SPEAKER ADAPTATION OF CONTEXT DEPENDENT DEEP NEURAL NETWORKS

Hank Liao

Google Inc.
New York, NY, USA.

ABSTRACT

There has been little work on examining how deep neural networks may be adapted to speakers for improved speech recognition accuracy. Past work has examined using a discriminatively trained affine transformation of the input features applied at a frame level or the re-training of the entire shallow network for a specific speaker. This work explores how deep neural networks may be adapted to speakers by re-training the input layer, the output layer or the entire network. We look at how L2 regularization using weight decay to the speaker independent model improves generalization. Other training factors are examined including the role momentum plays and stochastic mini-batch versus batch training. While improvements are significant for smaller networks, the largest show little gain from adaptation on a large vocabulary mobile speech recognition task.

Index Terms— Large vocabulary continuous speech recognition, Multilayer perceptrons, Deep neural networks, Speaker adaptation

1. INTRODUCTION

In automatic speech recognition systems it is common to adapt a well-trained, general acoustic model to new users or environmental conditions. There are a variety of common techniques for Gaussian mixture model-based (GMM) acoustic models of speech and much research in this area [1]. Recently, the multilayer perceptron (MLP) has shown excellent results for modeling speech acoustics [2]. Since many more layers, for example 5 to 9 layers, are used than was typically explored in the past, this has been described as a deep neural network or DNN. With the recent trend towards adopting this acoustic model, it is worth investigating if and how DNNs can be adapted for new users or environments. Whether adaptation is even necessary is debatable since the larger networks have been shown to be invariant to some speaker effects, although they may still gain from some feature space transformations [3]. This paper examines how specific layers in a DNN acoustic model can be adapted directly for specific speakers, and how the size of the network and regularization during training affect supervised speaker enrollment and unsupervised speaker adaptation strategies.

The paper is organized as follows. First, commonly used speaker adaptation techniques for GMM-based acoustic models are discussed. Second, an overview of state-of-the-art DNN acoustic modeling is described along with a review of techniques for adapting neural network models. Experiments contrasting these techniques are reported next. Finally conclusions based on the experimental results are summarized.

2. GMM SYSTEM ADAPTATION

Many state-of-the-art speech recognition system use a hidden Markov model of speech with GMMs modeling the output context-dependent state distributions. To improve performance, they may use a variety of adaptation techniques that will be described briefly here. A Gaussian in the speaker independent acoustic model, containing all states, can be indexed by m, with mean and variance parameters denoted by $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$. The acoustic model can be estimated directly from the speaker data, but using the well-trained speaker independent model as a prior. This results in a maximum a posteriori update of the model mean [4], also know as MAP estimation

$$\hat{\boldsymbol{\mu}}_m = \frac{\tau \boldsymbol{\mu}_m + \sum_t^T \gamma_{m,t} \boldsymbol{o}_t}{\tau + T}$$

where T total adaptation frames, τ controls weight of prior data, and $\gamma_{m,t}$ is the posterior probability of Gaussian component m at time t. This approach deals with data sparsity by using the prior model information.

An alternative is to tie the model transformation across the many acoustic model parameters, which leads to the formulation for maximum likelihood linear regression, or MLLR [5]. A shared affine transformation is applied to the model means

$$\mathcal{N}(\boldsymbol{o}_t; \hat{\boldsymbol{\mu}}_{m,s}, \boldsymbol{\Sigma}_m) = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{A}_s \boldsymbol{\mu}_m + \boldsymbol{b}_s, \boldsymbol{\Sigma}_m) \quad (1)$$

to maximize the likelihood of observed data from the speaker given the model parameters, where $\boldsymbol{o}_t$ is a frame of speech from the speaker, $\boldsymbol{A}_s$ and $\boldsymbol{b}_s$ the matrix and bias of the affine transform for speaker s. This requires an update of every model mean in the acoustic model every time the speaker changes.

