Robust Speech Recognition Using Articulatory Information

Der Technischen Fakultät der
Universität Bielefeld

zur Erlangung des Grades eines

Doktor–Ingenieur

vorgelegt von

Katrin Kirchhoff

Bielefeld — Juni 1999

Gedruckt auf alterungsbeständigem Papier ∞ ISO 9706

Abstract

Current automatic speech recognition systems make use of a single source of information about their input, viz. a preprocessed form of the acoustic speech signal, which encodes the time-frequency distribution of signal energy. The goal of this thesis is to investigate the benefits of integrating articulatory information into state-of-the art speech recognizers, either as a genuine alternative to standard acoustic representations, or as an additional source of information. Articulatory information is represented in terms of abstract articulatory classes or "features", which are extracted from the speech signal by means of statistical classifiers. A higher-level classifier then combines the scores for these features and maps them to standard subword unit probabilities.

The main motivation for this approach is to improve the robustness of speech recognition systems in adverse acoustic environments, such as background noise. Typically, recognition systems show a sharp decline of performance under these conditions. We argue and demonstrate empirically that the articulatory feature approach can lead to greater robustness by enhancing the accuracy of the bottom-up acoustic modeling component in a speech recognition system.

The second focus point of this thesis is to provide detailed analyses of the different types of information provided by the acoustic and the articulatory representations, respectively, and to develop strategies to optimally combine them. To this effect we investigate combination methods at the levels of feature extraction, subword unit probability estimation, and word recognition.

The feasibility of this approach is demonstrated with respect to two different speech recognition tasks. The first of these is an American English corpus of telephone-bandwidth speech; the recognition domain is continuous numbers. The second is a German database of studio-quality speech consisting of spontaneous dialogues. In both cases recognition performance will be tested not only under clean acoustic conditions but also under deteriorated conditions.

Diese Arbeit ist meinen Eltern, Heinz und Annegret Kirchhoff, gewidmet.

Acknowledgments

First of all, my thanks go to my advisor, Gerhard Sagerer, not only for his continuous support and encouragement throughout my PhD years, but also for giving me the freedom to spend part of this time abroad. I am furthermore grateful to the members of the Applied Computer Science Group at Bielefeld for their cooperation and for generally making this group a fun place to work at. I would like to single out Christoph Schillo, for being a great office mate, friend and invaluable knowledge source on graphics tools ;-) and Gernot Fink, whose competent advice on problems of speech recognition and on the intricacies of the ESMERALDA system was much appreciated.

Part of this work was completed while I was a research student in the Realization Group at the International Computer Science Institute, Berkeley, USA. I am grateful to Nelson Morgan and Steve Greenberg for making this visit possible. I benefited greatly from discussions with members of the Realization Group and gratefully acknowledge the use of their resources and software. Special thanks are due to Steve Greenberg for agreeing to be my external examiner and putting up with the “remote scheduling” problems this involved.

Last but not least I would like to thank my parents for their assistance throughout my education, and, above all, Jeff, for his affection, support, and inspiration.

Contents

List of Figures

Chapter 1

Introduction

The goal of this thesis is to increase the robustness of automatic speech recognition (ASR) systems by integrating information about articulation. Speech recognition, as opposed to the higher-level task of speech understanding, is concerned with identifying the word sequence of an utterance from the corresponding acoustic speech signal. Standard speech recognizers, of the structure depicted in Figure 1.1, employ a preprocessed form of the acoustic signal, which provides information about the distribution of signal energy across time and frequency. In most systems, this representation is used as the only source of information about the word sequence. However, different signal representations may be employed, either as genuine alternatives to an acoustic representation, or as additional sources of information. In this study we will demonstrate the viability and the potential of a speech signal representation which is based on *articulatory* categories, also termed *articulatory features*. These features describe properties of speech production rather than the properties of the acoustic signal resulting from it.

The articulatory features we are concerned with in this thesis are *not* detailed numerical descriptions of the movements of articulators during speech production. Rather, they are abstract classes which characterize the most essential aspects of articulation in a highly quantized, canonical form, leading to a representational level intermediate between the signal and the level of lexical units. We argue that this approach (schematically depicted in Figure 1.2) offers several advan-

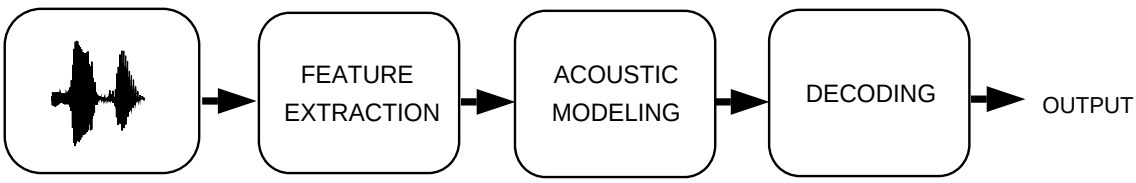

Figure 1.1: Basic structure of standard automatic speech recognition systems. Acoustic features are extracted from the incoming speech signal and passed to the acoustic modeling component, which estimates subword unit probabilities. These are subsequently used in the lexical decoding process.

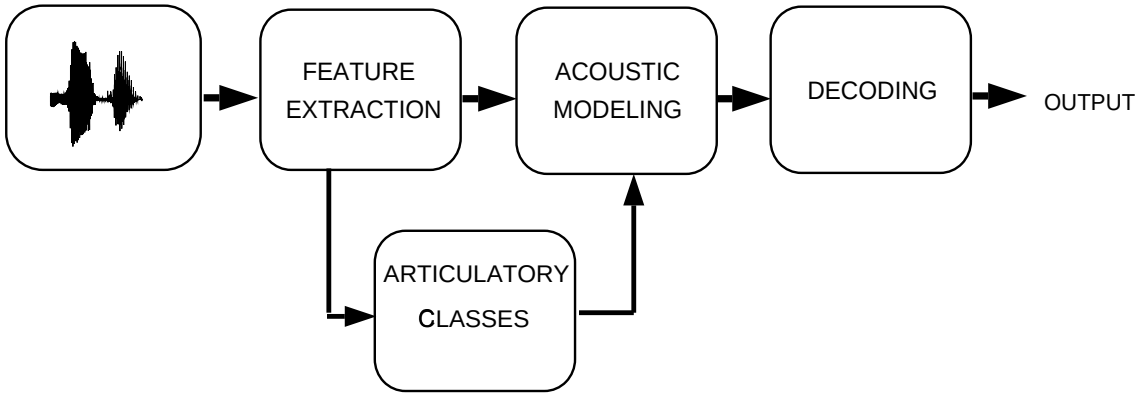

Figure 1.2: Recognition system extended by articulatory representation.

tages over the conventional, acoustics-only approach described above. More specifically, the use of an articulatory representation, either by itself or in combination with an acoustic representation, may lead to increased recognition robustness in adverse acoustic environments such as speech contaminated by background noise. In the course of this thesis we will investigate the feasibility of the articulatory feature (AF) approach by applying it to different languages (German and American English), different recognition tasks (small-vocabulary numbers recognition and large-vocabulary conversational speech recognition) and different recognition paradigms (hybrid HMM/ANN and Gaussian mixture HMM systems). In all cases we will analyze the characteristic differences between the acoustic and articulatory representations and we will develop strategies to optimally combine them.

1.1 Problems of Automatic Speech Recognition

ASR research efforts have been steadily intensified over the past thirty years, particularly in the last decade. As a result of the development of both efficient speech recognition algorithms and powerful hardware, the quality of ASR systems has increased drastically. A number of ASR applications are now commercially available, such as dictation systems, voice dialing, voice-controlled personal computer interfaces, or information systems with speech-based front-ends.

Nevertheless, speech recognizers are still far from being ubiquitous. Most of the applications listed above involve very limited recognition tasks, such as identifying the digits from 0 to 9 or a small number of isolated word commands. Others, such as dictation systems, can handle continuous speech and a large vocabulary, but they require good acoustic conditions (quiet environment and a microphone which meets the system requirements) and an extensive enrollment phase to adapt the system to every new user. The reason why speech recognition has not found more widespread use and why its commercial potential has not yet been fully exploited is the

Condition	Word Error
Baseline, speaker-independent	3.0%
Baseline, speaker-dependent	1.5%
Channel variation	12.0%
Transducer	10.0%
Speaking rate	15.0%
No language model	70%
Noise	30.0%
Dialectal speakers	20.0%
Non-native speakers	45%
Noise + non-native speakers	85%
Combined	98.0%

Table 1.1: Effects of adverse conditions on speech recognizer performance, according to [48].

lack of robustness of current ASR technology. Speech recognizers typically deteriorate sharply in adverse acoustic conditions, e.g. in the presence of noise or channel variability. Further difficulties are presented by low-quality, band-limited (telephone) speech and everyday conversational speech. Two recent studies elucidate the problems which need to be overcome before more sophisticated ASR applications can be developed.

The first study [48] describes a state-of-the-art recognizer for the Resource Management task [96] and the various effects which different recognition conditions have on its performance. These are listed in Table 1.1. The word error rate[1] of 3% obtained on clean, undisturbed speech increased to the word error rates shown in the right-hand column in Table 1.1 when the system was presented with channel variability, transducer differences, fast speaking rates, etc. Combining all these conditions led to a word error rate of 98%, demonstrating the detrimental effect of a mismatch between training and testing conditions. It is well known that human listeners are not affected by these conditions to the same degree. Background noise, room reverberation, and band-limited speech, let alone conversational speech, do not constitute problems for human speech perception. This difference in performance has been quantified in greater detail in the second study [85].

The author compares human and machine recognition performance on six different speech corpora, ranging from very limited tasks like digit recognition to conversational speech recognition using an unlimited vocabulary. These corpora are listed in Table 1.2, together with the best error rates obtained on these tasks both in human speech perception experiments and in ASR experi-

[1] Word error rate, the standard measure of speech recognizer performance, is defined as $100 - \frac{\text{Insertions + Deletions + Substitutions}}{\text{\# words in the test set}}$.

Corpus	Description	Vocabulary size	Human performance	Machine performance
Alphabet letters	Read alphabet letters	26	1.6%	5%
Resource Management	Read sentences	1000	0.1%	3.6%
North American Business News	Read sentences	5000-Unlimited	0.9%	7.2%
Switchboard	Spontaneous telephone conversations	2000-Unlimited	4%	40%
Switchboard wordspotting	Spontaneous telephone conversations	20 keywords	12.8%	31.3%

Table 1.2: Human vs. machine performance (measured in word error rate) on six different speech corpora, according to [85].

ments.

In addition to clean recognition conditions, mismatched conditions were tested. The latter included artificially reverberated speech, additive noise, vocoded speech, and channel variability caused by the use of different microphones for training and testing. Furthermore, some perception and recognition experiments were performed in the absence of contextual information: in the case of human listeners this was achieved by embedding the words to be recognized in non-informative carrier sentences which were then read aloud to the subjects; in the ASR experiments the recognizer was run without a language model. Table 1.3 lists some of the results for these conditions.

In general, the word error rates obtained by human listeners are more than an order of magnitude lower than those obtained by automatic speech recognizers. Furthermore, human error rates are low in “clean” listening conditions and do not deteriorate much in adverse conditions. Human performance is particularly superior at high noise levels and when little or no contextual information is available. It should be noted that the speech recognizers used in the mismatch experiments did in most cases include noise compensation or noise adaptation components. Nevertheless, their results do not even come close to human performance. As a conclusion to this study, the author suggests the following directions for future speech recognition research:

- **low-level acoustic-phonetic modeling**: the superiority of human perception in the ab-

Corpus	Recognition condition	Human performance	Machine performance
Resource Management	Nonsense sentences	2.0%	–
Resource Management	Null grammar	–	17%
North American Business News	Added car noise, 0 - 22 dB	1.1 - 0.9%	12.8 - 8.4%
North American Business New	Different microphones	0.4 - 0.8%	6.6 - 23.9%

Table 1.3: Human and machine recognition rates under mismatched conditions, according to [85].

sence of contextual information suggests that humans perform a very detailed low-level acoustic match. Therefore, emphasis should be placed on improving acoustic modeling in speech recognizers.

- **noise robustness**: better noise and channel adaptation algorithms should be developed to increase the performance in mismatched conditions.
- **modeling of spontaneous speech**: the high error rates obtained on conversational speech demonstrate the need for better modeling of spontaneous speech phenomena, such as variability of speaking rate or a high degree of coarticulation.
- **more sophisticated language modeling**: an unlimited vocabulary requires the adaptation of language models to speaker-style variability and rapid topic-switching.

The ASR approach we develop in this thesis may be regarded as a contribution to solving the first two of these problems. We will demonstrate that the acoustic modeling component in a speech recognizer can benefit greatly from the articulatory feature approach by enabling the acoustic classifier to make more accurate bottom-up decisions. Furthermore, the robustness of speech recognizers in the presence of noise at high signal-to-noise ratios can be increased significantly by including articulatory information.

1.2 Overview of Thesis

The remainder of this thesis is organized as follows:

In Chapter 2 the potential advantages and disadvantages of the articulatory feature approach will be explained. Previous work on articulatory and acoustic-phonetic features in ASR will be discussed and evaluated in the light of recent developments in speech recognition.

Chapter 3 presents an articulatory-feature based recognition system for a small vocabulary recognition task (continuous numbers recognition). In this context we will discuss articulatory feature classification, feature selection, and the mapping from features to higher-level lexical units. We compare this system to state-of-the-art acoustic baseline systems and provide an error analysis of the characteristic differences between the two systems. Recognition results will be presented for clean speech, as well as for reverberant speech and speech corrupted by additive pink noise. Furthermore, frame-level methods of combining both systems will be investigated and combination results will be presented.

Chapter 4 extends this study to a large vocabulary conversational speech recognition task (spontaneous scheduling dialogues). The problems inherent in the development of an articulatory feature-based system for large vocabulary will be discussed and analyzed. Word recognition results will be given for clean speech and for various noise conditions (pink noise and babble noise). Again, both qualitative and quantitative error analyses will be presented. Furthermore, word-level combination strategies and feature-level combination strategies will be described and evaluated.

Chapter 5 gives a summary, discussion, and suggestions for future work.

Notational conventions

Throughout this thesis we will use

- lowercase bold letters (e.g. **x**) to denote vectors
- uppercase Greek or calligraphic letters (e.g. Ω, $\mathcal{Q}$) to denote sets,
- uppercase Roman letters (e.g. X) to denote random variables,
- uppercase P (e.g. $P(x)$) to denote probability mass functions, and
- lowercase p (e.g. $p(x|\omega)$) to denote probability density functions.

Chapter 2

Articulatory Feature Representations for Automatic Speech Recognition

In this chapter we will introduce and discuss arguments both in favor of and against articulatory feature representations in speech recognition. We will first describe in greater detail the particular approach to acoustic modeling which is advocated in this thesis and explain its underlying theoretical and methodological assumptions. We will then give an overview of previous approaches which make use of articulatory or acoustic-phonetic feature representations and evaluate them in the light of recent developments in ASR.

2.1 The Articulatory Feature Approach

The articulatory feature approach to acoustic modeling is schematically depicted in Figure 2.1. The basic idea of this approach is to use a speech signal representation which is intermediate between the preprocessed acoustic signal and the level of subword unit probability estimation, and which bears an affinity to the articulatory processes underlying the speech signal. This representation is composed of probabilities (or, more generally, scores) for so-called *articulatory features*, which are abstract classes describing the most essential articulatory properties of speech sounds, e.g. *voiced*, *nasal*, *rounded*, etc.[1] Articulatory feature probabilities are extracted from the preprocessed acoustic signal by a set of parallel, independent statistical classifiers. In a second step, this articulatory feature representation is mapped to scores for higher-level subword units, such as phones or syllables.

The particular choice of articulatory features and the arrangement of their corresponding classifiers which is proposed here is loosely based on the structure of human speech production. Human speech production involves the interaction of several articulatory components or dimensions which are partially independent of each other. This is reflected by the existence of separate

[1] Overviews of basic articulatory phonetics can be found in [76, 83, 25].

classifiers for these articulatory dimensions in the model shown in Figure 2.1. The first of these dimensions is *voicing*, which describes the state of the glottis and the activity of the vocal chords and which is largely independent of articulatory activities in the oro-nasal tract. We can further distinguish between the *manner* of articulation, i.e. the shape of a constriction made by an articulator in the vocal tract, and the *place* of articulation (the constriction location). The fourth articulatory dimension, lip *rounding*, is largely independent of most tongue body or tongue tip movements and can affect longer stretches of the speech signal. Finally, the relative position of the tongue on the *front-back* axis is another articulatory property which often shows a temporally independent behavior. Some of these articulatory dimensions are not entirely independent: although most constriction shapes, for instance, can be produced at most points in the vocal tract, there are certain places of articulation which are incompatible with certain manners of articulation: e.g. there are no glottal consonants which have a lateral constriction shape. We have chosen to *not* incorporate these interdependencies in the form of explicit constraints on the parallel arrangement of feature classifiers. The higher-level classifier which performs the mapping from the articulatory feature representation to larger subword units should be capable of learning restrictions on the co-occurrence of certain articulatory features in a data-driven way.

What are the potential advantages of this acoustic modeling structure for ASR? The main arguments in favor of this approach fall into the following four categories, each of which we will discuss in turn:

- robust statistical estimation,
- coarticulation modeling,
- selective processing, and
- noise robustness/speaker independence.

2.1.1 Robust Statistical Estimation

The task of classifying an acoustic observation vector $\mathbf{x}$ as one of several phone classes (the most common type of subword unit in current speech recognizers) is very complex due to variability in the speech signal. For N phones, this classification involves estimating N probabilities for a phone given an acoustic observation $\mathbf{x}$, $P(phone|\mathbf{x})$ (or, depending on the classifier, the N likelihoods of the observation $\mathbf{x}$ given a phone, $P(\mathbf{x}|phone)$). Assuming that the subword units in question are context-independent phones, N typically ranges between 30 and 60. In the cascaded classifier structure described above, each of the lower-level articulatory classifiers only needs to

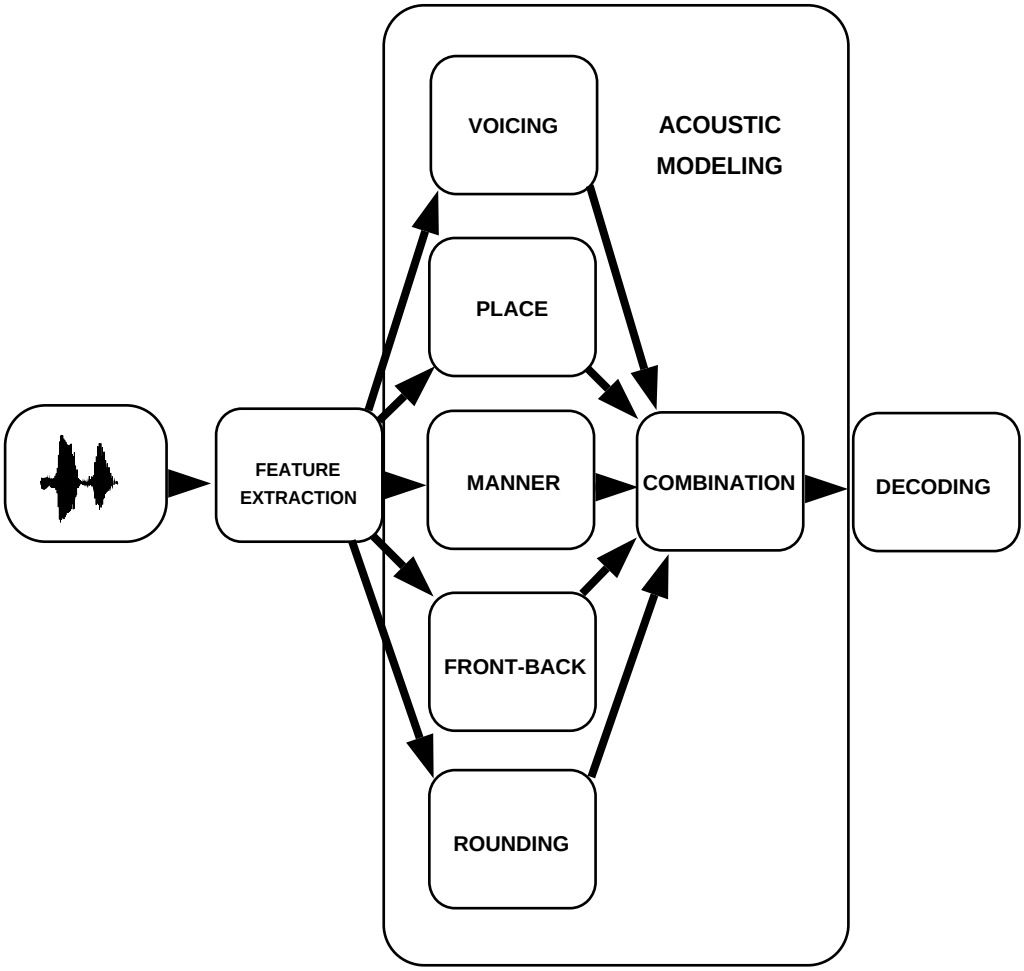

Figure 2.1: Articulatory feature approach to acoustic modeling.

distinguish between a very small number of output classes. Usually, the number of classes required to describe an articulatory dimension ranges from 2 or 3 (voicing) to approximately 10 (place). Thus, the complexity of each of the articulatory classifiers in terms of the number of output classes is lower than that of a monolithic phone classifier. In addition to this, the articulatory classifiers can exploit training data in a more efficient way: since manual articulatory feature annotations of speech signals are difficult and costly to produce, the only practicable way of generating training material for the articulatory classifiers is to convert phone-based training transcriptions into articulatory feature transcriptions. This can be done using a canonically defined phone-feature conversion table. Since articulatory features will generally occur in more than one phone, the training data for these features can effectively be shared across phones. This leads to a larger amount of training material for each feature classifier, which often exceeds the amount of phone training material by an order of magnitude. Table 2.1 shows the numbers of phone instances (and percentages of the total number of phone instances) in the German Verbmobil database, which is the speech corpus used for the recognition experiments described in

Phone	#	%	Phone	#	%	Phone	#	%
i:	18479	1.93	d	37139	3.88	I	62553	6.53
k	15870	1.66	y:	2809	0.29	g	14962	1.56
Y	6160	0.64	Q	22752	2.37	e:	18035	1.88
s	54491	5.69	E:	7418	0.77	z	15184	1.58
E	25566	2.67	f	22260	2.32	2:	1373	0.14
v	24987	2.61	9	3256	0.34	S	7325	0.76
a	64443	6.73	Z	24	0.00	a:	28469	2.97
C	18158	1.90	u:	6323	0.66	j	8122	0.88
U	20695	2.16	x	11327	1.18	o:	8883	0.93
h	6910	0.72	O	18417	1.92	m	37389	3.90
6	42440	4.43	n	90646	9.46	@	26627	2.78
N	6541	0.68	p	7352	0.77	l	19151	1.99
b	16529	1.73	r	15300	1.60	t	63915	6.67
<pause>	25377	2.67	<noise>	54555	5.69			

Table 2.1: Phone frequencies in the Verbmobil corpus. Phones are represented in SAMPA notation.

Chapter 4. These counts are based on an automatic labeling of the corpus produced at the Institute of Phonetics and Speech Communication at the University of Munich.[2] Table 2.2 shows the counts for feature labels derived from the phone annotations by means of the phone-feature conversion rules shown in Table A.2 in the Appendix, and the percentage of phones in which each feature occurs – note that since articulatory features occur in more than one phone, these percentages sum up to a value larger than 100.

These two properties (a smaller number of classes and more training material) should result in a higher recognition accuracy in each of the articulatory classifiers compared to that of a single, complex phone classifier. The hope is that this in turn leads to a higher accuracy of the overall classification procedure when the decisions made by the individual articulatory classifiers are combined by the higher-level classifier. Let us denote the entire set of articulatory classes by $\mathcal{A} = \alpha_1, \alpha_2, ..., \alpha_m$, which can be divided into k subsets $\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_k$ – in the particular partitioning which is used in this thesis, k equals five. The set of phone classes is denoted by $\Phi = \phi_1, \phi_2, ..., \phi_n$. Each phone ϕ_i, $1 \leq i \leq n$ can be encoded as a k-tuple of articulatory classes $\langle \alpha_1, ..., \alpha_k \rangle$, which is constrained such that every member of the k-tuple is contained in a different subset of $\mathcal{A}$. Thus, a phone is defined as a set of articulatory features, one from each articulatory feature group. Alternatively, all ϕ_i can be encoded as vectors of length m, with

[2] We are grateful to Florian Schiel, University of Munich, for providing these labels.

Feature	#	%	Label	#	%
+voice	648316	67.66	-voice	230387	24.04
vowel	319506	33.34	fricative	146528	15.29
stop	178519	18.63	lateral	19151	1.20
nasal	134576	14.04	coronal	280526	29.27
labial	71128	7.42	palatal	7349	0.77
high	143299	14.96	mid	57382	5.99
low	135352	14.13	velar	42159	4.40
uvular	15300	1.60	glottal	29662	3.10
front	142393	14.86	back	147230	15.37
+central	227717	23.77	-central	91789	9.60
+round	33623	3.51	-round	294030	39.69

Table 2.2: Feature frequencies in the Verbmobil corpus.

zeros for those articulatory features which do not contribute to the phone definition and ones for the relevant features. The probability $P(\phi_i|\mathbf{x})$ of a phone ϕ_i given an acoustic observation $\mathbf{x}$ can then be derived from the vector of probabilities of all α_j, $1 \leq j \leq m$, given observation $\mathbf{x}$. Ideally, the features which are relevant for the phone in question should have a probability of 1 whereas all others should have a probability of 0. Under what circumstances does this cascaded classification scheme lead to an improved accuracy of the final classification result? It is difficult to make precise predictions about this as the final classification result depends on several factors, e.g. the relevance of each of the lower-level classifier outputs for classifying the higher-level units, the statistical modeling assumptions of the higher-level classifier (e.g. whether a particular distribution of the data, such as a normal distribution, is assumed), the individual accuracies of the lower-level classifiers, and the (in)dependence of their errors.

Let us consider several possible situations. First, let us suppose that all individual articulatory classifiers produce correct classifications with a probability of 0.6 and incorrect classifications with a probability of 0.4. Let us additionally make the assumptions that the higher-level classifier produces an error if and only if *all* of the k lower-level classifiers produce an error, and that the errors of the lower-level classifiers are independent. In other words, incorrect classifications in the different articulatory classifiers are considered separate events occurring independently of each other. In this case, the error probability of the higher-level classifier is defined as the product of the lower-level classifiers' error probabilities:

$$P(e|\Phi) = \prod_{i=1}^{k} P(e|\mathcal{A}_k) \tag{2.1}$$

where $P(e|\Phi)$ is the error probability of the higher-level classifier and $P(e|\mathcal{A}_k)$ is the error prob-

ability of the $k'th$ lower-level classifier. For our current example this means that the probability of error in the higher-level classifier is 0.4^k and the probability of being correct is $1 - 0.4^k$. For $k = 5$, this equals 0.01 and 0.99, respectively. In this case, the cascaded classification scheme does produce a superior result compared to a simple classifier unless that classifier already achieves a very good performance, i.e. its probability of error already falls below 0.01.

Let us consider the situation where the higher-level classifier commits an error if *at least one* of the lower-level classifier makes an incorrect decision. In this case, the probability of error of the higher-level classifier can be defined as

$$P(e|\Phi) = 1 - \prod_{i=1}^{k}(1 - P(e|\mathcal{A}_k)) \tag{2.2}$$

which, in our current example, equals $1 - 0.4^5 = 0.92$. This might easily exceed the error probability of a one-step classifier.

Thus, the cascaded classification scheme may produce widely different outcomes depending on the sensitivity of the higher-level classifier to the estimation errors of the lower-level classifiers, on the probabilities of those estimation errors, and on the error correlation. In the first scenario (all individual classifiers must make an error for the higher-level classifier to be incorrect) good results may be achieved even when all or some the lower-level classifiers' error probabilities are higher than that of a comparable single-step classifier. This characterizes a situation where the individual classifiers may be not be very accurate but a large amount of redundancy exists between them. In the second scenario, there is little or no redundancy between the individual classifiers, and their error probabilities need to be very low if the final error probability is to be lower than that of a single-step classifier. In this case, the cascaded classification scheme may be useful only in situations where the lower-level classifiers are known to be highly accurate.

In practice, the lower-level classifier ensemble will be characterized by some amount of redundancy, and the decision of the higher-level classifier will not be dependent on either all classifiers being correct/incorrect or on a single incorrect decision. Furthermore, it will often be the case that some of the lower-level classes are more relevant than others for the classification of higher-level classes. A sufficiently powerful higher-level classifier should be able to compensate for these effects. In any case, it seems likely that the additional step of pre-classifying the highly variable acoustic signal into a set of classes which can be detected with higher accuracy provides for greater robustness of the overall classification process. In Chapter 3 these theoretical considerations will be further substantiated by empirical data.

2.1.2 Coarticulation Modeling

Coarticulation is the modification of a speech sound due to anticipation or preservation of adjacent speech sounds. These modifications are caused by the speech production mechanism: the sounds which listeners identify as speech segments are not produced in a serial, concatenative fashion but emerge from the coordination of parallel, overlapping articulatory gestures. In order to produce the sound [b], for instance, the jaw moves upwards, the lips form a closure, followed by a release, and the vocal folds vibrate. The timing of these gestures, however, is not simultaneous but highly overlapping, as can be seen from Figure 2.2. This figure shows the relative timing of the velum, tongue tip, tongue body, lips, and glottis gestures during the production of the English word *pan* and is an abstract representation of the actual articulator movements, as determined by X-ray studies [19]. The boxes represent the temporal extension of the movements of the articulators listed on the vertical axis. It is obvious that the gestures produced by different articulators which make up the phonetic segments on the horizontal axis do not follow identical temporal schemes but are largely desynchronized.

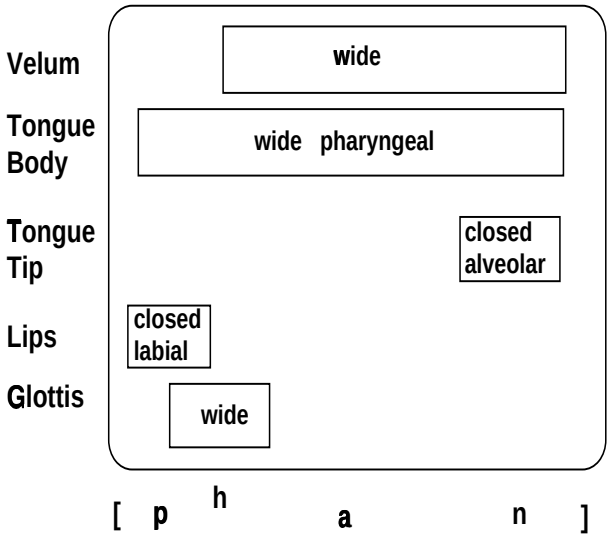

Figure 2.2: Relative timing of articulatory gestures for the production of the English word *pan*, after [19].

Due to the inherent inertia of articulators, articulatory constellations do not change rapidly from one speech segment to the next. Rather, gestures evolve relatively slowly over time and typically cover time spans containing several speech segments. The spectral properties of these segments are accordingly affected by the way the gesture changes the vocal tract resonance properties. These spectral modifications form a continuum from highly perceptible changes causing a shift in segment identity, over perceptible but non-distinctive changes, to subtle effects which can not be perceived by the human ear but which may have an effect on the statistical model of the

sound in question. An example of the first case is the assimilation of nasals to following stops common in many languages, e.g. *i/n/ Britain* → *i[m] Britain*. The second category includes phenomena like the shift in the place of articulation of velar plosives depending on the quality the following vowel: the [k] in *kitchen*, for instance, has a more palatal quality than the [k] in *car*. All instances of coarticulation influence the acoustic representation of speech sounds and may obscure the mapping from acoustic parameters to speech sound models.

The most common way to approach this problem in speech recognition is to use context-dependent acoustic models, such as biphones or triphones. In this case, separate models are constructed for segments (phones) in different contexts. Thus, a /u/ between /t/ and /b/, for instance, would receive a different model than a /u/ between /k/ and /l/. Depending on the vocabulary size, this approach may yield a very large model set. This in turn may lead to severe undertraining of the less frequent context-dependent models, for which only a few training instances may be present in the training database. Various remedies can be applied to alleviate this problem, such as interpolating the parameters of undertrained models with those of well-trained models [77], merging similar models on the basis of phonetic knowledge [32] or by means of data-driven clustering [79], setting thresholds on the frequency of occurrence of certain triphones [77], or tying the parameters of different context-dependent models [125]. Another way of modeling coarticulation is to add explicit pronunciation rules to the recognition lexicon in order to capture those coarticulation phenomena which can be described at the level of phone symbols, i.e. those which cause a change in segment identity. These usually take the form of alternative paths in the (phone-based) transition networks for lexicon entries.

These approaches ignore the actual source of coarticulation and the potential advantages which might be gained from a direct description of this source. As articulatory studies [18, 19] have shown, most coarticulatory phenomena can be traced back to a temporal and/or spatial reorganization of articulatory gestures. Gestures may be compressed, i.e. overlap for longer time spans, for instance due to increased speaking rate, or they may have a smaller magnitude (articulatory undershoot). If it were possible to construct a reliable articulatory representation, coarticulatory phenomena might be modeled simply in terms of these basic manipulations of articulatory gestures. In the spectral or cepstral domain, by contrast, these gestural modifications may generate complex patterns which are difficult to interpret or to model.

Articulatory features are related both to the acoustic signal and to higher-level linguistic units, such as phones and syllables. They therefore provide a more suitable description language for pronunciation variants, allowing words in the recognition lexicon to be represented not in terms of rigid phone sequences but in terms of parallel sequences of articulatory features which are loosely synchronized.

2.1.3 Selective Processing

It is reasonable to assume that different aspects of articulation exhibit different degrees of robustness and do not deteriorate (in terms of their ability of being recognized correctly) to the same degree under adverse acoustic conditions. Voicing distinctions, for instance, can be detected fairly robustly across a variety of acoustic conditions [26]. The detection of place features, by contrast, is presumably less robust as it requires recovering the point of articulatory constrictions in the vocal tract mediated by the acoustic signal. The acoustic changes induced by different constriction locations, however, are heavily dependent on speakers' vocal tract characteristics. A classifier structure which is based on the decomposition of speech sounds into their articulatory components can exploit this property by selectively applying different processing strategies to the different sub-classifiers independently. These strategies may involve the use of temporal windows of different lengths, separate feature extraction front-ends, and different speech enhancement or model adaptation algorithms. In addition to being able to selectively focus on the more robust properties, this technique opens up possibilities for more constrained adaptation procedures in that adaptation only needs to be applied to the models of those features which are most strongly affected by noise or speaker variability. Furthermore, the articulatory classifiers themselves may differ: the classifier type, the complexity (the number of free parameters), and the initialization or training procedures may be tuned to the specific tasks they need to perform.

In addition to using selective processing strategies at the first classification stage, the contributions of the sub-classifiers to the overall classification task may also be weighted differently by the combination module depending on the context. The combining module may, for instance, use an assessment of the signal-to-noise ratio as a basis for assigning weights to the various sub-classifiers. Alternatively, this kind of selective adaptation might be achieved by re-training the combination module on a small amount of noisy speech data.

