# Posterior-Based Speech Models and their Application to Hungarian Speech Recognition

László Tóth

Research Group on Artificial Intelligence

January 2006

A dissertation submitted for the degree of doctor of philosophy of the University of Szeged

University of Szeged
Doctoral School in Mathematics and Computer Science
Ph.D. Program in Informatics

# Preface

The current speech recognition technology is built on very simple statistical principles instead of speech-specific knowledge. Although there are constant attempts to incorporate what we know about human speech perception, these usually result in novel preprocessing methods and leave the statistical framework untouched. In particular, the 3-state left-to-right hidden Markov phone modelling methodology has been practically unchallenged for the last decade. Rather, performance improvement was attained mainly by collecting enormous training corpora and by building sophisticated language models. However, nowadays the technology seems to have reached its limits, its abilities still being far from that of humans. Probably it is time to step back and refine the acoustic models as well, retaining the statistical approach but narrowing the gap between the properties of the models and human speech comprehension.

This dissertation is written in this spirit and starts out by listing the general properties of an envisioned alternative speech recognition framework. In the subsequent chapters two acoustic modelling techniques are proposed that meet some of these requirements (although not all of them by far). Also, great care is taken to analyze and compare the behavior of the conventional HMM and these novel models on very basic tasks. Surely, understanding how they work – or do not work – and how they fulfill our intuitive expectations is a vital step in constructing novel – and hopefully better – speech recognition solutions.

The first of my **acknowledgements** goes to my supervisor, Prof. János Csirik for supporting my research and letting me work at an inspiring department, the Research Group on Artificial Intelligence. Secondly, I would like to thank András Kocsor for the fruitful discussions and for the constant stimulation to pursue my studies. My heartfelt thanks goes to all my colleagues in the Speech Technology Team, who offered an invaluable amount of help and assistance in getting the results presented in this dissertation. Without their contribution the OASIS recognition framework would not exist – neither conceptually, nor as a software environment. András Kocsor and Kornél Kovács designed the first version of the OASIS Speech Lab, and it was later totally redesigned and rewritten by László Felföldi. Gábor Gosztolya helped me a lot in executing the tests presented in this dissertation and Dénes Paczolay readily answered my program coding questions even in the most unsuitable moments. I would also like to express my gratitude to David Curley for scrutinizing and correcting this dissertation from a linguistic point of view and to Péter Balázs for all those beers we had together.

*László Tóth, December 2005.*

# Notations

In a general pattern classification context:

| | |
|---|---|
| $\mathcal{X}$ | the space of measurement vectors |
| $x$ | measurement vector; $x \in \mathcal{X}$ |
| $\mathcal{C} = \{c_1, ..., c_K\}$ | the set of possible class labels |

In a speech recognition context:

| | |
|---|---|
| $\mathcal{C} = \{c_1, ..., c_K\}$ | the set of possible phonetic class labels |
| $U = (u_1, ..., u_N)$ | a sequence of $N$ phonetic units; $u_i \in \mathcal{C}$ |
| $W = (u_1, ..., u_N)$ | a sequence of $N$ phonetic units that forms a word or a series of words; $u_i \in \mathcal{C}$ |
| $X = (x_1, ..., x_T)$ | a frame-based acoustic observation sequence of length $T$ |
| $S = (s_0, ..., s_N)$ | a segmentation consisting of $N$ segments given as a sequence of segment boundary time indices; $1 \leq s_i \leq T$ |
| $S_i = (s_{i-1}, s_i)$ | the $i$th segment of a segmentation |
| $x_{s_{i-1}}^{s_i - 1}$ | a short notation for $(x_{s_{i-1}}, ..., x_{s_i - 1})$, the observation vector subsequence of the $i$th segment |
| $X_i$ | the acoustic data of the $i$th segment, in particular when represented by a fixed-length segmental feature vector |
| $\{q_1, ..., q_M\}$ | the states of an HMM model; when using 1-state phone models it coincides with $\{c_1, ..., c_K\}$ |

# List of Figures

# List of Tables

# Contents

# Chapter 1

# Introduction

*"A physicist, a computer scientist and a mathematician are each locked into a room with canned food. After a week the cells are opened. [..] In the room of the mathematician the walls are full of formulas but the can is untouched and the mathematician is dead. The top line on the wall says: «We assume the can is open.»"*

Most of the early attempts at automatic speech recognition tried to achieve their goals by exploiting phonetic/linguistic knowledge. The most sophisticated of these knowledge-based systems were built on the then-popular rule-based expert systems framework and combined several knowledge sources of different levels (features, segments, syllables, words, etc.). However, because of our incomplete understanding of the factors behind the immense variability of speech signals the rule-based experts were fragile and their combination strategy was heuristic, cumbersome and far from optimal. The advent of the statistical machine learning algorithms, and especially the hidden Markov model promised a very elegant solution to these problems, as it offers the automatic trainability of its components, a very simple combination scheme and the mathematical guarantee of optimal performance at the highest level. During the 90s huge training corpora were collected, and thanks to the rapid increase in processor speed, memory and hard drive capacity, the HMM-based recognizers can now perform large vocabulary continuous speech recognition in real-time. However, their abilities are still a magnitude worse than those of human speech perception, and the technology now seems to have reached its limits. Probably it is time to find the way of phonetic/linguistic knowledge back into speech recognition research. There seems to be no reason for totally giving up on the statistical approach, as currently nothing better is known. A more suitable strategy seems to be to analyze the behavior of the current models, find their weakest points – where they differ the most from what intuition would expect – and refine them by incorporating linguistic knowledge. Of course, many such refinements were proposed in the last two decades, but most of these are related to the feature extraction step and leave the HMM phone models intact. This was done in spite of the fact that the HMM structure is overly simplistic and in many respects quite counter-intuitive.

The main motivation behind this dissertation was to find such alternative models that are still statistics-based, but hopefully are somewhat more closer to what we know about human speech perception. Two such models will be proposed in the subsequent chapters. One of them objects to the sequence-based nature of current speech recognizers, that is that they seek to identify every centisecond of a speech signal. As an alternative, the segment-based technology is proposed which tries to find and model whole phonetic segments in one. Another debatable feature of HMM models is that they combine the local probability estimates by multiplication, corresponding to an AND-like scheme (in the sense that one zero component can bring the whole product to zero). Intuition says that speech is a redundant code and so it is illogical to demand all the measurements to support a hypothesis. The other proposed model follows this spirit and recommends combining the local evidence by summation rather than multiplication.

One great appeal of HMM models is that they can be trained at the level of utterances. The importance of this is, of course, that optimality can be guaranteed at this level. It also has a side-effect, however, that the behavior of the model at lower levels can be ignored. Nowadays most papers deal with the recognition of continuous and/or noisy speech over large vocabularies – giving the false impression that 'simple' phonetic decoding is already solved. This is, however, far from being the case. But improving the very low levels would require taking the models apart and analyzing their behavior on very basic tasks, which does not sound too attractive to most developers. In this dissertation I always test the phonetic classification and phonetic recognition capabilities of a model before moving to word recognition or decoding tasks. I think that understanding what is going on at these levels is a prerequisite if we intend to identify those points where improvements are possible and if we want to bring the behavior of the models closer to that of humans. I consider the insight I gained from these experiments a much more important result of this dissertation than the proposed models themselves.

## 1.1 Summary by Chapters

There are two chapters in the dissertation that do not contain scientific contributions from the author but has the goal of reviewing certain areas. Thus, Chapter 4 gives an overview of the software environment and speech databases used throughout the dissertation, and Chapter 5 collects all the research results that I judged to be related to my investigations and I was familiar with during the period of my studies.

The remaining chapters basically follow the chronological order of my research efforts. Chapter 2 gives a detailed description of the critical issues of the current technology and presents some of the basic features that I would prefer to see from a alternative, novel one. It also introduces the mathematical tools used in current models, since our alternative models will apply the same decomposition tricks, only in slightly different ways.

Chapter 3 presents a generalized algorithmic framework that forms the basis of the implementation of our speech decoder. All the models tested in the dissertation – including the HMM – will be a special case of this decoding routine.

Chapter 6 introduces the posterior-based segmental model, our team's first attempt to create a viable alternative to HMM-based phone models. Although it turned out not long after that the segment-based representation could easily outperform HMMs in classifying phonetic segments, it took a lots of effort to bring it up to the level of HMMs in phonetic decoding or word recognition tasks. Hence, a large part of this chapter is concerned with improving the segment-based model by refining its so-called segmentation probability component.

Another consequence of confronting the difficulties with the segmental model was that I realized HMMs are in fact rather good – in spite of the quite obvious arguments against them. This revelation led to Chapter 7, in which a profound analysis is given on how HMMs perform phonetic decoding and why they can solve the problem of phonetic segmentation and classification while their probability estimates are very inaccurate. The unusual segment-based view and the comparison with the segmental model brought me a new insight into these issues.

In Chapter 8 the technique of segment-based interpretation is extended to the so-called HMM/ANN hybrid. Namely, we identify which component of the hybrid model corresponds to which component of the segment-based model. This analysis leads to the suggestion of an alternative hybrid model in which the frame-based posteriors are combined by averaging instead of multiplication. This 'averaging hybrid' turns out to behave similarly or slightly better than the conventional one on phone classification, phone recognition and word recognition tasks as well.

Chapter 9 proposes two slight refinements over the hybrid model of Chapter 8. One of these is the application of an explicit gamma-distribution based duration model instead of the exponential one inherent to HMMs. The other refinement concerns the training of the neural nets used in the hybrid. Both modifications result in a modest improvement in the word error rates.

## 1.2 Summary by Results

In the following a thesis-like listing of the most important results of the dissertation is given. Table 1.1 shows which thesis is described in which publication by the author.

I. ) The author developed a segment-based feature set for the representation of phonetic segments. He tested this feature set on several speech corpora and in combination with various machine learning algorithms, and demonstrated that in most cases it results in better phone classification scores than the conventional HMM phone models.

II. ) The author developed various strategies for estimating the segmentation probability component of the posterior-based segmental model, based on the concept of anti-phones. He tested the proposed modelling schemes by comparing their speech recognition performance on several speech databases.

| | [77] | [109] | [110] | [111] | [112] | [113] | [114] |
|---|---|---|---|---|---|---|---|
| I. | ● | ● | ● | | | | |
| II. | | ● | ● | | | | |
| III. | | | | ● | | | |
| IV. | | | | | ● | | |
| V. | | | | | | ● | |
| VI. | | | | | | ● | |
| VII. | | | | | | | ● |

Table 1.1: The relation between the theses and the corresponding publications

III.) The author investigated the applicability of replicator neural networks for the estimation of the segmentation probability component of segmental models.

IV. ) The author investigated how the modelling bias caused by the naive Bayes assumption influences the performance of HMM phone models. Based on the observations he argued that this bias is such that it does not deteriorate the phone classification performance of the models and it helps them in finding the correct segmentation of the input signal. These arguments together help explain why HMMs are good at phonetic decoding while their probability estimates are quite inaccurate.

V. ) The author examined the behavior of the conventional HMM/ANN hybrid model from a segment-based point of view. Based on the findings of this, he introduced a novel type of HMM/ANN hybrid which combines the frame-based posterior estimates by averaging instead of multiplication. He justified experimentally that the averaging hybrid is capable of a similar or slightly better performance than the conventional hybrid.

VI. ) The author examined the efficiency of using explicit duration models in the HMM/ANN framework. He found that the gamma-distribution based duration model leads to increased recognition performance over the conventional exponential model in both the conventional and the averaging hybrid.

VII.) The author proposed a resampling-based training scheme for the training of the neural nets used in the hybrid models. In experiments the proposed algorithm resulted in modest improvements in recognition accuracy.

# Chapter 2

# The Decomposition Problem in Statistical Speech Recognition

*"All models are wrong, but some models are useful".*
*George E.P. Box*

The main goal of this chapter is to explain our motivations for experimenting with the models introduced in the subsequent chapters. Our basis argument will be that having chosen to apply the statistical methodology to speech recognition inescapably leads us to a decomposition problem. Unfortunately, probability theory offers only a very limited range of tools for performing this decomposition, hence the resulting mathematical models will have several assumptions that contradict our intuition on how speech perception works. Giving up the statistical framework does not seem reasonable at present, but the models can probably be brought closer to our intuitive expectations. Most importantly, we propose that the generative and independency-based modelling structure should be replaced by a decoding-oriented one that aims at identifying familiar data blocks and combining them in a way that can exploit their redundancy.

## 2.1 The Basic Issues of Statistical Modelling in Speech Recognition

Out of the several theoretical frameworks of machine learning the most popular in practice is that of statistical pattern recognition [26]. At its core is the relatively simple Bayes decision rule, stating that if the task is to classify certain objects into one of the classes $\mathcal{C} = \{c_1, ..., c_K\}$ based on some measurement vector $x$, and we intend to minimize the number of misclassifications on the long term, then the optimal choice is the class having the largest posterior probability $P(c_k|x)$. Thus, the problem of learning boils down to obtaining an estimate of $P(c_k|x)$ that is as accurate as possible based on the set of training examples available. Several learning algorithms fulfilling this task have been invented over the decades and have been successfully applied to many practical fields. Hence, it is reasonable to use the statistical modelling approach to speech recognition as well.

However, the (general) speech recognition task has a special problem: the huge size of the space to be modelled. That is, the length of the utterances can be quite large, and even if we assume a reasonable upper bound on their length (the speaker sometimes must take a breath, after all), the number of corresponding possible transcriptions (phone or word strings) is so large that they cannot be managed directly. Moreover, the conventional algorithms developed for statistical pattern recognition assume both the class set and the feature set to be of a fixed and relatively small size.

Hence, it seems necessary to decompose both the transcript label $c_k$ and the measurement vector $x$ into some smaller units $c'$ and $x'$. Direct probability estimates $P(c'|x')$ are then created only over the subspaces of these units, and an estimate of the global probability $P(c_k|x)$ is obtained by properly combining the local estimates of the units. The necessity of decomposition makes the following modelling issues crucial in statistical speech recognition:

1. What is a proper choice for the recognition units $c'$ and for the corresponding measurement subspace $x'$?

2. What is the best way to model (parameterize and train) these units?

3. What is the proper way of combining the local probabilities $P(c'|x')$ into a global utterance-level probability $P(c_k|x)$?

There are two main driving forces when answering these modelling questions. One of them is, of course, a priori knowledge about human speech perception and language comprehension. The other one is mathematical tractability. Unfortunately, there are controversial issues with both.

First, human speech perception is a terribly complex problem that requires interdisciplinary knowledge from such fields as phonetics, neurophysiology and cognitive psychology. Moreover, it still has many uncharted spots. Even worse, the partial knowledge we have about it is not written in a language for electrical and software engineers. Because of these difficulties, the current speech recognition modelling technology is built almost exclusively on mathematical (more precisely, statistical) principles. Quoting Jelinek [63]:

«Statistics wins because its techniques are well developed. "Knowledge" loses because those who think they have it do not know how to incorporate what they have. So far, it has turned out more profitable to estimate reliably very simple parameters than to introduce a complex model whose parameter values cannot be verified.»

That is, the main advantage offered by the statistical approach is that its models are trainable from data. The price is that trainability requires relatively simple models. Moreover, probability theory yields a very limited toolkit for decomposing the probability $P(c_k|x)$ – at least, in a simple way. Hence the resulting models ignore even those quite basic facts that are known about how speech works and their behavior is in many cases counter-intuitive.

In terms of machine learning, we suspect that our statistical models are significantly biased. This means that increasing the amount of training data is in itself not sufficient to obtain perfect speech recognizers: the models themselves should also be refined. The easiest way of reducing the model bias is to make the model more flexible via the introduction of more free parameters – this of course increases variance and hence the proper tuning of the model parameters will require more and more training data. The other option is to replace the (mathematically inspired) bias with a priori speech-specific information. These two types of improvements do not really clash, and in fact could be pursued in parallel. But in general we can say that in the last decade significantly more effort has been devoted to improving the models by adding more parameters (and more training data) than by bringing them closer to what is known about speech perception[1]. The main motivation of this thesis is to propose alternative models that retain the statistical modelling framework, but hopefully are somewhat closer to what intuition and prior knowledge suggest.

Before going into the details of modelling techniques, there is one more general issue to be discussed, that of model verification. There seems to be the wide-spread misconception among engineers that the solid mathematical background of a model is sufficient to justify its use. In fact, however, mathematics *per se* cannot guarantee that a certain kind of model is suitable for describing a certain kind of natural phenomenon. It can only ensure the mathematical soundness of the model, and its practical usefulness can be verified only by experimentation. However, the 'no free lunch theorem' [26] says that for every model there are possible data sets that it would fit well and data sets that it would fit poorly. Hence, justifying that one model is better than another one for a certain type of task would require comparing their average performance over as many data sets as possible. Unfortunately, as training and testing a speech recognition model can take quite long, we usually confine ourselves to comparing the models on a few standardized databases only. A further complication is that we can only approximate the optimal parameters and the training process is prone to under- or overtraining. Thus, from one bad result it is hard to conclude whether there is a conceptional problem with the model or the training process has failed or the given data set has some kind of unwanted speciality. This again reinforces the need for exhaustive testing on many databases.

In the following section we will briefly summarize how the utterance-level probability is decomposed in conventional models. First, it will allow us to discuss and scrutinize its key features. Second, it will allow us to collect the mathematical tools that are applicable during decomposition. This is important because the alternative decompositions we are going to suggest in subsequent chapters utilize the same methods, only in slightly different ways.

[1] The reader may protest here saying that countless novel front-end algorithms were proposed based on speech perception and auditory principles. But these papers usually end with testing the new feature set within a conventional recognizer – and our point here is that the recognition algorithm should also be reviewed.

## 2.2 Decomposition in Conventional Statistical Speech Recognition Models

In the following we will assume that the speech signal is given as a sequence of uniformly sampled observation vectors $X = (x_1, ..., x_T)$ and the result of recognition is required in the form of a series of phonetic symbols $W = (u_1, ..., u_N)$ over a fixed set of phone labels $\{c_1, ..., c_K\}$[2].

For an optimal Bayesian decision we need an estimate of $P(W|X)$. Here we need the first decomposition technique: Bayes' rule. When applying it we obtain[3]

$$P(W|X) = \frac{p(X|W)P(W)}{p(X)}. \qquad (2.1)$$

During the recognition of a given observation sequence $p(X)$ is constant, so it can be dropped because it does not influence the maximization with respect to $W$. From the remaining two factors $P(W)$ does not depend on the acoustic observation $X$, hence it is usually called the *language model*, while the other component $p(X|W)$ is referred to as the *acoustic model*.

Although the motivation for applying Bayes' rule in Eq. (2.1) was to separate the acoustic and language models, it has another very important technical aspect: as the possible values of $W$ always form a finite set, while the observation vectors $X$ are usually from a continuous measurement space, quite different techniques are required to model the posterior probability $P(W|X)$ of the classes than the class-conditional distribution $p(X|W)$ of the feature vectors. In the literature the algorithms that approximate the latter are usually called *generative* while those that approximate the former are referred to as *discriminative* models[4]. Note that theoretically it does not matter which type of model we work with, as they can always be converted to each other via Bayes' rule (with the help of the priors). But in practice they produce only an approximation based on the training data, and because of their different structure and training algorithm these estimates might be quite different. The most important advantage of discriminative models over generative ones is that they are easier to train for optimal discrimination (hence their name). However, discriminative training algorithms also exist for generative models, so we prefer to call them *posterior-based* models instead of discriminative ones. The most well-known representative of the class of generative learners is the Gaussian mixture model (GMM), while the most popular posterior-based learner is the artificial neural network (ANN).

The second decomposition technique we apply is the introduction of latent variable(s). For this we will have the simplifying assumption that speech is a sequence of

[2] The symbol set used usually does not fully coincide with the phone or phoneme set of the language to be recognized. So technically we should talk about "phone-like units", but for brevity we simply say phones or phonemes.

[3] Throughout the dissertation discrete probabilities will be denoted by $P$ and continuous probability densities will be denoted by $p$.

[4] In speech recognition some authors refer to "recognition"[89], or "perception"[83] vs. "production" models.

phonetic segments – this view is also known as the "beads-on-a-string" paradigm. Although it is debated by modern phonetics/phonology, currently it is a standard assumption in speech recognition models. Based on this, the speech signal can be decomposed into a sequence of phonetic segments by segment boundary time markers. With this view phonetic decoding requires solving two problems in parallel: finding the segment boundaries and identifying the segments. Let us denote the possible segmentations by $S$ and introduce it as a latent variable. Then, using the law of total probability, the acoustic model likelihood $p(X|W)$ can be written as

$$p(X|W) = \sum_S p(X,S|W) = \sum_S p(X|S,W)P(S|W), \tag{2.2}$$

where in the second step the chain rule was applied (the third decomposition technique).

Here the $P(S|W)$ factor is responsible for finding the most probable segmentation, and it is the task of $p(X|S,W)$ to identify the segments defined by a given segmentation $S$. In practice it is usually not feasible to evaluate the sum over all possible $S$, so it is approximated by a maximization. This is based on the assumption that there is only one correct and hence high probability segmentation, and the contributions of the remaining ones are negligible. With this modification the phonetic decoding problem becomes a parallel search for a maximum over both $W$ and $S$. Note that the simplification of substituting a maximization for the summation was introduced based on purely practical computational reasons and not out of any good phonetic or perceptional argument.

The fourth decomposition scheme that is very frequently applied is the assumption of independence (also known as the naive Bayes assumption)[5], which allows a multivariate probability density function to be decomposed into a product of densities of fewer or one variables. In fact, probability theory offers no other trivial way of decomposing a multivariate probability, so there is a serious pressure to apply it even in cases when the independence assumption is quite obviously false.

We use the independence assumption to decompose $p(X|S,W)$ and $P(S|W)$ into segment-level scores. We will denote the $i$th segment by $S_i$, the signal section that belongs to it as $x_{s_{i-1}}^{s_i-1}$, and the corresponding phone label by $u_i$. Using this notation, the decomposition leads to the following approximations

$$p(X|S,W) \approx \prod_i p(x_{s_{i-1}}^{s_i-1}|u_i) \tag{2.3}$$

and

$$P(S|W) \approx \prod_i P(S_i|u_i). \tag{2.4}$$

Here the task of the $p(x_{s_{i-1}}^{s_i-1}|u_i)$ components is to tell us how likely the acoustic observations of the $i$th segment belong to (were generated by) the phone $u_i$. The other component, $P(S_i|u_i)$ has to tell us how probable the markers of the $i$th segment select

[5] A somewhat weaker assumption is to presume Markovian dependence on the preceding couple of events, but this is usually applied to the linguistic units and not the acoustic data.

a phonetic segment. Notice that this component has no access to the measurements $x_{s_{i-1}}^{s_i-1}$, but knows only the place of the boundaries of $S_i$. This is why $P(S_i|u_i)$ is implemented as a duration model in practice.

### 2.2.1 Hidden Markov Models

The currently dominant technology in speech recognition is hidden Markov modelling (HMM). In conventional HMMs the naive Bayes assumption is applied again to estimate the segment-level values $p(x_{s_{i-1}}^{s_i-1}|u_i)$ and $P(S_i|u_i)$ from likelihood estimates calculated over each observation frame. This results in a naive Bayes-type spectral model

$$p(x_{s_{i-1}}^{s_i-1}|u_i = c_k) \approx \prod_{j=s_{i-1}}^{s_i-1} p(x_j|u_i = c_k), \tag{2.5}$$

and in a geometric duration model

$$P(S_i|u_i = c_k) \approx (1 - a_k)a_k^{d-1}, \tag{2.6}$$

where $d = s_i - s_{i-1}$ is the length of the segment and $a_k$ is a value specific to phonetic class $c_k$ (in HMM terminology $a_k$ is a self-transition probability).

The way we introduced the hidden Markov model here may seem slightly unusual. We did so because in the subsequent chapters we will prefer working with a segment-based view on the decoding process, and we wanted to emphasize here that the conventional left-ro-right HMMs are just a special case of it. One reason why the derivation given here seems unexpected is that the usual way of discussing HMMs is to talk about state sequences and not segmentations. But it is quite easy to see that talking about state sequences or segments is practically the same. To see this, recall that the HMM works in such a way that it stays in the same state for a couple of steps, but every now and then it moves to a different state. Those time instances where the HMM changes state can be interpreted as the segment boundaries of the segmental view, while the state-homogeneous sections can be interpreted as the segments themselves. Thus, a bijective mapping is possible between state sequences and segment sequences.

The other slightly unusual thing with our decomposition is that we interpreted the $c_k$ labels and the corresponding segments as phonetic labels and segments. This way equations (2.5) and (2.6) define a 1-state HMM. However, in HMMs the states usually correspond to smaller units – mostly the three pronunciation phases of a phone. But if these units are combined to form phones in the usual left-to-right manner, that is we apply the simplest 3-state models, then the segment-based interpretation is still viable. In this case simply the $c_k$ labels and the segments will correspond to these smaller building units. Also, the language model will require a slight modification, as its items now have to be constructed from phone thirds instead of phones. But if no state-skipping is allowed in the phone models, then this requires only a simple automatic replacement of the phone symbols with the corresponding three building units in the pronunciation dictionary.

## 2.3 Some Critical Remarks and the Key Features of an Envisioned Alternative

As we saw earlier, the conventional HMM is constructed mainly along mathematical lines, practically ignoring what is known about human speech perception. This is why many of its key features can be argued against. Namely, we saw that the HMM is

- frame-based, that is it calculates a likelihood estimate over each observation frame;
- generative, that is the estimates calculated over the data frames are class-conditional likelihoods;
- independence-based, that is the frame-based likelihood estimates are integrated into a segment-based estimate by multiplication, corresponding to the naive Bayes assumption.

The argument against classifying frames (even if we do not make binary decisions but only estimate likelihoods) is that humans cannot really identify such small fragments of sounds, so it seems that trying to extract the phonetic information from these small uniform-sized chunks is not necessarily a good idea (in fact, this type of processing was again introduced simply for technical convenience). More and more research actually indicates that the acoustic correlates of phonetic information are such windows of the time-frequency plane that are longer along the time axis (up to 250 ms) and are narrower along the frequency axis (up to 1-2 octaves) than the conventional frames [73]. It is relatively simple to apply time windows larger than one frame at the probability estimation step, and in fact it is quite usual: ANN-based systems sometimes use time windows as large as 1 second [53], and in conventional HMMs it is a common practice to extract the so-called $\Delta$ and $\Delta\Delta$ features that correspond to derivative-like values estimated over a couple of neighboring frames. Hence one is tempted to claim this issue has been solved, but by using larger – and thus more overlapping – time windows we obviously infringe the independence assumption, so by alleviating one problem we exacerbate another.

Before discussing the problem of the independence assumption, we should mention that the fact that we process the frames – it does not matter which size – uniformly can also be debated. There is an alternative trend that suggests looking for the acoustic correlates of phonetic distinctive features – let them be called acoustic cues [66], acoustic events [67] or acoustic landmarks [48]. As each feature has different correlates, in this approach the signal may go through several quite different processing steps in parallel. This methodology has been repeatedly proposed over the decades, but it never broke into the mainstream – probably because its increased complexity did not result in a significant improvement in recognition performance.

Among the properties of the HMM, the strongest objection is obviously that against the independence assumption of the frames (see, for example [57] and [93]). Knowing the techniques by which the feature vectors are extracted (the $\Delta$ and $\Delta\Delta$ features, RASTA filtering, etc.), it is easy to argue (or even mathematically point out) that the neighboring frames are highly correlated.

But besides the mathematical arguments, one can also argue from an information combination point of view that the product combination rule should be replaced by a more general scheme. To see this, recall that speech as a code is very robust – thanks to its redundancy. That is, the same information is represented in it in several different ways. For example, the identity of a plosive is coded in its burst shape, the formant transitions before and after the burst, and also the linguistic context. In most cases one or a couple of these is ample for a full understanding of what was said. However, the independence-based generative modelling scheme has to account for every observation, and even worse, requires the hypothesis to be supported by each of them. That is, each frame-based likelihood is always made use of, and just one of them giving a value of zero is enough to zero out the whole product. We try to make this model more robust by filtering out everything we cannot deal with in the signal preprocessing step, and by collecting an enormous amount of data in the hope of covering all combinations possible. The author thinks that better robustness could be obtained by replacing the independence-based model by one that looks for such cues that are sufficient to indicate the presence of a feature and combines these by an OR-like rule.

In this thesis two main alternative modelling approaches will be proposed in the spirit of the above critical remarks. One of them will still use frame-based values, but combine them in an OR-like manner instead of the independence-based product rule. This will be the *averaging HMM/ANN hybrid* presented in Chapter 8. The other approach builds complex models to estimate the segmental probabilities 'in one go', instead of estimating them by combining frame-based scores. The latter type of systems will be referred to as *segment-based models*, and they will be presented in Chapter 6.

As regards the third issue, generative modelling, we have already mentioned that applying posterior-based learners instead of generative ones has certain technical advantages, a key one being that they are easier to train discriminatively. Another important issue is that if we intend to experiment with expert combination schemes other than the naive Bayes one, then working with posteriors is much more natural and easier. Because of these motivations most of the models presented and tested in this dissertation are built on posterior estimates. Note that the generative components can be turned into posterior-based ones (by applying Bayes' rule) either at the frame or at the segment-level, so we can create both frame-based and segment-based posterior models. Chapter 6 of this dissertation will deal with posterior-based segmental models. The HMM/ANN hybrids discussed in Chapter 8 belong to the class of frame-based and posterior-based models, while generative segmental models will appear in Chapter 7. And, of course, conventional HMMs represent the fourth combination, frame-based generative models.

Although this dissertation will focus on alternative modelling and combination techniques *within segments*, we should mention here that the multiplication-based combination method can be challenged at higher levels too. For example, at the very first step of decomposition (see Eq. (2.1)) we separated the acoustic and language models into a product form. Notice that their combination by multiplication is an AND-like combination in the sense that a hypothesis $W$ gets a non-zero probability only if there is both acoustic and linguistic evidence for it. This is counter-intuitive, as humans can recognize nonsense speech quite well under clean acoustic conditions, or deduce words totally buried in noise when the linguistic context is very restricted.

Besides human perception arguments, there are also practical problems with the product combination rule: in practice it is known that better recognition results are obtained if the language model score is raised to a proper (usually empirically tuned) power before multiplying it with the acoustic model score. This clearly contradicts the decomposition suggested by Eq. (2.1), and is normally explained by the fact that "the acoustic probability is usually underestimated" [58]. Although raising to a power can indeed be thought of as a compensation for this underestimation, there is also another possible interpretation: that the acoustic an linguistic models are combined via raising to a power and multiplication. The fact that it works better than simple multiplication – which was found to be disputable anyway – supports the idea that other, more flexible combination rules should be tested in place of the simple product rule.

In this spirit, a quite different model could be obtained if we explicitly denoted the language model by $L$ and decomposed $P(W|X, L)$ like so

$$P(W|X, L) \approx f(P(W|X), P(W|L)), \tag{2.7}$$

where $P(W|X)$ is an acoustic expert, $P(W|L)$ is a linguistic expert, and $f(.)$ is some kind of properly chosen expert combination rule. For example, Bourlard proposed combining the acoustic and linguistic evidence by weighted summation, the weights being proportional to the reliability of the experts [15]. Although this sounds quite reasonable, we do not know of any research group that is currently pursuing this approach.

## 2.4 Summary

In this chapter we emphasized that the conventional HMM technology of speech recognition is not a classification algorithm in a strict sense, but a generative model for stochastic random processes. Moreover, most of its properties are chosen for mathematical simplicity instead of psychoacoustic arguments.

Like Jelinek [63], we think that speech recognition should instead be considered as a code breaking activity. In particular, we propose that a proper speech recognition model should look like an expert combination framework with the following key features:

a) The local units of processing (probability or likelihood estimation) should not be frames but some larger (and not necessarily uniform-sized) time-frequency windows.

b) Because of redundancy, it is not necessary to make use of every single data block. Rather, the algorithm should spot and pick out those acoustic events from the observation stream that are informative and reliable and ignore the rest.

c) The local information content of these events should be combined in a fashion that allows for an OR-like fusion of information, and not simply and AND-like scheme (multiplication). Quite probably a two-level (OR-AND) combination method would work best.

In the next chapter we present a generalized speech decoding algorithm that allows us to experiment with combination schemes invented in this spirit. In the rest of the dissertation two types of models will be examined in more detail. The 'posterior segment-based' model satisfies (a), as it models whole phonetic segments as one unit instead of combining them from frames. The 'averaging HMM/ANN hybrid' model satisfies (c), as it combines the frames by averaging, which is an OR-like scheme. In both cases the segment-level experts are combined by multiplication after raising to a power, which is a generalization of the conventional AND-like combination (multiplication).

Although these models are admittedly less general than what the items above would allow and suggest (for example satisfying (b) is not pursued in this dissertation at all), we still think that these first steps are in the right direction.

# Chapter 3

# A Generalized Speech Decoding Algorithm

*"Measure with a micrometer. Mark with a chalk. Cut with an axe."*
*Ray's rule of precision*

In the following we present a general speech decoding scheme. Instead of the conventional generative view, this algorithm interprets the recognition task as a decoding process where certain building blocks have to be found, identified, and the information they provide has to be combined. That is, this approach considers speech recognition as a task of classifier combination integrated in a search process. We will show that this framework is general enough to allow experimentation with combination schemes that satisfy the requirements defined in Chapter 2. In addition, the conventional HMM model can be regarded as a special case of it. Furthermore, the two alternative models proposed in the subsequent chapters are also built on this decoding algorithm. Hence, it can serve as a general framework of all the models and experiments in this dissertation.

## 3.1 The Speech Decoding Algorithm

*Algorithm 1* shows the pseudo-code of our generalized speech decoder. Expressed simply, the algorithm works in the following way. Let us assume that our building blocks are denoted by the elements of the symbol set $\mathcal{C} = \{c_1, ..., c_K\}$. Let the speech signal be given by the series of measurements $X = (x_1, ..., x_T)$. The goal of recognition is to map the speech signal $X$ into a series of symbols $U = (u_1, ..., u_N)$, where $u_i \in \mathcal{C}$. The algorithm works from left to right, and stores its partial results in a priority queue. Having processed the signal up to a certain point $t$, the algorithm looks ahead in time and, from the corresponding measurements, it collects evidence that the next symbol belongs to the time interval being inspected. As neither the exact length nor the identity of the next segment is known, we examine every time index $t' = t + 1, t + 2, ...$ that might be the end point of the segment. Each element $c_k$ of the symbol set is matched to the interval $< t, t' >$, and from each $(t', c_k)$ pair a new hypothesis is formed and

**Algorithm 1** A Generalized Speech Decoding Algorithm

```
solutions := $\emptyset$
hypothesis queue := $h_0(t_0, "", 0)$
// a hypothesis consists of a time index, a phoneme string, and a score
while there is an extendible hypothesis do
  select an extendible hypothesis $H(t, U, w)$ according to some strategy
  if $t = T$ then
    if only the first solution is required then
      return $H$
    else
      put $H$ on the list of solutions
    end if
  end if
  for $t' = t + 1, t + 2, \cdots$ do
    for all $c \in \mathcal{C}$ do
      $w_c := g_1(c, < t, t' >)$ // where $g_1$ estimates the cost of fitting $c$ to $< t, t' >$
                                 // based on the relevant $x_j$ measurements
      $w' := g_2(w, w_c)$ // where $g_2$ is a proper aggregation function
      if pruning-criterion($w_c, w'$) then
        construct a new hypothesis $H'(t', Uc, w')$ and put it in the hypothesis queue
      end if
    end for
    if stopping-criterion($< t, t' >$) then
      break
    end if
  end for
end while
```

put in the hypothesis queue. As every hypothesis has several extensions, this means creating a search tree. By adjusting the hypothesis selection strategy, the pruning and the stopping criteria one can control how the search space is traversed and pruned.

