A TEACHER-STUDENT LEARNING APPROACH FOR UNSUPERVISED DOMAIN ADAPTATION OF SEQUENCE-TRAINED ASR MODELS

Vimal Manohar[1,2]*, Pegah Ghahremani*[1]*, Daniel Povey*[1,2]*, Sanjeev Khudanpur*[1,2]

[1]Center for Language and Speech Processing
[2]Human Language Technology Center Of Excellence
Johns Hopkins University, Baltimore, MD 21218
{vimal.manohar91,pegahgh,dpovey}@gmail.com,khudanpur@jhu.edu

ABSTRACT

Teacher-student (T-S) learning is a transfer learning approach, where a teacher network is used to "teach" a student network to make the same predictions as the teacher. Originally formulated for model compression, this approach has also been used for domain adaptation, and is particularly effective when parallel data is available in source and target domains. The standard approach uses a frame-level objective of minimizing the KL divergence between the frame-level posteriors of the teacher and student networks. However, for sequence-trained models for speech recognition, it is more appropriate to train the student to mimic the sequence-level posterior of the teacher network. In this work, we compare this sequence-level KL divergence objective with another semi-supervised sequence-training method, namely the lattice-free MMI, for unsupervised domain adaptation. We investigate the approaches in multiple scenarios including adapting from clean to noisy speech, bandwidth mismatch and channel mismatch.

Index Terms— sequence training, lattice-free, transfer learning, unsupervised adaptation, automatic speech recognition

1. INTRODUCTION

Transfer learning is the general machine learning approach of transferring knowledge from one model to another. Depending on the context it is used in, it might be called different things. In the case where we have to learn a smaller model on the same domain, the approach is called "model compression". In the case, where we have to learn a model in a different domain, the approach is called "domain adaptation". There is a rich survey of transfer learning methods in the literature [1, 2, 3]. Transfer learning methods have been applied to speech processing in various settings. Wang et. al [4] gives a good overall survey of methods used in speech processing.

This work was partially supported by NSF Grant No CRI-1513128 and IARPA MATERIAL award number FA8650-17-C-9115.

Sequence discriminative training, e.g. using Maximum Mutual Information (MMI) [5], has been shown to improve performance of frame-level cross-entropy trained neural networks. Of late, neural networks are trained from scratch using sequence objectives like Connectionist Temporal Classification (CTC) [6] and Lattice-free MMI (LF-MMI) [7], and these usually out-perform the frame-level trained ones. LF-MMI training has been investigated for supervised domain adaptation [8] and semi-supervised training [9]. Semi-supervised training methods like in [9] can also be applied when the unsupervised data is from a slightly different domain than the supervised data used to train the seed network. This was investigated for speaker adaptation in [10] using 1-best hypotheses from decoding. In this paper, we investigate this idea of using semi-supervised LF-MMI with lattice-based supervision for unsupervised domain adaptation.

One of the methods for transfer learning is the teacher-student (T-S) approach where a teacher network is used to "teach" a student network to make the same predictions as the teacher. It is traditionally used for model compression [11] as in [12, 13]. It has also been applied in context of domain adaptation [14], where the teacher network is trained on the source domain and the student network is trained on the target domain. It is particularly effective when parallel data is available in source and target domains [15]. Here, a large amount of unsupervised data in parallel source and target domains is used to improve performance of the model on target domain. However, these works do not compare with standard semi-supervised training methods that can also be used to do unsupervised adaptation. One of the goals of this work is to compare T-S learning objective with a standard semi-supervised training approch like using LF-MMI.

The standard approach for T-S learning uses an objective of minimizing the KL divergence between the frame-level posteriors of the teacher and student networks. However, this may not be applicable to state-of-the-art speech recognition models that are trained at the sequence-level, and in particular using LF-MMI. Alternatively, in this work, we use the KL divergence between sequence-level posteriors [16, 10] from the

teacher and student networks as the training objective. The similarity of this objective to LF-MMI allows it to be integrated easily into the lattice-free training framework.

In this paper, we investigate two sequence objectives for teacher-student type transfer learning for unsupervised adaptation – semi-supervised LF-MMI and sequence-level KL divergence. We describe our methods in Section 3. In Section 4, we describe experiments to evaluate our proposed method in the scenario of domain adaptation. We look at three scenario for adaptation – clean to noisy speech, 8kHz to 16kHz audio, and headset microphone to distant microphone. Finally, in Section 5, we present conclusions and future work.

