AN INVESTIGATION OF AUGMENTING SPEAKER REPRESENTATIONS TO IMPROVE SPEAKER NORMALISATION FOR DNN-BASED SPEECH RECOGNITION

Hengguan Huang and Khe Chai Sim

School of Computing, National University of Singapore, Republic of Singapore, 117417

ABSTRACT

The conventional short-term interval features used by the Deep Neural Networks (DNNs) lack the ability to learn longer term information. This poses a challenge for training a speaker-independent (SI) DNN since the short-term features do not provide sufficient information for the DNN to estimate the real robust factors of speaker-level variations. The key to this problem is to obtain a sufficiently robust and informative speaker representation. This paper compares several speaker representations. Firstly, a DNN speaker classifier is used to extract the bottleneck features as the speaker representation, called the Bottleneck Speaker Vector (BSV). To further improve the robustness of this representation, a first-order Bottleneck Speaker Super Vector (BSSV) is also proposed, where the BSV is expanded into a super vector space by incorporating the phoneme posterior probabilities. Finally, a more fine-grain speaker representation based on the FMLLR-shifted features is examined. The experimental results on the WSJ0 and WSJ1 datasets show that the proposed speaker representations are useful in normalising the speaker effects for robust DNN-based automatic speech recognition. The best performance is achieved by augmenting both the BSSV and the FMLLR-shifted representations, yielding 10.0% – 15.3% relatively performance gains over the SI DNN baseline.

Index Terms— speaker normalisation, augmented speaker representation, deep neural network, speech recognition

1. INTRODUCTION

Recently, deep neural network (DNN) acoustic models have been found to yield good performance for automatic speech recognition (ASR) due to the ability to cope with a long span of acoustic features and model a complex mapping function [1]. The DNNs are trained with discriminative objective functions and therefore able to implicitly reduce the acoustic variability from known or unknown sources with increasing number of hidden layers. Nevertheless, the DNN's ability to compensate these variabilities is still limited due to the following reasons: i) despite being powerful in detecting the phonetic events and implicitly learning the corresponding speaker characteristics, as a global modelling approach, speaker-independent (SI) DNNs can only learn the most common variabilities and thus less sensitive to the detailed speaker attributes; and ii) the conventional short-term features used by the DNNs do not capture the long term information, which are important for the DNNs to reliably compensate any speaker-level variation.

Therefore, the existing solutions to improve the robustness of DNNs against speaker variability focus on incorporating speaker-level information to the DNNs, such as augmenting the acoustic features with i-vectors [2] and introducing speaker-dependent weights [3]. These approaches essentially trains speaker-aware DNNs that automatically learn to compensate for speaker variability. Motivated by the simplicity of the i-vector approach, this paper compares several other forms of speaker representation for feature augmentation.

In this work, we examine three forms of speaker representations. Firstly, we consider extracting bottleneck features from a bottleneck DNN trained to classify speakers in order to derive the so called Bottleneck Speaker Vector (BSV) as speaker representation. Similar to the i-vectors [4], the BSV extractor is trained with speaker labels, without requiring the phonetic transcriptions. However, the BSV is based on the assumption that the speaker characteristics within the short-term segments are uniformly distributed across all the phonetic classes. To alleviate this assumption, the bottleneck features are *soft-clustered* into phonetic groups, based on the phoneme posteriors generated by a monophone DNN and gather the first-order statistics to construct a more detailed speaker representation, which is referred to as the Bottleneck Speaker Super Vector (BSSV). Finally, the FMLLR-shifted features, which are the differences between the original and the FMLLR-transformed features, are proposed as a more fine-grain *frame-level* speaker representation.

The rest of this paper is organised as follows. Section 2 gives an overview of speaker normalisation techniques for DNN. Section 3 introduces the proposed BSV, BSSV and FMLLR-shifted speaker representations. Section 4 presents the experimental results on WSJ0 and WSJ1 to evaluate the effectiveness of the proposed speaker representations.