When the whole signal has been processed, the best scoring leaf (or the $N$-best leaves) is (are) returned as the result. The score of a hypothesis is calculated in two steps. First, there is a function ($g_1$) to combine the evidences for each symbol as collected from the local information sources. Second, this local evidence is combined (via $g_2$) with the prefix of the hypothesis to obtain a global score. So, in effect, classifier combination occurs at two levels.

## 3.2 Discussion

Let us now examine the components of the algorithm and suggest some possible choices for them. As regards the selection of the building units, the most reasonable choice is the phoneme since phonemes are the smallest information carrying units of speech (in the sense that the insertion/deletion/substitution of a phoneme can turn a word into another one). Furthermore, in many languages (e.g. Hungarian) there is an almost one-to-one correspondence between phonemes and letters, so working with phonemes is an obvious choice when converting sound to its corresponding written form. Still, smaller or larger units could be used as well. For example, there are arguments that syllables give a more suitable representation of the English language [43]. Going the other way, current recognizers mostly decompose phonemes into three articulation phases [58]. In this dissertation we will always work with phonemes (more precisely, phoneme-like units).

The acoustic information sources $x_j$ display the greatest variation from system to system. Traditionally the acoustic signal $X$ is processed in small uniform-sized (20-50 ms) chunks called 'frames', and the spectral representation of these serves as the $x_j$ input vectors for the model. In HMM systems the spectral data is usually augmented with the $\Delta$ and $\Delta\Delta$ features which are a kind of first and second-order derivative estimates obtained with the help of a couple of neighboring frames [58]. In ANN-based systems it is usual to consider 4-4 neighboring frames from both sides during classification [14]. In TRAP-based systems the frequency bands are processed separately, but from each band data streams as long as 1 second may be used [52]. These examples show that acoustic information lying relatively far from the segment $< t, t' >$ may also be relevant for the identification of it. This is why in the algorithm we generally allowed to make use of any acoustic data that may be relevant for classifying a segment and not only those that fall within the time interval $< t, t' >$.

The choice of the functions $g_1$ and $g_2$ determines what scores are associated with the hypotheses examined during the search. There is a common agreement that the hypothesis evaluation should work on probabilistic grounds. In this case Bayes' decision theorem guarantees optimal performance, and statistical pattern recognition provides methods for approximating the probabilities from training corpora. The acoustic measurement vectors $x_j$ provide their information in the form of $p(x_j|c_k)$ or $P(c_k|x_j)$ estimates. These are then integrated into a segment-level probability by $g_2$; this integration may consider further factors as well, for example phone duration models or prior phone probabilities. Finally, this newly hypothesized phone is attached to the hypothesis prefix with the help of $g_2$. As this function is responsible for concatenating the building units into a string of symbols, linguistic information like phone or word $N$-grams, pronunciation dictionaries or formal grammars can be incorporated into the recognition process via $g_2$. Probabilistic language models would take the form of multiplying factors (transition probabilities), while formal grammars would appear as constraints that reject (associate a probability of zero to) certain unit combinations.

As regards traversing the search space, the strategies fall into two chief categories. One of them is the breadth-first search, which in our case can also be called time-synchronous search. As its name implies, it extends all hypotheses in parallel so that their ending points in time always coincide. The other strategy, best-first search keeps the hypotheses ordered and always extends the most promising path. With a proper heuristic this technique helps one find the best hypothesis in a depth-first manner, avoiding the evaluation of a lot of low-scoring paths in vain. Whichever strategy we choose, the number of hypothesis paths can grow exponentially, especially when the linguistic constraints are weak. In such cases for acceptable execution speed the effective pruning of the search space is crucial. In time-synchronous search the common strategy for pruning is to remove the paths that are farther away from the best hypothesis than a certain 'beam width' [64]. In best-first search (usually called stack decoding) the size of the ordered stack is limited, so paths pushed out of the stack because of their low score are automatically discarded [64].

Our implementation of *algorithm 1* performs a depth-first search. To make the recognition process more efficient we apply multi-stack decoding with several search tree pruning heuristics and a few refinements. More details on these can be found in the publications of my colleague, Gábor Gosztolya [39–42]. In our experiments we did not focus on execution speed; rather we chose such safe pruning parameter settings that guaranteed no degradation in recognition performance.

Lastly, we should mention two main weaknesses of the generalized algorithm. First, it insists on the conventional beads-on-a-string paradigm, that is it assumes that speech is a sequence of phonetic segments. However, this point is debatable, especially in the case of fluent, spontaneous speech. In that case the segments boundaries are frequently hard to identify, and modelling the signal as parallel streams of distinctive features seems more suitable. Although from time to time there is a renewed interest in making use of distinctive features in automatic speech decoding, they have never been able to break in to the mainstream of speech recognition technology. Such a level of generalization would have required too many changes in our algorithm, so we disregarded it.

The other controversial issue is the left-to-right nature of the decoding process. One could argue that it frequently occurs to us humans that some new piece of information at the end of a sentence makes us suddenly understand the very beginning of it. Although in theory it may happen to our algorithm (thanks to language modelling constraints) that a new word at the end corrects all the previous words, in practice it is more common that the search goes wrong at the beginning and later correct words are not able to put it back on the right track. A more reasonable decoding algorithm would first detect reliable 'islands' (either in acoustical or linguistical sense) and then fill in the holes. Although such an island parsing technique would be possible, it would require such big changes that we did not consider it for implementation. As both these further generalizations (distinctive feature technology and island parsing) are definitely unusual in speech recognition, our algorithm is much general even without these than the conventional speech decoding technology.

## 3.3 Special Cases

In the following we discuss how the models occurring in this dissertation can be evaluated in the framework of *Algorithm 1*, and especially how the functions $g_1$ and $g_2$ are chosen in their cases. The exact explanation of why they are chosen so will be given in subsequent chapters.

### 3.3.1 Hidden Markov Models

In spite of its unusual appearance, *Algorithm 1* is not so different from the standard technologies. In particular, its components can be chosen so that it becomes mathematically equivalent to the left-to-right hidden Markov models preferred in large-vocabulary speech recognition. In this setup the set of states of the Markov model will play the role of the symbol set in our algorithm. For the sake of simplicity, let us first assume that we are working with 1-state phone models. Then the phone symbols in $c_k$ directly correspond to the states of the HMM (as each phone model has exactly one state) and any state sequence determines a segmentation based on how long the model stayed in a given state. Owing to this, the probability corresponding to a given segmentation is calculated in two steps. The $g_1$ function of *Algorithm 1* will compute the probability corresponding to a given segment $< t, t' >$ and state $c_k$. That is, according to Eq. (2.5) and Eq. (2.6),

$$g_1(c_k, < t, t' >) = (1 - a_k) a_k^{(t'-t-1)} \cdot \prod_{j=t}^{t'-1} p(x_j | c_k). \tag{3.1}$$

In HMMs the probability corresponding to the whole segmentation is obtained by multiplying the segmental probabilities (cf. Eq. (2.3) and (2.4)). In terms of *Algorithm 1*, this multiplication will be performed by $g_2$. That is, if we have a hypothesis prefix $U$ that fits the input up to time index $t$ with a cost (probability) of $w$, then the cost (probability) associated with the hypothesis that extends $U$ to $Uc_k$ and fits the signal up to time index $t'$ is calculated as

$$g_2(w, g_1(c_k, < t, t' >)) = w \cdot g_1(c_k, < t, t' >). \tag{3.2}$$

Equations (3.1) and (3.2) together define how the acoustic model score for a given phone series $U$ and segmentation (state series) $S$ is evaluated. In practice we usually have a language model component as well, given in the form of $P(U)$. Multiplication by this factor can be incorporated into Eq. (3.2).

In practice, better results are normally obtained if the phones are decomposed into three states, one corresponding to the middle steady-state part, and two others describing the transitional phases before and after. These 3-state HMMs can also be simulated with our generalized model. The only modification required is that in this case the symbols $c_k$ will correspond to these phone thirds (the states of the models). Also, a multiplication by the state transition probabilities $P(u_i|u_{i-1})$ has to be incorporated into $g_2$ (Equation (3.2)). These may also be interpreted as parts of the language model, so taken together we can say that decoding with 3-state models is the same as with 1-state models, but the interpretation of the symbols and the language model both have to be modified slightly.

### 3.3.2 Hybrid HMM/ANN Models

Another class of models that will be used in this thesis is the class of HMM/ANN hybrids [14]. Fortunately, the conventional hybrid model is very similar to the HMM, so its simulation with *Algorithm 1* requires just one small modification. The only change will be that in Eq. (3.1) the estimates of $p(x_j|c_k)$ will be replaced by estimates of $P(c_k|x_j)/P(c_k)$. Nothing else is changed in the formulation, so everything said about HMM simulation will hold true for these models as well.

In Chapter 8 we will introduce the averaging hybrid model. In its simplest configuration Eq. (3.1) is replaced by

$$g_1(c_k, <t, t'>) = \frac{\sum_{j=t}^{t'} P(c_k|x_j)}{(t'-t)} \cdot (\sum_{k=1}^{K} \prod_{j=t}^{t'-1} p(c_k|x_j)) \cdot (1-a_k)a_k^{(t'-t-1)}. \tag{3.3}$$

When compared with Eq. (3.1), we see that in this model the average of the frame-based posteriors is taken instead of the product of the frame-based likelihoods. The geometric duration model is retained, but a new third factor is introduced. It will be called the segment probability value and is calculated from the frame-based posteriors $P(c_k|x_j)$ (for a detailed explanation on why this factor is necessary see Chapter 8). In a more general form of Eq. (3.3) the geometric duration model is replaced by a gamma distribution and the three factors are combined by raising to a power and multiplication. The other components of the decoding process, that is $g_2$ (Equation (3.2)) and combination with the language model work exactly the same way as with the HMM.

### 3.3.3 Segmental Models

The family of segmental models [93] recommends modelling phonemes in one step, instead of estimating their probabilities by combining frame-based scores. In our framework this means that $g_1$ in Eq. (3.1) is replaced by some more sophisticated approximation

$$g_1(c_k, <t, t'>) = p(X_i|c_k), \tag{3.4}$$

where $X_i$ denotes some specific set of acoustic features that is able to represent the whole segment. There are several possibilities available for parametrizing phonetic segments as one unit. The most popular approach is to create special models that fit parametric curves on the feature trajectories [34; 37; 55; 93]. Alternatively, we have the option of applying discriminative models. In this case the formula for $g_1$ becomes

$$g_1(c_k, <t, t'>) = P(c_k|X_i) \cdot P(<t, t'> |X_i), \tag{3.5}$$

where $P(<t, t'> |X_i)$ denotes the probability that the segment being investigated does indeed correspond to a phone. In this dissertation we will focus primarily on this latter type of formulation for segmental models. We will see in Chapter 6 that it is very straightforward to provide such an $X_i$ representation where the segmental model outperforms the conventional HMMs in phone classification. The more problematic component will be the segmentation probability factor $P(< t, t' > |X_i)$ and we will propose several alternative methods to estimate its value.

Whichever technology is applied for modelling the segments, in every cases combination by multiplication will be retained at the level of $g_2$. The independence assumption between the segments seems quite reasonable because the presence of all phonemes is required for the identity of a word. This makes an AND-like combination logical and this is why this combination strategy of HMMs is inherited by the segmental models.

## 3.4 Summary

This chapter presented a general speech decoding scheme that will serve as a framework for implementing all the models studied in the following chapters. Here we explained just how they fit the generalized decoding scheme but did not give a detailed derivation for them. This will be done in the following chapters. The posterior-based segmental model will be introduced and tested in Chapter 6, and the HMM/ANN hybrids will be presented in Chapter 8.

# Chapter 4

# Software and Database Resources

## 4.1 The OASIS System

All the recognition experiments presented in this dissertation were performed using the OASIS Speech Laboratory. This software system was developed by our Research Group on Artificial Intelligence. The system was designed with the aim of creating a general framework that is flexible enough to allow the experimentation with a wide range of techniques in speech recognition. In the following we will give a short overview of the system.

### 4.1.1 The Modular Structure and The Script-based User Interface

The basic execution units of the OASIS Speech Lab are the so-called objects. Similar to the VBScript system of Microsoft, the objects are handled by the component object model (other examples are COM, JavaBeans). Most of the objects may contain further objects and one can assign names to them for identification. On each object services or functions may be defined, and these may depend on other objects given as parameters.

Most of the objects used in the OASIS Speech Lab are so-called modules. The modules are, practically speaking, the kind of special objects that execute some sub-task of the whole signal processing or recognition process. The modules can be interconnected to form a processing work-flow graph – a directed acyclic graph that defines the data flow between the modules. The user's task is to construct a graph from modules and to start the processing. Then the system automatically performs the computations while any of the modules receive a new data block at its input.

The objects of the OASIS Speech Laboratory can be handled through a script-based user interface. Via the Oasis Script Language the user can create the objects, construct a graph from them by specifying their input-output relations, and finally start the processing chain. We will not give a detailed description here of the keywords of the script language and its syntax; rather we will only present an example script at the end of the chapter, for the reader to get an impression of how the system works.

### 4.1.2 Auxiliary Modules

The collective term "auxiliary module" here refers to all those modules that do not perform such scientific tasks as signal processing, machine learning or speech decoding. Rather, they are there to make the system more user-friendly. The most important from this group is the **"DatTraverse"** module. It facilitates the batch processing of files by scanning the lines of a file list and passing its items to the input of the subsequent module one by one. The other group of important auxiliary modules are of those that allow the user to graphically display some data. Spectral maps, feature values and segmentation boundaries can all be displayed using them. A special display module helps visualizing the winning hypothesis of a recognition step.

A third category of important auxiliary modules is of those routines that can read in and write out data blocks. In particular, sound files (in Microsoft PCM WAV format) can be read in and written out, but there are of course many other types of data that can be exported or imported (for example, spectrographic representations may be saved in BMP format). An interesting case is when we save train and test feature vectors in a text file, so that they can later be processed by machine learning algorithms. The system saves these data blocks in the data format common to the C4.5 learner and the UCI data repository [84].

A very special input module is the **"MicIn"** module that can accept sound data from the microphone. It has to be combined with the **"VoiceDetect"** module the detects speech activity – but this is now signal processing and leads us to the next group of modules.

### 4.1.3 Signal Processing and Feature Extraction Modules

The OASIS System implements most of the common signal processing algorithms such as FFT-based spectrum calculation, linear prediction coding and the extraction of cepstral coefficients. The FFT-based spectrum can be transformed to a logarithmic frequency scale by the simulation of Bark-band frequency filters. From these the conventional MFCC coefficients can also be readily obtained. But the HCopy routine of the HTK package [125] can also be called as an external executable, this way guaranteeing a front-end processing identical to that of the HTK.

The pre-processing algorithms listed above all result in a series of vectors – which are all 2-dimensional data sets, and hence in the OASIS system they are called **"Maps"**. The other group of data are of those that are 1-dimensional – in the system they have the collective name **"Feature"**. Of course, any component of a 'map' can be extracted and converted to a 'feature' stream (for example the $i$th cepstral coefficient or the energy of the $i$th spectral band). But such basic features as the short-term energy of the signal can also be calculated. Also, several different processing steps can be applied to the features like mean and deviance normalization, differentiation in time (i. e. the computation of $\Delta$ coefficients), RASTA filtering, adaptive gain control, and so on.

A special characteristic of the OASIS speech decoders is that they require a list of hypothesized segment boundaries – in short, a segmentation. Segmentations are stored in the so-called "**ClusterBound**" objects of the sytem. As the simplest type of segmentations, we can create a 'fake' segmentation of the signal by assuming a possible segment boundary at each frame. Using this fake segmentation in the decoder, the search space will be the same as that of the conventional frame-based recognizers. But we also have the option of constructing sophisticated segmentation algorithms that yield a much sparser segmentation – thus reducing the search space and speeding up the decoding process. In addition, as a special case of segmentations, we can read in the manually positioned phonetic segment boundary markers of a labelled database. This can be useful, for example, when we are interested in evaluating the classification abilities of a learning algorithm.

For segment-based recognition every segment has to be represented by the same number of features, independent of its duration. This feature set is called the segment-based features or acoustic cues. Such "**ACue**" objects can be constructed from frame-based features by calculating their mean, deviance, cosine transform coefficients and so on over the duration of the segment. Another way of creating segment-based features is to extract the value of certain frame-based features at special positions such as the start, end or middle points of the segment. Last, but not least, the duration of the segment is yet another important cue that can be extracted as a segment-based feature.

### 4.1.4 Evaluators

The task of the "Evaluator" modules of the system is to associate probabilities to a given set of a data. In the default case the data is a block of segment-based features, and hence the evaluator returns segment-based phone posteriors or class-conditional phone likelihoods. It is also possible to operate evaluator modules over a set of frame-based features – but, as the decoders work over segments, in that case an additional "**CombineEvaluator**" module is required to fuse the frame-based probability estimates into a segment-based one. Both the segment-based and frame-based evaluators have implementations that work with an artificial neural network (ANN), Gaussian mixture models (GMM), support vector machines (SVM) and a projection pursuit learner (PPL) – but so far we conducted extensive tests only with the ANN and GMM based evaluators.

Currently there are two special kinds of evaluators in the system that do not work with spectral data. One of them is the "**AprioriEvaluator**". As its name suggests, it simply returns the a priori probabilities of the phone classes, based on the frequency counts of the phone labels in the train set. The other one is the "**DurationEvaluator**" that models the phone durations using advanced techniques. Their estimates can be combined with the estimates of the conventional evaluators using proper "**CombineEvaluator**" modules.

### 4.1.5 The Matching Engine

The generalized decoding scheme presented in Chapter 3 is performed by the "**Matching Engine**" component of the system. The task of the matching engine is to traverse all the possible hypotheses (the search space being defined by the segmentation and the phone set), evaluate them (the score of a hypothesis being defined by the acoustic evaluators, the language model and the aggregation strategy inherent to the engine), and to return a ranked list of the best hypotheses. Currently there are three different matching engine modules implemented in the system; they differ in the strategy they apply for traversing the search space. The 'Viterbi Engine' performs a Viterbi-style decoding, that is, a time-synchronous or breadth-first search. The 'Priority Queue Engine' implements stack-decoding that corresponds to a best first-search. The 'Multiple Priority Queue Engine' is a refined version of the previous one in the sense that it stores the hypothesis belonging to different time end points in separate queues.

Although in theory the evaluation of all possible hypotheses guarantees optimal performance, in practice the processing time required for this is prohibitively long. Hence, for fast execution it is very important to find search space pruning heuristics that can throw away unpromising hypotheses without losing the good solutions. In Viterbi encoding it can be done by applying the so-called beam search. In the stack decoding scheme a natural solution is to limit the size of the stacks and thus allow them to discard the least promising partial solutions. These techniques are both implemented in OASIS; more details about efficient decoding in OASIS can be found in the articles by Gábor Gosztolya who developed the matching engines of the system [39–42].

The result of the recognition is evaluated by comparing it to the transcript belonging the sound file in the database. This may be an orthographic or a phonetic transcript, depending on whether we perform word or sentence-level recognition, or just phonetic decoding. In the case of isolated word recognition the comparison is quite simple and can be performed by the **"CompareResult"** module. In the case of recognizing phone or word sequences the comparison corresponds to an edit distance calculation. This can be executed by the **"CompareEditDist"** and the **"CompareSentence"** modules (for word and phone sequences, respectively).

### 4.1.6 Language Models

In most recognition tasks we have linguistic restrictions on the possible phone sequences. The role of the language model is to provide the decoder with the possible phone sequences, along with their corresponding probabilities. In line with the philosophy of the OASIS system, the language models are special kinds of evaluators, but because of their complexity and the conventional separation of the acoustic and language models we discuss them in this separate subsection.

Essentially, we can group the language models into three main categories. In the simplest case we are dealing with an isolated word recognition task. In that case it is sufficient to construct a pronunciation dictionary that simply lists the possible pronunciation(s) of each word. This simple form of language models is implemented by the **"Dictionary"** module of the OASIS system. For efficient storing and decoding, the dictionary is stored in a tree-like compressed form.

Another group of language models are the statistical ones. From these the so-called $N$-grams [58] are the most popular in speech recognition. These estimate the probability of a word or phone based on its 'history', that is the previous $N-1$ words or phones. The OASIS system is capable of supporting the usage of both word and phone $N$-grams. They are implemented via the **"BLanguage"** module of OASIS.

In the most difficult case the language model is formal (grammar-based), or a combination of formal and probabilistic techniques. In Hungarian the creation of such a language model raises special problems because of the agglutinative nature of the language. The **"SimpleRTN"** module of the OASIS system contains an implementation of a complex language model that combines context-free grammars and finite state systems to solve these problems. We intended to keep the structure of this module as similar to the language description techniques of other recognizers as possible. So, when designing this sophisticated language model we initially followed the interface of the Microsoft Speech API. It provides an XML description scheme for the definition of context-free grammars, the words themselves being the terminals of the language. However, in Hungarian listing all the agglutinated forms of a word stem is intractable. As it happens, Hungarian morphology can be well modelled by finite state systems [32]. Moreover, we observed that the agglutinated forms of a stem can be stored in a much smaller space with transducers than with a traditional compression algorithm. This led us to extend the SAPI description so that transducers could be embedded in the place of terminals. This results in a context-free grammar with its terminals being the words recognized by the transducer. Further compression can be achieved by applying special automaton compression algorithms which create the smallest possible transducer that models the same language [68]. Additional savings in storage are possible by storing the resulting transducer in a special data structure [71].

As regards the technical details, the implementation of the storage and traversal of the transducers was relatively straightforward. Managing the context-free grammar, however, required the implementation of a recursive transition network that was built on a stack automaton. We also had to store the actual values of the stack, which required special technical solutions.

The SAPI handles probabilities by allowing the user to associate weights with the right hand side alternatives of a rule. The transducers embedded in our extended scheme also allow the weighting of the transitions. So, by combining the two levels, the system is able to associate a probability to any phone sequence.

Independent of the type of modelling, the interface of the language models is adjusted to suit the requirements of the decoder modules. During the extension of a hypothesis the decoders ask for the possible extensions of a phone sequence, so the task of the language model is to return all the possible subsequent phones of a prefix. Based on this, the interface of the language models consists of two functions, together making it possible to iteratively traverse all the phone sequences of the model. These functions are:

`Enter`: Returns the first possible extension of a prefix, along with its probability (or returns a null pointer if there is no extension).

`Next`: Return the next possible extension of the same prefix, along with its probability (or returns a null pointer if there are no more extensions).

### 4.1.7 An Example Script

```
//this line is needed for displaying the results in a graphical window
sys "win = new Window()";
mod { //listing the elements of the work flow graph
//a boolean variable controlling when we are going to train or test
train = 0;
//reading the phone symbol table from a file
ph = new SimplePhonemes(root.mnt.data.’phonemes.gr’);
//loading the pronunciation dictionary
dict = new Dictionary(root.mnt.data.’dict.txt’, ph);
//this module goes through the elements of the file list one by one
dt = new DatTraverse(root.mnt.’filelist.txt’);
//reading the wave file obtained from the DatTraverse module
wfi = new WavFileIn(dt);
//the following modules calculate the MFCC coefficients along with
//their ‘delta’ and ‘delta-delta’ values; the processing steps are:
//preemphasis - Fourier spectrum - mel filter bank energy estimation -
//cosine transform - delta and delta-delta coefficient calculation
wfp = new PreEmpSB(wfi, 0.97);
sp = new Spectrum(wfp, 400, 160, 512, 1);
fbb = new FilterBankBA(sp, 26);
mfcc = new MFCCBA(fbb, 12, 22, 1);
de1 = new DeltaMapBA(mfcc);
de2 = new DeltaMapBA(de1);
//collecting the coefficients into the feature vector fe[0..38]
for i:0..12 fe[i] = new FBand(mfcc, [i]);
for i:0..12 fe[i+13] = new FBand(de1, [i]);
for i:0..12 fe[i+26] = new FBand(de2, [i]);
//extraction of segment-based acoustic cues; here they are simply the
//feature averages over the segment parts divided in a 3-7-3 ratio
for i:0..11 a[i] = new ACMean(fe[i], 0.0, 0.3);
for i:0..11 b[i] = new ACMean(fe[i], 0.3, 0.7);
for i:0..11 c[i] = new ACMean(fe[i], 0.7, 1.0);
//a further cue will be the segment duration
acd = new ACDuration();
//reading in the annotation file belonging to the wave file; it
//contains the orthographic transcript and may also contain manual
//segmentation and labelling info; the former is required for testing;
//the latter are required for training
df = new DatFile(dt, sp, ph);
if(not train){ //a block for testing the recognizer
//fake clustering by placing a boundary marker at every 2nd frame
cfall = new FakeClusters(2, sp);
//loading the parameters of the ANN-based Evaluator and specifying
//the segmental features as its input
anne = new ANNEvaluator("ann.wts", 1, a[0..11], b[0..11], c[0..11], acd);
//the evaluator results are cached in order to avoid processing
//the same segment twice
canne = new EvalCache2(anne);
//recognition using the Multiple Priority Queue Engine; its input
//modules are the evaluator, the segmentation, and the dictionary
//(along with the phonetic symbol table); the segments are
//restricted to be at least 3 frames and at most 200 ms long;
//the size of the stacks is set to 150
te = new MPQEngine(canne, 0, cfall, dict, ph, 200, 0, 3, 150);
//the resulting word is compared with the orthographic transcript
//given in the annotation file; the results are collected in cr
cr = new CompareResult(te, df);
//a block that displays the spectrogram, the manual segmentation
//markers and the segment boundaries of the winning hypothesis
md = new MapDisplay(sp, parent.win, "SP", 1, 50, 0, 32767);
cbd = new ClusterBoundDisplay(df, parent.win, "CB");
cbd = new HypothesisDisplay(te, cfall, parent.win, "HYP");
//building the graph of the modules and starting processing
build();
start();
//after processing all the files, this module displays the
//recognition statistics collected by CompareResult
? cr;
}
if(train){// a block that extracts training data from the files
//this module goes through all the segments given by the manual
//segmentation, labels them according to the labels given in df,
//and extracts the segmental features from them; the strategy of
//how to create anti-phone examples is specified by the code "162";
//the data extracted is then saved by StringFileOut
mkt = new MKTrain(df, "162", ph, a[0..11], b[0..11], c[0..11], acd);
sfo = new StringFileOut(mkt, "traindata.data");
//building the graph of the modules and starting processing
build();
start();
}
}
```

### 4.1.8 Acknowledgments

The modular framework and the script-based user interface of OASIS were designed and coded by László Felföldi. Most of the auxiliary modules were also written by him. The matching engine was originally designed by András Kocsor and later refined by Gábor Gosztolya. The $N$-gram language model was implemented by Dénes Paczolay. The SAPI-like complex language model is a result of a joint effort by László Felföldi and Attila Kertész-Farkas.

Most of the signal processing, feature extraction and evaluator modules were coded by myself, László Tóth.

## 4.2 Speech Databases Used in this Dissertation

### 4.2.1 The OASIS-Numbers Database

The OASIS-Numbers database consists of spoken Hungarian numbers. It was collected at the Research Group on Artificial Intelligence of the Hungarian Academy of Sciences within the framework of the SZT-IS-10 national grant. Thanks to the governmental support, the database is freely accessible to everyone. The recordings of the corpus are of reasonably good quality, having been recorded with several types of microphones at a sampling rate of 22050 Hz in 16-bit quality. The speakers of the database are mostly university students – 62 males and 49 females.

The utterances recorded can be grouped into two main categories. One of them contains the so-called base words. These correspond to 26 words that are selected so that from them all the Hungarian numbers between 0 and 1,000,000 can be constructed. All the base word recordings of the corpus are manually segmented and labelled at the phone level. Altogether 28 different phonemic labels occur in these transcripts.

The other group of recordings contain randomly chosen numbers between 0 and 1,000,000; these files are intended to be used for testing.

In the selection of the the train and test utterances we followed the recommendation of the database documentation. Thus, 2185 base word recordings were used for training and 1247 random utterances for testing purposes, respectively. For the test utterances we applied the pronunciation dictionaries given with the database. The phonetic transcripts for the compound numbers were simply generated by concatenating the transcripts of the proper base words.

### 4.2.2 The MTBA Hungarian Telephone Speech Database

The MTBA Hungarian Telephone Speech Database is the result of an IKTA project carried out in 2001-2003 by the Department of Informatics, University of Szeged, and the Department of Telecommunications and Media Informatics, Technical University of

Budapest. The MTBA Hungarian Telephone Speech Database is the first Hungarian speech corpus that is publicly available and has a reasonably large size. Besides several groups of recordings that contain isolated words (numbers, company names, city names, etc.), the database contains 6000 sentences recorded from 500 speakers (12 sentences from each). These sentences are relatively long (40-50 phones per sentence) and were selected so that their phonetic transcripts contains evey possible phone connection that occurs in Hungarian. Recordings were made via both mobile and line phones, and the phone calls were organized so that the recordings covered the whole area of the country. The speakers were chosen so that their distribution corresponded to the age and gender distribution of the Hungarian population. All the sentences were manually segmented and labelled at the phone level. A set of 58 phonetic symbols was used for this puprose, but after fusing certain rarely occurring allophones, we worked with only 52 phone classes in the experiments.

For the selection of training and test utterances, we first removed those sentences from the database that contained significant noise and/or half-cut phones (denoted by [spk] and [cut] symbols in the phonetic transcript). From the remaining sentences 1367 were randomly chosen for training purposes (containing 68333 phone instances). For phone recognition tests we used another set of 687 sentences (containing 34532 phone instances). The word recognition results reported in the dissertation are isolated word recognition tests performed on another block of the database that contained city names. All the 500 city names (each pronounced by a different caller) were different. Of the 500 recordings only 431 were employed in the tests as the rest contained significant non-stationary noise or were misread by the caller. The language model employed in the word recognition tests was a simple pronunciation dictionary (created by an automatic phonetic transcription routine) that contained one phonetic transcript for each word and assumed that each of them had equal priors.

In certain experiments reported in this dissertation several parameters will be fine-tuned on the city name recordings. In these cases further testing is required on an independent data block. For this purpose we chose an additional group of 438 recordings from the database, again containing city names, but this time over a smaller vocabulary.

More details about the construction and contents of the MTBA database can be found in [120] (in Hungarian).

### 4.2.3 The BeMe-Children Database

The BeMe-Children database was collected as part of an IKTA project carried out in 2002-2004 by the Department of Informatics, University of Szeged, the Gyula Juhász Teacher Training School of the University of Szeged and the School for the Hearing Impaired in Kaposvár. The goal of the project was the construction of the "SpeechMaster" software package for speech therapy and teaching reading, and the BeMe-Children corpus was originally recorded for the purpose of training and testing the software. The corpus contains recordings from 500 children from the lower classes of elementary schools and from a further 200 pupils with various levels of hearing impairment. In the experiments reported in this dissertation just the former block of data was used, so we give details only on these recordings.

The database contains samples of 100 words from each of the 500 children. From these 40 words were the same in every case and the remaining 60 words varied from speaker to speaker. Only the latter recordings were made use of in the experiments. To construct this data set the most frequent 2000 words were collected from 14 teaching reading books that are currently used in elementary schools. These 2000 words were distributed in the 500*60 recordings according to their frequency in the books, that is the more frequent words occur in more recordings. The recordings were collected in 14 schools all around the country from children of age 6-7, from 250 boys and 250 girls. The database is phonetically segmented and labelled.

For the experiments presented in this dissertation 4000/920 utterances were selected for training/testing purposes, respectively. For language modelling the phonetic transcripts of the 2000 words were created automatically. Owing to the high variability in the children's voices and the recording conditions, and because of the many similar-sounding words in the dictionary, this recognition task appeared to be quite difficult.

More details about the construction and contents of the BeMe-Children database can be found in [103]. The "SpeechMaster" software is described in [4] (both papers are in Hungarian).

# Chapter 5

# Related Work

In this chapter we give a brief survey of those speech recognition research efforts that deviate from the mainstream and have something in common with our work. These will be grouped into three main categories. The first section will overview the segment-based solutions. The second one is theoretically dedicated to posterior-based models in general, but as – apart from very few exceptions[1] – in practice these apply artificial neural nets (ANNs) as posterior estimators, the section will bear the title 'neural networks for speech recognition'. Finally, the third section seeks to collect those examples where some combination rule different from the naive Bayes rule is applied to fuse certain speech-related knowledge sources.

## 5.1 Segment-Based Speech Recognition

The concept of segment-based modelling seems to have arisen around 1989-1993 in several different research groups and under various guises [2; 24; 33; 55; 80]. Usually the false conditional independence assumption of HMMs and their limitations in modelling segmental features – especially duration – are mentioned as the main motivations for developing these models [3][93]. The generative and the posterior-based models appeared practically in parallel, but since then there has been more effort devoted to the generative ones. This is because their connection with HMMs is much more obvious, and because researchers were much more familiar with generative modelling techniques. The family of posterior-based models in practice always corresponds to ANN-based models. In the early days the connection between neural nets and posterior probabilities was not fully understood, hence the early papers simply talk about a decoding paradigm that combines neural nets and dynamic programming [121]. When it became widely known that ANNs can estimate posterior probabilities, researchers' dislike of neural nets quickly ceased, and soon many papers appeared that combined ANNs – now as probability estimators – with the conventional probabilistic HMM framework. A review of ANNs in speech recognition will be given in the next section, and in this section we focus only on those models that apply them in a segment-based manner.

[1] For example, many researchers have recently tried to use Support Vector Machines for this purpose.

The solutions belonging to the family of generative models seek to describe whole feature trajectories along a segment using parametric or nonparametric methods. The Segmental Hidden Markov Model (SHMM) of Gales and Young extends the HMM technology with an additional conditional dependency on the mean of the segment [33]. Holmes and Russel introduce the probabilistic trajectory segmental HMM (PTSHMM) that describes a segment via a stochastic process whose mean varies as a function of time according to the parameters of the trajectory. This model makes a distinction between two types of variability: the extra-segmental variation in the underlying trajectory and the intra-segmental variation of the observations around the trajectory [55]. The parametric trajectory model of Gish and Ng explicitly models the dynamics in a variable duration speech segment by using a time varying trajectory model of the features in the segment. The segment is represented by the trajectories (characterized by a constant, linear or quadratic curve), the residual error covariance around the trajectories, and the number of frames in the segment [34]. This model was later refined by Yun and Oh [126].

The basic idea behind the Mixture Stochastic Trajectory Model (MSTM) of Gong et al. is that speech can be considered as a point that moves in the acoustic observation parameter space as the articulatory system changes. A sequence of parameter vectors corresponding to the subsequent positions of this moving point is called the trajectory of speech. The model assumes that a trajectory can be represented by a fixed number of points; to find these points, a linear sampling of the observation vectors is used [37].

