Evaluation of Feature-Space Speaker Adaptation for End-to-End Acoustic Models

Natalia Tomashenko, Yannick Estève

Evaluation of Feature-Space Speaker Adaptation for End-to-End Acoustic Models

Natalia Tomashenko[1,2] **and Yannick Estève**[1]
[1] LIUM, University of Le Mans, France
[2] ITMO University, Saint-Petersburg, Russia
{natalia.tomashenko, yannick.esteve}@univ-lemans.fr

Abstract

This paper investigates speaker adaptation techniques for bidirectional long short term memory (BLSTM) recurrent neural network based acoustic models (AMs) trained with the connectionist temporal classification (CTC) objective function. BLSTM-CTC AMs play an important role in end-to-end automatic speech recognition systems. However, there is a lack of research in speaker adaptation algorithms for these models. We explore three different feature-space adaptation approaches for CTC AMs: feature-space maximum linear regression, i-vector based adaptation, and maximum a posteriori adaptation using GMM-derived features. Experimental results on the TED-LIUM corpus demonstrate that speaker adaptation, applied in combination with data augmentation techniques, provides, in an unsupervised adaptation mode, for different test sets, up to 11–20% of relative word error rate reduction over the baseline model built on the raw filter-bank features. In addition, the adaptation behavior is compared for BLSTM-CTC AMs and time-delay neural network AMs trained with the cross-entropy criterion.

Keywords: Speaker adaptation, end-to-end speech recognition, GMM-derived features, deep neural network, acoustic model

1. Introduction

Recently, various neural end-to-end approaches to automatic speech recognition (ASR) have been proposed in the literature (Hannun et al., 2014; Bahdanau et al., 2016; Collobert et al., 2016; Fritz and Burshtein, 2017; Chan et al., 2016; Audhkhasi et al., 2017). End-to-end acoustic models (AMs) (Chorowski et al., 2014; Graves and Jaitly, 2014; Miao et al., 2015; Zhang et al., 2017) attempt to map an acoustic signal to a phoneme or grapheme sequence directly by means of neural network models. They have been developed as an alternative to the traditional hybrid approach based on hidden Markov models coupled to deep neural networks (HMM-DNNs) (Hinton et al., 2012).

Speaker adaptation is an essential component of state-of-the-art hybrid HMM-DNN AMs, and a variety of adaptation methods have been developed for DNNs. They include linear transformations, that can be applied at different levels of the DNN-HMM architecture (Gemello et al., 2006; Seide et al., 2011); regularization techniques, such as L2-prior regularization (Liao, 2013) or Kullback-Leibler divergence regularization (Yu et al., 2013); model-space adaptation (Siniscalchi et al., 2013; Swietojanski and Renals, 2014; Huang et al., 2014); multi-task learning (MTL) (Price et al., 2014; Li et al., 2015; Huang et al., 2015); factorized adaptation (Li et al., 2014); adaptation with speaker codes (Xue et al., 2014); the use of auxiliary features, such as i-vectors (Saon et al., 2013; Senior and Lopez-Moreno, 2014) or GMM-derived (GMMD) features (Tomashenko and Khokhlov, 2014), and many others.

However, the major part of the published works, devoted to end-to-end technology, does not use any speaker adaptation techniques. This lack may be justified by the strong focus of these papers on the neural core of the technology they introduce.

A few papers have offered some preliminary and promising information about the benefits provided by some speaker adaptation techniques to end-to-end AMs. In (Miao et al., 2016), vocal tract length normalization (VTLN) (Lee and Rose, 1996) has been applied to filterbank features, for a neural end-to-end AM training through connectionist temporal classification (CTC), providing 3% of relative word error rate reduction (WERR). Speaker i-vectors, appended to the acoustic features, are used in (Audhkhasi et al., 2017) for training phone and word CTC models. Also features, adapted using feature-space maximum likelihood linear regression (fMLLR), are used to train attention-based RNNs (Chorowski et al., 2014). However in these works (Audhkhasi et al., 2017; Chorowski et al., 2014), no comparison results with the unadapted models are given. Work (Yi et al., 2016) proposes a CTC regularized model adaptation method for the accent adaptation task. Speaker adaptation with speaker codes of RNN-BLSTM AMs is studied in (Huang et al., 2016) for the phone recognition task, where AMs were trained with cross-entropy (CE) criterion, and the adaptation provides about 10% of relative reduction in phone error rate.

The aim of this paper is to explore the efficiency of speaker adaptation for end-to-end ASR systems on the example of CTC-BLSTM AMs (or shortly, CTC AMs). For this purpose we implemented three different speaker adaptation algorithms to this type of AMs and performed an experimental analysis of these methods. Furthermore, a comparative study of the adaptation techniques was conducted for CTC AMs and time-delay neural network (TDNN) AMs trained with traditional frame-wise cross-entropy (CE) criterion.

