Incorporating acoustic-phonetic knowledge in hybrid TDNN/HMM frameworks

Article *in* Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing · January 1992

DOI: 10.1109/ICASSP.1992.225882 · Source: IEEE Xplore

2 authors:

Christian Dugast
Deutsches Forschungszentrum für Künstliche…
17 PUBLICATIONS **323** CITATIONS
SEE PROFILE

Laurence Devillers
Computer Sciences Laboratory for Mechanic…
152 PUBLICATIONS **2,920** CITATIONS
SEE PROFILE

Combining TDNN and HMM in a Hybrid System for Improved Continuous-Speech Recognition

Christian Dugast, Laurence Devillers, and Xavier Aubert

Abstract— **The paper presents a hybrid continuous-speech recognition system that leads to improved results on the speaker dependent DARPA Resource Management task. This hybrid system, called the combined system, is based on a combination of normalized neural network output scores with hidden Markov model (HMM) emission probabilities. The neural network is trained under mean square error and the HMM is trained under maximum likelihood estimation. In theory, whatever criterion may be used, the same word error rate should be reached if enough training data is available. As this is never the case, the idea of combining two different criteria, each of them extracting complementary characteristics of the feature is interesting. A state-of-the-art HMM system will be combined with a time delay neural network (TDNN) integrated in a Viterbi framework. A hierarchical TDNN structure is described that splits training into subtasks corresponding to subsets of phonemes. This structure makes training of TDNNs on large-vocabulary tasks manageable on workstations. It will be shown that the combined system, despite the low accuracy of the hierarchical TDNN, achieves a word error rate reduction of 15% with respect to our state-of-the-art HMM system. This reduction is obtained with a 10% increase only in the number of parameters.**

I. Introduction

SINCE their resurgence, neural networks (NN) have induced a considerable activity in continuous-speech recognition. This renewed interest partly stems from the use of different training criteria together with highly connected structures that do not require any statistical assumption related to the underlying process. As opposed to the conventional maximum likelihood estimation (MLE) applied in most hidden Markov model (HMM) based systems, NN are generally trained under some discriminative criterion.

An experimental study is reported on that investigates the contribution of a time delay neural network (TDNN) trained under mean square error (MSE) [1], to a continuous HMM (CHMM) system trained under MLE [2]. The aim of this paper is to present a hybrid connectionist/Markov model recognizer with a simple design, running on a mono-processor computer. Experiments have been performed on a large-vocabulary continuous-speech recognition task, namely the speaker-dependent DARPA Resource Management database (RM1). By combining the normalized output scores of an integrated TDNN with the CHMM emission probabilities, improved recognition results have been observed with respect to a state-of-the-art HMM system.

An obvious bottleneck in NN systems is the tremendous computing power required for training. Therefore, hybrid-systems applied to large databases are usually implemented on multiprocessor computers (for example, 64 transputers or five ring-array processors in [3]–[5]). To cope with this CPU requirement, we resorted to an architecture of hierarchically ordered TDNNs running on a 20-MIPS workstation. This allows the training process to be divided into subtasks corresponding to subsets of phonemes.

TDNNs, by definition, do not rely on a time-alignment concept. This can be solved by integrating the TDNN in a Viterbi framework [6], [7]. As a matter of fact, the output scores are identified after appropriate normalization as posterior probabilities of the output labels [8]. Accordingly, after being divided by the priors, the normalized output scores can be interpreted as emission probabilities from an equivalent HMM state [6], [4]. The hybrid system resulting from such a treatment will be called an integrated TDNN, whatever the underlying TDNN structure, hierarchically ordered or not.

This way, a consistent framework is achieved that permits direct comparison of the TDNN/MSE pair with the HMM/MLE pair. Furthermore, it makes it easy to combine the scores resulting from each system, leading to improved recognition performances.

Three recognition systems will be presented in the next sections: a CHMM-based system, a first hybrid system called integrated hierarchical TDNN and a second hybrid system called the combined system. All three systems share a common acoustic analysis and decoding procedure, based on the same HMM topology (see Fig. 1). This permits comparison of the three systems solely on the training methods discussed.