2. RELATED WORKS

A sequence-KL objective for T-S learning was introduced in [16] for model compression from an ensemble. Unlike that work which used lattice-based discriminative training, here we apply sequence-level KL divergence in the lattice-free training framework for unsupervised domain adaptation. In [10], a lattice-free sequence-KL objective was introduced for model compression and speaker adaptation. Our work in this paper differs in how the supervision for training the student is generated. In particular, we propose a simpler way to get the supervision using the lattice supervision approach used for semi-supervised LF-MMI training in [9]. We also investigate the effect of using different LMs both when creating the numerator supervision and the denominator graph.

KL divergence objective is also viewed as a regularizer, which prevents the model from diverging too much from what the original model predicts [14]. A sequence-level KL version of this idea was used to regularize LF-MMI based DNN adaptation in [17, 10] to small adaptation sets. On the other hand, our work in this paper focuses on unsupervised domain adaptation when we have large unsupervised target-domain dataset. We also train our neural networks from scratch since the input features to the student network might be different from that of the teacher network (e.g. 16kHz vs 8kHz). In this context, we can view the sequence-level KL objective to be regularizing semi-supervised LF-MMI training to prevent to the model from over-fitting to the unsupervised data.

3. PROPOSED METHODS

3.1. Semi-supervised Lattice-free MMI

Semi-supervised LF-MMI training was proposed in [9]. Here, we extend that work to the scenario of domain adaptation when there is parallel unsupervised data in source and target domains. In [9], a seed network trained on the supervised data is used to decode the unsupervised data to generate lattices containing the hypothesized phone sequences. The lattices are converted into numerator graphs (denoted $\mathcal{G}_{\text{NUM}}$) using the smart-splitting method described in [9] and composing with a normalization FST [7] whereby we interpolate the word LM scores from the lattice and phone LM scores from the denominator graph $\mathcal{G}_{\text{DEN}}$. The LF-MMI objective is:

$$\mathcal{F}_{\text{MMI}} = \sum_r \log \sum_{\pi \in \mathcal{G}_{\text{NUM}}} P(\pi \mid \mathbf{O}_r) \tag{1}$$

$$= \sum_r \log \frac{\sum_{\pi \in \mathcal{G}_{\text{NUM}}} P(\mathbf{O}_r \mid \pi) P(\pi)}{P(\mathbf{O}_r)} \tag{2}$$

where $\mathbf{O}_r$ is the sequence of acoustic observations for utterance r, π is a HMM state sequence in the numerator graph $\mathcal{G}_{\text{NUM}}$ or denominator graph $\mathcal{G}_{\text{DEN}}$ and the likelihood $P(\mathbf{O}_r) \approx \sum_{\pi \in \mathcal{G}_{\text{DEN}}} P(\mathbf{O}_r \mid \pi) P(\pi)$. We extend this trivially for domain adaptation. Here, a seed network (referred to as the teacher model) is trained on the supervised data from the source domain. This is used to decode the unsupervised data in the source domain to generate lattices containing the hypothesized state sequences. These lattices are also the hypotheses for the corresponding parallel data in the target domain. So they are converted into supervision for training the student model in the target domain. The student model is trained with this as the supervision and the parallel target domain data as input. Such a use of parallel data for LF-MMI training was also found to be useful for far-field ASR in [18].

3.2. Sequence-KL objective

Sequence-KL objective was proposed for T-S learning in [16]. The objective here is to make the student network mimic the teacher network by maximizing the negative KL divergence between sequence-level posteriors from the teacher and student networks as shown in (3). We describe in this section our implementation in the lattice-free training framework and how it differs from those in other similar works in [16, 10].

$$\mathcal{F}_{\text{KL}} = -\sum_r \sum_{\pi \in \mathcal{G}_{\text{NUM}}} P(\pi \mid \mathbf{O}_r; \lambda^*) \log \left[\frac{P(\pi \mid \mathbf{O}_r; \lambda^*)}{P(\pi \mid \mathbf{O}_r; \lambda)} \right], \tag{3}$$

$$\propto \sum_r \Bigg(\sum_{\pi \in \mathcal{G}_{\text{NUM}}} P(\pi \mid \mathbf{O}_r; \lambda^*) \log P(\mathbf{O}_r \mid \pi; \lambda) - \log P(\mathbf{O}_r; \lambda) \Bigg), \tag{4}$$

where $P(\pi \mid \mathbf{O}_r; \lambda^*)$ and $P(\pi \mid \mathbf{O}_r; \lambda)$ are posterior probabilities of the HMM state sequence π obtained from the teacher network (parameterized by λ^*) and the student network (paramterized by λ) respectively. The former quantity is a constant since the teacher network is fixed when training the student. The simplification[1] to (4) makes it clear that the objective consists of numerator and denominator terms.