How does the AF approach compare to other approaches to decompositional acoustic modeling? Another prominent decompositional model which has recently gained popularity is the so-called subband model. In subband systems [17, 15, 16, 35, 90] the acoustic frequency band is decomposed into a number of narrower frequency bands, *subbands*. In each of these subbands, the subword unit probabilities are estimated separately and combined by a higher-level classifier (see Figure 2.3). This scheme has certain parallels to our model: both approaches employ a cascaded classifier structure where the higher-level classifier combines the probability estimates of the lower-level classifiers. However, there are also a number of important differences: first, the classifiers in each subband and the combining classifier share the same set of output classes, i.e. they all estimate probabilities for the same number and the same type of classes. Compared to a full-band classifier, however, the subband classifiers receive less information since they only have access to a small portion of the full frequency band. Thus, in a subband system the inputs to

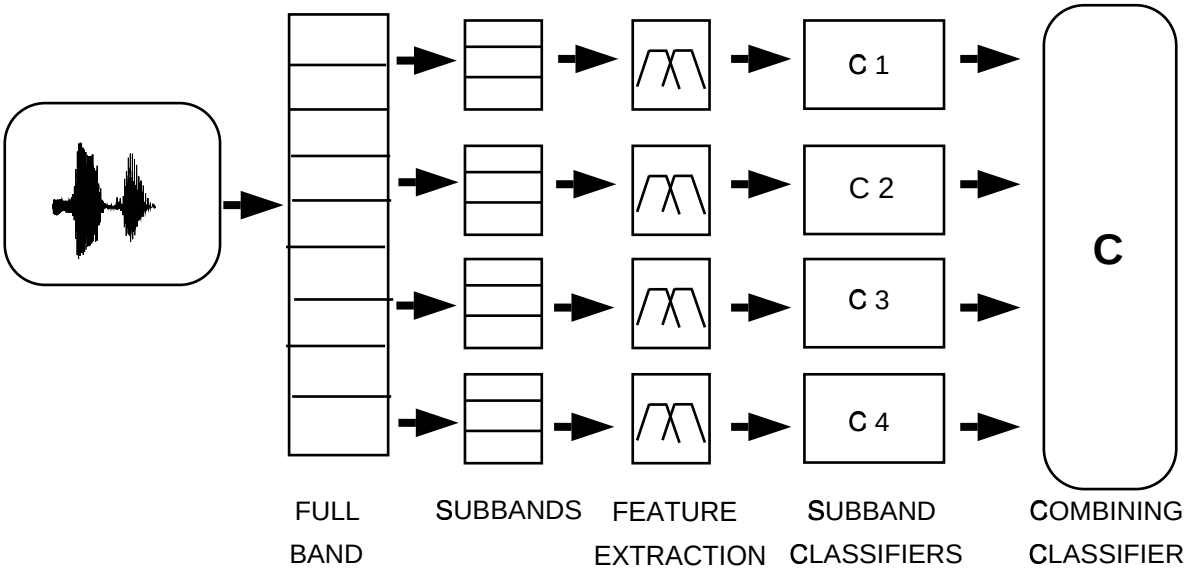

Figure 2.3: Subband system.

each lower-level classifier provide a reduced amount of information but the classifiers are nevertheless required to perform a task which has the same complexity as that of the higher-level classifier. In the AF model, by contrast, each lower-level classifier receives the same amount of information but is tuned to a specialized classification task, each of which has lower complexity than the task of the higher-level classifier. Another disadvantage of the subband approach is that a subband system is not able to optimally exploit training material by sharing data across phones. Furthermore, the subband approach relies on traditional (phonemic) subword units, which do not offer the same advantages as the AF approach with respect to coarticulatory modeling. Another difference is that there exists great uncertainty as to how many subbands should be used, which bandwidths they should have, and whether they should overlap or not. Various schemes have been suggested (see the references above); it seems to be the case that no single subband scheme can be identified which works equally well across different tasks and recognition conditions. This may entail the need for a redefinition of the subbands when switching to a different recognition task, which is undesirable. The AF approach does not suffer from this problem: whereas a certain amount of unanimity does exist with respect to the exact feature set which should be used in an AF system, the basic structure of the system and the choice of features are constrained by a model of the human speech production mechanism and are thus less task-dependent. The only case in which a subband system might theoretically show an advantage over an AF system is when (a) the signal is corrupted by narrow-band noise, and (b) the bandwidth of the noise is known in advance. In this case, subbands can be defined such that the noise-corrupted subband can be completely excluded from the higher-level classification step, which would not be possible in an AF-based system. However, the characteristics of background noise are generally not

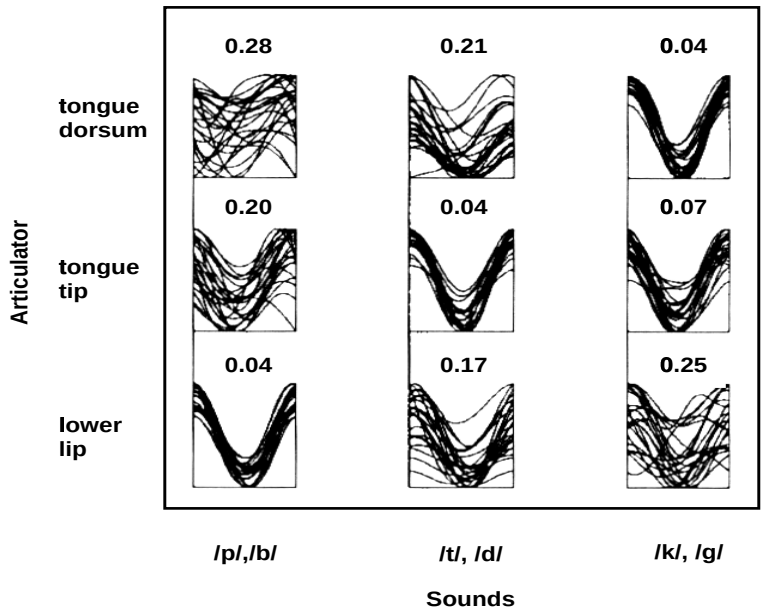

Figure 2.4: Trajectories of vertical movements of tongue dorsum, tongue tip and lower lip for labial, coronal, and velar sounds. From Papcun 1992 [97].

known in advance under realistic test conditions. An online adaptation of the subband split according to an estimate of the noise bandwidth would necessarily introduce a mismatch between training and testing conditions and is therefore infeasible. For these reasons, we consider the AF scheme the more advantageous decompositional acoustic modeling approach.

2.1.4 Noise Robustness/Speaker Independence

To the extent that abstract pseudo-articulatory features are able to reflect the actual articulatory properties of speech, they also offer the advantages of noise robustness and speaker independence, which are usually ascribed to genuinely articulatory representations (direct physical measurements of articulation).

Although speakers differ with respect to the precise shape and magnitude of their articulatory gestures, articulatory trajectories pertaining to particular speech sounds show remarkable uniformity across speakers. Figure 2.4 shows tongue tip, lower lip and tongue dorsum pellet trajectories for different consonants as recorded in an X-ray microbeam experiment [97]. These trajectories were normalized for duration and magnitude before computing the correlation across trajectories, also shown in Figure 2.4. It is obvious that each group of consonants which share the same place of articulation (location of the constriction in the vocal tract) can be identified by a characteristic trajectory of the active articulator, which shows very little variation.

These articulatory movements are often reflected in the acoustic signal by a characteristic spec-

tral pattern, such as the development of formant frequencies over time. For instance, it can be observed that lip rounding causes a downward shift of all formants[3] in the spectrum. These patterns are in principle independent of the vocal tract lengths of different speakers and acoustic conditions such as reverberation and additive noise.

It is well known that different vocal tract lengths have a severe effect on acoustic speech signal representations. The length of the vocal tract causes a quasi-linear shift of frequencies in the acoustic signal: a shorter vocal tract, as in children and female speakers, causes frequency energy to shift upwards whereas a longer vocal tract is characterized by lower frequencies. Moreover, the degree of the frequency shift depends on the vocal tract configuration: open vocal tract configurations (configurations during the production of open vowels) are more severely affected than closed configurations. These effects may be balanced by so-called Vocal Tract Length Normalization (VTN). In VTN [67, 36] a scaling factor k is applied to the preprocessed speech signal to achieve a linear frequency warping,

$$f' = kf \tag{2.3}$$

which is expected to normalize the variabilities introduced by different vocal tract lengths of different speakers. In [67], an exhaustive search is performed to find the optimal scaling factor k_s for each speaker s during training. During decoding, the utterances of each speaker were decoded with twenty different scaling factors. That decoding which maximized the likelihood of the data was then selected as the optimal hypothesis. In [36], a parametric approach is suggested which eliminates much of the computational overhead associated with the exhaustive search for the optimal scaling factor. Under this approach, a formant configuration is estimated for each speaker and the scaling factor is computed from the median of the third formant (F_3) in relation to the F_3 median across all other speakers. Although this and other recent improvements to VTN make the method more practicable, VTN is associated with additional computational effort. Articulatory classes, by contrast, are mainly determined by *relative* acoustic patterns, such as the direction of formant movements or the relative distance between formants. For this reason, they can be expected to be more robust to vocal tract length differences than direct spectral representations.

Most current acoustic speech representations are based on the log-spectrum of the signal. For the computation of the log-spectrum the signal is first subjected to a windowed Fast Fourier Transform (FFT). The output is then passed through a filterbank; in the case of mel-frequency cepstral coefficients (MFCCs), which are the most widely used speech features, it consists of a set of trapezoidal filters $f_1, f_2, ..., f_m$ equally spaced along the mel-scale. The mel-scale was

[3] Formants are those frequencies in the spectrum of the speech signal which are associated with relative maxima of energy.

developed on the basis of human auditory perception experiments [115] and is approximately linear below 1 kHz and logarithmic above. It can be approximated by the following formula [42]

$$Mel(f) = 2595 log_{10}(1 + \frac{f}{700}) \tag{2.4}$$

Subsequently, the log, l_i, is taken of the amplitude in each filterbank channel, f_i. Finally, the cepstrum is computed by means of the Discrete Cosine Transform (DCT):

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{M} l_j cos(\frac{\pi i}{M}(j - 0.5)) \tag{2.5}$$

MFCCs have the advantage of modeling the quasi-logarithmic frequency resolution of the human ear. The DCT yields the additional advantage of orthogonalizing the coefficients, which can be important if an acoustic classifier is employed which does not model the correlation between different coefficients, such as a single-mixture Gaussian classifier with diagonal covariance. However, the *log* operation has the effect of emphasizing even small levels of background noise. Furthermore, the DCT spreads this effect across all coefficients although noise may originally only be present in some frequency regions. For this reason, log-spectral representations in general, and MFCCs in particular, are especially sensitive to noise.

Both signal-based and model-based methods have been developed to compensate for noise and channel effects. A signal-based method for removing long-term spectral biases is Cepstral Mean Subtraction (CMS), where the mean of the cepstral coefficients is first computed over some amount of data (the entire utterance in the case of off-line applications) and is then subtracted from each feature vector. An example of a model-based adaptation method is Maximum Likelihood Linear Regression (MLLR) [80, 49]. Under this approach, a set of linear transformations is applied to the means and variances of the models in a Gaussian mixture system. These transformation matrices are estimated such that the likelihood of the adaptation data is maximized.

The various methods for speaker and noise adaptation are successful to some degree but it is obvious that they frequently involve considerable additional computational effort. It would be desirable to employ features which are inherently less sensitive to noise and speaker variability or a classifier structure which allows these algorithms to be employed selectively, in order to minimize adaptation requirements. The articulatory feature approach potentially offers both of these advantages.

2.2 Relation to Human Speech Perception

At this point it might be asked what relation, if any, the articulatory feature approach bears to human speech perception.

From the 1950s to the 1970s, a number of perceptual confusion studies were carried out, which have often been cited as a source of evidence supporting the assumption of perceptual reality of articulatory/acoustic-phonetic features. These studies (e.g. [89, 113, 120]) have shown that perceptual confusions of vowels and consonants under clean and noisy listening conditions often pattern along articulatory dimensions: segments which are highly confusable can mostly be described as differing in only one or two articulatory features. Miller and Nicely [89], for instance, studied the confusion of initial consonants before /a/ in nonsense syllables in clean speech, filtered speech, and speech with additive white noise at various signal-to-noise ratios. 200 syllables were produced by five different talkers and transcribed by four different listeners. For each acoustic condition, the confusion matrices of spoken vs. perceived consonants were computed and analyzed. Whereas high-pass filtering of the speech signal (the lower cut-off frequency ranging between 2000-4500 Hz) led to randomly distributed perception errors which did not show any phonetic pattern, clean, low-pass filtered and noise-masked speech mostly produced confusions among segments which were similar in articulatory terms, e.g. confusions of the voiced and voiceless stops /b/-/p/, /d/-/t/, /g/-/k/. All consonants were then grouped into super-classes defined by the five articulatory dimensions *voicing*, *nasality*, *frication*, *duration*, *place* and the mutual information between the spoken and the transcribed class was computed according to

$$I(X;Y) = -\sum_{X,Y} p(X,Y) log \frac{P(X)P(Y)}{P(X,Y)} \tag{2.6}$$

where X and Y are random variables representing the spoken and transcribed class, respectively. It was found that the five individual mutual information values added up to the mutual information for the phone classes, provided the sum was normalized by a corrective factor for the redundancy in the input. The authors concluded from this that the five feature groups were perceptually independent and proposed the *multi-channel* model of speech perception, which assumes the existence of independent perceptual channels associated with an articulatory interpretation, the output of which is combined to identify phone classes. The multi-channel model is thus very similar to the acoustic modeling approach proposed here. Note that Miller & Nicely's experiments do not *prove* the existence of five perceptual channels which correspond precisely to the above feature groups. The interpretation of their data is to some extent pre-determined by the *a priori* choice of the five descriptive categories. There is no evidence which disproves the existence of fewer or more than five perceptual channels. It should also be emphasized that a distinction should be made between the claim that articulatory features actually *are* the units of speech perception and between the fact that they may serve as convenient labels for describing certain perceptual patterns. Although articulatory features describe Miller and Nicely's findings most concisely and naturally, nothing suggests that human listeners actively exploit them in the process of speech perception.

An additional source of evidence in favor of the perceptual reality of articulatory features comes from similarity judgment experiments. In these studies it was found that sounds were judged more similar by human listeners when they had more articulatory/phonetic features in common [99, 54].

More recently, Ghitza [53] established a mapping between Jakobson's [64] distinctive phonological features (which partially overlap with articulatory features) and specific time-frequency regions in diphones through so-called "tiling" experiments. Subjects were presented with CVC (consonant-vowel-consonant) words differing in the initial or final diphone and were asked to perform an ABX listening test, i.e. three stimuli were played to the subject, of which the third had to be identified as either the first or the second one. The CVC lists were modified by dividing the signal portions belonging to the diphone into six equal time-frequency regions ("tiles"), which were then interchanged paradigmatically. By analyzing the resulting perceptual confusions a direct mapping could be established between certain time-frequency regions and distinctive features. Since time-frequency information can further be related to auditory-nerve function, it may be concluded from this study that the features investigated have perceptual reality by being coded in the auditory periphery.

There are some phonetic and psycholinguistic theories which take an extreme point of view on the question of the perceptual relevance of articulatory categories and claim that articulatory categories (also termed motor commands or articulatory gestures) exclusively form the basis for human speech perception, i.e. that listeners perceive sounds by reconstructing the relevant articulatory gestures from the speech signal. The first theory of this kind was the *Motor Theory of Speech Perception*, proposed by Liberman in 1967 [81] and revised Liberman and Mattingly in 1986 [82]. Fowler's Modularity Theory [45] and Browman and Goldstein's Articulatory Phonology [18] also fall into this category.

The main argument of the Motor Theory in support of articulation-based speech perception is the fact that acoustic correlates of phonetic categories are not uniquely identifiable. Due to coarticulation the acoustic properties of a given sound differ widely in different phonetic contexts. It has been shown in speech perception experiments (e.g.[86]) that phonetic distinctions such as voicing can be signaled by a variety of acoustic cues. All of these cues can assume a distinctive function in certain contexts; however, none of them is indispensable for the discrimination of phonetic categories because it can always be substituted by sets of other cues. Thus, it seems impossible to enlist those acoustic properties which are responsible for the perception of speech sounds across all contexts. Furthermore, the proponents of Motor Theory argue that it is cognitively more plausible to assume that humans possess a single representation and/or processing module for both the production and the perception of speech than to posit the existence of two separate, highly specialized modules for these tasks. It is proposed that production and per-

ception share a common representation and perhaps a common processing strategy, which are presumably innate.

The motor theory seems be supported by certain experimental findings and empirical observations, such as the fact that human listeners, in situations where they encounter speech perception problems, involuntarily mouth the corresponding articulatory movements. On the other hand, there are listeners (speech-impaired adults and prelinguistic infants) who are incapable of articulating speech sounds, but who are nevertheless able to process and understand speech. Thus, knowledge about articulatory gestures does not seem to be critical for the perception of speech. Furthermore, the argument adduced against an auditory theory of speech perception, viz. the variety of acoustic cues, can in principle also be applied to the Motor Theory itself: although articulatory gestures, as we have already noted, are contextually less variable than acoustic parameters, they need to be retrieved from the acoustic signal, which requires a similarly complex transformation as purely auditory speech perception. In sum, it is highly questionable that articulatory gestures are indispensable for the perception of human speech sounds. However, it does seem likely that in certain situations articulatory representations can *help* classify speech sounds into linguistic categories.

2.3 Drawbacks

Despite the advantages of articulatory representations described above, most state-of-the-art speech recognition systems do not incorporate any articulatory information. This can be explained by the two major drawbacks of articulatory representations: first, it is difficult to reliably extract articulatory parameters from the acoustic signal; second, the use of articulatory information requires additional processing, which has up to now prevented the integration of this approach into large-scale applications.

The only feasible way to make use of articulatory information in speech recognition is to map the acoustic signal to an articulatory representation. This is a necessary prerequisite because direct articulatory measurements, e.g. in the form of X-ray data, are not available in normal speech applications. However, the reconstruction of articulatory movements from the acoustic signal is greatly complicated by what is usually termed the *inversion problem*. The inversion problem denotes the lack of a one-to-one mapping between articulation and acoustics. Widely differing articulatory constellations can generate highly similar acoustic patterns. This has been demonstrated empirically in so-called bite-block experiments [51]. In these experiment, subjects were asked to produce speech while their articulation was artificially impaired by a physical obstruction in the vocal tract. Under a variety of different obstruction conditions, subjects were still able to generate perfectly intelligible speech showing similar acoustic patterns. The in-

version problem also besets the reverse transformation from acoustics to articulation. Atal et al. [3] investigated acoustic-articulatory mapping with the objective of computing an inverse of the articulatory-to-acoustic function. Using a computer-sorting technique to derive formant bandwidths and amplitudes from articulatory variables such as cross-sectional areas of the vocal tract, they noticed that "large ambiguities were observed. Large changes in the shape of the vocal tract can be made without changing the formant frequencies" ([3]:1555). This observation has been replicated in several subsequent studies concerned with finding an acoustic-articulatory mapping function, e.g. [13, 106, 107, 105]. In sum, the relation between acoustics and articulation is non-unique and highly non-linear. However, although the mapping problem cannot be solved deterministically, it can be conceived of as a problem of statistical pattern recognition, where powerful (nonlinear) classifiers and sufficient training material may contribute to establishing a robust mapping between acoustic input data and articulatory classes in a probabilistic fashion. Moreover, inversion may constitute a serious problem for local classification based on a single acoustic frame but may be easier to perform over a longer time span consisting of multiple frames. Thus, dynamic constraints incorporating statistics of the durations of articulatory gestures may greatly simplify the inversion task.

The second reason why articulatory representations have not been used extensively in speech recognition systems is the additional cost associated with them. If articulatory parameters are statistically extracted from the acoustic representation, an additional processing step is required. Computing an inverse transformation of the articulatory-acoustic mapping is similarly expensive, as it usually requires a search of the articulatory space. However, this additional cost may be acceptable if it simplifies higher-level processing, such as the evaluation of acoustic models or the lexical search during decoding. Furthermore, a modular, parallel recognition architecture may be capable of handling the additional processing requirements in real time.

2.4 Previous Work

There have been several previous attempts at using either articulatory or acoustic-phonetic information in ASR. These fall into three main categories, viz. the extraction of articulatory parameters from direct physical measurements, articulatory/acoustic-phonetic feature systems, and articulatory-based preprocessing.

2.4.1 Direct Physical Measurements

The most direct and accurate way of describing articulation is to physically record the movements of articulators. Various methods have been developed for this purpose, e.g. cineradiogra-

phy, where metal pellets are attached to a subject's articulators (typically lips, tongue tip, tongue dorsum, and jaw), whose movements are then recorded by X-ray. The displacement of the pellets from the neutral position across time yields a fairly accurate account of articulatory movements. Papcun et al. [97] and Zacks et al. [126], used X-ray microbeam data coupled with acoustic data (FFT-based bark-scaled coefficients) to map acoustic parameters to articulatory trajectories using a neural network. Whereas Papcun et al. [97] describe the mapping from acoustics to articulatory trajectories and articulatory gestures in detail, an actual application of the technique to vowel recognition is described in the paper by Zacks et al. [126]. The data set used for training and testing consisted of 45 words (3 repetitions of 3 words spoken by 3 speakers), each word containing one vowel. In one experiment, this data set was used for both training and testing; in a second experiment, all data for one speaker was deleted from the training set and used as the test set. A vowel recognition accuracy rate of 96% was obtained for the first experiment; the second experiment yielded 87% accuracy.

2.4.2 Articulatory Feature Systems

A number of speech recognition systems have been built which use articulatory features similar to the approach described in this thesis. A closely related concept is the concept of "distinctive features" in phonological theory [23]. Distinctive feature systems and articulatory feature systems often include similarly named features; however, there is an important distinction to be made. Distinctive feature systems were primarily developed for the purpose of phonological classification and thus may include features which bear little relation to physical parameters. Examples of these are *syllabic* or *consonantal* which have the functional purpose of grouping together certain classes of speech sounds – they do not have unique correlates either in acoustic or in articulatory space. In this thesis, we explicitly restrict our set of features to those which can be expected to have well-defined correlates in articulatory space and exclude any functional or purely acoustically defined features.

In the following discussion of feature-based approaches in ASR we will nevertheless include those studies which use distinctive or acoustic-phonetic features. The common characteristic of these approaches is that a pre-defined set of features is established for the language under investigation. These features are then detected from the acoustic speech signal by means of statistical classifiers and are subsequently used to define higher-level units like phones, syllables or words.

One of the earliest systems which make use of articulatory features was reported by Schmidbauer [110]. Schmidbauer developed a speech recognizer for German using 19 articulatory features describing manner and place of articulation. These were detected from the preprocessed speech

signal by means of a Bayes classifier. The posterior feature probabilities were concatenated to articulatory feature vectors (AFVs), which were then used as input to phonemic hidden Markov models (HMMs). On a small database (about 10-15 minutes for both training and testing), an improvement of 4% over a standard system based on MFCCs was achieved. The author observed that the AF system was more robust towards cross-speaker variability and showed a smaller variance of recognition accuracy across different phoneme classes than the MFCC system.

Dalsgaard and colleagues [29, 114] used three-valued articulatory features for the purpose of multi-lingual labeling. 20 (Danish) to 25 (British English) features were detected by a self-organizing neural network (SONN). The SONN output was used as input to multivariate Gaussian mixture phoneme models, which were used for automatic label alignment by a Viterbi algorithm. The result was evaluated with respect to a manual annotation of label boundaries, but no comparison of recognition accuracy to a standard acoustic system was given.

Eide et al. [37] used 14 acoustic-phonetic features for broad class phonetic classification and keyword spotting for an American English database. These features mostly had an articulatory interpretation but also included functional features like *consonantal* and *continuant*. Feature probabilities were derived using Gaussian classifiers after having obtained a broad class Viterbi segmentation of the waveform. Phoneme probabilities were then defined by a product combination of feature probabilities, together with a duration model. Under this scheme a phoneme classification accuracy of 70% was obtained on the TIMIT database. For the purpose of keyword spotting, the same approach was used on narrow-band (telephone) speech. Significant improvements were also obtained when a baseline MFCC-based system was combined with the feature-based system by means of a cost function involving hand-tuned weights.

Probably the most elaborate articulatory feature system has been developed by Deng and colleagues [31, 40, 34, 33, 41]. The authors used 18 multi-valued features to describe the four dimensions of voicing, place of articulation, vertical tongue body movement and horizontal tongue body movement. Furthermore, they modeled the speech signal as a sequence of target articulatory vectors interspersed with transitional vectors. Target vectors were defined by a rule-based combination of articulatory features. The transitional vectors were underspecified; features were allowed to assume any phonetically plausible value intermediate between the value of the previous target vector and that of the following target vector. Individual vectors corresponded to HMM states; all possible vectors were combined into a single ergodic HMM whose transitions and emissions were trained. Early experiments reported a relative improvement of 26% on average over a conventional phoneme HMM on a consonant-vowel speaker-independent classification task. Phone recognition experiments on the TIMIT database yielded a 9-14% relative improvement compared to the acoustic baseline system. Finally, on a speaker-independent medium-sized word recognition task, the AF recognizer achieved a 2.5% relative improvement

compared to a single-component Gaussian mixture phoneme recognizer.

Further results on phonetic-feature classification (including articulatory features) were reported in Windhauer et al. [123]. 18 features were detected using time-delay neural networks; the outputs were multiplied to yield phoneme probabilities. Recognition results were reported on the ALPH English spelling database; however, no comparison to a standard acoustic system was given.

Elenius et al. [39, 38] compared hybrid ANN/HMM classifiers for phoneme recognition based on spectral vs. articulatory representations. Seven phonetic features were used as an intermediate level in a phoneme classifier and compared to a phoneme MLP without any intermediate feature level. On a speaker-dependent speech database consisting of 50 sentences for both training and testing, the authors found that the spectral classifier performed better. However, they did observe an advantage of the articulatory feature-based classifier for speaker-independent data.

2.4.3 Articulatory Preprocessing

The approaches which can be grouped together under the label "articulatory preprocessing" seek to develop a parameterization of the speech signal which enhances acoustic-phonetic or articulatory information. No statistical classification is involved to extract these categories; instead, specialized preprocessing front-ends are designed based on knowledge about the acoustic correlates of the categories in question. Thus, correlates for features like *noncontinuant*, *fricated*, *palatal*, etc. are defined in terms of energy in specific frequency bands, energy ratios between different bands, zero-crossing rates, normalized auto-correlation coefficients, etc.

Bitar & Espy-Wilson [11, 12] used articulatory parameters as input to HMMs for phoneme classification. Compared to MFCCs, relative improvements of up to 11% were achieved on the TIMIT database. These improvements were particularly obvious in cases where there was a gender mismatch between the training and the test data.

Varnich-Hansen [57] also described the development of novel acoustic parameters for distinctive feature and phoneme classification. Classification results were given for the TIMIT database but were not compared to standard acoustic parameters.

Ali et al. [1] developed an acoustic front-end for the recognition of fricatives, which identified voicing and place of articulation. The front-end consisted of critical-band filters, a hair-cell synapse model, and a generalized synchrony detector. Relevant acoustic features included energy in different frequency bands, relative amplitude and spectral shape. This model was tested on fricatives excised from the continuous TIMIT database. Remarkably high accuracy rates (95% for voicing, 93% for place of articulation) were reported.

Ramesh and Niyogi [93, 102] concentrated exclusively on the voicing feature and on improving discrimination between stop consonants. A detector for voice onset time (VOT) based on cross correlation and dynamic programming was used to classify segments as either voiced or unvoiced. This classification was performed after a first-pass segmentation had been carried out by a conventional HMM system. This strategy led to an overall reduction of 48.7% of stop-consonant identification error rate obtained on a spelling database.

2.4.4 Evaluation

The first of the approaches described above, the use of direct articulatory measurements, has only marginal relevance to ASR because of the lack of large, speaker-independent databases of articulatory measurements. The existing articulatory databases do not provide sufficient training material for statistical models in order to learn articulatory parameters from the corresponding acoustic data for a broad variety of contexts.

Articulatory-based pre-processing can potentially be useful in improving speech recognition. Its advantage is that no intermediate level of articulatory feature classification is required; the parameters extracted from the speech signal can be directly used to estimate the distribution of subword units, in the same way as standard acoustic parameters. However, the articulatory pre-processing (*extraction*) approach is essentially a rule-based approach. The acoustic correlates of the categories to be extracted are defined on the basis of expert knowledge, which renders the development of successful feature extraction algorithms difficult and slow. Moreover, these algorithms have up to now only been tested on clean, full-band speech. It is to be expected that at least those definitions of acoustic correlates which rely on absolute frequency or energy values will have to be redefined for noisy or band-limited speech. As mentioned above, articulation exhibits context-dependence. Voicing, for instance, may be easily identifiable by voice-onset time for stop consonants in syllable-initial position. However, stops in different contexts and other consonants such as fricatives may have a less well-defined realization. Thus, a rule-based articulatory pre-processing approach may work well when a first-pass segmentation of the signal has already been performed or when certain “landmarks” have been detected first [87] so that the context can be defined more accurately, but it is suboptimal for purely bottom-up acoustic classification. Thus, in spite of the computational effort associated with it, the statistical classification component seems to be indispensable for an articulatory approach to ASR. An articulatory representation may in itself exhibit less variation than a spectral representation; however, the acoustic patterns from which the articulatory categories are to be recovered are too varied and too intransparent to be amenable to a rule-based feature extraction method. It may be for this reason that the most promising ASR results described above have been obtained by the articulatory feature

classification approach.

The preceding section has shown that the AF approach to speech recognition is not an entirely new idea. However, several important research issues have been neglected or entirely omitted in these previous studies, which explains the motivation of this thesis:

- First, little effort has been devoted in general to the articulatory approach. Attempts at developing AF systems have been sporadic and in many cases have been abandoned prematurely before their potential was exhaustively analyzed and exploited. Furthermore, the full range of statistical classification and optimization tools obviously has not been applied to most of these systems. In most cases, a set of articulatory features was selected heuristically. The output of the feature classification component was then used as input to the same higher-level processing components that were employed in the acoustic baseline systems, which may or may not have resulted in an improvement in performance. It is evident, however, that the introduction of a new set of features, such as articulatory features, always requires extensive system optimization and fine-tuning in order to achieve results comparable to those of an acoustic baseline system which in many cases has been developed and optimized over many years.

- Most of the research efforts in articulatory speech recognition have been directed at the extraction of articulatory parameters from the acoustic speech signal. However, it is at least equally important to find out how these parameters function in a complex speech recognition system.

- Most of the studies mentioned above do not provide a detailed error analysis of the strengths and weaknesses of the AF systems vs. the acoustic baselines. Systems are merely compared in terms of phoneme or word error rates, which yields very little information about the conditions in which AFs may or may not be useful. Eide et al. [37] note in passing that “the linguistic features extract information in the waveform differently from the HMM” ([37]:486) but a more detailed analysis has not been performed either in this or other studies.

- This introduces another issue which has not been sufficiently investigated: if AF representations and acoustic representations provide different information, it might be advantageous to combine them. However, very little effort has been directed towards this goal.

- Previous AF studies have only looked at limited speech recognition tasks, such as phoneme identification. To the best of our knowledge, the potential of AFs for conversational speech or large vocabulary has not been investigated.

- Although the potential of AFs in the presence of noise and other adverse acoustic environments has repeatedly been asserted, it has never been systematically tested.

This thesis will address all of these points by investigating AFs under a broad variety of acoustic and linguistic conditions, including clean, noisy, and reverberant speech, full-band as well as narrow-band speech, different recognition tasks – numbers recognition and conversational dialogue recognition – as well as two different languages, American English and German. Throughout this thesis, we will follow a feature classification rather than a feature extraction approach. Particular emphasis will be given to the performance of AFs within an entire speech recognition *system*. Moreover, detailed error analyses will be provided of the strengths and weaknesses of the different representations. Finally, different approaches to combining both systems will be presented.

Chapter 3

Small Vocabulary Recognition Using Articulatory Information: A Pilot Study

In this chapter we will present a pilot study which highlights some of the research issues which we will address in greater detail in the subsequent chapters of this thesis. We will describe the development of a recognition system based on articulatory features and compare its performance on a small-vocabulary continuous numbers recognition task to that of state-of-the-art acoustic recognizers. Characteristic differences between the acoustic and articulatory systems will be analyzed at various levels of the recognition system. We will show that the recognizers exhibit different error patterns and thus indicate potential for recognizer combination. Finally, we will discuss and investigate frame-level combination strategies and present combination results.

3.1 Corpus and Baseline Systems

3.1.1 Corpus

The corpus used for the experiments reported in this chapter is the OGI Numbers95 corpus [27]. This is an American English corpus consisting of a collection of continuously spoken numbers – a typical utterance in this corpus is *two hundred thirty six*. The utterance length ranges between one and ten words with an average of 3.9 words per utterance. The corpus was compiled at the Oregon Graduate Institute by extracting numbers (zip codes, dates, street numbers, etc.) from various other speech corpora, all of which had been recorded over the telephone (including both analogue and digital telephone lines). The data set used for training and cross validation consists of 3590 utterances (3233 for training, 357 for cross validation), corresponding to approximately two hours of speech. The test set comprises 1206 utterances (40 minutes). All utterances in these sets were manually transcribed at the phone level. The vocabulary consists of the 32 words shown in Table A.1 in the Appendix.

In addition to the original test set, four modified versions of the test set were created[1] by artificially adding noise or reverberation to the speech signal. The reverberant test set was produced by digitally convolving the signal with an impulse response function recorded in an echoic room with a reverberation time of 0.5s. For the noise test sets, pink noise from the Noisex database was added to the speech signal at various signal-to-noise (SNR) ratios: 0, 10, 20, and 30 dB.

3.1.2 Recognition System: The Hybrid Paradigm

All recognizers used for the experiments reported in this chapter are hybrid ANN/HMM systems. The hybrid approach to speech recognition combines artificial neural networks (ANNs) for the estimation of local subword unit probabilities with HMMs, which perform the task of temporally aligning the speech signal with a sequence of acoustic models. More precisely, this involves the recursive computation of the likelihood of the sequence of feature vectors $X = \mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_T}$ given the model λ, where λ is an HMM with N states $q_1, q_2, ..., q_N$:

$$P(X|\lambda) = \sum_{i=1}^{N} \prod_{t=1}^{T} a_{i,i-1} P(\mathbf{x_t}|q_i(t)) \tag{3.1}$$

Thus, the global probability of an observation sequence $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_T}$ given a HMM λ is defined as the product of all transition and observation probabilities over time points 1 through T, summed over all possible state sequences. Under the most widespread approach to ASR, the Gaussian mixture approach, the local emission probabilities $p(\mathbf{x_t}|q_i(t))$, with $1 \leq t \leq T$ and $i = 1 \leq i \leq N$ are computed by a weighted sum of Gaussian probability density functions (pdfs), so-called "mixture components":

$$p(\mathbf{x_t}|q_i(t)) = \sum_{m=1}^{M} c_{mi} \mathcal{N}(\mathbf{x_t}; \mu_{\mathbf{mi}}, \Sigma_{mi}) \tag{3.2}$$

where μ_{mi} and Σ_{mi} are the mean and covariance, respectively, of the $m'th$ mixture component at state i and c_{mi} is the mixture weight for that component. Each mixture component has the form of a Gaussian or Normal distribution:

$$\mathcal{N}(\mathbf{x}; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)' \Sigma^{-1} (\mathbf{x}-\mu)} \tag{3.3}$$

where μ is the mean vector and Σ the covariance matrix of the distribution and n is the dimensionality of the feature space.

In hybrid systems, ANNs are used to estimate the local emission probabilities of HMM states. The different types of ANNs which have been used for this purpose include Multi-Layer-Perceptrons (MLPs) [91], Radial-Basis-Function (RBF) networks [112], recurrent neural networks (RNNs)[108], and time-delay neural networks (TDNNs) [47]. All of these networks are

[1] We are grateful to Brian Kingsbury, ICSI, for supplying these test sets.

general function approximators which are trained to minimize a distance function relating the network outputs to a set of pre-specified target values. This error function is usually either the mean-squared error (MSE) or the relative entropy function. The MSE between the network's output x and the target y is defined as

$$D(x,y) = E[(x-y)^2] \tag{3.4}$$

where the expectation is taken over the set of training samples. The relative entropy between the distributions over X and Y, is defined as

$$D(X,Y) = E[\sum_k [p(y_k) ln \frac{p(y_k)}{p(x_k))}]] \tag{3.5}$$

The following reasons have been put forward (e.g. [91]) in favor of the hybrid approach and against the Gaussian mixture approach:

- **statistical modeling assumptions**
  It has been argued that ANNs permit an unconstrained approximation of the function underlying the distribution of the training data whereas Gaussian mixture classifiers are based on the assumption that the input data is normally distributed. This is only true to some extent – in principle, ANNs can approximate any objective function [62] but only under the conditions that a sufficient number of free parameters (hidden units, weights) are used and that sufficient training data are available to train these parameters. Moreover, any distribution can in principle be approximated by the weighted sum of a sufficiently large number of Gaussian mixture components. We will return to this point shortly.

- **incorporation of context**
  Another argument which has been adduced to support the hybrid approach is the possibility of including phonetic context in the estimation of local emission probabilities. The input to an ANN may consist either of the feature vector at time t only, or it may include the neighboring feature vectors at times $t-1, t-2, ..., t-n$ and $t+1, t+2, ..., t+n$ as well. In principle, context can also be included when the estimator is a Gaussian mixture classifier. Instead of estimating means and covariances for feature vectors of dimension, d, it is then necessary to estimate these parameters for $(n+1)d$-dimensional vectors which are formed by concatenating the original vector and n context vectors. Which approach is to be preferred depends on the particular application. Consider a fully connected three-layer MLP with 100 hidden units and 50 output units. Given a 20-dimensional feature vector and a context window of three frames, the MLP needs to find the values for (60$\times$100) + (100$\times$50) = 11,000 weights. The number of free parameters for a single-component full-covariance Gaussian mixture system with the same number of output classes would

be (1830 + 60)×50 = 94,500. ($\sum_{i=1}^{d} i$ parameters for the (triangular) covariance matrix, 60 for the mean vector, 50 classes). If, however, diagonal covariances are used, this number is reduced to (60 + 60)×50 = 6,000. The suitability of each approach thus depends on how much modeling effort is needed for the task at hand. MLPs may be the classifier of choice if the data has a highly non-Gaussian distribution which would require a large number of mixture components, and if the the use of diagonal covariance matrices in the individual mixture components would result in a sharp decline in accuracy. However, in other cases where the data distribution closely resembles a Normal distribution which could adequately be modeled by a small number of mixture components, or even a single Gaussian, the Gaussian mixture approach might be preferable.