The rest of the paper is organized as follows. A quick overview of the end-to-end AMs, studied in this paper, is introduced in Section 2. A speaker adaptation technique, based on the use of GMMD features and recently proposed by the authors for DNN-HMM AMs, is presented in Section 3. for CTC AMs. Section 4. describes the experimental results for different adaptation algorithms. Finally, the conclusions are given in Section 5.

2. Review of End-to-End Speech Recognition

One of the first major steps in the direction of end-to-end systems was introduced in (Graves et al., 2013) where, for the phoneme recognition task, a deep BLSTM recurrent neural network (RNN) model was trained to map directly acoustic sequences to phonetics ones. This was done by using the CTC objective function (Graves et al., 2006). The BLSTM-CTC models are used in this study. Alternative approaches to end-to-end ASR include attention mechanism (Chorowski et al., 2014; Bahdanau et al., 2016; Chan et al., 2016), convolutional neural networks (CNNs) trained with CTC loss (Collobert et al., 2016; Zhang et al., 2014; Wang et al., 2017), RNN transducers (Graves et al., 2013) and others.

2.1. Deep Bidirectional LSTMs

RNNs provide a powerful extension of feed-forward DNN models by adding connections between different types of units, including backward connections to previous layers. The use of recurrence over the temporal dimension allows RNNs to model the dynamic temporal behavior of the process.

In order to capture information from the whole input sequence, the bidirectional RNN (BRNN) architecture was proposed (Schuster and Paliwal, 1997). In BRNNs, data are processed in two directions with two hidden layers, which are then input further to the same output layer. As shown in the upper part of Figure 1, for a sequence of input vectors $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$, a recurrent forward hidden layer of a BRNN $\overrightarrow{\mathbf{H}} = \{\overrightarrow{\mathbf{h}}_1, \ldots, \overrightarrow{\mathbf{h}}_T\}$ computes a sequence of hidden outputs for $t = 1, \ldots, T$, and an additional recurrent layer $\overleftarrow{\mathbf{H}} = \{\overleftarrow{\mathbf{h}}_1, \ldots, \overleftarrow{\mathbf{h}}_T\}$ computes the backward sequence of hidden outputs for $t = T, \ldots, 1$:

$$\begin{cases} \overrightarrow{\mathbf{h}}_t = f(\mathbf{W}_{x\overrightarrow{h}}\mathbf{x}_t + \mathbf{W}_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overrightarrow{h}}), \\ \overleftarrow{\mathbf{h}}_t = f(\mathbf{W}_{x\overleftarrow{h}}\mathbf{x}_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overleftarrow{h}}), \\ \mathbf{y}_t = g(\mathbf{W}_{\overrightarrow{h}y}\overrightarrow{\mathbf{h}}_t + \mathbf{W}_{\overleftarrow{h}y}\overleftarrow{\mathbf{h}}_t + \mathbf{b}_y), \end{cases} \quad (1)$$

where $\mathbf{W}_{x\overrightarrow{h}}$, $\mathbf{W}_{x\overleftarrow{h}}$ are the weight matrices connecting inputs to hidden units; $\mathbf{W}_{\overrightarrow{h}\overrightarrow{h}}$, $\mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}$ are the weight matrices connecting hidden units from time $t-1$ to time t; $\mathbf{W}_{\overrightarrow{h}y}$, $\mathbf{W}_{\overleftarrow{h}y}$ are the weight matrices connecting the output layer to the hidden layer; $\mathbf{b}_h$, $\mathbf{b}_y$ are bias vectors for the hidden states and the outputs correspondingly; $f(\cdot)$, $g(\cdot)$ are the hidden and output layer activation functions correspondingly.

State-of-the-art ASR systems have deep architectures with several hidden layers, where the forward and backward hidden outputs at time t are concatenated and this concatenation $\overrightarrow{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t$ is input into the next recurrent layers.

For training RNN models, a back-propagation-through-time (BPTT) learning algorithm is typically used (Werbos, 1990). However, in practice, training RNNs to learn long-term temporal dependencies can be difficult due to the vanishing and exploding gradient problems (Bengio et al., 1994). To avoid the long-term dependency problem, LSTM neural networks were introduced in (Hochreiter and Schmidhuber, 1997). In the end-to-end ASR framework, LSTMs units are used as the structural elements of BRNNs (Miao et al., 2015; Miao et al., 2016; Graves et al., 2013; Sak et al., 2015).

2.2. Connectionist Temporal Classification

In the CTC approach, the alignment between the inputs and target labels is unknown. CTC can be implemented with a softmax output layer using an additional unit for the blank label $\emptyset$. The symbol $\emptyset$ corresponds to no output and is used to estimate the probability of outputting no label at a given time. The network is trained to optimize the total log-probability of all valid label sequences for training data. A set of valid label sequences for an input sequence is defined as the set of all possible label sequences of the input with the target labels in the correct order with repetitions and with label $\emptyset$ allowed between any labels. Targets for CTC training can be computed using finite state transducers (FSTs) (Sak et al., 2015), and the forward-backward algorithm can be used to calculate the CTC loss function.

