Articulatory and Spectrum Information Fusion Based on Deep Recurrent Neural Networks

Jianguo Yu, *Student Member, IEEE*, Konstantin Markov, *Member, IEEE*, and Tomoko Matsui, *Member, IEEE*

***Abstract*—Many studies have shown that articulatory features can significantly improve the performance of automatic speech recognition systems. Unfortunately, such features are not available at recognition time. There are two main approaches to solve this problem: a feature-based approach, the most popular example of which is the acoustic-to-articulatory inversion, where the missing articulatory features are generated from the speech signal, and a model-based approach, where articulatory information is embedded in the model structure and parameters in a way that allows recognition using only acoustic features. In this paper, we propose two new methods to integrate articulatory information into a phoneme recognition system. One of them is feature based, and the other is model based. In both cases, the underlying acoustic model (AM) is a deep neural networks-hidden Markov model (DNN-HMM) hybrid. In the feature-based method, the articulatory inversion DNN and the acoustic model DNN are trained jointly using a linear combination of their loss functions. In the model-based method, we utilize the generalized distillation framework to train the AM DNN. In this case, first, a teacher DNN is trained on both the acoustic and articulatory features, and then its outputs are used as additional targets during the AM DNN training with acoustic features only. A 7-fold cross-validation experiments using 42 speakers from the XRMB database showed that both the proposed methods provide about 22% to 25% performance improvement with respect to the DNN acoustic model trained with acoustic features only.**

***Index Terms*—Automatic speech recognition, deep recurrent neural networks, articulatory information, distillation training.**

I. Introduction

AUTOMATIC Speech Recognition technology is becoming good enough to enable many exciting applications, yet current ASR systems still suffer from acoustic variabilities such as background noises, speakers, accents, recording conditions, etc.

In order to make the systems more reliable and robust, researchers have been trying to utilize additional information such as articulatory organs (lips, tongue, velum, etc.) movements, which is more suitable to model the coarticulation effects [1]. Many studies [2]–[5] have shown that articulatory information can improve the ASR performance and increase the robustness against noise contamination and speaker variation. However, incorporating such information is challenging since it is impractical to obtain observations of articulators movements in real-life speech recognition scenarios. This constraint requires ASR systems to utilize articulatory data for training only, i.e. to be able to recognize without them.

One approach to incorporate the articulatory information is to utilize it at the feature level, which we call *feature based* approach.

The most straightforward and widely used implementation of this approach is the articulatory-to-acoustic inversion, where the missing articulatory features are generated from the acoustic signal. This, however, is not a simple task since the mapping between acoustic and articulatory data spaces is non-linear and not unique [6]. Various machine-learning methods haven been applied to model this mapping, for example, Hidden Markov Model (HMM) [7], Gaussian Mixture Model (GMM) [8], and Mixture Density Networks (MDN) [9]. The Canonical Correlation Analysis (CCA) used in [10] and its deep learning extension DCCA [11] are also the feature based approaches where transformations of the acoustic features are learned such that they become maximally correlated with the articulatory data. Since the Deep Neural Networks (DNN) have become the new state-of-the-art tool in a wide range of application domains, multiple studies [12]–[14] have shown that DNNs' ability to learn highly non-linear and complex functions results in better prediction of articulatory trajectories from acoustic speech data. Several Deep Autoencoder (DAE) architectures for articulatory inversion are also investigated in [15].

In this work, we use a bidirectional Deep Recurrent Network (biRNN) to approximate the acoustic-to-articulatory mapping. RNNs are better suited to model temporal processes such as speech and have been successfully used in acoustic models [16]. Our phoneme recognition system is also built with biRNN based DNN-HMM acoustic model. In contrast to other studies, however, we learn the articulatory inversion biRNN and the acoustic model biRNN jointly by combining their loss functions. This leads to training with a common goal and results in better performance. Further performance boost can be achieved if the networks are initialized with separately trained biRNNs.

Another way to integrate articulatory information in the ASR systems is the *model based* approach, where the articulatory data are used to adjust the parameters and optionally the structure of the acoustic model in a way that does not require articulatory observations during testing. Obviously, in this case no articulatory

Manuscript received July 30, 2018; revised November 28, 2018; accepted January 6, 2019. Date of publication January 21, 2019; date of current version February 15, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jun Du. *(Corresponding author: Jianguo Yu.)*

J. Yu and K. Markov are with the Human Interface Lab, University of Aizu, Fukushima 965-8580, Japan (e-mail: gxiiukk@gmail.com; markov@u-aizu.ac.jp).

T. Matsui is with the Department of Statistical Modeling, Institute of Statistical Mathematics, Tokyo 190-8562, Japan (e-mail: tmatsui@ism.ac.jp).

Digital Object Identifier 10.1109/TASLP.2019.2894554

inversion is necessary. In [2], a hybrid Bayesian network/HMM acoustic model incorporates the articulatory data. A relatively new way to incorporate knowledge into neural networks is the so called *Distillation Training*, where an additional loss function with soft targets is also being minimized during training. In [17], Hinton *et al.* have shown that soft targets from complex models can transfer knowledge to small models that are easy to deploy. Recently, the learning using privileged information [18] and the distillation methods have been combined into a *Generalized Distillation* framework [19] which utilizes the strengths of both methods. Knowledge is transferred through soft targets from a "teacher" model trained with additional features to a "student" model with no access to those features. In our previous work [20], we applied Generalized Distillation in a feedforward DNN-HMM system. The results showed that soft targets can transfer knowledge from the teacher trained with both articulatory and acoustic data to the student model learned from acoustic data only. In [21], the effectiveness of RNN models pretrained with soft targets was also investigated and compared with the distillation method. Both approaches lead to models that have higher generalization abilities.

In this paper, we apply the Generalized Distillation training framework to our biRNN based acoustic model in a way similar to our previous work with feedforward DNN [20]. This time, however, distillation training consistently reduced the Phoneme Error Rates (PER) in wide range of temperature parameter (T) setting, while we previously observed diverse performance improvement depending on this parameter.

The rest of the paper is organized as follows. Next Section describes several closely related studies. Section III briefly introduces the hybrid DNN-HMM acoustic models and in Section IV, we describe our methods for articulatory data fusion. In Section V, we present our experiments and the obtained results are summarized in Section VI. An analysis of the results with respect to the phoneme language model's influence as well as the lexical content of the utterances is given in Section VII. Finally, Section VIII includes our conclusions.

II. Related Studies

Most of the methods utilizing articulatory information are feature based and have to solve the difficult task of articulatory inversion. For example, in [22], a joint probability density of an articulatory and acoustic parameters is modeled using a GMM to provide articulatory inversion mapping. A Support Vector Regression (SVR) was applied in [23] to the task of transforming the acoustic speech signal onto EMA trajectories. In [24], an HMM-based speech production model was presented which consists of the articulatory HMM for each phoneme and an articulatory-to-acoustic mapping for each HMM state. All these studies use shallow models which have limited abilities to learn highly non-linear transformations such as articulatory information.

In [12], a deep belief network was implemented and obtained an average Root Mean Square Error (RMSE) of 0.95 on the MNGU0 test dataset and in [13], the result was improved to 0.885 by a mixture density network.