[1] using Bayes rule and removing the constant additive terms. Also $\sum_{\pi \in \mathcal{G}_{\text{NUM}}} P(\pi \mid \mathbf{O}_r; \lambda^*) = 1$

3.2.1. Denominator term

The denominator term $\log P(\mathbf{O}_r; \lambda)$ i.e. the log-likelihood under the student network is independent of the teacher network. In [16], this term was computed using a denominator lattice generated using a unigram LM. However, in lattice-free training, we compute this over a fixed denominator graph, $\mathcal{G}_{\text{DEN}}$, just as in the case of LF-MMI. The reader is directed to [7] for details of this forward-backward [19] computation on a GPU. As in [7], the denominator graph is created using a 4-gram phone LM. To bias it to the target domain, we use interpolated counts from source and target domains as in [20].

3.2.2. Numerator term

We compute the first term in (4) i.e. the numerator term as a summation over HMM state sequences $\pi = s_1 \dots s_T$ in the numerator graph $\mathcal{G}_{\text{NUM}}$ created by decoding the utterance using the teacher network. This is the same numerator graph that is generated for the semi-supervised LF-MMI training described in Section 3.1. This is also where we differ from [10]. In [10], this summation is done over the weak denominator graph $\mathcal{G}_{\text{DEN}}$. However, we are doing this summation over a lattice-based supervision that is generated using a strong 3-gram or 4-gram word LM. Our results in Section 4 show that using a strong LM here is generally better. This is also easier to implement since the lattice-based supervision is already generated for semi-supervised LF-MMI training.

3.2.3. Derivative computation

Since teacher network is fixed, the derivative of $\mathcal{F}_{\text{KL}}$ w.r.t. the student network output of utterance r at time t, $y_{rt}(j;\lambda)$, is:

$$\frac{\partial \mathcal{F}_{\text{KL}}}{\partial y_{rt}(j;\lambda)} = \gamma_{rt}^{NUM}(j;\lambda^*) - \gamma_{rt}^{DEN}(j;\lambda), \tag{5}$$

where $\gamma_{rt}^{NUM}(j;\lambda^*)$, the numerator posterior, is the posterior probability of senone j at time t computed over the numerator graph $\mathcal{G}_{\text{NUM}}$ using the teacher network and $\gamma_{rt}^{DEN}(j;\lambda)$, the denominator posterior, is the posterior probability of senone j at time t computed over the denominator graph $\mathcal{G}_{\text{DEN}}$ using the student network. These are computed as:

$$\gamma_{rt}^{NUM}(j;\lambda^*) = \frac{\sum_{\pi \in \mathcal{G}_{\text{NUM}}} \delta_{rt}(j) P(\mathbf{O}_r \mid \pi; \lambda^*) P(\pi)}{\sum_{\pi' \in \mathcal{G}_{\text{NUM}}} P(\mathbf{O}_r \mid \pi'; \lambda^*) P(\pi')}, \tag{6}$$

$$\gamma_{rt}^{DEN}(j;\lambda) = \frac{\sum_{\pi \in \mathcal{G}_{\text{DEN}}} \delta_{rt}(j) P(\mathbf{O}_r \mid \pi; \lambda) P(\pi)}{\sum_{\pi' \in \mathcal{G}_{\text{DEN}}} P(\mathbf{O}_r \mid \pi'; \lambda) P(\pi')}, \tag{7}$$

where $\delta_{rt}(j)$ is 1 iff HMM state s_t in sequence π corresponds to senone j and 0 otherwise. Both the numerator and denominator posteriors are computed over their respective graphs using forward-backward algorithm [19].

3.2.4. Lattice-free MMI and sequence-KL

From (5), we can see that the derivative is the difference of the numerator posterior computed using the *teacher network* and the denominator posterior computed using the student network. Note that this differs only in the first term from the derivative of the MMI objective (2):

$$\frac{\partial \mathcal{F}_{\text{MMI}}}{\partial y_{rt}(j;\lambda)} = \gamma_{rt}^{NUM}(j;\lambda) - \gamma_{rt}^{DEN}(j;\lambda), \tag{8}$$

where the first term is the numerator posterior computed using the *student network* i.e. with λ instead of λ^* in (6).

