
Proceedings Paper:
Liu, Y., Karanasou, P. and Hain, T. (2015) An Investigation into Speaker Informed DNN Front-end for LVCSR. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 19-24 April 2015, Brisbane, Australia. IEEE Conference Publications . IEEE , IEEE Xplore . ISBN 978-1-4673-6997-8/15

https://doi.org/10.1109/ICASSP.2015.7178782

AN INVESTIGATION INTO SPEAKER INFORMED DNN FRONT-END FOR LVCSR

Yulan Liu[1], Penny Karanasou[2], Thomas Hain[1]

[1]Speech and Hearing Research Group, The University of Sheffield, UK
[2]Speech Research Group of the Machine Intelligence Laboratory, University of Cambridge, UK
acp12yl@sheffield.ac.uk, p.karanasou@eng.cam.ac.uk, t.hain@dcs.shef.ac.uk

ABSTRACT

Deep Neural Network (DNN) has become a standard method in many ASR tasks. Recently there is considerable interest in "informed training" of DNNs, where DNN input is augmented with auxiliary codes, such as i-vectors, speaker codes, speaker separation bottleneck (SSBN) features, etc. This paper compares different speaker informed DNN training methods in LVCSR task. We discuss mathematical equivalence between speaker informed DNN training and "bias adaptation" which uses speaker dependent biases, and give detailed analysis on influential factors such as dimension, discrimination and stability of auxiliary codes. The analysis is supported by experiments on a meeting recognition task using bottleneck feature based system. Results show that i-vector based adaptation is also effective in bottleneck feature based system (not just hybrid systems). However all tested methods show poor generalisation to unseen speakers. We introduce a system based on speaker classification followed by speaker adaptation of biases, which yields equivalent performance to an i-vector based system with 10.4% relative improvement over baseline on seen speakers. The new approach can serve as a fast alternative especially for short utterances.

Index Terms— speech recognition, deep neural network, speaker adaptation, speaker informed training, bias adaptation

1. INTRODUCTION

DNN based ASR systems have been shown to consistently give best results in both DNN-HMM-GMM [1, 2] and DNN-HMM hybrid structures [3, 4]. Recently there has been considerable interest in adapting speaker independent DNN based systems to particular speakers. Related research mainly falls into four categories. The first category performs speaker normalisation at signal level, such as Vocal Tract Length Normalisation (VTLN [1]), or speaker transformation at feature level, such as feature-MLLR (fMLLR [3]). The second category includes speaker dependent discriminative transformations into DNN structures, for example Linear Input Network (LIN [5]), Linear Output Network (LON [6]), Linear Hidden Layer (LHN [7]) and feature-space Discriminative Linear Regression (fDLR [3]). The third category, "informed DNN training", informs DNNs with meta-information during training process by augmenting the DNN input with auxiliary codes that carry speaker information. Examples of auxiliary codes are eigenvectors in speaker space [8], speaker codes [9], i-vectors [10] and Speaker Separation BottleNeck features (SSBN [11] or speaker d-vectors). The fourth category splits DNNs into speaker independent part and speaker dependent part (output layer [12], or bottleneck layer [13]), or boosts some neurons while penalizes others depending on speaker [14].

Work in this paper contributes in several novel aspects: the mathematical implications of speaker informed DNN training is assessed, followed by a quantitative comparison of several speaker adaptation techniques which also gives first performance evidence of i-vector based adaptation over bottleneck (BN) features [1]; generalisation to unknown speakers is discussed; and a new system for speaker adaptation in short utterances is proposed.

This paper first focuses on mathematical equivalence and difference among different informed training methods. It is complex to track the genuine contribution of each DNN parameter, due to a mixture of linear and non-linear functions and the high redundancy and symmetry in parameter space that allow many equivalent parametric solutions. We simplify the problem by focusing on the most affected part, the input layer. Speaker informed DNN training is shown to be mathematically equivalent to DNN input layer bias adaptation (§3) which employs speaker dependent biases.

