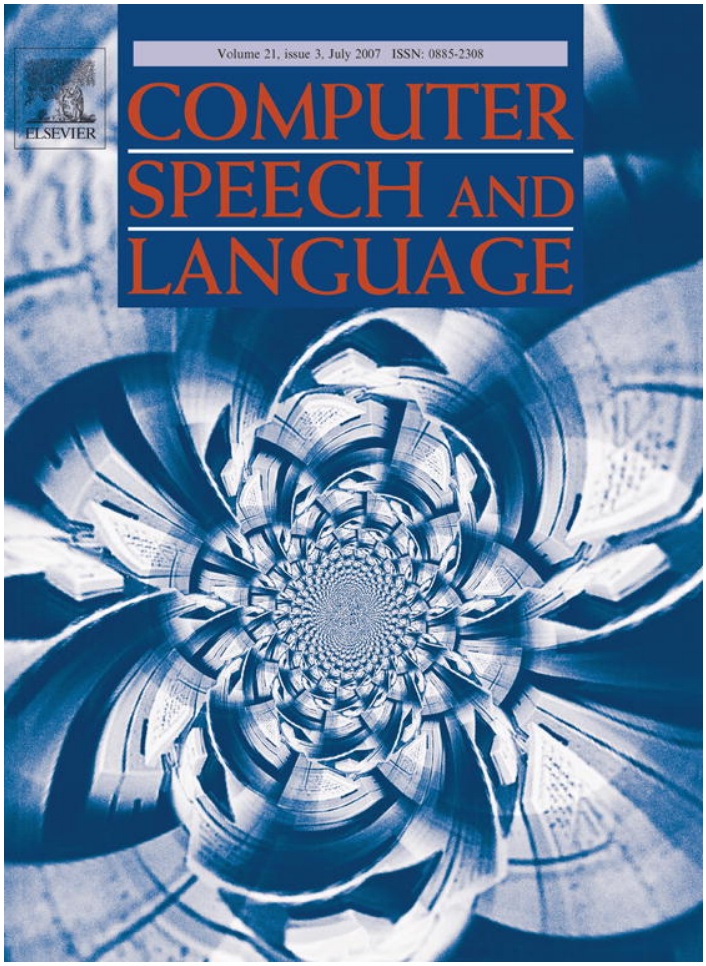

Available online at www.sciencedirect.com
ScienceDirect
Computer Speech and Language 21 (2007) 562–578
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

A segment-based interpretation of HMM/ANN hybrids

László Tóth *, András Kocsor

Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and the University of Szeged, Szeged, Aradi vértanúk tere 1, H-6720 Szeged, Hungary

Received 22 May 2006; received in revised form 5 November 2006; accepted 11 December 2006
Available online 27 December 2006

Abstract

Here we seek to understand the similarities and differences between two speech recognition approaches, namely the HMM/ANN hybrid and the posterior-based segmental model. Both these techniques create local posterior probability estimates and combine these estimates into global posteriors – but they are built on somewhat different concepts and mathematical derivations. The HMM/ANN hybrid combines the local estimates via a formulation that is inherited from the generative HMM concept, while the components of the segment-based model correspond quite directly to the two subtasks of phonetic decoding: segmentation and classification. In this paper we attempt to identify the corresponding components of the segmental model within the hybrid model, with the intent of gaining an insight from this unusual point of view. As regards one of these components, the segment-based phone posteriors, we show that the independence-based product rule combination applied in the hybrid produces strongly biased estimates. As for the other component, the segmentation probability factor, we argue that it is present in the hybrid thanks to the bias of the product rule – that is, the product rule goes wrong in such a special way that it helps the model find the best segmentation of the input. To prove this assertion, we combine this bias with the posterior estimates obtained by averaging, and find that the resulting 'averaging hybrid' slightly outperforms the standard one on a phone recognition task and a word recognition task as well. Overall we conclude that the contribution of the product rule to the decoding process is just as important for the segmentation subtask as it is for the segment classification subtask.

1. Introduction

According to the basic rule of statistical pattern recognition, to achieve optimal classification we need to know the posterior probabilities of the class labels conditioned on the observation vector. In speech recognition the dominant Hidden Markov Modelling (HMM) technology applies a generative, class-conditional description of the likelihoods, from which the posterior estimates are then obtained. There have been several models suggested that seek to describe the global posteriors more directly, in a discriminative manner. The most successful of these is the HMM/ANN hybrid modelling approach. As the name suggests, although these models create posterior estimates (using artificial neural nets (ANNs)) locally, the scheme of how these esti-

* Corresponding author. Tel.: +36 62 544 142; fax: +36 62 425 508.
E-mail addresses: tothl@inf.u-szeged.hu (L. Tóth), kocsor@inf.u-szeged.hu (A. Kocsor).

0885-2308/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2006.12.001

mates get combined into a global score is inherited from the classic HMM. But there exists an alternative approach to the decomposition of the global posteriors: the segment-based decomposition scheme. What makes this approach attractive is that it works directly with phonetic segments instead of hidden states. As the phone insertion/deletion and substitution errors we aim to minimize in the general phonetic decoding task are very closely related to the segmentation and classification errors of the acoustic model, in the segment-based representation the sources of these errors are easier to analyze and understand. In practice the segment-based models turn out to have their own strengths and weaknesses, but in general their typical behavior is quite different from that of the hybrid model. This gave us the idea of performing a component-wise comparison of the two schemes. More precisely, in the paper we first identify the counterparts of the components of the segmental model within the hybrid model, and then analyze these components and their impact on the decoding process. We do so with the hope of gaining a deeper insight into the workings of the hybrid from this unusual angle.

As the database used in the paper is not a standard one, first of all we get some baseline scores via HTK in Section 2. Then we briefly present the conventional HMM/ANN hybrid and the posterior-based segmental model in Sections 3 and 4. In Section 5 we go on to make a pairwise comparison of their corresponding components to discover their advantages and disadvantages.

The segment-based model has two main components, namely the unit probability factor and the segmentation probability factor. In Section 6 we examine what framework the hybrid uses for estimating the first component – that is, the phone posteriors over segments – and we suggest four other frame combination formulae that will be examined during the tests, for the sake of comparison. Next in Section 7 we try to assess how accurate these segmental posterior estimates are. Evaluating the phone classification performance of the phone models for this purpose is an obvious idea, but unfortunately not necessarily a proper indicator of the probability estimation accuracy. Hence we are going to propose an alternative technique that comes from the investigation of the marginal distributions. The findings suggest that the independence-based product rule which the hybrid uses is no better than the ad hoc averaging rule, and that its probability estimates are significantly biased.

In Section 8 we attempt to perform phone recognition with the various frame combination rules – at this point simply ignoring the fact that the model would require a further, segmentation probability component. The results show that the biased phone probability estimators can solve this task, while the more precise, normalized estimators cannot. From this observation we are led to suspect that the former do not require the segmentation probability component because it is inherently present in their bias. To corroborate this, in Section 9 we examine the (simplified) product rule and 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. To prove this assertion even more convincingly, we 'borrow' the bias of the product rule and combine it as a segmentation probability factor with averaging as a unit probability factor. We find that the construct we call the 'averaging hybrid' obtained this way outperforms both the conventional HMM and the standard HMM/ANN hybrid in phone recognition.