State transition probability distribution and state priors are not required for CTC approach, in contrast to the hybrid DNN-HMM system. Several types of output units for CTC training have been explored in the literature, such as phones (or graphemes) (Miao et al., 2015), words (Audhkhasi et al., 2017) or grams (Liu et al., 2017). Due to the large number of word outputs in acoustic-to-word CTC models, they require significantly more training data in comparison with traditional ASR systems (Audhkhasi et al., 2017). A maximum a posteriori (MAP) training criterion instead of CTC was used in (Fritz and Burshtein, 2017) to train an end-to-end ASR system.

3. Speaker Adaptation

In this paper we focus on the feature space adaptation techniques for end-to-end acoustic models. Three different types of AM adaptation were explored in this paper: (1) fMLLR (Gales, 1998), (2) adaptation using i-vectors (Senior and Lopez-Moreno, 2014), and (3) MAP (Gauvain and Lee, 1994) adaptation using GMMD features (Tomashenko and Khokhlov, 2014; Tomashenko et al., 2016b). In this section we describe the adaptation approach, which is based on using speaker-adapted GMMD features for training BLSTM-CTC models.

3.1. GMM-Derived Features for BLSTM-CTC Models

The use of log-likelihoods from a GMM model for training a multilayer perceptron (MLP) recognizer was investigated in (Pinto and Hermansky, 2008). Construction of GMMD features for adapting hybrid DNN-HMM AMs was proposed in (Tomashenko and Khokhlov, 2014; Tomashenko and Khokhlov, 2015; Tomashenko et al., 2016a), where it was demonstrated, using MAP and fMLLR adaptation as examples, that this type of features provide a solution for efficient transferring GMM-HMM adaptation algorithms into the DNN framework.

We can train DNN models directly on GMMD features, as it was done in (Tomashenko and Khokhlov, 2014; Tomashenko and Khokhlov, 2015; Tomashenko et al.,

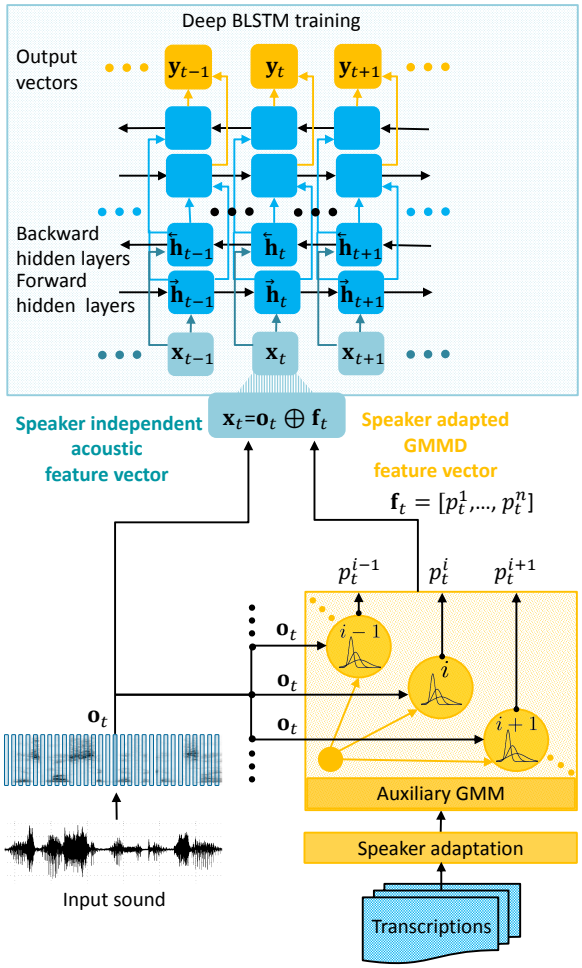


Figure 1: Speaker adaptive training for the BLSTM AM using GMMD features.

2016c), or use them in combination with other conventional features. In this paper we present incorporation of the adapted GMMD features into the recipe for training sequence-to-sequence AMs.

The scheme for speaker adaptive training (SAT) of AMs models with GMM-based adaptation framework is shown in Figure 1. An auxiliary monophone GMM-HMM model is used to transform acoustic feature vectors into log-likelihoods vectors. At this step, speaker adaptation of the auxiliary speaker-independent (SI) GMM-HMM model is performed for each speaker in the training corpus using correct transcriptions and a new speaker-adapted (SA) GMM-HMM model is created in order to obtain SA GMMD features.