Although a feedforward NN with large input window size gives better predictions, it also requires more data to train and the dimension of the input vector, as well as the weight matrix of the first layer, may become very large even though most weights are redundant. Instead of concatenating a window of frames together, an RNN can learn to store useful information from the time series data fed to it one by one. In [25] and [26], bidirectional LSTMs were used for speaker dependent articulatory inversion and have boosted the articulatory inversion performance on the MNGU0 test set. RNNs are good at learning the inversion mapping. A drawback of RNN-based inversion method for ASR is that when the acoustic model and inversion model are both based on RNNs, the computational cost may become prohibitive for practical real-time deployment. Another example of speaker dependent articulatory inversion method is described in [15] where multiple deep autoencoder architectures are investigated for the acoustic-acticulatory mapping. In addition to the MNGU0 dataset, the MSAK0 male voice from the MOCHA-TIMIT database is used. Speech recognition performance is evaluated in terms of frame level phone classification error (flPCE) as well as phone error rate (PER). The best results obtained for MNGU0 and MSAK0 are 11.6% and 27.5% PER respectively. Only few studies approach the the problem in a speaker independent way, such as [11] and [27], where the XRMB corpus is used. However, due to the specifics of the DCCA method they use, the number of test speakers is still rather small, only 12, which hardly makes the result of 24.5% PER truly speaker-independent.

In contrast to the feature based methods, model based methods introduce less or no extra computational cost during recognition since no estimation of the articulatory movements is necessary. For example, in [2], a hybrid HMM/BN model is adopted to embed the articulatory information inside the model. Not only the articulator position but also velocity and acceleration are taken into account and a latent discrete variable is used for each of them within a Bayesian Network that substitutes the traditional GMM state probability distribution. Similarly, the works [28]–[30] use Dynamic Bayesian Network to treat articulatory information as hidden variable.

III. DNNs as Acoustic Models in ASR

DNNs have gratly changed the pipeline of the current ASR systems. In the so-called DNN-HMM hybrid systems [31], [32], big performance boosts were achieved by replacing Gaussian Mixture Models (GMMs) by a feedforward Neural Network (FNN) which takes a window of several frames as input and produces posterior probabilities over HMM states as illustrated in Fig. 1.

The DNN-HMM training procedure can be summarized as follows:

1) Train a standard GMM-HMM based recognizer.
2) Use forced alignment to get the DNN target labels, i.e. state ID for each observation vector.
3) Count the occurrences of hidden states in the training set to compute the prior probabilities of HMM states.

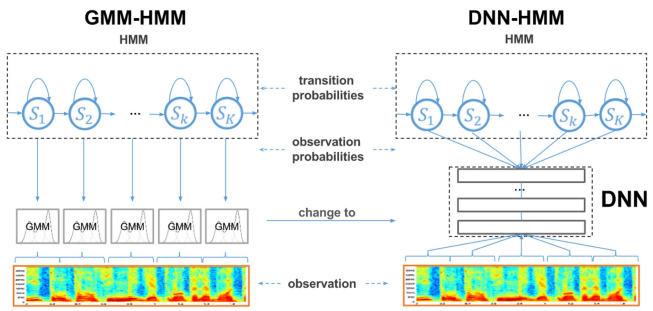


Fig. 1. GMM-HMM vs DNN-HMM acoustic models.

4) Train a DNN classifier that maps observations to HMM states.
5) Compute scaled observation likelihoods based on Bayes' rule and feed them to the HMM decoder.

The HMM state probabilities can also be obtained by RNNs instead of FNNs, where the current output does not only depend on the current input but also on previous hidden states. Instead of concatenating frames into a big vector, a typical RNN maps arbitrary length sequence to a fixed length vector by applying the same transition function f with the same parameters at every time step [33].

A natural extension of RNN based on the idea that the output at time t may not depend only on the previous inputs but also on the future inputs is called *bidirectional* RNN. It is implemented by processing the data sequence in both directions with two separate hidden layers. However, standard RNNs cannot learn long-term dependency due to the vanishing gradient problem [34].

Long short term memory (LSTM) [35] is a particular implementation of RNNs that uses input, output and forget gates to prevent the vanishing gradient problem. Gated Recurrent Unit (GRU) [36] is an LSTM variation that merges the forget and input gates into a single update gate resulting in a less complex structure, yet the performance is similar to that of the LSTM. The GRUs used in our experiments are specified by Eq. (1).

$$\begin{aligned} r_t &= \sigma_r(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \\ u_t &= \sigma_u(x_t W_{xu} + h_{t-1} W_{hu} + b_u) \\ c_t &= \sigma_c(x_t W_{xc} + r_t \odot (h_{t-1} W_{hc}) + b_c) \\ h_t &= (1 - u_t) \odot h_{t-1} + u_t \odot c_t \end{aligned} \tag{1}$$

Our biGRU is implemented by stacking two GRUs together and the biGRU hidden state is the sum of the forward GRU hidden state h_t^f and the backward GRU hidden state h_t^b.

IV. Articulatory and Spectrum Fusion Methods

A. Standard Articulatory Inversion

In the conventional feature based approach, the articulatory inversion model and the acoustic model are trained separately. Articulatory features are first generated using the inversion model and then combined with the acoustic features for acoustic model training. The same procedure is applied during recognition.

1) Training Procedure: Given the training data $\{(x_i, a_i)\}_{i=1}^n$, we would like to learn a mapping f_{INV} from the acoustic space to the articulatory space by minimizing a mean squared error (MSE) loss function L with some form of regularization

$$f_{INV} = \arg \min_{f_{inv} \in \mathcal{F}_{INV}} \frac{1}{n} \sum_{i=1}^{n} L(f_{inv}(x_i), a_i) + \Omega(||f_{inv}||) \tag{2}$$

$$L(f_{inv}(x_i), a_i) = \frac{1}{q} \sum_{j=1}^{q} (f_{inv}(x_i)_j - a_{ij})^2. \tag{3}$$

Here, $x_i \in \mathcal{R}^p$ and $a_i \in \mathcal{R}^q$ are the acoustic and articulatory feature vectors, F_{INV} is a space of mapping functions from $\mathcal{R}^p$ to $\mathcal{R}^q$ and Ω is L2 norm regularizer.

The acoustic model f_{AC} maps concatenated feature vectors into HMM state probabilities through a softmax function and is trained by minimizing the categorical cross entropy loss function H:

$$f_{AC} = \arg \min_{f_{ac} \in \mathcal{F}_{AC}} \frac{1}{n} \sum_{i=1}^{n} H(\sigma(f_{ac}(x_i^*)), y_i) + \Omega(||f_{ac}||) \tag{4}$$

$$H(y_i, \sigma(f_{ac}(x_i^*))) = -\sum_{j=1}^{c} y_{ij} \log \sigma(f_{ac}(x_i^*)_j) \tag{5}$$

where, $x_i^* = concat(x_i, f_{INV}(x_i)) \in \mathcal{R}^{p+q}$ is the concatenated vector of acoustic and reconstructed articulatory features, $y_i \in \Delta^c$ is a vector representing target HMM states, Δ^c is the c-dimensional space of probability vectors, F_{AC} is a class of functions from $\mathcal{R}^d$ to $\mathcal{R}^c$, $\sigma : \mathcal{R}^c \to \Delta^c$ is the softmax function.

2) Testing Procedure: During testing, the inversion model is used to generate the articulatory features which are concatenated with the acoustic features and used as input to the acoustic model.

B. Joint Articulatory Inversion and Acoustic Model Training

The recognition result of standard inversion depends on the outputs of the inversion model. However, the inversion model is only trained to minimize its MSE loss with respect to the articulatory vector and does not have any knowledge about how its outputs will be used later. It would be helpful to tell the inversion model what the final goal is. Thus, we train the inversion model and the acoustic model as a single network so that the parameters of the two models are trained to minimize the final objective jointly.

