box, its input is the sampled speech waveform and its output is a stream of parameter vectors, also called acoustic feature vectors, acoustic observation vectors, or briefly, acoustic features (ACFs) or observations.

The most popular acoustic features, in order of appearance, are the Mel-frequency cepstral coefficients (MFCCs) [24], the perceptually linear prediction (PLP) coefficients [52] and the linear prediction coefficients (LPCs) [5; 73]. The MFCCs are obtained as the Discrete Cosine Transform (DCT) of some kind of log-energy spectrum that is presumed to emerge from the spectral analysis taking place in the human ear. MFCCs were first introduced for speech recognition in 1980 by Davis and Mermelstein [24] and have since then become by far the most widely used acoustic features in CSR. Consequently, I also decided to use MFCCs as

ACFs in all my experiments.

The basic mechanisms involved in the transformation of a speech waveform into a sequence of MFCC vectors are:

- **Framing.** Each feature vector is computed on the basis of a speech fragment of $25 \cdots 35$ ms long, centered around the time of interest. Such a speech fragment is called a *speech frame* or just a *frame*. As the speech is usually sampled at a rate of 8 or more kiloHertz (kHz), a frame counts at least 200 speech samples.

- **Sampling.** Two subsequent acoustic vectors are usually computed on two frames which are shifted in time over an amount of 10 ms. This time shift determines the so-called *frame rate*. Since it is shorter than the length of a speech frame, it means that subsequent frames overlap in time. The overlap between analysis windows is one of the causes for correlations between subsequent frames.

- **Data reduction.** Each speech frame is first of all represented by a set of D features, with D usually being in the range of $12 \cdots 16$. Then the computed features are usually augmented with their first and second order time-derivatives. The *dimension* of the acoustic feature vector is then equal to $3D$ which is still, for the given D-range, a lot smaller than the number of speech samples in a speech frame.

In the experiments I conducted, the front end generated 12 MFCCs and a log-energy feature per frame. This means that $D = 13$ and that the dimension of the feature vector is equal to 39.

2.2 Hidden Markov Models

In this section I briefly introduce the HMM technique for acoustic modeling. For a more elaborate discussion of the matter the reader is referred to [88].

Architecture

An HMM in this dissertation is a finite state machine that models an acoustic feature vector sequence as being generated by a stochastic Markov process [54]. Usually the HMM of a phoneme has a 3-state, left-to-right, no-skip topology as illustrated on Figure 2.1. Each shaded circle represents a state and each arc represents a possible transition between states. The HMM can be viewed as an acoustic vector generator: every 10 ms

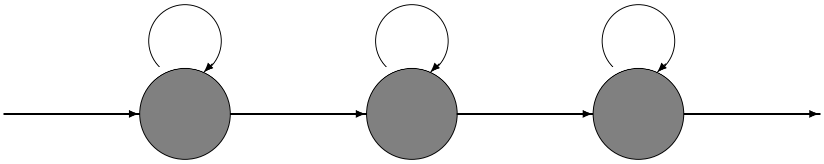


Fig. 2.1: General topology of an HMM

the active state is allowed to change and an acoustic vector is emitted in the new active state. The HMM is governed by two stochastic processes: (i) a transition process represented by *transition probabilities* a_{ij} on the arcs between states i and j, and (ii) an emission process characterized by *emission probability density functions* $b_j(\mathbf{x})$ associated with the states j.

Probability computation

On the basis of the two stochastic processes one can compute the probability that a specified acoustic vector sequence $\mathbf{X}$ of length T is generated along the state sequence S (of the same length). This probability is given by

$$P(\mathbf{X}, S|\lambda) = P(S|\lambda) \cdot P(\mathbf{X}|S, \lambda) \tag{2.1}$$

with λ representing the HMM (or its free parameters if you wish), with

$$P(S|\lambda) = \pi_{s_o} \prod_{t=1}^{T} a_{s_{t-1},s_t} \tag{2.2}$$

$$P(\mathbf{X}|S, \lambda) = \prod_{t=1}^{T} b_{s_t}(\mathbf{x}_t) \tag{2.3}$$

and with π_{s_o} representing the a priori probability of state s_o being the active state at time $t = 0$.

Equation (2.2) implies that only the previous state influences the probability of being in the current state. This is called the first order Markov hypothesis. Equation (2.3) implies that subsequent vectors emitted in the same state are presumed independent of each other. It is acknowledged that this so-called *independency assumption* is in disagreement with the nature of speech. Nevertheless, HMMs appear to lead to good recognition results provided they include the first and second order time derivatives of the individual frame features. In fact, by introducing these derivatives one can model some temporal correlation effects.

Due to the fact that the emission distributions of individual states can overlap in the feature space, the same acoustic vector can be emitted in different states. This means that the state sequence is not observable, or put in another way, remains *hidden*. The probability that an acoustic vector sequence can be generated by the model is then equal to the sum of the probabilities $P(\mathbf{X}, S|\lambda)$ over all legal state sequences through the model

$$P(\mathbf{X}|\lambda) = \sum_{S} P(\mathbf{X}|S, \lambda) \cdot P(S|\lambda) \tag{2.4}$$

Given the HMMs of the phonemes, the HMM of a phoneme sequence $/P/$ corresponding to a complete speech utterance can be obtained by simply concatenating the HMMs of the phonemes appearing in $/P/$. This yields a model with many states, but the computation of probabilities can still be achieved by means of Equations (2.1)-(2.4).

Best state sequence

In CSR, it seems to be important to have a means of computing the best state sequence, given the acoustic vectors and the phoneme sequence. This sequence is defined as

$$\hat{S} = \underset{S}{\operatorname{argmax}}\, P(\mathbf{X}, S|/P/, \lambda) \tag{2.5}$$

and it can be computed by means of the Viterbi algorithm [118]. An important remark is that this algorithm does not require prior information about the locations of the phonemes. In fact, the best sequence computed by the algorithm also reveals the best phonemic segmentation of the utterance. One often says that HMMs perform a segmentation by classification.

Model training

Obviously, the HMMs of the different phonemes do not fall out of the sky. Before they can do a proper job, they need to get proper free parameters: the transition probabilities and the free parameters of the functions that model the emission probabilities. In most cases, the continuous probability density function $b_j(\mathbf{x})$ is realized as a mixture of multivariate Gaussian components (GMM = Gaussian Mixture Model) with diagonal covariance matrices $\Sigma_{jk} = \text{diag}(\sigma_{jk})$.

$$b_j(\mathbf{x}) = \sum_{k=1}^{M} c_{jk} \cdot \mathcal{N}_{jk}(\mathbf{x}, \mu_{\mathbf{jk}}, \mathbf{\Sigma_{jk}}) \tag{2.6}$$

$$\mathcal{N}_{jk}(\mathbf{x}, \mu_{\mathbf{jk}}, \mathbf{\Sigma_{jk}}) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{jkd}^2}}\, e^{-\frac{(x_d-\mu_{jkd})^2}{2\sigma_{jkd}^2}} \tag{2.7}$$

The parameters of this model are the mixture weights (c_{jk}) which are positive and adding up to 1, and the means and variances (one per acoustic feature) characterizing the Gaussian mixture components. The total number of mixture components is M.

The major advantage of the HMM technology is that there exists a maximum likelihood (ML) framework for automatically retrieving good parameters from a large number of so-called *training utterances* of which the phonemic transcriptions are known. These utterances are said to constitute the *training corpus*. It can be considered as one long training utterance represented by a long acoustic feature vector sequence $\mathbf{X}$ and a corresponding long phonetic transcription $/P/$. The free parameters of all the models together (represented by λ) are then estimated in such a way that

$$\hat{\lambda} = \underset{\lambda}{\text{argmax}}\, P(\mathbf{X}, S \,|\, /P/, \lambda) \tag{2.8}$$

The training algorithm that automatically determines the free parameters is the iterative Estimate-Maximize (EM) algorithm.

One can prove (see [58] for the derivation) that the obtained reestimation formulae for μ_{jk} and Σ_{jk} take the following form:

$$\overline{\mu}_{jk} = \frac{\sum_{t=1}^{T} \zeta_t(j,k)\mathbf{x}_t}{\sum_{t=1}^{T} \zeta_t(j,k)} \tag{2.9}$$

$$\overline{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \zeta_t(j,k)(\mathbf{x}_t - \overline{\mu}_{jk})(\mathbf{x}_t - \overline{\mu}_{jk})^t}{\sum_{t=1}^{T} \zeta_t(j,k)} \tag{2.10}$$

with $\zeta_t(j,k)$ being defined as $p(s_t = j, k_t = k|\mathbf{X}, \lambda)$, the probability of being in state j and mixture component k at time t.

Context-independent (CI) and context-dependent (CD) models

A first important distinction between acoustic models is that of CI versus CD models. In the case of CI models, every phoneme is modeled independently of its phonemic context. It is clear however that phoneme realizations can be affected by the phonemic context, and consequently that it may be difficult to model them as monophones because of the large amount of acoustic variation that will have to be modeled then. By introducing different models for the same phoneme, e.g., one model per phonemic context in which the phoneme can occur, one can circumvent this problem.

Suppose that the phoneme sequence of which one needs to compute the acoustic probability is /m E n/. In the monophone case, one would compute that by using the three acoustic models created for /m/, /E/ and /n/ respectively. In the CD model case, one will create a model for /m/ in the context of a silence (/#/) to the left and an /E/ to the right and refer to this model by the notation /#-m+E/. In the same sense one will create models like /m-E+n/ and /E-n+#/ and use these models to compute the envisaged acoustic probability. Phoneme models conditioned on one left and one right phonemic symbol are usually called *triphone models* or *triphones*.

Clearly, there are dramatically more triphones than monophones in a language. This will have consequences for the decoding time and for the development of these models, as will be explained further on.

Parameter tying

It is clear that after training, an acoustic model is only expected to have reliable parameters if there were enough examples of the modeled phoneme in the training corpus. When using triphone models, this can become a bit of a problem since there can be 30,000 or more different triphones in a language, and thus, chances are high that not all of these triphones actually appear in sufficient numbers in the training corpus. As a consequence, a parameter tying scheme is needed to account for unseen (and hardly seen) triphones in an appropriate way.

The goal of tying is to learn the emission probability distributions of frequently occurring states only, and to use these distributions to compute the emission probabilities of less frequently occurring states. A popular scheme for achieving a good collection of emission distributions is *state tying* on the basis of *decision tree* (DT) clustering. For every state (1, 2 or 3) of a phoneme, one creates a tree. The root of the tree represents the emission distribution built on all the phoneme occurrences (as if the phoneme model were a monophone model), the other nodes represent emission distributions built on occurrences of the phoneme in more and more specific contexts.

2.3 Word lattices

The output of a recognizer can consist of just the most likely hypothesis, or the N-best list of the most likely N hypotheses, or a word lattice (graph) representing a whole lot of hypotheses. A lattice (Figure 2.2) is a directed acyclic graph composed of nodes and arcs.

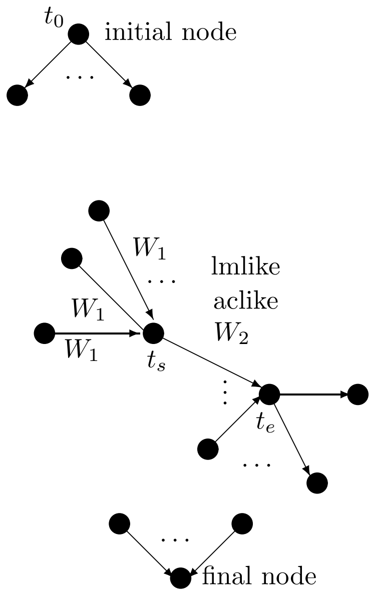


Fig. 2.2: Topology of a word lattice in HTK format

Node A node is a state in the lattice comparable with a state in the recognizer. A time stamp will be associated with each node in the lattice.

Arc An arc is a transition between two nodes. This transition defines the arc's start node (time stamp t_s) and end node (time stamp t_e). This makes the lattice directed. An arc can be labeled and things like scores which were generated during the decoding process by the recognizer (see further) can also be attached to an arc. A node with no incoming arcs, is called an initial node, whereas a node with no outgoing arcs is called a final node of the lattice. A lattice can have only one initial node and one final node.

Best hypothesis For retrieving the best hypothesis from a lattice, I only need to find the best path connecting the initial node with the final node. This can be performed by means of a Viterbi algorithm. Note however that the best hypothesis found in the word lattice *does not* necessarily correspond with the recognition output that would have been returned by the recognizer if it was asked to generate the best hypothesis only. The reason for this is that the search was constricted by a pruning strategy. This means that not all possible paths in the word graph were explored by the search engine. A Viterbi algorithm without pruning will find the optimal path, which I will call best hypothesis.

When the recognizer is operated in a CSR mode, it will generate lattices with word labels on the arcs. For a recognizer performing phoneme recognition the labels will be phonemes.

In my work I have been using two distinct recognizers: the HTK recognizer [127] and the ESAT recognizer [31], each capable of producing word graphs. Unfortunately, both recognizers use different types of lattices whose differences will be pointed out here. In HTK the arcs are carrying the following information: (i) the acoustic likelihood (aclike), (ii) the LM probability (lmlike), (iii) the pronunciation variant of the word, (iv) the start node number and the end node number. Nodes are given time stamps as well. The ESAT lattice format differs from the HTK format in that the LM information is not compiled in the lattice. So, the lattice only contains acoustic likelihoods attached to the transitions which were made in the recognizer during the search.

2.4 Multi Layer Perceptrons

During my research I have been using Multi Layer Perceptrons (MLP) quite often. They represent a special class of Artificial Neural Networks (ANN). The structure of a MLP is represented in Figure 2.3.

A MLP is composed of computing nodes which are distributed over a *hidden layer* of hidden nodes (n_h) and an *output layer* of output nodes (n_o). Each input can be connected to each hidden node by means of an arc. The output of each hidden node can be connected to each output node. A weight w is attached to each arc. A MLP in this dissertation is a pattern classifier which is trained to produce one high output referring to the class the input vector belongs to. The free parameters of a MLP are:

1. The number of hidden nodes (#HNodes)

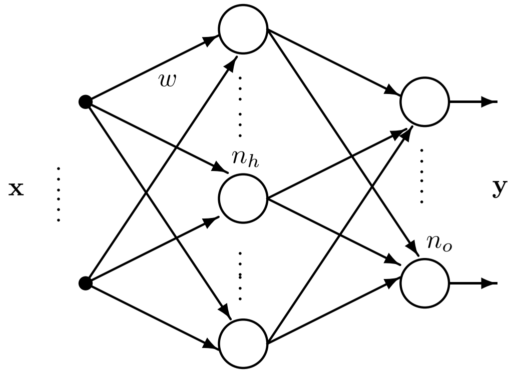


Fig. 2.3: Topology of a Multi Layer Perceptron (MLP)

2. The interconnection scheme between layers
   This means that not all input nodes have to be connected to all hidden nodes (and the same for the interconnections between the hidden and the output nodes).

3. The weights on the arcs

The estimation of the weights can be done by means of a training algorithm, denoted as Error Back-Propagation (EBP,[90]). Training requires training patterns $\mathbf{x}_i$ and corresponding learning outputs (targets) $\mathbf{t}_i$. The dimension of $\mathbf{t}_i$ is equal to the number of output nodes. The target vector has only one non-zero value which is 1 and which points to the class the input vector belongs to. During training the weights are updated in such a way that the mean square error

$$E = \sum_i \sum_k [y_{ik} - t_{ik}]^2 \tag{2.11}$$

is minimized (i running over all training patterns and k running over all output layer components). The EBP algorithm uses gradient descent to update the weights. The training of the weights proceeds in several passes which are called *epochs*. After each epoch the error function can be evaluated on the training data, or on a separate data set called the *validation set*. Training is stopped as soon as the error on the validation set reaches a plateau. Evaluation on a validation set is preferred since it will produce a MLP that is trained until it generalizes best to unseen test data.

The result of the EBP training is a set of weight estimates. It can be shown [55] that under favorable conditions (enough training material) the trained MLP will, during operation, produce outputs $y_k = P(C_k|\mathbf{x})$ $k = 1 \ldots, K$. I.e. it computes the posterior probabilities of the classes C_k.

MLPs can be used for all kinds of classification problems, in particular ASR [69; 112], which can also be considered as a multi-class classification problem. MLPs have been successfully applied to phoneme recognition [67], in combination with HMM-based recognition systems [14] and for the classification of sub-phonemic features on which I will come back in detail in later chapters.

2.5 Speech Databases

A speech database is a collection of labeled training and test utterances. The training utterances can be used to train acoustic models, whereas the test utterances can be used to evaluate these models. By performing benchmark experiments on internationally available databases, it is possible to compare the performances of systems across institutions.

2.5.1 TIMIT, a phonetically rich corpus

The DARPA TIMIT speech database [38] was originally designed for the development of acoustic models for phonetic research and was collected at Texas Instruments (TI) and MIT. It consists of utterances of 630 speakers that represent the major dialects of American English. Each speaker reads 10 sentences: 2 SA, 5 SX and 3 SI sentences. The same SA sentences were read by all speakers, and were meant to expose the dialectal variations between speakers. The SX sentences are phonetically compact sentences, designed to provide good coverage of phoneme-pairs in the language. The SI sentences are phonetically diverse sentences, designed to add diversity in phonemic contexts.

In my experiments the data was divided in a training set of 462 speakers and a test set of 168 speakers by the designers of the corpus. A small portion of the test set is known as the *core test set* and it consists of $24 \times 8 = 192$ sentences of 24 speakers. For the training $462 \times 8 = 3696$ training utterances (1124823 frames) are available. The number of words in the core test set is 1570.

It is important to remark that the TIMIT utterances come with an orthographic and a phonetic transcription. The latter is a manually verified sequence of phones (sub-phonemic units) with their start and end time.

The TIMIT lexicon comprises all the 6231 different words appearing in the training and test sentences. One canonical pronunciation is provided per word. Homonyms get a different transcription. The pronunciations are given in terms of the 61 TIMIT phones (including silence and pause).

If one wants to use TIMIT for ASR assessment, one needs a LM. Such a model is not provided with the data, but there exists a back-off bigram LM [126] that was trained from the SX and SI sentences and that was used for benchmarking purposes already [126; 64]. The perplexity of this LM is 89.3

2.5.2 Wall Street Journal (WSJ)

The WSJ CSR Corpus [85] described here is DARPA's first general-purpose English, large vocabulary, natural language, high perplexity, corpus containing significant quantities of both speech data (400 hrs) and text data (47M words). It primarily consists of read excerpts from Wall Street Journal articles. The corpus also contains some spontaneous dictation utterances. WSJ is composed of two parts commonly known as WSJ0 and WSJ1. The WSJ0 utterances are divided into a training (84 speakers), a development (10 speakers) and an evaluation (8 speakers) sub-corpus, and there is no speaker-overlap between the three subcorpora. In WSJ1 there are 200, 70 and 30 speakers in the training, development and evaluation set. To give you an idea of the size of the corpus, I mention that the WSJ1 training set represents about 73 hours of speech.

The texts to be read were selected according to the words they contained. There was a set to fall within either a 5k-word or a 20k-word subset of the WSJ text corpus. For these texts there were no Out-Of-Vocabulary words (OOV-words). No pronunciations of the words were provided. I got these pronunciations from the 129482 words CMU (Carnegie Mellon University) pronunciation dictionary [20]. The CMU lexicon contains on average about 1.08 transcriptions per word. The WSJ corpus is delivered with a number of vocabularies and their corresponding bigram and trigram back-off language models. I have performed experiments with the so-called 5k closed no verbal punctuation vocabulary and its corresponding bigram LM. The perplexity of this LM is 35.4

I trained my models on the complete training set of 284 speakers (37516 utterances, 29324850 frames) extracted from WSJ0 en WSJ1.

Throughout my work I have used three test sets:

1. The November 1992 ARPA CSR speaker-independent 5k vocabulary read no verbal punctuation benchmark test set taken from WSJ0. It contains speech of 8 speakers: 330 sentences and 5353 words.

2. The H2 subset containing speech from 10 native speakers: 215 utterances and 4064 words.

3. The S3 subset containing speech of 10 non-native speakers: 416 utterances and 7851 words.

2.5.3 AUTONOMATA Spoken Name Corpus

In chapter 7, I will perform ASR evaluations on a corpus of spoken names. This corpus was extracted from the AUTONOMATA Spoken name corpus. It is a corpus of proper names, recorded in the Netherlands and Flanders [116]. Every speaker has read 181 names from a list of 1810 names of Dutch, French and Moroccan origin. The speakers had different language backgrounds as well: Dutch, French, Moroccan and English. For my experiments, I have selected the recordings of 60 speakers with a Flemish background (mother tongue), and constructed test and development sets for the English (1000 and 380 word tokens), French (1000 and 380 word tokens) and Dutch (2000 and 760 word tokens) words uttered by these speakers. The test set comprised 4000 word tokens with an equal amount of native and foreign names: 2000 tokens were Dutch, 1000 were English and another 1000 were French. The development set followed the same lines. This set will be used for fine tuning of the free system parameters. The remaining 4440 Dutch word tokens will be used for adapting the acoustic models to the recording environment that characterizes the AUTONOMATA data. Since there are only 60 speakers, I decided to let them occur in all subsets because I wanted to use as many different speakers as possible for the test. The names however did not overlap in the different subsets. The division in subsets is summarized in Table 2.1.

2.5.4 Spoken Dutch Corpus (CGN)

The CGN (Ned: Corpus Gesproken Nederlands, CGN, [46; 75]) was collected between 1998 and 2004 at several Dutch and Flemish universities. The CGN project aimed at constructing a database of contemporary standard Dutch as spoken by adults in the Netherlands and Flanders.

	English	French	Dutch	total
adaptation	–	–	4440	4440
development	380	380	760	1520
test	1000	1000	2000	4000
total	1380	1380	7200	–

Tab. 2.1: Number of word tokens in each subset extracted from the AUTONOMATA corpus.

The corpus contains about 1000 hrs. of speech, two thirds originating from the Netherlands and one third from Flanders. The CGN is composed of 15 components each representing different communication settings and acoustic background conditions. The components are summarized in Table 2.2.

The components I used were a, b, f, g, i, j, k, l, m and n. I excluded components c and d because they contained telephone speech. Finally I did not use data from component o because my goal was to compose a training and test set of spontaneous speech only.

The files I selected represent about 12 hours of recordings containing spontaneous speech of 130 Flemish speakers. All together these recordings contain nearly 130k word tokens. Orthographic transcripts and manually verified word-level segmentations are available for all this material. The corpus excerpt is very diverse: it comprises spontaneous interviews, broadcast field reports, political debates and panel discussions. For reasons that will become clear later the corpus was further divided into a *training corpus* (91 files and about 11h of speech and 112k word tokens) and a *test corpus* (16 files and about 1h of speech and 7496 tokens). I have made sure that the 27 speakers appearing in the test corpus do not appear in the training corpus. The exact composition of the training corpus and the test corpus is revealed in Appendix A.

The CGN lexicon consists of the 40k most frequent words appearing in Dutch newspaper material. This lexicon was provided by ESAT. The LM is trained on newspaper material (33.3M word tokens) extended with 3M word tokens of spontaneous speech transcripts from the CGN. This was also done by ESAT. All OOV-words were listed in an extension file and given the same unigram probability. This was done to ensure that the measured differences were due to the applied techniques and not due to the OOV-words.

CGN components	
a	Spontaneous conversations ("face-to-face")
b	Interviews with teachers of Dutch
c	Spontaneous telephone dialogues (recorded via a switchboard)
d	Spontaneous telephone dialogues (recorded on MD via local interface)
e	Simulated business negotiations
f	Interviews/discussions/debates (broadcast)
g	(political) Discussions/debates/meetings (non-broadcast)
h	Lessons recorded in the classroom
i	Live (e.g. sports) commentaries (broadcast)
j	News reports/surveys (broadcast)
k	News (broadcast)
l	Commentaries/columns/reviews (broadcast)
m	Ceremonious speeches/sermons
n	Lectures/seminars
o	Read speech

Tab. 2.2: Components distinguished in the Spoken Dutch Corpus.

2.5.5 Switchboard (SWB)

SWB [45] is a large multi-speaker corpus of conversational speech and text which was intended to support research in speaker authentication and large vocabulary speech recognition. About 2500 conversations by 500 speakers from around the U.S. were collected automatically over T1 lines at Texas Instruments. Designed for training and testing of a variety of speech processing algorithms, especially in speaker verification, it has more than 1 hour of speech from each of 50 speakers, and several minutes each of hundreds of others. A time-aligned (at the word level) orthographic transcription accompanies each recording.

A Good-Turing [42] smoothed trigram language model was built on the basis of the 3M words in the Switchboard-1 conversation transcripts. The SWB lexicon consisted of all the 27k words appearing in the Switchboard-1 training data.

tool	database	WER (%)
HTK	TIMIT	4.59%
[127]	WSJ	6.05%
ESAT	CGN	36.1%
[25; 31]	SWB	29.8%

Tab. 2.3: Baseline systems.

A part of the 2001 HUB5 benchmark set was chosen as the test set, more particularly the part that corresponds to Switchboard-1. This set is composed of recordings of 20 informal telephone conversations (= 40 speakers, 1718 sentences, 20k word tokens) in American English. The recording time is about 2 hours. Orthographic transcripts of the material can be found at: *ftp://jaguar.ncsl.nist.gov/lvcsr/mar2001*.

2.6 Baseline Systems

During my research I experimented with several baseline ASR-systems that were built for the speech databases described above. Their performances serve as references against which I can compare my own results. The evaluation metric that is used for measuring this performance is the Word Error Rate (WER). This is defined as the amount of insertions, deletions and substitutions obtained after an alignment of the recognized word string with the reference word string, divided by the number of word tokens

The baseline ASR-systems that I considered are using either the HTK or the ESAT recognition engine. The accompanying acoustic model set, lexicon and LM is explained in the previous section. Table 2.3 gives an overview of the WERs of my baseline systems.

3 Disfluencies in Spontaneous Speech

Nowadays read speech recognition is already working pretty well, but the recognition of spontaneous speech is much more problematic. There are plenty of reasons for this, and I hypothesize that one of them is the regular occurrence of disfluencies in spontaneous speech. Disfluencies disrupt the normal course of the sentence and when for instance word interruptions are concerned, they also give rise to word-like speech elements which have no representation in the lexicon of the recognizer.

In this chapter I present some background information concerning the prevalence of different disfluency types in existing spontaneous speech corpora and I discuss how these disfluencies can affect the recognition of spontaneous speech. Further on, I will discuss the novel techniques I propose to remedy these problems. I make a distinction between disfluencies and regular words and I will express the prevalence of a particular disfluency type by means of its *disfluency rate*. The latter is defined as the number of disfluencies of that type divided by the total number of word tokens.

3.1 Disfluencies in spontaneous speech

In order to gain some insight in the language dependency of disfluency prevalences, I have performed measurements on two distinct corpora.

1. The statistical analysis presented here was performed on the CGN training corpus defined in chapter 2. The CGN test corpus (also defined in chapter 2) will be used to assess the effects of my disfluency handling methods on the speech recognition performance.

2. The second corpus is the 2001 HUB5 benchmark set from the

Switchboard corpus (see chapter 2). The experiments on this corpus were not carried out by myself, but by Jacques Duchateau, a colleague at KULeuven.

I have investigated three types of disfluencies, namely Filled Pauses (FPs), Word Repetitions (WRs) and Sentence Restarts (SRs). According to [94], these three types represent about 85% of all the disfluencies occurring in the Switchboard corpus. In the following sections I briefly describe the measurements I made and the results they produced.

3.1.1 Filled Pauses (FPs)

It is commonly acknowledged that FPs are the most frequently occurring disfluencies in spontaneous speech. A filled pause usually appears as an interjection, like “uh" or “uhm", but the acoustic properties of the interjections may be language dependent. Consider two examples which have been extracted from the CGN and translated to English:

1. oh I read all the *uh* books of Simenon but I *uh*.

2. *uh* particularly *uhm* Dutch literature *uhm*.

They illustrate that fillers can occur at many positions in the utterance.

Counting the number of FPs in the two corpora was rather easy because both the CGN and the Switchboard orthographic transcription protocols instructed the transcribers to stick to a restricted list of fillers to encode an FP.

The mean FP rate in the CGN data set was equal to 2.7%. However, a number of speakers had an FP rate of more than 10%. In the Switchboard dataset the measured FP rate was equal to 3%. This rate is much larger than the 1.7% reported by [94] for another subset of the Switchboard corpus.

Of all 596 FPs observed in Switchboard, 106 were sentence initial, 111 occurred at the end of a sentence, and 46 others were actually isolated sentences, meaning that they contained no other words. These figures imply that more than 50% of the FPs (333 in total) were sentence medial FPs. In the CGN data-set on the other hand, I found that 387 of the 445 FPs were sentence medial. This discrepancy may originate from the language or from the different types of speech material appearing in the two corpora.

3.1.2 Word repetitions (WRs)

One strategy for a speaker to gain some time to think is to repeat a word once or several times before continuing with the rest of the sentence. If this happens, the last word of the repeated word sequence is considered as the regular word whereas the others are designated as disfluencies. The following two examples were selected from the CGN and translated to English (the repeated word sequence is put in italic):

1. *I I* also work *with with* music. I work *I I I* mean dance music.
2. well *what what* are children supposed to do?

Counting word repetitions in a corpus is seemingly very simple: search for repetitions of the same word and count the number of disfluencies they represent. In practice, it is much more complicated than that. For instance, if an interjection occurs between identical words, there is still a word repetition involved. This complication is easy to accommodate. A more problematic complication is the presence in most languages, including English and Dutch, of grammatically correct word repetitions. Consider the following two English examples:

1. I think *that that* man at the station was drunk.
2. I still have to read *many many* articles on this topic.

None of the highlighted word repetitions should be counted as a disfluency, but I only know this because I understand the meaning of these sentences. Since it is currently impossible to make a simple and reliable semantic parser of spontaneous speech, I have chosen for a semi-automatic procedure to count the WR type disfluencies. In a first pass, I create a list of all sentences containing the following events: two identical words in a row, or two identical words separated by a filled pause. These events are then marked by a human expert as word repetitions or grammatically correct word sequences.

The WR-rate in the CGN data set turned out to be about 0.9%. In the Switchboard set it was about 1.4% and equal to the percentage that was also reported by [94] for another data set. In the Dutch data, there were less than 0.07% word repetitions consisting of three or more words. In the American data, this figure was much larger and close to 0.4%. Both datasets show only very few FP interjections between repeated words. This suggests that WRs and FPs are two clearly distinct speaker strategies for gaining time.

Since for the CGN data I also had manually verified word-level segmentations at my disposal, and since these segmentations also reveal inter-word pauses of more than 100 ms long, I was able to determine how many times repeated words are being separated by a (silent) pause. This happened in about 40% of the cases, meaning that pauses can possibly be considered as indicators of WRs in the recognition process. Another interesting finding was that the top-20 of repeated words in the CGN are all monosyllabic *function* words like "en", "een", "dat" .... Moreover, this top-20 can explain 78% of all the WRs encountered in the corpus. Figure 3.1 shows the word labels of the top-28.

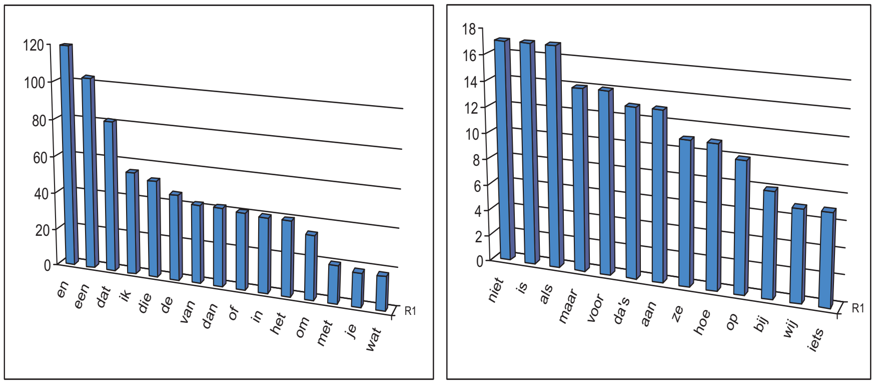


Fig. 3.1: The 28 most repeated words in the CGN training corpus.

3.1.3 Sentence Restarts (SRs)

A sentence restart (SR) is defined here as a situation in which the speaker makes the initial part of a started sentence obsolete by the succeeding words. The following examples represent instances of such SRs that were found in the CGN and translated to English (the obsolete part, also called the reparandum, is put in italics):

1. *is uh* did Agalev abandon you?
2. *in a situation with uh* in a country with two speed levels.

Obviously, sentence restarts cannot be retrieved automatically from the orthographic transcripts, unless they are explicitly annotated as such in these transcripts. Since no such annotations were available neither for CGN nor for Switchboard, I had to retrieve the SR-rate by means of a manual inspection of the transcripts. I have only performed this on the Switchboard data set. For this set I found 112 restarts corresponding to a SR-rate of about 0.5%. This rate was obtained as the number of sentence restarts (one per SR) divided by the total number of words. I also found that about 30% of the reparanda (33 instances, see next section for definition of reparandum) ended on a filled pause. Recalling that there were only 333 sentence medial FPs, this means that about 10% of all these FPs actually initiate a sentence restart.

3.2 Syntactic model

WRs, FPs and SRs can all be considered as special cases of a *generic* disfluency syntax model. Such a model discerns three subsequent parts:

1. **reparandum**
   This is the sentence part that has to be repaired, because it was wrong or because the speaker started a new sentence.

2. **interruption point**
   This is the point at which the speaker resumes the sentence by correcting the reparandum or by starting a new sentence. The interruption point may be marked by a filled or unfilled pause.

3. **reparans**
   This is the sentence part following the interruption point that is actually repairing the reparandum.

FPs can be markers for the interruption point, but they can also occur in situations where the reparandum (and reparans) is empty and the speaker is just hesitating. For WRs the reparandum is equal to the reparans. A SR is characterized by a reparandum that stretches until the beginning of the sentence.

Note that this scheme can be applied in a nested way, in which the reparans is itself a new reparandum. Multiple word repetitions are special cases of such a nested model.

3.3 Main effects on the automatic recognition process

One effect of an FP is that it introduces a new word (denoted as "uh") that is normally not included in the lexicon of a read speech recognizer. Obviously this effect can easily be accounted for by adding this word to the lexicon. One option is to add it with a pronunciation “uh" representing a dedicated whole-word speech unit whose acoustic model is trained on FP utterances. Another option is to add it with all its likely pronunciations in terms of regular phonemes of the language.

An effect that is common to all types of disfluencies is that they disrupt the normal word flow. This disruption implies that the spoken word sequence no longer matches well with a language model (LM) that was retrieved from text material not containing any disfluencies. Consequently, the LM probability of the correct word sequence may comprise a number of low back-off[1] probabilities, and the decoder may therefore be inclined to select a wrong but more likely sequence by assigning the FP interval to a short function word (e.g. “a", “the", etc.) that is acoustically similar to the FP, or to a syllable of a content word, a syllable that acoustically sounds like a FP (e.g. “**a**gree"). In both cases the decoder will produce wrong word hypotheses which will on their turn affect the word prediction capability of the LM in the vicinity of the disfluency. Consequently, it can be anticipated that one disfluency may be responsible for more than one error in the recognition output. In [3] the authors report a figure of about 1.5 errors per disfluency for a French spontaneous corpus. If this figure would generalize to my speech data it would mean that a disfluency rate of 3 to 5% could be responsible for a WER contribution of 4.5 to 7.5% absolute. In chapter 4 I will describe an experiment to assess the expected number of errors per filled pause on the Spoken Dutch corpus material. I will also find a figure of about 1.5 word errors per FP.

[1] A back-off LM probability is calculated on the basis of lower order N-grams.

4 Spontaneous speech recognition

In this chapter I explain all techniques that are elaborated in order to cope with the problem of disfluencies in spontaneous speech.

4.1 Introduction

The automatic recognition of spontaneous speech is currently a hot topic. Practical applications of spontaneous speech recognition include voice operated telephone services, automatic closed captioning for TV programs, automatic transcription of meetings, etc. Yet, the recognition accuracy of freely spoken language is still quite poor when compared to that of dictated speech: while the state-of-the-art word error rates (WER) for large vocabulary speaker-independent dictation and broadcast news transcription are of the order of 5% [107] and 15% [13; 44] respectively, the WER for the transcription of meetings [128] can be as large as 40%.

One important reason for this deficiency of spontaneous speech recognizers is the lack of a good language model built on a large amount of spontaneous speech transcripts. While typical stochastic language models for read speech recognition rely on vast amounts of training material [1], no comparable amounts of written transcripts of casual language are available.

On top of that, the occurrences of disfluencies in casual speech may further complicate the estimation of a robust spontaneous language model. In the literature different approaches to spontaneous language modeling have already been pursued. In [71], one tries to incorporate knowledge from discourse theory, and one argues that sentences typically start with given information and end with new information, and disfluencies mostly occur in the given information part of the sentence. By applying *special-*

ized language models for the two sentence parts, one obtains a marginal drop of the WER (0.3% absolute) on the recognition of spontaneous telephone conversations from a telephone conversation corpus (see [45] for more information on this corpus).

In [129] the potential of N-best list re-scoring on the basis of information from a chunk parser has been explored. The underlying assumption is that the chunker bears information that can help to discriminate between syntactically acceptable and syntactically anomalous hypotheses. It was demonstrated that this technique yields a marginal drop of the WER (0.3% absolute) on telephone conversations.

Even if one has a language model comprising context-dependent probabilities for disfluencies, it may be beneficial to remove the disfluency from the context when predicting *regular words* (as opposed to disfluencies) occurring right after that disfluency. Stolcke et al [98] have already investigated this technique, but their experiments on telephone conversations did not show any significant performance gain. I will propose a more flexible manipulation of the prediction context. In that approach, regular words will be predicted with and without the disfluencies being removed from the context. These predictions will then be allowed to compete with each other.

On the acoustical-lexical front, the literature also describes several solutions to disfluency handling in recognition systems. For instance in [91] a data-driven lexical modeling technique was applied to construct a lexical model with many pronunciation variants for a filled pause (FP). By substituting the baseline single-pronunciation FP model by this new more complex model, a 2% absolute (= 7.8% relative) reduction of the WER on a highly spontaneous medical transcription task was achieved.

A draw-back of the previously proposed techniques for disfluency handling is their assumption that the decoder part of the recognizer is capable of producing reliable disfluency hypotheses. I argue that in particular for filled pauses this assumption is often wrong. This means that the decoder will either hypothesize too few disfluencies due to a bad pronunciation model or maybe too many because the pronunciation model is too general. In fact, I will demonstrate that by introducing a specialized FP detector operating independently of the decoder, and by supplying the output of that detector to the decoder, it is possible to achieve a more significant improvement of the recognition accuracy with only a very limited increase of the computational load.

4.2 Novel methods for disfluency handling

In this chapter I propose a number of so-called *internally informed* and *externally informed* search strategies for coping with disfluencies in spontaneous speech recognition.

In an internally informed search strategy, the acoustic models, the lexicon and the LM are all together responsible for hypothesizing disfluencies, and the search engine must be adapted to undertake special actions when such hypotheses are generated.

An externally informed search strategy uses an external disfluency detector to spot the disfluencies and the decoder is adapted to take these hypotheses into account. I argue that such a strategy has a lot of potential in the case of FPs. First of all it is anticipated that FPs have some well-defined acoustic and prosodic properties [10; 87; 41; 82]. Secondly, the acoustic models of a speech recognizer are usually blind for prosody, and as such they may be unable to make a clear distinction between a FP and a function word like *a* (English) or *de* (Dutch), or between a FP and the initial syllable of a content word like *above* (English) or *getal* (Dutch).

In what follows three internally informed search strategies for dealing with FPs, WRs and SRs are proposed, as well as two novel externally informed search strategies for coping with FPs.