For practical applications pure phonetic recognition is not really interesting, as in practice there is always some language model support present. Hence in Section 10 we perform word recognition tests with the various models. We will see that the models that were the best at phonetic decoding perform the worse in word recognition and vice versa. Our averaging hybrid requires a scaling factor for its segmentation probability component to obtain a superior performance. These results imply that with posterior models special care should be taken when combining the acoustic and language (pronunciation) models, which accords with earlier findings (Morgan and Bourlard, 1995).

2. Database and baseline results from HTK

All the results presented in this paper were obtained using the MTBA Hungarian Telephone Speech Database (Vicsi et al., 2002). This is the first Hungarian speech corpus that is publicly available and of a reasonably large size. The most important data block of the corpus contains recorded sentences that were read out loud by 500 speakers. These sentences are relatively long (40–50 phones per sentence), and were selected in such a

way that together all the most frequent phone connections of Hungarian occur in them. Recordings were made via both mobile and line phones, and 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 purpose, but after fusing certain rarely occurring allophones, we worked with only 52 phone classes in the experiments.

For training purposes 1367 sentences were selected from the corpus. For the phone recognition tests we used another set of 687 sentences. The word recognition tests reported here were performed on another block of the database that contains city names. All the 500 city names (each pronounced by a different caller) were different; we constructed a pronunciation dictionary for them by using an automatic phonetic transcription routine (creating one transcript per word, so no alternative pronunciations were considered). From the 500 recordings only 431 were employed in the tests, as the rest contained significant non-stationary noise or were misread by the caller. All words were assumed to have equal priors in the word recognition tests.

To have a reference baseline result on the database, we trained standard Gaussian phone models with the well-known Hidden Markov Model Toolkit (HTK) (Young et al., 1995). For testing we used our own decoder, 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.

First we trained 3-state monophone models using the manual segmentation, that is no embedded training was applied. The best results were obtained with nine Gaussians, and the phone recognition scores are given in Table 1. The word recognition error rate obtained with these phone models was 7.66% on the city name test set. We tried to refine the models with a further embedded training step, but this brought only a slight improvement, resulting in an error rate of 6.73%.

In HMM/ANN hybrid systems it is quite usual to have only one state per phone (Hagen and Morris, 2005). In the following sections we are also going to assume 1-state phone models in the hybrid, because this will be required for the componentwise comparison with the segmental model. Hence, for curiosity we performed some experiments with 1-state phone models in the conventional GMM-based system as well. In this case for reasonable results we had to slightly modify our decoder algorithm so that it imposed a minimum duration restriction on the states (in a standard 3-state model a minimum phone duration of three frames is inherently present). The best scores were obtained with a minimum duration of four frames, so scores with this constraint are given in Tables 1 and 2. All the subsequent results obtained by 1-state HMM or HMM/ANN hybrid phone models should be understood with this four frames minimum duration constraint being active during decoding.

As is mentioned in the literature, the state transition probabilities have practically no effect on recognition performance (Bourlard et al., 1996). We replaced all self-transition probabilities by 0.6 in the 1-state model, and indeed found that the word error rate did not change. This result indicates that it is not the state transition

Table 1
Phone recognition performance obtained with a conventional HMM/GMM recognizer (HTK)

HMM/GMM	3-state	1-state
Correct (%)	61.60	56.29
Accuracy (%)	52.11	46.38
Insertion rate (%)	9.49	9.91

Table 2
Word error rates of a conventional HMM/GMM recognizer

HMM/GMM setup	WER (%)
3-State, embedded training	6.73
3-State, isolated training	7.66
1-State	8.13
1-State, shared transition probabilities	8.13

probabilities that force the hidden Markov model to find the correct segmentation of an observation sequence, but rather it is the emission probabilities that handle this. We will return to this point later in Section 9.

For the sake of completeness we should mention here that we did not try out any context-dependent models and, as far as we know, 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.

3. The conventional HMM/ANN hybrid

In the past one and half decades some interesting ways of applying ANNs to speech recognition have been proposed. The most successful approach is probably the HMM/ANN hybrid suggested by Morgan et al. (Morgan and Bourlard, 1995). In this construction the neural net is used to obtain posterior probability estimates for the states of the HMM, conditioned on a frame of data or on a series of neighboring frames. Compared to the usual Gaussian mixture based modelling, ANNs offer a more flexible representation with fewer parameters. Also, their standard training algorithms are discriminative by default, while GMMs require special dedicated training methods if one wishes to get maximal discrimination from them. But most importantly, it has been observed that ANNs are capable of capturing the information provided by a longer observation context, whereas GMMs are not. Hence, the ANN of the hybrid is typically trained on a window of 9–11 neighboring frames (Morgan and Bourlard, 1995). The training of the neural net can be performed on a manually segmented data set, but a Viterbi-style iterative embedded training is also possible (Morgan and Bourlard, 1995). Moreover, there exist sophisticated training algorithms for the hybrid that are discriminative not only at the level of frames, but at the utterance level as well (Bourlard et al., 1994). To mention also some drawbacks, as in ANN training all the parameters are tuned simultaneously, so it is much more time and memory consuming than the usual, separate maximum likelihood based training of GMM phone models. Moreover, adaptation (to speaker, noise or domain) currently seems much more difficult with ANNs than with generative techniques.

The popular, but probably conveniently oversimplified interpretation views the HMM/ANN hybrid model as a 'hacked' HMM that simply applies a different technology – neural nets instead of the conventional Gaussian mixtures – for the estimation of the emission likelihoods. As the ANN approximates the state posteriors $P(q|x)$, Bayes' rule has to be applied to convert the net's outputs to the state-conditional likelihoods $p(x|q)$ required by the HMM. This would require calculating $P(q|x)p(x)/P(q)$, but in practice we only divide the ANN-based estimates of $P(q|x)$ by the state priors $P(q)$. Thus we obtain a scaled version of the state-conditional likelihoods – but fortunately, as the scaling factors $p(x)$ will be present in each hypothesis, they will not influence the maximization step of the recognition process.

A different interpretation of the hybrid model is given by Hennebert et al., who argue that the hybrid system trained using local criteria estimates the global posterior probability, given certain well-defined assumptions; they also present a forward–backward training algorithm for optimizing the parameters of the model (Hennebert et al., 1997). In the following we explain the workings of the hybrid model based on their derivation.