The organization of the paper is as follows. In Section II we propose a hierarchical structure of TDNNs that is trainable on a workstation. We discuss its integration in a Viterbi framework. In Section II-C we briefly describe the CHMM training procedure (for more details see [2]) that makes use of a linear discriminant analysis (LDA). Section III will be devoted to the combined system. Section IV presents the general framework of the recognition system, together with the acoustic analysis and the decoding procedure. Finally, in Section V experiments are presented and discussed. Emphasis will be put on minimizing training time and number of parameters versus word error rates.

It will be shown that, despite the low accuracy of the hierarchical TDNN, the combined system leads to a relative

Manuscript received February 2, 1993; revised September 15, 1993.

C. Dugast and X. Aubert are with Philips Research Laboratories, P.O. Box 1980, D-52021 Aachen, Germany.

L. Devillers is with LIMSI-CNRS, BP 133, F-91403 Orsay Cedex, France.

This work was partly supported by the ESPRIT project number 2104 (Polyglot).

IEEE Log Number 9214673.

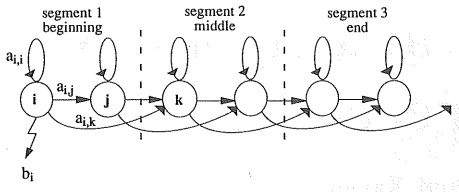

Fig. 1. Topology of a hidden Markov model. The emission probability b_i of state i is tied over the two states of the segment.

word error rate improvement of 15% to 20% with respect to our best CHMM system.

II. Integrated TDNN

TDNNs were introduced by Waibel [1]. Compared to a multilayer perceptron, the TDNN architecture offers the advantage of reducing the number of weighted connections and has the interesting property of time translation invariance. In contrast to recurrent networks (RN) that employ internal feedback to model context dependency, TDNNs use a fixed-length contextual input window to model dynamics of speech patterns. This simplifies the implementation of the gradient descent algorithm.

In the case of our TDNN, a phoneme is modeled with a unique network output, for which a complex time modeling like the one given in Fig. 1 is not necessary. But considering a phoneme along the time axis, it is obvious that the network acts differently at the beginning of the phoneme than at the end, varying in the type of errors made. A confusion matrix differentiating between beginning, middle and end of a phoneme is therefore helpful. This confusion matrix is evaluated during a Viterbi forced time-alignment procedure. We now have for each state s (see Fig. 1) a probability resulting in the multiplication of the TDNN emission probability by the confusion probability of being in state s (beginning, middle or end) while being in phoneme c. So, exactly the same topology will be used in all recognition systems.

A. Modular Architecture

The amount of data used in the DARPA RM1 database requires a huge network to discriminate between the 2500 phone units seen during training. Such a network needs an unreasonable number of cells, hence it is not trainable on a 20-MIPS machine in a reasonable amount of time. To overcome this problem, the "divide and conquer" principle has been used. Instead of training a large unique network to capture all the regions in the feature space, we have used a modular TDNN architecture [9]. The organization into subnetworks can be seen as introducing *a priori* knowledge or constraints about the likely phone ambiguities. Each subnetwork is trained to classify a subset or broad-class of phones. The set of phonemes has been partitioned without overlap into nine broad-classes of phonemes. The different consonant classes have been determined by the articulation mode: plosives, fricatives, glides... The position of the formants F1, F2 has been used to distinguish between three different vowel classes corresponding to the tops of the vocalic triangle [7]. To each broad-class of phonemes there is a corresponding subnetwork.