- **discriminative training**

  The third important argument is that certain ANNs, such as MLPs, are directly trained to discriminate between the output classes rather than to most closely match the data distribution. As several researchers (e.g. [122, 103]) have shown, the output activations of MLPs approximate Bayesian *a posteriori* probabilities in the mean squared error sense, under the conditions that (a) the mean-squared error or relative entropy error function is used during training, and (b) a sufficiently large number of hidden parameters is used to be able to correctly approximate the desired result. In order to distinguish between k classes $\omega_1, ...\omega_k$, we need k discriminant functions $d_k(x)$. These are defined by the k-dimensional vector of *a posteriori* probabilities for the set of classes Ω, $\mathbf{P}(\mathbf{\Omega}|\mathbf{x})$. Since the parameters of the MLP are trained such that the output most closely approximates this probability vector, all discriminant functions (and therefore all class boundaries) are optimized jointly during training. In the Gaussian mixture system, on the other hand, each model is usually trained individually to maximize the likelihood of the training data given the model, without considering the optimal class boundaries.[2]

At a given time frame, t, an ANN estimates the probability $P(q_i(t)|\mathbf{x_t})$ of the HMM state q_i given the observation vector $\mathbf{x_t}$. Since it is the likelihood $p(\mathbf{x_t}|q_i(t))$ which is required by Equation 3.1, the posterior probabilities output by the ANN are converted to scaled likelihoods by dividing them by the *a priori* class probabilities $P(q_i)$.[3]

The ANNs used in our experiments are three-layered MLPs (one input, one output and one

[2] More recently, however, various discriminative training criteria have been investigated in the context of Gaussian mixture HMM systems, e.g. [10, 22, 109].

[3] In the hybrid approach considered here, each individual HMM state corresponds to an output class, i.e. a subword unit.

hidden layer). The activation function of the hidden layer is the logistic function

$$f(x) = \frac{1}{1 + exp(-ax)} \tag{3.6}$$

where a is a constant controlling the slope of the function. The final output activation function is the softmax function:

$$f(x_i) = \frac{exp(x_i)}{\sum_{n=1}^{K} exp(x_n)} \tag{3.7}$$

where K is the number of units in the output layer. The MLPs are trained by online stochastic gradient descent using simulated annealing of the learning rate to control the amount of weight change at each epoch. Initially, the input data is normalized by dividing each training vector by the sample mean and subtracting the sample variance. During training, patterns are presented to the nets in a randomized fashion. It is possible to include not only the current frame but also n context frames on each side, such that training samples actually consist of windows of $2n + 1$ frames. Windowing respects edge conditions, i.e. it does not span utterance boundaries.

Two different types of training schemes are possible with this architecture: simple and embedded training. Simple training consists of training an MLP until the training algorithm converges and then performing recognition. During embedded training, the subword net obtained in a given training phase is used to realign the transcriptions with the training data. This produces a new set of labels which is then used to train a new ANN in the subsequent training phase.

3.1.3 Acoustic Baseline Systems

Two different acoustic recognition systems were used as reference baselines, corresponding to the clean vs. deteriorated test conditions described above. The baseline system for clean speech was developed by Nikki Mirghafori, those for reverberant and noisy speech were developed by Brian Kingsbury[4].

The recognizer for clean speech uses log-RASTA-PLP preprocessing [60]. This preprocessing method first applies a windowed FFT to the signal and computes the power spectrum of each FFT component. The power spectrum is then convolved with a Bark-scale trapezoidal filter bank to approximate critical-band frequency resolution, similar to the mel-scale warping in the case of MFCC preprocessing. After logarithmic compression a bandpass filter is applied which passes the frequency modulations between 1-12 Hz. The output is then exponentially expanded and equal-loudness weighting and cube-root compression are applied in order to model perceptual loudness. Finally, linear prediction coefficients are computed by an autoregressive all-pole model and the cepstrum is taken.

[4] both at International Computer Science Institute (ICSI), Berkeley, USA

In addition to the eight basic log-RASTA-PLP coefficients, computed every 10 ms with a window of 25 ms, the clean speech baseline system additionally uses their first derivatives. The number of hidden units (HUs) in the phone MLP is 400, the size of the context window is nine frames, corresponding to approximately 105 ms. The system was trained using the embedded training procedure.

Similar to MFCC processing, log-RASTA-PLP processing is based on the log-spectrum and is therefore likely to be susceptible to additive noise. For this reason, log-RASTA-PLP was not used for the deteriorated test conditions. Instead, modulation spectrogram (MODSPEC) processing was employed, which has been developed specifically for reverberant and noisy speech and which has demonstrated superior performance under these conditions [55].

The modulation spectrogram is based on an eighteen-channel critical-band FIR filter bank, followed by the computation of amplitude envelopes in each channel. The amplitude signals are downsampled and normalized at the utterance level. Subsequently, a band pass filter is applied which simulates the characteristics of a FFT with a 250-ms Kaiser window. Finally, cube-root compression is applied. The characteristic properties of MODSPEC preprocessing are the suppression of fine phonetic details such as onsets and transitions, and the emphasis of the gross distribution of energy across time and frequency. MODSPEC enhances modulations between 0 and 8 Hz, with a peak at 4 Hz, corresponding roughly to the syllabic rate of speech.

The baseline system for the noise/reverberation test conditions uses 15 modulation spectrogram features. The hidden layer of the phone MLP consists of 560 HUs; the size of the context window is nine frames. The system was trained by embedded training.

All systems used for the experiments in this chapter employ the same recognition lexicon[5], which contains both canonical pronunciations and pronunciation variants, yielding a total of approximately 200 different pronunciation forms. Lexical search is carried out by a one-best Viterbi decoder. A back-off bigram is used as a language model.

3.2 Articulatory Feature Based Systems

The first step in the development of an articulatory-feature based recognition system is the heuristic selection of a suitable set of features. Many feature systems have been proposed in the past, based on theories of speech production, acoustic phonetics, or phonology. We will not discuss the various advantages and disadvantages of these systems here but we will merely delineate the general criteria that any acoustic-phonetic or pseudo-articulatory feature system for ASR

[5] developed by Dan Gildea at ICSI

should fulfill.[6] Since these kinds of features typically serve as an intermediate representation between the acoustics and the lexicon, they should minimally meet the following criteria:

- **acoustic stability**
  The acoustic stability criterion requires that features be extractable from an acoustic representation of the speech signal. This limits the set of possible features to those which can reasonably be expected to have some acoustic correlate and excludes purely functional features.

- **lexical stability**
  Furthermore, features should bear a constant relation to the recognition lexicon. In particular, they should be able to distinguish between all of the higher-level linguistic units to which they are mapped in later stages of the recognition process.

- **economy**
  In order to keep developmental and computational efforts low, the initial set of features should be as small and compact as possible.

In general, any feature set which is chosen heuristically is unlikely to be optimal for the higher-level classification task (e.g. phone classification), since it does not take into account the properties of the corpus, such as the relative frequencies of features and classes, cross-feature redundancies, the structure of the recognition lexicon, etc. Therefore, a data-driven optimization stage is usually required.

The initial feature set chosen for the experiments reported here is not based on a specific phonetic or phonological theory. However, it does reflect the structure underlying human speech production by taking into account the relative independence between different dimensions of articulation, as described above in Chapter 2.

We use five feature groups (Table 3.1) describing *voicing* (vibration of the vocal folds), the *manner* of articulation, the *place* of articulation (the location of the constriction in the vocal tract during consonant production or the tongue height during vowel production), the position of the tongue on the *front-back* axis, and lip *rounding*. The feature values have in most cases a straightforward articulatory explanation, with the possible exception of *retroflex*, which describes a manner rather than a place of articulation. We have nevertheless included it in the *place* category in order to distinguish /r/ from the coronal consonants /n,s,t,d,S,Z/. Each feature group includes “silence” in addition to the other feature values. “Nil” values are assigned to those segments for which this feature is not relevant.

[6] See e.g. [25] for a comparison of feature systems in phonetics and phonology.

Features	Values
voicing	voiced, voiceless, silence
manner	vowel, nasal, lateral, approximant, fricative, silence
place	dental, coronal, labial, retroflex, velar, glottal, high, mid, low, silence
front-back	front, back, nil, silence
rounding	+round, -round, nil, silence

Table 3.1: Articulatory feature system for Numbers95.

In the present system, features are mapped to context-independent phones. Therefore they were selected in such a way as to distinguish among most phones in the phone set. However, in order to keep the set of features minimal, certain distinctions were neglected which would require the inclusion of additional feature groups and which can easily be resolved by higher-level recognition constraints. The phone set used was the ICSI phone set (Table A.4 given in the Appendix), which consists of 56 phones. The articulatory features in Table 3.1 are able to distinguish between most of these, with the exception of the syllabic vs. non-syllabic sonorants /l/-/el/, /m/-/em/, /n/-/en/, and /r/-/er/. These are mainly distinguished by durational as opposed to articulatory characteristics. Furthermore, certain vowel distinctions (/iy/-/ih/, /uw/-/uh/, /aa/-/ao/) were not preserved in order to limit the set of features as far as possible. The fact that some phonemes are assigned identical feature representations should result in those phonemes receiving similar classification scores; the conflicting choice should in principle be resolved by higher-level lexical search.

3.2.1 Feature Classification

As mentioned in the previous chapter, a feature classification rather than a feature extraction approach is pursued in this thesis. The phone-level transcriptions of the Numbers95 training set were converted into feature transcriptions according to the phone-feature conversion table (Table A.3) given in the Appendix. The resulting feature transcriptions and the parameterized speech signals[7] constituted the training material for a set of articulatory feature classifiers.

Based on the following considerations, MLPs were chosen as articulatory feature classifiers.

- It was explained above that the relation between acoustic and articulatory parameters is highly non-linear. This calls for a classifier which is able to non-linearly map the acoustic input space into an articulatory parameter space. This is generally true of MLPs: the

[7] The parameterization was identical to that used in the acoustic baseline systems.

sigmoidal output functions of both the hidden layer and the output layer are capable of performing a non-linear mapping. Moreover, the non-linear, quantal nature of the acoustic-articulatory relation most probably results in a non-Gaussian distribution of the feature vectors for a given articulatory class. It therefore seems best to impose as few constraints as possible on the statistical estimation of these distributions or of the posterior probabilities for these classes. MLPs perform the approximation to the objective function by the following steps:

The first layer of a MLP computes a set of (non-linear) basis functions which are linearly combined in the second layer, or several subsequent layers. At the output of the final layer, the output activation values are non-linearly mapped to the range [0,1] by a sigmoidal activation function, such as the logistic function (Equation 3.6) or the softmax function (Equation 3.7).

Each of the components of the vector of hidden variables **h** embodies a basis function ϕ_h applied to the input vector **x** which has the form

$$\phi_h(\mathbf{x}) = \sigma(\mathbf{w_h}'\mathbf{x}) \tag{3.8}$$

where **w** is the appropriate vector of weights connected to the hidden unit h and σ is the sigmoidal activation function of the hidden layer. Since the basis functions depend on the adjustable vector **w**, they are not predetermined but may change during training. This provides a flexible framework for learning discriminant functions in a feature space which may exhibit highly irregular patterns.

As we mentioned above, little explicit knowledge is available about the acoustic-articulatory relation for each class and each possible context. MLPs can act as non-linear feature detectors which gradually focus on the salient acoustic input features during training. To the extent that the result of the training process (i.e. the weight matrices) is interpretable, they can thus function as exploratory tools when a new set of features, such as articulatory features, is first investigated. Various data mining methods (see e.g. [28]) may be applied to the trained MLP in order to extract explicit symbolic knowledge about the relation between acoustics and articulatory features.

- We noted before that the inversion problem can be simplified by taking into account temporal context rather than considering each frame in isolation. This involves computing the probability for (or the distribution of) several feature vectors at a time and thus requires enlarging the number of input parameters. Given our assumptions about the non-Gaussian distribution of acoustic feature vectors for most articulatory classes, it is highly probable that several mixture components and full covariance matrices would be required in order

Network	# HUs	# context frames
voicing	50	9
manner	100	5
place	100	9
front-back	100	5
rounding	100	5

Table 3.2: Number of hidden units and context frames for different feature networks.

to achieve a reasonable level of accuracy. It is more likely that MLPs will require fewer parameters in this situation since only the first layer of the MLP is affected by the increased number of input parameters.

- Some of the articulatory features we have chosen have a gradual nature and can most easily be classified in relation to other features. Examples of these features are vowel features such as *high, mid, low*, which have relational rather than absolute definitions. Their corresponding class regions in feature space will most probably overlap – their class boundaries are bound to be fuzzier than those separating, e.g. *nasal* from *plosive*. To alleviate this problem, a classifier should be used which is trained discriminatively.

Preliminary experiments comparing MLPs and Gaussian mixture HMMs as classifiers revealed a slightly superior performance of the MLPs. The same result was obtained independently by King et al. [68] for distinctive feature recognition on the TIMIT database.

For each for the five different feature groups in Table 3.1 a separate MLP is trained, yielding a set of five parallel MLPs. Each of the networks receives the same acoustic input but is trained using a different set of labels. Thus, each MLP has the possibility of focusing on those aspects of the acoustic input space which provides the largest amount of information about its articulatory output classes.

The feature networks are trained in a single training pass as opposed to an embedded training procedure since it was found that embedded training did not yield any benefit over a single training pass. The number of hidden units and the number of context frames (Table 3.2) in each network were determined empirically with the objective to maximize classification accuracy while minimizing the number of parameters.

Figures 3.1 to 3.5 show the frame-level accuracy of each network in relation to the number of context frames. These data were obtained on the clean test set. Based on these values, the optimal number of context frames was determined for each network. Before discussing them, a caveat should be mentioned with respect to evaluating the feature classifiers. Similar to feature training,

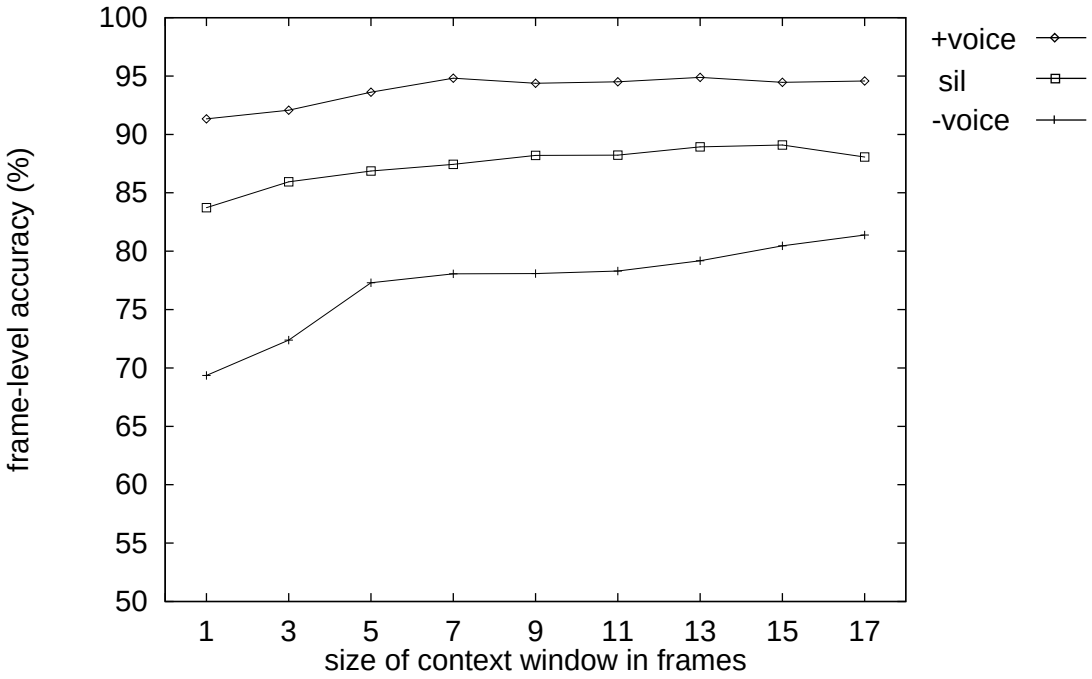

Figure 3.1: Accuracy rates for voicing features.

feature evaluation is based on phone transcriptions converted to feature labels. These labels do not reflect the actual feature boundaries but rather the hypothesized boundaries derived from the phone label boundaries. Thus, feature boundaries necessarily coincide with phone boundaries, which of course need not be the case in reality. It is possible that the manner network, for instance, classifies a strongly nasalized vowel as nasal; however, this would be considered wrong if the phone-level transcription contained a vowel at that position in the transcription. We believe that the accuracy rates obtained using this idealized data give a good approximate impression of the performance of the feature networks; however, it should be borne in mind that the "true" accuracy rates may deviate from these.

The data shown in Figures 3.1 to 3.5 reveal the relative context-dependence of individual features. In general, feature accuracy rates increase with the number of frames in the context window; however, not all features benefit from a larger context to the same degree. Different features exhibit peak accuracy rates at different context sizes.

The following observations are of importance:

- Most of the information about articulatory features provided by the acoustic input seems to be contained within a window of five frames, beyond which little increase in accuracy can be observed. This roughly corresponds to the average length of one to two phones. However, the fact that in some cases the accuracy rates do increase up to a context of 17 frames suggests that the acoustic correlates of the articulatory categories we use can be

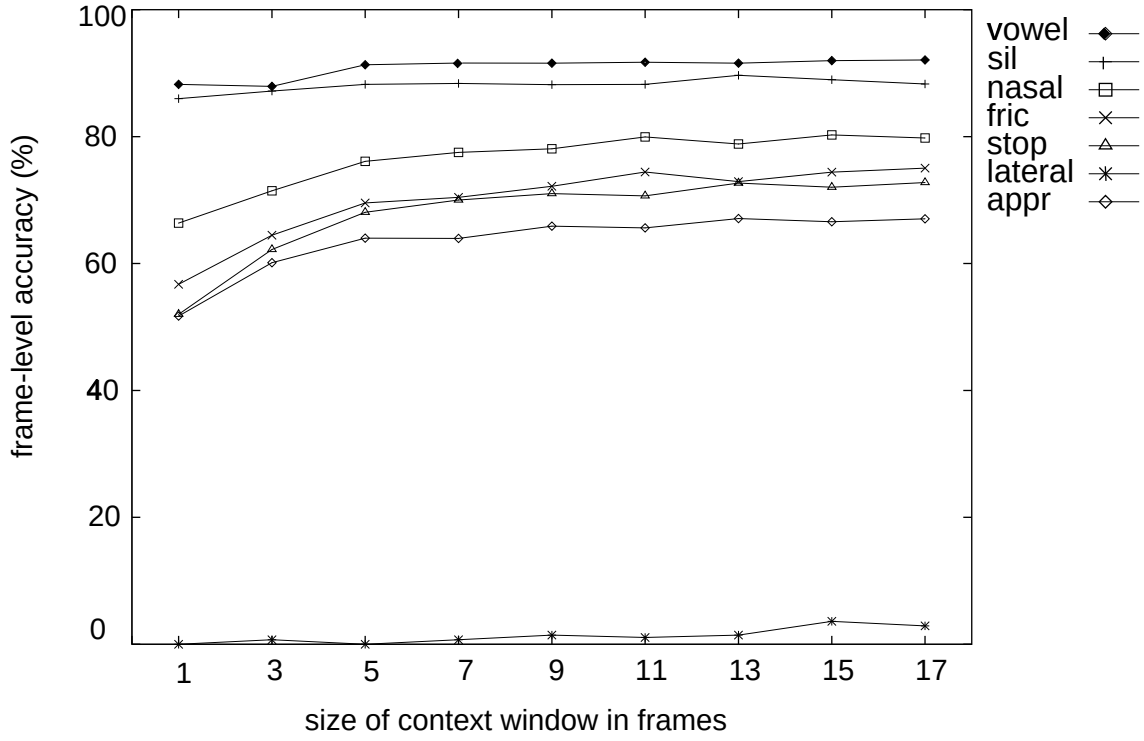

Figure 3.2: Accuracy rates for manner features.

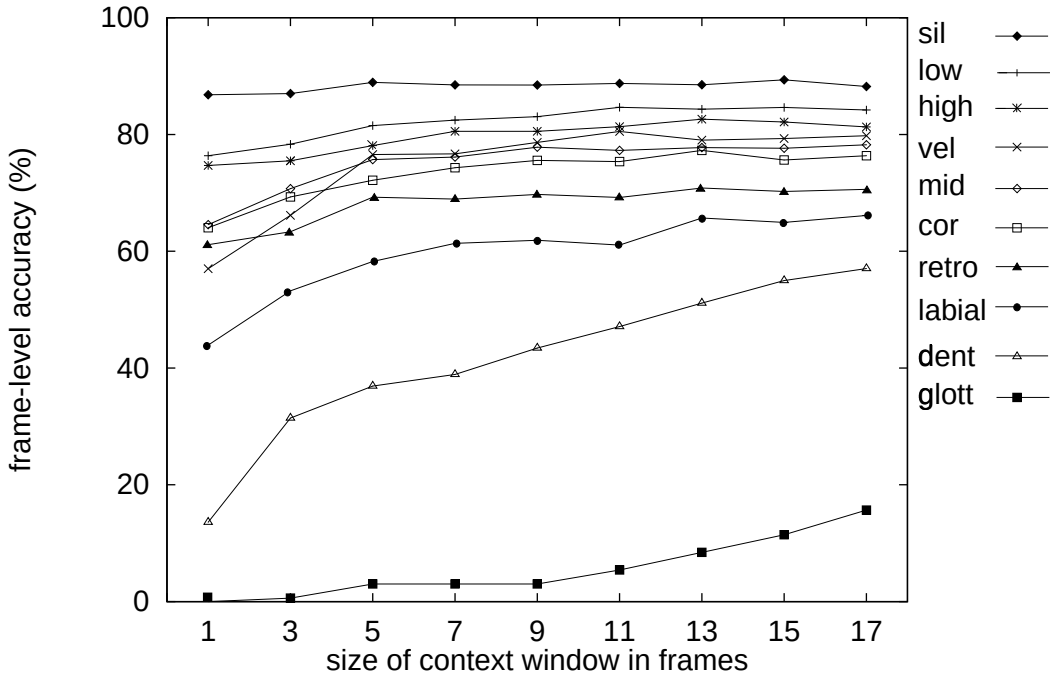

Figure 3.3: Accuracy rates for place features.

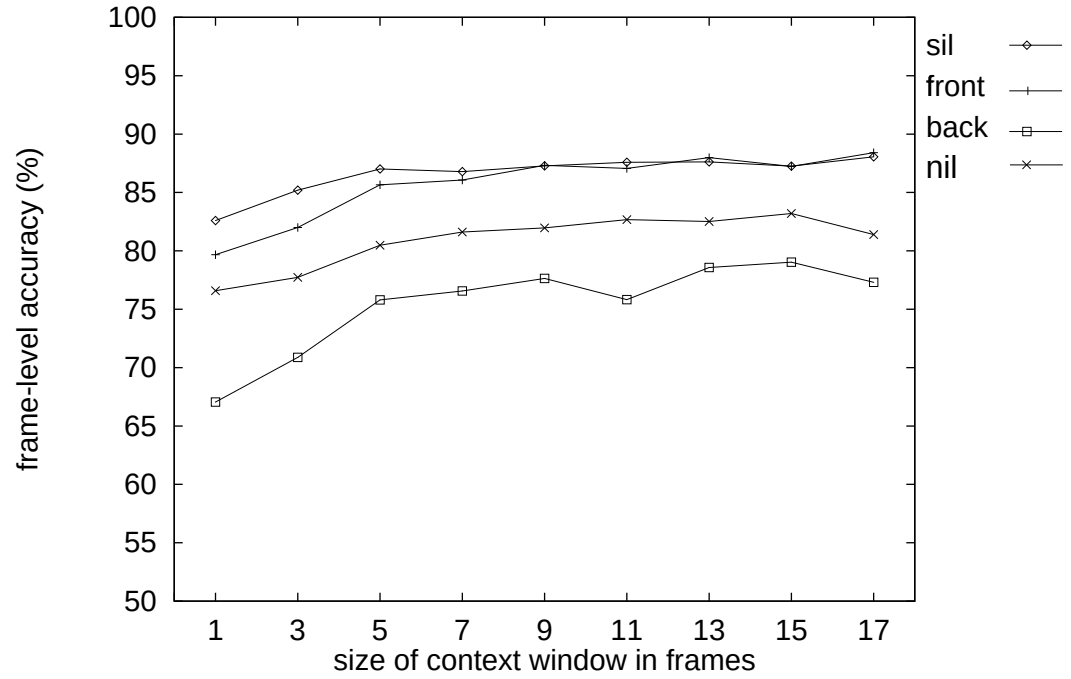

Figure 3.4: Accuracy rates for front-back features.

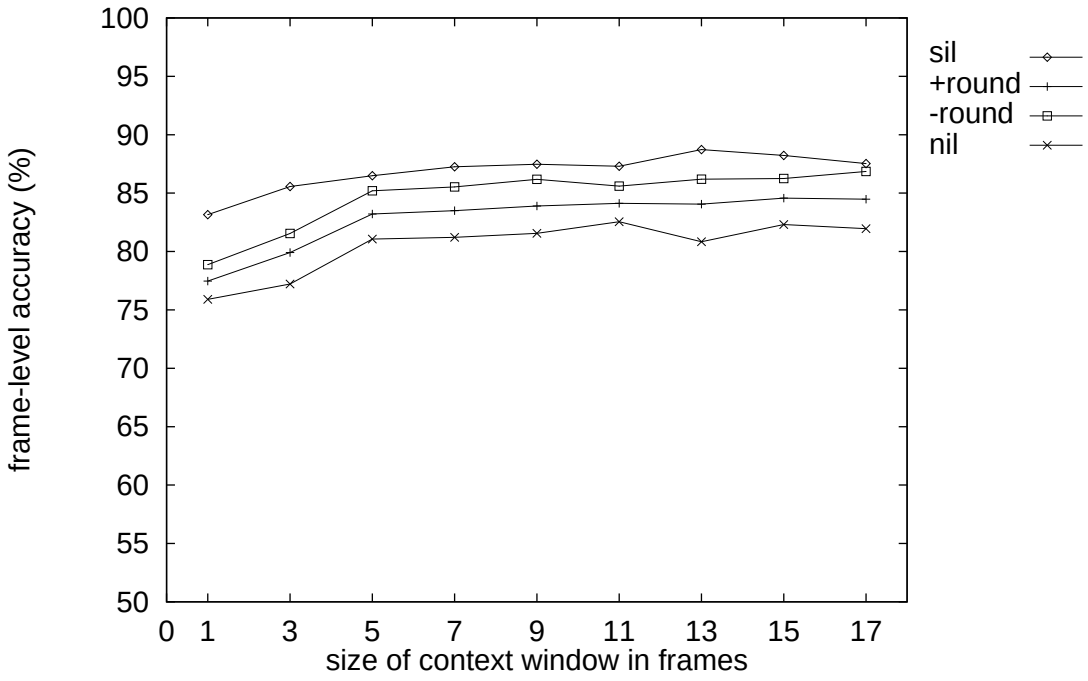

Figure 3.5: Accuracy rates for rounding features.

spread over time spans of the size of a syllable.

- The accuracy rates for a small set of features increase considerably even beyond five frames. This set includes primarily the manner feature *fricative* and the place features *labial*, *dental*, *glottal* and *velar*. There are several explanations for this: first, the data consists of telephone-bandwidth speech which excludes frequencies above 3400 Hz. Fricative consonants, however, are characterized by high-frequency energy above 3400 Hz; thus, most of the local information which may point to the presence of a fricative consonant has been filtered out, so that information in the surrounding context becomes more important for the identification of these sounds. Including the temporal context beyond five frames also has a positive effect on the accuracy rates of consonantal place features which are known from the phonetic literature to be heavily context-dependent, i.e. *velar* and *glottal* sounds, which are strongly influenced by surrounding vowels [30].

In order to ascertain whether the context effects can indeed be attributed to the wider context and not to a larger number of hidden parameters, feature recognition experiments were conducted where the number of parameters was increased by using more units in the hidden layer. However, the relative improvements were not as substantial as those induced by a larger temporal context.

One important conclusion to be drawn from this is that in order to reach optimal detection accuracy, articulatory feature recognition requires a smaller temporal context than phone recognition, which typically performs best with context-dependent phones integrating information over a sequence of at least three phones. This should prove beneficial for an articulatory-feature based acoustic modeling approach since the feature representation can be extracted from the acoustic signal using small, constrained classifiers with a small number of free parameters.

3.2.2 Feature-Phone Mapping

One of the goals of this pilot study is to find out which advantages, if any, the articulatory feature representation has over standard acoustic representations when no other changes are made to the recognition system. For this reason, the higher-level representation, i.e. the choice of subword units and the definition of the recognition lexicon, need to be the same as in the acoustic baseline systems. The subword units in these systems are context-independent phones – we therefore need to find a way of mapping articulatory feature probabilities to phone probabilities.

As described above, the articulatory feature probabilities can themselves be treated as data which is passed to a higher-level classifier. Under this approach, each of the lower-level classifiers partitions the input acoustic feature space in a different way and computes the posterior probabilities

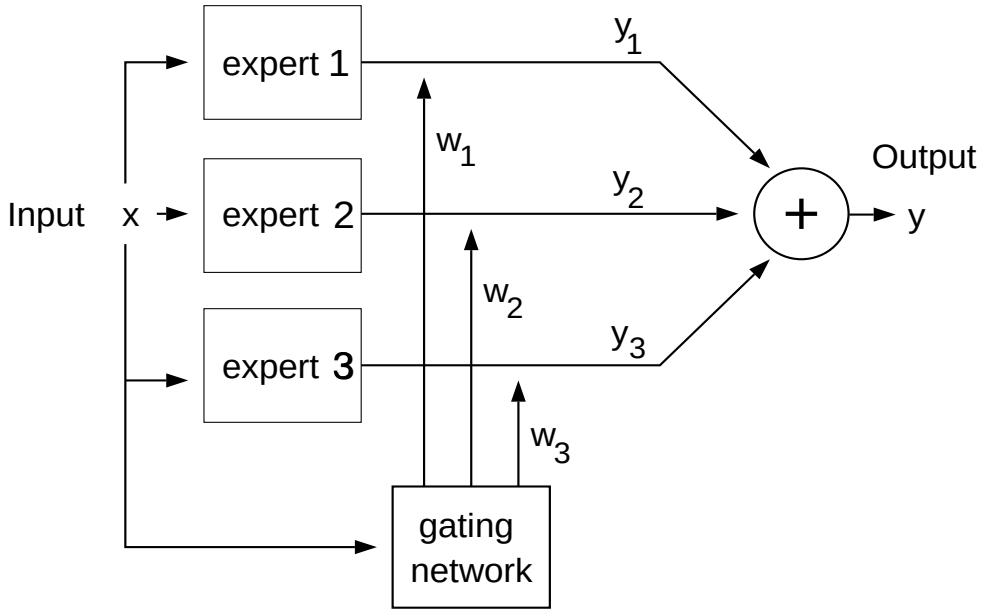

Figure 3.6: Mixture of experts.

for a different set of classes. The second classifier operates on the jointly presented probabilities for these classes and maps them to the final set of output classes.

This approach seems similar to the so-called "mixture of experts" (ME) architecture [63], where the output of a set of individually trained expert networks is combined by a *gating network* (see Figure 3.6). However, there are important differences with respect to the probabilistic interpretation of these architectures. In the ME model each expert network k, $k = 1, ..., K$ produces an output y_k. The gating network determines K suitable mixing coefficients $g_1, ..., g_K$ for the expert outputs, where the $k'th$ mixing coefficient may be interpreted as the probability that the $k'th$ expert network is able to predict the desired output y. The overall output y is then defined as

$$y = \sum_{k=1}^{K} g_k y_k \tag{3.9}$$

In the ME model, the gating network has access to the experts' outputs as well as to the input $\mathbf{x}$. It usually consists of a single layer of neurons with a single sigmoidal activation function. The individual expert networks, by contrast, are simple linear filters.

In our case the final output is not a mixture of the individual networks' outputs since every network in the first layer learns a different set of classes. Furthermore, the higher-level classifier does not have access to the feature input and does not estimate mixture weights but instead estimates the posterior probabilities of the phones given an articulatory feature vector.

Instead of using the continuous output from each network in the first layer, it may be sufficient

Combining Technique	**WER**	**INS**	**DEL**	**SUB**
winner-take-all	13.7%	3.6%	8.2%	2.0%
full distribution	8.8%	1.4%	5.4%	2.0%

Table 3.3: Word error rates obtained using different feature-phone mapping procedures: a winner-take-all method vs. using the full output distribution.

to use only the information about the winning output unit in each network (*winner-take-all* network). In this case the output probabilities from each network are mapped to either 0 or 1.

Both possibilities were investigated for the purpose of mapping articulatory feature probabilities to phone probabilities. The winner-take-all network used 400 hidden units and nine context frames, the merger MLP applied to the non-quantized output distribution had 380 hidden units and also used a 9-frame context window. Both networks employed the set of phones in Table A.4 in the Appendix as final output classes – however, many of these phones do not occur in the training or test set. The set of phone classes for which weights are actually updated is marked by $\star$ in Table A.4. All merger MLPs were trained using the embedded training procedure described earlier.

The results are shown in Table 3.3. Nor surprisingly, using the full output probability distribution yielded a significantly better result than the winner-take-all approach. The cascaded classifier approach using the full output distributions of the lower-level classifiers was therefore used for all further experiments reported in this chapter.

The use of a context window enables the higher-level classifier to learn, within certain limits, the statistical regularities of the temporal evolution and patterns of co-occurrence of articulatory feature probabilities. This may be regarded as a data-driven way of forming (abstract) generalizations about the shapes and overlaps of articulatory gesture trajectories described in Chapter 2. Thus, temporal overlaps between articulatory movements (as represented by high probabilities for the corresponding features) are not modeled explicitly but are acquired by the phone network in the course of the training process.

At this point we should review our initial hypothesis that a decompositional classification approach based on articulatory features leads to high feature recognition accuracy and therefore to a higher phone recognition accuracy than an acoustics-only phone classifier. Table 3.4 shows the frame-level feature recognition accuracy rates for different acoustic conditions, as well as the frame-level phone recognition rates obtained by both the acoustic (AC) phone classifier and the classifier which combines the articulatory features scores (AF). Each block of two rows shows the frame-level recognition accuracy (upper row) and, for the reverberant and noisy test cases,

Network	clean	reverb	noise 30 dB	noise 20 dB	noise 10 dB	noise 0 dB
voicing	89.12%	79.78%	81.62%	78.38%	73.49%	68.68%
Δ		-11%	-8%	-12%	-17%	-22%
manner	82.00%	67.10%	71.60%	67.27%	60.96%	54.01%
Δ		-18%	-13%	-18%	-26%	-34%
place	77.24%	60.96%	67.19%	63.38%	57.28%	48.72%
Δ		-21%	-13%	-18%	-26%	-37%
front-back	82.99%	71.02%	75.55%	72.58%	67.78%	61.08%
Δ		-14%	-9%	-13%	-18%	-27%
rounding	83.19%	70.89%	76.62%	73.58%	68.80%	62.34%
Δ		-15%	-8%	-12%	-17%	-25%
phone AC	77.05%	58.80%	62.70%	57.70%	49.33%	38.78%
Δ		-24%	-18%	-25%	-36%	-50%
phone AF	75.23%	64.80%	68.32%	64.05%	56.40%	46.20%
Δ		-14%	-9%	-15%	-25%	-39%

Table 3.4: Frame-level feature and phone recognition rates. AF = articulatory feature based classifier, AC = acoustic classifier.

the decline in accuracy relative to the clean test condition (lower row). We can see that the recognition accuracy differs among the different feature networks, which may be related, *inter alia*, to their different complexities in terms of the number of output classes. Voicing features (3 classes) are easiest to recognize whereas place features (10 classes) seem to present the most difficulties. The degree to which feature accuracy declines under adverse conditions also seems to be related to this property, with voicing features declining the least, followed by rounding, front-back, manner, and place features. The assumption that all individual feature networks should have a higher recognition accuracy than the acoustic phone classifier turns out to be correct for this particular classification task. The combination of the feature detectors leads to a higher accuracy in the phone classifier in reverberant and noisy speech, but not in clean speech. The reasons for this might be that the simple phone classifier already performs very well in clean speech and that the errors of the individual AF classifiers may be too correlated and thus prevent the higher-level articulatory classifier from making a more accurate decision than the acoustic classifier. Additional factors which contribute to the beneficial effect of the cascaded classification scheme in noise and reverberation might be the following:

- the use of context information at lower levels: in the AF model, not only the higher-level merging classifier but also the lower-level classifiers themselves make use of context information, which might have a particularly robust effect in highly noisy conditions.