4.3 Internally informed search strategies

As already stated in the introduction, one of the hypotheses for explaining the difficulty of modeling spontaneous language by means of N-grams points explicitly to disfluencies: as N-grams base their word prediction on a local context of N-1 previous words, intervening disfluencies render this context less uniform. Or put differently, the prediction of the next word would be more accurate if it were based on a context from which the disfluency was removed. Obviously, removing disfluencies (one or more in a row) also implies that the word context to take into account in the decoder is extended to the left with regular words appearing in front of these disfluencies.

Consider for instance the example “this is what *uhm* I think", and presume that the LM is a trigram model. I argue that in this case the word “I" would be better predicted by the context “is what" than by the context “what uhm". Nevertheless, Stolcke and Shriberg [98] came to the surprising conclusion that discarding FPs from the trigram context actually increases the perplexity[1]. However, they were looking at speech stretches that were isolated on the basis of acoustic criteria (the presence of large silent pauses), meaning that the FPs occurring at sentence boundaries often appeared in the middle of such a stretch. By only discarding true sentence internal FPs, the perplexity did decrease indeed. In the material of [86] the speech stretches all corresponded to sentences and therefore all FPs were sentence internal. For this material, the discarding strategy resulted in a 4% decrease of the overall perplexity and a 30% decrease of the perplexity of the first word after the FP.

In [96] and [95], it is nevertheless shown that in some cases FPs *are* good predictors for the following words: they often tend to precede a less frequently used word. Therefore, simply discarding the FPs from the context is perhaps not always the best solution. This conclusion also holds for repeated words which are part of a grammatically correct word sequence, like in the example “I hope that that work is at least done properly now".

In order to account for the above observations ESAT, my partner in the ATRANOS project [7] in which I performed this research, proposed some novel context manipulation methods. These novel methods provide multiple options, but leave it to the recognition system to select the most likely option after having exploited all its knowledge (acoustic, lexical and linguistic). I briefly describe the three models: one model that must be applied in case a word repetition is hypothesized, and two models that must be applied in case a filled pause is hypothesized and I will use it later for comparison.

4.3.1 The repetition model

The model for handling word repetitions is sketched in Figure 4.1. It presumes that the LM is a trigram model. The upper path illustrates the normal LM procedure. If the hypothesized word B appears to be a repetition of the previous word, then the prediction of the next word C is based on the context B B. The lower path represents the alternative of predicting the word C on the basis of A B, the context which is obtained by simply ignoring the repeated word B.

4.3.2 The hesitation model

The hesitation model is activated in case a filled pause is detected. The model is depicted in Figure 4.2 with “uh" denoting an FP. The model

[1] The perplexity is a measure for the unpredictability of a word by means of the LM

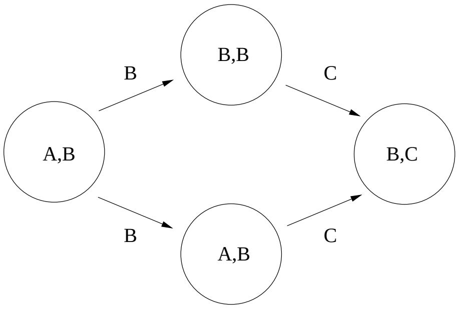


Fig. 4.1: The model for repetitions.

proposes two alternatives to the search engine: (1) the standard solution (upper path) in which the filled pause is kept in the context for predicting the subsequent words, and (2) an alternative solution (lower path) in which the filled pause is removed from that context.

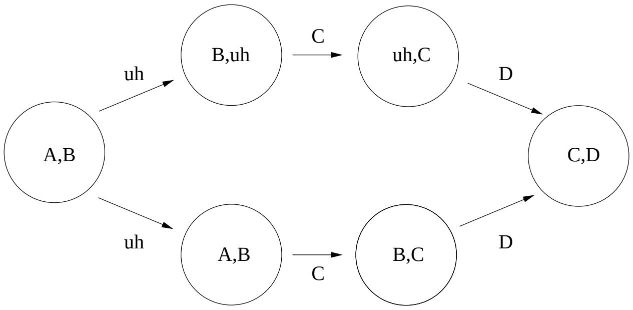


Fig. 4.2: The model for hesitations.

4.3.3 The restart model

Starting from the observation that many filled pauses announce a sentence restart, ESAT has conceived a restart model (Figure 4.3) that is activated every time a FP is hypothesized by the decoder. The lower path models the fact that a FP causes a reset of the language model: the left context is reset to the sentence start symbol <S>. It is clear that a sentence restart can also occur after a regular word, but a pilot experiment described in [32] demonstrated that activating the restart model for each word hypothesis causes an over-generation of restart hypotheses, and has a negative impact on the recognition accuracy.

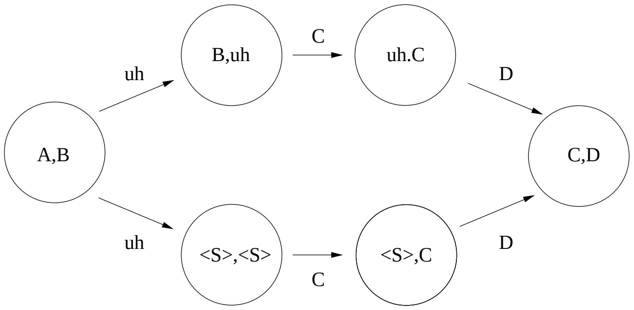


Fig. 4.3: The model for sentence restarts.

If successful, the hesitation and the restart model can obviously be combined into one more complex model to be applied whenever an “uh" is hypothesized.

4.4 Externally informed strategies for coping with FPs

Let me now propose two externally informed strategies for coping with filled pauses. They rely on the outputs of an external FP detector which works independently of the decoder part of the recognizer (see section 4.5). The developed FP detector produces variable length FP segments, and each segment comes with an associated posterior probability $P(\mathrm{FP}|\mathbf{X})$. The symbol $\mathbf{X}$ stands for the acoustic observations in and around the hypothesized FP segment and the posterior probability is hereafter called the FP score.

4.4.1 Frame dropping

If the FP score of a segment is high, one can expect that all the frames of that segment are FP frames. The idea of frame dropping is to let the decoder discard these frames. An advantage of the technique is that it can easily be integrated in any speech recognition system. All it takes is not to supply the FP frames to the decoder. Another advantage is that it can also be applied in combination with a decoder that does not even incorporate an FP model in its lexicon.

Obviously, frame dropping is a pretty drastic method which is bound to deteriorate the recognition performance if too many regular speech frames would be discarded. For instance, it can happen that the speaker starts by saying the word "the" and continues by prolonging the word, more or less gradually shifting to a filled pause. The correct handling of such a filled pause is problematic, because the FP detector is bound to indicate all the vocalic frames of the fragment as constituting an FP, and discarding all these frames may prevent the decoder from hypothesizing the word "the" as it should. Therefore, I expect that frame dropping will work best in combination with an FP detector that does not often produce large FP scores for non-FP (NFP) segments. This means an FP detector with a high precision.

4.4.2 Language model adaptation

Because of the potential danger of frame dropping I have also conceived a second strategy which is less categorical in its interpretation of the FP character of the frames. In this so-called LM adaptation (LMA) strategy, the normal LM probability of a word hypothesized in a time interval (t_1, t_2) which overlaps with an FP segment emerging from the external FP detector, will be replaced by a new probability that depends on (1) the identity of the word hypothesis, (2) the distance between t_1 and the FP segment start, (3) the fraction of (t_1, t_2) falling into this FP segment (the overlap fraction), and (4) the value of the FP score that was computed for this FP segment.

The LM adaptation procedure is activated every time a word hypothesis is generated. It works as follows (see also Figure 4.4):

1. If the hypothesized word starts at a time t_1 which is further than some threshold D away from the start of an FP segment detected by the external FP detector, then leave the LM probability unaltered, else continue with the next step.

2. If the overlap fraction between the hypothesized word and the FP segment is smaller than 50%, then take no special action either, else continue with the next step.

3. If the hypothesized word is "uh", then replace the normal LM score by a predefined value C_1.
   If the hypothesized word is not "uh", then subtract some predefined amount C_2 from the normal LM score which is log $P(\text{word}|\text{left context})$.

By manipulating C_1, C_2 and D it is possible to control the impact the FP detector can have on the recognition output. An alternative for the

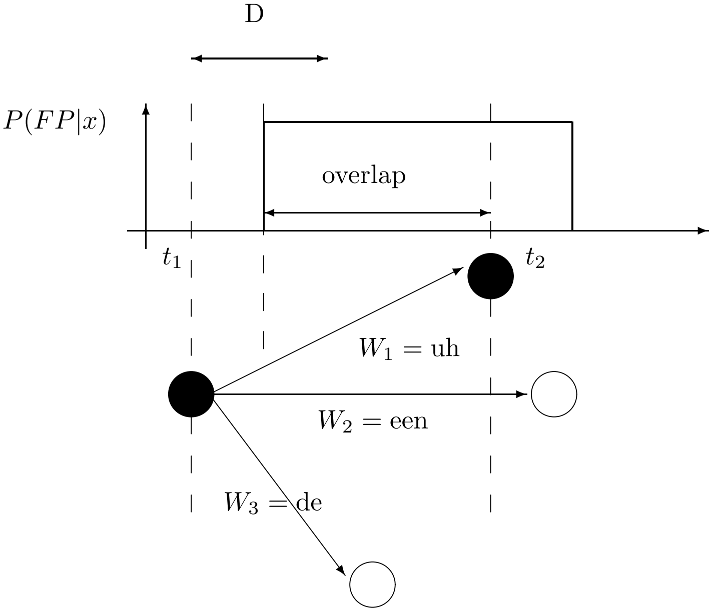


Fig. 4.4: LM adaptation is examined for word hypotheses (bottom) which exhibit some overlap with FP segments (top) produced by the FP detector.

probability substitution outlined in point (3) would be to replace log $P($"uh"$|\text{context})$ by the logarithm of the FP score emerging from the FP detector. I did test this approach, but it did not outperform the simpler and easier to control strategy outlined in point (3).

It is clear that the LM adaptation strategy uses all the available knowledge sources to make a distinction between the true and false FP segments proposed by the FP detector. Therefore LM adaptation could well be able to deal with false alarms emerging from the external FP detector. Consequently, I expect the technique to be most effective if it is applied on almost all the FP segments appearing in the speech. This means, when it is combined with an FP detector with a high recall.

For practical reasons I could not embed this LM adaptation in the recognition engine of the ESAT-recognizer. So I had to apply LM adaptation to the word graphs emerging from the recognizer. These word graphs are described in section 2.3.

4.5 An independent detector of filled pauses

I argue that the externally informed methods for coping with FPs will have an advantage over the internally informed strategies if they can rely on an FP detector that can spot the FP segments with a much higher accuracy than the decoder of the speech recognizer would be able to do. Therefore, I have conceived an FP detector that will not only base its decisions on the MFCC vectors which are used by the recognizer, but also on additional acoustic and prosodic cues that are not available to the acoustic models of the decoder [100]. The proposed detector first performs a blind segmentation of the speech into silent and phoneme-like segments. Then it classifies the non-silent segments as FP or NFP segments. I conjecture that the segmental framework facilitates the introduction of prosodic cues related to pitch and duration in the classification process. Such cues are invisible to the acoustic models embedded in the speech recognizer.

I will now describe the segmentation of the speech, the extraction of appropriate features to represent the segments and the classification of these segments into FP and NFP on the basis of these features. The feature selection and the training and evaluation of the FP detector are all achieved on the basis of the previously described CGN training and test corpora. I do believe however that the methods and results reviewed here are also relevant for the construction of e.g. an American English FP detector.

4.5.1 Speech segmentation

In order to create a segmental description of the speech, I first construct a feature change pattern, and I then hypothesize potential segment boundaries at the locations of the maxima in this pattern.

Starting from the standard vectors $\mathbf{x}_t$ extracted by the front-end of the recognizer (MFCC vectors with $M = 12$ components in my case), I derive first a feature change pattern d_t from

$$d_t^2 = \sum_{i=1}^{M} [\frac{\sum_{j=1}^{N_w} j\ [\mathbf{x}_{t+j}(i) - \mathbf{x}_{t-j}(i)]}{2\sum_{j=1}^{N_w} j^2}]^2 \tag{4.1}$$

Each term between the outer square brackets represents the norm of the slope of the best linear regression of the evolution of an individual feature in a window of $2N_w + 1$ frames centered around the time of interest t. The pattern d_t is then further smoothed by means of a three point FIR filter to

$$d_t^{'} = \frac{1}{4}d_{t-1} + \frac{1}{2}d_t + \frac{1}{4}d_{t+1} \tag{4.2}$$

and from this pattern the segment boundaries are derived. In order to do so, a robust left-to-right minimax algorithm [119] tracks the locations of prominent maxima in $d_t^{'}$. A prominent maximum is defined as one which is considerably higher than the largest of the two minima surrounding this maximum. A silence detector is also integrated in the segmentation algorithm. It detects intervals of at least three successive frames having a log-energy that is not more than 3 dB above an adaptive log-energy noise floor, computed on the basis of minimum statistics.

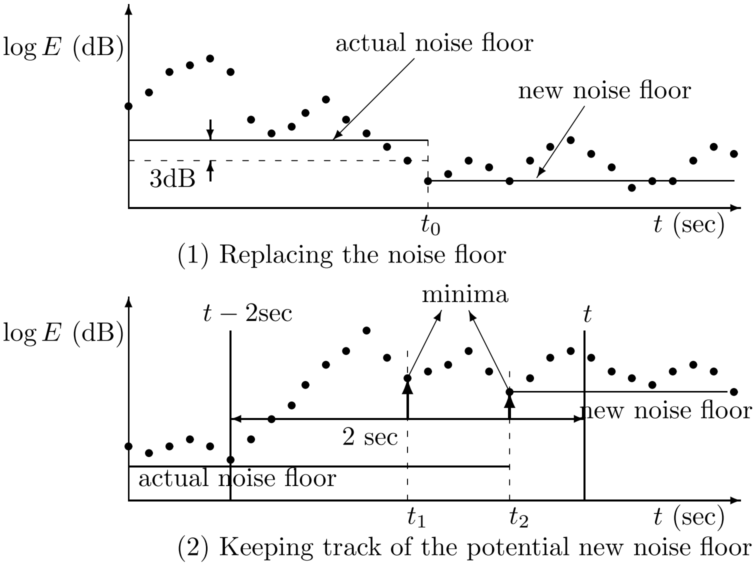


Fig. 4.5: Adapting the noise floor in the silence detector.

If a minimum is encountered in the maximum of the log-energy in three consecutive frames, then the following actions are taken.

1. If this minimum is more than 3dB below the actual noise floor, the latter is replaced by the former minimum.
2. If this minimum exceeds the actual noise floor, but is lower than the minimum that caused the last noise-floor update, then the position and the value of the minimum are stored in a minimum buffer.

If no silence was detected over the last 2 seconds, the noise floor is updated to the value of the lowest minimum in the buffer, the buffer will be cleared and the search for new silences will be resumed from the location of that minimum. Both situations are represented on Figure 4.5. In the upper panel the situation where the noise floor is replaced, is depicted. In the lower panel there were no silences detected since $t - 2$sec. Therefore the actual noise floor is replaced at time t_2, the lowest minimum in the buffer.

The detected silences will be used to discard silent speech segments from the classification and as features for the classification.

4.5.2 Feature identification

An appropriate acoustic and prosodic feature description for the created non-silent segments has to be conceived now. To that end I have performed a statistical analysis of the CGN training corpus which contains 3255 FP intervals, 75% of which are longer than 0.2 sec and 87% longer than 0.15 sec. In the rest of my research I have considered only the FPs which are longer than 0.15 sec as genuine FPs.

If more than 50% of the frames of a segment emerging from the previously described blind segmentation fall into such a genuine FP interval, the reference label of that segment is FP, otherwise it is NFP. By comparing the cumulative distribution functions (CDFs) of a feature for the FP and the NFP segments, I can identify features that are good candidates for contributing to the discrimination between FP and NFP segments. I will now discuss the features that were investigated and the discriminative power these features seem to have in the CGN training data.

Segment duration

The first feature I have investigated is segment duration. Our measurements showed that FP segments tend to be longer than NFP segments. This result confirms the observations also made by e.g. [41]. The FP and NFP segment durations both seem to exhibit Gamma distributions, but with clearly different parameters. The mean FP length is about 0.25 sec ($\sigma = 0.15$ sec), the mean NFP length is only 0.11 sec ($\sigma = 0.08$ sec).

Spectral stability

If $d_{i,j}$ is the Euclidean distance between the MFCC vectors of frames i and j, the distance D_{stab}, for segment (t_1, t_2), defined by

$$D_{stab} = \min_{t \in (t_1, t_2)} D_t, \qquad \text{with} \qquad D_t = \frac{d_{t,t-1} + d_{t,t+1}}{2} \tag{4.3}$$

is a measure of the maximum stability observed inside that segment. The frame t_s where D_t is minimal is called the most stable frame of the segment. Our measurements reveal that FP segments have a smaller D_{stab} than NFP segments. The mean value of D_{stab} for FP segments is 5.18 ($\sigma = 1.59$), whereas it is 9.10 for NFPs ($\sigma = 2.38$).

Stable interval durations

Starting from t_s, the stable interval of a segment can be defined as the largest interval around t_s for which $d_{t,t_s} < \Theta_d$ for all t in that interval. By selecting different values of Θ_d, one can determine different stable intervals and use the corresponding *stable interval durations* (SIDs) as segmental features. If $\Theta_d \leq D_{stab}$ the SID is equal to zero. Filled pauses clearly tend to have a longer SID than other speech segments. In my experiments I have considered the SIDs for $\Theta_d = 8, 10, 12, 14, 16$ and 18 as six distinct segmental features. The mean value of the SID for $\Theta_d = 12$ is 3.19 frames for NFP ($\sigma = 2.45$) and 11.59 for FP ($\sigma = 8.01$).

Silence before and after the FP

Another prosodic cue that was found to be effective for the detection of FPs is the presence of a silence (sil) in an adjacent segment (either before or after the segment under test). A silence is defined as an interval during which the log-energy is never more than 3 dB above the noise floor. Table 4.1 reveals that 80% of the FP segments are delimited by at least one silence, whereas this is only true for 61% of the NFP segments. I found that the adjacent silences are also longer in the case of FP segments than in the case of NFP segments. It is even so that the post-FP silences are bound to be longer (mean of 0.19 sec and σ of 0.23 sec) than the pre-FP silences (mean of 0.13 sec and σ of 0.16 sec). For the pre- and post-NFP silences I found a mean of 0.11 sec and a σ of 0.20 sec.

	sil before	no sil before	total
sil after	946	768	1714
no sil after	870	657	1527
total	1816	1425	3241

Tab. 4.1: Number of filled pauses with and without adjacent silences.

Spectral center of gravity

Another acoustic feature that was examined is the center of gravity of the mean log mel-power spectra observed in the identified stable interval of the segment:

$$g_S = \frac{\sum_{m=1}^{M} m\tilde{S}(m)}{\sum_{m=1}^{M} \tilde{S}(m)} \tag{4.4}$$

In this equation, $\tilde{S}(m)$ represents the log signal power in the m^{th} subband of an M channel mel-scale filter-bank ($M = 24$). I found that a center of gravity of 16 or more is a very good counter-indication for an FP. The mean center of gravity was 10.47 ($\sigma = 2.65$) for FP and 13.04 ($\sigma = 3.98$) for NFP respectively.

Simple filled pause model output

Another feature is the logarithm of the output of a 4 mixture GMM that was trained on all the frames belonging to filled pause intervals. Here too, the model inputs are the 12 MFCCs. I did not use delta MFCCs, because the spectral stability was already modeled by the SID features. The mean log score was -16.9 for FP ($\sigma = 1.4$) and -18.8 for NFP ($\sigma = 2.6$) respectively. This means that FP segments yield higher likelihoods in this simple model than NFP-segments.

Features related to the pitch

Goto [48] was successful in detecting the FPs in Japanese spontaneous utterances on the basis of features which represent frame-level changes of the fundamental frequency and the spectral envelope. On a set of 100 sentences, each containing at least one FP, the measured recall and precision rates were 84.9% and 91.5% respectively. I also investigated the discrimination capabilities of some segmental pitch features for the detection of FPs in Dutch spontaneous speech. The investigated features were:

- Pitch Regression Coefficient
  I defined this feature as the slope of the best fitting line through the nonzero pitches of the frames in a certain segment. No clear distinction was found between FP and NFP.

- Pitch Modulation Variance
  This feature is defined as the variance of the differences between the nonzero pitch values and the corresponding pitches predicted

by the linear regression model. Again this feature did not offer a significant discrimination between FP and NFP.

- Relative Pitch Ratio
  It is often supposed that filled pauses exhibit a low pitch compared to the surrounding speech segments. Therefore, the mean of the nonzero pitch values is computed for each segment and the ratio between this pitch and the mean pitch of the N preceding and N succeeding segments is defined as the relative pitch ratio of that segment. If the pitch of a segment is zero the relative pitch ratio of that segment is undefined and supposed to be equal to 1. If one of the contextual segments has a zero pitch it is excluded from the mean pitch computation of these segments. For $N = 7$ I observed a small ability to discriminate between FP and NFP segments. The mean relative pitch ratio for FP segments was 0.96 while it was 1.00 for the other segments.

Apparently, for a non-tonal language like Dutch, the pitch features are not that powerful for FP/NFP classification. Moreover, since the inclusion of a pitch extractor adds complexity to the acoustic front-end of the recognizer, I decided not to consider any of the pitch features for my FP detector.

4.5.3 Segment classification strategy

Since I aim to build a detector that can estimate the posterior probability of having an FP, given the acoustic observations, I propose to perform a classification of segments by means of an MLP (Multi-Layer Perceptron) with one hidden layer and one output which is, after proper training, supposed to provide exactly that probability.

A problem with the error back-propagation training of an MLP is that one often gets poor results when the prior probabilities of the classes are very different. Since only 1% of my segments have an FP label, I definitely am in that situation. Therefore, I first try to identify a large number of NFP segments by means of GMMs in order to discard them later and I perform the training of the MLP on the remaining segments. I have trained two GMMs to model the likelihoods $p(\mathbf{x}|\mathrm{FP})$ and $p(\mathbf{x}|\mathrm{NFP})$ with $\mathbf{x}$ representing the 12 features: (1) segment duration, (2) spectral stability, (3-8) SIDs for $\Theta_d = 8, 10, 12, 14, 16$ and 18, (9,10) silence duration before and after the segment, (11) spectral center of gravity, (12) simple filled pause model output. The FP model consisted of 8 and the NFP model of 64 mixtures with diagonal covariance matrices.

Since there is a good estimate of $P(\mathrm{FP})$, I can easily make an estimation of the *posterior* probabilities $P(\mathrm{FP}|\mathbf{x})$ and $P(\mathrm{NFP}|\mathbf{x})$ respectively.

Segments whose FP-to-NFP posterior probability ratio, denoted as PPR and defined as
$P(\mathrm{FP}|\mathbf{x})/P(\mathrm{NFP}|\mathbf{x})$ exceeds some threshold Θ_{PPR} are considered as candidate FP segments and are supplied to the MLP classifier. The others are considered as NFP segments. The number of candidate FP segments can be controlled by modifying the threshold (see Table 4.2). Given that there were 3255 FP and 344945 NFP segments in total, it is clear that with $\Theta_{PPR} = 10^{-6}$ I can retrieve about 89% of the FPs while eliminating 74% of the NFP segments, and increase the percentage of FP segments from 1 to 3.2%.

Θ_{PPR}	# FP segments	# NFP segments
10^{-4}	2429	41528
10^{-5}	2659	65169
10^{-6}	2879	91066

Tab. 4.2: Number of segments passing through the GMM filter.

In order to give the MLP some idea about the spectral envelope and the energy of the segment to classify, the MLP is supplied with 25 features: the 12 features that were used by the GMM, and 13 MFCCs characterizing the most stable frame of the segment t_s.

Since the MLP is presumed to estimate the posterior probability $P(\mathrm{FP}|\mathbf{x})$, segments are classified as FP if the MLP output exceeds some posterior FP probability threshold Θ_{PP}. The latter is used to control the desired balance between the recall and the precision of the FP detector.

4.5.4 FP detection results

The trained GMM-MLP tandem is evaluated on the previously mentioned CGN test corpus which contains 445 FPs, 440 of which are longer than 0.15 sec.

I tested the performance using a GMM filter with a threshold $\Theta_{PPR} = 10^{-6}$ and comprising an MLP with 15 sigmoidal hidden units (and thus 15x26+16 = 406 free parameters). The results of my tests are listed in Table 4.3 as a function of Θ_{PP}.

The data show that a precision of about 73% can be reached with a recall of 75%. Obviously, one can also operate the FP detector at a high precision (and a moderate recall) or at a high recall (and a moderate precision) if requested.

Θ_{PP}	Precision (%)	Recall (%)	F-rate (%)
0.05	44.4	91.1	29.8
0.15	59.8	83.6	34.8
0.20	65.1	81.6	36.2
0.25	68.5	78.9	36.6
0.30	72.9	74.5	36.8
0.40	77.5	65.7	35.5
0.60	83.5	50.7	31.5

Tab. 4.3: Precision and recall of the FP classification of the GMM-MLP tandem.

I tried to improve the MLP classifier by means of additional embedded training iterations on a larger database also including data for which no manually verified segmentation is available, but these attempts were not successful. A further increase in the number of hidden units did not result in any substantial improvement of the classification accuracy either.

4.6 Experimental evaluation

Now I discuss the recognition experiments I conducted with both the internally and externally informed disfluency handling approaches.

For the internally informed strategies, I report results for American English (Switchboard) and Dutch (CGN). For the externally informed strategies however, only results for Dutch are presented[2].

The recognition engine that I used was delivered by ESAT. It performs a single pass time synchronous beam search and it comprises gender independent acoustic models. A global phonetic decision tree defines a large number of tied states that are used in cross-word context and position dependent phoneme models. No speaker adaptation is applied. Both for Switchboard and CGN, the acoustic models were trained by ESAT.

The language model (LM) is a trigram back-off [59] language model which is retrieved from text material and/or orthographic transcripts of spontaneous speech. Good-Turing [42] is used as the smoothing technique. The LMs were also trained by ESAT, who chose to consider FPs as integral elements of the language because Pakhomov [83] obtained

[2] Since the research was sponsored by the Flemish Authority, the emphasis had to be on Dutch. However, the experiments on Switchboard are helpful to demonstrate that my baseline system exhibits a state-of-the-art recognition performance.

good results with this technique. He compared it to a baseline technique discarding all FPs and he found that keeping the FPs caused a reduction of the WER from 32.2 to 28.5% for spontaneous medical dictation.

For the CGN task the lexicon consists of the 40k most frequent words appearing in Dutch newspaper material, but supplemented with all the words that were needed to attain a full lexical coverage of the test set. This way, the results presented here are not influenced by the presence of out-of-vocabulary words. The lexicon contains a number of manually obtained pronunciation variants of the filled pause, such as @, @m, @@, @@m, @mm, @@@, . . . (SAMPA notation, see Appendix B). No probabilities were attached to these pronunciation variants, however.

4.6.1 Evaluation methodology

When evaluating the recognition systems, all FPs appearing in the reference and in the recognized word strings are removed, meaning that the WERs only measure the number of errors related to regular words. This approach which I will also adopt in all my further recognition experiments was also applied in [91].

4.6.2 Baseline system performances

For the Switchboard task, the acoustic models were estimated on 310 hours of Switchboard-1 data. A global phonetic decision tree defined 8k tied states and each tied state is modeled with a mixture of on average 220 tied gaussian distributions from a total set of 117k different Gaussians.

The software available at *http://www.nist.gov/speech/tools* is used to compute the WER and to assess the statistical significance of measured performance differences. Our baseline system obtains a WER of 29.8% on the Switchboard test set described in chapter 2.

For the CGN task the acoustic models are learned on 44 hours of Flemish spontaneous data from CGN. The global phonetic decision tree defines 3500 tied states. Per state, the Gaussians are selected from a set of 32k Gaussians.

The CGN test set comprises speech of 27 speakers and it contains 7041 regular words and 445 filled pauses. Hence, the FP rate is 445/7496 or 5.94% and thus significantly larger than the 2.7% which was measured on the CGN training corpus. The WER obtained with my baseline system on the test set is equal to 36.1%. The main reason for this may be that the LM for Switchboard is more adapted to the task than the LM for CGN. Other reasons could be the larger diversity of the data, the larger mismatch between the LM and the spontaneous data, the smaller size of the acoustic model training database, etc.

4.7 Testing internally informed strategies

In ESAT one has compared the three proposed LM context manipulation models individually with the baseline system (BS, never modify the context) and with the standard context manipulation (SCM) system (always modify the context) proposed in [98]. The results of this experiment are summarized in Table 4.4. They confirm that the standard method does not offer any significant improvement. They also show that the word repetition model yields a small but statistically significant improvement (highlighted result) whereas the hesitation and the restart model unfortunately are totally ineffective.

disfluency	model	BS	SCM	internal
repetition	repetition	29.8%	29.7%	**29.6%**
filled pause	hesitation	29.8%	29.9%	29.8%
	restart	29.8%	29.8%	29.9%

Tab. 4.4: WERs for the baseline system and for systems using standard context manipulation models (SCM) and newly proposed context manipulation models, respectively.

A more detailed analysis shows that the proposed repetition model changes less than 5% of the recognized sentences, but mostly in the right sense. The low number of changed recognition outputs is not that surprising given the low WR-rate in spontaneous speech (around 1.4% of the words, as shown in section 3.1.2).

I repeated the same experiments on the CGN corpus. Applying the standard context manipulation technique on FPs resulted in a WER of 35.9% (see Table 4.5). Using the proposed repetition and the hesitation model I got very similar WERs of 35.9% and 35.8% respectively. All three WERs are statistically significantly lower than the baseline WER of 36.1%, but the differences remain small due to the small WR-rate (below 1%) observed in the CGN data.

In general I can conclude that none of the tested context manipulation models can cause substantial gains in recognition accuracy. However, for both tasks the best performances were obtained with one of the models proposed by ESAT. For Switchboard the most successful internal model is the repetition model, whereas for CGN the best performing internal

system	disfluency handling	WER (%)
BS	none	36.1
BS+SCM	internal strategy	35.9
BS+repetition (proposed)		35.9
BS+hesitation (proposed)		35.8
BS+drop	external strategy	**34.5**
BS+LMA		**34.6**
BS+LMA+drop		**34.3**
BS+SCM+drop	combination	**34.3**
BS+SCM+LMA		**34.6**
BS+LMA+drop+hesitation		**34.1**

Tab. 4.5: WERs for systems using externally informed FP handling methods. Results in bold differ significantly from the baseline (BS)

model is the hesitation model. The latter seems logical since the CGN test set contains many FPs. In both cases the relative improvement of the WER is less than 1%.

I attribute the small gains of the two models which are triggered by a filled pause to the fact that these models rely on the detection of a filled pause by the decoder. In the current system it is fairly easy to hypothesize such a filled pause and thus to trigger a prediction context modification at too many places where there is no disfluency in the signal. A better alternative would be to create a *separate* acoustic model for the filled pause as a word-level unit, as was done in [91]. ESAT tried this on the Switchboard task but without any success.

4.8 Testing externally informed strategies

As already discussed before, one can anticipate that frame dropping will perform best in combination with a detector having a high precision (do not throw away useful frames) whereas LM adaptation (LMA) will profit most from a detector with a high recall (make it applicable at all places where an FP is likely to occur). Therefore, I have investigated the performance of the two proposed strategies in combination with the same FP detector but working at different operating points in the (precision,recall) plane.

By applying frame dropping in combination with an FP detector with a high precision (83.5%) it was possible to reduce the WER from 36.1 to

34.5% (see Table 4.5) which is a statistically significant reduction. If the precision is lowered to 50%, the WER increases to 35%.

By applying LM adaptation in combination with an FP detector with a high recall (91.1%) it was possible to reduce the WER from 36.1 to 34.6% (see Table 4.5) which is again a statistically significant reduction. For LMA, the attained performance gain is about the same (34.6%). Obviously, it depends on the values of the control parameters C_1, C_2 and D discussed in section 4.4.2. The best choices for C_1 and C_2 are 1 and 0 respectively. The value of D is not critical: as long as $D > 0.2$ sec, the performance gain changes by less than 0.2%. An advantage of imposing a small maximum delay D is of course that it constrains the maximum time delay introduced by LMA.

4.9 Combining FP handling techniques

Since both externally informed strategies yield an improvement, it was no more than logical to investigate whether they can complement each other, and whether they can also be combined effectively with the internally informed strategies presented in the previous section.

4.9.1 Combining frame dropping and LM adaptation

Since frame dropping is most effective with a high precision FP detector and LM adaptation with a high recall FP detector, it seems logical to apply frame dropping on FP segments, with a high score (> 0.5), and LM adaptation on FP segments with a moderate score (between 0.05 and 0.5).

With the proposed combination of techniques I was able to further reduce the WER a little bit, to 34.3% (see Table 4.5). Another advantage of this combination is that its results are less dependent of the choice of the control parameters. The effect of D is negligible and $C_1 = C_2 = 0$ seems to be a good choice for the other two parameters.

4.9.2 Adding standard context manipulation

Since SCM also seemed to offer a small improvement, I did investigate whether this improvement is maintained in combination with either one of the two externally informed methods for handling FPs. The results in Table 4.5 show the following

1. The system BS+SCM+drop performs equally well as BS+LMA+drop
2. The system BS+SCM+drop is better than BS+drop: the 0.2% gain of SCM is fully conserved

Where SCM is helpful in combination with drop, it is not in combination with LMA. Apparently, SCM cannot correct errors that are not already corrected by LMA. This is also confirmed by the fact that BS+SCM+LMA does not outperform BS+LMA.

4.9.3 Adding the hesitation and the repetition model

Since the hesitation model did offer a small improvement of the recognition accuracy on SWB, I have investigated whether this improvement is still present when the model is used in combination with frame dropping and LM adaptation. For CGN the system BS+hesitation was the best internal system. I tried to combine this system with BS+LMA+drop and I found that it further reduced the WER to 34.1% (see Table 4.5).

I also performed a test with a system incorporating all the techniques: frame dropping, LM adaptation, hesitation and word repetition context manipulation, but with this system, I obtained a WER of 34.2%, meaning that the repetition model is not effective on top of all the other methods.

4.10 Additional experiments and discussion

Although my externally informed search strategies result in a reduction of 2% absolute of the WER, this reduction is not as spectacular as I anticipated when I started my research. Therefore, I conducted a detailed error analysis in order to find an explanation for this. I also wanted to find out how much larger the improvement could have been if a perfect FP detector were available.

4.10.1 Detailed error analysis

In order to perform my error analysis I have selected a small corpus of 118 CGN test sentences (3737 words) containing at least one filled pause in their reference transcription. In total this small corpus comprised 250 FPs.

For the evaluation of my disfluency handling methods I first of all measured the over-all WER and the local WER which I defined as the WER observed in short windows covering the reference word in front of the filled pause, the filled pause and the reference word just after that filled pause. By comparing these two WERs, it should be possible to check whether or not FPs cause extra problems for the recognizer. In order to find out whether an FP is likely to trigger a chain reaction, I have also counted the number of consecutive regular word errors that were produced in the vicinity of each FP. The following example of an aligned reference and recognized sentence pair was found in my CGN test data:

reference:	...	elk	jaar	uh	of	tot	de	twee	...
recognized:	...	elk	jaar	*u*	*op*	*met*	de	*thee*	...

Translated to English (with loss of correspondence between speech and recognized words), this gives

reference:	...	every	year	uh	or	to	the	two	...
recognized:	...	every	year	*you*	*on*	*with*	the	*tea*	...

If one first removes the FP from the reference sentence before analyzing the errors, it becomes clear that in the above example (Dutch sentences) there are two local errors (the insertion of 'u' and the substitution of 'of') and three consecutive word errors (the insertion of 'u' and the substitutions of 'of' and 'tot').

The local and the global WERs and the number of consecutive word errors per FP for the baseline system and the BS+LMA+drop (Best) system are listed in Table 4.6. Apparently the over-all WERs obtained for the small sub-corpus are very representative of the WERs obtained for the full corpus, but this is of course not the most important observation to retrieve from the Table. The more important observations are the following:

system	global WER (%)	local WER (%)	consecutive errors/FP
BS	36.5	56.7	1.50
Best	34.3	42.0	1.17

Tab. 4.6: Detailed error analysis: the over-all WER, the local WER (in the vicinity of an FP) and the number of consecutive word errors per FP.

1. The local WER of the baseline system is substantially higher than its over-all WER. I consider this as a clear support of my hypothesis that FPs cause particular problems for the recognizer.

2. The expected chain reaction is less pronounced than originally anticipated: the average number of errors induced by an FP is only 1.5, meaning that the maximum gain in performance attainable with FP handling strategies is bound to be smaller than 1.5 times the FP rate.

3. Our best disfluency handling strategy does have a significant impact on the local WER (a reduction of 14.7% absolute), meaning that it does what it is supposed to do, namely deal with problems due to the presence of an FP.

I will now try to make a realistic estimate of the maximum performance gain that can be achieved by means of disfluency handling methods. If all the regular word errors occurring in the vicinity of an FP were effectively caused by the presence of that FP, the maximum gain would be 1.5 errors per FP.

However, if I randomly select 250 regular reference words and if I count the associated number of consecutive word errors in the same way as I did with the FPs, I find a number of 0.8 errors per word. Consequently, I argue that the number of additional errors that is on average induced by the presence of an FP is only of the order of 1.5 - 0.8 = 0.7.

Given that the FP rate in our data is 5.94%, this would finally lead to a maximum attainable gain in over-all WER of about 4.1% for these data. This gain is about 2 times larger than the gain of 2% I actually obtain with my best system.

This last result contrasts a bit with the fact that the local WER of my best system is already pretty close to the over-all WER of the baseline system. This must mean that my FP handling methods introduce a number of new errors in areas not corresponding to an FP in the speech. If I would be able to conceive a better external FP detector, I would expect less of these errors.

4.10.2 Impact of the external FP detector

In order to confirm the above hypothesis, I have evaluated the two externally informed FP handling methods in combination with a ’perfect’ FP detector which I define as the FP detector generating the manually labeled FP segments (provided with the CGN data) with an FP score of 1. In Table 4.7 I have collected the recognition performances when the FP handling systems are being supplied with the real and the perfect FP detector outputs respectively. The improvements with respect to the baseline system are expressed in terms of the WER and the average number of corrected words per FP, denoted as #cwrds/FP.

system	perfect FP detector		real FP detector	
	WER (%)	#cwrds/FP	WER (%)	#cwrds/FP
BS+drop	32.0	0.70	34.5	0.27
BS+LMA	34.4	0.30	34.6	0.25

Tab. 4.7: Achievable gains of two externally informed methods using a real and a perfect FP detector respectively.

My results (Table 4.7) do confirm that with a perfect FP detector, the WER can be reduced by 4.1% (0.7 corrections per FP), the estimated upper bound of the improvement. They also demonstrate that this gain can only be achieved by means of frame dropping since LM adaptation alone never yields more than 0.3 corrected words per FP. I argue that the latter result demonstrates that the decoder part of the speech recognizer itself is not able to give a correct interpretation to the FP frames it is confronted with.

4.10.3 Dependency on the FP rate

I also investigated whether there is a correlation between the attained performance gain and the FP rate of the speech utterances. Therefore I have performed a recognition experiment on 27 speakers selected from the CGN training corpus on the basis of their FP rate. Note that these speakers were not involved in the training of the acoustic models nor the language model of the recognizer.

I have divided this set into five subsets by grouping the speakers on the basis of their FP rate. Table 4.8 gives an overview of the characteristics of these five databases and the WERs obtained with some of my systems on these databases. Apparently, the data are significantly more difficult to recognize than the test data I used before. The average performance gain due to my disfluency handling methods is also smaller (around 0.9% absolute) than before, but it is still statistically significant.