Another valuable source for generative segmental models is the Ph.D. dissertation of Digalakis. He employs the collective term 'Stochastic Segment Model (SSM)' for the class of models he introduces, and then defines and investigates several special cases that are based on the theory of dynamical systems [24]. He is a co-author of the key paper by Ostendorf et al. that provides a nice, unified theoretical view for all the generative segment-based models. Moreover, this paper gives a summary of the possible implementation techniques, such as the constrained mean trajectory models, the conditionally Gaussian models, the dynamical system models, the nonlinear models and the segment-level mixture models. The posterior-based models are just mentioned, along with the comment that "since the area of posterior distribution modelling has received less attention than models based on class-conditional distributions, many of the questions of interest are not yet fully answered, and problems raised here will undoubtedly be addressed with further work" [93].

Before turning our attention to the posterior-based models, there is one more generative segmental system that we should mention: the SUMMIT recognizer of MIT. Although it employs generative phone models, it has many things in common with our approach. First, instead of applying the elaborated trajectory modelling techniques mentioned above, it uses a very simple downsampling strategy to represent the variable-length segments with a fixed number of features. The basic feature set obtained this way is practically the same as the one we are going to use in Chapter 6; however, both their team and ours extend this features set with several additional features that are more or less distinct. Second, they realize the necessity of normalizing the various segmentation paths during evaluation, and introduce the concept of the 'anti-phone' to handle this problem. Although we have admittedly borrowed this term from their work, we model and employ the anti-phones in a fundamentally different way, as we use a posterior-based decomposition and modelling scheme, while their framework is a generative one. They also experimented with pre-segmentation to speed up the recognition process – trying both signal processing and machine learning strategies – but their methods are totally different from the one we propose in Chapter 6. The details of the SUMMIT system were essentially published just in conference papers, masters theses and Ph.D. dissertations, by quite recently they wrote a thorough review paper in Computer Speech and Language [36].

The ANN-based segmental systems usually apply the very simple downsampling strategy (sometimes along with smoothing and a linear transform) to represent the segments with a fixed-length feature vector. As we shall see from our own experiments, even the very basic energy features are enough to produce a classification result similar to those of the HMM's, and with additional well-chosen segment-based features they can be easily outperformed. The really critical issue is how these models normalize the segmentation paths examined during decoding or, in other terms, how they approximate the segmentation probability component. The simplest solution for this is to run a conventional frame-based recognizer, take its $N$-best list of hypotheses, and evaluate the segmental models only over this, as a second pass. The advantage of this is that the hypotheses retained in the N-best list may be considered to have a similarly high segmentation probability. This way this component can be ignored (considered to have the same value) during the ANN-based evaluation of a segmentation. This strategy is followed in the BBN segmental neural network (SNN) of Zavaliagkos et al. [2]. It is interesting to note that in a later version of the system they still use the $N$-best paradigm, but they extend the training of the net by providing it with negative examples encountered during recognition – this can be considered as an precursor of our anti-phone concept, but yet without a full mathematical formalism [3].

A rescoring of The $N$-best output of an HMM recognizer is applied in Kimball's Ph.D. dissertation as well. He investigates generative (the segment-mixture model, SMM) and posterior-based models too; in his terminology the latter framework is called the "classification-in recognition" (CIR) scheme. In his CIR experiments he estimates the posteriors with the help of his generative SMM models. To estimate the segmentation probabilities he combines frame-based boundary probability estimates [69]. A similar strategy is applied in the ANN-based Stochastic Explicit Segment Model (SESM) of Zue et al. They use a special net to estimate boundary probabilities on a frame-by-frame basis, and then the segmentation probability is obtained as the product of the boundary probabilities [80].

Finally, we should mention the work of Verhasselt et al., which is the most recent one on posterior-based models and is the most similar to ours. In their earlier papers they simply call their model a "Dynamic Programming / Multi-layer Perceptron System" [121], but in later versions a precise mathematical formulation is given – with a posterior probability decomposition that is the same as the one we are going to use in Chapter

6. Their research results are summarized in the Speech Communication paper titled "Assessing the importance of the segmentation probability in segment-based speech recognition" [119]. As the title suggests, besides describing their models, the main message of this article is that the segmentation probability factor has an important role in speech decoding. This exactly coincides with our findings in Chapter 6, where we will conclude that the estimation of this component is much more difficult than that of the phone posteriors, and that the poor estimation of this term is responsible for our models not being able to outperform the HMM. To estimate the segmentation probability factor Verhasselt et al. use both a frame-based and a segment-based ANN; the former one estimates segment boundary probabilities at each frame, while the latter one combines these and further segment-based features into a segment-based estimate. Hence, it is similar to our anti-phone model, but is more complicated. It is as if our anti-phone model (see Section 6.5.1) contained the output of the pre-segmentation network (see Section 6.9) among its input features.

## 5.2 Neural Networks for Speech Recognition

In parallel with their rise in popularity in other fields of machine learning in the mid-eighties, artificial neural networks were tested for speech recognition as well. These early attempts applied the nets to classify short-time acoustic-phonetic units such as phones or short words [81]. It was soon realized that although their discriminative and context-modelling abilities are very good, ANNs are poor in handling time-sequences. Several modifications were suggested to adjust the ANN structure to the time-varying nature of speech; the most notable ones being Time-Delay Neural Networks (TDNN) [122] and Recurrent Neural Nets (RNN) [99]. Though with the latter top-performance results were reported, nowadays these technologies do not seem to be pursued any longer. The other group of solutions that addresses the problem of managing time sequences with ANNs proved more successful and more lasting. These proposed that – rather than modifying the neural network itself – they should be combined with another technique that can handle information integration along the time axis. In the early papers dynamic programming is used as a framework to deal with this – in fact, in certain cases the dynamic programming algorithm was directly incorporated into the ANN itself [44]. But dynamic programming was soon abandoned in favor of the hidden Markov scheme, as the latter offers a sound statistical formulation. Of course, this required the realization and promulgation of the fact that ANNs can function as posterior probability estimators [12]. Remarkably quickly after this, the first algorithm appeared for training HMM/ANN complexes globally at the utterance level [5].

Although a number of different ways were suggested for combining ANNs and HMMs, the 'hybrid HMM/ANN' scheme of Bourlard and Morgan [14] became the most wide-spread from among these. In this model the net is applied to estimate the HMM state posterior probabilities. This approach forms the most fundamental class of hybrid models, which had a powerful influence on a number of other approaches. The popularity of these systems is probably due to the fact that they can be very easily interpreted as HMMs that underwent a minor modification: the Gaussian mixture based observation emission likelihood estimates were replaced with ANN-based state posterior estimates. These state posterior estimates, after a division by the state priors, can be considered as (scaled) observation emission likelihood estimates, and this way the hybrid can be traced back to the conventional HMM. This is the interpretation we are going to use in Chapter 8, where it will be explained in more detail and in a more formal way. However, there is another possible view of how these models work: Hennebert et al. showed that the hybrid produces an estimate of the global (i.e. utterance-level) posteriors, and they also provided a global forward-backward training algorithm for the model [51]. In this dissertation the issue of global optimization is not pursued at all; the interested reader should check [65], where a brief comparison of hybrid architectures relying on global discriminative training can be found.

A further interesting combination of neural nets and HMMs is when the net is used as a nonlinear feature transformation algorithm. An early example of this is the work of Bengio et al. [6]. More recently, Ellis at al. apply such an approach, called by them the 'tandem' scheme. Here the output of the ANN – possibly after some further transformation - is used as the input feature vector of a conventional GMM-based recognizer. This way the recognizer itself does not have to be modified at all, hence the advantages of the ANN-based feature set over a conventional one can be demonstrated much more easily and clearly [54].

Another case of neural nets applied in the preprocessing step is when they are used as vector quantizers in combination with a discrete HMM. This class of hybrids is characterized by distinct training steps for the ANN and the HMM. The lack of a combined, global optimization scheme is compensated for by the reduced complexity of the overall machine – mainly due to the use of discrete HMMs versus continuous ones. An example of this type of hybrid can be found in [97], where a feedforward net is trained to perform vector quantization using an unsupervised training algorithm based on the maximum mutual information criterion.

Lastly, we mention that some very good survey papers [116], book chapters [16][115] and books [7][14] are available on the topic of applying neural nets to speech recognition. We recommend these for those readers more interested in this.

## 5.3 Knowledge Source Combination Schemes in Speech Recognition

Although the conditional independence assumption of HMMs has been criticized almost since their introduction, papers that experiment with other knowledge source combination schemes are surprisingly hard to find. The most notable exceptions can be found in the so-called multi-stream recognizers; but before discussing these, we will briefly mention some simpler constructs. One such example is in the work of Saul et al., who propose a two-level model along with a corresponding training algorithm. The model performs an AND-like combination at one level and an OR-like combination at the other level, resulting in a noise-robust integration scheme. Unfortunately, it is applied for the recognition of just one phonetic feature, so it would require a further extension for the classification or recognition of phones [100].

The reader probably gets a similar 'unfinished' feeling with the 'elitist' approach of Chang et al. Here an ANN-based classifier is used, and the term 'elitist' in their paper refers to the very simple concept that the frames where the net's self-confidence is low are simply dropped. But the technique is tested only for the recognition of certain gross phonetic categories, so it is not obvious whether the method could yield improvements in phone or word recognition results [21].

A concept which is more decoding-oriented is the one by Ming et al., who introduce the probabilistic union model. As the name suggest, this model replaces the conjunctive (AND-like) combination of the information sources with a scheme that contains disjunctive (OR-like) steps as well. The model is tested both for the combination of frames [85] and for the combination of frequency bands [61]. Recently, an efficient computation algorithm was proposed for this model by Chan [19].

Fuzzy HMMs and graphical models are two quite different approaches to the generalization of hidden Markov models. The first replaces probability measures by fuzzy measures that have weaker constraints; the resulting model is more flexible and does not require the statistical independence assumption of the HMM [86]. A further example of using fuzzy integration techniques for information aggregation in speech recognition can be found in [20]. Here the fuzzy operator is applied to evaluate the proximity of syllables, and the training of the operator is performed by a gradient-based algorithm. Graphical models represent the dependence relations by a graph-like structure, and the HMM can be considered as a special case of them. Various further special cases – which are all more general than HMMs – such as the dynamic Bayesian multinets and directed graphical models were tested for speech recognition tasks by Bilmes [9]. Both the fuzzy operator-based systems and the graphical model-based ones require quite a lot of involved mathematics, and only time will tell whether these are worthwhile.

The averaging combination rule employed in Chapter 8 of this dissertation is taken from the field of multiple classifier systems. There are other combination techniques preferred by this community, and these sometimes appear in the speech recognition literature too. Such examples are the articles by Kirchhoff and Bilmes [10][72]. These techniques are mostly quite simple and empirical, and there seems to be an aversion to these in electrical engineers who favor those methods with a sound mathematical basis. However, in [10] it is pointed out that the simple combination rules like the sum and product rule are just special cases of the directed graphical models. This yields a formal background for these models, and – besides allowing the introduction of novel combination rules – hopefully also makes them more attractive for the speech community.

The most important sub-field of speech recognition where the need for combination methods arises is that of multi-stream systems. This main motivation behind these systems is the general principle of statistics that if we have several different ways of obtaining an estimate of a value, then usually a better estimate can be obtained by combining these estimates. In speech recognition combination can be applied at several stages. One may combine various feature streams, the probability estimates gotten from these, or even the hypotheses obtained from different recognizers [27]. The combination of hypotheses can occur at the level of frames, phones, words or whole utterances. As for the merging technique, quite a few formulations are suggested and tested in the literature. The simplest of these coincide with those known from multiple classifier systems like the product or sum rules. Another popular combination scheme is that of weighted linear (sometimes log-linear) combination, where the weights are proportional to some kind of reliability measure. Yet another possibility is to merge the experts via an additional machine learning algorithm [105]. The multi-stream systems normally use neural nets, since these yield posterior estimates and these are easier to combine than the conventional GMM-based likelihoods.

A special case of the multi-stream approach is when spectral sub-bands play the role of the individual information streams. This arrangement is motivated by psychoacoustic evidence which shows that humans process the frequency bands quite independently of each other [1]. Thus, researchers of the multi-band paradigm train individual classifiers (almost always neural nets) on the frequency bands, and then merge the score of these for a final hypothesis. The optimal strategy for this merging has not yet been found, so quite a few possible ways are pursued in the papers. The goal is, of course, to find a scheme that can pick out reliable bands, or at least de-emphasize the contribution of the unreliable ones. A measure of reliability can be obtained by assessing the local data mismatch in the sub-bands [90]. To merge the scores of the classifiers a weighted linear combination (either in the linear or in the logarithmic domain) is applied in most cases [90]. Obviously, better results are obtained when the weights have been dynamically adapted to the properties of the underlying signal [90]. Alternatively, an additional neural net may be used for the fusion of the experts [105]. Entropy or mutual information based combination schemes have also been suggested [92]. An exhaustive solution is the 'full combination' method of Bourlard et al. This scheme examines every possible subset of the frequency bands, assuming that there is one good such subset, and handles this subset as a latent variable in the probability decomposition [17]. Although this is mathematically appealing, in practice it has the problem that the number of possible subsets increases exponentially with the number of bands. Finally, we should mention the union model again [61] – as it can be applied in the context of multi-band combination as well. For the reader interested in the technical details of these methods, we recommend the articles by Bourlard [17], Morris [90] and Hagen [45], and the papers of the "Multi-Stream ASR" Session of Eurospeech'99 as a starting point.

# Chapter 6

# A Posterior-Based Segmental Speech Recognition Model

*"It is easier to write ten volumes on theoretical principles than to put one into practice."*
*Leo Tolstoy*

As was summarized in Chapter 2, the HMM technology constructs its utterance-level likelihood scores from frame-based likelihood estimates, and these local estimates are obtained via generative modelling techniques. We also introduced arguments on why the signal should be processed in larger units and that posterior-based modelling techniques should be preferred instead. In this chapter we present an alternative modelling scheme where the utterance-level scores are built directly from segmental posterior estimates. Building this alternative model will require the derivation of a probability decomposition that is different from that in Chapter 2. As we shall see, the resulting model has two main components, the segment-based phone classifier and a segmentation probability component. The proposed segment-based model will be tested on two different databases. It will turn out that the phone classifier component has several technical advantages over the conventional HMM model and that in practice it also significantly outperforms it. Unfortunately, the segmentation probability component proves much more problematic, which is why the second half of the chapter deals with various methods of how this component can be parametrized and estimated.

## 6.1 A Decomposition into Segment-Based Posterior Probabilities

As in previous chapters, in the following the acoustic observation vector will be denoted by $X$ and the possible transcriptions by $W$. We are looking for the kind of decomposition of $P(W|X)$ that is built from segment-based posterior probabilities. One way of obtaining such a decomposition is to start from the conventional derivation. That is, we first separate the language model and acoustic model components by applying Bayes' rule. With this step the optimal solution $W^*$ can be written as

$$W^* = \underset{W}{\text{argmax}}\, P(W|X) = \underset{W}{\text{argmax}}\, \frac{p(X|W)P(W)}{p(X)}. \tag{6.1}$$

Now the latent variable $S$ is introduced, denoting an element from the space of possible segmentations. This way we obtain the approximation

$$p(X|W) = \sum_S p(X,S|W) = \sum_S p(X|S,W)P(S|W) \approx \max_S p(X|S,W)P(S|W). \tag{6.2}$$

In the next step both components are decomposed into the product of segment-level values, based on the naive Bayes assumption:

$$p(X|S,W) \approx \prod_i p(X_i|S_i,u_i) \tag{6.3}$$

and

$$P(S|W) \approx \prod_i P(S_i|u_i), \tag{6.4}$$

where we suppose that $S_i$ denotes the $i$th segment of $S$, $X_i$ denotes the corresponding portion of acoustic data $X$, and that the possible transcriptions $W$ are given as a series of phonemic labels $W = (u_1, u_2, ..., u_N)$.

Note that so far we have followed the conventional derivation described in Chapter 2. This is the point where we now deviate, since we do not intend to decompose the segments further. Moreover, we want to see the segmental posteriors in our description, rather than the class-conditionals. This is why we apply Bayes' rule to $p(X_i|S_i,u_i)$, and get the following:

$$p(X_i|S_i,u_i)P(S_i|u_i) = \frac{P(S_i,u_i|X_i)p(X_i)}{P(S_i,u_i)}P(S_i|u_i) =$$

$$= \frac{P(S_i,u_i|X_i)p(X_i)}{P(S_i|u_i)P(u_i)}P(S_i|u_i) = \frac{P(S_i,u_i|X_i)p(X_i)}{P(u_i)} \tag{6.5}$$

$p(X)$, the prior probability of the observation vector $X$ appears in the denominator of Eq. (6.1). Assuming that $p(X) \approx \prod_i p(X_i)$, $p(X_i)$ cancels out from the formula. A similar assumption cannot be made about $P(W)$ and $P(u_i)$ because the language model is usually not simply a product of phonetic unit probabilities, but operates with word probabilities. In the next section we provide an alternative derivation that does not contain the division by the phone priors; moreover, we will see in Chapter 8 that the necessity of this division is controversial in HMM/ANN hybrids, too. So in practice it seems to be the best solution to test the recognizers both with and without this division.

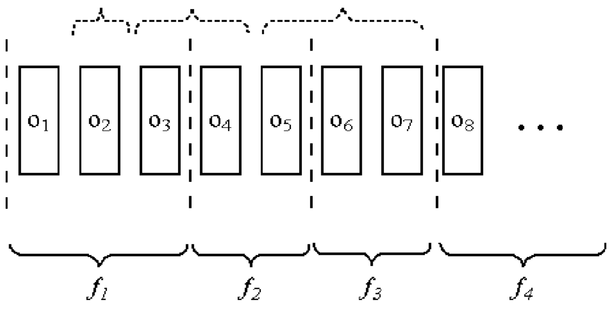

Figure 6.1: An illustration of the relation of observation vectors (solid boxes), phonetic segments (between dotted lines) and anti-phone segments (dotted brackets on top)

After the decomposition and the reductions outlined above, what is left to be evaluated for each segment is $P(S_i, u_i|X_i)$. It can be modelled directly or can be decomposed like so

$$P(S_i, u_i|X_i) \approx P(S_i|X_i)P(u_i|X_i) \tag{6.6}$$

Let us now discuss how these component might be interpreted. Given a segment $X_i$ of the observation vectors, the $P(u_i|X_i)$ component has to estimate the probability that the segment belongs to phone class $c_1, ..., c_K$. Hence, this components is practically responsible for the identification of the segment, which is why we will refer to it as the "phone classifier" module.

What is the role of $P(S_i|X_i)$? A randomly chosen signal segment does not necessarily correspond to a phone, but might be a subsegment of it, or might cover a longer portion of the signal (see Fig. 6.1 for examples). We will refer to these non-phonetic signal segments as "anti-phone" segments, and the component responsible for distinguishing these segments from real phonetic ones as the "anti-phone" model of the system[1]. Note that the phone classifier component is not capable of handling the anti-phone segments, as it assumes that its input belongs to one of the phone classes, and its posterior probability estimates over these classes add up to 1. This is why the estimate $P(S_i|X_i)$ is also required. Alternatively, one could use only one segment-based estimator which is, besides the phone classes, also suitable to report the $(K+1)$th class of the anti-phones. Such an implementation would correspond to modelling $P(S_i, u_i|X_i)$ directly, rather than in the decomposed form given in Eq. (6.6).

## 6.2 A Direct Decomposition

The derivation given in the previous section began the decomposition by applying Bayes' rule. This immediately turned the posteriors into class-conditionals, so in the end we had to apply it again – this time on the segmental components – to convert the class-conditionals back into posteriors. Now we give an alternative derivation that avoids this

[1] This terminology was borrowed from the MIT SUMMIT system. Perhaps it would have been better to call these segments "aphones" rather than "anti-phones".

step. To achieve this we directly represent the language model $L$ on the conditioning side, so we start from $P(W|X, L)$. In the first step the acoustic and language models are separated assuming that

$$P(W|X, L) \approx P(W|X)P(W|L). \tag{6.7}$$

Next the latent variable of segmentations $S$ is introduced, which is eventually removed by marginalization. Formally,

$$P(W|X) = \sum_S P(W, S|X) = \sum_S P(W|S, X)P(S|X) \approx \max_S P(W|S, X)P(S|X). \tag{6.8}$$

For a given $S$, $P(W|S, X)$ can be calculated using equation

$$P(W|X) \approx \prod_i P(u_i|X) \approx \prod_i P(u_i|X_i), \tag{6.9}$$

where the first equation makes the assumption that the phones are independent (we presume that phonetic correlation is modelled by the language model), and the second equation assumes that the identity of a particular phone $u_i$ depends only on the corresponding segment $X_i$ of the acoustic data.

The more problematic issue is with the segmentation probability $P(S|X)$. The simplest way to approximate $P(S|X)$ from the values $P(S_i|X_i)$ is

$$P(S|X) \approx \prod_i P(S_i|X_i), \tag{6.10}$$

but in Section 6.5 more sophisticated estimates will be given for this component.

Note that with this derivation we arrived at the same components $P(u_i|X_i)$ and $P(S_i|X_i)$ as in the previous section, but this time their combination formula does not contain the division by the phone priors $P(u_i)$. In the following we shall provide a very detailed analysis on both of these components. But before proceeding with this, we mention a possible way of generalizing the decomposition and certain implementation questions.

## 6.3 A Generalized Model for Combining the Knowledge Sources

In Chapter 2 we argued that most of the decomposition steps are based on mathematical considerations and are hard to explain perceptually and sometimes are quite counter-intuitive. This might persuade us to regard our components as discriminative knowledge sources, and combine them following some more general scheme borrowed from expert combination theory. Of course, in practice we cannot use very complicated combination techniques, since the more sources we combine the more complex the problem of finding the optimal combination becomes. Fortunately, the problem of knowledge source combination has recently become an area of active research, and optimization techniques that support discriminative modelling are becoming evermore popular in speech recognition [101]. One such possibility is the Discriminative Model Combination scheme of Beyerlein [8], which optimizes a combination scheme of the form:

$$P(W|X, L_1, ..., L_r) \approx \max_S \prod_i P(u_i, S_i|X)^{\alpha_0} P(u_i|L_1)^{\alpha_1} \cdots P(u_i|L_r)^{\alpha_r}, \qquad (6.11)$$

where – besides the acoustic information $X$ – we have $r$ knowledge sources $L_1, ..., L_r$ voting on the symbols $u_i$ in the form of posterior probabilities. Combining them is then performed by raising the values to a power and multiplying them.

In the following we will apply Eq. (6.11) for the combination of our three components – the language model, the phone classifier and the segmentation probability estimator. Note that it is more general than the product combination rule applied earlier only in one aspect: it allows one to raise the estimates of the components to a power. To find the optimal exponents of Eq. (6.11) we apply a global optimization algorithm called SNOBFIT [59].

## 6.4 The Phone Classifier Component

The task of the phone classifier component is to estimate the segmental posterior probabilities $P(u_i|X_i)$. Put another way, it has to associate a posterior probability to any given $(< t, t' >, c_k)$ segment-phone pair and so implement the $g_1$ function of the general decoding scheme of Chapter 3.

There are several ways of parameterizing phonetic segments as one unit. The most popular approach is to create special models that fit parametric curves on the feature trajectories [30; 34; 37; 55; 93]. Another possibility is to represent the variable-length segmental data by a fixed number of segmental features [35][23]. What makes this latter method attractive is that this way all the standard classification algorithms that are able to produce a probabilistic output become applicable to the segmental modelling task. Thus, while the segmental trajectory models are usually built on Gaussian curves, representation by segmental features allows one to use almost any machine learning algorithm. This is why we really prefer this approach. In our studies we reported experiments with a broad range of classifier methods, some of them being very new and not well known by the speech community [74]. Moreover, these classifiers permit the application of feature space transformation methods prior to classification. This introduces further room for experimentation and improvement in classification accuracy.

In the following subsections we will give a short general overview of all three components of segmental posterior modelling: the segmental features, the feature space transformation algorithms, and the classifiers. The exact details – like parameter settings – will be given in the experiments section for each individual experiment.

### 6.4.1 Segmental Features

General-purpose machine learning classifiers assume that each data item is represented by the same number of features. However, speech signals are conventionally processed at a uniform frame rate, resulting in different amounts of data (frame-based measurement vectors) for segments of different duration. This is why we needed a method to convert the frame-based measurements to a segmental feature set that has the same number of features, independent of the segment duration. To obtain a basic segmental feature set, we applied a very simple calculation that was proposed in the MIT SUMMIT system [35], but we found similar solutions from other authors as well [23]. This method takes the frame-level representation of the speech signals and calculates the averages of these features over segment thirds (divided in a 1-2-1 ratio). This calculation practically corresponds to a non-uniform smoothing, and its advantage is that it requires only trifling additional calculations following the computation of the frame-based features.

The feature averages over the segment thirds acts only as a basic feature set and we introduced several further features to improve the classification results (see Section 6.6.1 for a detailed example of how the step-by-step introduction of these features contributes to the overall classification performance). These additional features help not only in separating the phone classes but also in discriminating phones from anti-phones. We introduced the variances of the frame-based features along the segments as further segmental features in the hope of separating anti-phone segments that overlap phonetic boundaries. The derivatives of the frame-based features at the segment boundaries were introduced as additional segmental features so as to help recognize and reject segments with improbable start and end-points.

A special segmental feature is the duration of the phone. We consider it especially important for languages like Hungarian where phonetic duration can play a discriminative role. As our preliminary experiments found duration to be indeed useful, it was employed as a segmental feature in all our experiments.

### 6.4.2 Feature Space Transformations

Feature space transformation algorithms rearrange the input data in a way that hopefully reflects its internal structure more clearly and makes the separation of the classes easier. These methods may aid classification performance and can also reduce the dimensionality of the data. Linear discriminant analysis (LDA), principal component analysis (PCA) and independent component analysis (ICA) are the traditional (linear) transformation techniques [31][102]. Recently the non-linear version of these linear transformations have become a popular research topic in statistical learning theory. Our team performed experiments applying the so-called "kernel non-linearization idea" [117][102] to all of the methods mentioned above and published several papers that apply these pioneering techniques to phone classification. However, both the theoretical investigation of these methods and their empirical testing was carried out by my colleague András Kocsor, so for further details the reader should see his doctoral dissertation of his survey paper [77]. Apart from one case, the results reported in this thesis will all be obtained without applying any feature space transformation. In general, we can say that the transformation algorithms could bring about a 10-20% relative improvement over the phone classification results reported here.

### 6.4.3 Classifiers

For classifying the segments, one may use any general-purpose machine learning algorithm that is able to produce a probabilistic (posterior-like) output. It is well-known that the outputs of an artificial neural network (ANN), under proper conditions, approximate the posteriori probabilities of the classes [96]. But other machine learning methods such as support vector machines (SVM) [117] or decision trees algorithms (C4.5, CaRT) [95][18] can be adjusted so that their outputs can be interpreted as class posteriors. Another algorithm that we implemented and experimented with is the relatively less-known projection pursuit learner (PPL) [60]. In a thorough study [74] we compared the algorithms listed above and found ANNs to be the best choice, taking into consideration such aspects as classification performance, reliability, numerical stability, training and evaluation time. Thus most of the results reported in this dissertation were obtained using artificial neural nets.

In the simplest case the phone classifiers can be trained on a manually segmented and labelled corpus. If there is no such corpus available or some part of the training data is not segmented, then forced alignment can be applied to approximately segment the corpus. This can either be performed by our own – possibly partially trained – model or by a conventional HMM recognizer. Fortunately, all the corpora that we are going to experiment with contain a considerable portion of hand-segmented data, so we always used just manually segmented training data in the experiments we carried out.

## 6.5 The Segmentation Probability Component

As we saw in the previous sections, decomposing $P(W|S,X)$ into a series of segmental phone probabilities $P(u_i|X_i)$ and then modelling these units by a segmental classifier is quite intuitive and straightforward. But understanding the role of the segmentation probability component $P(S|X)$ and how to get a reasonable estimate for it is less clear. A possible practical interpretation is that it corresponds to the aggregation function $g_2$ of the generalized decoding scheme given in Chapter 3, and so its role is to weight the different phone model series and thus normalize the various segmentation paths. An alternative, segment-based interpretation is as follows. Let us assume that the recognizer works with posterior estimates $P(u_i|X_i)$ over segmental units – just the way it was described in the previous section – and examine the workings of the generalized decoding algorithm. We can then see that during recognition the algorithm encounters such $< t, t' >$ segments that do not correspond to real phones. The phone classifier is not automatically able to detect and report these segments. First, it was not trained on such segments; second, it has to return phone posteriors that add up to one and hence has neither a direct output assigned to 'outlier' segments nor any indirect way of reporting them. This is why the segmentation probability factor $P(S|X)$ is required. In the usual generative decomposition (see Chapter 2) this component does not arise, so the importance of this factor in posterior-based segmental models was realized and emphasized only relatively recently [119].

In the following we list several possible ways of modelling $P(S|X)$. Only the last two of these will be pursued further in the experiments, for reasons that are also given below.

- One might try constructing a heuristic 'aggregation function' that combines the segment-based classifier outputs in some weighted manner. Although this is theoretically viable, the resulting function will not guarantee optimal preformance and a bad strategy could lead to such typical errors as the preference of short or long words.

- The easiest solution for avoiding the problems associated with $P(S|X)$ is to run a frame-based (e.g. HMM) recognizer, take the $N$ best paths yielded by it, and evaluate only these paths by using the segmental model [127]. In this case one may assume that the segmentations proposed by the frame-based recognizer all have similarly high probabilities, and so the factor $P(S|X)$ can simply be ignored. Another advantage of this approach is that it enables one to combine the scores the of the frame-based and the segment-based recognizers. The price of this is, of course, increased computational complexity, as two recognition models have to be evaluated instead of just one.

- Alternatively, one may define frame-based scores, the proper combination of which can be used to assess the probability $P(S|X)$ of a segmentation [80]. Hence, this approach applies frame-based calculations for obtaining $P(S|X)$ and segment-based modelling to estimate $P(W|X,S)$.

- The method we are going to opt for here is to construct an estimate of $P(S|X)$ from segmental probabilities $P(S_i|X_i)$. The simplest way to doing this is by multiplying the segment-based estimates, as we did in Eq. (6.10). The main advantage of this approach is that it allows one to model both $P(u_i|X_i)$ and $P(S_i|X_i)$ using the same segment-based features and so the parallel computation of both a segment-based and a frame-based model is not required, unlike in the case of the previous two methods.

  A special problem of this approach is that a manually segmented training corpus contains only examples of real phonetic segments, so we have no natural training samples of the 'anti-phone' segments. These kind of learning tasks are known as 'outlier modelling' or '1-class learning'[2] problems in the machine learning litera-

[2] The task is 1-class learning in the sense that the classes of phonetic segments together form one class that has to be separated from the class of anti-phone segments, and for these latter we have no training examples.

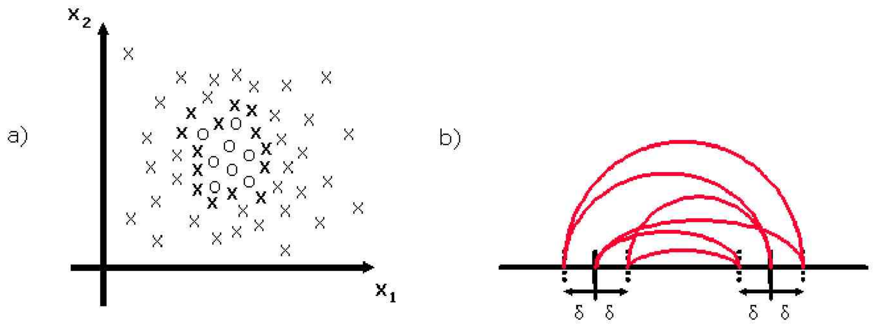

Figure 6.2: a) A 2D illustration of a 1-class learning task with its positive examples ('O's), outliers ('X's), and the outlier samples we try to generate ('X's in bold). b) A phonetic segment and the six anti-phone samples we generate from it

ture. There are dedicated algorithms that handle these tasks; one of these, the replicator neural net, will be introduced in Section 6.5.3.

- With the intention of keeping the computations as fast and as simple as possible, we looked for a way that allowed us to apply the same classifier for modelling both $P(u_i|X_i)$ and $P(S_i|X_i)$. Unfortunately, the conventional perceptron-based neural nets that we were using as a phone classifier could not easily be modified to handle an outlier class. But it was not hard to extend the neural net with an additional output that corresponds to the class of anti-phone segments. This way the same segmental feature set and segmental classifier can be used to describe and model the various phone classes *and* the anti-phone class at the same time. The only problem with this approach is that we had to artificially generate training examples for the anti-phone class, as the training corpus does not naturally contain such annotations. The next section describes the scheme that we followed for this purpose.

### 6.5.1 How to Generate Anti-Phone Examples

Having decided to model the anti-phone segments by extending the phone classifier with a further class responsible for reporting these segments, it was necessary to define a scheme for generating training examples for this new class. In essence, this class corresponds to each such segment that is a part of or a composite of some phonetic segments (see Fig. 6.1), so extracting all these segments from a corpus is clearly not feasible in practice. Hence we needed an algorithm that extracts those anti-phone samples that are most important for separating the anti-phone segments from the phonetic ones. Obviously, these are those points of the sample space that lie close to the border of the two classes, and so we sought to generate such anti-phone examples that hopefully form a hull around the set of points corresponding to phonetic segments (see Fig. 6.2a for a visualization).

To generate such examples we focused on those segments that are 'almost' correct phonetic segments in the sense that both of their boundaries lie close to a real phonetic boundary. Fig. 6.2b shows six such possible anti-phone segments associated with a real phonetic segment. In the experiments these six anti-phone examples were generated for each phone example, their boundaries being 30 ms away from the manually annotated phone segment boundaries.

### 6.5.2 A More Sophisticated Approximation of $P(S|X)$

With the strategy proposed in the previous section, we are able to generate anti-phone examples and hence create segment-based $P(S_i|X_i)$ estimates using a conventional multi-layer neural net. From these estimates an approximation of $P(S|X)$ can be obtained using

$$P(S|X) \approx \prod_i P(S_i|X_i). \qquad (6.12)$$

Unfortunately, in practice we found that this formula does not guarantee a proper normalization between different segmentations. That is, in many cases the system tended to prefer shorter words. It is very difficult to tell whether the formulation of Eq. (6.12) or a poor estimation of the components $P(S_i|X_i)$ themselves is responsible for this, but it led us to try other formulations as well.

One such alternative formula can be obtained if, for the evaluation of a segment, we consider not only the estimate $P(s_i|X_i)$ belonging to this segment, but all of its 'rivals' – precisely, these can be defined as all those possible segments that overlap its middle point [35]. Based on this concept, we arrive at the approximation

$$P(S|X) \approx \prod_i P(S_i|X_i) \prod_{s \in \overline{S}} (1 - P(s|X(s))), \qquad (6.13)$$

where $\overline{S}$ denotes the set of *all other* segments that occur in *any* other segmentation that is evaluated during the decoding process. That is, this formula always makes use of every segment-based estimate, each of these falling into the first or the second product depending on whether it is a part of the segmentation under evaluation or not. This is why we can expect a more balanced behavior from this formulation.

However, in practice the space of all segments is prohibitively large and cannot be efficiently evaluated. So we approximated the second product in (6.13) by considering only those elements of $\overline{S}$ that are "near-misses" of the elements of $S$. More precisely, for a given segment $S_i$ we made use of the anti-phone probability of the four nearest segments in $\overline{S}$ that miss one of the boundaries of $S_i$ (see Fig. 6.3). To get the best performance from the system the component $P(S|X)$ had to be raised to an empirically tuned power – as was suggested in Section 6.3.