- emphasis on gross spectral patterns rather than absolute frequency: the articulatory classifiers are trained to abstract away from absolute frequencies and to learn the overall relative properties of the time-frequency energy distribution instead.
- the additive effect of noise within the acoustic phone classifier: various disturbances of the spectrum may have a cumulative effect on the classification result of the acoustic phone classifier, whereas they may have more localized effects on the articulatory classifiers, which can then be weighted selectively by the higher-level classifier. This effect might even be more pronounced if the higher-level classifier were trained or adapted on noisy/reverberant speech as well.

In sum, our initial hypothesis about the benefits of the cascaded classification scheme has been by and large confirmed. Before presenting word recognition results obtained by the acoustic and articulatory systems, let us address the issue of optimizing the articulatory feature space.

3.2.3 Feature Optimization

The articulatory feature space has 28 dimensions – the AF system thus requires more parameters (network weights) than the corresponding acoustic baseline systems, which are based on 15-dimensional and 18-dimensional feature spaces, respectively. In order to ensure the comparability of the different systems, the articulatory feature space had to be reduced to a lower dimensionality. Furthermore, heuristically selected feature sets are rarely optimal and should generally be improved in a data-driven way. It was therefore necessary to find a way of reducing the dimensionality of the articulatory feature set while simultaneously improving the quality of the features with respect to phone classification. Two different methods were investigated, Principle Components Analysis (PCA) and an information-theoretic feature selection algorithm [74].

PCA performs a linear transformation

$$\mathbf{z} = \Phi(\mathbf{x} - \mu) \tag{3.10}$$

on the input feature space, where $\mathbf{x}$ is the n-dimensional input feature vector and $\mathbf{z}$ is the m-dimensional output feature vector, $m < n$. The $m \times n$ matrix Φ is a matrix whose rows consist of the eigenvectors of the sample covariance matrix of $\mathbf{x}_1, ..., \mathbf{x}_N$, and μ is the sample mean vector. This transformation has the effect of changing the original coordinate basis of the feature space to a different coordinate basis, with the result that the variances of the individual vector components are rendered maximally different (as measured by entropy). The original feature space can thus be projected onto a subspace defined by the m largest eigenvalues. The amount

# principal components	% variance covered	WER
5	85.9	11.6%
7	91.7	10.4%
9	95.3	9.6%
11	96.9	9.4%
13	97.8	9.8%
15	98.6	9.8%
17	99.2	9.8%
18	99.4	9.8%

Table 3.5: Word error rates obtained with a variable number of principal components.

of variance covered by the m principal components is the sum of the first m eigenvalues divided by the total sum of eigenvalues. PCA applied to the articulatory feature space yields an 18-dimensional feature space defined by the first 18 principal components, covering 99.4% of the variance of the original feature space. These were used to train the phone MLP using the same number of context frames and HUs and the same training procedure as before. Table 3.5 shows the word recognition results for different numbers of principal components.

The advantage of PCA is that it is purely feature-driven, i.e. no information about class labels or models is required. However, two drawbacks of this method became obvious: first, PCA interacted negatively with the embedded training procedure, which resulted in a slight increase in word error rate. When PCA was combined with a simple training procedure, identical word error rates were achieved with a smaller number of parameters. However, the combination of PCA and embedded training caused an increase in word error rate from 8.8% to 9.4%. Although this increase is statistically non-significant it would be desirable to employ a feature optimization method which does not counteract the beneficial effect of improving the match between the acoustic signal and the training transcription through embedded training. Furthermore, PCA involves an additional matrix multiplication for each feature vector. It would be more efficient to simply specify which features are required and which can be dropped from the feature representation before proceeding to the higher recognition levels. Finally, the feature space generated by PCA is not transparent in the sense that individual feature dimensions are no longer associated with an articulatory interpretation.

For these reasons another reduction method was investigated, viz. an information-theoretic feature selection algorithm [74]. This is a supervised algorithm which selects features both on the basis of their relations to the class set Ω and on the basis of the statistical dependencies among

the features themselves.

Let $\mathcal{F}$ be our initial feature set of size n, which we would like to project onto a feature set $\mathcal{G}$ of size m, $m < n$. For any assignment of feature values $\mathbf{f} = (f_1, f_2, ..., f_n)$ to the features $\mathcal{F} = (F_1, F_2, ..., F_N)$ there exists a conditional probability distribution of the class set $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ given the assignment $\mathcal{F} = \mathbf{f}$, $P(\Omega|\mathcal{F} = \mathbf{f})$. Similarly, for the reduced feature set $\mathcal{G}$, we have an assignment of feature values $\mathcal{G} = \mathbf{f_g}$ and a conditional probability distribution $P(\Omega|\mathcal{G} = \mathbf{f_g})$. The overall goal then is to select the set $\mathcal{G}$ is such a way that the difference between $P(\Omega|\mathcal{G} = \mathbf{f_g})$ and $P(\Omega|\mathcal{F} = \mathbf{f})$ is as small as possible. Selection is carried out by *backward elimination*, i.e. by iteratively discarding one feature F_i from the original feature set $\mathcal{F}$ until the desired number of features, m, has been reached. At each iteration we obtain a smaller feature set $\mathcal{G} = \mathcal{F} - \{F_i\}$; at the end of each iteration, the set $\mathcal{G}$ replaces the current set $\mathcal{F}$. The feature F_i which is eliminated is chosen such that the distance between $P(\Omega|\mathcal{G} = \mathbf{f_g})$ and $P(\Omega|\mathcal{F} = \mathbf{f})$ remains as small as possible. The distance between these conditional probability distributions is measured by relative entropy (or Kullback-Leibler (KL) distance). Let $\mu(\omega_j)$ stand for $P(\omega_j|\mathcal{F} = \mathbf{f})$ and $\sigma(\omega_j)$ for $P(\omega_j|\mathcal{G} = \mathbf{f_g})$. The KL-distance between μ and σ, $D(\mu||\sigma)$ is then defined as

$$D(\mu||\sigma) = \sum_{j=1}^{k} \mu(\omega_j) \frac{\mu(\omega_j)}{\sigma(\omega_j)} \tag{3.11}$$

Note that this is not a symmetric distance measure, i.e. $D(\mu||\sigma) \neq D(\sigma||\mu)$. The distribution μ is in this case considered the "correct" distribution and the deviation of σ with respect to μ is to be minimized.

If the relative frequencies of different feature value assignments are taken into account, the distance measure is refined to

$$\Delta_G = \sum_{f} P(\mathcal{F} = \mathbf{f}) D(P(\Omega|\mathcal{F} = \mathbf{f})||P(\Omega|\mathcal{G} = \mathbf{f_g})) \tag{3.12}$$

where $\mathbf{f}$ is a particular assignment of features to the set $\mathcal{F}$ and $\mathbf{f_g}$ is a feature assignment to set $\mathcal{G}$. At each iteration of backward feature elimination, Δ_G should increase as little as possible.

It is easy to see that the computational effort in computing Δ_G increases exponentially with the number of features in $\mathcal{F}$: as the number of features increases, the number of possible joint feature assignments is multiplied exponentially. This problem can be alleviated by considering only the most important statistical dependencies among the features. Both features and classes can be considered random variables. In a domain which is characterized by a sizable set of random variables it is often the case that some variables are conditionally dependent whereas others are conditionally independent of each other given a third set of variables. Koller and Sahami ([74]:283) define conditional independence as follows:

> *Two sets of variables [***A** *and* **B** *- K.K.] are said to be* conditionally independent *given some set of variables* **X** *if, for any assignment of values* **a**, **b**, *and* **x** *to the variables* **A**, **B**, *and* **X** *respectively,* $Pr(\mathbf{A} = \mathbf{a}|\mathbf{X} = \mathbf{x}, \mathbf{B} = \mathbf{b}) = Pr(\mathbf{A} = \mathbf{a}|\mathbf{X} = \mathbf{x})$*. That is,* **B** *gives us no information about* **A** *beyond what is already in* **X**.

Presumably, the features whose elimination increases Δ_G as little as possible are those features which provide little information about Ω in addition to the information already supplied by the features contained in $\mathcal{G}$. Therefore, at each iteration of the feature selection algorithm the task is to find that feature F_i which is *conditionally independent* of Ω given $\mathcal{G} = \mathcal{F} - \{F_i\}$. This can be done by finding the *Markov Blanket* of the feature F_i which is being considered for elimination. The concept of a Markov Blanket stems from the theory of probabilistic reasoning [98] and denotes a subset $\mathcal{S}$ of the entire set of variables $\mathcal{U}$ in a given domain such that given $\mathcal{S}$, the variable α is conditionally independent of all variables in the domain other than α and S, i.e. $\mathcal{U} - \mathcal{S} - \alpha$. In the present context the definition of a Markov Blanket is as follows:

Definition 1 *Let* $\mathcal{M}$ *be some set of features which does not contain* F_i*. We say that* $\mathcal{M}$ *is a Markov blanket for* F_i *if* F_i *is conditionally independent of* $(\mathcal{F} \cup \Omega) - \mathcal{M} - \{F_i\}$ *given* $\mathcal{M}$*.([74]:283, notational symbols adapted - K.K.)*

The feature elimination strategy can then be reduced to considering candidate sets of features which might constitute a Markov blanket M_i for a given feature F_i and to eliminate that feature for which M_i is closest to being a Markov blanket. The Markov Blanket property is evaluated using the expected KL-distance between the conditional probability distribution of the classes Ω given both M_i and F_i and the distribution of the classes given only M_i. The overall distance δ_G is a sum over all possible assignment of feature values to M_i and F_i, weighted by the prior probabilities of the feature value assignments:

$$\delta_G(F_i|M_i) = \sum_{\mathbf{f_{M_i}}, f_i} P(M_i = \mathbf{f_{M_i}}, F_i = \mathbf{f_i}) D(P(\Omega|M_i = \mathbf{f_{M_i}}, F_i = \mathbf{f_i})||P(\Omega|M_i = \mathbf{f_{M_i}})) \tag{3.13}$$

where $\mathbf{f_{M_i}}$ is a particular assignment of feature values to M_i. The closer M_i is to being a Markov Blanket for F_i the closer this quantity is to zero. The final point which needs to be explained is how the candidate feature sets for a Markov Blanket are selected in the first place. If a feature set is indeed a Markov Blanket for another feature, all these features can be expected to be strongly correlated. Thus, the correlations between features in $\mathcal{F}$ can be computed and, for any given feature, the k features with which it is most strongly correlated constitute a candidate set for a Markov Blanket. In [74], correlation is measured by the pair-wise relative entropy γ_{ij} between

the conditional distributions over the classes given two features F_i or F_j, respectively:

$$\gamma_{ij} = \sum_{f_i,f_j} D(P(\Omega|F_i = f_i, F_j = f_j)||P(\Omega|F_j = f_j)) \tag{3.14}$$

This leads to the following algorithm for obtain a subset of m features from the set $\mathcal{F}$:

Information-Theoretic Feature Selection

1: compute γ_{ij} for all $F_i, F_j \in \mathcal{F}, i \neq j$
2: set $\mathcal{G}$ = $\mathcal{F}$
3: **while** the dimension of $\mathcal{G}$ is larger than m **do**
4: **for** all $F_i \in \mathcal{G}$ **do**
5: set M_i to the set of k features in $\mathcal{G} - \{F_i\}$ with the smallest γ_{ij}
6: compute $\delta_G(F_i|M_i)$
7: **end for**
8: find the feature F_i with minimal δ_G and set $\mathcal{G} = \mathcal{G} - \{F_i\}$
9: **end while**

This procedure has the effect of eliminating those features which are either irrelevant for the classification task at hand or whose information is already subsumed by the other features in the feature set.

In applying this algorithm to selecting a subset of articulatory features, the probabilities in the above equations were approximated by histograms. Both feature values and class probabilities were quantized into 10 equal bins covering the range [0,1] and the relative frequencies of feature values and/or class probabilities, which were used in approximation of true probabilities, were summed over all bins.

The application of this algorithm with the objective of eliminating 10 articulatory features yielded better results than PCA: the maximum increase in word error rate was 0.1%, for the final set of 18 features. Table 3.6 shows the features which were eliminated.

What is the interpretation of these features being eliminated? First, it should be noted that one feature in each feature group is redundant because its value can always be predicted from the sum of all other values in the feature group – remember that the outputs of the feature networks are posterior probabilities which sum to one. Thus, it is not surprising that all *silence* values are eliminated. As far as the features *dental* and *approximant* are concerned, it is very likely that they are not relevant for classifying the majority of phone classes. These features occur very

Feature group	features eliminated
voicing	+voice, -voice, silence
manner	approximant, silence
place	dental, silence
front-back	nil, silence
rounding	silence

Table 3.6: Features eliminated by information-theoretic feature selection.

infrequently and are only relevant for the phones /th,dh,r,w,y/, of which only /th,r/, and /w/ appear in the recognition lexicon. The information provided by the voicing features is presumably subsumed by other features in the set, such as the distribution of vocalic features like *+round*, *-round*, *front*, etc.

It turns out that one advantage of this feature selection method is the fact that it is now possible to eliminate one entire feature network, the *voicing* network, in advance, i.e. before generating the input data to the higher-level classifier. The features which were not eliminated can be used without any additional transformation. The PCA transformation, by contrast, requires all input dimensions to be generated before applying the reduction transformation in the form of an additional matrix multiplication. Moreover, the information-theoretic feature selection process preserves the phonetic interpretation of the individual feature vector component, which may be important for higher-level analyses.

3.3 Recognition Results and Error Analysis

For the purpose of word recognition we again tested the effect of a variable temporal context, as well as different numbers of hidden units. The results are shown in Tables 3.7 and 3.8. We observe that a wider context leads to a reduction in word error rate. Here, the beneficial effect of including temporal context does not drop off at 9 frames but continues up to 15 frames. The word error rate rises as the context is extended to 17 frames; however, this may be due to the growing number of parameters in relation to the limited amount of training data.

These data suggest that the temporal misalignment among different features in relation to phone-sized units is more pronounced than the temporal extension of the acoustic correlates of articulatory features. Misalignment effects seem to spread across temporal units of syllabic size.

Table 3.9 shows the word error rates obtained under the different acoustic test conditions. Statistically significant differences between the acoustic and articulatory systems are shown in boldface. As can be seen from the results, the performance of the acoustic baseline systems and the

# context frames	WER
1	10.1%
3	9.8%
5	9.4%
7	8.9%
9	8.9%
11	9.1%
13	8.7%
15	7.9%
17	8.8%

Table 3.7: Word error rate on clean speech in relation to the context size in the phone network.

# hidden units	WER
465	9.3%
535	9.3%
600	8.9%
660	8.9%

Table 3.8: Word error rate on Numbers95 clean test set in relation to the number of hidden units in the phone network.

System	WER	INS	SUB	DEL
AC clean	8.4%	2.0%	4.7%	1.7%
AF clean	8.9%	1.5%	5.4%	2.0%
AC reverberant	24.7%	1.7%	16.5%	6.4%
AF reverberant	23.7%	3.1%	16.0%	4.7%
AC, noise 30 dB	17.2%	2.4%	11.6%	3.3%
AF, noise 30 dB	17.4%	2.4%	11.6%	3.4%
AC, noise 20 dB	22.8%	3.3%	14.8%	4.8%
AF, noise 20 dB	21.7%	4.3%	13.9%	3.6%
AC, noise 10 dB	**32.7%**	5.1%	20.3%	7.3%
AF, noise 10 dB	**30.0%**	6.1%	18.3%	5.7%
AC, noise 0 dB	**50.2%**	8.3%	29.7%	12.2%
AF, noise 0 dB	**43.6%**	7.1%	26.3%	10.2%

Table 3.9: Word error rates, insertions, deletions, and substitutions of the acoustic (AC) and articulatory (AF) recognizers, for clean, reverberant and noisy speech.

articulatory system is fairly similar under clean and reverberant conditions. In noisy conditions, the acoustic system performs better at a high SNR (30 dB) but deteriorates as the SNR decreases. The difference between the word error rates at 0 dB, 50.8% for the acoustic system vs. 43.6% for the articulatory system, is highly significant.

Word error rates alone provide little information about the strength and weaknesses of a recognition system. Although the overall word error rates for the acoustic and the articulatory systems are comparable, the systems may exhibit different properties which are not revealed by this measure. It is therefore appropriate to additionally analyze further system outputs, such as the frame-level error rates, phone confusion matrices, word-level error patterns, etc. Quantitative and qualitative analyses of the performance of the individual systems and the differences between the acoustic and articulatory systems were carried out both at the frame level and at the word level. A frame-level analysis is identical to analyzing the performance of the MLP phone classifier, whereas a word-level analysis takes into account the decoding process. As mentioned above, all systems used the same lexicon, language model and decoder. Thus, differences in performance mainly derive from differences among the phone MLPs.

We computed the frame-level error rate (Table 3.10), as well as the average entropy of the phone output distribution for correctly and incorrectly classified frames, respectively (Table 3.11). The simple frame-level error percentage is obtained by

$$100 - \frac{\text{\# frames correct}}{\text{\# total frames}} * 100.0 \tag{3.15}$$

A sample is counted as correct when, at a given frame, the index of the output unit with the maximum activation value corresponds to the class label for that frame. This measure in effect maps the network activations to either 1 or 0 and only considers the hard decision resulting from this quantization. It might, however, be useful to take into account the continuous activation values in order to assess how confident the network is of its decision. This confidence can be measured by entropy. The entropy of a random variable Y taking on i different values is defined as

$$H(Y) = -\sum_{i=1}^{N} log(p_i)p_i \tag{3.16}$$

In our case, Y ranges over the output values of the MLP at a given frame. These output activations can be equated with posterior phone probabilities. A sharply peaked (low-entropy) activation distribution of, say, 0.1-0.8-0.1 (for a three-output unit network) indicates a greater certainty of decision than a “flatter”, high-entropy distribution of e.g. 0.4-0.6-0.4. Low entropy is preferable when the decision for a particular class is correct but is less desirable when the decision is wrong – ideally, we would like the classifier to be confident about a right decision and less confident about a wrong decision. Each of the frames in the test set can be classified

System	Frame Error Rate
AC, clean	22.95%
AF, clean	24.77%
AC, reverberant	35.40%
AF, reverberant	36.10%
AC, noise 30 dB	37.27%
AF, noise 30 dB	31.68%
AC, noise 20 dB	42.82%
AF, noise 20 dB	35.95%
AC, noise 10 dB	50.68%
AF, noise 10 dB	43.60%
AC, noise 0 dB	61.23%
AF, noise 0 dB	53.80%

Table 3.10: Frame error rates for acoustic (AC) and articulatory (AF) recognizers, for clean, reverberant, and noisy speech.

System	entropy "correct"	entropy "incorrect"	entropy ratio
AC, clean	0.49	2.78	0.18
AF, clean	0.22	1.32	0.16
AC, reverberant	0.50	1.00	0.50
AF, reverberant	0.36	1.55	0.23
AC, noise 30 dB	0.46	1.78	0.27
AF, noise 30 dB	0.37	1.67	0.22
AC, noise 20 dB	0.50	1.70	0.29
AF, noise 20 dB	0.41	1.68	0.24
AC, noise 10 dB	0.55	1.61	0.34
AF, noise 10 dB	0.46	1.61	0.28
AC, noise 0 dB	0.60	1.49	0.39
AF, noise 0 dB	0.51	1.51	0.34

Table 3.11: Average entropy values for correctly/incorrectly classified frames.

as either correct or incorrect based on the networks' class decisions. The average entropy can then be computed for each of these sets yielding $H(Y_c)$ (for the "correct" set) and $H(Y_i)$ for the "incorrect" set. The *entropy ratio*, $H(Y_c)/H(Y_i)$ defines a suitable additional measure of the quality of frame-level decisions – the smaller the entropy ratio, the more confident the system is about its correct decision and the less confident it is about its wrong decisions.

The acoustic system produces a lower frame-level error rate for clean speech. The AF system, by contrast, yields markedly lower error rates for reverberant and noisy speech. All of these differences are statistically significant and correspond to the differences in word error rate. The entropy values reveal that the AF system exhibits a lower entropy throughout, i.e. it is generally more certain of its decisions, for both correctly and incorrectly classified frames and under all test conditions. The entropy of the distributions produced by the acoustic systems, by contrast, is globally higher. The statistical significance of the differences in the average entropy values between the two systems was determined by a difference-of-means t-test. It was found that the differences were significant at the 0.0001 level. The entropy ratios also reveal a noticeable difference between the acoustic and articulatory systems: the articulatory system shows smaller entropy ratios across all test conditions, Moreover, the distance to the entropy ratios for the acoustic system increases as the signal-to-noise ratio drops. This suggests that the quality of the decisions of the articulatory systems is superior to that of the acoustic systems, particularly at higher noise levels.

As a quantitative measure of the differences between the acoustic and the articulatory systems, we computed the correlation between the networks outputs and the *ensemble variance*. The first of these is expressed as Pearson's product moment correlation coefficient, defined as

$$r = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_i (x_i - \mu_x)^2}\sqrt{\sum_i (y_i - \mu_y)^2}} \tag{3.17}$$

where x and y are the values of two random variables X and Y and μ_x and μ_y are the means of the distributions of these values, respectively. A correlation coefficient close to 1 indicates positive correlation, -1 indicates negative correlation and a correlation coefficient of 0 means that the variables are uncorrelated. In this case the variables X and Y range over the outputs of the two phone MLPs.

Ensemble variance is a measure which is commonly used in the machine learning community (e.g. [75]) in order to describe the amount of disagreement between different (neural network) classifiers which are part of a pool or *ensemble* of classifiers. Let us assume that the ensemble contains N different networks, all of which compute a function from an input x to an output $g(x)$. If the input to network i is denoted as x_i and output for network i is written as $g_i(x_i)$, the ensemble output $\overline{g}(x_1, ..., x_N)$ can be defined as

Test condition	**total variance**	**per-frame average**	**correlation coefficient**
clean	12119.64	0.05	.77
reverb	17241.44	0.07	.62
noise, 30 dB	15957.28	0.07	.63
noise, 20 dB	18442.53	0.08	.56
noise, 10 dB	22443.53	0.10	.47
noise, 0 dB	25120.75	0.11	.36

Table 3.12: Ensemble variance (total and per-frame average) and correlation for pairs of acoustic and articulatory networks under different acoustic test conditions.

$$\overline{g}(x_1, ...x_N) = \sum_{i=1}^{N} w_i g_i(x_i) \tag{3.18}$$

i.e. as the weighted sum of all individual outputs. The ambiguity (or variance) of the $i'th$ member of the ensemble is the mean squared error between the $i'th$ member's output and the ensemble output:

$$V_i(x_i) = E[(g_i(x_i) - \overline{g}(x_1, ..., x_N))^2] \tag{3.19}$$

The expectation of the variances is taken over a set of samples, in this case the samples in the test set. The ensemble ambiguity $\overline{V}$ is the weighted sum of the individual variances:

$$\overline{V}(x_1, ..., x_N) = \sum_{i=1}^{N} w_i V_i(x_i) \tag{3.20}$$

The ensemble ambiguity describes the weighted variance of the ensemble with respect to the weighted mean. The larger this quantity, the more the networks disagree. The ensemble ambiguity was computed for the six different pairs of acoustic and articulatory networks corresponding to the different acoustic test conditions. The individual network's variances were computed over all samples in the test set, i.e. at each sample and for each output unit, the weighted mean of the corresponding outputs in the two different MLPs and the deviation of each individual output from this mean were determined. Finally, the weighted sum of these variances was computed and averaged over all output units, to arrive at an overall scalar variance value, i.e.

$$\overline{V}(x_1, ..., x_N) = \frac{1}{D} \sum_{d=1}^{D} \sum_{i=1}^{N} w_{id} V_{id}(x_i) \tag{3.21}$$

where D is the dimensionality of the networks' output layers. Uniform weights (i.e. 0.5,0.5) were used throughout. The results are shown in Table 3.12.

Test condition	both correct	AC correct AF incorrect	AC incorrect AF correct	both incorrect	same errors	diff errors
clean	67.56	9.50	7.67	15.27	61.92	38.08
reverb	53.46	12.17	11.39	22.98	52.69	47.31
noise, 30 dB	54.54	8.19	13.78	23.49	57.89	42.11
noise, 20 dB	48.81	8.90	14.23	27.05	51.69	48.31
noise, 10 dB	39.88	9.45	16.60	34.06	42.38	57.62
noise, 0 dB	34.52	11.68	14.80	38.99	36.68	63.32

Table 3.13: Percentages of frame-level error distribution for various acoustic test conditions.

We can see that the correlation between the classifiers' outputs is fairly high in clean conditions but drops rapidly in reverberant and noisy conditions. Similarly, the ensemble variance increases under acoustically distorted conditions, suggesting that the acoustic and articulatory networks increasingly disagree on their classification task when the signal is corrupted by noise or reverberation.

As another way of quantifying the inter-classifier differences, the number of different vs. the number of identical errors was counted. Table 3.13 lists, for each test condition, the percentage of frames on which both the acoustic and the articulatory MLP agree, the percentage of cases where one MLP was correct and the other was incorrect, the percentage of simultaneous errors, and, within the latter category, the amount of different vs. identical errors. These error percentages show that the number of frames which are classified correctly by the acoustic system and incorrectly by the articulatory system is higher under clean conditions; however, this relation is reversed under mismatched conditions, where the articulatory system achieves a higher proportion of correct classifications than the acoustic system. Furthermore, the percentage of different errors rises compared to the percentage of identical errors under noise conditions.

It is not surprising that the classifiers increasingly disagree in the presence of noise. The question we need to ask is whether there is a distinct qualitative pattern underlying the disagreement. It might be assumed, for instance, that the articulatory systems produce confusions which are more interpretable in phonetic or articulatory terms. In order to determine the qualitative differences between the acoustic and articulatory systems, the frame-level phone confusion matrices of each system were analyzed. Figures 3.7 to 3.9 show plots of the diagonals of the phone confusion matrices, indicating to which degree each system is able to correctly classify the various phone classes. There is no general pattern of errors which shows similar strengths or weaknesses of one system vs. the other across all different acoustic conditions. The phone accuracy rates for clean speech seem to indicate a slight tendency of the articulatory system to better classify consonants, especially stops and fricatives, than the acoustic system, which performs better at vowels.

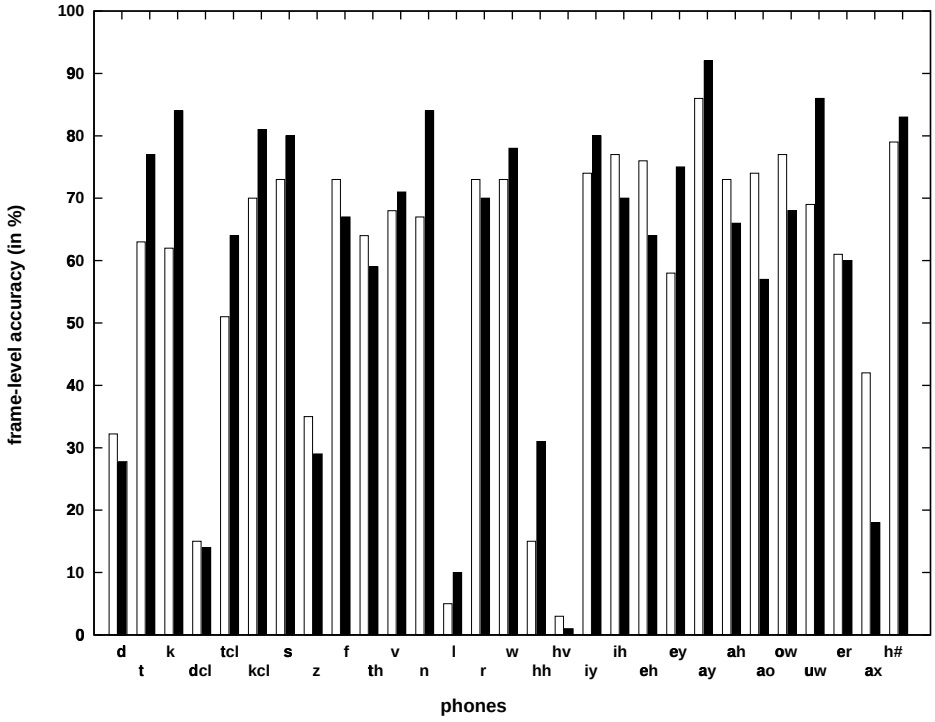

Figure 3.7: Frame-level accuracy rates for phone classes (clean speech). Shaded boxes represent the articulatory system, blank boxes represent the acoustic system.

However, this relation seems to be almost reversed under reverberant and noisy conditions: here, the articulatory systems are more accurate at classifying vowels than consonants. Additionally, they identify silence more accurately. Further information is provided by the most frequent off-diagonal confusions among the phone classes. The most frequently confused phone pairs and the confusion rates are shown in Table 3.14. Again, the data do not warrant a general statement about the strengths and weaknesses of each system.

At the word level, error difference percentages were computed analogous to the frame-level analysis. Based on the alignment with the reference transcription, the correctly and incorrectly recognized words were identified. For each pair of articulatory and acoustic recognizers it was determined whether

- both recognizers were correct,
- recognizer A was wrong and recognizer B was correct,
- recognizer B was wrong and recognizer A was correct,
- both recognizers were wrong and the errors were identical,
- both recognizers were wrong and the errors were different.

AF clean		**AC clean**		**AF reverb**		**AC reverb**	
phone pair	%	phone pair	%	phone pair	%	phone pair	%
ao ow	60.00	ax ah	54.50	z s	30.20	l n	20.10
k s	48.40	ao ow	46.70	hh ow	25.00	hv ay	20.00
ax ah	47.30	aa ay	40.80	k kcl	19.90	hv th	20.00
hv uw	40.00	d t	23.10	hh k	16.70	dcl tcl	18.90
hv ah	40.00	aa ah	22.40	hh w	16.70	dcl v	17.10
dcl tcl	31.20	l ow	21.20	hh n	16.70	hv w	15.00
d iy	31.20	ao r	20.20	er ay	15.40	d iy	13.80
dcl t	27.80	k s	18.60	hv th	14.80	z th	13.20
hh ah	23.40	ax n	18.20	ax v	13.70	l ow	12.90
l ow	20.20	hv hh	14.30	hv h#	13.10	hh hv	11.90
AF noise 30 dB		**AC noise 30 dB**		**AF noise 20 dB**		**AC noise 20 dB**	
phone pair	%	phone pair	%	phone pair	%	phone pair	%
hh v	31.60	hv f	66.67	z s	25.50	hv f	40.00
hv hh	27.50	hv hh	33.33	k kcl	20.40	l n	22.82
dcl er	25.20	l n	21.64	dcl er	19.70	s h#	20.34
z s	24.90	z th	20.67	hv h#	19.00	hv hh	20.00
k kcl	19.60	l ow	17.54	l n	18.70	hv ay	20.00
l ow	18.40	hh n	15.07	hh tcl	17.60	hv ah	20.00
kcl ih	17.50	s h#	14.38	kcl ih	17.00	l ow	19.25
hh hv	15.80	hh f	13.70	l s	15.50	z th	18.63
ax v	14.70	k kcl	13.37	s h#	15.10	hh n	18.18
t tcl	14.50	kcl ih	12.92	l ow	14.80	kcl ih	13.68
AF noise 10 dB		**AC noise 10 dB**		**AF noise 0 dB**		**AC noise 0 dB**	
phone pair	%	phone pair	%	phone pair	%	phone pair	%
hh h#	47.10	hv hh	60.00	hh h#	39.10	t h#	23.99
z s	24.70	s h#	22.20	er ay	26.20	hv w	23.08
k kcl	20.40	hv th	20.00	z s	23.30	hv tcl	23.08
l n	20.30	hv ay	20.00	hv h#	23.10	hv hh	23.08
er ay	19.10	l n	19.39	hh kcl	21.70	s h#	21.36
s h#	18.60	t h#	18.87	t h#	21.40	hh h#	20.29
hv hh	18.60	hh n	17.15	s h#	19.30	uw h#	19.56
t h#	18.20	z th	16.46	hv hh	19.20	k h#	19.14
hh tcl	17.60	hh h#	16.32	k kcl	19.10	dcl n	16.72
hh v	17.60	k h#	16.27	uw h#	17.20	iy h#	16.24

Table 3.14: Most frequent frame-level phone-pair confusions and confusion rates (in %). AC = acoustic system, AF = articulatory system.

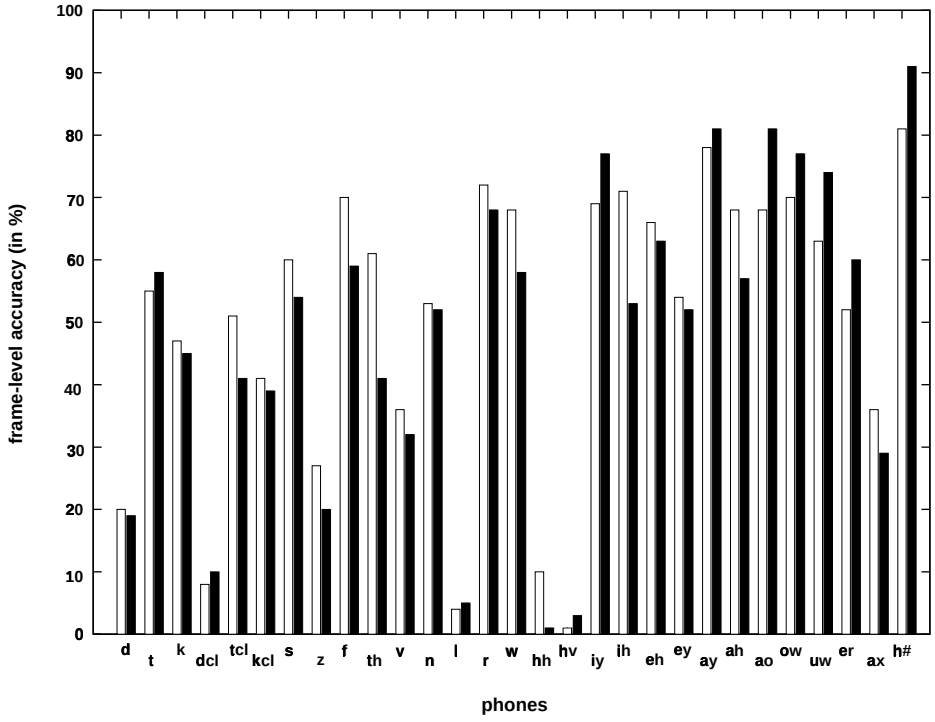

Figure 3.8: Frame-level accuracy rates for phone classes (reverberant speech). Shaded boxes represent the articulatory system, blank boxes represent the acoustic system.

Test Condition	**both correct**	**both wrong**	**AC only correct**	**AF only correct**	**same errors**	**different errors**
clean	90.99	4.15	2.61	2.27	70.10	29.90
reverb	72.69	13.64	7.04	6.63	57.93	42.07
noise, 30 dB	79.78	9.57	5.39	5.26	67.44	32.66
noise, 20 dB	75.13	12.06	5.37	7.43	64.90	35.10
noise, 10 dB	65.10	16.58	7.34	10.98	56.23	43.77
noise, 0 dB	49.65	27.71	8.51	14.06	51.66	48.34

Table 3.15: Error percentages (word-level) for different acoustic conditions. AC = acoustic system, AF = articulatory system

The results are shown in Table 3.15.

Similar to the frame-level error analysis, we again observe that the number of different errors increases as the signal quality deteriorates, indicating that the articulatory and acoustic systems focus on different information under these conditions.

To conclude this error analysis, Table 3.16 show the ten most frequent word-level confusions pairs for each system. Again, there is no distinct error pattern to be observed for either clean,

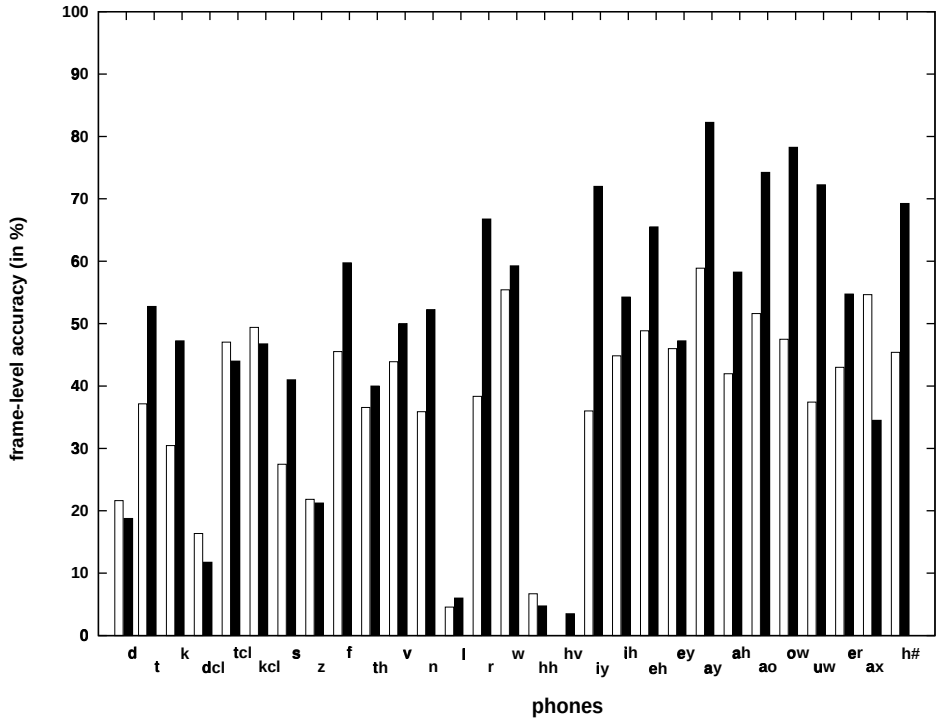

Figure 3.9: Frame-level accuracy rates for phone classes (noisy speech, averaged over all noise conditions. Shaded boxes represent the articulatory system, blank boxes represent the acoustic system.)