The performance of different informed training methods based on i-vectors, SSBN features, speaker separation DNN (SSDNN) posteriors and hand-crafted codes is compared for the first time. The performance difference is shown to relate to the dimension (§3.1), discrimination (§3.2), and numerical and temporal stability (§3.3) of auxiliary codes. I-vector based speaker informed training implemented on DNN front-end system shows a 10.4% relative performance improvement over speaker independent systems on seen speakers, identical to that obtained with bias adaptation (§5).

Test speakers that have already been observed in training are referred to as "seen speakers", otherwise "unseen speakers". Considerable performance difference on seen speakers is observed among different informed training methods on meeting recognition task. While for seen speakers DNNs can "remember" a specific optimal setting, for unseen speaker DNNs require a sense of proximity to observed speakers (§3.4). In our experiments all tested methods show poor generalisation to unseen speakers for different reasons.

Based on the findings, an alternative approach to fast speaker adaptation of DNN front-ends is proposed using Unique Binary Index Codes (UBIC). By first identifying speakers, equivalent performance to i-vector based informed training is obtained over seen speakers. While i-vector estimation requires sufficient data to be stable and accurate [15], the proposed system is effective on utterance of 5s on average (§5), hence a fast alternative.

2. BACKGROUND

2.1. Informed DNN training

A standard N-layered feed-forward DNN has M_n neurons in the n-th layer ($n\in[1, 2, ..., N]$). The input to n-th layer is denoted as $\mathbf{x}_n(t) = [x_{n,1}(t), x_{n,2}(t), ..., x_{n,M_n}(t)]^T$ with $\mathbf{x}_1(t)$ referring to DNN input features, which are naturally time dependent. With the activation function on all hidden neurons $f(\cdot)$, the output of first layer is given by:

$$x_{2,k}(t) = f\Big(\sum_{m=1}^{M_1} x_{1,m}(t)w_{1,m,k} + b_{1,k}\Big) \quad (k\in[1, 2, ..., M_2]) \quad (1)$$

where $w_{1,m,k}$ is weight and $b_{1,k}$ is bias, related to the m-th dimension in the input and the k-th input to the second layer.

For informed training, DNN input is augmented with an L dimensional time dependent vector $\mathbf{c}(t) = [c_1(t), c_2(t), ..., c_L(t)]^T$, which can be eigenvectors [8], i-vectors [10], SSBN features [11], speaker codes [9], etc. Then

$$x'_{2,k}(t) = f\Big(\sum_{m=1}^{M_1} x_{1,m}(t) w'_{1,m,k} + \sum_{l=1}^{L} c_l(t) h'_{l,k} + b'_{1,k} \Big), \quad (2)$$

where $h'_{l,k}$ is weight applied on the l-th dimension of codes for the k-th input to the second layer. While the codes can be time dependent, they are assumed to be noisy variants of a single centroid.

2.2. I-vectors

I-vectors are motivated by Joint Factor Analysis (JFA, [16]), and were originally proposed for speaker recognition [17]. An i-vector represents the specific characteristics of a speaker as a point in total variability space. Recently they are also used for speaker adaptation of speech recognition systems, for both the conventional HMM-GMM systems [18, 19] and the DNN-HMM hybrid systems [10, 20].

A Universal Background Model (UBM) is first built to represent the feature space. The mean vectors of all GMMs in this UBM are concatenated into a super-vector $\boldsymbol{\mu}_0$. Correspondingly, a set of speaker-dependent GMMs is derived for each speaker, and its mean vectors are concatenated into a speaker dependent super-vector, i.e. $\boldsymbol{\mu}^s$ for speaker s. The total variability matrix $\mathbf{M}$ spans the bases with highest variability in the mean super-vector space. Given the i-vector for speaker s as $\boldsymbol{\lambda}^s$, we obtain [17, 20]

$$\boldsymbol{\mu}^s = \boldsymbol{\mu}_0 + \mathbf{M}\boldsymbol{\lambda}^s. \quad (3)$$

3. INFORMED DNN TRAINING AND BIAS ADAPTATION

Based on Eq.(2), the effective overall bias in informed training is

$$\beta_k(t) = \sum_{l=1}^{L} c_l(t) h'_{l,k} + b'_{1,k} \quad (k \in [1, 2, ..., M_2]). \quad (4)$$