An alternative technique is called constrained MLLR (CMLLR) [1] which constrains a matrix in the transformation of the model means and variance. This can be manipulated into being a transform that is applied efficiently to the speech features, with the model parameters unchanged

$$\mathcal{N}(\boldsymbol{o}_t; \hat{\boldsymbol{\mu}}_{m,s}, \hat{\boldsymbol{\Sigma}}_{m,s}) = |\boldsymbol{A}_s| \mathcal{N}(\boldsymbol{A}_s \boldsymbol{o}_t + \boldsymbol{b}_s; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \quad (2)$$

Rather than using a single transformation per speaker, transforms can be estimated for similar Gaussians in the acoustic model by clustering them, e.g. using regression trees [6, 7]. This improves the power of MLLR and CMLLR transformations.

3. DEEP NEURAL NETWORK ACOUSTIC MODELS

The use of neural networks for acoustic modeling in speech recognition is not novel. The multilayer perceptron neural network was used for speech recognition in [8]. However the computation power over a decade ago limited their effectiveness. The recent successful

application of deep neural networks for acoustic modeling has been shown to be due to:

- *deep* networks of many layers,
- *wide* hidden layers of many nodes, and
- many *context dependent states* to model phonemes.

Graphics processing units have also enabled these deep networks to be trained in reasonable amounts of time. The power of DNNs over conventional GMMs for acoustic modeling in large vocabulary continuous speech recognition has been demonstrated in recent literature, where the number of hidden layers is between 5 and 9, with thousands of hidden nodes and context dependent output states [9, 10]. The DNN is also adept at de-correlating frames allowing a larger context window with many consecutively "stacked" frames of log filterbank features compared to mel frequency ceptral coefficients (MFCC) or perceptual linear prediction (PLP) features [11] (on the order of 26 frames compared to 11 respectively). While there is some debate whether pre-training [12], or other forms of neural network initialization [3] are needed, all continue to use back-propagation to fine-tune the DNN [13], that is using this weight update at time t

$$\Delta \mathbf{w}_t = -\epsilon \nabla_w E(\mathbf{w}_t) \tag{3}$$

where ∇_w represents the gradient operator with respect to the weight vector $\mathbf{w}$, $E(\mathbf{w})$ the error function and ϵ the learning rate. It has been found that using momentum [14] can speed up the training process by adding common contributions from previous updates to the gradient update in a second term

$$\Delta \mathbf{w}_t = -\epsilon \nabla_w E(\mathbf{w}_t) + \alpha \Delta \mathbf{w}_{t-1} \tag{4}$$

For example, with α set to 0.9, constant parts of the gradient are amplified by $1/(1-\alpha)$ or 10, while parts that oscillate are smoothed out over time.

To apply the neural network in acoustic modeling, the CD state emission likelihood is computed from the CD state posterior generated by the DNN as follows using Bayes' Rule

$$p(\mathbf{o}|s;\theta) \approx \frac{P(s|\mathbf{o};\theta)}{P(s;\theta)} \tag{5}$$

where $\mathbf{o}$ denotes the feaures, s a CD state, and $P(s)$ the state prior. The approximation would be an equality if the right hand side was multiplied by $p(\mathbf{o})$.

The first DNN deployed in production by Google [15] has the topology shown in figure 1. The number of parameter comparisons is shown for a GMM based system that had approximately the same real time speed.

3.1. Adaptation of Neural Networks

Previous work in [16] looked at speaker adaption of shallow neural networks and with context independent units. The networks examined were small: 9 stacked frames yielding 234 inputs, one hidden layer of 1000 units, and 48 outputs one for each context independent phone. Adaptation was performed by either estimating a normalization affine transform applied to each frame trained via back-propagation, re-training the entire network, or both combined. Significant gains were achieved using these techniques. Later this was referred to as feature-space discriminative linear regression and applied to a large vocbaulary task with a DNN [3]; the improvement on a 45M parameter network (16.9M non-zero) was less at 4% relative.