AF clean		**AC clean**		**AF reverb**		**AC reverb**	
#inst	words	#inst	words	#inst	words	#inst	words
35	<ins> oh	28	<ins> oh	41	eight <del>	60	eight <del>
20	oh <del>	16	oh <del>	35	<ins> oh	40	oh <del>
14	oh four	13	one <del>	33	oh <del>	37	<ins> oh
13	eight <del>	13	eight <del>	33	nine five	33	nine <del>
23	<ins> eight	13	<ins> eight	32	nine <del>	30	two <del>
11	<ins> four	11	nine five	28	<ins> two	24	eight three
10	<ins> two	11	fifty sixty	27	one four	22	five nine
9	two <del>	9	<ins> one	27	oh four	21	oh zero
9	<ins> six	8	<ins> two	26	one <del>	21	eight eighty
9	<ins> one	7	three thirty	22	oh zero	20	one <del>
AF noise 30 dB		**AC noise 30 dB**		**AF noise 20 dB**		**AC noise 20 dB**	
45	nine five	52	nine five	73	<ins> two	59	nine five
32	eight <del>	36	oh four	51	nine five	52	oh four
32	<ins> two	33	eight <del>	34	oh four	40	oh <del>
28	oh four	29	oh <del>	31	eight <del>	38	eight <del>
26	oh <del>	27	<ins> oh	26	one four	30	one four
22	one four	25	one four	26	oh <del>	29	nine <del>
19	one <del>	19	nine <del>	26	<ins> four	28	<ins> two
19	oh zero	19	eight three	24	one <del>	26	eight three
19	<ins> oh	18	one <del>	20	oh zero	24	two <del>
15	nine <del>	16	<ins> two	20	<ins> oh	22	one <del>
AF noise, 10 dB		**AC noise, 10 dB**		**AF noise, 0 dB**		**AC noise, 0 dB**	
103	<ins> two	63	oh four	122	<ins> two	101	<ins> three
53	nine five	61	<ins> two	66	one <del>	94	<ins> two
38	one <del>	57	nine five	65	oh <del>	79	one <del>
38	oh <del>	55	oh <del>	58	<ins> three	78	oh four
37	eight <del>	50	<ins> three	53	eight three	75	oh <del>
36	oh four	48	one <del>	52	eight <del>	72	two <del>
35	<ins> three	47	eight <del>	51	nine five	60	eight <del>
31	one four	44	two <del>	48	oh four	56	nine <del>
30	eight three	40	one four	47	nine <del>	48	nine five
30	<ins> four	40	eight three	41	two <del>	46	one four

Table 3.16: List of ten most frequent word confusion pairs; AC = acoustic system, AF = articulatory system.

reverberant, or noisy speech.

3.4 Combination Rules

In the previous section it was shown that the different recognition systems exhibit error patterns which differ both quantitatively and qualitatively. For this reason, a combination of both systems might be beneficial as one system may compensate for the errors made by the other system and vice versa. Speech recognizers may be combined at various levels in the recognition process: at the feature level, the frame level, the word level, or the utterance level. Feature-level combination involves concatenating different feature vectors and training the system on the combined vectors. A frame-level (or state-level) combination procedure merges the frame or state-level emission probabilities computed by different systems. At the word and utterance level several combination strategies are possible, such as two-level Viterbi decoding or N-best lattice rescoring. In this chapter we concentrate on frame-level combination; further combination strategies will be discussed in the following chapters.

At the frame-level, recognizer combination reduces to combining the local classifiers. In the current context of hybrid recognition systems this involves combining the outputs from the different phone MLPs. The topic of classifier combination in general, and that of neural network ensembles in particular, has received much attention in the machine learning community. When dealing with a complex pattern recognition task, such as speech recognition in various acoustic environments, it is often the case that no single classifier can be developed which can satisfactorily solve the task. However, an ensemble of classifiers may be capable of achieving a more robust performance. Different classifiers trained on the same input but differing in structure or with respect to initialization may develop different strengths and weaknesses during training. Classifiers which are trained using different inputs may extract partially different, or even complementary, information about the classes from their respective feature spaces. Furthermore, using a set of small classifiers and combining their decisions rather than training a large holistic classifier may reduce training and development effort and may lead to better convergence and generalization properties. For these reasons, classifier combination is often preferred to complex individual classifiers.

Several approaches to classifier combination have been proposed in the literature. In contrast to the previous section, where classifiers for different sets of output classes were arranged sequentially, the focus is now on those approaches which make use of parallel classifiers trained on the same set of output classes. Possible strategies for handling the output from a set of parallel classifiers include classifier selection, voting, or classifier merging, e.g. by mixture of experts (see above) or by a linear combination of the class probabilities. Classifier selection means that the output from one classifier is selected as the correct one among all outputs from the classifier ensemble. Various performance measures can be used to determine which classifier to select. In

[61], for instance, classifiers are selected on the basis of an estimate of their local accuracy. In a speech recognition system various types of confidence values might be used to determine the selection of classifier outputs.

The voting scheme (e.g. [46, 5, 7]) considers the decisions made by all classifiers and adopts that decision on which most members of the ensemble agree. Possible ties are broken arbitrarily. This method is most suitable for a large set of classifiers. In our case, it is suboptimal because only two classifiers are involved and too many tie situations may arise which cannot always be solved in a principled way.

The classifier-merging approach takes into account the probability distributions of the different classifiers instead of considering only the hard decisions in terms of the resulting class labels. The output distributions may be combined by means of linear combination rules, or in a non-linear way, e.g. by training another non-linear classifier on the combined output distributions. Generally speaking, combination by a non-linear classifier yields better results because the higher-level classifier can in principle approximate arbitrary mappings between the output probability distributions of the individual classifiers and the desired output distribution. However, this method involves another training phase and introduces additional complexity into the overall system. This can be avoided when using linear combination rules. A linear combination of classifier outputs can in certain cases be shown to provide an improvement to the overall classification performance. The case of ensemble averaging of regression-based classifiers, for instance, where outputs are combined by a weighted sum, has been shown (e.g. [75]) to provide an improvement over the individual classifiers. Let $g_i(x)$ denote the output of classifier i and $f(x)$ denote the target function. The ensemble output $\overline{g}(x)$ is defined as the weighted sum of the individual outputs:

$$\overline{g}(x) = \sum_i w_i g_i(x) \tag{3.22}$$

This is the same definition as the one in Equation 3.18. The approximation error (or *bias*) $\epsilon_i(x)$ of each individual classifier is the squared difference between the output and the target function

$$\epsilon_i(x) = E[(g_i(x) - f(x))^2] \tag{3.23}$$

where the expectation is computed over the training samples. The ensemble approximation error is the expected value of the squared difference between the ensemble output and the target function

$$e(x) = E[(f(x) - \overline{g}(x))^2] \tag{3.24}$$

The variance of each classifier, $v_i(x)$ is defined as the squared difference between its output and the ensemble output

$$v_i(x) = E[(g_i(x) - \overline{g}(x))^2] \tag{3.25}$$

and the ensemble variance is the weighted sum of the individual variances

$$\overline{v}(x) = \sum_i w_i v_i(x) \tag{3.26}$$

Equation 3.26 can be rewritten as

$$\overline{v}(x) = \sum_i w_i E[(g_i(x) - \overline{g}(x))^2] \tag{3.27}$$

Again, this is the definition of ensemble variance which already encountered in Equations 3.19 and 3.20. By adding and subtracting $f(x)$ to this we obtain

$$\overline{v}(x) = \sum_i w_i \epsilon_i(x) - e(x) \tag{3.28}$$

according to definitions 3.23 and 3.24.

If $\sum_i w_i \epsilon_i(x)$ is denoted as $\overline{\epsilon}(x)$ the ensemble approximation error can be redefined as

$$e(x) = \overline{\epsilon}(x) - \overline{v}(x) \tag{3.29}$$

That is, the ensemble approximation error is the weighted sum of the individual approximation errors minus the ensemble variance, which is guaranteed to be lower than the weighted sum of the individual errors unless the variance is zero. Thus, ensemble combination benefits from large differences among the outputs of the individual classifiers.

This analysis generalizes to the case of several different inputs $x_1, ..., x_N$ to the classifiers $c_1, ..., c_N$, since the target function $f(\cdot)$ is the same for all $x_1, ..., x_N$. The optimal combination rule should be that which both minimizes the first term on the right-hand side of equation 3.29 and which maximizes the ensemble variance.

A good overview of other widely used linear combination rules besides the averaging rule is presented in [71]. Assume that there are N different classifiers $C = c_1, c_2, ...c_N$, corresponding to different input representations $X = \mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}$, each of which is applied to the task of distinguishing between K output classes $\Omega = \{\omega_1, \omega_2, ..., \omega_K\}$. Each classifier yields a likelihood $p_n(\mathbf{x_n}|\omega_k)$ for a pattern $\mathbf{x_n}$ given class ω_k in recognizer c_n. The joint probability for a pattern to occur in the N different representations given k is

$$p(\mathbf{x_1}, ..., \mathbf{x_N}|\omega_k) \tag{3.30}$$

It is often computationally infeasible to estimate this joint probability directly; however, under the assumption that the input representations to the different classifiers are statistically independent given the classes, the above rule can be approximated by

$$p(\mathbf{x_1}, ..., \mathbf{x_N}|\omega_k) = \prod_{n=1}^{N} p(\mathbf{x_n}|\omega_k) \tag{3.31}$$

The Bayes decision rule for the optimal class given a pattern Y, K classes and N different classifiers is

$$Y \rightarrow \omega_j \quad if \quad P(\omega_j|\mathbf{x_1},...,\mathbf{x_N}) = \max_k P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) \tag{3.32}$$

where

$$P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) = \frac{p(\mathbf{x_1},...,\mathbf{x_N}|\omega_k)P(\omega_k)}{p(\mathbf{x_1},...,\mathbf{x_N})} \tag{3.33}$$

and where $P(\omega_k)$ is the *a priori* probability of class k. Substituting (3.31) in (3.33), we obtain

$$P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) = \frac{P(\omega_k)\prod_{n=1}^{N} p(\mathbf{x_n}|\omega_k)}{\sum_{k=1}^{K} P(\omega_k)\prod_{n=1}^{N} p(\mathbf{x_n}|\omega_k)} \tag{3.34}$$

If this combination rule is to be expressed in terms of the *a posteriori* probabilities of the different classifiers, we need to divide the product by the *a priori* probabilities, assuming that all classes have equal prior probabilities in the different input representations.

$$P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) = \frac{1}{P(\omega_k)^{N-1}} \prod_{n=1}^{N} P(\omega_k|\mathbf{x_n}) \tag{3.35}$$

Thus, the Bayes decision rule becomes

$$Y \rightarrow \omega_j \quad if \quad \frac{1}{P(\omega_j)^{N-1}} \prod_{n=1}^{N} P(\omega_j|\mathbf{x_n}) = \max_k \frac{1}{P(\omega_k)^{N-1}} \prod_{n=1}^{N} P(\omega_k|\mathbf{x_n}) \tag{3.36}$$

The drawback of this *product rule* is that the overall likelihood of a hypothesis becomes zero if one classifier outputs an *a posteriori* probability close to zero. The product rule thus implements an “and” function whose output is large if and only if both inputs are large.

The *min rule* selects that output which is smallest

$$P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) = \frac{min_n P(\omega_k|\mathbf{x_n})}{\sum_{k=1}^{K} min_n P(\omega_k|\mathbf{x_n})} \tag{3.37}$$

whereas the *max rule* selects the largest output:

$$P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) = \frac{max_n P(\omega_k|\mathbf{x_n})}{\sum_{k=1}^{K} max_n P(\omega_k|\mathbf{x_n})} \tag{3.38}$$

Similar to the product rule, the *min* rule implements an “and” function since the output is large if and only if both of the inputs are large. The *max* rule and the averaging or *sum* rule discussed above, however, have the effect of an “or” function: if one of the classifiers’ outputs is large, the final output will be large as well.

In various classification experiments, Kittler et al. [71] observed that the sum rule provided the best results. The authors explained this by the greater robustness of the sum rule to estimation errors. In the sum rule, the errors incurred by the individual classifiers (where (e_{kn}) denotes the estimation error for class k in classifier n, $0 < e_{kn} \ll P(\omega_k|\mathbf{x_n})$) are dampened whereas they are amplified by product rule combination. Each term in the product rule introduces the error factor

$$1 + \sum_{n=1}^{N} \frac{e_{kn}}{P(\omega_k|\mathbf{x_n})} \tag{3.39}$$

whereas each term in the sum rule is affected by

$$\frac{\sum_{n=1}^{N} e_{kn}}{\sum_{n=1}^{N} P(\omega_{k|\mathbf{x_n}})} \tag{3.40}$$

Based on this finding we can make predictions about the performance of the different linear probability combination rules in the current context. Due to the greater error robustness of the sum rule, a sum combination scheme might prove more advantageous in acoustically degraded conditions, such as reverberation and noise.

In the context of speech recognition several studies have investigated linear combinations of probabilities using some form of product and/or sum rule. In [69] and [124] the log-likelihoods derived from the posterior probabilities estimated by different MLPs (based on different feature inputs or representing different subword units) are combined by an unweighted sum. Halberstadt & Glass [56] compare a weighted sum rule to a product rule to combine likelihoods obtained from Gaussian mixture classifiers on heterogeneous acoustic measurements. Across a range of different experiments, they found that the product rule always yielded the best results. McMahon et al. [88] use a weighted sum of log-likelihoods for the recombination of subband features. In sum, successful linear combination methods which have previously been reported in speech recognition always involve a product combination, which either takes the form of a sum of log-likelihoods or a product of linear likelihoods. This is somewhat surprising considering the statistical independence assumption underlying the product rule and the supposedly greater sensitivity to estimation errors in the individual classifiers. In the following section we will compare the performance of the different combination rules in different acoustic conditions.

3.5 Combination Experiments and Results

For the present purpose of combining the outputs from the phone MLPs of the acoustic and articulatory systems, two different approaches were used: (a) the linear combination rules described above, and (b) a non-linear classifier in the form of an MLP whose input consisted of the

Test set	AF	AC	sum	product	max	min
clean	8.9%	8.4%	7.8%	**7.3%**	7.9%	7.8%
reverberant	23.7%	24.7	24.5%	**21.1%**	25.7%	**21.7%**
noise, 30 dB	17.4%	17.2%	17.4%	**15.1%**	18.2%	16.0%
noise, 20 dB	21.7%	22.8%	21.8%	**18.8%**	22.7%	**19.7%**
noise, 10 dB	30.0%	32.7%	31.0%	**28.3%**	32.7%	29.0%
noise, 0 dB	43.6%	50.2%	48.3%	**41.6%**	49.6%	45.1%

Table 3.17: Word error rates obtained by different linear combination rules. Statistically significant differences compared to the better of the AC/AF baselines are shown in boldface.

concatenated output probability distributions of the phone MLPs and whose output is another probability distribution over the subword phone classes. The product rule used as one of the linear combination rules differed slightly from the definition given above: the derivation given by [71] departs from the likelihoods $p(\mathbf{x_n}|\omega_k)$ – in our case, however, we start from the posterior probabilities $p(\omega_k|\mathbf{x_n})$ estimated by the MLPs, which are combined according to

$$P(\omega_k|\mathbf{x_1},...,\mathbf{x_N}) = \frac{\prod_{n=1}^{N} P(\omega_k|\mathbf{x_n})}{\sum_{k=1}^{K}\prod_{n=1}^{N} P(\omega_k|\mathbf{x_n})} \tag{3.41}$$

An initial experiment was conducted on the clean test set to evaluate the performance of the linear vs. the non-linear approach. It was found that the lowest word error rate obtained by the MLP combination method was 7.8%, which was higher than the best result obtained using a linear combination rule (7.3% obtained by the product rule). A reason for the inferior performance of the non-linear combination may be the limited amount of training data in relation to the large number of parameters which have to be estimated (due to the concatenation of the 56-dimensional phone probability vectors from both systems, the resulting input space has 112 dimensions). Combination experiments were then conducted on all acoustic test sets, using only the linear combination rules. Table 3.17 shows the resulting word error rates. It is obvious that the product rule consistently produces the lowest word error rates, followed by the *min* rule, the sum rule and the *max* rule. At first sight, this result contradicts the findings by Kittler et al. : according to our data, it is not the sum rule but the product rule which seems to be the most robust combination rule, in clean as well as in noisy conditions. Note that this is in line with the previous combination studies in the context of speech recognition described above. How can this apparent discrepancy be explained? The word error rates provide an evaluation of the entire recognition system, which includes the decoding process, whereas we wish to evaluate the performance of the local phone classifiers only. We should therefore take a look at the frame-level recognizer output in order to analyze this problem in greater detail. Similar to the error analysis

of the individual acoustic and articulatory classifiers, we computed the frame error rates and entropy ratios (of correctly vs. incorrectly classified frames) of the combined systems. These are shown in Table 3.18. As we can see, the frame error rates in the combined system are con-

Test set	**sum**			**prod**		
	WER	FER	ER	WER	FER	ER
clean	7.8%	22.06%	0.17	7.3%	21.49%	0.13
reverb	23.4%	31.09%	0.24	19.9%	30.46%	0.18
noise 30 dB	17.1%	27.11%	0.21	15.0%	26.49%	0.18
noise 20 dB	21.8%	31.66%	0.24	18.5%	31.08%	0.18
noise 10 dB	31.0%	39.99%	0.30	28.3%	39.26%	0.22
noise 0 dB	48.0%	51.46%	0.37	41.5%	50.75%	0.27
Test set	**min**			**max**		
	WER	FER	ER	WER	FER	ER
clean	7.8%	21.54%	0.14	7.9%	22.55%	0.18
reverb	20.3%	30.79%	0.20	24.5%	31.57%	0.25
noise 30 dB	15.8%	26.81%	0.18	17.8%	27.59%	0.23
noise 20 dB	19.7%	31.55%	0.20	22.6%	32.17%	0.25
noise 10 dB	29.0%	39.99%	0.24	32.1%	40.55%	0.31
noise 0 dB	45.1%	51.35%	0.29	48.4%	52.07%	0.39

Table 3.18: Word error (WER) and frame error rates (FER) as well as entropy ratios (ER) obtained using different linear combination rules.

sistently lower than the frame error rates obtained by the individual classifiers (see Table 3.10). Furthermore, frame error rates reveal that the differences in frame-level performance of the combined classifiers are only very slight – the differences in frame error rate between the sum rule and the product rule, for instance, are in most cases not significant. However, different combination rules have a very variable impact on the word error rate - whereas some rules produce word error rates which are lower than those of the individual system, others increase the word error rates to the extent that they exceed those of the baseline systems!

A good indication of the reason why this is the case is provided by the entropy ratios: the product rule and the *min* rule, which achieve the best results at the word level, also exhibit the lowest entropy ratios, whereas the entropy ratios of the sum and the *max* rule are markedly higher. This shows that the phone output distributions created by the different combination schemes are discriminative to varying degrees. As explained above, the optimal behavior of a recognition system is characterized by low-entropy probability distributions for correct classifications and high-entropy probability distributions for incorrect classifications, such that correct hypotheses

are maximally distinct from incorrect hypotheses when the system is right and that several (correct or incorrect) answers are maximally similar when the system is wrong. The consequences for word-level decoding are that, at a given frame, a correctly recognized class will receive most of the probability mass whereas the incorrect classes will have a very low probability and are thus likely to be pruned from the search beam. If the highest-scoring class is actually incorrect but it is close to other classes in terms of its score, the correct class might be preserved in the search beam and may contribute to finding the globally best path. Obviously, the product rule and the *min* rule combination schemes favor this situation whereas the other methods do not. It is important to realize that the performance of the overall speech recognizer does not solely depend on the accuracy of the frame or state-level classifier – the degree of discriminability between different classes is at least as significant with respect to higher-level search. When deciding between different combination methods of state-level probability distributions care should therefore be taken to choose a combination method which maximizes discriminability. It should also be pointed out, however, that the correlation between the classifiers' outputs plays an important role, too. If perfectly uncorrelated outputs were combined using the product or *min* rule, they would cancel each other out.

3.6 Summary and Discussion

In this chapter we described the development (initial feature selection, feature classification, and feature optimization) of an articulatory feature based recognition system for continuous numbers recognition. A comparison of the baseline recognition results obtained by the articulatory system and those achieved by a standard acoustics-only recognizer revealed a comparable level of performance for both systems on clean and moderately noisy speech (30 dB SNR). The articulatory system showed a small (although statistically not significant) improvement compared to the acoustic system on reverberant speech and noisy speech at 20 dB SNR. Statistically significant improvements were obtained by the articulatory system on noisy speech at low (10 and 0 dB SNR) signal-to-noise ratios. The single largest improvement relative to the performance of the acoustic system was a 13.1% decrease in word error rate. A detailed error analysis at the frame and at the word level showed that the outputs of both systems differ both quantitatively and qualitatively, i.e. errors are made with respect to different classes. The different error patterns, however, did not lend themselves to any phonetically-based interpretation. Furthermore, the types of errors became increasingly different with decreasing signal-to-noise ratio.

A comparison of several classifier combination schemes applied to the outputs of the phone MLPs in the different systems showed that performance can further be improved by simple linear combinations of the posterior phone probabilities. Of the four different combination rules

which were investigated, the product rule yielded the best results: across all acoustic conditions, the improvements were statistically significant. The largest individual improvement obtained by integrating articulatory information was a 17.3% relative decrease in word error rate in the pink noise 0 dB SNR test case compared to the performance of the acoustics-only system. The superior performance of the product rule could be explained by its effect on the entropy of the resulting output distribution: contrary to "or" function rules like the sum and *max* rule, the product rule and the *min* rule – which implement "and" functions" – decrease the ratio of the entropy of the phone distribution of the correct frames to those of the incorrect frames and thus lead to better discriminability of classes at higher-levels in the decoding process.

This preliminary study raises a number of further questions. First, the recognition task used for the experiments reported in this section is relatively limited. Since the Numbers95 vocabulary is very small, only a limited amount of phonetic variability is covered by the data. Although these preliminary results are promising, the potential of the articulatory feature based approach should be tested on larger recognition task. Second, further tests on noisy data, including more realistic types of noise, should be performed in order to verify the superior performance of the AF system in noise.

The classifier combination schemes presented in this section are capable of achieving a significant reduction in word error rate – however, even greater improvements might be gained when the systems are combined at higher levels, such as the word or utterance level. Typically, recognition hypotheses at higher levels are more robust because evidence from wider temporal contexts is available. Therefore, higher-level combination methods such as N-best lattice rescoring should be investigated. Furthermore, it has been shown that the most successful combination schemes enhance the ability of the system to discriminate between incorrect and correct hypotheses. Therefore, another promising combination strategy might be to use a weighted combination rule, e.g. a weighted product rule, where the weights are optimized with respect to a discriminative criterion, such as the Maximum A Posteriori probability (MAP) of the utterance or the Minimum Classification Error (MCE) [4, 66, 24].

Chapter 4

Articulatory Features for Large Vocabulary Conversational Speech Recognition

In this chapter we extend the pilot study described in the previous chapter to a large-vocabulary conversational speech recognition task. We will compare and analyze the performance of acoustic and articulatory baseline systems on this task and investigate techniques for combining the two systems. In addition to combination at the state level, we will investigate possibilities of feature-level and word-level system combination.

4.1 Corpus and Baseline System

4.1.1 Corpus

The corpus used for the experiments reported in this chapter is the German Verbmobil corpus [73], which is a collection of spontaneous dialogues within the domain of appointment scheduling. A typical turn exchange in this corpus is exemplified by the following extract:

Speaker 1: ja Frau Gehrmann <Atmen> wir haben <Pause> wiederum ein Arbeitstreffen zu vereinbaren <Atmen> ich schlage einfach mal vor vom zehnten bis vierzehnten Oktober <Pause> in Berlin wie sieht es <undeutlich> aus <Pause>

Speaker 2: <Schmatzen> <Atmen> ganz schlecht <Pause> da bin ich leider schon <Pause> unterwegs bei mir ginge es erst ab dreizehnten Oktober <Geräusch>

Approximate translation:
Speaker 1: yes Ms. Gehrmann <breathe> we need to <silence> schedule a work meeting again

<breathe> I would suggest October the tenth to October the fourteenth <silence> in Berlin how does that <incomprehensible> look

Speaker 2: <smack> <breathe> very bad <silence> I've already planned to be <silence> away then I can only make it after October the thirteenth <noise>

The data (studio-quality speech sampled at 16 kHz) was recorded at four different locations, viz. at the Universities of Kiel, Bonn, Karlsruhe and Munich, using different microphones and recording environments. The training set used for the present experiments comprises approximately 30 hrs, the test set (the official 1996 Verbmobil evaluation task) consists of 343 utterances (45 minutes). The number of speakers in the total set is 749. Since the corpus consists of spontaneous utterances it contains numerous hesitations, fillers, false starts, and other disfluencies, as well as noises like laughter, coughing and lip smacks. In addition to this, the test set contains out-of-vocabulary (OOV) words, in particular proper names and spelling sequences. The recognition lexicon consists of 5333 entries. The bigram perplexity is 64.2. It is obvious that this task is significantly more complex than the numbers recognition task used for the pilot study described in the previous chapter.

4.1.2 Recognition System

The recognition system which was used for the Verbmobil experiments is the ESMERALDA (**E**nvironment for **S**tatistical **M**odel **E**stimation and **R**ecognition on **A**rbitrary **L**inear **D**ata **A**rrays) system, which is a vector quantization (VQ) based HMM recognition system [43]. The core of the acoustic modeling component in this system is a VQ codebook with a pre-specified number of classes each of which is represented by a Gaussian pdf

$$\mathcal{N}(\mathbf{x};\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)} \tag{4.1}$$

where μ is the mean and Σ is the covariance of the pdf; n is the dimensionality of the feature space. HMM state emission probabilities (i.e. the likelihoods $p(\mathbf{x_t}|q_i)$ of an observation vector $\mathbf{x_t}$ given a HMM state q_i) are computed by a weighted sum (or *mixture*) of the M codebook pdfs:

$$p(\mathbf{x_t}|q_i) = \sum_{m=1}^{M} c_{mi}\mathcal{N}(\mathbf{x_t};\mu_m,\Sigma_m) \tag{4.2}$$

where $\mu_{\mathbf{mi}}$ and Σ_{mi} are the mean and covariance, respectively, of the $m'th$ Gaussian mixture component of the codebook and c_{mi} is the mixture weight of state q_i for that component. The mixture weights are associated with the individual HMM states, whereas the codebook pdfs are shared by all states.

The codebook is estimated using a variant of the Linde-Buzo-Gray (LBG) algorithm [84], which proceeds as follows:

LBG-Based Vector Quantization

1: initialize the codebook by assigning feature vectors to classes, either based on an externally provided labeling or by cyclically assigning the $n'th$ feature vector to the *N mod n'th* class.
2: delete classes with fewer than m feature vectors (where m is some percentage of the total number of feature vectors)
3: compute initial statistics (means and covariances) for all classes
4: **while** the number of classes is smaller than C (the desired number of classes) **do**
5: classify the feature vectors by assigning them to the nearest class, using some distance measure. In the present system, the distance is computed by evaluating the Gaussian pdf.
6: update means and covariances.
7: delete classes with fewer than m feature vectors.
8: **if** the global distance (the average distance of feature vectors to their class means) falls below the previous global distance by a pre-specified percentage k **then**
9: split those classes whose average distance (of feature vectors to the mean) is above a threshold θ. The means and covariances of the old classes are copied over to the new classes; the covariances are additionally perturbed by small constants derived from the covariances of the old classes.
10: **end if**
11: **end while**

After the codebook has been trained, HMM state emission probabilities are trained by updating the weights for the shared pdf's in the codebook. More precisely, HMM training consists of

- initialization of HMM states, followed by one pass of Baum-Welch re-estimation,
- clustering of HMM states,
- several iterations of embedded Baum-Welch reestimation of state mixture weights, possibly coupled with an update of the codebook.

Initialization of (context-independent) HMMs is performed by using an externally specified time alignment of phone-based transcriptions and the speech data files. In a first step, duration statistics are computed for each phone model. Either single-state or multi-state HMMs are then created for each phone. Unless the system is forced to create only single states for each model,

the number of states in each model is determined automatically based on the minimum duration of the model. More specifically, the number of states m is $f * mindur(m)$, where f is a user-specified scaling factor and $mindur(m)$ is the minimum duration of model m. States within the same model can be identical (hard-tied) or independent. The models thus created are then initialized by uniformly assigning the frames belonging to each model instance (as determined by the external alignment) to the model states and computing the initial state parameters as follows: The state-dependent weights for all codebook classes (mixture components) are determined by

$$w_{sk} = \frac{1}{N_s} \sum_{n=1}^{N_s} p(\mathbf{x_n}|\omega_k) \tag{4.3}$$

where w_{sk} is the weight for state s of class k, N_s is the number of training samples assigned to state s, and $p(\mathbf{x_n}|\omega_k)$ is the likelihood of the $n'th$ observation assigned to state s given class k. State transition probabilities are updated by

$$a_{ij} = \frac{N_{ij}}{N_i} \tag{4.4}$$

where a_{ij} is the transition probability from state i to state j, N_{ij} is the number of transitions from state i to state j and N_i is the total number of transitions out of state i.

In all subsequent training passes, re-estimation of the model parameters is then carried out using the Baum-Welch algorithm [6]. Additionally, the the means and variances of the codebook classes can be updated during training.

After the first training iteration, context-dependent phones are created using the reference transcriptions and triphone-based word definitions. First, a new state is created for each triphone whose count exceeds a minimum number of training samples. Triphone states are aliased to their corresponding basephone states; e.g., b/I/t would be aliased to /I/. A bottom-up agglomerative clustering algorithm is then applied to the resulting state space in order to reduce the number of free parameters. This algorithm iteratively merges states into clusters until a (user-defined) minimum number of training samples is present in each cluster. The details of this procedure (based on [78]) are as follows:

HMM State Clustering

1: **for** each set of triphones aliased to the same basephone **do**
2: **if** the alias group has enough training samples ($2 * N$, where N is the minimum cluster size) **then**
3: create a separate cluster for each state
4: **for** each cluster **do**
5: **while** the minimum cluster size is smaller than N **do**

merge the two clusters i and j with the smallest distance $D(i,j)$ which have not yet been merged
create a new cluster for the merged pair
end while
end for
create new state definitions for all resulting clusters
end if
end for

The distance $D(i,j)$ between clusters i and j is defined as

$$D(i,j) = N_{ij}H(i,j) - N_iH(i) - N_jH(j) \tag{4.5}$$

where $H(i)$ is the entropy of cluster i, $H(j)$ is the entropy of cluster j, and $H(i,j)$ is the joint entropy of the two clusters. The terms N_i, N_j, and N_{ij} are the number of samples in the clusters i, j, and the in cluster resulting from pooling i and j, respectively. Equation 4.5 is essentially an approximation to the negative of the mutual information $I(X;Y)$

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{4.6}$$

The cluster entropies are additionally weighted by the cluster size. The algorithm thus merges those clusters which have the highest mutual information, i.e. those which are highly predictable from each other. The cluster entropies are computed as averages of the entropies of the individual states assigned to a cluster. The entropy for state q_i, $H(q_i)$, is defined as

$$H(q_i) = \sum_{m=1}^{M} w_{im} log w_{im} \tag{4.7}$$

where M is the number of mixture components in the codebook and w_{im} is the weight of state q_i for the $m'th$ mixture component. Thus, state entropy is computed from the state weights only.

Lexical decoding proceeds incrementally, based on a time-synchronous beam-search algorithm and a tree-structured recognition lexicon. In contrast to conventional tree-based decoding where copies of the recognition tree are made at the end of each word, tree copies are made on a temporal basis, indexed by their starting times. This restricts the number of copies which need to be made and leads to greater efficiency. Since this decoder makes use of an incremental, frame-to-frame processing strategy the advantages of a multi-pass search (i.e. use of successively more refined and more complex language models) cannot be exploited. The incremental decoder has the advantage of fast processing (5x real time) but the word error rates are typically a little higher than those obtained by multi-pass decoding strategies. The language model which was used for the experiments reported in this chapter is a backoff bigram model [72].

Features	Values
voicing	+voice, -voice, sil
manner	stop, vowel, lateral, nasal, fricative, sil
place	labial, coronal, palatal, velar glottal, high, mid, low, sil
front-back	front, back, nil, sil
rounding	+round, -round, nil, sil

Table 4.1: Articulatory features for German.

4.1.3 Acoustic Baseline System

The acoustic baseline system uses 12 MFCC coefficients, one energy coefficient, and the first and second derivatives of these. The total number of features thus is 39. In order to compensate for the different recording conditions a simple channel adaptation is performed by cepstral mean subtraction. The acoustic codebook contains 256 classes, each of which is modeled by a mean vector and a full covariance matrix. The models are linear left-to-right models without skip transitions. The number of HMM states produced by the clustering method described above is 2883, using a clustering threshold of 75 samples per cluster. The codebook was iteratively updated during training.

4.2 Articulatory System

As before, the articulatory system uses a set of parallel MLPs to extract articulatory features from the preprocessed speech signal. The set of features that were employed for the present task is shown in Table 4.1. The features are largely similar to those used for American English in Chapter 2; some features, such as *dental* and *retroflex* are not included since they are not relevant for the definition of German phones. Other features, such as *palatal*, were added. The total number of feature values is 26. The feature transcriptions were derived from the automatic phone-based labeling produced the University of Munich (cf. Section 2.1.1). This system incorporates phonetic pronunciation rules and is reported to achieve an agreement with human labelers of approximately 90% [121]. The phone-feature conversion table is given in Table A.5 in the Appendix. Based on the Numbers95 experiments and some preliminary feature recognition experiments on the present corpus, the number of hidden units was set to 100 and the number of context frames was fixed at nine frames. A set of 10 000 utterances was used for feature training, 1000 utterances were used for cross-validation. The frame-level feature accuracy rates on the test set are given in Table 4.2. The feature probabilities were subsequently concatenated and

Network	Accuracy
voicing	87.39%
manner	81.49%
place	69.65%
front-back	81.37%
rounding	83.25%

Table 4.2: Feature accuracy rates on Verbmobil 96 test set.

used as data for codebook training as described above.

It was found that some difficulties were created by the form of the output distribution of the articulatory feature networks: the final output function in the MLPs is the softmax function (Equation 3.7), which constrains the output values to lie within the range [0,1] and to sum to 1. It thus is frequently the case that one output value is close to 1 whereas all others are close to 0. For this reason, the resulting output distribution has a strongly bimodal character, resembling that of a binary variable. This distribution is not well matched by the Gaussian modeling assumption underlying the design of the codebook. Therefore, the final non-linear activation function was omitted in the MLPs when generating the input data for the second-level classifier, and the pre-softmax values were used instead. This does not have an effect on the classification decisions of the feature networks – the softmax output function is a monotonic function affecting all feature dimensions. Its removal does not change the ranking of the output classes. The resulting values may be interpreted as features in *hidden* space (the space of the MLP weights). Their distribution, though not being strictly Gaussian, is bell-shaped and therefore matches the modeling assumptions better than the bimodal distribution of the probabilities used previously.

The class labels used for training the codebook were identical to those which were used for training the acoustic baseline system. After testing various codebook design choices, the number of classes was fixed at 384, using full covariance matrices. HMM states were initialized using a state-level alignment produced by another MFCC-based system trained on the Verbmobil data. The state clustering step in the HMM training procedure yielded 3359 states, using a clustering threshold of 75. The codebook was updated at each iteration of Baum-Welch re-estimation.

4.3 Recognition Results and Error Analysis

Table 4.3 shows the word error rates on the Verbmobil test set obtained by the MFCC and the AF systems. The word error rate of the MFCC system is lower than that of the AF system by a total of 1.44%. This difference is statistically significant.

System	WER	SUB	DEL	INS
MFCC	29.03%	19.16%	8.32%	1.83%
AF	30.47%	19.31%	9.03%	2.13%

Table 4.3: Word error rates, substitutions, deletions and insertions on the clean Verbmobil test set obtained by the baseline MFCC and AF systems.