An informed DNN as expressed in Eq.(2), can be equivalent to standard DNNs represented by Eq.(1) when the overall biases equal. If auxiliary code is fixed per speaker (as in [10]), informed training is equivalent to speaker level bias adaptation. In this way auxiliary codes help to build an implicit pool of speaker dependent biases based on whole training data. It is similar when auxiliary code is fixed per utterance or per cluster of utterances, while the size of such a "bias pool" can differ. The frame-wise auxiliary code as used in [11] can have strong time variation, however it can be interpreted as being centred around a mean (to be discussed in §3.3).

Although all mathematically equivalent to bias adaptation, different practical implementations of informed training yield different performance [10, 11, 20], even when their bias pools have the same size. This is mainly due to the difference in dimension, discrimination and variability of auxiliary codes, as will be discussed below.

3.1. Code dimension and number of speakers

For auxiliary codes fixed at speaker level $\mathbf{c}^s = [c_1^s, c_2^s, ..., c_L^s]^T$, assume the optimal bias on k-th neuron for speaker s is $\hat{b}_k^s$ (S speakers in total). Thus when the effective overall biases equal the optimal biases,

$$\begin{cases} \sum_{l=1}^{L} c_l^1 h'_{l,k} + b'_{1,k} = \hat{b}_k^1 \\ \sum_{l=1}^{L} c_l^2 h'_{l,k} + b'_{1,k} = \hat{b}_k^2 \\ \cdots \\ \sum_{l=1}^{L} c_l^S h'_{l,k} + b'_{1,k} = \hat{b}_k^S \end{cases}, \quad (5)$$

which can be written as $\mathbf{C}\mathbf{h}'_k = \hat{\mathbf{b}}_k - \mathbf{b}'_{1,k}$, or

$$\begin{pmatrix} c_1^1 & c_2^1 & \cdots & c_L^1 \\ c_1^2 & c_2^2 & \cdots & c_L^2 \\ \vdots & \vdots & \ddots & \vdots \\ c_1^S & c_2^S & \cdots & c_L^S \end{pmatrix} \begin{pmatrix} h'_{1,k} \\ h'_{2,k} \\ \vdots \\ h'_{L,k} \end{pmatrix} = \begin{pmatrix} \hat{b}_k^1 - b'_{1,k} \\ \hat{b}_k^2 - b'_{1,k} \\ \vdots \\ \hat{b}_k^S - b'_{1,k} \end{pmatrix}. \quad (6)$$

If $\hat{b}_k^s$ is distinct for each speaker: when $L > S$, there exists an infinite number of solutions for $\mathbf{h}'_k$; when L=S, if code matrix $\mathbf{C}$ is invertible there exits one set of solution, if $\mathbf{C}$ is not invertible there exists no accurate solutions unless $(\hat{b}_k^s - b'_{1,k})$ and $\mathbf{c}^s$ have the same linear dependence among different speakers; when $L < S$, there exist no accurate solutions. In the cases without accurate solutions, there can be solutions yielding minimal errors. With more solutions for $\mathbf{h}'_k$, informed DNN training is more likely to converge to one solution yielding optimal or approximately optimal biases in practice. In contrast if different speakers have the same or very similar optimal biases, i.e. $\hat{b}_k^i$=$\hat{b}_k^j$ or $\hat{b}_k^i \approx \hat{b}_k^j$ ($\exists i \neq j$), the necessary code dimension can be reduced as speaker i and j can be clustered.

To avoid confusion, a test speaker that has been observed in training data is referred to as a "seen speaker", otherwise an "unseen speaker". For generality, we call it an "informed condition" for any test speaker if its effective overall biases approximately equal the optimal biases, with possible deviation that does not cause significant performance degradation to speech recognition systems. Otherwise it is an "un-informed condition". Ideally the auxiliary code for an seen speaker during test can be the same one used in training, i.e. the code for an seen speaker is one part of code matrix $\mathbf{C}$. Since DNNs are optimized based on Eq.(6), all training speakers are informed. Thus for seen speakers, informed training can theoretically perform as well as optimal bias adaptation, if auxiliary codes are chosen rationally and if training is optimized properly.

