Context-dependent modeling and speaker normalization applied to reservoir-based phone recognition

Fabian Triefenbach, Azarakhsh Jalalvand, Kris Demuynck, Jean-Pierre Martens

ELIS Multimedia Lab, Ghent University/iMinds, Sint-Pietersnieuwstraat 41, B-9000, Ghent, Belgium
fabian.triefenbach@elis.ugent.be

Abstract

Reservoir Computing (RC) has recently been introduced as an interesting alternative for acoustic modeling. For phone and continuous digit recognition, the reservoir approach obtained quite promising results. In this work, we further elaborate this concept by porting some well-known techniques used to enhance recognition rates of GMM-based models to Reservoir Computing. In particular, we introduce context-dependent (CD) triphone states to model co-articulation and pronunciation mismatches arising from an imperfect lexicon. We also propose to incorporate two speaker normalization methods in the feature space, namely mean & variance normalization and vocal tract length normalization. The impact of the investigated techniques is studied in the context of phone recognition on the TIMIT corpus. Our CD-RC-HMM hybrid yields a speaker-independent phone error rate (PER) of 22% and a speaker-dependent PER of 20.5%. By combining GMM and RC-based likelihoods at the state level, these scores can be reduced further.

Index Terms: Reservoir Computing, Acoustic Modeling, context-dependency, speaker normalization

1. Introduction

Automatic speech recognition (ASR) has considerably improved over the last decades and has become an admitted enabling technology for multiple multimedia and information retrieval applications. A core component of an ASR is the Acoustic Model (AM) that captures the relation between the speech signal and the spoken sounds, the so-called phones. Most state-of-the-art speech recognizers rely on a combination of Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs). However, recent work on Multi-Layer Perceptrons (MLPs) [1, 2] and Deep Neural Networks (DNNs) [3–5] has shown that neural models can offer sustainable improvement over GMMs for continuous speech recognition on small and large vocabulary benchmarks.

Motivated by this success, we started to investigate Reservoir Computing (RC) [6–9] as an alternative neural-based approach to acoustic modeling. We already obtained quite competitive results for phone recognition [10–12] as well as for noise-robust continuous digit recognition [13, 14]. The underlying idea of RC is to combine the power of a Recurrent Neural Network (RNN) – a complex dynamical model – with the elegance of a linear regression model whose global optimum can be written in a closed form. The main differences with the traditional RNN-based approach [15] are: (1) in RC there are only recurrent connections between hidden neurons, (2) the output neurons are linear and (3) only the output weights are trained. Every hidden neuron can be connected to any other hidden neuron and the hidden layer can be considered as a pool of fixed (= non-trained) non-linear neurons that is called a *reservoir*.

Given that RC-based acoustic modeling is a fairly new approach, more research is needed to establish whether it can lead to improvements, and if so, whether these improvements will also lead to better large vocabulary continuous speech recognition (LVCSR). In this paper we address the first issue, and we take the context-independent RC-based phone recognizer developed in [10] as our point of departure. We investigate whether RC-based systems can benefit from techniques that were shown to improve GMM-based [16] and neural-based systems [1, 4]. More in particular, we introduce context-dependent phone states and speaker normalization methods such as Mean & Variance Normalization (MVN) and Vocal Tract Length Normalization (VTLN). In addition, we also investigate model combination (MC) at the state level.

In the following, we first describe our baseline RC-based phone recognizer and discuss the proposed extensions and improvements in Sections 3–5. The experimental validation is described in Sections 6 & 7. At the end of the paper we formulate our conclusions and propose some ideas for future work.

2. RC-based speech recognition

The combination of a reservoir of fixed non-linear neurons and a layer of linear output neurons driven by the reservoir state is called a reservoir network. The reservoir networks we employ in our systems are Echo State Networks (ESNs) [6], the most popular form of reservoir networks.