Let $X = X_1^T = (x_1, \ldots, x_T)$ denote the observation sequence and $X_{t-c}^{t+c} = (x_{t-c}, \ldots, x_{t+c})$ denote a short observation context of c frames (usually 4 or 5) around frame x_t. $Q = (q_1, \ldots, q_T)$ will stand for a state sequence of T states, and $U = u_1, \ldots, u_N$ for a sequence of phonetic units. Applying the state sequence as a latent variable, we can write the $P(U|X)$ posterior as

$$P(U \mid X) = \sum_Q P(U \mid Q, X) P(Q \mid X). \tag{1}$$

Limiting the dependence of q_t to a local context X_{t-c}^{t+c}, the second factor can be approximated as

$$P(Q \mid X) \approx \prod_t P(q_t \mid X_{t-c}^{t+c}), \tag{2}$$

and $P(q_t \mid X_{t-c}^{t+c})$ will be the component that we estimate using an ANN. In the first factor of Eq. (1) the dependence on X can be dropped, since the hidden state sequence already determines the corresponding unit sequence. Then Bayes' rule can be applied, yielding

$$P(U \mid Q,X) = P(U \mid Q) = \frac{P(Q \mid U)P(U)}{P(Q)}. \tag{3}$$

Here $P(Q)$ is approximated by $\prod_t P(q_t)$, corresponding to the usual division by the state priors in the hybrid. $P(Q|U)$ is modelled as a first-order Markov process, resulting in the product of the Markov model state transition probabilities $\prod_t P(q_t \mid q_{t-1}, U)$. When U is simply constructed by joining 1-state phone models, then this component can be also be interpreted as a product of geometric phone duration models. Finally, the model prior $P(U)$ can be approximated by a phone n-gram. Alternatively, if we have a language model and the goal is to decode not just phones but words, then we can estimate the utterance-level posteriors as

$$P(W \mid X) = \sum_U P(W \mid U,X)P(U \mid X) \approx \sum_U P(W \mid U)P(U \mid X). \tag{4}$$

If the language model information is in the usual form of $P(W)$ priors, then it can be included in the formulation by applying Bayes' rule to $P(W|U)$:

$$P(W \mid U) = \frac{P(U \mid W)P(W)}{P(U)}, \tag{5}$$

where $P(U|W)$ can be modelled with the help of a conventional word pronunciation dictionary and $P(U)$ will cancel out the same factor in Eq. (3).

Putting all the components together and assuming Viterbi decoding (maximizations instead of summations), we can write

$$P(W \mid X) \approx \arg\max_U \arg\max_Q P(Q \mid X)P(Q \mid U)P(U \mid W)P(W)/P(Q). \tag{6}$$

In their early works Morgan et al. argued that under certain conditions the division by the state priors $P(Q)$ was not really necessary, and the recognizer could work by using the posteriors alone (Morgan and Bourlard, 1995). Although the omission of this division in most cases decreases the word recognition performance, they conjectured that this performance drop was due to the fact that, owing to their discriminative nature, hybrid models are very sensitive to discrepancies between the pronunciation dictionary and the real acoustic content of an utterance. They suggested that with the proper design of the pronunciation alternatives of a word this performance gap might be reduced (Morgan and Bourlard, 1995). Though Hennebert et al. said that the derivation of the model given by Eqs. (1)–(5) "gives a clear justification for dividing by the training data priors" (Hennebert et al., 1997), as an aid to insight we will perform tests where this division has been omitted. Furthermore, during our work with the segment-based model (which will be presented in the following section) we also noticed that the pronunciation dictionary can be harmful. Hence, we will first conduct just phone recognition tests (without any language model), and move to the recognition of words only after. There we will return to the problem of pronunciation modelling again.

4. Posterior-based segmental models

The idea of segment-based modelling seems to have arisen around 1989–1993 in several different research groups and under various guises (Austin et al., 1992; Gales and Young, 1993; Leung et al., 1992; Ostendorf et al., 1996). Usually the false conditional independence assumption of HMMs and their limitations in modelling segmental features – especially duration – were mentioned as the main motivation for developing these models (Ostendorf et al., 1996). The generative and the posterior-based segmental models appeared practically in parallel, but there has been much more effort devoted to the generative ones – probably because their connection with HMMs is much more obvious, and because researchers were more familiar with generative modelling techniques. Here, however, we will not discuss the generative segment-based models (see Ostendorf et al., 1996 for a good survey), but instead focus on the posterior-based ones (see, for example, Verhasselt et al., 1998).

In the posterior-based segmental modelling framework the posterior probability of a sequence of phonetic units $U = u_1, \ldots, u_N$ is decomposed as

$$P(U \mid X) = \sum_S P(U \mid S,X)P(S \mid X), \tag{7}$$

where the summation is over all possible phonetic segmentations S. In the following, segmentations will be specified by $N+1$ segment boundary frame indices $S = s_0, \ldots, s_N$, so that the first and the last frames belonging to the ith segment are indexed by s_{i-1} and $s_i - 1$, respectively.

The first component of this decomposition will have the task of identifying ('labelling') the segments of segmentation S. Assuming that the information required for identifying the ith unit is basically contained in the underlying signal segment $X_{s_{i-1}}^{s_i-1}$ (although adding the observation context and/or the previous unit among the conditions would be possible), we can write

$$P(U \mid S,X) \approx \prod_{i=1}^{N} P(u_i \mid X_{s_{i-1}}^{s_i-1}). \tag{8}$$

Here we will refer to $P(u_i \mid X_{s_{i-1}}^{s_i-1})$ as the *unit classification probability*. If we intend to model this factor by means of neural nets, we have to tackle the technical problem that the segments are of different durations. As a neural net classifier works with a fixed number of input variables, each segment has to be represented by the same feature set. Fortunately, it is quite straightforward to find the kind of a segmental feature set that yields a segment classification rate similar to or better than those of HMM phone models. Such examples can be found in a paper by Glass (1996) and Clarkson and Moreno (1999).

The other factor of the decomposition, $P(S|X)$ will be called the *segmentation probability*, whose role can be explained as follows. During recognition the decoder evaluates all possible segmentations, so it will inevitably encounter segments that do not correspond to real phones. The unit classifier discussed above is not automatically able to detect and report these segments. First, it is not trained on such segments; second, it has to return phone posteriors that add up to one, so it has neither a direct output assigned to the 'outlier' segments nor any indirect way of reporting them. Thus, it is the task of the segmentation probability factor to push the decoder towards finding the most reasonable unit segmentation of the input.

There are several possibilities for modelling $P(S|X)$. Technically the easiest solution is to run a frame-based (e.g. HMM) recognizer, take the N best paths produced by it, and evaluate only these paths using the segmental model (Zavaliagkos et al., 1994). 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 (assumed to have the same value in each case). Alternatively, one may define frame-based scores (for example, to estimate the probability of each frame being a segment boundary position or not), and then combine these values to determine the probability $P(S|X)$ of a whole segmentation (Leung et al., 1992). A third option is to construct the estimate of $P(S|X)$ from segment-based scores $P(S_i \mid X_{s_{i-1}}^{s_i-1})$ using the approximation:

$$P(S \mid X) \approx \prod_{i=1}^{N} P(S_i \mid X_{s_{i-1}}^{s_i-1}), \tag{9}$$

where S_i denotes the event that the data segment selected by the boundary pair (s_{i-1}, s_i) corresponds to a phone. The segment probability $P(S_i \mid X_{s_{i-1}}^{s_i-1})$ can be modelled by using a dedicated two-class neural net or by extending the unit classifier net by a further class associated with non-phonetic – also called 'anti-phone' (Glass, 1996) – data segments. The anti-phone training examples can be generated artificially over a manually segmented corpus (Tóth et al., 2004) or in a corrective manner by providing the net with negative examples encountered during (mis)recognition (Austin et al., 1992).

Including language model information into the segment-based formalism can be done in exactly the same way as that for the hybrid model (that is, via Eqs. (4) and (5)). Putting all the components together and assuming Viterbi decoding, for the segment-based model we obtain:

$$P(W \mid X) \approx \arg\max_{U} \ \arg\max_{S} \ P(U \mid S,X)P(S \mid X)P(U \mid W)P(W)/P(U). \tag{10}$$

5. Examining the components of the two posterior models in parallel

As we saw in Sections 3 and 4, both the HMM/ANN hybrid model and the posterior-based segmental model claim to estimate the global posterior probability, but they are built on different modelling assumptions.