For a given acoustic feature vector, a new GMM-derived feature vector is obtained by calculating log-likelihoods across all the states of the auxiliary GMM model on the given vector. Suppose $\mathbf{o}_t$ is the acoustic feature vector at time t, then the new GMM-derived feature vector $\mathbf{f}_t$ is calculated as follows:

$$\mathbf{f}_t = [p_t^1, \ldots, p_t^n], \tag{2}$$

where n is the number of states in the auxiliary GMM-HMM model,

$$p_t^i = \log\left(P(\mathbf{o}_t \mid s_t = i)\right) \tag{3}$$

is the log-likelihood estimated using the GMM-HMM. Here s_t denotes the state index at time t.

The adapted GMMD feature vector $\mathbf{f}_t$ is concatenated with the original vector $\mathbf{o}_t$ to obtain vector $\mathbf{x}_t$. These features are used as the input for training a SAT BLSTM-CTC AM. The proposed approach can be considered a feature space transformation technique with respect to BLSTM-CTC AM trained on GMMD features.

3.2. MAP Adaptation

In this work we use the MAP adaptation algorithm (Gauvain and Lee, 1994) in order to adapt the SI GMM model. Speaker adaptation of a DNN-HMM model built on GMMD features is performed through the MAP adaptation of the auxiliary GMM model, which is used for calculating GMMD features. Let m denote an index of a Gaussian in the SI acoustic model (AM), and $\boldsymbol{\mu}_m$ the mean of this Gaussian. Then the MAP estimation of the mean vector is

$$\widehat{\boldsymbol{\mu}}_m = \frac{\tau \boldsymbol{\mu}_m + \sum_t \gamma_m(t)\mathbf{o}_t}{\tau + \sum_t \gamma_m(t)}, \tag{4}$$

where τ is the parameter that controls the balance between the maximum likelihood estimate of the mean and its prior value; $\gamma_m(t)$ is the posterior probability of Gaussian component m at time t.

4. Experiments

4.1. Data Sets

The experiments were conducted on the TED-LIUM corpus (Rousseau et al., 2014). We used the last (second) release of this corpus. This publicly available data set contains 1495 TED talks that amount to 207 hours (141 hours of male, 66 hours of female) speech data from 1242 speakers, 16kHz. For experiments with SAT and adaptation we removed from the original corpus data for those speakers, who had less than 5 minutes of data, and from the rest of the corpus we made four data sets: training set, development set and two test sets. Characteristics of the obtained data sets are given in Table 1. For evaluation a 4-gram lan-

Characteristic		Data set			
		Train	Dev.	Test_1	Test_2
Duration, hours	Total	171.66	3.49	3.49	4.90
	Male	120.50	1.76	1.76	3.51
	Female	51.15	1.73	1.73	1.39
Duration per speaker, minutes	Mean	10.0	15.0	15.0	21.0
	Min.	5.0	14.4	14.4	18.3
	Max.	18.3	15.4	15.4	24.9
Number of speakers	Total	1029	14	14	14
	Male	710	7	7	10
	Female	319	7	7	4
Number of words	Total	-	36672	35555	51452

Table 1: Data sets statistics.

guage model (LM) with 152K word vocabulary was used. The LM is similar to the "small" one, which is currently used in the Kaldi *tedlium s5_r2* recipe. The only difference is that we modified a little a training set, removing from it those data, that present in our test and development sets.

4.2. Baseline Systems

We used the open-source Kaldi toolkit (Povey et al., 2011) and the Eesen system (Miao et al., 2015) for the experiments presented in this paper. Three baseline SI AMs were trained using the Eesen system in a similar manner, and differ only in the front-end processing. The following three type of features were used:

1. *fbanks* $\oplus\Delta\oplus\Delta\Delta$ *(dimension* $= 120$*)*: 40-dimensional filterbank features appended with their first and second-order temporal derivatives;
2. *high-resolution MFCC features (dimension* $= 40$*)*: features extracted without dimensionality reduction, keeping all 40 cepstra;
3. *bottleneck (BN) features (dimension* $= 40$*)*.

The first type of features is the same, as proposed in the original Eesen recipe for the TED-LIUM corpus. For the AMs with the two other types of features, also the two types of data augmentation strategies were applied for the speech training data: speed perturbation (with factors 0.9, 1.0, 1.1) and volume perturbation, as in (Peddinti et al., 2015).
The first baseline AM was trained as described in (Miao et al., 2015) with the CTC criterion and the deep BLSTM architecture. The BLSTM network contains five bidirectional LSTM layers with 320 memory cells in each forward and backward sub-layer. The input features were normalized with per-speaker mean subtraction and variance normalization. The output layer is a 41-dimensional softmax layer with the units, corresponding to 39 context-independent phones, 1 noise model and 1 blank symbol.
The third SI AM was trained on BN features (Grézl et al., 2007). A DNN model for extraction 40-dimensional BN features was trained with the following topology: one 440-dimensional input layer; four hidden layers (HLs), where the third HL was a BN layer with 40 neurons and other three HLs were 1500-dimensional; the output layer was 4052-dimensional. The input features for training this BN extractor were 440-dimensional (40×11): 40-dimensional high-resolution MFCCs spliced across 11 neighboring frames (± 5).