The bottom row of Table 4.8 shows that the improvement of the best system over the baseline system is roughly correlated with the FP rate. The figures also show that frame dropping starts to harm the performance if the FP rate gets too high: the best system for databases 4 and 5 is the

	db 1	db 2	db 3	db 4	db 5
FP rate (%)	1.97	4.16	5.57	7.57	8.85
#words	5269	5711	5188	7089	4373
#speakers	6	6	5	5	5
#FPs	106	248	306	581	425
WER(BS)	35.50	47.66	44.79	36.12	44.31
WER(BS+LMA)	35.29	47.30	43.75	34.78	43.45
WER(BS+LMA+hesitation)	35.25	47.28	43.77	**34.60**	**43.33**
WER(BS+LMA+drop+hesitation)	**34.83**	**47.22**	**43.38**	35.13	43.78
WER(BS)-WER(best)	0.7	0.4	0.4	1.5	1.0

Tab. 4.8: Influence of the FP-rate on the several discussed methods.

one without frame dropping whereas for databases 1 to 3 it is the one incorporating both frame dropping and LM adaptation.

4.10.4 Dependency on the baseline WER

In the course of my research I have tested FP handling methods in combination with baseline systems having a lower recognition accuracy (because they did not yet incorporate a spontaneous speech language model or because their acoustic models were trained on read speech only). From these tests it follows that the improvement induced by my methods does not decrease when better baseline systems become available. On the contrary, while the baseline WER could be reduced from 45.6 to 36.1%, the improvement induced by my methods actually increased from 1.7 to 2%. This is a hopeful result in view of the expectation that acoustic and linguistic models of spontaneous speech will further improve now that more and more spontaneous speech corpora are becoming available to the speech community.

4.11 Conclusion

In this chapter I have proposed different strategies for coping with disfluencies in the search engine of a spontaneous speech recognizer. I made a distinction between internally informed approaches that totally rely on the standard knowledge sources (acoustic models, pronunciation models and language model) and externally informed approaches that also take into account evidences for disfluencies as they emerge from an external acoustic preprocessor.

The external acoustic preprocessor operates independently of the search and searches for FPs on the basis of acoustic and prosodic features that are not accessible to a standard recognizer. After having selected the appropriate features I could build an FP detector that is capable of detecting a large fraction of the filled pauses with a high precision. I proposed two strategies for incorporating the detector outputs into the search engine, namely frame dropping and language model adaptation.

Experiments on Flemish spontaneous speech showed that the externally informed approaches for handling FPs yielded a moderate but statistically significant gain in recognition performance: the error rate could be reduced from 36.1 to 34.3%. This improvement corresponds to the correction, on average, of about 0.3 regular word errors per FP occurring in the speech. This is still quite below the maximum of 0.7 word corrections per FP I have been able to demonstrate by using an oracle FP detector.

By also including the ESAT hesitation model in the decoder, I could finally reduce the WER to 34.1%. A detailed analysis of my results has demonstrated (1) that the size of the improvement is correlated with the disfluency rate of the speech, (2) that the attained improvement does not decrease when the WER of the baseline system decreases, and last but not least, (3) that the largest improvement obtained thus far is about 43% of the improvement that could have been achieved with an oracle (manual) FP detector. The latter conclusion allows me to assess the additional improvement that would be possible to attain if a significantly better FP detector could be conceived.

Since a continuation of the work on FP-handling methods was not bound to yield much more than an extra 1 or 2% drop in WER, and since this work could not be carried out in the context of any research project that was running at ELIS-DSSP at that time, my research was therefore re-oriented towards the use of phonological features (PHFs) for the processing of non-typical speech. Examples of such speech are disordered speech, speech from non-native speakers, and speech containing words of a foreign origin.

5 Detection of Phonological Features

In the three chapters of part II of my dissertation I discuss (1) the phonological feature detector I developed, (2) the way in which it was validated in an alignment assessment tool and (3) the ways in which it can be integrated in a speech recognizer, either alone or in combination with traditional models working with acoustic features. However, before I can start these discussions I have to provide some information about the physiology of the vocal tract. Sections 5.1 and 5.2 can be skipped by readers who are familiar with this material.

5.1 The physiology of the vocal tract

The physiology of the human speech production system is depicted in Figure 5.1. The names mentioned on the Figure are the so-called scientific names which are of a Latin origin. Some of these names sound unfamiliar and need some further explanation.

- **velum :**
  The back-ward part of the palate.
- **pharynx :**
  The space between the tongue root and the wall of the upper throat.
- **larynx :**
  The structure on which the vocal chords are attached.
- **epiglottis :**
  The tissue fold beneath the tongue root that makes sure the larynx is covered during swallowing, making the food going into the gullet and not to the lungs.

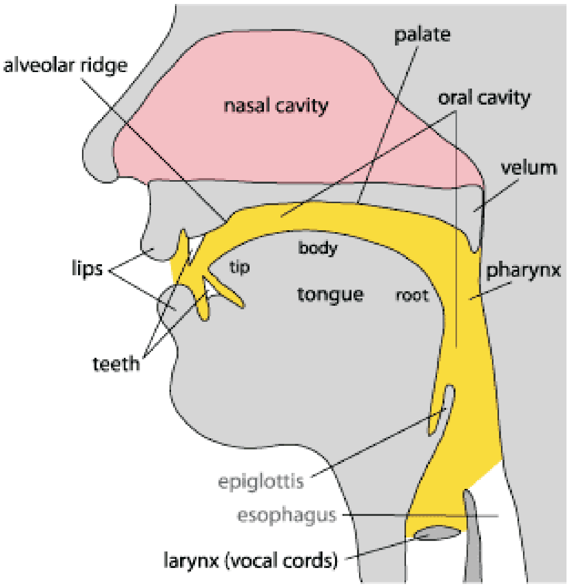


Fig. 5.1: The anatomy of the human speech production system.

Dutch name	English name	scientific name	adjective
lippen	lips		labial
tanden	teeth		dental
tandkas	alveolar ridge	alveolus	alveolar
hard gehemelte	hard palate		palatal
zacht gehemelte	soft palate	velum	velar
huig	uvula		uvular
keel	throat	pharynx	
strottenhoofd	voicebox	larynx	laryngeal
tongpunt	tongue tip	apex	
tongblad	tongue blade	lamina	
tonglichaam	tongue body	dorsum	dorsal
tongwortel	tongue root		
stembanden	vocal chords		

Tab. 5.1: Parts of the vocal tract and their Dutch, English and scientific names.

Table 5.1 contains the different names that are in use (in Dutch and English) for the different parts of the vocal tract. Of course the vocal tract is not able to produce the speech sounds without an air pressure. That air pressure mostly originates in the lungs.

5.2 Articulatory versus Phonological Features

Articulatory features are defined as *descriptors* of the positions of the articulators, such as the position of the tongue tip, tongue body, the palate, ... There exist methods based on X-Ray microbeam (XRMB) cinematography [124; 84] and electromagnetic articulography (EMA) [36] to measure these positions in a fairly reliable way. There also exist a number of small corpora with a limited number of speakers that come with articulatory features. An Example is the MOCHA-database for English [123].

Most of the speech data however come without articulatory features. Consequently, if I want to create a speech characterization in terms of these features, I will have to conceive speech analyzers that retrieve these features from the speech waveform, or from a standard parametric representation of this waveform like e.g. the MFCC parameters. Features like the MFCCs are called Acoustic Features (ACFs).

If one wants to derive articulatory features from the ACFs one has to solve the so-called inversion problem, which aims at finding the hidden cause of the observed result. Unfortunately, several articulatory configurations can produce identical (or very similar) acoustic features. The inversion is thus a *one-to-many* mapping problem. This inversion problem has been studied by speech scientists for some time [6; 92]. Having at my disposal the ACFs, I can adopt two major techniques for extracting articulatory information.

1. **Analysis by synthesis**
   This method [53; 89] uses a speech production model with free parameters. In order to obtain the free parameters (articulatory features), the error between the real (observed) ACFs from the training data and the ACFs obtained from the speech production model, is measured. On the basis of this error new and better estimates are made for the articulatory features.

2. **Nonlinear mapping**
   This method uses a nonlinear mapper to perform the mapping from the ACF space into the articulatory space.

In order to learn a mapping one needs *knowledge* about the relation between the acoustic features and their articulatory causes. These relations are studied in *phonology*. That is also why articulatory feature estimates derived by nonlinear mapping are called phonological features (PHFs), as opposed to the genuine articulatory features (ARFs) emerging from direct measurements of the articulatory configuration. In phonology one tries to classify sounds according to several *distinctive features*. Such a distinctive feature is what distinguishes one sound from another. The distinctive features aim to refer to three main aspects of articulation:

- **Phonation** : What is the role of the vocal chords in the speech production.

- **Manner of articulation** : What kind of phenomena (see later) are responsible for the acoustic properties of the sounds.

- **Place of articulation** : Where in the vocal tract do the critical articulation processes take place that determine the acoustic features of the sound.

The phonological correlates of these three aspects of articulation will be called feature dimensions and are explained in more detail now.

5.2.1 Voiced versus unvoiced

Phonation is treated as a separate process because there exist many sounds that only differ in this dimension but that are otherwise articulated in virtually the same way. The difference between /s/ and /z/ for instance resides in the fact that during the production of the /z/ the vocal chords vibrate while they do not during the production of the /s/.

When the vocal chords vibrate, the speech is said to be voiced (=phonological feature), otherwise it is unvoiced or voiceless.

5.2.2 Manner of articulation

5.2.2.1 Plosive

Plosives or stops are sounds which are caused by a process of creating an obstruction for building up the air pressure at some place in the vocal tract, and suddenly removing the obstruction so that the pressure is released. Consequently one can discern two phases or sub-phonemic units in the realization of a plosive: a *closure* interval (pressure build up) and a release interval. Obviously, a plosive cannot be sustained: once the air has escaped, the sound is finished. Examples of plosives are the initial sounds of 'pot', 'bat', 'tail', 'dog', 'cat' and 'gate'.

5.2.2.2 Fricative

A fricative is a hissing sound caused by air flowing through a small gap. Examples of fricatives are the initial sounds of 'four', 'think', 'shame', 'van', 'sick', 'hear', 'there', 'zinc'.

5.2.2.3 Nasal

A sound is is called nasal if the air is (partly) escaping via the nose because the soft palate is lowered. Examples of nasal sounds are the initial sounds of 'mean', 'nine' and the final sound of 'wing'.

5.2.2.4 Lateral

A sound is called lateral if the air is blocked in the central part of the vocal tract, but is escaping at both sides (left/right) of the tongue. A typical example of such a sound is the initial sound of 'left'.

5.2.2.5 Consonant/vowel/approximant

A sound is called a *consonant* if the air flow is restricted by a mechanism as described above. If the air flow is unrestricted the sound is called a *vowel*. However there are some sounds like the initial sound in ’rat’, ’young’ and ’what’, that match the definition of vowel, but are still not considered as true vowels. These sounds will from now on be called *approximants*, because the obstruction of the air flow necessary for producing a consonant is approximated.

5.2.2.6 Affricates

Sounds composed of a plosive followed immediately by a fricative on the same articulation place are given a special name: affricates. The final sounds in ’catch’ and ’edge’ are two examples.

5.2.3 Place of articulation

For consonants the place of articulation is determined by the place where the cross-section of the vocal tract is minimal. This can be at the upper lips (labial), the upper teeth (dental), the alveolar ridge (alveolar), the soft palate (velar), the epiglottis (glottal).

5.2.3.1 Lips

If both lips are used to articulate the sound, I denote the sound as labial or bilabial. English examples are the initial sounds in ’pot’, ’bad’ and ’mum’. Two English sounds make use of the lower lip together with the upper teeth and hence are called *labio-dental*: the initial sounds in ’fan’ and ’van’.

5.2.3.2 Teeth

The two well-known ’th’-sounds in English (initial sounds in ’the’ and ’think’) are produced by forcing the air between the tongue tip and the teeth.

5.2.3.3 Alveolar ridge

An alveolar sound is formed when the tongue tip hits the bony bulge behind the upper teeth, also called alveolar ridge. A lot of English consonants are alveolar for example the initial sounds in ’dark’, ’size’ and ’now’. As can be observed, alveolar consonants can have several manners of articulation. Four sounds get the label *palato-alveolar* or *post-alveolar*. This is because the tongue hits both the alveolar ridge and the front of the hard palate. The sounds were are talking about are the ending sounds of the so-called affricates.

5.2.3.4 Soft palate

The soft palate is situated at the rear of the mouth. Velar sounds are produced by pushing the tongue back to the soft palate. They are the initial sounds in 'cat', 'goat' and the final sound in 'wing'. The initial sound in 'what' is often considered as *labio-velar* because there are two points of constriction: the lips are closed and the tongue is pushed back towards the velum.

5.2.3.5 Glottis

Glottal sounds are produced by closing the glottis. An example is the initial sound in 'hat'.

5.2.3.6 Retroflex

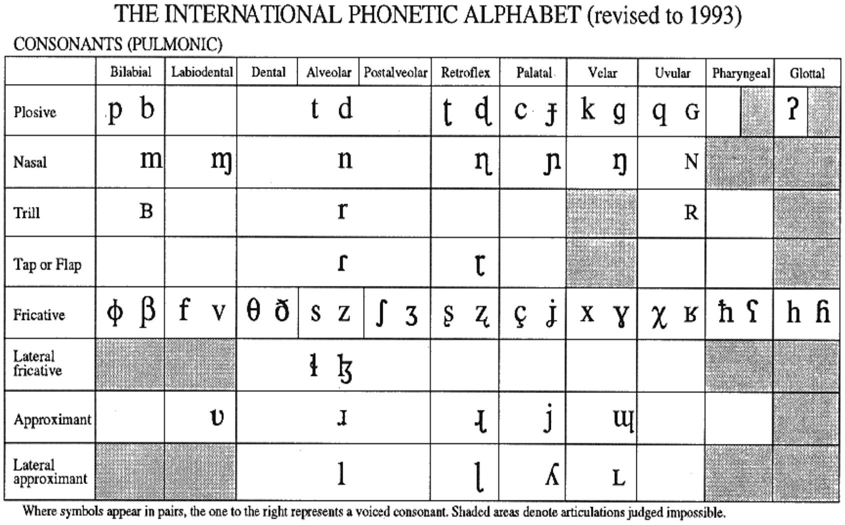
Retroflex sounds are produced with the tip of the tongue curled up, but more generally it means that it is post-alveolar without being palatalized (touching the hard palate with the tongue)

5.2.4 The International Phonetic Alphabet

The International Phonetic Association (IPA,[56]) has fixed an alphabet, called the International Phonetic Alphabet, for describing phonemes in all possible languages that fall in a certain category with respect to the manner and place dimension of articulation. All possible consonant symbols representing the IPA consonant chart can be found in Figure 5.2.

There are two manners of articulation in the IPA consonant chart that I did not explain yet. The first one is *trill* which is typical for the initial sound in the Dutch words 'rat' and 'rot'. A trill sound is a vibration of the tongue or uvula against the place of articulation.

The second manner of articulation is *flap* or *tap*. The term flap is often a synonym for the term tap. A flap involves a rapid movement of the tongue tip from a retraced vertical position to a more or less horizontal position, during which the tongue tip brushes the alveolar ridge. Intervocalic flapping is a phonological process found in many dialects of English,



THE INTERNATIONAL PHONETIC ALPHABET (revised to 1993)

CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ	▨	ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ	▨	▨
Trill	ʙ			r				▨	ʀ		▨
Tap or Flap				ɾ		ɽ		▨			▨
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative	▨	▨		ɬ ɮ						▨	▨
Approximant		ʋ		ɹ		ɻ	j	ɰ			▨
Lateral approximant	▨	▨		l		ɭ	ʎ	ʟ		▨	▨

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

Fig. 5.2: The International Phonetic Alphabet: the consonants.

especially North American English. In the word 'butter' for example the /t/ can be replaced by a flap.

The phonetic symbols for English and Dutch consonants that will be used from now on are represented in Table 5.2. If a symbol is used only for Dutch (D) or English (E), it is marked by a word of that language between brackets.

	lab.	lab-dent.	dental	alv.	post-alv.	retro.	velar	glot.
plos.	p b			t d			g k	
nas.	m			n			ŋ	
trill				r ('**r**at')				
fric.		f v	θ ('**th**in') ð ('**th**is')	s z	ʒ ('vi**s**ion') ʃ			h
appr.					j	ɻ('**r**oll')		
lat.				l				

Tab. 5.2: English and Dutch consonants in the IPA chart.

5.2.5 Vowel Features

Vowels differ from consonants in that the air can flow without obstruction through the vocal tract. The articulators affecting the properties of vowels are the position of the lips and the tongue body. Nevertheless, the discrimination between vowels can be very subtle and sometimes undetectable even by humans. The vowel chart of IPA in Figure 5.3 is an

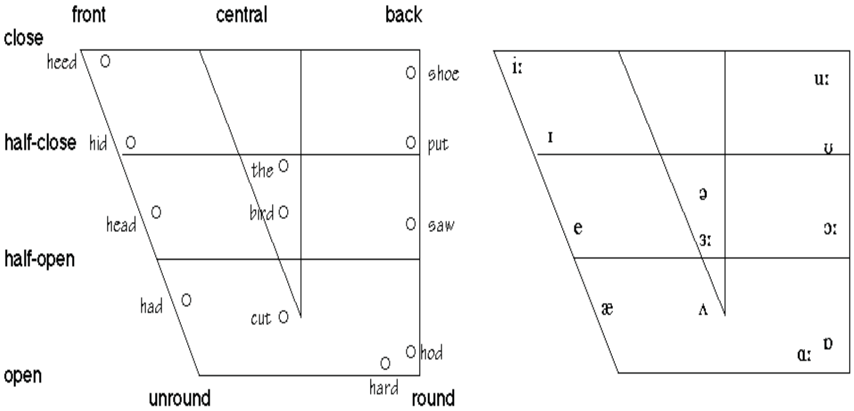


Fig. 5.3: The English vowels mapped on the IPA vowel-chart. Round/unround is not a dimension in this chart, but more like a binary variable. Many phones in the chart can be either round or unround.

attempt to represent all vowels in one single diagram. The X-axis of this diagram shows the horizontal place of articulation i.e. the place of the tongue body (front, central or back). The Y-axis is equivalent with the *vowel height* and represents the vertical position of the tongue body, also referred to as the open/close dimension, where close is identical to high and open to low. Vowels appearing on the same spot in the vowel chart can however differ from each other because of the lips. Some vowels can be either round or unround. Some vowels can be pronounced long or short. If they are pronounced long, a semi-colon (ː) is placed behind the symbol. Vowel duration is however not a distinctive feature in languages like English or Dutch, but more a practical issue.

5.2.6 Diphthongs

A diphthong is a sound that gradually shifts from one vowel to another. Basically diphthongs are no longer located in a certain area of the vowel chart, but they can be represented in the chart by a trajectory. In English there are 5 diphthongs:

- /eɪ/ : day, wait, vein,...
- /aɪ/ : buy, fine, sight,...
- /ɔɪ/ : boy, join, voice,...

- /əʊ/ : boat, slow, go, coat, ...
- /aʊ/ : now, fowl, cow, ...

Diphthongs always end in one of the upper corners of the vowel chart i.e. in /ɪ/ or /ʊ/. This is a language independent fact. There are however three more diphthongs like in 'sheer', 'share' and 'sure' (central diphthongs) that closely resemble the vowels /ɪ/, /e/ en /ʊ/. In many phonetic transcriptions these diphthongs are not used however.

5.3 An appropriate description for ACF-to-PHF mapping

I will first discuss some recent work on PHF extraction methods by means of the mapping paradigm before I will propose my feature set and detection strategy.

5.3.1 Previously proposed feature sets

Over the past decades, a multitude of feature sets have been proposed. In almost any case the phonological features are grouped in so-called dimensions, and each dimension is represented by a collection of binary features that can be on or off. Even dimensions such as "place of articulation" are not modeled by a single continuous variable but by a number of binary features corresponding to a number of "typical" places, usually associated with one of the parts of the vocal tract. One of the main advantages of this binary encoding of the phonological properties of a sound is that it achieves that all features can be treated in a unified way, that pattern classifiers can be trained to produce the posterior probabilities of these features and that they can be evaluated as classifiers.

In this section I recall some of the most popular feature sets that were proposed, and I also provide some information concerning their detectability. Table 5.3 is intended to provide a brief summary of what I found in the literature.

In [50] the features are grouped in 5 dimensions: (1) phonation, (2) manner of articulation, (3) place of articulation, (4) front-back and (5) roundness. Each dimension has a number of associated feature values: phonation has 2 values (voiced, unvoiced), manner of articulation 5 (approximant, fricative, nasal, stop, vowel), place of articulation 9 (labial, labio-dental, dental, alveolar, velar, glottal, high, mid, low), front-back

reference	#features	#dimensions	frame accuracies (%)	method
[50]	20	5	72-85	RNN
[17]	26	7	11-96	MLP
[70]	25	8	–	DBN
[40]	22	6	82	DBN
[120]	21	6	84	DBN
[61]	13	4	88-98	RNN
[61]	27	8	69-92	RNN

Tab. 5.3: Some PHF proposals (nr. of features and dimensions) and (reported) frame-level classification accuracies (corresponds to the range of accuracies across features).

2 (front, back) and rounding also 2 (round, unround). Recurrent neural networks (RNNs) were used for the detection of the features and the accuracy was measured both on frame level and on segmental level. For the manner features e.g., the error at the frame level was 15.5% and at the segment level 35.7%. For place of articulation the measured frame level error was 28.4% and the segmental error was 57.1%.

In [17] a system is proposed for the classification of PHFs that makes use of an array of 7 feed-forward MLPs, each treating one of the following dimensions: (1) place of articulation (9 values: labial, alveolar, velar, dental, glottal, rhotic, front, central and back), (2) manner of articulation (6 values: vocalic, nasal, stop, fricative, flap and silence), (3) phonation (2 values: voiced, unvoiced), (4) static/dynamic spectrum (2 values), (5) roundness (2 values: round, unround), (6) vowel height (3 values: high, mid, low) and (7) intrinsic vowel length (2 values: tense, lax). A 'nil' category is used to designate non-relevant features within a dimension. The total number of features is thus equal to 26. One could make two small remarks about the choice of the features. First 'rhotic' was used as a place feature. A rhotic speaker pronounces the /r/ after a vowel, like in /world/, whereas a nonrhotic speaker will not pronounce it or replace the /r/ by a schwa. Secondly, the rather strange assignment of 'velar' for segments like /sh/ and /zh/ was probably because the authors wanted to avoid an extra category 'post-alveolar'. Classification performance for the place of articulation features ranges between 11% correct for the 'dental' feature to 79% correct for the 'alveolar' feature. For manner of articulation the performance ranges from 45% for 'flap' to 96% for 'vocalic'. The main strategy in this work is to train separate place classifiers for each manner category. This is only meaningful if the manner detection can be done accurate enough. Therefore, only the frames with a high confidence can be considered for further classification. In order to show the potential of this method, a test was performed based on an ideal manner classification emerging from the reference labels. The conclusions were: (1) a considerable gain is obtained for the place classification when manner-specific networks are used (2) for the other feature dimensions the gain is less significant. In [121] the same approach is adopted for Dutch. Here, cross-lingual tests were implemented because the networks were trained on English and tested on Dutch. This gave a worse performance for the place classification.

In [70] eight feature dimensions were used: (1) phonation (voiced, unvoiced), (2) velum (closed, open), (3) manner of articulation (closure, sonorant, fricative, burst), (4) place of articulation (labial, labio-dental, dental, alveolar, post-alveolar), (5) retroflex (off, on), (6) tongueBody-LowHigh (low, mid-low, mid-high, high, nil), (7) tongueBodyBackFront (back, mid, front, nil), (8) roundness (off, on). A 'nil' category was used for non-relevant features. In this work the traditional HMM-based approach to automatic speech recognition is abandoned and a Bayesian approach is implemented. Experiments on Aurora 2.0 (small vocabulary and noisy speech) yielded an 8% relative reduction (from 1.3% to 1.2%).

In [40] PHFs are detected by means of a (Dynamic Bayesian Network) DBN, and a comparison is made with MLPs. The dependencies between the features are studied. The choice of the features is almost identical to the set used in [50]. There are six feature dimensions: (1) manner of articulation (approximant, fricative, nasal, stop, vowel, silence), (2) place of articulation (labial, labio-dental, dental, alveolar, velar, glottal, high, mid, low, silence), (3) phonation (voiced, unvoiced, silence), (4) rounding (round, unround, nil, silence), (5) front-back (front, back, nil, silence) and (6) static/dynamic spectrum (static, dynamic, silence). A separate model for each feature dimension was implemented. The baseline system consisted of a DBN in which all features were modeled independently. Next, two ways of modeling the dependencies between the features were implemented. The main conclusion was that a DBN does not perform any better than a neural network on the feature recognition task. A possible reason - according to the authors - is the necessity of having to use simple observation functions in a DBN.

King et al [61] investigated three phonological feature sets: (1) the Sound Pattern of English (SPE) system, (2) a multivalued feature system and (3) a feature system based on Government Phonology [51] which uses a set of structured primes. All feature extractors were based on RNNs and the tests were carried out on TIMIT. As King et al. argue, it is worth discussing the nature and design of the feature set. The three sets they tested are representative of three generations of research in phonological features, with the multivalued set being the most popular, the SPE set representing the original generative tradition and the Government Phonology set representing the more current phonological theory. There are however many more feature sets and phonological theories that could yield equally valid feature candidates as I have tried to demonstrate by this short overview of the literature. While some of these candidate feature sets have features in common with the three sets described here, many are completely different. Further on, King et al. question whether it makes sense to base a speech recognizer on a particular phonological theory given that there is so much disagreement in the linguistics literature over what the best phonological theory is. Although there is variation in the feature systems, these differences are not arbitrary. Phonologists agree as to what the desiderata of an ideal feature system are. In short, the perfect feature system will be compact, consist of independent features, and combine naturally with pronunciation mechanisms. In my opinion a sensible PHF set should also satisfy the following criteria.

1. **Distinctiveness**
   All the speech sounds (phonemic or sub-phonemic) must be located at another position in the PHF space.

2. **Detectability**
   It should be possible to extract the PHFs in a reliable way by means of an automatically trained feature mapper. Of course this criterion can only be verified a posteriori by evaluating the measured accuracies.

3. **Unambiguity**
   It should be possible to assign phonological feature values to all the speech sounds one wants to model in an unambiguous way.

I will now discuss one particular feature set i.e. the SPE system and verify whether it meets the three criteria. I will then discuss some disadvantages of this feature set and make some remarks about what should be taken into account when opting for another set.

In *The sound Pattern of English* [19] the authors Noam Chomsky and Morris Halle develop a phonological theory based on the use of binary distinctive features. The goal of their feature theory is to discover the most basic set of fundamental underlying units (the features) from which surface forms (e.g. phones) can be derived. The features can be grouped according to four categories. I use the notation [+feature] or [-feature] for the binary features.

1. **Major class features**
   This category comprises *vocalic* and *consonantal*. Vocalic sounds have a constriction which is less than the one found in /i/ and /u/. On top of that phonation must be on. Non-vocalic sounds do not match one or both of these criteria. [+consonantal] sounds are produced with a clear obstruction in the vocal tract.

2. **Mouth Cavity features**
   These include *coronal*, *anterior*, *high*, *low*, *back*, *round* and *nasal*. [+coronal] sounds are produced with the tongue blade shifted from the neutral position. In [+anterior] sounds the obstruction of the air flow is situated in front of the palato-alveolar region. Labial, dental and alveolar consonants are [+anterior], whereas palato-alveolar, velar or uvular consonants are [-anterior]. Vowels are always [-anterior]. The features [high,low,back,round] are identical to the ones discussed in section 5.2.5. [Nasal] was explained in section 5.2.2.

3. **Manner features**
   Continuant makes the distinction between plosives and non-plosives. *Tense* as opposed to *lax* refers to the intensity of producing the sound. This feature is most obvious with vowels where a tense vowel matches a long vowel

4. **Source features**
   Voiced was already explained. *Strident* means that the air flow is turbulent like in fricatives.

The total number of features is equal to 13. Let us now consider all the American English phones and sub-phonemic units. At this point I also introduce some new phonemic symbol sets that were not in Table 5.2 nor in Figure 5.3. All symbols are summarized in Appendix B, where both the ARPABET symbol, the IPA symbol and the SAMPA symbol is provided. I tried to assign a canonical value (+/-) to every binary feature for all 58 sub-phonemic units (according to the feature definitions in [19]). The result is represented in Table 5.4 (V = vocalic, C = consonantal, H = high, B = back, L = low, A = anterior, Cor = coronal, R = round, T = tense, Vo = voiced, Co = continuant, N = nasal and S = strident).

I am well aware that my interpretation of the SPE-feature set is only one possible interpretation. To illustrate the disagreement among authors concerning what are good SPE-feature values for the phones, I have compared the SPE-feature decomposition in [61] and [34]. I observed several differences, for instance /ow/ was given [-low] in [61], whereas it received [+low] in [34]. A second example: /ae/ received [+tense] in the former

	IPA	V	C	H	B	L	A	Cor	R	T	Vo	Co	N	S
aa	ɒ	+	-	-	+	+	-	-	-	(-)	+	+	-	-
ae	æ	+	-	-	-	+	-	-	-	(-)	+	+	-	-
ah	ʌ	+	-	-	+	-	-	-	-	-	+	+	-	-
ao	ɔː	+	-	-	+	(-)	-	-	+	(-)	+	+	-	-
aw	aʊ	+	-	-	+	+	-	-	+	+	+	+	-	-
ax	ə	+	-	-	+	-	-	-	-	-	+	+	-	-
ax-h	əʰ	+	-	-	+	-	-	-	-	-	+	+	-	-
ax-r	ɚ	+	-	-	-	-	-	-	-	-	+	+	-	-
ay	aɪ	(+)	-	-	+	+	-	-	-	+	+	+	-	-
b	b	-	+	-	-	-	+	-	-	-	+	-	-	-
bcl		-	+	-	-	-	+	-	-	-	-	-	-	-
ch	ʧ	-	+	(+)	(-)	-	-	+	-	-	-	-	-	+
d	d	-	+	(-)	-	-	+	+	-	-	+	-	-	-
dcl		-	+	-	-	-	+	+	-	-	-	-	-	-
dh	ð	-	+	-	-	-	+	+	-	-	+	+	-	-
dx		+	+	-	-	-	+	+	-	-	+	-	-	-
eh	e	(+)	-	-	-	-	-	-	-	-	+	+	-	-
el	ᵊl	(+)	+	-	-	-	+	+	-	-	+	+	-	-
em	ᵊm	-	+	-	-	-	+	-	-	-	+	-	+	-
en	ᵊn	-	+	-	-	-	+	+	-	-	+	-	+	-
eng	ᵊŋ	-	+	(+)	+	-	-	-	-	-	+	-	+	-
er	ɜː	+	-	-	(+)	-	-	-	-	(+)	+	+	-	-
ey	eɪ	(+)	-	-	-	-	-	-	-	+	+	+	-	-
f	f	-	+	-	-	-	+	-	-	-	-	+	-	+
g	ɡ	-	+	(+)	+	-	-	-	-	-	+	(-)	-	-
gcl		-	+	(+)	+	-	-	-	-	-	-	-	-	-
hh	h	-	(-)	-	-	+	-	-	-	-	-	+	-	-
hv	ɦ	(-)	(-)	-	-	+	-	-	-	-	+	+	-	-
ih	ɪ	+	-	(+)	-	-	-	-	-	-	+	+	-	-

ix		+	-	+	-	-	-	-	-	-	+	+	-	-
iy	iː	(+)	-	+	-	-	-	-	-	+	+	+	-	-
jh	ʤ	-	+	+	-	-	-	+	-	-	+	-	-	+
k	k	-	+	+	+	-	-	-	-	-	-	-	-	-
kcl		-	+	+	+	-	-	-	-	-	-	-	-	-
l	l	(+)	+	(-)	-	-	+	+	-	-	+	+	-	-
m	m	-	+	-	-	-	+	-	-	-	+	-	+	-
n	n	-	+	-	-	-	+	(+)	-	-	+	-	+	-
ng	ŋ	-	+	(+)	+	-	-	-	-	-	+	-	+	-
nx		+	+	(-)	-	-	+	+	-	-	+	-	+	-
ow	əʊ	+	-	-	+	-	-	-	+	+	+	+	-	-
oy	ɔɪ	(+)	-	-	+	+	-	-	-	+	+	+	-	-
p	p	-	+	(-)	-	-	+	-	-	-	-	-	-	-
pcl		-	+	-	-	-	+	-	-	-	-	-	-	-
q	ʔ	-	+	-	+	+	-	-	-	-	-	-	-	-
r	r	+	+	-	-	-	-	+	-	-	+	+	-	-
s	s	-	+	-	-	-	+	+	-	-	-	+	-	+
sh	ʃ	-	+	+	-	-	-	+	-	-	-	+	-	+
t	t	-	+	-	-	-	+	+	-	-	-	-	-	-
tcl		-	+	-	-	-	+	+	-	-	-	-	-	-
th	θ	-	+	-	-	-	+	+	-	-	(-)	+	-	-
uh	ʊ	+	-	+	+	-	-	-	(+)	-	+	+	-	-
uw	uː	+	-	+	+	-	-	-	+	+	+	+	-	-
ux		+	-	+	+	-	-	-	+	+	+	+	-	-
v	v	-	+	-	-	-	+	-	-	-	+	+	-	+
w	w	-	(-)	+	(+)	-	-	-	+	-	+	+	-	-
y	j	-	(-)	+	-	-	-	-	-	-	+	+	-	-
z	z	-	+	-	-	-	+	+	-	-	+	+	-	+
zh	ʒ	-	+	+	-	-	-	+	-	-	+	+	-	+

Tab. 5.4: SPE-Feature decomposition of the 58 American English phones (ARPABET notation) and sub-phonemic units. Whenever my feature value was different from the one in [61], I have put it between brackets.

work and [-tense] in the latter. As a last example the /n/ gets [-coronal] in [61] and [+coronal] in [34]. I favour the latter since I think /n/ is pronounced with the tongue blade shifted from the neutral position. I just give these three examples to show that there is no consensus on some of the SPE-features as to what their value should be.

At this stage I can thus remark that this feature set, though theoretically well motivated, does not meet my third criterion of unambiguity of the feature values, and consequently that there are reasons for using another feature set for my research,.

5.3.2 Language dependency

In [109] the authors test the hypothesis that PHFs are language independent. I will make use of this hypothesis for the experiments carried out in section 7.7, where I will do experiments on the recognition of foreign names comprising foreign phonemes with other phonological feature combinations than those that can be seen in the native language. Five languages were selected in [109]: Mandarin Chinese, German, Japanese, English and Spanish. A global feature set composed of 21 features was proposed. For the detection of the features, GMMs were trained in the same way acoustic models are trained for a speech recognizer. Each feature had two models: one for *feature present* and one for *feature absent* which means that the binary encoding paradigm was preferred here too. Training was done on the middle frames of a phonetic segment because the authors had to rely on automatic segmentations. The conclusions of this work were: (1) feature detection is possible even if the language on which the detector is trained differs from the language on which it is tested (cross-lingual). However the performance never is as good as in the monolingual case in which one feature extractor is trained on the same language as the one on which it is tested. (2) When selecting the feature extractor that yielded the highest score in any of a set of language dependent feature extractors, the performance turns out to be higher than in the monolingual case. This suggests that it is possible to detect features on an unknown language given a set of language-dependent extractors. (3) The performance of a multilingual detector, trained on data from several languages is worse than that of a monolingual detector.

5.4 Design of own feature set

In this section I briefly discuss the feature set I have chosen for my research, and the architecture of the feature extractor I have designed for it.

5.4.1 Selection of the feature set

When comparing the multivalued feature sets that were described in the previous section, I noticed that every time a feature value was not relevant, it was given the value ’nil’. This ’nil’ category must be used when consonantal place features and place features only relevant for vowels are grouped in one feature dimension (like in [17]). For a consonantal segment for instance the vowel place features have no meaning and are given the ’nil’ value. I want to avoid such feature values and therefore I propose to combine features which are simultaneously relevant and irrelevant into separate feature dimensions. I decided to discern four feature dimensions:

- **vocal source**: This dimension describes the presence/absence of vocal energy and phonation and it can take the values voiced, unvoiced or no-activation. In this definition the vocal source is presumed to describe the frame-level presence/absence of speech excitation and the nature (voiced/unvoiced) of that excitation.

- **manner**: This dimension corresponds to the manner of articulation. It can take the values closure (first part of a plosive), vowel, fricative, burst (second part of the plosive), nasal, approximant, lateral and silence. A silence is defined as a reasonably long (> 100ms) non-speech time interval in the signal.

- **place-consonant**: Here, the place of articulation features are encoded for consonants. Possible values are labial, labio-dental, dental, alveolar, post-alveolar, velar and glottal.

- **vowel-features**: Finally the place and rounding features for vowels are encoded as low, mid-low, mid-high, high, back, mid, front, retroflex and round.

This PHF definition thus consists of 27 features encoding four feature dimensions. Table 5.5 provides the feature decompositions for the American English phones.

phone	source	manner	place-C	feat-V
aa	voiced	vow.	N	(low,back,off,off)
ae	voiced	vow.	N	(mid-low,front,off,off)
ah	voiced	vow.	N	(mid-low,back,off,off)
ao	voiced	vow.	N	(mid-low,back,off,round)
aw	—			
ax	voiced	vow.	N	(mid-high,mid,off,off)
ax-h	—			
ax-r	—			
ay	—			
b	voiced	burst	labial	N
bcl	n.a.	clos.	0	N
ch	—			
d	voiced	burst	alveol.	N
dcl	n.a.	clos.	0	N
dh	voiced	fric.	dental	N
dx	voiced	burst	alveol.	N
eh	voiced	vow.	N	(mid-high,front,off,off)
el	—			
em	—			
en	—			
eng	—			
er	voiced	vow.	N	(mid-low,mid,retro,off)
ey	—			
f	unvoiced	fric.	lab-dent.	N
g	voiced	burst	velar	N
gcl	n.a.	clos.	0	N
hh	unvoiced	fric.	glottal	N
hv	voiced	fric.	glottal	N
ih	voiced	vow.	N	(mid-high,front,off,off)

ix	voiced	vow.	N	(mid-high,front,off,off)
iy	voiced	vow.	N	(high,front,off,off)
jh	—			
k	unvoiced	burst	velar	N
kcl	n.a.	clos.	0	N
l	voiced	lateral	alveol.	N
m	voiced	nasal	labial	N
n	voiced	nasal	alveol.	N
ng	voiced	nasal	velar	N
nx	voiced	nasal	alveol.	N
ow	—			
oy	—			
p	unvoiced	burst	labial	N
pcl	n.a.	clos.	0	N
q	unvoiced	burst	glottal	N
r (ɹ)	on	approx.	N	(N,N,retro,off)
s	unvoiced	fric.	alveol.	N
sh	unvoiced	fric.	post-alveol.	N
t	unvoiced	burst	alveol.	N
tcl	n.a.	clos.	0	N
th	unvoiced	fric.	dental	N
uh	voiced	vow.	N	(mid-high,back,off,off)
uw	voiced	vow.	N	(high,back,off,round)
ux	voiced	vow.	N	(high,back,off,round)
v	voiced	fric.	labio-dental	N
w	voiced	approx.	labial	(N,N,off,off,round)
y	voiced	approx.	N	(high,N,N,off,off)
z	voiced	fric.	alveol.	N
zh	voiced	fric.	post-alveol.	N

Tab. 5.5: Feature decomposition of American English phones according to the four feature dimensions (n.a.= no activation, off = no retroflex or round, N = not relevant).

It is obvious that the place-consonant features will only be relevant for consonantal segments and not for vowels. Likewise vowel-features carry only relevant information for vowel segments and not for consonants. Liquids like /l/ and /r/ and glides or semi-vowels like /w/ and /j/ can have relevant place-consonant as well as vowel features. Whenever a feature is to be considered as not relevant it is given the value 'N' in the canonical feature decomposition.

A second remark on Table 5.5 is that I did not assign features to phones like /aw/, /ax-h/, etc. because the PHFs of such phones are supposed to be *unstable*.

I think this feature set better meets the third criterion of unambiguity of canonical feature values than the SPE feature set, because the consensus among different authors [50; 17; 61; 70] on elements of this feature set is larger.