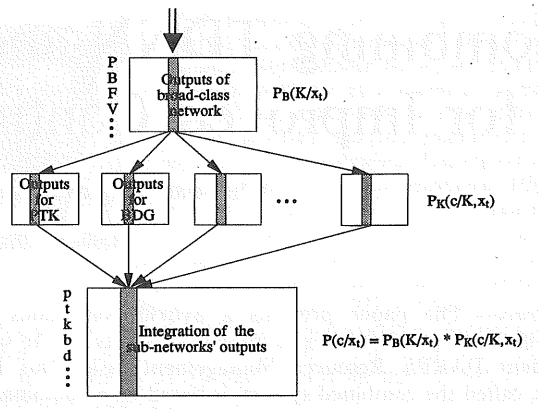

Fig. 2. Hierarchical structure of networks. The observation is noted x, the phoneme c belongs to subnet K. K is also a label of broad-class network B.

A broad-class network, the outputs of which are the labels of each subnet of phones, has also been trained; it has the role of connecting the subnetworks. A second training pass can be made to train all subnetworks and the broad-class network together in a single network. This second pass is called "connectionist glue."

This second training step requires an additional amount of training time with no recognition improvement in comparison to a direct integration of the set of networks in a Viterbi framework [7]. The proposed integration of the subnetworks follows a hierarchical structure.

B. Hierarchical Structure

The modular architecture described can be compared to a tree, with the root being the broad-class network and the leaves being the specialized subnetworks (see Fig. 2). We now have a two-step classifier that first decides between broad-classes, then within a broad-class. Once the different networks have been trained, integration in the HMM formalism is straightforward: the phoneme posterior probabilities within a broad-class have to be multiplied by their corresponding broad-class posterior probabilities.

Like for any two step probabilistic classifier, this structure implies suboptimality. Phones are trained *within* a broad-class ignoring anything about other phones that do not belong to their own class. To avoid this, the notion of *trap* has been introduced [7], [11]. A trap, attached within a broad-class, is an output node that identifies phones not belonging to the class. As it resulted in a rather small improvement of the word error rate for a more complex structure, it will not be further discussed here.

Generally speaking, when a broad-class network B discriminates between different classes of phonemes K, each of these classes discriminates in turn between phonemes (c) in a subnetwork K (the name of a broad-class label in the broad-class network is the same as the name of its corresponding subnetwork):

$$P(c|x_t) = P_B(K|x_t) \times P_K(c|K, x_t), \qquad (1)$$

where $P_B(K|x_t)$ denotes the posterior probability of broad-class K given the observed x_t in the broad-class network B

and $P_K(c|K, x_t)$ the posterior probability of phoneme c given the observed x_t in the subnetwork K (see Fig. 2).

In the case of a hierarchical structure of TDNNs, the posterior probability $P(c|x_t)$ that will be used to evaluate the probability density function of an HMM state is now provided by (1). The resulting system that integrates a hierarchically ordered set of TDNNs will be called integrated hierarchical TDNN, or simply Hierarchical TDNN (H-TDNN).

C. Training

Training of the subnetworks has been performed using a stochastic gradient descent method that has been shown to be faster than a deterministic gradient [12]; after one cycle, i.e., one presentation of a pattern for each label, the parameters are updated. In order to avoid overtraining, we used cross-validation during the training [6]. The TANH sigmoid function is applied to each node. All subnets are composed of three layers, where the dimension of the last one corresponds to the number of labels to discriminate. The width of the input window has been set to 70 ms of speech for generating an output vector, i.e., the width of the contextual window is 30 ms for the TDNN first layer and 50 ms for the second one. This 70 ms window corresponds to the average phone length in the DARPA database. A NN needs a segmented training set that provides it with initial targets. This segmentation has been generated for the TDNN by the continuous HMM system presented in this paper (with context-independent models). During training, the window was time-shifted, so the network produced an output vector every 10 ms.

Each subnet has been trained separately with a different number of hidden cells depending on the size of the recognition task to be performed: 18 hidden cells on average for a subnet, 50 hidden cells for the broad-class network. The difference between a subnet and a broad-class network comes from the definition of the desired outputs. In the subnets, we have associated a unique phone label with each output. In the broad-class networks, a set of phone labels is associated with each output.