2.1. A reservoir network

At time t each reservoir neuron is driven by an input vector $U[t]$ and a delayed reservoir state vector $R[t-1]$. We utilize so-called Leaky Integrator Neurons (LINs) [7] with a leak rate $\lambda < 1$. The reservoir state at time t is computed as

$$R[t] = (1-\lambda)R[t-1] + \lambda f_{res}(\mathbf{W_{in}}U[t] + \mathbf{W_{rec}}R[t-1]) \quad (1)$$

with f_{res} being a non-linear activation function (e.g. $\tanh(.)$), and with $\mathbf{W_{in}}$ and $\mathbf{W_{rec}}$ encompassing the weights of the external and the recurrent input connections. All elements of the weight matrices are independently drawn from a zero-mean normal distribution. The variance of the recurrent weight distribution controls the spectral radius ρ, defined as the largest absolute eigenvalue of $\mathbf{W_{rec}}$ [6, 8]. If it is smaller than 1, the reservoir network is stable and it provides a fading memory of the past. The variance of the external input distribution, V_{in}, controls the impact of the inputs $U[t]$ on the reservoir state.

The linear output neurons of the reservoir network are called *readouts* because they 'read out' the reservoir state. They are computed as $Y[t] = \mathbf{W_{out}}R[t]$, with $R[t]$ being the reservoir state augmented with a bias and with $\mathbf{W_{out}}$ comprising the output weights. The output weight matrix is designed (= trained) to minimize the mean squared error between the readouts $Y[t]$ and the desired readouts $D[t]$ over the training examples [10]. The reservoir state space can be compared to the inner space of a Support Vector Machine (SVM) [17] with the difference that the reservoir space is untrained whereas that of an SVM follows from a delicate supervised training procedure.

2.2. An RC-HMM hybrid for speech recognition

The simplest way to construct an RC-based ASR system is to create an RC-HMM hybrid [18] that derives acoustic likelihoods from the outputs of a reservoir network. This network can be a simple network with one reservoir or a deep network, obtained by cascading multiple simple reservoir networks. Each simple network is then called a layer. The general architecture of the hybrid is depicted in Figure 1. The first layer processes the inputs $U[t]$ and the further layers process the outputs of the preceding layer. The layers are trained one after the other. Per layer, the optimal settings of the reservoir (ρ, λ, ...) emerge from an efficient user-controlled search procedure (see [10]). We have shown that new layers can actually correct some of the mistakes made by the preceding layer. We attribute this to the fact that every new layer offers additional temporal modeling capacity and a new inner space. The readout neurons of each layer represent phone states (see Figure 1) and their weights are trained with the desired outputs $D[t]$. The vector $D[t]$ is a unit vector with a non-zero entry at the position that corresponds to the desired phone state label at time t. Since the outputs of a neural network adhere to posterior probabilities [19] and since we want to find the phone state sequence emerging from

$$\hat{Q} = \arg\max_{Q} P(Q|U) = \arg\max_{Q} P(U|Q)\ P(Q),$$

we insert an extra step for converting the outputs of the last layer to likelihoods using Bayes' law [18]. In the case of phone recognition, the admissible state sequences can represent an arbitrary sequence of phones (including silence) and the prior probability $P(Q)$ is computed by means of an n-gram phone language model (LM).

Since reservoirs only provide a fading memory of the past, they make no use of the future. Yet, the theory of co-articulation [20,21] claims that a phone is also influenced by the next phone (= anticipation). Therefore, we have also considered bi-directional reservoirs [10]. The backward reservoir is identical to the forward reservoir, but it processes the data stream from right-to-left. Per layer, the readout neurons are trained on the joined reservoir state $\hat{R}[t]$ composed of the neurons outputs of both reservoirs. As explained in [10], one can implement bi-directional reservoirs in a 'real-time' system with a latency of no more than 100 ms.