For an unseen speaker s_u, according to Eq.(4) the effective overall bias $\beta_k^{s_u} = (\mathbf{c}^{s_u})^T \mathbf{h}'_k + b'_{1,k}$ is determined by auxiliary codes $\mathbf{c}^{s_u}$ trained parameters $\mathbf{h}'_k$ and $b'_{1,k}$. Even though parameters $\mathbf{h}'_k$ are learned from auxiliary codes which can rationally represent a space of training speakers via $\mathbf{C}$ (like i-vectors), the implicit bias pool built during training is not always diverse enough to predict the optimal biases for all unseen speakers. Thus more training speakers are usually preferred to fewer, with a necessary increase in optimal auxiliary code dimension. If the amount of training speaker is not large enough, or if the auxiliary code is not designed rationally and estimated accurately, it is uncertain how much the effective overall biases of an unseen speaker would deviate from their optimum in practice. This can lead to performance degradation.

3.2. Code separation

As shown in Eq.(6), reduced separation among codes (i.e. higher linear dependence, lower discrimination) will lead to a higher condition number for $\mathbf{C}$. This will increase numerical instability and cause deviation from the optimal solutions in real numerical optimization. Speaker discrimination in code space is also related to speaker classification performance based on those codes. Thus one would expect a positive correlation between discrimination and informativeness in codes, if other conditions are kept unchanged. In a special case where the codes are designed to provide ideal speaker discrimination under informed conditions, the codes matrix can be an identity matrix, and $h'_{l,k} = \beta_k^l - b'_{1,k}$. That special case assumes orthogonal basis vectors for the auxiliary codes, and is further referred to as Unique Binary Index Codes (UBIC).

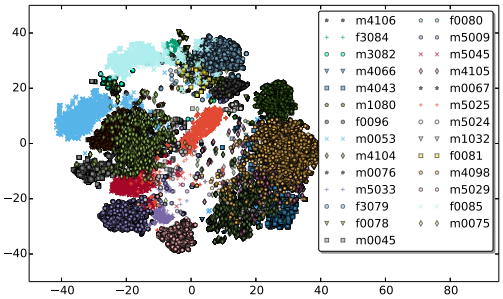

Fig. 1. 2D visualization of 13 dimensional SSBN features from 27 speakers in meeting test data using BH-tSNE.

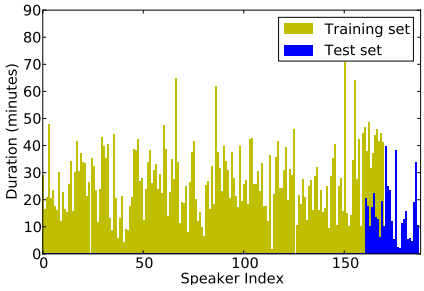

Fig. 2. Duration of the data per speaker. Average data duration per speaker: 27.3mins in training set, 15.4mins in test set.

3.3. Numerical and temporal stability

For Speaker Aware DNNs (SADNNs) [11], the auxiliary codes $\mathbf{c}(t)$ are estimated per frame. As a result the overall bias in the input layer $\beta_k(t)$ is time-variant. Changing codes inevitably introduce temporal noise to Eq.(6) in speaker informed training. Figure 1 shows a 2D representation of 13 dimensional SSBN features from 10 seen and 17 unseen speakers using Barnes-Hut t-distributed Stochastic Neighbor Embedding (BH-tSNE [21, 22]). One can clearly observe noise around speaker centroids. The numerical condition of the code matrix $\mathbf{C}$ largely depends on the noise level. During test, auxiliary codes estimated on frame level also vary more or less around the centroids. Thus informed training is more robust when using auxiliary codes estimated globally rather than locally. The advantage of reliable global codes will be more pronounced if code estimation is sensitive to noise, utterance duration, speech and silence ratio, etc.