5.4.2 Architecture of the feature extractor

In the literature one finds different architectures and different pattern classifiers as building blocks of these architectures. The building blocks can be HMMs, MLPs, SVMs, RNNs and DBNs. Especially DBNs seem to have become popular over the past five years [70; 40; 120], but after having read a number of papers on DBN-based PHF extraction, I came to the conclusion that there is no proof yet of their superiority over the more traditional systems. According to some authors [120], myself included, the DBN may indeed offer an improved modeling capacity, but this capacity cannot be exploited unless one makes use of simple factorized observation functions which can be reliably estimated on the amount of data one can hope to dispose of.

My aim was therefore to base my feature detector on either MLPs or RNNs. To that end I have compared the performances of RNNs and MLPs for the extraction of the SPE features. Since King et al. [61] have published performances of a RNN on the TIMIT corpus, all I had to do was to test a MLP on the same data.

The nonlinear feature mapper consists of a single MLP. For the definitions and technical aspects related to MLPs, I refer to section 2.4. The task of the feature extractor is to give for each frame good estimates of the canonical feature values. During the training of the network(s) I presented successive pairs of acoustic features and the corresponding canonical feature values. For the derivation of these canonical values I used the manual phonetic labels distributed with the TIMIT corpus. Whenever a feature is 'on', I assign 1 to its target value and call this a *positive* feature. If a feature is 'off', I assign 0 and call this a *negative* feature.

The network is provided not only with the current speech frame, but also with *context frames* at both sides of the current frame. The context is always taken symmetric. I performed tests with different context sizes: 1, 3 and 5 frames at both sides. The number of inputs is equal to 39 times the total number of frames (3, 7 or 11). The number of hidden nodes is chosen between 150 and 350 and the number of outputs is equal to the number of SPE features i.e. 13.

I used the fully manually segmented and labeled TIMIT corpus for training and testing of the network. Only the SI and SX sentences were retained. The training data consisted of 3695 utterances and the test data of 1344 utterances. I selected at random 100 utterances for validation, leaving 3596 utterances as effective training data. The number of frames in the training set, the validation set and the test set was 1095041, 29151 and 410711 respectively. Discarding the frames for which there was no unique characterization (affricates, diphthongs) further reduced these numbers by about 14%. All networks had 1 hidden layer and were fully interconnected. During training of the networks the performance on the validation set is monitored and the training stops when this performance reaches a plateau.

The results I obtained are represented in Table 5.6. Evaluation took place on all frames of the test set.

#Cfrs	1		3		5		a priori P	Perf.
#HNodes	150	250	150	250	250	350	(%)	King et al.
#pars	8114	13514	15914	26250	39514	55314		
CF val	0.273	0.279	0.267	0.253	0.258	0.246		
Voc	85.4	83.8	84.9	87.0	86.1	87.8	70.9	88
Cons	86.0	85.5	87.8	88.6	89.0	89.6	52.0	90
H	85.1	84.2	85.4	87.0	86.3	87.8	79.1	86
B	85.1	84.7	85.5	86.8	86.6	87.8	76.7	88
L	91.5	90.7	91.4	92.5	92.0	93.1	86.1	93
A	86.5	86.4	87.8	88.8	88.8	89.8	66.5	90
Cor	87.0	86.9	87.6	88.5	88.4	89.1	74.1	90
R	93.3	93.0	93.2	93.6	93.1	93.8	92.2	94
T	88.1	87.0	88.2	89.5	88.9	90.3	78.6	91
Voic	91.7	91.7	92.2	92.5	92.6	92.9	60.1	93
Cont	90.9	90.7	91.8	92.4	92.6	93.0	62.3	93
N	97.1	97.0	97.3	97.5	97.4	97.6	93.7	97
S	96.0	95.9	96.4	96.6	96.6	96.8	85.5	97
Sil	95.6	95.4	96.8	97.0	97.4	97.6	86.1	98

Tab. 5.6: Frame accuracy (%) for the SPE-feature detection. Different numbers of hidden nodes (HNodes), context frames (Cfrs) and parameters (#pars) were tried out. The value of the cost function measured on the validation set (CF val) is given too.

The evaluation is performed by counting the percentage of frames with the right output: > 0.5 if the feature is supposed to be 1 and ≤ 0.5 if it is supposed to be 0. Together with this I have also listed the prior chance level (a priori P) in Table 5.6. This is defined as $\max(A, 100-A)$ if A stands for the amount of frames (in %) for which the feature is positive. Note that in the case of context 1 increasing the number of hidden nodes does not help. For the longer contexts it does. The Table also shows that my results agree very well with the accuracies obtained by King et al. on the basis of a RNN. Therefore I conclude that the advantage of using a RNN over a MLP is negligible. Note that the accuracies do not improve that much anymore when the context is increased from 3 to 5. Only for the silence feature, the error rate drops from 3.0% to 2.4% which is a reduction by 20%. The Table shows that the frame accuracies are clearly above prior chance level which means that the MLP is able to extract information about the PHFs from the ACFs. Figure 5.4 represents the outputs of the MLP for the first 6 SPE-features. The grey line represents the target feature values that were used during training. The black line is the calculated output of the neural network. From these experiments I concluded to use MLPs instead of RNNs because they are easier to train. Once this choice was made, I conceived a novel architecture for the extraction of my features.

There is evidence [17; 121] that a hierarchical feature extractor can outperform a flat extractor, because features higher in the hierarchy can help to make distinctions between sounds at a lower level. I therefore have conceived a three-layer architecture (see Figure 5.5) that first extracts the phonation features, then the manner features and then the place-consonant or vowel features.

Each block in the Figure represents a feed-forward neural network. The four neural network detectors are fed with the ACFs of the incoming speech and the outputs of the detectors higher in the hierarchy. The vocal source is retrieved directly from the ACFs, the manner features get the vocal source output as a supplementary input and the consonant and vowel feature extraction can benefit from the manner features as well. The vocal source network is provided with an input of three context frames at each side. The goal of this detector is to give an indication of the phonation or the absence of activity in the current speech frame. If there is no activity, the frame is a non-speech frame. The manner network is provided with a more extended context which is obtained by extending it to the range $(t-5, t+5)$ but by encoding the three upper left and upper right frames by their mean (see Figure 5.6). This context extension might be helpful for the discrimination between *closure* and *silence*. Some properties of the networks are summarized in Table 5.7.

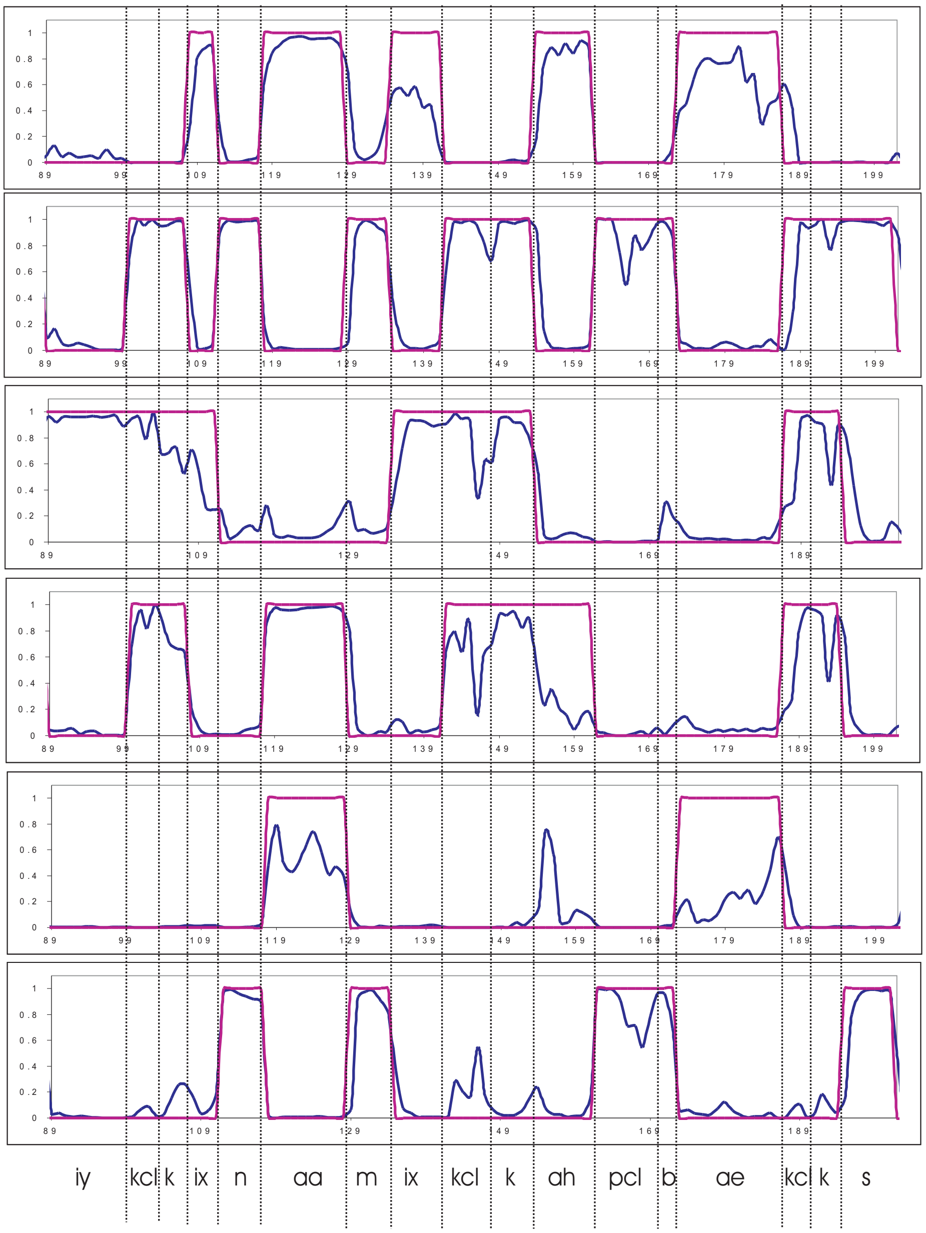


Fig. 5.4: The detected SPE-feature values compared with the canonic feature values (from top to bottom: Voc, Cons, H, B, L en A) for a feed-forward neural network with 150 hidden nodes, receiving a context of three frames. The fragment taken from TIMIT was dr2/FJAS0/SX50 (catastrophic **economic cutbacks** neglect the poor).

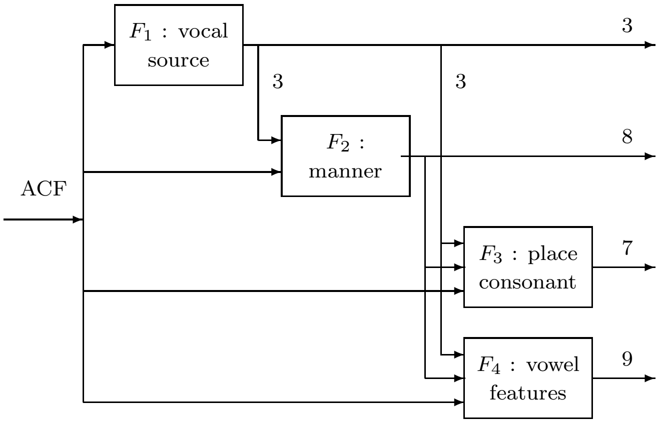


Fig. 5.5: The multivalued phonological feature extractor.

The number of inputs to network F_1 is $7 \times 13 = 91$. Similarly the number of inputs to network F_2 equals $7 \times 3 + 3 = 94$, because of the additional vocal source network outputs. The vocal-source and manner features are all relevant for all sub-phonemic units. The place-consonant detector and the vowel-feature detector only attach relevant features to consonant frames and vowel-frames respectively. Both receive the extended context. For the training I only consider the frames in sub-phonemic segments of which I know for sure what the features are. This means that diphthongs and affricates will not contribute frames to the training. Each network has one output per feature value e.g. the manner network will have eight outputs, and multiple features can be 'on' at a particular time.

During detection it is observed that the feature values can vary continuously between 0 and 1. Moreover, they also vary asynchronously at segment boundaries. During training the non-relevant features are

netw	#inputs	#HNodes	#outputs	#pars
F_1	91	100	3	9503
F_2	94	250	8	25008
F_3	101	250	7	27258
F_4	101	250	9	27760

Tab. 5.7: Properties of the neural networks (#HNodes = number of hidden nodes, #pars = number of weights) used to extract the multivalued features.

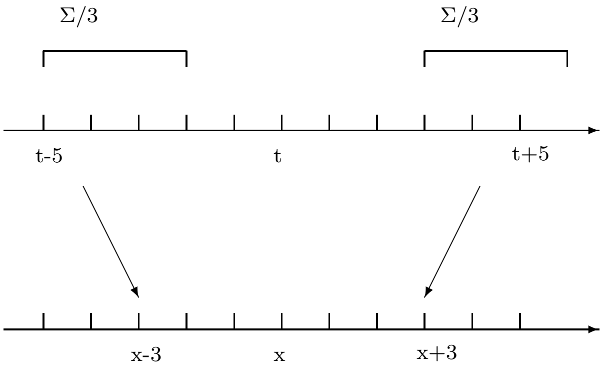


Fig. 5.6: Calculation of the extended context.

treated in a special way. By considering the output of the network as the target for such a feature, there will be no gradient and no update of the weights then.

5.5 Detection Results

In order to train and evaluate the hierarchical feature detector the TIMIT database was used again.

ref	calculated feature							
feat.	clos.	vow.	fric.	burst	nas.	app.	lat.	sil
clos	80.1	2.8	3.1	3.7	2.8	0.2	0.2	7.0
vow.	0.5	94.3	1.0	0.9	1.0	1.2	1.0	0.1
fric.	2.7	4.1	85.9	4.3	1.0	0.1	0.2	1.6
burst.	6.0	10.3	6.8	72.4	1.0	0.7	0.7	1.9
nas.	3.1	10.2	2.1	1.2	80.1	0.43	0.8	2.0
app.	0.7	46.0	2.1	2.1	2.3	41.5	4.7	0.6
lat.	0.3	33.3	2.4	2.0	2.6	1.7	57.0	0.6
sil	3.4	0.6	1.9	2.4	0.7	0.1	0.1	90.7

Tab. 5.8: Frame-wise classification results for the manner-network (confusion matrix) evaluated on 357256 frames.

Evaluation took place on the frames to which relevant features would have been assigned during training. The evaluation of networks F_1, F_2 and F_3 is given by means of confusion matrices. To that end I looked at the

MLP output with the highest value, I considered this as the calculated class and I compared it to the target class. The amount of reference frames from class i that were classified as class j occur on the (i, j)'th position in the confusion matrix. Rows sum up to 100%. The amount (in %) of reference frames which are correctly classified should appear on the diagonal of this matrix. Table 5.8 represents the confusion matrix for the manner network.

ref	calculated feature						
feat.	lab.	ld.	dent.	alv.	pa.	vel.	glot.
lab.	74.9	2.2	1.6	17.2	0.1	2.1	1.8
ld.	3.4	80.7	4.5	8.6	0.7	0.8	1.2
dent.	3.5	11.9	55.3	25.7	0.1	1.2	2.3
alv.	3.3	1.0	0.9	91.5	0.7	1.4	1.1
pa.	0.2	0.7	0.1	16.8	81.3	0.5	0.4
vel.	5.1	0.8	0.7	16.8	0.3	73.4	2.8
glot.	5.5	1.8	1.2	15.8	0.3	3.4	71.8

Tab. 5.9: Frame-wise classification results for the place-consonant-network (confusion-matrix) evaluated on frames belonging to consonant segments but no silences (135676 frames).

Five manner features (closure, vowel, fricative, nasal, silence) could be detected with an accuracy of more than 80 %. However the accuracy for *burst* was only 70%. The features *approximant* and *lateral* were found to be even more difficult to detect. Their high confusion with the vowel manner feature is due to their inherent semi-vowel character. One could argue that approximants like /w/, /j/ or /r/ have a dual nature: they are partly vowels and partly approximants.

The confusion matrix for the place-consonant-network is given in Table 5.9. Again, I have *labio-dental*, *alveolar* and *post-alveolar* reaching 80% or more. Then there is *labial*, *velar* and *glottal* having a frame accuracy rate of about 70-75%. Obviously the *dental* frames are highly confusable with *alveolar* frames, leading to the rather poor accuracy of ca. 55%. A closer look reveals that the English dentals are often confused with the alveolar /t/ or /d/, which is understandable. The (small) confusion matrix for the vocal source network is given in Table 5.10.

Classification of voiced, unvoiced and no-activation frames is clearly something that can be done with an accuracy of more than 83%. Figure 5.7 represents the outputs of the manner network for a short fragment from the TIMIT test data. Note that the phone /en/ did not receive target feature values because this phone was treated as a diphthong.

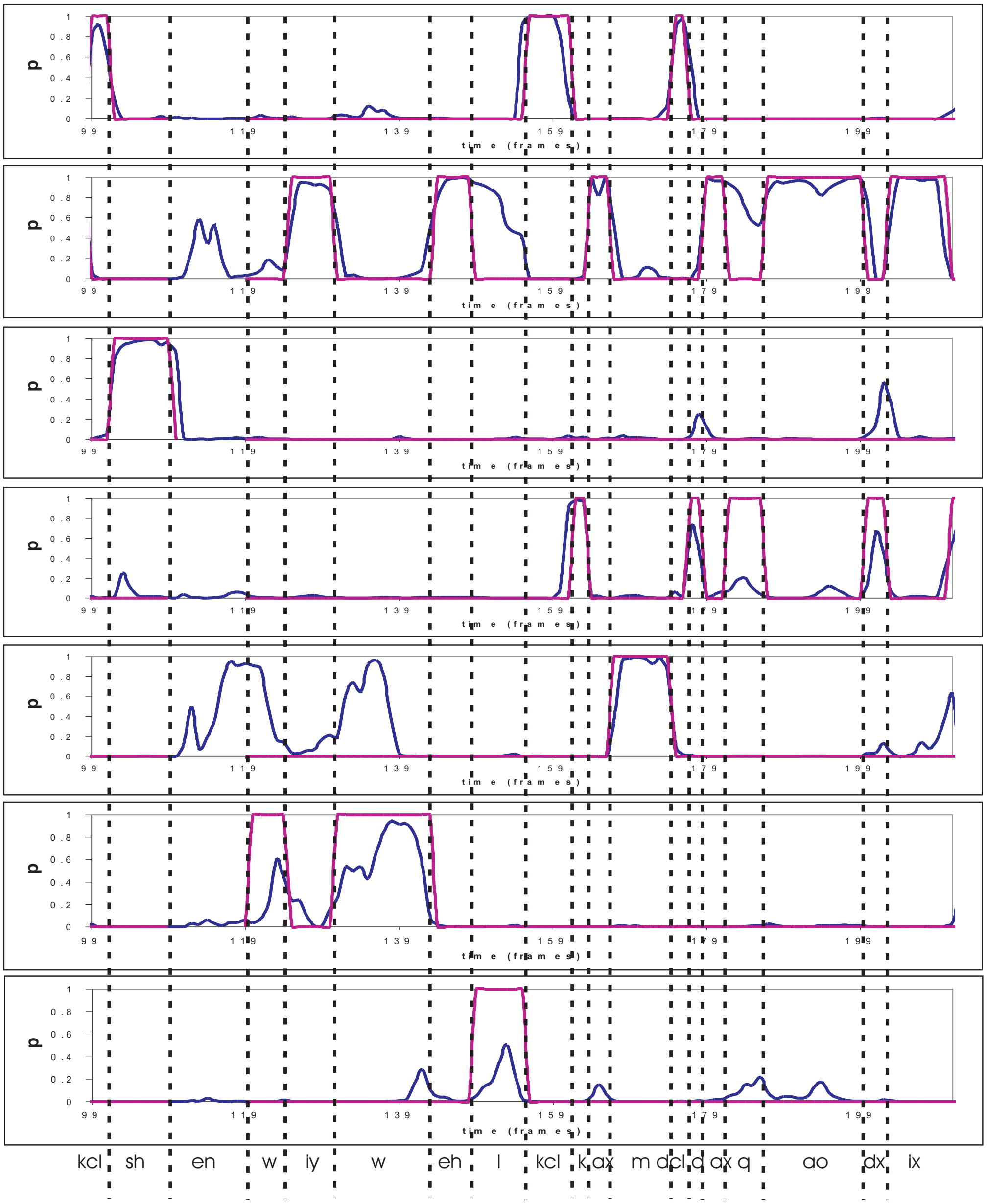


Fig. 5.7: Output of the manner-network (From top to bottom: closure, vowel, fricative, burst, nasal, approximant and lateral) for a feed-forward neural network with 250 hidden nodes, receiving the extended context. The fragment taken from TIMIT was dr2/MWEW0/SI1361 (but in this one se**ction we welcomed audit**ors).

ref	calculated feature		
feat.	on	off	n.a.
on	92.7	5.0	2.3
off	10.5	83.3	6.2
n.a.	4.9	3.1	92.0

Tab. 5.10: Frame-wise classification results for the vocal source network (confusion-matrix, n.a. = no activation).

The frame-level accuracies obtained with this system were comparable with the ones from the literature. In [61] the obtained classification performance for manner of articulation (6 feature values) ranges from 68.6% (my system yields 41.5%) for ’approximant’ to 91.5% (my system yields 94.3%) for ’vowel’. Measured over all classes the manner classification performance was 87% (my system yielded 83.9%) and the place of articulation (10 feature values) was 72% (my system yielded 83.2%). The differences can be explained due to the different numbers of classes involved.

5.6 Conclusion

In this chapter I proposed a hierarchical phonological feature detector. The detector was designed so as to group together features that were simultaneously relevant and irrelevant. Although I based my choice of the feature detector on examples from the literature, this idea of grouping relevant features is a new one proposed by me. The detection performance of my system was found to be comparable with the literature. Moreover, it was shown that MLPs perform almost as well as RNNs for this task. The obtained outputs of the system are easily interpretable as posterior probabilities, something that can be used for further applications.

6 Validation of the Phonological Features

In this chapter I will try to validate the extracted PHFs by conceiving a speech-to-orthography aligner and by showing that the outputs of that aligner are at least as reliable as those obtained with a more traditional aligner which uses triphone HMMs as acoustic models. By way of exploration, I have also demonstrated that the outputs of my aligner can provide a parametric characterization of a speaker that is for instance rich enough to distinguish native from non-native speakers.

6.1 Introduction

There is evidence that PHFs are a good representation for speech segmentation. In [113] the correlations between the manual phone segment boundaries and several distance metrics on the PHF vectors were measured. The cosine-based distance between consecutive pairs of PHF vectors was compared with the manual segment boundaries. A clear relation between the locations of the maxima in the cosine-distance and the boundaries in the manually segmentation was observed. The results suggest that phonetic segment boundaries are associated with local speed along the PHF trajectory.

PHFs should give us more information about the hidden process of speech production than the standard acoustic features do. The study in [60] about the correlation between human and automatic scores for pronunciation proficiency revealed that the ACF likelihood scores were actually only weakly correlated with the human scores. These confidence measures are computed for every phone in the same way, without taking into consideration the specific acoustic-phonetic features of each phone. A study that already makes use of acoustic-phonetic features for the detection of pronunciation errors is described in [114].

6.2 Segmentation and labeling of speech

With the PHF extractor described in the previous chapter and the knowledge of the orthography or the phonemic transcription of the utterance, an acoustic-phonetic labeling and segmentation of the speech frames can be performed. By performing a label-by-label analysis of the phonological features it is then possible to score the pronunciations of non-native speakers.

6.2.1 System architecture

In order to label and segment (align) the speech utterance, I must first define what the target label set is. I propose to use the phones or sub-phonemic units as the basic labels (see Appendix B). The two inputs to a segmentation and labeling system are (1) a sequence of ACF vectors and (2) a linguistic model of the utterance derived from the orthography and phonetic knowledge.

Two options are considered for the linguistic input: LING1 is the sequence of sub-phonemic units as it can be derived from the orthography by means of a pronunciation lexicon and knowledge about (1) coarticulations between words and (2) the sub-phonemic structure of the phonemes. LING2 is the sequence of manually annotated phones. The linguistic model is represented in Figure 6.1.

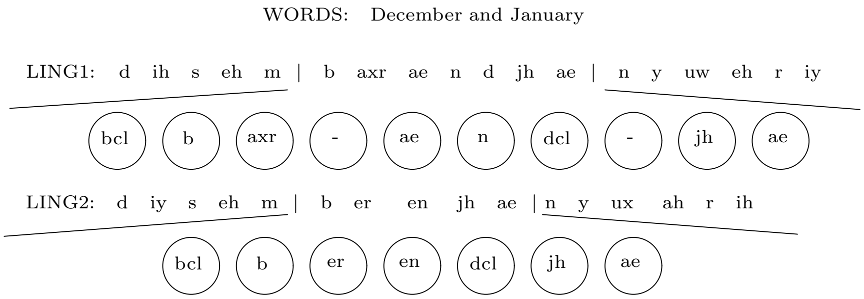

WORDS: December and January
LING1: d ih s eh m | b axr ae n d jh ae | n y uw eh r iy
bcl
b
axr
-
ae
n
dcl
-
jh
ae
LING2: d iy s eh m | b er en jh ae | n y ux ah r ih
bcl
b
er
en
dcl
jh
ae

Fig. 6.1: The two linguistic models. LING1 uses the orthography and a pronunciation lexicon to construct the phonemic transcription. LING2 uses the manual phonetic transcription and converts this to the phonemic labels.

Assume that the lexicon that was used for LING1 contained the following two entries

```
December   d ih s eh m b axr

January   jh ae n y uw eh r iy
```

Then the example given in the Figure demonstrates the knowledge based decomposition of plosives into the sub-phonemic closure and burst units (see /b/ e.g.). Also the affricate /jh/ was preceded by a closure unit /dcl/. The linguistic model LING1 automatically inserts silent pauses between each two words whereas the LING2 model only copies annotated pauses.

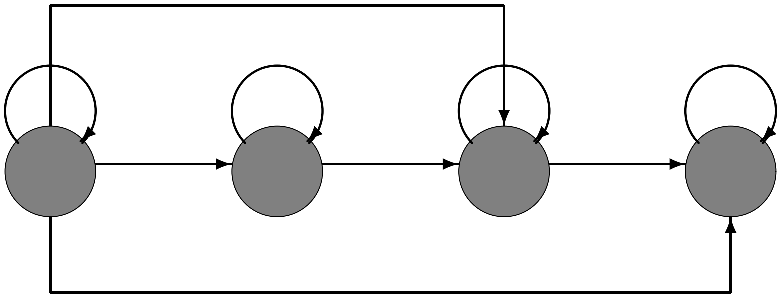


Fig. 6.2: Possible skip transitions implemented in the automaton, illustrated for first state only.

Based on this linguistic input I construct a linguistic model with one state per sub-phonemic unit. In this automaton (Figure 6.2) I also include skip transitions in order to cope with possible deletions of sub-phonemic units. In the case of LING2, it is a linear stochastic automaton. In the case of LING1, there can be words having different pronunciations. All these pronunciations are put in parallel in the stochastic automaton so that each of them can contribute to the best path.

The over-all system architecture of the aligner I constructed is depicted in Figure 6.3. The output of the system is a sequence of (boundary,label)-pairs: Each label (p_k) has a corresponding starting time (t_k). The block "PHF detect" corresponds to the hierarchical PHF extractor described in the previous chapter.

A Viterbi algorithm aligns the $\mathbf{y}_t$'s with the linguistic model, but in such a way that the joint probability $P(\mathbf{X}, S)$ is maximized. To this end a model is needed that fixes the relation between the $\mathbf{x}_t$, the $\mathbf{y}_t$ and the states s_t of the linguistic model. I have investigated two such models. In a

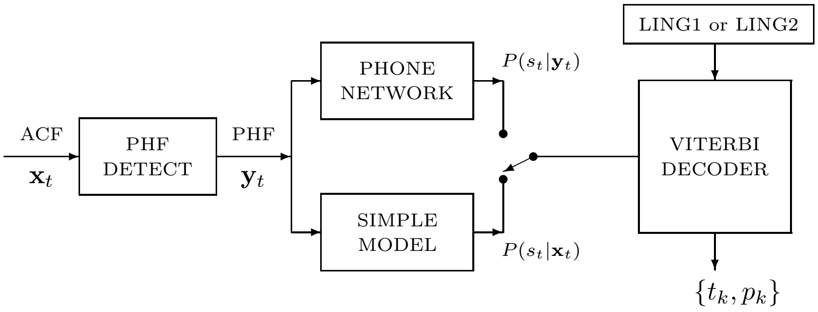


Fig. 6.3: System architecture of the PHF-based automatic aligner

first variant a neural network converts the PHFs to posterior probabilities $P(s_t|\mathbf{y}_t)$. This is denoted by the upper "phone network" block in the Figure. In a second the Viterbi decoder works directly on the PHFs via a simple model for converting PHFs to posterior probs $P(s_t|\mathbf{x}_t)$. This simple model is represented by the lower block in Figure 6.3.

6.2.2 Modeling $P(\mathbf{X}, S)$

If $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ is the ACF sequence the aligner is receiving at its input and if $S = \{s_1, \ldots, s_T\}$ is a legal path through the linguistic model, then the Viterbi decoder searches for the state sequence S maximizing the joint probability $P(\mathbf{X}, S)$. Relying on the Markov hypothesis, the latter can be factorized as

$$\begin{aligned} P(\mathbf{X}, S) &= \prod_t P(s_t, \mathbf{x}_t|s_{t-1}) \\ &= \prod_t P(\mathbf{x}_t|s_t)P(s_t|s_{t-1}) \\ &= \prod_t \frac{P(s_t|\mathbf{x}_t)P(\mathbf{x}_t)}{P(s_t)} P(s_t|s_{t-1}) \\ &= \prod_t \frac{P(s_t|\mathbf{x}_t)P(s_t|s_{t-1})}{P(s_t)} \prod_t P(\mathbf{x}_t) \end{aligned} \tag{6.1}$$

Finally the prior observation probability in the right hand side is the same for all states, hence it suffices to maximize.

$$Q(\mathbf{X}, S) = \prod_t P(s_t|\mathbf{x}_t) \frac{P(s_t|s_{t-1})}{P(s_t)} \tag{6.2}$$

The probability $P(s_t|s_{t-1})$ represents a transition probability and can be regarded as a parameter of the system. The probabilities $P(s_t)$ and $P(s_t|\mathbf{x}_t)$ are the prior and posterior probabilities of being in state s_t at time t respectively. The calculation of the latter in terms of the extracted PHFs $\mathbf{y}_t$ will depend on the system variant.

Variant 1 (phone network) In this variant it is assumed that $P(s_t|\mathbf{x}_t) = P(s_t|\mathbf{y}_t(\mathbf{x}_t))$. Therefore the function to be maximized is then

$$Q(\mathbf{X}, S) = \prod_t P(s_t|\mathbf{y}_t(\mathbf{x}_t)) \frac{P(s_t|s_{t-1})}{P(s_t)} \tag{6.3}$$

In this variant, a phone network (an MLP) is trained to estimate the $P(s_t|\mathbf{y}_t)$. The properties of this phone network are summarized in Table 6.1. The desired $P(s_t|\mathbf{y}_t)$ are then substituted by the phone network outputs. A similar architecture was proposed in [18] for the automatic transcription of spontaneous American English.

#inputs	175
#HNodes	400
#outputs	61
#pars	94861
#Cfrs	3
#epochs	62

Tab. 6.1: Properties of the phone network (epochs = number of training epochs).

Variant 2 (simple model) In the second approach, $P(s_t|\mathbf{x}_t)$ is derived by means of a predefined model that specifies how the $\mathbf{y}_t$ have to be invoked, given the phonological characterization of state s_t.

This predefined model relies on the PHF characterizations of the states. Given the phonological description of state q, the feature set can be divided into three subsets: P_q = the set of *positive* features that are supposed to be *on*, N_q = the set of negative features that are supposed to be *off* and I_q the set of features that are irrelevant for this state. The i-th component of $\mathbf{y}_t$ will be denoted as y_{ti} and the canonical features of state q as f_{qi}. Next, N_{qp} is the number of positive features, N_{qn} is the number of negative features and N_{qi} the number of irrelevant features for state q. Since state q is characterized by the sets of positive, negative and irrelevant features (P_q, N_q, I_q), I can write

$$\begin{aligned} P(q|\mathbf{x}) &= P(P_q, N_q, I_q|\mathbf{x}) \\ &= P(I_q|\mathbf{x})\ P(N_q|I_q, \mathbf{x})\ P(P_q|N_q, I_q, \mathbf{x}) \end{aligned} \tag{6.4}$$

Assuming independent phonological features then leads to the following expression for $P(q|\mathbf{x})$

$$\begin{aligned} P(q|\mathbf{x}) &= P(P_q|\mathbf{x})\ P(N_q|\mathbf{x})\ P(I_q|\mathbf{x}) \\ &= P(\bigcup_{f_{qi} \in P_q} f_{qi}|\mathbf{x})\ P(\bigcup_{f_{qi} \in N_q} \overline{f}_{qi}|\mathbf{x})\ P(\bigcup_{f_{qi} \in I_q} f_{qi}|\mathbf{x}) \\ &= \prod_{f_{qi} \in P_q} P(f_{qi}|\mathbf{x}) \prod_{f_{qi} \in N_q} [1 - P(f_{qi}|\mathbf{x})] \prod_{f_{qi} \in I_q} P(f_{qi}|\mathbf{x}) \end{aligned} \tag{6.5}$$

The last product in this expression is over all irrelevant features. Now it happens that the irrelevant features are defined as features that do not carry any information about state q. The definition of the mutual information between the ACFs $\mathbf{x}$ and the irrelevant features I_q of state q in terms of the difference between the entropy and the conditional entropy is

$$I(I_q, \mathbf{x}) = H(I_q) - H(I_q|\mathbf{x}) \tag{6.6}$$

I assume that this mutual information should be zero for irrelevant features. This gives the condition

$$\frac{H(I_q|\mathbf{x})}{H(I_q)} = 1 \tag{6.7}$$

Now the definitions of $H(I_q)$ and $H(I_q|\mathbf{x})$ are given by

$$H(I_q) = -E[\log P(I_q)] \tag{6.8}$$

$$H(I_q|\mathbf{x}) = -E[\log P(I_q|\mathbf{x})] \tag{6.9}$$

Assume that

$$\frac{P(I_q|\mathbf{x})}{P(I_q)} = C(I_q, \mathbf{x}) \tag{6.10}$$

then $E[\log C(I_q, \mathbf{x})]$ must be zero in order to fulfill condition (6.7). From (6.2) it follows that the cost function used during the Viterbi algorithm is

$$\log Q(\mathbf{X}, S) = \sum_t \log \frac{P(s_t|\mathbf{x}_t)}{P(s_t)} + \sum_t \log P(s_t|s_{t-1}) \tag{6.11}$$

The contribution of the irrelevant features to the cost according to (6.5) will then be given by

$$\sum_{t=1}^{T} \log \frac{P(I_{s_t}|\mathbf{x}_t)}{P(I_{s_t})} \approx T\,E[\log \frac{P(I_q|\mathbf{x})}{P(I_q)}] = 0 \tag{6.12}$$

if I consider the summation as an approximation of the mean value. Hence, I can discard this contribution and only consider the contributions of the relevant (=positive and negative) features to the cost by omitting the third factor in expression (6.5).

For sub-phonemic units or phones like diphthongs and affricates that are actually representing a transition from one phone (with state q_{head}) to another (with state q_{tail}), I did not assign positive features during training the PHF extractor. For states belonging to such phones I chose to evaluate the expression $P(q|\mathbf{x}_t)$ as,

$$P(q|\mathbf{x}_t) = \max(P(q_{head}|\mathbf{x}_t), P(q_{tail}|\mathbf{x}_t)) \tag{6.13}$$

An alternative would have been to use two consecutive states q_{head} and q_{tail} for such phones and to assign the respective feature decompositions to these states. All phones for which the (q_{head}, q_{tail}) model is used, are summarized in Table 6.2.

q	q_{head}	q_{tail}	q	q_{head}	q_{tail}
aw	aa	uh	en	ax	n
ax-h	ax	hh	eng	ax	ng
axr	ax	r	ey	eh	ih
ay	aa	ih	jh	d	zh
ch	t	sh	ow	ax	uh
el	ax	l	oy	ao	ih
em	ax	m			

Tab. 6.2: Decomposition of the composed phones.

For the calculation of the prior probability $P(q)$ I must proceed in the same way: remove $\mathbf{x}_t$ from $P(q|\mathbf{x}_t)$ to get,

$$P(q) = \prod_{f_{qi} \in P_q} P(f_{qi}) \prod_{f_{qi} \in N_q} [1 - P(f_{qi})] \tag{6.14}$$

with $P(f_{qi})$ the prior probability of feature f_{qi} over the entire corpus. This was measured as the percentage of frames having feature f_{qi} divided by the total number of frames in the corpus. Finally the transition probabilities are defined as,

$$P(q+j|q) = \begin{cases} 0.38 & j = 0 \text{ or } 1 \\ 0.2 & j = 2 \\ 0.04 & j = 3 \end{cases}$$

(It is acceptable that the transition probabilities do not exactly add up to 1)

6.3 Experiments

In this section I will evaluate the accuracy of the PHF-based segmenter and labeler. In order to do so I will first have to define the evaluation metric.

6.3.1 Evaluation metric and experimental setup

The segmentation and labeling experiments were carried out on the TIMIT core test set (see section 2.5.1). The set consists of 192 sentences (24 speakers times 8 sentences). The lexicon (for building model LING1) was the baseline TIMIT lexicon with one canonical pronunciation per word. The pronunciations are described in terms of 48 out of the 58 TIMIT phone unit symbols. This means that, apart from the subphonemic units like the six closure units, some allophones like /ax-h/, /eng/, /hv/, /nx/ (flap n), /ux/ (fronted /uw/) and /dx/ (flap d) were merged with other units. Also the glottal stop /q/ is ignored. For an appropriate evaluation, all labels were mapped to one of the 48 symbols. The same symbol set was also adopted in [66] for evaluating automatic segmentation and labeling results.

During evaluation the automatic segmentation and labeling is aligned with the manual one using the Dynamic Time Warping (DTW) procedure described in [76]. Due to the applied cost function, "gross" errors (defined as automatic segments having no overlap with their corresponding manual segments) are very unlikely to emerge from this alignment. Therefore, I distinguish three kinds of segmentation errors: deletions ("del", a manual segment boundary was omitted), insertions ("ins", an automatic boundary was inserted between two manual boundaries) and boundary deviations ("far", the placement of the automatic and the corresponding manual boundary differs by more than 20 ms). The same boundary deviation error criterion is also used by others (e.g. [22]). The labeling errors ("sub") indicate the number of times that an automatically labeled phone differs from the manual one to which it was assigned by the DTW-process. The total error is the sum of the segmentation and labeling errors. All errors are specified in percent, relative to the number of phones occurring in the manual labelings.