This paper is motivated by this previous work to examine how re-training only portions of the network and the size of the network affects speaker adaptation performance. This paper will also examine how regularization may be used to improve generalization. This can be done by adding half of a squared penalty term that to the error function to minimize the difference between the updated weight and unadapted network weight, which shall be referred to as L2 prior regularization. Here the weight update in equation 4 then becomes

$$\Delta \mathbf{w}_t = -\epsilon \nabla_w E(\mathbf{w}_t) + \alpha \Delta \mathbf{w}_{t-1} - \beta(\mathbf{w}_{t-1} - \mathbf{w}_0) \tag{6}$$

where β is the weight decay factor on the L2 penalty term which decays the weights towards the original model weights. The larger the penalty term, the more difficult it is for the updated weights to deviate from the original model weights $\mathbf{w}_0$. This is similar to what is described as MAP adaptation of a maximum entropy model in [17]. In [18], the solution to overfitting the constrained transform for adapting the first layer in a network to a speaker can also be viewed as L2 regularization.

This paper will also examine various aspects of the training procedure including the optimization hyperparameters such as learning rate and momentum, supervised enrollment versus unsupervised training, how the amount of data affects gains and stochastic mini-batch vs batch training.

4. EXPERIMENTAL RESULTS

The experiments are conducted on proprietary, anonymized mobile search data sets. Utterances are typically short at about 3-10 seconds in duration. The training set is approximately 3000 hours of mobile speech data. Two test sets were used, the first `Unified1a` is an uniform sampling of 30 hours of mobile search data which includes VoiceSearch queries and VoiceIME dictation. The second is sampled from users who have opted to donate their speech data [19]. This allows the construction of an anonymized data set of 80 speakers, each with about an hour of adaptation data to form `Pers1a_adapt` set and ten minutes of evaluation data to form an evaluation set `Pers1a_eval`.

The vocabulary of the recognition system is approximately one million words. The language model is a Katz smoothed language model trained on both the transcribed acoustic training data, written query sources and unsupervised mobile speech data. A variety of acoustic models are evaluated. The speaker independent, real-time GMM system uses PLP features [20], semi-tied covariances [21] and linear discriminant analysis for dimensionality reduction of

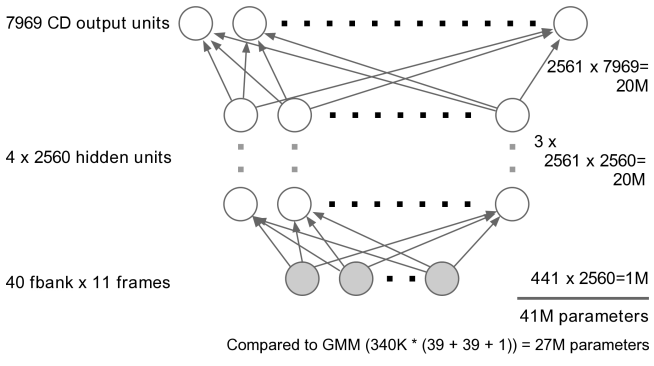

Fig. 1. Mobile speech recognition deep neural network.

the 9 consecutively stacked PLP features down to 39 dimensions, and boosted MMI discriminative training [22] of context-dependent states [23] clustered using decision trees [24] to 7969 states; the real time GMM system contained a total of 340k Gaussians, while a larger 500k Gaussian was also available. The deployed DNN system uses 11 stacked 40-dimensional log filterbank feature frames, 4 hidden layers of 2560 nodes each, and 7969 output nodes corresponding to the same context dependent state inventory. A much smaller DNN was trained with 16 stacked frames, 4 hidden layers of 512 nodes each and 1000 outputs. Also a larger DNN with 26 stacked frames, 6 hidden layers of 2176 nodes and 14247 outputs was trained to be more comparable to the results in [3].