### 6.5.3 Anti-Phone Modelling by Replicator Neural Nets

In the previous sections we proposed modelling the anti-phone segments encountered during recognition by extending the segmental classifier with an additional class that corresponds to these segments. This way no significant modification of the segmental

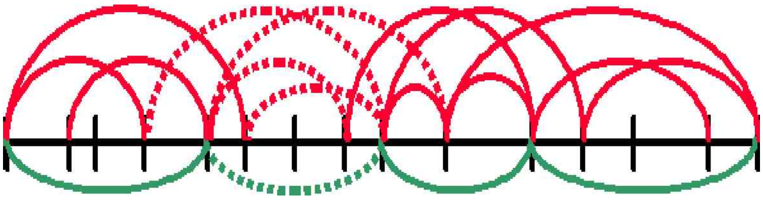

Figure 6.3: An illustration of near-miss segments. The bars along the time scale denote segment boundary hypotheses; the green arcs below denote a possible segmentation; the red arcs above show the corresponding near-miss segments. For the ease of comprehension, one segment and its near misses are displayed with dotted lines

modelling technique is required, that is the same multi-layer neural net technology can be applied. The price is that we have to artificially generate anti-phone samples, but this is tedious, error-prone and significantly increases the training time. In the following we examine an alternative technique, replicator neural nets, that requires relatively little modification in the neural net structure, but promises to handle the anti-phone segments without training examples. That is, this technology falls in the category of the 'outlier modelling' or '1-class learning' techniques.

The scheme proposed in Section 6.5.1 for generating anti-phone examples created six anti-phone samples per phone, and thus seriously increased the amount of data required to train the system. Still, in practice we found that these examples were not representative enough in the sense that the recognizer behaves unexpectedly in many cases (i.e. it accepts obvious outliers as phones). Generating even more outlier examples did not seem really attractive for several reasons. These are the following:

- Apart from obvious cases (e.g. segments that strongly overlap a real boundary) it is not a trivial matter to see how the anti-phone segments should be generated. It might, for example, be that the segments generated following the scheme of Fig. 6.2b could still sound like one phone so, perceptually, they are not really anti-phones. In addition, the manual segmentation of the training corpus may also contain mistakenly positioned boundaries.

- Generating even more anti-phones per segment would cause the training data to be overwhelmed by one class which, as we observed, has a detrimental effect on the learning process.

- One characteristic of speech recognition is that the training databases are enormous. Even the training corpora that are considered 'small' contain hundreds of thousands of phone instances. Creating dozens of outlier examples for each of these really did not seem appealing, especially when one considers the training time involved.

This is why we looked for a method that allows 1-class learning, that is learning from positive examples (in our case phonetic segments) only. Unfortunately, standard perceptron-based neural nets are not suitable for this task – mainly because their responses are not localized. A network with radial basis functions (RBFN) would have

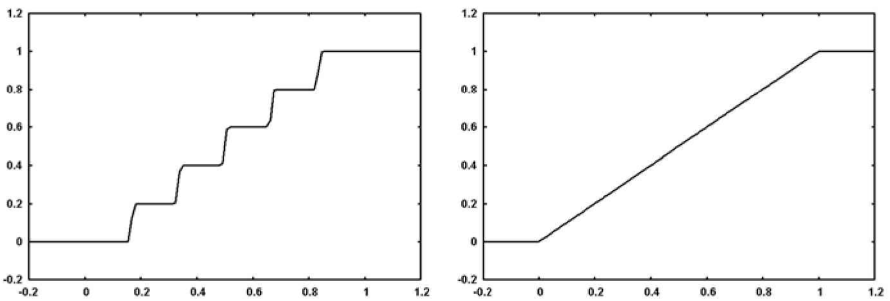

Figure 6.4: The staircase-like activation function and the ramp-like one obtained when increasing the number of steps to infinity

been a possible choice, but we did not want to give up our well-tried multilayer perceptron network. Instead, we sought some simple extension of our current system. This is where replicator neural networks came in the picture.

The basic idea behind a Replicator Neural Net (RNN) [49][50] is simple enough: the input data is also used as the desired output data. Consequently, by minimizing the mean square error during training we force the net to reconstruct its training patterns with the smallest error possible. During testing we hope that outlier patterns (patterns not in the training set) will be less well reproduced by the trained RNN and have a higher reconstruction error. Thus the reconstruction error can be used as a measure of 'outlyingness' of a test pattern.

RNNs were originally introduced in the field of data compression [50]. Hawkins et al. proposed using it for outlier modelling in the field of data mining [49]. In both papers a 5-layer structure is recommended, with a linear output layer and a special staircase-like activation function in the middle layer (see Fig. 6.4). The role of this activation function is to quantize the vector of middle hidden layer outputs into grid points and so arrange the data points into a number of clusters. Although this component plays a theoretically important role in the performance on the RNN, it makes learning by back-propagation practically impossible because its derivative is close to zero almost everywhere. Fortunately, Hecht-Nielsen argues that, by increasing the number of quantization levels to infinity, we arrive at a ramp-like activation function (see Fig. 6.4) by which "real-world problems might be solved" [50].

To go for sure, in the experiments we tried both the staircase, the ramp-like and the traditional sigmoid activation functions in the middle layer. All the other neurons were tested with both sigmoid and $tanh$ activations. In addition, we experimented with varying the number of layers and hidden neurons as well (for the experimental results see Section 6.8).

## 6.6 Phone Recognition Results on the MTBA Corpus

The aim of this and the following sections is to demonstrate the effectiveness of the posterior-based segmental model on real recognition tasks. In this section we perform recognition experiments on the phonetically rich sentences of the MTBA Hungarian Telephone Speech Database (see Chapter 4 for details on the database). As this database contains phonetically balanced sentences recorded from telephone calls from all parts of the country and from people of varying gender and age, we can say that it presents a very general and challenging problem for the acoustic component of any recognizer. Furthermore, this is currently the largest available speech corpus for Hungarian, and very few results have been reported on it so far. Unfortunately, this recognition task is too general in the sense that there was no way of applying any complex (word-level) language model. Hence, the tests reported leave the language model component of the system practically unexploited, and just assess the performance of the acoustic components. That is, we will report only phone recognition results and not word recognition scores. This allows us to focus on the performance of the acoustic models alone, which is advantageous in the sense that the application of a language model may hide the weaknesses of the acoustic level and give us a false impression of the acoustic component's performance.

The performance of the segmental model will be compared to HTK's, which is a sort of standard HMM recognizer in the speech community.

### 6.6.1 Acoustic Features and Phone Classification Scores

For the classification of segments we applied a 2-layer feed-forward neural net with 200 hidden neurons and a softmax output layer. The net was trained with the minimum cross-entropy training criterion, and training was stopped based on a cross-validation criterion [11].

To find a proper segmental feature set we started from a rather simple representation and gradually extended it with additional features. This process of gradual introduction of the features also allows one to demonstrate and understand the importance of the various features. In the following we present the segmental features and the phone classification results obtained with them.

**Baseline segmental features.** As a traditional frame-based representation, energies in 18 Bark-bands were calculated (via FFT, with triangular weighting and cube root compression) at a frame rate of 333 frames/sec[3]. Note that performing a cosine transform on these data vectors would result in the conventional MFCC coefficients. Our earlier results [74] however indicated, that while the conventional Gaussian modelling technique requires this step (for decorrelation), the neural net results do not improve following it. So we worked directly with the Bark-band energies.

[3] This is about three times more than the usual 100 frames/sec. We used this bigger value because in many experiments we found that it resulted in a slightly better classification performance.

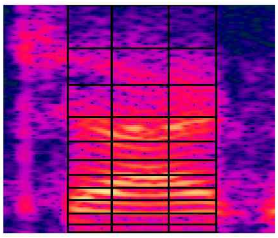

Figure 6.5: An example of how the baseline energy features tile a phonetic segment in the time-frequency plane

As the number of frames within a segment varies and the neural net used for segmental classification requires a fixed number of inputs, a conversion is necessary into a fixed-dimensional segmental feature set. At this stage we followed the very simple idea of the SUMMIT system [35]: the band energies were averaged over phone thirds, which essentially means a kind of non-uniform smoothing. We may say that the inputs to the neural net are really just average energies in cells that tile the time-frequency space in a special manner (see Fig. 6.5).

**The Importance of Phone Duration.** In Hungarian most phones have a 'short' and a 'long' counterpart, thus duration seems to be a vital piece of information. To model the duration we extended the baseline feature set with another feature containing the length of the segment. This way the neural net had the opportunity of forming any kind of durational description, based on the data. The introduction of the duration feature resulted in a significant error rate reduction, as shown in Table 6.1.

**Channel Normalization and Gain Control.** The variance in the transfer characteristics of telephone lines is known to have a detrimental effect on speech recognition. A somewhat similar issue is the varying (average) amplitude of the signal. Many normalization techniques have been suggested to counter these effects. Some of them are off-line, which means that they work *after* the whole signal has been recorded (and, consequently, are not suitable for real-time recognition). As we work directly with band energies and not MFCC coefficients, the most suitable normalization technique was to set the mean of the sentence-level energy to 0 and its variance to zero. This normalization can be performed either on the full signal or for each frequency channel separately.

The on-line algorithms base their processing on the last couple of (centi)seconds. We can normalize the means and deviances by calculating these values just over the most recent data block. Alternatively, we can apply a non-linear adaptive gain control (AGC) algorithm. This applies a 1-pole lowpass filter with time-constant $\tau$ (for more details on these AGC algorithms, see [70]).

As the results in Table 6.2 indicate, off-line methods performed slightly better than on-line ones. Out of the on-line methods the non-linear AGC was the best, with a time-constant of 1 second.

**Adding Observation Context.** In fluent (and fast) speech, phones may become so short that they cannot be recognized without their observation context. Auditory research suggests that approximately a 220-250 ms interval contains information about

| Classification error rate | |
|---|---|
| Baseline features | Baseline plus duration |
| 47.72% | 42.15% |

Table 6.1: Phone classification error rates without and with the duration feature

| *Off-line methods* | CER% |
|---|---|
| Mean and dev. normalization (full spectrum) | 40.27% |
| Mean and dev. normalization (per channel) | 37.75% |
| *On-line methods* | |
| RASTA filtering | 43.86% |
| Mean and dev. norm. (per channel, $\tau = 250ms$) | 41.12% |
| Mean and dev. norm. (per channel, $\tau = 1sec$) | 40.36% |
| Nonlinear AGC (per channel, $\tau = 250ms$) | 39.64% |
| Nonlinear AGC (per channel, $\tau = 1sec$) | 38.49% |

Table 6.2: Phone classification error rates with various channel normalization methods

| | Classification error rate | | |
|---|---|---|---|
| Normalization | $\tau = 150msec$ | $\tau = 250msec$ | $\tau = 1sec$ |
| Off-line mean and dev. norm. | 33.18% | 34.49% | 36.12% |
| Nonlinear AGC (*1sec*) | 33.51% | 34.85% | 36.25% |

Table 6.3: Phone classification error rates when using observation context

| | |
|---|---|
| CER with onset/offset feat. (off-line norm., 150ms obs.cont.) | 32.17% |

Table 6.4: Phone classification error rates with onset/offset features

the identity of a phone, but some researchers use observation windows as large as one second [52]. We tried three different settings of the observation length, defined as the phone length plus the context length. This means that a variable-sized observation context was considered, depending on the segment size. The context was represented by its average energy values in each Bark-band, resulting in two additional feature 'columns' on both sides of the phones (cf. Fig. 6.5). As Table 6.3 shows, the shortest observation length (150ms) performed best, which might be due to the large variance of the context over the training set.

**Adding Onset and Offset Detectors.** Human hearing has cells tuned to detect signal onsets and offsets. These onset and offset detectors may play an important role in the segmentation of a sound stream, especially in finding the boundaries of (certain) phonetic segments. So we implemented an algorithm to simulate these detectors, based on the directions described in [28]. Our detectors calculate the derivatives of the Bark-band energy trajectories and sum them over 3 (6-Bark wide) channels. These curves were evaluated at the phone start and end points and their values were added to the feature set as further features.

We combined these features just with the best feature set found so far. The result shown in Table 6.4 indicates that these new features brought only a marginal improvement in the classification scores. We should mention, however, that they seem to be very important in separating the phone and anti-phone segments.

### 6.6.2 Phone Recognition Results

To separate real phonetic segments from the anti-phones, a two-class neural network was used. The segmental feature set was similar to that of the phone classifier, but instead of the means of the band energy averages the *variances* were employed. This separate modelling of the phone classifier and the anti-phone probability proved slightly better than utilizing the same feature set and classifier for both tasks, as was suggested in Section 6.5. In all other respects the generation of the anti-phone training examples and their utilization in the decoding process followed the scheme that we described in Sections 6.5.1 and 6.5.2.

Although the anti-phone model could be evaluated in isolation if we generated test examples similar to the generation of the training data, its effect can only really be assessed by its influence on the decoding process. Hence we shall now report phone recognition results. These were obtained as follows.

As the vocabulary of the sentences in the corpus is not restricted in any sense, there was no way we could apply any sophisticated (word or morpheme-based) language model. The only thing we could do was to work with a statistical model like phone $N$-grams. From these we chose the simplest possible one, that is every phone was allowed at every position and with the same probability.

The evaluation of the recognition results was performed by comparing the manual phonetic transcription of a sentence to the transcription hypothesized by the recognizer. Clearly, the recognizer output may contain substitution, insertion and deletion errors as well. To count these the two strings are matched by calculating their edit distance with weights (4,3,3) for substitutions, insertions and deletions, respectively, these weights having been proposed by the HTK toolkit [125]. The scores reported below were calculated using the formula

$$Correct = \frac{N - S - D}{N}, \tag{6.14}$$

where $N$ is the number of all phone instances and $S$ and $D$ are the number of substitutions and deletions, respectively. Obviously the recognizer can increase this value by producing many insertion errors as the number of insertions is not included in the formula. To prevent this, the number of insertions was forced to stay around 10-12% by suitably punishing phone transitions in the aggregation formula. This value was suggested by [79].

Table 6.5 lists the recognition scores obtained with and without applying the anti-phone models. The figures clearly show the importance of the anti-phone component, with a slight preference for the complex model. Whether these scores are good or not is difficult to judge per se, so in the following subsection we will furnish some possible bases of comparison.

| Sentence-Level Recognition Scores | | |
|---|---|---|
| Without anti-phones | anti-phone model Eq. (6.12) | anti-phone model Eq. (6.13) |
| 53.44% | 58.74% | 61.34% |

Table 6.5: Phone recognition scores (% correct) on the MTBA corpus

### 6.6.3 Related Work

To our knowledge, apart from us only three teams have used the MTBA corpus so far. Unfortunately, the TSP Lab of the Technical University of Budapest and Hexium Ltd. have to date performed only isolated word or connected word recognition tests over a restricted vocabulary [29][107]. Although the LSA Lab of the Technical University of Budapest has experimented with a task similar to ours, in their tests both the train/test division of the data and the phonetic label set were slightly different. Hence, the phone recognition score of 55-60% they reported [118] allows only a gross comparison.

To obtain a more precise basis for comparison we trained the HTK Toolkit [125], which is a freely available HMM-based recognizer, and is very frequently used to obtain a baseline result when evaluating new technologies. The HTK recognizer was trained with 3-state monophone phone models, all having 15 diagonal Gaussian components (this was reported to be about optimal in [29]). Naturally the same train/test setup and phonetic labelling was employed as with the OASIS system, and the language model was also set up in a similar way. For signal processing we applied the standard 39-component MFCC vector proposed by the HTK manual. With these settings HTK recognized 61.60% of the phones correctly, with an insertion error rate very close to the one obtained with the OASIS system. This means that our system is capable of practically the same recognition performance as other common recognizers.

Alas, HTK cannot measure phone classification directly, so we could not obtain comparative scores to assess the performance of the phone classifier module in isolation. However, in the following section we will compare the phone classifier of our system to an HMM-based recognizer on a number recognition task. Moreover, we recently tested our phone classifier on the TIMIT corpus, for which several classification results are available in the literature [77]. In both cases we found that our segmental representation (along with an ANN or SVM classifier and suitably chosen transformation methods) yields slightly better results than the conventional HMM technology.

## 6.7 Experimental Results on the OASIS-Numbers Database

In this section we evaluate the posterior-based segmental model on the OASIS-Numbers database. This database contains spoken numbers recorded directly via a PC sound cards. Consequently, these recordings are of better quality than the sound files of the MTBA corpus and only a limited number of speech phones occur in them, so much better recognition results can be expected. Moreover, here the vocabulary of the files is restricted, so it is easy to create a simple (dictionary-based) language model for these utterances. Hence in this case we are going to report word-level recognition results as well (see Chapter 4 for a detailed description of the database).

For a comparison, an HMM system was also trained on the same corpus using monophone models (the corpus is too small to train triphones). This recognizer was developed by Máté Szarvas and his colleagues, and will allow us to make comparisons at the phone classification level too. The description of the HMM recognizer can be found in Szarvas [106].

### 6.7.1 Phone Classification Results

The speech segments were represented by a feature set quite similar to that described in Section 6.6.1. The speech signals were converted into critical-band log-energies, and the averages of the 24 critical-band log-energies over the segment thirds (divided in a 1-2-1 ratio) served as baseline segmental features. The variance of the band energies and the onset/offset features were also extracted, in order to support the separation of anti-phones. These latter features were calculated only over 4 wide frequency bands, as this proved sufficient. Thus, including duration, 77 features altogether were used to represent the segments.

For classification we tried both ANN and SVM classifiers. In the experiments below "ANN" means two-layer MLPs trained with back-propagation. The number of hidden neurons was 150 in the phone classification, and 50 in the phone/anti-phone classification tests. In all the experiments with SVMs a second-order polynomial kernel function was applied. Moreover, the effect of applying feature space transformations prior to classification was also tested. We experimented with linear discriminant analysis (LDA) and the kernelized version of it (K-LDA). These transformation methods are beyond the scope of this thesis and more detailed information on them can be found in the papers by András Kocsor [76][77].

Table 6.6 shows the resultant segmental classification errors. In the case of the phone classification task (28 classes) we have a comparative result from the HMM which shows that the segmental discriminative models give significantly better results. In addition, one should notice that the classifiers attained the same performance after LDA and K-LDA, in spite of fact that the transformations considerably reduced the number of features. Similar observations hold for the phone plus anti-phone (29 classes) and phone/anti-phone classification tasks (in the latter case no transformation was applied, as there were only two classes).

### 6.7.2 Word-Level Results

All the word-level recognition experiments were executed with ANN classifiers. Anti-phone modelling and training were performed as described in Sections 6.5.2 and 6.5.1. During recognition the possible segment boundaries were defined by a 5-frame uniform segmentation, that is the recognizer examined all possible segments that can be composed from 5-frame chunks of the signal. Two types of experiments were executed;

| | | No transf.<br>(77 feat.) | LDA<br>( 27 feat.) | K-LDA<br>(27 feat.) |
|---|---|---|---|---|
| 28 phonemes | HMM | 9.34% | — | — |
| | ANN | 7.78% | 7.81% | 5.79% |
| | SVM | 5.81% | 5.12% | 4.59% |
| 28 ph. + antiph. | ANN | 6.78% | 6.87% | 6.54% |
| | SVM | 7.90% | 6.14% | 5.89% |
| phone/ antiph. | ANN | 6.92% | — | — |
| | SVM | 5.10% | — | — |

Table 6.6: Segmental classification error rates

| Segmental Model | | HMM |
|---|---|---|
| No division by the priors | Division by the priors | (3-state monophone models) |
| 2.48% | 0.95% | 0.80% |

Table 6.7: Word error rates

in the first one the segmental probability estimates were divided by the corresponding class priors, while in the second one they were not.

Table 6.7 shows the word-level error rates. The results clearly indicate that the division by the class priors is necessary, as it reduces the word-level error rate by about a factor of 2.5. In Chapter 8 we will see that this division has a very similar effect on the behavior of HMM/ANN hybrids and we will return to this issue there. As regards a comparison with the HMM, we see that the better score obtained from the segmental model is still slightly worse than that for the HMM. As the segmental model was clearly superior in the segment classification task, the word-level scores show that our sophisticated anti-phone modelling technology is still not as good as it should be. In the following two sections we examine two further further ways of improving the segment probability estimates.

## 6.8 Experiments with Replicator Neural Nets

In the experiments with replicator neural nets the BeMe-Children database was utilized. This consists of isolated words pronounced by children from the lower classes of elementary schools, originally recorded for the purpose of a teaching reading software package. This recognition task proved quite difficult owing to the high variability in the children's voices and recording conditions, and because there were many similar-sounding words in the dictionary (for more details on the database see Chapter 4).

For the signal representation we tried two different segmental feature sets. One of them consisted of the 77 features described in the previous Section. The other set was an MFCC-based one, suggested by the literature [23] and contained 61 features.

The speech recognizer was run with three possible arrangements. In one case no anti-phone model was used at all – that is, the ANN was trained only on correct phonetic

| Anti-Phone Model | Feature Set | |
|---|---|---|
| | OASIS | SUMMIT |
| No anti-phone model | 67.17% | 68.58% |
| Anti-phone class /w examples | 72.28% | 77.28% |
| RNN | 72.39% | 75.21% |

Table 6.8: Word recognition accuracies on the two feature sets, depending on the anti-phone model used

segments. In the second arrangement the ANN was extended with an outlier class and its training examples were generated as described in Section 6.5.1; anti-phone modeling was performed as described in Section 6.5.2. Finally, in the third arrangement there was again no outlier class in the ANN, but an additional RNN was used to model the anti-phone segments. As mentioned earlier, the reconstruction error of this net can be used as an indicator of the outlyingness of a sample. We converted it into the (0,1) interval by a sigmoid, so this way it could be interpreted as a probability value.

With the RNN, we first experimented with the special staircase-like activation function of the middle layer. As expected, we could not get back-propagation to converge when using the staircase-like activation function. However, it converged nicely both with the ramp-like and sigmoid activations and these produced very similar results. In all the other layers both sigmoid and $tanh$ activations were employed, and the sigmoid was found to converge somewhat faster. Because of these findings, we applied sigmoid activations in all layers (apart from the linear output layer) in all the subsequent experiments.

When varying the structure of the net, we found no advantage of using five layers. We obtained similar results with just four or three layers, and with a faster training time, so we settled on using a 3-layer model. When varying the number of hidden neurons in its hidden layer, the optimal performance was found to be at about 25 hidden units. It was optimal in the sense that adding more units did not bring any further significant improvement.

The speech recognition results are listed in Table 6.8, both for our classic feature set (OASIS) and the one taken from the literature (SUMMIT). The first thing to notice is that the feature set we developed previously performed worse here than the one suggested by the literature, no matter which anti-phone model was applied. This is probably because our representation was fine-tuned to the MTBA telephone speech recognition task.

As regards the need for an anti-phone component, the definite improvement they bring over the "no anti-phone model" case clearly justifies their importance. It is hard to see, however, why they were less helpful on one feature set than on the other.

Now, let us examine how the RNN performs as an anti-phone model compared to our earlier methodology. In one case it led to exactly the same performance, while in the other it yielded only slightly worse results. This shows that RNN is a viable alternative to our previous method that required the generation of a huge amount of outlier samples and, consequently, a prolonged training time.

## 6.9 Speeding Up the Recognition by Acoustic Pre-Segmentation

The search space the decoder algorithm has to traverse during recognition is the Cartesian product of the set of possible phone sequences and the set of possible segmentations. Hence, by reducing the number of the possible segment boundary positions we could significantly decrease the size of the hypothesis space. Apart from an increase in speed, the recognition scores might also improve by such a reduction due to the removal of such incorrect hypotheses that the engine might otherwise not reject. Unfortunately, the automatic segmentation of speech signals is one of the most difficult problems of speech processing, and finding the perfect segmentation using signal processing techniques alone seems impossible. Actually, this difficulty is one of the reasons why the hidden Markov model that solves the recognition and segmentation in parallel is so successful. Still, we can say that assuming that every phonetic segment may start at any frame and end at any frame – although very safe – is an overkill. In the following we propose a method that yields a relatively sparse segmentation and hence speeds up the decoding process without causing a deterioration in the recognition accuracy.

The signal processing approaches to segmentation all work by measuring the changes in the signal. This strategy is based on the assumption that a change in the underlying phonetic quality always shows itself in a large spectral change. Although it is not always true, this is the best that signal processing can offer. After tedious experimentation we found a certain set of feature extraction steps to be generally the best for detecting these changes. This is the following:

First, the spectrum is decomposed into four bands. The bands were originally chosen to roughly correspond to formant bands, but we later realized that they practically cover 6 Bark wide ranges on the Bark scale. The frequency bands processed by the system are:

$$\begin{gathered}[20Hz; 635Hz],\\ [635Hz; 1790Hz],\\ [1790Hz; 4490Hz],\\ [4490Hz; 11000Hz].\end{gathered}$$

The system detects the changes of energy within these $f_i$ bands. The simplest way to measure changes is by examining the derivative. To avoid the detection of minor changes the data is smoothed first by the simplest possible method, averaging (however, a more sophisticated filter could obviously be used as well). After this, differentiation is approximated simply by calculating the difference between neighboring (or, depending on parameter $\delta$, positioned farther apart) data values:

$$d_i(t) = |f_i(t+\delta) - f_i(t-\delta)|. \tag{6.15}$$

We saw in Section 6.6.1 that the normalization of the channel energies can significantly improve the phone classification results. A similar normalization seemed useful here, too. For this purpose we applied a simple non-linear adaptive gain control (AGC) strategy that has the formulation

$$y_i(t) = \frac{f_i(t)}{1 + K \cdot \hat{f}_i(t)}, \tag{6.16}$$

where $K$ is a constant that controls the strength of non-linearity and $\hat{f}_i$ denotes a smoothed version of the signal. In our implementation this smoothing is performed by a 1-pole IIR filter, so it has only one parameter that can be used to control the strength of the smoothing. The AGC function can be employed both on the band energies and on their derivatives, and besides normalization it also has the advantage that it amplifies the changes.

As we mentioned earlier, simple signal processing methods cannot by themselves solve the segmentation problem. Hence we applied neural nets to automatically learn the phonetic boundaries. As inputs for the net we used the energy values of the four bands along with their derivatives, both with and without AGC processing. We also tried varying the number and width of the frequency channels, but the best results were obtained with the four bands stated above.

The training data was constructed using the manually segmented parts of our databases. Theoretically the frames where the boundary markers are positioned should have been used as positive examples, and the rest of them as negative ones. There are two problems with this simple strategy. First, there would have been many more negative examples than positive ones, and this imbalance can result in training inaccuracies. Second, the manually positioned markers can themselves be inaccurate. To avoid these problems we trained the neural nets to perform regression rather than classification. To construct target values for this regression we fitted the $x^6$ curve between the boundaries in such a way that it took a value of 1 at the boundaries and 0 in the middle of segments (see Fig 6.6). This way we created a continuous transition between the boundaries and the mid-points of the segments. The approximation that the neural net returned for this curve is shown in Fig. 6.7. Only 20 hidden neurons were required to obtain this result, owing to the small number of the features used.

The simplest way to convert the net's output to segment boundaries is to detect its local maxima (see Fig. 6.8). Upon inspection we found that this method is able to detect most of the segment boundaries. It makes mistakes mostly in those cases which are hard to determine even for humans – for example, the vowel-semivowel transitions. The other problematic case was found to be the detection of the end of the burst of voiced plosives. We examined the behavior of the algorithm on the first hundred words of the MTBA city name database (see Chapter 4 for details) and we have found that apart from one case, all the missed boundaries were related to plosives (the remaining case was a vowel-semivowel transition).

To quantify the efficacy of a segmentation algorithm we can compare its proposed boundaries with those of the manual segmentation. For this purpose we applied an edit distance algorithm. This algorithm pairs the boundaries of the two segmentations in such a way that the sum of the distances of the associated pairs is minimal (this can be

| Segmentation Method | WER | Testing Time |
|---|---|---|
| 5-frame uniform segmentation | 0.95% | 549,250 msec |
| ANN-based sparse segmentation | 0.78% | 204,734 msec |

Table 6.9: Word error rates on the OASIS-Numbers Database when using ANN-based pre-segmentation

performed by dynamic programming). Obviously, if the number of the boundaries of the two segmentations differ then there will be boundaries left out. Moreover, we should reject those pairs that have a distance greater than a given threshold. Afterwards, we can count the number of the boundaries that has no pairs in both segmentations, and these scores will correspond to the number of insertion and deletion errors.

We evaluated our segmentation algorithm on the OASIS-Numbers database (for details, see Chapter 4). At a threshold of 30 ms there were 28,262 insertion errors and 88 deletion errors – over 10,488 real boundaries. This means that the algorithm detected about four times more boundaries than there were in reality, but the number of deletion errors was less than 1%. Although this may seem very good at first sight, the problem with deletion errors is that the recognition engine is not able to compensate for these, so just one deletion error can result in the misrecognition of an entire word. For this reason we implemented a safer strategy that positions segment boundaries essentially everywhere, but with a density proportional to the neural net output. An example of the result of this segmentation strategy is shown in Fig. 6.9. With this method both the recognition scores and the decoding speed improved on the OASIS-Numbers database, as Table 6.9 clearly shows.

## 6.10 Conclusions and Summary

Our basic motivation for introducing the posterior-based segmental model was to overcome two main weaknesses of the conventional HMM technology. The segmental framework was chosen because it handles the phones as one unit – instead of building them up from frames – and thus eliminates the flaw caused by the independence assumption. Moreover, we opted for the posterior-based technology instead of the conventional generative one because it gives a better classification performance with fewer parameters. The phone classification tests justified our expectations, as our phone models always outperformed the conventional HMM phone models, even with a very simple segmental feature set. This accords with the findings of other authors (see [23], [35] or [55], for example).

However, when it comes to the recognition of phone series or words, one finds that the segmental model needs an additional component that estimates the probabilities of the segmentations evaluated during decoding. There are several possible ways of explaining the role this factor, but we preferred to interpret it from a segment-based point of view. According to this, the role of this component is to reject those outlier or 'anti-phone' segments that do not correspond to phones, and hence to help the decoding process in finding the proper segmentation of the input. We experimented with two types of technologies to handle these anti-phone segments. One of these was to extend our segment classifier neural net with an additional class for the anti-phones. The main advantage of this approach was that it required almost no change in the phone models, hence their simplicity and evaluation speed could be retained. The price was that an involved algorithm was necessary to generate anti-phone training examples and, of course, the training time also became much longer. Even worse, the resulting recognition scores were not really satisfactory, which led us to introduce a more sophisticated combination scheme for the anti-phone estimates, one introduced in Section 6.5.2. But while this actually brought about a modest improvement, the model lost its appealing conceptual simplicity.

The other approach was to apply a modelling technology that can learn outliers without training samples. We applied replicator neural networks for this purpose. This was motivated by the hope that, by doing this, a relatively simple and efficient model could replace the tedious process of generating and training outlier samples for a traditional MLP. The experiments justified our belief that RNNs indeed have the potential for this task as they yielded a performance similar to our anti-phone based methodology, but without the need for a huge amounts of outlier data.

Although with the application of the anti-phone model or the replicator neural network we could raise the performance of the system to the level of a traditional HMM recognizer, we certainly could not surpass it, as we had hoped. In the number recognition task the segmental system managed to catch up with the HMM only when an ANN-based acoustic pre-segmentation algorithm was applied. Even if its computational cost is negligible, it adds a further complexity to the system and hence makes its less attractive overall.

From our results (and the similar ones found in the literature) we have to conclude that although segmental models can quite easily outperform HMMs at the phone level, this gain can be easily lost at the utterance level. The simple strategies we proposed for modelling the anti-phones were not sufficient to make the system perform better than HMMs in phone and word recognition tasks. It seems that for this the segment-based models have to be combined with frame-based ones (the latter being used just for estimating the segmentation probabilities or yielding phone probability estimates as well). In the following chapter we will try to shed light on why the frame-based HMM is so good at finding the segmentation of a signal.

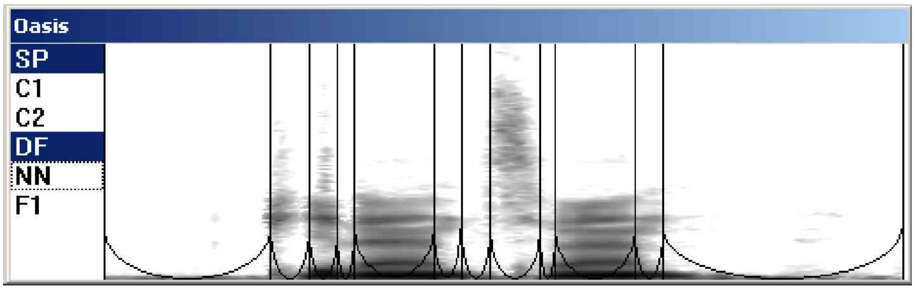

Figure 6.6: The manual segmentation of a sound file and the target function generated from it for the ANN-based regression

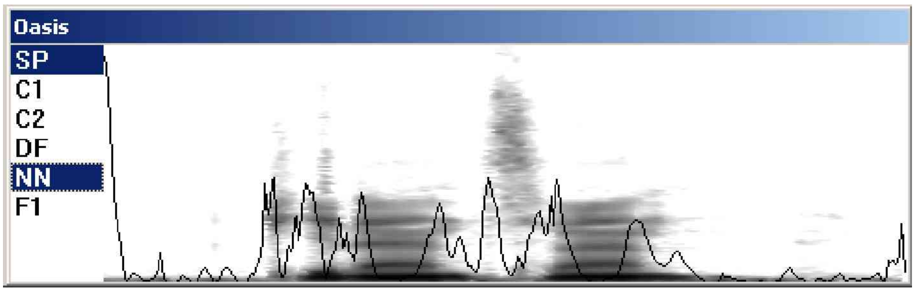

Figure 6.7: The ANN-based estimate of the boundary position probability

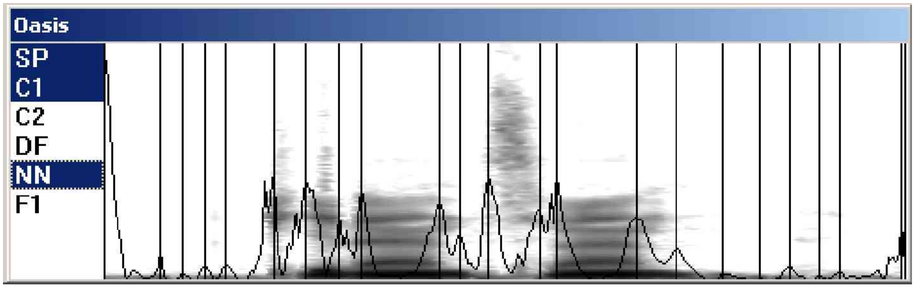

Figure 6.8: Segment boundary hypotheses generated based on the local maxima of the ANN output

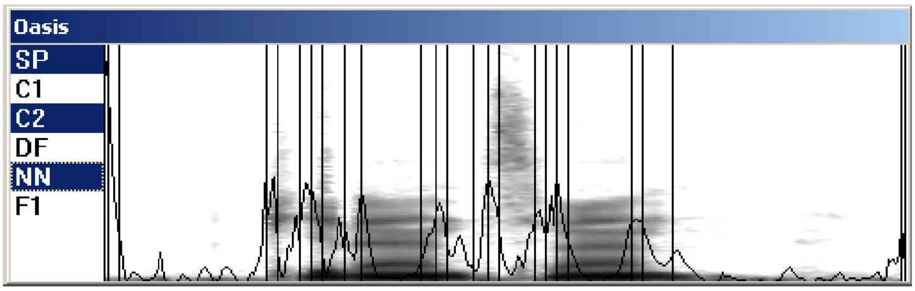

Figure 6.9: Segment boundary hypotheses generated based on the local density of the ANN output

# Chapter 7

# On Naive Bayes in Speech Recognition

*"The purpose of computing is insight, not numbers."*
*Richard W. Hamming*

This dissertation began in Chapter 2 by introducing intuitive arguments about why hidden Markov models are very poor approximators of speech signals and why segment-based modelling seems to be more reasonable. However, in the experiments of Chapter 6 we saw that – in spite of their better phone classification ability – segment-based models have problems even in catching up with HMMs, and even after the introduction of elaborated additional components and a lot of tinkering they can just slightly overcome them. This is in accordance with the literature: besides segmental models, many sophisticated alternatives to HMM have been suggested over the decades, but these have demonstrated only modest improvements and brought no paradigm shift in technology. Having seen the strong arguments against HMM, it may seem amazing that it has been able to preserve its number one position for over two decades. It is also strange that although many authors have criticized the HMM technology, we never saw any of them putting the question the other way round: why does it work then, if it should not?