2. SPEAKER NORMALIZATION FOR DNN

This section provides an overview of the existing speaker normalisation techniques for DNNs. Maximum Likelihood Linear Regression (MLLR) [5] is a popular model-based technique for adapting Gaussian Mixture Models based acoustic models. Feature-based MLLR (FMLLR) [6] is a special form of MLLR that allows a feature transformation matrix to be estimated to reduce the speaker variability in the acoustic features. Training DNNs using these FMLLR transformed features have been found to yield promising improvements over the standard SI DNN systems [7]. However, the FMLLR-transformed features may not be optimum for the DNN as they are optimised for the GMM/HMM systems. Besides, the FMLLR-transformed features may remove some information that the DNN can exploit.

Another speaker normalisation approach for DNN is to augment the acoustic features with additional speaker information and then let the DNN learn the appropriate way to compensate for speaker mismatch. For example, feature augmentation using the i-vectors was proposed in [2]. i-vectors offer a robust speaker representation, which have been successfully applied to speaker verification tasks [4] to capture speaker/session variability. In essence, this method attempts to compress the speaker segments variability onto

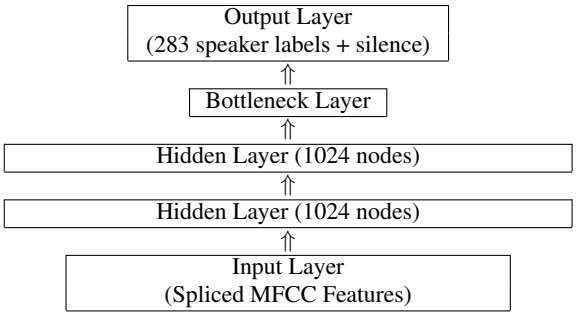

Fig. 1. The architecture of a speaker bottleneck DNN.

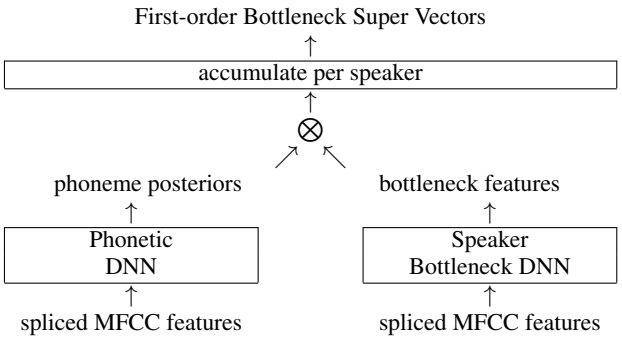

Fig. 2. The first-order bottleneck super vector feature extraction process.

a low-dimensional factor space. Similar to FMLLR, the i-vectors are learned separately and may not be optimum for the DNN training. Another advanced speaker adaptation approach was proposed in [8], where low-dimensional speaker representations (speaker codes) are jointly estimated with the DNN weights to optimise the same objective function. In fact, connecting the speaker code to the first hidden layer alone is the same as augmenting the speaker code to the acoustic features. In [8], it was found that i-vectors can be used as a better initialisation for the speaker code and further improvement can still be obtained by subsequently updating the speaker code. Another recent study [3] also shows that directly appending the i-vectors to the acoustic features is not optimum and further improvements can be obtained by learning a DNN to transform the i-vectors.

A discriminative factor analysis using DNN was proposed in [9] to address the speaker normalisation problem for DNNs. Firstly, two bottleneck DNNs were built, one is for speaker classification while the other one is for phone classification. Then, the linear activations from the output layers of these DNNs are combined using another independent DNN for the final speech recognition task. However, this approach was found to be sensitive to intra-speaker variability for speaker recognition [10].

3. PROPOSED SPEAKER REPRESENTATIONS

As discussed in the previous section, the key towards addressing the speaker mismatch problem in DNN-based speech recognition systems is to extract reliable and discriminative speaker representations and then train the speaker-aware DNNs to normalise the speaker effects. In the following, we will examine three different forms of speaker representation.