3. Context-dependent phone states

The baseline reservoir systems investigated thus far [10] utilized context-independent (CI) phone states, whereas research on LVCSR with GMM-HMM systems has clearly demonstrated the benefits of context-dependent (CD) phone states. Models for such states allow the system to cope in a transparent way with cross-phone co-articulations and mismatches between the word pronunciations found in the lexicon and those actually employed by the speaker [21]. The main reason for our former reluctance to introduce CD states was that RC-based systems seem to need very big reservoirs (e.g. 20K neurons). But, as the number of trainable parameters is equal to the number of reservoir neurons times the number of states, the number of trainable parameters significantly increases when moving from e.g. 150 CI states to e.g. a few thousand CD states (as in [22,23]).

In this work we model the CD phone states that followed from a decision tree clustering using phonological questions that was performed during the training of a CD-GMM-HMM recognition system. Although this clustering is sub-optimal because it does not take the properties of the reservoir into account, our strategy conforms with the approach also adopted in recent work on DNNs [3,24] and MLPs [1,23].

4. Speaker-dependent models

The models developed thus far are speaker-independent (SI) models, trained on speech uttered by a large number of speakers. However, in many real-world applications of ASR, the speaker identity is known from the user profile (mobile devices, GPS, etc.) and the ASR can, after some time, adapt itself to the specific speaker. Many state-of-the-art GMM-HMM speech recognizers therefore apply speaker normalization and/or adaptation of their acoustic model [16, 25, 26]. In this work we investigate what speaker normalization in the acoustic feature space can achieve in an RC-HMM system. We consider two approaches: Mean & Variance Normalization (MVN) and Vocal Tract Length Normalization (VTLN).

4.1. Mean & Variance Normalization (MVN)

Due to physiological differences, the observed distributions of acoustic features across states differ from speaker to speaker. A simple approach to handle this variability is to normalize the distributions by shifting and rescaling the individual features [25,26]. Based on the data available for a certain speaker, the feature-wise mean & variance is estimated over all frames of that speaker. The estimated mean values are then subtracted from the raw features and the estimated variances are used to make the variances of the normalized features equal to the global variances, measured over all speakers.

4.2. Vocal Tract Length Normalization (VTLN)

One of the important physiological differences between persons is the length of their vocal tract. That of a male speaker for instance is on average 10-15% larger than that of a female speaker. Since the formant frequencies are proportional to this length, a large part of the speaker variability can be annihilated by normalizing the frequency scale during feature extraction [25, 26]. In this work, we use the VTLN procedure available in the SPRAAK toolkit [27]. It first estimates the most likely gender of the speaker and then it uses this information to compute a suitable warping factor.

5. Model combination (MC)

The likelihoods emerging from the RC-based AM at time t are actually based on a fading memory of the past. Consequently, they are supposed to differ considerably from the likelihoods at time t computed by a traditional GMM-based AM that just takes $U[t]$ into account. Since the two AMs use exactly the same CD states, we can easily combine their likelihoods at the state-level by computing the weighted sum of likelihoods [28]. The weighting factor is tuned on the development data.

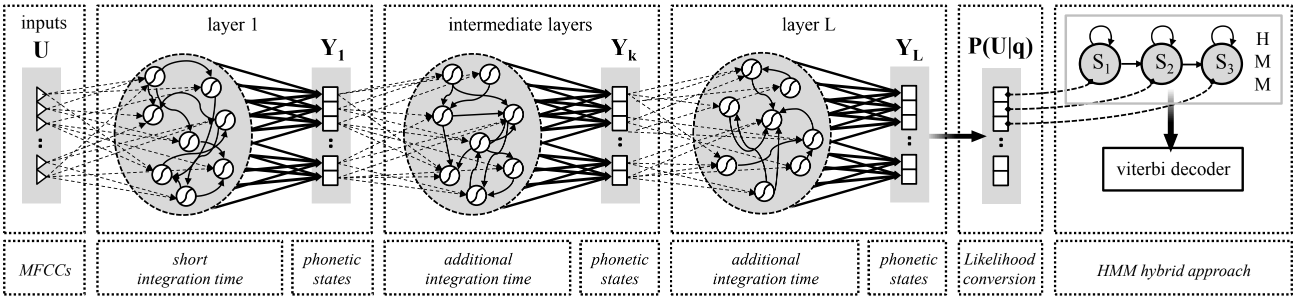

Figure 1: An RC-HMM hybrid speech recognition system comprising a multi-layer reservoir network. The outputs are converted to acoustic likelihoods and used by a Viterbi decoder that finds the most likely state sequence in a HMM representing all structural knowledge about the recognition task.