This is exactly the goal of this chapter: to gain an insight into the behavior of HMMs – especially their incorrect bias due to the naive Bayes assumption – and to understand why it does not significantly harm their performance. To this aim we shall consider the simplest possible HMM structure and compare its performance with a segmental model (also kept as simple as possible). As the segmental model is free of the bias peculiar to naive Bayes, such a comparison can shed light on how this bias influences the recognition process. In addition, to help our understanding, the subtasks of segment classification and finding the best segmentation will be examined separately. From the results we will argue that the bias peculiar to the naive Bayes rule is not really detrimental to its phoneme classification performance. Furthermore, it ensures a consistent behavior in outlier modelling, allowing the efficient management of insertion and deletion errors and thus helping HMMs to find the best phonetic segmentation during decoding.

## 7.1 Introduction

The main appeal of the hidden Markov technology is its mathematical tractability – in particular, that its (locally) optimal parameters can be found relatively easily [58]. But the price for this is that quite simplistic modelling assumptions have to be made that do not necessarily accord with the real behavior of the signals to be modelled. In such cases on may argue that *optimal* performance does not necessarily mean *good* performance, and that the gap between the two basically depends on the rate of the discrepancy – that is, the modelling bias – between the modelling assumptions and the real properties of the data.

When applied to speech signals we have good reasons to think that the modelling bias of HMMs is quite large. This led us to introduce the segmental modelling framework in Chapter 6. We introduced intuitive arguments of why these models have more reasonable modelling assumptions and hence a smaller modelling bias – and we had hoped that this automatically would result in significantly better recognition results. However, the experiments did not justify this and in this chapter we aim to gain an insight into the factors behind it. The question can be raised in two different forms. The more obvious one is to ask why the segment-based model did not perform much better – relative to the HMM? But it is just as good to turn the question around: why is the HMM so good – in an absolute sense – in spite of its unrealistic modelling assumptions? Why does its oversimplified structure, and especially the naive Bayes assumption does not harm its performance?

We will mainly focus on this form of the question, and for the ease of understanding we will examine the two subtasks of speech recognition in isolation. That is, first we shall analyze why HMMs are good at classifying phonetic segments. Then we will investigate why HMMs are also able to find the proper segmentation of the input series.

After collecting our arguments regarding these issues, we will also perform a series of experiments. In some of these we intend to examine what happens when the naive Bayes combination rule – that is, simple multiplication – is replaced by something else. Basically the segment-based based model will represent this case, but combination by averaging will also be tried. In the other group of experiments we will test what happens when we try to compensate for the naive Bayes modelling bias by root taking or imitate a similar bias in the segmental model by powering.

In the subsequent discussions the segment-based and the HMM terminology will be used interchangeably. As was explained in Section 2.2.1, any state sequence can be uniquely associated with a segmentation, where the segments corresponds to those subsequences when the HMM stays in the same state. For the sake of simplicity we will assume 1-state phone models, so the states will directly represent phone classes and the segments will correspond to hypothetized phonetic segments. But the arguments would also be valid for 3-state models after a simple substitution of 'phone thirds' for 'phones'.

## 7.2 Naive Bayes: The Cons

The naive Bayes assumption arises in HMMs in the form of the state-conditional independence assumption of acoustic vectors. In contrast, the neighboring speech frames are obviously correlated as speech is produced by a continuous movement of the articulators. Moreover, many signal processing methods applied in the feature extraction step (e.g. RASTA filtering) increase the correlation as they linearly combine the neighboring data vectors. To top of it all, we usually extend out feature set with the so-called delta features, which are again obtained as a combination of a few neighboring frames [58].

Based on speech perception experiments, we can also argue against combination by multiplication. Namely, it is known that humans can recognize speech quite well even when large portions of the spectral information are removed. In comparison, the production combination rule is too restrictive in the sense that any frame can 'veto' the classification by making the product zero.

As a final argument, classifier combination literature suggests that in general the production rule performs well when the classifiers work on independent features. When the features contain similar information – as in our case – then other schemes like combination by averaging are likely to yield better classification results [108].

## 7.3 Naive Bayes: The Pros

Many have critized the use of the naive Bayes assumption in HMM. But we are unaware of anyone in the speech community putting the question the other way round: why does it work so remarkably well when, in theory, it should not? Fortunately, we can find partial answers in the machine learning literature, since the unexpectedly good behavior of naive Bayes in classification attracted much research in that field. Most pertinently, it has been pointed out that in many cases naive Bayes provides optimal classification even though it incorrectly estimates the probabilities [25]. One such case is when there is full functional dependency among the features [98]. Even when the dependency is not completely deterministic, naive Bayes classification was found to perform nearly optimally in [98]. The explanation is that in these cases all features yield approximately the same probability estimates, so when we combine them by multiplication it is like raising one output to the number of classifiers combined. The resulting estimation tends to underestimate the real probabilities. Besides this, the probability value of the winning class dominates over that of the others. Quoting Hand, "the model will have a tendency to be too confident in its predictions and will tend to produce modes at the extremes 0 and 1" [47]. However, these values still lead to the same classification as raising the estimates to a power preserves rank order.

Knowing that the feature vectors in speech recognition are highly correlated, we might suspect that a similar effect must occur with HMMs. It has indeed been reported that HMMs are "overconfident of their recognition results" [57], and that "primarily due to invalid modelling assumptions, the HMM underestimates the probability of acoustic vector sequences" [124]. These observations support our argument and taken together may explain why HMMs perform well in phone classification in spite of the manifestly false independence assumption.

## 7.4 From Classification to Recognition

The arguments above explain how the HMM is able to correctly classify phonetic segments in spite of the probability estimates being inaccurate. However, as part of the recognition system, the phone models are embedded in an utterance-level model, and their role is not classification, but rather probability estimation! At first sight this seems to invalidate all our arguments for naive Bayes, making the explanation of its efficient classification irrelevant.

Fortunately, we should immediately realize that the utterance-level recognition performance is evaluated by the number of word/phone hits and not by the precision of the probability estimates. So our arguments can be saved if we can also explain why HMMs are able to find the proper segmentation of their inputs. Combined with the reasoning on good segment classification, these arguments together will explain the good recognition performance of HMMs – in spite of their probability estimates being very poor. As Jelinek wrote: "there is no question that HMMs estimate absolute probabilities (densities!) $P(X|W)$ very badly: just try to generate acoustic strings $X$ by HMMs! Yet the relative ratios $P(X|W)/P(X|W')$ between two alternative hypotheses $W$ and $W'$ may well provide a sufficiently accurate approximation for a choice between them" [63].

Let us now try to clarify what happens when we move from classification to recognition. During classification we assumed that the start and end points of the phonemes – that is, the correct segmentation of the signal – was known. Consequently, the only task was to identify the segments. During recognition, however, the proper segmentation also has to be found. Theoretically it is the state transition probabilities that govern what state sequence the HMM goes through during operation. From this one would suspect that it is these probabilities that largely determine which segmentations are preferred during decoding. Quite surprisingly, however, in practice it has been reported by many researchers that they "per se have virtually no effect on recognition performance" [15]! The probable explanation for this is that the observation emission likelihoods are usually many orders of magnitudes smaller than the state transition probabilities.

Having ruled out the state transition probabilities, the only possible explanation left is that in reality it is the naive Bayes combination rule that drives the system towards finding the correct segmentation. This requires preferring real phonetic segments to 'anti-phone' ones. Note that at the frame-level we have neither models dedicated to these non-phonetic segments nor training examples for them. In accordance with the explanations regarding segmental modelling in Chapter 6, this means that at the segment level we are faced with an outlier modelling problem. If our phone model is not able to reject these outliers, it will be prone to commit insertion and deletion errors. That is, it is going to cut the phonemes into more segments or fuse the frames of a segment with neighboring segments.

Let us now examine how the hidden Markov model behaves when it is allowed to evaluate all state sequences and segmentations. Obviously, the naive Bayes rule has a strong preference for short segments. This is because the frame-based likelihoods are very small (non-negative) values, so when we multiply them we will get progressively smaller values for progressively longer segments. In fact, the product of the emission likelihoods would give the highest value if the signal was cut into 1-frame long segments, and to each of these the state with the highest likelihood was selected. In practice this behavior is avoided with the help of the language model that forces the system to fuse neighboring frames: when using pronunciation dictionaries, a phone series containing a lot of phone insertion errors will quite probably be rejected as an impossible one. The preference of short segments becomes more obvious when performing phone recognition without a language model or with a phone $N$-gram only: in such cases usually the introduction of a large phone transition penalty factor is required to counterbalance the insertion errors.

Let us now see how the implicit rejection of the outliers occurs. When forced to fuse neighboring frames, the model will prefer those subsegments in which one of the states provides consistently high values. If the system performs reasonably well at the frame level, these subsegments will mostly coincide with correct phonetic segments. It is also known that the frame-level classification tends to be more stable in the middle of the segments and more inconsistent at the segment boundaries. This will 'push' the model towards fusing frames and thus forming segments close to the central portions of the real phonetic segments and position the state transitions near the real segment boundaries[1].

## 7.5 Experiments

Since our goal here was comprehension and not peak performance, we worked with very simple models. In order to keep the HMM and the segment-based model as similar as possible, the HMM applied 1-state phone units and the segmental model was a generative one. With these simplifications the probability of a phone sequence $U$ over an observation vector $X$ and a phonetic segmentation $S$ is estimated as

$$P(U,S|X) \propto p(X|S,U)P(S|U)P(U) \tag{7.1}$$

in both types of models. As usual, $p(X|S,U)$ is calculated as a product of the corresponding phone unit probabilities $p(X_i|u_i)$. Only the estimation of these components was varied, the language model $P(U)$ and the duration model $P(S|U)$ was the same in every case. This way we ensured that all the differences in the systems' behavior were due to the differences in the component $p(X_i|u_i)$.

To justify our reasoning we conducted experiments that help assess the influence

[1] It is interesting to note that in Chapter 6 those segments that are part of a real segment were also considered anti-phones by us. The HMM, however, will reject only those anti-phones that overlap at least one segment boundary.

of naive Bayes both on classification and recognition performance. For this purpose we replaced the naive Bayes product rule $p(X_i|u_i) \approx \prod_{j=s_{i-1}}^{s_i-1} p(x_j|u_i)$ with alternative combination formulae to obtain an estimate of $p(X_i|u_i)$. From a comparison of these results we hoped to get an indication of how beneficial or detrimental naive Bayes was on classification and on outlier modelling.

In the experiments the "Oasis-Numbers" speech corpus was used (see Chapter 4). For feature extraction we utilized the HCopy routine of the HTK toolkit [125]. We extracted 13 MFCC coefficients from each frame, along with their first and second derivatives. This feature set is the most widely used one in speech recognition [58].

The segment-based model requires an additional step, namely that the variable-length frame-based representation has to be converted into a fixed-dimensional feature set. To achieve this we extracted the simple baseline segmental feature set introduced in Chapter 6. That is, the segments were divided into three parts along the time axis, and each frame-based feature was averaged over these thirds. Additionally, the length of the segment was also included in the segmental feature set.

To model the frame-level and segmental likelihoods Gaussian mixtures were applied, which is again a standard technology in speech recognition. The model parameters were initialized by K-means clustering and trained with Expectation Maximization. 15 Gaussian components performed the best in the frame-level and 10 in the segmental modelling task. In both cases the covariance matrices were kept diagonal.

### 7.5.1 Classification

In the classification experiments we utilized the manual segmentation information of the database. This means that the search part of our decoding algorithm was deactivated by restricting the decoder to evaluate only the correct segmentation. All phoneme priors were assigned equal values in these experiments.

The percentage of correctly classified segments is shown in the first column of Table 7.1. The rows of the table correspond to the various methods applied to obtain the segment-based estimates $p(X_i|u_i)$. Besides the segmental representation and the standard frame-based one that combines the frame-level likelihoods by multiplication, out of curiosity we also tried combination by averaging. Furthermore, we tested two further possibilities. The first one was to compensate for the bias of the product rule by taking the $n$th root of its segmental likelihood estimates, where $n$ is the number of the frame-based scores multiplied (as suggested in [47]). The other idea was to introduce a similar bias into the segmental model by raising its estimates to the $n$th power. These manipulations clearly do not influence classification. However, they result in quite different likelihood estimations that may seriously affect the search process.

We have to emphasize again that our goal here was not to achieve high-performance classification but to compare the two approaches. The product rule combination of the frame-based likelihoods corresponds to a 1-state hidden Markov model, which could be outperformed by the usual 3-state representation. The segmental model could also be improved by adding further features. The results nevertheless reflect quite well the usual findings when comparing segmental models with HMMs, that is the modest superiority of the segmental representation.

We did not mention earlier that the frame-based Gaussian models were able to classify 71.54% of the frames correctly. The product rule brought a substantial improvement compared to this, while averaging outperformed it only modestly. In Chapter 8 we will see that when using neural nets instead of Gaussian mixtures, averaging will yield results quite similar to those of multiplication. A possible explanation is that the Gaussian-based and the ANN-based estimates behave quite differently. In particular, when a frame is classified correctly, the Gaussian-based likelihood estimate of the correct class is much higher than those of the competing ones. And if a frame is misclassified, the likelihood estimate of the correct class is still relatively high. As a consequence, the product rule does not get fooled by the erroneous frames, but the dominance of the correct ones tilts the product in the right direction. Averaging profits less from the high confidence of the correct decisions, and so is more vulnerable to the incorrect ones.

### 7.5.2 Recognition

In the recognition experiments the decoder algorithm was allowed to evaluate every possible segmentation. The segmental probabilities – that is, the $p(X|S,U)$ component of Eq. (7.1) – were calculated exactly as described in at the classification experiments, but now as a part of the whole search process. The duration modelling component $P(S|U)$ was simply implemented by using the same constant (0.5) for each transition probability. We did so because in the earlier discussions this components was judged to have a negligible effect on the decoding process.

As regards language modelling – that is, the prior probabilities $P(U)$ of phone sequences – two extreme cases were tried. In one case every phone was allowed to follow a phone, and with equal probability. This could be called a 'unigram' language model. In the other case the possible phone sequences were restricted to the 26-word vocabulary of the words in the database, with each word being equally probable. This corresponds to a very small vocabulary isolated word recognition task.

The scores reported when using the dictionary are simply the percentage of words recognized correctly. In the case of the unigram model, however, the result of recognition is a phone sequence that, besides misclassifications, can contain insertion and deletion errors as well. These strings were evaluated by comparing them to the manual phonemic transcription by calculating their edit distance [125]. Having obtained the best match, all three types of error are counted and included in the accuracy score.

When testing the product rule with the unigram model we found that – in accordance with our expectations – insertion errors tended to overwhelm the result. We compensated for this by introducing an empirically tuned phone insertion penalty factor. Following [79], this factor was adjusted so that the insertion errors went down to about 10% of the number of phone instances. A similar language model compensation was applied in every case when the number of insertion or deletion errors became seriously imbalanced.

| Phoneme Model | Classification Accuracy | Recognition Acc. | |
|---|---|---|---|
| | | Unigram | Vocabulary |
| Frame-based, product rule | 92.33% | 82.05% | 96.87% |
| Frame-based, averaging rule | 78.04% | — | 86.28% |
| Frame-based, product rule, $n$th root | 92.33% | — | 41.78% |
| Segmental | 94.58% | 46.25% | 87.00% |
| Segmental, $n$th power | 94.58% | 57.99% | 88.29% |

Table 7.1: Classification and recognition accuracies

The results are listed in the last two columns of Table 7.1. The most important finding is that the frame-based model with the product rule performed the best with both language models, and the segmental model could not even come close. This shows that better phoneme classification does not automatically warrant better recognition. This confirms our earlier observations that segmental modelling has difficulties with refusing outliers, and so the segmental recognizers need further component(s) to handle them. This means a further algorithmic and computational burden compared to HMM that 'automagically' handles this problem.

A further observation was that the product rule displayed a very consistent behavior regarding insertion and deletion errors. This means that by adjusting the phone insertion penalty we could easily tune the ratio of insertions and deletions in the Unigram experiments. In comparison, with the averaging rule we were unable to obtain reasonable results because certain phones tended to 'eat up' their neighbors, while some others were cut into lots of small segments. The segmental model displayed a quite similar capricious behavior, although to a lesser extent. Besides insufficient outlier modelling, weak duration modelling may also contribute to this. Although the segmental duration was one of the features and, in theory, the model had the option of making use of it, we noticed that the model still allowed ridiculously long or short segments.

As regards the compensation experiment, taking the $n$th root had a fatal result on recognition, leading to the chaotic behavior just mentioned. However, we have probably overcompensated for the bias of the product rule, so the experiment where we introduced a similar bias into the segmental model might be expected to yield more conclusive results. It showed that raising to a power did not cause any harm. Actually, it led to a slight improvement. This justifies our belief that the special bias of the product rule that gives preference to short segments is in practice helpful in finding the correct segmentation. Generally speaking, it indicates that an incorrect bias that severely punishes long segments performs better in finding the correct segmentation than a model that has not been trained to refuse fake segments and is not really good at duration modelling anyway.

Finally, we should also mention that segmental models are more prone to variance problems due to insufficient data. This is because the segmental models have more parameters than the frame-based ones and in a given training corpus there are of course many more examples of frames than of phones. This may also contribute to the instability of the segmental system.

## 7.6 Conclusions

This chapter sought to gain an insight into why HMM speech recognizers, built on the naive Bayes assumption, perform so well. We argued that speech recognition consists of two subtasks, namely phone classification and outlier modelling, and that the naive Bayes rule does well in both tasks – in spite of the fact that its probability estimates are very poor. As regards classification, we pointed out that the data frames are not independent, but are in fact just the opposite: they are highly correlated. However, we found evidence from the literature that this condition, although being detrimental on the resulting probability estimates, does not necessarily lead to poor classification. But this still does not explain why the recognition process is not fooled by the naive Bayes assumption, since during recognition the probability estimates are used, and not simply the classification results. We explained this here by pointing out that the probability estimates of the naive Bayes rule are such that they get smaller and smaller for longer and longer segments. This biases the model towards a strong preference for short segments, especially where the probability of one class is consistently high. This was clearly justified by the fact that in practice, when only a phone-unigram was used, a phone insertion penalty term had to be introduced, otherwise insertion errors overwhelmed the result. However, by carefully tuning this parameter or by using a pronunciation dictionary, this bias of the model could be nicely counterbalanced, so altogether we can say that naive Bayes reveals itself in a consistent and nicely manageable behavior from an outlier modelling point of view.

To underpin our arguments, a small set of experiments was also carried out where we compared the product rule with a segmental representation. We found that the segmental model performed only slightly better in classification and, in spite of being a better classifier, provided much worse recognition. Overall this shows that the simple product rule, although giving bad likelihood estimates, warrants stable and reliable behavior along with a decent recognition performance. In comparison, segmental recognizers have to take special care of outliers in order to obtain similar or better recognition results. The complications and inconveniences introduced by this fact makes the segmental modelling paradigm less attractive. We think that our arguments and experiments helped to shed light on why HMMs – in spite of their simplicity – behave so well in practice that quite complex alternative models like the segmental model can hardly compete with them.

# Chapter 8

# The Averaging Hybrid HMM/ANN Model

*"She's a model and she's looking good."*
*Kraftwerk*

This chapter deals with HMM/ANN hybrids. As the name suggests, this technology is rooted in the conventional hidden Markov model. The only difference is that the observation emission likelihoods are estimated via neural nets instead of Gaussian mixtures. The hybrid model inherits most of the properties from the HMM, most importantly the frame-oriented processing of data and the product combination rule based on the naive Bayes assumption. However, a significant difference is that, owing to the replacement of ANNs for the GMMs, in the hybrid model class posteriors rather than class-conditional likelihoods are combined. This will be important for us for two reasons. First, this makes the hybrid model much more analogous to the segment-based posterior models of Chapter 6 than the conventional HMM. Second, the multiplication of the frame-based posterior estimates can be interpreted as a multi-expert combination strategy. In the first part of this chapter we examine how this multiplication could be replaced by other combination rules like averaging. We shall argue that averaging actually gives a more accurate estimate of the segmental phone posteriors than the product rule. The second part aims to highlight the analogy between the hybrid model and the segment-based model. In particular, we try to identify the equivalent of the segmentation probability component of the segmental model, which at first look seems to be missing in the hybrid. We will argue that the product rule inherently contains this component. Having extracted the corresponding formula, we combine it with the averaging rule and we find that the resulting hybrid system outperforms the standard one on a phone recognition task and a word recognition task as well. The resulting new hybrid scheme will be named "the averaging hybrid".

## 8.1 Introduction

In Chapter 6 we applied ANNs to estimate the phone posteriors of whole segments. A more conventional way of using ANNs in speech recognition is to apply them for classifying data frames only. The advantage of this is that the resulting system will be more closer to the standard HMM, hence easier to develop and to understand. The drawback is that by doing this we will again face the problem of how to combine the frames – the main motivation that led us to introduce the segmental representation. Luckily, building a model on neural nets rather than on the Gaussian mixtures still has several advantages. First, ANNs yield estimates of phone posteriors that are easier to interpret. In particular, it will allow us to argue that the integration of the frame-based probabilities is practically a multi-expert combination problem, and people in this scientific field prefer to work with posterior probabilities. Second, the hybrid system will be quite similar to the posterior-based segmental model introduced in Chapter 6. To reinforce this, we will identify those components of the hybrid that correspond to the segment-based phone classifier and the segmentation probability estimator components of the segmental model.

Regarding its structure, the hybrid model lies somewhere between the conventional HMM and the segment-based model. We start to analyze its behavior by approaching it from the HMM side. However, as in previous chapters, we prefer to interpret the decoding process as a search over phonetic segmentations rather than state sequences. This will lead us to argue in Section 8.4 that the observation probability calculated over a segment can be interpreted as a segmental phone posterior estimate. In a standard HMM/ANN hybrid this segmental estimate is obtained by multiplication, which is based on the disputable independence assumption. We suggest trying other combination rules from multi-expert technology like averaging and then look at the results.

Before trying the averaging rule within the speech decoding process, we will first assess how accurate the segmental posterior estimates are. Although we are usually only interested in the utterance-level performance, we considered this step an important one towards a more complete understanding of how the model actually works. Evaluating the phone classification performance of the phone models is an obvious idea, but unfortunately not necessarily a proper indicator of the probability estimation accuracy. In Section 8.5 we propose an alternative technique that is based on the investigation of the marginal distributions. Both the phone classification error rate and the proposed method support the idea that averaging yields better segmental estimates than the product rule.

In Section 8.6 we test the averaging phone model in both word recognition and phone recognition. In the former it performs fairly well, but in the latter it fails. This urges us to reconsider our formulas, in particular to look for the equivalent of the segmentation probability component of the segmental model. We argue that although in theory the product of the state transition probabilities corresponds to this, they do not fulfill their task as they are not sufficient to force the model to find the correct segmentation. Instead, the product rule inherently contains a factor that expresses the incoherence of the frame-based experts. More precisely, by examining the (simplified) product rule we find that its posterior estimates do not sum to one, but rather their sum is proportional to a value that can be regarded as an indicator of the incoherence of the frame-based estimators. Hence we claim that the product rule is a very lucky choice in the sense that it automatically accounts for the segmentation probability factor as well. We will borrow this feature from the product rule and combine it with the averaging posterior estimation scheme. We find that with it this novel model outperforms both the conventional HMM and the standard HMM/ANN hybrid in both the phone recognition and word recognition tasks. We will call this new framework the "averaging hybrid".

## 8.2 Database and Baseline Results from HTK

All the results presented in this paper were obtained using the MTBA Hungarian Telephone Speech Database. For training we employed 1367 sentences; for the isolated word recognition tests we used a block of 431 city name recordings. For a more detailed description of the database and the train/test data see Chapter 4.

To have a reference baseline result on this database, we trained standard Gaussian phone models with the well-known Hidden Markov Model Toolkit (HTK) [125]. For testing we used our own decoder (the OASIS MPQEngine – see Chapter 4), but first of course we made certain that it produced results that were practically equivalent to those of the Hvite module of HTK. For preprocessing we applied the default preprocessor configuration. That is, we extracted 13 MFCC coefficients from each frame, along with the corresponding delta and delta-delta values, thus obtaining the usual 39-element feature vector. We should remark here that the same front-end and decoder algorithms were used in all the experiments carried out by us.

First we trained 3-state monophone models by using the manual segmentation, that is no embedded training was applied. The best results were obtained with 9 Gaussians, yielding a word recognition error rate of 7.66% on the city name test set. Embedded training brought only a slight improvement over this, resulting in an error rate of 6.73%.

Hybrid HMM/ANN systems have only one state per phone so, to be comparable, we also created phone models with just one state. The performance dropped dramatically, producing a word error rate of 17.17%. Then we slightly modified our decoder algorithm, forcing it to remain for at least three frames in every state (this can also be interpreted as using a 3-state model with all states sharing the same distribution). Somewhat surprisingly, the error rate decreased significantly, almost attaining that of the 3-state model. Considering that 3-state modelling was invented to account for the three pronunciation phases of phones, it is interesting to see that the bulk of the improvement of switching from a 1-state model to a 3-state one was in fact due to the minimal duration restriction. We obtained slightly better results than the previous one by increasing the minimum duration restriction to 4 frames, so this restriction was applied in every subsequent experiment (see Table 8.1 for a summary of the HMM/GMM results).

It is mentioned in the literature that the state transition probabilities have practically no effect on recognition performance [15]. We replaced all self-transition probabilities

| HMM/GMM Setup | WER |
|---|---|
| 3-state, embedded training | 6.73% |
| 3-state, isolated training | 7.66% |
| 1-state, isolated training | 17.17% |
| 1-state, min.duration=3 | 9.52% |
| 1-state, min.duration=4 | 8.13% |
| 1-state, min.dur=4, shared trans. prob. | 8.13% |

Table 8.1: Word error rates of a conventional HMM/GMM recognizer

| Setup | WER |
|---|---|
| division by the priors | 6.97% |
| no division by the priors | 23.44% |

Table 8.2: Word error rates of a standard HMM/ANN hybrid

by 0.6 in the 1-state model, and indeed found that the word error rate did not change. This finding is important in two respects. First, in HMM/ANN systems it is common practice to use the same fixed value for each transition probability [45], as we did in our HMM/ANN hybrid. After introducing the same simplification in the standard HMM we could be sure that all differences in the behavior of the two types of systems are due to the differences in how they model the observation emission probabilities. Second, this result indicates that it is not the state transition probabilities that drive the model to find the correct segmentation of an observation sequence, but rather the emission probabilities handle this. We will return to this point in Section 8.7.

For the sake of completeness we should mention here that no context-dependent models were tried and, to our knowledge, nobody has yet performed such tests on this database. However, considering the size of the corpus and its richness in phone connections, a very severe amount of parameter tying would be required to construct triphone models, say.

## 8.3 Results Obtained from a Standard HMM/ANN Hybrid

In the past one and half decades several ways of making use of ANNs in speech recognition have been proposed. The most successful approach is probably the HMM/ANN hybrid suggested by Morgan et al. [88]. Here the basic idea is very simple: in a conventional HMM, replace the state-conditional emission likelihood estimates $\hat{p}(x_t|q_k)$ by ANN-based posterior estimates[1]. That is, one should

[1] We will use the caret symbol to denote estimates.

- create frame-based posterior estimates $\hat{P}(q_k|x_t)$ using ANNs.
- apply Bayes' rule to convert the posteriors to state-conditional likelihoods. (This would require calculating $\hat{P}(q_k|x_t)p(x_t)/P(q_k)$, but the $p(x_t)$ values do not influence the maximization process and hence can be discarded. So in practice we only divide $\hat{P}(q_k|x_t)$ by the state priors $P(q_k)$, obtaining a scaled version of the state-conditional likelihoods.)
- integrate the frame-based estimates obtained this way into the usual HMM framework.

Compared to the usual Gaussian mixture based modelling, ANNs offer a more flexible representation with far fewer parameters, and a naturally inherent discriminative training (at the frame level). The training of the neural net can be performed on a manually segmented data set, but a Viterbi-style iterative embedded training is also viable [88]. Moreover, sophisticated training algorithms have been proposed that are discriminative at the utterance level as well [13].

In our experiments we chose to optimize the net at the frame level only, based on the manual segmentation. The neural net applied was a 2-layer MLP with 150 hidden neurons and a softmax output layer. Training was performed by back-propagation, with a cross-validation stopping criterion. The frame-level classification error obtained was 46.47%, and the word error rate of the HMM/ANN hybrid built on this net was 6.97% on the city name database.

Morgan et al. discuss whether the division by the state priors is really necessary, or if the recognizer could work by using the posteriors only [88]. We also carried out tests by omitting the division by the priors. Then the word error rate was 23.44% (see also Table 8.2). Morgan et al. observed a similar drop in the performance and conjectured that it might be due to the fact that, owing to their discriminative nature, hybrid models are more sensitive to discrepancies between the pronunciation dictionary and the real content of an utterance. They suggested that with the proper design of the pronunciation alternatives of a word this performance gap might be reduced [88]. Unfortunately, we could not find any paper in the literature that thoroughly examines this issue; we will return to it later and draw our own conclusions.

## 8.4 Alternative Ways of Combining the Posteriors

In the previous section HMM/ANN hybrids were introduced as a special kind of HMMs. In the following, however, we are going to examine them using the segment-based view that was applied throughout this dissertation. As we saw in Chapter 7, this view lead us to very important conclusions about how HMMs work. HMM/ANN hybrids are much closer to the segment-based models of Chapter 6 than HMMs, as they also work with phone posteriors. Hence, comparing the hybrid model with the segment-based model, and in particular identifying how the two main components of the latter – the segmental phone posterior estimate and the segmentation probability estimate – are computed in the former could be quite instructive.

First of all, let us examine how the hybrid model evaluates a supposed segment. Let $X = (x_1, ..., x_T)$ denote the observation sequence, $U = (u_1, ..., u_N)$ be a sequence of phonetic units over a phone set $\{q_1, ..., q_M\}$, and $S = (s_0, ..., s_N)$ be a segmentation (given as $N + 1$ segment boundary time indices). In a standard HMM each phonetic unit has to account for $x_{s_{i-1}}^{s_i - 1} = (x_{s_{i-1}}, ..., x_{s_i - 1})$, the signal segment mapped to it. More precisely, the model requires an estimate of $p(x_{s_{i-1}}^{s_i-1} | u_i = q_k)$, the likelihood that the given segment was generated by the corresponding phonetic unit. This likelihood is approximated by multiplying the frame-based likelihood estimates:

$$p(x_{s_{i-1}}^{s_i-1} | u_i) \approx \prod_{j=s_{i-1}}^{s_i-1} \hat{p}(x_j | u_i). \tag{8.1}$$

In the hybrid model Bayes' rule is invoked, and the above formula is replaced by

$$p(x_{s_{i-1}}^{s_i-1} | u_i) \approx \prod_{j=s_{i-1}}^{s_i-1} \frac{\hat{P}(u_i | x_j) p(x_j)}{P(u_i)}. \tag{8.2}$$

The $p(x_j)$ terms are common in every hypothesis, so moving them to the other side of the equation does not influence the recognition result. Furthermore, assuming that

$$p(x_{s_{i-1}}^{s_i-1}) = \prod_{j=s_{i-1}}^{s_i-1} p(x_j), \tag{8.3}$$

we obtain

$$\frac{p(x_{s_{i-1}}^{s_i-1} | u_i)}{p(x_{s_{i-1}}^{s_i-1})} \approx \prod_{j=s_{i-1}}^{s_i-1} \frac{\hat{P}(u_i | x_j)}{P(u_i)}, \tag{8.4}$$

the left-hand side being a scaled version of what the HMM requires, and the right-hand side is how we actually compute it in the hybrid model.

Let us now examine Eq. (8.4). For a clearer interpretation we apply Bayes' rule on the left and take the $P(u_i)$ constant out from the product on the right. This leads to

$$\frac{P(u_i | x_{s_{i-1}}^{s_i-1})}{P(u_i)} \approx \frac{\prod_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i | x_j)}{P(u_i)^{l(i)}} \tag{8.5}$$

where $l(i) = s_i - s_{i-1}$ is just a more compact notation for the length of the segment.

If we had multiplied both sides by $P(u_i)$ the formula would have been

$$P(u_i | x_{s_{i-1}}^{s_i-1}) \approx \frac{\prod_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i | x_j)}{P(u_i)^{l(i)-1}}. \tag{8.6}$$

In classifier combination theory Eq. (8.6) is known as the **product rule** for obtaining an estimate of the class posteriors from the estimate of $l(i)$ independent classifiers [108].

Based on this, the operation of the hybrid model can also be interpreted as consisting of the steps

- Take the frame-based posterior estimates.
- Apply the product rule to combine them into a segmental posterior estimate.
- Divide by the class prior to convert the posterior to a scaled version of the class-conditional likelihood.
- Integrate the segmental estimates obtained this way into the HMM framework.

The product rule is derived from the assumptions that both the $\{x_j\}_{j=s_{i-1}}^{s_i-1}$ and the $\{x_j|u_i\}_{j=s_{i-1}}^{s_i-1}$ values are independent within a segment. This is, however, far from being true, and is one of the chief criticisms of the HMM approach [93]. We listed the most important arguments against the independence assumption in Chapter 2, and these arguments lead us to introduce the segment-based modelling framework in Chapter 6.

In Chapter 6 the segment-based posteriors $P(u_i|x_{s_{i-1}}^{s_i-1})$ were estimated in one step by sophisticated segmental models. According to Eq. (8.6), the HMM/ANN hybrids can also be interpreted as if they were working with segment-based estimates $P(u_i|x_{s_{i-1}}^{s_i-1})$, but this time these are obtained via combining frame-based estimates using the product rule. A possible alternative of segment-based modelling is that we insist on working with frame based-scores, but – knowing that the product rule is based on an incorrect assumption – we try other (maybe similarly incorrect) combination rules and see what happens. For example, we could omit the division by the class priors and estimate the segmental posteriors as

$$P(u_i|x_{s_{i-1}}^{s_i-1}) \approx \prod_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i|x_j). \tag{8.7}$$

From now on we will refer to this formula as the **simplified product rule**.