6.3.2 Segmentation and labeling results

The error rates for the two system variants and the two types of linguistic input are listed in Table 6.3. Apparently, the alignment is much more reliable when a manual phonetic transcription is available (LING2). The number of substitutions is much lower in this case because the manual transcription already contains the right phone sequence. The extra phone network does not significantly outperform the simple model (there is only a small improvement when starting from a manual phonetic transcription). The associated cost of using the phone network for this segmentation task clearly outweighs the benefits.

system variant	linguistic input	err (%)	del (%)	ins (%)	far (%)	sub (%)
simple	LING1	39.6	10.3	8.3	5.8	15.1
model	LING2	24.2	7.5	6.8	6.3	3.5
phone	LING1	40.0	10.8	7.7	7.3	14.2
network	LING2	22.2	7.1	4.9	6.9	3.3

Tab. 6.3: Evaluation of segmentation and labeling for two systems and two types of linguistic input (core test set, 48 phonetic units)

I first compared my alignment results with the ones formerly obtained on the same data by Vorstermans et al. using an aligner based on stochastic segment models [119]. These results are marked as V in Table 6.4. The linguistic model was derived from LING1.

system	err (%)	del (%)	ins (%)	far (%)	sub (%)
V	38.5	6.4	3.8	12.1	16.2
B	56.6	2.7	11.3	28.7	13.9

Tab. 6.4: Evaluation of segmentation and labeling for two systems described in the literature (core test set, 48 phonetic units)

Another system I compared with is that of Brugnara et al. [15] (system

B in Table 6.4). The latter is a more traditional aligner based on HMMs. A comparison with the more recent work of Aversano et al. [8] is not that straightforward since it considers phoneme segmentation only. In order to make an attempt I introduce two new performance measures. Let S_t be the total number of "true" segmentation points in the test data and S_d the total number of segmentation points detected by my algorithm, then $D = S_d - S_t$ is a useful measure of over-segmentation. In the ideal situation S_d should be equal to S_t. An alternative measure of over-segmentation is $D' = 100\ (S_d/S_t - 1)$. A second quality measure is

$$P_c = \frac{\#\text{correctly detected segmentation points}}{\#\text{"true" segmentation points}} \tag{6.15}$$

In this work a phoneme boundary is defined as "correctly detected" if its distance from the true segmentation point is within 20 ms. When D is forced to be 0 (achieved by tuning the control parameters of the aligner), then the system of [8] yields a P_c of 73.58%. This result should be compared with the sum of "far" errors and deletions made by my system, but in the situation in which the number of deletions and insertions are equal to each other. From Table 6.3 it can be seen that there are about 2% more deletions than insertions which means that my D is equal to minus 2%. From the results in [8] it appears that for a certain degree of undersegmentation, P_c is even lower than 73.58%, whereas my P_c is about 83%.

For comparison I have also constructed a state-of-the-art HMM aligner with the context-dependent triphone models provided by ESAT. Such a system performs a segmentation into phonemes and therefore the evaluation has to be performed at the phoneme level as well (42 phonemes). In order to derive phoneme segments from my system (which uses sub-phonemic units), I concatenate the sub-phonemic units who constitute a single phoneme (e.g. closure + burst = plosive). The TIMIT phone references were processed in the same way. Table 6.5 shows that my system provides state-of-the-art segmentation and labeling performances, and

system variant	linguistic input	err (%)	del (%)	ins (%)	far (%)	sub (%)
simple model	LING1	42.1	11.9	9.2	7.3	13.7
	LING2	28.1	8.6	7.9	8.0	3.6
HMM system	LING1	48.2	8.8	12.3	12.7	14.4
	LING2	32.9	8.0	8.8	12.8	3.3

Tab. 6.5: Comparison of my aligners with an HMM-based system (core test set, 42 phonemes).

thus that it constitutes a good starting position for the assessment of articulation/pronunciation proficiency scores. The reason why my system yields better segmentation and labeling than the HMM system is probably due to the fact that the PHF extractor was trained with the manually segmented TIMIT data, whereas the ESAT system was trained in an embedded mode on other corpora. Since I artificially concatenated sub-phonemic units into phonemes, the phoneme boundaries may be argued as well. Especially for plosives it is arguable whether the plosive starts at the onset of the closure interval, since this interval does not contain any signal. The HMM aligner will probably segment plosives more according to the burst.

6.4 Scoring the pronunciation of non-native speakers

An alternative way of validating the outputs of an aligner is by investigating whether these outputs can be used as a basis for the construction of an interesting characterization of the speech of a particular speaker. Such a characterization is for instance important for the intelligibility assessment of pathological speakers and for the pronunciation proficiency assessment of non-native speakers learning the target language as a second language.

In this section I describe an exploratory experiment I conducted in order to investigate how well a certain set of speaker features derived from my aligner (ELIS) and from the HMM aligner (ESAT) can predict whether a certain speaker is a native or a non-native speaker of the language. If the two aligners would offer segmentations of a comparable quality, the native/non-native classification on the basis of the speaker features derived thereof should be comparable as well.

6.4.1 Definition of speaker features

Since the aligner is supposed to know what the speaker has said, the differences between native and non-native speakers have to originate mainly from differences in their so-called pronunciation proficiency. Therefore, the speaker features I am going to define are in a way aimed to tell something about the goodness of pronunciation of certain sounds. Consequently, these features are called GOP-features.

Once the segmentation and labeling is performed, each frame $\mathbf{x}_t$ of the utterance is associated to an acoustic model state s_t, and through this state, to a phone (ELIS system) or a phoneme (ESAT system) p. Furthermore, each state s_t has an associated set of canonical phonological features $f_i(s_t)$ which can be 1 or 0. It is thus possible to define at least three candidate GOP feature sets:

1. **Phoneme proficiencies**
   The phoneme proficiency $\mathrm{GOP}(p)$ of phone/phoneme p is calculated as the mean of $\log P(s_t|\mathbf{x}_t)$ over all frames that were assigned to a state that belongs to an acoustic model of p.
   $$\mathrm{GOP}(p) = \mathrm{E}_{t,s_t \in p}[\log P(s_t|\mathbf{x}_t)] \tag{6.16}$$
   The computation of $P(s_t|\mathbf{x}_t)$ does of course depend on the aligner that was used. For my aligner, this information can be found in Section 6.2.2, for the ESAT aligner, the posterior probabilities are derived from the likelihoods $p(\mathbf{x}_t|s_t)$:
   $$P(s_t|\mathbf{x}_t) \sim \frac{p(\mathbf{x}_t|s_t)P(s_t)}{\sum_q p(\mathbf{x}_t|q)P(q)} \tag{6.17}$$
   with the sum taken over all possible acoustic model states q.

2. **Feature proficiencies**
   The phonological feature proficiency $\mathrm{GOP}(i)$ of feature i is calculated as the mean of $\log P(f_i|\mathbf{x}_t)$ over all states:
   $$\mathrm{GOP}(i) = \mathrm{E}_t[\log P(f_i|\mathbf{x}_t)] \tag{6.18}$$
   with
   $$P(f_i|\mathbf{x}_t) = \begin{cases} y_{ti} & \text{if } f_i(s_t) = 1 \\ 1 - y_{ti} & \text{if } f_i(s_t) = 0 \end{cases}$$
   These features can only be computed in the context of my own aligner.

3. **Phonological class proficiencies**
   Instead of taking an average over the states belonging to a model of some phone/phoneme p, one can also take an average over the states which have an associated feature $f_c(s_t) = 1$ (c=1,..,25).
   $$\mathrm{GOP}(c) = \mathrm{E}_{t,s_t|f_c(s_t)=1}[\log P(s_t|\mathbf{x}_t)] \tag{6.19}$$
   The selected states implicitly define a phonological class of phones/-phonemes, and that is why the features derived in this way are called phonological class proficiencies.

In the next section I describe how exactly these features were applied for the native/non-native classification of speakers.

6.4.2 Native/non-native classification

The native/non-native classification of a speaker is based on an analysis of all the utterances that are available of that speaker. The classifier is actually a regression model, and this model is given access to one of the feature sets defined in the previous section. Since my experiment considers only 50 speakers, namely 40 native speakers from diverse test and development sets of the WSJ corpus and 10 non-native speakers from the Spoke-3 evaluation set of that same corpus, the regression model had to be a very simple model with not much more than 5 degrees of freedom. I therefore opted for linear regression models in combination with feature subset selection. The latter means that the linear model only takes a few of the available input features into account. If N features are selected, the model can be expressed in terms of $N + 1$ free parameters (the N features and a bias term).

In the meantime, it has been demonstrated [81] that such simple models can accurately predict the intelligibility of dysarthric speakers, but I did not know this at the time I conducted my validation experiments.

6.4.3 Experiments

I have evaluated five feature sets. Three feature sets GOP(p), GOP(i) and GOP(c) were derived from the alignments made by my own aligner, and two feature sets $\text{GOP}_E(p)$ and $\text{GOP}_E(c)$ were derived from the alignments made by the ESAT aligner. The speakers are divided into a native subset S_{nat} and non-native subset S_{non}. The non-native speakers are grouped according to their L1 (mother tongue) classes in Table 6.6.
The quality of a regression model is expressed in terms of two quality measures:

1. The Root Means Square Error (RMSE) between the regression model outputs (y_k) and the target outputs (1 = non-native, 0 = native).

2. The relative classifier margin (RCM) defined as the margin between (1) the minimum y_k that was produced for any of the non-native speakers and (2) the maximum y_k that was produced for any of the native speakers, but divided by the standard deviation of the y_k values found for all the speakers.

group	speaker	L1	native country
S	4nd	Spanish	Argentina
	4nh	Spanish	Israel
	4nm	Spanish	Nicaragua
F	4ne	French	France
	4nf	French	France
D	4ni	Danish	Denmark
	4nl	German	Germany
	4nj	Hebrew	Israel
	4nk	Japanese	Japan
	4nn	British English	Britain

Tab. 6.6: Non-native speakers and their mother tongue (L1) from the S3 subset of the WSJ corpus.

The classification is perfect as soon as the RCM is positive. However, the larger the RCM is the more chance there is that unseen speakers would also be classified in the right class.

Table 6.7 shows the RMSE and the RCM for five models (supplied with different feature sets) as a function of their complexity (1 to 5 selected features). The best performing features according to the RMSE criterion were selected. The target outputs and the calculated outputs emerging from the three of the five models that each selected four input features are depicted in Figure 6.4. The main results of the experiment can be summarized as follows:

1. The phoneme features derived from the two aligners both permit to achieve a perfect classification with a very comparable margin.

2. The phonological features are not as powerful as the other features for making a native/non-native classification.

3. The phonological class features derived from the two aligners both permit to achieve a perfect classification with a very comparable margin, and this margin is even larger than that observed for the phoneme features.

The first conclusion is that the experiment seems to indicate that the alignments produced by the two aligners are of a very comparable quality. This is a nice result in the sense that my aligner is a much simpler system than the ESAT aligner: it just uses 48 phone models whereas the ESAT system uses a set of a few thousand triphone models.

selected	RMSE				
features	GOP(i)	GOP(p)	GOP$_E$(p)	GOP(c)	GOP$_E$(c)
1	0.296	0.116	0.080	0.116	0.080
2	0.267	0.094	0.051	0.082	0.048
3	0.239	0.078	0.038	0.071	0.035
4	0.216	0.063	0.033	0.055	0.029
5	0.204	0.053	0.026	0.048	0.027
selected	RCM				
features	GOP(i)	GOP(p)	GOP$_E$(p)	GOP(c)	GOP$_E$(c)
1	-2.197	1.624	1.616	1.624	1.616
2	-0.941	1.816	1.888	1.799	1.955
3	-0.519	1.743	2.141	1.929	2.135
4	-0.202	2.045	2.114	2.121	2.245
5	0.394	1.971	2.183	2.133	2.229

Tab. 6.7: RMSE and RCM for different models (characterized by a different feature set) and model complexities (characterized by the number of selected features).

The second conclusion is that the phonological features are not very powerful for native/non-native discrimination. This is in line with my expectation that non-natives are bound to have more problems with getting all the features of one particular phoneme right than with getting one phonological feature right in all the phonemes. I do expect however (and in the meantime this is confirmed by evidence [81]) that the problems of pathological speakers on the other hand are more related to the phonological dimensions, and that the phonological features play their part in the assessment of such speakers.

6.5 Conclusions

I proposed a novel segmentation and labeling system that makes use of PHFs. This system was tested with two kinds of linguistic inputs: (1) a phonemic transcription derived from the orthography and a pronunciation lexicon an (2) a phonemic transcription derived from the sequence of manually annotated phonemes.

The system was implemented in two variants: (1) one that makes use of a phone MLP to convert the phonological features to phone posterior probabilities and (2) one that uses a simple knowledge based model to make that conversion. I found that adding the extra phone network did not improve the segmentation and labeling results very much, and thus, that the simple model is sufficient for the segmentation and labeling task.

I then compared my results with an HMM-based segmentation and labeling system and with some systems from the literature and found that my system performs equally well or better than these other systems. In order to make a fair comparison between my system and the HMM-based system, I also investigated the capacity of the two systems to create a segmental context from which one can derive interesting features for describing the pronunciation proficiency of a speaker. I found that both segmental contexts were equally suitable for providing features that permit a good native/non-native classification of a speaker.

All-together, the results provided thus far demonstrate that a state-of-the-art segmentation and labeling performance can be obtained by means of a system using PHFs as an intermediate representation of the speech sounds. The great advantage of my system resides in its simplicity compared to a HMM-based alignment system.

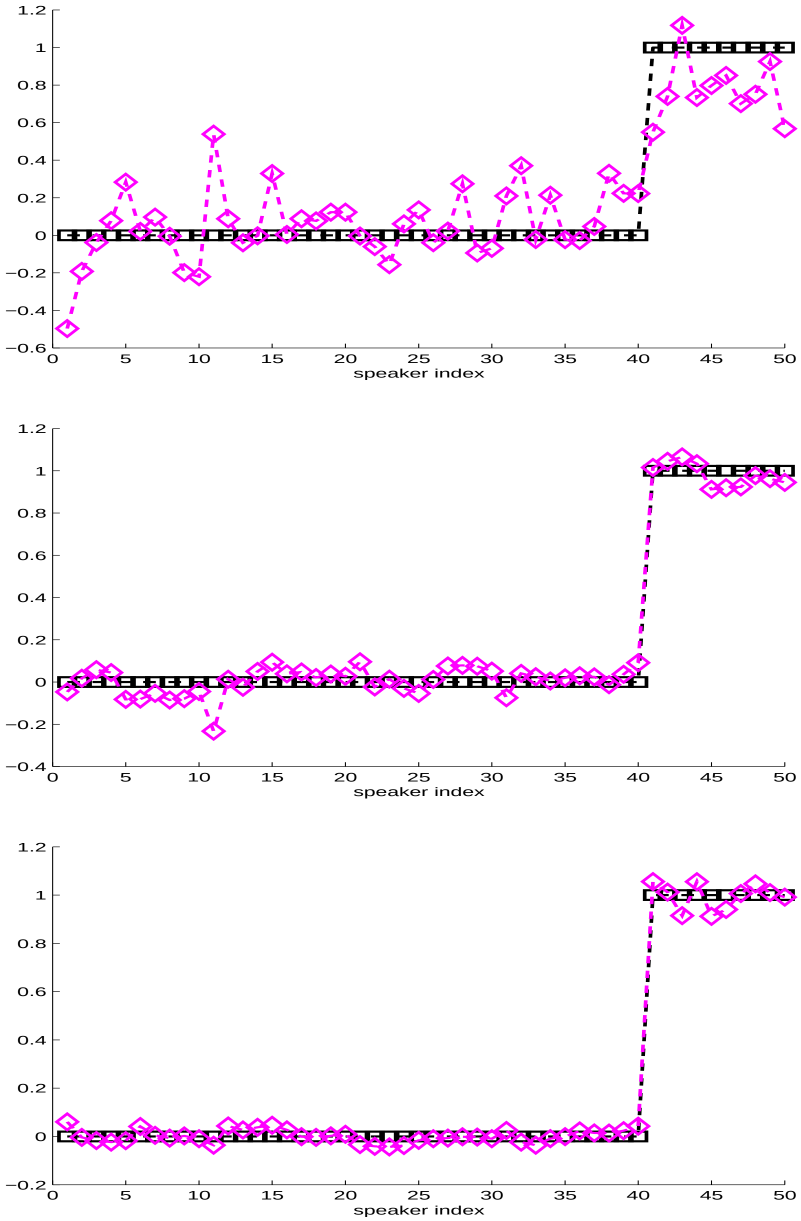


Fig. 6.4: Target (0 = native, 1 = non-native) values and calculated outputs of the models GOP(i), GOP(p) and GOP$_E$(p) for all the 50 speakers

7
ASR with PHFs

Now that the PHF-detector has been validated, it is time to explore its potential in the context of ASR. Some authors [62] have argued that phonological or articulatory features can be beneficial for automatic speech recognition e.g. because they provide a more convenient interface to the higher-level components of the ASR-system. Former research has not yet demonstrated that a purely PHF-based system can outperform a traditional system working with MFCCs as acoustic features (ACFs). However, there is proof already that a combination of the two feature types can lead to improved ASR. In this chapter I will propose a novel method for performing ASR on the basis of PHFs. I will first propose to take account of (1) the correlations between the PHFs and (2) the fact that not all PHFs are relevant for the description of a certain phonetic unit. Then I will investigate to what extent an ACF and a PHF driven ASR-system make different errors. Based on my findings, I will test two methods for combining PHFs and ACFs in one recognizer. One of these combination methods will be applied to a spoken name recognition task which involves a lot of words of a foreign origin.

7.1 Overview of the literature

The use of phonological features (PHFs) for ASR has been studied for more than a decade now. The main reasons for using such features are:

- They constitute an intermediate level between the raw acoustic observations (e.g. MFCCs) and the phonemes. Perhaps they represent the highest information level that can still be extracted reliably from the speech signal. The same feature values typically occur in more than one phone and more than one language. The available training material can thus be shared across phones and languages and may form a solid basis for multilingual and cross-lingual ASR. When using ACFs, all observations originate from the same phone and there is no easy method for extrapolating to non-native phones or foreign phones. When using PHFs, however, different phones contribute to the observations of one PHF, which possibly leads to more robustness. Moreover, PHFs offer means of flexible extrapolation to non-native phones or foreign phones.

- PHFs offer a sound basis for the lexical representation of words in a lexicon, and for the description of likely pronunciation variations [65]. Pronunciation variation can be described in terms of feature overlap or feature assimilation rather than in terms of phone substitutions, deletions or insertions.

- PHFs carry information with respect to highly context-dependent aspects of speech sounds. They therefore constitute an interesting framework for describing coarticulation phenomena in speech. In fact this aspect is already implemented in standard recognizers because the questions that control the state-tying during acoustic model training are phonologically inspired.

Obviously, the need for separate stochastic models to extract the PHFs adds complexity to the over-all recognition system, and the question is of course whether this additional effort is justified. Most of the work on PHF-based ASR has focused on phoneme recognition [35; 21; 27] or on small vocabulary word recognition [33]. Nevertheless, some important research on large vocabulary continuous speech recognition (LVCSR) has been conducted [108; 80; 79; 62] and has shown that combining PHFs and ACFs can improve the ASR performance. In Metze et al. [80] adding 6 to 10 well chosen PHFs to supplement the standard ACF stream resulted in a 15% relative reduction of the WER for a read BN task, and a 7.5% reduction on a spontaneous scheduling task. In her PhD, Kirchhoff [62] investigated several state-level and word-level combination techniques. For the state-level combination technique she found a 5.6% relative reduction (form 29.03% to 27.41%) of the WER when tested on the German Verbmobil corpus. For the word-level combination experiments the best combined system reached a WER of 27.97%. The purely PHF-driven system was always found to perform worse than the acoustic baseline.

In [110] a system for the integration of linguistic features in an isolated word recognizer is proposed. The linguistic features are actually the PHF dimensions 'place' and 'manner' of articulation. The nine manner values that were used were: schwa, vowel, diphthong, semi-vowel, plosive, closure, fricative, affricate and nasal. The eight place values were: alveolar, dental, open, labial, lateral, palatal, retroflex and velar. Each phone can be considered as a bundle of these two feature dimensions and acoustic models can be trained according to the two dimensions. For the IWR-tests the segment-based SUMMIT-system was used. This system considers two types of landmarks: the transitional landmarks which match transitions between segments and the segment-internal landmarks denoting events within a segment. For the integration of the features, three strategies were proposed. (1) Early integration: during the search each hypothesized landmark is scored along both the manner and place dimensions. (2) Intermediate integration: segment-internal landmarks are scored along the manner dimension while segment-transition landmarks are modeled along the place dimension. (3) Late integration: Two recognizers are built, one along the manner and one along the place dimension. The N-best lists for the two recognizers were then combined by fusing the hypotheses. Experiments on a small vocabulary (Phonebook) showed that the third way of integration yielded the best results.

In a second IWR experiment the same team proposes a two-stage system. In the first stage a recognition is done with PHF based acoustic models. The result of this recognition is stored under the form of a N-Best list or cohort. In the second stage a detailed phoneme based system searches for the best hypothesis. The feature based models are thus used to limit the size of the search space. This two-stage system gave a 10% relative improvement on Phonebook (large vocabulary) compared to the best result reported in the literature. The logical extension of this method to CSR is described in [111].

In [68] fusing ACF and PHF information was obtained by means of a two-stream model and applied to phoneme recognition on TIMIT. Synchronous and asynchronous fusion was considered. State asynchrony was only allowed within a phoneme. Figure 7.1 represents the allowed transitions in the two-stream HMM model with 3×3 states. The ACF and the PHF model both start in their initial state and end with the same final state. The combined models outperformed the two single feature models. Asynchronous combination gave a relative error reduction of 9.3% while synchronous combination only gave a small reduction compared to the PHF model baseline. However, when fusion was performed during recognition and training, the less time consuming synchronous combination, performed almost as good as the more complex asynchronous combination.

From this literature overview I conclude that it is definitely worth investigating PHFs as a feature representation for ASR. Since a lot of

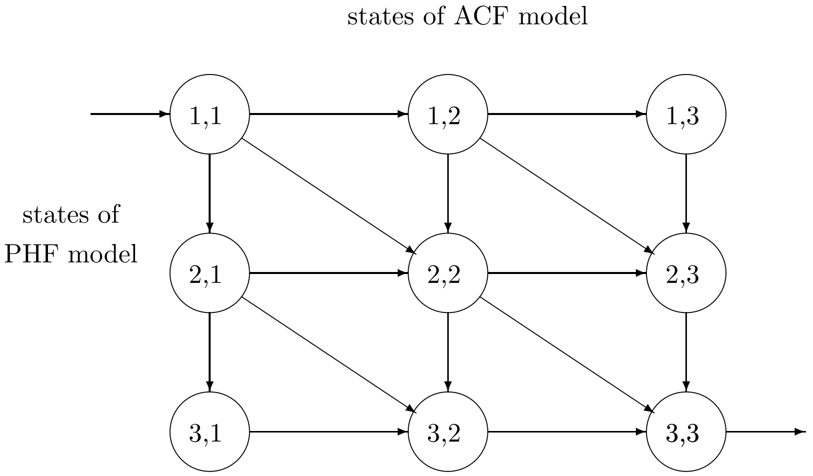


Fig. 7.1: Topology used in [68] for asynchronous combination of PHF model and ACF model.

authors tried to combine ACFs and PHFs in some way, this is something I will try out as well. Since I believe that not all possibilities of a synchronous combination have tried out yet, I will restrict to this and I will not consider asynchronous ways of combination (like in [68]). Instead I will discuss two other important problems which I consider responsible for the not so good performances of ASR systems based on PHFs alone.

7.2 Specific problems

I argue that part of the reason for the bad performance of purely PHF driven ASR resides in the suboptimal use of the PHFs in the standard HMM framework, a framework that is mainly optimized for MFCCs as the input features. I will first investigate methods of adapting this framework to the case of PHFs and to increase the accuracy of a purely PHF-based ASR-system. With this higher performance, it then makes more sense to reconsider the PHF system in combination with an ACF-based system. In the following sections I discuss two important differences between the MFCCs and PHFs that must be taken into account by the HMM framework.

7.2.1 Feature Correlations

One of the interesting properties of MFCCs is that their components are largely uncorrelated. This means that state-level emission distributions can successfully be modeled by a small number of Gaussian mixtures with diagonal covariance matrices. The binary PHFs on the other hand are expected to exhibit much larger correlations. The place of articulation of a consonant for instance is represented by 7 binary features, so it is clear that there will be correlations between those features. Due to these correlations the required emission distributions may no longer be represented efficiently by diagonal covariance GMMs. Of course, a GMM is able to model some degree of correlation between the features. But it should in the first place model the non-gaussian nature of the feature distribution and not so much the correlations. One approach would be to replace them by full covariance GMMs, but this would severely increase the number of model parameters per added Gaussian. I argue that it may be more efficient to adopt one of the following techniques:

- Feature selection with the aim to remove features carrying information already covered by other features. Mostly an information theoretic selection criterion is applied to control this process [62; 33].

- Global decorrelation such as PCA to transform the feature space in its whole to a lower-dimensional space of decorrelated features [29].

- State-dependent decorrelation that relies on state-dependent feature transformation matrices, like semi-tied covariance matrices [43].

The amount of correlation between two PHFs depends strongly on the state of the recognizer. For a particular consonant for example, the correlations between two or three place features will be important to model whereas they are maybe irrelevant for another phoneme. Therefore a global decorrelation or a global feature selection would not be meaningful. I will therefore explore the third technique.

Obviously, transforming the features in a given state and modeling the transformed features with diagonal covariance GMMs is equivalent to modeling the non-transformed features with full covariance GMMs. But if there exists an underlying model to obtain these transformations the total number of free parameters to be estimated is bound to be smaller and the use of the transformations is bound to be beneficial. Now it happens that Gales [43] has developed a ML training methodology to simultaneously train feature transformation matrices (MLLT-matrices) and HMM model parameters. I will adopt Gales' method and extend it in a way that it can also deal with another problem that is typical for PHFs and that is discussed in the next subsection.

7.2.2 Feature relevancy

From the description of the PHFs and their training (see also chapter 5) it became clear that not all features are relevant for all phones. Consequently, the emission distributions on a particular state should only be modeled in the subspace of the relevant features for that state. However, since working with different subspaces on different states causes problems of equivalence of likelihoods, the observation likelihoods may need to be factorized as the product of a relevant observation likelihood and an irrelevant observation likelihood. The latter can then be computed on the basis of global models and integrated in the system by adopting principles of missing data theory [57], more in particular likelihood imputation. In section 7.4 I will show that this likelihood imputation together with the state-dependent feature transformation concept can be embedded in a consistent probabilistic framework.

7.3 State-dependent feature transformations

An intermediate solution to the correlation problem of the PHFs would be to work with a *limited number* of full covariance matrices and to share them across states of which I think the correlations should be more or less the same. Typically these states will belong to models with the same central phoneme. Now it happens that first transforming the feature vector minus the mean vector $(\mathbf{x}_t - \mu)$ by means of a linear transformation (represented by a square, non-singular matrix A) and modeling the transformed feature vector with a diagonal covariance Gaussian is equivalent to modeling the feature vector with a full covariance Gaussian with a covariance matrix $\mathrm{A}^{-1}\Sigma\mathrm{A}^{t,-1}$ (Σ is the diagonal original covariance matrix of the Gaussian in the untransformed space). A set of full covariance matrices can thus be obtained by defining a number of transformation matrices that are shared by different states. The question that remains is how these additional matrices should be trained and how the training of the matrices affects the standard training of the means and variances of the Gaussians. Fortunately Gales [43] has developed an elegant training method using the ML-criterion for estimation of all free parameters of the system. Since the mathematical derivation is given in [43], I will restrict to the main principles of this method and I will just provide the necessary intermediate results that are needed to understand how the final result was obtained.

7.3.1 ML-estimation of transformation matrices

Equation (2.6) describes the probability density function (PDF) $b_j(\mathbf{x})$ as a mixture of M multivariate Gaussian PDFs $\mathcal{N}_{jk}(\mathbf{x}, \mu_{\mathbf{jk}}, \mathbf{\Sigma_{jk}})$ with a diagonal covariance matrix $\Sigma_{jk} = \text{diag}(\sigma^2_{jk1}, \ldots, \sigma^2_{jkD})$. Let us consider a full covariance matrix for $b_{jk}(\mathbf{x}) = \mathcal{N}_{jk}(\mathbf{x})$ instead of a diagonal one, but let this full covariance matrix be of a special nature i.e.

$$\Sigma^{'}_{jk} = \mathrm{A}^{-1}\Sigma_{jk}(\mathrm{A}^t)^{-1} \tag{7.1}$$

with A being a full square matrix of dimension D and
$\Sigma_{jk} = \text{diag}(\sigma^2_{jk1}, \ldots, \sigma^2_{jkD})$ the *diagonal* covariance matrix. This means that the determinant in the denominator of expression (2.7) must be replaced by

$$\begin{aligned}
|\Sigma^{'}_{jk}| &= \frac{|\Sigma_{jk}|}{|\mathrm{A}||\mathrm{A}^t|} && (7.2)\\
&= \frac{|\Sigma_{jk}|}{|\mathrm{A}|^2} && (7.3)
\end{aligned}$$

If S is the set of acoustic states, S can be divided into mutually exclusive subsets S_p $(p = 1, \ldots, P)$, such that

$$\mathrm{S} = \bigcup_{p=1}^{P} \mathrm{S}_p \tag{7.4}$$

and if a matrix A_p is assigned to S_p, then the emission function $b_j(\mathbf{x})$ of a state $j \in \mathrm{S}_p$ is composed of mixtures of the form

$$\begin{aligned}
b_{jk}(\mathbf{x}) &= \mathcal{N}(\mathbf{x}, \mu_{jk}, \Sigma^{'}_{jk}) && (7.5)\\
&= \frac{|A_p|}{(2\pi)^{D/2}|\Sigma_{jk}|^{1/2}} \exp[-\frac{1}{2}(\mathbf{x} - \mu_{jk})^t \mathrm{A}^t_p \Sigma^{-1}_{jk} \mathrm{A}_p(\mathbf{x} - \mu_{jk})] && (7.6)\\
&= \frac{|A_p|}{(2\pi)^{D/2}|\Sigma_{jk}|^{1/2}} \exp[-\frac{1}{2}[\mathrm{A}_p(\mathbf{x} - \mu_{jk})]^t \Sigma^{-1}_{jk} \mathrm{A}_p(\mathbf{x} - \mu_{jk})] && (7.7)
\end{aligned}$$

and which can best be evaluated in the $\mathrm{A}_p(\mathbf{x} - \mu_{jk})$-space. The first question to answer is how to subdivide the acoustic model state set. A logical response seems to be to make this division according to the identity of the phoneme that is modeled by this state (in certain contexts).

The second question is how to find re-estimation formulae that allow me to optimize the transformation matrices together with the traditional GMM parameters.

In order to find such re-estimation formulae I need to explain some basic concepts about the standard EM-algorithm that is used for parameter estimation. This algorithm uses an auxiliary function $Q(\lambda, \overline{\lambda})$ that depends on the model parameters $(\overline{\lambda})$ and the old parameters (λ). There are two important properties of this Q-function that I need at the moment.

1. The EM-algorithm will search for the $\overline{\lambda}$ that maximizes $Q(\lambda, \overline{\lambda})$. Whenever the Q-function raises, the log-likelihood will raise as well.

2. The Q-function can be written as a sum of terms with each term only depending on one parameter group. The parameter groups are: the transition probabilities a_{ij} (term Q_{a_i}), the Gaussian parameters $(\mu_{\mathbf{jk}}, \Sigma_{jk})$ (term Q_{b_j}) and the mixture weights c_{jk} (term Q_{c_j}). There is also a term that is function of the initial values π, but this is not important for the discussion here.

Since the mixture PDF b_{jk} now has an extra dependency on A_p, the term Q_{b_j} will also depend on the matrix A_p. It can be shown that the expression for the Q-function now boils down to

$$Q(\lambda, \overline{\lambda}) = Q_\pi(\lambda, \overline{\pi}) + \sum_i Q_{a_i}(\lambda, \overline{a}_{ij}) + \sum_{p=1}^{P} \sum_{j,k \in S_p} Q_{b_j}(\lambda, \overline{A}_p, \overline{\mu}_{jk}, \overline{\Sigma}_{jk}) + \sum_j Q_{c_j}(\lambda, \overline{c}_{jk}) \tag{7.8}$$

Here I explicitly wrote the dependency of b_{jk} on A_p, but I will from now on stick to the notation $Q(\lambda, \overline{b}_{jk})$. Now

$$Q_{b_j}(\lambda, \overline{b}_{jk}) = \sum_{t=1}^{T} \zeta_t(j,k) \log \overline{b}_{jk}(\mathbf{x}_t) \tag{7.9}$$

The introduced $\zeta_t(j,k)$ is actually the probability that $\mathbf{x}_t$ was emitted on state j by mixture component k given the previous estimates of the model parameters λ (including A_p) and given the observed vector sequence $\mathbf{X}$.

The re-estimation formulae for each parameter group are easily found by taking the derivatives of the associated term of the Q-function and by setting this to zero. So in order to find the re-estimation formulae for $\overline{\mathrm{A}}_p$, I will have to calculate the derivative of the third term in (7.8) with respect to $\overline{\mathrm{A}}_p$. From (7.7) it can be seen that this derivative will be a linear combination of the derivatives of $\overline{b}_{jk}$ with respect to $\overline{\mathrm{A}}_p$. One such a derivative can be written as,

$$\frac{d\overline{b}_{jk}(\mathbf{x_t})}{d\overline{\mathrm{A}}_p} = \mathcal{N}(\mathbf{x_t}, \overline{\mu}_{\mathbf{jk}}, \overline{\mathbf{\Sigma}'_{\mathbf{jk}}})[\overline{\mathrm{A}}_{\mathbf{p}}^{\mathbf{-1}} - (\mathbf{x_t} - \overline{\mu}_{\mathbf{jk}})(\mathbf{x_t} - \overline{\mu}_{\mathbf{jk}})^{\mathbf{t}}\overline{\mathrm{A}}_{\mathbf{p}}^{\mathbf{t}}\overline{\mathbf{\Sigma}}_{\mathbf{jk}}^{\mathbf{-1}}] \quad (7.10)$$

This result makes use of the following two mathematical rules for matrix differentiation, (X and C represent a $D \times D$-matrix and $\mathbf{a}$ is a D dimensional column vector).

$$\frac{d|X|}{dX} = |X|X^{-1} \quad (7.11)$$

$$\frac{d(\mathbf{a}^t X^t C X \mathbf{a})}{dX} = 2\mathbf{a}\mathbf{a}^t X^t C \quad (7.12)$$

It is a simple exercise to verify result (7.10) by substituting X by A_p, C by Σ_{jk}^{-1} and $\mathbf{a}$ by $(\mathbf{x} - \mu_{\mathbf{jk}})$.

The next step is to set the derivative to zero and to solve the equation for $\overline{\mathrm{A}}_p$. Unfortunately, there is no elegant way to solve this matrix equation, meaning that no simple re-estimation formula can be found.

The way that was proposed by Gales to circumvent this problem is to consider the *rows* of matrix $\overline{\mathrm{A}}_p$ and to try to find some re-estimation formula for the m-th row, $\overline{\mathbf{a}}_m$ (I will leave the index p of the matrix from now on). It is possible to rewrite expression (7.7) in function of $\overline{\mathbf{a}}_m$ by introducing another row-vector (of dimension D): $\mathbf{p}_m$, defined as the row vector containing the cofactors associated with the elements of $\mathbf{a}_m$. The inner product $\mathbf{p}_m\mathbf{a}_m^t$ is nothing else than the determinant of A_p. Since Σ_{jk} is diagonal, I can rewrite b_{jk} as

$$\overline{b}_{jk}(\mathbf{x}_t) = \frac{\mathbf{p}_m\overline{\mathbf{a}}_m^t}{(2\pi)^{D/2}|\overline{\Sigma}_{jk}|^{1/2}} \exp[-\frac{1}{2}\sum_{m=1}^{D}\frac{1}{(\overline{\sigma}_{jkm}^2)}(\mathbf{x}_t - \overline{\mu}_{jk})^t\overline{\mathbf{a}}_m^t\overline{\mathbf{a}}_m(\mathbf{x}_t - \overline{\mu}_{jk})] \quad (7.13)$$

Now the derivative of this expression to $\overline{\mathbf{a}}_m$ can be written as,

$$\frac{d\overline{b}_{jk}(\mathbf{x}_t)}{d\overline{\mathbf{a}}_m} = \mathcal{N}(\mathbf{x}_t, \overline{\mu}_{jk}, \overline{\Sigma}'_{jk})[\frac{\mathbf{p}_m}{\mathbf{p}_m\overline{\mathbf{a}}_p^t} - \frac{1}{(\overline{\sigma}_{jkm}^2)}\overline{\mathbf{a}}_m(\mathbf{x}_t - \overline{\mu}_{jk})(\mathbf{x}_t - \overline{\mu}_{jk})^t] \quad (7.14)$$

This result is due to the following differentiation rules, with $\mathbf{a}$, $\mathbf{b}^t$ and $\mathbf{x}$

being D dimensional row vectors.

$$\frac{d(\mathbf{a}\mathbf{x}^t)}{d\mathbf{x}} = \mathbf{a} \tag{7.15}$$

$$\frac{d(\mathbf{b}^t\mathbf{x}^t\mathbf{x}\mathbf{b})}{d\mathbf{x}} = 2\mathbf{x}(\mathbf{b}\mathbf{b}^t) \tag{7.16}$$

It is again an easy exercise to obtain expression (7.14) by replacing $\mathbf{a}$ with $\mathbf{p}_m$ and $\mathbf{b}$ with $(\mathbf{x}_t - \mu_{jk})$. Another way to obtain this result is to consider the m-th column in expression (7.10). The inverse matrix $\overline{\mathrm{A}}^{-1}$ can be written as a so-called *adjunct* matrix divided by the determinant. It happens that the m-th column of the adjunct matrix is exactly $\mathbf{p}_m$.

Putting everything together, I have to solve the following equation to find a re-estimation formula for $\overline{\mathbf{a}}_m$,

$$\beta\frac{\mathbf{p}_m}{\mathbf{p}_m\overline{\mathbf{a}}_m^t} - \overline{\mathbf{a}}_m G^{(m)} = 0 \tag{7.17}$$

After combination of (7.14),(7.9) and the third term of (7.8) I get for β and $G^{(m)}$,

$$\beta = \sum_{j,k\in S_p}\sum_{t=1}^{T}\zeta_t(j,k) \tag{7.18}$$

$$G^{(m)} = \sum_{j,k\in S_p}\frac{1}{(\overline{\sigma}^2_{jkm})}\sum_{t=1}^{T}\zeta_t(j,k)(\mathbf{x}_t - \overline{\mu}_{jk})(\mathbf{x}_t - \overline{\mu}_{jk})^t \tag{7.19}$$

At this point it is important to remark that the re-estimated $\overline{\mathbf{a}}_m$ emerging from equation (7.17) makes use of the still unknown re-estimated values for $\overline{\mu}_{jk}$ and $\overline{\Sigma}_{jk}$. That is why the algorithm that will solve (7.17) for $\overline{\mathbf{a}}_m$ will assume $G^{(m)}$ to be known or fixed for each step in the iterative solution for all parameters (see 7.3.2).

The problem has now been reduced to solving (7.17). Gales proved that each equation of the form,

$$\beta\frac{\mathbf{p}_i}{\mathbf{p}_i\mathbf{w}_i^t} = \mathbf{w}_i G^{(i)} - \mathbf{k}^{(i)} \tag{7.20}$$

has a solution for $\mathbf{w}_i$ that can be written as,

$$\mathbf{w}_i = (\alpha\mathbf{p}_i + \mathbf{k}^{(i)})(G^i)^{-1} \quad \forall i \in [1, D] \tag{7.21}$$

α is hereby a solution of the quadratic equation,

$$\alpha^2\mathbf{p}_i(G^i)^{-1}\mathbf{p}_i^t + \alpha\mathbf{p}_i(G^i)^{-1}\mathbf{k}^{(i)t} - \beta = 0 \tag{7.22}$$

Equation (7.17) is a special case of (7.20) with $\mathbf{k}^{(i)} = \mathbf{0}$ and α can easily be found from (7.22) (with $\mathbf{k}^{(i)} = \mathbf{0}$). The rows $\overline{\mathbf{a}}_m$ can thus be updated using,

$$\overline{\mathbf{a}}_m = \sqrt{\frac{\beta}{\mathbf{p}_m (G^m)^{-1} \mathbf{p}_m^t}} \mathbf{p}_m (G^m)^{-1} \quad \forall m \in [1, D] \tag{7.23}$$

After $\mathbf{a}_m$ has been re-estimated on the basis of the current $\mathbf{p}_m$, the cofactor row $\mathbf{p}_{m+1}$ will be needed in order to re-estimate $\mathbf{a}_{p,m+1}$. It is necessary to recalculate this $\mathbf{p}_{m+1}$ on the basis of the new estimated $\mathbf{a}_m$. Hence, for every row update $\mathbf{p}_m$ must be recalculated. Several iterations can be performed until the obtained solutions for all rows $\mathbf{a}_m$ are stable. This iterative solution will be called the row-by-row re-estimation of the matrices $\overline{\mathrm{A}}_p$ and it requires the calculation of D inverse matrices $(G^m)^{-1}$.