The joint training procedure and the network structure are illustrated in Fig. 2. The acoustic vector x_i is passed through the inversion DNN f_{inv} to calculate the MSE loss with respect to the articulatory vector a_i using Eq. (3). In addition, x_i concatenated with the inversion DNN output $f_{inv}(x_i)$ (which is expected to have a physical meaning of articulatory feature) and the vector $concat(x_i, f_{inv}(x_i))$ (denoted as x_i^*) is passed through the acoustic model DNN f_{ac} followed by a softmax output function σ to calculate the categorical cross entropy loss with respect to HMM state labels y_i using Eq. (5).

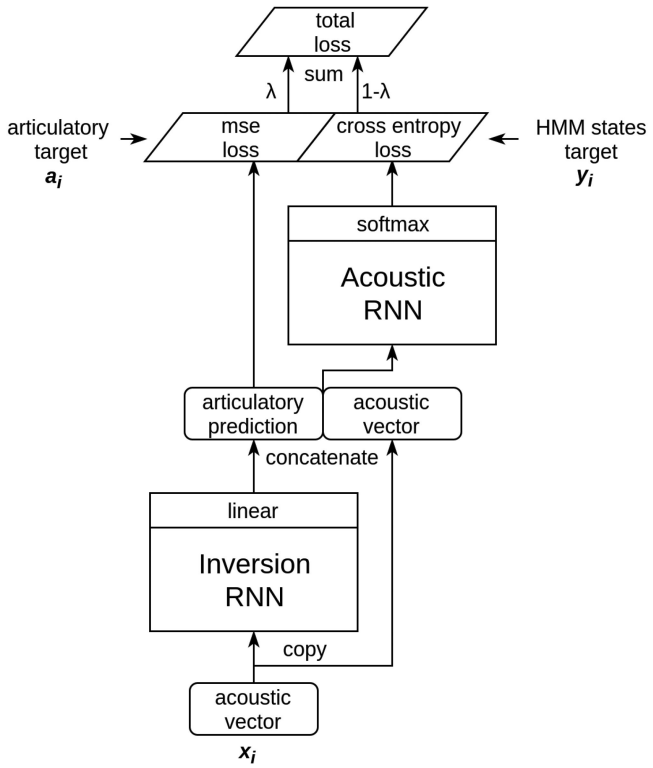


Fig. 2. Block diagram of the joint articulatory inversion and acoustic model DNN training.

1) Joint Training Procedure: During the training the entire network is trained by minimizing the weighted average of the two loss functions controlled by $\lambda \in [0, 1)$ using Eq. (6)

$$f_{INV}, f_{AC} = \arg \min_{f_{inv} \in \mathcal{F}_{INV}, f_{ac} \in \mathcal{F}_{AC}} \frac{1}{n} \sum_{i=1}^{n} [(1-\lambda) \times H(\sigma(f_{ac}(x_i^*)), y_i) + \lambda L(f_{inv}(x_i), a_i) + \Omega(||f_{ac}|| + ||f_{inv}||)] \quad (6)$$

We have to note that the weight λ which controls the contribution of each loss function in the weights update cannot be set to 1, because this will eliminate the HMM states as targets and destroy the acoustic model. On the other hand, $\lambda = 0$ means that articulatory targets are eliminated and no articulatory information is integrated.

2) Testing Procedure: During testing, the HMM state probabilities obtained from the joint model are fed to the HMM decoder as shown in Fig. 3. The only input data in this case are the acoustic features as in any articulatory inversion based system.

C. Acoustic Model Training using Generalized Distillation

Generalized distillation method has been proposed in [19] to combine two techniques - Hinton's distillation [17] and Vapnik's privileged information [18] which enables machines to learn from other machines. In this framework, an "intelligent teacher" is incorporated into machine learning and the training data is

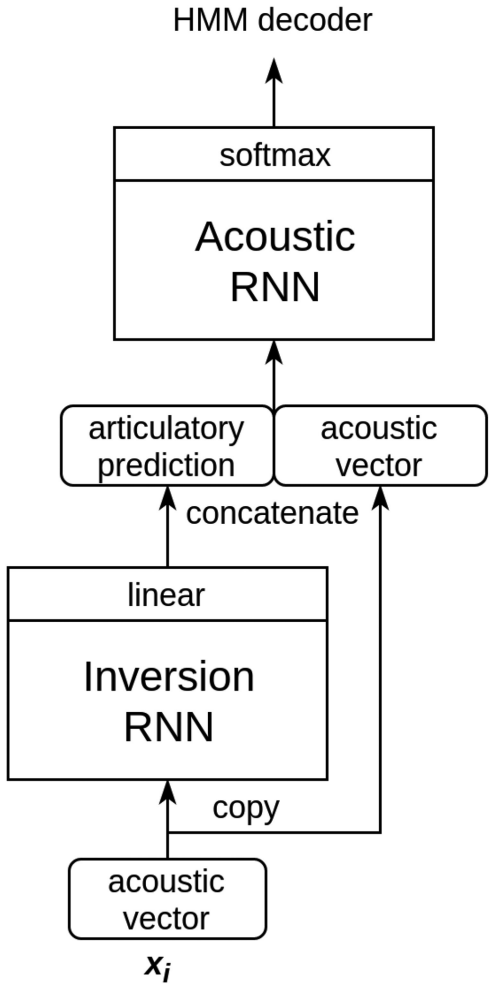


Fig. 3. Block diagram of the joint articulatory inversion testing.

formed by a collection of triplets

$$(x_1, x_1^*, y_1), \ldots, (x_n, x_n^*, y_n) \sim P^n(x, x^*, y),$$

where (x_i, y_i) is a feature-label pair and x_i^* represents a privileged information about x_i provided by an intelligent teacher and is supposed to have higher discriminating power than x_i itself. The teacher is assumed to develop a language that effectively communicates information to help the student come up with better representation and to enable to it learn characteristics about the decision boundary which are not contained in the student training data. In our task, combined acoustic and articulatory feature vectors are regarded as privileged information source, $x_i^* = \text{concat}(x_i, a_i)$.

The training procedure is as follows:

1) Learn teacher $f^T \in \mathcal{F}^T$ using $\{(x_i^*, y_i)\}_{i=1}^n$.

$$f^T = \arg \min_{f^t \in \mathcal{F}^T} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \sigma(f^t(x_i^*))) + \Omega(||f^t||) \quad (7)$$

where, $x_i^* \in \mathcal{R}^d$, d is the total dimension of acoustic and articulatory features, $y_i \in \Delta^c$, $\mathcal{F}^T$ is a class of functions from $\mathcal{R}^d$ to $\mathcal{R}^c$, $\sigma : \mathcal{R}^c \to \Delta^c$ is a softmax function, l is a loss function (in our case, it is the categorical cross

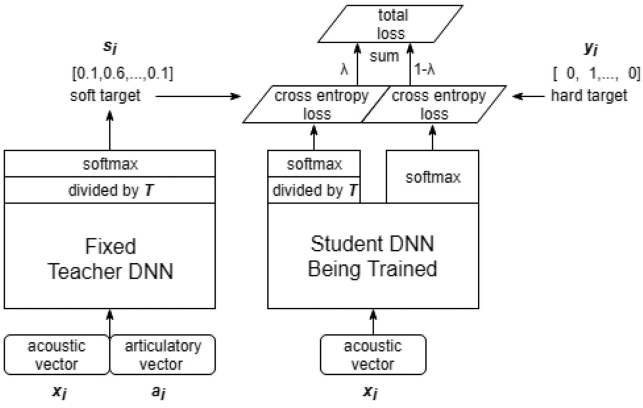


Fig. 4. Student training block diagram. In contrast to hard targets y_i, soft targets s_i provide information about between class relations.