III. Continuous Mixture Density HMM

This section gives a brief description of the baseline system that relies on continuous mixture density HMM's of triphones. What is described here is specific to the CHMM system and does not concern TDNNs. Linear discriminant analysis (LDA) is used as a pre-processing step in lieu of a feature extraction from a large acoustic vector. The training of the HMM parameters is based on a maximum likelihood criterion, the Viterbi approximation being applied at the level of both the state sequences and the emission probability contributions of the individual density components.

A. Acoustic–Phonetic Modeling

Each subword unit is represented by a three-state left-to-right HMM (see Fig. 1). Let $x_1, \cdots, x_t, \cdots, x_N$ be the time sequence of observation vectors and $s = 1, \cdots, S$ be any state of the Markov chain.

The emission probability density function associated with each state s is assumed to be of the form [13]:

$$Q(x_t|s, \Theta_s) = \sum_{k=1}^{K(s)} w_{k,s}\, b_k(x_t|s, \theta_{k,s}) \,, \qquad (2)$$

where each mixture component $b_k(.|.)$ is a unimodal density with parameter vector $\theta_{k,s}$ (e.g., mean and covariance), $w_{k,s}$ are the mixture weights subjected to stochastic constraints, $K(s)$ is the number of component densities and θ_s is the global parameter vector including $\theta_{k,s}$ and the weights.

In the present case, Laplacian-type densities have been implemented. The vector of absolute deviations has been pooled over all mixtures and states while the location vector has been treated specifically for each mixture component.

Triphones have been selected on the basis of their number of occurrences in the training script: 134 short function word triphones and 635 triphones were selected in addition to the 46 phoneme-like units, making a total of 815 models [2].

B. Linear Discriminant Analysis

The basic idea of linear discriminant analysis is to find a linear transformation such that the class separability is increased [8]. According to a previous study [14], classes are identified with context-dependent phoneme states and the affiliation of each training pattern is automatically obtained as a by-product of a standard HMM training carried out in the original acoustic space.

Concerning the acoustic features, the inclusion of time differences appears to be very important prior to the transformation and moreover, adjoining several centi-second frames leads to improved results.

C. HMM Learning Algorithm

The parameters of the system are trained with the Viterbi approximation, e.g., using the single best state sequence. In addition, the sum over the mixture components in (2) is replaced by the maximum operation. This is a common approximation that amounts to neglecting the overlap between mixture distributions and simplifies substantially the estimation procedure.

Each iteration consists of two alternating operations:

- Nonlinear time alignment: this is performed by dynamic programming and provides a new state labeling of the training data, e.g., a one-to-one correspondence between observation vector and state index.
- Maximum likelihood re-estimation: using this labeling, the parameters are updated by means of a decision-directed estimation technique [8].

As a starting point, the training sentences are linearly segmented into phoneme states and each mixture is initialized with a single Laplacian density. In the course of the training process, new density components are gradually introduced in a data-driven manner to provide an improved statistical fitting of the data [15], [2].

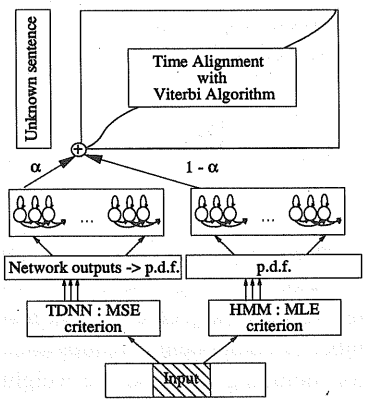

Fig. 3. Structure of the combined system. Input feeds the two structures, TDNN and HMM. The time alignment procedure is described in Section V-B.

IV. Combined-System

It is intended now to describe a hybrid system that linearly combines the output scores of Section II with the probabilities of Section II-C. We will first report on related work to position this paper clearly.

A. Related Work

Recently, several papers have been published describing attempts at using neural networks (NNs) in conjunction with dynamic programming (DP) and with the hidden Markov models (HMMs) time-alignment framework. Most of them propose theoretical issues [16], [12] but until now, on large databases, few results have been obtained showing better performances compared to state-of-the-art HMMs.