The matrices $\overline{\mathrm{A}}_p$ that are obtained in this way are called MLLT-matrices (Maximum Likelihood Linear Transformation matrices). Recently more elaborated algorithms have been proposed to train the matrices in a discriminative way (Discriminative Likelihood Linear Transformation matrices or DLLT-matrices [115]).

7.3.2 Training algorithm

Suppose that I have clustered my states and I want to perform a single re-estimation of the model parameters together with the MLLT-matrices. Then I initialize all MLLT-matrices to the unit matrix. The entire algorithm (that will be called MLLT-algorithm) to estimate all model parameters, can then be summarized as follows:

1. Estimate the mean of all Gaussians using the standard re-estimation formula Equation (2.9).

2. Use the current estimates of the MLLT-matrices $\mathrm{A}_p \;\; \forall p \in \{1, \ldots, P\}$ to estimate the component specific diagonal variances using (see [43] for proof),
$$\overline{\Sigma}_{jk} = \mathrm{diag}(\mathrm{A}_p W_{jk} \mathrm{A}_p^t) \quad \forall (j,k) \in \mathrm{S}_p \tag{7.24}$$
with W_{jk} being defined as the right-hand side of equation (2.10).

3. Estimate the MLLT-matrices $\overline{\mathrm{A}}_p$ using the current set of component specific diagonal variances. This is the row by row scheme explained in 7.3.1.

4. Got to (2) until convergence, or until some appropriate criterion is satisfied

In order to further reduce the number of parameters that must be estimated, the MLLT-matrices can be given a block-diagonal structure. In the case of MFCCs for instance the static parameters can be grouped in one block, the delta's in another and the delta-delta's in a third block.

Until now I have assumed that the number of mixture components M was fixed. Mostly training will start with $M = 1$ and end with the desired number of mixtures. Theoretically there are two ways to train the MLLT-matrices associated with a state with a certain mixture number.

1. First *mix up* to the desired mixture number without MLLT-matrices and then start the training of the matrices by means of the MLLT-algorithm applied to the current models.

2. For each intermediate mixture number, associated MLLT-matrices can be estimated and then transferred to the higher mixture number as initial values for further estimation.

The former method has the advantage of a limited training overhead because the MLLT estimation must only be done in the final mixture stage. The latter method requires more computations but it may lead to a better solution. The method that performs best depends on the experimental conditions. I chose to adopt the first method because it is computationally less expensive.

The last question that must be answered in order to have a full overview of the training algorithm is: How many re-estimations are needed for each fixed M? Normally 4 or 5 iterations are sufficient. Because of the extra MLLT-algorithm, there are now six iterations.

1. First two standard iterations of Baum-Welch re-estimations are carried out on the incoming models and matrices. For the first training method all matrices are initialized to the unit matrix.

2. Secondly the MLLT-algorithm is run to estimate the new MLLT-matrices.

3. Finally three more Baum-Welch re-estimation cycles were run to better estimate the model parameters for the new matrices.

The experimental results for this technique will be presented in section 7.5.

7.4 Feature relevancy training scheme

The second important problem associated with the use of PHFs for ASR is the fact that not all features are relevant for all phones. This can be formulated differently by stating that not all features are relevant for all states of the recognizer. In this section I propose a new technique that tries to model this aspect of PHFs. The technique can be combined with the decorrelation technique proposed in the previous section.

7.4.1 Adaptation of the re-estimation formulae

Suppose that the feature vector $\mathbf{x}_t$ has dimension D, that $R(j)$ is the set of relevant features for state j and that $I(j)$ is the complementary set of irrelevant features for that particular state. I can then define vector $\mathbf{x}_k^{R(j)}$ as the acoustic vector that is associated with mixture k of state j,

$$\mathbf{x}_k^{R(j)}(m) = \begin{cases} \mathbf{x}(m) & \text{if } m \in R(j) \\ \mu_{jk}(m) & \text{else} \end{cases}$$

The issue of which features are relevant for state j and which are not will be discussed in section 7.5. I propose to model the relevant features by standard state-dependent models (GMMs) and the irrelevant features by a single global model. That global model will be called an *imputation model* from now on. I propose to use an imputation GMM with G mixture components

$$b_G(\mathbf{x}) = \sum_{l=1}^{G} d_l \; g_l(\mathbf{x}) \tag{7.25}$$

with $g_l(\mathbf{x}) = \mathcal{N}(\mathbf{x}, \mu_l, \Sigma_l)$, a multivariate Gaussian PDF with mean vector μ_l and diagonal covariance matrix Σ_l. A second feature vector $\mathbf{x}_l^{I(j)}$ is now introduced as

$$\mathbf{x}_l^{I(j)}(m) = \begin{cases} \mathbf{x}(m) & \text{if } m \in I(j) \\ \mu_l(m) & \text{else} \end{cases}$$

This feature vector is associated with the irrelevant features of state j and is defined for each mixture component l of the imputation model.

I propose to write the PDF $b_j(\mathbf{x})$ of the emission function in state j as,

$$\begin{aligned} b_j(\mathbf{x}) &= \sum_{k=1}^{M} c_{jk} \left[b_{jk}(\mathbf{x}_k^{R(j)}) \sum_{l=1}^{G} d_l \; g_l(\mathbf{x}_l^{I(j)}) \right] && (7.26) \\ &= \{\sum_{l=1}^{G} d_l \; g_l(\mathbf{x}_l^{I(j)})\} \; \{\sum_{k=1}^{M} c_{jk} \; b_{jk}(\mathbf{x}_k^{R(j)})\} && (7.27) \end{aligned}$$

This means that the Gaussian PDF for each mixture component is now evaluated in $\mathbf{x}_k^{R(j)}$. Obviously the contributions of the vector components corresponding to irrelevant features are $\prod_{d=1}^{D_I} 1/\sqrt{2\pi\sigma_{d,i}^2}$, with D_I the number of irrelevant features and $\sigma_{d,i}^2$ the corresponding variances. I have to multiply with this factor in order to normalize $b_j(\mathbf{x})$. The likelihood factor which carries the discriminative information is multiplied with an *imputation* likelihood given by

$$\mathrm{l}_{imput} = \sum_{l=1}^{G} d_l \; g_l(\mathbf{x}_l^{I(j)}) \tag{7.28}$$

This imputation likelihood is obtained by evaluating each Gaussian component l of the imputation model in $\mathbf{x}_l^{I(j)}$. Again the contribution of the vector components corresponding to the relevant features are $\prod_{d=1}^{D_R} 1/\sqrt{2\pi\sigma_{d,r}^2}$, with D_R the number of relevant features and $\sigma_{d,r}^2$ the corresponding variances. Again I have to multiply with this factor in order to normalize $b_j(\mathbf{x})$. The imputation likelihood factor serves as a normalization. This normalization is necessary in order to make the likelihoods emerging from different states with different numbers of relevant features, compatible to each other.

As was explained in section 7.3.1, I have to maximize the Q-function in order to find the re-estimation formulae for the model parameters. Since there are now extra parameters (d_l, μ_l, Σ_l) to estimate, the expression of the Q-function (7.8) is no longer valid here. I therefore have to start from a more general expression of the Q-function that is given here as a starting point without proof,

$$Q(\lambda, \overline{\lambda}) = \sum_S \sum_K \frac{p(\mathbf{X}, S, K|\lambda)}{p(\mathbf{X}|\lambda)} \log p(\mathbf{X}, S, K|\overline{\lambda}) \tag{7.29}$$

I now try to find an expression for $p(\mathbf{X}, S, K|\lambda)$ in order to substitute it into (7.29) later on. I start from

$$p(\mathbf{X}|\lambda) = \sum_S p(\mathbf{X}, S|\lambda) \tag{7.30}$$

$$= \sum_S \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(\mathbf{x}_t) \tag{7.31}$$

in which I substitute $b_{s_t}(\mathbf{x}_t)$ by,

$$b_{s_t}(\mathbf{x}_t) = \sum_{l=1}^{G} d_{l_t} \; g_{l_t}(\mathbf{x}_{t,l}^{I(s_t)}) \; \sum_{k=1}^{M} c_{s_t k_t} \; b_{s_t k_t}(\mathbf{x}_{t,k}^{R(s_t)}) \tag{7.32}$$

After some calculation I get,

$$p(\mathbf{X}, S|\lambda) = \sum_{k\in\Omega^T}\sum_{l\in\Omega^T}\prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t k_t}(\mathbf{x}_t^{R(s_t)})\ c_{s_t k_t} d_{l_t} g_{l_t}(\mathbf{x}_t^{I(s_t)}) \tag{7.33}$$

$$= \sum_{k\in\Omega^T}\sum_{l\in\Omega^T} p(\mathbf{X}, S, K, L|\lambda) \tag{7.34}$$

with Ω^T the T-th product set of $\Omega = \{1, 2 \ldots, M\}$. Using $p(\mathbf{X}, S, K, L|\lambda)$ I can rewrite the Q-function as,

$$Q(\lambda, \overline{\lambda}) = \sum_S \sum_K \sum_L \frac{p(\mathbf{X}, S, K, L|\lambda)}{p(\mathbf{X}|\lambda)} \log p(\mathbf{X}, S, K, L|\overline{\lambda}) \tag{7.35}$$

by introducing an extra summation over the mixture components of the imputation model. Substituting the $\log p(\mathbf{X}, S, K, L|\lambda)$ into (7.35) I can rewrite this as,

$$Q(\lambda, \overline{\lambda}) = Q_\pi(\lambda, \overline{\pi}) + \sum_i Q_{a_i}(\lambda, \overline{a}_{ij}) \tag{7.36}$$

$$+ \sum_j \sum_{k=1}^{M} Q_{b_j}(\lambda, \overline{b}_{jk}) + \sum_j Q_{c_j}(\lambda, \overline{c}_{jk}) \tag{7.37}$$

$$+ \sum_l Q_g(\lambda, \overline{g}_l) + Q_d(\lambda, \overline{d}_l) \tag{7.38}$$

The main observation that can be made here is that there are two extra terms $Q_g(\lambda, \overline{g}_l)$ and $Q_d(\lambda, \overline{d}_l)$ in this expression compared to the standard expansion of the Q-function. The four functions that are affected by my modification are defined as,

$$\begin{cases} Q_{b_j}(\lambda, \overline{b}_{jk}) = \sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l) \log \overline{b}_{jk}(\mathbf{x}_{t,l}^{R(j)}) \\ Q_{c_j}(\lambda, \overline{c}_{jk}) = \sum_{k=1}^{M} \sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l) \log \overline{c}_{jk} \\ Q_g(\lambda, \overline{g}_l) \quad = \sum_j \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l) \log \overline{g}_l(\mathbf{x}_{t,k}^{I(j)}) \\ Q_d(\lambda, \overline{d}_l) \quad = \sum_j \sum_{k=1}^{M} \sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l) \log \overline{d}_l \end{cases}$$

in which the new $\zeta_t(j,k,l) = p(s_t = j, k_t = k, l_t = l \,|\, \mathbf{X}, \lambda)$ defines the probability of being in state j at time t with the relevant features being in mixture component k and the irrelevant features being in mixture component l of the imputation model, given the observation $\mathbf{X}$. These counts will have to be determined on the basis of an adapted forward-backward algorithm.

Let me now continue with the derivation of the re-estimation formulae for all model parameters. I start with the weights c_{jk} of the Gaussians. To this end the function $Q_{c_j}(\lambda, \overline{c}_{jk})$ must be maximized under the condition,

$$\sum_{k=1}^{M} \overline{c}_{jk} = 1 \tag{7.39}$$

The result is,

$$\overline{c}_{jk} = \frac{\sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l)}{\sum_{k=1}^{M} \sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l)} \tag{7.40}$$

For the global weights d_l the function $Q_d(\lambda, \overline{g}_l)$ must be maximized under the condition,

$$\sum_{l=1}^{G} \overline{d}_l = 1 \tag{7.41}$$

The result now is,

$$\overline{d}_l = \frac{\sum_j \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l)}{\sum_j \sum_{l=1}^{G} \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l)} \tag{7.42}$$

Equations (7.40) and (7.42) are the re-estimation formulae for the state specific weights and the global weights of the imputation model respectively.

To find the formulae for $\overline{\mu}_{jk}$ and $\overline{\Sigma}_{jk}$ I consider $Q_{b_j}(\lambda, \overline{b}_{jk})$. After maximization, the resulting formulae are,

$$\begin{cases} \overline{\mu}_{jk} = \frac{\sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l) \mathbf{x}_{t,l}^{R(j)}}{\sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l)} \\ \overline{\Sigma}_{jk} = \frac{\sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l) (\mathbf{x}_{t,l}^{R(j)} - \overline{\mu}_{jk})(\mathbf{x}_{t,l}^{R(j)} - \overline{\mu}_{jk})^t}{\sum_{l=1}^{G} \sum_{t=1}^{T} \zeta_t(j,k,l)} \end{cases}$$

whereas for the parameters of the imputation model I finally get,

$$\begin{cases} \overline{\mu}_{l} = \frac{\sum_j \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l) \mathbf{x}_{t,k}^{I(j)}}{\sum_j \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l)} \\ \overline{\Sigma}_{l} = \frac{\sum_j \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l) (\mathbf{x}_{t.k}^{I(j)} - \overline{\mu}^g)(\mathbf{x}_{t,k}^{I(j)} - \overline{\mu}^g)^t}{\sum_j \sum_{k=1}^{M} \sum_{t=1}^{T} \zeta_t(j,k,l)} \end{cases}$$

In order to obtain the counts $\zeta_t(j,k,l)$, I have to record the forward-backward probabilities $\alpha_t(i)$ and $\beta_t(i)$ at any grid point (t,i),

$$\alpha_t(i) = \left[\sum_j \alpha_{t-1}(j) a_{ji} \right] b_i(\mathbf{x}_t) \tag{7.43}$$

$$\beta_t(i) = \sum_j a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j) \tag{7.44}$$

I then substitute $b_j(\mathbf{x}_t)$ by expression (7.32) to calculate $\zeta_t(j,k,l)$ as

$$\zeta_t(j,k,l) = \frac{1}{p(\mathbf{X}|\lambda)} \sum_i \alpha_{t-1}(i)\ [a_{ij}\ c_{jk}\ b_{jk}(\mathbf{x}_t^{R(j)})\ d_l\ g_l(\mathbf{x}_t^{I(j)})]\ \beta_t(j) \tag{7.45}$$

which is a minor modification of the standard expression.

7.4.2 Training algorithm

My goal is to combine the state-dependent feature transformation matrices with the relevancy handling technique. This can be accomplished easily by considering only the relevant rows of the matrices A_p in expression (7.23). It is important to remark that the sets S_p must consist of states with the same set of relevant features $R(j)$ (and the same $I(j)$). Otherwise I cannot assign relevant rows and irrelevant rows to the matrix A_p associated with this S_p. This is the motivation why I use a clustering on the basis of the central phoneme of the triphone model the state belongs to. The irrelevant rows of A_p are just ignored and filled up with zero's. The training is then analogous to the MLLT-training except for two minor changes.

- There are now two extra re-estimation steps involved in the re-estimation of $\overline{\mu}_l$ and $\overline{\Sigma}_l$. They are performed in the first step of the MLLT-algorithm (section 7.3.2).

- It is not necessary to calculate the irrelevant variances nor the irrelevant rows for the MLLT-matrices during the second and third step of the MLLT-algorithm.

7.5 Validating the proposed techniques

I implemented my techniques in the HTK-toolkit and I applied it to construct a PHF-based and an ACF-based recognizer with this toolkit for TIMIT and WSJ. However due to memory limitations I was not able to perform the feature relevancy technique on WSJ. This drawback will be explained later. The PHFs consisted of 25 static PHFs with their first order derivatives yielding a 50 dimensional feature vector. No second order derivatives were included.

For TIMIT, the acoustic models are cross-word triphone HMMs with tied distributions (GMMs). State tying was performed using DT-based clustering yielding 1271 states and 6074 physical models for the ACF case and 1843 states and 9821 physical models for the PHF case.

For WSJ, training was done on the combined WSJ0+1 training set and testing on the November 92 5k closed vocabulary test set (330 sentences or 5353 words). The acoustic models were cross-word triphone HMMs. State tying was done using DT-based clustering yielding 9499 states and 27475 physical models for the ACF case and 12011 states and 30561 physical models for the PHF case. The lexicon contained 4988 entries and the language model was the bigram LM delivered with the corpus.

The baseline training involved 4 Baum-Welch re-estimation steps for each number of mixtures, and the number of mixtures M was changed from 1 to 2,3,4 and 6 for TIMIT and from 1 to 2,4,6,8 and 10 for WSJ. When applying the MLLT-method or the relevancy method the training proceeded as described in section 7.3.2 and 7.4.2. I performed 10 iterations of the MLLT-algorithm and I allowed 100 iterations in step 3 of that algorithm. I used 41 state sets S_p each grouping the states associated with models of the same phoneme (central symbol).

7.5.1 Results for the decorrelation handling

The recognition results for the systems with the MLLT technique included, for both ASR-systems working with ACF and PHF features for TIMIT are summarized in Table 7.1.

		ACF					PHF				
system	M	# pars.	WER	D	S	I	# pars.	WER	D	S	I
baseline	2	198276	6.05	21	66	8	368600	9.43	16	105	27
MLLT	2	+20787	5.29	25	50	8	+51250	6.31	20	68	11
baseline	3	297414	5.99	22	64	8	552900	9.17	14	103	27
MLLT	3	+20787	4.46	18	44	8	+51250	**5.73**	**15**	**66**	**9**
baseline	4	396552	5.16	20	56	5	737200	8.34	15	90	26
MLLT	4	+20787	4.20	16	43	7	+51250	5.99	15	64	15
baseline	6	594828	4.59	18	50	4	1105800	8.85	20	88	31
MLLT	6	+20787	**3.69**	**17**	**36**	**5**	+51250	6.11	18	63	15

Tab. 7.1: WER (%) for the baseline ACF ($D = 39$) and PHF system ($D = 50$) and the ACF and PHF system with MLLT-matrices for different numbers of Gaussian components tested on TIMIT. M is the number of mixtures.

From this Table I can conclude that the baseline PHF system cannot compete with the baseline ACF system. The MLLT-technique is very helpful: it yields a 20% relative improvement for the ACF system and a

31% relative improvement for the PHF system. This is in line with my expectation that the PHFs are more correlated than the ACFs. However I did not expect the ACF system to improve so much because of the common assumption that MFCCs are highly uncorrelated. Apparently the MFCCs are still correlated to some extent. Observe that the PHF system may be slightly overtrained, since the performance starts to decrease when 4 mixtures or more are used. This is due to the larger dimension of the feature vector. Here it would make sense to omit the delta PHFs from the feature vector. But since I will train the models on a larger database like WSJ0+1, this "overfitting" problem will disappear. Table 7.2 represents the results on WSJ. Here 14% relative improvement was obtained on the ACF baseline and 26% on the PHF system. Again the improvement is larger on the PHF system, but the improvements are not as large as on TIMIT. I also looked at the total log likelihood per frame measured during training. In order to illustrate the advantage of the MLLT method I adopted the second way to train the matrices i.e. for each intermediate mixture number. Figure 7.2 represents the log-likelihood during the ACF model training. The log-likelihood was plotted for the last four iterations of the MLLT training algorithm and standard Baum-Welch algorithm. The mixture number was thereby increased from 2 to 10 in steps of 2. The Figure reveals that for the MLLT-method this log likelihood is always significantly higher than for the baseline method.

		ACF					PHF				
sys.	M	#pars.	WER	D	S	I	#pars.	WER	D	S	I
bas.	4	2963688	7.53	59	287	57	4804400	12.78	100	475	109
MLLT	4	+20787	6.61	58	257	39	+51250	8.80	61	345	65
bas.	6	4445532	7.02	54	267	55	7206600	11.33	72	440	85
MLLT	6	+20787	6.11	56	234	37	+51250	8.23	51	326	60
bas.	8	5927376	7.36	52	273	69	9608800	11.20	72	426	85
MLLT	8	+20787	**6.05**	**52**	**236**	**36**	+51250	8.14	56	312	68
bas.	10	7409220	7.70	23	266	123	12011000	10.85	72	420	84
MLLT	10	+20787	6.22	51	240	42	+51250	**7.98**	**51**	**317**	**59**

Tab. 7.2: WER (%) for the baseline ACF ($D = 39$) and PHF system ($D = 50$) and the ACF and PHF system with MLLT-matrices for different numbers of Gaussian components tested on WSJ. M is the number of mixtures.

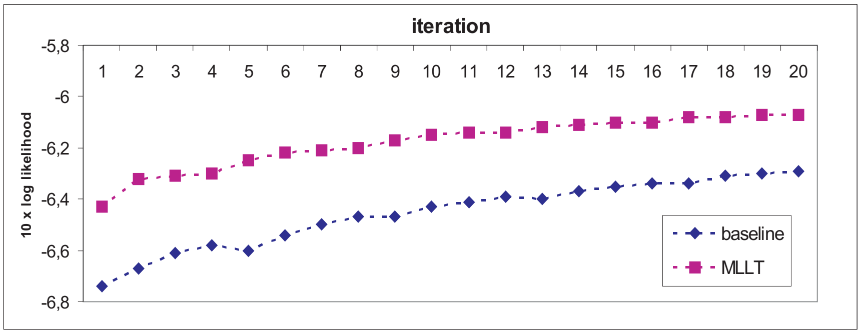


Fig. 7.2: The total log likelihood per frame for each iteration during training the ACF models on WSJ. The MLLT-method yields higher likelihoods.

7.5.2 Results for relevancy handling

Let us now look at the results for the feature relevancy handling technique. This technique could only be tested on TIMIT because of the extensive use of memory during the training of the models. In order to keep track of all sufficient statistics I need

$$M \times N_s \times G \times D/2(D/2-1) \times \text{sizeof}\{\text{float}\} \qquad (7.46)$$

memory: for WSJ, the number of states $N_s = 12011$ and the dimension of the PHF vector $D = 50$. I will now fill in some default values for M, the number of state specific mixtures and G, the number of mixtures in the imputation model. For $M = 6$ and $G = 16$ and for floating point numbers of 4 bytes, the requested memory is 2.7 Gbyte. Therefore I only tested the relevancy method on the smaller TIMIT system.

The results are represented in Tables 7.3. The number of state-dependent mixtures (M) was 3 (the best result in Table 7.1 for the TIMIT PHF system) and the number of mixtures for the imputation model G was raised from 8 to 32. This yielded 5.29% WER, a 7.7% relative improvement compared to the best MLLT-system trained on PHFs.

Since I only obtained a marginal improvement with this technique on TIMIT, I did not invest effort in optimizing the memory usage in order to be able to do experiments on WSJ.

M	G	WER (%)	D	S	I
3	8	5.54	16	56	15
3	16	**5.29**	**15**	**55**	**13**
3	32	5.48	17	57	12

Tab. 7.3: WER (%) for the MLLT system with GMM relevancy handling trained on the PHFs. M is the number of state-specific mixtures, whereas G is the number of mixtures of the imputation model.

7.5.3 Results with CMS

Until now the best performance I could reach on WSJ was 6.05%. Since it is well known that Cepstral Mean Subtraction (CMS) is an efficient method to further reduce the WER, I also added CMS to my best WSJ system. CMS consists of subtracting the mean cepstral vector, calculated over the utterance, from every incoming cepstral vector prior to training/decoding. The results of CMS added to the baseline and the MLLT-systems are represented in Table 7.4. From this Table I can conclude that CMS is helpful not only for the baseline system where it reduces the WER from 7.70% to 6.46% (16% relative) but also for MLLT system which is now improved by 9% relative: from 6.22% to 5.64%. Remark that the WER now decreases monotonically as the mixture number M increases.

system	M	WER	D	S	I
baseline+CMS	4	7.42	58	279	60
MLLT+CMS	4	6.54	68	237	45
baseline+CMS	6	6.78	51	257	55
MLLT+CMS	6	6.18	61	229	41
baseline+CMS	8	6.52	49	246	54
MLLT+CMS	8	5.85	51	225	37
baseline+CMS	10	6.46	48	247	51
MLLT+CMS	10	**5.64**	**47**	**222**	**33**

Tab. 7.4: WER (%) for the CMS technique applied on the baseline and MLLT ACF system tested on WSJ.

7.5.4 Comparison to MIDA

To conclude this study, I also performed some tests with the ESAT-recognizer [25; 31]. The latter can work with two kinds of features: (1) classical MFCCs and (2) MIDA-features, with MIDA being a decorrelation technique that has been proposed in [30] and which stands for Mutual Information Discriminative Analysis. A comparison with MIDA will allow me to assess whether it is better than MLLT or not. I did the following four tests:

1. Use the PHFs and apply MIDA to the PHF-vector.
2. Use the PHFs without applying MIDA
3. Apply MIDA on the Mel spectral coefficients+mean subtraction (MS).
4. Use the standard MFCCs+mean subtraction (MS).

In the third and fourth test, the mean of the logarithm of the spectral coefficients are subtracted. The results of my experiments are listed in Table 7.5. They reveal that PHF+MIDA and PHF+MLLT achieve a similar WER (7.83% versus 7.98%). Comparing tests 3 and 4, reveals that MIDA yields a significant improvement when compared to MFCC. Finally, comparing test 3 with my ACF system using MLLT+CMS (Table 7.4) shows that MIDA and MLLT perform comparably (from 6.31% to 5.19% versus from 6.46% to 5.64%). Another conclusion is that the ESAT-recognizer without MIDA performs significantly better on the PHFs than does my system (8.57% versus 10.85%). It is also obvious that MIDA does not help that much for PHFs (from 8.57% to 7.83%).

test	features	WER	D	S	I
1	PHF+MIDA	**7.83**	**56**	**329**	**34**
2	PHF	8.57	54	353	52
3	MELSPEC+MIDA+MS	**5.19**	**43**	**213**	**22**
4	MFCC+MS	6.31	64	249	25

Tab. 7.5: WER (%) for the ESAT-recognizer tested on WSJ with different features and different settings for the MIDA preprocessor.

7.6 Combination of two feature sets

If I can show that the ACF and PHF-driven systems behave differently, then I have an argument for investigating whether a combination of the systems would lead to a further improvement of the ASR performance. In order to show this, I have compared the errors made by the two TIMIT-systems (best configuration for each) and I found (see Table 7.6) that for 5.3% of the words, the ACF and the PHF-based ASR generate a different result. If I would be able to correct all the errors of the ACF system that correspond to a correct solution in the PHF system, and if I would be able to avoid the introduction of new errors at other places, the WER could be reduced from 3.7% to 2.6% (relative improvement = 30%). Obviously I will not be able to conceive such a good combination strategy. On the other hand, the maximum attainable improvement may be larger if not only the top-1 hypotheses but the top-N hypotheses of the individual ASR-systems were taken into account.

error type	word count	(%)
both correct	1470	93.6
ACF wrong and PHF correct	18	1.15
ACF correct and PHF wrong	47	3.00
both wrong, different errors	18	1.15
both wrong, same errors	17	1.10
total	1570	100

Tab. 7.6: Number of word errors in the outputs of MFCC and PHF recognizers.

Since the potential seemed to be large enough I have investigated 2 means of combining the two feature sets. This study is described now.

7.6.1 Word-level combination

One way of combining the two systems consists of trying to merge the word hypotheses generated by two independently working systems, a PHF and a ACF-driven system. This word-level combination can be based on three outputs of each system.

1. Single-best recognition result
2. N-Best list

3. Word graph (lattice)

For the combination of the one-best hypotheses of two different recognizers one can for instance apply the ROVER (Recognizer Output Voting Error Reduction) technique [37]. ROVER constructs a new word transition network from the two best word sequences and then rescores the new transition network by means of a voting module. The voting module parses the word network from left to right and, at each node, chooses the best-scoring outgoing arc. This rescoring procedure is based on the confidence values (ranging between 0 and 1) assigned to the arc labels by the respective recognizers. Several scoring schemes can be tested. One of the main drawbacks of ROVER is that it uses a dynamic programming alignment procedure which does not take into account the absolute time alignment of the individual word sequence hypotheses. It just finds the minimal cost match between two strings based on insertion, deletion and substitution penalties. In [62] an extension which can take the timing information into account is proposed.

Nevertheless I argue that a method combining word graphs instead of single-best hypotheses should perform better. The advantage of combining word graphs over single-best hypotheses is that the best hypothesis in the product graph may be based on partial paths from the second-best or third-best hypothesis from both individual word graphs. That is why I developed such a combination method.

7.6.2 Generation of the Product Lattice

Since each of both recognizers is able to produce a recognition result in the form of a word graph (I will use the HTK graphs mentioned in section 2.3), I investigated the possibility of combining the word graphs into one new graph which I will call the *product graph*. Let the output graphs of recognizers 1 and 2 be G_1 and G_2 respectively, and the product graph G_p. I will now explain the algorithm to construct the product lattice. Let $\mathcal{N}_k$ represent the set of nodes of G_k $(k = 1, 2)$. Each node n_i is characterized by a time t_{n_i} and by a word label W associated with all arcs arriving at this node. The algorithm is composed of the following steps.

```
Consider the product set $\mathcal{N} = \mathcal{N}_1 \times \mathcal{N}_2$
**for** all $(n_i, n_j) \in \mathcal{N}$ **do**
        **if** $|t_{n_i} - t_{n_j}| < \epsilon$ **and** the word label associated with the arcs
```

```
          arriving at node $n_i$
          is equal to the word label associated with the arcs arriving at
          node $n_j$ **then**
              create a new node $n = (n_i, n_j)$ and add it to $\mathcal{N}_p$
          **else**
              continue

    **end for**
    **for** all $n = (n_i, n_j) \in \mathcal{N}_p$ **do**
          **for** all $m = (n_k, n_l) \neq n$ with $t_{n_i} < t_{n_k} - \epsilon$ and $t_{n_j} < t_{n_l} - \epsilon$
          **do**

              **for** $G_1$ consider all original paths connecting
              $n_i$ with $n_k$
                  **if** a path consist of maximum $N$ arcs **then**
                  create those paths also between $n$ and $m$

              **for** $G_2$ consider all original paths connecting
              $n_j$ with $n_l$
                  **if** a path consist of maximum $N$ arcs **then**
                  create those paths also between $n$ and $m$

          **end for**
    **end for**
```

This algorithm has two free parameters: a parameter ϵ specifying the temporal margin (in seconds) between corresponding nodes in the two graphs, and a parameter N which is the maximum number of intermediate arcs between two product nodes that will be considered in the product graph. Figure 7.3 is a graphical representation of this algorithm.

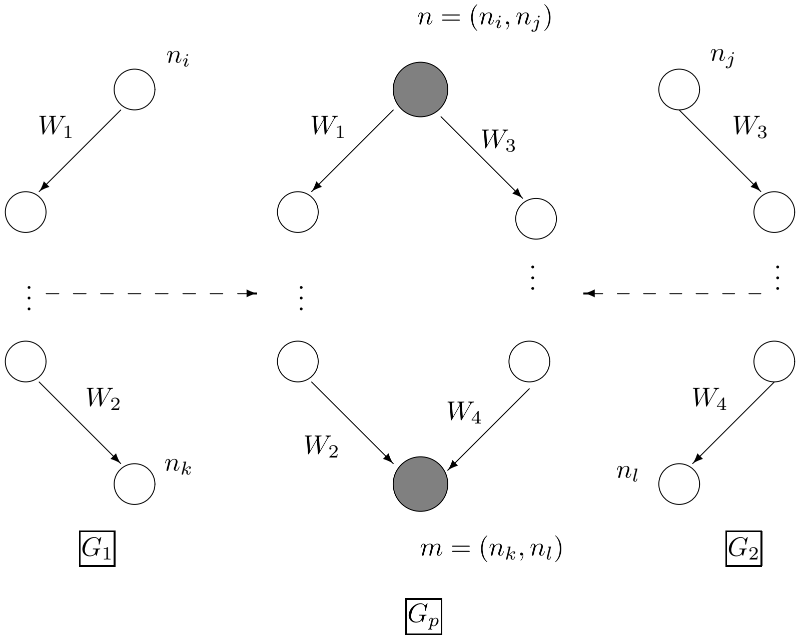


Fig. 7.3: Graphical representation of the word-level combination algorithm.

The product graph consists thus of arcs that originate either from G_1 or G_2. Each arc of G_i ($i \in [1, 2]$) has a total log-likelihood LL attached to it which is defined as

$$\mathrm{LL} = \mathrm{LL}_{ac} + s_{LM}\mathrm{LL}_{lm} + p_W \tag{7.47}$$

LL_{ac} is the acoustic likelihood, s_{LM} is the language model scale factor, LL_{lm} is the language model probability and p_W is the word insertion probability. Then some kind of score has to be attached to each arc of G_p. In this respect it has to be noticed that the two systems are using different (acoustic) models trained on different feature spaces and therefore that the acoustic likelihoods of the word arcs of G_1 and G_2 are not necessarily directly compatible. I have investigated two strategies to deal with this problem.

1. Normalization of acoustic log-likelihoods
   The first way to account for this is to transform the acoustic likelihoods emerging from G_1 and G_2 in such a way that the mean acoustic likelihoods in G_1 and G_2 are equal and that they have the

same variance.

$$\{\mathrm{E},\mathrm{Var}\}[a\,\mathrm{LL}_{ac,1} + b] = \{\mathrm{E},\mathrm{Var}\}\mathrm{LL}_{ac,2} \tag{7.48}$$

These acoustic log-likelihood will be used in the product graph. The a and b parameters are re-estimated per utterance. The language model scale factors used by the two systems were the same, so I can use the same value in the product graph. However since the word insertion probabilities differed, I used the mean word insertion probability in the product graph. Finally, the total log-likelihoods LL of arcs originating from system 1 will be scaled with a confidence factor α. This can account for the fact that one system is more reliable than the other. In my case the ACF system is the system with the lowest WER and thus the more reliable one.

2. Combination of acoustic likelihoods
The second and more elegant way to solve the problem of incompatible likelihoods is to combine acoustic log-likelihoods emerging from system 1 and 2 in some way and to attach a combined acoustic log-likelihood to the arcs in G_p. Suppose that I would want to combine the acoustic likelihood $\mathrm{LL}_{ac,1}$ of a certain arc a of G_1 with the acoustic likelihood that would emerge from system 2 when it was hypothesized in the same time interval. Let the latter log-likelihood be represented by $\mathrm{LL}_{ac,1|2}$. Then a combined acoustic log-likelihood of the form

$$\mathrm{LL}_{ac,p} = \alpha\mathrm{LL}_{ac,1} + (1-\alpha)\mathrm{LL}_{ac,1|2} \tag{7.49}$$

can be attached to the arcs in G_p. This method requires that all arcs of G_1 must be rescored by system 2 and vice versa. The parameter α can be considered as a weight that takes into account the differences in accuracy of the two systems and hence their respective confidence.

One would expect that the acoustic likelihoods $\mathrm{LL}_{ac,p}$ attached to arcs in G_p should be normalized for the duration $t_{n_k} - t_{n_i}$ (or $t_{n_l} - t_{n_j}$), because the Viterbi algorithm that seeks the best path in G_p may base its decision for the best partial path on the unequal number of consumed frames in two partial paths. However I chose to add to the total log likelihood of paths that were created between the same n and m, a mean log likelihood per frame multiplied by the time difference between the paths. However, if ϵ is chosen smaller than (or equal to) the frame rate, then all paths between n and m will have consumed a same number of frames, so I never have to add such a mean log likelihood.

Since there are lots of arcs in both graphs and since I want the algorithm to work in reasonable time, I propose to prune the incoming graphs in some way. As a first pruning strategy, I chose to retain only arcs with 'correct' word labels. A correct word label is defined as a word belonging to either one of the two single-best hypotheses emerging from the two recognizers. This drastically reduces the number of arcs that have to be investigated.

The single-best hypothesis found by the Viterbi algorithm in G_p is considered as the new recognition result. The optimal value for α (for both likelihood strategies) is determined by a grid search.

7.6.3 Word-level combination experiments

I have performed word-level combination experiments on TIMIT-ACF (system 1) and TIMIT-PHF (system 2). I used both the baseline ACF system with a WER of 4.59% and the best ACF system using MLLT and a WER of 3.69% as system 1. The PHF system was always the one with a WER of 5.29%. It is anticipated that the word-level combination technique works best when the WERs of both systems are not too different. The WER of the PHF system differs by 15% from that of the baseline ACF system, whereas it differs 43% from that of the best ACF system. This is why I first combined the baseline ACF system with the best PHF system.

I took $\epsilon = 0.02$ and $N = 6$. The results for the two likelihood strategies are represented in Table 7.7.

	baseline ACF and PHF					best ACF and PHF				
strategy	α	WER (%)	D	S	I	α	WER (%)	D	S	I
1	1	4.46	18	43	9	0.97	3.95	14	40	8
	0.99	**4.27**	**16**	**45**	**6**	0.96	3.76	13	38	8
	0.98	4.33	15	48	5	0.95	**3.69**	**15**	**36**	**7**
	0.97	4.46	16	48	6	0.94	3.89	18	36	7
2	0.51	4.90	23	48	6	0.51	4.78	22	46	7
	0.50	4.46	22	44	4	0.50	**4.08**	**21**	**36**	**7**
	0.49	**4.27**	**19**	**45**	**3**	0.49	4.46	23	39	8
	0.48	4.71	22	47	5	0.48	4.46	23	39	8

Tab. 7.7: WER (%) for the word-level combination technique.

The Table indicates that when combining the ACF baseline with the best PHF system, the WER of the combined system can be brought to a level which is slightly below the ACF baseline. However the obtained improvement of 7% relative is only marginal and not statistically significant for this database. Kirchhoff reached a relative improvement of only 3.7% with her word-level combination technique on the Verbmodil corpus (baseline of 29.03%). Combining the best ACF system with the best PHF system did not lead to any improvements at all. It can also be seen that both likelihood strategies reach the same combined WER when the baseline ACF and best PHF system are combined. However, when the best ACF and PHF systems are combined the first likelihood strategy performs better. Since word-level combination does not seem to offer any gain, I did not test it on other databases (like e.g. WSJ). Instead I developed a state-level combination technique which is described in the next section.

7.6.4 State-level combination

Suppose that q represents a state of a baseline triphone acoustic model, and that $\log p_A(\mathbf{x}|q)$ is the log-likelihood of acoustic vector $\mathbf{x}$ in this state. Then I propose to replace the baseline acoustic model score by a two-stream log-likelihood score

$$\mathrm{LL}(\mathbf{x}|q) = g_{1q} \log p_A(\mathbf{x}|q) + g_{2q}\ [\alpha\ \ \log p_B(\mathbf{x}|q) - \beta] \tag{7.50}$$

with $\log p_B(\mathbf{x}|q)$ representing the log-likelihood computed by means of a phonologically inspired context-independent model. I consider this model as a kind of back-off model because it is anticipated to be less discriminative than the acoustic model. The phonological scores are being used to 'correct' the ACF-scores in cases where the ACF-scores are not reliable.

g_{1q} and g_{2q} are the **state dependent** stream weights, and (α, β) are normalization coefficients whose role will be explained in a moment.

7.6.4.1 Phonological feature models

In section 5.4, I introduced a phonological feature set of 25 binary phonological features (PHFs) to characterize acoustic-phonetic units. These features are denoted as f_i $(i = 1, .., 25)$ and are grouped in four feature subsets: (1) **vocal source** (voiced, unvoiced, inactive), (2) **manner** (closure, vowel, fricative, burst, nasal, approximant, lateral, silence), (3) **place-consonant** (labial, labio-dental, dental, alveolar, post-alveolar, velar, glottal) and (4) **vowel-features** (low, mid-low, mid-high, high, back, mid, front, retroflex, rounded). Posterior probabilities $P(f_i|\mathbf{x})$ are estimated by a configuration of four neural networks (see section 5.4 for more details).