To use an interpolation of the two objectives, we can simply interpolate the numerator posteriors from teacher and student networks. This is a sequence-level analogue to the knowledge distillation idea [13], and this was also explored in [21]. In our work, we always compute the numerator of the LF-MMI objective using a supervision lattice generated using a strong 3-gram word LM. But in Section 4, we explore computing the numerator of the sequence-KL objective using a different supervision lattice such as one generated using a weak LM like a unigram LM.

4. EXPERIMENTS

We compare semi-supervised LF-MMI and sequence-level KL divergence for domain adaptation in the following scenario – Clean to noisy speech, 8kHz Fisher to 16kHz AMI, and headset microphone to distant microphone speech.

All the neural networks in our experiments have an architecture with time-delay neural network (TDNN) [22, 23] layers interleaved with LSTM [24] layers. We use per-frame dropout on the LSTM layers [25]. The reader is directed to [25] for training details. The scripts and code used for these experiments can be found in a personal Kaldi branch[2]. To avoid over-fitting, we apply the regularization methods suggested in [7] for both LF-MMI and sequence-KL training. We use online i-vectors [26, 27, 28] for speaker adaptation. The teacher and the student networks use i-vectors extracted from different i-vector extractors trained on their respective domains. Our method for creating supervision is described in sections 3.1 and 3.2. In some of the experiments, we use a semi-supervised style training where the supervised training uses LF-MMI objective computed on lattices generated by force-aligning the word transcription using a HMM-GMM system and the unsupervised training uses LF-MMI or sequence-KL objective computed on lattices generated by decoding as described in Section 3.1.

4.1. Clean to noisy speech

In this section, we report results of unsupervised adaptation from clean to noisy speech on Aspire corpus [29]. The training data consists of 1500 hours of Fisher English [30]. Of this,

[2] https://github.com/vimalmanohar/kaldi/tree/semisup-ts

we use 300 hours as supervised data with transcription and 1200 hours without transcription. The "Baseline" system is trained with LF-MMI objective on 300 hours supervised data augmented 3x with reverberation and noise [31] and 3x with speed and volume perturbation [32] (Hence 9x300 hours). The "Oracle" system is trained with LF-MMI objective using as supervised data all 1500 hours augmented 3x with reverberation and noise. We use the same i-vector extractor for baseline, oracle and all the student networks. This is trained on 1500 hours of Fisher data augmented 3x with reverberation and noise.

The teacher network is trained on "clean" 300 hours supervised data with only 3x speed perturbation, but with no reverberation or noise addition. This network is used to decode the whole 1500 hours[3] of "clean" Fisher data. For this decoding, we use a 3-gram LM trained on transcripts from the 300 hours supervised set. We create the supervision for training the student network as described in Section 3.1 with 1500 hours of reverberated and noise corrupted data (augmented 3x) parallel to the "clean" data. The denominator graph is generated using 4-gram phone LM created by averaging counts from supervised data alignments and 1-best alignments from unsupervised data. We additionally interpolate the phone LM scores with the word LM scores in the lattice with a scale of 0.5 as found to be optimum in [9]. The supervised training uses LF-MMI with supervision from a GMM system, while the unsupervised training uses an interpolated objective $(1-\beta)\mathcal{F}_{\text{MMI}} + \beta\mathcal{F}_{\text{KL}}$. The β used in each experiment is shown in Table 1. The columns "sup" and "unsup" show the amount of supervised and unsupervised data (prior to augmentation) respectively used in training the student network. The results in Table 1 show WER(%) on *dev* and *test* sets, which are 3 hour subsets heldout from the Fisher English corpus, but reverberated and corrupted with noise. These are part of Kaldi [33] Aspire recipe. We also report results on the official *aspire* development set [29].