In the following we will discuss their advantages and drawbacks, examining their corresponding components (that is, the various factors in Eqs. (6) and (10)) at the same time where possible.

The differences between the two models stem from the fact that the hybrid scheme applies a hidden state sequence as a latent variable, while in the segment-based decomposition the hypothesized phonetic segmentation plays that role. The representation built on hidden states in the hybrid model is inherited from the original HMM concept, and basically arises from its generative and frame-oriented nature (the need to give an output at every time frame). The components of the segment-based model correspond more directly to the two subtasks of phonetic decoding – that is, finding the phonetic segments and identifying them (in classic terminology, 'segmentation and labelling'). Hence the segment-based approach is probably a little more intuitive.

As regards the language model components $P(U|W)$ and $P(W)$, there is no difference between the two schemes. Although the priors $P(Q)$ and $P(U)$ are slightly different (frame-based vs. segment-based), in modelling there seems to be no apparent advantage of one over the other.

Turning to the neural net components, the net trained to estimate $P(q_t \mid X_{t-c}^{t+c})$ is evaluated at every frame position, while in the other model $P(u_i \mid X_{s_{i-1}}^{s_i-1})$ is trained and evaluated over segments. The problem with the former is that, intuitively, it is not obvious whether it is indeed necessary (or possible) to associate a phone class label to each frame.[1] The problem with the latter is that the segments have to be represented by the same feature set, independent of their duration, and this causes some technical difficulties. However, it is relatively easy to construct a segment-based feature set that yields excellent segment classification results. Moreover, a whole phonetic segment can be efficiently described with only 100–200 features – hence both the frame-based and the segment-based model will use a neural net of about the same size. The frame-based model will have one ANN evaluation per frame, while the segment-based model performs one evaluation per segment. As there are obviously more possible segments than frames (up to some constant factor, if we apply some reasonable segment duration constraint), thus the computational requirement of the segment-based model is clearly higher, and so in segment-based models pruning heuristics are frequently applied to reduce the number of segments that have to be evaluated (Lee and Glass, 1998). A further issue is that a given training database will obviously consist of more frames than phonetic segments, hence the training of the segmental model is more vulnerable with respect to variance and overtraining problems. Thus, as we see, both approaches have their own pros and cons.

To compare the capabilities of the two representations experimentally, we trained a neural network for both tasks. In both cases the neural net applied was a 2-layer MLP with 250 hidden neurons and a softmax output layer. Training was performed by back-propagation, with a cross-validation stopping criterion. The target phone labels were obtained simply by exploiting the manual segmentation of the training database. For the hybrid model the frames were represented by the usual 39 MFCC+Δ+ΔΔ coefficients (extracted with the HCopy module of HTK); the size of the observation context was set to $c = 2$ (as a value of 3 or 4 brought no improvement and only slowed the system down). The segment-based system represented the segments by a feature set of 165 elements. A precise description of the features can be found in Tóth et al. (2004), but a quite similar feature set was used, for example, by Clarkson and Moreno (1999) or Glass (1996).[2] The net trained for the hybrid system had a classification accuracy of 58.52%; The net trained for the segmental system achieved 67.83%. This shows that classifying whole segments is indeed easier – but, of course, the global performance of the segmental model will strongly depend on whether we are able to find the segments.

Clearly, the most dissimilar components of the two models are $P(Q|U)$ and $P(S|X)$. In the hidden Markov model $P(Q|U)$ determines which state sequence the model goes through during the production of $U = u_1, \ldots, u_N$. When using 1-state phone models – as is usual with HMM/ANN hybrids – each u_i phone can be generated by exactly one state, so Q can vary only in the number of steps it remains in the corresponding states. The product of the self-transition probabilities that controls this can be interpreted as a geometric phone duration model. As was explained in Section 4, the role of the $P(S|X)$ factor of the segment-based

[1] In particular, associating "hard" target labels to the frames at the phone transition phases during training may seem questionable – but this problem can be addressed by methods that calculate "soft" targets for each frame (Hennebert et al., 1997)

[2] In all these papers the basic idea is to divide the segment into thirds along the time axis, and calculate the averages of frame-based features such as MFCC coefficients or frequency band energies over these thirds. This set is then extended with derivative-like features extracted at the segment boundaries and duration as an additional feature.

model is to tell us whether the segments selected by S correspond to the phonetic segments of X. Hence both $P(Q|U)$ and $P(S|X)$ are involved in assessing where the phonetic units lie, but they do it in quite different ways: the former is conditioned on the phonetic units and so is implemented as a set of duration models, while the latter is conditioned on the acoustic data and this allows one to perform more sophisticated (machine learning based) calculations. Unfortunately, in practice there are problems with both $P(Q|U)$ and $P(S|X)$, as we shall see below.

As regards duration modelling, it has been pointed out several times in the literature that the geometric distribution provides a very poor approximation of phone durations. Even the classic book of Rabiner and Juang suggested replacing it with an explicit Gamma duration model or recommended one "to assume a uniform duration distribution over an appropriate range of durations" (Rabiner and Juang, 1993). In HMM/ANN hybrids it is common practice to use the same transition probability value for all phones (Hagen and Morris, 2005), and Bourlard et al. reported that the state transition probabilities have practically no effect on recognition performance (Bourlard et al., 1996). In an isolated-word recognition task we found that with a properly tuned phone insertion penalty the exponential duration model yields no advantage over using no duration model at all (Tóth and Kocsor, 2005). These facts indicate that although in the original, generative HMM concept the state transition probabilities determine which states the model traverses, these components play practically no role when the HMM is applied to decoding.

When it comes to estimating the segmentation probability component $P(S|X)$ of the segment-based model, our experience shows that this component is a weak point of the posterior-based segmental models. The importance of this factor on the efficiency of decoding was also emphasized in the article by Verhasselt et al. (1998). Even after a lot of refinements to the estimation of this component, the phonetic decoding performance of our segmental system could just attain that of the conventional HMM/ANN hybrid (for details see Tóth et al., 2004) and was still definitely worse than that for HTK. The results are summarized in Table 3.

Overall, then, we see that the local classification abilities of the segment-based model are much better than those of the hybrid scheme. Yet, when it comes to recognition this advantage is lost, and even with sophisticated estimates for $P(S|X)$ the segmental system can barely match the performance of the hybrid. And the hybrid does so well in decoding (segmentation) despite the fact that its $P(Q|U)$ component is practically ineffective.

The main goal of the paper has been to understand how phonetic segmentation and classification works in the hybrid model – from a point of view of the segment-based model. That is, in the following we seek to identify the counterparts of the unit probability and the segment probability components of the segmental model within the hybrid model. In the next section we start with the easier one, the unit probability component.