The various hybrid systems can be distinguished based on how the NNs are integrated in the HMM. NNs can be used as preprocessor of an HMM with or without global optimization of both systems, i.e., sequential learning of the two models [5], [3], [17] (see Section II) or embedded models [18], [19]. NNs can also be used as postprocessors of the HMMs [20]. Another approach, named alphanet [16], consists of viewing the forward pass of the Baum–Welch algorithm as a particular type of recurrent network. Our approach, which we name combined system, consists of a linear combination of HMM and NN probabilities during the recognition phase. The work of Renals *et al.* [4] is similar in spirit.

B. Combination of Models

Under combined system, we understand a recognition system consisting of two independent systems. Each of these systems (a TDNN and a continuous HMM) are trained separately: in our case the TDNN hierarchical structure is integrated in an HMM formalism (as in Section II-B) and the emission probabilities of the integrated H-TDNN are linearly combined with those of the CHMM from Section II-C (see Fig. 3).

The combination rule for a state s is as follows:

$$\log P_c(x_t|s) = \alpha \times \log P(x_t|s) + (1-\alpha) \times \log Q(x_t|s) \quad (3)$$

where $P(x_t|s)$ corresponds to the emission probability given by the H-TDNN (1) and $Q(x_t|s)$ corresponds to the CHMM (2) emission probability. The parameter α is chosen positive and less than one after appropriate scaling. α could be trained with deleted interpolation and separately evaluated for each phoneme.

V. General Framework

The same recognition system has been used for all trained models presented here. Acoustic analysis and decoding are common to all experiments, only the evaluation of the parameters has been done according to the training criteria and structures discussed here.

A. Acoustic Analysis

The acoustic signal is low-pass filtered and digitized with a sampling frequency of 16 kHz. A 512-point FFT is performed every 10 ms and 30 cepstrally smoothed logarithmic intensities are sampled in the frequency range from 200 Hz to 6400 Hz, corresponding roughly to a Mel-frequency scale. Each acoustic vector $y(t)$ is further augmented by slope and curvature information over the time axis, following:

$$x_t := \begin{bmatrix} y(t) \\ y'(t) \\ y''(t) \end{bmatrix} \approx \begin{bmatrix} y(t) \\ y(t) - y(t-T) \\ y(t+T) - 2y(t) + y(t-T) \end{bmatrix}. \quad (4)$$

The time delay T for computing the differences is 30 ms. For the slope and curvature of the 30 spectral intensities, pairs of adjacent channels are averaged so that the total number of components is 63. This will be referred to as "slope curvature'"" vector. To evaluate the importance of derivatives in the feature space for TDNN, only the 30 spectral intensities plus energy as well as a between solution have been considered. The between solution, called "short vector" has been obtained by taking only every other point along the frequency axis. These 15 spectral intensities are then in turn appended with their respective slope values together with the average energy value and its first and second derivatives, leading to 33 components in total.

B. Time Synchronous Decoding

The recognition procedure proceeds by searching for the most likely state sequence, this sequence being given by Fig. 1 for a phoneme-like unit. These units are concatenated to build words. With a 1000-word lexicon, the potential search space is relatively manageable without having to resort to particular techniques like lexical tree-organization and phoneme look-ahead [21]. Therefore, the recognition has been accomplished with a data-driven organization of the Viterbi beam search algorithm [22], with all lexical and syntactical constraints being dynamically expanded. The emission probability values are computed on demand according to the estimation criterion being tested (see (2) for CHMM, (1) for integrated H-TDNN and (3) for the combined system).