7.6.4.2 Computing phonological scores

In order to determine $p_B(\mathbf{x}|q)$ I need to characterize each state q of a baseline HMM by its phonological features. For most phonemes, all states of the phoneme inherit the phonological features of this phoneme. However, some phonemes like plosives for instance, are modeled in terms of two acoustic-phonetic units with different phonological feature sets. The state q of such a phoneme then takes the phonological feature set of the acoustic-phonetic unit that best explains the acoustic observations assigned to this state during an alignment of the training utterances with their orthographic transcriptions.

Since the phonological feature models compute posterior probabilities, log-likelihoods will be obtained as

$$\log p_B(\mathbf{x}|q) = \log \frac{P_B(q|\mathbf{x})}{P_B(q)} + \log p(\mathbf{x}) \tag{7.51}$$

where the subscript B indicates that these are probabilities according to the phonological model. Substituting this in Equation (7.50) leads to

$$\begin{aligned} \mathrm{LL}(\mathbf{x}|q) = g_{1q}\ \log p_A(\mathbf{x}|q) + \alpha\ g_{2q}\ \log p(\mathbf{x}) \\ + g_{2q}\ [\alpha\ \log \frac{P_B(q|\mathbf{x})}{P_B(q)} - \beta] \end{aligned}$$

I now assume that the second term is much less dependent on q than the other terms (just g_{2q} can depend on q), and I use

$$\mathrm{LL}(\mathbf{x}|q) = g_{1q} \log p_A(\mathbf{x}|q) + g_{2q}[\alpha\ \log \frac{P_B(q|\mathbf{x})}{P_B(q)} - \beta] \tag{7.52}$$

as the two-stream score. Now it is time to explain what the role of (α, β) is. I first aligned the training data with the baseline models so that each frame was assigned to a state q. Then α and β is chosen such that

$$\{\mathrm{E,Var}\}[\alpha \log \frac{P_B(q|\mathbf{x})}{P_B(q)} - \beta] = \{\mathrm{E,Var}\}[\log p_A(\mathbf{x}|q)] \tag{7.53}$$

taken over all frames. This makes the two stream scores more equivalent, and the interpretation of (g_{1q}, g_{2q}) as stream importances more plausible. The search for the optimal stream weights can then be restricted to $g_{1q} + g_{2q} = 1$.

Given the phonological description of q, the feature set can be divided in two subsets: P_q = the set of *positive* features that are supposed to be *on* ($\mathrm{card}P_q = N_{qp}$), and N_q = the set of negative features that are supposed to be *off* for that state ($\mathrm{card}N_q = N_{qn}$). Since I showed in section 6.2.2 that the irrelevant features do not contribute to the cost, I discard them from the computations. Assuming independent phonological features then leads to the following expression:

$$\log \frac{P_B(q|\mathbf{x})}{P_B(q)} = \sum_{f_i \in P_q} \log \frac{P(f_i|\mathbf{x})}{P(f_i)} + \sum_{f_i \in N_q} \log \frac{1 - P(f_i|\mathbf{x})}{1 - P(f_i)} \qquad (7.54)$$

Because a statistical analysis of real data has shown that the two components in the right hand side of expression (7.54) are correlated (correlation coefficient of 0.75), it makes sense to consider only one component as estimator. However when I use the positive and negative features only, there is a danger that the contributions of the phonological scores to the two-stream score on different states are not compatible. This incompatibility can be avoided by considering the mean phonological score per feature. I have investigated in particular what happens if only positive or negative features are retained and I found that using the sum of the mean of the two as the ultimate estimator

$$\log \frac{P_B(q|\mathbf{x})}{P_B(q)} = \frac{1}{N_{qp}} \sum_{f_i \in P_q} \log \frac{P(f_i|\mathbf{x})}{P(f_i)} + \frac{1}{N_{qn}} \sum_{f_i \in N_q} \log \frac{1 - P(f_i|\mathbf{x})}{1 - P(f_i)} \qquad (7.55)$$

yielded the best results. It is with this setting that I tried this combination approach on TIMIT and WSJ with a state independent $g_{1q} = g_1$ and $g_{2q} = g_2 = 1 - g_1$. The weight g_1 was varied from 1 to 0.5 and the best performing weight on some training utterances was used during the final test. The recognition results are represented in Table 7.8. It can be seen that I only obtain a marginal improvement for TIMIT but none for WSJ. An explanation for this negative result will be provided in the next section. Nevertheless, the result for TIMIT seems to indicate that there is a small form of complementarity between the two models. Kirchhoff reached a more substantial relative improvement of 5.6% with a similar technique on the Verbmobil corpus (baseline of 29.03%).

database	system	g_1	g_2	WER (%)	D	S	I
TIMIT	MLLT	1.0	0.0	3.69	17	36	5
	+ SLC	0.9	0.1	**3.45**	**18**	**33**	**3**
WSJ	MLLT + CMS	1.0	0.0	**5.64**	**47**	**222**	**33**
	+ SLC	0.9	0.1	5.74	58	219	30

Tab. 7.8: WER (%) for the state-level combination (SLC) technique tested on TIMIT and WSJ

7.7 Combination applied to Spoken Name Recognition

The state-level combination technique proposed in the previous section turned out not to be very successful for CSR, but I argue that for special tasks where the baseline models are not applied in their normal operating point, the phonological back-off model should be able to give a larger benefit. This situation occurs when the utterance can contain foreign phonemes that were not present in the training utterances that were used during the acoustic model training. In that case, a back-off model operating with presumably language-independent features (PHFs) can offer interesting information. Another situation in which the 'phonological' stream could be helpful is when there are a lot of native phonemes appearing in contexts that are uncommon in modal speech. In view of these arguments I have tested the state-level combination technique for the automatic recognition of spoken names.

7.7.1 Problem statement

It is a challenge to develop an automatic speech recognizer (ASR) that can accurately recognize proper names (e.g. person names, city names, street names, etc.) because in most applications (e.g. navigation, directory assistance) there is a huge number of names involved, and it would be extremely expensive to elicit from human experts typical phonetic transcriptions for all these names. Hence, one must rely on an automatic grapheme-to-phoneme (G2P) converter instead. Unfortunately commercially available G2P converters were designed to transcribe the regular words of a language. When confronted with foreign names, they often do not produce an acceptable output. Recent experiments on the transcription of person and geographical names occurring in the Netherlands showed that the state-of-the art Dutch G2P converter of Nuance was unable to produce an acceptable phoneme sequence (one of the manual transcriptions present in a lexical database) for about 30% of these names. When also considering wrong lexical stress assignments as errors, the error rate further increased to 50% [117].

Even if the G2P converter could be improved, there would still be a problem because there is clear evidence (e.g. [39]) that, depending on their familiarity with the language of origin, native speakers may use different pronunciations of a foreign name. These pronunciations can range from totally *nativized* pronunciations (using native phonemes and native G2P rules) to totally *foreignized* pronunciations (using foreign phonemes and foreign G2P rules). I therefore argue that the ASR should incorporate lexical and acoustic models that can cope with this type of pronunciation variability.

In [72] one proposes to use multiple G2Ps to produce multiple pronunciations of a name: one G2P for the native language and one for each likely language of origin of the name. Obviously, the outputs of the non-native G2Ps must be converted to native phoneme sequences that are compatible with the acoustic models of the ASR which was trained on native speech only. Adding the obtained pronunciations to the baseline dictionary usually causes a significant reduction of the word error rate (WER) (see below).

In [12], one also creates pronunciation variants, but this time in a data-driven way. This is achieved by using native acoustic models to align each name utterance with a graph of available initial pronunciations of that name (6 per name) as identified on the basis of expert knowledge. By seeking alternative phonemes for modeling the regions where the acoustics badly match the graph, new pronunciations were created. Including these pronunciations in the lexicon resulted in an improvement of the name recognition error rate by 20 to 40% relative. However, these figures may be optimistic because the tests were run on the same names that were also used to learn the new pronunciations.

A number of authors [9] argue that in order to perform well, some non-native phonemes should be kept in the phonetic transcriptions and separate acoustic models should be created for these phonemes. In [97] for instance, models of English phonemes that have no good German equivalent were trained on English speech spoken by German speakers and added to the inventory of acoustic models. By doing so the WER on a corpus of German sentences containing at least one English name dropped from 60 to 44%.

In [47], non-native pronunciation variants for names of an English origin are generated in a totally data-driven way. An English phoneme recognizer generates English pronunciations, and by aligning these pronunciations with the canonical pronunciations emerging from a German G2P converter, one obtains training examples for the automatic learning of decision trees that can be used for the generation of English-accented pronunciation variants. This method however only yields a small drop (5.2% relative) of the WER .

In cases where names from several languages have to be recognized, an approach that needs foreign phoneme models spoken by native speakers for each of these languages may turn out to be impractical. In that case one can try to create acoustic models for all the sounds in the IPA

(International Phonetic Alphabet) and use these models for the mapping of foreign phonemes to symbols that have an associated acoustic model (e.g. [63]).

Here I propose a novel method that is a bit related to the just mentioned IPA approach, in the sense that it uses a phonologically motivated back-off score in combination with the traditional acoustic likelihoods. I hereby rely on an earlier finding [122] that phonological feature models learned on native speech are also capable of characterizing foreign sounds.

I will first elaborate and motivate my model and I will then assess it on a substantial trilingual spoken name corpus. I will also demonstrate the capabilities of my method in combination with phonological models that were trained on multilingual instead of native speech data, because it was shown in [109] that such models are more reliable than monolingually trained models.

7.7.2 Methodology

The methodology starts with implementing the state-level combination technique, explained in section 7.6.4. However, I will extend the technique by introducing the concept of foreignizable phonemes.

7.7.2.1 Determination of the stream weights

In order to determine optimal stream weights for each state q, I could conceive an automatic weight optimization scheme. However, before starting to develop such a scheme, I will investigate what can be achieved with state-independent stream weights (g_1, g_2) which are optimized by tracking the WER, measured on a development set, as a function of $g_2 = 1 - g_1$, and by selecting the value yielding the minimal WER. The optimal stream weights will be used for the experiments in section 7.7.3.

7.7.2.2 Foreignizable phonemes

The standard transcription of a foreign name is normally obtained from its foreign transcription by mapping all foreign phonemes to their best equivalent in the native phoneme inventory. However, if this equivalent does not have the same phonological feature representation as the original phoneme, I consider the chosen equivalent as *foreignizable* to that original, meaning that it can be pronounced as a foreign phoneme. This is indicated in the lexicon by adding the foreign phoneme as an extension to the chosen native equivalent. Let me illustrate that for the case of Dutch as the native and English as the foreign language. When /r_rr/

English phoneme	Dutch equivalent	#occs in lexicon	French phoneme	Dutch equivalent	#occs in lexicon
Q	A	46	E	Ei	79
V	@	27	9	Y	2
3‘	Y r	30	e∼	E N	70
aI	A j	48	9∼	Y N	4
@U	O w	78	o∼	O N	74
rr	r	225	a∼	A N	92
			R	r	261

Tab. 7.9: English and French phonemes (SAMPA notation, see http://www.phon.ucl.ac.uk/home/sampa) but with /rr/ and /r/ as symbols for the English and Dutch /r/) for which the Dutch equivalent has a different phonological representation. The 6 English foreignizable phonemes account for 454 occurrences in the Autonomata lexicon, the 7 French phonemes occur 582 times.

appears in an English name that is part of a Dutch lexicon, it means that the Dutch phoneme /r/ (from the Dutch word *oo***r**) was obtained as an approximation of the English /rr/ (from the English word *o***r**) and that the acoustic score must be obtained by combining the model score emerging from a triphone model with /r/ as the central phoneme and a back-off score computed on the basis of the PHFs of /rr/. Note that it can happen (see Table 7.9) that two subsequent phonemes (e.g. the Dutch /Y/ (from *b***u***s*) + /r/) originate from just one foreign phoneme (e.g. the English /3:/ from *b***i***rd*) and vice versa.

The number of foreignizable phonemes depends on the (native, foreign) language combination: for (Dutch, English) I found 6 foreignizable phonemes, for (Dutch, French) I found 7.

name	transcription	
Burr Tupper	baseline	b Y r _ t Y p @ r
	alternative	b Y_3: r_3: _ t Y p @ r
Alan Presser	baseline	E l @ n _ p r E s @ r
	alternative 1	E l @ n _ p r_rr E s @ r
	alternative 2	E l @ n _ p r E s @ r_rr
	alternative 3	E l @ n _ p r_rr E s @ r_rr

Tab. 7.10: Two English names with their baseline and alternative native transcriptions.

7.7.2.3 Introduction of pronunciation variants

Foreignizable phonemes can also form a basis for the generation of pronunciation variants in the lexicon. A simple way to accomplish this is to produce alternative pronunciations by replacing one or more foreignizable phonemes by their pure native equivalents. Table 7.10 shows two names and the variants that were created for them in this way. The underlying motivation is that the user may adopt a nativized pronunciation for all or just for some of the foreignizable phonemes. In that case it may be advantageous to let the recognizer decide where to select nativized and where to select foreignized pronunciations.

7.7.3 Experiments

The experiments in [106] were restricted to the recognition of English names by a Dutch speech recognizer, and the number of different English names was quite limited. In this section I describe tests which were run on a much larger corpus of spoken names, and I report results for English, French and Dutch names, uttered by Dutch speakers.

The spoken name corpus was recorded in the AUTONOMATA project that was funded by the Dutch-Flemish STEVIN program [28]. The database will soon be made publicly available by the Dutch-Flemish Language & Speech Technology Center (www.tst.inl.nl). In the present study I selected the 60 Dutch speakers from Flanders (one of the two regions in Europe were Dutch is spoken). Each speaker uttered one of 10 lists of 120 Dutch, 23 English, 23 French and 15 Moroccan names and there was no overlap between these 10 name lists. One third of the speakers was between 12 and 18 years old, the remaining speakers were adults. The names were either person names (first name + family name), city names or street names.

In the present study the Moroccan names were omitted and the remaining data was divided in an adaptation set, a development set and a test set (see chapter 2).

In all experiments the ASR had a vocabulary of 1660 names: 1200 Dutch, 230 English and 230 French names and there was no overlap between the names in the development set and the test set. The ASR is assumed to have no prior knowledge of the language of origin of the names it has to recognize. The acoustic models are triphone models: either speaker-independent models (SIMs) that were trained using HTK [127] on a multi-speaker read speech corpus recorded in the Flanders [26] (CoGeN), or adapted models (AMs) obtained from these SIMs by MLLR adaptation to an adaptation set extracted from the spoken name corpus.

During adaptation I trained a set of model transformation matrices according to the procedure explained in the HTK-book. In this tutorial the number of leaf nodes of the regression tree was chosen to be 32, which I copied. In combination with the AMs, I used a PHF detector that was also adapted to the adaptation set. This was obtained by performing 10 extra training epochs of the MLPs on the adaptation data. Since there were no manual phoneme labels available for the adaptation data, I first had to segment and label the adaptation data using my segmenter and the unadapted networks.

Although the adaptation set contains the same speakers as the test set, it was verified in a separate experiment with a smaller test set and no speaker overlap between the adaptation set and the test set, that the WERs on the test set were very similar in that case. This means that the models are not so much adapting to the test speakers, but mainly to the acoustic circumstances appearing in the spoken name corpus recordings.

I will now describe the baseline experiments that have been run, and after that, the experiments that were conducted to assess the capabilities of my method.

7.7.3.1 Setting up a baseline system

In the baseline system, no back-off models nor pronunciation variants were created, but the effect of using different types of transcriptions in the lexicon was investigated. To that end I had available the Dutch, English and French versions of the Nuance G2P-converter, and a typical transcription of each name. The latter is a transcription that is delivered with the corpus and that, according to a human expert, is a likely and acceptable transcription of the name. It is hereafter called a manual transcription. Using these resources I composed the following lexicons:

DuAlone	all names transcribed by Dutch G2P
All	all names transcribed by three G2Ps
ManAlone	manual transcriptions of all names
DuMan	merge of DuAlone and ManAlone
AllMan	merge of All and ManAlone

The corresponding word (name) error rates obtained with the two acoustic model sets on the different parts of the test set are listed in Tables 7.11 (SIMs) and 7.12 (AMs).

The most important finding is that foreign G2Ps produce much better transcriptions of foreign names than the native G2P, even with the foreign phonemes being mapped to native phonemes. This can only mean that a

lexicon	English	French	Dutch	All
DuAlone	61.7	43.3	19.3	35.9
All	**50.8**	**32.5**	**21.3**	**31.5**
ManAlone	45.1	47.5	17.5	31.9
DuMan	42.7	37.4	17.7	28.9
AllMan	47.0	33.5	19.8	30.0

Tab. 7.11: Baseline performances (WER in %) obtained with the speaker independent acoustic models in combination with the different lexicons.

lot of native speakers adopt foreign name pronunciations that are closer to foreignized than to nativized pronunciations.

A second finding is that for French the manual transcriptions (ManAlone) perform a lot worse than the transcriptions generated by the foreign G2Ps.

A third finding is that the Dutch transcriptions are indispensable to get a good result: *DuMan* significantly outperforms *ManAlone*.

A last finding is that the foreign G2Ps do not attribute much anymore if the Dutch and manual transcriptions are already in the lexicon. For the Dutch names they are useless and only augmenting the lexical confusion, whereas for foreign names there is a balance between that effect and the positive effect of bringing in a better transcription than the manual one for some of these names. There is a small improvement for French names. Together with finding 2 this may indicate that the manual transcriptions of the French names are maybe not as good as the other manual transcriptions.

7.7.3.2 Testing the proposed methodology

Since one usually has no access to manual transcriptions I take *All* as the baseline lexicon and I assess my methodology when applied in combination with this lexicon. Figures in bold in the Tables refer to results that are significantly better than the baseline according to a Wilcoxon signed-rank test [23] with $p = 0.05$.

7.7.3.3 Back-off model with native phoneme representations

In a first experiment (called NATIVE), I just took the lexicon *All* as used in the baseline system. The phonological representations that served as a basis for the computation of the back-off scores (see section 7.6.4)

lexicon	English	French	Dutch	All
DuAlone	33.7	23.4	4.2	16.4
All	**20.7**	**12.8**	**4.4**	**10.6**
ManAlone	15.7	30.7	3.7	13.4
DuMan	13.3	17.8	3.7	9.6
AllMan	15.1	14.8	3.9	9.4

Tab. 7.12: Baseline performances (WER in %) obtained with the adapted acoustic models in combination with the different lexicons.

were those of the native phonemes that gave rise to the model states. I determined the optimal stream weight by performing recognition tests on the development set for several values of g_2. I tracked the WER as a function of g_2, smoothed the curve and located the minimum of the smoothed curve. The corresponding stream weights were then imputed in the ASR-system. The optimal stream weights were $(g_1, g_2) = (0.2{,}0.8)$. The corresponding WERs plus the absolute and relative improvements (AI and RI) over the baseline are summarized in Table 7.13. The first remarkable fact is that the improvement is modest in the SIM case but substantial in the AM case. Possibly, my method is not effective as long as the baseline acoustic models are insufficiently accurate.

triphones	measure	E	F	D	All
SIMs	WER	**47.3**	30.8	21.0	**30.0**
	AI	**3.5**	1.7	0.2	**1.5**
	RI	**6.9**	5.2	1.0	**5.5**
AMs	WER	**18.3**	**10.2**	**3.1**	**8.7**
	AI	**2.4**	**2.6**	**1.3**	**1.9**
	RI	**11.6**	**20.3**	**29.5**	**17.9**

Tab. 7.13: Performances (all in %) of an ASR with a two-stream acoustic model and native phonological representations of the model states (experiment NATIVE, AI = Absolute Improvement, RI = Relative Improvement).

A second remarkable fact is that in the AM case, the improvement is not only substantial for English and French names, but even more substantial for Dutch names. Apparently, the back-off model provides information that is not captured by the triphone model. I will come back to this issue later.

7.7.3.4 Back-off model with foreign phoneme representations

In a second experiment (called FOREIGN-UNIQUE), I replaced the former *All* lexicon by a lexicon with foreignizable phonemes in the foreign G2P outputs. Then I used the phonological characterization of the foreign phonemes to control the back-off score computation. The results of this experiment are summarized in Table 7.14. Apparently, the introduction of foreign phonological representations causes only a small gain, but it is a consistent one that is achievable at no extra cost.

triphones	measure	E	F	D	All
SIMs	WER	**46.7**	30.1	21.2	**29.8**
	AI	**4.1**	2.4	0.1	**1.8**
	RI	**8.0**	7.4	0.7	**5.5**
AMs	WER	**18.1**	**10.1**	**3.1**	**8.6**
	AI	**2.6**	**2.7**	**1.3**	**2.0**
	RI	**12.6**	**21.1**	**29.5**	**18.9**

Tab. 7.14: Performances (all in %) of an ASR with a two-stream acoustic model and foreignizable phonological representations of the model states (experiment FOREIGN-UNIQUE).

One of the possible explanations for the low gain is that the speakers not always use a foreign pronunciation, and thus that a back-off model on the basis of a foreign representation is not always offering the best solution. In order to test that hypothesis I have conducted an additional experiment.

7.7.3.5 Including pronunciation variants

In a third experiment (called FOREIGN-VARS) I have introduced pronunciation variants in the lexicon using the method proposed in section 7.7.2. The recognition results obtained with this lexicon are summarized in Table 7.15.

The Table reveals that the results have further improved, and that the improvement is now starting to be statistically significant for the speaker-independent case as well. Note too, that the improvement due to the pronunciation variants is confined to the English and French name subsets, as expected. However, the gain is moderate and adding variants is increasing the computational load. I therefore recommend to use the system with foreignizable representations but without variants.

triphones	measure	E	F	D	All
SIMs	WER	**45.9**	**29.4**	21.4	**29.5**
	AI	**4.9**	**3.1**	-0.1	**2.0**
	RI	**9.6**	**9.5**	-0.0	**6.3**
AMs	WER	**17.6**	**10.0**	**3.1**	**8.5**
	AI	**3.1**	**2.8**	**1.3**	**2.1**
	RI	**14.9**	**21.9**	**29.5**	**19.8**

Tab. 7.15: Performances (all in %) of an ASR with a two-stream acoustic model, foreignizable phonological representations of the model states and pronunciation variants (experiment FOREIGN-VARS).

7.7.3.6 Discussion of results

In order to better understand why the two-stream model is working, I analyzed the results in more detail. Let me first try to explain why the system with native phoneme representations yields an improvement. The improvement obtained for the English and French names may be intuitively attributed to the occurrence of foreignizable phonemes. But why does the system improve for the Dutch words when using my AMs?

To answer this question I compiled a list of the N most frequent triphones occurring in the training data. Then I counted for each name n under test the number of triphones in that name which - according to the chosen transcription - did not occur in the list. Let's call this number $\#U_{triphs,n}$ and the total number of triphones in the name $\#T_{triphs,n}$. I define the ratio

$$R_n = \frac{\#U_{triphs,n}}{\#T_{triphs,n}} \tag{7.56}$$

as the percentage of unlikely triphones in the word. After averaging over all names n, I get

$$R = \frac{1}{\#\text{names}} \sum_n R_n \tag{7.57}$$

This indicator should tell something about the dissimilarity of phonotaxis between training and test data. I now calculated R in the following three situations:

1. for all names to be recognized

2. for all wrongly recognized names by the baseline system

3. for names that were corrected by system NATIVE

For the Dutch words, the baseline system made 88 errors, the NATIVE system made 62 errors. Six new errors were introduced, while 32 were corrected. Hence 56 errors remained unaltered. I varied N from 5000 to 10000 in steps of 1000 and listed the value of R in Table 7.16 in the above three situations. The chosen transcriptions are those emerging from the Dutch G2P. The Table clearly reveals that there is a significant difference in R between situation (1) on the one hand and situations (2) and (3) on the other hand. More in particular, I also notice that the relative difference in R between (1) and (2) grows as N increases. This means that some Dutch names may be wrongly recognized due to the fact that they comprise triphones that were seen less frequently during training. The back-off model however models PHFs and not phonemes. Suppose /a-b+c/ is a rare triphone and suppose further that PHF "A" of /b/ is shared with phoneme /d/. Phoneme /d/ may occur in exactly the same context, namely /a-d+c/ in the training. As a result I have seen two instances of /a-A+c/, thus the training of PHF "A" in that particular context could be done more reliably than the training of the triphone /a-b+c/. In such case the back-off model gives a more reliable estimate. Since the errors are characterized by a higher amount of unreliable triphones, the back-off model may correct some of them. I hypothesize that it is this effect that is responsible for the improvement of the NATIVE system on the Dutch names. I then did the same analysis, but replaced the triphones by monophones or simply by 44 phonemes. N was varied from 10 to 35 in steps of 5 and R was measured again. The results can be seen in Table 7.17

	N					
situation	5000	6000	7000	8000	9000	10000
(1)	22.2	19.4	17.5	15.2	13.1	11.4
(2)	26.7	23.5	22.1	18.6	17.6	15.6
(3)	26.8	24.8	21.4	18.5	17.8	15.6
(2) vs. (1)	+19%	+21%	+26%	+22%	+34%	+37%

Tab. 7.16: R-indicator (in %) for different values of N (the N most frequent triphones seen in the training data). Words were extracted from the Dutch names in the Autonomata tests.

For $N = 35$, I found a relative difference of 48% between situation (1) and (2), which indicates that the errors not only occur in names with rare triphones, but even more in names with less frequently seen phonemes.

situation	N					
	10	15	20	25	30	35
(1)	43.8	32.5	16.4	8.4	3.2	0.9
(2)	42.0	30.5	18.0	9.2	4.6	1.3
(3)	43.8	28.1	15.0	7.8	4.1	2.1
(2) vs. (1)	-4%	-6%	+9%	+8%	+47%	+48%

Tab. 7.17: R-indicator (in %) for different values of N (the N most frequent phonemes seen in the training data). Words were extracted from the Dutch names in the Autonomata tests.

In a similar way as for the triphones I can argue that the back-off model is able to yield a better estimate of the likelihood of rare phonemes than the triphone acoustic models can. I subsequently compared R in the following three situations.

1. a list of 70492 Dutch words extracted from the Mediargus corpus [78]. The list is composed of 22957 names and 47535 regular words.
2. the list of 47535 regular words
3. the list of 22957 names

Again I varied N and took the Dutch G2P transcriber to generate the transcriptions. The result can be seen in Table 7.18.

situation	N					
	5000	6000	7000	8000	9000	10000
(1)	21.5	17.9	15.8	13.5	11.9	10.7
(2)	17.2	13.8	11.9	9.9	8.6	7.7
(3)	30.5	26.6	23.9	21.1	18.6	17.1
(3) vs.(2)	+78%	+92%	+100%	+112%	+115%	+121%

Tab. 7.18: R-indicator (in %) for different values of N (the N most frequent triphones seen in the training data) and for the three situations. Words were extracted from the Mediargus corpus.

When comparing situation (3) with (2), the tendency becomes clear: the names contain more rarely seen triphones than the regular words. The reason why names comprise triphones that were rarely seen during training is twofold:

1. They follow another morpho-syntax. Morphemes are concatenated in a different way in regular words and in names. Take a name like ’Alblas’ for example. The triphone /l-b+l/ can only be seen in a cross-word context like ’Heb je a**l bl**aderen zien vallen?’. Cross-word triphones are less frequent than word internal triphones.

2. They contain rare phonemes. Many so-called ’Dutch’ names (e.g Torricellistraat) also partly originate from another language like French, German,. . . with foreign phonemes or with other phoneme frequencies (like /ʃ/ in Torricelli). So I couldn’t really call them Dutch.

The picture for the English names is quite the same. Apart from the occurrence of foreignizable phonemes, the significantly higher amount of rarely seen triphones measured over the erroneous words, is responsible for the improvement made by the NATIVE system. In fact the mechanism is the same as it is for Dutch words. I measured a 33% higher R-value ($N = 10000$) for triphones for the erroneous words when compared to all words and a 40% higher R for phonemes ($N = 35$). For the French words the figures were less distinctive. Here the relative difference only was 8% ($N = 10000$) for triphones but 38% for phonemes ($N = 35$). This means that I expect the English words to improve more than the French words under the NATIVE system. This is confirmed by the SIM-results in Table 7.13. For both English and French words, the number of foreignizable phonemes for the erroneous words is higher than for the regular words. The percentage of French names containing at least one foreignizable phoneme is 90.4%, whereas this is 83.0% for the English names. From this, I expect the French words to improve more under the FOREIGN system - when compared to the NATIVE system - than the English words. However Table 7.15 shows larger improvements for English names than for French.

I also repeated my analysis on ’normal’ speech corpora. The global R for triphones was consistently lower for TIMIT and WSJ as compared to the Autonomata task. This indicates that the acoustic coverage was larger for TIMIT and WSJ than for Autonomata. I found the R for triphones to be lower for the erroneous words than for the other words. In fact, for TIMIT the erroneous words were often found to have an R equal to zero. This clearly points to other error mechanisms besides the reliability of the acoustic models. That is the reason why my method was not able to improve on these corpora.

7.7.3.7 Using multilingually trained feature extractors

There is strong evidence [109; 93] that phonological feature models learned on multilingual data are more reliable than monolingually trained models. Therefore I performed an additional experiment in which the back-off model now uses multilingually trained networks instead of networks that were trained on native (i.e. Dutch) speech only. I composed a trilingual dataset comprising Dutch, English and French data.

For the Dutch data I used CoGeN [26], a predecessor of the CGN. The English data was TIMIT and the French was BDSons [11]. The composition of this trilingual dataset in terms of number of sentences or paragraphs and number of training patterns is shown in Table 7.19.

subset	#sentences	#paragraphs	#training patterns
CoGeN	–	817	2158944 (53%)
TIMIT	3696	–	954422 (23%)
BDSons	–	201	955612 (23%)

Tab. 7.19: Composition of the trilingual training set.

After having retrained the MLPs for the feature extractors, I reran the three experiments (NATIVE, FOREIGN-UNIQUE, FOREIGN-VARS). The results of these experiments are represented in Table 7.20.

For the SIMs it is clear that the gains are now a bit higher in the case of English and French names for the experiments with foreignizable phonemes, but that the result for the Dutch names now turns out to be worse. For all names together the gain is negligible. For the AMs I took the multilingually trained networks and adapted them in the same way as the monolingually trained networks before. The results were slightly better compared to the monolingual case. In cross-lingual situations (native Dutch speakers speaking English and French words), the multilingually trained feature extractors seem to be a bit more reliable after adaptation, leading to better results. However, the monolingually trained networks already reached a performance that was very close to the one obtained with the multilingually trained networks.

7.8 Conclusions

In this chapter I have made an attempt to build a speech recognizer based on PHFs. Two shortcomings of PHFs have been pointed out: the strong

models	experiment	measure	E	F	D	All
AMs	NATIVE	WER	**17.9**	**9.9**	**3.0**	**8.4**
		AI	**2.8**	**2.9**	**1.4**	**2.2**
		RI	**13.5**	**22.6**	**31.8**	**20.7**
	FOREIGN-UNIQUE	WER	**17.8**	**9.7**	**3.0**	**8.4**
		AI	**2.9**	**3.1**	**1.4**	**2.2**
		RI	**14.0**	**24.2**	**31.8**	**20.7**
	FOREIGN-VARS	WER	**17.7**	**9.5**	**3.1**	**8.3**
		AI	**3.0**	**3.3**	**1.3**	**2.3**
		RI	**14.4**	**25.8**	**29.5**	**21.7**
SIMs	NATIVE	WER	**47.3**	**30.8**	24.0	31.5
		AI	**3.5**	**1.7**	-2.7	0.0
		RI	**6.9**	**5.2**	-12.7	0.0
	FOREIGN-UNIQUE	WER	**45.1**	**29.1**	24.3	30.7
		AI	**5.7**	**3.4**	-3.0	0.8
		RI	**11.2**	**10.5**	-14.1	2.5
	FOREIGN-VARS	WER	**46.2**	**30.4**	24.1	31.2
		AI	**4.6**	**2.1**	-2.8	0.3
		RI	**9.0**	**6.5**	-13.1	0.9

Tab. 7.20: Performances (all in %) of an ASR with a two-stream acoustic model for all three experimental conditions mentioned before, but with a back-off model using multilingually trained networks. The acoustic models were AMs and SIMs.

correlations and the abundance in the feature vector due to irrelevant features. I proposed a solution to each problem consisting of a decorrelation technique and a relevancy handling technique. The decorrelation technique was also tested on classical MFCCs and was found to yield better results in this case too. I found that significant improvements could be obtained with the decorrelation technique: 20% (14%) for the ACFs and 30% (26%) for the PHFs on TIMIT (WSJ). As expected, the gain for the PHFs was substantially higher. An extra 7.7% relative improvement was obtained with the relevancy technique (tested on TIMIT). But despite of this large improvement of the PHF recognizer, it is still performing worse than the ACF-recognizer. I was less successful in obtaining improvements when trying to combine ACF and PHF information for CSR. The measured gains on TIMIT and WSJ were not significant, but nevertheless the proposed method seems to work for the recognition of foreign names spoken by native speakers. In this context the method is further extended by the introduction of foreignizable phonemes. Important is that the presented method does not require any foreign phoneme models, nor a speech corpus containing foreign phonemes from which to train foreign pronunciations.

For the recognition of English and French names spoken by Dutch speakers, the method yielded significant reductions of the WER of 15% and 22% relative compared to a baseline system that already made use of name transcriptions produced by G2Ps for Dutch, English and French. Surprisingly, the recognition also improved for the Dutch names. A more detailed analysis of the errors revealed that the atypical distributions of triphones and phonemes in spoken names is responsible for the latter improvements. All the gains are achieved with only a small additional cost, originating from the computation of phonological scores and the inclusion of extra variants in the lexicon.

8 Conclusions and Perspectives

Since my dissertation clearly consists of two separate parts, the conclusions and perspectives are also separated in two sections.

8.1 Spontaneous Speech Recognition

In the first part of this dissertation I have focused on the complex problem of disfluencies in spontaneous speech and I have treated the case of filled pauses (FPs) more in particular. I proposed a disfluency handling technique that was informed by an external disfluency detector. I found that it was possible to detect FP intervals from running speech with a good precision and recall. Though, the question whether other disfluencies like word repetitions (WRs) can also be detected reliably is still unanswered. But based on the outcome of preliminary experiments that are not included in this dissertation, I tend to believe that the answer will be negative. Some disfluencies like sentence restarts (SRs) do not have strong acoustic correlates at all, so their detection can only be based on a linguistic analysis of the speech recognizer's output.

With the aid of the detected FPs I was able to improve the recognition of spontaneous Dutch by 2% absolute. This improvement corresponds to 0.3 regular words per FP occurring in the speech. Given the fact that each FP is responsible for ca. 0.7 regular word errors, this means that I can solve about half of the errors that can be owed to the occurrence of FP disfluencies. A double gain can be expected if an ideal FP detector could be conceived.

Research in the field of spontaneous speech recognition is currently a 'hot' topic, and several workshops on the subject of disfluencies and spontaneous speech processing and recognition have been organized. But it also turns out to be a difficult topic for as far as the recognition is concerned. The main problem regarding disfluencies is their unpredictability. They behave like random events whose relation to the context is rather complicated. One could hypothesize that disfluencies tend to occur when the local perplexity reaches a high level, but this hypothesis too could be subject of further research.

My personal opinion is that it will take much more than a proper handling of disfluency to raise the accuracy of spontaneous speech recognition. In fact I argue that the recognition of spontaneous speech is problematic due to the occurrence of what I would call *superfluencies* and which I would define as strongly reduced pronunciations of chuncks of syllables.

Let me consider a Dutch example of a superfluency with its (manually) annotated phonetic transcription.

chunk:	...	dat	is	hetgéen	we	gedáan	hebben	...
superfluent phonetic trans.:	...	/t s t G e w @ x d a n h E b n/						...

Clearly, the sequence of words “dat is hetgeen" is reduced to /t s t G e/, the word “we" remained unchanged, “gedaan" becomes /x d a n/ and “hebben" is contracted to /h E b n/. Now several questions can be asked about the processes governing the creation of superfluencies.

It is observed that the stress pattern of the sentence chunk (stresses are marked with an accent) plays an important role in the generation of the superfluent parts. Unstressed words or syllables can be reduced to superfluent forms, whereas the stressed words or syllables are far more resistant to reduction. Since pitch seems to be one mechanism governing the syllable reduction, it may turn out that one and the same chunk is reduced differently according to the pitch pattern it has. Suppose now that the stress pattern is different, like in

chunk:	...	dát	is	hetgeen	we	gedaan	hébben	...
superfluent phonetic trans.:	...	/d A s t G e w @ G d a n h E b @ n/						...

The superfluent form of this sentence chunk is clearly different from before. The word sequence “dat is hetgeen" is now pronounced as /d A s t G e/ because the word “dat" is stressed, making it more resistant to reduction. The same holds for the word “hebben", which is now pronounced as /h E b @ n/, a canonical pronunciation. Another important observation is that superfluencies are cross-word phenomena. So this means that the superfluency phenomenon must be studied for word sequences. This means that a second important factor that governs the probability of a syllable being reduced, is the frequency of the word N-gram the syllable makes part of. N-grams with a high frequency have typically a higher probability of being reduced than N-grams with a low frequency. It would be interesting to use the N-gram counts and N-gram stress patterns to predict the deletion probability of a phoneme in a syllable.

However I anticipate that stress patterns and frequency of N-grams are not the only mechanisms that tell us when a word sequence is likely to be reduced. The possible confusion that would arise with other words or word sequences plays an important role too. A word sequence that after reduction closely resembles another word sequence will not be reduced.

Another important factor that guides the reduction of syllables is the set of phonological features attributed to the syllable's onset, nucleus or coda. A statistical analysis of the stability of phonetic segments was carried out in [49; 2]. In this study phonological features were found to play an important role in the deletion mechanism of syllable parts. As a summary, the four most important elements that guide the shortening/reduction of syllables (or word sequences) according to me, are:

1. The stress level (high, intermediate or no stress) put on the syllable.

2. The phonological features of the onset, nucleus or coda of the syllable.

3. The loss in distinctive information or the increase in entropy when one or more phonemes are deleted from the syllable.

4. The frequency of the N-gram to which the syllable belongs.

Assume that we can measure all of the four mentioned quantities, then the next question is how to use this information to estimate the probability that a phoneme can be deleted. The only reliable way to accomplish this is by means of a data-driven method that takes as input the four features classes. Each class can have multiple multivalued features. For instance, stress level may be described as high, intermediate or none and the stress levels of neighboring syllables may be supplied as extra features. The number of phonological features is of course of the order of 20 or more etc. All these features should be supplied to a statistical model (e.g. a MLP) that can reliably estimate the deletion probability of a phoneme or a phoneme sequence. Once this probability is available, it could be used in some way to alter the pronunciation of the active word. In my opinion this can be done by using skip arcs in the recognizer and by increasing the transition probability of the skip arcs on the basis of the estimated deletion probability of the phoneme.

Modeling superfluent pronunciations is a kind of pronunciation variation. The extra pronunciations can be derived from the so-called canonical pronunciation by deleting and/or substituting phonemes. I now argue that substitution and deletion are two different mechanisms for pronunciation variation. Substitutions can be a *real* variation that can be understood in terms of region or dialectal background (compare the pronunciation of 'normaal', /nOrmal/ in Flanders with /n@rmal/ in the Netherlands) or it can be seen as a degree of fluency (non-stressed vowels can be substituted by schwa, voiced plosives by their unvoiced counterparts, etc.). Deletions on the other hand are always meant to speed up the speaking rate. This second variation in pronunciation is independent of region or dialect, but inherent to pronunciation. When people are involved in a spontaneous conversation they tend to speak as sloppy as possible so as to be just understandable. As such this pronunciation variation mechanism should *not* be modeled on the lexical level of a speech recognizer, but in the automaton itself. Since deletions constitute the largest part of the superfluent pronunciation variations, they could be modeled by skip arcs. Such a system is thus externally informed.