3.1. Bottleneck Speaker Vector (BSV)

The Bottleneck Speaker Vector (BSV) representation is based on the bottleneck features extracted using a bottleneck DNN, which is trained to classify speakers. The architecture of the bottleneck DNN used in this work is shown in Fig. 1. The DNN is trained using the MFCC features which are spliced with the left and right context frames. The network comprises two hidden layers with 1024 hidden units and a bottleneck layer whose size can be adjusted to extract the BSVs with the desired dimension. The size of the output layer is $S+1$, where S is the total number of training speakers. Each output unit corresponds to one training speaker and an additional unit is used to represent the silence frames. If the outputs of the bottleneck layer represent the posterior probabilities of each speaker, then the relationship between the bottleneck features and the speaker posteriors is given by

$$P(s_j|\boldsymbol{o}_t) = \frac{\exp\left(\sum_k f(a_t(k))w_{jk}\right)}{\sum_{j'}\exp\left(\sum_k f(a_t(k))w_{j'k}\right)} \quad (1)$$

where $P(s_j|\boldsymbol{o}_t)$ is the posterior probability of the jth speaker, s_j, given the observation, $\boldsymbol{o}_t$, at time t. $a_t(k)$ is the kth bottleneck feature at time t, $f(\cdot)$ is the sigmoid function and w_{jk} is the connection weight between the kth bottleneck layer unit and the jth output unit. If the output layer is modified to predict the deviation of the speaker from the Universal Background Model (UBM) and an utterance-based objective function is used [11], then the resulting bottleneck features will be similar to the i-vectors, except that the bottleneck features are extracted per frame while the i-vectors are extracted per utterance. The final BSV is obtained by simply averaging all the bottleneck features belonging to each speaker:

$$\texttt{BSV}_s = \frac{1}{|\mathcal{T}_s|}\sum_{t\in\mathcal{T}_s}\boldsymbol{a}_t \quad (2)$$

where $\texttt{BSV}_s$ denotes the BSV for speaker s, $\boldsymbol{a}_t$ is the bottleneck feature at time t, $\mathcal{T}_s$ is the set of speech frames that belong to speaker s and $|\cdot|$ is the cardinality operator. Since the bottleneck DNN is trained using a frame-based objective function, it may be sensitive to phonetic variations. In the following, we propose expanding the BSV into a super vector based on the phonetic classes, which will be described next.

3.2. First-order Bottleneck Speaker Super Vector (BSSV)

The BSV speaker representation described previously assumes that the speaker variability captured by the bottleneck features is independent of the phonetic class of the speech frames so that it is sufficient to take the average of all the bottleneck features from each speaker as the speaker vector. However, this assumption does not hold in practice. To address this problem, we propose extracting phoneme-dependent BSVs, which effectively projects the bottleneck features onto a super vector space. The resulting speaker representation is referred to as the Bottleneck Speaker Super Vector (BSSV), given by:

$$\texttt{BSSV}_s = \left[\begin{array}{cccc}\texttt{BSV}_s^{(1)^\top} & \texttt{BSV}_s^{(2)^\top} & \cdots & \texttt{BSV}_s^{(C)^\top}\end{array}\right]^\top \quad (3)$$

where $\texttt{BSV}_s^{(c)}$ denotes the BSV of speaker s for phonetic class c. The phonetic-class-dependent BSV can be obtained by soft-clustering

the speech segments of each speaker into different phonetic classes as follows:

$$\texttt{BSV}_s^{(c)} = \frac{\sum_{t \in \mathcal{T}_s} \gamma_c(t) \boldsymbol{a}_t}{\sum_{t \in \mathcal{T}_s} \gamma_c(t)} \quad (4)$$

where $\gamma_c(t)$ is the soft-assignment of the speech frame at time t to phonetic class c. Fig. 2 illustrates the BSSV extraction process. In practice, this information is not available and has to be estimated. In this work, a DNN is used to estimate $\gamma_c(t)$. The grouping used to extract the BSSV can also be estimated directly from the bottleneck speaker using distance-based unsupervised clustering.

3.3. FMLLR-shifted Features

So far, we have considered approaches that attempt to obtain a global representation for each speaker. As a result, a constant speaker vector is appended to the acoustic features for each speaker. However, this representation is not able to handle *intra-speaker* variabilities. To address this issue, we propose a more localised and fine-grain speaker representation, which is given by the difference between the original acoustic features and the FMLLR-transformed features. We refer to these features as the FMLLR-shifted features. Since FMLLR minimises the speaker variability in the acoustic features (including intra-speaker variabilities), the FMLLR-shifted features are believed to encode useful speaker variability information that the DNN can exploit. Although the FMLLR-shifted features are obtained using a global FMLLR transform, we believe that presenting the FMLLR-shifted features directly to the DNN is more advantageous as it allows other frame-level speaker attributes to be captured explicitly.