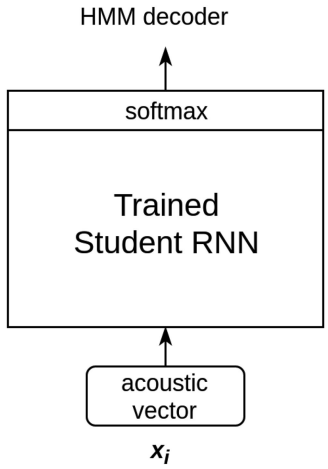


Fig. 5. Testing with student DNN. No extra cost is required during the test.

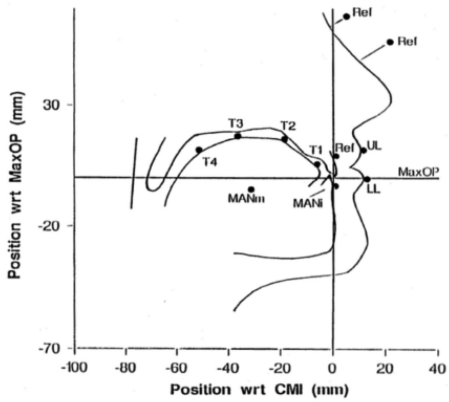


Fig. 6. Placement of the 8 pellets on T1, T2, T3, T4, MANm, MANi, UL, and LL points.

entropy from Eq. (5)). The paramters of teacher model are then fixed.

2) Compute teacher soft labels $\{s_i\}_{i=1}^n$ using temperature parameter T, where $s_i = \sigma(f^T(x_i^*)/T) \in \Delta^c$. T is normally set to 1. We use a higher value for T to soften the probability distribution over classes.
3) Learn student $f^S \in \mathcal{F}^S$ from Eq. (8) using $\{(x_i, y_i, s_i)\}_{i=1}^n$ and imitation parameter $\lambda \in [0, 1]$. Because the magnitudes of the gradients produced by the soft targets scale as $1/T^2$, multiplying the second loss by T^2 is necessary [17].

$$f^S = \arg\min_{f^s \in \mathcal{F}^S} \frac{1}{n} \sum_{i=1}^{n} [(1-\lambda) l(y_i, \sigma(f^s(x_i))) + T^2 \lambda l(s_i, \sigma(f^s(x_i)/T))] \quad (8)$$

The student DNN training procedure is illustrated in Fig. 4. The outputs of the teacher DNN softened by the temperature parameter T are used as soft targets s_i and together with the hard targets y_i act as arguments of the student DNN loss function as in Eq. (8). The input training data for the student DNN consists of acoustic features only. The corresponding concatenated acoustic and articulatory features, are given to the teacher DNN input in order to calculate the soft targets. However, only the student DNN parameters are updated during this procedure.

During testing, only the student DNN is used and the state probability predictions from the "hard" output, i.e. the output that was compared with the hard targets during training, are fed to the HMM decoder as shown in Fig. 5. Student DNN model trained using this method does not need articulatory feature nor extra computational resources during testing and is as fast as the standard DNN acoustic model.

V. Experiments

Every ASR system used in our experiment is a hybrid DNN-HMM system, where DNN is used to predict HMM state posterior probabilities given an input data vector. These probabilities are converted to likelihoods using state priors and standard decoding is performed to obtain the recognition result. Targets for the student (teacher) DNN learning are obtained by first training conventional GMM-HMM systems using acoustic (acoustic + articulatory) features. Then, target states are identified by forced alignment.

A. Database

The database used in our experiments is the University of Wisconsin X-ray microbeam database (XRMB) [37], which consists of simultaneously recorded acoustic and articulatory measurements from 47 American English speakers (22 males, 25 females). Each speaker's recordings comprise at most 118 tasks such as a sequence of numbers, TIMIT sentences spoken in different ways (normal, slow, fast), isolated word sequences, paragraphs as well as non-speech oral motor. In our experiments, only *word: standard*, *sentence: normal*, *counting(1-20)*, and *number sequences* were used. The articulatory measurements are horizontal and vertical displacements of 8 pellets on the tongue, lips, and jaw as shown in Fig. 6 [37].

We downsampled the acoustic signal from 21.74 kHz to 16 kHz and our acoustic features are 13-dimensional Mel-frequency cepstral coefficients (MFCCs) computed every 10 ms over a 25 ms window, along with their first and second derivatives, resulting in 39-dimensional vectors. We also downsampled the articulatory data from the original rate of 145.7 Hz to 100 Hz to match the frame rate of acoustic features

TABLE I
DETAILS OF THE TRAIN, VALIDATION, AND TEST SETS

	Train	Test	Validation	Total (Unique)
Speakers	32	6	4	42
Female	17	3	2	22
Male	15	3	2	20
Sentences	2652	491	295	3438 (81)
Words	24632	4105	2506	31243 (213)
Phonemes	86067	13407	8220	107694 (39)
Hours	2:12:46	0:22:8	0:14:6	2:49:0

and use the x, y coordinates of the 8 articulators along with their first and second derivatives as articulatory feature vectors of 48 dimensions. Including the first and second derivatives of the articulatory data is helpful since the movement itself cannot tell apart speech pause from other phonemes. Finally, all feature vectors are mean and variance normalized on per utterance basis.

Due to limitations in the recording technologies, articulatory measurements contain missing data when individual pellets are mistracked. Though there are methods to reconstruct missing data [38], we decided to use only complete data samples. Phoneme alignment was done using the Penn Phonetics Lab Forced Aligner [39] and the missing entries, as well as speech data which are not consistent with their orthographic transcripts, were removed. Utterances are split into files, each containing only one sentence with silence parts at the beginning and end reduced to 150 ms. After excluding the speakers who had only few utterances left, our dataset was reduced to about 3 hours, while the whole database has 19 hours in total. All experimental results are obtained from a 7-fold cross validation with 6 speakers for testing, 4 speakers for validation and 32 speakers for training in each fold. Unlike many other studies, this makes our models as *speaker-independent* as possible. Table I summarizes the details about our data sets.

We built two conventional GMM-HMM recognizers with 38 Gaussian components per state using the train and validation data. One uses only acoustic features (39 dim) and the other uses both the MFCCs and articulatory features (87 dim). They are both standard 3-state left-to-right context independent monophone HMM models. The phoneme language model is a simple bi-gram trained on data transcriptions including the paragraph task. We use 39 distinct phonemes extracted using the Carnegie Mellon University pronunciation dictionary-0.7b [40] and one silence HMM (120 HMM states in total). With the GMM-HMM systems, we generated frame level training targets for the neural networks with corresponding features.

B. Common DNN Settings

In our experiments, we used RNN for both the acoustic and inversion models. DNNs have a lot of hyper parameters, such as number of layers, number of nodes, activation function type, etc. In a series of preliminary experiments, we tried various RNN structures and parameters in order to achieve the best possible baseline performance. Finally, we chose two biGRU layers stacked in between feedforward dense layers which showed the best performance. Similar findings are reported in [26] and [41].