3.4. Uncertainty under un-informed conditions

Since DNNs are highly symmetric in structure, optimal biases $\hat{\mathbf{b}}_k$ might have infinite parametric solutions all providing the same overall performance. The value $\hat{\mathbf{b}}_k$ takes will depend on the value of other DNN parameters. Thus the optimal biases trained in one system cannot be easily transplanted to another DNN even for the same speaker. As a result, it is uncertain whether a code would work or not on a system not trained with the code. Informed training using i-vectors was shown to be effective for unseen speakers in [10, 20]. That is because unseen speakers are informed in the way that their i-vectors fall into the speaker space built with training speaker i-vectors of sufficient speaker diversity (as discussed in §3.1). However considerable computation and data resources are necessary to build up such an i-vector space, while the reliability and effectiveness during test would reduce with shorter utterances [15].

4. EXPERIMENTS

4.1. Data

The individual headset recordings from the AMI corpus [23] are used for experiments. The training set is composed of 77.5h speech from 170 speakers in 148 meetings. The test set includes 6.9h from 27 speakers in 20 meetings, in which 4.4 hours are from 17 unseen speakers and the rest 2.5 hour data are from 10 seen speakers. No meetings are shared between training and test sets. Average utterance duration is 4.2s in training set, and 5s in test set. Figure 2 shows the amount of data per speaker, on training and test sets.

4.2. Baseline

In all experiments DNNs are implemented using TNet[1], and Viterbi decoding is performed with the AMI RT09 trigram language model and dictionary [24]. In the baseline system, 23 dimensional log Mel-filterbank features with a context of 31 frames (+/-15) are compressed to 368 dimensions using DCT. The mean and variance are normalised globally over training data in each dimension. The normalised features are used to train a six layered DNN in a topology of 368:1745:1745:1745:26:1257. The 1257 output targets are tied triphone states and the training is done layer by layer. Linear bottleneck (BN) features are extracted from the 26-dimensional bottleneck layer. HMM-GMMs are trained over BN features with Single Pass Retraining (SPR) from canonical PLP based HMM-GMMs in the same manner with [11]. The BN features based HMM-GMMs are re-clustered to around 4000 states (16 Gaussian mixtures each) and optimised with maximal likelihood criterion.

4.3. SADNN based features

SADNN [11] augments DNN input with SSBN features extracted from a 5 layered Speaker Separation DNN (SSDNN). The SSDNN is a DNN in a topology of 368:1745:1745:L_{BN}:171 trained to classify speakers. It uses 170 training speaker IDs plus a silence as targets. The performance for $L_{\text{BN}} = 13, 40, 60, 80, 100$ is investigated.

Raw posteriors auxiliary codes provide better speaker discrimination than SSBN features, and hence are also investigated. In addition, the posteriors approximate UBIC and speaker dependent bias adaptation when SSDNN gives perfect speaker classification results, except for the dimension corresponding to silence.

4.4. I-vectors

For training set one i-vector is first estimated per speaker using all the data of that particular speaker. To train i-vectors, the approach presented in [20] is followed, where i-vectors are represented as the weights of a CAT system [25]. Since the underlying models are GMMs, it allows unsupervised training without the necessity of transcriptions. To estimate i-vectors for test set, i-vector weights are updated over test data per speaker using the model parameters learned during i-vector training. Experiments compare the performance using different i-vector dimensions, i.e. $L_{\text{iv}} = 13, 40, 60, 80, 100$. Before concatenating with log Mel-filterbank features, i-vectors are normalised globally over training data in order to have zero mean and unit variance in each dimension.

4.5. Hand-crafted binary codes

To compare the effectiveness of different informed training methods and the differences in performance due to codes design, three types of auxiliary codes are hand-crafted. The first type is an 8 dimensional binary code, derived from an 8-bit binary index of 188 speakers (sorted by spelling) from both training and test data. The second type is a 188 dimensional UBIC. The third type uses a 170 dimensional UBIC for the speakers in the training set and seen speakers in test set, while uses zero vectors for unseen speakers.

[1] http://speech.fit.vutbr.cz/software/neural-network-trainer-tnet

Table 1. Speaker classification performance comparison: SSDNN using different bottleneck dimensions.