4.3. Adapted Models

Three types of AM adaptation were empirically explored in this section: fMLLR, adaptation using i-vectors, and MAP adaptation using GMMD features. For all the adapted AMs the same data augmentation strategies were applied during the training, as for the SI ones. All the SAT models were trained with the same neural network topology (except for the input layer) and training criterion, as described in Section 4.2. for SI AMs. The six SAT AMs were trained on the following features:

4. *MFCC* $\oplus$ *i-vectors (dimension* $= 140$*)*;
5. *BN* $\oplus$ *i-vectors (dimension* $= 140$*)*;
6. *BN with fMLLR (dimension* $= 40$*)*;
7. *MFCC* $\oplus$ *GMMD (dimension* $= 167$*)*;
8. *BN* $\oplus$ *GMMD (dimension* $= 167$*)*;
9. *BN with fMLLR* $\oplus$ *GMMD (dimension* $= 167$*)*.

For the AMs trained on features #4 and #5, the 100-dimensional on-line i-vectors were calculated as in (Peddinti et al., 2015), and the statistic for i-vectors was updated every two utterances during the training.
For AMs #7–#9 we used BN features to train the auxiliary GMM model for GMMD feature extraction. The speaker-adapted GMMD features were obtained in the same way as described in Section 3. Parameter τ in MAP adaptation (see Formula (4)) was set equal to 5 for both acoustic model training and decoding.

4.4. Adaptation Results for CTC AMs

Unless explicitly stated otherwise, the adaptation experiments were conducted in an unsupervised mode on the test data using transcripts from the first decoding pass obtained by the best baseline SI model.

#	Features	WER,%		
		Dev.	Test_1	Test_2
1	fbanks $\oplus\Delta\oplus\Delta\Delta$	14.57	11.71	15.29
2	MFCC	13.21	11.16	14.15
3	BN	13.63	11.84	15.06
4	MFCC $\oplus$ i-vectors	12.92	10.45	14.09
5	BN $\oplus$ i-vectors	13.47	11.37	14.31
6	BN with fMLLR	12.45	10.96	13.79
7	MFCC $\oplus$ GMMD	11.95	10.20	14.04
8	BN $\oplus$ GMMD	11.66	10.14	13.88
9	BN with fMLLR $\oplus$ GMMD	**11.63**	**9.91**	**13.58**
10	BN $\oplus$ GMMD*	11.67	10.11	13.70
11	BN with fMLLR $\oplus$ GMMD*	**11.41**	9.93	**13.47**

Table 2: Summary of adaptation results for CTC AMs. GMMD* correspond to the GMMD features, obtained using the first decoding pass by the SAT AM (by default, in all other experiments, the SI model is used instead)

#	Features	WER,%		
		Dev.	Test_1	Test_2
2	MFCC	13.69	11.34	14.38
3	BN	12.32	10.48	14.00
4	MFCC $\oplus$ i-vectors	11.63	9.62	13.28
5	BN $\oplus$ i-vectors	11.62	9.75	13.30
6	BN with fMLLR	**10.70**	**9.28**	**12.84**
7	MFCC $\oplus$ GMMD	11.30	9.75	13.74
8	BN $\oplus$ GMMD	11.07	9.75	13.55
9	BN with fMLLR $\oplus$ GMMD	10.92	9.54	13.27
10	BN $\oplus$ GMMD*	10.29	9.20	13.04
11	BN with fMLLR $\oplus$ GMMD*	**10.15**	**9.06**	**12.84**

Table 3: Summary of adaptation results for TDNN AMs.

The performance results in terms of word error rate (WER) for SI and SAT AMs models are presented in Table 2. The first three lines of the table (#1–#3) correspond to the baseline SI AMs, which were trained as described in Section 4.2., where the very first line represents the Eesen baseline (Miao et al., 2015). The next six lines (#4–#9) show the results for the adapted models. The numeration in Table 2 coincides with the numeration in Sections 4.2. and 4.3..

The two last lines of the table (#10 and #11) are obtained with the same AMs as the lines #8 and #9 correspondingly, but for the extraction of GMMD-adapted features in #10 and #11 (marked with the "*" in Table 2, and further in Figure 2 and Tables 3, 4), we used the transcriptions from the adapted model #6). Notice, that for all other tests (#7–#9) we used transcriptions from the SI model #2.

The best result among all the systems #1–#9 is obtained by system #9, which corresponds to the use of MAP-adapted GMMD features appended with fMLLR-adapted BN features. It can be only slightly improved (#11) for two sets by using the adapted model in the first decoding pass (for GMMD*). Among all the adaptation methods, applied separately (#4–#8), the MAP adaptation of GMMD features shows the best performance with both BN and MFCC features.