As a third possibility, we could use the **averaging rule** of classifier combination theory:

$$P(u_i|x_{s_{i-1}}^{s_i-1}) \approx \frac{\sum_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i|x_j)}{l(i)}. \tag{8.8}$$

It is important to note that while averaging directly guarantees that the segmental estimates of the different phone classes add up to one (if the frame-based ones do), in the case of the product rule it would hold only if the independence assumption were correct. And we have neither direct, nor indirect guarantees that this is so in the case of the simplified product rule. Hence we will also experiment with versions of the product rules where the sum of the estimates is normalized to one. These will be referred to as the **normalized product rule** and the **normalized simplified product rule**, respectively.

| Combination rule | PhER |
|---|---|
| product rule | 43.19% |
| simplified product rule | 42.44% |
| averaging rule | 43.29% |
| normalized product rule | 43.19% |
| normalized simplified product rule | 42.44% |

Table 8.3: Phone classification errors of the different segmental posterior estimators

## 8.5 Assessing the Accuracy of the Segmental Posterior Estimates

In a HMM speech recognizer we create a hierarchy of embedded models (frames - phones - words - sentences). Such a model can be optimized directly for a minimal error rate at the highest hierarchical level, in which case the parameters of the building units are tuned implicitly. This sort of embedded training is very appealing if we are interested only in the best global performance. Theoretically, however, it might happen that global optimality requires that the embedded components behave suboptimally. Even worse, it might force the components to represent different things that they were intended to by the creator of the model. For example, a HMM might have very good utterance-level performance even if the values returned by its phone models had nothing to do with phone likelihoods or posteriors. This may not bother us if performance is all that matters, but it is most discouraging if we expect speech recognition to mimic human speech perception in all its components. This is our motivation here to examine and compare the five combination rules from the viewpoint of how well they fulfill their intended task – the modelling of the phone posteriors.

The motivation behind probabilistic pattern recognition is the theorem that by having the class posteriors we could do optimal classification (in the sense of minimal risk) [26]. In practice, however, we can only estimate the posteriors. First, the models themselves may be built on incorrect assumptions, and hence be biased. Second, the training algorithms would require an infinite amount of training data for an ideal approximation. Third, because of categorical perception, human subjects are not able to express the identity of, for example, a phonetic segment in terms of probabilities. Hence we cannot directly check the accuracy of our estimates. We have only an indirect indicator, the classification error rate. Examining the phone classification performance of the five posterior models, we obtained the results shown in Table 8.3. These scores suggest that averaging is in practice just as good as the product rule, and the simplified product rule, although not justified theoretically, is slightly better than the other two.

Unfortunately, good classification does not necessarily mean a good estimate of the probabilities. It is easy to see this when we realize that correct classification requires only that the correct class had the maximal probability score; the values themselves may not be related to the real probabilities at all. Obviously, this is the reason why normalizing did not influence the performance of the product rules, although it might have significantly changed the estimates themselves.

| Rule | MSE |
|---|---|
| product rule | $8.12 \cdot 10^{110}$ |
| simplified product rule | $7.16 \cdot 10^{-4}$ |
| averaging rule | $5.77 \cdot 10^{-5}$ |
| normalized product rule | $1.34 \cdot 10^{-4}$ |
| normalized simplified product rule | $5.09 \cdot 10^{-5}$ |

Table 8.4: Mean squared difference between $\hat{P}(u)$ and $\frac{1}{|x|}\sum_x \hat{P}(u|x)dx$

Because of this weak connection between the estimation accuracy and the classification error rate we introduced another, more sensitive strategy to assess the accuracy of the posterior estimates. This is based on the simple identity

$$\int_x p(x)P(u|x)dx = P(u). \tag{8.9}$$

With this, having an estimate $\hat{P}(u|x)$, we can examine how precisely the marginal of the distribution $p(x)\hat{P}(u|x) \approx p(u,x)$ over $x$ coincides with the class priors P(u). Of course, in practice $p(x)$ and $P(u)$ are available only in the form of estimates, but since they have much smaller spatial dimensions, we can assume them to be more accurate than $\hat{P}(u|x)$. In our case an estimate of the right-hand side $(\hat{P}(u))$ is obtained as the average occurrence of the different class labels in the data set. An estimate for the left-hand side is calculated by supposing that $p(x)$ is faithfully represented by the distribution of the data items, so we simply average the combination rule outputs over the corpus. The final step is a comparison of the two estimates, both visually and by calculating their mean-squared difference.

This operation was performed for each of the five combination rules. Figures 8.1.a and 8.1.b clearly show that the estimates of the averaging rule are much closer to $\hat{P}(u)$ than those of the simplified product rule that significantly underestimate the posteriors. Division by the priors could offset this, but in practice it results in an overcompensation: estimates obtained with the product rule are frequently bigger than one, and often have such large values that we could not even visualize the averages on a single scale with $\hat{P}(u)$. It is easy to understand how this might occur. Imagine a segment where the neural net very confidently identifies all frames, so the product of its outputs corresponding to the correct class is close to one. If the a priori probability of this class is around 1/50 and the segment consists of 11 frames, then the posterior estimate yielded by the product rule will be around $50^{10}$.

Evidently, normalization considerably alleviates the underestimation-overestimation problem of the product rules, which is clear from comparing Fig. 8.1.b with Fig. 8.1.c. It is also justified by the mean squared differences listed in Table 8.4. Based on the results of both this investigation and the phone classification performance, we may conclude that the normalized simplified product rule is the best estimator. The averaging rule is just slightly worse, while the product rule derived from the independence assumption is worse in both respects.

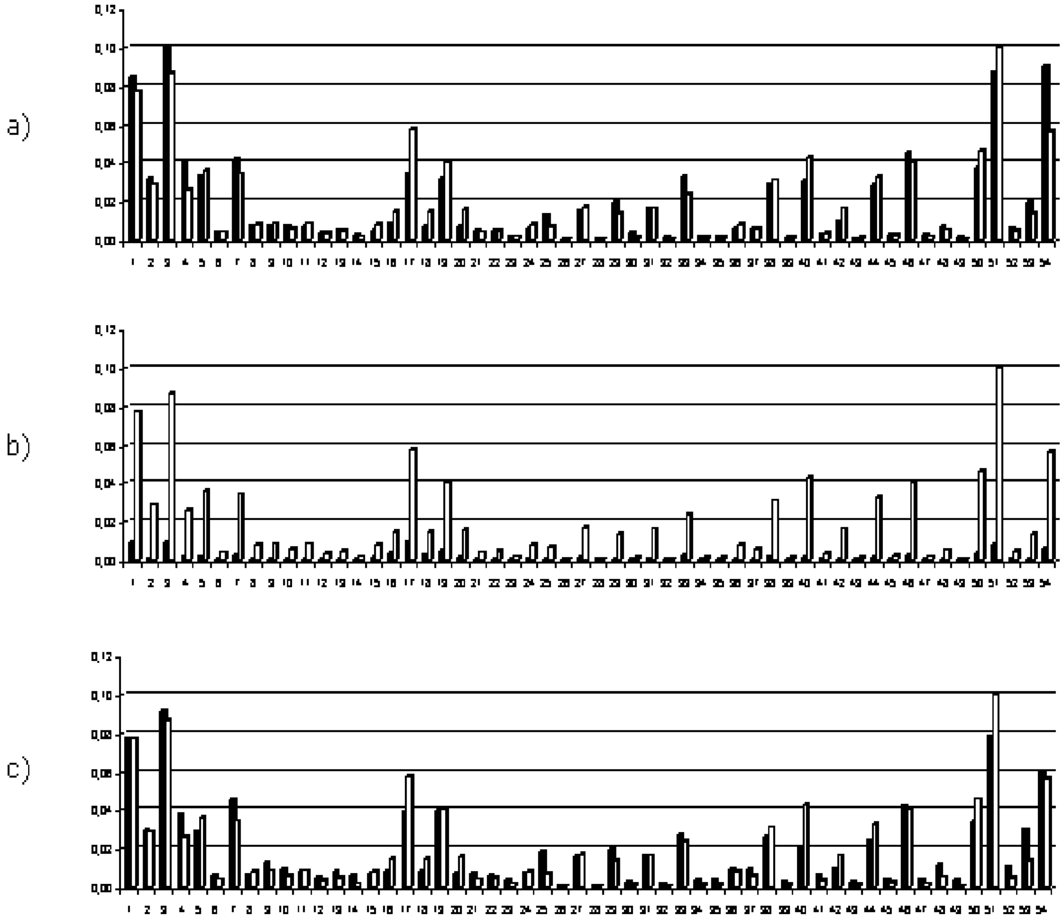

Figure 8.1: The estimates of P(u) obtained from using $\hat{P}(u)$, (white columns) and from marginalization (black columns) based on the estimates of (a) the averaging rule, (b) the simplified product rule and (c) the normalized simplified product rule

## 8.6 Decoding using the Segmental Phone Posteriors

Now let us examine how the different segmental posterior estimates behave when integrated into the recognition process. For this we have to give a decomposition of the utterance-level probabilities and see what other components are required besides the segment-based posteriors. The derivation given here is very similar to those given in Chapter 2 and Chapter 6, but for the sake of completeness we briefly repeat the workings.

First of all, HMM is a generative model, which means that although $P(U|X)$ is required, we model $p(X|U)P(U)$ instead. P(U), the prior probability of a word, is produced by the language model, and the HMM is responsible for $p(X|U)$. This factor is approximated by examining all possible state sequences or, in our terminology, segmentations $S$. That is,

$$p(X|U) = \sum_S p(X,S|U) = \sum_S p(X|S,U)P(S|U) \approx \max_S p(X|S,U)P(S|U). \tag{8.10}$$

Now both $p(X|S,U)$ and $P(S|U)$ are decomposed into phone-level scores. $P(S|U)$ corresponds to the probability of a state sequence in a HMM, but from a segment-based aspect it is better to regard it as a product of exponential phone duration models, $P(S_i|u_i)$ representing the probability that the $i$th unit corresponds to the segment $S_i = (s_{i-1}, s_i)$. With these duration models

$$P(S|U) \approx \prod_{i=1}^{N} P(S_i|u_i), \tag{8.11}$$

and, as mentioned earlier, we employed the same duration model $P(S_i|u_i) = 0.6^{l(i)}$ for each unit.

The other term, $p(X|S,U)$ is written as

$$p(X|S,U) \approx \prod_{i=1}^{N} p(x_{s_{i-1}}^{s_i-1}|u_i). \tag{8.12}$$

Substituting Eq. (8.1) for $p(x_{s_{i-1}}^{s_i-1}|u_i)$ yields a standard 1-state HMM, while taking the five segmental posterior estimates of Section 8.4, dividing them by $P(u_i)$ and substituting them for $p(x_{s_{i-1}}^{s_i-1}|u_i)$ results in five possible HMM/ANN hybrids. The product rule corresponds to the standard HMM/ANN hybrid, the simplified product rule to the standard HMM/ANN hybrid without division by the priors, and the remaining three are new ones.

Let us now compare the five different hybrid models on the city name recognition task. The word error rates are shown in Table 8.5 and clearly indicate the superiority of the product rule. Only the averaging rule could get reasonably close to it, all the other rules performing dismally. It is interesting to note that normalization had a detrimental effect on both product rules, in spite of our earlier finding that it improves their accuracy as segmental posterior estimators. Moreover, the product rule yielding the worse segmental estimates resulted in the best word error rates, while the normalized simplified product rule yielding the best segmental estimates gave the worse word-level results. This severe discrepancy between the phone level and the word level suggests the presence of some conceptual flaw. In the following we try to shed light on it with phone recognition tests.

There is no doubt that restricting the possible recognition results to a dictionary of a moderate size considerably influences the recognition process. To examine more closely how some new acoustic model behaves, it might be instructive to evaluate it without the help of a dictionary, that is on a phone recognition task. Hence we performed such tests, using only a phone-unigram language model assuming that $P(u_1, ..., u_N) = P(u_1) \cdots P(u_N)$. In the case of the hybrid models it was implemented by omitting the division by the prior in Eq. (8.5), while in the conventional Gaussian HMM a

| Combination Rule | WER |
|---|---|
| product rule | 6.97% |
| simplified product rule | 23.44% |
| averaging rule | 8.13% |
| normalized product rule | 32.66% |
| normalized simplified product rule | 49.19% |

Table 8.5: Word error rates of the HMM/ANN hybrid with the different combination rules

| Model | GMM | Prod. | S.Prod. | Avg. | N.Prod | N.S.Prod |
|---|---|---|---|---|---|---|
| PhIP | 7.0 | 4.0 | 0.3 | 3.9 | 2.505 | 2.5002 |
| CORR | 56.29% | 56.82% | 57.46% | 17.04% | 34.90% | 34.22% |
| ACC | 46.38% | 46.44% | 47.71% | 8.63% | 23.34% | 21.64% |
| INS | 9.91% | 10.38% | 9.74% | 8.41% | 11.55% | 12.58% |

Table 8.6: Phone recognition performance of the various models. *GMM:* 1-state GMM with 4 frames minimum duration; *Prod:* hybrid with the product rule; *S.Prod:* hybrid with the simplified product rule; *Avg:* hybrid with the averaging rule; *N.Prod:* hybrid with the normalized product rule; *N.S.Prod:* hybrid with the normalized simplified product rule; *PhIP: the value of the phone insertion penalty*; *CORR:* correctness; *ACC:* accuracy; *INS:* insertion rate

multiplication by $P(u_i)$ was included in the evaluation of a segment.

The phone strings obtained from the recognizer were quantified by their correctness and accuracy, calculated in the usual way from the phone insertion, deletion and substitution errors [125]. Because in certain settings the insertion or deletion errors swamped the result, the introduction of a phone insertion penalty factor [58] was required. Following [79], we empirically tuned the insertion penalty for each case until the number of insertion errors became about 10% of the number of phones. The results with a conventional HMM/GMM and the five HMM/ANN hybrids are summarized in Table 8.6.

Examining the results, we see that the conventional Gaussian model and the hybrids with the product rule and the simplified product rule performed just as well. The other three hybrids were not able to produce a reasonable recognition result. Having found earlier that all models performed very similarly in the phone classification task, this leads us to think that their failure in phone recognition was due to their inability to find the proper segmentation of the signal. It is reinforced by the fact that without a transition penalty (or, in this case, reward) they would have covered any sound file with a single phone. Overall these observations lead us to make the following conjectures:

- Normalization undermines the ability of those models built on the product rules to find the segmentation of a phone sequence. This indicates that the exponential duration models are not able to fulfill their role of forcing the system to find the correct segmentation, but the product rules themselves solve this problem. That is, the fact that their segmental posterior estimates do not sum to one is not a drawback, but is actually a very useful feature.

- The dictionary can be very helpful. With it the averaging model that proved useless in phone recognition performed quite well in word recognition.
- The dictionary can be harmful too. The simplified product rule that was the best in both phone classification and phone recognition was not really good at word recognition, perhaps because it could not work consistently with the dictionary.

## 8.7 The Need for a Segmentation Probability Component

The findings of the previous section seem to be in accord with those of Chapter 7 where we concluded that the naive Bayes rule can automatically handle the outlier or 'anti-phone' segments. The same thing appears to happen here with the product and simplified product rules. Our suspicion is reinforced by the fact that the normalization of the segmental posterior estimates obtained from these rules ruins the results. A possible explanation for this is that normalization destroys the inherent outlier modelling ability of these rules.

In the following we try to explain precisely why and how the normalization step influences the recognition process. To get a better insight of what is going on, we have to revisit the derivations of the previous section. We need to do so because they seem to contradict the decomposition derived in Chapter 6. That is, although in both cases we arrived at formulas built on segmental phone posteriors, in Chapter 6 the segmentation probability factor $P(S|X)$ appeared explicitly, while in equations (8.11) and (8.12) it seems to be missing. Upon re-examination we can see that we made a small mistake in Eq. (8.12) where, during decomposition, we did not explicitly denote $S$ among the conditions of the segment-based phone probabilities. This formula should have been

$$p(X|S,U) \approx \prod_{i=1}^{N} p(x_{s_{i-1}}^{s_i-1}|S_i,u_i), \tag{8.13}$$

where $S_i$ can be interpreted as the event that $x_{s_{i-1}}^{s_i-1}$ is a correct phonetic segment. After applying Bayes' rule on the components we would have arrived at

$$\frac{P(S_i,u_i|x_{s_{i-1}}^{s_i-1})p(x_{s_{i-1}}^{s_i-1})}{P(S_i,u_i)}. \tag{8.14}$$

Ignoring the observation prior probability and decomposing $P(S_i,u_i)$, we would have obtained

$$\frac{P(S_i,u_i|x_{s_{i-1}}^{s_i-1})}{P(S_i|u_i)P(u_i)}. \tag{8.15}$$

Substituting this into Eq. (8.10) we see that, after multiplication by the corresponding factor from Eq. (8.11), the duration probability $P(S_i|u_i)$ cancels out. The other difference is that now we have $P(S_i, u_i|x_{s_{i-1}}^{s_i-1})$ instead of $P(u_i|x_{s_{i-1}}^{s_i-1})$. That is, rather than having a separate duration model, the phone posterior estimator has to handle $S_i$ as well. An intuitive interpretation is that because a randomly chosen segment $x_{s_{i-1}}^{s_i-1}$ does not necessarily coincide with a phone, the model should accommodate not only the various phone classes but also the possibility that the segment does not belong to any of these.

Alternatively, we could decompose $P(S_i, u_i|x_{s_{i-1}}^{s_i-1})$ like so:

$$P(S_i, u_i|x_{s_{i-1}}^{s_i-1}) \approx P(S_i|x_{s_{i-1}}^{s_i-1})P(u_i|x_{s_{i-1}}^{s_i-1}), \tag{8.16}$$

then we can interpret it as follows: the component $P(u_i|x_{s_{i-1}}^{s_i-1})$ can be kept as before, but we have to introduce a further factor, $P(S_i|x_{s_{i-1}}^{s_i-1})$. Its role is to compute the probability that the given segment corresponds to a phone, and so to guide the model towards finding the correct segmentation of the signal.

Let us now examine how we could modify our averaging hybrid so as to conform with Eq. (8.16). To achieve this we have to introduce a further component that represents how much the given segment corresponds to a phone. The duration models used so far can be regarded as possible candidates because the duration information is implicitly present in $x_{s_{i-1}}^{s_i-1}$. But we can do much more, as $x_{s_{i-1}}^{s_i-1}$ is now on the conditioning side, so we can make use of the frame-based posterior estimates as well. A disagreement of the frame-based experts is likely to refer to a phonetically inhomogenious segment, so a reasonable idea is to look for a formula that expresses the coherence of the frame-based scores.

We could create such formulas from scratch, but rather than doing so, let us first examine why the simplified product rule did so well in phone recognition. From the phone recognition tests we suspected the fact that it was not normalized to be somehow connected with this. So let us examine the sum of the estimates produced by the simplified product rule. Knowing that our frame-based estimates are correct in the sense that they are guaranteed to add up to one (a softmax output layer was used), we can write

$$1 = \prod_{j=s_{i-1}}^{s_i-1} 1 = \prod_{j=s_{i-1}}^{s_i-1} \left( \sum_{k=1}^{M} \hat{P}(u_i = q_k|x_j) \right) = \tag{8.17}$$

$$\sum_{k=1}^{M} \left( \prod_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i = q_k|x_j) \right) + \sum_{\substack{1 \le k_{s_{i-1}}, \cdots, k_{s_i-1} \le M \\ \exists p, r \,:\, k_p \ne k_r}} \left( \prod_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i = q_{k_j}|x_j) \right).$$

Here the first term is the sum of the segmental posterior estimates of the simplified product rule, so we can rearrange it like so

$$\sum_{k=1}^{M} \hat{P}(u_i = q_k | x_{s_{i-1}}^{s_i-1}) = 1 - \sum_{\substack{1 \le k_{s_{i-1}}, \cdots, k_{s_i-1} \le M \\ \exists p,r \,:\, k_p \neq k_r}} \left( \prod_{j=s_{i-1}}^{s_i-1} \hat{P}(u_i = q_{k_j} | x_j) \right). \tag{8.18}$$

The term on the right-hand side of Eq. (8.18) contains all products with mixed $q_k$ class targets. The larger the disagreement between the frame-based experts, the larger this term becomes. Consequently, it may be interpreted as a measure of incoherence of the frame-based posteriors. So we see that the posterior estimates obtained from the simplified product rule do not add up to one, but rather to a term that can be viewed as an estimate of $P(S_i|x_{s_{i-1}}^{s_i-1})$. Then we can say that the simplified product rule not only estimates the class posteriors, but also inherently accounts for the outlierness of a segment. This effect is still present when we divide by the priors. The discovery in Section 8.5 that with normalization the estimates become much more accurate also strengthens the notion that we will obtain a better interpretation of the product rules if we view their result as a product of a phone posterior estimate and a segment probability estimate.

In the following we tried to use $\sum_{k=1}^{M} \hat{P}(u_i = c_k|x_{s_{i-1}}^{s_i-1})$ of the simplified product rule as an approximation for the $P(S_i|x_{s_{i-1}}^{s_i-1})$ component of the averaging hybrid. The best performance was found when the exponential duration models were also kept. Table 8.7 presents the phone recognition results with this extended model. As one can see, the introduction of this factor raised the performance of the averaging hybrid so that it achieved results similar to those obtained earlier.

The next step was to try this extended averaging model on the word recognition task. In this case we found that the model tended to prefer longer words, so we had to scale the newly introduced segmentation component (by raising it to a power). Although this is discouraging, since the formulas do not explain why such a scaling factor is necessary, a similar scaling is usually done with the language model of the HMM recognizers [58]. Moreover, we can find examples in the literature where the duration model is scaled in a similar way [91], so we think that scaling the segmentation factor the same way is not a bigger crime. The best word error rates here were obtained with exponents around 0.1, and the smallest error rate was 6.04%, beating all the previous results (cf. Tables 8.1 and 8.5).

As the exponent was optimized on the test dataset, correctness required that we repeat our testing on a different database. For this purpose we selected another block of 438 recordings from the MTBA database, again containing city names, but this time over a smaller vocabulary (see Chapter 4 for details on this test set). All the models that attained a reasonable score on the other test set were evaluated on these recordings, and the results are listed in Table 8.8. The scores show that the segmentation factor significantly improved the performance of the averaging hybrid, actually pushing it into the top position.

| Segm. prob. factor exponent | 1.0 |
|---|---|
| PhIP | 0.5 |
| CORR | 57.01% |
| ACC | 47.92% |
| INS | 9.09% |

Table 8.7: Phone recognition performance of the averaging hybrid with a segmentation probability factor

| Model | WER |
|---|---|
| 3-state HMM/GMM | 4.80% |
| 1-state HMM/GMM, min.dur=4, shared trans. prob. | 5.26% |
| standard HMM/ANN hybrid with division by the priors | 4.57% |
| averaging hybrid without segm.factor | 5.26% |
| averaging hybrid with segm.factor exp=0.1 | 3.20% |

Table 8.8: Word error rates of the different models on the second test set

## 8.8 Discussion and Conclusions

In this chapter we argued that if the independence assumption does not hold, the product rule is just as ad hoc as any other rule, and we proposed to try the averaging rule instead. Although we focused on proving its viability through recognition results, one can also supply intuitive arguments for it. From an expert combination point of view, the product rule is a better choice if the frames all contain complementary information, and all of this is required for a correct classification. Averaging is preferable if the frames all carry similar information, so their combination only reinforces the estimate, but brings no new knowledge. Intuitively we can say that in the case of speech frames the truth is somewhere in between, so neither of the two rules is optimal. It is also interesting to note that the error rates for phone classification seen in Section 8.5 are just slightly better (around 42%) than those for the frame classification given in Section 8.3 (46.47%). This is rather disappointing, and it also indicates that either the frames do not contain complementary information or that the combination rules fail to properly integrate their information content. The latter is supported by the fact that with a rather simple segmental representation we were able to obtain a phone classification error of 32.17% on the same database in Chapter 6. Both the intuitive argument and this result indicates that it is worth searching for better combination rules, either by selecting them empirically or by taking a family of parametric rules and optimizing their parameters algorithmically. Actually, experimenting with different posterior combination rules has now a long tradition when the posteriors are provided by different preprocessors or frequency bands (these are known as the multi-stream and multi-band approaches) [45]. But they are rarely applied to frames, probably because the usual decoding process has to be slightly modified. A notable exception is the 'extended union model' of Ming et al. [85] and Chan et al. [19], which is reported to be robust against short-time temporal noise corruptions. Although we did not test it, one expects the averaging rule to be more resistant to impulse-like noises than the product rule as averaging dampens the local classification errors, while multiplication amplifies them [108]. Hence robustness suggests a preference for an OR-like combination rule to an AND-like one, irrespective of whether we work with frames or frequency bands.

Our other main conclusion of the chapter was that, similar to the posterior-based segmental models of Chapter 6, the segmentation probability factor is present in the conventional HMM/ANN hybrid too, although in a somewhat hidden form. We argued that the simplified product rule implicitly contains this factor in the sum of its estimates. We proved it by removing this factor by normalization, after which the simplified product rule gave a more accurate estimate, based on its marginal distributions, but was no longer able to perform phone recognition. Moreover, when we incorporated this factor into the averaging model, its phone recognition performance became comparable to that of the simplified product rule. The formula that was found to approximate the segment probability value can be interpreted as a measure of the incoherence of the frame-based results, but is not optimal in any sense. Hence it would be a good idea to look for other formulations, possibly ones that can also be algorithmically optimized.

Comparing the phone recognition and word recognition tests, we can confirm the earlier reports that better phone recognition does not necessarily indicate better word recognition, and vice versa. In particular, we find it disturbing that the simplified product rule – found to be the best segmental estimator and the best model for phone recognition – did not perform well in word recognition. Division by the priors actually makes the segmental estimates worse, so we can say that it does not help by improving the acoustic models (as independence would suggest), but by making the cooperation with the language model more successful. The way of how and when the dictionary helps the recognition is not fully clear to us and deserves further analysis. Another unsatisfactory issue is that, in the case of our newly proposed averaging hybrid, a different scaling factor of the segmentation probability component was optimal for the word and the phone recognition tasks. A possible explanation for this is that with the introduction of an additional expert – the dictionary – a smaller weight is required for the segmentation expert. These two information sources can work quite well on their own, so we have the impression that combining them by multiplication is far from optimal and reasonable, but this issue requires additional study.

Another interesting finding is that the averaging rule and the normalized simplified product rule gave very similar results in the phone classification and the marginal distribution experiments. However, in the word recognition task they behaved quite differently (with a segmentation probability factor and without it). This suggests that the distributions they represent are rather different, but the classification results and the marginals are not able to reflect this difference. This phenomenon definitely deserves further investigation.

# Chapter 9

# Explicit Duration Modelling and Resampling-Based Training in HMM/ANN Hybrids

*"Speech recognition is more a craft than a science."*
*Paul Heisterkamp*

This chapter proposes two relatively small, but interesting refinements over the HMM/ANN hybrids of Chapter 8. One of them is the replacement of the implicit phone duration models (determined by the state transition probabilities) by explicit duration models built on the gamma distribution. The other refinement proposes a minor modification to the training method of the neural net used in the hybrid. Both improvements will bring a modest increase in the recognition performance.

## 9.1 Explicit Duration Modelling in HMM/ANN Hybrids

In some languages like Finnish or Hungarian phone duration is a very important distinctive acoustic cue. The conventional HMM speech recognition framework, however, is known to poorly model the duration information. In this section we compare different duration models within the framework of HMM/ANN hybrids. The tests will be performed with both the conventional hybrid and the averaging hybrid proposed in Chapter 8. Independent of the model configuration, we report that using the usual exponential duration model has no detectable advantage over using no duration model. Similarly, applying the same fixed value for all state transition probabilities, as is usual with HMM/ANN systems, is found to have no influence on the performance. However, the practical trick of imposing a minimum duration on the phones turns out to be very useful. The key part of the section is the introduction of the gamma distribution duration model, which proves superior to the exponential one, yielding a 12-20% relative improvement in the word error rate, thus justifying the use of sophisticated duration models in speech recognition.

### 9.1.1 The Need for Better Duration Modelling

In Hungarian the duration of a phone is very important, as in some cases it may be the only clue in discriminating certain words. Good duration modelling can therefore be an important issue. The conventional HMM speech recognition framework however does not really make use of the duration information. Though the state transition probabilities can be regarded as a geometric duration model, this model is not that effective. First, the geometric distribution is a very poor approximation of real phone durations. Second, several authors have reported that the state transition values have practically no influence on the recognition scores [15]. In this section we examine the issue of duration modelling within the framework of HMM/ANN hybrids. The proposed alternative duration models will be evaluated in combination with both the conventional and the averaging hybrids presented in Chapter 8. In both cases we seek to answer two questions. First, we want to either prove or refute the common view that the geometric duration model is wholly ineffective. Second, we would like to know whether the replacement of the geometric model with a more sophisticated gamma distribution can improve the performance of the two hybrids.

Since the models used here are basically the same as those described in Chapter 8, we will not re-introduce and repeat all the notations and derivations here. We will simply remind the reader that in our models the decomposition into segment-level scores has the form

$$p(X,S|U) \approx \prod_{i=1}^{N} \frac{P(u_i|x_{s_{i-1}}^{s_i-1}) \cdot P(S_i|x_{s_{i-1}}^{s_i-1})}{P(u_i)}. \qquad (9.1)$$

Eq. (9.1) has two main components. The first, $P(u_i|x_{s_{i-1}}^{s_i-1})$, which will be referred to as $P_U$ later on, represents the fact that each phonetic unit $u_i$ has to be identified from $x_{s_{i-1}}^{s_i-1} = (x_{s_{i-1}}, ..., x_{s_i-1})$, the signal segment mapped to it. For the estimation of this segment-based posterior probability we proposed the product rule and the averaging rule in Chapter 8. The other component, $P(S_i|x_{s_{i-1}}^{s_i-1})$ can be interpreted as the probability that $x_{s_{i-1}}^{s_i-1}$ is a correct phonetic segment, and its role is to guide the model towards finding the correct segmentation of the signal. Although in theory the state transition probabilities are responsible for this, in Chapter 8 we argued that they do not fulfill this task. We also argued that the product rule inherently contains a formula (Eq. (8.18)) for this and proposed to use the same formula in the averaging hybrid. In the following we will refer to (8.18)) as $P_S$, and the product of the state transition probabilities will simply be denoted by $P_D$. As in the models both $P_S$ and $P_D$ are present, we can say that altogether the $P_S \cdot P_D$ product estimates $P(S_i|x_{s_{i-1}}^{s_i-1})$ in the hybrids. In the experiments we are going to replace $P_D$ with various sophisticated duration models. The next section is devoted to a detailed discussion of the duration models possible.

### 9.1.2 Duration Models

**No Duration Model.** It has been observed by several researchers and reported in the literature that the values of the state transition probabilities have practically no effect on the recognition result [15]. Thus it is theoretically possible to have no duration model at all. The results obtained this way can serve as a baseline for comparing the effect of the various duration models.

**Exponential (Geometric) Duration Model.** Hidden Markov models incorporate an implicit duration model coded by the self-transition probabilities of the states. If the self-transition probability of a state $q$ is denoted by $a_{qq}$, then the probability that the models stays in state $q$ for $d$ steps (the duration of $d$ frames) is $P_D(d) = (1 - a_{qq})a_{qq}^{d-1}$. This corresponds to a discrete geometric distribution, or an exponential one if we think in term of a continuous distribution. The great advantage of this exponential duration model is that it can be calculated recursively, that is $P_D(d) = P_D(d-1) \cdot a_{qq}$, so it nicely fits the dynamic programming framework of HMMs. However, in practice the duration of phones does not follow an exponential distribution. The example in Fig. 9.1 clearly demonstrates this fact.

The proper values for $a_{qq}$ can be found quite easily. We only need one piece of data for this, namely the average duration for the model to stay in state $q$. In our one-state model the states $q$ directly correspond to phones, so this average duration can be estimated as the mean of the phone durations over a manually segmented speech corpus. From $M_q$, the empirical mean of the data $a_{qq}$ can be estimated by $a_{qq} = (M_q - 1)/M_q$ or $a_{qq} = exp(-1/M_q)$, depending on whether we are using a discrete geometric or a continuous exponential distribution.

**Shared Exponential Duration Model.** While in conventional HMM systems the state transition probabilities are estimated as part of the expectation maximization training procedure, in HMM/ANN systems it is common practice to use the same fixed value for all state transition probabilities [45]. It may be interpreted as if all phones had the same shared duration model. In our experiments the shared parameter value was set to 0.7.

**Exponential Duration Model with Minimum Duration Restriction.** If we compare the data histogram and the exponential curve fit over it in Fig. 9.1, we see that the largest mismatch is with small durations. A relatively simple remedy for this is to impose a minimal duration on the phones during the decoding process. For the duration model this corresponds to zeroing out the first couple of values (see Fig. 9.1). It is also interesting to observe that, in a 3-state model, phones are implicitly constrained to have at least 3 frames (if the skipping of states is forbidden). Restricting the minimal duration to 3 frames in a 1-state model will have a similar effect. Actually, in the experiments we set this value to 4 rather than 3 because this yielded slightly better results.

**Gamma Distribution Duration Model.** Quite evidently, the exponential duration model gives a very poor approximation of the real distribution, even with a minimum duration restriction. It is natural, then, to look for another type of distribution that is only slightly more complicated, but fits the data much better. One possibility is to use

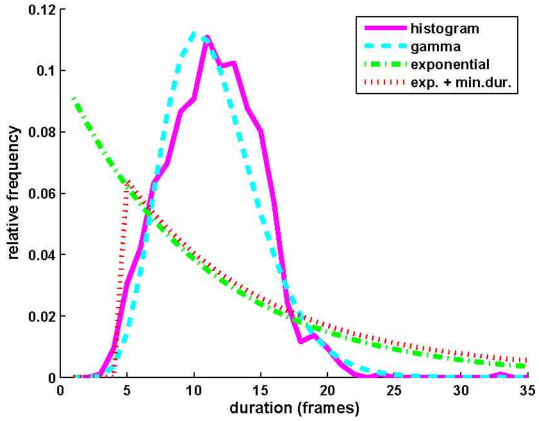

Figure 9.1: Fitting a duration histogram using various probability density functions

the gamma distribution for this purpose. Mathematically it has the form [104]:

$$P_D(d) = \frac{(d/\beta)^{\gamma-1}e^{-d/\beta}}{\beta\Gamma(\gamma)}, \tag{9.2}$$

where $\gamma$ is the shape parameter, $\beta$ is the scale parameter, and $\Gamma$ is the gamma function. The method of moments estimators of the gamma distribution are $\gamma = M_q^2/V_q$ and $\beta = V_q/M_q$, where $M_q$ and $V_q$ are the empirical mean and variation of the data [104].

A purely practical issue is that the gamma function cannot be computed directly but requires numerical approximations. Note, however, that it does not influence the shape of the curve but simply acts as a normalizing constant. Realizing this, we replaced it by a third parameter whose value is estimated by minimizing the mean square error between the histogram of durations and the approximation given by $P_D(d)$.