As before, we are not (only) interested in the word error rates obtained by the different systems but in differences at a qualitative level, which requires a more detailed error analysis. As mentioned above, the Verbmobil recognition task is much more complex than the Numbers95 recognition task which was the basis of the experiments described in Chapter 3. Not only are we dealing with numerous spontaneous speech phenomena such as disfluencies, strong coarticulation, etc., the system also has a much larger vocabulary, which, compared to a small-vocabulary task, introduces further error sources, such as search errors due to pruning. For this reason it did not seem adequate to limit the error analysis to a comparison of frame-level and word-level error rates. We therefore decided to apply a method which was better suited to evaluating a large-vocabulary recognition system. One such method was developed by Chase [21]. Her method of categorizing speech recognizers' errors is based on comparing the time alignment, language model scores and acoustic scores of each word in (a) the recognition output of the system to be evaluated, and (b) in the forced alignment of the test set transcription and the test data, using the same system. For each word in the two different outputs, the time alignment, the language model score and the acoustic score are recorded. Moving from left to right through each utterance, these outputs are then compared in order to identify so-called *error regions*. When a non-matching segment, i.e. different words or words with a different time alignment[1] is detected, the beginning of an error region has been identified. The error region continues up to the next matching segment. Within each error region, the (normalized) acoustic and language model scores are added up to yield a combined score. Error regions can then be classified depending on how the combined scores of the recognition output (HYP) compare to those of the forced alignment output (REF). There are six possible categories, as shown in Table 4.4. If the combined score of the REF system is better than that of the HYP system, the error region will be classified into one of the cells in the left-hand column. More precisely, it will be placed

- into cell REF-1 if the HYP language model (LM) score is better than the REF language model score and the REF acoustic (AC) score is better than the HYP acoustic score,
- into cell REF-2 if the REF language model score is better than the HYP language model score and the REF acoustic score is worse than the HYP acoustic score, and

[1] allowing for a small frame tolerance of 2 or 3 frames.

	REF total better	HYP total better
AC better	REF AC dominates HYP LM (REF-1)	HYP AC dominates REF LM (HYP-1)
LM better	REF LM dominates HYP AC (REF-2)	HYP LM dominates REF AC (HYP-2)
AC + LM larger	REF AC and LM dominate HYP (REF-3)	HYP AC and LM dominate REF (HYP-3)

Table 4.4: Classification of error regions according to [21].

- into cell REF-3 if both the language model and acoustic score are better in the REF than in the HYP system.

The classification into cells in the right-hand column proceeds analogously for the error regions where the total HYP score is better.

The error categories in Table 4.4 are interpretable with respect to the potential source of the error. If the error region has been assigned to one of the cells in the left-hand column a search error has occurred – both the language model and the acoustic models have favored the right solution, but the right solution was eliminated at some point during the search process. In order to minimize these types of errors, the beam search and pruning parameters of the decoder could be adjusted. The errors in the right-hand column, by contrast, are modeling errors. These occur when either the HYP acoustic score or the HYP language model score or both scores contribute to preferring the incorrect over the correct option. Possible causes for this might be

- confusions between different acoustic models,
- pronunciation variants which are not modeled in the recognition lexicon,
- incorrect reference transcriptions,
- word sequences in the test set classified as highly improbable by the language model, etc.

Chase uses further, more specific, error categories, viz. out-of-vocabulary (OOV) words and homophone substitutions. These are uniquely identifiable error sources which show characteristic patterns. OOV words are words which do not occur in the training set and, as a consequence, are not represented either in the recognition lexicon or in the language model. Unless the recognition system includes a specific strategy for identifying and transcribing OOV words, there is no chance of recognizing these words correctly. OOV words can be explicitly marked in the

reference transcription – if an error region has been found which includes an OOV word it is automatically assigned to the OOV category.

Homophone substitutions may occur when two or more words are present in the recognition lexicon which have the same phone transcription but a different orthography and/or semantics and which are confused in the recognizer's output. Examples of these in the VERBMOBIL recognition lexicon are *das* and *daß* (*that* (determiner) and *that* (conjunction)) or *Meier* and *Meyer* (proper names).

For our purposes we have added a further special category: error regions caused by disfluencies. This category includes all instances of false starts and partial or interrupted words. As in the case of OOV words, these are specially marked in the reference transcriptions, and an error region is automatically assigned to the corresponding category if it contains a disfluency mark.

According to these principles, an error analysis was carried out both for the acoustic and the articulatory system. In computing the language model and acoustic scores, the scores for OOV words were not taken into consideration as they would have distorted the overall relative distance between the HYP and REF scores in the rest the utterance. Furthermore, error regions which occurred solely due to the presence or absence of noise or silence were not considered. The same holds for errors due to confusions between different noise models. In computing the combined HYP and REF scores, the language model weighting factor used during decoding was taken into account. Error regions which contained joint occurrences of OOV words, disfluencies or contractions were multiply classified into the corresponding categories. However, multiple occurrences of the same special category within one error region did not lead to an accumulation of error counts. Thus, although several OOV words may occur in one error region, the entire region was only classified as OOV once. The results of the error analyses are shown in Table 4.5.

Two points should be noted in order to facilitate the interpretation of these results: first, the percentages sum up to a number greater than 100.0 because – as mentioned above – several error regions were multiply classified. Second, the percentage of OOV categories differs among the two systems although in each case the same number of OOV words was present in the test set. This is caused by two factors: (a) an error region may contain several OOV words and is then only classified as OOV once, whereas the same region may be broken down into several error regions in the other system; (b) in both systems a small number of test utterances did not receive an alignment due to pruning problems – however, these utterances are not identical, which may lead to more OOV words being present in one system's output compared to the other.

As can be seen, the error category which has markedly fewer instances in the MFCC system than in the AF system is the category HYP-1. This category groups together all cases where the HYP acoustic score overwhelms the REF language model score. As mentioned above, the potential

	MFCC	**AF**
Total no. of error regions	779	793
OOV	34.92%	31.78%
Disfluency	3.47%	3.15%
Homophones	0%	0%
REF-1	2.70%	3.40%
REF-2	2.82%	3.78%
REF-3	4.49%	4.29%
HYP-1	14.63%	**17.02%**
HYP-2	9.73%	10.84%
HYP-3	29.65%	26.99%

Table 4.5: Classification of error regions in MFCC and AF systems.

causes can be manifold: highly confusable acoustic models, inaccurate pronunciation modeling, incorrect word transcriptions, etc.

In order to further determine the source of the errors in this category, a bottom-up phone-only decoding was performed using monophone models and a phone bigram. The word language model and the recognition lexicon thus did not have any effect on the recognition output. In order to evaluate the phone decodings both the frame error rate (the percentage of frames differing in phone identity in the reference alignment and phone-only decoding) and the distances between the phone sequences were computed. Distance was defined in terms of phonetic features which are similar but not identical to our articulatory features (see Table A.6 in the Appendix). In line with Chase's procedure, the simple Hamming distance between the lists of binary feature-values was used in order to evaluate how strongly the phone sequences diverge. A difference in one feature (i.e. change of two binary feature values) thus incurred a distance value of 2. The phonetic distance values computed for each incorrect frame were added up and divided by the number of incorrect frames. Table 4.6 shows the frame error rates and distance values for the MFCC and AF system, computed separately for different error categories and for all frames in the test set (row six).

The median of the distribution of the AF system's distance values within category HYP-I (sorted into bins ranging from 0 to 8) is at bin 2, i.e. most error regions have an average distance value of 2, which corresponds to a difference in one phonetic feature. 82% of the error regions have a distance value less than or equal to 4, i.e. most error regions differ from the reference in one or two features only. This shows that the greater part of the error regions in this category is phonetically highly confusable with the "true" models, which indicates that the major source for

Error	**MFCC**		**AF**	
Category	FER	dist	FER	dist
OOV	50.2%1	5.05	51.59%	5.58
Disfluency	50.21%	5.78	54.74%	5.63
REF-1	44.34%	6.10	50.79%	6.48
REF-2	39.21%	5.47	52.19%	6.19
REF-3	48.11%	6.08	40.71%	5.13
HYP-1	47.43%	5.34	50.43%	5.26
HYP-2	51.58%	5.03	51.44%	5.73
HYP-3	45.97%	5.43	53.21%	5.79
All data	42.69%	4.92	43.67%	5.21

Table 4.6: Frame error rates (FER) and average phonetic distance values of monophone decodings and reference phone alignments for different error categories and the entire test data (bottom row).

errors in this category may be a lack of discriminability between different acoustic models.

The inferior quality of the acoustic models in the AF system may have several causes. First, the features themselves may not provide sufficient discriminability between the correct class and the set of incorrect classes. Second, the bottom-up acoustic-phonetic modeling accuracy of the articulatory system may be lower than that of the acoustic codebook. Third, there may be a loss of discriminability between different models when expanding the system from monophones to triphones, which may be related to the different results of the automatic state clustering procedure in the two systems.

The first of these possible error sources was investigated by computing a measure of class separability in both the acoustic and articulatory feature space. This is expressed as a discriminant ratio, Q, which is defined as

$$Q = \frac{V^2}{V^2 + D^2} \tag{4.8}$$

where

$$V^2 = \sum_{k=1}^{K} P_k \mathbf{trace}[\Sigma_k] \tag{4.9}$$

and

$$D^2 = \frac{1}{1 - \sum_{k=1}^{K} P_k^2} \sum_{k=1}^{K} \sum_{j=1}^{K} P_k P_j (\mu_\mathbf{k} - \mu_\mathbf{j})^2 \tag{4.10}$$

(see e.g. [111]). V^2 measures the within-class variance of the features with respect to the class means – this is simply the sum of the traces of the class-specific covariance matrices $\Sigma_1, ..., \Sigma_K$, weighted by the class priors. D^2 denotes the inter-class distance, i.e. the distance between class

Measure	**MFCC**	**AF**
WER	29.03%	30.47%
average state entropy	3.23	3.54
frame-level error rate	42.69%	43.67%
discriminant ratio	0.525	0.675

Table 4.7: Measures of accuracy and discriminability at various levels in the recognition system for MFCC and AF recognizers.

means, weighted by the class priors. Q finally is defined as the ratio of within-class distance to the between-class distance and ranges between 0 and 1. A smaller value of Q indicates better separability of the K classes. The Q value for the MFCC feature set was 0.525, compared to 0.675 for the AF system, showing that the MFCC feature set leads to better separability of the phone classes.

The bottom-up acoustic-phonetic modeling accuracy (i.e. the identification of the correct phone class without any information from the lexicon or language model) is already expressed by the frame-level phone accuracies quoted above. An indication of the differences at the level of context-dependent modeling is given by the average entropy of the HMM state distributions. The average state distribution entropy is computed by

$$H_{av}(\mathcal{Q}) = \frac{1}{N}\sum_{i=1}^{M} n_i H(q_i) \tag{4.11}$$

where $\mathcal{Q}$ is the total set of states $\mathcal{Q} = q_1, q_2, ..., q_M$, n_i is the number of training samples assigned to state q_i, N is the total sum of the number of training samples assigned to states,

$$N = \sum_{i=1}^{M} n_i \tag{4.12}$$

and $H(q_i)$ is the entropy of state q_i, which was already defined in Equation 4.7 We observed that a low-entropy distribution indicates high acoustic homogeneity of the training samples assigned to the state – these can be modeled by a small number of mixture components. A high-entropy distribution characterizes states whose training observations are more evenly spread across a larger number of codebook classes. It is thus more desirable to have low-entropy state distributions. The average state entropy is 3.23 in the MFCC system vs. 3.54 in the AF system.

The various separability and accuracy measures are summarized in Table 4.7. The interpretation of these data is that different phonetic classes are less separable in the articulatory feature space than in the acoustic feature space, as evidenced by the discriminant ratio. This entails a higher degree of uncertainty at the level of vector quantization, which in turn produces the higher aver-

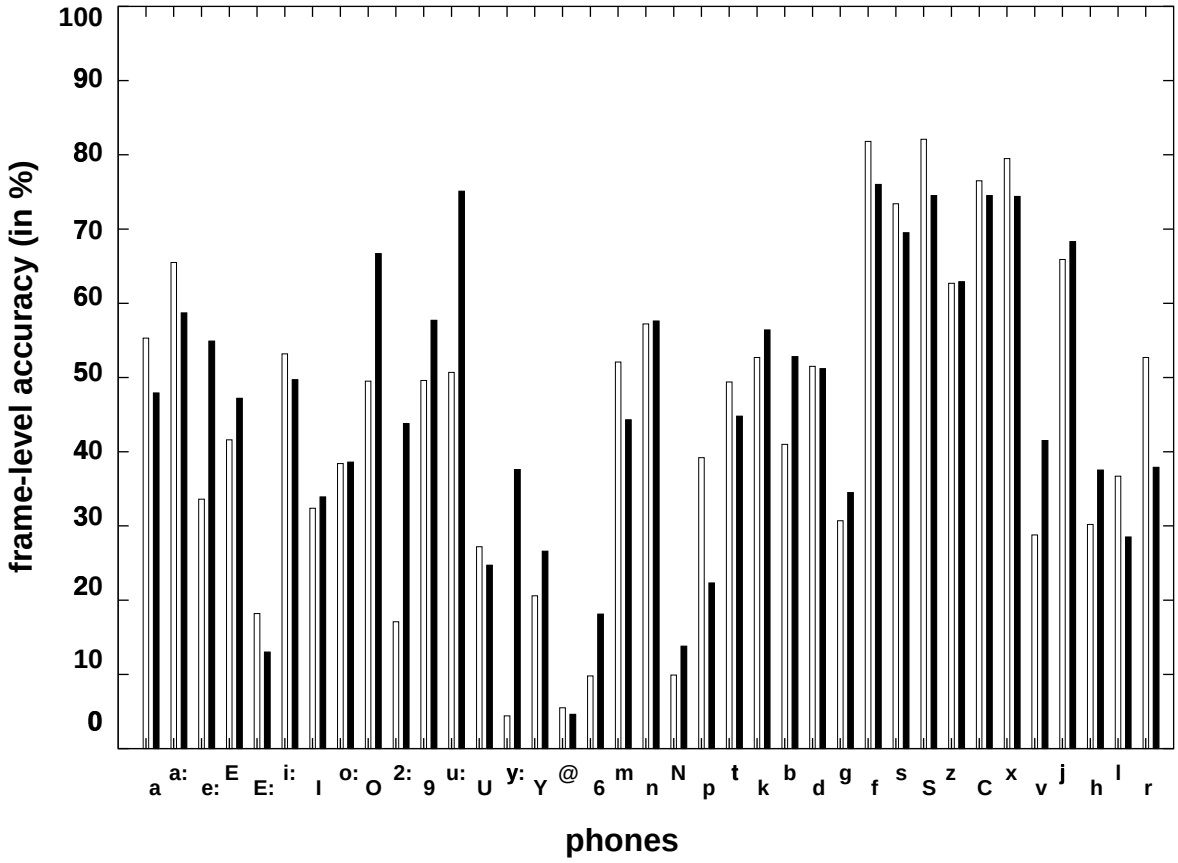

Figure 4.1: Diagonals of phone confusion matrices of AF and MFCC systems. Blank boxes represent the MFCC system, shaded boxes represent the AF system.

age entropy of the state-dependent probability distributions. The consequence is a larger number of confusions between different subword units, and ultimately, a higher word error rate.

A further illustration of qualitative differences is provided by the phone confusion matrices derived from the phone-only decoding described above. Figure 4.1 shows the diagonals of the phone confusion matrices. Again, we can see that the information provided by the two systems is partially complementary in that different phones are classified with varying degrees of accuracy by the two systems.

In order to give an overall quantitative assessment of the differences among the AF and the MFCC system, the percentages of different vs. identical word-level errors were computed. These are shown in Table 4.8. The numbers indicate that about two thirds of the errors made at the word level are not identical in the two systems.

Error type	**%**
both correct	61.44
AF correct, MFCC incorrect	5.99
MFCC correct, AF incorrect	11.52
both incorrect	21.05
different errors	66.66
identical errors	33.34

Table 4.8: Error percentages in the outputs of MFCC and AF recognizers.

4.4 Optimizing the Articulatory Recognizer

We have seen above that the major reason for the poorer performance of the AF system is the overall confusability between phone classes in articulatory space. A number of strategies were investigated in order to enhance the bottom-up classification accuracy and to optimize the articulatory representation.

First, further articulatory features were added to the feature set in order to ensure that certain higher-level phonemic distinctions could be made which were not supported by the original articulatory feature set, i.e. distinctions between tense and lax vowels, such as /u:/ - /U/. This involved adding another feature network with *tense*, *lax*, *nil*, and *silence*. The codebook contained 384 classes and had full covariance matrices. The best word error rate obtained by this system was 30.95%, i.e. no improvement was gained.

Second, in order to reduce the quantization loss in the codebook, the number of codebook cells was increased. Whereas an increase from 256 cells in the first prototype of an articulatory codebook to 384 cells in the current baseline system led to a 1% absolute improvement in word recognition accuracy, no further improvement could be observed when further increasing the number of classes to 512 – on the contrary, the system performance dropped slightly by 0.79%.

As a further strategy for optimizing the articulatory feature space, the dimensionality of the space was reduced by applying Principal Components Analysis. The first 18 principal components were selected, which covered 95.6% of the variance of the data. The size of the codebook was increased in order to match the number of parameters in the articulatory baseline system. The intention of this procedure was to restrict the modeling effort to modeling only the information-bearing components of the articulatory representation. The resulting system, however, showed a slight loss in accuracy – the best word error rate obtained in this experiment was 31.81%.

A further experiment involved the addition of first-order temporal derivatives to the basic articulatory features. Delta coefficients were computed with a window of five frames and added to

the basic articulatory features. However, during the VQ training the delta coefficients had the effect of collapsing many of the initial classes in the codebook (as determined by the initialization labels) into a small number of very large classes, an indication that the delta coefficients were too similar across different classes and thus tended to dominate the basic features in the classification process. The system trained up using deltas achieved a word error rate of 31.42%.

A number of different options were also investigated with respect to HMM initialization and the state clustering algorithm. The initial AF system used hard state-tying in the initialization procedure, i.e. identical states were created for the beginning, middle, and end parts of phonemes. In a different initialization procedure, physically different states were created for the various temporal phases of a phoneme. However, this initialization procedure led to a system with no significant difference in error rate. Finally, different clustering thresholds were tested in order to control the number of distinct hidden states. A lower clustering threshold of 50 led to a larger number of states but also had the effect of slightly reducing increasing the word error rate to 30.80%. A higher threshold of 100 lowered the number of states and decreased the word error rate slightly to 30.21%. No further improvement could be observed by further raising the clustering threshold.

In sum, these standard optimization strategies did not yield any major improvement. It seems likely that in order to improve the accuracy of the AF system, more fundamental changes would have to be made to the modeling approach. For instance, entirely different modeling schemes such as segmental HMMs could be used in order to capture the temporal evolution of articulatory features. Another possibility might be to automatically adapt the lexical representation to the articulatory feature representation, e.g. by clustering the phone classes such that the clustered classes are more easily distinguishable. However, this may in turn lead to increased confusability between different words in the recognition lexicon, so that this procedure would need extensive optimization.

4.5 Combination

Although the AF-based representation does not seem to provide major advantages over the MFCC representation, it does yield information which is partially complementary to that in the MFCC system. This is evident from both the phone confusion matrices and the large percentage of different word-level errors. Although these data do not yield a general phonetic interpretation of the strengths and weaknesses of the different systems, they clearly demonstrate that the MFCC system and the AF system focus on different speech sounds.

Given that the systems make different errors, we should again take a look at various possibilities

of combining their outputs. We will first address the issue of combining systems at the HMM state level before investigating word-level combining strategies. Finally, we look at different methods for combining the articulatory and acoustic representations at the feature level.

4.5.1 State-Level Combination

In the context of a large vocabulary recognition system the procedure of training a higher-level classifier to combine the state likelihoods of the individual systems involves a prohibitively large computational effort. For this reason, we limited state-level combination experiments to the linear combination schemes of the type discussed in Chapter 3. The linear combination rules (repeated here for convenience) were formulated above in terms of posterior probabilities, where $p(\omega_k|\mathbf{x_n})$ was the probability of class k given observation $\mathbf{x_n}$ which serves as input to the $n'th$ classifier.

- product rule
$$P(\omega_k|\mathbf{x_1}, ..., \mathbf{x_N}) = \frac{\prod_{n=1}^{N} P(\omega_k|\mathbf{x_n})}{\sum_{k=1}^{K} \prod_{n=1}^{N} P(\omega_k|\mathbf{x_n})} \tag{4.13}$$
- sum rule
$$P(\omega_k|\mathbf{x_1}, ..., \mathbf{x_N}) = \frac{1}{N} \sum_{n=1}^{N} P(\omega_k|\mathbf{x_n}) \tag{4.14}$$
- max rule
$$P(\omega_k|\mathbf{x_1}, ..., \mathbf{x_N}) = \frac{max_n P(\omega_k|\mathbf{x_n})}{\sum_{k=1}^{K} max_n P(\omega_k|\mathbf{x_n})} \tag{4.15}$$
- min rule
$$P(\omega_k|\mathbf{x_1}, ..., \mathbf{x_N}) = \frac{min_n P(\omega_k|\mathbf{x_n})}{\sum_{k=1}^{K} min_n P(\omega_k|\mathbf{x_n})} \tag{4.16}$$

This formulation was directly applicable to the outputs of the Multi-Layer-Perceptrons in the hybrid system, which, as we noted before, approximate Bayesian posterior probabilities in the least-square sense. By contrast, the Gaussian mixture classifiers in the current recognition system estimate the likelihoods $p(\mathbf{x_n}|\omega_k)$, which are then converted to log-likelihoods for decoding. It should be noted that the likelihood $p(\mathbf{x_n}|\omega_k)$ is dependent on the dimensionality of the feature vector $\mathbf{x}$ – the likelihoods resulting from different systems relying on feature vectors of different dimensionalities will therefore have different ranges, with fewer features typically resulting in larger likelihood values. The likelihoods should therefore be appropriately scaled. We perform this scaling by normalizing by the acoustic likelihood $p(\mathbf{x_n})$, which is simply the sum of the likelihoods $p(\mathbf{x_n}|\omega_k)$ over all classes k:

$$p_{norm}(\mathbf{x_n}|\omega_k) = \frac{p(\mathbf{x_n}|\omega_k)}{\sum_k p(\mathbf{x_n}|\omega_k)} \tag{4.17}$$

If we assume that classes (states) have uniform prior probabilities in the different systems, the combination of normalized likelihoods will be proportional to the combination of the posterior probabilities $p(\omega_k|\mathbf{x_n})$.

The HMM/ANN hybrid systems discussed in the previous chapters had identical numbers of states – every output unit of the phone MLP corresponded to a state in the recognition lexicon. In this case, however, the systems have different numbers of physical states due to the automatic state clustering method described above. In order to be able to combine state emission probabilities we combine emissions for those states which are referenced by the same logical name – thus, for any given state, emissions can be combined although they stem from physically different state definitions. An additional question is raised by the different state transition probabilities in the different systems. We tested the following possibilities:

- using the transition probabilities of the MFCC system,
- using the transition probabilities of the AF system,
- using their average, i.e.

$$a_{ij} = \frac{(a_{ij}^1 + a_{ij}^2)}{2} \tag{4.18}$$

  where a_{ij}^n is the transition probability from state i to state j in the n'th system,

- using their normalized product

$$a_{ij} = \frac{a_{ij}^1 a_{ij}^2}{\sum_{k=1}^{K} a_{ik}^1 a_{ik}^2} \tag{4.19}$$

  where K is the number of different outgoing transitions for state i.

The best results were in each case obtained by taking the normalized product of the transition probabilities. Table 4.9 lists the results of the state-level combination experiments and, for comparison, the baseline recognition results.

As we can see, improvements over the baseline MFCC system are obtained by the *min* rule and the product rule; however, the only significant improvement is the product rule combination result. We thus see our previous observations confirmed, viz. that the optimal combination scheme is based on a product combination of scores.

In the above experiments the scores from both systems were not weighted with respect to each other. However, the individual contributions may be modified by either static or dynamic weights. Several experiments were carried out where a weighted product combination rule was used, i.e.

System	WER	INS	DEL	SUB
AF	30.47%	2.13%	9.03%	19.31%
MFCC	29.03%	1.83%	8.32%	19.16%
product	**27.65%**	2.75%	6.53%	18.38%
max	30.63%	4.84%	5.36%	20.43%
min	28.73%	2.59%	6.94%	19.20%
sum	31.98%	4.24%	5.09%	21.65%

Table 4.9: Word error rates obtained on the Verbmobil test set by the baseline AF and MFCC systems and by different linear probability combination rules.

AF weight	MFCC weight	WER	SUB	DEL	INS
0.1	0.9	29.03%	19.57%	6.22%	3.24%
0.2	0.8	28.84%	18.86%	6.27%	3.26%
0.3	0.7	27.82%	18.47%	6.35%	3.00%
0.4	0.6	27.55%	18.36%	6.35%	2.84%
0.6	0.4	27.65%	18.46%	6.52%	2.67%
0.7	0.3	27.44%	18.43%	6.30%	2.72%
0.8	0.2	**27.41%**	18.39%	6.32	2.70%
0.9	0.1	27.94%	18.74%	6.05%	3.15%

Table 4.10: Word error rates obtained by different weights in a weighted product combination rule.

$$P(\mathbf{x_1}, ..., \mathbf{x_N}|\omega_k) = \frac{\prod_{n=1}^{N} P(\mathbf{x_n}|\omega_k)^{\gamma_n}}{\sum_{k=1}^{K} \prod_{n=1}^{N} P(\mathbf{x_n}|\omega_k)^{\gamma_n}} \qquad 0 \leq \gamma_n \leq 1, \sum_{n=1}^{N} \gamma_n = 1 \tag{4.20}$$

where the weights were found by a search over possible weights between 0 and 1. Several different weighted combinations led to an improvement over the unweighted combination scheme (see Table 4.10).

Most of the results obtained in the weighted combination experiments led to a significant decrease in word error rate compared to the MFCC baseline. However, with respect to the unweighted product scheme, they only showed marginal additional improvements which are not statistically significant. More substantial improvements might be gained if dynamic weights were used instead of static weights. As shown in [70], weighted frame-level classifier combination in speech recognition can benefit from higher-level information, i.e. information about the correctness of the word or utterance. Thus, confidence values derived from the individual systems' recognition passes can be used at a post-processing stage as dynamic state-level combina-

tion weights. Alternatively, combination weights could be associated with words or individual HMM states and could be trained according to a discriminative criterion.

4.5.2 Word-Level Combination

Typically, speech recognition systems show a greater confidence of decision at later stages in the recognition process. A better way of combining decisions from two different systems might therefore be to re-evaluate their joint outputs at the word or utterance level. Two methods of system combination above the state level which have been employed previously are HMM recombination and N-best list rescoring.

In HMM recombination (also referred to as HMM decomposition) [118, 15] two (or more) HMMs are combined by a product operation. Let λ_1 be a HMM with state space $\mathcal{S}_1$, a vector of start probabilities $\pi_{\mathbf{1}}$, a matrix of state transition probabilities A_1, and a matrix of state observation probabilities B_1. Analogously, let λ_2 be another HMM with $\lambda_2 =< \mathcal{S}_2, \pi_{\mathbf{2}}, A_2, B_2 >$. The product HMM is a HMM λ' with $\mathcal{S}' = \mathcal{S}_1 \otimes \mathcal{S}_2$, $\pi' = \pi_{\mathbf{1}} \otimes \pi_{\mathbf{2}}$, $A' = A_1 \otimes A_2$, and $B' = B_1 \otimes B_2$. Thus, the state space of the new HMM is formed by the Cartesian product of the state spaces of the two original HMMs, the components of the vector of start probabilities are products of the corresponding components in the original start probability vectors, and the elements of the matrices A' and B' are products of the corresponding elements in A_1 and A_2 and B_1 and B_2, respectively. Although this method can be used for simple state-level combination (as in [118] and [116]), the original HMMs can also be equated with larger units such as diphones or syllables while disregarding their internal structure, thus enabling higher-level combination. This possibility was used e.g. in [15] for syllable-level combination of subband streams.

Combination by N-best list rescoring (e.g. [124]) involves merging the lists of the top N hypotheses for each utterance output by the different recognition systems. These are combined into a lattice which can be rescored using additional information, such as normalized acoustic scores, language model information, and confidence values.

The first of these methods, HMM recombination, was not suitable for our task. The size of the state space of the HMM produced by HMM recombination increases exponentially with the number of states in the original HMMs. For a large vocabulary system with a large number of internal states and a sizeable search space, this procedure turns out to be computationally too expensive. N-best list merging was not directly applicable either since the present systems are based on a one-best decoder. However, the single best output sequences from the different systems may be combined in a similar way.

The prevalent approach for combining one-best hypotheses from different recognizers is the **R**ecognition **O**utput **V**oting **E**rror **R**eduction (ROVER) algorithm developed by Fiscus [44].

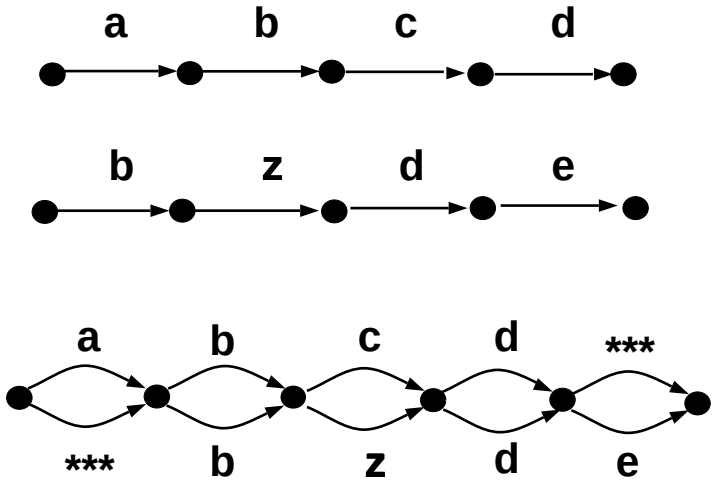

Figure 4.2: Recognizer hypotheses combination by the ROVER method.

ROVER works by constructing a new word transition network from the best word sequences output by a number of different recognition systems, followed by rescoring the new transition network by a voting module. The alignment module arbitrarily designates one word sequence as the reference word sequence. All other word sequences are then iteratively aligned against this reference sequence by means of a minimal-cost dynamic programming procedure. If the two words that have been aligned with each other are both correct, identical copies are made in the resulting word transition network. If they differ, alternative branches are added. Insertions and deletions cause null transition to be added (cf. Figure 4.2). The voting module transitions 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 their respective recognizers. Three different rescoring schemes were tested:
(a) voting by frequency of occurrence,
(b) a weighted sum of frequency of occurrence and average word confidence,
(c) a weighted sum of frequency of occurrence and maximum word confidence.

At each position i in the word transition network, the first method picks that word w which has the largest number of occurrences at position i, $N(w,i)$, relative to the total number of hypotheses, N_{tot}. The other methods compute scores according to

$$Score(w) = \alpha(N(w,i)/N_{tot}) + (1-\alpha)C(w,i), \qquad 0 \leq \alpha \leq 1 \tag{4.21}$$

where $C(w,i)$ is either an average or a maximum confidence score. Null transitions are set to a constant, $Conf(@)$. The values of both $Conf(@)$ and α are trained by minimizing the word error rate on a training set – this is done by a greedy search over the space of possible values for these parameters. The ROVER algorithm was used in the LVCSR 1997 Num 5-E Benchmark Test Evaluations in order to combine the outputs from five different recognition systems. An

wir	müssen	wir	fünf	Tage	zusammen	gekommen
wir	müssen	wieder	fünf	Tage	$\star\star\star$	zusammenbekommen

Figure 4.3: Dynamic-programming alignment of hypotheses from MFCC system and AF system.

wir	[181..191]	wir	[179..189]
müssen	[192..217]	müssen	[190..217]
wir	[218..232]	wieder	[218..232]
fünf	[233..258]	fünf	[233..257]
Tage	[259..287]	Tage	[258..285]
zusammen	[288..324]	zusammenbekommen	[286..365]
gekommen	[325..363]		

Figure 4.4: Temporal alignment of word sequences shown in Figure 4.3.

absolute word error reduction of 5.3% (a relative reduction of 11.8%) was achieved in these experiments.

Although this procedure is simple and intuitive, it has two major drawbacks. One of these is the dynamic programming alignment procedure. This alignment algorithm does not take into account the absolute time alignment of the individual word sequence hypotheses – it finds the minimal-cost match between two strings based on penalty functions for insertions, deletions and substitutions. For this reason, the temporal ordering of hypotheses stemming from different systems may be distorted. Consider the example from the Verbmobil test set shown in Figure 4.3, where the output from the AF based system was string-aligned to the output from the MFCC system, which was used as the reference transcription. Compare this to the actual time alignment of the utterances in Figure 4.4. Here, the start and end times of the word hypotheses are shown in terms of frame numbers. For this utterance the ROVER algorithm would construct a word transition network where the words *zusammen* and *zusammenbekommen* form a sequence. However, as can be seen from the temporal alignment of the different word hypothesis sequences, the two separate words *zusammen* and *gekommen* in the MFCC system cover the same part of the signal as *zusammenbekommen* in the AF system. If the rescoring module happened to choose the sequence *zusammen – zusammenbekommen*, it would incorrectly introduce an insertion. These hypotheses should instead constitute parallel paths in the word transition network in order to prevent this effect.

In the Verbmobil corpus, this is not an isolated example; similar misalignments occur frequently.

One of the reasons is that word formation by compounding is extremely common in German. There are numerous compounds in the recognition lexicon whose components are either identical to, or closely resemble, other (shorter) words in the lexicon. Grammatical inflections are another factor contributing to this effect as they are responsible for a large number of words which are acoustically highly similar and may easily be confused by the recognition system. As a consequence, a longer word may be split up into two or more shorter words which are then incorrectly aligned by the dynamic programming procedure. Further examples of this problem which were observed in the test set are “kurzer Termin” vs. “Kurztermin”, “da wär” vs. “dabei”,“vierzehnten” vs. “vierzehn den”, “dreißigsten” vs. “dreißig denn” etc.

The algorithm which was developed in this thesis is different from ROVER in that it makes use of time alignment information during the construction of the word transition network. Furthermore, unlike ROVER it uses context information (in the form of language model scores) during the rescoring process and is based on a different rescoring formula. In order to describe the algorithm we will first define some general graph-theoretical concepts.[2] In the contexts of speech recognition, the concept of a *word graph* has previously been used with various different interpretations e.g. [94, 2]). For our purposes, we define it as follows:

Definition 2 Word Graph*: We define a word graph $\mathcal{G}$ as a a quadruple $\mathcal{G} =< \mathcal{N}, \mathcal{E}, \mathcal{L}, \mathcal{N} >$, where*

- $\mathcal{N} = \{n_1, n_2, ..., n_x\}$ *is a nonempty set of nodes. Each node is represented by a pair* $< i, t >$*, where* i *is an index and* t *a time point.*

- $\mathcal{E} = \mathcal{N} \times \mathcal{N} \times \mathcal{W} \times \mathcal{L}$ *is a nonempty set of edges. Edges are directed, i.e.* $(n, n', w, l) \in \mathcal{E} \Rightarrow (n', n, w, l) \notin \mathcal{E}$*. If* n *is a start point for edge* e *we say that* e *is* incident from n*. If* n *is an end point for edge* e *we say that* e *is* incident to n*.*

- $\mathcal{L} = \{l_1, l_2, ..., l_x\}$ *is a nonempty set of edge labels. In the current context, these are word hypothesis labels.*

- $\mathcal{W} = \{w_1, w_2, ..., w_x\}$ *is a nonempty set of weights. These typically represent the acoustic scores associated with the word labels.*

Definition 3 Adjacency*: Two nodes* n_i *and* n_j *in a word graph are adjacent if* $\exists e = (n_i, n_j, w_i, l_i) \in \mathcal{E}$*. Two edges* e_i *and* e_j *in a directed graph are adjacent if* $e_i = (n_i, n_j, w_i, l_i)$ *and* $e_j = (n_j, n_k, w_j, l_j)$*, i.e. the end node of the first edge and the start node of the second edge are identical. We denote adjacency by* $n_i \rightarrow n_j$ *and* $e_i \rightarrow e_j$*, respectively.*

[2] See e.g. [20] for an introduction to graph theory.

Definition 4 Path*: A path p in a directed graph is a sequence of edges, $p = (e_1, e_2, ..., e_r)$ such that for any two edges e_i and e_{i+1}, $e_i \rightarrow e_{i+1}$. The length l_p of path p is the number of edges in the path.*

Definition 5 Spanning Edge*: A spanning edge occurs when there exist an edge $e = (n_i, n_m, w_i, l_i) \in \mathcal{E}$ and a path p in $\mathcal{G}$ such that p starts in n_i and ends in n_m and has a length $l_p \geq 2$, i.e. one single edge spans a sequence of several edges.*

There are certain conditions on a word graph which must be fulfilled:

- $\mathcal{G}$ has a distinct start node r, representing the start of the utterance. Thus, $\exists r \in \mathcal{N} \wedge \exists s \in \mathcal{N}$ such that $r \rightarrow s \wedge \forall n \in \mathcal{N}, n \neq r, n \not\rightarrow r$. That is, there exists a node in the set $\mathcal{N}$, which has at least one successor, s, and which is not a successor to any other node.