4.5. Comparison of Adaptation Behavior for BLSTM-CTC and TDNN AMs.

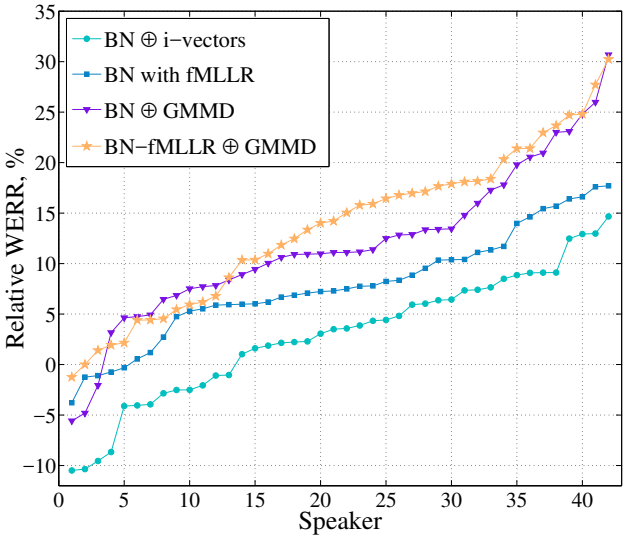
In this series of experiments we aim to compare the adaptation behavior of SAT CTC models with the different type of neural network AMs. For this purpose we chose a TDNN model topology, because such models are shown to achieve the best result in many state-of-the ASR systems (Peddinti et al., 2015). These AMs were trained the with the CE criterion.

We built the same set of SI and SAT AMs, as before for CTC-AMs (see Sections 4.2. and 4.3.), except for #1. All SI ans SAT TDNN models were trained in a similar way and have the same model topology. They differ only in the type of the input features.

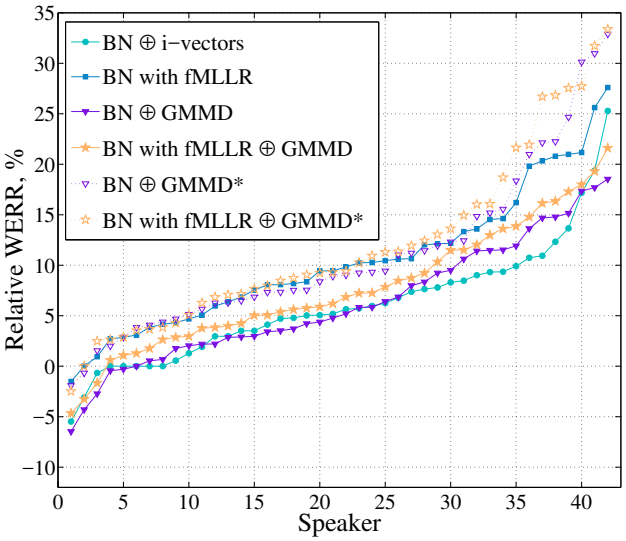
The topology of the TDNN models was similar to the one described in (Peddinti et al., 2015), except for the number of hidden layers and slightly different subsequences of splicing and sub-sampling indexes. The temporal context was $[t-16, t+12]$ and the splicing indexes used here were $[-2, 2]$, $\{-1, 2\}$, $\{-3, 3\}$, $\{-7, 2\}$, $\{0\}$, $\{0\}$. This model had 850-dimensional hidden layers with rectified linear units (ReLU) (Dahl et al., 2013) activation functions and about 4000-dimensional output layer.

The results for TDNN AMs are reported in Table 3. Also Figure 2 presents the comparison of different adaptation algorithms in terms of relative WERR for the speakers from test and development datasets for BLSTM-CTC (Figure 2a) and TDNN (Figure 2b) AMs. The relative WERR is calculated with respect to the SI AMs trained on BN features. For TDNN AMs we also added in Figure 2b the results obtained with the use of SAT AMs for the first decoding pass, because they provide a consistent additional improvement in performance in comparison with the use of SI AMs.

Table 4 shows relative WERRs for BLSTM-CTC and



(a) CTC



(b) TDNN

Figure 2: Relative WER reduction (WERR) for the speakers from test and development datasets for different adaptation algorithms with respect to the SI AMs, trained on BN features (#3). Results are ordered in ascending WERR order for each AM.

TDNN AMs in comparison with the best corresponding SI AMs (#2 for CTC and #3 for TDNN). We can see, that the optimal choice of features depends on the AM architecture. For SI AMs, BNs have appeared to perform better than MFCCs for TDNN AMs, but for CTC AMs the situation is reversed. Also for SAT CTC and SAT TDNN AMs the ranking of the systems by the WER is different.