6. Obtaining segment-based posteriors estimates from frame-based ones

To see how the hybrid model estimates the segment-based phone posteriors $P(u_i \mid X_{s_{i-1}}^{s_i-1})$, let us examine how formulas Eqs. (1)–(3) work when the model U to be evaluated is simply a model of one phone. We will exploit the fact that we have 1-state phone models, so the set of phonetic units labels $\{c_1,\ldots,c_M\}$ coincides with the set of hidden state labels. Thus there is only one possible state sequence to be traversed during the recognition of unit c_k: the one that remains in state c_k for the whole duration of the segment. Noting this point, the formula for the hybrid model reduces to

$$P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1}) \approx \frac{\prod_{t=s_{i-1}}^{s_i-1} P(q_t = c_k \mid X_{t-c}^{t+c})}{P(q_t = c_k)^{l(i)}} \cdot P(c_k \mid c_k)^{l(i)-1} \cdot P(u_i = c_k), \tag{11}$$

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

Table 3
Phone recognition performance of the various models

Model	HMM	Hybrid	Segmental
CORR (%)	61.60	54.89	59.34
ACC (%)	52.11	45.83	46.56
INS (%)	9.49	9.05	11.71

HMM: 3-state HMM/GMM (HTK); hybrid: conventional HMM/ANN hybrid; segmental: segment-based model.

The phone model we obtained in this way consists of two main components. $P(c_k|c_k)$ denotes the self-transition probability of state c_k which, when raised to the duration of the segment, forms a geometric phone duration model. As we mentioned earlier, this component has a minimal impact on the decoding process, and it could work just as well without it.

What we are going to examine here in more detail is the other, spectral component of Eq. (11). Let us assume that $P(u_i = c_k) \approx P(q_t = c_k)$ (although they do not fully coincide, as the former is the segment, while the latter is the frame level prior of label c_k). With this approximation (and ignoring the duration model) we get

$$P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1}) \approx \frac{\prod_{t=s_{i-1}}^{s_i-1} P(q_t = c_k \mid X_{t-c}^{t+c})}{P(q_t = c_k)^{l(i)-1}}. \tag{12}$$

In classifier combination theory Eq. (12) is known as the *product rule* for obtaining an estimate of the class posteriors from $l(i)$ independent estimates (produced by different classifiers, and/or feature sets) (Tax et al., 2000).

In the subsequent sections we will experiment with some other combination rules as well – not so much with the intent of finding a better one by empirical means, but rather we wish to gain more insight into the recognizer's behavior. In one of these rules we will omit the division by the class priors from the product rule and estimate the segmental posteriors as

$$P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1}) \approx \prod_{t=s_{i-1}}^{s_i-1} P(q_t = c_k \mid X_{t-c}^{t+c}). \tag{13}$$

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

As a third possibility, we will test the *averaging rule* of classifier combination theory:

$$P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1}) \approx \frac{\sum_{t=s_{i-1}}^{s_i-1} P(q_t = c_k \mid X_{t-c}^{t+c})}{l(i)}. \tag{14}$$

It is important to note that while averaging directly guarantees that the segmental estimates of the different phone classes add up to one (when 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 for 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.

7. Assessing the accuracy of the segmental posterior estimates

The goal of this section is to examine how accurate the segment-based posterior estimates obtained via the various combination rules are. The problem with this is that in practice we cannot directly check the accuracy of our estimates. Owing to categorical perception, human subjects are not able to describe the identity of, for example, a phonetic segment in terms of probabilities. As an indirect indicator, we usually examine the classification error rate to assess the accuracy of the posterior estimates. Measuring the phone classification performance of the five posterior models, we obtained the results shown in Table 4. These scores suggest that

Table 4
Phone classification accuracy of the various segmental posterior estimators

Combination rule	Correct (%)
Product rule	60.39
Simplified product rule	63.45
Averaging rule	62.83
Normalized product rule	60.39
Normalized simplified product rule	63.45

averaging and the simplified product rule can in practice be just as good as the product rule – or, in this case, even slightly better.

Unfortunately, good classification does not necessarily mean a good estimate of the probabilities. It is not hard 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 close to the real probabilities at all. Evidently, this is the reason why normalizing did not influence the performance of the product rules, although it might have significantly changed the estimates themselves.

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

$$\int_X P(X)P(u \mid X)\mathrm{d}x = P(u). \tag{15}$$

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

This operation was performed for all the five combination rules. Fig. 1a and b clearly show that the simplified product rule significantly underestimates the posteriors, and that the estimates of the averaging rule are much closer to $\widehat{P}(u)$. The underestimation is also reflected by the fact that the sum of the values yielded by the simplified product rule was 0.12 on average – rather than 1, as would be required from a probability distribution. Division by the priors could offset this underestimation, but in practice it results in an overcompensation: the estimates obtained with the product rule are frequently bigger than one, and the average value of the sum over the classes was $1.07\text{e}10^{59}$. In fact, the estimates of the product rule often took such large values that we could not even visualize them on a single scale with $\widehat{P}(u)$, and that is why the corresponding figure is not shown. 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 that correspond 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 produced by the product rule will be around 50^{10}. We note here that the significant bias of the product rule is well known in both the machine learning and the speech recognition community; we will discuss it in more detail in Section 11.

Evidently, normalization considerably alleviates the underestimation–overestimation problem of the product rules, which is clear from comparing Fig. 1b with Fig. 1c. It is also justified by the mean squared differences listed in Table 5. Based on the results of this investigation and the phone classification performance, we may conclude that the normalized simplified product rule and the averaging rule are the better estimators, while the product rule used in the conventional hybrid is worse according to both measures.

8. Decoding using the segmental phone posteriors only

In this section we try to perform phone recognition using the segmental model, applying the various combination rules for estimating $P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1})$, and simply ignoring the segmentation probability factor $P(S|X)$. We have two reasons for doing this. One is that we want to see what improvement $P(S|X)$ will bring when we add it later on. The other is that the conventional HMM/ANN hybrid seems to have no such factor, and still works fine. Such an experiment could help us understand why this is so.

In the phone recognition experiments no language model was used at all. 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 (Young et al., 1995). Because with certain settings the insertion or deletion errors swamped the result, the introduction of a phone insertion penalty factor (Huang et al., 2001)