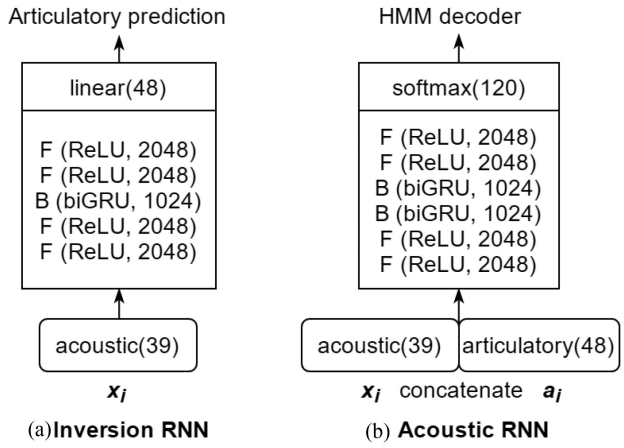


Fig. 7. Inversion and acoustic RNN structures. The number of nodes and the activation function of each layer is given. Note that the biGRU layer consists of two GRUs layers, so the number of nodes is for each of them.

TABLE II
COMMON DNN PARAMETERS

Regularization	dropout (30%), L2 (1e-3)
Regression loss function	MSE
Regression Output activation	Linear
Classification loss function	Categorical Cross-entropy
Classification Output activation	SoftMax
Hidden feedforward activation	ReLU
Hidden feedforward nodes	2048 per layer
GRU (per direction) nodes	1024 per layer
Update	Adam

Although in principle a DNN with more recurrent layers should be able to provide similar performance, yet it takes much longer to propagate the information through the recurrent layers than feedforward layers and deeper RNNs easily become over-fitted after several epochs of training. Thus, most DNNs in our experiments have following hidden layers: 2 feedforward (F) layers with ReLU activation followed by 2 biGRU (B) layers on top of which there are another 2 ReLU feedforward layers. This structure is denoted as FFBBFF and is shown in Fig. 7 for both the acoustic and inversion DNNs. The other common settings are summarized in Table. II.

For regularization, a dropout layer with 30% dropout rate is inserted after every feedforward layer and biGRU layer and L2 regularizations of feedforward layers with a rate of 0.001 are also added to the final loss, which is $10^{-3}\sum(\|\theta\|^2)/2$ and θ is the weights of a layer.

For the training, a gradient clipping norm of 10.0 is set to prevent the exploding gradient and the weights of gates in GRU layers are initialized using orthogonal matrix initialization [42], which we found important for the training. Finally, all DNNs were first trained with 9e-5 learning rate and fine-tuned with 5e-6 learning rate once the validation data losses did not go down for 3 epochs.

C. Articulatory Inversion Experiments

1) Baseline System: As a baseline, we adopt a system where the inversion model and acoustic model are trained separately. The inversion model architecture is FFBFF illustrated in

Fig. 7-a. The inputs and outputs are the MFCC (39 dim) and articulatory feature (48 dim) vectors respectively.

Our acoustic model architecture is FFBBFF and is shown in Fig. 7-b. The train data x_i^* are concatenated acoustic and generated articulatory vectors (87 im) and the "hard" targets y_i are one-hot vectors (120 dim) where the component corresponding to the target state is 1 and all other components are set to 0.

2) Joint Inversion and Acoustic RNN Training: The architectures of inversion and acoustic RNN in the joint training experiment are the same as in baseline system. The difference is that the two models are trained jointly as illustrated in Fig. 2.

In the first series of experiments, we initialized weights of the RNNs randomly. However, since the number of model parameters has doubled, finding a good initialization strategy is essential for the success of the training. Here, we use a pretraining strategy to help the network to start from a good position. In this case, the weights of the network are initialized with the weights from well trained inversion model and acoustic model of the baseline system. This initialization reduced 2 to 3 times the number of iterations necessary to train the models.

During training, the joint loss function parameter λ was varied from 0 to 0.9 in steps of 0.1. As we explained above, $\lambda = 1.0$ is meaningless with respect to the training goal.

D. Acoustic RNN Training using Generalized Distillation

In a similar way to our previous work [20], we applied the Generalized Distillation framework, but this time for RNN training. The teacher model in this case is the same as the acoustic RNN used in the articulatory inversion baseline system. However, here it is used only to obtain the soft targets for the student RNN model training which has the same architecture, except for the input layer. It takes only acoustic feature vectors (39 dim).

The two hyper-parameters of the distillation training, the temperature T and the imitation parameter λ were changed as follows. T was set to 1, 2, and 5, and λ was varied from 0 to 1 with steps of 0.2. Note that $\lambda = 0$ reduces the distillation training to conventional training, with the only difference that the hard targets are obtained from the GMM-HMM model trained with both the acoustic and articulatory features. On the other hand, $\lambda = 1$ means that the training is done using only the soft targets.

VI. Results and Analysis

A. Lower and Upper Performance Bounds

Although impractical, it is possible to train and evaluate the system performance using the true articulatory data. This would give us the maximum achievable performance, or in terms of phoneme error rate, the PER lower bound. On the other hand, performance of the system trained on acoustic data only would serve as the PER upper bound. Any PER in between those bounds would show improvement, but the goal is to get as close as possible to the lower PER bound.

Figure 8 shows those bounds for the GMM-HMM and RNN-HMM acoustic models. Previously, we have built an DNN-HMM model with feedforward layers only, and its results are also shown as FNN-HMM.

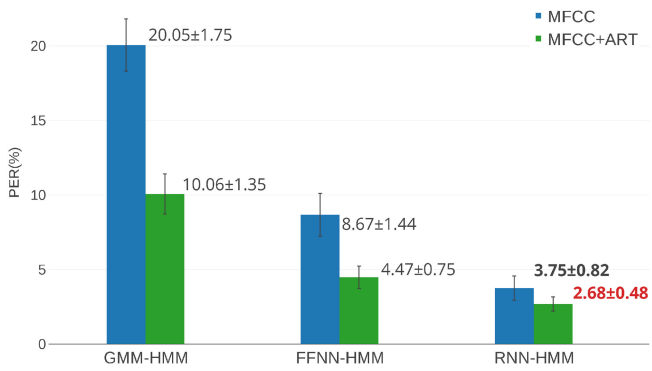


Fig. 8. The PER results of different acoustic models with and without true articulatory (ART) features. The numbers correspond to the upper and lower PER bounds for each model.

TABLE III
Speaker Independent Inversion Results (the 1st and 2nd Derivatives are Excluded)

Inversion & acoustic Model	RMSE	r	PER(%)
FNN	0.632±0.021	0.770±0.029	7.35±1.04
RNN	0.618±0.023	0.923±0.006	**3.15±0.59**

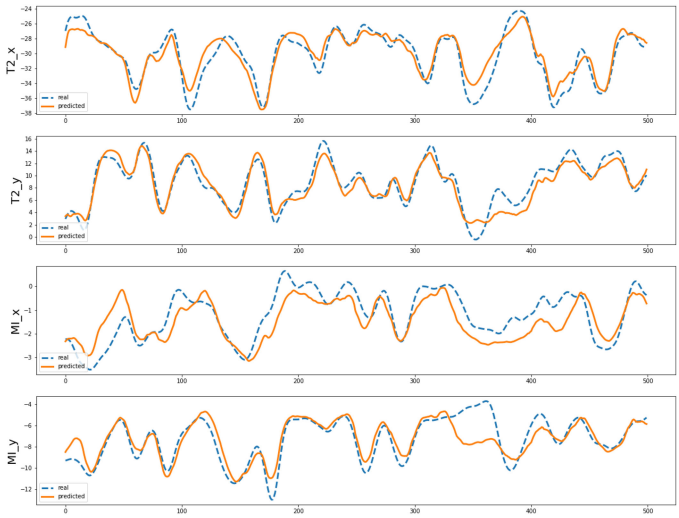


Fig. 9. The predictions of T2_x, T2_y, MI_x, MI_y articulatory movements obtained from the RNN inversion model.