6. Experimental setup

6.1. Speech corpus & acoustic front-end

We perform phone recognition on the TIMIT corpus [29] that contains recordings of 630 native American speakers (438 males & 192 females) divided over eight dialect regions. Each speaker reads eight phonetically rich sentences (calibration sentences are discarded). The training set contains 462 speakers and the test set the remaining 168. 48 training speakers are used as a development set. We report Phone Error Rates (PERs) in % on the so-called core test set (24 speakers) which is a subset of the full test set. The PER counts the percentage of deletions, insertions and confusions between 39 phone classes (see [30]).

The acoustic front-end computes Mel Frequency Cepstral Coefficients (MFCCs) [31] based on 25 ms Hamming-windowed speech frames with a shift of 10 ms. A 24 channel mel-filterbank is used to compute the intermediate spectrum. Utterance-based Cepstral Mean Subtraction (CMS) is applied and the final feature vectors are composed of 13 normalized MFCCs (c_0,...,c_{12}) and their first and second order derivatives.

6.2. Acoustic models

In our experiments we investigate different RC-based and GMM-based Acoustic Models. We use three states per phone and there are 3x51 CI phone states[1]. In the case of CD phone states, we create models for 516 tied states resulting from a decision tree clustering [27].

The GMMs compute weighted sums of Gaussians selected from a fixed pool and the corresponding mixture models are developed by means of the SPRAAK toolkit [27]. The mixture weights are trained using Maximum Likelihood (ML) training. The number of Gaussians, states and mixtures are determined automatically from the size and the statistics of the data, so that the risk of over-fitting is low.

The reservoir neurons are sparsely connected. Each neuron is driven by 5 randomly selected inputs and by 5 randomly selected reservoir state components. The reservoir networks are trained by means of a Tikhonov regression [32]. As described in [12], the inputs to the first layer are divided in three subgroups, namely C, ΔC, $\Delta\Delta C$, and the features of each subgroup are rescaled so that the mean squared norm of the subvectors in the sub-groups are equal to 1.0, 0.7 and 0.3 respectively. The latter values express the relative importances of the sub-groups. Per layer, we determine the reservoir control parameters (V_{in}, ρ, λ) on the development set. The spectral radius was chosen differently for the first two layers (ρ_1=0.5,ρ_2=0.8). All further layers have a ρ = 0.6 (see [10] for more details). Leaky integration (λ=0.3) is only applied in the first two layers.

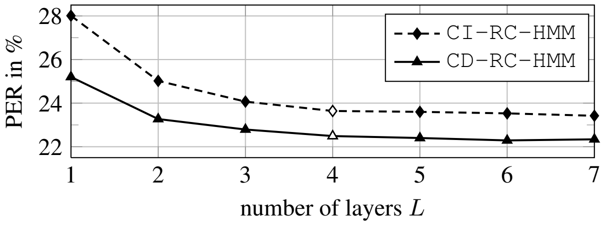

Figure 2: PER in % as a function of the number of layers.

6.3. Viterbi-decoder

Each phone is modeled by a three-state HMM and the language model (LM) is an n-gram stochastic phone LM derived from the hand label strings provided for the training utterances. The LMs are developed with the SRI language modeling toolkit [33] using Kneser-Ney discounting [34] with a discount of 5. We have designed a unigram, a bigram and a trigram. The test set perplexities are 42.6, 16.7 and 15.3 respectively.

The relative importance of the LM with respect to the AM is controlled by an exponent α_{LM} on the n-gram probabilities. The balance between phone deletions and insertions is controlled by a phone insertion penalty P_{ins} which is applied every time one enters a new phone. The decoder parameters α_{LM} and P_{ins} are optimized on the development set.