Fig. 9.1 shows that a gamma distribution indeed fits the data much better than an exponential distribution. The price to be paid for this is that the former cannot be computed recursively, so the usual dynamic programming decoding scheme has to be modified. This brings some additional complexity to the decoding process. Fortunately, this extra burden is manageable, because the other components ($P_U$ and $P_S$) can still be computed recursively, and evaluating $P_D(d)$ for different $d$ values is not cpu demanding. The reader should see [94] and [91] for more on how the conventional HMM or HMM/ANN structure has to be modified to incorporate explicit duration models in them.

### 9.1.3 Experimental Settings

**Database.** All the results presented here were obtained using the MTBA Hungarian Telephone Speech Database. For training we applied 1367 sentences; for the isolated word recognition tests we used a block of 431 city name recordings. For a more detailed description of the database and the train/test data see Chapter 4.

**Preprocessing.** For acoustic preprocessing we applied the Hvite module of the well-known Hidden Markov Model Toolkit (HTK) [125]. We used the most popular preprocessor configuration, that is we extracted 13 MFCC coefficients along with the corresponding $\Delta$ and $\Delta\Delta$ values, thus obtaining the usual 39-element feature vector [125]. For recognition we used our own HMM/ANN decoder implementation, which was earlier found to have a performance similar to that of the standard HTK recognizer.

**Model Configurations.** The 2-layer neural net employed in the system contained 150 sigmoidal hidden neurons and a softmax output layer. Training was performed by conventional backpropagation. The net was trained by making use of the manual segmentation of the database, that is no embedded training was applied here (although a Viterbi-like embedded training scheme is known to be applicable to hybrid models [14]).

In our shorthand notation, the formula evaluated for each segment is

$$\frac{P_U^{\alpha_U} \cdot P_S^{\alpha_S} \cdot P_D^{\alpha_D} \cdot I}{P(u_i)}. \tag{9.3}$$

Note that it is slightly more general than the formula given in Eq. (9.1). The $\alpha$ exponents were introduced based on experience with a similar weighting factor for the language model [58]. $I$ is a phone insertion penalty that can be used to balance the phone insertions and deletions; again, such a factor is known to be useful in language modelling [58].

Two different model configurations were examined in the experiments. In the first model configuration $P_U$ is calculated using the product rule (Eq. (8.6)), $P_S$ is obtained from Eq. (8.18), and the duration model $P_D$ and the value for the insertion penalty $I$ will be varied from experiment to experiment. Both the $\alpha_U$ and $\alpha_S$ exponents will be set to 1. Notice that this configuration is equivalent to the conventional HMM/ANN model – apart from, of course, the duration component that we are going to experiment with.

In the second configuration $P_U$ is calculated using the averaging rule (Eq. (8.8)), $P_S$ is obtained from Eq. (8.18), and the duration model $P_D$ and insertion penalty $I$ will again be varied. $\alpha_U$ will be set to 1, but $\alpha_S$ in this case will be set to 0.1, which was found to be optimal in Chapter 8. We will refer to this configuration as the averaging hybrid model.

### 9.1.4 Results and Discussion

In the first series of experiments we were interested in finding out how the minimum duration restriction and/or sharing a common number base influences the performance of the exponential duration model. In these experiments the $\alpha_D$ exponent and the insertion penalty $I$ were always set to 1. Table 9.1 summarizes the results. From the scores it is quite apparent that the minimum duration constraint significantly improves the recognition performance (not to mention that it also dramatically decreases the run time). As regards the other question, it was surprising to see that both exponential models can be detrimental to the recognition score, and the model using the same fixed value performed better than the phone-specifically tuned one. But this was probably

| Duration Model | Model Configuration | |
|---|---|---|
| | Conventional | Averaging |
| No duration model | 18.10% | 34.11% |
| No dur. model, min.dur=4 | 6.04% | 12.06% |
| Shared exponential | 15.32% | 10.21% |
| Shared exp., min.dur=4 | 6.96% | 5.10% |
| Exponential | 13.00% | 10.21% |
| Exponential, min.dur=4 | 7.20% | 9.28% |

Table 9.1: Word error rates for various exponential model settings

| Duration Model | Conventional Hybrid | | | Averaging Hybrid | | |
|---|---|---|---|---|---|---|
| | $\alpha_D$ | $I$ | WER | $\alpha_D$ | $I$ | WER |
| No duration model | – | 1.511 | **5.80%** | – | 0.254 | **4.87%** |
| Shared exp. dur. mod. | 0.266 | 2.036 | **5.80%** | 0.934 | 0.806 | **4.87%** |
| Exponential dur. mod. | 0.340 | 3.804 | **5.80%** | 0.560 | 1.098 | **4.87%** |
| Gamma duration model | 0.382 | 3.311 | **5.10%** | 0.306 | 0.415 | **3.94%** |

Table 9.2: Word error rates (WER) after fine-tuning $\alpha_D$ and $I$

due to an improper choice of $\alpha_D$ and $I$ (the averaging hybrid turned out to be especially sensitive to these). So the optimization of these parameters was a reasonable next step.

In the second set of experiments the weight factor $\alpha_D$ and insertion penalty $I$ were fine-tuned (with the minimum duration restriction always being turned on). The optimal parameter values were found by a global optimization algorithm called SNOBFIT [59]. The resulting values along with the recognition scores are shown in Table 9.2. The results apparently underpin the belief that the exponential duration model brings no advantage over using no duration model at all (and, according to Table 9.1, with an improperly chosen exponent it can be even detrimental!). Furthermore, the practice of using one shared exponential base value instead of phone-specific ones also proved reasonable, as these models did not differ in performance. These findings seem independent of the model configuration used – conventional or averaging. In both cases only the gamma duration model was better than not applying a duration model at all. It achieved a 12-20% relative improvement in the word error rate, depending on the system configuration.

### 9.1.5 Conclusions

This section investigated the feasibility of applying sophisticated duration models – in our case the gamma distribution – within the framework of HMM/ANN hybrids. In addition, we were also curious to see whether the exponential duration model was indeed ineffective. Two kinds of hybrid model configurations were examined in the tests, the conventional one and the "averaging hybrid". Independent of the configuration used, we found that the exponential duration model had no detectable influence on the recognition performance. Hence the practice of replacing the phone-based self-transition probabilities by a quasi-ad hoc constant is indeed harmless – as this simplified exponential duration model is just as ineffective as the original one. On the contrary, we found that imposing a minimum duration constraint on the phonetic segments not only speeds up the decoding process, but also significantly improves the results. The other thing that yielded an improvement was the gamma duration model. Thus, altogether we are justified in saying that the exponential duration model inherent to the HMM is a really poor one, and that replacing it with just a slightly more complicated model can certainly bring a modest improvement to the error rate.

Finally, let us remark that we did not discuss the differences between the conventional and the averaging hybrids because we were more interested in the duration models. But the scores clearly show the superiority of the averaging hybrid – at least, on this corpus. Moreover, during the experiments we found that the averaging model is much more tunable so, hopefully, with the introduction of new components it can be more easily improved.

## 9.2 Training HMM/ANN Hybrids by Probabilistic Sampling

Throughout this dissertation we assumed that the neural nets applied give a very precise estimate of the phone posteriors – frame-based or segment-based, depending on the actual structure of the system. Reality is not that nice, however. For example, it is known that most machine learning algorithms are sensitive to class imbalances of the training data and tend to behave inaccurately on classes represented by only a few examples. The case of neural nets applied to speech recognition is no exception, but this situation is unusual in the sense that the neural nets here act as posterior probability estimators and not as classifiers. This fact may introduce difficulties, because most remedies designed to handle the class imbalance problem in classification invalidate the proof that justifies the use of neural nets as posterior probability models. In this section we examine one of these, the training scheme called probabilistic sampling, and show that it is fortunately still applicable when training HMM/ANN hybrids. First, we argue that theoretically it makes the net estimate scaled class-conditionals instead of class posteriors, but for the hidden Markov model speech recognition framework this causes no problems, and in fact fits it even better. Second, we will carry out experiments to demonstrate the feasibility of this training scheme. In the experiments we create and examine a transition between the conventional and the class-based sampling, knowing that in practice the conditions of the mathematical proofs are unrealistic. The results show that the optimal performance can indeed be attained somewhere in between, and is slightly better than the scores obtained in the traditional way.

### 9.2.1 The Class Imbalance Problem

Most machine learning algorithms are prone to inferior performance when the training data is imbalanced, that is when the number of training examples accessible from the various classes is significantly different. In such cases it is frequently observed that the classifier is biased towards predicting the more common classes, performing worse on the rarer classes. Although the precise explanation of this behavior may differ from algorithm to algorithm (see [123] for general reasons), in the hope of an improvement it is always possible to alter the effective class frequencies by presenting more examples from the rarer classes to the learning algorithm. These methods come under the general name of "resampling techniques" [123]. (See the material of the workshops [22] and [62] for more details on techniques proposed to handle imbalanced classes.)

The class imbalance problem is also present in speech recognition because the natural distribution of speech sounds (phones) is not uniform. However, the solutions proposed by the machine learning community are not necessarily applicable here. This is because most machine learning papers dealing with the topic focus on classification performance, while in speech recognizers the sub-unit models are used as probability estimators. In particular, the HMM/ANN hybrid recognizers presented in Chapter 8 apply ANNs to estimate the posterior probabilities of the classes. This is made possible by a nice theoretical proof which shows that, under ideal conditions, ANNs estimate the class posteriors [11]. In practice, however, the class imbalance of the training set can lead to inaccurate estimates. A natural idea is to apply the resampling techniques, but these invalidate the proof, so their application is theoretically questionable. In this section we examine one peculiar resampling method, the "probabilistic sampling" training technique recommended by Lawrence et al. [78], and argue that it is still usable in training ANNs for HMM/ANN hybrids. First, in Section 9.2.2 we point out that theoretically it forces the network to estimate scaled class-conditional probabilities instead of class posteriors and this poses no real problem as the recognizer can be easily modified to work with these. Then we show experimentally in Section 9.2.3 that when the recognizer is built on a net trained by probabilistic sampling, it yields the same good or slightly better performance than that with the conventional training.

In the subsequent discussions and experiments we will use the conventional (product rule based) HMM/ANN hybrid as described in Chapter 8. As we explained there, in the hybrid model the ANN is employed to approximate the class posteriors. That is, denoting the local feature vectors by $x$ and the set of class labels by $\mathcal{C} = \{c_1, ..., c_K\}$, we can use them to estimate $P(c_k|x)$. In the hybrid model the HMM states play the role of the classes of the ANN, and the states usually directly correspond to phone classes. The HMM framework requires the class-conditionals $p(x|c_k)$, which can be calculated from the posteriors by Bayes' rule as $p(x|c_k) = P(c_k|x) \cdot p(x)/P(c_k)$. From the HMM optimization point of view $p(x)$ is a constant scaling factor and can be ignored. So the HMM/ANN hybrids work with $P(c_k|x)/P(c_k)$, which thus gives an estimate of $p(x|c_k)$ to within a scaling factor. The $P(c_k|x)$ values are produced by an ANN, and the $P(c_k)$ values are in practice obtained by a simple frequency counting of the class labels over the training corpus.

### 9.2.2 Probabilistic Sampling

Let us now see why and when ANNs estimate the class posteriors, and what happens if training is performed by probabilistic sampling. Let us assume that the network has $K$ outputs denoted by $y_k$ $(k = 1, ..., K)$, and that it is trained by minimizing the sum-of-squares error[1]. We will also assume that the training data is sampled in such a way that its distribution follows the real distribution $p(x)$ of the data points. Under these conditions it can be shown that if the size of the training data is allowed to go to infinity, the error function can be written as

$$E = \frac{1}{2}\sum_k \int [y_k(x) - < t_k|x >]^2 p(x)dx + B, \tag{9.4}$$

where $B$ is a constant that is not important here, and $< t_k|x >$ is the conditional average of the target values $t_k$ at $x$ [11]. Obviously, Eq. (9.4) takes its minimum when $y_k =< t_k|x >$. Now, if the network structure and the labelling of the training data follow the 1-of-K coding scheme (that is $t_k$ takes a value of 1 for the correct class output and 0 for the rest), it is easy to show that $< t_k|x >$ approximates $P(c_k|x)$ (again assuming a representative sampling and an infinite amount of sample data at point $x$).

Examining Eq. (9.4) more closely, we see that at any point $x$ of the input space $\mathcal{X}$ it is $< t_k|x >$, the local ratio of positive and negative examples from class $c_k$, that determines the optimal value for $y_k$. The local errors of these estimates are in turn weighted by $p(x)$, which forces the network to give a closer approximation in those regions of the input space where the density of input data is high, and permits it to give a poorer approximation in regions where the data density is lower. If class labels correlate well with certain regions of the input space $\mathcal{X}$ (which we may assume, otherwise the learning task would be insoluble), then the data density will be lower in those regions where the sparsely represented classes lie. This is the main reason why the network will perform worse on these classes.

This observation leads to the idea of altering the effective class frequencies by presenting more examples from the rarer classes to the learner. In practice, of course, we usually have no way of generating further samples from any class, so resampling is simulated by replicating some of the samples of the rarer classes. An extreme case of this is when the training data set is manipulated so that it contains the same amount of training examples from each class. When training an ANN with the backpropagation algorithm, there is of course no need to really replicate the samples: only the algorithm has to be modified slightly. Usually the training data items are presented to the algorithm in a random order, that is at each iteration a data item is randomly chosen from the full database. We will refer to this method as "full sampling". A possible alternative is to first choose a class at random, and then randomly pick a training sample from the samples belonging to this class. We will call this general, two-step sampling scheme "probabilistic sampling" [78], and the special case when each class is chosen

[1] A similar proof exists for the minimum cross-entropy error criterion as well [11].

with uniform probability "uniform class sampling". In general, however, the choice of the class can follow any distribution, not just a uniform one. For example, if class $c_k$ is chosen with probability $P(c_k)$, that is its own prior probability, then the two-step sampling approach will be practically equal to the traditional one-step full sampling scheme. This will allow us to generate a continuum between full sampling and uniform class sampling by linearly interpolating the probability of class $c_k$ between $P(c_k)$ and $\frac{1}{K}$.

Let us now discuss how the optimum of the error function of Eq (9.4) changes when using uniform class sampling instead of full sampling. We will see that manipulating the class frequencies influences both the global data distribution and the local conditional averages. First let us examine the data distribution, which was originally written as

$$p(x) = \sum_k p(x|c_k)P(c_k). \tag{9.5}$$

The manipulation of the class frequencies can be formalized by weighting the terms as

$$p'(x) = \sum_k p(x|c_k)P(c_k)W_k, \tag{9.6}$$

where $W_k$ are class-dependent weights. From this we can see that modifying the class frequencies changes the focus of the error function, as it modifies $p(x)$. If class labels correlate well with certain regions of the input space, then giving more samples from the sparse classes indeed corresponds to giving more samples from the low data density regions, thus forcing the net to give a better approximation in these areas.

However, the local posterior probabilities are also influenced by this weighting. Clearly, the new $P'(c_k|x)$ values can be written as

$$P'(c_k|x) = \frac{p(x|c_k)P(c_k)W_k}{\sum_j p(x|c_j)P(c_j)W_j}. \tag{9.7}$$

We can think of the denominator as a normalizing factor required to make the local estimates add up to one. In the case of uniform class sampling $W_k$ is inversely proportional to $P(c_k)$ and cancels it out, so overall the $P'(c_k|x)$ values will be proportional to $p(x|c_k)$. These will be the local targets of the network, so we can say that with uniform class sampling the neural network learns the class-conditionals $p(x|c_k)$ within a scaling factor. This causes no problem when integrating the network into the HMM framework, and in fact makes it even simpler: the division by the class priors $P(c_k)$ can be omitted, and the scaling factor will not affect the final maximization process.

### 9.2.3 Experimental Results

All the results presented here were obtained using the MTBA Hungarian Telephone Speech Database. The train/test sets were exactly the same as those used in Section 9.1.3 , that is 1367 sentences were employed for training and 431 city name recordings for testing. For more details on the database please see Chapter 4.

For acoustic preprocessing we applied the Hvite module of the well-known Hidden

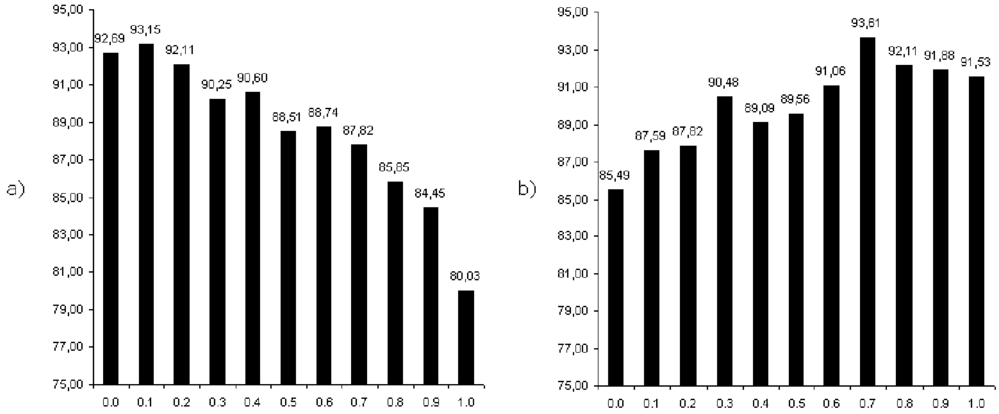

Figure 9.2: Word recognition accuracies (%) as a function of $\lambda$, with and without division by the priors

Markov Model Toolkit (HTK) [125]. We used the most popular preprocessor configuration, that is we extracted 13 MFCC coefficients along with the corresponding $\Delta$ and $\Delta\Delta$ values, thus obtaining the usual 39-element feature vector [125]. For recognition we used our own HMM/ANN decoder implementation, which was earlier found to have a performance similar to that of the standard HTK recognizer.

The neural net used in the system contained 150 sigmoidal hidden neurons and a softmax output layer. Training was performed by conventional backpropagation. Besides comparing the full sampling and uniform class sampling methods, we decided to create a transition between them by making the algorithm select class $c_k$ with a probability $(1-\lambda)P(c_k)+\lambda\frac{1}{K}$, and tested it with various $\lambda$ values between 0 and 1. We did so for purely empirical reasons. It should not be forgotten that the whole investigation here originated from the observation that the mathematical proof regarding the estimation of the posteriors assumes ideal conditions, and that in practice problems with imbalanced classes were reported. Our argument of Section 9.2.2 regarding the estimation of scaled class-conditionals also assumes ideal conditions that do not hold in reality. So while full sampling tends to behave poorly on rarer classes, uniform class sampling may do just the opposite due to over-compensation. This is why it seemed a good idea to create a transition between the two extremes.

As regards division by the class priors, we argued that theoretically it is required when using full sampling and not when using uniform class sampling. However, it is not obvious whether we should use it when the training scheme is somewhere in between. Furthermore, there is evidence that under certain conditions even the conventional model may not require this division [14]. Owing to these uncertainties, we decided to always run the recognizer with the division factor and without it.

The stopping criterion is always a critical issue with every gradient-based algorithm. With our system we have the long-known observation that a certain fixed number of iterations (with a gradually decreased learning rate) produces a nearly optimal solution which cannot be significantly improved either by further iterations or applying subtle stopping criteria. However, because uniform class sampling changes the distribution of the data, we could not be sure that the usual amount of iterations was enough in this case. So in each case we allowed two further series of 10 iterations. The results reported are the averages of the three scores obtained after the three iteration cycles. We should mention here that these never differed significantly, their deviation always being around 1-1.5%, which can be attributed to the random factors present in the whole training process.

Figure 9.2 shows the recognition results for different $\lambda$ values, both with and without division by the priors. Clearly, a $\lambda$ around 0.1 seems optimal when dividing by the priors, and a $\lambda$ of 0.7 yielded the best results when no division by the priors was applied. These are both better than the corresponding results at $\lambda = 0.0$ and $\lambda = 1.0$ which should have performed the best, according to the proofs discussed in Section 9.2.2. This justifies our belief that in practice it is worth using the probabilistic sampling scheme for the training of ANNs of HMM/ANN hybrids as it can bring about a modest improvement over the conventional method ($\lambda = 0.0$, division by the priors).

### 9.2.4 Conclusions

This section investigated the feasibility of the probabilistic sampling training scheme for the training of the ANN components of HMM/ANN hybrid speech recognizers. First we examined uniform class sampling, which is a special case of probabilistic sampling. We argued that although it invalidates the a posteriori probability proof of the conventional training scheme, it is still usable because it gives estimates of the class-conditional probabilities (within a scaling factor) and, in fact, the recognition system requires just these anyway. Second, we suspected that in practice it might be worth interpolating between the conventional full sampling and uniform class sampling, as the mathematical proofs made unrealistic assumptions. In the experiments we indeed found that the optima are somewhere in between – around $\lambda = 0.1$ and $\lambda = 0.7$ respectively, depending on whether we divide by the class priors or not. In both cases our results were slightly better that those obtained by the conventional approach ($\lambda = 0$, division by the priors). This justifies our use of the proposed training scheme in HMM/ANN hybrids.

## 9.3 Summary

In this chapter we proposed two refinements of the HMM/ANN hybrids. One of them was the replacement of the state transition probabilities with gamma-distribution based explicit duration models. The experiments confirmed that these duration models indeed bring a modest improvement – while the conventional exponential ones have no advantage over having no duration model. It is interesting to note that the averaging hybrid phone model with a gamma duration representation has almost nothing in common with an HMM – as it inherits neither the product combination rule nor the state transition probability products specific to the HMM.

The other modification proposed concerned the training scheme of the neural nets embedded in the models. The most important conclusion of this section was that although we have theoretical proofs both for full sampling and uniform class sampling, in practice neither of them is optimal. Rather, the best performance was obtained when we interpolated between the two strategies. This example very nicely reflects what makes the construction of good speech recognizers so difficult: although we have neat mathematical models – good science – behind them, in practice a number of experimental refinements and tunings are possible (and necessary) to obtain really good recognition results. And as this requires a lot of experience, it makes speech recognition more of an art than a science.

# Chapter 10

# Conclusions

Probability theory offers only a very limited range of tools for decomposing a multivariate probability distribution. Yet, by applying these in different ways we can obtain various kinds of speech models. In this dissertation we examined two alternatives of the conventional hidden Markov technology. The decomposition applied in these alternative models differs from the conventional decomposition in two basic ways. First, they are both built on posterior probability estimates instead of the more common generative approach that estimates the class-conditional likelihoods. Second, both models try to replace the independence assumption of the frames (within a segment) with some other decomposition technique. In the case of the segment-based model this was done simply by not decomposing the segments into frames – rather, they were modelled as one unit. The experiments convincingly justified that this technology is much better at classifying phonetic segments than HMMs. However, when it came to the recognition of phone sequences or words, we had to face the fact that the segment-based models have serious difficulties with finding the proper segmentation. We overcame this problem by introducing a sophisticated anti-phone modelling scheme and by applying replicator neural networks. Although with these extensions the model performed practically the same as the HMM, it has, admittedly, lost its attractive simplicity. We are convinced that with further refinements – for example, by combining it with a frame-based model – it could be even better, but it is questionable whether the increased complexity and computation time would make it all worthwhile.

In the other alternative model we returned to the conventional frame-based approach, but now we tried to combine the frame-based posterior estimates by averaging instead of multiplication. Although this may seem nonsensical at first, we found several arguments for it from classifier combination literature. The phone classification experiments and our investigation of the marginals showed that averaging is indeed not worse than multiplication. However, to make it suitable for recognition tasks we had to extend it with a segmentation probability component – a lesson learned from the segmental model approach. To achieve this, we analyzed the product-rule based conventional HMM/ANN hybrid and studied how it handles the segmentation problem. Afterwards we extended the averaging phone model with the segmentation probability component identified in the conventional model and called the resulting framework the 'averaging HMM/ANN hybrid'. In all the experiments we did – phone recognition and isolated word recognition tasks – the averaging model performed the same as or better than the conventional hybrid.

For me the most important result of the dissertation is not the two proposed models, but the insight gained with their help. The segment-based model taught me that phonetic segments can be classified better via a simple and intuitive representation than by HMMs. Although the segment-based approach did not prove superior in decoding tasks, the segment-based view itself provided me with a new insight into what is going on in frame-based recognizers. Both the conventional HMM and the HMM/ANN hybrid were examined from a segment-based point of view, and in both cases it led to the conclusion that it is basically the multiplication-based combination rule that enables these models to hypothesize reasonable segmentations. Having seen that both the segmental model and the averaging rule performed the same or better in phone classification, this means that the principal contribution of the product rule to the decoding process lies in its segmentation ability rather than its classification ability. For me this was definitely a surprising finding and justified the view that sometimes it is worth examining an old problem from an unorthodox angle.

# Appendix A

# Summary in English

The current speech recognition technology is built on statistical principles instead of speech-specific knowledge. Although there are constant attempts to incorporate what is known about human speech perception, these usually refine only the preprocessing step and leave the statistical framework untouched. In particular, the 3-state left-to-right hidden Markov phone modelling (HMM) methodology has been practically unchallenged for the last decade. Rather, development efforts have focused mainly on collecting enormous training corpora and on building sophisticated language models. However, nowadays the technology seems to have reached its limits, its abilities still being far from that of humans. We think that now is a good time to step back and refine the acoustic models as well, retaining the statistical approach but narrowing the gap between the properties of the models and human speech comprehension.

The dissertation began by gathering the main critical remarks on hidden Markov models from a speech perception point of view, and we also discussed some general properties of an envisioned alternative recognition framework. After, we pointed out that the main issue of statistical modelling in speech recognition is that the utterance-level probabilities have to be decomposed into the probabilities of some smaller units. Unfortunately, probability theory offers only a very limited range of tools for decomposing a multivariate probability. Yet, by applying these in different ways we can obtain various kinds of speech models that differ from HMMs in several aspects. A common property of the models we used in the dissertation is that they combine posterior probabilities, while the HMM builds from class-conditional likelihoods. As posterior probability estimators we applied neural networks (ANN) – this approach is known to have a number of advantages over modelling class-conditional likelihoods by Gaussian mixtures, as is usual in HMMs.

The most dubious feature of the hidden Markov model is that its probability decomposition goes down to the level of speech frames – which are then assumed to be conditionally independent, and their likelihood values are combined by multiplication. We proposed two alternative decompositions that avoid this so-called 'naive Bayes' assumption. In one case we simply did not decompose the phonetic segments into frames, but modelled them as one unit. This approach leads to the family of segment-based models, and a significant part of the dissertation dealt with the issues of how to parametrize and train these. As it turned out, they have their advantages, for example, they are much better at classifying phonetic segments. But their special drawback is that they have difficulties in finding the proper segmentation during decoding. Hence we suggested and tested various methods to overcome this problem.

Seeing the special problems of the segment-based framework, in the other model examined we returned to the conventional frame-based approach, but we tried to combine the frame-based posterior probabilities by averaging instead of multiplication. Although this may sound nonsensical at first, we introduced several arguments for it from classifier combination literature. The experiments showed that in classification tasks averaging is indeed no worse than multiplication. However, to make it able to perform phonetic decoding, it had to be extended with a segmentation probability component – a lesson learned from the segmental model approach. We called the resulting framework the 'averaging HMM/ANN hybrid'. In all the experiments we did – phone recognition and isolated word recognition tasks – the averaging model performed the same as or better than the conventional hidden Markov model. In the next chapter we increased its performance even more by extending it with an explicit duration model and a resampling-based training scheme.

An interesting feature of the dissertation is its segment-based view on the decoding task. It came from the experiments with the segment-based model, but we applied it to the frame-based systems as well. Most importantly, it turned our attention to the question of how the frame-based models solve the segmentation problem of the decoding task. Both the conventional HMM and the HMM/ANN hybrid were examined from a segment-based point of view, and in both cases we concluded that it is basically the multiplication-based combination of the frames that enables these models to hypothesize reasonable segmentations. This insight gained into the workings of the frame-based systems was probably a more important result of this dissertation than the two posterior-based models suggested and studied.

All the recognition experiments of this dissertation were carried out on Hungarian speech databases. As in most cases there were no comparative results available, in each case the well-known hidden Markov model toolkit (HTK) was used to obtain a basis for comparison.

## A.1 Summary by Chapters

There are two chapters in the dissertation that do not contain scientific contributions from the author but has the goal of reviewing certain areas. Thus, Chapter 4 gave an overview of the software environment and speech databases used throughout the dissertation, and Chapter 5 gathered all the research results that was judged to be relevant to the topics and experiments described in the dissertation.

The remaining chapters essentially followed the chronological order of the author's research efforts. Chapter 2 gave a detailed description of the critical issues of the current speech recognition technology and presented some of the basic features that would be preferable for a alternative, novel one. It also introduced the mathematical tools used in current models, since the alternative models proposed in the thesis apply the same decomposition tricks, only in slightly different ways.

Chapter 3 presented a generalized algorithmic framework that forms the basis of the implementation of our speech decoder. All the models tested in the dissertation – including the HMM – are special cases of this decoding routine.

Chapter 6 introduced the posterior-based segmental model, our team's first attempt to create a viable alternative to HMM-based phone models. Although it turned out not long after that the segment-based representation could easily outperform HMMs in classifying phonetic segments, it took a lot of effort to bring it up to the level of HMMs in phonetic decoding or word recognition tasks. Hence, a large part of this chapter was concerned with improving the segment-based model by refining its so-called segmentation probability component. Most importantly, two methods were suggested for this. One of them proposed modelling those segments that do not correspond to phones by extending the phone classifier with an additional class, and artificially creating training examples for this 'anti-phone' class. The other solution applied replicator neural networks (RNNs) to handle these outlier segments. Briefly summarizing the results of this chapter, we found altogether, that in phone classification the segment-based method is superior to HMMs, but in decoding tasks it requires involved extensions to give a performance similar to those of HMMs.

By confronting the difficulties with the segmental model, we had to realize the fact that HMMs are in fact rather good – in spite of the quite obvious arguments against them. This revelation led to Chapter 7, in which an in-depth analysis was given on how HMMs perform phonetic decoding and why they can solve the problem of phonetic segmentation and classification while their probability estimates are very inaccurate. In order to understand its behavior, the HMM was compared experimentally with a generative segment-based model; moreover it was analyzed from the unusual segment-based point of view. The findings show that the controversial naive Bayes modelling assumption does not significantly harm the HMM's ability of classifying phonetic segments, and it even helps them solve the problem of phonetic segmentation.

In Chapter 8 the technique of segment-based interpretation was extended to the so-called HMM/ANN hybrid. Namely, we identified which component of the hybrid model corresponds to which component of the segment-based model. This analysis led to the suggestion of an alternative hybrid model in which the frame-based posteriors are combined by averaging instead of multiplication. This 'averaging hybrid' turned out to behave similarly or slightly better than the conventional one on phone classification, phone recognition and word recognition tasks as well.

Chapter 9 proposed two slight refinements over the hybrid model of Chapter 8. One of these was the application of an explicit gamma-distribution based duration model instead of the exponential one inherent to HMMs. The other refinement concerned the training of the neural nets used in the hybrid. Both modifications resulted in a modest improvement in the word error rates.

## A.2 Key Points of the Thesis

In the following a listing of the most important results of the dissertation is given. Table A.1 summarizes which thesis is described in which publication by the author.

I. ) The author developed a segment-based feature set for the representation of phonetic segments. He tested this feature set on several speech corpora and in combination with various machine learning algorithms, and demonstrated that in most cases it results in better phone classification scores than the conventional HMM phone models.

II. ) The author developed various strategies for estimating the segmentation probability component of the posterior-based segmental model, based on the concept of anti-phones. He tested the proposed modelling schemes by comparing their speech recognition performance on several speech databases.

III.) The author investigated the applicability of replicator neural networks for the estimation of the segmentation probability component of segmental models.

IV. ) The author investigated how the modelling bias caused by the naive Bayes assumption influences the performance of HMM phone models. Based on the observations he argued that this bias is such that it does not deteriorate the phone classification performance of the models and it helps them in finding the correct segmentation of the input signal. These arguments together help explain why HMMs are good at phonetic decoding while their probability estimates are quite inaccurate.

V. ) The author examined the behavior of the conventional HMM/ANN hybrid model from a segment-based point of view. Based on the findings of this, he introduced a novel type of HMM/ANN hybrid which combines the frame-based posterior estimates by averaging instead of multiplication. He justified experimentally that the averaging hybrid is capable of a similar or slightly better performance than the conventional hybrid.

VI. ) The author examined the efficiency of using explicit duration models in the HMM/ANN framework. He found that the gamma-distribution based duration model leads to increased recognition performance over the conventional exponential model in both the conventional and the averaging hybrid.

VII.) The author proposed a resampling-based training scheme for the training of the neural nets used in the hybrid models. In experiments the proposed algorithm resulted in modest improvements in recognition accuracy.

| | [77] | [109] | [110] | [111] | [112] | [113] | [114] |
|---|---|---|---|---|---|---|---|
| I. | ● | ● | ● | | | | |
| II. | | ● | ● | | | | |
| III. | | | | ● | | | |
| IV. | | | | | ● | | |
| V. | | | | | | ● | |
| VI. | | | | | | ● | |
| VII. | | | | | | | ● |

Table A.1: The relation between the theses and the corresponding publications

# Appendix B

# Summary in Hungarian

A jelenleg használatos beszédfelismerési technológia matematikai alapelvekre épül és csak minimálisan veszi figyelembe az emberi beszédpercepcióra vonatkozó eredményeket. Történnek ugyan kísérletek ezen ismeretek felhasználására, de ezek a próbálkozások általában csak az előfeldolgozási lépést finomítják, a statisztikai keretrendszert érintetlenül hagyva. A háromállapotú rejtett Markov-modellekre (HMM) épülő, beszédhang-alapú modellezési technológia célszerűségét nagyon kevesen kérdőjelezték meg az elmúlt évtizedben. A fejlesztők az eredmények javítása céljából inkább az adatbázisok méretét növelték és a nyelvi modelleket finomították. Egyre inkább úgy tűnik azonban, hogy a jelenlegi metodológia elérte képessége határait. Talán itt az ideje, hogy az akusztikai modelleken is javítsunk, közelítsük tulajdonságaikat az emberi hallás működéséhez – mindeközben a statisztikai megközelítést is megtartva.

Jelen disszertáció indításaként összegyűjtjük a rejtett Markov-modellek azon jellemzőit, amelyek az emberi beszédpercepció tulajdonságainak ellentmondanak, majd felvázoljuk, hogy ezek helyett mit várnánk el egy fejlettebb megoldástól. Rávilágítunk a beszédjelek statisztikai modellezésének kulcskérdésére, miszerint a mondatokhoz rendelt valószínűségi értékeket valamely kisebb egységek valószínűségéből vagyunk kénytelenek összerakni. A valószínűségszámítás sajnos elég kevés eszközt nyújt egy sokváltozós valószínűségi eloszlás felbontására. Mégis, ezeket a szokványostól eltérő módon alkalmazva különféle, a HMM-től eltérő modelleket vezethetünk le. A disszertációban vizsgált modellek közös jellemvonása, hogy az építőelemek posterior valószínűségét közelítik és azokból építkeznek, míg a HMM az adatok osztályonkénti eloszlását igyekszik leírni. A posterior valószínűségek közelítésére mesterséges neuronhálókat (ANN) alkalmazunk. Ennek különféle előnyei vannak a szokványos – az osztályonkénti eloszlások Gauss-komponensekkel történő – modellezésével szemben.