VI. Experiments / Results

After briefly describing the database on which experiments were run, we will report on key-features that lead to improvements of a TDNN (without any time alignment concept). Next,

TABLE III
AVERAGE WORD ERROR RATE (WER) IN THE WORD-PAIR GRAMMAR CASE WHEN COMBINING THREE DIFFERENT CHMM SYSTEMS WITH THE SAME H-TDNN INTEGRATION, OVER THE THREE SPEAKERS ON 100 DEVELOPMENT TEST SET SENTENCES

Continuous HMM		H-TDNN	Combination H-TDNN ⊕ CHMM
Acoustic-Phonetics	WER	WER	WER
47 Context Indep. Phones	3.2%	6.0%	2.5%
+ 769 Context Dep. Phones	2.2%	-	1.8%
+ LDA	2.0%	-	1.7%

TABLE IV
ERROR RATES PER SPEAKER FOR THE H-TDNN INTEGRATION, CONTINUOUS HMM AND COMBINATION OF BOTH, IN THE WORD-PAIR GRAMMAR CASE

Speaker	H-TDNN	CHMM	Combination
JWS0	4.7%	2.1%	1.8%
CMR0	4.5%	2.7%	2.3%
BEF0	8.9%	1.1%	1.0%
Average	**6.0%**	**2.0%**	**1.7%**

three speakers selected are summarized in Table III; Table IV gives detailed results of the best CHMM system configuration for the three selected speakers.

As for the combined system, first, the log probabilities of the two systems to be combined had to be scaled the same way.

The value of α ((3)) has been empirically varied between 0.05 and 0.50 and a good value of 0.20 has been determined whereby 0.15 and 0.25 where nearly as good. With an α value greater than 0.35, the word error rates were getting near to those obtained with the integrated H-TDNN system alone. More tests will be necessary in order to find the best interpolation factor between both probabilities.

Table III gives the average results obtained on three speakers with the two hybrid systems discussed here: integrated hierarchical TDNN (H-TDNN) hybrid system and combined H-TDNN $\oplus$ CHMM system. Results involving the main features which lead to improvements of the CHMM system are also shown for the sake of comparison, starting with the base system with 47 context-independent models, then with an addition of 769 context-dependent phones, and the last one including a linear discriminant analysis (LDA) before modeling the 769 + 47 phones. The latter configuration gives results that are among the best published on the same database. The same H-TDNN system (modeled only with 47 outputs corresponding to the 47 context-independent phones) is combined with the three different CHMM configurations. It can be observed that despite its rather low accuracy when taken alone, the H-TDNN improves results when combined with CHMMs. A relative improvement of 20% with respect to the base CHMM system is achieved. This relative improvement is still at 15% when the H-TDNN is combined with the state-of-the-art CHMM system. Table IV gives detailed results per speaker. Recovered errors have been observed only on substitutions: one on a long word, the others on short functional words.

TABLE V
FIGURES FOR ONE SPEAKER[a]

	CHMM	H-TDNN	Combination
# parameters	430	40	470
local distance	4×rt	1×rt	5×rt
training time	8 hours	1 month	1 month
WER	2.0%	6.0%	1.7%

[a]The number of parameters is in 1,000; rt stands for real-time (a 20-MIPS workstation has been considered); WER stands for word error rate.

VII. CONCLUSION

Two practices are of importance to improve TDNN frame error rates: 1) including derivatives from the feature space vector and 2) setting the target outputs to $+/-0.9$. These two techniques together helped reduce the frame error rate by a factor of two. It is surprising that TDNNs are unable to extract derivatives from their input window. This has also been observed for LDA on CHMMs [14]. As for the practice of setting the desired outputs to $+/-0.9$, no freezing of connections is allowed during training: they would lead to early local optima.

The combined system gave impressively good results relative to the rather poor results from the integrated H-TDNN system. What is decisive here is the use of a different training scheme, offering the extraction of other classification characteristics that are complementary to the first ones. MSE used to train the neural networks extracts parameters more at class boundaries than the HMMs trained with MLE that extracts parameters from class centers. In theory, whatever training criterion may be used, the same word error rate should be reached if enough training data is available. For large-vocabulary continuous-speech recognition, this is never the case. The idea of combining two different criteria, each of them extracting different characteristics of the feature space, is a way to cope with the problem of not enough training data in smoothing the emission probabilities obtained during training.