Obviously, models using the articulatory features are always better than those without them. As the model becomes more and more powerful, i.e. GMM→ FNN→ RNN (whose numbers of parameters are about 0.8, 20, 42.4 millions respectively), the gap between the upper and lower bounds reduces significantly.

B. Inversion Baseline Results

First, we investigated how our models perform the acoustic-to-articulatory mapping. Table III shows the inversion results for a 5 hidden layers feedforward network (FNN) with input window size of 17 frames and the RNN (from Fig. 7a) in terms of Root Mean Squared Error (RMSE) and the Pearson correlation coefficient r computed using the true articulatory data. In Fig. 9 plots of predicted trajectories for several articulatory features

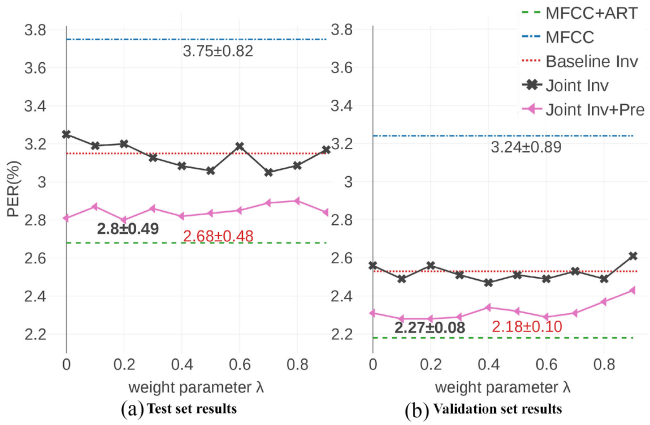


Fig. 10. Results of the joint inversion and acoustic model training. $\lambda = 0$ corresponds to the case when the acoustic model uses MFCC features only but is trained with targets obtained from the MFCC+ART GMM-HMM.

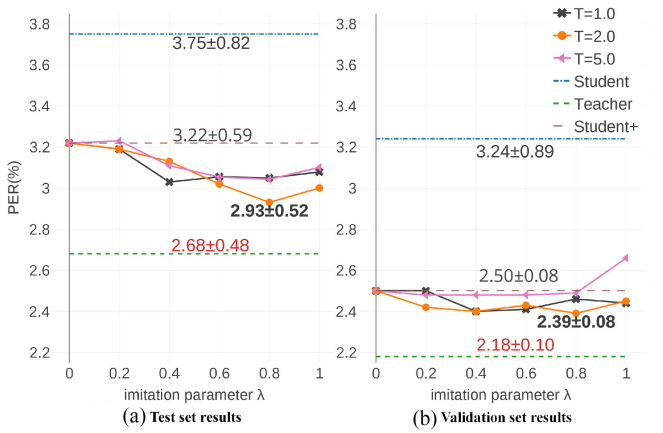


Fig. 11. Results of distillation training. The lower and upper bound for the PER are shown as teacher and student only results. $\lambda = 0$ corresponds to the case when the student is trained using hard targets only. "Student+" corresponds to the case when the acoustic model uses MFCC features only but is trained with targets obtained from the MFCC+ART GMM-HMM.

using the RNN inversion model are given. The predictions from the RNN inversion model are smooth enough even without any post-processing as mentioned in [25]. This indicates that RNN accounts for the previous and future information quite well. When inversion model predictions are used as articulatory features in the corresponding DNN-HMM acoustic models, clear error reduction is observed as shown in PER(%) column.

C. Joint Inversion and Acoustic Models Training Results

The joint inversion + acoustic model results are summarized in Fig. 10, where Fig. 10-a shows the results for the test set and Fig. 10-b gives the results for the validation set, the blue dashed line and green dashed line represent the upper and lower bounds respectively and the red dashed line is the inversion baseline result. The "Joint Inv" curve shows the results of jointly trained model with different λ. The "Joint Inv+Pre" denotes the results with pretraining, i.e when networks are initialized with the weights from the separately trained inversion and acoustic DNNs as explained in Section V-C2.

When the RNN networks are randomly initialized, the joint training gives slight improvement for several values of λ. However, the effect of the pretraining is obvious.

From the figure we can see that the test set and validation set both achieve best results with $\lambda = 0.2$. Because the validation data was used to tune the DNN parameters and to train the GMM-HMM systems, the results are better than the ones on test set. The best joint training result with $\lambda = 0.2$ is $\mathbf{2.80 \pm 0.49\%}$, which is very close to the lower bound of 2.68%.

D. Generalized Distillation Training Results

The RNN distillation training results in terms of PER are summarized in Fig. 11, Fig. 11-a and Fig. 11-b show the results on test and validation sets respectively. The blue dashed line represents the result of the student when trained alone which corresponds to the upper PER bound. The teacher's result is the lower bound distilled student can achieve. As can be seen, for $T = 2$ and $\lambda = 0.8$, both sets achieve the best results. The distillation result of $\mathbf{2.93 \pm 0.52\%}$ PER is 21.9% better than the result of the student alone. When $\lambda = 0$, the distillation training is reduced to standard training with MFCC features using hard targets provided by the GMM-HMM trained on MFCC+ART vectors, which is already much better than training without any articulatory information. On the other hand, $\lambda = 1$ means the model is trained using the soft targets only and is even better than training with hard targets from the teacher. This suggests that soft targets provide a more informative objective than the hard targets alone.

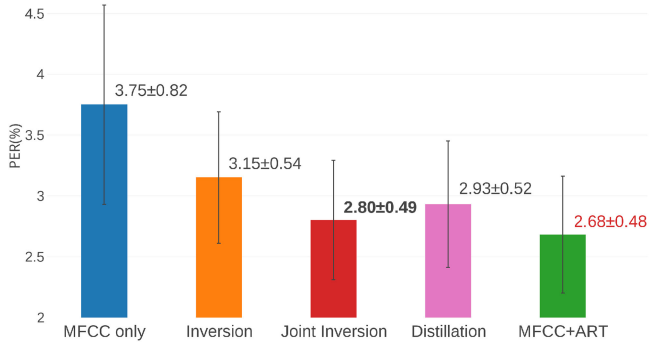


Fig. 12. Performance of different methods and two acoustic baseline models. The "MFCC only" is the PER upper bound result. The "MFCC+ART" is the lower bound.

Finally, we compare the best performances from all the different methods in Fig. 12. Here, the most left and most right bars show the upper and lower performance bounds respectively. The best articulatory information fusion result is $\mathbf{2.80 \pm 0.49\%}$ obtained from the joint inversion and acoustic model pretraining method. Distillation training result is little bit worse, but network size in this case is two times smaller and consequently faster to train and operate.