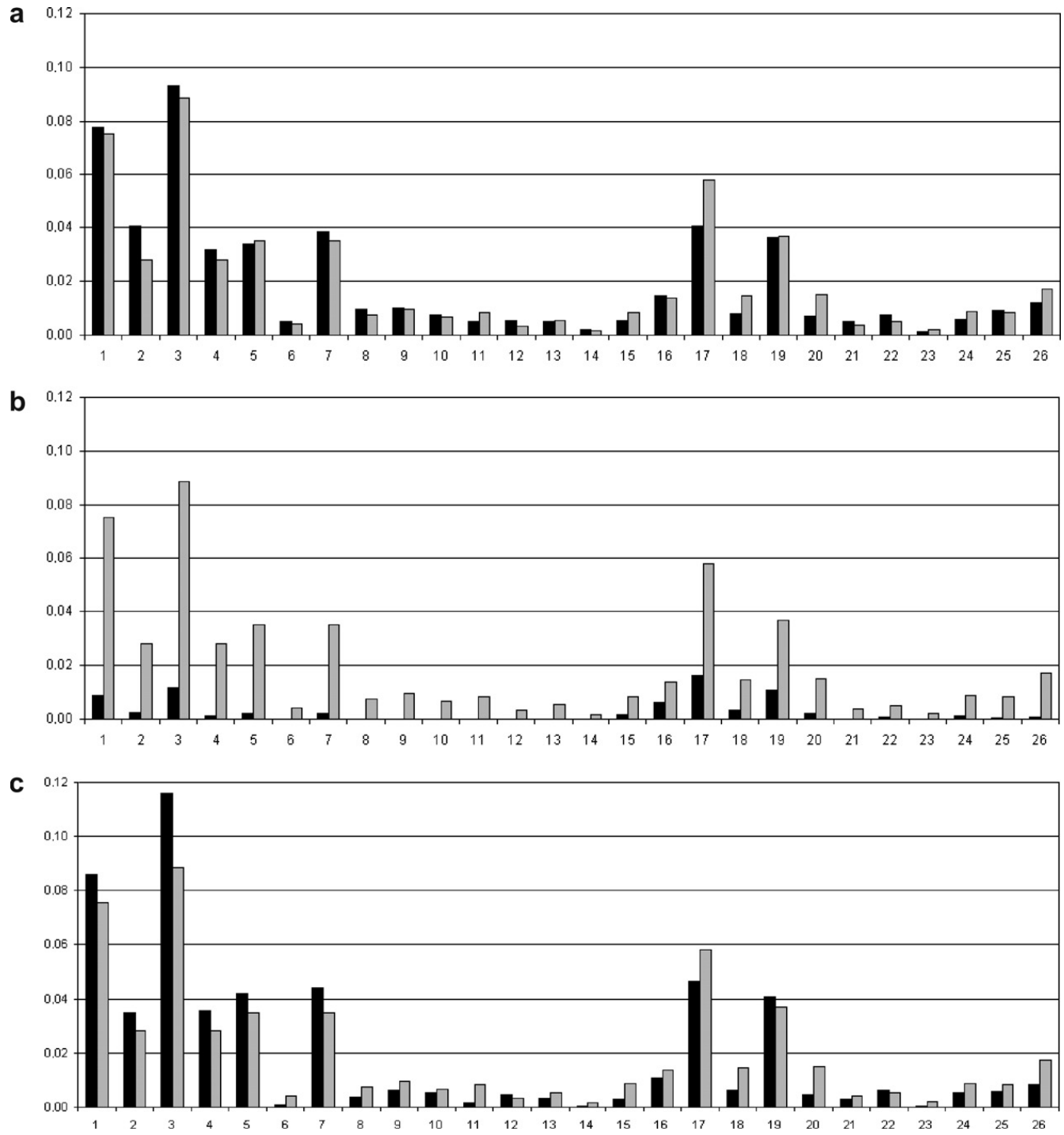

Fig. 1. The estimates for $P(u)$ obtained by using $\widehat{P}(u)$, (grey columns) and by marginalization (black columns) from the estimates of (a) the averaging rule, (b) the simplified product rule and (c) the normalized simplified product rule. (Only half of the columns are displayed for legibility.)

was required. Following Lee and Hon (1989), 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 the five combination rules are summarized in Table 6 (the HTK result is also repeated for comparison).

Examining the results, we see that the product rule and the simplified product rule performed just as well as the traditional HMM system. The other three rules were not able to produce a reasonable recognition result. Having found earlier that all rules performed very similarly in phone classification leads us to think that their failure in phone recognition was due to their inability to find the proper segmentation of the signal. A common

Table 5
Mean squared difference between $\widehat{P}(u)$ and $\frac{1}{|X|}\sum \widehat{P}(u \mid X)$

Rule	MSE
Product rule	1.15e118
Simplified product rule	3.46e−2
Averaging rule	1.40e−3
Normalized product rule	2.00e−3
Normalized simplified product rule	3.13e−3

Table 6
Phone recognition performance of the various models

Model	GMM	Prod.	S.Prod.	Avg.	N.Prod.	N.S.Prod.
CORR (%)	61.60	58.89	61.53	25.57	37.08	36.04
ACC (%)	52.11	48.01	52.25	15.20	26.77	26.69
INS (%)	9.49	10.88	9.27	10.37	10.31	9.35

GMM: conventional 3-state GMM-based HMM; Prod.: segmental model with the product rule; S.Prod.: segmental model with the simplified product rule; Avg.: segmental model with the averaging rule; N.Prod.: segmental model with the normalized product rule; N.S.Prod.: segmental model with the normalized simplified product rule.

property of the 'losers' is that they normalize their estimates to sum to one, while the 'winners' do not. This makes us suspect that the normalization step somehow plays a part in the recognition failure. Hence we are now going to investigate this conjecture in more detail.

9. Tracking down the segmentation probability component

In this section we attempt to explain why and how the normalization of the segmental posterior estimates influences the recognition process. To this end 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

$$1 = \prod_{t=s_{i-1}}^{s_i-1} 1 = \prod_{t=s_{i-1}}^{s_i-1}\left(\sum_{k=1}^{M} P(q_t = c_k \mid X_{t-c}^{t+c})\right) = \sum_{k=1}^{M}\left(\prod_{t=s_{i-1}}^{s_i-1} P(q_t = c_k \mid X_{t-c}^{t+c})\right) + \sum_{\substack{1 \leqslant k_{s_{i-1}},\ldots,k_{s_i-1} \leqslant M \\ \exists p,r : k_p \neq k_r}}\left(\prod_{t=s_{i-1}}^{s_i-1} P(q_{k_t} = c_k \mid X_{t-c}^{t+c})\right). \tag{16}$$

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

$$\sum_{k=1}^{M} P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1}) = 1 - \sum_{\substack{1 \leqslant k_{s_{i-1}},\ldots,k_{s_i-1} \leqslant M \\ \exists p,r : k_p \neq k_r}}\left(\prod_{t=s_{i-1}}^{s_i-1} P(q_{k_t} = c_k \mid X_{t-c}^{t+c})\right). \tag{17}$$

As we see, the posterior estimates obtained from the simplified product rule do not add up to one, as would be required for a probability distribution. Examining the term on the right-hand side of Eq. (17), we notice that it contains all products with mixed c_k class targets. The less coherent and self-confident the frame-based experts are, the larger this term becomes. Such an incoherence of the frame-based posteriors is likely to indicate a phonetically inhomogeneous segment; consequently, the sum on the right-hand side can be viewed as an estimate of $P(S_i \mid X_{s_{i-1}}^{s_i-1})$. 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. In other words, it does not just estimate $P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1})$, but rather the product $P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1})P(S_i \mid X_{s_{i-1}}^{s_i-1})$. The discovery in Section 7 that with normalization the estimates become much more accurate also strengthens the belief that we get

a better interpretation of the product rules if we view their result as a product of a posterior estimate and a segment probability estimate.

To fully convince ourselves that Eq. (17) can indeed be interpreted as an estimate for $P(S_i \mid X_{s_{i-1}}^{s_i-1})$, we combined it with the averaging rule for $P(u_i = c_k \mid X_{s_{i-1}}^{s_i-1})$ in the segment-based decoder. Table 7 presents the phone recognition results with this extended model. As one can see, the introduction of the $P(S_i \mid X_{s_{i-1}}^{s_i-1})$ factor raised the performance of the averaging model so that it achieved results similar to those obtained with HMM. We will refer to the model that uses the averaging rule for unit probability estimation and the sum of the simplified product rule estimates for segment probability estimation as the 'averaging hybrid'.