From rows 2 and 3 showing results from training **only** on the unsupervised data, we see that sequence-KL is significantly better than using LF-MMI. The WER with LF-MMI is worse than even the baseline on the *aspire* set. But with semi-supervised training by including supervised data in a multi-task architecture [9] (Rows 4-6), we always get significant improvement over the baseline. Further, using either sequence-KL (Row 5) or an interpolation of LF-MMI and sequence-KL (Row 6) is slightly better than using LF-MMI (Row 3). We then tried to use a unigram LM instead of a 3-gram LM for decoding when generating numerator posteriors for sequence-KL objective, while still using 3-gram for generating lattices for MMI training. From the row 7 in Table 1, we see that this does not work as well. Since, it is easier to do decoding just once, we recommend just using the 3-gram LM for generated lattices for both MMI and sequence-KL training.

[3]Note that this includes the 300 hours of audio from supervised dataset, but we are only using the audio and not the transcripts. This is like using soft posteriors for labeled data in conventional T-S learning [13].

Table 1. WER(%) results for unsupervised adaptation from clean to noisy. The objective is $(1-\beta)\mathcal{F}_{\text{MMI}} + \beta\mathcal{F}_{\text{KL}}$.

System	β	hrs		WER(%)		
		sup	unsup	*dev*	*test*	*aspire*
Baseline	0.0	300	0	23.6	22.5	26.6
Unsup	0.0	0	1500	23.0	22.0	27.0
Unsup	1.0	0	1500	21.8	21.0	25.9
Semisup	0.0	300	1500	21.6	21.0	25.1
Semisup	1.0	300	1500	21.0	20.3	24.4
Semisup	0.5	300	1500	**21.0**	**20.2**	**24.2**
+ UG	0.5	300	1500	21.2	20.6	24.5
Oracle	0.0	1500	0	19.1	18.4	23.3

4.2. 8kHz Fisher to 16kHz AMI

In this section, we report results for domain adaptation from 8kHz Fisher to 16kHz AMI [34] individual headset microphone (IHM) speech. There are multiple sources of mismatch here including bandwidth, channel and language domain. However, we only have parallel data to deal with the bandwidth mismatch (8kHz vs. 16kHz).

The preliminary results are in Table 2. The "Baseline" network here is same as the one in Section 4.1. This is also the teacher network for T-S learning and is used to decode the target AMI-IHM data (downsampled to 8kHz to use with Fisher's teacher network) to generate lattices for training student network. A 3-gram Fisher LM is used for this decoding. As in Section 4.1, we again interpolate with a 0.5 weight the phone LM scores with word LM scores from the lattice when creating the numerator supervision. While the supervision is created using the 8kHz AMI-IHM data, we use the parallel 16kHz AMI-IHM data for training the student network. As input to the student network, we use i-vectors extracted using an i-vector extractor trained on 16kHz AMI-IHM data.

The rows 2-4 compare using LF-MMI, sequence-KL and an interpolated objective for training student network on the unsupervised target data from AMI-IHM. We get 1-2% absolute improvement over the "Baseline" using either of these. However, all of these systems are quite behind the "Oracle" system (Row 5), which is trained only on the AMI-IHM in a supervision fashion using LF-MMI. This is the case even when training the "Oracle" system on 8kHz data (Row 6), which only degrades performance by less than 2% over using 16kHz data. This suggests that bandwidth mismatch by itself is not a major issue in these experiments, but that other forms of domain mismatch such as language mismatch (dialect, topics etc.) between Fisher and AMI is more prominent.

Table 2. Preliminary WER(%) results for 8kHz Fisher to 16kHz AMI-IHM

System	Target domain			WER(%)	
	sup (hrs)	un-sup (hrs)	Rate (kHz)	*dev*	*eval*
Baseline	0	0	8	30.6	33.3
MMI	0	80	16	29.8	31.3
KL	0	80	16	29.4	31.5
0.5*(MMI+KL)	0	80	16	29.3	31.2
Oracle	80	0	16	18.7	18.6
Oracle	80	0	8	20.4	19.9

4.2.1. LM for numerator computation

For semi-supervised training, it is generally better to use a strong LM for decoding unsupervised data to generate lattices. From [9], the best phone LM scale for interpolating normalization FST's phone LM scores and lattice's word LM scores when generating numerator supervision is 0.5.