VII. Discussion

As we mentioned in Section V-A, the XRMB data were collected by asking all the speakers perform the same tasks. This makes the lexical content of the speech data the same for all

TABLE IV
7-FOLD CV RESULTS ON DIFFERENT DATA SETS WITH DIFFERENT LANGUAGE MODELS. RESULTS ARE SHOWN FOR MFCC / MFCC+ART FEATURES

Language Model	Train	Test
GMM acoustic model		
NO LM	19.3 / 7.44	30.9 / 11.1
TIMIT LM	17.1 / 7.67	25.1 / 10.3
XRMB LM	13.3 / 6.35	19.5 / 8.92
FNN acoustic model		
NO LM	2.90 / 2.16	9.82 / 5.12
TIMIT LM	2.75 / 2.04	9.54 / 4.82
XRMB LM	2.67 / 1.88	9.06 / 4.50
RNN acoustic model		
NO LM	1.46 / 1.17	4.23 / 2.99
TIMIT LM	1.38 / 1.11	4.02 / 2.85
XRMB LM	1.33 / 0.97	3.87 / 2.66

the speakers. While the focus of this study is the acoustic and articulatory information fusion, we cannot ignore the fact that lexically the training and test data are the same. With respect to the phoneme language model, this would mean that it is a closed set LM and may have a boosting effect on all the results. In addition, the RNNs input consists of full utterances, so they may learn not only the acoustic dependencies, but the linguistic ones as well, which in turn can lead to biased performance. To check this hypothesis we performed a series of additional experiments.

A. The Closed Set Language Model Effect

To explore the effect of the LM on our results, we did tests without LM as well as with a LM trained on the TIMIT database transcriptions which can be considered as a "general purpose" LM for this task. The results of the upper and lower PER bounds for the GMM, FNN and RNN acoustic models are summarized in Table IV.[1]

Table IV shows that the closed set XRMB LM has big effect on the GMM model performance, but less effect on the DNN acoustic models. Furthermore, even without LM their performance is quite good. This suggests that DNN may have learned some lexical information as well.

B. The Lexical Content Effect on DNN Training

Although the utterances in the test set are from different speakers, they contain words and word sequences seen in the training set. For the neural network based acoustic models this could be significant since the input context in NNs is much larger (the whole utterance in RNNs) and the long span dependencies learned during training to some extend would match those in the test data.

To check this assumption, we repeated all joint inversion and distillation training experiments using an updated setting. This time we split the data into seven folds in terms of both speaker and sentence ids, so we got a two dimensional split with 49 sets as shown in Fig. 13 and we used the 7 sets on the diagonal that are unique in both speakers and sentences for testing. All

[1]In these experiments, we excluded 5 utterances (per speaker) with the same lexical content across the speakers, so the results are slightly different from those in Fig 8.

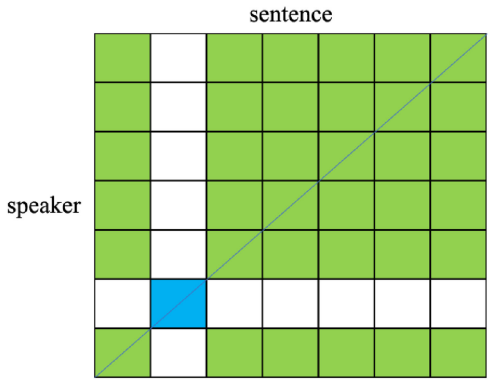


Fig. 13. Train (green) and test (blue) datasets split for the second fold. Similarly, for other folds diagonal boxes data are used for testing.

TABLE V
DETAILS OF THE TRAIN, VALIDATION, AND TEST SETS WHEN UTTERANCES WITH THE SAME LEXICAL CONTENT ARE REMOVED. THE NUMBER IN ( ) IS THE PERCENTAGE OF THE AMOUNT FROM TABLE I

	Train	Test	Validation	Total
Sentences	1394 (53)	66 (13)	155 (53)	1515 (44)
Words	14532 (59)	691 (17)	1615 (64)	16838 (54)
Phonemes	41393 (48)	1945 (15)	4599 (56)	47937 (45)
Hours	1:05:12 (49)	0:02:59 (13)	0:07:15 (51)	1:15:26 (45)

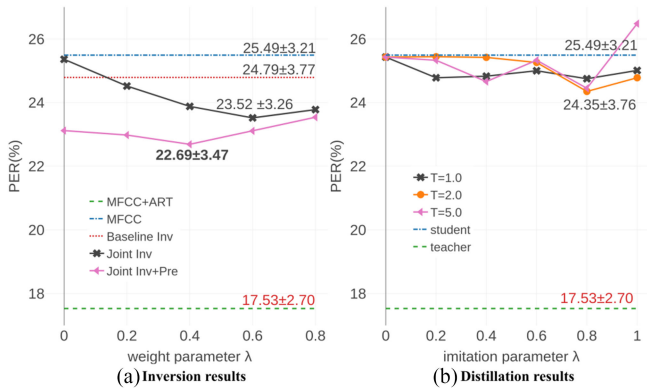


Fig. 14. 7-fold CV results when utterances with the same lexical content are removed. a) PER of the inversion methods, b) PER of the distillation training.

sets from the rows and columns other than those of the test set are used for training. This reduced the amount of data by more than half. Table V summarizes the details about the new dataset. We used the same DNN hyperparameters and XRMB language model.

The RNN-HMM inversion and distillation results are summarized in Fig. 14 and results for all usable systems are summarized in Fig. 15. As can be seen, both proposed methods work in this case as well. The absolute values of the PERs, however, are about ten times higher. Since the presence of the same lexical material in both train and test data and both the acoustic and inversion models are better suited for such test data, the results show less improvement using the proposed methods when utterances with the same lexical content are removed. Nevertheless, the same performance pattern can be observed in this case: the pretrained joint inversion is the best; the joint inversion is better than the distillation training, which in turn is better than the standard inversion. The optimized

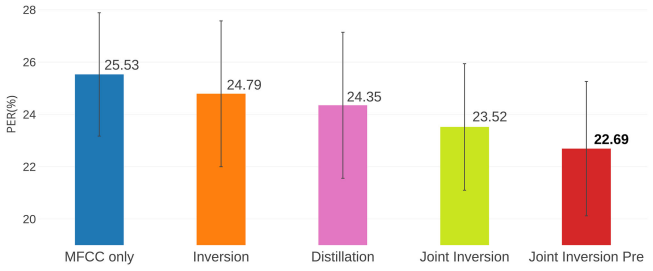


Fig. 15. The results with 95% confidence intervals for all systems that can be used in practice after removing lexical content effect.

hyperparameters for the distillation training still are $T = 2.0$ and $\lambda = 0.8$, while the optimized hyperparameter for the joint inversion is $\lambda = 0.4$. Although $\lambda = 0$ means no articulatory information is integrated in the joint inversion, it is introduced by the pretraining, which is the reason that the pretrained joint inversion in this case is still better than MFCC only.

It is difficult to directly compare our results with results from other studies because the experimental conditions vary significantly. The closest experimental settings are the ones reported in [27] and [43] where DCCA method showed significant improvements.

VIII. Conclusion

In this work, we proposed two methods to integrate articulatory information into ASR systems. One method utilizes the Generalized Distillation framework to build a biGRU-RNN based acoustic model which is trained with the guidance of the soft targets from a teacher biGRU-RNN learned from "rich" data which include articulatory features. The other method combines the inversion model and acoustic model into a single neural network which is trained jointly. When properly initialized, it achieves significant improvements.

The main findings of this study are:

1) Using deep RNNs as acoustic and inversion models provides big performance boost compared to the deep FNNs.
2) An RNN acoustic model trained using generalized distillation framework leads to up to 21.9% PER reduction having the same number of parameters as standard MFCC AM.
3) The PER is reduced by 25.3% using the joint inversion training strategy at the expense of increasing the size of the neural network.
4) The long term dependency learning capabilities of the RNNs are powerful enough to learn not only the temporal acoustic but also lexical information. As our experiments showed, this however may lead to biased results when the data set is rather small and the train and test data are lexically similar.