7. Experimental results

7.1. Context-dependent states

In a first experiment we test two types of uni-directional RC-HMM hybrids, one type utilizes CI state and one type utilizes CD states. All reservoirs comprise 20K neurons. Figure 2 shows the PER as a function of the number of layers per type. The PER monotonously decreases, but it starts to saturate at a depth of 4 layers. At that point the PER is 23.6% for the CI models and 22.5% for the CD models. Note that the CD models have 516 output classes, and thus incorporate about three times more trainable parameters than the CI models working with 153 classes. However, a CI system cannot reach the performance of

[1] Note that in [10] we reduced the number of states to 51 in the second and higher layers, but maintaining 3x51 states throughout is a more uniform approach and it performs marginally better.

Table 1: PER in % of different speaker-independent models (GMM-HMM, RC-HMM, bRC-HMM). Comparison between context-independent (CI) and context-dependent (CD) phone state modeling [LM: bigram].

model	**CI**	**CD**
GMM-HMM	27.8	25.5
RC-HMM (uni-directional)	23.6	22.5
bRC-HMM (bi-directional)	22.6	22.0

Table 2: PER in % of different speaker-independent context-dependent models (CD-GMM-HMM, CD-RC-HMM, CD-bRC-HMM). Model evaluation with different LMs. Additional results for model combination (MC) listed.

model (CD)	**unigram**	**bigram**	**trigram**
GMM-HMM	27.9	25.5	24.5
RC-HMM	22.8	22.5	22.1
bRC-HMM	22.3	22.0	21.5
MC: GMM+RC	22.2	21.7	21.2
MC: GMM+bRC	21.4	21.1	20.5

the CD system, even if it is allowed to comprise the same number of trainable parameters.

In Table 1 we list the benefit of CD state modeling. Clearly, GMM-based models benefit more from CD states than RC-based models. This was somehow expected since the reservoir already handles part of the cross-phone co-articulation through its temporal modeling capability. This fact also explains why a bi-directional system (bRC-HMM) with the same number of trainable parameters is even less improved by the introduction of CD states. Important for us was to establish that in a large reservoir state space it is possible to find a good regression model with a few hundred phonetic classes, in spite of the large number of training parameters such a model implies (516×20K≈10M parameters per layer). Apparently, the reservoir approach does not suffer from over-training. Note that a CI-DNN-HMM used for the same phone recognition experiments [5] has approximately the same complexity per layer (3K×3K=9M), but it uses 8 layers. Even though CD reservoir models do not yield a large improvement, we prefer them over CI models for two reasons. First of all, they are slightly better, but more importantly, they will be required in LVCSR [3,16,24] to cope with mismatches between the actual word pronunciations and the pronunciations available in the lexicon [21].

In Table 2 (upper part) we show results obtained with different phone LMs. Clearly, the RC-HMM systems do not benefit that much from higher-order LMs. This is not because they cannot exploit the contextual information, but because they automatically cope with this information through the reservoir memory. This explains why an RC-HMM system already performs so well in combination with a unigram LM. Nevertheless, the stronger LMs still have some effect on the recognition results of the RC-models. In case of a trigram LM the bRC-HMM system achieves a PER of 21.5%. The fact that reservoir models learn some phonotactics is a promising result in view of LVCSR. The word-level n-grams and the phonotactic constraints imposed by the reservoir may together constitute a better LM. Note that our speaker-independent results are very competitive with those of DNNs [5,35] (PER=22-23%) and CD-MLPs [1] (PER=21.2%) when the same acoustic features are used.