4.1. Experiments on `Unified1a`

First some experiments are done on the `Unified1a` test set. Table 1 compares the relative performance of the various acoustic models discussed above. The results demonstrate that for a similar number of parameters (41M) the DNN is significantly better than the GMM system. The DNN system is also effective at smaller and larger network sizes, whereas the GMM system degrades badly at 100k Gaussians and doesn't improve much by increasing the number of parameters beyond 580k Gaussians. However on an embedded device the smallest 100k Gaussian GMM system was able to run in realtime on a recent mobile device, whereas the small 4x512 DNN is slightly slower on the same device.

Model	# of params	`Unified1a`
GMM, 100k Gaussian	7.5M	20.6
GMM, 340k Gaussian	27M	16.0
GMM, 580k Gaussian	46M	15.4
DNN, 4x512+1000	1.6M	16.1
DNN, 4x2560+7969	41M	12.3
DNN, 6x2176+14000	60M	9.9

Table 1. Comparison of unadapted GMM and DNN acoustic models, number of parameters and % WER.

4.2. Experiments on `Pers1a`

Some comparisons can be made between a GMM system and a DNN. We choose to compare a slightly smaller GMM system (27M parameters) with the 41M parameter DNN system since it is only slightly worse than the bigger system and from experience adaptation of GMM based systems tends to work better when the models are smaller. Table 2 compares various GMM system adaptation techniques with the 4x2560 hidden layer DNN. CMLLR and MLLR give similar gains and improve as expected with increased number of transforms; at 256 transforms the relative reduction in word error rate (WER) is 12-14%. MAP adaptation does surprisingly poorly, yielding the equivalent gain of 1 MLLR transform; perhaps this is due to the errors in the adaptation hypothesis. The unadapted DNN performs better than any of the adapted GMM systems; although it is larger, it is unlikely that a larger adapted GMM system would be better than the unadapted DNN system since the absolute difference on `Unified1a` between the smaller and bigger GMM system was 0.6%.

System	Speaker Adaptation	# of Transforms (%WER) —	2	16	256
GMM	—	18.3			
	CMLLR		16.7	16.1	16.1
	MLLR		16.7	16.1	15.7
	MAP, τ=1.0	16.7			
DNN	—	14.1			

Table 2. Comparison of unsupervised speaker adaptation, using 1 hour of adaptation data, of 340k Gaussian GMM acoustic model with 4x2560 hidden layer DNN on `Pers1a` (%WER).

Neural Network Adaptation

The DNN can be adapted as suggested previously. Instead of applying a discriminatively trained transform of the features, in this work for speaker adaptation we look at only re-training the input layer, the output layer or the entire network. When training and re-training, for adaptation, the neural networks a default step size of 0.02 was used, along with momentum set to 0.9, weight decay set to 0.01 and a mini-batch size of 200 frames. First some initial experimentation was done with one speaker to test some of the hyperparameters settings. A single speaker has about 1500 words for adaptation and 1500 words for the held-out evaluation testing. Table 3 shows the difference between using momentum or not during adaptation.

Adapt Style	α	ϵ	`Adapt` (Epochs) 1	10	100	`Eval` (Epochs) 1	10	100
—			11.7			15.0		
Mini-Batch	0.9	0.02	7.6	6.8	5.9	11.7	11.6	11.9
	0.0	0.2	7.9	7.0	6.4	12.0	11.4	11.5
Batch	0.9	0.02	11.3	11.0	8.2	14.3	13.5	11.9
	0.0	0.2	11.5	10.5	8.2	14.3	13.3	12.0

Table 3. Comparing effect of momentum (α) for adapting a 4x512 hidden layer DNN to one speaker (% WER): `Adapt` is a ten minute portion of the adaptation data, whereas `Eval` is the held out test. Mini-batch training used momentum, batch training did not.