Table V shows general figures for one speaker that assist in understanding the relations between the different systems discussed here. First, an H-TDNN with a factor of ten parameters less than a CHMM has a word error rate a factor of three higher. What is more interesting, is the fact that for the combined system, an increase of only 10% in the number of parameters leads to a decrease of 15% in the word error rate. Furthermore, the resulting 25% increase in local distance calculation should not be seen as hindrance: in a real-life operating system, a special NN hardware would process the NN outputs in parallel with the CHMM local distance calculation.

The critical point that arises when viewing Table V is the CPU time needed to train the H-TDNN: on a 20-MIPS workstation, one month of CPU time is needed to train the whole H-TDNN structure as compared to one night for CHMMs. But, in taking advantage of the hierarchical structure, this one month of CPU time can be distributed over several workstations: each subnetwork can be trained on

a separate machine. Furthermore, the proposed hierarchical structure makes NN training manageable on workstations in dramatically increasing experiments rotation: even though a broad-class network needs 15 days of training time, experiments can be performed on subnetworks for which 1 to 2 days training time is needed. For the sake of comparison, about seven months of CPU time would be required for training a single network, instead of using the hierarchical structure presented here, for which 1,000 hidden nodes would be necessary on the RM1 task.

To conclude, the word error rate of a state-of-the-art system could be reduced by relatively 15%. This has been achieved in a very simple way despite the rather low accuracy of the hierarchical TDNN structure described.

REFERENCES

[1] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang, "Phoneme recognition using time delay neural networks," *IEEE Trans. Acoust. Speech Signal Processing*, 1989.

[2] X. Aubert, R. Haeb-Umbach, H. Ney, "Continuous mixture densities and linear discriminant analysis for improved context-dependent acoustic models,"*Proc. ICASSP'93*, Vol. II-648, Mineapolis, 1993.

[3] A. Robinson, "A real-time recurrent error propagation network word recognition system," *Proc. ICASSP'92*, Vol. I-617, San Francisco, CA, 1992.

[4] S. Renals, N. Morgan, M. Cohen, H. Franco, "Connectionist probability estimation in the decipher speech recognition system," in *Proc. ICASSP'92*, Vol. I-601, San Francisco, CA, 1992.

[5] H. Bourlard, N. Morgan, Ch. Wooters, S. Renals, "CDNN: A context dependent neural network for continuous-speech recognition," in *Proc. ICASSP'92*, Vol. II-349, San Francisco, CA, 1992.

[6] H. Bourlard, N. Morgan, and Ch. Wooters, *Connectionist Approaches to the Use of Markov Models for Speech Recognition, Advances in Neural Information Processing Systems 3.* ***What City??:*** Morgan Kaufman, 1991.

[7] L. Devillers and Ch. Dugast, "Comparison of continuous mixture densities and TDNN in a Viterbi-framework: Experiment on speaker dependent DARPA RM1," in *Proc. Eurospeech'91*, pp. 991–994, Genova, Italy, 1991.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience Publication, 1973 p. 155.

[9] A. Waibel, "Connectionist glue: Modular design of neural speech systems," in *Proc. 1988 Connectionist Models Summer School*, Carnegie Mellon Univ., Pittsburgh, PA, pp. 417-421, 1988.

[10] J. Bengio, R. De Mori, G. Flammia, R. Kampe, "Artificial neural networks and their application to sequence recognition," Ph.D. dissertation, McGill Univ., Montreal, PQ, Canada, 1991.

[11] L. Devillers, "Continuous Speech Recognition using a hybrid system combining neural networks and hidden Markov models," Ph.D. thesis (in french), Paris-South Univ., Orsay, France, 1992.

[12] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations.* Berlin: Springer-Verlag, 1990

[13] L. R. Rabiner, B. H. Juang, S. E. Levinson, M. M. Sondhi, "Recognition of isolated digits using HMMs with continuous mixture densities," *AT&T Tech. J*, vol. 64, no. 6, pp. 1211–1234, 1985.