In this section, we try to find:

1. the best phone LM scale (0.0 vs 0.5) for interpolating phone LM and word LM scores to get numerator posteriors for sequence-KL
2. the best LM (3-gram vs 1-gram) to use for generating lattices to get numerator posteriors for sequence-KL

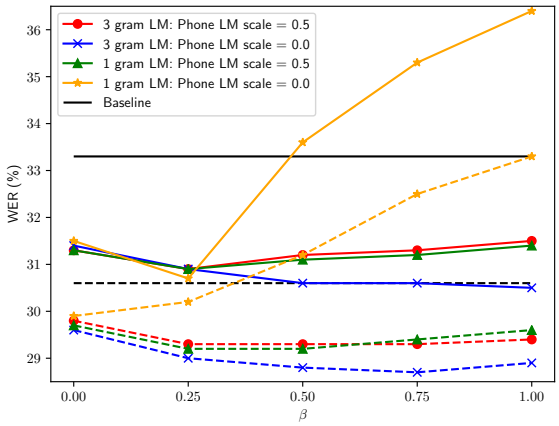
The legend in Figure 1 shows the LM used for decoding and the phone LM scale when generting supervision for sequence-KL objective. In the experiments in this section, we use the interpolated objective $(1-\beta)\mathcal{F}_{\text{MMI}}+\beta\mathcal{F}_{\text{KL}}$. Note that the LF-MMI objective in all cases uses a 3-gram word LM for decoding and a phone LM scale of 0.5 for interpolating phone LM and word LM scores.

From Figure 1, when using a 3-gram word LM for decoding, using a phone LM scale of 0.0 (Blue $\times$) works better than a phone LM scale of 0.5 (Red $\circ$). However, when using a 1-gram LM for decoding, the WER degrades with a phone LM scale of 0.0 (Orange $*$) and gets even worse than baseline for large β. This problem is alleviated if a phone LM scale of 0.5 is used (Green $\triangle$), but is still worse than using 3-gram word LM for decoding.

From this, we conclude that for unsupervised domain adaptation, it is better to use a strong LM like 3-gram for generating numerator supervision. This is also computationally advantageous because using strong 3-gram LM requires only a single generation of lattices for both MMI and sequence-KL, while using 1-gram LM requires regeneration of lattices for sequence-KL. Further, when using a strong word LM, interpolating the LM scores with phone LM scores is not required and using a phone LM scale of 0.0 works the best.

The performance degradation when using a weak LM was also reported in [10] for unsupervised speaker adaptation. But we believe the degradation is larger in our case because we are training the student network from scratch instead of initializing from the teacher network. However, initializing from teacher network is not straight-forward in our case since the input features are different (16kHz vs 8kHz).

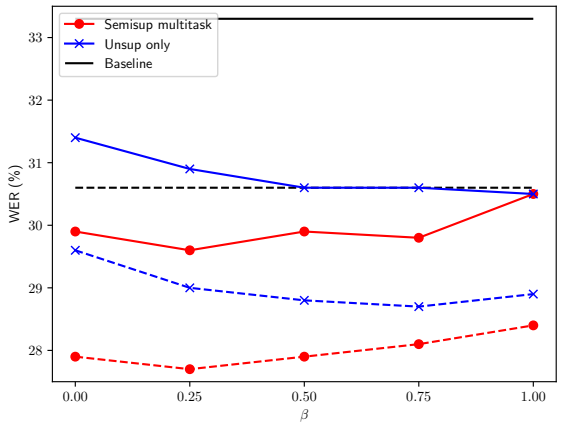
Fig. 1. 8kHz Fisher ->16kHZ AMI-IHM WER(%) results: Unigram vs 3-gram for sequence-KL. The solid lines show results on *eval* and dashed lines on *dev*.



4.2.2. Multitask learning for domain mismatch

Multitask learning is one of the methods for transfer learning in domain mismatch conditions. In [8], this was used for supervised adaptation. Here we apply it in the semi-supervised setting by training on the Fisher data (upsampled to 16kHz) using LF-MMI and on the AMI-IHM data using an interpolated objective $(1-\beta)\mathcal{F}_{\text{MMI}}+\beta\mathcal{F}_{\text{KL}}$. Based on the results in Section 4.2.1, we get numerator posteriors for sequence-KL from lattices obtained by decoding using a 3-gram Fisher word LM and using LM scores only from the word LM. We share all the layers including the output for both Fisher and AMI tasks. Figure 2 compares the two for various interpolation factors. We see that semi-supervised training in the multitask-type architecture (Red $\circ$) is better than training only on the unsupervised data (Blue $\times$) in the target domain, giving an improvement of around 3% over the "Baseline". It is possible that training on a larger amount of data and also regularizing with supervised Fisher data (even if out-of-domain) is helping the cause here. Since smaller β is better, we can say LF-MMI is more effective than using sequence-KL for multitask training in this domain mismatch case. We believe that since the domains of the data used to train the teacher and student networks are different (Fisher vs. AMI), the numerator posteriors from the teacher are not very good for training the student using sequence-KL. But, we can get better posteriors from the student network by training using LF-MMI.