5. Conclusions

This paper has explored how the end-to-end ASR technology can benefit from speaker adaptation and demonstrated that speaker adaptation has remained an essential mecha-

#	Features	**CTC**: rel. WERR,%			**TDNN**: rel. WERR,%		
		Dev.	Test_1	Test_2	Dev.	Test_1	Test_2
4	MFCC $\oplus$ i-vectors	2.2	6.4	0.4	5.6	8.2	5.1
5	BN $\oplus$ i-vectors	-2.0	-1.9	-1.1	5.7	7.0	5.0
6	BN with fMLLR	5.8	1.8	2.5	13.2	11.5	8.3
7	MFCC $\oplus$ GMMD	9.5	8.6	0.8	8.3	7.0	1.9
8	BN $\oplus$ GMMD	11.7	9.1	1.9	10.2	7.0	3.2
9	BN with fMLLR $\oplus$ GMMD	12.0	**11.2**	4.0	11.4	9.0	5.2
10	BN $\oplus$ GMMD*	11.7	9.4	3.2	16.5	12.2	6.9
11	BN with fMLLR $\oplus$ GMMD*	**13.6**	11.0	**4.8**	**17.6**	**13.6**	**8.3**

Table 4: Relative WER reduction (WERR) for adapted BLSTM-CTC and TDNN AMs in comparison with the best SI AMs for each AM type (#2 for CTC and #3 for TDNN). Relative WERR values are calculated based on the results from Tables 2 and 3.

nism for improving the performance of an ASR system in the new end-to-end speech recognition paradigm. Experimental results on the TED-LIUM corpus showed that in an unsupervised adaptation mode, the adaptation and data augmentation techniques can provide approximately a 10–20% relative WERR on different adaptation sets, compared to the SI BLSTM-CTC system built on filter-bank features. The best results, for BLSTM-CTC AMs, in average, were obtained using GMM-derived features and MAP adaptation, which can be further slightly improved by combination with fMLLR adaptation technique.

We found out, that the type of the neural network AM architecture can differently influence the adaptation performance. The comparison with the TDNN-CE AMs showed that for these models, in contradiction to BLSTM-CTC AMs, MAP adaptation using GMMD features outperforms fMLLR only when it uses SAT model in the first decoding pass to obtain transcriptions for adaptation.

Also the obtained results allow us to compare TDNN-CE and BLSTM-CTC AMs in the realistic conditions, when the speaker adaptation is applied, which is important because usually end-to-end and hybrid AMs are compared on incomplete unadapted systems. The best SI TDNN-CE AM outperforms the best SI BLSTM-CTC AM on 1–7% of relative WER reduction for different test sets. For the best SAT AMs this gap in WER for TDNN-CE and BLSTM-CTC AMs increases and reaches 5–13% of relative WER reduction.

6. Bibliographical References

Audhkhasi, K., Ramabhadran, B., Saon, G., Picheny, M., and Nahamoo, D. (2017). Direct acoustics-to-word models for english conversational speech recognition. *arXiv preprint arXiv:1703.07754*.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. (2016). End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE.

Chorowski, J., Bahdanau, D., Cho, K., and Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*.

Collobert, R., Puhrsch, C., and Synnaeve, G. (2016). Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

Dahl, G. E., Sainath, T. N., and Hinton, G. E. (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613, May.

Fritz, L. and Burshtein, D. (2017). End-to-end map training of a hybrid hmm-dnn model. *arXiv preprint arXiv:1703.10356*.

Gales, M. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech and language*, 12(2):75–98.

Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. *IEEE Trans. Speech and Audio Proc.*, 2:291–298.

Gemello, R., Mana, F., Scanzio, S., Laface, P., and De Mori, R. (2006). Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training. In *Proc. ICASSP*, pages 1189–1192.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, volume 14, pages 1764–1772.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM.

Graves, A., Mohamed, A.-R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Grézl, F., Karafiát, M., Kontár, S., and Cernocky, J. (2007).

Probabilistic and bottle-neck features for LVCSR of meetings. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–757. IEEE.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, Z., Li, J., Siniscalchi, S. M., Chen, I.-F., Weng, C., and Lee, C.-H. (2014). Feature space maximum a posteriori linear regression for adaptation of deep neural networks. In *Proc. INTERSPEECH*, pages 2992–2996.

Huang, Z., Li, J., Siniscalchi, S. M., Chen, I.-F., Wu, J., and Lee, C.-H. (2015). Rapid adaptation for deep neural networks through multi-task learning. In *Proc. INTERSPEECH*, pages 2329–9290.

Huang, Z., Tang, J., Xue, S., and Dai, L. (2016). Speaker adaptation of RNN-BLSTM for speech recognition based on speaker code. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5305–5309. IEEE.

Lee, L. and Rose, R. C. (1996). Speaker normalization using efficient frequency warping procedures. In *Proc. ICASSP*, volume 1, pages 353–356. IEEE.