- $\mathcal{G}$ has a distinct end node f, representing the end of the utterance, i.e. $\exists f \in \mathcal{N} \wedge \exists s \in \mathcal{N}$ such that $s \rightarrow f \wedge \forall n \in \mathcal{N}, n \neq f, f \not\rightarrow n$. That is, there is a node in $\mathcal{N}$ which is a successor to at least one other node and has not successor itself.

- the graph must be *acyclic*. This means that there exists no path p in the graph such that $p = (e_i, e_{i+1}, ..., e_i)$.

- the graph must be *connected*. For all nodes $n_i \in \mathcal{N}$ other than the designated end node f, there is an edge incident from that node.

Let us now turn to the construction of a word graph from the best word sequences output by the individual recognition systems. Unlike the ROVER algorithm, our algorithm simultaneously combines these sequences into a word graph. It is thus not the case that one word sequence is considered the basic reference sequence against which the other is aligned. The word sequences are represented as word lattices, where each entry consists of a word hypothesis label, a start time, an end time, and a score. Both lattices together constitute the lattice database. The word graph is then constructed as follows:

Word Graph Construction from Hypotheses Lattices

1: sort all word hypotheses in the lattice database according to their start times
2: create a root node with start time t = 0
3: **for** each word hypothesis in the lattice database **do**
4: **for** all nodes in the partial word graph constructed so far (starting with the most recently created node and going back in time) **do**

5: check if the node is a possible start node. A possible start node is defined as a node whose time index t_{node} lies within a temporal window w around the start time t_{hyp} of the current word hypothesis. The window w is defined as t_{hyp} +/- n frames, where n is a user-defined parameter.
6: **if** the node is not a possible start node and its time index is larger than $t_{hyp} - n$ **then**
7: continue
8: **else if** the node is not a possible start node and its time index t_{node} is smaller than $t_{hyp} - n$ **then**
9: break
10: **else if** the node is a possible start node **then**
11: attach the word hypothesis as an edge incident to that node.
12: **end if**
13: **end for**
14: **for** all nodes in partial word graph constructed so far (starting with the most recently created node and going back in time) **do**
15: check if the node is a possible end node, in analogy to the procedure in the previous step
16: **if** the node is not a possible end node and its time index is larger than $t_{hyp} - n$ **then**
17: continue
18: **else if** the node is not a possible end node and its time index is smaller than $t_{hyp} - n$ **then**
19: break
20: **else if** the node is a possible end node and no edge with the same label and start node is incident to that node yet **then**
21: attach the current word hypothesis as an edge incident to that node.
22: **else**
23: create a new node in the word graph and attach the hypothesis as an edge incident to that node.
24: **end if**
25: **end for**
26: **end for**
27: create an end node f and edges between the end node and all other nodes in G not having any successor edges
28: **while** two edges a and b exist in the graph which are spanned by a single edge **do**
29: join a and b
30: **end while**

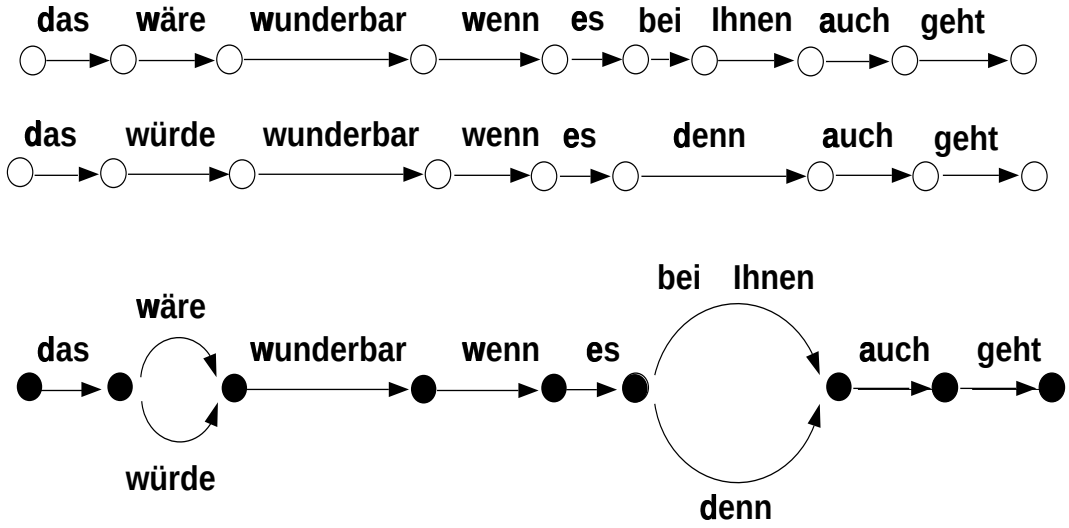

Figure 4.5: Combination of word sequences into word graph.

This algorithm generates a compact representation of competing hypotheses and, moreover, ensures that hypotheses which cover the same temporal interval are represented as alternative paths in the graph. The parameter n determines the density of the graph as it decides, among other things, whether hypotheses with the same label but different start or end times should be collapsed or not. If the difference exceeds n frames, they are represented as two distinct hypotheses, if not, they are collapsed into one. The final step of joining edges spanned by a single edge is performed in order to ensure that all individual edges incident from a node cover the same temporal interval. The join operation can formally be defined as follows: given a path $p = (n_i, n_j, w_i, l_i), ..., (n_{r-1}, n_r, w_{r-1}, l_{r-1}))$ construct an edge with start node n_i and end node n_r, a weight $w = \frac{(w_i + ... + w_{r-1})}{l_p}$ and a label $l = l_i \circ ... \circ l_{r-1}$, where $\circ$ is a concatenation operator. The combination of word sequences into a word graph is schematically shown in Figure 4.5.

A positive side-effect of this combination method is that the complexity of this algorithm is linear as opposed to the quadratic complexity of the dynamic programming alignment algorithm which is used in ROVER. String alignment via dynamic programming has the complexity $O(n^2)$ in time, where n is the length of the longer of the two sequences which are to be aligned. In our algorithm, two basic operations need to be carried out for each entry in the list (of length $n + m$) of lattice entries: finding a start node and finding an end node. In each of these steps, only a small number of word graph nodes in the immediate vicinity of the current time stamp is considered for attaching the word lattice entry – the precise number of node candidates varies but is not dependent on the length of the list of lattice entries. With respect to the complexity of the algorithm, the number of candidate nodes can therefore be considered a constant k. Thus, the time complexity of our algorithm is $O((n+m)k)$, where n and m are the lengths of the two word sequences, respectively. The initial sorting procedure applied to the union of the word lattice

entries can be done in $O((n+m)log(n+m))$ time, thus the entire algorithm is below quadratic complexity. If only two word sequences are considered, this may not be of great significance; however, if more than two word sequences are to be combined, e.g. in the case of combining the outputs from more than two recognizers or when combining N-best sequences, the advantages quickly become obvious.

After the word graph has been constructed, it is traversed in search of the best-scoring path, which is assumed to correspond to the best word sequence. This search is performed by breadth-first graph traversal which, at each node, evaluates all edges incident from that node.

Word Graph Rescoring

1: set i = 0, `string` = $\emptyset$
2: **while** $i \leq N$ (the number of nodes in the graph) **do**
3: **for** each edge e_k incident from $node_i$ **do**
4: compute the score w_k
5: **end for**
6: select the edge e_k with minimum w_k
7: set i to the index of the end node of e_k
8: set `string` = `string` $\circ\ l_k$
9: **end while**
10: return `string` as the best word sequence

In the rescoring experiments the edge scores were computed from the acoustic log-likelihoods output by each system, language model scores (optional) and confidence values (optional). Various normalization procedures had to be applied to the acoustic likelihoods: first, since the decoder outputs acoustic likelihoods which have been accumulated over all frames in the the word, they need to be normalized by the duration of the word. This can be done in the linear domain (i.e. dividing the likelihood by the number of frames in the word) or in the log domain (subtracting the logarithm of the duration). Additionally, the likelihoods from the two different systems need to be normalized since (as explained above) they are not directly comparable – in particular, the likelihoods in the AF system are globally lower than in the MFCC system. This normalization can be achieved by applying scaling factors α and (1 - α) to the likelihoods from the different systems. The values for these scaling factors were determined by a search over all possible values in the range [0,1]. Language model information is represented in terms of the bigram scores at each node transition. The confidence values were computed by generating multiple decodings (a total number of ten decodings was used for the present experiments) of the test set for both the

WER	INS	DEL	SUB
27.97%	2.30%	7.25%	18.43%

Table 4.11: Lowest word error rate obtained using word-level rescoring.

acoustic and articulatory system, respectively, while varying the language model scaling factor ('language model jitter'). The number of times each word in the best-scoring decoding appeared in all other decodings, divided by the total number of decodings, was then used as a confidence value for that word.

The best result was obtained using linearly normalized acoustic likelihoods and language model scores. The optimal scaling factor for the AF and MFCC systems turned out to be 0.7 and 0.3, respectively. The word error rate obtained by this constellation was 27.97%, which is a significant improvement over the MFCC baseline word error rate.

Whereas the use of context information in the form of language model scores generally led to an improvement, it was found that confidence values were less useful. The reason for this may be that the confidence values were derived using a very simple procedure and were not accurate enough in order to further improve results. However, confidence values might turn out to be helpful if more sophisticated confidence estimation methods were used. Furthermore, the rescoring procedure proved fairly sensitive to different normalization schemes applied to the acoustic log-likelihoods. For successful word-level combination it is crucial that the all of the individual factors contributing to the word hypothesis scores, as well as their relative weights, are optimized.

4.5.3 Feature-Level Combination

We have seen in the preceding sections that recognizer combination at the state or word-level involves a fair amount of computational effort for large-vocabulary systems. In each case two complete systems have to be trained before their decisions can be combined. In the case of word-level rescoring the combination procedure takes place at the post-processing stage; thus, in addition to training, two complete recognition passes have to be carried out as well. How can we take advantage of the fact that the AF system provides information complementary to that in the MFCC system without having to go undergo the effort of double training and/or recognition procedures? The obvious choice is to combine the acoustic and the articulatory representations at the feature level, i.e. the recognition system is based on a combined feature space which includes both MFCCs and AFs. However, it would be too impractical to simply use all MFCCs and AFs jointly at this level as this would lead to a prohibitively large feature space of 65 dimensions. We therefore need to select a subset of the union of the sets of MFCC features and articulatory

WER	INS	DEL	SUB
33.30%	2.06%	9.44%	21.79%

Table 4.12: Word error rates obtained by PCA on the joint MFCC+AF feature space.

features.

4.5.3.1 Principal Components Analysis

As a first attempt at reducing the combined feature space while retaining as much information as possible we applied Principal Components Analysis (see Section 3.2.3). The transformation matrix ϕ was derived from the combined MFCC+AF feature vectors – subsequently, the 39 largest principal components were selected for the new system. These covered 97.6% of the variance. A complete system was then trained on the basis of these principal components, using a 256-class codebook with full covariance matrices. Table 4.12 shows the result. The word error rate of 33.30% is significantly worse than either of the word error rates obtained by the baseline systems. An explanation for this might be the negative interaction between embedded training and a representation in terms of principal components which was already observed in Chapter 3.

4.5.3.2 Discriminative Feature Selection

The objective of this study is not only to improve the performance of a speech recognition system by adding articulatory information but also to analyze what information, if any, articulatory features can yield in addition to standard speech features. For this reason, it seemed desirable to additionally apply a selection method which would allow us to retain the interpretations of the individual feature vector components. One such feature selection method was the information-theoretic selection algorithm presented in Section 3.2.3. The selection criterion was to eliminate those features whose elimination changes the overall conditional distribution of the classes Ω given the feature set $\mathcal{F}$, $P(\Omega|\mathcal{F})$, as little as possible. In the subsequent combination experiments, however, we saw that combination works best when the resulting probability distribution is as discriminative as possible, i.e. the true class is distinguished as well as possible from the incorrect classes. How can we define a feature selection procedure that takes account of this fact and selects a feature subset which is maximally discriminative?

One discriminative feature selection method was presented by Bocchieri & Wilpon [14]. The intention of their study was to select the most significant components from a standard feature vector composed of 12 LPC coefficients, their first and second derivatives, plus the first and second derivatives of the frame energy. Under their approach the selection of a feature subset

is based on rank-ordering the features with respect to an optimization criterion and then choosing the N most significant features. The optimization criterion which they employed can be interpreted as the maximization of the ratio of the between-class distance to the within-class distance. The feature selection method was developed within a Gaussian mixture acoustic modeling framework; the distance of observations to acoustic models is therefore expressed in terms of the likelihood computed by evaluating the Gaussian mixtures. More precisely, the distance of a feature vector component x_n to a single Gaussian pdf θ with diagonal covariance[3] is defined as

$$D(x_n, \theta) = E[\frac{(x_n - \mu_n^\theta)^2}{\sigma_n^\theta}] \tag{4.22}$$

where μ_n^θ is the $n'th$ component of the mean vector of θ and σ_n^θ is the $n'th$ component of the variance vector of θ. Thus, the distance of each feature to the model is computed by evaluating the exponent of a one-dimensional Gaussian pdf.

In order to compute these distance values, the training data was aligned to the reference transcriptions at the phone level. For any given phone, the distance values of the feature vector components were evaluated only with respect to the largest (highest-scoring) mixture component of the corresponding phone model. The final rank-ordering according to the distance values then determined the optimal feature set. The application of this method to speech recognition on small corpora like TIMIT and TI digits enabled the authors to reduce the number of parameters by a factor of 2 without any significant loss in word recognition performance.

This feature selection method was directly applicable to our Gaussian mixture based recognition system. In order to apply it to the present task of selecting the most discriminative subset of the MFCC+AF feature set, a simple bootstrap system was trained based on the 65-dimensional combined MFCC+AF feature space. To speed up development, only 256 classes and diagonal covariance matrices were used. An algorithm similar to the feature selection method described above was then applied to a representative subset (about 30%) of the training data using the acoustic models of the bootstrap system. This data subset was aligned at the state level with the sequence of acoustic models based on the reference transcription. The distance values were then computed for each frame, using the parameters of the codebook pdfs and the weights of the HMM state aligned to that frame. The distance values for the correct model vs. the incorrect models at each frame were averaged over the entire data set; finally the 39 features were selected which showed the smallest ratio of correct vs. incorrect distance values.

[3] Only diagonal covariance matrices were used in this study.

The precise selection algorithm is as follows:

Discriminative Feature Selection

for each frame **do**
  for each acoustic model θ **do**
    evaluate the mixture of Gaussians, using the shared codebook and state-dependent weights
    select the largest (highest-scoring) mixture component m
    for each feature vector component n **do**
      compute $D(n, m)$ (as in Equation 4.22)
      if θ is the correct model for this frame **then**
        add $D(n, m)$ to $D(n)_{correct}$
      else
        add $D(n, m)$ to $D(n)_{incorrect}$
      end if
    end for
  end for
end for
select the top k features which minimize

$$\frac{D(n)_{correct}}{D(n)_{incorrect}} \tag{4.23}$$

The application of this selection algorithm with the purpose of selecting the 39 most discriminative features eliminated most of the articulatory features; only the features *vowel*, *fricative*, *labial*, *coronal*, *glottal*, *-round* and *low* were retained. Of the 39 MFCC features, the following coefficients were discarded: the zero'th and second basis coefficients, C_0 and C_2, the first derivatives of the first, second, third and fourth basis coefficients, $\Delta C_1, \Delta C_2, \Delta C_3, \Delta C_4$, and the second derivative of the fourth coefficient, $\Delta\Delta C_4$. The remaining features were then used to train up a combined system with a 256-class codebook with full covariance matrices. The final system achieved a word error rate of 31.08%, which was higher than either of the word error rates obtained by the baseline systems.

There may be various reasons for this increase in word error rate: first, it may be the case that the bootstrap system is not accurate enough to begin with: the combined feature space has 65 dimensions but the codebook only contains 256 classes with diagonal covariance matrices, which may lead to a poor modeling performance. This is also evidenced by the fact that the bootstrap

system achieved a fairly high word error rate of 32.23%. Second, the frequencies of different models were not taken into account explicitly. Since this is a supervised selection algorithm which takes into account information about the model identity, different models could in principle be weighted or de-weighted selectively, e.g. according to their frequency or their information content. If no weighting is used, frequent models contribute more strongly to the selection procedure than infrequent models. This may not be desirable. As an example, consider particular applications, such as word spotting or speech understanding, where it may be necessary to correctly recognize certain infrequent but highly informative words. In these cases the contributions from the acoustic models in these words could be weighted more strongly. On a more general note, models could always be weighted with respect to their relevance for higher-level distinctions. If, for instance, the recognition task consists of a spelling task involving highly similar alphabet letters such as *b*, *c*, *d*, *e*, *g* etc. it is obvious that the distinctive information is provided by the initial consonant. It is therefore important that the feature set is designed such that it can accurately discriminate between these consonants.

A third reason for the drop in performance may be the fact that this feature selection method does not take into account the statistical dependencies between features – although the features may be discriminative when considered in isolation, the resulting overall feature set may be highly redundant. An optimal selection method should therefore be based on a criterion which jointly minimizes redundancy and maximizes discriminability.

Nevertheless, this feature selection method yields some interesting insights into the kind of information which articulatory features may provide in addition to standard MFCC features. Compare the set of articulatory features which were retained by the selection algorithm, *vowel*, *fricative*, *labial*, *velar*, *glottal*, *-round* and *low*, to the plot of the diagonals of the phone confusion matrices (Figure 4.1 on page 88). We can see that the phones which are recognized more correctly by the AF system include the rounded vowels /o,O,2:,9,u:,y:,Y/, and the consonants /k,g,N,b,v,h/. These show a correspondence with the above features in that one or several of these features can be used to define the phones. For instance, it is likely that particular feature values for *vowel* and *-round* are picked out by the system to discriminate between rounded and unrounded vowels. Similarly, the consonantal place features *labial*, *velar*, and *glottal* may help to more accurately identify the phones /k,g,N,b,v,h/.

4.6 Experiments on Noisy Data

In order to verify the observations made in Chapter 3 about the performance of articulatory features in noise, additional experiments on noisy data were conducted using the Verbmobil database. Since noisy test data was not included in the distribution of the corpus, it was artifi-

System	WER	INS	DEL	SUB	Δ
MFCC pink 30 dB	36.93%	1.66%	15.37%	19.90%	+27%
AF pink 30 dB	39.20%	1.58%	15.88%	21.73%	+29%
MFCC pink 20 dB	56.18%	2.06%	26.19%	27.93%	+93%
AF pink 20 dB	58.34%	1.83%	26.54%	29.96%	+91%
MFCC pink 10 dB	82.94%	0.39%	48.46%	34.09%	+185%
AF pink 10 dB	86.54%	0.17%	50.47%	35.81%	+184%
MFCC pink 0 dB	98.76%	0.00%	93.40%	5.36%	+240%
AF pink 0 dB	97.84%	0.00%	88.22%	9.62%	+221%

Table 4.13: Word error rates obtained on the Verbmobil test set, clean speech with added pink noise. Delta values indicate the relative increase in word error rate compared to the results on the clean test set.

cially generated by adding different noise signals from the Noisex database [119] to the clean speech signals. Specifically, pink noise or babble noise (a recording of 100 people talking in a canteen) were added to the signal at various SNRs, viz. 0, 10, 20 and 30 dB.

Training was in each case carried out on clean speech; tests were performed on noisy speech. Other than the usual channel adaptation by cepstral mean subtraction mentioned above, no noise adaptation was performed in either the articulatory or the acoustic system. Tables 4.13 and 4.14 show the word error rates obtained by the different systems across different acoustic conditions, as well as the increase in word error rate relative to the clean test condition.

In general, the word error rates increase drastically with decreasing signal-to-noise ratio. Whereas the increase is moderate at a high SNR (30 dB), system performance declines rapidly at 10 or 0 dB SNR. In the latter two cases the large number of deletions vs. the small number of insertions is explained by the fact that many of the actual reference words are replaced by noise models in the recognition output, which are not taken into consideration by the dynamic programming scoring module.

We can see that the AF system performs worse than the MFCC system in almost all test conditions. Under highly noisy conditions (0 and 10 dB SNR) the AF system shows a smaller relative increase in word error rate compared to the clean test condition than the MFCC system, both on pink noise and babble noise. It should be observed, however, that both systems exhibit a very poor performance in these cases in general; the extremely high word error rates of over 90% indicate that hardly anything has been recognized correctly. Thus, it is questionable if the difference in relative degradation is meaningful at all in this context.

Finally, let us take a look at how a combined system performs under these conditions. Since

System	WER	INS	DEL	SUB	Δ
MFCC babble 30 dB	30.08%	1.71%	8.74%	19.64%	+4%
AF babble 30 dB	32.57%	1.80%	10.42%	20.35%	+7%
MFCC babble 20 dB	40.37%	2.53%	13.85%	23.99%	+39%
AF babble 20 dB	41.77%	1.61%	14.72%	25.44%	+37%
MFCC babble 10 dB	65.57%	1.01%	30.78%	33.78%	+125%
AF babble 10 dB	74.68%	0.54%	38.95%	35.19%	+145%
MFCC babble 0 dB	93.95%	0.06%	73.38%	20.51%	+223%
AF babble 0 dB	96.03%	0.00%	81.62%	14.41%	+215%

Table 4.14: Word error rates obtained on the Verbmobil test set, clean speech with added babble noise. Delta values indicate the relative increase in word error rate compared to the results on the clean test set.

System	WER	INS	DEL	SUB
pink 30 dB	**35.53%**	2.17%	13.24%	20.12%
pink 20 dB	**54.32%**	2.81%	22.57%	28.94%
pink 10 dB	**81.08%**	0.64%	45.28%	19.56%
pink 0 dB	98.45%	0.00%	92.41%	6.04%

Table 4.15: Word error rates obtained on the Verbmobil test set, clean speech with added pink noise, state-level combined system.

state-level combination yielded the best results on the clean test set this method was also used for combination experiments on the noisy test sets. As before, normalized likelihoods were combined by means of a weighted product rule, using a weight of 0.8 for the MFCC system and 0.2 for the AF system. Tables 4.15 and 4.16 present the word error rates obtained by the combined acoustic/articulatory system. Those word error rates which constitute significant improvements compared to the better of the two baseline systems are shown in boldface. We can see that, with the exception of the 0 dB SNR pink noise test condition, the combined system always achieves a significant reduction in word error rate.

System	WER	INS	DEL	SUB
babble 30dB	**28.41%**	2.56%	6.95%	18.89%
babble 20dB	**34.96%**	2.81%	9.41%	22.74%
babble 10dB	**64.05%**	1.58%	27.21%	35.53%
babble 0dB	**93.09%**	0.06%	74.32%	18.71%

Table 4.16: Word error rates obtained on the Verbmobil test set, clean speech with added babble noise, state-level combined system.

4.7 Summary and Discussion

In this chapter we have presented an articulatory feature based recognizer for a large-vocabulary conversational recognition task, viz. the German Verbmobil corpus. Contrary to the study presented in the previous chapter, which was carried out in the hybrid HMM/ANN modeling paradigm, the experiments in this chapter were based on a tied-mixture HMM acoustic modeling approach. We have seen that the AF system is capable of achieving a recognition performance close to that of the MFCC baseline system (30.47% vs. 29.03%); however, the difference of 1.44% was statistically significant. The error analyses applied to the two systems showed that the major cause of this difference seemed to be the larger number of confusions between different acoustic models in the AF system. Further analyses indicated that the cause of these confusions may be the poorer separability of phonetic classes in articulatory feature space compared to the acoustic feature space, which naturally influences the capability of the system of discriminating between different models at higher levels, i.e. phone and word recognition.

What are the reasons for the fuzzier phone class distributions in articulatory space? One potential cause may be the high degree of quantization inherent in the articulatory feature representation. The articulatory feature networks map the preprocessed acoustic signal (which, in this case, has 39 dimensions) to a set of 26 broad articulatory classes. This quantization is in some cases certainly inadequate: vowels, for instance, are classified as either *high*, *mid*, or *low*, *front* or *back*, and *rounded* or *unrounded*. Considering the wide range of possible vowel spectra, this classification may be too coarse to preserve all the information encoded in the acoustic representation. Obviously, this loss of information did not crucially affect recognition performance on the small-vocabulary task described in Chapter 3. However, in the case of large-vocabulary recognition is seems to be important to use a more fine-grained representation which enables the higher-level recognition modules to maintain distinctions between a large number of lexical items. In view of the strongly simplified articulatory speech signal representation, it even is somewhat surprising that the AF system still achieves a reasonable recognition rate – however, this may be due to the high accuracy of the feature detection networks which yields a fairly robust, although not

necessarily discriminative, representation.

In spite of the fact that the articulatory representation does not seem to be sufficient in itself for large-vocabulary speech recognition (at least within the predominant Gaussian mixture/HMM modeling paradigm), we have seen that it does provide information which is not contained in the standard MFCC representation. From the phone confusion characteristics and the discriminative feature selection experiment we can tentatively conclude that this type of information is related to place of articulation features as well as features describing lip rounding. It is interesting to note that most of these features, e.g. *velar* and *glottal*, are heavily context-dependent – it may thus be the case that the use of temporal context at the first classification level in the AF system leads to more reliable scores for these features and thereby to a more accurate identification of the phones which they characterize.

The combination of the AF and MFCC based recognition systems at the HMM state level and at the word level yielded significant improvements over the MFCC baseline system in both cases. The absolute improvements were 1.62% (state-level combination) and 1.06% (word-level combination), respectively. By contrast, the feature-level combination methods which were investigated, PCA and discriminative feature selection both led to significant increases in word error rate. In addition to the possible explanations mentioned above (interactions between PCA and embedded training, feature selection without taking account of the statistical dependencies between features, etc.), some problems might be posed by the joint distributional modeling required when systems are combined at the feature level. MFCC features have near Gaussian distributions which can be modeled well with a moderately- sized Gaussian codebook. In the joint feature space, by contrast, the class distributions may look more complicated and require a larger number of mixture components than was used in the experiments described above.

Chapter 5

Conclusions

In this chapter we will give a summary of the work presented in the course of this thesis and evaluate the results in the light of our initial hypotheses about the use of articulatory features in speech recognition. We conclude with suggestions about possible future work in this area of speech research.

5.1 Summary and Discussion

We began this thesis by describing the various shortcomings of current state-of-the-art speech recognition systems, viz. lack of noise robustness, inaccurate bottom-up acoustic modeling, sensitivity to conversational speech phenomena, and rigid language model constraints. We then expounded the articulatory feature based approach to acoustic modeling and explained its possible contributions to solving the first two of these problems. In particular, we mentioned its potential for more accurate bottom-up statistical classification, the possibilities if offers for selective processing of different aspects of speech sounds, as well as for improved coarticulatory modeling, and, finally, the greater robustness of articulatory features towards speaker variability and noise.

Previous approaches to employing articulatory representations in speech recognition were discussed and evaluated. It was found that, to a large extent, many of these approaches were abandoned prematurely – specifically, they failed to address the potential of articulatory representations in adverse acoustic environments and their application to large vocabulary speech recognition. In addition to this, they did not provide detailed analyses of the characteristic differences between articulatory and standard acoustic speech representations, and, as a consequence, did not offer principled strategies for combining them. These questions were subsequently addressed in two application studies.

In the first study, carried out within the hybrid HMM/ANN paradigm, we described the application of the articulatory approach to a small-vocabulary continuous numbers recognition task

(OGI Numbers95) on band-limited (telephone) speech. In addition to the unmodified clean test set, reverberant speech and speech corrupted by pink noise were used as test conditions. We observed that the acoustic and articulatory systems did not exhibit any significant quantitative differences in clean conditions; in reverberant and noisy speech, however, the articulatory system superseded the acoustic baseline system. The relative improvements over the baseline system in highly noisy conditions (10 dB and 0 dB SNR), 8.3% and 13.1%, respectively, were statistically significant.

Furthermore, it was shown that the acoustic and articulatory systems extract different information from the speech signal. Although these differences were not amenable to an overall interpretation in phonetic or articulatory terms, they demonstrated the potential for recognizer combination. Various frame-level combination strategies were investigated. It was found that the best results in terms of the tradeoff between training/testing requirements and recognition rate were obtained by simple linear combinations of the posterior phone probabilities at the frame-level. Of the four combination schemes which were tested (product, averaging, *max*, *min*), the product and the *min* schemes turned out to be the most successful. An analysis showed that these rules enhance the discriminability of the phone classifier most by producing sharp, low-entropy phone probability distributions in the case of correct decisions and flatter, high-entropy distributions in the case of incorrect classifications. This in turn led to a better differentiation of correct and incorrect hypotheses at the word level. Significant improvements of the word error rate were obtained across all acoustic conditions by product rule combination.

The second application study was concerned with large-vocabulary conversational speech recognition (the German Verbmobil corpus). This study differed from the previous application in terms of the language (German vs. American English), the vocabulary size (5300 vs. 32), the speech signal quality (full-bandwidth vs. telephone speech), the speech mode (spontaneous dialogues vs. continuous numbers), and the recognition system (tied-mixture HMMs vs. hybrid HMM/ANN modeling). The word error rates of the acoustic and articulatory baseline systems were fairly close; however, the acoustic system exceeded the articulatory system by 1.44%, which was statistically significant. A subsequent error analysis revealed that most of the errors in the articulatory system were caused by confusions between different acoustic models. Again, it was shown that the information provided by the articulatory system was partially different from that in the acoustic system.

State-level combination techniques similar to those used in the pilot study yielded a significant improvement over the acoustic baseline of 1.62%. Additionally, word-level and feature-level combination schemes were investigated. The word-level combination scheme was based on combining the best word sequences emitted by the acoustic and articulatory systems into a word graph and searching the best path among these hypotheses. This technique also led to

Corpus	Acoustic Test Condition	Acoustic System	Articulatory System	Combined System
Numbers95	clean	8.4%	8.9%	7.3%
	reverberant	24.7%	23.7%	21.1%
	pink noise, 30 dB	17.2%	17.4%	15.1%
	pink noise, 20 dB	22.8%	21.7%	18.8%
	pink noise, 10 dB	32.7%	30.0%	28.3%
	pink noise, 0 dB	50.2%	43.6%	41.6%
Verbmobil	clean	29.03%	30.47%	27.41%
	babble noise, 30 dB	30.08%	32.57%	28.41%
	babble noise, 20 dB	40.37%	41.77%	34.96%
	babble noise, 10 dB	65.57%	74.68%	64.05%
	babble noise, 0 dB	93.95%	96.03%	93.09%
	pink noise, 30 dB	36.93%	39.20%	35.53%
	pink noise, 20 dB	56.18%	58.34%	54.32%
	pink noise, 10 dB	82.94%	86.54%	81.08%
	pink noise, 0 dB	98.76%	97.84%	98.45%

Table 5.1: Summary of quantitative results.

a significant, albeit somewhat smaller, improvement over the acoustic baseline of 1.06%. The main problem with this method turned out to be the normalization of the acoustic likelihoods which formed part of the word hypothesis scores in the combined word graph. Feature-level combination involved (a) Principal Component Analysis, as a standard feature-space reduction and optimization technique, and (b) a discriminative feature selection algorithm. Both methods led to an increase in word error rate.

Finally, the performance of these systems was tested on variants of the clean test set corrupted by added pink or babble noise. It was found that the performance of the acoustic system exceeded that of the acoustic baseline system in most cases. A system combining model scores at the state level, however, obtained significant improvements over the better of the two baseline systems in almost all conditions. Table 5.1 summarizes the most important quantitative results obtained in this thesis.

Let us review our initial hypotheses in the light of the experimental evidence we have gathered.

The first of these hypotheses was that a cascaded classification scheme with a set of articulatory feature classifiers at the first level and a combining module at the second level should lead to a better classification accuracy of the entire the bottom-up acoustic modeling component.

This prediction was shown to be largely correct in the first application study. All articulatory recognizers achieved a higher accuracy than the single-step phone classifier. Although this did not lead to a higher phone classification accuracy of the overall articulatory classifier in clean conditions, it did produce a significantly higher classification accuracy in reverberant and noisy conditions.

No conclusive evidence in support of the effectiveness of the decompositional modeling approach could be gained from the large-vocabulary experiments. This may have several reasons: as mentioned in the beginning, the success of this approach depends on a variety of factors, such as the accuracy of the individual first-level classifiers, the correlation among their errors, and the sensitivity of the higher-level combination module to estimation errors in the lower-level classifiers. An important difference to the experiments carried out in the hybrid modeling paradigm is the type of higher-level classifiers, i.e. a Gaussian mixture classifier as opposed to an MLP. In the case of an MLP, the weights for each input feature (i.e. for each articulatory feature score) can be adjusted directly with respect to the class discriminant functions. In the Gaussian mixture classifier, by contrast, the individual features are not weighted according to this criterion. Instead, the classifier attempts to most closely approximate the distribution of the articulatory features scores by a mixture of normal distributions where all first-level classifier outputs contribute equally to computing the likelihoods of individual mixture components.

The second hypothesis was that the articulatory approach should provide for greater robustness in noise. This assumption is not entirely separable from the first hypothesis as greater robustness may also be induced by decompositional classification. Further contributing factors, however, might be the use of temporal context at the lowest classification level, the greater insensitivity of articulatory features to noise *per se*, as well as the focus on relative as opposed to absolute frequency patterns. Again, we saw this hypothesis confirmed in the numbers recognition pilot study but not in the large-vocabulary study. However, it was the case in both studies that a combination of the acoustic and articulatory representations led to improvements over the acoustic baseline in noisy conditions, showing that some of the articulatory information is beneficial in noise.

The major goal then was to identify the kind of information provided by the AF system in addition to standard speech features. This is important in so far as knowledge about this information may result in a more accurate and specialized design of feature detectors or in an improvement of, or extensions to, current preprocessing techniques with the goal of incorporating this information. To this end we applied a discriminative feature selection algorithm with the objective of selecting that subset of the combined acoustic and articulatory feature set which would provide the best discrimination among phone models. Although most articulatory features were discarded by this process, seven features were retained, most of which describe consonantal

place of articulation categories. It was noticeable that most of these features were correlated with those phones which were also classified more accurately in the baseline AF system. These conclusions are, naturally, tentative and need to be corroborated by further experimental evidence. However, it seems very likely that the information which is provided by the AF system involves information about highly context-dependent aspects of speech sounds which cannot be distinguished very well on the basis of individual acoustic feature vectors. Two important factors may be the inclusion of temporal context at the level of articulatory feature probability estimation (which may be more effective than simply including delta coefficients), or the focus of articulatory feature classifiers on relative as opposed to absolute frequency information.

So far, most of the work on articulatory representations in ASR has been directed at extracting articulatory features or parameters from the raw or preprocessed acoustic signal. The behavior of these parameters in a complex speech recognition system, by contrast, has been studied less intensively, if at all. This thesis is, to our knowledge, the first comprehensive study which has tested and analyzed articulatory feature based recognition *systems* across different languages, different recognition tasks, different modeling paradigms, and different acoustic conditions. The main conclusions to be drawn from these analyses is that

- it is possible for articulatory feature based systems to achieve a performance comparable to that of state-of-the art acoustic systems,
- in certain deteriorated acoustic conditions (telephone speech, noise) and on small vocabulary they may show a distinctly superior performance,
- articulatory features provide information which is partially complementary to the information provided by commonly used acoustic representations,
- in most cases, the combination of acoustic and articulatory feature representations leads to a better system performance.

These facts should be sufficient to put articulatory representations “back on the map” for automatic speech recognition.

5.2 Future Work

There remain a number of issues which could not be entirely resolved within the scope of this thesis and which deserve additional research. We will discuss each of these in turn, starting with the “low-level” aspects of the AF system and proceeding to the top level. At the lowest level, i.e. at the level of mapping the acoustic signal to articulatory feature scores, several things might

be improved. As we already noted before, specialized acoustic feature extractors may be used as front-ends for different articulatory classifiers. The classifier for voicing features, for instance, might benefit most from a front-end which computes e.g. the zero-crossing rate, normalized log energy or the ratio of the energy in low vs. high frequency bands. Various speech analysis studies have examined combinations of specialized feature extractors with statistical classifiers for certain articulatory features, such as voicing [8, 52], nasality [101], or vowel height [9]. The feature detection error rates reported in these studies generally are below 5%. It should be noted, however, that very small databases were used for these experiments, e.g. 30 speaker-dependent sentences from the Resource Management database in [52]. Similarly low error rates might therefore not be obtained if these or comparable methods were applied to the corpora used in this thesis. Nevertheless, it is reasonable to assume that specialized feature extraction front-ends will improve feature recognition. At the same time, speech enhancement or noise compensation algorithms could be applied selectively to different feature classifiers at this level.