4. EXPERIMENTAL RESULTS

4.1. Experimental Setup

To evaluate the effectiveness of the proposed speaker representations, we perform experiments on the Wall Street Journal (WSJ) corpus [12]. The training data from both WSJ0 and WSJ1 were used, including 283 speakers and around 80 hours speech recordings. We report the ASR performance in terms of Word Error Rate (WER) on three test sets: DEV93, EVAL92 and EVAL93. All the systems use Mel Frequency Cepstral Coefficient (MFCC) with 39 dimensions as the basic features, including 12 static coefficients, energy, and the first two derivatives. All the inputs to the DNNs are normalised to have zero mean and unit variance. This training data is also used to estimate the loading matrix for the speaker i-vectors. All the DNNs used in this work are trained using the Kaldi toolkit. The training labels are obtained from the state alignments using an MMI-trained GMM/HMM system. Each DNN has five hidden layers with 1024 nodes. The basic inputs are made up of 15 frames of features.

4.2. i-vector vs. Bottleneck speaker vectors

Table 1 compares the WER performance of i-vectors (IVEC) and BSV. The numbers in parentheses denote the size of the speaker vector. For the extraction of the BSVs, a speaker bottleneck DNN is trained with two hidden layers (each with 1024 nodes) and a bottleneck layer. We investigated two bottleneck layer sizes: 25 and 100. The output layer has 284 units (283 training speakers and silence). 'None' corresponds to the baseline SI DNN system without appending any speaker representation to the input features. All the systems with additional speaker representation are trained with a *warm-start* configuration, where the well-train baseline system is used as the initial model and the weights connecting the additional inputs and the first hidden layer units are randomly initialised. 10% of the training data is used for cross-validation, which contains utterances from *all* the training speakers. This configuration is important to ensure that the ability to compensate for the speaker variability is properly considered during training.

Speaker Representation	DEV93	EVAL92	EVAL93
None	8.3	4.4	7.2
IVEC(25)	7.7	4.3	7.1
BSV(25)	7.9	4.2	7.0
IVEC(100)	**7.4**	4.3	**6.7**
BSV(100)	7.6	4.4	**6.7**
BSV(100) + IVEC(25)	7.5	**4.2**	6.9

Table 1. WER comparison between the IVEC and BSV speaker representations.

Speaker Representation	DEV93	EVAL92	EVAL93
None	8.3	4.4	7.2
BSV(25)	7.9	4.2	7.0
First-order BSSV(25)	7.2	4.4	7.0

Table 2. WER of first order speaker supervectors

In general, adding speaker representation information consistently improve the performance over the baseline system across all the test sets. With the size of 25, both BSV and IVEC achieved comparable performance. Slight improvements can be obtained on DEV93 and EVAL93 if the size is increased to 100. The improvement is statistically significant for BSV, but not significant for IVEC. Finally, appending both BSV and IVEC achieved further improvements on DEV93 and EVAL92 compared to adding either IVEC or BSV alone. Overall, the 'BSV(100) + IVEC(25)' speaker representation achieved 9.3%, 0.7% and 4.8% relative WER reductions over the baseline SI system on the three test sets, respectively. Note that the DEV93 development test set was collected with the training data and therefore may have a closer recording condition to the training data as compared to EVAL92 and EVAL93. This might explain the smaller gains on EVAL92 and EVAL93. Significance tests show that all the performance differences among all the systems on EVAL92 and EVAL93 were statistically insignificant.

4.3. First order bottleneck speaker supervectors

Table 2 compares the results of using the BSV and the first-order BSSV as the speaker representation. The first-order BSSV is extracted according to the process described in Section 3.2. In this work, the phonetic classes correspond to the 40 monophones. A triphone DNN is used to predict the triphone state posteriors, which are then mapped to the monophone posteriors and used as the soft-alignments for the phonetic classes, $\gamma_c(t)$. The first-order BSSV(25) is obtained by expanding BSV(25), which leads to a 1000-dimensional speaker representation (40×25). Based on the results, BSSV(25) achieved a larger improvement compared to BSV(25) on DEV93. A slight degradation was observed on EVAL92 and the improvement on EVAL93 was marginal. Significance tests show that BSSV(25) is significantly better than BSV(25) on DEV93 but the improvements on EVAL92 and EVAL93 were not significant. This suggests that the proposed BSSV may be more effective when there is less mismatched between the training and testing conditions.