Li, J., Huang, J.-T., and Gong, Y. (2014). Factorized adaptation for deep neural network. In *Proc. ICASSP*, pages 5537–5541. IEEE.

Li, S., Lu, X., Akita, Y., and Kawahara, T. (2015). Ensemble speaker modeling using speaker adaptive training deep neural network for speaker adaptation. In *Proc. INTERSPEECH*, pages 2892–2896.

Liao, H. (2013). Speaker adaptation of context dependent deep neural networks. In *Proc. ICASSP*, pages 7947–7951. IEEE.

Liu, H., Zhu, Z., Li, X., and Satheesh, S. (2017). Gram-CTC: Automatic unit selection and target decomposition for sequence labelling. *arXiv preprint arXiv:1703.00096*.

Miao, Y., Gowayyed, M., and Metze, F. (2015). Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 167–174. IEEE.

Miao, Y., Gowayyed, M., Na, X., Ko, T., Metze, F., and Waibel, A. (2016). An empirical exploration of CTC acoustic models. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 2623–2627. IEEE.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *INTERSPEECH*, pages 3214–3218.

Pinto, J. P. and Hermansky, H. (2008). Combining evidence from a generative and a discriminative model in phoneme recognition. Technical report, IDIAP.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The Kaldi speech recognition toolkit. In *Proc. ASRU*.

Price, R., Iso, K., and Shinoda, K. (2014). Speaker adaptation of deep neural networks using a hierarchy of output layers. In *Proc. SLT*, pages 153–158. IEEE.

Rousseau, A., Deléglise, P., and Estève, Y. (2014). Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In *Proc. LREC*, pages 3935–3939.

Sak, H., Senior, A., Rao, K., Irsoy, O., Graves, A., Beaufays, F., and Schalkwyk, J. (2015). Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4280–4284. IEEE.

Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *ASRU*, pages 55–59.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Seide, F., Li, G., Chen, X., and Yu, D. (2011). Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Proc. ASRU*, pages 24–29. IEEE.

Senior, A. and Lopez-Moreno, I. (2014). Improving DNN speaker independence with i-vector inputs. In *Proc. ICASSP*, pages 225–229.

Siniscalchi, S. M., Li, J., and Lee, C.-H. (2013). Hermitian polynomial for speaker adaptation of connectionist speech recognition systems. *Audio, Speech, and Language Processing, IEEE Trans. on*, 21(10):2152–2161.

Swietojanski, P. and Renals, S. (2014). Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Proc. SLT*, pages 171–176. IEEE.

Tomashenko, N. and Khokhlov, Y. (2014). Speaker adaptation of context dependent deep neural networks based on MAP-adaptation and GMM-derived feature processing. In *Proc. INTERSPEECH*, pages 2997–3001.

Tomashenko, N. and Khokhlov, Y. (2015). GMM-derived features for effective unsupervised adaptation of deep neural network acoustic models. In *Proc. INTERSPEECH*, pages 2882–2886.

Tomashenko, N., Khokhlov, Y., and Esteve, Y. (2016a). On the use of Gaussian mixture model framework to improve speaker adaptation of deep neural network acoustic models. In *Proc. INTERSPEECH*, pages 3788–3792.

Tomashenko, N., Khokhlov, Y., and Esteve, Y. (2016b). A new perspective on combining GMM and DNN frameworks for speaker adaptation. In *Statistical Language and Speech Processing: 4th International Confer-*

ence, SLSP 2016, Pilsen, Czech Republic, October 11-12, 2016, Proceedings, volume 9918, pages 120–132. Springer.

Tomashenko, N., Khokhlov, Y., Larcher, A., and Estève, Y. (2016c). Exploring GMM-derived features for unsupervised adaptation of deep neural network acoustic models. In *International Conference on Speech and Computer*, pages 304–311. Springer.

Wang, Y., Deng, X., Pu, S., and Huang, Z. (2017). Residual convolutional CTC networks for automatic speech recognition. *arXiv preprint arXiv:1702.07793*.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Xue, S., Abdel-Hamid, O., Jiang, H., Dai, L., and Liu, Q. (2014). Fast adaptation of deep neural network based on discriminant codes for speech recognition. *Audio, Speech, and Language Processing, IEEE/ACM Trans. on*, 22(12):1713–1725.

Yi, J., Ni, H., Wen, Z., Liu, B., and Tao, J. (2016). CTC regularized model adaptation for improving LSTM RNN based multi-accent mandarin speech recognition. In *Chinese Spoken Language Processing (ISCSLP), 2016 10th International Symposium on*, pages 1–5. IEEE.

Yu, D., Yao, K., Su, H., Li, G., and Seide, F. (2013). KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *Proc. ICASSP*, pages 7893–7897.

Zhang, X., Trmal, J., Povey, D., and Khudanpur, S. (2014). Improving deep neural network acoustic models using generalized maxout networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 215–219. IEEE.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C. L. Y., and Courville, A. (2017). Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720*.