Compared to multiplying the step size by ten to 0.2, momentum was found to give similar results but nevertheless applied for consistency with the original neural network training regime. We also found stochastic mini-batch to converge faster than batch updates, e.g. within 10 epochs versus 100-1000 epochs in a batch training scenario. With mini-batch training and without regularization, the error rates could be reduced by more than half on the adaptation data, but didn't generalize on the held out evaluation data; using L2 prior regularization improved results as shown in table 4. The L2 prior regularization was found to be not useful with batch training,

Adapt Style	β	`Adapt` (Epochs) 1	10	100	`Eval` (Epochs) 1	10	100
—		11.7			15.0		
Mini-batch	0.01	7.6	6.8	5.9	11.7	11.6	11.9
	0	7.2	4.8	2.8	11.9	12.5	14.8
Batch	0.01	11.5	10.5	8.2	14.3	13.3	12.0
	0	11.5	10.4	8.3	14.3	13.3	12.2

Table 4. Comparing using L2 prior regularization ($\beta :=$ weight decay) for adapting a 4x512 hidden layer DNN to one speaker (% WER): `Adapt` is a ten minute portion of the adaptation data, whereas `Eval` is the held out test. Mini-batch training used momentum, batch training did not.

probably due to the smoother gradients. Further network adaptation results all use L2 prior regularization.

Enrollment-style network adaptation

One approach to speaker adaptation is to have speakers read some material where the transcript is known and use this labeled data to fine-tune the network. This is often called speaker enrollment. In this work though we do not have speakers read data, rather we use the labeled results for 10 minutes of the adaptation data as the enrollment data for supervised training. Table 5 shows the results for this type of speaker adaptation where the number of epochs is shown and different parts of the DNN are adapted. Note that while labeled data is used to adapt the network to each speaker, the results are on held out evaluation data.

	# Epochs (%WER)		
Adaptation	1	10	100
—		17.1	
Input layer	15.7	15.4	15.2
Output layer	16.4	16.0	15.9
All layers	14.4	14.2	14.4

Table 5. Enrollment style speaker adaptation of a 4x512 hidden layer DNN evaluated on `Pers1a` (%WER).

Although there are more output parameters, only updating the input layer gives better results than only updating the output layer. Even better is updating all 1.6M parameters in the network. Surprisingly, after 1 epoch most of the gains for adaptation are achieved. Only updating a single layer gives a relative improvement of around 10% whereas the entire network more than 15%. These are similar to the gains found when adapting GMM acoustic models with common techniques.

Unsupervised network adaptation

The previous section looked at adaptation when the labels for the adaptation data were known. Whether the adaptation improvements hold when the labels are unknown is not certain. In this section, results are obtained by adapting the network per speaker with labels determined from the large 6x2176 deep DNN, i.e. with a WER of 10.2% on the 10 minutes of adaptation data. These results are shown in table 6. The upper three rows demonstrate large gains of

		# Epochs (%WER)		
Test set	Adaptation	1	10	100
`Adapt10min`	—		16.5	
	Input layer	14.6	13.8	13.5
	Output layer	15.5	14.5	14.0
	All layers	12.5	11.5	11.2
`Eval`	—		17.1	
	Input layer	15.8	15.5	15.3
	Output layer	16.5	16.2	16.0
	All layers	14.7	14.5	14.5

Table 6. Unsupervised speaker adaptation of a 4x512 hidden layer DNN evaluated on `Pers1a` (%WER).

6-30% relative when evaluating on the same unsupervised adaptation data. Thus this approach could be used for offline systems that use multi-pass decoding strategies. In the bottom three rows, the unsupervised enrollment adaptation shows less improvement than with the speaker enrollment in table 5, but the adaptation of the entire network still shows about 15% relative gain indicating an incremental unsupervised adaptation strategy could work well.