Speaker Representation	DEV93	EVAL92	EVAL93
None	8.3	4.4	7.2
FMLLR	7.3	4.1	6.8
LDA + ΔFMLLR	7.1	4.2	6.6
+ BSV(100)	7.0	4.1	**6.4**
+ BSV(100) + IVEC(25)	7.1	4.0	6.5
+ First-order BSSV(25)	**7.0**	**3.9**	6.5

Table 3. WER of FMLLR-shifted features for speaker normalisation

4.4. FMLLR-shifted features

The results for incorporating the FMLLR-shifted features (ΔFMLLR) are shown in Table 3. In this work, the FMLLR feature extraction is performed as follows. Linear Discriminant Analysis (LDA) is first used to reduce the acoustic feature dimension and the LDA-transformed features are then used to estimate the FMLLR transform. So, the FMLLR-shifted features are computed as the difference between the LDA-transformed and FMLLR-transformed features. From Table 3, it is observed that training DNN using the FMLLR transformed features yield relative WER improvements of 12.1%, 7.1% and 5.5% on the three test sets, respectively. In fact, these results are comparable or better than all the results reported in Table 1 and Table 2. This shows the effectiveness of FMLLR in removing speaker variabilities in the data. If the DNN is trained using the ΔFMLLR features appended to the LDA-transformed features, instead of using the FMLLR transformed features, further improvements of about 0.2% absolute can be achieved on DEV93 and EVAL93. However, the performance on EVAL92 is slightly worse. This suggests that the ΔFMLLR features do contain some useful information that the DNN can exploit.

In the last three rows of Table 3, we examined the complementariness between the ΔFMLLR features and the other global speaker representations. In most cases, combining the LDA + ΔFMLLR features with the other speaker representations improved the ASR performance. The overall best configuration is achieved by combining ΔFMLLR with the first-order BSSV(25), yielding 7.0%, 3.9% and 6.5% WERs on the three test sets, respectively. These translates to 10.0%–15.3% relative WER reductions compared to the SI DNN system and 3.6%–4.7% relative WER reductions compared to the FMLLR DNN system.

5. CONCLUSIONS

This paper has investigated several forms of speaker representation that can be augmented to the acoustic features to train DNN-based acoustic models for robust automatic speech recognition. The incorporation of such speaker-level information lets the DNNs learn the appropriate parameters to implicitly compensate for any speaker variability found in the speech data. Firstly, we examined the Bottleneck Speaker Vector (BSV) representation, which is obtained by averaging the speaker bottleneck features over each speaker. Next, we improved BSV by expanding it into a phonetic-class-dependent bottleneck speaker super vector (BSSV), where phone posteriors are used to accumulate first-order BSV statistics. Finally, we proposed using the FMLLR-shifted features as a frame-dependent speaker representation to introduce more detailed speaker information for the DNN to exploit. Experimental results on the WSJ0 and WSJ1 datasets show that these speaker representations achieved comparable or better performance compared to the existing i-vector speaker representation. Besides, adding more than one speaker representation also led to further performance improvements. The best configuration is achieved by combining the FMLLR-shited features and the first-order BSSV, which gave 10.0%–15.3% relative WER reduction over the baseline DNN system.

6. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[2] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2013.

[3] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Proc. Interspeech*, 2014.

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 14481460, 2007.

[5] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, 1995.

[6] M.J.F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech & Language*, vol. 12, 1998.

[7] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*, 2011, pp. 24–29.

[8] Shaofei Xue, O. Abdel-Hamid, Hui Jiang, Lirong Dai, and Qingfeng Liu, "Speaker and session variability in GMM-based speaker verification," *IEEE/ACM Transactions on Audio, Speech & Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[9] M. Ferras and H. Bourlard, "MLP-based factor analysis for tandem speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, IEEE.

[10] Hagai Aronowitz, Dror Irony, and David Burstein, "Modeling intra-speaker variability for speaker recognition," in *Proc. Interspeech*, 2005.

[11] Sibel Yaman, Jason Pelecanos, and Ruhi Sarikaya, "Bottleneck features for speaker recognition," in *Proc. Odyssey*, 2012, vol. 12.

[12] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. ICSLP*, 1992.