[14] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large-vocabulary continuous-speech recognition," in *Proc. ICASSP'92*, Vol. I-13, San Francisco, CA, 1992.

[15] H. Ney, "Experiments on mixture-density phoneme-modeling for the speaker-independent 1000-word speech recognition DARPA task," in *Proc. ICASSP'90*, Albuquerque, NM, 1990, pp. 713–716.

[16] J.S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Advances in Neural Information*, 1990.

[17] Ch. Dugast and L. Devillers, "Incorporating acoustic-phonetic knowledge in hybrid TDNN/HMM frameworks," in *Proc. ICASSP'92*, vol. I-421, San Francisco, CA, 1992.

[18] P. Haffner, "Connectionist word-level classification in speech recognition," in *Proc. ICASSP'92*, vol. I-621, San Francisco, CA, 1992.

[19] M. Franzini, K.F. Lee, and A. Waibel, "Connectionist viterbi training: A new hybrid method for continuous-speech recognition," in *Proc. ICASSP'90*, Albuquerque, NM, 1990, pp. 425-428.

[20] S. Austin, G. Zavaliagkos, J. Makhoul, and R. Schwartz, "Speech recognition using segmental neural nets," in *Proc. ICASSP'92*, Vol. I-625, San Francisco, CA, 1992.

[21] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder, "Improvements in beam search for 10000-word continuous-speech recognition," *Proc. ICASSP'92*, Vol. I-9, San Francisco, CA, 1992.

[22] H. Ney, D. Mergel, A. Noll, and A. Paeseler, "A data driven organization of the dynamic programming beam search for continuous-speech recognition," *Proc. ICASSP'87*, Vol. II-833, Dallas, TX, 1987.

[23] P.J. Price, W. Fisher, J. Bernstein, and D. Pallett, "A database for continuous-speech recognition in a 1000-word domain," in *Proc. ICASSP'88*, New York, NY, 1988, pp. 651–654.

Christian Dugast was born in Fianarantsoa, Madagascar, in 1960. He received the Diplome degree in computer science, the Ph.D. degree (Doctorat de 3^{eme} cycle) in computer science from the University of Toulouse, Toulouse, France, respectively in 1983 and 1987. His Ph.D. work concerned search algorithms for continuous speech recognition.

From 1987 to 1989, he was a Knowledge Engineer involved in the design of an expert system in chemistry for BASF. He joined Philips Research Laboratories in Germany as a Research Scientist in 1989 to work on continuous speech recognition. His main research interests concerns acoustic-phonetic modelling based on neural network and continuous hidden Markov models. He his now responsible for work on American English. During his study at the University Paul Sabatier, Toulouse, he received the Toulouse Prize for its Diplome degree and a Research and Industry Ministery grant for his Ph.D. He was work-package manager of the "Continuous Speech Recognition" work-package within the European Esprit Polyglot project.

Laurence Devillers was born in France in 1962. She received the M.S. degree in electrical engineering in 1986 and the Ph.D. degree in computer science in 1992, both from the University of Orsay, France.

Since 1987 she is working at LIMSI-CNRS (France) in the field of speech recognition. She was involved in the ESPRIT project 860 exloring linguistic aspects of speech recognition. Between 1989 and 1992 she has developed a research group on neural networks, involved in the ESPRIT Polyglot project. She is currently lecturer at the University Institute of Technologies at Orsay and continues research at LIMSI. Her current research interests include artificial neural networks and hybrid systems combining neural networks and hidden Markov models.

Xavier Aubert received in 1977 the degree of engineer in applied mathematics from the University of Louvain, Louvain, Belgium, and received a Ph.D. in 1983 in the field of numerical simulation of free surface flows at the same University.

In 1985, he became Research Scientist at the Philips Research Laboratory Brussels where his activities have been essentially concerned with hidden Markov modeling applied to continuous speech decoding. In June 1991, he joined the Philips Research Laboratories in Aachen, Germany, where he is now working on large vocabulary continuous speech recognition. His current interests include acoustic modeling and search algorithms.