Adaptation of a large network

The previous DNN adaptation results demonstrated that speaker adaptation could work well for a small network. Table 7 describes results of unsupervised adaptation of a large DNN. Here the results

		# Epochs (%WER)	
Test set	Adaptation	1	10
`Adapt10min`	—	10.2	
	Input layer	10.4	10.0
	Output layer	10.2	10.0
	All layers	9.8	10.0
`Eval`	—	10.8	
	Input layer	11.0	10.9
	Output layer	11.0	10.9
	All layers	10.2	10.3

Table 7. Unsupervised speaker adaptation of a 6x2176 hidden layer DNN evaluated on `Pers1a` (%WER).

are more mixed. While results with supervised data show improvements, unsupervised adaptation of the large neural network shows no gain for adapting just the input or output layer. There is a small improvement from adapting the entire network, of 4.9% relative, but storing an entire network of 60M parameters per speaker is unwieldy. This minimal gain is slightly more than the 4% relative found by using a speaker-level discriminative transform with a DNN in [3] on the Fisher task; the discriminative transform is much more compact and can be estimated more rapidly. The minimal gain can be attributed to the large number of layers in the network that perform a powerful speaker normalizing feature extraction [11].

5. CONCLUSIONS

This paper compares some standard speaker adapted GMM systems with adaptation of deep neural networks. As shown in prior work an unadapted DNN system is handily better than an adapted GMM-based acoustic model. In this work we show that L2 prior regularization is helpful in improving generalization when adapting neural networks to speaker specific data. While momentum wasn't found to be appreciably useful, mini-batch training converged considerably faster than batch and yielded slightly better results. It is also shown that small networks can benefit from both supervised and unsupervised speaker adaptation. Similar to previous work, large neural networks do not benefit as much from adaptation techniques.

6. REFERENCES

[1] M.J.F. Gales, "Maximum Likelihood Linear Transformations For HMM-Based Speech Recognition," *Computer Speech and Language*, vol. 12, Jan. 1998.

[2] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent dnn-hmms for real-world speech recognition," in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Dec. 2010.

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011.

[4] J.L. Gauvain and C.H. Lee, "Bayesian learning of gaussian mixture densities for hidden Markove models," in *Proc. DARPA Speech and Natural Language Workshop*, 1991.

[5] C.J. Legetter and P.C. Woodland, "Maximum likelihood linear regression speaker adaptation of contiuous density HMMs," *Computer Speech and Languages*, 1997.

[6] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous control using tree structure," in *Proc. Eurospeech*, 1995.

[7] M.J.F. Gales, "The generation and the use of regression class trees for MLLR adaptation," Tech. Rep. CUED/F-INFENG/TR263, University of Cambridge, 1996, Available from http://mi.eng.cam.ac.uk/reports/index-speech.html.

[8] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Hybrid neural network/hidden Markov model continuous-speech recognition," in *Proc. Eurospeech*, 1992.

[9] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011.

[10] T. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. ASRU*, 2011.

[11] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. ICASSP*, 2012.

[12] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Acoustics, Speech, and Language Processing*, 2012.

[13] D. Rumelhart, G. Hinton, and R. Williams, "Learning representation by back-propagating errors," *Nature*, vol. 323, Oct. 1986.

[14] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation," *Parallel distributed processing*, vol. 1, 1986.

[15] N. Jaitly, P. Nguyen, A.W. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. Interspeech*, 2012.

[16] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Proc. Eurospeech*, 1995.

[17] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," in *Proc. ACL*, 2004.

[18] J. Trmal, J. Zelinka, and L. M:uller, "Adaptation of a feedforward artificial neural network using a linear transform," in *Proc. TSD*, 2010.

[19] A. Mané and G. Shires, "Voice search gets personal," Dec. 2010, http://googlemobile.blogspot.com/2010/12/voice-search-gets-personal.html.

[20] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *Journal of the Acoustic Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[21] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. on Speech and Audio Processing*, May 1999.

[22] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. ICASSP*, 2008.

[23] L.R. Bahl, P.V. de Souza, P.S. Gopalkrishnan, D. Nahamoo, and M.A. Picheny, "Context dependent modelling of phones in continuous speech using decision trees," in *Proc. DARPA Speech and Natural Language Processing Workshop*, 1991, pp. 264–270.

[24] S.J. Young, J.J. Odell, and P.C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.