In the future work, we are going to investigate how much reduction in the network size is possible by the joint inversion training. We also plan to experiment with bigger databases which don't provide articulatory measurements and try to integrate the available articulatory data based on our joint training approach.

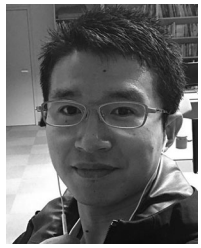
Acknowledgment

The authors would like to thank J. Westbury of the University of Wisconsin for sharing the x-ray microbeam database. They also thank the editor and the anonymous reviewers for helping to improve the quality of this paper.

References

[1] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *J. Acoust. Soc. Am.*, vol. 121, no. 2, pp. 723–742, 2007.
[2] K. Markov, J. Dang, and S. Nakamura, "Integration of articulatory and spectrum features based on the hybrid HMM/BN modeling framework," *Speech Commun.*, vol. 48, no. 2, pp. 161–175, 2006.
[3] K. Livescu *et al.*, "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 jhu summer workshop," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2007, vol. 4, pp. IV-621–IV-624.
[4] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Commun.*, vol. 37, no. 3, pp. 303–319, 2002.
[5] J. Frankel, K. Richmond, S. King, and P. Taylor, "An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces," in *Proc. 6th Int. Conf. Spoken Lang. Process.*, 2000, vol. 4, pp. 254–257.
[6] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, Univ. Edinburgh, Edinburgh, U.K., 2002.
[7] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory hmm," *IEEE Signal Process. Lett.*, vol. 15, pp. 245–248, Feb. 2008.
[8] T. Toda, A. W. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with gaussian mixture model," in *Proc. 8th Int. Conf. Spoken Lang. Process.*, 2004, pp. 1129–1132.
[9] K. Richmond, "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion," in *Proc. Int. Conf. Nonlinear Speech Process.*, 2007, pp. 263–272.
[10] R. Arora and K. Livescu, "Multi-view cca-based acoustic features for phonetic recognition across speakers and domains," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7135–7139.
[11] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4590–4594.
[12] B. Uria, S. Renals, and K. Richmond, "A deep neural network for acoustic-articulatory speech inversion," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011.
[13] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion." in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 867–870.
[14] V. Mitra, G. Sivaraman, H. Nam, C. Espy-Wilson, and E. Saltzman, "Articulatory features from deep neural networks and their role in speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 3017–3021.
[15] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "Integrating articulatory data in deep neural network-based acoustic modeling," *Comput. Speech Lang.*, vol. 36, pp. 173–195, 2016.
[16] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6645–6649.
[17] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, arXiv preprint arXiv:1503.02531.
[18] V. Vapnik and R. Izmailov, "Learning using privileged information: Similarity control and knowledge transfer," *J. Mach. Learn. Res.*, vol. 16, pp. 2023–2049, 2015.
[19] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, "Unifying distillation and privileged information," in *Proc. Int. Conf. Learn. Representations*, 2016.
[20] J. Yu, K. Markov, and T. Matsui, "Articulatory and spectrum features integration using generalized distillation framework," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process.*, 2016, pp. 1–6.
[21] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5900–5904.

[22] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech Commun.*, vol. 50, no. 3, pp. 215–227, 2008.
[23] A. Toutios and K. G. Margaritis, "A support vector approach to the acoustic-to-articulatory mapping," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2005, pp. 3221–3224.
[24] S. Hiroya and M. Honda, "Estimation of articulatory movements from speech acoustics using an hmm-based speech production model," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 2, pp. 175–185, Mar. 2004.
[25] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4450–4454.
[26] P. Zhu, L. Xie, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *Proc. 16th Annu. Conf. Int. Speech Commun. Association*, 2015.
[27] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.
[28] G. Zweig and S. Russell, "Speech recognition with dynamic Bayesian networks," in *Proc. 15th National/10th Conf. Artif. Intel./Innovative Appl. Artif. Intell.*, 1998, pp. 173–180.
[29] K. Livescu, J. R. Glass, and J. A. Bilmes, "Hidden feature models for speech recognition using dynamic Bayesian networks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2003.
[30] V. Mitra, H. Nam, and C. Y. Espy-Wilson, "Robust speech recognition using articulatory gestures in a dynamic Bayesian network framework," in *Proc. IEEE Workshop Autom. Speech Recog. Understanding*, 2011, pp. 131–136.
[31] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
[32] G. Hinton *et al.*, "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.
[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.
[34] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[36] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. NIPS Workshop Deep Learn.*, 2014.
[37] J. Westbury, "X-ray microbeam speech production database user's handbook. 1994," Waisman Center, Univ. Wisconsin, Madison, WI, USA, pp. 1–100, 1994.
[38] W. Wang, R. Arora, and K. Livescu, "Reconstruction of articulatory measurements with smoothed low-rank matrix completion," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 54–59.
[39] J. Yuan and M. Liberman, "Speaker identification on the scotus corpus," *J. Acoust. Soc. Am.*, vol. 123, no. 5, 2008, Art. no. 3878.
[40] "Carnegie mellon university open-source grapheme-to-phoneme dictionary," 2015. [Online]. Available: http://svn.code.sf.net/p/cmusphinx/code/trunk/cmudict/cmudict-0.7b.
[41] W. Chan and I. Lane, "Deep recurrent neural networks for acoustic modelling," 2015, arXiv preprint arXiv:1504.01482.
[42] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," 2013, arXiv preprint arXiv:1312.6120.
[43] Q. Tang, W. Wang, and K. Livescu, "Acoustic feature learning using cross-domain articulatory measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4849–4853.



Jianguo Yu received the M.S. degree from the University of Aizu, Aizuwakamatsu, Japan, in 2016, where he is currently working toward the Ph.D. degree. His main research interests include deep learning, multimodal learning, and affective computing.

Konstantin Markov (M'04) received the Ph.D. degree from Toyohashi University of Technology, Toyohashi, Japan, in 1999. After that, he was a Research Scientist with the Advanced Telecommunications Research, Kyoto, Japan, till 2009. He has been a Senior Associate Professor with the University of Aizu, Aizuwakamatsu, Japan, since 2009. His research interests include machine learning, deep learning, and signal processing with applications to speech, music, and natural language processing. He has been a Program Member of various international conferences such as Interspeech, ICASSP, EUSIPCO, and SpeCom as well as Reviewer for several IEEE and Elsevier scientific journals.

Tomoko Matsui (M'91) received the Ph.D. degree from the Computer Science Department, Tokyo Institute of Technology, Tokyo, Japan, in 1997. From 1988 to 2002, she was with NTT, where she worked on speaker and speech recognition. From 1998 to 2002, she was with the Spoken Language Translation Research Laboratory, Advanced Telecommunications Research, Kyoto, Japan, as a Senior Researcher and worked on speech recognition. From January to June 2001, she was an Invited Researcher with the Acoustic and Speech Research Department, Bell Laboratories, Murray Hill, NJ, USA, working on finding effective confidence measures for verifying speech recognition results. She is currently a Professor with the Institute of Statistical Mathematics, Tachikawa, Tokyo, working on statistical modeling for speech and speaker recognition applications. He received the paper award of the Institute of Electronics, Information, and Communication Engineers of Japan in 1993.