Further optimizations can also be made to the classifiers chosen to extract articulatory features. Although MLPs have shown a good performance across all recognition conditions investigated in this study, there may be some features, especially in the place feature group, which are not amenable to classification by an MLP. More powerful classifiers, such as Support-Vector-Machines [117], recently applied to the detection of phonetic features by [92], might yield better results since they can be trained in such a way as to find global instead of local minima with respect to the error criterion.

Even with respect to our MLP classifiers not all possibilities of optimization were exhausted. In particular, no restriction was made on the continuity of the output of a given MLP across time. An explicit continuity constraint could be integrated in the form of a different objective function used during training. As mentioned in Chapter 3, the objective function we use is the mean-squared error between the network output x and the target y:

$$MSE(x, y) = E[(x - y)^2] \tag{5.1}$$

Another criterion could be used instead which seeks to jointly minimize the mean-squared error between the network output and the target and the mean-squared error between the current output and the previous (or n previous) output(s). Both these components could additionally be weighted, leading to the following objective function

$$F(x_t, y_t, x_{t-1}, ..., x_{t-n}) = E[\alpha(x_t - y_t)^2 + \beta \sum_{i=1}^{n} (x - x_{t-i})^2] \tag{5.2}$$

where n is the temporal window on the network output x, x_t and y_t are output and target at time t, and α and β are the weights for the different terms of the function. An objective function of this form might prevent strong frame-to-frame oscillations of the MLP outputs. A similar smoothing

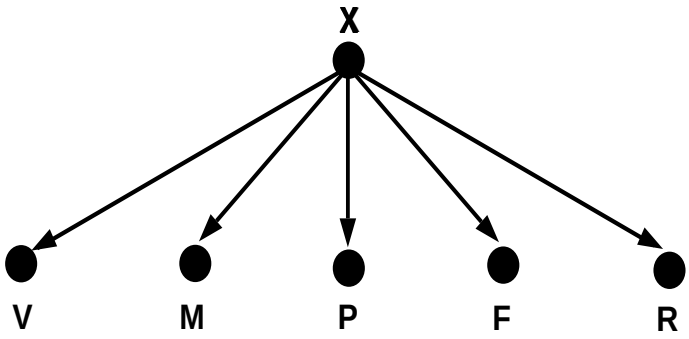

Figure 5.1: Simple Bayesian network structure for acoustic/articulatory dependencies.

function is also used in articulatory codebook lookup in the context of transformation-based acoustic-articulatory mapping [100, 104].

Throughout this thesis we have been using a parallel arrangement of neural network feature classifiers without any explicit dependency relations between the different networks or between individual features. The reason for this was that the higher-level classifier should be able to learn any restrictions on feature co-occurrences automatically during training. However, statistical dependency relations between individual features or entire feature groups might be exploited when estimating the articulatory feature probabilities themselves. For instance, the probabilities of the features in the *rounding* group might be dependent on those in the *voicing* group, or the *manner* features might be dependent on *place* features. This can conveniently be expressed using a graphical dependency model, where nodes represent random variables and (directed) arcs represent conditional dependence relations between these variables [98, 65]. An arc going from node A to node B means that B is conditionally dependent on A. Consider first Figure 5.1. The root node in the graph is associated with a random variable X, representing the acoustic feature vector. The other variables, V,M,P,F, and R represent random variables for the outputs of the *voicing*, *manner*, *place*, *front-back* and *rounding* network, respectively. In Figure 5.1, all network outputs depend on the acoustic variable X but not on each other; they are *conditionally independent* of each other given X. In Figure 5.2, however, additional dependencies between the articulatory variables have been introduced, so that the *place* variable P, for instance, is now dependent on both M and X. These additional dependencies can be specified heuristically, or they can be learnt in a data-driven way, using standard model selection techniques (see e.g. [59, 58] for Bayesian Network structure learning). If multiple dependencies were taken into account, articulatory feature probability estimation might become more robust.

As we saw in Chapter 3 the feature-phone mapping benefits from large temporal contexts. This suggests that units larger than phones should be used for the integration of AF probabilities. An obvious candidate would be the syllable. However, there are a number of problems associated

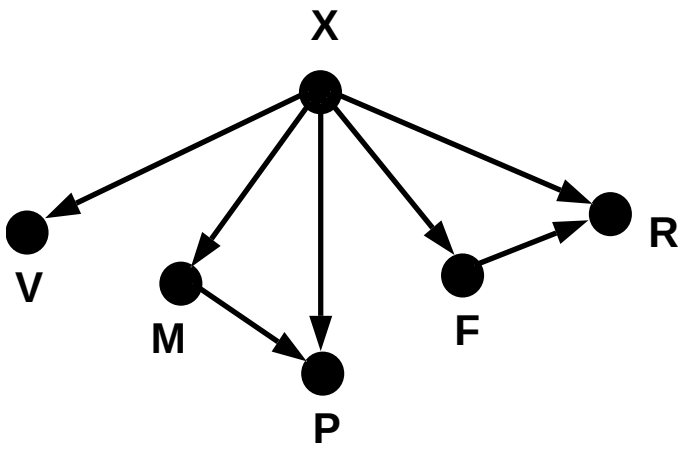

Figure 5.2: Bayesian network structure with added dependencies between articulatory variables.

with using syllable-sized units in statistical speech recognition. The most severe of these problems is that larger subword units go hand in hand with a reduced amount of training material for each unit, leading to undertraining of the majority of models. This problem can be circumvented by using a combined system which uses syllable-sized models for only the most frequent syllables and phone models elsewhere [50]; however, this already constitutes a severe compromise with respect to the temporal modeling power. A further possibility might be the use of diphones as a unit intermediate between phones and syllables, or shifting to an entirely different modeling paradigm, such as segmental models [95]. A segmental model tries to approximate the joint distribution of a variable length-sequence of feature vectors $\mathbf{x_1}...,\mathbf{x_n}$ given the length n and some model s:

$$p(\mathbf{x_1},...,\mathbf{x_n}|n,s) \tag{5.3}$$

This can be represented as

$$b_{s,n}(\mathbf{x_1^n})p(n|s) \tag{5.4}$$

i.e. it can be factorized into the observation probability $b_{s,n}(\mathbf{x_1^n})$ for the sequence of feature vectors $\mathbf{x_1}...,\mathbf{x_n}$ and the duration distribution $p(n|s)$. Segmental models are not restricted to any particular type of unit; thus, s could be a syllable model and $\mathbf{x_1}...,\mathbf{x_n}$ could be a sequence of articulatory feature vectors.

As far as the combination of acoustic and AF representations, or of models based on these representations, is concerned, it has become clear that discriminative combination algorithms offer the greatest potential for successful combination. In the context of state-level or word-level combination, combination weights could be used which are trained discriminatively, in order to minimize the classification error or to maximize the distance between models' (state, word, sentence models) posterior probabilities. Discriminative training algorithms have recently been studied in some detail [10, 22, 109]. It has been shown that, in the majority of cases, they lead

to an improvement over conventional Maximum-Likelihood training. Thus, it is to be expected that these methods would also yield good results when applied to training combination weights.

It has become clear in the course of this thesis that one major advantage of the AF approach might be the decompositional classification aspect, which leads to greater statistical robustness in certain situations, such as acoustically deteriorated speech. However, the particular approach we have suggested is suboptimal in that it makes heuristic assumptions about the type of subphonemic components that are extracted from the acoustic signal at the first classification stage: we are assuming the existence of certain articulatory components based on our knowledge about human speech production. However, as evidenced by the comparatively low detection rates for certain articulatory features as opposed to others (e.g. the best detection rate for *dental* was 57.03% , compared to 79.78% for *velar* on the Numbers95 corpus) these assumptions may not always correspond closely to reality. The same criticism can in principle be leveled against the other decompositional approach to acoustic modeling we mentioned in this thesis, viz. the subband approach, which incorporates heuristic assumptions about the number and bandwidths of frequency subbands. The optimal solution would be a form of *data-driven decompositional acoustic modeling* – the number and the type of subphonemic components should be extracted from the data rather than specified in advance. This could be done, for instance, by using a multiple clustering procedure to arrive at subphonemic "features" or classes. The idea is to subject the acoustic data to several parallel unsupervised clustering procedures which are distinguished by different clustering criteria or different initial transformations of the data, such as different feature extractions algorithms, or filters with different temporal resolutions. The resulting clusters would then be assigned abstract identifiers (e.g. numbers) and the data would be relabeled in terms of these identifiers. A set of classifiers could then be trained on these "features" and their outputs could be combined in analogy to the AF or subband approach. The recognition of these "features" should be fairly accurate because they are known to form clusters in the input space. If the number of final classes in each codebook derived by clustering is smaller than the number of subword units, advantage can be taken of the same data-sharing properties which are characteristic of articulatory features. Thus, this method would combine the advantages of being able to exploit training data in an optimal way and being able to focus on classes which are known to have corresponding clusters in the input space. Potentially, this type of clustering could also be applied across different acoustic conditions to yield a "feature" set which is maximally robust in the presence of highly variable input data. This or other ways of detecting the salient properties of speech in a self-organized, data-driven way may eventually become the method of choice for devising robust classification schemes.

Chapter A

Appendix

zero	fifteen
oh	sixteen
one	seventeen
two	eighteen
three	nineteen
four	twenty
five	thirty
six	forty
seven	fifty
eight	sixty
nine	seventy
ten	eighty
eleven	ninety
twelve	hundred
thirteen	uh
fourteen	um

Table A.1: Word list of the Numbers95 corpus

Phone	**Features**	Description
i:	+voice, vowel, high, front,-round, -central	M**ie**te
I	+voice, vowel, high, front,-round, +central	M**i**tte
y:	+voice, vowel, high, front, +round, -central	H**ü**te
Y	+voice, vowel, high, front, +round, +central	H**ü**tte
e:	+voice, vowel, mid, front, -round, -central	B**ee**t
E	+voice, vowel, low, front, -round, +central	B**e**tt
E:	+voice, vowel, low, front, -round, -central	R**ä**te
a:	+voice, vowel, low, back, -round, -central	R**a**t
a	+voice, vowel, low, back, -round, +central	R**a**tte
6	+voice, vowel, low, -round, +central	Rett**er**
@	+voice, vowel, mid, -round, +central	Ratt**e**
u:	+voice, vowel, high, back, +round, -central	M**u**t
U	+voice, vowel, high, back, +round, +central	M**u**tter
o:	+voice, vowel, mid, back, +round, -central	B**oo**t
O	+voice, vowel, mid, back, +round, +central	M**o**tte
p	-voice, stop, labial	**P**ein
b	+voice, stop, labial	**B**ein
t	-voice, stop, coronal	**T**ank
d	+voice, stop, coronal	**D**ank
k	-voice, stop, velar	**K**uß
g	+voice, stop, velar	**G**uß
Q	-voice, stop, glottal	
f	-voice, fricative, labial	**v**ier
v	+voice, fricative, labial	**w**ir
s	-voice, fricative, coronal	Ro**s**e
z	+voice, fricative, coronal	Ro**ß**
S	-voice, fricative, palatal	**Sch**ule
Z	+voice, fricative, palatal	In**g**enieur
C	-voice, fricative, high	i**ch**
j	+voice, fricative, high	**j**a
x	-voice, fricative, velar	a**ch**
h	-voice, fricative, glottal	**h**allo
m	+voice, nasal, labial	**m**ein
n	+voice, nasal, coronal	**n**eun
N	+voice, nasal, velar	Gesa**ng**
r	+voice, fricative, velar	**r**ot
l	+voice, lateral, coronal	Hal**l**e

Table A.2: Phone-feature conversion table for German. Phone transcriptions are in SAMPA notation.

Phone	Features	Phone	Features
b	+voice, stop, labial, nil, nil	m	+voice, nasal, labial, nil, nil
⋆ d	+voice, stop, coronal, nil, nil	em	+voice, nasal, labial, nil, nil
g	+voice, stop, velar, nil, nil	⋆ n	+voice, nasal, coronal, nil, nil
p	-voice, stop, labial, nil, nil	nx	+voice, approximant, coronal, nil, nil
⋆ t	-voice, stop, coronal, nil, nil	ng	+voice, nasal, velar, nil, nil
⋆ k	-voice, stop, velar, nil, nil	en	+voice, nasal, coronal, nil, nil
dx	+voice, stop, coronal, nil, nil	⋆ l	+voice, lateral, coronal, nil, nil
bcl	+voice, stop, labial, nil, nil	el	+voice, lateral, coronal, nil, nil
⋆ dcl	+voice, stop, coronal, nil, nil	⋆ r	+voice, approximant, retroflex, nil, nil
gcl	+voice, stop, velar, nil, nil	⋆ w	+voice, approximant, labial, nil, nil
pcl	-voice, stop, labial, nil, nil	y	+voice, approximant, high, nil, nil
⋆ tcl	-voice, stop, coronal, nil, nil	⋆ hh	-voice, fricative, glottal, nil, nil
⋆ kcl	-voice, stop, velar, nil, nil	⋆ hv	+voice, fricative, glottal, nil, nil
jh	+voice, fricative, high, nil, nil	⋆ iy	+voice, vowel, high, front, -round
ch	-voice, fricative, high, nil, nil	⋆ ih	+voice, vowel, high, front, -round
⋆ s	-voice, fricative, coronal, nil, nil	⋆ eh	+voice, vowel, mid, front, -round
sh	-voice, fricative, high, nil, nil	⋆ ey	+voice, vowel, mid, front, -round
⋆ z	+voice, fricative, coronal, nil, nil	ae	+voice, vowel, low, front, -round
zh	+voice, fricative, high, nil, nil	aa	+voice, vowel, low, back, -round
⋆ f	-voice, fricative, labial, nil, nil	aw	+voice, vowel, low, back, +round
⋆ th	-voice, fricative, dent, nil, nil	⋆ ay	+voice, vowel, low, front, -round
⋆ v	+voice, fricative, labial, nil, nil	⋆ ah	+voice, vowel, mid, back, -round
dh	+voice, fricative, dent, nil, nil	⋆ ao	+voice, vowel, low, back, +round
oy	+voice, vowel, low, back, -round	⋆ ow	+voice, vowel, mid, back, +round
uh	+voice, vowel, high, back, -round	⋆ uw	+voice, vowel, high, back, +round
⋆ er	+voice, vowel, retroflex, nil, -round	axr	+voice, vowel, mid, nil, -round
⋆ ax	+voice, vowel, mid, back, -round	ix	+voice, vowel, high, front, -round
⋆ h#	sil, sil, sil, sil, sil	q	-voice, vowel, glottal, nil, nil

Table A.3: Phone-feature conversion table for Numbers95. Phone transcriptions are in the ICSI phonetic alphabet.

ICSI Phone Set			
Symbol	Description	Symbol	Description
p	**p**ea	em	botto**m**
t	**t**ea	en	butto**n**
k	**k**ey	nx	wi**nn**er
pcl	*p closure*	l	**l**ike
tcl	*t closure*	el	bott**le**
kcl	*k closure*	r	**r**ight
b	**b**ee	er	bi**r**d
d	**d**ay	axr	butt**er**
g	**g**ay	y	**y**es
bcl	b closure	w	**w**ire
dcl	d closure	iy	b**ee**t
gcl	g closure	ih	b**i**b
ch	**ch**oke	ey	b**ai**t
dx	dir**t**y	eh	b**e**t
jh	**j**oke	ae	b**a**t
th	**th**in	aa	f**a**ther
dh	**th**en	ao	b**ou**ght
f	**f**ish	ah	b**u**t
v	**v**ote	ow	b**oa**t
s	**s**ound	uh	b**oo**k
z	**z**oo	uw	b**oo**t
sh	**sh**out	ix	deb**i**t
zh	a**z**ure	aw	**ou**t
hh	**h**ay	ay	b**i**te
hv	a**h**ead	oy	b**oy**
m	**m**oon	ax	**a**bout
n	**n**oon	h#	*silence*
ng	sin**g**		

Table A.4: ICSI phone set

Phone	**Features**	Phone	Features
i:	+voice, vowel, high, front,-round	I	+voice, vowel, high, front,-round
y:	+voice, vowel, high, front, +round	Y	+voice, vowel, high, front, +round
e:	+voice, vowel, mid, front, -round	E	+voice, vowel, low, front, -round
E:	+voice, vowel, low, front, -round, -central	a:	+voice, vowel, low, back, -round
a	+voice, vowel, low, back, -round, +central	6	+voice, vowel, low, -round, +central
@	+voice, vowel, mid, -round, +central	u:	+voice, vowel, high, back, +round
U	+voice, vowel, high, back, +round	o:	+voice, vowel, mid, back, +round
O	+voice, vowel, mid, back, +round, +central	p	-voice, stop, labial, nil, nil
b	+voice, stop, labial, nil, nil	t	-voice, stop, coronal, nil, nil
d	+voice, stop, coronal, nil, nil	k	-voice, stop, velar, nil, nil
g	+voice, stop, velar, nil, nil	Q	-voice, stop, glottal, nil, nil
f	-voice, fricative, labial, nil, nil	v	+voice, fricative, labial, nil, nil
s	-voice, fricative, coronal, nil, nil	z	+voice, fricative, coronal, nil, nil
S	-voice, fricative, palatal, nil, nil	Z	+voice, fricative, palatal, nil, nil
C	-voice, fricative, high, nil, nil	j	+voice, fricative, high, nil, nil
x	-voice, fricative, velar, nil, nil	h	-voice, fricative, glottal, nil, nil
m	+voice, nasal, labial, nil, nil	n	+voice, nasal, coronal, nil, nil
N	+voice, nasal, velar, nil, nil	r	+voice, fricative, velar, nil, nil
l	+voice, lateral, coronal, nil, nil		

Table A.5: Phone-feature conversion table for German. Phone transcriptions are in SAMPA notation.

Phone	+voi	-voi	sil	vo	stop	fric	son	lab	cor	pal	vel	glo	hi	mid	lo	+ro	-ro	r-nil	front	back	f-nil	lax	ten	tl-nil
sil	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NIB	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0
a:	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0
e	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0
e:	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0
E	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0
E:	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
i	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0
i:	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0
I	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0
o	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0
o:	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0
O	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0
2	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0
2:	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0
u	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
u:	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
U	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0
y	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0
y:	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0
Y	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0
@	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0
6	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0
m	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
n	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
N	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1
p	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
t	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
k	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1
b	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
d	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
g	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1
f	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
s	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
S	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1
z	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
Z	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1
C	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1
x	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1
v	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
j	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1
h	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
l	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
r	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1

Table A.6: Phonetic features used to compute phone distance in error analysis: voiced (+voi), voiceless (-voi), silence (sil), vowel (vo), stop, fricative (fric), sonorant (son), labial (lab), coronal (cor), palatal (pal), velar (vel), glottal (glo), high (hi), mid, low (lo), rounded (+ro), unrounded (-ro), neither (r-nil), front, back, neither (f-nil), lax, tense (ten), neigher (tl-nil)

References

[1] A.M. Abdelatty Ali, J. van der Spiegel, and Paul Mueller. An acoustic-phonetic feature-based system for the automatic recognition of fricative consonants. *Proceedings ICASSP-98*, pages 961–964, 1998.

[2] Jan W. Amtrup, H. Heine, and U. Jost. What's in a word graph – evaluation and enhancement of word lattices. *Proceedings of Eurospeech-97*, pages 2663–2666, 1997.

[3] B.S. Atal, J.J. Chang, M.V. Mathews, and J.W. Turkey. Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique. *JASA*, 63(5):1535–1555, 1978.

[4] L. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. A new algorithm for the estimation of hidden Markov model parameters. *Proceedings ICASSP-88*, pages 493–496, 1988.

[5] R. Battiti and A.M. Colla. Democracy in neural nets: voting schemes for classification. *Neural Networks*, 7:691–707, 1994.

[6] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

[7] E. Bax. Validation of voting committees. *Neural Computation*, 11(4):975–986, 1998.

[8] A. Bendiksen and K. Steiglitz. Neural Networks for voiced/unvoiced speech classification. *Proceedings ICASSP-90*, pages 521–524, 1990.

[9] Y. Bengio and R. De Mori. Use of neural networks for the recognition of place of articulation. *Proceedings ICASSP-88*, pages 103–106, 1988.

[10] P. Beyerlein. Discriminative model combination. *Proceedings ICASSP-98*, pages 481–484, 1998.

[11] N.N. Bitar and C.Y. Espy-Wilson. Knowledge-based parameters for HMM speech recognition. *Proceedings ICASSP-96*, pages 29–32, 1996.

[12] N.N. Bitar and C.Y. Espy-Wilson. The design of acoustic parameters for speaker-independent speech recognition. *Proceedings Eurospeech-97*, pages 1239–1242, 1997.

[13] C.S. Blackburn and S.J. Young. Towards improved speech recognition using a speech production model. *Proceedings Eurospeech-95*, pages 1623–1626, 1995.

[14] E.L. Bocchieri and J.G. Wilpon. Discriminative feature selection for speech recognition. *Computer, Speech and Language*, 7:229–246, 1993.

[15] H. Bourlard and S. Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. *Proceedings ICSLP-96*, pages 422–425, 1996.

[16] H. Bourlard and S. Dupont. Subband-based speech recognition. *Proceedings ICASSP-97*, pages 1251–1254, 1997.

[17] H. Bourlard, S. Dupont, H. Hermansky, and N. Morgan. Towards sub-band-based speech recognition. *Proceedings EUSIPCO-96*, pages 1579–1582, 1996.

[18] C.P. Browman and L. Goldstein. Towards an articulatory phonology. In C. Ewen and J. Anderson, editors, *Phonology Yearbook 2*, pages 219–252, Cambridge, UK, 1986. Cambridge University Press.

[19] C.P. Browman and L. Goldstein. Articulatory phonology: an overview. *Phonetica*, 49:155–180, 1992.

[20] B. Carré. *Graphs and Networks*. Clarendon Press, Oxford, 1979.

[21] L.L. Chase. *Error-Responsive Feedback Mechanisms for Speech Recognizers*. PhD thesis, Carnegie-Mellon University, 1997.

[22] C. Chesta, A. Girardi, and P. Laface. Discriminative training of Hidden Markov Models. *Proceedings ICASSP-98*, pages 449–452, 1998.

[23] N. Chomsky and M. Halle. *The Sound Pattern of English*. Harper & Row, New York, 1968.

[24] W. Chou, B. Juan, and C. Lee. Segmental GPD training of HMM based speech recogniser. *Proceedings of ICASSP-92*, pages 473–476, 1992.

[25] J. Clark and C. Yallop. *An Introduction to Phonetics and Phonology*. Blackwell, London, 1990.

[26] R.P. Cohn. Robust voiced/unvoiced speech classification using a neural net. *Proceedings ICASSP-91*, pages 437–440, 1991.

[27] R.A. Cole, M. Noel, T. Lander, and T. Durham. New telephone speech corpora at CLSU. *Eurospeech 95*, pages 821–824, 1995.

[28] M.W. Craven. *Extracting Comprehensible Models from Trained Neural Networks*. PhD thesis, Department of Computer Sciences, University of Wisconsin-Madison, 1996.

[29] P. Dalsgaard. Phoneme label alignment using acoustic-phonetic features and Gaussian probability density functions. *Computer, Speech and Language*, 6:303–329, 1992.

[30] P. Delattre. From acoustic cues to distinctive features. *Phonetica*, 18:198–230, 1968.

[31] L. Deng and K. Erler. Microstructural speech units and their HMM representations for discrete utterance speech recognition. *Proceedings ICASSP-91*, pages 193–196, 1991.

[32] L. Deng, M. Lennig, F. Seitz, and P. Mermelstein. Large vocabulary word-recognition using context-dependent allophonic hidden Markov models. *Computer, Speech and Language*, 4:345–357, 1990.

[33] L. Deng and D. Sun. Phonetic classification and recognition using HMM representation of overlapping articulator features for all classes of English sounds. *Proceedings ICASSP-94*, pages 45–47, 1994.

[34] L. Deng and D. Sun. A statistical approach to ASR using atomic units constructed from overlapping articulatory features. *JASA*, 95:2702–2719, 1994.

[35] S. Dupont and H. Bourlard. Using multiple time scales in a multi-stream recognition system. *Proceedings Eurospeech-97*, pages 3–6, 1997.

[36] E. Eide and H. Gish. A parametric approach to vocal tract length normalization. *Proceedings ICASSP-96*, pages 346–348, 1996.

[37] E. Eide, J.R. Rohlicek, H. Gish, and S. Mitter. A linguistic feature representation of the speech waveform. *Proceedings ICASSP-93*, pages 483–486, 1993.

[38] K. Elenius and M. Blomberg. Comparing phoneme and feature based speech recognition using artificial neural networks. *Proceedings ICSLP-92*, pages 1279–1282, 1992.

[39] K. Elenius and G. Tacacs. Phoneme recognition with an artificial neural network. *Proceedings Eurospeech-91*, pages 121–124, 1991.

[40] K. Erler and L. Deng. Hidden Markov model representation of quantized articulatory features for speech recognition. *Computer, Speech, and Language*, 7:265–282, 1993.

[41] K. Erler and G.H. Freeman. An HMM-based speech recognizer using overlapping articulatory features. *JASA*, pages 2500–2513, 1996.

[42] S. Young et. al. *The HTK Book for HTK 2.1*. Entropics, 1997.

[43] G.A. Fink. Developing HMM-based recognizers with ESMERALDA. In *Proceedings of Workshop on Text, Speech, and Dialogue*, Pilsen, Czech Republic, September 1999.

[44] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997.

[45] C.A. Fowler and M.R. Smith. Speech perception as vector analysis. In J. Perkell and D. Klatt, editors, *Invariance and Variability in Speech Processes*. Hillsdale, N.J.: Erlbaum, 1986.

[46] J. Franke and E. Mandler. A comparison of two approaches for combining the votes of cooperating classifiers. *Proceedings of the 11th International Conference of Pattern Recognition*, pages 611–614, 1992.

[47] M. Franzini, K.F. Lee, and A. Waibel. Connectionist Viterbi training: a new hybrid method for continuous speech recognition. *Proceedings ICASSP-90*, pages 425–428, 1990.

[48] Sadaoki Furui and Chin-Hui Lee. Robust speech recognition – an overview. *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 1995.

[49] M.F.J. Gales, D. Pye, and P.C. Woodland. Variance compensation within the MLLR framework for robust speech recognition and speaker adaptation. *Proceedings ICSLP-96*, pages 1832–1835, 1996.

[50] A. Ganapathiraju, V. Goel, J. Picone, A. Corrada, G. Doddington, K. Kirchhoff, M. Ordowski, and B. Wheatley. Syllable – a promising recognition unit for LVCSR. *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, pages 207–214, 1997.

[51] S. Gay, B. Lindblom, and J. Lubker. Production of bite-block vowels: acoustic equivalence by selective compensation. *JASA*, 69:802–810, 1981.

[52] T. Ghiselli-Crippa and A. El-Jaroudi. A fast neural net training algorithm and its application to voiced-unvoiced-silence classification of speech. *Proceedings ICASSP-91*, pages 441–444, 1991.

[53] O. Ghitza. Processing of spoken CVCs in the auditory periphery. *JASA*, 94:2507–2516, 1993.

[54] J.H. Greenberg and J.J. Jenkins. Studies in the psychological correlates of the sound system of American English. *Word*, 20:157–177, 1964.

[55] S. Greenberg and B.E.D. Kingsbury. The modulation spectrogram: In pursuit of an invariant representation of speech. *Proceedings ICASSP-97*, 2:1647–1650, 1997.

[56] A.K. Halberstadt and J.R. Glass. Heterogeneous measurements and multiple classifiers for speech recognition. *Proceedings ICSLP-98*, pages 995–998, 1998.

[57] A.V. Hansen. Acoustic parameters optimised for recognition of phonetic features. *Proceedings Eurospeech-97*, pages 397–400, 1997.

[58] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft, 1995.

[59] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks. Technical Report MSR-TR-94-09, Microsoft, 1994.

[60] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[61] T.K. Ho, J.J. HULL, and S.N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:66–75, 1994.

[62] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[63] R.A. Jacobs, M.I. Jordan, S.J. Nowland, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1994.

[64] R. Jakobson, G. Fant, and M. Halle. *Preliminaries to Speech Analysis: the Distinctive Features and their Correlates*. MIT Press, Cambridge, Mass., 1952.

[65] F.V. Jensen. *An Introduction to Bayesian Networks*. Springer, Berlin, 1996.

[66] B. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40:3043–3054, 1992.

[67] T. Kamm, A. Andreou, and J. Cohen. Vocal tract length normalization in speech recognition: compensating for systematic speaker variability. In *Proceedings of the Fifteenth Annnual Speech Research Symposium*, Baltimore, 1995. Johns Hopkins University.

[68] S. King, T. Stephenson, S. Isard, P. Taylor, and A. Strachan. Speech recognition via phonetically featured syllables. *Proceedings ICSLP-98*, pages 1031–1034, 1998.

[69] B.E.D. Kingsbury and N. Morgan. Recognizing reverberant speech with RASTA-PLP. *Proceedings ICASSP-97*, 1997.

[70] K. Kirchhoff and J. Bilmes. Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values. *Proceedings ICASSP-99*, 1999.

[71] J. Kittler, M. Hataf, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[72] R. Kneser and H. Ney. Improved backing-off for M-gram language modeling. In *Proceedings ICASSP-95*, pages 181–184, Detroit, MI, May 1995.

[73] K. Kohler, G. Lex, M. Pätzold, M. Scheffers, A. Simpson, and W. Thon. Handbuch zur Datenaufnahmen und Transliteration in TP14 von VERBMOBIL – 3.0. Verbmobil Technical Report 11, IPDS Kiel, 1994.

[74] D. Koller and M. Sahami. Toward optimal feature selection. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 281–289. Morgan Kaufman, 1996.

[75] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*. MIT Press, 1995.

[76] P. Ladefoged. *A Course in Phonetics*. Harcourt Brace Jovanovich, London, 1982.

[77] C.H. Lee, L.R. Rabiner, R. Pieraccini, and J.G. Wilpon. Acoustic modeling for large vocabulary speech recognition. *Computer, Speech and Language*, 4:127–165, 1990.

[78] K.F Lee. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer, Boston, 1989.

[79] K.F. Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:599–608, 1990.

[80] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer, Speech and Language*, 9:171–186, 1995.

[81] A.M. Liberman, F.S. Cooper, D.S Shankweiler, and M. Studdert-Kennedy. Perception of the speech code. *Psychological Review*, 74:431–461, 1967.

[82] A.M. Liberman and I.G. Mattingly. The Motor Theory of Speech Perception revised. *Cognition*, 21:1–36, 1986.

[83] P. Lieberman and S.E. Blumstein. *Speech Physiology, Speech Perception, and Acoustic Phonetics*. Cambridge University Press, Cambridge, 1988.

[84] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communication*, 28:84–95, 1980.

[85] R.P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1):1–15, 1997.

[86] L. Lisker. Rapid vs. rapid: a catalogue of acoustic features that may cue the distinction. *Haskins Laboratories Status Report on Speech Research SR-54*, pages 127–132, 1978.

[87] S.A. Liu. Landmark detection for distinctive feature-based speech recognition. *Journal of the Acoustical Society of America*, 100(5):3417–3430, 1996.

[88] P. McMahon, P. Court, and S Vaseghi. Discriminative weighting of multi-resolution sub-band cepstral features for speech recognition. *Proceedings ICSLP-98*, pages 1055–1058, 1998.

[89] G. Miller and P. Nicely. An analysis of some perceptual confusions among some English consonants. *JASA*, 27:338–352, 1955.

[90] N. Mirghafori and N. Morgan. Combining connectionist multi-band and full-band probability streams for speech recognition of natural numbers. *Proceedings ICSLP-98*, pages 743–746, 1998.

[91] N. Morgan and H. Bourlard. An introduction to hybrid HMM/Connectionist continuous speech recognition. *IEEE Signal Processing Magazine*, pages 25–42, 1995.

[92] P. Niyogi, C. Burges, and P. Ramesh. Distinctive feature detection using Support Vector Machines. *Proceedings ICASSP-99*, 1999.

[93] P. Niyogi and P. Ramesh. Incorporating voice onset time to improve letter recognition accuracies. *Proceedings ICASSP-97*, pages 13–16, 1997.

[94] S. Ortmanns and X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, November 1996.

[95] M. Ostendorf and S. Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 1857–1869, 1989.

[96] J. Bernstein P. Price, W.M. Fisher and D.S. Pallett. The DARPA 1000-word resource management database for continuous speech recognition. *Proceedings ICASSP-88*, pages 651–654, 1988.

[97] J. Papcun, T.R. 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. *JASA*, 92:688–700, 1992.

[98] J. Pearl. *Probabilistic Reasoning*. Morgan Kaufman, San Mateo, CA, 1988.

[99] R.W Peters. Dimension of perception for consonants. *JAcS*, 35:1985–1989, 1963.

[100] M.G. Rahim. *Artificial Neural Networks for Speech Analysis/Synthesis*. Chapman & Hall, London, 1994.

[101] M.G. Rahim and C.C. Goodyear. Parameter estimation for spectral matching in articulatory synthesis. *Colloquium on Spectral Estimation Techniques for Speech Processing*, 1989.

[102] P. Ramesh and P. Niyogi. The voicing feature for stop consonants: acoustic phonetic analyses and automatic speech recognition. *Proceedings ICSLP-98*, pages 2263–2266, 1998.

[103] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation*, 3:461–483, 1991.

[104] H.B. Richards. *The Use of Articulatory Parameters for Speech Analysis*. PhD thesis, University of Wales, Swansea, 1997.

[105] H.B. Richards, J.S. Mason, J.S. Bridle, and M.J. Hunt. Vocal tract shape trajectory estimation using MLP analysis-by-synthesis. *Proceedings ICASSP-97*, pages 1287–1290, 1997.

[106] H.B. Richards, J.S. Mason, M.J. Hunt, and J.S. Bridle. Deriving articulatory representations of speech. *Proceedings Eurospeech-95*, pages 761–7764, 1995.

[107] H.B. Richards, J.S. Mason, M.J. Hunt, and J.S. Bridle. Deriving articulatory representations of speech with various excitation modes. *Proceedings ICSLP-96*, pages 1229–1232, 1996.

[108] Tony Robinson and F. Fallside. A recurrent error propagation network speech recognition system. *Computer, Speech, and Language*, 5:259–274, 1991.

[109] R. Schlueter and W. Macherey. Comparison of discriminative training criteria. *Proceedings ICASSP-98*, pages 493–496, 1998.

[110] O. Schmidbauer. Robust statistic modelling of systematic variabilities in continuous speech incorporating acoustic-articulatory relations. *Proceedings ICASSP-89*, pages 616–619, 1989.

[111] J. Schürmann. *Pattern Classification: a Unified View of Statistical and Neural Approaches*. John Wiley, New York, 1996.

[112] E. Singer and R. Lippmann. A speech recognizer using radial basis function neural networks in an HMM framework. *Proceedings ICASSP-92*, pages 629–632, 1992.

[113] S. Singh. Cross-language study of perceptual confusions of plosive phonemes in two conditions of distortion. *JASA*, 39:635–656, 1966.

[114] P. Steingrimsson, B. Markussen, O. Andersen, P. Dalsgaard, and W. Barry. From acoustic signal to phonetic features: a dynamically constrained self-organising neural network. *Proceedings of International Congress of Phonetic Sciences*, 1995.

[115] S.S. Stevens and J. Volkman. The relation of pitch to frequency. *American Journal of Psychology*, 53, 1940.

[116] M. Tomlinson, M. Russell, R. Moore, A. Buckland, and M. Fawley. Modelling asynchrony in speech using elementary single-signal decomposition. *Proceedings ICASSP-97*, pages 1247–1250, 1997.

[117] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[118] A.P. Varga and R.K. Moore. Hidden Markov Model decomposition of speech and noise. *Proceedings ICASSP-90*, pages 845–848, 1990.

[119] A.P. Varga, H.J.M. Steeneken, M. Tomlinsons, and D. Jones. The Noisex-92 study on the effect of additive noise on automatic speech recognition. Technical report, DRA Speech Research Unit, 1992.

[120] M.D. Wang and R.C. Bilger. Consonant confusions in noise: a study of perceptual features. *JASA*, 54:1248–1266, 1973.

[121] M.B. Wesenick and A. Kipp. Estimating the quality of phonetic transcriptions and segmentations of speech signals. *Proceedings ICSLP-96*, pages 129–132, 1996.

[122] H. White. Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464, 1989.

[123] C. Windheuser, F. Bimbot, and P. Haffner. A probabilistic framework for word recognition using phonetic features. *Proceedings ICSLP-94*, pages 287–290, 1994.

[124] S.-L. Wu, B.E.D. Kingsbury, N. Morgan, and S. Greenberg. Incorporating information from syllable-length time scales into automatic speech recognition. *Proceedings ICASSP-98*, pages 721–724, 1998.

[125] S.J. Young. The general use of tying in phoneme-based HMM speech recognisers. *Proceedings ICASSP-92*, pages 569–572, 1992.

[126] J. Zacks and T.R. Thomas. A new neural network for articulatory speech recognition and its application to vowel identification. *Computer, Speech and Language*, 8:189–209, 1994.