10. Word recognition tests

The product rule that the HMM/ANN hybrid is built on did not show any superiority in the phone classification and recognition test. In fact, both the simplified product rule and the averaging rule (extended with a segmentation probability factor) worked slightly better in both experiments. Now we will move on to word recognition tests. Comparing Eqs. (7) and (10), we see that this requires two extensions to the decoding process. First, an additional search has to be performed over all possible U transcripts of a word W. As in our case we have only one transcript given for each word, we will simply try to force this transcript on the input signal and $P(U|W)$ will play no real role; furthermore, the $P(W)$ prior will be assumed to be equal for each word and hence can be ignored. Second, a division by the $P(U)$ prior is required. Here we implement it as a simple phone 1-gram, that is, $P(U) \approx P(u_1), \ldots, P(u_N)$. This means one more division per segment, and assuming that $P(u_i = c_k) \approx P(q_t = c_k)$, overall there will be one division by the priors for each frame (as in the product rule of Eq. (12) there is one fewer). Hence, in word recognition the segment-based model built on the product rule will operate in a practically identical way to the conventional HMM/ANN hybrid.

The word error rates obtained with the various models on the city name recognition task are shown in Table 8 (the HTK score is also repeated for comparison). As regards the relation of the product rule and the simplified product rule, our results agree with those stated in the literature, namely that the omission of the division by the priors leads to a decreased word recognition performance (Morgan and Bourlard, 1995). As the next paragraph discusses, a similar phenomenon – that different configurations performed the best in phone and in word recognition – was observed with the averaging hybrid, too. We will return to this problem in Section 11.

The first results with the averaging hybrid (averaging rule with a segmentation probability factor) were very poor, and closer inspection showed that the model had a distinct preference for longer words. Notice that the use of a pronunciation dictionary – that is, forcing given phonetic transcripts on the input – evidently strongly

Table 7
Phone recognition performance of the averaging hybrid when extended with a segmentation probability factor

Averaging hybrid	
CORR (%)	61.34
ACC (%)	52.48
INS (%)	8.86

Table 8
Word error rates of the models with no phone insertion penalty applied

Combination rule	WER (%)
HMM/GMM	6.73
Product rule (standard HMM/ANN hybrid)	7.66
Simplified product rule	11.61
Averaging rule without segmentation probability factor	6.97
Averaging rule with segmentation probability factor	24.37
Averaging rule with segmental factor exp = 0.1	5.81

helps the recognizer in fusing neighboring frames and thus finding reasonable segmentations. This is convincingly supported by the fact that the averaging rule performed very well even without the segmentation factor (remember that in the phonetic decoding task this configuration totally failed). This observation and the model's bias for longer words when using a segmentation probability factor suggested that the influence of the segmentation component should be weakened. We accomplished this by raising $P(S|X)$ to a power. The application of such a scaling factor is quite common with the language model of HMM recognizers (Huang et al., 2001), and we can find examples in the literature where an explicit duration model is scaled in a similar way (Morris et al., 2002). Hence we think that scaling the segmentation factor the same way is similarly acceptable. The best word error rates here were obtained with exponents around 0.1, and the smallest error rate was 5.81%, beating all the other results.

Although in word recognition tasks it is not usual to apply a phone insertion penalty because the dictionary seems to easily solve the phone insertion problems, out of curiosity we examined whether the results could be improved by fine-tuned phone insertion penalty factors. The results in Table 9 indicate that the averaging hybrid could be considerably improved with the help of an insertion penalty factor, and that it can bring a modest improvement also in the case of the two product rules (in the remaining two configurations no significant improvement was observed, so these models were not included in the table).

As the exponent of $P(S|X)$ and the phone insertion penalties were optimized on the test dataset, correctness required that we repeat our testing on another, separate database. For this purpose we selected another block of 438 recordings from the MTBA database, again containing city names, but this time over a somewhat smaller vocabulary. All the models that achieved a reasonable score on the other test set were evaluated on these recordings, and the results are given in Table 10. The scores again show that the segmentation factor significantly improved the performance of the averaging hybrid, actually raising it to a winning position.

11. Discussion and conclusions

In this paper we set out to examine the HMM/ANN hybrid model by comparing it with a posterior-based segmental model. Our motivation for this was twofold. First, in our earlier experiments with the segment-based approach we found that getting a superior segment classification score was relatively easy, but getting a decent decoding performance was much more difficult. Hence we were really keen to understand how the hybrid handles the task of segmentation – especially when we know that the state transition probabilities have almost no effect on the decoding process. Second, we preferred the segment-based formulation because the two subtasks – segmentation and classification – are reflected more directly by the components of the segmental model than they are by the conventional state-based decomposition. We hoped that this unusual segment-oriented point of view could shed light on some internal properties of the hybrid.

Examining the components of the segmental model one at a time, we first looked at how well the hybrid model estimates the phone posteriors. To this end we compared its estimates with those obtained by two other frame combination rules. The segmental classification results did not justify any superiority of the product rule

Table 9
Word error rates of the models with fine-tuned phone insertion penalties

Combination rule	WER (%)
Product rule (standard HMM/ANN hybrid)	6.27
Simplified product rule	10.21
Averaging rule with segmentation probability factor	11.84

Table 10
Word error rates of the different models on the second test set

Model	WER (%)
HMM/GMM	4.80
Standard HMM/ANN hybrid (product rule)	2.52
Averaging hybrid without segmental factor	3.89
Averaging hybrid with segmental factor exp = 0.1	2.06

used in the hybrid, and the study of the marginals revealed that it significantly overestimates the segmental probabilities. This behavior of the product rule (more precisely, the naive Bayes rule it is based on) is well known in machine learning research, and considerable effort has been devoted to its analysis. Most pertinently, it has been pointed out that in many cases naive Bayes provides optimal classification even though it incorrectly estimates the probabilities (Domingos and Pazzani, 1997). It is reasonable to expect that the same phenomenon arises in speech recognizers as well. With conventional HMMs it is widely known that "the acoustic probability is usually underestimated, owing to the fallacy of the Markov and independence assumptions" (Huang et al., 2001). Yet, in practice this bias of the estimates does not necessarily cause a problem, as at the highest level we always measure just the classification error. Quoting Jelinek: "there is no question that HMMs estimate absolute probabilities (densities!) very badly. Yet the relative ratios between two alternative hypotheses may well provide a sufficiently accurate approximation for a choice between them." (Jelinek, 1996). Still, in the speech community the product rule is usually considered superior and theoretically justified because it can be derived from probability theory rules, while averaging is regarded as 'ad hoc'. However the premise of independence that the product rule is based on is just a convenient modelling assumption that does not exactly hold true in reality. As a consequence, the values produced by the product rule will only be estimates that – as we saw – are significantly biased. It is also important to notice that probability theory does not tell anything about the accuracy of the product rule estimates for the case when the independence is just approximately true. Hence, in that case we have no theoretical justification for claiming that the product rule is the optimal way of combining the local posteriors. We surely do not want to conclude here based on just one experiment that the averaging rule is better than the product rule. Our message is rather that averaging cannot not be ruled out by simple theoretical claims, as in some cases it may be just as good as multiplication. What makes the product rule a lucky choice is that its huge bias is not really detrimental for classification performance – while it is definitely beneficial for segmentation.