Fig. 2. 8kHz Fisher ->16kHz AMI-IHM WER(%) results: Unsupervised vs semi-supervised multitask training. The solid lines show results on *eval* and dashed lines on *dev*.



4.3. Headset to Distant microphone speech

In this section, we report results on domain adaptation from AMI individual headset microphone (IHM) speech to AMI single distant microphone (SDM) speech. For the baseline system, we use AMI-SDM data mixed with AMI-IHM data augmented with reverberation and noise. For supervision, we use lattices generated from a GMM system for AMI-IHM data and use it for parallel reverberated AMI-IHM data and AMI-SDM data as done in [18]. For unsupervised domain adaptation experiments, we consider two unsupervised dataset – Mixer 6 [35] (We use only the telephone calls portion) and ICSI [36]. As teacher network, we use a TDNN-LSTM network trained on AMI-IHM data that is mixed with revererated and noise augmented version of the same. This teacher network was selected as it gave the best performance on AMI-IHM *dev* and *eval* sets. For adaptation using mixer 6, we decode the mixer 6 headset microphone (MIC02) data using the teacher network and a 3-gram Fisher word LM to generate lattices. These lattices are converted into supervision for data from the parallel far-field microphones (MIC04-MIC13). Since the same data was recorded in multiple microphones, we only kept a subset of 30% of the parallel far-field data. For adaptation using ICSI, we decode the ICSI-IHM data using the teacher network and the 3-gram AMI word LM to generate lattices. These lattices are converted into supervision for data from the parallel ICSI-SDM data. We used all 4 available distant microphones, but adjusted the training to train on these for one-fourth the number of epochs as the rest of the data. In both the baseline and T-S learning networks, we use the same i-vector extractor, which is trained on the same AMI data used to train the baseline. For both the adaptation, we use semi-supervised training in a multi-task architecture wth supervised training with LF-MMI on the same AMI data as the baseline network and unsupervised training with an interpolated objective $(1-\beta)\mathcal{F}_{\text{MMI}} + \beta\mathcal{F}_{\text{KL}}$ on a mix of augmented (using reverberation and noise addition) headset mic data and distant mic data from Mixer 6 or ICSI corpora.

Table 3. IHM ->SDM adaptation: WER(%) on AMI-SDM *dev* and *eval* sets

Method	β	Hours		*dev*	*eval*
		sup	unsup		
Baseline	0.0	80	0	34.0	37.2
T-S (Mixer 6)	0.5	80	110	31.8	35.3
T-S (ICSI)	0.5	80	80	32.5	36.3

The results in Table 3 show that T-S learning using parallel data in Mixer 6 or ICSI for adaptation from IHM to SDM improves WER over the baseline which uses only AMI data. The improvement is found to be larger when using ICSI corpus, probably because of the similarity of ICSI and AMI corpora.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a teacher-student learning approach for unsupervised domain adaptation. Here, we use a teacher network to decode the source domain data to generate supervision. The supervision is used with parallel target domain data to train a student network using lattice-free MMI, sequence-KL divergence or an interpolation of the two objectives. We evaluated the performance on various domain adaptation scenarios. Our main conclusion is to use semi-supervised training in a multitask architecture with supervised training using LF-MMI and unsupervised training using an interpolation of LF-MMI and sequence-KL objectives with 0.5 weight. When parallel data is available to deal with feature domain mismatch such as for adaptation from clean to noisy speech, we observe that sequence-KL is very effective even when used for purely unsupervised training. However when there is also large language domain mismatch such as for adaptation from Fisher to AMI, semi-supervised LF-MMI is preferable to using sequence-KL. For sequence-KL objective, our proposed approach of using a strong LM for getting numerator posteriors is better than using a weak LM both in terms of WER and computational cost as we can re-use the supervision generated for LF-MMI.

We hypothesize that sequence-KL might be helpful in the beginning of training when initializing the network from scratch, and will try in the future to vary the interpolation factor with LF-MMI as the training progresses. In the future, we will also explore cases of adaptation using semi-supervised LF-MMI and sequence-KL for retraining the network without initializing from scratch. Further, we will explore these ideas in the context of model compression.