However, accounting for superfluencies and disfluencies may yet not be good enough to model spontaneous speech. Spontaneous speech recognition requires a much more profound handling of all available knowledge sources than is currently the case with traditional HMM-based speech recognizers. We all know that in order to represent speech we need multiple tiers. Which tiers are contributing most to speech recognition is an open research question, but there is growing evidence that at least a linguistic tier, a syllable tier and an articulatory tier is needed. Surprisingly many of these tiers are not considered by a standard HMM-based speech recognizer, where only a word and a phoneme level tier are considered. It may be questioned whether words and phonemes still constitute stable segments in spontaneous speech. Recent investigations [49] favor the syllable as the more stable segment. It is not necessary to combine information from all tiers at all times. Time instances at which the evidence from all tiers is high are interesting landmarks and could be considered as *pivot* elements during the recognition process. At such points a lexical access can be made.

Presumably, speech recognition is not a strict left-to-right process. Psychological tests with reading behaviour show that readers return to pivot elements in the text more than once. Although reading and recognizing speech are different neurological processes, the high similarity between them is obviously there.

8.2 Phonological Features

In the second part of this dissertation the focus was shifted to the application of PHFs for speech processing: segmentation and labeling, speaker characterization, pronunciation scoring, and ASR.

Concerning the use of PHFs for speech recognition, I have been successful in the specialized domain of spoken name recognition where substantial gains were achieved with a system incorporating a phonologically inspired back-off model to supplement the traditional phoneme models. Not only foreign names spoken by native Dutch speakers, but also native names could be better recognized. A statistical analysis revealed that even for these names the acoustic models are not working in their normal operating point. By using a back-off model that is based on PHFs which are shared across different phonemes and even across different languages it is possible to obtain more reliable acoustic scores for the scarcely observed phonemes. However for common CSR tasks like TIMIT and WSJ, I was not able to find any substantial gain with the methods I conceived for the combination of acoustic and phonological features in one ASR. In these tasks the acoustic models are applied in their normal operating point and do not benefit from the extra information emerging from the phonological back-off model.

A promising new research line in this field would be the exploitation of asynchrony during the training of PHF models as well as during the search for the best hypothesis. The feature asynchrony is inherent to phonological features as already discussed in chapter 5. Interesting work in this area has already been done by Leung et al. and reported in [68].

A
CGN subsets

This Appendix describes the composition and construction of the CGN training and test corpus used for the experiments in chapters 3 and 4.

For the training files we used the entire file. The 16 test files, which were used for the recognition experiments, were not used entirely because of the following two main reasons:

1. Some chunks are overlapping. Especially files with multiple speakers and/or background chunks may contain such overlapping chunks.

2. Some chunks just contain noises or background sounds.

Therefore we selected only non-overlapping chunks with relevant information from the files. The CGN-files that were used for training and testing are shown in Table A.1. We did not use files from the components c,d (spontaneous telephone dialogues), e (simulated business negotiations), h (lessons recorded in the classroom) and o (read speech). The chunks that were used from the 16 test files are represented in Tables A.2 and A.3.

component	training files	test files
a	fv400073, fv400089, fv400083, fv400086, fv400195 fv400198, fv400216, fv400219, fv400269, fv400305 fv400306, fv400363, fv400364, fv400370, fv400436 fv400439, fv400441	
b	fv400109, fv400112, fv400165, fv400117, fv400145 fv400155, fv400169, fv400118	
f	fv600029, fv600049, fv600083, fv600134, fv600145 fv600217, fv600223, fv600227, fv600243, fv600272 fv600373, fv600430, fv600471, fv600604, fv600614 fv600622, fv600798, fv600840, fv600882, fv600990 fv601128	fv600159 fv600600 fv600839 fv601318 fv600997, fv600079
g	fv600005, fv600012, fv600014, fv600625 fv600677, fv600686, fv600713, fv600718	fv600594, fv600591 fv600595, fv600593
i	fv600388, fv600764, fv600739, fv600749, fv601112 fv600757, fv600759, fv600768, fv600770, fv600773 fv600787, fv600793, fv600795, fv601088, fv601115	fv600127 fv600361 fv600341
j	fv600091, fv600551	fv601194, fv601357
k	fv600053, fv600121, fv600268	
l	fv600253, fv600035, fv600281, fv600386, fv600458 fv600529, fv600855, fv600863, fv600870, fv600877 fv600998, fv600999, fv601074	fv600854
m	fv400007	
n	fv400011, fv400016, fv400019	
total	91	16

Tab. A.1: *Composition of the* training *and* test *corpus extracted from the Spoken Dutch Corpus (CGN).*

file	start time (sec)	end time (sec)	file	start time (sec)	end time (sec)
fv600127	0.00	11.98	fv600361	0.00	7.03
	11.98	19.13		7.03	17.13
	19.13	31.15		17.13	25.38
	31.15	43.31		25.38	36.14
	43.31	50.07		36.14	51.14
	50.07	58.45		51.14	65.69
	58.45	67.85		65.69	75.53
	67.85	75.25		75.53	83.50
	75.25	81.63		83.50	93.66
	81.63	96.11		93.66	106.88
	96.11	103.76	fv600594	0.00	14.00
	103.76	113.55		14.00	28.18
	113.55	121.09		28.18	39.05
	121.09	133.02		39.05	47.95
	133.02	146.78		47.95	54.47
fv600159	0.00	14.13		55.80	57.90
	14.13	22.68		60.69	71.12
	22.68	29.93		71.12	82.53
	29.93	37.31		82.53	91.22
	38.27	48.83		91.22	102.45
	48.83	54.91	fv600595	0.00	8.97
	54.91	67.57		9.59	25.53
	67.57	73.61		26.23	28.66
	73.61	80.04		29.33	32.39
	80.04	84.10		33.01	45.93
	84.10	100.91		46.57	56.91
	100.91	110.85		56.91	71.87
	110.85	123.82		71.87	81.57
	123.82	135.73		82.44	88.06
	135.73	142.98		88.44	89.57
	142.98	153.91		92.30	105.61
	153.91	165.23		106.23	121.72
fv601194	0.00	14.56		121.72	139.80
	14.56	21.70		139.80	153.88
	22.30	34.71		153.88	165.95
	34.71	42.73		165.95	170.87
	42.73	55.09		170.87	188.12
	55.09	69.29		188.62	199.84
	70.12	78.95		201.99	203.17
	79.40	90.49		203.93	220.50
	90.49	105.71			
	105.71	118.28			
	118.28	121.25			

Tab. A.2: *The selected chunks used in the CGN test corpus (1).*

file	start time (sec)	end time (sec)	file	start time (sec)	end time (sec)
fv600839	0.00	6.25	fv600591	0.00	21.74
	7.83	16.75		22.41	41.66
	7.83	29.71		41.66	52.84
	32.73	48.49		52.84	64.27
	48.89	50.85		64.27	69.73
	56.34	71.26		71.26	72.40
	71.69	82.64		80.16	93.62
	82.64	89.39		93.62	104.53
	89.39	101.86		104.53	117.31
	105.04	106.57		118.04	122.92
	110.26	123.94		123.59	128.08
	123.94	133.71		128.86	134.62
	135.19	138.40		135.72	155.70
	139.68	150.11		156.43	177.56
	150.11	159.70		178.13	190.10
	159.70	175.41		190.10	203.46
	175.41	186.72		203.46	212.44
	186.72	198.30		214.57	216.14
fv600997	0.00	3.43		217.19	228.51
	7.01	19.97		228.51	242.38
	19.97	30.61	fv601318	2.91	17.24
	30.61	43.64		18.02	21.06
	43.64	49.61		23.52	39.17
	53.99	63.03		39.17	48.18
	63.03	70.96		48.18	59.49
	70.96	81.95		59.49	67.13
	81.95	86.43		88.66	89.98
	87.28	96.51		90.07	101.73
	99.53	109.17		101.73	112.55
	109.17	119.85		112.55	131.34
	119.85	129.85		131.34	148.53
	129.85	139.07		152.16	158.31
	139.07	151.38		158.31	171.92
	156.65	159.99		178.83	190.39
	163.63	174.71		195.21	200.20
	174.71	178.94		204.44	208.93
	178.94	186.97	fv600600	0.00	5.67
	186.97	197.41		6.34	29.79
	197.41	212.57		49.54	60.35
	215.18	217.68		60.44	73.28
	218.55	223.68		73.36	89.04
	223.68	241.09		89.15	108.95
	241.09	246.58		109.18	110.58
	246.58	253.44			
	257.70	266.40			
	266.40	274.82			

Tab. A.3: *The selected chunks used in the CGN test corpus (2).*

file	start time (sec)	end time (sec)	file	start time (sec)	end time (sec)
fv600079	0.00	3.18	fv600341	0.00	13.79
	3.25	4.18		13.79	20.60
	4.31	5.32		20.60	31.32
	5.78	7.20		31.32	36.36
	7.36	9.49		36.36	47.90
	9.57	12.97		47.90	58.83
	25.26	38.54		58.83	66.55
	44.79	51.08		66.55	73.88
	71.59	73.49		73.88	82.92
	76.14	84.78		82.92	89.73
	84.92	86.68		90.40	99.13
	100.99	103.34		99.13	107.04
	103.52	106.22		107.04	118.14
	126.34	129.15		118.14	123.94
	148.84	149.59		123.94	139.02
fv600593	0.00	14.66		139.02	146.98
	14.66	21.91		146.98	154.96
	21.91	34.83		154.96	165.95
	38.12	49.47		165.95	167.96
	49.47	59.60		167.96	181.17
	59.60	69.09	fv601357	0.00	11.31
	69.09	83.17		11.94	29.38
	83.17	93.13		29.38	34.95
	93.13	102.46		34.95	43.78
	102.46	111.65		43.78	50.33
	111.65	121.59		50.33	59.37
	121.59	128.05		59.37	74.71
	128.05	136.62		74.71	83.69
	136.62	143.62		83.69	99.79
	144.37	156.23		99.79	105.59
	156.23	168.91		106.17	115.33
	168.91	184.03		115.33	122.99
	184.03	191.64		122.99	134.91
	191.64	205.23		134.99	142.79
	205.23	213.61		143.45	153.25
fv600854	0.00	16.47		153.25	158.37
	17.02	23.66		158.37	167.38
	42.57	53.39		167.38	170.74
	53.39	61.69		170.74	178.58
	61.69	73.77		179.27	190.49
	73.77	87.72		190.49	197.27
	89.14	99.80		197.27	201.92
	99.80	112.27		206.33	214.30
	115.07	126.33		214.30	218.72
	126.33	132.21			
	132.21	146.04			

Tab. A.4: *The selected chunks used in the CGN test corpus (3).*

B
Phonemic symbols

This Appendix summarizes all phonemic symbols that are used in my dissertation. I first give two Tables with the IPA (International Phonetic Alphabet) notation for the phonemes of the four most important European languages (English, French, German and Spanish) and Dutch. Table B.1 represents the vowels and Table B.2 the consonants. The degree of sharing by the five languages is represented in the first column.

shared by	Dutch	English	French	German	Spanish
5	i,u (2)				
4	a,e,o (3)	—			
4	ɛ,ɔ,ə(3)				—
3	ɪ (1)		—		—
3	y,ø(2)	—			—
3	ɑ (1)			—	—
2	ɛɪ(eɪ) (1)		—	—	—
2	—		—	aɪ,ʊ,aʊ(3)	—
2		—	œ(1)		—
1	—	æ,ɒ,ʌ,ɜː,ɔɪ,əʊ (6)	—		
1	œ,œy,ɑu (3)	—			
1	—		ɛ∼,œ∼,ɔ∼,ɑ∼ (4)	—	
1	—			ɤ,ɔɤ(2)	—
total	16	17	16	17	5

Tab. B.1: *Vowel sharing between five European languages.*

From the Tables the total number of phonemes in each of these languages can be derived. English and German have 41 phonemes, Dutch 38, French 35 and Spanish only 24.

I now give a Table with the ARPABET notation, the IPA notation and the SAMPA notation of the English, Dutch and French phonemes because these languages were used in this dissertation.

<table>
<tr><th>shared by</th><th>Dutch</th><th>English</th><th>French</th><th>German</th><th>Spanish</th></tr>
<tr><td>5</td><td colspan="5">p,b,t,d,k,g,f,s,ʃ,m,n,l,j (13)</td></tr>
<tr><td>4</td><td colspan="4">v,z (2)</td><td>—</td></tr>
<tr><td>4</td><td colspan="3">w (1)</td><td>—</td><td></td></tr>
<tr><td>3</td><td colspan="2">h, ŋ (2)</td><td>—</td><td></td><td>—</td></tr>
<tr><td>3</td><td></td><td colspan="2">—</td><td colspan="2">x (1)</td></tr>
<tr><td>3</td><td>—</td><td></td><td>—</td><td colspan="2">tʃ (1)</td></tr>
<tr><td>3</td><td colspan="4">ʒ (1)</td><td>—</td></tr>
<tr><td>2</td><td>—</td><td></td><td>—</td><td>dʒ (1)</td><td>—</td></tr>
<tr><td>2</td><td>—</td><td>θ (1)</td><td colspan="2">—</td><td></td></tr>
<tr><td>2</td><td colspan="2">—</td><td colspan="2">ʁ (1)</td><td>—</td></tr>
<tr><td>2</td><td>r (1)</td><td colspan="3">—</td><td></td></tr>
<tr><td>2</td><td colspan="2">—</td><td>ɲ(1)</td><td>—</td><td></td></tr>
<tr><td>1</td><td>—</td><td>ɹ,ð (2)</td><td colspan="3">—</td></tr>
<tr><td>1</td><td colspan="3">—</td><td>pf,ts (2)</td><td>—</td></tr>
<tr><td>1</td><td>ɣ(1)</td><td colspan="4">—</td></tr>
<tr><td>Tot.</td><td>22</td><td>24</td><td>19</td><td>24</td><td>19</td></tr>
</table>

Tab. B.2: *Consonant sharing between five European languages.*

English			Dutch		French	
ARPABET	IPA	SAMPA	IPA	SAMPA	IPA	SAMPA
			a	a	a	a (a∼)
aa	ɒ,ɑ	Q,A	ɑ	A	ɑ	A
ae	æ	{				
ah	ʌ	V				
ao	ɔː	O	ɔ	O	ɔ	O
			ɑu	Au		
aw	aʊ	aU				
ax	ə	@	ə	@	ə	@
ax-h	ə^{h}					
ax-r	ɚ					
ay	aɪ	aI				
b	b	b	b	b	b	b
bcl						
ch	ʧ	tS				
d	d	d	d	d	d	d
dcl						
dh	ð	D				
dx						
			e	e	e	e (e∼)
			ø	2	ø	2
eh	e	E	ɛ	E	ɛ	E
el	^{ə}l					
em	^{ə}m					
en	^{ə}n					
eng	^{ə}ŋ					
er	ɜː	3'				
			ɛɪ	Ei		
ey	eɪ	eI				
f	f	f	f	f	f	f
g	ɡ	g	ɡ	g	ɡ	g
			ɣ	G		
gcl						
hh	h	h	h	h		
hv	ɦ					
ih	ɪ	I	ɪ	I		

Tab. B.3: *All phonemic symbols used throughout my work.*

English			Dutch		French	
ARPABET	IPA	SAMPA	IPA	SAMPA	IPA	SAMPA
ix						
iy	i	i	i	i	i	i
jh	ʤ	dZ				
k	k	k	k	k	k	k
kcl						
l	l	l	l	l	l	l
m	m	m	m	m	m	m
n	n	n	n	n	n	n
ng	ŋ	N	ŋ	N		
nx						
			o	o	o	o (o∼)
ow	əʊ	@U				
oy	ɔɪ	OI				
p	p	p	p	p	p	p
pcl						
q	ʔ					
r	ɻ	r	r	r	ʁ	R
s	s	s	s	s	s	s
sh	ʃ	S	ʃ	S	ʃ	S
t	t	t	t	t	t	t
tcl						
th	θ	T				
			œy	9y		
uh	ʊ	U				
uw	u	u	u	u	u	u
ux						
v	v	v	v	v	v	v
w	w	w	w	w	w	w
			x	x		
			œ	Y		
					œ	9 (9∼)
y	j	j	j	j	j	j
			y	y	y	y
					ɲ	J
z	z	z	z	z	z	z
zh	ʒ	Z	ʒ	Z	ʒ	Z

Tab. B.4: *All phonemic symbols used throughout my work (continued).*

Bibliography

[1] G. Adda, M. Jardino, and J. Gauvain, "Language modeling for broadcast news transcription," in *Proc. European Conference on Speech Communication and Technology*, vol. IV, (Budapest, Hungary), pp. 1759–1762, Sept. 1999.

[2] M. Adda-Decker, P. B. de Mareüil, G. Adda, and L. Lamel, "Investigating syllabic structure and its variation in speech from french radio interviews," in *ISCA Tutorial and Research Workshop PMLA*, (Colorado, USA), pp. 89–94, 2002.

[3] M. Adda-Decker, B. Habert, C. Barras, G. Adda, P. Boula de Mareuil, and P. Paroubek, "A disfluency study for cleaning spontaneous speech automatic transcripts and improving speech language models," in *Proc. of ITRW on Disfluency in Spontaneous Speech*, (Göteborg, Sweden), pp. 67–70, Sept. 2003.

[4] O. Anderson, R. Kuhn, A. Lazarides, P. Dalsgaard, J. Haas, and E. Nöth, "Comparison of two tree-structured approaches for grapheme-to-phoneme conversion," in *Proc. International Conference on Spoken Language Processing*, (Kobe, Japan), pp. 1700–1703, 1996.

[5] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *The Journal of the Acoustical Society of America*, vol. 50, pp. 637–655, February 1971.

[6] B. Atal, J. Chang, M. Mathews, and J. Turkey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique," *The Journal of the Acoustical Society of America*, vol. 63, pp. 1535–1555, May 1978.

[7] *ATRANOS-project*. [online], 2000. URL: `http://www.esat.kuleuven.be/psi/spraak/projects/`.

[8] G. Aversano, A. Esposito, A. Esposito, and M. Marinaro, “A new text-independent method for phoneme segmentation,” in *Proc. IEEE Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 516–519, 2001.

[9] K. Bartkova and D. Jouvet, “On using units trained on foreign data for improved multiple accent recognition,” *Speech Communication*, vol. 49, pp. 836–846, 2007.

[10] A. Batliner, A. Kiessling, S. Burger, and E. Nöth, “Filled pauses in spontaneous speech,” in *Proc. International Congress of Phonetic Sciences*, (Stockholm, Sweden), Aug. 1995.

[11] *BDSons: Base de Donnée des Sons du Français*. [online], 2002. URL: `http://www.elda.org/catalogue/en/speech/S0005.html`.

[12] F. Beaufays, A. Sankar, S. Williams, and M. Weintraub, “Learning name pronunciations in automatic speech recognition systems,” in *International Conference on Tools with Artificial Intelligence*, (Washington, USA), pp. 233–240, 2003.

[13] P. Beyerlein, X. Aubert, R. Haeb-Umbach, M. Harris, D. Klakow, A. Wendemuth, S. Molau, M. Pitz, and A. Sixtus, “The Philips/RWTH system for transcription of broadcast news,” in *Proc. European Conference on Speech Communication and Technology*, vol. II, (Budapest, Hungary), pp. 647–650, Sept. 1999.

[14] H. Bourlard and N. Morgan, *Hybrid HMM/ANN systems for speech recognition: Overview and new research directions*. Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 1998.

[15] F. Brugnara, D. Falavigna, and M. Omologo, “Automatic segmentation and labelling of speech based on hidden markov models,” *Speech Communication*, vol. 12, pp. 357–370, April 1993.

[16] *Celex, The Dutch Centre for Lexical Information*. [online], 2001. URL: `http://www.ru.nl/celex`.

[17] S. Chang, S. Greenberg, and M. Wester, “An elitist approach to articulatory-acoustic feature classification,” in *Proc. European Conference on Speech Communication and Technology*, (Aalborg, Denmark), pp. 1725–1728, 2001.

[18] S. Chang, L. Shastri, and S. Greenberg, “Automatic phonetic transcription of spontaneous speech (american english),” in *Proc. International Conference on Spoken Language Processing*, (Beijing, China), pp. 330–333, 2000.

[19] N. Chomsky and M. Halle, *The Sound Pattern of English.* MIT Press, 1968.

[20] *The CMU pronunciation dictionary.* [online], 1995. URL: `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

[21] P. Dalsgaard, “Phoneme label alignment using acoustic-phonetic features and gaussian probability density functions,” *Computer, Speech and Language*, vol. 6, pp. 303–329, 1992.

[22] P. Dalsgaard, O. Andersen, W. Barry, and R. Jørgensen, “On the use of acoustic-phonetic features in interactive labeling of multi-lingual speech corpora,” in *Proc. International Conference on Acoustics, Speech and Signal Processing*, pp. 549–552, 1992.

[23] W. Daniels, *Applied nonparametric statistics.* 1978.

[24] S. B. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[25] K. Demuynck, J. Duchateau, D. V. Compernolle, and P. Wambacq, “An efficient search space representation for large vocabulary continuous speech recognition,” *Speech Communication*, vol. 30, pp. 37–53, January 2000.

[26] K. Demuynck, D. Van-Compernolle, C. Van-Hove, and J.-P. Martens, “CoGeN - Een corpus gesproken Nederlands voor spraaktechnologisch onderzoek,” final report, Katholieke Universiteit Leuven and Universiteit Gent, October 1997.

[27] L. Deng and D. Sun, “A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features,” *The Journal of the Acoustical Society of America*, vol. 95, pp. 2702–2719, May 1994.

[28] E. D’Halleweyn, J. Odijk, L. Teunissen, and C. Cucchiarini, “The dutch-flemish hlt programme stevin: Essential speech and language technology resources,” in *international conference on language resources and evaluation*, pp. 761–766, 2006.

[29] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. International Conference on Machine Learning*, (Banff, Alberta, Canada), pp. 225–232, 2004.

[30] J. Duchateau, K. Demuynck, D. V. Compernolle, and P. Wambacq, "Class definition in discriminant feature analysis," in *Proc. European Conference on Speech Communication and Technology*, (Aalborg, Denmark), pp. 1621–1624, 2001.

[31] J. Duchateau, K. Demuynck, and D. Van Compernolle, "Fast and accurate acoustic modelling with semi-continuous HMMs," *Speech Communication*, vol. 24, pp. 5–17, Apr. 1998.

[32] J. Duchateau, T. Laureys, K. Demuynck, and P. Wambacq, "Handling disfluencies in spontaneous language models," in *Computational Linguistics in the Netherlands* (T. Gaustad, ed.), Language and Computers. Studies in Practical Linguistics, (Amsterdam (The Netherlands) and New York (U.S.A)), pp. 39–50, Rodopi, 2003.

[33] E. Eide, "Distinctive features for use in an automatic speech recognition system," in *Proc. European Conference on Speech Communication and Technology*, (Scandinavia, Aalborg, Denmark), pp. 1613–1616, 2001.

[34] E. Eide, J. Rohlicek, H. Gish, and S. Mitter, "A linguistic feature representation of the speech waveform," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Minneapolis, USA), pp. 483–486, 1993.

[35] K. Elenius, "Phoneme recognition with an neural network," in *Proc. European Conference on Speech Communication and Technology*, (Genove, Italy), pp. 121–124, 1991.

[36] *UCLA Phonetics Lab - Electromagnetic articulography (EMA)*. [online], 2000. URL: `http://www.linguistics.ucla.edu /faciliti/facilities/physiology/ema.html`.

[37] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, (Santa Barbara, USA), pp. 347–354, 1997.

[38] W. Fisher, G. Doddington, and K. Goudie-Marshall, "The darpa speech recognition research database: Specifications and status," in *DARPA Workshop on Speech Recognition*, pp. 93–99, 1986.

[39] S. Fitt, "The pronunciation of unfamiliar native and non-native town names," in *Proc. European Conference on Speech Communication and Technology*, (Madrid, Spain), pp. 2227–2230, 1995.

[40] J. Frankel, M. Wester, and S. King, "Articulatory feature recognition using dynamic bayesian networks," in *Proc. International Conference on Spoken Language Processing*, (Jeju Island, South-Korea), 2004.

[41] M. Gabrea and D. O'Shaugnessy, "Detection of filled pauses in spontaneous conversational speech," in *Proc. International Conference on Spoken Language Processing*, vol. III, (Beijing, China), pp. 678–681, Sept. 2000.

[42] W. Gale and G. Sampson, "Good-turing frequency estimation without tears," *Journal of Quantitative Linguistics*, vol. 2, pp. 217–237, 1995.

[43] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 272–281, May 1999.

[44] J. Gauvain, L. Lamel, G. Adda, and M. Jardino, "Recent advances in transcribing television and radio broadcasts," in *Proc. European Conference on Speech Communication and Technology*, vol. II, (Budapest, Hungary), pp. 655–658, Sept. 1999.

[45] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, vol. I, (San Francisco, U.S.A.), pp. 517–520, Mar. 1992.

[46] W. Goedertier, S. Goddijn, and J.-P. Martens, "Orthographic transcription of the spoken Dutch corpus," in *international conference on language resources and evaluation*, (Athens, Greece), pp. 909–914, May 2000.

[47] S. Goronzy, S. Rapp, and R. Kompe, "Generating non-native pronunciation variants for lexicon adaptation," *Speech Communication*, vol. 42, pp. 109–123, 2004.

[48] M. Goto, K. Itou, and S. Hayamizu, "A real-time filled pause detection system for spontaneous speech," in *Proc. European Conference on Speech Communication and Technology*, vol. I, (Budapest, Hungary), pp. 227–230, Sept. 1999.

[49] S. Greenberg, H. Carvey, L. Hitchcock, and S. Chang, "The phonetic patterning of spontaneous american english discourse," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, (Tokyo, Japan), 2003.

[50] K. Hacioglu, B. Pellom, and W. Wayne, "Parsing speech into articulatory events," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Montreal, Canada), pp. 925–928, 2004.

[51] J. Harris, *English Sound Structure*. Blackwell, 1994.

[52] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[53] S. Hiroya and M. Honda, "Acoustic-to-articulatory inverse mapping using an HMM-based speech production model," in *Proc. International Conference on Spoken Language Processing*, (Denver, U.S.A), pp. 2305–2308, 2002.

[54] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov models for speech recognition*. Edinburgh University Press, 1990.

[55] M. Hung, M. Hu, M. Shanker, and B. Patuwo, "Estimating posterior probabilities in classification problems with neural networks," *International Journal of Computational Intelligence and Organizations*, vol. 1, no. 1, pp. 49–60, 1996.

[56] *The International Phonetic Association*. [online], 2006. URL: `http://www.arts.gla.ac.uk/IPA/ipa.html`.

[57] L. Josifovski, M. Cooke, P. Green, and A. Vizinho, "State based imputation of missing data for robust speech recognition and speech enhancement," in *Proc. European Conference on Speech Communication and Technology*, (Budapest, Hungary), pp. 2833–2836, 1999.

[58] B. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of markov chains," *IEEE Transactions on Information Theory*, vol. 32, pp. 307–309, March 1986.

[59] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, pp. 400–401, 1987.

[60] Y. Kim, H. Franco, and L. Neumeyer, "Automatic pronunciation scoring of specific phone segments for language instruction," in *Proc. European Conference on Speech Communication and Technology*, (Rhodes,Greece), pp. 645–648, 1997.

[61] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer, Speech and Language*, vol. 14, pp. 333–353, April 2000.

[62] K. Kirchhoff, *Robust Speech Recognition Using Articulatory Information.* PhD thesis, Universität Bielefeld, Germany, 1999.

[63] S. Kunzmann, V. Fisher, J. Gonzalez, C. Emam, C. Günther, and E. Janke, "Multilingual acoustic models for speech recognition and synthesis," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Montreal, Canada), pp. 745–748, 2004.

[64] K.-T. Lee, L. Melnar, J. Talley, and C. Wellekens, "Symbolic speaker adaptation with phone inventory expansion," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Tokyo, Japan), pp. 296–299, 2003.

[65] K.-T. Lee and C. Wellekens, "Dynamic lexicon using phonetic features," in *Proc. European Conference on Speech Communication and Technology*, (Aalborg, Denmark), pp. 1413–1416, 2001.

[66] K. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1641–1648, November 1989.

[67] H. Leung and V. Zue, "Phonetic classification using multi-layer perceptrons," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, vol. 1, (Albuquerque, NM, USA), pp. 525–528, April 1990.

[68] K. A. Leung and M. Siu, "Integration of acoustic and articulatory information with application to speech recognition," *Information Fusion*, vol. 5, pp. 141–151, June 2004.

[69] R. Lippman, "Review of neural network for speech recognition," *Neural Computation*, no. 1, pp. 1–38, 1989.

[70] K. Livescu, J. Glass, and J. Bilmes, "Hidden feature models for speech recognition using dynamic bayesian networks," in *Proc. European Conference on Speech Communication and Technology*, (Geneva, Switzerland), pp. 2529–2532, 2003.

[71] K. Ma, G. Zavaliagkos, and M. Meteer, "Bi-modal sentence structure for language modeling," *Speech Communication*, vol. 31, pp. 51–67, May 2000.

[72] B. Maison, S. Chen, and P. Cohen, "Pronunciation modeling for names of foreign origin," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, (Virgin Islands, USA), pp. 429–434, 2003.

[73] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, pp. 561–580, April 1975.

[74] J.-P. Martens, *Spraakverwerking.* Ghent University, Belgium, 2005.

[75] J.-P. Martens, D. Binnenpoorte, K. Demuynck, R. Van Parys, T. Laureys, W. Goedertier, and J. Duchateau, "Word segmentation in the spoken Dutch corpus," in *international conference on language resources and evaluation*, vol. V, (Las Palmas, Canary Islands, Spain), pp. 1432–1437, May 2002.

[76] J.-P. Martens and L. Depuydt, "Broad phonetic classification and segmentation of continuous speech by means of neural network and dynamic programming," *Speech Communication*, vol. 10, pp. 81–90, 1991.

[77] G. McLachlan and T. Krishnan, *The EM algorithm and extensions.* Wiley series in probability and statistics. John Wiley and Sons, 1997.

[78] *Resources for N-Best 2008.* [online], 2008. URL: `http://speech.tm.tno.nl/n-best/eval/data`.

[79] F. Metze, *Articulatory features for conversational speech recognition.* PhD thesis, Universität Fridericiana (TH), Karlsruhe, Germany, 2005.

[80] F. Metze and A. Waibel, "A flexible stream architecture for asr using articulatory features," in *Proc. International Conference on Spoken Language Processing*, (Denver, Colorado), pp. 2133–2136, 2002.

[81] G. V. Nuffelen, C. Middag, J.-P. Martens, and M. D. Bodt, "Speech technology based assessment of dysarthric speech: preliminary results," in *Proc. of 27th World Congress of the International Association of Logopedics and Phoniatrics (IALP)*, (Copenhagen, Denmark), 2007.

[82] D. O'Shaugnessy, "Locating disfluencies in spontaneous speech: An acoustical analysis," in *Proc. European Conference on Speech Communication and Technology*, vol. III, (Berlin, Germany), pp. 2187–2190, Sept. 1993.

[83] S. V. Pakhomov, "Modeling filled pauses in medical dictations," in *Proc. Association for Computational Linguistics (ACL)*, (College Park, Maryland, USA), pp. 619–624, June 1999.

[84] J. Papcun, T. Hochberg, F. Thomas, J. Larouche, J. Zacks, and S. Levy, "Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data," *The Journal of the Acoustical Society of America*, vol. 92, pp. 688–700, 1992.

[85] D. Paul and J. Baker, "The design for the wall street journal-based CSR corpus," in *Workshop on Speech and Natural Language*, (New York, USA), pp. 357–362, 1992.

[86] J. Peters, "Lm studies on filled pauses in spontaneous medical dictation," in *Proc. Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, (Edmonton, Canada), pp. 82–84, May 2003.

[87] F. C. Quimbo, T. Kawahara, and S. Doshita, "Prosodic analysis of fillers and self-repair in Japanese speech," in *Proc. International Conference on Spoken Language Processing*, (Sydney, Australia), pp. 3313–3316, Dec. 1998.

[88] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition.* Prentice Hall, 1993.

[89] M. Rahim, W. Kleijn, J. Schroeter, and C. Goodyear, "Acoustic to articulatory parameter mapping using an assembly of neural networks," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Toronto, Canada), pp. 485–488, 1991.

[90] D. Rummelhart, G. Hinton, and R. Williams, *Parallel Distributed Processing: Exploration of the Micro-Structure of Cognition.* Cambridge MA, MIT Press, 1986.

[91] H. Schramm, X. L. Aubert, C. Meyer, and J. Peters, "Filled-pause modeling for medical transcriptions," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, (Tokyo, Japan), Apr. 2003.

[92] J. Schroeter and M. Sondhi, "Techniques for estimating vocal-tract shapes from the speech signal," *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 133–149, February 1994.

[93] T. Schultz and K. Kirchhoff, *Multilingual speech processing.* Academic Press, 2006.

[94] E. Shriberg, "Disfluencies in Switchboard," in *Proc. International Conference on Spoken Language Processing*, vol. Addendum, (Philadelphia, U.S.A.), pp. 11–14, Oct. 1996.

[95] E. Shriberg and A. Stolcke, "Word predictability after hesitations: a corpus-based study," in *Proc. International Conference on Spoken Language Processing*, vol. III, (Philadelphia, U.S.A.), pp. 1868–1871, Oct. 1996.

[96] M. Siu and M. Ostendorf, "Modeling disfluencies in conversational speech," in *Proc. International Conference on Spoken Language Processing*, vol. I, (Atlanta, U.S.A.), pp. 386–389, Oct. 1996.

[97] G. Stemmer, E. Nöth, and H. Niemann, "Acoustic modeling of foreign words in a german speech recognition system," in *Proc. European Conference on Speech Communication and Technology*, (Aalborg, Denmark), pp. 2745–2748, 2001.

[98] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, vol. I, (Atlanta, U.S.A.), pp. 405–408, May 1996.

[99] F. Stouten, J. Duchateau, J.-P. Martens, and P. Wambacq, "Coping with disfluencies in spontaneous speech recognition: acoustic detection and linguistic context manipulation," *Speech Communication*, vol. 48, pp. 1590–1606, November 2006.

[100] F. Stouten and J.-P. Martens, "A feature-based filled pause detection system for dutch," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, (Virgin Islands, USA), pp. 309–314, 2003.

[101] F. Stouten and J.-P. Martens, "Benefits of disfluency detection in spontaneous speech recognition," in *Cost 278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, (Norwich, UK), 2004.

[102] F. Stouten and J.-P. Martens, "Coping with disfluencies in spontaneous speech recognition," in *Proc. International Conference on Spoken Language Processing*, (Jeju Island, Korea), pp. 1513–1516, 2004.

[103] F. Stouten and J.-P. Martens, "On the use of phonological features for pronunciation scoring," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Toulouse, France), pp. 329–333, 2006.

[104] F. Stouten and J.-P. Martens, "Speech recognition with phonological features: Some issues to attend," in *Proc. International Conference on Spoken Language Processing*, (Pittsburgh, USA), pp. 357–360, 2006.

[105] F. Stouten and J.-P. Martens, "Dealing with cross-lingual aspects in spoken name recognition," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, (Kyoto, Japan), pp. 2133–2136, 2007.

[106] F. Stouten and J.-P. Martens, "Recognition of foreign names spoken by native speakers," in *Proc. of Interspeech*, (Antwerp, Belgium), pp. 2133–2136, 2007.

[107] V. Stouten, H. Van hamme, J. Duchateau, and P. Wambacq, "Evaluation of model-based feature enhancement on the AURORA-4 task," in *Proc. European Conference on Speech Communication and Technology*, (Geneva, Switzerland), pp. 349–352, Sept. 2003.

[108] S. Stüker, F. Metze, T. Schultz, and A. Waibel, "Integrating multilingual articulatory features into speech recognition," in *Proc. European Conference on Speech Communication and Technology*, (Geneva, Switzerland), pp. 1033–1036, 2003.

[109] S. Stüker, T. Schultz, F. Metze, and A. Waibel, "Multilingual articulatory features," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, (Hong Kong, China), pp. 144–147, 2003.

[110] M. Tang, S. Seneff, and V. Zue, "Modeling linguistic features in speech recognition," in *Proc. European Conference on Speech Communication and Technology*, (Geneva, Switzerland), pp. 2585–2588, 2003.

[111] M. Tang, S. Seneff, and V. Zue, "Two-stage continuous speech recognition using feature-based models: A preliminary study,"

in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, (Virgen Islands, USA), pp. 49–54, 2003.

[112] J. Tebelskis, *Speech Recognition using Neural Network*. PhD thesis, Carnegie Mellon University, 1995.

[113] L. ten Bosch, "Speech variation and the use of distance metrics on the articulatory feature space," in *Workshop on Speech Recognition and Intrinsic Variation (SRIV)*, (Toulouse, France), pp. 27–32, 2006.

[114] K. Truong, A. Neri, C. Cuchiarini, and H. Strik, "Automatic pronunciation error detection: an acoustic-phonetic approach," in *Proc. of the InSTIL/ICALL Symposium*, pp. 135–138, 2004.

[115] S. Tsakalidis, V. Doumpiotis, and W. Byrne, "Discriminative linear transforms for feature normalization and speaker adaptation in HMM estimation," in *Proc. International Conference on Spoken Language Processing*, (Denver, USA), pp. 2585–2588, 2002.

[116] H. van den Heuvel, J.-P. Martens, B. D'hoore, K. D'hanens, and N. Konings, "The autonomata spoken name corpus. design, recording, transcription and distribution of the corpus," in *international conference on language resources and evaluation*, (Marrakech, Morroco), p. to appear, 2008.

[117] H. van den Heuvel, J.-P. Martens, and N. Konings, "G2P-conversion of names. what can we do (better)?," in *Proc. of Interspeech*, (Antwerp, Belgium), pp. 1773–1776, 2007.

[118] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, 1967.

[119] A. Vorstermans, J. Martens, and B. Van Coile, "Automatic segmentation and labelling of multi-lingual speech data," *Speech Communication*, vol. 19, pp. 271–293, April 1996.

[120] M. Wester, "Asynchronous articulatory feature recognition using dynamic bayesian networks," in *Technical Report of the Institute of Electronics, Information and Communication Engineers*, (Kyoto, Japan), 2004.

[121] M. Wester, S. Greenberg, and S. Chang, "A dutch treatment of an elitist approach to articulatory-acoustic feature classification," in *Proc. European Conference on Speech Communication and Technology*, (Aalborg, Denmark), pp. 1729–1732, 2001.

[122] G. Williams, M. Terry, and J. Kaye, "Phonological elements as a basis for language-independent asr," in *Proc. International Conference on Spoken Language Processing*, (Sydney, Australia), pp. 88–91, 1998.

[123] A. Wrench and W. Hardcastle, "A multichannel articulatory speech database and its applications for automatic speech recognition," in *5th Seminar on Speech Production*, (Kloster Seeon, Bavaria), pp. 305–308, 2000.

[124] *University of Wisconsin X-Ray Microbeam Facility.* [online], 1995. URL: `http://www.medsch.wisc.edu/ubeam`.

[125] Q. Yang, *Data-Driven Approaches to Pronunciation Variation Modeling for Automatic Speech Recognition.* PhD thesis, Ghent University, Belgium, 2005.

[126] Q. Yang, J.-P. Martens, P.-J. Ghesquiere, and D. Van Compernolle, "Pronunciation variation modeling for asr: Large improvements are possible but small ones are likely," in *ISCA Tutorial and Research Workshop PMLA*, (Estes Park, Colorado, USA), pp. 123–128, 2002.

[127] S. Young, D. Kershaw, J. Odell, D. Ollasson, V. Valtchev, and P. Woodland, *The HTK-book version 3.0.* Cambridge University, Engineering Department, 2000.

[128] H. Yu, T. Tomokiyo, Z. Wang, and A. Waibel, "New developments in automatic meeting transcription," in *Proc. International Conference on Spoken Language Processing*, vol. IV, (Beijing, China), pp. 310–313, Sept. 2000.

[129] K. Zechner and A. Waibel, "Using chunk based partial parsing of spontaneous speech in unrestricted domains for reducing word error rate in speech recognition," in *Proc. 17th Conference on Computational Linguistics (COLING/ACL'98)*, (Montreal, Canada), pp. 1453–1459, Aug. 1998.