Table 3: PER in % of different speaker-dependent context-dependent models (CD-GMM-HMM, CD-bRC-HMM). Comparison between different normalization methods. Additional results for model combination (MC) listed.

model (CD)	SI	**VTLN**	**MVN**	**VTLN +MVN**
GMM-HMM	25.5	24.8	25.0	24.4
bRC-HMM	22.0	21.1	21.1	20.5
MC: GMM+bRC	21.1			19.8

By applying state-wise likelihood combination with a mixture weight of 0.5 we could further reduce the PER with all investigated LMs by about 1% (Table 2, lower part). This proves that the RC-based and GMM-based models are to some extend complementary. Note that in [4] the combination of a GMM and a DNN only yielded an improvement of 0.1%.

7.2. Speaker-dependent (SD) modeling

In this section we discuss the impact of speaker normalization. The MVN and VTLN measurements are performed on the 8 test sentences of each speaker (as in [4,16]). Table 3 shows that both approaches applied independently lead to improved recognition results for a GMM and a reservoir-based system, with MVN being more beneficial for the neural approach. If both normalization methods are applied together, the individual gains nearly add up and the PER of the RC system can be reduced to 20.5%. A combination of the RC and GMM-based model on state-level leads to a final speaker-dependent PER of 19.8%.

8. Conclusions & future work

In this work we introduced context-dependent phone states for RC-based acoustic modeling and showed that reservoirs with large linear output layers can be trained successfully. The resulting models provide a small improvement for phone recognition and a final speaker-independent phone error rate of 22.0% is obtained. This figure is very competitive with the figures published for other methods applied to phone recognition. In spite of the relative small gain we were able to demonstrate on TIMIT, we are contented with this result because it manifests that a context-dependent reservoir network can be trained uncomplicated. We are convinced that these CD models are indispensable for good LVCSR which is, after all, our final goal. In future work we will investigate how beneficial the temporal modeling of a reservoir still is in a full recognizer using a dictionary, transcription-based segmentation and a word-level LM.

We have also analyzed some speaker normalization approaches, namely VTLN and MVN, to canonicalize the speakers. Both methods improve the recognition accuracy of an RC-based system and lead to a PER of 20.5% in the case of a bigram phone LM. Additional model adaptation [13] in complement to this feature normalization may provide additional improvements. This is another route to explore in future work.

9. Acknowledgments

The work presented in this publication was accomplished within the RECAP project funded by the Research Foundation Flanders (FWO) under grand G.0088.09N.

10. References

[1] L. Toth, “A hierarchical, context-dependent neural network architecture for improved phone recognition,” in *Proc. ICASSP*, 2011, pp. 5040–5043.

[2] J. Pinto, G. S. V. S. Sivaram, M. Magimai-Doss, H. Hermansky, and H. Bourlard, “Analysis of MLP-based hierarchical phoneme posterior probability estimator,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 2, no. 19, pp. 225–241, Feb 2011.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[4] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny, “Deep belief networks using discriminative features for phone recognition,” in *Proc. ICASSP*, 2011, pp. 5060–5063.

[5] A. Mohamed, G. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan 2012.

[6] H. Jaeger, “The echo state approach to analysing and training recurrent neural networks - with an erratum note,” GMD Report 148, German National Research Center for Information Technology, Tech. Rep., 2001.

[7] D. Verstraeten, B. Schrauwen, M. D’Haene, and D. Stroobandt, “An experimental unification of reservoir computing methods,” *Neural Networks*, vol. 20, no. 3, pp. 391–403, Apr 2007.

[8] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, Aug 2009.

[9] M. Lukoševičius, H. Jaeger, and B. Schrauwen, “Reservoir computing trends,” *KI - Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, Nov 2012.

[10] F. Triefenbach, A. Jalalvand, K. Demuynck, and J.-P. Martens, “Acoustic modeling with hierarchical reservoirs,” *IEEE Trans. Audio, Speech, Language Process.*, accepted for publication 2013.

[11] F. Triefenbach and J.-P. Martens, “Can non-linear readout nodes enhance the performance of reservoir-based speech recognizers?” in *Proc. ICI*, 2011, pp. 262–267.

[12] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, “Phoneme recognition with large hierarchical reservoirs,” in *Proc. NIPS*, 2010, pp. 2307–2315.