Further intuitive arguments can be brought for the averaging rule from information extraction considerations. 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 to be preferred 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 phone classification accuracies presented in Section 7 are just slightly better (around 62%) than the frame-based classification accuracy given in Section 3 (58.52%). This 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 the segmental representation is able to yield a phone classification accuracy of 67.83%. Both the intuitive arguments above and these results suggest 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 now has 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) (Hagen and Morris, 2005). But these models 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. (Ming and Smith, 2001) and Chan et al. (Chan and Siu, 2005), which is reported to be robust against short-time temporal corruptions. Although we did not test it, one expects the averaging rule to be more tolerant to impulse-like noises than the product rule as averaging dampens the local classification errors, while multiplication amplifies them (Tax et al., 2000). 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 most important argument in this paper was that the underestimation/overestimation bias of the product rules is such that it automatically enables these rules to handle outlier, non-phonetic segments. Put another way, we claimed that the factor $P(S_i \mid X_{s_{i-1}}^{s_i-1})$ is inherently contained in the product rules by means of the sum of their estimates. We proved this conjecture by removing this factor by normalization, after which the product rules gave more accurate posterior estimates, according to their marginal distributions, but were 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. To explain why this works, we argued that the sum of the estimates obtained with the product rule is proportional to the coherence

of the frame-based results. Overall, we may conclude that the product rule forms the basis of not only the classification performance of the hybrid, but also its segmentation performance. Taking into consideration the fact that averaging could handle the classification task just as well, we might say that the product rule's contribution to the segmentation subtask of decoding is much more important. This is indeed a surprising finding, at least compared to the original generative HMM concept where the product rule is supposed to model the observations emission probabilities within a state, while the positioning of the segments is supposed to be described by the state transition probabilities.

Comparing the phone recognition and word recognition tests, we saw that better phone recognition does necessarily not mean better word recognition, and vice versa. First, the product rule that was clearly inferior in phonetic classification worked fine in word decoding, while the simplified product rule that was much better in phone recognition worked abysmally in word recognition. Second, our newly proposed averaging hybrid also required some modification – a scaling factor for the segmentation probability component – for optimal word recognition. It has been observed by many researchers that although better phone recognition usually results in a lower word error rate, it is not necessarily the case (Morgan and Bourlard, 1995; Greenberg and Chang, 2000; Saraçlar et al., 2000). All authors agree that the lack of very good pronunciation models that account for all the possible pronunciation variants of the words can easily result in degraded word recognition. But the situation is not that simple: it seems that there are many other factors that influence this behavior. For example, it was found important if the models are trained using word-level or phone-level annotations and if the refined pronunciation models are used during the training of the models as well, or only during recognition (Saraçlar et al., 2000). As regards the division by the priors in the hybrid, based on our segmental probability measurements we can state that the division by the priors actually makes the segmental estimates worse, so it does not help by improving the acoustic models, but rather by making the cooperation with the language model (pronunciation dictionary) more successful, in a way that is not fully clear to us. On the need for the scaling factor in the averaging hybrid a possible explanation might be that with the introduction of another expert – the dictionary – a smaller weight is required for the segmentation expert. But this is only a possible hypothesis, and taken together with the findings in the literature, we have to conclude that the connection between optimal phone and word recognition performance, and the role of pronunciation modelling in it definitely requires further research.

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. Currently we cannot satisfactorily explain this phenomenon. Hence we plan to study these distributions with other, more sensitive methods in order to learn more about their actual behavior.

References

Austin, S., Zavaliagkos, G., Makhoul, J., Schwartz, R., 1992. Speech recognition using segmental neural nets. In: Proceedings of ICASSP'92, vol. 1, pp. 625–628.

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

Bourlard, H., Hermansky, H., Morgan, N., 1996. Towards increasing speech recognition error rates. Speech Communication 18, 205–231.

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

Clarkson, P., Moreno, P.J., 1999. On the Use of Support Vector Machines for Phonetic Classification. In: Proceedings of ICASSP'99, pp. 585–588.

Domingos, P., Pazzani, M., 1997. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–130.

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

Glass, J.R., 1996. A probabilistic framework for feature-based speech recognition. In: Proceedings of ICSLP'96, pp. 2277–2280.

Greenberg, S., Chang S., 2000. Linguistic dissection of switchboard-corpus automatic speech recognition systems. In: Proceedings of ISCA Workshop on ASR: Challenges for the New Millenium, pp. 195–202.

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

Hennebert, J., Ris, C., Bourlard, H., Renals, S., Morgan, N., 1997. Estimation of global posteriors and forward–backward training of hybrid HMM/ANN systems. In: Proceedings of Eurospeech'97, pp. 1951–1954.

Huang, X., Acero, A., Hon, H-W., 2001. Spoken Language Processing. Prentice Hall, New Jersey, USA.

Jelinek, F., 1996. Five speculations (and a divertimento) on the themes of H. Bourlard, H. Hermansky, and N. Morgan. Speech Communication 18, 242–246.

Lee, S.C., Glass, J.R., 1998. Real-time probabilistic segmentation for segment-based speech recognition. In: Proceedings of ICSLP'98, pp.1803–1806.

Lee, K.-F., Hon, H.-W., 1989. Speaker-independent phone recognition using hidden Markov models. IEEE Transactions on Acoustics, Speech and Signal Processing 37 (11), 1641–1648.

Leung, H.C., Hetherington, I.L., Zue, V.W., 1992. Speech recognition using stochastic segment neural networks. In: Proceedings of ICASSP'92, vol. 1, pp. 613–616.

Ming, J., Smith, F.J., 2001. Union: A model for partial temporal corruption of speech. Computer Speech and Language 15, 217–231.

Morgan, N., Bourlard, H., 1995. An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition. Signal Processing Magazine, May, 25–42.

Morris, A. C., Payne, S., Bourlard, H., 2002. Low cost duration modeling for noise robust speech recognition. In: Proceedings of ICSLP 2002, pp. 1025–1028.

Ostendorf, M., Digalakis, V., Kimball, O.A., 1996. From HMMs to segment models: a unified view of stochastic modeling for speech recognition. IEEE Transactions on Speech and Audio Processing 4 (5), 1063–6676.

Rabiner, L., Juang, B.-H., 1993. Fundamentals of Speech Recognition. Prentice Hall, New Jersey, USA.

Saraçlar, M., Nock, H., Khudanpur, S., 2000. Pronunciation modeling by sharing Gaussian densities across phonetic models. Computer Speech and Language 14, 137–160.

Tax, D.M.J., van Breukelen, M., Duin, R.P.W., Kittler, J., 2000. Combining multiple classifiers by averaging or by multiplying? Pattern Recognition 33, 1475–1485.

Tóth, L., Kocsor, A., 2005. Explicit duration modelling in HMM/ANN Hybrids. Proceedings of TSD'2005, pp. 310–317.

Tóth, L., Kocsor, A., Gosztolya, G., 2004. Telephone speech recognition via the combination of knowledge sources in a segmental speech model. Acta Cybernetica 16, 643–657.

Verhasselt, J., Illina, I., Martens, J.-P., Gong, Y., Haton, J.-P., 1998. Assessing the importance of the segmentation probability in segment-based speech recognition. Speech Communication 24 (1), 51–72.

Vicsi, K., Tóth, L., Kocsor, A., Csirik, J., 2002. MTBA – A Hungarian Telephone Speech Database. Híradástechnika, LVII (8) (in Hungarian). http://alpha.ttt.bme.hu/speech/hdbMTBA.php.

Young, S. et al., 1995. The HMM Toolkit (HTK) – software and manual. http://htk.eng.cam.ac.uk.

Zavaliagkos, G., Zhao, J., Schwartz, R., Makhoul, J., 1994. A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition. IEEE Trans. Speech and Audio Proc., 2(1), Part II, pp. 151–159.