SSBN dim	13	40	60	80	100
Acc (%)	92.96	94.03	**95.61**	95.10	94.05

4.6. Combining SSDNN with UBIC for informed training

Though hand-crafted codes like UBIC can provide ideal speaker discrimination without the computation cost in codes estimation, they highly depend on prior knowledge of the speaker identity, which is not always available. This could be solved by using utterance level speaker classification results, for example from an independently trained SSDNN. During test, the auxiliary UBIC corresponding to the speaker candidate with maximal average log posterior over that utterance is used. Posteriors corresponding to silence are ignored in this speaker classification. Table 1 shows segmental speaker classification accuracies, i.e. the ratio between the number of utterances with speaker ID correctly recognised and the total number of utterances from seen speakers in test set (1775). The highest accuracy is observed with a 60 dimensional BN layer. Using a larger BN dimension decreases the accuracy because the SSDNN starts to overfit to the silence target. The results indicate that the SSDNN speaker classification is a good candidate to estimate seen speaker IDs in a combined SSDNN-UBIC framework.

5. RESULTS

Table 2 compares the performance of methods described in sections §4.2–4.6. As shown, all informed training improves the speech recognition performance over baseline for seen speakers (observed in both training and test sets) while degrades for unseen speakers (observed only in test set). I-vectors based method degrades over unseen speakers due to insufficient speaker diversity in i-vector training. All hand-crafted codes fail to give any improvement over unseen speakers due to a total absence of speaker information as well as the DNN numerical uncertainty (§3.4). Note that the overall performance does improve in some cases, with the best results achieved using 40 and 80 dimensional i-vectors.

Results over seen speakers are further analysed here. Increasing SSBN dimension from 13 to 100 does not improve recognition performance, because SSDNN overfits to silence target (Table 1) and because increased codes dimension introduces more numerical noise. Using SSDNN raw posteriors generally outperforms bottleneck features (SSBN) due to higher discrimination. Since raw posteriors are temporally noisy, its best performance (19.80%) is worse than 8 dimensional hand-crafted binary index codes (19.61%). The effectiveness of posteriors and hand-crafted codes suggests that the absolute value of auxiliary codes is less important than the discrimination, unless numerical stability becomes a main issue. Expanding the 8 dimensional binary index codes into UBIC further improves performance due to increased discrimination. Given the same discrimination, 170 dimensional UBIC (19.30%) outperforms 188 dimensional UBIC (19.41%) due to less numerical noise. The best performance on seen speakers (19.30%) is observed with i-vectors and 170 dimensional UBIC, achieving 10.4% relative improvement over baseline. The latter is ideal speaker bias adaptation over seen speakers implemented in informed training framework, while i-vectors manage to well discriminate all seen speakers.

In the proposed SSDNN-UBIC framework (§4.6), the speaker classification results per utterance using SSDNN are used to select UBIC for that utterance. The effectiveness of informed training on speech recognition is show to positively correlate with speaker classification accuracy (Table 1 and 2). With all SSDNNs illustrating segmental speaker classification accuracy above 92% (Table 1), the worst performance of the new system (19.36%) is still better than the best performance using SSDNN posteriors (19.80%) or SSBN (20.31%), while the best performance (19.31%) is approximately the same with i-vector informed training and DNN bias adaptation. Thus the proposed system could serve as an alternative to i-vectors based informed training. It is well suitable for fast adaptation over short utterances (5s on average in our experiments) when test speakers are mostly seen or all seen.

Table 2. %WERs of DNN baseline and informed training.