A rejtett Markov-modelles technológia leginkább vitatható tulajdonsága, hogy a valószínűségek felbontásával egészen a beszédkeretek szintjéig megy le. Ezekről aztán feltételezi, hogy az egyes szegmentumokon belül függetlenek (ez az ún. naív Bayes feltételezés), s ezért a hozzájuk rendelt valószínűségi értékeket összeszorozza. A disszertációban két olyan modellezési megoldást vizsgálunk meg, amelyek elkerülik ezt a felbontási lépést. Az egyik esetben egyáltalán fel sem bontjuk a szegmentumokat keretekre, hanem egy egységként parametrizáljuk őket. Így jutunk el az ún. szegmentum-alapú modellek családjához, és a disszertáció egy jelentős része ezekkel foglalkozik. Mint kiderül, ennek a reprezentációnak számos jó tulajdonsága van, például sokkal jobbnak bizonyul a fonetikai szegmentumok osztályozásában. Nehézségei vannak viszont a felismerés során a helyes szegmentálás megtalálásában, így ez utóbbi probléma megoldására többféle megoldást javasolunk és vetünk össze.

A szegmentum-alapú módszercsalád gyenge pontjaival szembesülve a másik vizsgált modellben visszatérünk a keretalapú feldolgozáshoz. Azonban, váratlan módon, a keretekhez rendelt valószínűségi értékeket szorzás helyett átlagoljuk. Ez első pillanatra ésszerűtlennek tűnhet, de számos érvet hozunk fel a használhatóságára. A beszédhang-osztályozási kísérletek is azt igazolják, hogy az értékek átlagolása nem rosszabb megoldás, mint a szorzás. A teljes felismerési feladat megoldására azonban az átlagolós modell csak akkor képes, ha kiegészítjük egy, a szegmentálás valószínűségét megadó komponenssel - a szegmentum-alapú rendszerrel szerzett tapasztalatokkal összhangban. Az ily módon előálló rendszert 'átlagolós HMM/ANN hibridnek' nevezzük el. Ez az új konstrukció mind a beszédhang-felismerési, mint az izolált szavas felismerési feladatokban valamivel jobb teljesítményt nyújt, mint a hagyományos HMM. Az utolsó fejezetben egy picivel még tovább fokozzuk a hibrid modellek képességeit, amikor is kiegészítjük őket egy explicit hosszmodellel, illetve egy újramintavételezés-alapú tanítási sémával.

A disszertáció egyik különlegessége, hogy az összes előforduló modellt szegmentum-alapú nézőpontból vizsgálja. Ezt a látásmódot eredetileg a szegmentum-alapú modellekkel való kísérletezés tette szükségessé, de a keretalapú rendszerek elemzésében is nagyon hasznosnak bizonyult. Ami a legfontosabb, ráirányította a figyelmünket a szegmentálási részfeladat megoldásának fontosságára. Mind a hagyományos, mind a hibrid HMM modellt megvizsgáltuk ebből a szemszögből, és mindkét esetben arra jutottunk, hogy lényegében a keretek valószínűségének szorzással való kombinálása teszi képessé ezeket a rendszereket a beszédjel fonetikai szegmentumainak megtalálására. Maga ez a szemléletmód és az általa nyert betekintés a keretalapú rendszerek működésébe fontosabb eredménynek tűnik, mint a két javasolt újszerű modell.

A disszertációban előforduló beszédfelismerési teszteket magyar beszédkorpuszokon hajtottuk végre. Mivel néhány kivételtől eltekintve más még nem végzett hasonló vizsgálatokat ezeken az adatbázisokon, ezért összehasonlítás céljából a HTK nevű rendszerrel állítottunk elő rejtett Markov-modelles eredményeket.

## B.1. A fejezetek áttekintése

A dolgozat 4. és 5. fejezete nem a Szerző eredményeivel foglalkozik, hanem áttekintő jellegű. Előbbi a disszertációban használt szoftver-környezetet és beszédadatbázisokat mutatja be, míg az utóbbi összegzi azokat a korábbi tudományos munkákat, amelyeket a Szerző a saját kutatásai előzményeként relevánsnak ítélt.

A disszertáció többi fejezete lényegében időrendben követi a Szerző tudományos vizsgálódásait. A 2. fejezet részletesen elemzi a jelenlegi beszédfelismerési technológia gyenge pontjait és összegzi azokat a fő tulajdonságokat, amelyeket egy reménybeli újszerű megközelítéstől várnánk. Áttekintjük továbbá a jelenlegi technológiában alkalmazott matematikai eszközöket, ugyanis az általunk javasolt alternatív megoldások is ezeket fogják használni, csak némileg más módon.

A 3. fejezet egy általánosított algoritmikai keretet mutat be. Az általunk használt beszédfelismerő-implementáció erre a dekódoló rutinra épül, és a disszertációban előforduló összes felismerési algoritmus – a rejtett Markov-modellt is beleértve – ezen rutin speciális esete.

A 6. fejezet bemutatja a posterior valószínűségekre épülő szegmentális beszédmodellt – beszédkutató csoportunk legkorábbi próbálkozását a HMM-től eltérő beszédmodell kifejlesztésére. Habár a vizsgálatok elég hamar igazolták, hogy a szegmentum-alapú reprezentáció sokkal kézenfekvőbb, és jobb eredményeket nyújt a beszédhangok osztályozásában, mint a HMM, később szembesülnünk kellett a ténnyel, hogy a rendszer megbízhatatlanul viselkedik a fonémafelismerési és izolált szavas felismerési feladatokban. Ennek oka, hogy a modell pontatlanul becsli a jelekhez rendelt ún. szegmentálási valószínűséget. Ennek javítására két fő megoldást javasolunk és tesztelünk a fejezetben. Ezek közül az egyik külön osztályként kezeli azokat a szegmentumokat, amelyek nem felelnek meg semmilyen beszédhangnak, és ezen új osztály tanulásához automatikusan generál 'antifón' tanítópéldákat. A másik megoldás ún. replikátor neuronhálókat alkalmaz ezen szegmentumok kezelésére. Röviden összefoglalva a fejezet eredményeit, azt találjuk, hogy a szegmentum-alapú megközelítés jobb a beszédhangok modellezésében, de a teljes beszéd-dekódolási feladatban csak különféle komplikált kiegészítések és finomítások után tudja utolérni a HMM-et.

A szegmentum-alapú rendszer nehézségeit látva rá kellett döbbennünk, hogy a HMM az egészen nyilvánvaló elvi gyengeségei ellenére is bámulatosan jó megoldás. A 7. fejezetben a szegmentum-alapú szemszögből nézve vizsgáljuk meg a HMM működését, különösképpen azt, hogy miért képes jól megoldani a beszédjelek szegmentálásának és azonosításának feladatát, miközben valószínűségi becslései a naív Bayes feltételezés miatt meglehetősen pontatlanok. Az elméleti okfejtés mellett néhány egyszerű kísérletet is elvégzünk, amelyekben a HMM-et egy generatív szegmentális modellel vetjük össze. Az eredményekből arra következtetünk, hogy a naív Bayes feltevés nem rontja számottevően a HMM beszédhang-azonosítási képességét, a szegmentálásban pedig határozottan segíti.

A 8. fejezetben a szegmentum-alapú látásmódot kiterjesztjük az úgynevezett hibrid HMM/ANN modellekre. Egész pontosan, megvizsgáljuk, hogy a szegmentális modell egyes összetevőinek a hibrid mely komponensei felelnek meg. Következtetéseinket felhasználva egy újszerű hibrid modellt vezetünk be, amelyben a keretekhez tartozó valószínűség-értékeket szorzás helyett átlagoljuk. Ez az újszerű 'átlagolós hibrid' mind a beszédhang-felismerési, mind az izolált szavas felismerési tesztekben kicsit jobbnak bizonyul, mint a hagyományos hibrid.

A 9. fejezet két további finomítást javasol a hibrid modellek javítására. Ezek egyike a gamma-eloszlásokon alapuló explicit hosszmodellek alkalmazása a HMM-ben implicit módon megtalálható exponenciális hosszmodell helyett. A másik finomítás a hibridek részét képező neuronháló tanításának módját változtatja meg kissé. Mindkét módosítás kisebb javulást eredményez a hibrid modell szófelismerési pontosságában.

## B.2. Az eredmények tézisszerű összefoglalása

Az alábbiakban hét tézispontba rendezve összegezzük a Szerző kutatási eredményeit. A kutatásokból származó publikációkat, valamint azok tartalmának az egyes tézispontokhoz való viszonyát a B.1. táblázat tekinti át.

I. ) A Szerző összeállított egy fonetikai szegmentumok parametrizálására alkalmas szegmentális jellemzőkészletet. A jellemzőkészlet reprezentációs erejét többféle adatbázison, többféle tanulóalgoritmussal kombinálva tesztelte. Ezek alapján azt találta, hogy a jellemzőkészlet segítségével általában jobb beszédhang-felismerési eredmények érhetők el, mint a hagyományos HMM beszédhangmodellekkel.

II. ) A Szerző különféle, az ún. 'antifón' koncepción alapuló stratégiákat fejlesztett ki a posterior-alapú szegmentális beszédmodell szegmentálási valószínűségeket leíró komponensének közelítésére. A javasolt modellezési módszerek beszédfelismerési képességeit több magyar beszédkorpuszon is kiértékelte.

III.) A Szerző megvizsgálta a replikátor neuronhálók alkalmazhatóságát a szegmentum-alapú modell szegmentálási valószínűségének közelítésére.

IV. ) A Szerző tanulmányozta a naív Bayes feltevés által okozott modellezési torzítás kihatását a HMM beszédhangmodellek beszédfelismerési képességeire. A vizsgálatok alapján amellett érvelt, hogy a torzítás tulajdonságainál fogva csak kis mértékben ártalmas a HMM beszédhang-azonosítási képességére nézve, miközben számottevően segíti a modellt a beszédjelek fonetikai szegmentumokra tagolásában. Az eredmények megmagyarázzák, hogyan lehet képes a HMM nagyon jó felismerési eredmények elérésére annak ellenére, hogy a valószínűségi becslései meglehetősen pontatlanok.

V. ) A Szerző megvizsgálta a HMM/ANN hibrid modell működését, összevetve azt a szegmentum-alapú modellel. A levont következtetések alapján bevezetett egy újszerű hibrid struktúrát, amely a keretekhez rendelt posterior valószínűségi értékeket szorzás helyett átlagolással kombinálja. Felismerési kísérletekkel igazolta, hogy az 'átlagolós hibrid' hasonló vagy jobb felismerési eredményekre képes, mint a hagyományos hibrid.

VI. ) A Szerző megvizsgálta a gamma-eloszláson alapuló explicit hosszmodellek alkalmazhatóságát hibrid HMM/ANN beszédfelismerőkben. Azt találta, hogy a gamma-eloszlást használó hosszmodell mind a hagyományos, mind az átlagolós hibrid felismerési teljesítményén javít.

VII.) A Szerző egy újramintavételezési sémán alapuló tanítási technológiát javasolt a hibrid modellekben alkalmazott neuronhálók betanításához. A kísérletekben a javasolt módszer a felismerési eredmények kis mértékű javulását eredményezte.

| | [77] | [109] | [110] | [111] | [112] | [113] | [114] |
|---|---|---|---|---|---|---|---|
| I. | ● | ● | ● | | | | |
| II. | | ● | ● | | | | |
| III. | | | | ● | | | |
| IV. | | | | | ● | | |
| V. | | | | | | ● | |
| VI. | | | | | | ● | |
| VII. | | | | | | | ● |

B.1. táblázat. A tézispontok és a Szerző publikációinak viszonya

# Bibliography

[1] Allen, J. B., How do humans process and recognize speech?, IEEE Trans. on Speech and Audio Processing, Vol. 2, No. 4, pp. 567-577, 1994.

[2] Austin, S., Makhoul, J., Schwartz, R., Zavaliagkos, G., Continuous Speech Recognition Using Segmental Neural Nets, Proceedings of the DARPA workshop on Speech and Natural Language, pp. 249-252, February 1991.

[3] Austin, S., Zavaliagkos, G., Makhoul, J., Schwartz, R., Speech Recognition using Segmental Neural Nets, Proceedings of ICASSP'92, Vol. 1, pp. 625-628, 1992.

[4] Bácsi, J., Kerekes, J., Lódiné, Szabó K., Sejtes, Gy., "BeszédMester" – computer-aided teaching reading and speech therapy, Módszertani Közlemények, 2004/2, pp. 61-69., 2004. (in Hungarian)

[5] Bengio, Y., De Mori, R., Flammia, G., Kompe, R., Global Optimization of a Neural Network - Hidden Markov Model Hybrid, IEEE Trans. on Neural Networks, Vol. 3, No. 2, pp. 252-259, March 1992.

[6] Bengio, Y., De Mori, R., Flammia, G., Kompe, R., Neural network - gaussian mixture hybrid for speech recognition or density estimation, In: J.E. Moody, S.J. Hanson, and R.P. Lipmann (eds), Advances in Neural Information Processing Systems 4, pp. 175-182, Morgan Kaufmann, 1992.

[7] Bengio, Y., Neural Networks for Speech and Sequence Recognition, International Thomson Computer Press, London, UK, 1996.

[8] Beyerlein, P., Discriminative Model Combination, Proc. ICASSP'98, pp. 481-484., 1998.

[9] Bilmes, J. A., Natural Statistic Models for Automatic Speech Recognition, Ph.D. Dissertation, University of California, Berkeley, 1999.

[10] Bilmes, J. A., Kirchoff, K., Directed Graphical Models of Classifier Combination: Application to Phone Recognition, Proceedings of ICSLP'2000, pp., 921-924, 2000.

[11] Bishop C. M., Neural Networks for Pattern Recognition, Clarendon Press, 1995.

[12] Bourlard, H., Wellekens, C. J., Links Between Markov Models and Multilayer Perceptrons, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 12, pp. 1167-1178, December 1990.

[13] Bourlard, H., Konig, Y., Morgan, N., REMAP: recursive estimation and maximization of a posteriori probabilities – Application to transition-based connectionist speech recognition, ICSI Technical Report TR-94-064, 1994.

[14] Bourlard, H. A., Morgan, N., Connectionist Speech Recognition – A Hybrid Approach, Kluwer Academic, 1994.

[15] Bourlard, H., Hermansky, H. and Morgan, N., Towards Increasing Speech Recognition Error Rates, Speech Communication, pp. 205-231, May 1996.

[16] Bourlard, H. A., Morgan, N., Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions, In: Giles, C. L. and Gori, M. (eds.), Adaptive Processing, Springer LNAI 1387, pp. 389-417, 1998.

[17] Bourlard, H., Non-Stationary Multi-Channel (Multi-Stream) Processing Towards Robust and Adaptive ASR, Proceedings of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions, pp. 1-10, 1999.

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

[19] Chan, Y.-C., Siu, M., Efficient computation of the frame-based extended union model and its application in speech recognition against partial temporal corruptions, Computer Speech and Language, Vol. 19, pp. 301-319, 2005.

[20] Chang, S., Greenberg, S., Application of Fuzzy-Integration-Based Multiple-Information Aggregation in Automatic Speech Recognition, Proceedings of the IEEE Conf. on Fuzzy Integration Processing, Beijing, 2003.

[21] Chang, S., Wester, M., Greenberg, S., An Elitist Approach to Automatic Articulatory-Acoustic Feature Classification for Phonetic Characterization of Spoken Language, Computer Speech and Language, Vol. 47, pp. 290-311, 2005.

[22] Chawla, N. V., Japkowicz, N. and Kolcz, A. (eds.), Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets, http://www.site.uottawa.ca/ nat/Workshop2003/workshop2003.html, 2003.

[23] Clarkson, P. and Moreno, P. J., On the Use of Support Vector Machines for Phonetic Classification, Proceedings of ICASSP'99, pp. 585-588, 1999.

[24] Digalakis, V. V., Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition, Ph.D. Dissertation, Boston University Graduate School, 1992.

[25] Domingos P. and Pazzani M., On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Machine Learning, Vol. 29, pp. 103-130, 1997.

[26] Duda, R. O., Hart, P. E. and Stork, D. G., Pattern Classification, Wiley and Sons, 2001.

[27] Ellis, D. P. W., Improved recognition by combining different features and different systems, Proceedings of AVIOS'2000, pp. 236-242, 2000.

[28] Evans, E. F., Modelling Characteristics of Onset-I Cells in Guinea Pig Cochlear Nucleus, Proceedings of the NATO ASI on Computational Hearing, pp. 1-6, 1998.

[29] Fegyó, T., Mihajlik, P. and Tatai, P., A Comparative Study on Hungarian Acoustic Model Sets and Training Methods, Proc. Eurospeech 2003, 2003.

[30] Fukada, T., Sagisaka, Y. and Paliwal, K. K., Model Parameter Estimation for Mixture Density Polynomial Segment Models, Proc. of ICASSP'97, pp. 1403-1406, Munich, Germany, 1997.

[31] Fukunaga, K., Statistical Pattern Recognition, New York, Academic Press, 1989.

[32] Futó, I. (ed.), Artificial Intelligence, Aula, 1999. (in Hungarian)

[33] Gales, M. J.F., Young, S. J., The Theory of Segmental Hidden Markov Models, Technical Report CUED/F-INFENG/TR133, Cambridge University Engineering Department, June 1993.

[34] Gish, H. and Ng, K., A Segmental Speech Model With Applications To Word Spotting, Proceedings of ICASSP'93, pp. 447-450, 1993.

[35] Glass, J. R., A Probabilistic Framework for Feature-Based Speech Recognition, Proceedings of ICSLP'96, pp. 2277-2280, 1996.

[36] Glass, J. R., A Probabilistic framework for segment-based speech recognition, Computer Speech and Language, Vol. 17, pp. 137-152, 2003.

[37] Gong, Y. and Haton, J. P., Stochastic Trajectory Modeling For Speech Recognition, Proceedings of ICASSP'94, pp. 57-60, 1994.

[38] Goos, G., Hartmanis, J. and van Leeuwen, J. (eds.), Multiple Classifier Systems, Lecture Notes in Computer Science Vol. 2709, Springer, 2003.

[39] Gosztolya, G. and Kocsor, A., Tóth, L., Various Robust Search Methods in a Hungarian Speech Recognition System, Acta Cybernetica, Vol. 16., pp. 229-240, 2003.

[40] Gosztolya, G. and Kocsor, A., Improving the Multi-stack Decoding Algorithm in a Segment-based Speech Recognizer, In: P. W. H. Chung et al. (eds.), Proceedings of the 16th Int. Conf. on IEA/AIE 2003, LNAI 2718, pp. 744-749, Springer Verlag, 2003.

[41] Gosztolya, G., Kocsor, A., Speeding Up Dynamic Search Methods in Speech Recognition, In: Ali, M., Esposito, F. (eds.), Proceedings of the 18th Int. Conf. on IEA/AIE, LNCS 3533, pp. 98-100, Springer Verlag, 2005.

[42] Gosztolya, G., Kocsor, A., A Hierarchical Evaluation Methodology in Speech Recognition, Acta Cybernetica, Vol. 17, pp. 213-224, 2005.

[43] Greenberg, S., Speaking in shorthand – A syllable-centric perspective for understanding pronunciation variation, Speech Communication, Vol. 29, pp. 159-176, 1999.

[44] Haffner, P., Franzini, M., Waibel, A., Integrating time alignment and neural networks for high performance continuous speech recognition, Proceedings of ICASSP'91, pp. 105-108, 1991.

[45] Hagen, A., Morris, A., Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR, Computer Speech and Language, Vol. 19, pp. 3-30, 2005.

[46] Halberstadt, A. K., Heterogeneous Measurements and Multiple Classifiers for Speech Recognition, Ph.D. Thesis, Dep. Electrical Engineering and Computer Science, MIT, 1998.

[47] Hand D. J. and Yu K., Idiot's Bayes - Not so stupid after all? – Int. Statistical Review, Vol. 69, pp. 385-398, 2001.

[48] Hasegawa-Johnson, M., Baker, J., Borys, S., Chen, K., Coogan, E., Greenberg, S., Juneja A., Kirchhoff, K., Livescu, L., Mohan, S., Muller, J., Sonmez, K., Wang, T., Landmark-Based Speech Recognition: Report of the 2004 Johns Hopkins Summer Workshop, Proceedings of ICASSP 2005., pp. 213-216, 2005.

[49] Hawkins, S., He, H. X., Williams, G. J., Baxter,R. A., Outlier Detection Using Replicator Neural Networks, Proc. DaWak'02, 2002.

[50] Hecht-Nielsen, R., Replicator Neural Networks for Universal Optimal Source Coding, Science, Vol. 269, pp. 1860-1863, 1995.

[51] Hennebert, J., Ris, C., Bourlard, H., Renals, S., Morgan, N., Estimation of Global Posteriors and Forward-Backward Training of Hybrid HMM/ANN Systems, Proceedings of Eurospeech'97, pp. 1951-1954, 1997.

[52] Hermansky, H., Modulation Spectrum In Speech Processing, In: A. Prochazka et al. (eds.), Signal Analysis and Prediction, Birkhauser, pp. 385-398, 1998.

[53] Hermansky, H., Sharma, S., Temporal Patterns (TRAPs) in ASR of Noisy Speech, Proceedings of ICASSP'99, pp. 289-292, March 1999.

[54] Hermansky, H., Ellis, D. P. W., Sharma, S., Tandem connectionist feature extraction for conventional HMM systems, Proceedings of ICASSP'2000, pp. 1635-1638, 2000.

[55] Holmes, W. J. and Russel, M. J., Probabilistic-trajectory Segmental HMMs, Computer Speech and Language, Vol. 13, pp. 3-37, 1999.

[56] Hopcroft, J. E., An *n*log(n)* algorithm for minimizing states in a finite automaton, In: Y. Kohavi and A. Paz (eds.), Theory of Machines and Computations, Academic Press, New York, pp. 189-196, 1971.

[57] Van Horn, K. S., A Maximum-entropy Solution to the Frame-dependency Problem in Speech Recognition, Tech. Rep., Dept. of Computer Science, North Dakota State Univ., Nov. 2001.

[58] Huang, X. D., Acero, A. and Hon, H-W., Spoken language processing, Prentice Hall, 2001.

[59] Huyer, W., Neumaier, A., SNOBFIT - Stable Noisy Optimization by Branch and Fit, Submitted for Publication

[60] Hwang, J. N., Lay, S. R., Maechler, M., Martin, R. D., Schimert, J., Regression modelling in back-propagation and projection pursuit learning, IEEE Trans. on Neural Networks, Vol. 5, No. 3, pp. 342-353, 1994.

[61] Jančovič, P., Ming, J., Hanna, P., Stewart, D., Smith, J., Combining Multi-Band and Frequency-Filtering Techniques for Speech Recognition in Noisy Environments, Proceedings of TSD'2000, pp. 265-270, 2000.

[62] Japkowicz, N. (ed.), Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets, AAAI Tech. Report WS-00-05, 2000.

[63] Jelinek, F., Five speculations (and a divertimento) on the themes of H. Bourlard, H. Hermansky, and N. Morgan, Speech Communication, Vol. 18, pp. 242-246, 1996.

[64] Jelinek, F., Statistical Methods for Speech Recognition, MIT Press, 1997.

[65] Johansen, F. T., A comparison of hybrid HMM-architectures using global discriminative training, Proceedings of ICSLP'96, pp. 498-501, 1996.

[66] Juneja, A. and Espy-Wilson C., Significance of Invariant Acoustic Cues in a Probabilistic Framework for Landmark-Based Speech Recognition, Proceedings of From Sound to Sense: Fifty+ Years of Discoveries in Speech Communication, MIT, Cambridge, Jun. 2004.

[67] Juneja, A. and Espy-Wilson, C., An event-based acoustic-phonetic approach to speech segmentation and E-set recognition, Proceedings of International Congress of Phonetic Sciences, Barcelona, 2003.

[68] Kertész-Farkas, A., Fülöp, Z. and Kocsor, A., Compressed Storage of Hungarian vocabularies using Nondeterministic Automaton, Proc. MSZNY, pp. 231-236, 2003. (in Hungarian)

[69] Kimball, O. A., Segment Modeling Alternatives for Continuous Speech Recognition, Ph.D. Dissertation, Boston University College of Engineering, 1995.

[70] Kingsbury, B. E. D., Perceptually Inspired Signal-processing Strategies for Robust Speech Recognition in Reverberant Environments, Ph.D. Dissertation, University of California, Berkeley, 1998.

[71] Kiraz, G. A., Compressed Storage of Sparse Finite-State Transducers, In: O. Boldt and H. Jürgensen (eds.), Proc. of WIA'99, LNCS Vol. 2214, pp. 109-122, Springer, 2001.

[72] Kirchhoff, K., Bilmes, J. A., Combination and joint training of acoustic classifiers for speech recognition, Proceedings of ISCA ASR 2000, pp. 17-23, 2000.

[73] Kleinschmidt, M., Localized Spectro-Temporal Features for Automatic Speech Recognition, Proceedings of EuroSpeech 2003, pp. 2573-2576, 2003.

[74] Kocsor, A., Tóth, L., Kuba Jr., A., Kovács, K., Jelasity, M., Gyimóthy, T., Csirik, J., A Comparative Study of Several Feature Space Transformation and Learning Methods for Phoneme Classification, International Journal of Speech Technology, Vol. 3, Number 3/4, pp. 263-276, 2000.

[75] Kocsor, A., Tóth, L., Felföldi, L., Application of Feature Transformation and Learning Methods to Phoneme Classification, In Monostori et al. (eds.), Proceedings of IEA/AIE 2001, LNAI 2070, pp. 502-512, Springer, 2001.

[76] Kocsor, A. and Tóth, L., Application of Kernel-Based Feature Space Transformation and Learning Methods to Phoneme Classification, Applied Intelligence, Vol. 21, No. 2, pp. 129-142, 2004.

[77] Kocsor, A. and Tóth, L., Kernel-Based Feature Extraction with a Speech Technology Application, IEEE Transactions on Signal Processing, Vol. 52, No. 8, pp. 2250-2263, 2004.

[78] Lawrence, S., Burns, I., Back, A., Tsoi, A. C., Giles, C. L., Neural Network Classification and Prior Class Probabilities, In: Orr, G. et al. (eds.), Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys, Springer, pp. 299-314, 1998.

[79] Lee, K.-F., Hon, H.-W., Speaker-Independent Phone Recognition Using Hidden Markov Models, IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 37, No. 11, pp. 1641-1648, 1989.

[80] Leung, H. C., Hetherington, I. L., Zue, V. W., Speech Recognition using Stochastic Segment Neural Networks, Proceedings of ICASSP'92, Vol 1, pp. 613-616, 1992.

[81] Makino, S., Kawabata, T., Kido, K., Recognition of consonants based on the Perceptron Model, Proceedings of ICASSP'83, Vol. 2, pp. 738-741, 1983.

[82] Manning, C. D. and Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 2000.

[83] Mariani, J., Gauvain, J. L., Lamel, L., Comments on "Towards increasing speech recognition error rates" by H. Bourlard, H. Hermansky, and N. Morgan, Speech Communication, Vol. 18, pp. 249-252, 1996.

[84] Merz, C. and Murphy, P., UCI repository of machine learning databases, http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[85] Ming, J., Smith, F. J., Union: A model for partial temporal corruption of speech, Computer Speech and Language, Vol. 15, pp. 217-231, 2001.

[86] Mohammed, M., Gader, P., Generalized Hidden Markov Models - Parts I and II, IEEE Trans. on Fuzzy Systems, Vol. 8, No. 1, pp. 67-94, February 2000.

[87] Mohri, M., Pereira, F. and Riley, M., Weighted finite-state transducers in speech recognition, Computer Speech and Language, Vol. 16, pp. 69-88, 2002.

[88] Morgan, N., Bourlard, H., An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition, Signal Processing Magazine, pp. 25-42, May 1995.

[89] Morgan, N., Bourlard, H., Greenberg, S., Hermansky, H., Stochastic Perceptual Auditory-Event-Based Models for Speech Recognition, Proceedings of ICSLP'94, pp. 1943-1946, 1994.

[90] Morris, A., Hagen, A., Glotin, H., Bourlard, H., Multi-stream adaptive evidence combination for noise robust ASR, Speech Communication, Vol. 34, pp. 25-40, 2001.

[91] Morris, A. C., Payne, S., Bourlard, H., Low Cost Duration Modelling for Noise Robust Speech Recognition, Proc. ICSLP' 2002, pp. 1025-1028, 2002.

[92] Okawa, S., Nakajima, T., Shirai, K., A Recombination Strategy for Multi-Band Speech Recognition Based On Mutual Information Criterion, Proceedings of Eurospeech'99, pp. 603-606, 1999.

[93] Ostendorf, M., Digalakis, V., Kimball, O. A., From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 4. pp. 360-378, 1996.

[94] Pylkönnen, J., Kurimo, M., Duration Modeling Techniques for Continuous Speech Recognition, Proceedings of ICSLP' 2004, pp. 385-388, 2004.

[95] Quinlan, J. R., C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, California, 1993.

[96] Richard, M. D. and Lippmann, R. P., Neural network classifiers estimate Bayesian a posteriori probabilities, Neural Computation, 3(4):461:483, 1991.

[97] Rigoll, G., Maximum mutual information neural networks for hybrid connectionist-HMM speech recognition systems, IEEE Trans. on Speech and Audio Processing, Vol. 2, No. 1, pp. 175-184, 1994.

[98] Rish, I., Hellerstein, J. and Thathachar, J., An analysis of data characteristics that affect naive Bayes performance, IBM Technical Report RC1993, 2001.

[99] Robinson, A., Fallside, F., A recurrent error propagation network speech recognition system, Computer Speech and Language, Vol. 5, No. 3, pp. 259-274, 1991.

[100] Saul, L. K., Rahim, M. G., Allen, J. B., A Statistical Model for Robust Integration of Narrowband Cues in Speech, Computer Speech and Language, Vol. 15, No. 2, pp. 175-194, April 2001.

[101] Schlüter, R., Macherey, W., Müller, B. and Ney, H., Comparison of discriminative training criteria and optimization methods for speech recognition, Speech Communication, Vol. 34., pp. 287-310., 2001.

[102] Schölkopf, B., Smola, A. And Müller, K. -R., Nonlinear Component Analysis as a Kernel Eigenvalue Problem, Neural Computation, Vol. 10(5), 1998.

[103] Sejtes, Gy., Kocsor, A., The Database Specification of SpeechMaster, Alkalmazott Nyelvtudomány, Vol. IV, No. 1, pp. 81-89, 2004. (in Hungarian)

[104] NIST/SEMATECH e-Handbook of Statistical Methods
http://www.itl.nist.gov/div898/handbook/

[105] Sharma, S. R., Multi-Stream Approach To Robust Speech Recognition, Ph.D. Dissertation, Oregon Graduate Institute of Science and Technology, 1999.

[106] Szarvas, M., Mihajlik, P., Fegyó, T. and Tatai, P., Automatic Recognition of Hungarian: Theory and Practice, International Journal of Speech Technology, Vol. 3, No. 3/4, pp. 277-287, 2000.

[107] Csaba Szepesvári, personal communication

[108] Tax, D. M. J., van Breukelen, M., Duin, R. P W. and Kittler, J., Combining multiple classifiers by averaging or by multiplying?, Pattern Recognition, Vol. 33, pp. 1475-1485, 2000.

[109] Tóth, L., Kocsor, A., Kovács, K., A Discriminative Segmental Speech Model and its Application to Hungarian Number Recognition, In: Sojka, P. et al. (eds.), Proceedings of Int. Conf. on Text, Speech and Dialogue TSD'2000, Lecture Notes in Artificial Intelligence Vol. 1902, pp. 307-313, Springer, 2000.

[110] Tóth, L., Kocsor, A., Gosztolya, G., Telephone Speech Recognition via the Combination of Knowledge Sources in a Segmental Speech Model, Acta Cybernetica, Vol. 16, pp. 643-657, 2004.

[111] Tóth, L., Gosztolya, G., Replicator Neural Networks for Outlier Modeling in Segmental Speech Recognition, In: Yin, F. et al. (eds.), Proceedings of Int. Symp. on Neural Networks ISNN'2004, Lecture Notes in Computer Science Vol. 3173, pp. 996-1001, Springer, 2004.

[112] Tóth, L., Kocsor, A., Csirik, J., On Naive Bayes in Speech Recognition, International Journal of Applied Mathematics and Computer Science, Vol. 15, No. 2, pp. 287-294, 2005.

[113] Tóth, L., Kocsor, A., Explicit Duration Modelling in HMM/ANN Hybrids, In: Matousek et al. (eds.), Proceedings of Int Conf. on Text, Speech and Dialogue TSD 2005, Lecture Notes in Artificial Intelligence Vol. 3658, pp. 310-317, Springer, 2005.

[114] Tóth, L., Kocsor, A., Training HMM/ANN Hybrid Speech Recognizers by Probabilistic Sampling, In: Duch et al. (eds.), Proceedings of Int. Conf. on Artificial Neural Networks ICANN'2005, Lecture Notes in Computer Science Vol. 3696, pp. 597-603, Springer, 2005.

[115] Trentin, E., Bengio, Y., Furnlanello, C., De Mori, R., Neural Networks for Speech Recognition, In De Mori (ed.), Spoken Dialogues with Computers, Academic Press, pp. 311-361, 1998.

[116] Trentin, E., Gori, M., A survey of hybrid ANN/HMM models for automatic speech recognition, Neurocomputing, Vol. 37, pp. 91-126, 2001.

[117] Vapnik, V. N., Statistical Learning Theory, John Wiley & Sons Inc., 1998.

[118] Szabolcs Velkei, personal communication

[119] Verhasselt, J., Illina, I., Martens, J.-P., Gong, Y., Haton, J.-P., Assessing the importance of the segmentation probability in segment-based speech recognition, Speech Communication, Vol. 24, No. 1, pp. 51-72, 1998.

[120] Vicsi, K, Tóth, L., Kocsor, A., Csirik, J., MTBA – A Hungarian Telephone Speech Database, Híradástechnika, Vol. LVII, No. 8, pp. 35- 43, 2002. (in Hungarian)

[121] Vorstermans, A., Martens, J.-P., Cremelie, N., Speaker-Independent Phone Recognition with a Dynamic Programming/Multi-Layer Perceptron System, Proceedings of ProRISC/IEEE Workshop, pp. 335-340, 1993.

[122] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K., Phoneme Recognition using time-delay neural networks, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 37, pp. 328-339, 1989.

[123] Weiss, G. M., Provost, F., The Effect of Class Distribution on Classifier Learning: An Empirical Study, Tech. Report ML-TR-44, Dep. Comp. Sci., Rutgers Univ, 2002.

[124] Woodland, P.C. and Povey, D., Large Scale Discriminative Training for Speech Recognition, Proc. ISCA ITRW Automatic Speech Recognition: Challenges for the New Millenium, pp. 7-16, 2000.

[125] Young, S. et al., The HMM Toolkit (HTK) – software and manual, http://htk.eng.cam.ac.uk, 2005.

[126] Yun, Y.-S. and Oh, Y.-H., A Segmental-Feature HMM for Speech Pattern Modeling, IEEE Signal Processing Letters, Vol. 7, No. 6, June 2000.

[127] Zavaliagkos, G., Zhao, J., Schwartz, R. and Makhoul, J., A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition, IEEE Trans. Speech and Audio Proc., Vol. 2, No. 1, Part II, January 1994.