[13] A. Jalalvand, F. Triefenbach, and J.-P. Martens, “Continuous digit recognition in noise: Reservoirs can do an excellent job,” in *Proc. Interspeech*, 2012, paper ID 644.

[14] A. Jalalvand, F. Triefenbach, D. Verstraeten, and J.-P. Martens, “Connected digit recognition by means of reservoir computing,” in *Proc. Interspeech*, 2011, pp. 1725–1728.

[15] T. Robinson and F. Fallside, “A recurrent error propagation network speech recognition system,” *Computer Speech & Language*, vol. 5, no. 3, pp. 259–274, Jul 1991.

[16] T. N. Sainath, B. Ramabhadran, and M. Picheny, “An exploration of large vocabulary tools for small vocabulary phonetic recognition,” in *Proc. ASRU workshop*, 2009, pp. 359–364.

[17] P. Clarkson and P. Moreno, “On the use of support vector machines for phonetic classification,” in *Proc. ICASSP*, 1999, pp. 585–588.

[18] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*, ser. The Kluwer International Series in Engineering and Computer Science. Springer US, 1994, vol. 247.

[19] M. Richard and R. Lippmann, “Neural network classifiers estimate posterior probabilities,” *Neural Computation*, vol. 3, no. 4, pp. 461–483, Winter 1991.

[20] W. J. Hardcastle and N. Hewlett, *Coarticulation: Theory, data and techniques*. Cambridge University Press, 1999.

[21] D. Jurafsky, W. Ward, Z. Banping, K. Herold, Y. Xiuyang, and Z. Sen, “What kind of pronunciation variation is hard for triphones to model?” in *Proc. ICASSP*, 2001, pp. 577–580.

[22] L. Bahl, P. De Souza, P. Gopalakrishnan, D. Nahamoo, and M. Picheny, “Context dependent modeling of phones in continuous speech using decision trees,” in *Proc. DARPA Speech and Natural Language Processing Workshop*, 1991, pp. 264–270.

[23] A. Abad, T. Pellegrini, I. Trancoso, and J. Neto, “Context dependent modelling approaches for hybrid speech recognizers,” in *Proc. Interspeech*, 2010, pp. 2950–2953.

[24] G. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pretrained deep neural networks for large-vocabulary speech recognition,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan 2012.

[25] S. Molau, F. Hilger, and H. Ney, “Feature space normalization in adverse acoustic conditions,” in *Proc. ICASSP*, 2003, pp. 656–659.

[26] R. Haeb-Umbach, “Investigations on inter-speaker variability in the feature space,” in *Proc. ICASSP*, 1999, pp. 397–400.

[27] K. Demuynck, J. Roelens, D. Van Compernolle, and P. Wambacq, “SPRAAK: An open source speech recognition and automatic annotation kit,” in *Proc. ICSLP*, 2008, p. 495.

[28] F. Triefenbach, K. Demuynck, and J.-P. Martens, “Improving large vocabulary continuous speech recognition by combining GMM-based and reservoir-based acoustic modeling,” in *Proc. SLT workshop*, 2012, 107-112.

[29] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, “The DARPA TIMIT acoustic-phonetic continuous speech corpus cd-rom,” NIST, Tech. Rep., 1993.

[30] K. F. Lee and H.-W. Hon, “Speaker-independent phone recognition using hidden markov models,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 11, pp. 1641–1648, Nov 1989.

[31] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[32] C. M. Bishop, “Training with noise is equivalent to tikhonov regularization,” *Neural Computation*, vol. 7, no. 1, pp. 108–116, Jan 1994.

[33] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proc. ICSLP*, 2002, pp. 901–904.

[34] R. Kneser and H. Ney, “Improved backing-off for m-gram language modeling,” in *Proc. ICASSP*, 1995, pp. 181–184.

[35] B. Hutchinson, L. Deng, and D. Yu, “A deep architecture with bilinear modeling of hidden representations: applications to phonetic recognition,” in *Proc. ICASSP*, 2012, pp. 4805–4808.