6. REFERENCES

[1] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang, "Transfer learning using computational intelligence: a survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[3] Yoshua Bengio et al., "Deep learning of representations for unsupervised and transfer learning.," *ICML Unsupervised and Transfer Learning*, vol. 27, pp. 17–36, 2012.

[4] Dong Wang and Thomas Fang Zheng, "Transfer learning for speech and language processing," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 1225–1237.

[5] L. Bahl, P. Brown, P.V. de Souza, and R. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model parameters for Speech Recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, Apr 1986, vol. 11, pp. 49–52.

[6] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*. ACM, 2006, pp. 369–376.

[7] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI," in *Proc. INTERSPEECH*, 2016, pp. 2751–2755.

[8] Pegah Ghahrehmani, Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur, "Investigation of Transfer Learning for LF-MMI Trained Neural Networks for ASR," in *Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop on*, 2017.

[9] Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur, "Semisupervised training of acoustic models using lattice-free mmi," in *ICASSP*, 2018.

[10] Naoyuki Kanda, Yusuke Fujita, and Kenji Nagamatsu, "Investigation of lattice-free maximum mutual information-based acoustic models with sequence-level kullback-leibler divergence," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 69–76.

[11] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 535–541.

[12] Jimmy Ba and Rich Caruana, "Do deep nets really need to be deep?," in *Advances in neural information processing systems*, 2014, pp. 2654–2662.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[14] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7893–7897.

[15] Jinyu Li, Michael L Seltzer, Xi Wang, Rui Zhao, and Yifan Gong, "Large-scale domain adaptation via teacher-student learning," *arXiv preprint arXiv:1708.05466*, 2017.

[16] JHM Wong and MJF Gales, "Sequence student-teacher training of deep neural networks," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2016, vol. 8, pp. 2761–2765.

[17] Yanhua Long, Yijie Li, Hone Ye, and Hongwei Mao, "Domain adaptation of lattice-free mmi based tdnn models for speech recognition," *International Journal of Speech Technology*, vol. 20, no. 1, pp. 171–178, 2017.

[18] Vijayaditya Peddinti, Vimal Manohar, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Far-field asr without parallel data.," in *INTERSPEECH*, 2016, pp. 1996–2000.

[19] Lawrence R Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[20] Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur, "JHU Kaldi System for Arabic MGB-3 ASR Challenge using Diarization, Audio-Transcript alignment and Transfer learning," in *Proc. ASRU 2017*, 2017.

[21] Naoyuki Kanda, Yusuke Fujita, and Kenji Nagamatsu, "Sequence distillation for purely sequence trained acoustic models," in *Proc. ICASSP 2018*, 2018.

[22] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[23] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH*, 2015, pp. 3214–3218.

[24] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[25] Gaofeng Cheng, Vijayaditya Peddinti, Daniel Povey, Vimal Manohar, Sanjeev Khudanpur, and Yonghong Yan, "An exploration of dropout with LSTMs," in *Proc. INTERSPEECH*, 2017.

[26] Martin Karafiat, Lukas Burget, Pavel Matejka, Ondrej Glembek, and Jan Cernocky, "iVector-based discriminative adaptation for automatic speech recognition," in *Proc. Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. Dec. 2011, pp. 152–157, IEEE.

[27] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[28] Vijayaditya Peddinti, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[29] Mary Harper, "The automatic speech recogition in reverberant environments (aspire) challenge," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 547–554.

[30] Christopher Cieri, David Miller, and Kevin Walker, "The fisher corpus: a resource for the next generations of speech-to-text.," in *LREC*, 2004, vol. 4, pp. 69–71.

[31] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5220–5224.

[32] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition.," in *INTERSPEECH*, 2015, pp. 3586–3589.

[33] D. Povey, A. Ghoshal, et al., "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[34] Iain McCowan, Jean Carletta, W Kraaij, S Ashby, S Bourban, M Flynn, M Guillemot, T Hain, J Kadlec, V Karaiskos, et al., "The ami meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, 2005, vol. 88, p. 100.

[35] L Brandschain, D Graff, C Cieri, K Walker, C Caruso, and A Neely, "The mixer 6 corpus: Resources for cross-channel and text independent speaker recognition," in *Proc. of LREC*, 2010.

[36] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al., "The icsi meeting corpus," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. IEEE, 2003, vol. 1, pp. I–I.