Auxiliary codes	SSBN dim	Seen	Unseen	Overall
- (baseline §4.2)	-	**21.54**	**25.01**	**23.8**
SSBN (§4.3)	13	**20.31**	25.48	23.6
	40	20.42	25.29	23.5
	60	20.39	26.87	24.5
	80	20.49	25.88	23.9
	100	21.03	25.86	24.1
SSDNN raw posteriors (§4.3)	13	19.97	25.81	23.7
	40	20.45	25.48	23.7
	60	**19.80**	26.03	23.8
	80	20.10	25.89	23.8
	100	19.86	25.56	23.5
8dim codes (§4.5)	-	19.61	25.59	23.4
170dim UBIC (§4.5)	-	**19.30**	28.77	25.4
188dim UBIC (§4.5)	-	19.41	28.25	25.1
170 dim UBIC selected by SSDNN (§4.6)	13	19.35	26.36	23.8
	40	**19.31**	26.78	24.1
	60	19.32	26.79	24.1
	80	19.34	26.82	24.1
	100	19.36	26.74	24.1
Auxiliary codes	**ivector dim**	**Seen**	**Unseen**	**Overall**
i-vector (§4.4)	13	**19.30**	26.26	23.8
	40	19.56	25.37	**23.3**
	60	20.62	26.59	24.4
	80	19.60	25.43	**23.3**
	100	19.52	26.48	24.0

6. CONCLUSION

This paper presented a unified investigation on speaker informed DNN training and compared the performance of different methods that include speaker information in the DNN front-end of a DNN-HMM-GMM system. We showed the underlying mathematical equivalence between informed training and DNN bias adaptation, and illustrated the key factors impacting the effectiveness of informed training as the dimension, discrimination and stability of auxiliary codes. I-vector based informed training was shown to be effective in a DNN front-end configuration using bottleneck features, achieving the same performance as speaker level DNN bias adaptation, with 10.4% relative improvement over baseline on seen speakers. A new informed training structure was presented as an alternative to i-vectors based adaptation, which especially targets on short utterances and speed issues. With more than 92% speaker classification accuracy from SSDNN, in our experiments the new system shows approximately the same performance as the i-vector based method and ideal speaker dependent bias adaptation.

7. ACKNOWLEDGEMENT

The authors would like to acknowledge the EPSRC poject Natural Speech Technology (NST) for supporting this research.

8. REFERENCES

[1] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, April 2007, vol. 4, pp. IV–757–IV–760.

[2] D. Yu and M. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Interspeech*. August 2011, International Speech Communication Association.

[3] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH*, 2011, pp. 437–440.

[4] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 285–290.

[5] J. P. Neto, L. B. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system.," in *EUROSPEECH*. 1995, ISCA.

[6] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems.," in *INTERSPEECH*, 2010, pp. 526–529.

[7] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 1011, pp. 827 – 835, 2007, Intrinsic Speech Variations.

[8] S. Dupont and L. Cheboub, "Fast speaker adaptation of artificial neural networks for automatic speech recognition," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 3, pp. 1795–1798.

[9] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7942–7946.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 55–59.

[11] Y. Liu, P. Zhang, and T. Hain, "Using neural network frontends on far field multiple microphones based speech recognition," in *ICASSP2014 - Speech and Language Processing (ICASSP2014 - SLTC)*, Florence, Italy, May 2014.

[12] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *SLT 2012*, December 2012.

[13] R. Doddipatla, M. Hasan, and T. Hain, "Speaker dependent bottleneck layer training forspeaker adaptation in automatic speech recognition," in *Interspeech 2014*, 2014.

[14] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE Workshop on Spoken Language Technology*, South Lake Tahoe, USA, December 2014.

[15] C. Fredouille and D. Charlet, "Analysis of i-vector framework for speaker identification in TV-shows," in *Proceedings of Interspeech'14*, 2014.

[16] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[17] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, May 2011.

[18] K. Kumatani, J. W. McDonough, and B. Raj, "Maximum kurtosis beamforming with a subspace filter for distant speech recognition," in *ASRU'11*, 2011, pp. 179–184.

[19] M. Bacchiani, "Rapid adaptation for mobile speech applications," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7903–7907.

[20] P. Karanasou, Y. Wang, M. Gales, and P. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proceedings of Interspeech'14*, 2014.

[21] van der L. Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[22] van der L. Maaten, "Barnes-Hut-SNE," *Proceedings of the International Conference on Learning Representations*, vol. abs/1301.3342, 2013.

[23] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The AMI meeting corpus: A pre-announcement," vol. 3869, pp. 28–39, 2006.

[24] T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grezl, el A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech and Language Processing*, Aug. 2011.

[25] M. J. F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 417–428, 1999.